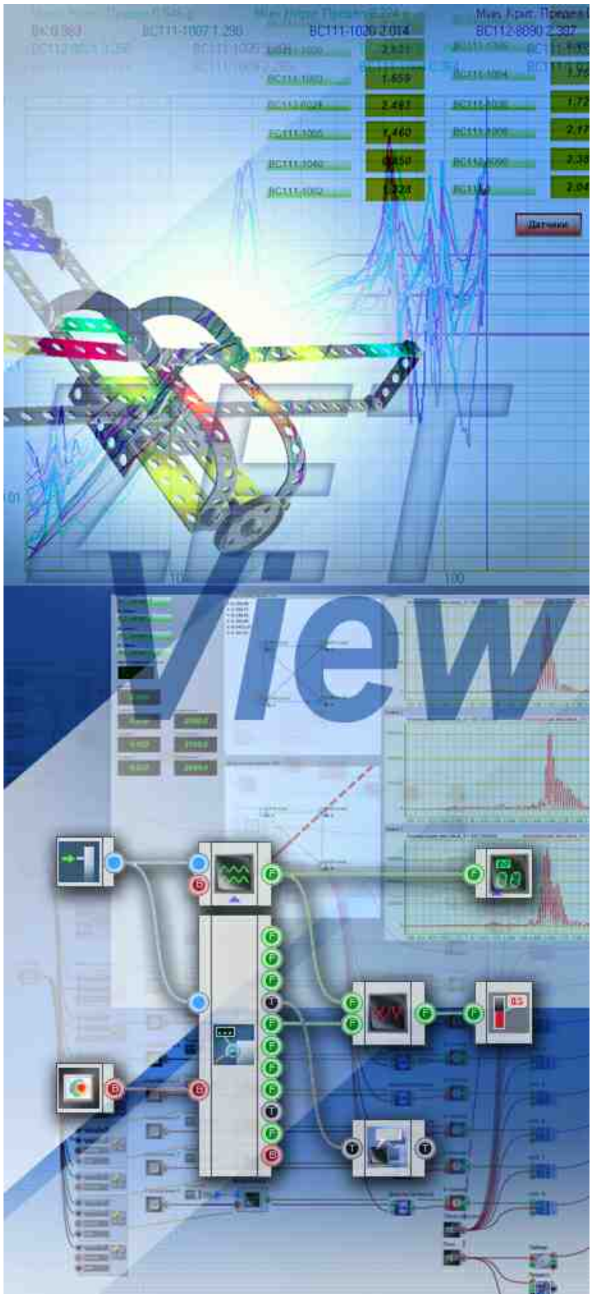


SCADA ZETView. User manual.

LLC "ETMS".05000-01 34 PO

Contents

SCADA ZETView. User manual

Help and technical support

About SCADA ZETView

ZETView is a powerful environment used for development of automated workstations based on PC with software and hardware measurement system. May be it would be interesting for the user to learn more of the system features. The main feature of SCADA ZETView is that it is actually a programming environment, hence, it allows to solve a wide variety of tasks such as creation of unique virtual instruments, measuring complexes, automated test systems, monitoring and control systems, ets.

Without implementation of special instruments, it takes a lot of time to solve most of usual tasks (e.g. to perform equipment testing and to obtain characteristics from several channels). It will take a lot of time and effort to perform these tasks manually. Automation of this process by developing a software program requires certain skills and takes a lot of time as well. Powerful graphical design environment has been developed for creation of applications to be implemented in various spheres of industry, which allows to considerably increase operational efficiency.

ZETView is a simple and user friendly graphical system with flexible programming language, which provides the user with simple and convenient solutions for a wide range of engineering tasks.

SCADA ZETView has been developed in order to provide you with simple solutions of your tasks. The system has a wide range of components, which can be used for development of projects of any complexity degree. ZETView contains libraries of various virtual instruments performing measurements and analysis of data received from hardware components using previously recorded signal. The components have a wide range of properties that can be adjusted in the course of project development. These features are classified into constant and those set by the operator in the course of project development.

ZETView has a large variety of display and indication options such as various graphics (including those in 3D format), graphs or indicators: pointer-type, digital, color and process indicators. All the display components have adjustable parameters. It is possible to implement project control by using graphical control and indication elements, which simplifies the implementation and perception of the project. Some companies have standards for data display and representation, thus, using SCADA ZETView, you can represent the results obtained in the required format.

SCADA ZETView is an optimal choice for the development of fully-automated processes allowing to simplify measurements, analysis, control and display of the results obtained. All you have to do is to start the program and the remaining process will be performed automatically.

Programming in ZETView

Development of applications in SCADA ZETView fundamentally differs from development process based on traditional programming language. Programming language is based on the use of program commands and parameters forming program code, while ZETView creates graphical components instead of text commands. Interaction between the components is established by linking them with each other, thus, the system consists of interrelated components. SCADA ZETView does not require any route work related to creation of correct code, memory distribution and other typical activities of software developer. Besides, troubleshooting is much easier in the case if the system components have graphical representation. ZETView has two display modes: design interface and operator's interface. Thus, the user working with your project does not have to be aware of its operating principles. Among special features of SCADA ZETView one should mention its ability to work with constant data flow – this very approach is applicable to most of the components (which has to be taken into consideration in the course of project algorithm development). Graphical design environment SCADA ZETView is user-friendly, so, even if the user does not have any programming or development experience, he can easily use it.

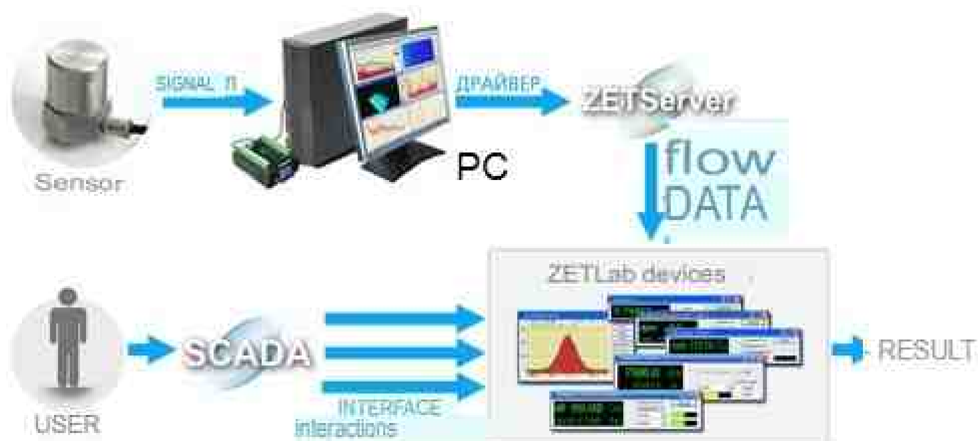
Structural layout of measurement system can be represented by a scheme. Then it is necessary to make the same very scheme in graphical design environment ZETView. Thus, in order to solve a task one should just clearly formulate it.

The links stand for data transfer, cause-effect ratios, sequence of actions.

The icons depict the function and simplify visual perception of the component.

The program represented by a mnemonic scheme clearly depicts the function of each component.

SCADA ZETView is closely connected with ZETLab software package. ZETLab is a set of virtual instruments integrated SCADA ZETView system. In fact, ZETView is a software shell for ZETLab.



Let us consider operation of ZETView by means of example of building structure dynamic characteristics evaluation. Seismic sensors receiving structure oscillations and sending signal to seismic receivers are installed near the bearing structures of the building. The signals from seismic receivers undergo digital processing by seismic station and are further forwarded to PC in real time mode via Ethernet. The driver allows to send the data to ZET Server. Data flow from channels server is sent to virtual tool.

As one can see from the figure above, SCADA ZETView system provides multiple access to ZETLab instruments, which means, that, if we have, e.g., 5 voltmeters in our project, then ZETView will start five copies of "Voltmeter" program from ZETLab scope (each program copy will have a memory of its own). This enables normal project operation even in the case if, due to some reasons, there is no signal from one of the voltmeters.

Working examples

1. Functions

The «Fundamental Tone» software is a set of programs, which are used for determination of free oscillations period and logarithmic decrement in accordance with the applicable standards.

Period and decrement of free oscillations' fundamental tone is an important parameter, which is used for analysis of building's stress-strain status changes as well as in the course of building's technical condition control.

2. Fundamental principles

For the purpose of fundamental tone period evaluation an indirect measurements method is used. The essence of this method is that the fundamental tone period for each of the axes is determined by means of building's oscillations spectrum analysis.

The decrement value for each of the mutually transverse axes is calculated by means of the same method, which is used for fundamental tone period calculations.

Building's oscillations are represented by a composition of free and forced oscillations, caused by inner and external impacts. Industrial buildings normally have a lot of manufacturing machinery inside (which is on and moves around the building). In this case, one should perform measurements when the manufacturing process is stopped, or when the manufacturing machinery is stable.

Among the factors that may distort the period measurements results one should mention the snow load stress – hence, this factor is to be taken into consideration during measurements performance.

3. Setting-up procedures

3.1 Installation of ZETLAB Software and ZETVIEW Software

Install ZETLab Software (if it hasn't been previously installed) from a USB stick to your PC, which is going to be used for further operation, then run ZETLab.msi installation file. Follow instructions of set-up wizard and install ZETLab to the directory C:\ZETLab.

3.2 Installation of «Fundamental Tone» Software

In order to use "Fundamental Tone" software it is necessary to copy the folder "Buildings, structures and environment monitoring system" to your PC.

To begin working with "Fundamental tone" software, run the file «Fundamental tone.exe». For more convenience, one can place the shortcut for the file «Fundamental tone.exe» on the desktop.

4. Operation of Fundamental Tone software

4.1 Starting the Software

In order to start «Fundamental Tone» software, double click the «Fundamental tone.exe» to run it (Figure 4.1)

Figure 4.1 «Fundamental tone.exe» file

You will see the window of «Fundamental Tone» program (Figure 4.2 12).

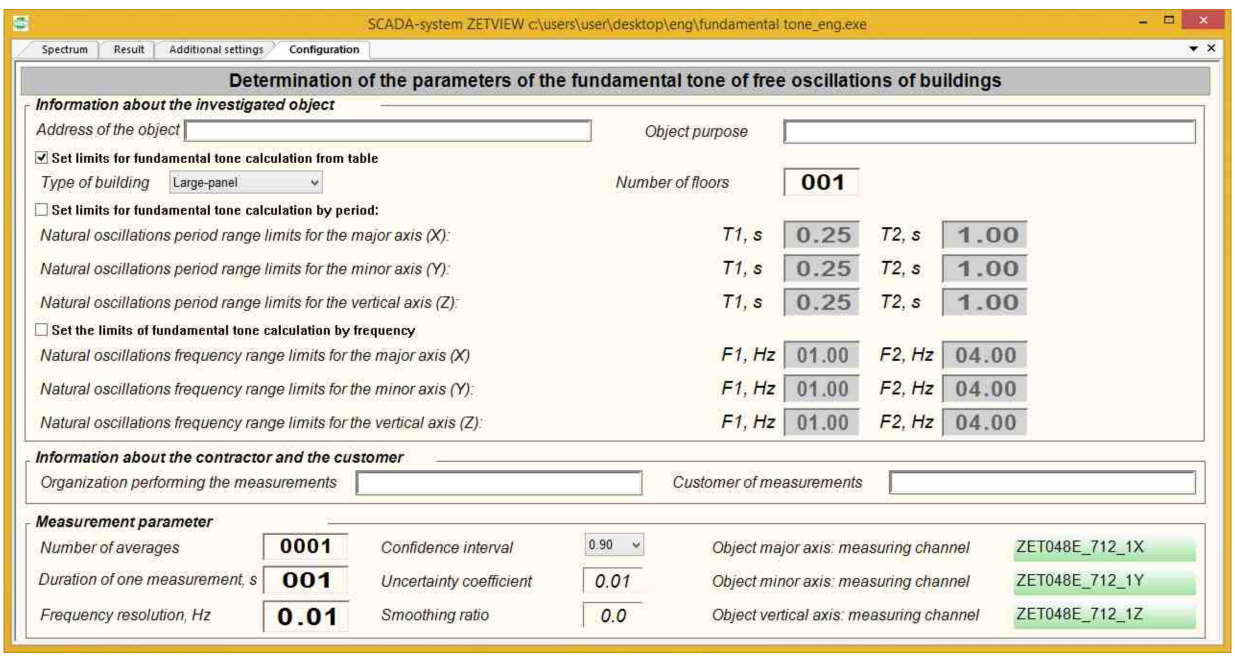


Figure 4.2 «Fundamental tone» program window

The program has several tabs:

- Configuration;
- Spectrum;
- Result;

- Additional settings.

4.2 Tabs description

4.2.1 «Configuration» tab

«Configuration» tab (Figure 4.3 12) is used to complete informational fields as well as to change settings.

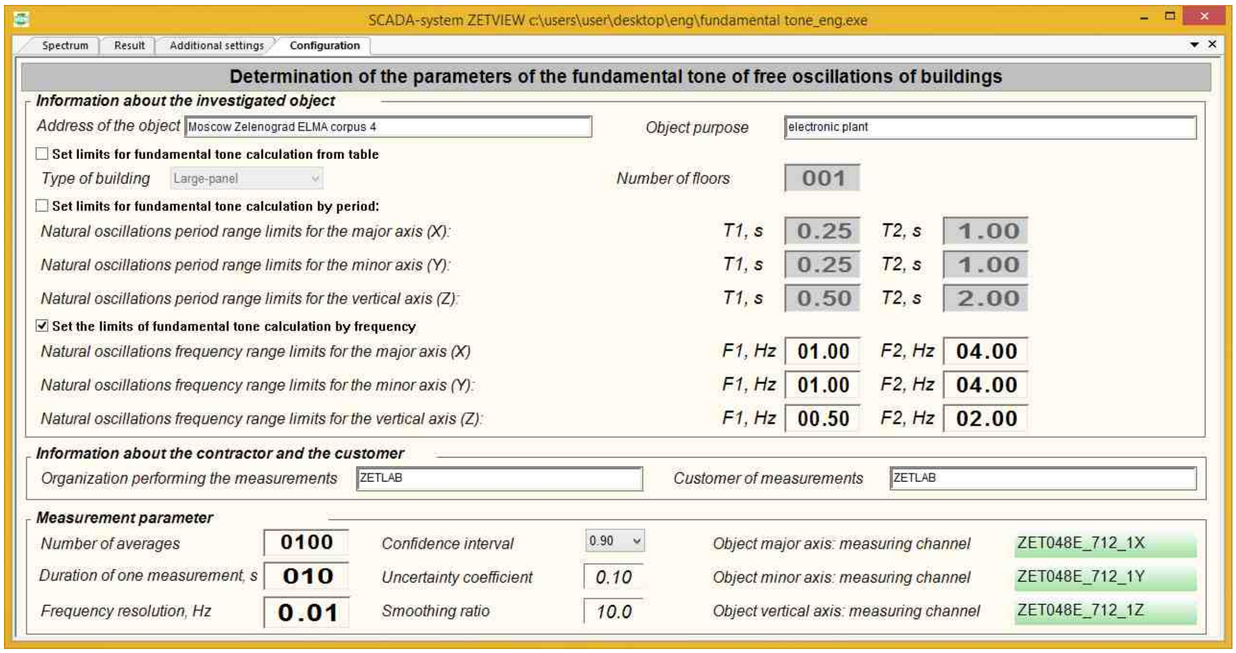


Figure 4.3 «Configuration» tab

The fields «Address of the object», «Object purpose», «Organization performing the measurements», «Customer of measurements» serve for informational purposes only and are used for report creation.

In the «Configuration» tab there is a section «Information about the investigated object» in which you should choose one of the three ways to set limits for fundamental tone calculation:

1) «Set limits for fundamental tone calculation from table»

This option is selected if the building has a standard structure. The pop-up list (Figure 4.4) is used to select corresponding type of the building as well as to set the number of floors.

Figure 4.4 Building's structure type - selection menu

2) «Set limits for fundamental tone calculation by period»

This option is selected in the case if the building's structure is not typical, but the natural oscillations limits are known. In this case, one have to set oscillations limit values for X, Y, Z axes in corresponding fields (Figure 4.5).

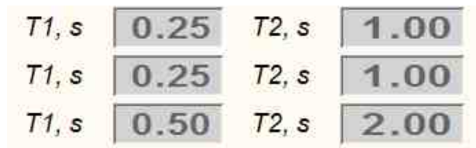


Figure 4.5 Free oscillations period: range limits

3) «Set the limits of fundamental tone calculation by frequency»

This option is selected in the case if the building's structure is not typical, but free oscillations frequency limits are known. In this case, one have to set oscillations frequency limit values for X, Y, Z axes in corresponding fields.

Figure 4.6 Free oscillations frequency: range limits

In the «Configuration» tab, there is a «Measurement parameter» section, where the following parameters are to be set:

- Frequency resolution – spectrum band density in the frequency axis. The lower is the frequency resolution value, the more bands there are.
- Duration of one measurement – the time for a single spectrum values averaging.
- Number of averages – the amount of recorded spectra, which is going to be used for values averaging.
- Confidence interval – confidence interval value stands for probability of the parameter in question being within the set interval. Confidence interval value is a number from 0 to 1 showing the probability of the measured variable being within the specified range. The more is the confidence interval, the higher is the probability that the actual value falls into particular interval.

- Uncertainty coefficient – is a ratio, which is equal to the product of frequency resolution by measurements duration. It corresponds to the amount of noise in the spectrum graphic. High value of uncertainty coefficient leads to stable measurements performance and decrease of noise impact. In order to consider the measurements results to be reliable, the uncertainty coefficient value should be ≥ 1.
- Smoothing ratio – is a value, which is equal to the product of uncertainty coefficient by the number of averages. It correlates to the noise suppression degree. In order to consider the measurements results to be reliable, the smoothing ratio value should be ≥ 100.
- Object major axis: measuring channel – in this section one should set the measuring channel to be used for oscillations control along major axis (X-axis).
- Object minor axis: measuring channel – in this section one should set the measuring channel to be used for oscillations control along minor axis (Y-axis).
- Object vertical axis: measuring channel – in this section one should set the measuring channel to be used for oscillations control along vertical axis (Z-axis).

4.2.2 «Spectrum» tab

«Spectrum» tab (Figure 4.7) shows spectra graphics for X, Y and Z measuring channels within the set measurements range. The obtained spectra graphics depend on the values set for the parameters «Frequency resolution», «Duration of one measurement», «Number of averages» in the «Configuration» tab.

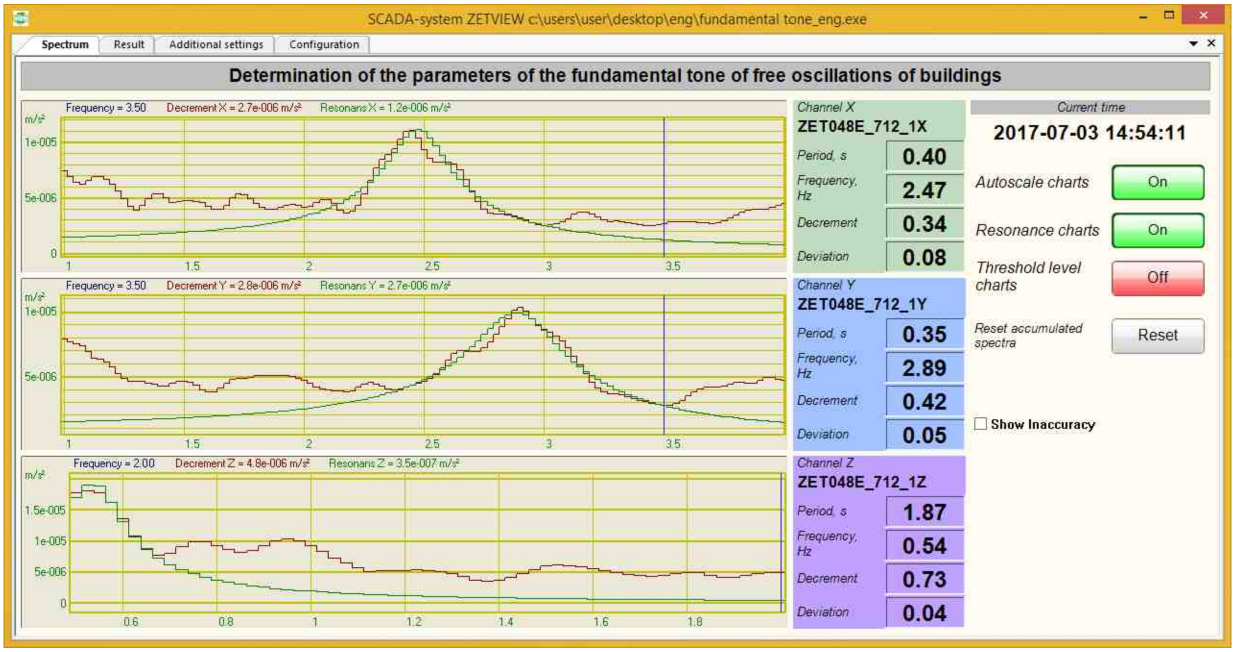


Figure 4.7 «Spectrum» tab

In order to depict resonance graphics one should enable corresponding option (Figure 4.8)

Figure 4.8 Key for resonance graphics representation

Threshold level graphic is a horizontal line shown in the spectrum graphic. The spectrum graphic section, which is located above the threshold level line, is used for further calculations. To depict the threshold level line one should enable corresponding option (Figure 4.9).

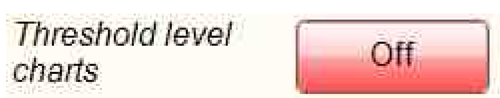


Figure 4.9 Key for threshold level graphics representation

Digital indicator «Period, s» stands for average period value of building's natural oscillations period for a set time span.

Digital indicator «Frequency, Hz» stands for average value of building's natural oscillations for a set time span.

Digital indicator «Decrement» stands for average value of building's natural oscillations decrement for a set time span.

Digital indicator «Deviation» stands for spectrum graphic deviation from the resonance graphic. The higher the «Deviation» indicator value is, the more reliable the measurements results are.

In order to reset the accumulated spectra and related parameters, one should use «Reset accumulated spectra» key (Figure 4.10 12).

Figure 4.10 Key used to reset the accumulated spectra

4.2.3 «Result» tab

The «Result» tab (Figure 4.11 12) shows dependence graphics of frequency, period and decrement values from the set point (measurement number).

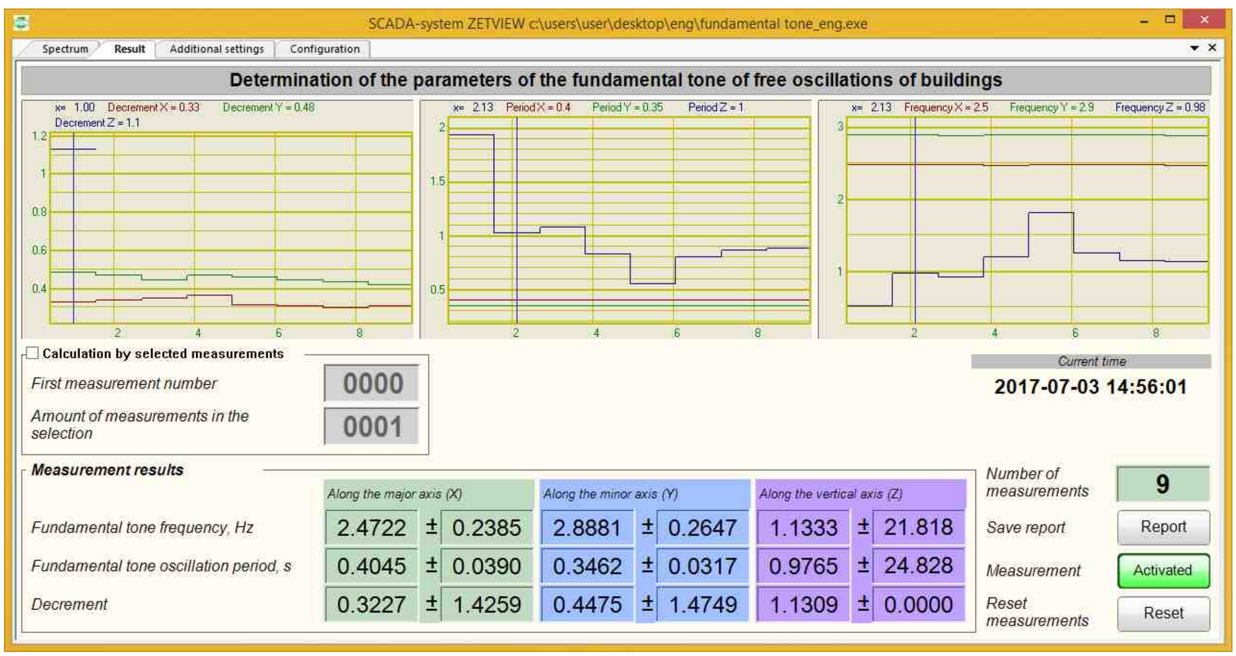


Figure 4.11 «Result» tab

The graphics are depicted for each of the three measuring axes for the period from "Measurement" key activation (Figure 4.12) until the current moment or de-activation of the "Measurement" key.

Figure 4.12 «Measurement» activation key

It is also possible to display selected graphic section – in order to do that it is necessary to enable the function "Calculation by selected measurements" (Figure 4.13) by selecting corresponding checkbox.

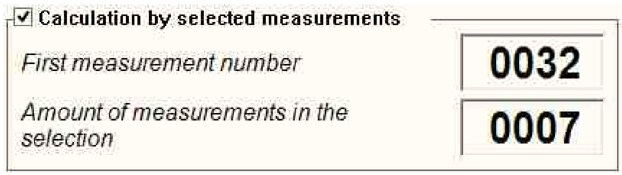


Figure 4.13 «Calculation by selected measurements» option

In order to set the limits of the displayed graphic section one have to enter the required values in the fields «First measurement number» and «Amount of measurements in the selection». After that, there will be displayed graphics with set limits (Figure 4.14 12).

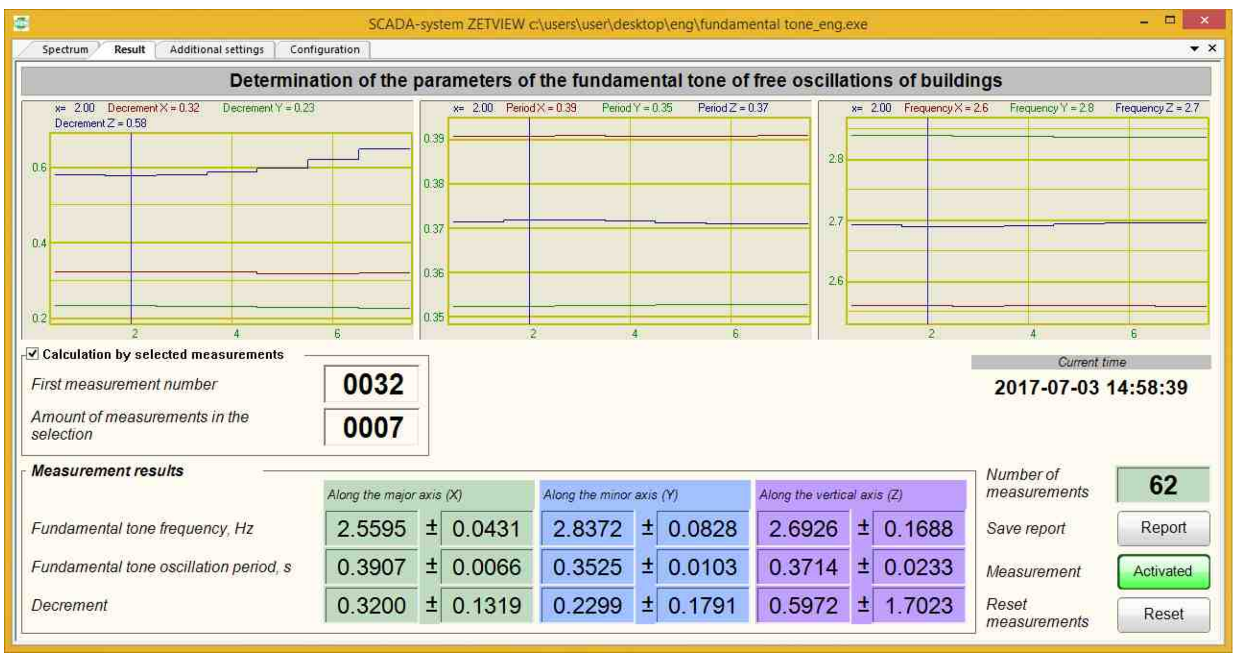


Figure 4.14 Calculation by selected measurements

Digital indicators "Fundamental tone frequency", "Fundamental tone oscillation period" and "Decrement" depict averaged values of the corresponding parameters, calculated for the period from "Measurement" (Figure 4.12) key activation until the current moment or de-activation of the "Measurement" key.

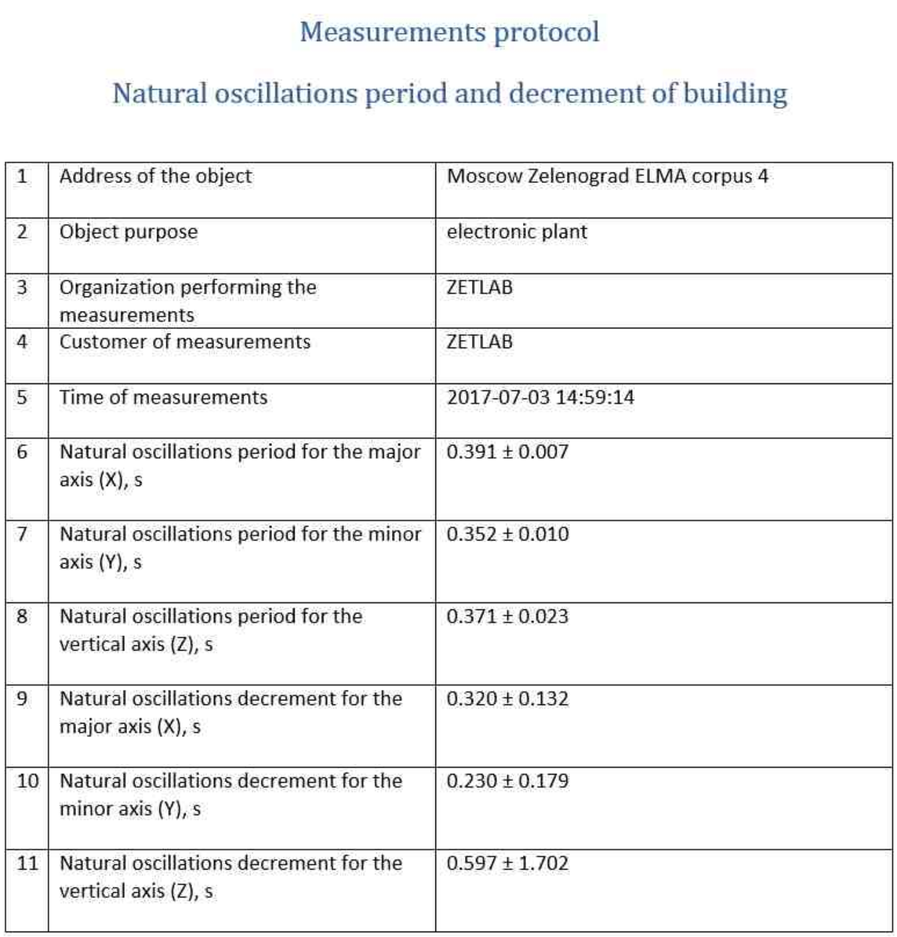
In order to reset the accumulated data, click "Reset measurements" key (Figure 4.15).

Figure 4.15 "Reset measurements" key

The program also has an option of saving measurements results to a report file using "Save report" key (Figure 4.16).

Figure 4.16 "Save report" key

The results are saved in «docx» file. Figure 4.17 shows an example of a saved report file.



Measurements protocol

Natural oscillations period and decrement of building

1	Address of the object	Moscow Zelenograd ELMA corpus 4
2	Object purpose	electronic plant
3	Organization performing the measurements	ZETLAB
4	Customer of measurements	ZETLAB
5	Time of measurements	2017-07-03 14:59:14
6	Natural oscillations period for the major axis (X), s	0.391 ± 0.007
7	Natural oscillations period for the minor axis (Y), s	0.352 ± 0.010
8	Natural oscillations period for the vertical axis (Z), s	0.371 ± 0.023
9	Natural oscillations decrement for the major axis (X), s	0.320 ± 0.132
10	Natural oscillations decrement for the minor axis (Y), s	0.230 ± 0.179
11	Natural oscillations decrement for the vertical axis (Z), s	0.597 ± 1.702

Figure 4.17 Report file

4.2.3 «Additional settings» tab

«Additional settings» tab (Figure 4.18 12) allows to change algorithm of decrement calculation. By adjusting such parameters as "Threshold coefficient relative to amplitude", "Maximum number of points in the calculation" and "Coefficient of culling points after the approximation" one should try to get minimal possible value of "Deviation" parameter in the "Spectrum" tab.

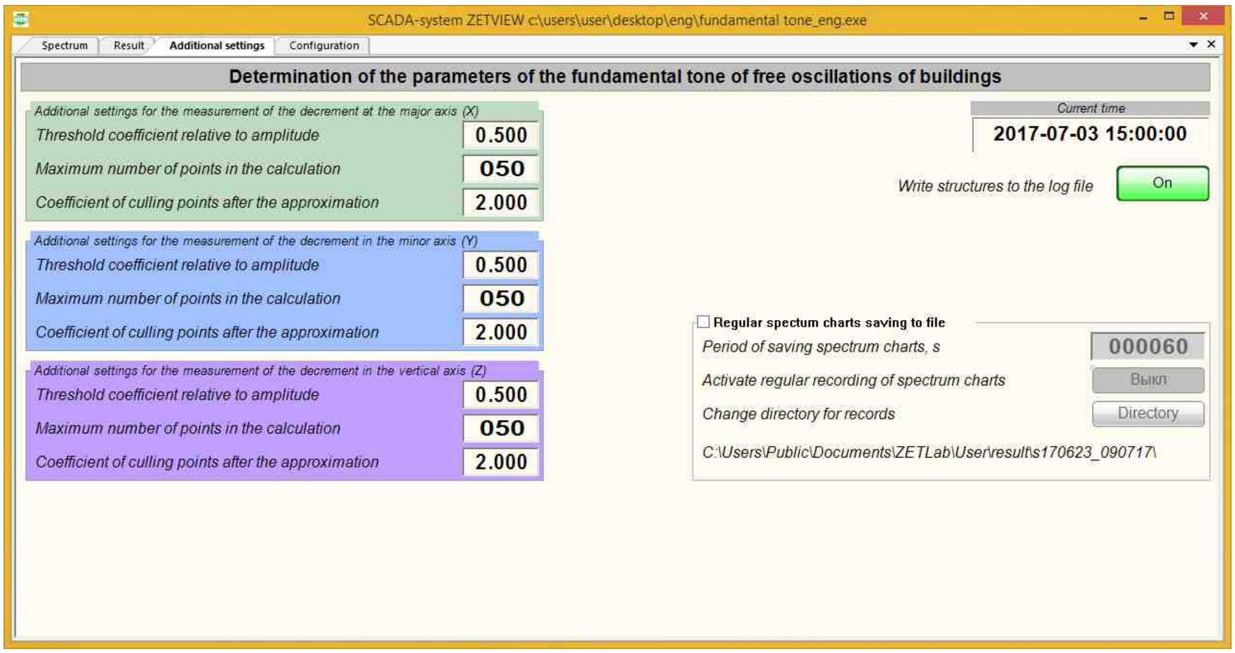


Figure 4.18 Additional settings tab

For periodical saving of spectra graphics it is necessary to enable the function "Regular spectrum graphics saving to a file" (Figure 4.19) by selecting corresponding check-box.

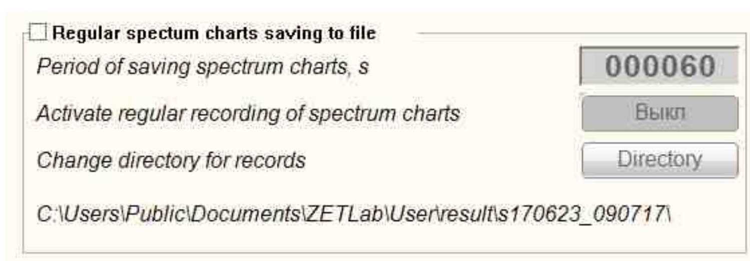


Figure. 4.19 «Regular spectrum graphics saving to a file» function

The parameter «Period of saving spectrum graphics» determines how often the spectrum graphics will be saved. The parameter «Change directory for records» allows to choose records directory. In order to start spectrum graphics recording one should enable the «Activate regular recording of spectrum graphics» option.

User manual contents

The present User Manual consists of 4 sections.
The first Section provides the user with general notion of SCADA ZETView operation. It describes installation, settings and operation of SCADA ZETView.
The second part of the document is to deal with system components. Even though the number of components constantly increases, after having studied the second part of the present document, the user will have notion of basic components of the system. The user will also be able to use the virtual instruments, understand data flow operation logics, and decide which components to use for a particular task. This knowledge will enable the user to come to grips with the new components.

The third section is to deal with complex projects, description of their logics and features, tips on using modules and components for solving any particular tasks. (This clause is currently under development and is not yet included into the present User Manual.)

The fourth section of the present user manual is a guide on components. Description of each component has a simple example of its operation. The examples are shown in *.zvx format, thus providing the user with an opportunity to test operation of turn-key program projects.

Part 1. getting started with SCADA ZETView

Глава 1.System requirements

ZETLAB software is developed to be used on PC-s of IBM PC Intel® Pentium®/Celeron®/ type or any other compatible russified or localized OS versions:

1 Microsoft® Windows® XP with SP3 or later update package (not supported since 11.07.2014)
2 Microsoft® Windows® Vista with SP1 update package (not supported since 11.07.2014)
3 Microsoft® Windows® 7 32 bit with SP1 update package.
4 Microsoft® Windows® 7 64 bit with SP1 update package.
5 Microsoft® Windows® 8 32 bit.
6 Microsoft® Windows® 8 64 bit.
7 Microsoft® Windows® 8.1 32 bit.
8 Microsoft® Windows® 8.1 64 bit
9 Microsoft® Windows® 10 32 bit.
10 Microsoft® Windows® 10 64 bit.
11 Microsoft® Windows® Server 2003.
12 Microsoft® Windows® Server 2008 32 bit
13 Microsoft® Windows® Server 2008 64 bit with SP2 update package.
14 Microsoft® Windows® Server 2008 R2 with SP1 update package.
15 Microsoft® Windows® Server 2012 64 bit
16 Microsoft® Windows® Server 2012 R2 64 bit
17 Microsoft® Windows® Starter (without limitation on simultaneously used programs).

PC configuration for installation and start of ZETLAB software and devices drivers:

- Dual or more core processor;
- Processor speed - over 1,6 GHz;
- HighSpeed USB 2.0* interface;
- RAM – more than 2 Gb;
- Hard disk free space – more than 20 Gb;
- Videocard with 3D-graphical acceleration, support of OpenGL, DirectX, memory - over 128 Mb;
- display resolution 1280Ч1024;
- mouse or any other pointing device (touch screen, track ball, TouchPad, graphic pad);
- standard keyboard or any other input device (sensor screen, graphic pad);
- CD-ROM for software installation.

* ZET devices support HighSpeed USB 2.0 interface only. However, it is possible to connect ZET device to PC via USB 3.0, in the case if controller bus is compatible with USB 2.0 interface (e.g., NEC controllers).

Note: currently there may occur mistakes in the course of Asmedia USB 3.0 controllers use (during driver installation error message "10" is displayed). In this case, it is recommended to use USB 2.0 bus for connection to PC.

In the case if industrial PC-s are used for operation on ZETLAB and ZETVIEW software, we recommend to use 64-bit OS Windows.

When using industrial computers to work on them in ZETLAB and ZETVIEW, we recommend you to use the 64-bit version of Windows.*Equipment ZET interface only supports USB 2.0 HighSpeed. But ZET devices can be connected to a PC via USB 3.0, if the controller of this bus is backward compatible with USB 2.0 interface, such as controllers NEC.
Note: at the moment when working with USB 3.0 controllers Asmedia production problems can occur (if the driver installation, an error occurs with the code "10"). In this case, we recommend to use for PC connection USB 2.0.

When using industrial computers to work on them in ZETLAB and ZETVIEW, we recommend you to use the 64-bit version of Windows

Глава 2.ZETLab: installation

To install ***ZETLab*** software, insert the original CD with ***ZETLab*** software into the CD-ROM drive. The system will automatically recognize the CD and launch the ***ZETLab*** software and drivers installation wizard for the inserted (connected) devices.

If the OS failed to launch automatically the ***ZETLab*** software and drivers installation wizard, then start the ***ZETLab*** software and drivers installation program from the root folder on the CD, ***Setup.exe***.

Attention! If no card device has been inserted into the PCI slot, or no external module device has been connected to the HighSpeed USB 2.0 port, then ***ZETLab*** software will not be installed, and a pop-up message, **Supported device is not found**, will be generated by the installation program.

After the wizard launch, an **Installation ZETLab.msi** or **ZETLab_beta.msi** (Figure 2 .7) window will open suggesting to install ***ZETLab*** software and drivers onto the PC. To continue installation, press **Next >** button and license agreement window will appear, as shown in Figure 2 .8.

After getting familiar with the license agreement, to continue installation, one has to accept this agreement by left-clicking **I accept license agreement terms and conditions** inscription and press **Next >**, otherwise the user will exit the installation program.

In the next window of the installation program (Figure 2 .9), specify the user name and the name of the organization, as well as select the application installation variant: for all PC users or just for one user, and press **Next >** button.

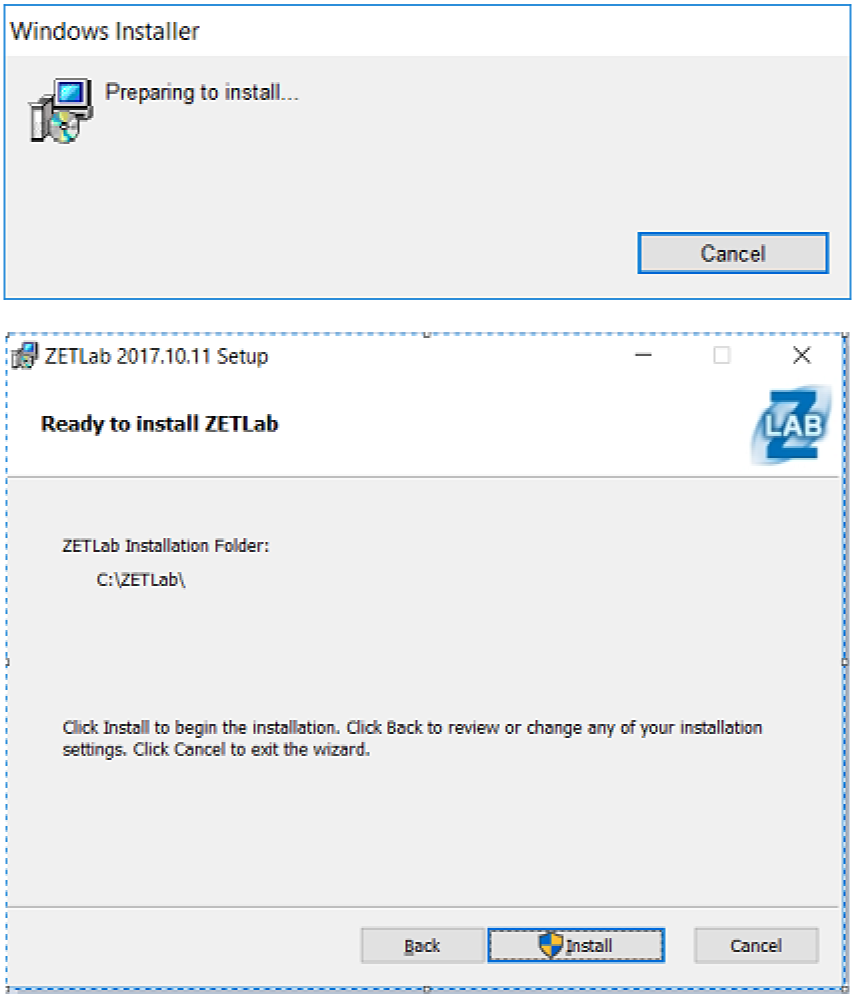

Windows Installer
Preparing to install...
Cancel
ZETLab 2017.10.11 Setup
Ready to install ZETLab
LAB
ZETLab Installation Folder:
C:\ZETLab\
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel

Figure 2.7

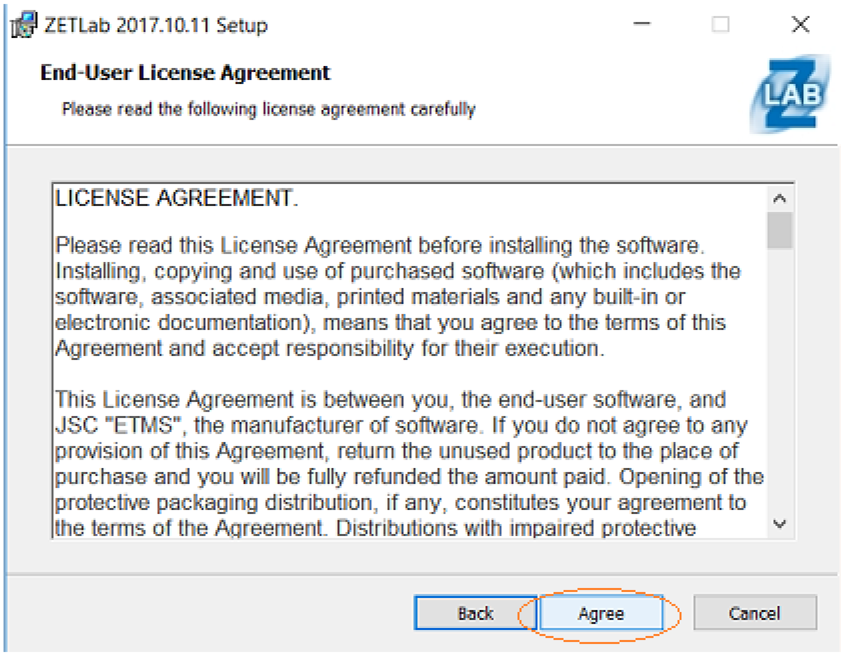


Figure 2.8

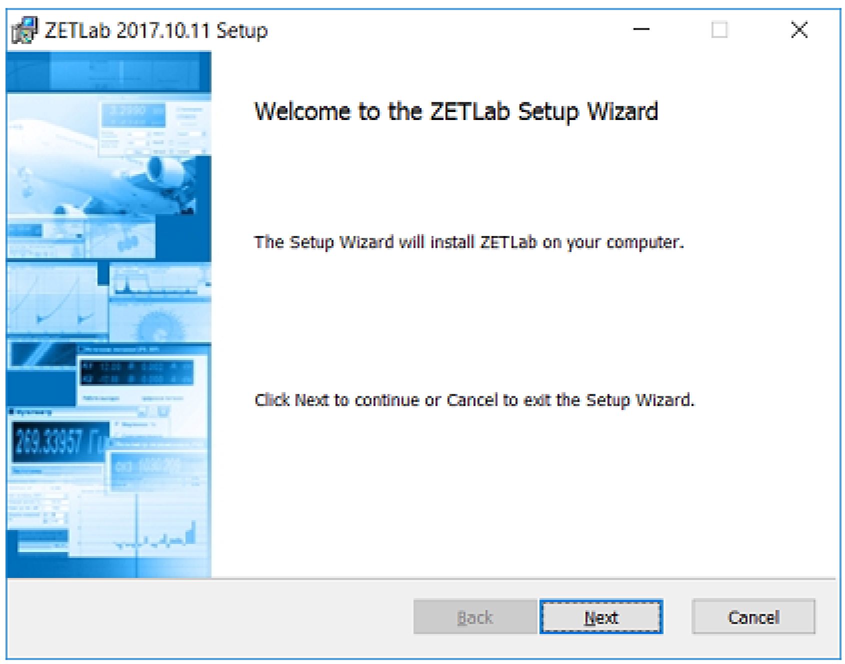


Figure 2.9

Next, you will prompted to choose the installation directory **ZETLAB**. By default, the **ZETLAB** programs are installed in the directory **ZETLAB** (figure 2.10).

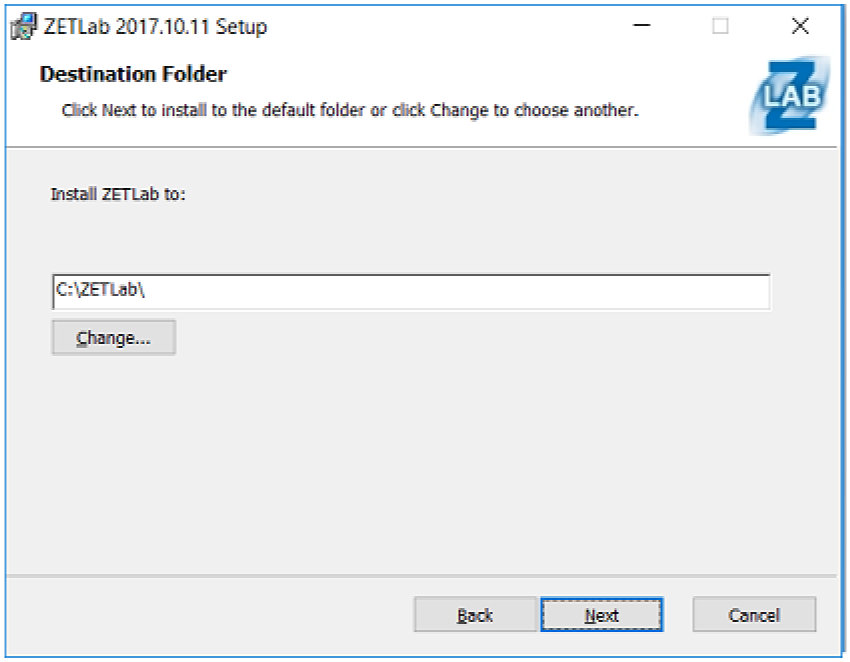


figure 2.10

ZETLAB installation will take several minutes. At the end of the installation will become active the Next button to be pressed to complete the installation (figure 2.11)

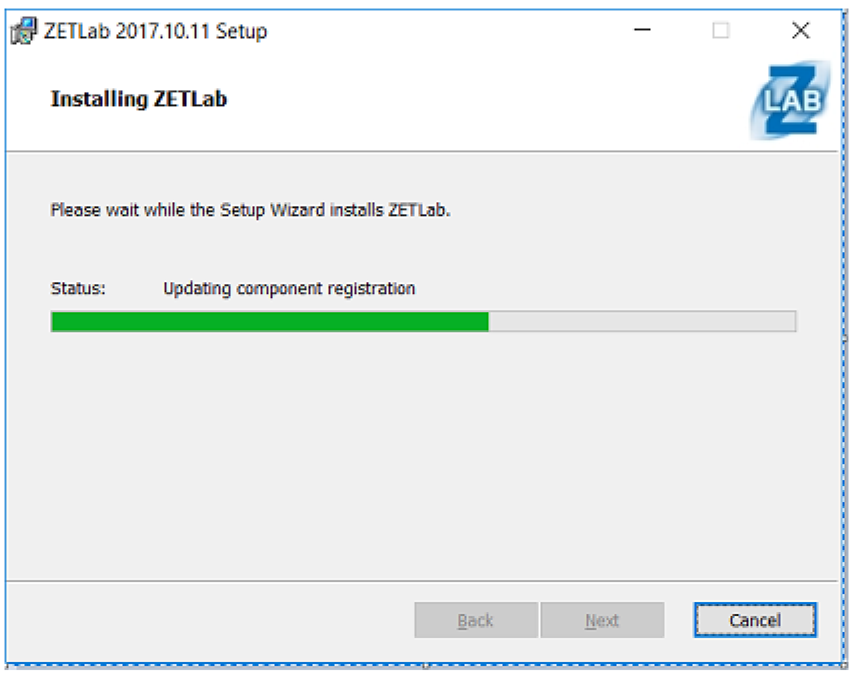


figure 2.11

To exit setup, press "Finish" (figure 2.11).

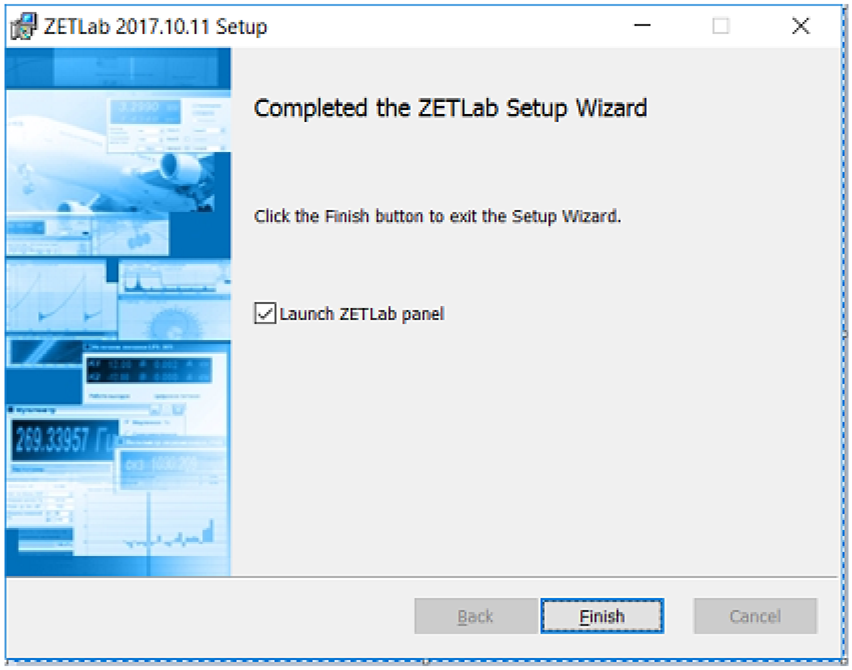


figure 2.11

Cocked Flag to Run the ZETLab panel allows immediately after installation Setup to launch **ZETlab** without starting with the labels program.

Глава 3.Starting ZETView

SCADA-system ZETView is a multi-window Windows application allowing operator to develop automated process control system.

It is possible to start SCADA ZETView using the icon on the desktop or by means of main menu (Main menu->ZETLab->ZETView). It is also possible to start ZETView from ZET-panel (ZETLab): click "Automation", select "SCADA ZETView". If ZET-panel does not appear, make sure that it is not hidden or already started. It is possible to start ZET-panel using the icon on the desktop or via main menu (Main menu->ZETLab->ZETLab).

Top left section of ZETView window contains main program window. Upper section consists of instrument and quick access panel. Central part depicts the project. The left section contains components window and the right one – notification and information window.

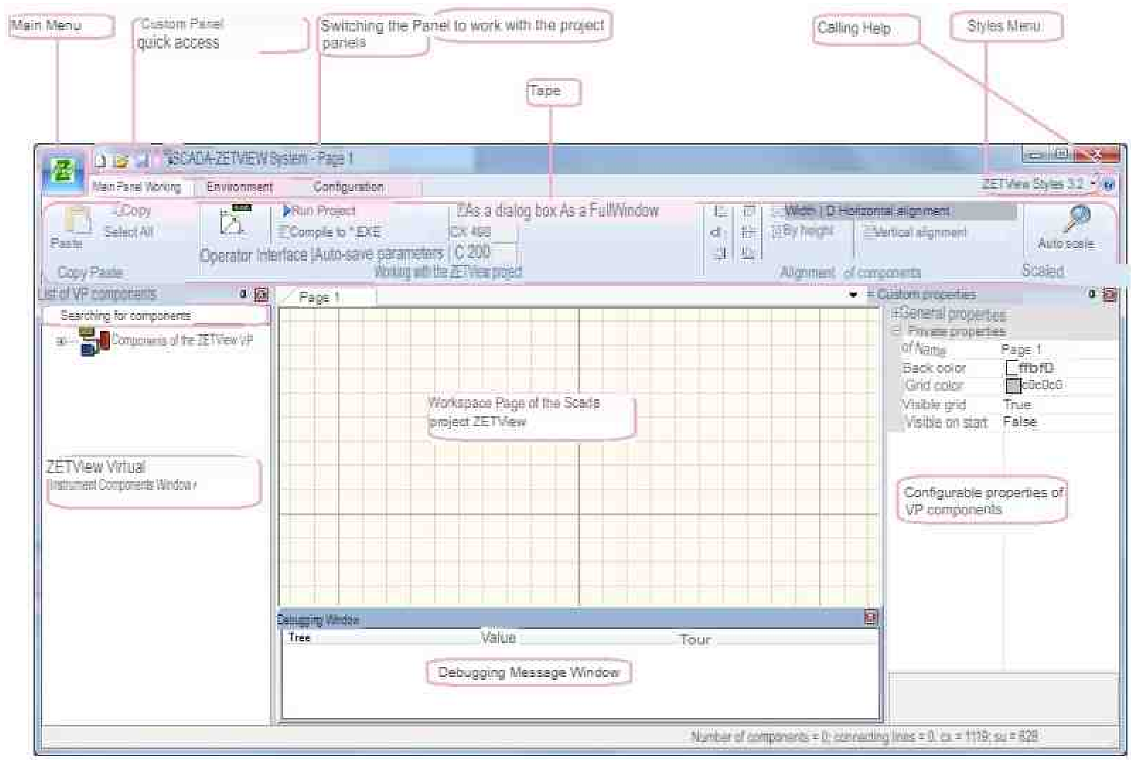


Figure 1.5.1. ZETView window view

Component is a part of the system under development, which performs certain action (e.g. the "Key" element provides certain response as operator activates it – in particular, it allows to enable / disable corresponding virtual instrument).

Main menu

Main menu used for general operation of the project.

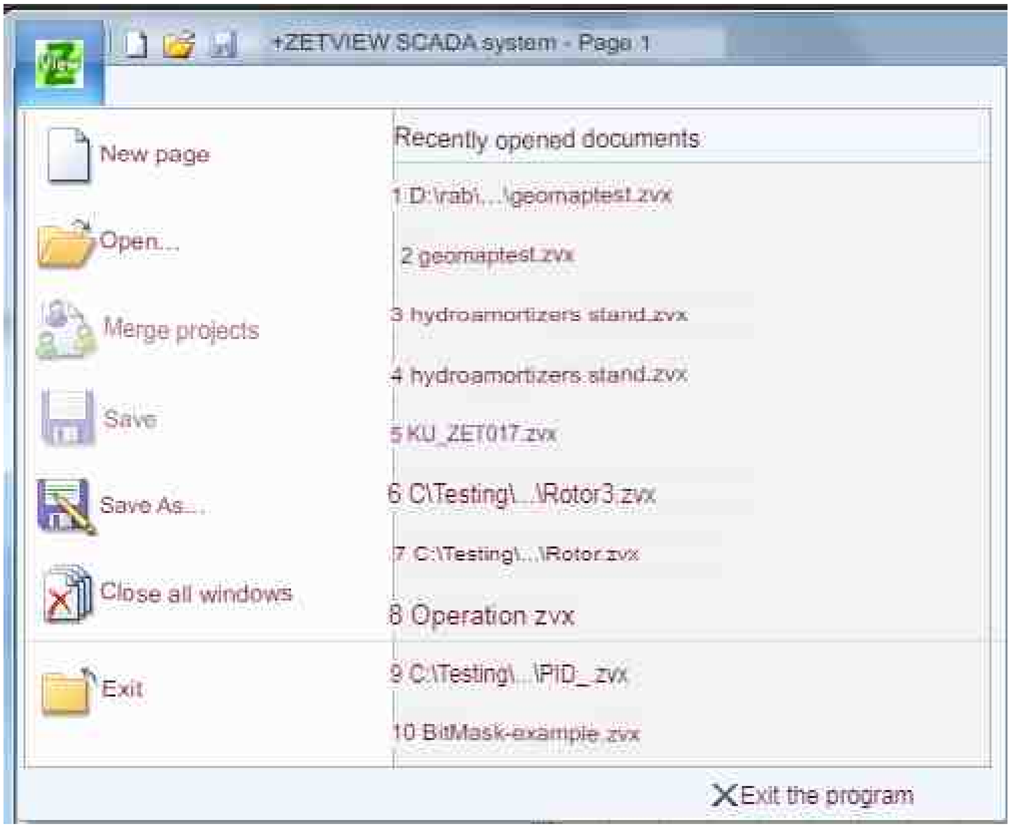


Figure 1.5.2. Main menu

- «New page» - add a page to the project.
- «Open» - selection of already existing project for viewing and editing.
- «Combine projects» - allows to add selected pages from a project to the project under development. Open the project, select this option and choose corresponding project
- «Save» - saving the current project to the same directory with the same project name.
- «Save as» - allows to save a new project or already existing one as a copy with different name and directory.
- «Close all widows» - close all the projects.
- «Exit» - exit ZETView.

User panel

Tape

Display area of project panels:

- Main panel
- Operating environment
- Configuring

The "Main panel" is one of the most often used elements for ZETView operation. It allows to use clipboard, select project operation modes, set equal size of the selected components, change their location on the page, set ZETView program window. Interface settings of ZETView panel are available at "Operating environment" panel. ZET-devices configuration control is performed by means of the commands from "Configuring" panel.

Styles menu

Allows to set ZETView window view.

Help

Select a component in developer or operator mode, click "F1" key – you will see information related to this particular component.

Virtual instruments components window ZETView

Components window contains a list of virtual instruments components ZETView. The components can be classified into corresponding groups (e.g. parameters measurements, logical operations), by alphabethical order, by type, etc. It is possible to add a component into the project from components window as well as from right-click menu of the page.

Working area

In this area, the user can place the components and establish links between them.

Properties window

Properties window is used for viewing and changing components properties. The properties are separated into groups: common (properties that all the components have) and individual (component's individual settings). The parameters controlling linear units and components positioning on project's page are located separately.

Window debug messages

Window of debug messages contains information of components loading to SCADA-system ZETView library, error notifications, exceptions, ZETView operating environment notifications.

Глава 4.ZETView: settings

Among obvious advantages of SCADA-system ZETView one should mention the possibility of customizing the program window. For a new user it is convenient to see all the windows so that to find the necessary components in device window and see the settings configuration of the active component. However, it is possible to activate any of ZETView functions using the context menu or hot keys.

Quick access panel settings

Quick access panel enables access to the most frequently used functions without searching for them or switching between the tabs. The user can set panel keys combination. In order to enter quick access panel settings, click the arrow key to the right from the panel.

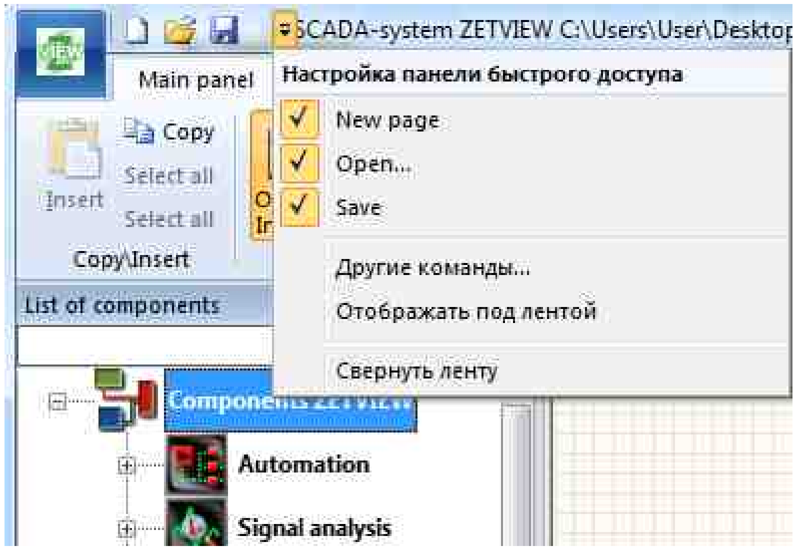


Figure 1.6.1. Quick access panel settings menu

It is possible to add / delete the keys "New page", "Open" and "Save" by checking / unchecking corresponding check boxes in the quick access panel settings menu. The key "Depict under tray" moves the quick access panel under the tray as shown in the figure below.

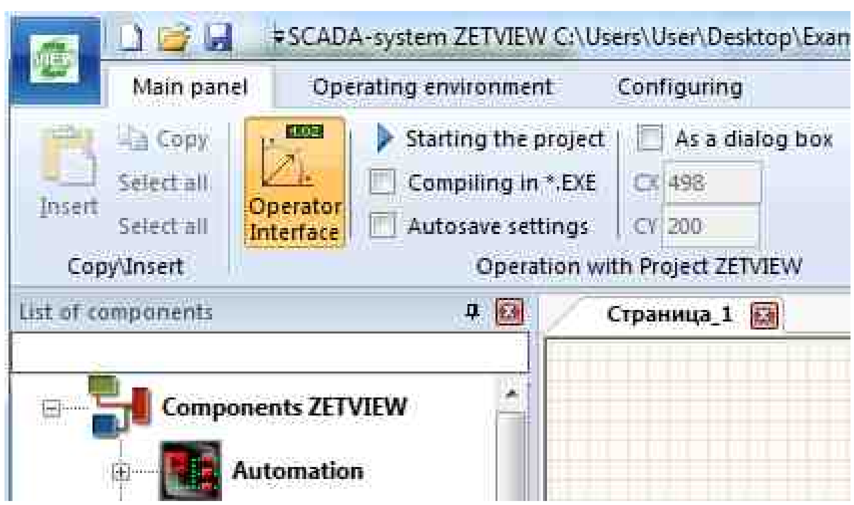


Figure 1.6.2. Quick access menu (under the tray)

The key "Hide tray" allows to hide / open the tray. If the tray is hidden, only project tabs names are displayed, while the panels are available in pop-up mode, i.e. it is possible to open a panel by left-clicking it. The panel will be displayed until any of the commands is selected or a click on free space of ZETView working are is made.

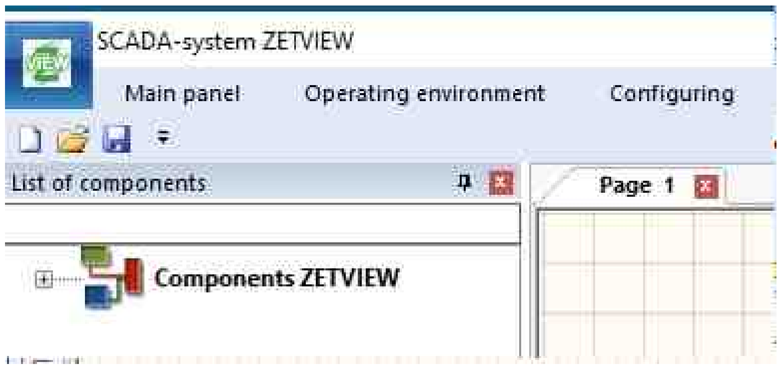


Figure 1.6.3. ZETView window (the tray is closed)

In order to open the tray, uncheck the box "Hide tray" in quick access panel menu. Double-click on "Main panel", "Working environment" or "Configuring" also allows to open / hide the tray.

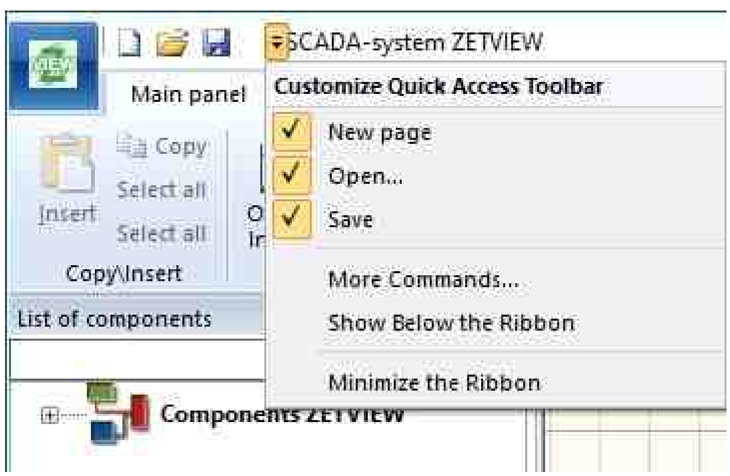


Figure 1.6.4. Quick access panel settings menu

"Other commands" key allows to activate "Settings" window, which is used for adding additional keys to the panel.

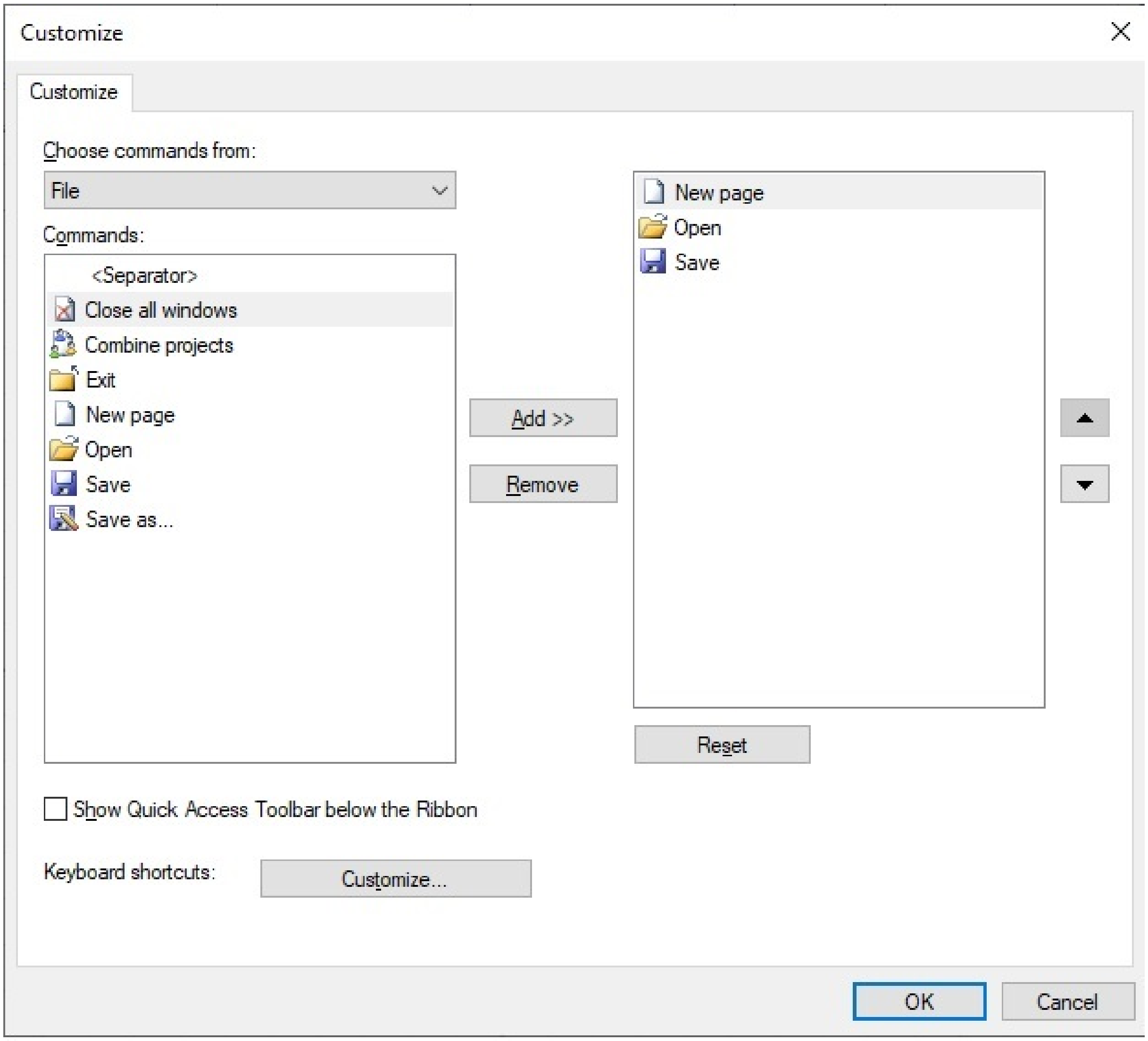


Figure 1.6.5. Quick access panel: settings of additional keys

In the dropdown menu "Select commands from", select menu or panel with the commands you need. Depending on particular choice, there appears list of commands in "Commands" menu. To place a command to the quick access panel, left-click it and select "Add" option – it will appear in the right window. It is also possible to add the command to the panel with a double-click on it in "Commands" section. One can change the sequence of the commands using the arrows or remove the selected commands with "Reset" option.

Hot keys settings

See the figure 1.6.6

Go to quick access menu settings, => other commands. Click «Settings»

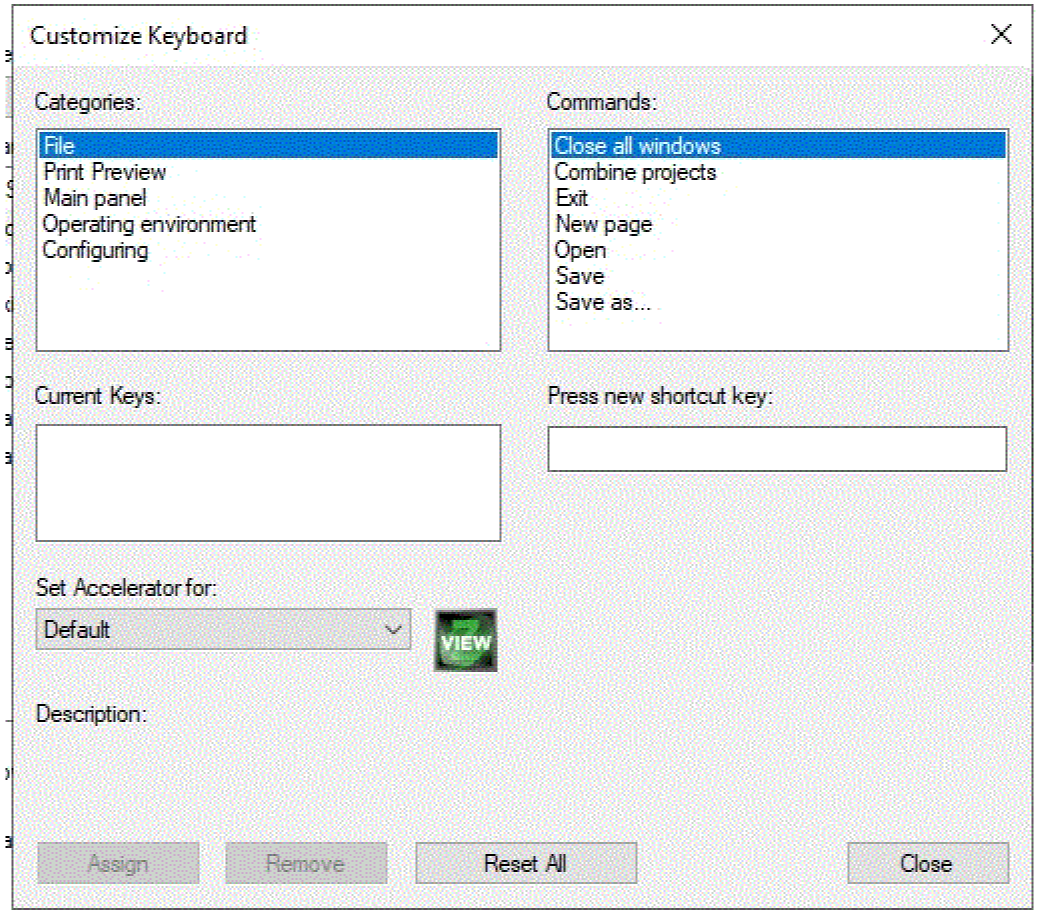


Figure 1.6.6. Hot keys settings menu

In order to set the hot keys combination, it is necessary to select the category and the command. If the key already has a hot key combination, it will be displayed in "Current keys" section. To assign a new key combination, set the mouse pointer in the field "Set new key combination", click "Assign", the new key combination will appear in the "Current keys" list. To delete keys combination, select it in "Current keys" list, click "Delete". The key "Delete all" deletes all user settings of hot keys and sets the hot keys combinations by default.

Settings of ZETView main panel

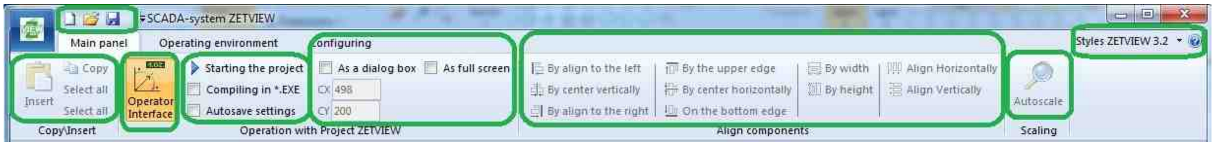


Figure 1.6.7. Main panel: menu and tray

1) In the clause "Add/open/save" the commands «New page» <Ctrl>+<N>, «Open» <Ctrl>+<O> and «Save» <Ctrl>+<S> are used for project operation (figure 1.6.7).

2) The clause "Copy/paste" contains the keys for operations with the components: «Copy» <Ctrl>+<C>, «Copy ll» and «Paste» <Ctrl>+<V> (figure 1.6.7).

3) The clause "Operator interface" is used for viewing and settings adjustment of the components.

4) Operation of ZETView project (figure 1.6.7):

-The key "Start project" allows to start a project to be viewed. It is possible to go back to the working area by means of <F5> key or with a right-click (see the figure 1.6.8)

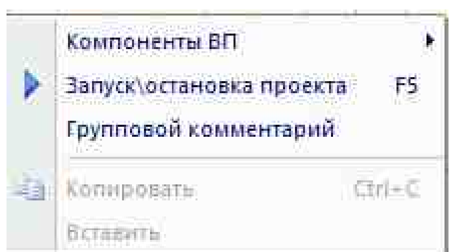


Figure 1.6.8. Project control

-Flag "Compiling in *.exe" creates the *,exe file to be started.

-Flag "Autosave settings" is used for saving / not saving the user settings

5) Flag "As a dialog box" allows to set the units of the source program. Flag "As Full Windows" sets the source program to full screen.

6) "Align components" allows to change position of the components in the working area.

7) "Autoscaling" displays all the components within one page simultaneously.

8) Styles ZETView 3.2 (see the figure 1.6.8a)

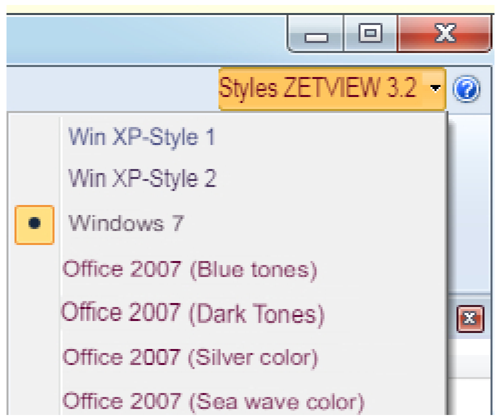


Figure 1.6.8a. Project control

Additional features of ZETView main panel:

a) «Hide list of components» allows to hide / open the components list (figure 1.6.9).

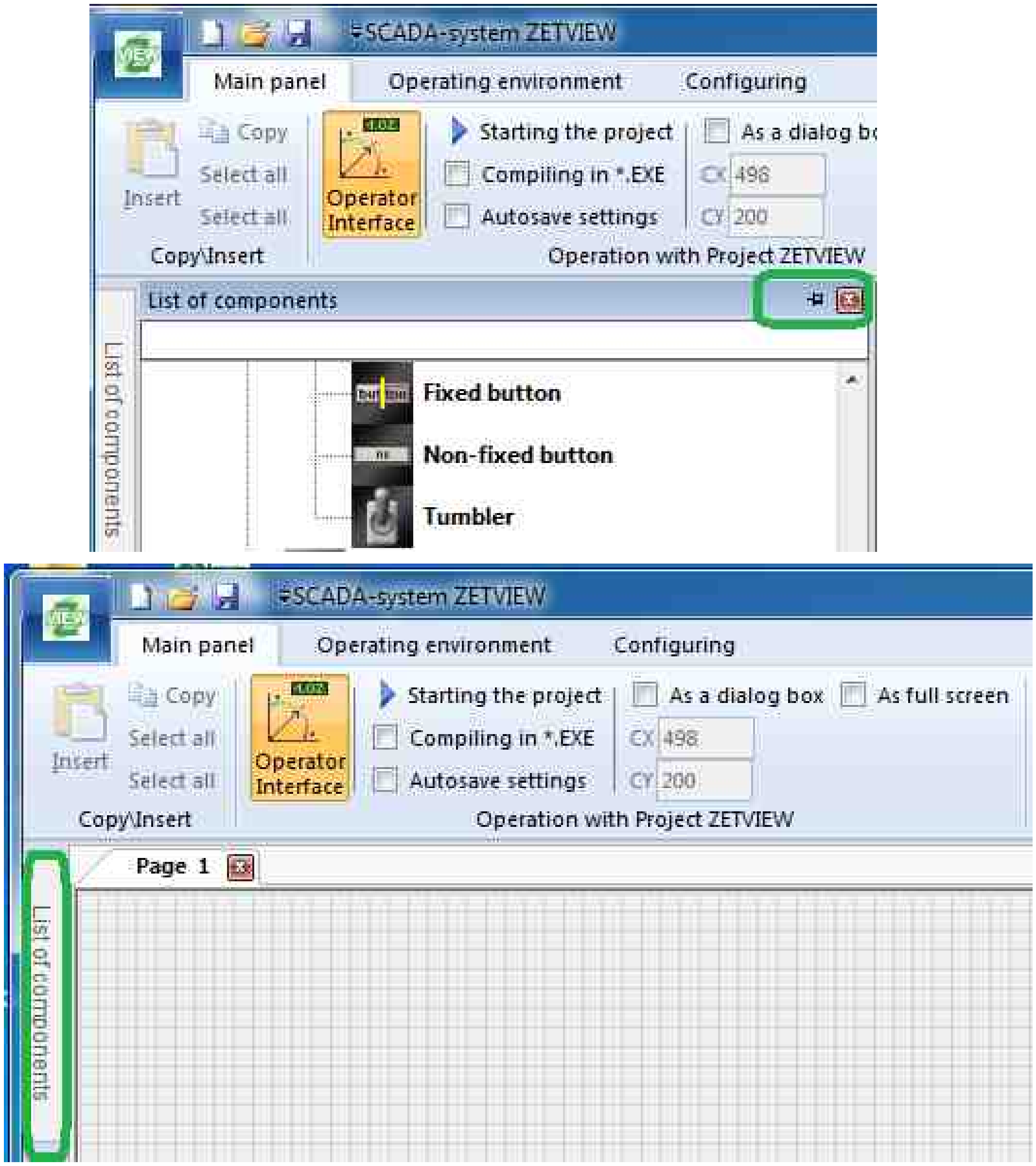


Figure 1.6.9. ZETView window (with hidden and open components list)

b) «Close components list» hides the components list. The components list can be opened in the following way: Operating environment => Components list (figure 1.6.10).

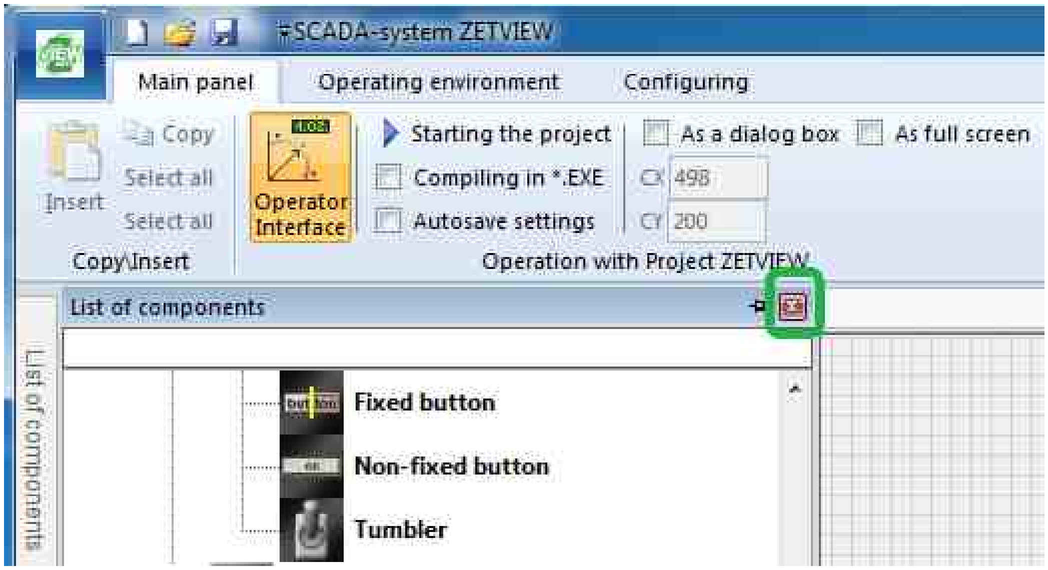


Figure 1.6.10. Components list settings menu

c) Search of components. Enter the name of the component to be found in virtual instruments section of ZETView window (figure 1.6.11).

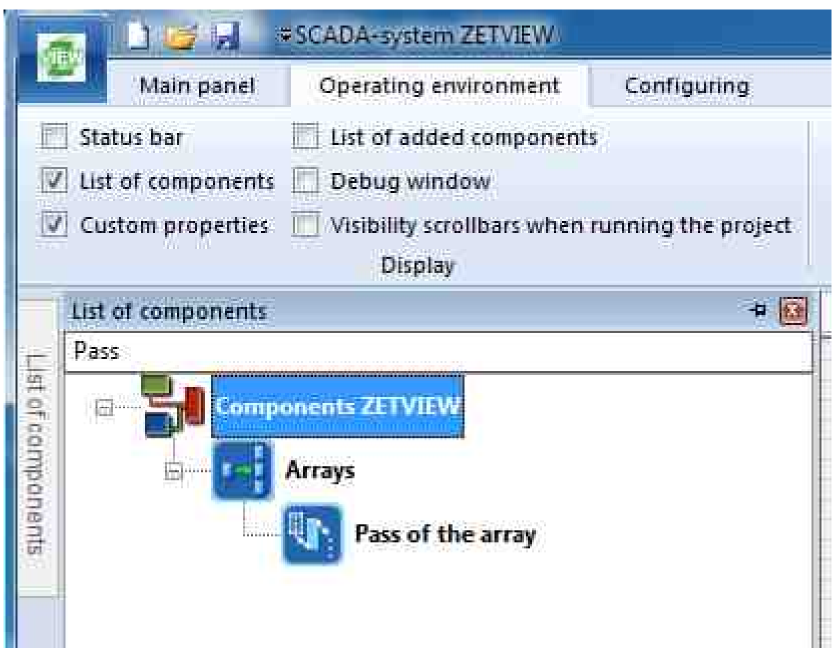


Figure 1.6.11. Search of components

d) Creating comments to the project. Right-click the working area in developer mode – select "Add in debugging" and adjust its properties (figure 1.6.12).

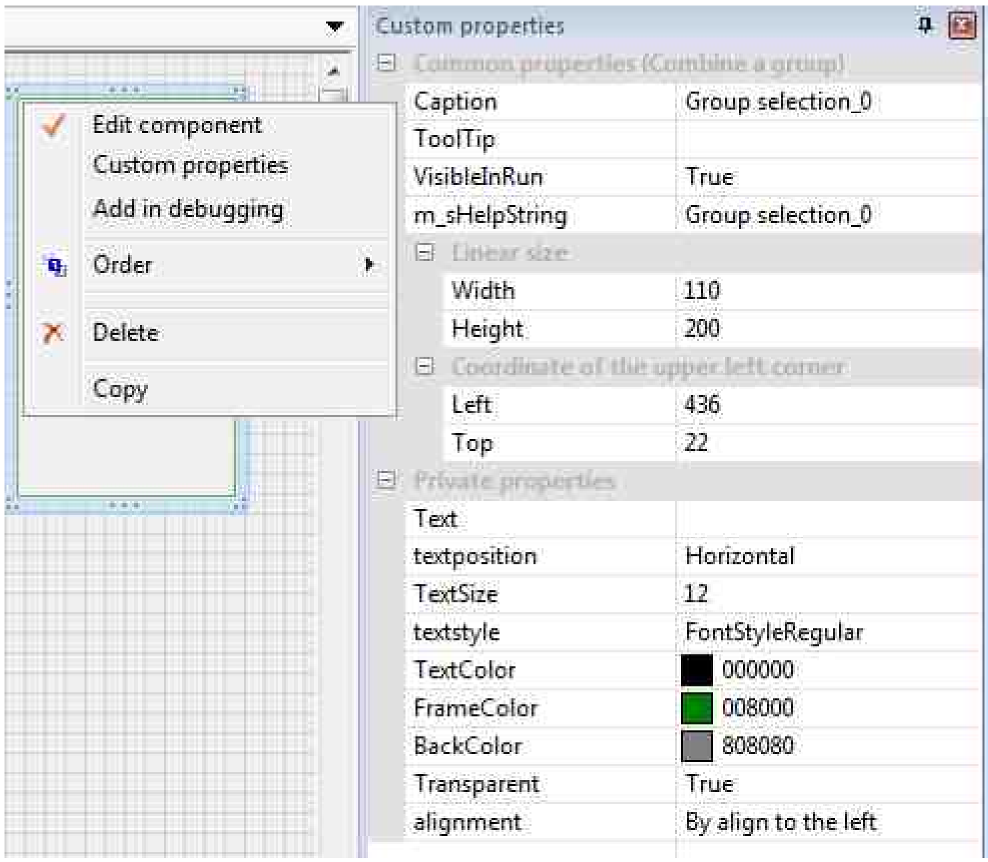


Figure 1.6.12. Creating comments to the project

Operating environment settings

In order to change operating environment settings, enter «Operating environment» tab (figure 1.6.13).

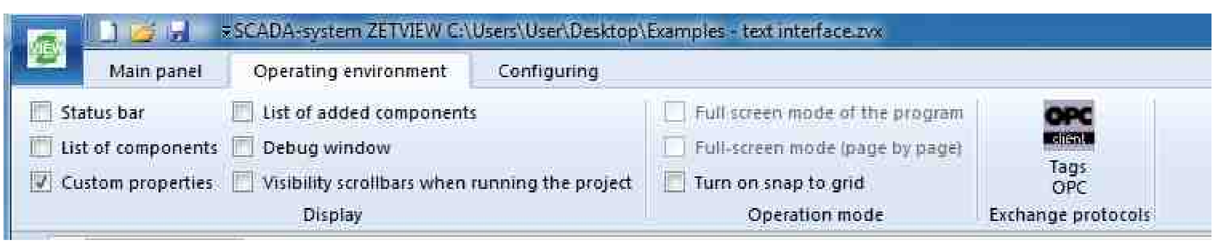


Figure 1.6.13. Operating environment settings panel

In «Display» and "Operation mode" sections you can select corresponding options and windows (also see figure 1.5.1).

In the clause "Exchange protocols" OPC tags are used. OPC is an industrial standard for joint operation of software and hardware components (including those provided by different manufacturers). The standard has been developed for the purpose of establishing of a universal standard architecture and minimizing efforts and costs for establishing hardware and software complex. OPC is often used for automation purposes. ZETLAB Company widely implements OPC for automation, monitoring and measurements purposes. Instruments and software components developed by ZETLAB Company are easy to integrate with OPC infrastructure, thus providing the user with powerful tools for visual representation, spectral analysis, electrical parameters measurements, generation, recording and replay of signals in the sphere of seismics, vibration, temperature and strain measurements.

The standard implements COM and DCOM technologies by Microsoft, thus allowing the applications to perform data exchange on one or several PC-s by means of client/ server architecture. The standard determines the general set of interfaces, so the applications receive data in universal format, even though the data source may be represented by controller, control system, analyzer or another application.

Configuration settings

In order to change configuration settings, enter "Configuring" tab (figure 1.6.14). The flag "Checking the configuration" allows to compare current settings of devices and channels with the previous settings and, thus, to reveal mistakes.

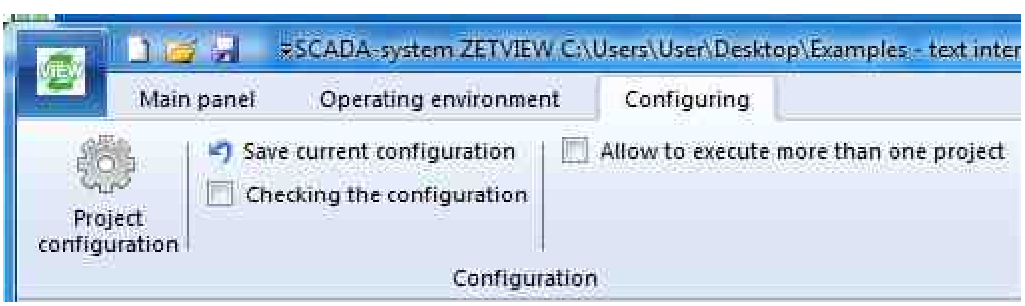


Figure 1.6.14. Configuration settings panel.

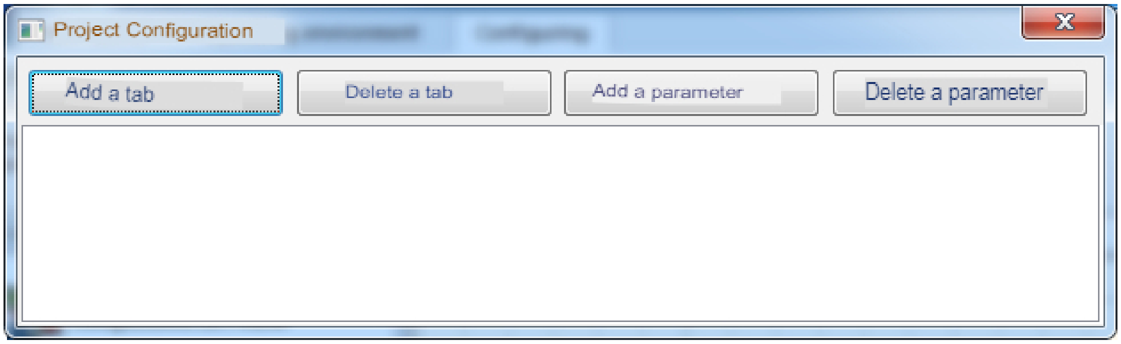


Figure 1.6.15. Project Configurationl.

Глава 5.Operation of components

SCADA system ZETView – is a graphical design environment, in other words – it is an instrument for creation of unique applications. The SCADA-project consists of components.

The components are added to the project from device window, which is also referred to as components library. Depending on their functions, the components are classified into multiple groups: measurements and analysis components, display and indication components, components for selection and control, arithmetic and logical operations control components, components for array processing, programmable components, etc.

Most of the components are used for simple/ complete operations: "Digital indicator" is used to depict digital value, "Frequency meter" is used for signal frequency measurements, etc. Some of the components are used for control of other components, e.g. "Script" component allows to change properties of other components or to activate their function. In addition to that, SCADA-system ZETView has "service" components: "Cross-page link", "Property management", etc. The components allow to perform various tasks and have a wide range of adjustable properties that are described in "Components properties" section.

In order to simplify understanding of project structure, each component has an individual icon describing its function. Besides, for the purpose of establishing links to other components, each component has several contacts of various types, that are different in color and type of the data being transmitted. The clause "Making connections" contains additional information about contacts pads and connections.

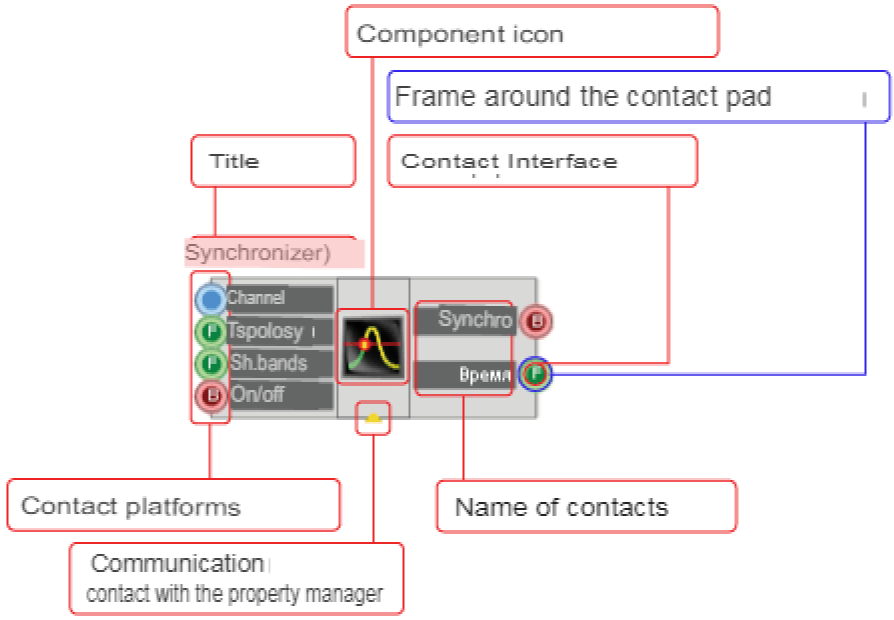


Figure 1.7.1. ZETView component

The components of SCADA ZETView are classified in accordance with their functions. Virtual instruments components window is displayed in the left section of SCADA ZETView program window (in the case if the window is not depicted, see Clause 6 – SCADA ZETView settings):

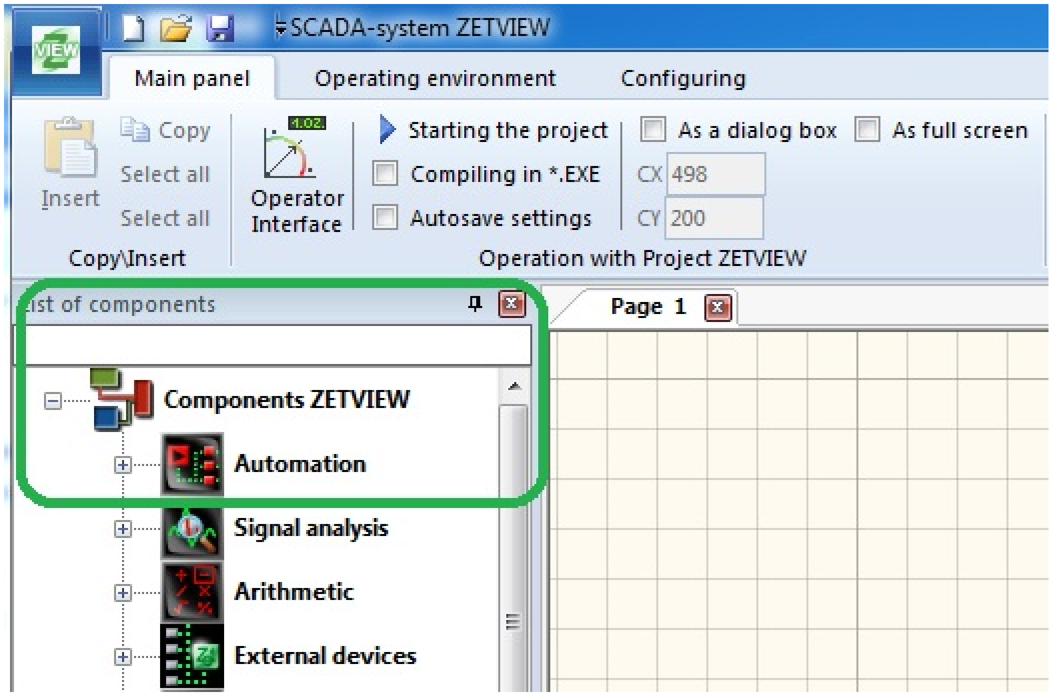


Figure 1.7.2. ZETView components library

Adding a new component

In order to open components list, double click the corresponding section and click "+" to the left of particular item. To add the component to the project, left-click it drag it to the working area.

It is possible to add components to the project from the context menu of the page (right-click to activate it). Selecting "Components of external devices" activates the list of library components. Select particular clause of the library to see corresponding components. Left-click the component – it will be added to project page in the area where the context menu has been activated.

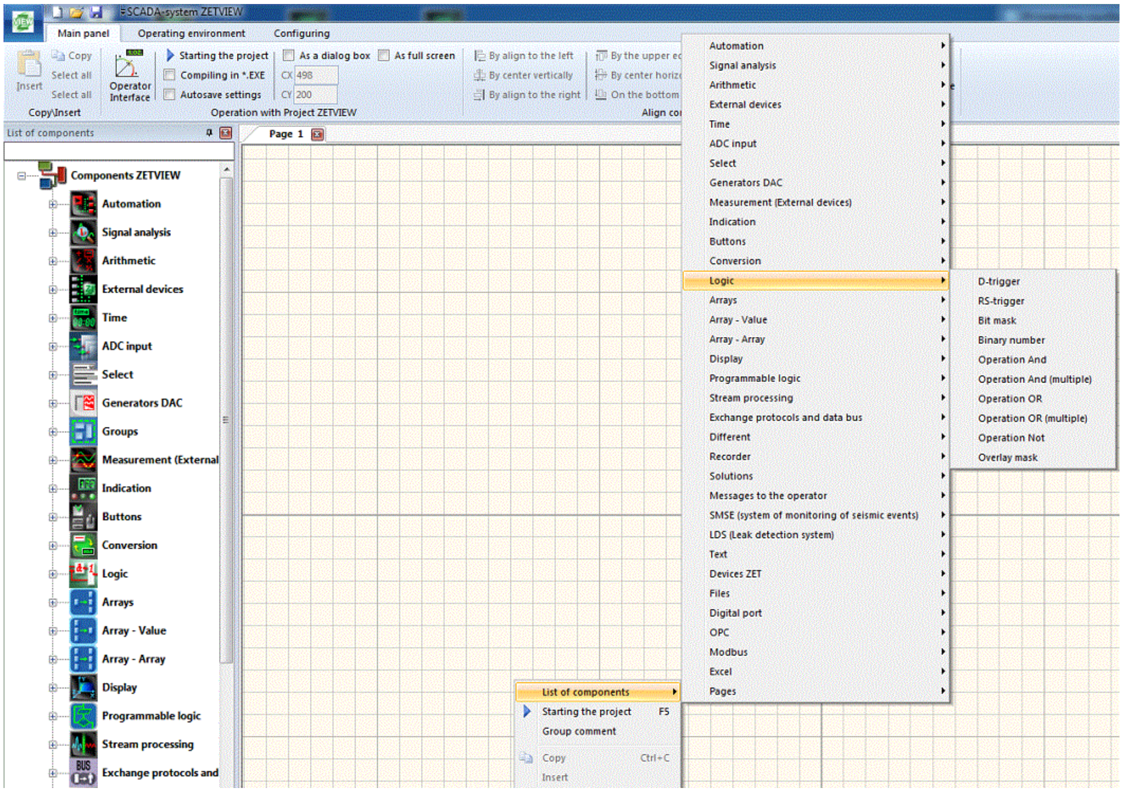


Figure 1.7.3. Selecting components from the context menu

When a component (which can be displayed both in developer and operator interface) is added to one of the interfaces, it will also appear in other interface at the top left section of the page.

How to select and move a component

It is possible to move the components within the working area. To do this, click the component – as it is selected, drag it to the required area. It is also possible to move a group of components by clicking <Shift>+left key, spread the selected area to the components you need. The selected components will be highlighted. While the components are selected, it is possible to move them both with the mouse and arrow keys. To deselect, click any free area of the working section.

How to copy a component

There are several ways to copy a component or a group of components. Select a component or a group of components. Right-click the selected components to activate context menu, select "Copy". Right-click any free space of the working area, select "Paste" in the context menu. The copied component or a group of components will appear above the copied elements with a displacement by one block in horizontal and vertical direction. It is also possible to use «Ctrl+C» and «Ctrl+V» combinations and main panel keys.

How to delete a component

In order to delete a component or a group of components, select them and press Delete key.

Selecting component, changing working area scale

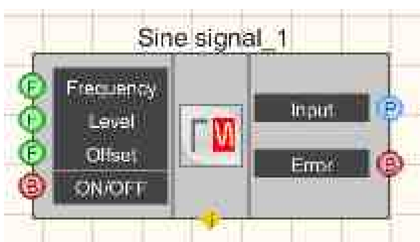
Select the components of the working area, use the scroll key to change working area scale. Large scale allows to read signs of the components contacts.

Hide component

The component on project page can be both hidden and open (by double-clicking it). In the first case it is displayed in compact form, in the second case names of component's contacts are displayed. This makes operation of ZETView system more user-friendly.

Full view of the component

Hidden component



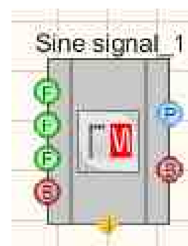


Figure 1.7.4 Full view of the component

Figure 1.7.5. Hidden view of the component

Глава 6.Communication of components

Establishing connections between components

Connections between SCADA components are established by means of contact pads and connections. In ZETView system, the contact pads are classified based on particular type of of the data to be sent / received and have different color and alphabetical marking. Below you can see a table listing various types of contact pads.

Icon	Color	Index	Contact interface	Data type
F	Green	F	Floating point number	Floating point number
B	Red	B	Key	True / False
P	Blue	…	Channel Id	Integral number
...	Green	…	Any type of connection	Any
T	Gray	T	Text line	Text
A	Violet	A	Digit array	Digit array
A	Light-cyan	A	Line array	Line array
E	Light-green	E	Book/sheet/cell Excel	Book/sheet/cell Excel
D	Green	D	Double-accuracy floating point number	Floating point number
S	Violet	S	Structure with set data	Structure with set data

The contacts are also classified into 3 types: input, output and special contacts for communication with properties manager. Input contacts are located to the left, output – to the right.

The contacts for communication with device manager, allowing to increase the amount of component’s contacts, are located at the bottom part of the component (triangle icon).

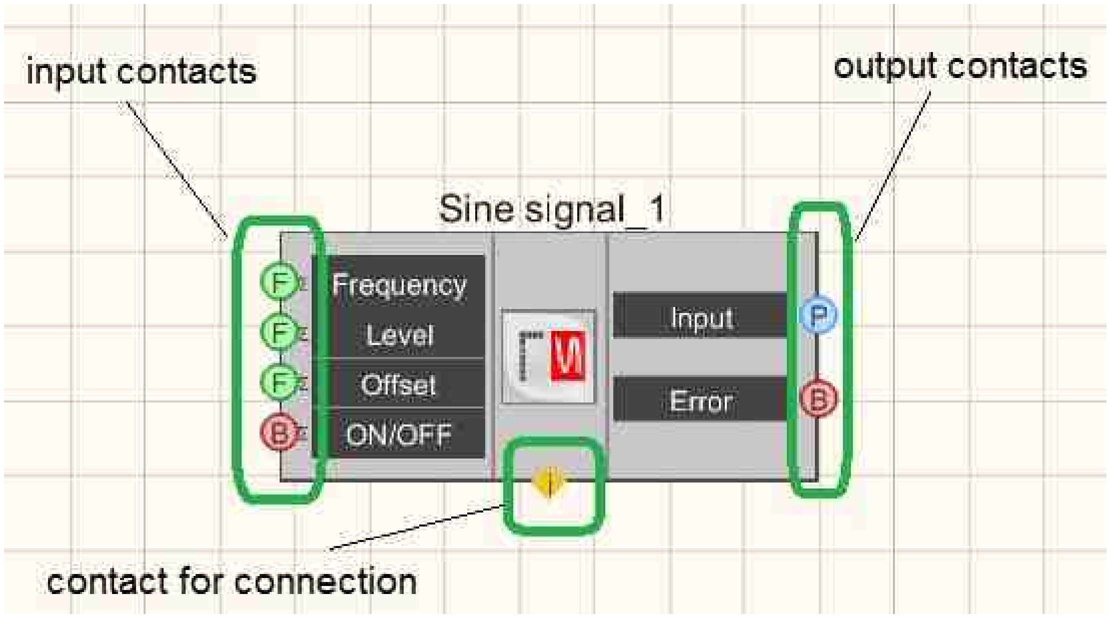


Figure 1.8.1. Contacts functions

It is possible to establish connection between contact pads of two and more contacts. Connection can be established only between contacts of one and the same type (the only exception is the universal contact).

In order to establish connection between two components it is necessary to do the following:

1 Place the required components to the working area (instructions are available in the previous section). For instance, "Button" (Components of external devices ZETView => Buttons => Fixed button) and "Light indicator" (Components of external devices ZETView => Indication => Light indicator)

2 Switch over to developer mode. To do that, make sure that "Operator interface" is disabled.

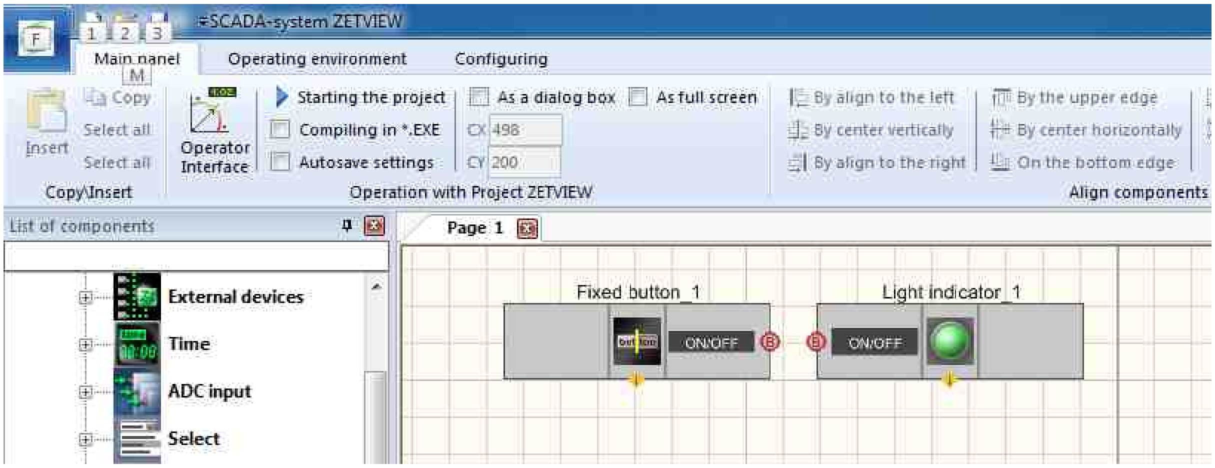


Figure 1.8.2. Developer mode

1 To make a connection, click the button contact (then, as the mouse pointer is moved, there is a line behind it) and click light indicator contact.

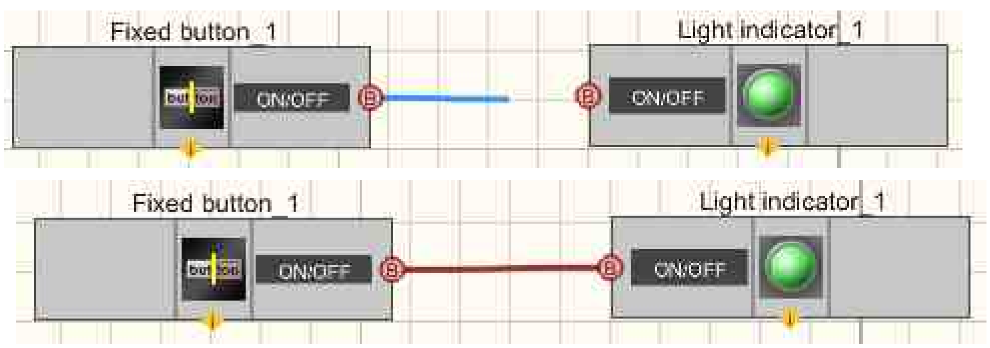


Figure 1.8.3. Making connection

Notes:

- *In order to cancel the process of connection creation, click <Esc>.*
- *The connections can be established from output contact to input contact. It is impossible to establish connection in reverse direction.*
- *In developer mode the component is activated by left-click.*

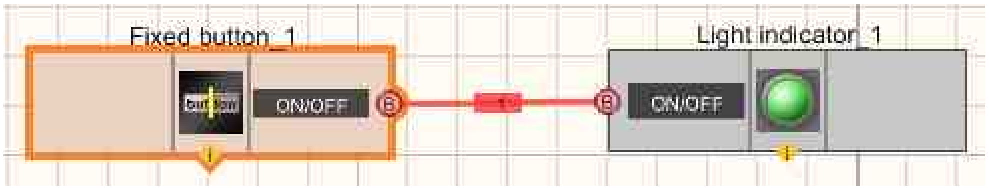


Figure 1.8.4. Active component with connection number

Multiple connections

For most of the components, it is possible to establish multiple connections. In order to create additional connection, follow the instructions below:

Place the third component to the working area. Exit connection creation mode by clicking "Creating connection" key. For example, you can use Sine signal generator (Components of external devices ZETView => Generators (DAC) => Sine signal).

Left-click the button contact (connection line will "follow" mouse pointer). To complete connection creation, click the contact of the component, to which the connection should be established.

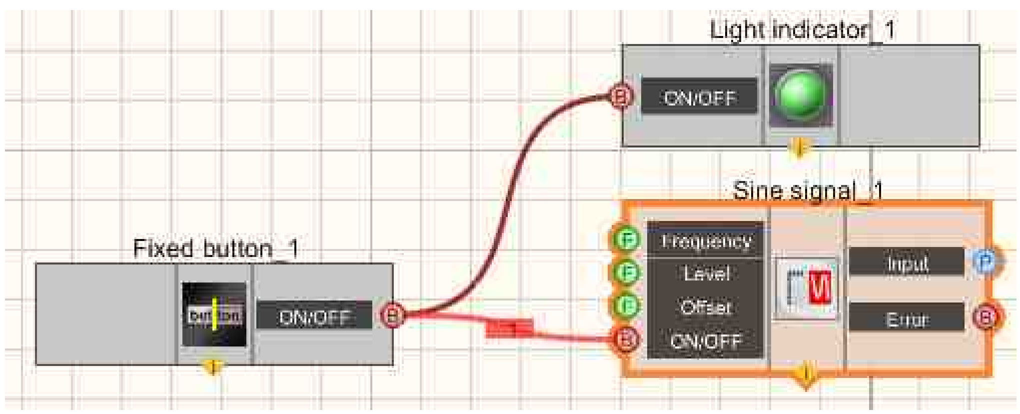


Figure 1.8.5. Creation of multiple connection

To make an additional connection for a contact, select option "Create connection" (connection line will "follow" mouse pointer). To complete connection creation, click the contact of the component, to which the connection should be established.

Note:

- *It is possible to create connections only between contacts of the same type; the only exception is "Any type of connection" contact. In the case if it is impossible to create a connection between the contacts, there will appear a red frame around the contact*
- *Now it is possible to delete the connection lines. Double-click the connection line – there will appear rhombs at the contact points – double-click them to reset the connection.*

Removing connections

To remove a connection, left-click it and press "Delete" key.

Creating connections: sequence

ZETView has a feature of analyzing connections sequence. If a key has several connections, then, upon its activation, the signal will be forwarded to the first contact connected. In the course of project development, it is necessary to bear in mind the sequence of data transfer between the components connected. In the case of wrong sequence of establishing conections there may appear mistakes, which will be hard to find.

Глава 7.Operator and developer interface

ZETView project is represented by one or several ZETView pages with two or more program components interconnected with logical links.

Project development includes several stages:

- Adding components to the project
- Establishing connections between them
- Setting components parameters
- Setting of project display parameters

ZETView project pages have two interfaces: for operator and for developer. The main difference is that operator mode allows to set project display parameters while developer interface depicts communication lines between the components.

To select interface and operation mode, use the main instrument panel at the top section of ZETView window.

Figure 1.9.1. "Operator interface" key in inactive mode

To enter operator interface, click "Operator interface" icon on main instruments panel. It will change the icon view.

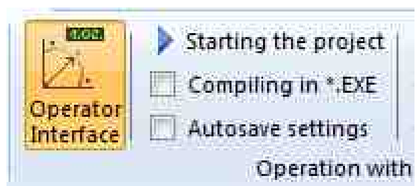


Figure 1.9.2. "Operator interface" key in active mode

Note:

- *SCADA-system ZETView has a function of adding commands from instrument panel to the quick access panel. As the commands from "Operator interface" are added to the quick access panel, it is possible to switch between operation modes and display interfaces without using ZETView main panel. The appearance of the icons on the quick access panel depends on the selected interface and mode.*

The figures below show the project view in developer and operator interface.

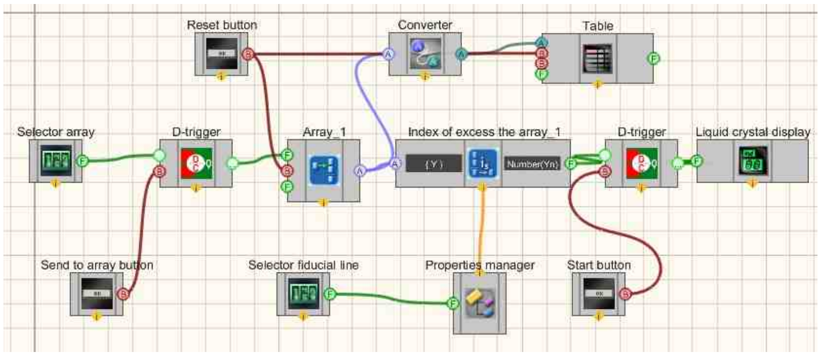


Figure 1.9.3. Appearance of the project in the design interface.

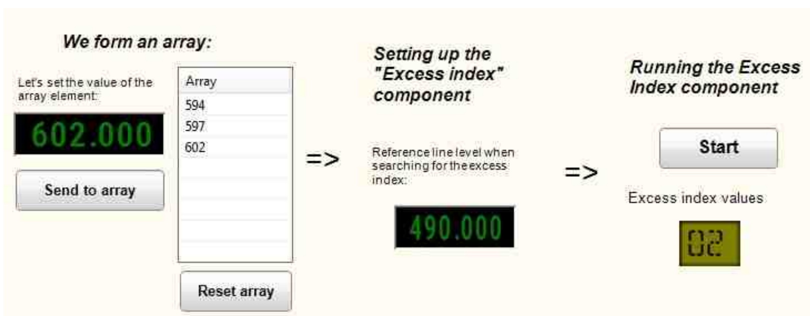


Figure 1.9.4. Project view in operator mode

View the example in ZetView

A lot of components are not displayed in operator mode. Among the components that are depicted in operator mode, there are components controlled by operator; selection, indication and display components. In developer mode, it is possible to arrange the components, set the properties, establish connections, while in operator mode one can set components display, their position, add comments.

Note:

- *For the components that are depicted in both interfaces, it is possible to disable their display in one of the modes. In general properties of developer mode VisibleInRun is implemented (it is used to enable / disable component visibility at the start of the program).*

It is possible to switch between the modes only at the stage of project development or editing. As the project is started, it is displayed in accordance with operator's mode settings. The user has access only to those components, which are necessary for measurements control and data acquisition, while all the other components are hidden. Thus, it allows both to simplify the project operation and introducing amendments into current project.

Глава 8.Components properties

In order to view component's properties, select in in the working area. Corresponding properties will be displayed in custom properties window. If the custom properties window is not displayed, enable it (see Clause 6 – ZETView settings, "Working environment settings"). It is also possible to view component properties by right-clicking it and selecting the "Custom properties" in the context menu. Then the user will see properties window of the selected component.

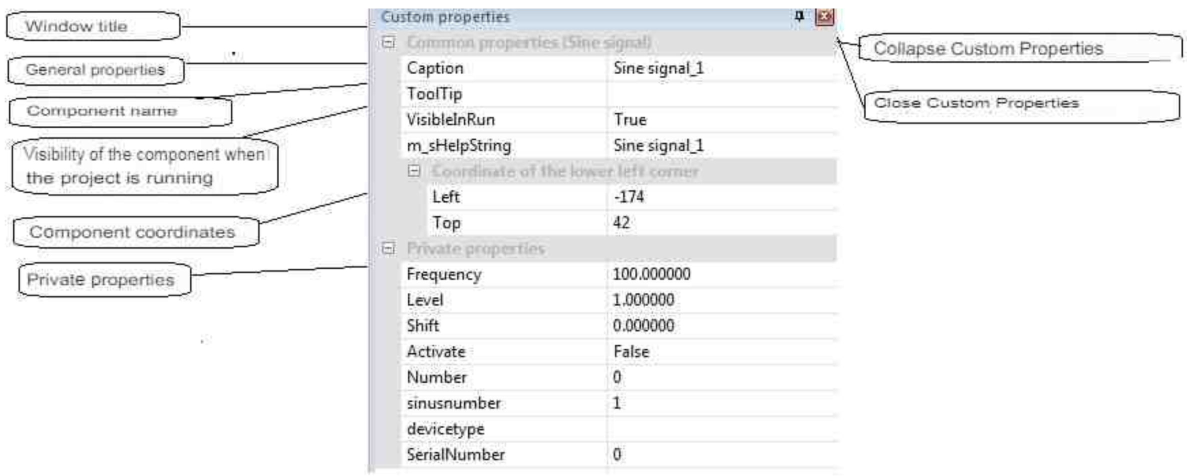


Figure 1.10.1. Custom properties window

Components properties are classified into general and individual. All the components have general properties.

Below the "Custom properties" window there is a menu of the set object. The clause "Components of external devices" is selected when setting parameters of the current component. "Pages" menu is used to set display parameters of the current page.

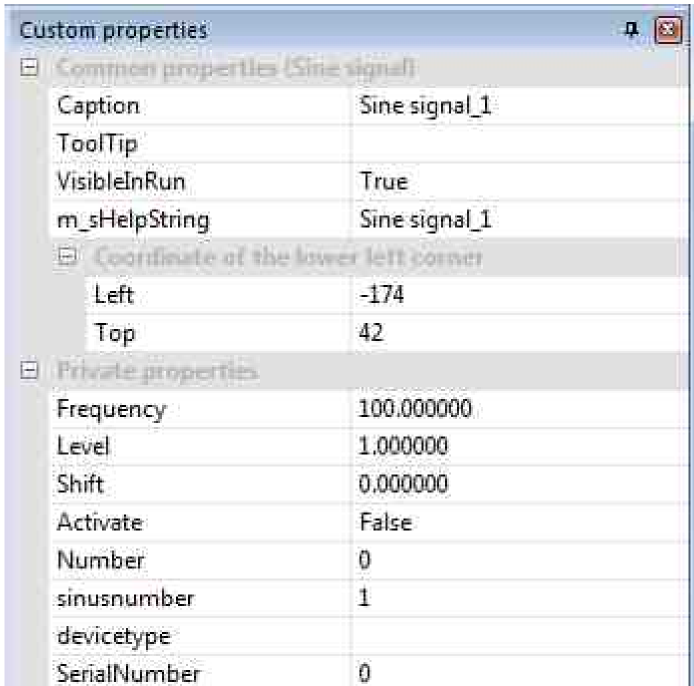


Figure 1.10.2. Page properties

Bottom part of "Custom properties" allows to click any of the properties to view information of its functions.

General properties of the components

- Total properties (environment):

Caption - each component of SCADA-system ZETView has a name. As the component is added to the project, it has a name by default, for instance, "DC Voltmeter". The developer can change the component name at his discretion, e.g. for "Generator voltage". It is possible to change the name of the component in "Custom properties window" in "Caption" line. As scenario is using corresponding component, its name is depicted (so, it is reasonable to assign individual caption for each of the components). Components names are depicted above them only in developer mode, while in operator mode it is possible to see the component name in properties window.

ToolTip - tooltip for the user.

VisibleInRun– component visibility in operator mode. Allows to enable/ disable component visibility in operator mode when the project is being run.

m_sHelpingString – identifier string. All the components have it by default (it is filled automatically). The identifier string is used in the case when the properties of a component are to be used for another component. For instance, converter (numeric array – line array) + Data table. This feature is also often used for scripts creation (programmable component) – identifier string.

m_sHelpingString - string id. By default, it is present for all components, it is filled in automatically. Used when the properties of one component are inherited by others. As in Example converter(num.array-str.array) + data table. Also widely used for scripting (component of "Programmable component" 940)

- Linear units:

Width - component width.

Height - component height.

- Left angle coordinates:

Left - X coordinates of left angle.

Top - Y coordinates of left angle.

Linear units and left angle coordinates are set individually for each display interface

Setting properties

Components properties are set in custom properties window. In order to change any of the properties, set the mouse pointer in the field of necessary property, delete the current value and enter a new one, or choose the required property from the list.

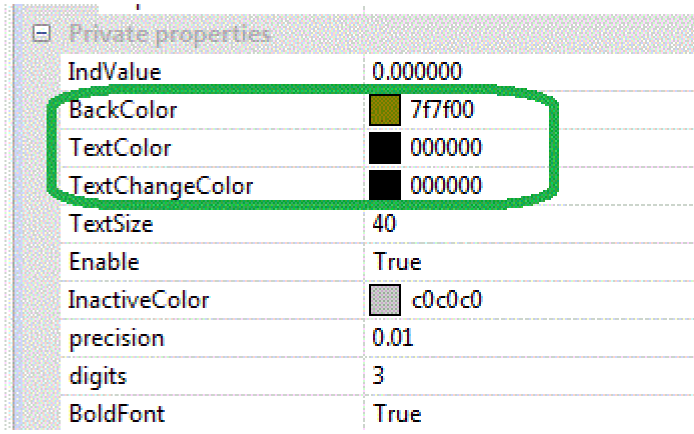


Figure 1.10.3. Changing properties

Components properties are not just parameters necessary for proper operation of the components. Components properties provide for flexibility of SCADA-system ZETView. Various settings combinations enable different functions of one and the same component. It is possible to set components properties in settings window in the course of SCADA-system development, or by using control components connected to corresponding contacts of the component (selector, properties manager).

Глава 9.Display settings, alignment and positioning

In ZETView system, the developer can set the display mode of the project, so that the operator would see only the components necessary for project operation and corresponding results representation.

Positioning of the components

"Custom properties" window in SCADA-system ZETView contains clauses "Linear units" and "Upper left angle coordinates", in which corresponding parameters of the component are displayed.

Note:

➢*Linear unit and upper left angle coordinates belong to display parameters, hence, they are to be set in developer mode.*

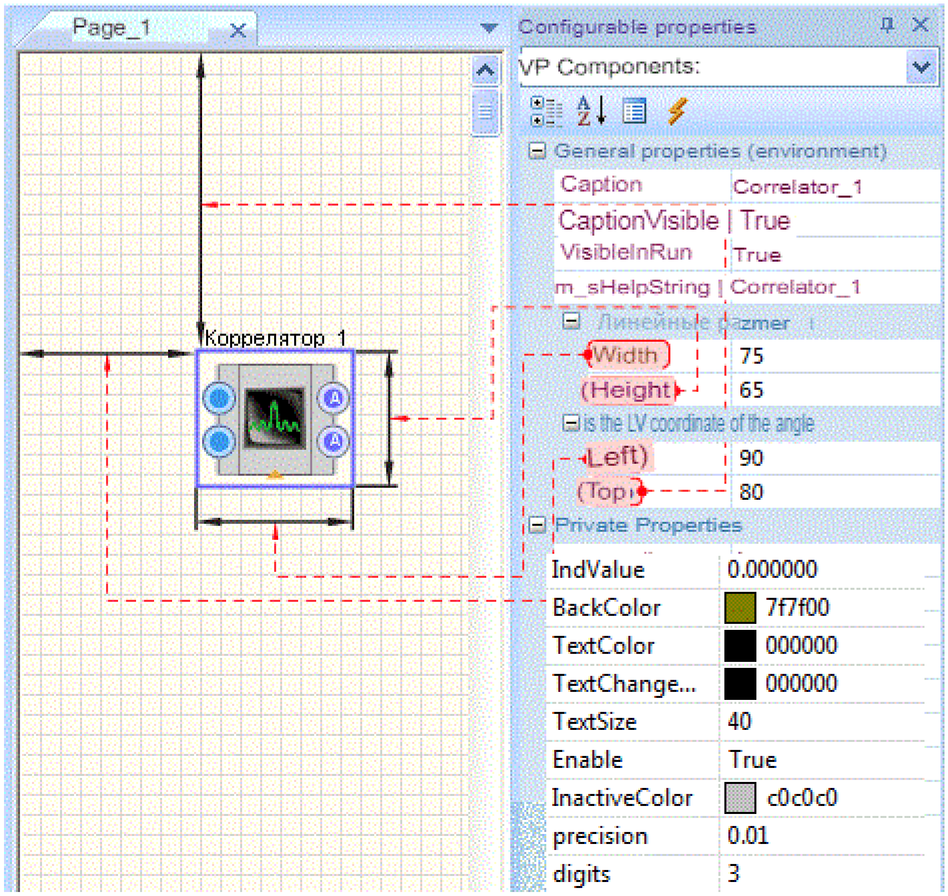


Figure 1.11.1. Linear units

Linear units (width and height) are set for each of SCADA-System component. Measuring units – px. Component width is set in "Width" line, height – in height line respectively.

Positioning of the component on ZETView page is set by coordinates of its left upper angle. X coordinate is set in "Left" line, Y coordinate – in "Top" line. Page coordinates are set in px. Point with 0.0 coordinates is the top left angle of the page.

In order to set identical units for several components and to align the components with each other, use corresponding commands from "Align components" section of ZETView main panel:

- *Align left – aligns all the selected components to the left (sets identical Left property value for all the components),*
- *Align top – aligns all the selected components by top (sets identical Top property value for all the components),*
- *Align by width- aligns all the selected components by width (sets identical Width property value for all the components),*
- *Align by height - aligns all the selected components by height (sets identical Height property value for all the components).*

As several components are selected, one of them will have green frame, while the other – blue frames. The alignment will be performed in ratio to the component with green frame.

It is also possible to use hot-keys combinations instead of "Custom properties" menu (see the table below). The hot-keys allow to change units and positioning of the selected components. As the component units are changed, coordinates of its top left angle remain unchanged.

<right arrow key> shifts the component to the right by 5 units	<Shift>+< right arrow key > increases component width by 5 units
< left arrow key > shifts the component to the left by 5 units	<Shift>+< left arrow key > decreases component width by 5 units
<up arrow> shifts the component up by 5 units	<Shift>+< up arrow > decreases component height by 5 units
<down arrow> shifts the component down by 5 units	<Shift>+< down arrow > increases component height by 5 units
<Ctrl>+< right arrow key > shifts the component to the right by 1 unit	<Ctrl>+< right arrow key > shifts the component down by 1 unit
<Ctrl>+<left arrow key > shifts the component to the left by 1 unit	<Ctrl>+<up arrow> shifts the component up by 1 unit

1 unit = 1 px. Page square size is 10 units.

Components order

Components order in the case of their overlapping is controlled by context menu available for each of the components. The context menu is activated by right-clicking the active component. Select "Order" in context menu, then – "In the foreground" or "In the background" to place the component to the foreground or background of the page.

Changing components view in operator mode

Project design can be changed depending on particular requirements. Each component has properties allowing to change its display mode. For instance, liquid crystal display may have the following design:

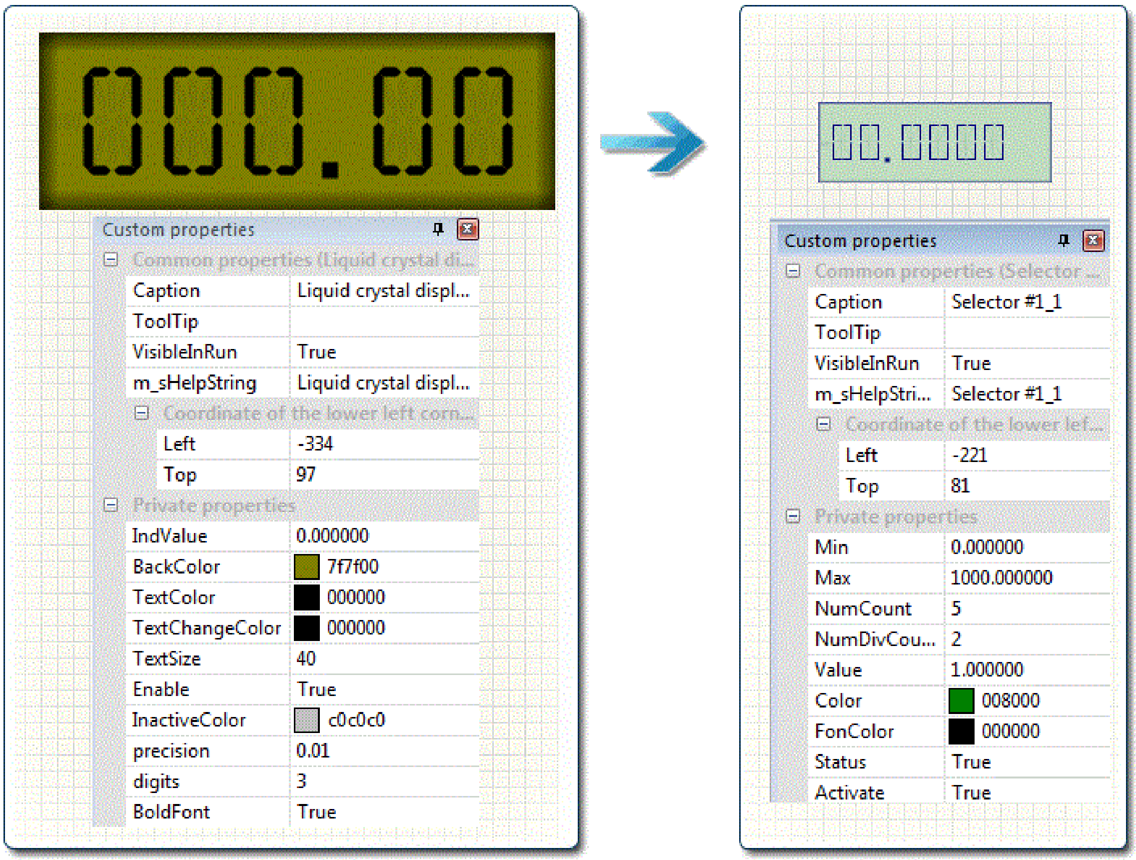


Figure 1.11.2. Design options of liquid crystal display

Глава 10.Multipage projects

ZETView project is a part of automatic process control system developed by the user. It is possible to create a project on one or several pages (by means of interconnecting them with each other). Each of the pages, being a part of the project, should perform functionally complete action, related to a particular task of the user.

"Cross-page link" – component. It is used to establish a link between the components located at different pages of the project. In order to add a page, enter main menu, select option "New page" or use hot-key combination <Ctrl>+<N>. Under the tray there will appear a tab with the name of the new page (Figure 1.12.1a).

Figure 1.12.1. Multi-page project

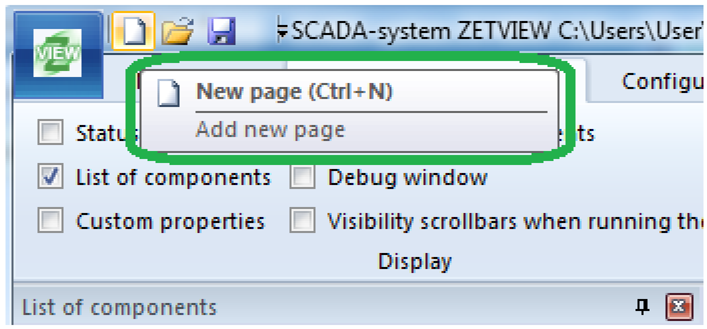


Figure 1.12.1a. New page <Ctrl>+<N> project

In order to delete a page from the project, enter the page you want to delete, click the cross icon to the right from the tabs. If the page is not empty, ZETView will require confirmation of action.

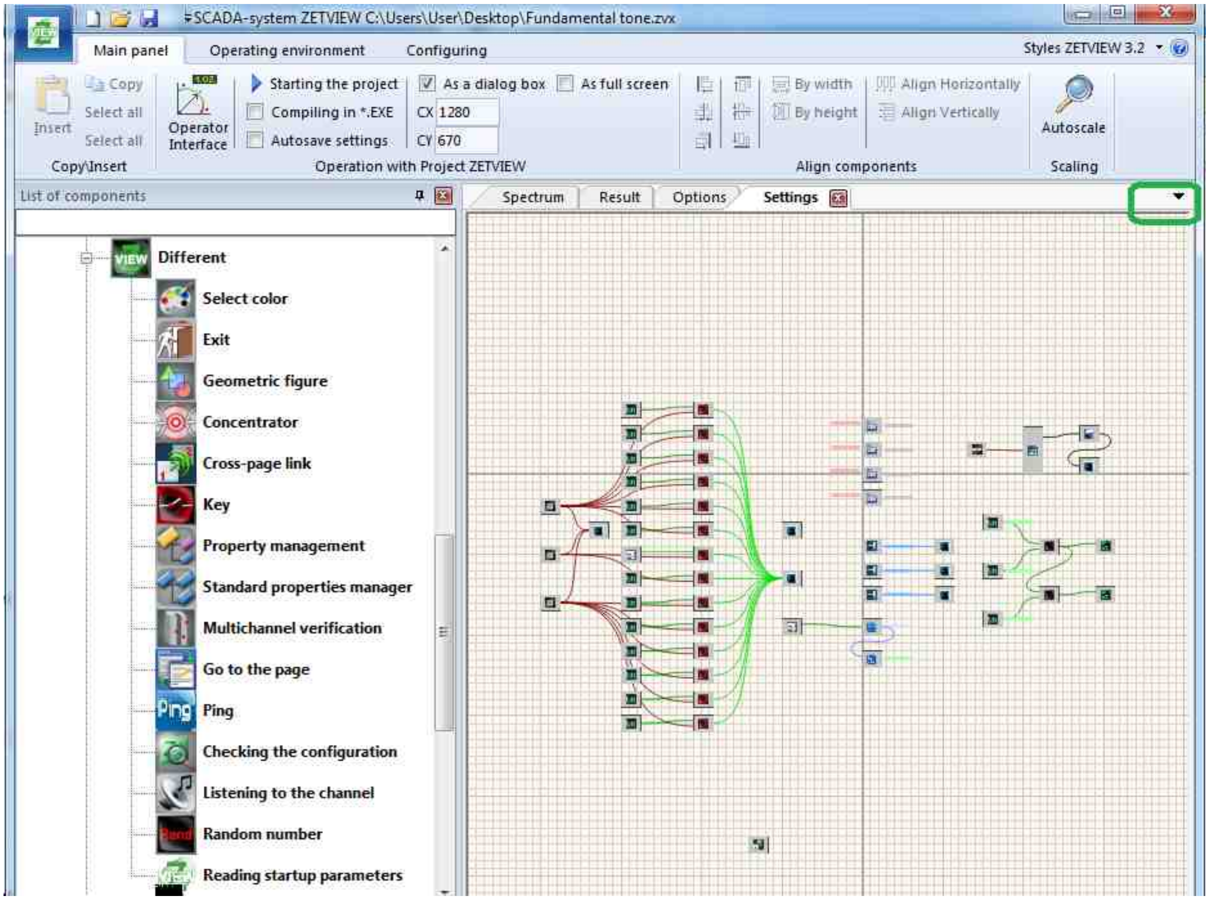


Figure 1.12.2. Page deletion process

Name of the active page is shown in bold type, active page has a cross icon allowing to close the page (i.e., to delete it from the project). Page display section has a frame. Frame color changes depending on which page is now active. Names of the pages and other properties of the pages are available in "Custom proprties" window.

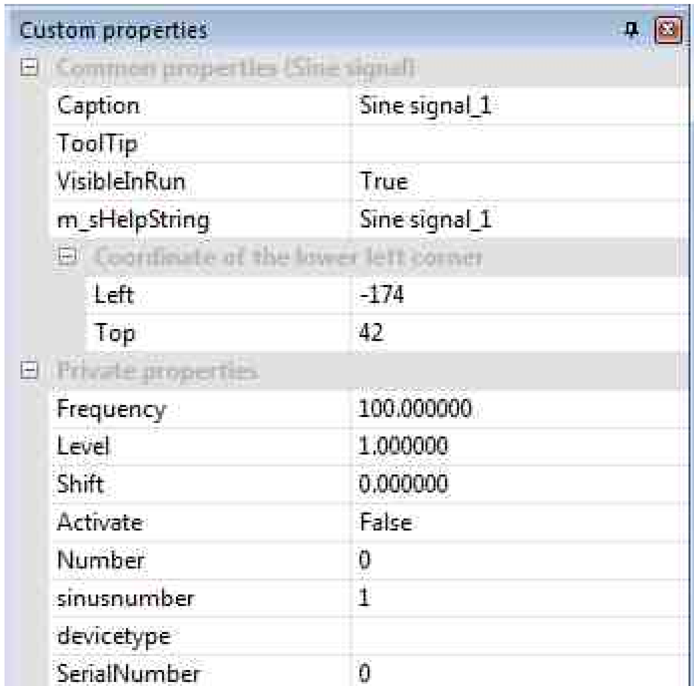


Figure 1.12.3. Custom properties of the page

In order to change appearance of the page or to change its name, enter the page, select "Page" option in the menu "Custom properties".

StartVisibility (True/False)	Page visibility at the start of the project. True – the page will be displayed as the project is started, False – the page will be hidden
Back color (Background color)	The page will have the set color both in development and operation mode.
Grid color (Grid color)	Coordinate grid visibility and color. Note: coordinate grid will be depicted only in the course of project development. As the project is in operation mode, the coordinate grid is not displayed.
Visible grid (true/false)	Coordinate grid visibility. True – the coordinate grid is displayed, False – the coordinate grid is not displayed. Note: the coordinate grid can only be represented in the course of project development. As the project is in operation mode, the coordinate grid is not displayed whatever the settings of the present parameter are.

Visible on start (true/false)	Visibility of the page at the start of the project. True – the page is displayed as the project is run. False - the page is not displayed as the project is run.

In order to create a link between the components located at different pages, use the component "Cross-page link" (External devices components - > Miscellaneous).

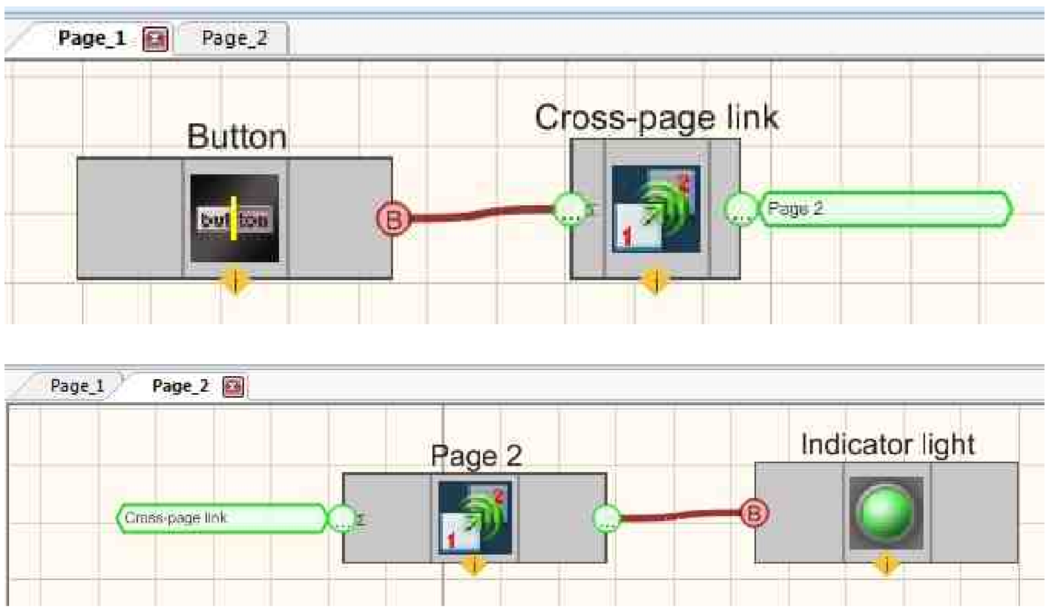


Figure 1.12.4. Creating a link between components from different pages using "Cross-page link" component

Note: For the convenience of the user, it is also possible to create a link between components from different pages without using "Cross-page link" components. Click the output contact of the first component at first page, and then click the input contact of other component from the second page.

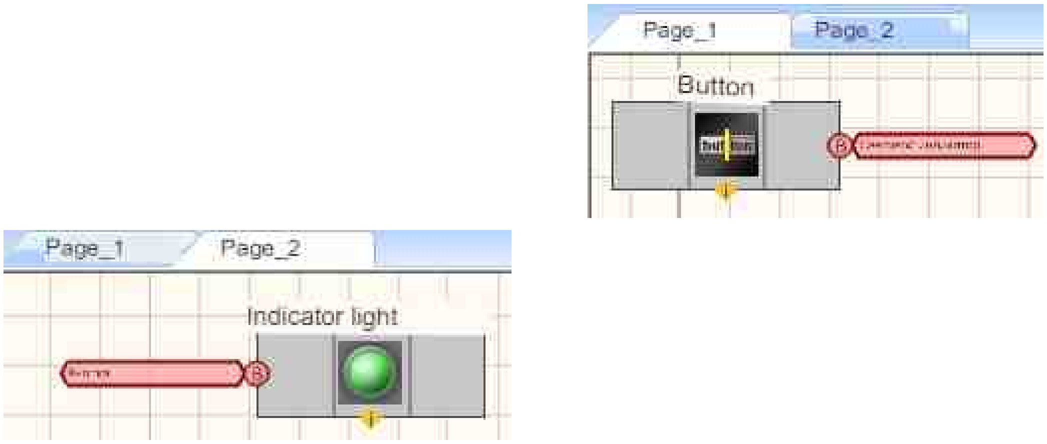
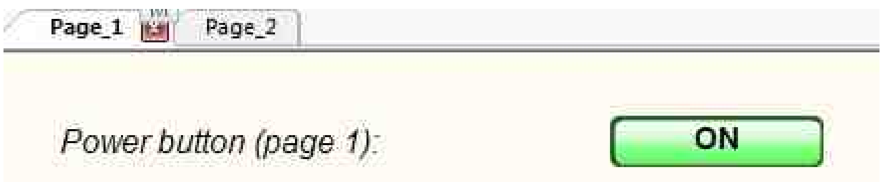


Figure 1.12.5. Creating a link between components from different pages without using the link "Cross-page link"

Project operation result



View the example in ZetView

Глава 11.Start, stop and saving the project, COMPILING to EXE

Upon completion of project development and setting, it is necessary to save it and to perform test run.

To save a project, use "Save as…" / "Save" in the main menu or <Ctrl>+<S>. If a new project is saved, ZETView program will offer to select a directory for saving the file. By default, ZETView projects are saved to the following directory:

C:\Documents and Settings\All Users\ZETLab\SCADA\Projects

When saving the project, the user can select a folder or create a new one. It is recommended to save the project everytime before starting it (so that to save the changes and to avoid mistakes).

To start a project, click "Starting the project" key at the main panel (the key will change its color for orange) or use <F5>. As the project starts, the control elements (keys, selectors, lists) and display elements (graphs, tables, indicators) will become active, i.e. it will be possible to use them.

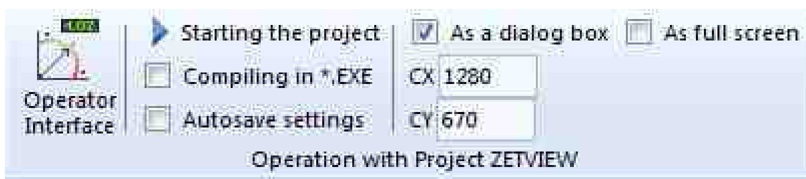


Figure 1.13.1.Starting a project

To stop a project that is being run, click <F5> or select the command "Start the project" at the main panel of ZETView.

Saved SCADA-project has *.zvx extension. Double-click *.zvx file to start the project. It can be stopped for further editing in ZETView. If the *.zvx file is opened with "Open" command from ZETView main menu the project will be opened in ZETView and it will be also available for editing.

ZETView also has a possibility of compiling the project to executable application – EXE – file. In the case if SCADA-project is compiled to EXE-file, it will not be available for viewing or editing. In this case, it will only be possible to start it and to change some of the parameters settings (the parameters that can be changed are set by the developer).

In order to compile the project to EXE, set the flag "Compile to *.EXE" at the main panel of project operation (at this stage the view of "Start project" key should change) and click "Start project".

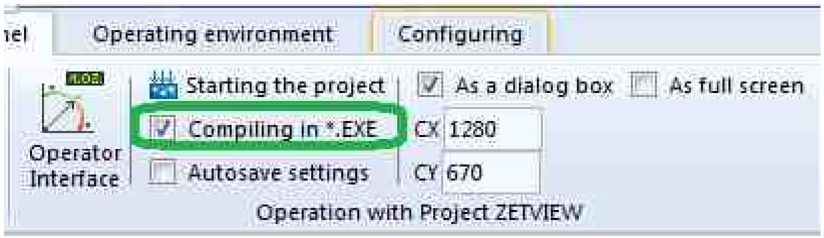


Figure 1.13.2.Compiling the project to EXE

After having completed compiling the EXE-file, ZETView program will produce notification of executable program module readiness and will offer to start the module. The executable file is saved to the same directory and has the same name, as the source *.zvx file of the project.

ZETView also allows to set widow units in the course of compiling the project to *.EXE. In order to do that, set the flag "As a dialog window" and set window units in fields CX and CY. The units are set in px. It is possible to evaluate the units of the compiled object window using the line in the bottom right part of the program (it shows current units of the window). Flag "As Full Windows" is used to create a full-screen program view.

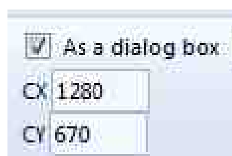


Figure 1.13.3. Current units of the window

Part 2. SCADA ZETView: basic design principles

Глава 1.Introduction

We have just finished the first stage of ZETView training – so now the user can make the first conclusions. Since simple manipulations with ZETView and its components do not cause any difficulties, we can now proceed with the second part of the present user manual, which has informative rather than instructive character. SCADA ZETView has a large number of components, which still increases. Therefore, it is difficult for the beginner to learn about so many components. That is exactly why this chapter is to deal with minor tasks of various complexity that may arise in the course of SCADA ZETView operation (from text processing and data input to operation of virtual measuring instruments). Just like other systems of higher complexity, SCADA ZETView and some of its components have certain operational specifics, which is described in this part of the present manual.

The structure of the second part of the present manual does not imply that the user has to study it completely. It will be good enough to revise a particular clause when addressing a particular task.

In order to create a project for addressing particular tasks in ZETView SCADA system, a components library is required. This library will be used for adding components to the project page. These components together with connection links between them form a full-scaled system. Each component is located in corresponding part of the library along with other components with similar functions. This classification of the components allows the user to save considerable amount of time necessary for the search of the necessary component, which contributes to universal features of the program.

USER-FRIENDLY

graphical programming process allows the user to fully focus on addressing the assigned task without going into details of the programming specifics!

POR (price on request)

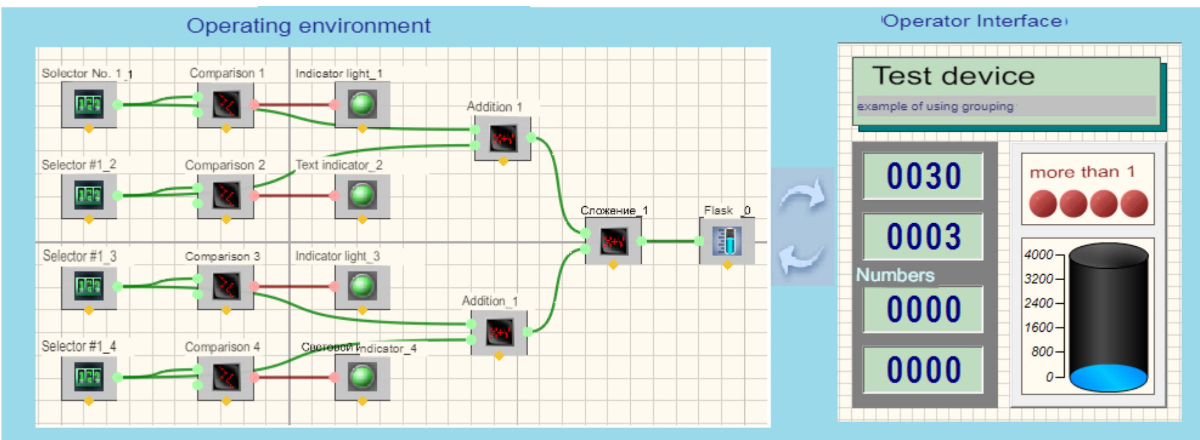
Our SCADA system allows to create flexible and scalable applications for measurements, control, and testing with minimum time and material expenses.

ZETVIEW SCADA serves as a system for collecting and processing data from devices connected to the PC (FFT Spectrum Analysis, ADC and DAC boards, multimeters). The SCADA system is also designed for process management automation.

Creating applications in ZETVIEW environment is extremely simple:
1. Adding components and configuring their attributes,
2. Linking the components,
3. Designing visual appearance of the application (UI customization).

Definition:

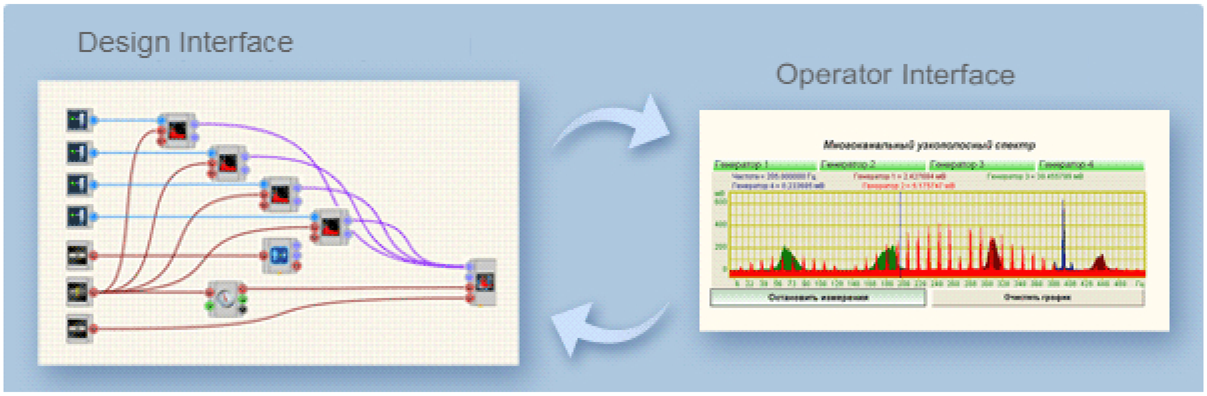
A component is a program unit that has a specific function (e.g. a button).



SCADA SYSTEM FEATURES

With its flexibility and scalability, ZETVIEW SCADA system can be used at all process stages, from modeling and prototyping products to a large-scale production testing. ZETVIEW integrated environment for measuring signals, processing results, and exchanging data allow to improve performance across the enterprise.

At the same time, ZETVIEW SCADA system is very simple and user-friendly. A newbie user without programming experience can in a relatively short time (from several minutes to several hours) create a sophisticated program for data collection and facility management having a beautiful and user-friendly man-machine interface. The created applications can be used independently (beyond the scope of ZETVIEW programming environment), which considerably facilitates operation and training process.

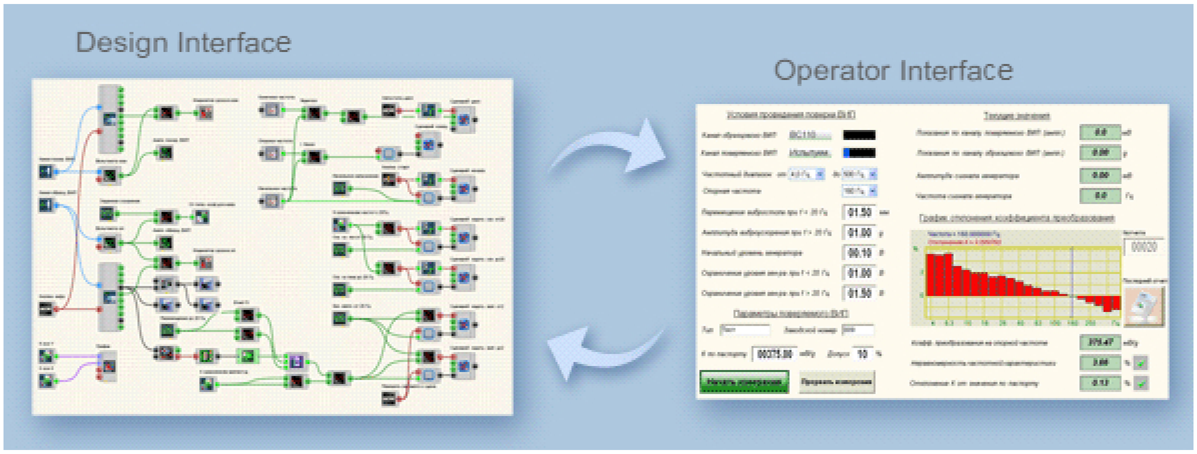


ZETVIEW SCADA system: key advantages

- a full-fledged programming language Visual Basic and Visual C++;
- user-friendly graphical programming process;
- broad opportunities for data collection, processing, and analysis, management of virtual and real instruments;
- various forms of reporting and graphicing;
- data transfer via USB, Bluetooth, Ethernet, Wi-Fi;
- compatibility with dozens of ADC/DAC modules, FFT Spectrum Analysis, voltmeters, and power generators;
- compatibility with third-party controllers and transducer modules based on OPC client-server technology;
- application templates, dozens of examples;
- DEMO-mode operation without connection to the equipment using ZETKEY;
- training services and technical support.

ZETVIEW SCADA system: application range

- automotive industry;
- telecommunications;
- aerospace industry;
- semiconductor industry;
- petroleum industry;
- electronics development and manufacture;
- process management;
- bio-medicine.



NEW FEATURES OF SCADA SYSTEM ZETVIEW

How can one evaluate the condition of the test sample?

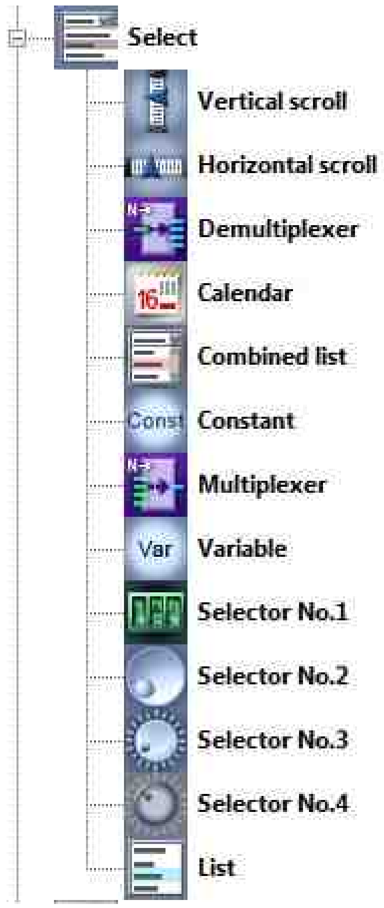
It is very easy!

In SCADA system ZETView for this purpose there is a simple and intuitively clear technology. To use it you need:
Create a three-unital model of the object under test.
Download it to SCADA system ZETView
Indicate the location of the sensors and their corresponding measuring channels
Run the test program
On the signal from the SCADA sensors, the ZETView system will determine the levels of the measured parameters in the test points and interpolate them for the entire model. The overall picture will be a model, colored depending on the intensity of the color from blue (low level) to red (high level).

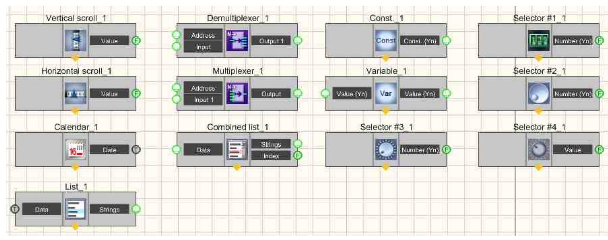
A single look at the test sample will be enough to assess the status of the object, draw conclusions and, if necessary, take a decision.

Глава 2.Data, select, indication, buttons

In the present, it is hardly possible to find a program that would not inform the user of its actions and intermediate results, especially when it comes to measurements and data processing. Graphical interface, control elements, various notifications and other means of user interaction form an integral part of any modern application. In the first part of the present user manual, it has been mentioned that it is possible to compile the projects to exe-file. This format implies that the operator does not have a possibility to get access to project composition or to change settings of components properties. In fact, the project is represented as an independent program. Hence, it is important to provide the operator with an opportunity to interact with the project and to view the results.
The components used for data input are available in "Select" clause.



2.2.1a. Devices from "Select" clause



2.2.1b. Components of "Select" menu- developer interface

Selectors are normally used to enter numerical values. In the "Select" clause there are six types of selectors available. All these components are used a get an output value set by the user. The components differ in design and convenience of implementation in particular project.

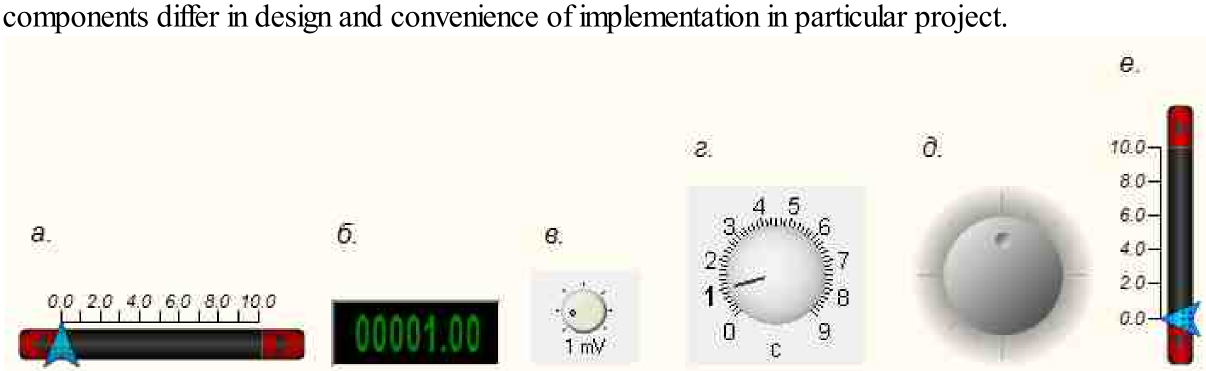


Figure 2.2.2. Components of "Select" menu – operator interface. Selector types.

To switch between status it is more convenient to use selector of second type while for entering a precise numerical value selector of first type is normally used. Selector #1 allows to set a precise value (double-click the component while the project is in operation, enter the required value in corresponding field and click "Enter"). It is possible to set input values limits and other parameters in custom properties menu.

Operational logics of SCADA ZETView implies signals processing in real time mode, hence, it is important to differentiate between constant values of the channel and single values. For example, selector values are used only one time – they are used only as the value is changed. "Constant" component value is also used only one time.

In the case of complex distributed systems, "Multiplexer" and "Demultiplexer" components are used. These components allow to to commutate a channel in accordance with a particular address.
Lists are also an important element of selection menu. ZETView system has usual and combined lists.

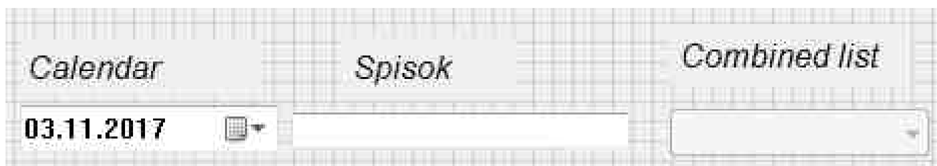


Figure 2.2.3. "Select" menu components – operator interface. Calendar and lists elements.

These components are often used by the operator for settings selection. The list can be produced manually or automatically as lists components are set in the value line.

SCADA ZETView system has a number of various indicators used for results representation. For numerical values input a liquid crystal display and digital indicator are used. Liquid crystal and digital indicator are often used for voltmeter and other measuring instruments results representation. Just like many other ZETView components, the indicators are used for signals processing in real time mode and allow to display the last received value till a new value is obtained.

For indication of measured values in real time mode, level and pointed indicators are used. In the case if, e. g., it is necessary to inform the operator of process completion, a process indicator is used. As you can see in the figure, ZETView has two types of process indicators: with numerical and control signal input. The first type is used to show the degree of process completion, while the second type produces response to input signal and is used in order to inform the operator that the process is being run or is temporary suspended due to mistakes in settings or project development. Besides, there is a light indicator changing its color depending on logical status at the input.

Figure 2.2.4a. "Indication" window.

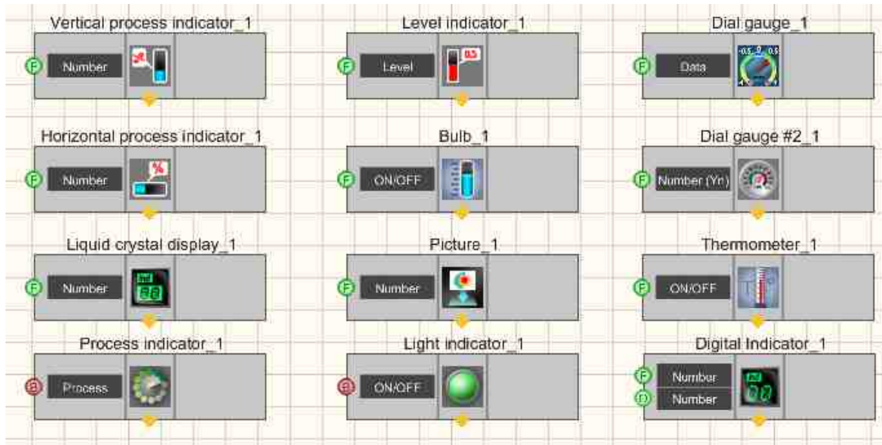
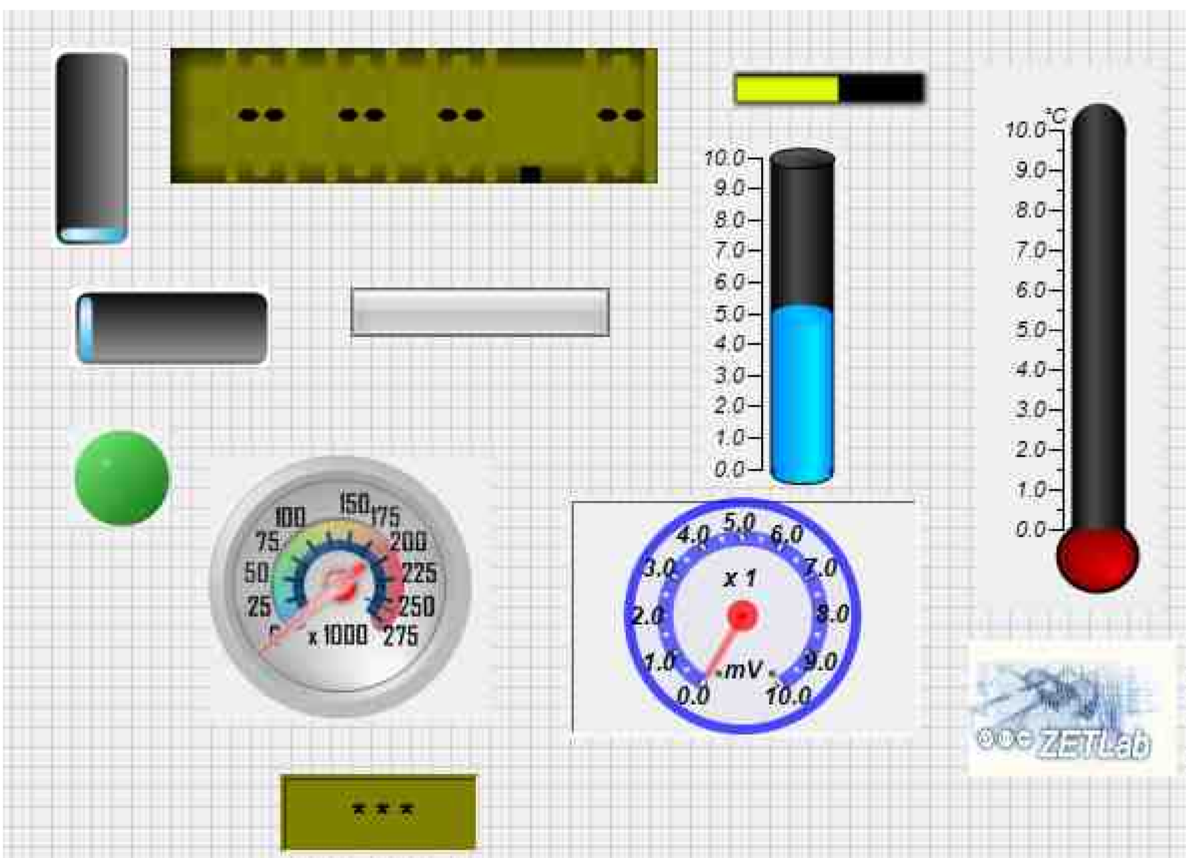
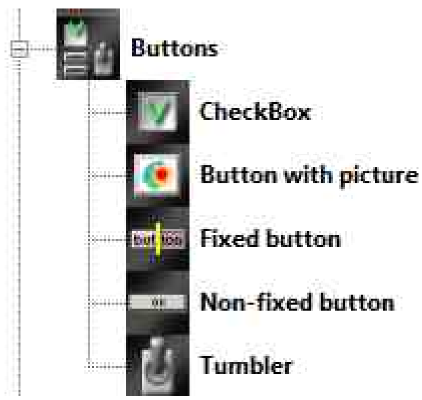


Figure 2.2.4b. "Indication" section components – developer interface Figure



2.2.4c. Components of "Indication" menu – operator interface

Components of "Buttons" menu are often used for projects control. The components of "Buttons" menu are used for control of corresponding components. SCADA ZETView has several types of buttons. Actually, the keys are classified into two groups: fixed and non-fixed. Fixed keys are used for producing a constant logical level signal – "1" or "0". Non-fixed buttons have initial logical level "0". Upon their activation, they produce a single impulse "1" and then go back to initial status "0". Tumbler, fixed button and flag produce "1" value as they are activated and "0" as they are disabled. In most cases they are connected to inputs of "on/off" type. A key with icon sends "1" impulse in the case of activation. In most cases, keys with icons are connected to contacts of "Synch" and "Impulse" contacts.



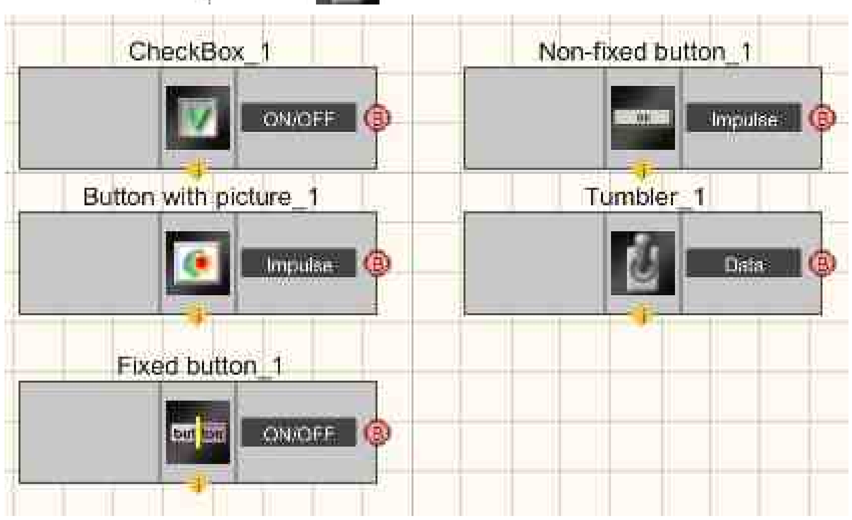


Figure 2.2.5a. "Buttons" window.

Figure 2.2.5b. "Buttons" section components – developer interface Figure

2.2.5c. "Buttons" section components – operator interface

Figure 2.2.5a-c depicts components of "Buttons" section in SCADA ZETView. Except for the component "Non-fixed button", all the remaining buttons are fixed and differ from each in configuration parameters. CheckBox sends logical "1" to the channel as the flag is selected and "0" as the flag is deselected. A button with icon operates in the same way as a usual fixed button (the only difference is that the user can select a particular icon for the button).

Let us assemble a SCADA ZETView project.

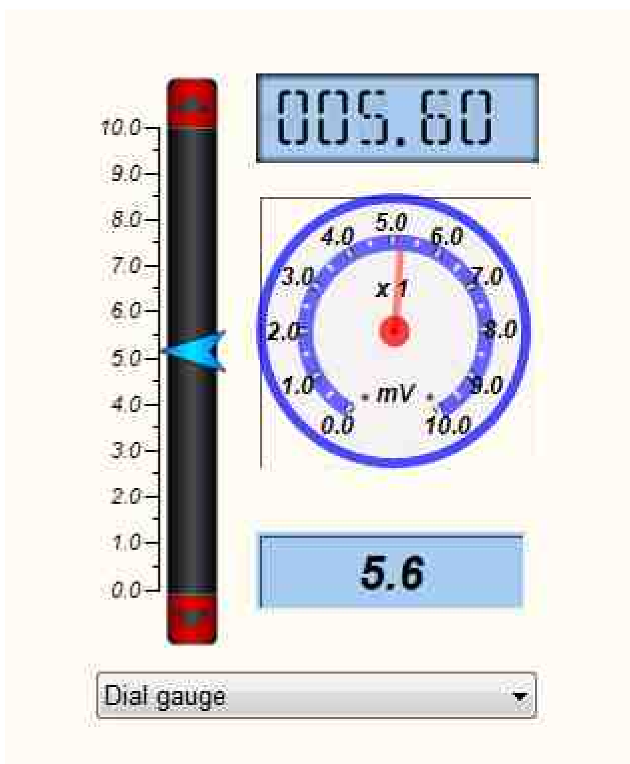


Figure 2.2.6. First project.

The user can select indicator type from the list (the indicator will show the result from vertical scroll). The selected indicator will show the value change upon scroll.
Let us start SCADA ZETView and arrange the components shown below in accordance with their position in figure 2.2.7.
"Select" -> Combined list

"Arithmetic" -> Adding
"Select" -> Vertical scroll
"Select" -> Demultiplexer
"Indication" -> Liquid crystal display
"Indication " -> Dial gauge
"Indication " -> Digital indicator

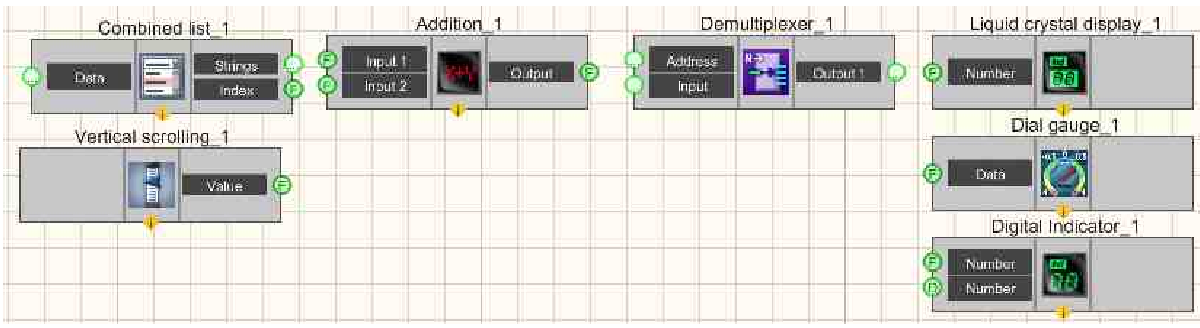


Figure 2.2.7. Deployment of components.

The component "Demultiplexer" may have several outputs depending on particular settings configuration. As the output number is forwarded to "Address" input, the demultiplexer will forward the data from "Input" contact to the specified output. Let us configure demultiplexer parameters. To do it, right-click the component and select "Custom properties". Then in the right part of SCADA ZETView system there will appear custom properties window.

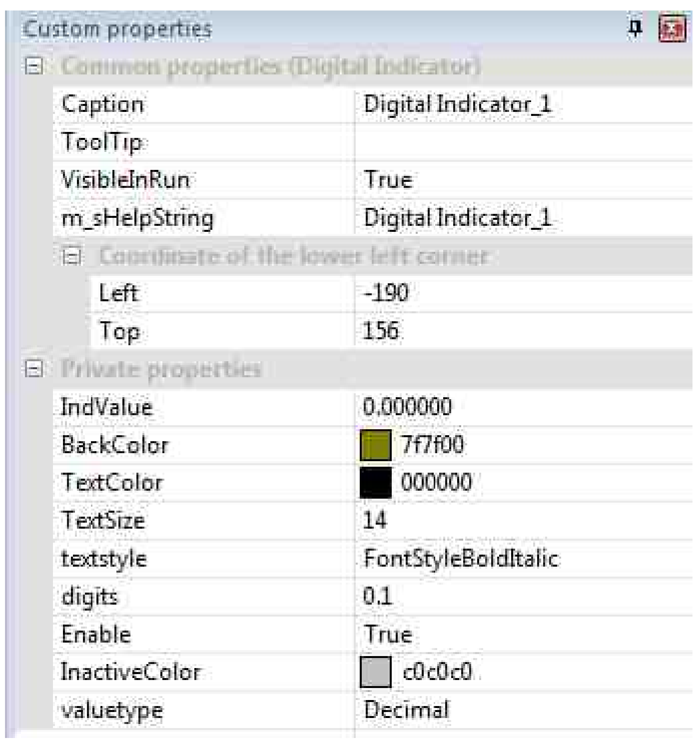


Figure 2.2.8. Demultiplexer: custom properties.

Let us set 3 as "Output" value (see figure 2.2.8.). The tooltip shows that the "Output value" controls the amount of demultiplexer outputs. As the component properties are changed, there appears two more outputs (see figure 2.2.9.).

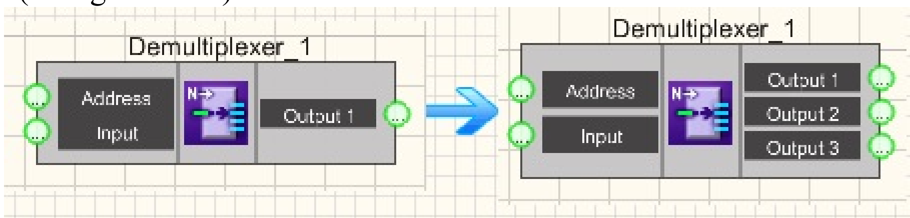


Figure 2.2.9. Changing demultiplexer parameters.

Let us establish connections as shown in figure 2.2.10.

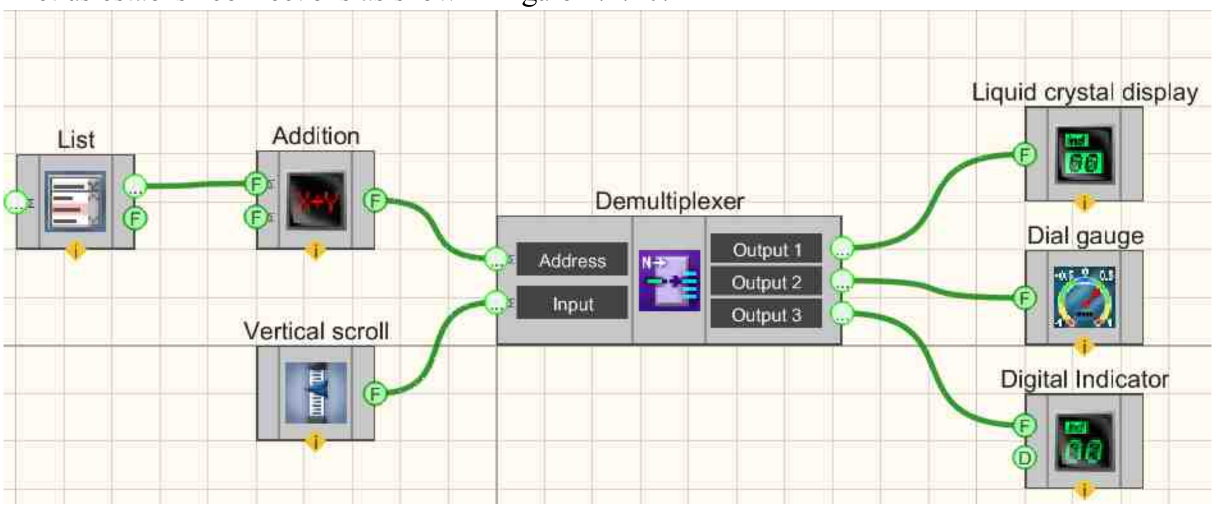


Figure 2.2.10. Establishing connections.

Now let us change settings of the remaining components (see figure 2.2.11)

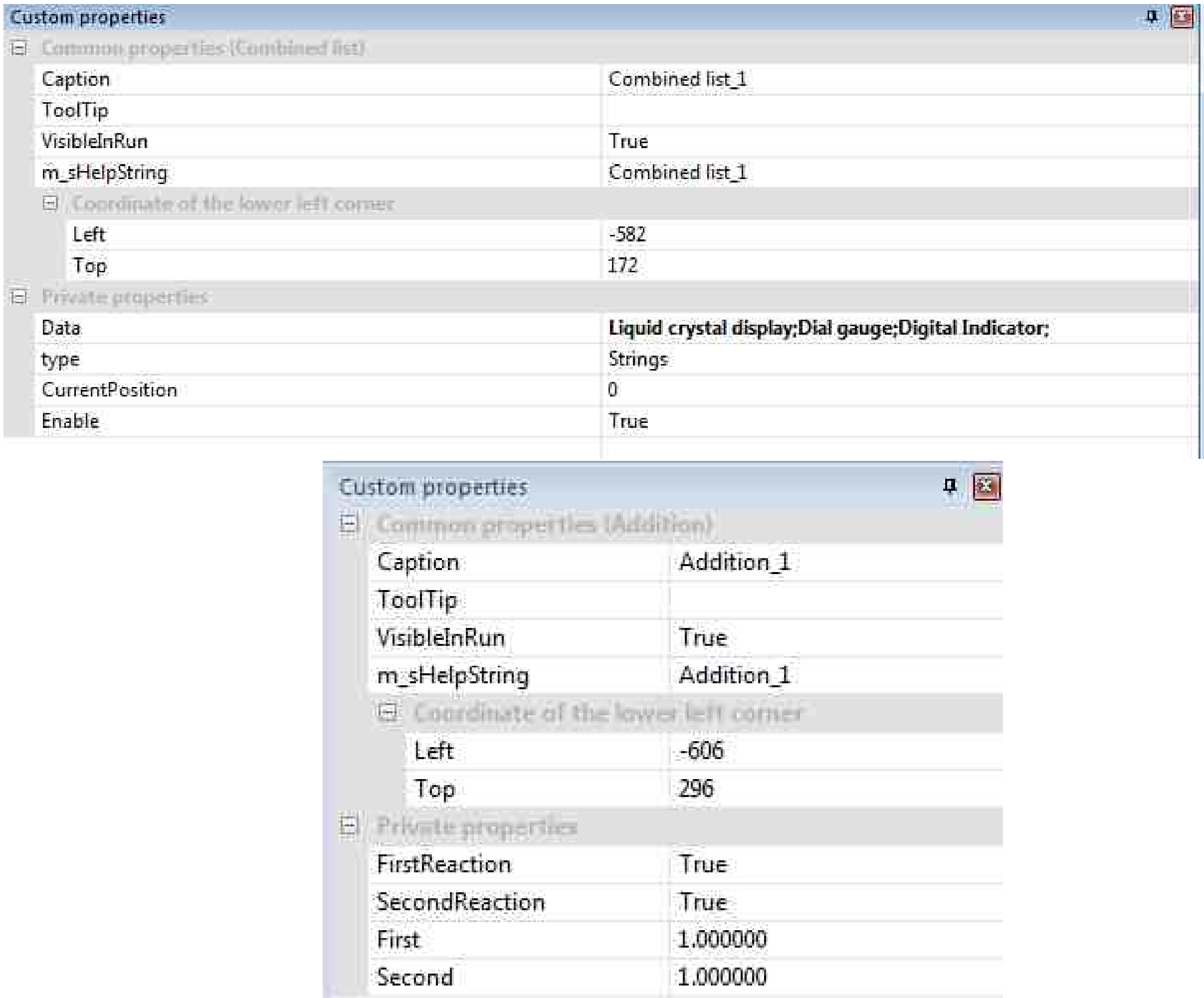


Figure 2.2.11. Components properties.

Enter combined list properties, set the items separating them with ";", set "Indexes" value in "type" property. Thus, as one of the three items of the list is selected, the component will set the selected index to the output of the component. The items of the list are arranged by order starting from 0. This is exactly why there is "Adding" component in the project. Demultiplexer receives output contact number (to which the data is to be transfered) to its input and numbers them starting from 1. In order to provide compliance of list items to outputs of demultiplexer, the "Adding" component is used. It adds 1 to list item. Let us change settings of adding component, in particular, the "Second" property (value of the second output – the second summand), the first summand – item from the list.

Enter operator interface, set the project display mode. Make sure that "Main panel" is available in the tray, click "Operator interface" key.
After that, all the components will be depicted together. For the purpose of convenience, arrange them as shown in figure 2.2.6. "Operator interface" allows to change linear units of the components, e.g. font size (TextSize property) of liquid crystal indicator, so that the text would fit into the field of compact indicator. Additional information of display mode settings is available in chapter 11 62.

Let us save the project from the main menu and start it using "Start project" key of the main panel of F5 key.

Now it is possible to select one of the indicators from the list, move the scroll and obtain corresponding results. When switching between indicators, the previous indicator depicts the last value. It is a feature of SCADA ZETView system, since the system is supposed to process the signals in real time mode in the range from $-\infty$ to $+\infty$. Both liquid crystal and digital indicators change their values as the input values are changed. If a real or generated signal is processed, then, in the absence of the signal, the indicator value is 0.

View the example in ZetView

Глава 3.Additional properties

In the previous chapter, we have considered a simple example of components properties settings. A wide range of settings available provides convenient operation of the system. However, sometimes properties of a component depend on particular status of another component. The projects normally use complicated script, which will be considered in a separate chapter.

Properties manager - component. It is used for control of other components properties and allows to add additional contacts to the component (which are available in additional properties of the component).

The components have a special input for connection of properties manager (triangle figure at the bottom of component’s icon). If a component does not have a special input, it means that the component is controlled by means of input contacts and has no additional properties.

Figure 2.3.1. Properties manager input

When properties manager is connected to the component input, the operator can change parameters of component’s additional properties.
Please, note, that the view of properties manager depends on a particular component to which it is connected to:

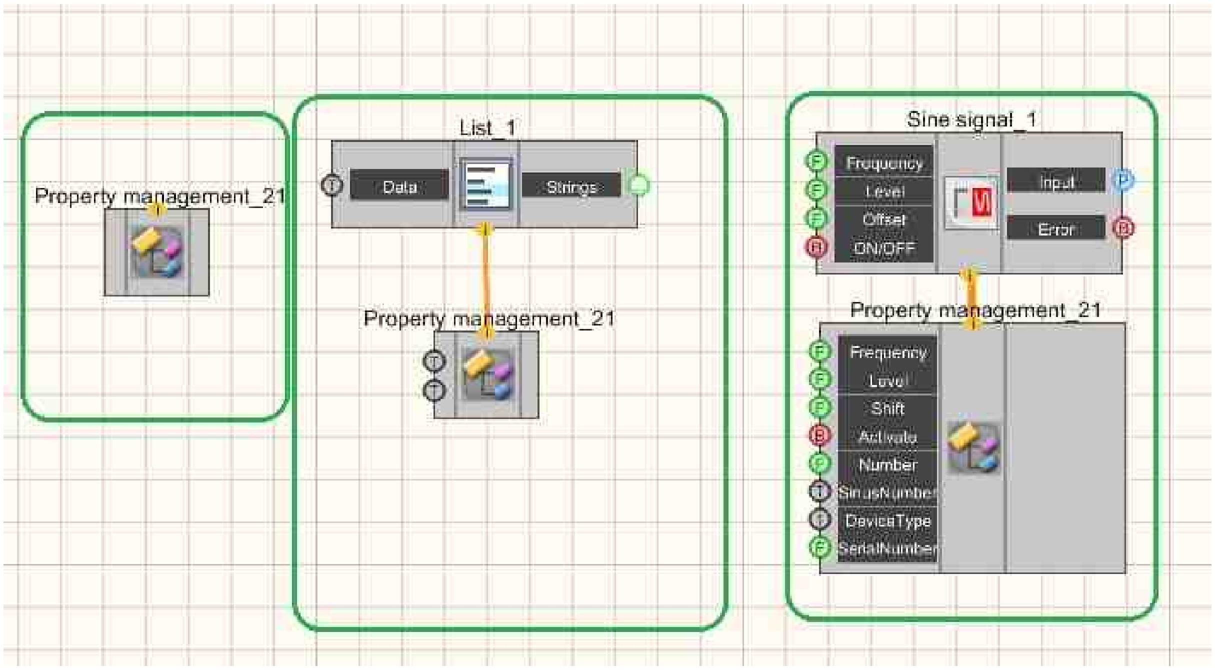


Figure 2.3.2. Various icons of properties manager element

Example No. 1

Properties manager provides the operator with access to component properties, which previously were unavailable and could be assigned only in manual mode.
For instance, figure 2.3.3 shows a usual liquid crystal indicator. Using the keys shown in the left, it is possible to change its view.

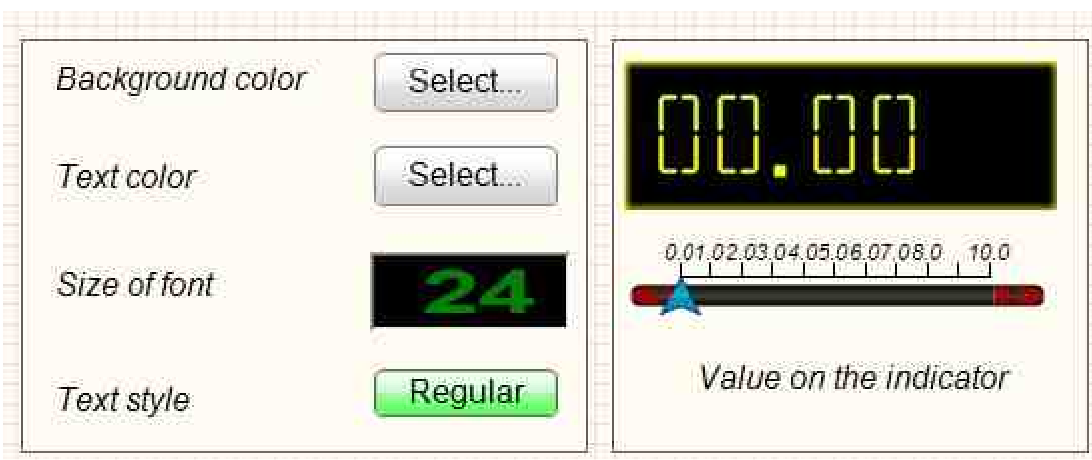


Figure 2.3.3. Using properties manager for changing indicator parameters

Let us assemble a project in SCADA ZETView.
Put the following components to the working area:
"Select" -> Horizontal scrolling

"Buttons" ->Non-fixed button – 2 pcs
"Buttons" ->Fixed button
"Select" ->Selector No.1
"Different" ->Select color - 2 pcs
"Different" ->Properties manager
"Indication -> Liquid crystal display

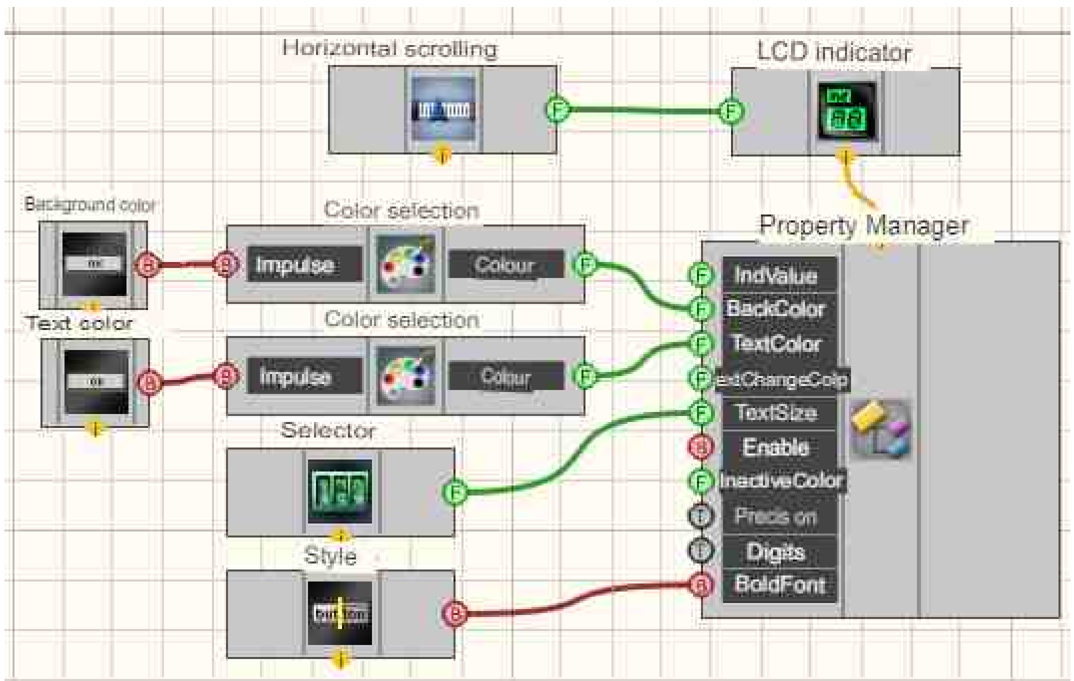
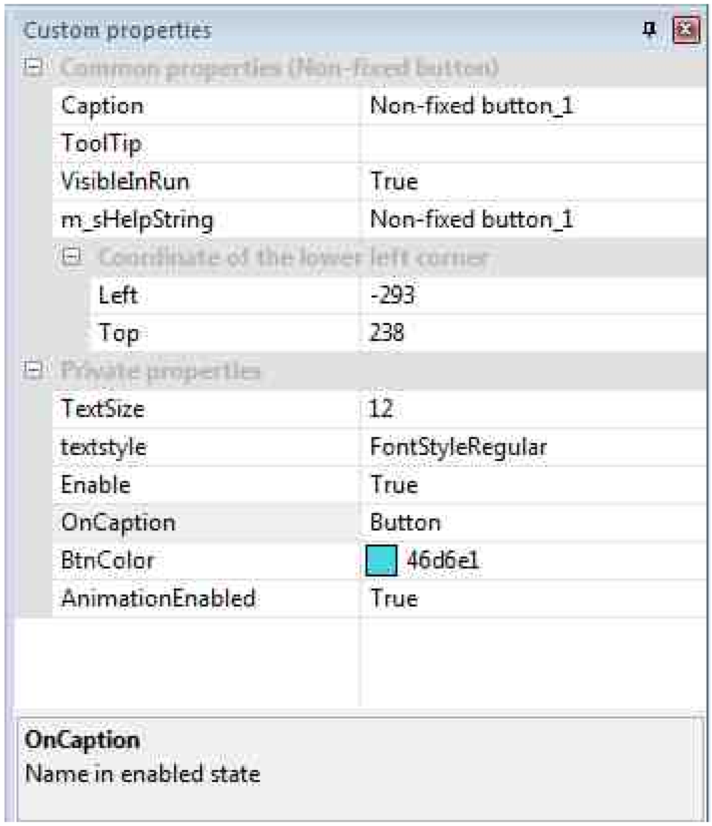


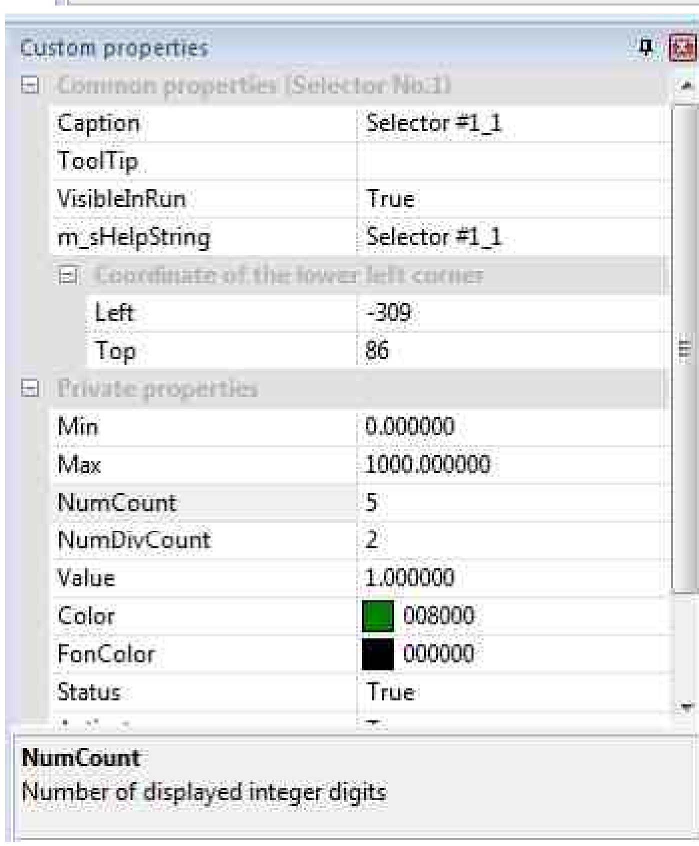
Figure 2.3.4. Project in SCADA ZETView

Connect horizontal scrolling and properties manager to the liquid crystal indicator component. As the properties manager is connected, it will change its appearance and will have several inputs.

Connect "Choose color" components to «BackColor» and «TextColor» components. Connect "Button without fixation" to each "Change color" component. Connect Selector#1 to «TextSize» input and Fixed button to the «BoldFont» input.

Then set the parameters of components "Button without fixation", "Selector#1", "Fixed button".


Custom properties
Common properties (Non-fixed button)
Caption
Non-fixed button_1
ToolTip
VisibleInRun
True
m_sHelpString
Non-fixed button_1
Coordinate of the lower left corner
Left
-293
Top
238
Private properties
TextSize
12
textstyle
FontStyleRegular
Enable
True
OnCaption
Button
BtnColor
46d6e1
AnimationEnabled
True
OnCaption
Name in enabled state


Custom properties
Common properties (Selector No.1)
Caption
Selector #1_1
ToolTip
VisibleInRun
True
m_sHelpString
Selector #1_1
Coordinate of the lower left corner
Left
-309
Top
86
Private properties
Min
0.000000
Max
1000.000000
NumCount
5
NumDivCount
2
Value
1.000000
Color
008000
FonColor
000000
Status
True
NumCount
Number of displayed integer digits

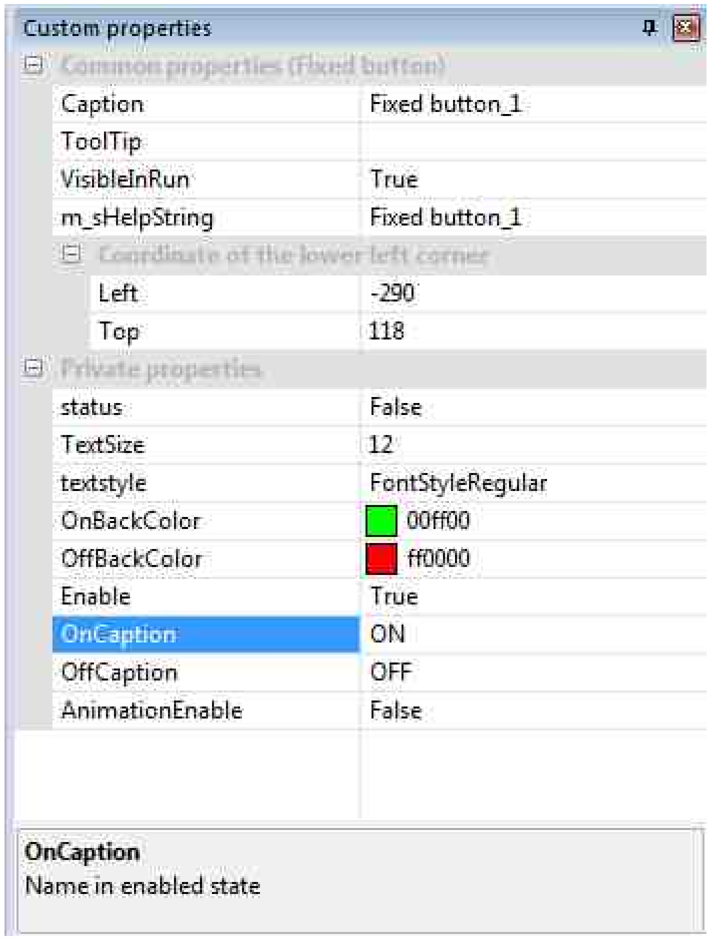


Figure 2.3.5 Components properties

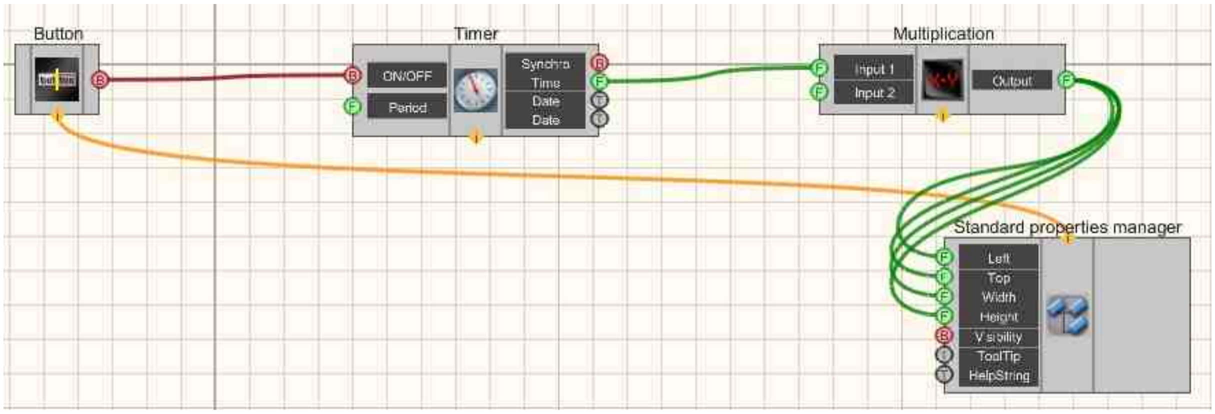
After having established the connections and setting components parameters, it is possible to set project parameters in operator mode.
As you can see from figure 2.3.2, there are comments sections near keys and indicators. To make a comment, use "Static text" component from text section. This component is available for use only in operator mode, since it is not displayed in developer mode.

View the example in ZetView

Example No. 2

Standard properties manager. It is used for the control of object location standard properties (Left, Top, Width, Height).

Project in SCADA ZETView



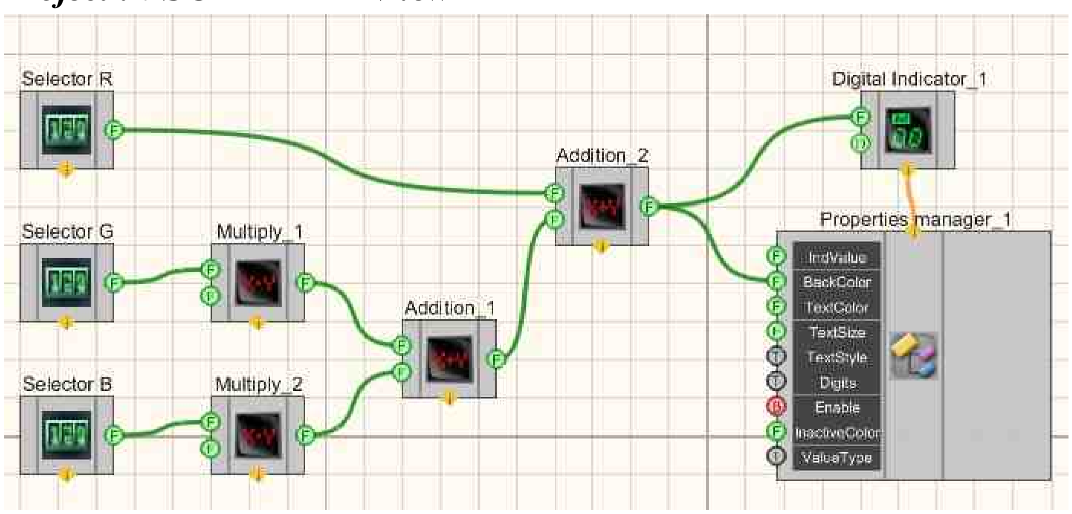
This diagram shows Standard properties 999 component, enabling control of component's coordinates anf general properties. The component changes and moves along the display.

Project operation results

View the example in ZetView

Example No. 3 Properties manager.

Project in SCADA ZETView



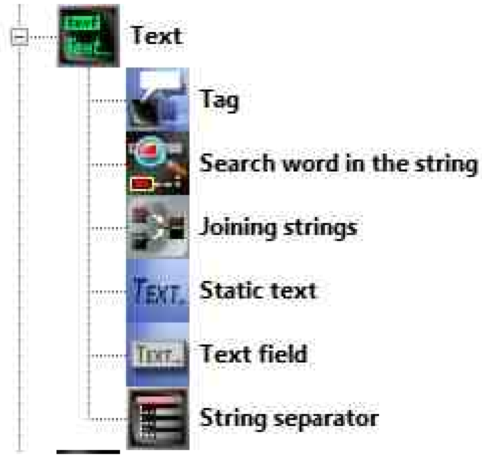
In this diagram Addition 314 component is used to find a sum of two input values. The Select 428 are used to set the numerical values of the summand. A Digital indicator 682 is used for graphical representation of output values.

Project operation results

View the example in ZetView

Глава 4. Operations with text, files, operator messages, Excel

One of the key functions of user interaction is represented by display and processing of textual information. SCADA ZETView has components for work with textual data, allowing to inform the user of project current stage, to produce warnings and error messages as well as to enter parameters and values, record the data to tables.



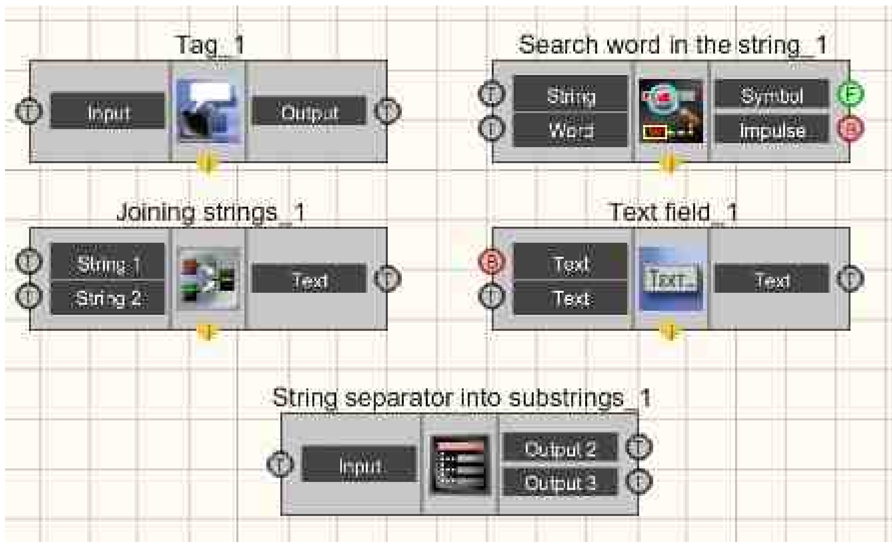


Figure 2.3.1a. Devices window, clause "Text".

Figure 2.3.1b. "Text" clause components developer interface

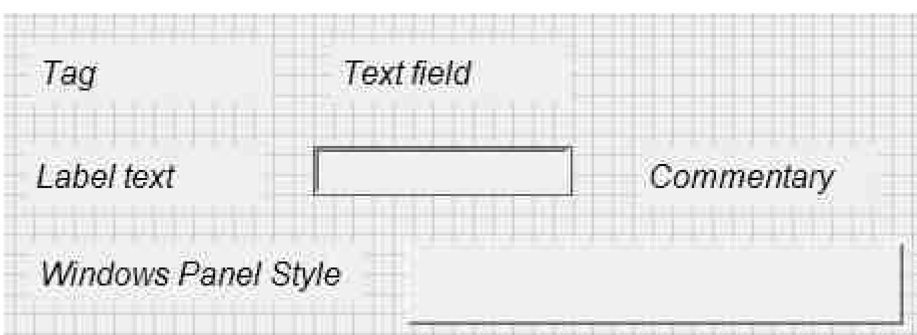


Figure 2.3.1c. "Text" clause components operator interface

"Text" section contains components for operations with textual information. These components are often used for notification, textual data receipt and processing.

The component "Text field" is used for textual data input. It allows the user to enter the text and to send it to the next component. The data is sent to the next component as it is changed or as a control signal is sent to a particular input. E.g., if a button is connected to the input, the component will transfer the data from the line upon button activation.

In addition to that, the "Static text" field has tools for textual data processing.

The component "String separator" allows to divide the string into two or more strings in accordance with the set symbol. The amount of input strings is also specified in settings.

The component "Tag" is used for text string display. In the case if it is necessary to display text array, "Data table" component is used.

"Static text" component is not available in developer mode, but it is visible in operator mode. This component is used for comments. It allows to produce text notifications for the operator.

The component "Search word in the string" allows to find a combination of symbols in corresponding text string.

"Joining string" component is used to combine two strings into a single one.

Let us assemble a project in SCADA ZETView.

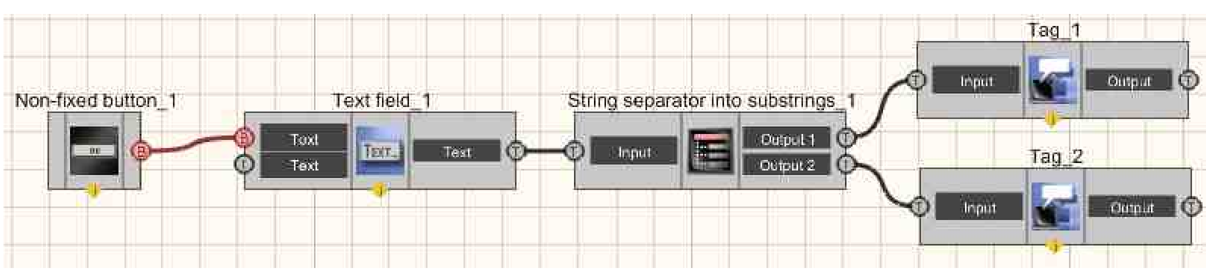
Bring the following components into working area:

"Buttons" -> Non-fixed button

"Text" -> Text field

"Text" -> String separator

"Text" -> Tag 2 pcs

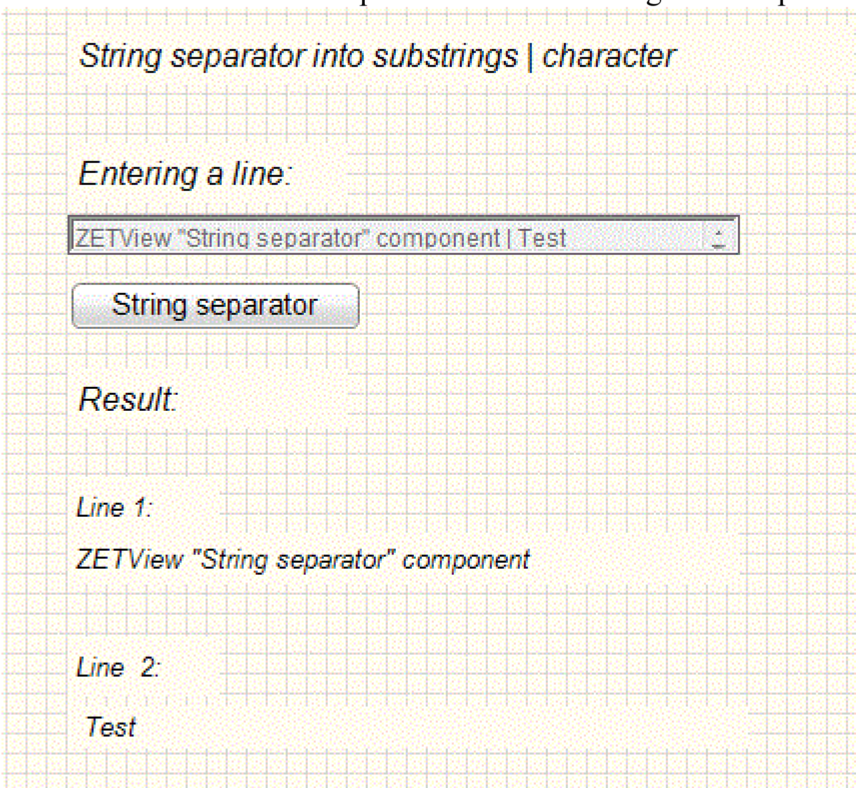


Set components properties:

Button Non-fixed button – OnCaption property: String separator ;

String separator - Delimiter property: set the symbol "|" (without quotation mark).

Then switch over to operator mode and arrange the components as shown in the figure.



View the example in ZetView

Components of "File" clause are used for operations with files in SCADA-system ZETView:

- Databases - is used for allows to work with SQL data bases in SCADA-system ZETView.
- Choose the directory - allows the operator to select a folder and set corresponding directory for the components. Thus, the operator may assign a folder for automated saving of the reports.
- Select the file - is used for selecting the file name, shows the file directory.
- Calculation of hash - is used for to verify the integrity and authenticity of files by calculating the checksum.
- Recording in a file – allows to record data to a file in the course of project operation. Thus, it is possible to use dynamic data, that do not relate to project operation.
- Run program by name – allows to run programs (or files) from the project.
- Report - is used for recording various values to a file in accordance with user template.
- Search files in the directory- allows to find the files in particular category.

- ZETLab directories – it allows to set ZETLab main directory, directories for recorded signals, data processing results, tooltips files, etc. This component contributes to convenience of using standard ZETLab directories; it is used for report purposes, file operations, etc.
- Access point (APN) - is used to select the current user (operator, engineer, etc. with corresponding access rights). The name of the selected user can be saved to database and be used for personal informational notification within the framework of complex systems, or be saved in reports made in the course of usual activities.
- The "**Structural recording**" and "**Structural reading**" components are designed to Signals recording and Play recorded data transmitted between the components of the Scada-system ZETVIEW.
- Reading system log - contains information about events occurring on the computer. This information is useful for diagnosing problems, especially on servers. Designed for viewing and managing event logs. This is an indispensable tool for monitoring the performance of the system and troubleshooting problems that have arisen. The Windows service that manages event logging is called the "Zetlab event journal". In the event that it is running, Windows records important data to the logs.


Files
Databases
Choose the directory
Select the file
Calculation of hash
Recording in a file
Run program by name
Report
Search files in the directory
ZETLAB directories
Structural recording
Structural reading
Access point
Reading from a file
Reading system log

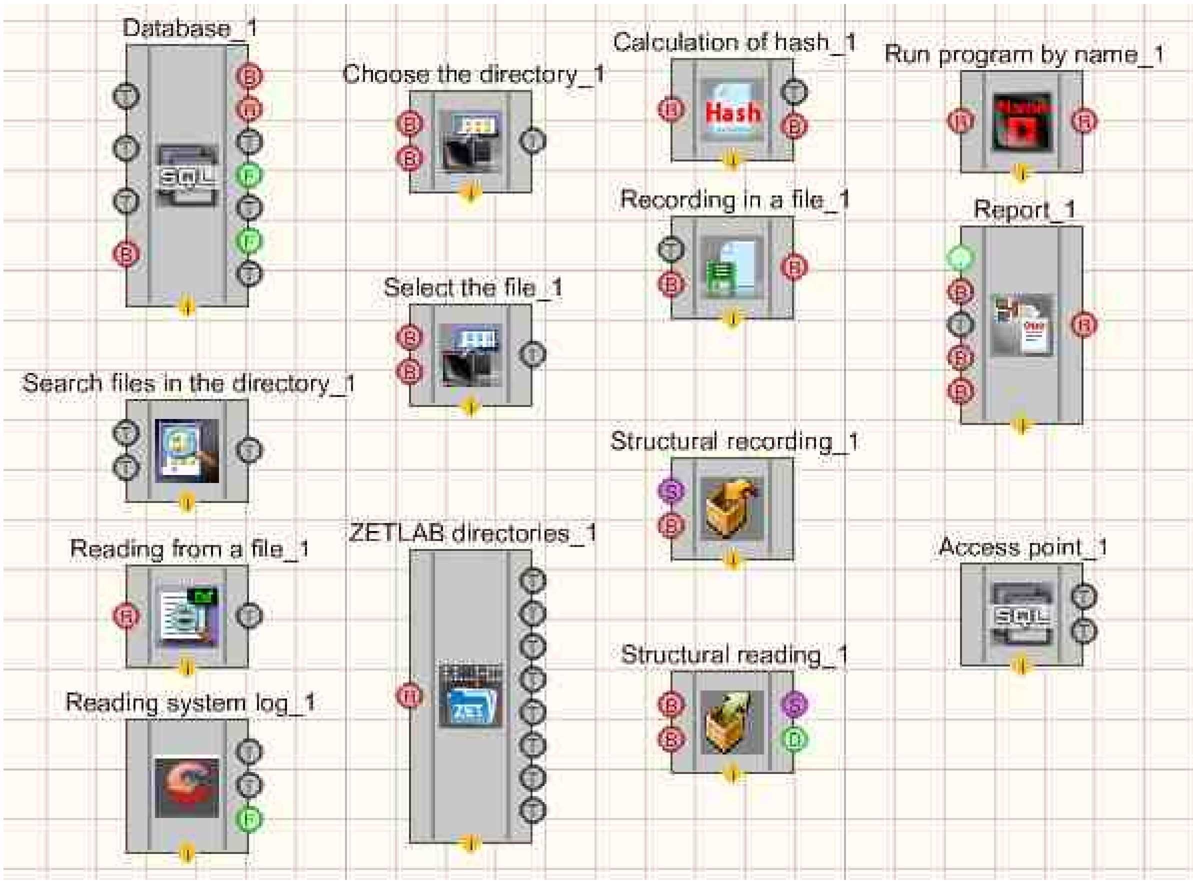
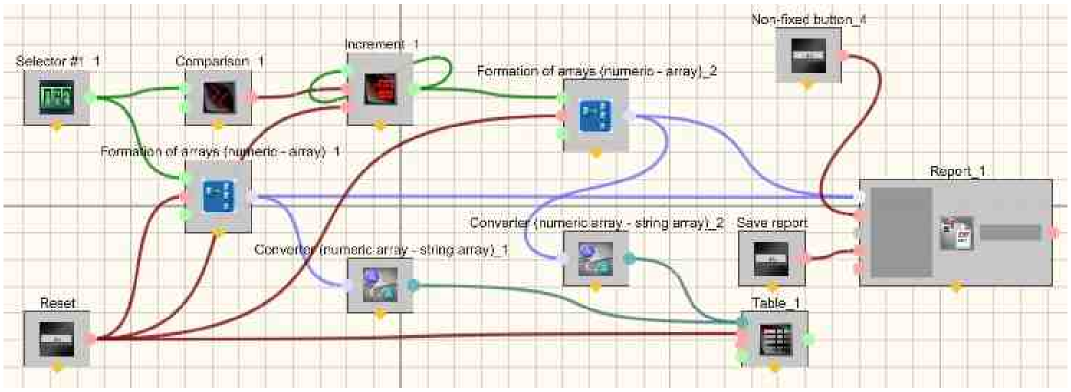
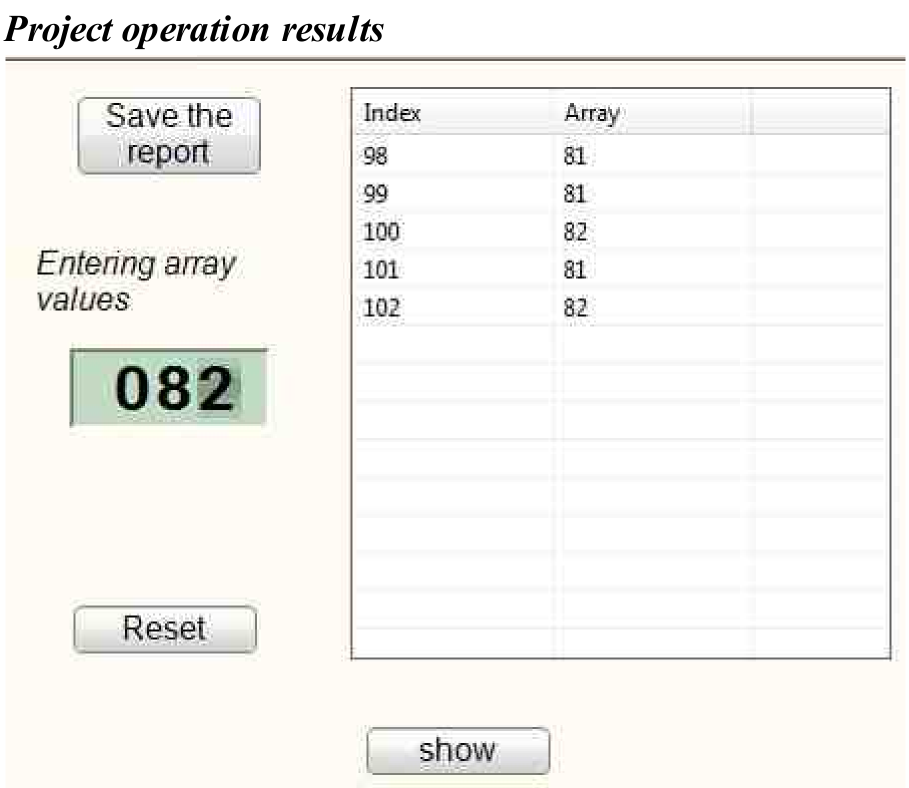


Figure 2.3.2a. Device window, "File" section. Figure 2.3.2b. Components of "File" section developer interface

Figure 2.3.2c. "File" section components, database operator interface

Let us assemble a project in SCADA ZETView. The diagram shows report creation process.

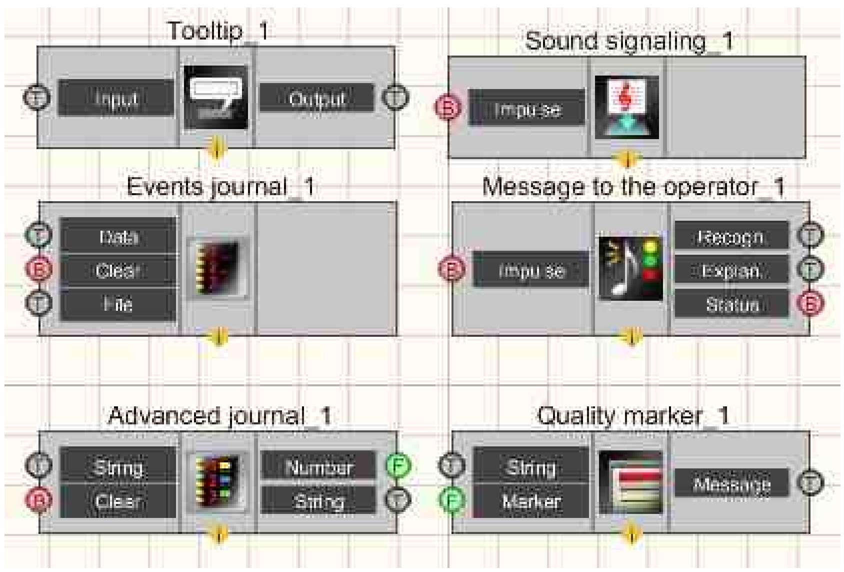




View the example in ZetView

Components of "Messages to the operator" section are used for operations with notifications in SCADA-system ZETView:

- Tooltip – depicts additional information in operator interface.
- Event journal - is used for events recording and representation in tabular form.
- Sound signaling - produces sound signal for the operator.
- Quality marker (to event journal) - is used for urgent notifications for the operator.
- Advanced journal - events recording and representation as a text note in the log.
- Message - urgent messages for the operator.



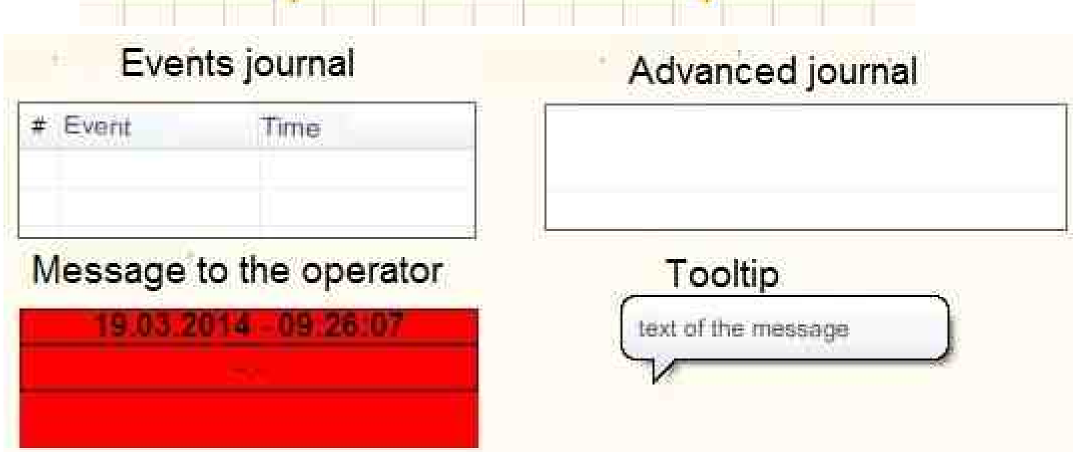


Figure 2.3.3a. Devices window "Notifications for the operator". Figure 2.3.3b. Components of "Operator notifications" section developer interface Figure 2.3.3c. Components of "Operator notifications" section in operator interface

Let us assemble a project in SCADA ZETView. The diagram shows operation of notifications for the operator.

Project in SCADA ZETView

In this diagram the component [Messages for the operator](1052) is used to display a notification for the operator. [Button without fixation](694) is used to start the notification. [Tag](1087) is used for display of additional information upon processing of [Messages for the operator](1052).

Project operation results

View the example in ZetView

Components of "Excel" section are used for adding Excel-document to the project, creating a link to a particular sheet of Excel-file and data recording to a particular cell:

- Excel Document – component for adding Excel-document to the project, output of the component contains a link to Excel-file.
- Excel Page connection to a particular sheet of Excel-file, has additional function of printing a page.
- Excel Cell – is used for data recording to a particular cell.

Components of "Excel" section are not depicted in operator mode and can be added to the project only in developer mode.

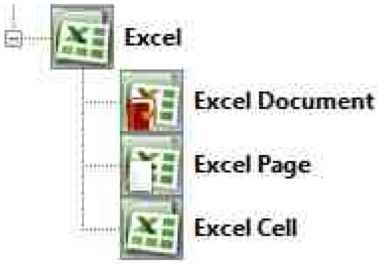
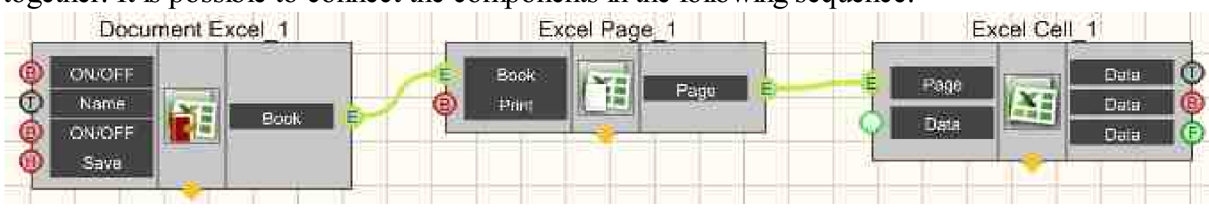


Figure 2.3.4a. Devices window, "Excel" section Figure 2.3.4b. Components of "Excel" section, developer mode

Description:

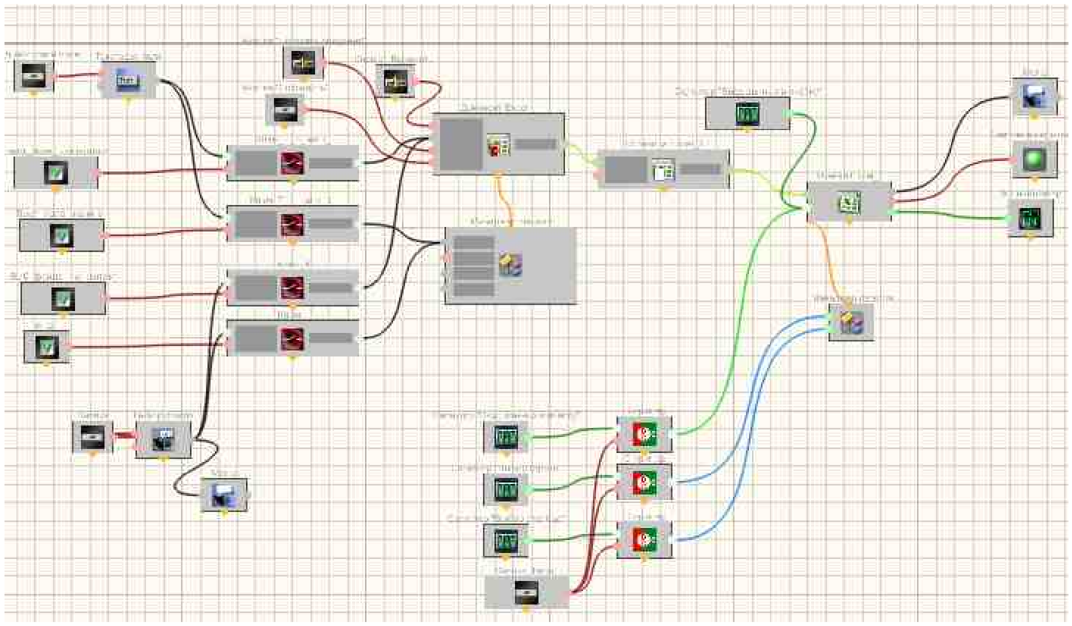
In order to record data to excel-file, "Excel Page" and "Excel Cell" components are used together. It is possible to connect the components in the following sequence:



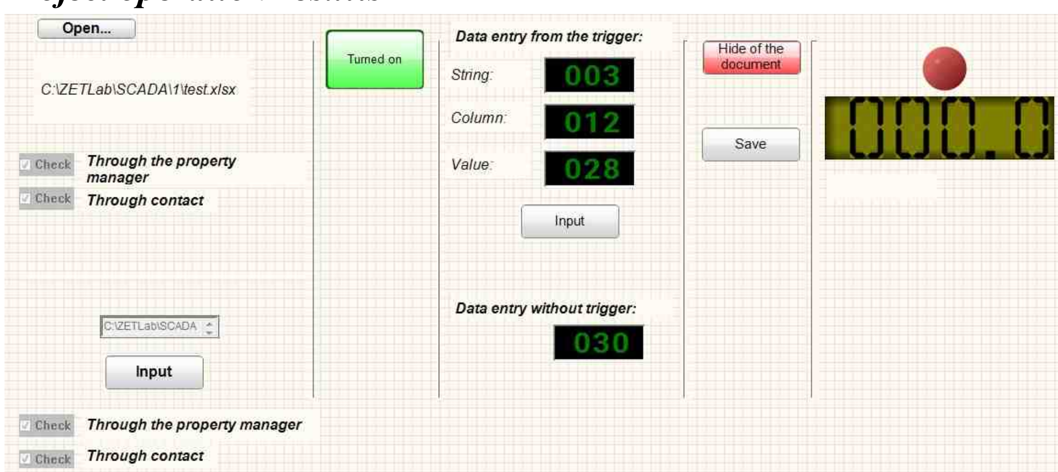
This component is an auxiliary unit rather than and independent component. It is used for operations with Excel-files. Even though the components "Excel document" 1277, "Excel page" 1283 and "Excel cell" 1289 have input and output contacts of the same type, their contacts are used for transfer of different data. "Excel document" 1277 contains a link for Excel-file, while "Excel page" 1283 leads to a particular page. Hence, it is impossible to establish direct connection between "Excel document" 1277 and "Excel cell" 1289.

The component has compatibility with Microsoft Office Excel 2007.

Let us assemble a project in SCADA ZETView. The diagram below shows operations with Excel documents.



Project operation results



View the example in ZetView

Глава 5.Basic operations with arrays

Many operations in SCADA system are to deal with arrays. Array is the most convenient data type for storage of information related to measured signal values in real-time mode or any other numerical or textual data. Many components of SCADA ZETView system use arrays as input parameter. At this stage, we will make an overview of arrays using simple numerals. After having considered operation of virtual instruments, we will study the arrays again in ratio to signals processing in real-time mode.

There are three components classes for operations with arrays:

1 – Arrays

2 – Array-value
3 – Array-array

The name of component class determines its input and output parameters. "Array" class contains components used for arrays forming and recording. "Array-value" includes components used for performing certain operations with input array and producing a value, e.g., array elements sum.
The clause "Array-array" includes components that receive array to the input and produce array at the output contact. Components of "Array", "Array-value" and "Array-array" sections are not depicted in operator mode and can be added to the project only in developer mode.

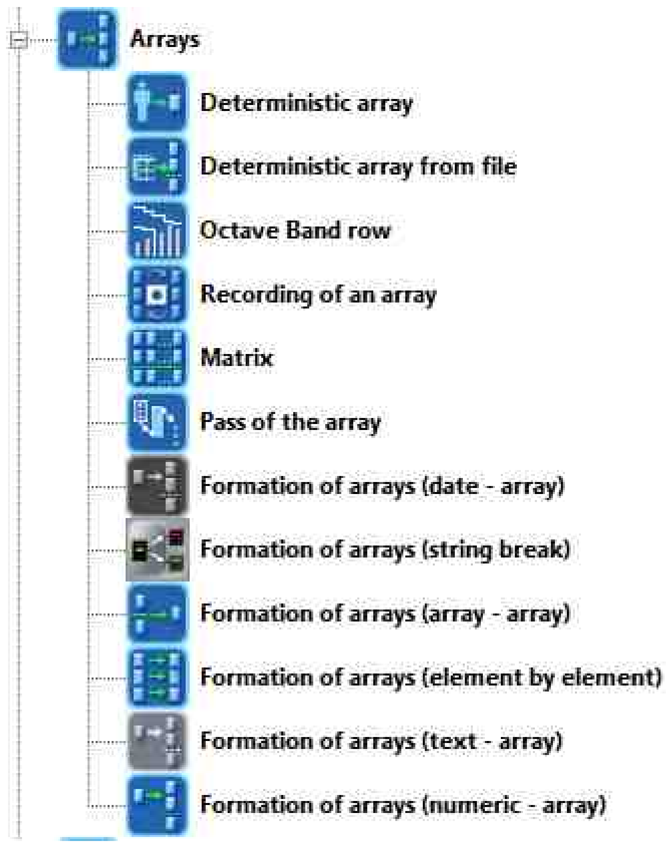


Figure 2.4.1a. Devices window, "Array" section

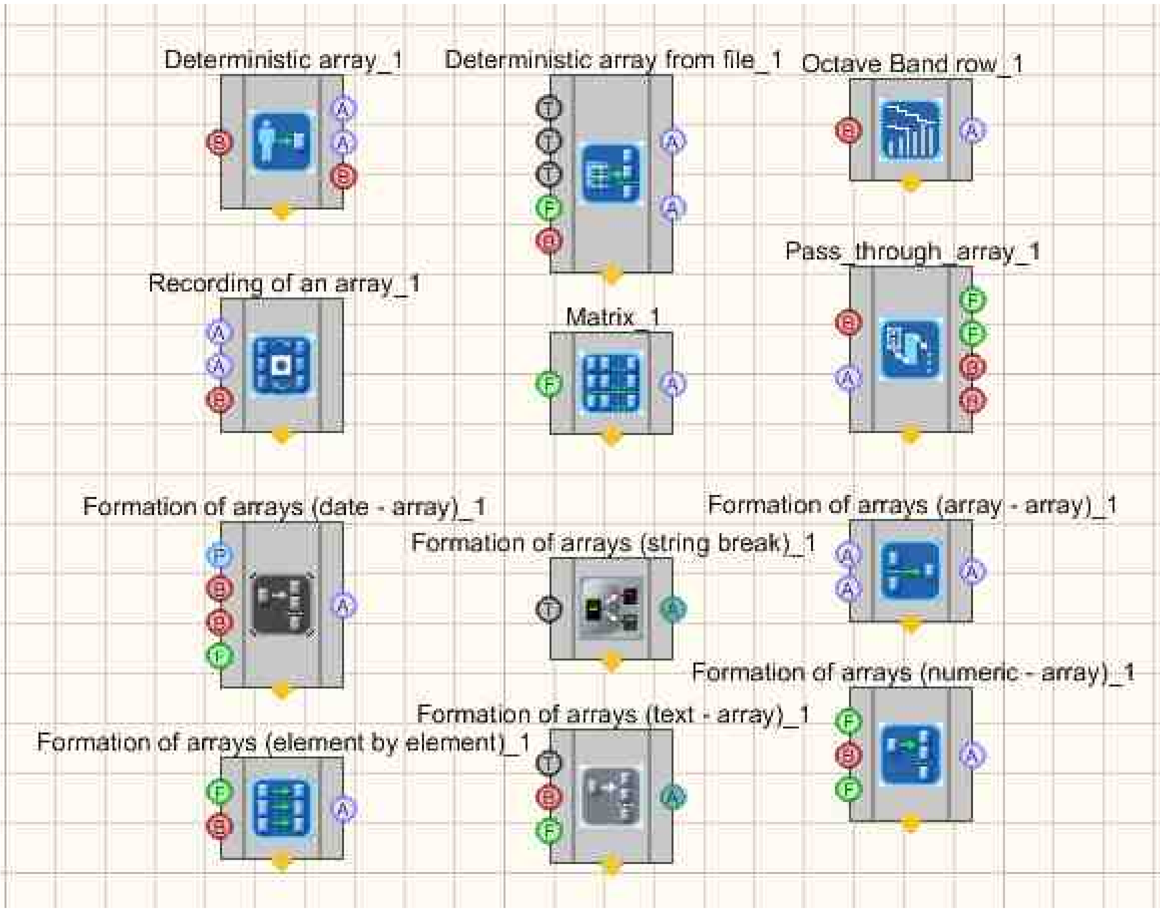


Figure 2.4.1b. Components of "Array section" developer interface

Let us consider "Array" clause in more details. This group contains elements for producing arrays of various types represented by components of "Formation of arrays type". "Formation of arrays (array-numeric)" is also widely used. In many cases, this component is an obligatory one, since in the course of signals measurements the virtual instruments produce a numerical value, while the components used for data processing have array as an input data.

As you can see from the figure, SCADA ZETView has several types of array formation. They are used to convert the data received into array of certain size or a dynamic array, the size of which depends on the amount of elements received. Elements of array formation of this type are widely implemented in projects related to measurements performance.

The following elements are available for the user:
Formation of arrays (element by element) - this component receives numerical data and produces array for the elements.
Formation of arrays (array-numeric) - this component receives numeric values and produces numeric array.

Formation of arrays (text - array) - the component receives strings and produces string array with their further representation in tabular form, since it is the string array that is used for tabular data representation.
Formation of arrays (array - array) - the component produces a single array from two input arrays using various methods: adding, averaging, minimum or maximum value, etc.
Formation of arrays (string break) - the component operation principle is similar to that of "String separator" component. The only difference is that the user does not have to specify particular amount of output contacts for the strings. All the strings received will be recorded to text array.
Formation of arrays (date - array) records current date and time upon impulse receipt at the input. The component is used for measurements indication and formation of reports containing both measuring values and precise time of measurements.
Matrix - used for display of a particular string or column from the previously recorded data. It is used as a multi-unital array.
Deterministic array - used for automated array formation based on formula type (constant, linear, square) and the parameters set.
Deterministic array from file - is used for array reproduction based on the previously recorded signal by means of "Signals recording" component.
Recording of an array - a component used for recording of the array to a file upon receipt of signal at the input of the component.

Spectral analysis of a range of octave bands – a component used for range of octave bands generation.

Pass of the array – the component is used for array elements analysis. E.g., this component may be used for frequency response control project in accordance with the set frequency array.

This component receives array to the input and upon receipt of command, sends array elements to the output.

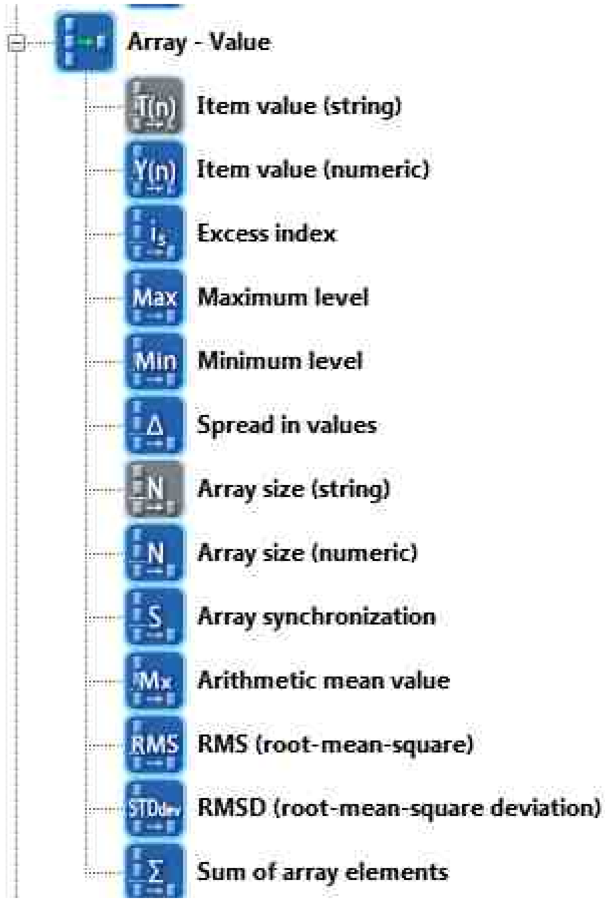


Figure 2.4.2a. Devices window, "Array-value" section.

Figure 2.4.2b. Components of "Array-value" section developer interface

The section "Array-value" contains elements used for the processing of obtained array and producing the required value. Components of the present section allow to set top and bottom limit, calculate element value within the array or a sum of elements.

Components of "Array-value" and "Array-array" section are used for processing of data arrays.

- Formation of arrays is used a get arrays from the values sent to the input of the component.
- Arrays handler is used for arrays processing (sorting, filtration and synchronization of data).

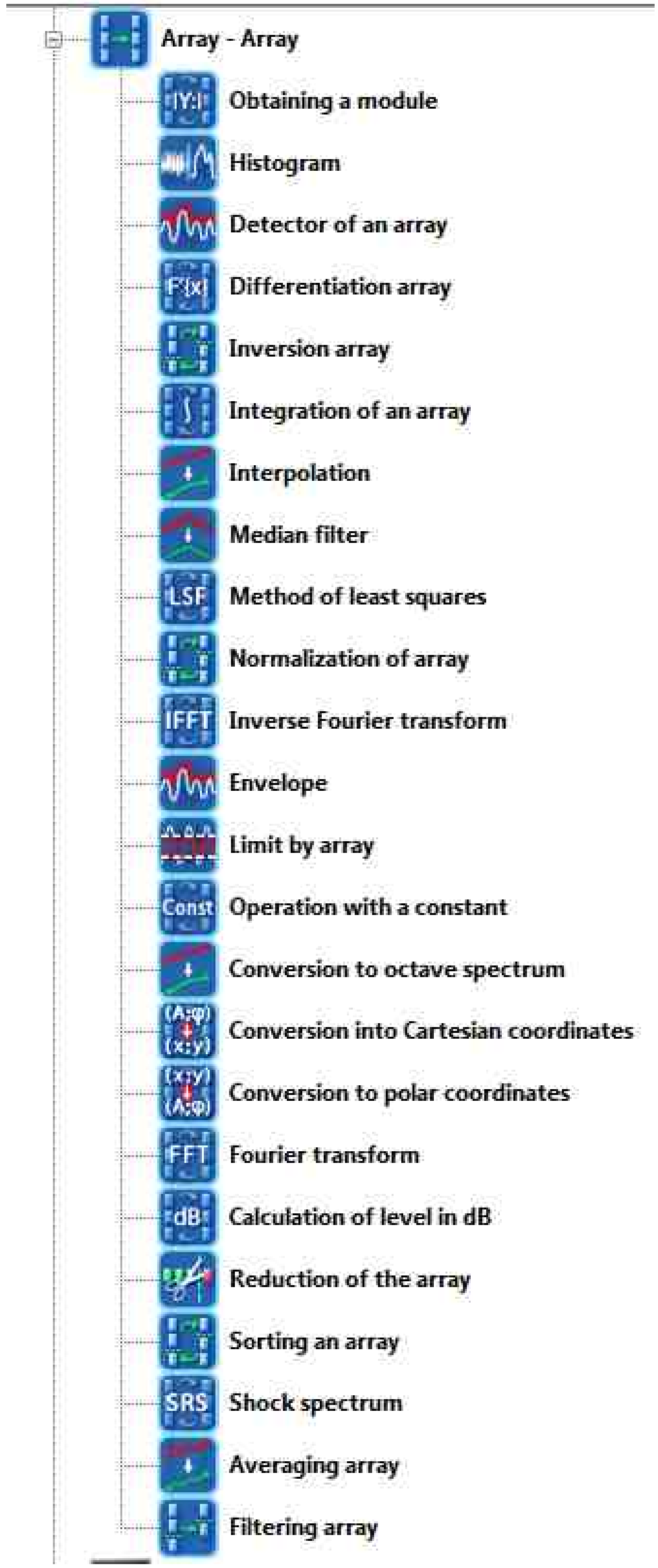


Figure 2.4.3a. Devices window, "Array-array" section.

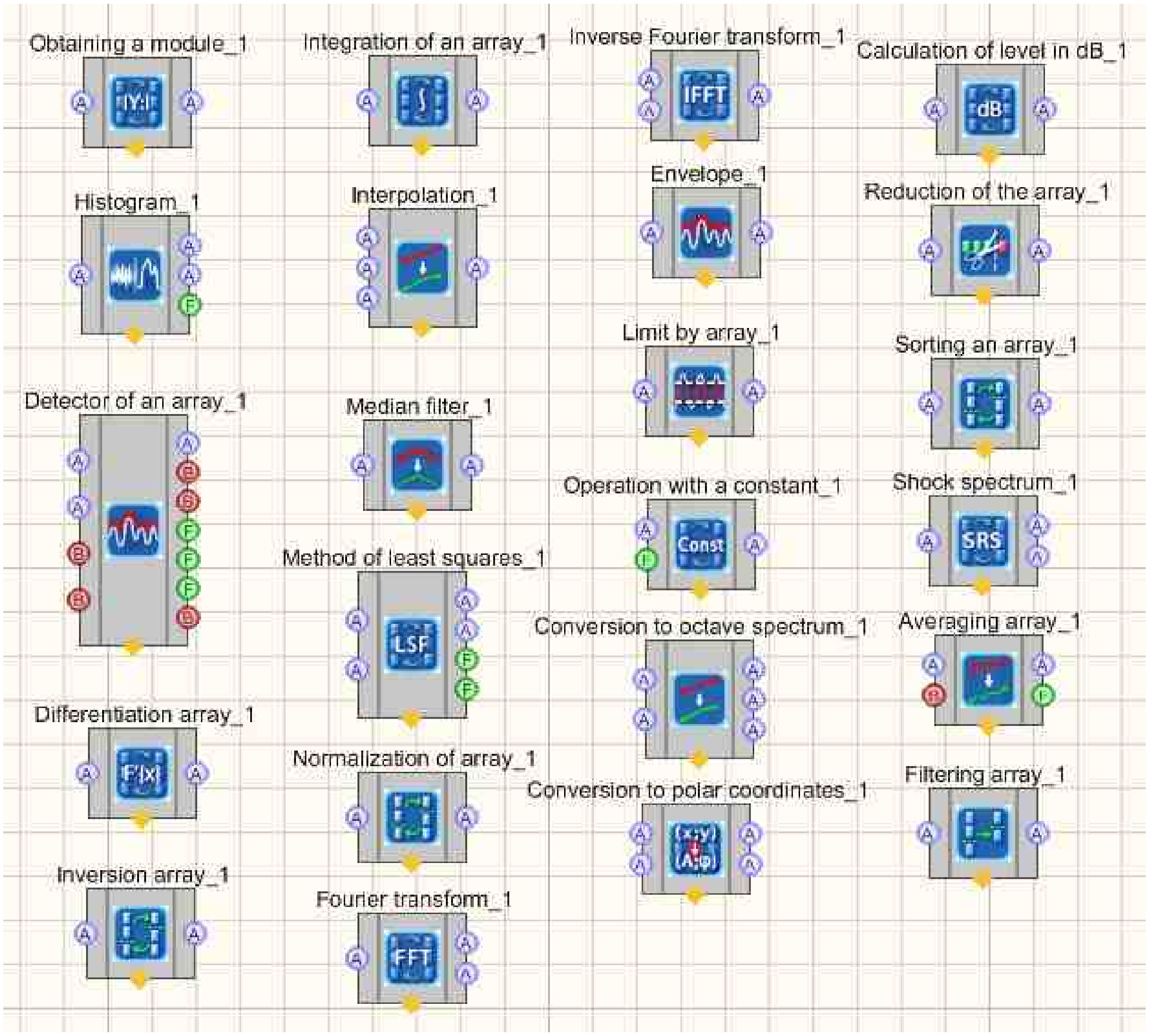


Figure 2.4.3b. "Array-array" section components developer interface

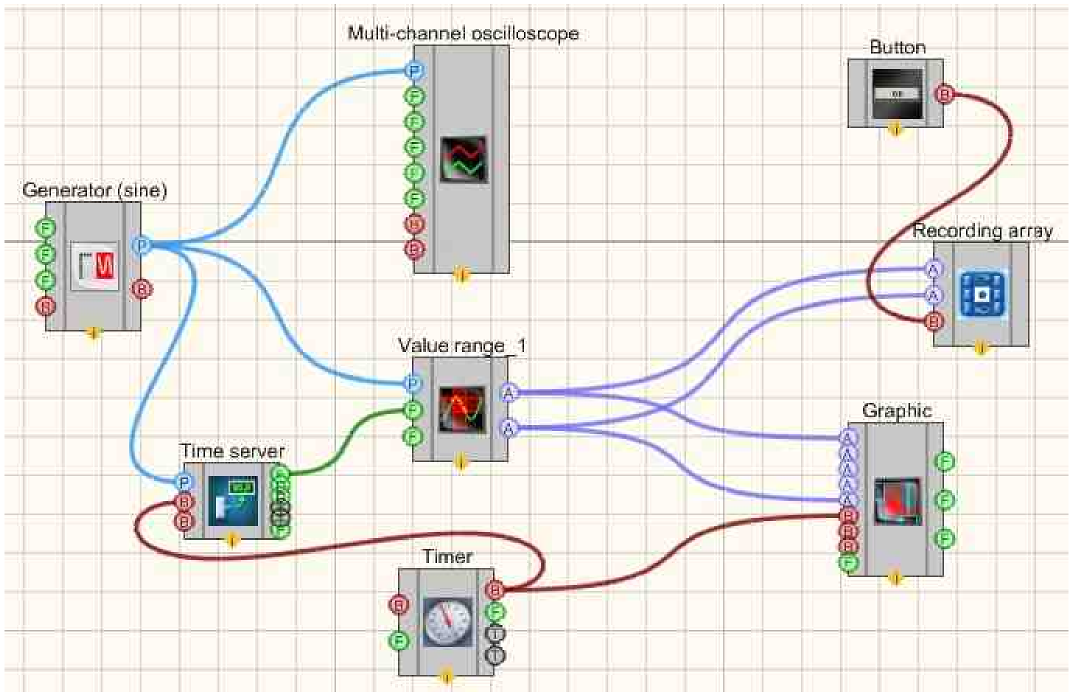
Let us assemble a project in SCADA ZETView system. "Range of values" 594 is used for obtaining instant values of channel signals in the set time range. "Timer" 370 is used for producing synchronous signals for "Server time" 366 and "Graph" 909. (so that the "Graph" 909 and "Range of values" 594 would have identical time parameters). "Sine wave" 530 is used a get test signal and "Multichannel oscillograph" 928 is used to provide visual representation of the signal. "Recording of an array" 773 allows to record the signal to a separate file.

Place the components in the working area as shown in Figure 2.3.2.

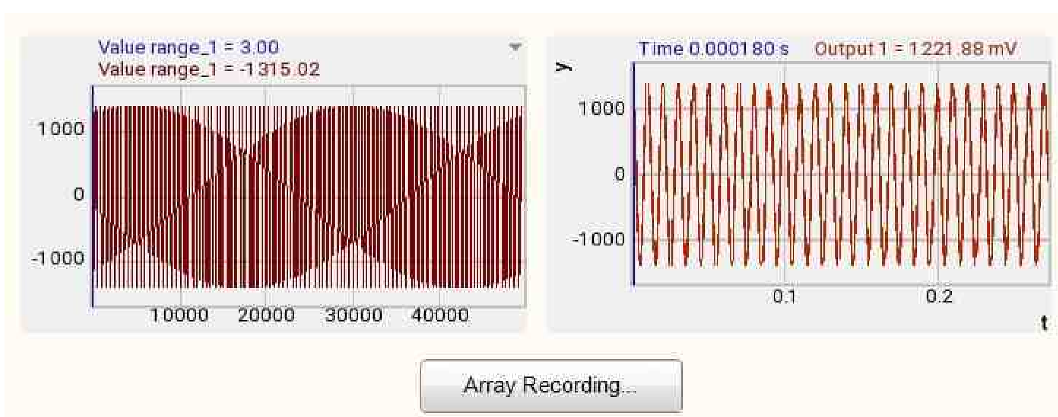
"Generator's DAC" => component "Sine signal;
"Time" => component "Timer";

"Time" => component "Server time"
"Representation" => component "Multichannel oscillograph";
"Measurement of External devices" => component "Range of values";
"Buttons" => component "Button without fixation";
"Arrays" => component "Recording of an array";
"Representation" => component "Graph";

Project in SCADA ZETView



Project operation results



View the example in ZetView

Глава 6.Convertion

As you can see from the previous chapter, SCADA ZETView has various types of data. In most cases, each component receives and produces a particular data type. Sometimes it is necessary to convert data. A typical example of such conversion is representation of digital data in tabular form. This component receives textual data to its input contact, thus, in order to represent it in tabular form, it is necessary to display it in textual form. Components of "Converting" section are normally used for such purposes. This section contains all necessary components for converting all data types in SCADA ZETView system. Components of "Converting" section are not displayed in operator mode, they can only be added to the project in developer mode.

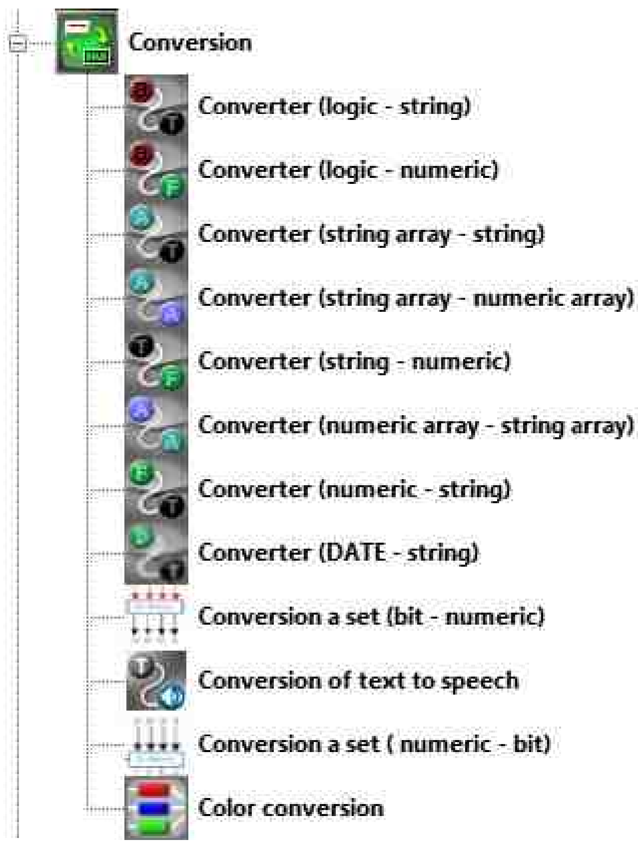


Figure 2.5.1a. Devices window, "Converting" section.

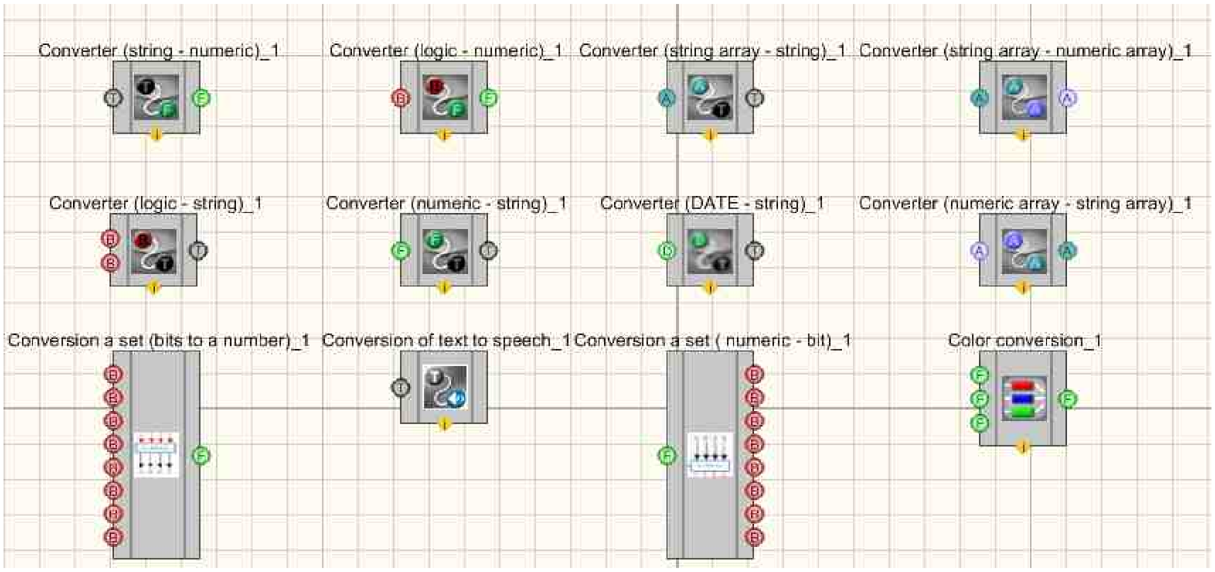
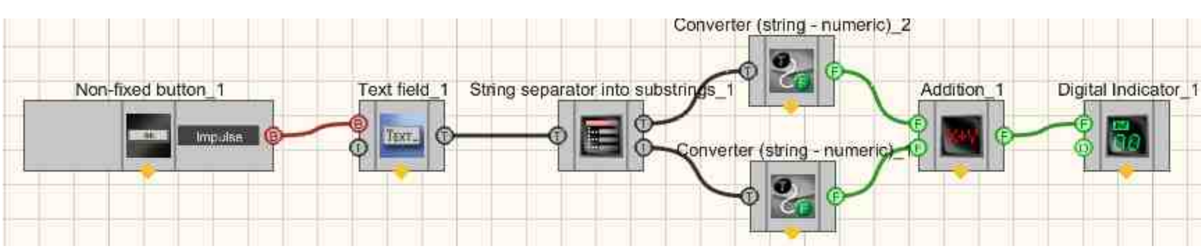


Figure 2.5.1b. "Converting" section components developer interface

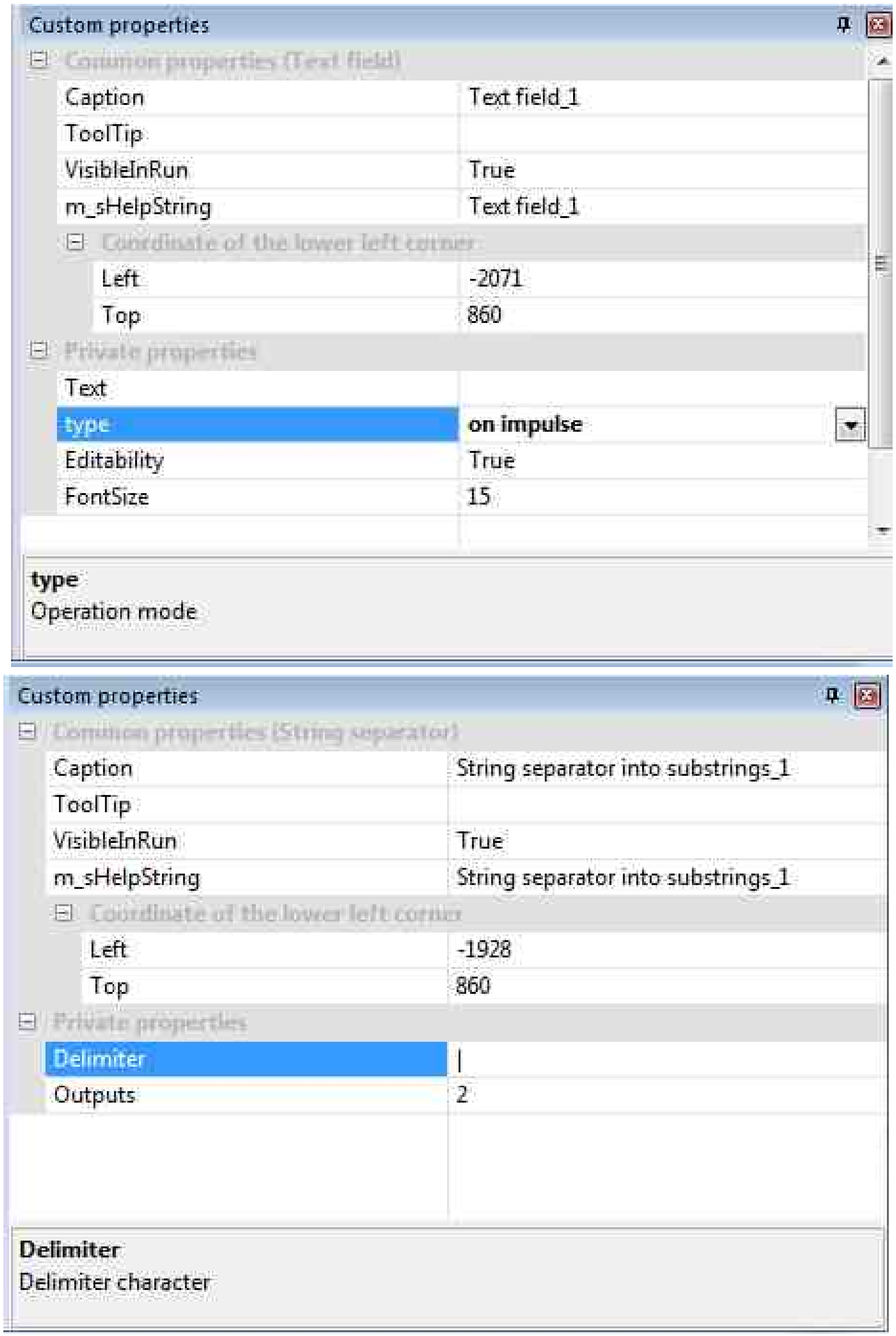
Let us assemble a project in SCADA ZETView. In this project, the user can enter a text string of 1+2 type. To simplify the project, we will use "+" symbol only. Then the numerals from the string are converted from string data type into digital data and then there appears the sum of the entered numerals.

Place the components into the working area as shown in the Figure 2.3.2.

"Text" => component "Text field"
" Text " => component "Separator string"
"Buttons" => component "Non-fixed button"
"Converting" => component "Converter (string - numeric)" – 2 pcs
"Arithmetic" => component "Addition"
"Indication" => component "Digital indicator"



As you can see, the figure shows the components in "Hidden" view (to open/ hide a component, double-click it). Set the properties of "Separator string", select separator symbol "+", so that the component could identify the string entered by the user. Set the "Text field" component, so that it would transfer the corresponding value as the control impulse is sent to its input.



As the control impulse is forwarded to the component input, "Text field" component will send the string entered by the user to the following component "Separator string". In "Separator string" component properties "+" symbol is set as a string separator. Hence, if you enter a string of 1+2 type, two text strings will appear: "1" and "2". It is impossible to add two numerals 1 and 2, since for the program it looks like an attempt to add "A" and "B". The program classifies these symbols as two strings. That is why the symbols are to be converted to digital format. Then we use the component "Converter (string-array)". Outputs of the converters produce two numerals that are further sent to "Adding" component. Then the sum is shown at the digital indicator.

View the example in ZetView

The example shows operation of string-numeral converter. In addition to that, the conversion section includes other converters, allowing to convert a string to numerical, string and digital arrays. In the case if it is impossible to convert the string array into numerical one, or a string to numeral, the converter will skip the corresponding value. The section also contains Bool-string and Bool-numeral components. Bool-numeral produces 0 or 1 depending on the logical level at the input. Bool-string produces one of the two strings, set by the user in component properties (depending on the logical level at the input).

Глава 7.Programmable logic, logic, arithmetic and groups

In SCADA ZETView projects, the control functions are performed by means of logic components and programmable components.

The section "Programmable logics" contains components used for extending the functions of components and implementation of simple algorithms based on VBScript. In some cases, it allows to considerably reduce the amount of components used in the project and to enable components functions. E.g., activate a button with a script, form a message, change color or component’s properties, i.e. to implement functions beyond the scope of those provided for by the system.

The component "Script" is used for implementation of simple algorithms by means of certain commands. The editor has a tabular form with a list of commands representing the sequence of actions. In some cases, it allows to considerably reduce the amount of components used in the project.
"List of variables to the script" allows to create a list of variables to be further used in "Script" component operation depending on the choice of a particular HelpString.

The component "FOR loop" is used for implementation of loop cycle with a counter. This component allows to set certain condition, which the program will fulfill till its completion. For instance if we have to cycle the program 5 or n times, the "FOR loop" component can be used. FOR loop has a distinctive feature – it is normally identified by "i" or "j" letter. Example: for i:= 1 to n do // set i for 1, 2, 3 …n. After first loop of i variable, set 1, then 2,.. n. "to" stands for increase, "downto" – for decrease.

The component "WHILE loop" is used to loop the cycle several times. The loop is stopped as fault result is obtained. Since the fault may be found at once, it is possible that a loop will not be completed.

The component "Programmable logic" section are not displayed in operator mode and can be added to the project only in developer mode.

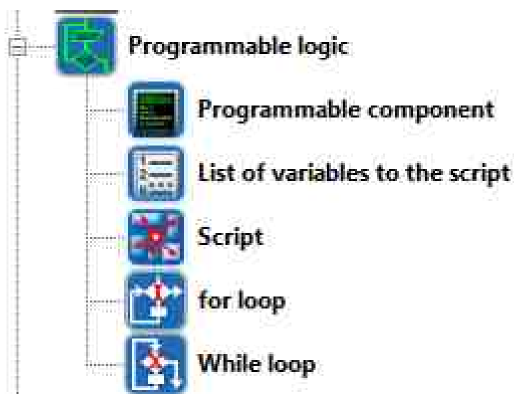


Figure 2.6.1a. Devices window, "Programmable logic" section

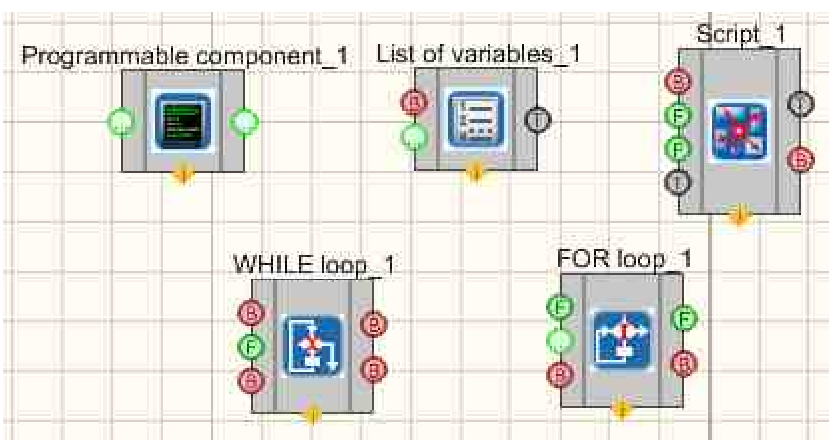


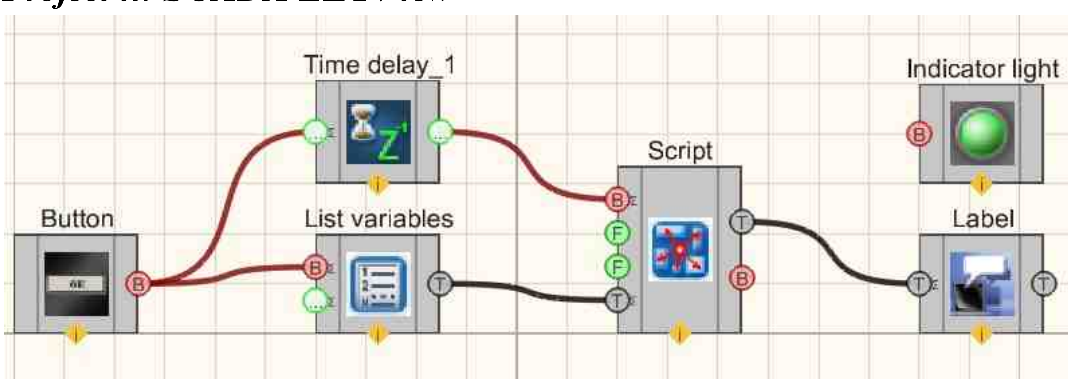
Figure 2.6.1b. Components of "Programmable logic" section developer interface

Let us assemble a project in SCADA ZETView. In this circuit, script is used to enable activation (on/ off) of Light indicator. List of variables is used to create a table with variables values. The push button connected to List of variables provides signal to Script and forwards variables values to Script component. The Tag allows to display text notification of Script performance status. The Button component is used to provide the signal for starting the script.

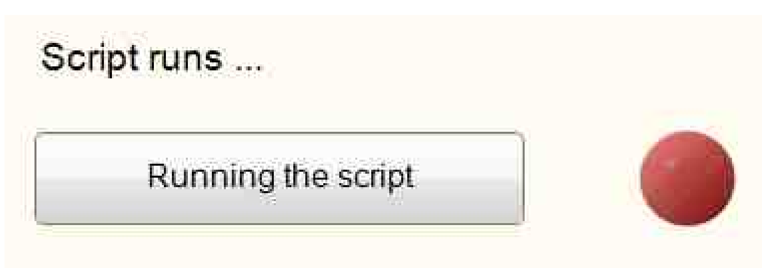
Let us arrange the components in the working area as is shown in Figure 2.6.1c.

"Buttons" => component "Non-fixed button" 2 pcs
"Programmable logic" => component "List of variables".
"Programmable logic" => component "Script"
"Text" => component "Tag"
"Indication" => component "Light indicator"

Project in SCADA ZETView



Project operation results



View the example in ZetView

"Logic" section contains components implementing such logical operations as "Operation AND", "Operation AND (multiple)", "Operation OR", "Operation OR (multiple)", "Overlay mask" (the difference is that these components can receive two ore more input values). One more component is D-trigger, allowing to transfer data when a impulse is received at the input of "Synchro" component. RS-trigger transfers logical variable depending on input combination. Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

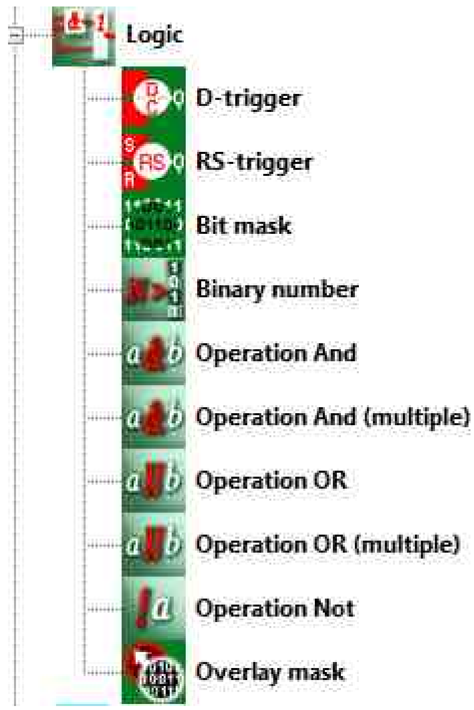


Figure 2.6.2a. Devices window, "Logic"

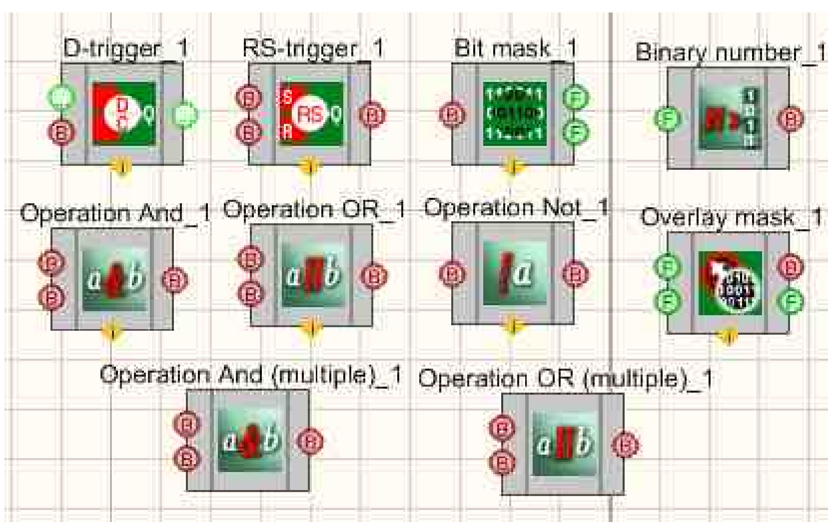


Figure 2.6.2b. Components of "Logic" section developer interface

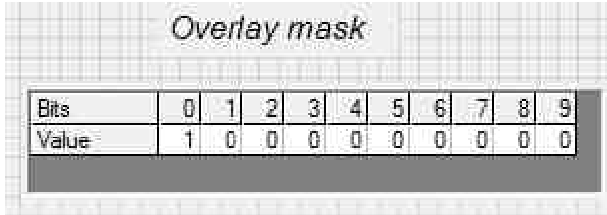


Figure 2.6.2c. Components of "Logic" section operator interface

Components of "Arithmetic" section are used for arithmetic operations with numerical data:

Components of "Arithmetic" section are not displayed in operator mode and can be added to the project only in developer mode.
In operator mode, the following components are available:

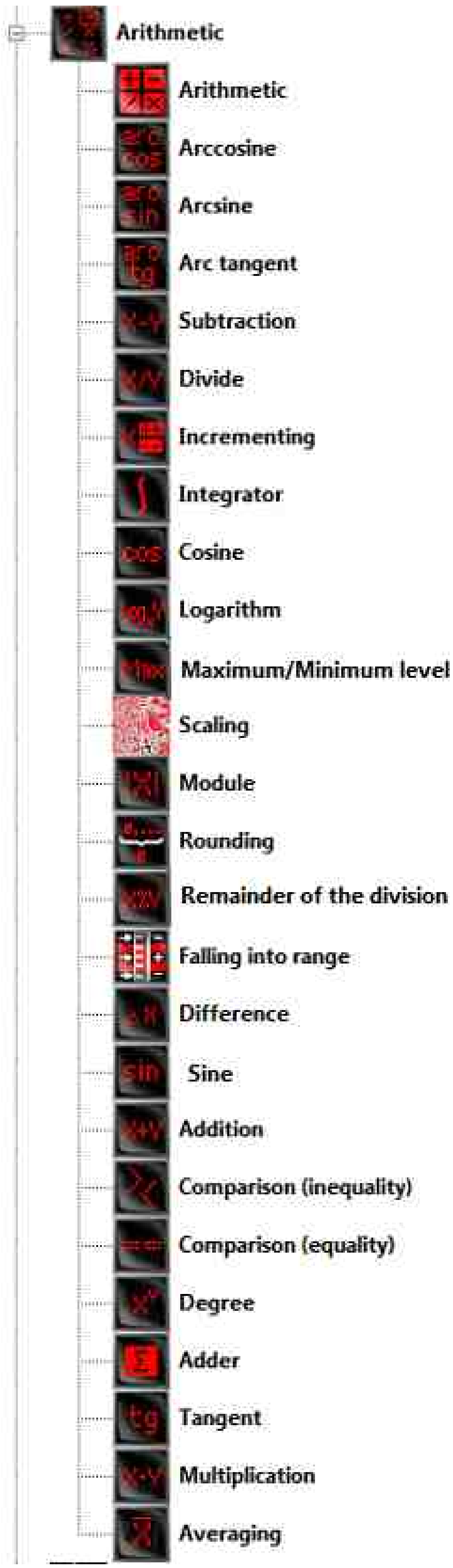

Arithmetic
Arithmetic
Arccosine
Arcsine
Arc tangent
Subtraction
Divide
Incrementing
Integrator
Cosine
Logarithm
Maximum/Minimum level
Scaling
Module
Rounding
Remainder of the division
Falling into range
Difference
Sine
Addition
Comparison (inequality)
Comparison (equality)
Degree
Adder
Tangent
Multiplication
Averaging

Figure 2.6.3a. Devices window, "Arithmetic"

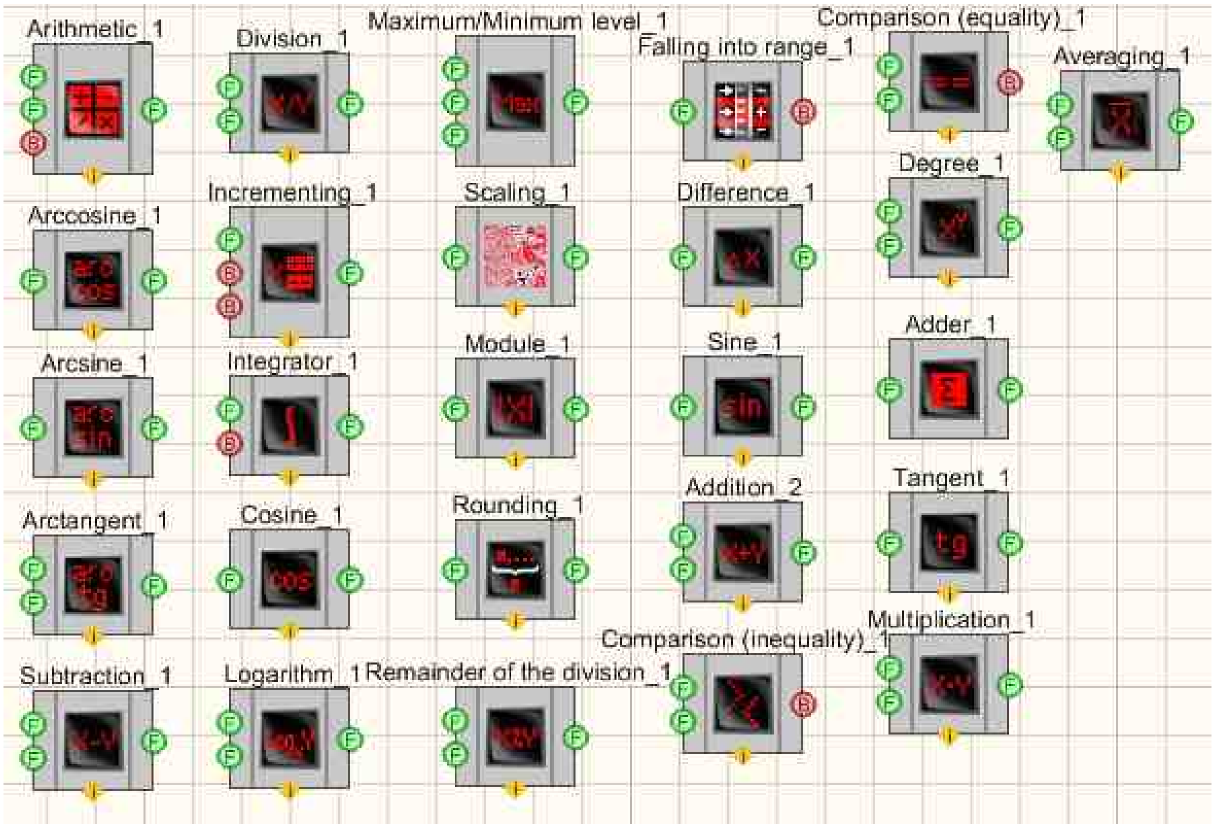


Figure 2.6.3b. Components of "Arithmetic" section, developer interface

Let us assemble a project in SCADA ZETView. This project shows the operation of the "Addition" 327 performs adding of all input values set by "Select" 428. "Timer" 370 provides synchronization impulses to the input of "D-triggers" 732, thus enabling sequential sending of numerical values to the "Addition" 327 (the adding of signals is only possible in the case if signal values at the output contacts of the adder are changed simultaneously). The "Digital indicator" 682 is used for graphical representation of data.

Let us arrange the components in the working area as is shown in figure 2.6.3c.

"Time" => component "Timer"
"Select" => component "Selector No.1" – 3 pcs.
"Logic" => component "D-trigger" – 3 pcs.
"Arithmetic" => component "Addition"
"Indication" => component "Digital indicator"

As you can see from the figure, the components are displayed in "Hidden" view (to open / hide the component, double-click it). Set the properties of "Addition" component: Input=3 (so that the component would have 3 output contacts).

Project in SCADA ZETView

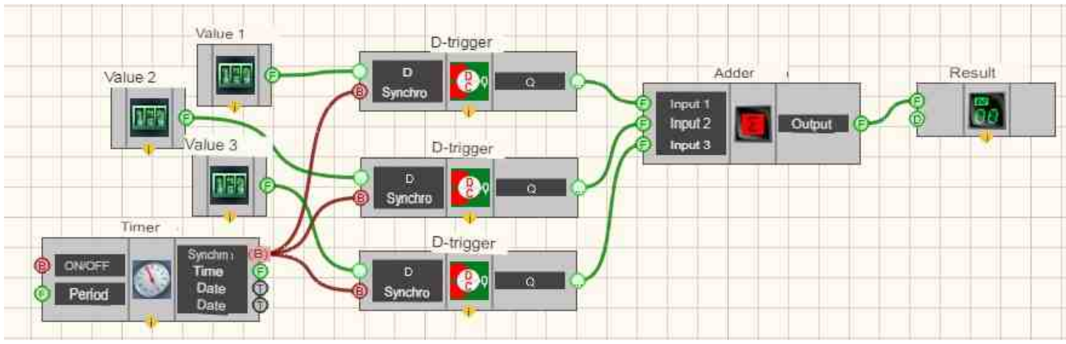


Figure 2.6.3c. Display of components in the working area developer interface

Project operation results

View the example in ZetView

"Groups" section contains components are used to unite the components into groups. Components of "Groups" section are displayed only in operator mode and cannot be added to the project in developer mode.

Note:

In other words, the component is represented by a simple block. It is important to place it into the background or place the components inside of the block in the foreground.

Otherwise the block will cover the components, since the components are placed onto each other in the order of being added to the working area.

The component "Combine a group" is used to unite components groups into logical blocks in operator mode (the frame type is selected by the user).

The component "Windows Panel Style" is used as a comment field. In allows to make text notes for the operator.

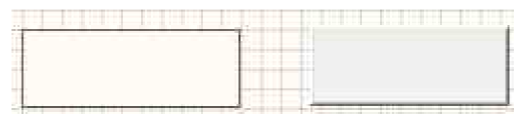
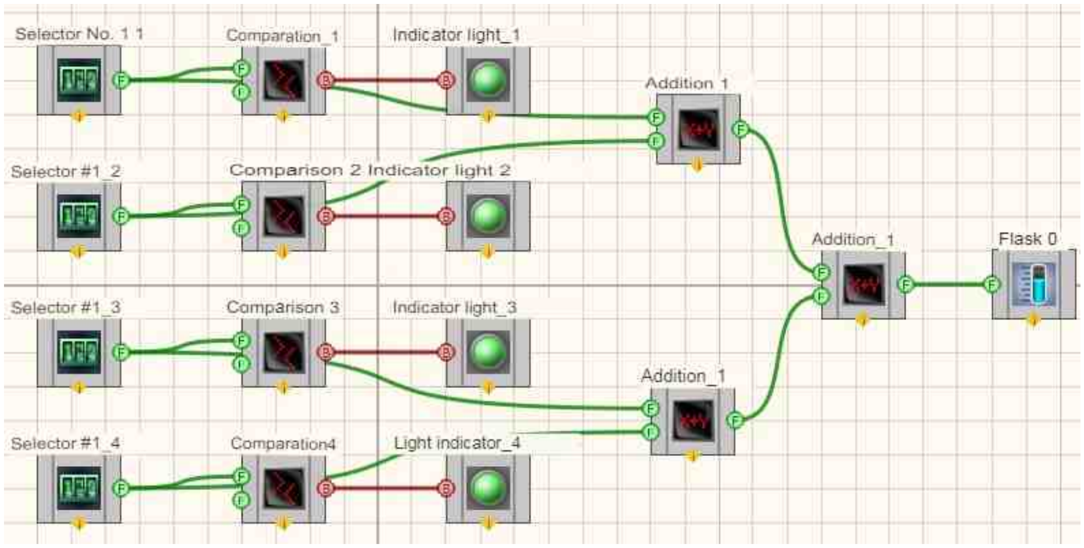


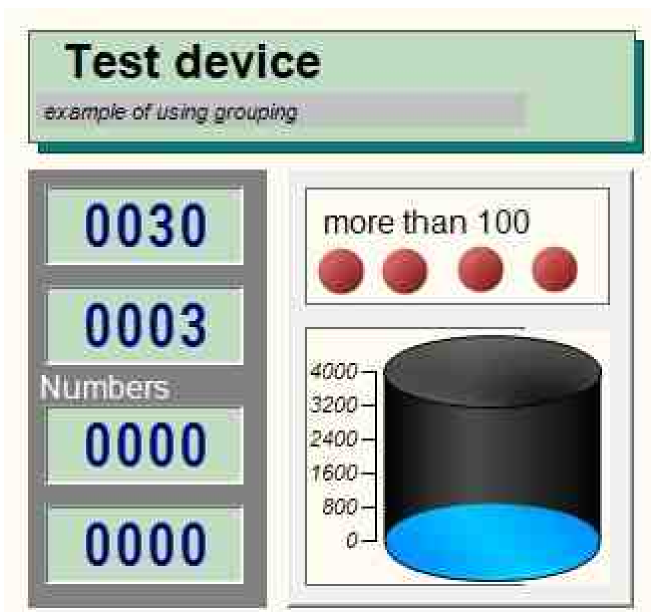
Figure 2.6.4a. Devices window, "Groups" section Figure 2.6.4b. Components of "Groups" section operator interface

Let us arrange the components in the working area as it is shown in figure 2.6.4b. In this diagram you can see the input values of four indicators, their sum, the result displayed at the bulb, light indicators (in the case if value 100 is exceeded). All the components used are classified into groups for user convenience.

"Time" => component "Selector No. 1" 4 pcs.
"Arithmetic" => component "Comparison (inequality)" - 4 pcs.
"Indication" => component "Light indicator" 4 pcs.
"Arithmetic " => component "Addition"" 3 pcs.
"Indication " => component "Bulb"



Project operation results



View the example in ZetView

Глава 8.Signal analysis and measurement (external devices)

The components FFT (Fast Fourier Transform) Spectrum Analysis, Cross-Spectrum CPB (constant percentage bandwidth) analysis, Cross-Correlation Analysis, Cross-Spectrum FFT Analysis, Cross-Spectrum FFT Analysis, Histogram, Modal analysis, STA/LTA detector, Harmonic Distortion Analysis allow to start corresponding programs from ZETLAB scope, to set corresponding properties and to obtain results of programs operation.

The component High-frequency oscilloscope is used for signals processing of ZET 302 oscillograph. It starts ZETScope program and sets its properties.

Thus, the programs from "Signal analysis" scope fully represent properties of the corresponding controlled programs: FFT Spectrum Analysis, Cross-Spectrum CPB, Cross-Correlation Analysis, Cross-Spectrum Cross-Spectrum FFT Analysis, High-frequency oscilloscope(ZETScope.), Histogram, Damping rate of the oscillations, Earthquake detector, STA/LTA Detector, Intensity of the earthquake on the MSK-64, Modal analysis, Calculation of impedance, Event histogram, Super-resolution Spectrum, Spectrum CPB Analysis, Harmonic Distortion Analysis.

The components of "Signal analysis" section are not displayed in operator mode and can be added to the project only in developer mode.

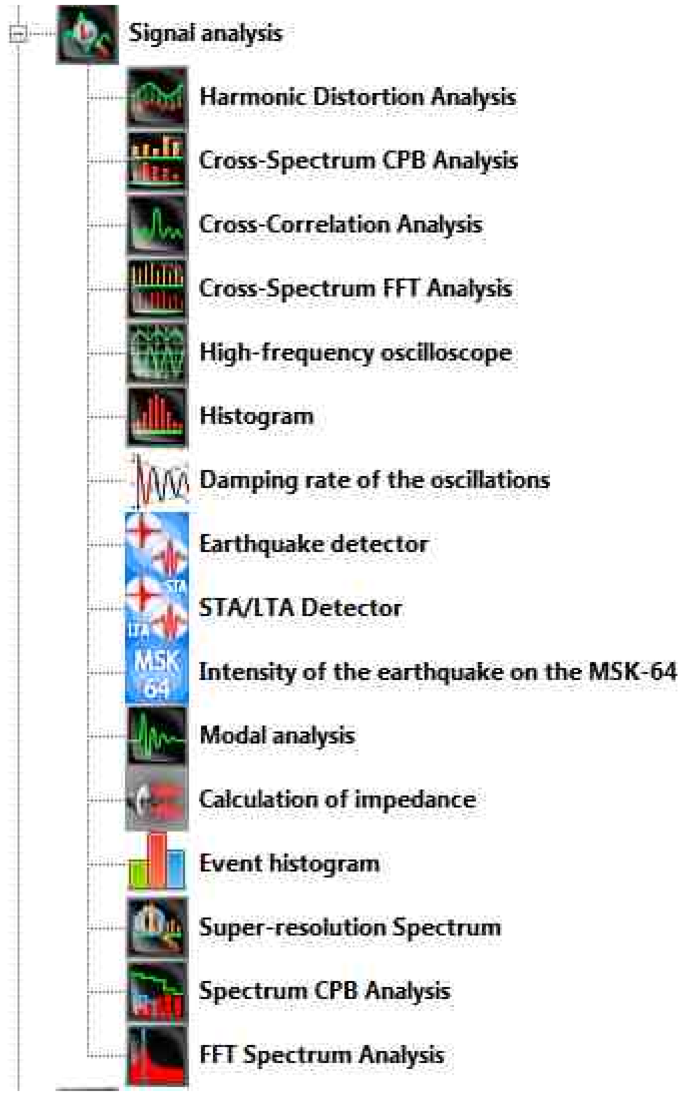


Figure 2.7.1a. Devices window, section "Signal analysis"

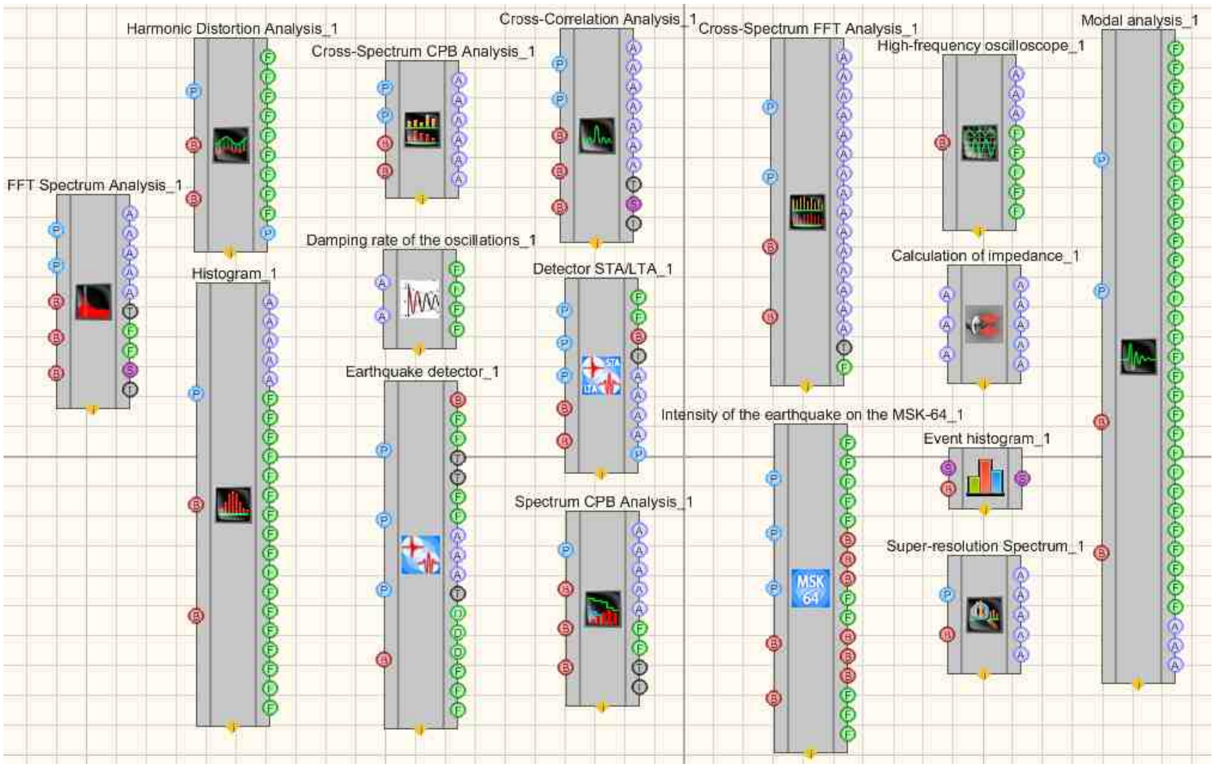
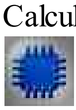


Figure 2.7.1b. Components of "Signal analysis" section developer interface

Sometimes in the course of signal analysis it is necessary to perform data accumulation and then calculate maximum, minimum and average values.

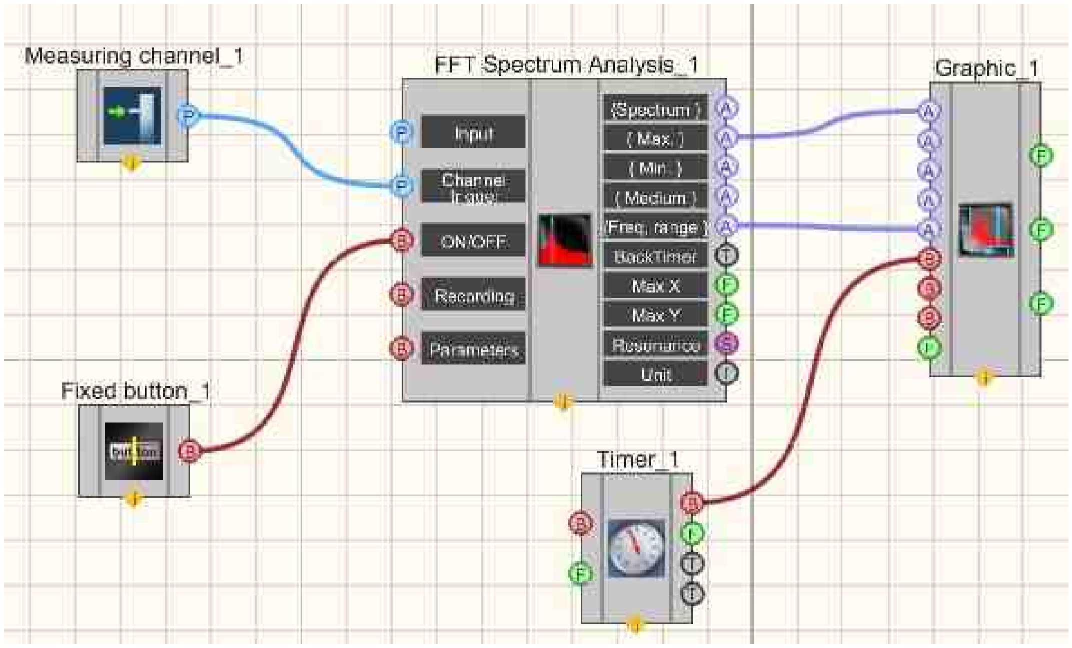
Examples of obtaining maximum, minimum and average spectra, correlograms, etc. are shown in the project (as an example of FFT Spectrum Analysis).

Calculating the maximum values

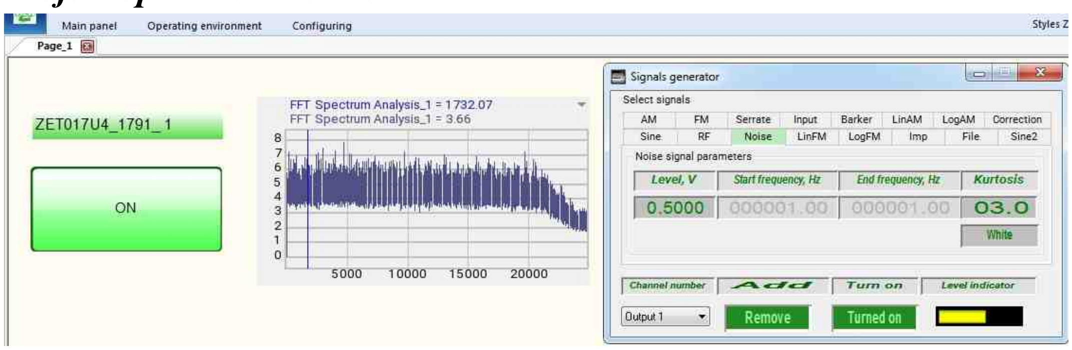


Example No. 1 calculating the maximum values:

Project in SCADA ZETView



Project operation results



View the example in ZetView

Example No. 2 - calculating the minimum values:

View the example in ZetView

Example No. 3 - calculating the average values:

View the example in ZetView

Components of "Measurements (External devices)" section are used for measurements of various parameters of signals received at the input modules of ADC-DAC channels, Narrow-band spectrum program, strain-gauge stations and virtual channels created by such programs as ZETFormula, Signal generator, Filtration by channels, etc:

Components of "Measurements (External devices)" are not displayed in operator module and can only be added to the project in developer mode.

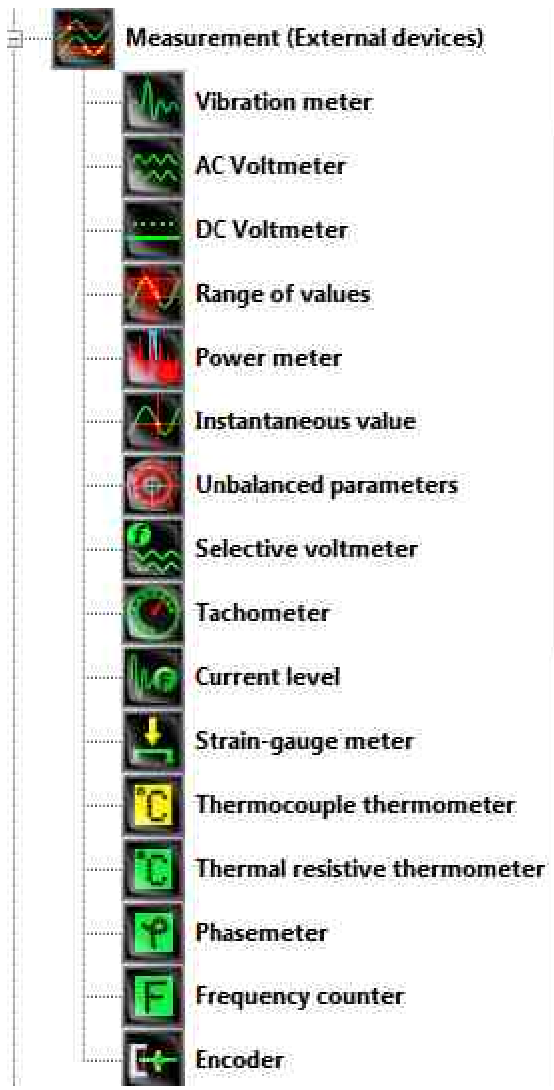


Figure 2.7.2a. Devices window, "Measurements (External devices)" window.

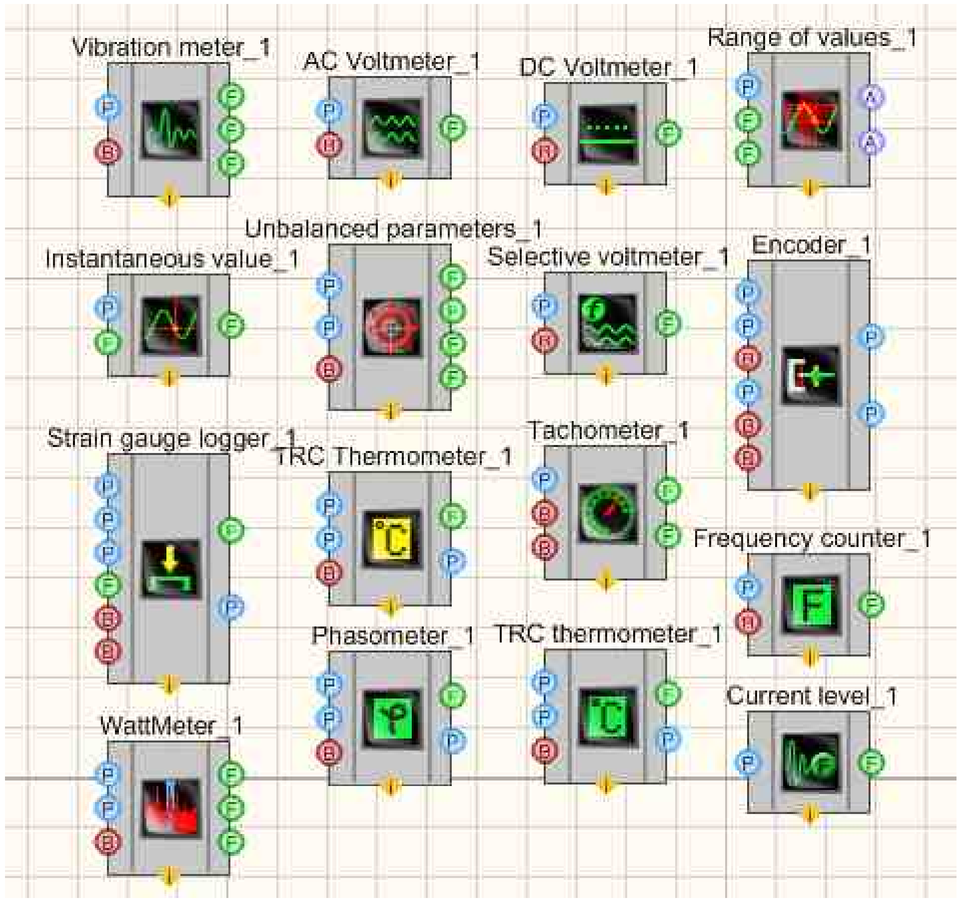


Figure 2.7.2b. "Measurements (External devices)" section components developer interface

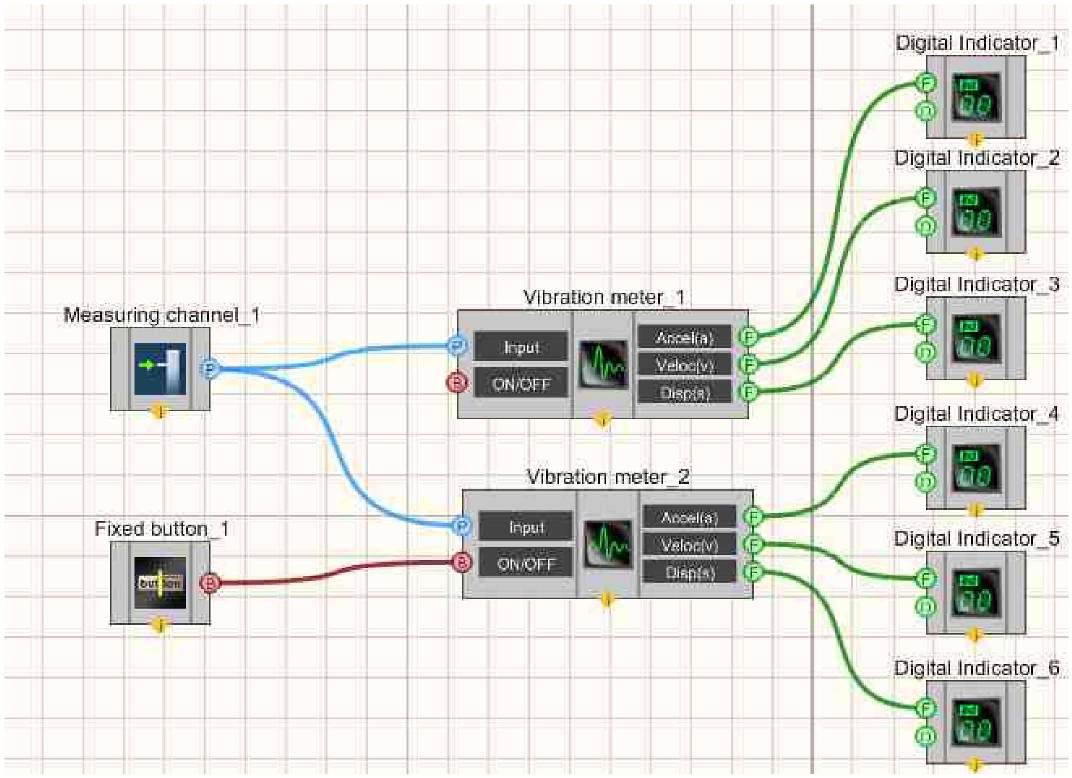
Attention!

Some of the measurement programs (DC/AC voltmeter, vibration meter and some others) can operate in multichannel mode in the case if they are loaded by means of UNIT interface. In this mode the program can perform measurements by several channels simultaneously (max amount of channels is 250). Each channel of the program may have individual parameters (averaging, channel sampling frequency, vibrometer band filter, etc.). Measurements results are available in parent program via UNIT interface.

The vibration meter operates only with accelerometers channels; it identifies them by measurements unit: "g" or "m/sI".

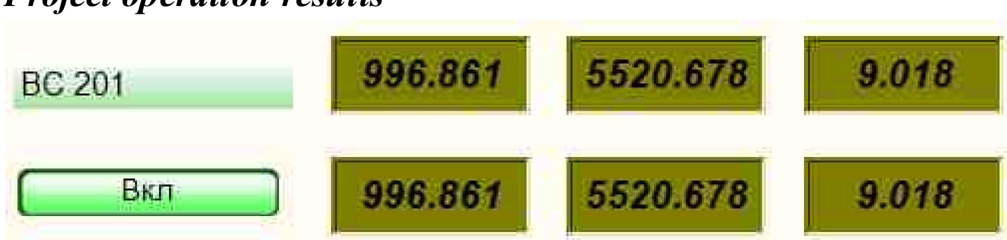
Example

Project in SCADA ZETView



In this project, the multi-channel component "Vibration meter" 572 is used for measurements of signal received from "Measuring channel" 375. "Fixed button" 691 is used to switch on/off the vibration meters. "Liquid crystal indicators" 655 are used for graphical display of the task.

Project operation results



View the example in ZetView

Глава 9.Generators DAC and ADC input

The components of "Generators DAC) section is designed to generate signals at the output channels of ADC-DAC modules, FFT Spectrum Analysis, Strain-gauge station as well as to create virtual channels necessary for projects operation.

The components listed below are used to start and control "Signal generator" program from the scope of ZETLab software (each of the components corresponds to a tab of the program):

- **Amplitude modulation** - allows to generate a signal in accordance with the applicable law of frequency modulation (the variable parameter of the signal is represented by its amplitude value),
- **Frequency modulation** - allows to generate a signal in accordance with the applicable law of frequency modulation. The informational signal controls the frequency of reference oscillation (the difference from AM is that the amplitude value of this signal remains constant),
- **Sine signal** - allows to produce harmonic signal in accordance with the set parameters: amplitude, frequency and constant component displacement. For component's operation it is necessary to have a ZET-device with DAC connected to the PC,
- **Signal serrated** - allows to generate saw-tooth signal of a certain type in accordance with the set parameters (amplitude, frequency and constant component displacement),
- **Barker** - allows to generate phase-modulated signals
- **File** - generates a signal based on the parameters from a file
- **Linear amplitude modulation** - allows to generate a signal in accordance with the applicable law of amplitude-modulated (oscillations frequency changes in accordance with linear law within a certain range),
- **Logarithmic amplitude modulation** - allows to generate a signal in accordance with special law of amplitude-modulated, (oscillations frequency changes in accordance with logarithmic law within a certain range),
- **Linear frequency modulation** - allows to generate a signal in accordance with the applicable law of frequency modulation (oscillations frequency changes in accordance with linear law within a certain range),
- **Logarithmic frequency modulation** - allows to generate a signal in accordance with special law of frequency modulation, (oscillations frequency changes in accordance with logarithmic law within a certain range),
- **Impulse** - allows to generate rectangular impulses with the set frequency, cycling ratio and constant component,
- **RF impulse signal** - allows to generate RF impulse signal (limited sine signal) with a set filling impulse, level. Repetition frequency and duration,
- **Noise** - allows to generate stationary noise with spectral components evenly distributed along the frequency range.
- **Input channel** - allows to select one of the channels of ZET-device connected. It send the ID of the selected component to the next comportment, which, in its turn, receives the signals.
- **Superpostion of generators** - allows to create a signal based on a set of input signals (saw, impulse, etc.). It is used a get new signals of the required shape.

- **Synchronous generator** - allows to generate a signal with any amount of channels (up to 20) with the set frequency of periodical signal. For component operation, it is necessary to have a ZET-device with DAC connected to the PC.

The component Superposition of generators is used a get a signal of complex shape (components from the list above are connected to it). The resulting signal is produced at the DAC output of the device connected.

The component Synchronous generator is used for start and control of corresponding program from ZETLab scope. It is used a get synchronized signals at the DAC outputs of the devices connected.

The components of "Generator" section are not displayed in operator mode and can only be added to the project in operator mode.

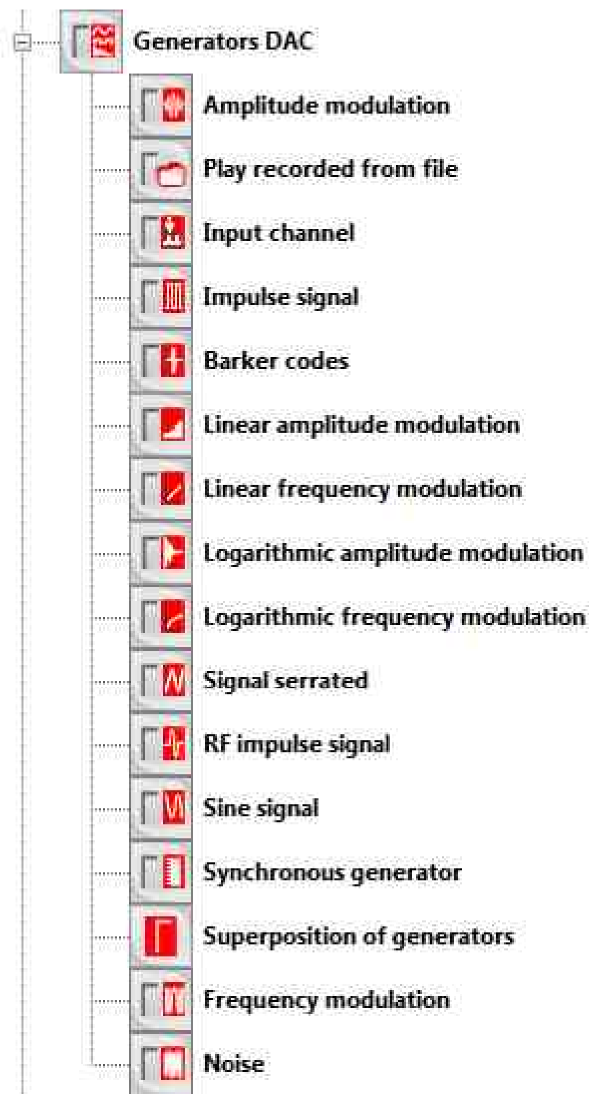


Figure 2.8.1a. Devices window "Generators DAC"

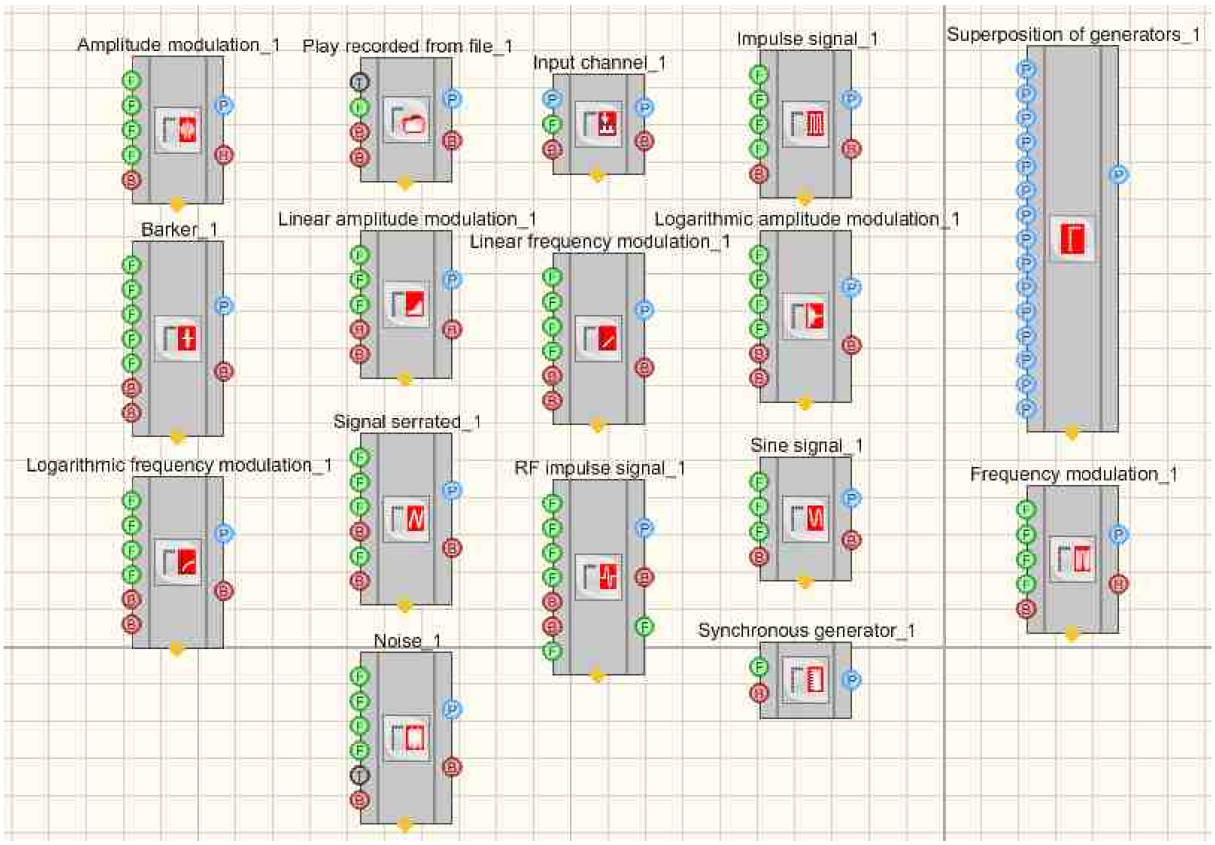


Figure 2.8.1b. Components of "Generators" section developer interface

The components of "Input" section are used for operations with ZET-server channels in SCADA-system ZETView. The list of channels is formed based on real measurements channels (active ADC channels of the devices connected to PC) and virtual measurements channels. Virtual channels are created by generators (signals, sent from DAC output of the devices connected) and by programs for signals processing (signals filtration and processing in accordance with a particular algorithm). In the course of "Play recorded signals" program operation the list of channels consists of previously recorded and virtual channels, recorded in the course of data processing:

- **Virtual channel** - component. Special channel used for emulation in ZETView environment. It is an analog of Input signal module (the only difference is that Input signal module receives data from real device, while Virtual channel displays a signal created by PC).
- **Measuring channel** - component. Allows to accept a signal from the output of an external device. It has only one parameter – the selected channel that is used for signal transfer. It is impossible to receive data from external devices without this component.

- **Device information** - component. Allows you to receive information on the connected channel information about the number of the device.
- **Channel of the formula** - component. Allows to produce signals of various shape as well as to perform mathematic and arithmetic operations with them.
- **Signal quality** - component. Is used for operation with the program SynchronizationControl.exe (Synchronization control), which allows to control quality of data, power supply and synchronization by channels of the devices having the option of synchronization, power supply and data control.
- **Inquiry channel parameters** - component. Allows to obtain signal parameters from the selected channel upon external impact or periodically;
- **Channels list** - component. The component "Channels sorting" is used for sorting of the signals. It is possible to set the list of channels for sorting in properties settings window or to select it in the course of project development. View and parameters of sorting are set in components parameters.
- **Commit changes** - component. It allows to save the properties changes of devices, system channels (corresponding data is obtained from ZETServer component).

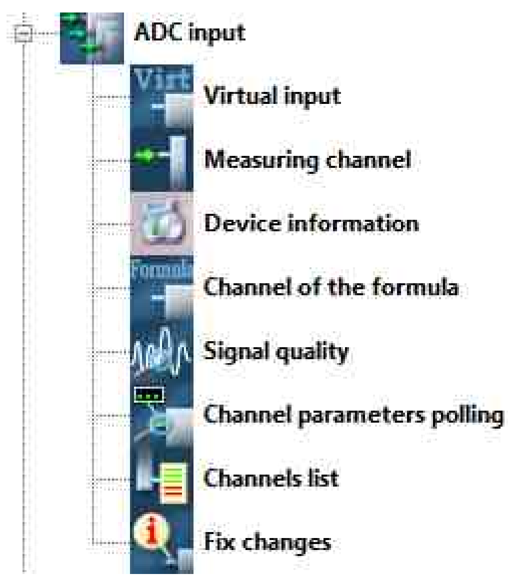


Figure 2.8.2a. Devices window, clause "ADC Input".

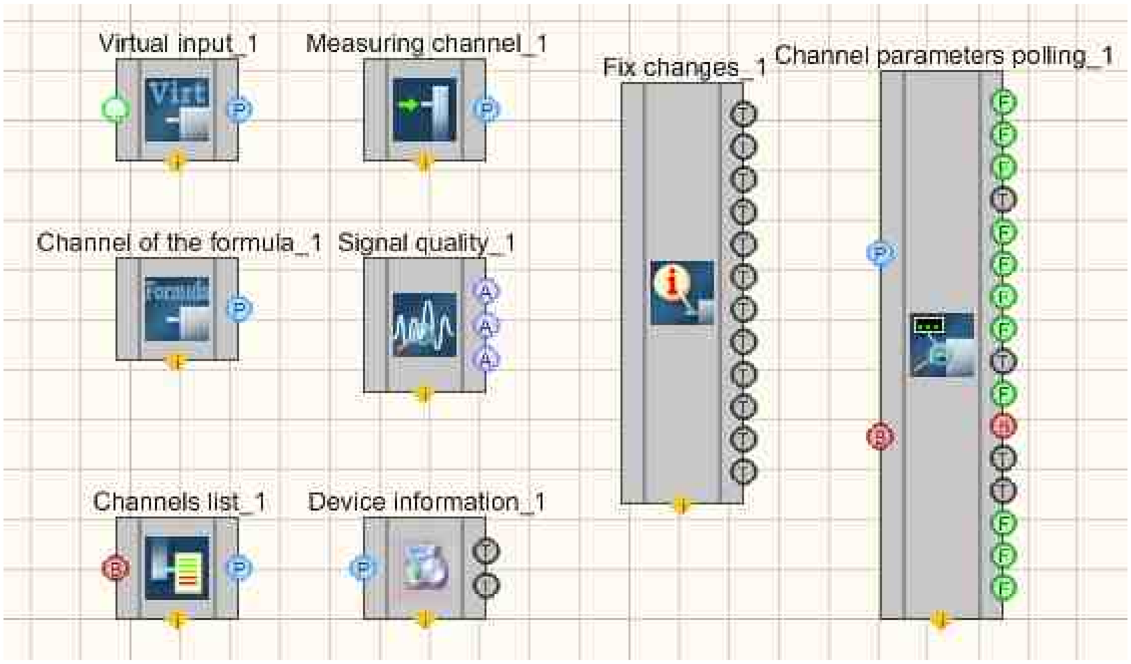


Figure 2.8.2b. Components of "ADC Input"section developer interface

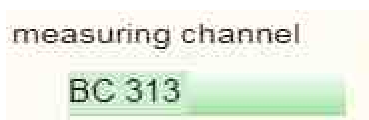
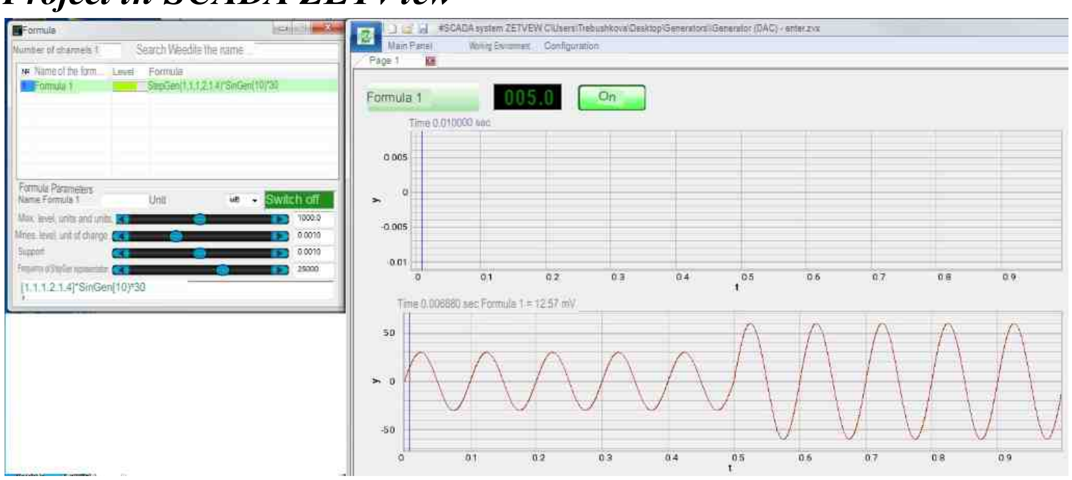


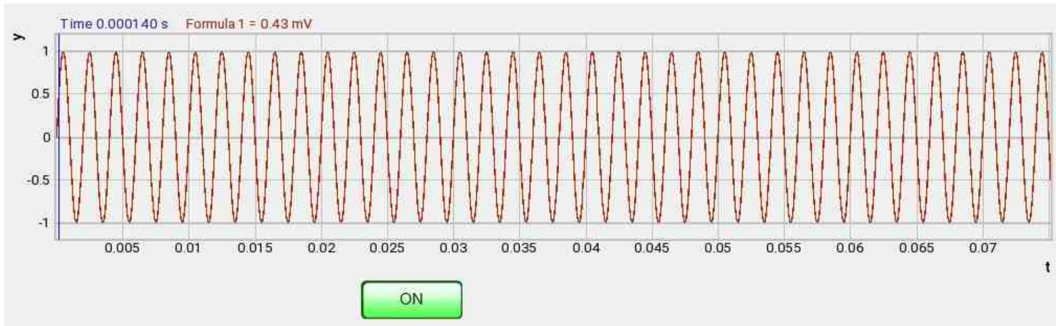
Figure 2.8.3c. Components of "Input (ADC)"section operator interface

Example

Project in SCADA ZETView



Project operation results



View the example in ZetView

Глава 10.External devices and devices ZET

The clause "**External devices**" contains the components enabling control of external devices connected to PC

- **Acoustic modem** - is used for conversion of digital signals into sound signals and vice versa. The device allows to send and receive sound frequency signals via phone lines, which allows to establish connection between PC and terminal by means of a modem or a cell-phone,
- **Scales BP05mc** – used as scales.
- **Generator DS360** – is used for reproduction of electromagnetic signals. DS360 allows to reproduce signal of a particular shape, which is often necessary in the course of research and settings of various electronic systems. The device allows to produce certain types of impulses with a preset parameters. It also enables easy testing of computing devices.
- **Power supply LPS-305** – is used for control of programmable power supply source LPS-305;
- **Power supply MCA 750-3000** – is used for control of programmable power supply source MCA 750-3000;
- **Power supply PPE-3323** - is used for control of programmable power supply source PPE-3323;
- **Power supply PSH-3610** - is used for control of programmable power supply source PSH-3610;
- **Power supply PSM-2010** - is used for control of programmable power supply source PSH-2010;
- **Power supply source B5-85/1** - is used for control of programmable power supply source B5_85/1;
- **Portable pressure calibrator Yokogawa-CA150** - The component is used for sensors power supply circuits testing, field instruments, digital multimeters, records and various measuring instruments calibration. Can be used both for field instruments maintenance and settings of laboratory instruments. Multimeter calibrator CA150 has a light weight (1kg) and compact units.

- **Multimeter Agilent 34401a** - is used for constant and alternating current parameters measurements, measurements of alternating current frequency, circuit section impedance, circuits and p-n transitions control;
- **Nanovoltmeter Keithley 2182a** - is used for super-low voltage measurements, has a low level of intrinsic noise and enables high speed of measurements performance.
- **Control of external devices** is used for control external devices.

It is possible to perform external control of the devices in SCADA-system ZETView as well as to create precise automated control and settings systems for electrical units and components.

Components of "External devices" section are not depicted in operator mode, they can only be added to the SCADA-project in operator node

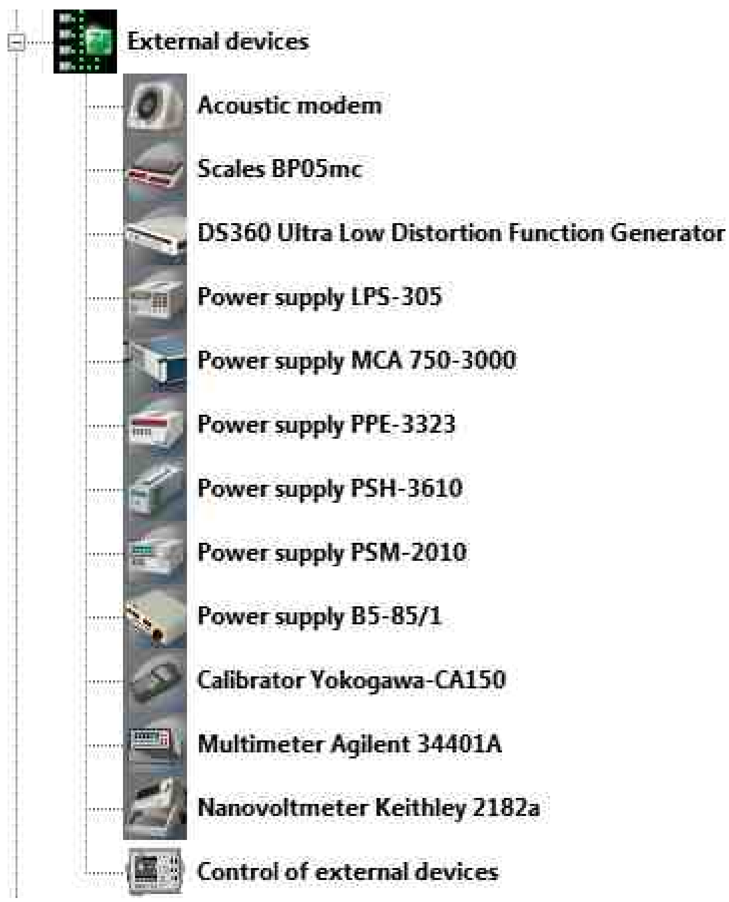


Figure 2.9.1a. Devices window, "External devices"

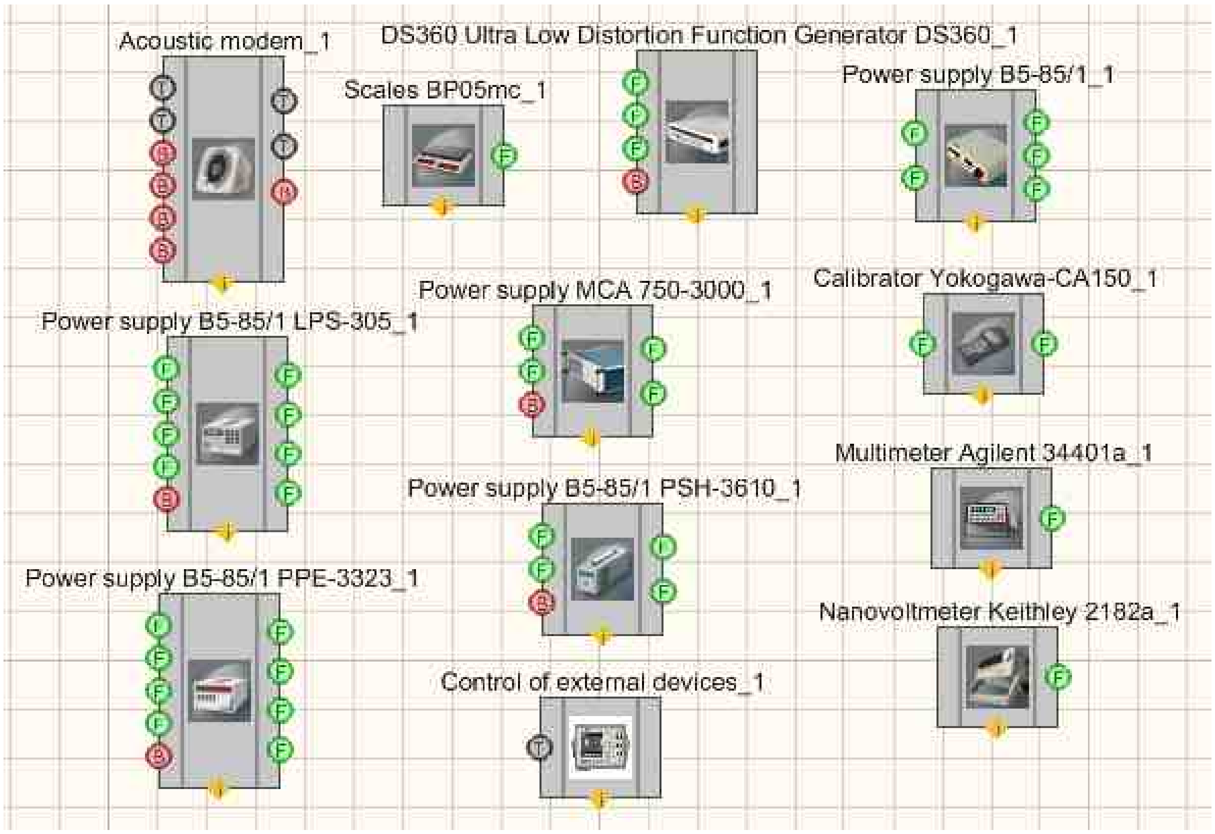


Figure 2.9.1b. Components of "External devices" section developer interface

Components of "Devices" section are used for external devices control (e.g. Narrow-band spectrum analyzer, calibrator), control of digital ports of the connected devices (ADC DAC module ZET 210), as well as to receive data from server channels. The section "Devices" contains the following components:

- **Switching unit relay keys** - The component is used for operation with LAN_connect equipped with integrated ZET 048 and allows to suspend communication for input output channels (the program sends codes with channels).

- **Switching unit** - it is used for control of switching unit module, connected to ADC DAC modules outputs. It is used to control relays status connected to commutator bar by means of digital port of the following modules: ZET 220, ZET 230;

- **ZET 0xxx2 and ZET 0xxx4/0xxx8** - it is used for measuring the parameters of spectral signals components from various signals, correlational structure of the signals, generation of electrical signals with preset metrological parameters, noise and vibration measurements in 1/3 octave band, thus allowing to use it as vibration and noise meter. It is used for visual representation, data acquisition and signals processing from various sensors: vibration transducers, measuring microphones, hydrophones with ICP integrated amplifiers, etc. The components is necessary for control, measuring and diagnostic equipment used for various measurements and tests, for long-term processes monitoring (both stationary and dynamic);

- **ZET 110** - the component is used for measuring various signals parameters, noise and vibration measurements, representation of the measured parameters, constant recording of signals to non-volatile integrated memory (volume – up to 2 Gb);
- **ZET 210** - it is used for measuring signals parameters in wide frequency range (with sampling frequency up to 400 kHz), received from various primary transducers. Digital (DB-15) and analog (DB-25) inputs can be used for the purpose of various mechanisms control;
- **ZET 220** - the component is used for measuring signals parameters with high accuracy and in wide frequency range (the signals are received from various primary transducers: thermal resistances, thermocouples, sensors with universal current output 4… 20 mA, accelerometers BC 201/202);
- **ZET 230** - the component is used for measuring signals parameters with high accuracy and in wide frequency range (the signals are received from various primary transducers: accelerometers BC 201, BC 202, microphones BC 501;
- **ZET 240** - is used for measuring and registering of signals from seismic receivers, the component also operates as a measuring instrument for electrical signals parameters;

- **ZET 440** - The component is used for conversion of electrical transducer's high-impedance charge signal (from vibration sensor, accelerometer or hydrophone) into current signal and conversion of ICP sensors signals into voltage signal;

  The component allows to connect accelerometers with charge output or ICP to FFT Spectrum Analysis and ADC / DAC modules.

  Thus, it is possible to connect hydrophones BC 311 to Narrow-band spectrum analyzer ZET 017. The amplifier ZET 440 is used in sensors control system in order to connect reference accelerometer AP10 to Narrow-band spectrum analyzer ZET 017;

- **ZET 7000 channel reading** - The component is used for reading the data from sensors of 7xxx series. This component allows to calculate the current measured value of the signal by device's serial number, channel number and device type. Serial number and device type are obtained from "Device manager" program. Serial number is shown in 16-digit format, without particular type indication (i.e., without 0x index);

- **Indicator ZET7x78** - It is intended for connection to measuring networks on the basis of digital sensors with CAN interface and serves for indication of values of measured parameters. Also, the module is used to monitor the voltage on the power bus in the network using the "Oscilloscope" program from the ZETLAB package or on the module screen.

- **Relay control module ZET7161** - the component is used for producing "dry contact" signal, it has 3 input relays, activated upon receipt of signal from CAN interface;
- **Digital port control module ZET7x60** – the component digital port / logical analyzer with RS-485 or CAN interface. It is used for the control of the connected device. The component is used in control and automation systems.
- **Synchronous generator ZET7090** - ZET 7090 Generator-485 - analog two-channel synchronous generator with RS-485 interface. It is used a get sine, impulse or saw-shape signals. The component has 2 synchronous channels and can be used independently or in control and automation systems;
- **Change of address ZET7xxx** – the component allows to change addresses of Zet7xxx sensors;
- **List of ZET7xxx devices** – It is intended for reading the list of devices of ZET7xxx modules;
- **TRC thermometer ZET7x20** - the component is used for temperature measurements in liquid, gaseous and granular environments that are non-aggressive to the TRC package. Application spheres: heat energy, chemical industry, metallurgy. The component is used together with intelligent temperature sensors with RS-485 interface;
- **TRC thermometer ZET7x21**- the component is used for temperature measurements. The component is used for temperature measurements and control in liquid, solid, gaseous and granular environments in various spheres of industry. It is used together with intelligent temperature sensors with RS-485 interface;
- **Management of network devices ZET7x76** - the component is used for connection of measuring networks based on digital sensors with RS-485 or CAN interface to PC via Ethernet network.
- **Relay control ZET7062** - is used for relay control. Settings: by default. The data is saved to flash-memory. The component is used to set initial configuration of the relay;
- **Device ZET** - the component is used for connection and setting parameters of ZET devices;
- **Digital generator ZET7060g** - the component is used for generation of impulse signals with frequency range up to 100 mHz;
- **Encoder ZET7060E** - the component is used for angular and linear displacement, shaft rotation velocity, acceleration and position measurements. The data is transferred by RS-485 interface (Modbus protocol);

The components of "Device" section are not displayed in operator mode and can be added to the project only in developer mode.

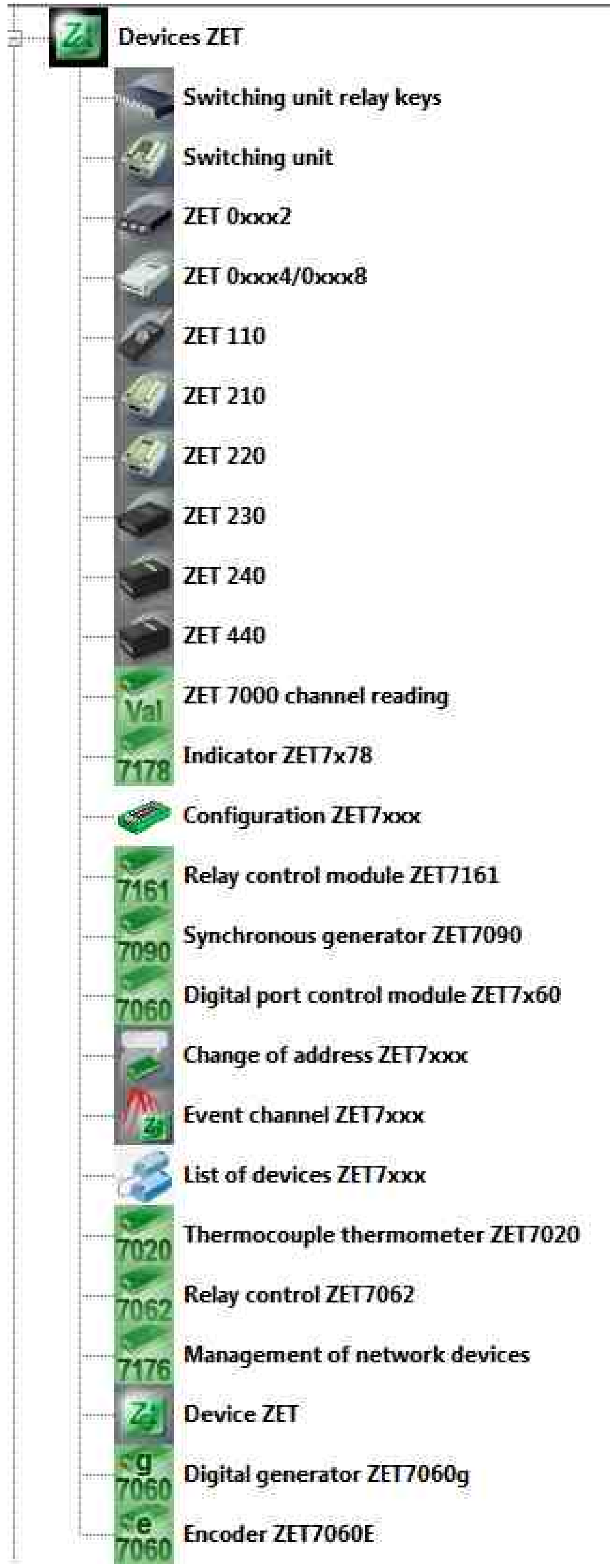

Devices ZET
Switching unit relay keys
Switching unit
ZET 0xxx2
ZET 0xxx4/0xxx8
ZET 110
ZET 210
ZET 220
ZET 230
ZET 240
ZET 440
ZET 7000 channel reading
Indicator ZET7x78
Configuration ZET7xxx
Relay control module ZET7161
Synchronous generator ZET7090
Digital port control module ZET7x60
Change of address ZET7xxx
Event channel ZET7xxx
List of devices ZET7xxx
Thermocouple thermometer ZET7020
Relay control ZET7062
Management of network devices
Device ZET
Digital generator ZET7060g
Encoder ZET7060E

Figure 2.9.2a. Devices window, section "Devices ZET".

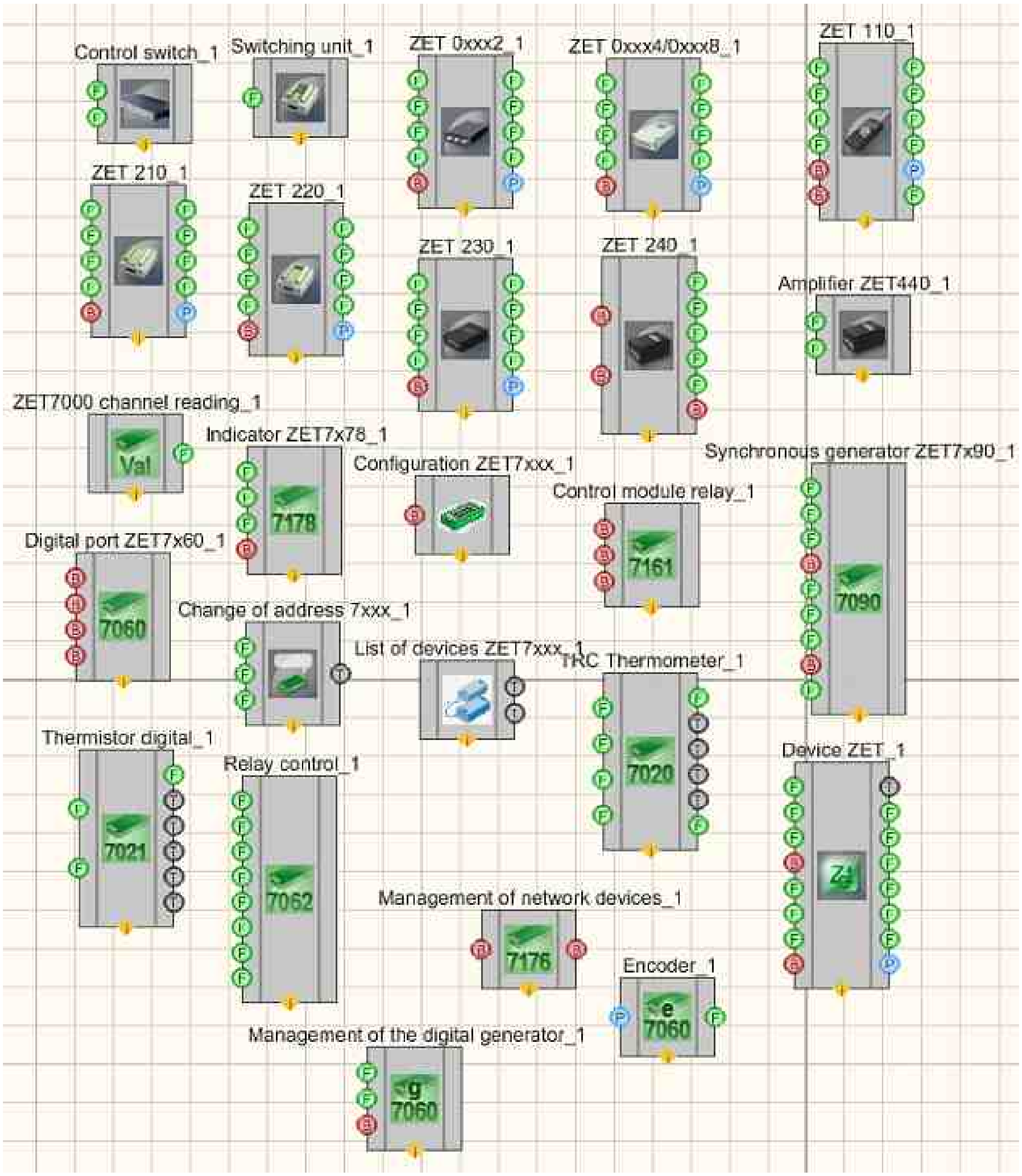
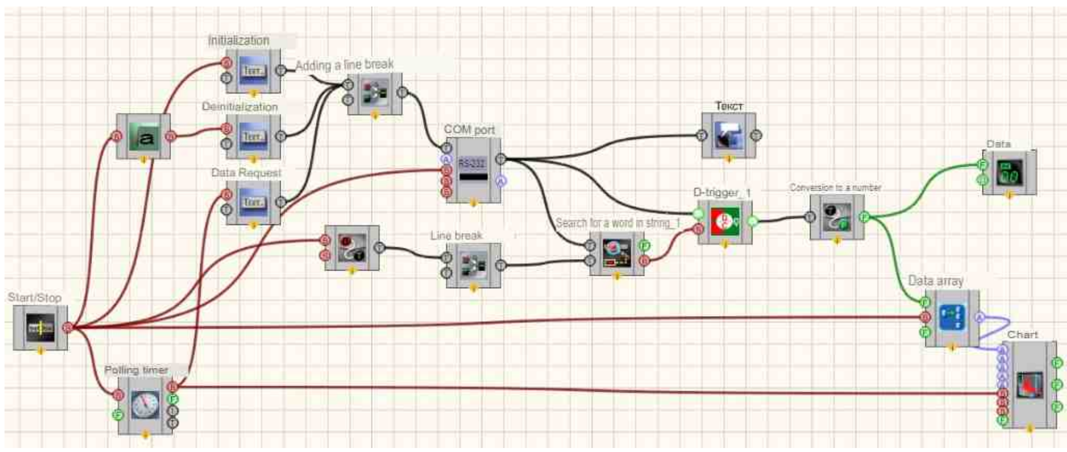


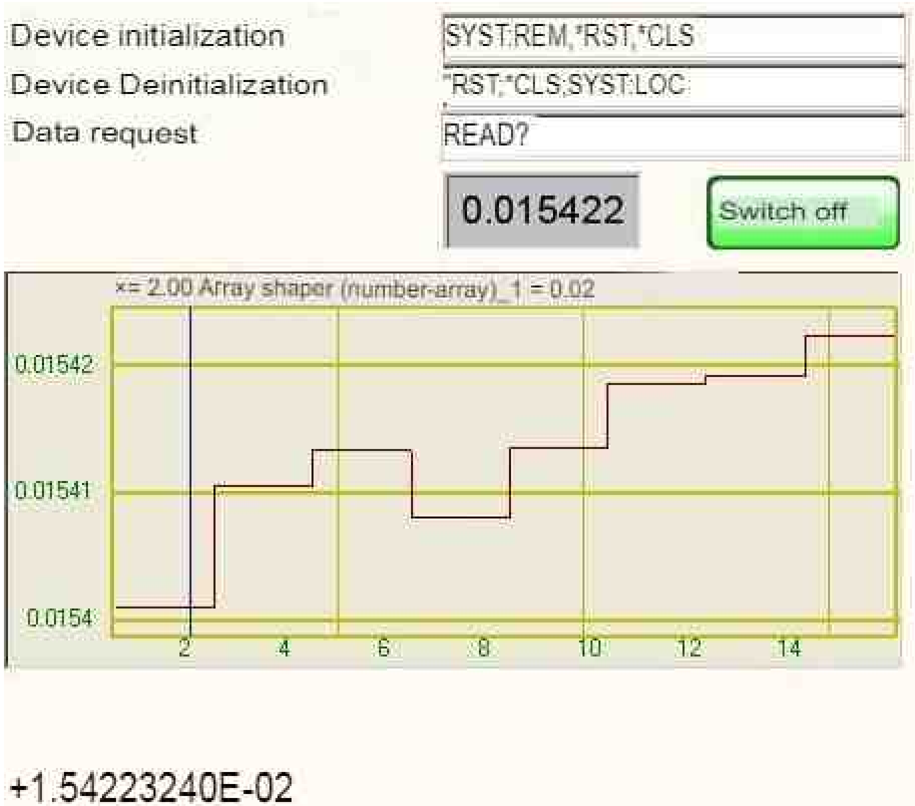
Figure 2.9.2b. Components of "Devices ZET" section developer interface

In order to perform actions described in the example, it is necessary to have a connected ZET-device, equipped with ADC and multimeter Agilent 34401a, set for PortNumber=3.

Example
Project in SCADA ZETView



Project operation results



View the example in ZetView

Part 3. Debugging projects in the SCADA system ZETVIEW

Глава 1.Debugging projects in the SCADA system ZETVIEW

Debugging projects in the SCADA system ZETVIEW

During the design of the project in the ZETVIEW SCADA system (and in any other), designers will have a question about checking the work of the project. There is a need to constantly monitor the status of all elements. It is possible to connect the corresponding indicators to the outputs of all components, but it is better to use the debugging window.

The debugging window opens in the "Working Environment" tab - "Debugging Window". It lists the components selected by the project developer for debugging. Each component has some contacts and some properties, they can be disclosed to monitor their condition and hidden to save space.

To add a component to debugging, right-click on it and select "Add to Debugging" from the drop-down menu. The body of the component will then turn red, and a line with the name of the component will appear in the debugging window. To remove a component from debugging, right-click on the component and select "Remove from debugging" from the drop-down menu. The body of the component will turn gray again, and the corresponding line will disappear from the debugging window.

In the debugging window, all custom properties and contact pads are recorded in the form of a tree and grouped into two groups: contact pads in the "contacts" group, custom properties in "properties". Each row of the tree has two cells on the right in the table (except for the rows with the name of the component and the rows "contacts" and "properties"): "value" and "type". The cells containing the data types of the contacts have a color corresponding to the color of the contact pad.

For debugging, contacts are available that transmit numbers (green contacts with the letter "F"), impulses (red contacts with the letter "B"), strings (black contacts with the letter "T") and arrays (purple contacts with the letter "A"). For other types of contacts, a dash is written in the "value" cell.

When you start the project, the debugging window will remain visible. It can be "separated" from the main window and "attached" to another place or left as a separate window. When saving the project, the components added to the debugging window will retain this property and will also be added to the debugging window when loading. When compiling the executable file, the debugging property will also be saved and when the project starts, a debugging window will appear with a list of observed components.

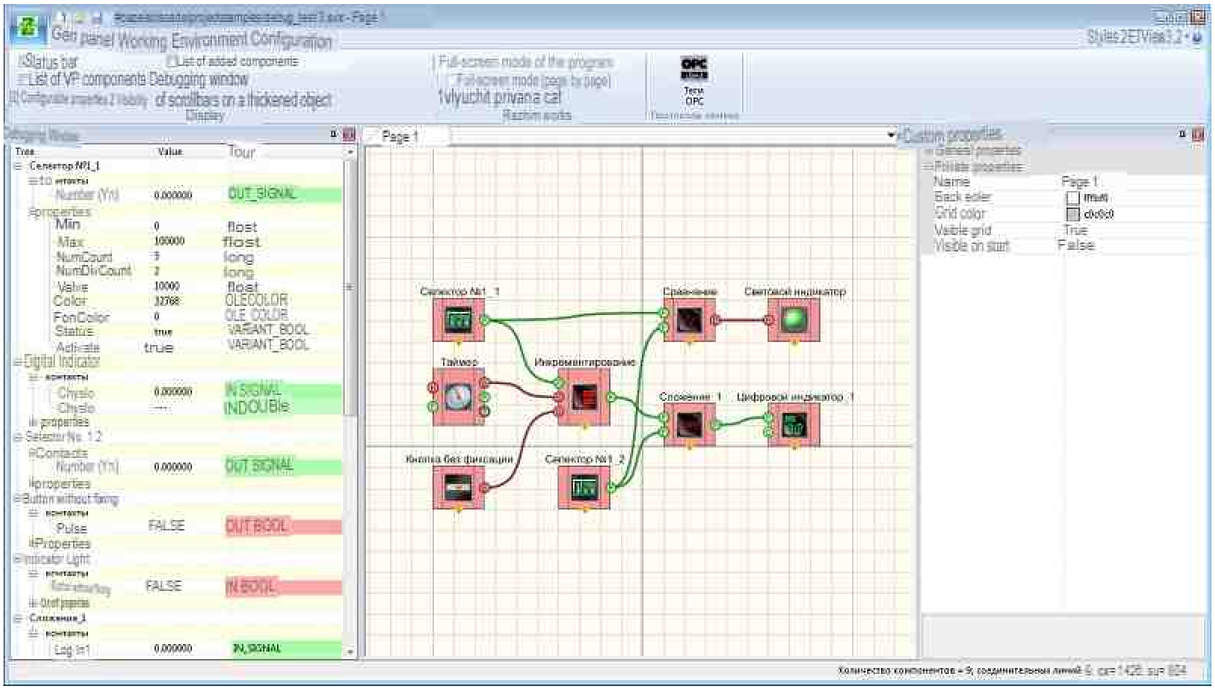
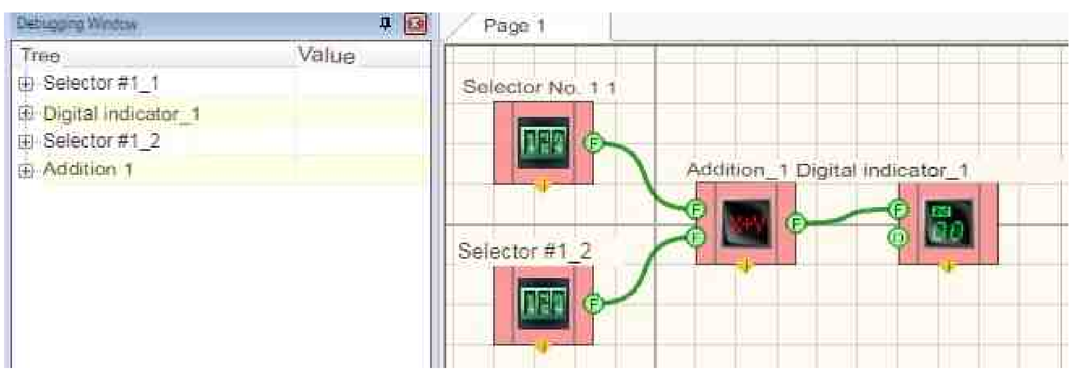


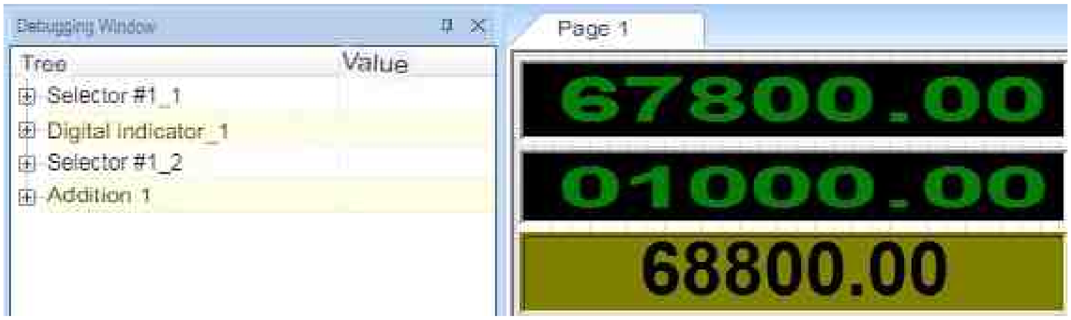
Figure 3.1. Debugging projects in SCADA- system ZETVIEW

Example No. 1

Project in SCADA ZETView



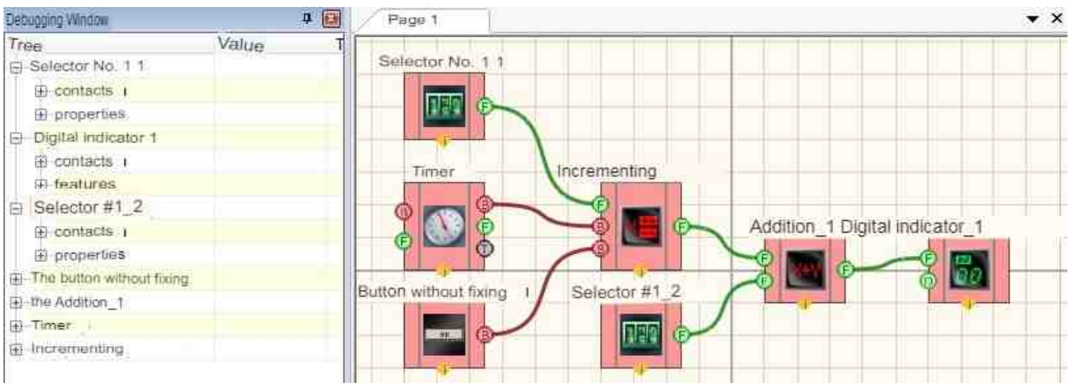
Project operation results



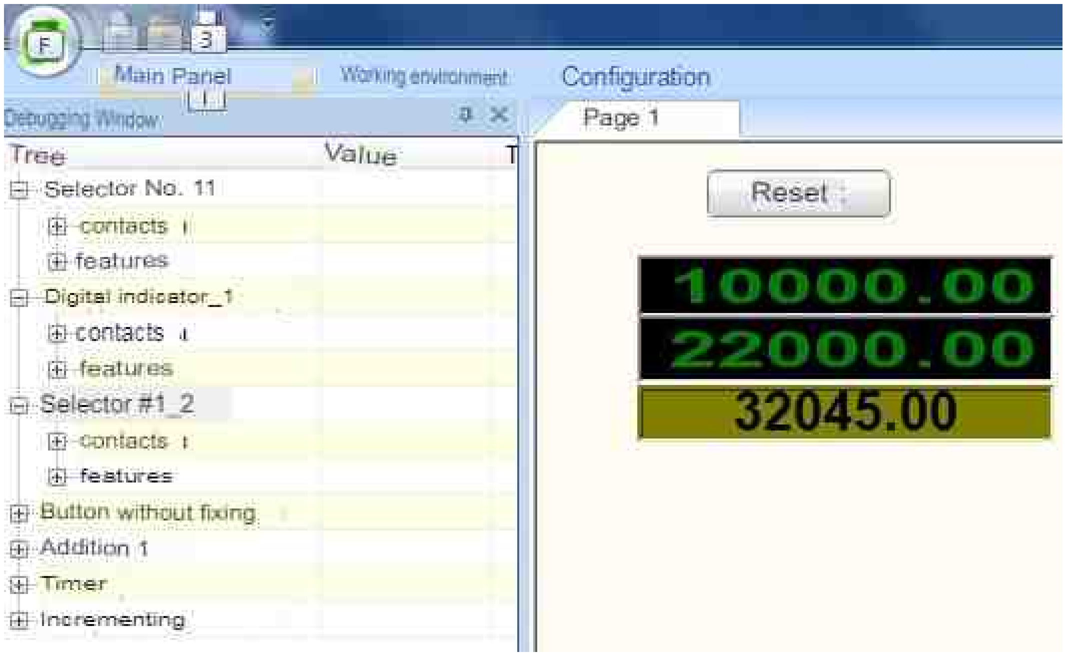
View the example in ZetView

Example No. 2

Project in SCADA ZETView



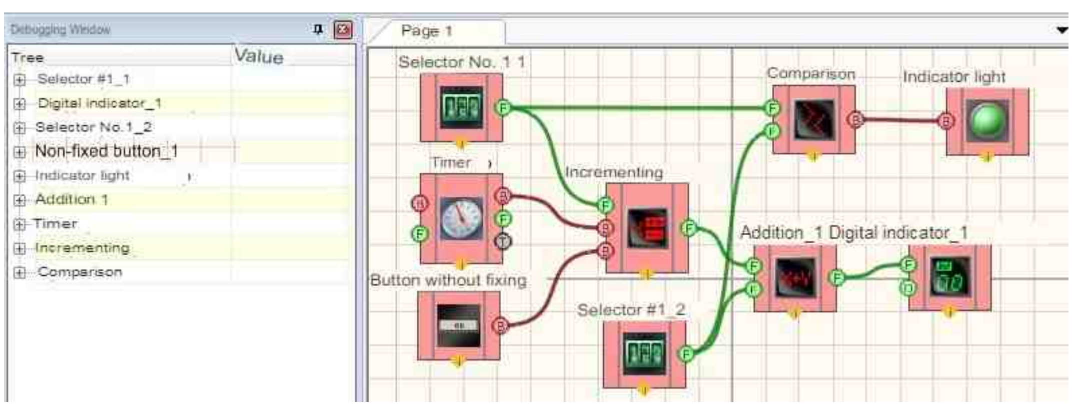
Project operation results



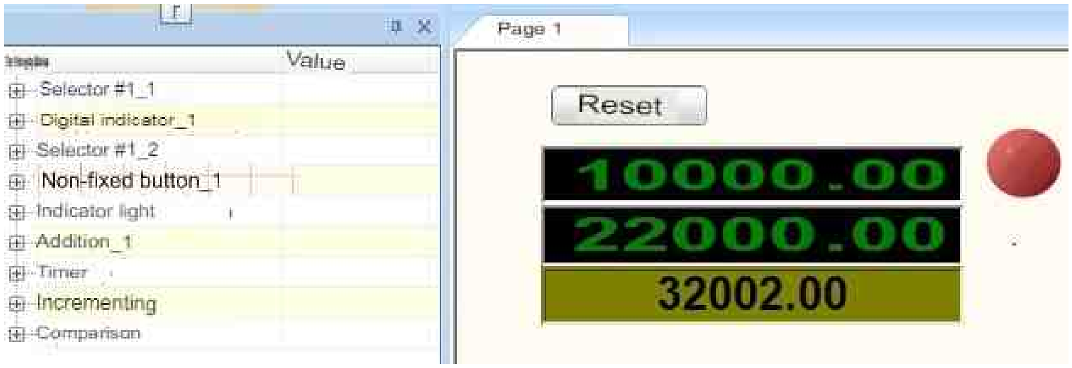
View the example in ZetView

Example No. 3

Project in SCADA ZETView



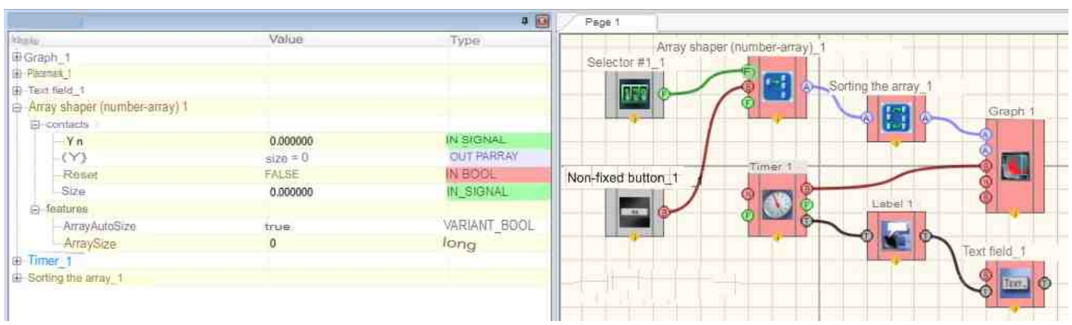
Project operation results



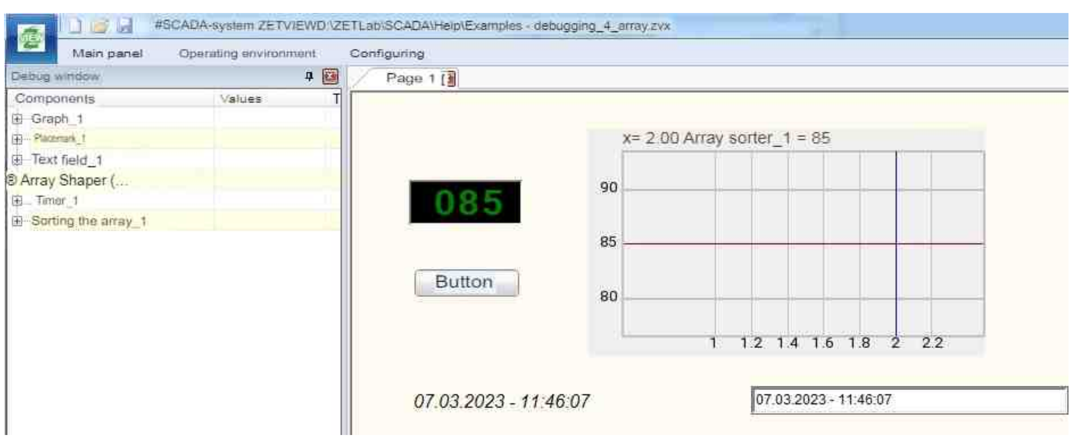
View the example in ZetView

Example No. 4

Project in SCADA ZETView



Project operation results

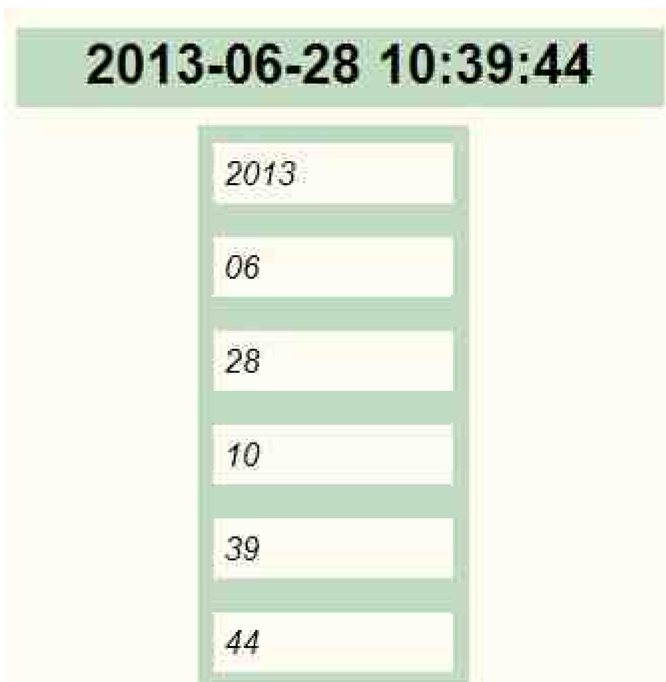


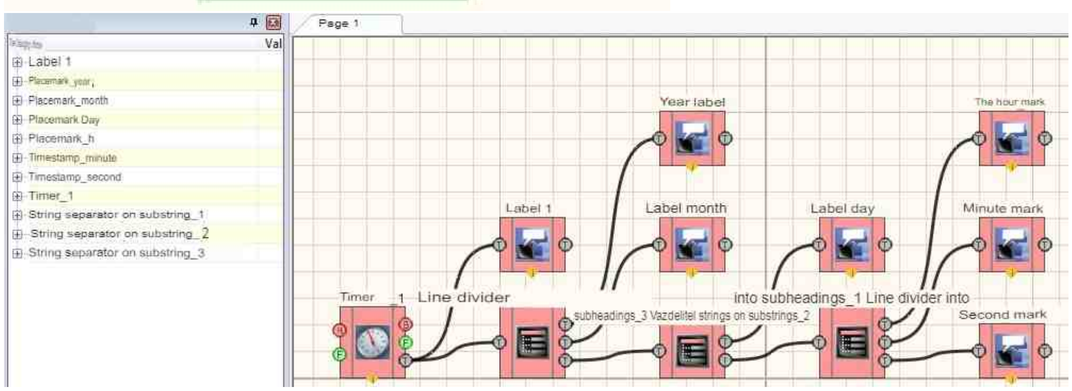
View the example in ZetView

Example No. 5

Project in SCADA ZETView

Project operation results



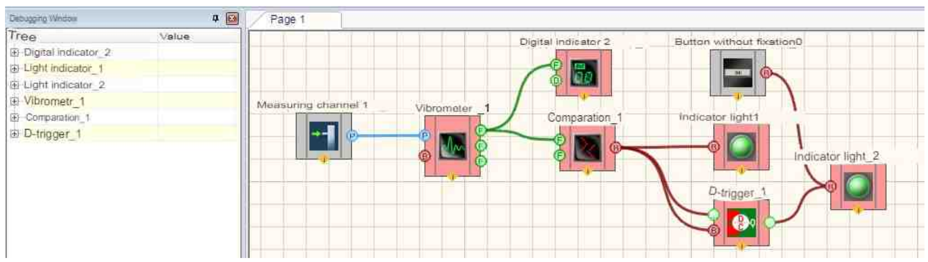


View the example in ZetView

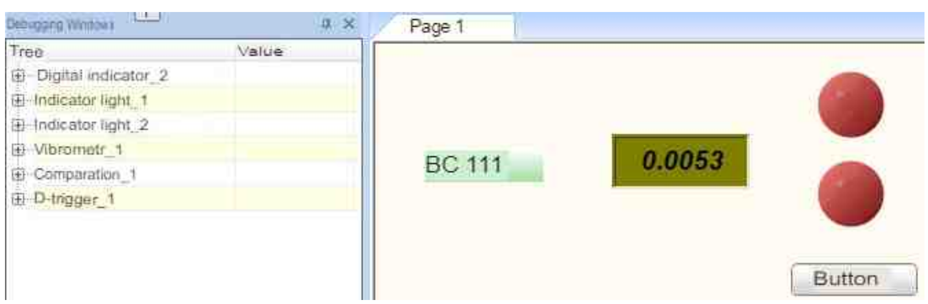
Example No. 5

Example using a vibrometer. The vibrometer works only with accelerometer channels, and it recognizes them by units of measurement: "g" or "m/sI".

Project in SCADA ZETView



Project operation results



View the example in ZetView

Note:

Debugging is a time – consuming process that is inevitable when creating projects.

Viewing of running programs is possible via unit in the project for debugging in ZETView.exe . The units become visible when entering the sha 1 command and are hidden by shu0. The command is entered from the keyboard after clicking on any of the ZETView pages. In order for Units to appear, they must be launched, often units are launched after the start of the project.

Also added to SCADA components;

- DC voltmeter;

- AC voltmeter;
- Signal generator;
- Synchronous generator;
- Vibrometer;
- Spectrum CPB Analysis.

Part 4. Components of ZETView

Глава 1.General properties of the components

General properties of the components - this is a set of graphic and identification properties that all components of SCADA ZETView possess.

➢ Common properties (environment):

- Caption - each component of SCADA-system ZETView has a name. As the component is added to the project, it has a name by default, for instance, "DC Voltmeter". The developer can change the component name at his discretion, e.g. for "Generator voltage". It is possible to change the name of the component in "Custom properties window" in "Caption" line. As scenario is using corresponding component, its name is depicted (so, it is reasonable to assign individual caption for each of the components). Components names are depicted above them only in developer mode, while in operator mode it is possible to see the component name in properties window.
- ToolTip - tooltip for the user.
- VisibleInRun - component visibility in operator mode. Allows to enable/ disable component visibility in operator mode when the project is being run.
- m_sHelpString - identifier string. All the components have it by default (it is filled automatically). The identifier string is used in the case when the properties of a component are to be used for another component. For instance, converter (numeric array – line array) + Data table. This feature is also often used for scripts creation (programmable component) – identifier string.

➢ Coordinate of the lower left angle:

- Left - X coordinate of the upper left angle.
- Top - Y coordinate of the upper left angle.

➢ Configuration:

- Debug - configuration collected under Debug.
- Release-configuration assembled under Release.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

➢**Custom common properties:**

- BSTR Caption - Set the name of the component.
- BSTR ToolTip - Set the prompt for the user.
- VARIANT_BOOL VisibleInRu - Set the component visibility in operator mode. Allows to enable/ disable component visibility in operator mode when the project is being run.
- BSTR m_sHelpString - Set the identifier string. All the components have it by default (it is filled automatically). The identifier string is used in the case when the properties of a component are to be used for another component. For instance, converter (numeric array – line array) + Data table. This feature is also often used for scripts creation (programmable component) – identifier string.
- ULONG Width - Set the component width (number).
- ULONG Height - Set the component height (number).
- ULONG Left - Set the X coordinates of left angle (number).
- ULONG Top - Set the Y coordinates of left angle (number).

- **Input:**

- Key - Input channel with logical values (0 - no, off / 1 yes, on, true).

➢**Methods:**

- void AboutBox() - Sets the information about the component.

Глава 2.Automation

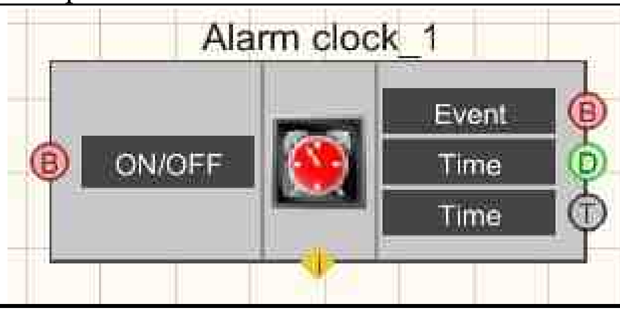
In SCADA ZETView projects, the control functions are performed by means of automation components.

2.1.Alarm clock

The component "**Alarm clock**" is intended for use in automated control systems developed on the basis of ZetView. The alarm is triggered according to the computer time. The response time is determined to the nearest second.

The SCADA component "**Alarm Clock**" works with the values of its properties that were set at the time of the component launch with the SCADA project running or at the time of the project launch, provided the component is autorun. Therefore, in order to change the properties of a component during the project, it is necessary to stop the component, change the values of the properties (or vice versa), and then restart the component

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- On/Off – enabling/disabling the component (duplicating the **Activate** property).

➢ Output:

- Event – at the moment the alarm is triggered, a logical unit is briefly applied to this contact.
- Time – when the alarm is triggered, the alarm time is given to this contact in the DATE format
- Time - when the alarm is triggered, a string containing the date/time of the alarm in the format specified by the **DateInString** property is issued to this contact.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- **Activate** (true) – switching the component on/off (duplicating the input contact On/Off).
- **OperationType** (once a week) – determines the trigger period. In this case, the type of the first-actuation will be determined by the First Type property.
 1. once – triggered once at a time determined by properties that have the word "First" in their name;
 2. once an hour;
 3. once a day;
 4. once a week;
 5. once a month;
 6. with a specified period – the period is set in the IntervalInSec property.
- **FirstType** (at the specified time) – sets the type of the first-actuation:
 1. at a given time – the first-actuation occurs at a time determined by properties whose name contains the word "**First**";
 2. with a specified delay – the first-actuation occurs with a delay after the start of the project. The delay value is determined by the **DelayInSec** property.

DateInString (DD.MM.YYYY HH:MM:SS) – defines the text output format with the response time to the output trigger. Possible options:
1. Date and time – date in DD.MM.YYYY format and time in HH format:MM:SS (DD.MM.YYYY HH:MM:SS);
- 2. date only - date in the format DD.MM.YYYY (DD.MM.YYYY);
3. time only - time in HH format:MM:SS (HH:MM:SS);
4. Date and time – date in YYYY-MM-DD format and time in HH format:MM:SS (YYYY-MM-DD HH:MM:SS);
5. Date and time – date in YYYY_MM_DD format and time in HH_MM_SS format (YYYY_MM_DD_CH_MM_SS).
- **FirstYear** (2016) – the year of the first-actuation.
- **FirstMonth** (12) – the month of the first-actuation, from 1 to 12.
- **FirstDay** (6) – the day of the first trigger, from 1 to the maximum number of days in the month specified by the FirstMonth property in the year specified by the FirstDay property. **FirstDay**.
- **FirstHour** (12) – hour of the first-actuation, from 0 to 23.
- **FirstMinute** (20) – minute of the first-actuation, from 0 to 59.
- **FirstSecond** (11) – second of the first-actuation, from 0 to 59.
- **DayOfWeek** (Tuesday) – When set to trigger once a week, specifies the day of the week the trigger is triggered.
- **IntervalInSec** (5) – when setting operation with a given period, determines the value of this period in seconds.
- **DelayInSec** (5) – when setting the first-actuation with a specified delay, determines the value of this delay in seconds.
- **Beep** (false) – determines whether or not to emit a sound signal when triggered.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
➢

VARIANT_BOOL Activate - Set alarm status:
- true - the alarm is on;
  false - the alarm is off.

- BSTR OperationType - Set frequency of operation: Once; Once an hour; Once a day; Once a week; Once a month; With a specified period (string).
- BSTR FirstType - Set first-actuation: At a specified time; With a specified delay; (string).
- BSTR DateInString - Set display of the trigger time: Date and time; Date only; Time only (string).
- LONG FirstYear - Set year of the first-actuation (any year).
- USHORT FirstMonth - Set month of the first-actuation (from 1 to 12).
- USHORT FirstDay - Set day of the first-actuation (from 1 to the maximum number of days in the month defined by the **FirstMonth** property, in the year defined by the **FirstDay** property).
- USHORT FirstHour - Set hour of the first-actuation (from 0 to 59).
- USHORT FirstMinute - Set minute of the first-actuation (from 0 to 59).
- USHORT FirstSecond - Set second of the first-actuation (from 0 to 59).
- BSTR DayOfWeek - Set days of the trigger week: Monday; Tuesday; Wednesday; Thursday; Friday; Saturday; Sunday (string).
- LONG IntervalInSec - Set trigger interval in seconds (any value).
- LONG DelayInSec- Set delay of the first-actuation in seconds (any value).
- VARIANT_BOOL Beep - Set sound signal:
  true - the sound signal is turned On;
  false - the sound signal is turned Off.

The component is not visible in operation mode.

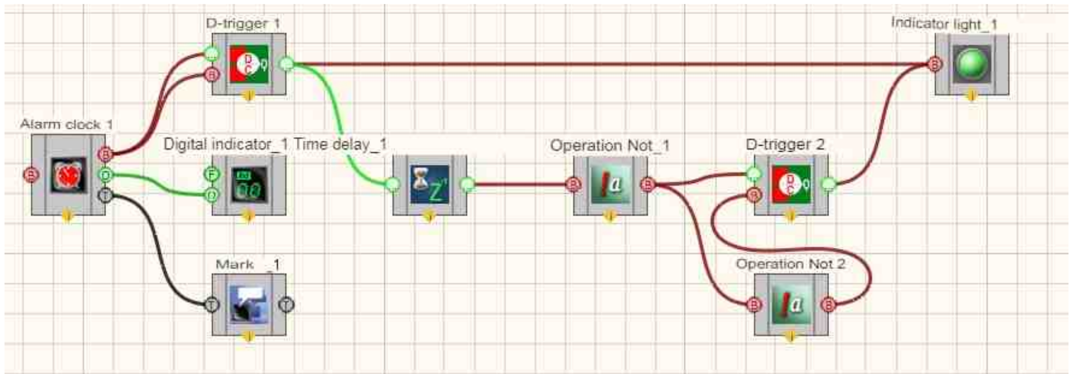
When creating a component, i.e. when placing it on the field in development mode, the time of the first operation, determined by the properties of the component, is equal to the time of component creation. The day of the trigger week will be set as the day of the component creation week.

When set the first-actuation at a given time, there may be a situation in which the project launch time will be later than the first-actuation time set by the properties. In this case, with a single operation mode, the component will work immediately after startup. In all other cases, the timed responses will be ignored.

When triggered, the output contacts will be given the appropriate information. The duration of holding the logical unit on the "Event" contact is so small that when the "Light Indicator" component is connected to it, the human eye does not have time to notice the color change of the indicator when triggered.

Example

Project in SCADA ZETView



Project operation results

View the example in ZetView

2.2.Control circuits

The component "**Control circuits**" for electrical circuit monitoring devices, the ZET 452 is designed to measure the resistance of electrical circuits and the insulation resistance of electrical circuits in automatic mode according to a preset program without the participation of an operator. The device reduces the time required for inspections by more than 20 times compared to using standard measuring instruments.

The operation of the ZET 452 electrical circuit monitoring device is based on the sequential measurement of the resistance of the circuits of the product being tested. The software "Control of electrical circuits" included in the delivery package has an intuitive graphical interface for controlling and displaying measurement results and is used to:

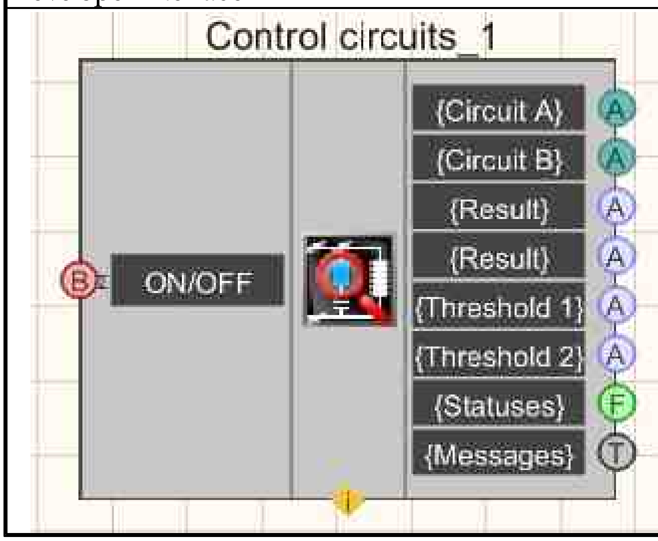
- set the sequence of checks,
- save test scenarios to a file,

- set tolerances for measured values to compare the result,
- and run an automatic check of the connected specimen.

The measurement results are displayed on the screen in tabular form, the rows of the table are colored depending on the results of checking the received resistance value. It is possible to save the results to a file.

Using the component requires a connected ZET 452.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- On/Off – enables/disables the component.

➢ Output:
- {Circuit A} is an array of circuits A.
- {Circuit B} is an array of circuits B.
- {Result} is an array of measured parameters.
- {Result} is an array of measured parameters.
- {Threshold 1} is an array of threshold values 1.
- {Threshold 2} - an array of threshold values 2.
- {Statuses} – check status indicator.
- {Messages} - messages about the results of the check.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- **DeviceNumber** (0) – serial number of the device.

Checktype (Resistance) – determines the type of control.
- 1. Resistance;
  2. Isolation;
  3. Control of circuits.
- **NetFileName** ( ) – the name of the compliance file for control the parameters of electrical circuits.
- **Profile** ( ) – the name of the test profile file.
- **RoundResult** (true) – rounding results.
- **Usedvoltage** (2,5 V) – voltage used (2.5V or 5V).

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Custom common properties:**
- Are available by the link below. 154

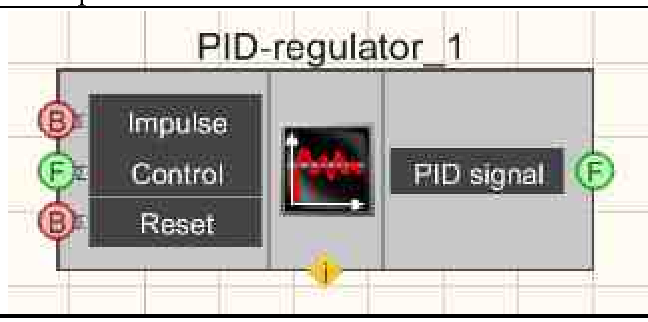
➢**Custom private properties (the default value is shown in parentheses):**
➢
- LONG DeviceNumber - Set serial number of the device (any value).
- BSTR CheckType - Set type of control: Resistance; Insulation; Circuit control (string).
- BSTR NetFileName - Set the name of the compliance file for control the parameters of electrical circuits (string). *.cbl format file (after setting this parameter, the matching net names are read from the specified file).
- BSTR Profile - Set the name of the test profile file into a *.ctp file (after setting this parameter, the profile is read from the specified file).
- VARIANT_BOOL RoundResults - Set rounding results:
  true - results rounding is turned On;
  false - results rounding is turned Off.
- BSTR UsedVoltage - Set the voltage used: 2.5V; 5V (string).

➢**Methods:**
- void Run(VARIANT_BOOL action) - Sets the method to start/stop the check.

2.3.PID-regulator

The component "**PID-regulator**" is designed to create automated control systems based on ADC-DAC modules, on Example module "SigmaUSB", to maintain a controlled value equal to a given value.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - when an impulse is given, the regulation starts/stops.
- Control - feedback of the regulated component.
- Reset - the integral component is reset (the sum of accumulated errors becomes 0).

➢ Output:

- PID signal - regulated value at the output.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Value (0) - Current value.
- Enabled (true) - enable/disable the component.
- PropKoef (1) -coefficient at the proportional component (opposes the deviation of the controlled variable from the set value).
- IntegrKoef (0) - coefficient at the integral component (taking into account the previous values, helps to eliminate the static error).
- DiffKoef (0) - coefficient at the differential component (provides a quick response to a deviation from the set value).
- MaxCurrentValue (0) - maximum value of the control signal. If the value is 0, then it is not used.
- MinCurrentValue (0) - minimum value of the control signal. If the value is 0, then it is not used.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Control - The value of the controlled parameter for calculating the output value of the PID controller (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

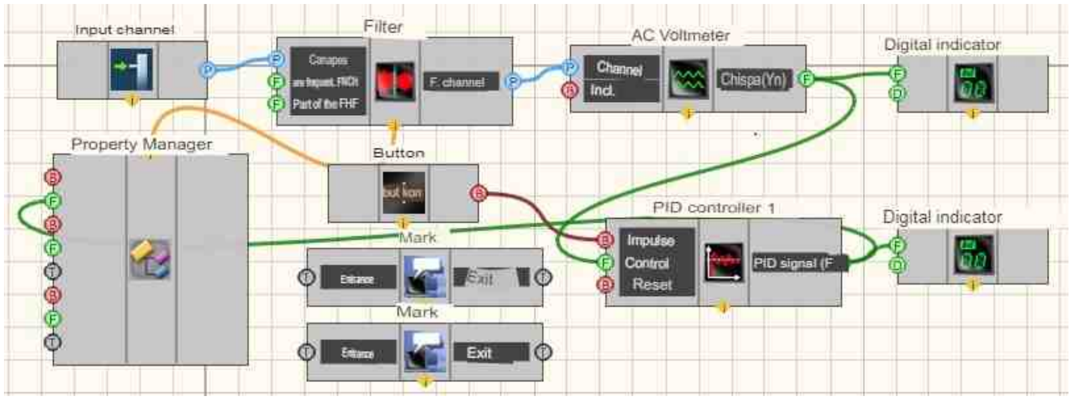
- FLOAT Value - Set the PID-regulator set value (from minus infinity to plus infinity).
- VARIANT_BOOL Enabled - Set the running status of the PID-regulator:
  true - PID-regulator is turned On;
  false - PID-regulator is turned Off.
- FLOAT PropKoef - Set the proportional gain of the PID-regulator (any value).
- FLOAT IntegrKoef - Set the integral gain of the PID-regulator (any value).
- FLOAT DiffKoef - Set the PID-regulator differential gain (any value).
- FLOAT MaxCurrentValue - Set the maximum output value of the PID-regulator (any value).
- FLOAT MinCurrentValue - Set the minimum value of the PID-regulator output signal (any value).

➢**Methods:**

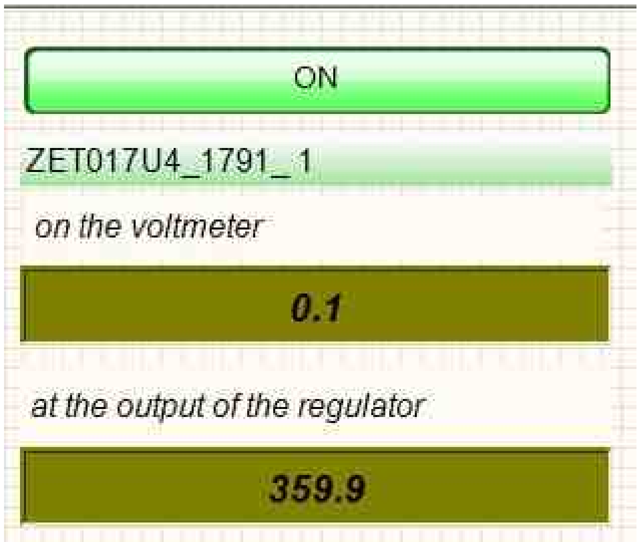
- void ResetIntegr(LONG newVal) - Sets the integral term of the PID-regulator where newVal - the value of the integral component (any value).

Example

Project in SCADA ZETView



Project operation results



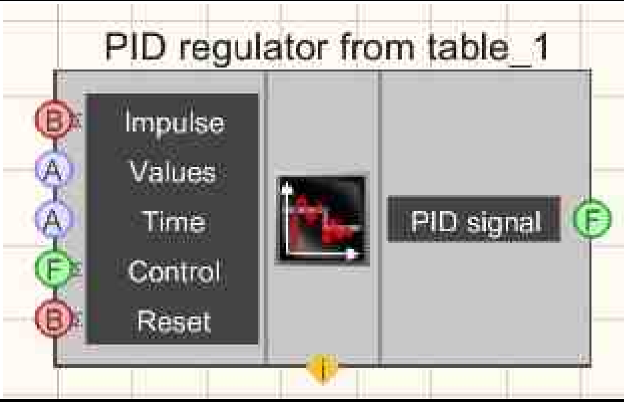
View the example in ZetView

2.4.PID-regulator from table

The component "**PID-regulator from table**" is designed to adjust the signal level in accordance with the specified profile in the table.

Appearance of the component:

Developer interface	Operator interface



Setting:

➢Input:
- Impulse - when an impulse is given, the regulation starts/stops.
- Values - a numeric array of values is supplied.
- Time - a numeric array of time is supplied, s.
- Control - feedback of the regulated component.
- Reset - the integral component is reset (the sum of accumulated errors becomes 0).
-

➢ Output:
- PID signal - regulated value at the output

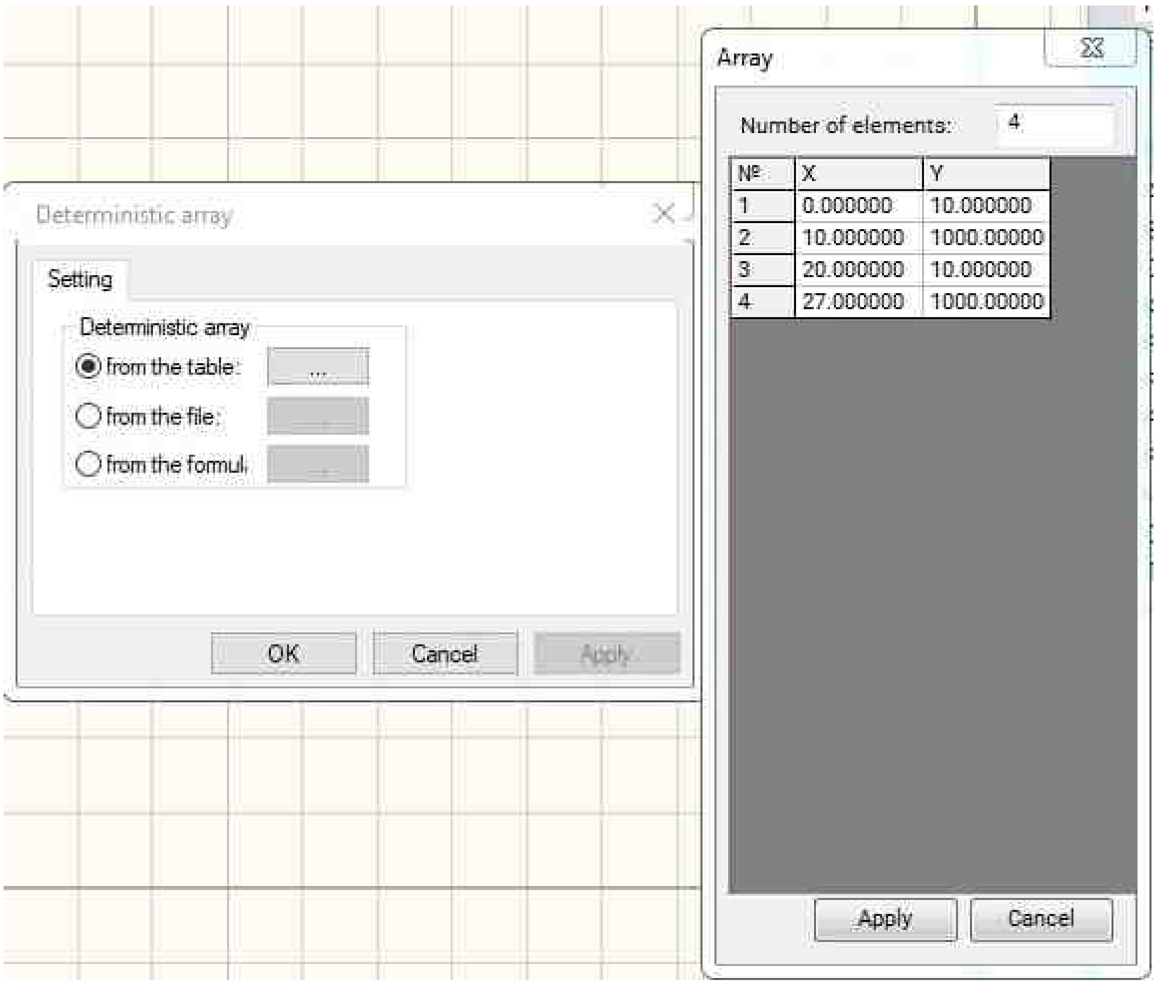
Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Enabled (true) - enabling/disabling the PID-regulator.
- PropKoef (1) - the coefficient with a proportional component (counteracts the deviation of the controlled value from the set value).
- IntegrKoef (0) - the coefficient with the integral component (taking into account the previous values, helps to eliminate the static error).
- DiffKoef (0) - coefficient with a differential component (provides a quick response to a deviation from a given value).
- MaxCurrentValue (0) - the maximum value of the control signal. If the value is 0, it is not used.
- MinCurrentValue (0) - the minimum value of the control signal. If the value is 0, it is not used.
- ProcessVisible (false) - enabling/disabling the display of the regulation progress bar.

Additional information:

This component is convenient to use in conjunction with the Deterministic array component when specifying a data source "from a table" in it.

On Example:



In this table, column X is used as an array of time (time is specified in seconds), column Y is an array of values. The first value of the time array must always be 0, and the first value of the data array is the first control action sent to the managed object.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Values - An input numeric array for specifying the values of the nodes of the regulation table (any number).
- Time is an input numeric array for specifying the value of the times to which the value of the regulated parameter should be equal to the corresponding cell from the regulation table (any number).
- Control - The value of the adjustable parameter for calculating the output value of the PID-regulator (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

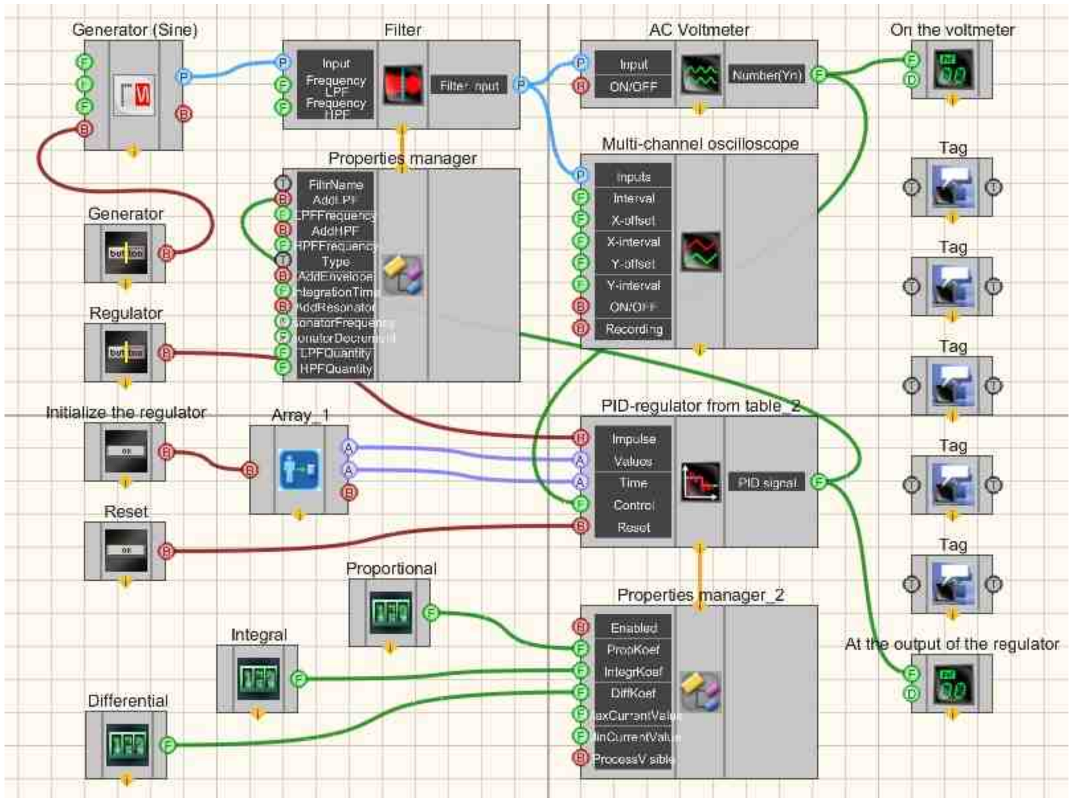
- VARIANT_BOOL Enabled - Set the operation status of the PID-regulator:
  true - the PID-regulator is turned On;
  false - the PID-regulator is turned Off.
- FLOAT PropKoef - Set the proportional coefficient of the PID-regulator (any value).
- FLOAT IntegrKoef - Set the integral coefficient of the PID-regulator (any value).
- FLOAT DiffKoef - Set the differential coefficient of the PID-regulator (any value).
- FLOAT MaxCurrentValue - Set the maximum value of the output signal of the PID-regulator (any value).
- FLOAT MinCurrentValue - Set the minimum value of the output signal of the PID-regulator (any value).
- VARIANT_BOOL ProcessVisible - Set the status of the progress bar execution display:
  true - the progress bar execution display is turned On;
  false - the progress bar execution display is turned Off.

➢**Methods:**

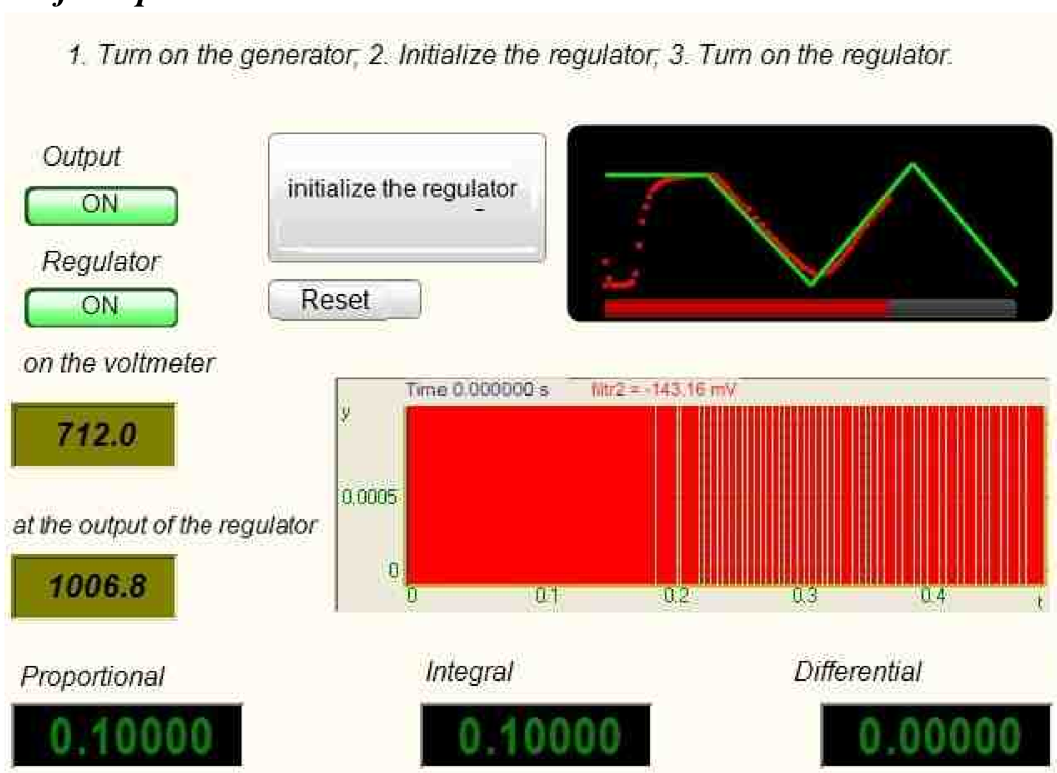
- void ResetIntegr(LONG newVal) - Sets the integral component of the PID-regulator where **newVal** is the value of the integral component (any value).

Example

Project in SCADA ZETView



Project operation results

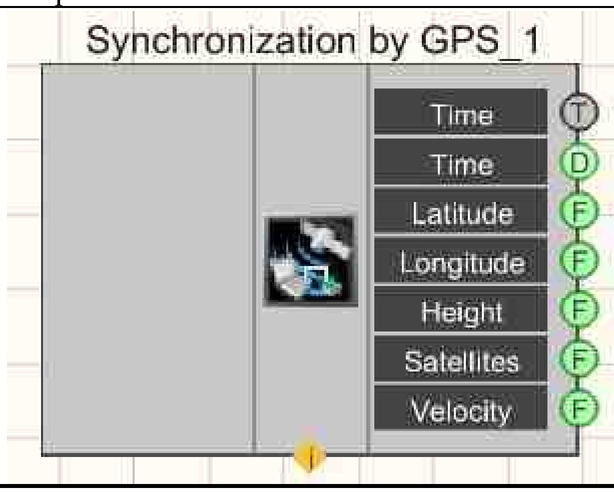


View the example in ZetView

2.5.Synchronization by GPS

The component "**Synchronization by GPS**" is designed to synchronize remote objects with each other, as well as to link synchronization to a single time scale.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

Doesn't have.

➢ Output:

- Time - GPS time (GPST).
- Time - GPS time (GPST).
- Latitude - the latitude of the device location, degr.
- Longitude - the longitude of the device location, degr.
- Height - the height of the device location, m.
- Satellites - the number of satellites on which work is carried out.
- Velocity - the velocity of the device, km/h.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- inputport () - is a serial port for communication with a GPS receiver.
- outputport (No) - serial port for GPS data output.
- baudrate(9600) - is the NMEA data reception rate of the stream, bits/s.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

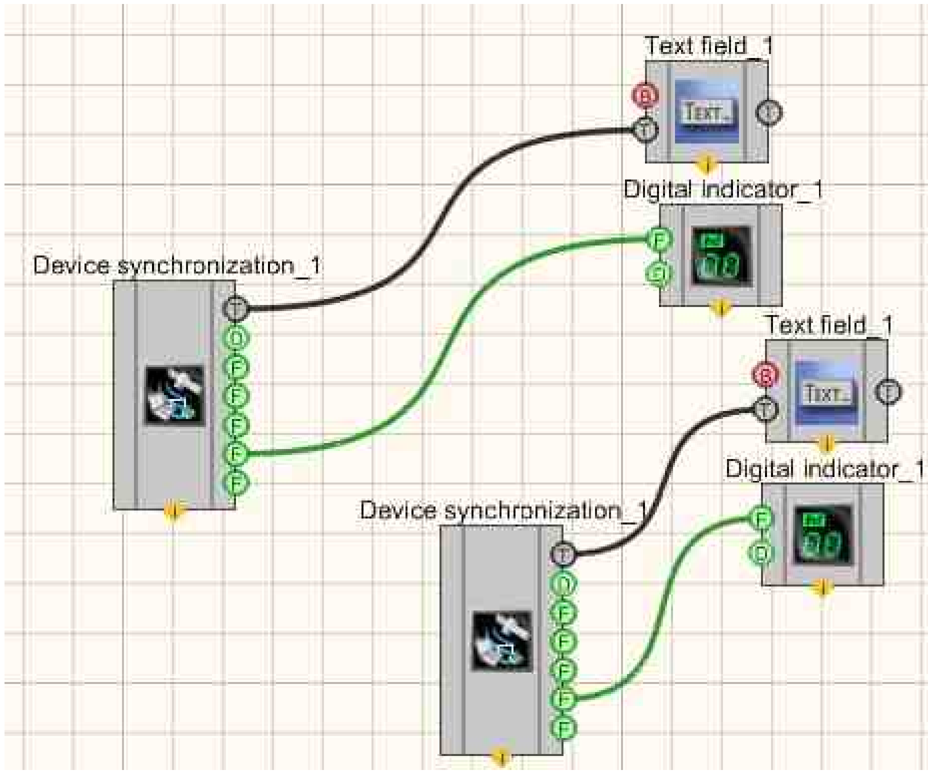
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

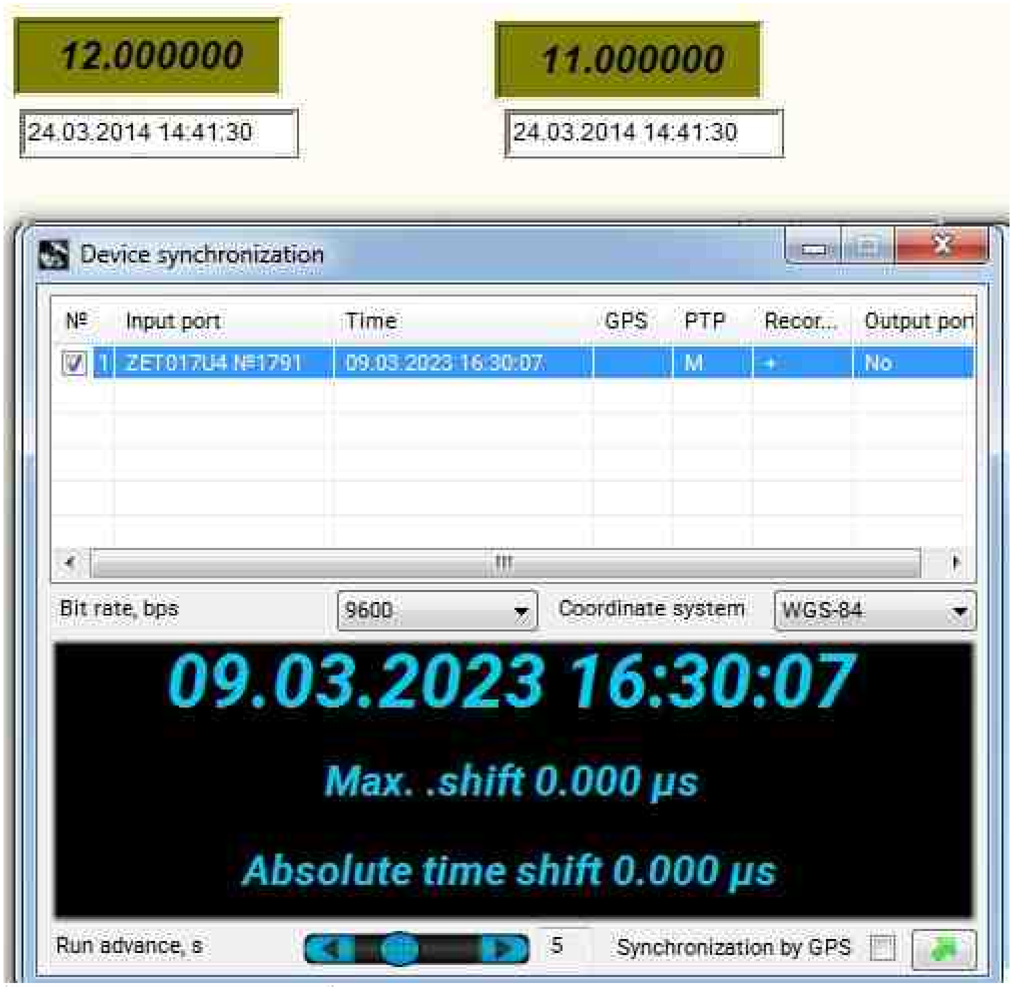
- BSTR inputport - Set the serial port for exchange with the GPS receiver (number).
- BSTR outputport - Set the serial port for GPS data output (number).
- BSTR baudrate - Set the NMEA data reception rate of the stream, bit/s (string):
  57600 - 57600 bits/s;
  38400 - 38400 bits/s;
  19200 - 19200 bits/s;
  9600 - 9600 bits/s;
  4800 - 4800 bits/s.

Example

Project in SCADA ZETView



Project operation results



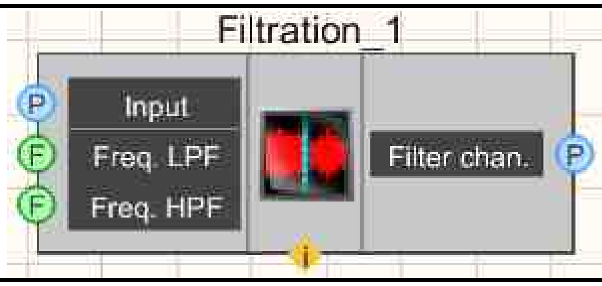
View the example in ZetView

2.6.Filtration

The component "**Filtration**" creates additional virtual channels for signal processing. All signals - real and virtual - have internal synchronization, which allows their joint processing. It is intended for reliable estimation of any signal parameter, for Example, level, frequency, correlation coefficient with another signal.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- Input - input filtered channel.
- Fraq. LPF - the LPF cutoff frequency, Hz
- Fraq. HPF - the HPF cutoff frequency, Hz

➢ Output:
- Filter chan. - filter output channel

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- FiltrName () - the name of the filter channel.
- AddLPF (false) - low-pass filter (LPF) operation permissions
- LPFFrequency (1000) - the cutoff frequency of the low-pass filter (LPF), Hz
- AddHPF (false) - high-pass filter (HPF) operation permissions
- HPFFrequency (100) - the cutoff frequency of the high-pass filter (HPF), Hz
- type (linear) - filter type:
  1. linear;
  2. differentiating 1st order;
  3. differentiating 2nd order;
  4. integrating the 1st order;
  5. integrating the 2nd order.
- AddEnvelope(false) - the resolution of the envelope.
- IntegrationTime (0) - integration time, ms.
- AddResonator (false) - resolution of a valid resonator.
- ResonatorFrequency (1.0) - resonance frequency.
- ResonatorDecrement (0.1) - decay decrement.
- LPFQuantity (1) - number of repetitions of low-pass filters (LPF) .
- HPFQuantity (1) - number of repetitions of high-pass filters (HPF).M

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Input - Filtered channel (from 0 to (number of channels - 1)).
- Freq. LPF - Set the cutoff frequency of the low-pass filter (from 0.01 to half the sampling frequency of the filtered channel), Hz.
- Freq. HPF - Set the cutoff frequency of the high-pass filter (from 0.01 to half the sampling frequency of the filtered channel), Hz.

➢**Custom common properties:**

- Are available by the link below. 154

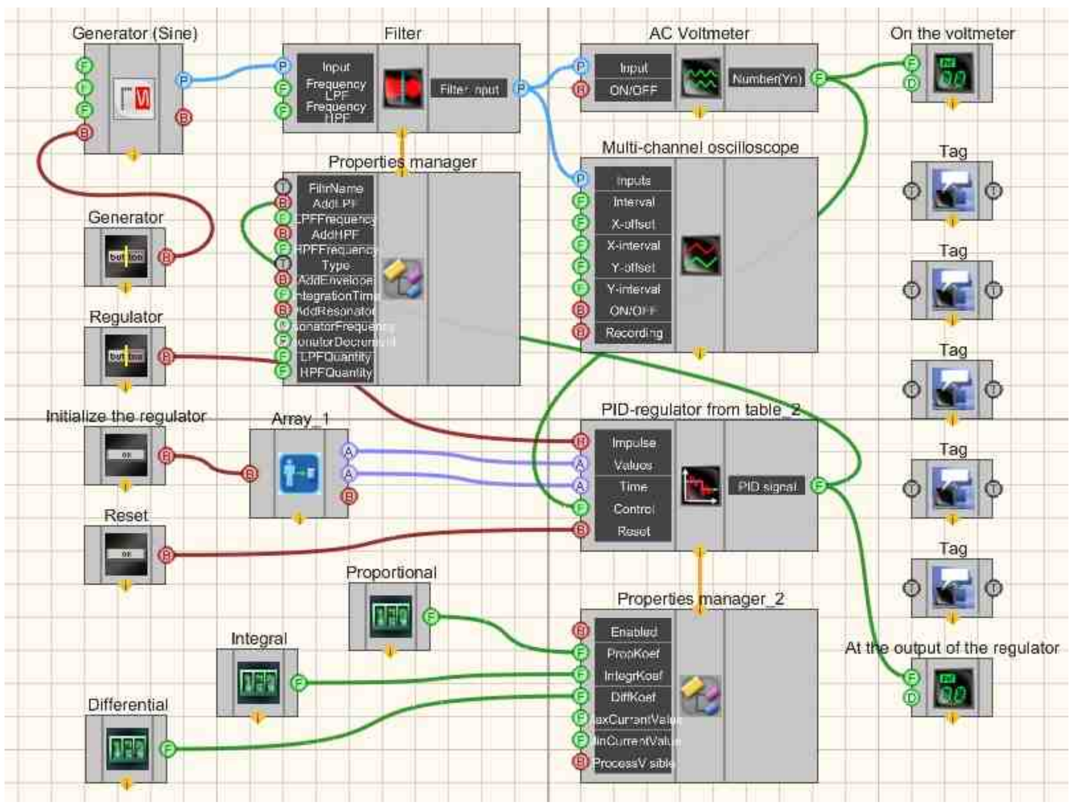
➢**Custom private properties (the default value is shown in parentheses):**

- FiltrName - Set the filter channel name (string).
- VARIANT_BOOL AddLPF - Set the low-pass filter operation status:
  true - the low pass filter is turned On.
  false - the low pass filter is turned Off.
- FLOAT LPFFrequency - Set the cutoff frequency of the low-pass filter (from 0.01 to half the sampling frequency of the filtered channel), Hz.
- VARIANT_BOOL AddHPF - Set the operation status of the high-pass filter:
  true - the high pass filter is turned On;
  false - the high pass filter is turned Off.
- FLOAT HPFFrequency - Set the cutoff frequency of the high-pass filter (from 0.01 to half the sampling frequency of the filtered channel), Hz.
- BSTR Type - Set the filter type:
  1. Linear - filter without integration and differentiation;
  2. Differentiating 1 - differentiating filter of the 1st order;
  3. Differentiating 2 - differentiating filter of the 2nd order;
  4. Integrating 1 - integrating filter of the 1st order;
  5. Integrating 2 - integrating filter of the 2nd order.
- VARIANT_BOOL AddEnvelope - Set the envelope resolution:
  true - envelope resolution is turned On.
  false - envelope resolution is turned Off.
- FLOAT IntegrationTime - Set the integration time of the envelope detector (from (2000 / sampling frequency of the filtered channel) ms to (500000000 / sampling frequency of the filtered channel) ms).
- VARIANT_BOOL AddResonator - Set the resolution of a valid resonator:
  true - resolution of a valid resonator is turned On.
  false - the resolution of the actual resonator is turned Off.

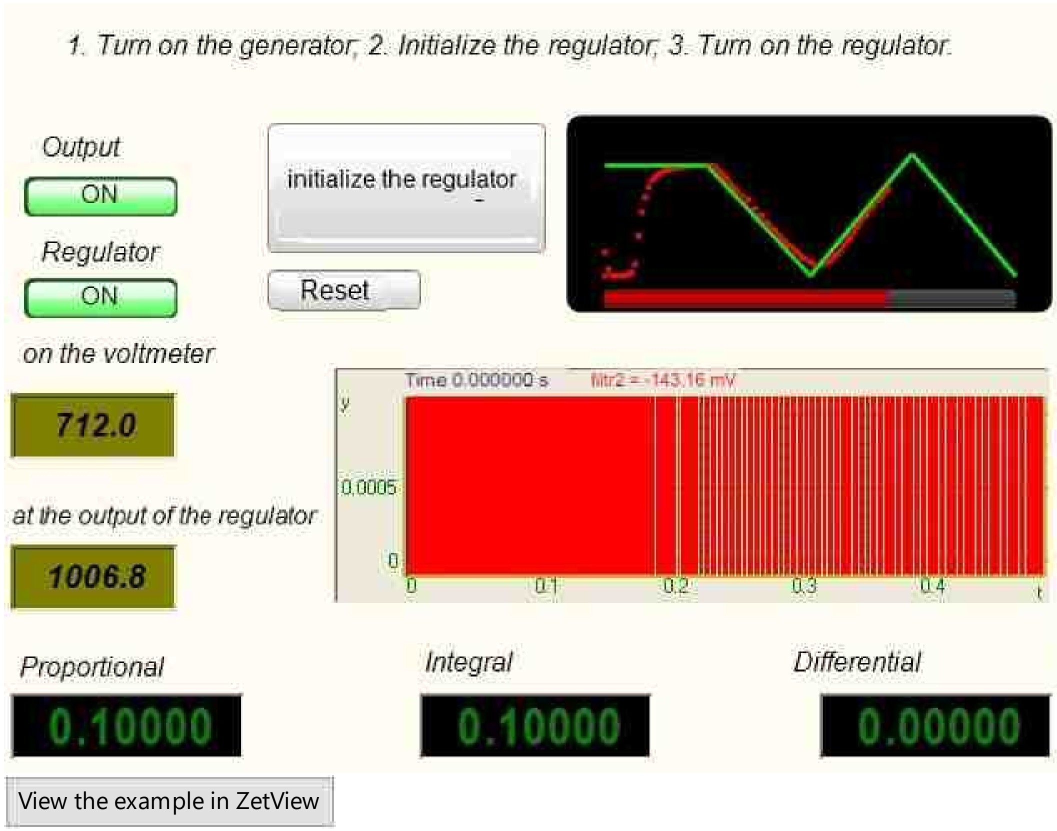
- FLOAT ResonatorFrequency - Set the resonance frequency (number).
- FLOAT ResonatorDecrement - Set the decay decrement (number).
- FLOAT LPFQuantity - Set the number of repeats of the LPF (number).
- FLOAT HPFQuantity - Set the number of HPF repeats (number).

Example

Project in SCADA ZETView



Project operation results



Глава 3.Signal analysis

The components **FFT Spectrum Analysis, Spectrum CPB Analysis, Cross-Correlation analysis, Cross-Spectrum FFT Analysis, Cross-Spectrum CPB (constant percentage bandwidth) analysis, Histogram, Modal analysis, STA/LTA detector, Harmonic Distortion Analysis** allow to start corresponding programs from ZETLAB scope, to set corresponding properties and to obtain results of programs operation.

The component High-frequency oscilloscope is used for signals processing of ZET 302 oscillograph. It starts ZETScope program and sets its properties.

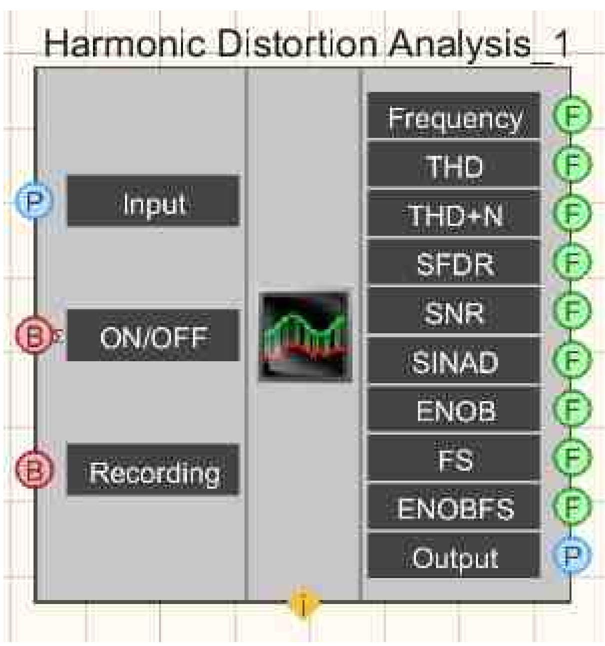
Thus, the programs from "Signal analysis" scope fully represent properties of the corresponding controlled programs: ***Harmonic Distortion Analysis, Spectrum CPB Analysis, Cross-Correlation analysis, Cross-Spectrum FFT Analysis, High-frequency oscilloscope(ZETScope.), Histogram, Damping rate of the oscillations, Earthquake detector, STA/LTA detector, Intensity of the earthquake on the MSK-64, Modal analysis,***

***Calculation of impedance, Event histogram, Super-resolution Spectrum, Spectrum CPB Analysis,* FFT *Spectrum Analysis*.**

3.1.Harmonic Distortion Analysis

The component "**Harmonic Distortion Analysis**" is designed for automatic measurement of the coefficient of nonlinear distortion and RMS voltage values of the studied signals entering the input channels of spectrum analyzers.
The software is used for automatic measurement of the harmonic distortion factor and voltage RMS. of signals coming to the input channels of FFT Spectrum Analysis. Adjustment to the frequency of the main component is performed automatically. The program is based on the measurement of the voltage r.m.s. of higher components by an external or internal reference signal.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - input measuring channel;
- ON/OFF - enabling/disabling the Harmonic Distortion Analysis channel;
- Recording - recording in a file.

➢ Output:

- Frequency - the frequency of the fundamental tone in the signal;
- THD - The total harmonic distortion (THD or THDi) is a measurement of the harmonic distortion present in a signal and is defined as the ratio of the sum of the powers of all harmonic components to the power of the fundamental frequency.
- THD +N - Total harmonic distortion plus noise (THD+N) is a common metric in audio interface specifications.
- SFDR - Spurious-free dynamic range (SFDR) is the strength ratio of the fundamental signal to the strongest spurious signal in the output;
- SNR - Signal-to-noise ratio (SNR or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise;
- SINAD - Signal-to-noise and distortion ratio (SINAD) is a measure of the quality of a signal from a communications device;
- ENOB - In electrical engineering ENOB is an abbreviation for the "Effective Number of Bits" of a DAC or ADC;
- FS - the remainder of the full range;
- ENOBFS - effective number of bits in the full range;
- Output - the Harmonic Distortion Analysis channel.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- AverageTime (1) - averaging time, s.
- freqrange (50000) - frequency range, Hz.
- calctype (linear) - the type of representation of the calculation of the harmonic distortion factor:
    1. linear scale (percentage);
    2. logarithmic scale (in decibels).
- FileName - the name of the files to record the results.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - the measuring channel (from 0 to (number of channels - 1)).

➤**Custom common properties:**

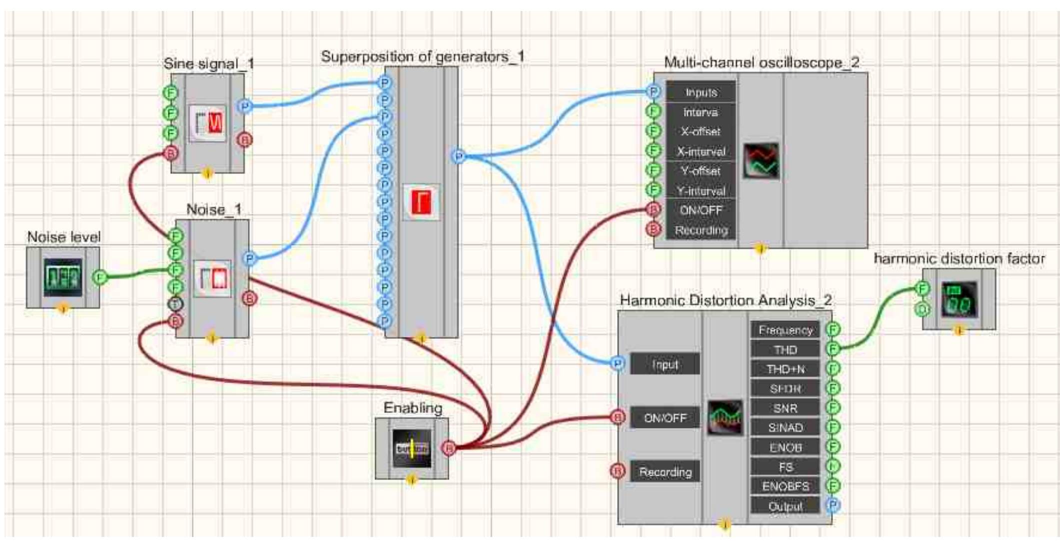
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

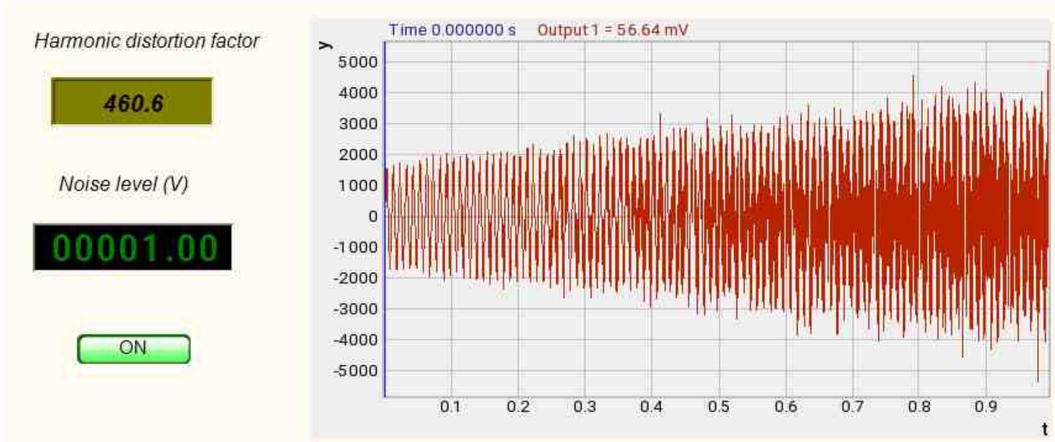
- FLOAT AverageTime - Set the averaging time, s (from 0.1 seconds to 10 seconds).
- BSTR FreqRange - Set the frequency range, Hz:
  0 – from 0 to (sampling frequency / 2);
  1 – from 0 to (sampling frequency / 20);
  2 – from 0 to(sampling frequency / 200);
  3 – from 0 to (sampling frequency / 2000);
  4 – from 0 to (sampling frequency / 20000).
- BSTR CalcType - Set the calculation representation: Linear; Logarithmic, dB (string).
- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).

Example

Project in SCADA ZETView



Project operation results

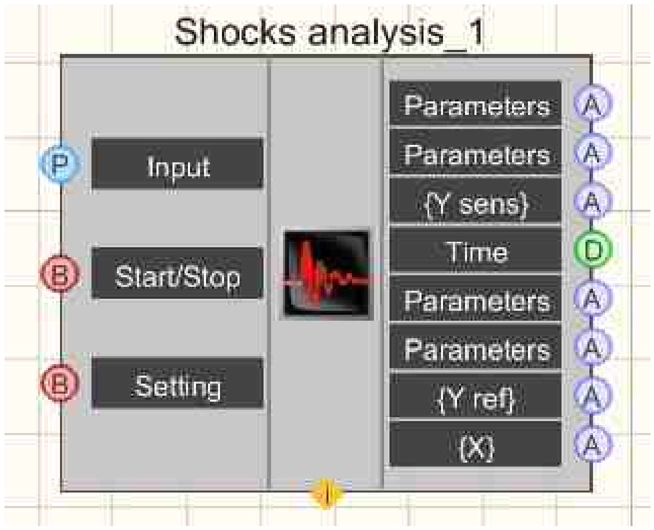


View the example in ZetView

3.2.Shocks analysis

The component **Shocks analysis**. The Shocks analysis 180 component (ZetShockAnalysis.ocx, hereinafter referred to as the "component") is an intermediate link between the ShockAnalysis.exe program of the same name (hereinafter referred to as the "program") and the SCADA project (hereinafter referred to as the "project"). The component is designed to transfer settings to the program and receive work results from the program and transfer these results to the project. Therefore, you must first familiarize yourself with the description of the program. At the development stage, one or more instances of the component are included in the SCADA project. One copy for each measuring channel.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- "Input" ("Measuring channel") - allows you to set and change the channel (both at the design stage of the project and during the work of the project).allows you to set and change the channel (both at the design stage of the project and during the work of the project).
- "Start/Stop" ("Start/Stop the strike search") - allows you to programmatically click on the "Start" button of the main window of the run program.
- "Settings" ("Launching the program settings window") - allows you to display the program settings window during the execution of the project to familiarize yourself with the real values of the parameters of the run program. **At the** same time, the operator will not be able to change these values, because the program settings window will work in measurement mode, i.e. with the "Apply" button unavailable.

.

➢ Output:

- "Parameters" ("Values of shock parameters (measuring channel)") - an array of values for the impact parameters of the measuring channel, see the channel column in the table of the main program window.
- "Parameters" ("Dimensions of impact parameters (measuring channel)") - an array of dimensions of the impact parameters of the measuring channel, see the channel column in the table of the main program window.
- "{Y sens}" ("Waveform{Y} (measuring channel)") - the values of the oscillogram of the measuring channel displayed in the grid of measuring channel on the main program window.
- "Time" ("Time in DATE format") is the absolute time of the shock front of the reference channel (DATE format is a double number, the integer part of which is the integer number of days that have passed since 00:00: 00 01.01.1970, the fractional part is the elapsed part of the day.

- "Parameters" ("Values of impact parameters (reference channel)") - an array of values of impact parameters of the reference channel, see the channel column in the table of the main program window.
- "Parameters" ("Dimensions of impact parameters (reference channel)") - an array of dimensions of measuring impact parameters, see the channel column in the table of the main program window.
- "{Y ref}" ("Waveform{Y} (reference channel)") - the values of the reference channel waveform displayed in the reference channel grid on the main program window.
- "{X}" ("Waveform{X} (all channels)") - time values of the waveforms of the reference and measuring channels displayed in the main program window.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- IntervalView (100) - display interval, ms.
- ValueAbsoluteThreshold (100) - value of the absolute threshold.
- FreqFindReson_F1 (0) - frequency range of resonance search (F1), Hz.
- FreqFindReson_F2 (0) - frequency range of resonance search (F2), Hz.
- BOOL Win_H1 (False) - displaying the dynamic characteristics calculation window.
  True - display of the dynamic characteristics calculation window is enabled;
  False - display of the dynamic characteristics calculation window is disabled.

The first component, see Figure 1, has three input contacts and eight output contacts. The component allows you to set the first measuring channel, as well as set values for all common program parameters, i.e. parameters that do not depend on channels. This is implemented by selecting the "Edit component" position in the context menu of the component. The menu appears by right-clicking on the component with its preliminary selection, see Figure 2. After selecting this position, the modal window "Settings" will be displayed, exactly the same as the program. The only exception is that the "Select of measuring channels" element will work in single channel selection mode, i.e. only the first measuring channel can be set.

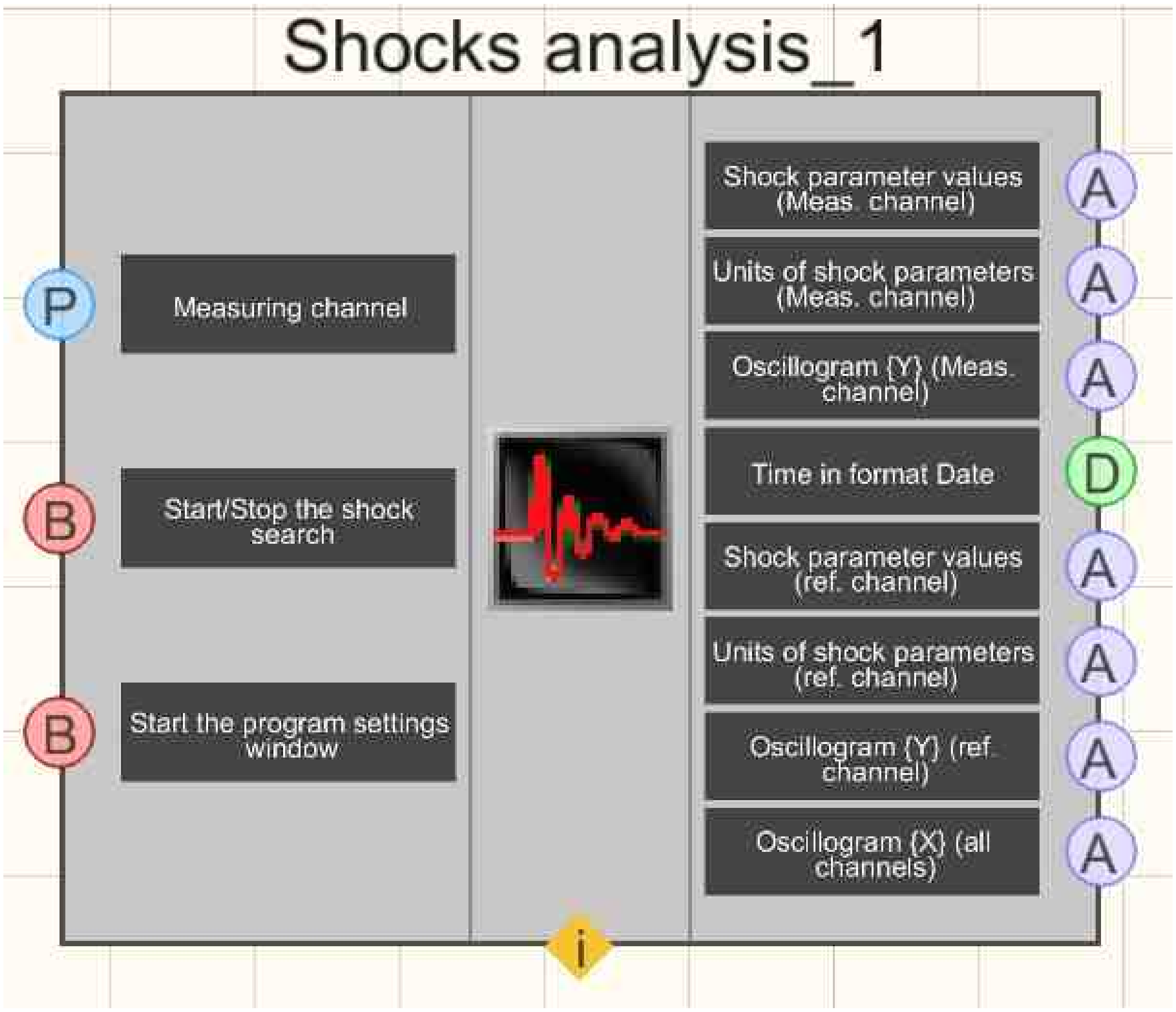

Shocks analysis_1
P
Measuring channel
B
Start/Stop the shock search
B
Start the program settings window
Shock parameter values (Meas. channel)
A
Units of shock parameters (Meas. channel)
A
Oscillogram {Y} (Meas. channel)
A
Time in format Date
D
Shock parameter values (ref. channel)
A
Units of shock parameters (ref. channel)
A
Oscillogram {Y} (ref. channel)
A
Oscillogram {X} (all channels)
A
i

Figure 1

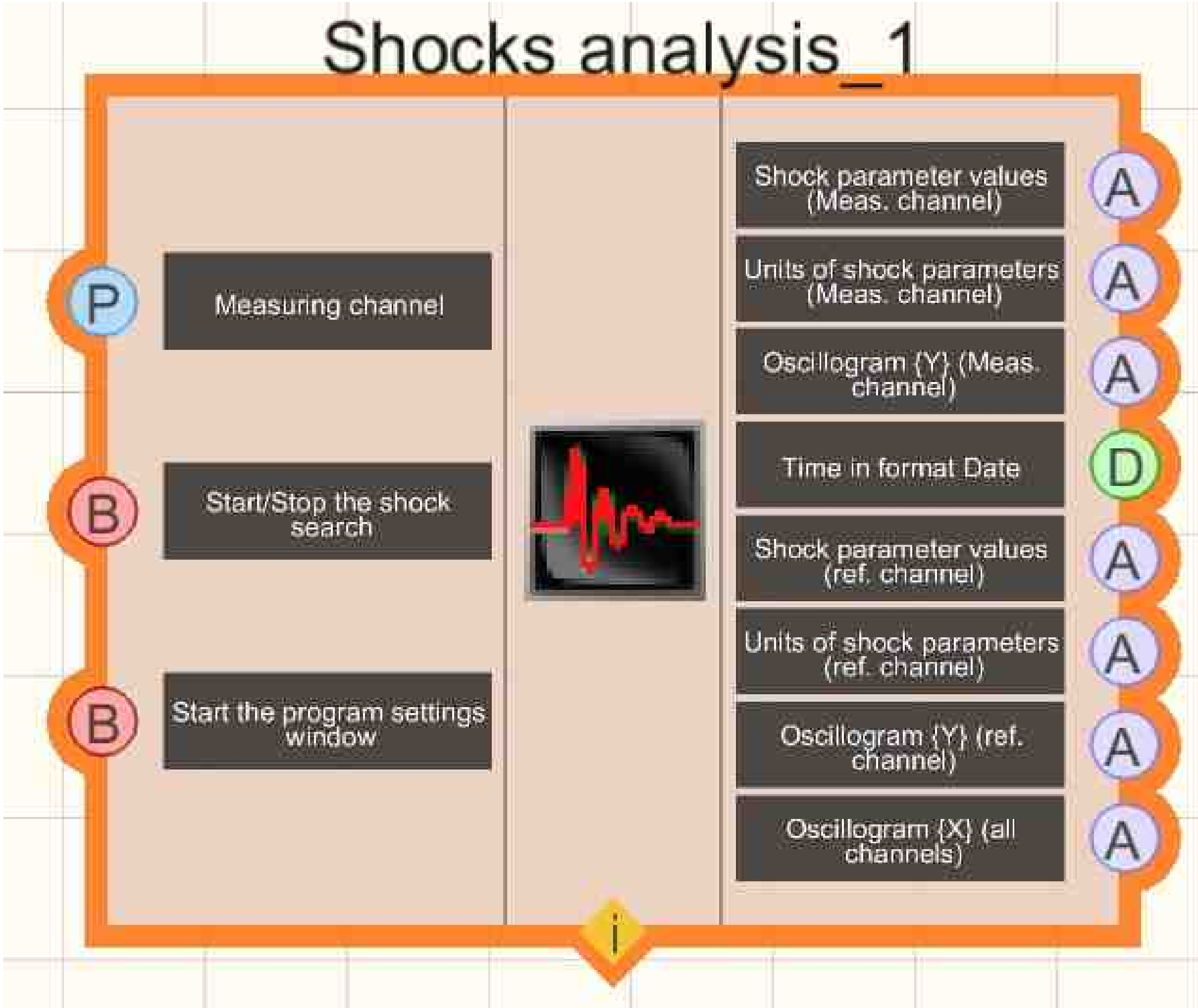


Figure 2

The second and all subsequent components (see Figure 3) have fewer contacts. If you take the contacts of the first component and leave only those connected to the measuring channel, then the contacts of the second component will remain, namely: the input contact "Channel" and three output contacts of the measuring channel. Thus, the number of contacts of the first component is greater than that of the other components. "Additional" contacts are contacts for data transmission, which are common to all components. Therefore, only the component that always has such contacts, i.e. the first one. If you remove the first component, then additional contacts will appear in the second, which will become the first in the new numbering. However, the links already created with the deleted contacts will also be deleted. When editing more than one component using the context menu, keep in mind that it is possible to set the measuring channel only for this component. General parameters can be set and/or changed in the settings window of any component.

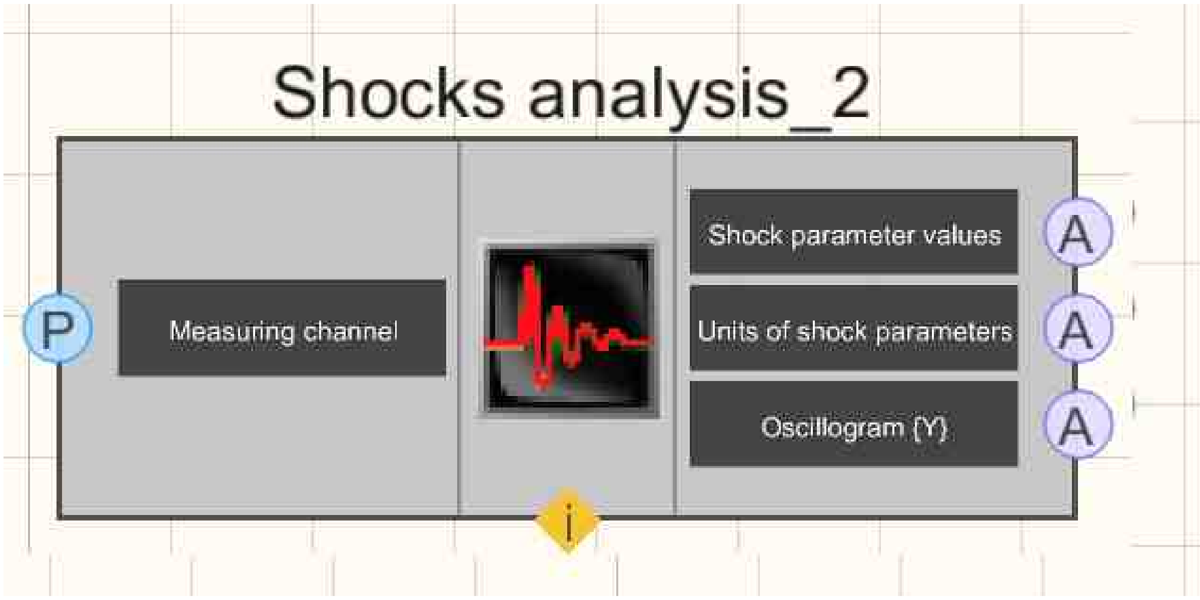


Figure 3

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - measuring channel (from 0 to (number of channels - 1)).

➢**Custom common properties:**

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

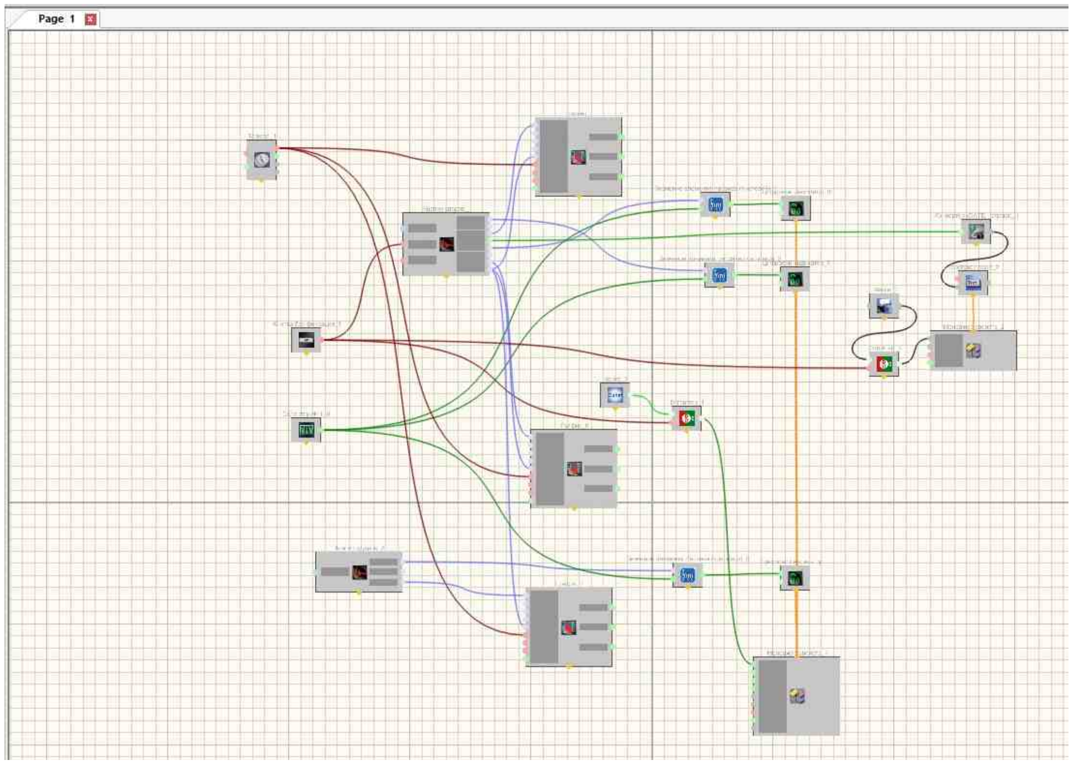
- LONG IntervalView (100) - Set the display interval, ms.
- FLOAT ValueAbsoluteThreshold (100) - Set the value of the absolute threshold.
- FLOATFreqFindReson_F1 (0) - Set the frequency range of resonance search (F1), Hz.
- FLOATFreqFindReson_F2 (0) - Set the frequency range of resonance search (F2), Hz.
- VARIANT_BOOL Win_H1 (False) - Set the displaying the dynamic characteristics calculation window.

  True - display of the dynamic characteristics calculation window is enabled;

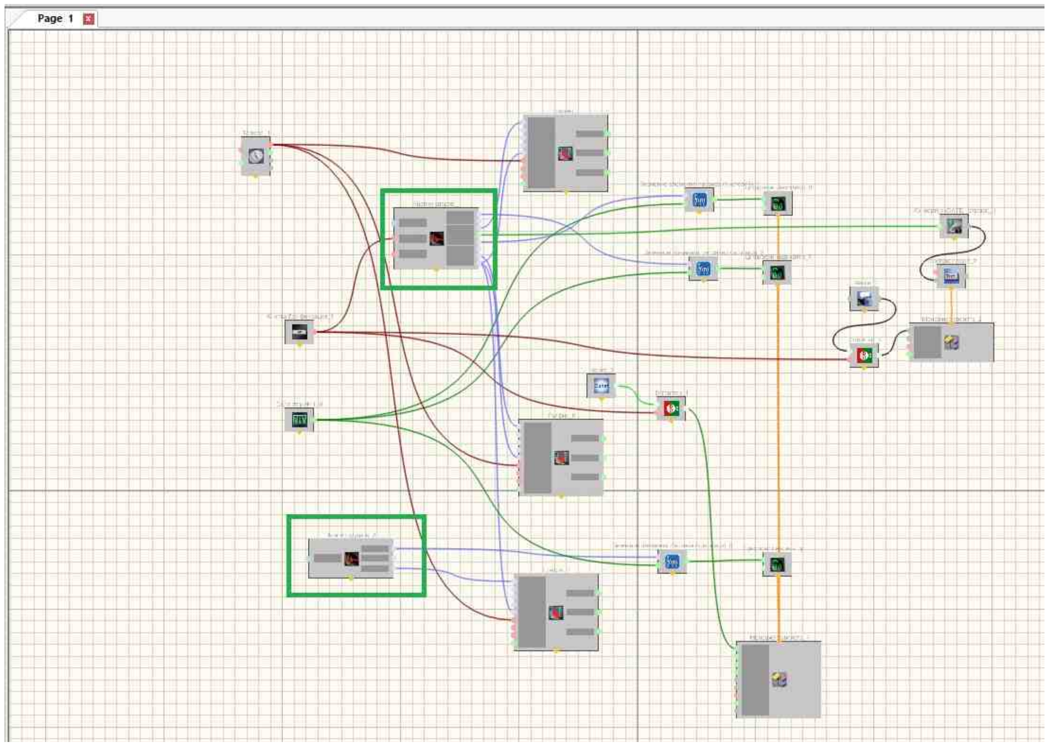
False - display of the dynamic characteristics calculation window is disabled.

Example

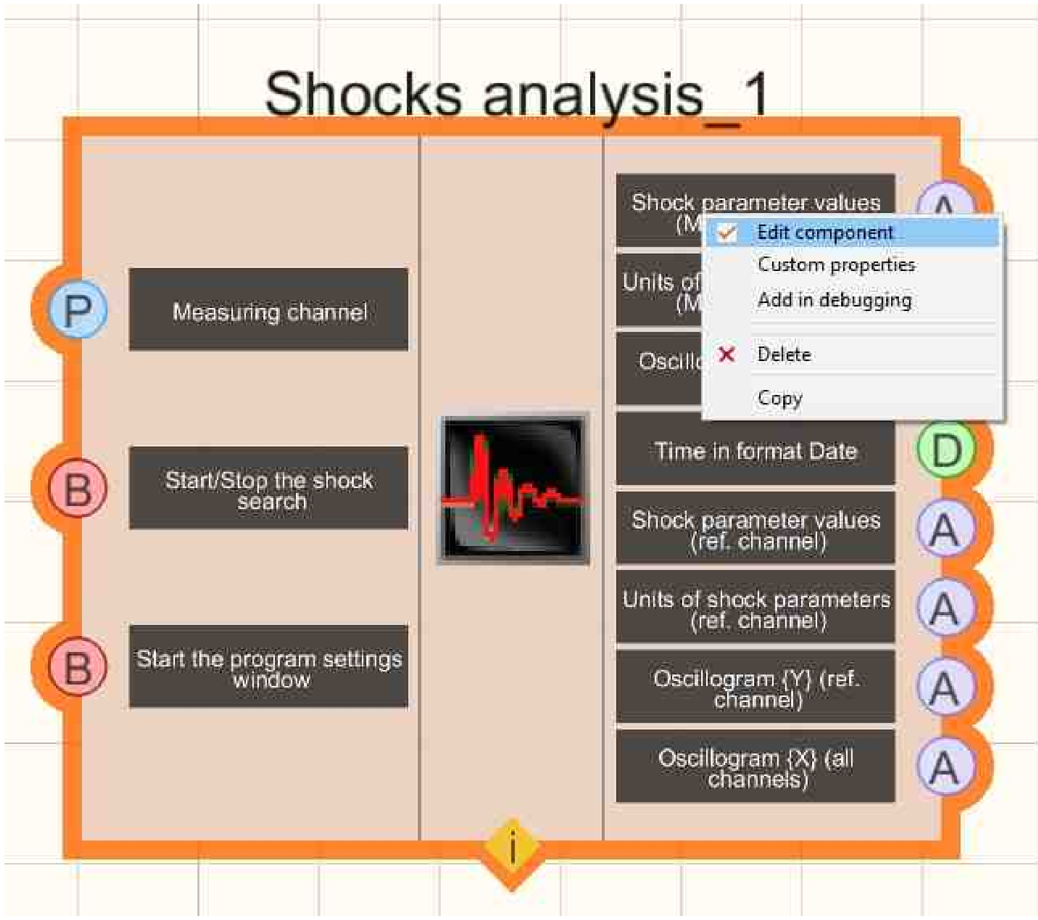
Project in SCADA ZETView



In this project, the Shocks analysis 180 component allows us to get registered shocks. The example uses 2 components of the Shocks analysis 180 program



For the components to work properly, they need to be configured. Select the desired component of Shocks analysis 180, select it and click the right mouse button. Select Edit component.



The program setting appears

Be sure to set up the measuring and reference channels in the Inputls tab.
We configure the component correctly and click "Apply", when the project is running or the program generated from the project, the "Apply" button is not available. All settings need to be changed only in the project to get unambiguous results.

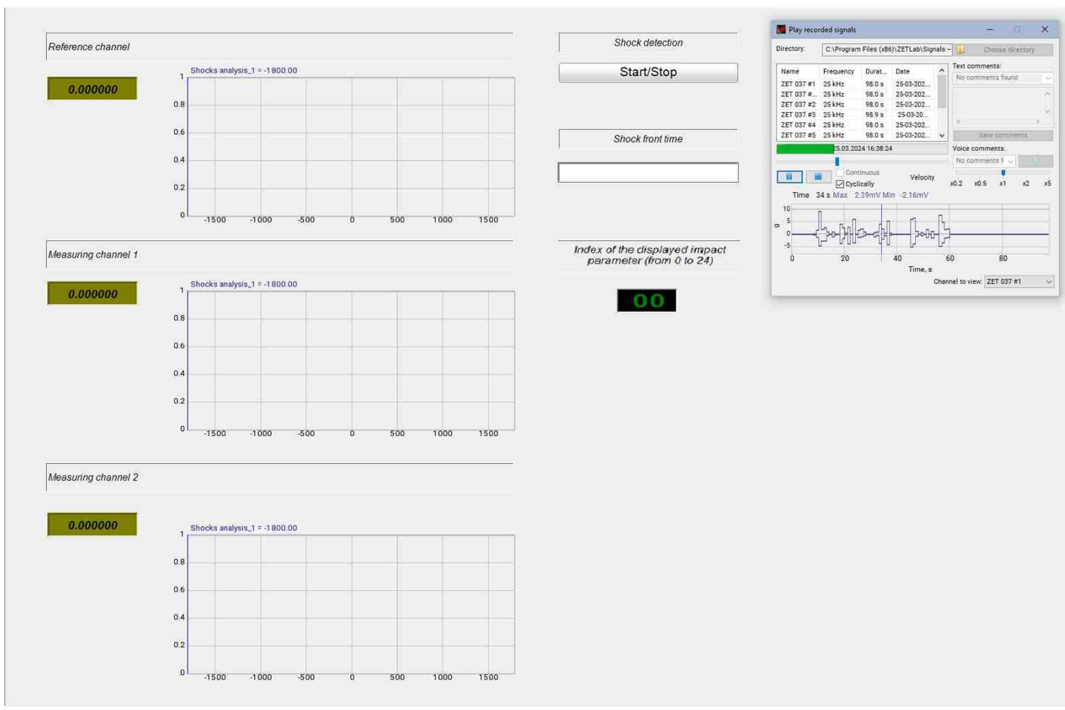
Select the settings of the Shocks tab. The remaining tabs of the program are configured as needed.

Project operation results

An example of using a component is a project from a file. "..\ZetLab\Help\ Analysis of signals - shock analysis.zvx", designed to work with two measuring channels. Before starting this project, it is necessary to configure the measuring channels of both components, as well as the general parameters. The following parameters are set for the components:

- display time interval = 3000 ms;
- the absolute threshold of operation;
- the value of the trigger threshold = 100.

The project has graphs for the waveforms of the reference channel and both measuring channels. To the left of each of the charts there is a digital indicator that displays the value of the impact parameter of this channel, which is taken from the table of the main window. The index of the displayed parameters is set using the selector, which is located to the right of the "Start" button/Stop." The text field displays the absolute time of the impact front of the reference channel, i.e. the time of the first countdown of the reference channel waveform, the value of which exceeded the specified threshold.



View the example in ZetView

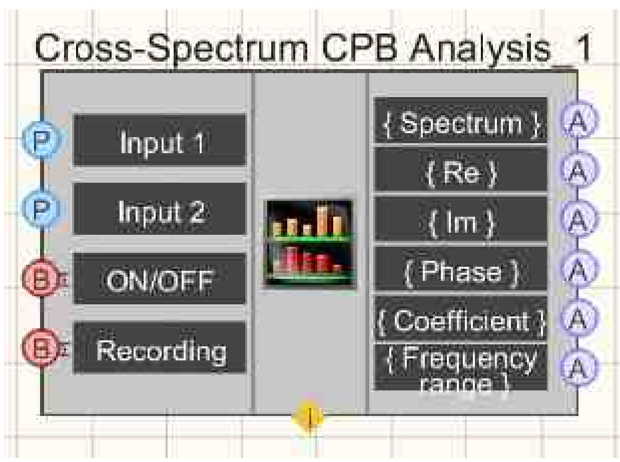
3.3.Cross-Spectrum CPB Analysis

The component "**Cross-Spectrum CPB (constant percentage bandwidth) analysis**" is used for transfer fractional octave (1/1, 1/3, 1/12, and 1/24 octave) spectral analysis of signals coming from the input channels of FFT Spectrum Analysis (in real time or recorded time realization view mode), as well as for viewing various spectral characteristics of signals.

Main software features

Measurement of signal levels in octave, 1/3, 1/12, 1/24-octave spectral bands. The number of bands is 17, 51, 204, 406, respectively. General technical requirements and methods of testing" for the first accuracy class. Octave-band and fractional-octave-band filters. Technical requirements and test methods";
Measurement and display of signal fractional-octave characteristics with various averaging (linear, exponential), processing (integration, signal differentiation) and view types (r.m.s. or crest factor values);
Measurement and display of the real and imaginary components, phase difference and signal coherence factor;
Measurement and display of the cross-spectrum module;
Measurement of the complex frequency response and spectral component coherent power.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input 1 - is the first measuring channel.
- Input 2 - is the second measuring channel
- On/Off - is turned On/Off the spectrum.
- Recording - when a impulse is applied to this contact, data is saved to a separate file (the name and location of the file are specified in the Custom private properties of the component).

➢ Output:

- {Spectrum} - the current instantaneous mutual fraction-octave spectrum.
- {Re} - is the real part of the Cross-Spectrum CPB Analysis.
- {Im} - is the imaginary part of the Cross-Spectrum CPB Analysis.

- {Phase} - phase of the Cross-Spectrum CPB Analysis.
- {Coefficient} is the coherence coefficient of the Cross-Spectrum CPB Analysis.
- {Frequency range} - is the frequency range of the Cross-Spectrum CPB (constant percentage bandwidth) analysis.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- type (1/3 octave) - type of analysis:
- 1. 1/1 octave;
  2. 1/3 octave;
  3. 1/12 octave;
  4. 1/24 octave.
- Average Time (1) - averaging time, s.
- spectrview (linear) - type of representation of the level of spectral components:
  1. Linear;
  2. logarithmic, dB.
- Activate (true) - the operation status.
- CalcRe (true) - the calculation of the real part of the spectrum.
- CalcIm (true) - the calculation of the imaginary part of the spectrum.
- CalcFase (true) - the calculation of the phase of the spectrum.
- CalcKoef (true) - the calculation of the spectrum coherence coefficient.
- FileName - the name of the file to record the results.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - the first measuring channel (from 0 to (number of channels - 1)).
- Input 2 - the second measuring channel (from 0 to (number of channels - 1)).

➢**Custom common properties:**

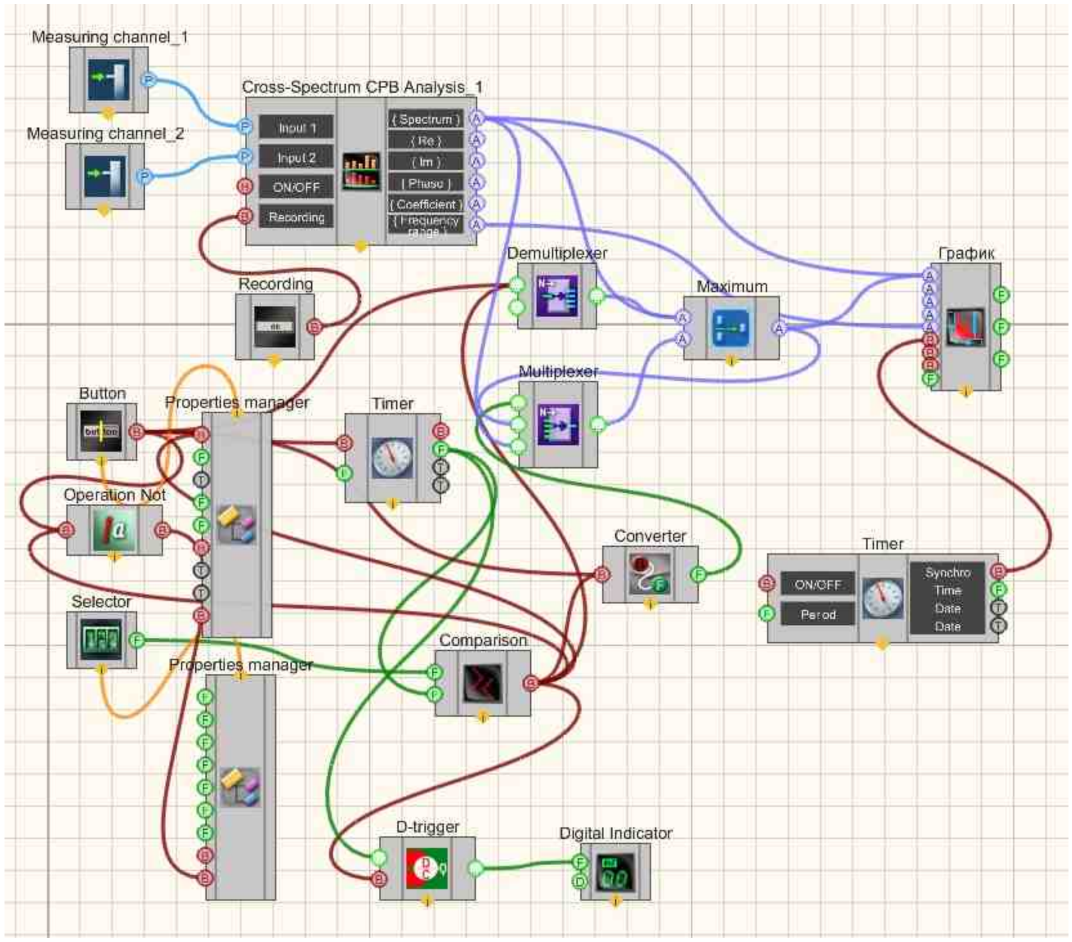
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

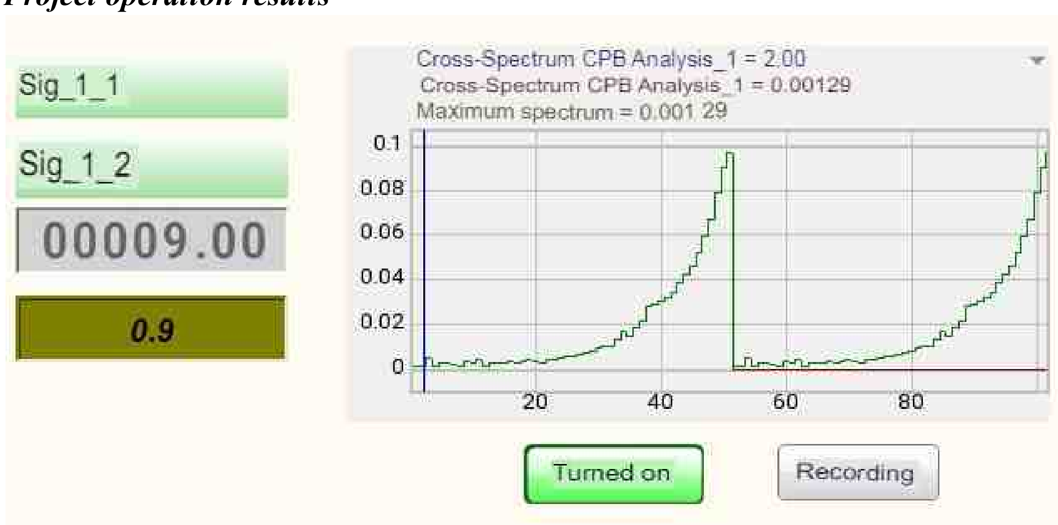
- BSTR Type - Set the analysis type:1/1-octave; 1/3-octave; 1/12-octave; 1/24-octave (string).
- FLOAT AverageTime - Set and read the averaging time, s (from 0.1 seconds to 100 seconds).
- BSTR SpectrView - Set the spectrum display: Linear; Logarithmic, dB (string).
- VARIANT_BOOL Activate - Set the operation status of the Cross-Spectrum CPB Analysis:
  - true - the operation status of the spectrum is turned On;
  - false - the operation status of the spectrum is turned Off;
- VARIANT_BOOL CalcRe - Set the calculation of the real part of the Cross-Spectrum CPB Analysis:
  - true - the calculation of the real part of the spectrum is turned On;
  - false - the calculation of the real part of the spectrum is turned Off;
- VARIANT_BOOL CalcIm - Set the calculation of the imaginary part of the Cross-Spectrum CPB Analysis:
  - true - the calculation of the imaginary part of the spectrum is turned On;
  - false - the calculation of the imaginary part of the spectrum is turned Off;
- VARIANT_BOOL CalcFase - Set the calculation of phase of the Cross-Spectrum CPB Analysis:
  - true - the calculation of phase of the spectrum is turned On;
  - false - the calculation of phase of the spectrum is turned Off.
- VARIANT_BOOL CalcKoef - Set the calculation of coherence coefficient of the Cross-Spectrum CPB Analysis:
  - true - the calculation of coherence coefficient of the spectrum is turned On;
  - false - the calculation of coherence coefficient of the spectrum is turned Off.
- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).

Example

Project in SCADA ZETView



Project operation results



View the example in ZetView

Sometimes the statistical analysis of signals requires the accumulation of data, and then the calculation of the maximum, minimum, average values of the values.
Examples of obtaining maximum, minimum, average spectra, correlograms, etc. are presented in the projects (on the Example of a **FFT Spectrum Analysis**).

Example No. 1 - calculating the maximum values:

View the example in ZetView

Example No. 2 - calculating the minimum values:

View the example in ZetView

Example No. 3 - calculating the average values:

View the example in ZetView

3.4.Cross-Correlation Analysis

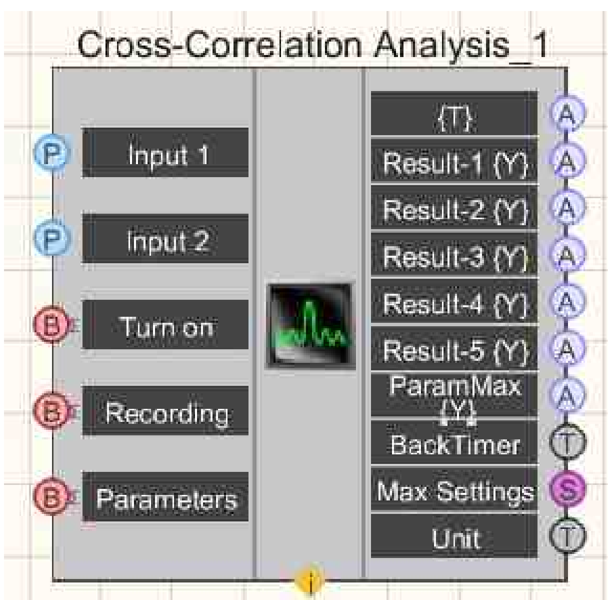
The component "**Cross-Correlation analysis**" is designed for mutual correlation analysis of signals coming from the input channels of spectrum analyzers to view various correlation characteristics of signals.

The software is used for correlation analysis of signals coming from the input channels of FFT Spectrum Analysis in real time or recorded time realization view mode, as well as for viewing various correlation characteristics of signals. **Cross-Correlation analysis** is a set of methods based on the mathematical correlation theory and is used for detecting the correlation dependence between two random attributes or factors. For solving a number of diagnostic tasks, cross-correlation analysis of signals in two or more control points distributed across space is often used.

Cross-Correlation analysis is often used for detecting a weak signal in the environment of strong static unbound interferences, as well as for determining the spatialce coordinates of their source based on the time shift value. It facilitates the detection of not only narrowband signals but also correlated random broadband signals, which is why it is successfully used, for instance, for searching leakage in pipelines based on noises occurring in the leakage area and spreading across the liquid (gas) flow.

It should be noted that spectral methods are used more often than the correlation methods for leakage detecting, since cross-spectral analysis of stationary processes provides more information (see "Cross-Spectrum FFT Analysis" and "Cross-Spectrum CPB Analysis"). But **Cross-Correlation analysis** is more effective in the spatial detection of non-stationary processes, including impulse ones, and can be used for detecting and determining the coordinates of acoustic emission sources in uniform loaded metallic structures.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - the first measuring channel.
- Input 2 - the second measuring channel.
- Turned On/turned Off - enables/disables correlation analysis.

- Recording - when a impulse is applied to this contact, data is saved to a separate file (the name and location of the file are specified in the **Custom common properties** of the component).
- Parameters - when a impulse is applied to this contact, the Settings of the program parameters are called, and if the settings are changed in these parameters, they will be applied, both in the program and the component, when the **Apply** button is pressed.

➢ Output:
- {T} - Array of delay values of the correlation function {Y}.
- Result-1 {Y} - array of values Result-1 {Y}.
- Result-2 {Y} - array of values Result-2 {Y}.
- Result-3 {Y} - array of values Result-3 {Y}.
- Result-4 {Y} - array of values Result-4 {Y}.
- Result-5 {Y} - array of values Result-5 {Y}.
- ParamMax {Y} - the updated value of the correlation peak time.
- BackTimer - the value of the remaining time.
- Max Settings - the value of the maximum settings.
- {Unit} - the unit of the result.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- AverageTime (1) - averaging, s.
- length (4096 points) - the size of the correlation function:1024 points; 2048 points;4096 points;8192 points;16384 points;32768 points;65536 points;131072 points;262144 points.
- Filter (false) - turned On/Off the Band-pass filter.
- FreqLPF (0) - the low-pass filter cutoff frequencies, Hz.
- FreqHPF (0) - the high-pass filter cutoff frequencies, Hz.
- FilterDiscret (false) - turned On/Off filtering of discrete interference.
- Inversion (false) - turned On/Off inversion.
- ModeDC (With DC) - turned On/Off the operation mode with a constant component of signals:
  - Remove DC (remove the constant component);
  - Filtering DC (constant component filtering);
  - With DC (with a constant component).
- Activate (false) - the status of the correlation analysis.
- FileName - the name of the files to record the results.
- Decimation (1) - decimation of the original signal: 1; 10.
- CalcDistance (false) - Calculation of distances.
- TypeView (Correlation) - Set the display: Correlation; Envelope; Plausible; Normalized function; Compensation of noise; Compensation auto; Correlation coefficient; Noise suppression 2 channel; Noise suppression.

- ShiftFunc (0.0) - Set the mixing in the calculations of normalized and difference functions.
- DistanceChannel1 (0.0) - Set the distance of the sensor channel 1, m.
- DistanceChannel2 (0.0) - Set the distance of the sensor channel 2, m.
- InterChannelDelay (10.0) - Set inter-channel delay, ms.
- TypeAver (Linear) - Set the averaging type:Linear;Exponential.
- Decrease (1) - Set the size reduction ratio, times: 1; 2; 4; 8; 16.
- X_left - Set the left border.
- X_rigth - Set the right border.
- AdjustmentWindow - Set the use of the correction window.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input 1 - the first measuring channel (from 0 to (number of channels - 1)).
- Input 2 - the second measuring channel (from 0 to (number of channels - 1)).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

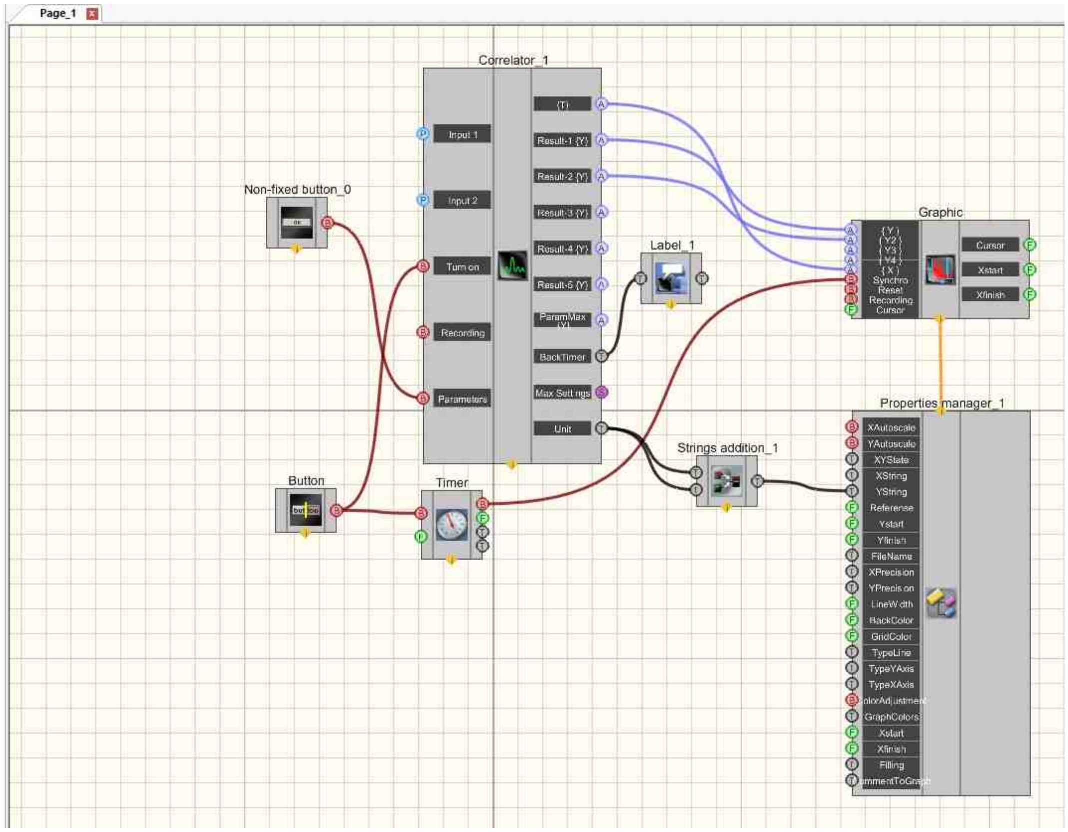
- FLOAT AverageTime - Set the averaging time, s (from 0.1 seconds to 100 seconds).
- BSTR Length - Set of the correlation function: 128 points; 256 points; 512 points; 1024 points; 2048 points; 4096 points; 8192 points; 16384 points; 32768 points; 65536 points; 6536 points;
- VARIANT_BOOL Filter - Set turned On/Off the Band-pass filter:
  true - the band-pass filter is turned On;
  false - The band-pass filter is turned Off.
- FLOAT FreqLPF - Set the cutoff frequency of the low-pass filter, Hz (from the cutoff frequency of the high-pass filter to (sampling frequency / 2)).
- FLOAT FreqHPF - Set the cutoff frequency of the high-pass filter, Hz (from 0 to the cutoff frequency of the low-pass filter).
- VARIANT_BOOL FilterDiscret - Set for filtering discrete interference (interference in the form of tonal stationary signals):
  true - the discret filter is turned On;
  false - the discret filter is turned Off.
- VARIANT_BOOL Inversion - Set enabling signal inversion:

true - the invert signal is turned On;
false - the invert signal is turned Off.

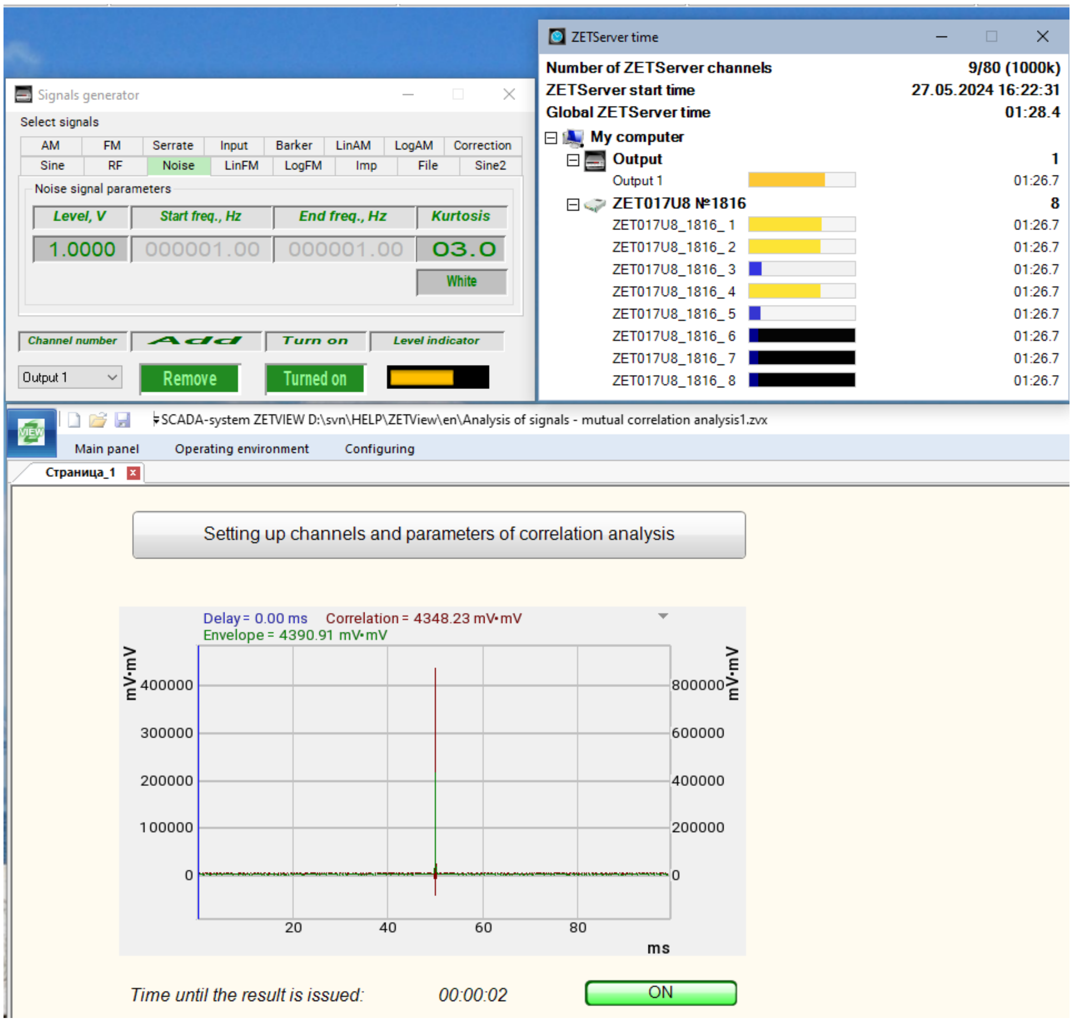
- BSTR ModeDC - Set the mode of operation with a constant component of signals: Remove DC; Filtering DC ;With DC (string).
- VARIANT_BOOL Activate - Set the operation status of the Cross-Correlation analysis:
  true - the operation status is turned On;
  false - the operation status is turned Off.
- VARIANT_BOOL Envelope - Set the data of the envelope correlation function.
- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).
- BSTR Decimation - Set the decimation of the original signal: 1; 10 (string).
- VARIANT_BOOL CalcDistance - Set distance calculation:
  true - Distance calculation is turned On;
  false - Distance calculation is turned Off.
- BSTR TypeView (Correlation) - Set the display: Correlation; Envelope; Plausible; Normalized function; Compensation of noise; Compensation auto; Correlation coefficient; Noise suppression 2 channel; Noise suppression.
- FLOAT ShiftFunc (0.0) - Set the mixing in the calculations of normalized and difference functions.
- FLOAT DistanceChannel1 (0.0) - Set the distance of the channel sensor1, m.
- FLOAT DistanceChannel2 (0.0) - Set the distance of the channel sensor 2, m.
- FLOAT InterChannelDelay (10.0) - Set inter-channel delay, ms.
- BSTR TypeAver (10.0) - Set the averaging type: linear; exponential (string).
- BSTR Decrease - Set size reduction, times: 1; 2; 4; 8; 16 (string).
- BSTR X_left - Set the left border (string).
- BSTR X_rigth - Set the right border (string).
- VARIANT_BOOL AdjustmentWindow - Set the use of the correction window.
  true - the use of an adjustment window is turned On;
  false - the use of an adjustment window is turned Off.

Example

Project in SCADA ZETView



Project operation results



View the example in ZetView

3.5.Cross-Spectrum FFT Analysis

The component "**Cross-Spectrum FFT Analysis**" is designed for simultaneous (cross) display of several Cross-Spectrum FFT Analysis. The instantaneous Cross-Spectrum is calculated by complex conjugate multiplication of the spectra of two channels obtained using the Fourier transform.

The program **Cross-Spectrum FFT Analysis** is used for evaluation of the interrelation between the signals' parameters obtained from the two primary transducers installed at various parts of the controlled object. This program can be used for detection of noise source location, for sound absorption level evaluation and the researched object acoustic properties control, for evaluation of space-time distribution of the directional energy flux (Poynting vector), for the ground cross-section FR characteristic evaluation (Nakamura method), etc.

The program **Cross-Spectrum FFT Analysis** is used for calculation of power cross-spectrum, phase spectra, coherence spectra, impulse-response characteristics, as well as for resonance calculation.

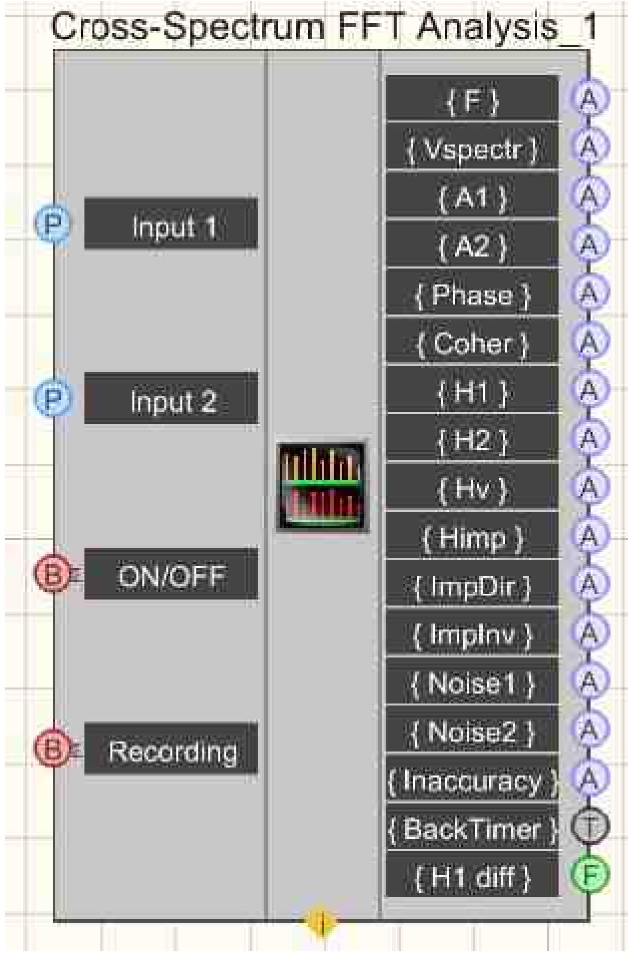
Cross-Spectrum FFT Analysis program enables graphical representation of various spectral characteristics of the signals, thus, allowing to reveal the enhanced vibration amplitudes at resonance frequencies of the researched object or the components of a complex system. These functions allow to timely reveal the defects and to undertake corresponding preventive measures.

Implementation of the **Cross-Spectrum FFT Analysis** of the signals received from the input channels of the **FFT Spectrum Analysis** is possible both in the real-time mode and the accumulated data post-processing mode.

Cross-Spectrum FFT Analysis belongs to the group of classical signal analysis methods and is widely applicable for almost all classes of signals having stationary properties.

Appearance of the component:

Developer interface	Operator interface

<table>
<tr>
<td>
</td>
<td>Doesn't have</td>
</tr>
</table>

Setting:

➢Input:

- Input 1 - the first measuring channel.
- Input 2 - the second measuring channel.
- ON/OFF - enables/disables **Cross-Spectrum FFT Analysis**.
- Recording - when a impulse is applied to this contact, data is saved to a separate file (the name and location of the file are specified in the **Custom common properties** of the component).

➢ Output:

- {F} is an array of frequency range values.
- {Vspectr} is an array of values of the module of the **Cross-Spectrum FFT Analysis**.
- {A1} is an array of values of auto spectrum 1.
- {A2} is an array of values of auto-spectrum 2.
- {Phase} - an array of phases of the **Cross-Spectrum FFT Analysis**.
- {Coher} - array of coherence coefficient values
- {H1} - an array of values of the transition characteristic H1

- {H2} - an array of values of the transition characteristic H2
- {Hv} - an array of values of the transition characteristic Hv
- {Himp} - an array of values of the transition characteristic Himp
- {ImpDir} - direct impulse response
- {ImpInv} - reverse impulse response
- {Noise 1} - array of values of the spectral density of natural noise 1
- {Noise 2} - array of values of the spectral density of natural noise 2
- {Inaccuracy} - array of values for the error that occurs after the passage of 1/3 octave

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Decimation (1) - the decimation of the original signal:1;10;100;1000;10000.

- AverageTime (10) - averaging time, s.

- Analysistype (discrete Fourier Transform) - type of signal processing:
  1. Fast Fourier transform.
  2. Discrete Fourier transform.

- Functiontype (Sine window) - type of weight function:
  1. Rectangular.
  2. Hann.
  3. Hamming.
  4. Blackman.
  5. Bartlett.
  6. Kaiser.
  7. Reef-Vincent (4).
  8. Blackman-Harris (3).
  9. Blackman-Harris (4).
  10. Natalla.
  11. Blackman-Natalla.
  12. Flat-top window.
  13. Sine window.

- Calctype(without processing) - type of signal processing:
  1. Double differentiation (differentiation of the second order).
  2. Differentiation (differentiation of the first order).
  3. Unedited (without processing integration and differentiation).
  4. Integration (integration of the first order).
  5. Double integration (integration of the second order).

- AverageType (Exponencial) - type of accumulation and averaging mode.
  1. Linear.
  2. Exponential.
  3. Peak detector.

- FreqResolutionFFT (128; 256; 512; 1024; 2048; 4096; 8192; 16384; 32768;65536;131072;262144;524288;) - the number of bands at FFT.

- freqResolutionDFT (1) - frequency resolution at DFT.

- FiltrBandpass (false) - Band-pass filter operation resolution.
- FreqHpf (0) - The cutoff frequency of the high-pass filter, Hz.
- FreqLpf (0) - The cutoff frequency of the low-pass filter, Hz.

- Activate (false) - operation status.

- CalcVspectr (true) - the resolution for calculating the module of the Cross-Spectrum FFT Analysis.
- CalcA1(false) - resolution of calculation of the auto power spectrum of the first input.
- CalcA2 (false) - resolution of calculation of the auto power spectrum of the second input.
- CalcPhase (false) - the phase response calculation resolution.
- CalcCoher (false) - the resolution of coherence coefficient calculation.

- CalcH1 (false) - the resolution of transient response calculation H1.
- CalcH2 (false) - the resolution of transient response calculation H2.
- CalcHv (false) - the resolution of transient response calculation Hv.

- CalcHimp (false) - the resolution of transient response calculation Himp.
- CalcNoise (false) - the resolution of the calculation of the spectral density of own noise.
- CalcImpDirect (false) - the resolution of direct impulse response calculation.
- CalcImpInverse (false) - the resolution of the calculation of the inverse impulse response.
- CalcH1Diff (false) - the resolution of the calculation of parameters from the impulse response.

- ThirdOctave (false) - 1/3-octave calculation resolution.
- FileName - the name of the files to record the results
- ImpLengthFrom - the beginning of the impulse response, s.
- ImpLengthTo - the end of the impulse response, s.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - the first measuring channel (from 0 to (number of channels - 1)).
- Input 2 - the second measuring channel (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR Decimation - Set the decimation of the original signal: 1; 10; 100; 1000; 10000 (string).
- FLOAT AverageTime - Set averaging time, s.
- BSTR AnalysisType - Set signal processing type: Fast Fourier transform (FFT); Discrete Fourier transform (DFT) (string).
- BSTR FunctionType - Set the weight function type:Rectangular;Hann;Hamming;Blackman;Bartlett;Kaiser;Reef-Vincent (4);Blackman-Harris (3);Blackman-Harris (4);Natalla;Blackman-Natalla;Flat-top window;Sine window (string).
- BSTR CalcType - Set types of signal processing:Double differentiation; Differentiation; Unedited; Integration; Double integration; (string).
- BSTR AverageType - Set the type of accumulation and averaging mode: Linear; Exponential; Peak detector (string).
- BSTR FreqResolutionFFT - Set the number of bands in the FFT: 128; 256; 512; 1024; 2048; 4096; 8192; 16384; 32768; 65536; 131072; 262144; 524288 (string).
- FLOAT FreqResolutionDFT - Set frequency resolution in DFT (string).
- VARIANT_BOOL FiltrBandpass - Set the band-pass filter enable:
  true - Band-pass filter is turned On;
  false - Band pass filter is turned Off.
- FLOAT FreqHpf - Set the cutoff frequency of the high-pass filter (string).
- FLOAT FreqLpf - Set the cutoff frequency of the low-pass filter (string).
- VARIANT_BOOL Activate - Set the operation status of the Cross-Spectrum FFT Analysis:
  true - the operation status of the Cross-Spectrum FFT Analysis is turned On;
  false - the operation status of the Cross-Spectrum FFT Analysis is turned Off.
- VARIANT_BOOL CalcVspectr - Set the resolution of the calculation of the module of the Cross-Spectrum FFT Analysis
  true - the calculation of the modulus of the Cross-Spectrum FFT Analysis is turned On;
  false - the calculation of the modulus of the Cross-Spectrum FFT Analysis is turned Off;

- VARIANT_BOOL CalcA1 - Set the resolution of the calculation of the auto power spectrum of the first input:
  true - the calculation of the auto power spectrum of the first input is turned On;
  false - the calculation of the auto power spectrum of the first input is turned Off;
- VARIANT_BOOL CalcA2 - Set the resolution of the calculation of the auto power spectrum of the second input:
  true - the calculation of the auto power spectrum of the second input is turned On;
  false - the calculation of the auto power spectrum of the second input is turned Off;
- VARIANT_BOOL CalcPhase - Set the resolution of the phase response calculation:
  true - the resolution of the phase response calculation is turned On;
  false - the resolution of the phase response calculation is turned Off;
- VARIANT_BOOL CalcCoher - Set the calculation of the coherence coefficient:
  true - the calculation of the coherence coefficient is turned On;
  false - the calculation of the coherence coefficient is turned Off;
- VARIANT_BOOL CalcH1 - Set transient H1 calculation:
  true - the calculation of transient response H1 is turned On;
  false - the calculation of transient response H1 is turned Off;
- VARIANT_BOOL CalcH2 - Set the calculation of transient response H2:
  true - the calculation of transient response H2 is turned On;
  false - the calculation of transient response H2 is turned Off;
- VARIANT_BOOL CalcHv - Set the calculation of transient response Hv:
  true - the calculation of transient response Hv is turned On;
  false - the calculation of transient response Hv is turned Off;
- VARIANT_BOOL CalcHimp - Set the calculation of transient response Himp:
  true - the calculation of transient response Himp is turned On;
  false - the calculation of transient response Himp is turned Off;
- VARIANT_BOOL CalcNoise - Set the calculation of the spectral density of own noise:
  true - the calculation of the spectral density of own noise is turned On;
  false - the calculation of the spectral density of own noise is turned Off;
- VARIANT_BOOL CalcImpDirec - Set the resolution of the direct impulse response calculation:
  true - the resolution of the direct impulse response calculation is turned On;
  false - the resolution of the direct impulse response calculation is turned Off;
- VARIANT_BOOL CalcImpInverse - Set the resolution of the calculation of the inverse impulse response:
  true - the resolution of the calculation of the inverse impulse response is turned On;
  false - the resolution of the calculation of the inverse impulse response is turned Off.
- VARIANT_BOOL CalcH1Diff - Set the resolution of the calculation of parameters from the impulse response:
  true - the calculation of parameters from the impulse response is turned On;
  false - the calculation of parameters from the impulse response is turned Off.
- VARIANT_BOOL ThirdOctave - Set calculation 1/3-octave:
  true - the calculation 1/3-octave is turned On;
  false - the calculation 1/3-octave is turned Off.
- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).

VARIANT_BOOL ImpLengthFrom - Set the beginning of the impulse response, s:

- true - the beginning of the impulse response is turned On;
  false - the beginning of the impulse response is turned Off.

VARIANT_BOOL ImpLengthTo - Set the end of the impulse response, s:

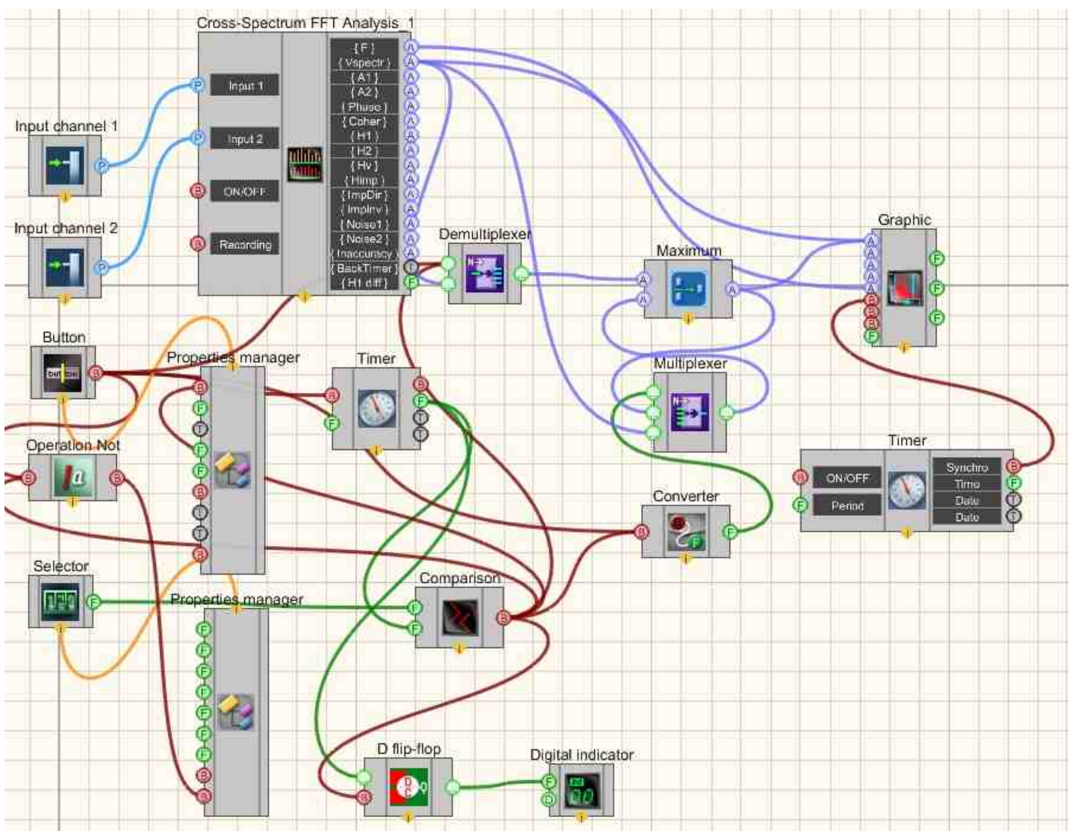
- true - the end of the impulse response is turned On;
  false - the end of the impulse response is turned Off.

Sometimes the statistical analysis of signals requires the accumulation of data, and then the calculation of the maximum value of the quantities.

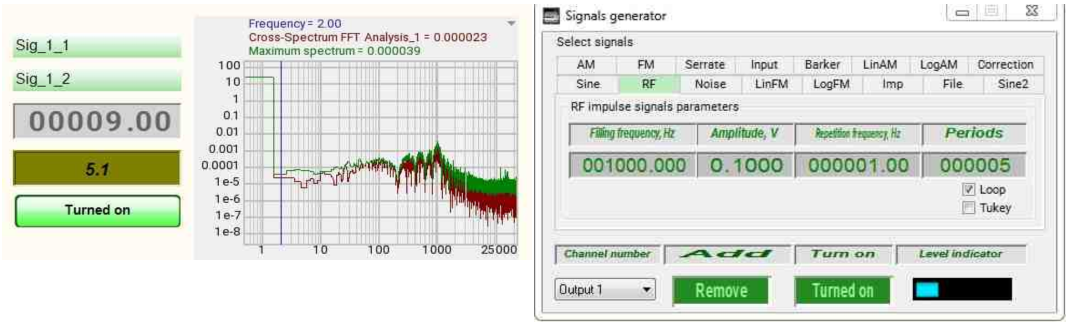
An example of obtaining the maximum spectrum is presented in the project.

Example - calculation of maximum values:

Project in SCADA ZETView



Project operation results

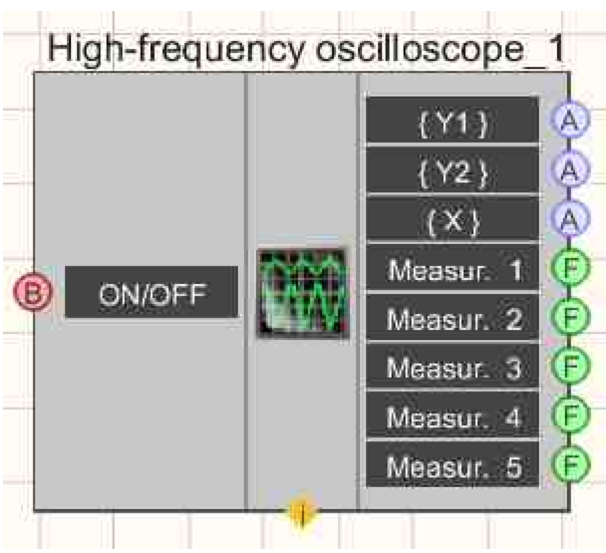


View the example in ZetView

3.6.High-frequency oscilloscope

The component "**High-frequency oscilloscope**" is designed for monitoring high-frequency signals with a frequency of up to 20 MHz.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- ON/OFF - turn on/off the oscilloscope.

➢ Output:
- {Y1} - output array of the first channel
- {Y2} - output array of the second channel
- {X} - dimension of array;
- Measur. 1 - value of the first measured parameter;
- Measur. 2 - value of the second measured parameter;
- Measur. 3 - value of the third measured parameter;
- Measur. 4 - value of the fourth measured parameter;
- Measur. 5 - value of the fifth measured parameter;

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Activate (true) - the operation status of high frequency oscilloscope.
- FirstInput (true) - the status of the first channel.
- SecondInput (true) - the status of the second channel.
- firstresolution (5) - the resolution on the first channel, V/cell: 0.002;0.005;0.01;0.02;0.05;0.1;0.2;0.5;1;2;5.
- secondresolution (5) - the resolution on the second channel resolution, V/cell:0.002;0.005;0.01;0.02;0.05;0.1;0.2;0.5;1;2;5.
- firstinputtype (DC) - input type of the first channel: DC;AC.
- secondinputtype (DC) - input type of the second channel: DC;AC.
- firstinputprobe (1x) - first channel probe: 1x;10x.
- secondinputprobe (1x) - second channel probe: 1x;10x.
- FirstInversion (false) - data inversion of the first channel.
- SecondInversion (false) - data inversion of the second channel.
- synchroinput (First) - synchronization channel.
- synchrofront (Upward) - synchronization front: Upward;Downward.
- synchromode (Automatic) - synchronization mode: Automatic;Normal.
- SynchroLevel (0) - synchronization level, cells.
- timebase (1) - time synchronized, s/cell: 0.00000001; 0.000000025; 0.00000005; 0.0000001; 0.00000025; 0.0000005; 0.000001; 0.0000025; 0.000005; 0.00001; 0.000025; 0.00005; 0.0001; 0.00025; 0.0005; 0.001; 0.0025; 0.005; 0.01; 0.025; 0.05; 0.1; 0.25; 0.5; 1; 2.5; 5; 10; 25; 50.
- Show (true) - enable/disable display of the oscilloscope window.
- measerechannel1 (1) - measuring parameter channel: 1: 1; 2.
- measure1 (Frequency, Hz) - measuring parameter 1: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr.
- measurechannel2 (1) - measuring parameter channel 2: 1; 2.

- measure2 (Frequency, Hz) - measuring parameter 2: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr.
- measurechannel3 (1) - measuring parameter channel 3: 1; 2.
- measure3 (Frequency, Hz) - measuring parameter 3: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr.
- measurechannel4 (1) - measuring parameter channel 4: 1; 2.
- measure4 (Frequency, Hz) - measuring parameter 4: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr.
- measurechannel5 (1) - measuring parameter channel 5: 1; 2.
- measure5 (Frequency, Hz) - measuring parameter 5: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr.
- averagequantity (4) - number of averages: 4; 8; 16; 32; 64.
- FirstPosition (0) - position of the first channel, cells.
- SecondPosition (0) - position of the second channel, cells.
- TimePosition (0) - time zero position, cells.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL Activate - Set the operation status of the high-frequency oscilloscope:
- true - the operation status of the oscilloscope is On;
  false - the operation status of the oscilloscope is Off.

VARIANT_BOOL FirstInput - Set the operation status of the first channel:
- true - the operation status of the first channel is On;
  false - the operation status of the first channel is Off.

VARIANT_BOOL SecondInput - Set the operation status of the second channel:
- true - the operation status of the second channel is On;

false - the operation status of the second channel is Off.

- BSTR FirstResolution - Set resolution on the first channel, V/cell: 0.002; 0.005; 0.01; 0.02; 0.05; 0.1; 0.2; 0.5; 1; 2; 5 (string).
- BSTR SecondResolution - Set resolution on the second channel, V/cell: 0.002; 0.005; 0.01; 0.02; 0.05; 0.1; 0.2; 0.5; 1; 2; 5 (string).
- BSTR FirstInputType - Set the input type of the first channel: DC; AC (string).
- BSTR SecondInputType - Set the input type of the second channel: DC; AC (string).
- BSTR FirstInputProbe - Set the probe of the first channel: 1x; 10x (string).
- BSTR SecondInputProbe - Set the probe of the second channel: 1x; 10x (string).
- VARIANT_BOOL SecondInversion - Set data inversion of the second channel:
  true - Data inversion of the second channel is On;
  false - Data inversion of the second channel is Off.
- BSTR SynchroInput - Set the synchronization channel: First; Second (string).
- BSTR SynchroFront - Set the front of the synchronization: Upward; Downward (string).
- BSTR SynchroMode - Set the timing mode: Automatic; Normal (string).
- FLOAT SynchroLevel - Set the synchronization trigger level on the screen horizontally in cells relative to the center of the screen (from minus 5 to 5).
- BSTR TimeBase - Set the time synchronized, s / cell: 0.00000001; 0.000000025; 0.00000005; 0.0000001; 0.00000025; 0.0000005; 0.000001; 0.0000025; 0.000005; 0.00001; 0.000025; 0.00005; 0.0001; 0.00025; 0.0005; 0.001; 0.0025; 0.005; 0.01; 0.025; 0.05; 0.1; 0.25; 0.5; 1; 2.5; 5; 10; 25; 50 (string).
- VARIANT_BOOL Show - Set the display resolution of the oscilloscope window:
  true - Permission to display the oscilloscope window is On;
  false - Permission to display the oscilloscope window is Off.
- BSTR MeasureChannel1 - Set the channel of the measured parameter 1: 1; 2 (string).
- BSTR Measure1 - Set the channel of the measured parameter 1: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr (string).
- BSTR MeasureChannel2 - Set the channel of the measured parameter 2: 1; 2 (string).
- BSTR Measure2 - Set the channel of the measured parameter 2: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr (string).
- BSTR MeasureChannel3 - Set the channel of the measured parameter 3: 1; 2 (string).
- BSTR Measure3 - Set the channel of the measured parameter 3: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr (string).
- BSTR MeasureChannel4 - Set the channel of the measured parameter 4: 1; 2 (string).
- BSTR Measure4 - Set the channel of the measured parameter 4: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr (string).
- BSTR MeasureChannel5 - Set the channel of the measured parameter 5: 1; 2 (string).
- BSTR Measure5 - Set the channel of the measured parameter 5: Frequency, Hz; Period, s; Average, V; Span, V; RMS, V; Minimum, V; Maximum, V; Rise, s; Decrease, s; Positive

impulse, s; Negative impulse, s; Positive peak, V; Negative peak, V; High, V; Low, V; Phase, degr (string).

- BSTR AverageQuantity - Set the number of averages: 4; 8; 16; 32; 64 (string).
- FLOAT FirstPosition - Set the position of the first channel, cell (number).
- FLOAT SecondPosition - Set the position of the second channel, cell (number).
- FLOAT TimePosition - Set the position of zero time, cells (number).

➢**Methods:**

void Synchronization(LONG channel, LONG front, LONG type, FLOAT level) - Sets the timing settings on the screen:

- LONG channel - Set the synchronization channel: 0 - the first channel; 1 - second channel;
  LONG front - Set the front of synchronization: 0 - Upward; 1 - Downward;
  LONG type - Set the synchronization mode: 0 - Automatic; 1 - Normal;
  FLOAT level - Set the synchronization level, cells (from minus 5 to 5).

3.7.Histogram

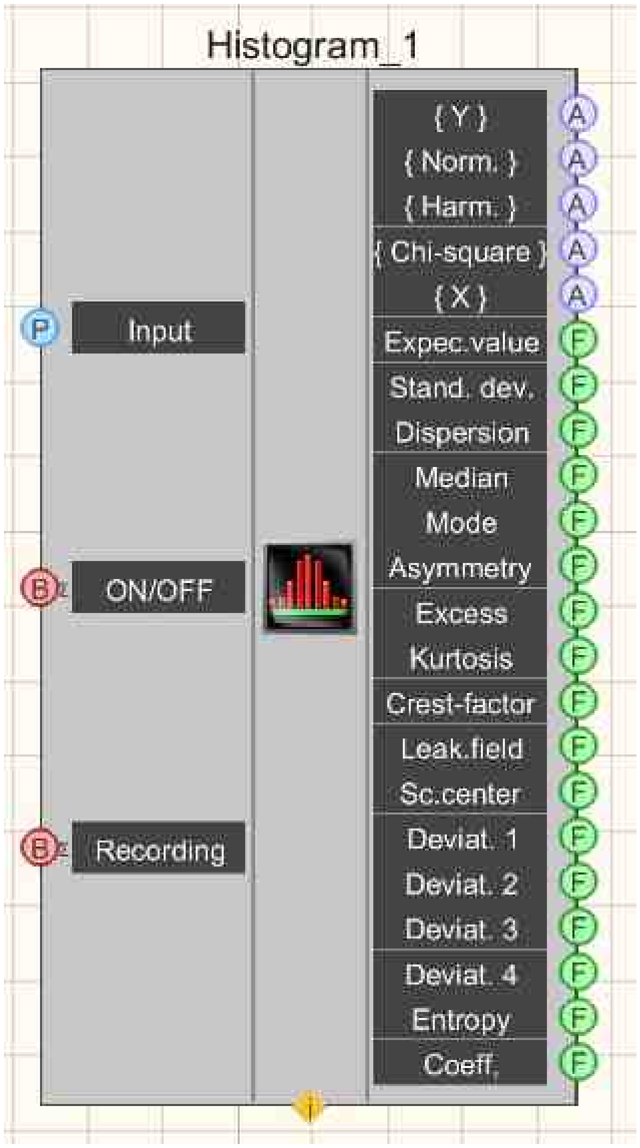
The component "**Histogram**" is intended for histographic analysis of the signal, finding statistical values characterizing the signal and constructing theoretical histograms based on the data obtained. The "**Histograph**" software is used for statistical signal analysis: finding the statistical values characterizing the signal and building the theoretic histograms based on the data obtained.
The software allows to build the following diagrams:

- Histogram
- Probability
- Probability density function

Input data for Histograph includes digital data of the ZETLAB server channel (signals registered by a FFT Spectrum Analysis, a strain-gauge station, or a seismic station).

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- Input - measuring channel.
- ON/OFF- enable/disable the histogram.
- Recording - when an impulse is applied to this contact, the data is saved to a separate file (the name and location of the file are specified in the private properties of the component).

➢ Output:

- {Y} - calculated values along the y-axis.
- {Norm} - calculated normal distribution.
- {Harm.} - calculated harmonic distribution.
- {Chi-square} - calculated Chi-square.

- {X} - marking of the abscissa axis.
- Expec. value - expected value.
- Stand.dev. - standard deviation.
- Dispersion - output channel dispersion.
- Median - the middle of the distribution.
- Mode - prevailing importance.
- Asymmetry - asymmetry output channel.
- Excess - output channel excess.
- Kurtosis - output channel kurtosis .
- Crest-factor - output channel crest-factor.
- Leak.field - output channel leakage field.
- Leak.center - output channel leakage center.
- Deviat. 1 - deviation from the normal distribution.
- Deviat. 2 - deviation from the harmonic distribution.
- Deviat. 3 - deviation from the Chi-squared distribution.
- Deviat. 4 - deviation from the distribution of what was recorded to the file.
- Entropy - Entropy output channel.
- Coeff. - entropy coefficient of quality.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- calcmode (By the voltage) - option for calculating histogram values:
  1. By the voltage.
  2. By digits.
- ColumnWidth (1) - histogram column width (in units):
- calctype (Histogram) - calculation view type:
- 1. Histogram.
- 2. Probability density.
- 3. Probability.
- Interval (1) - data accumulation time, s.
- AverageTime (1) - averaging time, s
- CalcNormal (false) - calculation of the normal distribution.
- CalcHarmonic (false) - calculation of harmonic distribution.
- CalcHiSquare (false) - calculation of the Chi-squared distribution.
- HiSquareNumber (1) - number of degrees of freedom distribution Chi-square.
- Activate (true) - operation status.
- FileName - the name of the file to recording the results.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - measuring channel (from 0 to (number of channels - 1)).

➢**Custom common properties:**

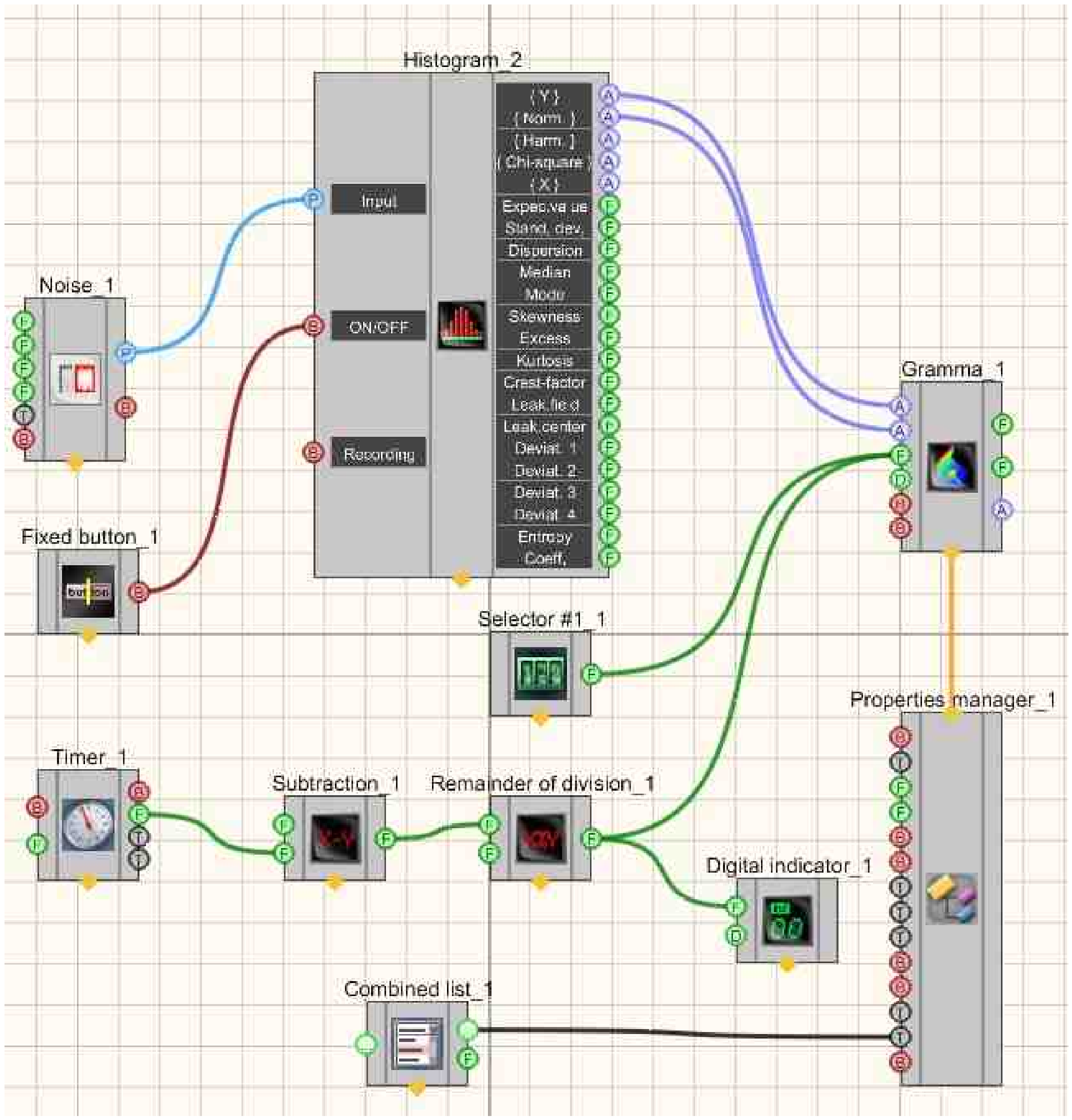
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

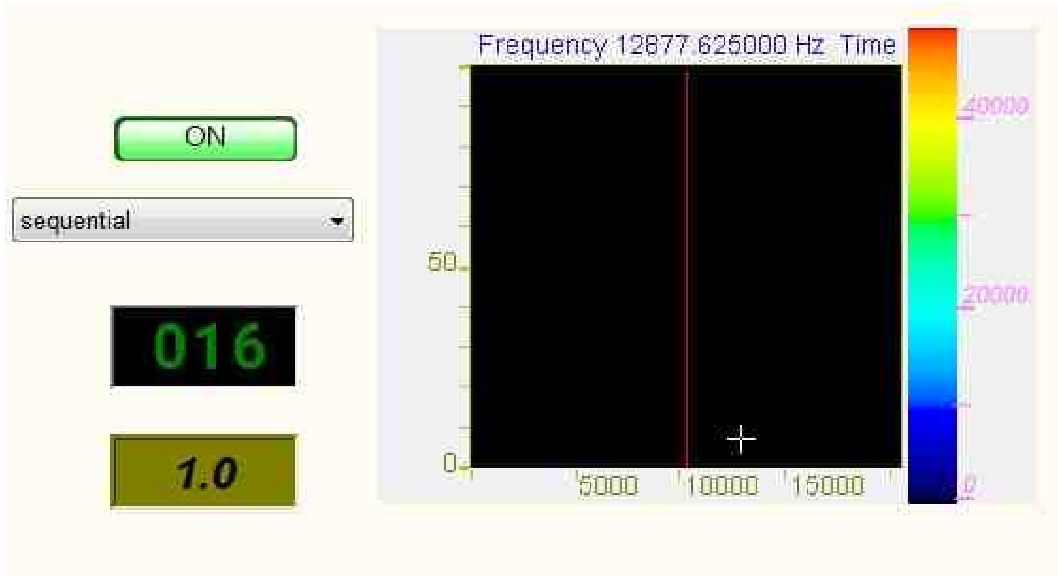
- BSTR CalcMode - Set the histogram calculation option: By the voltage; By digits (string).
- FLOAT ColumnWidth - Set and read the width of the histogram column, units meas (when set the type of calculation of values by voltage, it is set from the weight of the low-order digit of the ADC to the maximum limit input level divided by 16 - when set the type of calculation of values by digits, it is set from 1 to (digit capacity of the ADC is 5).
- BSTR CalcType - Set calculation view type: Histogram; Probability density; Probability (string).
- FLOAT Interval - Set the data accumulation time, s (from the data averaging time to 500 seconds).
- FLOAT AverageTime - Set and readaveraging: 0.1 s; 1 s; 10 s (number).
- VARIANT_BOOL CalcNormal - Set normal distribution calculation:
  true - the normal distribution calculation is turned On;
  false - the normal distribution calculation is turned Off.
- VARIANT_BOOL CalcHarmonic - Set the calculation of the harmonic distribution:
  true - the calculation of the harmonic distribution is turned On;
  false - the calculation of the harmonic distribution is turned Off.
- VARIANT_BOOL CalcHiSquare - Set the calculation of the Chi-square distribution:
  true - the calculation of the Chi-square distribution is turned On;
  false - the calculation of the Chi-square distribution is turned Off.
- LONG HiSquareNumber - Set the number of degrees of freedom of the Chi-squared distribution (from 1 to 15)
- VARIANT_BOOL Activate - Set the operation status of the histogram:
  true - the operation status of the histogram is turned On;
  false - the operation status of the histogram is turned Off.
- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).

Example

Project in SCADA ZETView



Project operation results



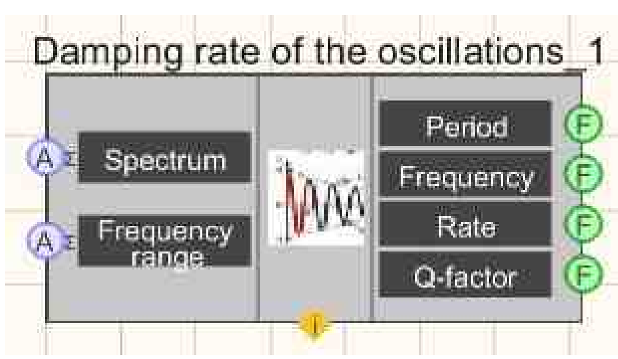
View the example in ZetView

3.8.Damping rate of the vibrations

The component "Damping rate of the vibrations" is designed to determine the main period of natural vibrations and the logarithmic damping rate of the vibrations and performs:

- calculation of the value of the natural frequency by determining the frequency at which the maximum of the constructed envelope is located;
- calculation of the value of the main period of natural oscillations;
- calculation of the logarithmic rate of damping according to the formulas D.3, D.4, D.5 and D.6 (GOST 54859-2011 Appendix D).

Appearance of the component:

<table>
<tr><td>Developer interface</td><td>Operator interface</td></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:
- Spectrum - a numeric array of power spectrum.
- Frequency range - a numerical array of the frequency range of the power spectrum.

➢ Output:
- Period - the numerical value of the main period of natural vibrations.
- Frequency - the numerical value of the natural vibration frequencies.
- Decrement - numerical value of the logarithmic rate of damping.
- Q-factor - the numerical value of the quality factor of vibrations.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Doesn't have.

Component operation algorithm

The component must work according to the following algorithm:
- getting input data (power spectrum and frequency range of the spectrum);
- select of the frequency range of the analysis in accordance with the boundaries configured through the parameters;
- search for the maximum frequency in the spectrum;
- calculation of the boundaries of the frequency range, within which it is necessary to build the envelope of the spectrum;
- construction of the spectrum envelope within the calculated boundaries using the least squares method;
- calculation of the natural frequency value by determining the frequency at which the maximum of the constructed envelope is located;
- calculation of the value of the main period of natural vibrations;
- calculation of the logarithmic attenuation decrement according to formulas D.3, D.4, D.5 and D.6 (GOST 54859-2011 Appendix D);
- calculation of the quality factor of vibrations according to the formula D.1 (GOST 54859-2011 Appendix D);
- transmission of the calculated values of the natural vibration frequencies, the fundamental tone of natural vibrations, the logarithmic rate of damping and the quality factor of vibrations to the outputs of the component.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Spectrum - a numeric array of power spectrum.
- Frequency range - a numerical array of the frequency range of the power spectrum.
- Spectrum - numeric values of the array to which you want to apply the power spectrum array.
- Frequency range - numeric values of the array to which the frequency range of the power spectrum array should be fed.

➢**Custom common properties:**

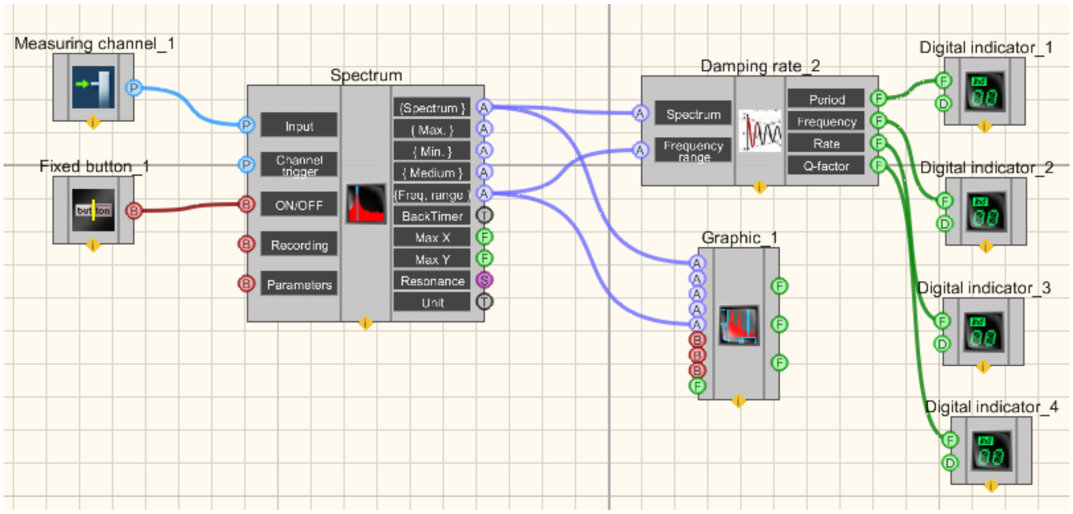
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

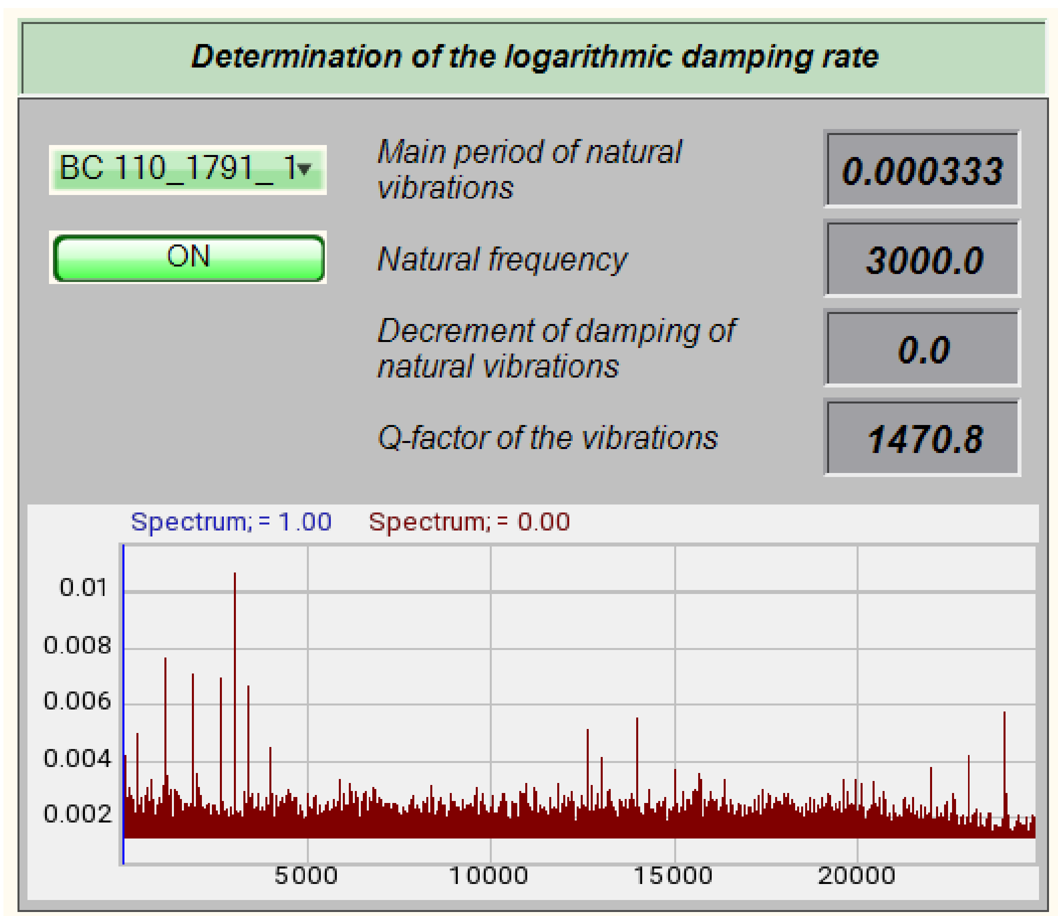
Doesn't have

Example

Project in SCADA ZETView



Project operation results

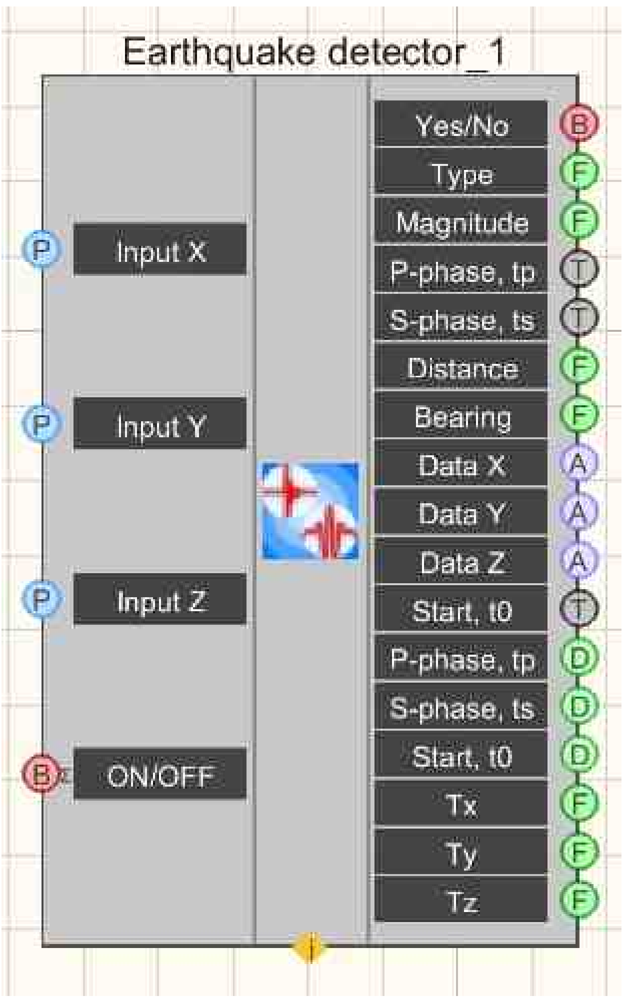


View the example in ZetView

3.9.Earthquake detector

The component "**Earthquake detector**" is designed to detect and classify seismic events within a radius of about 200 km.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input X - channel name of the X component;
- Input Y - channel name of the Y component;

- Input Z - channel name of the Z component.
- ON/OFF - turn on the earthquake detector.

➢ Output:

- Yes/No - the event is observed or not;
- Type - the type of the last event;
- Magnitude - the magnitude of the last event;
- P-phase, tp - entry time of the P-phase of the last event;
- S-phase, ts - entry time of the S-phase of the last event;
- Distance - distance to the epicenter of the last event, km;
- Bearing - bearing of the direction of arrival of the P-phase wave, degr. from 0 to 360;
- Data X - initial data of the last event on the X component;
- Data Y - initial data of the last event on the Y component;
- Data Z - initial data of the last event on the Z component.
- Start, t0 - the beginning of the event in earthquake focus.
- P-phase, tp - time of entry of the P-phase.
- S-phase, ts - time of entry of the S-phase.
- Start, t0 - the beginning of the event in earthquake focus.
- Tx - the main period for X.
- Ty - the main period for Y.
- Tz - the main period for Z.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Fmin (1) - the lower frequency of the band-pass filter during detection, Hz;
- Fmax (10) - the upper frequency of the band-pass filter during detection, Hz;
- PreHistory (2) - the prehistory time with background noise, times the STA duration (at least 2);
- ThresholdSTA (3) - STA/LTA detector threshold;
- ThresholdAccelEvent (0.03) - the earthquake threshold by peak acceleration per event, m/sI;
- LevelWavelet (4) - the decomposition level for signal wavelet filtering;
- TimeSTA (10) - the duration of the short window of the STA/LTA detector, s;
- TimeLTA (190) - the duration of the long window of the STA/LTA detector, s;
- Typewavelet (db2) - the mother wavelet when performing wavelet filtering of the signal;
- Activate (False) - the operation status;
- SpeedPhaseP (6.8) - P-phase propagation velocity, km/s;
- RatioSpeedPSpeedS (1.74) - ratio of P-phase and S-phase propagation velocities;
- BearingY0 (0) - bearing of the Y vector of the seismic receiver used, degr. from 0 to 360;
- GroupWork (False) - the ability of the component to work in a group;

- FminForMag (0.2) - the lower frequency of the band-pass filter when calculating the magnitude, Hz;
- FmaxForMag (20) - the upper frequency of the band-pass filter when calculating the magnitude, Hz;
- TimeAverage (0.2) - averaging time when using polarization analysis, in times of the STA duration (no more than 1);
- ThressholdZigZag (1) - significance time when using polarization analysis, from 0.1 to 10 s;
- NeedDTU (True) - need to record intermediate results to dtu-files;
- SeismicSensorOk (True) - the status of the seismic receiver;
- Magn_A (0) - multiplicative coefficient of the magnitude calculation formula;
- Magn_B (0.6) - additive coefficient of the magnitude calculation formula.
- Magn_C (0) - additive magnitude correction, taking into account the sensor installation conditions.
- TimeAverageForMainPeriod (0.5) - averaging time when determining prevailing periods.
- AmplitudeRation (0.3) - allowable ratio of the amplitudes of the X, Y, Z components.

Types of events:

- 1 - an event with only one phase defined;
- 2 - an event in which two phases have been determined, but the maximum acceleration amplitude of the second phase is less than the maximum acceleration amplitude of the first phase (not an earthquake);
- 3 - an earthquake whose magnitude is less than the specified threshold in magnitude;
- 4 - an earthquake with a magnitude is greater than the specified threshold in magnitude;
- -1 - an event that has an incorrect value when calculating parameters.

Description of the component

Wavelet filtering of signals
is used when detecting the first phase (background noise is filtered out, therefore it is necessary to set the signal prehistory time without an event) and when detecting the second phase (first phase signals are filtered out). Changing the type of the mother wavelet and the level of decomposition can lead to an increase in the workload of the computer processor without improving the accuracy of determining the timing of the earthquake phases.

Polarization analysis of signals
it is used in determining the duration of the first phase, as well as in determining the bearing of the direction of arrival of the P-phase wave (not to be confused with the bearing to the epicenter of the event!).

Parameters, TimeAverage and ThressholdZigZag

it is not recommended to change it, because this may lead to a deterioration in the accuracy of determining the timing of the entry of phases of an earthquake.

For output contacts "P-phase, tp" and "S-phase, ts"
the time of entry of the corresponding phase is given in text form in the format "DD.MM.YYYY HH:MM:SS.DS" (DS - fractions of a second), onExample: "13.05.2012 13:17:05.12" - 13 May 2012 13 hours, 17 minutes, 5 seconds and 12 hundredths of a second. The computer time is used (not UTC!!!). In the case of working with the program "Play recorded signals", the time will correspond to the time of recording signals.

On the output contacts "Data ..."
arrays of source data for components X, Y and Z. are submitted. These data have not been subjected to frequency filtering, and therefore have a constant component of the output stage of the signal amplifier used (or accelerometer pre-amplifier). The time of the first count of these arrays is the time of the entry of the P-phase, i.e. tp.

The magnitude of an earthquake
is calculated according to a well - known formula for determining the magnitude by volume waves:

$$M = \lg\left({A_{max}}/{A_0}\right) + 2\lg(2)$$
,

where A_{max} is the maximum displacement value recorded during the event, and A_0 = 1 µm. This formula corresponds to local earthquakes within a radius of about 200 km. The magnitude on the Richter scale cannot be greater than 9.0.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input X - channel name of the X component (from 0 to (number of channels- 1)).
- Input Y - channel name of the Y component (from 0 to (number of channels - 1)).

- Input Z - channel name of the Z component (from 0 to (number of channels - 1)).

➢**Custom common properties:**

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- FLOAT Fmin - Set the lower cutoff frequency of the band-pass filter, Hz (from 0 to the upper cutoff frequency of the band-pass filter).
- FLOAT Fmax - Set the upper cutoff frequency of the band-pass filter, Hz (from the lower cutoff frequency of the band-pass filter to (sampling frequency / 2)).
- FLOAT PreHistory - Set the prehistory time with background noise, in times of the STA duration (at least 2) (number).
- FLOAT ThresholdSTA - Set the threshold of the STA/LTA detector (from 3 to 60).
- FLOAT ThresholdAccelEvent - Set earthquake threshold by peak acceleration per event, m/sI (number).
- LONG LevelWavelet - Set the decomposition level in the wavelet transform of a signal (number).
- FLOAT TimeSTA - Set and readthe duration of the short window of the STA\LTA detector (STA duration) (from 0 to (LTA duration / 10)), s
- FLOAT TimeLTA - Set and readthe duration of the STA\LTA detector long window (LTA duration) (from (STA * 10 duration) to infinity), s
- BSTR TypeWavelet - Set and readthe mother wavelet when performing the wavelet transform of the signal: Haar; db1; db2; db3; db4; db5; db6; db7; db8; db9; db10; sym1; sym2; sym3; sym4; sym5; sym6; sym7; coif1; coif2; coif3; coif4; coif5 (string).
- VARIANT_BOOL Activate - Set and readthe operation status of the earthquake detector:
  true - the operation status of the earthquake detector is turned On;
  false - the operation status of the earthquake detector is turned Off.
- FLOAT SpeedPhaseP - Set and readthe P-phase propagation velocity, km/s (number).
- FLOAT RatioSpeedPSpeedS - Set and readthe ratio of P-phase and S-phase propagation velocities (number).
- FLOAT BearingY0 - Set and readthe bearing of the Y vector of the seismic receiver used, degr (from 0 to 360).
- VARIANT_BOOL GroupWork - Set and readthe ability of a component to work in a group:
  true - the ability of a component to work in a group is turned On;
  false - the ability of a component to work in a group is turned Off.
- FLOAT FminForMag - Set and readthe lower frequency of the band-pass filter when calculating the magnitude, Hz (number).
- FLOAT FmaxForMag - Set and readthe upper frequency of the band-pass filter when calculating the magnitude, Hz (number).
- FLOAT TimeAverage - Set and readthe averaging time when using polarization analysis, in times of the STA duration (number).
- FLOAT ThresholdZigZag - Set and readthe significance time when using polarization analysis (from 0.1 to 1 s).
- VARIANT_BOOL NeedDTU - Set and readthe need to record intermediate results to dtu-files.

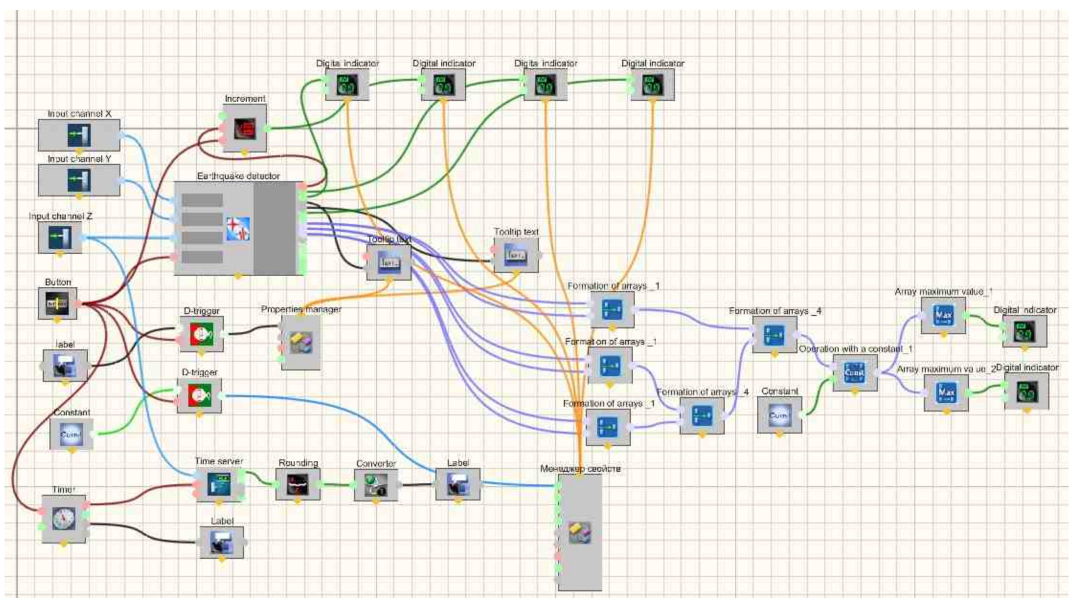
true - the need to record intermediate results to dtu-files is turned On;
false - the need to record intermediate results to dtu-files is turned Off.

VARIANT_BOOL SeismicSensorOk - Set the status of the seismic receiver.

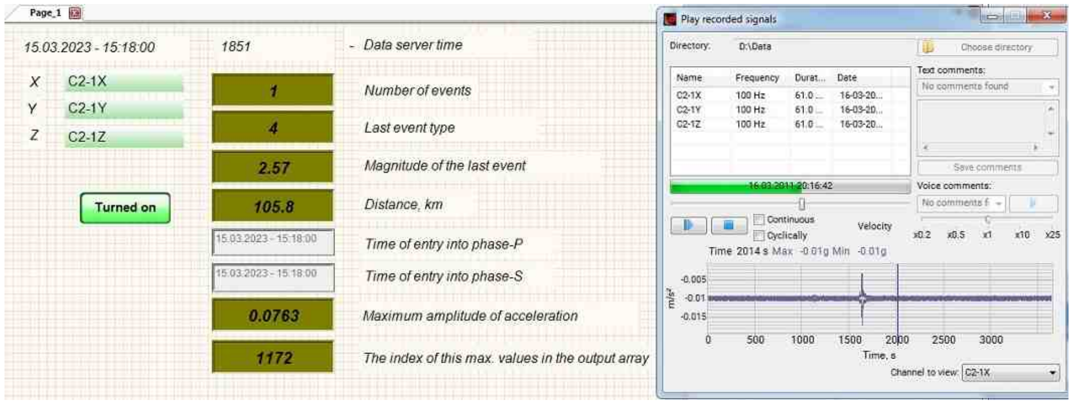
- true - the status of the seismic receiver is turned On;
  false - the status of the seismic receiver is turned Off.
- FLOAT Magn_A - Set and readthe multiplicative coefficient of the magnitude calculation formula (number).
- FLOAT Magn_B - Set and readthe additive coefficient of the magnitude calculation formula (number).
- FLOAT Magn_C - Set and readthe additive magnitude correction, taking into account the sensor installation conditions (number).
- FLOAT TimeAverageForMainPeriod - Set and readthe averaging time when determining prevailing periods (number).
- FLOAT AmplitudeRation (0.3) - Set and readthe allowable amplitude ratio of the X, Y, Z components. components (number).

Example

Project in SCADA ZETView



Project operation results



View the example in ZetView

3.10.STA/LTA Detector

The component "**STA/LTA Detector**" is designed for detecting seismic events recorded by one- or three-component seismic receivers.
The operating work of the component is the product of the impulse duration of STA (TimeSTA property) in seconds by the sampling frequency of the ADC in Hz should be more than 10.

The STA/LTA Detector program is a representation of a detector of various events in triaxial or single-component time signals. The software is based on one of the STA/LTA detector types.

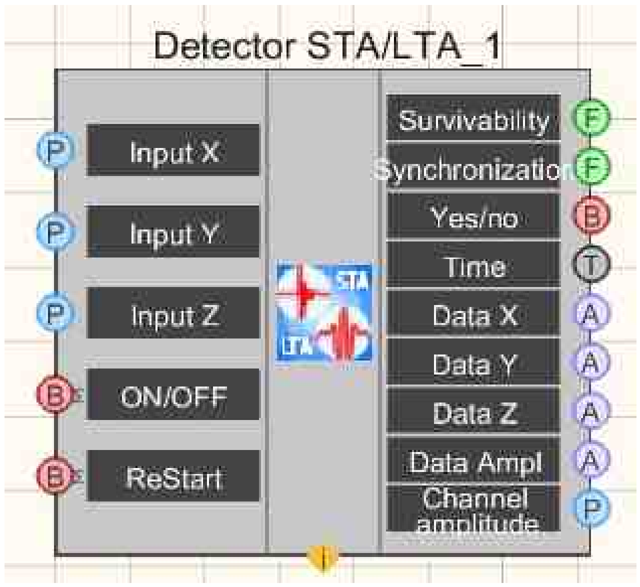
STA/LTA Detector can be used for detecting any events accompanied by a short-term increase in the input signal amplitude, e.g.:

- seismic events;
- explosions of various nature;
- events expected during safeguarding activities;
- various structures initial defects detection (using acoustic emission).

The figures below show the results of the STA/LTA Detector software operation when reproducing the signals of the 5.0 magnitude. The earthquake was registered by the **BC 1313 sensor** located approximately at a distance of 210 km from the earthquake focus.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- Input X - digitized data X of the seismic receiver signal;
- Input Y - digitized data Y of the seismic receiver signal;
- Input Z - digitized data Z of the seismic receiver signal;
- ON/OFF - enable/disable the STA/LTA detector;
- ReStart - command for "hot" restart of the component.

➢ Output:

➢ Survivability - component survivability signal;
➢ Synchronization - the maximum value of desynchronization of X, Y, Z signals in time;
➢ Yes/no - logical signal whether the event is observed or not;
➢ Time - absolute time (in text form) of the first frame of the event;
➢ Data X - filtered event data by X;
➢ Data Y - filtered event data by Y;
➢ Data Z - filtered event data by Z;
➢ Ampl data - filtered event data by amplitude;
➢ Amplitude channel - ID of the amplitude channel.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Activate (True) - the component operation mode;
- signal (Vectorial 3D) - the signal type: Scalar or Vectorial;

- PeriodVitality (1) - the period with which the component will generate a survivability signal, s;
- Fmin (0,5) - the high-pass filter cutoff frequency with band-pass filtering of the signal, Hz;
- Fmax (10) - the low-pass filter cutoff frequency with band-pass filtering of the signal, Hz;
- TimeSTA (10) - the duration of the short window of the STA/LTA detector, s;
- TimeLTA (190) - the duration of the long window of the STA/LTA detector, s;
- Threshold (3) - the STA/LTA detector threshold;
- TimeBefore (5) - the time before the start of the event, which will be included in the interval of the output data of the event, can be equal to 0, s;
- TimeAfter (10) - the time after the start of the event, which will be included in the interval of the output data of the event, s.

Description of the component

Purpose

The STA/LTA Detector component (hereinafter referred to as the component) is designed to detect seismic events recorded by one- or three-component seismic receivers.

DSP (Digital signal processing) methods used by the component

The component performs band-pass filtering of the seismic receiver output signals, determines the absolute start time of the event, and provides output contact of the event data arrays in X, Y, Z, and signal amplitude. In addition, the component creates a virtual channel with data on the amplitude of the filtered seismic receiver signal.

Signal filtering is performed by digital Butterworth filters. For event detection, one of the varieties of widely used STA/LTA detectors is used. The exact start time of an event is determined using wavelet filtering of the background noise of the signals recorded immediately before the event.

The component properties set the signal durations before and after the start of the event for issuing arrays to the output contacts.

Component operation

With the Activate = true property, the component starts its work immediately after the project is launched. Otherwise, work begins when a logical unit arrives at the On / Off input contact. The arrival of a logical zero on this contact in both cases stops the operation of the component.

Immediately after the start, as well as after a restart during the initialization of the detector, the component generates a survivability signal equal to one. The detector initialization time is equal to the sum of TimeSTA and TimeLTA. After the detector initialization is completed, the vitality signal is updated with the PeriodVitality period. In the absence of an event, the component's liveness signal should be around one (but the probability that the new value will be exactly equal to the old value is very small), during the observation of the event - more than one, immediately after the event - less than one, and may even be less than 0.

After any event is detected, a logical unit will be immediately issued to the Yes/No contact. After a time equal to TimeAfter, the component will determine the start time of the event, which will be sent in text form to the "Time" contact in the format "DD.MM.YYYY HH:MM:SS.xx", where xx is a fraction of a second. Also, output contacts whose names begin with the word "Data" will receive data arrays. X, Y, Z data and their amplitude are the digitized values (and their amplitude) of the original signals after band pass filtering.

After the event is completed (the component decides on this on its own), a logical zero will be output to the Yes/No contact.

If the value of the live signal stops changing, then this may either mean that the component is executing looped code for some reason, or that the original X, Y, Z channels are out of sync (on Example, when working with network channels due to problems with network transmission). To diagnose the problems of the second case, the component has a "Synchronization" contact, to which the maximum time difference of the X, Y, Z channels is displayed, if this value is greater than 1 s. Small values given to this pin do not affect the operation of the component. However, such a status of the original signals may occur when the difference in their times may lead to the inability for the component to receive new data through these channels. In this case (when the live signal does not change, but the synchronization signal is greater than 1), data will be lost. When the normal input of the initial data on the X, Y, Z channels resumes, the component will resume its work, and, as a result, the survivability signal will begin to change.

When the component loops (when the vitality signal does not change, and the synchronization signal is equal to 0), it is necessary to briefly apply a logical one signal to the "ReStart" input contact, which will lead to a "hot" restart of the component. In case of out of sync input data, restarting the component can also be useful.

In addition, the component creates a virtual channel that contains the amplitude data (square root of the sum of the squares of the X, Y, and Z signal components) of the filtered signal (filtered by a band-pass filter with cutoff frequencies Fmin and Fmax). Stopping the server time on this virtual signal will mean that the component does not receive initial data for X, Y and Z.

You can use multiple instances of a component in one project.
Remarks

When determining the maximum value of a signal in a detected event, it is necessary to set the TimeAfter property with a margin, because it is possible that the maximum value will be observed later than the start of the event than the time equal to TimeAfter. In the example, for an earthquake whose epicenter is at a distance of about

200 km from the seismic receiver, the S-phase comes after the entry of the P-phase after a time of about 26 s. Therefore, if you set TimeAfter and TimeSTA to 10 s, then the maximum value that is always observed during the S-phase may be skipped.

On the other hand, the sum of TimeAfter and TimeSTA multiplied by the ADC sampling frequency must not exceed the size of the component detector buffer (20000 samples).

Immediately after the start of the component, the data on the amplitude of the filtered signal in the virtual channel will begin to change. And on the "Vitality" contact, a single value will be kept for a time equal to the sum of TimeSTA and TimeLTA.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Input X - the name of the channel of the X component (from 0 to (number of channel - 1)).
- Input Y - the name of the channel of the Y component (from 0 to (number of channel - 1)).
- Input Z - the name of the channel of the Z component (from 0 to (number of channel - 1)), when set the scalar dimension of the original signal, it is the Z component that is set as the input.

➢Custom common properties:

- **Are available by the link below.** 154

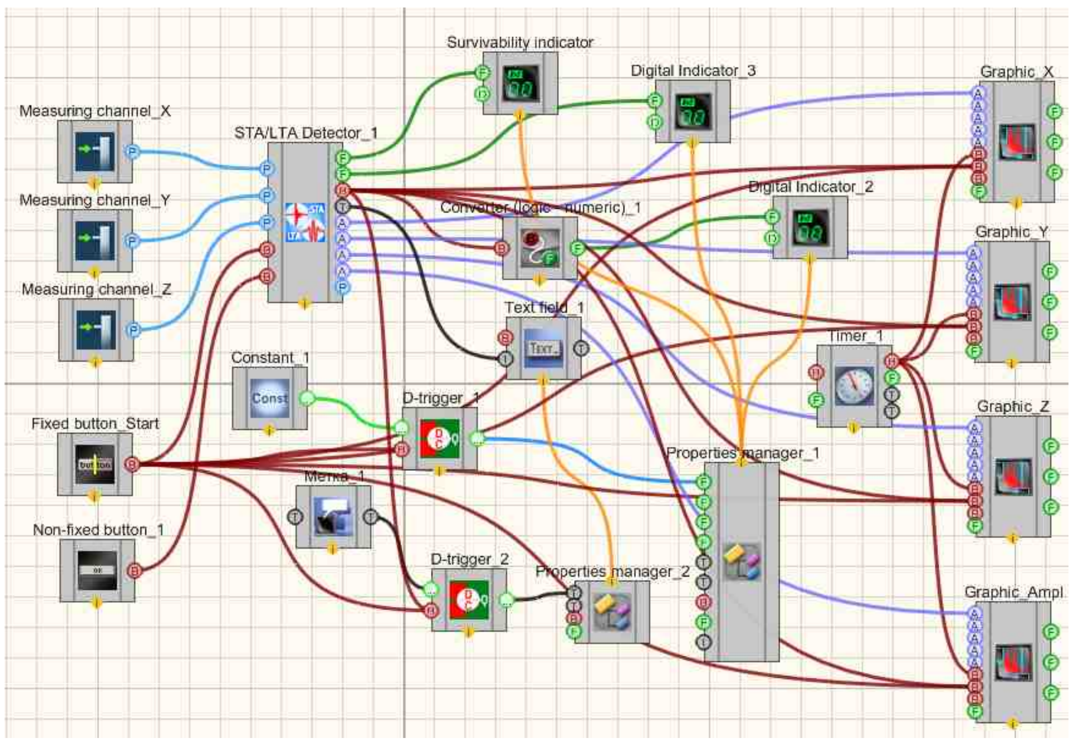
➢Custom private properties (the default value is shown in parentheses):

- VARIANT_BOOL Activate - Set the operation status of the STA/LTA detector:
  true - the operation status of the STA/LTA detector is turned On;
  false - the operation status of the STA/LTA detector is turned Off.
- BSTR Signal - Set input signal: Scalar; Vectorial 3D (string).
- FLOAT PeriodVitality - Set the period for issuing a survivability signal, s (number).
- FLOAT Fmin - Set the lower cutoff frequency of the band-pass filter, Hz (from 0 to the upper cutoff frequency of the band-pass filter).
- FLOAT Fmax - Set the upper cutoff frequency of the band-pass filter, Hz (from the lower cutoff frequency of the band-pass filter to (sampling frequency / 2)).
- FLOAT TimeSTA - Set the duration of the short window of the STA\LTA detector (STA duration) (from 0 to (LTA duration / 10)), s

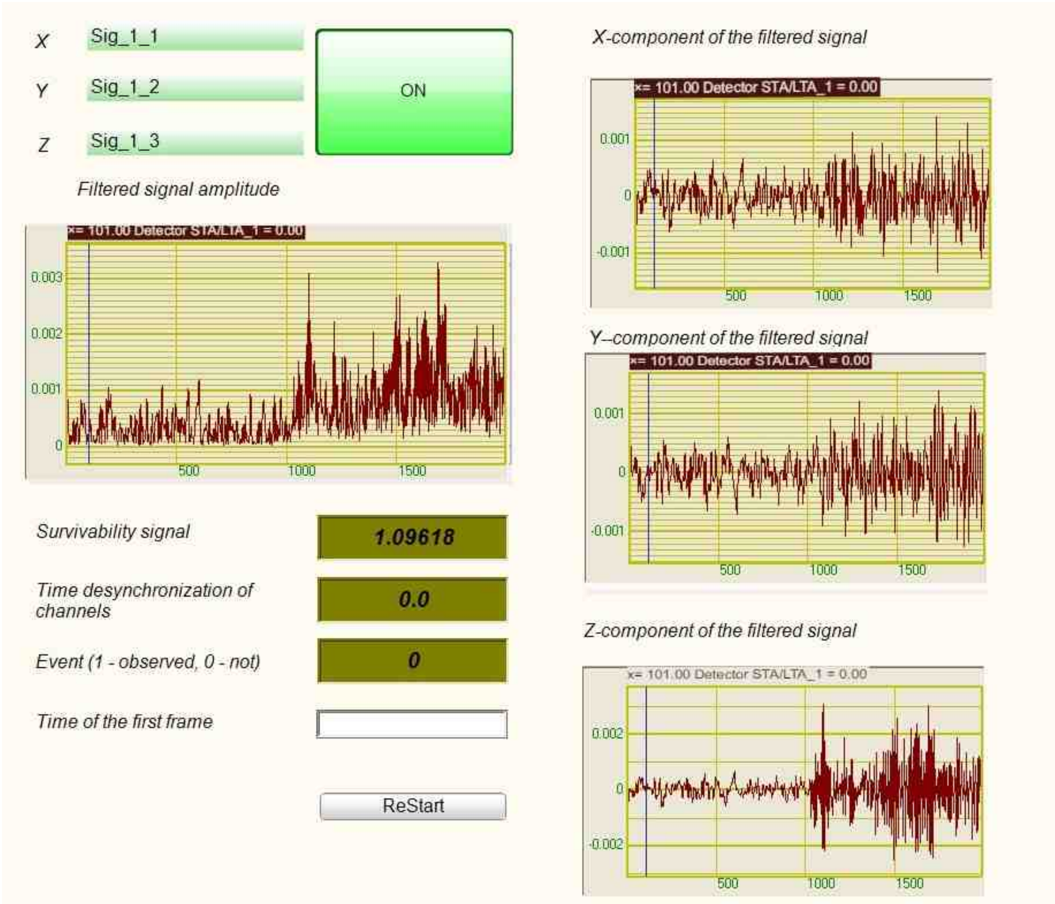
- FLOAT TimeLTA - Set the duration of the STA\LTA detector long window (LTA duration) (from (STA * 10 duration) to infinity), s
- FLOAT Threshold - Set the threshold of the STA/LTA detector (from 3 to 60).
- FLOAT TimeBefore - Set the time of issuing signals to the first frame of the event, s.
- FLOAT TimeAfter - Set the time of issuing signals after the first frame of the event, s

Example

Project in SCADA ZETView



Project operation results



View the example in ZetView

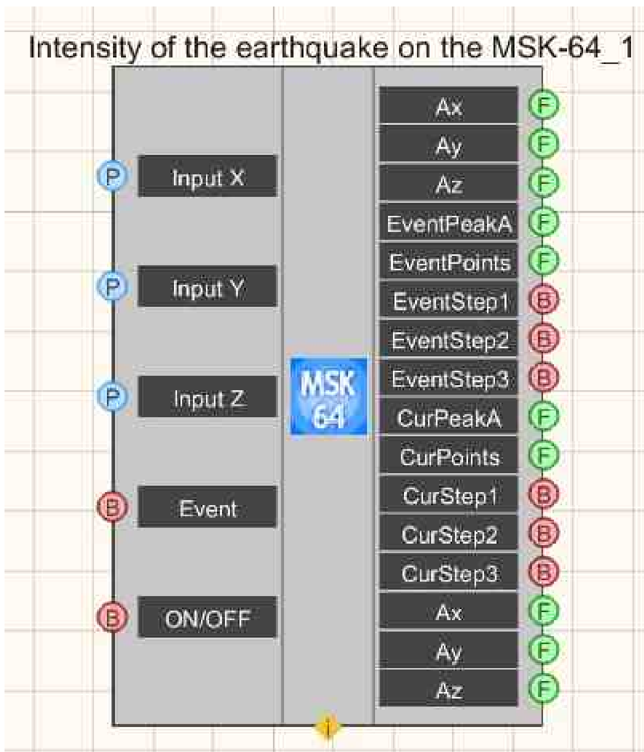
3.11.Intensity of the earthquake on the MSK-64

The component "**Intensity of the earthquake on the MSK-64**" is designed to determine the intensity of earthquakes on the MSK-64 scale using three-component seismic accelerometers (BC 1313 seismic sensors).
The Medvedev-Sponheuer-Karnik scale (MSK-64) is used to determine the intensity of seismic impacts in 1964. It was developed and widely used in Europe and the former USSR. The MSK-64 macroseismic scale describes the strength of an earthquake by the nature of its perception by a person, the nature of the destruction of buildings and the degree of changes in the environment. The assessment of the strength of an earthquake on this scale depends on its magnitude and the location of the observation point. At the epicenter (a point on the earth's surface above the hearth), the intensity of seismic impacts will depend on the depth of the hearth.

The MC-64 scale is the basis of SNiP-II-7-81 "Construction in seismic areas" and is used in Russia and CIS countries.
The magnitude of the earthquake intensity is determined by the component in accordance with GOST R 53166-2008 "Earthquakes. General characteristics".

Appearance of the component:

<table>
<tr><td>Developer interface</td><td>Operator interface</td></tr>
<tr><td>



</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:

- Input X - name of the X channel of the seismic receiver;
- Input Y - name of the Y channel of the seismic receiver;
- Input Z - name of the Z channel of the seismic receiver;
- Event - event (yes/no).
- ON/OFF - enable/disable the component;

➢ Output:

- Ax - current peak acceleration in X;
- Ay - current peak acceleration in Y;
- Az - current peak acceleration in Z;

- EventPeakA - peak acceleration during the event;
- EventPoints - intensity in points during the event;
- EventStep1 - exceeding the first threshold during the event;
- EventStep2 - exceeding the second threshold during the event;
- EventStep3 - exceeding the third threshold during the event.
- CurPeakA - current peak acceleration.
- CurPoints - current intensity in points.
- CurStep1 - exceeding the first threshold by the current peak acceleration;
- CurStep2 - exceeding the second threshold by the current peak acceleration;
- CurStep3 - exceeding the third threshold by the current peak acceleration;
- Ax - maximum acceleration in X per event;
- Ay - maximum acceleration in Y per event;
- Az - maximum acceleration in Z per event.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Activate (false) - the component operation mode (ON/OFF);
- Fmin (1) - the lower frequency of the band-pass filter, Hz;
- Fmax (10) - the upper frequency of the band-pass filter, Hz;
- typepeak (by amplitude) - type of peak value (By the amplitude/By components);
- Group (false) - the component operation in a group.
- SensorOk (true) - the seismic receivers status.
- Correction (0) - the additive correction factor for soils, points.
- Step1 (4) - the intensity threshold value 1, points.
- Step2 (6) - the intensity threshold value 2, points.
- Step3 (2) - the intensity threshold value 3, points.
- ChoiceReactionTime (3) - the decision time integer s, greater than 0.
- principledecision (Majority without one) - the principle of decision making (Majority less one/For all).
- BearingY(0.0) - the seismic receiver Y-axis bearing, from 0 to 360 deg.

Description of the component

The results of the seismic station, determined by the MSK64 component

The component produces the following results:

1 current peak value of acceleration according to the seismic station, hereinafter referred to as CDR (current peak value);

2 flags for exceeding the CDR (current peak value) of the specified thresholds (you can set no more than 3 thresholds);

3 peak value of acceleration along the seismic station during the event, the beginning and end of which is determined outside the component, hereinafter referred to as EPV (event peak value);

4 flags for exceeding the EPV of the specified thresholds (thresholds are the same).

The values of CDR and EPV are presented in "m/s^2", and are also converted into points on the MSK-64 scale.

When set the flags for exceeding the CDR specified thresholds, for each operating sensor, the fact that the CDR minimum threshold is exceeded is determined, taking into account the time filter, hereinafter referred to as EF (exceeding fact). The length of the time filter for each of the sensors can be different and is set in the "ChoiceReactionTime" component property - the decision time as an integer number of seconds.

The minimum threshold (out of three) is the minimum threshold value of the given thresholds. Not set threshold = 0.

Principles for determining results

The component can work in accordance with the following principles for determining the results in ratio to the working sensors of the seismic station:

1 majoritarian without one;

2 for all.

The "PrincipleDecision" property in the component determines the type of calculation implemented in the component.

The principle of "Majority without one"

1 CDR is defined as the second from the bottom among all current values of the working sensors.

2.The flags for exceeding the CDR of the specified thresholds are set if a EF is observed on all working sensors without one and the CDR is greater than or equal to the value of this threshold.

3 The EPV is defined as the maximum of the CDR during the duration of the event.

4 The flags for exceeding the EPV of the specified thresholds are set if the EPV is greater than or equal to the value of the given threshold.

The principle of "All"

1 CDR is defined as the arithmetic mean value among all current values of operating sensors.

2 The flags for exceeding the CDR of the specified thresholds are set if a EF is observed on all working sensors and the CDR is greater than or equal to the value of this threshold.

3 The EPV is defined as the maximum of the CDR during the duration of the event.

4 The flags for exceeding the EPV of the specified thresholds are set if the EPV is greater than or equal to the value of the given threshold.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input X - measuring channel X (from 0 to (number of channel - 1)).
- Input Y - measuring channel Y (from 0 to (number of channel - 1)).
- Input Z - measuring channel Z (from 0 to (number of channel - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

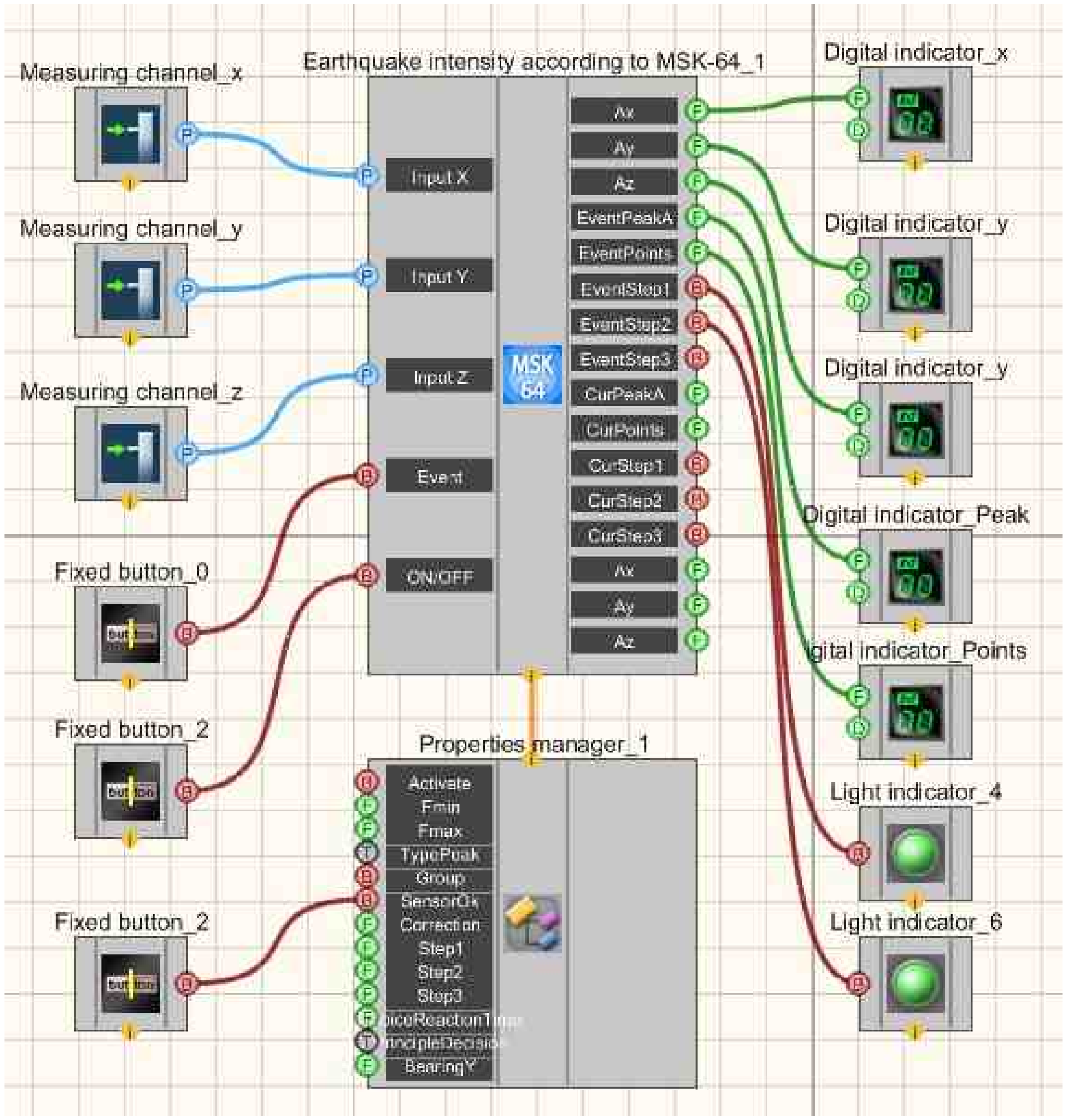
➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL Activate (false) - Set the operation status of the mode (ON/OFF);
  true - the operation status of the mode is turned On;
  false - the operation status of the mode is turned Off.
- FLOAT Fmin - Set and reading the lower cutoff frequency of the band-pass filter, Hz (from 0 to the upper cutoff frequency of the band-pass filter).
- FLOAT Fmax - Set and reading the upper cutoff frequency of the bandpass filter, Hz (from the lower cutoff frequency of the bandpass filter to (sampling frequency / 2)).
- LONG Step1 - Set and reading intensity threshold 1, points (number).
- LONG Step2 - Set and reading intensity threshold 2, points (number).
- VARIANT_BOOL Group - Set and reading the operation of a component in a group:
  true - the operation of a component in a group is turned On;
  false - the operation of a component in a group is turned Off.

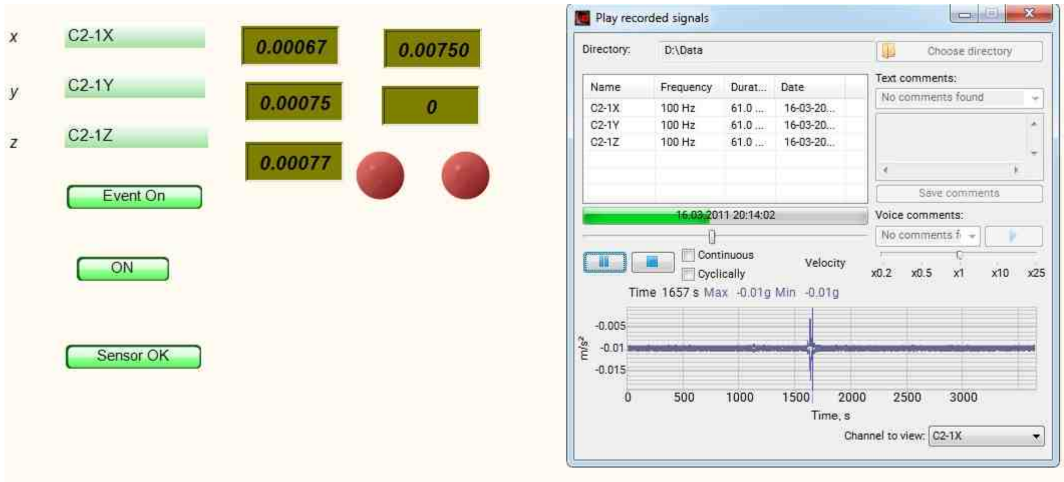
- VARIANT_BOOL SensorOk - Set seismic receiver status:
  true - The status of the seismic receiver is turned On;
  false - The status of the seismic receiver is turned Off.
- LONG Correction - Set and reading of the additive correction factor for soils, points (number);
- VARIANT_BOOL Activate - Set and reading the operation status of the earthquake intensity component according to MSK-64:
  true - The status of the component is turned On;
  false - The status of the component is turned Off.
- BSTR TypePeak -Setting and reading the type of peak value: By the amplitude; By components (string).
- LONG ChoiceReactionTime - Set and reading decision time, integer s, greater than 0 (number).
- BSTR PrincipleDecision - Set and reading the decision principle: Majority less one; For all (string).
- FLOAT BearingY - Set the Y-axis bearing of the geophone (from 0 to 360 degrees).

Example

Project in SCADA ZETView



Project operation results



View the example in ZetView

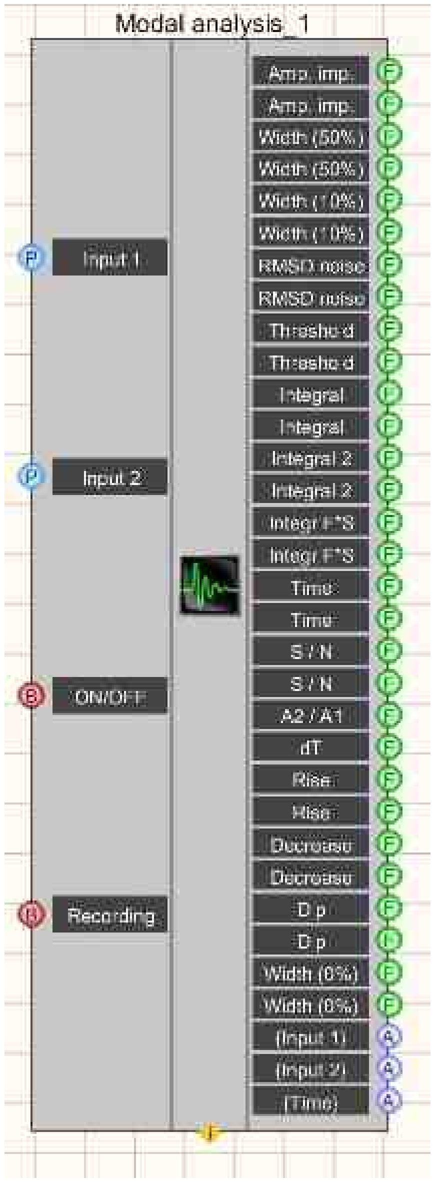
3.12.Modal analysis

The component "**Modal analysis**" is designed to analyze the impulse and transient characteristics of signals coming from the input channels of spectrum analyzers and seismic stations in real time or in the Play recorded signals mode of recorded time realizations.

Modal analysis is used for analyzing impulse and transition characteristics of signals coming from the input channels of FFT Spectrum Analysis and seismic stations in real time or recorded time realization view mode.

Appearance of the component:

Developer interface	Operator interface

<table>
<tr>
<td>
</td>
<td>Doesn't have</td>
</tr>
</table>

Setting:

➤Input:
- Input 1 - reference channel.
- Input 2 - measuring channel.
- ON/OFF - enable modal analysis.
- Recording - recording in a file.

➤ Output:
- Amp.imp. - amplitude of the impulse on the reference channel.
- Amp.imp. - amplitude of the impulse on the measuring channel.
- Width(50%) - the impulse width on the reference channel at 50% level.

- Width(50%) - the impulse width on the measuring channel at the 50% level.
- Width(10%) - the impulse width on the reference channel at a 10% level.
- Width(10%) - the impulse width on the measuring channel by 10% level.
- RMS Noise - RMS noise on the reference channel before the impulse.
- RMS Noise - RMS noise on the measuring channel before the impulse.
- Threshold - threshold for response to an impulse on the reference channel.
- Threshold - threshold for response to an impulse on the measuring channel.
- Integral - first-order integral of the momentum on the reference channel.
- Integral - first-order integral of the impulse on the measuring channel.
- Integral 2 - second-order integral of the impulse on the reference channel.
- Integral 2 - second-order integral of the impulse on the measuring channel.
- Integ.F*S - integral of the product of force and displacement on the reference channel.
- Integ.F*S - integral of the product of force and displacement on the measuring channel.
- Time - the time of passage of the impulse on the reference channel.
- Time - the time of passage of the impulse on the measuring channel.
- S/N - signal-to-noise ratio on the reference channel
- S/N - the signal-to-noise ratio on the measuring channel.
- A2/A1 - the ratio of impulse amplitudes on the reference and measuring channels.
- dT- the difference in impulse repetition time on the measuring and reference channels.
- Rise - the rise time of the signal on the reference channel.
- Rise - the rise time of the signal on the measuring channel.
- Decrease- the time of the decline of the signal on the reference channel.
- Decrease - the time of the decline of the signal on the measuring channel.
- Dip - the minimum signal before shock on the reference channel.
- Dip - the minimum signal before shock on the measuring channel.
- Width (0%) is the width of the impulse on the reference channel at the 0th level.
- Width (0%) is the width of the impulse on the measuring channel at the 10th level.
- {Input 1} - data from the reference channel.
- {Input 2} - data from the measuring channel.
- {Time} - a time series.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Freqrange (50000) - the frequency range, Hz.
- ReferenceChanInversion (false) - the reference channel inversion.
- MeasuringChanInversion (false) - the measuring channel inversion.
- MeasuringChanLocalization (false) - the normalization of the measuring channel.
- Thresholdtype (Adaptive (RMS*K)) - noise RMS threshold (in units of measurement).
- Coefficient (10) - adaptive threshold RMS multiplier.

- AbsoluteReferenceChan (10) - the absolute RMS level of the reference channel (in units of measurement).
- AbsoluteMeasuringChan (10) - the absolute level of RMS of the measuring channel (in units of measurement).
- Fronttype (any) - start front.
- QualityInterval (50) - Q factor calculation interval, %.
- Interval (1) - the calculation interval, s.
- Auto (false) - autorun installation.
- AutoTime (10) - autorun interval, s.
- Activate (true) - operation status.
- FileName - the name of the files to record the results.
- MedianFiltr (false) - set the median filter.
- MedianFiltrLength(0) - set the length of discret.
- BandPassFlitr (false) - set the band-pass filter.
- HiPassFiltr(0) - set the high-pass filter, Hz (number).
- LowPassFiltr(0) - set the low-pass filter, Hz (number).
- OrderHiPassFiltr(0) - set the HPF order, 0 - automatic (number).
- OrderLowPassFiltr(0) - set the LPF order, 0 - automatic (number).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**
- Input 1 - the reference channel (from 0 to (number of channels - 1)).
- Input 2 - the measuring channel (from 0 to (number of channels - 1)).

➤**Custom common properties:**
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR FreqRange - Set the decade of the frequency range, Hz:
  0 – from 0 to (sampling frequency / 1)
  1 – from 0 to (sampling frequency / 10)
  2 – from 0 to (sampling frequency / 100)
  3 – from 0 to (sampling frequency / 1000)
  4 – from 0 to (sampling frequency / 10000).

VARIANT_BOOL ReferenceChanInversion - Set the reference channel inversion:

- true - the reference channel inversion is turned On;
  false - the reference channel inversion is turned Off.

VARIANT_BOOL MeasuringChanInversion - Set the inversion of the measuring channel:

- true - the inversion of the measuring channel is turned On;
  false - the inversion of the measuring channel is turned Off.

VARIANT_BOOL MeasuringChanNormalization - Set the normalization of the measuring channel:

- true - the normalization of the measuring channel is turned On;
  false - the normalization of the measuring channel is turned Off.

- BSTR ThresholdType - Set the threshold for RMS noise, (in units): Adaptive (RMS * K); Absolute (string).
- FLOAT Coefficient - Set adaptive threshold RMS multiplier (number).
- FLOAT AbsoluteReferenceChan - Set the absolute RMS level of the reference channel, (in units) (number).
- FLOAT AbsoluteMeasuringChan - Set the absolute RMS level of the measuring channel (in units) (number).
- BSTR FrontType - Set the trigger edge: Any; Positive (string).
- FLOAT QualityInterval - Set of the Q factor calculation interval, % (from 0.1% to 100%).
- FLOAT Interval - Set the calculation interval, s (number).

VARIANT_BOOL Auto - Set the enable analysis autorun:

- true - analysis autorun is turned On;
  false - analysis autorun is turned Off.

- FLOAT AutoTime - Set the time interval after which autorun occurs (from 0.1 seconds to 1000 seconds).

VARIANT_BOOL Activate - Set the operation status of modal analysis:

- true - the operation status of modal analysis is turned On;
  false - the operation status of modal analysis is turned Off.

- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).

VARIANT_BOOL MedianFiltr - Set the median filter:

- true - the median filter is turned On;
  false - the median filter is turned Off.

- ULONG MedianFiltrLength - Set the length of discretes (number).

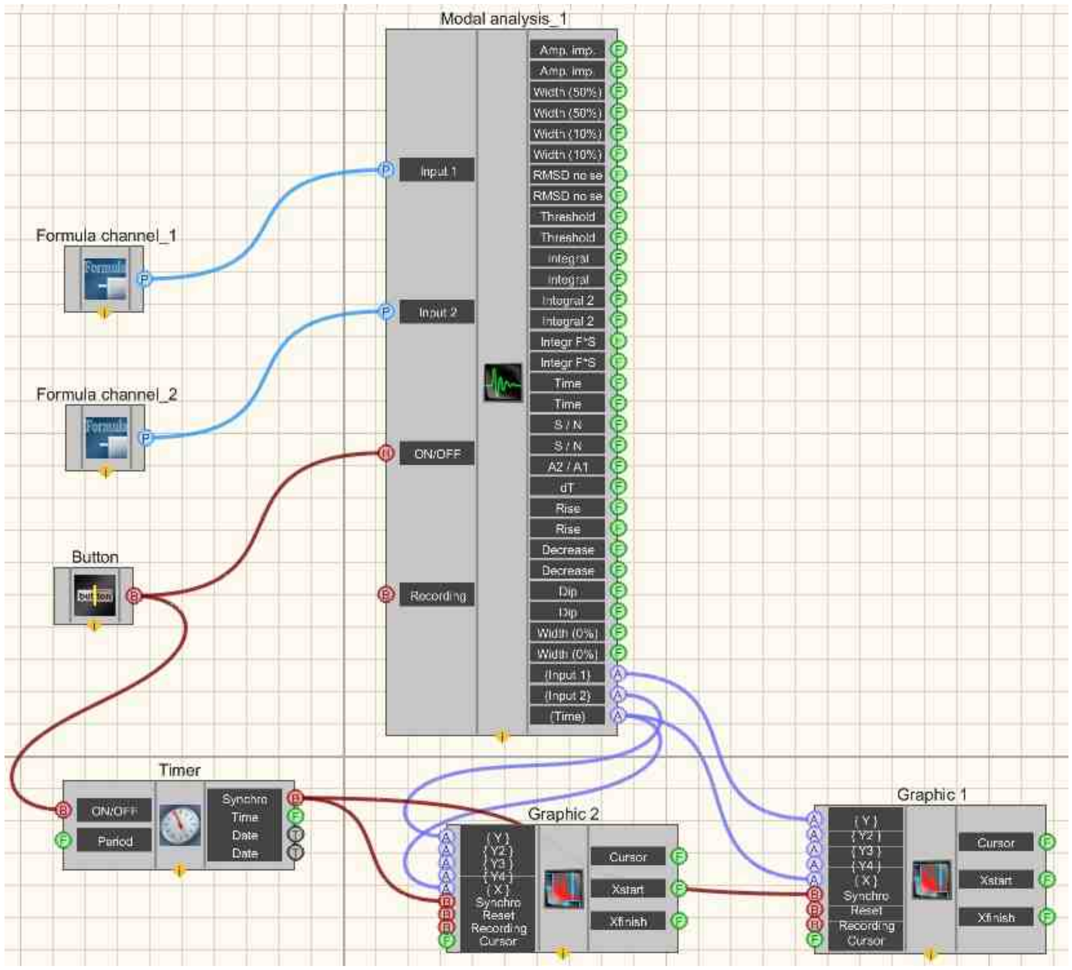
VARIANT_BOOL BandPassFlitr - Set the band-pass filter:

- true - the band-pass filter operation status is turned On;
  false - the band-pass filter operation status is turned Off.

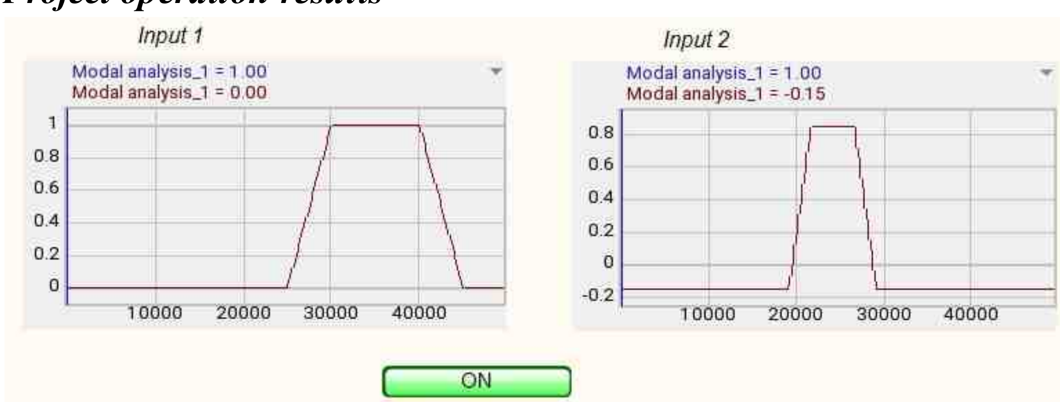
- FLOAT HiPassFiltr - Set the high pass filter, Hz (number).
- FLOAT LowPassFiltr - Set the low pass filter, Hz (number).
- ULONG OrderHiPassFiltr(0) - Set the HPF order, 0 - automatic (number).
- ULONG OrderLowPassFiltr(0) - Set the LPF order, 0 - automatic (number).

Example

Project in SCADA ZETView



Project operation results

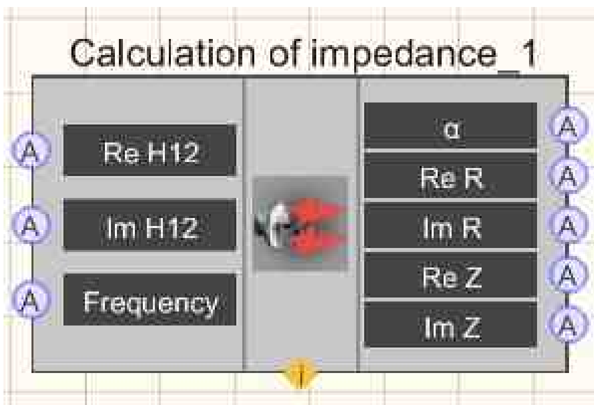


View the example in ZetView

3.13.Calculation of impedance

The component "**Calculation of impedance**" is designed to calculate acoustic impedance parameters in accordance with international standards ISO 10534-2.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Re H12 - the real part of the transfer function H12.
- Im H12 - the imaginary part of the transfer function H12.
- Frequency - frequency range for the transfer function H12.

➢ Output:
- α - sound absorption coefficient α.
- Re R - the real part of the reflection coefficient R.
- . Im R - the imaginary part of the reflection coefficient R.
- Re Z - the real part of the acoustic impedance Z.
- Im Z - the imaginary part of the acoustic impedance Z.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Distance (0) - the distance between microphones, m
- EndMicrDistance (0) - the distance from the specimen to the far microphone, m
- TubeDiametr (0) - the impedance pipe diameter, m
- SoundSpeed (0) - the sound velocity, m/s.
- AirDensity (0) - the air density, kg/m^3.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

-
- Re H12 - the real part of the H12 transfer function (any number).
- Im H12 - the imaginary part of the transfer function H12 (any number).
- Frequency - the frequency range for the transfer function H12 (any number).

➢**Custom common properties:**

- Are available by the link below. 154

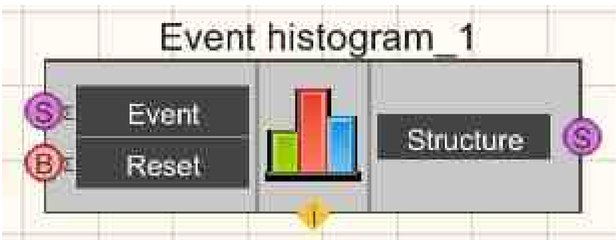
➢**Custom private properties (the default value is shown in parentheses):**

•FLOAT Distance - Set the distance between the microphones, m.
•FLOAT EndMicrDistance - Set the distance from the specimen to the far microphone, m
•FLOAT TubeDiametr - Set the impedance pipe diameter, m
•FLOAT SoundSpeed - Set the sound velocity, m/s.
•FLOAT AirDensity - Set the air density, kg/m^3.

3.14.Event histogram

The component "**Event histogram**" is designed to build a histogram based on the values of the fields of the received structures for a given time interval in seconds. Output contacts are created dynamically according to the number of fields in the structure, arrays containing histogram values are output from them. And the last leg of the component gives out a structure, the values of the fields of which are filled with the value of the middle of the interval of the maximum of the histogram.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Event - the structure with seismic events.
- Reset - the reset histogram

➢Output:

- Structure - the output structure.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- FileName - the full name of the file with the description of the structure.
- Interval (0) - time interval, s.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Event - structure with seismic events (structure).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•BSTR FileName - Set the full name of the file with a description of the structure.
•Interval - Set the time interval, s.

3.15.Super-resolution Spectrum

The component "**Super-resolution Spectrum**" is designed for spectral analysis of signals with high frequency resolution.

Super-resolution Spectrum is used for spectral signal analysis with high frequency resolution. The program is used in diagnostics and monitoring systems of buildings and for analyzing non-stationary signals. The Software enables solving the following tasks:

- FFT Spectrum Analysis parameters long-term dependencies analysis
- signal harmonic components spectral separation.

The program does not serve as a substitution for "FFT Spectrum Analysis" software which is used to determine primary signal frequency band in the case of narrow-band spectrum analysis.

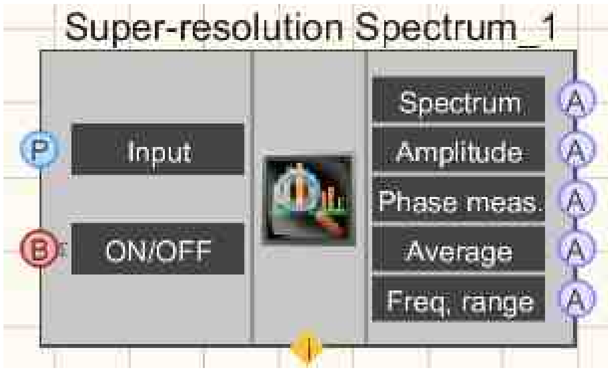
Then the "Super-resolution Spectrum" program enables perfomance of spectral analysis with a maximum resolution value available in the "FFT Spectrum Analysis" software.

The program creates a two-unital sonogram of the selected signal. Horizontal axis represents the frequency value of the selected band and the vertical axis shows the time (time value maximum limit is up to 83 hours).

The sonogram has a color representation of spectral density power. The color sonogram allows the operator to easily determine the maximum value change nature, which (depending on the particular task) can be a resonance, intrinsic frequency or other parameter of the researched FFT Spectrum Analysis.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- Input - the measuring channel.
- ON/OFF - enable/disable Super-resolution Spectrum.
-

➢ Output:
- {Spectrum} - current data with spectrum.
- {Amplitude} - current amplitude data of the center frequency.
- {Phase meas.} - current data of the change in the phase of the center frequency.
- {Average} - current amplitude data of the averaged signal.
- {Freq. range} - frequency range data.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Activate (True) - operation status Super-resolution Spectrum.
- FreqF0 (12.5) - designed to set the central frequency, Hz.
- FreqBand (1.25) - designed to set the frequency band, Hz.
- TimeView (10) - display time, s.
- TypeResolution (0.1) - type of frequency resolution: 0.1; 0.05; 0.025; 0.01; 0.005; 0.0025; 0.001;

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**
- Input - measuring channel (from 0 to (number of channels - 1)).

➤**Custom common properties:**
➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**
- VARIANT_BOOL Activate (True) - Set the operation status (ON/OFF);
  true - the operation status of the mode is turned On;
  false - the operation status of the mode is turned Off.
- FLOAT FreqF0 (12.5) - Set the center frequency, Hz.
- FLOAT FreqBand (1.25) - Set the frequency band, Hz.
- LONG TimeView (10) - Set the display time, s.
- BSTR TypeResolution (0.1) - Set the type of frequency resolution: 0.1; 0.05; 0.025; 0.01; 0.005; 0.0025; 0.001 (string).;

3.16.Spectrum CPB Analysis

The component "**Spectrum CPB Analysis**" is designed to display various parameters of the Spectrum CPB Analysis.

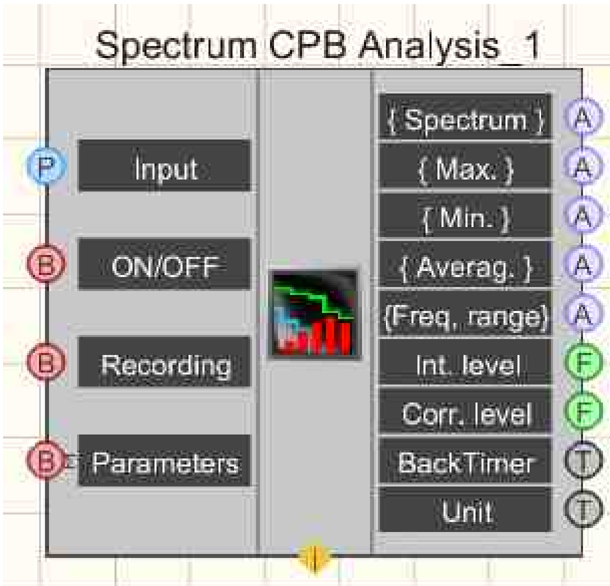
CPB (CONSTANT PERCENTAGE BANDWIDTH) ANALYSIS

CPB analysis is used for fractional octave spectral processing of signals coming from the input channels of FFT Spectrum Analysis (in real time or recorded time realization view mode), as well as for viewing various spectral characteristics of signals.

CPB analysis is used for separating signals into basic constituents in the frequency area in 1/3-, 1/12-, 1/24-octave spectral bands. The software is used for noise spectral analysis within the scope of acoustic and vibrational measurements.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- Input - measuring channel.
- ON/OFF - is turned On/Off the Spectrum CPB Analysis.
- Recording - when a impulse is applied to this contact, data is saved to a separate file (the name and location of the file are specified in the Custom private properties of the component).

➢ Output:

- {Spectrum} - current instant Spectrum.
- {Max.} - current maximum Spectrum.
- {Min.} - current minimum Spectrum.
- {Averag.} - current average Spectrum.
- {Fraq. range} - frequency range.
- Int. level. - integral level of the spectrum without taking into account the corrective function.
- Corr. level - integral level of the spectrum, taking into account the corrective function.
- BackTimer - estimated time left, s.
- {Unit} - unit of the result.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- type (1/3-octave) - Type of analysis:
  1. 1/1 octave;

2. 1/3 octave;
3. 1/12 octave;
4. 1/24 octave;
5. Logarithmic, dB.

- Average Time (1) - averaging time, s.
- Activate (true) - operation status.
- calctype - Spectrum calculation view type (RMS):
  1. RMS;
  2. Peak value.
- spectrview (logarithmic, dB) - type of spectral component level representation:
  1. linear scale (in units);
  2. logarithmic, dB.
- CalcMax (false) - calculation of the maximum spectrum.
- CalcMin (false) - calculation of the maximum spectrum.
- CalcSred (false) - calculation of the average spectrum.
- CalcTime (600.000000) - calculation time of additional Spectrum, s.
- FileName - the name of the files to record the results.
- intdiff (Unedited) choose the data processing type:
  1. Double differentiation;
  2. Differentiation;
  3. Unedited (Without processing integration and differentiation);
  4. Integration;
  5. Double integration.
- correction (Without correction) - Frequency correction function:
  1. Without correction;
  2. Correction A;
  3. Correction B;
  4. Correction C;
  5. Correction D.
- CalcOnlyLevel - Set the calculation with the correction of only the integral level.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - the first measuring channel (from 0 to (number of channels - 1)).

➢**Custom common properties:**

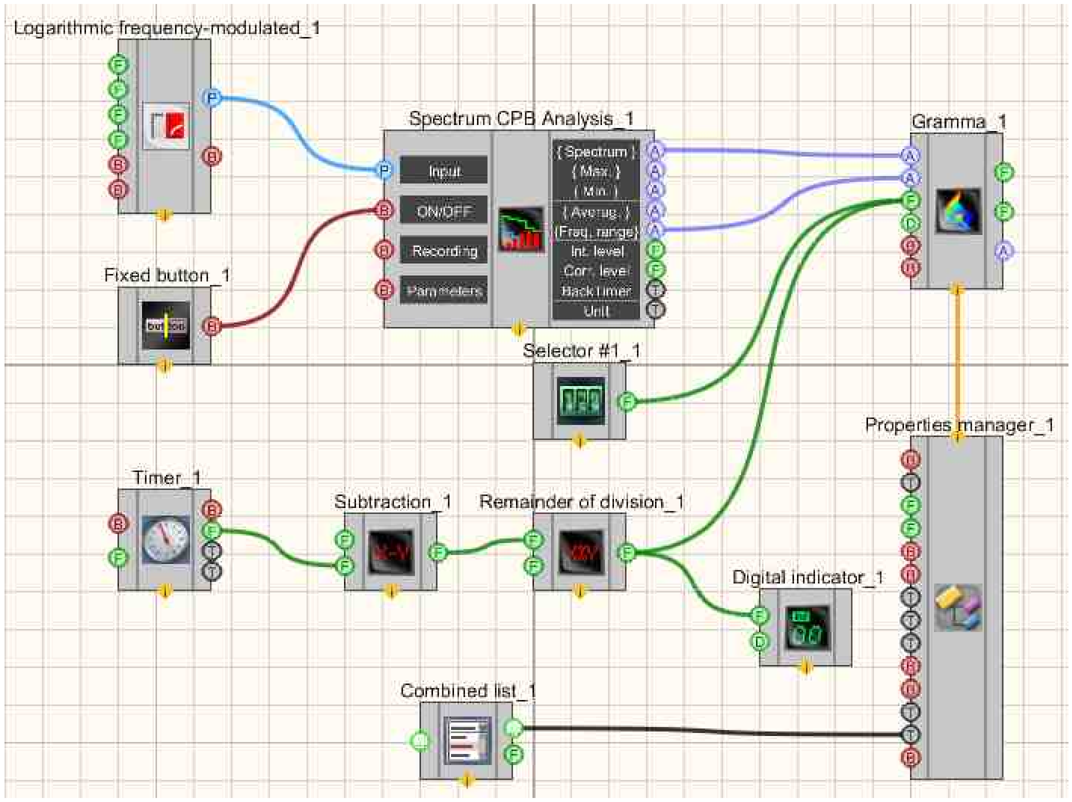
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

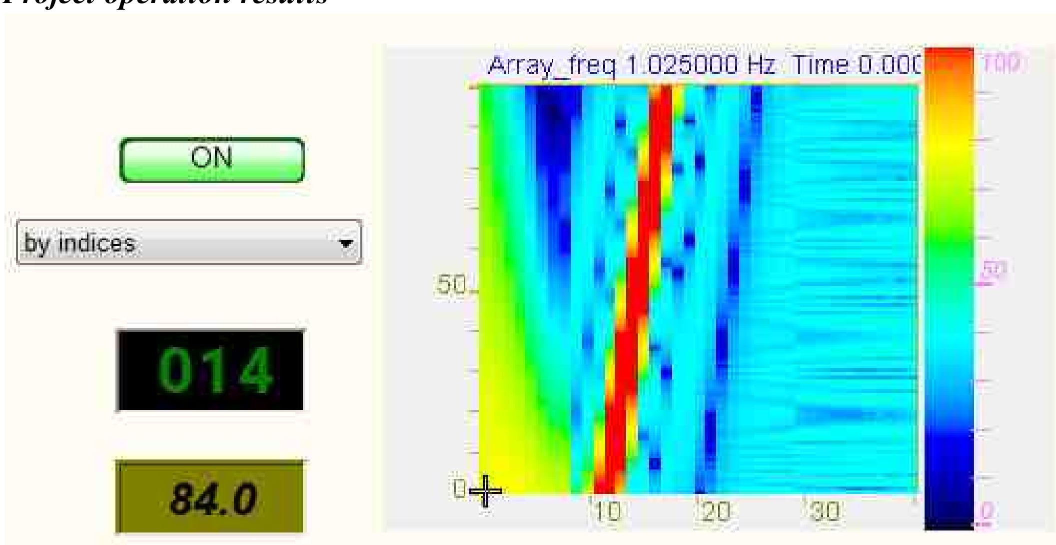
- BSTR Type - Set the type of analysis:1/1-octave;1/3-octave;1/12-octave;1/24-octave (string).
- FLOAT AverageTime - Set the averaging time, s (from 0.1 seconds to 100 seconds).
- VARIANT_BOOL Activate - Set the operation status of the Spectrum CPB Analysis:
  - true - the operation status of the spectrum is turned On;
    false - the operation status of the spectrum is turned Off;
- BSTR CalcType - Set the type of data presentation: RMS; Peak value (string).
- BSTR SpectrView - Set the type of spectral component level representation: Linear; Logarithmic, dB (string).
- VARIANT_BOOL CalcMax - Set the calculation of the maximum spectrum:
  - true - the calculation of the maximum spectrum is turned On;
    false - the calculation of the maximum spectrum is turned Off;
- VARIANT_BOOL CalcMin - Set the calculation of the minimum spectrum:
  - true - the calculation of the minimum spectrum is turned On;
    false - the calculation of the minimum spectrum is turned Off;
- VARIANT_BOOL CalcSred - Set the calculation of the average spectrum:
  - true - the calculation of the average spectrum is turned On;
    false - the calculation of the average spectrum is turned Off;
- FLOAT CalcTime - Set the calculation time for additional spectrum, s (from 600 seconds to 100000 seconds).
- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).
- BSTR IntDiff - Set integration/differentiation of signal processing type: Double differentiation; Differentiation; Unedited; Integration; Double integration (string).
- BSTR Correction - Set the correction function:Without correction; Correction A; Correction B; Correction C; Correction D (string).
- VARIANT_BOOL CalcOnlyLevel - Set the calculation with integral level correction only:
  - true - the calculation with integral level correction only is turned On;
    false - the calculation with integral level correction only is turned Off;

Example No. 1

Project in SCADA ZETView



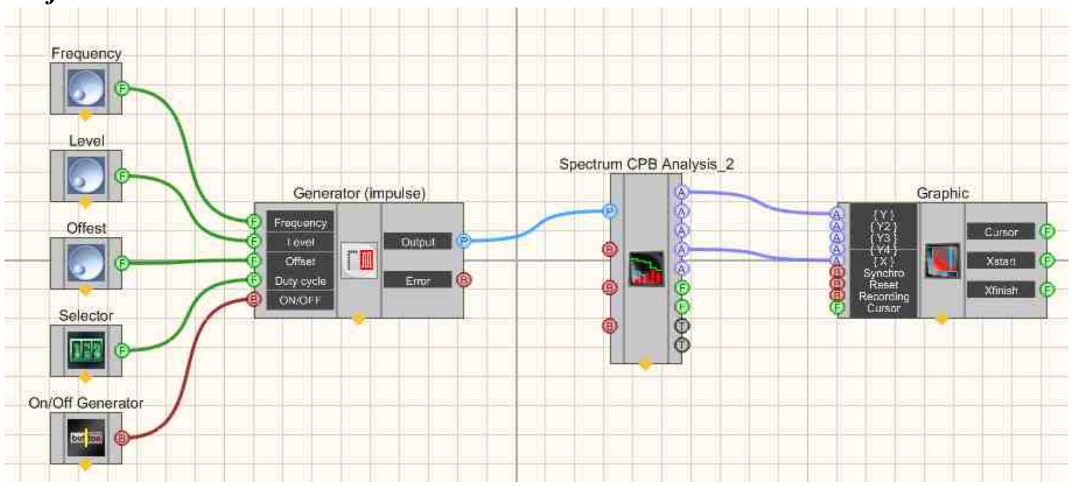
Project operation results



View the example in ZetView

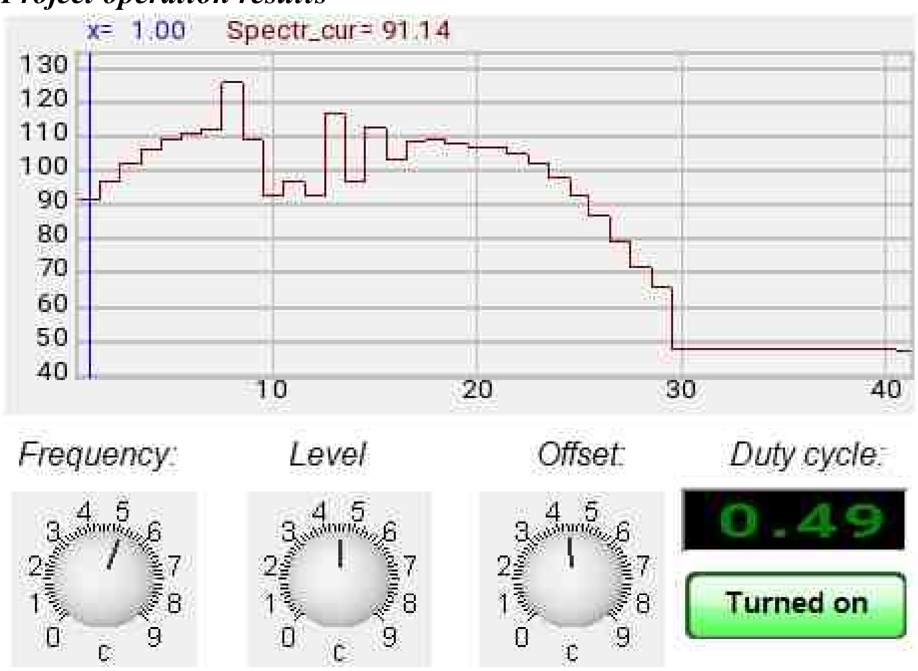
Example No. 2

Project in SCADA ZETView



In the above project, the Spectrum CPB Analysis 254 component measures the RMS values of the sawtooth signal coming from the Serrated signal 515 generator. Selectors 428 (also of a different type) 441 are used to set the frequency, level, offset, and sawtooth waveform values. The Graphic 909 is needed for visual perception of the signal. The Liquid crystal display 655 is used for graphical representation of the signal.

Project operation results



View the example in ZetView

3.17.FFT spectrum Analyzers

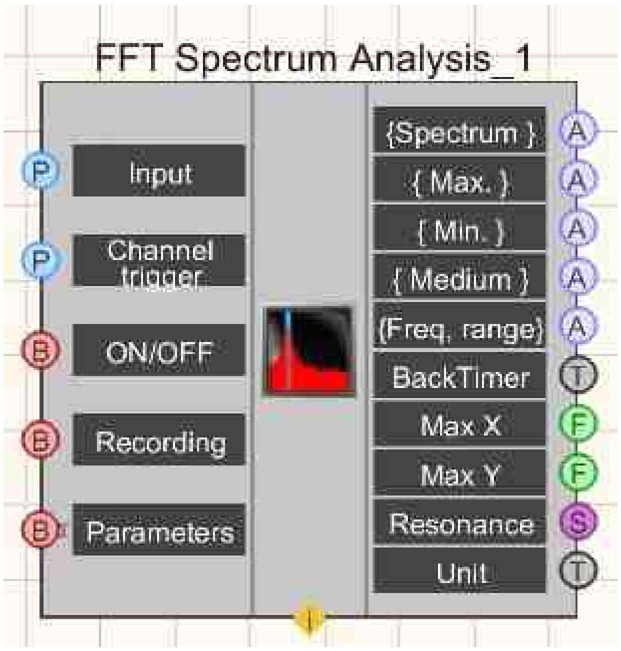
The component "**FFT Spectrum Analysis**" is designed for frequency analysis of the signal. According to the time realization of the signal, there are responses to a set of frequency filters. The central frequencies of the filters are evenly distributed along the frequency axis.

The program **FFT (Fast Fourier Transform) Spectrum Analysis** is used for narrow-band spectral processing of signals coming from the input channels of ADC modules and FFT Spectrum Analysis (in real-time or recorded time realization view mode), as well as for viewing various spectral characteristics of signals.
Spectral analysis is used for dividing signals into basic constituents in the frequency area. This analysis is based on signal time realization decomposition into the frequency spectrum with an even frequency increment by means of Fourier transform.

Using the **FFT Spectrum Analysis** program, based on the spectrum shape, the user can determine the presence of any signal tones (discrete constituents) and noise components in the measuring channel. Additional options of spectrogram building (in a 2- and 3-unital view) enable tracking of the non-stationary processes dynamics. A spectrogram is a spectral time representation of a signal, calculated for even time intervals. Building spectrogram sections based on time and frequency enables control of the non-stationary

Appearance of the component:

Developer interface	Operator interface

<table>
<tr>
<td>


</td>
<td>Doesn't have</td>
</tr>
</table>

Setting:

➢Input:

- Input - the first measuring channel.
- Channel trigger- the second measurement channel that is used for synchronization.
- ON/OFF - enables/disables FFT Spectrum Analysis.
- Recording - when a impulse is applied to this contact, data is saved to a separate file (the name and location of the file are specified in the **Custom common properties** of the component).
- Parameters - when a impulse is applied to this contact, the Settings of the program parameters are called, and if the settings are changed in these parameters, they will be applied, both in the program and the component, when the **Apply** button is pressed.

➢Output:

- {Spectrum} - current instant Spectrum.
- {Max.} - current maximum Spectrum.
- {Min.} - current minimum Spectrum.
- {Averag.} - current average Spectrum.
- {Fraq. range} - frequency range.
- BackTimer - estimated time left, s.
- {Max X} - frequency, Hz.
- {Max Y} - value in units of the displayed spectrum.
- {Resonance} - resonance parameters.
- {Unit} - unit of the result.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Average Time (1) - the average time, s.
- CalcType (RMS) - Spectrum calculation view type:
  1. Spectral density;
  2. Spectral power;
  3. RMS (root-mean-square);
  4. Amplitude value.
- AnalysisType (Discrete Fourier Transform) - type of analysis processing:
  1. Fast Fourier transform;
  2. Discrete Fourier transform.
- Functiontype (Hann) - type of weight function:
  1. Rectangular;
  2. Hann;
  3. Hamming;
  4. Blackman;
  5. Bartlett;
  6. Kaiser;
  7. Reef-Vincent (4);
  8. Blackman-Harris (3);
  9. Blackman-Harris (4);
  10. Natalla;
  11. Blackman-Natalla;
  12. Flat-top window.
- ScaleType (Linear) - the scale type:
  1. Linear;
  2. Logarithmic, dB.
- Activate (true) - operation status.
- FreqResolution (50) - number of bands in DFT:50;62;100;125;200;250;400;500;625;1000;1250;2000;2500;4000;5000; 6250;10000;12500;20000;25000;40000;50000.
- CalcMax (false) - the calculation of the maximum spectrum.
- CalcSred (false) - the calculation of the average spectrum.
- CalcTime (600) - the additional spectra calculation time (from 600 to 86400), s.
- intdiff (Unedited) - choose the data processing type:
  1. Double differentiation;
  2. Differentiation;
  3. Unedited (Without processing integration and differentiation);
  4. Integration;
  5. Double integration.
- Filter (false) - clearing the spectrum with a median filter.

- FileName - the name of the files to record the results.
- Decimation(1) - use the original signal decimation: 1; 10; 100; 1000; 10000.
- Resolution (1) - use the frequency resolution at DFT, Hz.
- UseResolution (True) - use the frequency resolution for the DFT, Hz.
- Averaging (Linear) - set the average type:
  1. Linear;
  2. Exponential.
- BandsNumberFFT (512) - number of bands at FFT: 64; 128; 256; 512; 1024; 2048; 4096; 8192; 16384; 32768.
- ResonanceCalc (false) - use the calculation of resonances.
- ResonanceFreq1(0) - use the calculation of resonances. Frequency 1, Hz.
- ResonanceFreq2(0) - use the calculation of resonances. Frequency 2, Hz
- CalcMin (false) - use the calculation of the minimum spectrum.

Description of the component

SCADA-component "FFT spectrum Analyzers",
version from 03/23/2023 year

Setting the frequency resolution when performing spectral analysis

In order to be able to set the frequency resolution when performing an FFT, a new property has been added in the new version of the component (see Table 1), which specifies the number of bands in the FFT. In addition, for a better understanding of the operation of the component, some properties have had their descriptions changed, see Table 2, which does not affect the operation of existing SCADA projects in any way.

Set the frequency resolution in the new version of the component is performed as follows.

First, the AnalysisType property specifies the type of analysis to be performed - FFT or DFT.

If the FFT is set, then the frequency resolution is set using the BandsNumberFFT (Number of bands in FFT). property. In this case, the resolution in Hz, and dF is calculated by the formula:

dF = FreqADC / (2 * BandsNumberFFT * Decimation),

where: FreqADC – sampling frequency of the selected channel in Hz; Decimation - The value of the Decimation property.

If DFT is selected, then the frequency resolution type is set by the UseResolution flag property (Use frequency resolution in Hz for DFT):

- true — the frequency resolution in Hz is given by the value of the Resolution property;
- false – the frequency resolution is set by specifying the number of frequency bands using the FreqResolution property, while the resolution in Hz dF is calculated by the formula:

dF = FreqADC / (2 * FreqResolution * Decimation).

During the transition from FFT to DFT and vice versa, the property values that determine the frequency resolution for both types of analysis are preserved.

Setting the calculation time for additional spectra

The CalcTime property, which sets the time for calculating additional spectra, has had its range of allowable values changed — from 600 seconds (10 minutes) to 86400 seconds (day).

Table 1 - Added component property

Property number	Name	Description
20	BandsNumberFFT	Number of bands with FFT:64;128;256;512;1024;2048;4096;8192;16384;32768;

Table 2 - Properties of the component whose description has changed

Property number	Name	Description
8	FreqResolution	Number of bands with DFT: 50;62;100;125;200;250;400;500;625;1000;1250;2000;2500;4000;5000;6250;10000;12500;20000;25000;40000;50000;
11	CalcTime	Additional spectra calculation time (from 600 to 86400), s

17	Resolution	Frequency resolution at DFT, Hz
18	UseResolution	Use DFT frequency resolution, Hz

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Input - measuring channel (from 0 to (number of channels - 1)).
- Channel trigger- the second measurement channel (from 0 to (number of channels - 1)), that is used for synchronization.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT AverageTime - Set average time, s (from 0.1 second to 100 seconds).
- BSTR CalcType - Set the calculation view type: Spectral density; Spectral power; RMS (root-mean-square); Amplitude value (string).
- BSTR AnalysisType - - Set signal processing type: Fast Fourier transform (FFT); Discrete Fourier transform (DFT) (string).
- BSTR FunctionType - Set the weight function type:Rectangular;Hann;Hamming;Blackman;Bartlett;Kaiser;Reef-Vincent (4);Blackman-Harris (3);Blackman-Harris (4);Natalla;Blackman-Natalla;Flat-top window;Sine window (string).
- BSTR ScaleType - Set the scale type: Linear; Logarithmic, dB (string).
- VARIANT_BOOL Activate - Set the operation status of the FFT Spectrum Analysis:
  true - the operation status of spectrum is turned On;
  false - the operation status of spectrum is turned Off.
- BSTR FreqResolution (50) - Set the number of bands in the DFT:50;62;100;125;200;250;400;500;625;1000;1250;2000;2500;4000;5000;6250;10000;12500;20000;25000;40000;50000.
- VARIANT_BOOL CalcMax - Set the calculation of the maximum spectrum:
  true - the calculation of the maximum spectrum is turned On;
  false - the calculation of the maximum spectrum is turned Off.
- VARIANT_BOOL CalcSred - Set the calculation of the average spectrum:
  true - the calculation of the average spectrum is turned On;
  false - the calculation of the average spectrum is turned Off.

- FLOAT CalcTime - Set the calculation time for additional spectra (from 600 to 86400), s
- BSTR IntDiff - Choose the data processing type: Double differentiation;Differentiation;Unedited (Without processing integration and differentiation);Integration;Double integration.(string).
- VARIANT_BOOL Filter - Clearing the spectrum with a median filter:
  true - the spectrum with a median filter is turned On;
  false - the spectrum with a median filter is turned Off.
- BSTR FileName - Set the file name to record the results to a *.dtu file (after setting this parameter, it is recorded to the specified file).
- BSTR Decimation - Set the decimation of the original signal:1; 10; 100; 1000; 10000 (string).
- FLOAT Resolution - Set the frequency resolution at DFT, Hz.
- VARIANT_BOOL UseResolution - Set the DFT to use the frequency resolution in Hz:
  true - the DFT to use the frequency resolution is turned On;
  false - the DFT to use the frequency resolution is turned Off.
- BSTR Averaging - set the average type: Linear; Exponential (string).
- BSTR BandsNumberFFT (512) - the number of bands at FFT: 64; 128; 256; 512; 1024 ;2048; 4096 ;8192; 16384; 32768.

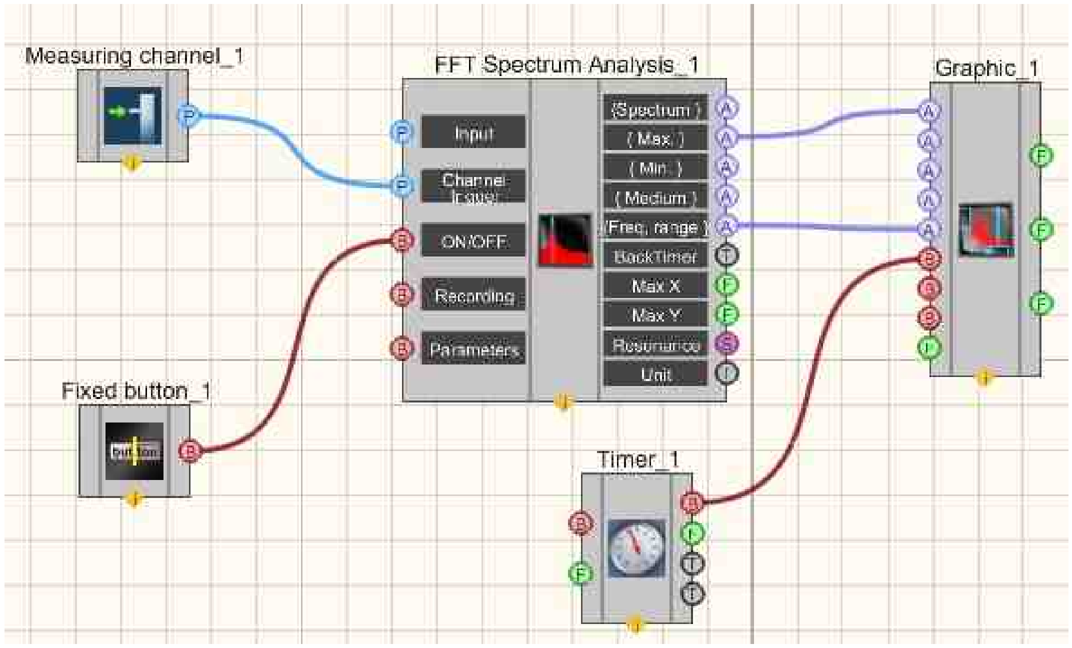
VARIANT_BOOL ResonanceCalc - Set the calculation of resonances:
  true - the calculation of resonances is turned On;
  false - the calculation of resonances is turned Off.
VARIANT_BOOL ResonanceTracing - Set the calculation of resonances using the tracking mode:
  true - the calculation of resonances using the tracking mode is turned On;
  false - the calculation of resonances using the tracking mode is turned Off.
DOUBLE ResonanceFreq1 - Set the calculation of resonances. Frequency 1, Hz
DOUBLE ResonanceFreq2 - Set the calculation of resonances. Frequency 2, Hz

- VARIANT_BOOL CalcMin - Set the calculation of the minimum spectrum:
  true - the calculation of the minimum spectrum is turned On;
  false - the calculation of the minimum spectrum is turned Off.

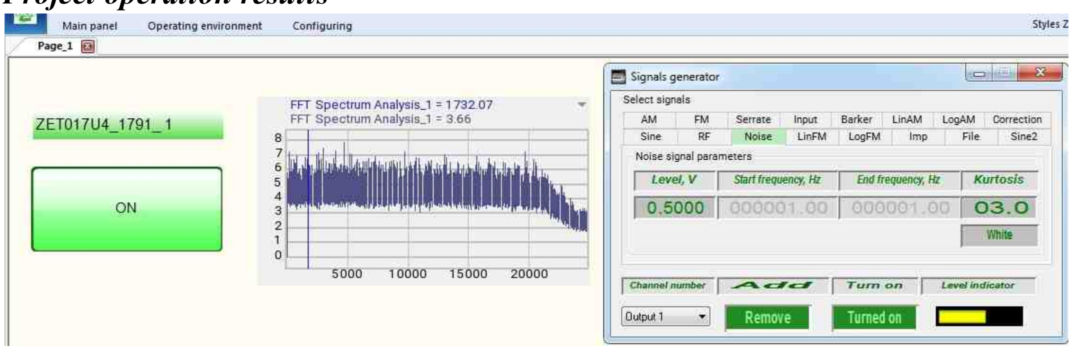
Sometimes the statistical analysis of signals requires the accumulation of data, and then the calculation of the maximum, minimum, average values.
Examples of obtaining maximum, minimum, average spectra, correlograms, etc. are presented in the projects (on the Example of a FFT Spectrum Analysis).

Example No. 1 - calculating the maximum values:

Project in SCADA ZETView



Project operation results

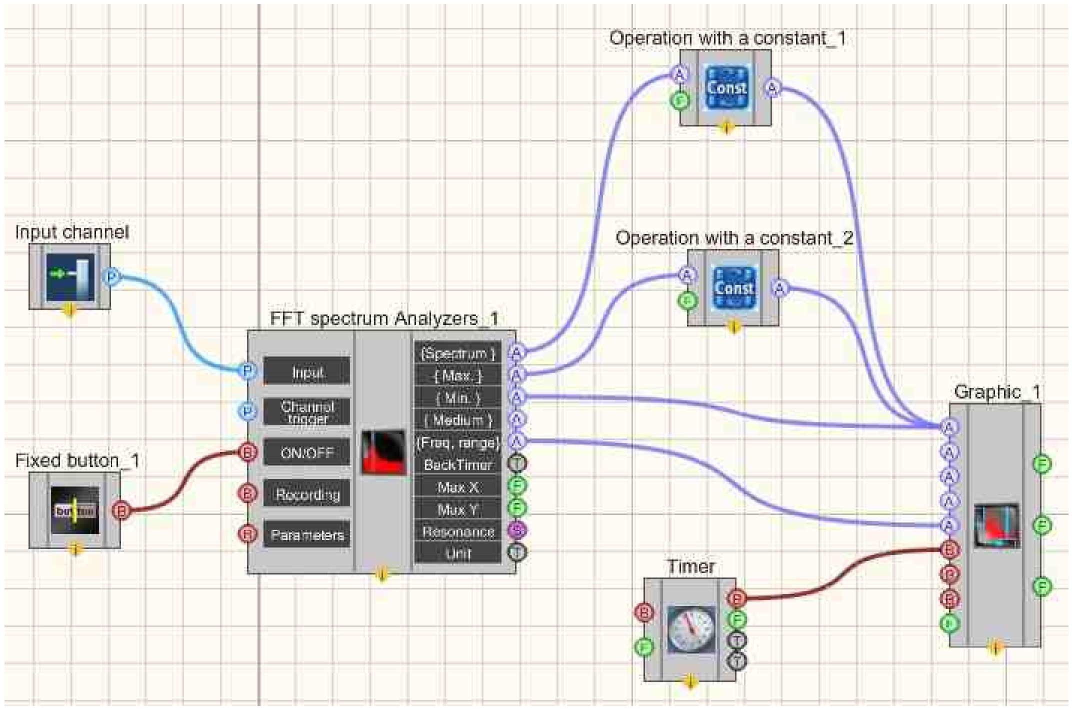


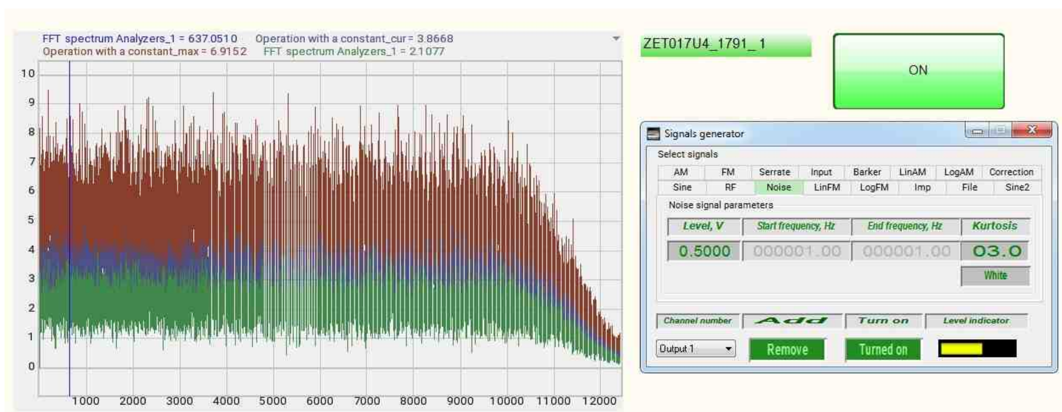
View the example in ZetView

Example No. 2 - calculating the minimum values:

View the example in ZetView

Example No. 3 - calculating the current range:





View the example in ZetView

Глава 4.Arithmetic

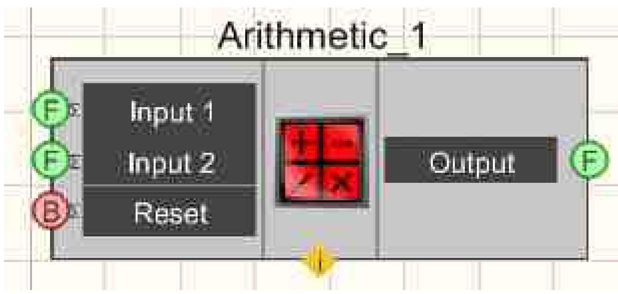
Components of "Arithmetic" section are used for arithmetic operations with numerical data:

Components of "Arithmetic" section are not displayed in operator mode and can be added to the project only in developer mode.
In operator mode, the following components are available:

4.1.Arithmetic

The component "**Arithmetic**" is designed a universal **Arithmetic** component that takes two values and performs one of the selected operations on them: addition, subtraction, multiplication, division, degree, logarithm, root maximum, minimum, modulus, and others.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input 1 - the input is the first number;
- Input 2 - the input is the second number;
- Reset - the Reset the value at inputs 1 and 2 to zero.

➢ Output:

- Output - the output is the result of an operation on numbers.

Assignment of component channels, depending on the selected action:

Action	Formula	Input No.1 (A)	Input No.2 (B)	Output channel (C)
Addition	A+B=C	First term	Second term	Sum
Subtraction	A-B=C	Minuend	Subtrahend	Difference
Mult	A·B=C	First multiplier	Seco	Prod

Action	Formula	Input No.1 (A)	Input No.2 (B)	Output channel (C)
iplication			multiplier	uct
Division	A/B=C	Dividend	Divider	Private
Degree	$A^B=C$	Base	Expon	Resu

Action	Formula	Input No.1 (A)	Input No.2 (B)	Output channel (C)
ee			ent	lt
Logarithm	$\log A_B=C$, $B^C=A$	Number	Base of the logarithm	Exponent
Root	$\sqrt[B]{A}=C$	Root number	Order	Resu

Action	Formula	Input No.1 (A)	Input No.2 (B)	Output channel (C)
			of root	lt
Maximum	Max(A,B)	Value 1	Value 2	Maximum value
Minimum	Min(A,B)	Value 1	Value 2	Minimum v

Action	Formula	Input No.1 (A)	Input No.2 (B)	Output channel (C)
				alue
Module	$|A,B| = \sqrt{A^2+B^2}$	Value 1	Value 2	Module

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- (Addition) - Select a calculation operation.
- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - the input is the first number to calculate the result (any number);
- Input 2 - the input is the second number to calculate the result (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

BSTR Operation - Select a calculation operation setting (string):

- Addition - the addition of the first and second number;
  Subtraction - the subtraction of the second number from the first;
  Multiplication - the multiplying the first and second number;
  Division - the division of the first number by the second;
  Degree - the first number to the power of the second;
  Logarithm - the logarithm of the first number to the base of the second number;
  Root - the root of the degree of the second number from the first number;
  Maximum - the maximum number of two;
  Minimum - the minimum number of two;
  Modulus - modulus of the first number.

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:

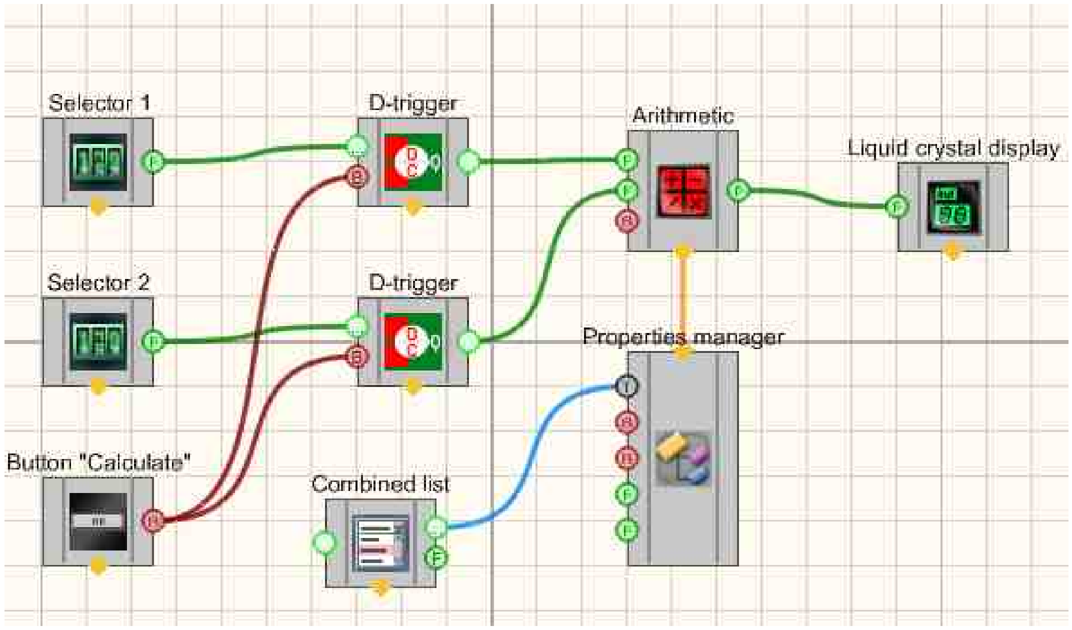
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:

- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

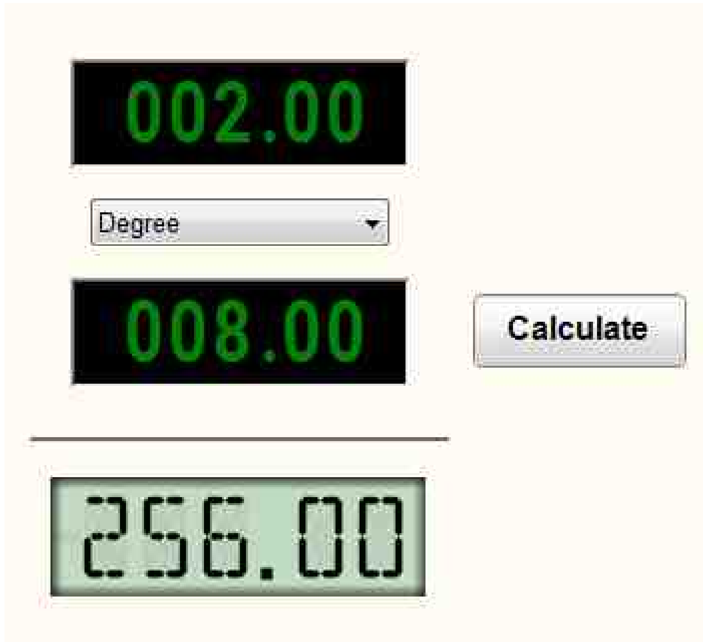
Example

Project in SCADA ZETView



This project considers the work of the "Arithmetic" 269. component. "Selectors 1 and 2" 428 are used to set the values on the first and second inputs of the "Arithmetic 269" component. The values from each of the selectors go to the "D-triggers" 732, "Non-fixed button" 694 "Calculate" button sends a control impulse to both triggers, passing the values set on the selectors to the first and second inputs of the "Arithmetic 269" component. Since "Arithmetic 269" is a multifunctional component, you can select the action to be performed with the received values on it. To do this, a "Properties manager" 999 is connected to the "Arithmetic 269" to a special input, which is used to display inputs for additional settings of the component. Let's connect the "Combined list" 414 component to the "Operation" input on the "Properties manager" 999, which is responsible for selecting the operation of the "Arithmetic 269" component, and enter in its properties the names of all actions available for the "Arithmetic 269" component. Next, Arithmetic 269 performs the selected action on the values from inputs 1 and 2 and sends the result to a "Liquid crystal display" 655.

Project operation results



First, the values are set on the selectors, an action is selected from the combo box, then you need to press the "Calculate" button, and then you can see the result on the digital indicator.

View the example in ZetView

4.2.Arccosine

The component "**Arccosine**" is designed to implement such an arithmetic operation as **Arccosine** on the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Value - the value of the arccosine.

➢ Output:
- Angle - the angle for the corresponding cosine value.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- angleType (degrees) - set the angle unit (degrees or radians).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

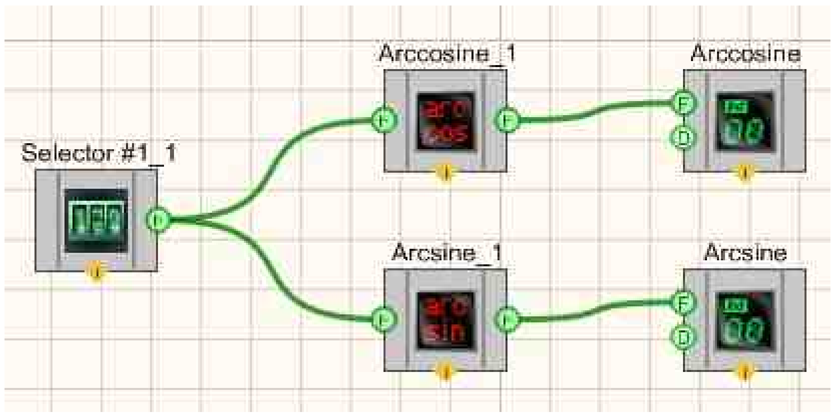
➢**Input:**
- Value - the value of the arccosine (any number).

➢**Custom common properties:**
- Are available by the link below. 154

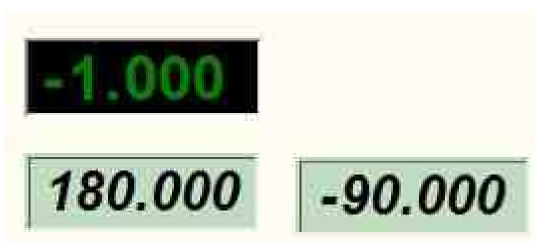
➢**Custom private properties (the default value is shown in parentheses):**
- BSTR AngleType - Set the angle unit: degrees; radians (string).

Example

Project in SCADA ZETView



Project operation result



View the example in ZetView

4.3.Arcsine

The component "**Arcsine**" is designed to implement such an arithmetic operation as **Arcsine** on the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Value - the value of the arcsine.

➢ Output:
- Angle - the angle for the corresponding sine value.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- angleType (degrees) - set the angle unit (degrees or radians).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Value - the value of the arcsine (any number).

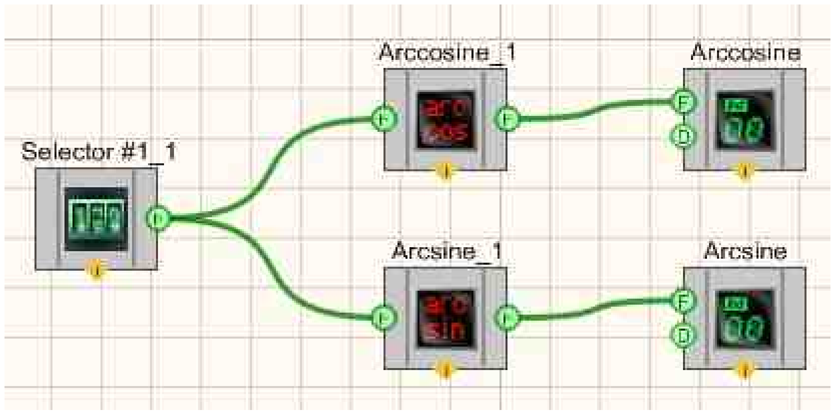
➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

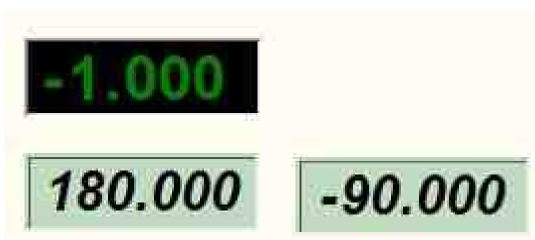
- BSTR AngleType - Set the angle unit: degrees; radians (string).

Example

Project in SCADA ZETView



Project operation result



View the example in ZetView

4.4.Arctangent

The component "**Arctangent**" is designed to implement such an arithmetic operation as **Arctangent** on the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
➢ Input X - the X coordinate for calculating the arc tangent value.
➢ Input Y - the Y coordinate for calculating the arc tangent value.
➢ Output:
- Output - the arctangent of the operand is formed at the output.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FirstReaction (True) - set the reaction of changing the output signal to changing the values at the input X:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.
- SecondReaction (True) - set the reaction of changing the output signal to changing the values at the input Y:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.
- First (0) - the default value of the first operand (X).
- Second (0) - the default value of the second operand (Y).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Input 1 - the X coordinate for calculating the arc tangent value (any number).
- Input 2 - the Y coordinate for calculating the arc tangent value (any number).

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL FirstReaction - Sets the value of the reaction to a data change on input X:

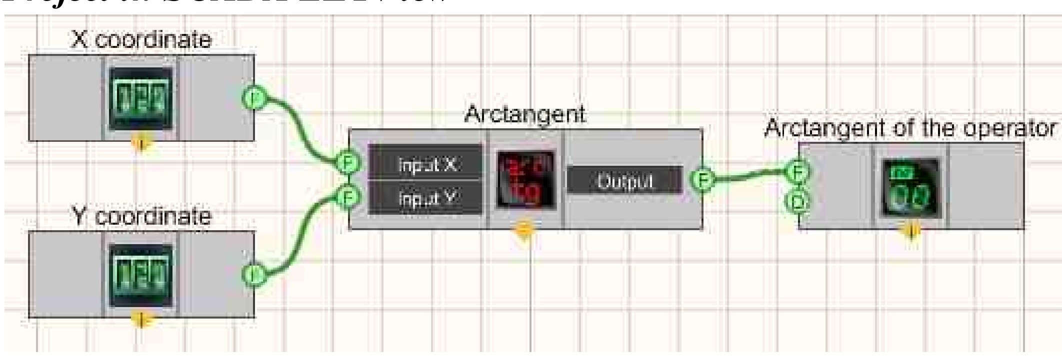
- true - the output value is given when data arrives at input X;
  false - the output value is not given when input X is received.

VARIANT_BOOL SecondReaction - Set the value of the reaction to a change in data at input Y:

- true - the output value is given when data arrives at input Y
  false - the output value is not given when data is received at input Y
- FLOAT First - Set the X coordinate value for calculating the arc tangent value (any number).
- FLOAT Second - Set the Y coordinate for calculating the arc tangent value (any number).

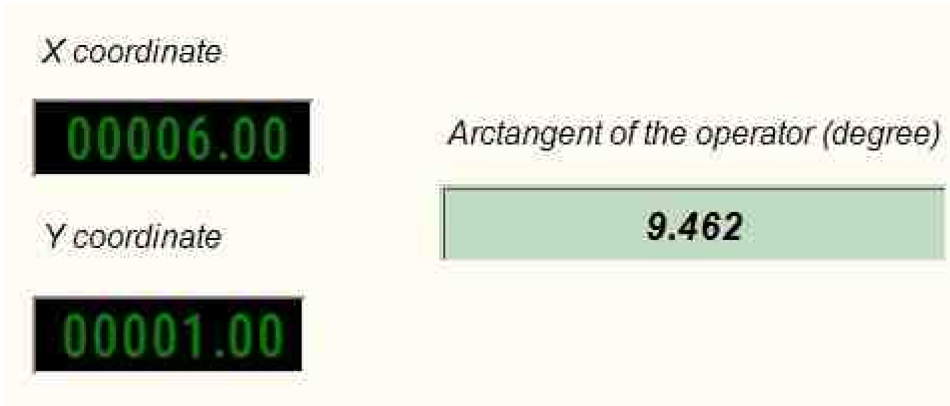
Example

Project in SCADA ZETView



This project shows the operation of the Arctangent 281 component performs the operation of the arctangent of two numbers. Selectors 428 are used to enter numerical values to the input of the Arctangent 281 component. A Digital Indicator 682 is used for graphical output of the resulting.

Project operation result

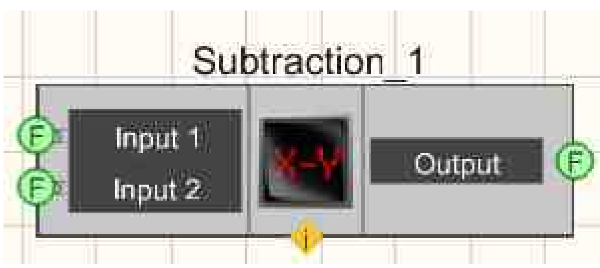


View the example in ZetView

4.5.Subtraction

The component "**Subtraction**" is designed to implement such an arithmetic operation as **Subtraction** on the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input 1 - the input is the first number;
- Input 2 - the input is the second number;

➢ Output:

- Output - the difference of numerical values is formed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - the input is the first number to calculate the result (any number);
- Input 2 - the input is the second number to calculate the result (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:

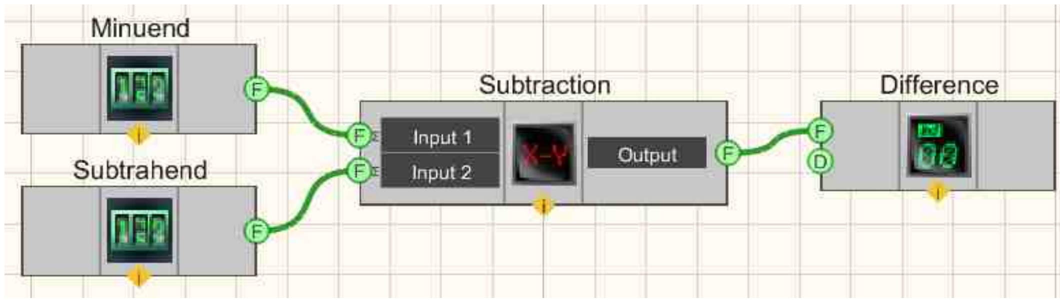
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:

- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

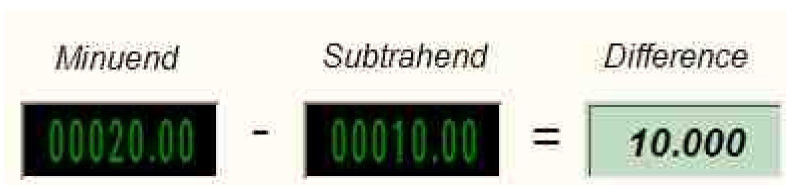
Example

Project in SCADA ZETView



This project shows the operation of the Subtraction 284 component performs the operation of the difference of two numbers. Selectors 428 are used to enter numerical values to the input of the Subtraction 284 component. A Digital Indicator 682 is used for graphical output of the resulting.

Project operation result



4.6.Division

The component "**Division**" is designed to implement such an arithmetic operation as division on the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input 1 - the input is the first number;
- Input 2 - the input is the second number;

➢ Output:

- Output - the quotient of numbers is formed.

Custom common properties:

➢Total properties (environment):

- [Are available by the link below.](154) 154

➢Custom private properties (the default value is shown in parentheses):

- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Input 1 - the input is the first number to calculate the result (any number);
- Input 2 - the input is the second number to calculate the result (any number).

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:

- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:

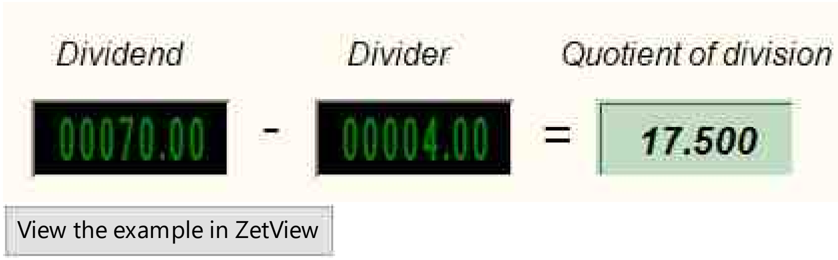
- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

Example

Project in SCADA ZETView

This project shows the operation of the Division 286 component performs the operation of dividing two numbers.. Selectors 428 are used to enter numerical values to the input of the Division 286 component. A Digital Indicator 682 is used for graphical output of the resulting.

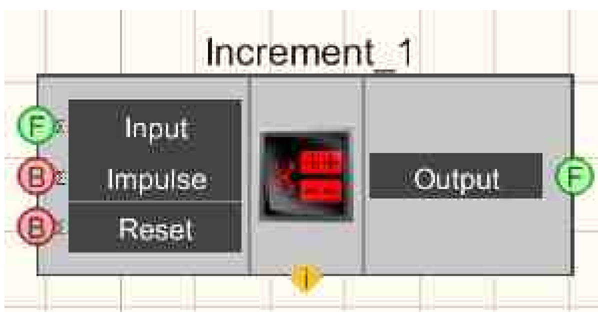
Project operation result



4.7.Increment

The component "**Increment**" is designed to implement such an arithmetic operation as adding a constant to the numerical value of the input signal.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - current value.
- Impulse - is an input contact, when the logic level changes, a constant is added to the numerical value of the input channel.
- Reset - input contact, the value coming to the input of the component, when changing the logic level, is reset to zero.

➢ Output:

- Output - the result of the increment is formed..

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Value (0) - current value.
- Step (1) - Step of increment.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - current value (any number).

➢**Custom common properties:**

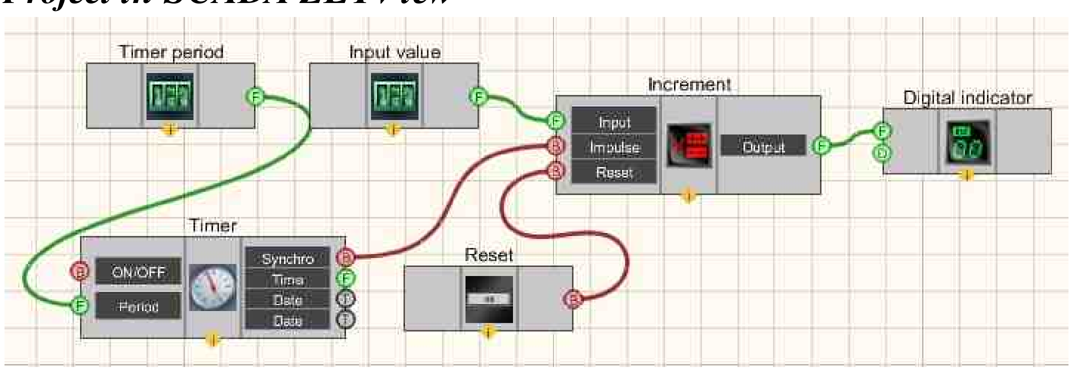
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Value - Set the current value (any number).
- FLOAT Step - Set the increment step value (any number).

Example

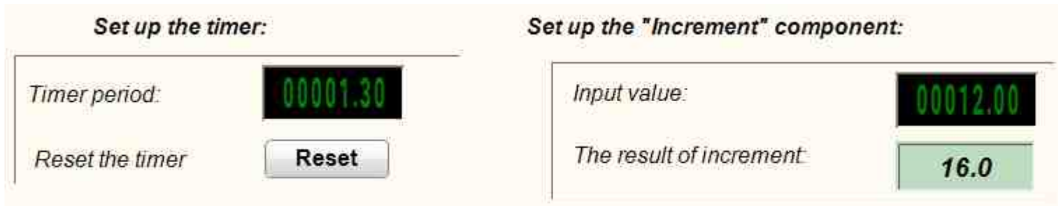
Project in SCADA ZETView



This project shows the operation of the Increment 289 constantly adds one to the numerical value of the input channel. Timer 370 it is used to generate input impulses

Increment 289, upon which an increment of the input value is made. Selectors 428 they are used to set the input value and the timer period. A Digital Indicator 682 is used for graphical output of the resulting.

Project operation result



View the example in ZetView

4.8.Integrator

The component "**Integrator**" is designed to implement such a mathematical operation as integration.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - the value for calculating the result.
- Reset - the input contact, when changing the logic level, resets to 0 the value received at the input of the component.

➢ Output:
- Output - the result of integration is formed, which is the sum of all previous values and the current one.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

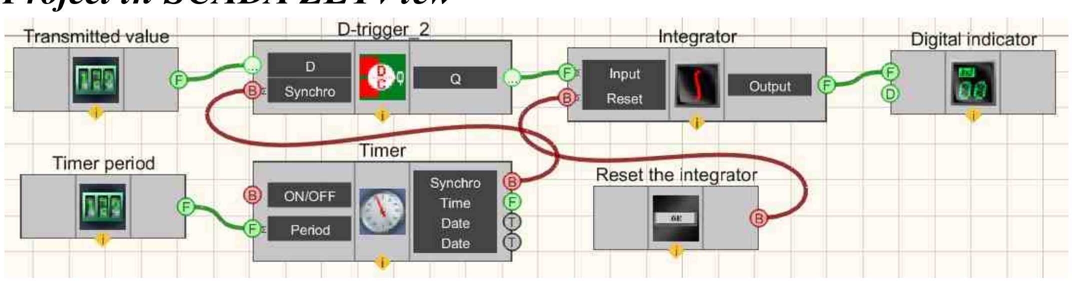
- Input - current value (any number).

➢**Custom common properties:**

- Are available by the link below. 154

Example

Project in SCADA ZETView



This project shows the operation of the Integrator 291 constantly sums up the input value that is set Selector 428. Timer 370 sends sync impulses to the input D-trigger 732, to ensure the frequency of receipt of numerical values at Integrator 291. Selector 428 the timer is used to set the frequency of these sync impulses. Button Reset 694 is used to return the output value Integrator 291 to the initial value. A Digital Indicator 682 is used for graphical output of the resulting.

Project operation result

4.9.Cosine

The component "**Cosine**" is designed to implement an arithmetic operation at the output of the **Cosine** of the angle.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Angle - Angle value.

➢ Output:
- Output - the value of the cosine of the angle is formed.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- angleType (degrees) - Sets the unit of measure for the angle (degrees or radians).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Angle - Angle value (any number).

➤**Custom common properties:**

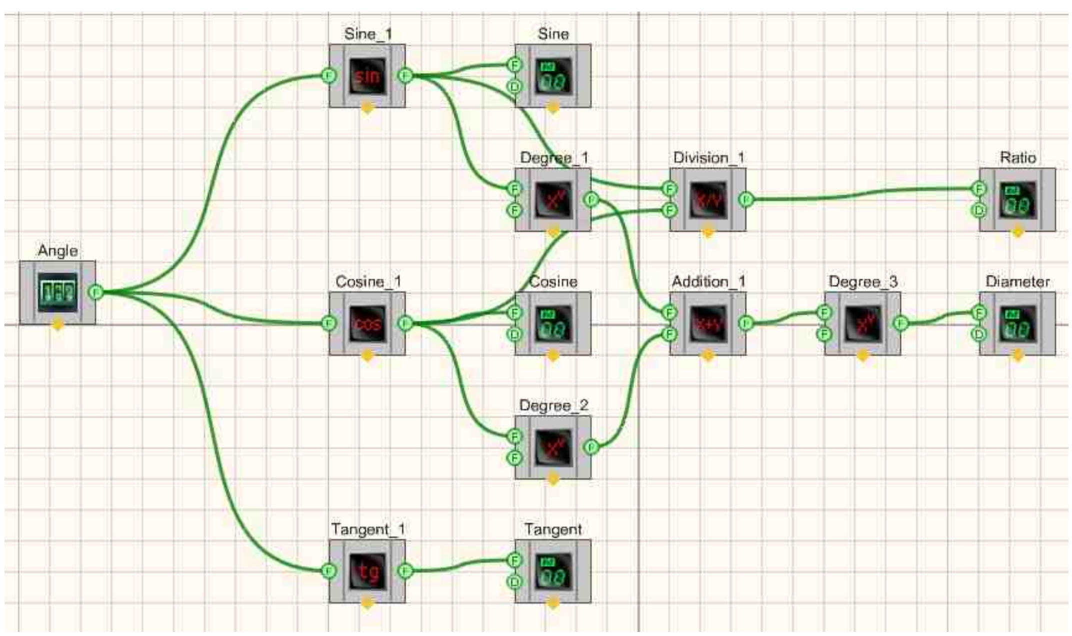
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR AngleType - Set the angle unit: degrees; radians (string).

Example

Project in SCADA ZETView

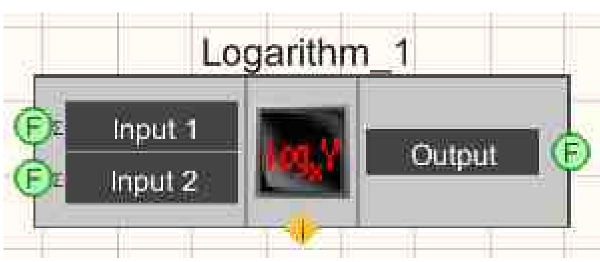


Project operation result

4.10.Logarithm

The component "**Logarithm**" is designed to implement such a mathematical operation as the logarithm.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input 1 - the input channel of the component to which the base of the logarithm is applied.
- Input 2 - input contact, the number under the logarithm (the number when taking the logarithm of which in the base (Input 1) will get the output value).

➢ Output:

- Output - the logarithm of the second operand to the base of the first is formed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - first operand default value
- Second (1) - second operand default value.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Input 1 - the input is the first number to calculate the result (any number);
- Input 2 - the input is the second number to calculate the result (any number).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:

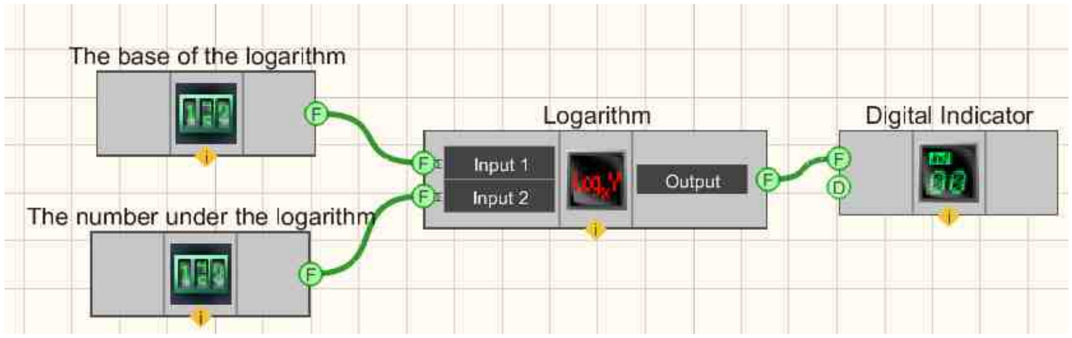
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:

- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

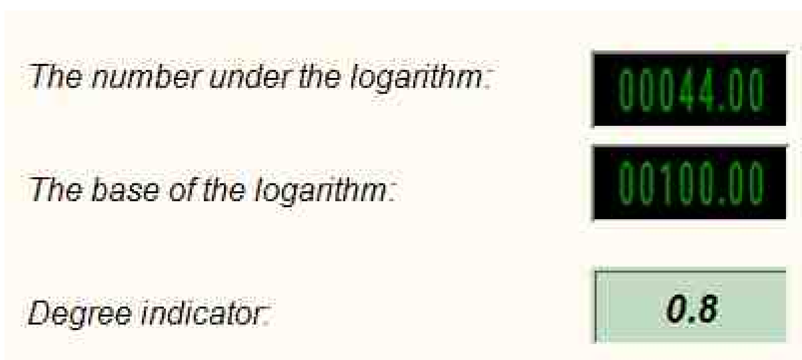
Example

Project in SCADA ZETView



This project shows the operation of the Logarithm[295] is used to find the value of the logarithm of the value coming from the Selector[428] (Number under the logarithm) to Input-2 by the base specified by the Selector[428] (Base) to Input-1. A Digital Indicator[682] is used for graphical output of the resulting.

Project operation result



View the example in ZetView

4.11.Maximum/Minimum level

The component "**Maximum/Minimum level**" is designed to select the maximum or minimum level among the input values.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Input - input value.
- ....
- Input - input value.
  The number of input values can be changed starting from 2.

➤ Output:
- Output - the result is the maximum or minimum level. The result will be obtained only when all input values are filled.

Custom common properties:

➤Total properties (environment):
- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):
- NumberInput (3) - set the number of inputs;
- MinMax (maximum) - set the level selection to maximum or minimum;
- BuffersSizeMax (10) - set the maximum size of input data buffers;
- OperatingMode (Synchronous) - set the choice of operating mode values: Synchronous; Asynchronous.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

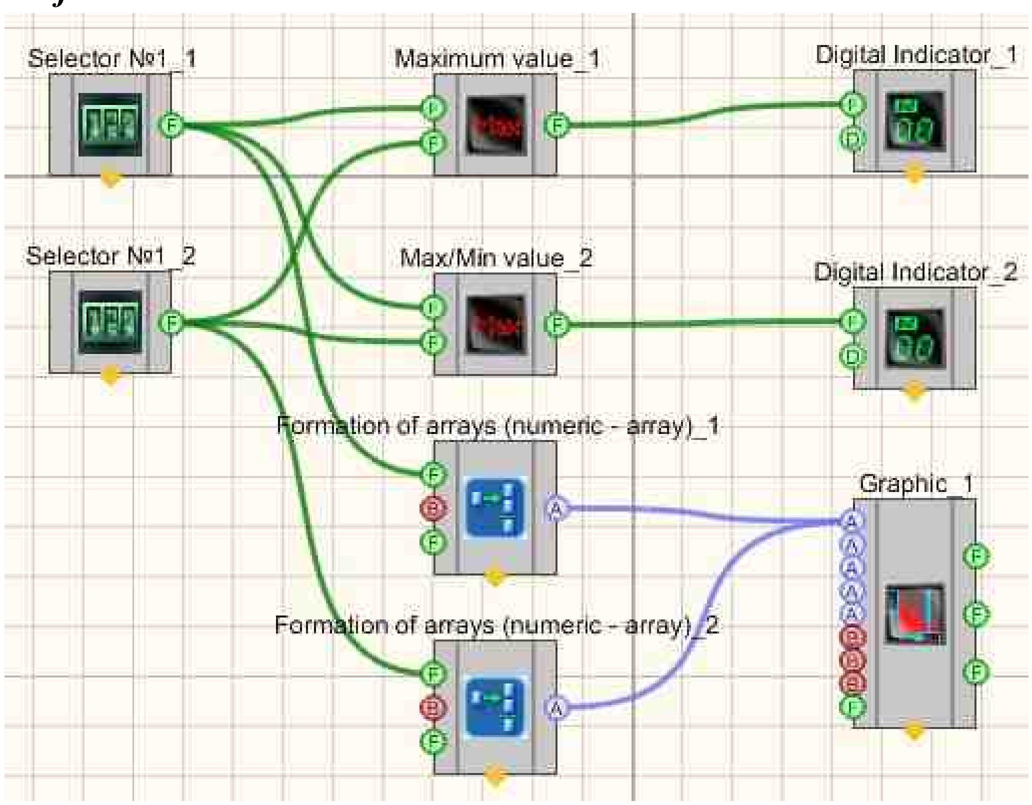
➢**Input:**
- Input - Input value (any number).
- ....
- Input - Input value (any number).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- LONG NumberInput - Set the number of inputs.
- BSTR MinMax - Set the level selection maximum or minimum;
- LONG BuffersSizeMax - Set the maximum size of input data buffers.
- BSTR OperatingMode - Set the operating mode: Synchronous; Asynchronous.

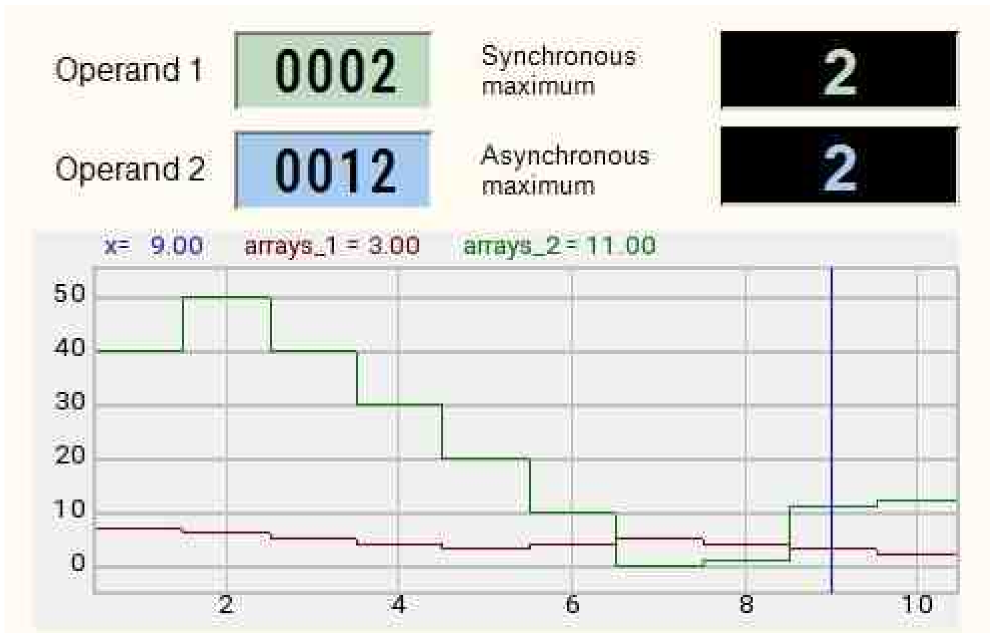
Example

Project in SCADA ZETView



This project shows the operation of the Maximum/Minimum level 297 calculates and displays the Maximum values on the digital displays set via Operand 1 and Operand 2 in Synchronous and Asynchronous operation mode.

Project operation result

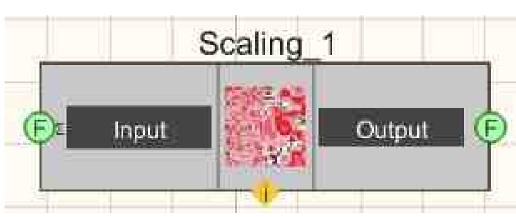


View the example in ZetView

4.12.Scaling

The component "**Scaling**" is designed for scaling.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - the input is a number for scaling.

➢ Output:
- Output - the result of scaling at the output.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FLOAT Min (0.000000) - set the minimum value when scaling.
- FLOAT Max (0.000000) - set the maximum value when scaling.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input - the input is a number for scaling (any number).

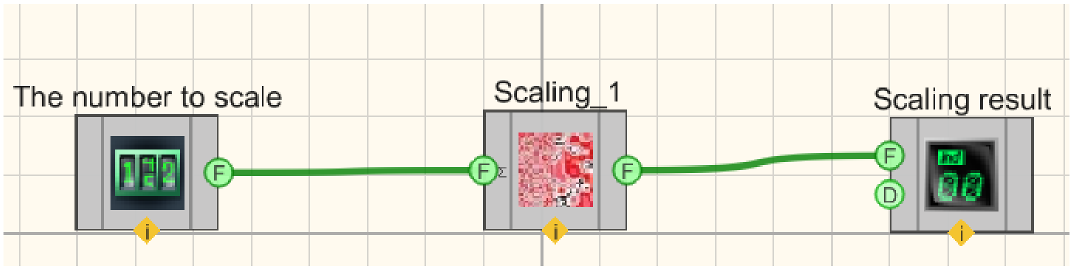
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Min - Set the minimum value when scaling.
- FLOAT Max - Set the maximum value when scaling.

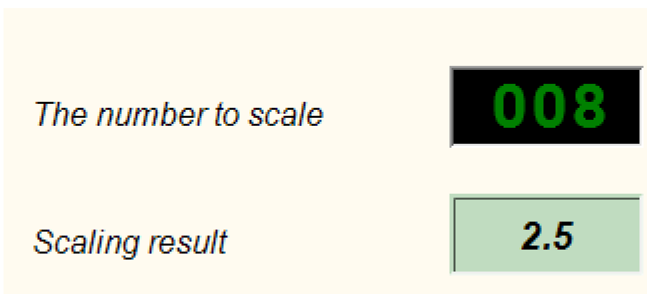
Example

Project in SCADA ZETView



This project shows the operation of the Scaling 300 is used a get from number operation for Scaling 300 in the channel. The Selector 428 is used to enter numerical values at the input of the Scaling 300 component. A Digital Indicator 682 is needed for graphical output of the scaling result.

Project operation result



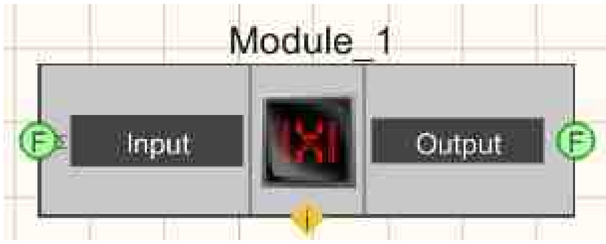
View the example in ZetView

4.13.Module

The component "**Module**" is designed to implement such an arithmetic operation as obtaining the absolute value of a quantity.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- Input - operand is supplied.

➢ Output:
- Output - the modulus of the number is formed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

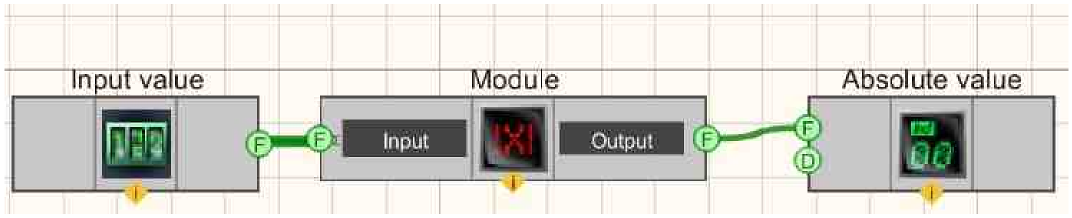
Setting:

➢**Input:**
- Input - Value for module calculation (any number).

➢**Custom common properties:**
- Are available by the link below. 154

Example

Project in SCADA ZETView



This project shows the operation of the Module 302 is used the operation of obtaining the absolute value of the numeric data in the channel. Selector 428 are used to enter numerical values at the input of the Module 302 component. Digital Indicator 682 is needed for graphical output of absolute values.

Project operation result

4.14.Rounding

The component "**Rounding**" is designed to implement such an arithmetic operation as rounding..

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - an operand is fed into the input.

➢ Output:
- Output - a rounded value is generated.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Precision (0) - number of decimal places (rounding precision).
- Method (rounding) - set rounding method: rounding; to the big side; down; cutting off the fractional part;

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input - an operand (any number). is fed into the input.

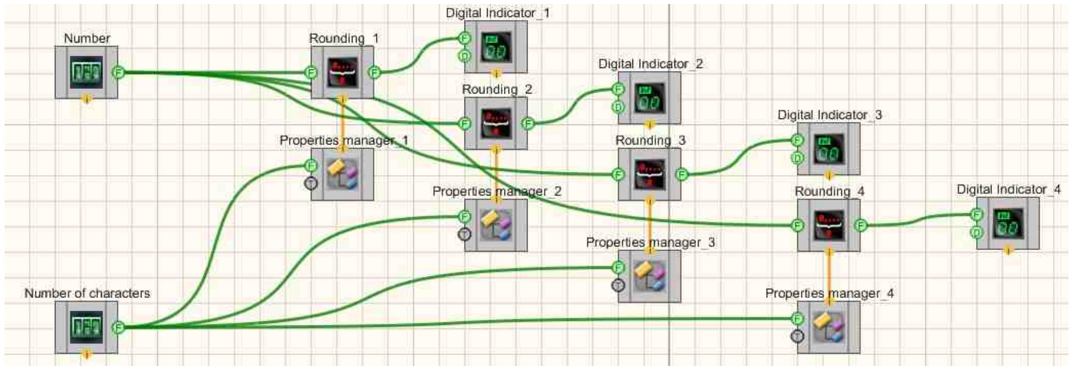
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

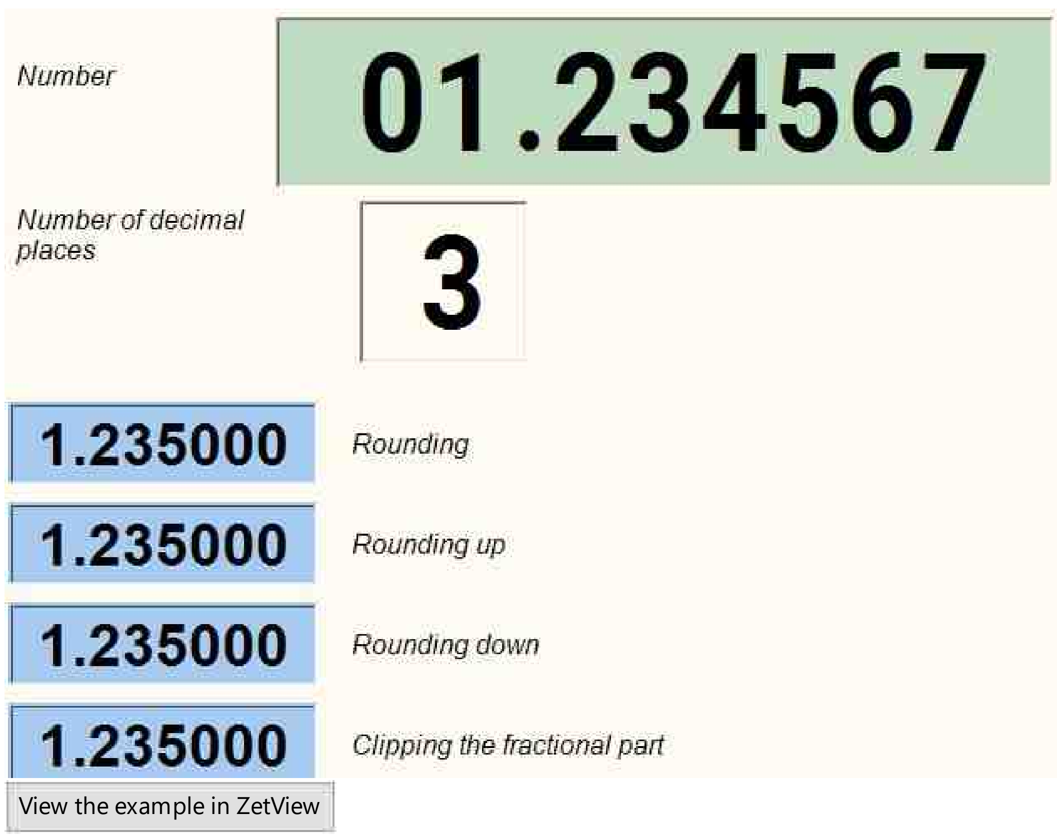
- LONG Precision - Set the value of the number of decimal places after rounding (from 0 to 6).
- BSTR Method - Set the rounding method: rounding; to the big side; down; cutting off the fractional part (string).

Example No. 1

Project in SCADA ZETView



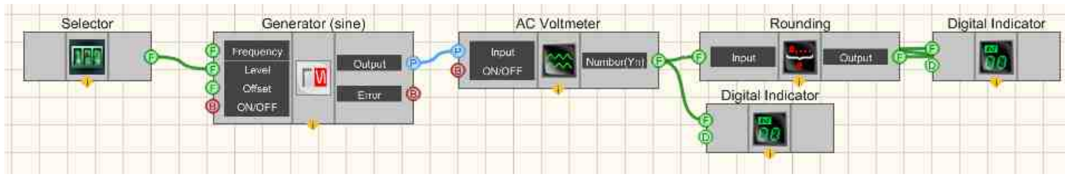
Project operation result



Example No. 2

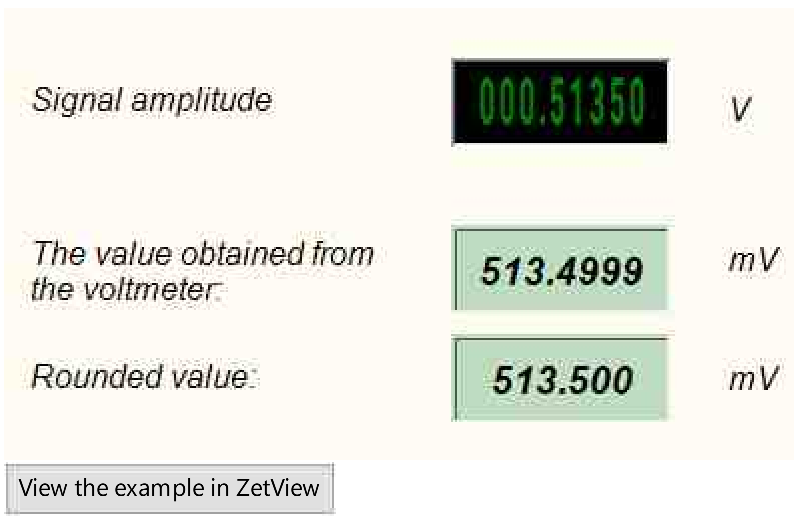
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



The Selector 428 is used to enter the value of the amplitude of the Sine signal. The AC voltmeter 585 measures the RMS value of the signal. This project shows the operation of the Rounding component performs a Rounding 304 operation to one decimal place after the busy one. A Digital Indicator 682 is needed for graphical output of an already rounded value. the Generator (sine) 530 is needed to obtain a harmonic signal. The second Digital Indicator 682 is needed to display the voltage value taken directly from the voltmeter.

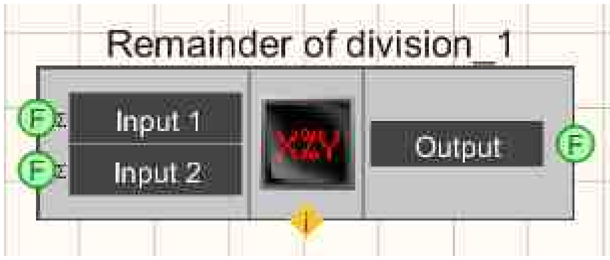
Project operation result



4.15.Remainder of division

The component "**Remainder of division**" is designed to implement such an arithmetic operation as getting the remainder of a division.

Appearance of the component:

<table>
<tr><td>Developer interface</td><td>Operator interface</td></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:
- Input 1 - the input is the first number;
- Input 2 - the input is the second number;

➢ Output:
- Output - the remainder of the division is formed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input 1 - the input is the first number to calculate the result (any number);
- Input 2 - the input is the second number to calculate the result (any number).

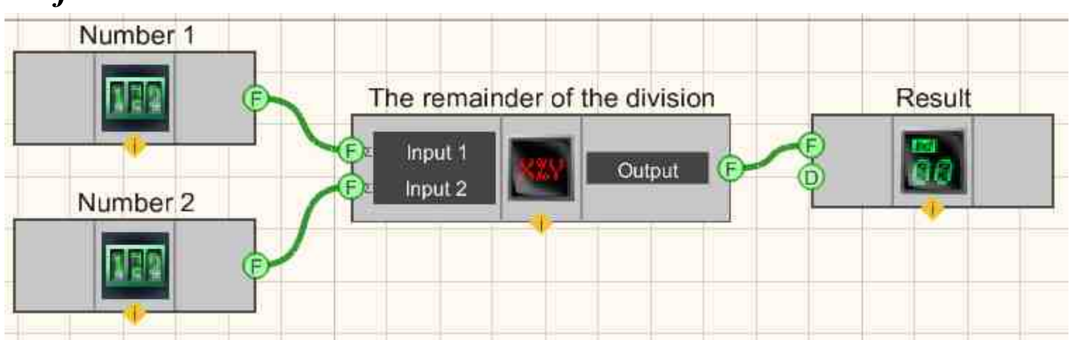
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:
- true - the output value is given when data arrives at input 1;
false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:
- true - the output value is given when data arrives at input 2;
false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

Example

Project in SCADA ZETView



This project shows the operation of the Remainder of division 307 component allows us to get the remainder of dividing the number 1 by the number 2. The Selector 428 is used to enter numerical values into the input of the Remainder of division 307 component. A Digital Indicator 682 is needed for graphical output of an already rounded value.

Project operation result

View the example in ZetView

4.16.Falling into range

The component "**Falling into range**" is designed you to set the upper and lower limits of the range, checking whether the value received at the input is within the specified range. The output is 1- if the value is in the range and 0 otherwise.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Value - The value to check for hitting the range.

➢ Output:

- Result - the result of the check for falling into the range, from which the value "1" is supplied in the case when the input number falls into the specified range, "0" - in the case when the input number does not fall into the specified range.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- fRangeStart (0) - the lower limit of the range.
- fRangeEnd (0) - the upper limit of the range.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - Value to check for hitting the range (any number).

➢**Custom common properties:**

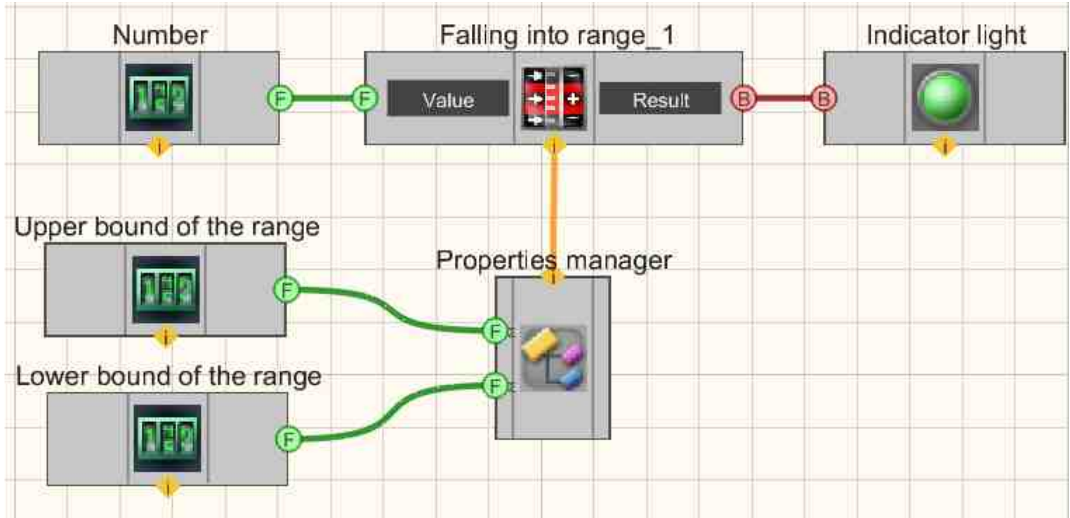
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT fRangeStart - Set the value of the lower limit of the range (any number).
- FLOAT fRangeEnd - Set the value of the upper limit of the range (any number).

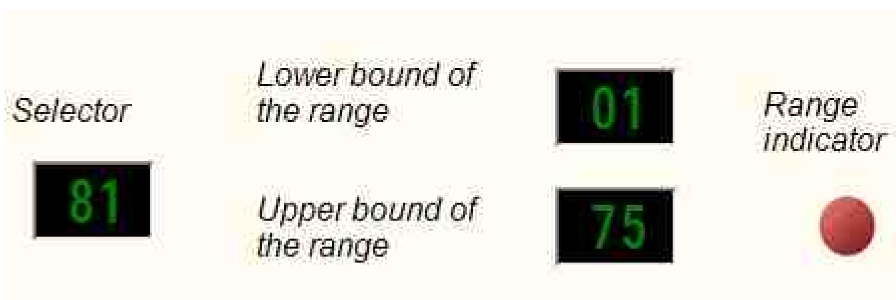
Example

Project in SCADA ZETView



This project shows the operation of the Falling into range 310 component is considered. Selectors 428 are used to set the input value for the Falling into range 310 component, as well as the upper and lower range values. Property Manager 999, is used to display additional inputs for set the properties of the Falling into range 310 component. The result is fed to the input of the Light indicator 669 component. If it falls within the range, it will turn green, otherwise it will turn red.

Project operation result



View the example in ZetView

4.17.Difference

The component "**Difference**" is designed to implement such a mathematical operation as finding the difference of two values at the same input.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - Value for calculating the result.

➢ Output:
- Output - a value is generated that is the difference between the current and previous values at the input.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

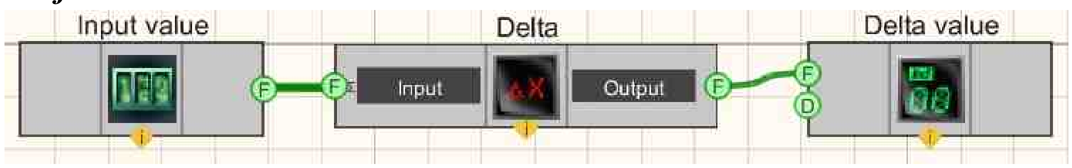
- Input - Value for calculating the result (any number).

➢**Custom common properties:**

- Are available by the link below. 154

Example

Project in SCADA ZETView



This project shows the operation of the Difference 313 component is used to find the difference between the current and previous values. The Selector 428 is needed to set the input values. The Digital Indicator 682 is used for graphical output of the resulting.

Project operation result

4.18.Sine

The component "**Sine**" is designed to implement an arithmetic operation on the output of the sine of an angle.

Appearance of the component:

<table>
<tr><td>Developer interface</td><td>Operator interface</td></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:

- Angle - Angle value.

➢ Output:

- Output - the value of the sine of the angle is formed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- angleType (degrees) - set the angle unit (degrees or radians).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Angle - Angle value (any number).

➢Custom common properties:

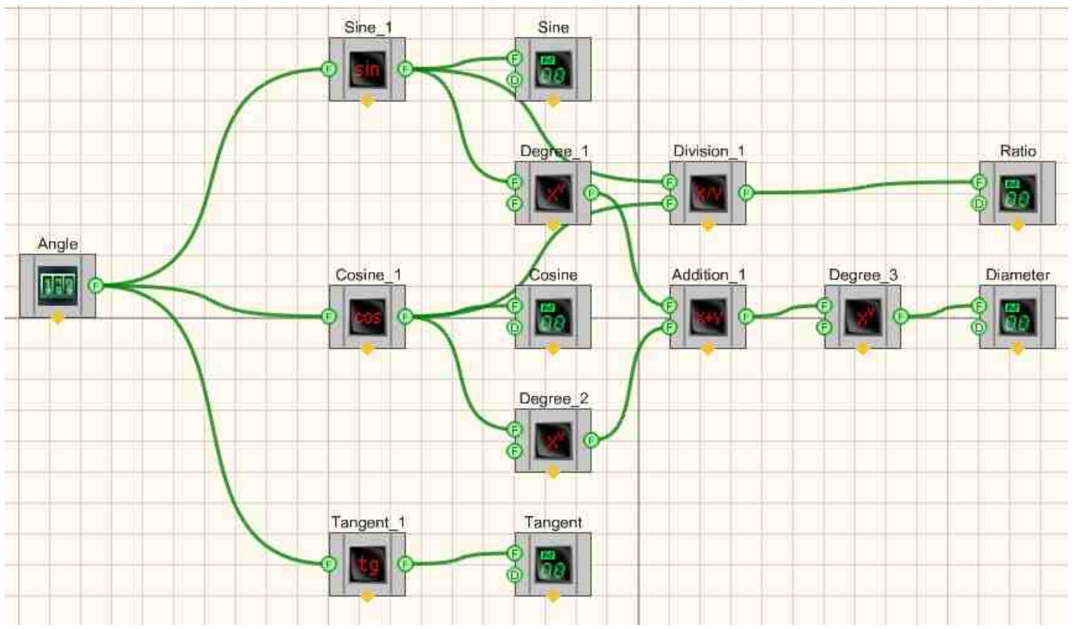
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

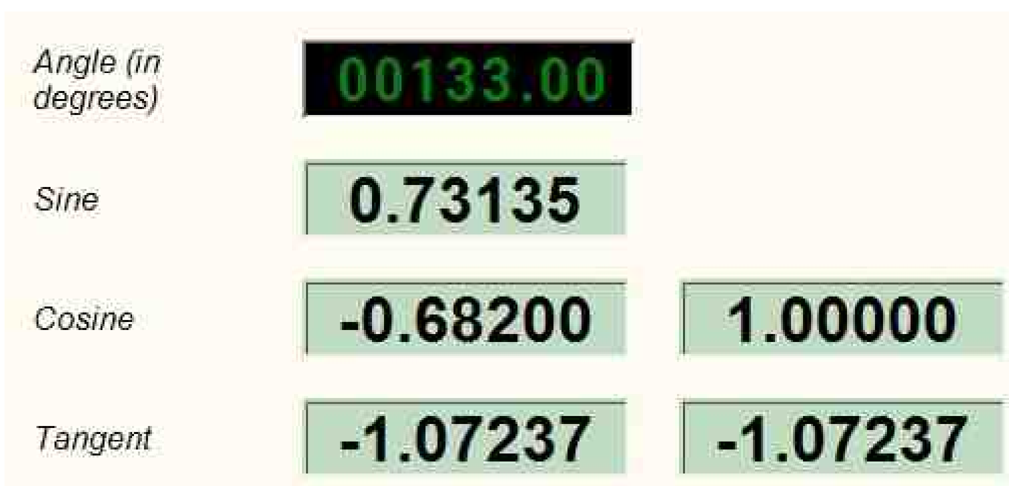
- BSTR AngleType - Set the angle unit: degrees; radians (string).

Example

Project in SCADA ZETView



Project operation result



View the example in ZetView

4.19.Addition

The component "**Addition**" is designed to implement such a mathematical operation as the sum of two numerical values.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - the input is the first number;
- Input 2 - the input is the second number;

➢ Output:
- Output - the sum of two input numerical values is formed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input 1 - the input is the first number to calculate the result (any number);
- Input 2 - the input is the second number to calculate the result (any number).

➢**Custom common properties:**
- Are available by the link below. 154

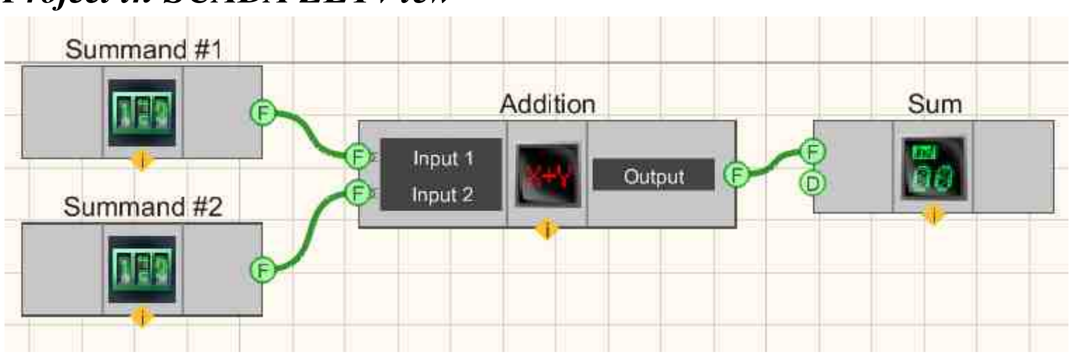
➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:
- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

Example No. 1

Project in SCADA ZETView



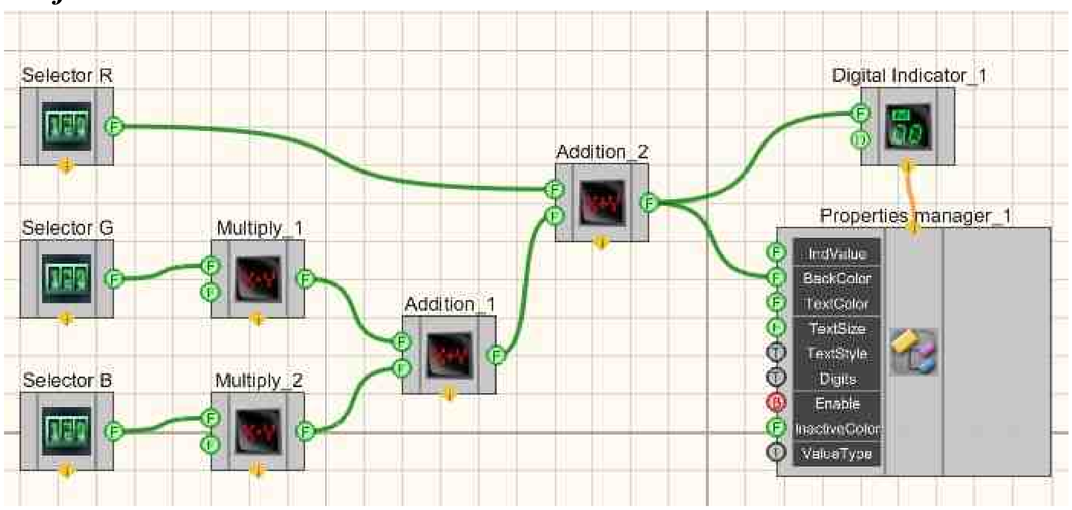
This project shows the operation of the Addition 314 component is used to find the sum of two input values. Selectors 428 are needed to set numerical values for terms. The Digital Indicator 682 is used for graphical output of the resulting.

Project operation result

View the example in ZetView

Example No. 2

Project in SCADA ZETView



This project shows the operation of the Addition 314 component is used to find the sum of two input values. Selectors 428 are needed to set numerical values for terms. The Digital Indicator 682 is used to graphically represent the output values.

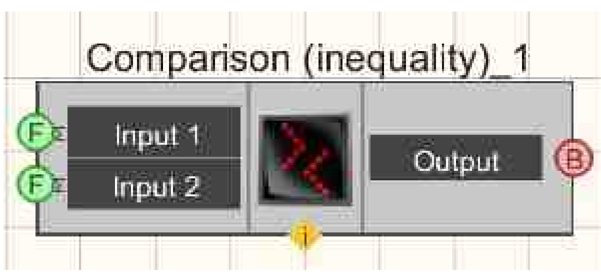
Project operation result

View the example in ZetView

4.20.Comparison (inequality)

The component "**Comparison (inequality)**" is designed to implement such a mathematical operation as comparing two numerical values.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - The value of the first operand to compare.
- Input 2 - The value of the second operand to compare.

➢ Output:
- Output - a high logic level is generated if the value at input No.1 is greater than or equal to the value at input No.2, otherwise it is low.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - The value of the first operand for calculating the result (any number).
- Input 2 - The value of the second operand for calculating the result (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:

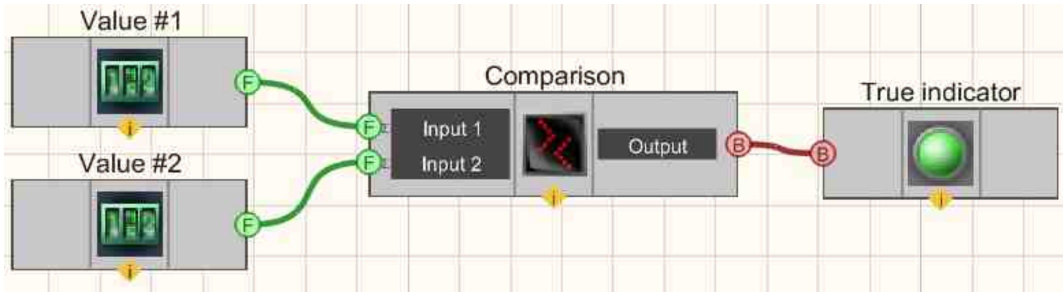
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:

- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

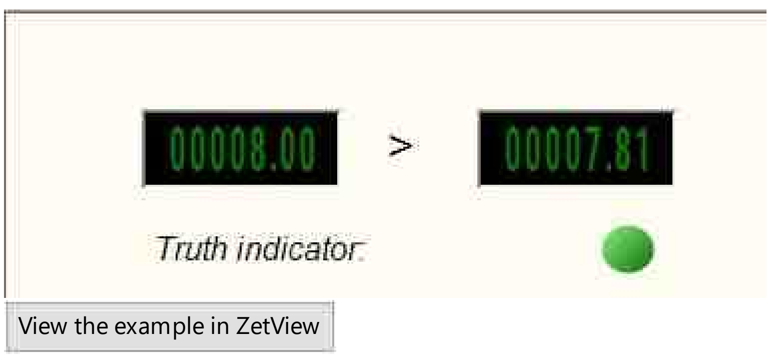
Example

Project in SCADA ZETView



This project shows the operation of the Compare Comparison (inequality) [320] component is used to compare two input values with each other. Selectors [428] are needed to assign numerical values to the compared values. The Light indicator [669] is used to visually represent the result of the comparison. If the value of 1 is greater than or equal to the value of 2, then the indicator lights up green, otherwise - red.

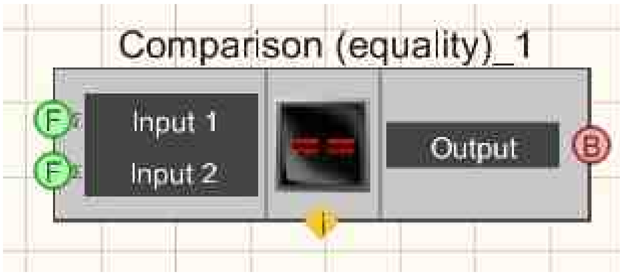
Project operation result



4.21.Comparison (equality)

The component "**Comparison (equality)**" is designed to implement such a mathematical operation as checking for equality of two numerical values.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - The value of the first operand to compare.
- Input 2 - The value of the second operand to compare.

➢ Output:
- Output - a high logic level is formed if the value at input No.1 is equal to the value at input No.2, otherwise it is low.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - The value of the first operand for calculating the result (any number).
- Input 2 - The value of the second operand for calculating the result (any number).

➢**Custom common properties:**

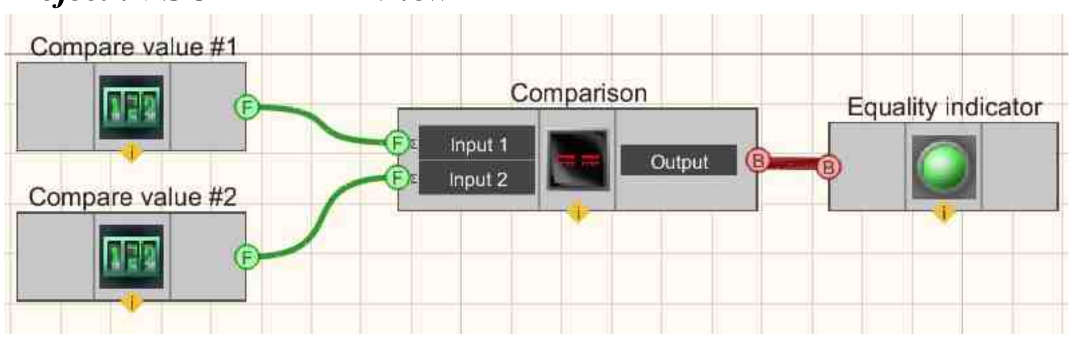
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.
  VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:
- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

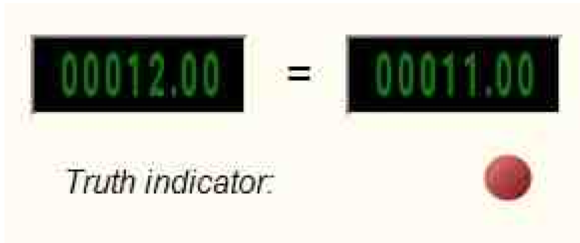
Example

Project in SCADA ZETView



This project shows the operation of the Comparison (equality) 322 component is used to compare two input values with each other. Selectors 428 are needed to assign numerical values to the compared values. The indicator light is used to visually represent the result of the comparison. If the input values are equal, the Light indicator 669 up green, otherwise - red.

Project operation result

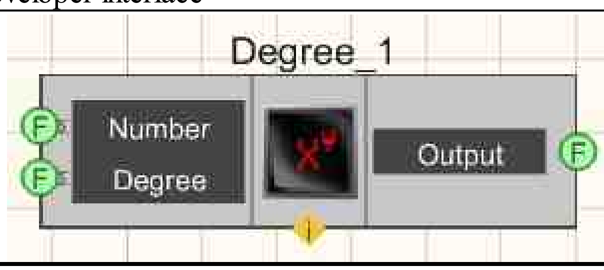


View the example in ZetView

4.22.Degree

The component "**Degree**" is designed to implement such a mathematical operation as raising a number to a power.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Number - the number raised to a power is given.
- Degree - the degree to which the number will be raised is given.

➤ Output:

- Output - the first operand is formed to the power of the second.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.

- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Input 1 - the number being raised to a power is given (any number).
- Input 2 - the degree to which the number will be raised is given (any number).

➢Custom common properties:
- Are available by the link below. 154

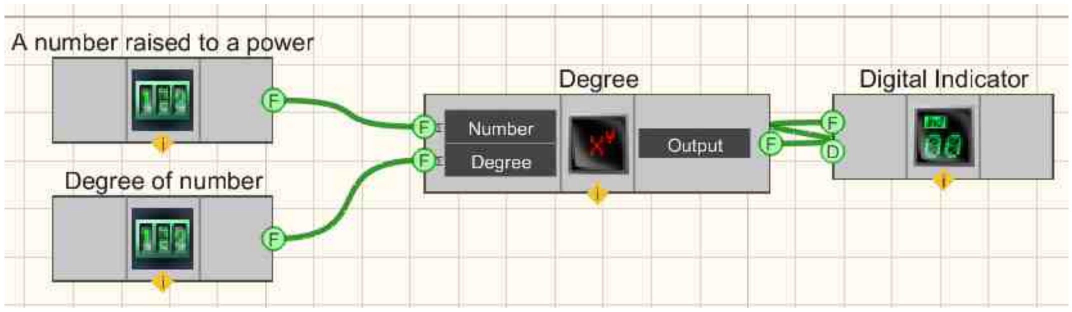
➢Custom private properties (the default value is shown in parentheses):

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:
- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

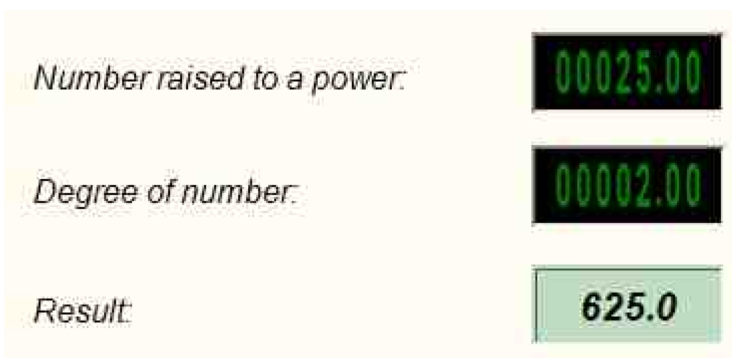
Example

Project in SCADA ZETView



This project shows the operation of the Degree[325] component is used to find the value of a number after raising it to a power. Selectors[428] are needed to enter a number and the desired degree. The Digital Indicator[682] is used for graphical output of the resulting.

Project operation result

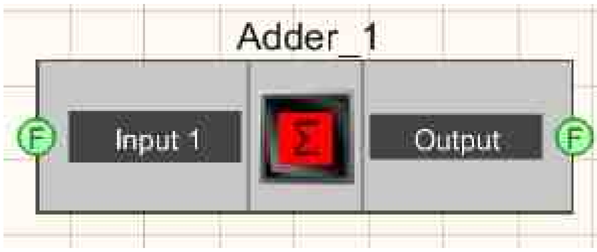


View the example in ZetView

4.23.Adder

The component **"Adder"** is designed to implement such a mathematical operation as a one-time summation of incoming values.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input 1 - the first input number to sum.
- Input 2 - the second input number to sum.

  ...
- Input N - Nth input number to sum.

➢ Output:

- Output - the sum of all previous values and the current one is formed.

Note:
The adder component outputs a value only when the value of all inputs changes.

Custom common properties:
➢Total properties (environment):

- [Are available by the link below.](154)

➢Custom private properties (the default value is shown in parentheses):

- Input (1) - set the number of inputs.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

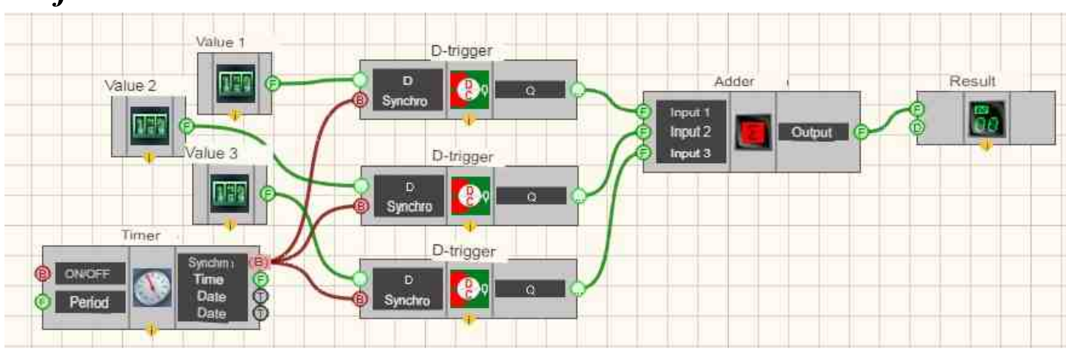
- Input 1 - the first input number to sum (any number).
- Input 2 - the second input number to sum (any number).
- Input N - Nth input number to sum (any number).

➤**Custom common properties:**
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**
- LONG Input - Set the number of totalizer inputs (from 1 to 50).

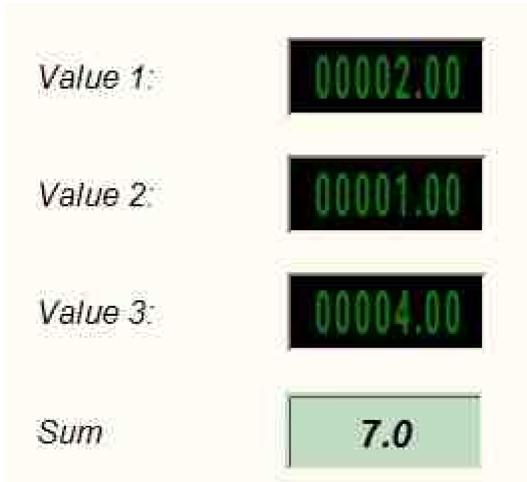
Example

Project in SCADA ZETView



This project shows the operation of the Adder 327 component adds up all the input values, which is specified by the Selectors 428. The Timer 370 sends clock impulses to the input of D-triggers to ensure the frequency of receipt of numerical values in the Adder 327 (because the addition of signals occurs only when the values of the signals at the input contacts of the adder change simultaneously). The Digital Indicator 682 is used for graphical output of the resulting.

Project operation result



View the example in ZetView

4.24.Tangent

The component **"Tangent"** is designed to implement an arithmetic operation on the output of the tangent of an angle.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Angle - the angle value.

➢ Output:
- Output - the value of the tangent of the angle is formed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- angleType (degrees) - set the angle unit (degrees or radians).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Angle - the angle value (any number).

➢**Custom common properties:**

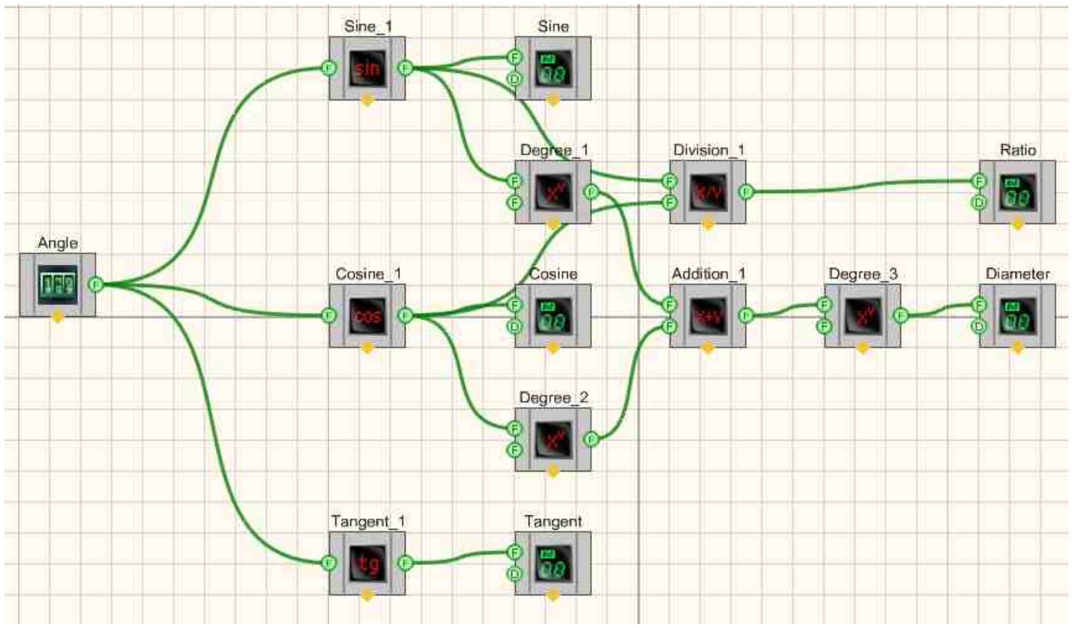
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

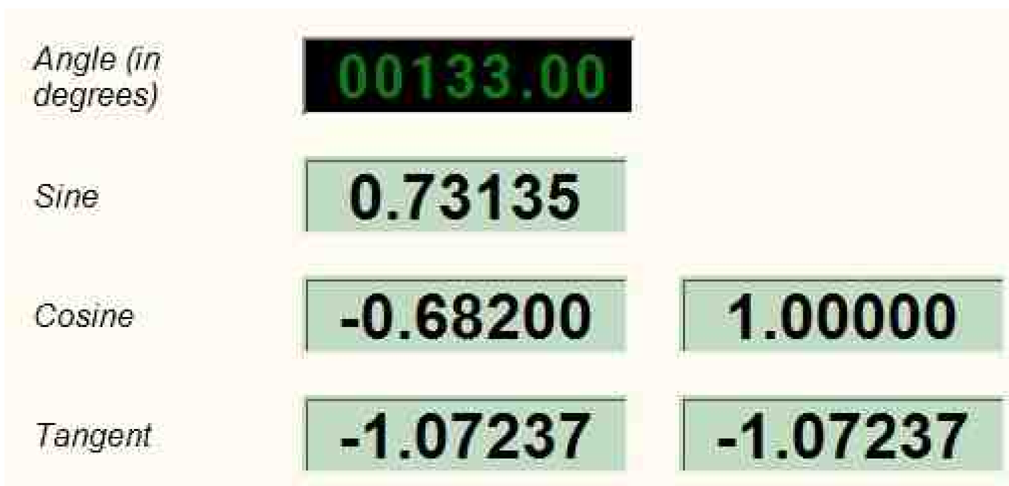
- BSTR AngleType - Set the angle unit: degrees; radians (string).

Example

Project in SCADA ZETView



Project operation result

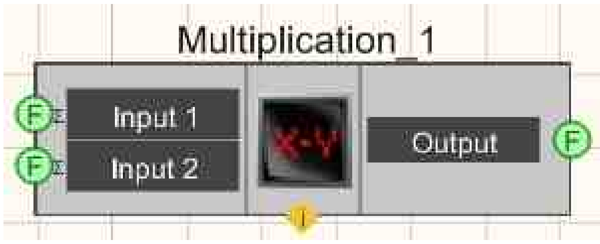


View the example in ZetView

4.25.Multiplication

The component **"Multiplication"** is designed to implement such an arithmetic operation as finding is the result of multiplication of two numbers.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

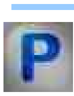
➢Input:
- Input 1 - The value of the first operand to calculate the result.
- Input 2 - The value of the second operand for calculating the result.

➢ Output:
- Output - the multiplication of all previous values and the current one is formed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- First Reaction (True) - Reaction to data change at input No.1:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- Second Reaction (True) - Reaction to data change at input No.2:
  True - when the input signal changes, the output value will change;
  False - when the input signal changes, the output value will not change.
- First (1) - The default value of the first operand.
- Second (1) - The default value of the second operand.



Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input 1 - the value of the first operand to calculate the result (any number).
- Input 2 - the value of the second operand to calculate the result (any number).

➢**Custom common properties:**
- Are available by the link below. 154

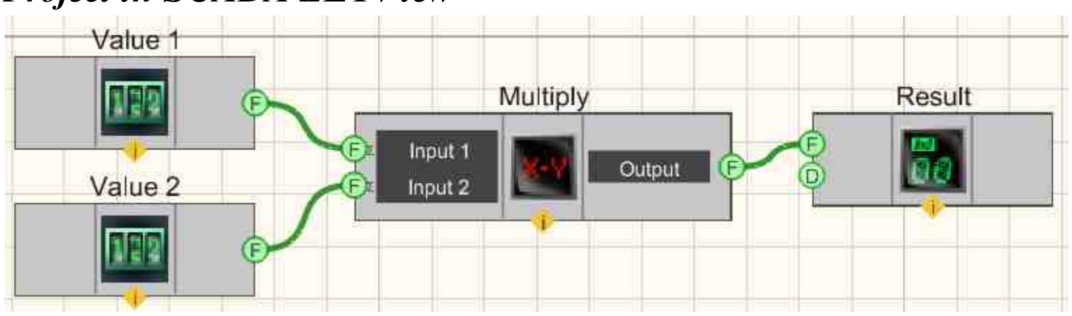
➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:
- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.
- FLOAT First - Set the value at input 1 for the calculation (any number).
- FLOAT Second - Set the value at input 2 for the calculation (any number).

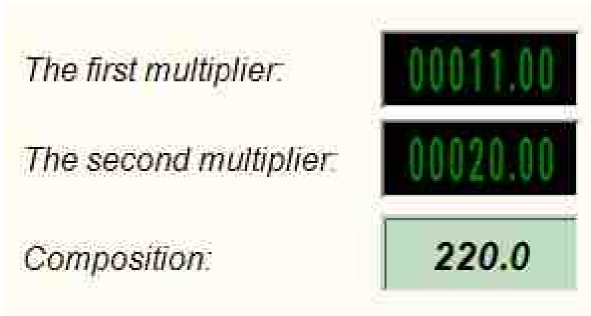
Example

Project in SCADA ZETView



This project shows the operation of the Multiplication 330 component performs the operation of finding the product of two numbers. Selectors 428 are used to enter numerical values at the input of the component. A Digital Indicator 682 is needed for graphical output of the resulting numerical value.

Project operation result

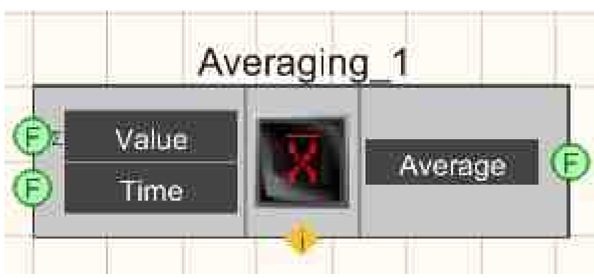


View the example in ZetView

4.26.Averaging

The component **"Averaging"** It is used to implement such a mathematical operation as finding the arithmetic mean or rms of all values coming to the input for a given period of time.is the result of multiplication of two numbers.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- The value is the averaged data.
- Time - averaging time (from 0 to 100 seconds).

➢ Output:
- Average - an average value is formed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- AvarageTime (1) - set the averaging time, with
- averagetype (Arithmetic mean value) - type of averaged value:
    1. Arithmetic mean value - the arithmetic mean value.
    2. RMS (root-mean-square) **-** the average square value.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- The value is the averaged data (any number).
- Time - averaging time (from 0 to 100 seconds).

➢**Custom common properties:**

- Are available by the link below. 154

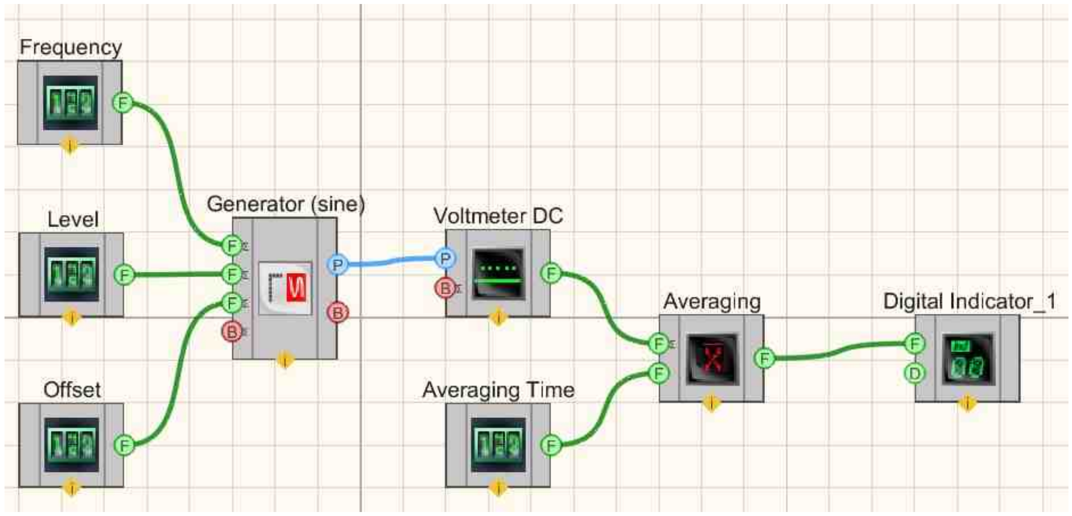
➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT AverageTime - Set the values coming in for averaging (any number).
- BSTR AverageType - Set the type of the averaged value (string):
    1. Arithmetic mean value - Arithmetic mean value.
    2. RMS (root-mean-square) - the average square value.

Example

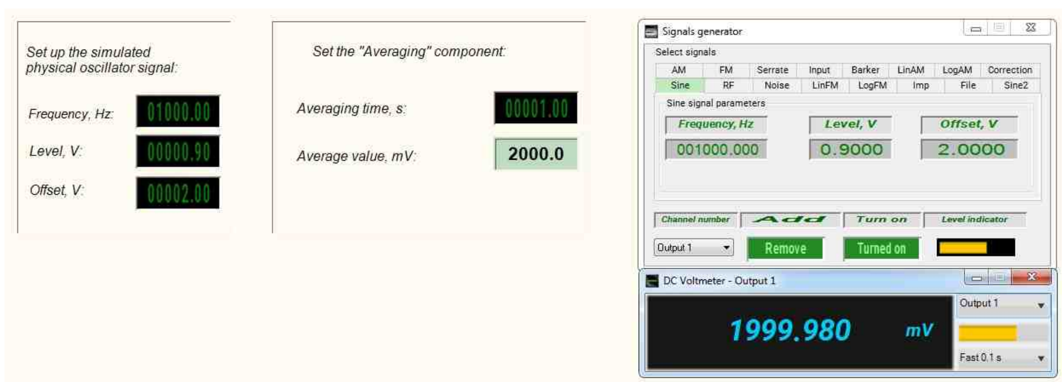
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



In this scheme of operation of the "Averaging 335" component. The "Generator (sine)" 530 component, receiving a signal from a physical DAC of a ZET device connected to a PC, generates a signal according to the specified parameters. The parameters are set using the corresponding Selectors 428 connected to the Generator (sine) 530. Next, DC Voltmeter 591 takes instantaneous voltage values and sends them to the "Averaging 335" component. Using a selector called "Averaging Time", the time is set during which the component will take values from the voltmeter for averaging. The result is displayed on the Digital Indicator 682.

Project operation result



When starting the project, the averaged value will take an unchanged value, if you change the value of the signal offset on the Generator (sine), the averaging value will change. By default, in this Example, the component calculates the arithmetic mean value.

View the example in ZetView

Глава 5.External devices

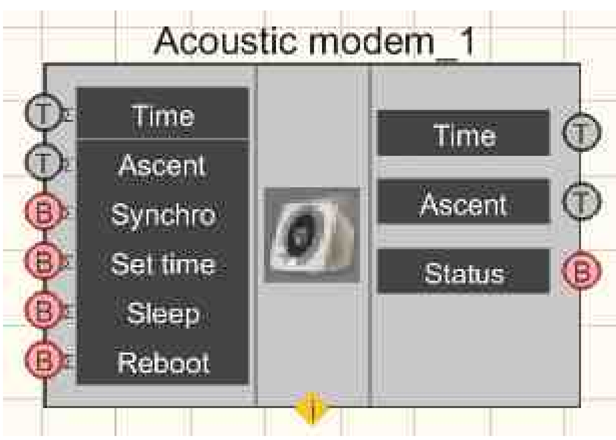
Components of "Measurements (External devices)" section are used for measurements of various parameters of signals received at the input modules of ADC-DAC channels, Narrow-band spectrum program, strain-gauge stations and virtual channels created by such programs as ZETFormula, Signal generator, Filtration by channels, etc:

Components of "Measurements (External devices)" are not displayed in operator module and can only be added to the project in developer mode.

5.1.Acoustic modem

The component "**Acoustic modem**" - is used for conversion of digital signals into sound signals and vice versa. The device allows to send and receive sound frequency signals via phone lines, which allows to establish connection between PC and terminal by means of a modem or a cell-phone,

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Time - AK time setting.
- Ascent - set the ascent time.
- Synchro - get data.

- Set time - set the time.
- Sleep - switch the modem to sleep mode.
- Reboot - restart the modem.

➤ Output:

- Time - current AK time.
- Ascent - current ascent time.
- Status - status of connection with the modem.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses**):

- Device (0) - the serial number of the device is set.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Time - An input numeric array to set the value of the AK time setting (any number).
- Ascent - An input numeric array to set the value of the ascent time setting (any number).

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- LONG Device - Set and reading the serial number of the device.

➤**Methods:**

- void EnableAU(BYTE enable) - Set Enable/disable the acoustic modem.

5.2.Scales BP05mc

The component "**Scales BP05mc**" used as scales.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

Doesn't have.

➢ Output:

- Weight - the output values of the scales.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

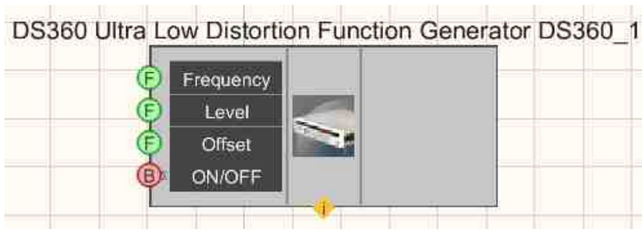
➢Custom common properties:

- Are available by the link below. 154

5.3.DS360 Ultra Low Distortion Function Generator

The component "DS360 Ultra Low Distortion Function Generator" – is used for reproduction of electromagnetic signals. DS360 allows to reproduce signal of a particular shape, which is often necessary in the course of research and settings of various electronic systems. The device allows to produce certain types of impulses with a preset parameters. It also enables easy testing of computing devices.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Frequency - generated signal frequency, Hz.
- Level - generated signal level, V.
- Offset - offset of the generated signal, V.
- ON/OFF - the status of the generator.

➢ Output:

- Doesn't have

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- PortNumber (1) - number of the serial (COM) port for working with the generator
- BaudRate (9600) - data exchange rate on the serial (COM) port with power supply, bps.
- signaltype (Sine) - the type of generated signal:
- 1. Sinus.
- 2. Meander.
- Frequency (1000) - generated signal frequency, Hz.
- Amplitude (1) - RMS level of the generated signal, V
- Offset (0) - zero offset of the generated signal, V
- OutputStatus (false) - generator operation status.
- outputmode (Unbalanced) - output mode.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Frequency - The value of the output frequency of the generated signal, Hz (number).
- Level. - The value of the output level of the generated signal, V (number).
- Offset - The value of the output offset of the generated signal, V (number).

➢**Custom common properties:**

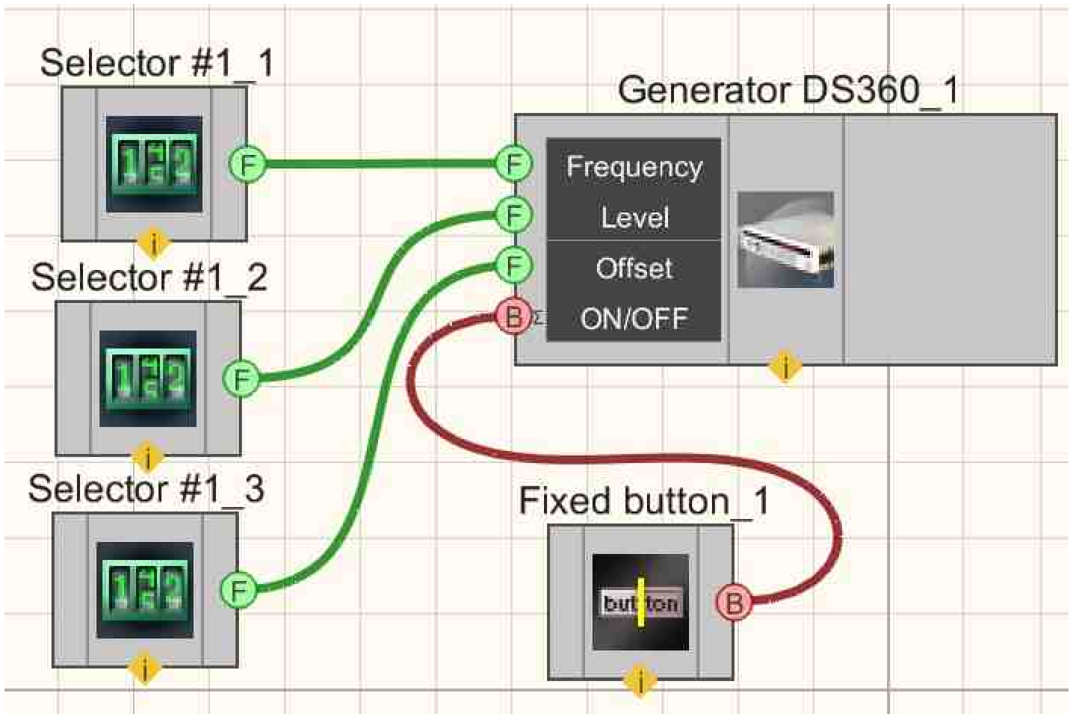
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

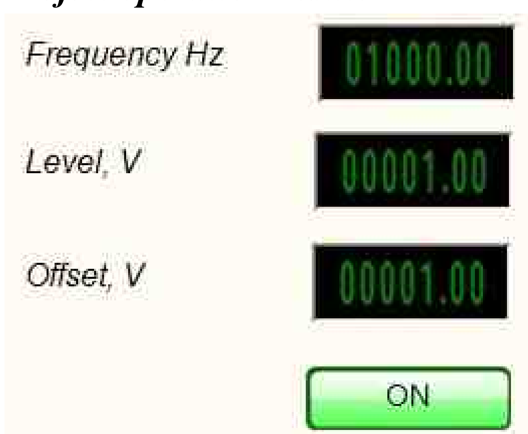
- LONG PortNumber - Set the serial (COM) port number for working with the generator (from 1 to 255).
- LONG BaudRate - Set the data exchange rate on the serial (COM) port with the generator, bps (number).
- BSTR SignalType - Set the type of generated signal: Sine; Meander (number).
- FLOAT Frequency - Set the frequency of the generated signal, Hz (from 0.01 to (sampling frequency of the DAC / 2)), Hz.
- FLOAT Amplitude - Set the RMS level of the generated signal (from 0 V to the maximum limit level of this model).
- FLOAT Offset - Set the offset of the constant component of the generated signal, V (from 0 to the maximum limit for this model), V.
- VARIANT_BOOL OutputStatus - Set the generator running status:
  true - the status of the generator is enabled;
  false - the status of the generator is disabled.
- BSTR OutputMode - Output mode setting: Unbalanced; Balanced (string).

Example

Project in SCADA ZETView



Project operation result

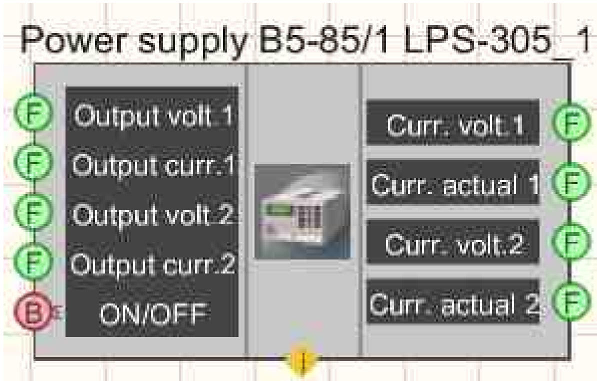


View the example in ZetView

5.4.Power supply LPS-305

The component **Power supply LPS-305**" – is used for control of programmable power supply source LPS-305;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Output volt. 1 - the value of the output voltage of the power source on the first channel, V.
- Output curr. 1 - the value of the output current of the power source on the first channel, A.
- Output volt. 2 - the value of the output voltage of the power supply on the second channel, V.
- Output curr. 2 - the value of the output current of the power supply on the second channel, A.
- ON/OFF - enable/disable the status of the power supply.

➢ Output:

- Curr. volt.1 - the current value of the output voltage of the power supply on the first channel, V.
- Curr. actual 1 - the current value of the output current of the power supply on the first channel, A.
- Curr. volt.2 - the current value of the output voltage of the power supply on the second channel, V.
- Curr. actual 2 - the current value of the current of the second channel on the second channel, A.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- PortNumber (1) - number of the serial (COM) port for working with the power supply
- Voltage1 (0) - set the value of the output voltage of the power supply on the first channel, V;
- Current1 (0) - set the value of the output current of the power source on the first channel, A;

- Voltage2 (0) - set the value of the output voltage of the power supply on the second channel, V;
- Current2 (0) - set the value of the output current of the power source on the second channel, A;
- digitalvoltage (off) - the value of the output voltage of the power supply on the third channel;
- OutputStatus (false) - the status of the power supply:
  1. False - Off;
  2. True - On.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢ **Input:**

- Output volt. 1. - The value of the output voltage of the power supply on the first channel, V (number).
- Output curr. 1 - The value of the output current of the power source on the first channel, A (number).
- Output volt. 2 - The value of the output voltage of the power supply on the second channel, V (number).
- Output curr. 2 - The value of the output current of the power supply on the second channel, A (number).

➢ **Custom common properties:**

- Are available by the link below. 154

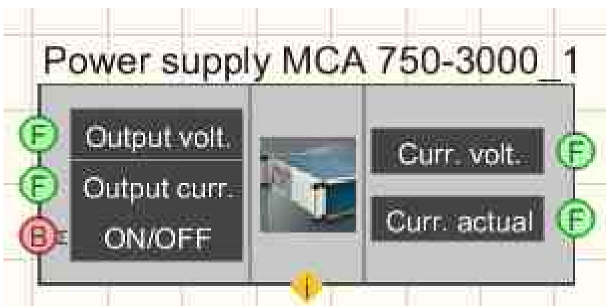
- LONG PortNumber - Set the serial (COM) port number for power supply operation (from 1 to 255).
- FLOAT Voltage1 - Set the value of the output voltage of the power supply on the first channel (from 0 V to the maximum limit level of this model).
- FLOAT Current1 - Set the value of the output current of the power supply on the first channel (from 0 A to the maximum limit level of this model).
- FLOAT Voltage2 - Set the value of the output voltage of the power supply on the second channel (from 0 V to the maximum limit level of this model).
- FLOAT Current2 - Set the value of the output current of the power supply on the second channel (from 0 A to the maximum limit level of this model).
- BSTR DigitalVoltage - Set the value of the output voltage of the power supply on the third channel: 3.3 V; 5 V (string).

- FLOAT OVPValue2 - Set the value of the protective voltage level on the second channel (from 0 V to the maximum limit level of this model).
VARIANT_BOOL OutputStatus - Set the status of the power supply:
- true - The status of the power supply is on;
false - The power supply operation status is off.

5.5.Power supply MCA 750-3000

The component "**Power supply MCA 750-3000**" – is used for control of programmable power supply source MCA 750-3000;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Output volt. - the value of the output voltage of the power source on the channel, V.
- Output curr. - the value of the output current of the power source on the channel, A.
- ON/OFF - enable/disable the status of the power supply.

➢ Output:
- Curr. volt. - the current value of the output voltage of the power supply on the channel, V.
- Curr. actual - the current value of the output current of the power supply on the channel, A.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- PortNumber (1) - number of the serial (COM) port for working with the power supply
- Voltage (0) - set the value of the output voltage of the power supply on the channel, V;

- Current (0) - set the value of the output current of the power source on the channel, A;
- OutputStatus (false) - the status of the power supply:
    1. False - Off;
    2. True - On.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Output volt. - The value of the output voltage of the power supply on the channel, V (number).
- Output curr. - The value of the output current of the power source on the channel, A (number).

➢**Custom common properties:**

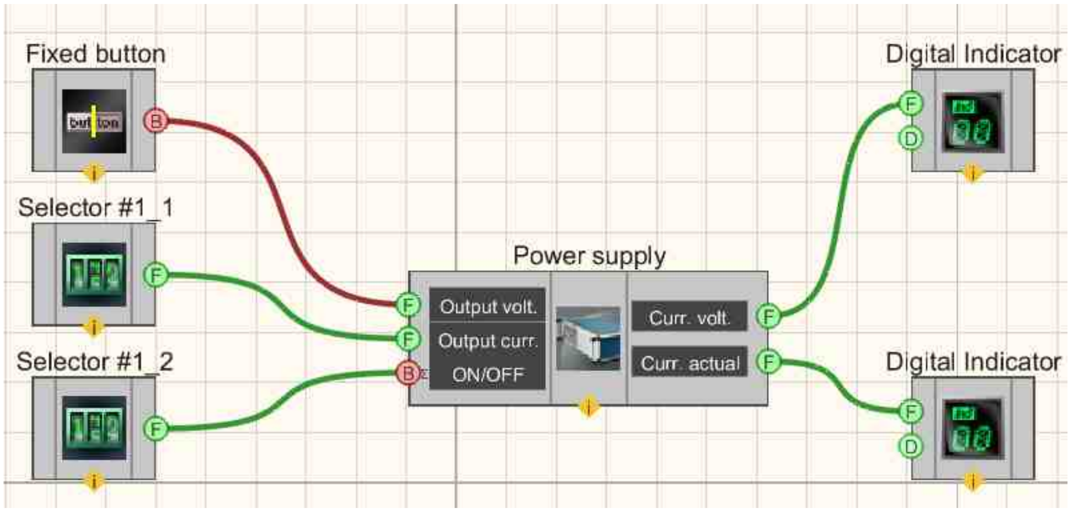
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG PortNumber - set the number of the serial (COM) port for working with the power supply (from 1 to 255).
- FLOAT Voltage - set the value of the output voltage of the power supply on the channel, V (from 0 V to the maximum limit level of this model).
- FLOAT Current - set the value of the output current of the power source on the channel, A (from 0 V to the maximum limit level of this model).
- VARIANT_BOOL Activate - the status of the power supply:
    1. False - Off;
    2. True - On.

Example

Project in SCADA ZETView



Project operation result

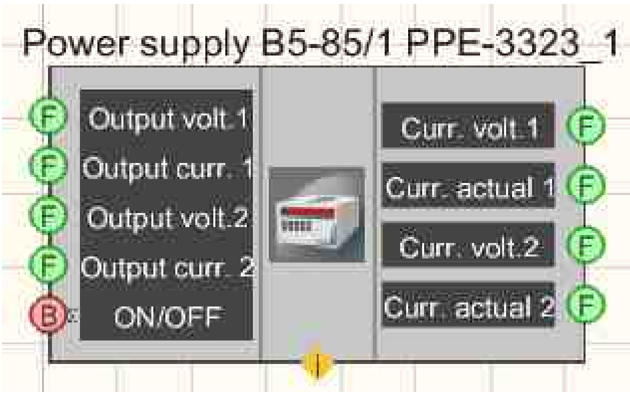
View the example in ZetView

5.6.Power supply PPE-3323

The component "**Power supply PPE-3323**" is used for control of programmable power supply source PPE-3323;

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- Output volt. 1 - the value of the output voltage of the power source on the first channel, V.
- Output curr. 1 - the value of the output current of the power source on the first channel, A.
- Output volt. 2 - the value of the output voltage of the power supply on the second channel, V.
- Output curr. 2 - the value of the output current of the power supply on the second channel, A.
- ON/OFF - enable/disable the status of the power supply.

➢ Output:
- Curr. volt.1 - the current value of the output voltage of the power supply on the first channel, V.
- Curr. actual 1 - the current value of the output current of the power supply on the first channel, A.
- Curr. volt.2 - the current value of the output voltage of the power supply on the second channel, V.
- Curr. actual 2 - the current value of the current of the second channel on the second channel, A.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- PortNumber (1) - serial (COM) port number for working with the power supply.
- Voltage1 (0) is the value of the output voltage of the power supply on the first channel, V.
- Current1 (0) is the value of the output current of the power supply on the first channel, A.
- Voltage2 (0) is the value of the output voltage of the power supply on the second channel, V.
- Current2 (0) is the value of the output current of the power supply on the second channel, A.

- OVPValue1 (34) - the value of the protective voltage level on the first channel, V.
- OVPValue2 (34) - the value of the protective voltage level on the second channel, V.
- OCPStatus (false) - overload protection of outputs.
- digitalvoltage (3.3 V) - the value of the output voltage of the power supply on the third channel.
- OutputStatus (false) - the status of operation of the power supply.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Output volt. 1. - The value of the output voltage of the power supply on the first channel, V (number).
- Output curr. 1 - The value of the output current of the power source on the first channel, A (number).
- Output volt. 2 - The value of the output voltage of the power supply on the second channel, V (number).
- Output curr. 2 - The value of the output current of the power supply on the second channel, A (number).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG PortNumber - Set the serial (COM) port number for working with the power supply (from 1 to 255).
- FLOAT Voltage1 - Set the value of the output voltage of the power supply on the first channel (from 0 V to the maximum limit level of this model).
- FLOAT Current1 - Set the value of the output current of the power supply on the first channel (from 0 A to the maximum limit level of this model).
- FLOAT Voltage2 - Set the output voltage of the power supply via the second channel (from 0 V to the maximum limit level of this model).
- FLOAT Current2 - Set the value of the output current of the power supply on the second channel (from 0 A to the maximum limit level of this model).
- FLOAT OVPValue1 - Set the value of the protective voltage level on the first channel (from 0 V to the maximum permissible level of this model).

- FLOAT OVPValue2 - Set the value of the protective voltage level on the second channel (from 0 V to the maximum permissible level of this model).
- VARIANT_BOOL OCPStatus - Set overload protection of outputs:
  - true - the operating status overload protection of outputs is ON;
  - false - the operating status overload protection of outputs is OFF.
- BSTR DigitalVoltage - Set the value of the output voltage of the power supply on the third channel: 3.3 V; 5 V (string).
- VARIANT_BOOL OutputStatus - Set the operating status of the power supply:
  - true - the operating status of the power supply is ON;
  - false - the operating status of the power supply is OFF.

5.7.Power supply PSH-3610

The component "**Power supply PSH-3610** " is used for control of programmable power supply source PSH-3610;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Output volt. - the value of the output voltage of the power source on the channel, V.
- Output curr. - the value of the output current of the power source on the channel, A.
- ON/OFF - enable/disable the status of the power supply.

➢ Output:

- Curr. volt. - the current value of the output voltage of the power supply on the channel, V.
- Curr. actual - the current value of the output current of the power supply on the channel, A.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- PortNumber (1) - serial (COM) port number for working with the power supply.
- BaudRate (9600) - data exchange rate on the serial (COM) port with power supply, bps.
- Voltage (0) is the value of the output voltage of the power supply on the channel, V.
- Current (0) is the value of the output current of the power supply on the channel, A.
- OVPValue (38) - the value of the protective voltage level on the channel, V.
- OCPStatus (false) - overload protection of outputs.
- OutputStatus (false) - the status of operation of the power supply.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Output volt. - The value of the output voltage of the power supply on the channel, V (number).
- Output curr. - The value of the output current of the power source on the channel, A (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG PortNumber - Set the serial (COM) port number for working with the generator (from 1 to 255).
- LONG BaudRate - Set the data exchange rate on the serial (COM) port with the generator, bps (number).
- FLOAT Voltage - Set the value of the output voltage of the power supply (from 0 V to the maximum limit level of this model).
- FLOAT Current - Set the value of the output current of the power supply (from 0 A to the maximum limit level of this model).
- FLOAT OVPValue - Set the value of the protective voltage level (from 0 V to the maximum limit level of this model).

VARIANT_BOOL OCPStatus - Set overload protection of outputs:

- true - the operating status overload protection of outputs is ON;
  false - the operating status overload protection of outputs is OFF.

VARIANT_BOOL OutputStatus - Set the operating status of the power supply:
- true - the operating status of the power supply is ON;
  false - the operating status of the power supply is OFF.

5.8.Power supply PSM-2010

The component "**Power supply PSM-2010**" is used for control of programmable power supply source PSH-2010;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- U - channel voltage limit, V
- I - channel current limit, A
- ON/OFF - the status of the power supply.

➢ Output:
- Voltage - the current value of the output voltage of the power supply on the channel, V.
- Current - the current value of the output current of the power supply on the channel, A.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
Survey(true) - periodic monitoring of the device status is set.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- U - the value of the output voltage of the power source, V (number).
- I - the value of the output current of the power source, A (number).

➢**Custom common properties:**
- Are available by the link below. 154

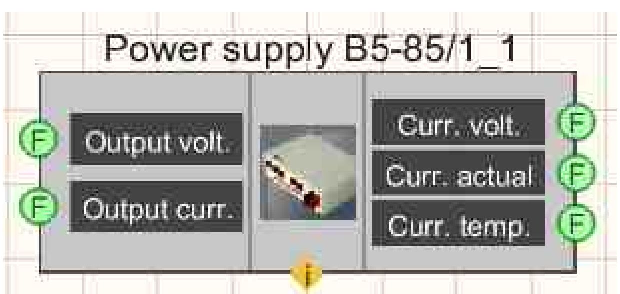
➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL Survey - Set the device status to be periodically monitored:
  true - Periodic device status monitoring is ON;
  false - Periodic device status monitoring is OFF.

5.9.Power supply B5-85/1

The component "**Power supply B5-85/1**" - is used for control of programmable power supply source B5_85/1;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Output volt. - the value of the output voltage of the power source on the channel, V.
- Output curr. - the value of the output current of the power source on the channel, A.

➢ Output:

- Curr. volt. - the current value of the output voltage of the power supply on the channel, V.
- Curr. actual - the current value of the output current of the power supply on the channel, A.
- Curr. temp. - the current value of the temperature inside the power supply.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- PortNumber (1) - serial (COM) port number for working with the power supply.
- Voltage (0) is the value of the output voltage of the power supply on the first channel, V.
- Current (0) is the value of the output current of the power supply on the first channel, A.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Output volt. - The value of the output voltage of the power supply on the channel, V (number).
- Output curr. - The value of the output current of the power source on the channel, A (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG PortNumber - Set the serial (COM) port number for working with the generator (from 1 to 255).
- FLOAT Voltage - Set the value of the output voltage of the power supply on the first channel (from 0 V to the maximum limit level of this model).
- FLOAT Current - Set the value of the output current of the power supply on the first channel (from 0 A to the maximum limit level of this model).

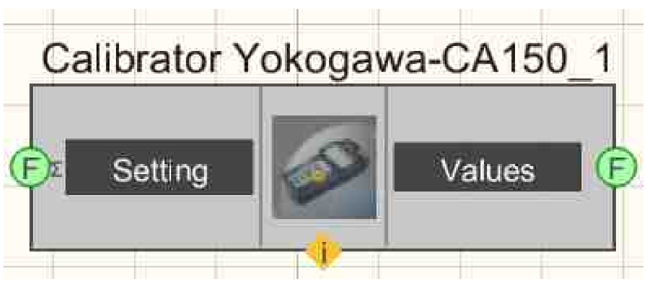
5.10.Calibrator Yokogawa-CA150

The component "**Calibrator Yokogawa-CA150**" is used for sensors power supply circuits testing, field instruments, digital multimeters, records and various measuring instruments calibration. Can be used both for field instruments maintenance and settings of laboratory instruments. Multimeter calibrator CA150 has a light weight (1kg) and compact units.

Features of Portable Pressure Calibrator CA150Yokogawa:

- Easy to learn;
- Freedom of use: vertical mounting, desktop or strap-on;
- Simultaneous signal generation and measurement;
- Low measurement and generation error: 0.02% for DC and voltage;
- The ability to measure resistance, thermocouple signals and thermal resistances (10 types), frequency.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Setting - the value of the calibrator setting.

➢ Output:
- Value - the measured value of the calibrator.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- PortNumber (1) - serial (COM) port number for working with the power supply
- SignalFunction (Constant voltage) - set the signal generation.

- MeasureFunction (Constant voltage) - the measurement value.
- SignalOutput (false) - The status of the generation mode.
- MeasureOutput (false) - Measurement status.
- SignalRangeDCV(100 mV) - DC voltage generation range: 100 mV; 1 V; 10 V; 30 V.
- SignalRangeDCA(20 mA) - DC generation range: 20 mA; 4 ч 20 mA.
- SignalRangeR (500 Ω) - Resistance generation range: 500 Ω; 5 kΩ; 50 kΩ.
- SignalRangeTC (K) - Type of thermocouple generation: K; E; J; T; R; B; S; N; L; U.
- SignalRangeRTD (PT100) - Type of thermal resistance generation: PT100; JPT100.
- SignalRangeImpulse (100 Hz) - Frequency signal generation range:100 Hz;1000 Hz;10 kHz;50 kHz.
- MeasureRangeDCV (500 mV) - DC voltage measurement range: 500 mV; 5 V; 35 V.
- MeasureRangeDCA (20 mA) - DC measurement range: 20 mA; 100 mA.
- MeasureRangeR (500 ohms) - Resistance measurement range: 500 Ω; 5 kΩ; 50 kΩ.
- MeasureRangeTC (K) - Type of thermocouple measurement: K; E; J; T; R; B; S; N; L; U.
- MeasureRangeRTD (PT100) - Type of thermal resistance measurement: PT100; JPT100.
- MeasureRangeImpulse (100 Hz) - Frequency signal measurement range: 100 Hz; 1000 Hz; 10 kHz; CPM.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Setting - The value of the calibrator setting (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG PortNumber - Set the serial (COM) port number for working with the generator (from 1 to 255).
- BSTR SignalFunction - Set the signal generation function: Constant voltage; Direct current (D.C); Resistance; Thermocouple; Thermal resistance; Frequency signal (string).
- BSTR MeasureFunction - Set the measurement function: Constant voltage; Direct current (D.C); Resistance; Thermocouple; Thermal resistance; Frequency signal (string).
- VARIANT_BOOL SignalOutput - Set the status of the generation mode:

true - the generation mode status is ON;
false - the generation mode status is OFF;

VARIANT_BOOL MeasureOutput - Set the measurement status:

- true - the measurement status is ON;
false - the measurement status is ON.
- BSTR SignalRangeDCV - Set the DC voltage generation range: 100 mV; 1 V; 10 V; 30 V (string).
- BSTR SignalRangeDCA - Set the DC generation range:20 mA;4 ч 20 mA (string).
- BSTR SignalRangeR - Set the resistance generation range: 500 Ω; 5 kΩ; 50 kΩ (string).
- BSTR SignalRangeTC - Set the type of thermocouple (TC) generation: K; E; J; T; R; B; S; N; L; U (string).
- BSTR SignalRangeRTD - Set the thermal resistive thermometer (TRC) generation type: PT100; JPT100 (string).
- BSTR SignalRangeImpulse - Set the frequency signal generation range: 100 Hz; 1000 Hz; 10 kHz; 50 kHz (string).
- BSTR MeasureRangeDCV - Setting the DC voltage measurement range: 500 mV; 5 V; 35 V (string).
- BSTR MeasureRangeDCA - Setting the DC current measurement range: 20 mA; 100 mA (string).
- BSTR MeasureRangeR - Set the resistance measurement range: 500 Ω; 5 kΩ; 50 kΩ (string).
- BSTR MeasureRangeTC - Set the thermocouple (TC) measurement type: K; E; J; T; R; B; S; N; L; U (string).
- BSTR MeasureRangeRTD - Set the thermal resistive thermometer (TRC) measurement type: PT100; JPT100 (string).
- BSTR MeasureRangeImpulse - Set the measurement range of the frequency signal: 100 Hz; 1000 Hz; 10 kHz; CPM (string).

5.11.Multimeter Agilent 34401a

The component "**Multimeter Agilent 34401a**" is used for constant and alternating current parameters measurements, measurements of alternating current frequency, circuit section impedance, circuits and p-n transitions control;

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- Doesn't have.

➢ Output:

- Values - output values of the multimeter.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- PortNumber (1) - number of the serial (COM) port for working with the generator.
- BaudRate (9600) - data exchange rate on the serial (COM) port with the generator, bps.
- measuretype (constant voltage) - type of measured value:
    1. constant voltage;
    2. ratio of DC voltage;
    3. alternating voltage;
    4. constant current;
    5. alternating current;
    6. resistance (2 wire);
    7. resistance (4 wire);
    8. frequency;
    9. period;
    10. incessancy of electrical circuits;
    11. diode test.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below.[154]

➢**Custom private properties (the default value is shown in parentheses):**

- LONG PortNumber - Set the serial (COM) port number for power supply operation (from 1 to 255).
- LONG BaudRate - Set the data exchange rate on the serial (COM) port with the generator, bps (number).
- BSTR MeasureType - Set the type of measured value: constant voltage (DC voltage); ratio of DC voltage; alternating voltage (AC voltage); constant current (DC); alternating current (AC) ;Resistance (2 wire); Resistance (4 wire); frequency; period; incessancy of electrical circuits; diode test (string).

5.12.Nanovoltmeter Keithley 2182a

The component "**Nanovoltmeter Keithley 2182a**" is used for super-low voltage measurements, has a low level of intrinsic noise and enables high rate of measurements performance.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Doesn't have.

➢ Output:

- Values - nanovoltmeter values.

Custom common properties:

➢Total properties (environment):

- Are available by the link below.[154]

➢**Custom private properties (the default value is shown in parentheses):**

- PortNumber (1) - number of the serial (COM) port for working with the generator
- BaudRate (9600) - speed of data exchange on the serial (COM) port with the generator, bps
- Channel (DCV2) - measured channel.
- range (10 mV) - measurement range, mV.
- rate (Fast) - data update rate in the device, 3 rate options are possible: Fast, Medium, Slow.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

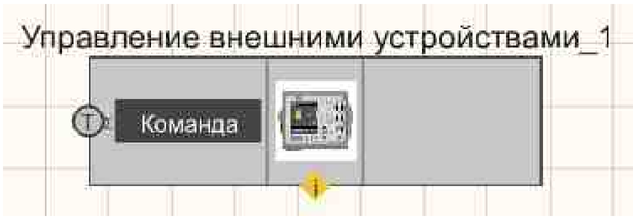
- LONG PortNumber - Set the serial (COM) port number for working with the generator (from 1 to 255).
- LONG BaudRate - Set the data exchange rate on the serial (COM) port with the generator, bps (number).
- BSTR Channel - Set the measurement channel setting: DCV1; DCV2; V1/V2 (string).
- BSTR Range - Set the measurement range setting: 10 mV; 100 mV; 1 V; 10 V; 100 V (string).
- BSTR Rate - Set the level of data update rate: Fast; medium; Slow (string).

5.13. Control of external devices

The component "**Control of external devices**" is used for designed to control external devices.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- Doesn't have.

➢ Output:

- Command - designed to send a command to control the device.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Serial () - the device serial number.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Command - designed to send a command to control the device (text).

Setting:

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

•BSTR Serial - Set the instrument serial number (string).

Глава 6.Time

6.1.Time delay

The component "**Time delay**" is used for designed to generate a delay in the transmission of input data by a given amount.

Often, this component is used as an auxiliary element in schemes where several data streams flow to one component. Since the data goes through chains of components of different lengths and goes through different stages of processing that require some time, a desynchronization may occur at the input of the receiving component. Data streams that should arrive at the same time may be delayed. This can cause conflicts in the work of the project. A visual Example can be found in the description of the Histogram 835 component.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - data is received at the input, which is delayed for a specified time.

➢ Output:
- Output - data is output after a delay.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Delay (1000) - the value of the delay time, ms.
- type (By time) - set the delay type:

  1. By time - transmitting the value that came to the input after the specified time.
  2. For one clock cycle - the value that came to the input will be transmitted from the output at the time when the next value arrives at the input.
- SendAll (true) - send all data (if the delay time has not expired) or only the latest.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - The channel in which you want to organize the delay (any type).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

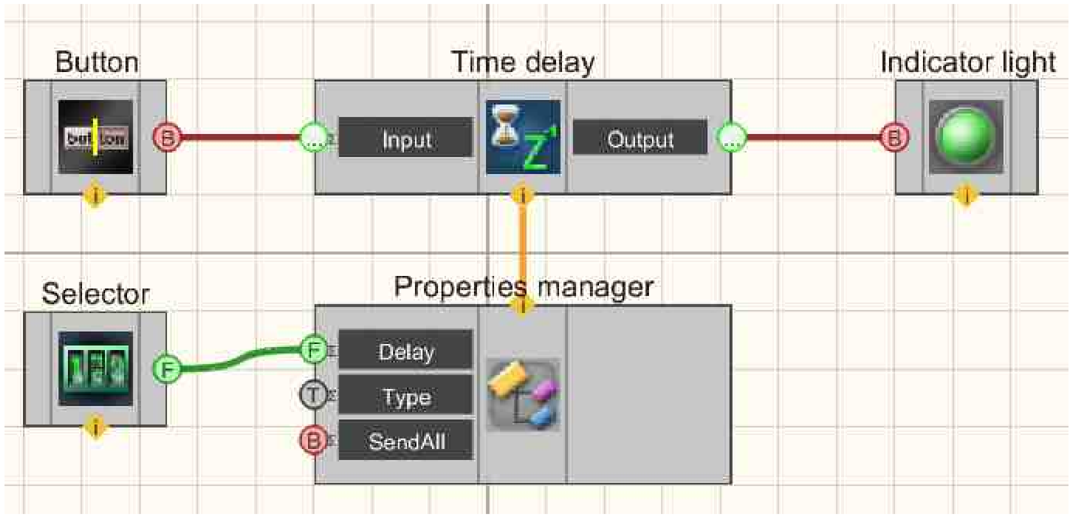
- FLOAT Delay - Set the delay time value, ms (from 0 to plus infinity).
- BSTR Type - Set the delay type: By time; By one clock cycle (string).
- VARIANT_BOOL SendAll - Set to transmit all data (if the delay time has not expired) or only the latest:
  true - transmit all data enabled;
  false - transmit all data is disabled.

➢**Methods:**

- void StopWorking() - Set the method for starting/stopping worker threads.

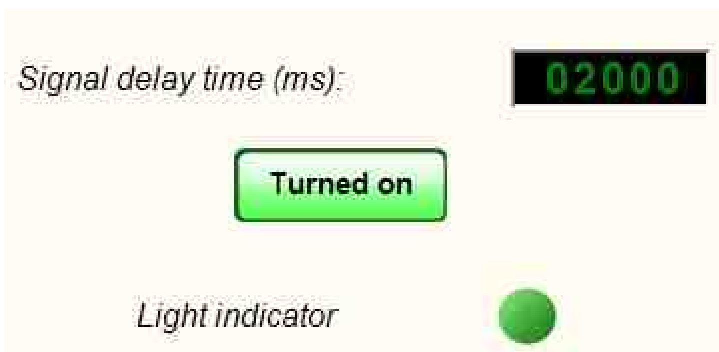
Example

Project in SCADA ZETView



In the presented scheme, the Time Delay 363 component generates a delay in the signal coming from with Fixed button 694. Through the Properties manager 999, we can set the "Delay" property (delay time) of the Time Delay 363 component using the Selector 428 connected to it. As a result, the Digital Indicator 682 goes out and lights up with a delay.

Project operation result



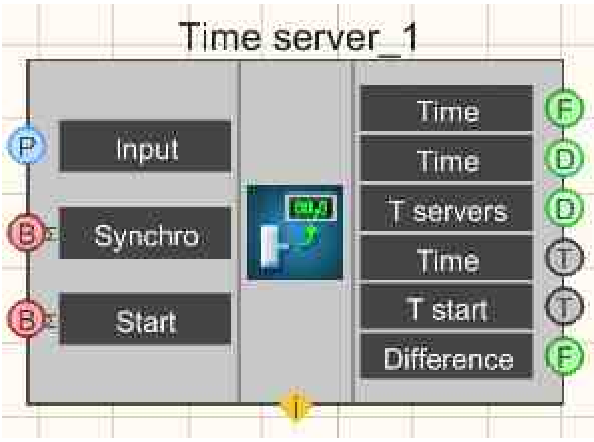
Set the delay time to 2000 ms, press the button "Turned off". The indicator light will light up with a delay of 2 seconds.

View the example in ZetView

6.2.Time server

The component "**Time server**" is used you to get the channel time.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - the name of the measuring channel, the time of which you want to determine;
- Synchro - synchronization, when a high logical level is applied, the channel time is interrogated;
- Start - synchronization of the server start time output.

➢ Output:
- Time - time on the measuring channel, s;
- Time - channel time in double format, s;
- T Server - the current server time in DATE format;
- Time - the time of the channel in text format (day.month.year clock.minutes.seconds);
- T Start - the start time of the server in text format (day.month.year clock.minutes.seconds).
- Difference - is the time since the start of the ADC, s

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Auto (false) - enable or disable automatic time monitoring.
- Frequency (1) - monitoring frequency, Hz.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Channel - Measuring channel whose time is to be determined (any channel).

➢**Custom common properties:**

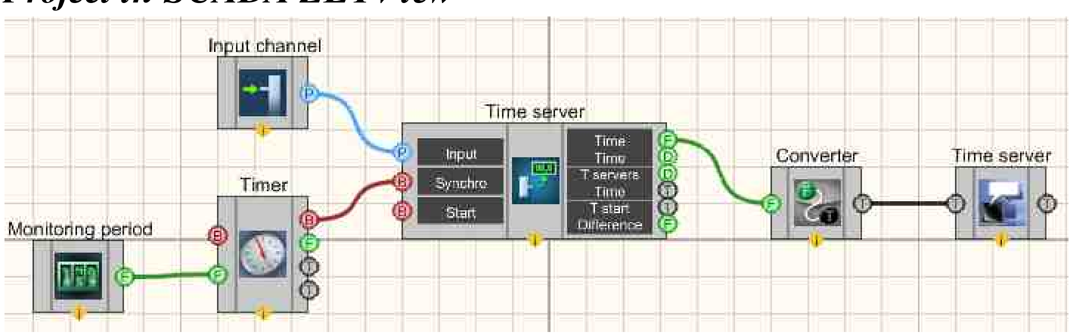
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL Auto - Set automatic time monitoring:
  true - automatic time monitoring is enabled;
  false - automatic time monitoring is disabled.
- LONG Frequency - Set the auto-query frequency, Hz (from 0.01 to half the sampling frequency of the measuring channel).

Example

Project in SCADA ZETView



This project shows the operation of the Time server 366 component is used to determine the channel time. The Timer 370 is needed to generate the clock impulses that will be used to monitoring the channel. The Selector 428 is used for the problem of the periodicity of the creation of clock impulses. A Converter 716 is needed to convert a numeric type to a string type. The Label 1087 is used for graphical representation of information.

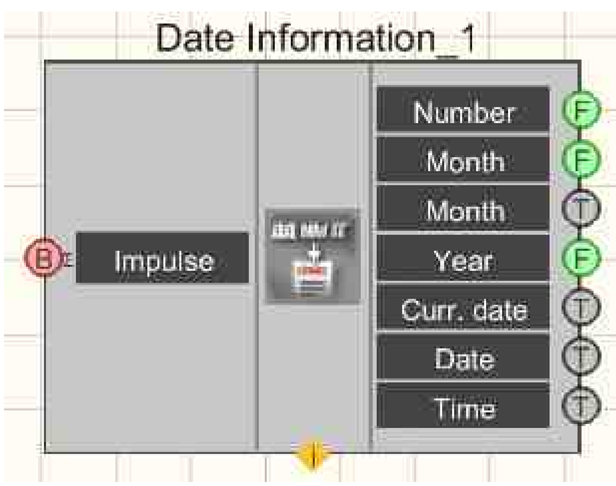
Project operation result

View the example in ZetView

6.3.Date Information

The component "**Date Information**" is used you to get the current date when the input of the control action component.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - The Impulse to send the date.

➢ Output:

- Number - the current calendar day in numerical form.
- Month - the current month in numerical form.
- Month - the current month as a text value.
- Year - the current year in numerical form.
- Curr. date - the current date as a text string (DD.MM.YY).
- Date - the current date with the number of additional months and years (DD.MM.YY) constantly added to it.
- Time - current time.

Custom common properties:

➢ Total properties (environment):

- Are available by the link below. 154

➢ Custom private properties (the default value is shown in parentheses):

- Declination (false) - enable/disable declination of months.
- AddMonth (0) - the number of months to add to the current date.
- AddYear (0) - the number of years to add to the current date.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢ Custom common properties:

- Are available by the link below. 154

➢ Custom private properties (the default value is shown in parentheses):

VARIANT_BOOL Declination - Set the declination of the months:

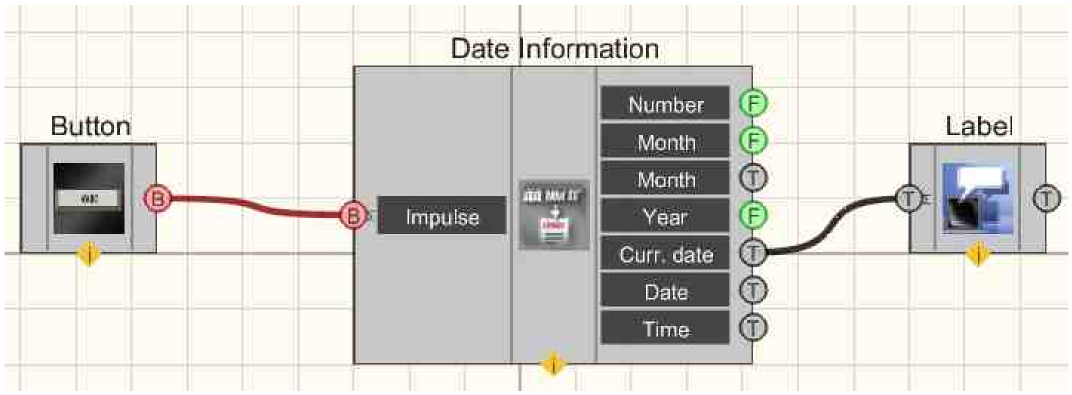
- true - month declination is enabled;
  false - declination of months is disabled.

•LONG AddMonth - Set additional months to the current date to calculate the next date (number).

•LONG AddYear - Set years to add to the current date to calculate the next date (number).

Example

Project in SCADA ZETView



This project shows the operation of the Date Information 368 component passes the current date to the Label 1087 input channel. The Button 694 is used to send Date Information 368 of the control impulse to the input, in order to transfer the date to the output channels of the component.

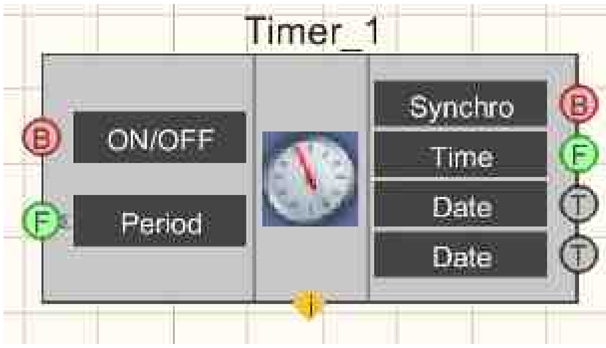
Project operation result

View the example in ZetView

6.4.Timer

The component "**Timer**" is used you to gives out impulses with a certain frequency.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- ON/OFF - enable/disable the timer.
- Period - set the period of the timer, s.

➢ Output:
- Synchro - a sync impulse when the timer is triggered.
- Time - current timer time, s.
- Date - current date and time.
- Date - current date and time on the server.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- status (false) - the status of the timer.
- timeInterval (1.00000) - time interval for the timer, s.
- dataformat (DD.MM.YY - HH:MM:CC) - date type.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Period - The value of the time interval for the period of the timer, s (any number).

➢Custom common properties:

- Are available by the link below. 154

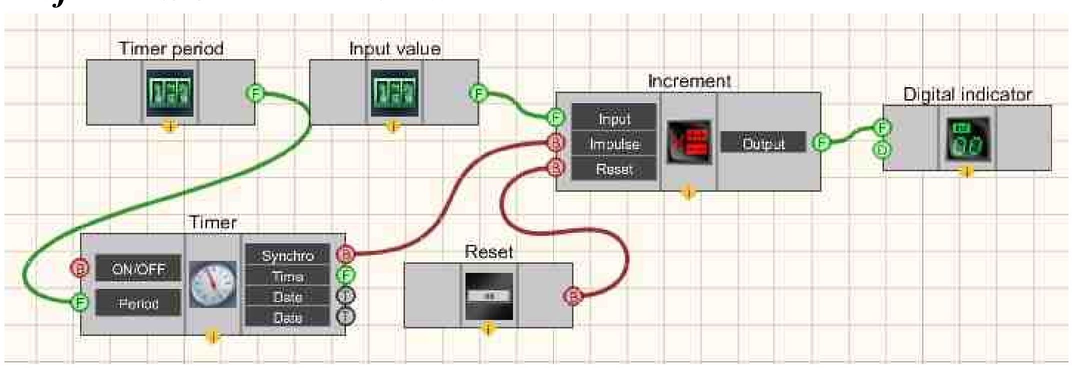
➢Custom private properties (the default value is shown in parentheses):

VARIANT_BOOL status - Set the timer operation status:

- true - the timer is enabled;
  false - the timer is disabled.
- FLOAT timeInterval - Set the timer trigger time interval, s (number).
- BSTR DateFormat - Set the date format: DD.MM.YYYY - HH:MM:SS; YYYY-MM-DD HH:MM:SS; YYYY_MM_DD_CH_MM_SS (string).

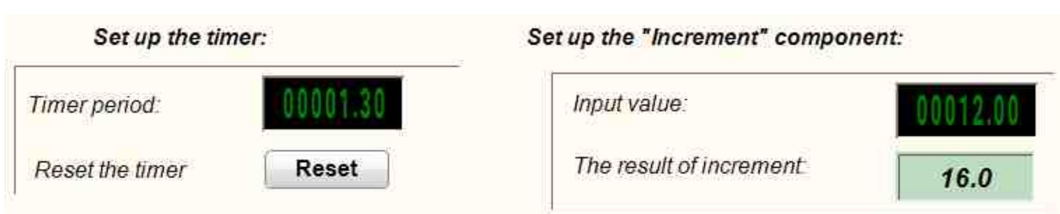
Example

Project in SCADA ZETView



This project shows the operation of the Increment 370 constantly adds one to the numerical value of the input channel. Timer 370 it is used to generate input impulses Increment 370, upon which an increment of the input value is made. Selectors 428 they are used to set the input value and the timer period. A Digital Indicator 682 is used for graphical output of the resulting.

Project operation result



View the example in ZetView

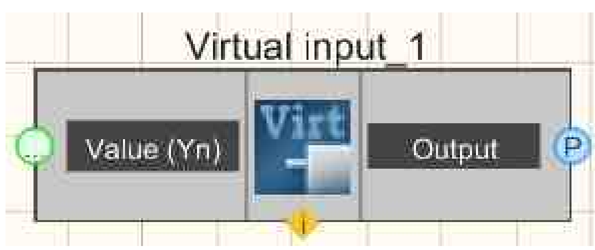
Глава 7.Input (ADC)

The components of "Input" section are used for operations with ZET-server channels in SCADA-system ZETView. The list of channels is formed based on real measurements channels (active ADC channels of the devices connected to PC) and virtual measurements channels. Virtual channels are created by generators (signals, sent from DAC output of the devices connected) and by programs for signals processing (signals filtration and processing in accordance with a particular algorithm). In the course of "Play recorded signals" program operation the list of channels consists of previously recorded and virtual channels, recorded in the course of data processing:

7.1.Virtual input

The component "**Virtual input**" component. Special channel used for emulation in ZETView environment. It is an analog of Input signal module (the only difference is that Input signal module receives data from real device, while **Virtual input** displays a signal created by PC).

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Value (Yn) - the input value written to the created virtual channel.

➢ Output:
- Output - the output signal.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Name (Channel) - alphanumeric designation of the channel.
- Conversion (mV) - unit of measure for the channel.
- Frequency (50000) - Sampling frequency by channel, Hz.
- MaxLevel (1000) - maximum limit level (in channel units).
- MinLevel (0) - minimum limit level (in channel units).
- Referense (0.001) - reference for calculating the level in dB (in channel units, zero level for the reference of the scale in dB).
- TimeMasterChannel (0) - the channel by which the time of the virtual channel will be synchronized.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- Value (Yn) - the value recorded to the create virtual channel (any type).

➢**Custom common properties:**
- Are available by the link below. 154

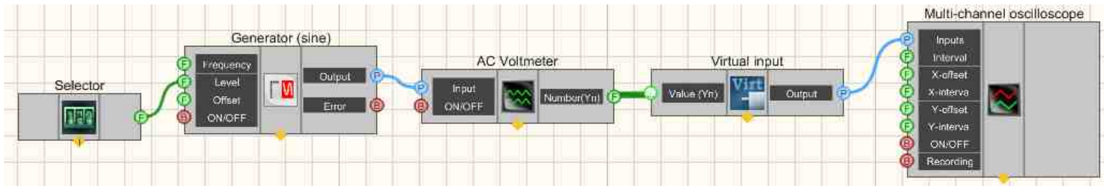
➢**Custom private properties (the default value is shown in parentheses):**

- BSTR Name - Set the channel name (string).
- BSTR Conversion - Set the unit of measurement by channel (string).
- FLOAT Frequency - Set the sampling frequency by channel (from 0.01 to the sampling frequency of the channel).
- FLOAT MaxLevel - Set the maximum limit channel level, units (any number).
- FLOAT MinLevel - Set the minimum limit level on the channel, unit of change (any number).
- FLOAT Referense - Set the reference for calculating the level in dB, units of change (any value).
- LONG TimeMasterChannel - Set the channel by which the time of the virtual channel will be synchronized (any value)

Example

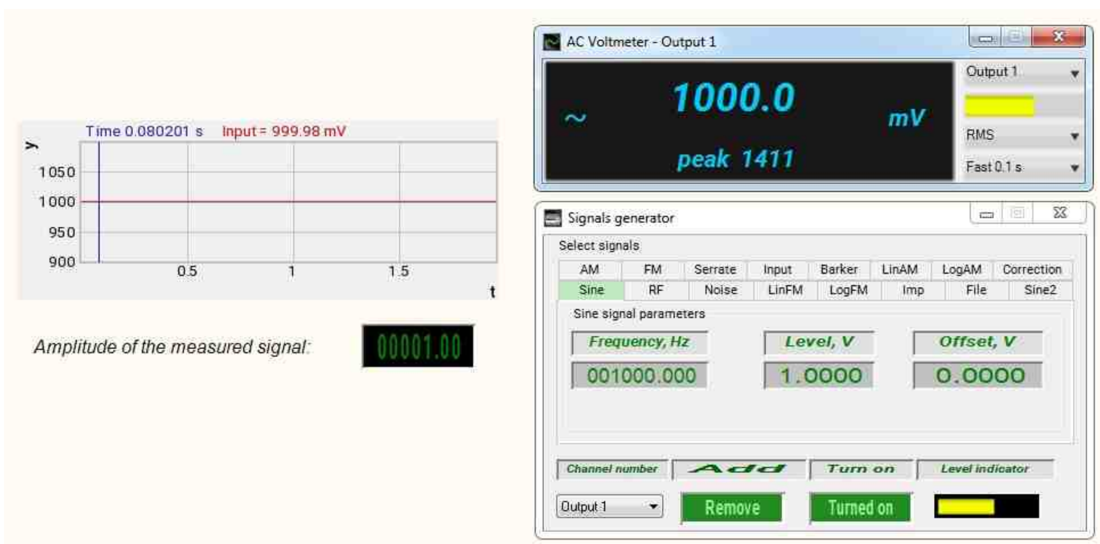
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



This project shows the operation of the [Virtual input](373) component converts a set of RMS values taken from an [AC Voltmeter](585) into a channel, the values of which are displayed on the screen using a [Multi-channel oscilloscope](928). The [Selector](428) is needed to set the amplitude of the Sine signal received using a [Sine signal](530).

Project operation result



View the example in ZetView

7.2.Measuring channel

The component "**Measuring channel**" component. Allows to accept a signal from the output of an external device. It has only one parameter – the selected channel that is used for signal transfer. It is impossible to receive data from external devices without this component.

For the Examples to work, it is necessary to have a connected physical ZET device.

Developer interface	Operator interface
	

Setting:

➢Input:

Doesn't have.

➢ Output:

- Output - measuring channel with which the component connected to it will operate (data from the ADC channel, virtual DAC).

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Channel (text) - set the channel name.
- ActiveColor (a0e1a0) - set the background color in the active status. The default is green.
- InactiveColor (e1a0a0) - set the background color in the inactive status. The default is red.
- showlist (Yes) - enable / disable the display of a drop-down list with channels.
- NewStyle (false) - new display style.
- Filter (No) - channel list filter: No;String;Unit;Group name;
- filterText (false) - string for filtering channels.

By double-clicking the left mouse button on the Measuring channel 375 component in the Operator Interface at the design stage of the project, and by clicking the left or right mouse button on the component during project execution, the channel selection menu appears:

When a component is added to a project, it does not pass any value from the output. When selecting a measurement or virtual channel, the component is displayed in green if the channel exists in the server's channel list. If a non-existent channel is set, the component changes its color to red:

Connecting contacts

Contact No.1 - "Channel". Designed to transfer the ID (identification number) of the server channel to the connected components. This pin supports multiple connection and is output. The interface of the contact is "Integer".

Returned measurement channel statuses:

-4: channel >= QuanChan;
-3: channel < 0;
-2: ZetServer.exe not running;
-1: there was no connect;
0: ADC (can be network);
1: DAC;
2: virtual;
3: digital;
4: disabled channel ADC;
5: channel of the disabled ADC;
6: digital sensor (series 7000, FreqADC can be greater than 100 Hz);
7: disabled digital sensor;
8: fast channel (with a large FreqADC, which issues data in packets, by timer or by event);
9: inactive virtual channel;
11: the DAC channel of the new analyzer;
12: demo channel.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- [Are available by the link below.](154)

➢**Custom private properties (the default value is shown in parentheses):**

•BSTR Channel - Set the channel name (String).
•LONG ActiveColor - Set the background color in the active status (number).
•LONG InactiveColor - Set the background color in the inactive status (number).
•BSTR ShowList - Set to show a drop-down list with channels: Yes; No (string).
VARIANT_BOOL NewStyle - Seta new display style:

- true - the new display style enabled;
  false - the new display style is disabled.

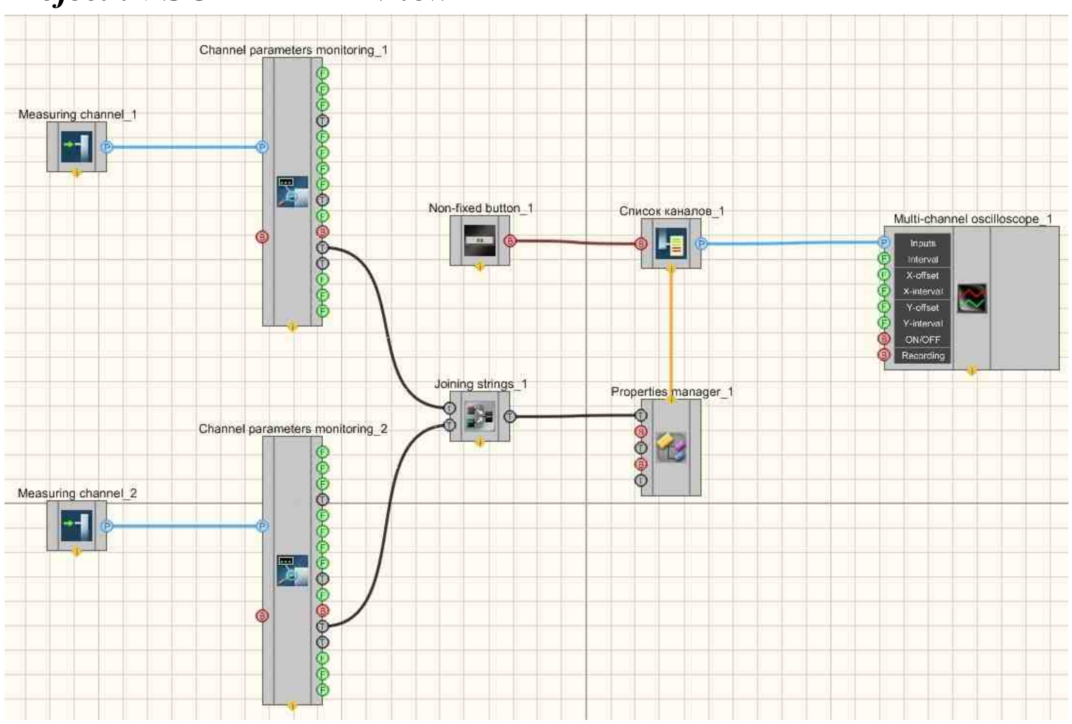
•BSTR Filter - Set the filter for channel list: No; String; Unit of measurement; Group name (string).
•BSTR filterText - Set the string for filtering channels (string).

➢**Methods:**

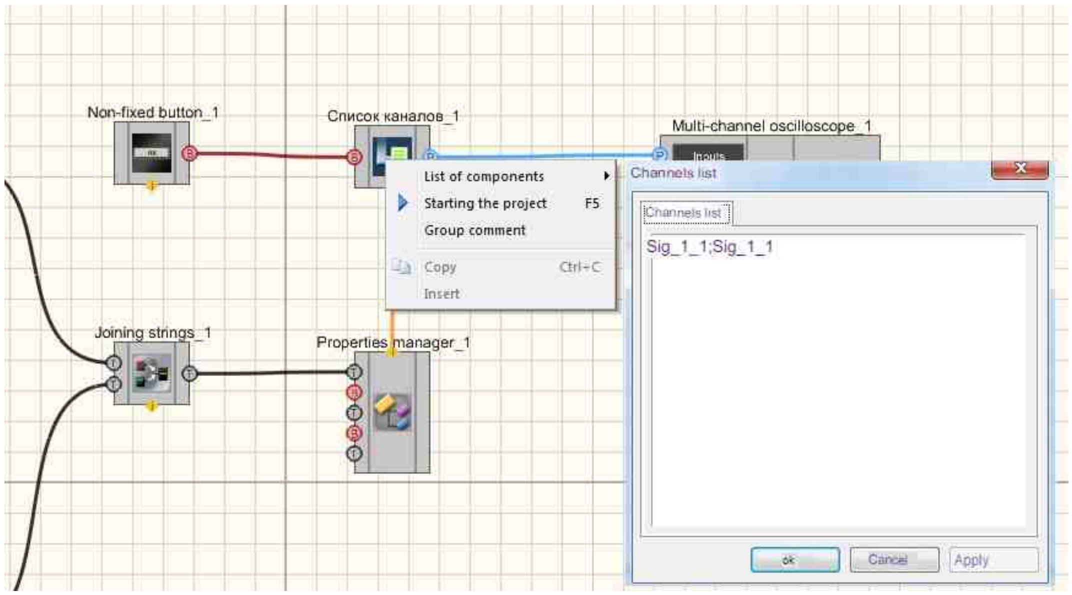
- void StopWorking() - Set the method for starting/stopping worker threads.

Example No. 1

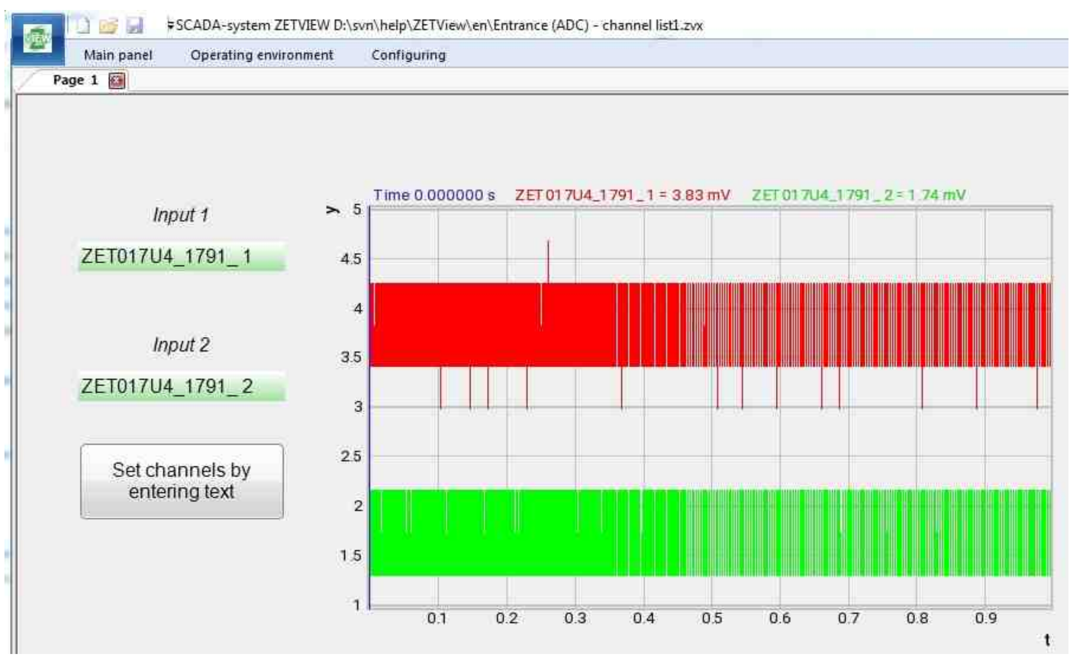
Project in SCADA ZETView



The figures below show an example of the implementation of a Multi-channel oscilloscope. Used components: Measuring channels, Monitoringing channel parameters, Non-fixed button, Channel list with channel enumeration, Joining strings, Property manager, Multi-channel oscilloscope. Editing of channel lists occurs when you click on the component with the right mouse button. Next, click Edit Component and then list the names of the channels and put a separator;



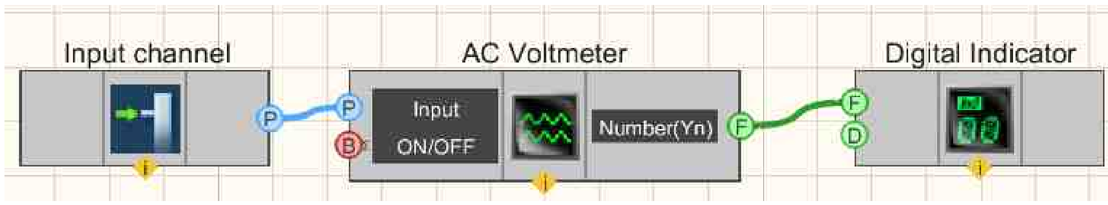
Project operation result



View the example in ZetView

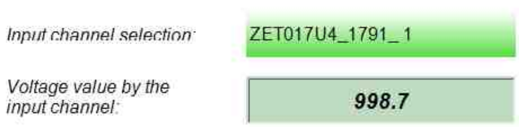
Example No. 2

Project in SCADA ZETView



This project shows the operation of the component [Measuring channel](375) sends data from the selected channels. Further, in order to graphically present the information on the display, the converted signal from the [AC Voltmeter](585) goes to the [Digital Indicator](682). The result of the compiled program can be seen in the operator interface.

Project operation result



View the example in ZetView

7.3. Device information

The component "**Device information**" is allows the user to receive information about the device number from the information received via the connected channel.

For the Examples to work, it is necessary to have a connected physical ZET device.

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Channel - the channel of the device on which information is required.
-

➢ Output:

- Ser. nom. - serial number of the device.
- Type - device name.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

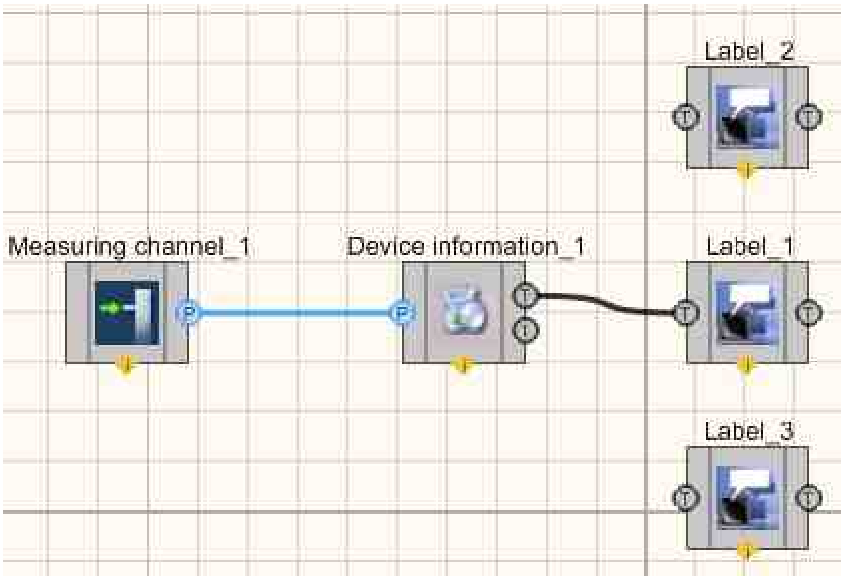
Setting:

➢**Custom common properties:**

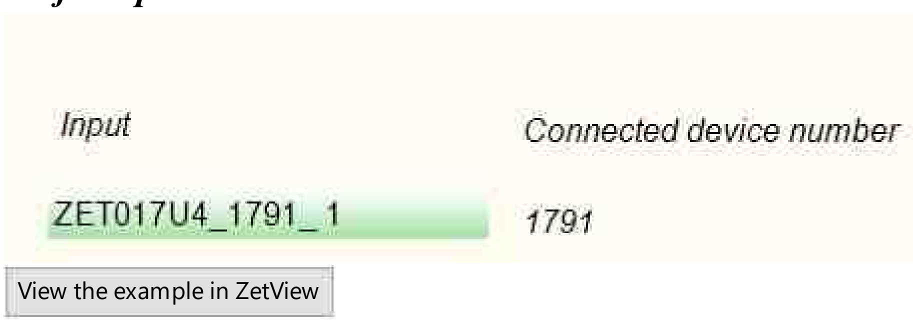
- Are available by the link below. 154

Example

Project in SCADA ZETView



Project operation result

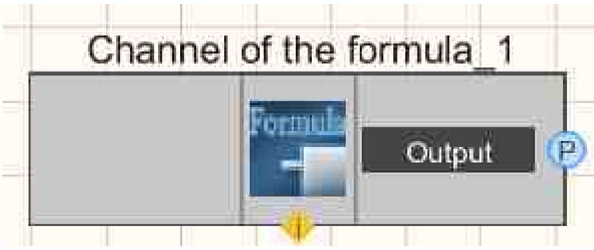


7.4.Channel of the formula

The component "**Channel of the formula**" is allows to create signals of various forms as well as to make mathematics and arithmetics operations with them.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

Doesn't have.

➢ Output:

- Output - a virtual output channel from which the instantaneous values obtained from the formula come.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- ChannelName (Output) - set the channel name which must have an alphanumeric designation.
- Conversion (mV) - unit of measurement by channel.
- MaxLevel (1000) - maximum limit level by channel, units.
- Referense (0.001) - the reference for calculating the level in dB (zero level for the reference of the scale in dB).
- Formula (0) - formula.
- MinLevel (0.001) - the minimum level by channel, units.
- Frequency (100) - the channel sampling frequency, Hz.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

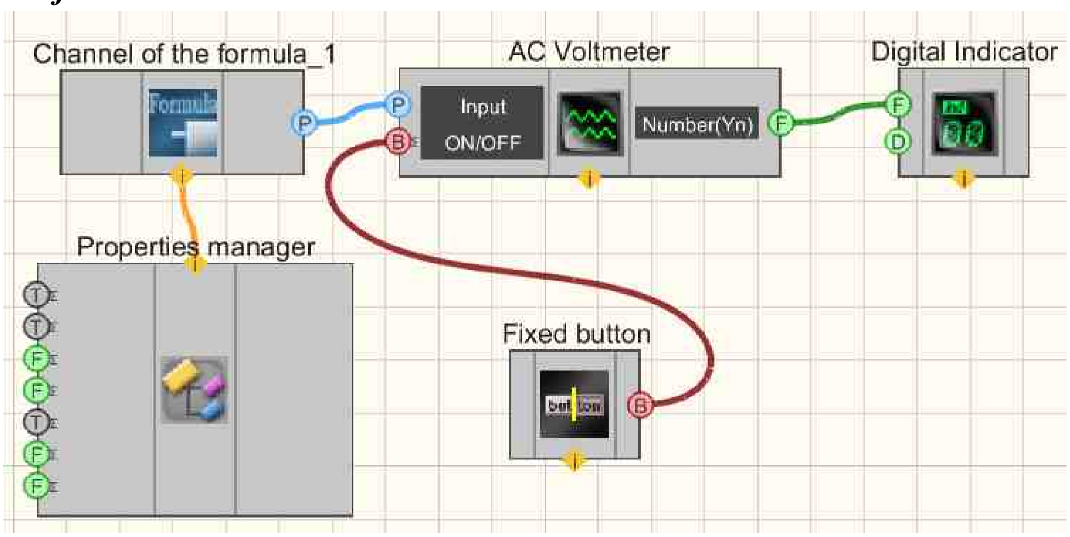
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

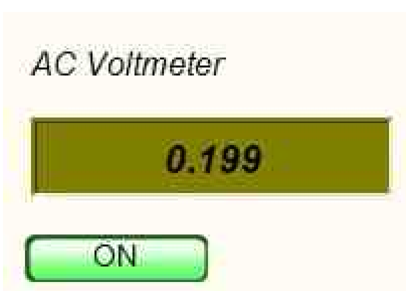
- BSTR ChannelName - set the channel name which must have an alphanumeric designation (string).
- BSTR Conversion - set the unit of measurement by channel (string).
- FLOAT MaxLevel - set the maximum limit level by channel, units (any value).
- FLOAT Referense - set the reference for calculating the level in dB (any value).
- BSTR Formula - set the formula (string).
- FLOAT MinLevel - set the minimum level by channel, units (any value).
- FLOAT Frequency - Set the channel sampling frequency, Hz (any value).

Example

Project in SCADA ZETView



Project operation result

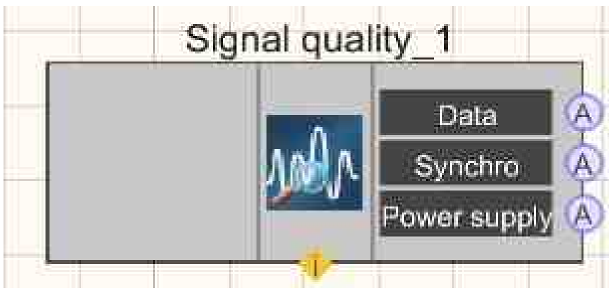


View the example in ZetView

7.5.Signal quality

The component "**Signal quality**" component. Is used for operation with the program SynchronizationControl.exe (Synchronization control), which allows to control quality of data, power supply and synchronization by channels of the devices having the option of synchronization, power supply and data control.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

Doesn't have.

➤ Output:

- Data - data quality.
- Synchro - quality of synchronization.
- Power supply - the power supply quality of the signal.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

- ChanListSource(All channels) - channel list source:All channels;Channels from the list.
- ChannelList - the list of monitored channels (transmitted through the separator ;).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

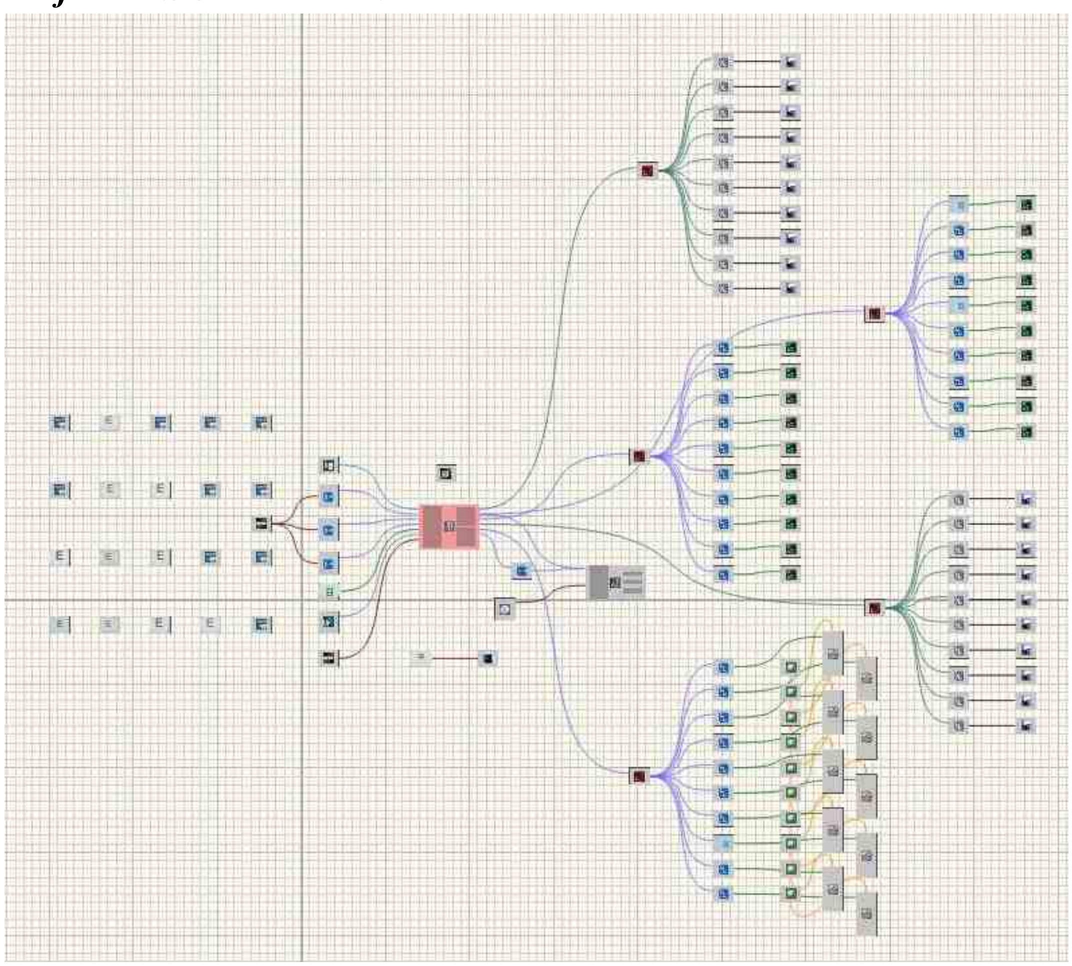
- [Are available by the link below.](154)

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR ChanListSource - Set the channel list source: All channels; Channels from the list (string).
- BSTR ChannelList - Set the list of monitored channels (listed through the delimiter ;) (string).

Example

Project in SCADA ZETView



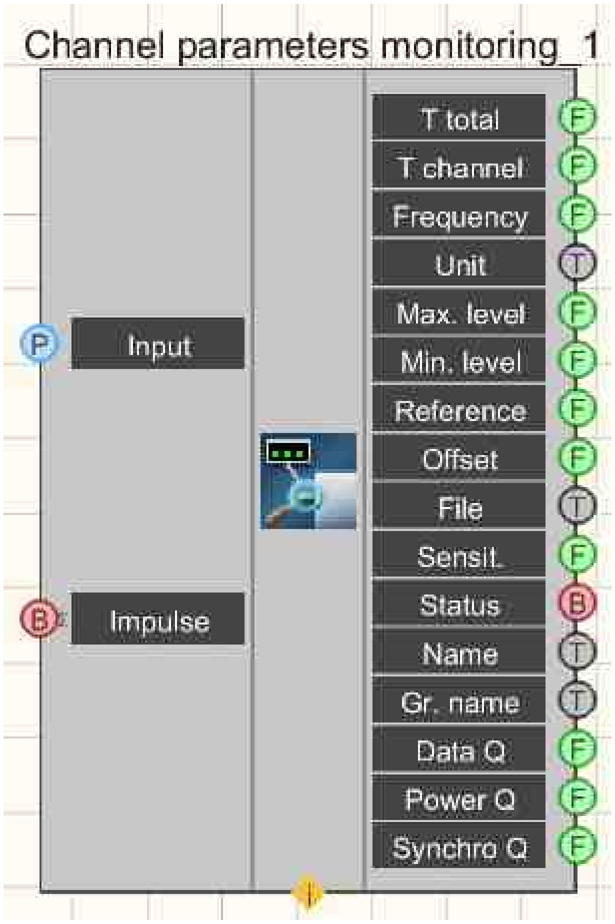
Project operation result

View the example in ZetView

7.6.Channel parameters monitoring

The component "**Channel parameters polling**" is allows to obtain signal parameters from the selected channel upon external impact or periodically;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - the measuring channel whose parameters should be monitored.
- Impulse - a synchronization at which, when the logical status changes (from a low logical level to a high one), the input channel will be monitored.

➢ Output:

- T total - output of the common (system time) channel.
- T channel - the output of the channel time.
- Frequency - the sampling frequency by channel, Hz.
- Unit - the channel units that are configured in the Device Manager program from the Service menu.
- Max. level - outputs the maximum limit level by the channel.
- Min. level - outputs the minimum limit level by the channel
- Reference - the reference value relative to which the signal level in dB is considered.

- Offset - the offset of the constant component in the channel.
- File - the name of the file in which user information about the channel can be stored, for example, the frequency-dependent frequency response of the path.
- Sensit. - channel conversion sensitivity (V/unit of measurement).
- Status - is used to determine whether the channel is working.
- Name - gives the name of the server channel.
- Gr. name - the group name of the channel in the server.
- Data Q - outputs data quality
- Power Q - outputs power supply quality.
- Synchro Q - displays the quality of synchronization.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- synchrotype (on request) - channel monitoring methods (two ways).
- 1. On request.
- 2. Auto (1 s)
- Timeout (5) - time after which the status of the channel is issued, when time does not flow through it, s.

Note. In the "**Channel parameters monitoring"** component, there is a contact "T total", with which the absolute time of the channel goes, i.e. time since the last restart of ZETServer.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**

- Input - the value of the measuring channel whose parameters should be monitored (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR SynchroType - set the type of information output: On request; Auto (1 s) (string).
- FLOAT Timeout - Set the time after which the channel status is displayed when time does not flow through it, s (from 0 to plus infinity).

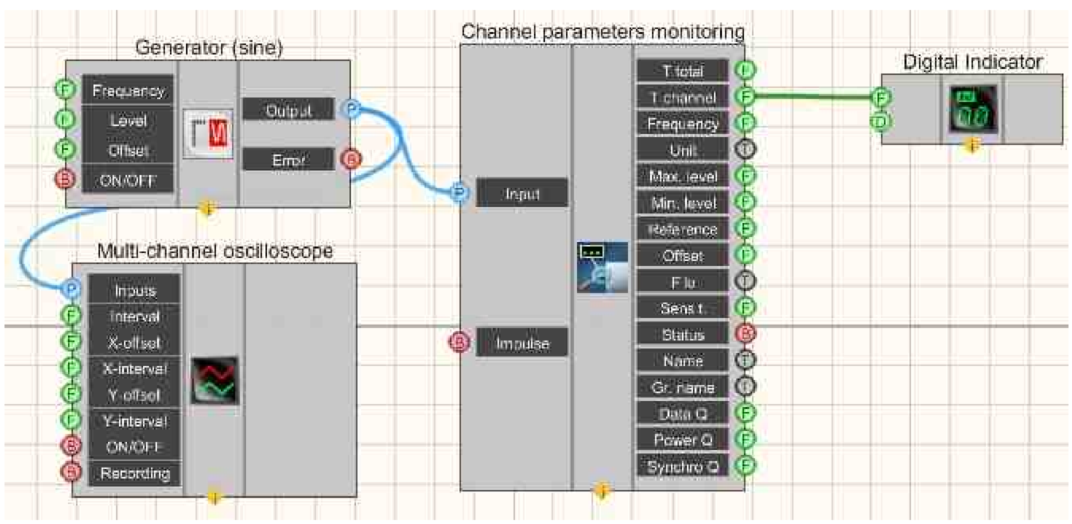
➢**Methods:**

- float GetWorkingTime(void) - set the total time of ZETServer, s.
- float GetChannelTime(void) - set the current time by the channel, s.
- void Synchro(void) - Updates data for a channel.

Example

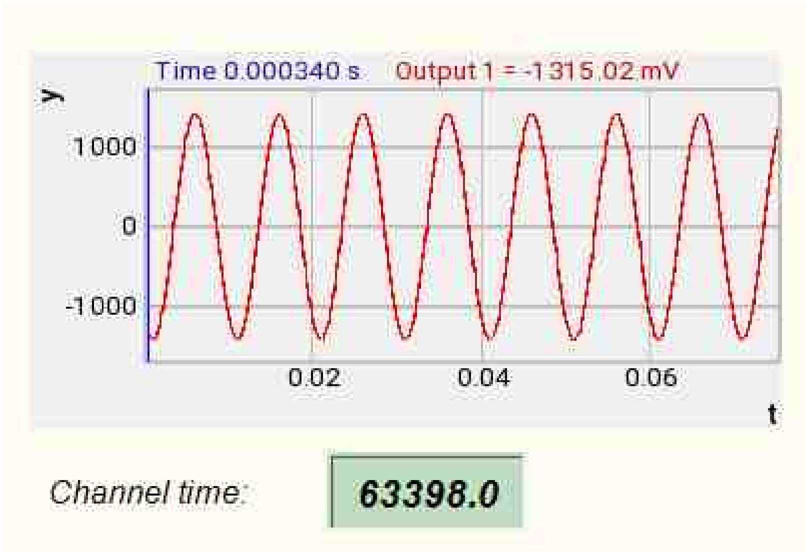
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



In this scheme, a Sine signal 530 is fed from a sine signal to the Channel parameters monitoring 387. The Digital Indicator 682 is used to show the channel time on the display. A Multi-channel oscilloscope 928 is needed to view the signal coming from the generator.

Project operation result



View the example in ZetView

7.7.Channels list

The "**Channels list**" component is used for sorting and can be set in the properties setting window or selected by the operator during the project. The type and sorting conditions are specified when setting up the component.

The "**Channels list**" component selects the list of channels by the operator and transmits the ID of the selected channels to the connected components. When the component receives a impulse, a dialog box opens in which the operator selects the channels to display. When the window is closed with the OK button, the list of selected channels is transferred to the connected components. Additionally, the **Channels list** component implements the functions of filtering the selected channels by name and/or by measurement units.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- Impulse - impulse to correct the channel lists.

➢ Output:
- Outputs - the output is a sorted list of channels.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Channels - list of channels for sorting (listed with a "semicolon" separator).
- SortName (false) - enable channel sorting by name part.
- Name (Signal) - part of the channel name by which the channels are sorted when sorting by part of the channel name is enabled.
- SortConversion (false) - enable sorting of channels by unit of measure per channel.
- Conversion (mV) - unit of measure for the channel.

Working with a component

After starting the project, when the "**Channels list**" component receives an impulse (in the Example, from a connected button), the "**Channels list**" window opens, which displays the channels selected for sorting. If no channel is selected, the field remains empty. When you right-click the "**Channels list**" field, a context menu appears that displays all the channels of the zet-server. When you select a channel, it is added to the list for sorting. When you select All channels, all channels of the zet-server are added to the list. When the window is closed with the "OK" button, the list of selected channels is sorted and the component outputs the list of channels that meet the sorting conditions. When the window is closed with the "Cancel" button, the list change is ignored and sorting is not performed.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

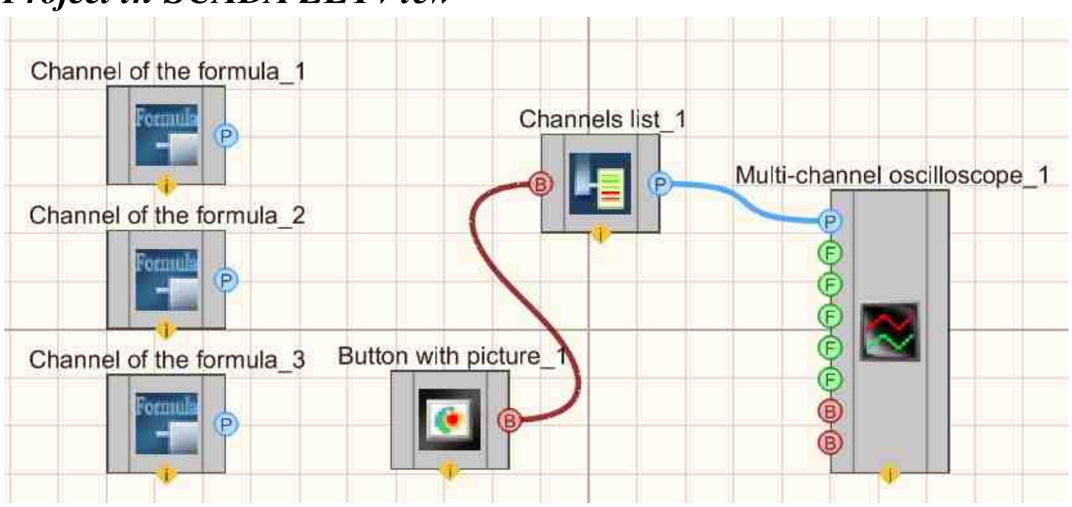
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

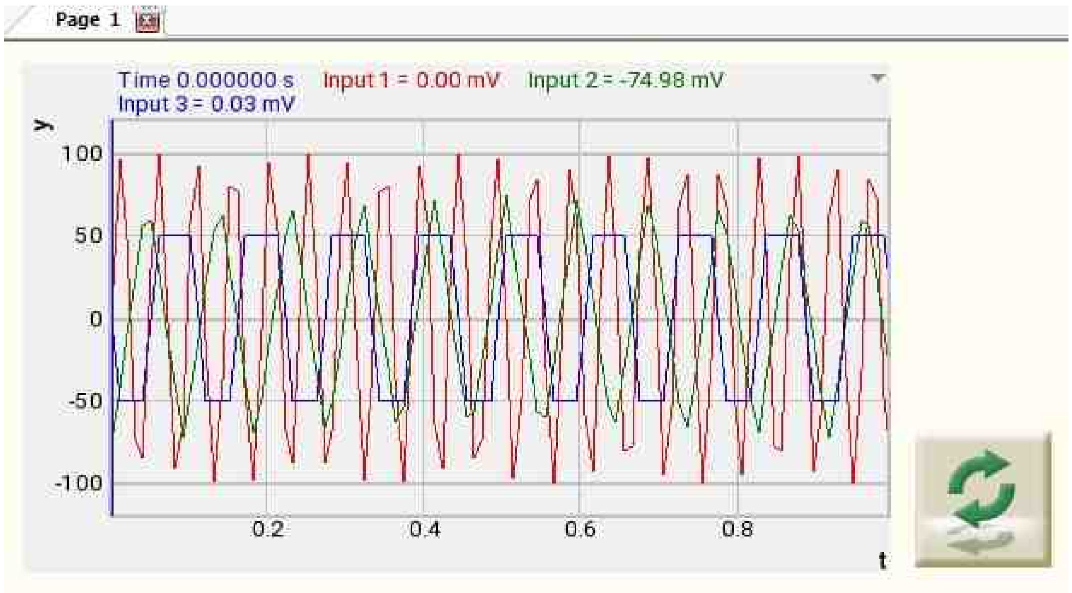
- BSTR Channels - Set the list of channels for sorting (listed through the delimiter ;) (string).
- VARIANT_BOOL SortName - Set the permission to sort channels by part of the name:
- true - enabled to sort channels by part of the name;
- false - disabled ability to sort channels by part of the name.
- BSTR Name - Set the part of the channel name by which channels are sorted when sorting by part of the channel name is enabled (string).
- VARIANT_BOOL SortConversion - Set the resolution of channel sorting by unit of measure per channel:
- true - enabled to sort of channels by unit of measure per channel;
- false - disabled the ability to sort channels by unit of measure per channel.
- BSTR Conversion - Set the unit of measure by channel by which channels are sorted when sorting by unit of measure (string). is enabled.

Example No. 1

Project in SCADA ZETView



The figures below show an example implementation of a Multi-channel oscilloscope. Used components: Button with picture, Channel list, Multi-channel oscilloscope. The Channel list component is used here to create a list of channels to display, since simply connecting multiple channels to the Multi-channel Scope is not enough to display them all at the same time.

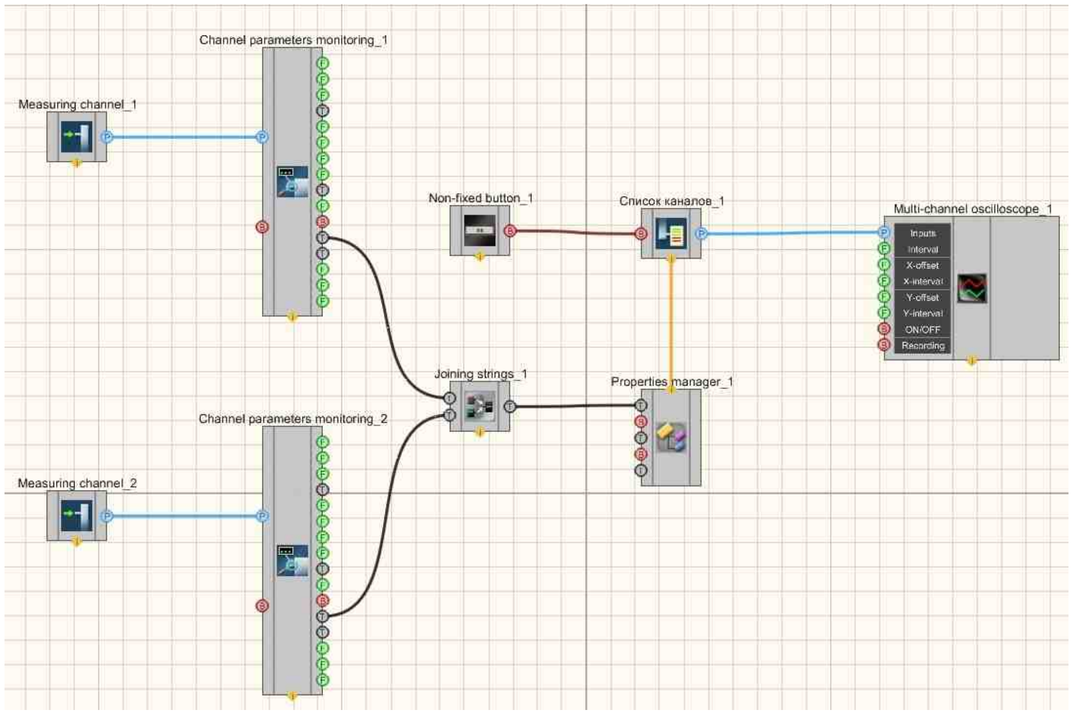


Project operation result

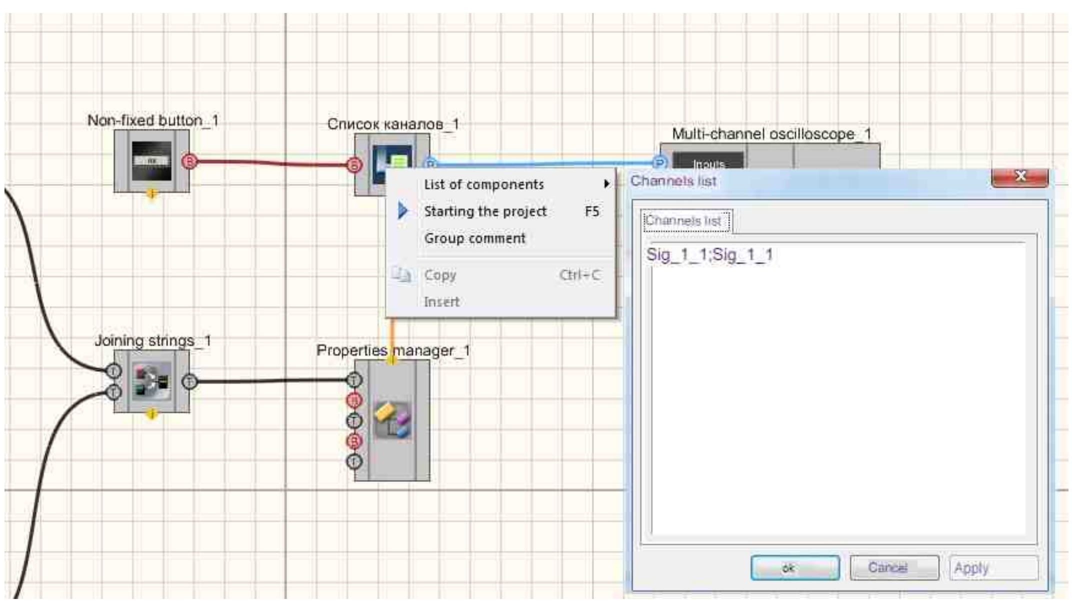
View the example in ZetView

Example No. 2

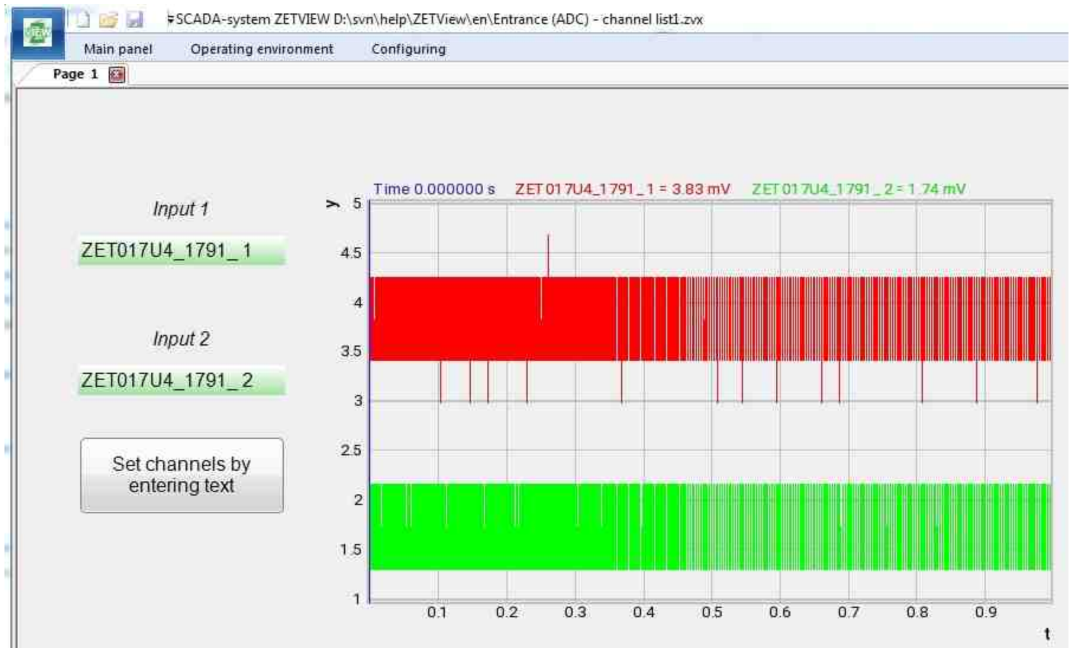
Project in SCADA ZETView



The figures below show an example of the implementation of a Multi-channel oscilloscope. Used components: Measuring channels, Monitoringing channel parameters, Non-fixed button, Channel list with channel enumeration, Joining strings, Property manager, Multi-channel oscilloscope. Editing of channel lists occurs when you click on the component with the right mouse button. Next, click Edit Component and then list the names of the channels and put a separator;



Project operation result

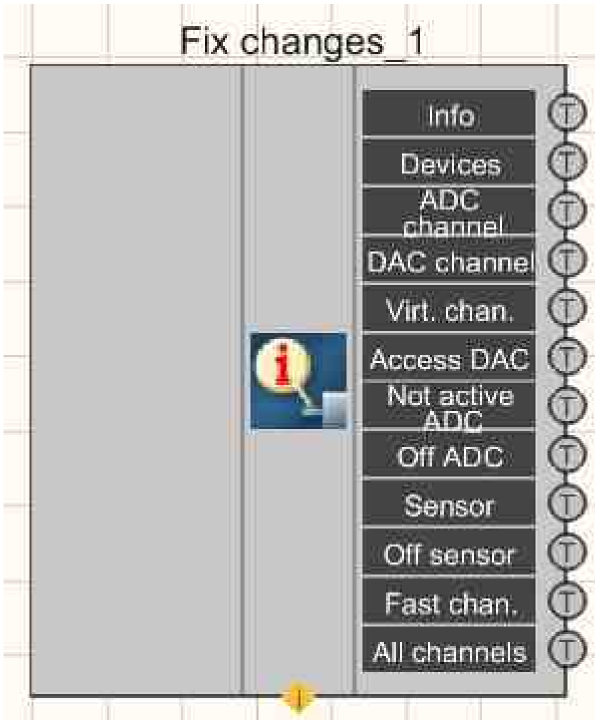


View the example in ZetView

7.8.Fix changes

The component "**Fix changes**" is allows to save the properties changes of devices, system channels (corresponding data is obtained from ZETServer component).

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤ Output:

- Info - contains information about channels.
- Devices - contains a list of devices.
- ADC channel - the current list of enabled ADC channels.
- DAC channel - the current list of enabled DAC channels.
- Virt. chan. - the current list of enabled virtual channels.
- Access DAC - the current list of available DAC.
- Not active ADC - the list of disabled ADC channels.
- Off ADC - the channel of the disabled ADC.
- Sensor - an digital sensor.
- Off sensor - disabled digital sensor.
- Fast channel - fast channel - contains in order the list of channels present on the server.
- All channels - a complete list of channels in order present on the server.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- ADCChannels (true) - monitoring of ADC channels.
- DACChannels (true) - monitoring of DAC channels.
- VirtualChannels (true) - monitoring of virtual channels.
- Devices (true) - monitoring of external devices.

Note:
1. The list of possible generator channels is formed in the component itself, so it must be named or translated in the same way as in the programs that create generators, in Example, in the SRV.ocx server.
2. It is worth getting the list of generator channels not from the "Access DAC" contact, but from the "DAC channel" contact - there the name of the channels is taken from the SRV.ocx server

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- Value (Yn) - the value recorded to the create virtual channel (any type).

➢**Custom common properties:**
- Are available by the link below. 154

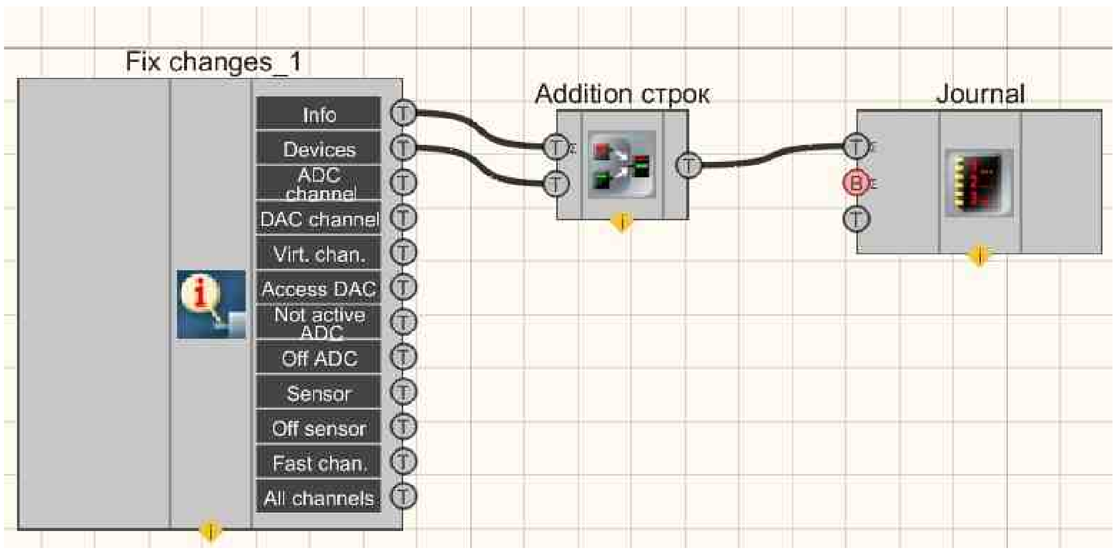
➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL ADCChannels - Set regular monitoring of ADC channels (monitoring):
- true - the regular monitoring of ADC channels is enabled;
- false - the regular monitoring of ADC channels is disabled.
- VARIANT_BOOL DACChannels - Set regular monitoring of DAC channels (monitoring):
- true - the regular monitoring of DAC channels is enabled;
- false - the regular monitoring of DAC channels is disabled.
- VARIANT_BOOL VirtualChannels - Set regular monitoring of virtual channels (monitoring):
- true - regular monitoring of virtual channels is enabled;
- false - regular monitoring of virtual channels is disabled.
- VARIANT_BOOL Devices - Set regular monitoring of devices (monitoring):
- true - regular monitoring of devices is enabled;

- false - regular monitoring of devices is disabled.

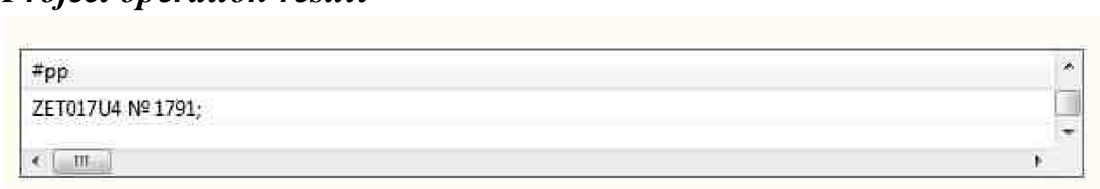
Example

Project in SCADA ZETView



This project shows the operation of the Fix changes 396 component logs changes that occur with one of the external plug-ins in the Events journal 1037 component. But after preliminary gluing of information about the device and the changes occurring with it (Joining strings 1094 component).

Project operation result



View the example in ZetView

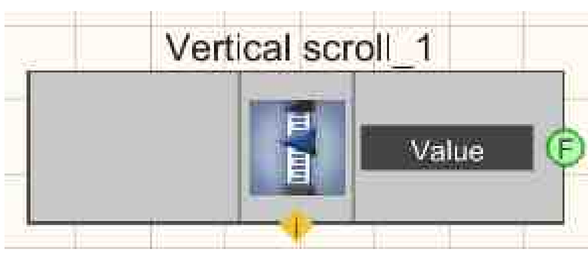
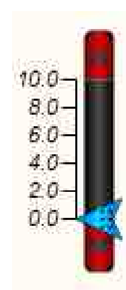
Глава 8.Option

In the present, it is hardly possible to find a program that would not inform the user of its actions and intermediate results, especially when it comes to measurements and data processing. Graphical interface, control elements, various notifications and other means of user interaction form an integral part of any modern application. In the first part of the present user manual, it has been mentioned that it is possible to compile the projects to exe-file. This format implies that the operator does not have a possibility to get access to project composition or to change settings of components properties. In fact, the project is represented as an independent program. Hence, it is important to provide the operator with an opportunity to interact with the project and to view the results.

8.1.Vertical scroll

The component "**Vertical scroll**" is designed to select a value by moving the pointer on the scale and then transferring the value to the channel.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢ Output:
- Value - the control signal at the output.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- TextSize (8) - text size.
- textstyle (FontStyleItalic) - text style
- CurrentValue (0) - set the current value.

- DigitFrom (0) - initial value.
- DigitTo (10) - final value.
- ScrollStep (0) - the cursor step.
- accuracy (0.1) - the measurement accuracy.
- LinearGap (4) - the ruler offset from the scroll bar.
- LinearsHeight (7) - the size of the large strokes of the ruler.
- LinearsUnderline (true) - the underline of the ruler.
- ScrollerWidth (20) - the width of the pointer.
- WidthSideRects (25) - the width of the side rectangles.
- TextColor (000000) - the text color.
- BackColorUp (000000) - the up background color.
- BackColorDown (808080) - the down background color.
- SideRectsUpColor (f40000) - the up color of the side rectangle gradient.
- SideRectsDownColor (7f0000) - the down background of the side rectangle gradient.
- CursorUpColor (00f4ff) - the up color of cursor gradient
- CursorDownColor (007fff) - the down color of cursor gradient
- cursorform (Arrow) - cursor form:
- 1. Arrow.
- 2. Ellipse.
- 3. Rectangle.
- 4. Triangle.
- LinearVisible (true) - linear visibility.
- linearorientation (left) - display side of the ruler:
- 1. left - left side.
- 2. right - right side.
- Enabled (true) - enable/disable the component.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

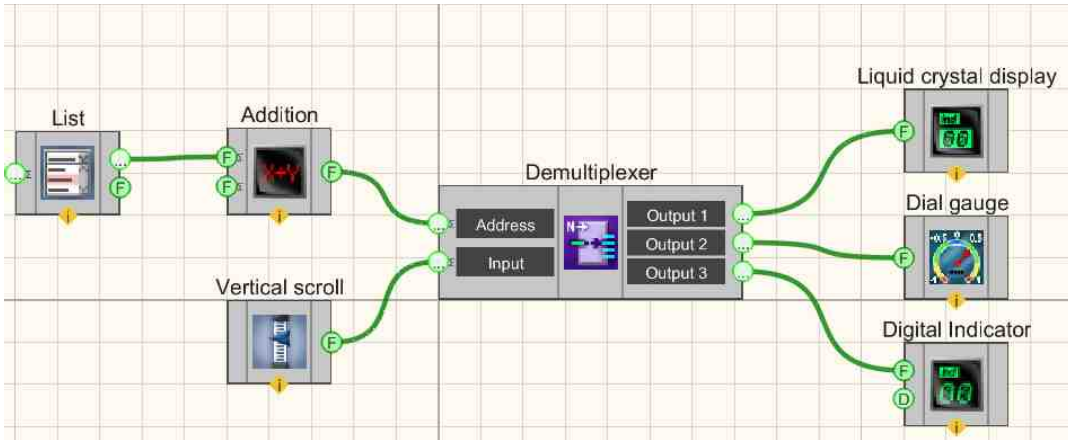
➢**Custom private properties (the default value is shown in parentheses):**

- SHORT TextSize - Set the text size (any number).

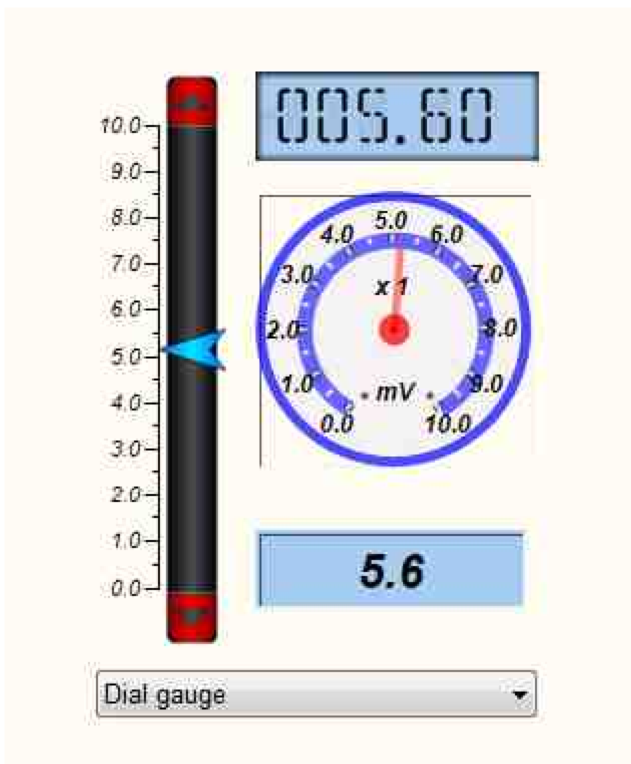
- BSTR TextStyle - Set the text style:FontStyleRegular; FontStyleBold; FontStyleItalic; FontStyleBoldItalic; FontStyleUnderline; FontStyleStrikeout (string).
- float CurrentValue - Set the current value (from minus infinity to plus infinity).
- float DigitFrom - Set the initial value (from minus infinity to plus infinity).
- float DigitTo - Set the final value (from minus infinity to plus infinity).
- float ScrollStep - Set cursor step (from minus infinity to plus infinity).
- BSTR Accuracy - Set the accuracy: 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- SHORT LinearGap - Set the ruler offset from the scroll bar (any number).
- SHORT LinearsHeight - Set the size of large ruler strokes (any number).
- VARIANT_BOOL LinearsUnderline - Set the underline of the ruler:
  true - the output value is given when the ruler underline is enabled;
  false - the output value is not given when the ruler underline is disabled.
- SHORT ScrollerWidth - Set the pointer width (any number).
- SHORT WidthSideRects - Set the width of the side rectangles (any number).
- LONG TextColor - Set the text color (any number).
- LONG BackColorUp - Set the up background color (any number).
- LONG BackColorDown - Set the down background color (any number).
- LONG SideRectsUpColor - Set the up gradient color of the side rectangles (any number).
- LONG SideRectsDownColor - Set the down gradient color of the side rectangles (any number).
- LONG CursorUpColor - Set the up color of the cursor gradient (any number).
- LONG CursorDownColor - Set the down color of the cursor gradient (any number).
- BSTR CursorForm - Set the cursor form: Arrow; Ellipse; Rectangle; Triangle (string).
- VARIANT_BOOL LinearVisible - Set ruler visibility:
  true - the ruler visibility is enabled;
  false - the ruler visibility is disabled.
- BSTR LinearOrientation - Set the display side of the ruler: left side; right side (string).
- VARIANT_BOOL Enabled - Set the enable/disable the component:
  true - the output value is given when the component is turned ON;
  false - the output value is not given when the component is turned OFF.

Example

Project in SCADA ZETView



Project operation result

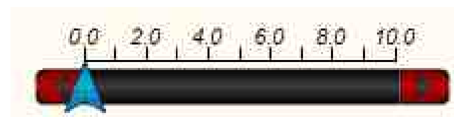


View the example in ZetView

8.2.Horizontal scroll

The component "**Horizontal scroll**" is designed to select a value by moving the pointer on the scale and then transmitting the value to the channel.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢ Output:

- Value - the control signal at the output.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- TextSize (8) - text size.
- textstyle (FontStyleItalic) - text style
- CurrentValue (0) - set the current value.
- DigitFrom (0) - initial value.
- DigitTo (10) - final value.
- ScrollStep (0) - the cursor step.
- accuracy (0.1) - the measurement accuracy.
- LinearGap (4) - the ruler offset from the scroll bar.
- LinearsHeight (7) - the size of the large strokes of the ruler.
- LinearsUnderline (true) - the underline of the ruler.
- ScrollerWidth (20) - the width of the pointer.
- WidthSideRects (25) - the width of the side rectangles.
- CurrentValVisible (false) - the display of the current value.
- TextColor (000000) - the text color.

- BackColorUp (000000) - the up background color.
- BackColorDown (808080) - the down background color.
- SideRectsUpColor (f40000) - the up color of the side rectangle gradient.
- SideRectsDownColor (7f0000) - the down background of the side rectangle gradient.
- CursorUpColor (00f4ff) - the up color of cursor gradient
- CursorDownColor (007fff) - the down color of cursor gradient
- cursorform (Arrow) - cursor form:
  1. Arrow.
  2. Ellipse.
  3. Rectangle.
  4. Triangle.
- LinearVisible (true) - linear visibility.
- linearorientation (Up) - ruler display side:
  1. Up - above the ruler.
  2. Down - under the ruler.
- Enabled (true) - enable/disable the component.
- scale (linear) - scale scale linear/logarithmic.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

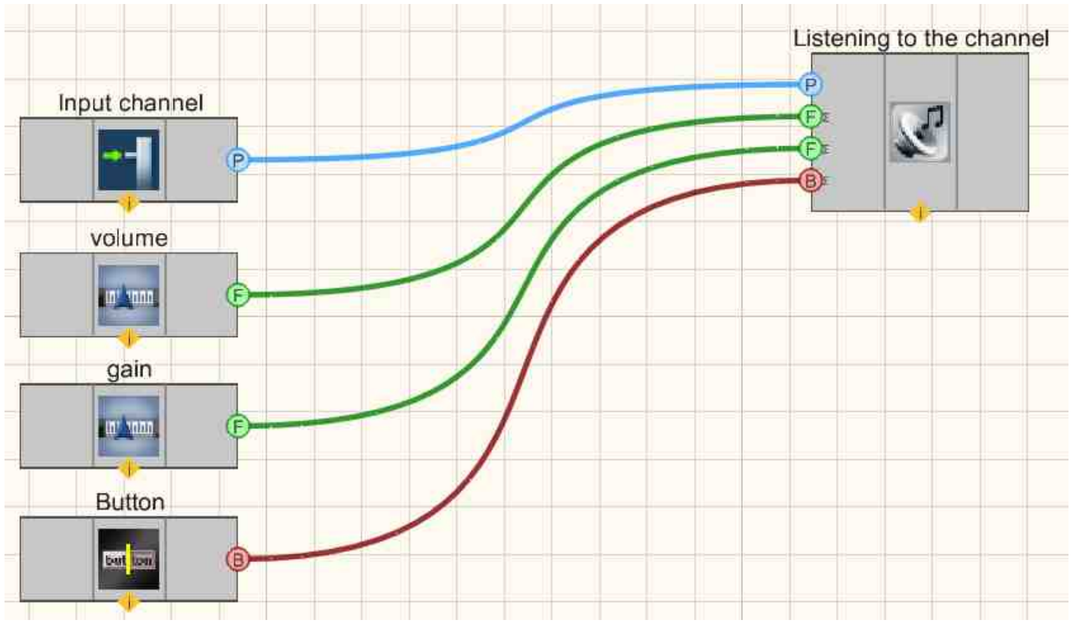
➢**Custom private properties (the default value is shown in parentheses):**

- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set the text style:FontStyleRegular; FontStyleBold; FontStyleItalic; FontStyleBoldItalic; FontStyleUnderline; FontStyleStrikeout (string).
- float CurrentValue - Set the current value (from minus infinity to plus infinity).
- float DigitFrom - Set the initial value (from minus infinity to plus infinity).
- float DigitTo - Set the final value (from minus infinity to plus infinity).
- float ScrollStep - Set cursor step (from minus infinity to plus infinity).
- BSTR Accuracy - Set the accuracy: 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- SHORT LinearGap - Set the ruler offset from the scroll bar (any number).
- SHORT LinearsHeight - Set the size of large ruler strokes (any number).

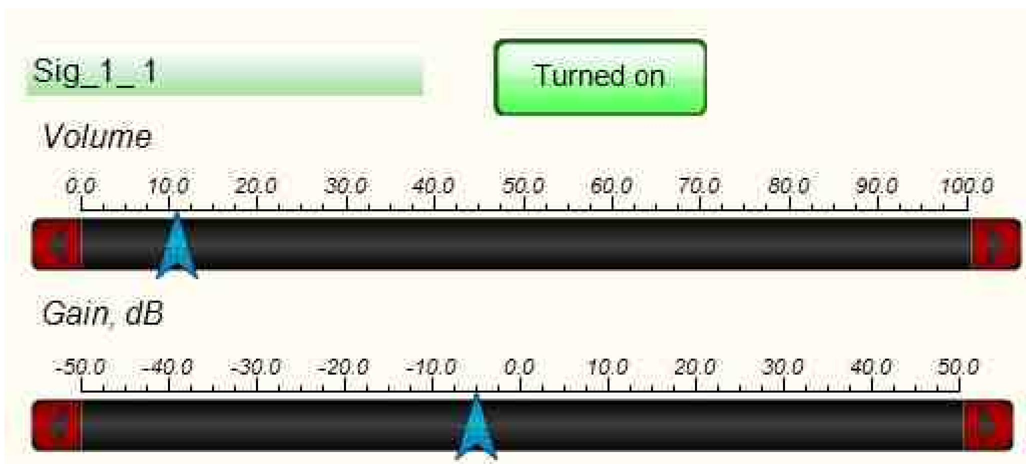
- VARIANT_BOOL LinearsUnderline - Set the underline of the ruler:
  true - the output value is given when the ruler underline is enabled;
  false - the output value is not given when the ruler underline is disabled.
- SHORT ScrollerWidth - Set the pointer width (any number).
- SHORT WidthSideRects - Set the width of the side rectangles (any number).
- VARIANT_BOOL CurrentValVisible - Set the current value display:
  true - display of the current value is enabled;
  false - display of the current value is disabled;
- LONG TextColor - Set the text color (any number).
- LONG BackColorUp - Set the up background color (any number).
- LONG BackColorDown - Set the down background color (any number).
- LONG SideRectsUpColor - Set the up gradient color of the side rectangles (any number).
- LONG SideRectsDownColor - Set the down gradient color of the side rectangles (any number).
- LONG CursorUpColor - Set the up color of the cursor gradient (any number).
- LONG CursorDownColor - Set the down color of the cursor gradient (any number).
- BSTR CursorForm - Set the cursor form: Arrow; Ellipse; Rectangle; Triangle (string).
- VARIANT_BOOL LinearVisible - Set ruler visibility:
  true - the ruler visibility is enabled;
  false - the ruler visibility is disabled.
- BSTR LinearOrientation - Set the display side of the ruler:Up;Down (string).
- VARIANT_BOOL Enabled - Set the enable/disable the component:
  true - the output value is given when the component is turned ON;
  false - the output value is not given when the component is turned OFF.
- BSTR Scale - Set the scale:Linear;Logarithmic (string).

Example

Project in SCADA ZETView



Project operation result

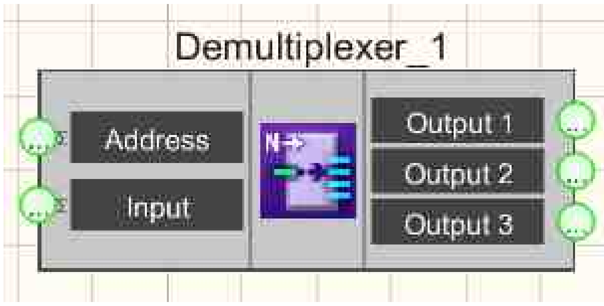


View the example in ZetView

8.3.Demultiplexer

The component "**Demultiplexer**" is designed to connect a single input channel to one of several outputs defined by the address.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Address - receives the address of the output to which the input will be connected.
- Input - a channel is connected, which will be switched in the future.

➢ Output:
- Output - the input channel is connected if the address is selected with using Output 1.
-
- ...
- Output N - the input channel is connected if the address is selected with using Output N.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Output (1) - set the number of outputs.
- Status (1) - the current connection of the demultiplexer.

Note;

The multiplexer will only transmit data further if it has the "m_sHelpString" field filled in.Without specifying "m_sHelpString", the Graphic 909 component will not display an array passed through the multiplexer.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Address - receives the address of the output to which the input will be connected (any type).
- Input - a channel is connected, which will be switched in the future (any type).

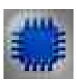
➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

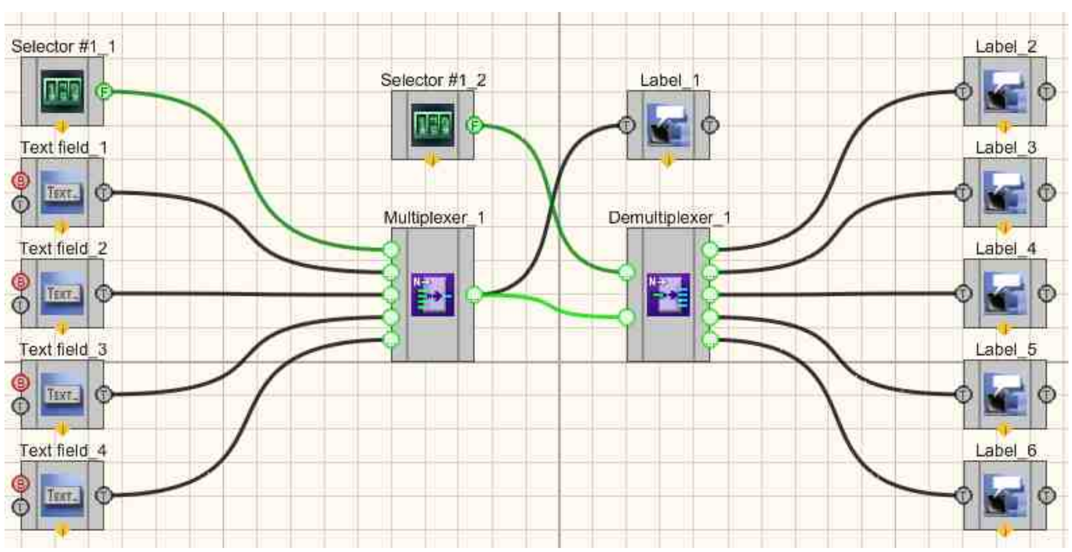
•LONG Ouput - Set the number of outputs (0 to 50)

•LONG Status - Set the current status of the demultiplexer (any number).

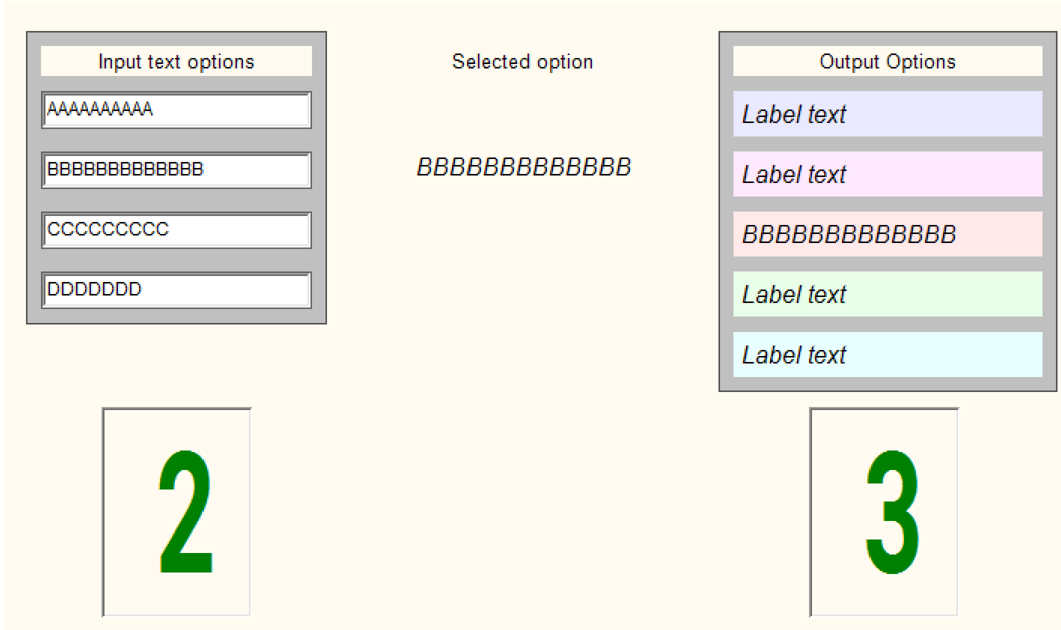


Example No. 1

Project in SCADA ZETView



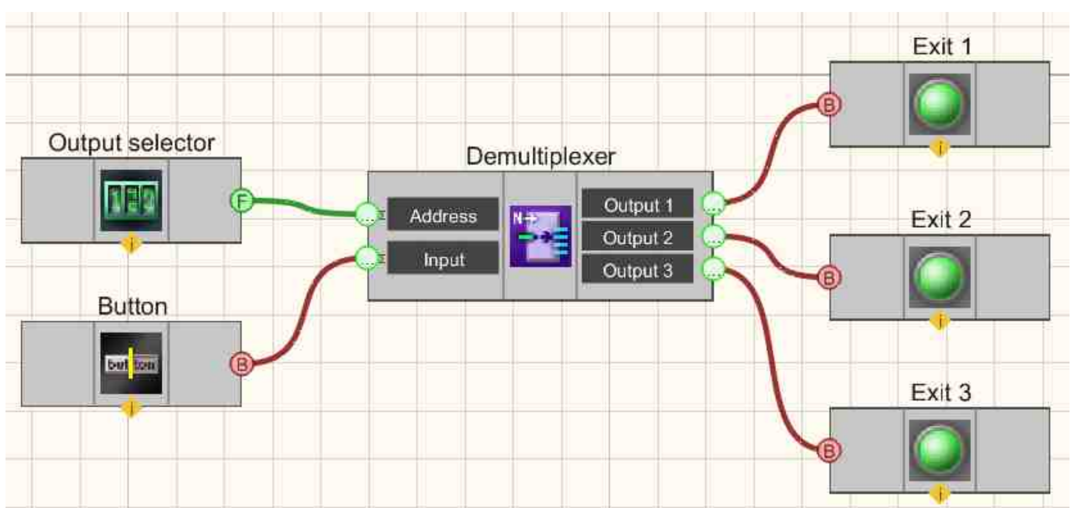
Project operation result



View the example in ZetView

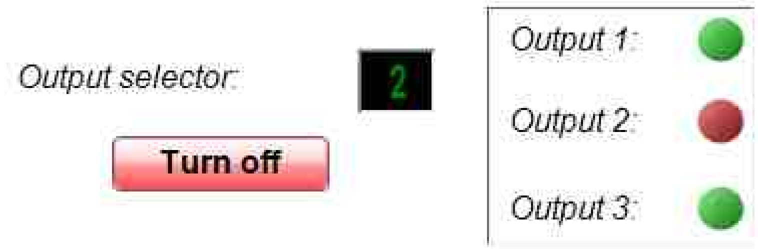
Example No. 2

Project in SCADA ZETView



This project shows the operation of the Demultiplexer 407 component is used to connect a channel with a Fixed button 691 (in this channel, a high or low logic level is set depending on the status of pressing the button). The Selector 428 allows us to choose which output channel we will connect the channel with the button to.

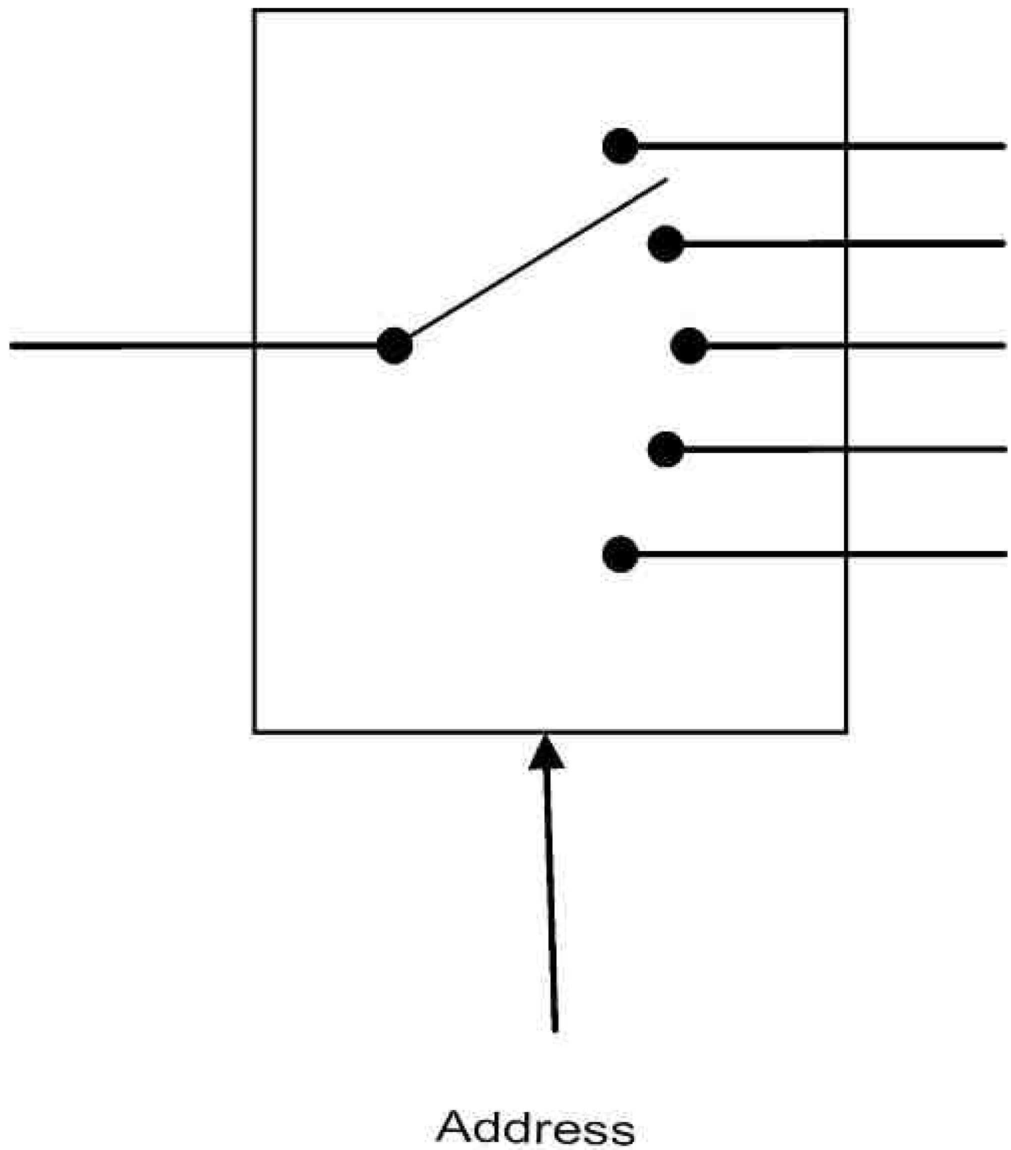
Project operation result



View the example in ZetView

Mathematical description

A demultiplexer is a device that receives an input signal and directs it to one of several outputs in accordance with the binary code acting on the address inputs. In this case, the remaining outputs are either in an inactive status or in an open circuit status. The decoder works similarly. The only difference is that only the address that excites one of the n possible outputs is fed to the inputs.

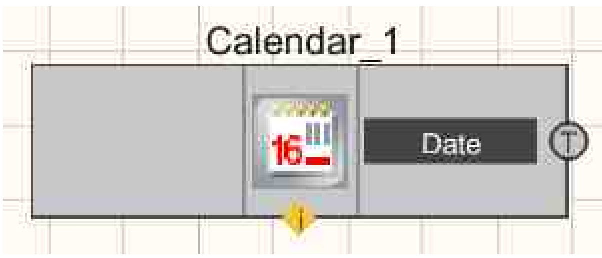
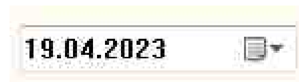


Decoders are usually used when interfacing with a microprocessor, when it is necessary to perform various actions depending on the address. Another application of the common use of the decoder is the organization (resolution) of the sequence of actions, according to the reached address specified by the output of the binary counter.

8.4.Calendar

The component "**Calendar**" is designed to select and set a specific date and subsequently use it in projects.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢ Output:
- Date - the selected date.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- dateformat (DD.MM.YY) - set the current date.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

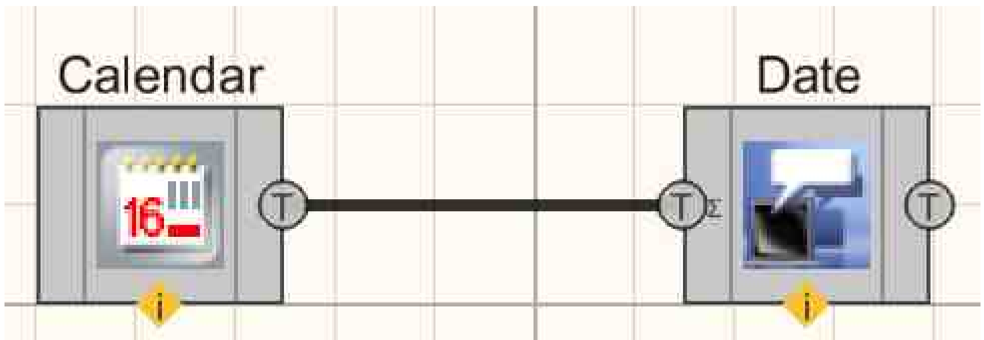
➢Custom common properties:
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•BSTR DateFormat - Set Date format: DD.MM.YY; YYYY-MM-DD (string).

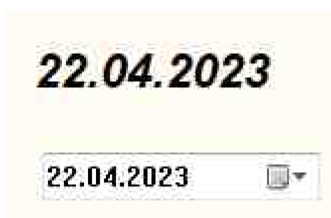
Example

Project in SCADA ZETView



This project shows the operation of the Calendar 413 component is used to set the current date value. The Label 1087 is used to additionally display the transmitted date value.

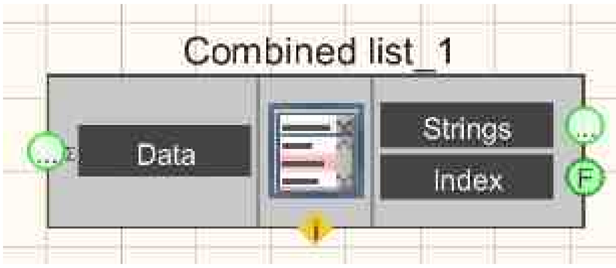
Project operation result



8.5.Combined list

The component "**Combined list**" is designed to select and set a specific date and subsequently use it in projects.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Data - strings displayed in the list.

➢ Output:

- Strings - the selected string of the list is output.
- Index - to output the index selected in the list.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Data - list data (written with a semicolon separator).
- Type (Strings) - selection of the type of output values:
- 1. Strings.
- 2. Numerical values.
- 3. Indexes.
- CurrentPosition (0) - set the current element of the list.
- Enabled (True) - set the accessibility of the control (enable/disable the component).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Data - string value displayed in the list (any type).

➢**Custom common properties:**

- [Are available by the link below.](154)

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR Data - Set the list data (written with a semicolon separator) (string).
- BSTR Type - Set the format of output values: Strings; Numerical values; Indexes (string).
- LONG CurrentPosition - Set the current element of the list (any number).

VARIANT_BOOL Enable - Set the accessibility of a control:

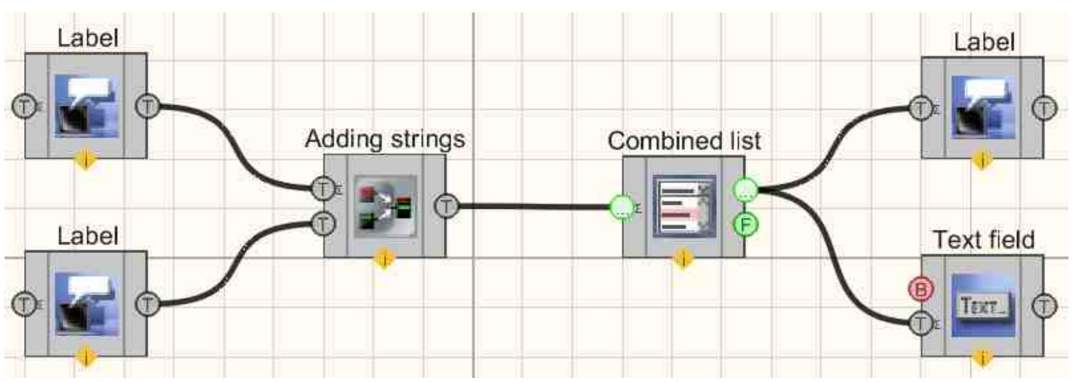
- true - the output value is emitted when the control's accessibility is enabled;
  false - the no output value is emitted when the control's accessibility is enabled.

➢**Methods:**

- void SetPosition(LONG position) - Set the number of the current element from the combo box.
- BSTR GetElement(void) - Returns the currently selected element.

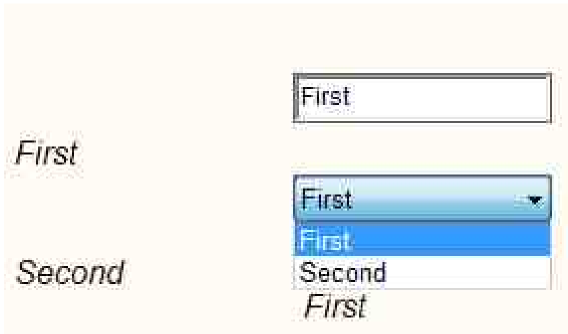
Example

Project in SCADA ZETView



This project shows the operation of the Combined list 414 component is used to selectively pass text from one of the labels to the Text field 1100. The Adding strings 1094 component is needed to concatenate two channels with text into one common one. It should be noted that in the private properties of Adding strings 1094, the delimiter symbol (in this case) is set, which is necessary to pass concatenated strings to the Combined list 414 as 2 strings.

Project operation result

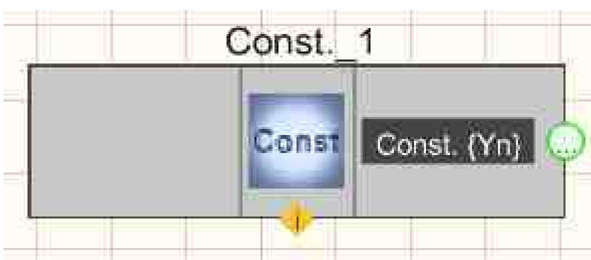


View the example in ZetView

8.6.Constant

The component "**Constant**" is designed to select and set a certain constant value, which is constantly transmitted to the channel.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢ Output:

- Constant (Yn) - the set constant value is received.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Value (0) - Set the value of a float type constant.
- Value Bool (False) - Set the value of a constant of type bool.
- ValueLong (0) - Set the value of a constant of type long.
- ValueDouble (0) - Set the value of a constant of type double.
- ValueType (float) - Set the constant value type: float; bool; long; double.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

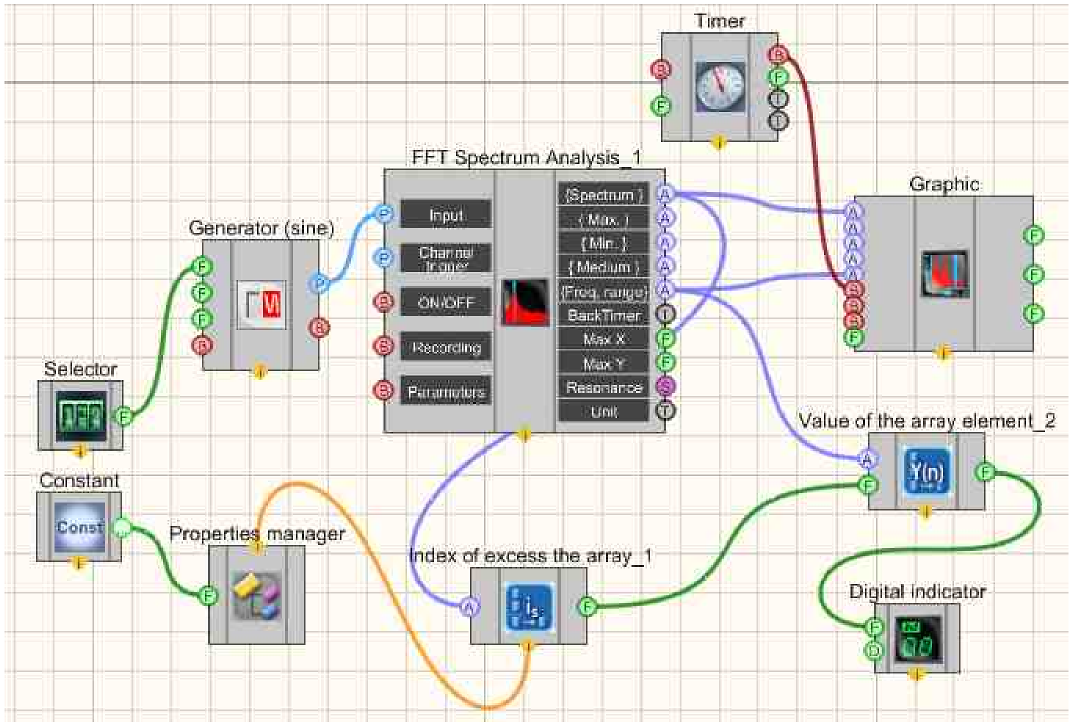
➢**Custom private properties (the default value is shown in parentheses):**

•FLOAT Value - set the value of a float type constant.
•VARIANT_BOOL ValueBool - set the value of a constant of type bool.
•LONG ValueLong - set the value of a constant of type long.
•DOUBLE ValueDouble - set the value of a constant of type double.
•BSTR ValueType - set the constant value type: float; bool; long; double.

Example

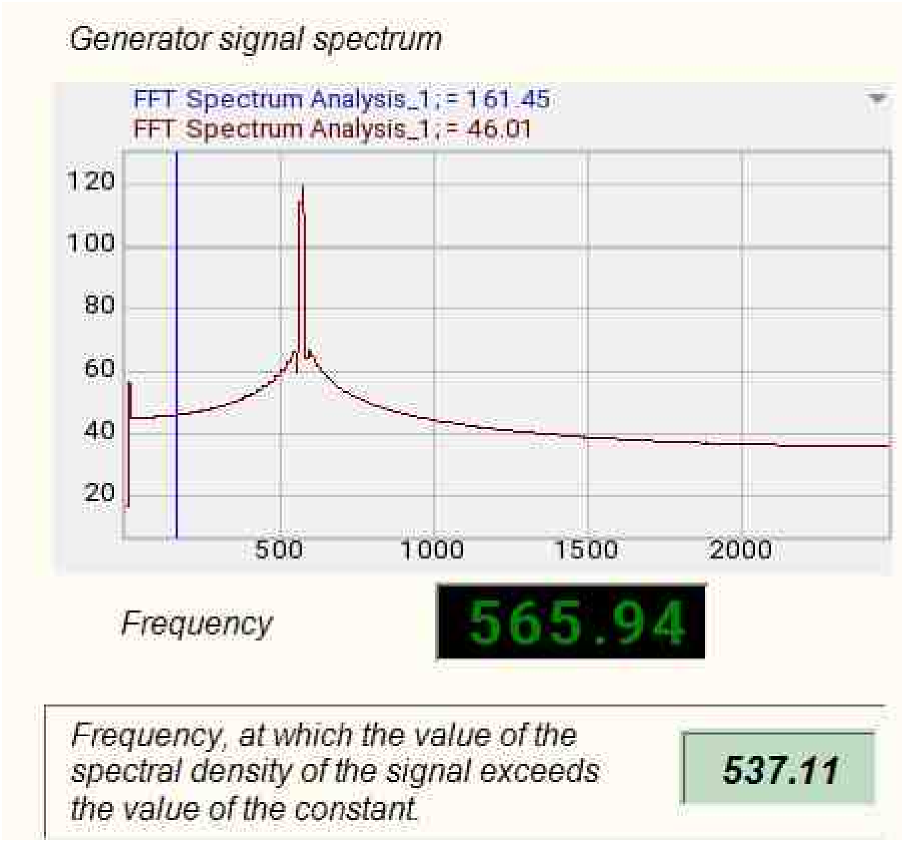
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



This project shows the operation of the Constant[417] component is used to set the value of the level with which all values of the spectral density of the signal at different frequencies will be compared. If this level is exceeded, the value of the frequency at which this occurred will be displayed on the Digital Indicator[682]. A Sine signal[530] is needed to generate a harmonic signal. The FFT Spectrum Analysis[260] is used to obtain the signal spectrum. The Graphic[909] is necessary to display the dependence of the spectral density on the frequency on the display. The Index of excess the array[807] is needed to find the frequency at which the spectral density value will be exceeded, the Value of the array element[803] is necessary in order to get a specific frequency value from the array at which the excess will be performed.

Project operation result

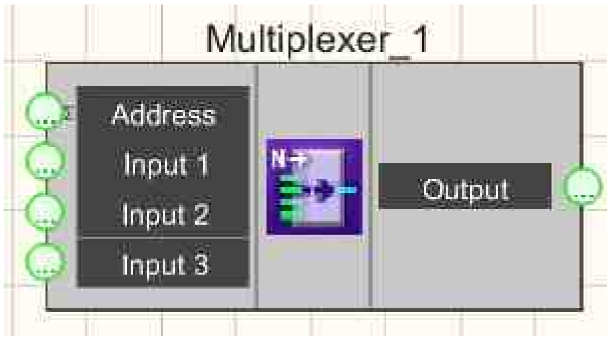


View the example in ZetView

8.7.Multiplexer

The component "**Multiplexer**" is designed to send a signal from one of the inputs to the output, while the desired input is selected using the address.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Address - the address of the input to which the output will be connected is received.
- Input 1 - the input channel is connected if the address is selected with using Input 1.
-
- ...
- Input N - the input channel is connected if the address is selected with using Input N.

➤ Output:
- Output - the channel that will be switched in the future.

Custom common properties:

➤Total properties (environment):
- Are available by the link below.154

➤Custom private properties (the default value is shown in parentheses):
- Input (1) - set the number of input channels.
- Status (1) - the current connection of the multiplexer.

Note;

The multiplexer will only transmit data further if it has the "m_sHelpString" field filled in.Without specifying "m_sHelpString", the Graphic909 component will not display an array passed through the multiplexer.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Address - the address of the input to which the output will be connected is received (any type).
- Input 1 - the input channel is connected if the address is selected with using Input 1 (any type).
- ...
- Input N - the input channel is connected if the address is selected with using Input N (any type).

➢**Custom common properties:**

- Are available by the link below. 154

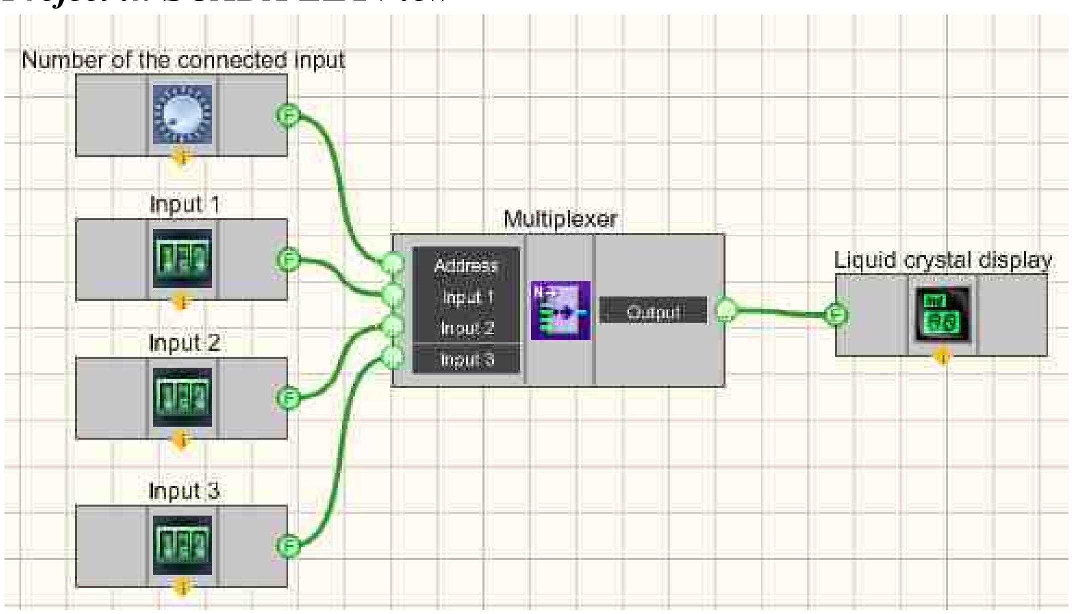
➢**Custom private properties (the default value is shown in parentheses):**

•LONG Input - Set the number of inputs (from 0 to 50).

•LONG Status - Set the current status of the multiplexer (any number).

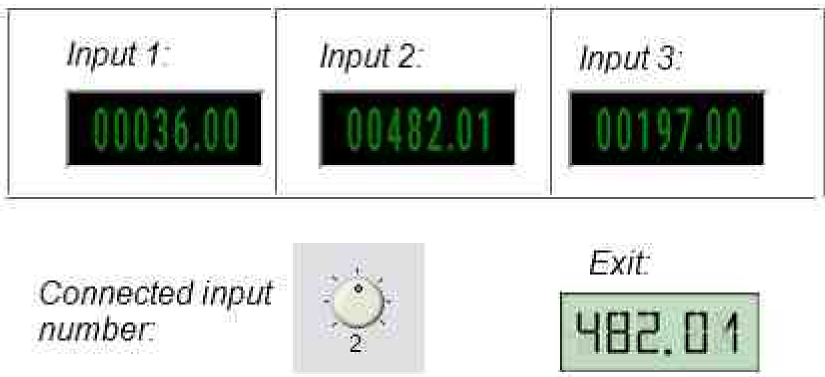
Example No. 1

Project in SCADA ZETView



This project shows the operation of the Multiplexer 420 component is used to connect a channel with a Selector No. 3 441 (a high or low logical level is set in this channel, depending on from the status of pressing the selector). The Selector 435 allows us to choose which output channel we will connect the channel with the button to.

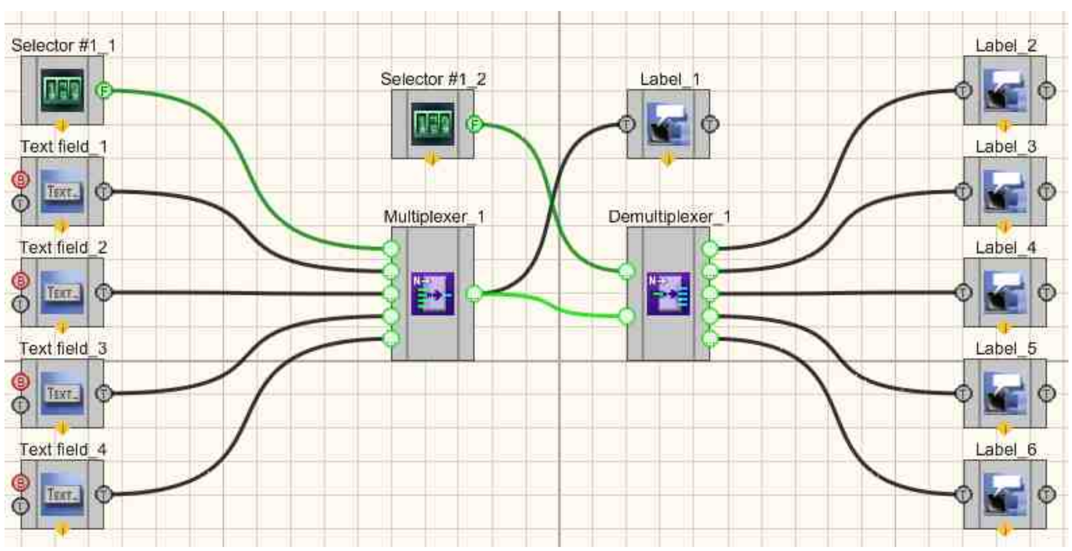
Project operation result



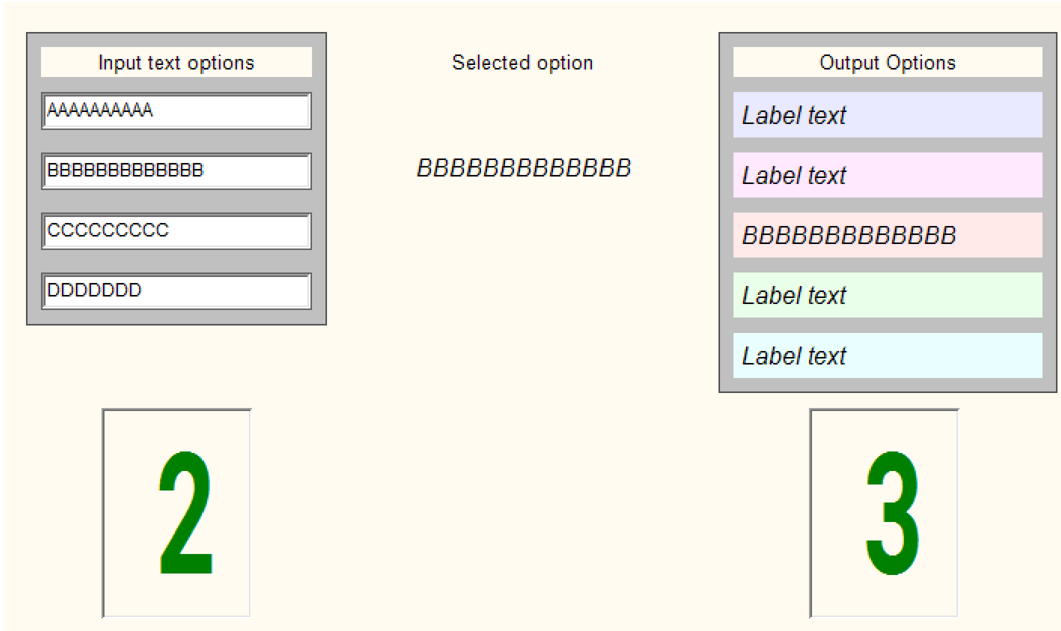
View the example in ZetView

Example No. 2

Project in SCADA ZETView



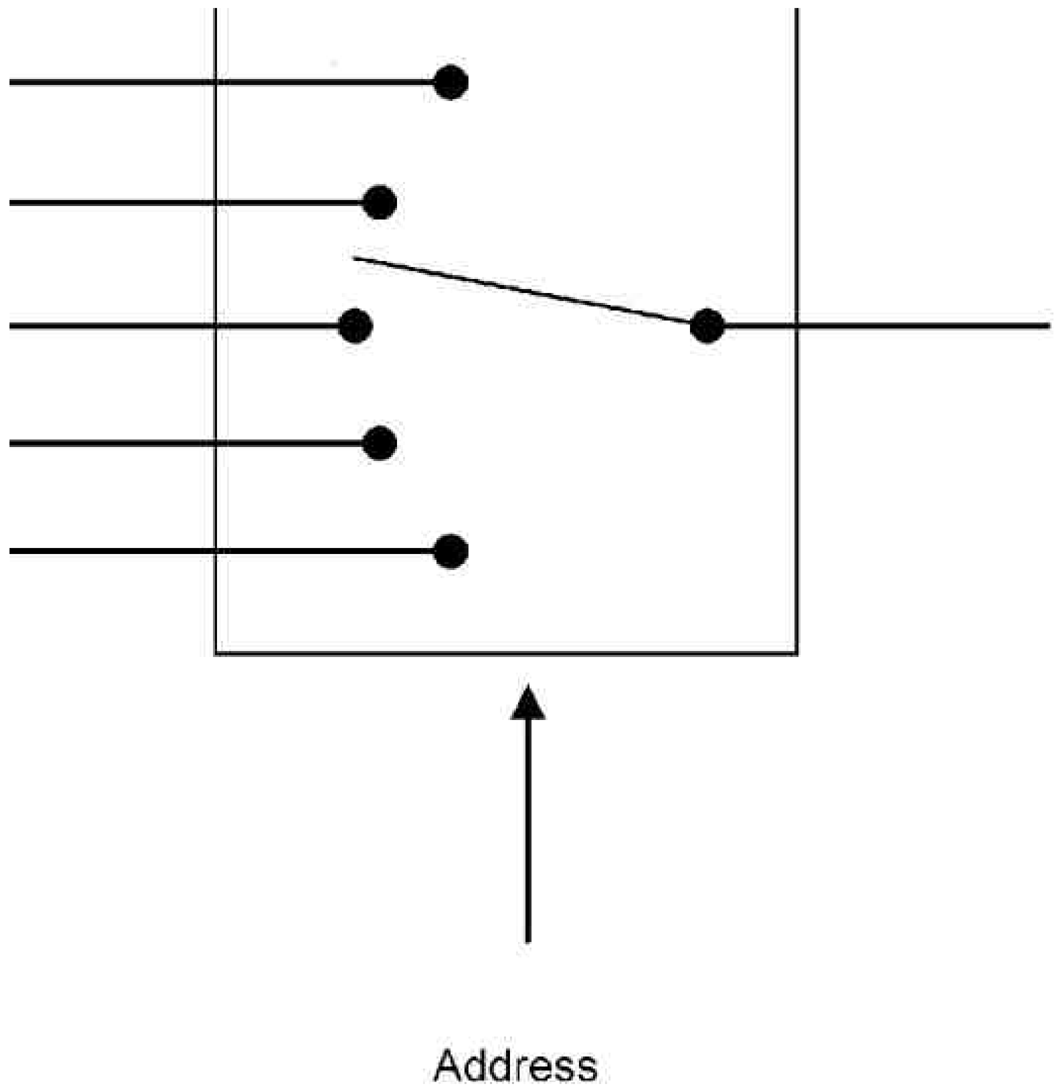
Project operation result



View the example in ZetView

Mathematical description

A multiplexer is a device that connects one of the n input signals to a single output line. The choice of the connected input is carried out using the address transmitted via special lines. Multiplexers are sometimes called data selectors.



If we compare the principles of operation of analog and digital multiplexers, then they are fundamentally different from each other. In analog, an electrical contact is formed between the input and output. In digital, the value of the input signal is copied to the output of the circuit.

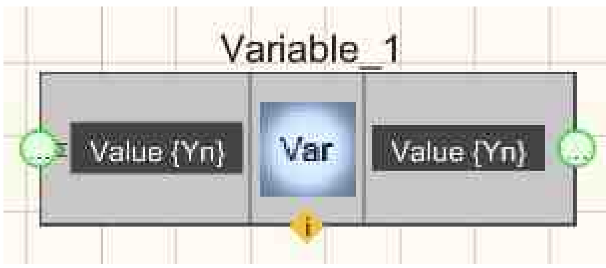
Structurally, multiplexers consist of a commutator, which provides connection of inputs with one output. The switch is controlled by address and sometimes enabling (strobe) inputs.

In digital multiplexers, the logic elements of the switch and decoder are usually combined.

8.8.Variable

The component "**Variable**" is designed to transfer a variable value to the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Value (Yn) - input value.

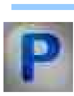
➢ Output:
- Value (Yn) - output value.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- ValueFloat (0) - Set the value of a float type constant.
- Value Bool (False) - Set the value of a constant of type bool.
- ValueLong (0) - Set the value of a constant of type long.
- ValueDouble (0) - Set the value of a constant of type double.



Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Value (Yn) - input value (any number).

➢**Custom common properties:**

- Are available by the link below. 154

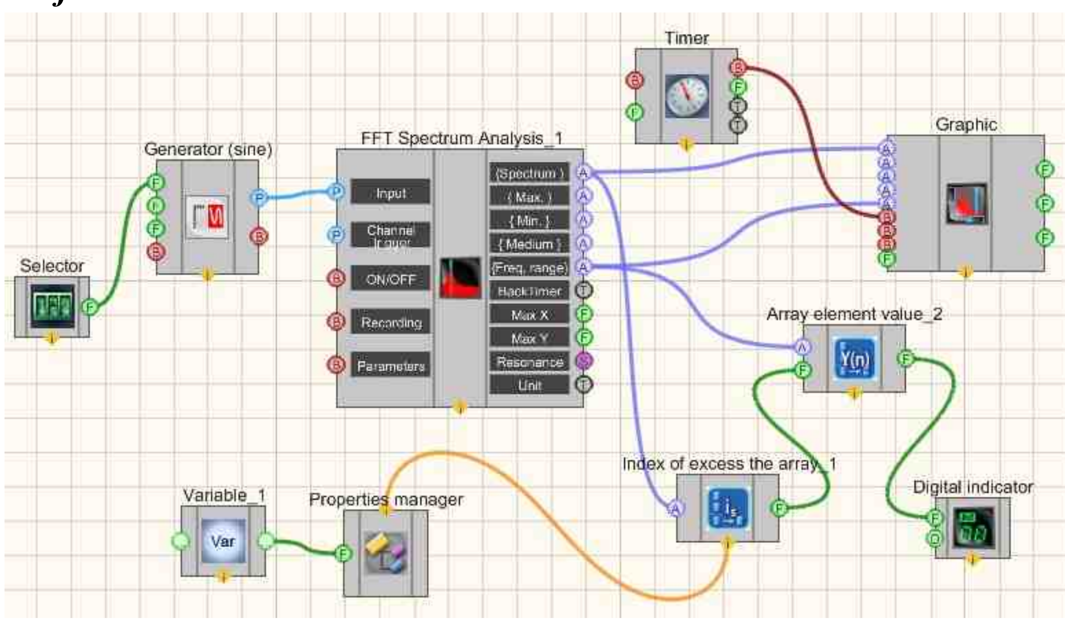
➢**Custom private properties (the default value is shown in parentheses):**

•FLOAT ValueFloat - Set the current float value (any number).
•VARIANT_BOOL ValueBool - Set the current bool value (0 - no, off / 1 - yes, on, true).
•LONG ValueLong - Set the current value of type long (any number).
•DOUBLE ValueDouble - Set the current value of the double type (any number).

Example

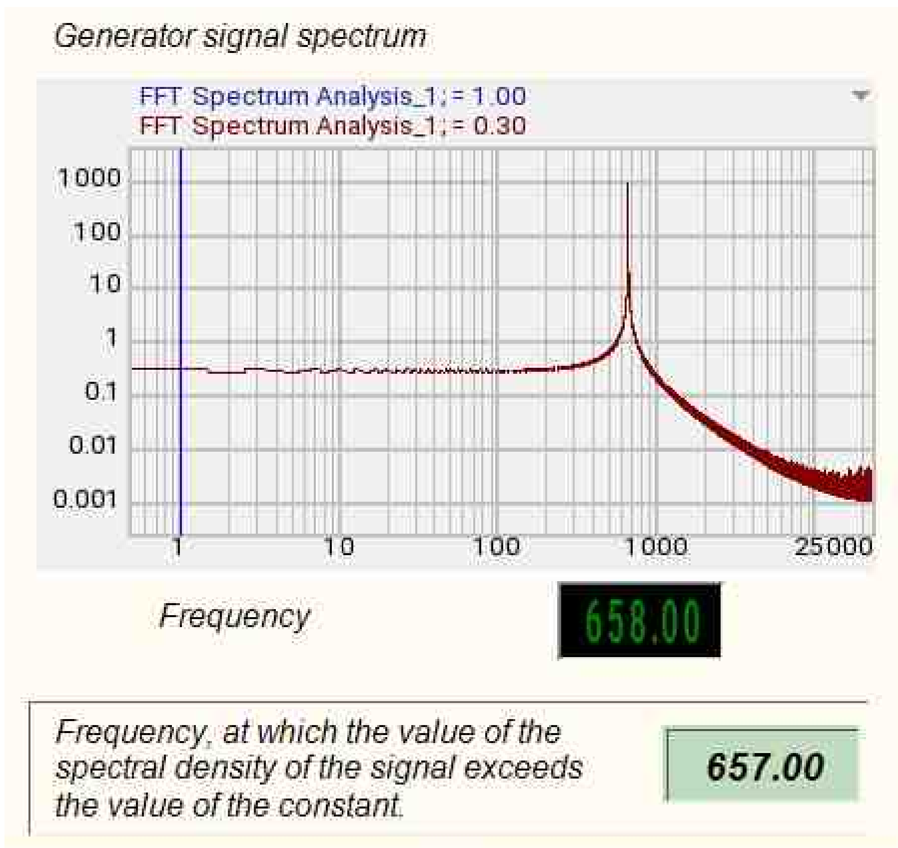
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



This project shows the operation of the Variable 426 component is used to set the level value against which all values of the signal spectral density at different frequencies will be compared. If this level is exceeded, then the value of the frequency at which this happened will be displayed on the Digital Indicator 682. A Sine signal 530 is needed to generate a harmonic signal. The FFT Spectrum Analysis 260 is used to obtain the spectrum of the signal. The Graphic 909 is needed to show the dependence of the spectral density on the frequency on the display. The Index of excess the array 807 is needed to find the frequency at which the value of the spectral density will be exceeded. The Value of the array element 803 is needed in order to get a specific frequency value from the array at which the overshoot occurs.

Project operation result



View the example in ZetView

8.9.Selector No. 1

The component "**Selector No. 1**" is designed to select numerical values and transfer them to the channel.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

Doesn't have.

➢ Output:

- Number (Yn) - output value from which comes the number entered by the user on the selector.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Min (0) - the minimum possible set value.
- Max (1000.000000) - the maximum possible value to be set.
- NumCount (5) - the number of displayed integer characters. The default is 5. Whatever the value of the Min and Max parameters, if the number of characters is not enough, the user will be able to enter a number as close as possible to the specified one, but limited by the number of characters. in Example, if Max=1000 and NumCount=3, the maximum number that can be entered from such a selector is 999.
- NumDivCount (2) - the number of decimal places to display.
- Value (1) - the current value set on the selector is saved after the project is closed. ZETView notifies the user if the parameters are changed and offers to save the data or cancel the saving. If you save the project, then the current values on the selectors will also be saved.
- Color (008000) - text color. The default is green.
- FonColor (000000) - background color. The default is black.
- Status (true) - enable/disable the selector.
- Activate (true) - Operation status. If value = false, the user will not be able to interact with the selector.

Description of the component

To start working with the selector, you need to run the project for execution. To enter data from the selector, you need to click 1 time with the left mouse button on the desired category and scroll the mouse wheel.

The second method of using this component:

1) Double click on the component, a window for entering the value will appear:

2) Next, enter the value and press the Enter button

The second method of using the Selector component is more correct. First, it is easier to enter the required value in this way. Secondly, if you need to send 2 identical values in a row from the selector, you need to use this method, since the selector transfers data to the channel only when the value changes. That is, if the number "1" is set on the selector, it will pass its channel only once, in order for the component to pass the value again, the value needs to be changed.
To enter two identical values in a row, just double-click on the selector and press Enter. Then the selector will again pass the last value (See Example1). A similar input method may be necessary when working with some components, in Example, such as an Excel Cell 1289.

The selector is a very common and frequently used component in ZETView. It is used as a universal means of entering numbers.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

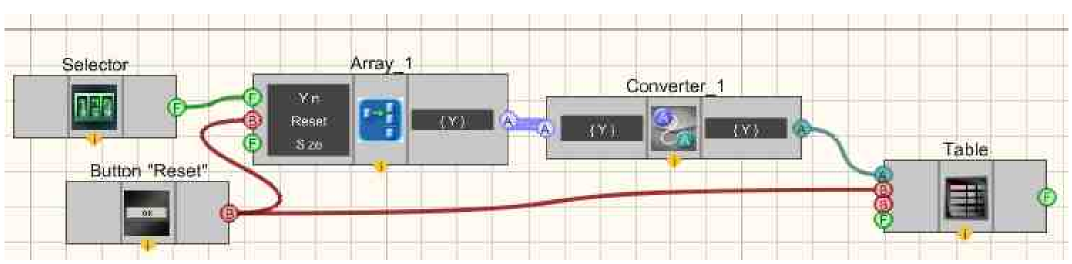
- FLOAT Min - Set the minimum value (any number).
- FLOAT Max - Set the maximum value (any number).
- LONG NumCount - Set the number of displayed integer characters (any number).
- LONG NumDivCount - Set the number of decimal places to display after the decimal point (from 0 to 6).
- FLOAT Value - Set the current value (any number).
- LONG Color - Set the text color (any number).
- LONG FonColor - Set the background color (any number).
- VARIANT_BOOL Status - Set the status:
  - true - the status is turned ON;
  - false - the status is turned OFF.
- VARIANT_BOOL Activate - Set the operation status of selector No.1:
  - true - selector #1 operation status is turned ON;
  - false - selector #1 operation status is turned OFF.

➢Methods:

- void Increment (FLOAT value) - Set the value of the increment (any value).

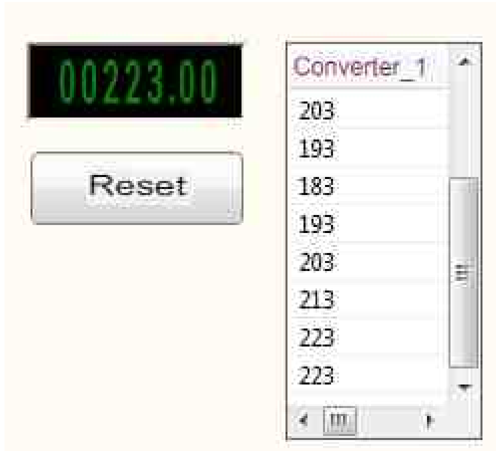
Example No. 1

Project in SCADA ZETView



In this Example, the user enters numbers from the Selector 428. Then they go to the Formation of arrays (numeric - array) 796. Next, the numeric array goes to the Converter (string array - numeric array) 704, where it is converted into a string array and output to the Data table 934. The Non-fixed button 694 "Reset" clears the table and resets the values on the Formation of arrays (numeric - array) 796.

Project operation result

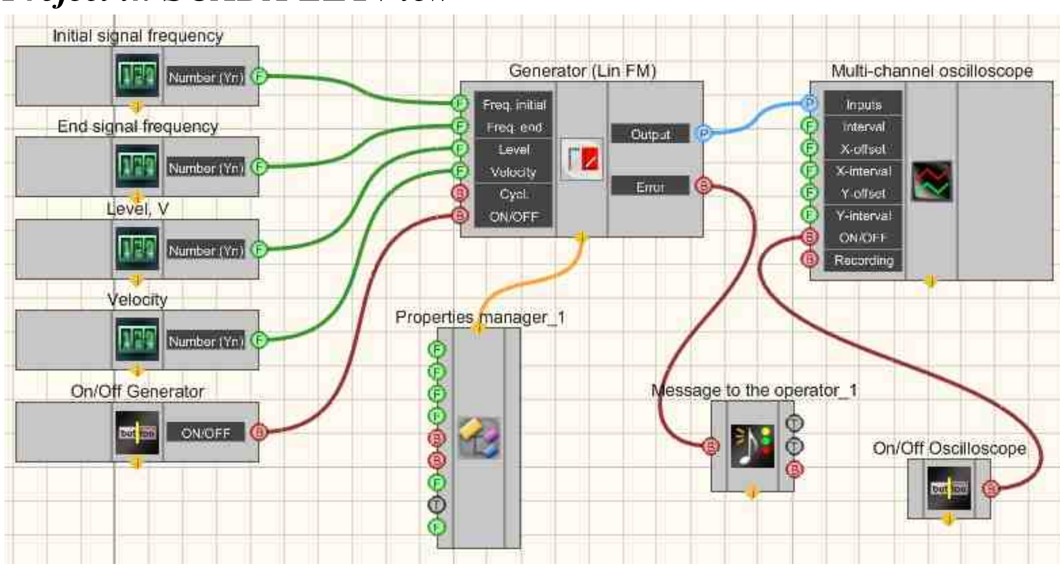


View the example in ZetView

Example No. 2

For the Examples to work, it is necessary to have a connected physical ZET device.

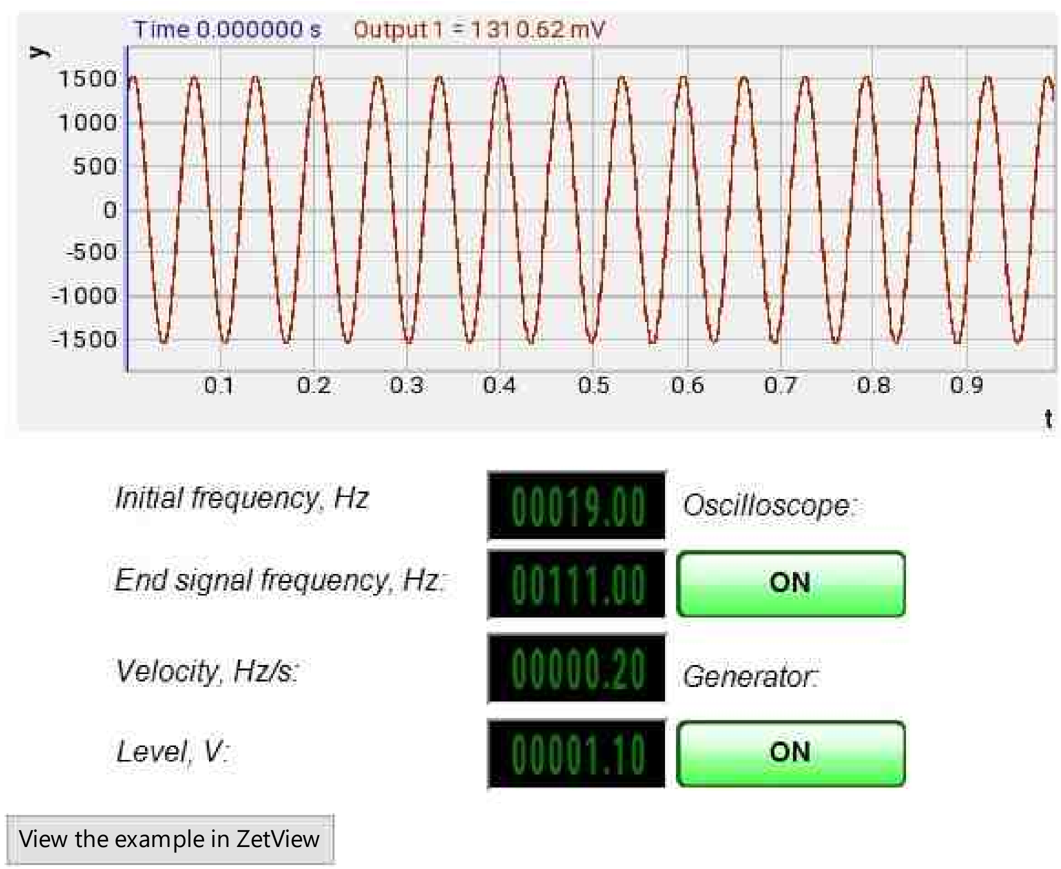
Project in SCADA ZETView



This project shows the operation of the [Linear frequency modulation (LinFM)](486) component generates a linear-frequency-modulated signal, the parameters of which are set using [Selectors](428) (initial signal frequency, end signal frequency, level,

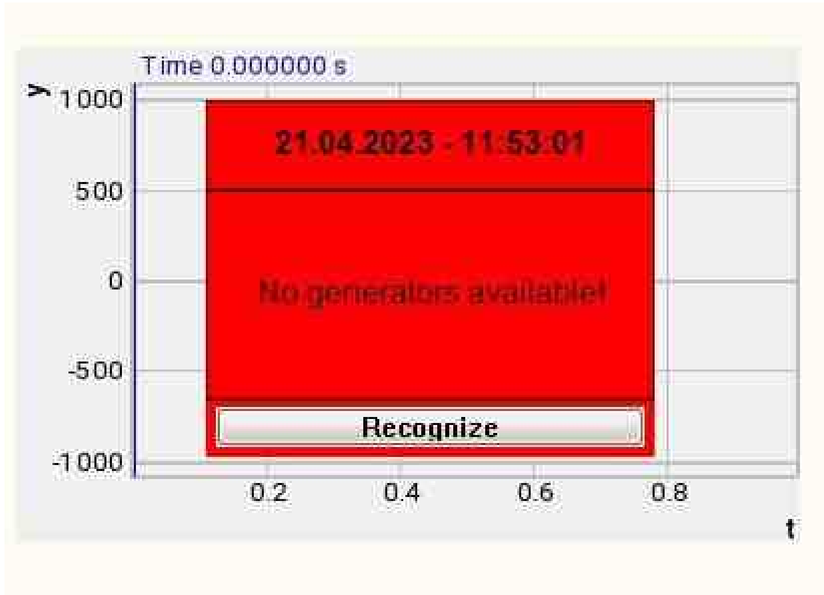
velocity). The buttons allow you to turn on and off the Sine signal 530 and the Multi-channel oscilloscope 928. The oscilloscope is used to view the end form of the signal. The Fixed button 441 is used to turn ON or OFF.
The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a Multi-channel oscilloscope.

Project operation results



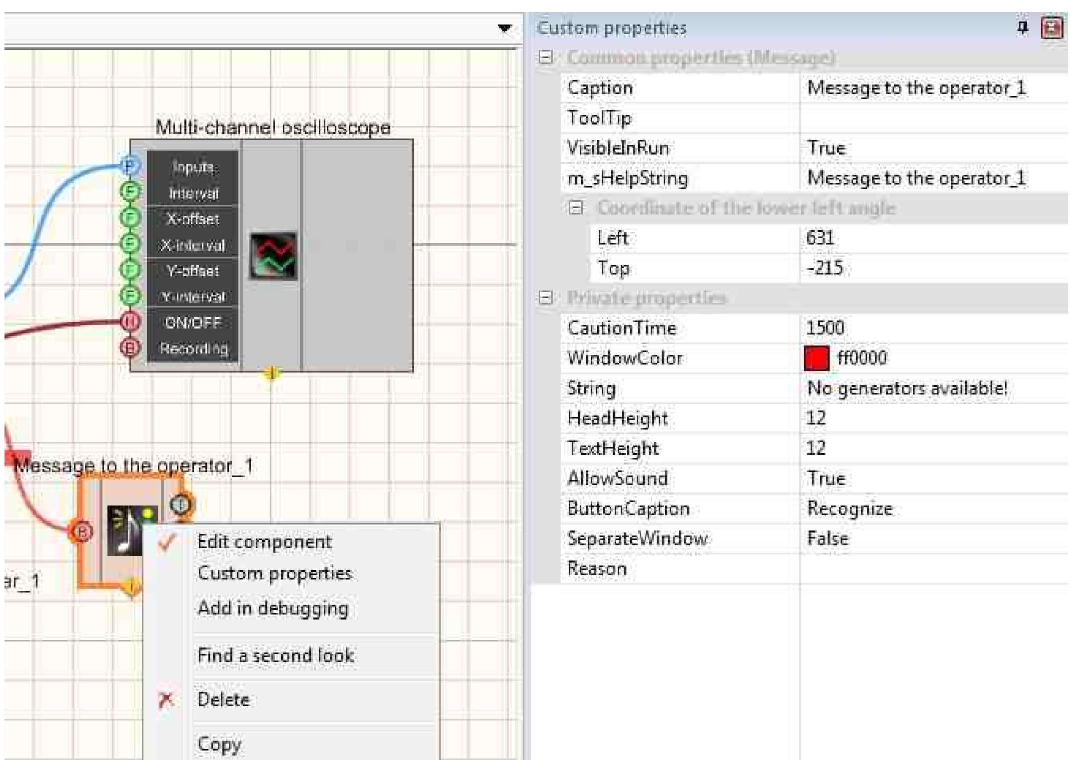
Output about an error in the project for Example No. 2

The component Message to the operator 1052 is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal 530. In addition, the message Sine signal No. 1 530 connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

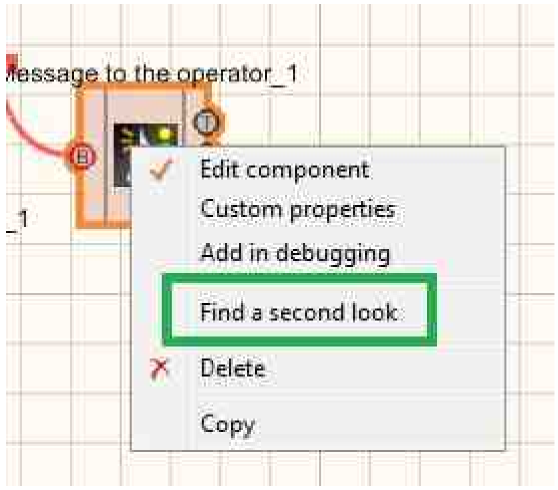


Message about the error to the operator.

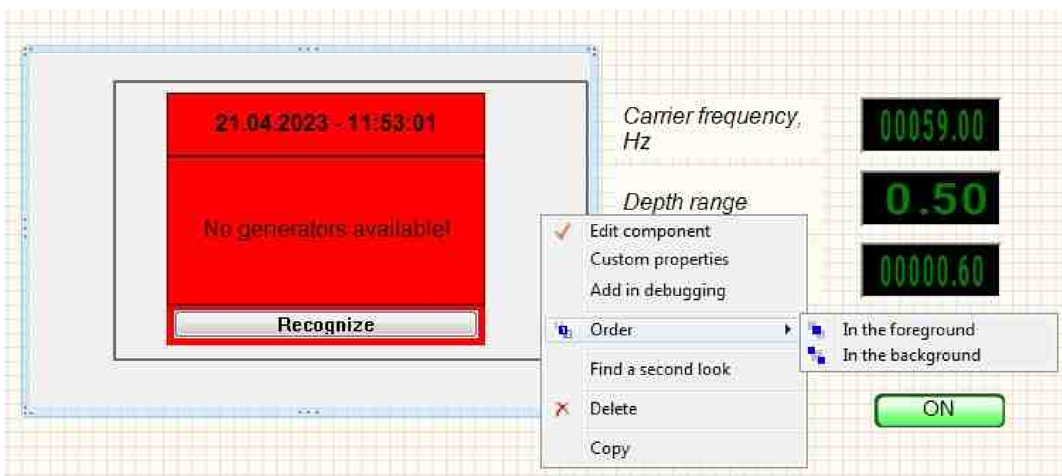
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



8.10.Selector No. 2

The component "**Selector No. 2**" is designed to select numerical values and transfer them to the channel. The selector is made in the form of a rotating regulator. The values that can be selected on this selector are determined by its settings: the number of divisions and step.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢ Output:

- Number (Yn) - the output value of the selector, which is specified by the user.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Value (1) - Current value.
- Status (true) - enable/disable the selector.
- Min (0) - the minimum possible set value.
- Max (9) - maximum possible set value.
- Fix (false) - set the switching mode:
  1. False - is used to switched over the entire range of the scale.
  2. True - is used to switched only between integer values.
- FontName (Arial) - the font of numbers on the scale.
- FontSize (14) - the font size.
- Mark (10) - the number of main divisions of the scale.
- Line (5) - the number of additional scale divisions.
- Point (false) - set the additional intermediate marks.
- Unit (s) - set the units of measurement.
- FonColor (f0f0f0) - set the background color.
- Color (cccccc) - the color of the internal area of the scale.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•FLOAT Value - Set the current value (any number).
VARIANT_BOOL Status - Set the status:
- true - the status is turned ON;
  false - the status is turned OFF.

•FLOAT Min - Set the minimum value (any number).
•FLOAT Max - Set the maximum value (any number).
VARIANT_BOOL Fix - Set the fixing:
- true - fixation is turned ON;
  false - fixation is turned OFF.

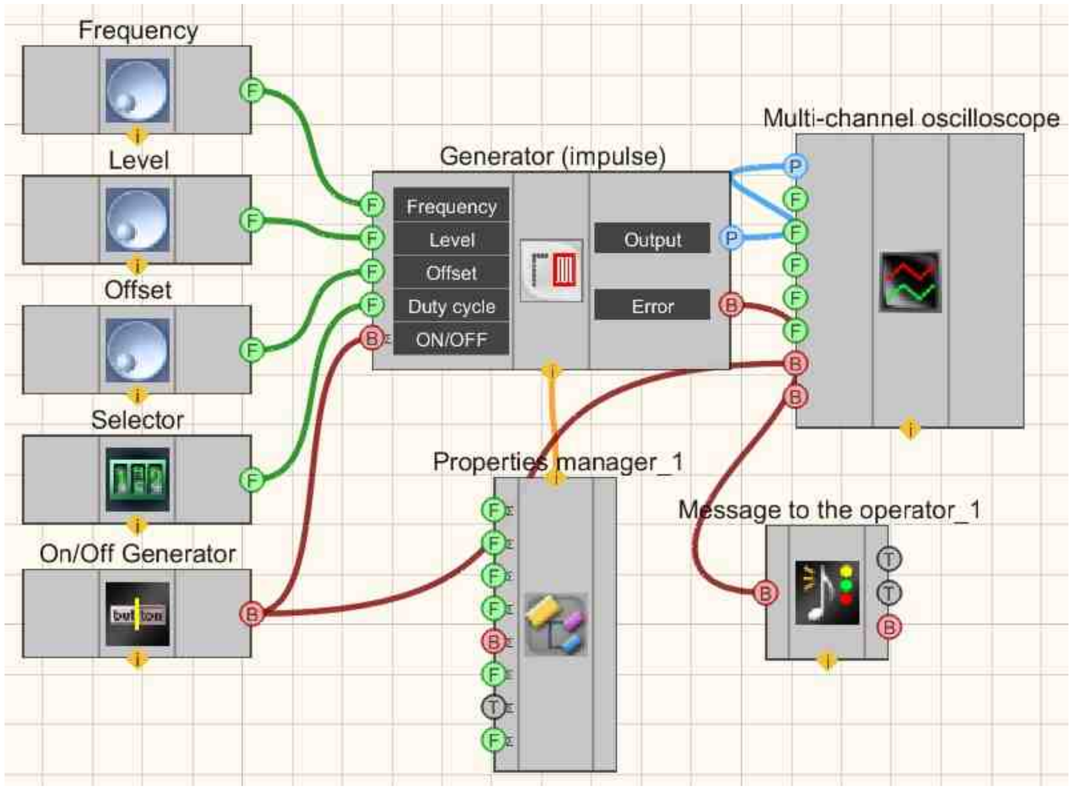
•BSTR FontName - Set the font name (string).
•LONG FontSize - Set the font size (any number).
•LONG Mark - Set the number of main divisions of the scale (any number).
•LONG Line - Set the number of additional scale divisions (any number).
VARIANT_BOOL Point - Set the small labels:
- true - small labels operation status is turned ON;
  false - small labels operation status is turned OFF.

•BSTR Unit - Set the unit of measure (string).
•LONG FonColor - Set the background color (any number).
•LONG Color - Set the text color (any number).
.

Example

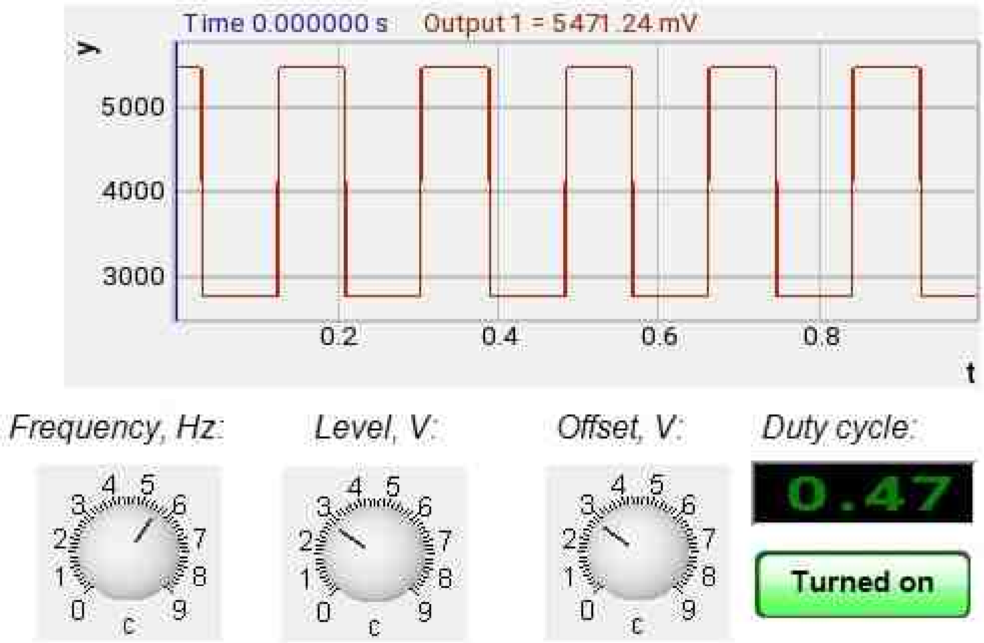
Project in SCADA ZETView

For the Examples to work, it is necessary to have a connected physical ZET device.



This project shows the operation of the Impulse signal 473 component generates rectangular impulses. Selectors 428 (as well as another type) are used to set the frequency, level, offset, duty cycle. The Fixed button 691 is required to enable/disable the Impulse signal 473. A Multi-channel oscilloscope 928 is used to graphically display the signal. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change indicators in real time and transfer data to the signal generator to display them on a Multi-channel oscilloscope 928.

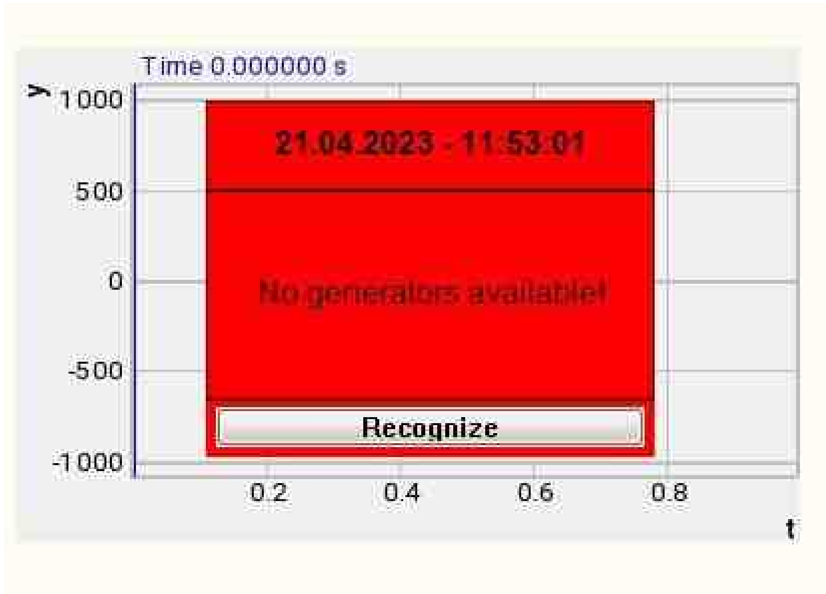
Project operation results



View the example in ZetView

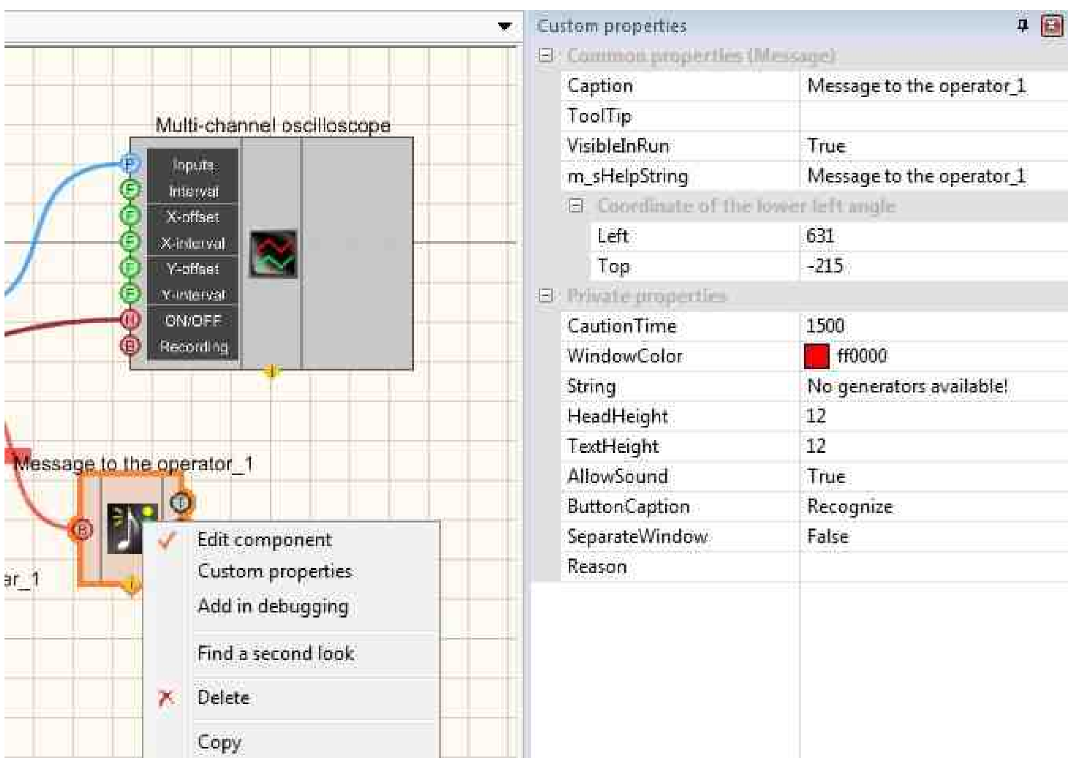
Output about an error in the project

The component [Message to the operator](1052) is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the [Sine signal](530). In addition, the message [Sine signal No. 1](530) connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

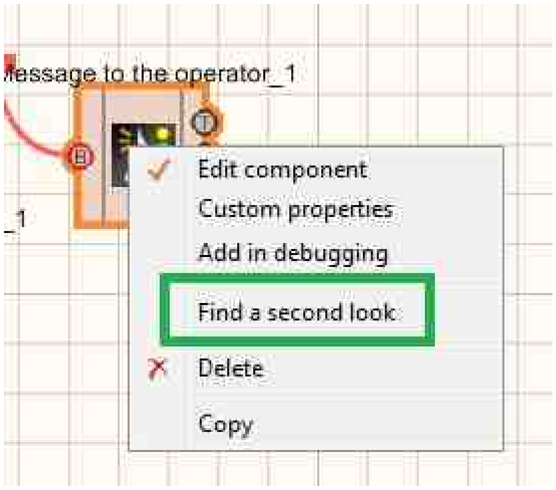


Message about the error to the operator.

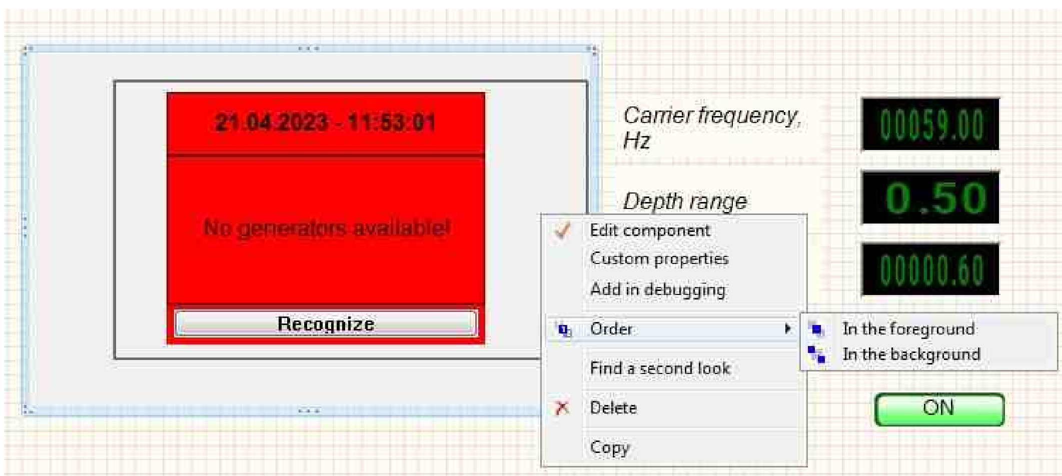
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



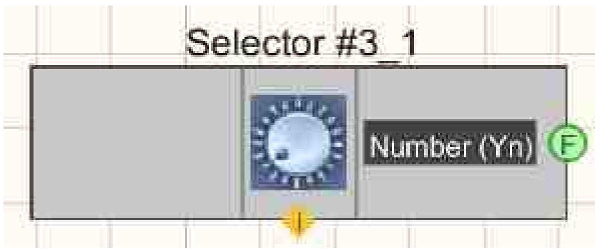
Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



8.11.Selector No. 3

The component "**Selector No. 3**" is designed to select numerical values and transfer them to the channel. The selector is made in the form of a rotating regulator. The values that can be selected on this selector are determined by its settings: the number of divisions and step.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢ Output:
- Number (Yn) - the output value that the user specifies.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Min (0) - the minimum possible set value.
- Max (10) - maximum possible set value.
- Status (true) - enable/disable the selector.
- Position (1) - the current value of the selector.
- UnitName (mV) - unit of measure.
- Scale (1) - value multiplier. Actually defines the step between scale divisions.
- BackColor (f0f0f0) - scale background color.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:
- Are available by the link below. 154

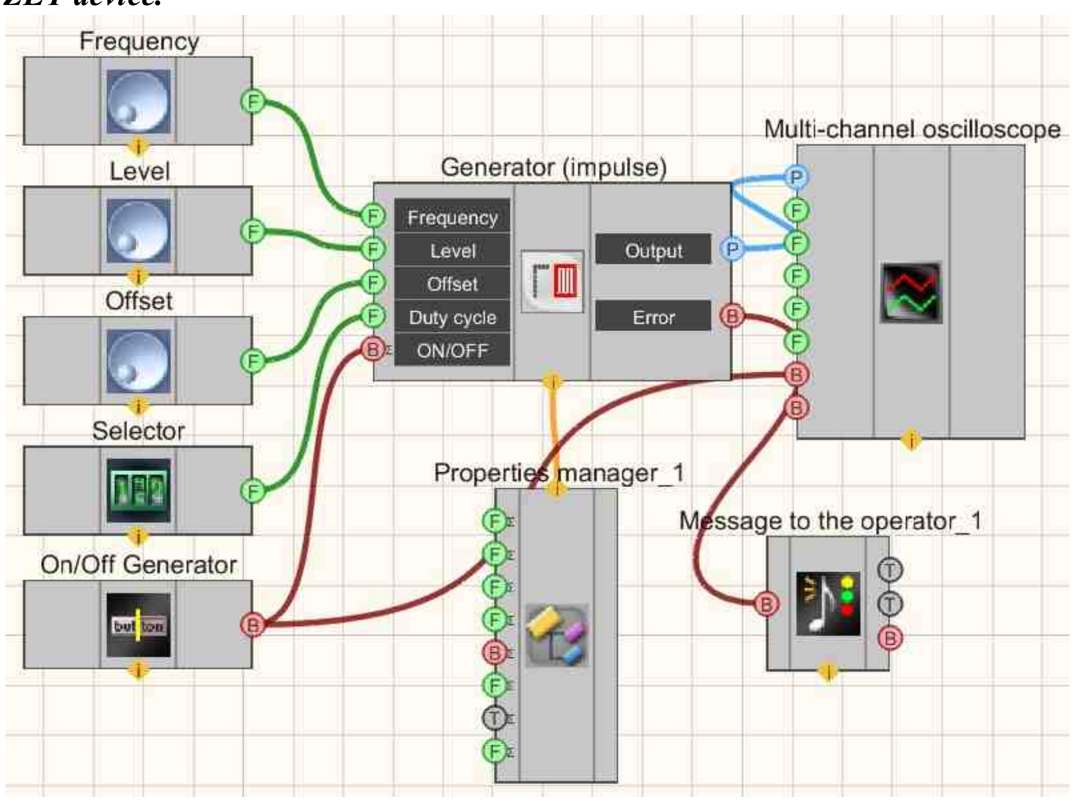
➢**Custom private properties (the default value is shown in parentheses):**

- LONG Min - Set the minimum value (any number).
- LONG Max - Set the maximum value (any number).
- VARIANT_BOOL Status - Set the status:
  - true - the status is turned ON;
  - false - the status is turned OFF.
- LONG Position - Set the current value of the selector (any number).
- BSTR Unit - Set the unit of measure (string).
- FLOAT Scale - Set the value multiplier (any number).
- LONG BackColor - Set the background color (any number).

Example

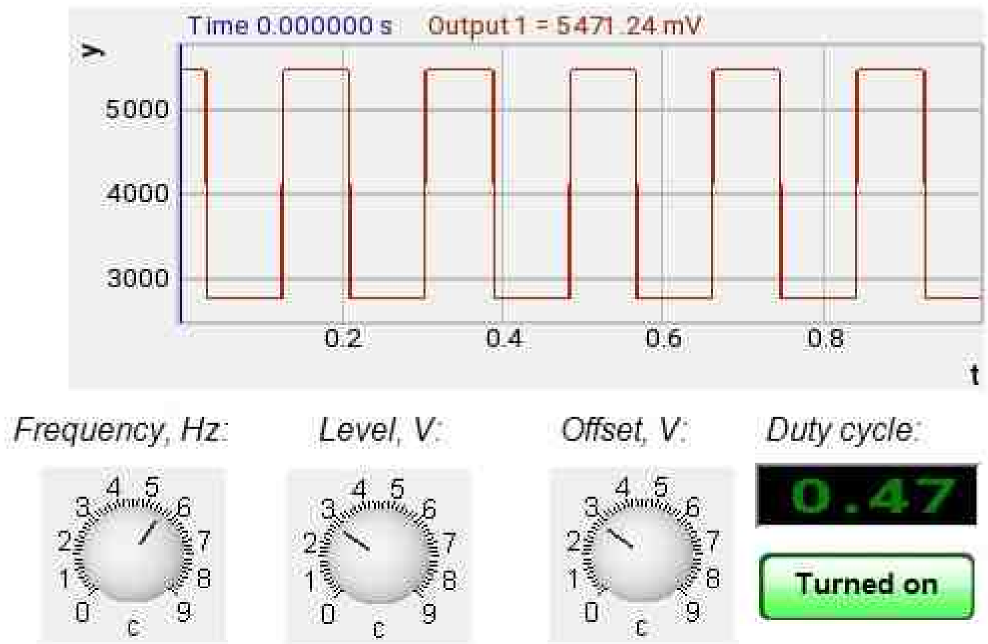
Project in SCADA ZETView

For the Examples to work, it is necessary to have a connected physical ZET device.



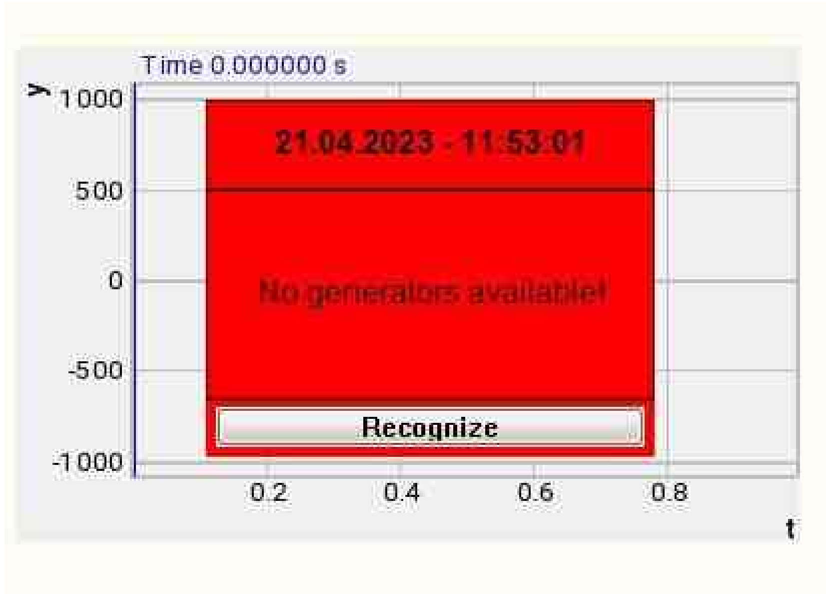
This project shows the operation of the Impulse signal[473] component generates rectangular impulses. Selectors[428] (as well as another type) are used to set the frequency, level, offset, duty cycle. The button is required to enable/disable the Impulse signal[473]. A Multi-channel oscilloscope[928] is used to graphically display the signal. The Fixed button[691] is used to turn on or off. The Property manager[999] is used to change indicators in real time and transfer data to the signal generator to display them on a multi-channel oscilloscope.

Project operation results



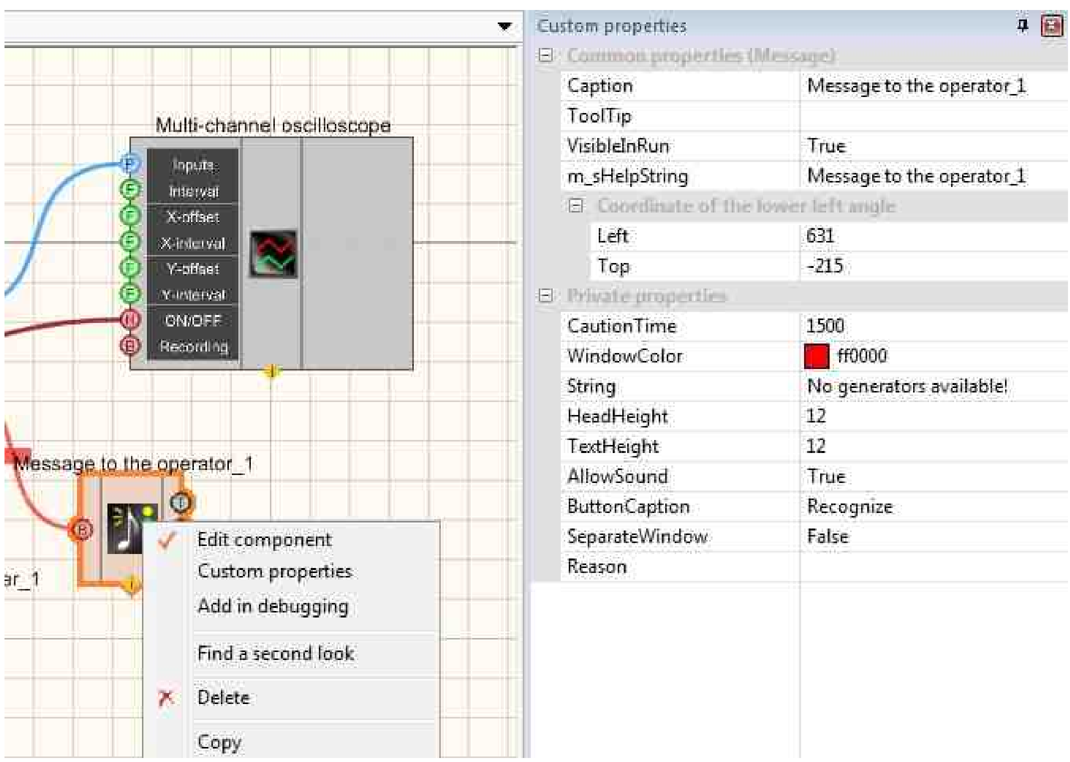
Output about an error in the project

The component Message to the operator[1052] is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal[530]. In addition, the message Sine signal No. 1[530] connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

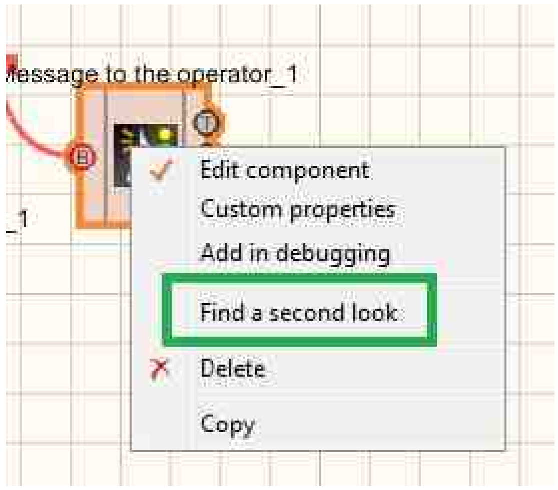


Message about the error to the operator.

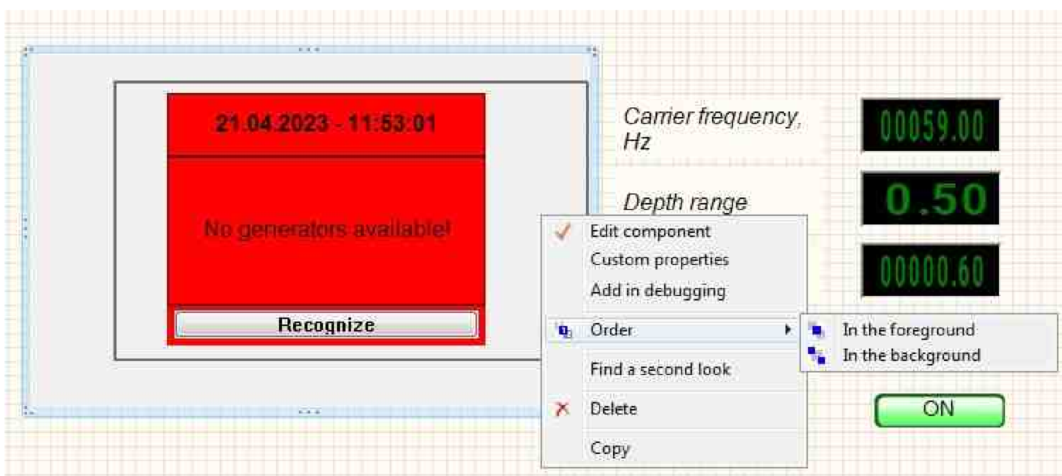
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



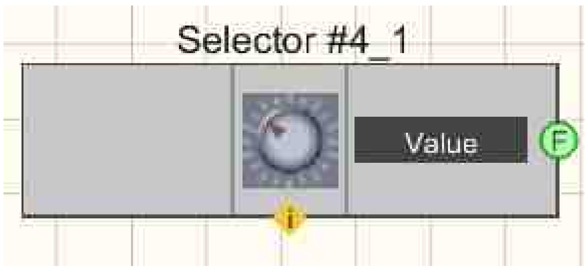
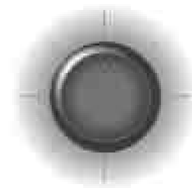
Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



8.12.Selector No. 4

The component "**Selector No. 4**" is designed to select numerical values. The selector is made in the form of an infinite rotating regulator. The values that can be selected on this selector are determined by its settings: color properties, change per one revolution, current value, rotation resolution.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢ Output:

- Value - at the output of the control signal, which is set by the user.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- GradientBackColor (373737) - Set the color of the selector.
- selectorform (Circle) - determines the form of the selector:
  1. Cylinder.
  2. Circle.
  3. Sphere.
- cursorform (Circle) - determines the form of the cursor:
  1. Triangle.
  2. Circle.
- CursorColor (005eff) - defines the cursor color (for the "Filled" style). The default is blue.
- cursorstyle (Gradient) - set the cursor style:
  1. Filled
  2. Gradient
- shade (Dark) - Set the shade:
  1. No.
  2. Dark.
  3. Light.
- LinearVisible (True) - set the visibility of the ruler (True/False).
- Linearstyle (Different lengths) - the style of the ruler is determined:
  1. One length
  2. Different length

- LinearColor (a0a0a4) - set the color of the ruler.
- ValueOnLap (10) - determines the change in value when turning 360 degrees.
- CurrentValue (0) - Set the Current value.
- StepMove (false) - determines the movement in steps (On/Off).
- ValueInStep (0.5) - step discreteness is set (cannot be more than ValueOnLap / 8).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•LONG GradientBackColor - Set the color of the selector (any number).
•BSTR SelectorForm - Set the determines the form of the selector: Cylinder; Circle; Sphere (string).
•BSTR CursorForm - Set the cursor form: Triangle; Circle (string).
•LONG CursorColor - Set cursor color ("for the "Filled" style") (any number).

•BSTR CursorStyle - Set the cursor style: Filled; Gradient(string).
•BSTR Shade - Set the shade: No; dark; Light (string).
VARIANT_BOOL LinearVisible - Set ruler visibility:
- true - the output value is given when the ruler visibility is turned ON;
  false - the output value is given when the ruler visibility is turned OFF.

•BSTR LinearStyle - Set the ruler style: One length; Different length (string).

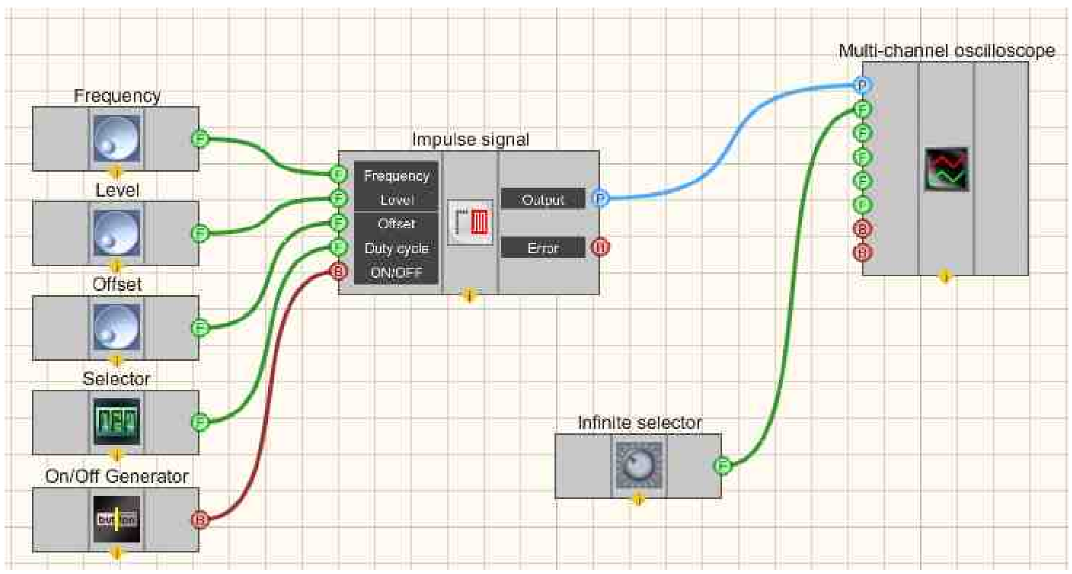
•LONG LinearColor - Set the color of the ruler (string).
•FLOAT ValueOnLap - Set the value to change when rotated 360 degrees (any number).
•FLOAT CurrentValue - Set the current value of the selector (any number).
VARIANT_BOOL StepMove - Set the step motion (On/Off):
- true - stepping is enabled;
  false - stepping is disabled.

•FLOAT ValueInStep - Set the step resolution (cannot be greater than ValueOnLap / 8).

Example

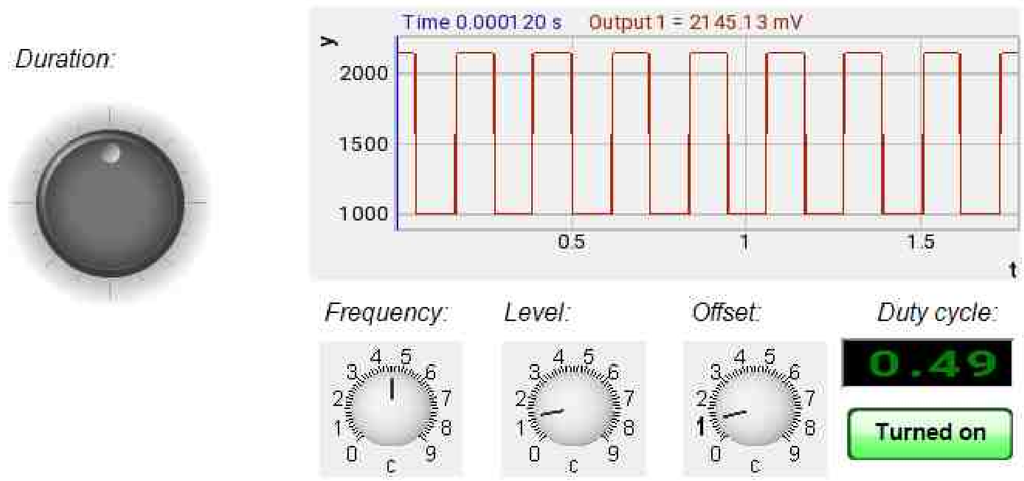
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



This project shows the operation of the Impulse signal 473 component creates a sawtooth signal, with parameters that are set using Selectors 428. To view the generated signal, use the Multi-channel oscilloscope 928 component. An additional Selector 428 is used to set the type of saw (Upward, downward, triangular). The Infinite selector 446 is used to change the span on a Multi-channel oscilloscope 928. The result of the compiled program can be seen in the operator interface.

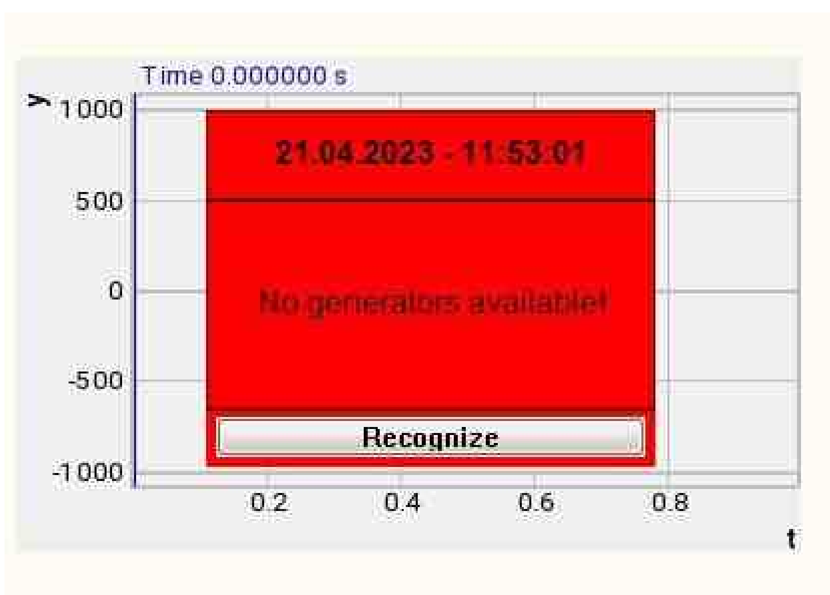
Project operation result



View the example in ZetView

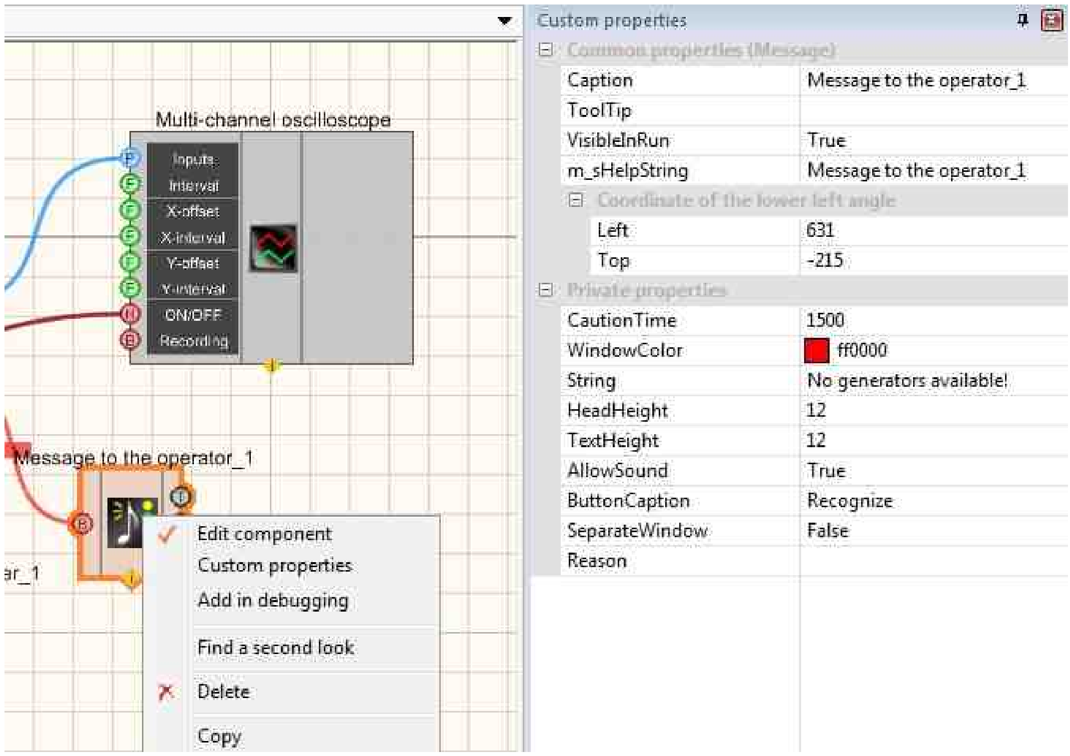
Output about an error in the project

The component [Message to the operator](1052) is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the [Sine signal](530). In addition, the message [Sine signal No. 1](530) connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

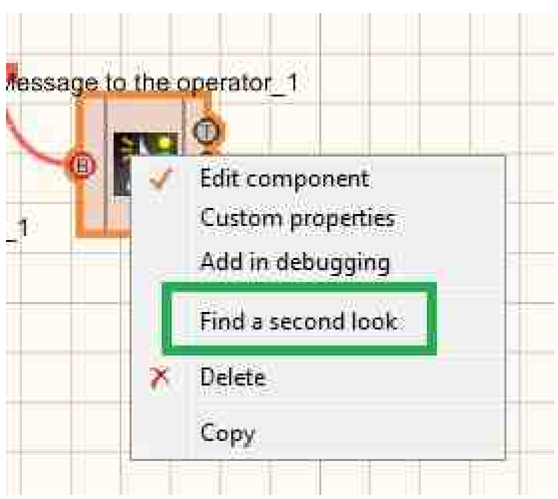


Message about the error to the operator.

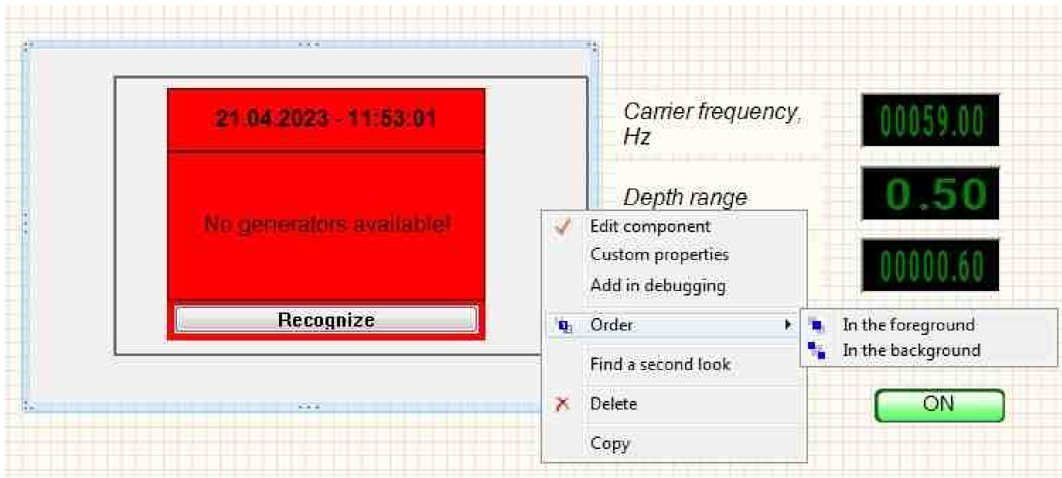
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



8.13.List

The component "**List**" is designed to provide a choice of the data stream by selecting it in the drop-down list.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Data - strings displayed in the list.

➢ Output:

- Strings - the selected line of the list is fed to the output.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Data - list data.
- Type (Strings) - selection of the list type:
  1. Strings.
  2. Numerical values.
  3. Indexes.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Data - strings displayed in the list (text).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

•BSTR Data - Set the list data (string).

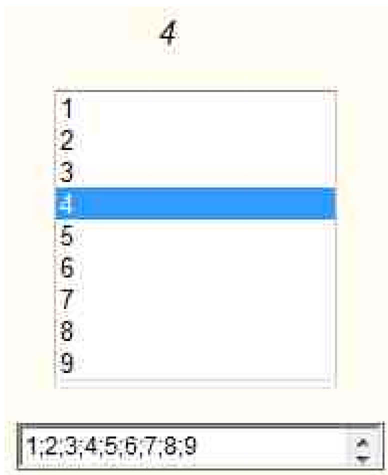
•BSTR Type - Set the list type: Strings; Numeric values; Indexes (string).

Example

Project in SCADA ZETView

This project shows the operation of the List[452] component is used for selective transmission of information from a Text field[1100].. The Label[1087] component is needed to display the selected part of the message on the display.

Project operation result



View the example in ZetView

Глава 9.Generator's DAC

The components of "Generators (DAC)" section are used a get signals at the output channels of ADC-DAC modules, FFT Spectrum Analysis, strain-gauge stations as well as to create virtual channels necessary for projects operation.

The components listed below are used to start and control "Signal generator" program from the scope of ZETLab software (each of the components corresponds to a tab of the program):

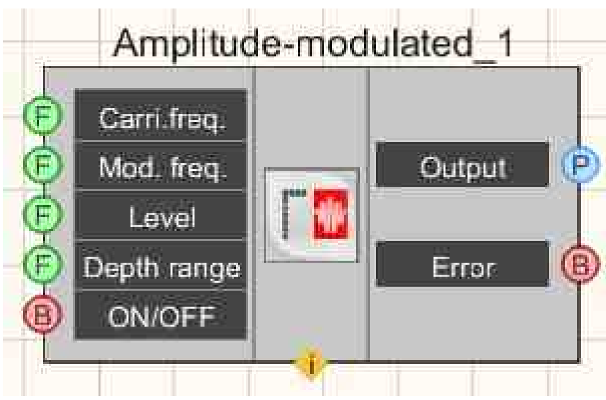
9.1.Amplitude-modulation

The component "**Amplitude-modulation**" is designed to generate a signal generated by a special frequency modulation law, at which the variable parameter of the carrier signal is its amplitude.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Carri.freq. - the carrier frequency of the signal, Hz.
- Mod.freq. - frequency of amplitude modulation, Hz.
- Level - the level of the amplitude modulation signal, V.
- Depth range - the amplitude modulation coefficient.
- ON\OFF - turn ON and OFF the generator.

➢Output

- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- CentralFrequency (1000) - the frequency of the carrier signal, Hz.
- ModFrequency (1) - the frequency of amplitude modulation, Hz.
- Level (1) - the amplitude of the output signal, V.
- Depth (0) - the depth range of the amplitude modulation.
- Activate (false) - the status of the signal (applied or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Carri.freq. - the value of the carrier frequency of the signal, Hz (number).
- Mod.freq. - the value of the amplitude modulation frequency, Hz (number).
- Level - the value of the amplitude modulation signal level, V (number).
- Depth range - the value of the amplitude modulation depth range (number).
-

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT CentralFrequency - Set the frequency of the carrier signal (from 0.01 to the maximum limit level of the DAC signal), Hz.
- FLOAT ModFrequency - Set the amplitude modulation frequency (from 0.01 to the maximum limit level of the DAC signal), Hz.
- FLOAT Level - Set the signal level (from 0 V to the maximum limit DAC signal level), V
- FLOAT Depth - Set the amplitude modulation depth (from 0 to 1).
- VARIANT_BOOL Activate - Set the signal presence status:

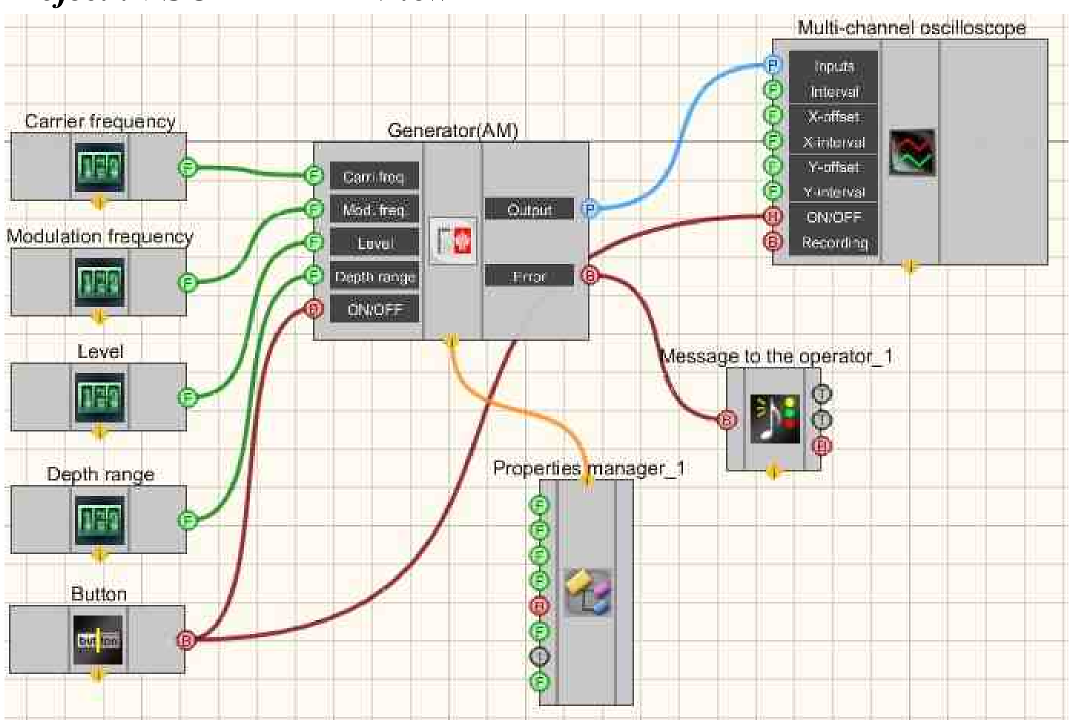
- true - the signal presence status is turned ON;
- false - the signal presence status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢Methods:

- void SetSignal(FLOAT CentralFrequency, FLOAT ModFrequency, FLOAT level, FLOAT Depth) - Set the carrier signal frequency, amplitude modulation frequency, signal level, amplitude modulation depth.
- void DeleteSignal(void) - Stop the generator.

Example

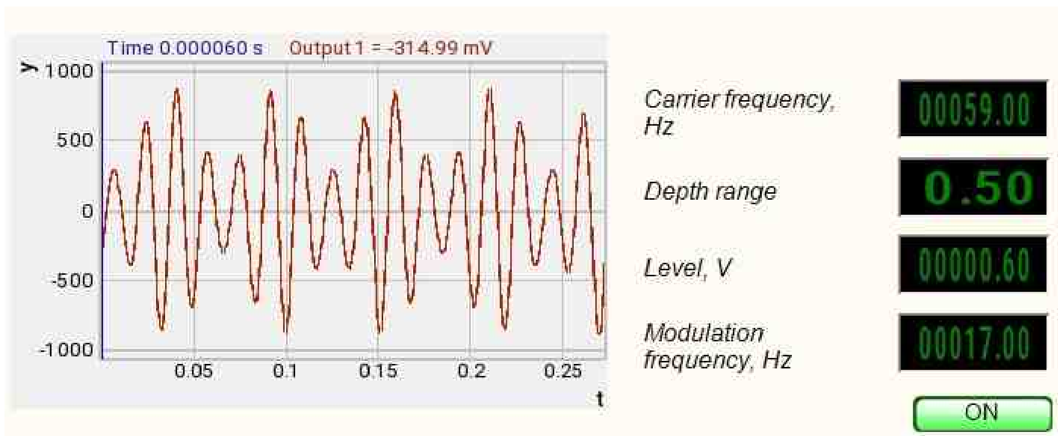
Project in SCADA ZETView



This project shows the operation of the Amplitude-modulation 455 component at the carrier frequency produces amplitude modulation of the signal. Selectors 428 are used to set the carrier frequency, level, depth and frequency of amplitude modulation. A

Multi-channel oscilloscope 928 is used to graphically display the signal. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

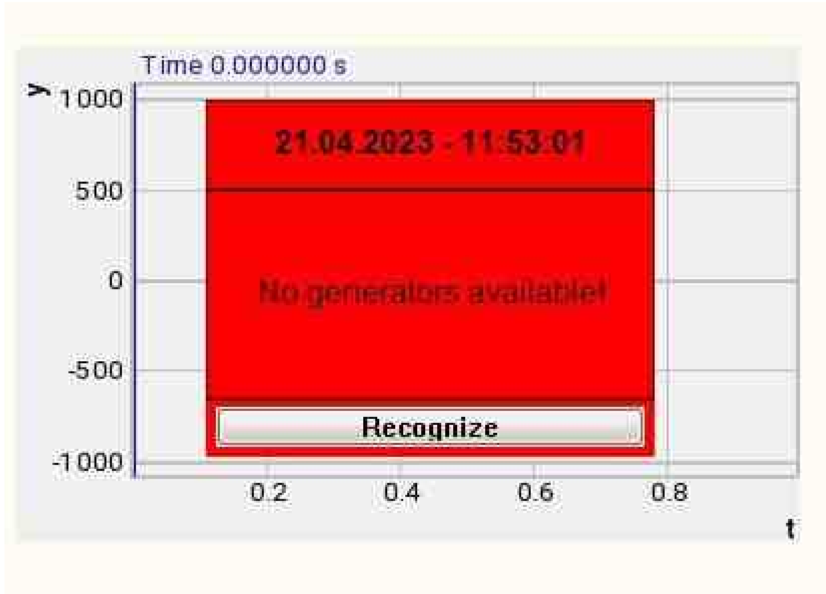
Project operation results



View the example in ZetView

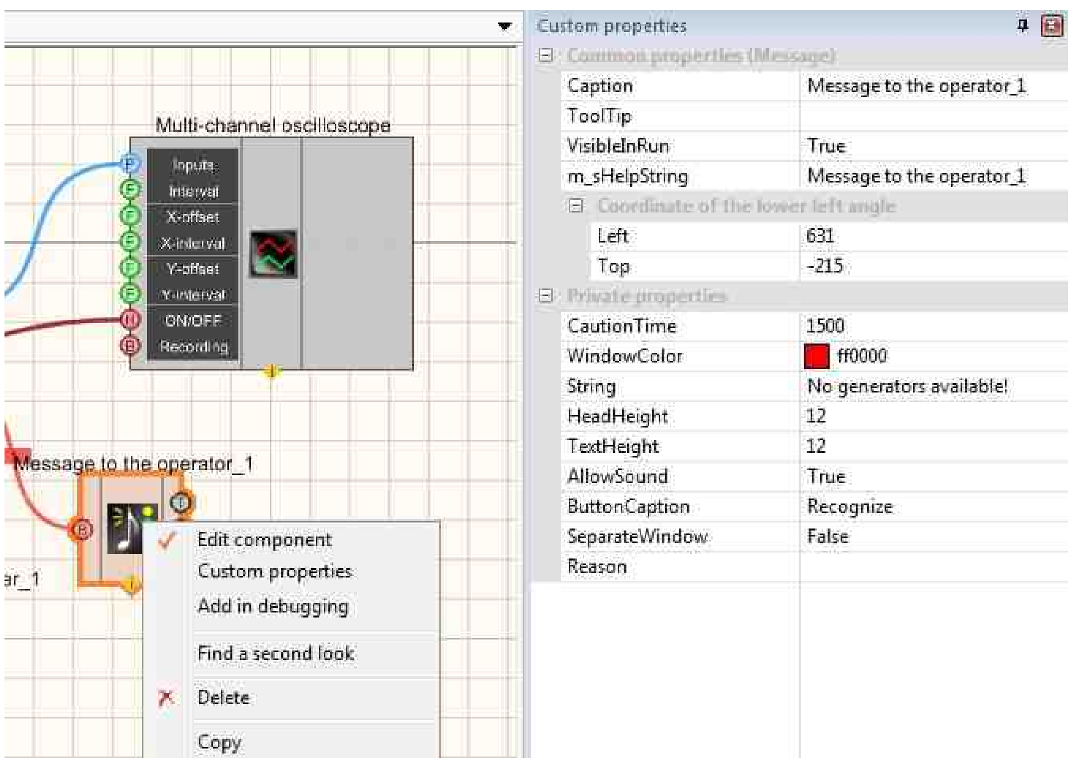
Output about an error in the project

The component Message to the operator 1052 is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal 530. In addition, the message Sine signal No. 1 530 connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

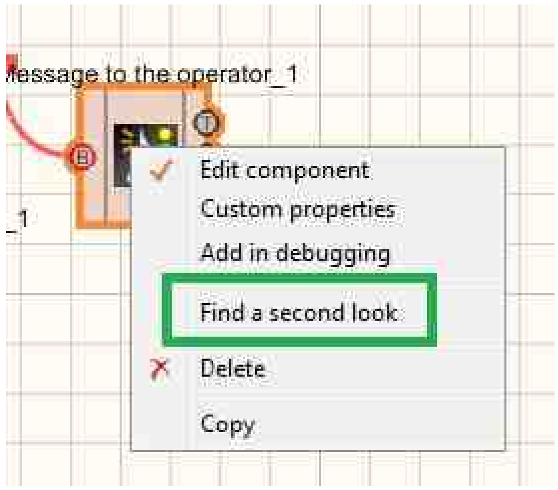


Message about the error to the operator.

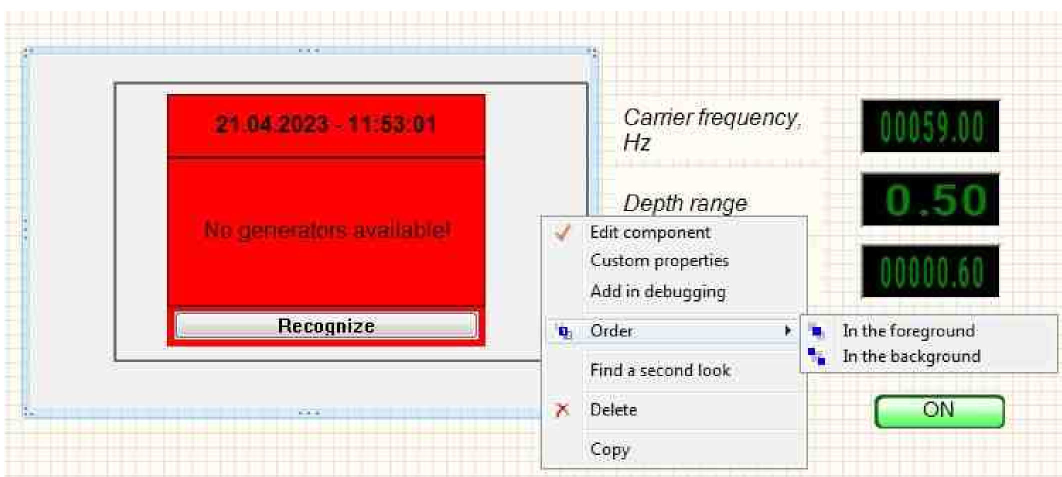
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



Mathematical description

Amplitude modulation is a type of modulation in which the variable parameter of the carrier signal is its amplitude.

Let S(t) be an information signal, |S(t)<1|.

$U_c(t)$ carrier vibration.

Then the amplitude modulated signal U_{am} can be written like this:

$$U_{am}(t) = U_c(t)[1 + mS(t)]. \quad (I)$$.

Here m is some constant called the modulation factor. Formula (I) describes the carrier signal U_c, modulated in amplitude by a signal S(t) with a modulation factor m. It is also assumed that the following conditions are met:

$$|S(t)| < 1; \quad 0 < m \leq 1. \quad (II)$$.

The fulfillment of conditions (II) is necessary so that the expression in square brackets in (I) is always positive. If it can take negative values at some point in time, then the so-called overmodulation (overmodulation) occurs.

Let's say we want to modulate a carrier wave with a monoharmonic signal. Expression for carrier oscillation with frequency ω_c, we set the initial phase equal to zero, has the form:

$$U_c(t) = C\sin(\omega_c t)$$.

Expression for a sine signal with a frequency ω_s (signal to be transmitted) has the form:

$$U_s(t) = U_0 \sin(\omega_s t + \varphi)$$,

where φ is the initial phase. Then according to (I)

$$U_{am}(t) = C[1 + mU_0 \sin(\omega_s + \varphi)]\sin(\omega_c t)$$.

The above formula for y(t) can be written as follows:

$$U_{am}(t) = C\sin(\omega_c t) + \frac{mCU_0}{2}\left(\cos\left((\omega_c - \omega_s)t - \varphi\right) - \cos\left((\omega_c - \omega_s)t + \varphi\right)\right)$$.

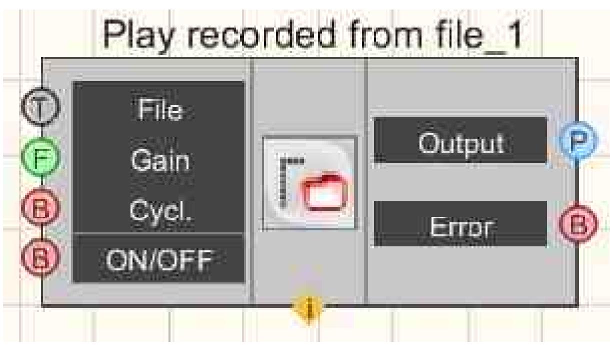
9.2.Play recorded from file

The component "**Play recorded from file**" is designed to generate a signal generated signal according to the parameters taken from the file.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

<table>
<tr><td>Developer interface</td><td>Operator interface</td></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:

- File - file-name for play recorded signals.
- Gain - gain of the signal from the file.
- Cycl. - number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- ON\OFF - turn on and off the generator.

➢Output

- Output - output of the generator generated from the signal file.
- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FileName - select of the required file with the signal pre-recorded in it.
- Amplify (1) - the gain of the signal from the file.
- Cycle (true) - the parameter responsible for the number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- Activate (false) - the status of the signal (enabled or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or Serial Number is 0, then the selection goes among all available channels of all devices.
- DeviceType () - device type: ZET 220; ZET 230; ZZT240; ZET017-U2; ZET017-U4/8, the generator of which must be activated.
- SerialNumber (0) - serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- File - the value of the file name from which the waveform data will be taken for play recorded signals (text).
- Gain - the value of the signal gain from the file (number).
-

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR FileName - Sets the selection of the required file with the signal pre-recorded in *.dtu format (after setting this parameter, reading from the specified file is specified).
- FLOAT Amplify - Set the signal gain from the file (from 0.001 to 999.9).
- VARIANT_BOOL Cyclic - Set the cyclic of the signal:
  true - the signal cyclic status is turned ON;
  false - the signal cyclic status is turned OFF.
- . VARIANT_BOOL Activate - Set the signal activate status:

true - the status of the presence of the signal is turned ON;
false - the status of the presence of the signal is turned OFF.

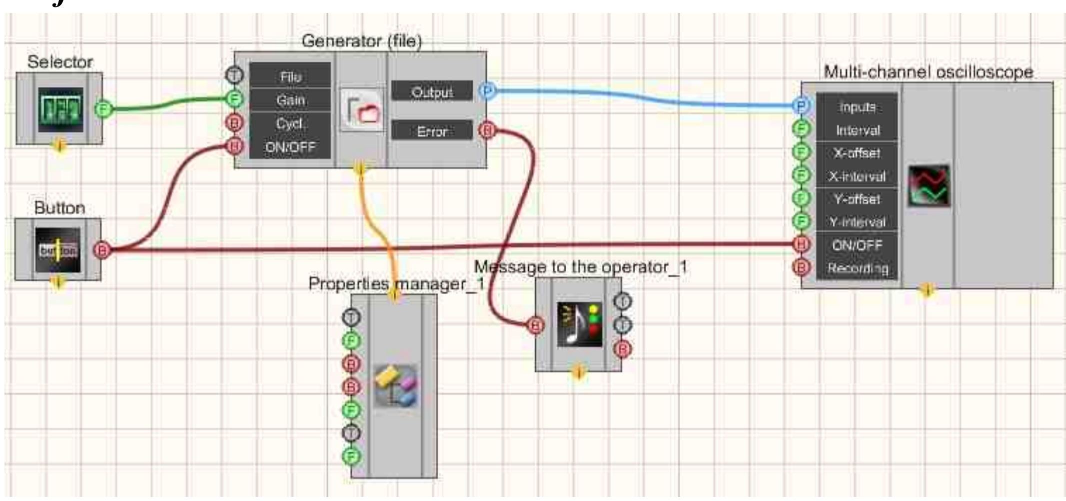
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set device type: ZET 220; ZET 230; ZET 240; ZET017-U2; ZET017-U4/8 (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢**Methods:**

- void SetSignal(BSTR fileName, FLOAT coeff, VARIANT_BOOL cycle) - Set the file name, gain, cycle.
- void DeleteSignal(void) - Stop the generator

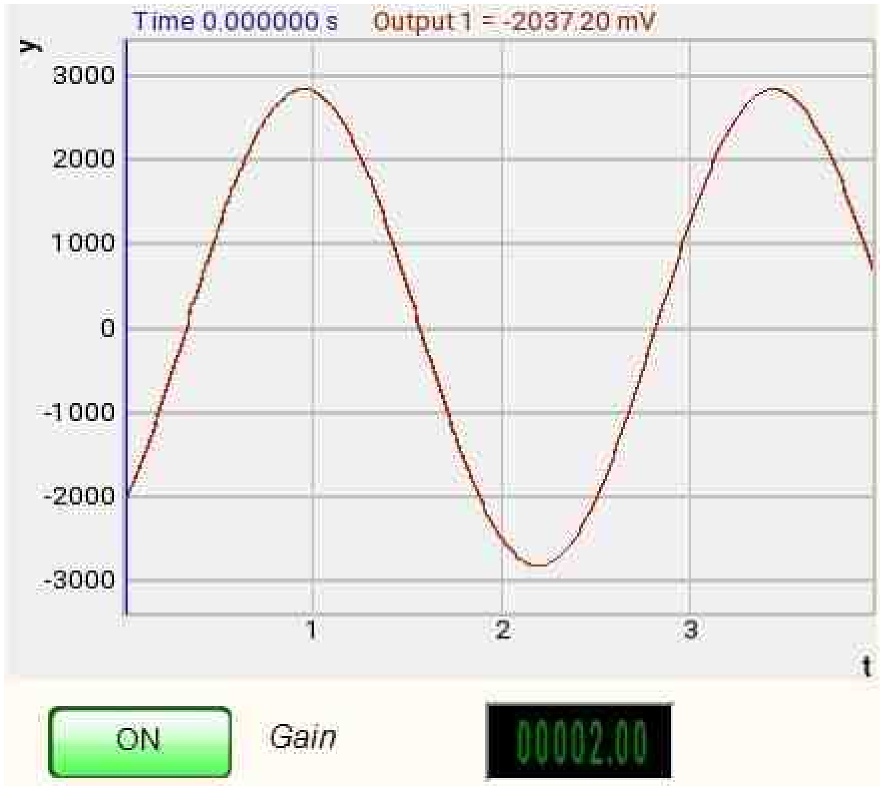
Example

Project in SCADA ZETView



This project shows the operation of the Play recorded from file 462 component outputs a previously recorded signal from a file to the input of the Multi-channel oscilloscope 928. Using the Selector 428, you can set the gain for the signal reproduced from the file. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change indicators in real time and transfer data to the signal generator to display them on a multi-channel oscilloscope.

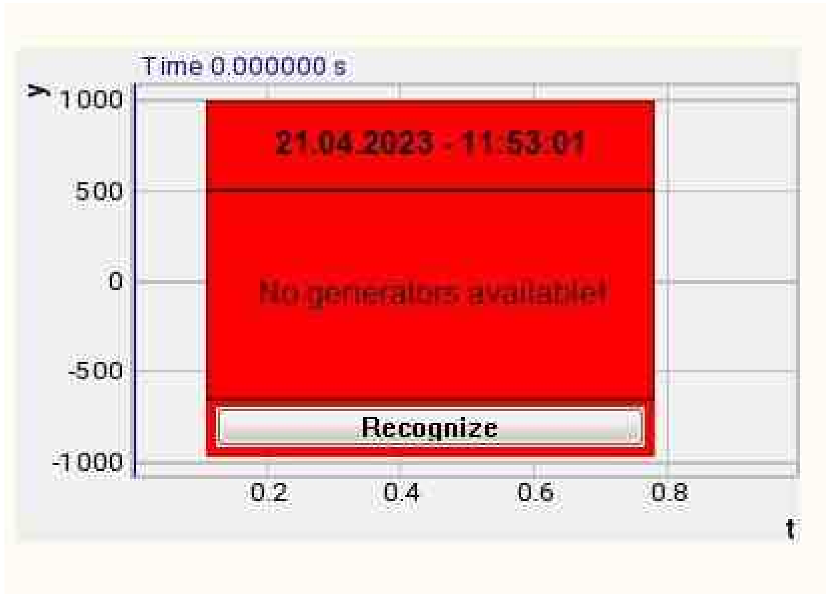
Project operation results



View the example in ZetView

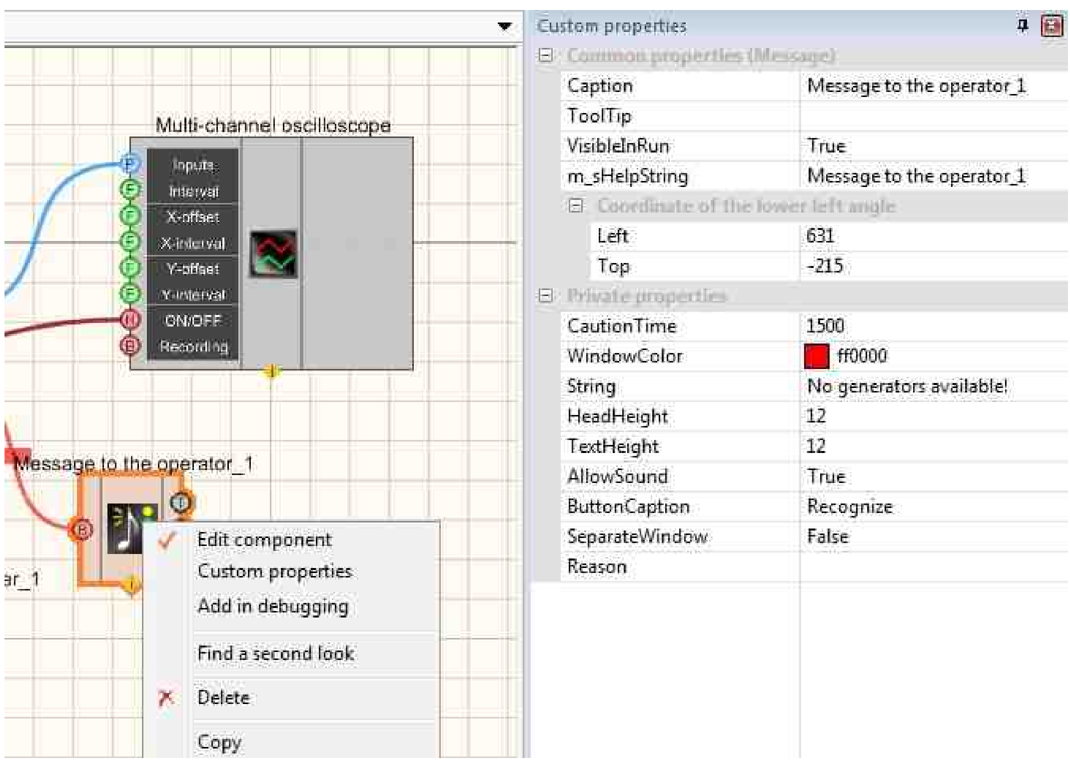
Output about an error in the project

The component Message to the operator 1052 is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal 530. In addition, the message Sine signal # 1 530 connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

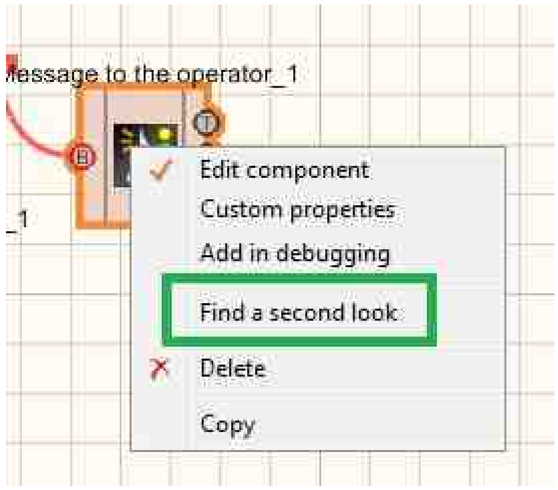


Message about the error to the operator.

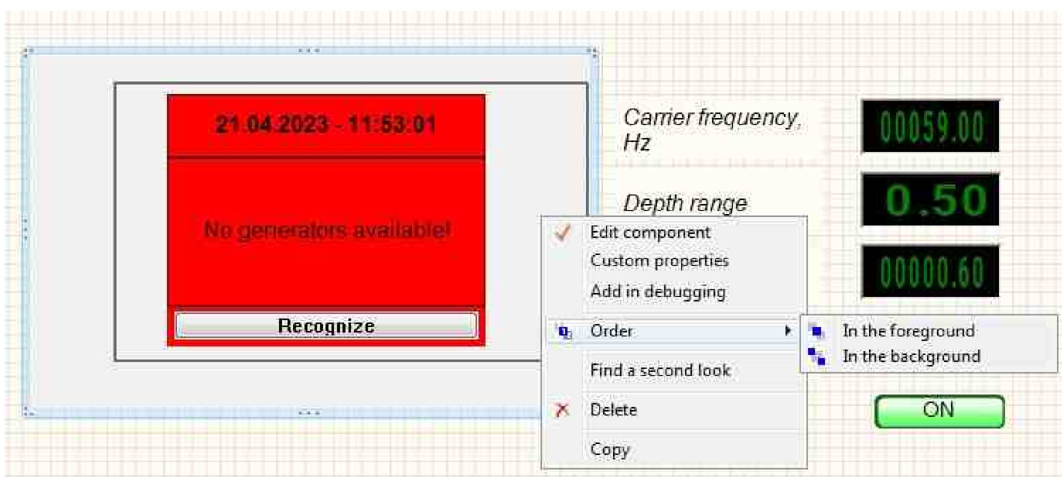
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.

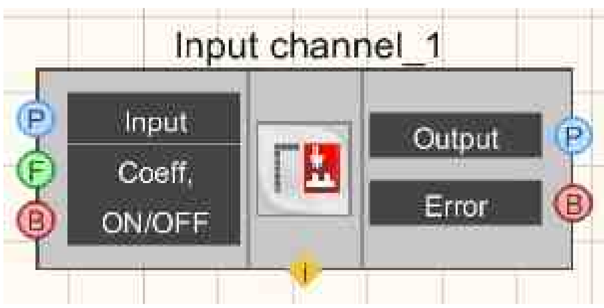


9.3.Input channel

The component "**Input channel**" is user for select one of the channels of the connected physical ZET device. Sends the ID of the selected channel to the output of the next component, which, in turn, receives signals from the ZET device and processes them.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - input channel.
- Coeff. - signal gain/attenuation factor.
- ON\OFF - turn on and off the generator.

➢Output
- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

➢**Custom common properties:**

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Koefficient (1) - the signal gain/attenuation factor.
- Activate (false) - the status of the signal (applied or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of

the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.

- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢Input:

- Input - Measuring channel (from 0 to (number of channels - 1)).
- Coeff.- The value of the signal gain/attenuation factor (number).

➢Custom common properties:

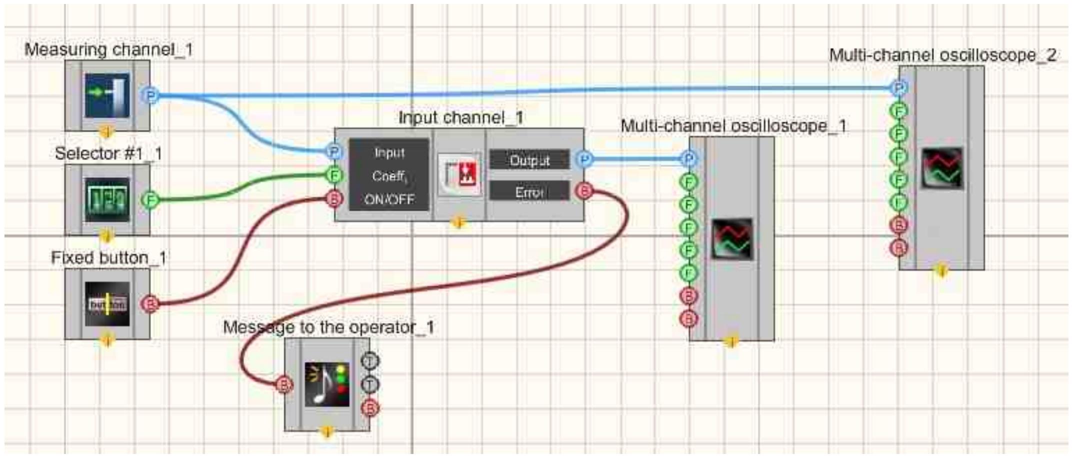
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- FLOAT Koefficient - Set the gain/attenuation factor) of the channel file to generate a signal from the channel (from 0.001 to 99.9).
- VARIANT_BOOL Activate - Set the signal presence status:
- true - the signal presence status is turned ON;
- false - the signal presence status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

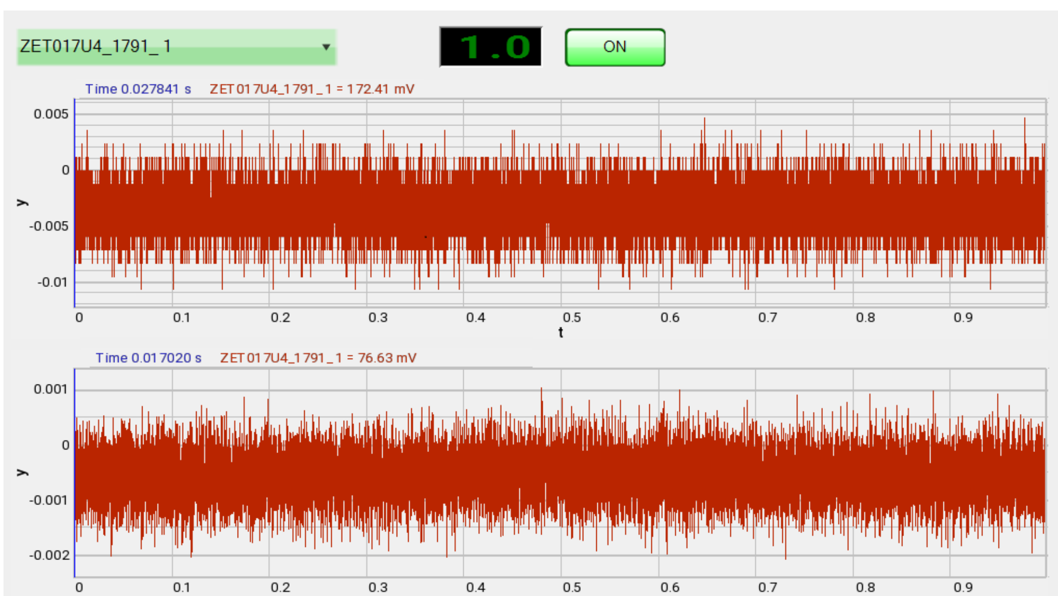
Example

Project in SCADA ZETView



This project shows the operation of the Play recorded from file 462 component outputs the previously recorded signal from the file to the input of a Multi-channel oscilloscope 928. Using the Selector 428, you can set the gain for the signal reproduced from the file. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

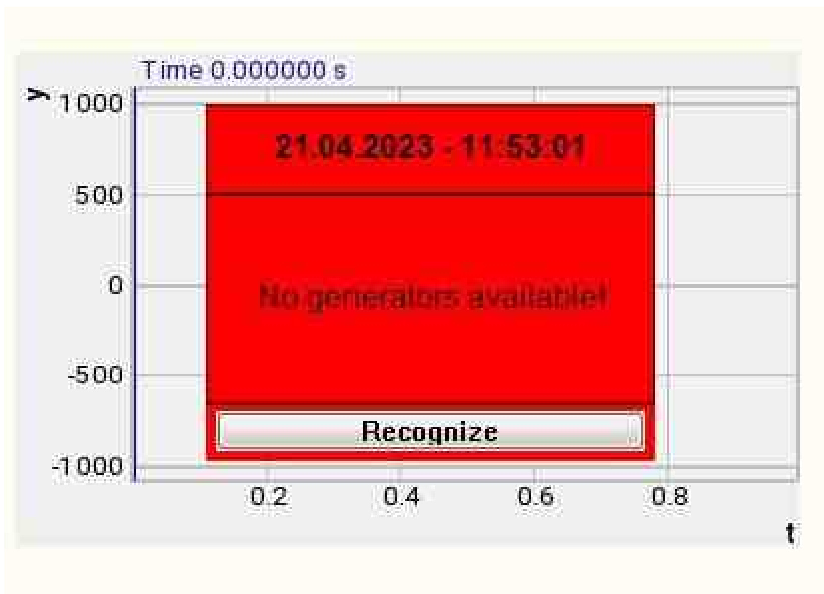
Project operation results



View the example in ZetView

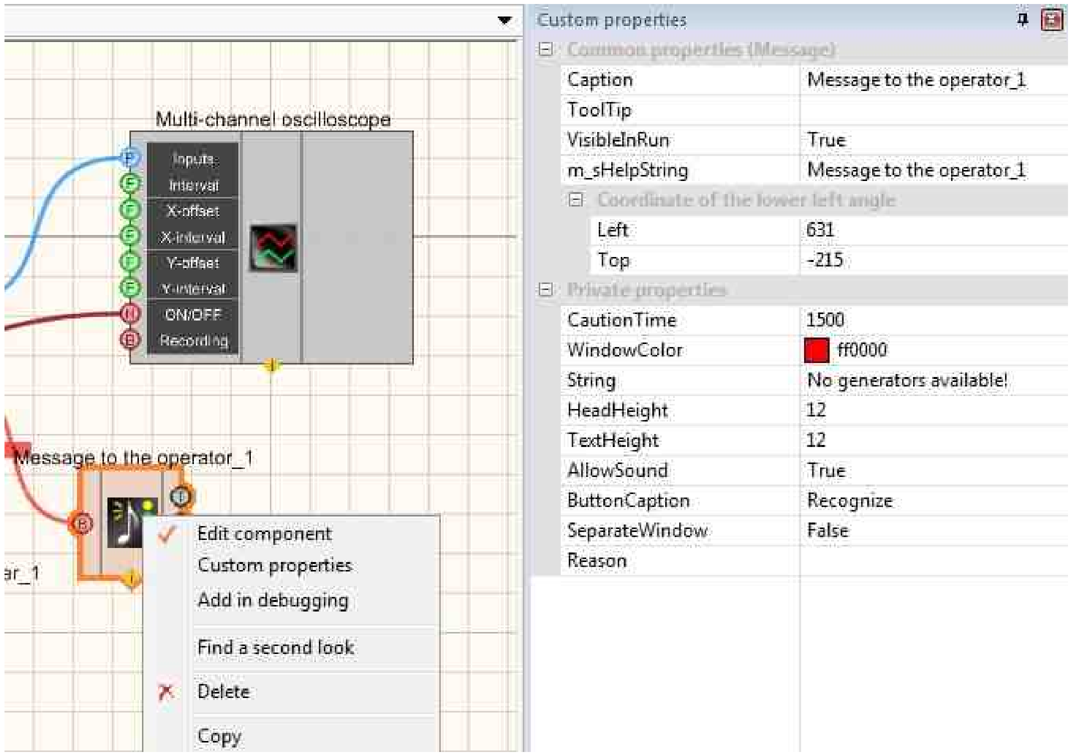
Output about an error in the project

The component Message to the operator1052 is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal530. In addition, the message Sine signal # 1530 connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

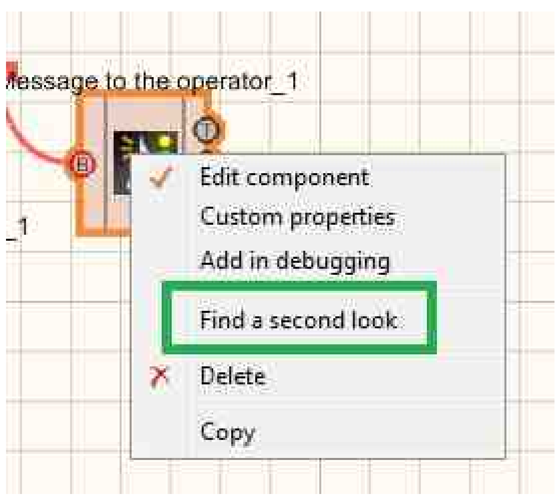


Message about the error to the operator.

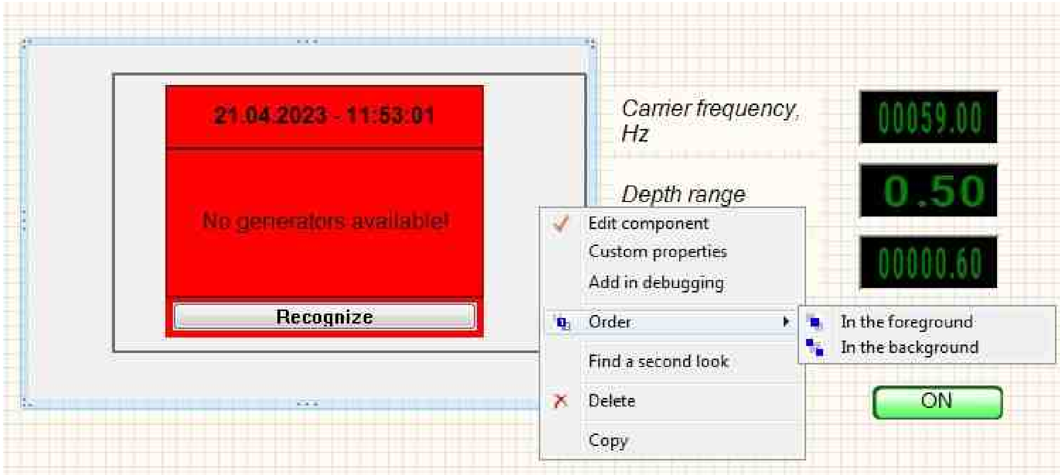
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



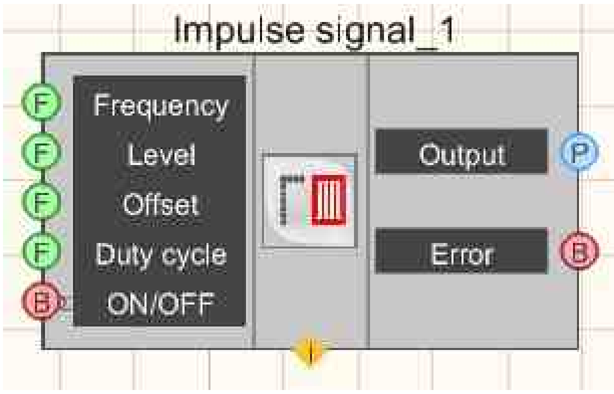
9.4.Impulse signal

The component "**Impulse signal**" is user to generate rectangular impulses with a given frequency, duty cycle, level and constant component.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Frequency - impulse signal frequency, Hz.
- Level - impulse signal level, V.
- Offset - offset of the constant component of the impulse signal, V.
- Duty cycle - (duty factor) is designed to set the duty factor (the ratio of the impulse duration to the repetition period), with which the signal will be generated.
- ON\OFF - turn ON and OFF the generator.

➤Output

- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- Frequency (100) - output signal frequency, Hz.
- Level (1) - output signal amplitude, V.
- Shift (0) - offset of the constant component, V.
- Porosity (0.5) - parameter responsible for the duty cycle of the impulse (the ratio of the impulse duration to the repetition period).
- Activate (false) - the status of the signal (enabled or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.

- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Frequency - impulse signal frequency, Hz (number).
- Level - impulse signal level, V (number).
- Offset - offset of the constant component of the impulse signal, V (number).
- Duty cycle - (duty factor) is designed to set the duty factor (the ratio of the impulse duration to the repetition period), with which the signal will be generated.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

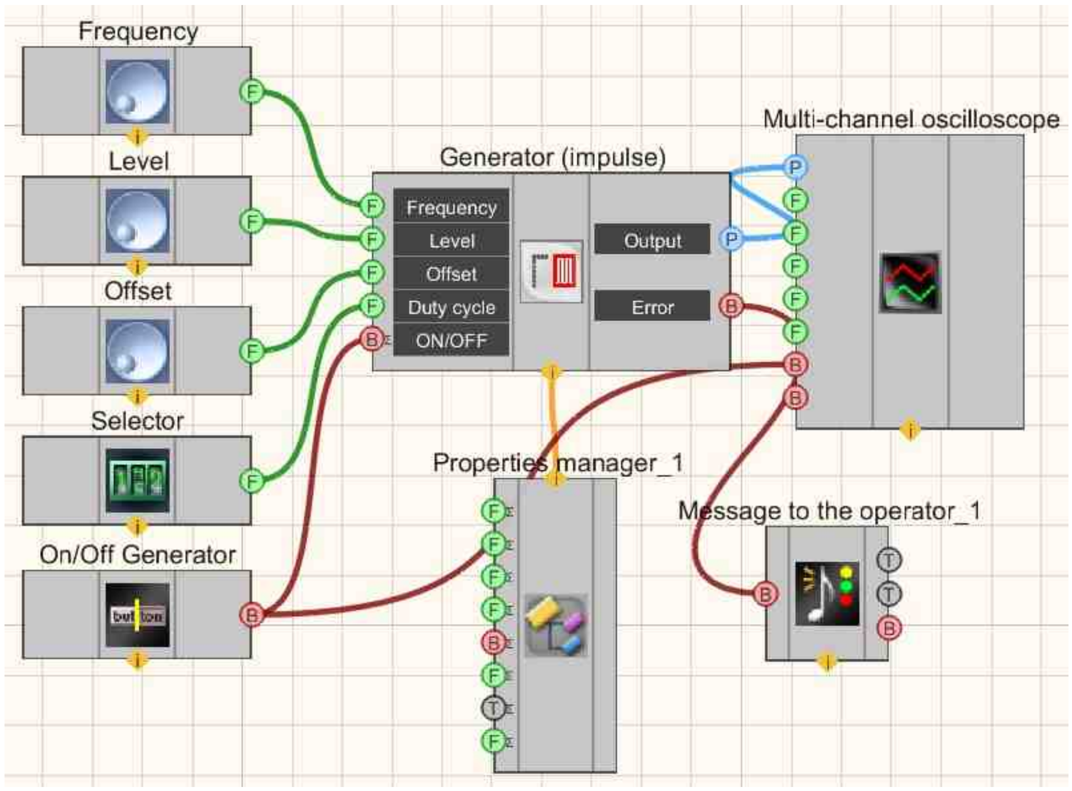
- FLOAT Frequency - Set the signal frequency (from 0.01 Hz to (DAC sampling frequency / 2)), Hz
- FLOAT Level - Set the signal level (from 0 V to the maximum limit DAC signal level), V
- FLOAT Shift - Set the DC offset (from minus the maximum limit DAC signal level to the maximum limit DAC signal level), V.
- FLOAT Porosity - Set the duty cycle of the signal (from 0 to 1).
- VARIANT_BOOL Activate - Set the signal presence status:
- true - the signal presence status is turned ON;
- false - the signal presence status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢**Methods:**

- void SetSignal(FLOAT frequency, FLOAT level, FLOAT smeshenie, FLOAT porosity) - Set the frequency, level, offset, duty cycle.
- void DeleteSignal(void) - Stop the generator.

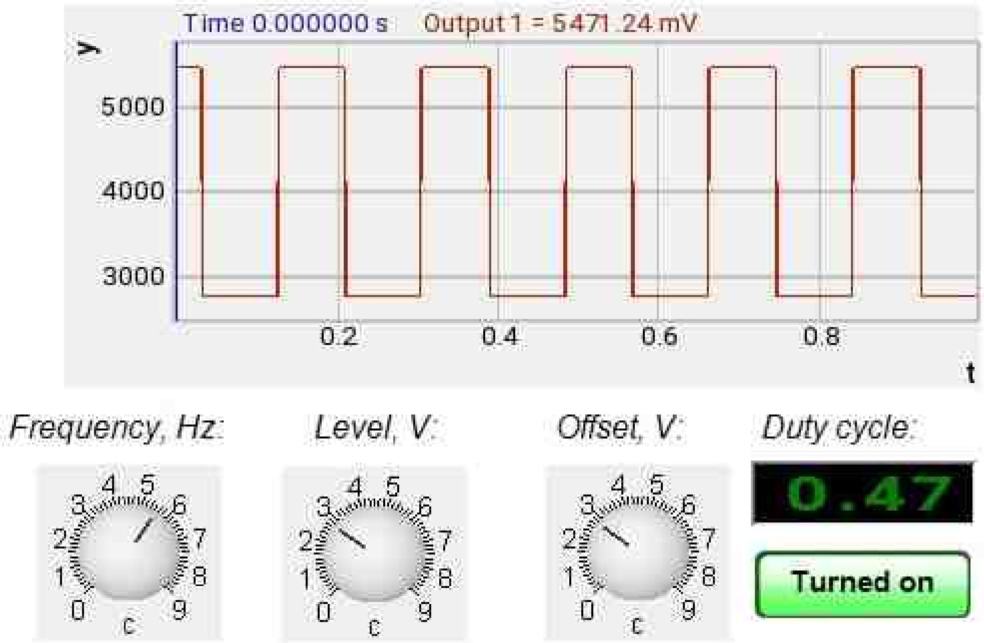
Example

Project in SCADA ZETView



This project shows the operation of the Impulse signal 473 component generates rectangular impulses. Selectors 428 (as well as another type) are used to set the frequency, level, offset, duty cycle. The button is required to enable/disable the Impulse signal 473. A Multi-channel oscilloscope 928 is used to graphically display the signal. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change indicators in real time and transfer data to the signal generator to display them on a multi-channel oscilloscope.

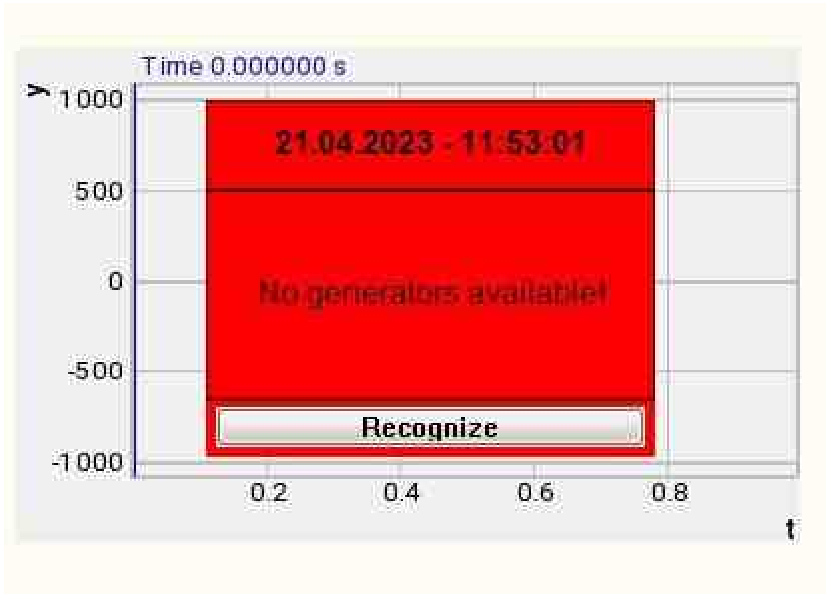
Project operation results



View the example in ZetView

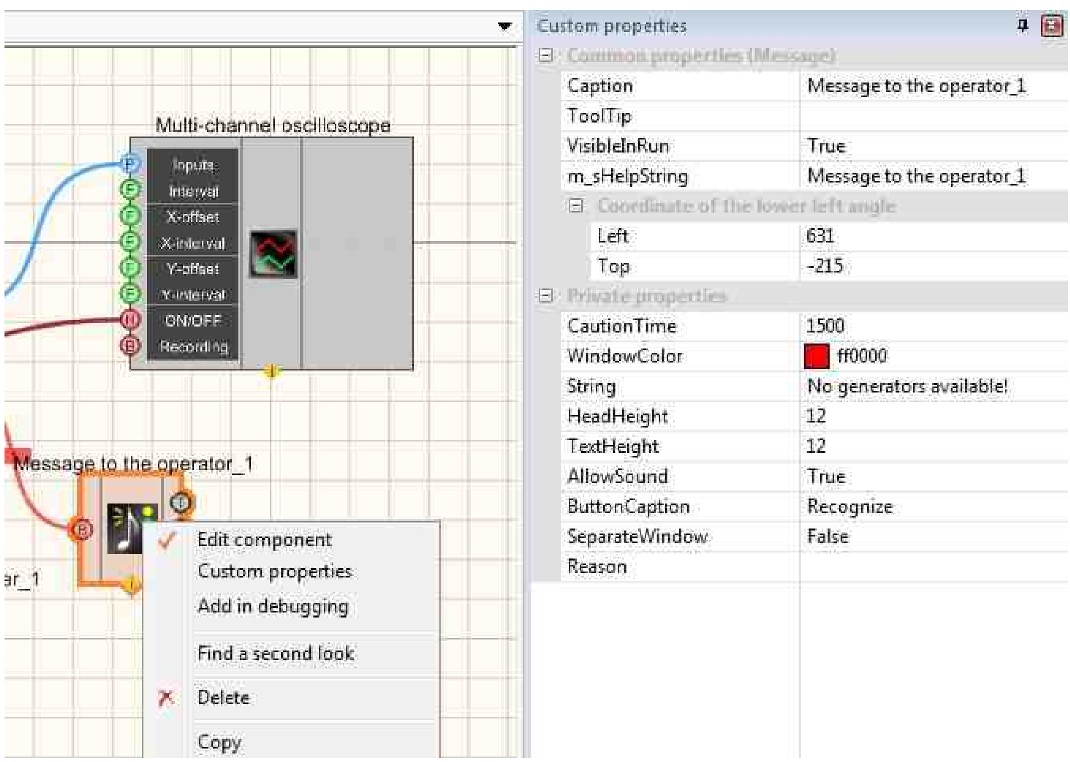
Output about an error in the project

The component [Message to the operator](1052) is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the [Sine signal](530). In addition, the message [Sine signal # 1](530) connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

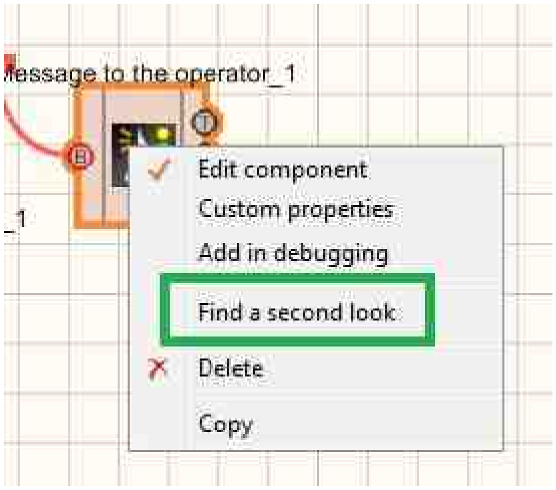


Message about the error to the operator.

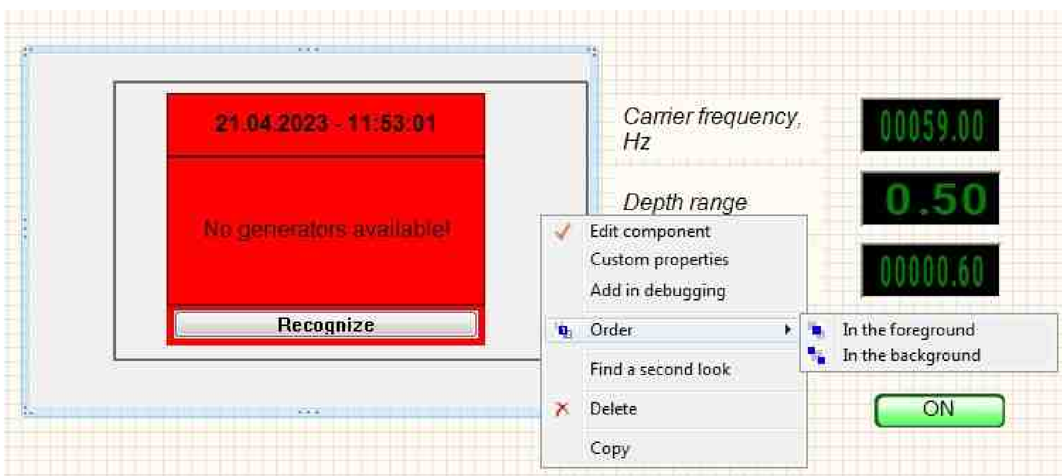
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



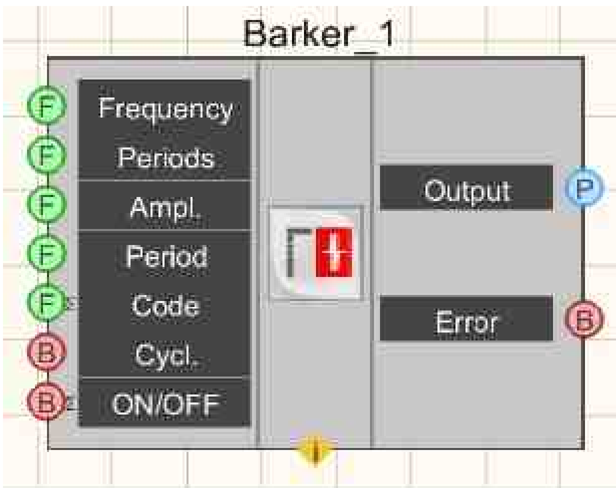
9.5.Code Barker

The component "**Code Barker**" is user to generate phase-modulated signals

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Frequency - the frequency of the signal, Hz.
- Periods - the periods are set.
- Ampl. - the amplitude of the Barker code, V.
- Period - the period is set, Hz.
- Barker code - the Barker code is set.
- Cycl. - number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- ON\OFF - turn on and off the generator.

➢Output

- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Frequency (1000) - the signal frequency, Hz.
- Periods (25) - the periods are set.
- Amplitude (1) - the amplitude of the output signal, V.
- Period (1) - Barker code period, Hz.
- Kod (1) - the Barker code.
- Cycle (true) - the parameter responsible for the number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- Activate (false) - the status of the signal (enabled or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Frequency, Hz - the value of the signal frequency, Hz (number).
- Periods - the value of periods (number).
- Amplitude, V - the amplitude of the Barker code, V (number).
- Period, Hz - set the period, Hz (number).
- Barker code - set the Barker code (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

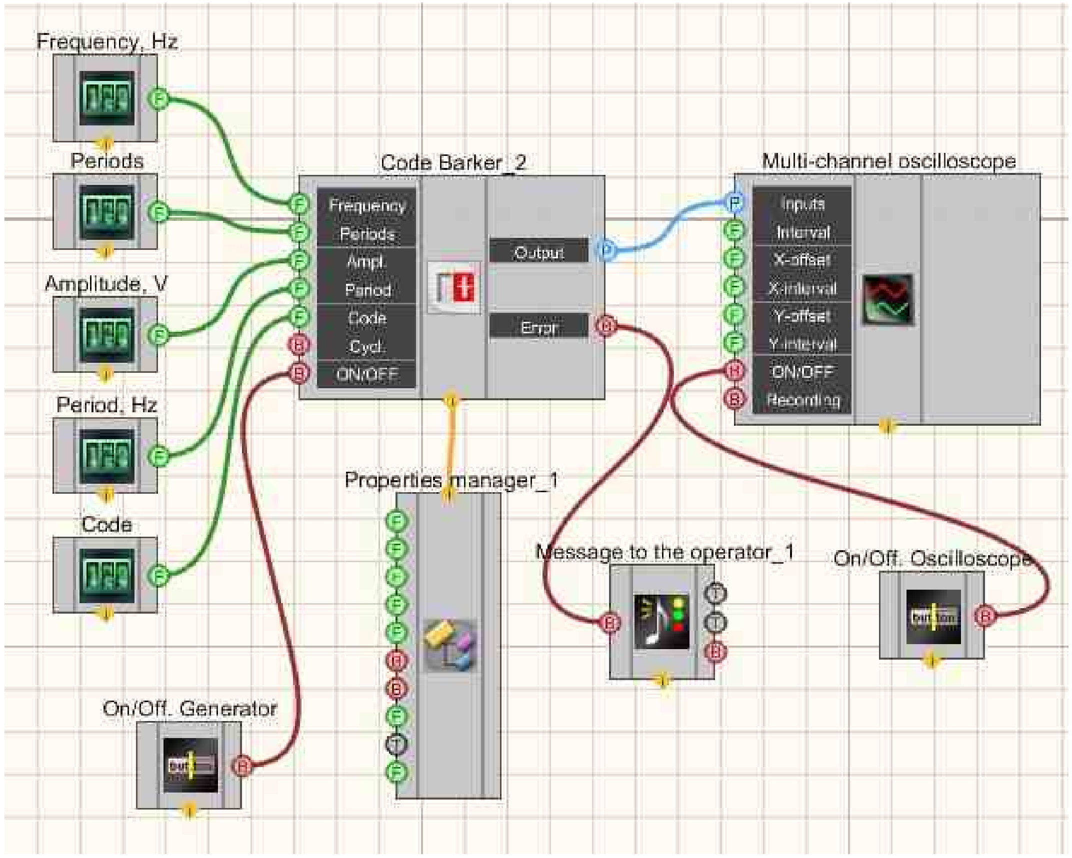
- FLOAT Frequency - Set the signal frequency, Hz.
- FLOAT Periods - Set periods.
- FLOAT Amplitude - Set the amplitude, V.
- FLOAT Period - Set the period, Hz.
- FLOAT Kod - Set the Barker code.
- VARIANT_BOOL Activate - Set the signal presence status:
- true - the signal presence status is turned ON;
- false - the signal presence status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢**Methods:**

- void SetSignal(FLOAT Frequency, FLOAT Periods, FLOAT Amplitude, FLOAT Period, FLOAT Kod, BOOL Cycle) - Set The Frequency, Periods, Amplitude, Period, Code.
- void DeleteSignal(void) - Stop the generator.

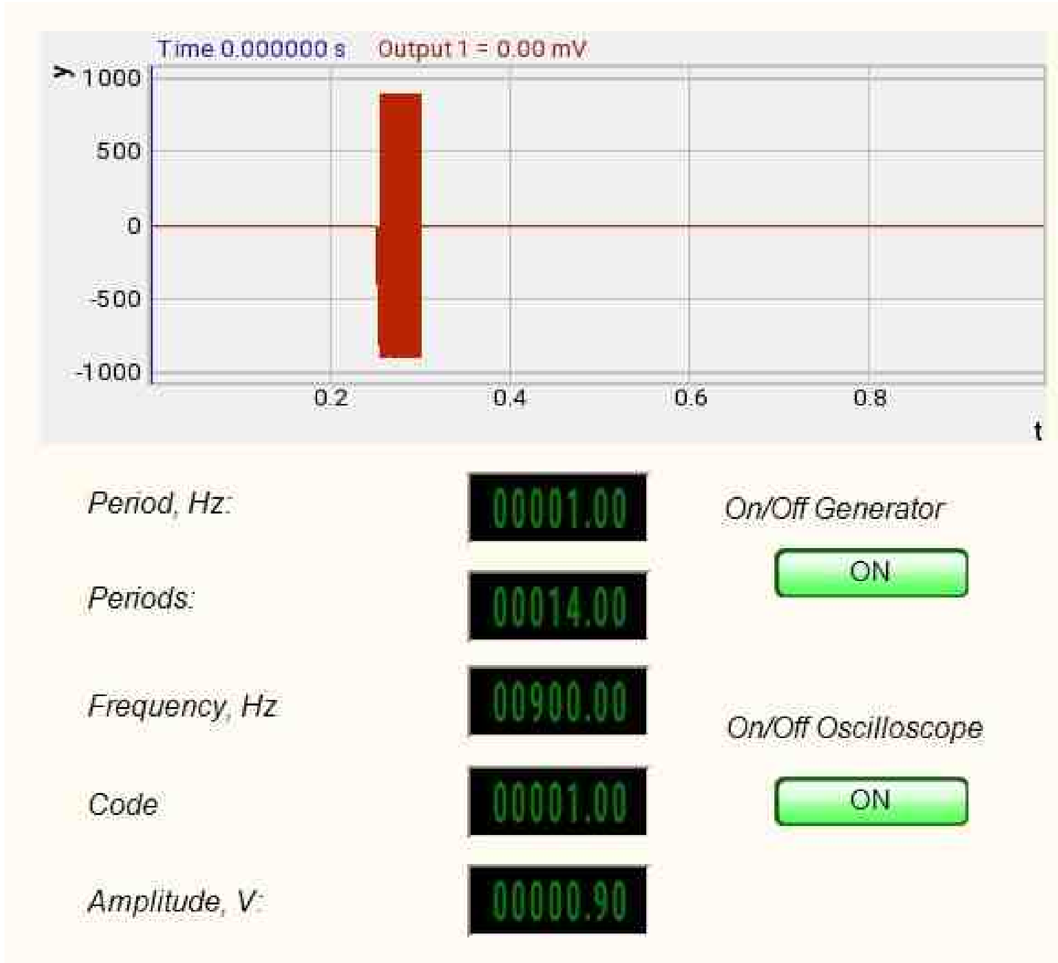
Example

Project in SCADA ZETView



This project shows the operation of the Code Barker 479 component generates the Barker codes. Selectors 428 (as well as other types) are used to set frequencies, periods, amplitudes, period, Barker code. A Multi-channel oscilloscope 928 is used to graphically display the signal. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

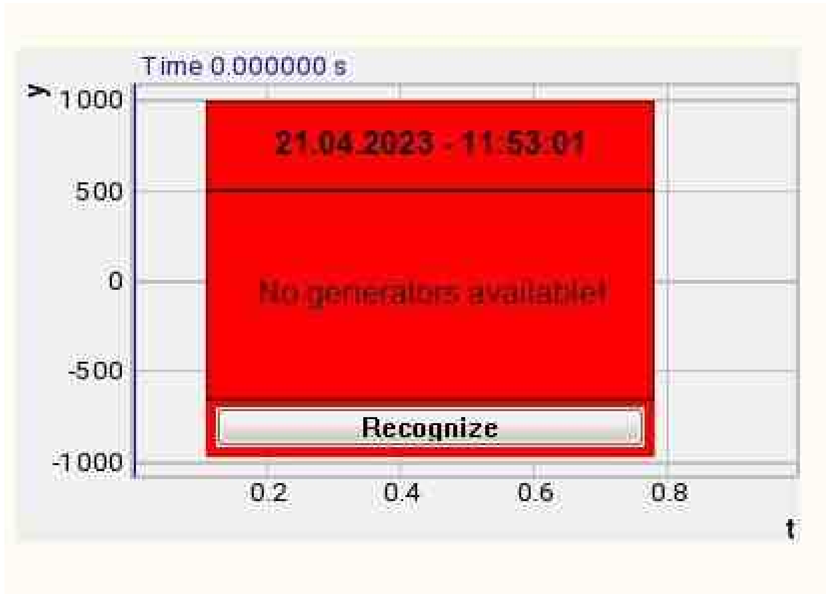
Project operation results



View the example in ZetView

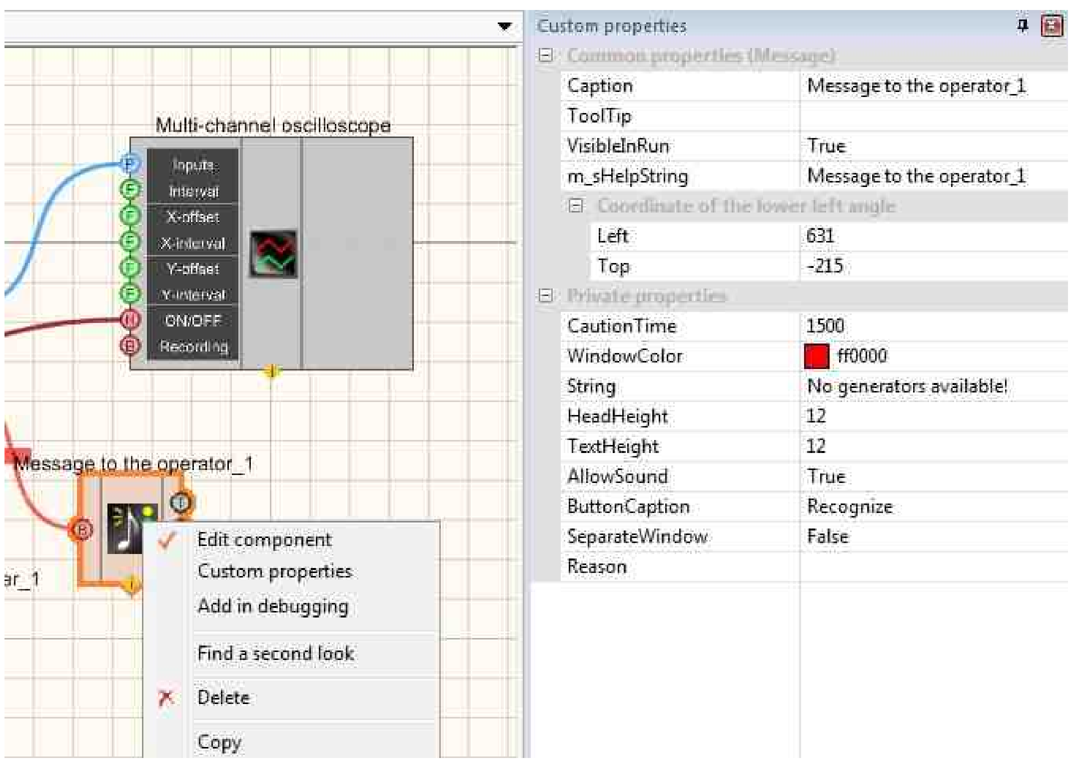
Output about an error in the project

The component Message to the operator [1052] is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal [530]. In addition, the message Sine signal # 1 [530] connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

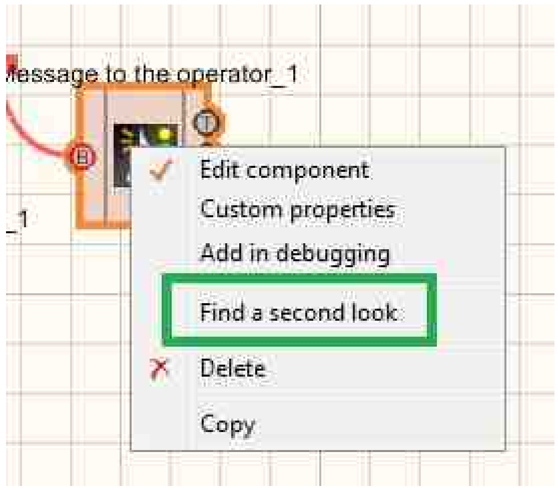


Message about the error to the operator.

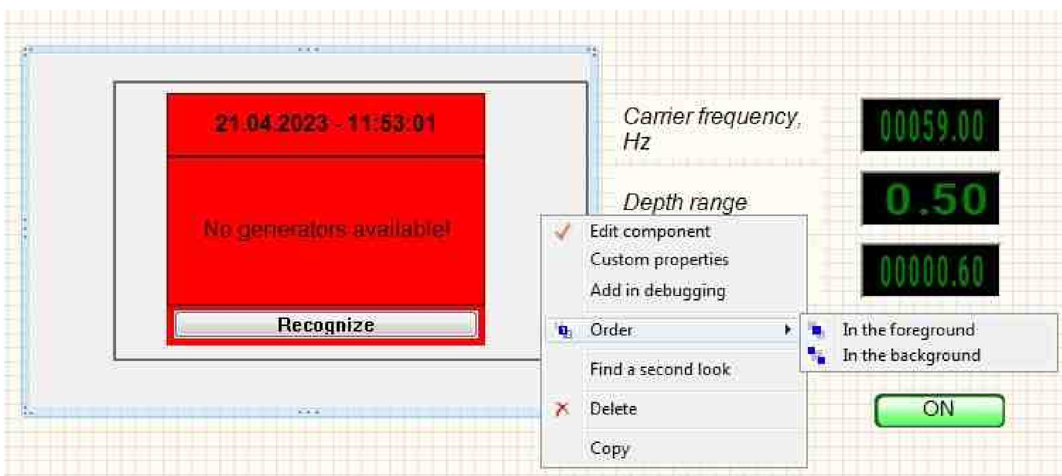
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



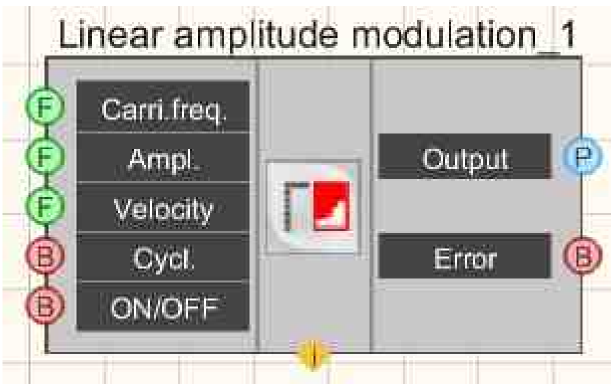
9.6.Linear amplitude modulation

The component "**Linear amplitude modulation**" is user to generate a signal generated according to a special amplitude modulation law, in which the amplitude of the oscillations varies according to a linear law in a given range cyclically.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Carri.freq. - the carrier frequency of the signal with which the LinAM will be generated, Hz.
- Ampl. - frequency of amplitude modulation of the signal with which the LinAM will be generated, V.
- Velocity - the velocity of change of the signal with which the LinAM will be generated, V/s.
- Cycl. - number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- ON\OFF - turn ON and OFF the generator.

➤Output

- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

- Carriersfrequency (0) - set the carrier frequency of the signal with which the LinAM will be generated, Hz.
- Amplitude (1000) - the amplitude of the signal with which the LinAM will be generated, V.
- Speed (1) - the velocity of change of the signal with which the LinAM will be generated, V/s.
- Activate (false) - the status of the signal (enabled or disabled).
- Cycle (true) - the parameter responsible for the number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Carrier frequency - the value of the input value of the carrier frequency in which the LinAM will be generated, Hz (number).
- Amplitude, V - the value of the input amplitude value in which the LinAM will be generated, V (number).
- Velocity - the value for setting the velocity of change of the amplitude with which the LinAM will be generated, V/s. (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Carriersfrequency - Set the carrier frequency of the signal, Hz.
- FLOAT Amplitude - Set the amplitude, Hz.
- FLOAT Speed - Set the rate of change of the signal, V/s (string).
- VARIANT_BOOL Activate - Set the signal presence status:
  true - the signal presence status is turned ON;
  false - the signal presence status is turned OFF.

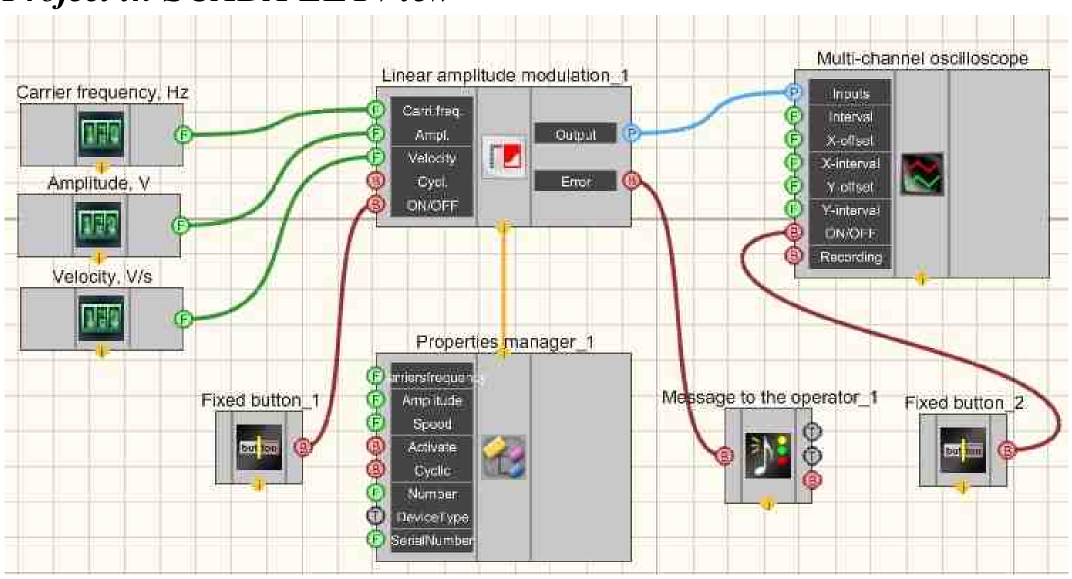
- VARIANT_BOOL Cyclic - Set the cyclic of the signal:
- true - the signal cyclic status is turned ON;
- false - the signal cyclic status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➤**Methods:**

- void SetSignal(FLOAT StartFrequency, FLOAT EndFrequency, FLOAT level, FLOAT speed, BOOL cycle) - Set the carrier frequency, Amplitude, Velocity, Cyclicity.
- void DeleteSignal(void) - Stop the generator.

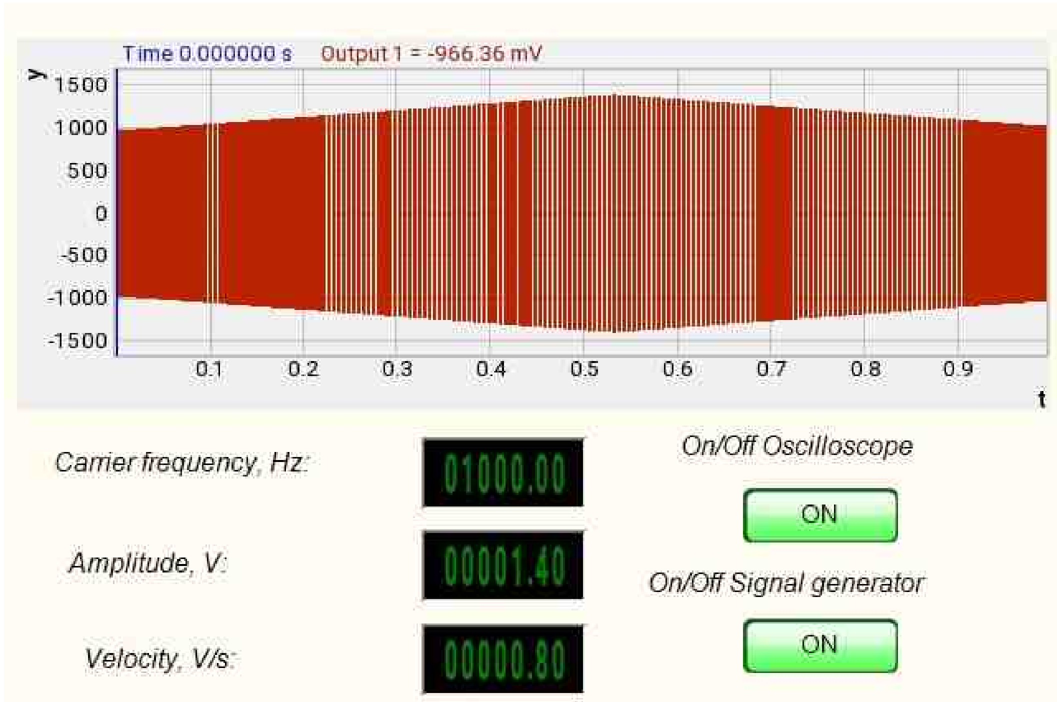
Example

Project in SCADA ZETView



This project shows the operation of the Linear amplitude modulation 486 component generates a signal generated according to a special amplitude modulation law, in which the amplitude of the oscillations varies according to a linear law in a given range cyclically. With the help of Selectors 428 (as well as another type), they are used to set the frequency, amplitude, and velocity. A Multi-channel oscilloscope 928 is used to graphically display the signal. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

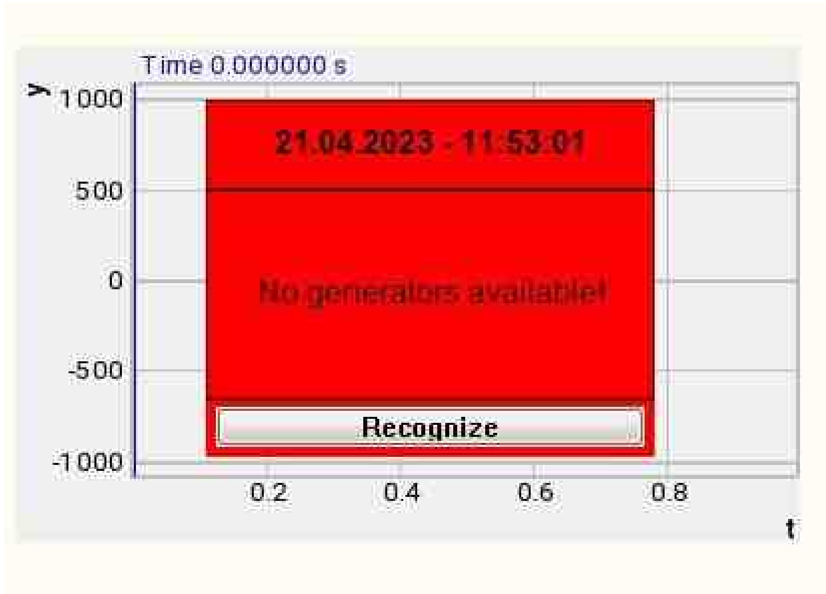
Project operation results



View the example in ZetView

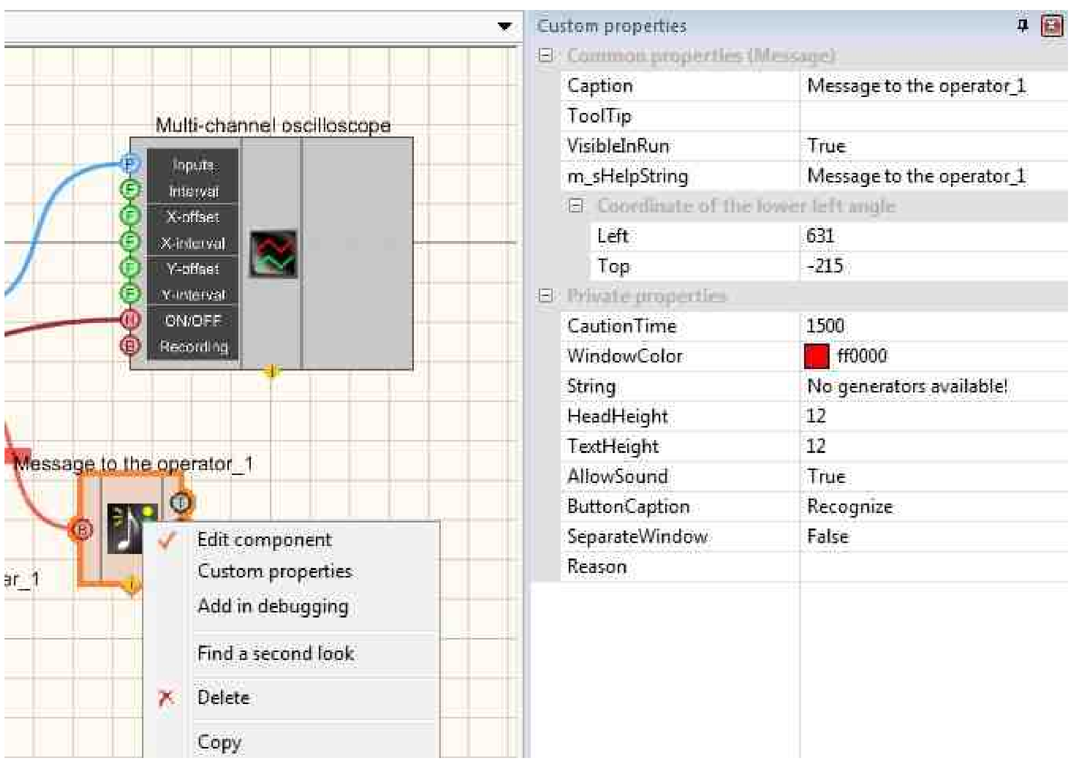
Output about an error in the project

The component Message to the operator[1052] is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal[530]. In addition, the message Sine signal # 1[530] connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

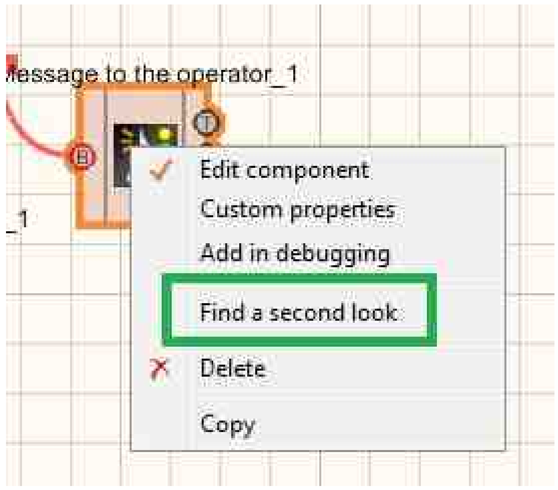


Message about the error to the operator.

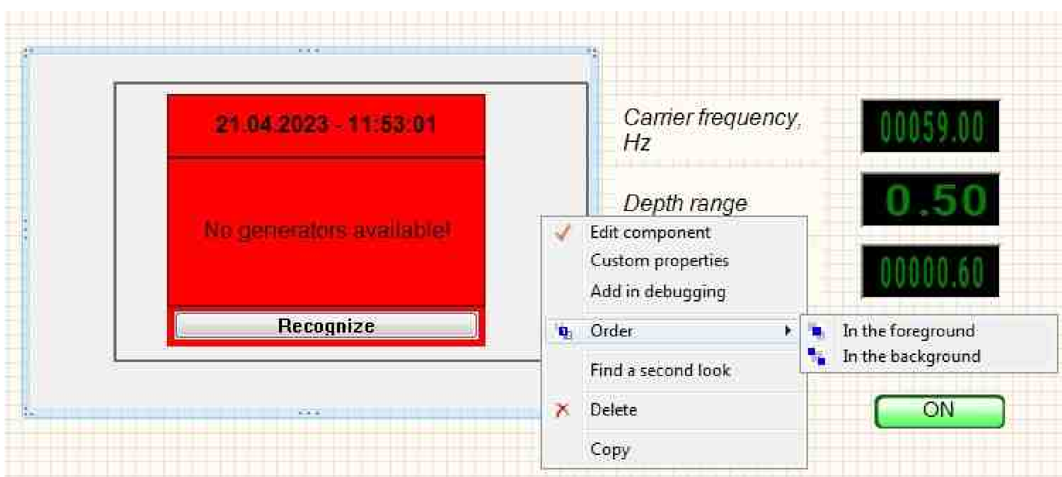
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



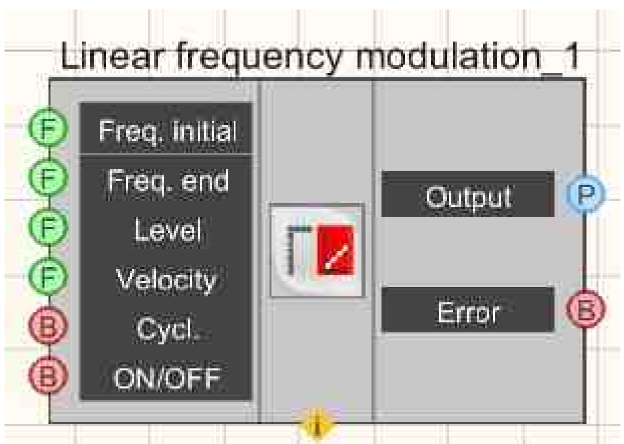
9.7.Linear frequency modulation

The component "**Linear frequency modulation**" is user to generate a signal generated by a special frequency modulation law, at which the oscillation frequency changes linearly in a certain specified range cyclically.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Frequency Initial - the initial frequency of the signal of the frequency range in which the LinFM will be generated, Hz.
- Frequency end - the end frequency of the signal of the frequency range in which the LinFM will be generated, Hz.
- Level - the level of the root-mean-square value (RMS) with which the LinFM will be generated, V.
- Velocity - the rate of change of the signal with which the LinFM will be generated, Hz/s.
- Cycl. - number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- ON\OFF - turn ON and OFF the generator.

➢Output
- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- StartFrequency (0) - the initial frequency of the signal of the frequency range in which the LinFM will be generated, Hz.
- EndFrequency (1000) - the end frequency of the signal of the frequency range in which the LinFM will be generated, Hz.
- Level (1) - the level of the root-mean-square value (RMS) with which the LinFM will be generated, V.
- Speed (1) - the parameter designed to set the rate of change of the frequency with which the LinFM will be generated, Hz/s.
- Activate (false) - the status of the signal (enabled or disabled).
- Cycle (true) - the parameter responsible for the number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**
- Initial signal frequency - the value of the input value of the initial frequency of the signal of the frequency range in which the LinFM will be generated, Hz (number)
- End signal frequency - the value of the input value of the end frequency of the frequency range in which the LinFM will be generated, Hz (number)

- Level - the value for setting the root-mean-square value (RMS) of the level with which the LinFM will be generated, V (number).
- Velocity - the value for setting the rate of change of the frequency with which the LinFM will be generated, Hz/s (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

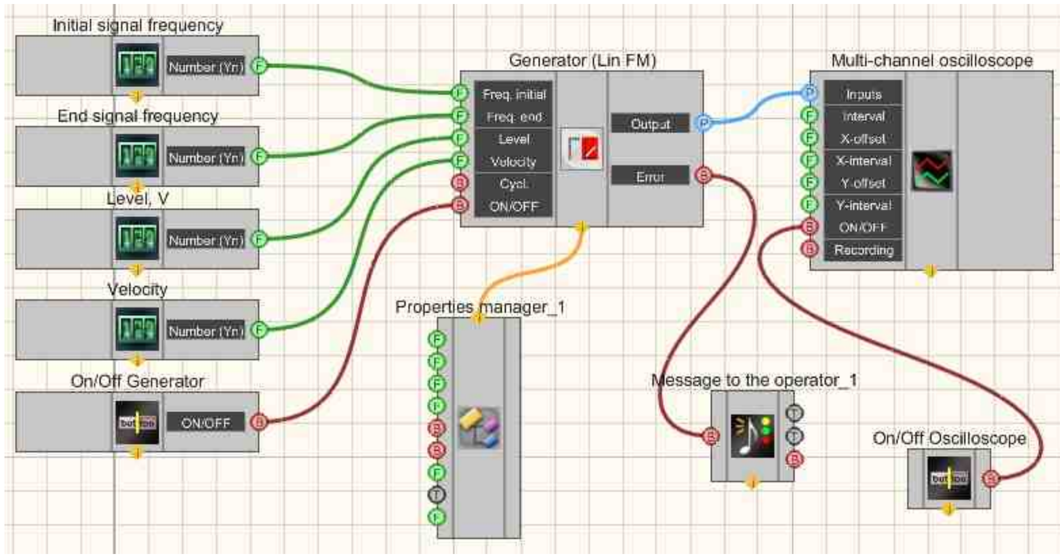
- FLOAT StartFrequency - Set the initial signal frequency of the signal (from 0.01 Hz to the end frequency of the signal of linear frequency modulation), Hz.
- FLOAT EndFrequency - Set the end signal frequency of the signal (from the initial frequency of linear frequency modulation to (DAC sampling frequency / 2)), Hz.
- FLOAT Level - Set the signal level (from 0 V to the maximum limit DAC signal level), V.
- FLOAT Speed - Set the rate of change of the signal, Hz/s (string).
- VARIANT_BOOL Activate - Set the signal presence status:
  true - the signal presence status is turned ON;
  false - the signal presence status is turned OFF.
- VARIANT_BOOL Cyclic - Set the cyclic of the signal:
- true - the signal cyclic status is turned ON;
- false - the signal cyclic status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢**Methods:**

- void SetSignal(FLOAT StartFrequency, FLOAT EndFrequency, FLOAT level, FLOAT speed, BOOL cycle) - Sets the Initial signal frequency, end signal frequency, level, velocity, cycle.
- void DeleteSignal(void) - Stop the generator.

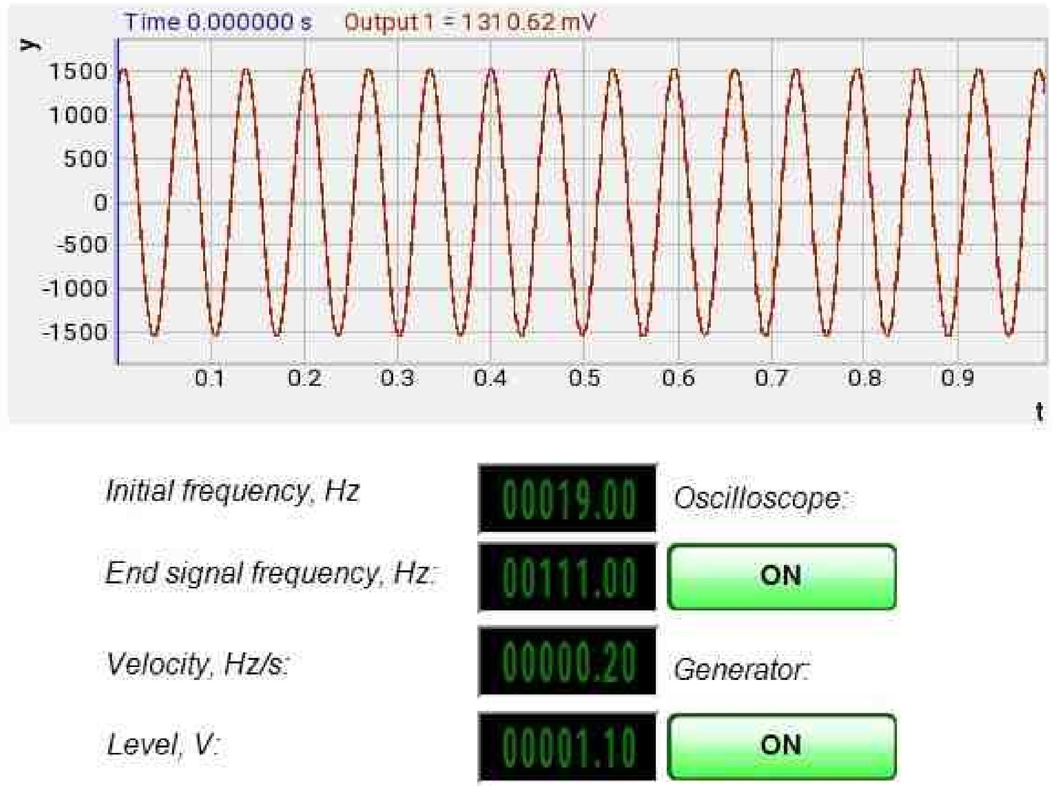
Example

Project in SCADA ZETView



This project shows the operation of the Linear frequency modulation (LinFM) 486 component generates a linear-frequency-modulated signal, the parameters of which are set using Selectors 428 (initial signal frequency, end signal frequency, level, velocity). The buttons allow you to turn on and off the Sine signal 530 and the Multi-channel oscilloscope 928. The oscilloscope is used to view the end form of the signal. The Fixed button 691 is used to turn ON or OFF.
The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a Multi-channel oscilloscope.

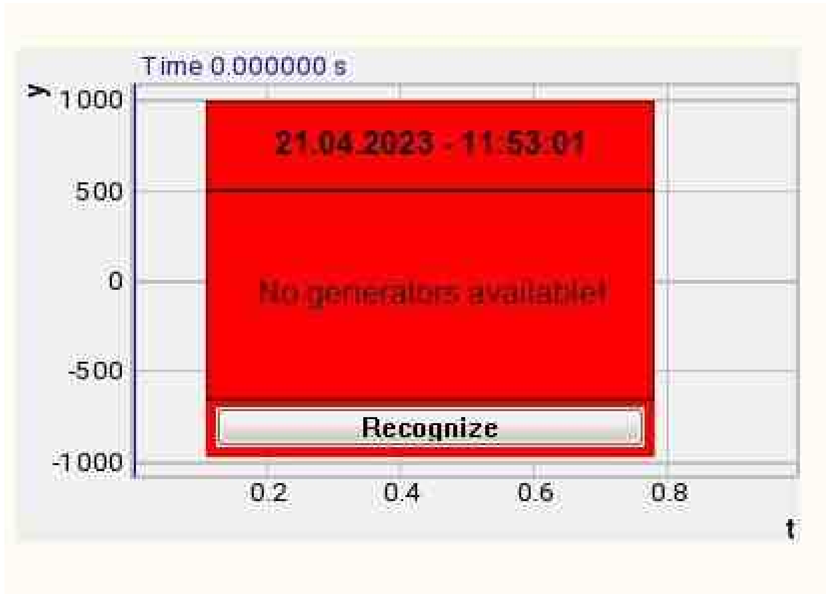
Project operation results



View the example in ZetView

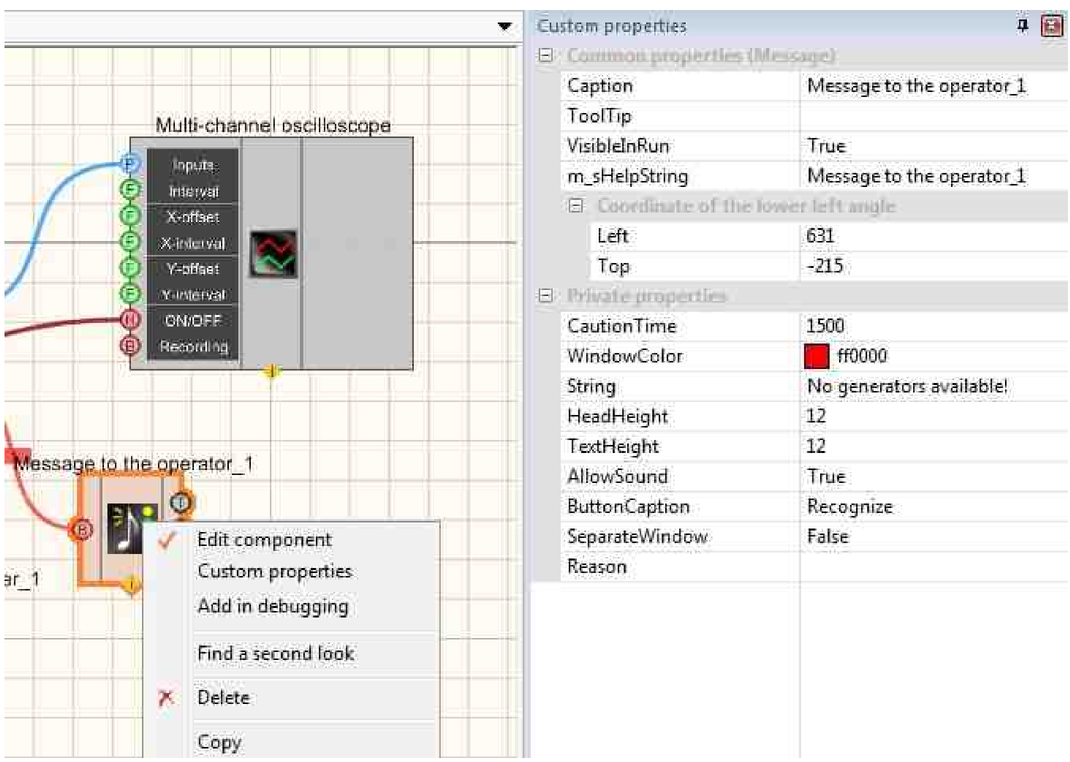
Output about an error in the project

The component [Message to the operator](1052) is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the [Sine signal](530). In addition, the message [Sine signal # 1](530) connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

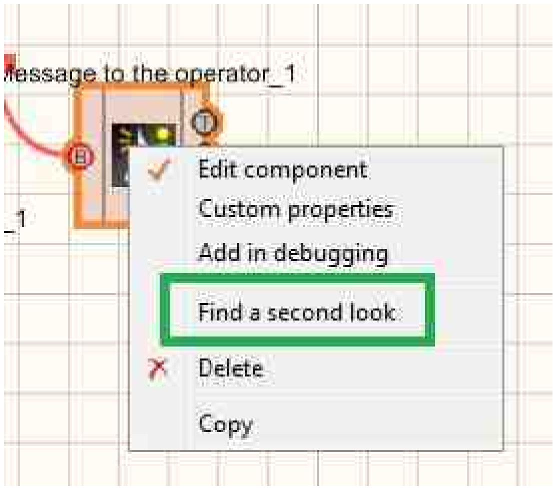


Message about the error to the operator.

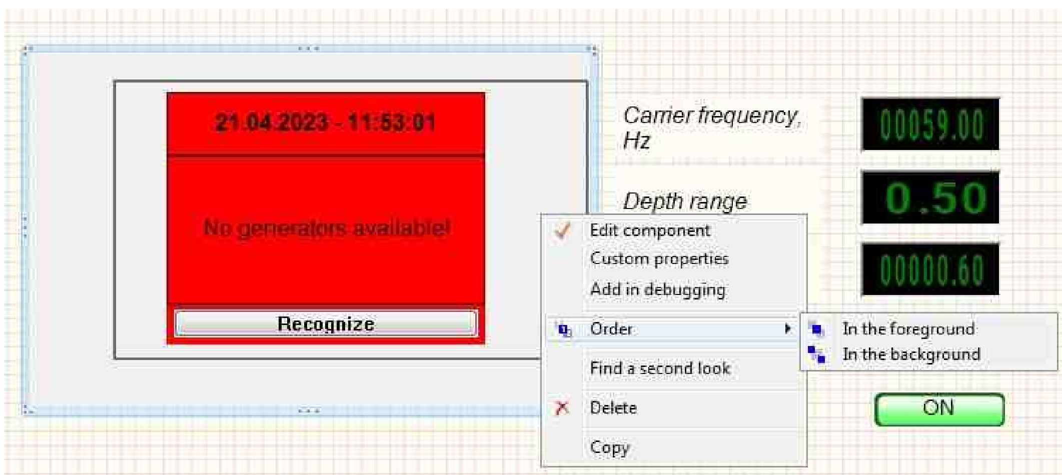
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



Mathematical description

The change in the frequency f(t) inside the impulses with LinFM occurs according to a linear law.

$$f(t) = f_0 + bt, -\frac{T_c}{2} \leq t \leq \frac{T_c}{2}$$

Where $f_0 = \frac{(F_{max}+F_{min})}{2}$ - the central value of the carrier frequency. $b = \frac{F_{max}-F_{min}}{T_c}$ - base (steepness of frequency change) LinFM signal. Tc is the duration of the signal. F_{max}, F_{min} – the maximum and minimum values of the frequency of the RF impulse signal.

The phase of the signal with LinFM is defined as:

$$\varphi(t) = 2\pi \int_0^t f(t)\,dt = 2\pi(f_0 t + \frac{b}{2}t^2)$$

Then the LinFM signal can be described by the expression:

$$S_{LFM}(t) = S_0 \cos\{\varphi_0 + \varphi(t)\} = S_0 \cos\{\varphi_0 + 2\pi(f_0 t + \frac{b}{2}t^2)\}$$

Or in a complex form:

$$S_{LFM}(t) = S_0 e^{j\{\varphi_0 + 2\pi(f_0 t + \frac{b}{2}t^2)\}}$$

Where S_0- signal amplitude; j – imaginary unit; φ_0 – initial phase.

LinFM signals are used in radar as a method of forming and processing a probing pulse. The use of the LinFM signal makes it possible to increase the accuracy of measurements in radar. Also, this type of signals has found a place in image processing, where perspective takes place (the phenomenon of apparent distortion of the proportions and shape of bodies during their visual observation. for example, two parallel rails seem to converge to a point on the horizon.). In three-dimensional radiotomography, where the use of a LinFM signal allows remote non-contact investigation of hidden objects and their internal structure.

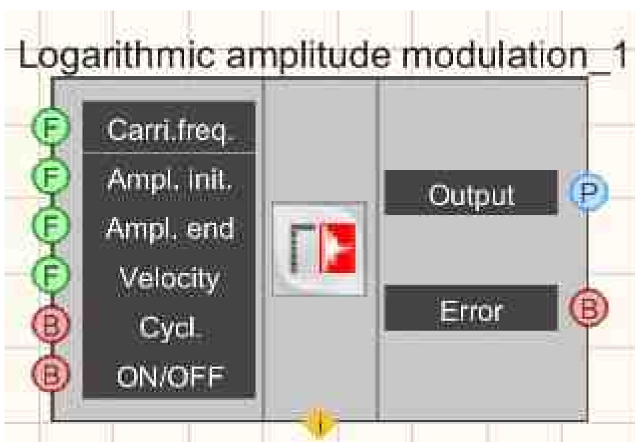
9.8.Logarithmic amplitude modulation

The component "**Logarithmic amplitude modulation**" is user to generate a signal generated by a special amplitude modulation law, in which the amplitude of the oscillations varies according to a logarithmic law in a certain specified range cyclically..

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Carri.freq. - the carrier frequency of the signal with which the LogAM will be generated, Hz.
- Ampl. init., V - the initial amplitude of the signal in which the LogAM will be generated, V.
- Ampl. end, V - the end amplitude of the signal in which the LogAM will be generated, V.
- Velocity - the rate of change of the frequency signal relative to dB per second, with which the LogAM will be generated, dB/s.
- Cycl. - number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- ON\OFF - turn ON and OFF the generator.

➢Output
- Output - the output of the generator generated by the specified parameters of the signal.

- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Carrier frequency (1000) - setting the frequency of the carrier signal, Hz.
- StartAmplitude (1) - the initial amplitude of the signal, V.
- EndEndAmplitude (1,5) - the end amplitude of the signal, V.
- Speed (1) - the rate of change of the frequency signal relative to dB per second, with which the LogAM will be generated, dB/s.
- Activate (false) - the status of the signal (enabled or disabled).
- Cycle (true) - the parameter responsible for the number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Carrier frequency - the value of the input value of the carrier frequency in which the LogAM will be generated (number), Hz
- Initial signal amplitude - the value of the input value of the initial amplitude of the signal of the frequency range in which the LogAM will be generated, V (number)
- End signal amplitude - the value of the input value of the end signal amplitude of the signal of the frequency range in which the LogAM will be generated, V (number)
- Velocity - the value for setting the rate of change of the frequency with which the LogAM will be generated, dB/s (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

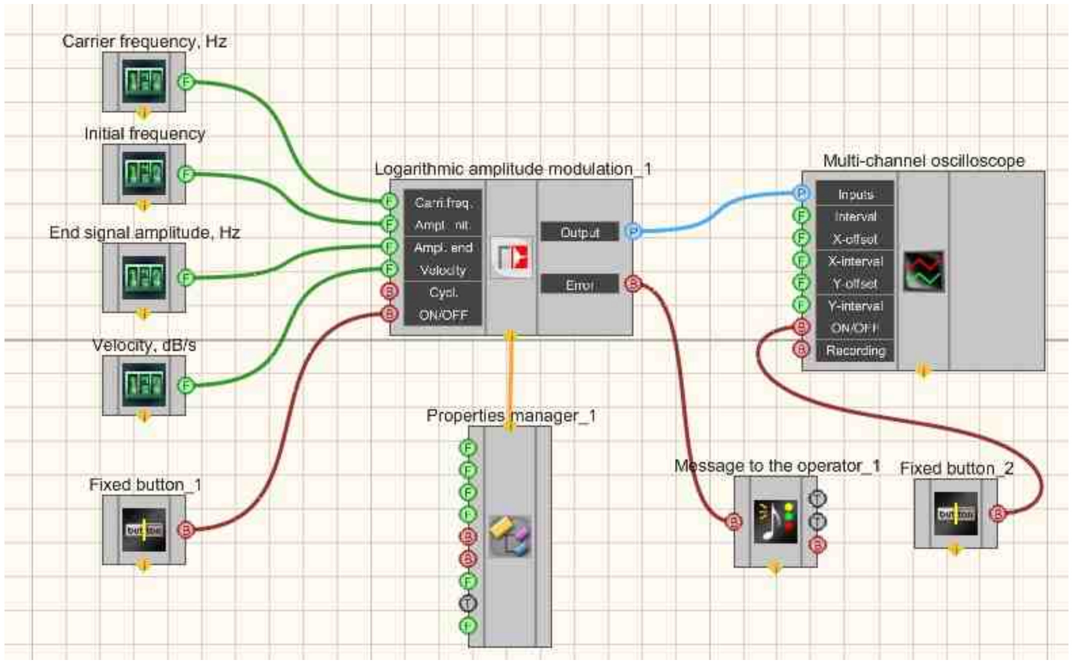
- FLOAT Carriersfrequency - Set the the carrier frequency (number), Hz
- FLOAT StartAmplitude - Set the initial amplitude of the signal (number), V.
- FLOAT EndEndAmplitude - Set the end signal amplitude of the signal (number), V.
- FLOAT Speed - Set the rate of change of the signal, dB/s (number).
- VARIANT_BOOL Activate - Set the signal presence status:
  true - the signal presence status is turned ON;
  false - the signal presence status is turned OFF.
- VARIANT_BOOL Cyclic - Set the cyclic of the signal:
- true - the signal cyclic status is turned ON;
- false - the signal cyclic status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢**Methods:**

- void SetSignal(FLOAT Carriersfrequency, FLOAT StartAmplitude, FLOAT EndAmplitude, FLOAT speed, BOOL cycle) - Set the carrier frequency, Initial signal amplitude, End signal amplitude, Velocity, Cyclicity.
- void DeleteSignal(void) - Stop the generator.

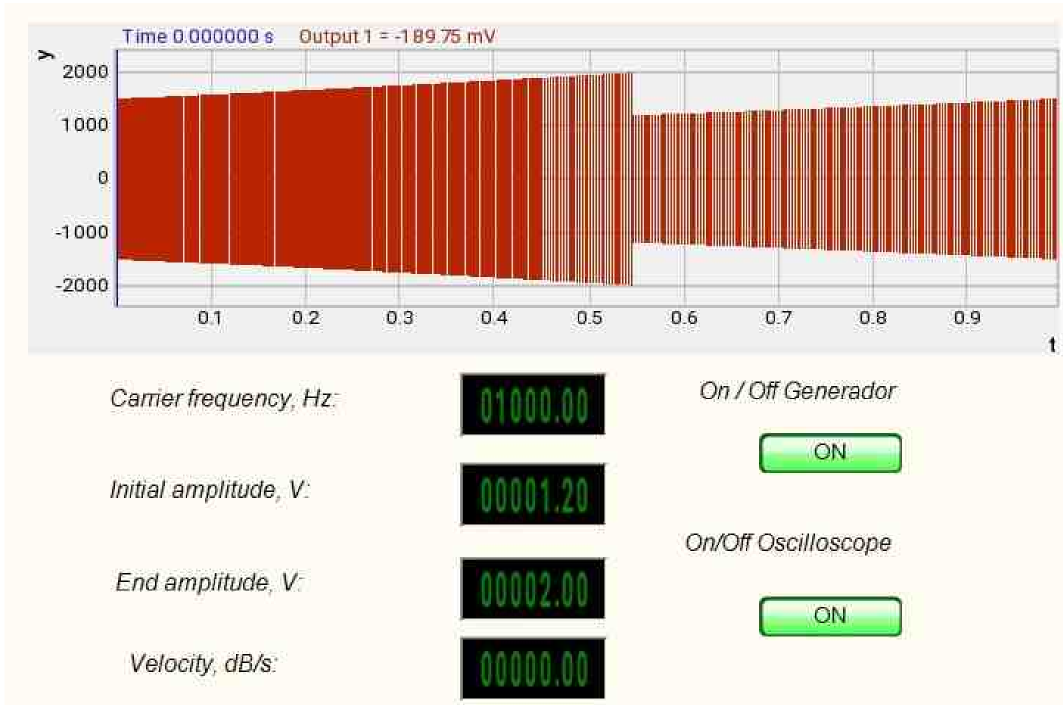
Example

Project in SCADA ZETView



This project shows the operation of the Logarithmic amplitude modulation(LogAM) 501 component generates a logarithmically-amplitude-modulated signal, the parameters of which are set using Selectors 428 (Carrier frequency, Initial amplitude, End amplitude, Velocity, Cyclicity). A Multi-channel oscilloscope 928 is used to graphically display the signal. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

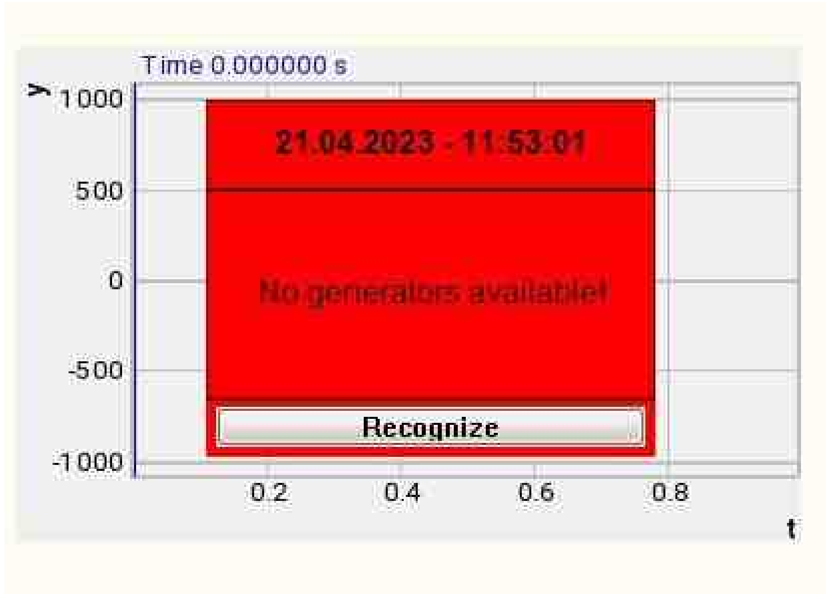
Project operation results



View the example in ZetView

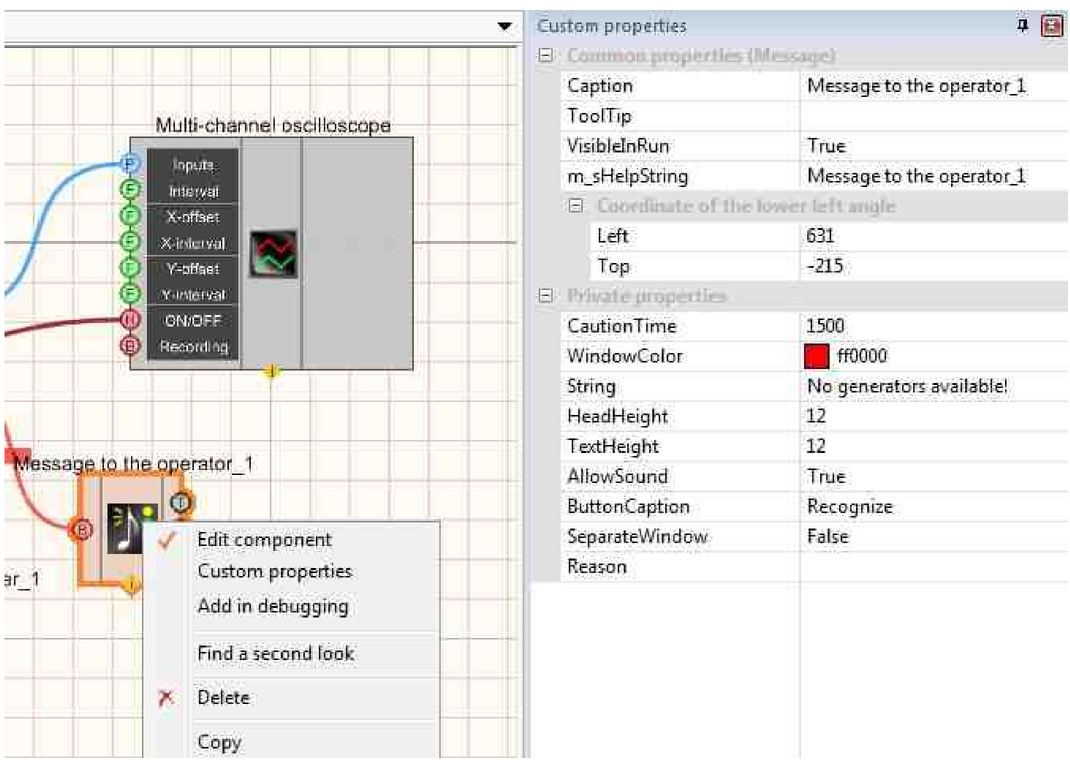
Output about an error in the project

The component Message to the operator [1052] is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal [530]. In addition, the message Sine signal # 1 [530] connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

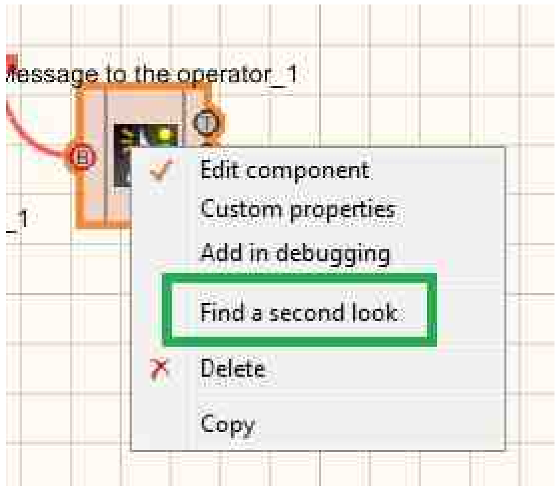


Message about the error to the operator.

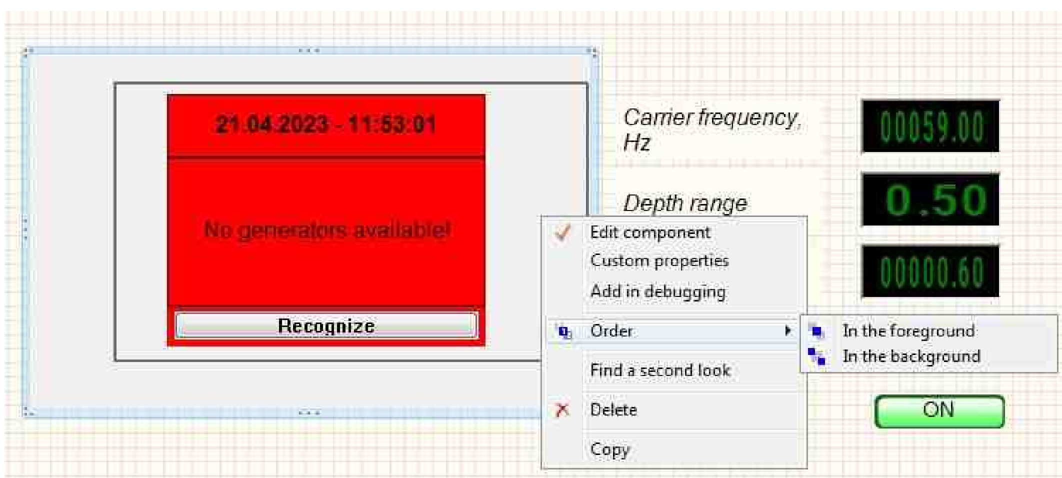
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



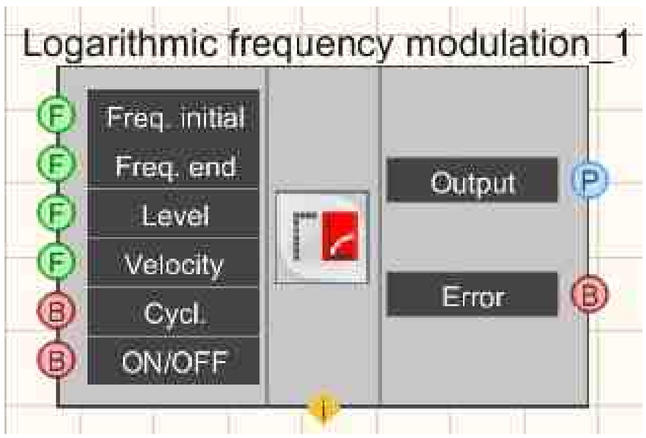
9.9.Logarithmic frequency modulation

The component "**Logarithmic frequency modulation**" is user to generate a signal generated by a special frequency modulation law, at which the oscillation frequency changes according to a logarithmic law in a certain specified range cyclically.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Freq. initial - the initial frequency of the signal of the frequency range in which the LogFM will be generated, Hz.
- Freq. end - the end frequency of the signal of the frequency range in which the LogFM will be generated, Hz.
- Level - the level of the root-mean-square value (RMS) with which the LogFM will be generated, V.
- Velocity - the rate of change of the frequency signal relative to octaves per minute, with which the LogFM will be generated, okt/min.
- Cycl. - number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- ON\OFF - turn ON and OFF the generator.

➢Output

- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- StartFrequency (0) - the initial frequency of the signal of the frequency range in which the Log will be generated, Hz.
- EndFrequency (1000) - the end frequency of the signal of the frequency range in which the LogFM will be generated, Hz.
- Level (1) - the level of the root-mean-square value (RMS) with which the LogFM will be generated, V.
- Speed (1) - the rate of change of the frequency signal relative to octaves per minute, with which the LogFM will be generated, okt/min.
- Activate (false) - the status of the signal (enabled or disabled).
- Cyclic (true) - the parameter responsible for the number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Initial signal frequency - the value of the input value of the initial frequency of the signal of the frequency range in which the LogFM will be generated, Hz (number)
- End signal frequency - the value of the input value of the end frequency of the frequency range in which the LogFM will be generated, Hz (number)
- Level - the value for setting the root-mean-square value (RMS) of the level with which the LogFM will be generated, V (number).
- Velocity - the value for setting the rate of change of the frequency with which the LogFM will be generated, okt/min (number).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

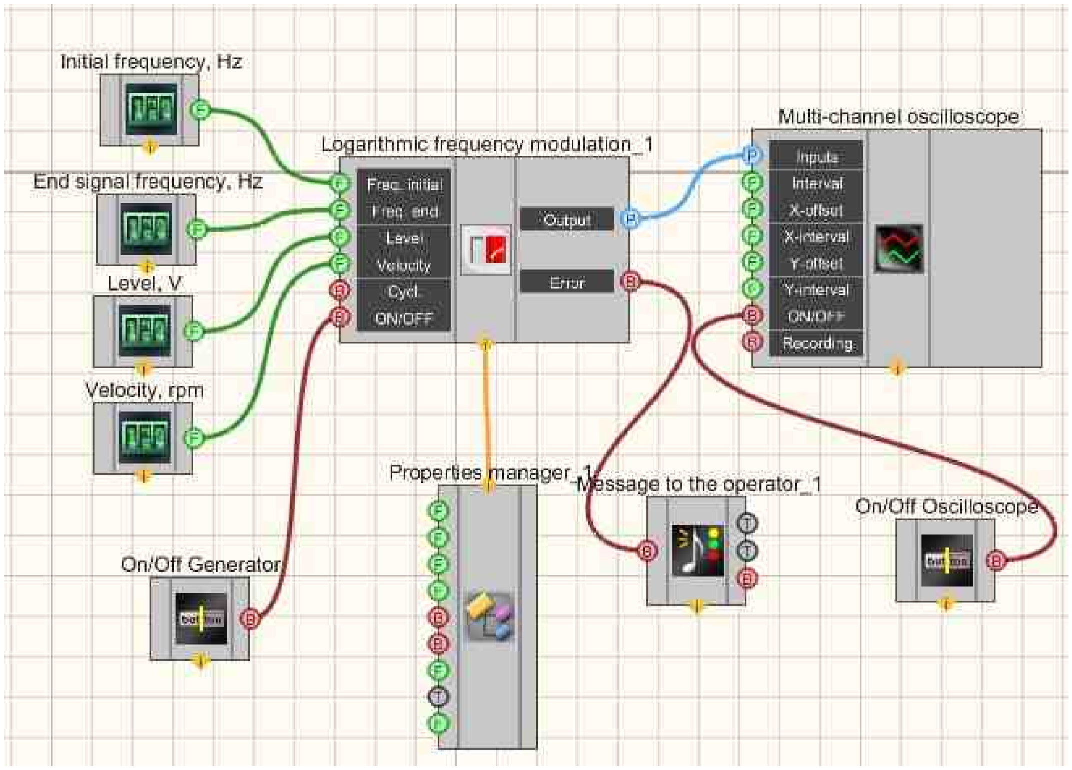
- FLOAT StartFrequency - Set the initial signal frequency of the signal (from 0.01 Hz to the end frequency of the signal of logarithmic frequency modulation), Hz.
- FLOAT EndFrequency - Set the end signal frequency of the signal (from the initial frequency of logarithmic frequency modulation to (DAC sampling frequency / 2)), Hz.
- FLOAT Level - Set the signal level (from 0 V to the maximum limit DAC signal level), V.
- FLOAT Speed - Set the rate of change of the signal, okt/min (string).
- VARIANT_BOOL Activate - Set the signal presence status:
  true - the signal presence status is turned ON;
  false - the signal presence status is turned OFF.
- VARIANT_BOOL Cyclic - Set the cyclic of the signal:
  true - the signal cyclic status is turned ON;
  false - the signal cyclic status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢Methods:

- void SetSignal(FLOAT StartFrequency, FLOAT EndFrequency, FLOAT level, FLOAT speed, BOOL cycle) - Set the initial signal frequency, the end signal frequency, the level, the velocity, the cyclicity.
- void DeleteSignal(void) - Stop the generator.

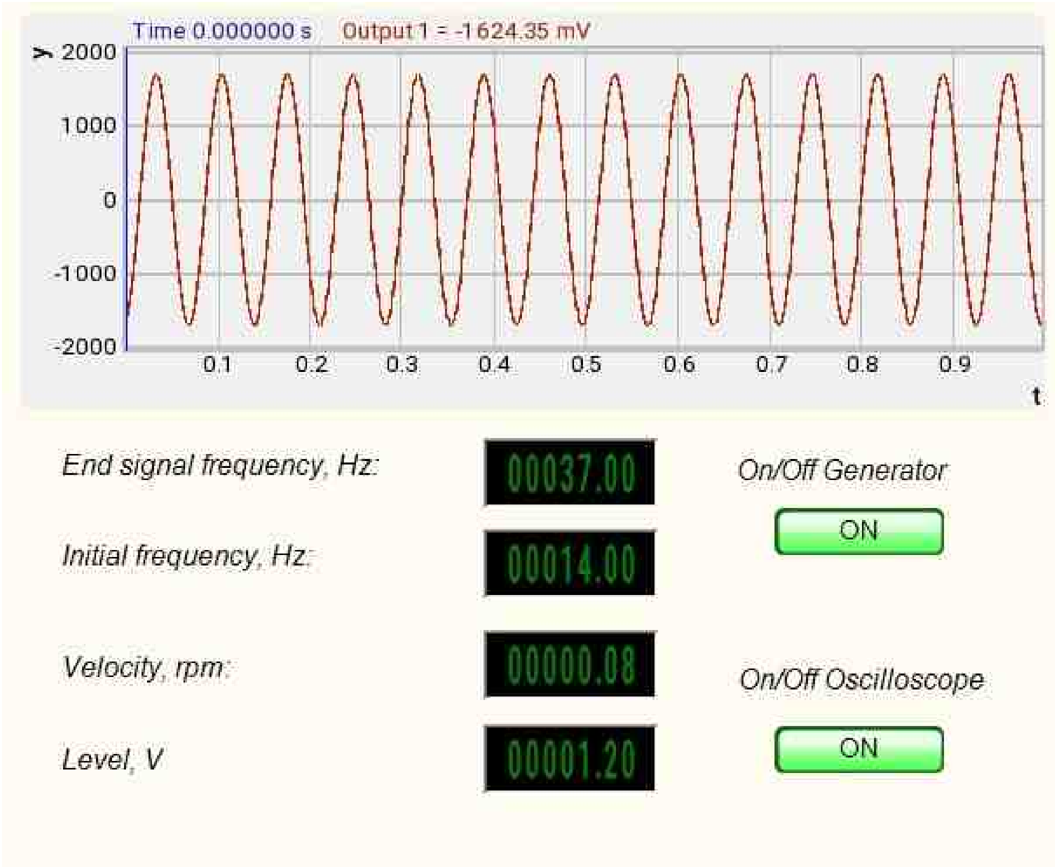
Example

Project in SCADA ZETView



This project shows the operation of the Logarithmic frequency modulation (LogFM) 501 component generates a logarithmic-frequency-modulated signal, the parameters of which are set using Selectors 428 (initial frequency, final frequency, level, velocity). A Multi-channel oscilloscope 928 is used to graphically display the signal. The Fixed button 691 is used to turn on or off.
The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

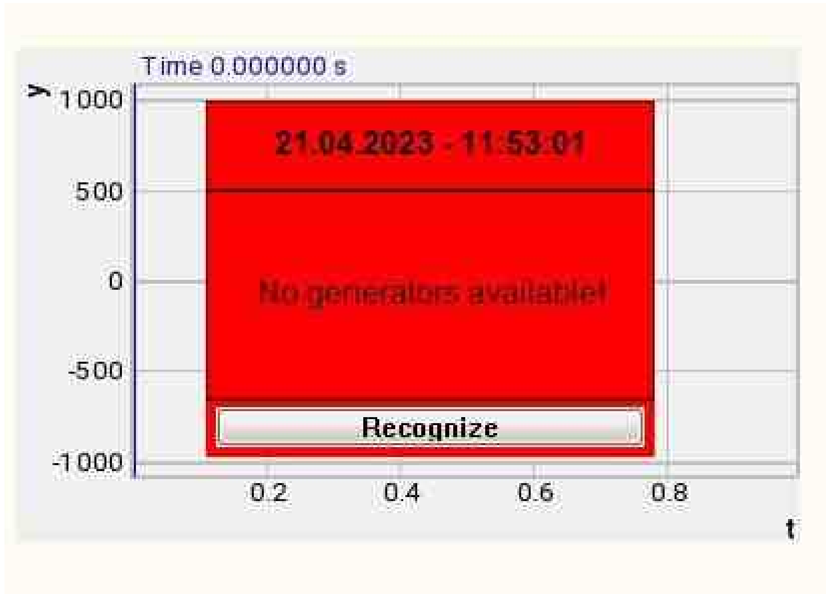
Project operation results



View the example in ZetView

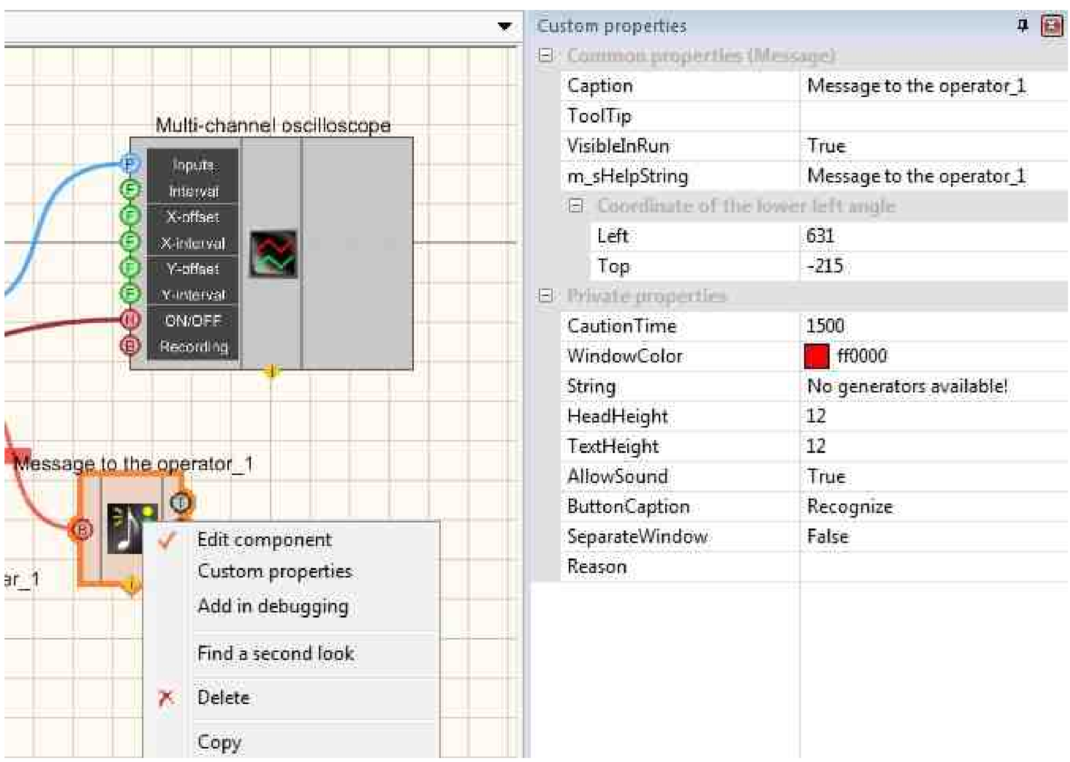
Output about an error in the project

The component [Message to the operator](1052) is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the [Sine signal](530). In addition, the message [Sine signal # 1](530) connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

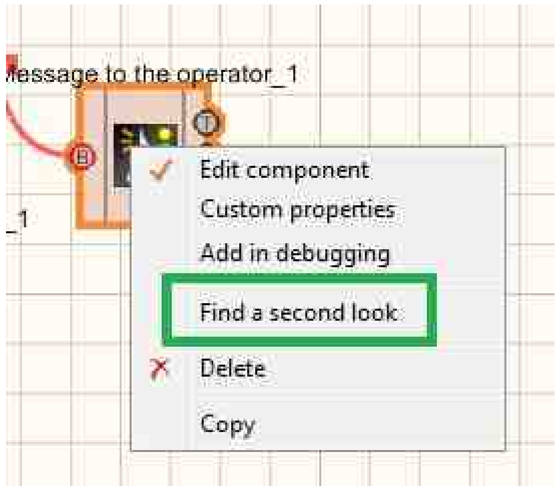


Message about the error to the operator.

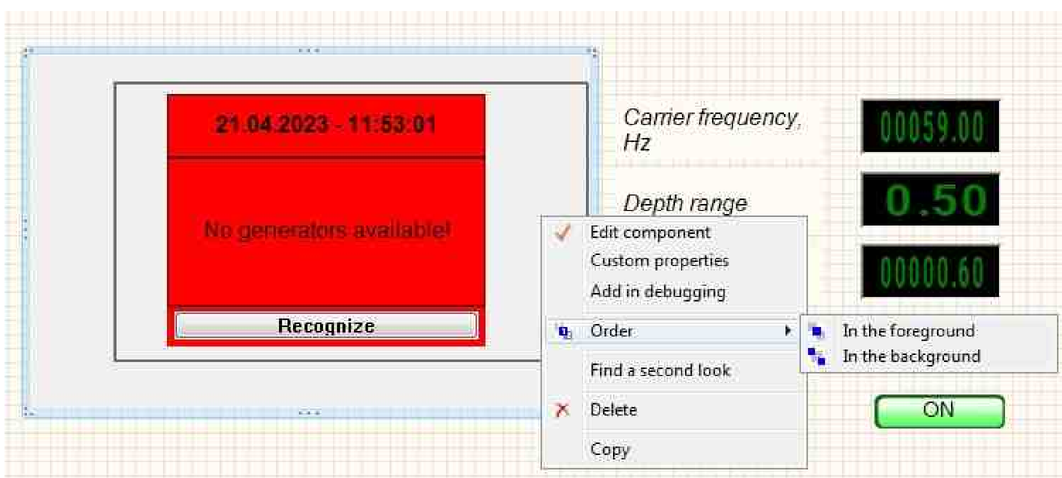
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



Mathematical description

Frequency-modulated signal with logarithmic frequency sweep (LogFM).

A frequency-modulated signal with a logarithmic frequency sweep (LogFM) is a sinusoid with a logarithmically increasing frequency over time.

LogFM is calculated by the formula:

$$A = A_0 \sin\left(\frac{\omega T}{\ln\left(\frac{f_k}{f_0}\right)}\left(\frac{f_k}{f_0}\right)^{\frac{t}{T}} + \varphi_0\right)$$

Where $A_0 = \frac{2V_{rms}}{\sqrt{2}}$ - signal phase, φ_0 – initial phase of the signal, T - the time of frequency change, t – current time, f_0 – initial signal frequency, f_k – the end frequency of the signal..

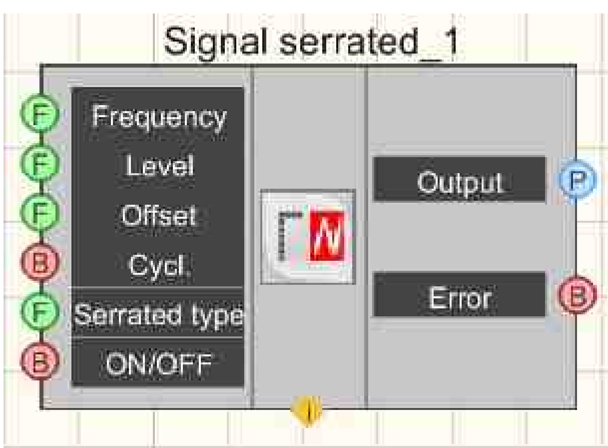
9.10.Signal serrated

The component "**Signal serrated**" is user to generate a signal serrated signal of a certain type according to the specified parameters: amplitude, frequency and offset of the constant component.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

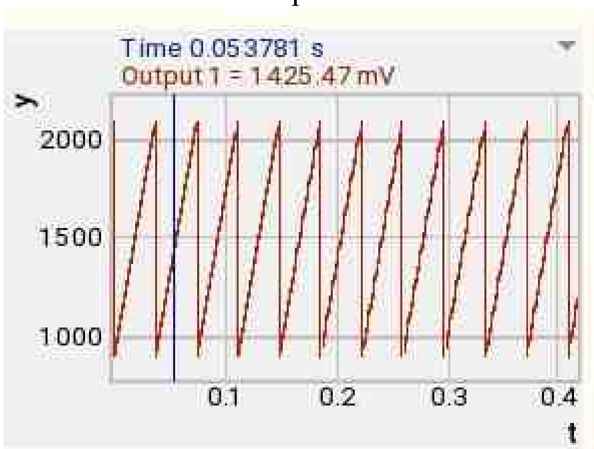
Setting:

➢Input:

- Frequency - the frequency of the signal serrated, Hz.
- Level - the level of the signal serrated, V.
- Offset - the offset of the constant component of the signal serrated, V.
- Cycl. - number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- Serrated type - signal serrated waveform type: Upward, Downward, Triangular. The pictures below show all 3 types.

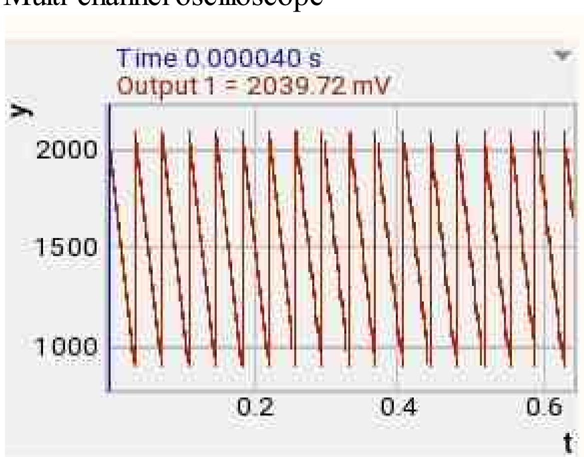
1. Upward of serrated type is characterized by a rising interval.

Multi-channel oscilloscope



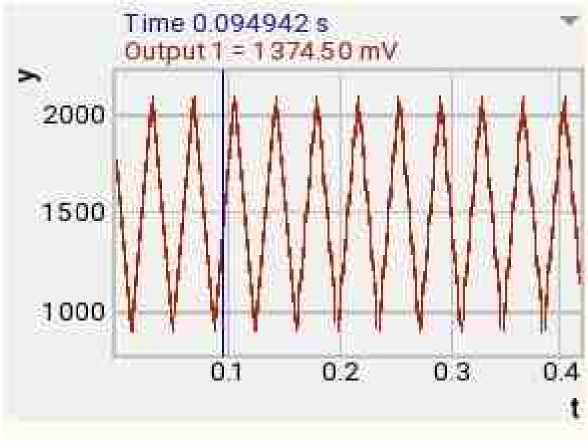
Increase of serrated (Upward)

Multi-channel oscilloscope



Downward of serrated (Downward)

Multi-channel oscilloscope



Triangular of serrated (Triangular)

- ON\OFF - turn on and off the generator.

➢Output
- Output - the channel of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Frequency (100) - output signal frequency, Hz.
- Level (1) - output signal amplitude, V.
- Shift (0) - offset of the constant component, V.
- Activate (false) - the status of the signal (enabled or disabled).
- Cyclic (true) - the parameter responsible for the number of play recorded signal. Either continuous play recorded signal in a cycle or its one-time play recorded signal.
- pilatype (Upward) - output signal type (Upward, Downward, Triangular).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of *the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.*

Setting:

➢**Input:**
- Frequency - Frequency input value, Hz (number).
- Level - Input signal amplitude value, V (number).
- Offset - Offset value of the constant component of the input signal, V (number).
- Signal serrated type - signal serrated waveform type::
  0 – Upward;
  1 – Downward;
  2 – Triangular.

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

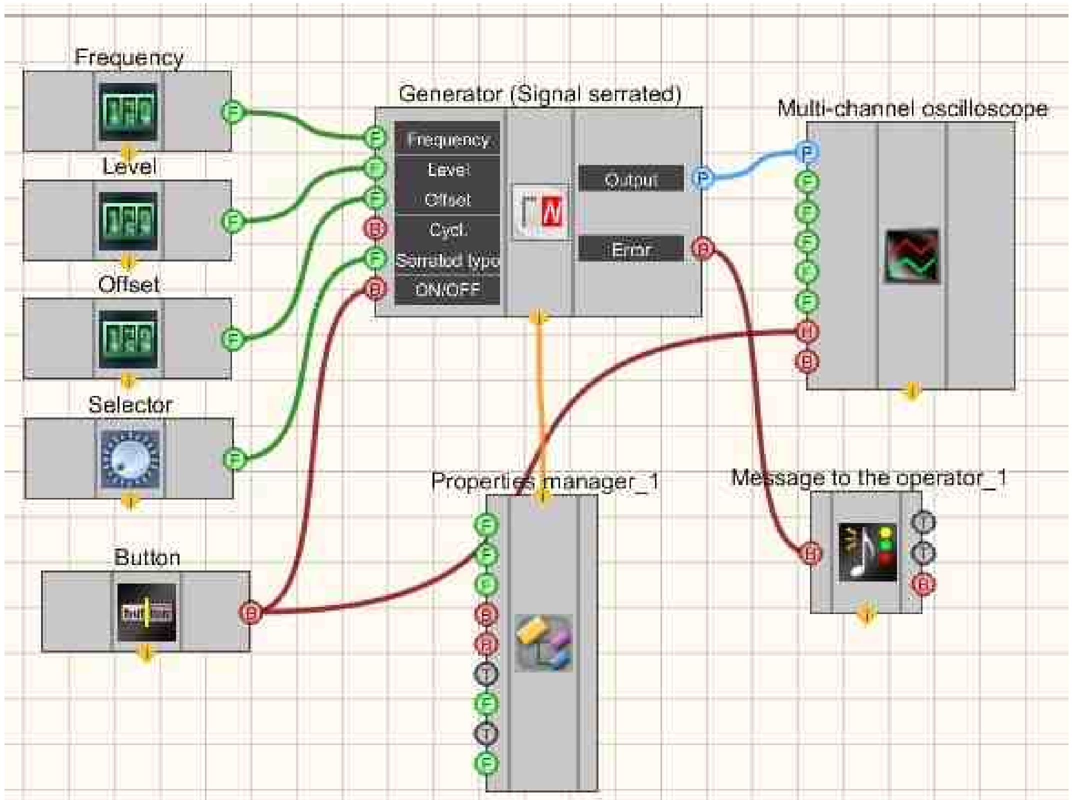
- FLOAT Frequency - Set the signal frequency (from 0.01 Hz to (DAC sampling frequency / 2)), Hz
- FLOAT Level - Set the signal level (from 0 V to the maximum limit DAC signal level), V
- FLOAT Shift - Set the DC offset (from minus the maximum limit DAC signal level to the maximum limit DAC signal level), V.
- VARIANT_BOOL Activate - Set the signal presence status:
  true - the signal presence status is turned ON;
  false - the signal presence status is turned OFF.
- VARIANT_BOOL Cyclic - Set the cyclic of the signal:
  true - the signal cyclic status is turned ON;
  false - the signal cyclic status is turned OFF.
- BSTR PilaType - Set the signal serrated type: Upward; Downward; Triangular (string).
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢**Methods:**

- void SetSignal(FLOAT frequency, FLOAT level, FLOAT smeshenie, BOOL bCycle, LONG nPilaType) - Set the frequency, level and offset, cyclic and signal serrated type.
- void DeleteSignal(void) - Stop the generator.

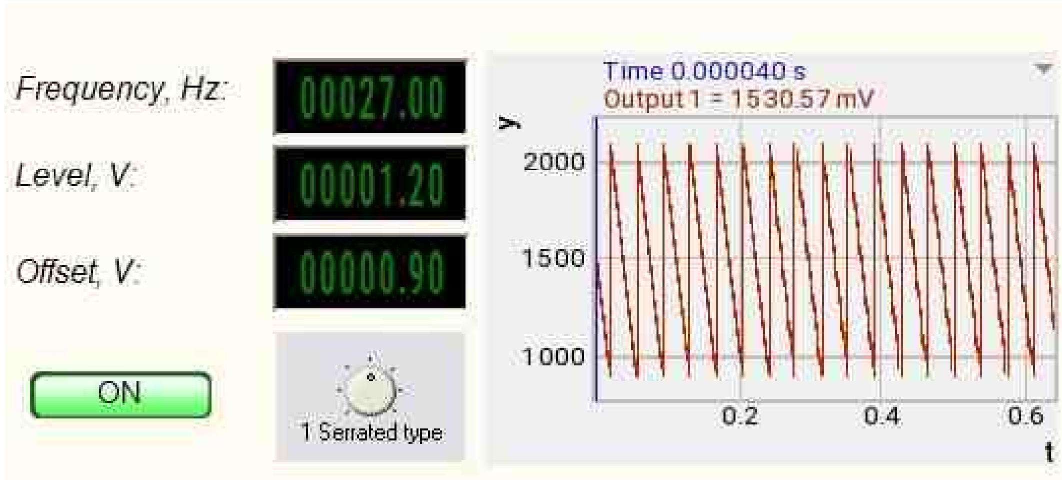
Example

Project in SCADA ZETView



This project shows the operation of the Signal serrated 515 component creates a signal serrated, with parameters that are set using Selectors 428. To view the generated signal, use the Multi-channel oscilloscope 928 component. An additional Selector 428 is used to set the signal serrated type (Upward; Downward; Triangular). The result of the compiled program can be seen in the operator interface. The Fixed button 691 is used to turn on or off.
The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

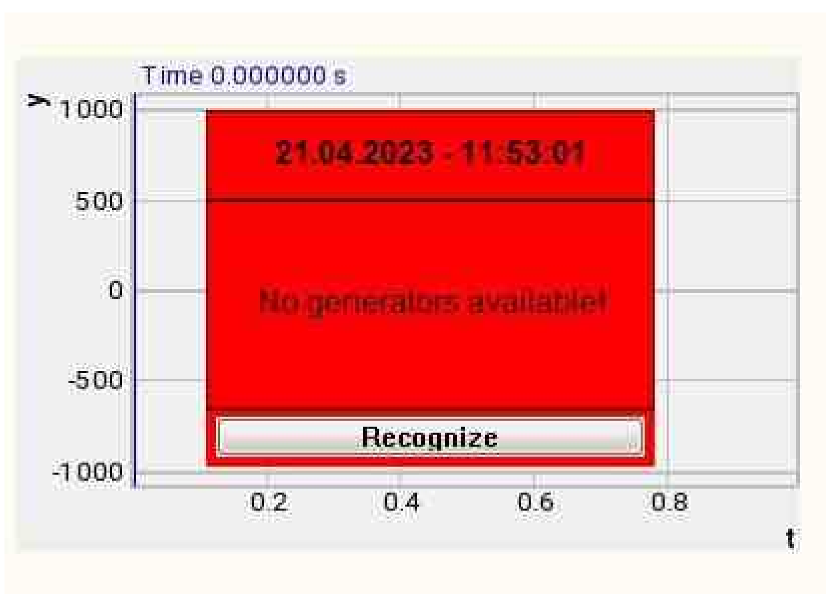
Project operation results



View the example in ZetView

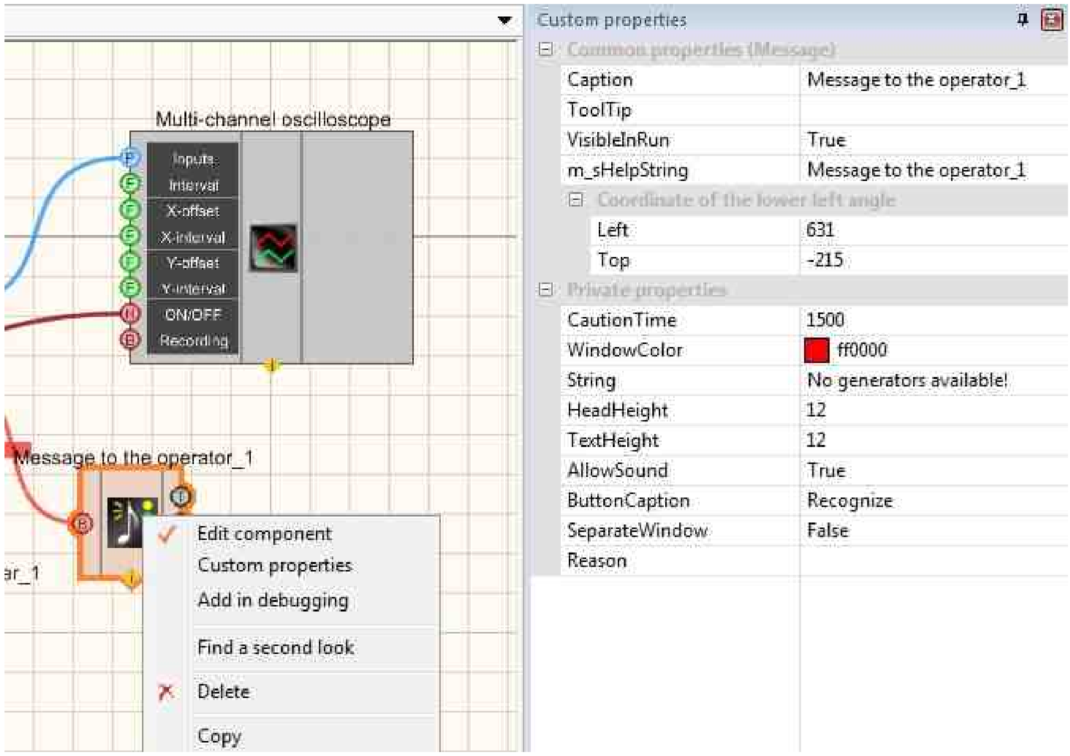
Output about an error in the project

The component Message to the operator 1052 is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal 530. In addition, the message Sine signal # 1 530 connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

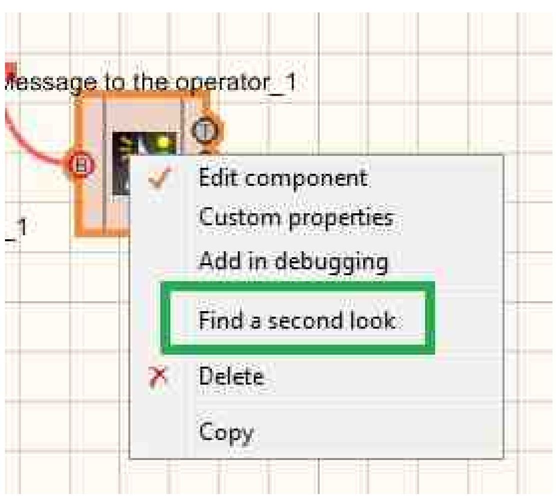


Message about the error to the operator.

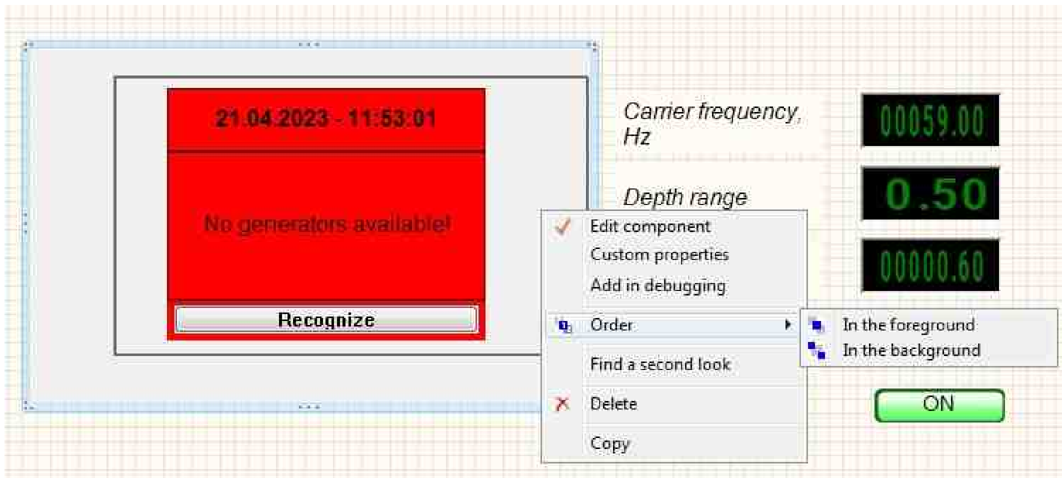
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



Mathematical description

Periodic signals are signals that repeat themselves at certain time intervals T. The value of T is called the period of the signal f(t).

Any periodic signal with period T can be represented as a sum of harmonic signals. In this case, the lowest (fundamental) frequency will be equal to 1/T. All other frequencies of the signal components (harmonics) are the products of this frequency by integers.

If the signal f(t) is periodic with period T, then it can be represented as a Fourier series:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)]$$

Where ω is called the fundamental (angular) frequency of the signal:

$$\omega = \frac{2\pi}{T} = 2\pi f.$$

Fourier coefficients a_n or b_n are defined by the following expressions:

$$a_n = \frac{2}{T}\int_0^T f(t)\cos(n\omega t)dt$$

$$b_n = \frac{2}{T}\int_0^T f(t)\sin(n\omega t)dt$$

At n=0, 1, 2, …

- Value

  $$\frac{a_0}{2} = \frac{1}{T}\int_0^T f(t)dt$$

  is the average value of the signal for one period, i.e. the constant component of the signal. Note that b_0 is always zero.

- The trigonometric representation of the Fourier series depends on the value of the signal at the selected initial time t=0.

Skew-symmetric triangle signal, no DC component:

$$f(t) = A\frac{2}{\pi}\sum_{k=1}^{\infty}\frac{\sin(2\pi kft)}{k} = A\frac{2}{\pi}[\sin(\omega t) + \frac{1}{2}\sin(2\omega t) + \frac{1}{3}\sin(3\omega t) + \cdots]$$

Such signals are widely used in communication equipment, television, and radar. Most often they are used to create a time base of a beam in cathode ray tubes of oscilloscopes, televisions, etc. Another important area of application of signal serrated voltage is the conversion of voltage to a time interval in impulse-phase modulation devices, when comparing currents and voltages, and when replacing voltage with digital code, etc.

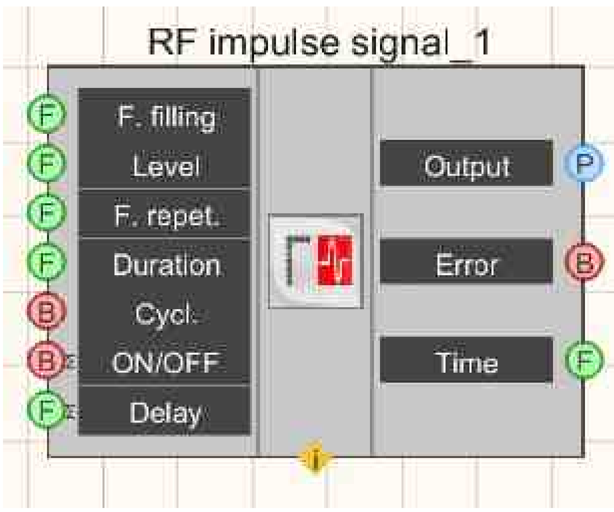
9.11.RF impulse signal

The component "**RF impulse signal**" is user to generate RF impulses (a RF impulse is a time-limited Sine signal) with a given duty cycle, level, repetition frequency and duration.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- F. filling - the filling frequency of the RF impulse signal, Hz.
- Level - RF impulse signal level, V.
- F. repet. - repetition frequency of the RF impulse signal, Hz.
- Duration - the duration of the RF impulse signal in periods (number of periods).
- Cycl. - the number parameter of signal play records:
  1. True - constant generation of RF impulse.
  2. False - generation of a single RF impulse.
- ON\OFF - turn on and off the generator.
- Delay - Signal delay value, s.

➢Output
- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.
- Time - displays the time of the next portion.

Custom common properties:
➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FillFrequency (100) - the signal filling frequency, Hz.
- Level (1) - the amplitude of the output signal, V.
- FollowFrequency (1) - the signal repetition frequency, Hz.
- Duration (1) - the signal duration in periods (number of periods).
- Cyclic (true) - the number of times of signal play records (continuous signal generation mode/single impulse).
- Tukey (false) - apply the Tukey window.
- Activate (false) - the status of the signal (enabled or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Filling frequency - The value of the input value of the frequency of the RF impulse within the packet, Hz (number).
- Level - The value of the amplitude of the input signal, V (number).
- Repetition frequency - The value of the input value is the frequency of occurrence of RF impulse, Hz (number).
- Duration - The value of the duration of the RF impulse signal in periods (number).
- Signal delay, s - Signal delay value, s (number).

➢**Custom common properties:**

- Are available by the link below. 154

Custom private properties (the default value is shown in parentheses):

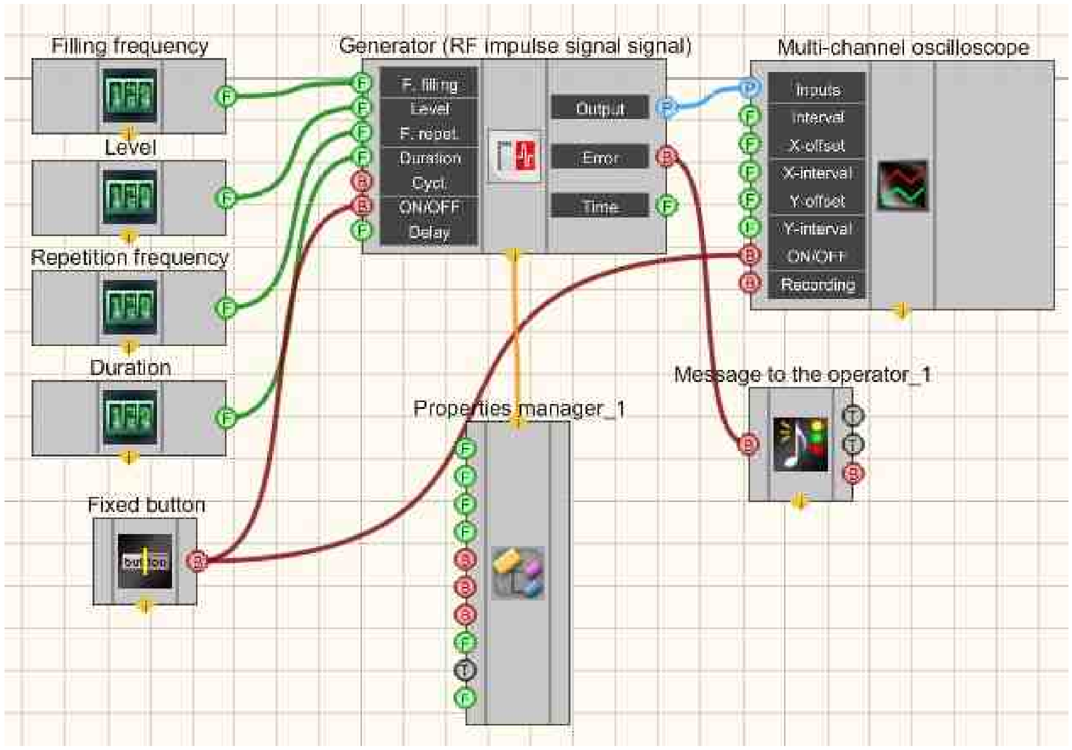
- FLOAT FillFrequency - Set the signal filling frequency (from the repetition frequency of the RF impulse to (sampling frequency of the DAC / 2)), Hz.
- FLOAT Level - Set the signal level (from 0 V to the maximum limit DAC signal level), V.
- FLOAT FollowFrequency - Set the signal repetition frequency (from 0.01 Hz to the frequency of filling the RF impulse signal), Hz.
- FLOAT Duration - Set the duration of the signal in periods (from 0 to (frequency of filling the RF impulse signal / repetition frequency of the RF impulse signal)).
- VARIANT_BOOL Cyclic - Set the cyclic of the signal:
  true - the signal cyclic status is turned ON;
  false - the signal cyclic status is turned OFF.
- VARIANT_BOOL Tukey - Installation apply Tukey window:
  true - status apply Tukey window is turned ON;
  false - status apply Tukey window is turned OFF.
- VARIANT_BOOL Activate - Set the signal presence status:
  true - the signal presence status is turned ON;
  false - the signal presence status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➢**Methods:**

- void SetSignal(FLOAT fillfrequency, FLOAT level, FLOAT followfrequency, FLOAT duration, BOOL cycle) - Set the fill frequency, level, repetition frequency, duration, cyclic.
- void DeleteSignal(void) - Stop the generator.

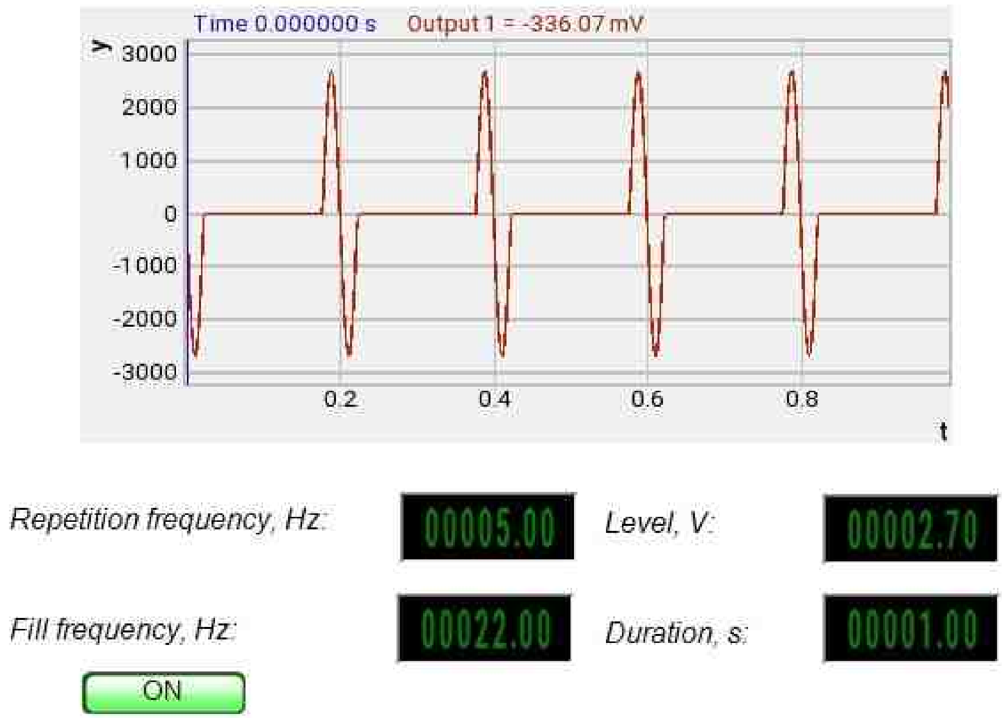
Example

Project in SCADA ZETView



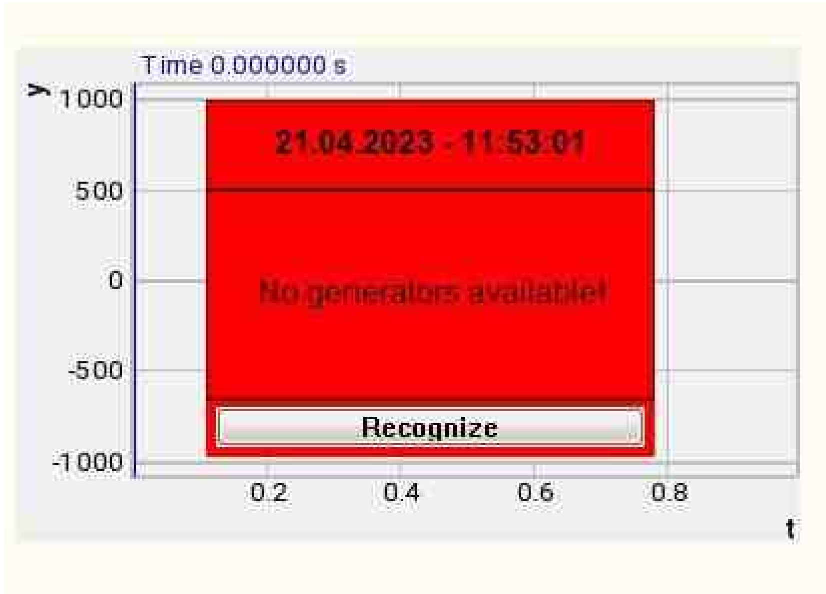
This project shows the operation of the RF impulse signal 523 component creates a signal serrated, with parameters that are set using Selectors 428. To view the generated signal, use the Multi-channel oscilloscope 928 component. The result of the compiled program can be seen in the operator interface. The Fixed button 691 is used to turn on or off.
The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a multi-channel oscilloscope.

Project operation results



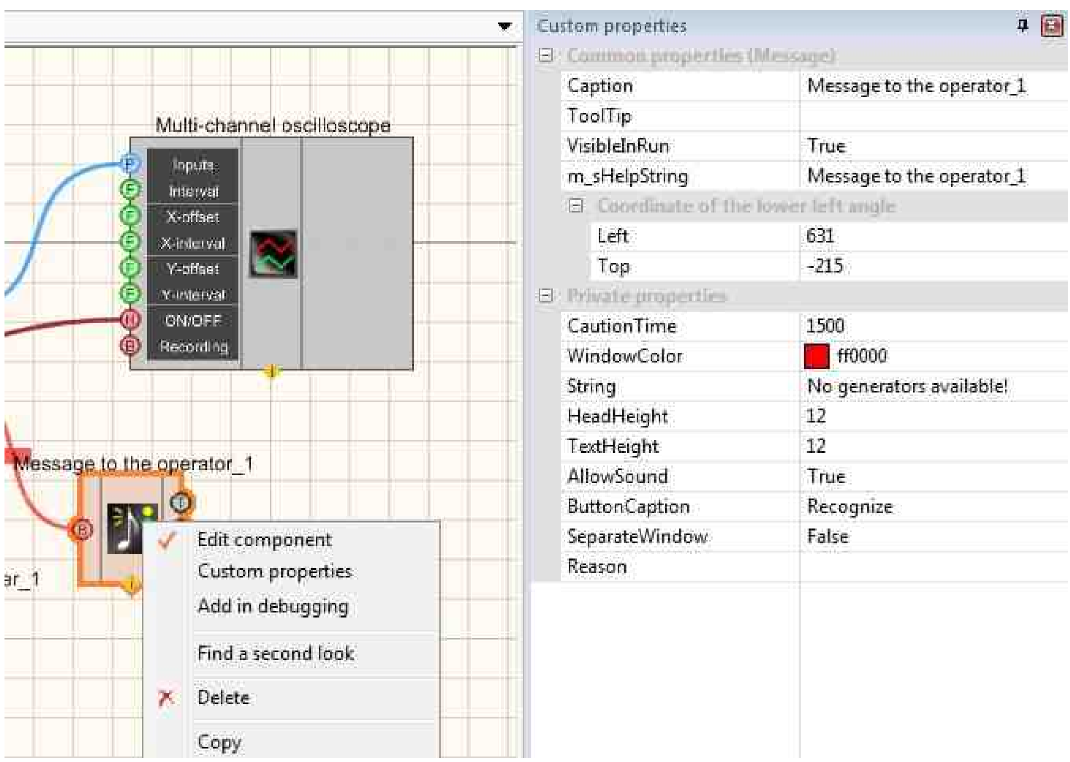
Output about an error in the project

The component Message to the operator[1052] is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal[530]. In addition, the message Sine signal # 1[530] connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

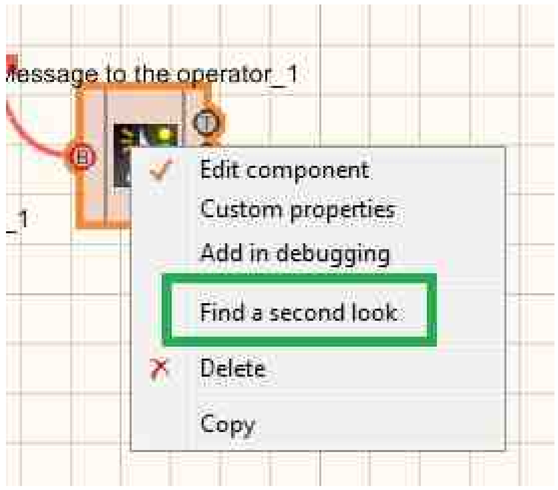


Message about the error to the operator.

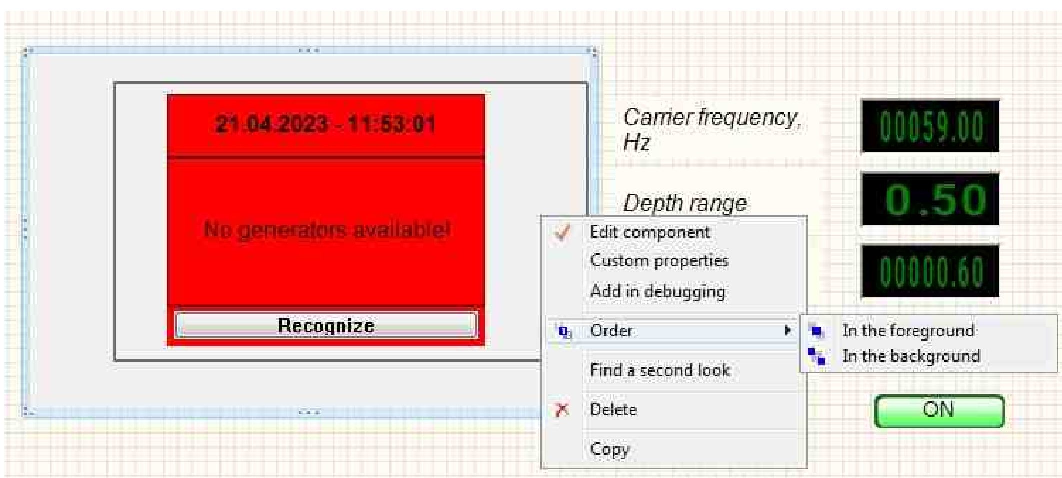
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



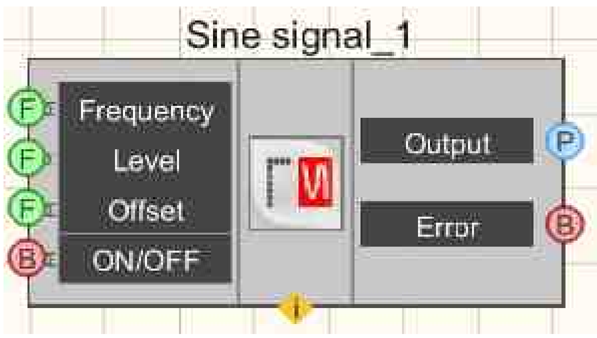
9.12.Sine signal

The component "**Sine signal**" is user to generate a harmonic signal according to the specified parameters: amplitude, frequency and DC offset.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Frequency - frequency of the Sine signal, Hz
- Level - the level of the Sine signal, V.
- Offset - the offset of the constant component of the Sine signal, V.
- ON\OFF - turn on and off the generator.

➤Output
- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:
➤Total properties (environment):
- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):
- Frequency (100) - the frequency of the Sine signal, Hz.
- Level (1) - the amplitude of the Sine signal, V.
- Shift (0) - the offset of the constant component, V.
- Activate (false) - the status of the signal (enabled or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of

the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.

- SinusNumber (1) - the number of the Sine signal: 1; 2, etc.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Frequency - The value of the input frequency value, Hz (number).
- Level - The value of the input signal amplitude, V (number).
- Offset - The value of the offset of the constant component of the input signal, V (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

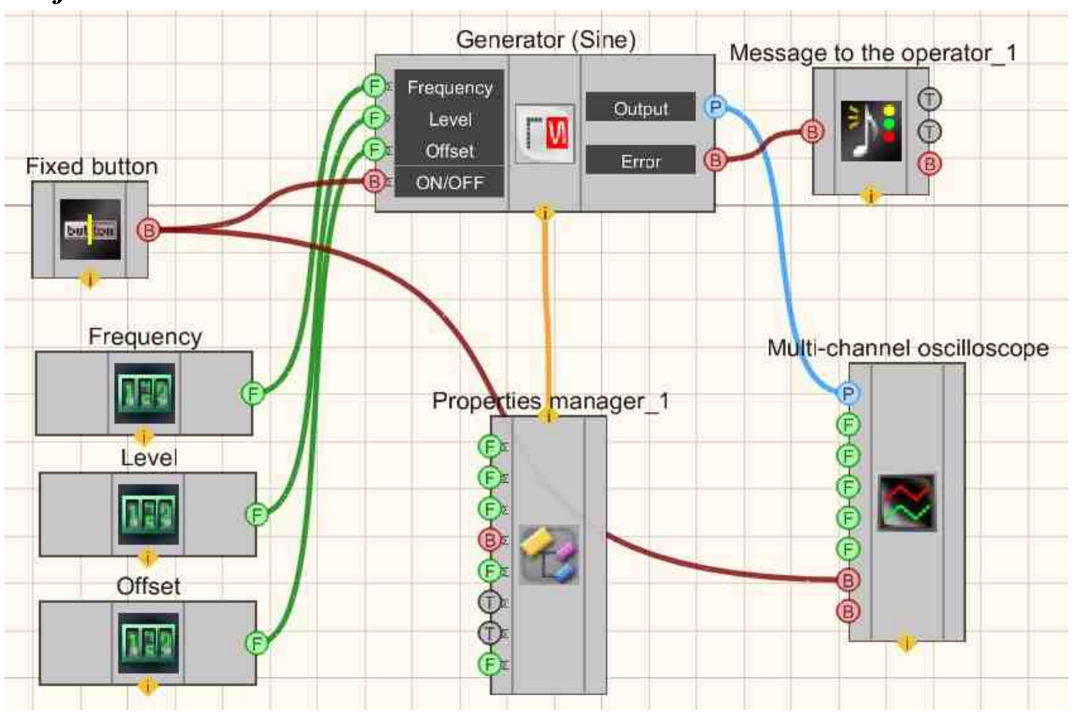
- FLOAT Frequency - Set the signal frequency (from 0.01 to (DACsampling frequency / 2)), Hz.
- FLOAT Level - Set the signal level (from 0 V to the maximum permissible DAC signal level), Hz
- FLOAT Shift - Set the offset of the constant component of the signal (from minus the maximum limit level of the DAC signal to the maximum limit level of the DAC signal), V.
- VARIANT_BOOL Activate - Set the signal presence status:
  true - the signal presence status is turned ON;
  false - the signal presence status is turned OFF.
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

➢**Methods:**

- void SetSignal(FLOAT frequency, FLOAT level, FLOAT smeshenie) - Set the frequency, level, and offset.
- void DeleteSignal(void) - Stop the generator.

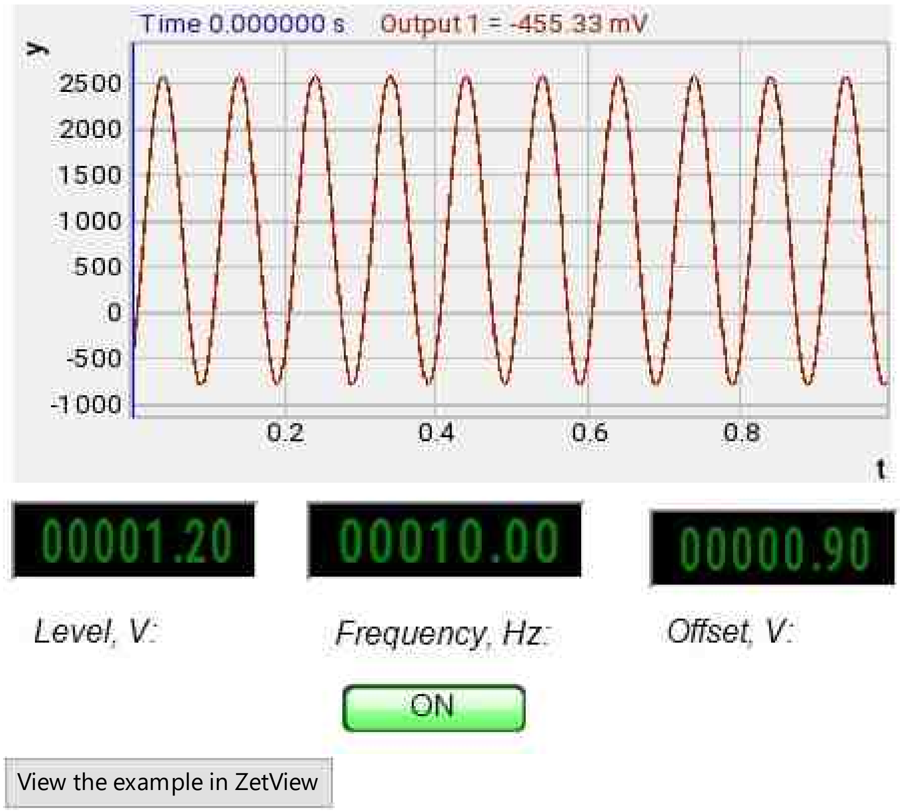
Example

Project in SCADA ZETView



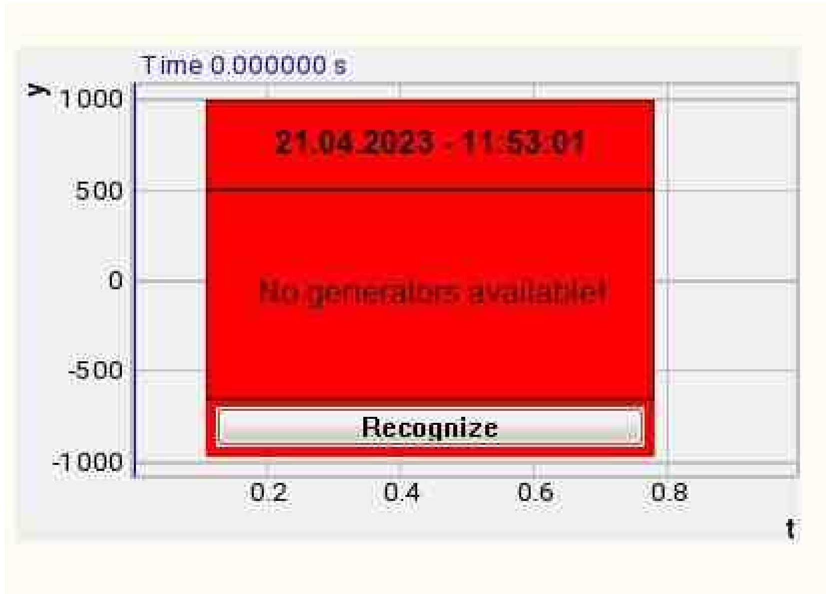
This project shows the operation of the Sine signal 530 component creates a Sine signal, with parameters that are set using Selectors 428. To view the generated signal, use the Multi-channel oscilloscope 928 component. The result of the compiled program can be seen in the operator interface. The Fixed button 691 is used to turn on or off. The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a Multi-channel oscilloscope.

Project operation results



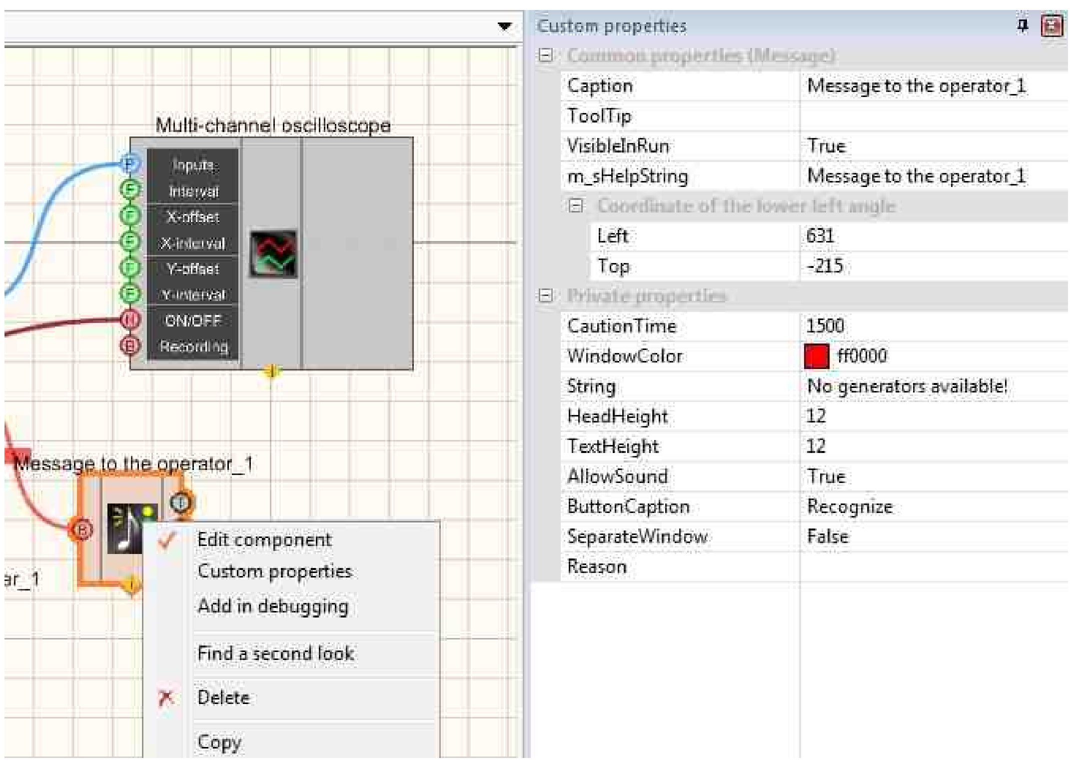
Output about an error in the project

The component Message to the operator 1052 is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal 530. In addition, the message Sine signal # 1 530 connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

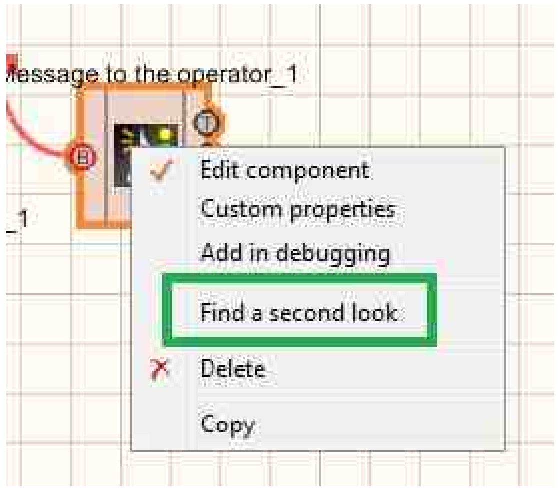


Message about the error to the operator.

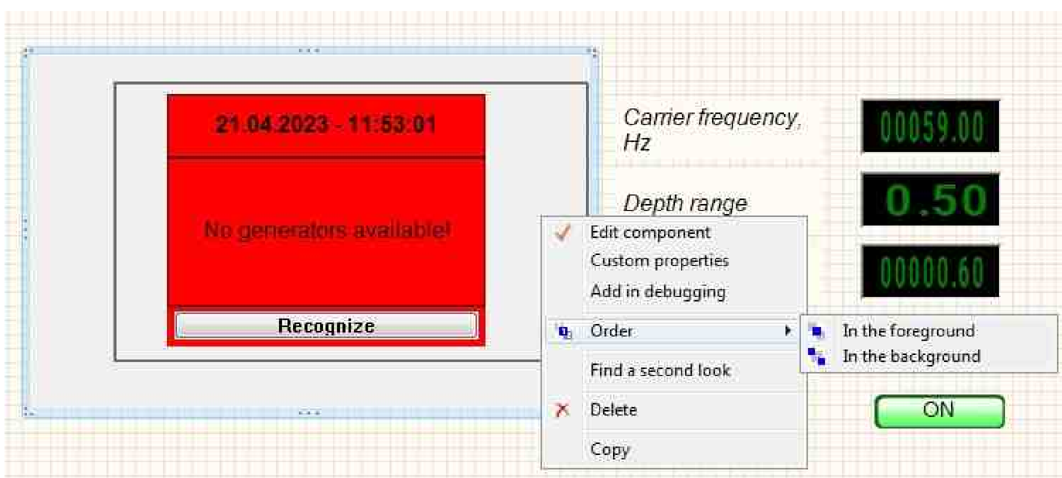
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



Mathematical description

The sine and cosine functions are most often used to describe AC circuits. The Sine function has the following form:

$$A(t) = A_m \sin(\omega t + \varphi_0)$$

Value A(t) is called the instantaneous or actual value of the function, ω - angular frequency, φ_0 - phase shift. The Sine is a periodic function with a period equal to 2π.

The period T is the time interval between two identical values of the function. The frequency f of the sine function is the reciprocal of the period:

$$T = \frac{2\pi}{\omega};\ f = \frac{1}{T};\ \omega = 2\pi f$$

The cosine function is similar to the sine function. It can be written as:

$$A = A_m \cos\varphi$$

The functions of the sine and cosine are related by the following ratios:

$$\sin\varphi = \cos\left(\frac{\pi}{2} - \varphi\right)$$

$$\cos\varphi = \sin\left(\frac{\pi}{2} + \varphi\right)$$

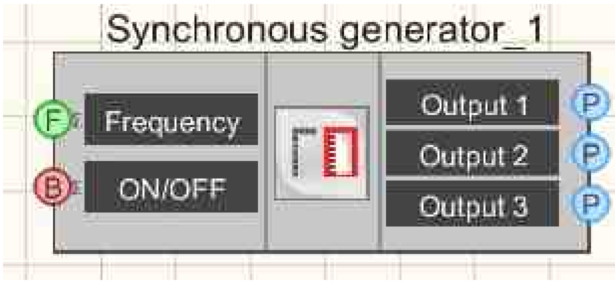
9.13.Synchronous generator

The component "**Synchronous generator**" is user to generate a signal with any number of channels (up to 20) with a given frequency of the periodic signal.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Frequency - frequency of the periodic signal, Hz
- ON\OFF - control of the status of operation of the synchronous generator.

➢Output
- Output 1 - channel 1 of the generator generated by the specified parameters of the harmonic signal.
- ...
- Output N - channel N of the generator generated by the specified parameters of the harmonic signal.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Quantity (1) - the number of channels of the synchronous generator.
- Frequency (1) - the frequency of the periodic signal of the synchronous generator, Hz.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Frequency - set the frequency value of the periodic signal, Hz (number).

➢**Custom common properties:**

- [Are available by the link below.](154)

➢**Custom private properties (the default value is shown in parentheses):**

- LONG Quantity - Set the number of synchronous generator channels (from 1 to 10, or from 1 to (number of generators - 1)).
- BSTR GenList - Set parameters of the synchronous generator (string).
- FLOAT Frequency - Set the frequency of the periodic signal of the synchronous generator (from 0.01 to (DACsampling frequency / 2)), Hz.

➢**Methods:**

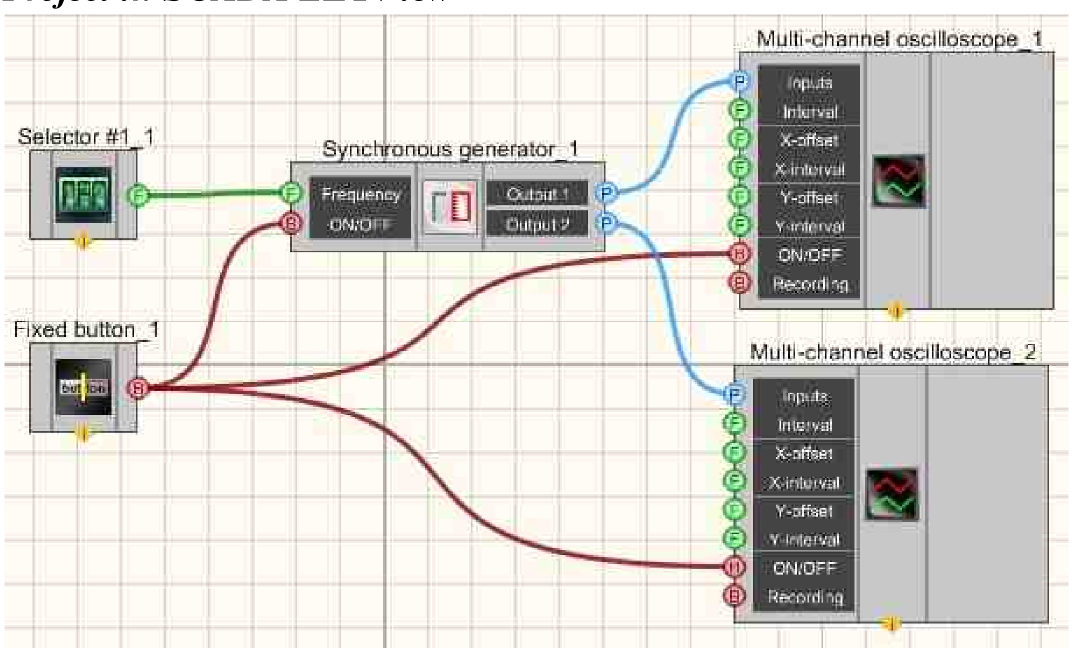
- LONG SignalType(LONG lNumber) - Returns the signal type of the current channel of the synchronous generator:
  0 – Sinel signal;
  1 – Impulse signal;
  2 – the signal from the file;
  3 – logarithmic frequency modulation;
  4 – RF impulse signal.
- void SignalType(LONG lNumber, LONG newVal) - Set the type and values of the signal of the current channel of the synchronous generator.
- void SetSinusParameters(LONG lNumber, FLOAT fLevel, FLOAT fShift, FLOAT fPhase) - Set the parameters of the Sine signal of the synchronous generator.
- void SetImpulseParameters(LONG lNumber, FLOAT fAmplitude, FLOAT fShift, FLOAT fPorosity, FLOAT fPhase) - Set the parameters of the Impulse signal.
- void SetFileParameters(LONG lNumber, BSTR sName, FLOAT fKoeff, FLOAT fDelay, FLOAT fCompression, VARIANT_BOOL bCycle, FLOAT fStartTime) - Set the signal parameters from the file.
- void SetRadioImpulseParameters(LONG lNumber, FLOAT fLevel, FLOAT fFillFrequency, FLOAT fFollowFrequency, FLOAT fShift, LONG lQuantity, FLOAT fAddLevel, FLOAT fAddPhase) - Set the parameters of the RF impulse signal.
- FLOAT SinusLevel(LONG lNumber) - Return the level of the Sine signal of the synchronous generator, (from 0 V to the maximum limit level of the DAC signal), V.
- void SinusLevel(LONG lNumber, FLOAT newVal) - Set the level of the Sine signal of the synchronous generator, V.
- FLOAT SinusShift(LONG lNumber) - Return the offset of the constant component of the Sine signal of the synchronous generator (from 0 V to the maximum limit level of the DAC signal), V.
- void SinusShift(LONG lNumber, FLOAT newVal) - Set the offset of the constant component of the Sine signal of the synchronous generator, V.

- FLOAT SinusPhase(LONG lNumber) - Return the phase offset of the Sine signal of the synchronous generator, ° (from 0° to 360°).
- void SinusPhase(LONG lNumber, FLOAT newVal) - Set the phase offset of the Sine signal of the synchronous generator, °.
- FLOAT ImpulseAmplitude(LONG lNumber) - Return the amplitude of the Impulse signal of the synchronous generator (from 0 V to the maximum limit level of the DAC signal), V.
- void ImpulseAmplitude(LONG lNumber, FLOAT newVal) - Sets the amplitude of the Impulse signal of the synchronous generator, V.
- FLOAT ImpulseShift(LONG lNumber) - Return the offset of the constant component of the Impulse signal of the synchronous generator (from 0 to the maximum limit level of the DAC signal), V.
- void ImpulseShift(LONG lNumber, FLOAT newVal) - Set the offset of the constant component of the Impulse signal of the synchronous generator, V.
- FLOAT ImpulsePorosity(LONG lNumber) - Return the duty cycle of the synchronous generator Impulse signal (from 0 to 1).
- void ImpulsePorosity(LONG lNumber, FLOAT newVal) - Set the duty cycle of the synchronous generator Impulse signal.
- FLOAT ImpulsePhase(LONG lNumber) - Return the phase offset of the synchronous generator Impulse signal, ° (from 0° to 360°).
- void ImpulsePhase(LONG lNumber, FLOAT newVal) - Set the phase offset of the synchronous generator RF impulse signal, °.
- FLOAT RadioImpulseLevel(LONG lNumber) - Returns the level of the RF impulse signal (from 0 V to the maximum limit DAC signal level), V.
- void RadioImpulseLevel(LONG lNumber, FLOAT newVal) - Set the level of the RF impulse signal, V.
- FLOAT RadioImpulseFillFrequensy(LONG lNumber) - Returns the frequency of filling of the RF impulse signal (from the repetition frequency of the radio Impulse signal to (DACsampling frequency / 2)), Hz.
- void RadioImpulseFillFrequensy(LONG lNumber, FLOAT newVal) - Sets the frequency of filling of the RF impulse signal, Hz.
- FLOAT RadioImpulseFollowFrequency(LONG lNumber) - Return the frequency of the RF impulse signal (from 0.01 Hz to the frequency of filling the RF impulse signal), Hz.
- void RadioImpulseFollowFrequency(LONG lNumber, FLOAT newVal) - Set the frequency of the RF impulse signal, Hz.
- FLOAT RadioImpulseShift(LONG lNumber) - Return the offset of the constant component of the RF impulse signal (from 0 V to the maximum limit level of the DAC signal), V.
- void RadioImpulseShift(LONG lNumber, FLOAT newVal) - Set the offset of the constant component of the RF impulse signal, V.
- LONG RadioImpulseQuantity(LONG lNumber) - Return the number of impulses of the RF impulse signal (from 1 to 10, or from 1 to (the number of generators is 1)).

- void RadioImpulseQuantity(LONG lNumber, LONG newVal) - Set the number of impulses of the RF impulse signal.
- FLOAT RadioImpulseAddLevel(LONG lNumber) - Return the level of the additional impulse of the RF impulse signal (from 0 V to the maximum limit level of the DAC signal), V.
- void RadioImpulseAddLevel(LONG lNumber, FLOAT newVal) - Set the level of the additional impulse of the RF impulse signal, V.
- FLOAT RadioImpulseAddPhase(LONG lNumber) - Returns the phase offset of the additional impulse of the RF impulse signal, ° (from 0° to 360°).
- void RadioImpulseAddPhase(LONG lNumber, FLOAT newVal) - Set the phase offset of the additional impulse of the RF impulse signal, °.

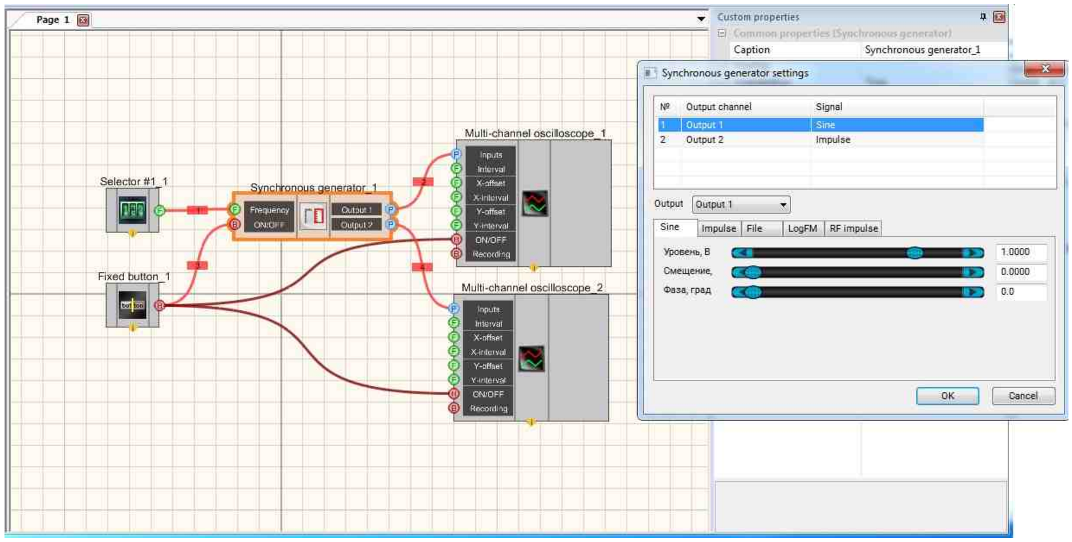
Example

Project in SCADA ZETView

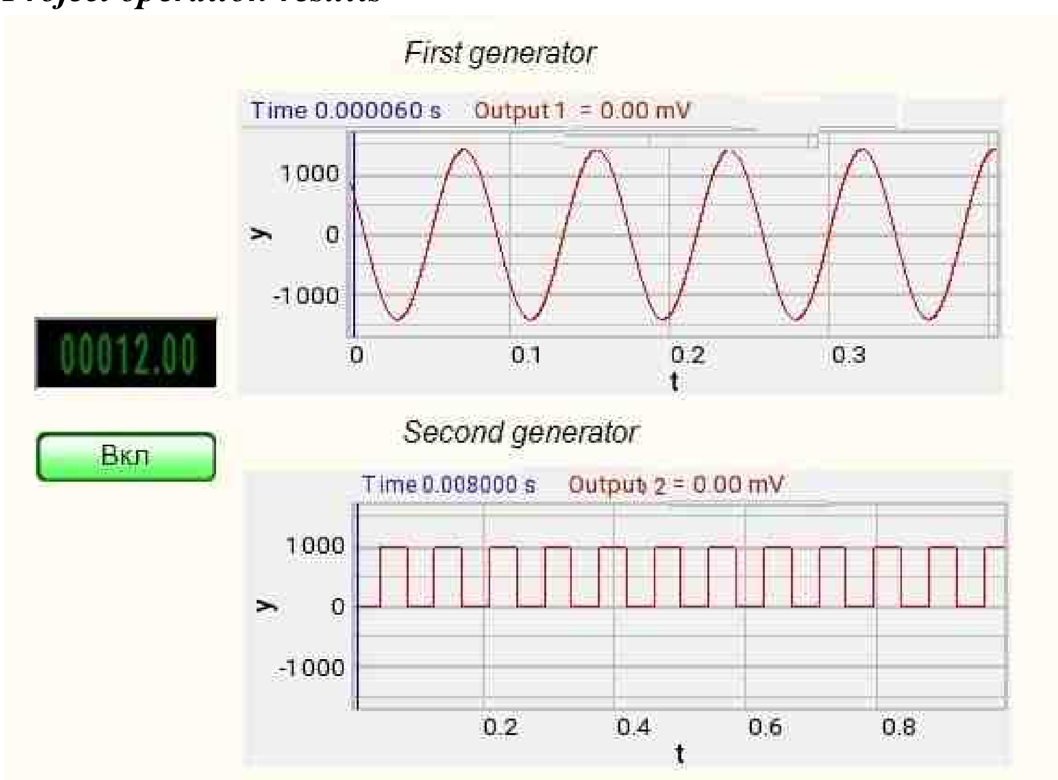


This project shows the operation of the Synchronous generator 537 component creates Sine, Impulse, file, LogFM, and RF impulses signals with parameters that are set using a Selector 428. The Fixed button 691 is used to turn on and off the Multi-channel oscilloscope 928 and Synchronous generator 537. The Multi-channel oscilloscope 928 component is used to view the generated signal. The result of the compiled program can be seen in the operator interface.

You can configure the Synchronous generator 537 by standing on the component and clicking Edit.



Project operation results



View the example in ZetView

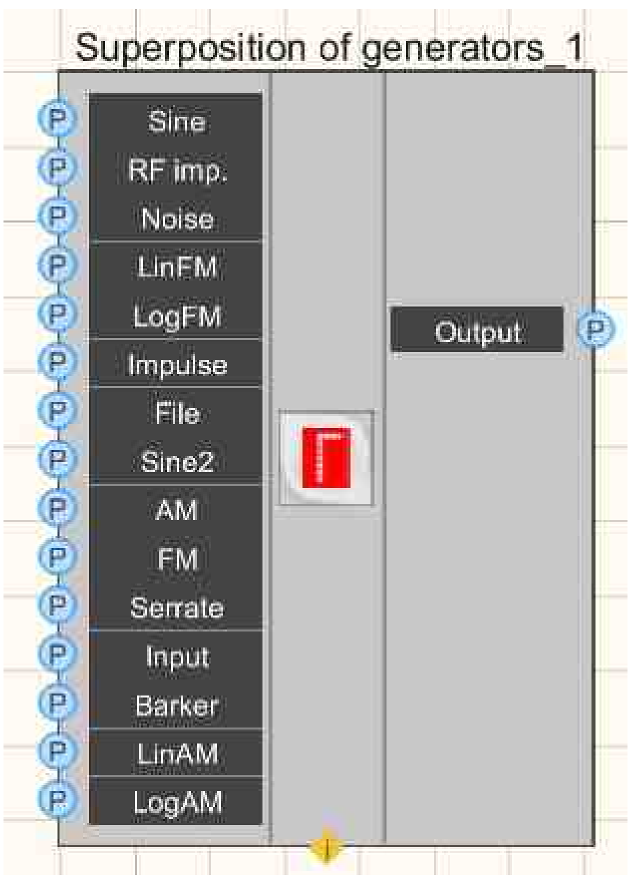
9.14.Superposition of generators

The component "**Superposition of generators**" is user to create a resulting signal from a set of input (Signal serrated, Impulse, and others). It is used to create new, desired waveforms.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

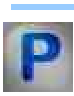
➢Input:

- Sine - is used to connect a Sine signal.
- RF Imp. - is used to connect a RF impulse signal.
- Noise - is used to connect a noise signal.
- LinFM - is used to connect a linear frequency modulation signal.
- LogFM - is used to connect a logarithmically frequency modulation signal.
- Impulse - is used to connect a signal consisting of rectangular impulses.
- File - is used to connect the signal Play recorded from file.
- Sine2 - is used to connect an additional Sine signal.
- AM - is used to connect an amplitude modulation signal.
- FM - is used to connect a frequency modulation signal.
- Serrate - is used to connect a Signal serrated.
- Input - is used to connect a Virtual channel or a measuring channel.
- Barker - is used to connect a signal generator according to Barker codes.
- LinAM - is used to connect a signal generator by linear amplitude modulation.
- LogAM - is used to connect a signal generator by logarithmic amplitude modulation.

➢Output
- Output - the output of the generator generated by the specified parameters of the signal.

Custom property:
➢Custom common properties:
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.



Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Sine - is used to connect a Sine signal.
- RF Imp. - is used to connect a RF impulse signal.
- Noise - is used to connect a noise signal.
- LinFM - is used to connect a linear frequency modulation signal.
- LogFM - is used to connect a logarithmically frequency modulation signal.
- Impulse - is used to connect a signal consisting of rectangular impulses.
- File - is used to connect the signal Play recorded from file.
- Sine2 - is used to connect an additional Sine signal.
- AM - is used to connect an amplitude modulation signal.
- FM - is used to connect a frequency modulation signal.
- Serrate - is used to connect a Signal serrated.
- Input - is used to connect a Virtual channel or a measuring channel.
- Barker - is used to connect a signal generator according to Barker codes.
- LinAM - is used to connect a signal generator by linear amplitude modulation.
- LogAM - is used to connect a signal generator by logarithmic amplitude modulation.

➢**Custom common properties:**

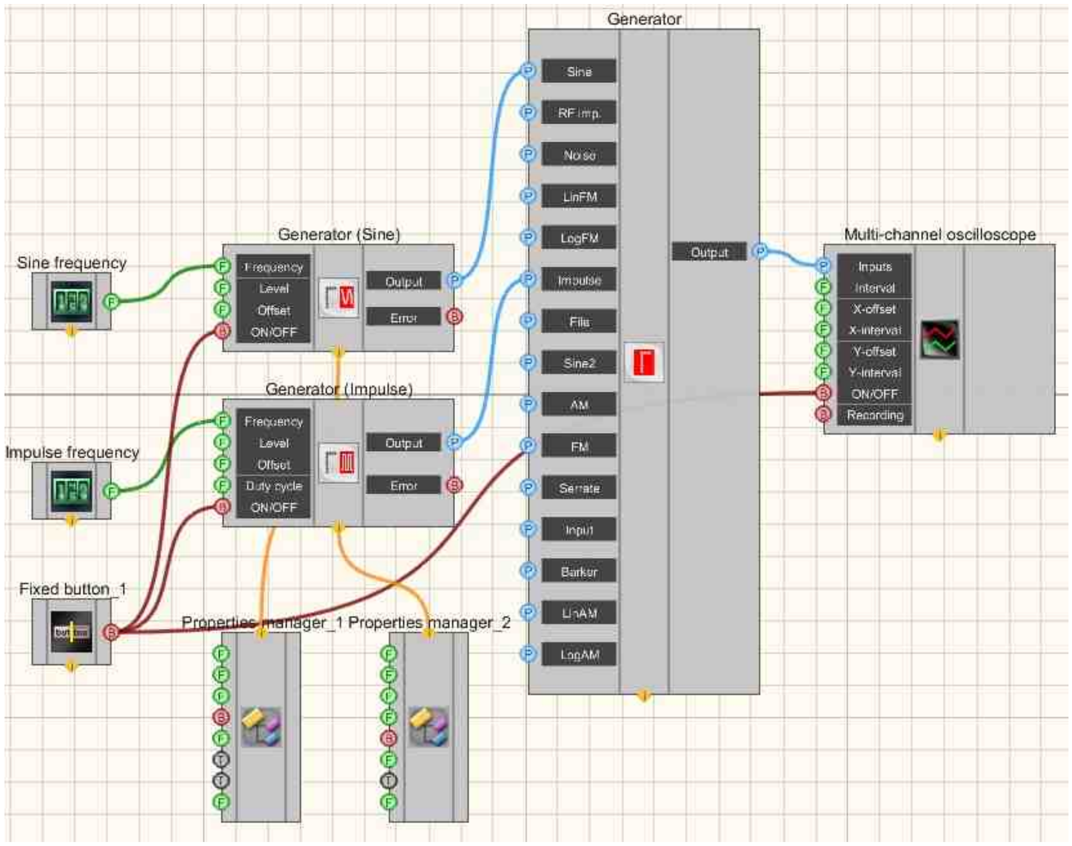
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

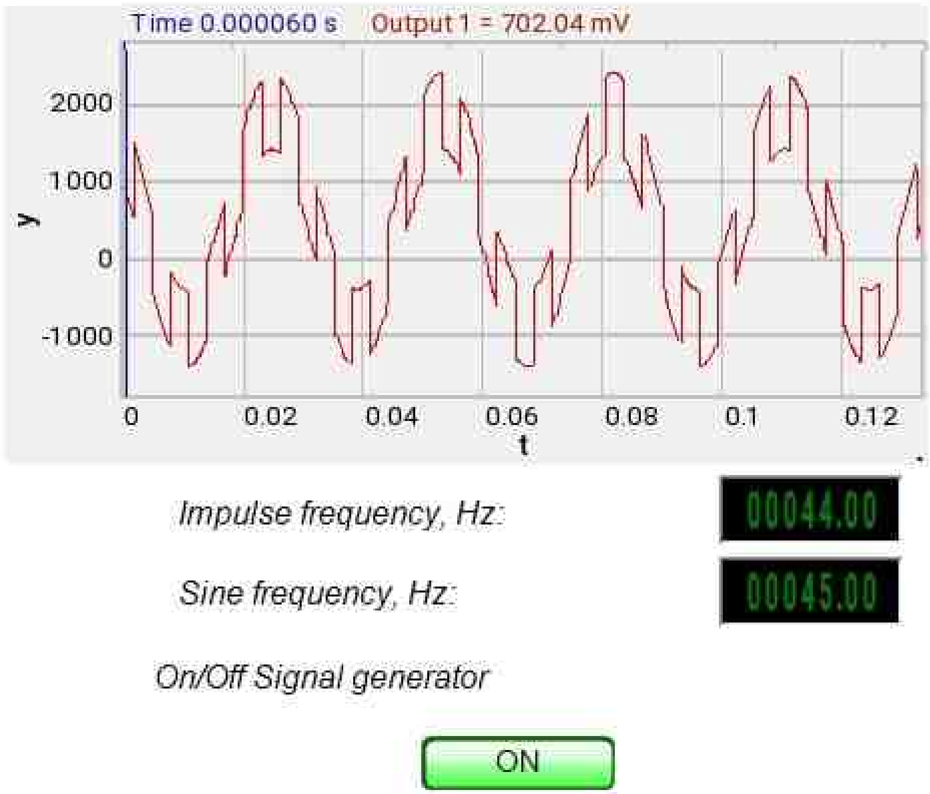
Example

Project in SCADA ZETView



This project shows the operation of the Superposition of generators 543 component creates a resulting signal consisting of a sum of Sine and Impulse signals. To create these signals, the components Sine signal 530 and Impulse signal 523 are used. Selectors 428 are needed to set frequency values. A Multi-channel oscilloscope 928 is used to graphically display the received signal.the operation of the signal generator by Barker codes.

Project operation results



View the example in ZetView

Mathematical description

In the 1950s and 1960s, whole classes of discrete signals with perfect correlation properties were developed.

The autocorrelation function of phase-modulated signals has the form typical for all types of a broadband signal. The normalized autocorrelation function consists of a central (basic) type with an amplitude of 1, placed on the interval (-τ,τ) and side (background) maxima distributed over the interval (-T,τ) And (τ,T).

The amplitudes of the side types take on different values, but for signals with a "good" correlation they are small, i.e. much less than the amplitude of the central peak.

The ratio of the amplitude of the central peak (in this case 1) to the maximum amplitude of the side maxima is called the rejection factor K. For arbitrary broadband signals with base B

$$K \approx \frac{1}{\sqrt{B}}$$

For phase modulated wideband signals $K \approx \frac{1}{\sqrt{N}}$. An example of the autocorrelation function of a broadband signal is given in Figure 2. The value of K significantly depends on the type of code sequence A. With the right choice of the formation law A, it is possible to achieve maximum suppression, and in some cases, equality of the amplitudes of all side maxima.

Barker signals have power spectra that deviate least (in a quadratic sense) from the spectrum of a single sample. Their autocorrelation functions correspondingly approach the auto-correlation function of the sample. Below is a table of Barker codes.

N	*k*												The level of the side lobes
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	
2	+1	-1											-
3	+1	+1	-1										-1/3
4	+1	+1	-1										1/4
5	+1	+1	+1	+1									1/5
7	+1	+1	+1	-1	+1	-1							-1/7
11	+1	+1	+1	-1	-1	+1	-1	-1	+1	-1			-1/11
13	+1	+1	+1	+1	-1	-1	+1	+1	-1	+1	-1	+1	1/13

Table 1

The code sequence of the Barker signal consists of the symbols±1 and is characterized by a normalized autocorrelation function of the form:

$$B(\tau) = \begin{cases} 1 \text{ для } \tau = 0, \\ 0 \text{ для } \tau = 2l + 1, \\ \pm\frac{1}{N} \text{ для } \tau = 2l. \end{cases}$$ - (1)

Where $l = 0, 1, \ldots \frac{N-1}{2}$

The sign in the last line depends on the value of N. Figures 1-2 show the phase-modulated signal, its complex envelope, and the autocorrelation function of the seven-digit Barker code.

From (1) it follows that one of the features of the Barker signal is the equality of the amplitudes of all (N-1) side maxima of the ACF, and all of them have the lowest possible level, not exceeding 1/N. Table 1 lists known Barker code sequences and their levels of side ACF types. Code sequences with B(τ) properties for N>13 not found.

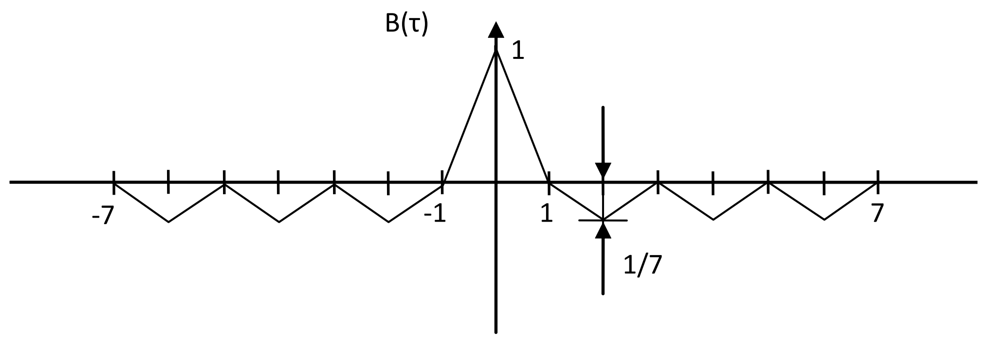


Fig.1 - Autocorrelation function of the seven-digit Barker code.

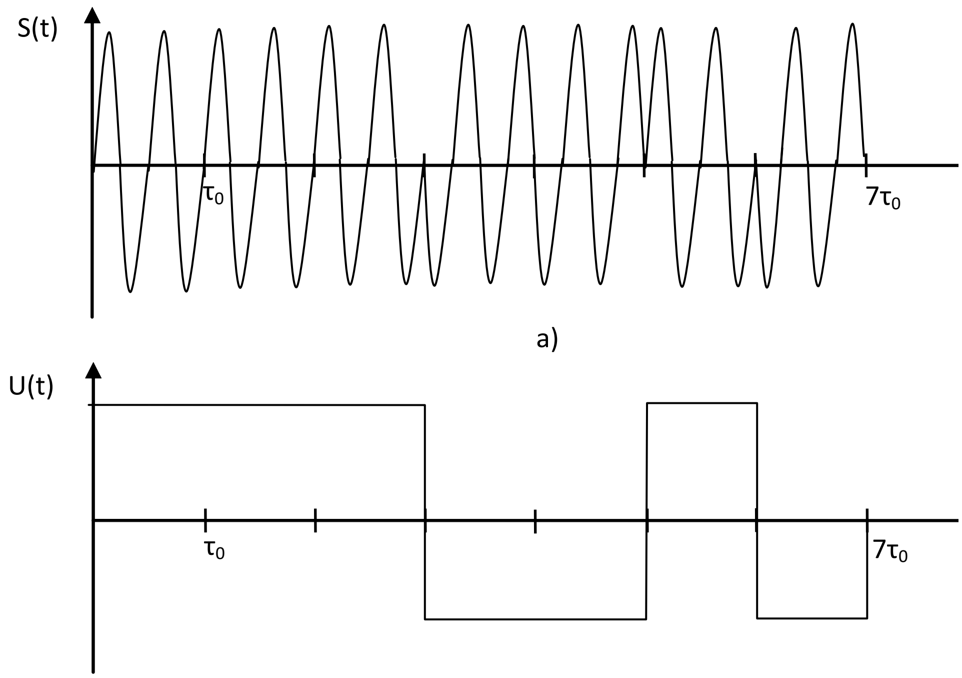


Fig. 2-a) Phase-modulated signal, b) its complex envelope.

Barker codes have the best noise-like properties among known pseudo-random sequences, which led to their widespread use: they are used in the IEEE 802.11 wireless network protocol. The 802.11 family protocols use an 11-chip Barker code (11100010010). In order to transmit a signal, a logical one is transmitted by the forward Barker sequence, and a logical zero by the inverse sequence.

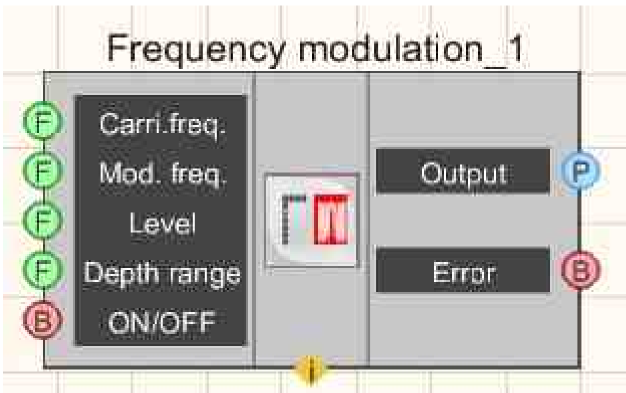
9.15.Frequency modulation

The component "**Frequency modulation (FM)**" is user to generate a signal generated by a special frequency modulation law, in which the information signal controls the frequency of the carrier oscillation. In comparison with amplitude modulation, here the amplitude remains constant.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Carri. freq. - carrier signal frequency, Hz.
- Mod. freq. - frequency modulation frequency, Hz.
- Level - frequency modulation level, V.
- Depth range - the frequency modulation factor.
- ON\OFF - turn on and off the generator.

➢Output
- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- CentralFrequency (1000) - frequency of the carrier signal, Hz.
- ModFrequency (1) - frequency modulation frequency, Hz.
- Level (1) - the amplitude of the output signal, V.
- Depth (0) - depth of frequency modulation.
- Activate (false) - the status of the signal (enabled or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of

the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.

- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Carrier frequency - The value of the output value of the frequency of the carrier signal, Hz (number).
- Modulation frequency - The value of the output frequency of frequency modulation, Hz (number).
- Level - The value of the amplitude of the output signal, V (number).
- Depth range - The value of the frequency modulation depth (number).

➢**Custom common properties:**

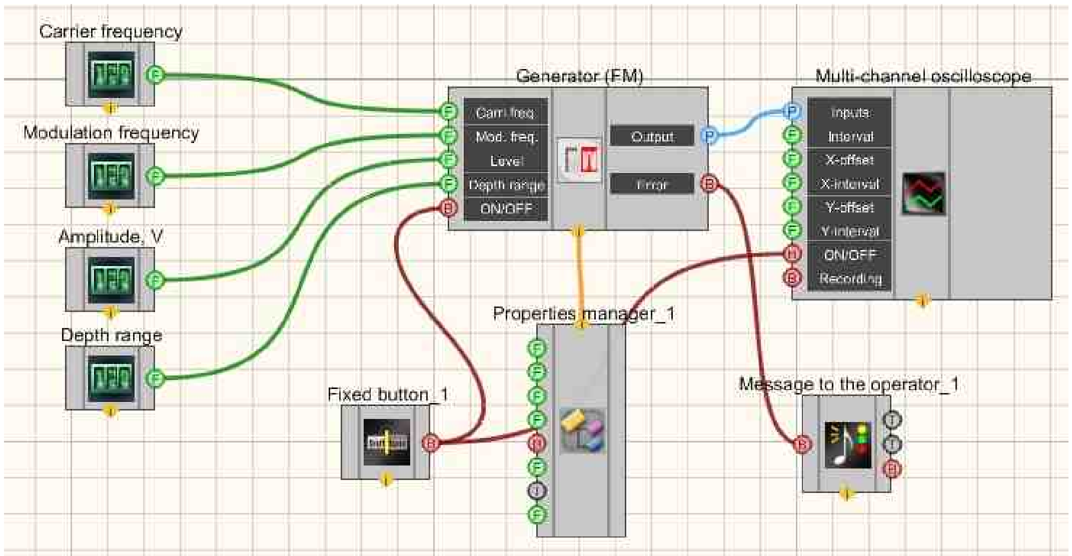
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

- FLOAT CentralFrequency - Set the frequency of the carrier signal (from 0.01 to the maximum limit level of the DAC signal), Hz.
- FLOAT ModFrequency - Set the frequency modulation frequency (from 0.01 to the maximum limit DAC signal level), Hz.
- FLOAT Level - Set the signal level (from 0 V to the maximum limit DAC signal level), V.
- FLOAT Depth - Set the depth of the frequency modulation (from 0 to 1).
- VARIANT_BOOL Activate - Set the signal presence status:
  - true - the signal presence status is turned ON;
  - false - the signal presence status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

Example

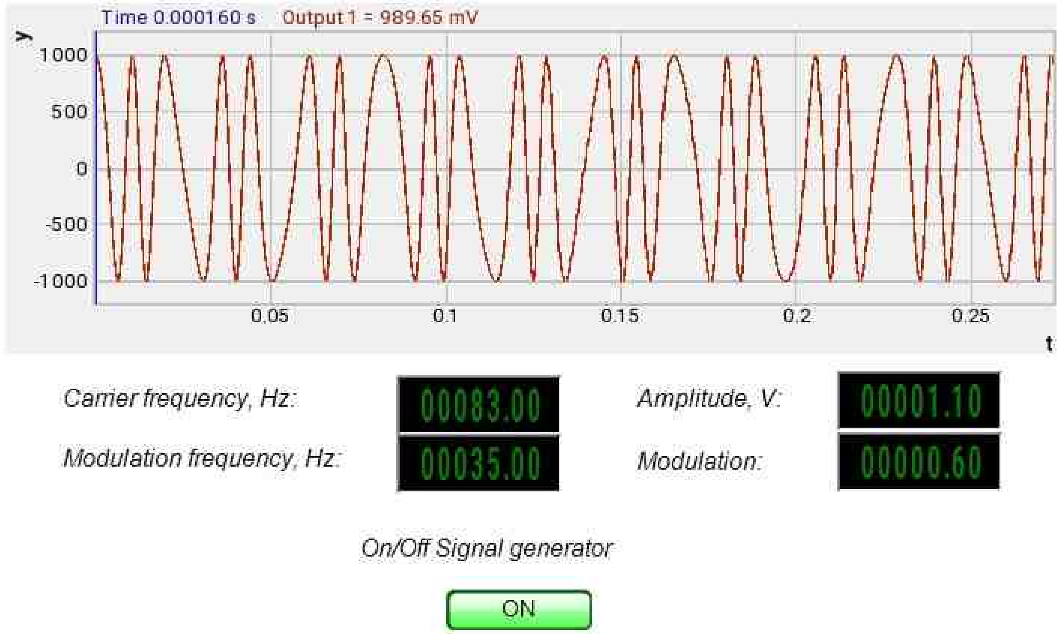
Project in SCADA ZETView



In this scheme, Frequency modulation 550 component performs frequency modulation of the signal. The Selectors 428 are used to set the carrier and modulation frequencies, level, depth range. A Multi-channel oscilloscope 928 is needed for visual presentation modulated signal.

The Property manager 999 allows you to change indicators in real time and transfer data to the signal generator to display them on a Multi-channel oscilloscope.

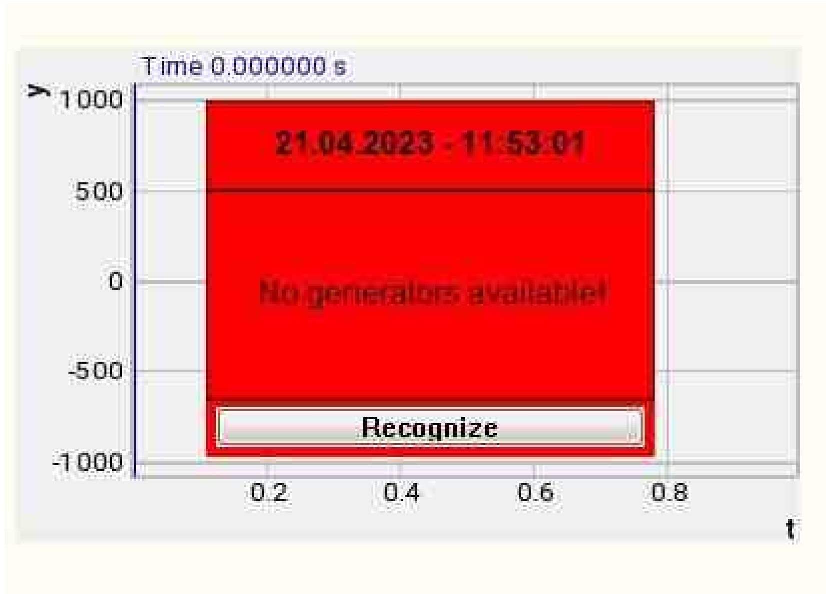
Project operation results



View the example in ZetView

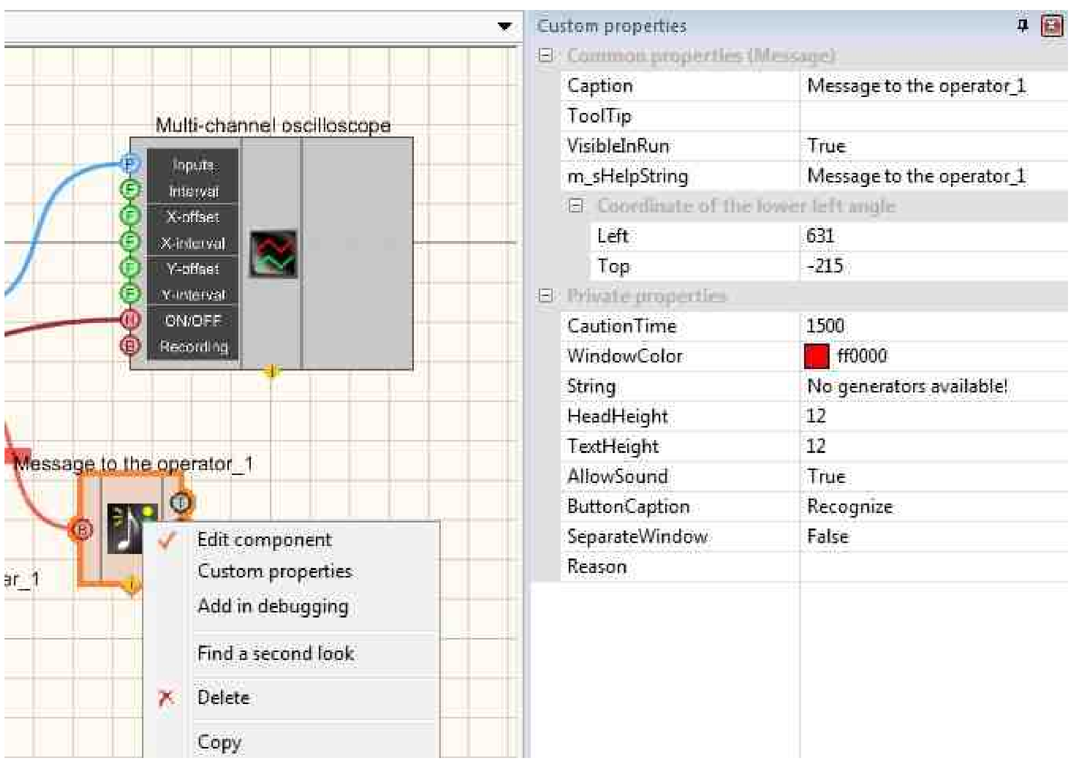
Output about an error in the project

The component Message to the operator [1052] is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal [530]. In addition, the message Sine signal # 1 [530] connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

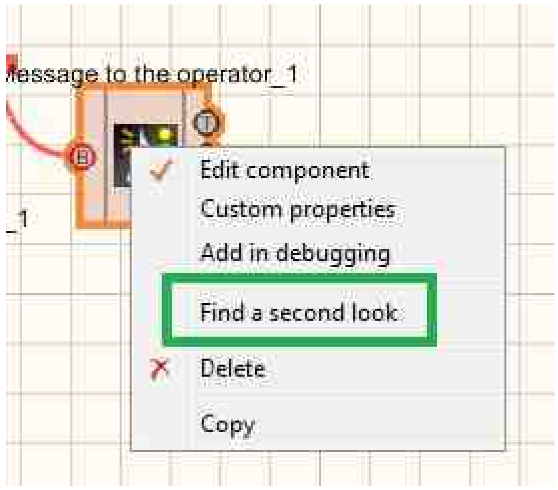


Message about the error to the operator.

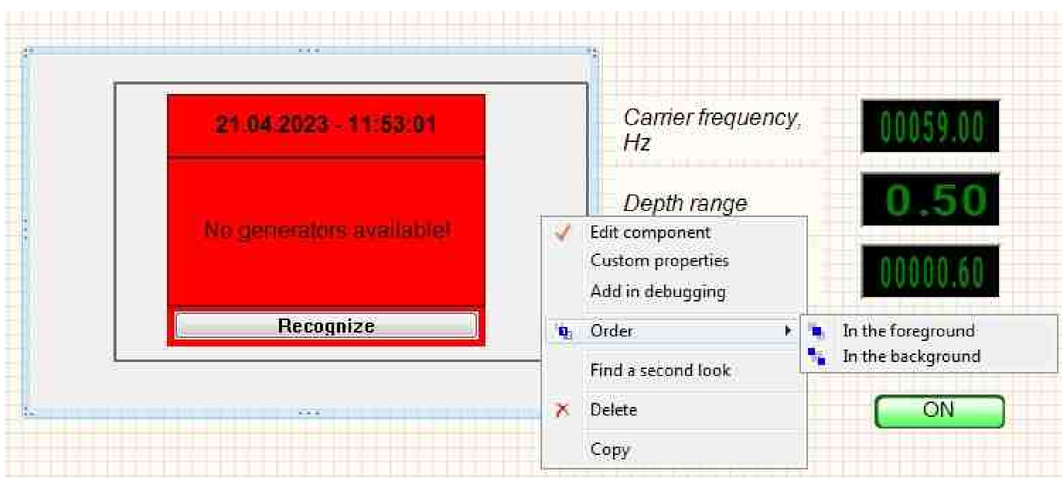
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



Mathematical description

Phase and frequency modulation of the signal consist in changing the frequency or phase of the RF oscillation according to the law of the control signal. FM was known in the 1920s, but it has been put into practice since the mid 1930s, with the development of

VHF communication technology. At present, FM is widely used in commercial radio communications, VHF broadcasting and television for sound transmission, etc. As you know, periodic oscillations can be expressed by the equation:

$$i = J_m \cos(\omega_0 t + \varphi),$$

Where φ is the oscillation phase characterizing the status of the oscillatory process at a given time, ω_0- frequency of high-frequency oscillations. Phase is a function of time and is related to frequency as follows:

$$\varphi = \int \omega dt.$$

In its turn:

$$\omega = \frac{d\varphi}{dt}.$$

The mutual dependence of the phase and the angular frequency is such that any deviation of the phase from the linear law leads to a deviation of the frequency from the initial value and vice versa. For these reasons, Phase modulation (PM) always changes the frequency, while Frequency modulation (FM) always changes the phase.

Despite their close relationship, frequency and phase modulation can be distinguished by which of the parameters (frequency or phase) is influenced by the modulating factor. Frequency modulation is much better than amplitude modulation.

If the modulating factor changes phase, then the modulation should be considered phase (although the frequency also changes) and, conversely, if the modulating factor changes frequency (although the phase changes), then the modulation should be considered frequency.

Consider their mutual dependence. With Phase modulation (PM), the phase changes according to the law of the modulating voltage UΩ(t), i.e. $\varphi = \omega_0 e + \varphi_0$,

Where $\omega_0 t$ is a phase component that varies linearly;

$\varphi_0 = kU_{m\Omega} \cos \Omega t$ - phase component that changes according to the voltage law UΩ(t), where k is the proportionality factor.

As a result, with phase modulation:

$$i = J_m \cos(\omega_0 t + kU_{m\Omega} \cos A\Omega t) \text{ или } i = J_m \cos(\omega_0 t + m_\varphi \cos \Omega t),$$

where $m\varphi = kUm\Omega = \Delta\varphi$ is the maximum phase deviation from the linear law or the phase modulation index. The Phase modulation (PM) index, as well as the modulation coefficient (with amplitude modulation) depends on the amplitude of the low frequency $Um\Omega$. A change in phase leads to a change in frequency: when the phase is advanced, the frequency ω increases, and when the phase is lagged, it decreases.

The law of frequency change during Phase modulation (PM) can be determined from the equation

$$\omega = \frac{d\varphi}{dt} = \frac{d(\omega_0 t + m_\varphi \cos \Omega t)}{dt} = \omega_0 - m_\varphi \Omega \sin \Omega t = \omega_0 - \Delta\omega \sin \Omega t,$$

Where $\Delta\omega = m_\varphi \Omega = kU_{m\Omega}\Omega$ - maximum frequency deviation from the initial value or frequency deviation. It follows from this expression that in the case of Phase modulation (PM), the direct proportionality between the frequency deviation Δω and the amplitude of the modulating voltage $Um\Omega$ is not observed. The frequency deviation Δφ depends on the modulation frequency Ω and is different at different modulating frequencies. This dependence is explained as follows: the greater the modulation frequency Ω, the more often (faster) the phase φ changes and, consequently, the greater the rate of phase change in time (i.e., the higher the angular frequency). With FM, the oscillation frequency changes according to the law of the modulating voltage: $\omega = \omega_0 + \omega_1$, Where $\omega_1 = kU_{m\Omega} \cos \Omega t$ – characterizes the frequency deviation from the initial value. Value $\Delta\omega = kU_{m\Omega}$ - there is a maximum frequency deviation from the average value or frequency deviation. At the moments of maximum, i.e. + $Um\Omega$, the frequency increases, and at the moments of minimum - $Um\Omega$, the frequency decreases. In this case, the phase of oscillations changes. Let's define the oscillation phase for frequency modulation:

$$\varphi = \int \omega dt = \int (\omega_0 + \Delta\omega \cos \Omega t) dt = \omega_0 t + \frac{\Delta\omega}{\Omega} \sin \Omega t + \varphi_0.$$

Assuming for simplification φ0 = 0, we get:

$$\varphi = \omega_0 t + \frac{\Delta\omega}{\Omega} \sin \Omega t = \omega_0 t + m_f \sin \Omega t,$$

Where $m_f = \frac{\frac{\Delta\omega}{\Omega}}{\Omega} = \Delta\varphi$ - the maximum phase deviation from the linear law or the frequency modulation index. Therefore, as we see, from the above formula for φ - the phase of oscillations during frequency modulation changes according to a different law than frequency and is not proportional to the amplitude of the modulating voltage $Um\Omega$. Based on the FM equation, oscillations can be represented as follows:

$$i = J_m \cos(\omega_0 t + m_f \sin \Omega t).$$

The form of the FM waveform is similar in appearance to the Phase modulation (PM) waveform. However, the laws of frequency change in them are different. So, if the phase changes according to the sin Ωt law, then the frequency changes according to the cos Ωt law and vice versa.

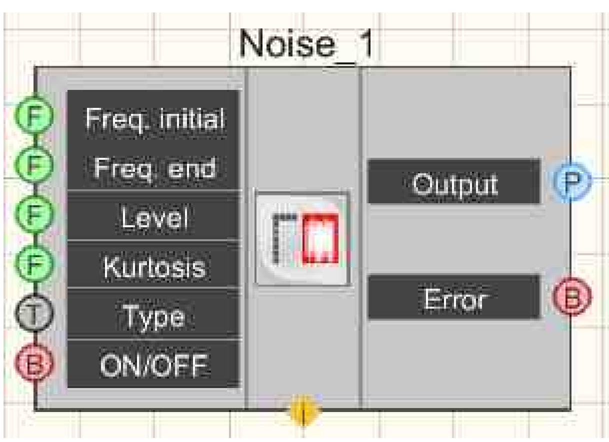
9.16.Noise

The component "**Noise**" to generate stationary noise, the spectral components of which are evenly distributed over the entire range of frequencies involved.

For the component to work, it is necessary to have a physical ZET device containing a DAC connected to a PC.

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Freq. initial - the initial signal frequency of the noise in which the noise will be generated, Hz.
- Freq. end signal - the end signal frequency of the noise in which the noise will be generated, Hz.
- Level - noise signal level, V.
- Kurtosis - sets the value of kurtosis from 2 to 10.
- Signal type - 6 types of noise are implemented:
  1) White;
  2) Bandpass;
  3) Pink;
  4) Red;
  5) Determinate;
  6) Semi-white.
  6) Colored.
- ON\OFF - turn on and off the generator.

➢Output
- Output - the output of the generator generated by the specified parameters of the signal.
- Error - an error message is displayed.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- StartFrequency (0) - the initial signal frequency of the noise in which the noise will be generated, Hz.
- EndFrequency (1000) - the end signal frequency of the output signal, Hz.
- Level (1) - the amplitude of the output signal, V.
- type (White) - a parameter that Set the type of noise:
  1) White;
  2) Bandpass;
  3) Pink;
  4) Red;
  5) Determinate;
  6) Semi-white.
  6) Colored.
- Activate (false) - the status of the signal (enabled or disabled).
- Number (0) - the number of the generator channel to be activated. If the value of DeviceType and Serial Number are not equal to 0, but the selection goes among the channels of the specified device. If there is no such device in the system or the value of DeviceType or SerialNumber is 0, then the selection goes among all available channels of all devices.
- DeviceType () - the type of device whose generator needs to be activated.
- SerialNumber (0) - the serial number of the device whose generator needs to be activated.

- Kurtosis(3) - value of kurtosis.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Initial frequency - The value of the output value of the initial signal frequency of the frequency range in which white noise will be generated, Hz (number).
- End frequency - The value of the output value of the end signal frequency of the frequency range in which white noise will be generated, Hz (number).
- Level - Noise level value, V (number).
- Kurtosis - Kurtosis value from 2 to 10 (number).
- Signal type - 6 types of noise are implemented (list):
  1) White;
  2) Bandpass;
  3) Pink;
  4) Red;
  5) Determinate;
  6) Semi-white.
  6) Colored.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOATStartFrequency - Set the initial frequency of the signal (from 0.01 Hz to the final frequency of the noise), Hz.
- FLOATEndFrequency - Set the end frequency of the signal (from the initial noise frequency to (DAC sampling frequency / 2)), Hz.
- FLOATLevel - Set the signal level (from 0 V to (maximum allowable DAC signal level / 5)), V.
- BSTRType - Set signal type: White; Bandpass; Pink; Red; Determinate; Semi-white; Colored (string).
- VARIANT_BOOL Activate - Set the signal presence status:

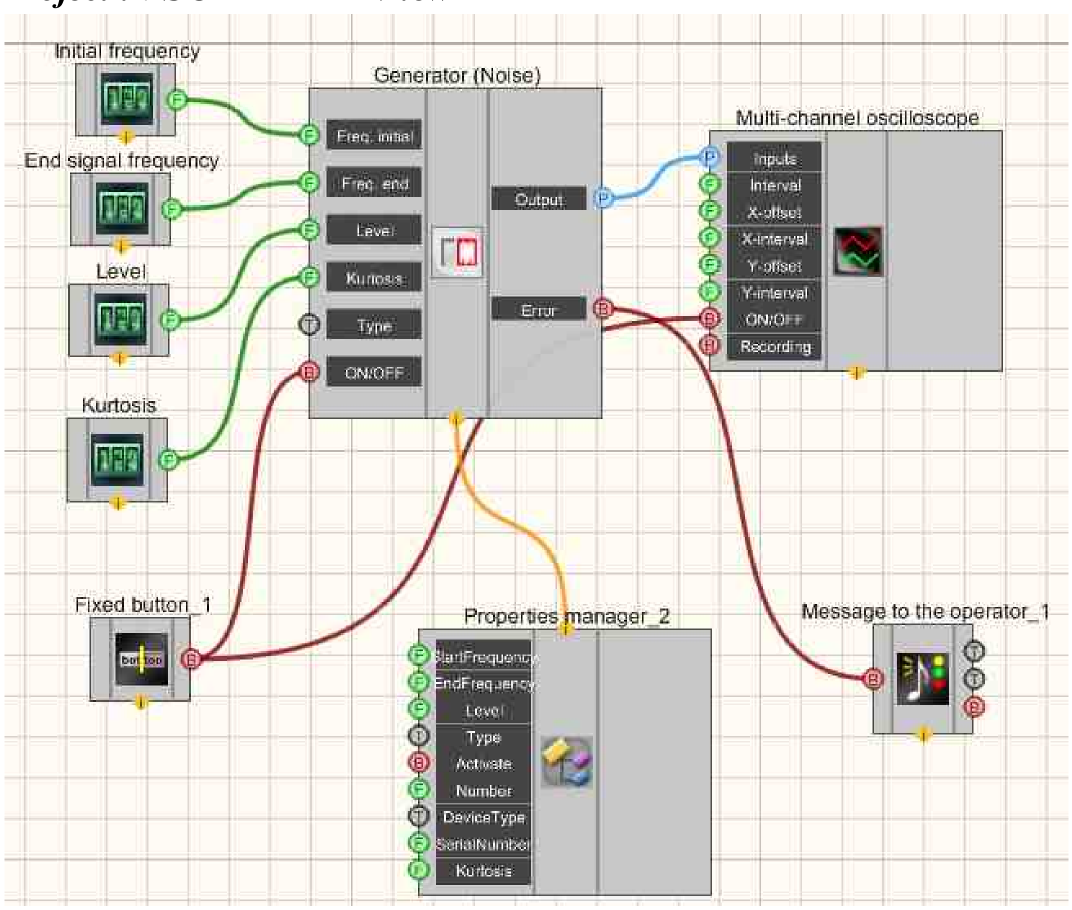
- true - the signal presence status is turned ON;
- false - the signal presence status is turned OFF.
- LONG Number - Set the generator number in the system (from 0 to (number of generators - 1)).
- BSTR DeviceType - Set the type of device whose generator is to be enabled (string).
- LONG SerialNumber - Set the serial number of the device (number).

➤**Methods:**

- void SetSignal(FLOAT StartFrequency, FLOAT EndFrequency, FLOAT Level, FLOAT Kurtosis, BSTR Type) - Set initial signal frequency frequency, end signal frequency, level, kurtosis, noise type.
- void DeleteSignal(void) - Stop the generator.

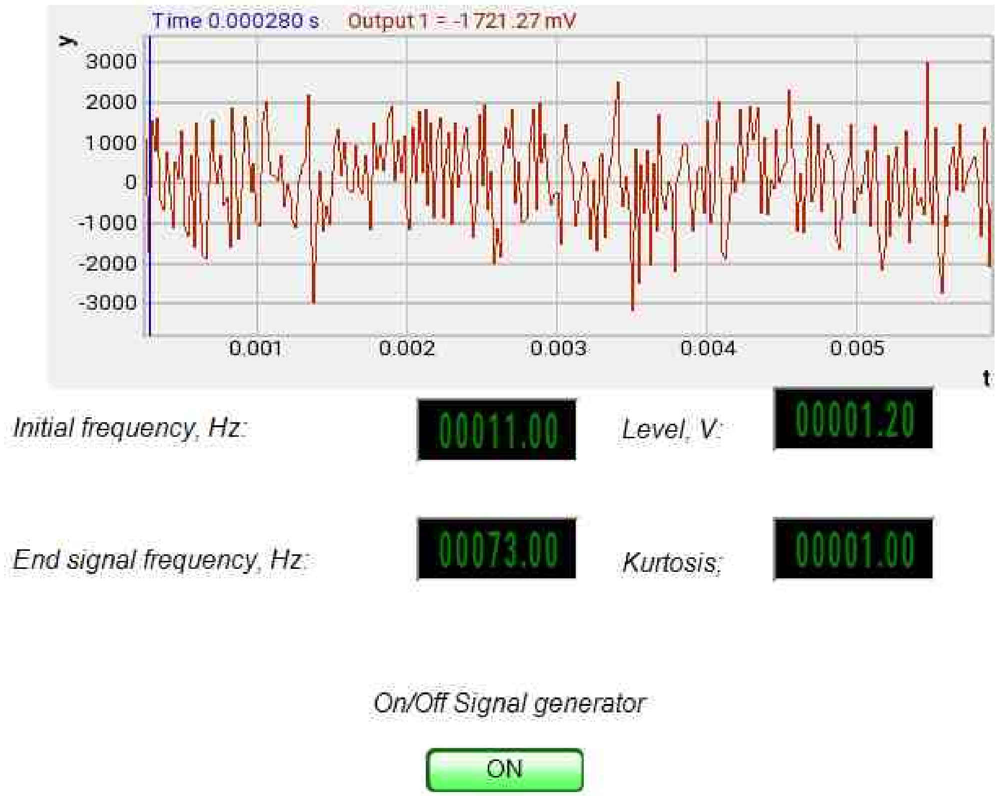
Example

Project in SCADA ZETView



This project shows the operation of the Noise 559 component generates white noise. Selectors 428 are used to set the initial, end signal frequency, noise level.The Fixed button 691 is used to turn on or off.
The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a Multi-channel oscilloscope.

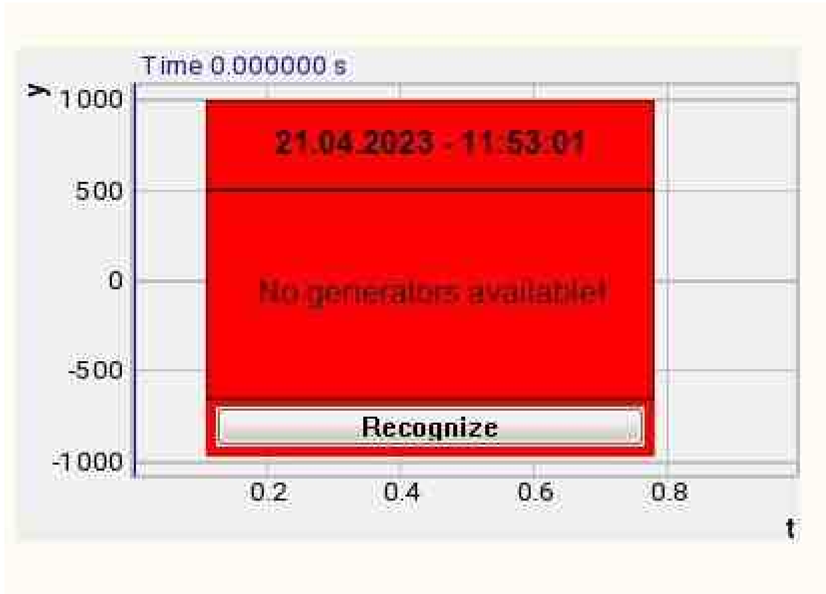
Project operation results



View the example in ZetView

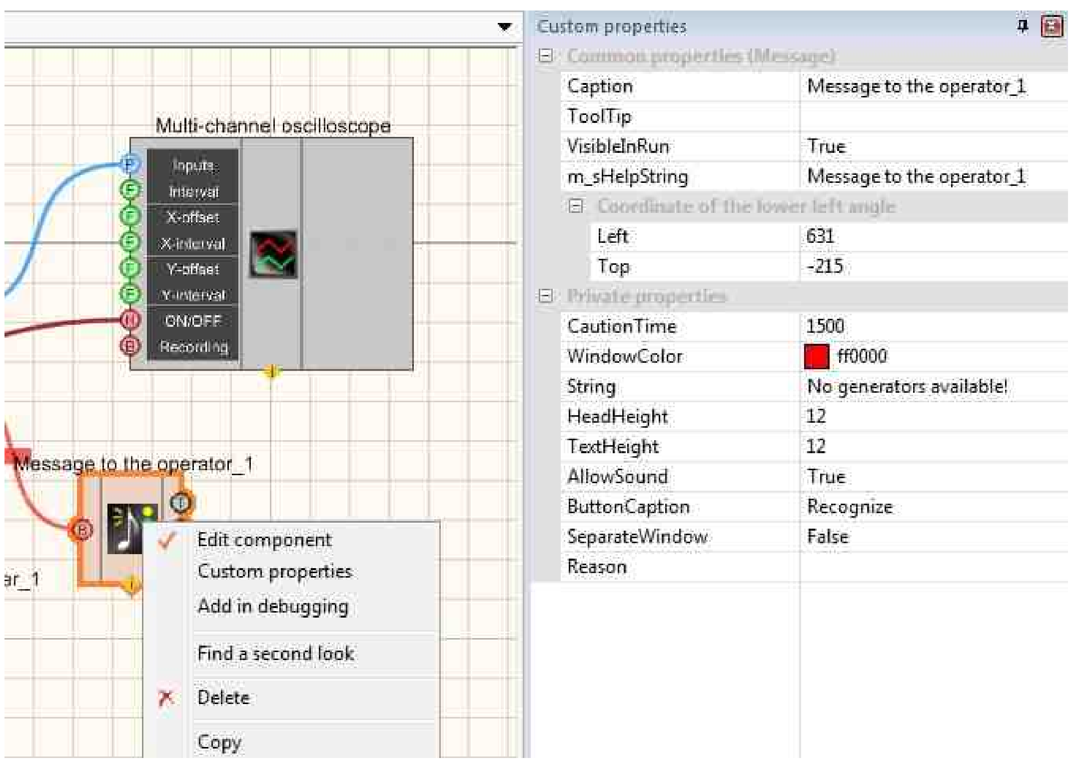
Output about an error in the project

The component Message to the operator 1052 is used to display a message about detected problems, in Example, the instrument is turned off and it is impossible to start the Sine signal 530. In addition, the message Sine signal # 1 530 connected devices" is displayed in the Zetlab Error journal and "No generators available!" in the modal window of the ZETView program.
Sine signal

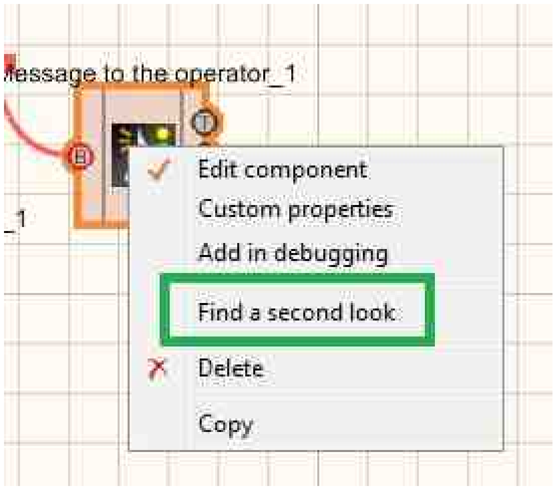


Message about the error to the operator.

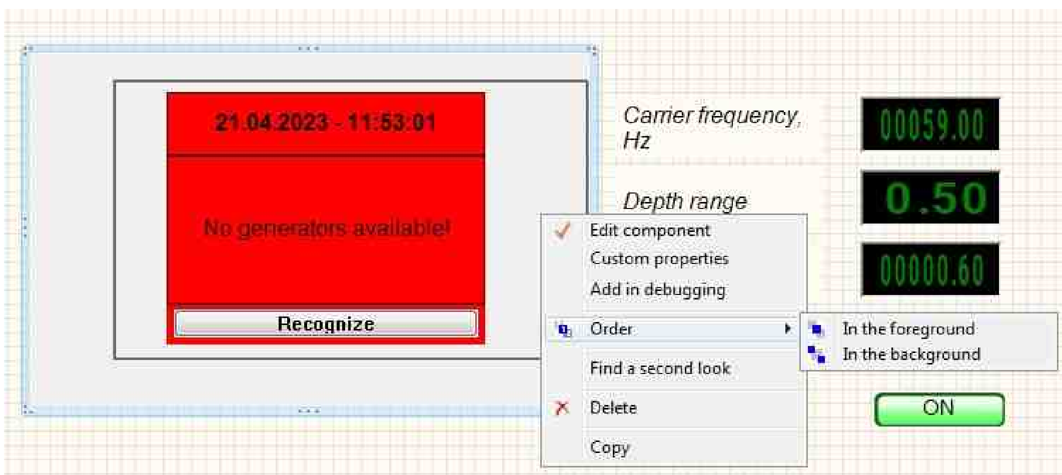
More detailed:
In the property of the component "Message to the operator", you can make changes to the labels



Select the "Message to the operator" component in developer mode and right-click and select the Find the second view line and click it to go to the Operator interface.



Right-click again and select the Order and determine whether the message will be displayed in the foreground or in the background.



Mathematical description

White noise is stationary noise, the spectral components of which are evenly distributed over the entire range of frequencies involved.

White noise is calculated using the formula:

$A = A_0(\sum_1^{12} \text{rand}() - 6)$,

Where $A_0 = \frac{2V_{rms}}{\sqrt{2}}$ is the initial amplitude of the signal.

Rand() is a function for calculating a random number, with 0<rand()<1.

The term "white noise" is usually applied to a signal that has an autocorrelation function*, mathematically described by the Dirac delta function** over all dimensions of the multidimensional space in which this signal is considered. Signals with this property can be considered as white noise. This statistical property is fundamental for signals of this type.

Bandpass noise is a noise signal with a limited frequency interval.

Pink noise is a noise signal whose spectral level decreases with increasing frequency with a roll-off of 3 dB per octave.

Deterministic noise is white noise limited to a given frequency range.

-- ----------------------------

* In signal processing, the autocorrelation function is determined by the integral

$\psi(\tau) = \int f(t)f(t-\tau)dt$

and shows the connection of the signal (function f(t)) with a copy of itself, shifted by τ.

** The Dirac δ-function with domain Rn for a point a is defined by the formal relation

$(\delta; f) = \int_{R^n} \delta(\overline{x} - \overline{a}) f(\overline{x})\, d^m x$.

***

In almost any area of measurement, the value of an extremely distinguishable weak signal is determined by noise - an interfering signal that clogs the useful signal. Some noise cannot be eliminated (fluctuations in the measured value). Some can be eliminated with filters. The term "noise" is applied to anything that masks a useful signal. "Noise" may be another electrical signal ("interference"), but more often it is a random noise of a physical nature.

Johnson noise. Any resistor on the board generates some noise voltage across its terminals, known as "Johnson noise" (Thermal noise). It has a horizontal frequency spectrum, i.e. the same noise power at all frequencies. Noise with a horizontal spectrum is called "white noise". The actual noise voltage in an open circuit, generated by a resistance R at a temperature T, is expressed by the formula:

$$U_{ш.эфф} = U_{ш\,R} = \sqrt{4kTRB},$$

Where k is the Boltzmann constant, T is the absolute temperature in Kelvin, B is the frequency band in Hz.

The voltage amplitude of the Johnson noise is, generally speaking, currently unpredictable, but it follows a Gaussian distribution. This type of noise sets the lower limit of the noise voltage of any detector, signal source containing resistors in its circuit.

Shot noise - noise that occurs in an electric current due to the movement of discrete charge carriers. The finiteness (quantization) of the charge leads to statistical current fluctuations. If the charges act independently of each other, then the fluctuating current is given by:

$$I_{ш.эфф} = I_{шR} = \sqrt{2qI_{=}B},$$

Where q is the electron charge, I= is the constant component ("steady value") of the current, B is the measurement bandwidth.

Noise 1/f (flicker noise). Shot noise and thermal noise are irreducible types of noise that occur in accordance with the laws of physics. Differently manufactured resistors have the same thermal noise figure. Real resistors are subject to resistance fluctuations, which generate additional noise voltages proportional to the DC current flowing through the resistor. This noise has a spectrum exemplified by 1/f (constant power per decade of frequency) and is sometimes referred to as "pink noise". Examples of such noise are base current noise in a transistor and cathode current noise in vacuum tubes. This type of noise is also found in the flow of sand in an hourglass, passenger flows on high-speed railroads, the speed of ocean currents.

Глава 10.Groups

The component "Combine a group" is used to unite components groups into logical blocks in operator mode (the frame type is selected by the user).

The component "Windows panel style" is used as a comment field. In allows to make text notes for the operator.

10.1.Combine a group

The component "**Combine a group"** is used to unite components groups into logical blocks in operator mode (the frame type is selected by the user).

Appearance of the component:

Developer interface	Operator interface
Doesn't have.	

Setting:

➢Input:

Doesn't have.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Text - text inscription.
- Textposition(Horizontal) - text position.
- TextSize (12) - text size.
- textstyle (FontStyleRegular) - text style
- TextColor (000000) - text color.
- FrameColor (808080) - frame color.
- BackColor (808080) - background color.
- Transparent (true) - enable/disable transparency. Transparent - transparency of the component. When enabled (true), the background color of the component matches the color of the page; when disabled (false), the background color is set by the BackColor property.
- alignment (By align the left edge) - type of text alignment:
  1. By align the left edge.
  2. In the middle.
  3. By align the right edge.

Note:

In other words, this component is just a frame. It is important not to forget to place it in the background, or the components that will be located inside the frame, put in the foreground. Otherwise, the frame may obscure other components, as the components overlap in the order in which they are placed in the workspace.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

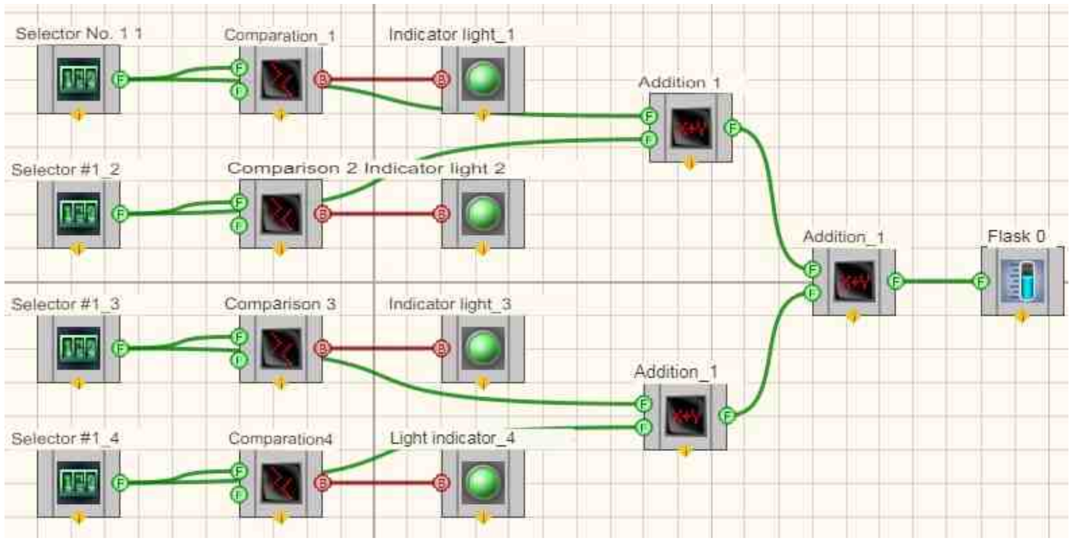
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

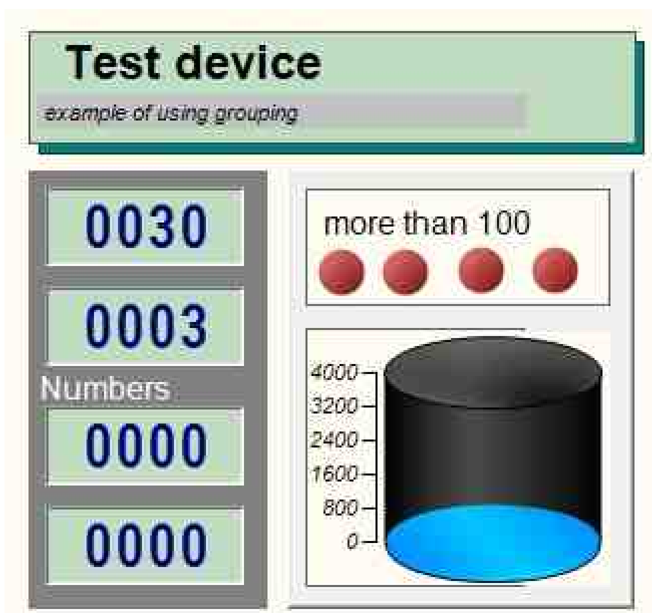
- BSTR Text - Set the text inscription (string).
- BSTR TextPosition - Set text position: Horizontal; Rotatation 90°; Rotatation -90° (string).
- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- LONG TextColor - Set text color (any number).
- LONG FrameColor - Set frame color (any number).
- LONG BackColor - Set the background color (any number).
- VARIANT_BOOL Transparent - Set the display status to transparent:
  true - the display status to transparent is turned ON;
  false - the display status to transparent is turned OFF.
- BSTR Alignment - Set the alignment: By align the left edge; In the middle; By align the right edge (string).

Example

Project in SCADA ZETView



Project operation result

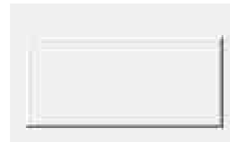


View the example in ZetView

10.2.Windows Panel Style

The component "**Windows Panel Style**" is used as a comment field. In allows to make text notes for the operator.

Appearance of the component:

Developer interface	Operator interface
Doesn't have.	

Setting:

➢Input:

Doesn't have.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

Doesn't have.

Note:

In other words, this component is just a block. It is important not to forget to put it in the background, or the components that will be located inside the block, put in the foreground. Otherwise, the block may cover other components, as the components overlap in the order in which they are placed in the workspace.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

➢Custom common properties:

- Are available by the link below. 154

Example

Example for use **Windows style** (is in the Component Example Combine a group 568)

Глава 11.Measurements (External devices)

Components of "Measurements (External devices)" section are used for measurements of various parameters of signals received at the input modules of ADC-DAC channels, Narrow-band spectrum program, strain-gauge stations and virtual channels created by such programs as ZETFormula, Signal generator, Filtration by channels, etc:

Attention!

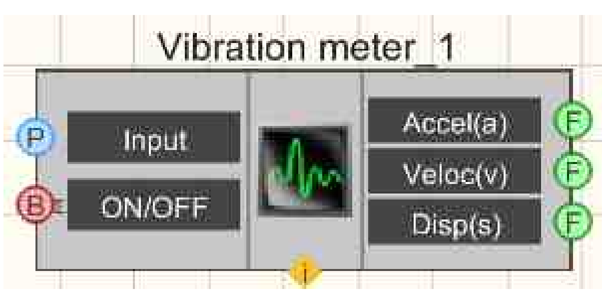
Some of the measurement programs (DC/AC voltmeter, vibration meter and some others) can operate in multichannel mode in the case if they are loaded by means of UNIT interface. In this mode the program can perform measurements by several channels simultaneously (max amount of channels is 250). Each channel of the program may have individual parameters (averaging, channel sampling frequency, vibration meter, band filter, etc.). Measurements results are available in parent program via UNIT interface.

The vibration meter operates only with accelerometers channels; it identifies them by measurements unit: "g" or "m/sI".

11.1.Vibration meter

The component "**Vibration mete**" is designed to measure RMS and peak values of Acceleration, Velocity and Displacement using piezo sensors (accelerometers) connected to input channels of signal analyzers.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - input channel, to which receives the signal from which the required values need to be measured.
- ON\OFF - turn on and off the vibrometer.

➢ Output:

- Accel(a) - current average value of acceleration over the set time.
- Speed(v) - current average velocity value for the set time.
- Disp(s) - the current average value of the displacement for the set time.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- averagetime (1) - the time for which the average value is calculated, s:
  1. 0.1 s
  2. 1 s.
  3. 10 s.
- measuretype (Peak) - the type of measurements taken, V:
  1. RMS - root mean square value.
  2. Peak - the maximum value during the entire operation time (peak value >= amplitude value).
  3. Amplitude - the maximum value of the displacement or change of a variable from the mean value.
- ReactionTime (0) - minimum server time after which the component will start working, s.
- Activate (true) - the operation status (enabled/disabled).
- fir (0.5 ч 200) - useful signal transmission filter:
  1. 0.5 ч 200 Hz.
  2. 10 ч 1000 Hz.
  3. 3 ч 10000 Hz.
  4. 1 ч 10 Hz.
- NeedVelocity (true) - Velocity data is required.
- NeedMotion (true) - Need motion data.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - an input channel that receives a signal for which you want to measure the required values (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR AverageTime - Set the average: 0.1 s; 1 s; 10 s (string).

  Note: In multi-channel mode, setting the averaging time for the current vibrometer channel number.

- BSTR MeasureType - Set the measurement type: RMS; Peak; Amplitude (string). In multi-channel mode, Set the type for the current channel number of the vibrometer.
- FLOAT ReactionTime - Set the minimum server time to work, s (0).
- VARIANT_BOOL Activate - Set the operation status vibrometer:
  true - the operation status of the vibrometer is enabled;
  false - the operation status of the vibrometer is disabled.
- BSTR FIR - Set the filter: 0.5 ч 200 Hz; 10 ч 1000 Hz; 3 ч 10000 Hz; 1 ч 10 Hz (string).
- VARIANT_BOOL NeedVelocity - Set the flag for the need to calculate velocity data:
  true - the need to calculate velocity data is enabled;
  false - the need to calculate velocity data by speed is disabled.
- VARIANT_BOOL NeedMotion - NeedMotion - Set the flag for the need to calculate movement data:
  true - the need to calculate displacement data is enabled;
  false - the need to calculate displacement data is disabled.

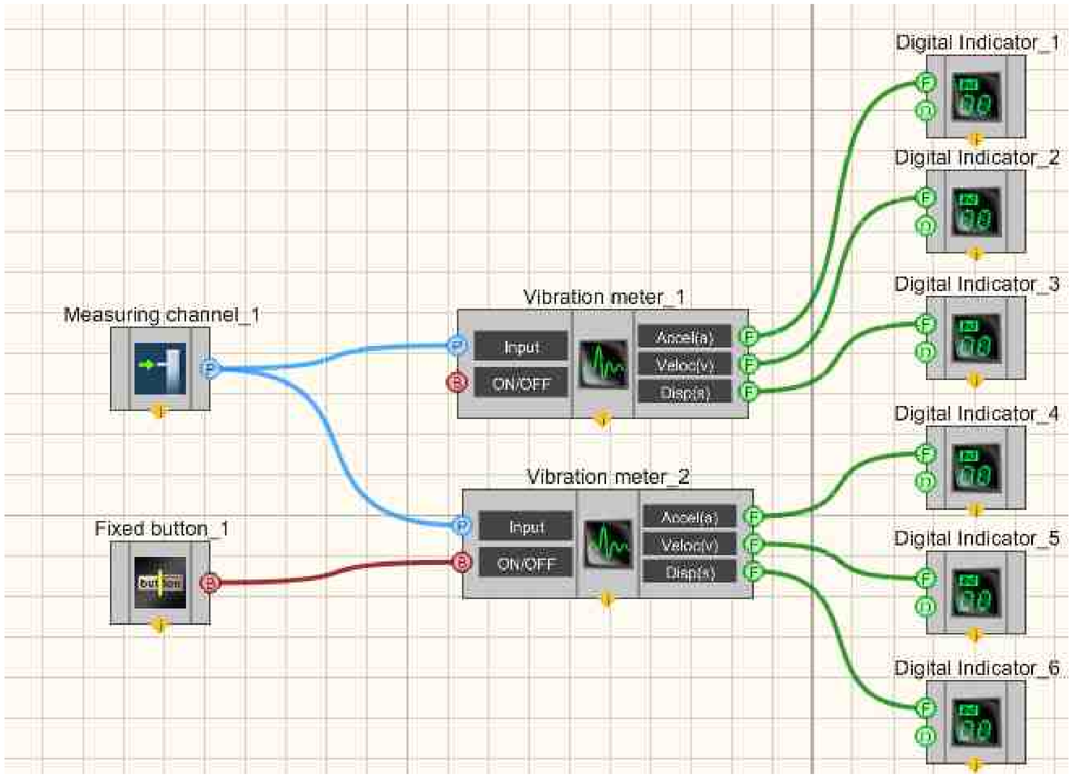
Example

Attention!

Some measuring programs (DC and AC voltmeter, vibration meter, and also, in the future, some others) when downloaded using the UNIT interface (onExample, in ZETView) can work in multi-channel mode. In this mode, the program can perform its measurements not on one, but simultaneously on several channels (no more than 250). In this case, each program channel can have its own parameters, such as averaging time, channel sampling frequency, type of bandpass filter used for the vibrometer, etc. The measurement results are available to the parent program via UNIT.

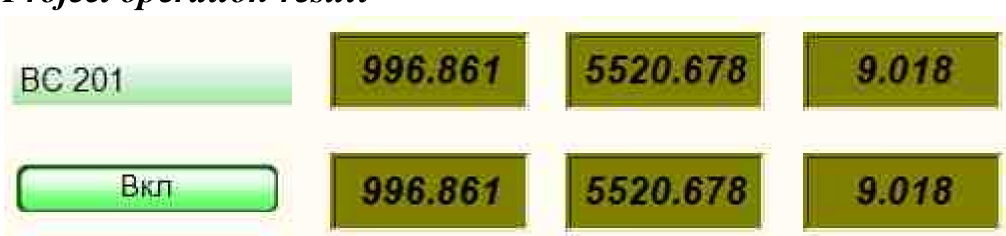
The vibrometer works only with accelerometer channels, and it recognizes them by units of measurement: "g" or "m/sI".

Project in SCADA ZETView



In the above project, the multichannel Vibrometer 572 component measures the values of the signal coming from the Measuring channel 375. A Fixed button 691 is used to enable or disable the vibrometers. Digital Indicators 682 are used for graphical representation of the value.

Project operation result



View the example in ZetView

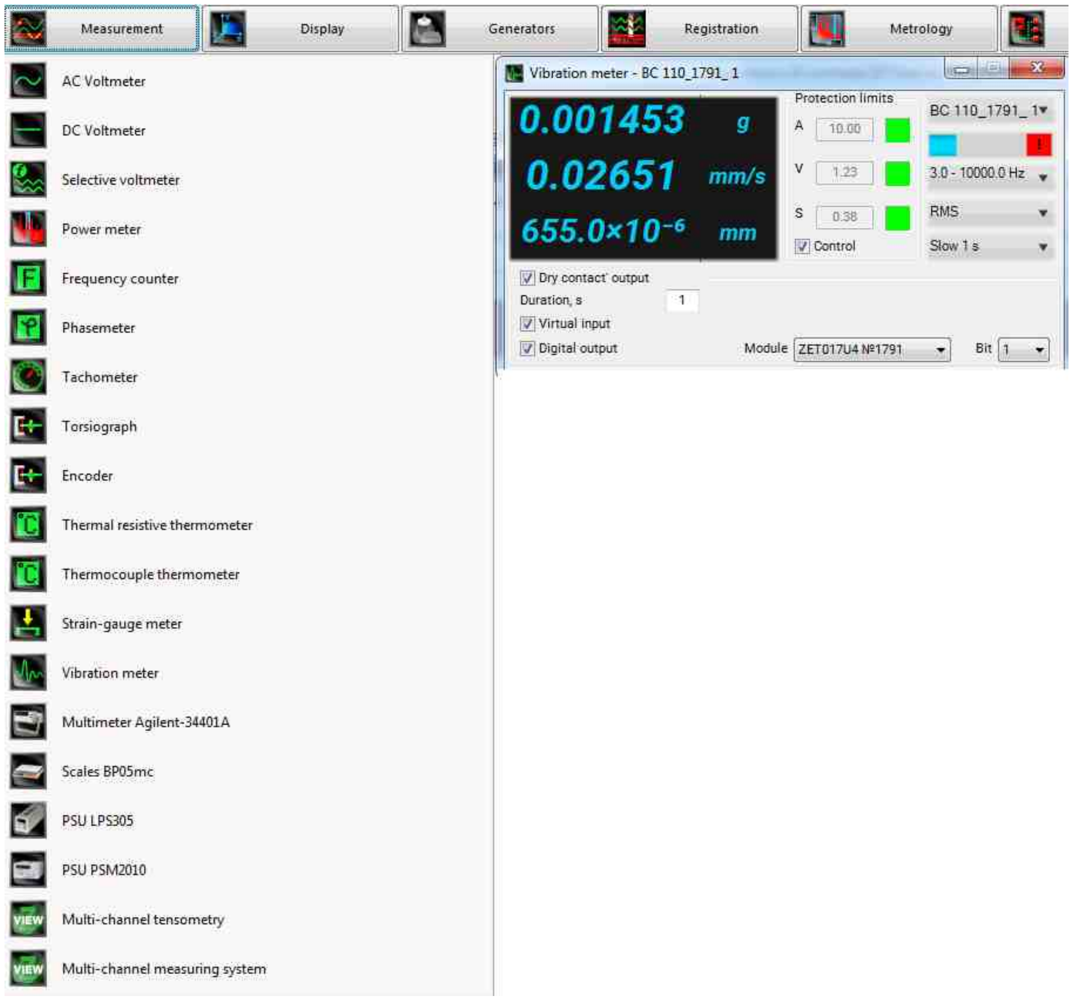
More details about the program "Vibrometer":

The program is designed to measure vibration parameters from vibration sensors connected to the input channels of spectrum analyzers.

The following Settings are measured in the program: Acceleration, Velocity and Displacement.

To measure vibration parameters, you must select the channel of the spectrum analyzer to which the vibration transducer is connected. In the list of channels, only those channels of the spectrum analyzer are available, the dimension of which is the value of vibration acceleration: g or m/s^2 (1 g = 9.807 m/s^2 or 10 m/s^2 = 1.02 g). The value of vibration acceleration is displayed in the upper line in the left part of the program window (parameter "A" - Acceleration). Further, by integrating the Acceleration, the Velocity and is calculated and displayed in the second line (parameter "V" - velocity - speed). By repeated integration, the value of the Displacement is obtained (parameter "S" - offset - displacement).

The measured values can be either RMS or Peak. When measuring RMS values, it is possible to change the averaging time - 0.1 s or 1 s.



In the "Thresholds" frame, the user sets the maximum allowable threshold values for all three parameters: vibration acceleration, vibration velocity, and vibration displacement. When the "Control" checkbox is checked, real-time monitoring of the excess of the current vibration parameters over the specified threshold levels begins. A color indicator informs the operator about the status of the vibration parameters. Green color indicates that the parameter is normal. When and exceeding the threshold levels for any of the parameters, the color indicator changes color to red. Additionally, a global message is sent to the system, which forces the output signal from the generator to turn off, if it is turned on. This option is another factor that provides additional equipment safety when used, in Example,

The program can be used as a vibrostop to shut down equipment in case of high vibration levels.

When carrying out measurements, it is possible to use common frequency filters:
HP1 - to measure vibration acceleration in the frequency range from 1 to 20 000 Hz;
HP3 - to measure vibration acceleration in the frequency range from 3 to 20 000 Hz;
HP10 - to measure vibration acceleration in the frequency range from 10 to 20 000 Hz.

These filters are used to measure vibration acceleration without correction. The HP1 and HP10 filters are used to remove low-frequency signal components from the measurement results, which are often associated with spurious interference and distortion.

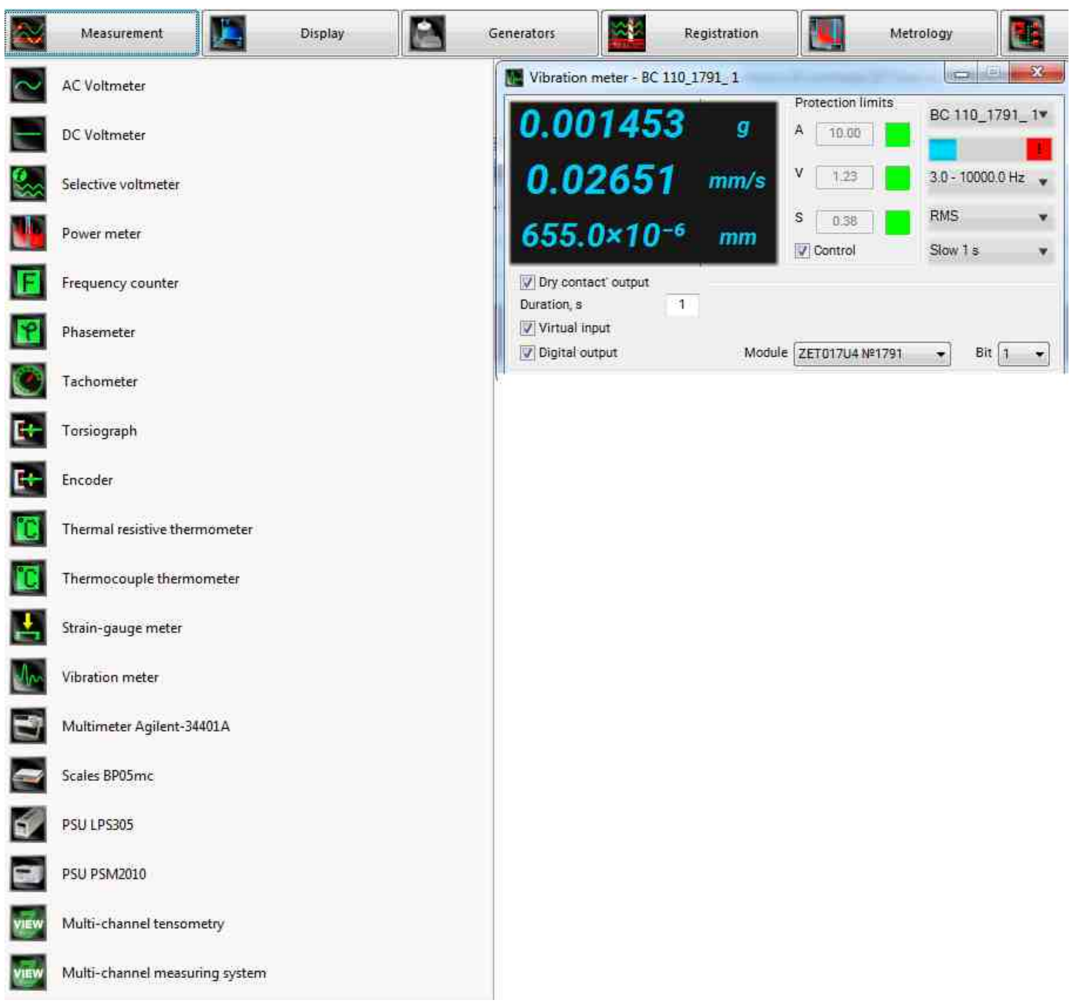
It is also possible to use integrating filters:
Vel1 - to measure vibration velocity in the frequency range from 1 to 330 Hz;
Vel3 - for measuring vibration velocity in the frequency range from 3 to 1000 Hz;
Vel10 - for measuring vibration velocity in the frequency range from 10 to 3000 Hz;
Dil1 - for vibration displacement measurement in the frequency range from 1 to 18 Hz;
Dil3 - for vibration displacement measurement in the frequency range from 3 to 57 Hz;
Dil10 - to measure vibration in the frequency range from 10 to 181 Hz;
VelMF (machine filter) - for measuring vibration velocity on stationary machines in the frequency range from 10 to 1000 Hz (in accordance with ISO 10816 and GOST 25275 (ST SEV 3173).

To measure the impact of vibration on the human body, it is possible to use frequency-correcting filters:
W-Bz - to measure the total vibration (corrected vibration acceleration) in the vertical plane of the Z axis (in accordance with ISO 8041 and CH 2.2.4/2.1.8.566-96);
W-Bxy - to measure the total vibration (corrected vibration acceleration) in the horizontal plane along the XY axes (in accordance with ISO 8041 and CH 2.2.4/2.1.8.566-96);
W-Bc - to assess the impact of vibration on a person on the back of the seats (according to ISO 2631 and ISO 8041);
Wk - to measure the total vibration affecting a person along the Z axis (in accordance with ISO 8041, ISO 2631-1 and GOST 12.1.012) (currently not used in Russia);
Wd - to measure the total transport vibration affecting a person along the XY axes (in accordance with ISO 8041, ISO 2631-1 and GOST 12.1.012) (currently not used in Russia);
Wh - to measure the general vibration affecting a person (in accordance with ISO 8041, ISO 2631-1 and GOST 12.1.012);
Wc - to assess the impact of vibration on a person through the seat back (in accordance with ISO 8041, ISO 2631-1 and GOST 12.1.012) (currently not used in Russia);
Wj - to assess the impact of vibration on the head of a lying person (in accordance with ISO 8041, ISO 2631-1 and GOST 12.1.012) (currently not used in Russia);
HA - to measure local vibration (corrected vibration acceleration) acting on parts of the human body along the three XYZ axes (in accordance with ISO 5349 and CH 2.2.4/2.1.8.566-96);
KB - for measuring general vibration on sea and river vessels (CV = W-Bc + 28.9 dB).

Three types of filters are currently used in Russia for certification of workplaces and sanitary and hygienic assessment: W-Bz, W-Bxy, HA.

Purpose of the program

The Vibrometer program is designed to measure the RMS and peak values of Acceleration, Velocity and Displacement using piezo sensors (accelerometers) connected to the input channels of signal analyzers. The Vibrometer program creates virtual channels of instantaneous values of Velocity and Displacement. The Velocity and vibration signals are low-pass filtered with a cutoff frequency of 3 dB 1000 Hz and high-pass filtered with a cutoff frequency of 3 dB 10 Hz. These channels are available for further analysis in other ZETLab programs.
Program description
To run the Vibrometer program in the Measurement menu of the ZETLab panel, select the Vibrometer command. The working window of the Vibrometer program will be displayed on the monitor screen. The title of the program window will display the name of the program and the name of the channel on which the measurements are made.



Note: The Vibrometer program can be launched directly from the ZETLab working directory (by default: c:\ZETLab\). Executable file name: VibroMeter.exe

In the left part of the working window of the Vibrometer program there is a graphical indicator, which displays the values of Acceleration - opposite the letter A, Velocity - opposite the letter V and Displacement opposite the letter S.

In the Protection thresholds frame, the maximum allowable levels of Acceleration, Velocity and Displacement are set in the fields to the right of the inscriptions A, V and S, respectively. Values are entered from the keyboard. The separator symbol for fractional numbers is a dot.

To enable the vibration stop mode, you need to set the thresholds and enable the Control checkbox. In this mode, when the specified thresholds for acceleration, speed or displacement are exceeded, the indicator located to the right of the corresponding threshold changes its color from green to red and the program issues a global message that can be used, in Example, to automatically stop the equipment.

On the right, in the upper field with an arrow (list) - the signal input channel is selected. In the Vibrometer program, channels with g or m/sI units are available, including virtual channels (generated by such programs as ZETFormula or Signal Filtering).

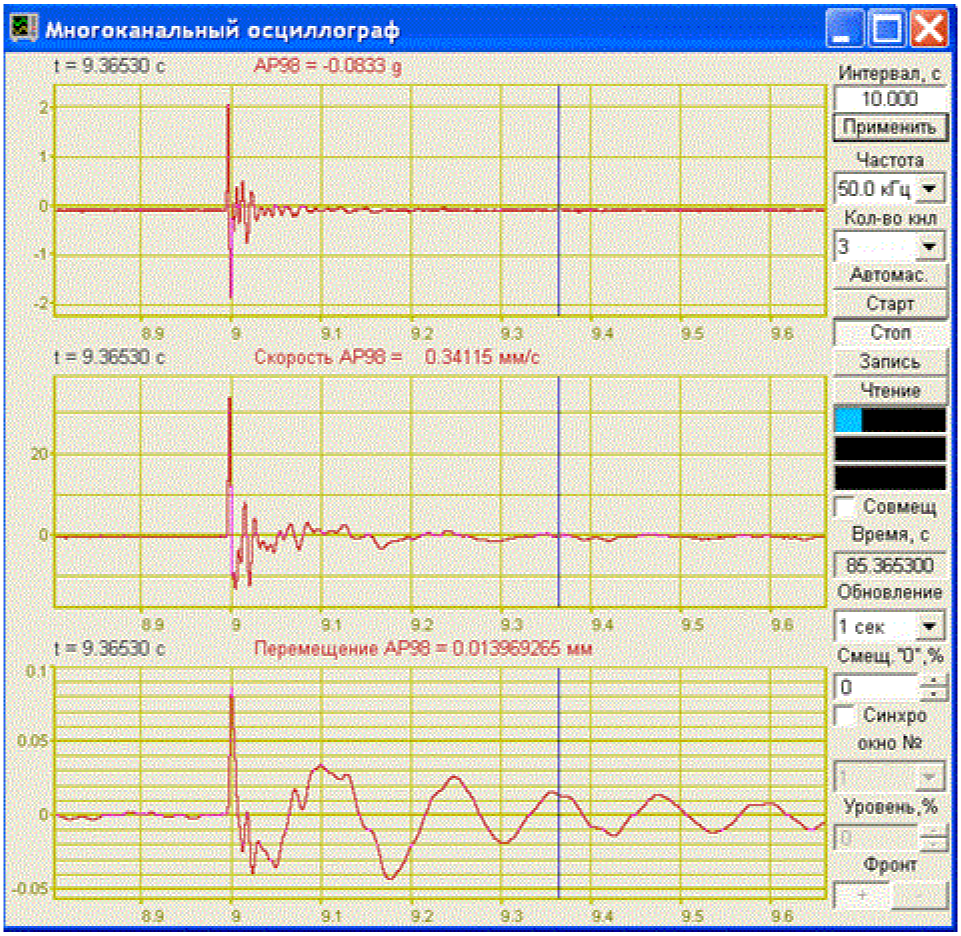
In the fields with arrows (lists), you can select parameter values in two ways:
- click on the field arrow and select the desired value from the drop-down list with the "mouse";
- click the left mouse button on the field and use the mouse roller or the keyboard arrow keys to select the desired element.

The indicator Integral signal level shows the level and overload. If the signal level exceeds the maximum allowable level, the indicator turns completely red, without the black right side. The right edge of the indicator remains red until the user clicks on it with the left mouse button.
Below there is a field with an arrow (list), in which the range for pre-filtering the signal is selected.

In the Value group, you can select the root mean square (RMS), amplitude or peak value of the displayed values.

In the Averaging group, you can choose to average 0.1 second, 1 second, or 10 seconds.
To exit the program, click the cross in the upper right corner of the window.

The figure below shows the vibration shock waveform. on the upper graphic is the acceleration signal, on the middle graphic is the Velocity signal, on the lower graphic is the Displacement signal.



Mathematical description

Vibration is the mechanical vibrations of the body.

The simplest form of vibration is the oscillating or repetitive movement of an object around an equilibrium position. This type of vibration is called general vibration, because the body moves as a whole and all its parts have the same speed in magnitude and direction. The equilibrium position is the position in which the body is at rest or the position that it will occupy if the sum of the forces acting on it is zero.

FREE VIBRATIONS - (natural oscillations), oscillations in mechanical, electrical. or k.l. other system, performed in the absence of external. impact due to the initially introduced energy (potential or kinetic).

Forced oscillations are oscillations that occur under the influence of external forces that change over time.

Conservative (physical system, the work of non-conservative forces of which is equal to zero and for which the law of conservation of mechanical energy takes place) harmonic oscillator.

Newton's second law for such an oscillator can be written as:

$ma = -kx + F_0 \cos(\Omega t)$. If we introduce the notation:

$\omega_0^2 = \frac{k}{m}, \Phi_0 = \frac{F_0}{m}$,

and replace the acceleration with the second derivative of the coordinate with respect to time, we get the following ordinary differential equation:

$\ddot{x} + \omega_0^2 x = \Phi_0 \cos(\Omega t)$.

The solution of this equation will be the sum of the general solution of the homogeneous equation and the particular solution of the inhomogeneous one. The general solution of the homogeneous equation has already been obtained here and it has the form:

$x(t) = A * \sin(\omega_0 t + \varphi)$,

where A,φ - arbitrary constants, which are determined from the initial conditions.

Let's find a particular solution. To do this, we substitute a solution of the form: x(t) =B*cos(Ωt) into the equation and get the value for the constant:

$B = \frac{\Phi_0}{\omega_0^2 - \Omega^2}$.

Then the final solution will be written as:

$x(t) = A * \sin(\omega_0 t + \phi) + \frac{\Phi_0}{\omega_0^2 - \Omega^2} \cos(\Omega t)$.

Resonance.

It can be seen from the solution that when the frequency of the driving force is equal to the frequency of free oscillations, it is not suitable - a resonance occurs, that is, an "unlimited" linear increase in amplitude with time. From the course of mathematical analysis it is known that the solution in this case must be sought in the form:

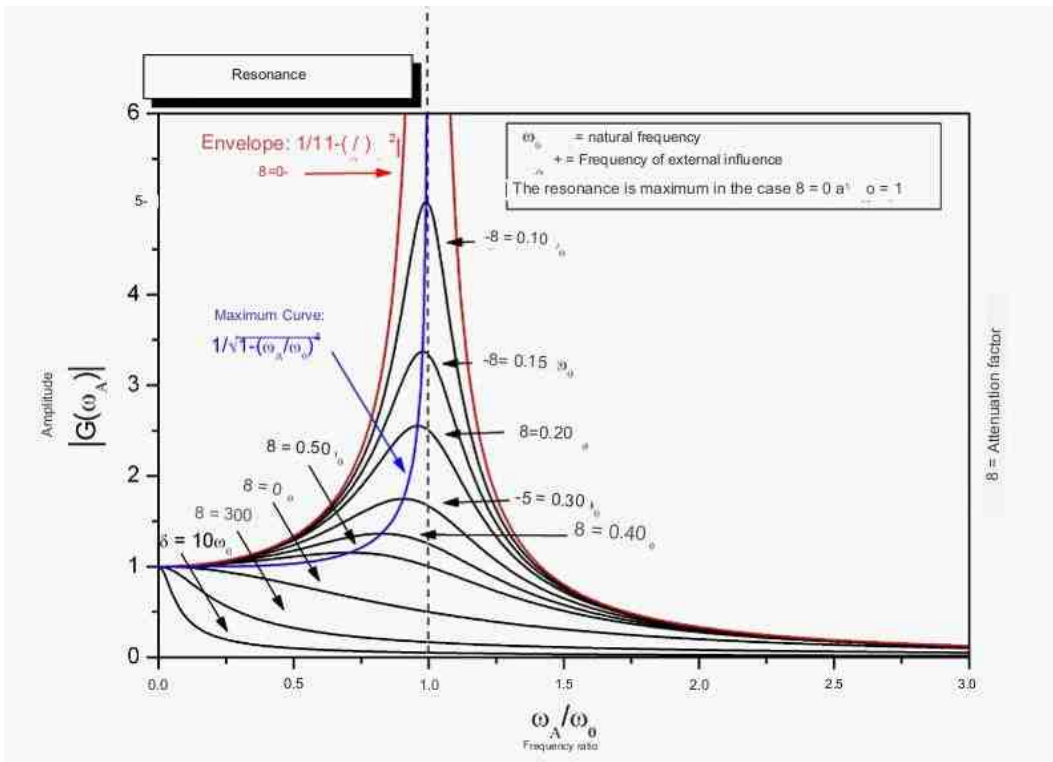
$x(t) = T(A * \sin(\Omega t) + B * \cos(\Omega t))$.

We substitute part of the expression into the differential equation and get:

$A = 0,\ B = \frac{\Phi_0}{2\Omega}$.

Thus, oscillations at resonance will be described by the following relationship:

$x(t) = \frac{\Phi_0}{2\Omega} * t * \sin(\Omega t)$.



Damped harmonic oscillator.

Newton's second law:

$ma = -kx - \alpha v + F_0 * \cos(\Omega t)$.

Redesignations:

$\omega_0^2 = \frac{k}{m},\ \Phi_0 = \frac{F_0}{m},\ \zeta = \frac{\alpha}{2\sqrt{km}}$.

Differential equation:

$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = \Phi_0 \cos(\Omega t)$.

Its solution will be constructed as the sum of solutions of a homogeneous equation and a particular solution of an inhomogeneous one. An analysis of the homogeneous equation is given here. We obtain and analyze a particular solution.

We record the driving force as follows:

$$\Phi_0 \cos \Omega t = \Phi_0 \operatorname{Re} e^{-i\Omega t},$$

then we will look for the solution in the form:

$$x(t) = A * e^{-i\Omega t}, A \in C.$$

Substitute this solution into the equation and find an expression for A:

$$A = \frac{\Phi_0}{\omega_0^2 - \Omega^2 - 2i\beta\Omega} = \frac{\Phi_0(\omega_0^2 - \Omega^2 + 2i\beta\omega)}{(\omega_0^2 - \Omega^2) + 4\beta^2\Omega^2} = |A| * e^{-i\varphi},$$

Where $|A| = \frac{\Phi_0}{\sqrt{(\omega_0^2 - \Omega^2) + 4\beta^2\Omega^2}}, \varphi = -\arctan\frac{2\beta\Omega}{\omega_0^2 - \Omega^2}$.

The complete solution looks like:

$$x(t) = e^{-\zeta\omega_0 t}\left(C_1 \cos(\omega_d t) + C_2 \sin(\omega_d t)\right) + \operatorname{Re}\left[\frac{\Phi_0(\omega_0^2 - \Omega^2 + 2i\beta\Omega)}{(\omega_0^2 - \Omega^2)^2 + 4\beta^2\Omega^2} e^{-i\Omega t}\right],$$

Where $\omega_d = \omega_0\sqrt{1 - \zeta^2}$ - natural frequency of damped oscillations.

The constants C1 and C2 in each case are determined from the initial conditions:

$$\begin{cases} x(0) = x_0 \\ \dot{x}(0) = v_0 \end{cases}$$

In this case, in contrast to a frictionless oscillator, the oscillation amplitude at resonance has a finite value.

If we consider a stable process, that is, the situation at $t \to \infty$, then the solution of the homogeneous equation will tend to zero and only a particular solution will remain:

$$x(t \to \infty) = \Phi_0 \frac{(\omega_0^2 - \Omega^2)\cos\Omega t + 2\beta\Omega \sin\Omega t}{(\omega_0^2 - \Omega^2) + 4\beta^2\Omega^2} = \frac{\Phi_0}{\sqrt{(\omega_0^2 - \Omega^2) + 4\zeta^2\omega_0^2\Omega^2}} \cos(\Omega t - \varphi).$$

This means that at t→∞ the system "forgets" the initial conditions, and the nature of the oscillations depends only on the driving force. The work done by the driving force F(t) $=F_0\cos(\Omega t)$ over time dt is equal to Fdx, and the power $P = F\frac{dx}{dt}$. From the equation

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = \Phi_0 \cos(\Omega t),$$

Follows that

$$P(t) = F\dot{x} = (\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x)m\dot{x}.$$

If we take into account that with steady forced oscillations:

$$x = A * \cos(\Omega t - \varphi)$$

$$\dot{x} = -A\Omega \sin(\Omega t - \varphi)$$

$$\ddot{x} = -A\Omega^2 \cos(\Omega t - \varphi)$$

Then the average for the period $T = \frac{2\pi}{\Omega}$ power:

$$P = \frac{m}{T}\int_0^T (\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x)\dot{x} = A^2 m\zeta\omega_0\Omega^2$$

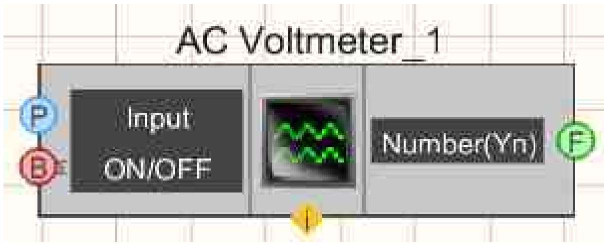
Work for the period:

$$A = m\int_0^T (\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x)\dot{x} = A^2 m\zeta\omega_0\Omega^2 T$$

11.2.AC Voltmeter

The component "**AC Voltmeter**" is designed to measure the RMS (true RMS) and peak values of the signal of the selected channel in the set units.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - input channel, to which receives the signal from which the required values need to be measured.
- ON\OFF - turn on and off the AC voltmeter.

➢ Output:

- Number (Yn) - output channel from which already measured values of the input signal arrive in the stream.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- averagetime (1 s) - the time for which the average value is calculated, s:

1. 0.1 s (with RMS and peak correctly measured for a signal with a frequency of at least 20 Hz).
2. 1 s (with RMS and peak measured correctly for a signal with a frequency of at least 2 Hz).
3. 10 s (with RMS and peak measured correctly for a signal with a frequency of at least 0.2 Hz).

- measuretype (RMS) - Set the measurement type:

1. RMS - RMS value of the signal.
2. Peak - the maximum value during the entire operation time (peak value >= amplitude value).
3. Amplitude - the maximum value of the displacement or change of a variable from the mean value.

- Activate (true) - the operation status (enabled/disabled).
- scale (Linear) - scale of data reading:

1. Linear.
2. Decibel.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - the channel for which you want to measure the required values (from 0 to (number of channels - 1)).
-

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

BSTR AverageTime - Set the average: 0.1 s; 1 s; 10 s (string).

- ***Note:*** In multi-channel operation mode, Set the average time for the current voltmeter channel number.

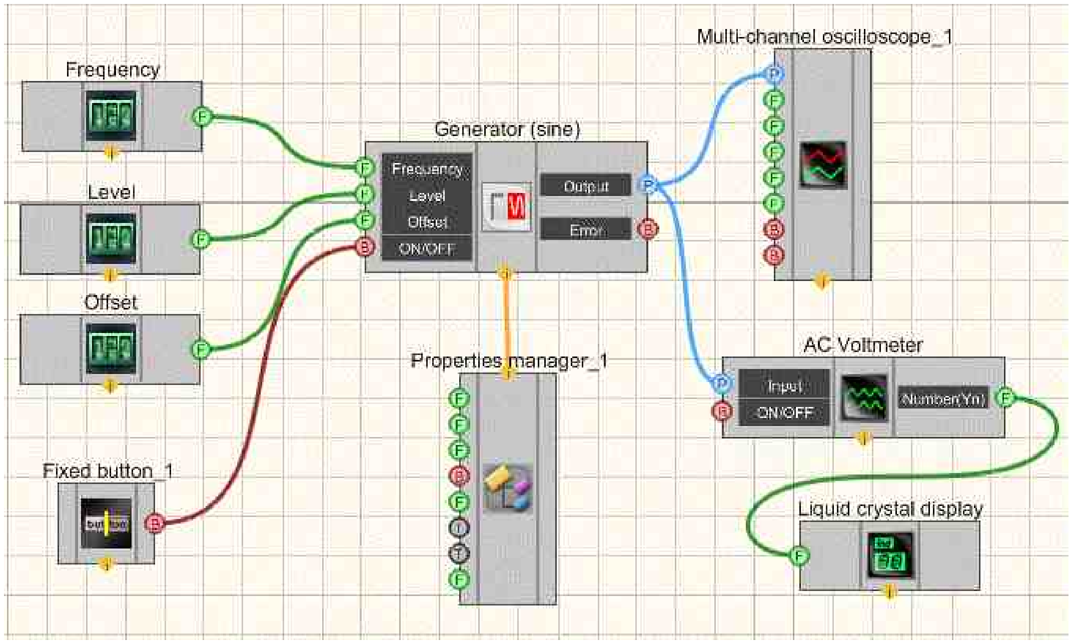
- BSTR MeasureType - Set the measurement type: RMS; Peak; Amplitude (string). In multi-channel operation mode, Set the type of transmitted value for the current channel of the voltmeter.

- VARIANT_BOOL Activate - Set the operation status of the voltmeter:
  true - the operation status of the voltmeter is on;
  false - the operation status of the voltmeter is off.

- BSTR Scale - Set the data reading scale: Linear; Decibel (string). In multi-channel operation mode, type setting type for the current channel.

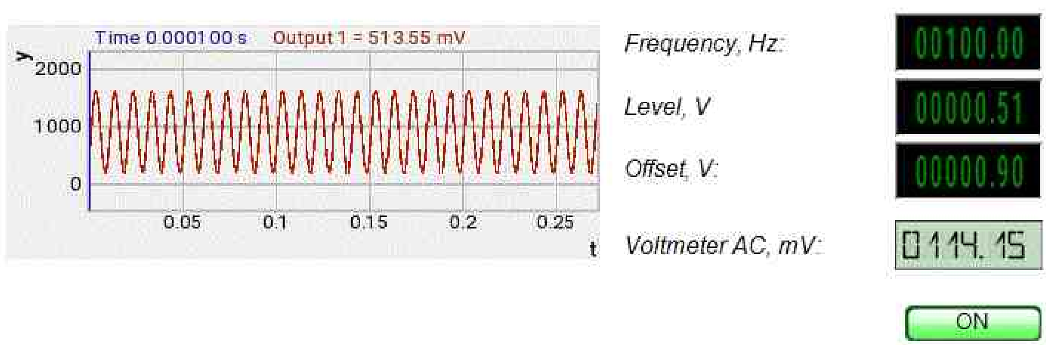
Example No. 1

Project in SCADA ZETView



This project shows the operation of the AC Voltmeter[585] component measures the value of the AC voltage coming from a Sine signal[585]. The Selectors[428] are used to set the frequency, level and offset of the generator. Further, the measured value is sent to the Liquid crystal display[655] for graphical presentation of information.To view the generated signal, use the Multi-channel oscilloscope[928] component. The result of the compiled program can be seen in the operator interface. The Fixed button[441] is used to turn on or off.
The Property manager[999] is used to change the indicators in real time and transmit data to the signal generator to display them on a Multi-channel oscilloscope.

Project operation result



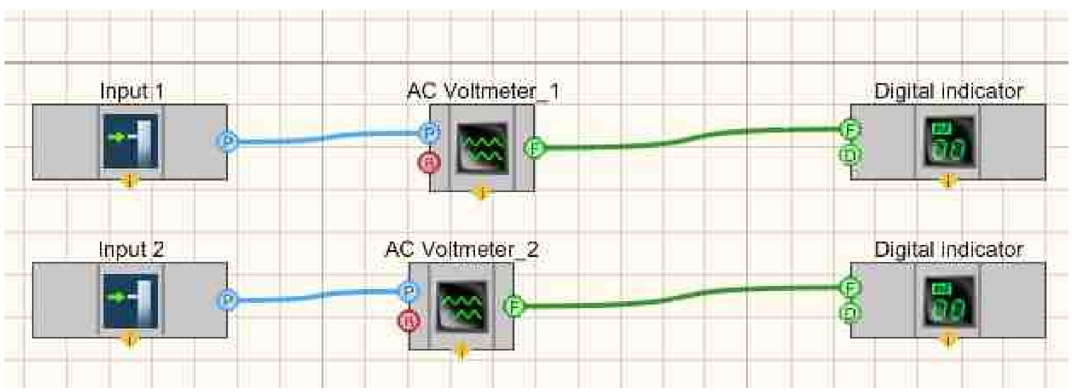
View the example in ZetView

Example No. 2

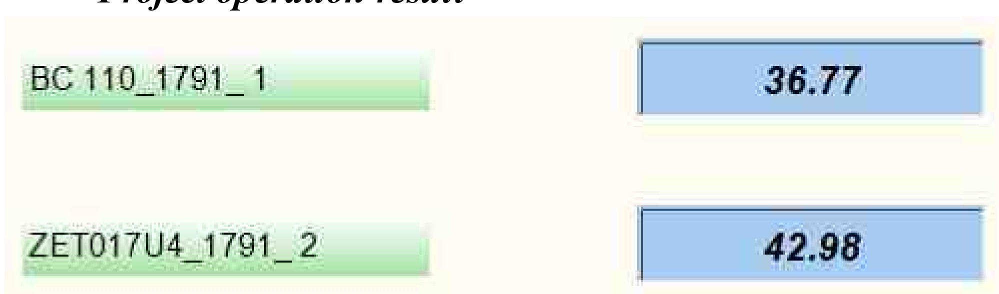
Attention!

Some measuring programs (DC and AC voltmeter, vibration meter, and also, in the future, some others) when downloaded using the UNIT interface (onExample, in ZETView) can work in multi-channel mode. In this mode, the program can perform its measurements not on one, but simultaneously on several channels (no more than 250). In this case, each program channel can have its own parameters, such as averaging time, channel sampling frequency, type of bandpass filter used for the vibrometer, etc. The measurement results are available to the parent program via UNIT.

Project in SCADA ZETView



Project operation result



View the example in ZetView

Mathematical description

When describing variable signals, the following quantities are often used.

Average or the arithmetic mean is defined as follows:

$$\bar{v} = \frac{1}{T}\int_{t_0}^{t_0+T} v(\tau) = \frac{1}{T}\int_0^T v(\tau)\,d\tau$$

.

This value is equal to the area under the curve of the time function calculated over one period. Because the function is periodic, the arithmetic mean v is independent of the starting point t0. For harmonic functions, the mean value of v is zero.

Modulo mean value defined as:

$$\bar{v} = \frac{1}{T}\int_0^T |v(\tau)|\,d\tau$$

.

This value is equal to the average value of the signal.

Note: The modulo average value is used when calculating the charge value of a capacitor, when working with rectified signals, or in calculations of electrolytic processes. The choice of rectifier type is also based on the modulo average current value, since the drop across the diode is almost always constant.

For a sinusoidal voltage, the following is true:

$$\bar{v} = \frac{1}{T}\int_0^T \hat{v}|\sin\omega t|\,dt = \frac{2}{T}\hat{v}\int_0^{T/2} \sin\omega t\,dt = \frac{1}{\pi}\hat{v}[-\cos\omega t]_{\omega t=0}^{\omega t=\pi} = \frac{2}{\pi}\hat{v} \approx 0.637\hat{v}$$

A similar relationship can be written for a sinusoidal current.

Root Mean Square (RMS) voltage characterizes the power in the alternating current circuit. By definition, the RMS function is equal to:

$$V_{СКЗ} = \sqrt{\frac{1}{T}\int_0^T v^2(t)\,dt}$$

.

For a sinusoidal voltage, the following is true:

$$V_{СКЗ} = \sqrt{\frac{1}{T}\int_0^T (\hat{v}\sin\omega t)^2\,dt} = \sqrt{\frac{1}{T}\hat{v}^2\int_0^T \sin^2\omega t\,dt} = \frac{\hat{v}}{\sqrt{2}} \approx 0{,}707\hat{v}$$

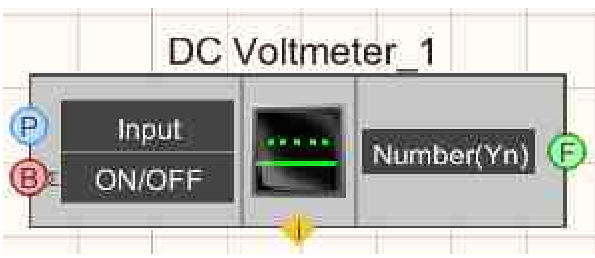
A similar relationship can be written for a sinusoidal current.

For variable signals, it is true that the RMS value is always less than or equal to the amplitude value.

11.3.DC Voltmeter

The component "**DC Voltmeter**" is designed to measure the level of the constant signal of the selected channel in the set units.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - input channel, to which the signal is received, for which the required values need to be measured.
- ON\OFF - turn on and off the DC voltmeter.

➢ Output:

- Number (Yn) - output channel from which already measured values of the input signal arrive in the stream.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- averagetime (1 s) - the time for which the average value is calculated, s:
  1. 0.1 s.
  2. 1 s.

3. 10 s.

- Activate (true) - the operation status (enabled/disabled).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Input - the channel for which you want to measure the required values (from 0 to (number of channels - 1)).

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

BSTR AverageTime - Set the average: 0.1 s; 1 s; 10 s (string).

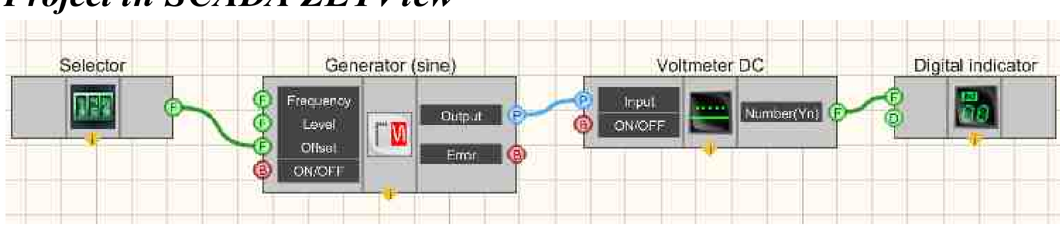
• ***Note:*** In multi-channel operation mode, set the average time for the current voltmeter channel number.

VARIANT_BOOL Activate - Set the operation status of the voltmeter:

- true - the operation status of the voltmeter is on;
  false - the operation status of the voltmeter is off.

Example No. 1

Project in SCADA ZETView



This project shows the operation of the DC Voltmeter 591 component measures the value of the DC component of a signal coming from a Sine signal 591 The Selector 428 is used to set the offset value (the measured DC component). Further, the measured value is sent to the Digital Indicator 682 for graphical presentation of information.

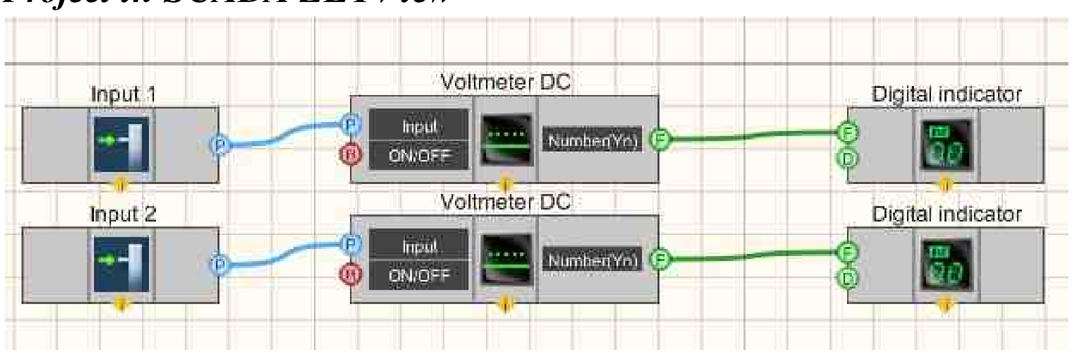
Project operation result

Example No. 2

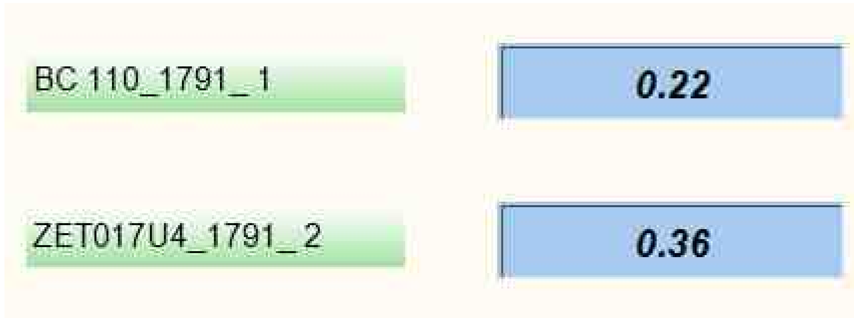
Attention!

Some measuring programs (DC and AC voltmeter, vibration meter, and also, in the future, some others) when downloaded using the UNIT interface (onExample, in ZETView) can work in multi-channel mode. In this mode, the program can perform its measurements not on one, but simultaneously on several channels (no more than 250). In this case, each program channel can have its own parameters, such as averaging time, channel sampling frequency, type of bandpass filter used for the vibrometer, etc. The measurement results are available to the parent program via UNIT.

Project in SCADA ZETView



Project operation result



View the example in ZetView

Mathematical description

Voltage (potential difference) between points A and B is the ratio of the work of the electric field during the transfer of electric charge from point A to point B to the value of the test charge.

$$\varphi_A - \varphi_B = U_{AB} = \frac{A^{\text{поле}}_{q:A\to B}}{q}$$

It is assumed that the test charge transfer does not change the distribution of charges on the field sources (according to the test charge definition).

Alternative definition (for an electrostatic field):

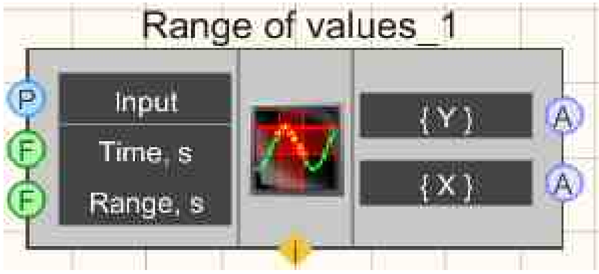
$$\varphi_A - \varphi_B = U_{AB} = \int_A^B (\vec{E}, d\vec{l}).$$

This is the integral of the field projection (field strength) over the distance between points A and B along any path going from point A to point B.

11.4.Range of values

The component "**Range of values**" is designed to obtain instantaneous channel signal values in a given time range.

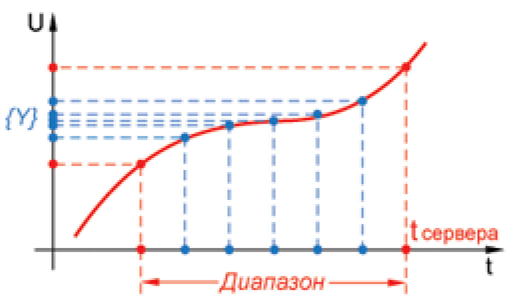
Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - the channel in which you want to get an array of instantaneous values.
- Time(s) - designed to get the server time value. This value is the last point in the range of transmitted values (server t in the figure below). The component fires when a value is received by this contact.
- Range(s) - the value of the time interval during which the values will be taken.

➢ Output:
- Channel Y - at the output we get a range of values.
- Channel X - array dimension.



Custom common properties:

➢Total properties (environment):
- [Are available by the link below.](154)

➢Custom private properties (the default value is shown in parentheses):
- Interval (1) - set the value of the time interval, s.
- Decimation (1) - set the decimation of the original signal: 1; 10; 100; 1000; 10000.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - measuring channel in which it is required to get an array of instantaneous values (from 0 to (number of channels - 1)).
- Time(s) - server time value. This value is the last point of the range of transmitted values (server t in the figure above). The component fires when a value is received by this contact.
- Range(s) - the value of the time interval during which the values will be taken.

➢**Custom common properties:**

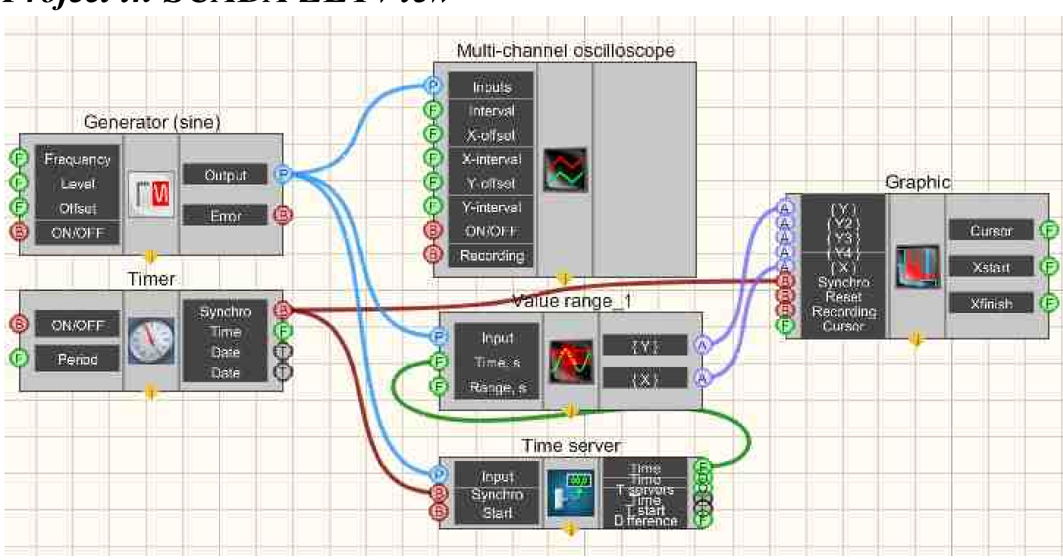
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Interval - Set the time interval, s (number).
- BSTR Decimation - Set the decimation of the original signal: 1; 10; 100; 1000; 10000 (string).

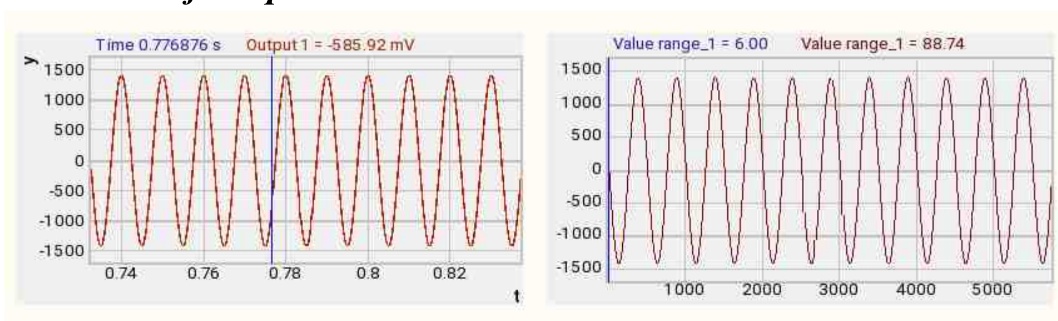
Example

Project in SCADA ZETView



This project shows the operation of the Range of values 594 component is designed to obtain instantaneous values of the channel signal in a given time range. The Timer 370 is used to supply synchronization pulses to the Server time 366 and Graphic 909. This is necessary so that the Graphic 909 and the Range of values 594 work in a single time space. The Sine signal 594 is needed to generate the test signal, and the Multi-channel oscilloscope 928 is for its current graphical display.

Project operation result

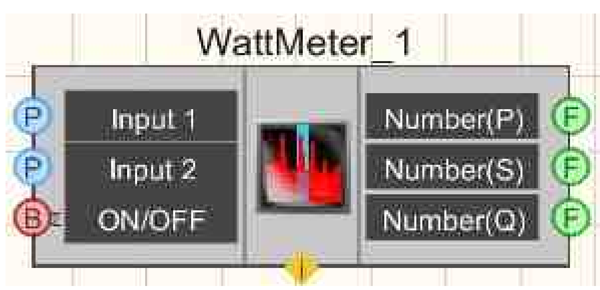


View the example in ZetView

11.5.Power meter

The component "**Power meter**" is designed to determine the power of an electric current or an electromagnetic signal.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Input 1 - incomingid of channel 1 of the voltage for which the required values are to be measured.
- Input 2 – incomingid of channel 2 id of the current for which the required values are to be measured.
- On/Off - turn the wattmeter on and off.

➤ Output:
- Number (P) - output channel from which already measured active power values are received.
- Number (S) - output channel from which the already measured values of the total power are received.
- Number (Q) - output channel from which already measured reactive power values are received.

Custom common properties:
➤Total properties (environment):
- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):
- averagetime (1 s) - set the average time:

1. 0.1 s (in this case, the power values are correctly measured for a signal with a frequency of at least 20 Hz).
2. 1 s (in this case, the power values are correctly measured for a signal with a frequency of at least 2 Hz).

- Activate (true) – the operation status (enabled/disabled).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤Input:
- Input 1 - voltage, for which the required values are to be measured (from 0 to (number of channels - 1)).
- Input 2 - current, which the required values are to be measured (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

BSTR AverageTime - Set the average: 0.1 s; 1 s (string).

- ***Note:*** In multi-channel operation mode, setting the averaging time for the current wattmeter channel number.

VARIANT_BOOL Activate - Set the status of the wattmeter:

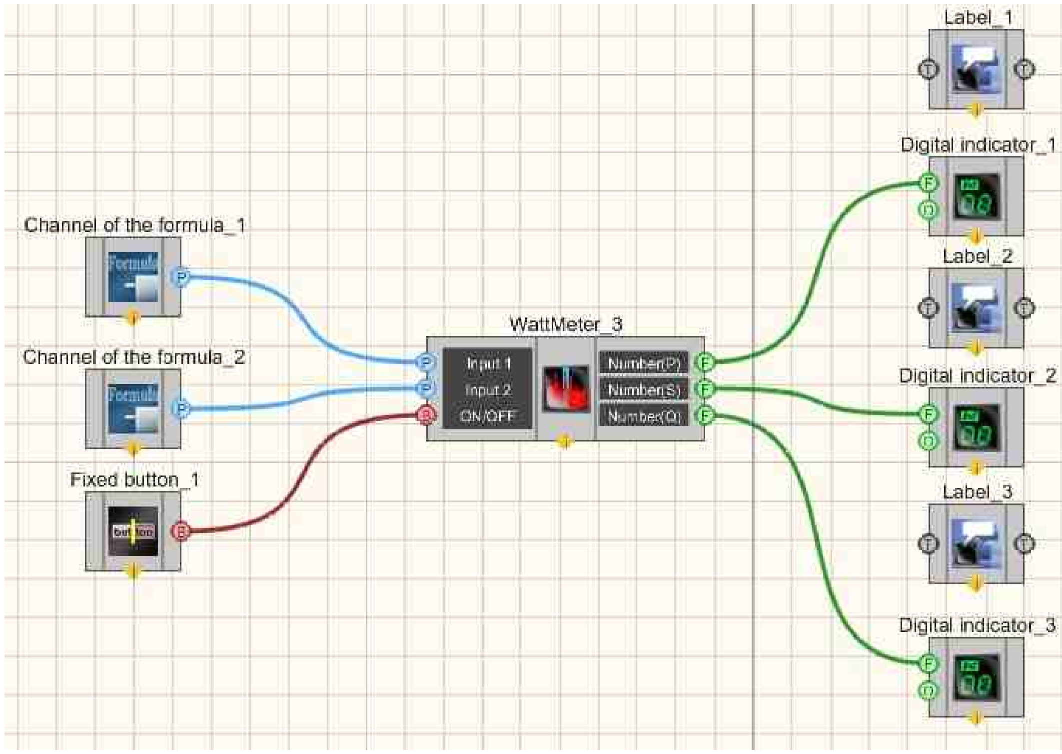
- true - the status of the wattmeter operation is enabled;
false - the status of the wattmeter operation is disabled.

Example

Project in SCADA ZETView

Attention!

Some measuring programs (wattmeter, AC voltmeter and DC voltmeter, vibration meter, and also, in the future, some others) when loaded using the UNIT interface (in Example, in ZETView) can work in multichannel mode. In this mode, the program can perform its measurements not on one, but simultaneously on several channels (no more than 128). In this case, each program channel can have its own parameters, such as averaging time, channel sampling frequency, type of bandpass filter used for the vibrometer, etc. The measurement results are available to the parent program via UNIT.



Project operation result

View the example in ZetView

11.6.Instantaneous value

The component "**Instantaneous value**" is designed to read the instantaneous value of the signal in the channel at a specified time.

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➤Input:

- Input - measuring channel in which the instantaneous values will be determined.
- Time - the values of the moments of time, the instantaneous value of which is required to be determined.

➤ Output:

- Number(Yn) - getting the instantaneous value of the channel signal at the required time.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- TimeWait (0.2 s) - limit desynchronization time, s.

Note: If the time on the channel lags behind the requested time (which may be the case for virtual channels), then the component will wait for some time.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - measuring channel, in which instantaneous values will be determined (from 0 to (number of channels - 1)).
- Time - values of moments of time, the instantaneous value of which is required to be determined (any number).

➢**Custom common properties:**

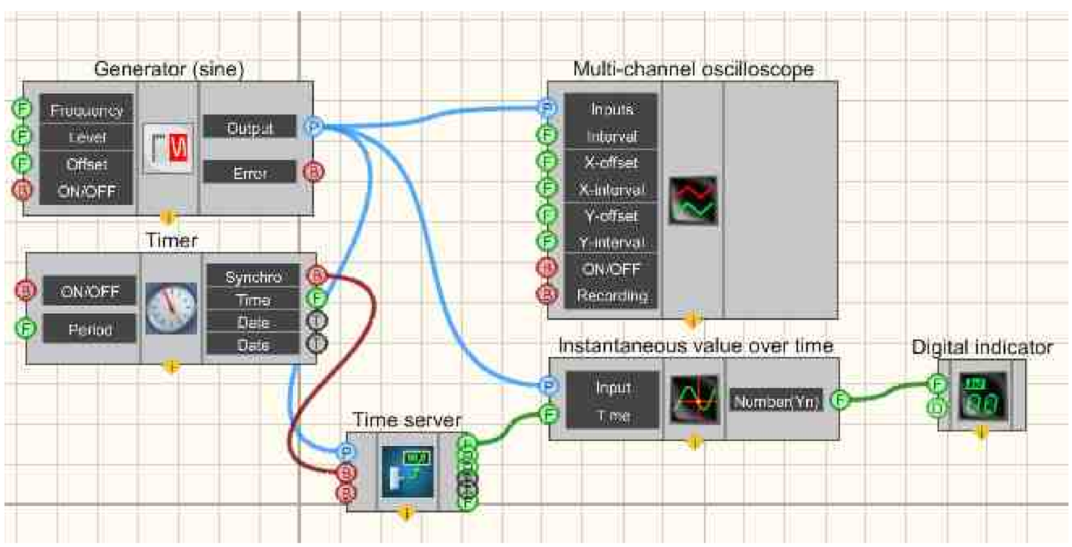
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

•FLOAT TimeWait - Set the limit desynchronization time, s (from 0 to 10).

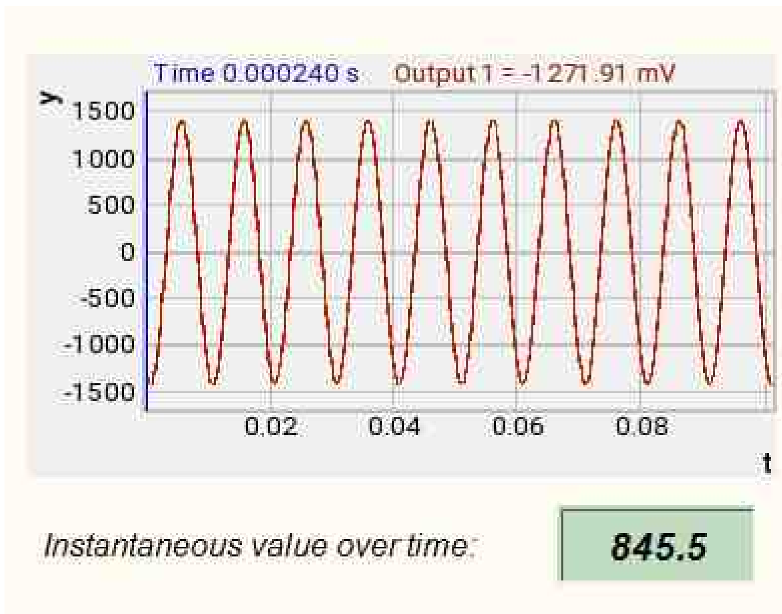
Example

Project in SCADA ZETView



This project shows the operation of the Sine signal 601 component outputs a sine waveform. A Multi-channel oscilloscope 928 is needed for graphical representation of the signal. The Timer 370 component generates clock impulses every 0.3 s, which are fed to the Server time 366 input. A Digital Indicator 682 is needed to display numerical information. The Instantaneous value over time input receives channel information and time from Server time 366 (with a frequency of once every 0.3 s, while each time at a given moment in time the Instantaneous value over time sends the numerical value of the signal).

Project operation result

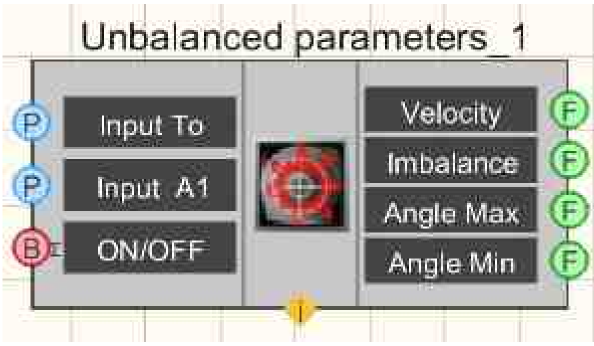


View the example in ZetView

11.7.Unbalanced parameters

The component "**Unbalanced parameters**" is designed to determine the inbalance parameters.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Input To - a channel containing the name of the RPM sensor channel.
- Input A1 - the channel with the accelerometer name No.1.
- . ON\OFF - turn on and off the unbalance parameters.

➤ Output:

- Velocity - rotational velocity, rpm.
- Unbalance - unbalance number, g * cm.
- Angle Max - maximum angle, degr. from 0 to 360.
- Angle Min - minimum angle, degr. from 0 to 360.

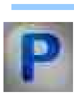
Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

- timeaverage (1.0 s) - averaging time, s.
- FreqBalance (12000) - nominal balancing frequency, rpm (frequency in Hz must be a multiple of ADC sampling frequency in Hz).
- Activate (1) - the operation status (enabled/disabled).
- MultiFactor (0.000130) - set the value of the multiplier factor.
- Degrefactor - (3,5) - set the value of the exponent.



Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Input To - name of the RPM sensor channel (from 0 to (number of channels - 1)).
- Input A1 - name of the channel with accelerometer No.1 (from 0 to (number of channels - 1)).

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR TimeAverage - Set the time for which the average value is calculated, s: 0.1; 0.2; 0.5; 1.0; 2.0; 5.0; 10.0 (string).
- FLOAT FreqBalance - Set the nominal balancing frequency, rpm (frequency in Hz must be a multiple of the ADC sampling frequency in Hz).
- LONG Activate - Set the operation status of the component:
  1 - the operation status of the unbalance parameters component enabled;
  0 - the operation status of the unbalance parameters component is disabled.

- FLOAT MultiFactor - Set the value of the multiplicative factor (number).
- FLOAT DegreFactor - Set the value of the exponent (number).

11.8.Selective voltmeter

The component "**Selective voltmeter**" is designed to measure root mean square (RMS, True RMS) and peak (peak-to-peak) value of AC voltage at the main (carrier) signal frequency. A feature of the selective voltmeter is the exclusion of the influence of harmonics on the readings.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - input channel, to which the signal is received, for which the required values need to be measured.
- ON\OFF - turn on and off the selective voltmeter.

➢ Output:
- Voltage - an output channel from which already measured values of the input signal are received in the stream.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- averagetime (1 s) - the time for which the average value is calculated, s:

1. 0.1 s (with RMS and peak correctly measured for a signal with a frequency of at least 20 Hz).
2. 1 s (with RMS and peak measured correctly for a signal with a frequency of at least 2 Hz).
3. 10 s (with RMS and peak measured correctly for a signal with a frequency of at least 0.2 Hz).

- AutoFrequency (false) - automatic setting of the voltmeter to the measured frequency.
- Frequency (0) - the frequency at which the voltmeter will make measurements, Hz
- FrequencyBar (0) - the frequency band at which the voltmeter will make measurements, Hz.
- Activate (true) - the operation status (enabled/disabled).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input - the channel for which you want to measure the required values (from 0 to (number of channels - 1)).

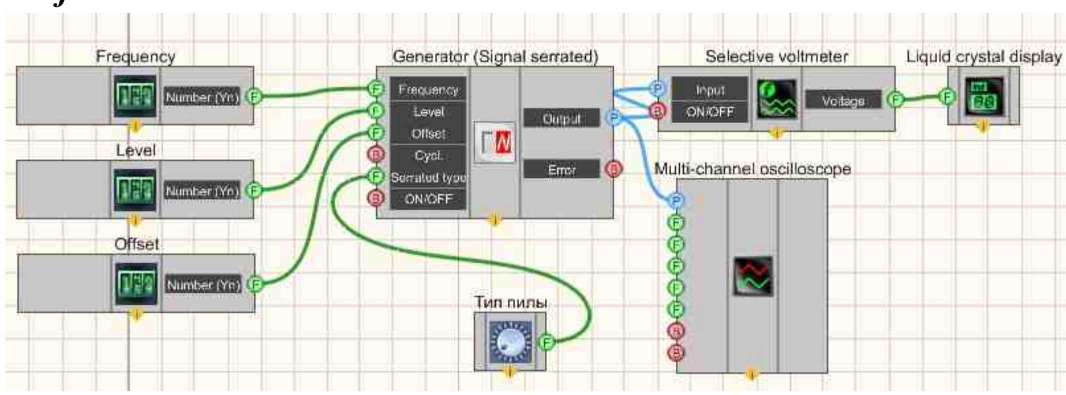
➢**Custom common properties:**
- Are available by the link below. 154

➢Custom Custom private properties (the default value is shown in parentheses):

- BSTR AverageTime - Set the average: 0.1 s; 1 s; 10 s (string).
- VARIANT_BOOL AutoFrequency - Set the auto frequency:
  true - the auto frequency is enabled;
  false - the auto frequency is disabled.
- FLOAT Frequency - Set the central frequency of the selective filter (from 0.01 to half the sampling frequency of the measured channel), Hz.
- FLOAT FrequencyBar - Set the selective filter band (from 0.01 to half the sampling frequency of the measured channel), Hz.
- VARIANT_BOOL Activate - Set the operation of the selective voltmeter:
  true - the status of operation of the selective voltmeter is on;
  false - the status of operation of the selective voltmeter is off.

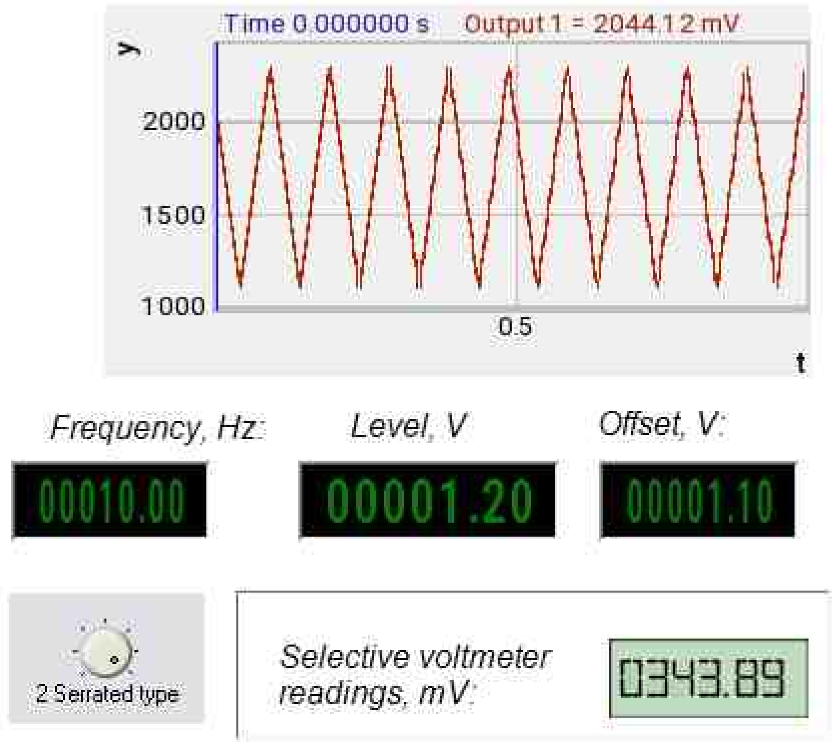
Example

Project in SCADA ZETView



In the design shown, the Selective voltmeter 605 component measures the RMS values of the Signal serrated 1124 coming from the Sine signal. Selectors 428 (also of a different type) are used to set the frequency, level, offset, and sawtooth waveform values. A Multi-channel oscilloscope 928 is needed for visual perception of the signal. The Liquid crystal display 655 is used for graphical representation of the RMS value of the sawtooth signal.

Project operation result



View the example in ZetView

Mathematical description

When describing variable signals, the following quantities are often used.

Average or the arithmetic mean is defined as follows:

$$\bar{v} = \frac{1}{T}\int_{t_0}^{t_0+T} v(\tau) = \frac{1}{T}\int_0^T v(\tau)\,d\tau$$

This value is equal to the area under the curve of the time function calculated over one period. Because the function is periodic, the arithmetic mean v is independent of the starting point t_0. For harmonic functions, the mean value of v is zero.

Modulo mean value defined as:

$$\bar{v} = \frac{1}{T}\int_0^T |v(\tau)|\,d\tau$$

This value is equal to the average value of the signal.

Note: The modulo average value is used when calculating the charge value of a capacitor, when working with rectified signals, or in calculations of electrolytic processes. The choice of rectifier type is also based on the modulo average current value, since the drop across the diode is almost always constant.

For a sinusoidal voltage, the following is true:

$$\bar{v} = \frac{1}{T}\int_0^T \hat{v}|\sin\omega t|dt = \frac{2}{T}\hat{v}\int_0^{T/2} \sin\omega t\, dt = \frac{1}{\pi}\hat{v}[-\cos\omega t]_{\omega t=0}^{\omega t=\pi} = \frac{2}{\pi}\hat{v} \approx 0.637\hat{v}$$

A similar relationship can be written for a sinusoidal current.

Root Mean Square (RMS) voltage characterizes the power in the alternating current circuit. By definition, the RMS function is equal to:

$$V_{СКЗ} = \sqrt{\frac{1}{T}\int_0^T v^2(t)dt}.$$

For a sinusoidal voltage, the following is true:

$$V_{СКЗ} = \sqrt{\frac{1}{T}\int_0^T (\hat{v}\sin\omega t)^2 dt} = \sqrt{\frac{1}{T}\hat{v}^2\int_0^T \sin^2\omega t\, dt} = \frac{\hat{v}}{\sqrt{2}} \approx 0{,}707\hat{v}$$

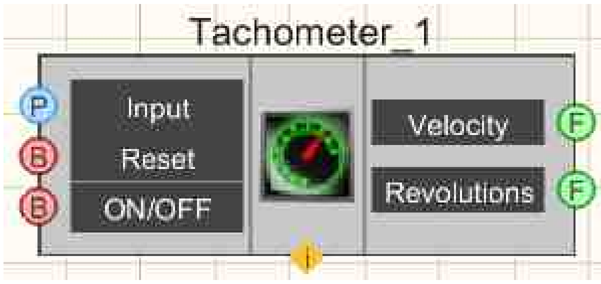
A similar relationship can be written for a sinusoidal current.

For variable signals, it is true that the RMS value is always less than or equal to the amplitude value.

11.9.Tachometer

The component "**Tachometer**" is designed to measure of measuring the frequency of rotation (the number of revolutions per unit of time) of machine parts and mechanisms.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - contact to which the measuring channel is connected.
- Reset - reset the accumulated number of revolutions to zero.
- ON/OFF -turn on and off tachometer.

➢ Output:

- Velocity - measured value of rotation velocity, rpm.
- Revolutions - the number of completed revolutions with since the last reset.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- DriveParameter1 (1) - kinematic parameter of drive shaft 1.
- DriveParameter2 (1) - kinematic parameter of drive shaft 2.
- DriveParameter3 (1) - kinematic parameter of drive shaft 3.
- DrivenParameter1 (1) - kinematic parameter of driven shaft 1.
- DrivenParameter2 (1) - kinematic parameter of driven shaft 2.
- DrivenParameter3 (1) - kinematic parameter of driven shaft 3.
- AutoLevel (false) - automatic threshold setting.
- Lowlevel (0) - lower threshold (in units).
- HighLevel (0) - upper threshold (in units).
- Activate (true) - tachometer operation status.
- RPS (0) - Set the unit of rotation frequency (0 - rpm, 1 - rpm).
- VirtChanRevs (0) - creating a virtual channel for the number of revolutions (1 - create, 0 - no)
- Amplify (1) - Set the multiplier (1, 10, 100, 1000, 10000, 100000).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - input channel for which you want to measure the required values (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG DriveParametr1 - Set the first numerator of the drive shaft kinematic parameter (from 1 to 199).
- LONG DriveParametr2 - Set the second numerator of the drive shaft kinematic parameter (from 1 to 199).
- LONG DriveParametr3 - Set the third numerator of the drive shaft kinematic parameter (from 1 to 199).
- LONG DrivenParameter1 - Set the first denominator of the kinematic parameter of the driven shaft (from 1 to 199).
- LONG DrivenParametr2 - Set the second denominator of the kinematic parameter of the driven shaft (from 1 to 199).
- LONG DrivenParameter3 - Set the third denominator of the kinematic parameter of the driven shaft (from 1 to 199).
- VARIANT_BOOL AutoLevel - Set the automatic setting of the threshold selection:

true - Automatic threshold setting is enabled;
false - Disabled automatic threshold setting (manual mode).

- FLOAT LowLevel - Set the lower threshold in channel units (number).
- FLOAT HighLevel - Set the upper threshold in channel units (number).
- VARIANT_BOOL Activate - Set the status of the tachometer:

true - The status of the tachometer is on;
false - The status of the tachometer is off.

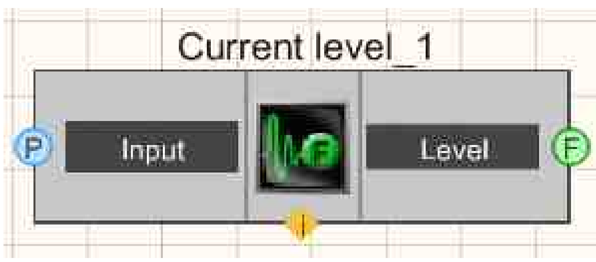
- LONG RPS - Set the unit of rotation speed: (0 - rpm, 1 - rps)
- LONG VirtChanRevs - Set the creation of a virtual channel of the number of revolutions (1 - create, 0 - no)

LONG Amplify - Multiplier setting (1, 10, 100, 1000, 10000, 100000).

11.10.Current level

The component "**Current level**" is designed for measuring the current signal level by channels.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - contact to which the measuring channel is connected.

➢ Output:

- Level - the measured value of the current level.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

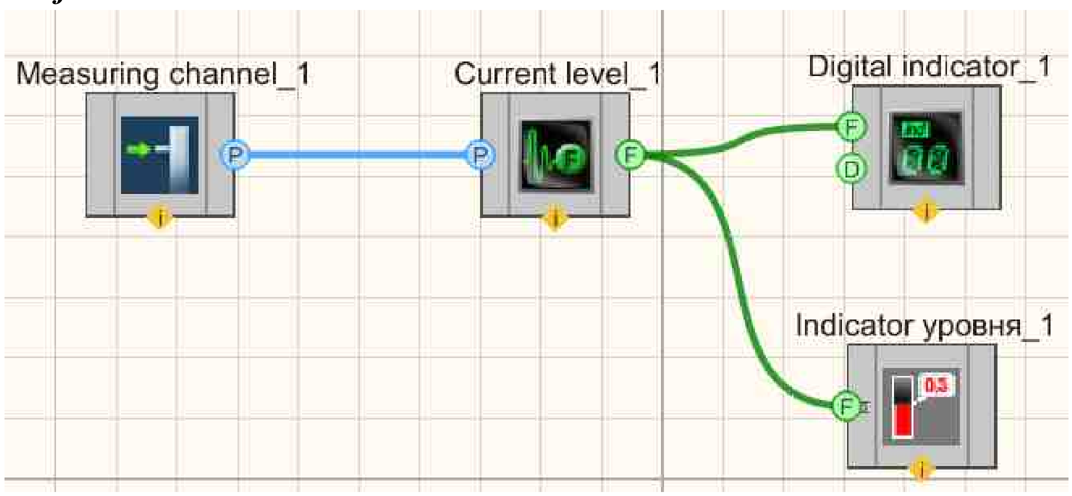
Setting:

➢**Input:**

- Input - Measuring channel (from 0 to (number of channels - 1)).

Example

Project in SCADA ZETView

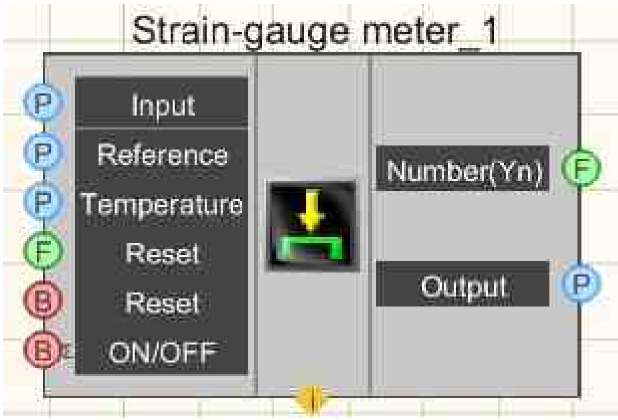


Project operation result

11.11.Strain-gauge meter

The component "**Strain-gauge meter**" is designed to work with data coming from the load cell channel. Used to obtain mechanical voltages in structures.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - measuring channel, which receives a signal for which you want to measure the required values.
- Reference - reference channel.
- Temperature - temperature compensation channel.
- Reset - value to which the current indicator is reset.
- Reset - when applied to this contact, the value is reset.
- ON/OFF -turn on and off Strain-gauge meter.

➢ Output:

- Number(Yn) - Strain-gauge meter output values.
- Output - virtual output channel from which the instantaneous values received from the Strain-gauge meter come.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Unit - indication of units of measure..
- measuretypemeasuretype (Absolute) - type of measurements:
  1. Absolute.
  2. Relative (the value of units relative to the reference channel).
- powertype (Direct current) - sensor power type:
  1. Direct current.

2. Alternating current - the calculation of values will be carried out according to the selective voltmeter algorithm, and, accordingly, the measuring circuit (primary converter) must be powered by alternating current.

- Smooth (0) - smoothing (time in [ms] for which the measured values will be averaged)
- Inversion (false) - is responsible for changing the sign of measurements (allows you to invert the load cell operation mode: compression-tension and vice versa).
- mode (Meter resistance) - the sensor operation mode:

1. Meter resistance - in this mode.

2. Strain-gauge - in this mode, you need to enter the value of Sensitivity (sensitivity, mV/V) and limit (measurement limit in units of Unit).

- Sensitivity (0) - Strain-gauge sensitivity, mV/V.
- Limit (0) - Strain-gauge measurement limit.
- UseClbFile (true) - permission to use the calibration file, which contains previously saved settings and calibration tables for the meter resistance.
- ClbFileName - specifies the path to calibration file for meter resistance.
- Value (0) - the number to which the current reading is reset.
- Activate (true) - the operation status (enabled/disabled).
- UseTremoChan (false) - use the temperature compensation channel.
- AmplifyCoeff (10) - the constant gain for the measuring channel: 1; 10; 100; 1000.
- Coefficient(1) - coefficient.
- UseVirtualChannel (false) - virtual channel usage (from 0 to 1).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - the channel to which the measuring channel is connected (from 0 to (number of channels - 1)).
- Reference - channel to which the power supply of the measuring bridge is connected (from 0 to (number of channels - 1)).
- Temperature - channel to which the temperature compensation channel is connected (from 0 to (number of channels - 1)).
- Reset - value to which the current reading is reset (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

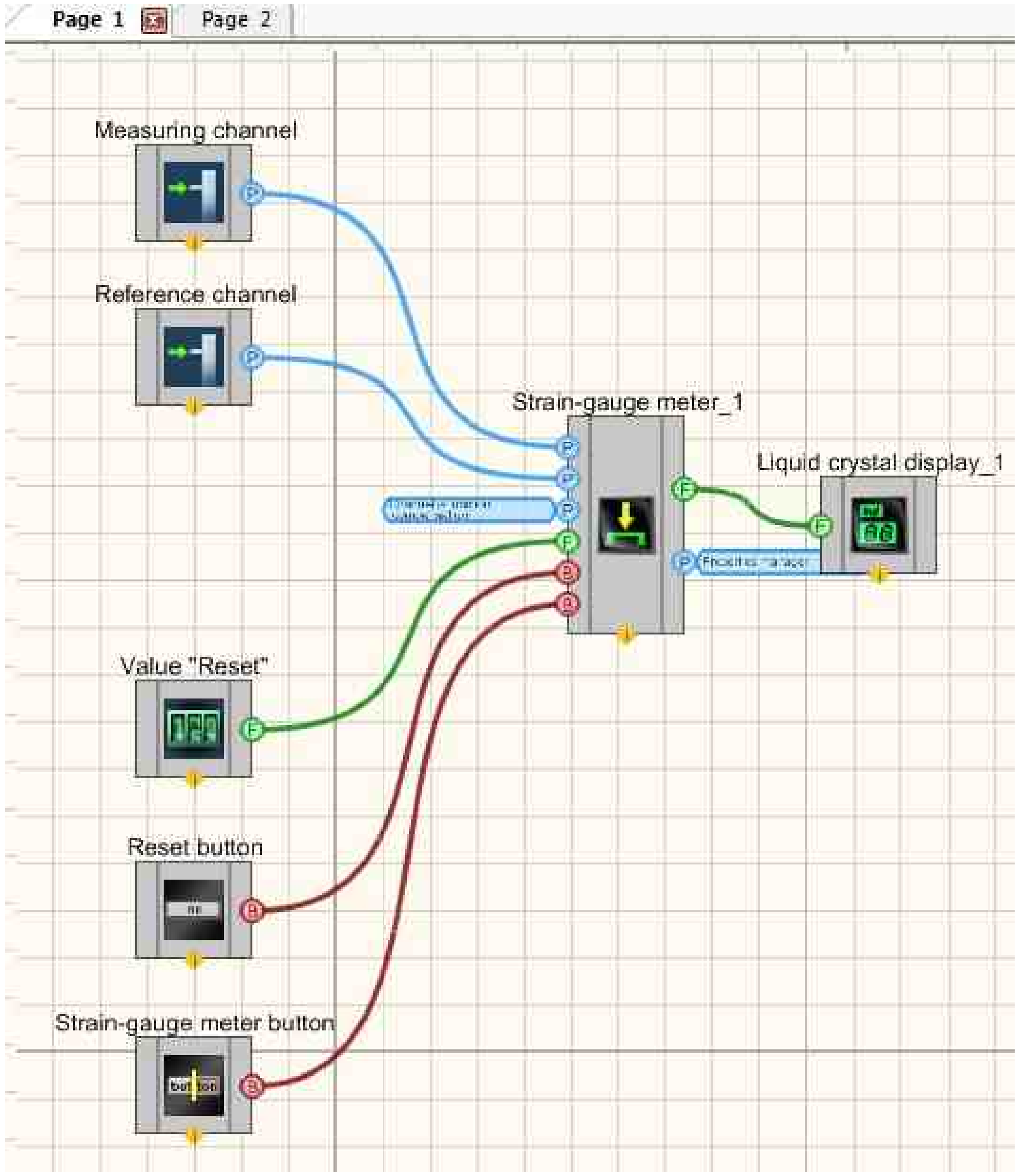
- BSTR Unit - Set the unit of measurement (string).
- BSTR MeasureType - Set the measurement type: Absolute; Relative (string).
- BSTR PowerType - Set the power: Direct current; Alternating current (string).
- FLOAT Smooth - Set the smoothing length, ms (number).
- VARIANT_BOOL Inversion - Set the inversion of the measured load value:
  true - Enabled inverting the measured load value.enabled;
  false - Disabled inverting the measured load value.
- BSTR Mode - Set the measurement mode: Meter resistance; Strain-gauge (string).
- FLOAT Sensitivity - Set the sensitivity of the Strain-gauge, mV / V (number).
- FLOAT Limit - Set the limit Strain-gauge measurement (number).
- VARIANT_BOOL UseClbFile - Set the permission to use the calibration file for meter resistance:
  true - Enabled permission to use the calibration file for meter resistance.
  false - Disabled permission to use the calibration file for meter resistance
- BSTR ClbFileName - Set meter resistance calibration file (string).
- FLOAT Value - Set the number to which the current indicator is reset (number).
- VARIANT_BOOL Activate - Set the operation status of the Strain-gauge meter:
  true - the Strain-gauge meter operation status is enabled;
  false – the Strain-gauge meter operation status is disabled.
- VARIANT_BOOL UseTremoChan - Set the temperature compensation channel:
  true - the status of the temperature compensation channel is enabled;
  false - the status of the temperature compensation channel is disabled.
- BSTR AmplifyCoeff - Set the constant gain for the measuring channel: 1; 10; 100; 1000 (string).
- FLOAT Coefficient - Set the coefficient.
- VARIANT_BOOL UseTremoChan - Set virtual channel usage:
  true - Virtual channel usage status is enabled;
  false - Virtual channel usage status is disabled.

➢**Methods:**

- LONG MultiTenzoData(LONG *data) - Set internal message method with ZETView.
- void ResetValue() – Set the Reset value (any value).

Example

Project in SCADA ZETView



This project shows the operation of the "Strain-gauge meter" component is represented as a device from ZETLab.

Project operation result

View the example in ZetView

Mathematical description

The operation of strain-resistive transducers is based on the property of materials to change their electrical resistance during mechanical deformations under the action of an applied force. Structurally, most meters resistances (Fig. 1) are produced in the form of conductors rigidly connected to a paper or film base 2. Conductor 3 is a so-called lattice of zigzag laid thin wire with a diameter of 0.02-0.05 mm, to the ends of which by soldering or welding output copper conductors are connected 4. From above, the conductors are covered with paper or film or varnished 1. After the Strain-gauge meter substrate is glued to the surface, the deformation of this surface is transmitted by the conductors and leads to a change in their resistance.

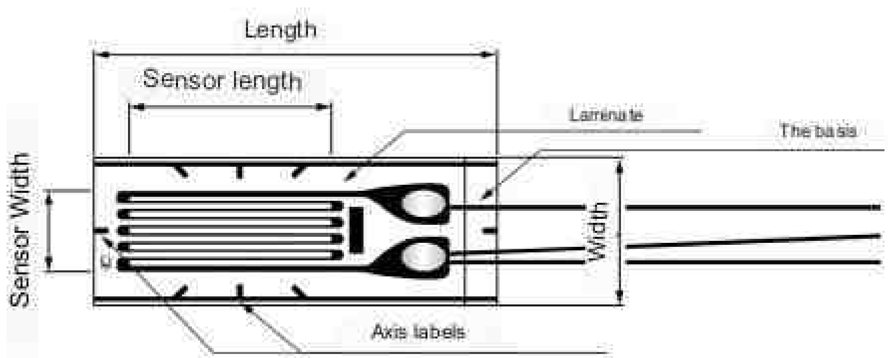


Fig. 1.

As can be seen in Fig, 1, the sensor consists of a base on which a conductive layer is applied, forming a snake, at the "turns" the thickness of the conductor is increased to reduce the sensitivity to tension perpendicular to the main axis. The main axis runs along the serpentine direction lines (horizontal in the figure) and stretching the sensor along this direction causes the maximum change in the resistance of the sensor. On top of the sensor is covered with a layer of transparent laminate, which protects the resistive layer from damage. The sensor also has marks indicating the direction of the axes, which simplifies its installation, usually pass through the center of the sensor at angles of 90 °, ± 45 ° relative to the main axis of sensitivity. When building a load control system for various mechanical structures, several Strain-gauges or sensor systems are usually used,

It is known that under the action of a force or a system of forces on a metal object, it begins to deform, and up to the material's yield point, the deformation is elastic in nature and obeys Hooke's law. This means that when the force is removed, the object takes on its original dimensions, and the magnitude of the stress is equal to the linear deformation multiplied by Young's modulus. The formulas for calculating stress and strain are given below:

$$\varepsilon = \frac{\Delta L}{L};$$

$$\sigma = \frac{F}{S};$$

$$\delta = E * \varepsilon;$$

where ε is linear deformation, E is Young's modulus, σ is stress, F is the force that caused the deformation, S is the cross-sectional area on which this force acts, L is the initial length, Δ is the change in length under the action of the force. There is also the concept of longitudinal and transverse deformation. Longitudinal deformation is the deformation of the body along the line of action of the force. Transverse - deformation of the body about an axis perpendicular to the direction of the force. Their ratio is called Poisson's ratio:

$$\nu = \left|\frac{\varepsilon_{\text{поп}}}{\varepsilon_{\text{продол}}}\right|.$$

For meters resistances sensors, the basic formula is as follows:

$$\frac{\Delta R}{R} = K\varepsilon,$$

where ΔR is the change in resistance of the Strain-gauge meter caused by deformation ε, K is the Strain-gauge coefficient of the gauge (table value), R is the initial resistance. If we record this expression in more detail, we get:

$$\Delta R = \frac{RKF}{SE}.$$

It follows from the formula that the change in the resistance of a Strain-gauge mounted on a metal beam, cross-section S, is directly proportional to the cross-sectional area of the beam and Young's modulus. By measuring the change in the resistance of the Strain-gauge, it is possible to calculate the load on the considered beam, knowing its cross section and the property of the material from which it is made. This formula is only valid for tension/compression cases.

For a bending situation, the whole process looks somewhat different, since the deformation is more complicated to calculate. Consideration must be given to the cross-sectional shape of the beam and the distance from where the force is applied to the center of the Strain-gauge. The deformation in this case can be calculated as follows:

$$\varepsilon = \frac{M}{ZE},$$

where M is the moment of force, Z is the moment of section resistance. For different Strain-gauge placements, beam attachments, and the location of the point of application of the force, it is calculated in different ways.

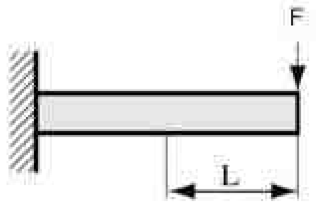
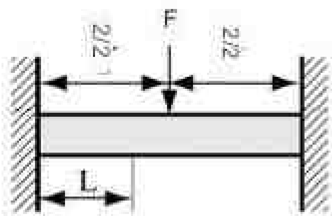
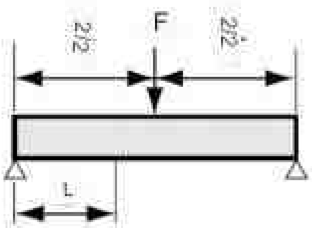
Force application pattern	Calculation Formula
	$M=FL$
	$L=\frac{l}{2};\Rightarrow M=\frac{Wl}{8}$
	$L=\frac{l}{2};\Rightarrow M=\frac{Wl}{4}$

Table 1. Calculation of the moment of force for various options for the position of the beam and Strain-gauge.

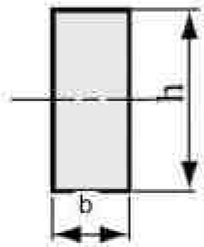
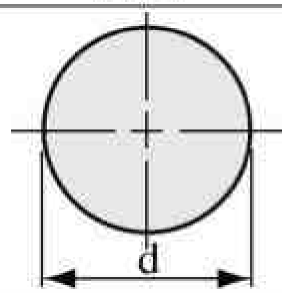
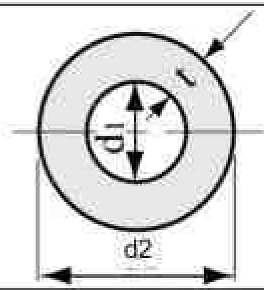
Cross section	Calculation formula
	$\frac{1}{6} bh^2$
	$\frac{\pi}{32} d^3$
	$\frac{\pi (d_2^4 - d_1^4)}{32 d_2}$

Table 2. Calculation of section modulus for various section configurations.

To monitor the torque, Strain-gauges are used, located on the beam at an angle of 45 ° relative to the axis of rotation (Fig. 2). One, two or four sensors are used. When installing two sensors, their axes are located at an angle of 90° relative to each other and at an angle of 45° and -45° relative to the axis of rotation. In this case, one of the sensors experiences compression deformation, the other - tension. When using four meters resistances, they are arranged in a "cross".

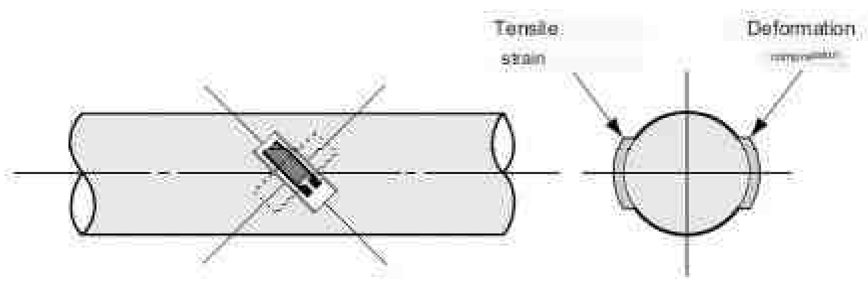


Fig. 2

The rotating force can be calculated as follows:

$$T = \frac{\varepsilon * E * Z_p}{1+\nu},$$

where ν - Poisson's ratio, T is torque, E is Young's modulus, Zp is the angular resistance of the section. For a solid cylindrical rod:

$$Z_p = \frac{\pi d^3}{16}.$$

For a cylindrical hollow pipe with outer diameter d_2 and inner diameter d_1:

$$Z_p = \frac{\pi(d_2^4 - d_1^4)}{32 d_2}.$$

Since ΔR is a very small value compared to R, it is measured by including the sensor in the Wheatstone bridge circuit.

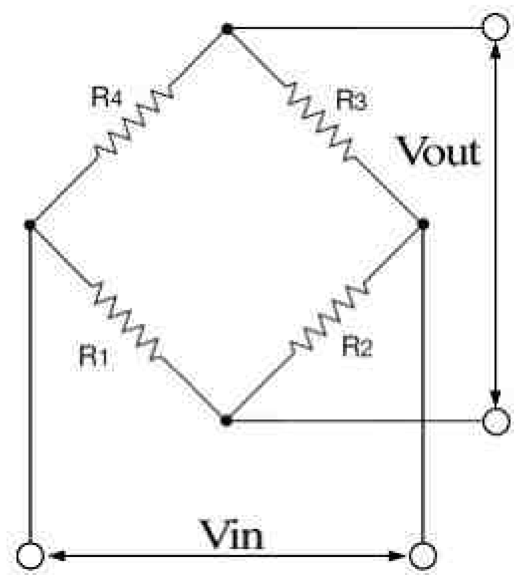


Fig. 3.

The measurement principle is based on the mutual compensation of the resistances of two links, one of which includes the measured resistance. As an indicator, a sensitive galvanometer is usually used, the readings of which should be equal to zero at the moment of equilibrium of the bridge. In the diagram, R_1, R_2, R_3, R_4 are the shoulders of the bridge. R_3 is an unknown resistance; R_1, R_2 and R_4 are known resistances, and the value of R_2 can be adjusted. If the ratio of resistances (R_2/R_1) is equal to the ratio of the resistances of the other (R_3/R_4), then the potential difference between the two midpoints will be zero, and no current will flow between them. The resistance R_2 is adjusted until equilibrium is obtained, and the direction of current flow shows in which direction R_2 should be adjusted.

With a galvanometer, the equilibrium moment can be set with great accuracy, and if the resistances R_1, R_2 and R_4 have a small error, then R_3 can be measured very accurately, because even small changes in Rx cause a noticeable imbalance of the entire bridge. When the bridge is balanced:

$$R_3 = \frac{R_2 R_4}{R_1}.$$

If the values of all four resistances are known, as well as the voltage (V_{in}), then the voltage across the bridge arms can be found using the voltage divider formulas, and then subtracted from each other to find V_{out}:

$$V_{out} = \left(\frac{R_3}{R_4+R_3} - \frac{R_2}{R_1+R_2}\right)V_{in} = \frac{R_3R_1 - R_4R_2}{(R_4+R_3)(R_1+R_2)}V_{in}.$$

If our meter resistance (R_3) will have a resistance of R + ΔR, and the remaining resistors are equal to R, then we have:

$$V_{out} = \frac{(R+\Delta R)R - RR}{(R+R+\Delta R)(R+R)}V_{in} = \frac{\Delta R R}{2R(2R+\Delta R)}V_{in} = \frac{\Delta R}{4R+2\Delta R}V_{in},$$

since ΔR is a very small value compared to R, the output voltage will be:

$$V_{out} = \frac{1}{4}\frac{\Delta R}{R}V_{in} = \frac{1}{4}K\varepsilon E.$$

When solving problems of measuring the load on certain structures, one, two or meters resistances are usually included in the measuring bridge.

The first and most important factor is the temperature, or rather its influence not only on the resistance of the Strain-gauge itself. But also on the linear dimensions of the controlled object. There are 2 ways to solve this problem:

- Use of temperature-compensated Strain-gauges.

- Inclusion of an additional Strain-gauge in the measuring bridge.

Let's consider the second way. To compensate for the temperature effect on the balance of the measuring bridge itself, an additional Strain-gauge is often used, which is included in the bridge in such a way that the change in its resistance due to temperature drift is consistent with the change in the resistance of the main Strain-gauge. In this case, an additional sensor is mounted on a beam, to which no force is applied, but at the same temperature as the controlled structure. in Example, to compensate for the Strain-gauge included as R_1, an additional Strain-gauge is put in place of R_4. In this case, the change in resistance due to the change in temperature on the main Strain-gauge is equal to the change in resistance in the additional Strain-gauge, and the balance of the measuring bridge is maintained.

When building a measuring bridge, in practice it usually turns out that the sensors themselves are at a fairly large distance from each other. In this case, the length of the wire can be several hundred meters. In this case, it will also be necessary to take into account the temperature effect in the wires.

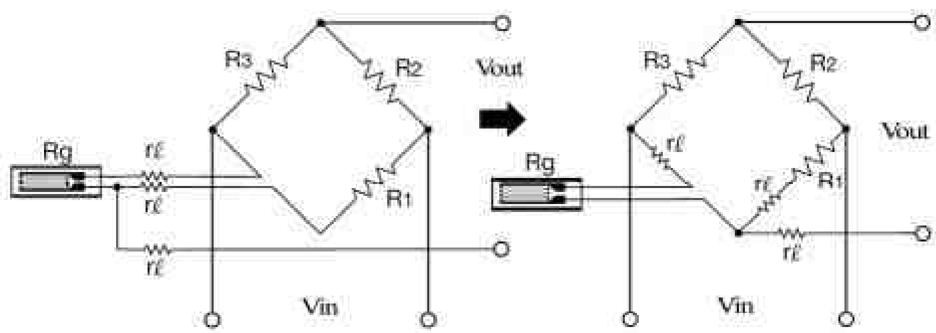


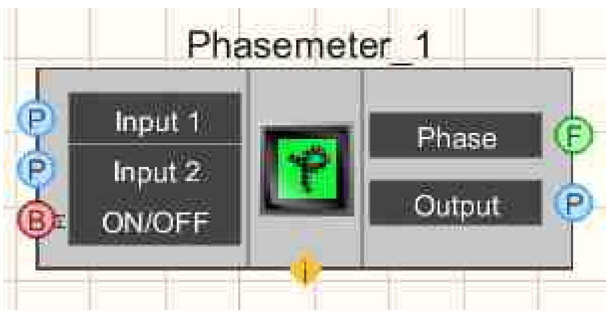
Fig. 4.

The diagram shows the compensation scheme. As can be seen from the figure, the resistance of two wires enter the measuring bridge and compensate each other, the resistance of the third is connected in series with the output voltage. When using a meter with an input resistance that is much larger, the resistance of the resistors can be ignored..

11.12.Phasemeter

The component "**Phasemeter**" is designed to determine the phase difference between two periodic signals.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - input channel 1, which receives signal 1 (base channel, relative to which the phase shift is measured).
- Input 2 - input channel 2, which receives signal 2.
- ON/OFF -turn on and off of the phasemeter.

➢ Output:
- Phase - measured values of the phase difference between two channels (phase is displayed, relative to signal 2).
- Output - virtual output channel of the measured values of the phase difference of two incoming channels.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- averagetime (1 s) - the time for which the average value is calculated, s:
  1. 0.1 seconds.
  2. 1 second.
  3. 10 seconds.
- meashuretype (Degrees) - measurement type (Degrees, Radians):

- Activate (true) - the operation status (enabled/disabled).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

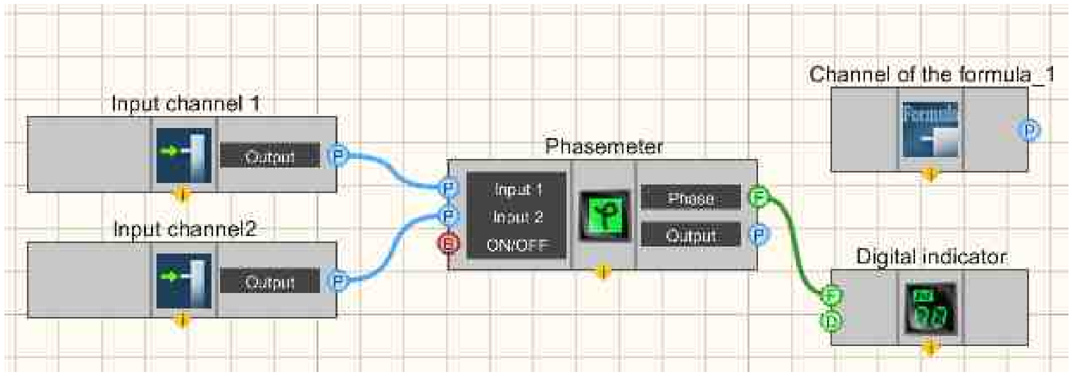
➢**Input:**
- Input 1 - measuring channel No. 1, which receives the signal No. 1 (base channel against which the phase shift is measured) (from 0 to (number of channels - 1)).
- Input 2 - measuring channel No. 2, which receives the signal No. 2 (from 0 to (number of channels - 1)).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- BSTR AverageTime - Set the average: 0.1 s; 1 s; 10 s (string).
- BSTR MeasureType - Set the type of measurement: Degrees; Radians (string).
- VARIANT_BOOL Activate - Set the status of the phasemeter:
  true - the status of the phasemeter is enabled;
  false - the status of the phasemeter is disabled.

Example

Project in SCADA ZETView



This project shows the operation of the component Phasemeter 626 measures the phase difference between the channels Input channel 1 375 and Input channel 2 375. ZETFormula 626 is used to generate two sine signals, and one of the sinusoids has a phase shift relative to the other. The Digital Indicator 682 component is used for graphical display of the phase difference value.

Project operation result

View the example in ZetView

Mathematical description

A phasemeter is an electrical measuring device designed to measure the phase angles between two periodically changing electrical oscillations, for example, in a three-phase power supply system

The active power P is understood as the average value of the instantaneous power p for the period T:

$$p = \frac{1}{T}\int_0^T p\,dt = \frac{1}{T}\int_0^T ui\,dt$$
.

If current:

$$i = I_m \sin \omega t$$
.

And the voltage in the circuit

$u = U_m \sin(\omega t + \varphi)$. That

$$p = \frac{1}{T}\int_0^T I_m U_m \sin \omega t \sin(\omega t + \varphi)\,dt = \frac{U_m I_m}{2}\cos\varphi$$

The expression for the effective power in a sinusoidal current circuit has the form:

$$\mathrm{P} = \mathrm{V} * \mathrm{I} * \cos\varphi$$
,

where V and I are the rms voltage and current. The multiplier cos φ included in the expression is called the power factor. The unit of effective power is [W].

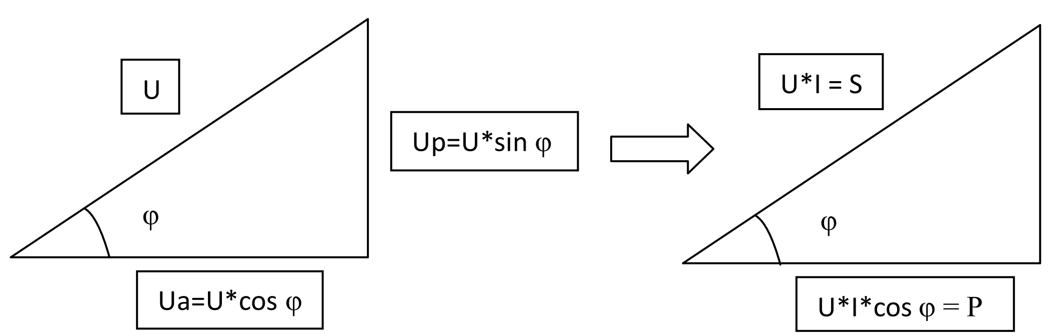
- In purely resistive circuits (φ = 0), the power factor is 1, and the effective power is given by P=V*I.
- In purely reactive circuits (φ = ±90°), the power factor, and hence the effective power, is 0.
- In resistive-capacitive and resistive-inductive circuits (-90° < φ < 90°), the effective power is always positive.

The expression for reactive power in a sinusoidal current circuit is:

$$Q = U * I * \sin\varphi$$

From the voltage triangle:

:



The expression for the total power in a sinusoidal current circuit is:

$$S = U * I$$

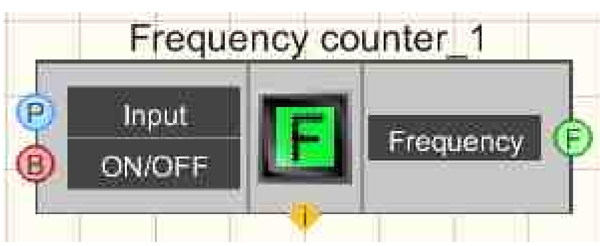
Effective electrical power can be converted into other forms of power (thermal, mechanical, etc.). And reactive power, in turn, cannot be converted into any other types of energy.

The phasemeter also allows you to determine the sign of the angle, on the basis of which it is possible to draw a conclusion about the type of load (resistive-capacitive, resistive-inductive, etc.).

11.13.Frequency counter

The component "**Frequency counter**" is designed to determine the frequency of a periodic process or the frequencies of the harmonic components of the signal spectrum.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - measuring channel for which the required values are to be measured.
- ON/OFF -turn on and off the voltmeter.

➢ Output:
- Frequency - measured values of signal frequency, Hz

Custom common properties:
➢Total properties (environment):

- [Are available by the link below.](154)

➢**Custom private properties (the default value is shown in parentheses):**

- averagetime (1 s) - the time for which the average value is calculated, s:
  1. 0.1 seconds.
  2. 1 second.
  3. 10 seconds.
- Activate (true) - the operation status (enabled/disabled).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - the value of the signal for which you want to measure the required values (from 0 to (number of channels - 1)).

➢**Custom common properties:**

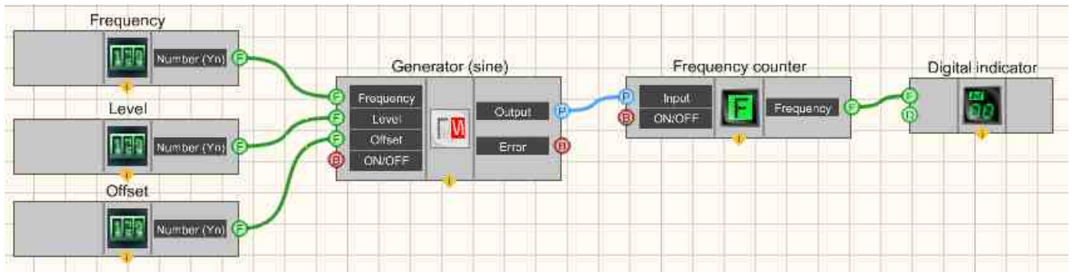
- [Are available by the link below.](154)

➢**Custom Custom private properties (the default value is shown in parentheses):**

- BSTR AverageTime - Set the average: 0.1 s; 1 s; 10 s (string).
- VARIANT_BOOL Activate - Set the operation status of the frequency counter:
  true - The status of the frequency counter is enabled;
  false - The status of the frequency counter is disabled.

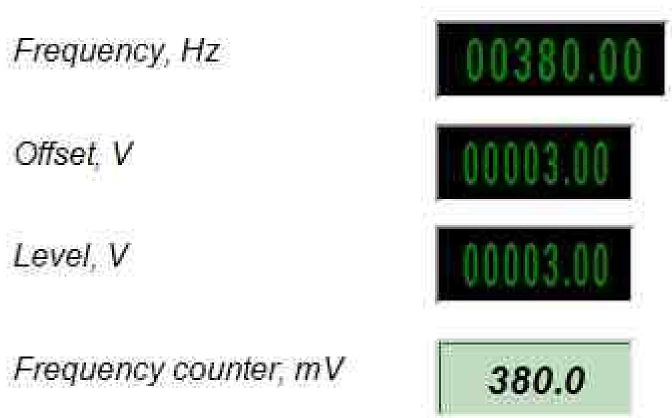
Example

Project in SCADA ZETView



This project shows the operation of the [Frequency counter 630] component measures the frequency of a Sine signal coming from a [Sine signal 630]. The [Selectors 428] are used to set the values of frequency, level, offset of the sine signal. The [Digital Indicator 682] is used for graphical representation of the signal frequency value.

Project operation result

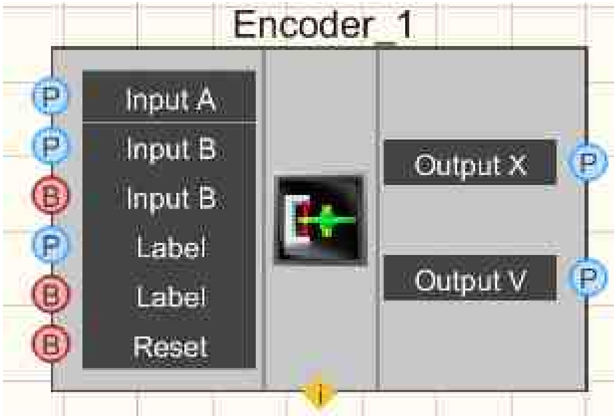


View the example in ZetView

11.14.Encoder

The component "**Encoder**" is designed to measure the relative position (displacement), velocity and direction of movement using optical sensors of angular or linear displacements (encoders) connected to the input channels of the ADC. The Encoder program generates virtual channels of movement and movement velocity.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input A - phase A channel.
- Input B - phase B channel.
- Input B - turn on the phase B channel.
- Label - label channel.
- Label - turn the label channel on and off.
- Reset - Reset all values to zero.
-

➢ Output:
- Output X - output channel of movement.
- Output V - velocity output channel.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- channelA_name - the name of the phase A measurement channel.
- channelB_name- the name of phase B measurement channel.
- channelLabel_name - the name of the label measurement channel.
- channelB_Enabled (true) - enable/disable phase B channel.
- channelLabel_Enabled (true) - enable/disable the label channel.
- channelPath_Enabled (true) - enable/disable the virtual displacement channel.
- channelVelocity_Enabled (true) - enable/disable virtual velocity channel.
- units (mm) - selection of the unit of measurement:
  mm - millimeter;

cm - centimeter;
m - meter;
degree - degree;
revolution - revolution.

- resolution (250) - resolution of the encoder (specified in labels per unit).
- Level_autoSet (true) - enable/disable automatic threshold setting.
- Level_low (-2000.000000) - lower threshold (in channel units).
- Level_hight (2000.000000) - upper threshold (in channel units).
- Invertion (false) - inversion of the direction of movement (necessary condition - enabled phase B)

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input A - the channel to which phase (channel) A is connected (from 0 to (number of channels - 1)).
- Input B – the channel to which phase (channel) B is connected (from 0 to (number of channels - 1)).
- Label - the channel to which the label channel is connected (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•BSTR channelA_name - Set the name of the phase A measurement channel (string).
•BSTR channelB_name - Set the name of the phase B measurement channel (string).
•BSTR channelLabel_name - Set the label measurement channel name (string).
VARIANT_BOOL channelB_enable - Set to enable phase B channel:

- true - the phase B channel is enabled;
  false - the phase B channel is disabled.

VARIANT_BOOL channelLabel_enable - Set to enable the label channel:

- true - the label channel is enabled;

false - the label channel is disabled.

VARIANT_BOOL channelPath_enable - Set to enable the virtual displacement channel:

- true - the virtual displacement channel is enabled;
  false - the virtual displacement channel is disabled.

VARIANT_BOOL channelSpeed_enable - Set virtual channel velocity enable:

- true - the virtual velocity channel is enabled;
  false - the virtual velocity channel is disabled.

- BSTR units - Set the choice of unit of measure: mm; cm; m; degree; resolution (string).

- LONG resolution - Set the resolution, which is given in labels per unit of measure.

VARIANT_BOOL Level_autoSet - Set the automatic threshold setting on/off:

- true - Automatic threshold setting is enabled;
  false - Automatic threshold setting is disabled (manual mode).

- FLOAT Level_low - Set the lower threshold in channel units (number).
- FLOAT Level_hight - Set the upper threshold in channel units (number).

VARIANT_BOOL Invertion - Set the direction of displacement to be inverted (works when phase B is on):

- true - Invert direction of displacement is enabled;
  false - Invert direction of displacement is disabled.

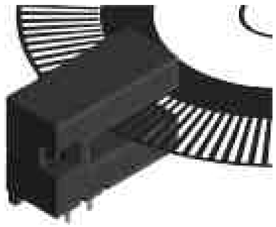
Learn more about using the Encoder:

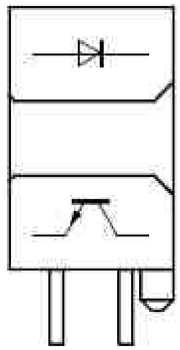
The "Encoder" program is designed to measure the relative position (displacement), velocity and direction of movement using optical displacement sensors (encoders) connected to the input channels of ADC modules and spectrum analyzers.

On the basis of optical sensors, sensors of linear and angular displacements are created. The accuracies of such sensors can be from 1 µm to 1 mm with a measuring base length of 8 mm to 3 m. resolution can be up to 5 minutes.

Optical technology has offered a number of classic ways to build an encoder - a sensor that represents motion, position or direction information either directly in digital form, or generating a impulse train from which, after digitization, a digital code can be generated.

The principle of operation





The principle of operation of encoders is illustrated in Figure 1. An optical encoder consists of a thin optical disk and a stationary block - a measuring head, which includes a light source and a photodetector. The optical disc includes a surface of transparent and opaque areas. Markers can be, for example, holes in a metal sheet or marks on a glass disc. When the disc rotates, depending on its type, the markers pass or block the beam of light directed from the light source to the photodetector.

The photodetector generates a signal with a frequency equal to the code element repetition rate in digital form or an analog impulse signal, which can also be amplified and digitized. By adding a second LED-phototransistor pair with an angular displacement relative to the first, corresponding to a quarter of the signal period, a second sequence of pulses can be obtained - channel B with a phase shift relative to channel A by 90°. The incremental encoder, which uses three optical encoders, can simultaneously double the resolution of position and velocity measurements and detect direction.

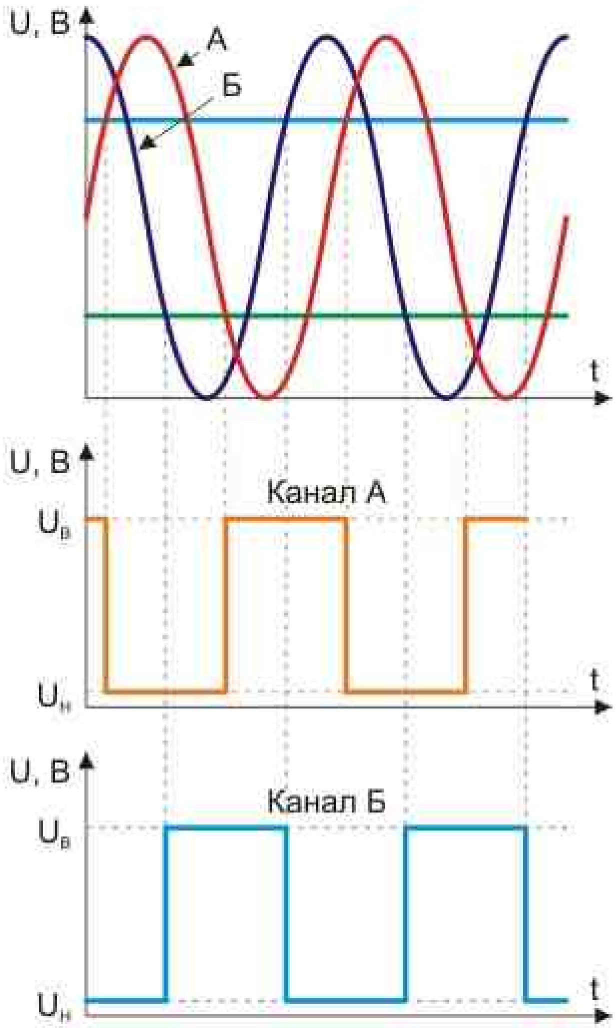


Fig. 1

Linear and angular displacement sensors are connected directly to the ADC modules. The generator output can be used to power the sensors. The resolution of incremental encoders is measured in pulses per revolution (ppr). In the "Encoder" program, the user is given the opportunity to select the resolution of the encoder used (the "Resolution, labels / u" window). "E.i." - unit of measurement, which can be selected from the range "mm, cm, m, gr (degrees), vol (turns)" or entered manually in the "Unit of measurement" window.

Also, in the "Phase A" and "Phase B" drop-down lists, the encoder connection channels are selected, to which the corresponding signals "Channel A" and "Channel B" are connected. The drop-down list "Label 0" is intended for selecting the channel of the ADC module or spectrum analyzer, to which the synchronization signal is connected. Thresholds for triggering the synchronization signal are set manually or automatically.

When the "Displacement" and "Velocity" checkboxes are checked in the "Encoder" program window, additional virtual channels are created in the data server containing, respectively, information about displacement and velocity.

Figure 2 shows the waveforms from the channels "Phase A", "Phase B" and "Mark 0", obtained using the "Multichannel oscilloscope" program. Figure 3 shows the waveforms from the virtual channels of movement and velocity and the signal "Label 0".

Fig. 2

Fig. 3

When using the "Encoder" program in conjunction with programs from the ZETLab and angular displacement sensors, the user can analyze torsional vibrations and use this equipment to replace torsiographs.

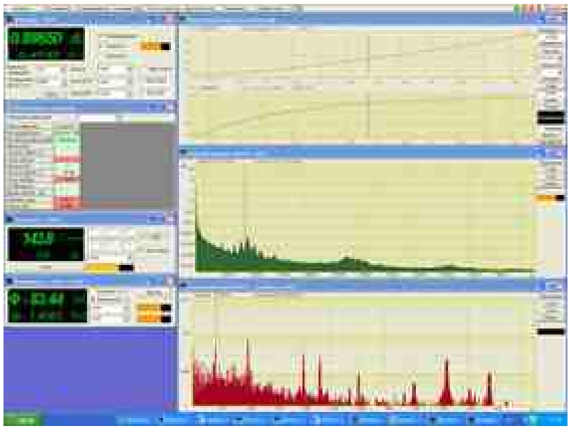


Fig. 4

The control and automation module built into the program from the ZETLab Studio provides simplicity and convenience when building your own software and measuring systems..

Purpose of the program

The Encoder program is designed to measure the relative position (displacement), velocity and direction of movement using optical sensors of angular or linear displacements (encoders) connected to the input channels of the ADC. The Encoder program generates virtual channels of movement and movement velocity. These channels are available for further analysis by other ZETLab programs.
On the basis of optical sensors, sensors of linear and angular displacements are created. The accuracy of such sensors can be from 1 µm to 1 mm with a measuring base length of 8 mm to 3 m. resolution can be from a few degrees to 5 minutes.
Optical technology has offered a number of classical ways to build a sensor encoder representing motion, position or direction information either directly in digital form (absolute encoders) or generating a impulse train (incremental encoders).

Further, in the description of the Encoder program, we will only talk about incremental encoders, since the Encoder program is designed to work only with this type of these sensors.

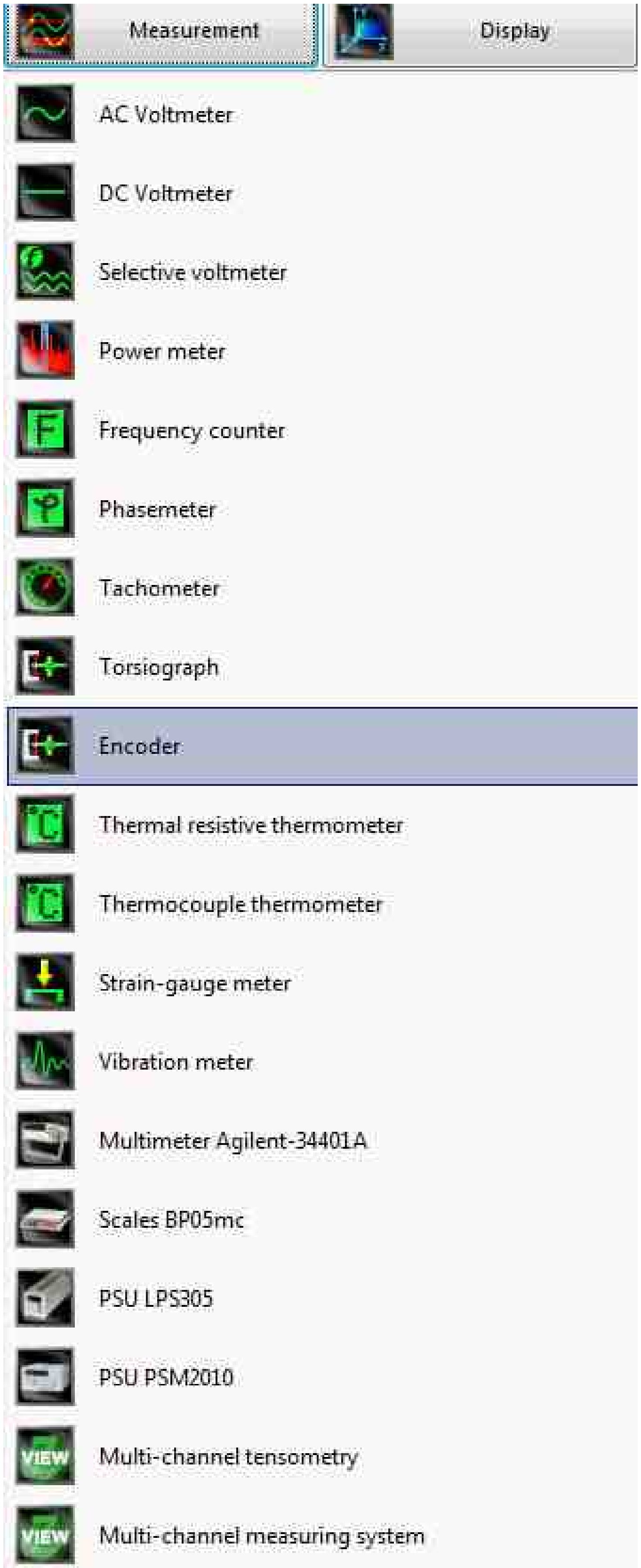
An optical encoder consists of a thin optical disk and a stationary block of the measuring head, which includes a light source and a photodetector. The optical disc contains a surface of transparent and opaque areas. Markers can be, for example, holes in a metal sheet or marks on a glass disc. When the disc rotates, depending on its type, the markers pass or block the beam of light directed from the light source to the photodetector.

The photodetector generates a signal with a frequency equal to the frequency of the code elements in digital form or an analog impulse signal, which can also be amplified and digitized. When adding a second pair of LED phototransistors with an angular displacement relative to the first, corresponding to a quarter of the signal period, a second sequence of pulses can be obtained - channel B with a phase shift relative to channel A by 90°. The incremental encoder, which uses two optical channels, allows you to simultaneously double the resolution of position and velocity measurements and determine the direction. The third channel is used to bind to the start mark (label "0").

Program description

To start the Encoder program, select the Encoder command from the Measurement menu of the ZETLab panel. The working window of the Encoder program will be displayed on the monitor screen. The title of the program window will display the name of the program and the name of the channel to which channel A (phase A) of the optical displacement sensor (encoder) is connected.

Note: The Encoder program can be launched directly from the ZETLab working directory (by default: c:\ZETLab\). Executable file name: Encoder.exe


Measurement
Display
AC Voltmeter
DC Voltmeter
Selective voltmeter
Power meter
Frequency counter
Phasemeter
Tachometer
Torsiograph
Encoder
Thermal resistive thermometer
Thermocouple thermometer
Strain-gauge meter
Vibration meter
Multimeter Agilent-34401A
Scales BP05mc
PSU LPS305
PSU PSM2010
Multi-channel tensometry
Multi-channel measuring system
Fig. 5

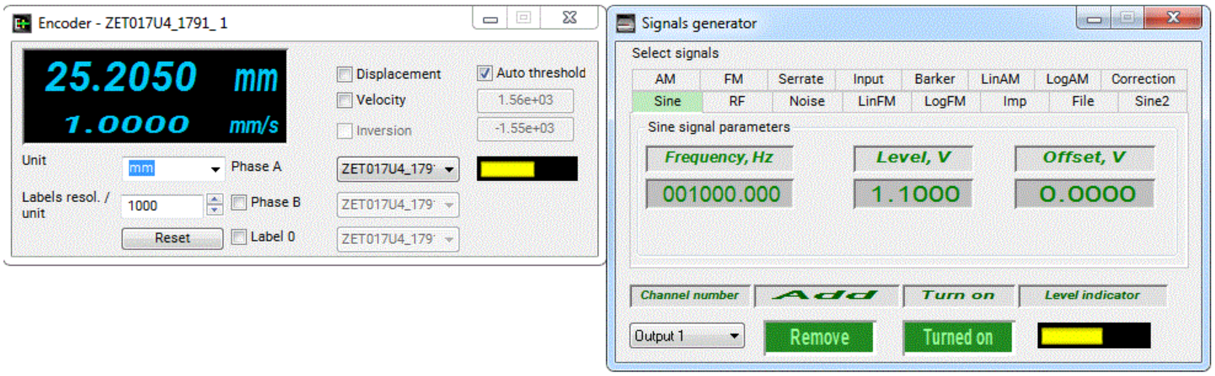


Fig. 6

In the list located to the right of the inscription Unit of measurement, select or enter from the keyboard the unit of measurement in which the signal on the selected channel will be displayed. To select the required unit of measure, left-click on the list button, and, in the list that opens, select the desired unit of measure. If the required unit of measure is not in the list, then by clicking the right mouse button on the list field, enter the required unit of measure from the keyboard.

The resolution of incremental encoders is determined by the number of pulses per revolution (ppr). In the list located to the right of the inscription Label resolution / u, the required number of labels in the set unit of measurement is selected. For example, an incremental rotary encoder has 1080 marks per revolution, corresponding to 3 marks per degree of rotation. It is necessary to measure the position of the encoder in degrees with a measurement accuracy of one degree. To do this, in the Unit of measurement list, select the unit of measurement - degrees (g), and in the Label resolution / u. set to 3 (three marks per degree of encoder rotation). The required resolution is set by pressing the left mouse button on the permission list buttons, or by clicking the left mouse button on the permission list field, scrolling the mouse roller, or,
Below the list box label resolution/e.i. the Reset button is located, pressing which leads to zeroing the value of the displacement (position).

The flags Movement and Velocity, located to the right of the graphical indicator, enable/disable the virtual channels Movement and Velocity generated by the Encoder program. These channels are available for further analysis by other programs. Checked boxes – virtual channels are enabled, unchecked – are disabled. The data in these virtual channels goes at the rate of processing each impulse without averaging. This allows you to explore not only the movement and velocity of movement, but also their non-uniformity. When the encoder is turned on together with other sensors, such as pressure or temperature, the data from these sensors and the data of movement and movement velocity are synchronized with an accuracy of one encoder pulse.

The Invert check box, which becomes available for checking or unchecking when the Phase B check box is checked, allows you to invert the direction of movement signal. Checked box - inversion occurs, unchecked - the signal is not inverted.

To measure movement and movement velocity on the physical channel to which channel A of the encoder is connected, it is necessary to select the name of this channel in the list field (with an arrow) located to the right of the Phase A inscription.

When using the encoder channel B in measurements, it is necessary to check the box located to the right of the Phase B inscription, and, in the list field that has become available, select the name of the enabled physical channel to which the encoder channel B is connected. Unchecking the Phase B checkbox disables the Channel B selection list box and the Invert setting checkbox.

If it is necessary to calculate the absolute position, you must set the checkbox located to the right of the Label 0 inscription, and, in the list field that has become available, select the name of the enabled physical channel to which the encoder zero mark channel is connected. Each time, when passing the zero mark of a pair of LED-phototransistor, zeroing occurs on the graphic indicator of the measured displacement readings. For example, this is useful when measuring linear displacements in which reciprocating motion occurs. By Set the zero mark in the middle, you can measure the movement in one direction or another relative to the set zero mark.

The Auto threshold checkbox is used to enable/disable automatic/manual setting of the upper and lower thresholds of the input level, which will be used to measure movement. Checked box – the program itself automatically sets the upper and lower thresholds for the signal level. Unchecked – manual input of the upper and lower signal level thresholds is allowed.

The input fields located under the Auto threshold checkbox are used to set the upper and lower thresholds in manual mode (the Auto threshold checkbox is unchecked). The upper input field is used to set the upper threshold, the lower one - to set the lower threshold. The upper and lower threshold values are entered from the keyboard. After entering the values, press the keyboard key. If the Auto threshold checkbox is checked, the upper and lower threshold input fields are not available for entering values. The upper and lower thresholds are set to eliminate false positives when measuring displacement. For correct measurement of movement, the upper threshold must not exceed the maximum signal level on this channel, the lower threshold must not be lower than the minimum level. You can determine the maximum and minimum signal level for the channel to which the encoder is connected, for example, by launching the Multi-channel oscilloscope program, select this channel and evaluate these levels from the oscillogram. To exit the program, click the button located in the upper right corner of the window.

When using the Encoder program in conjunction with ZETLab programs and angular displacement sensors, the user can analyze torsional vibrations and use this equipment to replace torsiographs.

Connecting sensors

Linear or angular displacement sensors are connected directly to the input channels (ADC) of devices manufactured by Electronic Technologies and Metrological Systems LLC.

After connecting the encoder to the input channels, you need to enable these channels in the ADC and DAC settings program or make sure that they are enabled.

To measure movement (position) and movement velocity, it is necessary to configure the parameters of the channels to which the encoder is connected in the Editing Parameter Files program. The parameters of the measuring channels must be set in relation to the voltage measurement.

Any channels for measurement are selected, the user enters the name of the channels at his discretion.
To power the sensors, you can use both the output of the built-in generator (if equipped with a built-in generator) in the mode of generating a Sine signal with a constant zero offset, or an external power supply.

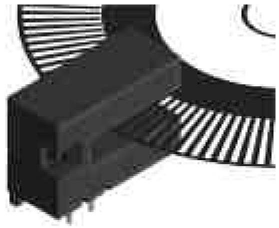
Mathematical description

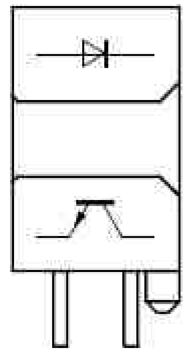
Encoder- a device designed to convert the angle of rotation of a rotating object (shaft) into electrical signals, allowing to determine the angle of its rotation. Encoders are divided into incremental and absolute. An incremental encoder generates a certain number of pulses per revolution. And absolute encoders allow you to know the current angle of rotation of the axis at any time, including after power failure and restoration. And multi-turn absolute encoders, in addition, also count and store the number of complete revolutions of the axis. Encoders can be either optical, resistor or magnetic and can operate via bus interfaces or industrial network.

Incremental encoders are designed to determine the angle of rotation of rotating objects. They generate a serial impulse digital code containing information regarding the rotation angle of the object. If the shaft stops, then the transmission of impulses also stops. The main operating parameter of the sensor is the number of pulses per revolution. The instantaneous value of the angle of rotation of the object is determined by counting the impulses from the start. To calculate the angular velocity of an object, the processor in the tachometer differentiates the number of pulses over time, thus immediately showing the velocity value, that is, the number of revolutions per minute. The output signal has two channels in which identical impulse sequences are shifted by 90° relative to each other (paraphase pulses), which makes it possible to determine the direction of rotation.

The principle of operation of incremental encoders is illustrated in Figure 1. An optical encoder consists of a thin optical disk and a stationary block - a measuring head, which includes a light source and a photodetector. The optical disc contains a surface of transparent and opaque areas. Markers can be, for example, holes in a metal sheet or marks on a glass disc. When the disc rotates, depending on its type, the markers pass or block the beam of light directed from the light source to the photodetector.

.





The photodetector generates a digital signal with a frequency equal to the code element repetition rate or an analog impulse signal, which can also be amplified and digitized. By adding a second pair of LED-phototransistor with an angular displacement relative to the first, corresponding to a quarter of the signal period, a second sequence of pulses can be obtained - channel B with a phase shift relative to channel A by 90°. The incremental encoder, which uses two optical channels, allows you to simultaneously double the resolution of position and velocity measurements and determine the direction. The third channel is used to bind to the start mark (label "0").

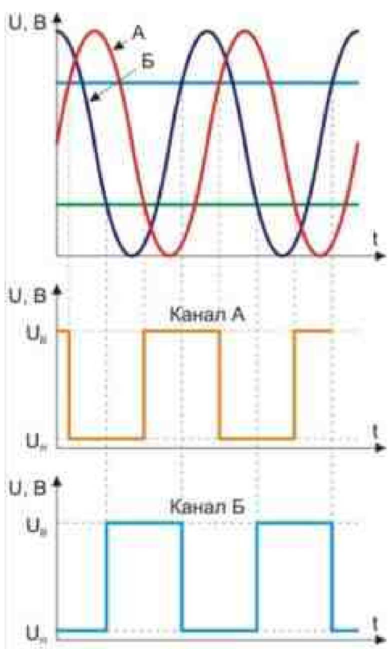
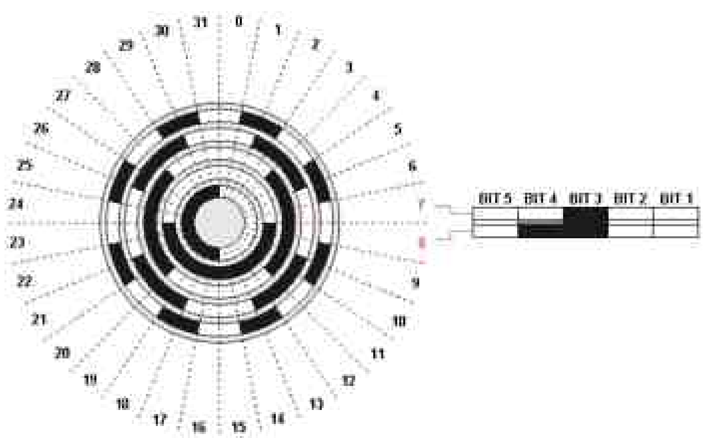


Fig. 7

Absolute encoders, both optical and magnetic, have as their main operating characteristic the number of steps, that is, unique codes per revolution and the number of such revolutions, and the initial installation and initialization of the sensor is not required. Therefore, absolute encoders do not lose their position when the voltage fails.

The most common types of signal outputs are Gray code, parallel code, Profibus-DP, CANopen, DeviceNet, SSI, LWL interfaces, through which sensors are also programmed.

An absolute encoder refers to a type of encoder that executes a unique code for each shaft position. Unlike an incremental encoder, a impulse counter is not needed, as the angle of rotation is always known. An absolute encoder generates a signal both during rotation and at rest. An absolute encoder wheel differs from a step encoder wheel in that it has several concentric tracks. Each track generates a unique binary code for a particular shaft position.



An absolute encoder does not lose its value when power is lost and does not need to be returned to its original position. The absolute encoder signal is not subject to interference and does not require fine shaft alignment. Also, even if the encoded signal cannot be read by the encoder if, for example, the shaft rotates too fast, the correct rotation angle will be registered when the rotation velocity is reduced. The absolute encoder is vibration resistant.

The Gray code is preferable to the usual binary one in that it has the property of the continuity of the binary combination: a change in the encoded number by one corresponds to a change in the code combination in only one bit. It is built on the basis of binary according to the following rule: the most significant digit remains unchanged; each subsequent bit is inverted if the previous bit of the original binary code is equal to one. This construction algorithm can be formally represented as the result of modulo two addition of the original binary code combination with the same combination, but shifted one bit to the right. In this case, the rightmost bit of the shifted combination is discarded.

Thus, the Gray code is the so-called one-step code, because when moving from one number to another, only one bit always changes. An error in reading information from a mechanical code disk when moving from one number to another will only lead to the fact that the transition from one position to another will be only slightly shifted in time, however, the issuance of a completely incorrect value of the angular position when moving from one position to another is completely excluded . The advantage of the Gray code is also its ability to mirror information. So, by inverting the most significant bit, you can easily change the direction of counting and, thus, match the actual (physical) direction of rotation of the axis. The reversal of the count direction can be easily changed by controlling the so-called Complement input.

Since the information expressed in the Gray code is purely encoded and does not carry real numerical information, it must first be converted into a standard binary code before further processing. This is done using a code converter (Grey-Binar decoder), which, fortunately, is easily implemented using a chain of XOR logic elements, both in software and in hardware (see diagram below).

The table shows that when moving from one number to another (adjacent), only one bit of information changes its status if the number is represented by a Gray code, while in a binary code several bits can change their status at the same time. Gray code is an output, hence it never has a read error and is used in many absolute encoders.

Decimal code	Binary code				Gray code			
	2^3	2^2	2^1	2^0				
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Bits that change their status when moving from one number to another are marked in red.

Single turn encoder.

Single-turn (Single - Turn) sensors are sensors that give absolute values within one revolution, i.e. within a 360° radius. After one revolution, the code is completely traversed and starts again from its initial value. These sensors are mainly used to measure the angle of rotation and are used, for example, in antenna systems, eccentric crank presses, etc.

Multi-turn encoder

Linear movements require the use of a measuring system with n-number of revolutions. For example, with linear actuators or measuring tasks with a toothed measuring rod, the use of single-turn encoders is unacceptable. In this case, sensors come to the rescue, where, in addition to measuring the angle of rotation within one revolution, the number of revolutions is also recorded using an additionally built-in transmission mechanism, i.e. a kind of reducer of several coded optical disks, thus forming a multi-turn encoder (Multi-Turn).

Optical encoders

Optical encoders have a glass disk rigidly and coaxially fixed to the shaft with a accuracy optical scale. When the object rotates, the optocoupler reads information, and the electronics converts it into a sequence of discrete electrical impulses. Absolute optical encoders are rotation angle sensors, where each shaft position corresponds to a unique digital output code, which, along with the number of revolutions, is the main operating parameter of the sensor. Absolute optical encoders, as well as incremental encoders, read and fix the rotation parameters of the optical disk.

Magnetic encoders

Magnetic encoders register with high accuracy the passage of the magnetic poles of a rotating magnetic element directly near the sensing element, converting this data into the corresponding digital code.

Mechanical and optical encoders with serial output

Contain a disk of dielectric or glass with applied convex, conductive or opaque areas. The reading of the absolute angle of rotation of the disk is made by a line of switches or contacts in the case of a mechanical circuit and by a line of optocouplers in the case of an optical one. The output signals are a Gray code that allows you to get rid of the ambiguity of signal interpretation.

Глава 12.Indication

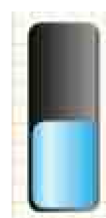
SCADA ZETView system has a number of various indicators used for results representation. For numerical values input a liquid crystal display and digital indicator are used. Liquid crystal and digital indicator are often used for voltmeter and other measuring instruments results representation. Just like many other ZETView components, the indicators are used for signals processing in real time mode and allow to display the last received value till a new value is obtained.

For indication of measured values in real time mode, level and pointed indicators are used. In the case if, e. g., it is necessary to inform the operator of process completion, a process indicator is used. As you can see in the figure, ZETView has two types of process indicators: with numerical and control signal input. The first type is used to show the degree of process completion, while the second type produces response to input signal and is used in order to inform the operator that the process is being run or is temporary suspended due to mistakes in settings or project development. Besides, there is a light indicator changing its color depending on logical status at the input.

12.1.Vertical process indicator

The component "**Vertical process indicator**" is designed for graphical display of the status of the process with different settings.

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>
</td><td></td></tr>
</table>

Setting:

➢Input:

- Number - the display data on the progress bar.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- [Are available by the link below.](154)

➢Custom private properties (the default value is shown in parentheses):

- maxValue (0) - the maximum value that the indicator can display.
- currValue (0) - Set the current status of the indicator.
- fillColorUp (00a2e8) - Set the up color of the indicator gradient.
- fillColorDown (ffffff) - Set the down color of the indicator gradient.

- backColorUp (000000) - Set the up color of the background gradient.
- backColorDown (c0c0c0) - Set the down color of the background gradient.
- FrameVisible (false) - Set the visibility of the frame.
- ProcentVisible (false) - Set percentage display.
- style (convex) - Set the style of the process-bar:
  1 - oblique lines;
  2 - crushed;
  3 - convex;
  4 - Vista.
- AnimationEnabled (true) - Set the animation is Enable/Disable.
- Enabled (true) - Set the component is Enable/Disable.
- backColor (ffffff) - Set the background color of the indicator.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Number - The value of the number to calculate the display on the progress bar (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

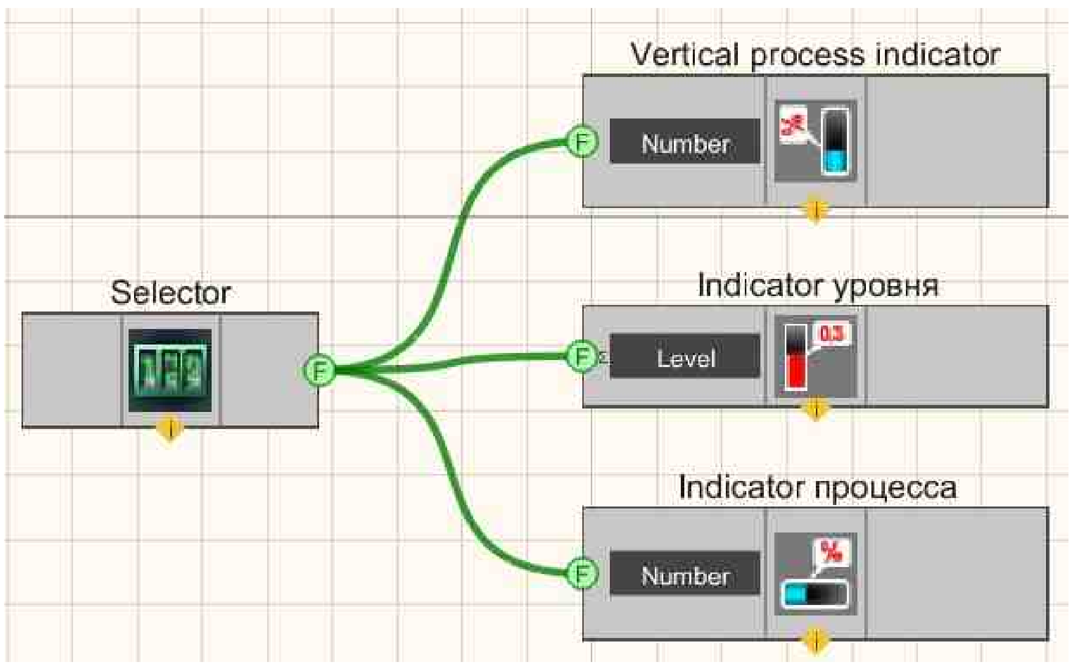
- FLOAT maxValue - Set the maximum value (from minus infinity to plus infinity).
- FLOAT currValue - Set the current value (from minus infinity to plus infinity).
- LONG fillColorUp - Set the up color of the indicator gradient (any number).
- LONG fillColorDown - Set the down color of the indicator gradient (any number).
- LONG backColorUp - Set the up color of the background gradient (any number).
- LONG backColorDown - Set the down color of the background gradient (any number).
- VARIANT_BOOL FrameVisible - Set frame visibility:
  true - the output value is given when the frame visibility is enabled;
  false - the output value is not given when the border visibility is enabled.
- VARIANT_BOOL ProcentVisible - Set percentage display:
  true - the output value is given when the display of percentages is enabled;

false - the output value is not given when percentage display is enabled.

- BSTR Style - Set the progress-bar style: Oblique lines; Shattered; Convex; Vista (string).
- VARIANT_BOOL AnimationEnabled - Set the animation Enable/Disable:
  true - the output value is given when the animation is enabled;
  false - the output value is not given when the animation is enabled.
- VARIANT_BOOL Enabled - Set the component Enable/Disable:
  true - the output value is given when the component is enabled;
  false - the output value is not given when the component is enabled.
- LONG backColor - Set the background of the indicator (any number).

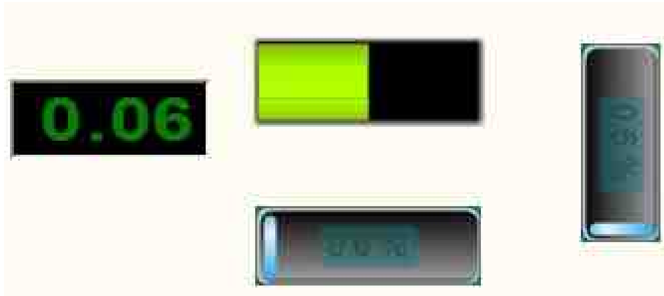
Example

Project in SCADA ZETView



This project shows the operation of the Vertical process indicator 649, Level indicator 659, Horizontal process indicator 652 components show the level of the signal coming from the Selector 428.

Project operation result

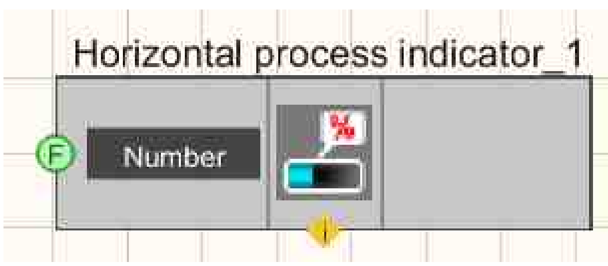
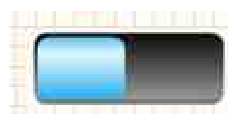


View the example in ZetView

12.2.Horizontal process indicator

The component "**Horizontal process indicator**" is designed for graphical display of the status of the process with different settings.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Number - the display data on the progress bar.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- maxValue (0) - the maximum value that the indicator can display.

- currValue (0) - Set the current status of the indicator.
- fillColorUp (00a2e8) - Set the up color of the indicator gradient.
- fillColorDown (ffffff) - Set the down color of the indicator gradient.
- backColorUp (00000) - Set the up color of the background gradient.
- backColorDown (c0c0c0) - Set the down color of the background gradient.
- FrameVisible (false) - Set the visibility of the frame.
- ProcentVisible (false) - Set percentage display.
- style (convex) - Set the style of the processbar:
  1 - oblique lines
  2 - crushed
  3 - convex
  4 - Vista
  5 - Volumetric.
- AnimationEnabled (true) - Set the animation is Enable/Disable.
- Enabled (true) - Set the component is Enable/Disable.
- backColor (ffffff) - Set the background color of the indicator.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Number - The value of the number to calculate the display on the progress bar (any number).

➢**Custom common properties:**

- Are available by the link below. 154

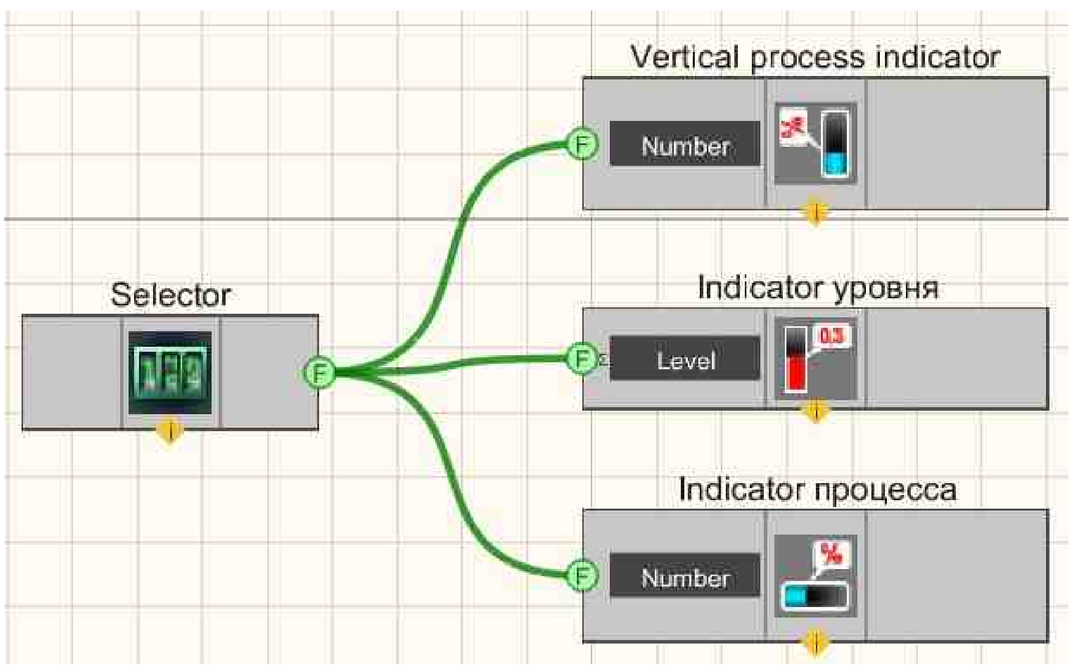
➢**Custom private properties (the default value is shown in parentheses):**

- LONG backColor - Set the background of the indicator (any number).
- FLOAT maxValue - Set the maximum value (from minus infinity to plus infinity).
- FLOAT currValue - Set the current value (from minus infinity to plus infinity).
- LONG fillColorUp - Set the up color of the indicator gradient (any number).
- LONG fillColorDown - Set the down color of the indicator gradient (any number).
- LONG backColorUp - Set the up color of the background gradient (any number).
- LONG backColorDown - Set the down color of the background gradient (any number).

- VARIANT_BOOL FrameVisible - Set frame visibility:
  true - the output value is given when the frame visibility is enabled;
  false - the output value is not given when the border visibility is enabled.
- VARIANT_BOOL ProcentVisible - Set percentage display:
  true - the output value is given when the display of percentages is enabled;
  false - the output value is not given when percentage display is enabled.
- BSTR Style - Set the progress-bar style: Oblique lines; Shattered; Convex; Vista (string).
- VARIANT_BOOL AnimationEnabled - Set the animation Enable/Disable:
  true - the output value is given when the animation is enabled;
  false - the output value is not given when the animation is enabled.
- VARIANT_BOOL Enabled - Set the component Enable/Disable:
  true - the output value is given when the component is enabled;
  false - the output value is not given when the component is enabled.
- LONG backColor - Set the background of the indicator (any number).

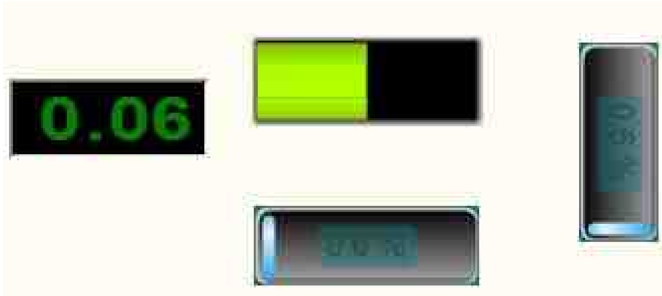
Example

Project in SCADA ZETView



This project shows the operation of the Vertical process indicator[652], Level indicator[659], Horizontal process indicator[652] components show the level of the signal coming from the Selector[428].

Project operation result



View the example in ZetView

12.3.Liquid crystal display

The component "**Liquid crystal display**" is designed for graphic display of numerical parameters and values.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Number - the display data on the Liquid crystal display.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- IndValue (0) - the displayed value that comes to the indicator.
- BackColor (7f7f00) - Set the background color.

- TextColor (000000) - Set the text color.
- TextChangeColor (000000) - Set the color of the digit with the changed value.
- TextSize (40) - Set the text size.
- Enabled (true) - Set the indicator is enable/disable.
- InactiveColor (c0c0c0) - Controls the background color when inactive.
- precision (0.01) - Set the display precision.
- digits (3) - Set the number of digits before the decimal point.
- BoldFont (true) - Set and disables bold font.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Number - The value of the number to calculate the display on the progress bar (any number).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- FLOAT IndValue - Set the displayed value (from minus infinity to plus infinity).
- LONG BackColor - Set the background color (any number).
- LONG TextColor - Set the text color (any number).
- LONG TextChangeColor - Set the color of the bit with the changed value (any number).
- SHORT TextSize - Set the text size (any number).
- VARIANT_BOOL Enable - Set the level is Active\Not active:
  true - the output value is given when the level is active;
  false - the output value is not given when the level is not active.
- LONG InactiveColor - Set the background color of the not active status (any number).
- BSTR Precision - Set the precision: 1; 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- BSTR Digits - Set digits before decimal point (string): 1; 2; 3; 4; 5; 6 (string).
- VARIANT_BOOL BoldFont - Set the font is Bold\Non-bold:
  true - the output value is given when the font is bold;
  false - the output value is not given when the font is bold;

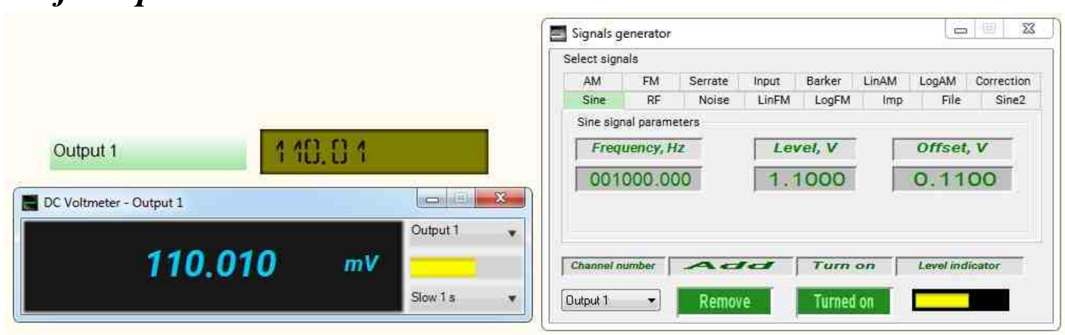
➢**Methods:**

- FLOAT IndValue - Set the displayed value (from minus infinity to plus infinity).
- LONG BackColor - Set the background color (any number).

Example

Project operation result:

Project operation result

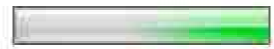


View the example in ZetView

12.4.Process indicator

The component "**Process indicator**" is designed for graphical display of the binary status of the process. Works from the button.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Process - the channel with logical values-statuses.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- process (0) - process status value.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

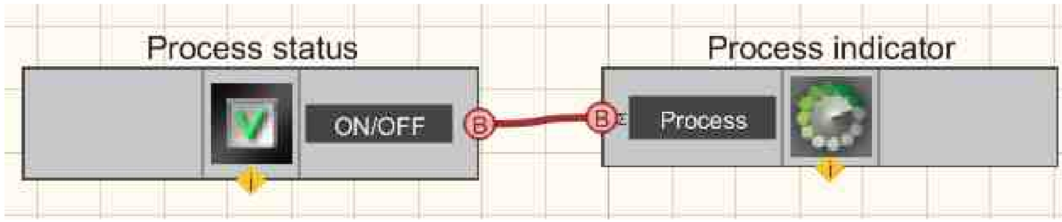
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- LONG Process - Set the process visibility status when the project is running (from 0 - not visible, 1 - visible).

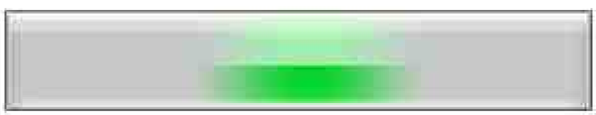
Example

Project operation result:



When the status of the CheckBox 685 component is pressed on the monitor screen (in operator mode), the Process indicator 657 becomes visible and a process progress bar appears.

Project operation result

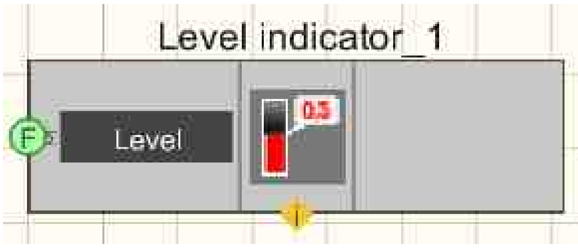


12.5.Level indicator

The component "**Level indicator**" is designed for graphic display of the integral level of the signal. It works on the principle of the Integrator, that is, it indicates a level equal to the sum of all previous values and the current one.

Takes a value up to 1.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Level - current signal level.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- [Are available by the link below.](154)

Custom private properties (the default value is shown in parentheses):
- Enabled (true) - Set the indicator is enable/disable.
- InactiveColor (c0c0c0) - Set the background color in the inactive status.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Level - Data value for level calculation (any number).

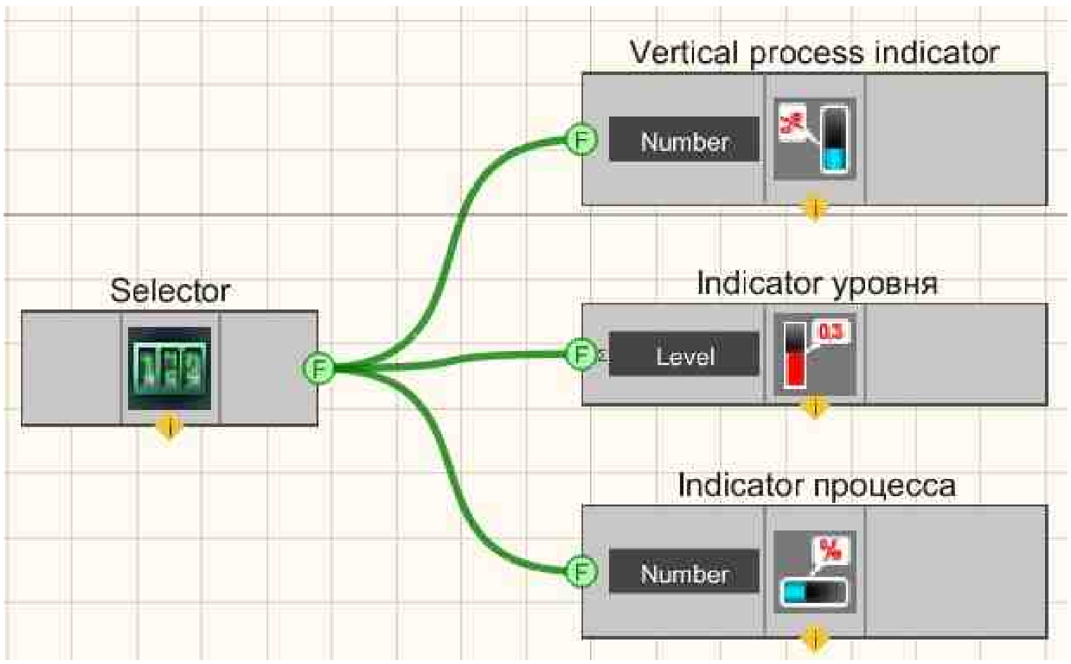
➢**Custom common properties:**
- [Are available by the link below.](154)

➢**Custom Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL Enabled - Set the level is Active\Not active:
  true - the output value is given when the level is active;
  false - the output value is not given when the level is not active.
- LONG InactiveColor - Set the background color in the not active status (any number).

Example

Project in SCADA ZETView



This project shows the operation of the Vertical process indicator 659, Level indicator 659, Horizontal process indicator 652 components show the level of the signal coming from the Selector 428.

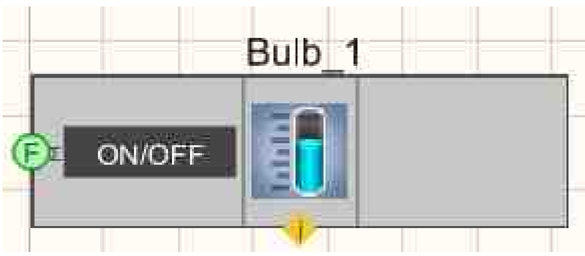
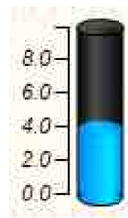
Project operation result

View the example in ZetView

12.6.Bulb

The component "**Bulb**" is designed for graphical display of data in the form of a bulb indicator.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➤Input:
- On\Off - value to display on the indicator.

➤ Output:

Doesn't have.

Custom common properties:

➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**
- TextSize (8) - Set the text size.
- textstyle (FontStyleItalic) - Set the text style.
- TextColor (000000) - Set the text color.
- CurrentValue (0) - Set the current value.
- DigitFrom (0) - Set the start value.
- DigitTo (10) - Set the end value.
- accuracy (0.1) - Set the measurement accuracy.
- LinearsUnderline (true) - Set the underline of the ruler.
- LinearVisible (true) - Set the line visibility.
- linearorientation (left) - Set the display side of the ruler:
  1. left - left side;
  2. right - right side.
- BackColorUp (000000) - Set the up background color.

- BackColorDown (808080) - Set the down background color.
- LiquidUpColor (00f4ff) - Set the up color of the cursor gradient.
- LiquidDownColor (00f7ff) - Set the down color of cursor gradient.
- Enabled (true) - Set the component is Enable/Disable.
- AnimationEnabled (true) - Set the animation is Enable/Disable.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- On\Off - the value that will be displayed on the indicator (any number).

➢**Custom common properties:**
- Are available by the link below. 154

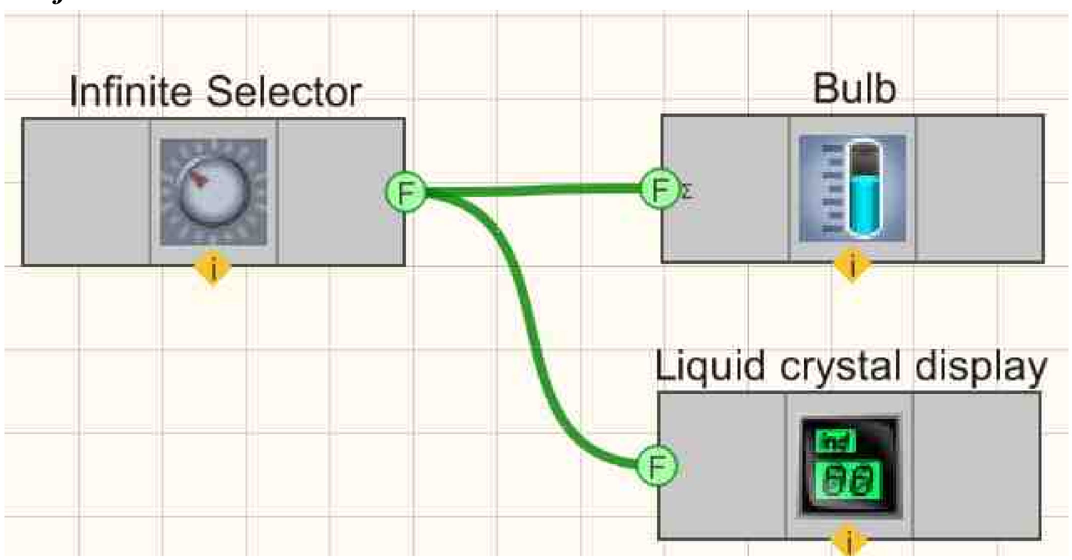
➢**Custom private properties (the default value is shown in parentheses):**

- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set the text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- LONG TextColor - Set the text color (any number).
- float CurrentValue - Set the current value (from minus infinity to plus infinity).
- float DigitFrom - Set the start value (from minus infinity to plus infinity).
- float DigitTo - Set the end value (any value, but greater than the start value).
- BSTR Accuracy - Set the accuracy: 1; 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- VARIANT_BOOL LinearsUnderline - Set the underline of the ruler:
  true - the output value is given when the ruler underline is enabled;
  false - the output value is not given when ruler underline is enabled.
- VARIANT_BOOL LinearVisible - Set the linear visibility:
  true - the output value is given when the ruler visibility is enabled;
  false - the output value is not given when ruler visibility is enabled;

- BSTR LinearOrientation - Set the text: left side; right side (string).
- LONG BackColorUp - Set the up background color (any number).
- LONG BackColorDown - Set the down background color (any number).
- LONG LiquidUpColor - Set the up color of the cursor gradient (any number).
- LONG LiquidDownColor - Set the down color of the cursor gradient (any number).
- VARIANT_BOOL Enabled - Set the component Enable/Disable:
  true - the output value is given when the component is enabled;
  false - the output value is not given when the component is enabled.
- VARIANT_BOOL AnimationEnabled - Set the animation Enable/Disable:
  true - the output value is given when the animation is enabled;
  false - the output value is not given when the animation is enabled.

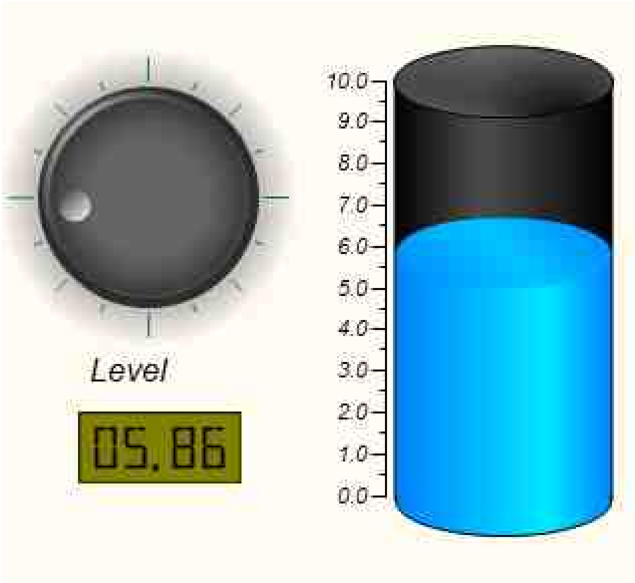
Example

Project in SCADA ZETView



This project shows the operation of the Infinite Selector 446 component is used to connect the Bulb 662 indicator and allows us to increase or decrease the level in the Bulb. The Selector 435 allows us to determine the exact value of the flask level.

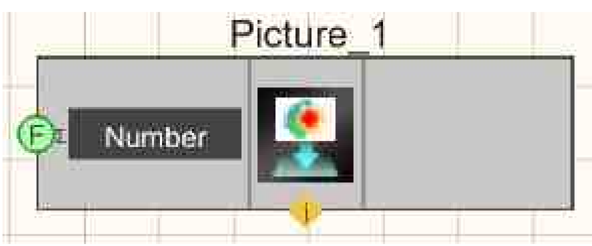
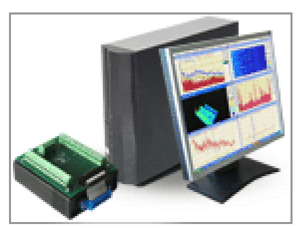
Project operation result



12.7.Picture

The component "**Picture**" is designed to organize the selection of a specific image from a set depending on the input value.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➤Input:

- Number - the serial number of the displayed picture, which must be displayed in the operator mode.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- CurrentPicture (0) - Set the current picture number.
- Pic1_FileName - Select of the picture that will be displayed on the display in operator mode when the number 1 component is input.
- ...
- Pic10_FileName - Select of the picture that will be displayed on the display in operator mode when the number 10 component is input.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- Input:
- Number - The value of the picture number to display on the display in operator mode (from 0 to 9).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG CurrentPicture - Set the current picture number (from 0 to 9).
- BSTR Pic1_FileName - Set the name of the first picture (any string).
- BSTR Pic2_FileName - Set the name of the first picture (any string).
- BSTR Pic3_FileName - Set the name of the first picture (any string).
- BSTR Pic4_FileName - Set the name of the first picture (any string).
- BSTR Pic5_FileName - Set the name of the first picture (any string).
- BSTR Pic6_FileName - Set the name of the first picture (any string).
- BSTR Pic7_FileName - Set the name of the first picture (any string).
- BSTR Pic8_FileName - Set the name of the first picture (any string).

- BSTR Pic9_FileName - Set the name of the first picture (any string).
- BSTR Pic10_FileName - Set the name of the first picture (any string).

➤**Methods:**

LONG RefreshData(OLE_HANDLE sourceWHND, BSTR helpString, LONG parametr, LONG parametrType) - Returns data where:

1) OLE_HANDLE sourceWHND - image handle in OLE_HANDLE format.
2) BSTR helpString - contains information about the name of the component (string).
3) LONG parametr - parameters of transmitted data (from -2147483647 before 2147483647)
4) LONG parameterType - defines the type of parameters:

TYPE_LONG 0
TYPE_FLOAT 1
TYPE_PARAM 2
TYPE_MANAGE 3
TYPE_AUTOFORMAT 100
TYPE_POINTER 101
TYPE_POINTER_STRING 104
TYPE_STRING 4
TYPE_CHANNEL 5
TYPE_COLORSTRING 6

//float with quality marker
TYPE_FLOAT_BAD 7
TYPE_FLOAT_UNCERTAIN 8
TYPE_FLOAT_GOOD 1

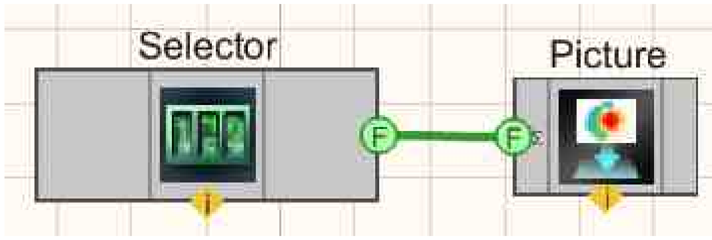
//integrator with quality marker
TYPE_LONG_BAD 10
TYPE_LONG_UNCERTAIN 11
TYPE_LONG_GOOD 12

//Interface pointer
TYPE_INTERFACE 13

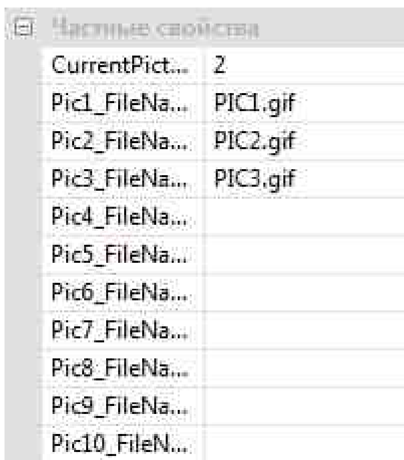
// double
TYPE_DOUBLE 14
TYPE_DOUBLE_GOOD 14
DOUBLE_BAD 15
TYPE_DOUBLE_UNCERTAIN 16

Example

Project in SCADA ZETView



The select of pictures for each of the values is done by editing the Custom private properties of the component:



Частные свойства	
CurrentPict...	2
Pic1_FileNa...	PIC1.gif
Pic2_FileNa...	PIC2.gif
Pic3_FileNa...	PIC3.gif
Pic4_FileNa...	
Pic5_FileNa...	
Pic6_FileNa...	
Pic7_FileNa...	
Pic8_FileNa...	
Pic9_FileNa...	
Pic10_FileN...	

This project shows the operation of the Picture 665 components are used to display images that depend on the entered number. The Selector 428 component is used to select a particular picture.

Project operation result

View the example in ZetView

12.8.Light indicator

The component "**Light indicator**" is designed for graphical display of binary values of the input status in the form of an indicator that changes color.

Appearance of the component:

<table>
<tr><td>Developer interface</td><td>Operator interface</td></tr>
<tr><td>
</td><td></td></tr>
</table>

Setting:

➢Input:

- On\Off - "switch", status true, false.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- onColor (00ff00) - Set the indicator color when on.
- offColor (ff0000) - Set the indicator color in the off status.
- Enabled (true) - Set the general status of the indicator (enabled/disabled).
- DisableColor (c0c0c0) - Set the color when not active.
- currentStatus (true) - Set the current status of the indicator.
- status (true) - the same as currentStatus.
- figure (Circle) - Set the geometric form of the indicator in mode of the operator:
- 1. Circle.
- 2. Square.

P ***Programming***

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

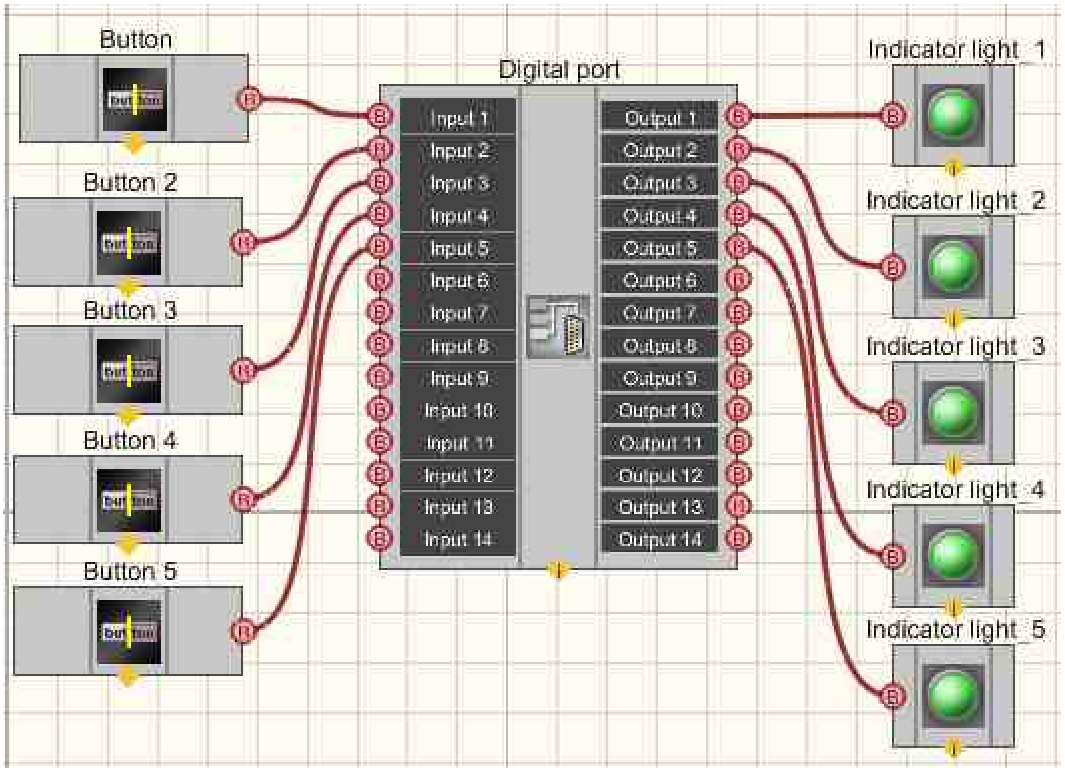
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG onColor - Set the color when on (any number).
- LONG offColor - Set the color when off (any number).
- VARIANT_BOOL Enable - Set the active\Not active :
  true - the output value is given when the level is active;
  false - the output value is not given when the level is not active.
- LONG DisableColor - Set the color in the not active status (any number).
- VARIANT_BOOL currentStatus - Set the current status:
  true - the output value is given in the current status;
  false - the output value is not given in the current status.
- VARIANT_BOOL status - Set the status is Enable/Disable:
  true - the output value is given when the status is enabled;
  false - the output value is not given when the status is enabled.
- BSTR Figure - Set the form of the light indicator Figure: Circle; Square (string).

Example

Project in SCADA ZETView



This project shows the operation of the Digital port 1255 component is a bridge between a digital input and an output. The Fixed button 691 allows you to control the status of the input. The Light indicator 669 indicates the status of the input.

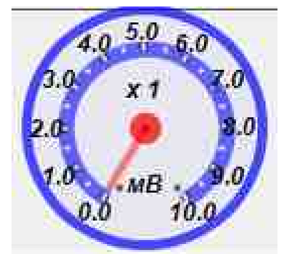
Project operation result

View the example in ZetView

12.9.Dial gauge

The component "**Dial gauge**" is designed for graphical display of data in the form of an arrow type indicator.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Data - data to display on the indicator.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Max (10) - the maximum value that the indicator can display.
- DivCount (10) - the total number of divisions on the scale.
- Value (0) - the current value.
- TextSize (9) - the text size of numbers on the indicator.
- textstyle (FontStyleBoldItalic) - the text style of numbers on the indicator.
- BackColor (0000ff) - the color of the background and indicator rims.
- ArrowColor (ff0000) - the arrow color.
- ArrowWidth (4) - the arrow width.
- haveArrow (false) - Set the form of the pointer in the form of an arrow is enable/disable.
- digits (0.1) - Set the accuracy.
- Enabled (true) - Set the general status of the indicator (enabled/disabled).
- Amplifuer (1) - Set the scale multiplier.
- Unit (mV) - Set the unit of measure.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Data - the value to display (any number).

➢**Custom common properties:**

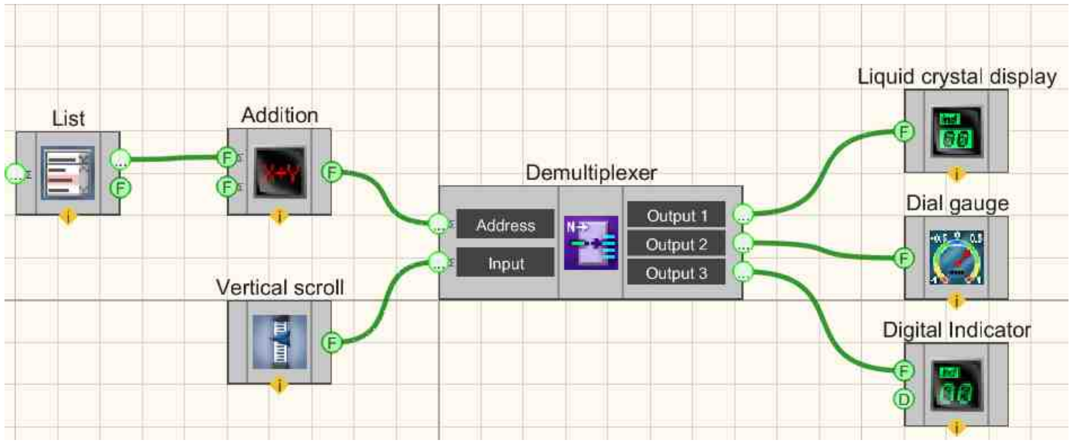
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

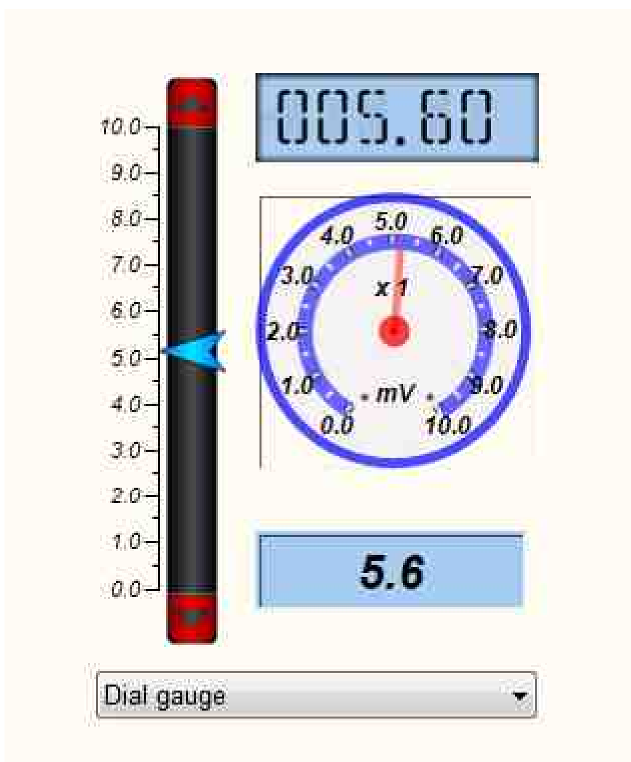
- FLOAT Max - Set the maximum value that the indicator can display (any number).
- LONG DivCount - Set the total number of divisions on the scale (any number).
- FLOAT Value - Set the current value (from minus infinity to plus infinity).
- LONG TextSize - Set the text size of numbers on the indicator (any number).
- BSTR TextStyle - Set the text style of numbers on the indicator (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- LONG BackColor - Set the color of the background and indicator rims. (any number).
- LONG ArrowColor - Set the color of the arrow (any number).
- LONG ArrowWidth - Set the pointer width (any number).
- VARIANT_BOOL haveArrow - Set the arrow pointer:
  true - the output value is given when data arrives;
  false - the output value is not given when data arrives.
- BSTR Digits - Set the accuracy: 1; 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- VARIANT_BOOL Enable - Set activity:
  true - the output value is given when data arrives;
  false - the output value is not given when data arrives.
- LONG Amplifier - Set the multiplier (any number).
- BSTR Unit - Set the unit of measurement (string).

Example

Project in SCADA ZETView



Project operation result

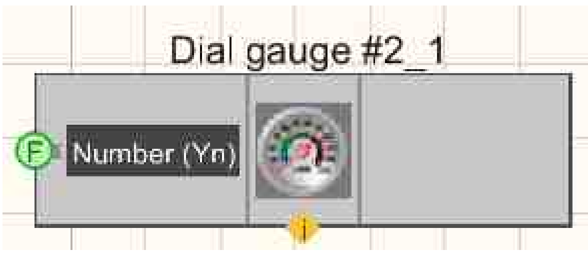


View the example in ZetView

12.10.Dial gauge No. 2

The component "**Dial gauge No. 2**" is designed for graphical display of data in the form of an arrow type indicator.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Number (Yn) - values to be displayed on the indicator.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- CurrentValue (0) - the current value;
- style (Volumetric) - the style display type (Volumetric/Plane);
- DrawBorder (false) - the display a frame (No/Yes);
- BorderColor (003366) - the border color;
- GaugeColor (b9b9b9) - the color of the outer part of the gauge;
- DialBackColor (a5a5a5) - the dial color;
- DigitsColor (000000) - color of numbers;
- BigLinesColor (003366) - color of big lines;
- LittleLinesColor (003366) - color of small lines;
- NeedleColor (fa000a) - the color of the arrow;
- UnderlineColor (fafa0a) - the color of the underline ;
- UnderlineVisible (false) - the display of an underline (No/Yes);

- DigitsVisible (true) - the display of numbers (No/Yes);
- BigLinesLength (10) - the length of big lines;
- LitteleLinesLength (5) - the length of small lines;
- BigLinesWidth (2) - the width of big lines;
- LitteleLinesWidth (1) - the width of small lines;
- UnderlineWidth (1) - the width of underline;
- NeedleLength (74) - the length of the arrow;
- needlewidth (6) - a parameter showing how many times the width of the half of the arrow is less than the width of the round base of the arrow;
- NeedleArrowWidth (1) - the width of the arrow tip;
- AngleFrom (-40) - the angle from which the scale starts;
- AngleTo (220) - the angle to which the scale is marked;
- ValueFrom (0) - the value at the start point;
- ValueTo (260) - the value at the end point;
- Multiplexor (1000) - the number displayed under the arrow
- fontstyle (FontStyleBold) - the font style;
- FontSize (10) - the font size;
- WidthZoneLine (10) - the width of the line that displays the zones;
- RedZonePercent (30) - the percentage value of the red zone area;
- YellowZonePercent (25) - the percentage value of the yellow zone area;
- GreenZonePercent (44.9) - the percentage value of the green zone area;
- GradientZoneColorTone (a5a5a5) - set the tone of the line that displays zones when the GDT_APART_GRADIENT style;
- digitspos (numbers over lines) - arrangement of numbers relative to lines (numbers over lines/numbers under lines);
- llvs (10) - frequency of small lines;
- zonedialtype (Separately gradient) - the type of line that displays the zones:
    1. Missing;
    2. Separately gradient;
    3. Separately clear boundaries;
    4. Separately only red;
    5. At the lines;
    6. At the lines of only the red;
- needlestyle (Style_2) - Set the arrow style (from Style_1 to Style 7).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Number (Yn) - Number value (Yn) for display calculation (any number).

➢**Custom common properties:**
- Are available by the link below. 154

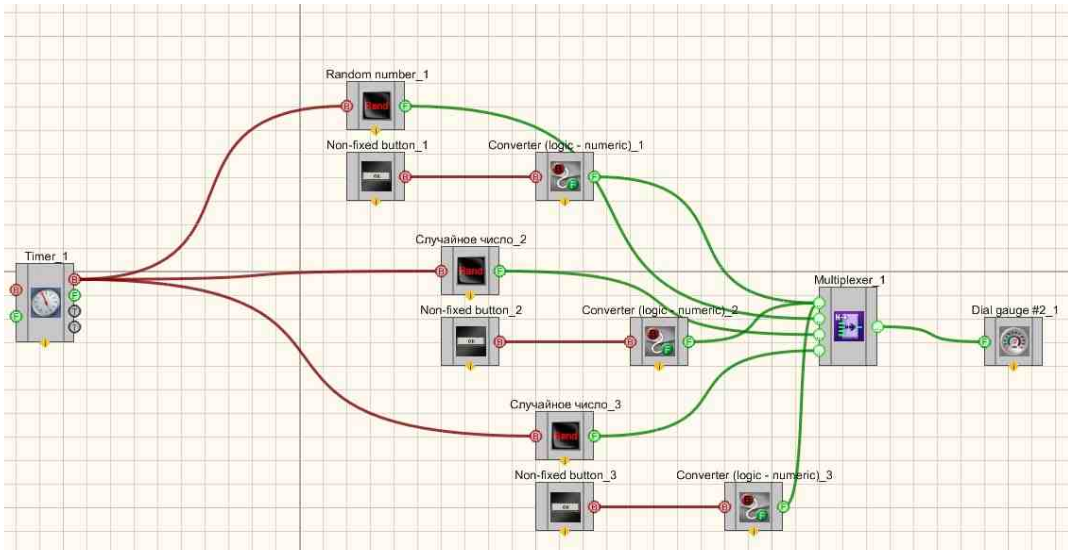
➢**Custom private properties (the default value is shown in parentheses):**

- float CurrentValue - Set the current value (from minus infinity to plus infinity).
- BSTR Style - Set the style display type: Volumetric; Plane (string).
- VARIANT_BOOL DrawBorder - Set the display of the frame:
  true - the output value is given when the frame is displayed;
  false - the output value is not given when the frame is displayed.
- LONG BorderColor - Set the border color (any number).
- LONG GaugeColor - Set the color of the outer part of the gauge (any number).
- LONG DialBackColor - Set the color of the dial (any number).
- LONG DigitsColor - Set the color of numbers (any number).
- LONG BigLinesColor - Set the color of big lines (any number).
- LONG LittleLinesColor - Set the color of small lines (any number).
- LONG NeedleColor - Set the color of the arrow (any number).
- LONG UnderlineColor - Set the color of the underline (any number).
- VARIANT_BOOL UnderlineVisible - Set the display of an underline:
  true - the output value is given when the underline is displayed;
  false - the output value is not given when an underline is displayed.
- VARIANT_BOOL DigitsVisible - Set the display of numbers:
  true - the output value is given when displaying numbers;
  false - the output value is not given when displaying numbers.
- LONG BigLinesLength - Set the length of big lines (any number).
- LONG LittleLinesLength - Set the length of small lines (any number).
- LONG BiglinesWidth - Set the width of big lines (any number).
- LONG LittleLinesWidth - Set the width of small lines (any number).
- LONG UnderlineWidth - Set the width of the underline (any number).
- LONG NeedleLength - Set the length of the arrow (any number).
- BSTR NeedleWidth - Set a parameter show how many times the width of the half of the arrow is less than the width of the round base of the arrow: 2; 4; 6; 8; 10; 12; 16; 20; 24 (string).
- LONG NeedleArrowWidth - Set the width of the arrow tip (any number).
- LONG AngleFrom - Set the angle from which the scale starts (any number).
- LONG AngleTo - Set the angle to which the scale is marked (any number).
- FLOAT ValueFrom - Set the value at the start point (any value, but less than the value at the end point).

- FLOAT ValueTo - Set the value at the end point (any value, but greater than the value at the start point).
- LONG Multiplexor - Set the number that is displayed under the arrow (any number).
- BSTR FontStyle - Set text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- LONG FontSize - Set the font size (any number).
- LONG WidthZoneLine - Set the width of the line that displays the zones (any number).
- FLOAT RedZonePercent - Set the percentage value of the red zone area (from minus infinity to plus infinity).
- FLOAT YellowZonePercent - Set the percentage value of the yellow zone area (from minus infinity to plus infinity).
- FLOAT GreenZonePercent - Set the percentage value of the green zone area (from minus infinity to plus infinity).
- LONG GradientZoneColorTone - Set the tone of the line that displays the zones when the style is set to GDT_APART_GRADIENT (any number).
- BSTR DigitsPos - Set the location of numbers relative to lines: numbers over lines; numbers under lines (string).
- BSTR LLVS - Set the frequency of small lines: 1; 2; 4; 5; 10 (string).
- BSTR ZoneDialType - Set the type of line that displays zones (string):
  1. Missing;
  2. Separately gradient;
  3. Separately clear boundaries;
  4. Separately only red;
  5. At the lines;
  6. At the lines of only the red;
- BSTR NeedleStyle - Set the arrow style: Style_1; Style_2; Style_3; Style_4; Style_5; Style_6; Style_7 (string).

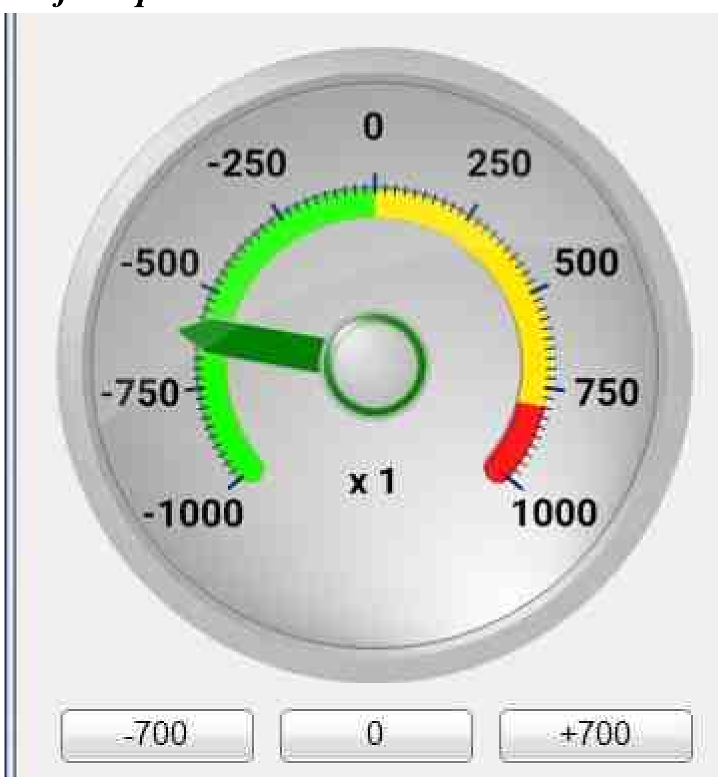
Example

Project in SCADA ZETView



Dial gauge No. 2

Project operation results

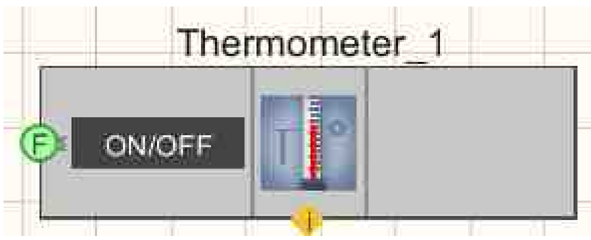
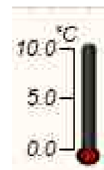


View the example in ZetView

12.11.Thermometer

The component "**Thermometer**" is designed for graphical display of data in the form of a thermometer indicator.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- On\Off - the value that will be displayed on the thermometer.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- TextSize (8) - set the text size.
- textstyle (FontStyleItalic) - set the text style.
- TextColor (000000) - set the text color.
- CurrentValue (0) - set the current value.
- DigitFrom (0) - set the start value.
- DigitTo (10) - set the end value.
- accuracy (0.1) - set the measurement accuracy.
- LinearsUnderline (true) - set the underline of the ruler:
- LinearVisible (true) - set the ruler visibility.
- linearorientation (left) - set the display side of the ruler:
  1. left - left side;
  2. right - right side.
- BackColorUp (000000) - set the up background color.

- BackColorDown (808080) - set the down background color.
- LiquidUpColor (f40000) - set the up color of the cursor gradient.
- LiquidDownColor (7f0000) - set the down color of the cursor gradient.
- Enabled (true) - set the enable/disable the component.
- measuring (°C) - set the units of measurement.
- AnimationEnabled (true) - set the enable/disable animation.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- On\Off - the value that will be displayed on the indicator (any number).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- LONG TextColor - Set text color (any number).
- float CurrentValue - Set the current value (from minus infinity to plus infinity).
- float DigitFrom - Set the start value (from minus infinity to plus infinity).
- float DigitTo - Set the end (any value, but greater than the start value).
- BSTR Accuracy - Set the accuracy: 1; 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- VARIANT_BOOL LinearsUnderline - Set the underline of the ruler:
  true - the output value is given when the ruler underline is enabled;
  false - the output value is not given when ruler underline is enabled.
- VARIANT_BOOL LinearVisible - Set the ruler visibility.
  true - the output value is given when the ruler visibility is turned on;

false - the output value is not given when ruler visibility is turned on.

- BSTR LinearOrientation - Set the text style: left - left side; right - right side (string).
- LONG BackColorUp - Set the up background color (any number).
- LONG BackColorDown - Set the down background color (any number).
- LONG LiquidUpColor - Set the up color of the cursor gradient (any number).
- LONG LiquidDownColor - Set the down color of the cursor gradient (any number).
- VARIANT_BOOL Enabled - Set the enable/disable component:
  true - the output value is given when the component is enabled;
  false - the output value is not given when the component is enabled.
- BSTR Measuring - Set the unit of measurement: °C; °F; K; °Ra; °H; °D; °L (string).
- VARIANT_BOOL AnimationEnabled - Set the animation Enable/Disable:
  true - the output value is given when the animation is enabled;
  false - the output value is not given when the animation is enabled.

12.12.Digital Indicator

The component "**Digital Indicator**" is designed for graphical display of data in the form of a digital indicator.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➤Input:

- Number - the data to be displayed.
- Number - Number is the double precision data to be displayed.

➤ Output:

Doesn't have.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- IndValue (0) - displayed value.
- BackColor (7f7f00) - background color.
- TextColor (000000) - text color.
- TextSize (14) - text size of numbers.
- textstyle (FontBoldStyleItalic) - text style of numbers.
- digits (0.1) - set the precision.
- Enabled (true) - Activate/Not activate of the indicator.
- InactiveColor (c0c0c0) - the background color in not active status.
- valuetype (Decimal) - number display format:
  1. Octal;
  2. Decimal;
  3. Hexadecimal.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Number - Number value for display calculation (any number).
- Number - Number value for display calculation (any double precision number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT IndValue - Set the displayed value (from minus infinity to plus infinity).
- LONG BackColor - Set the background color (any number).
- LONG TextColor - Set text color (any number).
- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set text style (string):

  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;

FontStyleBoldItalic;
FontStyleUnderline;
FontStyleStrikeout.

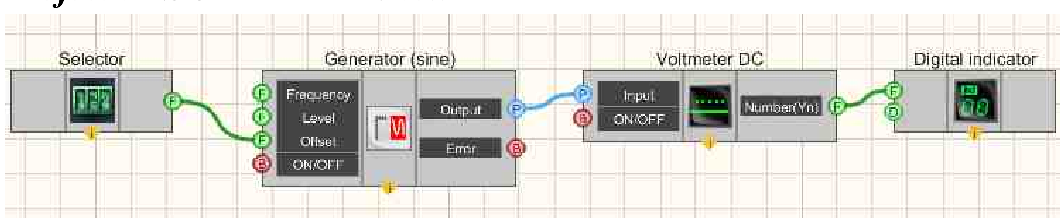
- BSTR Digits - Set the accuracy of the calculation: 1; 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- VARIANT_BOOL Enable - Set Active\Not active level:
  true - the output value is given when the level is active;
  false - the output value is not given when the level is not active.
- LONG InactiveColor - Set the background color of the not active status (any number).
- BSTR ValueType- Set the number display format (string):
  1. Octal;
  2. Decimal;
  3. Hexadecimal.

➢Methods:

- void resetValue(void) - Set reset to zero (any value).

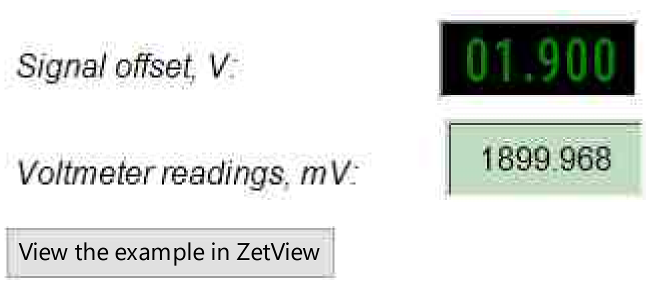
Example

Project in SCADA ZETView



This project shows the operation of the DC Voltmeter 682 component measures the value of the DC component of a signal coming from a Sine signal 682. The Selector 428 is used to set the offset value (the measured DC component). Further, the measured value is sent to the Digital Indicator 682 for graphical presentation of information.

Project operation result



Глава 13.Buttons

Components of "Buttons" menu are often used for projects control. The components of "Buttons" menu are used for control of corresponding components. SCADA ZETView has several types of buttons. Actually, the keys are classified into two groups: fixed and non-fixed. Fixed keys are used for producing a constant logical level signal – "1" or "0". Non-fixed buttons have initial logical level "0". Upon their activation, they produce a single impulse "1" and then go back to initial status "0". Tumbler, fixed button and flag produce "1" value as they are activated and "0" as they are disabled. In most cases they are connected to inputs of "on/off" type. A key with icon sends "1" impulse in the case of activation. In most cases, keys with icons are connected to contacts of "Synch" and "Impulse" contacts.

13.1.CheckBox

The component "**CheckBox**" is used to provide the user with a choice between the discrete statuses True (true) and False (false).

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

Doesn't have.

➢ Output:

- On\Off - the output is a control signal that sets two logical statuses, which depend on whether the flag is set or not.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Check (false) - the component status.
- Name (Check) - the text of the control component.
- Color (c0c0c0) - the background color.
- Transparent (false) - Set the enable/disable transparency. Transparent - transparency of the component. When enabled (true), the background color of the component matches the color of the page; when disabled (false), the background color is set by the color property.
- Enabled (true) - the accessibility of the control.

Note:
When update data from channels, the focus is transferred to the main project window.
The transfer of focus to the main window of the project occurs when the buttons are pressed through the script.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

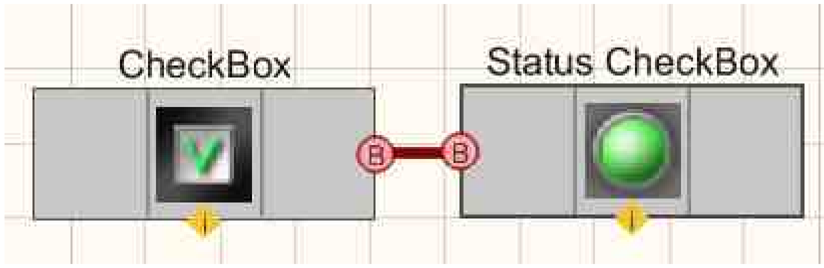
➤**Custom common properties:**
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL Check - Set component status:
  true - enabled status of the component;
  false - disabled status of the component;
- BSTR Name - Set the text of the control component (string).
- LONG Color - Set the background color (number).
- VARIANT_BOOL Transparent - Set the enable/disable transparency:
  true - the output value is given when transparency is enabled;
  false - the output value is not given when transparency is enabled.
- VARIANT_BOOL Enable - Set the accessibility of the control:
  true - the control's accessibility status is enabled;
  false - the control's accessibility status is disabled.

Example

Project in SCADA ZETView



This project shows the operation of the Light indicator 669 component used as a graphical representation of the status **CheckBox**.

Appearance of the Example in the operator interface.

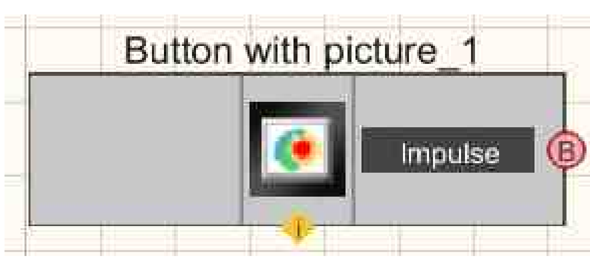
Project operation result

View the example in ZetView

13.2. Button with picture

The component "**Button with picture**" is used to install a button in the project, which, in fact, is a button with a fixation, but instead of an inscription, it has an image.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

Doesn't have.

➢ Output:

- Impulse - a control impulse at the output, on which, when the button is pressed, a short-term changing logic level will appear.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- imagesource (Load from list) - set the image source (Load from list/Load from file).
- image (Clear) - Select of the appearance of the button on the operator panel:
  1. Calculator.
  2. Clock.
  3. Floppy disk.
  4. Document.
  5. Parametrs.
  6. Printer.
  7. Configuration.
  8. Shaker 1.
  9. Shaker 2.
  10. Information.
  11. Timer.
  12. Update.
  13. Select the folder.
  14. Save.
  15. Open.
  16. Forward.
  17. Backward.
  18. Upward.
  19. Start.
  20. Stop.
  21. Clear.

- FileName - set the file name of the uploaded image and location on disk. Allows you to upload your own drawings.
- ToolTip - set the tooltip text.

Note:

When update data from channels, the focus is transferred to the main project window.
The transfer of focus to the main window of the project occurs when the buttons are pressed through the script.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

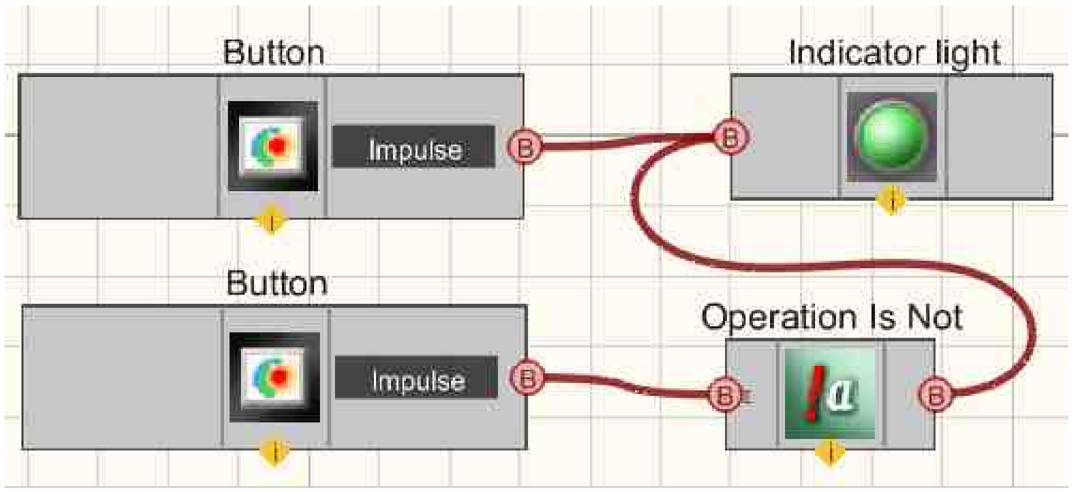
- BSTR ImageSource - Set the source of image loading Load from the list; Load from file (string).
- BSTR Image - Select of the appearance of the button on the operator panel (string):
    1. Calculator.
    2. Clock.
    3. Floppy disk.
    4. Document.
    5. Parametrs.
    6. Printer.
    7. Configuration.
    8. Shaker 1.
    9. Shaker 2.
    10. Information.
    11. Timer.
    12. Update.
    13. Select the folder.
    14. Save.
    15. Open.
    16. Forward.
    17. Backward.
    18. Upward.
    19. Start.
    20. Stop.
    21. Clear.
- BSTR FileName - Set the file name of the uploaded image and location on disk. Allows you to upload your own drawings (string).
- BSTR ToolTip - Set the tooltip text (string).

➤**Methods:**

- Press(void) - Set to press the button (any value).

Example

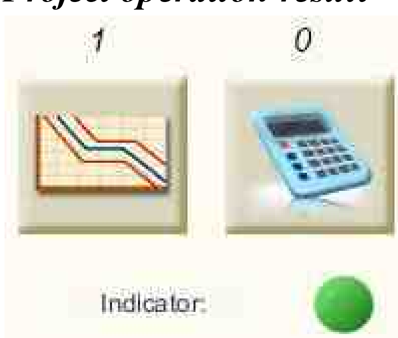
Project in SCADA ZETView



This project shows the operation of the Light indicator 669 component is used for graphical representation of the status of the last pressed Button with picture 687.

Appearance of the Example in the operator interface.

Project operation result



View the example in ZetView

13.3.Fixed button

The component "**Fixed button**" is used to install button that returns to its original status when pressed. When pressed, the button transmits a "1" impulse, then returns to "0".

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

Doesn't have.

➢ Output:

- Impulse - a control signal at the output, on which the logic level changes briefly when pressed (low->high->low).

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- status (false) - the current status of the component. If the parameter is True, then the button will already be pressed when starting the project.
- TextSize (12) - Set the text size.
- textstyle (FontStyleRegular) - Set the text style.
- OnCaption (ON) - Set the caption is ON.
- OffCaption (OFF) - Set the caption is OFF.
- Enabled (true) - Set the enable/disable the component.
- OnBackColor (ON) - Set the color when ON.
- OffBackColor (OFF) - Set the color when OFF.
- AnimationEnabled (false) - Set the animation is Enable/Disable.

Note:

When update data from channels, the focus is transferred to the main project window.

The transfer of focus to the main window of the project occurs when the buttons are pressed through the script.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

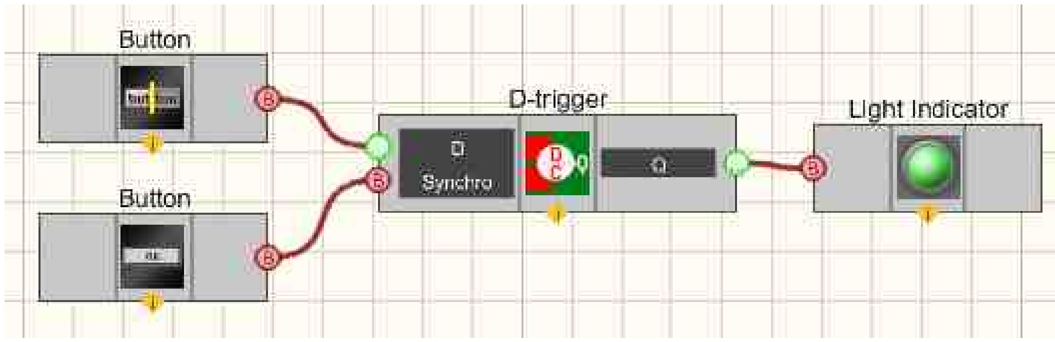
- VARIANT_BOOL status - Set the enable/disable status:
  true - the output value is given when the status is enabled;
  false - the output value is not given when the status is enabled.
- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- BSTR OnCaption - Set the caption is ON (string).
- BSTR OffCaption - Set the caption is OFF (string).
- VARIANT_BOOL Enable - Set the availability of exiting the program:
  true - The availability status is enabled;
  false - The availability status is disabled.
- LONG OnBackColor - Set the color when ON. (any number).
- LONG OffBackColor - Set the color when OFF (any number).
- VARIANT_BOOL AnimationEnable - Set the animation is enable/disable.
  true - the output value is given when the animation is enabled;
  false - the output value is not given when the animation is enabled.

➢**Methods:**

- void Press(LONG newVal) - Set button press (any value).
  newVal - button click value (any value).

Example

Project in SCADA ZETView



This project shows the operation of the Fixed button 691 component is used to set the binary status that will be transferred to the Light indicator 669 using a D-trigger. A Non-fixed button 694 is needed to send a clock impulse to the clocked input of the trigger (so that it transfers its status from input to output). The Light indicator 669 is needed for graphical display of discrete statuses.

The appearance of the Example in the operator interface.

Project operation result

View the example in ZetView

13.4.Non-fixed button

The component "**Non-fixed button**" is used to install button that returns to its original status when pressed. When pressed, the button transmits a "1" pulse, then returns to "0".

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

Doesn't have.

➢ Output:

- Impulse - a control signal at the output, on which the logic level changes briefly when pressed (low->high->low).

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- TextSize (12) - Set the text size.
- textstyle (FontStyleRegular) - Set the text style.
- Enabled (true) - Set the enable/disable the component.
- OnCaption (Button) - Set the caption is ON.
- BtnColor (46d6e1) - Set the color of the button in the running status of the project, when hovering over it with the mouse.
- AnimationEnabled (true) - Set the animation is enable/disable.

Note:

When update data from channels, the focus is transferred to the main project window.

The transfer of focus to the main window of the project occurs when the buttons are pressed through the script.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

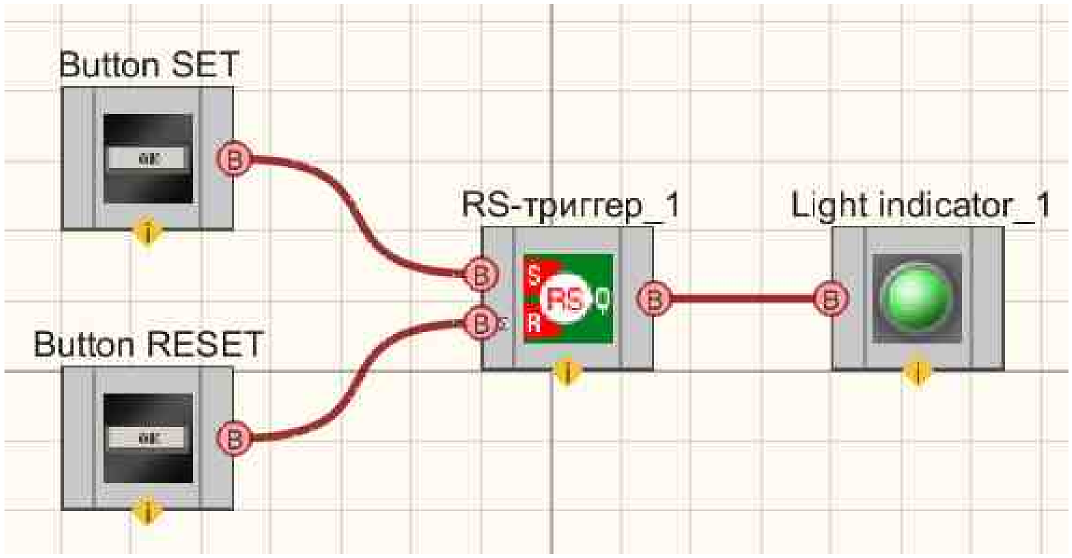
- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set the text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- VARIANT_BOOL Enable - Set the availability of exiting the program:
  true - The availability status is enabled;
  false - The availability status is disabled.
- LONG OnBackColor - Set the color when ON. (any number).
- LONG BtnColor - Set the color of the button in the running status of the project, when hovering over it with the mouse.
- VARIANT_BOOL AnimationEnable - Set the animation is enable/disable.
  true - the output value is given when the animation is enabled;
  false - the output value is not given when the animation is enabled.

➢**Methods:**

- void press(void) - Set button press (any value).

Example

Project in SCADA ZETView



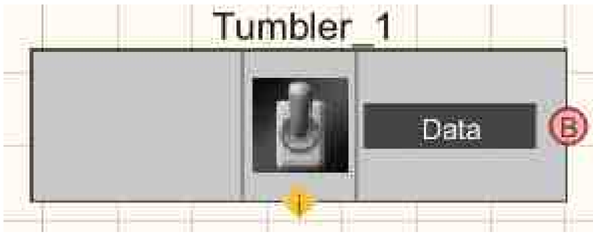
This project shows the operation of the Non-fixed button 694 components are used to change statuses at the inputs of the RS-trigger. The Light indicator 669 component is used to display the status of the trigger output. When the Non-fixed button 694 are pressed in succession without fixation, the status of the Light indicator 669 changes.

Appearance of the Example in the operator interface.

13.5.Tumbler

The component "**Tumbler**" is intended for installation in a project of a Tumbler type switch with two positions, which correspond to high and low logic levels.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

Doesn't have.

➢ Output:

- Data - the output value of the Tumbler switch, on which two different logical statuses are set, which depend on the position of the Tumbler switch handle.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Color (505050) - Set the color of the Tumbler switch handle.
- Status (false) - Set the Tumbler switch position (enabled (true)/disabled (false)).

Note:

When update data from channels, the focus is transferred to the main project window.

The transfer of focus to the main window of the project occurs when the buttons are pressed through the script.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

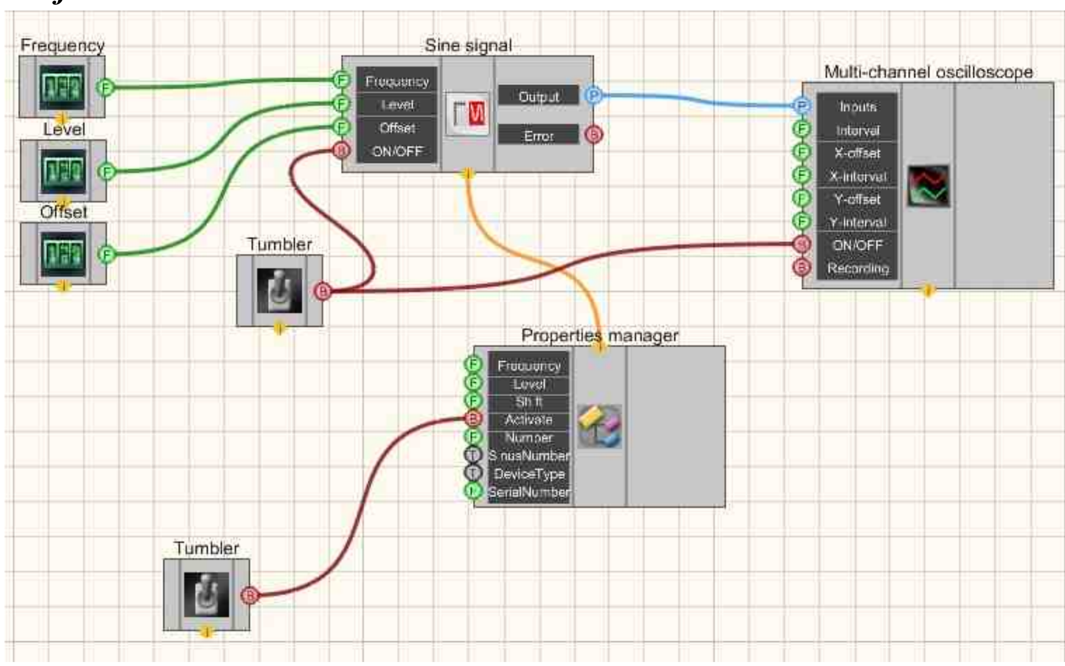
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG Color - Set the color of the Tumbler switch handle (number).
- VARIANT_BOOL Status - Set the position of the Tumbler switch:
  - true – the Tumbler switch is on (up position);
  - false - the Tumbler switch is off (down position).

Example

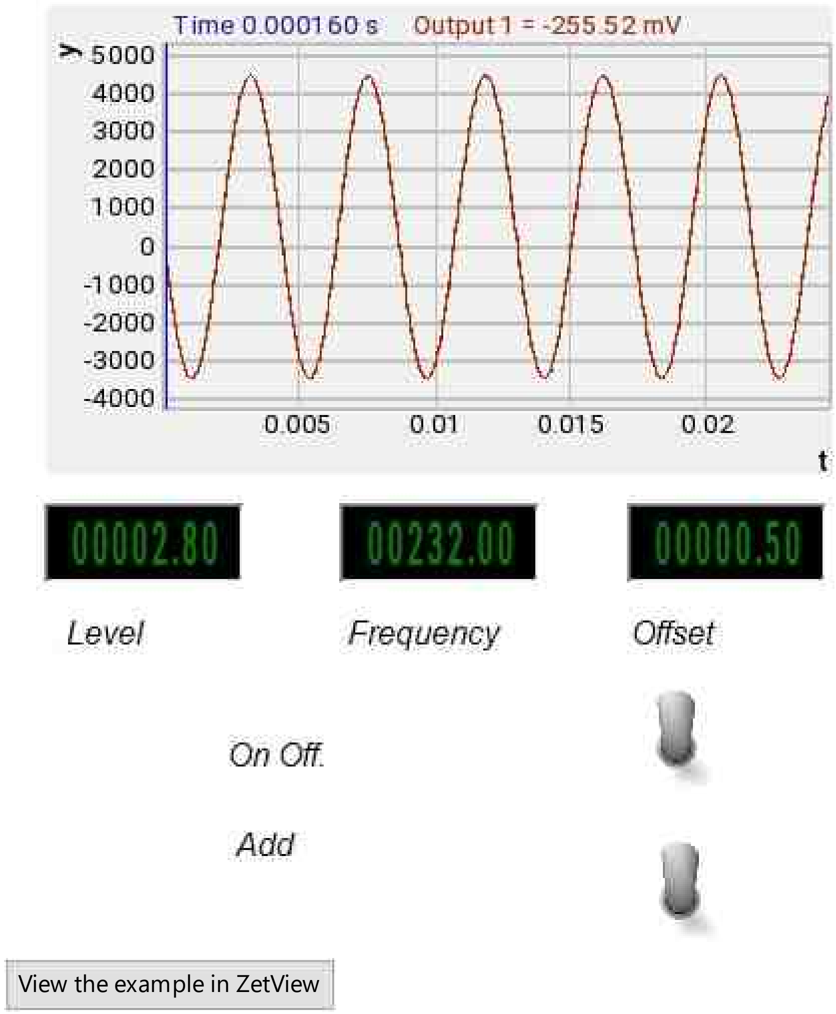
Project in SCADA ZETView



This project shows the operation of the Sine signal 696 component creates a Sine signal, with parameters that are set using the Selectors 428. The Tumbler 696 allows you to turn on and off the Multi-channel oscilloscope 928 with the Sine signal Signals generator 530. To view the generated signal, use the Multi-channel oscilloscope 928 component. The result of the compiled program can be seen in the operator interface. The Tumblers 696 is used to turn on or off. The Property manager 999 is used to change the indicators in real time and transmit data to the signal generator to display them on a Multi-channel oscilloscope.

Appearance of the Example in the operator interface.

Project operation result



Глава 14.Conversion

As you can see from the previous chapter, SCADA ZETView has various types of data. In most cases, each component receives and produces a particular data type. Sometimes it is necessary to convert data. A typical example of such conversion is representation of digital data in tabular form. This component receives textual data to its input contact, thus, in order to represent it in tabular form, it is necessary to display it in textual form. Components of "Converting" section are normally used for such purposes. This section contains all necessary components for converting all data types in SCADA ZETView system. Components of "Converting" section are not displayed in operator mode, they can only be added to the project in developer mode.

14.1.Converter (logic - string)

The component **"Converter (logic - string)"** is intended for data transfer from Boolean type to text messages.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - input channel to which boolean type data is received.
- ON/OFF - enabling/disabling the component (duplicating the **Activate** property).

➢ Output:
- Output - output channel, which, depending on the logical level received at the input, generates a text message of interpretation of the value at the output.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- TrueString (Logical 1) – Set a message that will be displayed if a logical unit is received at the input.
- FalseString(Logical 0) – Set a message that will be displayed if a logical zero is received at the input.
- RepeatData (true) - Set the enable/disable for repeat message.
- Enabled (true) - Set the active/not active status of the component.
- InactiveText (Component in not active status) – Set the message that will be displayed when the transition to the not active status.
- ActiveText (Component in the active status) – Set the message that will be displayed when the transition to the active status.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

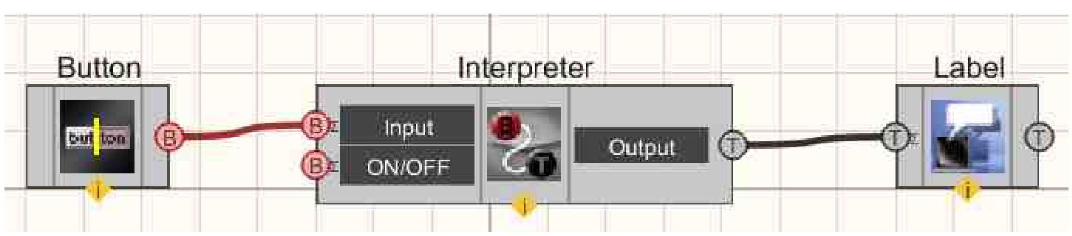
•BSTR TrueString - Set the string corresponding to a logical one (string).
•BSTR FalseString - Set the string corresponding to logical zero (string).
VARIANT_BOOL RepeatData - Set the response to a repeated message:
- true - Reaction to repeated message is enabled;
false - Reaction to repeated message is disabled.

VARIANT_BOOL Enable - Set the enable/disable component:
- true - the output value is given when the component is enabled;
false - the output value is not given when the component is enabled.

•BSTR InactiveText - Set the text on transition to the not active status (string).
•BSTR ActiveText - Set text on transition to active status (string).

Example

Project in SCADA ZETView



This project shows the operation of the Converter (logic - string) 700, depending on the input logical level, transmits the specified text message at the output. A Non-fixed button 694 connected to the Converter (logic - string) 700 The Label 1087 is required for the graphical display of messages.

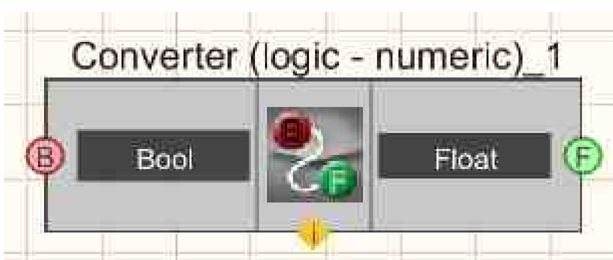
Project operation result

View the example in ZetView

14.2.Converter (logic - numeric)

The component "**Converter (logic - numeric)**" is intended for data transfer from Boolean to numeric type.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Bool - the input channel that receives boolean data.

➤ Output:

- Float - returns a number corresponding to the input value.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- TrueConvertion (1) - Set the number corresponding to true (one in Boolean algebra).
- FalseConvertion (0) - Set the number corresponding to a false value (zero in boolean algebra).
- RepeatData (true) - Set the response to a repeated message (true/false).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- <u>Are available by the link below.</u> 154

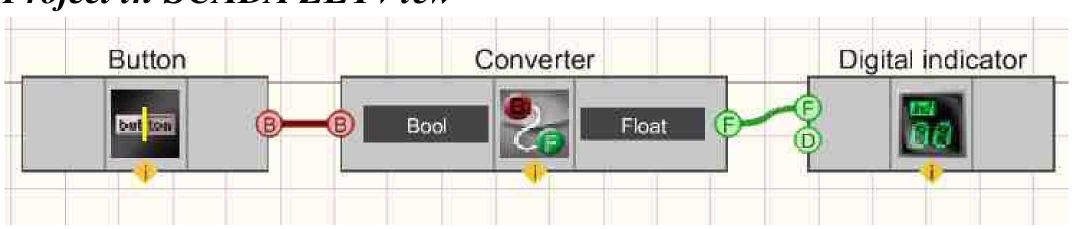
➢**Custom private properties (the default value is shown in parentheses):**

•FLOAT TrueConvertion - Set the number corresponding to True (number).
•FLOAT FalseConvertion - Set the number corresponding to False (number).
VARIANT_BOOL RepeatData - Set the response to a repeated message:

- true - Reaction to repeated message is enabled;
  false - Reaction to repeated message is disabled.

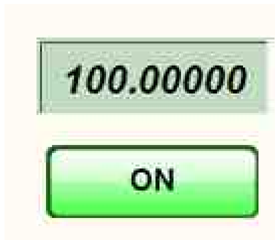
Example

Project in SCADA ZETView



This project shows the operation of the <u>Converter (logic - numeric)</u> 702 translates the statuses of the <u>Fixed button</u> 691 (transmits binary statuses) into real values specified by the user in the settings, in this Example it is 100 when the logic level is high at the input and -250 when it is low. The Digital indicator 682 is used for graphical representation of numerical values.

Project operation result



View the example in ZetView

14.3.Converter (string array - string)

The component **"Converter (string array - string)"** is intended for translation of text arrays in a line.

When non-convertible characters arrive at the input, the component sends a convertible string to the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- {Y} - Input string array.

➢ Output:

- Text - The output string to be converted.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Separator - set the separator character.
- UseSpecSeparator (false) - Set the use a special separator symbol.
- SpecSeparator(Space) - special separator symbol: Space; Tabulation; Line break.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- {Y} - value of the input string array (string).

➢Custom common properties:

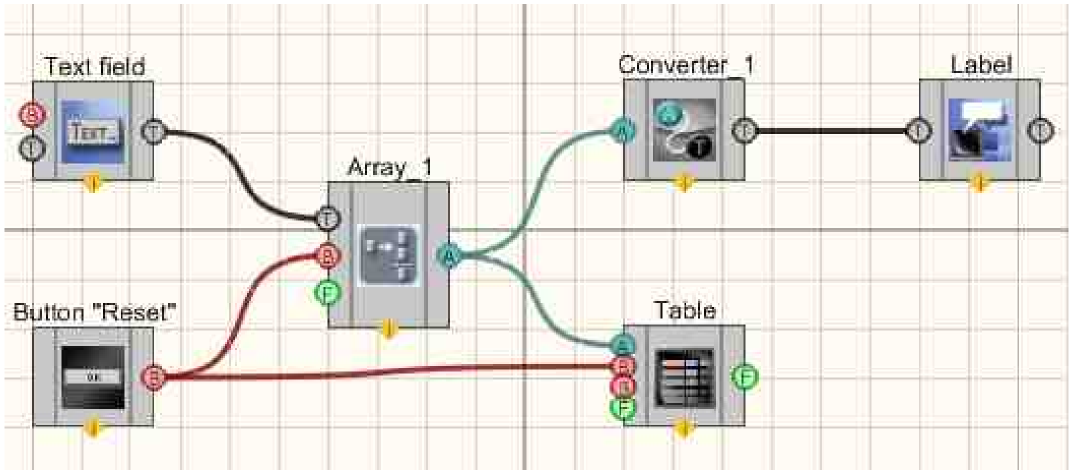
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

•BSTR Separator - Set the separator symbol (any string).
VARIANT_BOOL UseSpecSeparator - Set the use of a special separator symbol:
•true - The status of using the special delimiter symbol is enabled;
false - The status of using the special delimiter symbol is disabled.
•BSTR SpecSeparator - Set a special separator symbol: Space; Tabulation; Line break (any line).

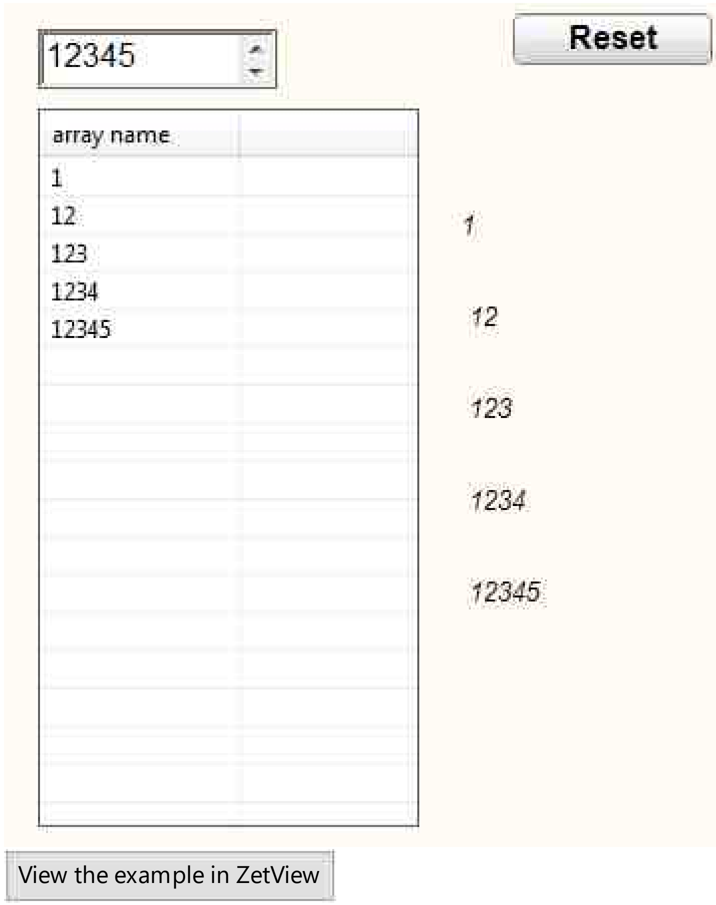
Example No. 1

Project in SCADA ZETView



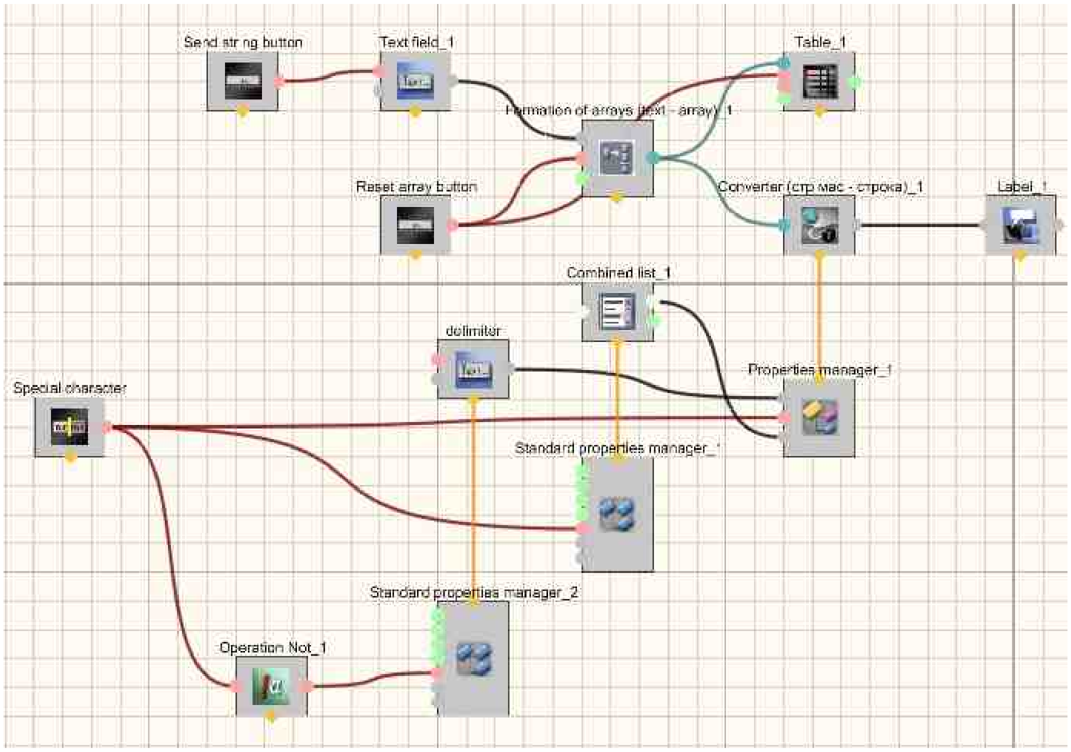
The diagram below shows the operation of the Converter (string array - string) 704 component. The string array at the input of the component is formed in the following way: the user enters numbers into the Text field 1100, which are sent to the Formation of arrays (text - array) 793, then the array is output to the Data table 934 and gets to the Converter (string array - string) 704. The received data is displayed on the label. The Non-fixed button 694 is used in this project to reset the array and values in the table.

Project operation result



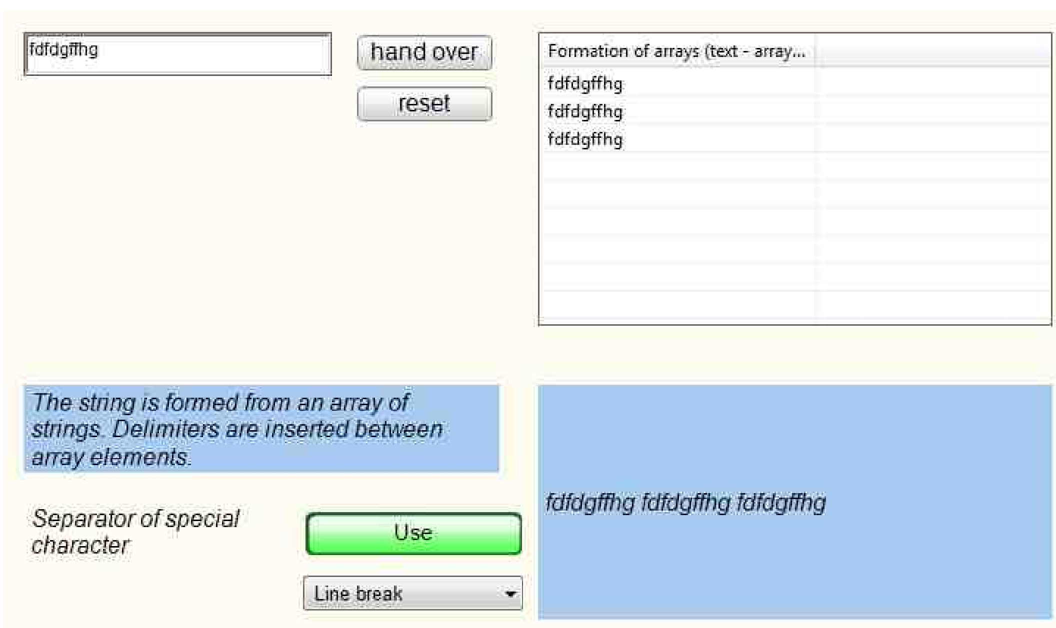
Example No. 2

Project in SCADA ZETView



The presented scheme shows the operation of the Converter (string array - string) 704, component, which allows you to insert a newline character.

Project operation result



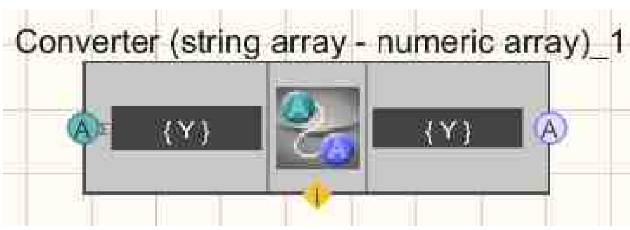
View the example in ZetView

14.4.Converter (string array - numeric array)

The component "**Converter (string array - numeric array)**" is intended for translation of textual arrays into numerical arrays.

When non-convertible characters arrive at the input, the component passes them to the output as a number of 0.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- {Y} - Input string array.

➢ Output:
- {Y} - Output numeric array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

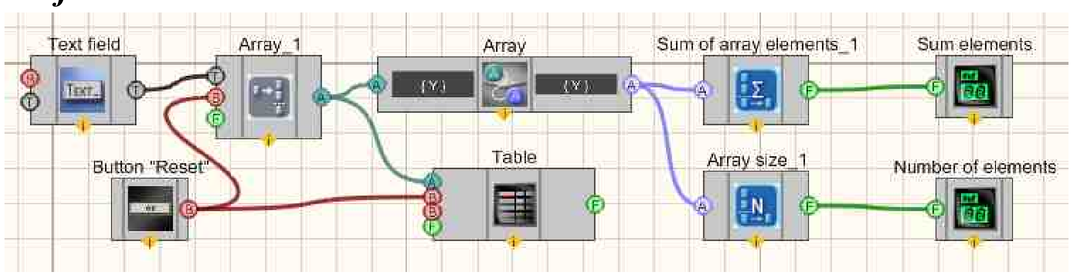
- {Y} - value of the input string array (string).

➢**Custom common properties:**

- Are available by the link below. 154

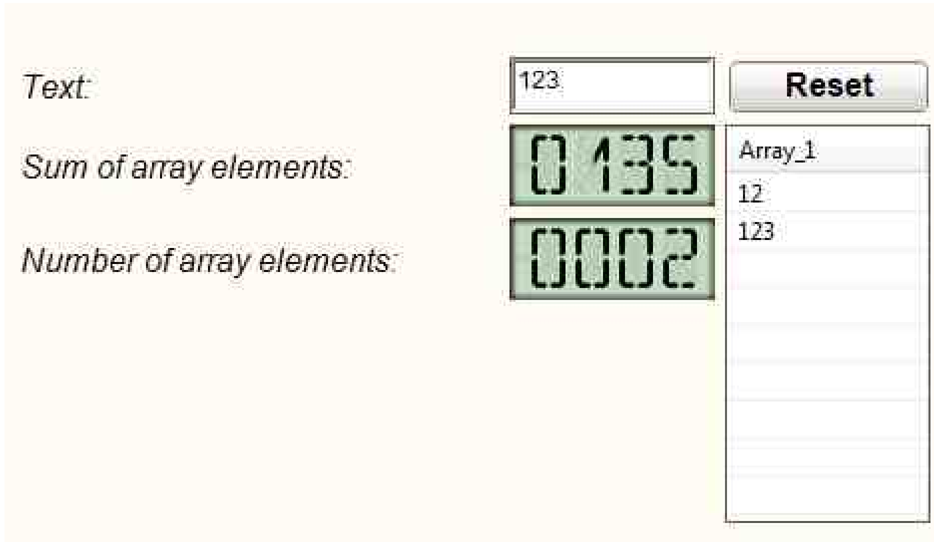
Example

Project in SCADA ZETView



The presented scheme shows the operation of the Converter (string array - numeric array) 704 component. The string array at the input of the component is formed in the following way: in the Text field 1100, the user enters the numbers that come to the Formation of arrays (text - array) 793, then the array is output to the Data table 934 and gets to the Converter (string array - numeric array) 704, where from string array is converted to numeric. Further, all elements of the resulting numeric array are added using the Sum of array elements 830 component and the result is displayed on the Digital indicator 682. The Non-fixed button 694 is used in this project to reset the array and values in the table.

Project operation result



View the example in ZetView

14.5.Converter (string - numeric)

The component "**Converter (string - numeric)**" is designed to translate numbers represented in a text type to a numeric value.
When non-convertible characters arrive at the input, the component passes them to the output as a number of 0.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- String - text data is being received.

➢ Output:

- Number - passes text values converted to a numeric type.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- String - the value of the input text data (text).

➢**Custom common properties:**

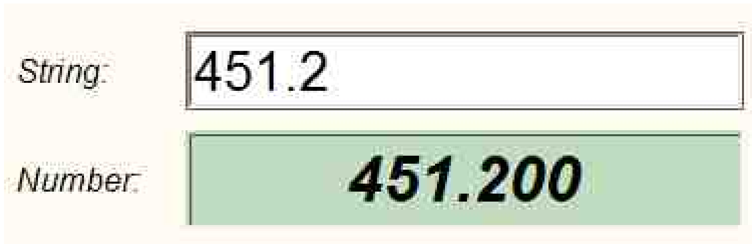
- Are available by the link below. 154

Example

Project in SCADA ZETView

This project shows the operation of the Converter (string - numeric) 711 component converts values presented in text form to a numeric type. The Digital indicator 682 is used for graphical presentation of information. The Text field 1100 is intended for entering numbers in text form.

Project operation result

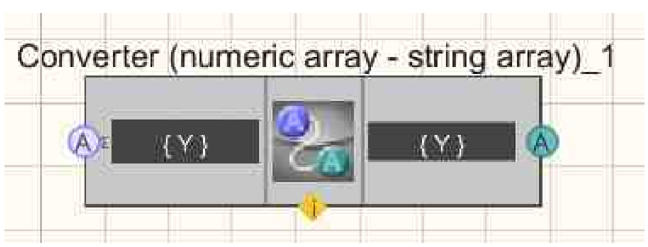


View the example in ZetView

14.6.Converter (numeric array - string array)

The component "**Converter (numeric array - string array)**" is intended for transfer of numerical arrays to text.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- {Y} - Input numeric array.

➤ Output:

- {Y} - Output string array.

Custom common properties:

➤Total properties (environment):

Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- precision (0) - Set the allows display precision.
- type (Text) - Set the conversion type is set (Text/Symbol).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - value of the input numeric array (number).

➢**Custom common properties:**

- Are available by the link below. 154

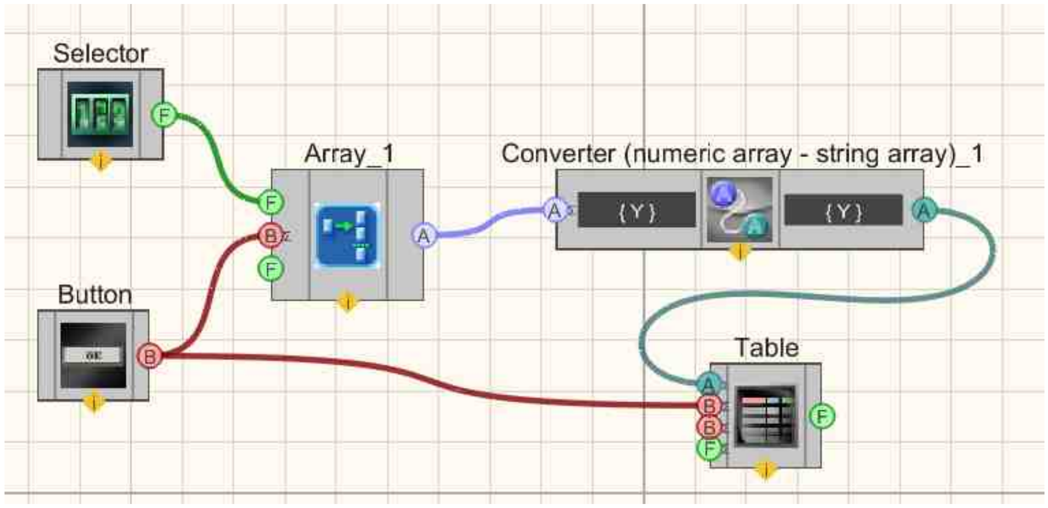
➢**Custom private properties (the default value is shown in parentheses):**

•LONG Precision - Set the number of decimal places (conversion precision) (from 0 to 6).
•BSTR Type - Set the conversion type: Text; Symbol (string).

Example

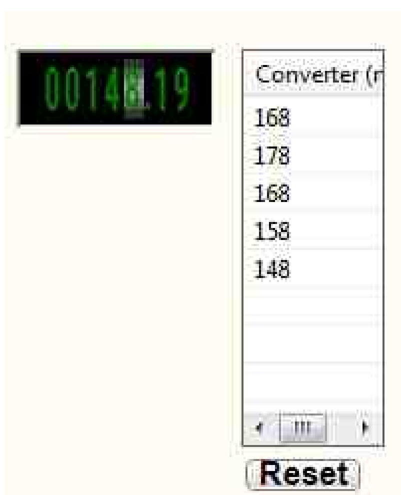
Project in SCADA ZETView

Most often, this component is used to display results in a table:



In this scheme show the operation of the component Converter (numeric array - string array) 713. Numbers from the selector get into the Formation of arrays (numeric - array) 796, after which the Converter (numeric array - string array) 713 component converts the numeric array into a text one, which allows you to output the result to the Data table 934. The Non-fixed button 694 is used in this project to reset the array and values in the table.

Project operation result

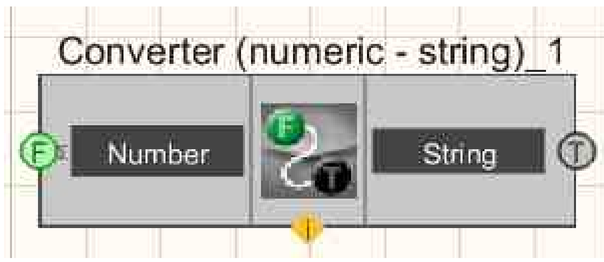


View the example in ZetView

14.7.Converter (numeric - string)

The component "**Converter (numeric -** string)" is intended for data transfer from a numerical type to a string type.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Number - Numeric type data is being received.

➢ Output:
- String - passes numeric values converted to string type.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- type (Decimal) - Set the type of conversion (there is a conversion from one number system to another before converting to a string type):
    1. Octal.
    2. Decimal.
    3. Hexadecimal.
- RepeatData (true) - Set the response to a repeated message. If the value is True, the component will convert all messages coming to the input, if it is False, then when a repeated message is received, the component will ignore it, it only works for consecutive messages:

10,12,10,13,14 will be converted as 10,12,10,13,14
10,12,10,10,14 will be converted as 10,12,10,14
- Precision (6) - Set the number of decimal places.
- ResultStringLength (0) - Set the maximum string length limit after conversion.
- FillLeadingZeros (false) - Set to pad with leading zeros.
- LimitStringLength (false) - Set the Enable/disable string length limit after conversion.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**
- Number - the value of the input numeric array (number).

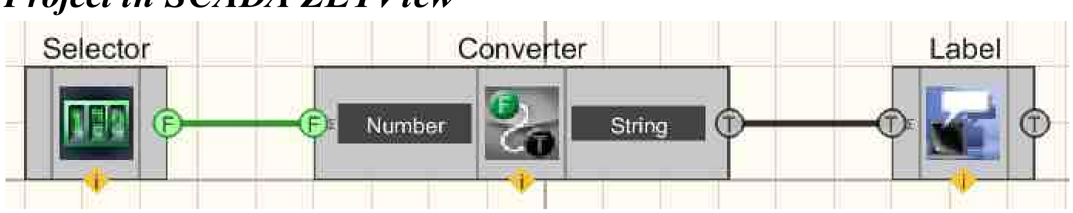
➤**Custom common properties:**
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR Type - Set conversion type: Octal; Decimal; Hexadecimal (string).
- VARIANT_BOOL RepeatData - Set the response to a repeated message:
  true - the reaction to repeated message is enabled;
  false - the reaction to repeated message is disabled.
- LONG Precision - Set the number of decimal places (number).
- LONG ResultStringLength - Set the maximum string length limit after conversion (number).
- VARIANT_BOOL FillLeadingZeros - Set to fill with leading zeros:
  true - do fill with leading zeros;
  false - do not fill with leading zeros.
- VARIANT_BOOL LimitStringLength - Enable/disable string length limit after conversion:
  true - the reaction to the inclusion of a string length limit after conversion;
  false - the response to disabling the string length limit after conversion.

Example

Project in SCADA ZETView



This project shows the operation of the Converter (numeric - string)[716] component converts the numeric values received from Selectors[428] to a string type. Label[1087] is used for graphic display of received textual information.

Project operation result

View the example in ZetView

14.8.Converter (DATE - string)

The component "**Converter (DATE - string)**" is used to convert time in DATE format to a string.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢ Input:
- DATE - type input double - input time in DATE format.

➢ Output:
- String - output contact type string - output contact - string.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Precision (0.010000) - The transformation is performed with the precision specified by the Precision property.

The value of this property can be from 0.000001 to 1 second. The default accuracy is 0.01 s (sampling frequency 100 Hz, as for SMSE).

- NeedData (true) - determines if the output string contains a date.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- DATE - input type value double (number).

➢Custom common properties:

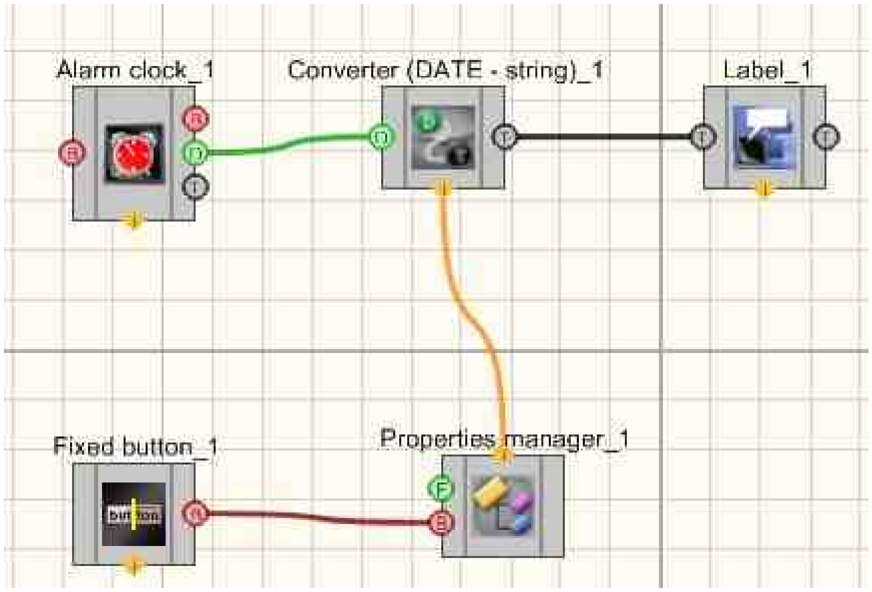
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

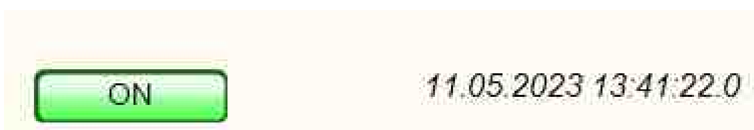
- FLOAT Precision - Set the conversion precision, s (double number).
- VARIANT_BOOL NeedData - Set the output of the date in the string:
  true - the reading date output in a string is enabled;
  false - the reading date output in a string is disabled.

Example

Project in SCADA ZETView



Project operation result

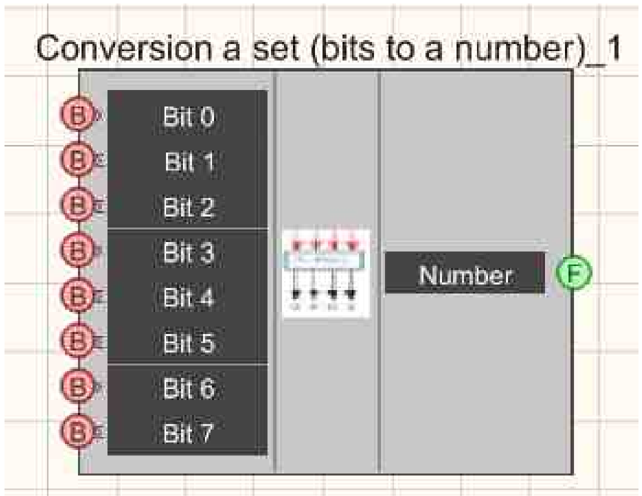


View the example in ZetView

14.9.Conversion a set (bits to a number)

The component "**Conversion a set (bits to a number)**" is intended for converting a set of bits into a number.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Bit 1 - a bit output that receives 1 output bit.
- ...
- Bit 7 - the bit output that receives the 8th output bit.

➢ Output:

- Number - the input value on which depending on the input bit set.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Quantity (8) - set the number of bits is determined: 8; 16; 32 (string).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

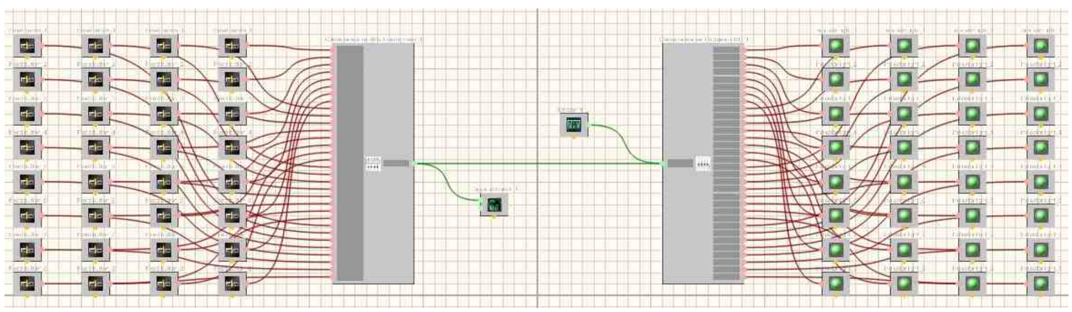
- [Are available by the link below.](154)

➢**Custom private properties (the default value is shown in parentheses):**

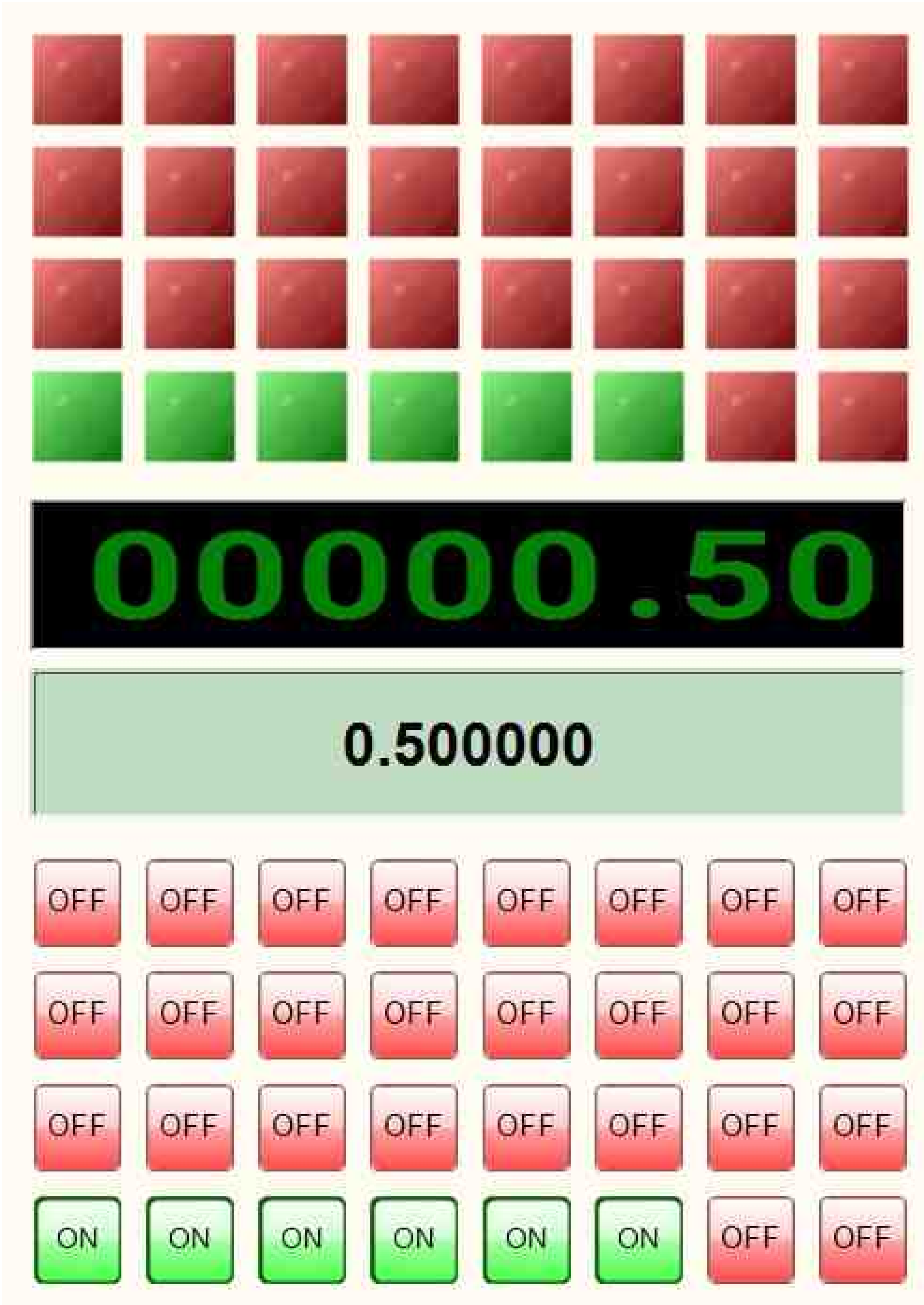
•BSTR Quantity - Set the number of bits: 8; 16; 32 (string).

Example

Project in SCADA ZETView



Project operation result



View the example in ZetView

14.10.Conversion of text to speech

The component " **Conversion of text to speech**" is used for scoring of the text information.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - text that will later be converted to speech.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- SpeechText - Set the enter the text to be spoken.
- SpeechEngine - specifies the speech module that will be used for dubbing (to connect a speech module, you need to select a component, then call the context menu, select Edit component. A list of speech modules installed in the system will appear, you must select one of them).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

· Input - the value of the input text data (text).

➤**Custom common properties:**

- Are available by the link below. 154

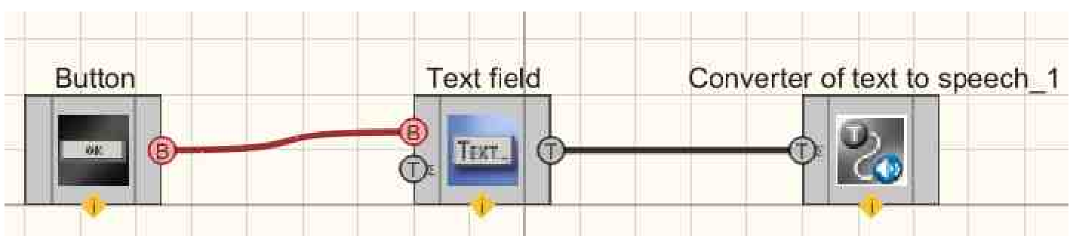
➤**Custom private properties (the default value is shown in parentheses):**

➤

·BSTR SpeechText - Set the speech text (string).
·BSTR SpeechEngine - Set the speech module (string).

Example

Project in SCADA ZETView



This project shows the operation of the Conversion of text to speech 724 component is used to convert text messages into sound messages. The Text field 1100 is required to enter text into the component.

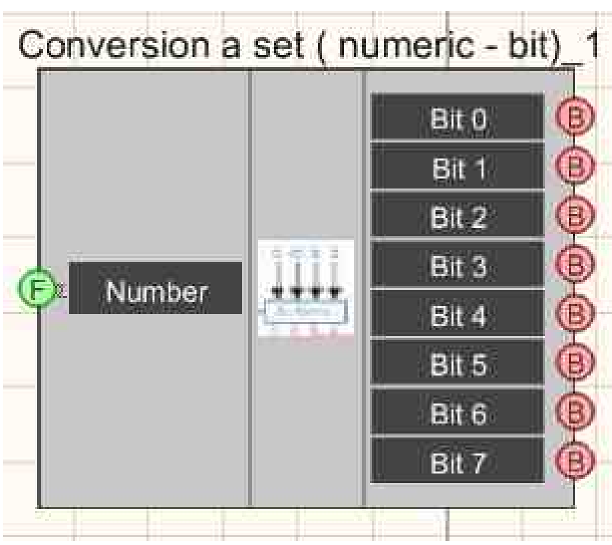
Project operation result

View the example in ZetView

14.11.Conversion a set (numeric - bit)

The component "**Conversion a set (numeric - bit)**" is designed to convert a number into a set of bits.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Number - the input value on which depending on the input bit set.

➢ Output:
- Bit 1 - a bit output that receives 1 output bit.
- ...
- Bit 8 - the bit output that receives the 8th output bit.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Quantity (8) - Set the number of bits is determined: 8; 16; 32 (string).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Number - the input value on which depending on the input bit set.

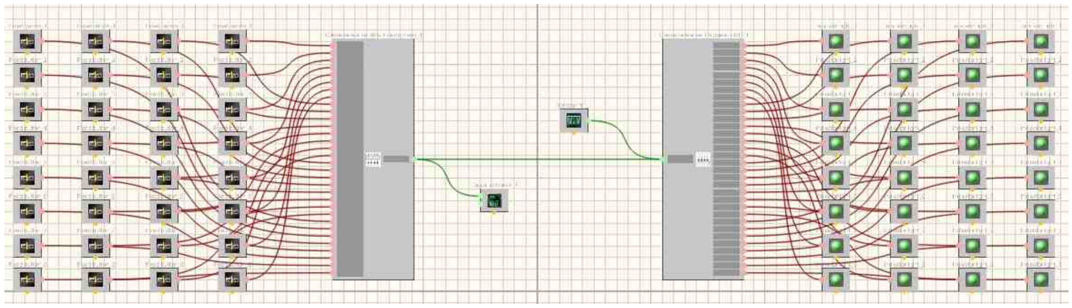
➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

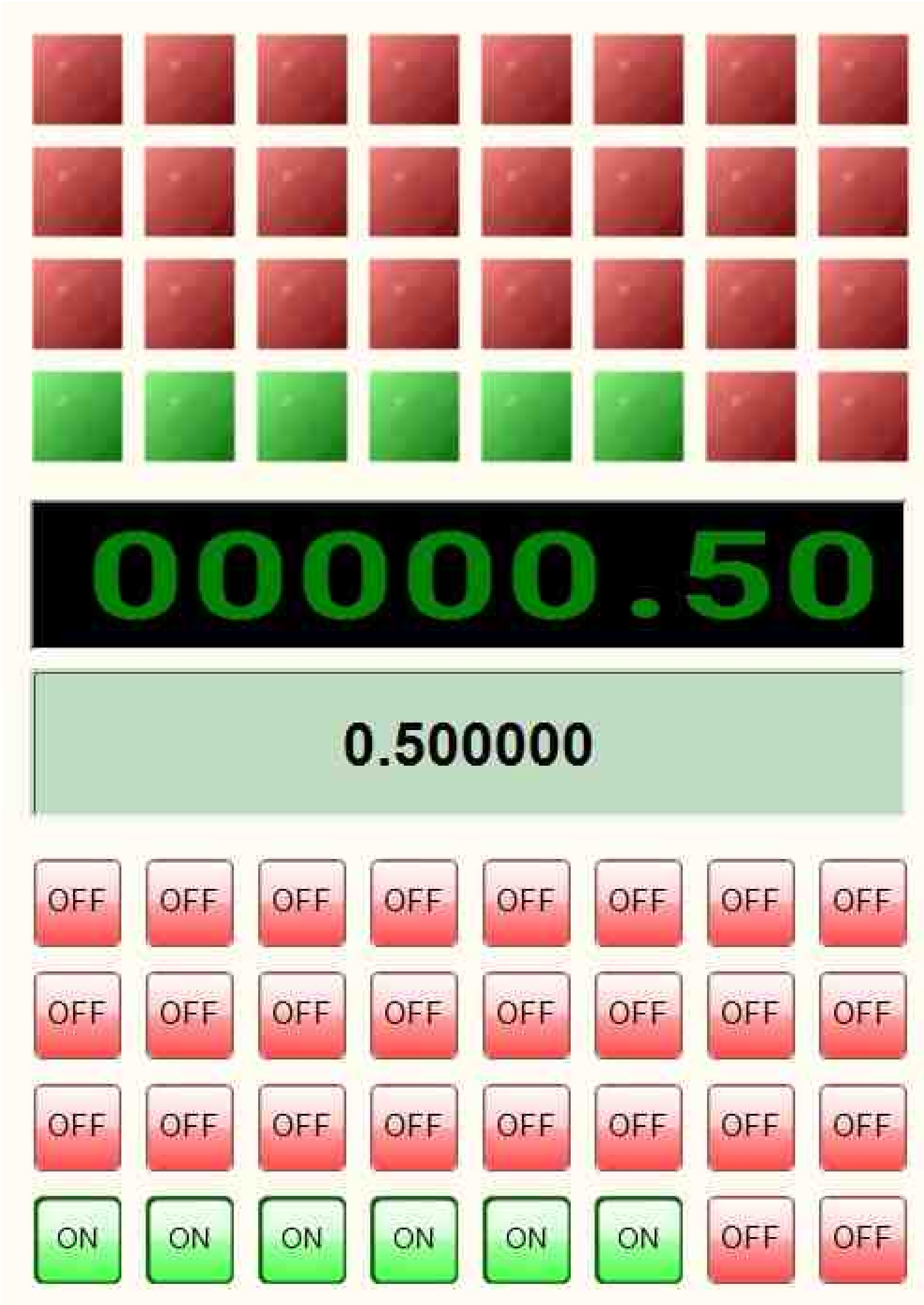
•BSTR Quantity - Set the number of bits: 8; 16; 32 (string).

Example

Project in SCADA ZETView



Project operation result



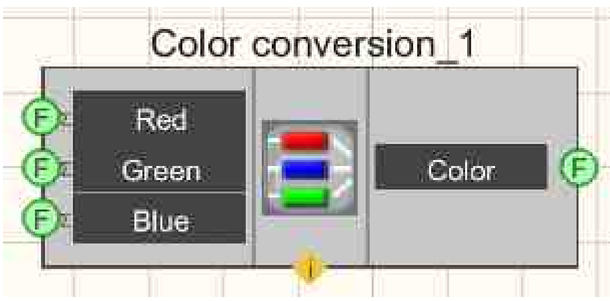
View the example in ZetView

14.12.Color conversion

The component "**Color conversion**" is intended for transfer of the received numerical values of parameters RGB in RGB-code.

RGB (abbreviation of English words Red, Green, Blue - red, green, blue) - an additive color model, usually describing the way of color synthesis for color reproduction, each color is represented as a combination of red (R - red), green (G - green) and blue (B - blue) colors in certain proportions.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Red - the ntensity of the red component of the color (from 0 to 255).
- Green - the intensity of the green component of the color (from 0 to 255).
- Blue - the intensity of the blue component of the color (from 0 to 255).

➢ Output:
- Color - the output color value of the RGB color model. The result is presented as a code of 6 digits in decimal notation.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Red (0) - Set the value of the red color parameter at the input of the component by default.
- Green (0) - Set the value of the green color parameter at the input of the component by default.
- Blue (0) - Set the value of the blue color parameter on the input of the component by default.
-

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Red - the intensity value of the red component of the color (from 0 to 255).
- Green - the intensity value of the green component of the color (from 0 to 255).
- Blue - the intensity value of the blue component of the color (from 0 to 255).

➤**Custom common properties:**

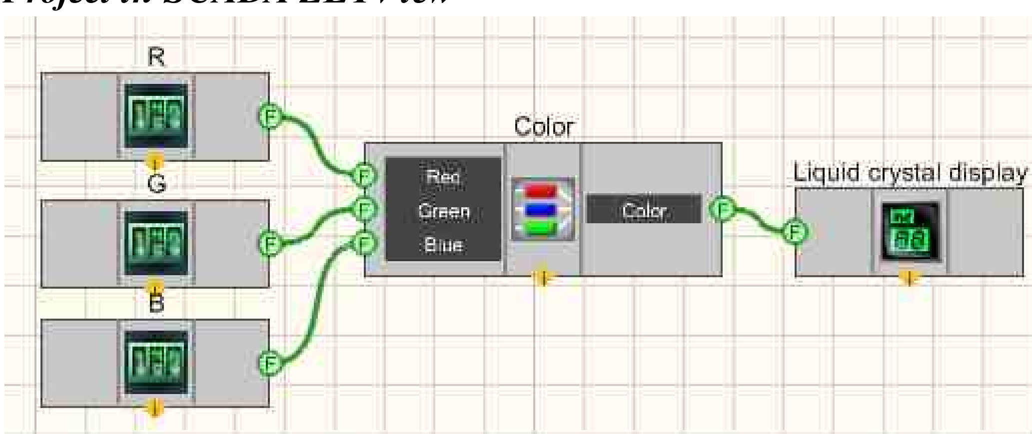
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

•LONG Red - Set the intensity of the red component of the color (from 0 to 255).
•LONG Green - Set the intensity of the green component of the color (from 0 to 255).
•LONG Blue - Set the intensity of the blue component of the color (from 0 to 255).

Example

Project in SCADA ZETView



Three Selectors 428 are used to set the RGB inputs. The result is displayed on Liquid crystal display 655.

Project operation result

Глава 15.Logic

"Logic" section contains components implementing such logical operations as "Operation AND", "Operation AND (multiple)", "Operation OR", "Operation OR (multiple)", "Overlay mask" (the difference is that these components can receive two ore more input values). One more component is D-trigger, allowing to transfer data when a impulse is received at the input of "Synchro" component. RS-trigger transfers logical variable depending on input combination. Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

15.1.D-trigger

The component **"D-trigger"**, allowing to transfer data when a impulse is received at the input of "Synchro" component.

Designed for use as a clocked D-trigger. The input D receives data of any type, but is not transmitted to the next component, while at the input B of the trigger is set to 0 (the default). At the moment when the D-trigger receives control impulse 1 at input B, it transmits the last value obtained to the next component. After that, the input B is set to 0 again.

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

- Input:
  - D - data information channel.
  - Sync - synchronization input, data from pin D will be transferred to output Q with boolean values.

- Output:
  - Q - data is received from the input.

Custom common properties:

- Total properties (environment):
  - Are available by the link below. 154

- **Custom private properties (the default value is shown in parentheses):**

  Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- D - Set data value (any type).

➢**Custom common properties**:

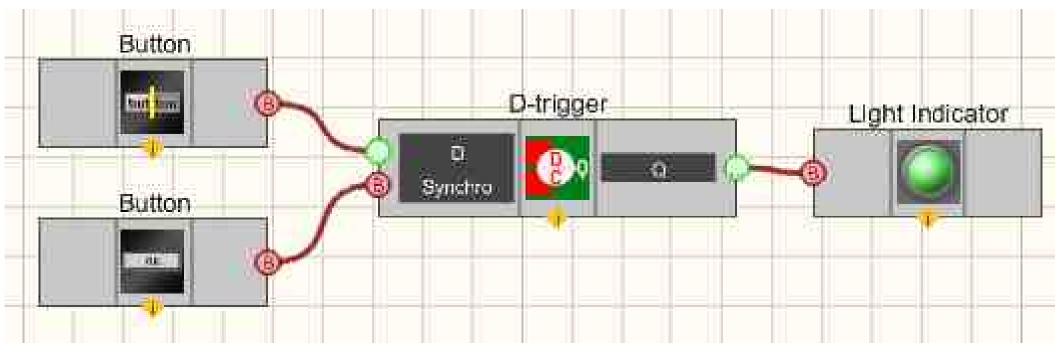
- Are available by the link below. 154

➢**Methods:**

- void Synchro(void) - Set the sent values.

Example

Project in SCADA ZETView



This project shows the operation of the Fixed button 691 component is used to set the binary status that will be transferred to the Light indicator 669 using a D-trigger. A Non-fixed button 694 is needed to send a clock impulse to the clocked input of the trigger (so that it transfers its status from input to output). The Light indicator 669 is needed for graphical display of discrete statuses.

The appearance of the Example in the operator interface.

Project operation result

View the example in ZetView

Mathematical description

A trigger is an electrical circuit that has two stable states, which are established when an appropriate combination of signals is applied to the control inputs of the trigger and persists for a specified time after the end of these signals. A trigger is a logical device capable of storing 1 bit of data.

In the D-trigger, with the help of additional logical elements, it was possible to avoid the forbidden situation when the status of the trigger is indeterminate. The truth table in this case looks like:

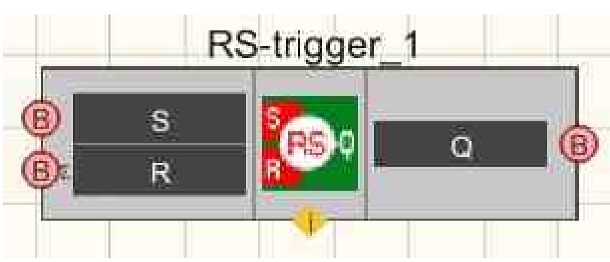
CLK	D	Q	Q	
0	X	Q-1	Q-1	Information storage mode
1	0	0	1	Information recording mode
1	1	1	0	

15.2.RS-trigger

The component **"RS-trigger"**, transfers logical variable depending on input combination.

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- S - Set, when applied to which the impulse sets the output to a high level with logical values 1.
- R - Reset, when applied to which the impulse sets the output to a low level with logical values 0.
-

➢ Output:

- Q - depending on the input values, a certain logical level with boolean values is set.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

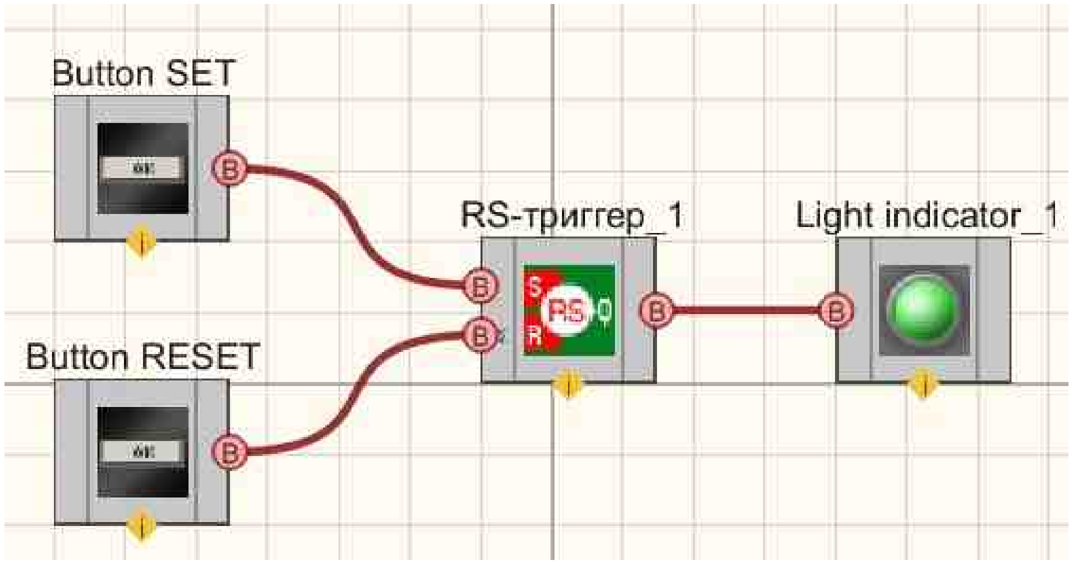
Setting:

➢**Custom common properties**:

- Are available by the link below. 154

Example

Project in SCADA ZETView



This project shows the operation of the Non-fixed button[694] components are used to change statuses at the inputs of the RS-trigger. The Light indicator[669] component is used to display the status of the trigger output. When the Non-fixed button[694] are pressed in succession without fixation, the status of the Light indicator[669] changes.

Appearance of the Example in the operator interface.

Project operation result

View the example in ZetView

Mathematical description

A trigger is an electrical circuit that has two stable states, which are established when an appropriate combination of signals is applied to the control inputs of the trigger and persists for a specified time after the end of these signals. A trigger is a logical device capable of storing 1 bit of data.

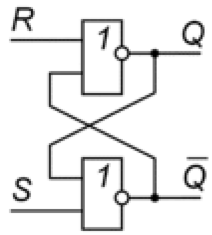


Fig. 1

The RS-trigger is implemented on the basis of two cross-feedback inverters. The RS-trigger, shown in Fig. 1, consists of two NOR elements. The trigger inputs are labeled S (Set) and R (Reset). His work is described by the following truth table:

S	R	Q	¬Q
0	0	Q_{-1}	$\neg Q_{-1}$
0	1	0	1
1	0	1	0
1	1	0	0

Q_{-1} - previous status

When S=0, R=0, the status of the trigger does not change, this mode is called the information storage mode. When S=R=1, both trigger outputs must be equal to zero, which is logically impossible, therefore, therefore, such a status is called undefined or forbidden, and such situations should be avoided. The S=1, R=0 mode is the logical one writing mode (set mode), and the S=0, R=1 mode is the logical zero mode (reset mode).

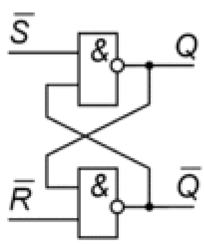


Fig. 2

Fig. 2 shows an RS-trigger with inverted inputs, whose status are switched by low-level signals. For such a trigger, the truth table looks like this:

¬S	¬R	Q	¬Q

0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q_{-1}	$\neg Q_{-1}$

Q_{-1} - previous status

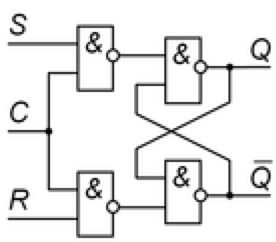


Fig. 3

Fig. 3 shows a modification of the RS-trigger, called a controlled or synchronous RS-trigger. Such a flip-flop changes its status only when a high level signal is applied to the CLK clock input. When this input is driven low, the flip-flop retains its previous status. The controlled RS-trigger is characterized by the following truth table.

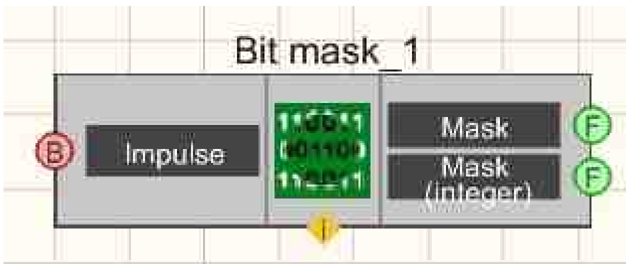
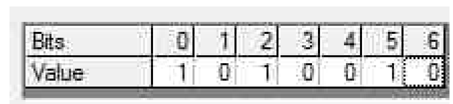
CLK	S	R	Q	¬Q	
1	0	0	Q_{-1}	$\neg Q_{-1}$	Like an RS-trigger
1	0	1	0	1	
1	1	0	1	0	
1	1	1	?	?	
0	X	X	Q_{-1}	$\neg Q_{-1}$	Information storage mode

15.3.Bit mask

The component **"Bit mask"**, It is intended for implement masking - select and set individual bits or sets of several bits from a binary string or number (allows to control devices having 2 status).

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Impulse - when a sync impulse is applied, it forms the set mask at the output.

➢ Output:
- Mask - a decimal number corresponding to the bitmask.
- Mask (integer) - the mask is passed as an integer.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Mask (1) - current decimal value of the mask.
- BitQuantity (1) - set the number of bits in the mask.
- FalseValue (0) - designation of logical zero.
- TrueValue (1) - designation of a logical unit.
- Bitsname (Bits) - set the name of the string with bits.
- ValuesName - set the name of the bit value string.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

- [Are available by the link below.] 154

➢Custom private properties (the default value is shown in parentheses):

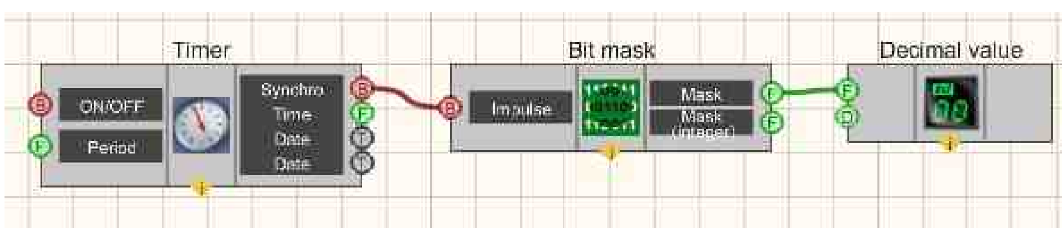
- LONGMask - Set the mask value (any integer of Long type).
- LONGBitQuantity - Set the number of digits (any number).
- BSTRFalseValue - Set the designation of a logical zero (string).
- BSTRTrueValue - Set the designation of a logical unit (string).
- BSTRBitsName - Set the name of the bits (string).
- BSTRValuesName - Set the names of the bit values (string).

➢Methods:

- LONG Bit(LONG num) - Returns the value of a bit.
- void Bit(LONG num, LONG newVal) - Set the name of the bits (string).
- Apply(void) - Set the names of the bit values (string).

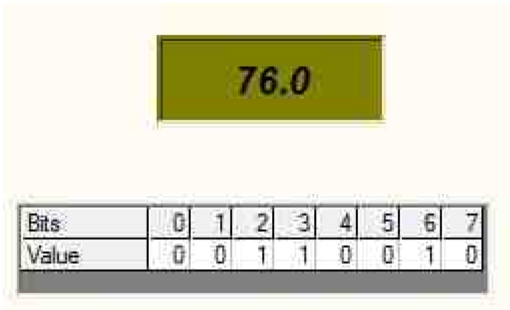
Example

Project in SCADA ZETView



This project shows the operation of the Bit mask 739 component performs a masking operation according to the set bits. The Timer 370 is used to give an impulse, according to which the mask will be applied. The Digital Indicator 682 is needed to display the mask in decimal form.

Project operation result



View the example in ZetView

Mathematical description

A bit mask is a set of bits that, using bitwise conjunction or disjunction, is superimposed on a set of bits in which you want to set or reset some bits.

in Example, in the number 01101111_2, the fifth bit must be discarded. To do this, we make a mask 11011111_2 and use the bitwise AND operation to perform a logical multiplication of these binary numbers.

Bitwise AND	01101111
	11011111
Result	01001111

If the fourth bit is to be set in the same number, 01101111_2, then this is done using a bitwise OR operation. To do this, we create a mask 00010000_2, then perform a logical addition.

Bitwise OR	01101111
	00010000
Result	01111111

15.4.Binary number

The component **"Binary number".** It is intended for transferring a decimal integer to its binary representation.

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Number - is used to convert a decimal integer to a binary representation.

➢ Output:
- Bit 1 - the status of the least significant (0th) bit of the translated number.
- ...
- Bit N - status N-th bit of the translated number.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- BitQuantity (1) - set the number of bits to translate the number.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Number - binary representation of a decimal integer(any number).

➢**Custom common properties:**

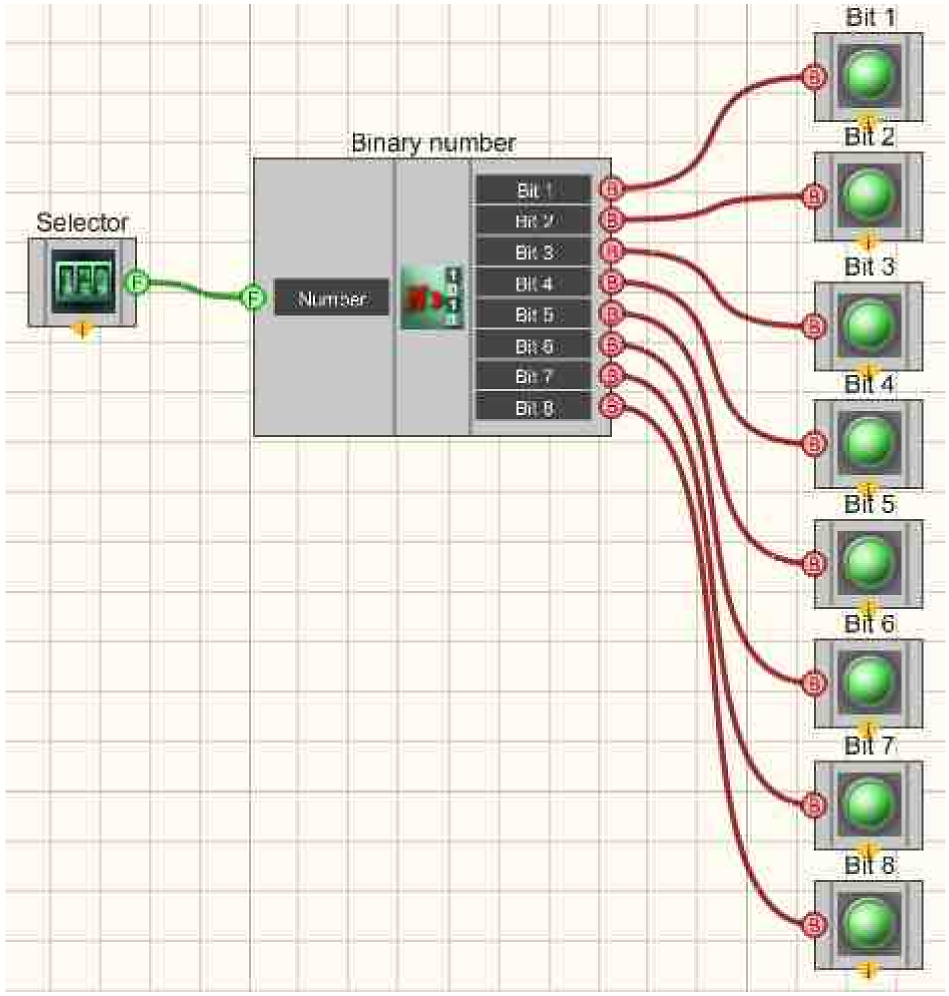
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG BitQuantity - Set the number of bits (from 1 to 249).

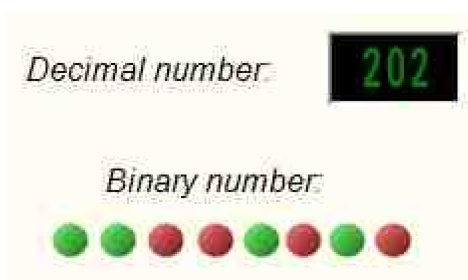
Example

Project in SCADA ZETView



This project shows the operation of the Binary number[743] component is used to convert a whole decimal number to binary form. The selector is needed to enter the desired number. Light indicators[669] user for graphic display of the binary form (green color - logical unit, red - zero).

Project operation result



View the example in ZetView

Mathematical description

The binary number system is a positional number system with base 2. In this system, a number is written only using 0 and 1. A bit is one bit of a binary code (binary digit). The binary system has become widespread in electronics because of its simplicity: it is immeasurably easier to simulate two states of an electronic circuit and then accurately distinguish them than three, four or more. Eight bits make up a byte. This has developed historically, and it’s more convenient this way: the number is a multiple of a power of two, i.e. scales easily. A sixteen-bit number is just two bytes, a tetrad (4 bits) is a nibble. There is also a binary-coded decimal code (binary-coded decimal, BCD code). They are widely used in ADCs. Each significant decimal digit in such a code is represented by four binary digits and contains ten signal values from 0 to 9. So in Example 13710 it is 0001 0011 0111BCD. Note that this representation is not equivalent to 13710=100010012. Obviously, binary coded decimal representation is not economical in terms of the use of digits, since each group of four bits is capable of "accommodating" 16 states, and only 10 of them are used.

Benefits of Binary number

:

- simplified input-output of numbers for indication;
- for fractional numbers (both fixed and floating point), when converted to a human-readable decimal format and vice versa, accuracy is not lost;
- multiplication and division by 10, as well as rounding, are simplified.

Disadvantages of Binary number

:

- complicated arithmetic operations;
- more memory is required.

By digital electronics, we mean circuits in which, as a rule, only two states can be determined at each point. Hence such attention to the binary system of calculus. in Example, the transistor is open or closed. As a rule, voltage is chosen as a parameter, not current. Hence the choice of levels: High and Low. In the Example for high-speed CMOS logic, input voltages from ground level to 1.5V are represented as Low, and voltages between 1.5V and 5V supply are represented as High. These states can be represented by different bits (binary digits) of information. in Example, the switch is closed or open, there is or is not a signal, the level of the analog signal is above or below the level - all this can be represented by one bit of the number.

Most logic circuits belong to one of the varieties - TTL (TTL, Transistor-Transistor Logic) and CMOS (CMOS, Complementary [type transistors] metal-oxide-semiconductor). CMOS technology uses insulated gate field-effect transistors with channels of different conductivities. A distinctive feature of CMOS circuits compared to bipolar technologies (TTL, ESL, etc.) is:

- very low power consumption in static mode (in most cases, it can be considered that energy is consumed only during switching states);
- the logic unit voltage is almost equal to the supply voltage, and the logic zero voltage is almost equal to the ground potential (with unloaded outputs);
- the switching threshold is almost equal to half the supply voltage;
- wide operating voltage range.

But compared to TTL, CMOS has a slow response time due to the fact that the insulated gate of a MOSFET is a rather large capacitor. Together with the output resistive resistance, such a capacitor forms a low-pass filter. In digital electronics, it is not just the frequency properties that are important, but the signal propagation delay time per logical element. The delay occurs due to the fact that the edge of the signal is not strictly vertical, but sloping. In this regard, when the signal at the output will only increase (decrease), while at the input it will already reach a value sufficient to switch the level. Another consequence of the high input capacitance of the CMOS is that the switching produces a current impulse to recharge this capacitance. As a result, extended fronts appear, hence the element is in active mode for quite a long time, when the output transistors are ajar. In combination with high-impedance inputs, this all leads to a decrease in noise immunity during switching.

15.5.Operation AND

The component **"Operation AND".** It is intended for implement such a logical operation as a conjunction.

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - the first channel with boolean values.
- Input 2 - the second channel with logical values.

➢ Output:
- Result - set the value of the Boolean function Logical AND from all input arguments.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FirstReaction (true) – Set the reaction to data change at input No. 1.
- SecondReaction (true) - Set the reaction to data change at input No. 2.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties**:

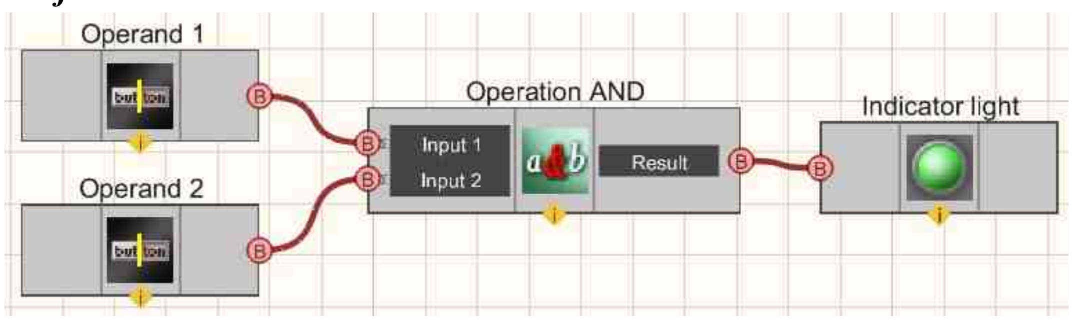
- Are available by the link below. 154

➢**Adjustable custom properties:**

- VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:
  true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.
- VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:
  true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.

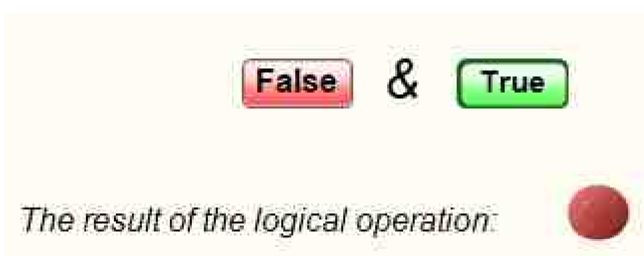
Example

Project in SCADA ZETView



In this diagram, the Operation AND 747 component is used to demonstrate the operation of the Boolean function Logical AND. The Fixed button 691 are needed to enter binary values. The Light indicator 669 is used to graphically display the result of the Boolean function Logic AND.

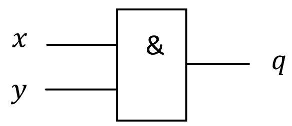
Project operation result



View the example in ZetView

Mathematical description

Logical element AND (conjunction, logical multiplication).



Element truth table.

x	y	q
0	0	0
0	1	0
1	0	0
1	1	1

The logical element that implements the conjunction function is called the match circuit. The mnemonic rule for conjunction with any number of inputs is: The output will be:

- 1 if and only if all inputs are "1";
- 0 if and only if at least one input is acted upon by "0".

15.6.Operation AND (multiple)

The component **"Operation AND (multiple)".** It is intended for implement such a logical operation as a conjunction. The component takes two or more input Boolean values.

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - the first channel with boolean values.
- Input 2 - the second channel with logical values.

➢ Output:
- Output - set the value of the Boolean function Logical AND from the input arguments.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Input (2) - set the number of inputs.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

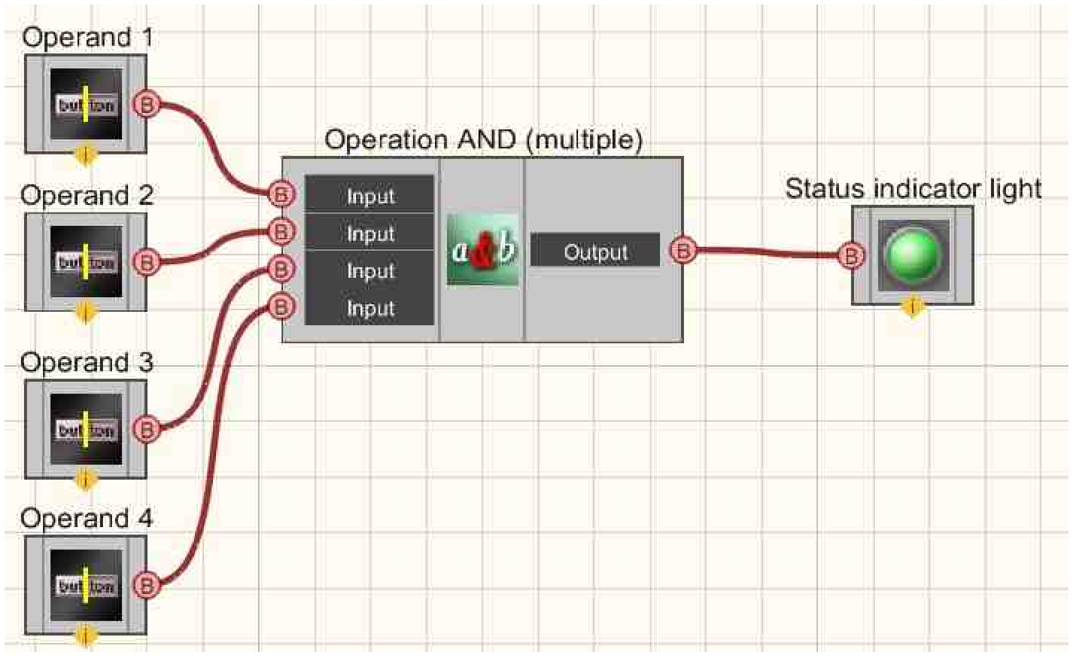
➢Custom common properties:
- Are available by the link below. 154

➢Adjustable custom properties:

- LONG Input - Set the number of inputs (from 1 to 249).

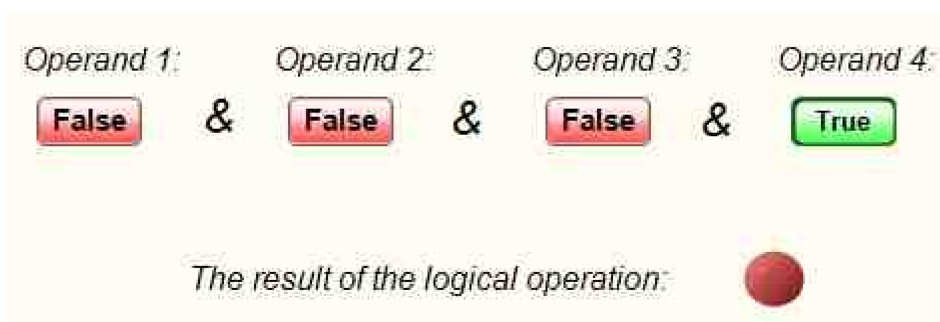
Example

Project in SCADA ZETView



This project shows the operation of the Operation AND (multiple) 750 component is used to implement the logical multiplication function. Fixed button 691 are needed for entering binary values-status. The Light indicator 669 is used to graphically display the result of the Boolean function Logical AND (for multiple arguments).

Project operation result



View the example in ZetView

15.7.Operation OR

The component **"Operation OR".** It is intended for implement such a logical operation as disjunction.

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Operand 1 is the first channel with boolean values.
- Operand 2 is the second channel with boolean values.

➢ Output:
- Value - set the value of the Boolean function Logical OR from all Operand arguments.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- FirstReaction (true) – Set the reaction to data change at input No. 1.
- SecondReaction (true) - Set the reaction to data change at input No. 2.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

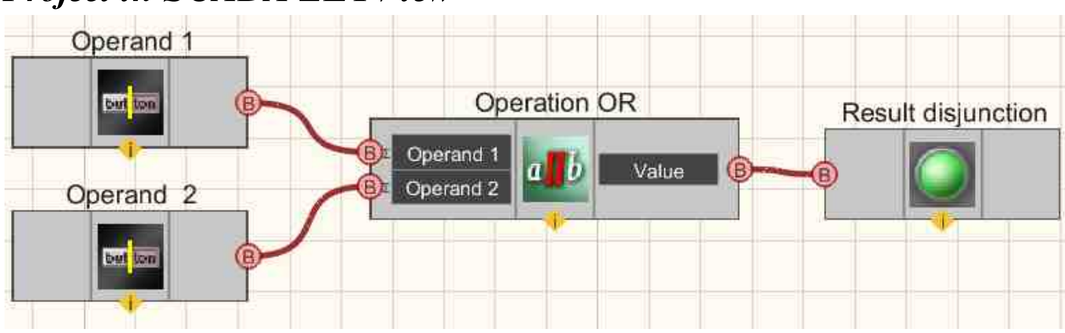
- Are available by the link below. 154

➢**Adjustable custom properties:**

- VARIANT_BOOL FirstReaction - Set the value of the reaction to data change at input 1:
  true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.
- VARIANT_BOOL SecondRaction - Set the value of the reaction to data change at input 2:
  true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.

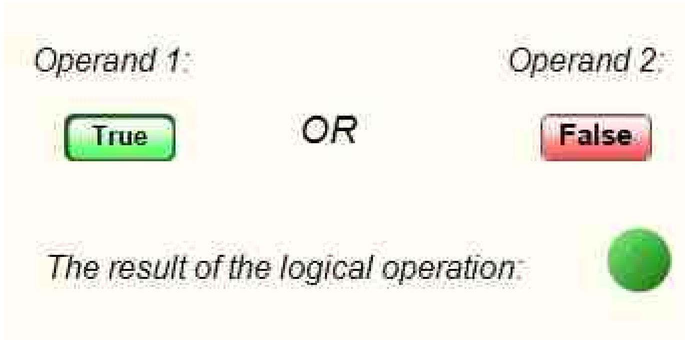
Example

Project in SCADA ZETView



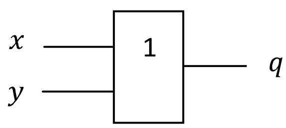
This project shows the operation of the Operation OR 753 component performs a disjunction operation on two input values. The Fixed button 691 are needed for entering binary values-status. The Light indicator 669 is used to graphically display the result of the Boolean function Logical OR (for multiple arguments).

Project operation result



Mathematical description

Logical element OR (logical addition, disjunction).



Element truth table.

x	y	q
0	0	0
0	1	1
1	0	1
1	1	1

The mnemonic rule for disjunction with any number of inputs is: The output will be:

- 1 if and only if "1" acts on at least one input;
- 0 if and only if all inputs are "0".

15.8.Operation OR (multiple)

The component **"Operation OR (multiple)".** It is intended for implement such a logical operation as disjunction. The component takes two or more input Boolean values.

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - the first channel with boolean values.
  Input 2 - the second channel with logical values.

➢ Output:
- set the value of the boolean function Logical OR from the input arguments.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Input (2) - set the number of inputs.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties**:

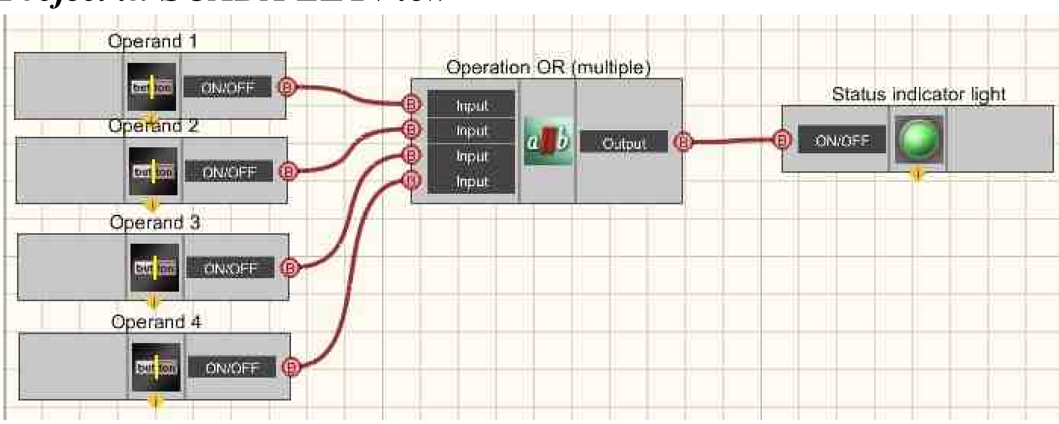
- Are available by the link below. 154

➢**Adjustable custom properties:**

- LONG Input - Set the number of inputs (from 1 to 249).

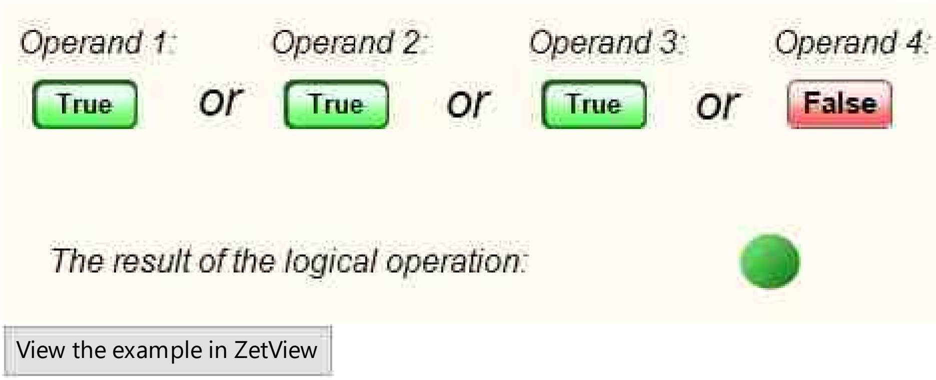
Example

Project in SCADA ZETView



This project shows the operation of the Operation OR (multiple) 750 component is used to implement the logical multiplication function. The Fixed button 691 are needed for entering binary values-status. The Light indicator 669 is used to graphically display the result of the Boolean function Logical OR (for multiple arguments).

Project operation result



15.9.Operation Not

The component **"Operation Not".** It is intended for designed to implement a logical negation operation..

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Operand - binary values are received that need to be inverted.

➢ Output:

- ! Operand - inverted values are received.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Adjustable custom properties:**Doesn't have.

Doesn't have.

Example

Project in SCADA ZETView

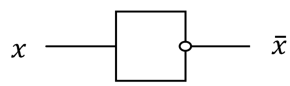
This project shows the operation of the Operation Not 758 component serves to invert the discrete value coming from the ZET.Push.1 758. The Light indicator 669 is needed to graphically display the logic level at the output.

Project operation result

View the example in ZetView

Mathematical description

Logic element Does not invert the input signal. Inversion is indicated by a circle at the output of the circuit.



Element truth table

x	NOT
0	1
1	0

The mnemonic rule for negation is: The output will be:

- 1 if and only if the input is "0";
- 0 if and only if the input is "1".

15.10.Overlay mask

The component **"Overlay mask"** (the difference is that these components can receive two ore more input values).
It is intended for masking - setting, resetting or checking individual bits (allows to control devices having 2 status).

Components of "Logic" section are not displayed in operator mode and can be added to the project only in developer mode (the only exception is "Bit mask" component). Components of "Logic" section allow to perform logical operations with data.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Number - the number with which the masking operation will be performed.
- Mask - a mask is supplied, which, using a bitwise conjunction, will be superimposed on the required number.

➢ Output:

- Output - set to a low logic level when all mask bits are zero, and high when the mask value is non-zero.
- Output - the value of the number is displayed after the masking operation is performed on it.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FirstReaction (true) - reaction to a change in the masked number.
- SecondReaction (true) - reaction to mask change.
- Value (0) - the default value of the number.
- Mask (0) - the default value of the mask.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Number - The value of the masked number (number).
- Mask - Mask value (number).

➤**Custom common properties:**

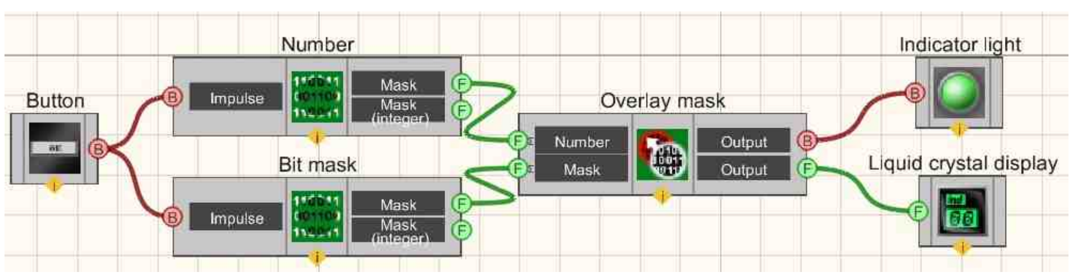
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL FirstReaction - Set the value of the response to a change in the data of the masked number:
  true - the output value is given when the masked number data arrives;
  false - the output value is not given when the data of the masked number is received.
- VARIANT_BOOL SecondReaction - Set the value of the response to a change in data to a change in the mask:
  true - the output value is given when the mask data arrives;
  false - the output value is not given when the mask data arrives.
- LONG Value - Set the default value of the number (number).
- LONG Mask - Setting The default value of the mask (number).

Example

Project in SCADA ZETView



This project shows the operation of the Overlay mask 760 component performs a masking operation. Bit masks 739 are used to represent Number and Mask in binary. By clicking on the Non-fixed button 694, the input values are transferred to the Overlay mask 760. The Liquid crystal display 655 is necessary for the graphical display in decimal form of the number after the masking operation. The Light indicator 669 is needed to indicate the status of the mask (all zeros or at least one one).

Project operation result

View the example in ZetView

Глава 16.Arrays

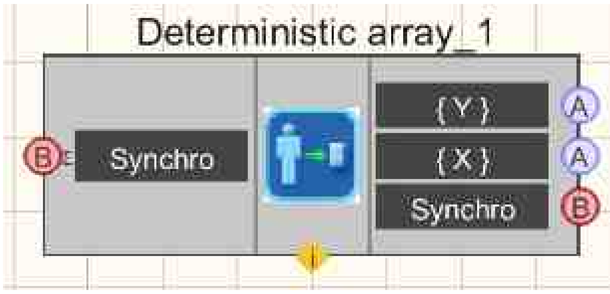
Let us consider "Array" clause in more details. This group contains elements for producing arrays of various types represented by components of "Formation of arrays type". "Formation of arrays (array-numeric)" is also widely used. In many cases, this component is an obligatory one, since in the course of signals measurements the virtual instruments produce a numerical value, while the components used for data processing have array as an input data.

As you can see from the figure, SCADA ZETView has several types of array formation. They are used to convert the data received into array of certain size or a dynamic array, the size of which depends on the amount of elements received. Elements of array formation of this type are widely implemented in projects related to measurements performance.

16.1.Deterministic array

The component "**Deterministic array**" is used for automated array formation based on formula type (constant, linear, square) and the parameters set.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Synchro - a impulse is applied to the input, after which the array is read from a file or other source into memory and the data is then transferred to the project.

➢ Output:
- { Y } - numerical values of array Y are received.
- { X } - numerical values of array X are received.
- Synchro - depending on the status of reading array elements from the selected source, 1 or 0 is set (1 - data reading, 0 - standby mode).

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Apolynomial (0) – Set the coefficient a when specifying an array through the formula ax*x+bx+c.
- Bpolynomial (0) - Set the coefficient b when specifying an array through the formula ax*x+bx+c.
- Cpolynomial (0) - Set the coefficient c when specifying an array through the formula ax*x+bx+c.
- Kexponenta (1) - Set the coefficient k when specifying an array through the formula exp(kx).
- FormulaQuantity (0) - Set the number of array calculation values by formula.
- FormulaStart (0) - Set the start value of the x argument when calculating the array of values using the formula.
- FormulaFinish (0) - Set the end value of the x argument when calculating an array of values using a formula.
- FileName - Set the location of the file with data for working with the file.

- formtype (Table) - Set the selection of the generated array (either from a table, or from a file, or a polynomial, or an octave, or a third octave series):

1. Table - manual input of array values.
2. File - getting array values from a file.
3. Polynomial - obtaining array values using linear, quadratic dependence and constant formulas.
4. Exponent - Set the getting exponent valuesexp(kx), where k is the parameter written in the field Kexponenta.
5. Octave row - Set the obtaining an octave row (values are taken in the range fromFormulaStart to FormulaFinish by Formula $10^{(3*x/10)}$, where x are integers).
6. Semi-octave row - Set the obtaining a semi-octave row (values are taken in the range from FormulaStart to FormulaFinish according to formula $10^{(3*x/20)}$, where x are integers).
7. Third-octave row - Set the obtaining a one-third octave row (values are taken in the range from FormulaStart to FormulaFinish according to formula $10^{(x/10)}$, where x are integers).
8. Six-octave row - Set the obtaining a semi-octave row (values are taken in the range from FormulaStart to FormulaFinish according to formula $10^{(x/20)}$, where x are integers).
9. Twelve-octave row - Set the obtaining a semi-octave row (values are taken in the range from FormulaStart to FormulaFinish according to formula $10^{(x/40)}$, where x are integers).

- ColNumber (1) - Set the number of the column in the data file to read the array. - Numbering starts from 0.
- TableSize (0) - Set the table size.
- readchannel (Sig_1_1) - Set the channel number when reading a wav-file.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

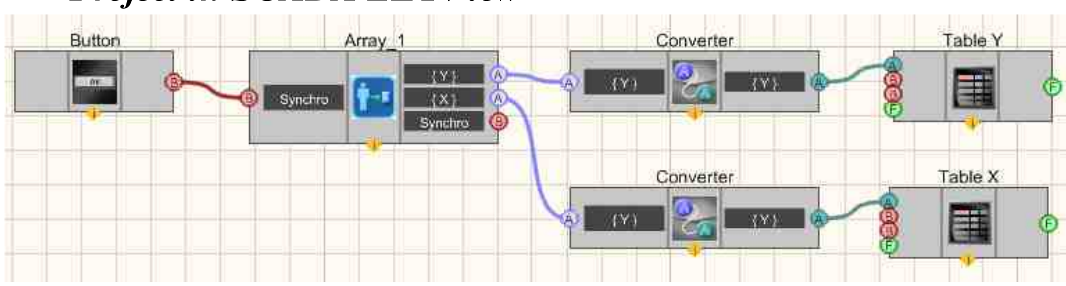
➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Apolynomial - Set the coefficient a when specifying an array through the formula ax*x + bx + c (any value).
- FLOAT Bpolynomial - Set the coefficient b when specifying an array through the formula ax*x + bx + c (any value).

- FLOAT Cpolynomial - Set the coefficient c when specifying an array through the formula ax*x + bx + c (any value).
- FLOAT Kexponenta - Set the coefficient k when specifying an array through the formula exp(kx) (any value).
- FLOAT FormulaQuantity - Set the number of values for calculating the array by the formula (any value).
- FLOAT FormulaStart - Set the start value of x when calculating the array by the formula (any value).
- FLOAT FormulaFinish - Set the end value of x when calculating an array using the formula (any value).
- BSTR FileName - Set the name of the file in which the array is stored (any string).
- BSTR FormType - Set the source of deterministic array (string): Table; File; Polynomial; Exponent; Octave row; Semi-octave row; Third-octave row; Six-octave row; Twelve-octave row.
- LONG ColNumber - Set the number of the column in the data file to read the array (number).
- LONG TableSize - Set the table size (number).
- BSTR ReadChannel - Set the channel number when reading a wav-file (string): Sig_1_1; Sig_1_2; Sig_1_3; Sig_1_4; Sig_1_5; Sig_1_6.

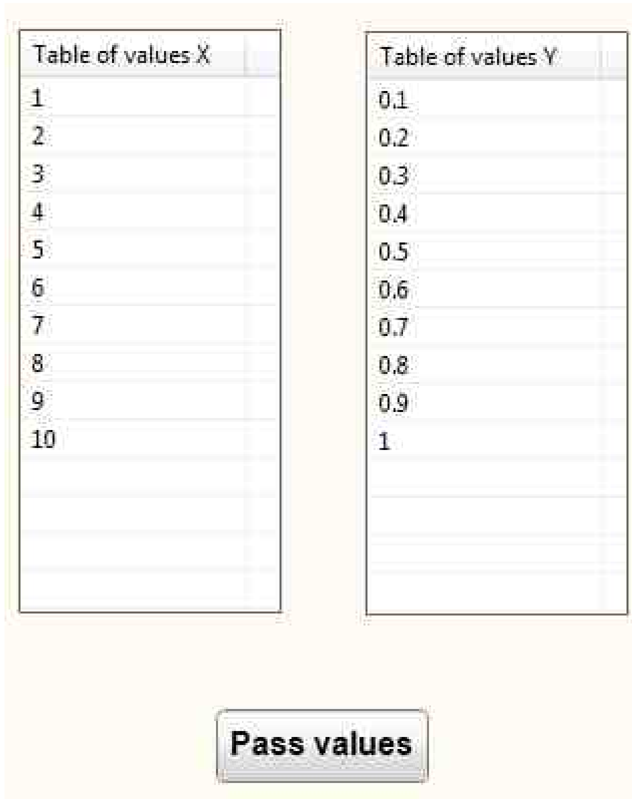
Example No. 1

Project in SCADA ZETView



This project shows the operation of the Deterministic array 763 component is designed to form an array of values calculated by the linear dependence formula. Converter (numeric array - string array) 763 are needed to convert the data type to a form compatible with Data tables 934 (this component allows you to display array data in the form of a table). A Non-fixed button 694 is needed to send a sync pulse, which will be used to transfer elements from the Deterministic array to tables 769.

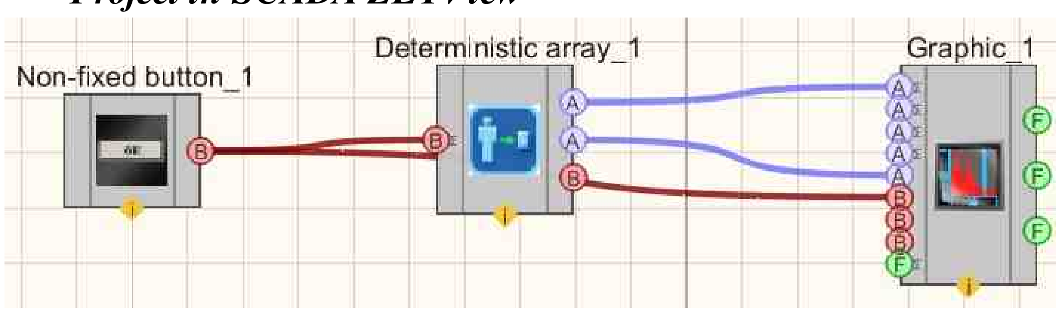
Project operation result



View the example in ZetView

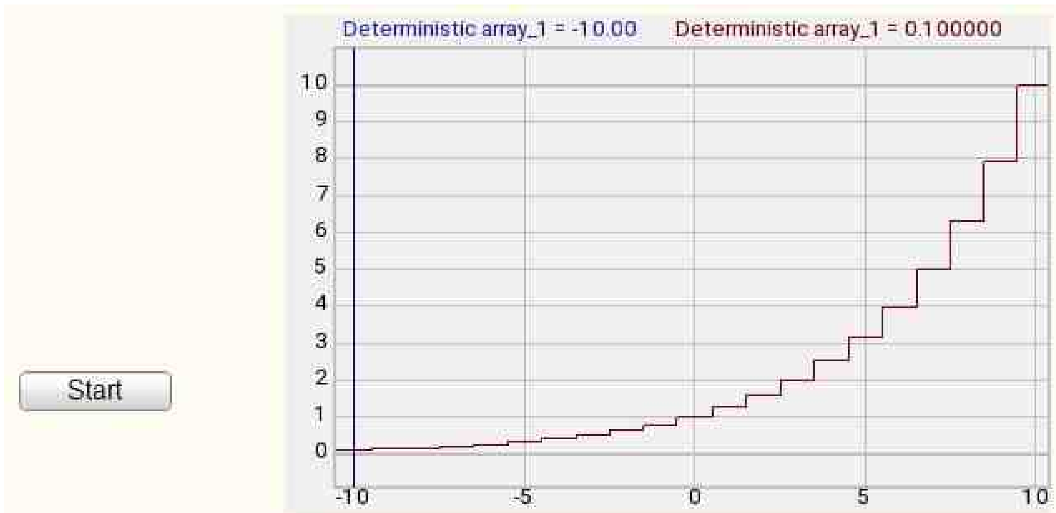
Example No. 2

Project in SCADA ZETView



This project shows the operation of the [Deterministic array](763) component is designed to form a receive third-octave row (values are taken in the range from FormulaStart to FormulaFinish according to formula $10^{(x/10)}$, where x are integers). A [Non-fixed button](694) is needed to send a sync inpulse, according to which elements from the [Deterministic array](763) will be transferred.

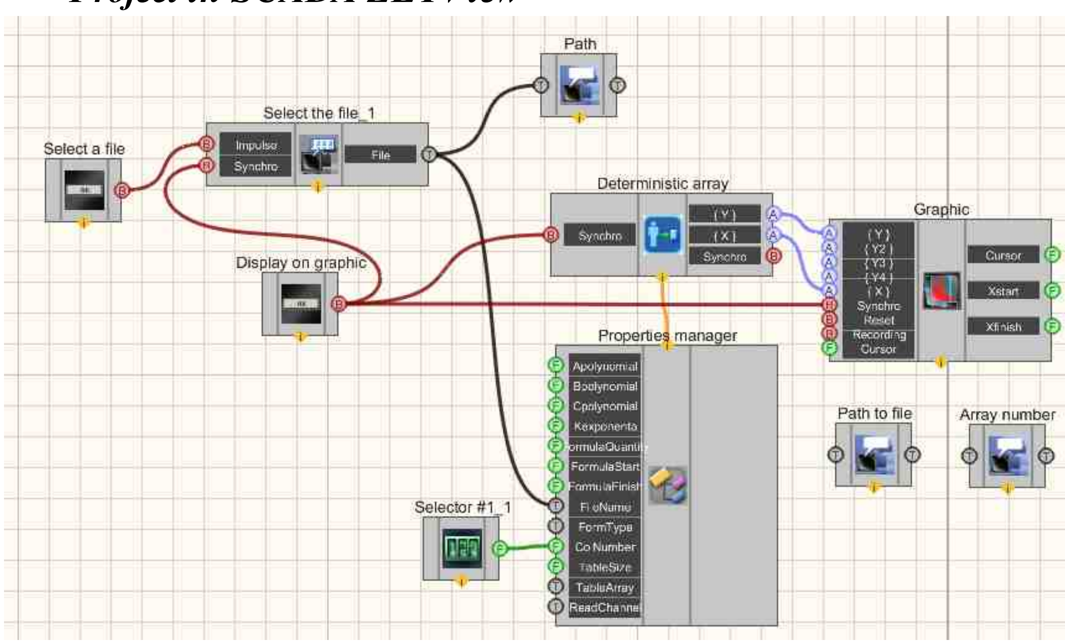
Project operation result



View the example in ZetView

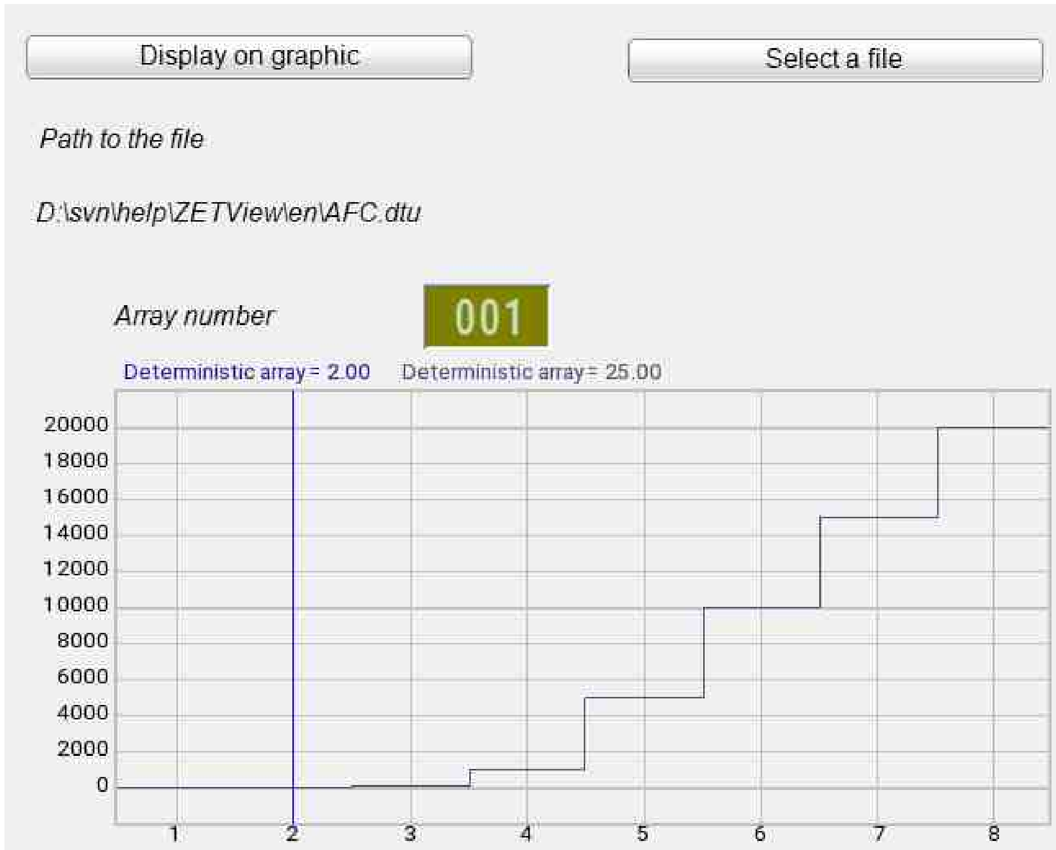
Example No. 3

Project in SCADA ZETView



This project shows the operation of the Deterministic array 763 component is designed to get a graphic from *.dtx files.

Project operation result



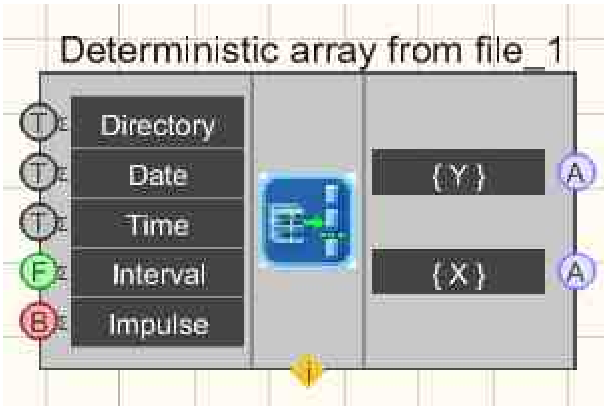
View the example in ZetView

16.2. Deterministic array from file

The component "**Deterministic array from file**" is used for array reproduction based on the previously recorded signal by means of "Signals recording" component.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- Directory - path to the folder containing the data archive *.ana, *.anp. and their locations on disk.
- Date - start time of data in DD.MM.YYYY format.
- Time - data start time in HH.MM.SS format.
- Interval - data collection interval, s.
- Impulse - impulse for reading data from the archive.

➢ Output:

- { Y } - numerical values of the array (instant signal values) Y.
- { X } - numerical values of the array (instant signal values) X.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Directory - path to the folder containing *.ana, *.anp data archive.
- Number (1) - channel number in the archive.
- Date - data start date in DD.MM.YYYY format.
- Time - data start time in HH.MM.SS format.
- Interval (1) - data collection interval, s.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Directory - A text string value indicating the path to the folder containing the data archive *.ana, *.anp. and locations on disk (text).
- Date - An input array for setting the time value, which contains the start time of the data in the format DD.MM.YYYY.
- Time - An input array for specifying the value of times, which contains the start time of the data in the HH.MM.SS format.
- Interval - The value of the data collection interval, s (any value).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•BSTR Directory - Set the path to the folder containing the data archive *.ana *.anp (any string).
•LONG Number - Set the channel number in the archive (number).
•BSTR Date - Set the data start date in the format DD.MM.YYYY (any string).
•BSTR Time - Set the data start time in HH:MM:SS format (any string).
•FLOAT Interval - Set the data collection interval, s (number).

16.3.Octave band row

The component "**Octave band row**" is used for range of octave bands generation.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - impulse for issuing the octave band row.

➢ Output:

- { Y } - numerical values of the array (instant signal values) Y are received.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- StartFrequency (125) - Set the start frequency for calculating the octave band row, Hz.
- EndFrequency (2000) - Set the end frequency for calculating the octave band row, Hz.
- octavepart (1/1) - Set the part of an octave (1/1 or 1/3).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

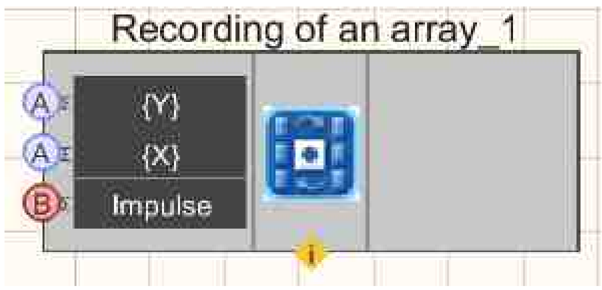
➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT StartFrequency - Set the start frequency for calculating the octave band row (from 0.1 to the end frequency for calculating the octave band row), Hz.
- FLOAT EndFrequency - Set the end frequency for calculating the octave band row (from the start frequency for calculating the octave band row to 1000000), Hz.
- BSTR OctavePart - Set the part of an octave (1/1 or 1/3).

16.4.Recording of an array

The component "**Recording of an array**" is used for recording of the array to a file upon receipt of signal at the input of the component.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - array of instantaneous values the recorded signal.
- { X } - array of indexes of array elements {Y}.
- Impulse - impulse for recording of an array to the file.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FileName - set the recording file name and location on the disk;
- UnitAxisX - set the data unit along the X axis;
- UnitAxisY - set the unit of data measurement along the Y axis.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} – the input array of instantaneous values the recorded signal.
- {X} - the input array of indexes of array elements {Y}.

➢**Custom common properties:**

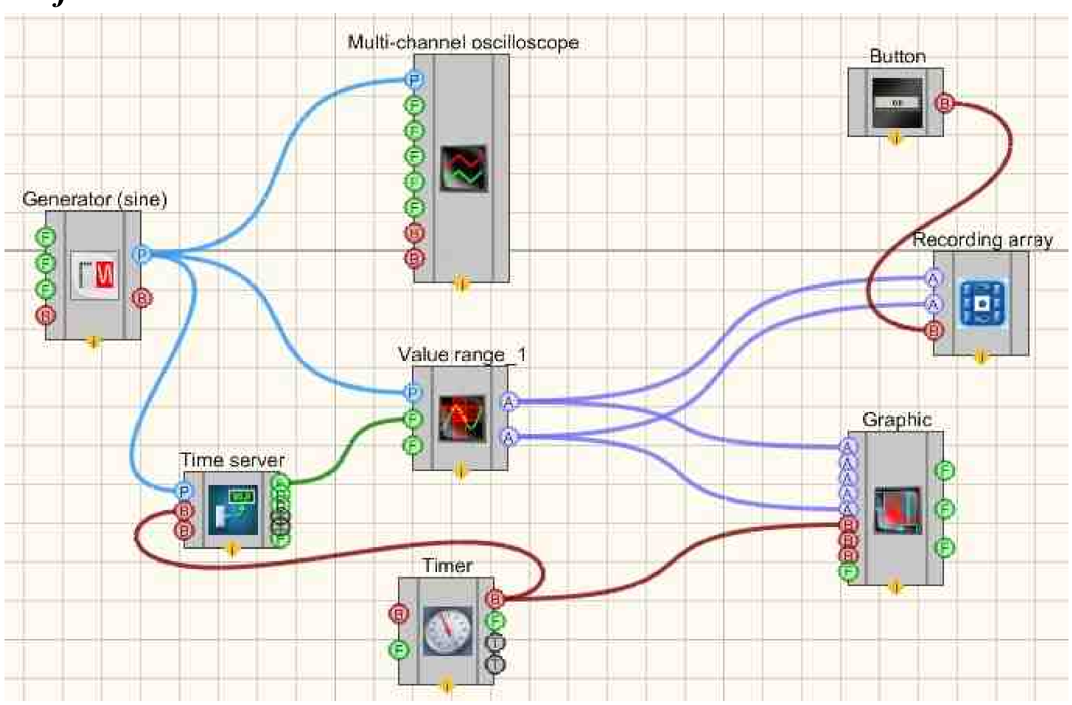
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•• BSTR FileName - Set the file name *.txt (string).

Example

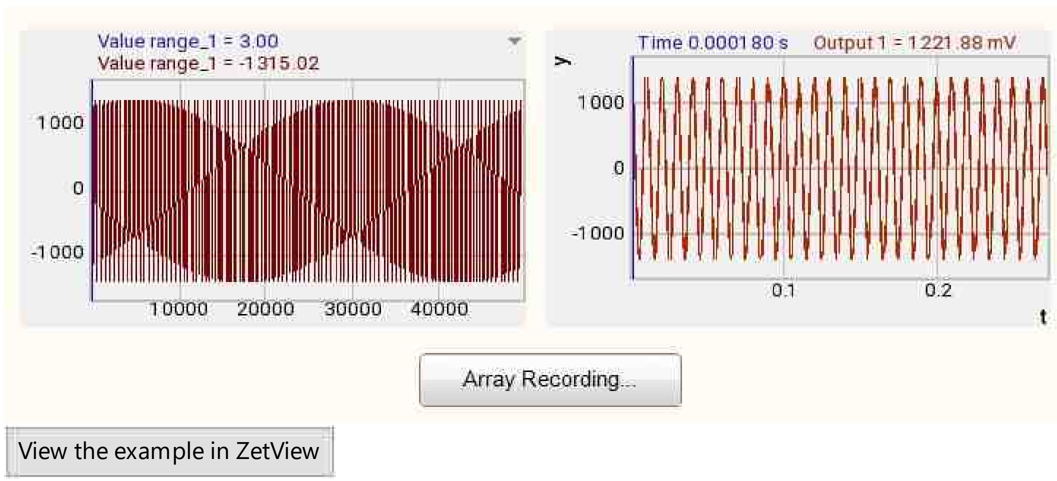
Project in SCADA ZETView



This project shows the operation of the Range of values 594 component is designed to obtain instantaneous values of the channel signal in a given time range. The Timer 370 is used to supply synchronization impulses to the Server time 366 and

Graphic 909. This is necessary so that the Graphic 909 and the Range of values 594 work in a single time space. The Sine signal 530 is needed to generate the test signal, and the Multi-channel oscilloscope 928 is for its current graphical display. Recording an array allows you to record a signal to a separate file.

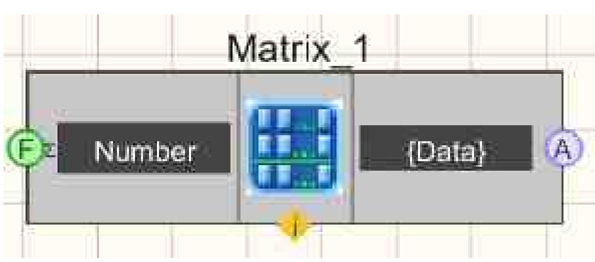
Project operation result



16.5.Matrix

The component "**Matrix**" is used for display of a particular string or column from the previously recorded data. It is used as a multi-unital array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Number - the number of the row/column to display.

➢ Output:

- {Data} - the output column/row data array.

Custom common properties:

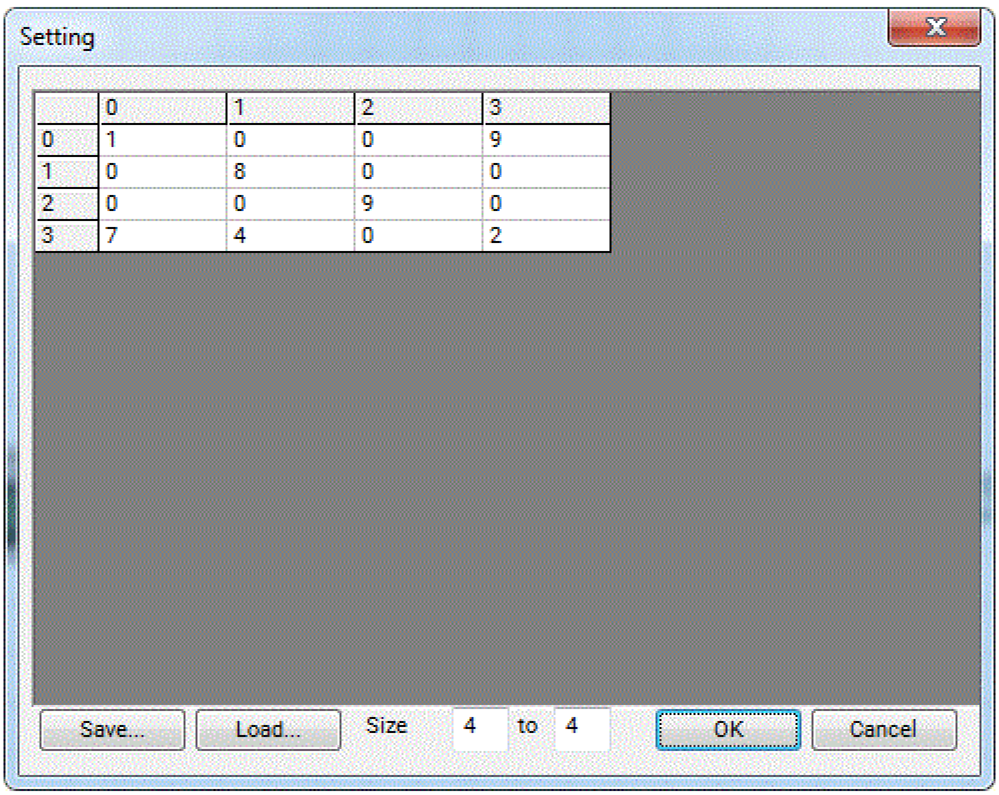
➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- outtype (Issuing a string by number) - a way to issue a matrix:
  1. Issuing a string by number;
  2. Issuing a column by number;
- Number (0) - the number of the returned row/column.

When calling the context menu of a component, **select Edit** component, after which matrix editing will open:



Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Number - the value of the number of the row/column to be displayed (number).

➢Custom common properties:

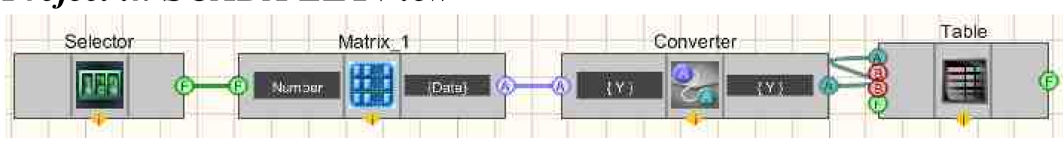
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR OutType - Set the way to output the matrix: Issuing a string by number; Issuing a column by number (any string).
- LONG Number - Set the number of output row\column (number).

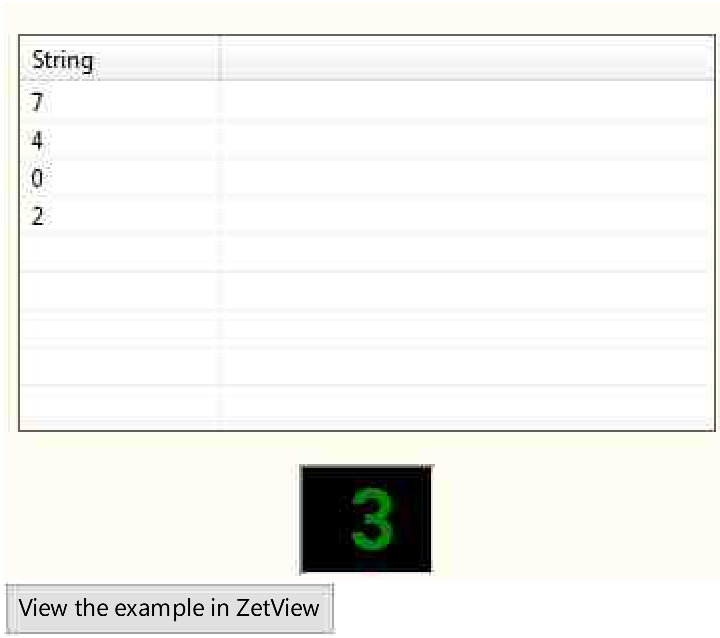
Example

Project in SCADA ZETView



This project shows the operation of the Matrix 775 component is designed to display, on request, the values written in a row. The Converter (numeric array - string array) 713 is used to convert a numeric array into a string array. The Selector 428 is needed to transfer the number of the required row to the Matrix 775. The Table 934 is used to display the values of a row.

Project operation result



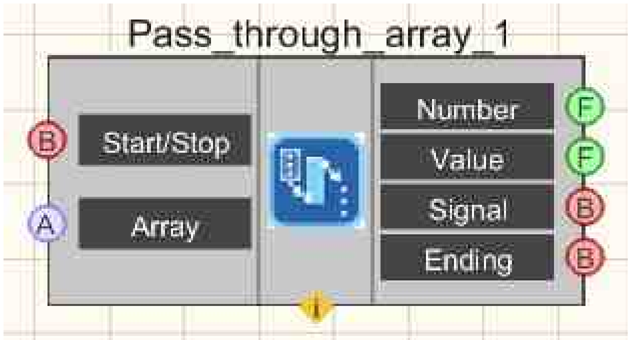
16.6. Pass of the array

The component "**Pass of the array**" is used for array elements analysis. E.g., this component may be used for frequency response control project in accordance with the set frequency array.

This component receives an array as input, and after receiving the start command, it outputs array elements one by one. The component has two configurable parameters:

1) DelayPass - delay between issuing array elements;
2) DelaySignal - delay of the impulse generated after updating the value of the current array element.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Start/Stop - a signal to start the passage of the array.
- Array - the array to be traversed.

➤ Output:
- Number - the current number of the returned array element.
- Value - the current value of the returned array element.
- Signal - the output impulse after the expiration of the set delay (a impulse with a certain delay after the output of the element).
- End – the impulse at the end of the array pass.

Custom common properties:

➤Total properties (environment):
- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):
- DelayPass (1) - delay between issuing array elements.
- DelaySignal (0.5) - delay of the impulse generated after updating the value of the current array element.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Array - The values of the array to be the array to be passed through (number).

➢**Custom common properties:**

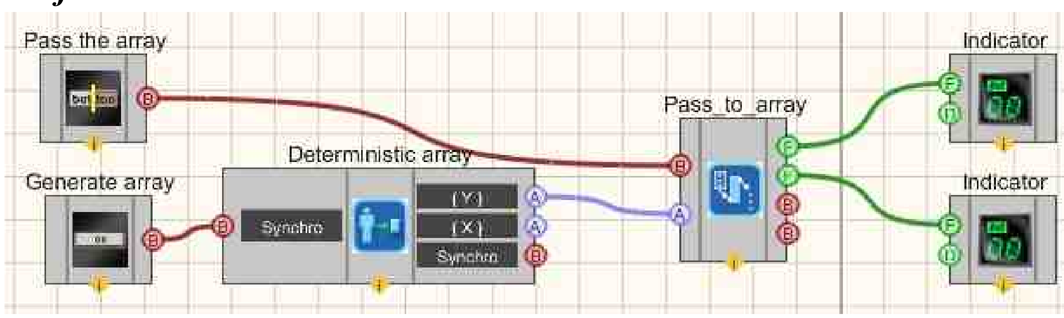
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

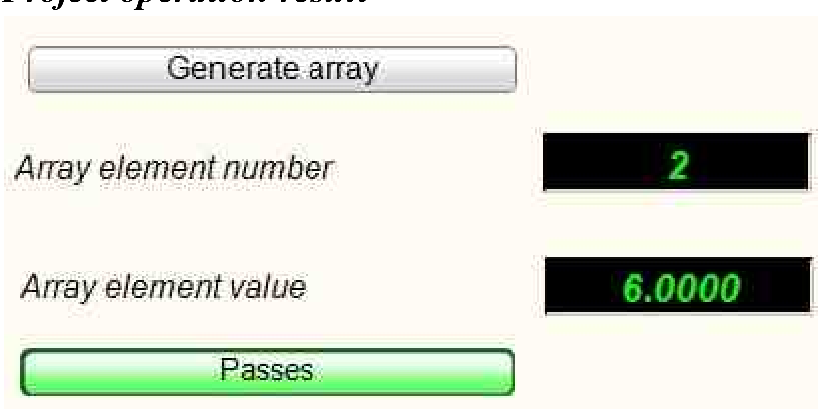
•FLOAT DelayPass - Set the delay for one pass (number).
•FLOAT DelaySignal - Set the output signal delay (number).

Example

Project in SCADA ZETView



Project operation result

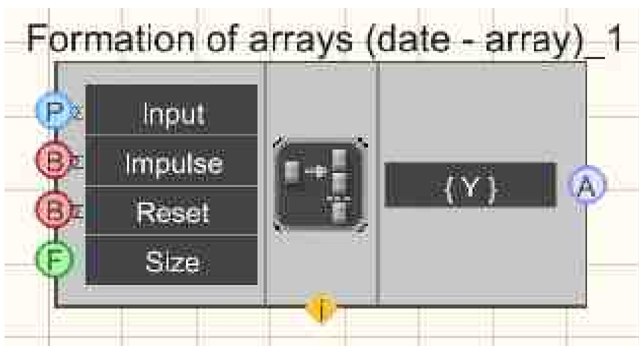


View the example in ZetView

16.7.Formation of arrays (date - array)

The component "**Formation of arrays (date - array)**" is records current date and time upon impulse receipt at the input. The component is used for measurements indication and formation of reports containing both measuring values and precise time of measurements.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - measuring channel.
- Impulse - impulse to record the current time-date value to the array.
- Reset - when a impulse is applied to this contact, the array is reset to zero.
- Size - set the number of elements in the array.

➢ Output:
- { Y } - generated time-date array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- ArrayAutoSize (true) - enable/disable array autosize.
- ArraySize (0) - set the size of the array.
- TimeSource(Computer) - set the time source: Computer; ZET-server; Measuring channel.

Note: When set the "ArrayAutoSize" property to TRUE, the component sets within itself an array of a certain size, specified by the "ArraySize" property. Each new number that comes to the input of the component is added to the beginning of the array, and the last number in the array is thrown out of the array.

Question. The program does not record data to the array.
Answer. When using the Formation of arrays (date - array) 781 component in the program, it is best to put the Non-fixed button 694 component on the "Reset" input of the component, which, when pressed, will send a impulse "1" and "0" in succession. Reset will occur when a "1" impulse is received. The Tumblers 696 component works the same as the Fixed button 691 component, it transmits its new status at the moment of switching. To reset the array using it, you must first turn it off, and then turn it on again.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Size - set the number of elements in the array (numeric value).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL ArrayAutoSize - Set the Enable/Disable array autosize:
  true - the array autosize is enabled;
  false - the array autosize is disabled.
- LONG ArraySize - Set the size of the array (number).
- BSTR TimeSource - Set the time source: Computer; ZET server; Measuring channel (string).

➢**Methods:**
- void ClearArray(void) - Clear array (any value).

16.8.Formation of arrays (string split)

The component "**Formation of arrays (string split)**" is operation principle is similar to that of "String separator" component. The only difference is that the user does not have to specify particular amount of output contacts for the strings. All the strings received will be recorded to text array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - serves the text that you want to split into parts.

➢ Output:

- Array - output string array.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Delimiter (;) - delimiter symbol.
- UseSpecDelimiter (False) - use a special symbol as a delimiter.
- SpecDelimiter (Tabulation) - special delimiter symbol: Space; Tabulation; Line break.

P ***Programming***

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - the value of the text string to be split into parts (text).

➢**Custom common properties:**

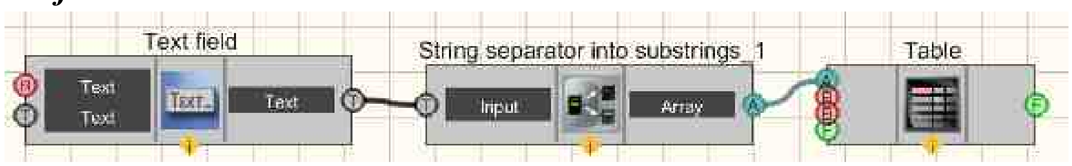
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

•BSTR Delimiter - Set the delimiter symbol (any string).

Example

Project in SCADA ZETView



This project shows the operation of the Formation of arrays (string split) 783 component is designed to split the input text into parts. The Table 934 is used to graphically display the result. A Text field 1100 is needed to organize text input.

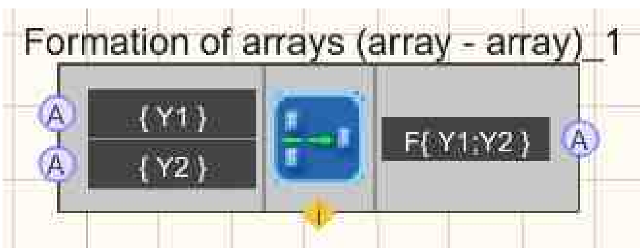
Project operation result

View the example in ZetView

16.9.Formation of arrays (array - array)

The component "**Formation of arrays (array - array)**" is used a single array from two input arrays using various methods: adding, averaging, minimum or maximum value, etc.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y1 } - the first array to perform actions with.
- { Y2 } - the second array to perform actions with.

➢ Output:
- F{ Y1; Y2 } - formed array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FirstReaction (true) - enable/disable the response to a change at input No. 1 (when the mode is activated, the data at the output is recalculated upon a change at this input).
- SecondReaction (true) - enable/disable the response to a change at input No. 2 (when the mode is activated, the data at the output is recalculated upon a change at this input).
- formtype (Union) - the type of transformation performed on the input arrays:
  1. Union - connection of two arrays into one.

2. Element-by-element addition - finding the sum of the corresponding elements of the array No. 1 and No. 2.

3. Element-by-element subtraction - finding the difference of the corresponding elements of the array No. 2 of No. 1.

4. Element-by-element multiplication - finding the product of the corresponding elements of the array No. 1 and No. 2.

5. Element-by-element division - finding the quotient of the corresponding elements of the array No. 1 to array No. 2.

6. Mutual correlation - finding correlation between arrays.

7. Element-by-element mean - finding the arithmetic mean of the corresponding array elements No. 1 and array No. 2.

8. Selection by indexes - use of additionally supported index structures to obtain data.

9. Element-by-element maximum - select among matching array values No. 1 and array No. 2 maximum values.

10. Element-by-element minimum - select among corresponding array values No. 1 and array No. 2 minimum values.

11. Element-by-element maximum (absolute value) - select among matching array values No. 1 and array No. 2 maximum values (absolute value).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y1} - The input first array with which you want to perform actions.
- {Y2} - The input second array with which you want to perform actions.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL FirstReaction - Set the value of the reaction to data change on input 1:

- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL FirstReaction - Set the value of the reaction to the data change on input 2:

- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.

BSTR FormType - Set the formation type:

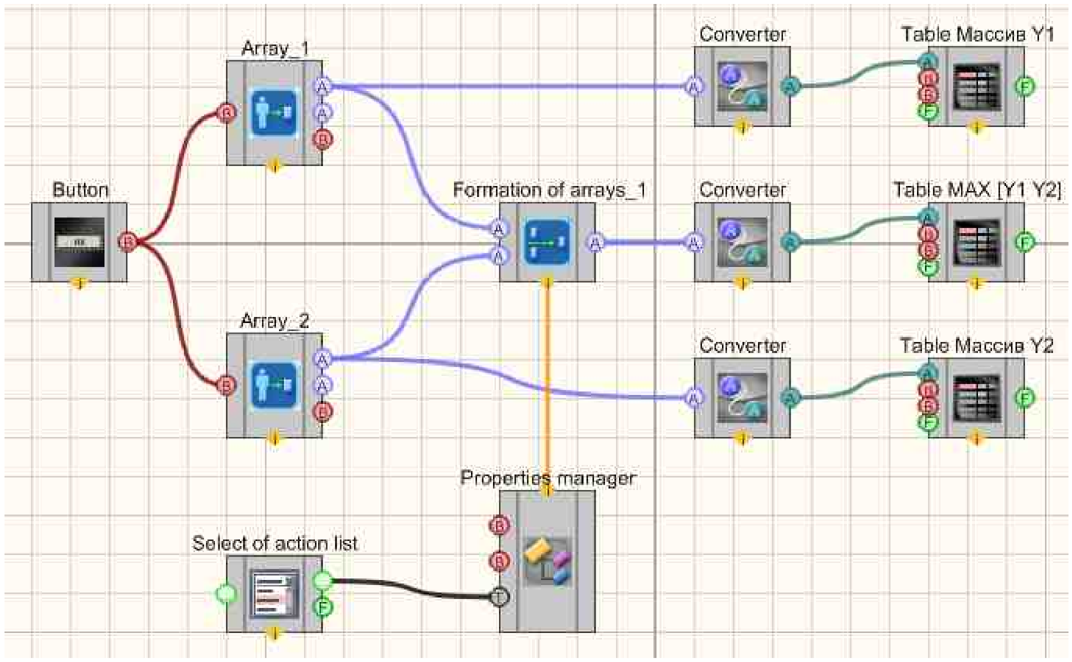
1. Union - connection of two arrays into one.
2. Element-by-element addition - finding the sum of the corresponding elements of the array No. 1 and No. 2.
3. Element-by-element subtraction - finding the difference of the corresponding elements of the array No. 2 of No. 1.
4. Element-by-element multiplication - finding the product of the corresponding elements of the array No. 1 and No. 2.
5. Element-by-element division - finding the quotient of the corresponding elements of the array No. 1 to array No. 2.
6. Mutual correlation - finding correlation between arrays.
7. Element-by-element mean - finding the arithmetic mean of the corresponding array elements No. 1 and array No. 2.
8. Selection by indexes - use of additionally supported index structures to obtain data.
9. Element-by-element maximum - select among matching array values No. 1 and array No. 2 maximum values.
10. Element-by-element minimum - select among corresponding array values No. 1 and array No. 2 minimum values.
11. Element-by-element maximum (absolute value) - select among matching array values No. 1 and array No. 2 maximum values (absolute value).

➢**Methods:**

- ULONG GetArraySize(void) - Returns the size of an array (any number).
- FLOAT GetValueFromIndex(LONG index) - Returns the element at the specified index from 1 to GetArraySize.

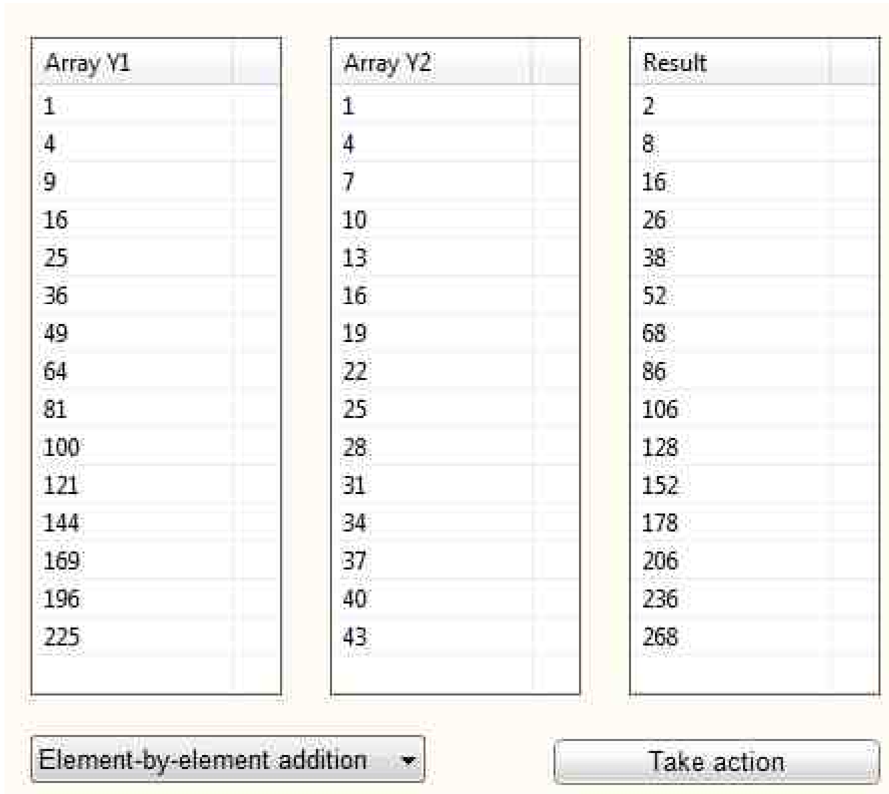
Example No. 1

Project in SCADA ZETView



This project shows the operation of the Formation of arrays (array - array) 785 component is designed to find the maximum values among the corresponding array elements No. 1 and array No. 2. Converters (numeric array - string array) 713 are needed to convert the data type to a form compatible with Data tables 934 (this component allows you to display array data in the form of a table). Deterministic array 763 are needed to create array No. 1 and array No. 2 using the built-in value generator.

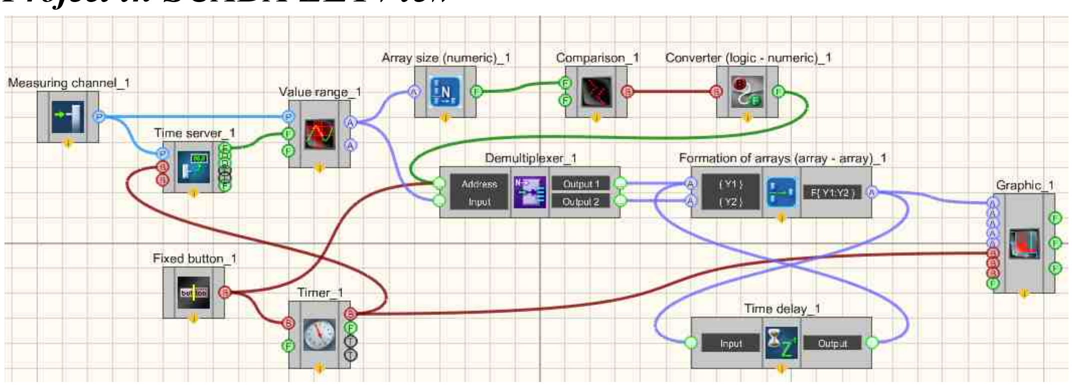
Project operation result



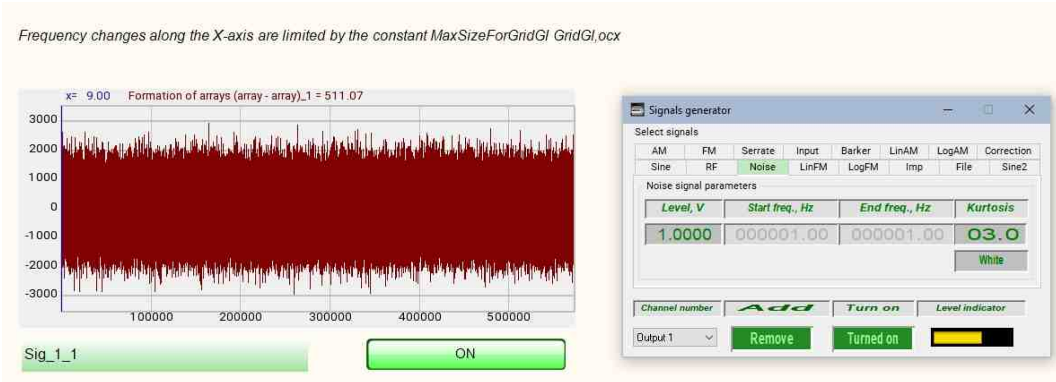
View the example in ZetView

Example No. 2

Project in SCADA ZETView



Project operation result



View the example in ZetView

16.10.Formation of arrays (element by element)

The component "**Formation of arrays (element by element)**" is receives numerical data and produces array for the elements.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- {Yn } - values to form an array.
- Reset - when a impulse is applied to this contact, the array is reset to zero.

➢ Output:
- { Y } - formed array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Size (500000) - set the size of the array.
- ProtectFromNaN (Use) - set the execution of the component's action when receiving NaNs (unrepresentable number) and Infinites (Infinity): Use; Previous; Replace by zero.

Question. The program does not record data to the array.
Answer. When using the Formation of arrays (element by element) component in the program, it is best to put the Non-fixed button 694 component on the "Reset" input of the component, which, when pressed, will send a impulse "1" and "0" in succession. Reset will occur when a "1" impulse is received. The Tumblers 696 component works the same as the Fixed button 691 component, it transmits its new status at the moment of switching. To reset the array using it, you must first turn it off, and then turn it on again.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

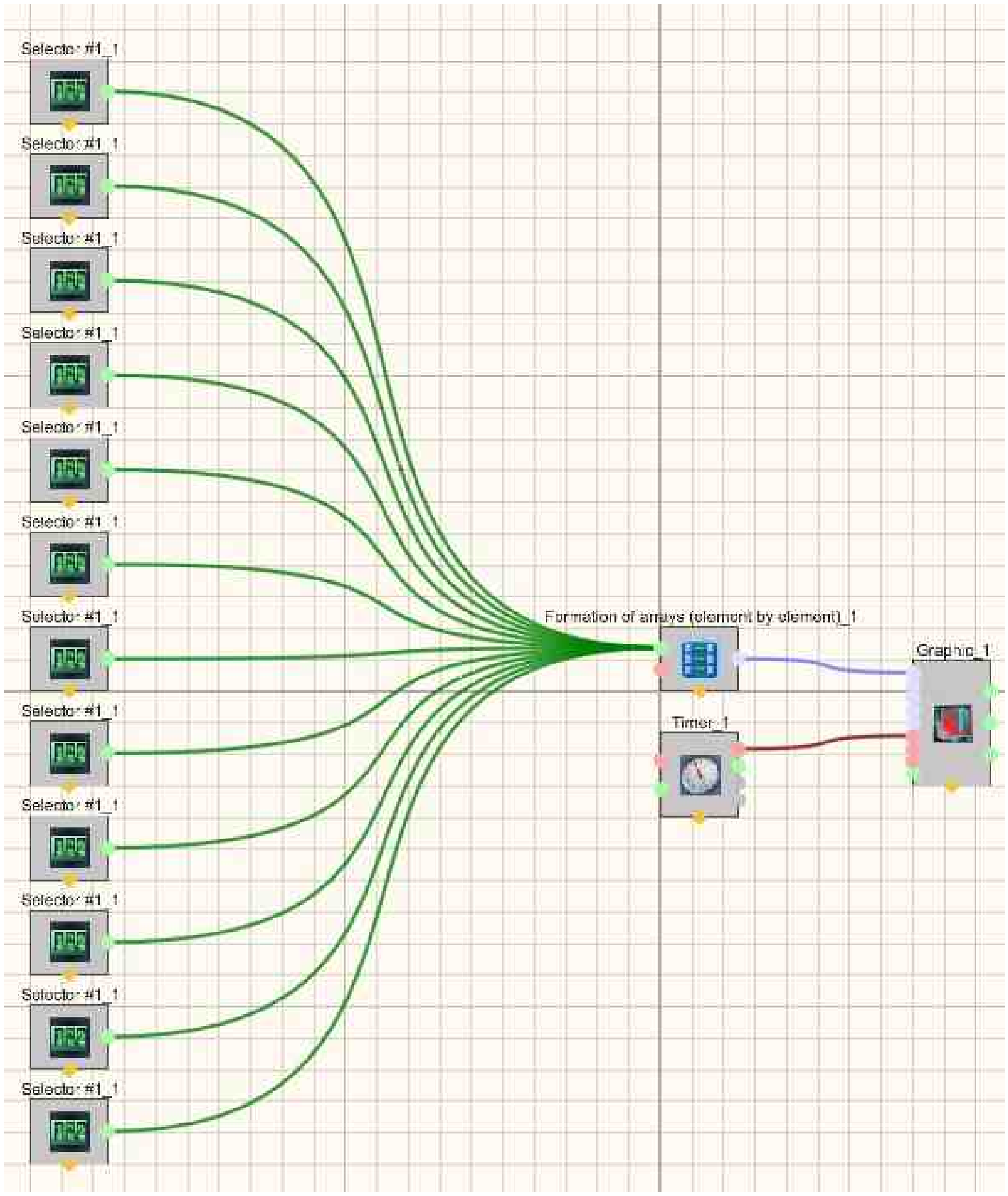
➢**Input:**
- {Yn } - a numeric value that contains information about the part number of the array.

➢**Custom common properties:**
- Are available by the link below. 154

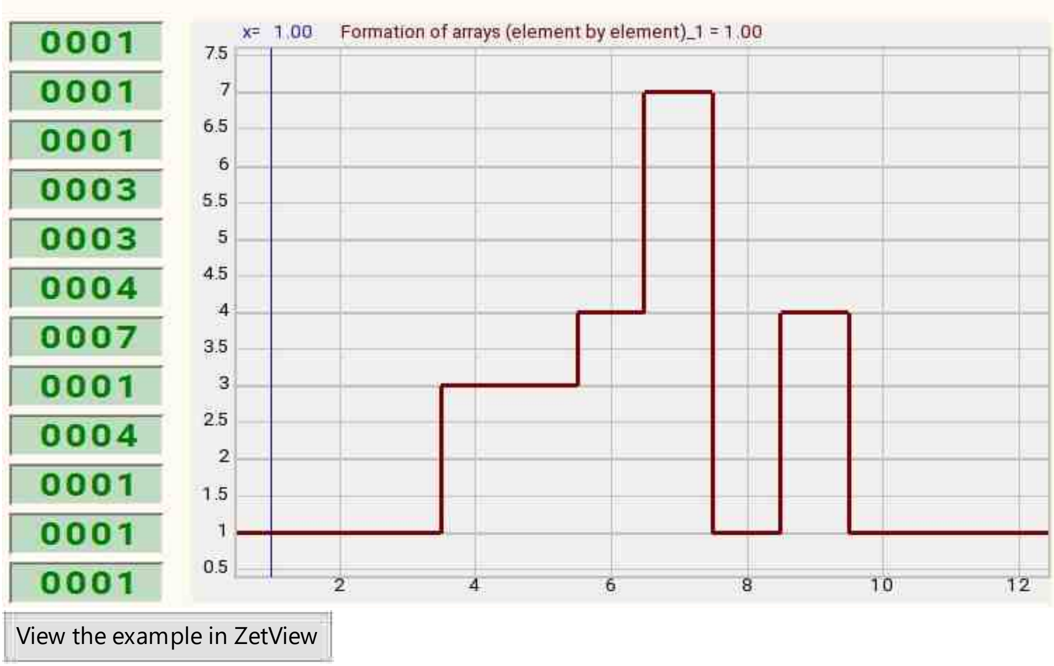
➢**Custom private properties (the default value is shown in parentheses):**
LONG Size - Set the size of the array (from 1 to 500000).
•BSTR ProtectFromNaN - Set the execution of the component's action when receiving NaNs (unrepresentable number) and Infinites (Infinity): Use; Previous; Replace by zero (string).

Example

Project in SCADA ZETView



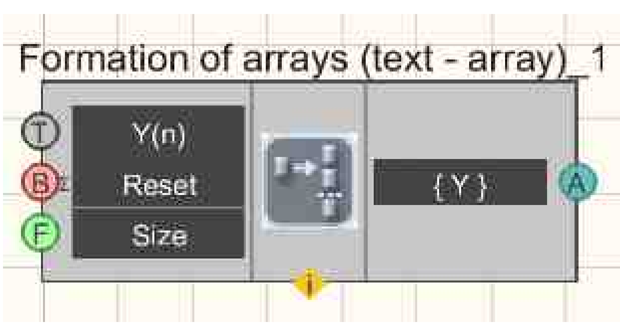
Project operation result



16.11.Formation of arrays (text - array)

The component "**Formation of arrays (text - array)**" is receives strings and produces string array with their further representation in tabular form, since it is the string array that is used for tabular data representation.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Y(n) - new symbol values to be written to the array.
- Reset - when a impulse is applied to this contact, the array is reset to zero.
- Size - set the number of elements in the array.

➢ Output:

- { Y } - formed array.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- ArrayAutoSize (true) - enable/disable array autosize.
- ArraySize (0) - set the size of the array.

Note: When set the "ArrayAutoSize" property to TRUE, the component sets within itself an array of a certain size, specified by the "ArraySize" property. Each new number that comes to the input of the component is added to the beginning of the array, and the last number in the array is thrown out of the array.

Question. The program does not record data to the array.
Answer. When using the Formation of arrays (text - array) 793 component in the program, it is best to put the Non-fixed button 793 component on the "Reset" input of the component, which, when pressed, will send a impulse "1" and "0" in succession. Reset will occur when a "1" impulse is received. The Tumblers 793 component works the same as the Fixed button 793 component, it transmits its new status at the moment of switching. To reset the array using it, you must first turn it off, and then turn it on again.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Y(n) - Character values to be written to the array.

- Size - Set the number of elements in the array.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL ArrayAutoSize - Set Enable/Disable array autosize:

- true - array autosize is enabled;
  false - array autosize is disabled.

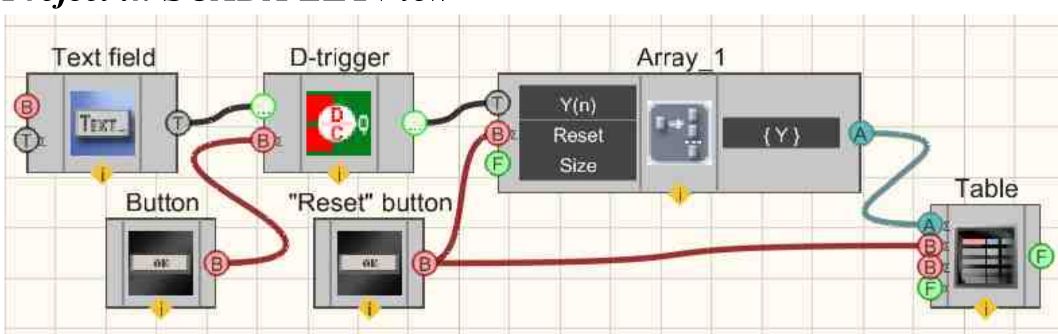
•LONG ArraySize - Set the size of the array (number).

➢**Methods:**

- void ClearArray(void) - Clear array (any value).
- BSTR GetValueFromIndex(LONG Index) - Read the value at the given array index (String).
- void SetValueToIndex(LONG Index, BSTR Value) - Set the value at the given array index (any value).
- void SendArray(); - Send array (any value).

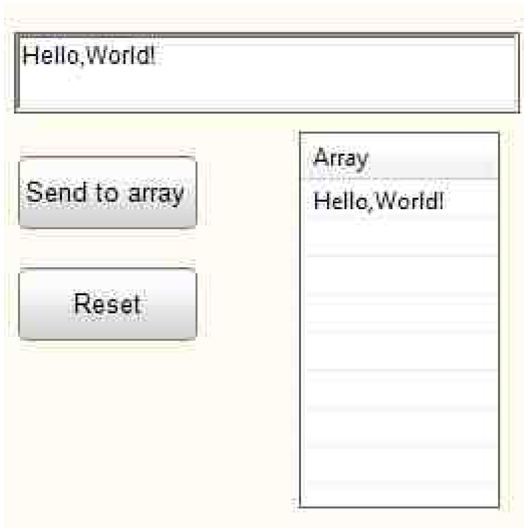
Example

Project in SCADA ZETView



This project shows the operation of the Formation of arrays (text - array) 793 component is designed to display the values of the Text field, when you click the "Send to array" button connected to the D-trigger 732, the data from the Text field 1100 is sent further. The Table 934 is needed to graphically display an array of strings.

Project operation result

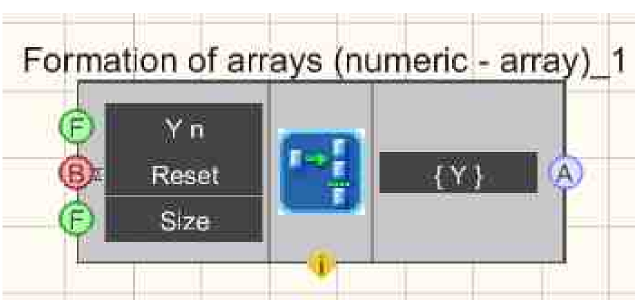


View the example in ZetView

16.12.Formation of arrays (numeric - array)

The component "**Formation of arrays (numeric - array)**" is receives numeric values and produces numeric array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- {Yn } - new numeric values to be written to the array.

- Reset - when a impulse is applied to this contact, the array is reset to zero.
- Size - set the number of elements in the array.

➢ Output:
- { Y } - formed numeric array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- ArrayAutoSize (true) - enable/disable array autosize:
 -true (automatic resizing enabled) - when a new number is received as input, it will be added to the end of the array, and the array size will increase by 1. In this case, the value of the "ArraySize" property of the Doesn't have value is always reset to 0 and is not saved . When starting a project, the array size is always 0.
 -false (automatic resizing disabled) - the size of the array is initially set by the "ArraySize" property. When an element of the next number enters the input, it will be added to the end of the array, the size of the array will not change, and the remaining elements of the array will be shifted by 1 element to the beginning, the first element from the array will be excluded;
- ArraySize (0) - set the size of the array.
- TopLimit (1) - set the size of the array when using autosize.

Note: When set the "ArrayAutoSize" property to TRUE, the component sets within itself an array of a certain size, specified by the "ArraySize" property. Each new number that comes to the input of the component is added to the beginning of the array, and the last number in the array is thrown out of the array.

Component operation algorithm

To the Formation of arrays (array-numeric) component, the "upper limit" property has been added. The property only works when the array property is selected "autosize".

The default property value is the maximum possible number of elements.
When the number of elements reaches this number, old values are removed from the array, and the array itself is shifted to fill with new ones.

Question. The program does not record data to the array.
Answer. When using the Formation of arrays (numeric - array) 796 component in the program, it is best to put the Non-fixed button 694 component on the "Reset" input of the component, which, when pressed, will send a impulse "1" and "0" in succession. Reset will occur when a "1" impulse is received. The Tumblers 696 component works the same as the Fixed button 691 component, it transmits its new status at the moment of switching. To reset the array using it, you must first turn it off, and then turn it on again.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**
- {Yn } - new numeric values to be written to the array.
- Size - Set the number of elements in the array.

➤**Custom common properties:**
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

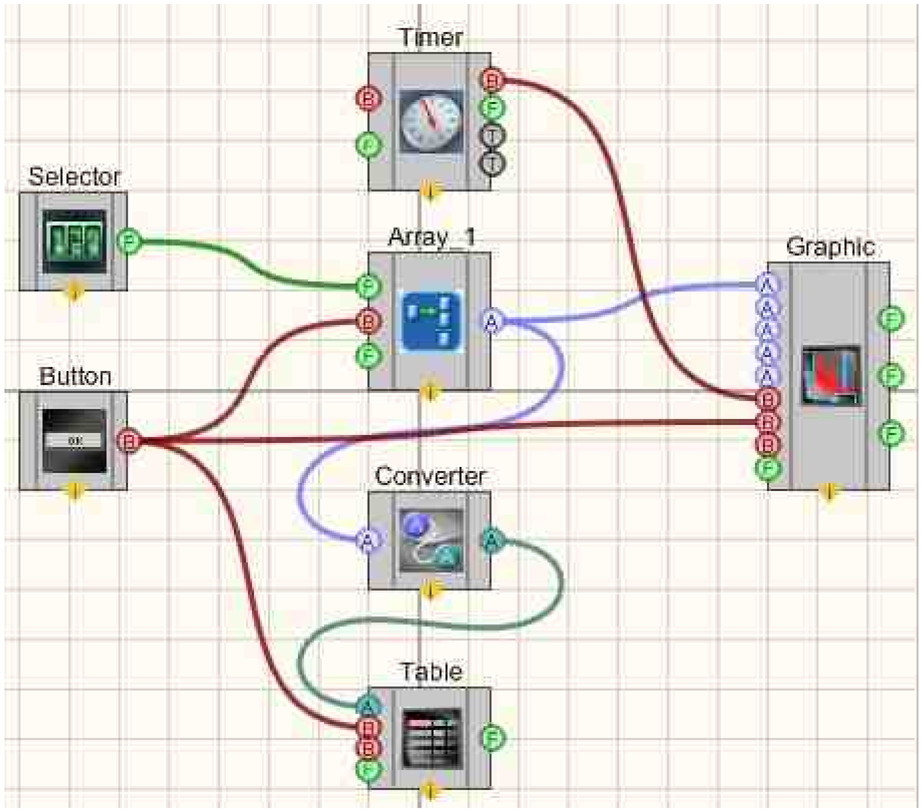
VARIANT_BOOL ArrayAutoSize - Set Enable/Disable array autosize:
- true - array autosize is enabled;
  false - array autosize is disabled.

•LONG ArraySize - Set the size of the array (number).
•LONG TopLimit - Set the size of the array when using autosize (number).

➤**Methods:**
- void ClearArray(void) - Clear an array (any value).
- void AddValueToArr(FLOAT newVal) - Add new values to the array (any value).
- void SetValueToIndex(LONG index, FLOAT value) - Set values at the given array index (any value).
- FLOAT GetValueFromIndex(LONG index) - Read the values at the given array index (any number).
- void SendArray(void) - Send an array (any value).

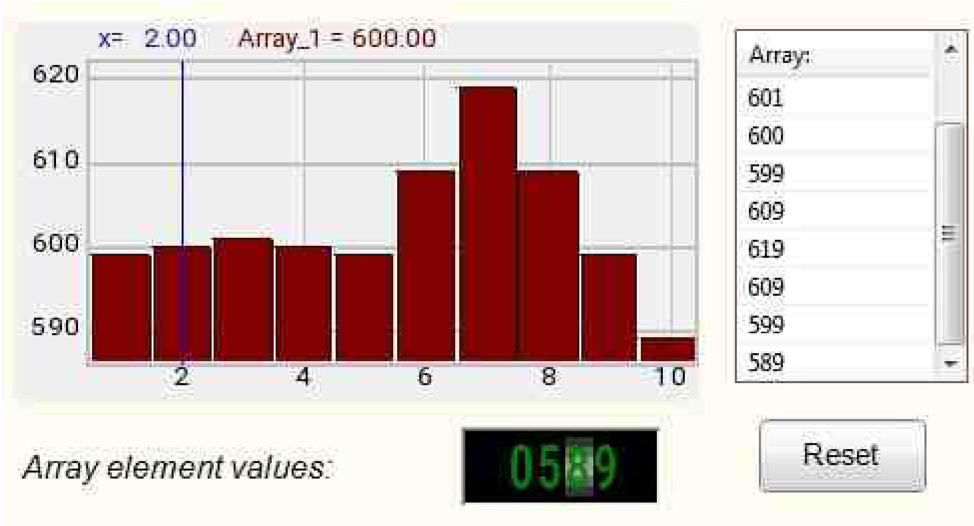
Example

Project in SCADA ZETView



This project shows the operation of the Formation of arrays (numeric - array) 796 component is designed to fill the array with numbers that come from Selector No. 1 428. A Non-fixed button 796 is used to reset the array. The Timer 370 is used to supply sync impulses to the Graph. The Converter (numeric array - string array) 796 is needed to convert the data type to a form compatible with the Data table 934 (this component allows you to display array data in the form of a table).

Project operation result



View the example in ZetView

Глава 17.Array - Value

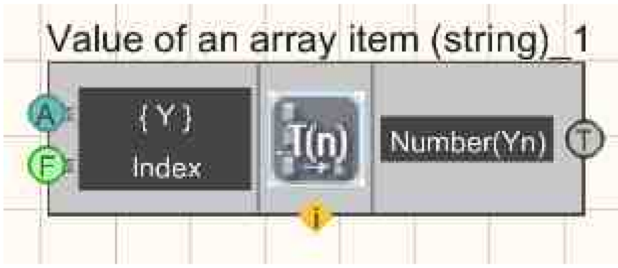
The section "Array-value" contains elements used for the processing of obtained array and producing the required value. Components of the present section allow to set top and bottom limit, calculate element value within the array or a sum of elements.

Components of "Array-value" and "Array-array" section are used for processing of data arrays.
Formation of arrays is used a get arrays from the values sent to the input of the component.
Arrays handler is used for arrays processing (sorting, filtration and synchronization of data).

17.1.Item value (string)

The component "**Item value (string)**" is intended for obtaining an array element by index.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- {Y } - string array to be taken at the given index.
- Index - the number (index) of the element, which will subsequently be passed to the output of the component.

➢ Output:
- Number(Yn) - an element arrives, with a given index.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- IndexOfValue (0) - index of the required array element.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- {Y} - values of the string array to be taken with the given index.
- Index - the number (index) of the element, which will subsequently be passed to the output of the component.

➤**Custom common properties:**

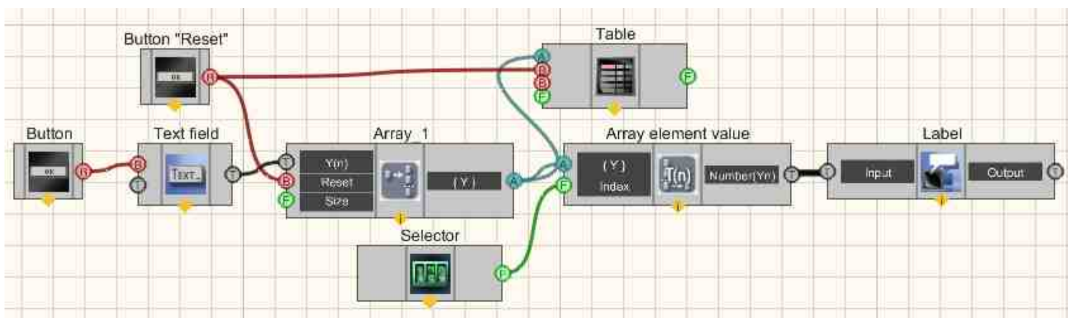
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

•FLOAT IndexOfValue - Set the index of the desired array element (number).

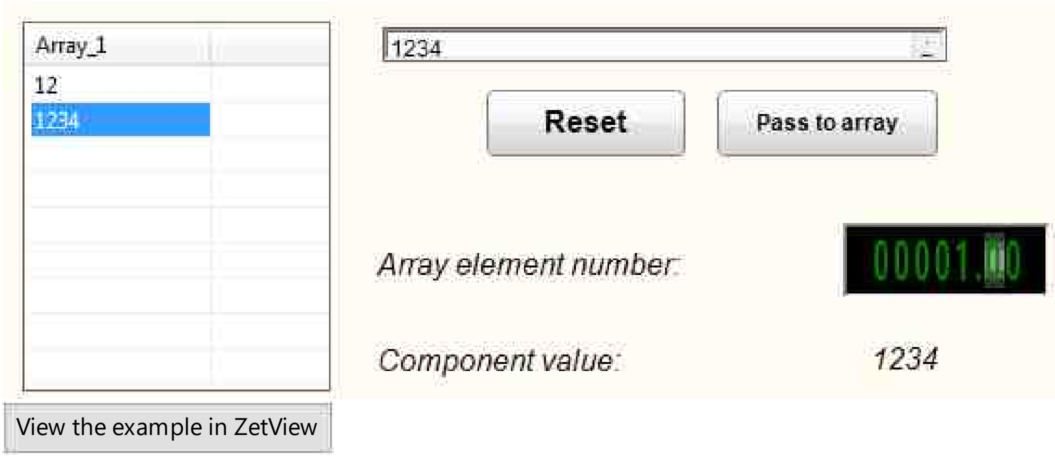
Example

Project in SCADA ZETView



This project shows the operation of the Item value (string) 800 component. The array is formed by the user by entering values in the Text field 1100 (the "transfer by impulse" setting is set on the component) and pressing the "Transfer to array" Non-fixed button 694. After these actions, the string is sent to the Formation of arrays (text - array) 793 from which it enters the Data table 934 (so that the user can see what values he enters into the array). The Selector 428 sets the index of the desired array element. The Item value (string) 800 component returns the value of the element at the specified index. The result is displayed using the Label 1087. Non-fixed button 694 "Reset" connected to Formation of arrays (text - array) 793 and Data table 934 resets the array values and clears the table.

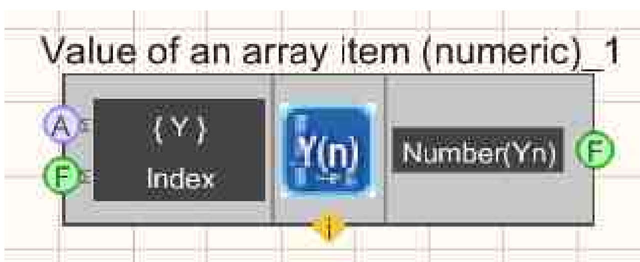
Project operation result



17.2.Item value (numeric)

The component "**Item value (numeric)**" is intended for obtaining an array element by index.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - numeric array values that will be taken with the given index.
- Index - the number (index) of the element is received, which will subsequently be passed to the output of the component.

➢ Output:

- Number(Yn) - an element arrives, with a given index.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- IndexOfValue (0) - index of the required array element.
- FirstReaction (true) - reaction to data change at input No. 1
- SecondReaction (true) - reaction to data change at input No. 2

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - numeric values of the array to be taken with the given index.
- Index - the number (index) of the element, which will subsequently be passed to the output of the component (number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•FLOAT IndexOfValue - Set the index of the required array element (number).
VARIANT_BOOL FirstReaction - Set the value of the reaction to data change on input 1:

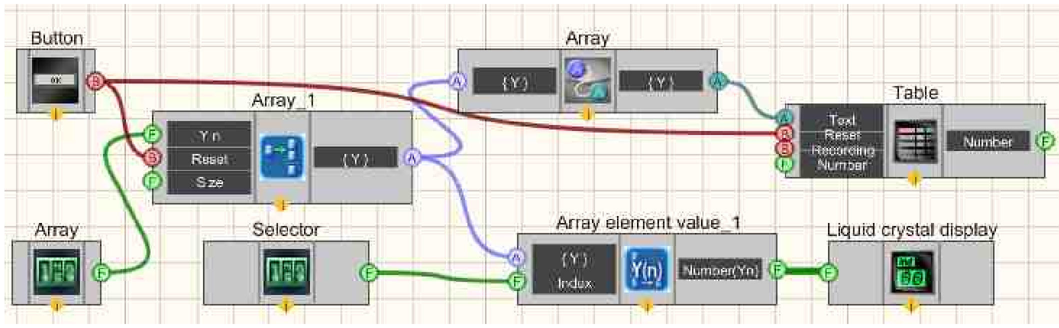
- true - the output value is given when data arrives at input 1;
  false - the output value is not given when data arrives at input 1.

VARIANT_BOOL FirstReaction - Set the value of the reaction to the data change on input 2:

- true - the output value is given when data arrives at input 2;
  false - the output value is not given when data arrives at input 2.

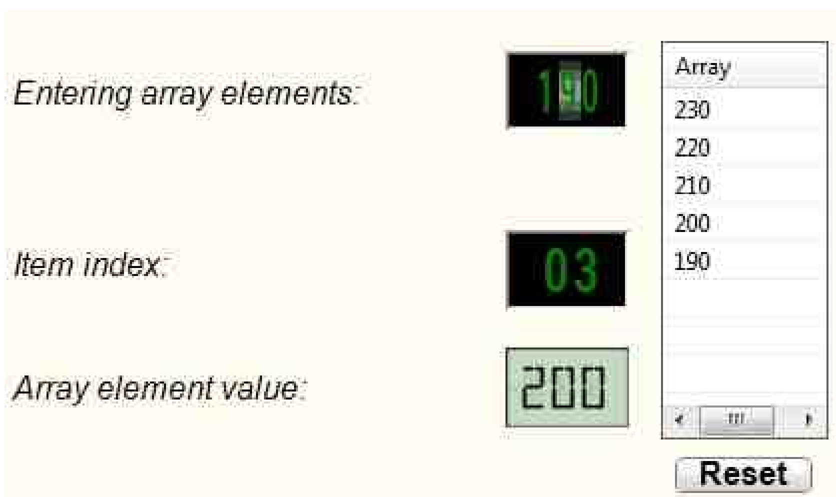
Example No. 1

Project in SCADA ZETView



This project shows the operation of the Item value (numeric) 803. An array is entered from the Selector 428 to the Formation of arrays (numeric - array) 796, after which the data is sent to the Converter (numeric array - string array) 803, where it is converted from a numeric array to a string array and output to the Data table 934. The Item value (numeric) 803 component receives the array entered by the user and the element index coming from the second selector and transmits the value of the element having the desired index to the Liquid crystal display 655.

Project operation result

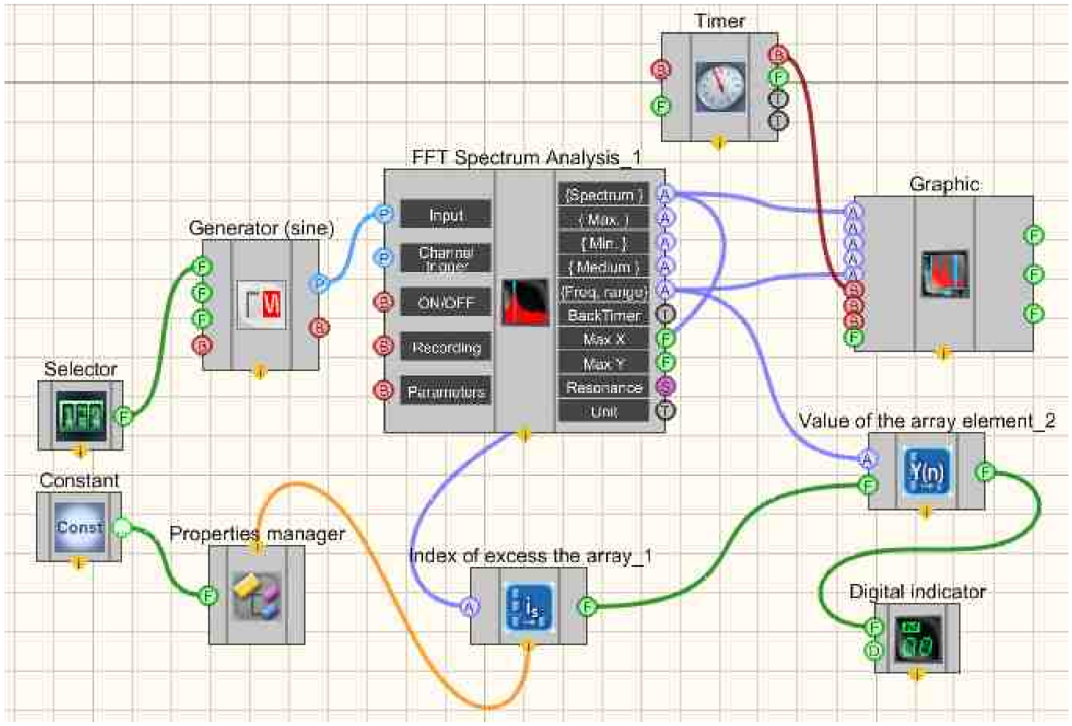


View the example in ZetView

Example No. 2

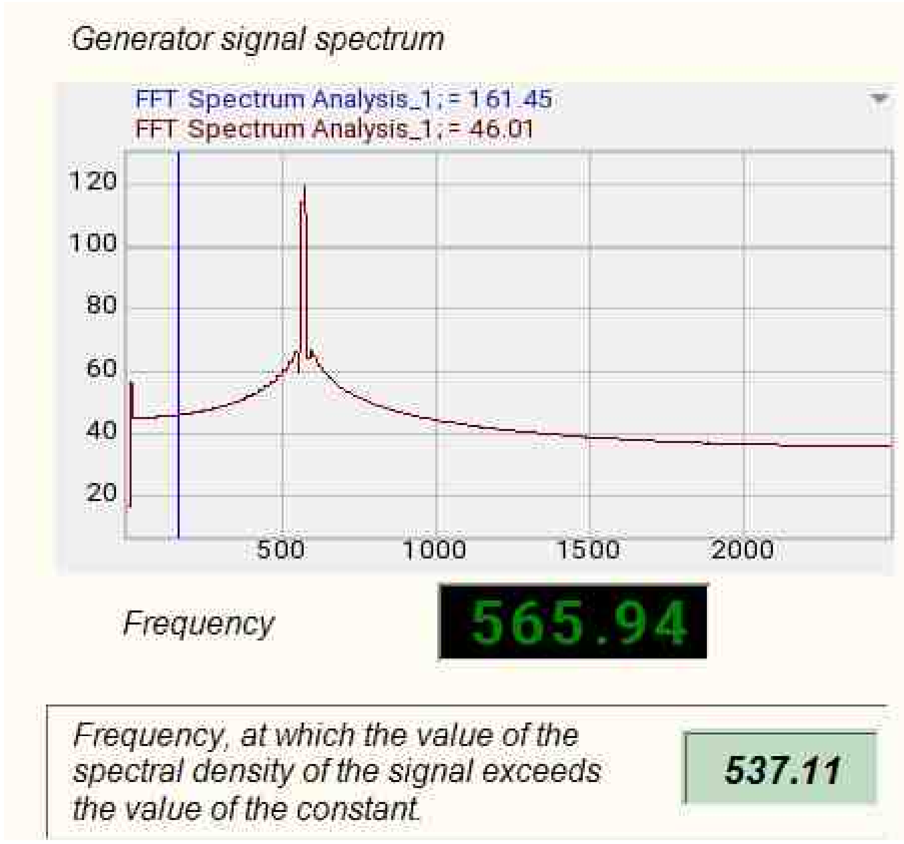
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



This project shows the operation of the Constant[803] component is used to set the value of the level with which all values of the spectral density of the signal at different frequencies will be compared. If this level is exceeded, the value of the frequency at which this occurred will be displayed on the Digital Indicator[682]. A Sine signal[530] is needed to generate a harmonic signal. The FFT Spectrum Analysis[260] is used to obtain the signal spectrum. The Graphic[909] is necessary to display the dependence of the spectral density on the frequency on the display. The Index of excess the array[807] is needed to find the frequency at which the spectral density value will be exceeded, the Value of the array element[803] is necessary in order to get a specific frequency value from the array at which the excess will be performed.

Project operation result

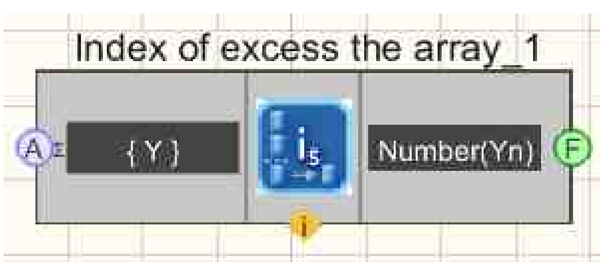


View the example in ZetView

17.3.Excess index

The component "**Excess index**" is intended for obtaining the index of a number whose value exceeds the specified value.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- {Y } - array input value.

➢ Output:

- Number(Yn) - the index of the element is received, the value of which exceeds the specified one.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- ReperLev (0) - the level of the reference line when searching for the excess index.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - Array values that will be taken to check the overage index.

➢**Custom common properties:**

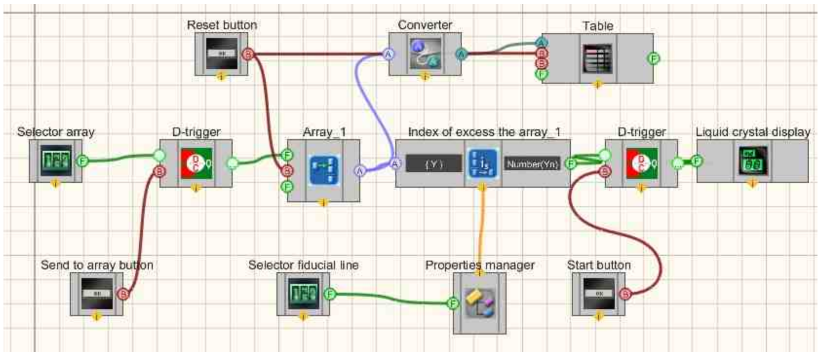
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT ReperLev - Set the level of the reference line when searching for the excess index (number).

Example

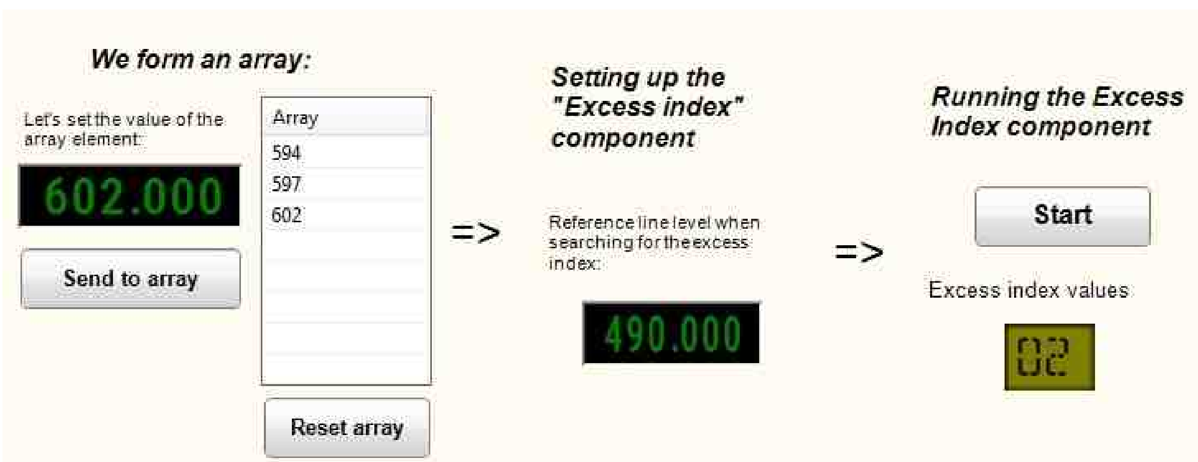
Project in SCADA ZETView



This project shows the operation of the Excess index 807 component. The {Y} input of the Excess index 807 component receives an Formation of arrays (array - array) 796 component. The elements of the array are formed using the Selector 428, on which we can manually set the number we need, then the number from the Selector 428 is sent to the D-trigger 732. With the non-latching Send to Array button, we instruct the trigger to pass the value on the Selector 428, to the Formation of arrays (array - array) 796. The Reset button, also a Non-fixed button 694, resets the Formation of arrays (array - array) 796, in case you need to rebuild the array. Next, the Excess index 807 component adds up the elements of the resulting array and passes it to the output Number(Yn). After that, the number goes to the Digital Indicator 682.

For greater clarity of the circuit, a Converter is also connected to it Converters (numeric array - string array) 713 and Data table 934, which allow us to record the array we enter into a table.

Project operation result



View the example in ZetView

17.4.Maximum level

The component "**Maximum level**" is designed to get the maximum value of the input array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array to search maximum level.

➢ Output:

- Number(Yn) - received maximum level.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Maxtype (Value) - when searching for the maximum level in the array: Value; Index in the array (string).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - Array values that will be taken to find the maximum level.

➢**Custom common properties:**

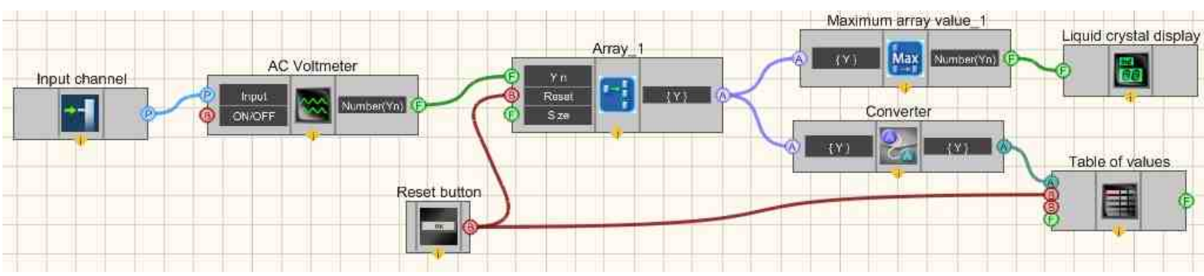
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR MaxType - Set the variant of the desired value when searching for the maximum level in the array: Value; Index in the array (string).

Example

Project in SCADA ZETView

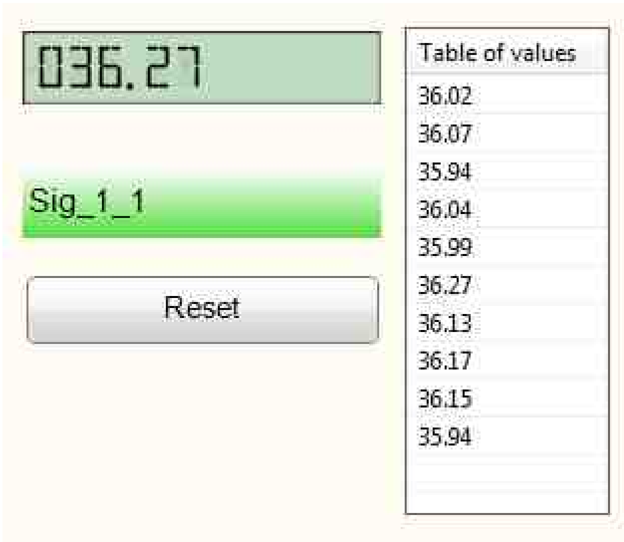


This project shows the operation of the Maximum level value 810 component. The {Y} input of the Maximum level value 810 component receives an Formation of arrays (numeric - array) 796 component with the ArraySize=10 (Array Size) parameter. A Selector 428 is connected to the Formation of arrays (numeric - array) 796, which allows you to manually set the value of an array element, and a reset button, in case you need to rebuild the array. The Maximum level value 810 component, having received 10 array element values, selects the largest one and passes it to the Number(Yn) output. After that, the number goes to the Digital Indicator 682.

For greater clarity of the scheme, a Converters (numeric array - string array) 713 and a Data table 934 are also connected to it, which allow us to record the array we enter into a table.

In order to see the result, it is necessary that a physical ZET device is connected to the computer. In the list of channels, you need to select one of the ZETLab generators.

Project operation result



View the example in ZetView

17.5.Minimum level

The component "**Minimum level**" is designed to get the minimum level of the input array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array to find the minimum level.

➢ Output:

- Number(Yn) - received minimum level.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Mintype(Value) - a variant of the desired value when searching for the minimum level: Value; Index in the array (string).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- {Y} - Array values that will be taken to find the minimum level.

➢Custom common properties:

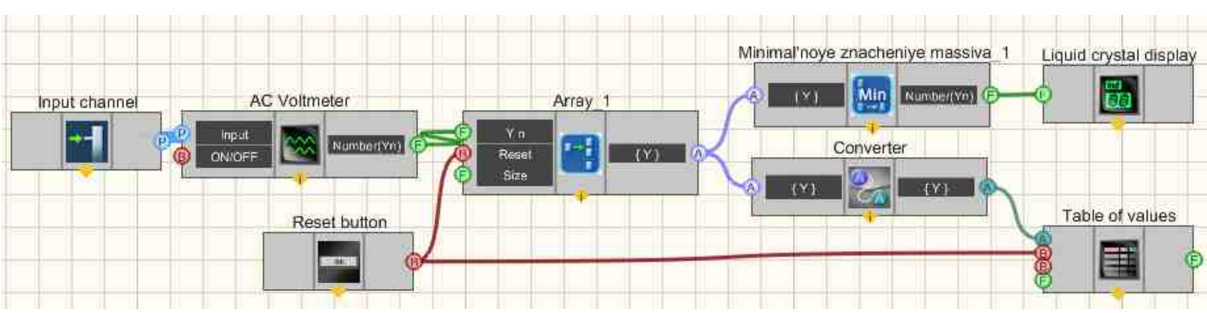
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR MinType - Set the variant of the desired value when searching for the minimum level: Value; Index in the array (string).

Example

Project in SCADA ZETView

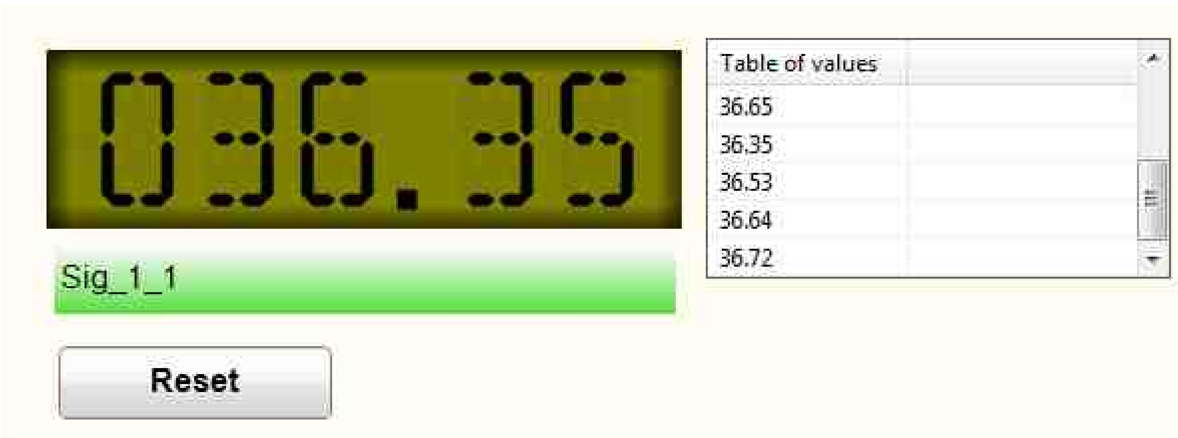


This project shows the operation of the Minimum level value [812] component works. The {Y} input of the Minimum level value [812] component receives an Formation of arrays (numeric - array) [796] component with the ArraySize=10 (Array size) parameter. Connected to the Formation of arrays (numeric - array) [796] Minimum level value [812], which transmits a signal from one of the channels of the ZET device connected to the computer. Reset button, in case you need to rebuild the array. The Minimum level value [812] component, having received 10 values of the array elements, selects the smallest one and passes it to the Number(Yn) output. After that, the number goes to Digital Indicator [682].

For greater clarity of the scheme, a Converters (numeric array - string array) [713] and a Data table [934] are also connected to it, which allow us to record the array we enter into a table.

In order to see the result, it is necessary that a physical ZET device is connected to the computer. In the list of channels, you need to select one of the ZETLab generators.

Project operation result

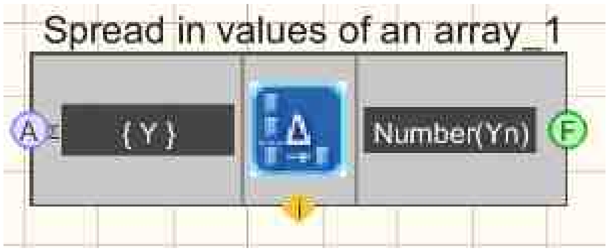


View the example in ZetView

17.6.Spread in values

The component "**Spread of values**" is intended for displaying the difference of values between the maximum and minimum elements of the array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - numeric input value of the array to search for the spread of values between the maximum and minimum.

➢ Output:

- Number(Yn) - spread of values.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

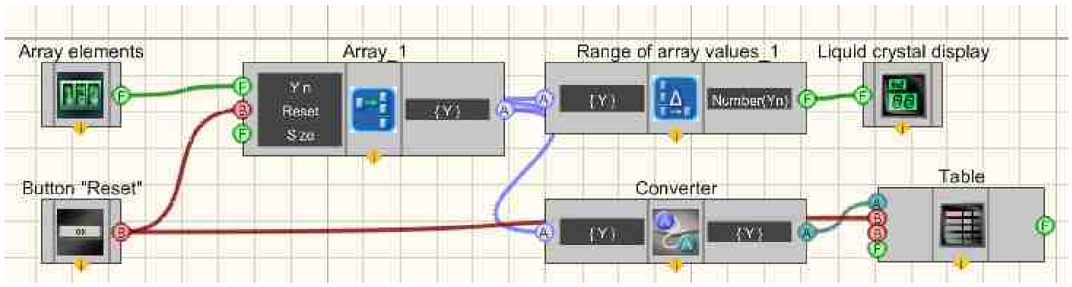
- {Y} - The numeric value of the array that will be taken to determine the spread of values between the maximum and minimum.

➢Custom common properties:

- Are available by the link below. 154

Example

Project in SCADA ZETView

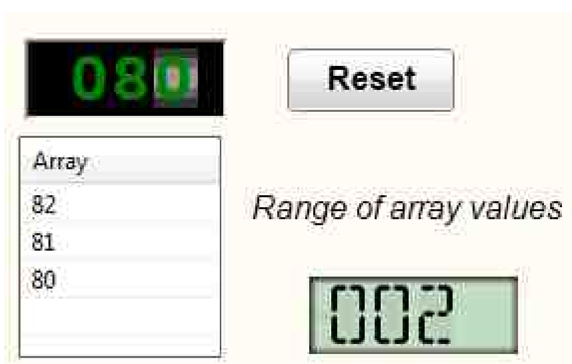


This project shows the operation of the Spread in values 814 component is considered. The Formation of arrays (numeric - array) 796 component by entering values on the Selector 428, the "Reset" Non-fixed button 814 resets the array and values in the Table 934. The Formation of arrays (numeric - array) 796 sends the resulting array to the Spread in values 814 component, which, in turn, calculates the required value and sends the result to the Digital Indicator 682.

For greater clarity of the scheme, a Converters (numeric array - string array) 713 and a Data table 934 are also connected to it, which allow us to record the array we enter into a table.

In order to see the result, it is necessary that a physical ZET device is connected to the computer. In the list of channels, you need to select one of the ZETLab generators.

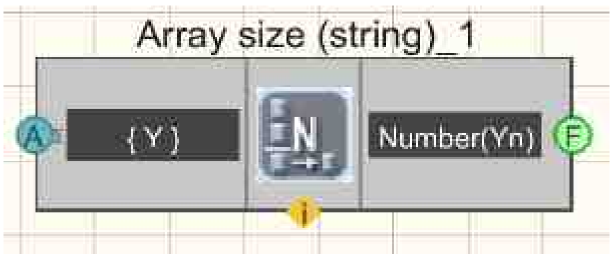
Project operation result



17.7.Array size (string)

The component "**Array size** (string)" is intended for obtaining the size of a string array. Output data - the number of elements in a string array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - input string array.

➢ Output:
- Number(Yn) - the number of elements in the string array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

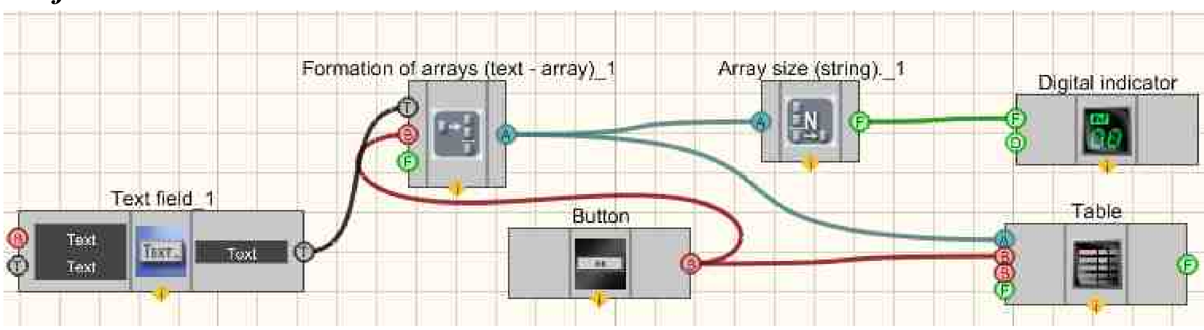
- {Y} - Values of the input string array, to which the array values are supplied, based on which the size of the array will be calculated.

➢**Custom common properties:**

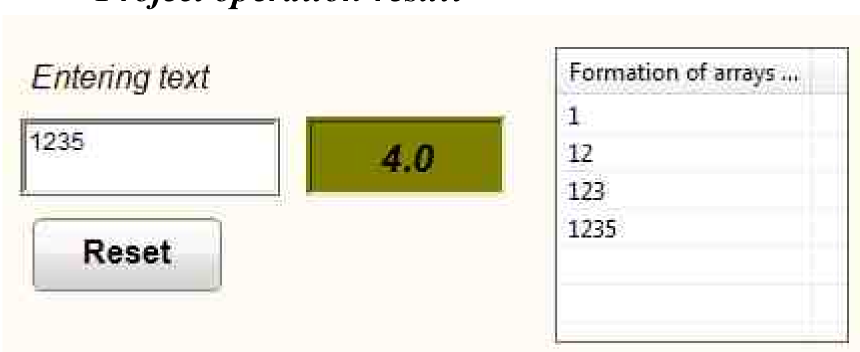
- Are available by the link below. 154

Example

Project in SCADA ZETView



Project operation result



View the example in ZetView

17.8.Array size (numeric)

The component "**Array size (numeric)**" is designed to get the size of a numeric array. Output data - the number of elements in a numeric array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - input numeric array.

➢ Output:
- NumberY(n) - the number of elements in the numeric array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

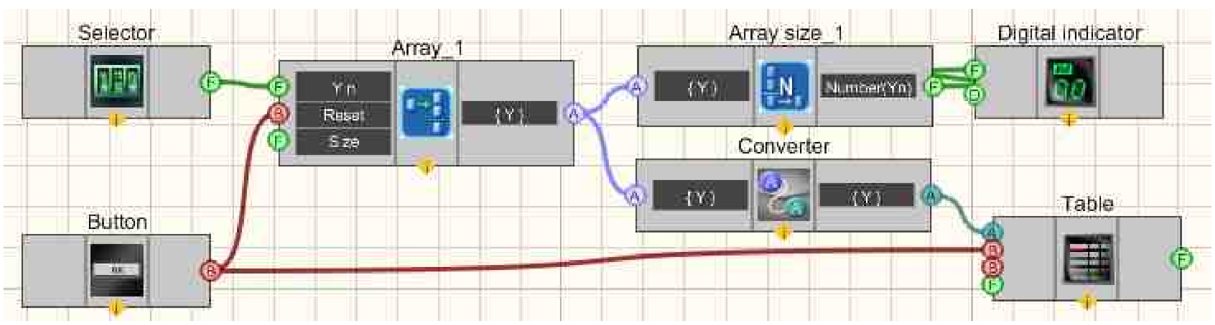
- {Y} - values of the input numeric array, to which the array values are supplied, based on which the size of the array will be calculated.

➢**Custom common properties:**

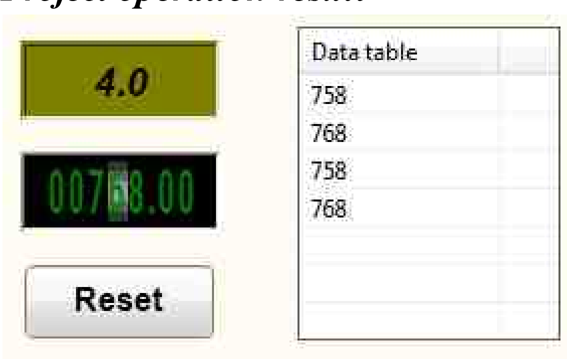
- Are available by the link below. 154

Example

Project in SCADA ZETView



Project operation result

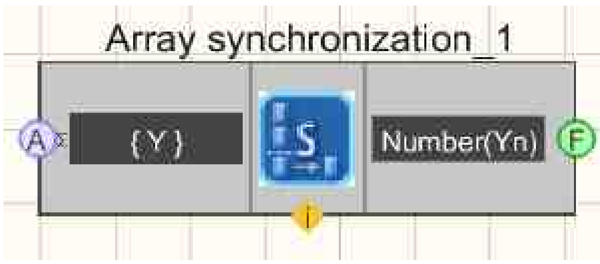


View the example in ZetView

17.9.Array synchronization

The component "**Array synchronization**" is designed for processing arrays with synchronization on the front.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input numeric array.

➢ Output:

- Number(Yn) - array output value.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- synchrotypesynchrotype (On the upward fronts) - the type of synchronization when processing an array using synchronization.
  1. On the upward fronts;
  2. On the downward fronts.
- SynchroLev (0) - the synchronization level when processing an array using synchronization.
- SynchroWid (0) - the width of the sync level when processing an array using sync.
- synchrometh(Impulse width) - a variant of the desired value when processing an array using synchronization:
  1. Impulse width;
  2. Duration of the front;
  3, Number of fronts.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - array numeric values to be synchronized.

➢**Custom common properties:**

- [Are available by the link below.](154)

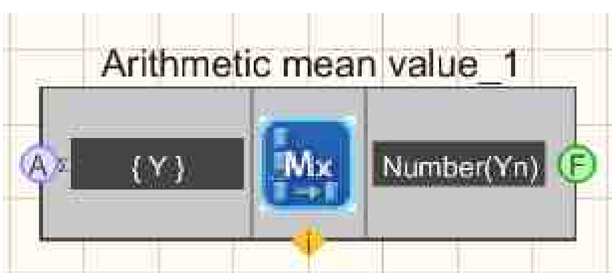
➢**Custom private properties (the default value is shown in parentheses):**

- BSTR SynchroType - Set the type of synchronization when processing an array using synchronization: On the upward fronts; On the downward fronts (string).
- FLOAT SynchroLev - Set the synchronization level when processing an array using synchronization (number).
- FLOAT SynchroWid - Set the width of the sync level when processing an array using sync (number).
- BSTR SynchroMeth - Set the variant of the desired value when processing an array using synchronization: Impulse width; Duration of the front; Number of fronts (string).

17.10.Arithmetic mean value

The component "**Arithmetic mean value**" is calculates the arithmetic mean of the elements of the input array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - input array.

➢ Output:
- Number(Yn) - the resulting arithmetic mean of the elements of the input array.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

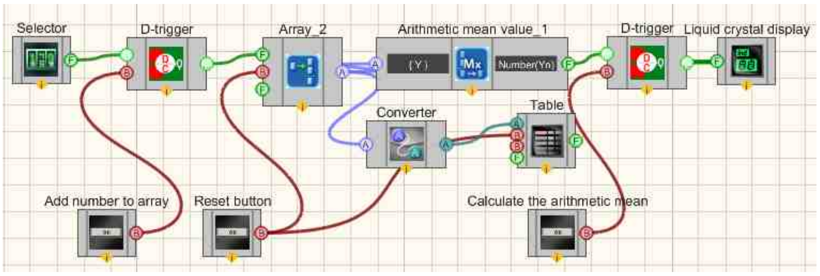
Setting:

➢**Input:**
- {Y} - values of the input array, to which the array values are supplied, based on which the arithmetic mean

➢**Custom common properties:**
- Are available by the link below. 154

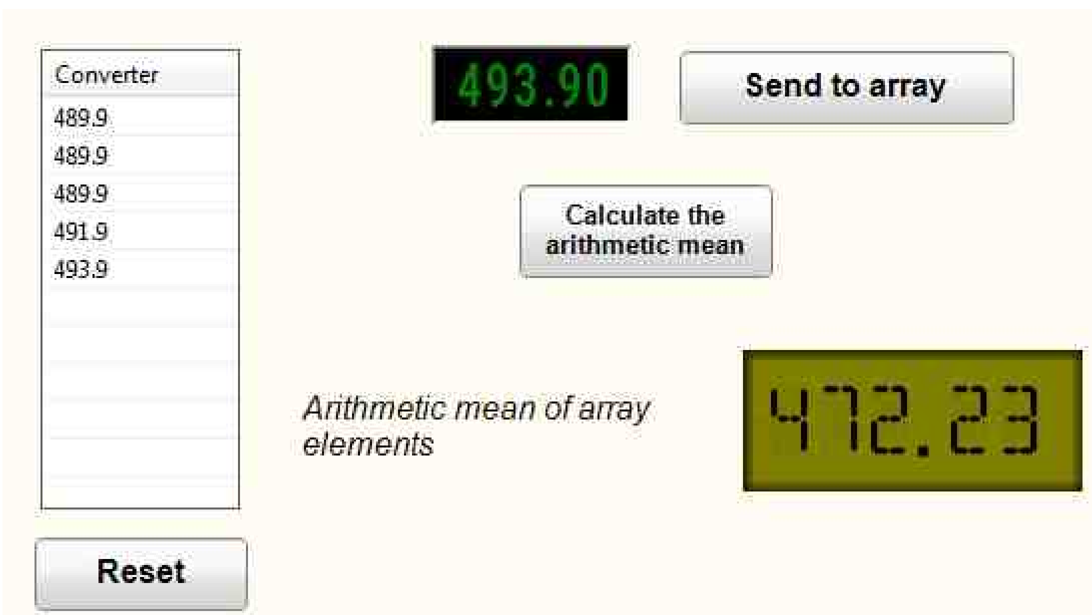
Example

Project in SCADA ZETView



This project shows the operation of the Arithmetic mean value 822 component. The array is formed as follows:
the value we need is set on the Selector 428 and sent to the D-trigger 732. The Non-fixed button 822 Add number to array button, when pressed, sends a control impulse to the D-trigger 732, allowing you to pass the value received from the Selector 428 to the Formation of arrays (numeric - array) 796. Next, the array goes to the component Arithmetic mean value 822, at the output of which we get a number - the Arithmetic mean value 822, which then goes to the LCD indicator. The components Data table 934 and Converters (numeric array - string array) output an array to a table.

Project operation result



View the example in ZetView

Mathematical description

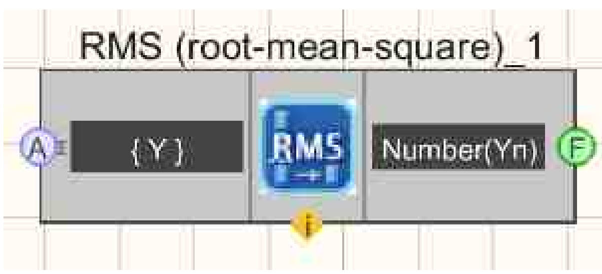
The arithmetic mean of a set of numbers is the sum of all the numbers in that set divided by their number.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}(x_1 + \cdots + x_n).$$

17.11.RMS (root-mean-square)

The component "**RMS (root-mean-square)**" is designed to obtain the mean-square value of array elements, obtained by determining the average value of the set of squared quantities.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- Number(Yn) - the result root mean square value of the elements of the input array.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

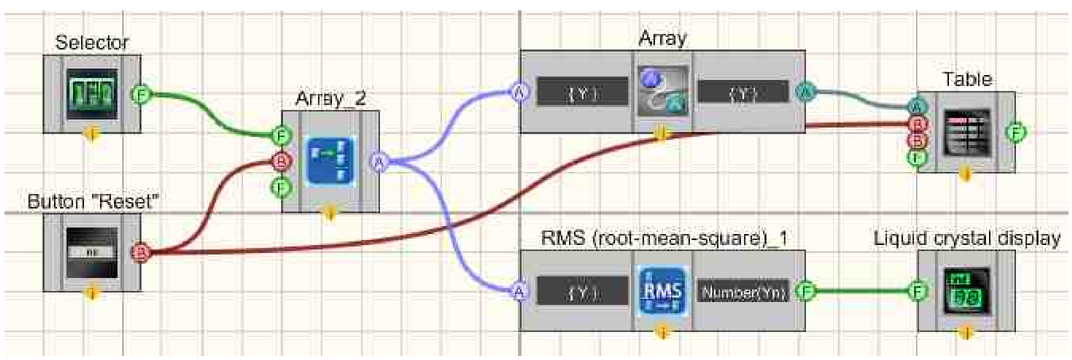
- {Y} - the values of the input array to which the array values are fed, based on which the RMS value will be calculated.

➤**Custom common properties:**

- Are available by the link below. 154

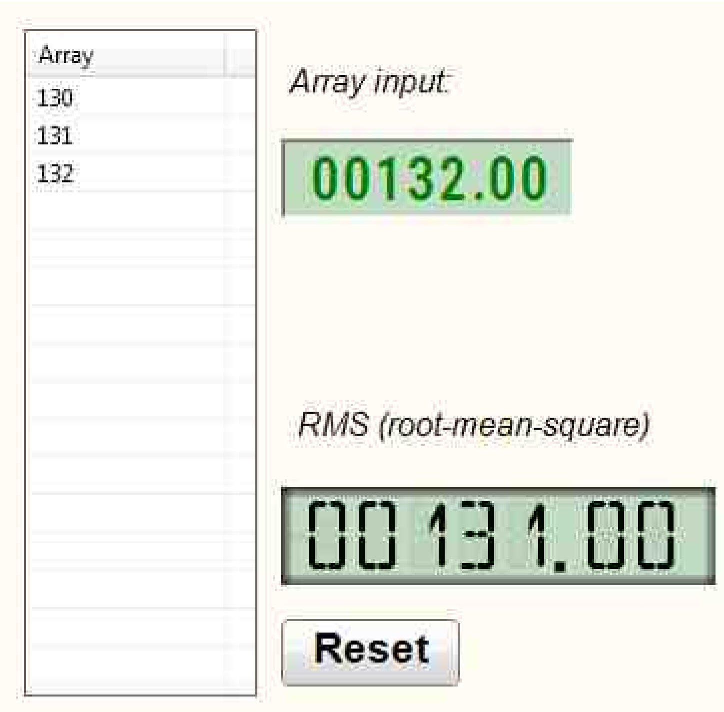
Example

Project in SCADA ZETView



This project shows the operation of the RMS (root-mean-square) 825 component. The array is formed by the user using a Selector 428, from which the array is sent to the Converters (numeric array - string array) 713 where it is converted from a numeric array to a string array and then sent to the Data table 934 (so that the user can see what values he enters into the array). Also, from the Array Shaper (number-array), the array goes to the RMS (root-mean-square) 825 component, and the result is displayed on the Liquid crystal display 655. Non-fixed button 694 "Reset" connected to Converters (numeric array - string array) 713 and Data table 934 resets the array values and clears the table.

Project operation result

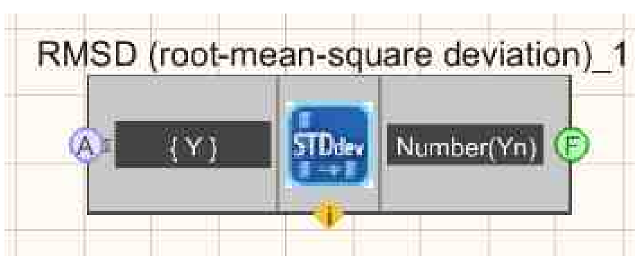


View the example in ZetView

17.12.RMSD (root-mean-square deviation)

The component "**RMSD (root-mean-square deviation)**" is intended for obtaining the value of root-mean-square deviation of array elements.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - array values, based on which the standard deviation will be calculated.

➢ Output:

- Number(Yn) - the value of the standard deviation of the elements is received.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

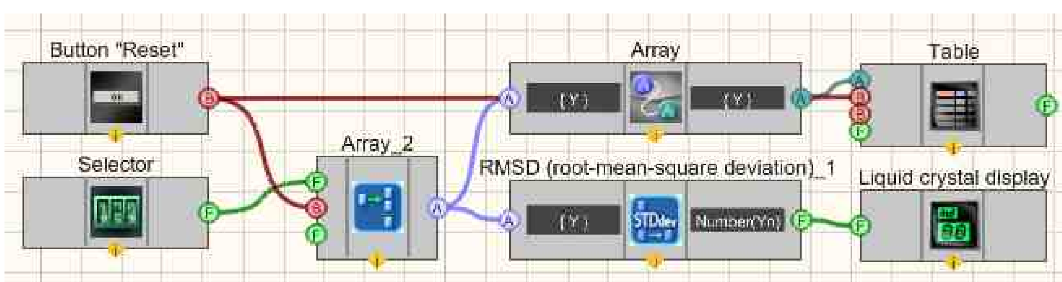
Setting:

➢**Input:**
- {Y} - values of the input array, to which the array values are fed, based on which the standard deviation will be calculated.

➢**Custom common properties:**
- Are available by the link below. 154

Example

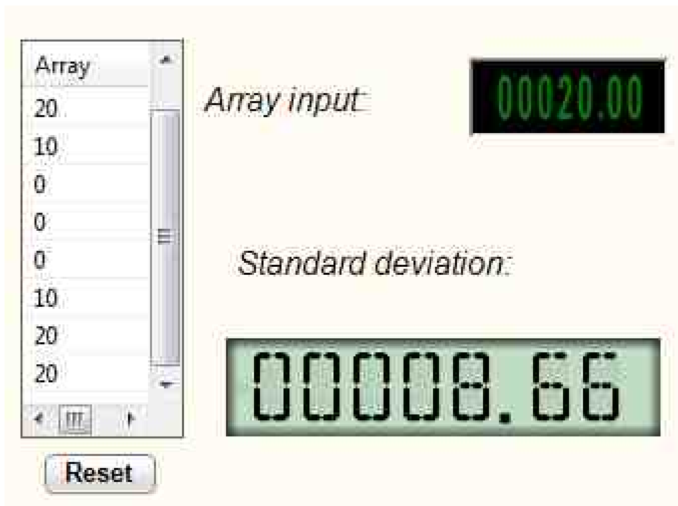
Project in SCADA ZETView



This project shows the operation of the RMSD (root-mean-square deviation) 827 component.

The array is formed by the user using a Selector 428, from which the array is sent to the Converters (numeric array - string array) 713 where it is converted from a numeric array to a string array and then sent to the Data table 934 (so that the user can see what values he enters into the array). Also, from the Formation of arrays (numeric - array) 796, the array goes to the RMSD (root-mean-square deviation) 827 component, and the result is displayed on the Liquid crystal display 655. Non-fixed button 694 "Reset" connected to Converters (numeric array - string array) 713 and Data table 934 resets the array values and clears the table.

Project operation result



View the example in ZetView

Mathematical description

The standard deviation is equal to the square root of the variance of the random variable. The standard deviation is used when calculating the standard error of the arithmetic mean, when constructing confidence intervals, when statistically testing hypotheses, when measuring a linear relationship between random variables.

RMSD (root-mean-square deviation):

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2};$$

The RMSD (root-mean-square deviation) component calculates the value of the root-mean-square deviation using
standard deviation (unbiased estimate of the standard deviation of a random variable x relative to its mathematical expectation):

$$s = \sqrt{\frac{n}{n-1}\sigma^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2};$$

where σ — dispersion; x_i — i-th element of the sample; n — sample size; $\bar{x}$ — arithmetic mean of the sample:

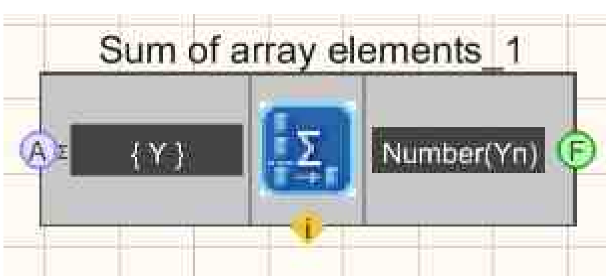
$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}(x_1 + \cdots + x_n).$$

It is measured in units of measurement of the random variable itself.

17.13.Sum of array elements

The component "**Sum of array elements**" is calculates the sum of all elements of the array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - the values of the array to be summed.

➢ Output:

- Number(Yn) - Sum of array elements.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

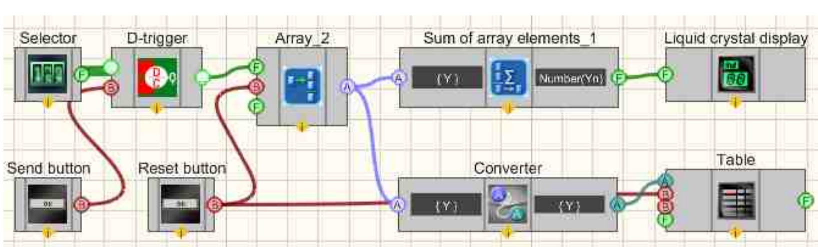
- {Y} - the values of array values to be summed..

➢Custom common properties:

- Are available by the link below. 154

Example

Project in SCADA ZETView

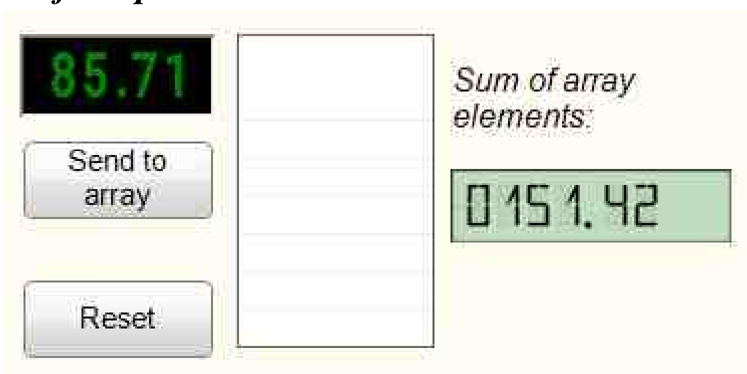


This project shows the operation of the Sum of array elements 830 component.

The input {Y} of the Sum of array elements[830] component receives an array Formation of arrays (numeric - array)[796] component. The elements of the array are formed using the Selector[428], on which we can manually set the number we need, then the number from the selector is sent to the D-trigger[732]. With the non-latching Send to Array button, the trigger is instructed to pass the value on the Selector[428] to the Formation of arrays (numeric - array)[796]. The reset button, also a Non-fixed button[694], resets the Formation of arrays (numeric - array)[796], in case you need to rebuild the array. Next, the Sum of array elements[830] component adds up the elements of the resulting array and passes it to the output Number(Yn). After that, the number goes to the Digital Indicator[682].

For greater clarity of the circuit, a Converter is also connected to it Converters (numeric array - string array)[713] and Data table[934], which allow us to record the array we enter into a table.

Project operation result



View the example in ZetView

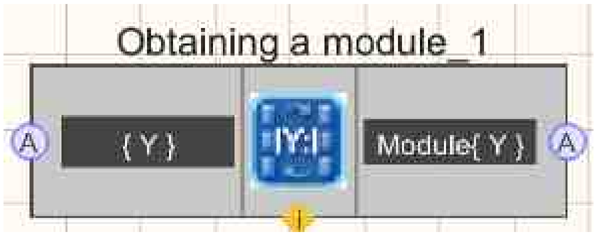
Глава 18.Array -Array

Components of "Array-value" and "Array-array" section are used for processing of data arrays.
Formation of arrays is used a get arrays from the values sent to the input of the component.
Arrays handler is used for arrays processing (sorting, filtration and synchronization of data).

18.1.Obtaining a module

The component "**Obtaining a module**" is intended for obtaining a module of elements of an input array;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- { Y } - input array.

➤ Output:
- Module { Y } - an array is received, the elements of which are equal to the module of the elements of the input array.

Custom common properties:

➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

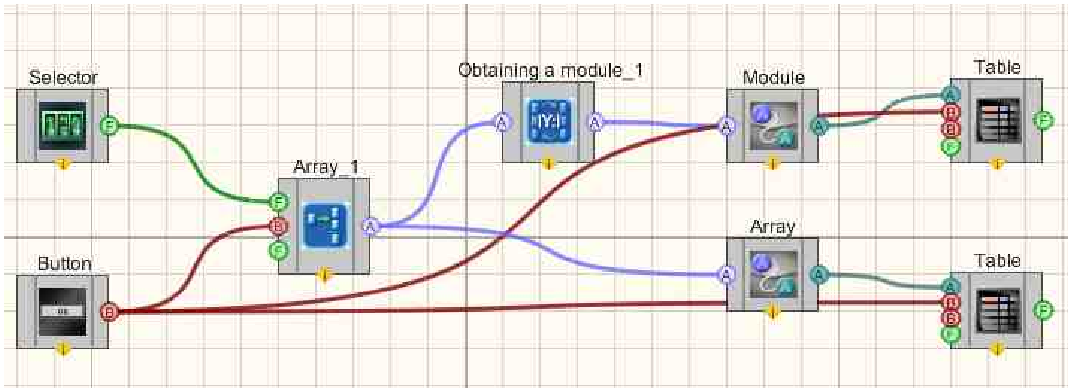
➤**Input:**
- {Y} - Values of the input array, to which the array values are supplied, based on which the module will be calculated.

➤**Custom Custom private properties (the default value is shown in parentheses):**

Are available by the link below. 154

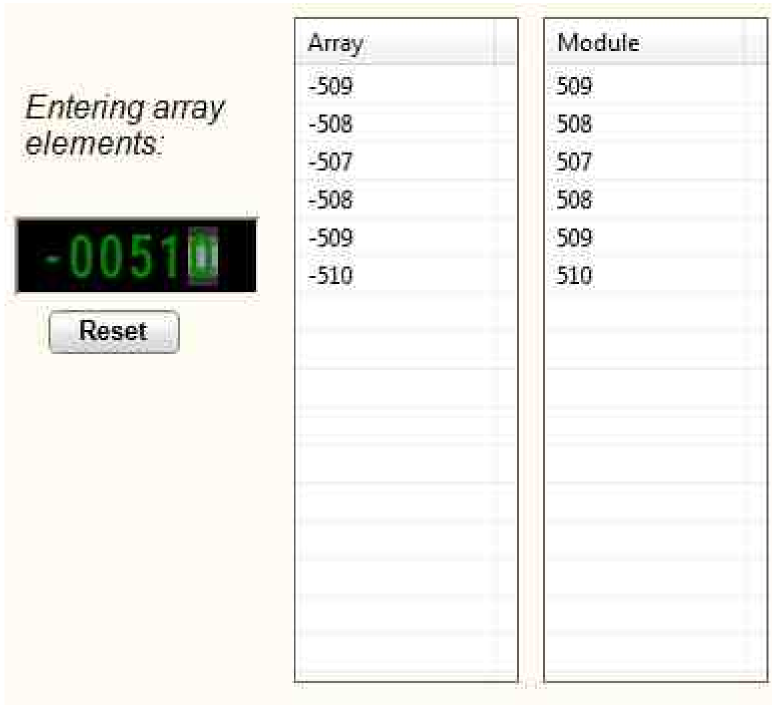
Example

Project in SCADA ZETView



This project shows the operation of the module Obtaining a module 832.
The array is formed by the user using the Selector 428 and the Formation of arrays (numeric - array) 796 from which it goes to the Obtaining a module 832 component and to the Converters (numeric array - string array) 713 + Data table 934. At the output from the Obtaining a module 832 component, the result is also sent to the Converters (numeric array - string array) 713 + Data table 934. Therefore, we can compare the initial array and the result after the work of the Obtaining a module 832 component when the program is started. Non-fixed button 694 "Reset" connected to Converters (numeric array - string array) 713 and Data table 934 resets the array values and clears the table.

Project operation result

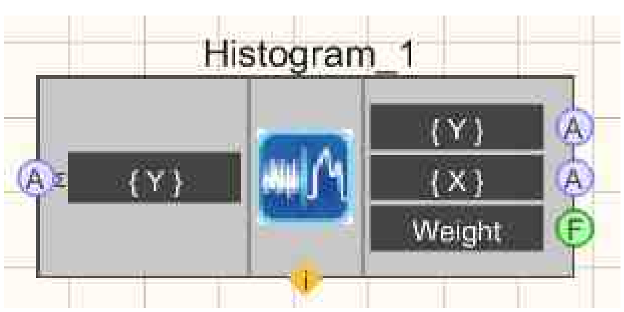


View the example in ZetView

18.2.Histogram

The component "**Histogram**" is intended for calculation of an output array of values for plotting a histogram.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - input array.

➢ Output:
- { Y } - output array.
- { X } - array dimension.
- Weight - LSB (least significant bit).

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Start (0) - set the start value of the histogram calculation.
- Finish (10) - set the end value of histogram calculation.
- Quantity (10) - set the number of histogram calculation bands.
- Norm (false) - set the enable normalization.
- BitADC (16) - ADC bit depth: 16; 24.
- Uref (7620.0) - ADC reference voltage, mV.
- Mode (Mode) - set the operating mode: Hist & LSB; Hist; LSB.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- {Y} - numeric values of the array to which the input array should be fed.

➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

•FLOAT Start - Set the start value of the histogram calculation (number).
•FLOAT Finish - Set the end value of the histogram calculation (number).

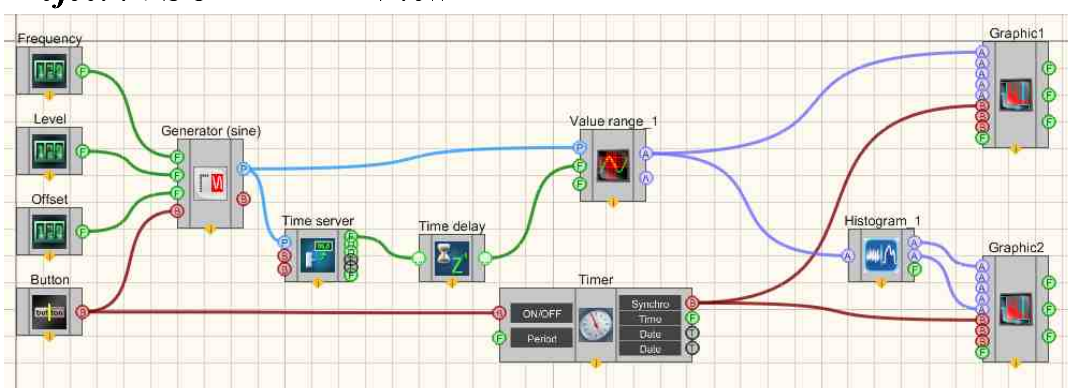
- LONG Quantity - Set the number of histogram calculation bands (number).
- VARIANT_BOOL Norm - Set the array normalization permission:
  - true - enable array normalization;
  - false - enabled array normalization disabled.

- BSTR BitADC - Set the bit depth of the ADC: 16; 24 (string).
- FLOAT Uref (7620.0) - Set the ADC reference voltage, mV.
- BSTR Mode (Hist & LSB) - operation mode: Hist & LSB; Hist ; LSB(string).

Example

For the Examples to work, it is necessary to have a connected physical ZET device.

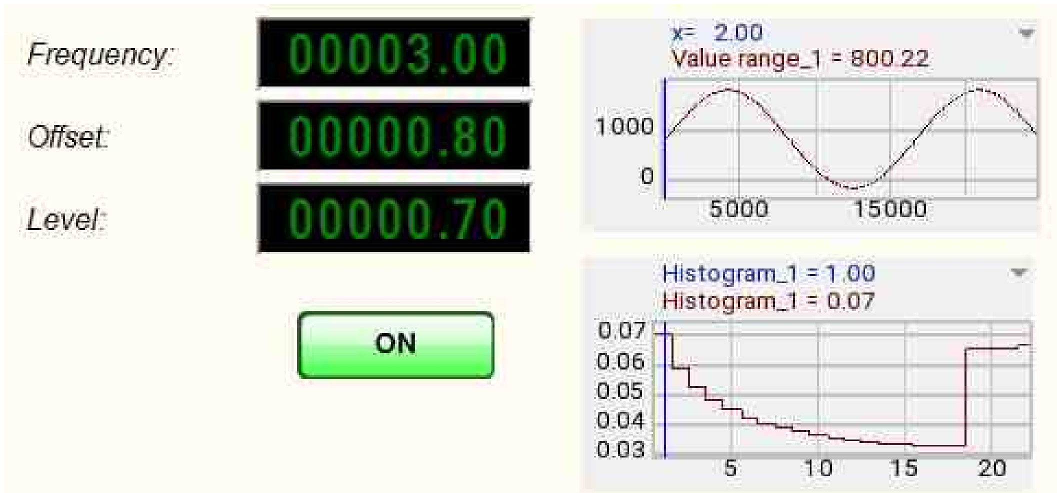
Project in SCADA ZETView



The example shows how the Histogram 835 component operations. After adjusting the input signal from the generator, the data is displayed on two graphs: a signal graphic and a histogram graph. Set the generator Generator (sine) 530 is performed using three Selectors 428 that allow you to adjust the frequency, level and offset, respectively. The signal from the generator goes to the Range of values 594 component, which allows you to measure the instantaneous values of the signal in a specified period of time. The value of the time interval component Range of values 594 receives on the input contact from the component Server Time 366, which is specially adapted for this task. You can notice that the Time delay 363 component is connected to the Server Time 366 component - it is not necessary, but it is desirable to use it so that there is no desynchronization on the Range of values 594 component between the received instantaneous signal values and the server time. Further, the Range of values 594 component generates two arrays at the output: an array of signal values and an array of indexes. The resulting array is displayed on the graphic - thus, the signal Graphic 909 is obtained. Also, data from the Range of values 594 is sent to the histogram component, where an array of histogram values and an array of indexes are formed. From the Histogram 835 the array also gets to Graphic2 909

- this is how the histogram graphic is obtained. Graphics are synchronized using a Timer370.

Project operation result



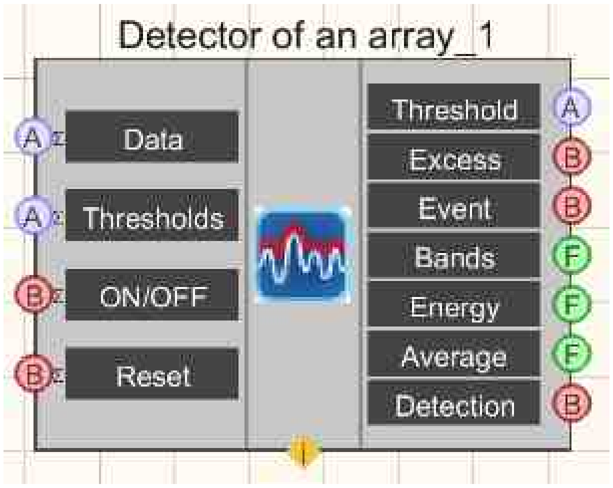
View the example in ZetView

18.3.Detector of an array

The component "**Detector of an array**" - The "**Threshold exceeding threshold**" component is used to determine whether the threshold values exceeded for some time in the past have been exceeded, as well as the calculation of the exceedance characteristics.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- Data - input array of regular data as arrays.
- Thresholds - input array of threshold values.
  - ON/OFF - set the sed to enable/disable the component.
- Reset - reset the current threshold values and start calculating new ones.

➢ Output:
- Threshold - outputs threshold values as an array.
- Excee - set the designed to display a flag about the presence of exceeding the thresholds.
- Event - set the designed to display a flag about the presence or absence of an event.
- Bands - set the designed to display the number of bands in which there is an excess of threshold values.
- Energy - set the outputs the energy of the signal in the bands where the excess is observed.
- Average - set the designed to display the arithmetic mean value of energy from the moment the threshold is exceeded until the message about the presence of an event is issued.
- Detection - set the designed to display the current status of the component: accumulation and calculation of threshold values or tracking the current values by thresholds.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- ArrayNumbers (100) - set the number of input arrays used for threshold calculation.
- Coefficient (2.000000) - set the type of averaging when calculating threshold values: maximum or arithmetic mean.

- ThresholdType (Maximum) - set the coefficient of influence of dispersion for calculating threshold values.
- WaitNumber (40) - set the number of exceedances required to confidently determine the event.
- ResetNumber (6) - set the number of arrays without exceeding those required to reset the event.
- MinExcess (72692592) - set the number of excesses in the bands for issuing an impulse from the "Excess" contact (in percent).

Mathematical description

The "On/Off" contact should turn the component on and off. When a component is disabled, it ignores all input except for signals to itself and a signal to the "Reset" pin. The output contacts should also be "silent" when the component is switched off. The current status of internal variables must remain unchanged until the component is enabled.

The "Reset" contact must reset the current threshold value, reset the array counter. The "Detect" contact should give a single FALSE pulse.

The "Data" contact must accept input arrays. The size of the input arrays must be determined by the size of the first input array. The following arrays must be the same size or be converted (trimmed or padded with zeros).

During operation, a component can be in two states: accumulation and discovery. During the operation of the component in the accumulation mode, one calculated array must be calculated from all input arrays according to the type of averaging - either as the maximum in bands, or as the arithmetic average. Regardless of the type of averaging, the component must calculate the variance of the distribution of values in the bands. The array of thresholds must be continuously calculated using the formula:

for averaging type "maximum"

THRESHOLD[i] = MAX[i] + RATIO[i]

for averaging type "average"

THRESHOLD[i] = AVERAGE[i] + RATIO[i]

The "Threshold" contact must constantly produce a newly calculated array of threshold values. The "Event" contact should always return "FALSE". Contacts "Strips" and "Energy" must constantly give the number "0".

After accumulating a certain number of arrays specified in the component property, the component should go into the detection state, and the "Detection" contact should emit a single impulse TRUE.

While operating in the detection mode, the component must compare each incoming array with an array of threshold values. If all elements of the incoming array do not exceed the threshold values, then a FALSE impulse should be generated from the "Excess" output, and the number "0" should be output from the "Bands" and "Energy" outputs. If at least one element exceeded the threshold, then a TRUE impulse should be generated from the "Excess" contact, the number of bands in which the excess is observed should be output from the "Bands" contact, and an "energy" value should be output from the "Energy" contact excess. The "energy" value is calculated as the ratio of the sum of the squares of the incoming array values to the sum of the squares of the threshold array values, calculated only for the bands in which an excess is observed.

According to the status of the "Excess" contact, the value of the "Event" contact should be calculated. If a TRUE impulse appears on the "Excess" pin, then the component must turn on the counter and count the input arrays. If the counter exceeds the value of the "Event waiting time" property, then a TRUE impulse must be sent from the "Event" output. As soon as the FALSE impulse appears on the "Excess" contact and lasts for the time specified by the "Event interruption time" property, then a FALSE impulse must be sent from the "Event" output.

After starting the counter, the component must sum the energy values for the input arrays that exceed the threshold value in the specified number of bands. After issuing an impulse from the "Event" contact, the component must average the accumulated energy value and output it from the "Average" contact.

Changing the properties while the component is running, that is, when it is enabled, is not allowed, except for the dispersion excess factor. When changing the coefficient of excess dispersion, the array of threshold values is immediately recalculated and output. The remaining properties are remembered and applied only after the next activation of the component.

If the "Thresholds" input receives an array with non-zero values, then the component should automatically apply this array as a threshold array for its further work. If at this time the component was in the accumulation state, then it should immediately switch to the discovery state. Changing the dispersion influence coefficient should not affect the operation of the component in any way. If the "Thresholds" input receives an array with zero elements, then the component must reset its current thresholds array and switch to the accumulation mode. A positive impulse at the "Reset" input should also put the component into the accumulation mode, regardless of how many threshold arrays were applied to the input of the component.

The algorithm of the methods

1. The method must save the values of the last incoming array and the current threshold array to the file DTU/
2. The method must set a number - a counter of the number of arrays that will be ignored when they come to the input and should not affect the operation of the component in any way. Reset the component should reset the counter. When a method is called with a non-zero value, a FALSE impulse should be sent from the Detect output to indicate that the component is not processing input. When the counter reaches 0, a TRUE impulse must be sent from the "Detection" output, signaling the start of processing the input data by the component.

.

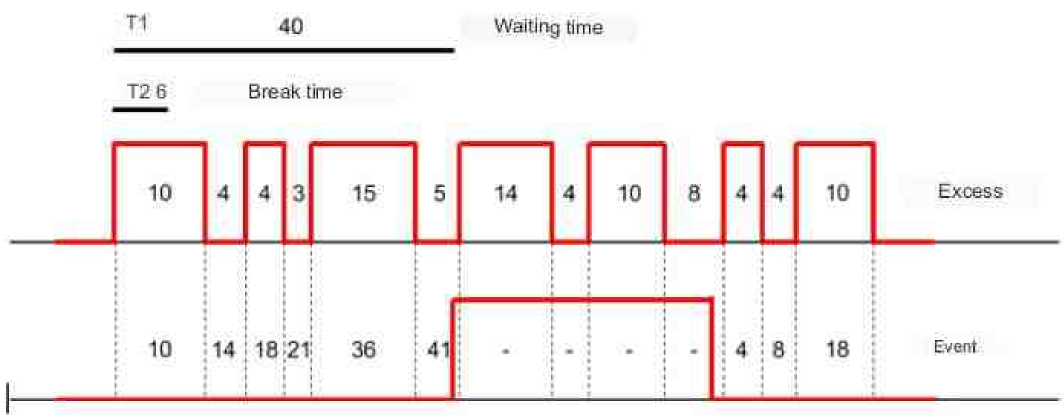
Note.

Sco - standard deviation is calculated through the dispersion according to the formula:

$$\sigma = \sqrt{D} = \sqrt{M[X - M(X)]^2}$$

https://en.wikipedia.org/wiki/%C4%E8%F1%EF%E5%F0%F1%E8%FF_%F1%EB%F3%F7%E0%E9%ED%EE%E9_%E2 %E5%EB%E8%F7%E8%ED%FB

Event definition example:



Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Data - Values of the input array, to which the next data is entered in the form of arrays.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•LONG ArrayNumbers - Set the number of arrays needed to calculate the threshold (number).
•FLOAT Coefficient - Set the coefficient of influence of dispersion on threshold values (number).

- BSTR ThresholdType - Set the type of threshold calculation: Maximum; Medium (string).
- LONG WaitNumber - Set the number of times it takes to confidently determine the event (number).
- LONG ResetNumber - Set the number of arrays without exceeding those required to reset the event (number).
- LONG MinExcess - Set the minimum number of excesses in the bands for issuing an impulse from the "Excess" contact (in percent) (number).

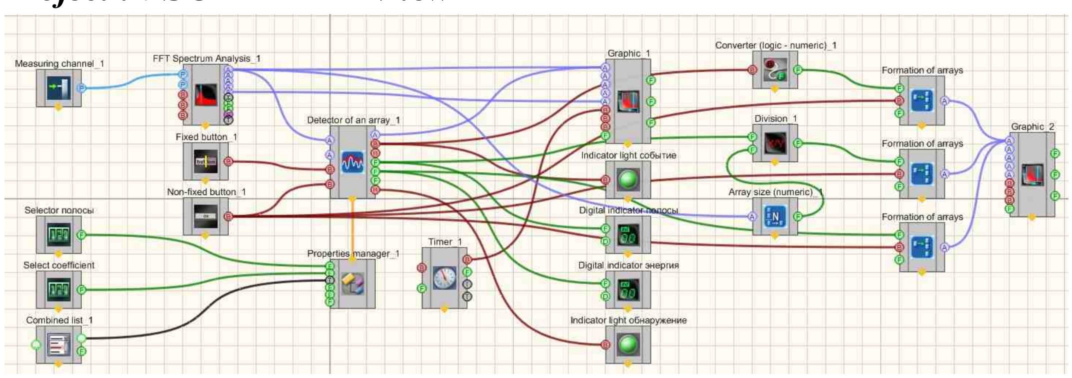
➢**Methods:**

- void SaveArraysToFile(BSTR FileName) - Save the current arrays to a file.
- LONG SetToleranceNumber(LONG number) - Set the number of input arrays to skip.

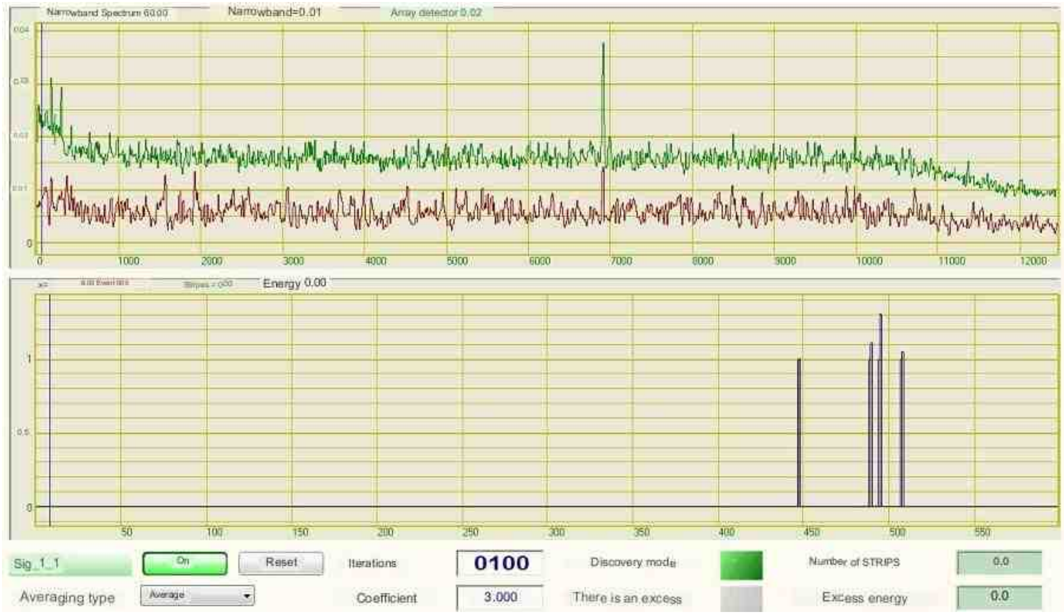
Example

For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



Project operation result



View the example in ZetView

18.4.Differentiation array

The component "**Differentiation array**" is differentiates the array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- A{ Y } - a differentiated array arrives

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

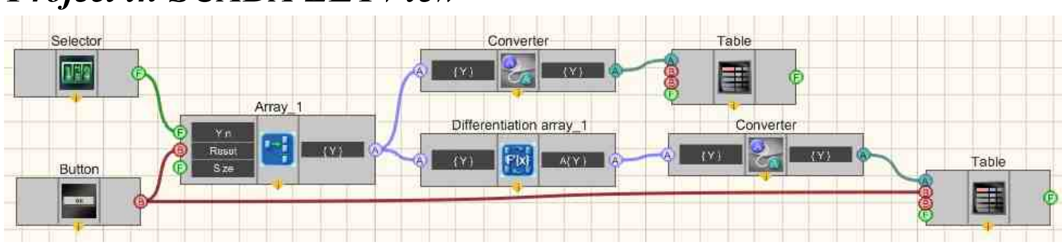
- {Y} - Values of the input array, to which the array values are fed, based on which the array will be differentiated.

➢**Custom common properties:**

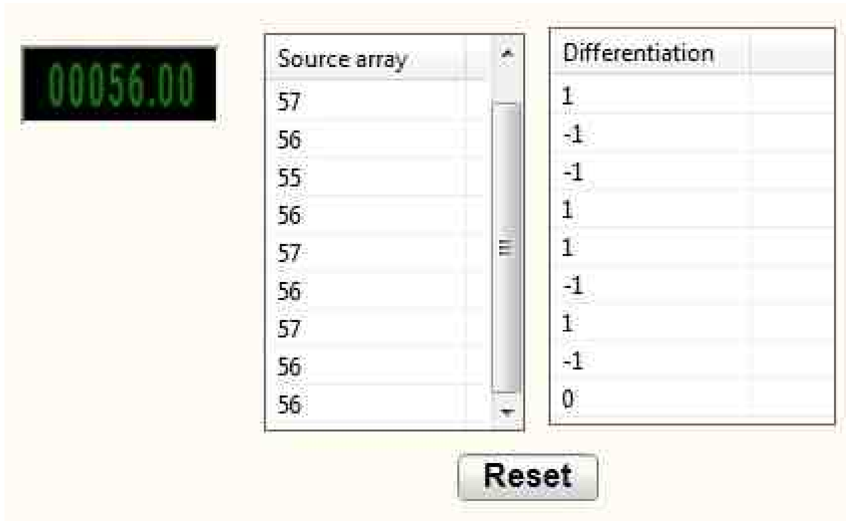
- Are available by the link below. 154

Example

Project in SCADA ZETView



Project operation result

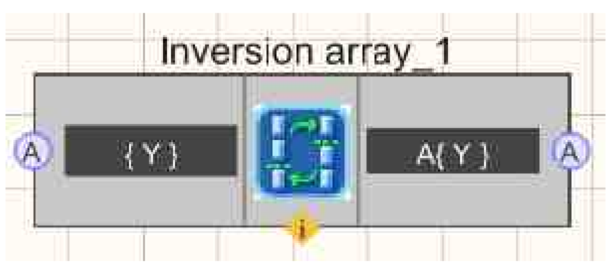


View the example in ZetView

18.5.Inversion array

The component "**Inversion array**" is inverts the input array. Inverts the input array by reversing the order of elements in a one-dimensional Array or in a portion of an Array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - input array.

➢ Output:
- A{ Y } - an inverse array arrives.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

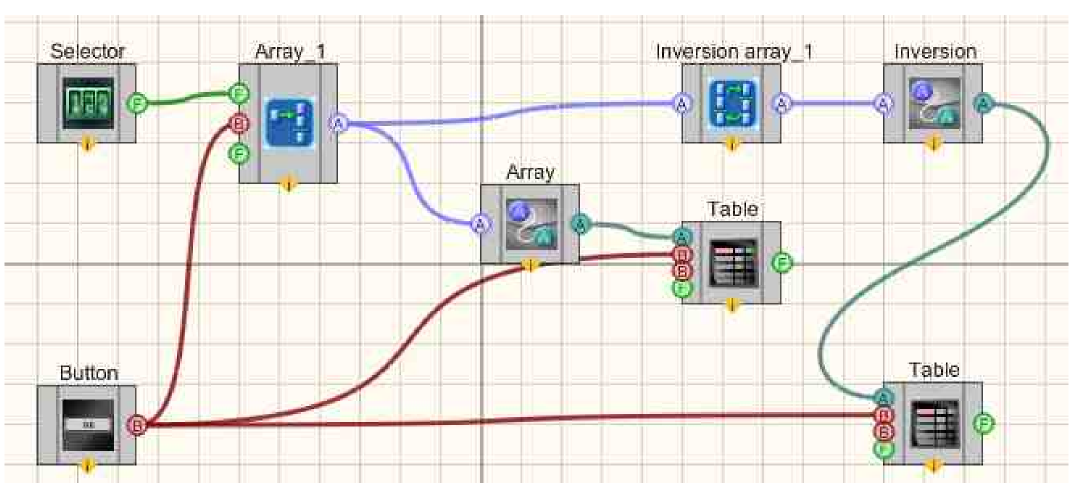
- {Y} - Values of the input array, to which the array values are supplied, based on which the array inversion will be calculated.

➢**Custom common properties**:

- Are available by the link below. 154

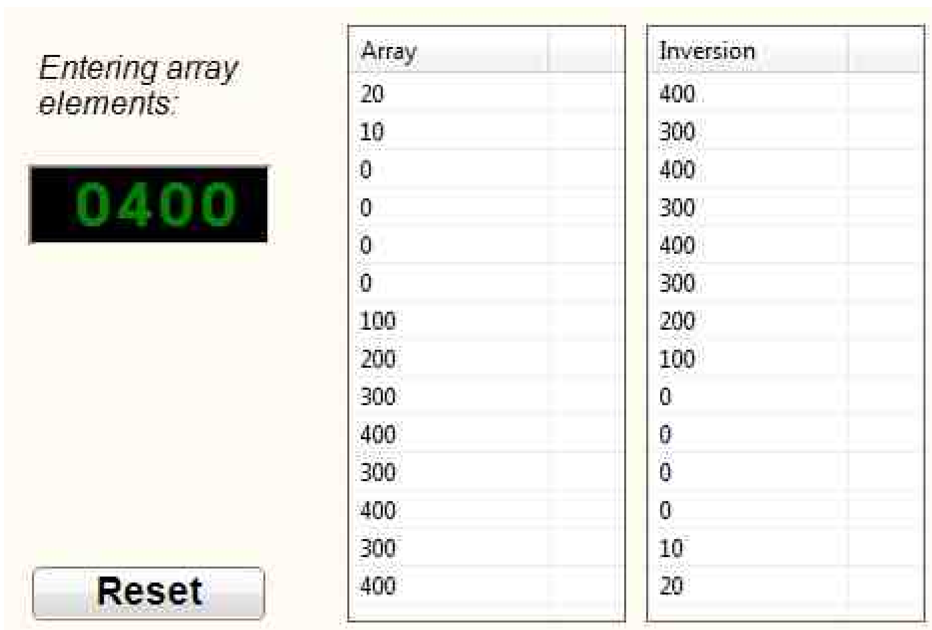
Example

Project in SCADA ZETView



This project shows the operation of the Inversion array[846] component
The array is formed by the user using a Selector[428] and the Formation of arrays (numeric - array)[796] from which it goes to the Inversion array[846] component and to the Converters (numeric array - string array)[713] + Data table[934]. At the output from the Inversion array[846] component, the result is also sent to the Converters (numeric array - string array)[713] + Data table[934]. Therefore, we can compare the initial array and the result after the work of the Inversion array[846] component when the program is started. Non-fixed button[694] "Reset" connected to Converters (numeric array - string array)[713] and Data table[934] resets the array values and clears the table.

Project operation result



View the example in ZetView

18.6.Integration of an array

The component "**Integration of an array**" is integrates the input array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- { Y } - input array.

➤ Output:

- A{ Y } - an integrated array arrives.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

- dTime (1) - set the interval between readings, s.
- MeanSubstruction (false) - set the subtraction of the constant component.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤Input:

- {Y} - set the numeric values of the array to which the input array should be fed for integration.

➤Custom common properties:

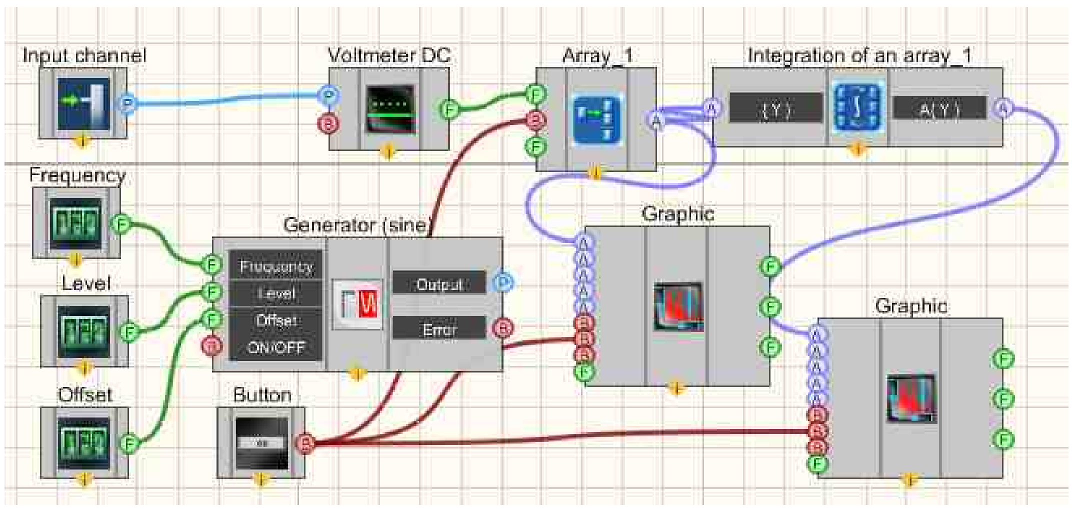
- Are available by the link below. 154

➤**Custom Custom private properties (the default value is shown in parentheses):**

•FLOAT dTime - Set the interval between readings, s (number)
VARIANT_BOOL MeanSubstruction - Set the subtraction of the DC component:
• true - DC subtraction is enabled;
false - DC subtraction is disabled.

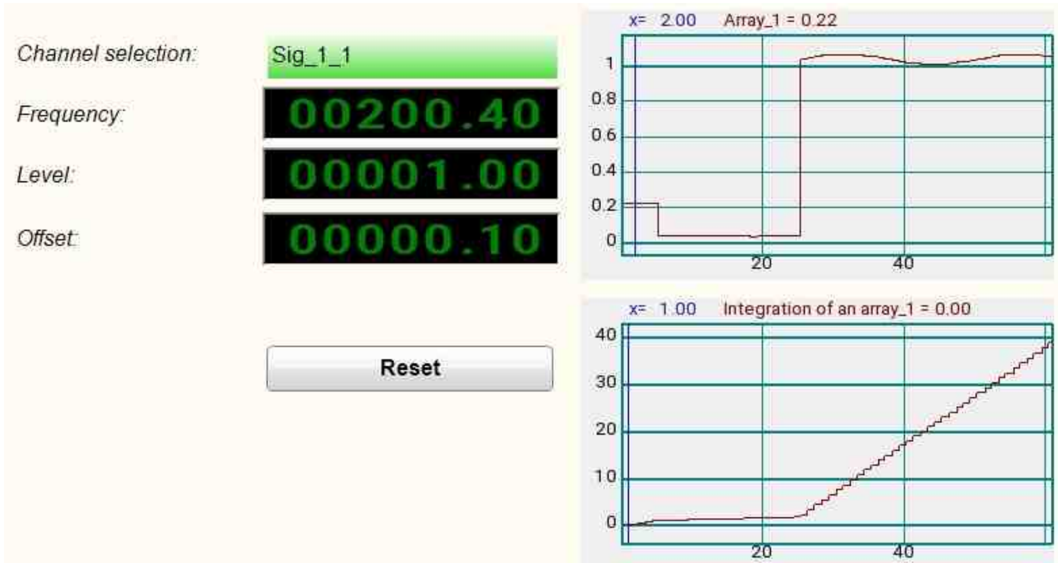
Example

Project in SCADA ZETView



This project shows the operation of the Integration of an array 848 component
The array is formed by the user using the Selector 428 and the Formation of arrays (numeric - array) 796 from which it goes to the Integration of an array 848 component and to the Converters (numeric array - string array) 713 + Data table 934. At the output from the Integration of an array 848 component, the result is also sent to the Converters (numeric array - string array) 713 + Data. table 934. Therefore, we can compare the initial array and the result after the work of the Integration of an array 848 component when the program is started. The "Reset" Non-fixed button 694 connected to the Formation of arrays (numeric - array) 796 resets the array values.

Project operation result

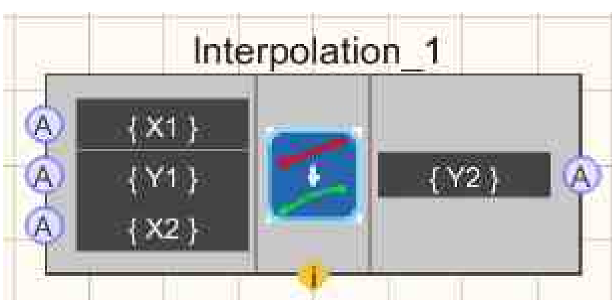


View the example in ZetView

18.7.Interpolation

The component "**Interpolation**" is interpolates the input array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { X1 } - input array X1;
- { Y1 } - input array Y1;
- { X2 } - input array X2.

➢ Output:

- { Y2 } - an interpolation array arrives.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- type (Linear) - interpolation type;
  1.Linear;
  2. Linear (log.);
  3. Spline.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- {X1} - values of the array to which the input array X1 is supplied.
- {Y1} - values of the array to which the input array Y1 is supplied.
- {X2} - values of the array to which the input array X2 is supplied.

➢Custom common properties:

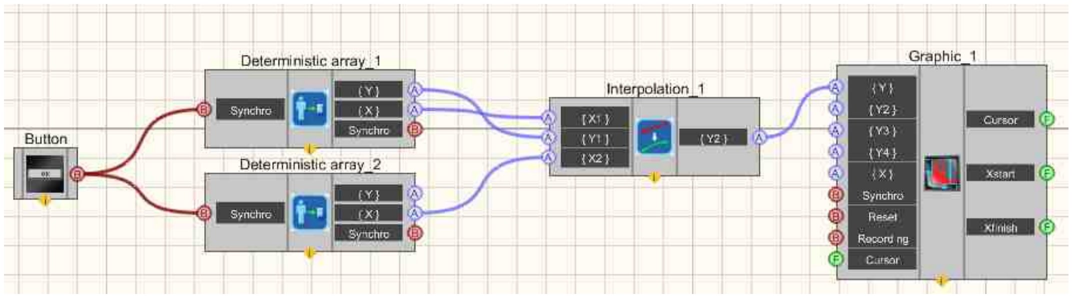
- Are available by the link below. 154

➢Custom Custom private properties (the default value is shown in parentheses):

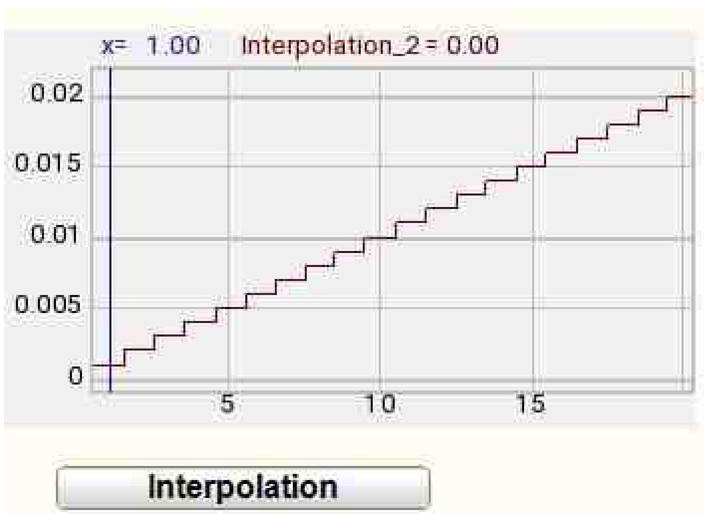
•BSTR Type - Interpolation type setting: Linear; Linear (log.); Spline (string).

Example

Project in SCADA ZETView



Project operation result



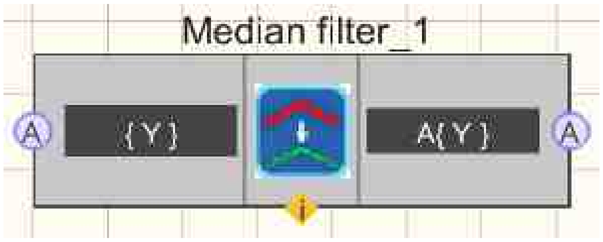
View the example in ZetView

18.8.Median filter

The component "**Median filter**" is designed for efficient processing of signals subject to impulse noise. The median filter selects from the data stream N values (where N is the window size) around each element of the input array. The selected data is sorted in ascending order and an element from the middle of the sorted array is written to the output array.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- A{ Y } - set the array filtered by the median filter arrives.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- WindowLen (3) - set the median filter window width.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- {Y} - values of the array to which the input array is supplied.

➢Custom common properties:

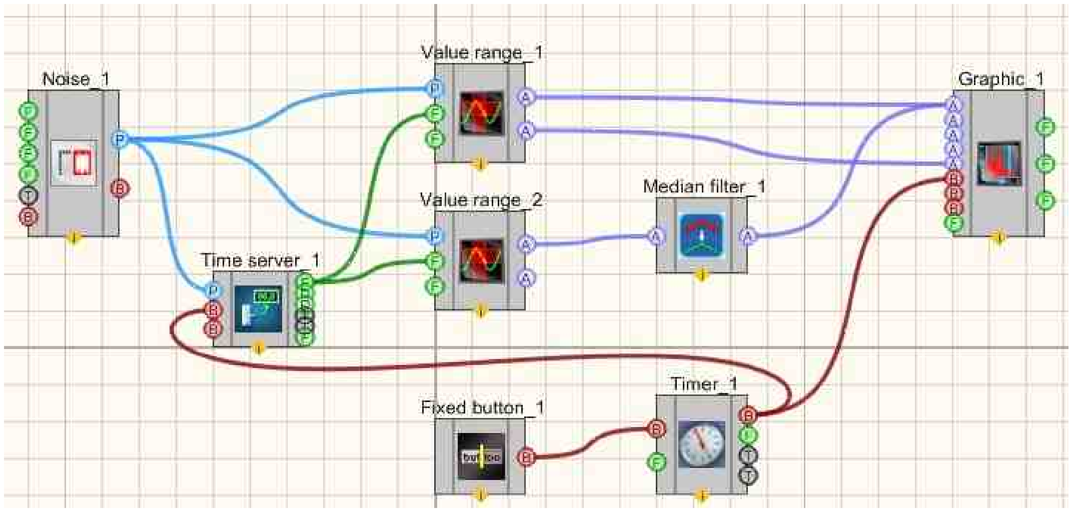
- Are available by the link below. 154

➢Custom Custom private properties (the default value is shown in parentheses):

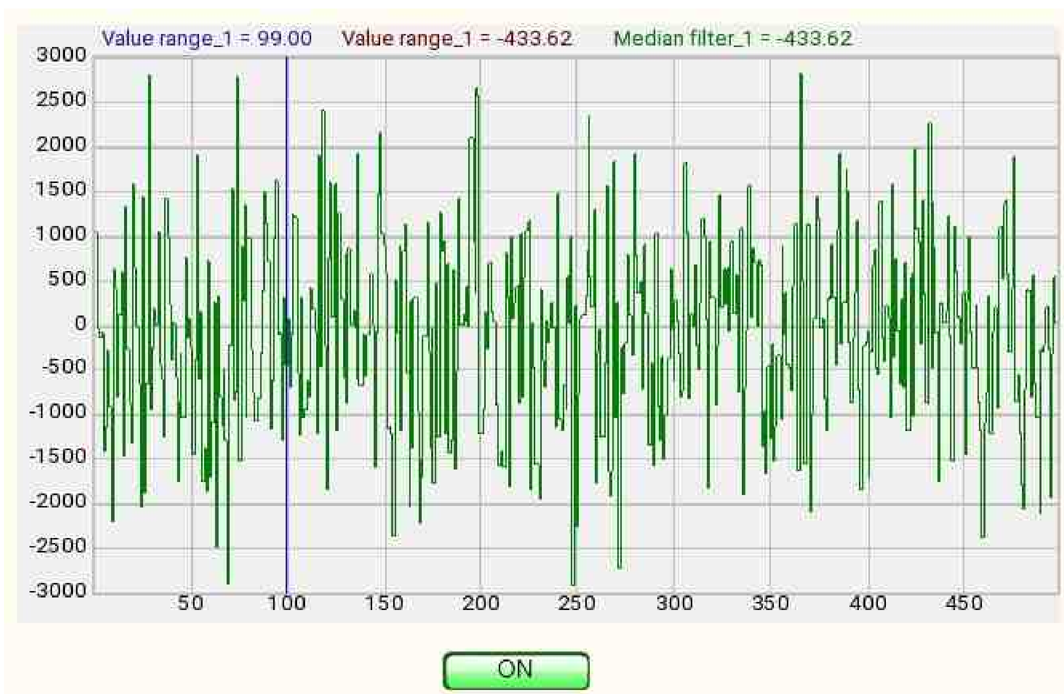
- LONG WindowLen - Set the median filter window width (number).

Example

Project in SCADA ZETView



Project operation result

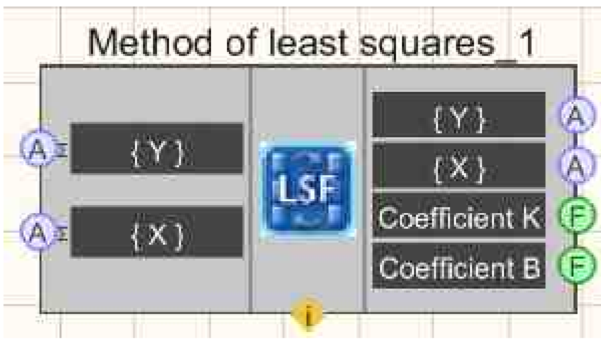


View the example in ZetView

18.9.Method of least squares

The component "**Method of least squares**" is designed to estimate unknown quantities from the results of measurements containing random errors.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - input array Y.
- { X } - input array X.

➢ Output:
- { Y } - output array.
- { X } - array dimension.
- Coefficient K - coefficient K of the resulting formula.
- Coefficient B - coefficient B of the resulting formula.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- type (Linear) - set the type the least squares dependency:
  - Linear;
  - Power;
  - Exponential;
  - Linear fractional function;
  - Fractional-rational function;
  - Logarithmic;
  - Hyperbolic.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - values of the array to which the input array Y is supplied.
- {X} - values of the array to which the input array X is supplied.

➢**Custom common properties:**

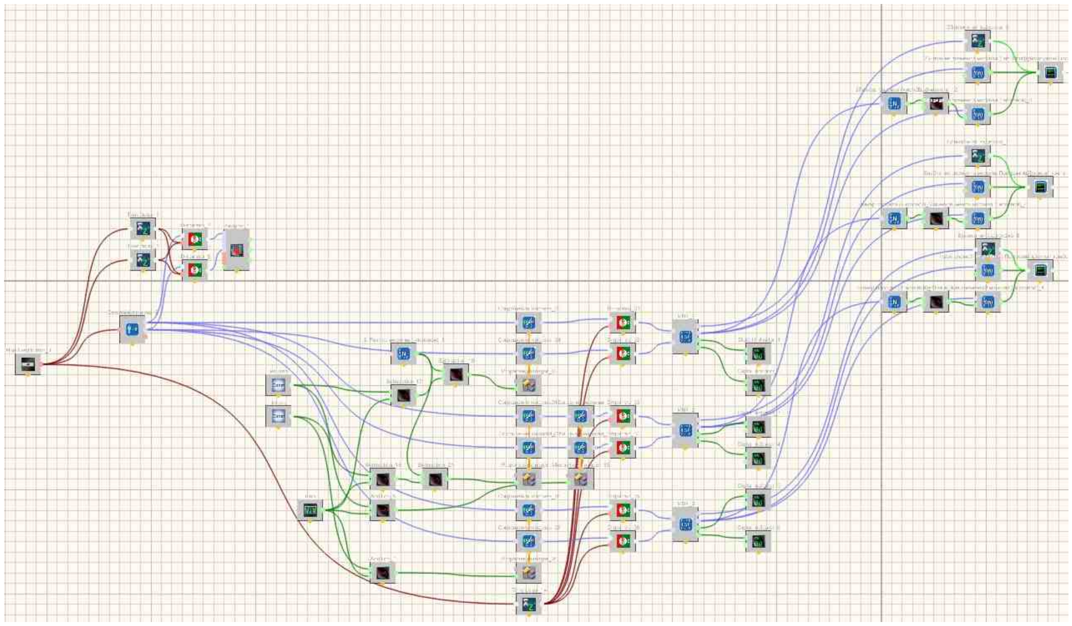
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

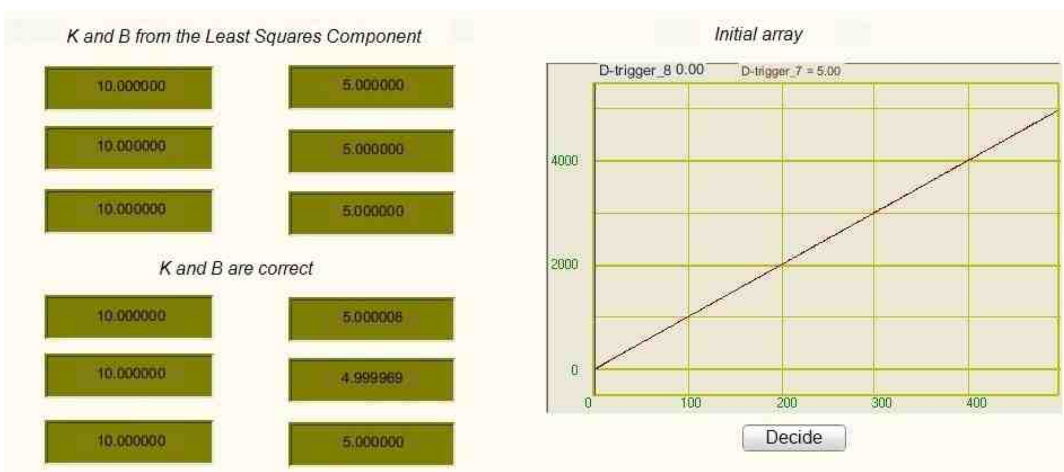
- BSTR Type - set the type the least squares dependency: Linear; Power; Exponential; Linear fractional function; Fractional-rational function; Logarithmic; Hyperbolic (string).

Example

Project in SCADA ZETView



Project operation result

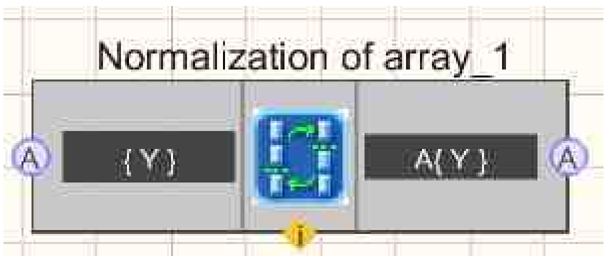


View the example in ZetView

18.10.Normalization of array

The component "**Normalization of array**" is divides the elements of the array into the maximum element.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- { Y } - input array.

➢ Output:
- A{ Y } - a normalized array arrives.

Custom common properties:

➢Total properties (environment):
- [Are available by the link below.](154)

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

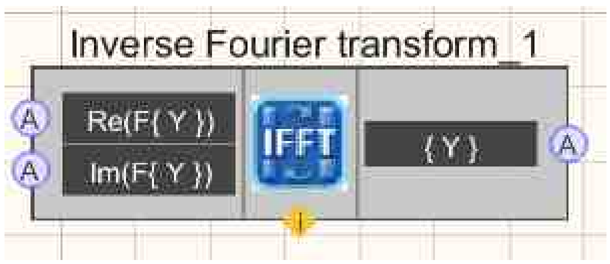
➢**Input:**
- {Y} - Values of the input array, to which the array values are supplied, based on which the normalization of the array will be calculated.

➢**Custom common properties:**
- [Are available by the link below.](154)

18.11.Inverse Fourier transform

The component "**Inverse Fourier transform**" is used a fast inverse Fourier transform, where the input is an array of real and imaginary components, and an array at the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Re (F{ Y }) - real Fourier transform coefficients.
- Im (F{ Y }) - imaginary coefficients of the Fourier transform.

➤ Output:
- { Y } - output array.

Custom common properties:

➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Re (F{Y}) - set the values of the real Fourier transform coefficients.
- Im (F{Y}) - set the values of imaginary Fourier transform coefficients.
-

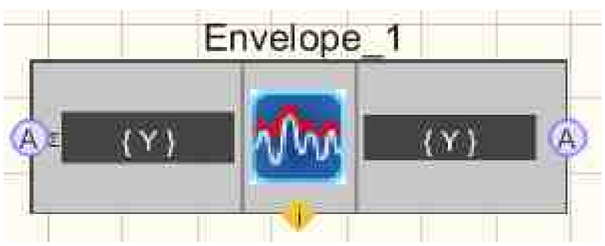
➢**Custom common properties:**

- Are available by the link below. 154

18.12.Envelope

The component "**Envelope**" is a curve is called the surround of a family of curves depending on a parameter if it touches at least one curve of the family at each of its points and touches an infinite set of these curves with each of its segments.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- { Y } - output array.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - set the values of the input array, to which the array values are supplied, based on which the envelope will be calculated.

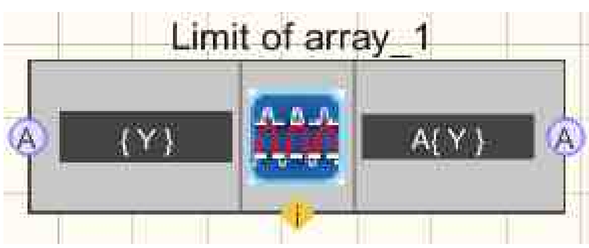
➢**Custom common properties:**

- Are available by the link below.154

18.13.Limit of array

The component "**Limit of array**" is a restricts the array to the level set and the width of the level that defines the boundaries. Elements of the array that do not fall within the specified range are replaced by either the level value or the nearest boundary value.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- A{ Y } - a limited array arrives.

Custom common properties:

➢Total properties (environment):

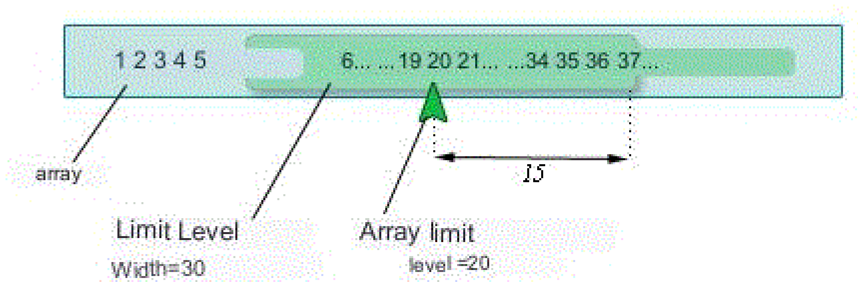
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

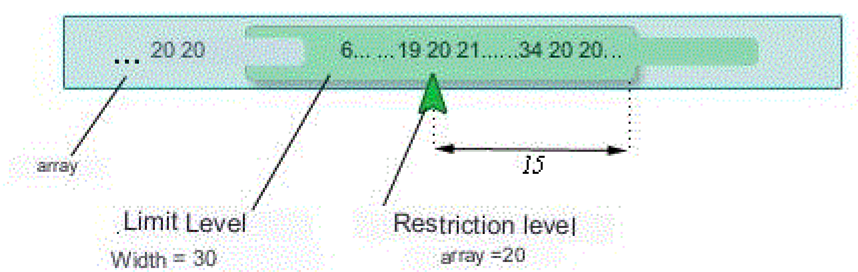
- Level (0) - set the amplitude of the output signal (in channel units).
- LevelWidth (1) - set the level width (in channel units).
- changetype extremum) - change type.
  1. extremum.
  2. level.

Features of the component:

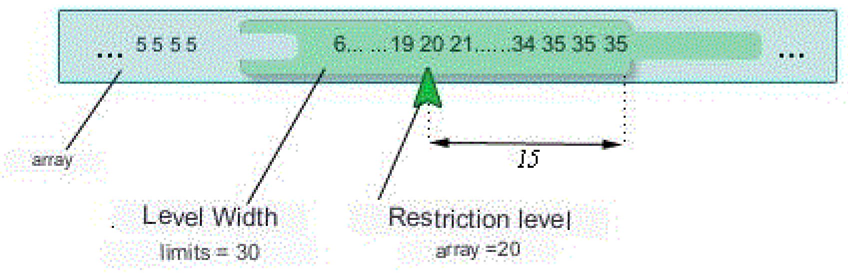
Consider the input array:



After passing the array through the "Array Constraint" component with the replacement by level setting, we get an array:



As can be seen from the figure, all values that do not fall within the specified range, including boundary values, are replaced by the value of the restriction level specified by the user.
If you set the replacement setting to "extremum", then the values that do not fall into the specified range will be replaced by the nearest extremum:



In this example it's 5 and 35.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - values of the array to which the input array should be applied.

➢**Custom common properties:**

- Are available by the link below. 154

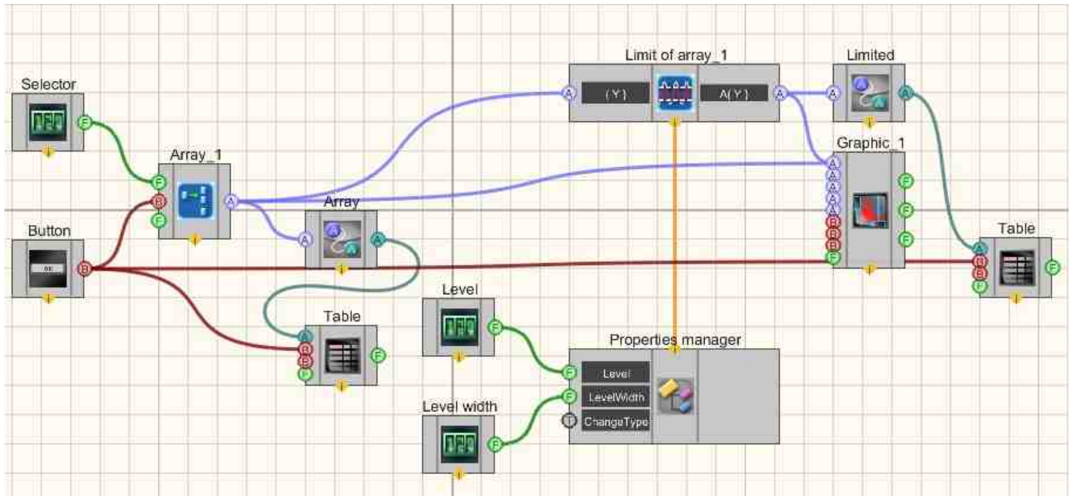
➢**Custom Custom private properties (the default value is shown in parentheses):**

•FLOAT Level - Set the limit level (number).
•FLOAT LevelWidth - Set the level width (number).
•BSTR ChangeType - Set the type of replacement: Extreme; Level (string).

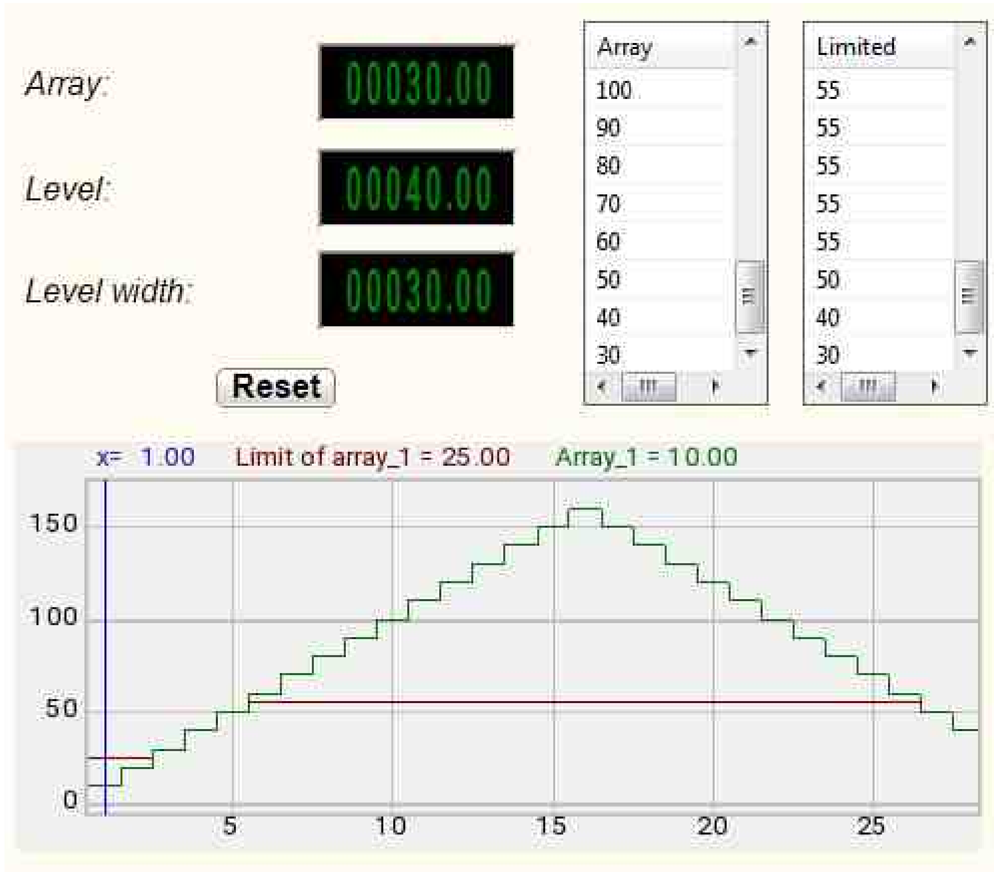
Example No. 1

Replacement setting: "Extremum"

Project in SCADA ZETView



This project shows the operation of the Limit of array 862 component.
Using the Selector 428, array elements are entered, which then go to the array builder. Next, the Formation of arrays (numeric - array) 796 to the input of the Limit of array 862 component and is also converted into a string array using the Converters (numeric array - string array) 713, and displayed in the Data table 934. The output constrained array from the Limit of array 862 component is also converted and output to the Data table 934. Also, two selectors "Level" and "Level Width" are connected to the Limit of array 862 via the Property Manager 999, allowing the user to adjust the corresponding characteristics. Since initially the Limit of array 862 component does not have contact data, but only Custom common properties, the selectors are connected to them via the Property Manager 999.

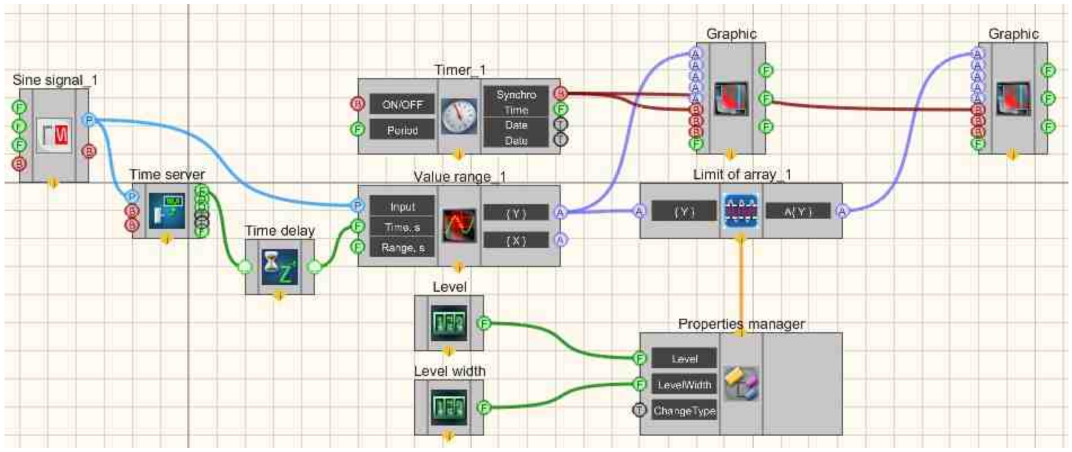


View the example in ZetView

Example No. 2

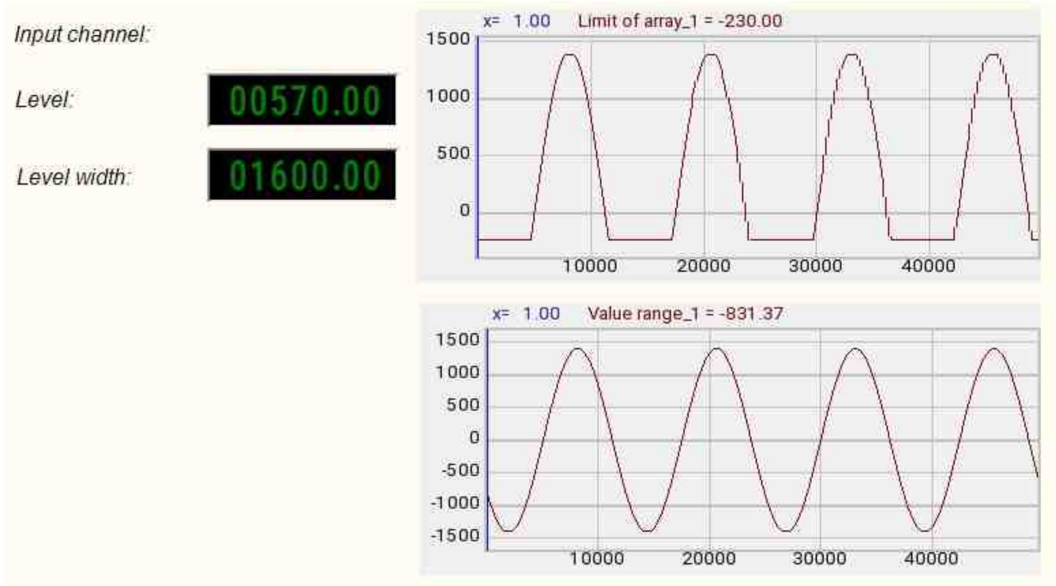
Replacement setting: "Level"

Project in SCADA ZETView



This project shows the operation of the Limit of array 862 component. Using the Input channel 468 component, a signal is received from a physical ZET device connected to the computer. The Range of values 310 component forms an array of instantaneous signal values in a certain period of time. Channel time at the input of the component The Range of values 310 is formed using the link Server Time 366 + Time delay 363. Moreover, the second component is optional, thanks to it the value at the "Channel" input of the Range of values 310 component has time to accumulate. Such a move protects the designer from making a mistake when the time values come before the channel values. After that, the array enters the Limit of array 862 component, and is also displayed on the Graphic 909. After passing the Limit of array 862 component, the output to the Graphic 909 also takes place. The two selectors "Level" and "Level Width" are connected to the Limit of array 862 via the Property Manager 999, allowing the user to adjust the corresponding characteristics. Since the Limit of array 862 component does not initially have contact data, but only custom properties, Selectors 428 are connected to them through the Property Manager 999.

Project operation result



View the example in ZetView

18.14.Operation with a constant

The component "**Operation with a constant**" is performs the selected action on the resulting constant and the input array.

When you select an action in the properties of the "**Operation with a constant**" component, on the Example "**Addition**", it will be performed element-by-element with each element of the array.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- { Y } - input array.
- Constant - the value of the constant.

➢ Output:
- A{ Y } - output array.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Constant (0) - a constant for arithmetic operations on an array.
- operationtype (Addition) - a variant of the arithmetic operation on an array:
  1. Addition.
  2. Subtraction.
  3. Multiplication.
  4. Division.
  5. Constant in the power of array elements.
  6. Array elements to the power of a constant.
  7 Logarithm of elements to the base of a constant.
- FirstReaction (true) - set the reaction of changing the output signal to changing the values at input No.1:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.
- SecondReaction (true) - set the reaction of changing the output signal to changing values at input No.2:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- {Y} - array numeric values.
- Constant - numeric values of the array.

➢Custom common properties:

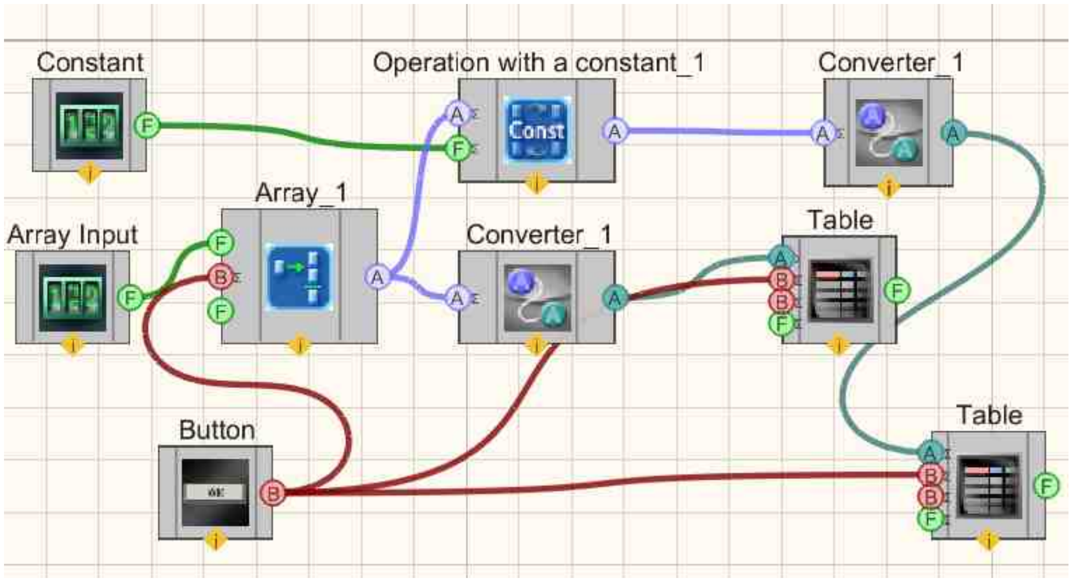
- Are available by the link below. 154

➢Custom Custom private properties (the default value is shown in parentheses):

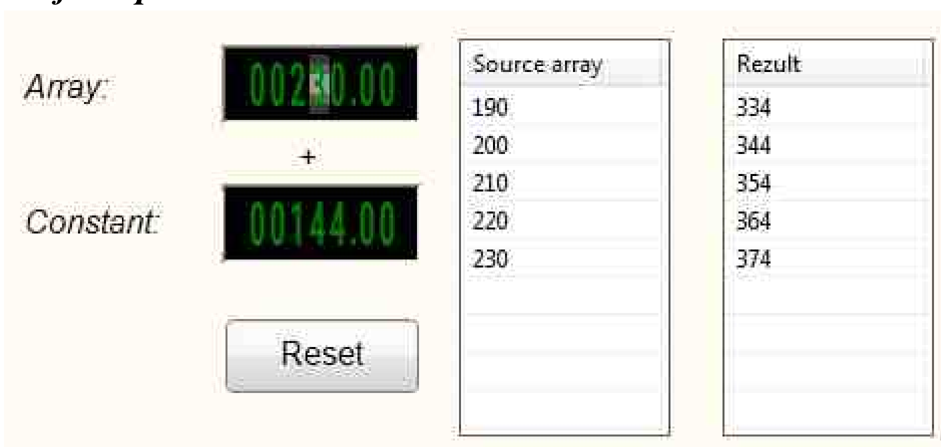
- FLOAT Constant - Set a constant for arithmetic operations on an array (number).
- BSTR OperationType - Set the variant of the arithmetic operation on the array: Addition; Subtraction; Multiplication; Division; Constant to the power of array elements; Array elements to the power of a constant; The logarithm of the array elements to the base of the constant (string).
- VARIANT_BOOL FirstReaction - Set the value of the reaction to data change on input 1:
  - true - the output value is issued when data arrives at input 1;
  - false - the output value is not issued when data arrives at input 1.
- VARIANT_BOOL SecondReaction - Set the value of the reaction to data change at input 2:
  - true - the output value is issued when data arrives at input 2;
  - false - the output value is not issued when data arrives at input 2.

Example

Project in SCADA ZETView



Project operation result



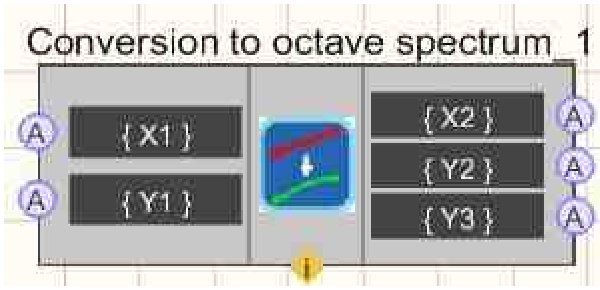
View the example in ZetView

18.15.Conversion into octave spectrum

The component "**Conversion into octave spectrum**" is designed to convert an ordinary spectrum into a one-third octave spectrum.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- { X1 } - input array X1 of the usual spectrum.
- { Y1 } - input array Y1 of the usual spectrum.

➢Output:
- { X2 } - output array X2 of one-third octave spectrum.
- { Y2 } - output array Y2 of one-third octave spectrum.
- { Y3 } - output array Y3 of one-third octave spectrum.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- operationtype (maximum) - a variant of the arithmetic operation on the array:
  1. average.
  2. maximum.

Программирование

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- {X1} - Values of the input array X1 of the usual spectrum.
- {Y1} - Values of the input array Y1 of the usual spectrum.

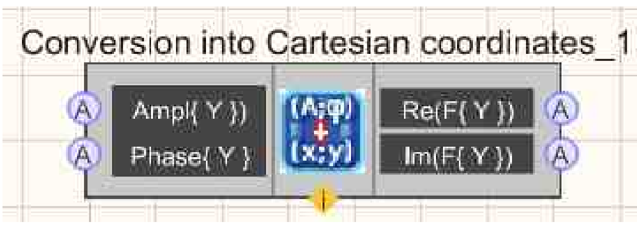
➢Total properties (environment):

- [Are available by the link below.](#) 154

18.16.Conversion into Cartesian coordinates

The component "**Conversion into Cartesian coordinates**" is designed for conversion from polar coordinates to Cartesian

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Re (F{ Y }) - real parts of an array of complex numbers.
- Im (F{ Y }) - imaginary parts of an array of complex numbers.

➢ Output:

- Ampl{ Y } - output array of moduli array of complex numbers.
- Phase { Y } - output array of complex number phases.

Custom common properties:

➢Total properties (environment):

- [Are available by the link below.](#) 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

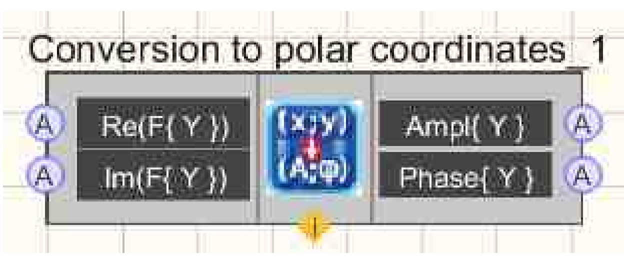
➤**Input:**
- Re (F{Y}) - Values are the real parts of an array of complex numbers.
- Im (F{Y}) - Values of the imaginary part of an array of complex numbers.

➤**Custom common properties:**
- Are available by the link below. 154

18.17.Conversion to polar coordinates

The component "**Conversion to polar coordinates**" is designed to convert from Cartesian coordinates to polar coordinates.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Re (F{ Y }) - real parts of an array of complex numbers.
- Im (F{ Y }) - imaginary parts of an array of complex numbers.

➤ Output:
- Ampl{ Y } - output array of moduli array of complex numbers.
- Phase { Y } - output array of complex number phases.

Custom common properties:

➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**
Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

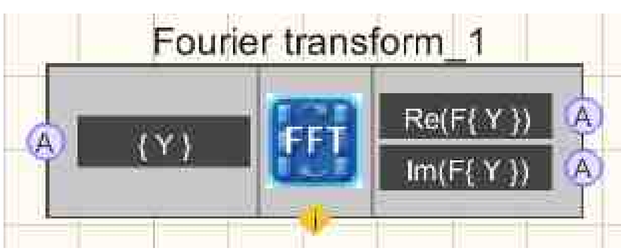
➤**Input:**
- Re (F{Y}) - Values are the real parts of an array of complex numbers.
- Im (F{Y}) - Values of the imaginary part of an array of complex numbers.

➤**Custom common properties:**
- Are available by the link below. 154

18.18. Fourier transform

The component "**Fourier transform**" is performs a fast Fourier transform on the input array, outputting arrays of real and imaginary components at the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- { Y } - input array.

➢ Output:
- Re (F{ Y }) - real Fourier transform coefficients.
- Im (F{ Y }) - imaginary coefficients of the Fourier transform.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

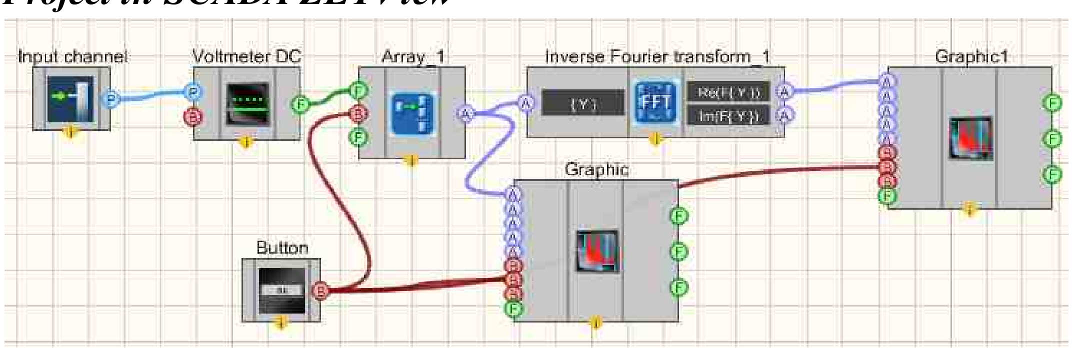
Setting:

➢**Input:**
- {Y} - values of the input array, to which the array values are supplied, based on which the Fourier transform will be calculated.

➢**Custom common properties:**
- Are available by the link below. 154

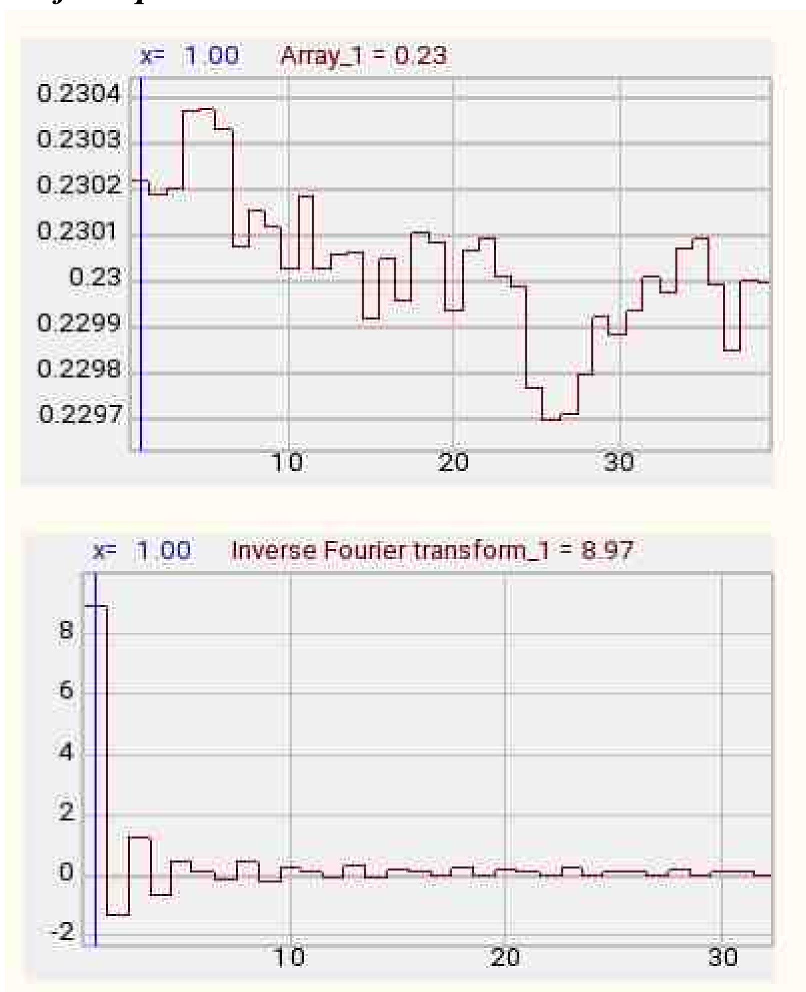
Example

Project in SCADA ZETView



In this project, the ZETLab "Signal Generator" virtual device is used, from which the signal is received through the Measuring channel 468 component. The values from the generator are sent to the DC Voltmeter 591, after which the measured voltage values are sent to the Formation of arrays (numeric - array) 796 component, then the array of numbers goes to the "Graph", thereby the original signal is displayed on the first Graphic1 909, and on the Fourier transform 875 where is subjected to the FFT and is also displayed on the second Graphic 909.

Project operation result



View the example in ZetView

18.19.Calculation of level in dB

The component "**Calculation of level in dB**" is intended for conversion of absolute values into relative values.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- A{ Y } - the calculation of the value in dB is received.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Referense (0.001) - value for level calculation in dB.
- factor (20) - multiplier for calculation in dB.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- {Y} - input numeric array values.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

•FLOAT Referense - Set the value for level calculation in dB (number).
•FLOAT factor - Set the multiplier for calculation in dB (number).

18.20.Reduction of the array

The component "**Reduction of the array**" is reduces the array by cutting the Nth number of elements to the right or left, where N is the size of the reduction of the array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- A{ Y } - a reduced array arrives.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- cuttype (Right) - The cut type of the array. It takes two values to choose from: Left and Right. This setting selects from which side the elements of the array will be cut off.
- CutSize (0) - (0) - Number of cut elements. Accepts any integer value. Indicates the number of array elements to cut off.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**
- {Y} - Array values that will be taken to reduce the array.

➤**Custom common properties:**
- Are available by the link below. 154

➤**Custom Custom private properties (the default value is shown in parentheses):**

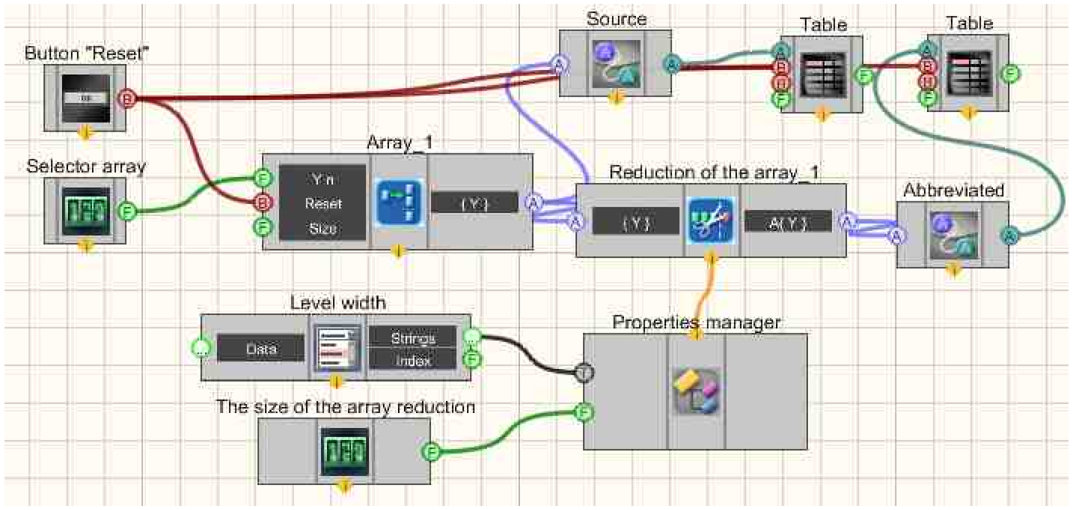
•BSTR CutType - Set the type of array reduction: Right; Left (string).
•LONG CutSize - Set the number of cut elements (number).

➤**Methods:**

- FLOAT GetValueFromIndex(LONG index) - Read the array value at the given index.

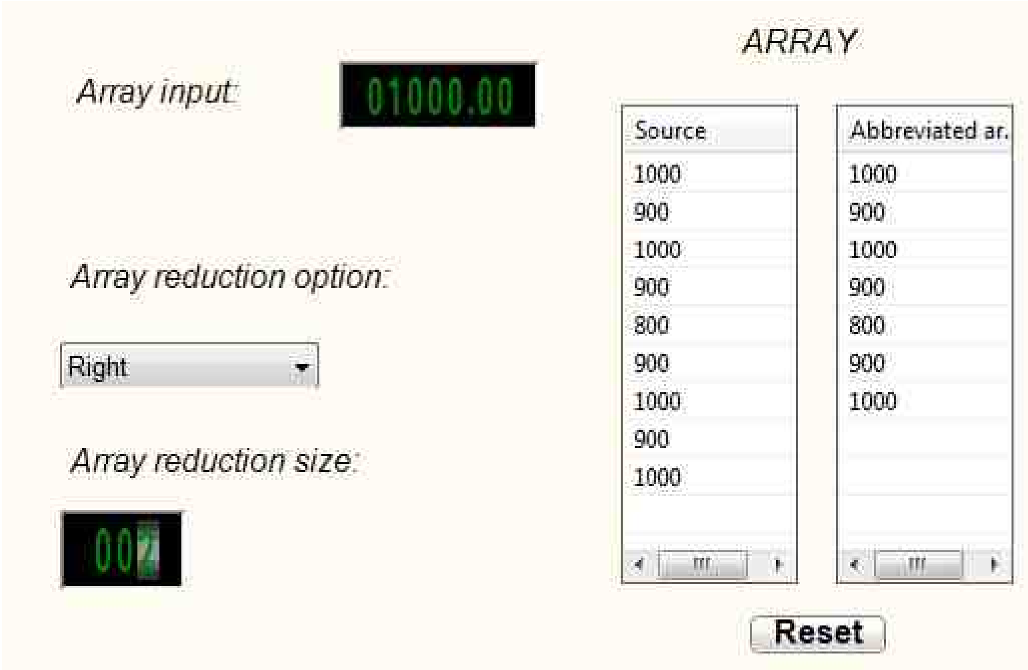
Example

Project in SCADA ZETView



This project shows the operation of the Reduction of the array[879] component. Using the Selector[428], array elements are entered, which then go to the array builder. Next, the generated array is fed to the input of the Reduction of the array[879] component, is also converted into a string array using the Converters (numeric array - string array)[713], and displayed in the Data table[934]. The output reduced array from the Reduction of the array[879] component is also converted and output to the Data table[934].
The "The size of the array reduction" Selector[428] and the "Level width" Combined list[414] are connected to the Reduction of the array[879] via the Property Manager[999], allowing the user to adjust the corresponding characteristics. Since the Reduction of the array[879] component does not initially have contact data, but only custom properties, selectors are connected to them through the Property Manager[999].

Project operation result

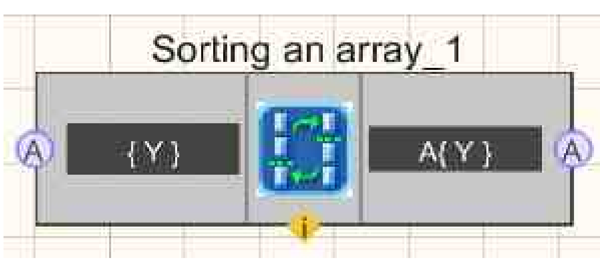


View the example in ZetView

18.21.Sorting an array

The component "**Sorting an array**" is sorts the input array by placing items in ascending or descending order.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array.

➢ Output:

- A{ Y } - a sorted output array arrives.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- sorttype (By descending order) - sort type:
  - By descending order;
  - By ascending order.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - set the values of the input array, which receives the array to perform sorting.

➢**Custom common properties:**

- Are available by the link below. 154

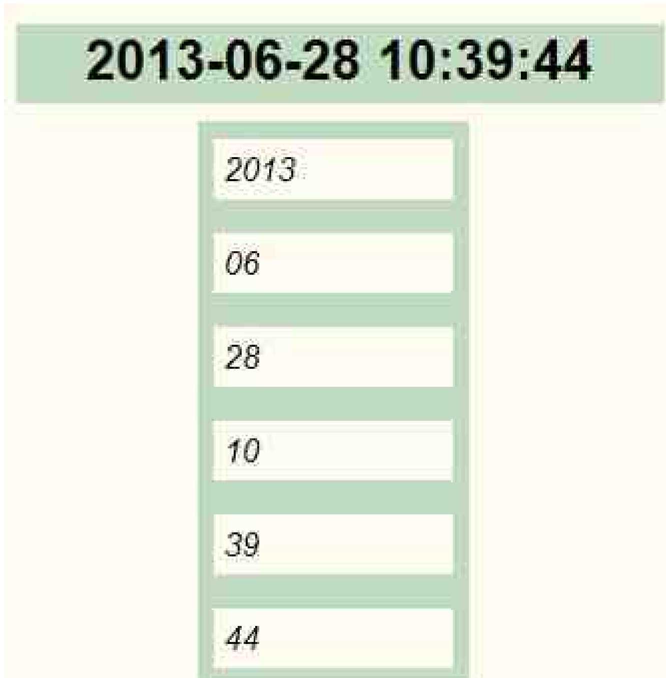
➢**Custom Custom private properties (the default value is shown in parentheses):**

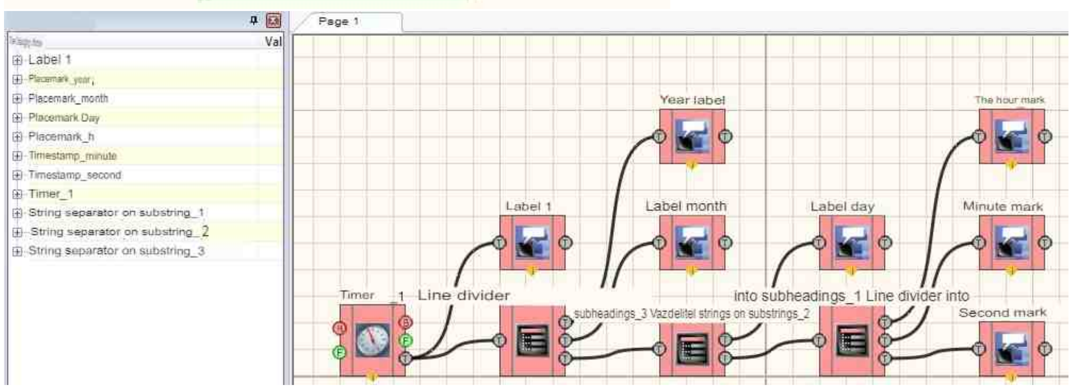
•BSTR SortType - Set sort type: By descending order; By ascending order (string).

Example of creating an array sort (found in the Example Debugging projects in the ZETView SCADA system in Example No.4 146)

Project in SCADA ZETView

Project operation results



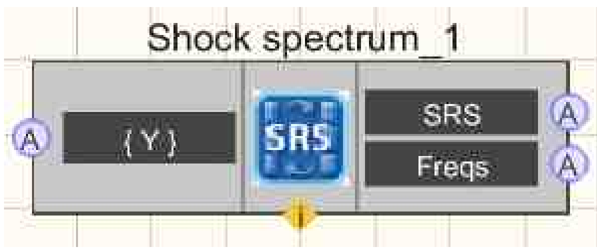


View the example in ZetView

18.22.Shock spectrum

The component "**Shock spectrum**" (the response response spectrum) is the dependence of the maximum responses on the perturbing effect of the system from resonators with one degree of freedom, arranged in order of increasing natural frequencies.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array with data.

➢ Output:

- SRS - shock response spectrum.
- Freqs - frequency bans.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- SamplingRate (25000) – sampling frequency of input data;
- FirstFrequency (10) – the first frequency in the frequency bands;
- LastFrequency (5000) – the last frequency in the frequency bands;
- DampingKoef (0,05)– the damping coefficient of oscillatory systems;
- FreqsInOctave (16) – the number of frequency bands in one octave.

- **Notes:**

- The last frequency in the frequency bands should not be more than j of the sampling frequency.
- The default damping factor is 0.05.

- The algorithm is very resource-intensive, you should not submit large amounts of data to the input too often.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - Values of the input array with data.

➢**Custom common properties:**

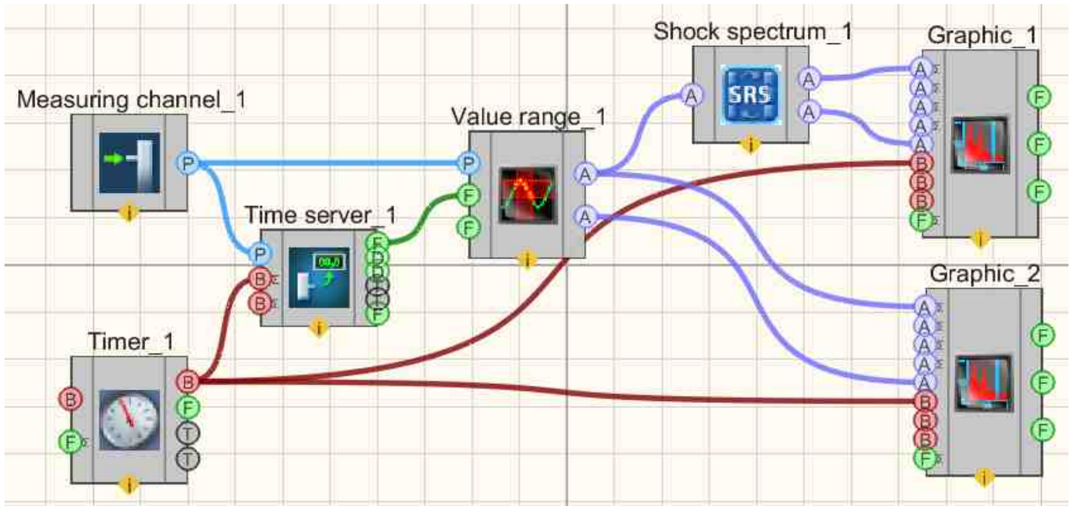
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

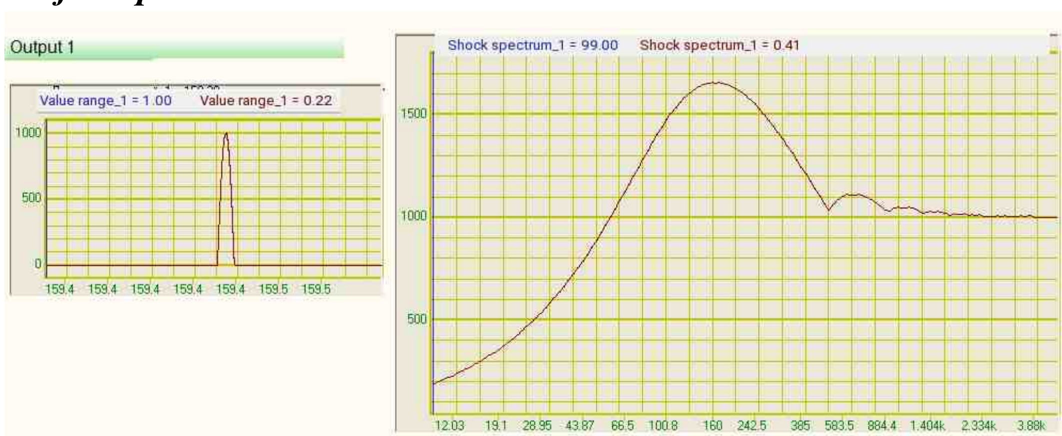
- FLOAT SamplingRate - Set the sampling frequency (from 0.001 to half the sampling frequency of the input data), Hz.
- FLOAT FirstFrequency - Set the first frequency in the frequency bands (from 0.01 Hz to the last frequency in the frequency bands), Hz.
- FLOAT LastFrequency - Set the last frequency in the frequency bands (from the first frequency in the frequency bands to (DAC sample frequency / 4)), Hz
- FLOAT DampingKoef - Set the damping coefficient of oscillatory systems (number).
- LONG FreqsInOctave - Set the number of frequency bands in one octave (number).

Example

Project in SCADA ZETView



Project operation result

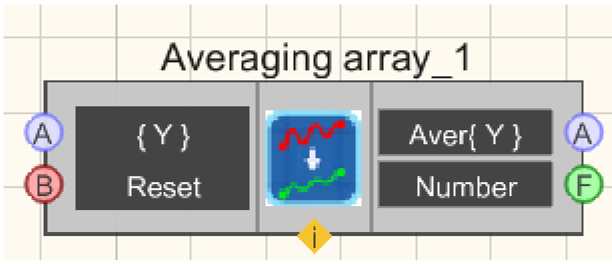


View the example in ZetView

18.23.Averaging array

The component "**Averaging array**" is averages the input array.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array with data.
  Reset - reset averaging.

➢ Output:

- Aver{ Y } - output averaged array.
- Number - the amount over which the average was performed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Aver (Arithmetical) - selection of averaging: Arithmetical; Exponential;
- Alpha (0) – Alpha parameter;

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - values of the input array with data.

➢**Custom common properties:**

- Are available by the link below. 154

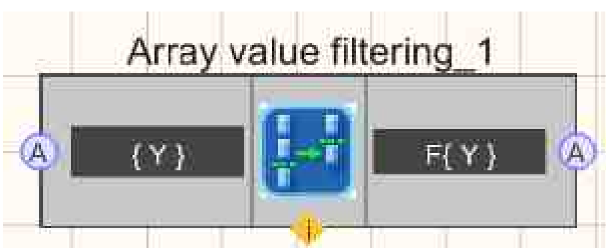
➢**Custom Custom private properties (the default value is shown in parentheses):**

• BSTR Aver - Set the choice of averaging: Arithmetical; Exponential(string);
•FLOAT Alpha - Set the Alpha parameter (number).

18.24.Filtering array

The component "**Filtering array**" is intended for filtering signals arriving at the input channels of spectrum analyzers, strain-gauge station, seismic stations for further processing by ZETLab programs

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- { Y } - input array for filtering.

➢ Output:

- F{ Y } - the filtered array arrives.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- fMax (100) - maximum cutoff frequency of the low-pass filter, Hz.

- fMin (10) - minimum cutoff frequency of the high-pass filter, Hz.
- frequency (25000) - sampling frequency of the original signal, Hz
- filtertype () - filter type:
  1. linear.
  2. integrating.
  3. integrating 2.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- {Y} - numeric values of the array to which the input array should be applied for filtering.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

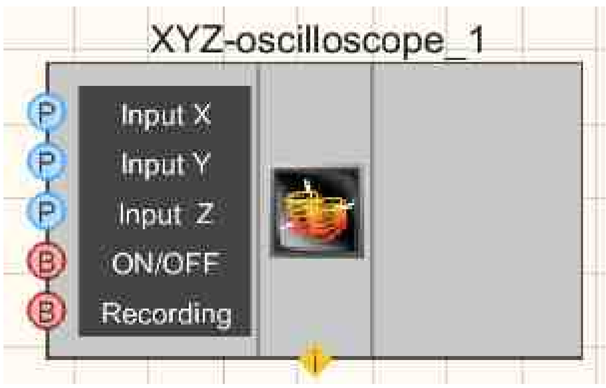
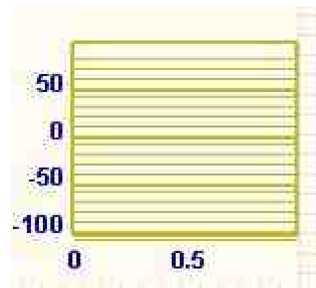
- FLOAT fMax - Set the cutoff frequency of the low-pass filter, Hz (from 0.01 to half the sampling frequency of the filtered channel).
- FLOAT fMin - Set the cutoff frequency of the high-pass filter, Hz (from 0.01 to half the sampling frequency of the filtered channel).
- FLOAT frequency - Set the sampling frequency of the original signal (from 0.01 to (sampling frequency DAC / 2)), Hz.
- BSTR filterType - Set filter type: linear; integrating; integrating 2 (string).

Глава 19.Display

19.1.XYZ oscilloscope

The component "**XYZ oscilloscope**" is designed to view the waveform, measure the instantaneous values of the signal and display the parametric dependence of the signals.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Input X - data channel along the X axis.
- Input Y - data channel along the Y axis.
- Input Z - data channel along the Z axis.
- ON\OFF - Turns the XYZ oscilloscope on and off.
- Recording - when an impulse is applied to this contact, the data is saved to a separate file (the name and location of the file are specified in the private properties of the component).

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- GridColor (bfbf20) - set the color of the grid.
- CursorColor (000080) - set the cursor color.
- BackColor (ffffbf0) - set the background color.
- StartGraphColor (f0f0f0) - set the initial graphic color.
- EndGraphColor (601010) - set the end color of the graph.

- DigitsColor (0000be) - set the color of the labels on the axes.
- LegendColor (009400) - set the color of the header.
- coordinatesystem (XT) - set the coordinate system:
  1. XT.
  2. YT.
  3. ZT.
  4. XY.
  5. XZ.
  6. YZ.
  7. XYT.
  8. XZT
  9. YZT.
  10. XYZ.
- BSTR FreqRange - Set the decade of the frequency band, Hz:
  0 - from 0 to (sampling frequency / 1).
  1 - from 0 to (sampling frequency / 10).
  2 - from 0 to (sampling frequency / 100).
  3 - from 0 to (sampling frequency / 1000).
  4 - from 0 to (sampling frequency / 10000).
- coordinategrid (Grid in X and Y) - set the coordinate grid:
  1. No grid.
  2. Grid in the X.
  3. Grid in the Y.
  4. Grid in the X and Y.
- update (1 s) – the graphic update (0.1 or 1 s).
- interval (1) - the display interval, s.
- TStart (0) - the zero offset along the time axis, s.
- TInterval (1) - the display length along the time axis, s.
- Xstart (-100) - the start of display along the X axis, units.
- Xend (100) - the end of the display along the X axis, units.
- Ystart (-100) - the start of display along the Y axis, units.
- Yend (100) - the end of the display along the Y axis, units.
- Zstart (-100) - the beginning of the display along the Z axis, units.
- Zend (100) - the end of the display along the Z axis, units.
- FileName - file name for rocoding results.
- Activate (true) - the operation status (enabled (true)/disabled (false)).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**

- Input X - An input numeric array with values to be displayed on the X axis (any number).
- Input Y - An input numeric array with values to be displayed along the Y axis (any number).
- Input Z - An input numeric array with values to be displayed on the Z axis (any number).

➢**Custom common properties**:
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- LONG GridColor - Set the grid color (number).
- LONG CursorColor - Set cursor color (number).
- LONG BackColor - Sets the background color (number).
- LONG StartGraphColor - Set the initial graphic color (number).
- LONG EndGraphColor - Sets the end color of the graphic (number).
- LONG DigitsColor - Sets the color value of the axis labels (number).
- LONG LegendColor - Set the color of the header header (number).
- BSTR FreqRange - Set the decade of the frequency range, Hz:
  0 - from 0 to (sampling frequency / 1)
  1 - from 0 to (sampling frequency / 10)
  2 - from 0 to (sampling frequency / 100)
  3 - from 0 to (sampling frequency / 1000).
  4 - from 0 to (sampling frequency / 10000).
- BSTR CoordinateSystem - set the coordinate system (string):
  1. XT.
  2. YT.
  3. ZT.
  4. XY.
  5. XZ.
  6. YZ.
  7. XYT.
  8. XZT
  9. YZT.
  10. XYZ.
- BSTR CoordinateGrid - Set the coordinate grid (string):
  1. No grid.
  2. Grid in the X.

3. Grid in the Y.
4. Grid in the X and Y.

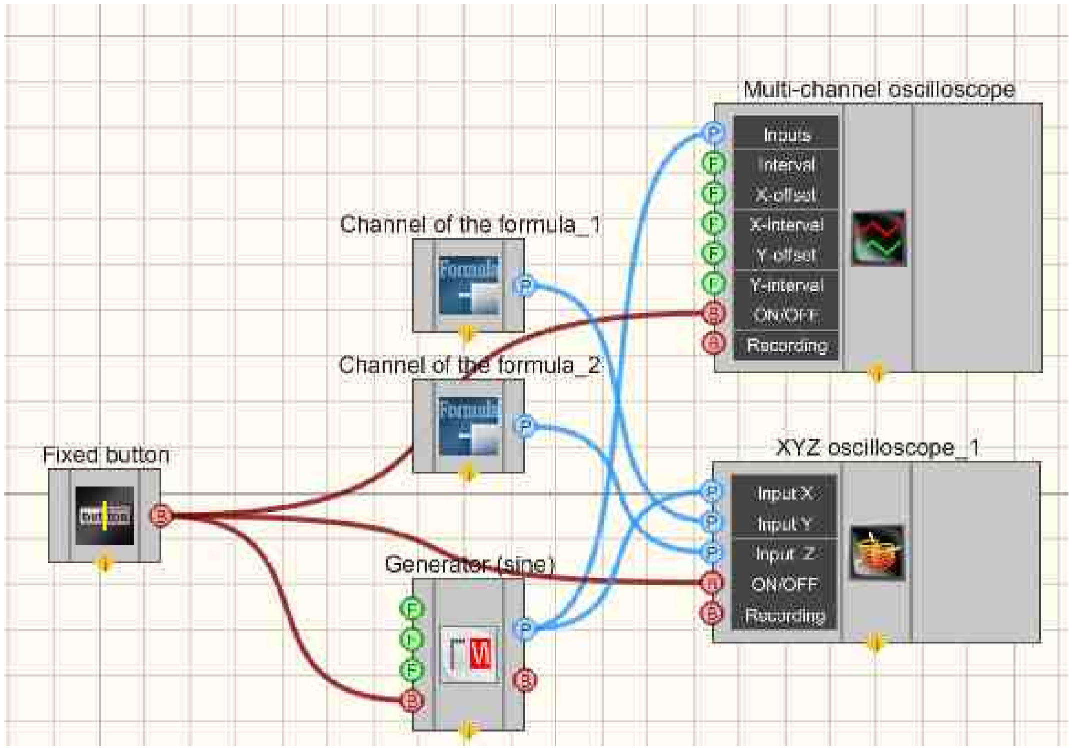
- BSTR Update - Set the graphic update: 0.1 s; 1 s (string).
- FLOAT Interval - Set the display interval, s (number).
- FLOAT TStart - Set zero offset along the time axis, s (number).
- FLOAT TInterval - Set the display length along the time axis, s (number).
- FLOAT Xstart - Set the display start on the X axis (in units) (number).
- FLOAT Xend - Set the end of the display along the X axis (in units) (number).
- FLOAT Ystart - Set the start of the display along the Y axis (in units) (number).
- FLOAT Yend - Set the end of the display along the Y axis (in units) (number).
- FLOAT Zstart - Set the start of the display along the Z axis (in units) (number).
- FLOAT Zend - Set the end of the display along the Z axis (in units) (number).
- BSTR FileName - Set the file name for record the results to a *.dtu file (after setting this parameter, the file is recorded to the specified file).
- VARIANT_BOOL Activate - Set the status of the XYZ-oscilloscope:
  true - the status of the XYZ-oscilloscope is on;
  false - the status of the XYZ-oscilloscope is off;

➢**Methods:**

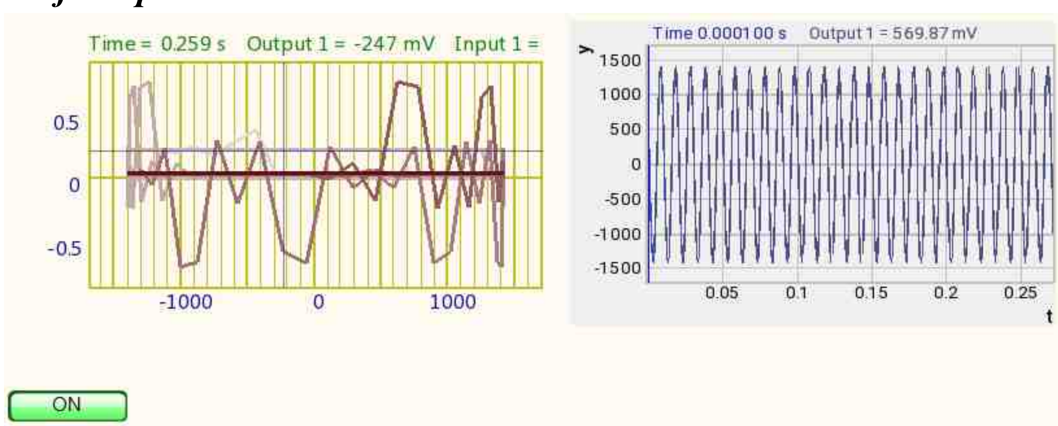
- void CopyPicture (void) - Copies the graphic to the clipboard.
- void CopyData (void) - Copies data to the clipboard.

Example

Project in SCADA ZETView



Project operation result

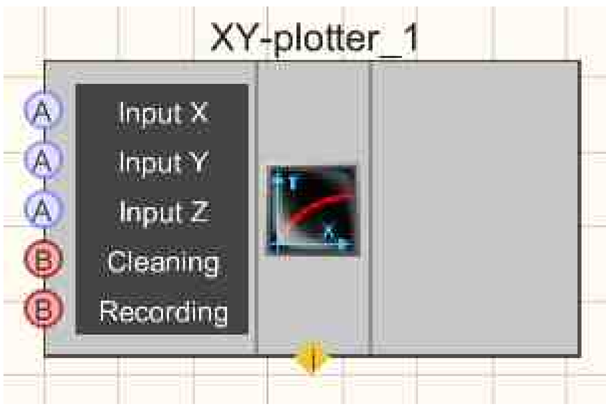
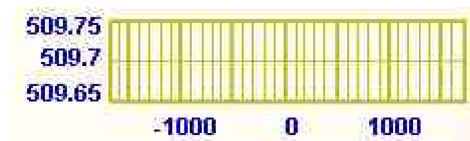


View the example in ZetView

19.2.XYZ plotter

The component "**XY plotter**" is designed for visualization (form evaluation) of mutual characteristics of two measured quantities.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
➢ Input X - data channel along the X axis.
➢ Input Y - data channel along the Y axis.
➢ Input Z - data channel along the Z axis.
- Clearing - when a impulse is applied to which, the display is cleared.
- Recording - when an impulse is applied to this contact, the data is saved to a separate file (the name and location of the file are specified in the private properties of the component).

➢ Output:
Doesn't have.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- GridColor (bfbf20) - set the color of the grid.
- CursorColor (000080) - set the cursor color.
- BackColor (fffbf0) - set the background color.
- StartGraphColor (f0f0f0) - set the initial graphic color.
- EndGraphColor (601010) - set the end color of the graph.

- DigitsColor (0000be) - set the color of the labels on the axes.
- LegendColor (009400) - set the color of the header.
- coordinatesystem (XT) - set the coordinate system:
  1. XT.
  2. YT.
  3. ZT.
  4. XY.
  5. XZ.
  6. YZ.
  7. XYT.
  8. XZT
  9. YZT.
  10. XYZ.
- coordinategrid(Grid in X and Y) - set the coordinate grid:
  1. No grid.
  2. Grid in the X.
  3. Grid in the Y.
  4. Grid in the X and Y.
- Autoscale (false) - set the autoscale.
- Xstart (-100) - the start of display along the X axis, units.
- Xend (100) - the end of the display along the X axis, units.
- Ystart (-100) - the start of display along the Y axis, units.
- Yend (100) - the end of display along the Y axis, units.
- Zstart (-100) - the beginning of the display along the Z axis, units.
- Zend (100) - the end of the display along the Z axis, units.
- FileName - the name of the file to recorded the results.
- XUnit - the unit of measurement along the X-axis.
- YUnit - the unit of measurement along the Y-axis.
- ZUnit - the unit of measurement along the Z-axis.
- ShowTimeLegend - display the time axis signature.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**

- Input X - An input numeric array with values to be displayed along the X axis (any number).
- Input Y - An input numeric array with values to be displayed along the Y axis (any number).
- Input Z - An input numeric array with values to be displayed along the Z axis (any number).

➢**Custom common properties**:

- Are available by the link below. 154

➢**Custom private properties**:

- LONG GridColor - Set the grid color (number).
- LONG CursorColor - Set cursor color (number).
- LONG BackColor - Set the background color (number).
- LONG StartGraphColor - Set the initial graphic color (number).
- LONG EndGraphColor - Set the end color of the graphic (number).
- LONG DigitsColor - Set the color value of the axis labels (number).
- LONG LegendColor - Set the color of the header header (number).
- BSTR CoordinateSystem - Set the coordinate system: XT; YT; ZT; XY; XZ; YZ; XYT; XZT; YZT; XYZ (string).
- BSTR CoordinateGrid - Set the coordinate grid: No grid; Grid on X; Y grid; Grid in X and Y (string).
- VARIANT_BOOL Autoscale - Set the autoscale:
  true - Autoscale is enabled;
  false - Autoscale is disabled.
- FLOAT Xstart - Set the display start on the X axis (in units) (number).
- FLOAT Xend - Set the end of the display along the X axis (in units) (number).
- FLOAT Ystart - Set the start of the display along the Y axis (in units) (number).
- FLOAT Yend - Set the end of the display along the Y axis (in units) (number).
- FLOAT Zstart - Set the start of the display along the Z axis (in units) (number).
- FLOAT Zend - Set the end of the display along the Z axis (in units) (number).
- BSTR FileName - Set the file name for recording the results to a *.dtu file (after setting this parameter, the file is recording to the specified file).
- BSTR XUnit - Set the unit of measurement along the X-axis.
- BSTR YUnit - Set the unit of measurement along the Y-axis.
- BSTR ZUnit - Set the unit of measurement along the Z-axis.
- VARIANT_BOOL ShowTimeLegend - display the time axis signature:
  true - Signature display is enabled;
  false - The signature display is disabled.

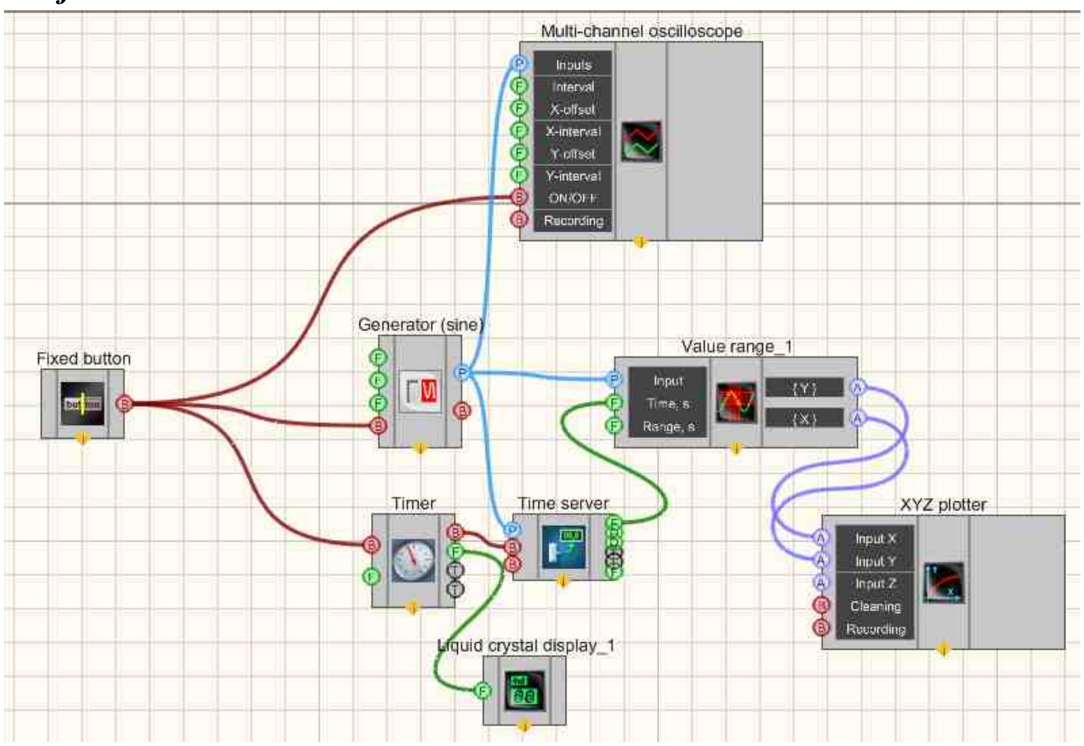
➢**Methods:**

- void PrintScreen (void) - Saves pictures to the clipboard (PrintScreen).

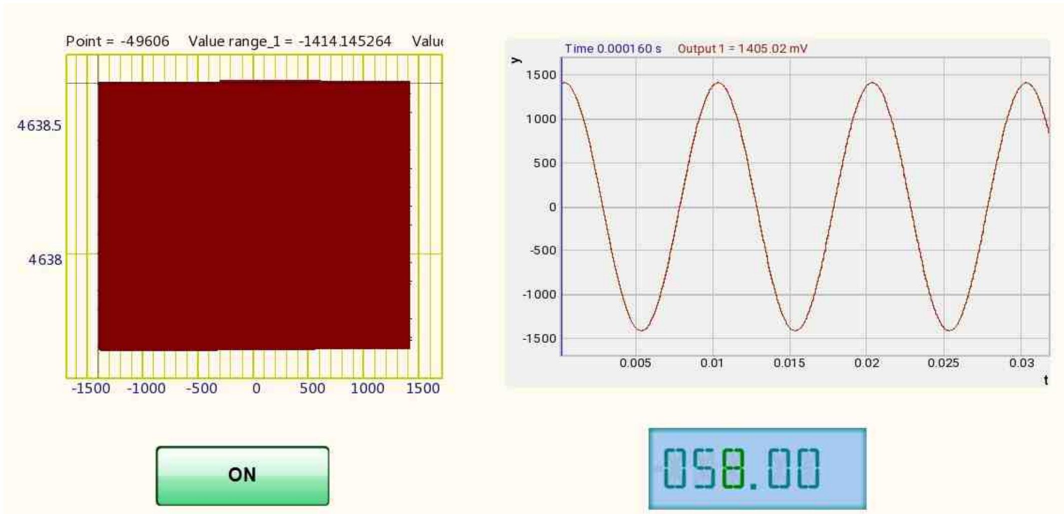
- void CopyData (void) - Copies data to the clipboard.
- void SaveGridToFile(BSTR) - Saves graphics to the BMP file.

Example

Project in SCADA ZETView



Project operation result

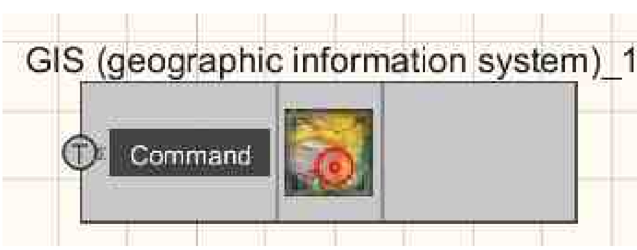


View the example in ZetView

19.3.GIS (geographic information system)

The component "**GIS (geographic information system)**" is designed for collection, storage, analysis and graphical visualization of spatial (geographical) data and related information about the required objects.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Command - adding or changing objects on the map.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- baseImageFile - set the name and location of the background file containing the image in GeoTiff format .
- riversShapeFile - Sets the name and location of the linear water layer file in ERSI ShapeFile format .
- lakesShapeFile - set the name and location of the polygonal water layer file containing the picture in *. shp .
- townsShapeFile - set the name and location of the point layer file for cities, containing a picture in *. shp .
- citiesShapeFile - set the name and location of the polygon layer file for cities, containing a picture in *. shp .
- railroadsShapeFile - set the name and location of the railroads layer file containing the *. shp .
- autoShapeFile - set the name and location of the road layer file containing the *. shp .
- buildingsShapeFile - set the name and location of the building layer file containing the *. shp .
- pointShapeFile - set the name and location of a custom point layer file containing a drawing in *. shp .
- lineShapeFile - set the name and location of a custom line layer file containing a drawing in *. shp .
- pointMarkerFile - set the name and location of the icon file for a custom point layer containing a drawing in *. png .
- srcProjection - set the projection of incoming coordinates, for Example : + proj = longlat + ellps =WGS84 + datum =WGS84 + no_defs . The coordinates in this projection will be used in commands.
- dstProjection - set the current projection of the map files, for Example : + proj = longlat + ellps =WGS84 + datum =WGS84 + no_defs . All coordinates passed in commands will be converted to this projection.
- pointIconsDir - set the name and location of the directory with icons in *. png . These icons will be used when drawing points with user commands.

The list of commands that can be sent to the map:

- point ;< lat >;< lon >;<file name> - draw a point with < lat > latitude, < lon > longitude. At this point, an icon with the appropriate name from the icon directory will be drawn.
- text ;< lat >;< lon >;<text>;<color> - a label <text> will be displayed at a point with < lat > latitude and < lon > longitude, with background color <color>, color in **#XXXXXX** RGB format .

- circle ;< lat >;< lon >;< radius >;<width>;<color> - draw a circle centered on < lat > latitude and < lon > longitude, < radius > - radius, <width> - circle width and <color> is a color in **#XXXXXX RGB format** .

In addition, the following actions can be performed on these commands:
- delete; point ; <lat> ; <lon>
- delete;text ;<lat> ; <lon> ; <text>
- delete;circle ;< lat >;< lon >;< radius >;<width>

deleting elements that are not currently needed on the map.

clear clears the data on the map.

goto ;< lat >;< lon >;<lat1>;<lon1> scales the map to the area with the coordinates of the upper left corner <lat1>,<lon1> and the lower right corner - <lat2>,<lon2>.

goto;start - show the whole map.

snapshot ;< dir >;< name >, allows you to save a PrintScreen, where <DIR> is the directory and name of the image. If the data is not specified, then the data is saved in the current folder with the file name with the current date and time.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- Command - An input text array that allows adding or changing objects on the map (text).

➢**Custom common properties**:
- Are available by the link below. 154

➢**Custom private properties**:

- baseImageFile - Sets the name and location of the background file containing the GeoTiff image (string).
- riversShapeFile - Sets the location of the linear water layer file in ERSI ShapeFile (string) format.

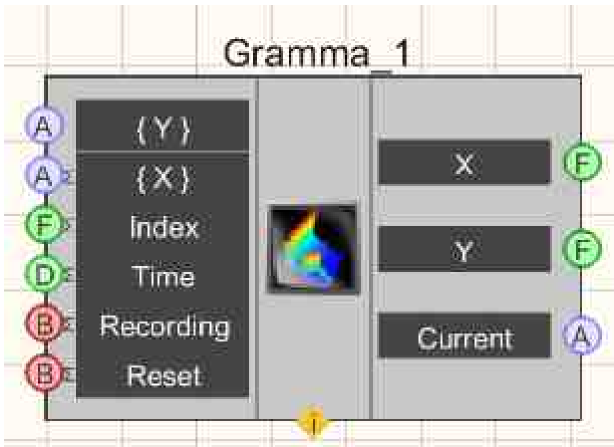
- lakesShapeFile - set the location of the polygonal water layer file containing the picture in *. shp (string).
- townsShapeFile - set the file location of a point layer file for cities containing a picture in *. shp (string).
- citiesShapeFile - set the location of the polygon layer file for cities, containing a picture in *. shp (string).
- railroadsShapeFile - set the location of the railroads layer file containing the *. shp (string).
- autoShapeFile - Sets the name and location of the road layer file containing the *. shp (string).
- buildingsShapeFile - Sets the name and location of the building layer file containing the *. shp (string).
- pointShapeFile - set the file location of a custom point layer containing a picture in *. shp (string).
- lineShapeFile - Sets the file name and location of a custom line layer file containing a *. shp (string).
- pointMarkerFile - Set the name and location of the icon file for a custom point layer containing a drawing in *. png (string).
- srcProjection - set the projection of incoming coordinates, for Example : + proj = longlat + ellps =WGS84 + datum =WGS84 + no_defs . Coordinates in this projection will be used in commands (string).
- dstProjection - set the current projection of the map files, for Example : + proj = longlat + ellps =WGS84 + datum =WGS84 + no_defs . All coordinates passed in commands will be converted to this projection (string).
- pointIconsDir - Set the name and location of the directory with icons in *. png . These icons will be used when drawing points with user commands (string).

19.4.Gramma

The component "**Gramma**" is designed to display numerical values on a specially designed area in a 3D image.

Appearance of the component:

Developer interface	Operator interface



Setting:

➢Input:

- {Y} - an array with values that will be displayed along the y-axis.
- {X} - an array with values that will be displayed for marking along the x-axis.
- Index - fills only the line with the number that came through the "index" contact.
- Time - the current time of the added data.
- Recording - when an impulse is applied to this contact, the data is saved to a separate file (the name and location of the file are specified in the private properties of the component).
- Reset - when an impulse is applied to which, the graphic is cleared.

➢ Output:

- X - The coordinate of the cursor along the X axis on the graphic.
- Y - Cursor coordinate along the Y axis on the graphic.
- Current - An array of values at the given level.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- allowPicture (false) - permission to load a picture from a file.
- PictureFile - setting the name and location of a file containing a picture in *. bmp .
- LineQuantity (10) - number of readings.
- DeltaT (1) - setting the interval between lines along the Y axis (in units)
- XAutoscale (true) - enable/disable autoscaling along the X axis.
- YAutoscale (true) - enable/disable autoscaling along the Y axis.
- XString (Hz) - setting the unit of measure along the X axis.
- YString (s) - setting the unit of measure along the Y axis.
- ZString (mV) - setting the unit of measure along the Z axis.

- BlackMode (false) - set display in black and white mode.
- XAxis (false) - setting the markup along the X axis.
- FileName - Sets the name and location of the file where the data will be saved as a separate file.
- FillType (consecutively) - setting the type of gram filling: sequential; by indexes;
- UseDateTime (false) - setting the use of time and date for the Y-axis label;

Note :
When selecting the "by index" property, the input array Y fills only the line with the number that came from the "index" contact.
When choosing the "sequential" property, the input array Y is written to the end and the picture moves.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- {Y} - Input numeric array with values to be displayed on the y-axis (any number).
- {X} - Input numeric array with values to be displayed on the x-axis (any number).
- Index - The value that contains the index (any number).
- Time - A value that contains the current time of the added data.

➤**Custom common properties**:
- Are available by the link below. 154

➤**Custom private properties**:

- VARIANT_BOOL allowPicture - Set the permission to load a picture from a file:
  true - Permission to load a picture from a file is enabled;
  false - Permission to load a picture from a file is disabled.
- BSTR PictureFile - Set the name of the file containing the picture in *. bmp (string).
- LONG LineQuantity - Set the number of samples (number).
- FLOAT DeltaT - Set line spacing in Y units (number).
- VARIANT_BOOL XAutoscale - Set autoscale on the X axis:
  true - X-axis autoscale enabled
  false - X-axis autoscale disabled.

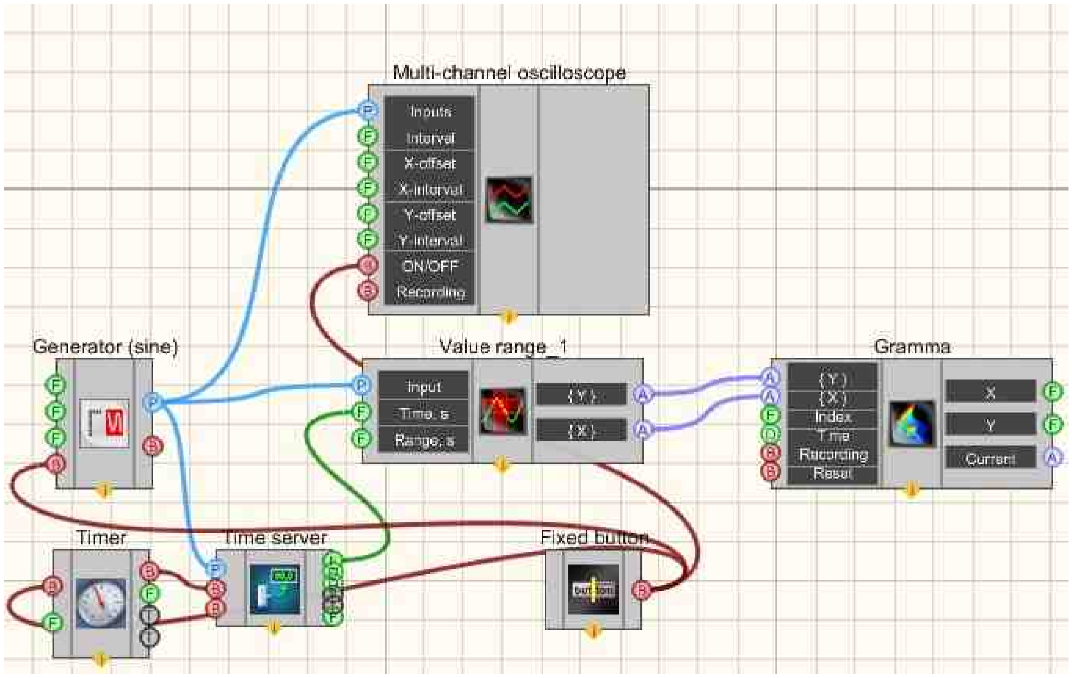
- VARIANT_BOOL YAutoscale - Set autoscale along the Y axis:
  true - Y-axis autoscale enabled
  false - Autoscale on the Y axis is disabled.
- BSTR XString - Sets the unit of measure for the X-axis (string).
- BSTR YString - Set the unit of measure for the Y-axis (string).
- BSTR ZString - Sets the unit of measure for the Z-axis (string).
- VARIANT_BOOL BlackMode - Set display in black and white mode:
  true - Display in black and white mode is enabled;
  false - Display in black and white mode is disabled.
- VARIANT_BOOL XAxis - Sets the use of layout on the X axis:
  true - X-axis markup is enabled;
  false - X-axis markup is disabled.
- BSTR FileName - Sets the file name for writing the results to a *.dtu file (after setting this parameter, the file is written to the specified file).
- BSTR FillType - Setting the type of gram filling: sequential; by indexes.
- VARIANT_BOOL UseDateTime - Set the use of time and date for the Y-axis label;

➢**Methods:**

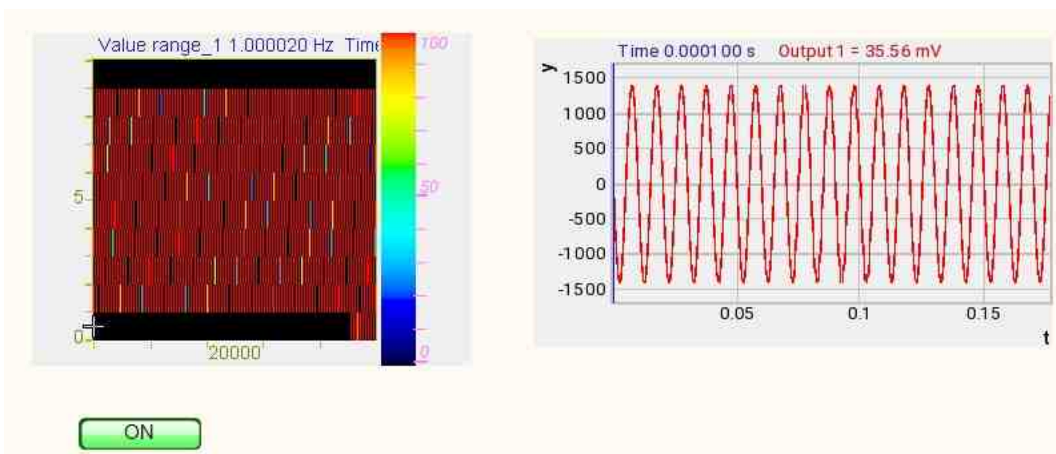
- void CopyPicture(void) - Copies the graphic to the clipboard.

Example No. 1

Project in SCADA ZETView



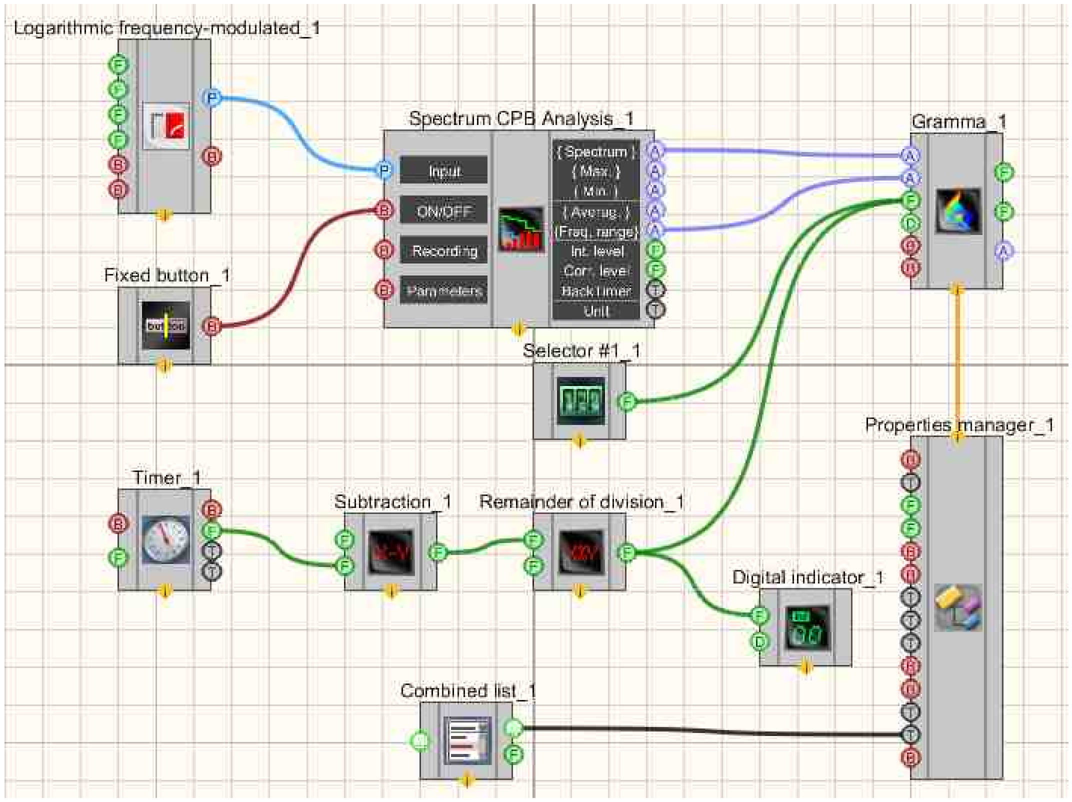
Project operation result



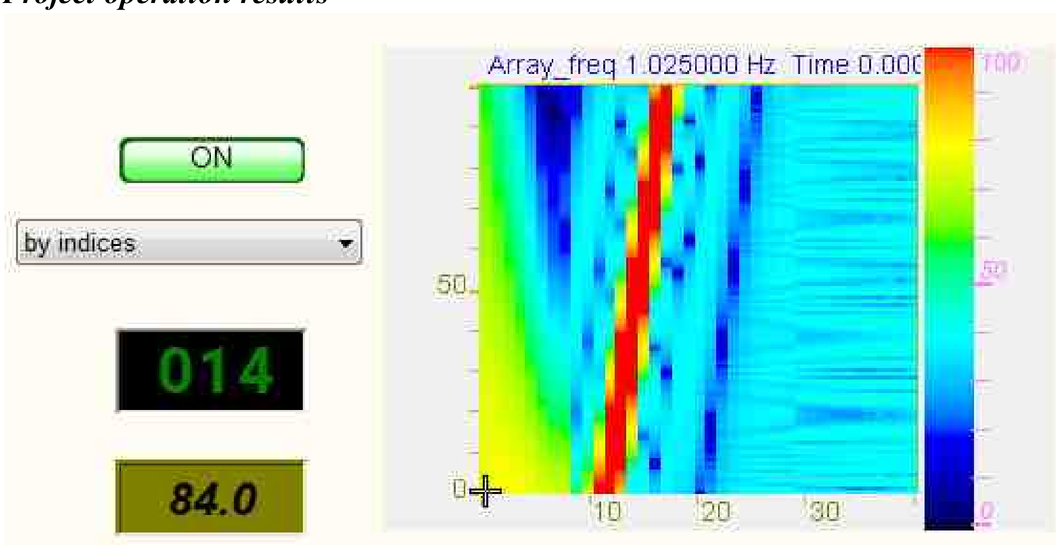
View the example in ZetView

Example No. 2

Project in SCADA ZETView



Project operation results

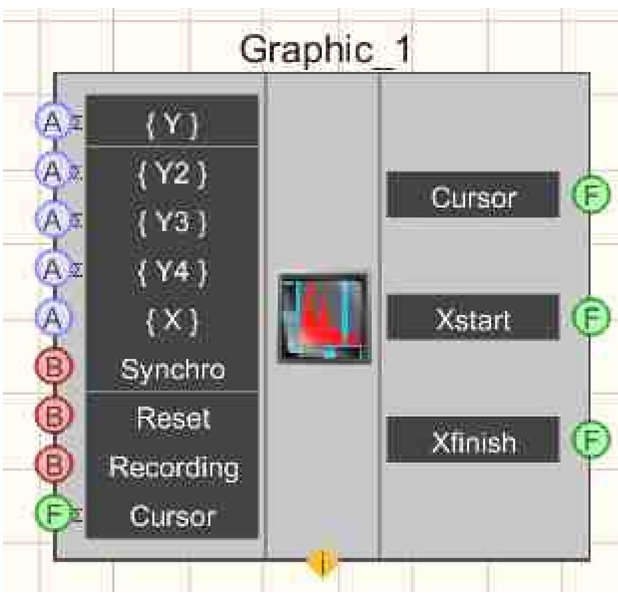
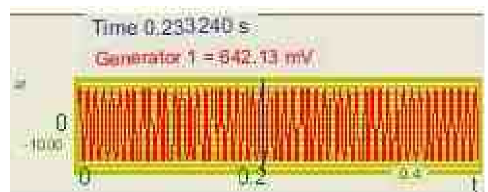


View the example in ZetView

19.5.Graphic

The component "**Graphic**" is designed to display data arrays in the form of a graphic. Unlike the Multi-channel oscilloscope component, which is designed to display the waveform and work with measurement channels, the component. One component can be used to display multiple graphics (array sizes must match).

Appearance of the component:

<table>
<tr><td>Developer interface</td><td>Operator interface</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Setting:

➢Input:

-
- {Y} - getting an array of data displayed on the graphic along the Y axis. The component allows you to display several graphics in the same axes. Contact interface - "Numerical array". The most commonly connected components are the analysis components (FFT spectrum Analyzers, Spectrum CPB Analysis).
- {Y2} - getting an array of data displayed on the graphic along the second Y2 axis. The component allows you to display multiple graphics in the same axes. Contact interface - "Numerical array". The most commonly connected components are the analysis components (FFT spectrum Analyzers, Spectrum CPB Analysis).

- {Y3} - getting an array of data displayed on the graphic along the third Y3 axis. The component allows you to display multiple graphics in the same axes. Contact interface - "Numerical array". The most commonly connected components are the analysis components (FFT spectrum Analyzers, Spectrum CPB Analysis).
- {Y4} - getting an array of data displayed on the graphic along the fourth Y4 axis. The component allows you to display multiple graphics in the same axes. Contact interface - "Numerical array". The most commonly connected components are the analysis components (FFT spectrum Analyzers, Spectrum CPB Analysis).
- {X} - Data displayed on the X axis. The contact does not support multiple connections. Contact interface - "Numerical array".
- Cursor - Determines the position of the cursor on the graphic.
- Synchro - Impulse to update the graphic. The contact interface is a boolean value (0/1).
- Reset - An impulse to clear the graphic field. The contact interface is a boolean value (0/1).
- Recording - An impulse to record the displayed data to a file.
- Cursor - Position of the cursor on the graphic.

➢ Output :

Cursor - Allows you to save the position of the cursor on the graphic.
XStart - Allows you to save the start of the display along the X axis.
XFinish - Allows you to save the end of the display along the X axis.

To display data, it is important that the component from which the array comes from has an identifier (common property m_sHelpString). The IDs of the components connected to the "Y" and "X" inputs are used as the name of the graphic or x-axis and are displayed in the graph's legend. If the component from which the data comes from does not have an identifier, then the data will not be displayed on the graphic. When receiving multiple arrays from one component, identifiers are specified separated by semicolons. The identifiers are specified in the order of the output pads.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- XAutoscale (true) - set the enable/disable autoscaling along the X axis.
- YAutoscale (true) - set the enable/disable autoscaling along the Y axis.
- xystatus (y(n)) - set the graphic display mode: Plot in polar coordinates

1. y(n) - the array of data received at the input "Y" will be displayed along the Y axis, while the numbers of the array elements will be displayed along the X axis
2. y(x) - is uniform. Display along the Y axis of the data array that came to the "Y" input, along the X axis - the data array that came to the "X" input. Also used to display data that has a non-uniform distribution of values - this requires you to specify the type of sweep.
3. y(t) - is continuous.
4. y(x) - is really.

- XString - setting the unit of measure along the X axis.
- YString - setting the unit of measure along the Y axis.
- Referense (0.001) - value for setting the calculation of levels in dB.
- Ystart (0) - start of display along the Y axis.
- Yfinish (1) - the end of the display along the Y axis.
- FileName - setting the name and location of the file in which the data will be saved as a separate *.dtu, *.dtx,* file. xls , *. png .
- xprecision (0.01) - precision of displaying values along the x-axis.
- yprecision (0.01) - the accuracy of displaying values along the Y axis.
- LineWidth (1) - setting the thickness of the graphic line.
- BackColor (ece9d8) - background color.
- GridColor (c0c000) - grid color.
- typeline (Horizontal) - graphic line type:
  1. Horizontals.
  2. Polygonals.
- typeyaxis (Uniform) - vertical scale sweep:
  1. Uniform.
  2. Logarithmic.
  3. Decibel.
- typexaxis (Uniform) - sweep of the horizontal scale:
  1. Uniform.
  2. Logarithmic.
  3. Octave band.
- ColorAdjustment (False) - permission to adjust graphic colors:
  1. True - allow customization of graphic colors.
  2. False - do not allow customization of graphic colors.
- GraphColors(128;32768;8388608;255;65280;16711680;8388672;8421376;8388736;12615680;16384;16711935;16711808;16744576;16744448;8421440;8388863;64;64;16744576;8421631;10485760;4194304;16744703; 9125927;52582;6513646;2500301;3355597;17803;7737549;7877325;9127773;16760576;10156544;2263842;2237106;9639167;13828244;9109504;13487360;52582;9639167;15597806;16724123;5197647;8983278;9109504;9145088; 35653;13434880;25600;8721863;3107669;5197615;14053594;13459258;3816075;9109643; 550283;) - graphic colors:
- Xstart (0.000000) - X start display.
- Xfinish (100.0) - end of display along the X axis.
- Filling (Without filling) - type of filling of the graphic.
- CommentToGraph (Signal implementation) - a comment to the graphic that will be displayed in the file after saving.

Possibilities of saving data in Excel format :

We launch the ZETView program, data in Excel format can be saved through the graphic component and the multichannel oscilloscope. The process of saving them is the same.

Let's take a look at the example of the graphic component.

1. Option to save without specifying a file name:
we take the graphic component and to the input of the "Record" contact and connect the "Momentary Button". Next, run the project -> click on the "Non-fix button" -> select the format "Spreadsheet *. xls ".

2. Option to save with a name through the property manager. We connect the property manager to the graphic component. Next, we connect the component "Catalog Selection" and "Text field". We add them using the "Add strings" component. After we add the resulting string with the format ".xls " .

Note :
Question. The program does not record data to the array and does not pass data to the graphic (A more complete question and answer is in the " Programmable " component component ")
Answer. The "Graphic" component has a property " xystatus ", which is responsible for the appearance of the graphic axes. In the "y(n)" option you choose, the graphic doesn't care about the array on the "X" input pin and builds a new graphic as it receives arrays on the "Y" pin. It is important to remember that the field " m_sHelpString " is not empty, because the line in this field specifies the name of the graphic in the legend, and if there is no name, there is no graphic. To show the X-axis as a date, you must select the option "y(x) - real".

Description for Example No.3

There are two methods for assessing trends:

1. Parametric — consider the time series as a smooth function of t: X (t) = f (t) , t = 1... n ; {\displaystyle X(t)=f(t),t=1...n;} .First, one or more admissible function types f (t) {\displaystyle f(t)} ; then, various methods (for example , LSM) evaluate the parameters of these functions, after which, based on the verification of the adequacy criteria, the final trend model is selected. Of great importance for practical applications are linearized trends, that is, trends reduced to a linear form with respect to parameters using certain algebraic transformations.

2. Non-parametric - these are different methods of smoothing the original time series - moving averages (simple, weighted), exponential smoothing. These methods are used for both trend assessment and forecasting. They are useful when it is not possible to find a suitable function for trend estimation.

In this project, these two types are averaged. In OLS, you can choose from any of six trend types.

Trend - the main tendency for something to change: in Example , in mathematics - a time series . Trends can be described by various equations - linear, logarithmic, exponential, and so on. The actual trend type is established on the basis of the selection of its functional model by statistical methods or by smoothing the initial time series .

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- {Y} - Input numeric array with values to be displayed on the y-axis (any number).
- {X} - Input numeric array with values to be displayed on the x-axis (any number).
- Cursor - Sets the position of the cursor on the graphic.

➢**Custom common properties**:

- Are available by the link below. 154

➢**Custom private properties**:

- VARIANT_BOOL XAutoscale - Set autoscale on the X axis:
  true - Autoscale along the X axis is enabled;
  false - X-axis autoscale disabled.
- VARIANT_BOOL YAutoscale - Set autoscale along the Y axis:
  true - Autoscale on the Y axis is ON;
  false - Autoscale on the Y axis is OFF.
- BSTR XYStatus - Set the graphic display mode: y(n); y(x) - is uniform; y(t) - is continuous; y(x) - really (string).
- BSTR XString - Set the unit of measure for the X axis (string).
- BSTR YString - Set the unit of measure along the Y axis (string).
- FLOAT Referense - Set value for calculating levels in dB (any value).
- FLOAT Ystart - Set the start of the display along the Y axis (any value).
- FLOAT Yfinish - Set the end of the display along the Y axis (any value).
- BSTR FileName - Set the file name for writing the results to a *.dtu file (after setting this parameter, the file is written to the specified file).
- BSTR XPrecision - Set the display precision along the X axis: 1;0.1;0.01;0.001;0.0001;0.00001;0.000001 (string).
- BSTR YPrecision - Set the display precision along the Y axis: 1;0.1;0.01;0.001;0.0001;0.00001;0.000001 (string).
- LONG LineWidth - Set the width of all graphic lines (number).
- LONG BackColor - Set the background color (number).
- LONG GridColor - Set the grid color (number).
- BSTR TypeLine - Set graphic line type: Horizontals; Polygonals (string).
- BSTR TypeYAxis - Set the vertical scale sweep: Uniform; logarithmic; Decibel (string).

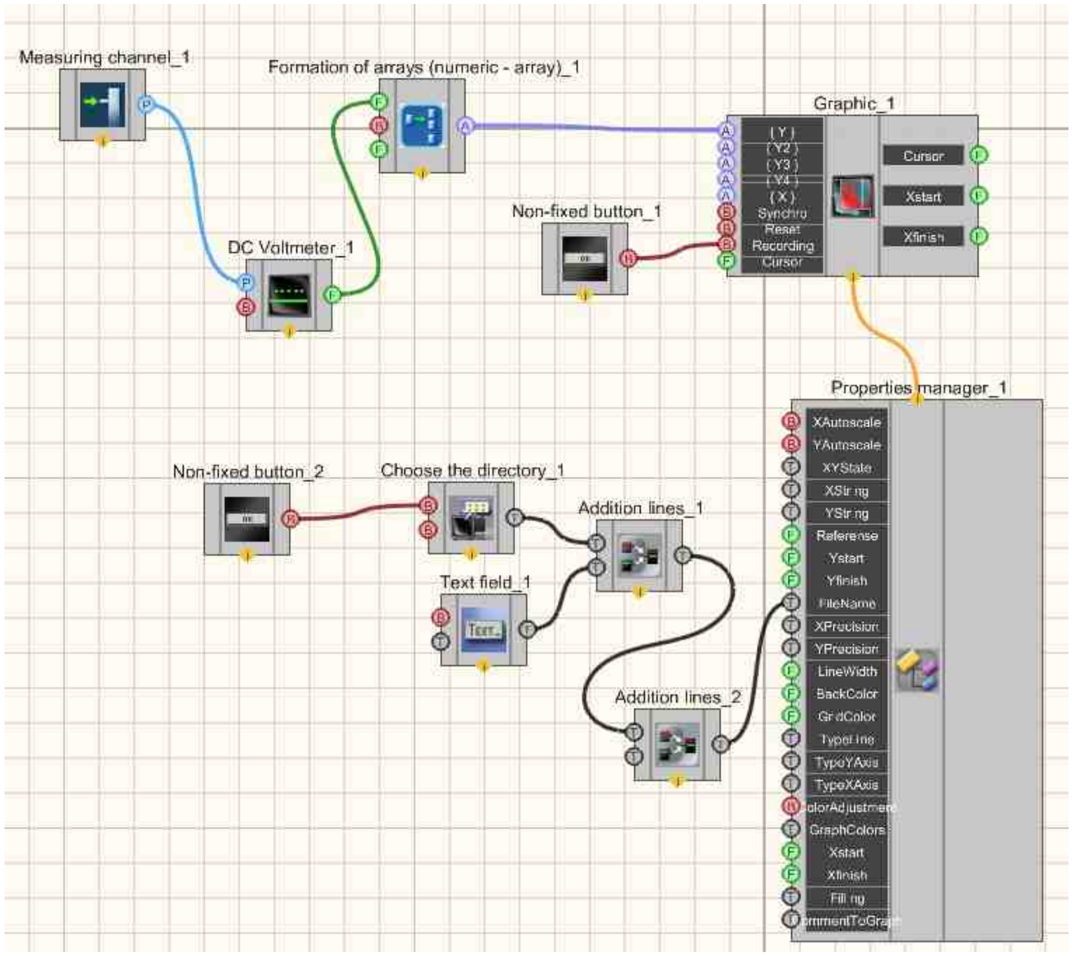
- BSTR TypeXAxis - Set the horizontal scale sweep: Uniform; logarithmic; Octave band (string).
- VARIANT_BOOL ColorAdjustment - Set the resolution of graphic color settings:
  true - Permission to customize graphic colors is ON;
  false - Permission to customize graphic colors is OFF.
- BSTR GraphColors - Set the color of the graphs (string).
- FLOAT Xstart - Set the display start on the X axis (number).
- FLOAT Xfinish - Set the end of the display along the X axis (number).
- BSTR Filling - Set the type of graphic filling: Without filling;Positive;Alternating (string).
- BSTR CommentToGraph (Signal implementation) - a comment to the graphic that will be displayed in the file after saving.

➢**Methods:**

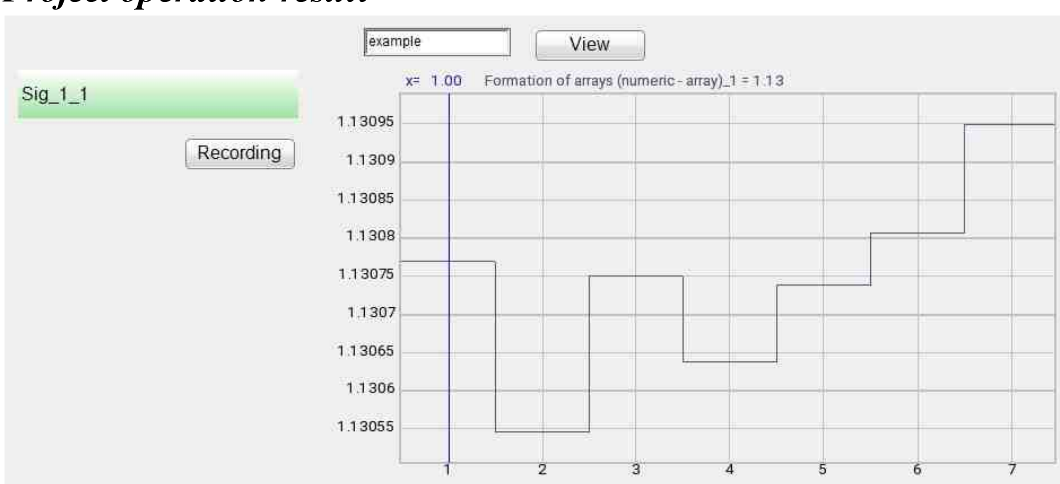
- void ClearGrafs (void) - Clears the graphics.
- void CopyPicture (void) - Copies the graphic to the clipboard.
- void CopyData (void) - Copies data to the clipboard.
- void UpdateGrafs (void) - Updates the graphics.
- void SetGrafCursorValue (FLOAT) - Set the cursor by value (the blue bar of the oscilloscope cursor is set next to the set value).
- void SetGrafCursorPosition (LONG) - Set the cursor to the array element number (the blue bar of the oscilloscope cursor is set next to the specified array element number).

Example No. 2

Project in SCADA ZETView



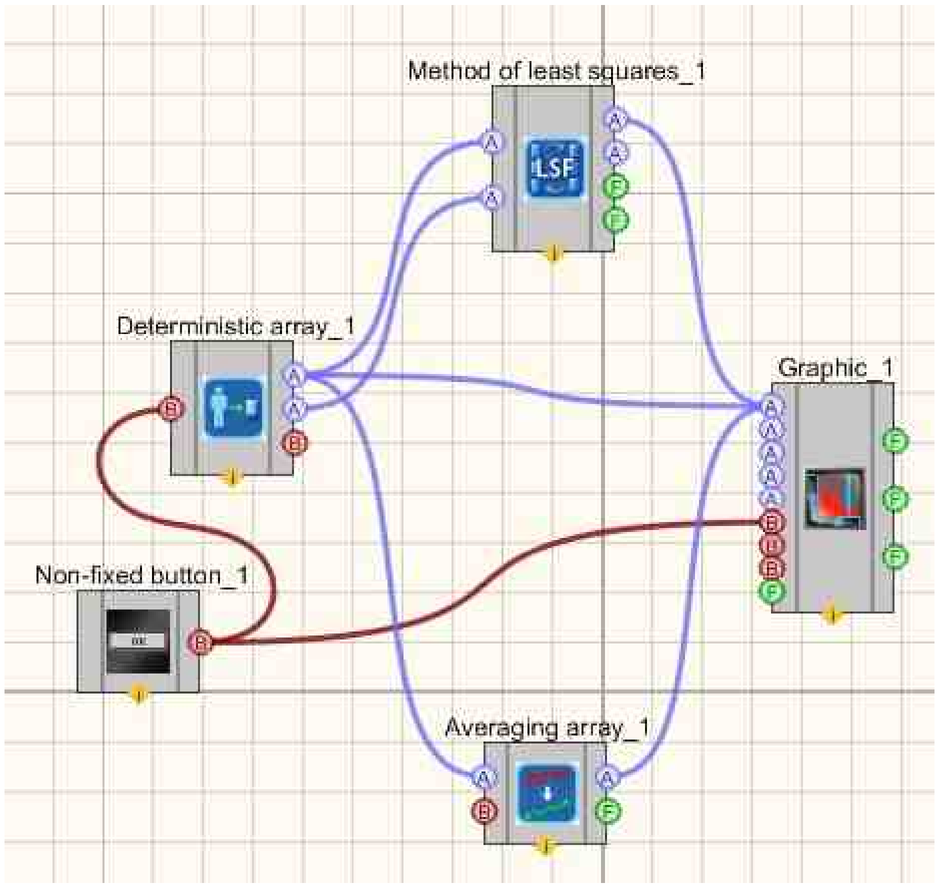
Project operation result



View the example in ZetView

Example No. 3

Project in SCADA ZETView



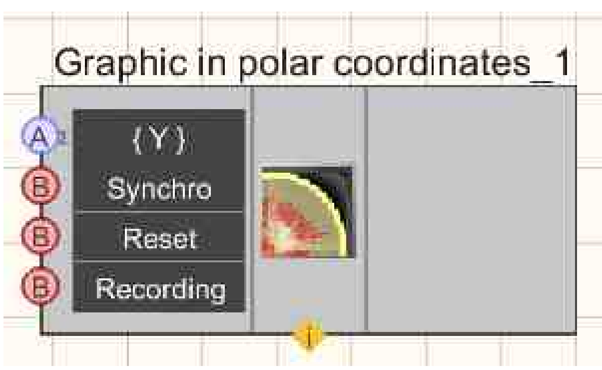
Project operation result

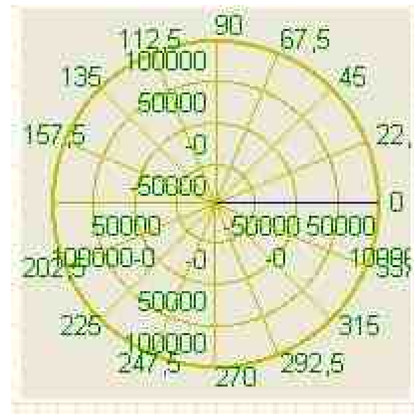
View the example in ZetView

19.6.Graphic in polar coordinates

The component "**Graphic in polar coordinates**" is designed to display data in polar coordinates.

Appearance of the component:

Developer interface	Operator interface
	



Setting:

➢Input:

- {Y} - an array with values that will be displayed along the y-axis.
- Synchronization - synchronization is carried out by transferring data to the drawing field.
- Reset - when an impulse is applied to which, the graphic is cleared.
- Recording - when an impulse is applied to this contact, the data is saved to a separate file (the name and location of the file are specified in the private properties of the component).
-

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- YAutoscale (true) - Set the enable/disable autoscaling along the Y axis.
- YString - set the unit of measure along the Y axis.
- Ystart (0) - Set the start of display along the Y axis.
- Yfinish (1) - Set the end of the display along the Y axis.
- FileName - Set the name and location of the file where the data will be saved as a separate file.
- xprecision (0.01) - Set the precision of displaying values along the x-axis.
- yprecision (0.01) - Set the accuracy of displaying values along the Y axis.
- BackColor (ece9d8) - Set the background color.
- GridColor (c0c000) - Set the grid color.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- {Y} - Input numeric array with values to be displayed on the y-axis (any number).
- {X} - Input numeric array with values to be displayed on the x-axis (any number).

➢**Custom common properties**:

- Are available by the link below. 154

➢**Custom private properties**:

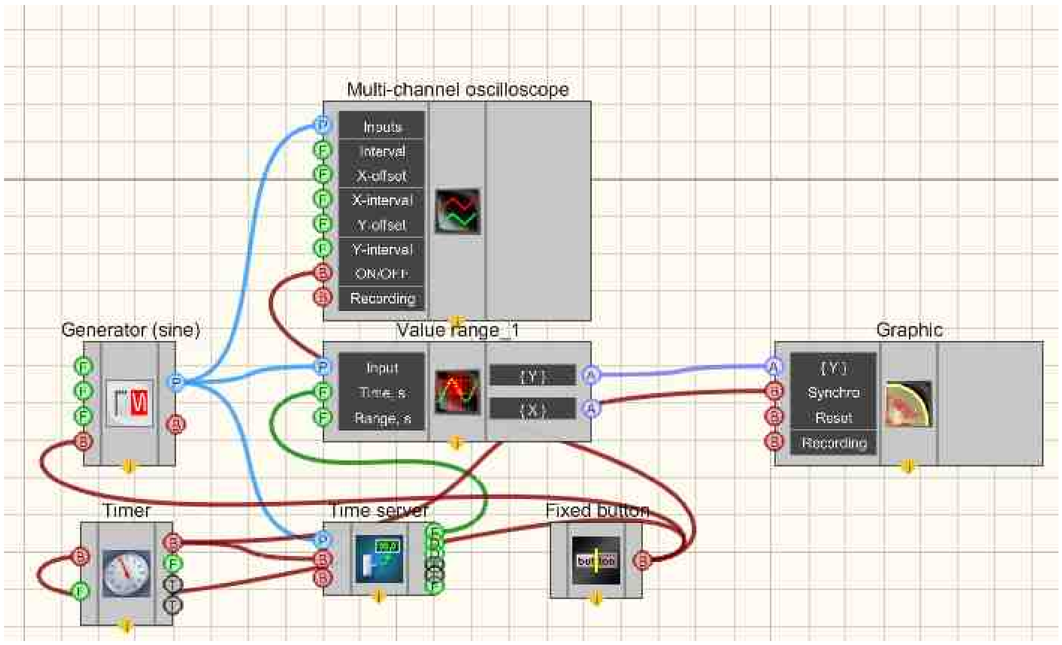
- VARIANT_BOOL YAutoscale - Set autoscale along the Y axis:
  true - Autoscale along the Y axis is ON;
  false - Autoscale along the Y axis is OFF.
- BSTR YString - Set the unit of measure along the Y axis (string).
- FLOAT Ystart - Set the start of the display along the Y axis (number).
- FLOAT Yfinish - Set the end of the display along the Y axis (number).
- BSTR FileName - Set the file name for record the results to a *.dtu file (after setting this parameter, the file is recorded to the specified file).
- BSTR XPrecision - Set the display accuracy along the X axis: 1; 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- BSTR YPrecision - Set the display precision along the Y axis: 1; 0.1; 0.01; 0.001; 0.0001; 0.00001; 0.000001 (string).
- LONG BackColor - Set the background color (number).
- LONG GridColor - Set the grid color (number).

➢**Methods:**

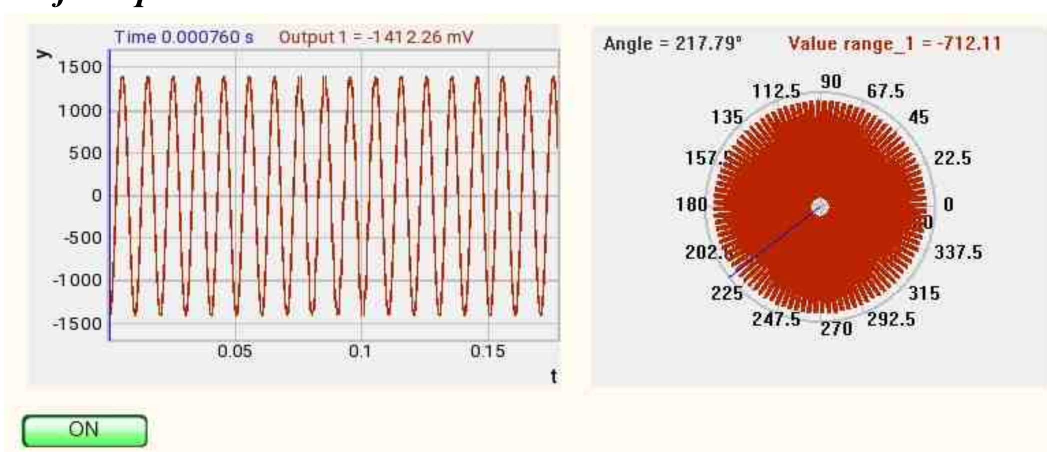
- void CopyPicture (void) - Copy the graphic to the clipboard.
- void CopyData (void) - Copy data to clipboard.
-

Example

Project in SCADA ZETView



Project operation result

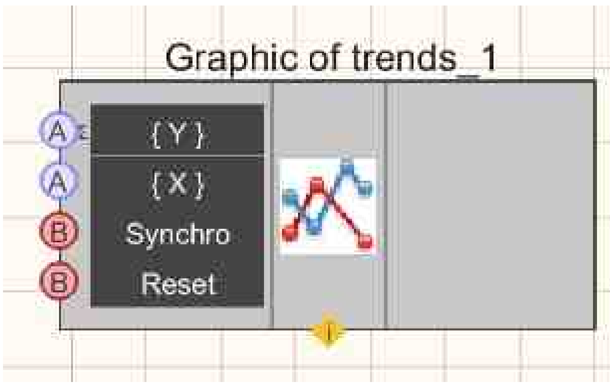


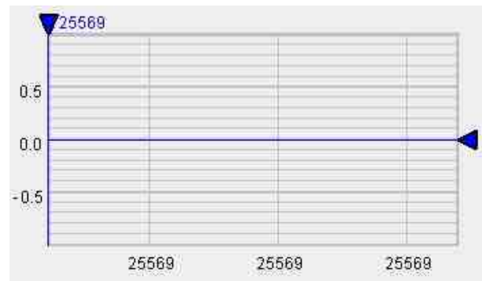
View the example in ZetView

19.7.Graphic of trends

The component "**Graphic of trends**" is designed for displaying trend data.

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Setting:

➢Input:

- {Y Data } - an array with values that will be displayed along the y - axis.
- {X Data } - an array with values that will be displayed along the x - axis .
- Synchro - synchronization is carried out by transferring data to the drawing field.
- Reset - when an impulse is applied to which, the graphic is cleared.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- XAutoscale (false) - enable/disable autoscaling along the X axis.
- YAutoscale (false) - enable/disable autoscaling along the Y axis.
- MathLeftX (0.0) - The first visible value along the X axis.
- MathRightX (100.0) - Last visible X value.
- ApplyLimit (false) - Apply limits on the visible ranges along the axes.
- LimitMinY (-0.1) - The minimum value along the Y axis.
- LimitMaxY (1.0) - The maximum value along the Y axis.
- MathMinY (-1.0) - The minimum visible value along the Y axis.
- MathMaxY (1.0) - The maximum visible value along the Y axis.
- LimitDeltaY (0.001) - Y-axis scaling limit.
- GridMode - graphic display mode: combined; separate; seismic;

- VerticalGrid (false) - The vertical x-axis.
- TypeAxisX - Type of X axis: time; numbers;
- LineWidth (1) - The thickness of the graphics.
- DrawCursorValue (false) - Show graphic values on vertical cursors.
- UnitAxisX - Units along the X axis.
- UnitAxisY - Units of measure along the Y axis.
- NameAxisX - X axis name.
- NameAxisY - The name of the Y axis.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- {Y} - Input numeric array with values to be displayed on the y-axis (any number).
- {X} - Input numeric array with values to be displayed on the x-axis (any number).

➢**Custom common properties**:
- Are available by the link below. 154

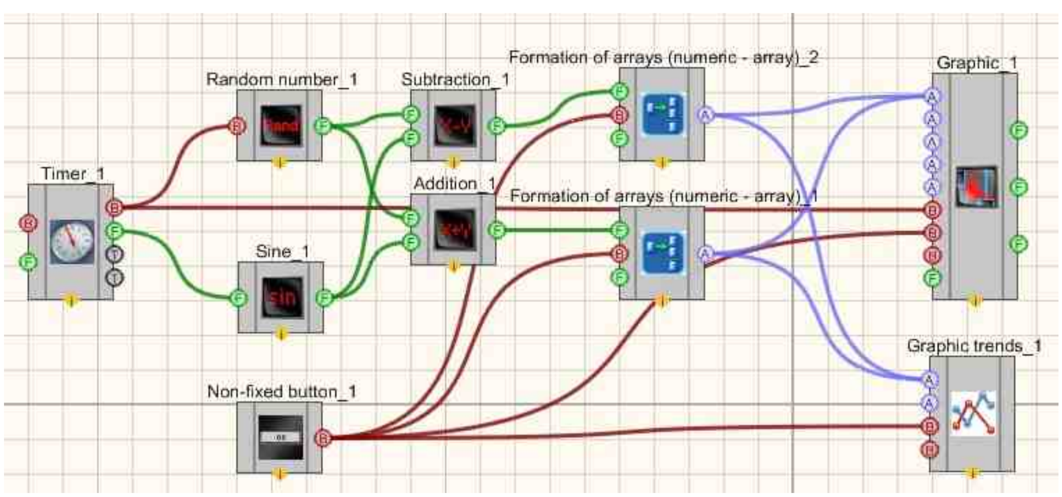
➢**Custom private properties**:

-
- VARIANT_BOOL XAutoscale - Set to enable/disable autoscaling on the X axis:
  true - enable autoscaling along the X axis;
  false - turn off autoscaling along the X axis.
- VARIANT_BOOL YAutoscale - Set to enable/disable autoscaling along the Y axis:
  true - enable autoscaling along the Y axis;
  false - turn off autoscaling along the Y axis.
- DOUBLE MathLeftX (0.0) - First visible value along the X axis.
- DOUBLE MathRightX (100.0) - The last visible value on the X axis.
- VARIANT_BOOL ApplyLimit - Sets the application of limits on the visible ranges along the axes:
  true - apply restrictions on visible ranges along the axes;
  false - do not apply restrictions on visible ranges along the axes.
- DOUBLE LimitMinY (-0.1) - The minimum value along the Y axis.

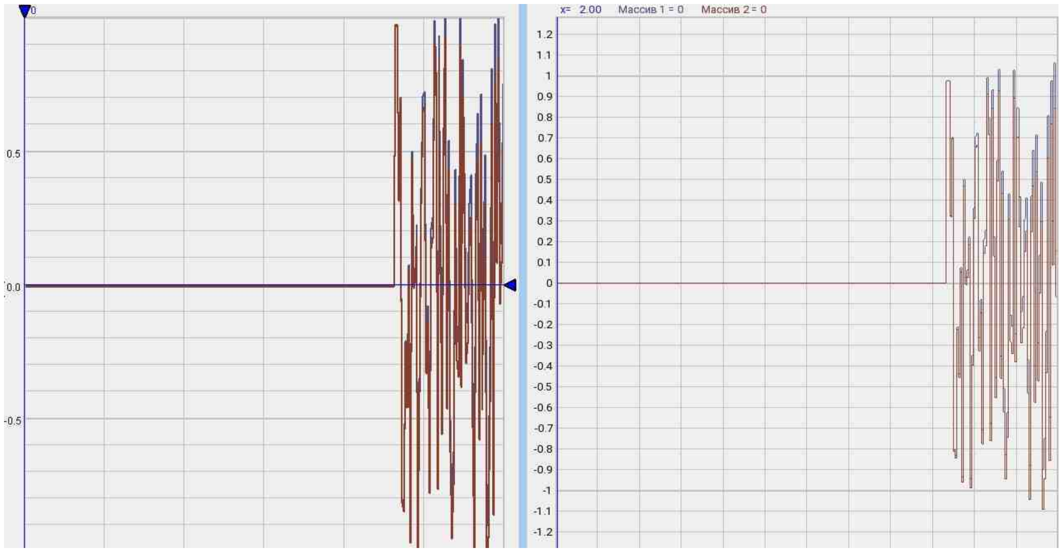
- DOUBLE LimitMaxY (1.0) - The maximum value along the Y axis.
- DOUBLE MathMinY (-1.0) - The minimum visible value along the Y axis.
- DOUBLE MathMaxY (1.0) - The maximum visible value along the Y axis.
- DOUBLE LimitDeltaY (0.001) - Limit of scaling along the Y axis.
- BSTR GridMode - graphic display mode: combined; separated; seismic;
- VARIANT_BOOL VerticalGrid - Set the vertical x-axis:
  true - show values of the vertical X axis:;
  false - do not show values of the vertical x-axis.
- BSTR TypeAxisX - Type of X axis: time; numbers;
- LONG LineWidth (1) - The thickness of the graphics.
- VARIANT_BOOL DrawCursorValue - Setting to show graphic values on vertical cursors:
  true - show graphic values on vertical cursors;
  false - do not show graphic values on vertical cursors.
- BSTR UnitAxisX - Units of measure along the X axis.
- BSTR UnitAxisY - Units of measure along the Y axis.
- BSTR NameAxisX - X axis name.
- BSTR NameAxisY - Y axis name.

Example

Project in SCADA ZETView



Project operation result



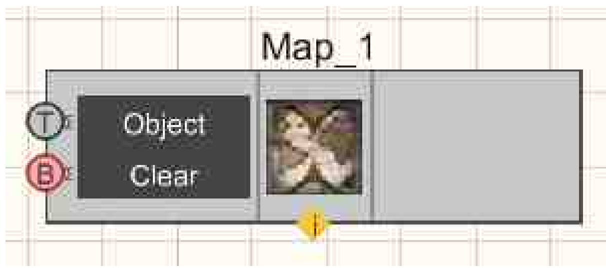
View the example in ZetView

19.8.Map

The component "**Map**" is designed to display maps and allows you to make marks on it.

Appearance of the component:

Developer interface	Operator interface



Setting:

➢Input:
- Object - adding a new object to the map.
- Clear - when a impulse is applied to which, the card is cleared.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- MapFileName - picture with a map in *. png (string).
- LeftTopLat (0) - GPS coordinates of the top left corner (latitude).
- LeftTopLong (0) - GPS coordinates of the top left corner (longitude).
- RightBottomLat (0) - GPS coordinates of the bottom right corner (latitude).
- RightBottomLong (0) - GPS coordinates of the bottom right corner (longitude).
- StaticObject - Static objects separated by a semicolon. The format is latitude, longitude, radius1, radius2, color, label.
- ToolTip - auxiliary tooltip. They are written with a semicolon.
- fKmPerDegree - determines the kilometer/degree ratio for drawing a circle on the map.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- Object - String value for adding a new object to the map.

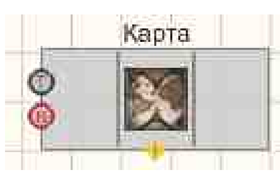
➢**Custom common properties**:
- Are available by the link below. 154

➢**Custom private properties**:

- BSTR MapFileName - Installing a picture with a map in *. png (string).
- FLOAT LeftTopLat - Set the GPS coordinates of the upper left corner (latitude) (number).
- FLOAT LeftTopLong - Set the GPS coordinates of the upper left corner (longitude) (number).
- FLOAT RightBottomLat - Set the GPS coordinates of the bottom right corner (latitude) (number).
- FLOAT RightBottomLong - Set the GPS coordinates of the bottom right corner (longitude) (number).
- BSTR StaticObjects - Setting static objects through a semicolon. Format - latitude, longitude, radius1, radius2, color, label (string).
- BSTR ToolTip - Set auxiliary tooltip. They are written separated by a semicolon (string).
- FLOAT fKmPerDegree - Sets the kilometer/degree ratio for drawing a circle on the map (number).

Example No. 1

Project in SCADA ZETView

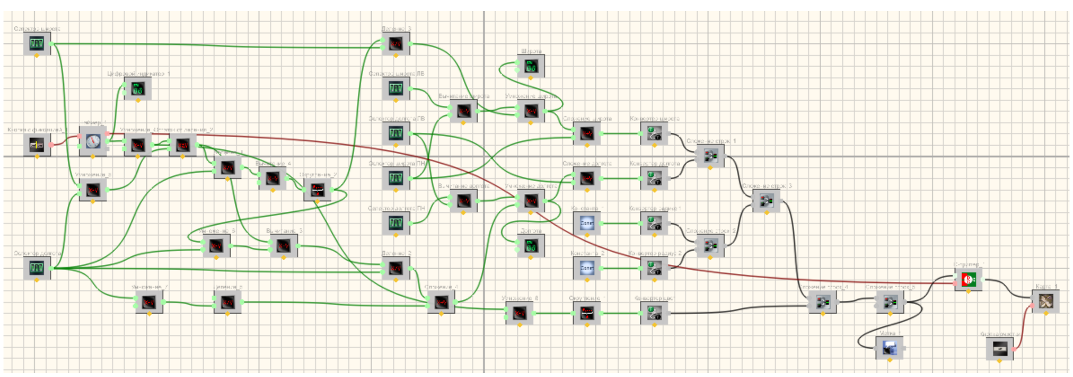


Project operation result

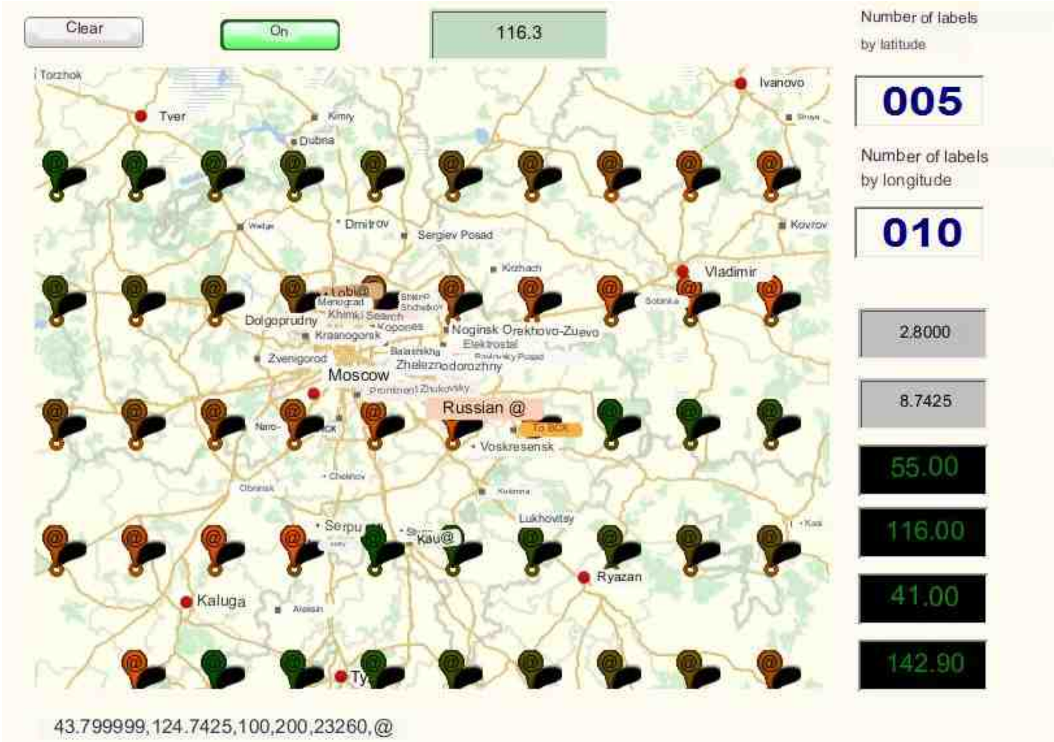
View the example in ZetView

Example No. 2

Project in SCADA ZETView



Project operation result

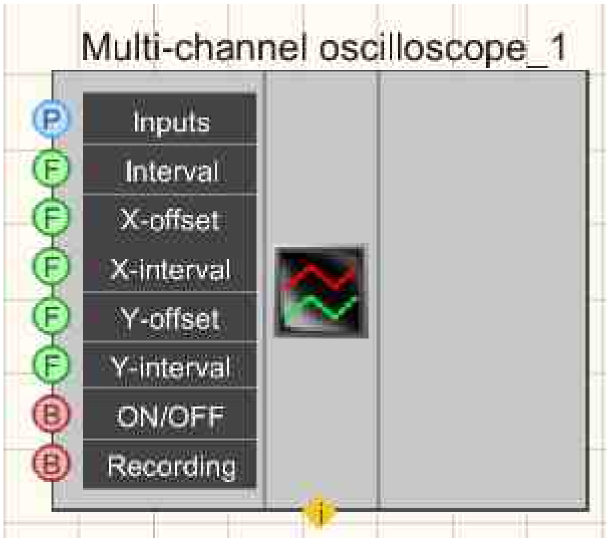
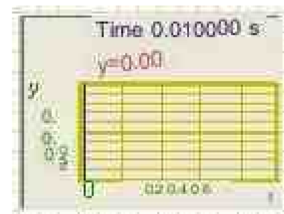


View the example in ZetView

19.9.Multi-channel oscilloscope

The component "**Multi-channel oscilloscope**" is an interface for interaction with the virtual device ZETLab "Multi-channel oscilloscope"

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Setting:

➢Input:

- Inputs - a list of channels that you want to display.
- Interval - display interval, s.
- X-offset - with which you can shift the waveform along the x-axis.
- X-interval - to set the interval along the X-axis.
- Y-offset - with which you can shift the waveform along the y-axis.
- Y-interval - .to set the Y-interval.
- On/Off - Turns the multichannel oscilloscope on and off.
- Recording - when an impulse is applied to this contact, the data is saved to a separate file (the name and location of the file are specified in the private properties of the component).

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Interval (1) - set the display interval, s.
- XStart (0) - set the zero offset along the X axis, s.
- XInterval (1) - display length along the X axis, s.
- YStart (0) - set the zero offset along the Y axis (in units).

- YInterval (1) - set the display interval along the Y axis (in units).
- update (1) - display update (0.1 or 1 s).
- Synchro (false) - status of synchronization (enabled or disabled).
- SynchroLevel (0) - synchronization level (in units).
- Activate (true) - the status of the signal (on or off).
- Autoscale (false) - autoscale.
- FileName - set the name and location of the file in which the data will be saved as a separate *.dtu, *.dtx,* file. xls , *. png .
- Decimation (1) - decimation of the original signal: 1; 10; 100; 1000; 10000.

Questions and answers to users:

Question: Please clarify the question of how to display the boundary (limit) values on a multichannel oscilloscope in ZETVIEW
The project looks like
this . signal - a multi-channel oscilloscope
tried all the options program additional question about 7010DS at the start of work, the oscilloscope graphic shows a position of 0 µm then flies off by a certain value on Example -10 µm - IMHO this is due to the resulting stretching of the sensors during the installation of sensors - the question itself is whether it is possible to accept the actual value of the sensor for zero and make a software reset or adjustment on it, so that the sensor thinks that the position -10 is zero (there is no such possibility through remote access over the network in the properties of the sensors) ???

Answer:
Good afternoon
You can see the setting for displaying boundary values in the Example :
https://goo.gl/photos/tDMGRy9qGfYKYP498 .
Sensor adjustment by standard methods is carried out as follows: you need to open the sensor properties in the device manager. Then select the “Offset” tab, select the offset status “on.” and set the value that the oscilloscope is currently showing. To obtain a more accurate offset value, it is better to use a DC voltmeter. If I understand you correctly, then you do not have such an opportunity. In this case, you can use the "Formula" program from the Automation menu of the ZETLAB panel (https://zetlab.com/shop/programmnoe-obespechenie/funktsii-zetlab/avtomatizatsiya/formula/), where you can set the required offset.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**

- Inputs - Measuring channel, the signal in which you want to display (from 0 to (number of channels - 1)).
- Interval - An input numeric array to set the value of times to set the display interval (any number).
- X-offset - An input numeric array with values that can be used to shift the waveform along the x-axis (any number).
- X-interval - An input numeric array with values to set the x-interval (any number).
- Y-offset - An input numeric array with values that can be used to shift the waveform along the y-axis (any number).
- Y-interval - An input numeric array with values to set the Y-interval (any number).

➢**Custom common properties**:
- Are available by the link below. 154

➢**Custom private properties**:

- FLOAT Interval - Set the display interval, s (number).
- FLOAT XStart - Set zero offset on the X axis, s (number).
- FLOAT XInterval - Set the display length along the X axis, s (number).
- FLOAT YStart - Set the zero offset on the Y axis (in units) (number).
- FLOAT YInterval - Set the display interval along the Y axis (in units) (number).
- BSTR Update - Update installation: 0.1 s; 1 s (string).
- VARIANT_BOOL Synchro - Synchro setting:
  true - Synchronization is enabled;
  false - Synchronization is disabled.
- FLOAT SynchroLevel - Set the synchronization level (in units) (string).
- VARIANT_BOOL Activate - Set the status of the oscilloscope:
  true - The status of the oscilloscope is on;
  false - The status of the oscilloscope is off.
- VARIANT_BOOL Autoscale - Set autoscale:
  true - Autoscale is enabled;
  false - Autoscale is disabled.
- BSTR FileName - Sets the file name for writing the results to a *.dtu file (after setting this parameter, the file is written to the specified file).
- BSTR GraphColors - Set the color of the graphs (string).
- BSTR Decimation - Set the decimation of the original signal: 1; 10; 100; 1000; 10000 (string).

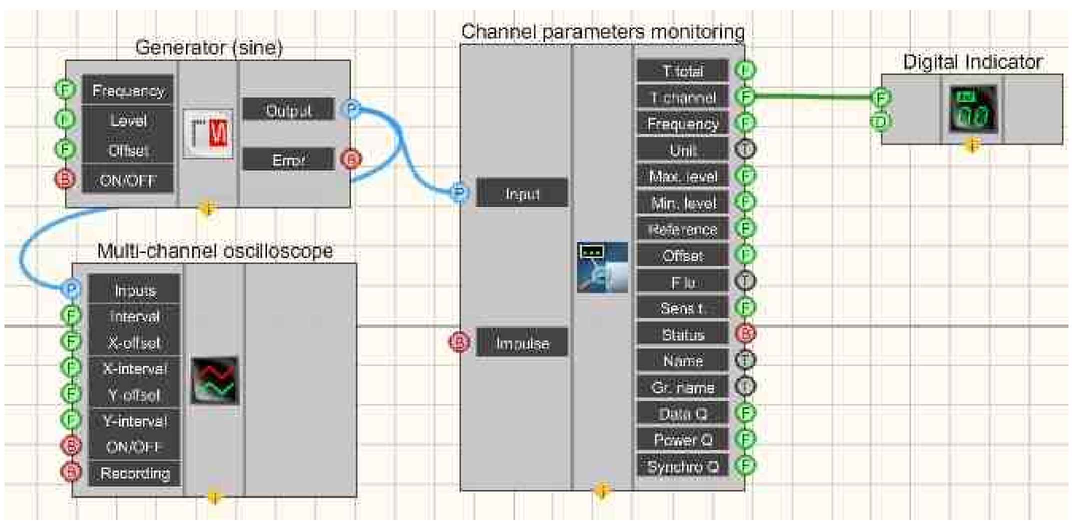
➢**Methods:**

- void SetChannelList (BSTR chanlist) - Set the channel list (channel list).
- void CopyPicture (void) - Copies the graphic to the clipboard.
- void CopyData (void) - Copies data to the clipboard.

Example No. 1

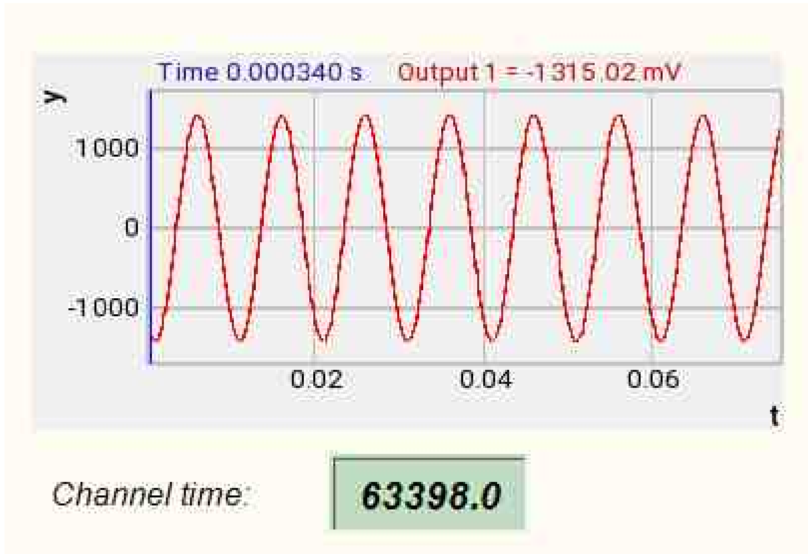
For the Examples to work, it is necessary to have a connected physical ZET device.

Project in SCADA ZETView



In this scheme, a Sine signal 530 is fed from a sine signal to the Channel parameters monitoring 396. The Digital Indicator 682 is used to show the channel time on the display. A Multi-channel oscilloscope 928 is needed to view the signal coming from the generator.

Project operation result

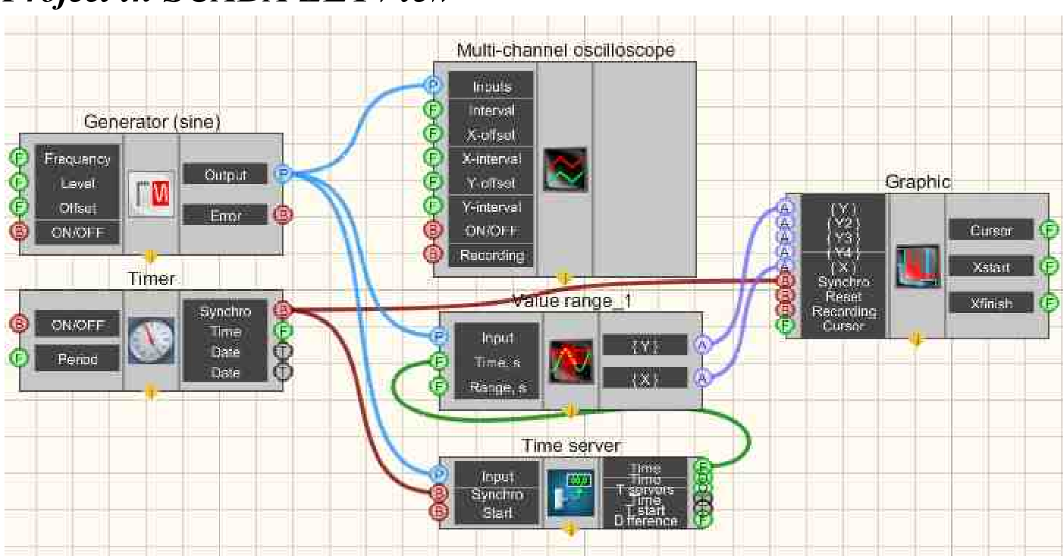


View the example in ZetView

Example No. 2

For the Examples to work, it is necessary to have a connected physical ZET device.

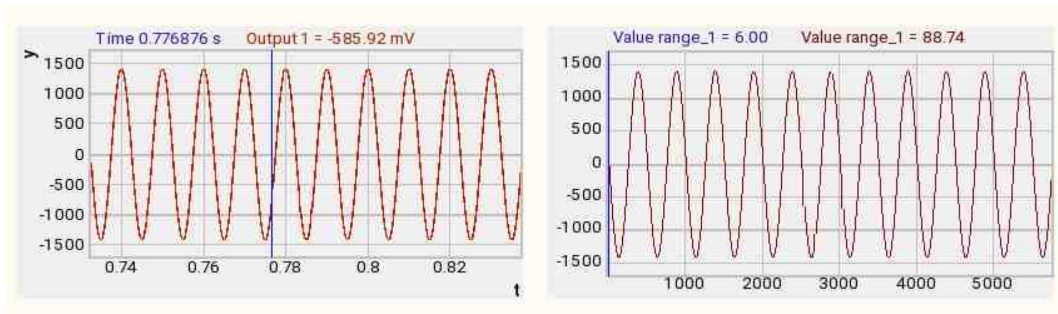
Project in SCADA ZETView



This project shows the operation of the Range of values 928 component is designed to obtain instantaneous values of the channel signal in a given time range. The

Timer 370 is used to supply synchronization pulses to the Server time 366 and Graphic 909. This is necessary so that the Graphic 909 and the Range of values 928 work in a single time space. The Sine signal 928 is needed to generate the test signal, and the Multi-channel oscilloscope 928 is for its current graphical display.

Project operation result

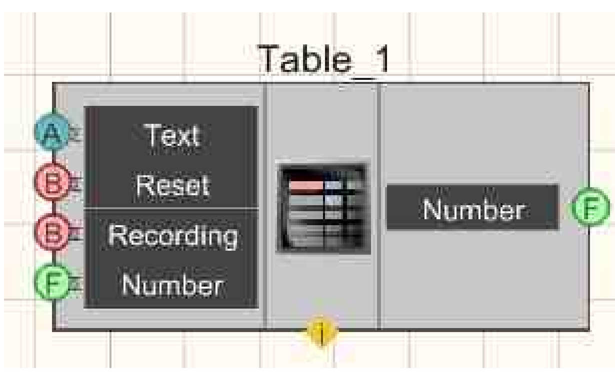


View the example in ZetView

19.10.Data table

The component "**Data table**" is designed to display text values on a specially designed area.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Text - data to display. Each element will be displayed on a separate line.

- Reset - clears the table when receiving an impulse.
- Recording - recording content to a CSV file.
- Number - when getting the number of the selected row.

➢ Output:

Number - displays the number of the selected line.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- CollumnsWidth () - setting the width of the columns.
- *FileName is the file name for writing the content to the file in CSV format.*

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- {Text} - The value of the string to which you want to feed the string array. Each element will be displayed on a separate line.
- {Size} - when getting the number of the selected row.

➢**Custom common properties**:

- Are available by the link below. 154

➢**Custom private properties**:

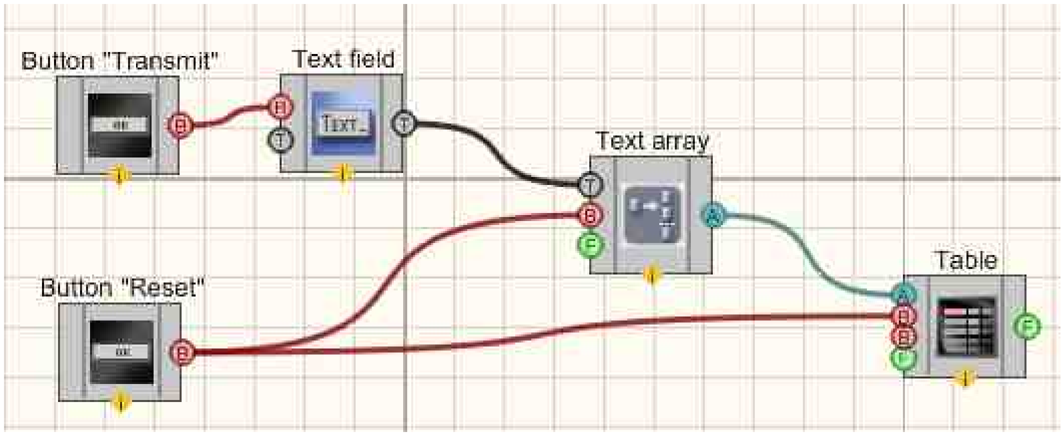
- BSTR CollumnWidth - Set column width (string).
- BSTR FileName - Set the file name for record the results to a *.CSV file (after setting this parameter, the file is recorded to the specified file).

➢**Methods:**

- void Clear(void) - The table is cleared.

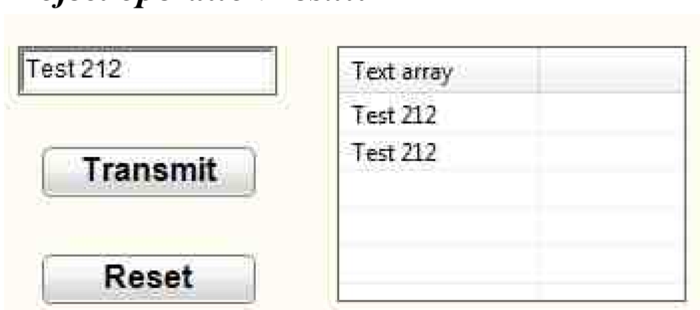
Example

Project in SCADA ZETView



This project shows the operation of the Data table 934 component. Array elements are entered by the user in the Text field 1100 and sent to the Formation of arrays (text - array) 793 when the Non-fixed button 694, the array is displayed in the Data table 934.

Project operation result



View the example in ZetView

19.11.3D - Modeling

The component "**3D - Modeling**" is designed to display sensor data on a specially designed area in a 3D image.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➤Input:

- Save - to save the picture in *. bmp file.
- Data 1 - data for the sensor.

➤ Output:

Doesn't have.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- ModelFileName - the name of the file that stores the 3D model in *.3DS format.
- InfoFileName - file name that stores the settings of the loaded model in *. txt .
- SensorCount (1) - the number of installed sensors.
- SensorVisibility (true) - visibility of sensors when the project is running.
- SensorSize (2.5) - sensor size, set as a percentage of the value equal to the arithmetic mean between the linear dimensions of the model.
- MaxSensorVal (1) - maximum value (corresponds to red).
- MinSensorVal (0) - minimum value (corresponds to blue color).

- SpotSizeProcent (10) - the size of the sensor impact area, set as a percentage of the diagonal (the maximum distance between model points).
- LightCount (1) - the number of installed light sources.
- LightVisibility (true) - Visibility of light sources when the project is running.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- **Input:**
- Save - An input channel with logical values, which is used to save the picture to *. bmp file (0 - no, off, false / 1 - yes, on, true).
- Data 1 - Value that contains data for the sensor (any number).

➢**Custom common properties**:

- Are available by the link below. 154

➢**Custom private properties**:

- BSTR ModelFileName - Set the name of the file that stores the 3D model in *.3DS format (after setting this parameter, the file is read from the specified file).
- BSTR InfoFileName - Set the name of the file that stores the settings of the loaded model in *. txt (string).
- LONG SensorCount - Set the number of installed sensors (number).
- VARIANT_BOOL SensorVisibility - Set the visibility of sensors when the project is running:
  true - Visibility of sensors when the project is running is enabled;
  false - Visibility of sensors when the project is running is disabled.
- FLOAT SensorSize - Setting the sensor size, set as a percentage of the value equal to the arithmetic mean between the linear dimensions of the model (number).
- FLOAT MaxSensorVal - Set the maximum value (corresponds to red) (any value).
- FLOAT MinSensorVal - Set the minimum value (corresponds to blue) (any value).
- FLOAT SpotSizeProcent - Setting the size of the sensor impact area, set as a percentage of the diagonal (the maximum distance between model points) (any value).
- LONG LightCount - Sets the number of installed lights (number).
- VARIANT_BOOL LightVisibility - Set the visibility of light sources when the project is running:

true - Visibility of lights when the project is running is enabled;
false - Visibility of lights when the project is running is turned off.

Глава 20.Programmable logics

In SCADA ZETView projects, the control functions are performed by means of logic components and programmable components.

The section "Programmable logics" contains components used for extending the functions of components and implementation of simple algorithms based on VBScript. In some cases, it allows to considerably reduce the amount of components used in the project and to enable components functions. E.g., activate a button with a script, form a message, change color or component's properties, i.e. to implement functions beyond the scope of those provided for by the system.

"Programmable component**"** component is used designed to expand the capabilities of the components and implement simple algorithms through the use of the programming language VBScript. In some cases, allows to minimize the number used in the project components. Also is used to call methods functions. Click on Example script the button to display, change the color or properties of the component, i.e. to realize functions that are not provided in the system.

"List of variables to the script**"** component is used designed to transfer the list of variables to "Scenario" based on choice HelpString. In some cases, allows to minimize the number used in the project components.

"Script" component is used for implementation of simple algorithms by means of certain commands. The editor has a tabular form with a list of commands representing the sequence of actions. In some cases, it allows to considerably reduce the amount of components used in the project.
"List of variables to the script" allows to create a list of variables to be further used in "Script" component operation depending on the choice of a particular HelpString.

"FOR loop" is used for implementation of loop cycle with a counter. This component allows to set certain condition, which the program will fulfill till its completion. For instance if we have to cycle the program 5 or n times, the "FOR loop" component can be used. FOR loop has a distinctive feature – it is normally identified by "i" or "j" letter. Example: for i:= 1 to n do // set i for 1, 2, 3 …n. After first loop of i variable, set 1, then 2,.. n. "to" stands for increase, "downto" – for decrease.

"WHILE loop" is used to loop the cycle several times. The loop is stopped as fault result is obtained. Since the fault may be found at once, it is possible that a loop will not be completed.

20.1.Programmable component

The component "**Programmable component**" is used to expand the capabilities of the components and implement simple algorithms through the use of the programming language VBScript. In some cases, allows to minimize the number used in the project components. Also is used to call methods functions. Click on Example script the button to display, change the color or properties of the component, i.e. to realize functions that are not provided in the system.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Test - designed to connect to the script components that affect it or data from which will be used in the program.

➢ Output:

- Connection - designed to connect to the script of the components that it will affect.

Comment:

The component does not have to be attached to other components. Basically, the connection shows the relationship between the blocks

Custom common properties:

➢Total properties (environment):

Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Note:
When updating data from channels, the focus is transferred to the main project window.
The transfer of focus to the main window of the project occurs when the buttons are pressed through the script.

The Setup from 07/15/2016 includes a modified ZETView component "Programmable component".
The new version of the component is a Unicode version, which led to the fact that in existing ZETView projects using this component, the project needs to be rebuilt in the new Setup . To do this, in each such project, for each element of the “Programmable component”, perform the following actions:
- by clicking the right mouse button on the component, open the context menu;
- in this menu, select "Edit component", a window for programming will open;
- in this window, click on the "Apply" button;
- save the project;
- if necessary, compile the project to *. exe .

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Test - Designed to connect to the script components that affect it or data from which will be used in the program (any type).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom common properties:**
- BSTR ScriptText - Set ScriptText "Action Script" (any text). This Custom private properties is not visible in the ZETView design interface.

➢**Methods:**

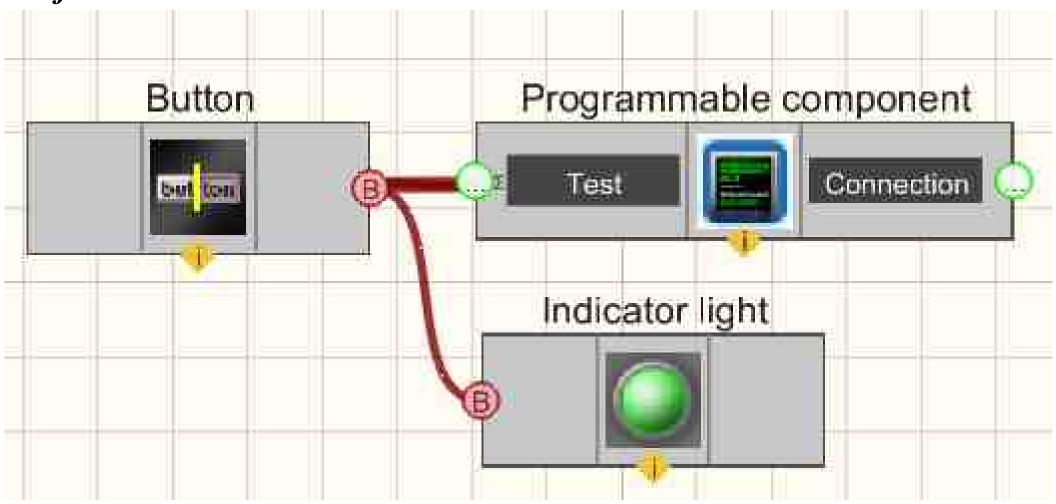
- LONG FillNaibList(LONG naibHWND, BSTR name, IDispatch* naibDisp) - Fills in the list of project components.

- LONG RefreshData(OLE_HANDLE sourceWHND, LONG pinNum, LPCTSTR helpString, LONG parametr, LONG parametrType) - Refreshes data (description is in the Component Example Figure).
- LONG scriptReady(LONG isReady) - Checks if the Script component is ready to work.
- LONG Sleep(LONG time) - Set the time delay, ms (number).
- void UnitSetParamString(LONG param, BSTR str) - Send a SetParam command to the Unit.
- void RefreshNeighbor(BSTR NeighborName, BSTR helpString, DOUBLE parametr, LONG parametrType) - Gives a command to update the neighbor component (Neighbor component name, Help string, Parameter, Parameter type).
- LONG GetHandle(void) - Get the handle of its window (number).
- LONG GetHandleNeighbor(BSTR NeighborName) - Get the window handle of the neighbor component (Component name).
- FLOAT mLtoF(LONG val, LONG accuracy) - Convert data format from LONG to FLOAT
- LONG mFtoL(FLOAT val) - Converts data format from LONG to FLOAT.
- void Send(LPCTSTR helpString, FLOAT parametr, LONG parametrType) - Send a message to the connected component.
- void PutValueToArray(FLOAT value, LONG index) - Add a value to an array, which can then be sent to other components using the SendArray function .
- void defNaibCall(void) - Call the script data arrival function.
- void ShellExecute(BSTR progName, BSTR param) - Run a program by name.
- void ShellKill(BSTR progName) - Stop a program by name.
- void ShowConsole(void) - Displays the results on the console.
- void HideConsole(void) - Hides the console.
- void PrintConsole(BSTR string) - Print the results from the console.
- void SendArray(LPCTSTR helpString, FLOAT* pA, LONG uBound) - Send an array to the component that is connected to the connector. The array must first be filled using the PutValueToArray function.
- void SendDouble(BSTR helpString, DOUBLE parameter, LONG parameterType) - Support for sending data to a Double contact.
- IDispatch* component(BSTR name) - Accessing a component by name. The method allows accessing components by name at runtime
- void DebugOut(BSTR debugString) - Function This.DebugOut - outputs a string through OutputDebugString . Debugging line output.
- void PutLongValueToArray(LONG value, LONG index) - Add a value to an array, which can then be sent to other components using the SendArray function.
- BSTR Translate(BSTR text) - Translates a text string into another language.
- BSTR TranslateDivided(BSTR text) - Divides and translates a text string into another language.
- void SendFloat(BSTR helpString, FLOAT Value) - Send a floating point number from the output contact.

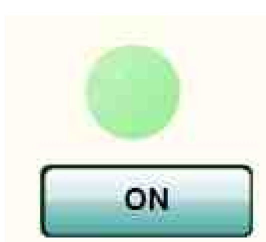
- void SendText(BSTR helpString, BSTR String) - Send a string from the output contact.

Example No. 1

Project in SCADA ZETView



This project shows the operation of the Programmable component 940 is used to initialize the colors in the Non-fixed button 694 and Light indicator 669 components. In this case, the implementation of setting colors for blocks using **a** Programmable component 940 has been simplified. When you click on the Non-fixed button 694, a dialog box appears, which tells you what parameter value was passed to the Light indicator 669 from the Non-fixed button 694 component.



View the example in ZetView

```
'******************* program entry point  ******************
 ' called by a message when the main program starts
 ' CallMessageName - comment to the passed parameter
```

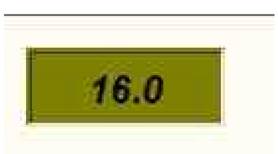
```
Sub Initialization(param)
        button.OnBackColor = &H808000
        button.OffBackColor = &HCD5C5C
        light_indicator.onColor = &H90EE90
        light_indicator.offColor = &H800000

End Sub
'****************************************************************
  'called by the connected component's message
Sub NeighborCall(pType, helpString, param)
        msgbox param


End Sub
Sub NeighborCallArr(arrSize, helpString, ByRef arrVector)
   Imax = UBound(arrVector)   'max index
   If IsArray(arrVector) Then
      ' msgbox arrVector ( Imax ) ' Example accessing an array
element
   Else
      msgbox No array detected
   End If
End Sub

'Script actions when removing a component
'or some other command that terminates the script
Function onClose(param)
   ' the text of your program
End Function
```

Project operation result



View the example in ZetView

Example No. 2

Question.
When working with a programmable component, the following problem arises: its procedures and functions are written in VBScript , then when the script ends, the values of all its global variables are lost. In other words, if the script uses a procedure that needs the data obtained as a result of a previous call to the script, which is in global variables, then it will not receive them. There is no efficient way to save the values of global script variables between iterations. How to solve this problem?
Answer. The Example below shows how to work with global variables.

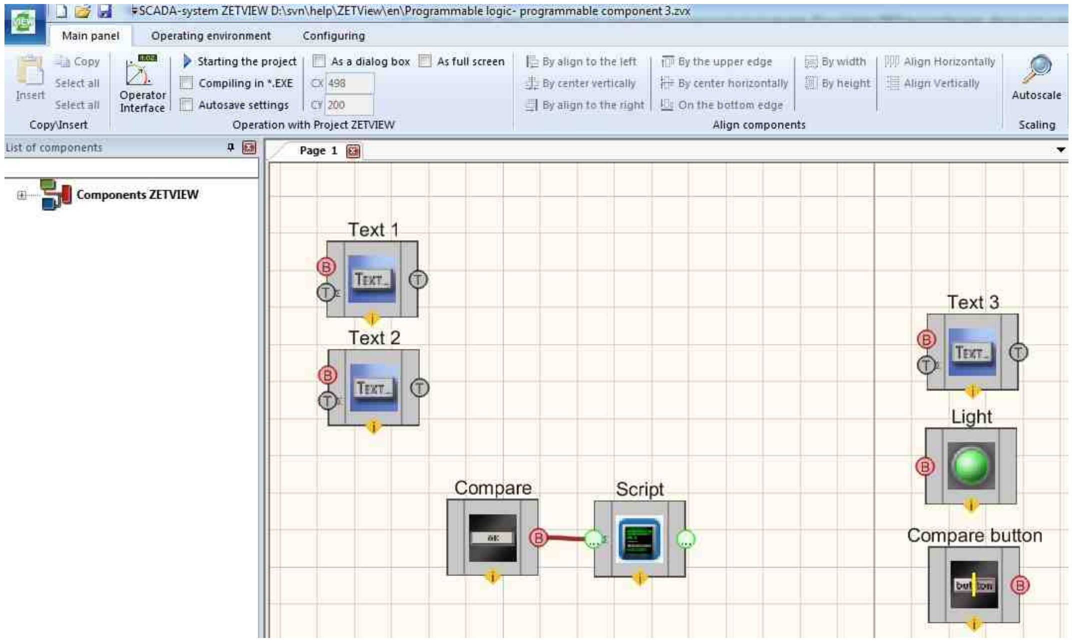
Note.
1) Components in VBScript can be accessed not only by HelpString, but also by Caption , replacing spaces with underscores "_".
2) It is not possible to record your own component for ZETView, which can be added to the List of Virtual Instrument Components list.

View the example in ZetView

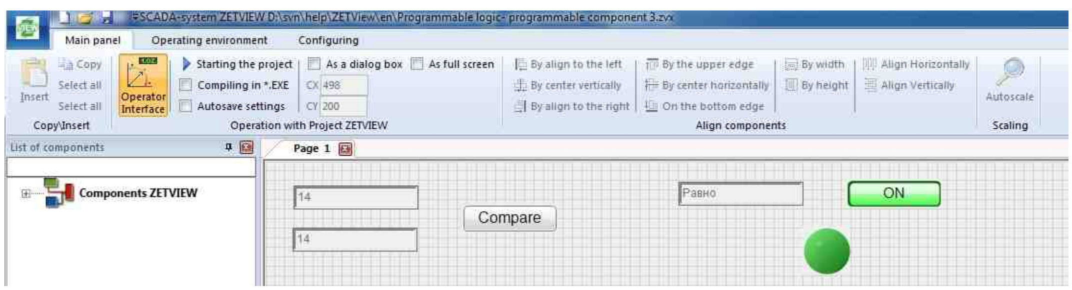
Example No. 3

Question.
I needed to compare two characters entered by the operator in the program, how can I do this?
comparison operation works only with numbers, and I need to compare text.
Answer. The Example below shows how to work and compare text.

project in SCADA ZETView in the Developer Interface

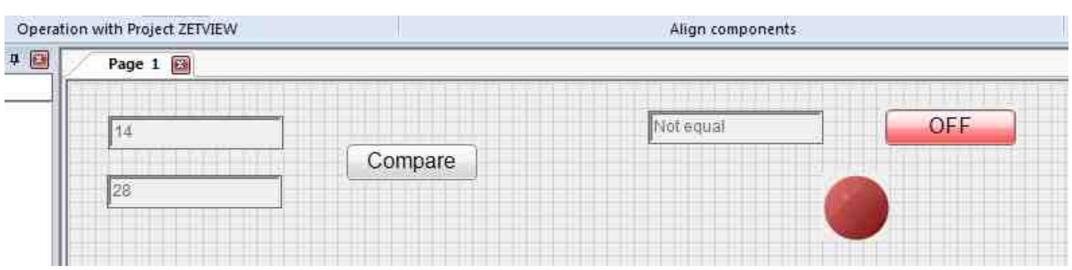


project in SCADA ZETView in Operator Interface

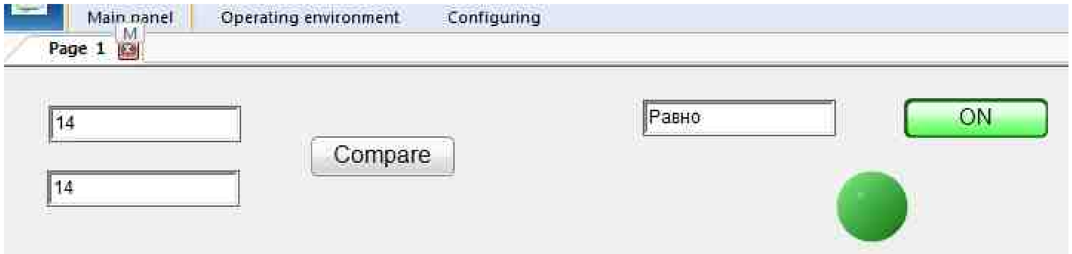


Project operation results

The result of the comparison, the string values are not equal.



The result of the comparison, the string values are equal.

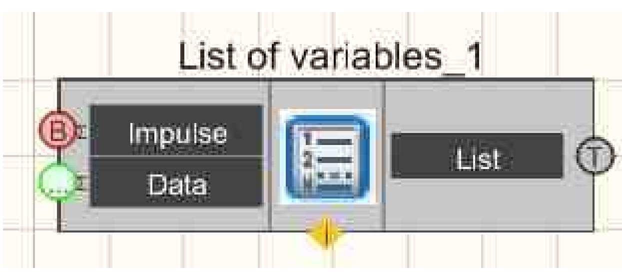


View the example in ZetView

20.2.List of variables to the script

The component "**List of variables to the script**" is used to transfer the list of variables to "Scenario" based on choice HelpString. In some cases, allows to minimize the number used in the project components.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Impulse - when an impulse arrives at this contact, the values of the variables from the table are transferred to the contact list.
- Data - the number of cycles to repeat the script execution.

➤ Output:

- List - from this contact the values of variables are transferred to other components.

Custom common properties:

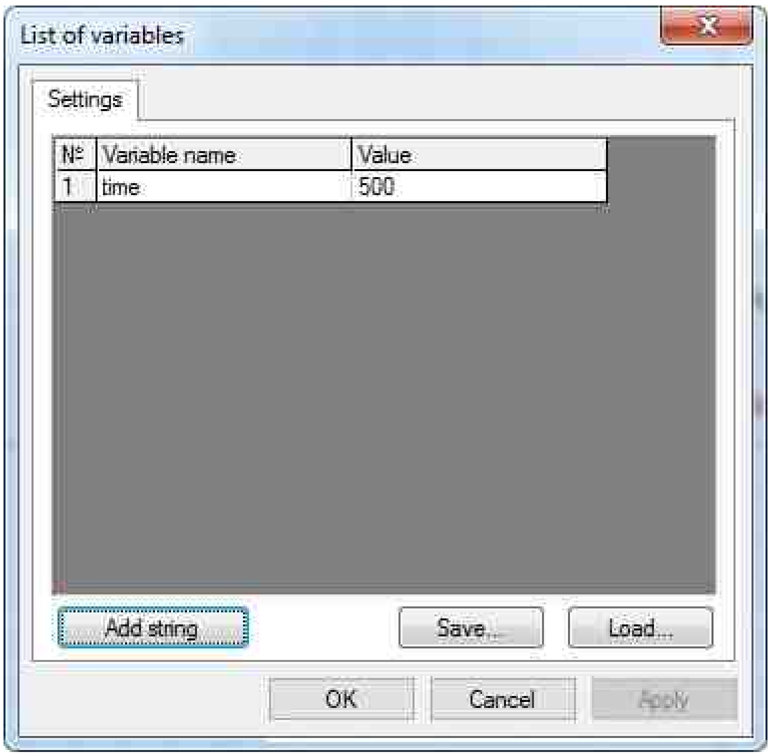
➤Total properties (environment):

- Are available by the link below. 154

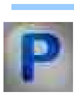
➢**Custom private properties (the default value is shown in parentheses):**

- Doesn't have.

Editor window:
When calling the context menu of a component, select Edit component, after which the variable editor will open:



The "List of variables" to the script window is divided into three columns. The first column is the variable number. Second Variable name - the name of the variable. Third column Value - the value of the variable.



Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Data - Designed to count the number of repetition cycles of script execution (any type).).

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

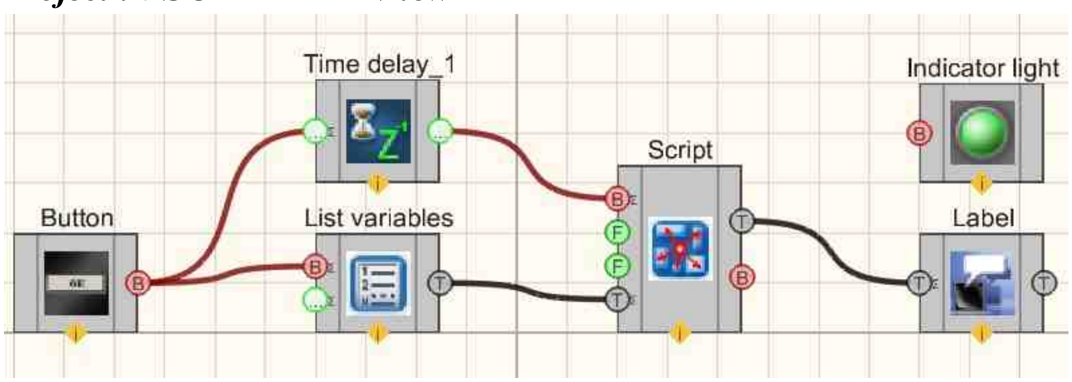
- BSTR VariableList - Set the list of variables (string).

➤**Methods:**

- void SendVariables (void) - Send a list of variables.

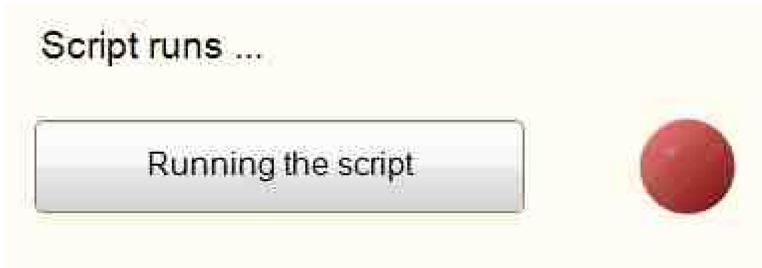
Example

Project in SCADA ZETView



This project shows the operation of the Script 950 component is used to implement the triggering (switching on/off) of the Light indicator 669. The List of variables to the script 947 is needed to set the table of variable values. A Non-fixed button 694 connected to the List of variables to the script 947 sends an impulse to the Script 950, after which the values of the variables are transferred to the Script 950 component. The Label 1087 displays a text message about the status of the Script 950 execution. The remaining component the Non-fixed button 694 used to give an impulse to start the execution of the Script 950.

Project operation result

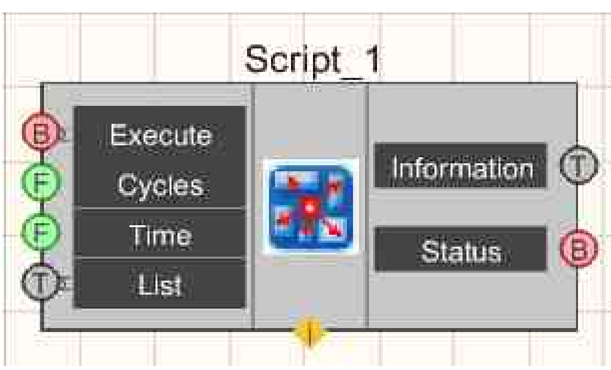


View the example in ZetView

20.3.Script

The component "**Script**" is used for implementation of simple algorithms by means of certain commands. The editor has a tabular form with a list of commands representing the sequence of actions. In some cases, it allows to considerably reduce the amount of components used in the project.
"List of variables to the script" allows to create a list of variables to be further used in "Script" component operation depending on the choice of a particular HelpString.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Execute - script execution starts.

- Cycles - the number of cycles to repeat the script execution.
- Time - setting the numerical value of the pause between two cycles, ms.
- List - connection list of variables for this scenario.

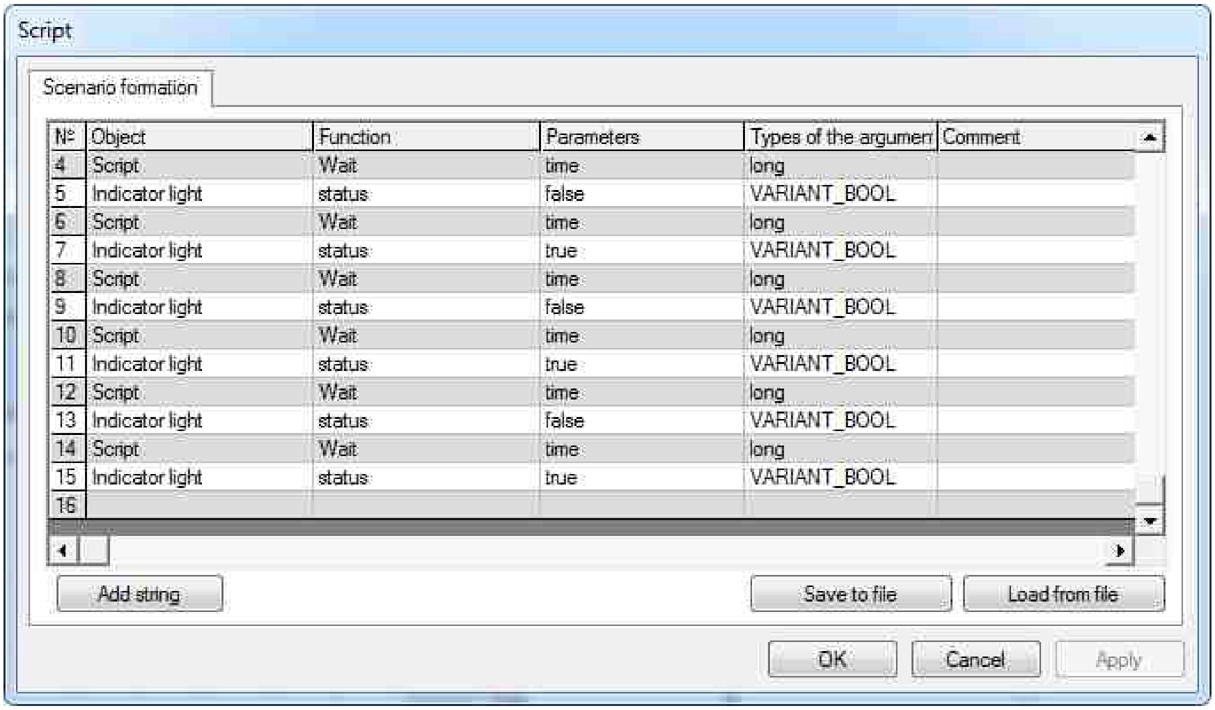
➢ Output:
- Information - the information about the progress of the script.
- Status - end of script execution.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- CycleQuantity (1) - the number of cycles of script execution.
- CycleWaitTime (0) - waiting time between cycles, s.
- HappyEndString (Script completed successfully!) - string, the value of which will be displayed in case of successful script execution.
- BadEndString (Script execution aborted!) - the string that will be displayed if the script is terminated.
- ProcessString (The script is running...) is the string that is output during the execution of the script.
- StopPosition (500) - line number where script execution will be stopped.

Editor window:

When calling the context menu of the component, select Edit component, after which the script editor will open in the form:



№	Object	Function	Parameters	Types of the argumen	Comment
4	Script	Wait	time	long	
5	Indicator light	status	false	VARIANT_BOOL	
6	Script	Wait	time	long	
7	Indicator light	status	true	VARIANT_BOOL	
8	Script	Wait	time	long	
9	Indicator light	status	false	VARIANT_BOOL	
10	Script	Wait	time	long	
11	Indicator light	status	true	VARIANT_BOOL	
12	Script	Wait	time	long	
13	Indicator light	status	false	VARIANT_BOOL	
14	Script	Wait	time	long	
15	Indicator light	status	true	VARIANT_BOOL	
16					

The editor window is divided into four columns. The first column is the number of the command being executed. Second Object - the name of the object with which the work is performed. The third column Parameters is the value of the parameter with which the change takes place. The fourth column Additional information - additional information about the property of the object.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Cycles - The value of the number of cycles the script "program" executes (any number).
- Time - Pause value between two scenario execution cycles, ms (any number).
- List - Designed to connect the list of variables for work (text).

➢**Custom common properties:**

- Are available by the link below. 154

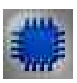
➢Custom private properties (the default value is shown in parentheses):

- BSTR ScriptText - Set the action script (string).
- LONG CycleQuantity - Set the number of execution cycles (number).
- FLOAT CycleWaitTime - Set the waiting time between cycles, s (number).
- BSTR HappyEndString - Set the success string for the script (string).
- BSTR BadEndString - Set the script's break string (string).
- BSTR ProcessString - Sets the script execution string (string).
- LONG StopPosition - Set the stop position (number).

➢Methods:

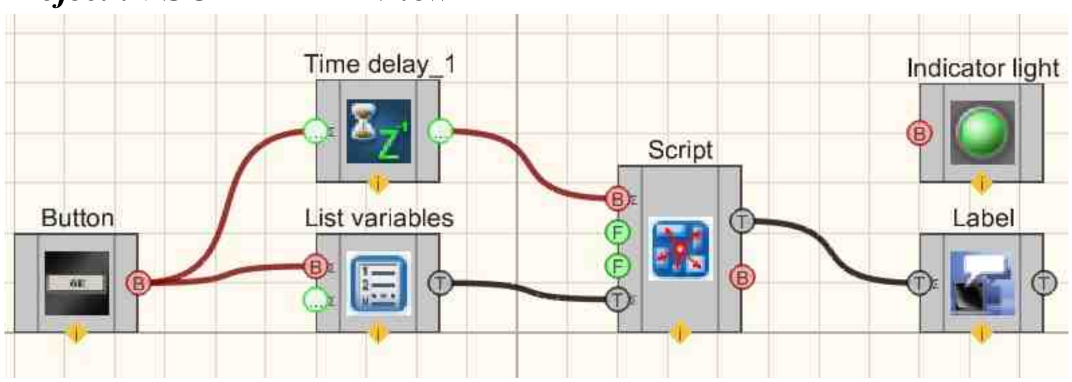
- void AboutBox(void) - Set the padding of additional information.
- LONG FillNeighbourList(LONG naibHWND, LPCTSTR name, IDispatch* naibDisp) - Return the population of the project component list.
- LONG RefreshData(OLE_HANDLE sourceWHND, LONG pinNum, LPCTSTR helpString, LONG parametr, LONG parametrType) - Refreshe data (description is in the Component Example Figure from the Display menu.
- LONG scriptReady(LONG isReady) - Function to check if the Script component is ready to work.
- LONG Sleep(LONG time) - Delay (time in ms).
- void UnitSetParamString(LONG param, BSTR str) - Send the SetParam command to the Unit.
- void RefreshNeighbor(BSTR NeighborName, BSTR helpString, DOUBLE parametr, LONG parametrType) - Send a command to update the neighbor component (Neighbor component name, Help string, Parameter, Parameter type).
- LONG GetHandle(void) - Get the handle of your window.
- LONG GetHandleNeighbor(BSTR NeighborName) - Get the window handle of the neighbor component (Component name).
- FLOAT mLtoF(LONG val, LONG accuracy) - Convert the data format from FLOAT to LONG.
- void PutValueToArray(FLOAT value, LONG index) - Add a value to an array, which can then be sent to other components using the SendArray function.
- LONG mFtoL(FLOAT val) - Convert data format from LONG to FLOAT.
- void Send(LPCTSTR helpString, VARIANT* parametr, LONG parametrType) - Send a message to the connected component.
- void defNaibCall(void) - Call the script data arrival function.
- void ShellExecute(BSTR progName, BSTR param) - Run a program by name.
- void ShellKill(BSTR progName) - Stop a program by name.
- void ShowConsole(void) - Display the results on the console.
- void HideConsole(void) - Hide the console.
- void PrintConsole(BSTR string) - Print the results from the console.

- void SendArray(LPCTSTR helpString, FLOAT* pA, LONG uBound) - Send an array to the component that is connected to the connector. The array must first be filled using the PutValueToArray function
- void SendDouble(BSTR helpString, DOUBLE parameter, LONG parameterType) - Support for sending data to a Double contact.
- IDispatch* component(BSTR name) - Accessing a component by name.
- void DebugOut(BSTR debugString) - Output debug string.
- void PutLongValueToArray(LONG value, LONG index) - Translate a text string to another language.
- BSTR Translate(BSTR text) - Split and translates a text string into another language.
- void SendFloat(BSTR helpString, FLOAT Value) - Send a floating point number from the output contact.
- void SendText(BSTR helpString, BSTR String) - Send a string from an output contact.
- BSTR GetModuleFileName(void) - Set the application name.
- LONG GetSystemMetrics(LONG _lID) - Get the screen dimensions.
- LONG GetWindowsRect(LONG _lNum) - Get the dimensions of the current window.



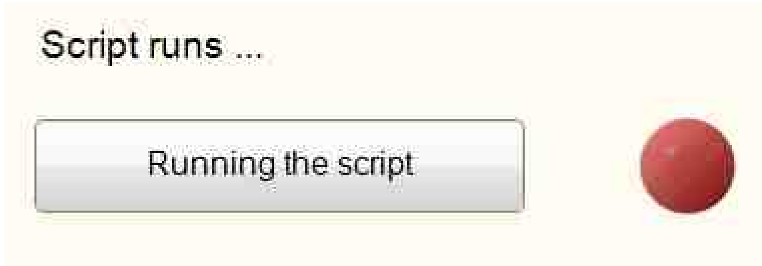
Example

Project in SCADA ZETView



This project shows the operation of the Script [950] component is used to implement the triggering (switching on/off) of the Light indicator [669]. The List of variables to the script [950] is needed to set the table of variable values. A Non-fixed button [694] connected to the List of variables to the script [950] sends an impulse to the Script [950], after which the values of the variables are transferred to the Script [950] component. The Label [1087] displays a text message about the status of the Script [950] execution. The remaining component the Non-fixed button [694] used to give an impulse to start the execution of the Script [950].

Project operation result

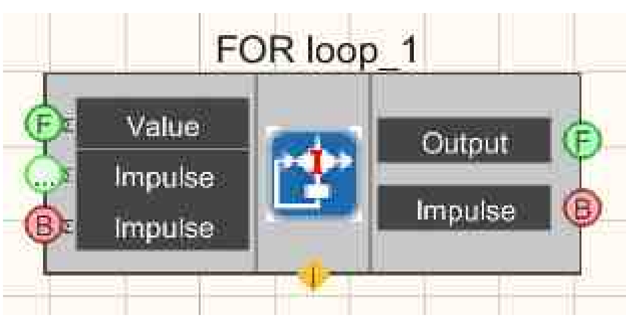


View the example in ZetView

20.4.FOR loop

The component "**FOR loop**" is used for implementation of loop cycle with a counter. This component allows to set certain condition, which the program will fulfill till its completion. For instance if we have to cycle the program 5 or n times, the "FOR loop" component can be used. FOR loop has a distinctive feature – it is normally identified by "i" or "j" letter. Example: for i:= 1 to n do // set i for 1, 2, 3 …n. After first loop of i variable, set 1, then 2,.. n. "to" stands for increase, "downto" – for decrease.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Value - the initial value of the loop variable.

- Impulse - the impulse of the end of the cycle cycle.
- Impulse - cycle interruption impulse.

➢ Output:
- Output - output variable to the loop body.
- Impulse - an impulse about the end of the cycle.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Quantity (1) - set the number of cycles.
- Increment (0) - the value of the output increment.
- Delay (0) - delay in the execution of the next cycle cycle, s.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

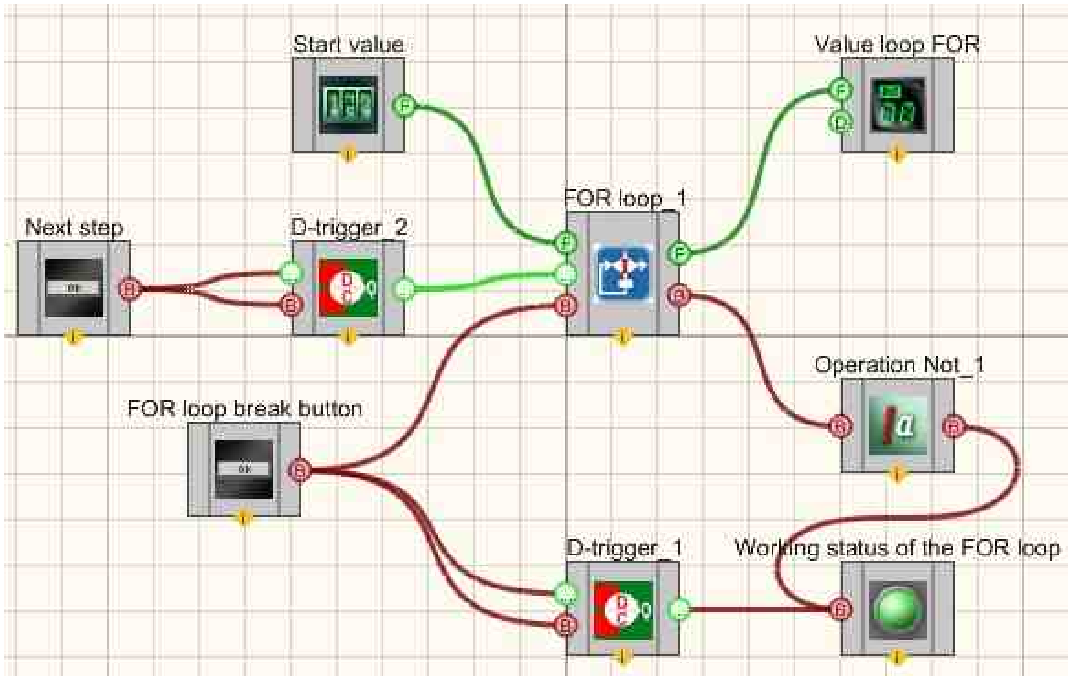
- Value - The initial value of the loop variable (number).
- Impulse - the impulse of the end of the cycle cycle (any type).

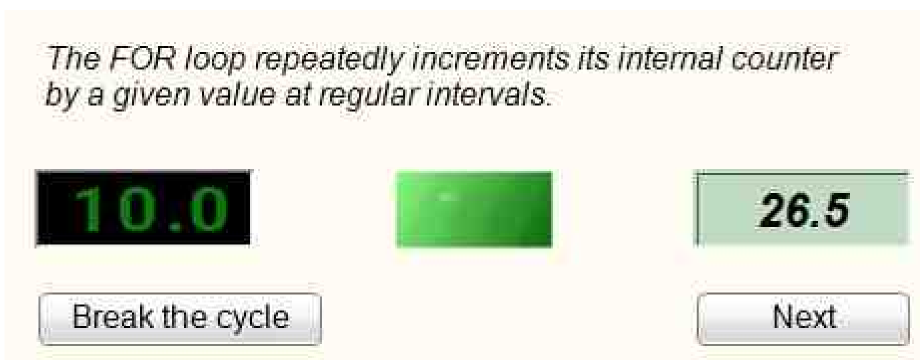
➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- LONG Quantity - Set the number of script execution cycles (number).
- FLOAT Increment - Set the output value increment (number).
- FLOAT Delay - Set the delay for the next cycle cycle, s (number).

Example No. 1

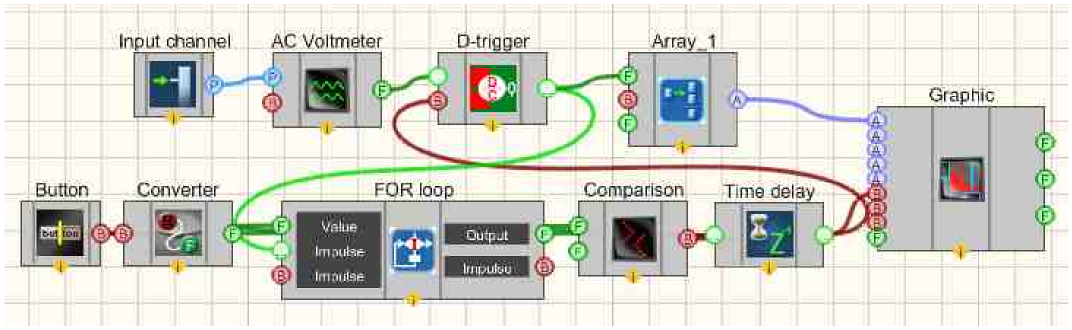




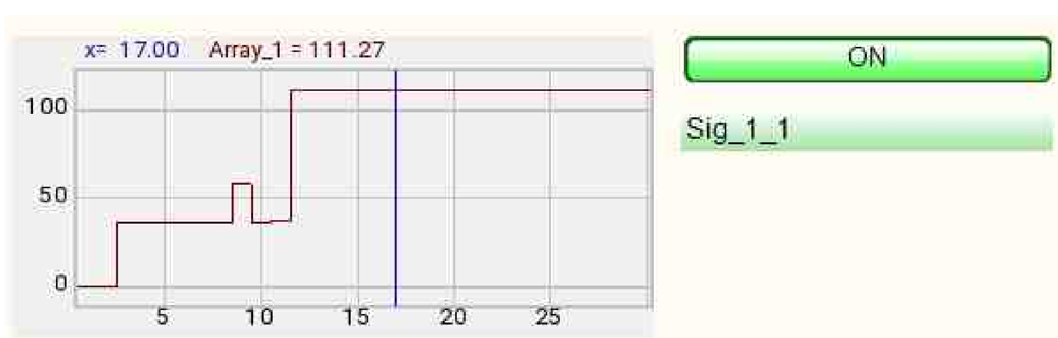
View the example in ZetView

Example No. 2

For the Examples to work, it is necessary to have a connected physical ZET device.



This project shows the operation of the " For loop " component. Data from the generator is sent to the "AC Voltmeter" component. The measured values of the voltmeter are sent to the "Formation of arrays (numeric - array)" component, after which the data is displayed on the graphic.
The values from the voltmeter come at the moment the D-triggers is triggered. The trigger is controlled by the " For loop " component. With each iteration of the loop, a impulse is sent to the trigger sync input. Therefore, the number of trigger firings depends on the number of iterations specified in the settings of the " For loop " component. When considering the circuit, you can select the measurement block (top) and the cycle block (bottom). The whole circuit begins to work when the cycle block is turned on with a Fixed button. The button is connected to the Converter (logic - numeric) component, which converts the logical data type to a numeric one. That is, with the default settings and a high logic level at the input, the converter will output 1. At a low -0. Then the number 1 goes to the "Value" input of the "For loop" component , thereby set the initial value of the loop counter and starting the component. The output value of the for loop is compared to a negative value on the Comparison (inequality) component. This is done in order to simply convert the numerical output value from the "for loop" component into a impulse that will drive the D-triggers clock input. Therefore, a condition is set on the "Comparison" component, which will always be true, respectively, the impulses will arrive at the D-trigger at each iteration of the loop. The Time delay component allows control the timing of the trigger.



View the example in ZetView

20.5.WHILE loop

The component "**WHILE loop**" is used to loop the cycle several times. The loop is stopped as fault result is obtained. Since the fault may be found at once, it is possible that a loop will not be completed.

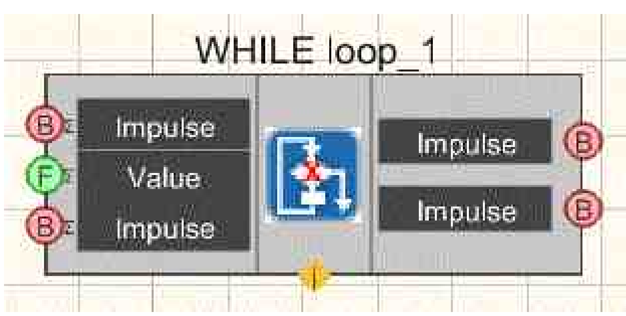
Loop structure with precondition:

```
WHILE  <condition> DO
        begin
                <loop body>
        end;
```

- a logical expression, the truth of which is checked at the beginning of the execution of the cyclic operator;
- any executable statements of the language.

Loop execution order:
While the condition is true, the body of the loop is executed. As soon as the condition becomes false, the loop is terminated.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - the impulse is the beginning of the cycle.

- Value - the output value of the loop variable.
- Impulse - cycle interruption impulse.

➢ Output:

- Impulse - the impulse of the beginning of the next cycle time.
- Impulse - impulse about the end of the cycle.
-

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Value (0) - cycle end threshold.
- Delay (0) - delay in the execution of the next cycle time, s.
- stop (on exceeding the threshold) - the end of the cycle:
- 1. By exceeding the threshold limit.
- 2. By lowering the threshold limit.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Value - The output value of the loop variable (number).

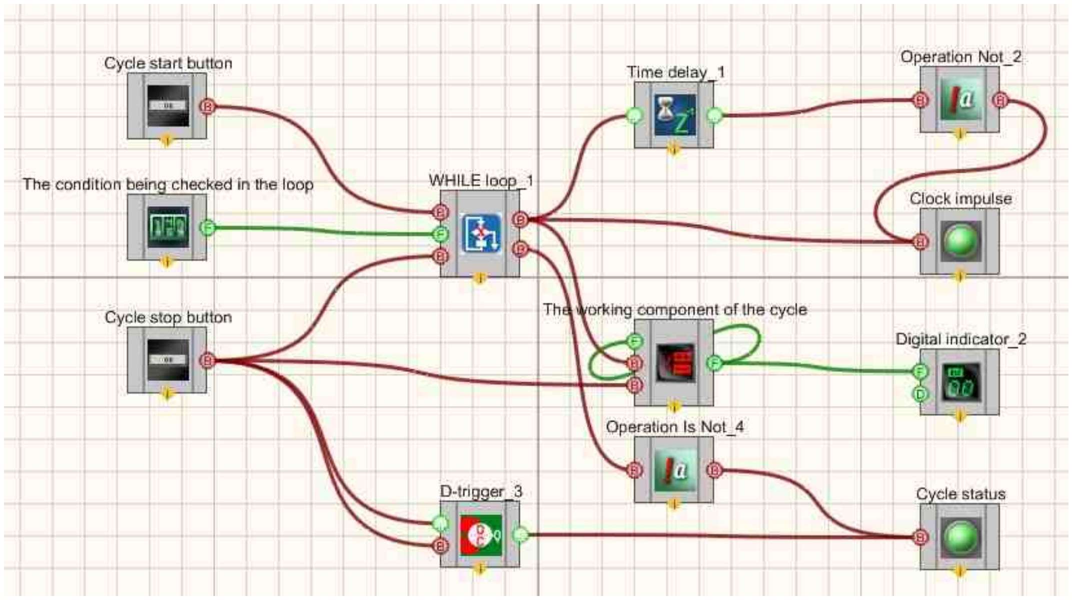
➢**Custom common properties:**

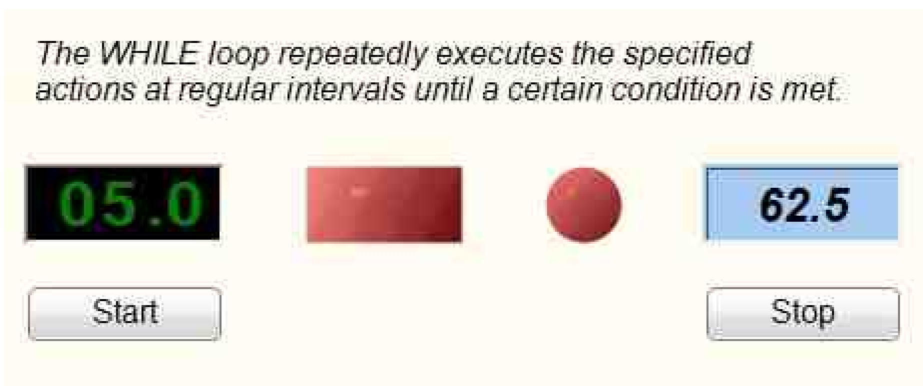
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Value - Set the value of the loop end threshold (number).
- FLOAT Delay - Set the delay for the next cycle time, s (number).
- BSTR Stop - Set the end of the cycle: By exceeding the threshold limit; By lowering the threshold limit (string).

Example No. 1





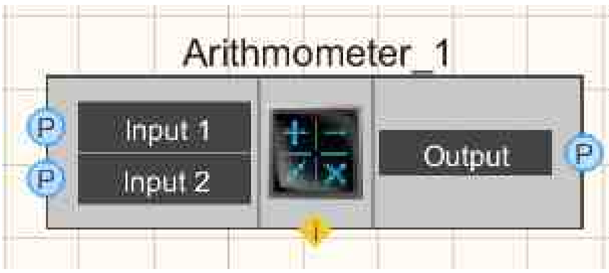
View the example in ZetView

Глава 21.Stream processing

21.1.Arithmometer

The component "**Arithmometer**" is designed to perform mathematical operations with two channels in real time.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - Channel No.1 data stream.
- Input 2 - Channel No.2 data stream.

➢ Output:
- Output - the resulting data stream (constant and real time).

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- operation (Addition) - set the type of arithmetic operation performed with input channels:

1. Addition - set the summing the values of the input channels.
2. Subtraction - set the difference between the values of the first and second channels (the values of channel No. 2 are subtracted from channel No. 1).
3. Multiplication - set the product of channel values.
4. Division - set the quotient of the first and second channels (the values of channel No. 1 are divided by the values of channel No. 2).
5. Maximum - set the output only maximum values at each moment of operation.
6. Minimum - set the display only the minimum values at each moment of operation.
7. Arithmetic mean - set the displays the average value of two signals (the sum of the values of channel No. 1 and channel No. 2 divided by 2).

8. Module - set the output of the modulus of the signal (i.e. the square root of the sum of the squares of the channel input values).

9. Geometric mean - output of the geometric mean value of two signals (square root of the product of channel No. 1 and channel No. 2 values).

- MultConstant (1) - set the constant for multiplication.
- AddConstant (0) - set the constant for addition.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - Measuring channel, which receives the data stream of channel No. 1 (from 0 to (number of channels - 1)).
- Input 2 - Measuring channel, which receives the data stream of channel No. 2 (from 0 to (number of channels - 1)).

➢**Custom common properties:**

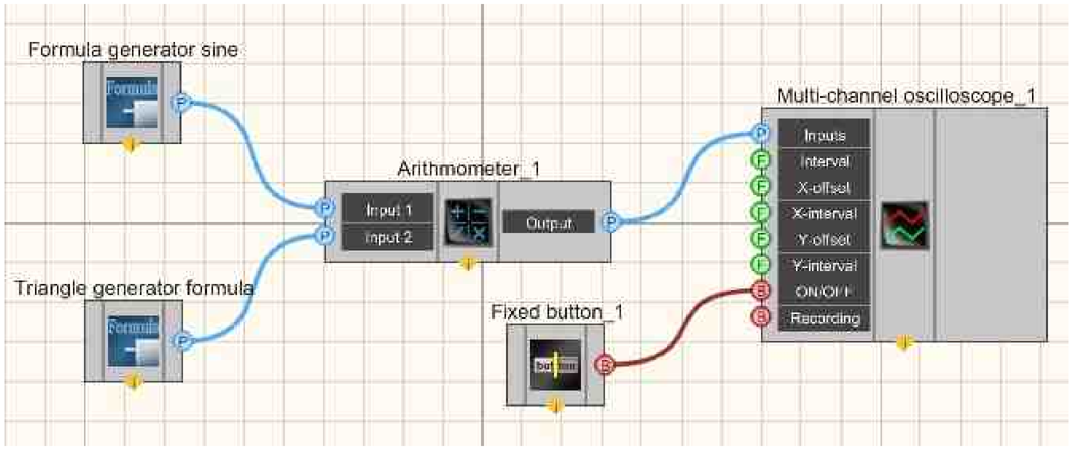
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR Operation - set the type of arithmetic operation performed with input channels: Addition; Subtraction; Multiplication; Division; Maximum; Minimum; Average; Module; Geometric mean (string).
- FLOAT MultConstant - Set the constant for multiplication (number).
- FLOAT AddConstant - Set the constant for addition (number).

Example

Project in SCADA ZETView



This project shows the operation of the Arithmometer 962 component is used to multiply two independent signals into one. A Multi-channel oscilloscope 928 is needed to graphically display the resulting signal. Sine signal 962 and Signal serrated 962 are used a get from two different types of signal.

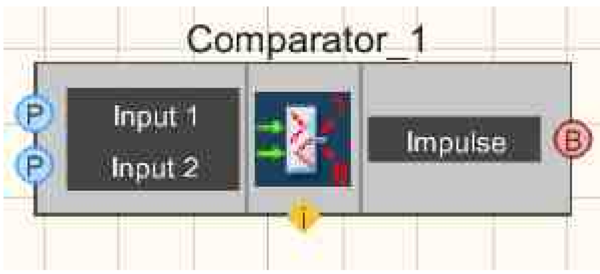
Project operation result

View the example in ZetView

21.2.Comparator

The component "**Comparator**" is designed to compare two input signal values, and depending on whether the level at the first input exceeds the level at the second at the output, a high or low logic level is set at the output.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input 1 - the channel No.1 data stream.
- Input 2 - the channel No.2 data stream.

➢ Output:
- Impulse - a logical unit is set if the signal value in channel No. 1 exceeds the value in channel No. 2, and zero, if vice versa.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input 1 - the measuring channel, which receives the data stream of channel No. 1 (from 0 to (number of channels - 1)).
- Input 2 - the measuring channel, which receives the data stream of channel No. 2 (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below.154

Mathematical description

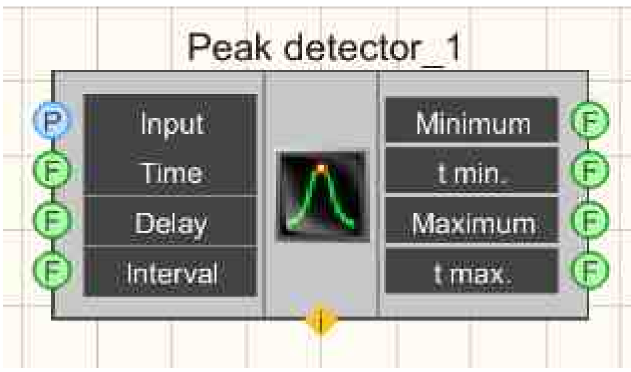
Very often it is necessary to establish which of the two signals is greater, or to determine when the signal reaches a given value. in Example , when generating a triangular waveform, a positive or negative current is passed through the capacitor, the polarity of the current changes at the moment when the amplitude reaches a given peak value. Another example is a digital voltmeter. In order to convert the voltage into a code, an unknown voltage is applied to one of the inputs of the comparator, and a linearly increasing voltage (capacitor + current source) is applied to the other. The digital counter counts the oscillator cycles as long as the ramp voltage is less than the unknown; at the moment of equality of amplitudes, the result obtained on the counter is read. The result is proportional to the input voltage. Such a transformation is called integration with one angle of inclination; in more complex devices, integration with two tilt angles is used.

The simplest comparator i is a high gain differential amplifier based on transistors or operational amplifiers. Depending on the sign of the input voltage difference, the operational amplifier is in positive or negative saturation. The voltage gain is typically in excess of 100,000, so to ensure that the output of the amplifier does not saturate, the voltage at the inputs must be sub-millivolts. For comparators, the term "rate of increase" is usually not used, instead one speaks of a propagation delay relative to the signal given to the input.

21.3.Peak detector

The component "**Peak detector**" is designed to search for the maximum and minimum values of the signal at a given interval.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - channel data stream is coming.
- Time - server time, s.
- Delay - Delay in searching for the peak value.
- Interval - peak value search interval, s.

➢ Output:
- Minimum - at the output we get the minimum peak value.
- t min. - time of the minimum peak value, s.
- Maximum - at the output we get the maximum peak value.
- t max. - time of maximum peak value, s.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Maximum (true) - Specifies the maximum peak value.
- Minimum (true) - Specifies the minimum peak value.
- Delay (0) - indent to the past for the start time of the interval, s (negative number).
- Interval (0.1) - duration of the extremum search interval, s (positive number).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - Measuring channel on which the channel data stream is received (from 0 to (number of channels - 1)).
- Time - Input numeric array for setting the server time value, s (any number).
- Delay - An input numeric array to set the value of the peak value search delay time, s (any number).
- Interval - Input numeric array to set the time value of the peak value search interval, s (any number).

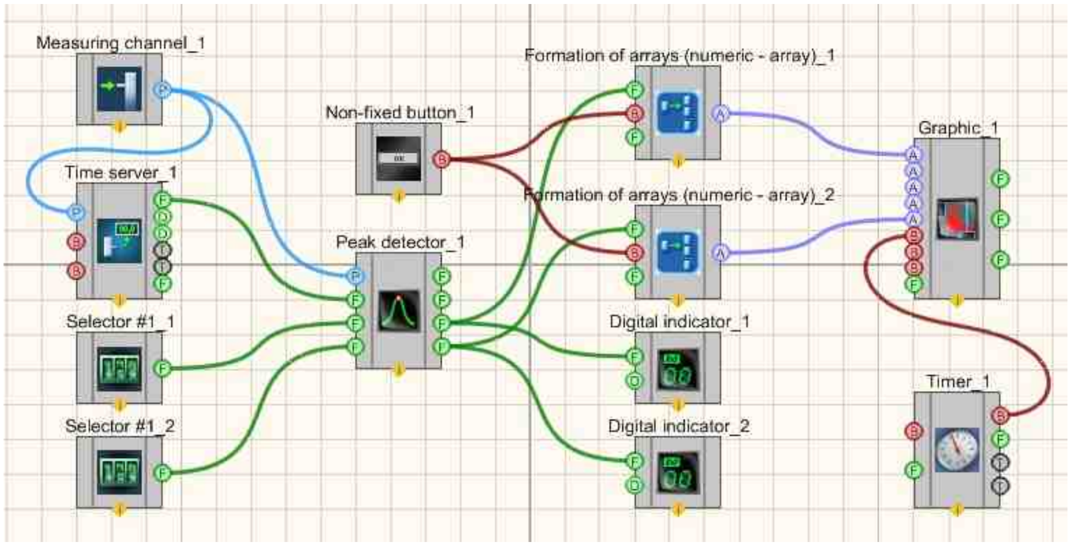
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL Maximum - Set the definition of the maximum peak value:
  true - Maximum peak detection is enabled;
  false - Maximum peak detection is disabled
- VARIANT_BOOL Minimum - Set the definition of the minimum peak value:
  true - Minimum peak detection is enabled;
  false - Minimum peak detection is disabled
- FLOAT Delay - Set the offset to the past for the start time of the interval, s (negative number).
- FLOAT Interval - Set the duration of the extremum search interval, s (positive number).

Example

Project in SCADA ZETView



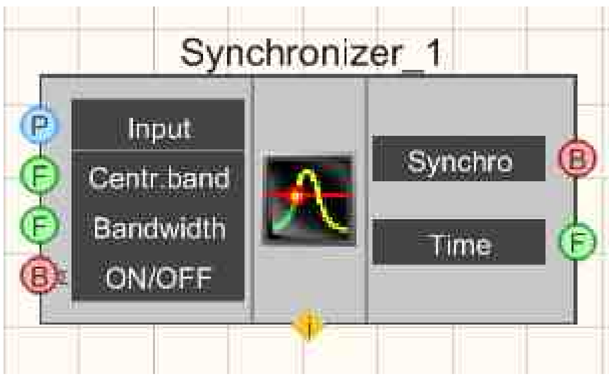
Project operation result

View the example in ZetView

21.4.Synchronizer 1-channel

The component "**Synchronizer 1-channel**" is designed to determine the point in time of one channel when the level of the analyzed signal reaches the specified value.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - the channel data stream.
- Centr.band - the central value of the band.
- Bandwidth - band width.
- ON/OFF - enable/disable the synchronizer channel.

➢ Output:
- Synchro - synchronization impulse.
- Time - the time of the synchronization impulse.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Level (0) - Set the synchronization level (any value).
- LevelWidth (0.05) - Set the width of the sync level (any value).
- fronttype (Upward) - Set the type of synchronization front:
  1. Upward;

2. Downward;
3. Both.

- Active (true) - Set the status of the synchronizer (1 channel).
- Module (false) - Set the data handling modulo.
- Unsensetivity (false) - Set the lack of sensitivity for a certain time after triggering:
- UnsensetivityTime (0) - Set the time of lack of sensitivity after triggering, s (number).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢ **Input:**

- Input - Measuring channel on which the data stream is received (from 0 to (number of channels - 1)).
- Centr.band - An input numeric array for specifying the central value of the band (any number).
- Bandwidth - An input numeric array for specifying the bandwidth (any number).

➢ **Custom common properties:**

- Are available by the link below. 154

➢ **Custom private properties (the default value is shown in parentheses):**

- FLOAT Level - Set the synchronization level (any value).
- FLOAT LevelWidth - Set the width of the sync level (any value).
- BSTR FrontType - Set the type of synchronization front (string):
  1. Upward;
  2. Downward;
  3. Both.
- VARIANT_BOOL Active - Set the status of the synchronizer (1 channel):
  true - The status of the synchronizer is enabled;
  false - The status of the synchronizer is disabled.
- VARIANT_BOOL Module - Set the data handling modulo:
  true - Modulo data processing is enabled;
  false - Modulo data processing is disabled.
- VARIANT_BOOL Unsensetivity - Set the lack of sensitivity for a certain time after triggering:
  true - No sensitivity enabled;

false - No sensitivity disabled.

- FLOAT UnsensetivityTime - Set the time of lack of sensitivity after triggering, s (number).

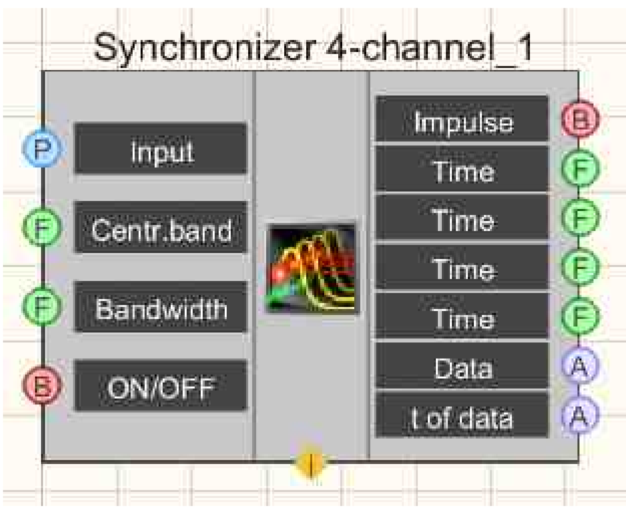
➢**Methods:**

- void SynchroImpulse(void) - Set the sync impulse.

21.5.Synchronizer 4-channel

The component "**Synchronizer 4-channel**" is designed to determine the time of four channels when the level of the analyzed signal reaches the specified value.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - the channel data stream.
- Centr.band - the central value of the band.
- Bandwidth - band width.
- ON/OFF - enable/disable the synchronizer channel.

➢ Output:

- Impulse - we get an impulse at the output (Yes - more, No - less)
- Time - the time of the impulse of the first channel.
- Time - the time of the impulse of the second channel.

- Time - the time of the impulse of the third channel.
- Time - the time of the fourth channel impulse.
- Data - an array of data.
- t data - dimension.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- ABSInputData (false) - take input data module.
- JustFirstValue (false) - only synchronize on the first value.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - Measuring channel on which the data stream is received (from 0 to (number of channels - 1)).
- Centr.band - An input numeric array for specifying the central value of the band (any number).
- Bandwidth - An input numeric array for specifying the bandwidth (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL ABSInputData - Set the input data to be taken module:
  true - Input data to be taken module included;
  false - Input data to be taken module disabled.
- VARIANT_BOOL JustFirstValue - Set to only sync on the first value:
  true - Synchronize only on the first value is enabled;
  false - Synchronize only on the first value is disabled.

➢**Methods:**

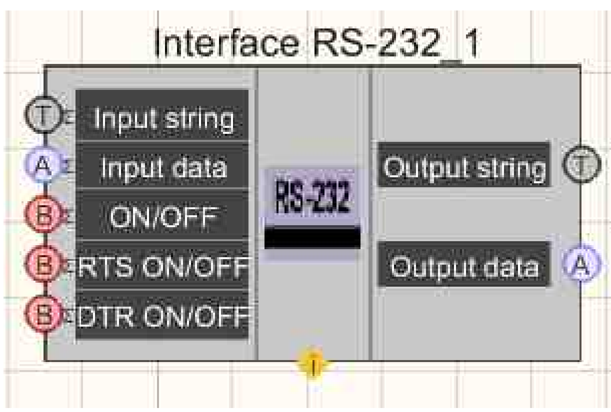
- void SynchroImpulse(void) - Set the sync inpulse.

Глава 22.Exchange protocols and data bus

22.1.Interface RS-232

The component "**Interface RS-232**" is designed for connecting standard external devices (printer, scanner, modem, mouse, etc.) to a computer, as well as for connecting computers to each other.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input string - string of data to be sent.
- Input data - sent data in binary form.
- ON/OFF- start or end of work via RS-232 interface.
- RTS ON/OFF- enable or disable the interface contact **Request for RS-232 transmission.**
- DTR ON/OFF- enabling or disabling interface contact **RS-232 data receiver readiness.**

➢Output:
- Output string - the string of received data.
- Output data - received data in binary form.

Custom common properties:
➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- PortNumber (1) - set the number of the serial (COM) port for working with the power supply.
- bode (9600) - set the baud rate: 075; 110; 150; 300; 600; 1200; 1800; 2400; 4800; 7200; 9600; 14400; 19200; 38400; 115200; 56K; 57600; 128K.
- parity (No) - set the parity: No; Odd; Even; Mark; Space.
- data (8) - set the data bits: 4; 5; 6; 7; 8.
- stopbit (1) - set the number of stop bits: 1; 1.5; 2.
- ReadTimeoutInterval (0) - set the timeout between two symbols.
- ReadTimeoutMultiplier (0) - set the total read operation timeout.
- ReadTimeoutConstant (0) - set a constant for the total timeout of a read operation.
- WriteTimeoutMultiplier (0) - set the total record operation timeout.
- WriteTimeoutConstant (0) - set a constant for the total record operation timeout.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Input string - The value of the string that contains the data to be sent (text).
- Input data - Array values that will send data in binary form.

➢**Custom common properties:**

- Are available by the link below. 154

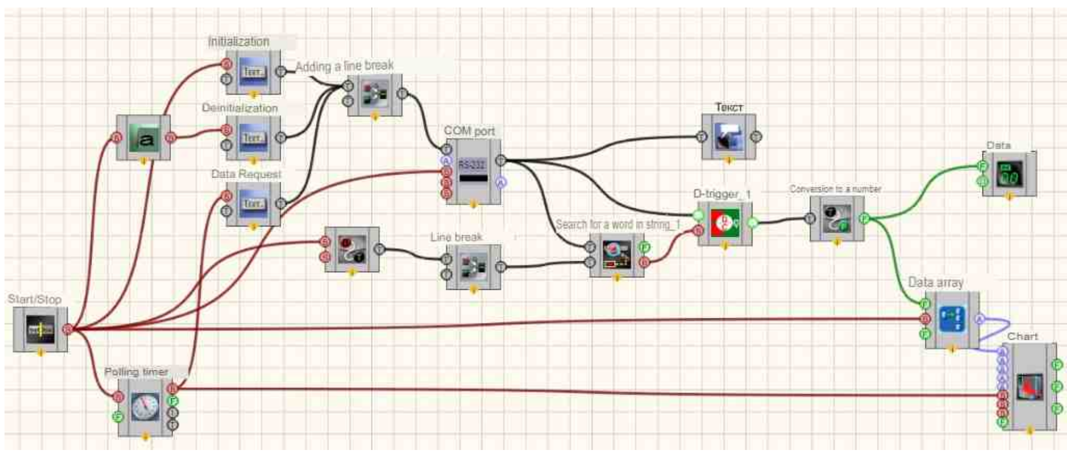
➢**Custom private properties (the default value is shown in parentheses):**

- LONG PortNumber - Set the port number (number).
- BSTR Bode - Set the baud rate: 075;110;150;300;600;1200;1800;2400;4800;7200;9600;14400;19200;38400;115200;56K;57600;128K (string).
- BSTR Parity - Set the parity: No; Odd; Even; Mark; Space (string).
- BSTR Data - Set the data bits: : 4; 5; 6; 7; 8 (string).
- BSTR StopBit - Set the number of stop bits: 1; 1.5; 2 (string).
- LONG ReadTimeoutInterval - Set the timeout between two characters (number).
- LONG ReadTimeoutMultiplier - Set the total timeout for a read operation (number).

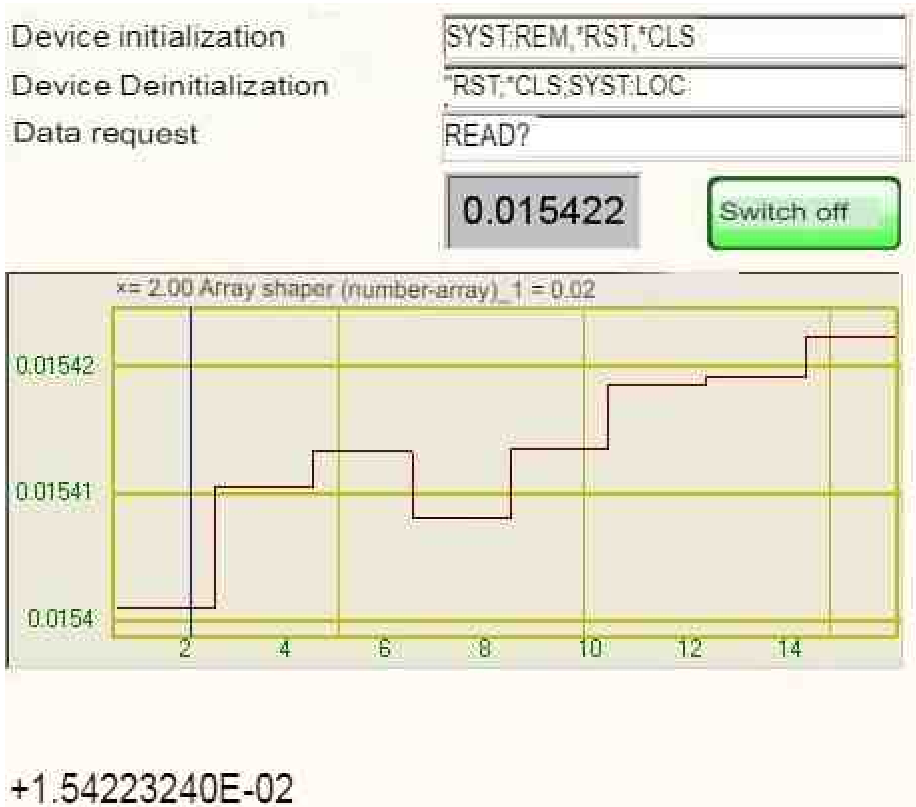
- LONG ReadTimeoutConstant - Set a constant for the total read operation timeout (number).
- BSTR Data - Set the total record operation timeout (number).
- BSTR StopBit - Set a constant for the total record operation timeout (number).

Using Example 1 requires a connected ZET device containing a DAC and an Agilent 34401a multimeter set to PortNumber=3.

Example No. 1
Project in SCADA ZETView



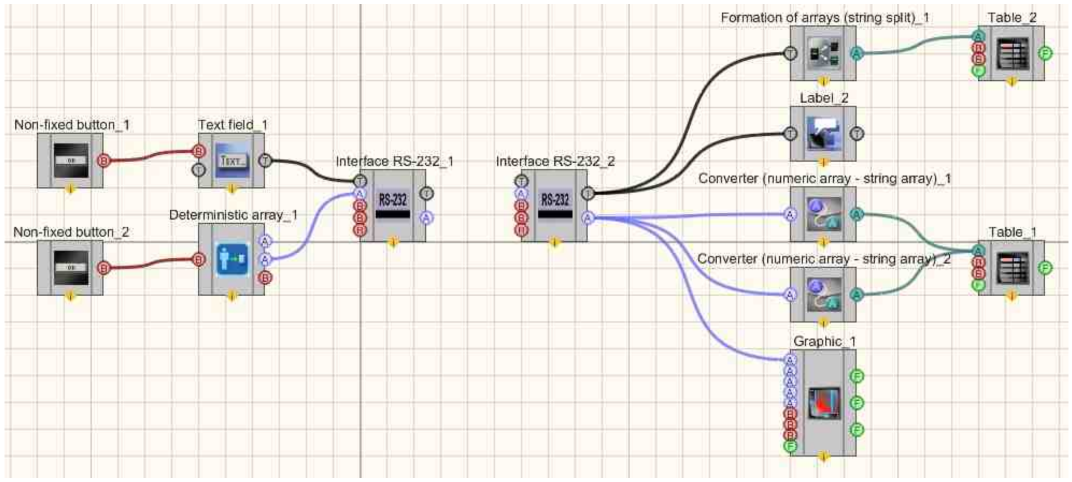
Project operation results



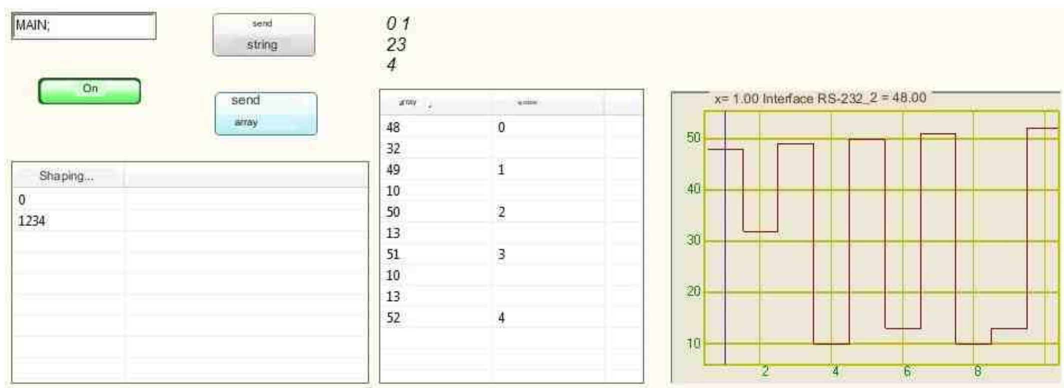
View the example in ZetView

Using Example 2 requires two com ports to be connected to each other.

Example No. 2
Project in SCADA ZETView



Project operation results



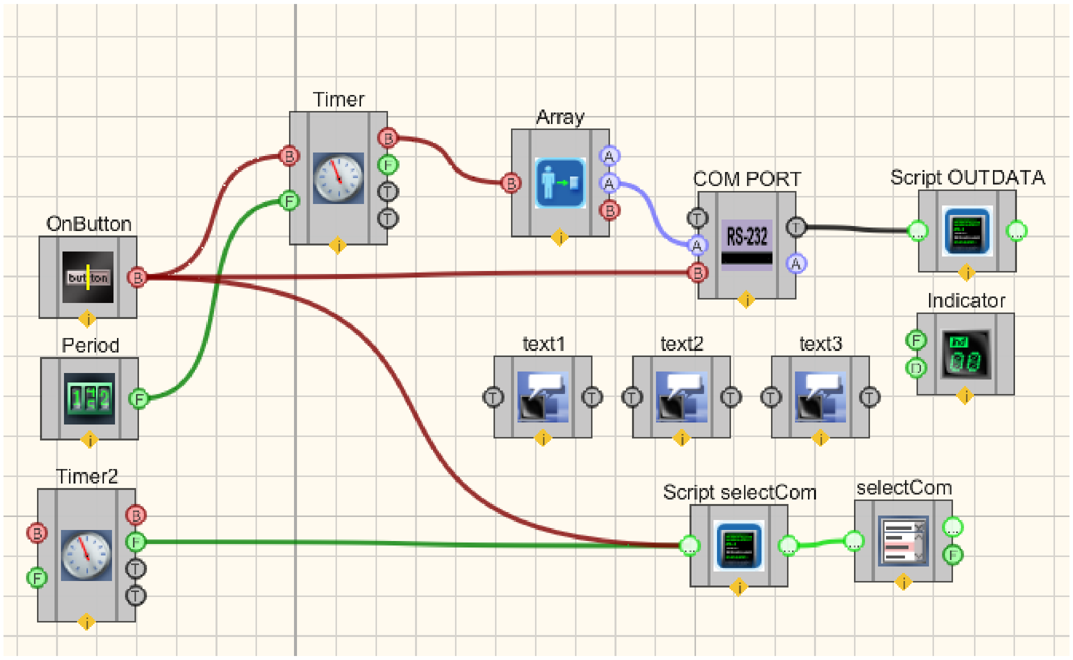
View the example in ZetView

Example No. 3

Connecting third-party RS-232 sensors via UAI protocol

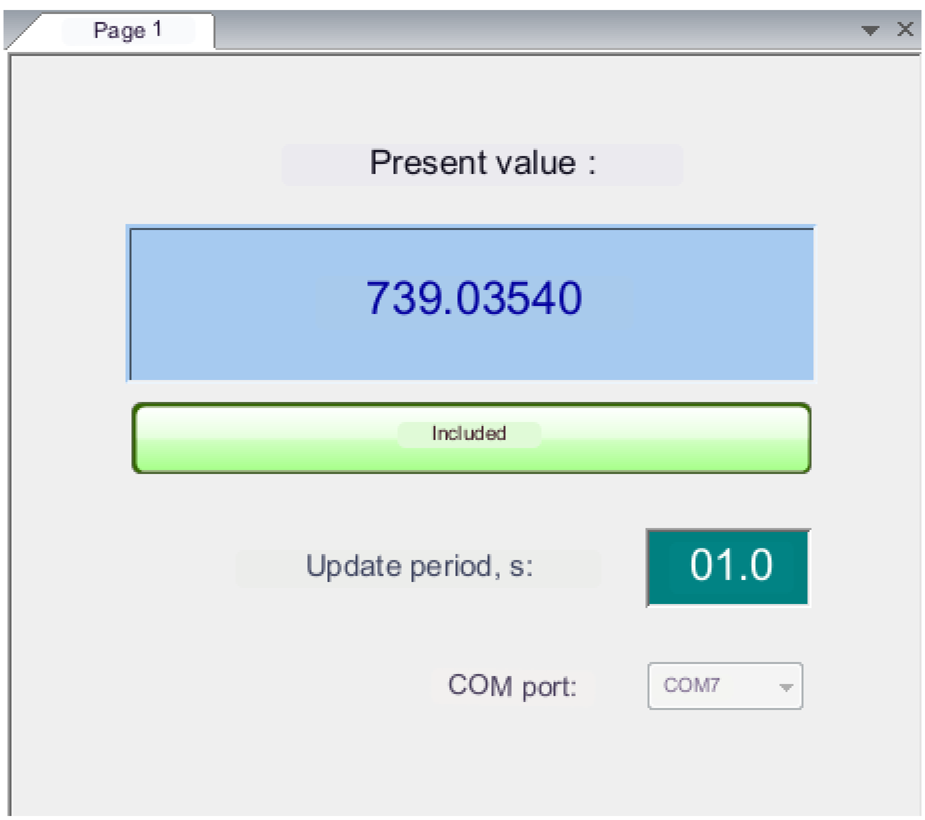
To use the PDE-020(I) reference pressure transducers manufactured by ELEMER Research and Production Enterprise LLC in the designed automated control and measuring systems, manufactured by ZETLAB, a specialized program was created to display the measured data on the operator's monitor.

The SCADA project for communication PDE-020(I) is shown in the figure.



SCADA project for communication PDE-020(I) with ZETLAB software

Project operation results



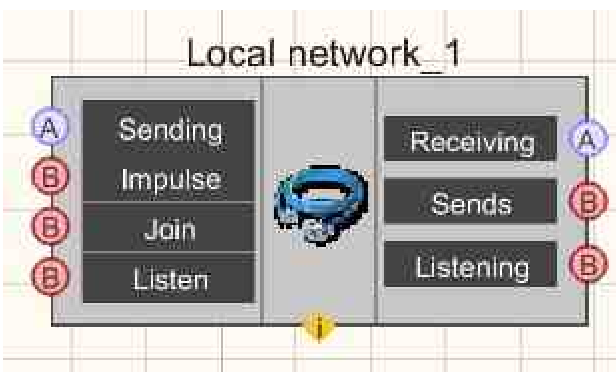
Operator program interface

Since the PDE-020(I) reference pressure transducers use the closed UAI(L) exchange protocol, difficulties arise when integrating the recorded data into third-party systems. This SCADA project allows you to build a convenient interface for displaying information and organize data transfer to third-party systems via OPC without much effort and programming skills

22.2.Local network

The component "**Local network**" is designed to exchange information over an Ethernet network using the TCP/IP protocol.
Used for communication between components located on different computers.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Sending - set the input array of bytes to be sent.
- Impulse - set a signal to send a packet.
- Connect - set the signal to connect/disconnect from a remote computer
- Listen - set the signal to enable/disable listening.

➢Output:

- Receiving - output array of received bytes.

- Sends - status of the port that sends packets.
- Listening - status of the port receiving packets.

Custom common properties:

➢Total properties (environment):

- Are available by the link below 154

➢Custom private properties (the default value is shown in parentheses):

- IPAddress () - Address of the receiving computer.
- Port (0) - The number of the receiving port on the receiving computer.
- RecievePort (0) - Listening port number.
- Sender (False) - Connect to a remote computer to send data.
- Reciever(False) - Listen on the receiving port.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Sending - Array values that will send bytes in binary form.

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR IPAddress - Set the address of the receiving computer (string).
- LONG Port - Set the receiving port number on the receiving computer. (number).
- LONG RecievePort (0) - Set the listening port number. (number).
- VARIANT_BOOL Sender (False) - Set to connect to a remote computer to send data:
- true - Connect to a remote computer to send data;
  false - Do not connect to a remote computer to send data.
- VARIANT_BOOL Reciever(False) - Set to listen on the receive data port:
- true - Listen on the data receiving port;
  false - Don't listen on the receiving port.

Глава 23.Different

23.1.Select color

The component "**Select color**" is designed for selecting and displaying the color code for creating access to the private properties of components, duplicating all these parameters as a separate module on the workspace in developer mode.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - a color palette appears in which the user can select the color that will be transmitted from the output contact.

➢ Output:

- Color - a number that defines the color in RGB format is transmitted.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

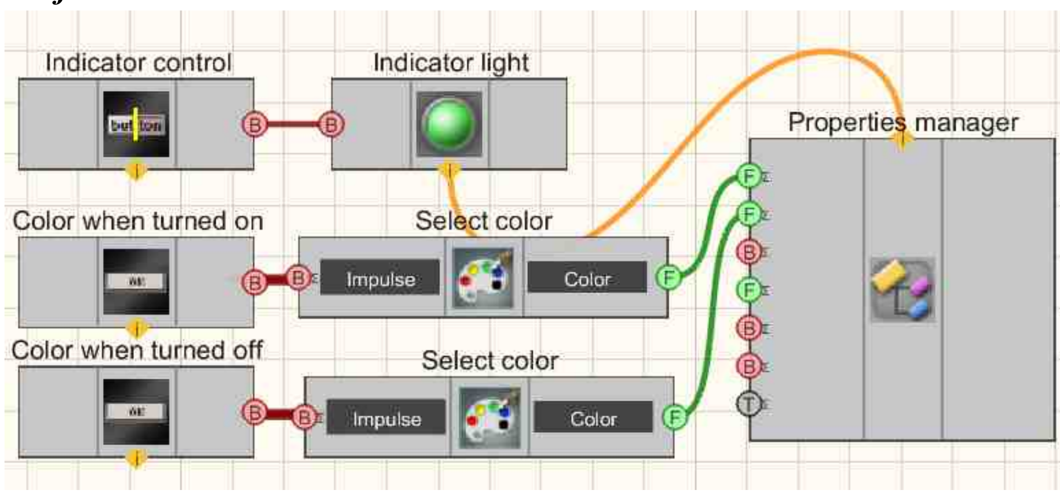
component, component properties values range as well as the range of component's methods parameters values.

➤**Custom common properties:**

- [Are available by the link below.] 154

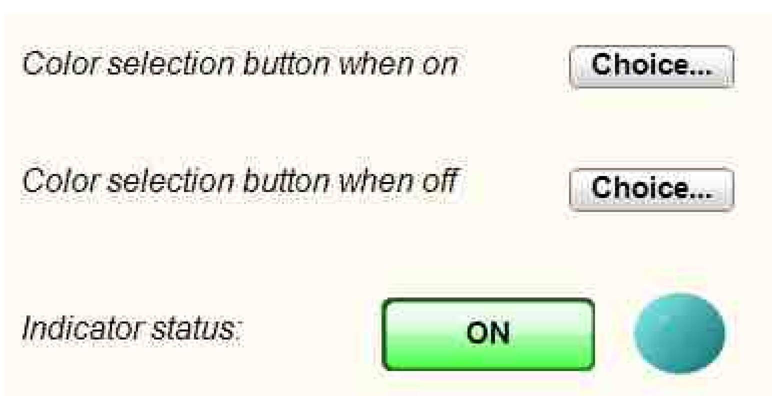
Example No. 1

Project in SCADA ZETView:



This project shows the operation of the [Property Manager] 999 component exposes private properties to other modules. The [Light indicator] 669 of the component is used to set the color of the indicator in different modes. A [Non-fixed buttons] 694 are needed to call the color palette. A [Fixed button] 691 is used to switch the indicator status. [Select colors] 982 is used to graphically display the change in color.

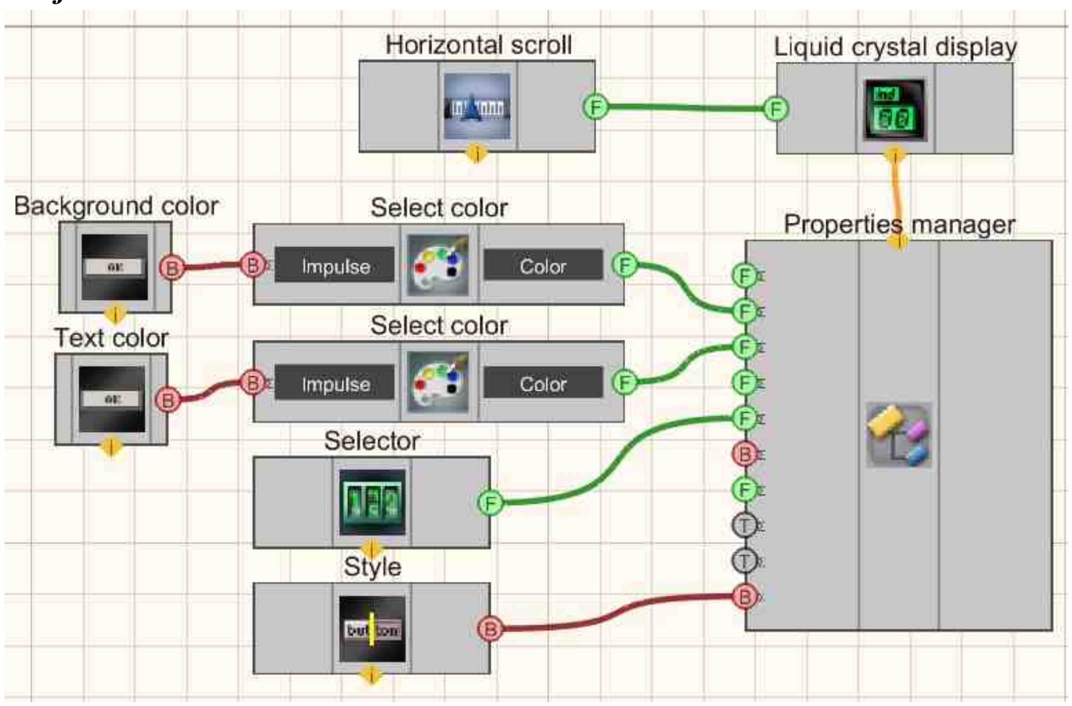
Project operation result



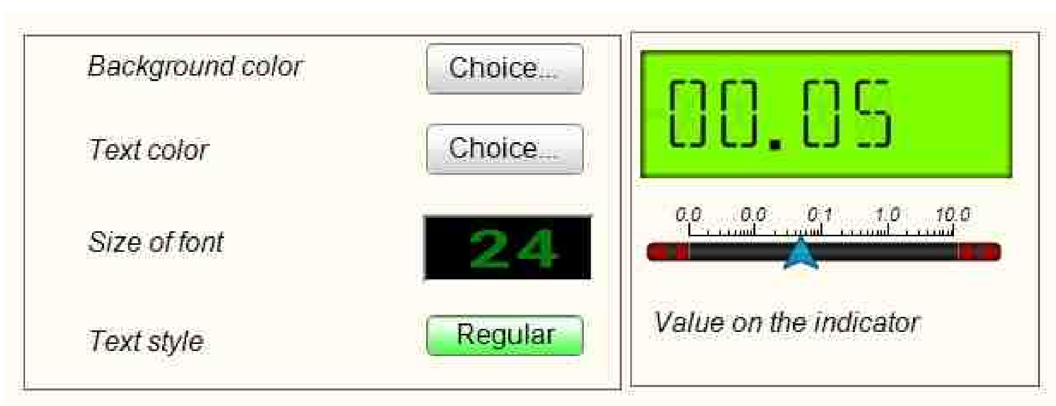
View the example in ZetView

Example No. 2

Project in SCADA ZETView:



Project operation result



View the example in ZetView

23.2.Exit

The component "**Exit**" is designed to exit the ZETView program upon receipt of a control pulse to the input, or when a button is pressed in the operator mode.
If changes have been made in the project, then before exiting ZETView will ask whether to save the changes, and will exit after the user selects an answer.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Impulse - control impulse to exit the program.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- TextSize (12) - the text size of the label on the button in operator mode.
- textstyle (FontStyleBold) - the text style of the inscription on the button in operator mode.
- Enabled (true) - availability of the component.
- ButtonCaption (Exit) - the caption text on the button. "Exit" is the default.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

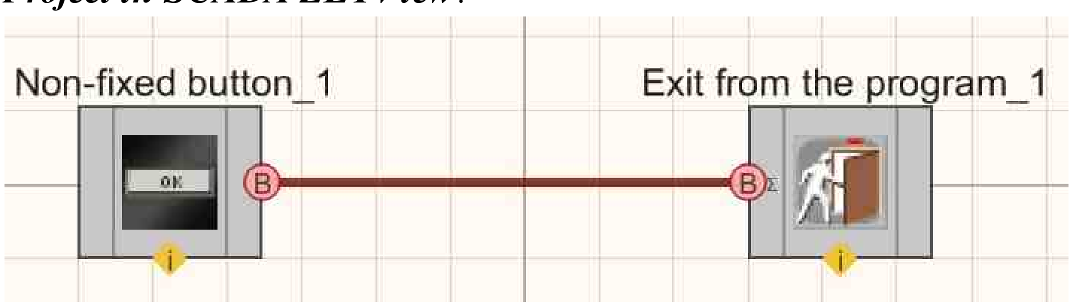
Setting:

➢**Custom common properties:**
- Are available by the link below. 154

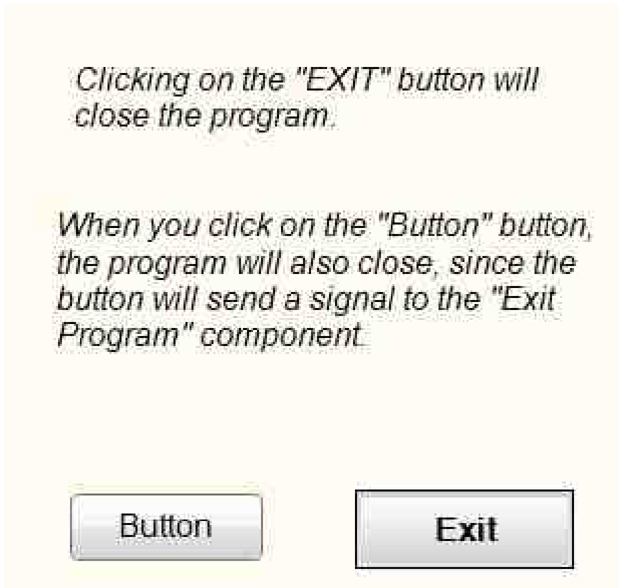
➢**Custom private properties (the default value is shown in parentheses):**
- SHORT TextSize - Set the text size (number).
- BSTR TextStyle - Set text style: FontStyleRegular ; FontStyleBold; FontStyleItalic; FontStyleBoldItalic; FontStyleUnderline; FontStyleStrikeout(string).
- VARIANT_BOOL Enable - Set the availability of exiting the program:
  true - the availability status is on.
  false - the availability status is off.
- BSTR ButtonCaption - Set button caption (string).

Example

Project in SCADA ZETView:



Project operation result

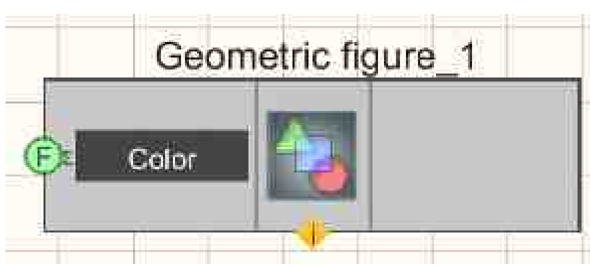
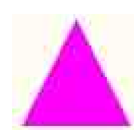


View the example in ZetView

23.3.Geometric figure

The component "**Geometric figure**" is designed to select the shape: rectangle (circle, triangle), colors and controlled coordinates (X,Y).

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Color - the color of the geometric figure is assigned.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Color (000000) - select the color of the geometric figure.
- Figuretype (Circle) - type of geometric figure:
  1. Circle.
  2. Rectangle.
  3. Triangle.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Color - Adjustable color value of the geometric shape (from 0 to 255).

➢**Custom common properties:**
- Are available by the link below. 154

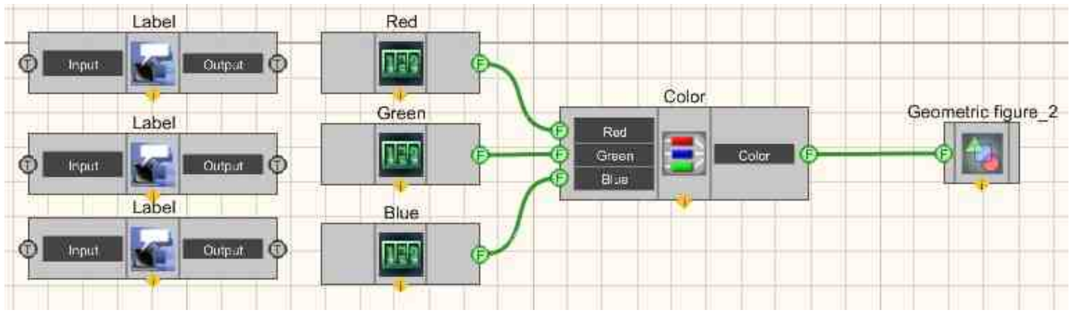
➢**Custom private properties (the default value is shown in parentheses):**

•LONG Color - Set the color of the geometric shape (from 0 to 255).

•BSTR FigureType - Set the type of geometric figure: Circle; Rectangle; Triangle (string).

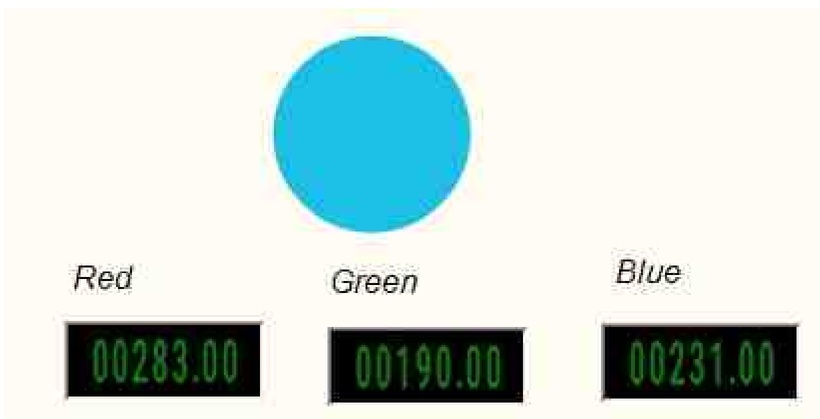
Example No. 1

Project in SCADA ZETView



This project shows the operation of the Geometric figure 987 component is intended for color changes whose values come from Selector No.1 428, Selector No.2 428, Selector No.3 428.

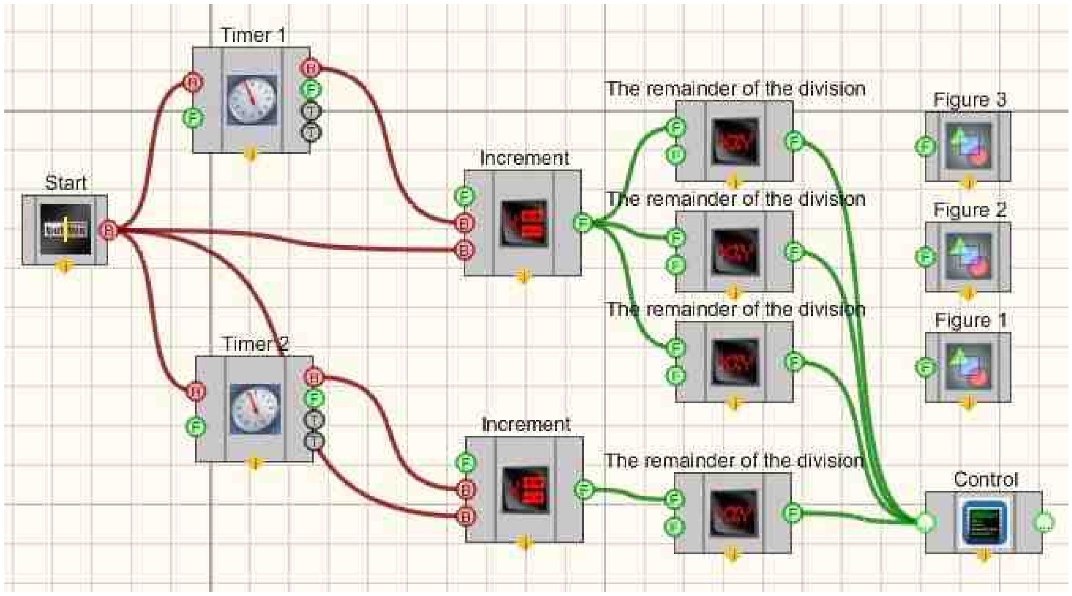
Project operation result



View the example in ZetView

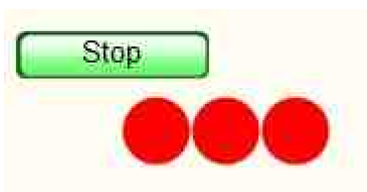
Example No. 2

Project in SCADA ZETView



This project shows the operation of the Geometric figure [987] component is intended for controlled (X,Y) coordinates.

Project operation result



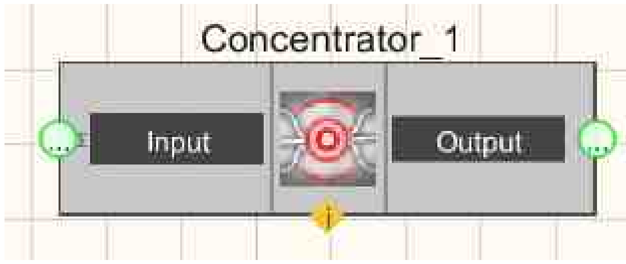
View the example in ZetView

23.4.Concentrator

The component "**Concentrator**" is intended solely for the convenience of arranging components in the designer mode. Allows you to swap the input and output, as well as transfer them to one side. Does not perform any action on the received data.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have.

Setting:

➢Input:
- Input - input contact.

➢ Output:
- Output - output contact.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- input (left) - input contact position (left/right).
- output (right) - position of the output contact (left/right).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Input - Input contact value (any number).

➢Custom common properties:
- Are available by the link below. 154

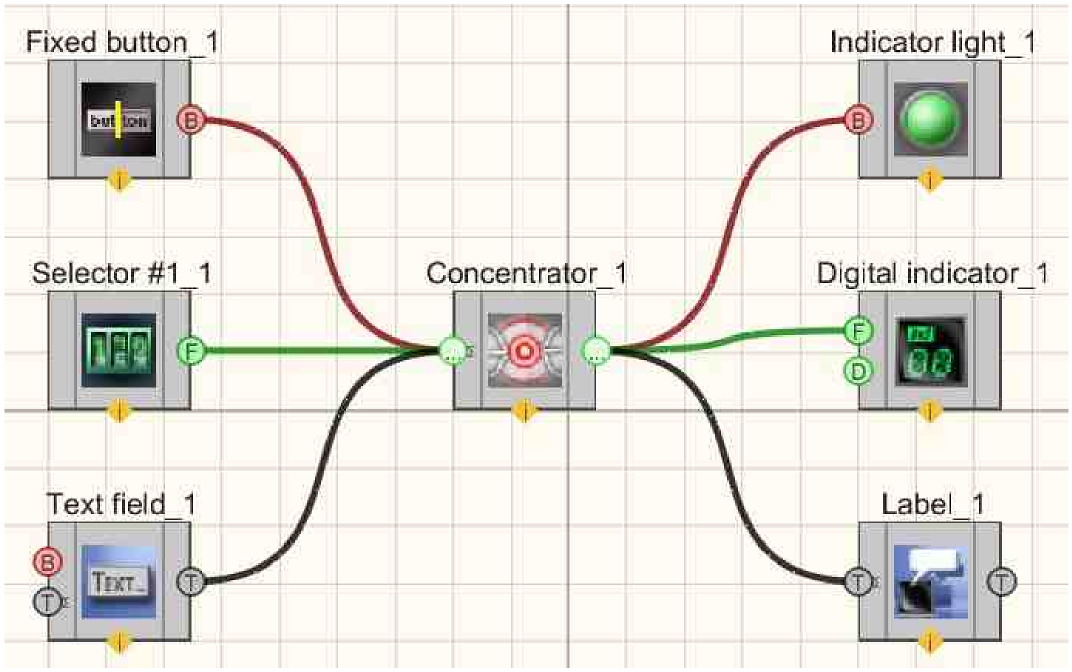
➢Custom private properties (the default value is shown in parentheses):

•BSTR Input - set the input: Left; Right (string).

•BSTR Output - set the output setting: Left; Right (string).

Example

Project in SCADA ZETView:



Project operation result

View the example in ZetView

23.5.Cross-page link

The component "**Cross-page link**" is designed for communication between components located on different pages of the project.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input (only available in transmit mode) - an input pin to which any type of data can be transmitted.

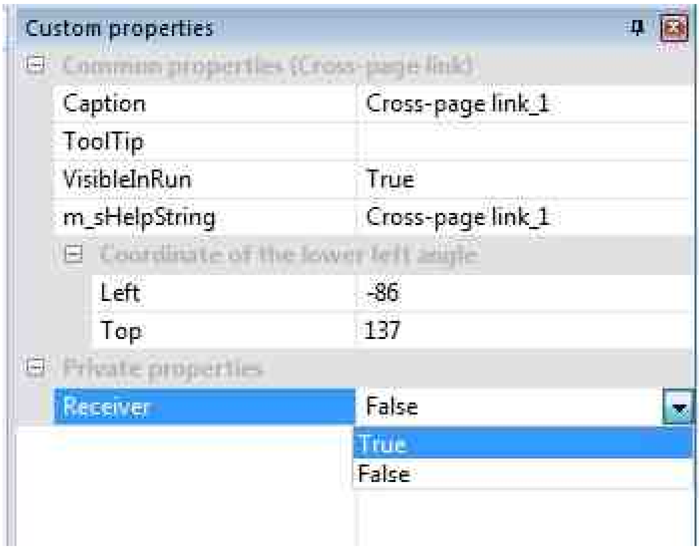
➢ Output:
- Output (only available in receive mode) - an output contact from which data of any type received from the input is transmitted.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Reciever (false) - accepts True / False values , puts the component in transmit or receive mode. Depending on the set mode, the component's appearance and arrangement of pins changes: the component has only an input pin if Reciever = True , and only an output pin if Reciever = False

The component has a customization feature.

First you need to set the property Receiver = True , on those "Inter-page link" components that will play the role of receiving components:



After that, on the transmitting component, using the context menu, select the "Edit component" item, then a menu will appear with a choice of receiving components "Cross-page link":

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input (only available in transmit mode) - Values to which data of any type can be transmitted.

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

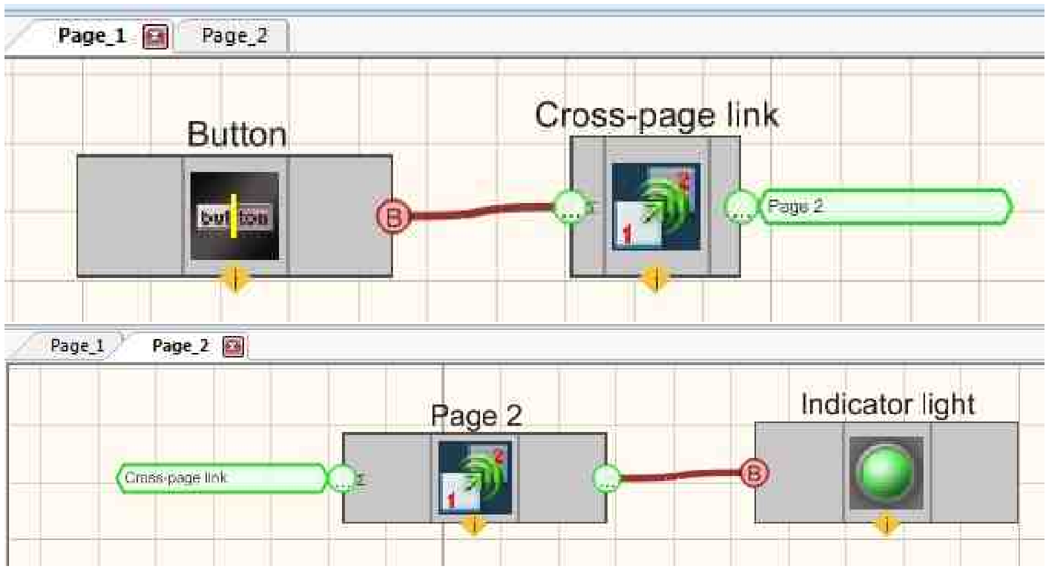
- VARIANT_BOOL Receiver - Set the type of Cross-page link:
  true - On cross-page link type;
  false - Off cross-page link type.

➢**Methods:**

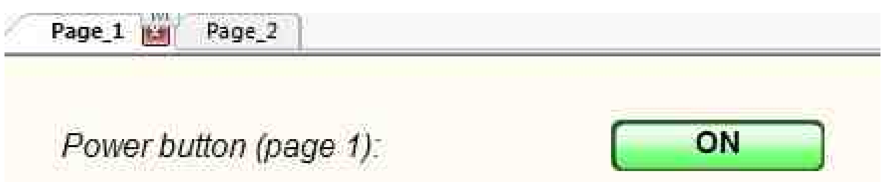
- PrepareDataForRequest(BSTR* helpStr, LONG* param, LONG* paramT) - Set the data value for the request.

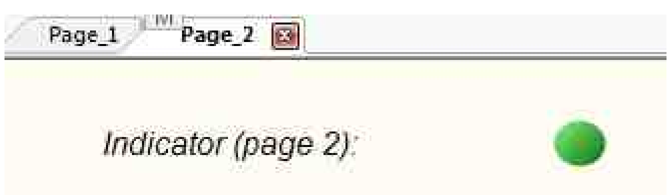
Example

Project in SCADA ZETView



Project operation result

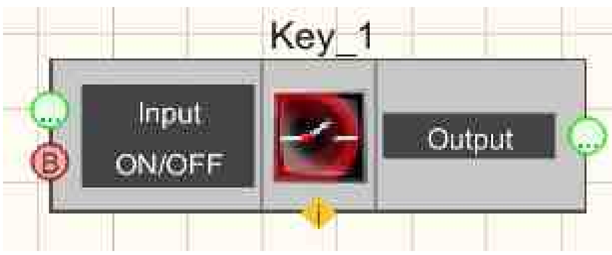




23.6.Key

The component "**Key**" is disconnects or maintains the connection, depending on the status of the control input.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - data of any type is transferred.
- ON/OFF - control impulse. If "1" is constantly applied to this input, the key is closed and data from the input is transferred to the output, if "0" - the key is open and no data is sent to the output.

➢ Output:

- Output - the data received from the input is transmitted with the key closed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Status (false) - key activation/deactivation current status.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - A value to which data of any type can be passed.

➢**Custom common properties:**

- Are available by the link below. 154

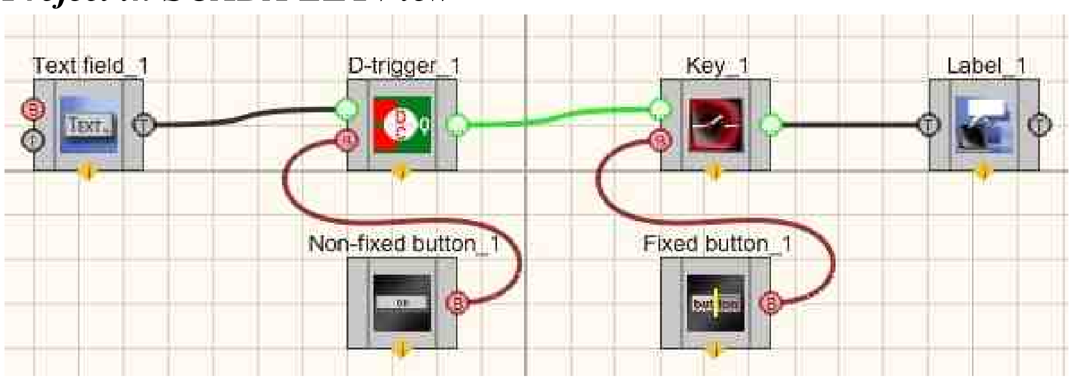
➢**Custom private properties (the default value is shown in parentheses):**

VARIANT_BOOL Status - Set the current status:

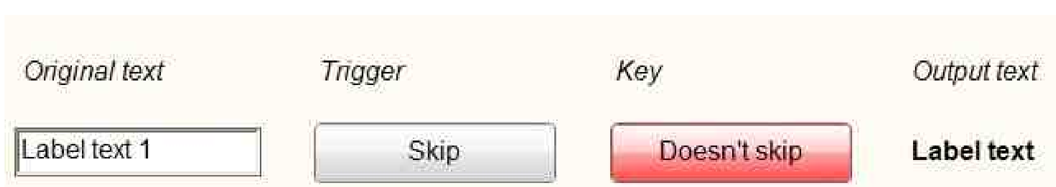
- true - the running status of the current status is on;
  false - the running status of the current status is off.

Example No. 1

Project in SCADA ZETView



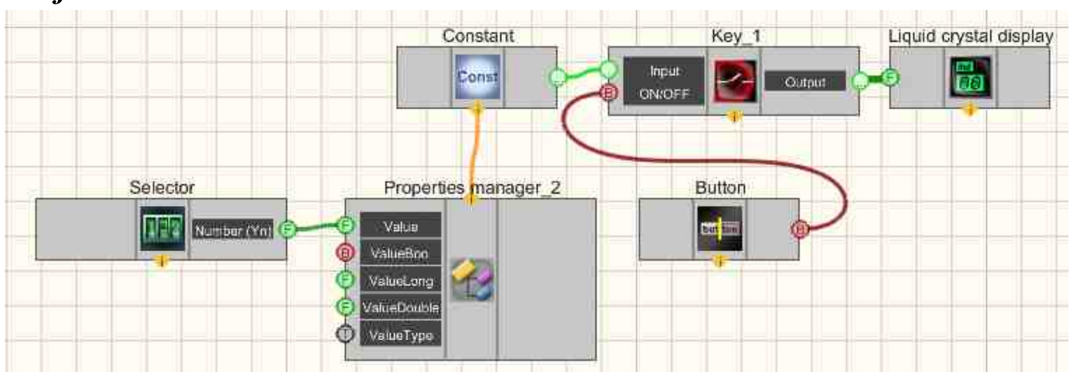
Project operation result



View the example in ZetView

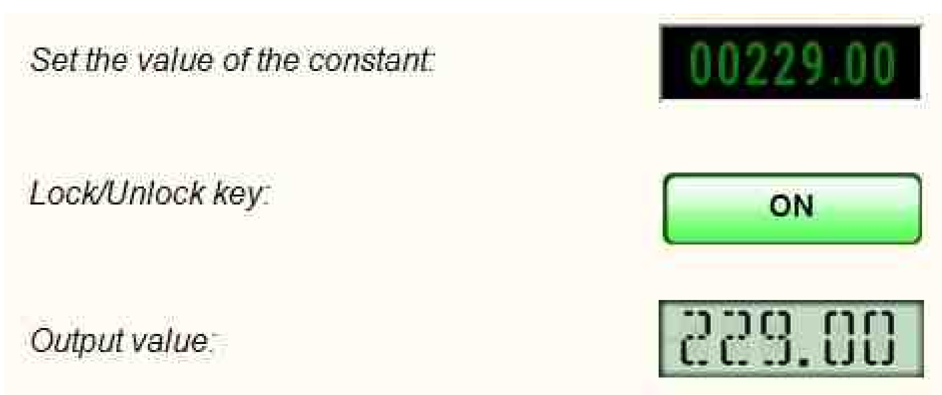
Example No. 2

Project in SCADA ZETView



To the input of the Key 996 component a Constant 417 is supplied, the component is controlled by a Fixed button 691. In order to get access to additional inputs of the Constant component, the Property Manager 999 is connected to it, with the help of which the input becomes available, for setting the value of the transmitted constant using a Selector 428.

Project operation result



View the example in ZetView

23.7.Property management

The component "**Property management**" is designed to manage the properties of other components and allows you to add the missing pins to the component, which are available in the private properties of the component.

Components have a special input for connecting the property manager, which is located at the bottom of the component in the form of a triangle. If there is no special input, then the component can only be controlled through input pins, and Doesn't have private properties.

Each ZETView SCADA component has contact pads for interaction with other components. Also, each component has different properties (the color of the indicator digits, the accuracy of displaying the numerical values of the digital indicator, the time of accumulation of data from measuring instruments, etc., etc.). Usually, the properties of a component are set at the design stage and are not changed anymore. In situations where it is necessary to manage some property of a component, and the corresponding pad is not provided for the component, the Property Manager is used.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input parameters completely repeat Custom common properties : the component to which the Property Manager is connected.

➢ Output:

Doesn't have.

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Note:

Question. I can’t edit the property manager and why blocks are without links
Answer. We have moved the connecting lines into a block in the file format. Property manager contacts are created automatically when attached to a component according to its property list.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

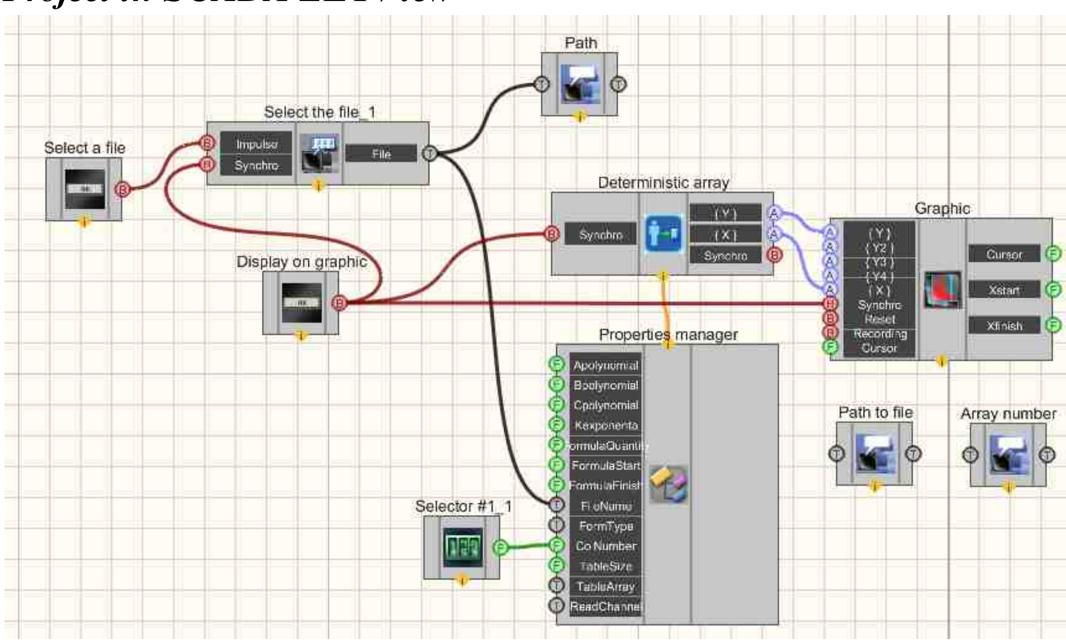
Setting:

➢**Custom private properties (the default value is shown in parentheses):**

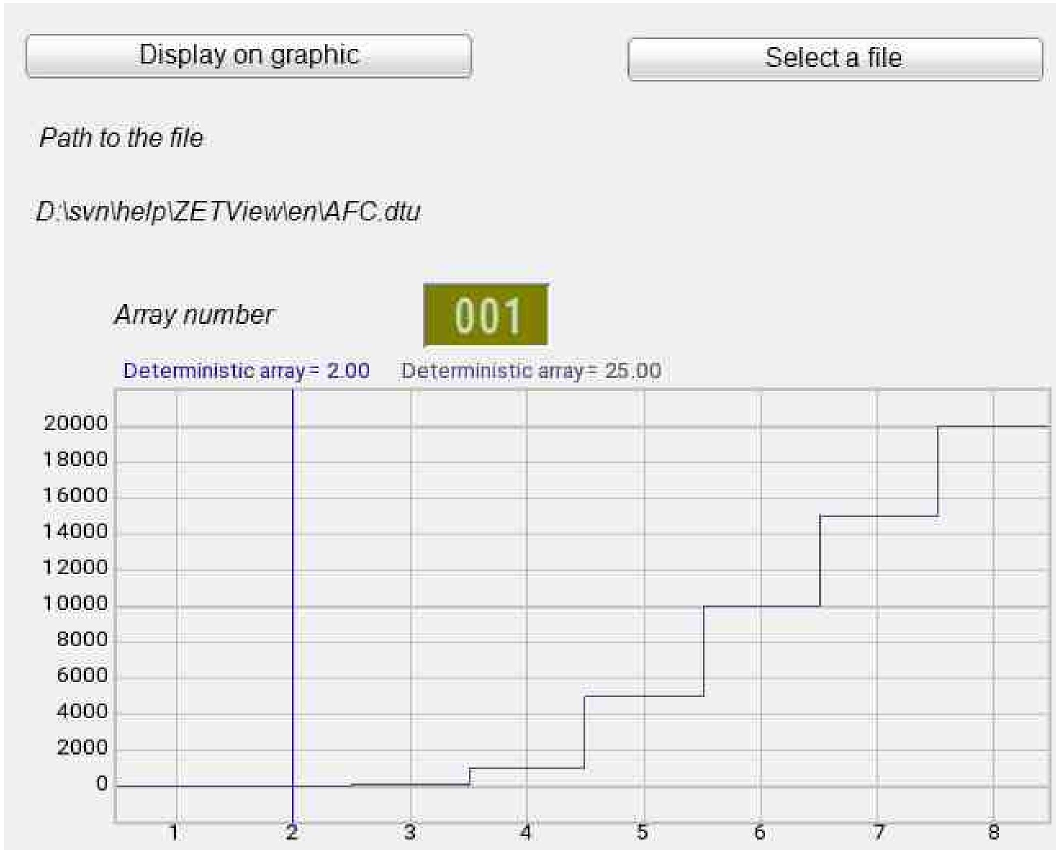
- Are available by the link below.154

Example No. 1

Project in SCADA ZETView



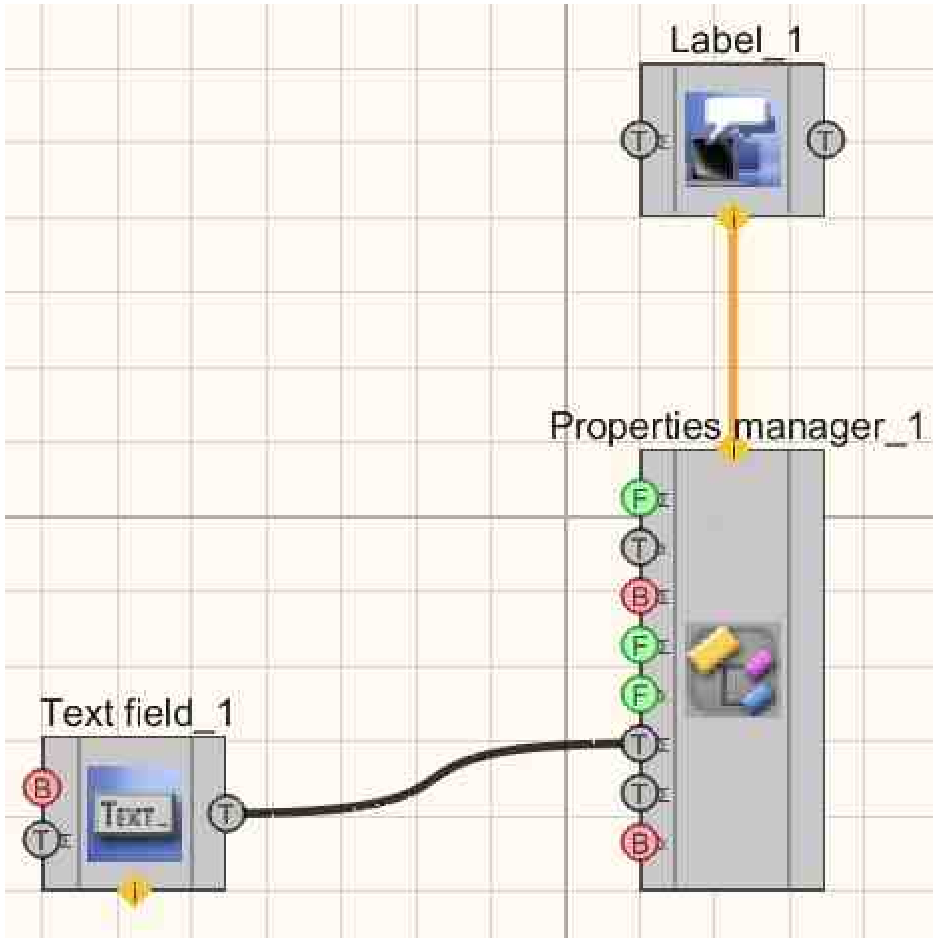
Project operation result



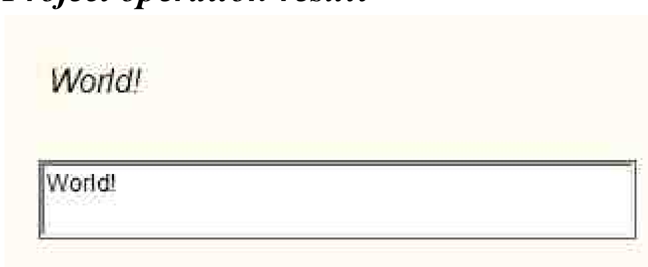
View the example in ZetView

Example No. 2

Project in SCADA ZETView



Project operation result

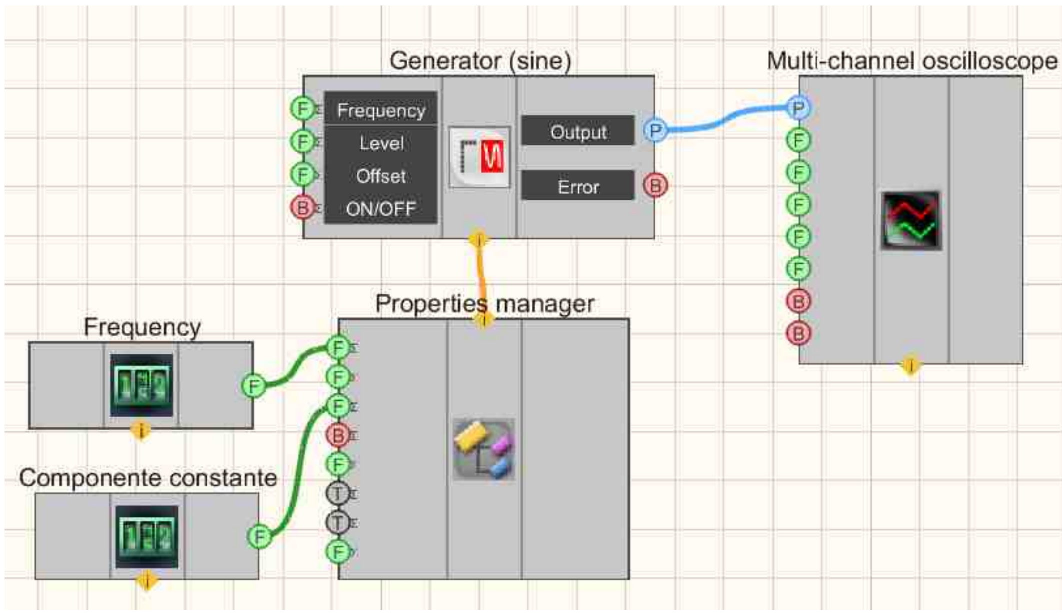


View the example in ZetView

Example No. 3

Note: You can run as many programs in SCADA or in ZETLab as there are DAC generators connected, otherwise the programs will not work correctly.

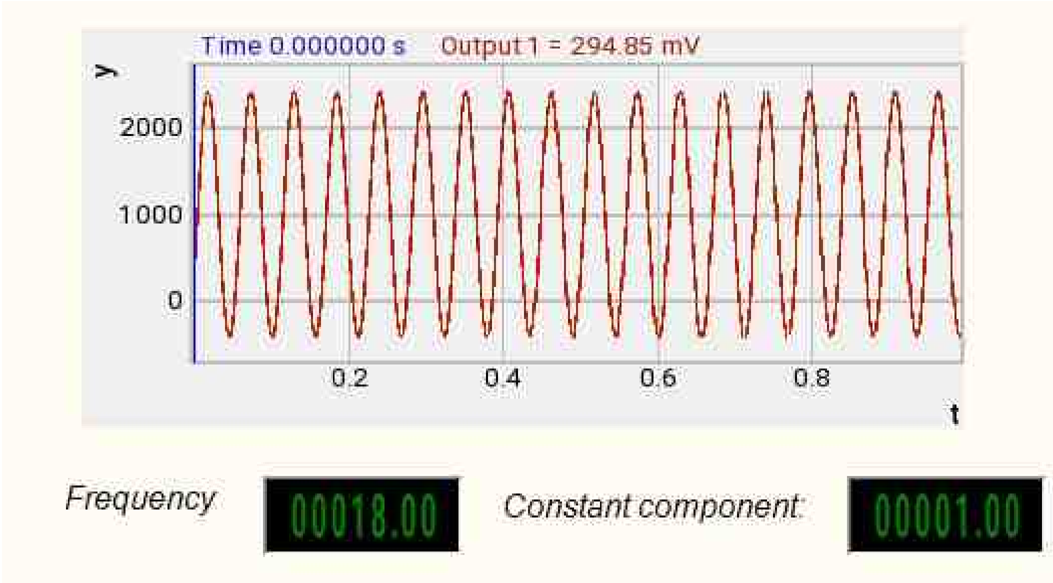
Project in SCADA ZETView



Project in SCADA ZETView

This project shows the operation of the Property Manager 999 component exposes the private properties of Sine Wave to other modules. The Selectors 428 are used to set the frequency and the DC component of the Sine signal 530. A Multi-channel oscilloscope 928 is used to graphically display the generator signal.

Project operation result

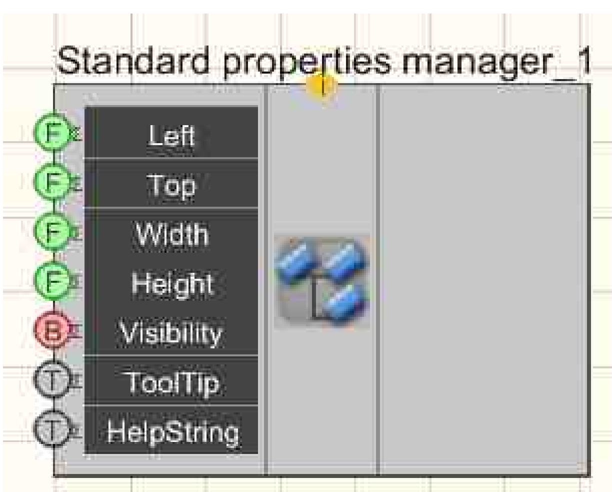


View the example in ZetView

23.8.Standard properties manager

The component "**Standard properties manager**" is designed to control the standard properties of the component location (Left, Top, Width, Height).

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:
- Left - sets the X coordinate of the upper left corner of the component (relative to the left side of the page)
- Top - set the Y coordinate of the upper left corner of the component (relative to the left side of the page)
- Width - sets the width of the component
- Heigth - sets the height of the component
- Visibility - sets the visibility of the component.
- ToolTip - hint for the user.
- HelpString - component description.

➢ Output:

Doesn't have.

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

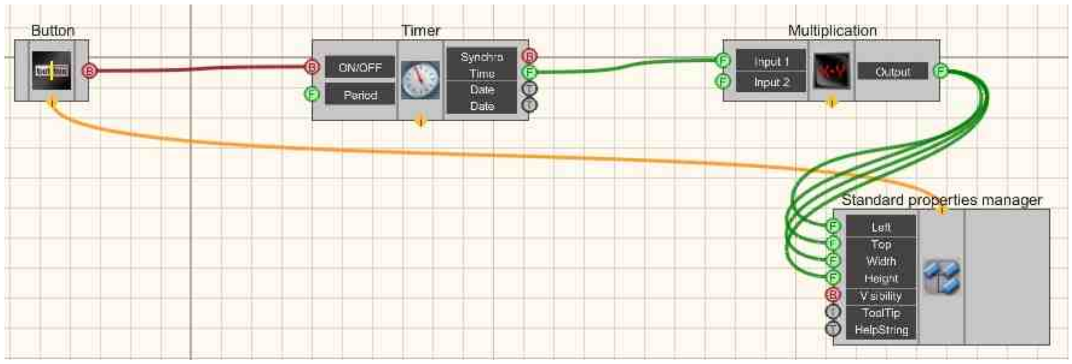
Setting:

➢**Input:**
- Left - Set the X coordinate of the top left corner of the component (relative to the left side of the page) (any number).
- Top - Set the Y coordinate of the top left corner of the component (relative to the left side of the page) (any number).
- Width - Set the width of the component (any number).
- Heigth - Set the height of the component (any number).
- ToolTip - Set the tooltip for the user.
- HelpString - Set the description of the component.

➢**Custom common properties:**
- Are available by the link below. 154

Example

Project in SCADA ZETView



This project shows the operation of the Standard properties manager 999 component allows you to manage component coordinates and Custom common properties. The component moves across the screen as it changes.

Project operation result

View the example in ZetView

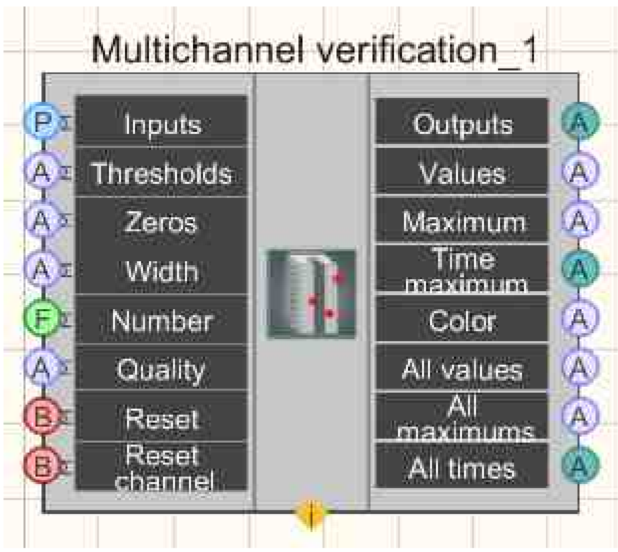
23.9.Multichannel verification

The component "**Multichannel verification**" is designed for processing a large number of measuring channels.

The component is used:

to determine the status of the measuring channels (availability, quality of data on the channel),
to display the current values by channels as an array,
to display the maximum channel values sorted in descending order,
to display information about out-of-range values.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Inputs - contains a list of input channels (If there is a "-" character before the channel name, then the channel is excluded from the calculation).
- Thresholds - an array of threshold values for channels.
- Zeros - an array of zero values for channels.
- Width - an array of values for the width of the working range.
- Number - number of the reset channel (-1 - all).
- Qualities - an array of quality flags for the input signal.
- Reset - reset the maximum fixed values for all channels.
- Reset channel - reset the maximum fixed value for the specified channel.

➤ Output:

- Outputs - an array of channel names sorted in descending order of their maximum recorded values
- Values - an array of current values, sorted according to the output array of channel names.
- Maximum - an array of maximums, sorted according to the output array of channel names.

- Maximum time - an array of times of obtaining the maximum value, sorted according to the output array of channel names.
- Color - an array of colors (numbers in RGB format) that displays the status of the channels, according to the input list.
- All values - an array of measured values for all channels, according to the input list.
- All maximums - an array of maximum values for all channels, according to the input list.
- All Times - An array of peak times for all channels, according to the input list.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- averagetime (1) - set the averaging time, s:
  1. 0.1 s;
  2. 1 s;
  3. 10 s.

Attention! *The sampling frequency of the channels connected to the component must be greater than 1/(averagetime).*
- ReturnQuantity - the size of returned arrays.
- measuretype (relative) - method of processing values, affects the form in which the output data will be displayed: absolute; relative.
- calcmethod (by range) - method for determining if the range is out of bounds: by range; along the threshold.
- thresholdtype (higher) - selection of threshold exceeding direction: higher; below.

➢Deciphering colors (in order of priority of their issuance):
- RGB(255, 0, 255), magenta, update information
- RGB(128, 128, 128), grey, channel excluded from calculation
- RGB(0, 255, 255), cyan, no channel
- RGB(255, 165, 0), orange, poor data quality
- RGB(255, 255, 0), yellow, data quality unknown
- RGB(255, 0, 0), red, out of range
- RGB(0, 255, 0), green, signal within limits

➢calculation using absolute values (measuretype = Absolute):
- Threshold comparison (calcmethod = threshold)
  - above (thresholdtype = above)
    - If the maximum value is less than the threshold value, then the norm
  - below (thresholdtype = below)
    - If the maximum value is greater than the threshold value, then the norm
- Comparison by width (calcmethod = by width)

- If the modulus of the maximum value is less than the modulus of the width, then the norm

➢calculation using relative values (measuretype = Relative):
- The ratio of the channel value to the width is calculated as a percentage. If it is less than 100%, then the norm.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Inputs - contains a list of input channels.
- Thresholds - an array of threshold values for channels.
- Zeros - an array of zero values for channels.
- Width - an array of values for the width of the working range.
- Number - number of the reset channel (-1 - all).
- Qualities - an array of quality flags for the input signal

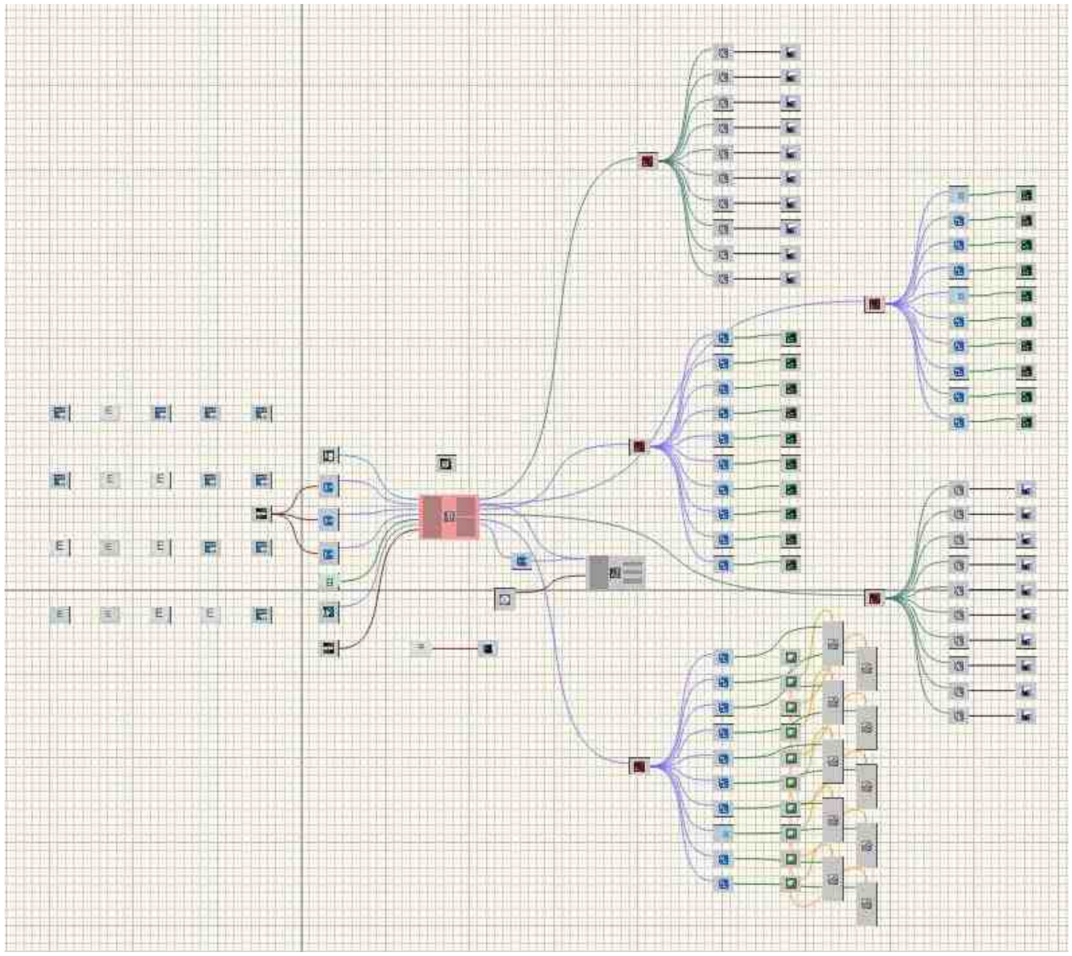
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR AverageTime - Set the averaging: 0.1 s; 1 s; 10 s (string)..
- LONG ReturnQuantity - Set the size of returned arrays (number).
- BSTR MeasureType - Set the method of measuring values: absolute; relative (string)..
- BSTR CalcMethod - Set the calculation method by range; by threshold (string)..
- BSTR ThresholdType - Set a warning when the value becomes higher or lower than the threshold (string)..

Example

Project in SCADA ZETView



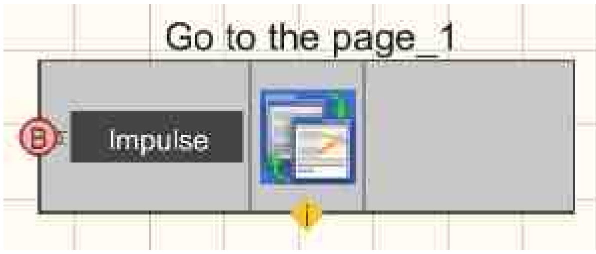
Project operation result

View the example in ZetView

23.10.Go to the page

The component "**Go to the page**" is used to upon receipt of a control pulse to the input, it goes to the specified page.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Impulse - impulse to move to another page of the project.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- pagenametogo - the name of the page to go to.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

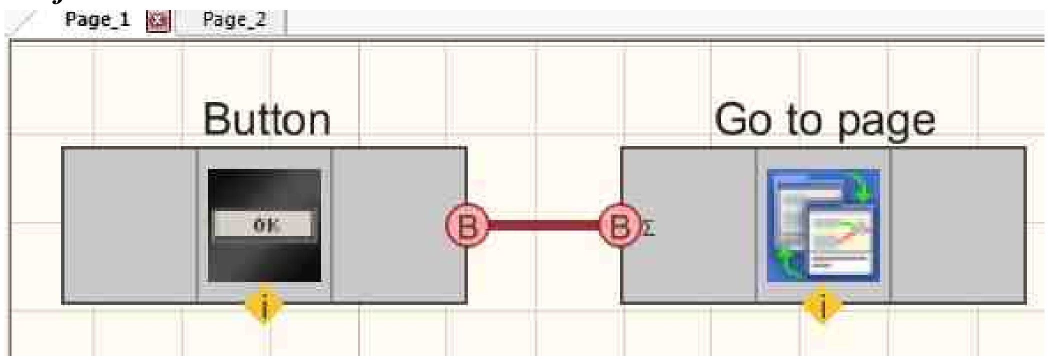
➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

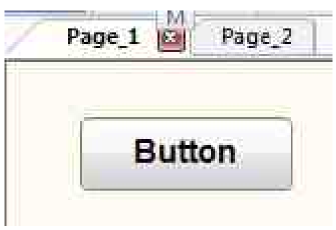
•BSTR PageNameToGo - Set the name of the page to go to (string).

Example

Project in SCADA ZETView



Project operation result



23.11.Ping

The component "**Ping**" is used to pings the given address.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Synchro - ping starts on an impulse.

➢ Output:

- Status - true if the ping was successful at least once; false if the ping never went through.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Host (164.0.0.1) - pinged address.
- Count (3) - number of ping attempts.
- Timeout (2000) - timeout for one attempt, ms.
- PacketSize (32) - packet size, bytes.
- TypeResult (By all) - type of result of ping to several addresses: By all; By anyway.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

- Are available by the link below. 154

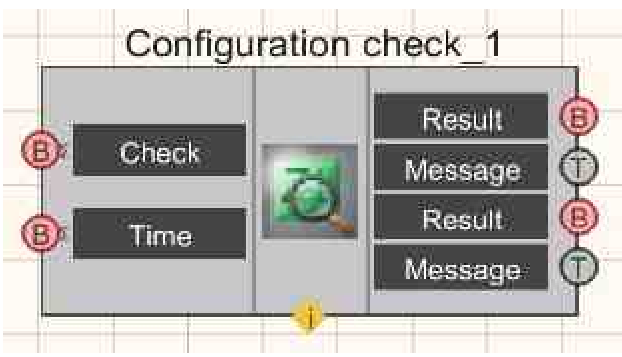
➢Custom private properties (the default value is shown in parentheses):

- BSTR Host - Set the name or IP of the pinged address (string).
- LONG Count - Set the number of attempts.
- LONG Timeout - Set the timeout for one attempt, ms.
- LONG PacketSize - Set the packet size, bytes.
- BSTR TypeResult(By all) - Set the type of result of ping to several addresses: By all; By anyway (string).

23.12.Checking the configuration

The component "**Configuration check**" is designed to check the configuration.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Check - check configuration: devices, channels, programs.
- Time - checking the performance of the program.

➢ Output:
- Result - the result of checking the configuration.
- Message - a message about the status of the system.
- Result - the result of checking the status of the channels.
- Message - a message about the status of the channels.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- ConfigFileName - the name of the configuration file to check the current status of the system.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•BSTR ConfigFileName - Set the configuration file to check the current status of the system (string).

23.13.Listening to the channel

The component "**Listening to the channel**" is used for designed for play recorded sound coming from channels.

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:

- Input - the input channel to which the signal is received, from which the required values need to be measured.
- Volume - the volume level.
- Gain - signal gain.
- ON\OFF - turn on and off listening to the channel.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Volume (50) - the volume.
- Gain (0) - the gain, dB.
- Activate (false) - listening status (enabled or disabled).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - the channel to which the signal channel is connected (from 0 to (number of channels - 1)).
- Volume - the channel to which the volume level (number) is connected.
- Gain - the channel to which the signal gain (number) is connected.

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•FLOAT Volume - Set the volume (number).

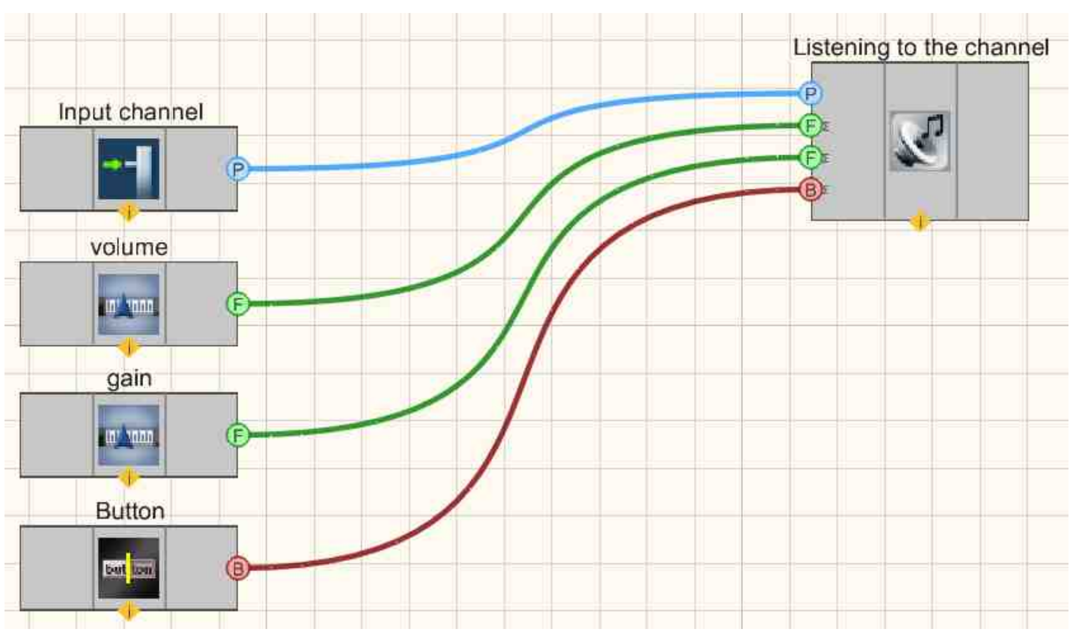
•FLOAT Gain - Set the gain, dB (number).

VARIANT_BOOL Activate - Set the operation status of listening to a channel:

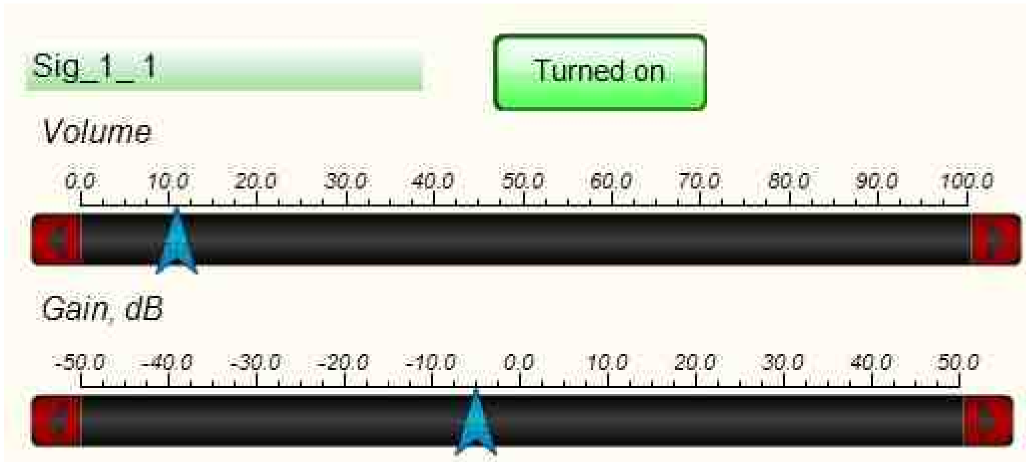
- true - the status of the operation to the channel is enabled;
  false - the status of the operation listening to the channel is disabled.

Example

Project in SCADA ZETView



Project operation result



View the example in ZetView

23.14.Random number

The component "**Random number**" is designed to generate random numbers.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- ON\OFF - turn on and off random numbers.

➢ Output:

- Output - a random number is obtained.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Activate (true) - the status of the random number generator.
- MinimumOfDistribution (-1) - minimum distribution value.
- MaximumOfDistribution (1) - maximum distribution value.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

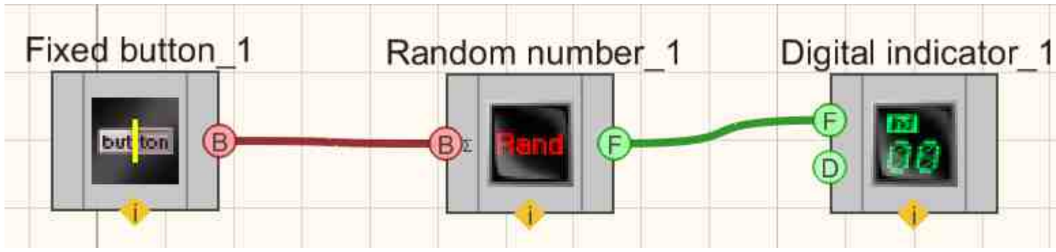
VARIANT_BOOL Activate - Set the status of the random number generator:

- true - The status of the random number generator is enabled;
  false - The status of the random number generator is disabled.

- FLOAT MinimumOfDistribution - Set the minimum distribution value (number).
- FLOAT MaximumOfDistribution - Set the maximum distribution value (number).

Example No. 1

Project in SCADA ZETView

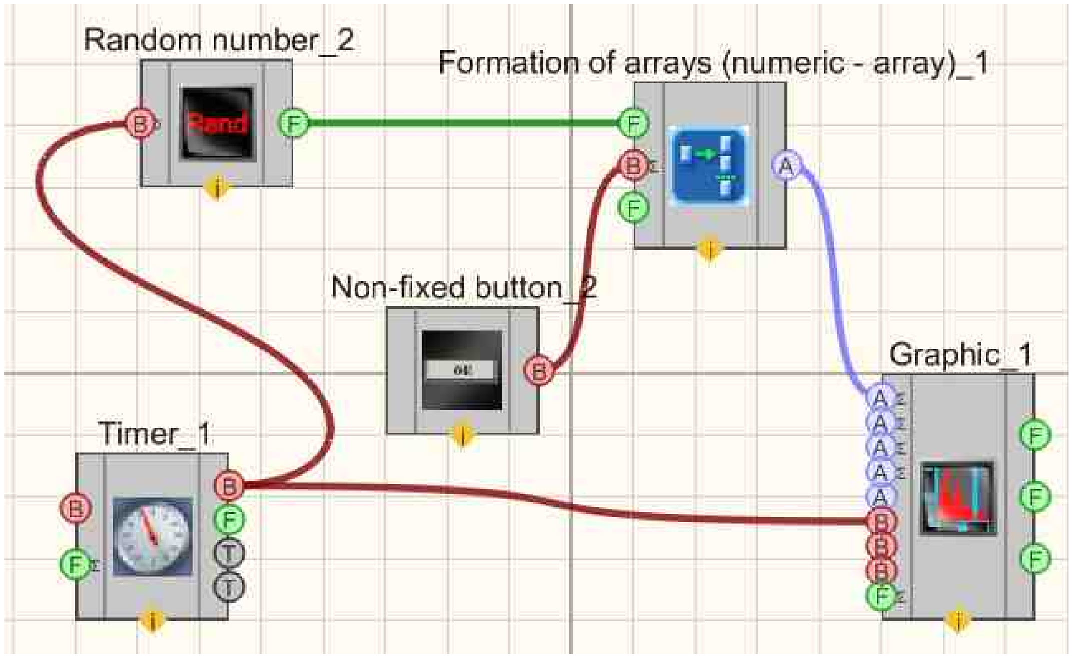


Project operation result

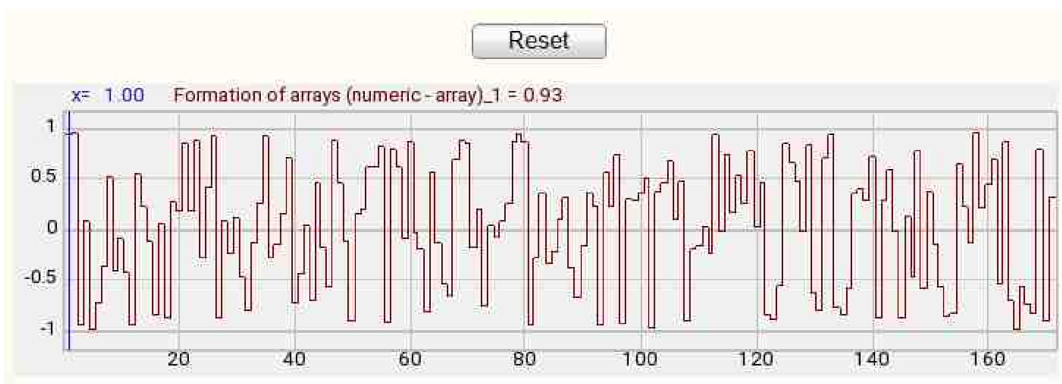
View the example in ZetView

Example No. 2

Project in SCADA ZETView



Project operation result

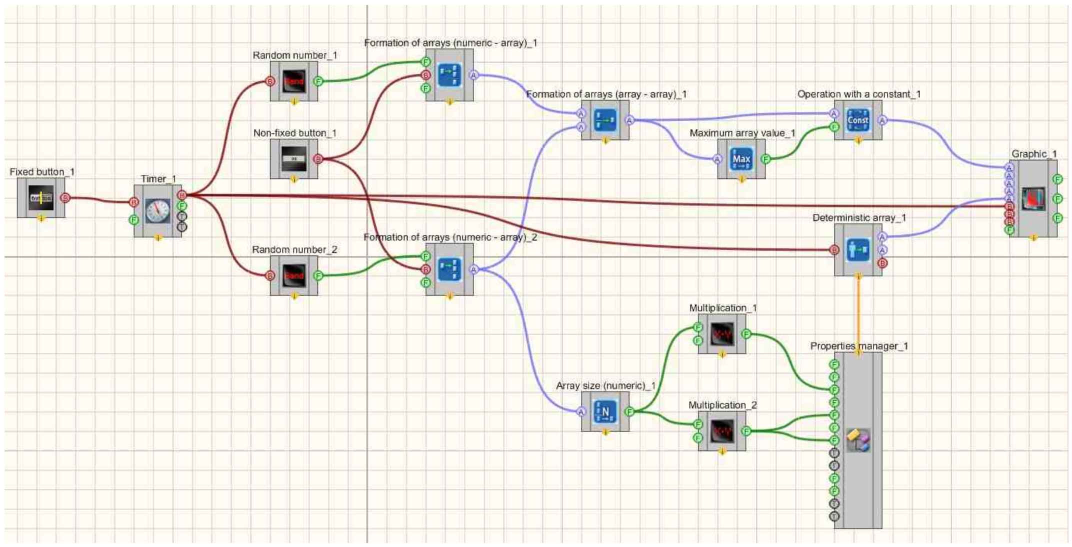


View the example in ZetView

Example No. 3

Example "Miscellaneous - random corr.zvx" indicates that the random numbers are "good" and running two random number generators at the same time produces uncorrelated random number sequences.

Project in SCADA ZETView



Project operation result

View the example in ZetView

23.15.Reading startup parameters

The component "**Reading startup parameters**" is designed to read project launch options from the command line. The entire text of the command to launch the project from the command line is sent to the output contact of the component at startup.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢ Input:

Doesn't have.

➢ Output:

- String - Gets the command line.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

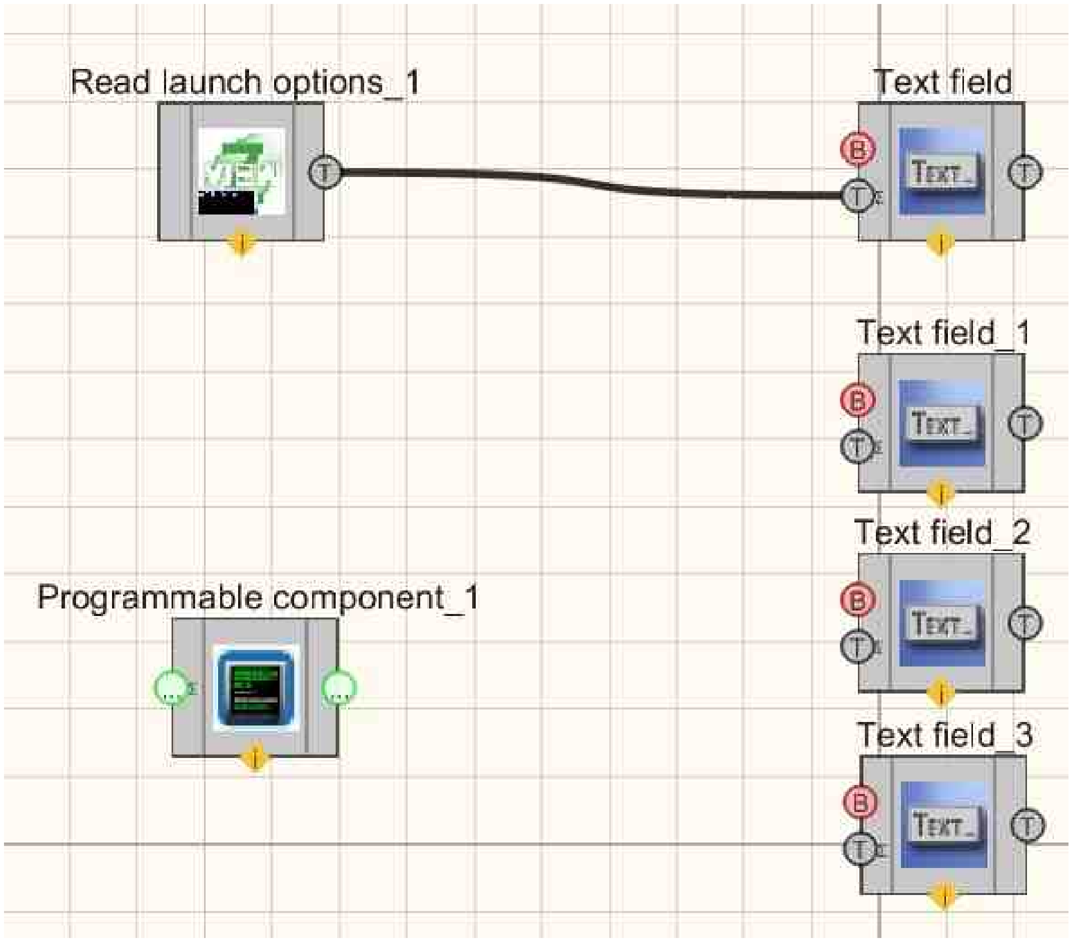
➢**Methods:**

- BSTR GetCommandLineText() - Sets the command line text (string). Take a completely text command to launch the project from the command line.
- LONG GetCommandLineArgsNumber() - Sets the number of command line arguments (number). Take the number of project startup arguments from the command line.
- BSTR GetCommandLineArg(LONG number_) - Sets the command line argument text. take a specific argument from the text command for calling the project from the command line by its number.

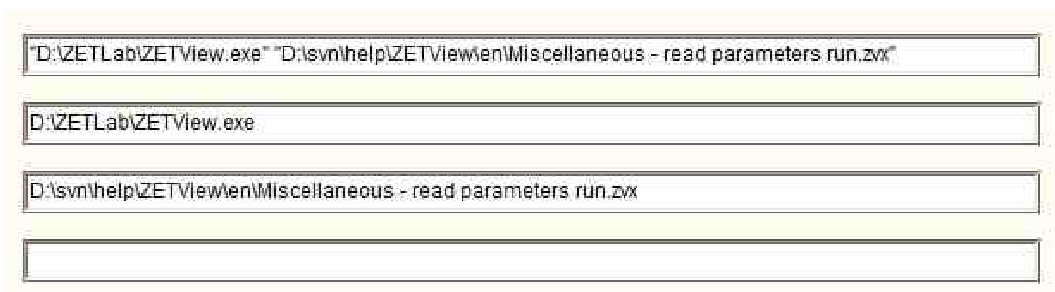
Take a specific argument from the text command for calling the project from the command line by its number.

Example

Project in SCADA ZETView



Project operation result



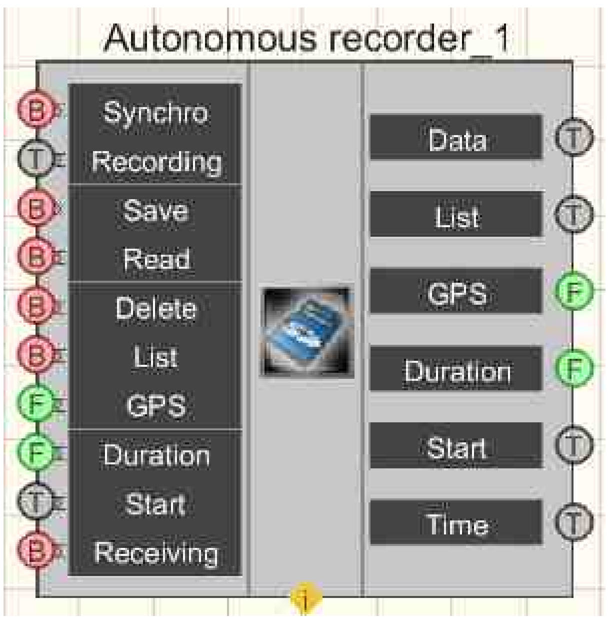
View the example in ZetView

Глава 24.Registrator

24.1.Autonomous recorder

The component "**Autonomous recorder**" is designed to create a scenario for the operation of a stand-alone recorder: synchronization, data exchange and setting recording modes.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Synchro - synchronize the recorder's clock with the computer's clock.
- Recoding - text to record to the log file.
- Save - command to record to the log file.
- Read - command to read a file from the offline recorder's media.
- Delete - delete a file from the registrar.
- List - get a list of files from the registrar.
- GPS - set the delay to turn on the GPS, s.
- Duration - set the registration duration, s.

- Start - set the start time of registration.
- Receiving - Receive the value of variables.

➢ Output:

- Data - log file data.
- List - list of log files.
- GPS - delay value for turning on GPS, s.
- Duration - duration of registration, s.
- Start - the start time of registration.
- Time - the internal time of the recorder.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- LogFileName (LOG.log) - the name of the log file LOG.log.
- CompFileName - file name on the computer *. txt.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Recoding - Set to log messages to a log file.
- GPS - Delay value for turning on GPS, s (from 1 to 30).
- Duration - Registration duration value, s (calculated by the Autonomous Recorder program and depends on the flash volume, the number of channels, the sampling frequency and the size of the allowable program buffer).
- Start - String value to set the start time of registration (text).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR LogFileName - Set the name of the log file to LOG.log (string).

- BSTR CompFileName - Set the file name to be written to the *. txt (after setting this parameter, writing to the specified file occurs) (string).

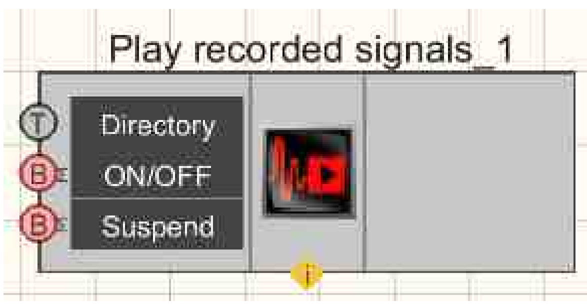
➢**Methods:**

- void AllowUpdateTime(BYTE allow) - set the update time permission.

24.2.Play recorded signals

The component "**Play recorded signals**" is designed to read recorded temporary implementations from data files for the purpose of processing, studying and analyzing, for Example, when this could not be done under the conditions of the measurement.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Directory - Directory that holds recordings for Play recorded signals.
- OnN/OFF - enable/disable Play recorded signals.
- Suspend - stop and resume Play recorded signals.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Directory - Set the name of the directory.

- Velocity (true) - Set the flag for increased playback velocity.
- Infinite (false) - Set the flag for continuous playback.
- UnitVisible (false) - Set the visibility of the program being run by Reader.exe.
- Activate (false) - Set the operation status.
- Paused (false) - Set the pause Play recorded signals.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Directory - Text string value indicating the path to the folder containing *.ana, *.anp data archive. and locations on disk (text).

➢**Custom common properties:**

- Are available by the link below.154

➢**Custom private properties (the default value is shown in parentheses):**

•BSTR Directory - Set the directory from which files are taken for Play recorded signals.
VARIANT_BOOL Velocity - Set the flag for increased Play recorded signals velocity:

- true - flag of increased playback speed is enabled;
  false - flag of increased playback speed is disabled.

VARIANT_BOOL Infinite - Set the flag for continuous playback:

- true - the continuous Play recorded signals velocity flag is enabled;
  false - the continuous Play recorded signals velocity flag is disabled.

VARIANT_BOOL UnitVisible - Set the visibility of the running program Reader.exe:

- true - program visibility is enabled;
  false - program visibility is disabled.

VARIANT_BOOL Activate - Set the status of the signal Play recorded signals operation:

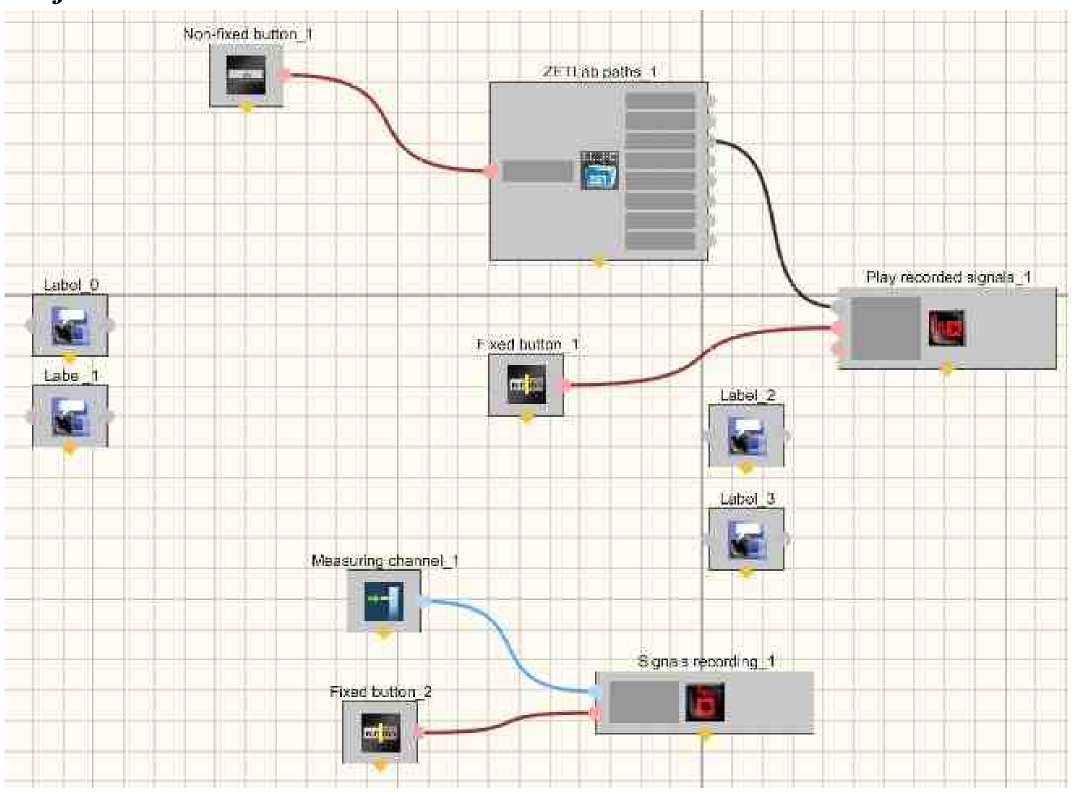
- true - the operation status of Play recorded signals is enabled;
  false - the operation status of Play recorded signals is disabled.

VARIANT_BOOL Paused - Set Play recorded signals pause:

- true - Play recorded signals pause is enabled;
  false - Play recorded signals pause is disabled.

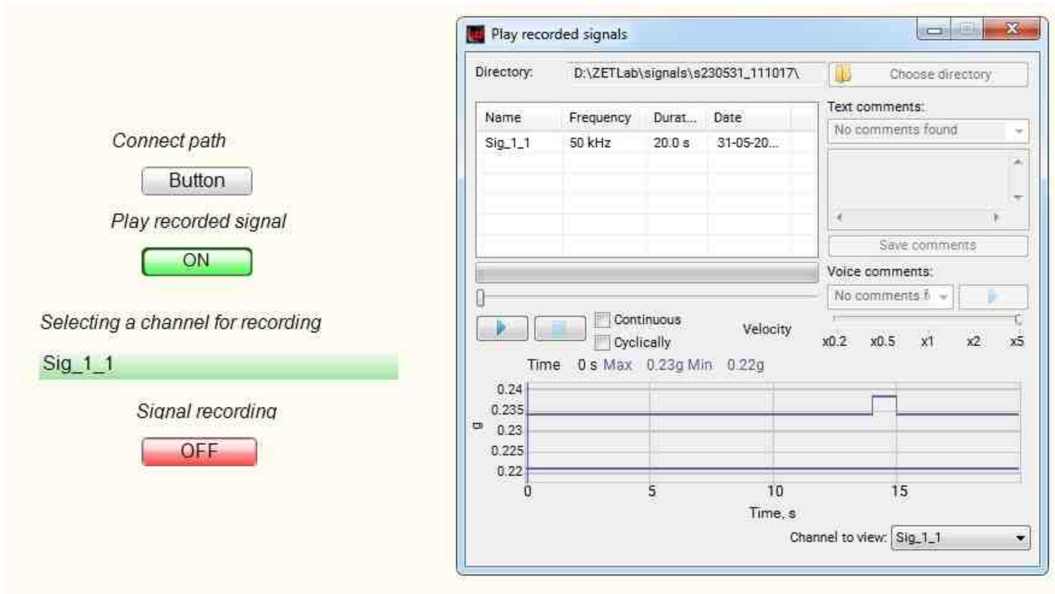
Example

Project in SCADA ZETView



Note: Project List of the project execution sequence:
1) Select the signal to be recorded.
2) Turn on the Signals recording.
3) We connect the path for the Play recorded signals program.
4) We reproduce the received signal.

Project operation result



View the example in ZetView

24.3.Signals recording

The component "**Signals recording**" is designed for continuous recording (recording) of signals from ADC channels to a file.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Inputs - list of recorded channels.
- On/OFF - enable/disable signals recording.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Duration (100) - recording duration, s.
- ChannelList - the current list of channels (if the list is empty, then the default is to record all channels).
- ExcludeList - words or parts of words (separated by \";\"), if found in the channel name, this channel will not be recorded.
- Infinite (false) - the continuous recording flag.
- UnitVisible (false) - the visibility of the program being run SignalWriter.exe.
- Activate (false) - the operation status.
- Preamble (0) - the preamble duration, s.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Inputs - the channel to which the list of recorded channels is connected (from 0 to (number of channels - 1)).

➢**Custom common properties:**

- Are available by the link below. 154

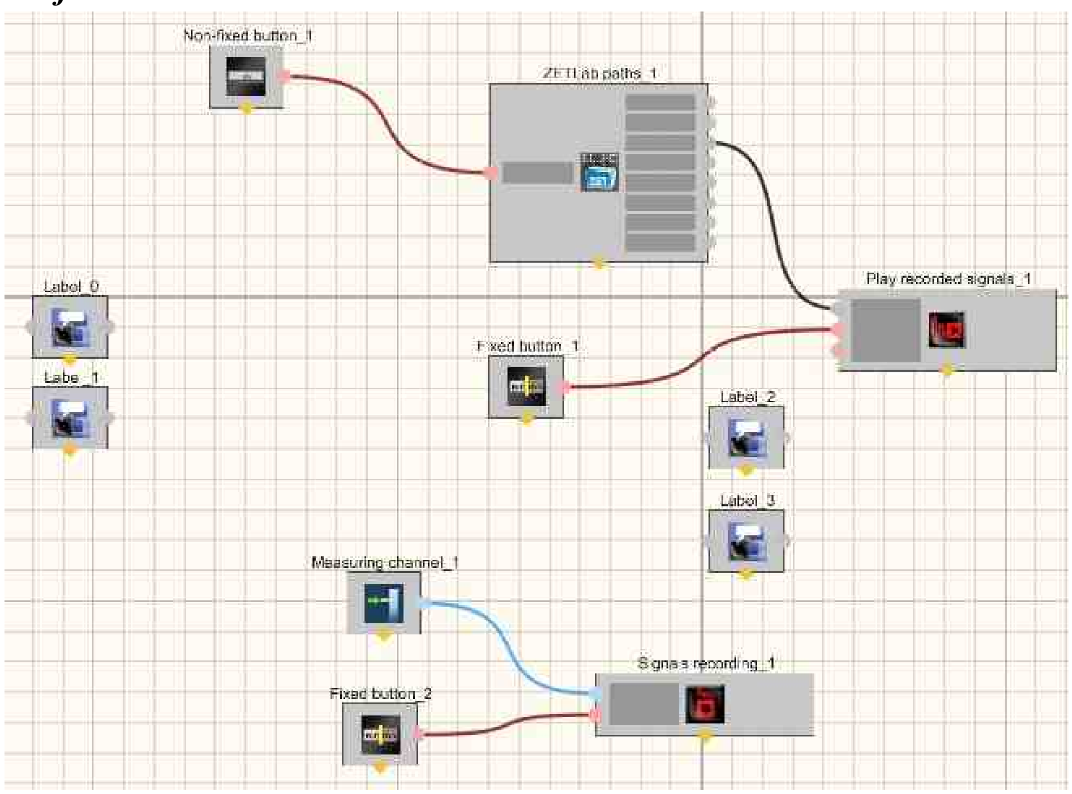
➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Duration - Set the duration of the recording (from 10 s to 3600 s).
- BSTR ChannelList - Set the current list of channels (if the list is empty, then the default is to record all channels) (string).

- BSTR ExcludeList - Set words or parts of words (delimited by \";\"), if found in the channel name, this channel will not be recorded (string).
- VARIANT_BOOL Infinite - Infinite - Set the continuous record flag:
  - true - the continuous recording flag is enabled;
  - false - the continuous recording flag is disabled.
- VARIANT_BOOL UnitVisible - Set the visibility of the running program SignalWriter.exe.
  - true - the program visibility is enabled;
  - false - the program visibility is disabled.
- VARIANT_BOOL Activate - Set the status of the signal recording operation:
  - true - the status of signal recording is enabled;
  - false - the signal recording operation status is disabled.
- FLOAT Preamble - Set the duration of the preamble (from 0 to 5 s).

Example

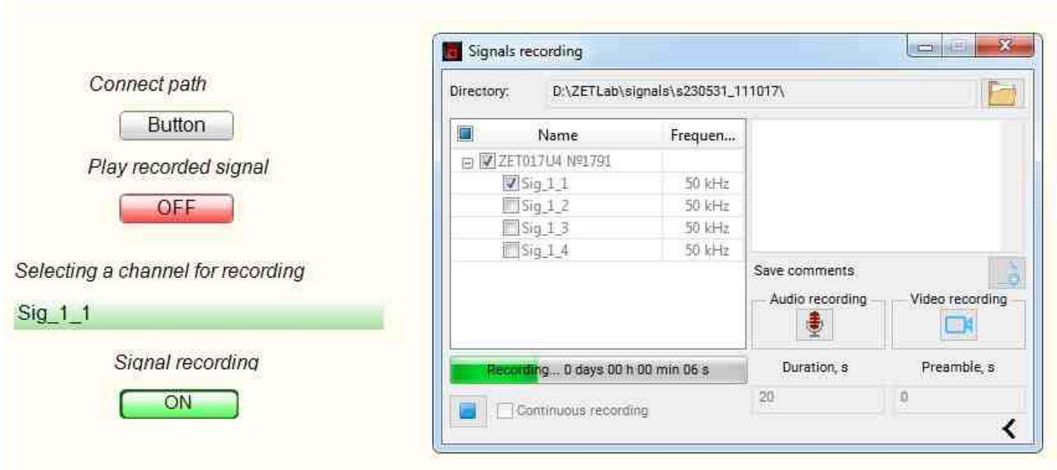
Project in SCADA ZETView



Note: Project List of the project execution sequence:
1) Select the signal to be recorded.
2) Turn on the Signals recording.
3) We connect the path for the Play recorded signals program.

4) We reproduce the received signal.

Project operation result



View the example in ZetView

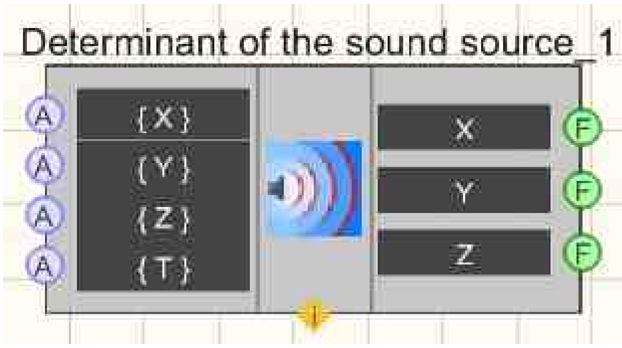
Глава 25.Solutions

25.1.Determinant of the sound source

The component "**Determinant of the sound source**" is designed to determine the location of the source of seismic waves.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- {X} - array of X coordinates.
- {Y} - array of Y coordinates.
- {Z} - array of Z coordinates.
- {T} - array of times.

➢ Output:

- {X} - X coordinate of the sound source.
- {Y} - Y coordinate of the sound source.
- {Z} - Z coordinate of the sound source.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- MaxDeltaT (1) - Set the maximum difference in event arrival times, at which calculation is possible (number).
- Velocity (1000) - Set the rate of sound in the environment, m/s (number).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- {X} - values of the array to which the input array of X coordinates is supplied.
- {Y} - values of the array to which the input array of Y coordinates is supplied.
- {Z} - values of the array to which the input array of Z coordinates is supplied.
- {T} - values of the array to which the input array of times T is supplied.

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- FLOAT MaxDeltaT - Set the maximum difference in event arrival time, at which calculation is possible (number).
- FLOAT Velocity - Set the speed of sound in the medium, m/s (number).

Глава 26.Messages to the operator

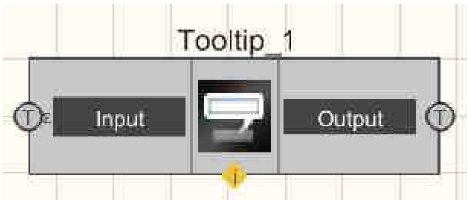
Components of "Messages to the operator" section are used for operations with notifications in SCADA-system ZETView:

- Tooltip – depicts additional information in operator interface.
- Event log - is used for events recording and representation in tabular form.
- Sound signaling - produces sound signal for the operator.
- Quality marker (to event log) - is used for urgent notifications for the operator.
- Advanced journal - events recording and representation as a text note in the log.
- Message - urgent messages for the operator.

26.1.Tooltip

The component "**Tooltip**" is used for depicts additional information in operator interface.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Input - a string is given as input.

➢ Output:
- Output - a string is output.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- MessageText (enter message text) - tooltip text.
- TimeDelay (500) - time delay, ms.
- OnOff (true) - enable or disable the tooltip.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Input - the value of the string that contains the tooltip entries (text).

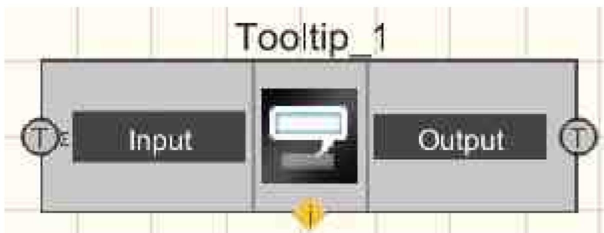
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- BSTR MessageText - Set the text of the tooltip (string).

- LONG TimeDelay - Set the time delay , ms (number).
- VARIANT_BOOL OnOff - Set whether to enable or disable the tooltip:
  true - The tooltip operation status is enabled;
  false - The tooltip operation status is disabled.

Example

Project in SCADA ZETView



The tooltip is used to increase the information content of the operator interface.

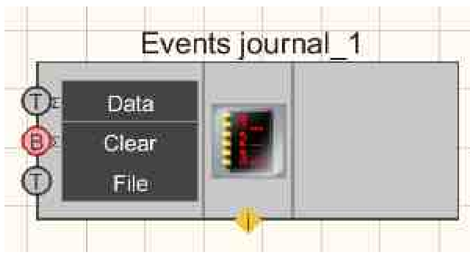
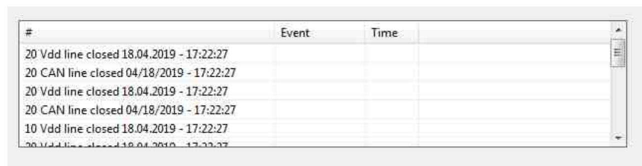
Project operation result

View the example in ZetView

26.2.Events journal

The component "**Events journal**" is used for events recording and representation in tabular form.

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td></td><td></td></tr>
</table>

Setting:

➢Input:

- Data - a line for logging data.
- Clear - when a pulse is given, the log is cleared.
- File is a string that contains the name of the file.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- m_sFileName (C:\TextLog.txt) - select the required file to which events will be recorded.
- CollumnHeaders (#; Event;Time ;) - sequential enumeration of column headers separated by a delimiter character.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Data - The value of the string that contains the data entries in the log (text)
- File - The value of the string that contains the file name (text)

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

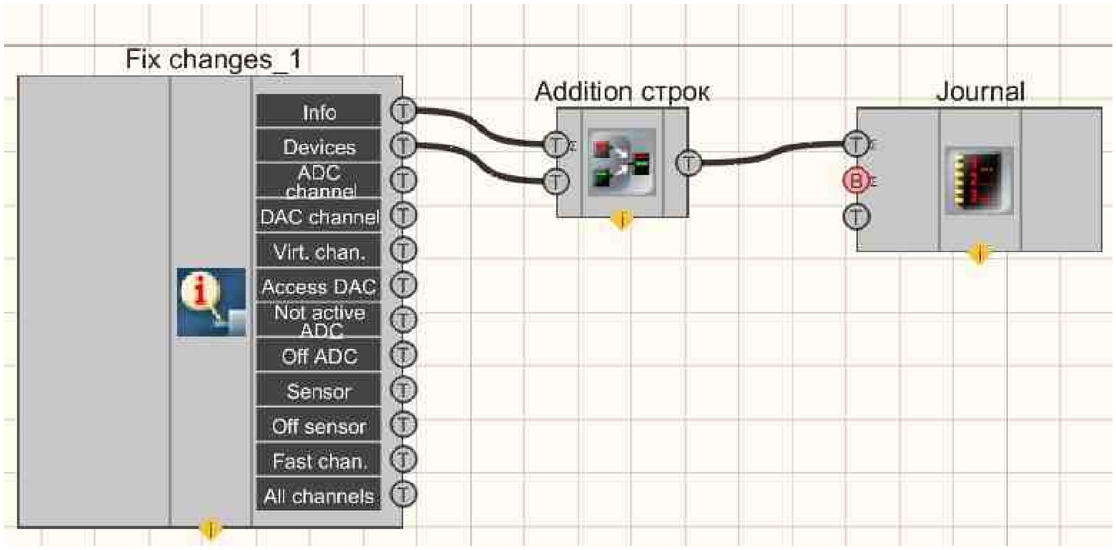
- BSTR m_sFileName - Set the path to the log file (string).
- BSTR CollumnHeaders - Set column headers (string).

➢**Methods:**

- LONG GetSelectedItem (void) - Get the current position of the cursor in the list.
- BSTR GetColumnText (LONG numLine , LONG numColumn) - Get text from selected column.
- void DeleteLine (LONG numLine) - Delete a line.
- void SortItems (LONG numColumn) - Sort by column text.
- void ReWriteFile (void) - Rerecord the event table.
- void ReReadFile (void) - Reread the event table.
- FLOAT CaclMeanBySelected (LONG numColumn) - Return the arithmetic mean of all selected elements in a column.

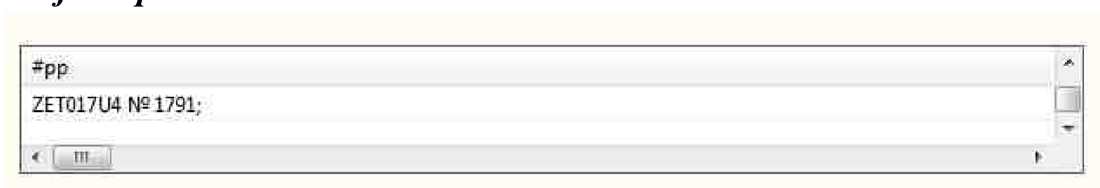
Example

Project in SCADA ZETView



This project shows the operation of the Fix changes 1040 component logs changes that occur with one of the external plug-ins in the Events journal 1037 component. But after preliminary gluing of information about the device and the changes occurring with it (Joining strings 1094 component).

Project operation result



26.3.Sound signaling

The component "**Sound signaling**" is used for produces sound signal for the operator.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Impulse - a contact, when an impulse is applied to it, a sound file is played.

➢ Output:
Doesn't have.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- FileName – select of a file that will be played upon receipt of an impulse.
- FlagLoopSound - Select to use the file to play recorded signals repeatedly.
-

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

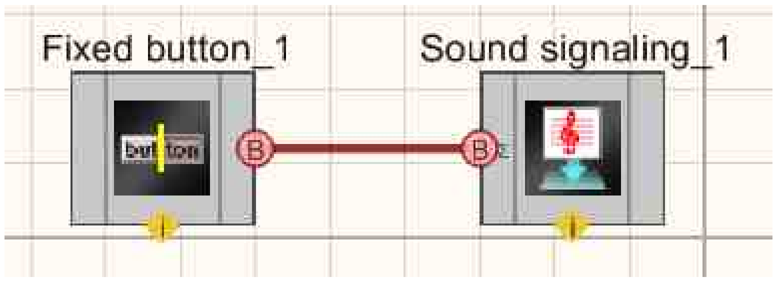
Setting:

➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- BSTR FileName - Set the file name (string).
- VARIANT_BOOL FlagLoopSound - Set the file to play recorded signals repeatedly.

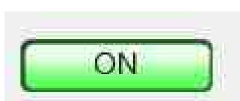
Example

Project in SCADA ZETView



This project shows the operation of the Sound signaling1042 component plays the specified sound file on pressing the Fixed button691 it.

Project operation result

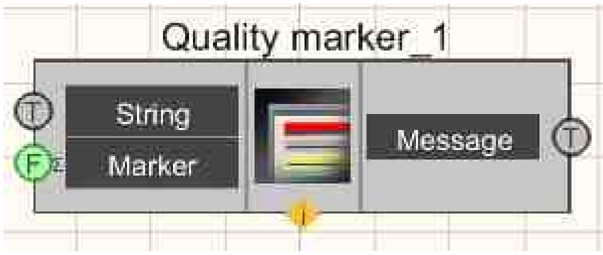


View the example in ZetView

26.4.Quality marker (to event journal)

The component "**Quality marker (to event log)**" is used for urgent notifications for the operator.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- String - The string that is needed for the quality marker.
- Marker - quality marker.

➢ Output:
- Message - the concatenation of a string and a quality marker

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- AddTime (true) - whether or not to add a timestamp to the string.
- AddDate (true) - Add or not add a date stamp to the string.
- NormalColor (ffffff) - the color of the normal text to display.
- EventColor (ffed0f) - text color in case of an event that requires the operator's attention.
- AlarmColor (ff0000) - text color in case of an error or alarm that requires the operator's attention.
- PerfectColor (00ff00) - text color, in case of transition from an error to a normal status.
- devider (Do not add the symbol) - separator:
  1. Do not add the symbol.
  2. Tabulation.
  3. Space.

Note :
The quality marker is a deprecated component, so an Example of its use has not been developed. You can use the database or the Advanced journal to get the highlighting of the event log lines depending on the type of message that arrived.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- String - The value of the string that is required for the quality marker (text).
- Marker - The value of the quality marker (any number).

➢**Custom common properties:**
- Are available by the link below. 154

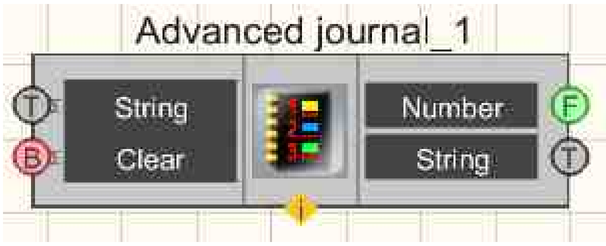
➢**Custom private properties (the default value is shown in parentheses):**

- VARIANT_BOOL AddTime - Set whether or not to add a timestamp to the string:
  true - add timestamp to string;
  false - do not add a timestamp to the string.
- VARIANT_BOOL AddDate - Set whether or not to add a date stamp to a string:
  true - add a date stamp to the string;
  false - do not add a date stamp to the string.
- LONG NormalColor - Set the normal text color for output (number).
- LONG EventColor - Set the color of the text in case of an event that requires the attention of the operator (number).
- LONG AlarmColor - Set the color of the text in case of an error or alarm that requires the attention of the operator (number).
- LONG PerfectColor - Set the color of the text, in case of transition from an error to a normal status (number).
- BSTR devider - Set the separator: Do not add the symbol; Tabulation; Space (string).

26.5.Advanced journal

The component "**Advanced journal**" is used for events recording and representation as a text note in the log.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- String - a formatted text string.
- Clear - the log is being cleared.

➢ Output:

- Number - the number of the selected line.
- String - the selected string.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Columns - set the table header.
- Widths - set the width of the columns.
- Delimiter - set the separator symbol, with which it is possible to format the table (set column names and their width).
- autoincrement(No) - autoincrement of the first column:

1. Yes - in this mode, the first column is filled with numbers automatically, increasing each subsequent column by one.
2. No - auto-increment is disabled in this mode.

- SpecialKey - set a keyword, when found, the entire line will be highlighted in one color.
- SpecialColor (ffff00) - select the color that will be used to highlight the line.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**
- String - the value of a string that contains a formatted text string (text).

➤**Custom common properties:**
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

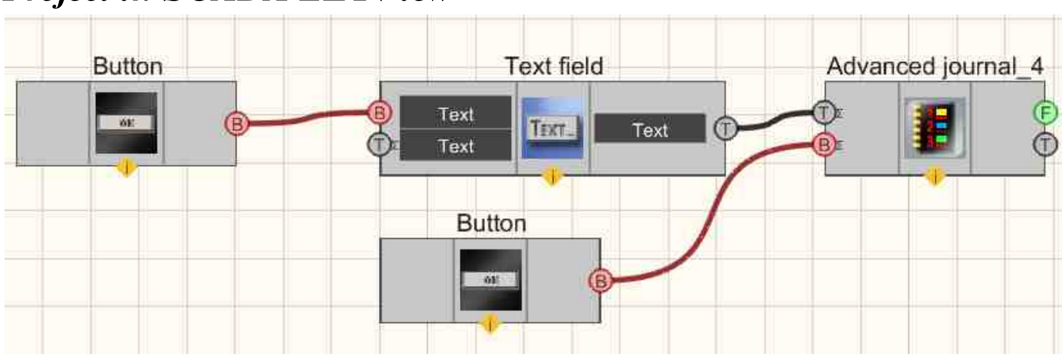
- BSTR Columns - Set table header (string).
- BSTR Widths - Set the width of the columns (string).
- BSTR Delimiter - Set the separator character, with which it is possible to format the table (set column names and their width) (string).
- BSTR Autoincrement - Set the autoincrement of the first column:

1. Yes - in this mode, the first column is filled with numbers automatically, increasing each subsequent column by one;
2. No - auto-increment is disabled in this mode.

- BSTR SpecialKey - Set a keyword, when found, the entire string will be highlighted in one color (string).
- LONG SpecialColor - Set the highlight color for keywords (number).

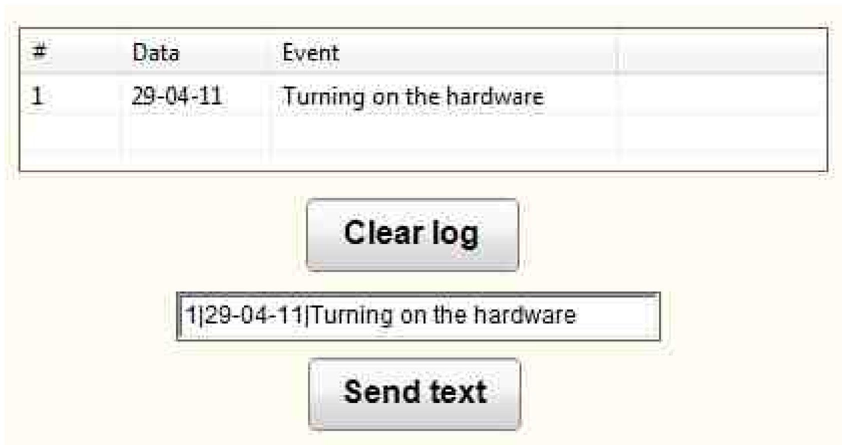
Example No. 1

Project in SCADA ZETView



This project shows the operation of the Advanced journal 1046 component allows you to create a structural record of incoming data in the form of a table. A Non-fixed button 694 connected to the Text field 1100 sends a sync pulse, according to which all text information from the component is transferred to the Advanced journal 1046. The remaining Non-fixed button 694 allows you to clear the contents of the entire table.

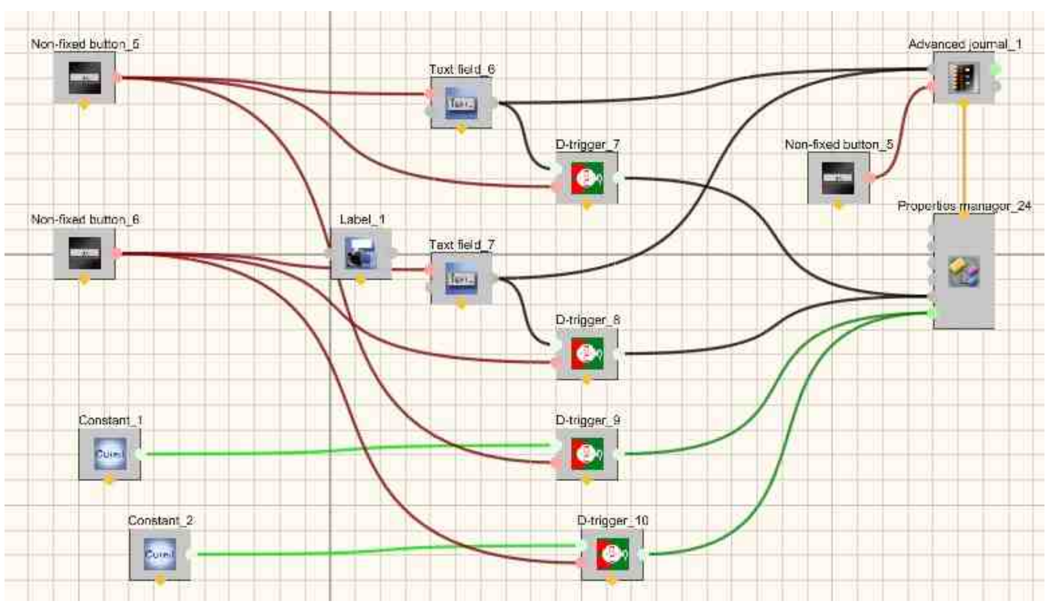
Project operation result



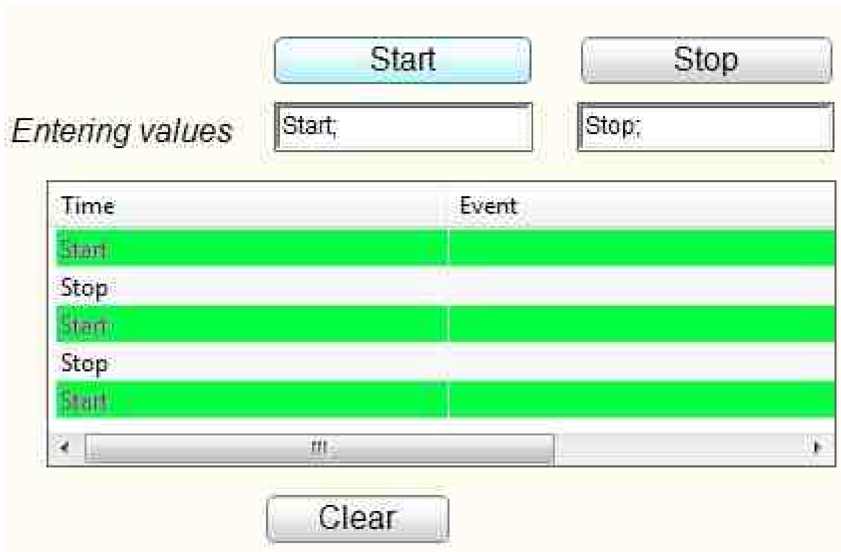
View the example in ZetView

Example No. 2

Project in SCADA ZETView



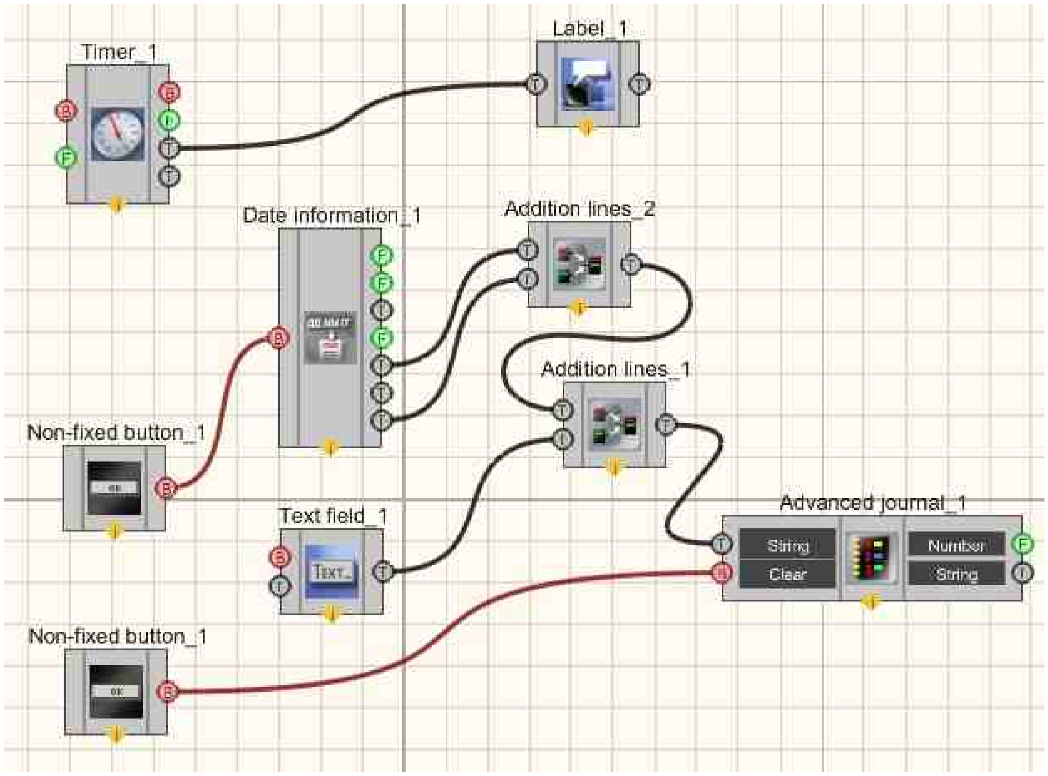
Project operation result



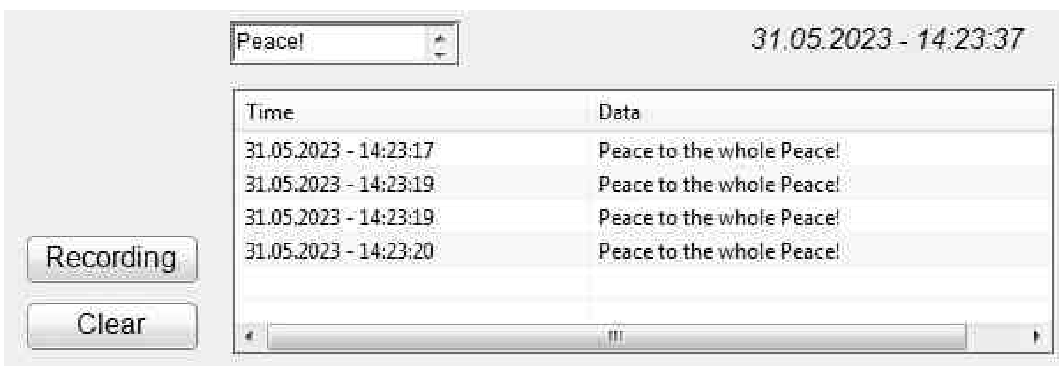
View the example in ZetView

Example No. 3

Project in SCADA ZETView



Project operation results



View the example in ZetView

26.6.Message to the operator

The component "**Message to the operator**" is used to display urgent messages to the operator in the form of a special form that appears on the screen and attracts attention.
The component "**Message to the operator**" urgent messages for the operator.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Impulse - a impulse is applied to this contact, causing the appearance of a special form of message for the operator.

➢ Output:
- Recognize - output channel from which text information is received when the message is recognized by the operator.
- Explanation - output channel from which text information is received when explaining to the note.
- Status - output channel from which information about the status of the current message is received.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- CautionTime (1500) - time to attract attention by the message in milliseconds, ms.
- WindowColor (ff0000) - set the window color.
- String (-.-) - content of the message string.
- HeadHeight (12) - heading text height.

- TextHeight (12) - height of the main text.
- AllowSound (true) - set the enable/disable the sound of the message.
- ButtonCaption (Recognize) - Set the button caption.
- SeparateWindow (false) - display the message in a separate window.
- Reason - the explanation when closing the notification window.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- LONG CautionTime - Set the time to attract attention to the message , ms (number).
- LONG WindowColor - Set the window color (number).
- BSTR String - Set the content of the message string (string).
- LONG HeadHeight - Set the height of the heading text (number).
- LONG TextHeight - Set the height of the main text (number).
- VARIANT_BOOL AllowSound - Set permission for sound alarm:
  true - permission is enabled for sound signaling;
  false - permission is disabled for sound signaling.
- BSTR ButtonCaption - Set button caption (string).
- VARIANT_BOOL SeparateWindow - - Set the permission to display a message in a separate window:
  true - permission is enabled to display a message in a separate window;
  false - the permission is disabled to display a message in a separate window.
- BSTR Reason - Set the explanation when the notification window closes (string).

➢Methods:

- void showAlarm(void) - Throws an alarm.
- void hideAlarm(void) - Hides the alarm.

Example

Project in SCADA ZETView

This project shows the operation of the Message to the operator 1052 component displays a special information letter to the operator. The Non-fixed button 694 is needed to launch this special message. The Label 1087 is used to output a more meaningful letter after the Message to the operator 1052 is recognize.

Project operation result

View the example in ZetView

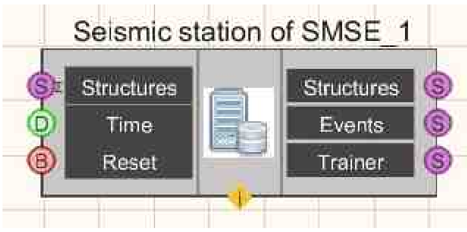
Глава 27.SMSE (system of monitoring of seismic events)

27.1.Allocation of seismic events from the group

The component "**Allocation of seismic events from the group**" is designed to isolate seismic events from a group for operation as part of the SCWS.

The component is used as part of the SMSE (see the documentation on the SMSE located on the supplied disk with setup, the distributions are updated documentation can be downloaded https://file.zetlab.com/Document/.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Structures - input channel with structures from sensors.
- Time - forced start time (DATE).
- Reset - restart the component.

➤ Output:
- Structures - the output channel from which every second structures from the component come.
- Events - the output channel from which event structures come from the component.
- Trainer - the output channel, with data for the trainer.

Custom common properties:

➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

// properties of the component itself
- StationID (1) - The identifier of the seismic station.
- StationName (Station 1) - The name of the seismic station.
- StationLatitude (55.59) - The latitude of the seismic station.
- StationLongitude (37.13) - The longitude of the seismic station.
- StationAltitude (150.0) - The seismic station height.
- StationMileage (0.0) - The seismic station kilometer.
- NeedCopySecStrToLogger (false) - Copy every second structures to the repository.
- NeedDebug (false) - the issuance of Debug information.

- OperationMode (Detection) - the seismic station operation mode: Detection; Modeling; Checking sensors; Disabled.

// properties of the Detector,
- VelocityPhaseP (6.8) - Velocity of propagation of P-phase.
- VelocityRatio (1.74) - Ratio of the velocities of propagation of P - and S-phases.
- CoefficientA (2.0) - Multiplicative coefficient of the magnitude formula.
- CoefficientB (2.3) - Additive coefficient of the magnitude formula.
- CoefficientC (0.0) - Correction factor for the magnitude formula.
- ThresholdAccel (0.03) - Threshold value for classifying an event into strong and weak.
- TimeLiveSecStructure (10.0) - Lifetime of every second structures, s.

// Detection
- Dtct_Type (Event) - Detection. Detector type: Event; By external launch.
- Dtct_ThresoldNumberSens (49.0) - Detection. Threshold number of checking sensors, percent.
- Dtct_TimeLiveEvent (600.0) - Detection. Event lifetime, s.
- Dtct_DurationTimeWindow (10.0) - Detection. The duration of the detection time window, s.
-

// Classification.
- Algr_Accel_OnOff (true)) - Classification. Algorithm for analysis of acceleration peak values. ON.
- Algr_Accel_Threshold (10.0) - Classification. Threshold value of peak acceleration ratios.
- Algr_Velocity_OnOff (true) - Classification. Algorithm for analyzing peak speed values. ON.
- Algr_Velocity_Threshold (10.0) - Classification. Threshold value of peak rate ratios.
- Algr_Motion_OnOff (true) - Classification. Algorithm for analyzing peak displacement values. ON.
- Algr_Motion_Threshold (10.0) - Classification. Threshold value of ratios of displacement peaks.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Structures - input channel with structures from sensors.
- Time - forced start time (DATE).
- Reset - restart the component.

➢ Output:
- Structures - the output channel from which every second structures from the component come.
- Events - the output channel from which event structures come from the component.
- Trainer - the output channel, with data for the trainer.

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- LONG StationID (1) - set the identifier of the seismic station.
- BSTR StationName (Station 1) - set the name of the seismic station (string).
- DOUBLE StationLatitude (55.59) - set the latitude of the seismic station.
- DOUBLE StationLongitude (37.13) - set the seismic station longitude.
- DOUBLE StationAltitude (150.0) - set the seismic station altitude.
- DOUBLE StationMileage (0.0) - set the seismic station kilometer.
- VARIANT_BOOL NeedCopySecStrToLogger (false) - Copy every second structures to the repository .
- VARIANT_BOOL NeedDebug (false) - Return Debug information.
- BSTR OperationMode (Detection) - set the seismic station operation mode: Detection; Modeling; Checking sensors; Disabled (string).

- FLOAT VelocityPhaseP (6.8) - set the P-phase propagation velocity.
- FLOAT VelocityRatio (1.74) - set the ratio of the propagation velocities of the P- and S-phases.
- FLOAT CoefficientA (2.0) - set the multiplicative coefficient of the magnitude formula.
- FLOAT CoefficientB (2.3) - set the additive coefficient of the magnitude formula.
- FLOAT CoefficientC (0.0) - set the correction factor for the magnitude formula.
- FLOAT ThresholdAccel (0.03) - set the threshold value for classifying an event into strong and weak.
- FLOAT TimeLiveSecStructure (10.0) - set the lifetime of every second structures, s.

- BSTR Dtct_Type(Event) - Detection. set the detector type: Event; By external launch (string).
- FLOAT Dtct_ThresoldNumberSens (49.0) - Detection. set the threshold number of checking sensors, percent.
- FLOAT Dtct_TimeLiveEvent (600.0) - Detection. set the event lifetime, s.

- FLOAT Dtct_DurationTimeWindow (10.0) - Detection. set the duration of the detection time window, s.

- VARIANT_BOOL Algr_Accel_OnOff (true) - Classification. Algorithm for analysis of acceleration peak values. ON.
- FLOAT Algr_Accel_Threshold (10.0) - Classification. Threshold value of peak acceleration ratios.
- VARIANT_BOOL Algr_Velocity_OnOff (true) - Classification. Algorithm for analyzing peak speed values. ON.
- FLOAT Algr_Velocity_Threshold (10.0) - Classification. Threshold value of the ratio of peak speed values.
- VARIANT_BOOL Algr_Motion_OnOff (true) - Classification. Algorithm for analyzing peak displacement values. ON.
- FLOAT Algr_Motion_Threshold (10.0) - Classification. Threshold value of ratios of displacement peaks.

27.2.Damping rate

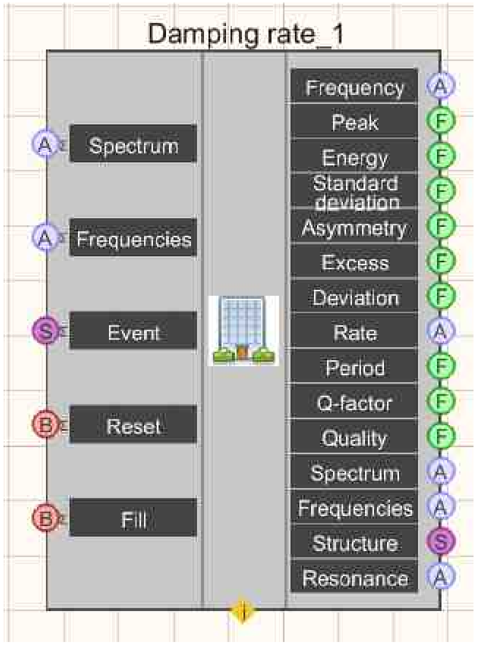
The component "**Damping rate**" is designed to calculate the damping factor in the composition of the SMSE.
The component is used as part of the SMSE (see the documentation on the SMSE located on the supplied disk with setup, the distributions are updated documentation can be downloaded https://file.zetlab.com/Document/.

The component "Decrement of damping" was added to the ZETVIEW SCADA system, designed to measure the frequency and decrement of natural oscillations. On the basis of this component, the project "Determining the fundamental tone and the logarithmic decrement of the fundamental tone of the building's natural oscillations" was developed in accordance with GOST R 54859-2011.

Project window with current calculation results.

With the help of the project, it is possible to process the records of seismic records ZET7173 and ZET048, which were recorded offline, and save the report in the required form.
Also, when the above-mentioned records are connected to the LAN, the project can continuously monitor the parameters of building vibrations: frequency and decrement, as well as the errors in their calculation. The obtained monitoring results will be recorded in the event logging and display system.
On the basis of the "Decrement of attenuation" component, it is possible to build a system of continuous monitoring and control of buildings and structures.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Spectrum - Spectrum array input channel.
- Frequencies - the input channel of an array of frequency bands.
- Event - the input channel of the structure with seismic events.
- Reset - input channel for resetting data in the array drive.
- Fill - input channel for filling the accumulator with the latest array.

➢ Output:

- Frequency - array of average weighted peak frequencies.
- Peak - an array of peak values.
- Energy - an array of peak energy values at level 0.7.
- Standard deviation - An array of standard deviations.
- Asymmetry - an array of peak asymmetry values.
- Excess - an array of peak kurtosis values.
- Deviation - an array of signal-to-noise ratio values.
- Decrement - an array of logarithmic damping decrement values.
- Period - an array of oscillation period values.
- Q - factor - an array of Q - factor values (according to the width of the resonance).
- Quality - the size of the sample used in the calculations.
- Spectrum - an array of the average power spectrum.

- Frequencies - an array of frequencies for the axis of the spectrum graphic.
- Structure - output structure with measured parameters.
- Resonance - Resonance graphic with calculated parameters.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- MinFrequency (0.0) - Minimum natural frequency, Hz.
- MaxFrequency (0.0) - Maximum natural frequency, Hz.
- ThresholdCoefficient (0.7) - Threshold coefficient for selecting points for calculation..
- WriteLog (False) - Recording outgoing structures to a log file.
- AccumulatorSize (1) - The size of the array accumulator for averaging.
- DetectorThresold (3.0) - STA/LTA detector threshold for excluding data.
- IgnoreNumber (0) - The number of inputs to ignore after the event occurs.
- UseEMA (false) - Use exponential moving average instead of linear accumulator.
- CoefficientEMA (1) - EMA coefficient.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Spectrum - Spectrum array input channel.
- Frequencies - the input channel of an array of frequency bands.
- Event - the input channel of the structure with seismic events.
- Reset - input channel for resetting data in the array drive.
- Fill - input channel for filling the accumulator with the latest array.

➢ Output:

- Frequency - array of average weighted peak frequencies.
- Peak - an array of peak values.
- Energy - an array of peak energy values at level 0.7.
- Standard Deviation - An array of standard deviations.

- Asymmetry - an array of peak asymmetry values.
- Excess - an array of peak kurtosis values.
- Deviation - an array of signal-to-noise ratio values.
- Decrement - an array of logarithmic damping decrement values.
- Period - an array of oscillation period values.
- Q - factor - an array of Q - factor values (according to the width of the resonance).
- Quality - the size of the sample used in the calculations.
- Spectrum - an array of the average power spectrum.
- Frequencies - an array of frequencies for the axis of the spectrum graphic.
- Structure - output structure with measured parameters.
- Resonance - Resonance graphic with calculated parameters.

➢**Custom common properties:**
- Are available by the link below. 154

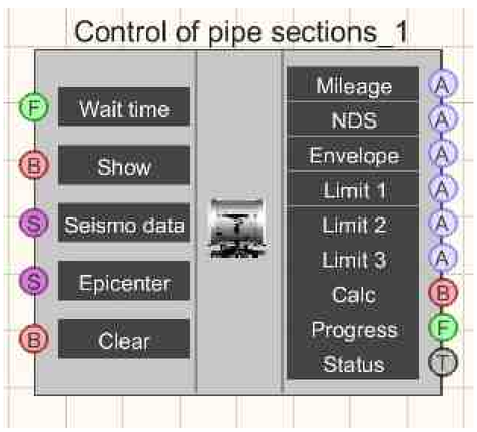
➢**Custom private properties (the default value is shown in parentheses):**
- FLOAT MinFrequency (0.0) - Minimum natural frequency, Hz.
- FLOAT MaxFrequency (0.0) - Maximum natural frequency, Hz.
- FLOAT ThresholdCoefficient (0.7) - Threshold coefficient for selecting points for calculation.
- VARIANT_BOOL WriteLog (False) - Recording outgoing structures to log file Disabled
- LONG AccumulatorSize (1) - The size of the array accumulator for averaging.
- FLOAT DetectorThresold (3.0) - STA/LTA detector threshold value for excluding data.
- LONG IgnoreNumber (0) - Number of inputs to ignore after the event occurs.VARIANT_BOOL
- VARIANT_BOOL UseEMA (false) - Use exponential moving average instead of linear accumulator
- FLOAT CoefficientEMA (1) - EMA coefficient.

27.3.Control of pipe sections

The component "**Control of pipe sections**" is intended for control of pipe sections as a part of SMSE.
The component is used as part of the SMSE (see the documentation on the SMSE located on the supplied disk with setup, the distributions are updated documentation can be downloaded https://file.zetlab.com/Document/.

The following are other uses of the component:
ZETLAB facility status monitoring system
Estimation of the parameters of the stress-strain status of the object using specialized software "Control of pipe sections"
System for monitoring seismic impacts and monitoring (assessment) of the integrity of structures
The main components, characteristics and capabilities of the ZET 048 seismic control system.
Digital processing in ZETLAB for identification of seismic signal parameters
Methods of non-destructive testing. Assessment of the technical condition of the building by means of the hardware-software complex ZETLAB.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Wait time - input channel of waiting time for the next event.
- Show - show the program window .
- Seismo data - data about the seismic station.
- Epicenter - epicenter data.
- Clear - clear seismic station data.

➢ Output:
- Mileage - array of pipeline mileage data.
- NDS - pipe section stress array.
- Envelope - the envelope graphic of the tension of the pipe sections.
- Limit 1 - plastic strain limit array.
- Limit 2 - an array of yield strength.
- Limit 3 - array of irreversible deformation limit.
- Calc - end of SSS calculations for the stress-strain state.

- Progress - data calculation progress in percent.
- Status - the status of the current work of the program.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- SaveProtocol (True) - Save calculation results to disk.
- ShowProtocol (True) - Show calculation results after saving.
- AccelThreshold (0.03) - Threshold of acceleration, m/sI.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Wait time - input channel of waiting time for the next event.
- Show - show the program window .
- Seismo data - data about the seismic station.
- Epicenter - epicenter data.
- Clear - clear seismic station data.

➢ Output:

- Mileage - array of pipeline mileage data.
- NDS - pipe section stress array.
- Envelope - the envelope graphic of the tension of the pipe sections.
- Limit 1 - plastic strain limit array.
- Limit 2 - an array of yield strength.
- Limit 3 - array of irreversible deformation limit.
- Calc - end of SSS calculations for the stress-strain state.
- Progress - data calculation progress in percent.
- Status - the status of the current work of the program.

➢Custom common properties:

- Are available by the link below. 154

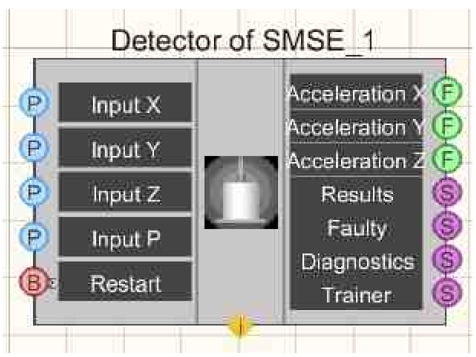
➢Custom private properties (the default value is shown in parentheses):

- VARIANT_BOOL SaveProtocol (True) - Save calculation results to disk.
- VARIANT_BOOL ShowProtocol (True) - Show calculation results after saving.
- FLOAT AccelThreshold (0.03) - Threshold of acceleration, m/sI.

27.4.Detector seismic events

The component "**Detector seismic events**" is designed designed to detect seismic events as part of the SMSE.
The component is used as part of the SMSE (see the documentation on the SMSE located on the supplied disk with setup, the distributions are updated documentation can be downloaded https://file.zetlab.com/Document/.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input X - input channel X, for which the required values are to be measured.
- Input Y - input channel Y, for which you want to measure the required values.
- Input Z - input channel Z, for which you want to measure the required values.
- Input P - input power channel for which you want to measure the required values.
- Restart - sensor restart input channel.

➢ Output:
- Acceleration X - output channel X, from which the peak acceleration in X is received every second.
- Acceleration Y - output channel Y, from which the peak acceleration in Y is received every second.

- Acceleration Z - output channel Z, from which the peak acceleration in Z is received every second.
- Results - the output channel contains the results of the component.
- Faulty - setting up faulty.
- Diagnostics - get diagnostic data.
- Trainer - get data for the trainer.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

// properties of the component itself

- Sens_Activate (true) - Sensor. Working status.
- Sens_ID (1) - Sensor. Identifier.
- Sens_Type (Accelerometer) - Sensor. Type: Accelerometer; Velocimeter; Geophones for seismic device.
- Sens_BandFreq_F1 (1.0) - Sensor. Working frequency band. Lower frequency, Hz.
- Sens_BandFreq_F2 (10.0) - Sensor. Working frequency band. Upper frequency, Hz.
- Sens_bNeedOutputDebug (false) - Sensor. Issuance of Debug information.
- Sens_bNeedDTU (false) - Sensor. Issuing intermediate data to dtu-files.
- Sens_bNeedCopySecStrToLogger (false) - Sensor. Copying every second structures to the repository.

// properties of the Detector,

- Dtct_TypeDetector (STA/LTA) - Detector. Detector type: STA/LTA; Threshold; By timer.
- Dtct_PreHistory (20.0) - Detector. Detector pre-history size, s.
- Dtct_STA_TimeSTA (10.0) - Detector. Duration of the short window of the STA/LTA detector, s.
- Dtct_STA_TimeLTA (190.0) - Detector. Duration of the long window of the STA/LTA detector, s.
- Dtct_STA_Threshold (3.0) - Detector. STA/LTA detector threshold.
- Dtct_STA_MinDurationEvent (20,0) - Detector. Minimum duration of an STA/LTA detector event, s.
- Dtct_STA_Factor (0,4) - Detector. STA/LTA detector factor value.
- Dtct_Thr_sizeEvent (5,0) - Detector. Threshold detector event size, s.
- Dtct_Thr_Threshold (0.03) - Detector. Threshold value of the threshold detector, m/sI.
- Dtct_Tmr_sizeEvent (120.0) - Detector. Timer detector event size, s.

- // calculation parameters
- Calc_AccelThreshold (0.03) - Calculations. Threshold value of peak acceleration in earthquake classification, m/sI.

- Calc_BandFreq_F1 (0.2) - Calculations. Bandwidth when determining displacements. Lower frequency, Hz.
- Calc_BandFreq_F2 (20.0) - Calculations. Bandwidth when determining displacements. Upper frequency, Hz.
- Calc_TimeAver_ZigZag (4000.0) - Calculations. Averaging time in determining the arrival times of P- and S-phases.
- Calc_TimeAver_Txyz (20,0) - Calculations. Averaging time in determining the main periods during the earthquake.

- // channel parameters
- ChanP_valueMin (true) - Channel of the power supply. Minimum value, mV.
- ChanP_valueMax (true) - Channel of the power supply. Maximum value, mV.
- ChanXYZ_valueMax (true) - XYZ channels. The maximum value of the signal.

// polarization algorithm

- Algr_Polar_OnOff (true) - Algorithm of polarization. ON.
- Algr_Polar_TimeAver_Array (0.2) - Algorithm of polarization. The time of averaging the elements of the polarization matrix.
- Algr_Polar_TimeAver_Signal (10.0) - Algorithm of polarization. Algorithm signal averaging time.

// algorithm Correlation

- Algr_Corr_OnOff (true) - Algorithm of correlation. ON.
- Algr_Corr_Threshold (3.0) - Algorithm of correlation. Algorithm threshold.

// amplitude ratio algorithm

- Algr_AmplRation_OnOff (true) - Algorithm of the ratio of amplitudes. ON.
- Algr_AmplRatio_ThresholdAmpl (10.0) - Algorithm of the ratio of amplitudes. Threshold value for amplitudes.

// broadband noise detection algorithm

- Algr_BBN_OnOff (true) - Algorithm of broadband noise. ON.
- Algr_BBN_Threshold (2.0) - Algorithm of broadband noise. The threshold value of the signal-to-noise ratio.

//Click detection algorithm

- Algr_UnipolarClick_OnOff (true) - Algorithm clicks. ON.
- Algr_UnipolarClick_ThresholdRatio (3.0) - Algorithm clicks. The threshold value of the ratio.

// Wavelet analysis algorithm

- Algr_Wavelet_OnOff (true) - Algorithm for wavelets. ON.

- Algr_Wavelet_TypeFDWT (db2) - Algorithm for wavelets. Mother wavelet type in wavelet filtering of signals: ;coif4;coif5.
- Algr_Wavelet_LevelFDWT (4) - Algorithm for wavelets. Decomposition level in wavelet filtering of signals.
- Algr_Wavelet_ThresholdNumZero (17) - Algorithm for wavelets. Permissible number of zero values of the algorithm signal, percent.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Input X - input channel X, for which the required values are to be measured.
- Input Y - input channel Y, for which you want to measure the required values.
- Input Z - input channel Z, for which you want to measure the required values.
- Input P - input power channel for which you want to measure the required values.
- Restart - sensor restart input channel.

➢ Output:
- Acceleration X - output channel X, from which the peak acceleration in X is received every second.
- Acceleration Y - output channel Y, from which the peak acceleration in Y is received every second.
- Acceleration Z - output channel Z, from which the peak acceleration in Z is received every second.
- Results - the output channel contains the results of the component.
- Faulty - setting up faulty.
- Diagnostics - get diagnostic data.
- Trainer - get data for the trainer.

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

// properties of the component itself
- Sens_Activate (true) - Sensor. Working status.
- Sens_ID (1) - Sensor. Identifier.

- Sens_Type (Accelerometer) - Sensor. Type: Accelerometer; Velocimeter; Geophones for seismic device.
- Sens_BandFreq_F1 (1.0) - Sensor. Working frequency band. Lower frequency, Hz.
- Sens_BandFreq_F2 (10.0) - Sensor. Working frequency band. Upper frequency, Hz.
- Sens_bNeedOutputDebug (false) - Sensor. Issuance of Debug information.
- Sens_bNeedDTU (false) - Sensor. Issuing intermediate data to dtu-files.
- Sens_bNeedCopySecStrToLogger (false) - Sensor. Copying every second structures to the repository.

// properties of the Detector,
- Dtct_TypeDetector (STA/LTA) - Detector. Detector type: STA/LTA; Threshold; By timer.
- Dtct_PreHistory (20.0) - Detector. Detector pre-history size, s.
- Dtct_STA_TimeSTA (10.0) - Detector. Duration of the short window of the STA/LTA detector, s.
- Dtct_STA_TimeLTA (190.0) - Detector. Duration of the long window of the STA/LTA detector, s.
- Dtct_STA_Threshold (3.0) - Detector. STA/LTA detector threshold.
- Dtct_STA_MinDurationEvent (20,0) - Detector. Minimum duration of an STA/LTA detector event, s.
- Dtct_STA_Factor (0,4) - Detector. STA/LTA detector factor value.
- Dtct_Thr_sizeEvent (5,0) - Detector. Threshold detector event size, s.
- Dtct_Thr_Threshold (0.03) - Detector. Threshold value of the threshold detector, m/sI.
- Dtct_Tmr_sizeEvent (120.0) - Detector. Timer detector event size, s.

- // calculation parameters
- Calc_AccelThreshold (0.03) - Calculations. Threshold value of peak acceleration in earthquake classification, m/sI.
- Calc_BandFreq_F1 (0.2) - Calculations. Bandwidth when determining displacements. Lower frequency, Hz.
- Calc_BandFreq_F2 (20.0) - Calculations. Bandwidth when determining displacements. Upper frequency, Hz.
- Calc_TimeAver_ZigZag (4000.0) - Calculations. Averaging time in determining the arrival times of P- and S-phases.
- Calc_TimeAver_Txyz (20,0) - Calculations. Averaging time in determining the main periods during the earthquake.

- // channel parameters
- ChanP_valueMin (true) - Channel of the power supply. Minimum value, mV.
- ChanP_valueMax (true) - Channel of the power supply. Maximum value, mV.
- ChanXYZ_valueMax (true) - XYZ channels. The maximum value of the signal.

// polarization algorithm
- Algr_Polar_OnOff (true) - Algorithm of polarization. ON.

- Algr_Polar_TimeAver_Array (0.2) - Algorithm of polarization. The time of averaging the elements of the polarization matrix.
- Algr_Polar_TimeAver_Signal (10.0) - Algorithm of polarization. Algorithm signal averaging time.

// algorithm Correlation
- Algr_Corr_OnOff (true) - Algorithm of correlation. ON.
- Algr_Corr_Threshold (3.0) - Algorithm of correlation. Algorithm threshold.

// amplitude ratio algorithm
- Algr_AmplRation_OnOff (true) - Algorithm of the ratio of amplitudes. ON.
- Algr_AmplRatio_ThresholdAmpl (10.0) - Algorithm of the ratio of amplitudes. Threshold value for amplitudes.

// broadband noise detection algorithm
- Algr_BBN_OnOff (true) - Algorithm of broadband noise. ON.
- Algr_BBN_Threshold (2.0) - Algorithm of broadband noise. The threshold value of the signal-to-noise ratio.

//Click detection algorithm
- Algr_UnipolarClick_OnOff (true) - Algorithm clicks. ON.
- Algr_UnipolarClick_ThresholdRatio (3.0) - Algorithm clicks. The threshold value of the ratio.

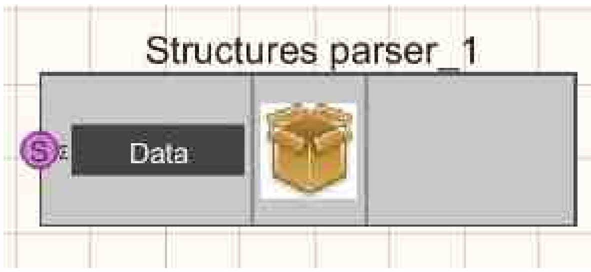
// Wavelet analysis algorithm
- Algr_Wavelet_OnOff (true) - Algorithm for wavelets. ON.
- Algr_Wavelet_TypeFDWT (db2) - Algorithm for wavelets. Mother wavelet type in wavelet filtering of signals: ;coif4;coif5.
- Algr_Wavelet_LevelFDWT (4) - Algorithm for wavelets. Decomposition level in wavelet filtering of signals.
- Algr_Wavelet_ThresholdNumZero (17) - Algorithm for wavelets. Permissible number of zero values of the algorithm signal, percent.

27.5.Structures parser

The component "**Structures parser**" is designed to separate the structures in the composition of the SMSE.

The component is used as part of the SMSE (see the documentation on the SMSE located on the supplied disk with setup, the distributions are updated documentation can be downloaded https://file.zetlab.com/Document/.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Data - input data with a description of the structure.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FileName () - The full name of the file with a description of the structure.
- ConvertTimeFromUTCtoLocal (True) - Convert structure header time from UTC to local time. Turned on.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Input data - input data with a description of the structure

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- BSTR FileName () - Full file name with structure description (string).
- VARIANT_BOOL ConvertTimeFromUTCtoLocal (True) - Convert structure header time from UTC to local time. Turned on.
-

27.6.Region of SMSE

The component "**Region of SMSE**" is designed to work as part of the SMSE.
The component is used as part of the SMSE (see the documentation on the SMSE located on the supplied disk with setup, the distributions are updated documentation can be downloaded https://file.zetlab.com/Document/.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Data - input data from seismic stations for calculating the hypocenter.
- Restart - restart the component.

➢ Output:
- Data - output of AddManagerContact (); with calculated hypocenter.

Custom common properties:

➤Total properties (environment):

- [Are available by the link below.](154) 154

➤**Custom private properties (the default value is shown in parentheses):**

// properties of the component itself

- ShortName (Region MSK) - The short name of the region.
- MinLatitude (50.0) - Latitude of the upper left corner of the region, degr.
- MinLongitude (32.0) - Longitude of the upper left corner of the region, degr.
- MaxLatitude (60.0) - Latitude of the lower right corner of the region, degr.
- MaxLongitude (42.0) - Longitude of the lower right corner of the region, degr.
- EventTimeWindow (300.0) - Station event time window to identify the source of these events, s.
- bNeedOutputDebug (False) - Accuracy of determining the latitude of the earthquake hypocenter, degr.
- PrecisionLatitude (0.01) - The accuracy of determining the longitude of the earthquake hypocenter, degr.
- PrecisionLongitude (0.01) - Accuracy of determining the depth of the earthquake hypocenter, km.
- PrecisionDepth (1) - The detector. Detector pre-history size, s.
- PrecisionT0 (0.1) - Accuracy of determining the start time of an earthquake, s.
- ResidualThreshold (20.0) - Residual threshold value when determining the earthquake epicenter, km.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤Input:

- Data - input data from seismic stations for calculating the hypocenter.
- Restart - restart the component.

➤ Output:

- Data - output of AddManagerContact (); with calculated hypocenter.

➤**Custom common properties:**

- Are available by the link below. 154

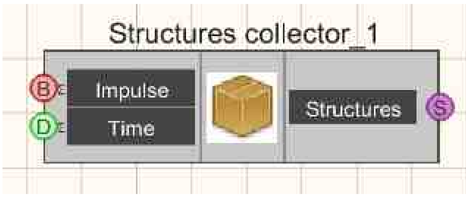
➤**Custom private properties (the default value is shown in parentheses):**

- ShortName (Region MSK) - The short name of the region.
- MinLatitude (50.0) - Latitude of the upper left corner of the region, degr.
- MinLongitude (32.0) - Longitude of the upper left corner of the region, degr.
- MaxLatitude (60.0) - Latitude of the lower right corner of the region, degr.
- MaxLongitude (42.0) - Longitude of the lower right corner of the region, degr.
- EventTimeWindow (300.0) - Station event time window to identify the source of these events, s.
- bNeedOutputDebug (False) - Accuracy of determining the latitude of the earthquake hypocenter, degr.
- PrecisionLatitude (0.01) - The accuracy of determining the longitude of the earthquake hypocenter, degr.
- PrecisionLongitude (0.01) - Accuracy of determining the depth of the earthquake hypocenter, km.
- PrecisionDepth (1) - The detector. Detector pre-history size, s.
- PrecisionT0 (0.1) - Accuracy of determining the start time of an earthquake, s.
- ResidualThreshold (20.0) - Residual threshold value when determining the earthquake epicenter, km.

27.7.Structures collector

The component "**Structures collector**" is designed to assemble various data into a single set of bytes for sending it to another component capable of receiving it. The component is supposed to be used in trainers and testing projects.
The component is used as part of the SMSE (see the documentation on the SMSE located on the supplied disk with setup, the distributions are updated documentation can be downloaded https://file.zetlab.com/Document/.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - designed to send the next portion of data.
- Time - intended for the time for the assembled structure.

➢ Output:

- Structure - used to issue a structure with data.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FileName - path to the file with structure description. The file can be in binary form, as a schema, or as an XML document.

Component operation algorithm:

After specifying a file with a description of the structure, the component creates a set of input contacts. The file can only be specified while editing the project. At runtime, changing the property must be blocked.

When a TRUE logical pulse is applied to the "Impulse" contact, the component must form a structure according to the descriptor loaded from the file and send it from the "Structure" contact. If the descriptor file is not loaded, then do not send anything.

If at the time of the pulse any contacts have not received any data, then the structure should be filled with default values .

Format of data representation in contacts

The structure description file must be created based on the data that needs to be transferred. That is, the component developer first describes the structure of the data that the components will exchange, and then creates a schema file with a description of the structure. After compiling the schema into a binary file, the developer must open the included h-file and compare the created structure with his own for matching data types and sizes. If there are differences, then the scheme needs to be redone. If there are no differences, then the schema file and the binary file must be saved and set in the property of the "Structure Generator" component in order to be able to test the developed components.

Example of contact format:

1. Contact float number (F)
   float value {
   description " Just a simple value "
   }
1. Contact number of type long (F)
   u32 Index {
   description " We must have some index "
   }
1. double number (F)
   double Calculation {
   description " Result of any calculation "
   }
1. Contact array of floats (A)
   float Array [16] {
   description " We need arrays too "
   }
1. Logic contact (B)
   u32Button { _
   description " Boolean type requires for a buttons "
   control checkbox
   }
1. Contact type string (T)
   char String [32] {
   description " Strings is too heavy data "
   }
   The schema file must start with a line (the structure number must be written in hexadecimal)
   struct Example : 0xEA00 {
   and end with a closing curly brace character
   }

Format notes:

1. The Russian language in the descriptor text is not allowed.
2. The name of the variable is displayed in the contact description field at normal scale. The description (description) is displayed when you zoom in.
3. The sizes of the array and the text string set the limit on the incoming data. Extra data will be discarded.
4. In the absence of a separate boolean variable type, boolean contacts are defined by the presence of the field " control checkbox " in the declaration of a variable of type u32.
5. The recommended extension for text schema files is ".zshema1" and for binary files ".zbinary1".
6. Comments in schema files will not be preserved when converting to binary format.

7. ZetSchemaSyntax extension has been created for Visual Studio, which performs syntax highlighting of schema files in the Visual Studio editor window .

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

➢Input:
- Time - intended for the time for the assembled structure.

Setting:

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

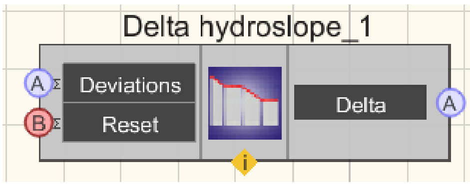
- BSTR FileName - Set the path to the file with the description of the structure (string).

Глава 28.LDS (Leak detection system)

28.1.LDS - hydroslope

The component "**Leak detection system - hydroslope**" is designed to calculate the hydroslope.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Deviations - an array of current values of hydroslope deviations.
- Reset - reset the history of filters.

➢ Output:
- Delta - array of filter results difference.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- PointsNumber - Number of control points.
- CutoffFrequency1 - Cutoff frequency of the first low pass filter.
- CutoffFrequency2 - Cutoff frequency of the second low pass filter.
- InputFrequency - The sample frequency of the input data.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Deviations - an array of current values of hydroslope deviations.

➢ Output:
- Delta - array of filter results difference.

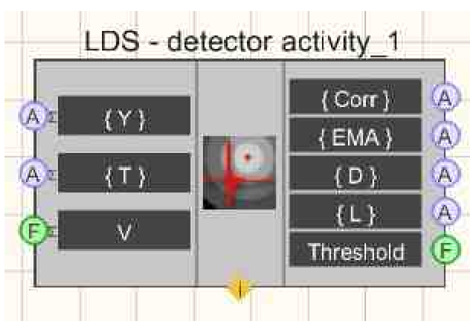
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- LONG PointsNumber (0) - Number of control points.
- FLOAT CutoffFrequency1(0) - Cutoff frequency of the first low pass filter.
- FLOAT CutoffFrequency2(0) - Cutoff frequency of the second low pass filter.
- FLOAT InputFrequency (1) - Sampling frequency of input data.

28.2.LDS - detector activity

The component "**LDS - detector activity**" is designed to determine the activity of the sensor.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- {Y} - input array of correlation.
- {T} - offset input array.
- {V} - speed of sound, m/s.

➢Output
- {Corr} - the specified resampled section of the array is set.

- {EMA}. - set the result of EMA filtering.
- {D} - the result of the detector operation is set.
- {L} - reduced coordinates.
- {Threshold} - detection threshold.

Custom property:

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Reference - Coordinate (m) corresponding to the zero offset of the correlation;
- Velocity - Sound speed, m/s;
- Start - Beginning of the area being viewed, m;
- Finish - End of the viewed area, m;
- Step - Step of resampling, m;
- Exclude - Areas excluded from the calculation in the format n1,k1; n2,k2.. (m). Ascending without crossing;
- DetectorMax - Limiting the maximum value of the detector.
- UseEMA (false) - Use exponential moving average
- EMA(1) - number of measurements averaged by the EMA filter.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢Input:

- {Y} - input array of correlation.
- {T} - offset input array.
- {V} - speed of sound, m/s.

➢Output

- {Corr} - the specified resampled section of the array is set.
- {EMA}. - set the result of EMA filtering.
- {D} - the result of the detector operation is set.

- {L} - reduced coordinates.
- {Threshold} - detection threshold.

➢**Custom common properties:**

- Are available by the link below. 154

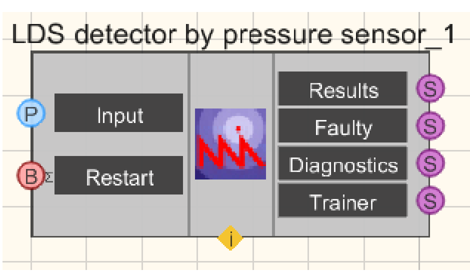
➢**Custom Custom private properties (the default value is shown in parentheses):**

- Reference - Coordinate (m) corresponding to the zero offset of the correlation;
- Velocity - Sound speed, m/s;
- Start - Beginning of the area being viewed, m;
- Finish - End of the viewed area, m;
- Step - Step of resampling, m;
- Exclude - Areas excluded from the calculation in the format n1,k1; n2,k2.. (m). Ascending without crossing;
- DetectorMax - Limiting the maximum value of the detector.
- UseEMA (false) - Use exponential moving average
- EMA(1) - number of measurements averaged by the EMA filter.

28.3.LDS - detector for pressure sensor

The component "**LDS - detector for pressure sensor**" is designed to detect leaks by pressure sensor.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Input - input array with data from the pressure sensor.
- Restart - restarts the sensor.

➤ Output:
- Results - displaying the results of the component.
- Faulty _ - Fault parameters are set.
- Diagnostics - output array with values with diagnostic data.
- Trainer - output array with data for the trainer.

Custom common properties:
➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

// properties of the "Sensor. " component itself
- Sens_Activate - Sensor. Working status.
- Sens_ID - Sensor. Identifier.
- Sens_bNeedOutputDebug - Sensor. Issuance of Debug information.
- Sens_bNeedDTU - Sensor. Issuing intermediate data to dtu-files.
- Sens_bNeedCopySecStrToLogger - Sensor. Copying every second structures to the repository.

// properties of the "Detector" Detector
- Dtct_TypeDetector (STA/LTA) - Detector. Detector type: STA/LTA; Threshold; By timer (string).
- Dtct_PreHistory - Detector. Detector pre-history size, s.
- Dtct_STA_TimeSTA - Detector. Duration of the short window of the STA/LTA detector, s.
- Dtct_STA_TimeLTA - Detector. Duration of the long window of the STA/LTA detector, s.
- Dtct_STA_Threshold - Detector. STA/LTA detector threshold.
- Dtct_STA_MinDurationEvent - Detector. Minimum duration of an STA/LTA detector event, s.
- Dtct_STA_Factor - Detector. STA/LTA detector factor value.

- Sens_Quantity(1) - Number of channels.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Input - input array with data from the pressure sensor.
- Restart - restarts the sensor.

➢ Output:
- Results - displaying the results of the component.
- Faulty _ - Fault parameters are set.
- Diagnostics - output array with values with diagnostic data.
- Trainer - output array with data for the trainer.

➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

// properties of the "Sensor. " component itself
- Sens_Activate - Sensor. Working status.
- Sens_ID - Sensor. Identifier.
- Sens_bNeedOutputDebug - Sensor. Issuance of Debug information.
- Sens_bNeedDTU - Sensor. Issuing intermediate data to dtu-files.
- Sens_bNeedCopySecStrToLogger - Sensor. Copying every second structures to the repository.

// properties of the "Detector" Detector
- Dtct_TypeDetector (STA/LTA) - Detector. Detector type: STA/LTA; Threshold; By timer (string).
- Dtct_PreHistory - Detector. Detector pre-history size, s.
- Dtct_STA_TimeSTA - Detector. Duration of the short window of the STA/LTA detector, s.
- Dtct_STA_TimeLTA - Detector. Duration of the long window of the STA/LTA detector, s.
- Dtct_STA_Threshold - Detector. STA/LTA detector threshold.
- Dtct_STA_MinDurationEvent - Detector. Minimum duration of an STA/LTA detector event, s.
- Dtct_STA_Factor - Detector. STA/LTA detector factor value.

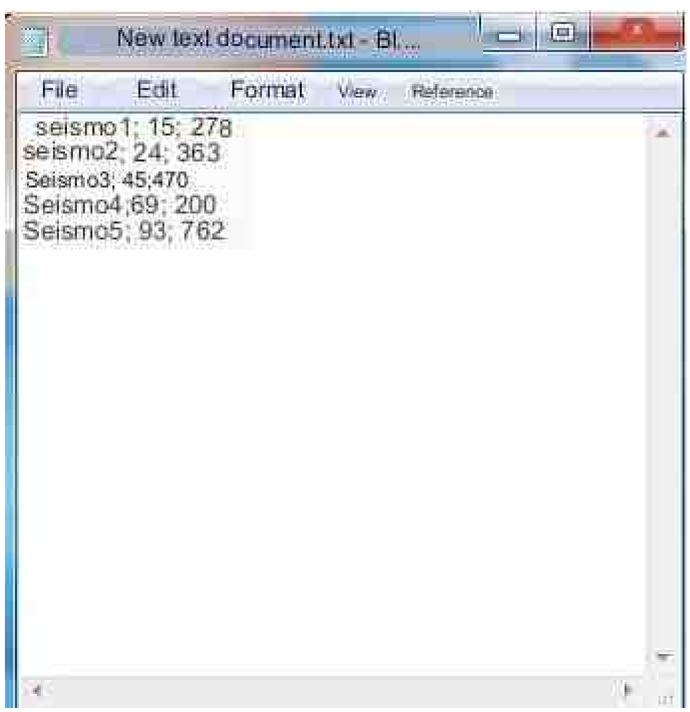
- Sens_Quantity(1) - Number of channels.

28.4.LDS - mode detector

The component "**LDS - mode detector**" is designed to determine what status the system is in now, possible statuses:

1. Stopped transfer
2. Minimum pumping
3. big pumping
4. Damper installed

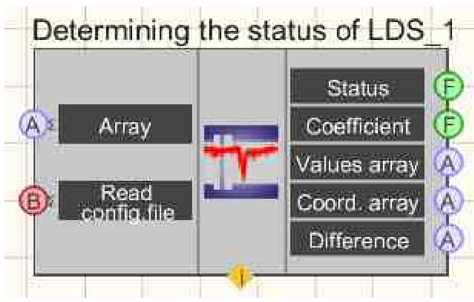
The component must work with a configuration file that will contain information about the channels the component works with. The file must contain the name of the sensor, the coordinate of the sensor (the kilometer on which the sensor is located relative to the beginning of the pipe) and the correction for pressure, all these parameters must be specified separated by a comma and set separately for each sensor. Appearance of the configuration file:



The input of the component should receive an array with data for each sensor. The array size must match the number of sensors specified in the configuration file.

At the output, the component should output the status of the system as an integer: 0 - Stopped pumping, 1 - Minimum pumping, 2 - Large pumping. The component should also return an array with liquid level values for each sensor and an array with the location of each sensor on the pipe.
Also, the component must calculate the correction to pressure on its own and, when closing the program, save it to the configuration file with which it worked.
It should also be possible to set the coefficients by which the change in the status of the system will be determined

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Array - input array with data from sensors.
- Read configuration file - read data from the configuration file.

➢ Output:

- System status - the status of the system.
- Line slope factor - set the line slope coefficient.
- Values array - output array with pressure values for each sensor.
- Coord. array - output array with the coordinates of each sensor.
- Difference - output array of difference between theoretical and practical curves.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- ConfigFileName () - Location of the configuration file with information about the sensors.
- ModeFileName () - Location of the configuration file with information about the modes.
- ResultFileName () - The location of the file in which the received values will be written.
- ResultWriteTime (5,0) - Interval for recording data to the result file, min.

- NonstationaryTime (100.0) - Nonstationary time, s.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

➢Input:
- Array - input array with data from sensors.

➢Custom private properties (the default value is shown in parentheses):

- BSTR ConfigFileName () - Location of the configuration file with information about the sensors.
- BSTR ModeFileName () - Location of the configuration file with mode information.
- BSTR ResultFileName () - The location of the file in which the received values will be written.
- FLOAT ResultWriteTime (5,0) - Interval for recording data to the result file, min.
- FLOAT NonstationaryTime (100.0) - Nonstationary time, s.

Глава 29.Text

"Text" section contains components for operations with textual information. These components are often used for notification, textual data receipt and processing.

The component "Text field" is used for textual data input. It allows the user to enter the text and to send it to the next component. The data is sent to the next component as it is changed or as a control signal is sent to a particular input. E.g., if a button is connected to the input, the component will transfer the data from the line upon button activation.

In addition to that, the "Text" field has tools for textual data processing.

The component "Separator string" allows to divide the string into two or more strings in accordance with the set symbol. The amount of input strings is also specified in settings.

The component "Tag" is used for text string display. In the case if it is necessary to display text array, "Data table" component is used.

"Static text" component is not available in developer mode, but it is visible in operator mode. This component is used for comments. It allows to produce text notifications for the operator.

The component "Search word in the string" allows to find a combination of symbols in corresponding text string.

"Addition string" component is used to combine two strings into a single one.

29.1.Label

The component "**Label**" is used for text string display. In the case if it is necessary to display text array, "Data table" component is used.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Input - text string, label caption.

➢ Output:

- Output - the text string entered into the component arrives, the label inscription.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- TextSize (12) - text size.
- textstyle (FontStyleItalic) - text style
- Transparent (true) - set the enable/disable transparency. Transparent - transparency of the component. When enabled (true), the background color of the component matches the color of the page; when disabled (false), the background color is set by the BackgroundColor property .
- BackgroundColor (000000) - text color.
- TextColor (ffffff) - background color.
- Text (Label text) - the inscription displayed on the display.
- alignment (By align to the left) - type of text alignment:
    1. By align to the left;
    2. In the middle.
    3. By align to right.
- Frame (false) - set the enable/disable the frame around the text.

Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:
➢Input:
- Input - the value to which the text string arrives, the inscription of the label (text).

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set the text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- VARIANT_BOOL Transparent - Set the display status to transparent:
  true - transparency is enabled;
  false - transparency is disabled.

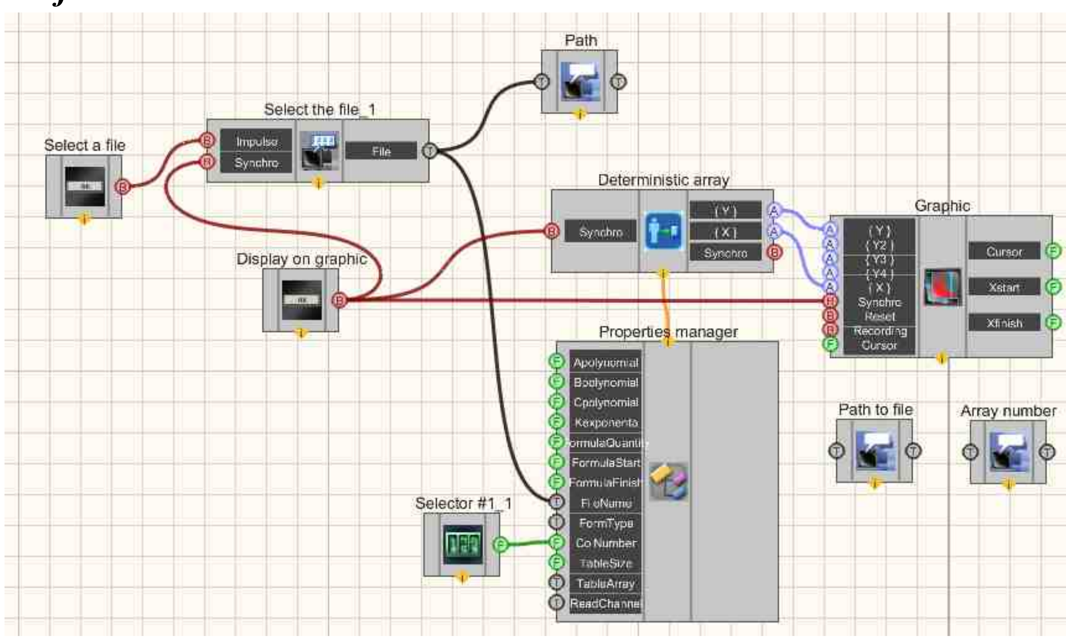
- LONG BackgroundColor - Set the text color (any number).
- LONG TextColor - Set the background color (any number).
- BSTR Text - Set the inscription (string).
- BSTR Alignment - Set the type of text alignment: By align to the left; In the middle; By align to right (string).
- VARIANT_BOOL Frame - set the enable/disable the frame around the text:
  true - framing status is enabled;
  false - framing status is disabled.

➤**Methods:**

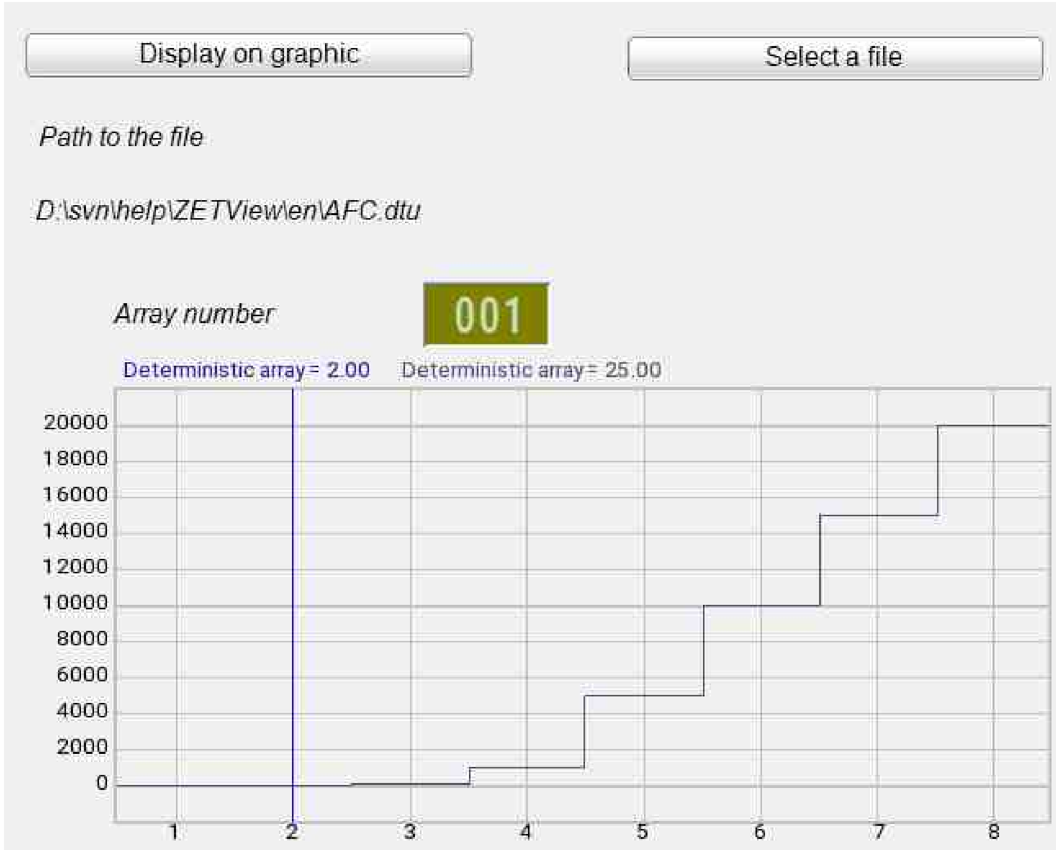
- void SendText(void) - Set the value of sending text.

Example No. 1

Project in SCADA ZETView



Project operation result

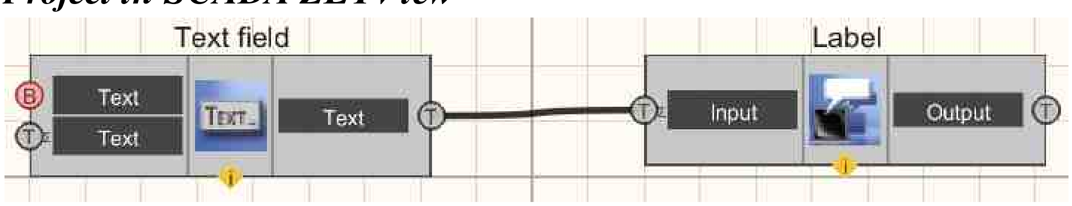


View the example in ZetView

Examples

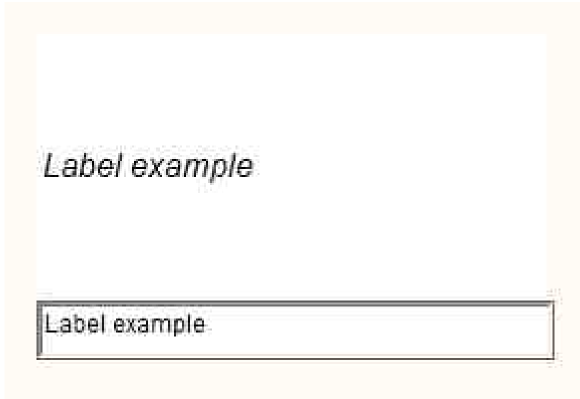
➢Example No. 1 (Using a Label as an output component).

Project in SCADA ZETView



This project shows the operation of the Label[1087] component is used for graphical representation of textual information that comes from the Text field[1100].

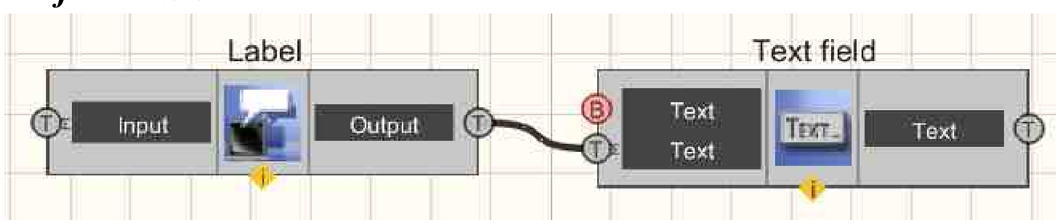
Project operation result



View the example in ZetView

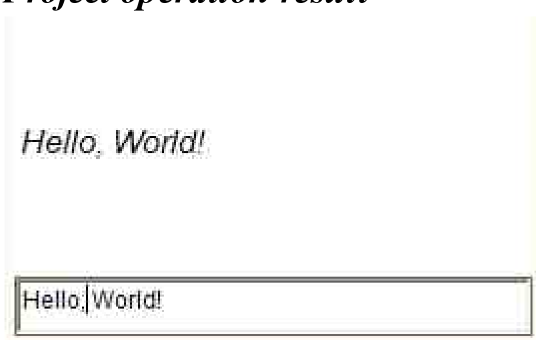
➢Example No. 2 (Using a Label as an output component).

Project in SCADA ZETView



This project shows the operation of the Label[1087] component is used to pass a text string to the Text field[1100].

Project operation result

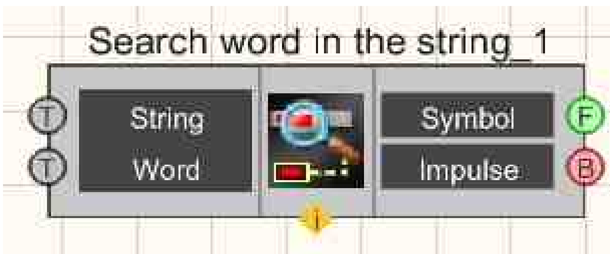


View the example in ZetView

29.2.Search word in the string

The component "**Search word in the string**" allows to find a combination of symbols in corresponding text string.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- String - the string to search for.
- Word - a set of characters (word) to be found in the string.

➢ Output:

- Symbol - receives the symbol number in case of a match.
- Impulse - a high or low logic level is set depending on whether a word is found or not.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- String - a string is connected to this output, in which the search will take place.
- Word - the word to be searched for in the connected string.
- FirstReaction (True) - set the reaction of changing the output channel to changing values at the input of String No. 1:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.
- SecondReaction (True) - set the reaction of changing the output channel to changing values at the input of String No. 2:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.

- reactiontype (Both events) - set the condition for sending a single impulse from the output:
  1. Match - send an impulse if the entered word is found in the line.
  2. Mismatch - send an impulse if the entered word does not exist in the line.
  3. Both events - send an impulse in case of a change in the line channel or in the word channel.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

- Input:
- String - The value of the string to search for (text).
- Word - The value of the character sets (word) to be found in the string (text).

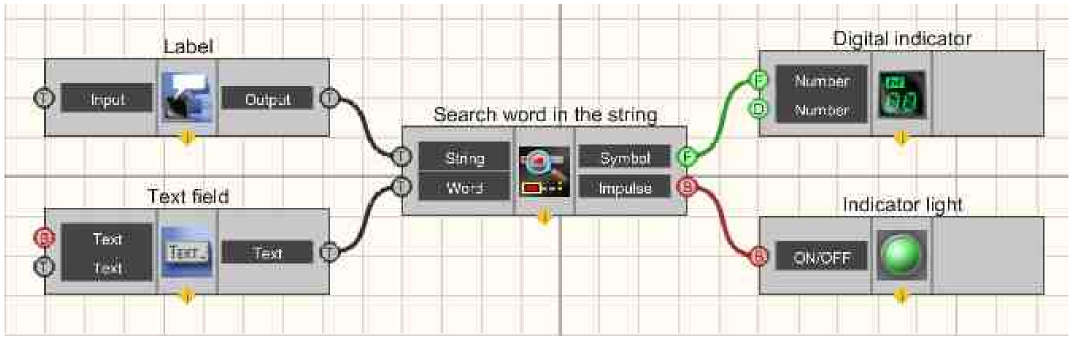
➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

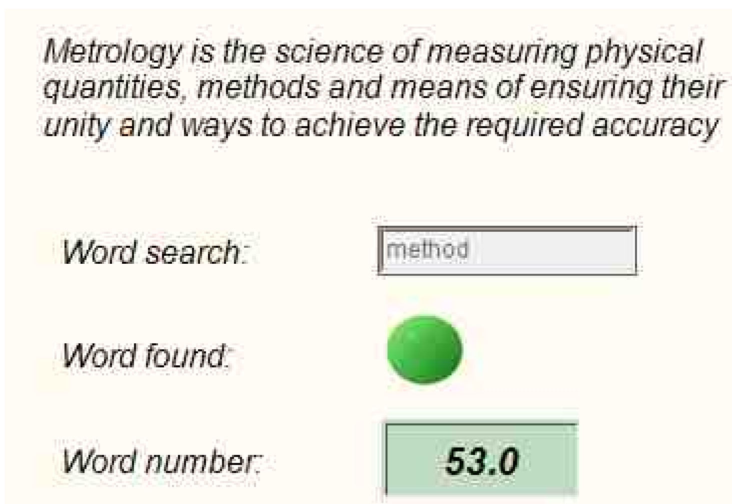
- BSTR String - Set the string to search for a word (string).
- BSTR Word - Set and read the word to search in the string (string).
- VARIANT_BOOL FirstReaction - Set the reaction value to the first contact:
  true - the output value is issued when data arrives at the first contact;
  false - the output value is not issued when data is received on the first contact.
- VARIANT_BOOL SecondReaction - Set the reaction value to the second contact:
  true - the output value is issued when data arrives at the second contact;
- BSTR ReactionType - Set reaction type to (string): Match; Mismatch; Both events.

Example
Project in SCADA ZETView



This project shows the operation of the Search word in the string 1092 component is used to search for a word that is entered in the Text field 1100. If this is found, the Light indicator 669 turns green, and the number of the character in the line is displayed on the Digital Indicator 682.

Project operation result

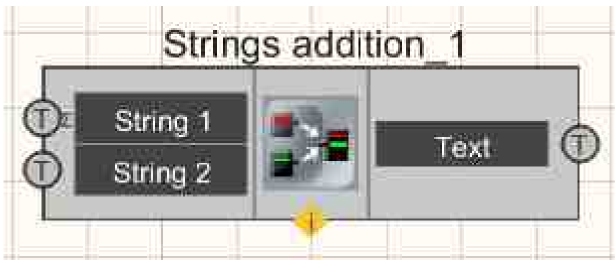


View the example in ZetView

29.3. Strings addition

The component "**Strings addition**" is used to combine two strings into a single one.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- String 1 - text string No. 1.
- String 2 - text string No. 2.

➢ Output:
- Text - the resulting text string (already glued together).

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Text - the current result of the addition.
- FirstString - the first term string.
- SecondString - the second string is a term.
- FirstReaction (True) - set the reaction of changing the output channel to changing values at the input of String No. 1:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.
- SecondReaction (True) - set the reaction of changing the output channel to changing values at the input of String No. 2:
  1. True - when the input signal changes, the output value will change.
  2. False - when the input signal changes, the output value will not change.
- specialdividerAdd - set the one special separator symbol between lines:
  1. Tabulation - adds a horizontal tab between lines.
  2. Space - adds a single blank space between lines.
  3. Line break - adds a second line to the first, but starts it on a new line.
- SpecSymbol (False) - (False) - set the enable or disable special symbols:
  1. True - in this mode, a special symbol is added between the lines, set in the special divider property.

2. False - in this mode, the symbol specified in the Divider property is added between lines.

- Divider - set a separator symbol set by the user.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- String 1 - The value of the line to which the text string 1 (text) arrives.
- String 2 - The value of the line to which the text string 2 (text) arrives.

➢**Custom common properties:**

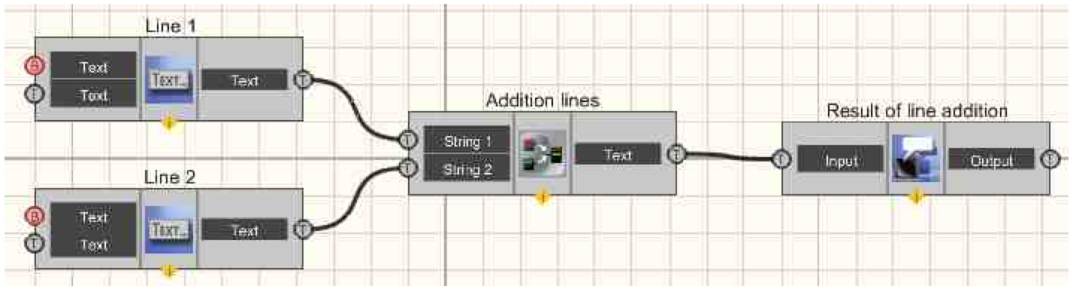
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR Text - Set the current addition result (text string).
- BSTR FirstString - Set the first string - term (text string).
- BSTR SecondString - Set the second string - term (text string).
- VARIANT_BOOL FirstReaction - Set the value of the reaction to change the first string:
  true - the output value is issued when data is received to change the first string;
  false - the output value is not issued when data is received to change first string.
- VARIANT_BOOL SecondReaction - Set the value of the reaction to change the second string:
  true - the output value is issued when data is received to change the second string;
  false - the output value is not issued when data is received to change the second string.
- BSTR SpecialDivider - Set a special separator symbol (string): Tabulation; Space; Line break.
- VARIANT_BOOL SpecSymbol - Set the value to use a special character:
  true - the output value is issued when data is received for the use of a special symbol;
  false - the output value is not issued when data is received for the use of a special symbol.
- BSTR Divider - Set the separator symbol (string).

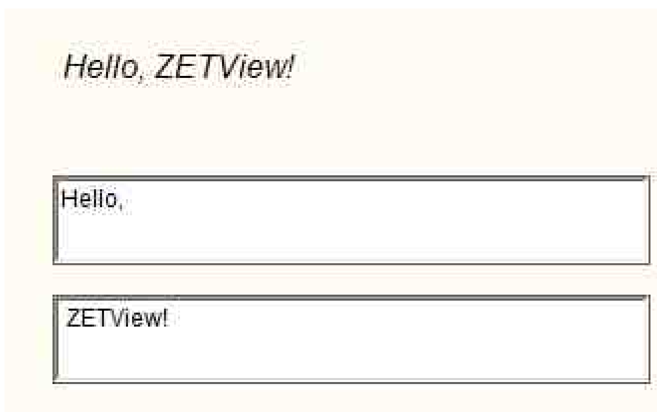
Example

Project in SCADA ZETView



This project shows the operation of the Strings addition[1094] component is used to convert two independent strings into a single one. The Label[1087] is needed to display the resulting string. Text fields[1100] are needed to enter two independent strings, which will undergo a string concatenation (gluing) operation.

Project operation result



View the example in ZetView

29.4.Static text Windows style

The component "**Static text Windows style"** is not available in developer mode, but it is visible in operator mode. This component is used for comments. It allows to produce text notifications for the operator.

Appearance of the component:

Developer interface	Operator interface
Doesn't have	

Setting:

➤Input:

Doesn't have.

➤ Output:

Doesn't have.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

- TextSize (12) - text size.
- textstyle (FontStyleItalic) - text style.
- Transparent (true) - set the enable/disable transparency. Transparent - transparency of the component. When enabled (true), the background color of the component matches the color of the page; when disabled (false), the background color is set by the BackColor property.
- BackColor (808080) - background color.
- TextColor (000000) - text color.
- String (Comment) - the inscription displayed.
- alignment (By align to the left) - type of text alignment:
  1. By align to the left;
  2. In the middle.
  3. By align to right.
- Frame (false) - set the enable/disable the frame around the text.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

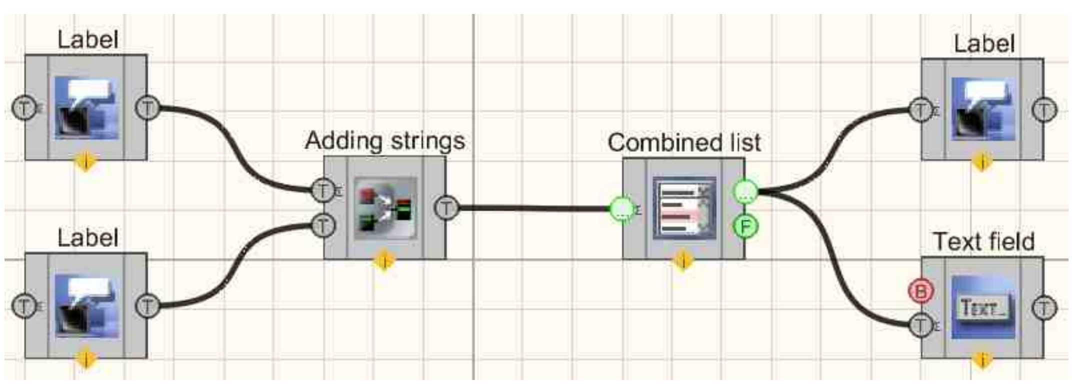
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- SHORT TextSize - Set the text size (any number).
- BSTR TextStyle - Set the text style (string):
  FontStyleBold;
  FontStyleBold;
  FontStyleItalic;
  FontStyleBoldItalic;
  FontStyleUnderline;
  FontStyleStrikeout.
- VARIANT_BOOL Transparent - Set the display status to transparent:
  true - transparency is enabled;
  false - transparency is disabled.
- LONG BackColor - Set the background color (any number).
- LONG TextColor - Set the text color (any number).
- BSTR String - Set the inscription (string).
- BSTR Alignment - Set the type of text alignment: By align to the left; In the middle; By align to right (string).
- VARIANT_BOOL Frame - set the enable/disable the frame around the text:
  true - framing status is enabled;
  false - framing status is disabled.

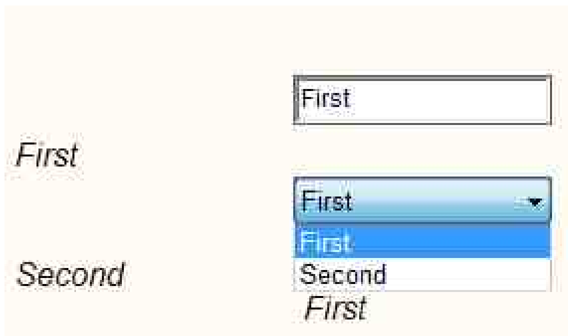
Example

Project in SCADA ZETView



This project shows the operation of the Combined list 1097 component is used to selectively pass text from one of the labels to the Text field 1100. The Adding strings 1094 component is needed to concatenate two channels with text into one common one. It should be noted that in the private properties of Adding strings 1094, the delimiter symbol (in this case) is set, which is necessary to pass concatenated strings to the Combined list1097 as 2 strings.

Project operation result

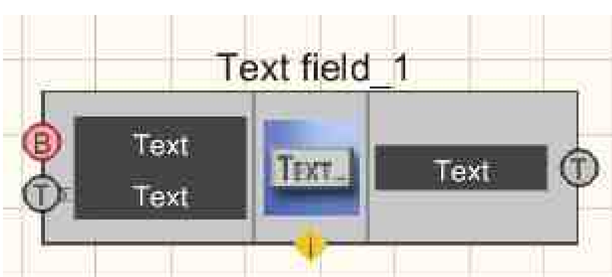
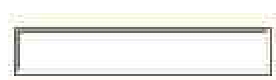


View the example in ZetView

29.5.Text field

The component "**Text field**" is used for textual data input. It allows the user to enter the text and to send it to the next component. The data is sent to the next component as it is changed or as a control signal is sent to a particular input. E.g., if a button is connected to the input, the component will transfer the data from the line upon button activation.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Text - when a sync impulse is applied, text is being transmitted to the output
- Text - a text string that will be displayed by the component.

➢ Output:

- Text - the text string entered into the component is received.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Text - the text inside the field. When the project is running, it is possible to paste from the clipboard into the "Text field" component. Added reaction to Ctrl + V Ctrl + C Ctrl + A Ctrl + X in the component.
- type type (To change) - the text output type:

1. By change - the text output to the display in case of a change in the status of the input.
2. By impulse - the text output to the display in case of input of an impulse.

- Editability (true) - the ability to change the test field.
- FontSize (15) - the font size.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Text - the value of the string that will be displayed by the component (text).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR Text - set the text (text string).
- BSTR Type - set the operating mode: By change; By impulse (string).
- VARIANT_BOOL Editability - Set the editability of the field in operator mode:

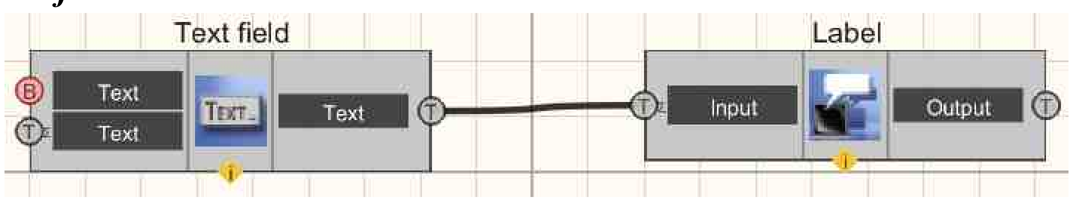
true - the field editability in operator mode is enabled;
false - the field editability in operator mode is disabled.

- LONG FontSize - Set the font size.

Examples

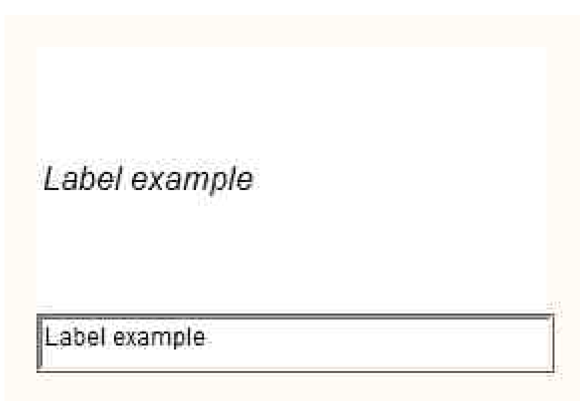
➢Example No. 1 (Using a Label as an output component).

Project in SCADA ZETView



This project shows the operation of the [Label](1100) component is used for graphical representation of textual information that comes from the [Text field](1100).

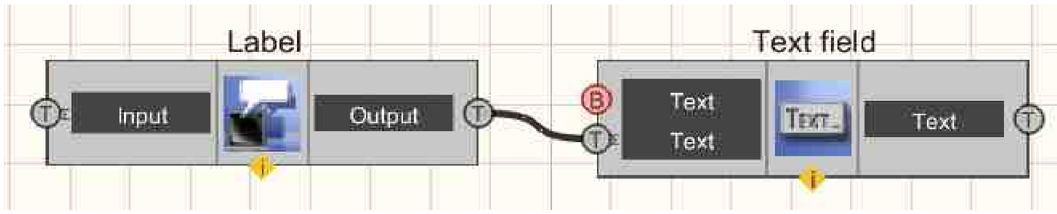
Project operation result



View the example in ZetView

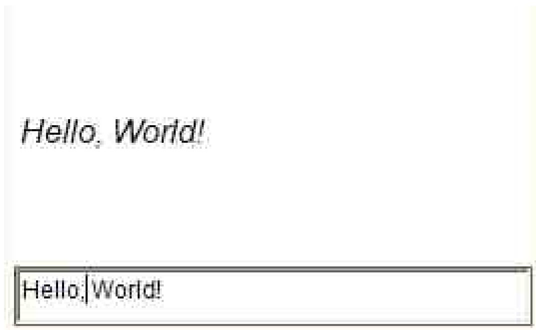
➢Example No. 2 (Using a Label as an output component).

Project in SCADA ZETView



This project shows the operation of the Label 1100 component is used to pass a text string to the Text field 1100.

Project operation result

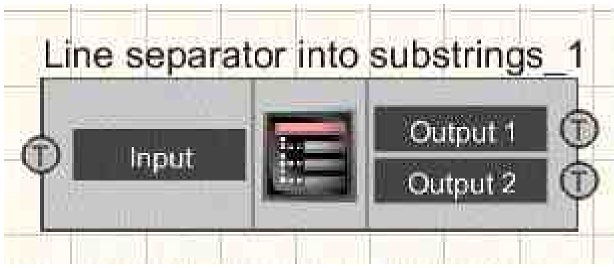


View the example in ZetView

29.6.Line separator

The component "Line separator" allows to divide the string into two or more strings in accordance with the set symbol. The amount of input strings is also specified in settings.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - a text string to be split into multiple strings.

➢ Output:
- Output 1 - Output 49 - output lines. The number depends on the Outputs parameter .

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Delimiter (|) - line separator symbol. Does not appear in split lines. The default symbol is "|"
- Outputs (2) - number of outputs. The default is 2. The maximum number is 49.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Input - the value of the string to be split into multiple (text).

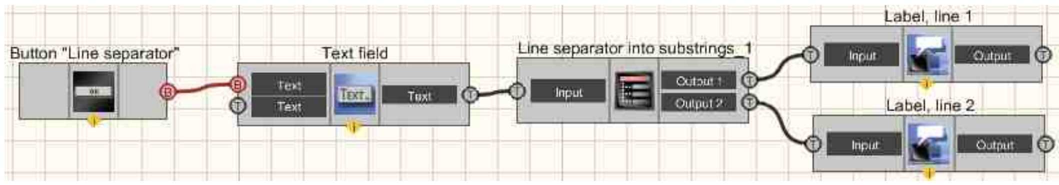
➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR Delimiter - Set the delimiter character (string).
- LONG Outputs - Set the number of outputs (from 0 to 49).

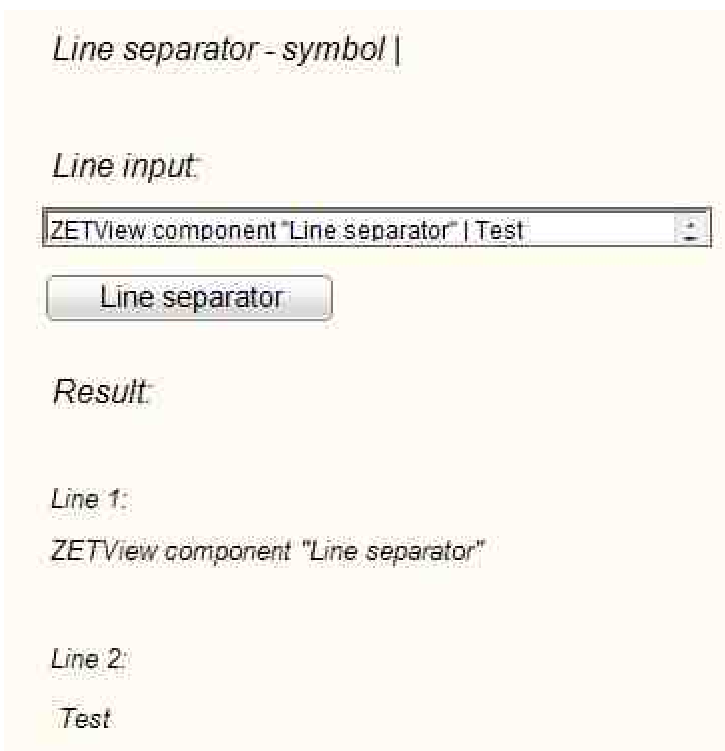
Example

Project in SCADA ZETView



This project shows the operation of the Line separator 1103 component is considered.
The user enters a string into the Text field 1100. After receiving a control impulse from a Non-fixed button 694 called Line separator 1103, the entered string is passed to the Line separator 1103 component, where the lines are split when the "|" symbol is received. The result is displayed on the Label 1087 component.

Project operation result



View the example in ZetView

Глава 30.devices ZET

Components of "Devices" section are used for external devices control (e.g. Narrow-band spectrum analyzer, calibrator), control of digital ports of the connected devices (ADC DAC module ZET 210), as well as to receive data from server channels. The section "Devices" contains the following components:

- **Switching unit relay keys** - The component is used for operation with LAN_connect equipped with integrated ZET 048 and allows to suspend communication for input output channels (the program sends codes with channels).

- **Switching unit** - is used for control of switching unit module, connected to ADC DAC modules outputs. It is used to control relays status connected to commutator bar by means of digital port of the following modules: ZET 220, ZET 230;

- **ZET 0xxx2 and ZET 0xxx4/0xxx8** - is used for measuring the parameters of spectral signals components from various signals, correlational structure of the signals, generation of electrical signals with preset metrological parameters, noise and vibration measurements in 1/3 octave band, thus allowing to use it as vibration and noise meter. It is used for visual representation, data acquisition and signals processing from various sensors: vibration transducers, measuring microphones, hydrophones with ICP integrated amplifiers, etc. The components is necessary for control, measuring and diagnostic equipment used for various measurements and tests, for long-term processes monitoring (both stationary and dynamic);
- **ZET 110** - the component is used for measuring various signals parameters, noise and vibration measurements, representation of the measured parameters, constant recording of signals to non-volatile integrated memory (volume – up to 2 Gb);
- **ZET 210** - is used for measuring signals parameters in wide frequency range (with sampling frequency up to 400 kHz), received from various primary transducers. Digital (DB-15) and analog (DB-25) inputs can be used for the purpose of various mechanisms control;
- **ZET 220** - the component is used for measuring signals parameters with high precision and in wide frequency range (the signals are received from various primary transducers: thermal resistances, thermocouples, sensors with universal current output 4… 20 mA, accelerometers BC 201/202);
- **ZET 230** - the component is used for measuring signals parameters with high precision and in wide frequency range (the signals are received from various primary transducers: accelerometers BC 201, BC 202, microphones BC 501;
- **ZET 240** - is used for measuring and registering of signals from seismic receivers, the component also operates as a measuring instrument for electrical signals parameters;

- **ZET 440** - The component is used for conversion of electrical transducer’s high-impedance charge signal (from vibration sensor, accelerometer or hydrophone) into current signal and conversion of ICP sensors signals into voltage signal;

  The component allows to connect accelerometers with charge output or ICP to FFT Spectrum Analysis and ADC / DAC modules.

  Thus, it is possible to connect hydrophones BC 311 to Narrow-band spectrum analyzer ZET 017. The amplifier ZET 440 is used in sensors control system in order to connect reference accelerometer AP10 to Narrow-band spectrum analyzer ZET 017;

- **ZET 7000 channel reading** - The component is used for reading the data from sensors of 7xxx series. This component allows to calculate the current measured value of the signal by device’s serial number, channel number and device type. Serial number and device type are obtained from "Device manager" program. Serial number is shown in 16-digit format, without particular type indication (i.e., without 0x index);

- **Indicator ZET7x78** - It is intended for connection to measuring networks on the basis of digital sensors with CAN interface and serves for indication of values of measured parameters. Also, the module is used to monitor the voltage on the power bus in the network using the "Oscilloscope" program from the ZETLAB package or on the module screen.

- **Relay control module ZET7161** - the component is used for producing "dry contact" signal, it has 3 input relays, activated upon receipt of signal from CAN interface;

- **Digital port control module ZET7x60** – the component digital port / logical analyzer with RS-485 or CAN interface. It is used for the control of the connected device. The component is used in control and automation systems.

- **Synchronous generator ZET7090** - ZET 7090 Generator-485 - analog two-channel synchronous generator with RS-485 interface. It is used a get sine, impulse or saw-shape signals. The component has 2 synchronous channels and can be used independently or in control and automation systems;

- **Change of address ZET7xxx** – the component allows to change addresses of Zet7xxx sensors;

- **List of ZET7xxx devices** – It is intended for reading the list of devices of ZET7xxx modules;

- **TRC thermometer ZET7x20** - the component is used for temperature measurements in liquid, gaseous and granular environments that are non-aggressive to the TRC package. Application spheres: heat energy, chemical industry, metallurgy. The component is used together with intelligent temperature sensors with RS-485 interface;

- **TRC thermometer ZET7x21**- the component is used for temperature measurements. The component is used for temperature measurements and control in liquid, solid, gaseous and granular environments in various spheres of industry. It is used together with intelligent temperature sensors with RS-485 interface;

- **Management of network devices ZET7x76** - the component is used for connection of measuring networks based on digital sensors with RS-485 or CAN interface to PC via Ethernet network.

- **Relay control ZET7062** - is used for relay control. Settings: by default. The data is saved to flash-memory. The component is used to set initial configuration of the relay;

- **Device ZET** - the component is used for connection and setting parameters of ZET devices;

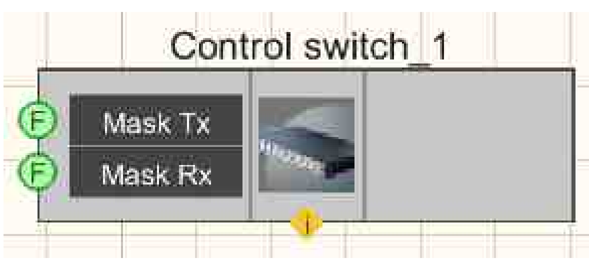
- **Digital generator ZET7060g** - the component is used for generation of impulse signals with frequency range up to 100 mHz;

- **Encoder ZET7060E** - the component is used for angular and linear displacement, shaft rotation velocity, acceleration and position measurements. The data is transferred by RS-485 interface (Modbus protocol);

30.1.Control switch

The component "**Control switch**" is used for operation with LAN_connect equipped with integrated ZET 048 and allows to suspend communication for input output channels (the program sends codes with channels).

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Tx mask - value of the RX relay state mask.
- Rx mask - value of the TX relay state mask.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- MaskTX (1048575) - TX bit mask.
- MaskRX (1048575) - RX bit mask.
- DeviceNumb (0) - the serial number of the device.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Mask - RX relay status mask value (from 0 to 1048575).
- Mask - TX relay status mask value (from 0 to 1048575).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

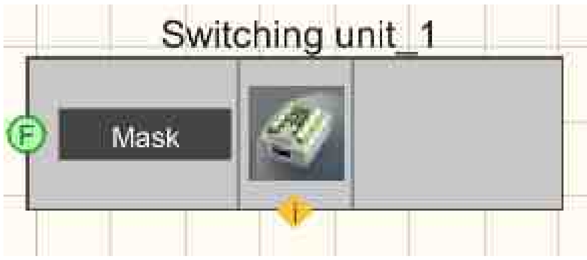
➢

- LONG MaskTX - Set the TX bitmask (number).
- LONG MaskRX - Set the RX bitmask (number).
- LONG DeviceNumb - Set the serial number of the device (number).

30.2.Switching unit

The component "**Switching unit**" is used for control of switching unit module, connected to ADC DAC modules outputs. It is used to control relays status connected to commutator bar by means of digital port of the following modules: ZET 220, ZET 230;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Mask - the value of the relay status mask.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- Combination (0) - status of all relays.
- DeviceNum (0) - serial number of the device.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Mask - The value of the relay status mask (number).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- LONG Combination - Set the status of all relays (from 0 to infinity).
- LONG DeviceNum - Set the serial number of the device (number).

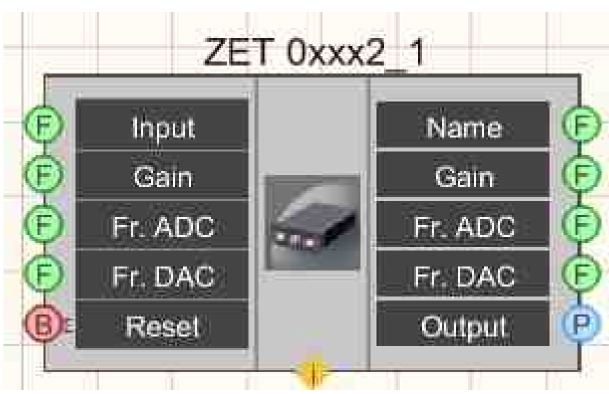
➤**Methods:**

- LONG Rele(LONG num) - Returns the status of a specific relay.
- void Rele(LONG num, LONG newVal) - Set the specific relay's status and value, where :
  num - new relay number;
  newVal - new value of the relay status .
- void Reset() - Resets to status B, where Reset () is the reset value to status B (any value).
- void Set() - Sets to state B, where Set() is the value of setting to state B (any value).

30.3.ZET 0xxx2

The component **"ZET 0xxx2"** is used for measuring the parameters of spectral signals components from various signals, correlational structure of the signals, generation of electrical signals with preset metrological parameters, noise and vibration measurements in 1/3 octave band, thus allowing to use it as vibration and noise meter. It is used for visual representation, data acquisition and signals processing from various sensors: vibration transducers, measuring microphones, hydrophones with ICP integrated amplifiers, etc. The components is necessary for control, measuring and diagnostic equipment used for various measurements and tests, for long-term processes monitoring (both stationary and dynamic);

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:

- Input - set the current channel.
- Gain - set the channel gain.

- Fr. ADC - Set the sampling frequency of the ADC.
- Fr. DAC - Set the sampling frequency of the DAC.
- Reset - Update device settings.

➤ Output:

- Name - device name.
- Gain - gain of the current channel.
- Fr. ADC - sampling frequency of the ADC.
- Fr. DAC - sampling frequency of the DAC.
- Output - selected analyzer channel.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤Input:

- Input - The value of the current channel number (from 0 to (number of channels - 1)).
- Gain - The value of the gain of the current channel (from 1 to the maximum gain).
- Fr.ADC - The value of the ADC sampling frequency of the current channel (from the minimum ADC sampling frequency to the maximum ADC sampling frequency).
- Fr.DAC - The value of the DAC sampling frequency of the current channel (from the minimum DAC sampling frequency to the maximum DAC sampling frequency).

➤Custom common properties:

- Are available by the link below. 154

➤Custom private properties (the default value is shown in parentheses):

Doesn't have.

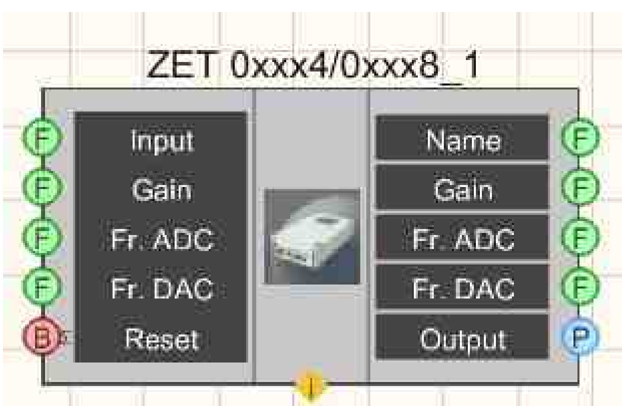
➢**Methods:**

- DOUBLE Amplify(LONG channum) - Returns the channel gain factor.
- void Amplify(LONG channum, DOUBLE newVal) - Set the channel number and gain, where:
  channum - new channel number;
  newVal - new channel gain value .
- DOUBLE FreqADC() - Returns the sampling frequency of the ADC .
- void FreqADC(DOUBLE newVal) - Set the sampling frequency of the ADC, where: newVal is the new value of the sampling frequency.
- DOUBLE FreqDAC() - Returns the sample frequency of the DAC .
- void FreqDAC(DOUBLE newVal) - Set the sampling frequency of the DAC, where newVal is the new value of the sampling frequency.

30.4.ZET 0xxx4/0xxx8

The component "**ZET 0xxx4/0xxx8**" is used for measuring the parameters of spectral signals components from various signals, correlational structure of the signals, generation of electrical signals with preset metrological parameters, noise and vibration measurements in 1/3 octave band, thus allowing to use it as vibration and noise meter. It is used for visual representation, data acquisition and signals processing from various sensors: vibration transducers, measuring microphones, hydrophones with ICP integrated amplifiers, etc. The components is necessary for control, measuring and diagnostic equipment used for various measurements and tests, for long-term processes monitoring (both stationary and dynamic);

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - set the current channel.
- Gain - set the channel gain.
- Fr. ADC - Set the sampling frequency of the ADC.
- Fr. DAC - Set the sampling frequency of the DAC.
- Reset - Update device settings.

➢ Output:

- Name - device name.
- Gain - gain of the current channel.
- Fr. ADC - sampling frequency of the ADC.
- Fr. DAC - sampling frequency of the DAC.
- Output - selected analyzer channel.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - The value of the current channel number (from 0 to (number of channels - 1)).
- Gain - The value of the gain of the current channel (from 1 to the maximum gain).
- Fr.ADC - The value of the ADC sampling frequency of the current channel (from the minimum ADC sampling frequency to the maximum ADC sampling frequency).
- Fr.DAC - The value of the DAC sampling frequency of the current channel (from the minimum DAC sampling frequency to the maximum DAC sampling frequency).

➢**Custom common properties:**

- Are available by the link below. 154

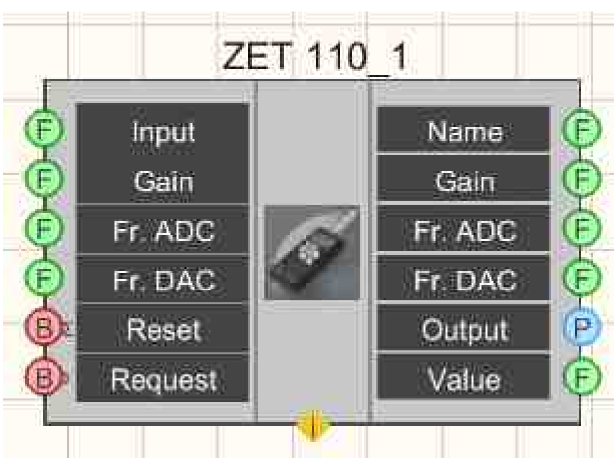
➢**Custom private properties (the default value is shown in parentheses):**
Doesn't have.

➢**Methods:**
- DOUBLE Amplify(LONG channum) - Returns the channel gain factor.
- void Amplify(LONG channum, DOUBLE newVal) - Set the channel number and gain, where:
  channum - new channel number;
  newVal - new channel gain value .
- DOUBLE FreqADC() - Returns the sampling frequency of the ADC .
- void FreqADC(DOUBLE newVal) - Set the sampling frequency of the ADC, where: newVal is the new value of the sampling frequency.
- DOUBLE FreqDAC() - Returns the sample frequency of the DAC .
- void FreqDAC(DOUBLE newVal) - Set the sampling frequency of the DAC, where newVal is the new value of the sampling frequency.

30.5.ZET 110

The component "**ZET 110**" is used for measuring various signals parameters, noise and vibration measurements, representation of the measured parameters, constant recording of signals to non-volatile integrated memory (volume – up to 2 Gb);

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - set the current channel.
- Gain - set the channel gain.
- Fr. ADC - Set the sampling frequency of the ADC.
- Fr. DAC - Set the sampling frequency of the DAC.
- Reset - Update device settings.

➢ Output:

- Name - device name.
- Gain - gain of the current channel.
- Fr. ADC - sampling frequency of the ADC.
- Fr. DAC - sampling frequency of the DAC.
- Output - selected analyzer channel.

➢Input:

- Input - set the current channel.
- Gain - set the channel gain.
- Fr. ADC - Set the sampling frequency of the ADC.
- Fr. DAC - Set the sampling frequency of the DAC.
- Reset - Update device settings.
- Request - request for the measured value.

➢ Output:

- Name - device name.
- Gain - gain of the current channel.
- Fr. ADC - sampling frequency of the ADC.
- Fr. DAC - sampling frequency of the DAC.
- Output - selected analyzer channel.
- Value - the channel that outputs the measured value.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- typecalc (PEAK) - type of calculated values.
- fltrnm (HP1) - set the filter.
- device (Audio-noise meter) - device type:
    1. Audio-noise meter.
    2. Vibration meter.
    3. Recorder.
- sensor (Profile 1) - Set the sensor profile.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - The value of the current channel number (from 0 to (number of channels - 1)).
- Gain - The value of the gain of the current channel (from 1 to the maximum gain).
- Fr.ADC - The value of the ADC sampling frequency of the current channel (from the minimum ADC sampling frequency to the maximum ADC sampling frequency).
- Fr.DAC - The value of the DAC sampling frequency of the current channel (from the minimum DAC sampling frequency to the maximum DAC sampling frequency).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR TypeCalc - Set the type of calculated values: PEAK; RMS; DC; PP; RMS-MAX; RMS-MIN; SLOW; S-MAX; S-MIN; FAST; F-MAX; F-MIN; Imp; I-MAX; LEQ; PEAK-MAX;AMP (string).
- BSTR FltrNM - Set the filter: HP1; HP3; HP10; Vel1; Vel3; Vel10; MFV; Dil1; Dil3; Dil10; wb; WC; wd; We; wj; wk; wm; Fk; fm; wh; Fh; A; C; LIN (string).
- BSTR Device - Set the device type: Audio-noise meter; Vibration meter; Recorder (string).
- BSTR Sensor - Set the sensor profile: Profile 1; Profile 2; Profile 3; Profile 4; Profile 5 (string).

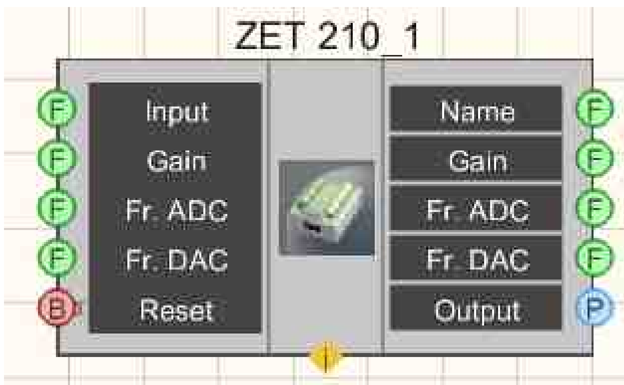
➢**Methods:**

- DOUBLE Amplify(LONG channum) - Returns the channel gain factor.
- void Amplify(LONG channum, DOUBLE newVal) - Set the channel number and gain, where:
  channum - new channel number;
  newVal - new channel gain value .
- DOUBLE FreqADC() - Returns the sampling frequency of the ADC .
- void FreqADC(DOUBLE newVal) - Set the sampling frequency of the ADC, where: newVal is the new value of the sampling frequency.
- DOUBLE FreqDAC() - Returns the sample frequency of the DAC .
- void FreqDAC(DOUBLE newVal) - Set the sampling frequency of the DAC, where newVal is the new value of the sampling frequency.

30.6.ZET 210

The component "**ZET 210**" is used for measuring signals parameters in wide frequency range (with sampling frequency up to 400 kHz), received from various primary transducers. Digital (DB-15) and analog (DB-25) inputs can be used for the purpose of various mechanisms control;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - set the current channel.
- Gain - set the channel gain.
- Fr. ADC - Set the sampling frequency of the ADC.
- Fr. DAC - Set the sampling frequency of the DAC.
- Reset - Update device settings.

➢ Output:
- Name - device name.
- Gain - gain of the current channel.
- Fr. ADC - sampling frequency of the ADC.
- Fr. DAC - sampling frequency of the DAC.
- Output - selected analyzer channel.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - The value of the current channel number (from 0 to (number of channels - 1)).
- Gain - The value of the gain of the current channel (from 1 to the maximum gain).
- Fr.ADC - The value of the ADC sampling frequency of the current channel (from the minimum ADC sampling frequency to the maximum ADC sampling frequency).
- Fr.DAC - The value of the DAC sampling frequency of the current channel (from the minimum DAC sampling frequency to the maximum DAC sampling frequency).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

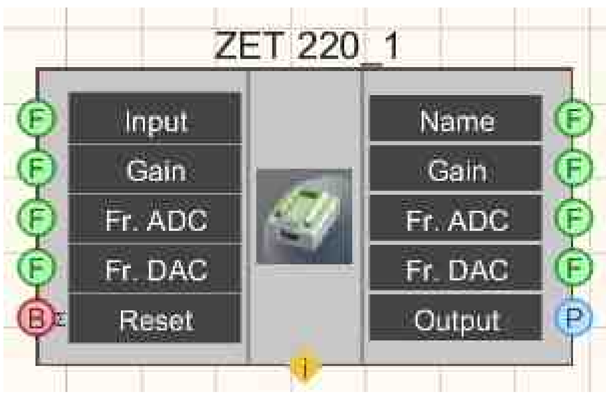
➢**Methods:**

- DOUBLE Amplify(LONG channum) - Returns the channel gain factor.
- void Amplify(LONG channum, DOUBLE newVal) - Set the channel number and gain, where:
  channum - new channel number;
  newVal - new channel gain value .
- DOUBLE FreqADC() - Returns the sampling frequency of the ADC .
- void FreqADC(DOUBLE newVal) - Set the sampling frequency of the ADC, where: newVal is the new value of the sampling frequency.
- DOUBLE FreqDAC() - Returns the sample frequency of the DAC .
- void FreqDAC(DOUBLE newVal) - Set the sampling frequency of the DAC, where newVal is the new value of the sampling frequency.

30.7.ZET 220

The component "**ZET 220**"" is used for measuring signals parameters with high precision and in wide frequency range (the signals are received from various primary transducers: thermal resistances, thermocouples, sensors with universal current output 4… 20 mA, accelerometers BC 201/202);

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - set the current channel.
- Gain - set the channel gain.
- Fr. ADC - Set the sampling frequency of the ADC.
- Fr. DAC - Set the sampling frequency of the DAC.
- Reset - Update device settings.

➢ Output:

- Name - device name.
- Gain - gain of the current channel.
- Fr. ADC - sampling frequency of the ADC.
- Fr. DAC - sampling frequency of the DAC.
- Output - selected analyzer channel.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - The value of the current channel number (from 0 to (number of channels - 1)).
- Gain - The value of the gain of the current channel (from 1 to the maximum gain).
- Fr.ADC - The value of the ADC sampling frequency of the current channel (from the minimum ADC sampling frequency to the maximum ADC sampling frequency).
- Fr.DAC - The value of the DAC sampling frequency of the current channel (from the minimum DAC sampling frequency to the maximum DAC sampling frequency).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

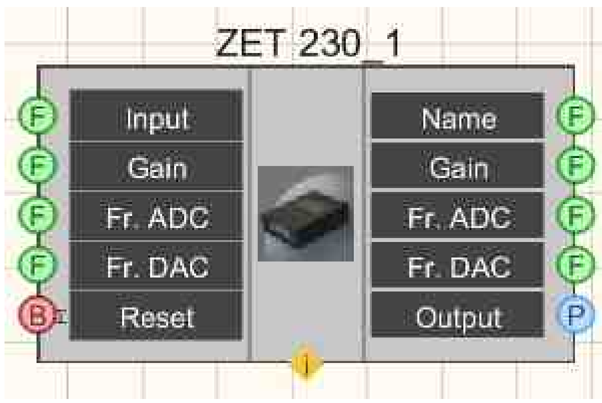
➢**Methods:**

- DOUBLE Amplify(LONG channum) - Returns the channel gain factor.
- void Amplify(LONG channum, DOUBLE newVal) - Set the channel number and gain, where:
  channum - new channel number;
  newVal - new channel gain value .
- DOUBLE FreqADC() - Returns the sampling frequency of the ADC .
- void FreqADC(DOUBLE newVal) - Set the sampling frequency of the ADC, where: newVal is the new value of the sampling frequency.
- DOUBLE FreqDAC() - Returns the sample frequency of the DAC .
- void FreqDAC(DOUBLE newVal) - Set the sampling frequency of the DAC, where newVal is the new value of the sampling frequency.

30.8.ZET 230

The component "**ZET 230**" is used for measuring signals parameters with high precision and in wide frequency range (the signals are received from various primary transducers: accelerometers BC 201, BC 202, microphones BC 501;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Input - set the current channel.
- Gain - set the channel gain.
- Fr. ADC - Set the sampling frequency of the ADC.
- Fr. DAC - Set the sampling frequency of the DAC.
- Reset - Update device settings.

➢ Output:

- Name - device name.
- Gain - gain of the current channel.
- Fr. ADC - sampling frequency of the ADC.
- Fr. DAC - sampling frequency of the DAC.
- Output - selected analyzer channel.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - The value of the current channel number (from 0 to (number of channels - 1)).
- Gain - The value of the gain of the current channel (from 1 to the maximum gain).
- Fr.ADC - The value of the ADC sampling frequency of the current channel (from the minimum ADC sampling frequency to the maximum ADC sampling frequency).
- Fr.DAC - The value of the DAC sampling frequency of the current channel (from the minimum DAC sampling frequency to the maximum DAC sampling frequency).

➢Total properties (environment):

- Are available by the link below. 154

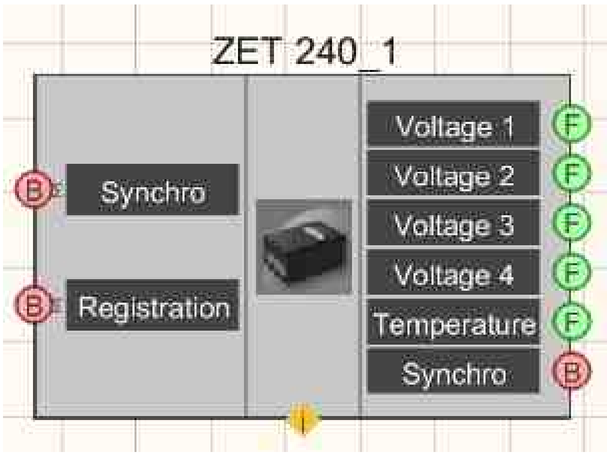
➢**Methods:**

- DOUBLE Amplify(LONG channum) - Returns the channel gain factor.
- void Amplify(LONG channum, DOUBLE newVal) - Set the channel number and gain, where:
    channum - new channel number;
    newVal - new channel gain value .
- DOUBLE FreqADC() - Returns the sampling frequency of the ADC .
- void FreqADC(DOUBLE newVal) - Set the sampling frequency of the ADC, where: newVal is the new value of the sampling frequency.
- DOUBLE FreqDAC() - Returns the sample frequency of the DAC .
- void FreqDAC(DOUBLE newVal) - Set the sampling frequency of the DAC, where newVal is the new value of the sampling frequency.

30.9.ZET 240

The component **"ZET 240"** is used for measuring and registering of signals from seismic receivers, the component also operates as a measuring instrument for electrical signals parameters;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

- Input:
- Synchro - a channel for receiving data from the device.
- Register. - channel to switch to registration mode

➢ Output:
- Voltage 1 - voltage level on the first channel of the device, V.
- Voltage 2 - voltage level on the second channel of the device, V.
- Voltage 3 - voltage level on the third channel of the device, V.
- Voltage 4 - voltage level on the fourth channel of the device, V.
- Temperature - temperature on the device.
- Synchro - status of sampling in GPS .

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Device (0) - the serial number of the device is set.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR Device - Set the serial number of the selected device (string).

30.10.ZET 440

The component "**ZET 440**" is used for conversion of electrical transducer's high-impedance charge signal (from vibration sensor, accelerometer or hydrophone) into current signal and conversion of ICP sensors signals into voltage signal;

The component allows to connect accelerometers with charge output or ICP to FFT Spectrum Analysis and ADC / DAC modules.

Thus, it is possible to connect hydrophones BC 311 to Narrow-band spectrum analyzer ZET 017. The amplifier ZET 440 is used in sensors control system in order to connect reference accelerometer AP10 to Narrow-band spectrum analyzer ZET 017;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Gain - a channel for receiving data from a device.
- Filter - channel for switching to registration mode

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- SerialNumber (0 x 0000000000000000) - serial number of the device.
- amplification (x 1) - device gain.
- filtfrequecy (0,1) - high pass filter cutoff frequency, Hz:

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Ser. number - The value of the serial number of the interface board (number).
- Filter - Signal filtering value (any number).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR SerialNumber - Set the serial number (string).
- BSTR Amplification - Gain setting: x 1; x 10; x 100 (string).
- BSTR FiltrFrequency - Set the cutoff frequency of the high-pass filter: 0.1 Hz; 1 Hz; 10 Hz

30.11.ZET 7000 channel reading

The component "**ZET 7000 channel reading**" is used for reading the data from sensors of 7xxx series. This component allows to calculate the current measured value of the signal by device's serial number, channel number and device type. Serial number and device type are obtained from "Device manager" program. Serial number is shown in 16-digit format, without particular type indication (i.e., without 0x index);

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Doesn't have

➢ Output:
- Value - the channel to get the channel value.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- termotype (UNKNOWN) - Set the device type.
- DeviceSerNum (0) - serial number of the device.
- NumChannel (0) - set the channel number.

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

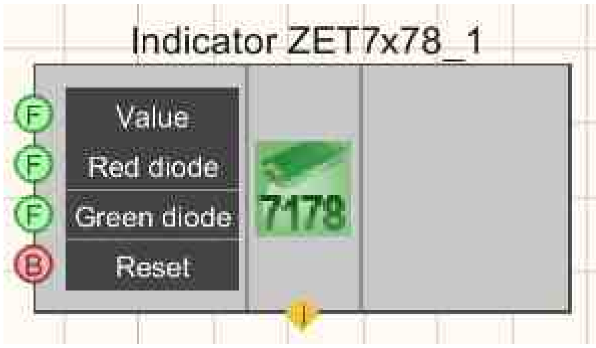
- BSTR TermoType - set the device type: UNKNOWN; 7070; ZETKey; 7010; 7021; 7120; 7174; 7172m; 7070; 7140; 7020; 7012; 7140S; 7052; 7060; 7060s; 7060g; 7090; 7051;SIMPLE_UART; 7110; 7152; 7111; 7175; 7141; 7176; 7140RMS; 7153; 7060e; 7160e; 7160g; 7178; 7151; 7190; 7190r; 7160; 7080i; 7080v; 7062; 7180v; 7180i; 7121; 7112; 7172s (string).
- BSTR DeviceSerNum - Set the serial number (string).
- ULONG NumChannel - Set the channel number (string).

30.12.Indicator ZET7x78

The component "**Indicator ZET7x78**" is intended for connection to measuring networks on the basis of digital sensors with CAN interface and serves for indication of values of measured parameters. Also, the module is used to monitor the voltage on the power bus in the network using the "Oscilloscope" program from the ZETLAB package or on the module screen.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢ Input:

- Value - a channel for getting the display value on the indicator.
- Red diode - red LED indication mode.
- Green diode - green LED indication mode.
- Reset - reset the device's light indicators.

➢Output:

Doesn't have

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- SerialNumber (0) - set the serial number of the device.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR SerialNumber - set the serial number of the device (string).

30.13.Configurator ZET7xxx

The component "**Configurator ZET7xxx**" is designed to configure ZET7xxx.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Save - set to save the ZET7xxx settings.

➢Output:
Doesn't have

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- SerialNumber (0) - the serial number of the device.
- StuctType (0) - The type of structure to tune.
- StuctNumber (0) - Ordinal number of the structure to tune (among the same types).
- ChannelNumber (0) - Ordinal number of the tuning channel (among the same types).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

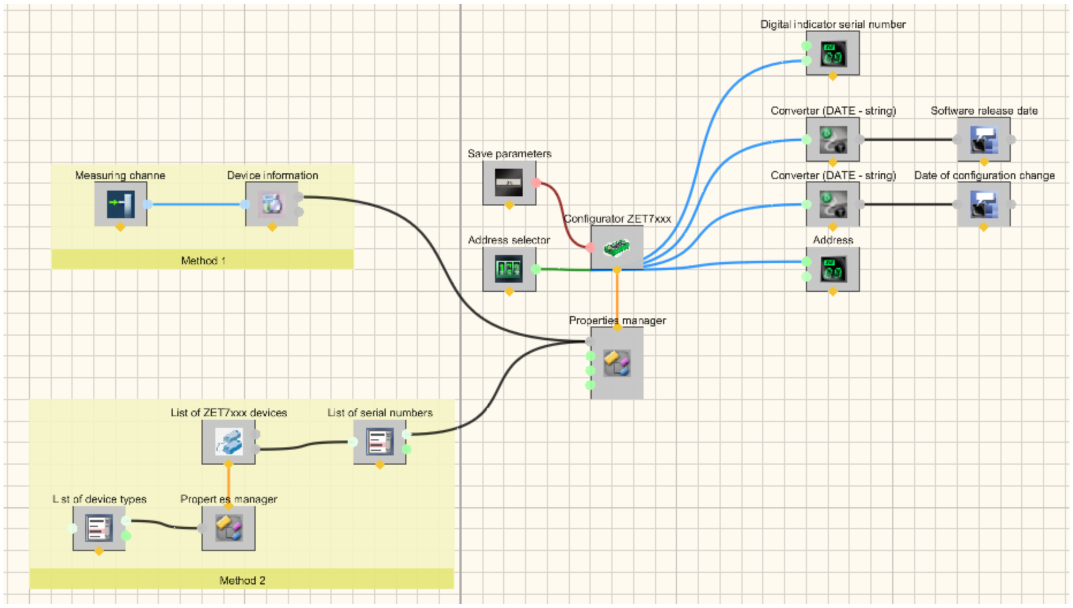
- BSTR SerialNumber - Set the serial number of the device (string).
- LONG StuctType - Set the type of structure to tune (number).
- LONG StuctNumber - Set the ordinal number of the structure to tune (among the same types) (number).
- LONG ChannelNumber - Set the ordinal number of the tuning channel (among the same types) (number).

➢**Methods:**

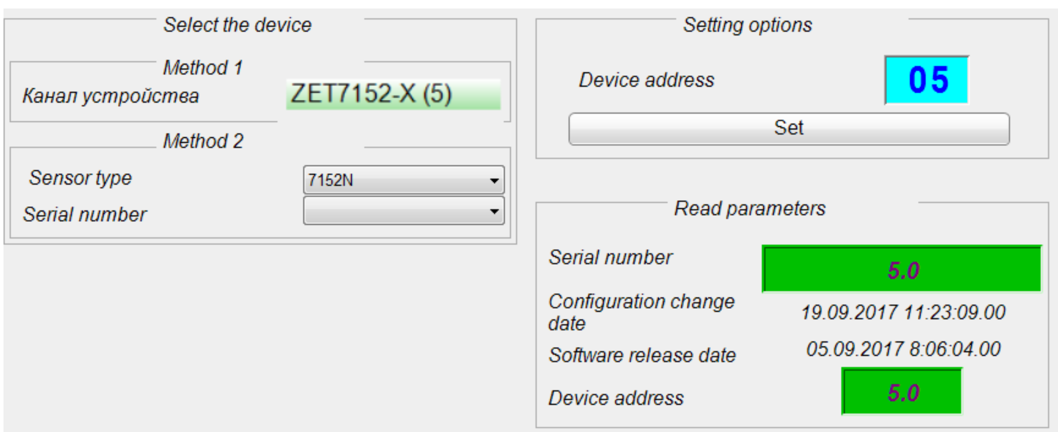
- void Synchro() - Command to save and read parameters from the sensor

Example No. 1

Project in SCADA ZETView



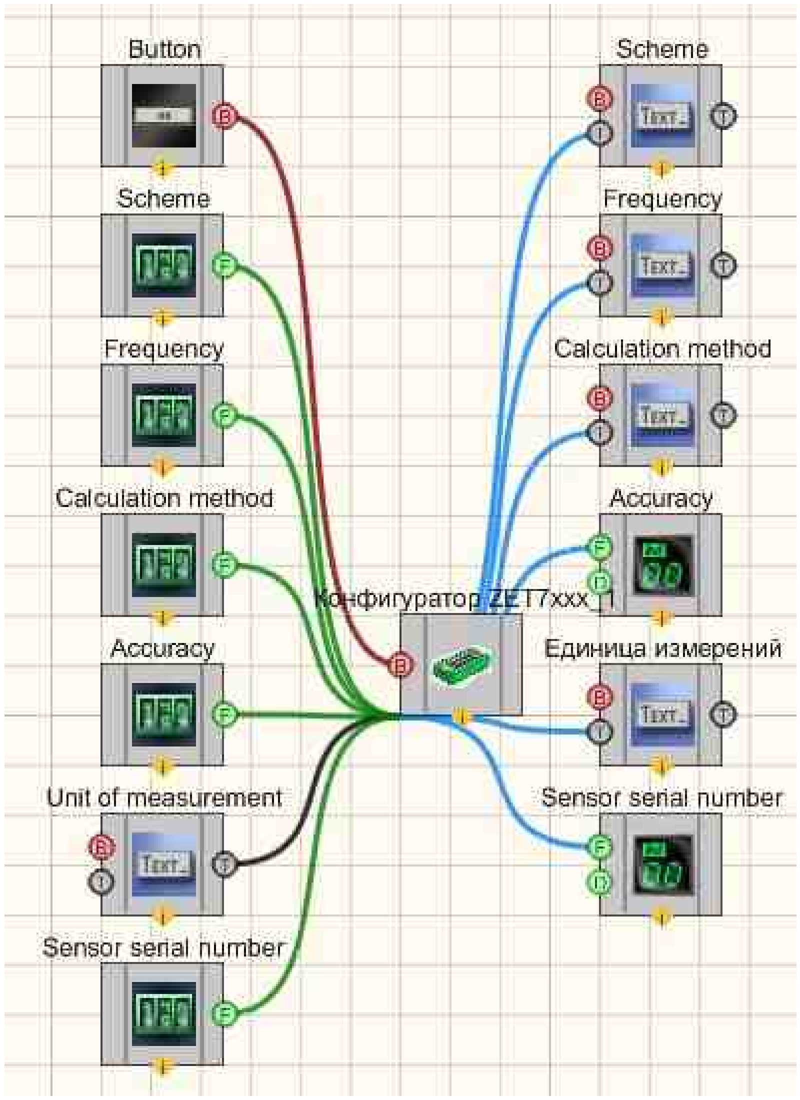
Project operation result



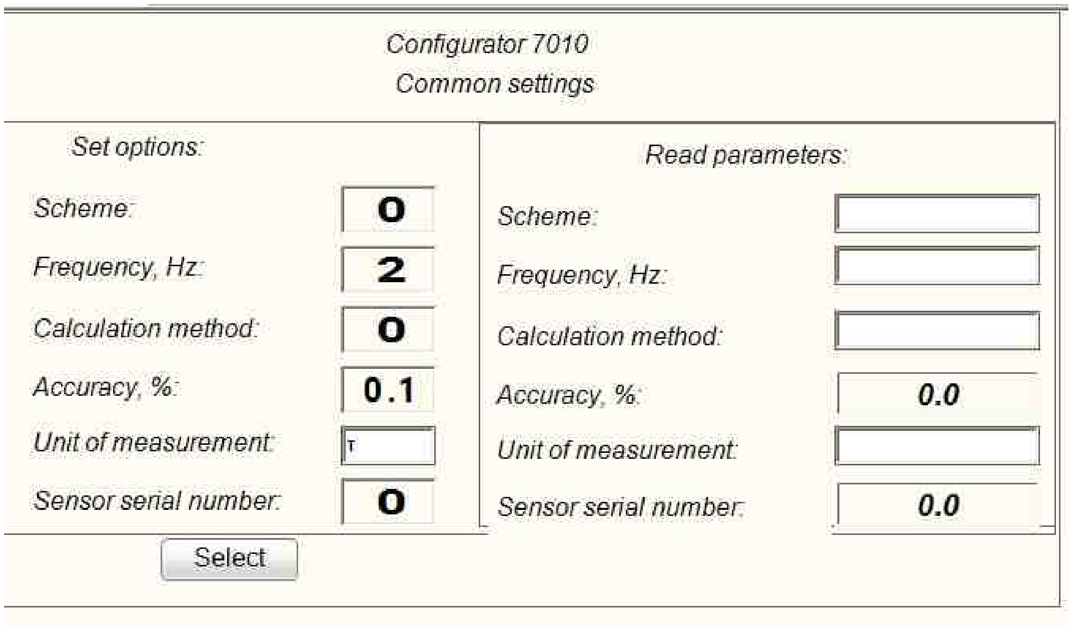
View the example in ZetView

Example No. 2

Project in SCADA ZETView



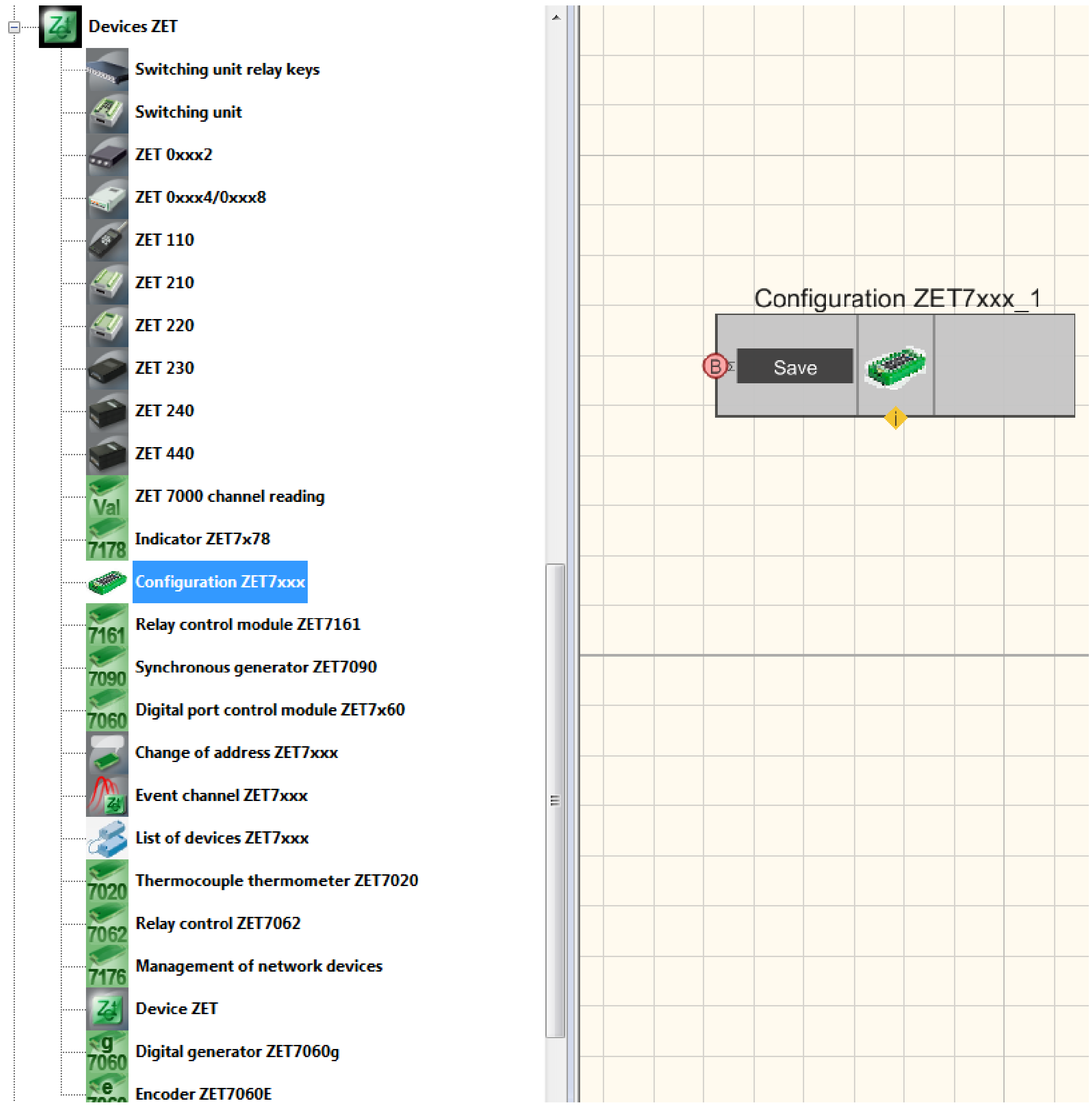
Project operation result



View the example in ZetView

Working with ZET7xxx sensors via the ZETView SCADA system

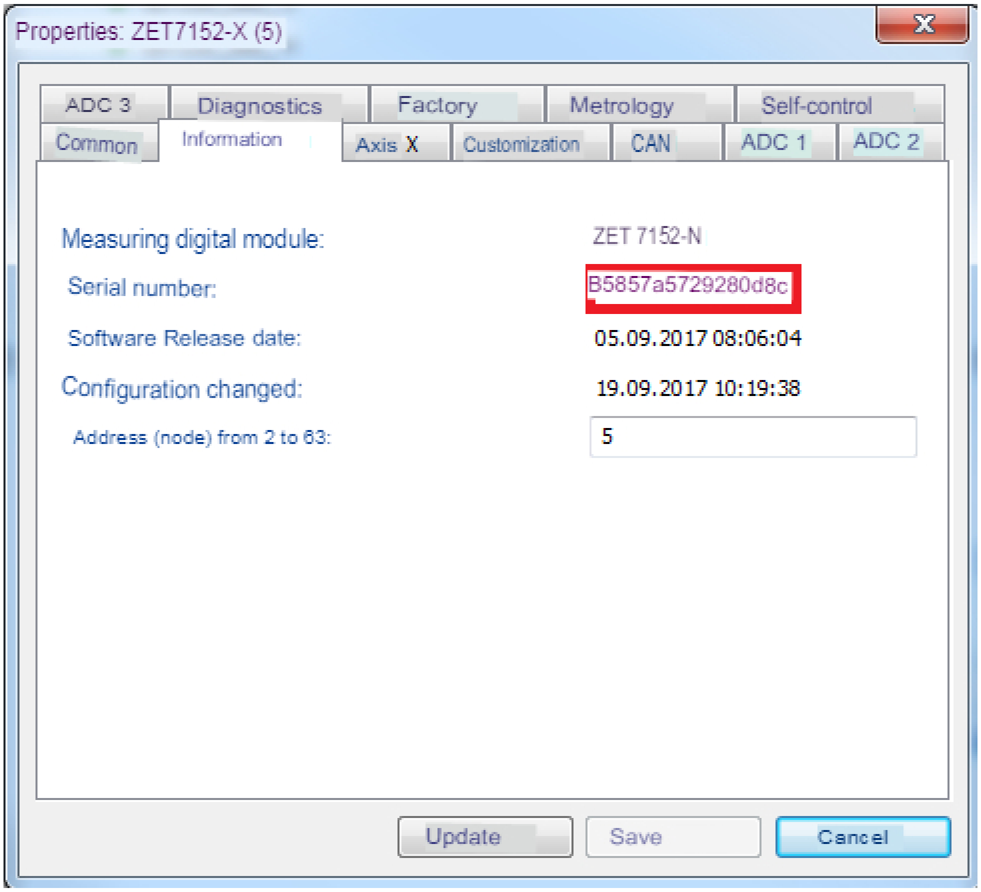
To control sensors of the ZET 7 xxx series in the ZETView SCADA system, the ZET 7 xxx Configuration component from the ZET Devices tab is used.



Component "ZET7xxx Configuration"

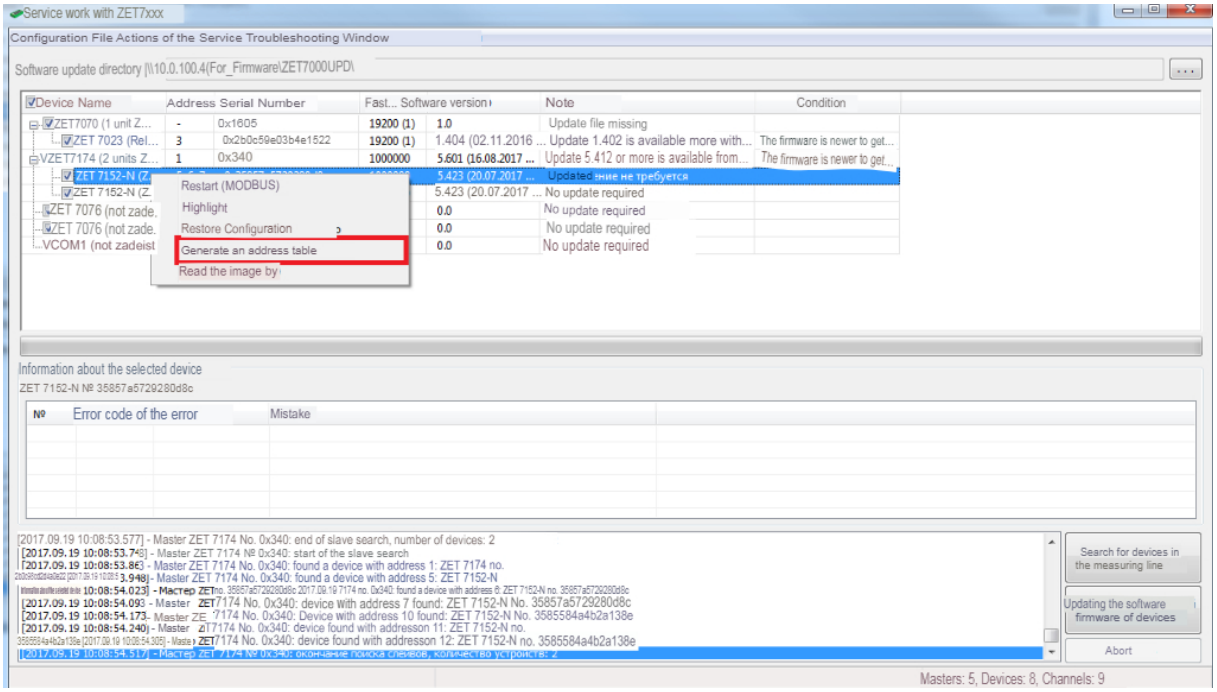
For this component to work, you need to set its settings, namely: serial number, structure type, structure number (some structures have the same structure type, so they are separated by numbers), and the channel number that needs to be configured (if there are several channels in the device, then this parameter is used to determine the setting of which channel you are interested in).

The serial number of the device can be taken from the "Device Manager" program from the "Service" tab. After opening the device parameters, in the Information tab, you can view the serial number of the device and set it to the component.



Serial number of the device

We can get the structure type from the program "Service work with ZET7xxx " from the main menu of the ZETLab panel . Please note that for the correct operation of this program, it is necessary to close other ZETLab programs working with the device. Also, if you do not have the sensor you need in the list, make sure that you have enabled the device through which you have connected the sensor (ZET7070/ ZET7176), to do this, right-click on the desired sensor in the program list and select "Enable". Next, right-click on the sensor you are interested in and select "Generate address table", when the generation is completed, the folder with the location of the generated address table will open.



Type of structure from the program "Service work with ZET7xxx"

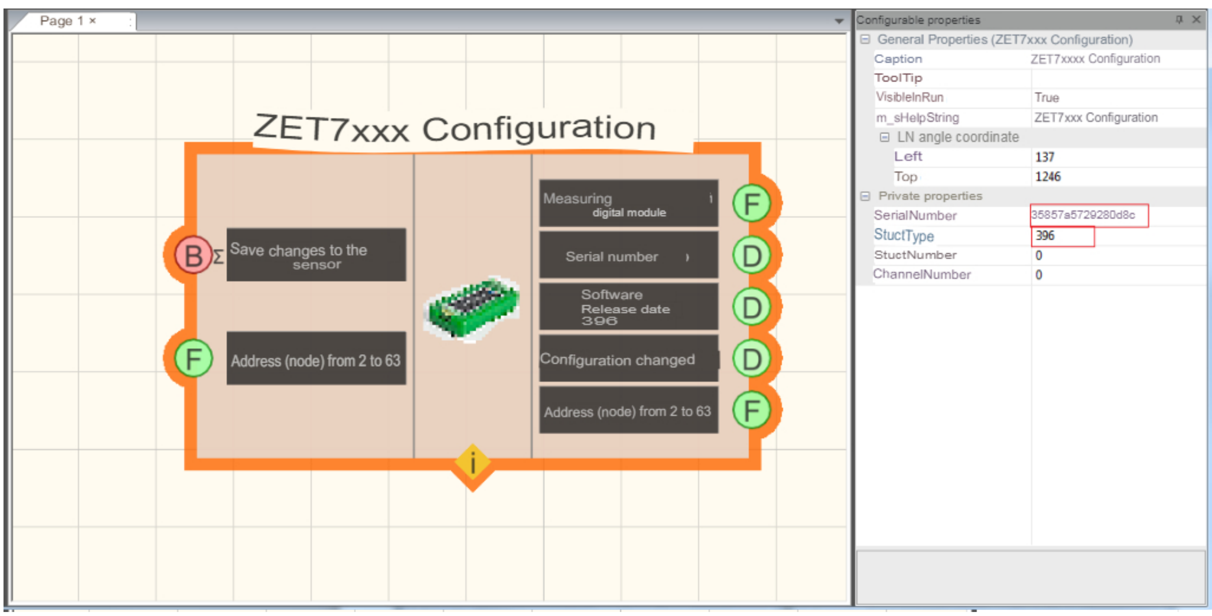
Open the resulting table. From it we need to get the structure type number. To do this, find the required structure in the list; next to its name, the ID of this structure will be written in hexadecimal form. Using a standard calculator, convert the resulting number to a decimal form, as shown in the screenshots below, and enter it into the settings of the ZET7xxx Configurator.

MODBUS ZET 7152-N register address table No. 0x35857a5729280d8c_5

Name of the parameter	Address, WORD hex (WORD dec)	Address in the structure, WORD hex (WORD dec)	Data type	Number registers (in of words)	Accepted values
Information ( ), ID = 0x18c address = 0x00 (00)					
Measuring digital module	0x04 (04)	0x04 (04)	int (type 17)	2	Arbitrary value (read-only)
Serial number	0x06 (06)	0x06 (06)	longlong (type 14)	4	Arbitrary value (read-only)
Software Release date	0x0a (10)	0x0a (10)	time (type 11)	2	Arbitrary value (read-only)
Configuration changed	0x0c (12)	0x0c (12)	time (type 11)	2	Arbitrary value (read-only)
Address (node) from 2 to 63	0x0e (14)	0x0e (14)	int (type 3)	2	Arbitrary value
X-axis (Measurement parameters), ID = 0xd0, Address = 0x10 (16)					
Current measured value (in units)	0x14 (20)	0x04 (04)	float (type 6)	2	Arbitrary value (read-only)
Data refresh rate, Hz	0x16 (22)	0x06 (06)	float (type b)	2	Arbitrary value (read-only)
Unit of measurement	0x18 (24)	0x08 (08)	char[8] (type 1)	4	Arbitrary value (read-only)
Name of the X axis	0x1c (28)	0x0c (12)	char[32] (type 1)	16	Arbitrary value
Minimum value (in units)	0x2c (44)	0x1c (28)	float (type 6)	2	Arbitrary value (read-only)
Maximum value (in units)	0x2e (46)	0x1e (30)	float (type b)	2	Arbitrary value (read-only)
Reference value for calculation in dB	0x30 (48)	0x20 (32)	float (type b)	2	Arbitrary value (read only)
Sensitivity, In/unit.	0x32 (50)	0x22 (34)	float (type b)	2	Arbitrary value (read-only)
Sensitivity threshold (in units)	0x34 (52)	0x24 (36)	float (type b)	2	Arbitrary value (read-only)
Setup (Digital sensor setup 7152N), ID = 0x33, Address = 0x36 (54)					
Sensor Type	0x3a (58)	0x04 (04)	int (type 3)	2	Arbitrary value (read-only)
Conversion frequency, Hz	0x3c (60)	0x06 (06)	float (type 7)	2	2000 1000 500 250 125

The number of the structure type, next to its name, the ID of this structure will be written in hexadecimal form.

After setting the serial number and structure type, the component will automatically have Output and input contacts with the corresponding fields.

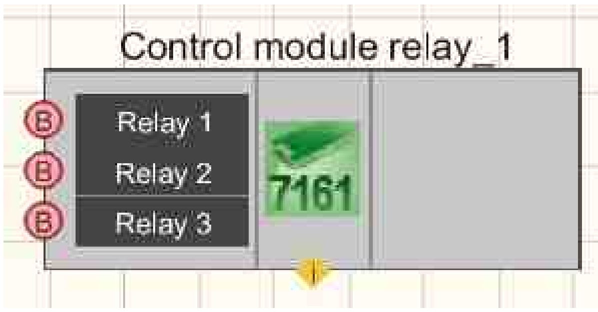


Further, you can receive and record data to the device by connecting control components to the corresponding contact.

30.14.Relay control module ZET7161

The component "**Relay control module ZET7161**" is used producing "dry contact" signal, it has 3 input relays, activated upon receipt of signal from CAN interface;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Relay 1 - control of the status of relay 1.
- Relay 2 - control of the status of relay 2.
- Relay 3 - control of the status of relay 3.

➢Output:

Doesn't have

The relay is activated by a command received via the CAN interface.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- SerialNumber (0) - set the serial number of the interface converter.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

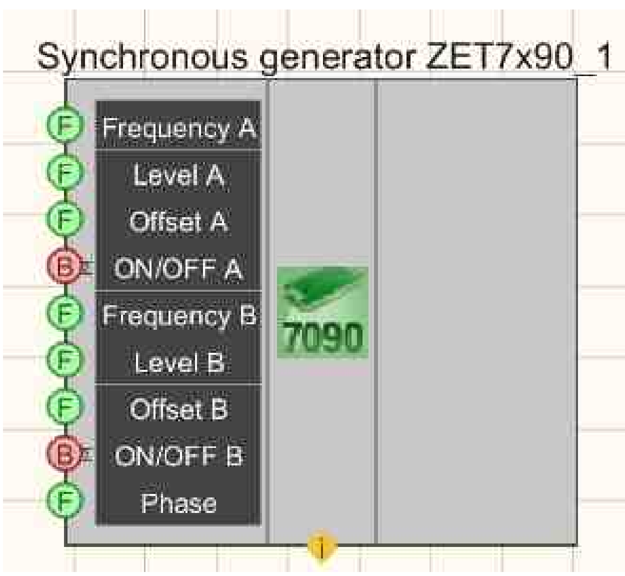
➢**Custom private properties (the default value is shown in parentheses):**

•BSTR SerialNumber - Set the serial number of the device (string).

30.15.Synchronous generator ZET7090

The component "**Synchronous generator ZET7090**" - analog two-channel synchronous generator with RS-485 interface. It is used a get sine, impulse or saw-shape signals. The component has 2 synchronous channels and can be used independently or in control and automation systems;

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Frequency A - frequency channel of signal A of the generator.
- Level A - signal level channel A of the generator.
- Offset A - channel offset signal A generator relative to zero.
- ON/OFF A - the status of operation of the generator A.

- Frequency B - signal frequency channel B of the generator.
- Level B - signal level channel B of the generator.
- Offset B - channel offset signal In the generator relative to zero.
- ON/OFF B - the status of operation of the generator B.
- Phase - the phase of the signal B of the generator relative to the signal A of the generator.

➢Output:

Doesn't have

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- SerialNumber(0) - serial number of the device.
- Frequency A (1000) - generator signal frequency A, Hz.
- Level A (1) - generator signal level A, V
- Shift A (0) - offset of the generator signal A relative to zero, B.
- Status A (false) - Enabled (true)/Disabled (false) the status of generator A operation.
- Frequency V (1000) - frequency of signal V of the generator, Hz.
- Level B (1) - signal level B of the generator, V
- Shift B (0) - offset of the signal of the second generator relative to zero, V.
- Status B (false) - Enabled (true)/Disabled (false) the status of operation of the B generator.
- Phase (0) - phase of the signal B of the generator relative to the signal A of the generator, degr.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Frequency A - Set the frequency channel of the signal A of the generator (from 0.001 to 30 kHz).
- Level A - Set the signal level channel A of the generator (from 0 to 1.7 V).

- Offset A - the channel is set to offset the signal A of the generator relative to zero (from 0 to 5 V).
- Frequency B - Set the frequency channel of the generator signal A (from 0.001 to 30 kHz).
- Level B - Set the signal level channel A of the generator (from 0 to 1.7 V).
- Offset B - the channel is set to offset the signal A of the generator relative to zero (from 0 to 5 V).
- Phase - the phase of the signal B of the generator relative to the signal A of the generator (from 0 to 360 degrees).

➤**Custom common properties:**

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

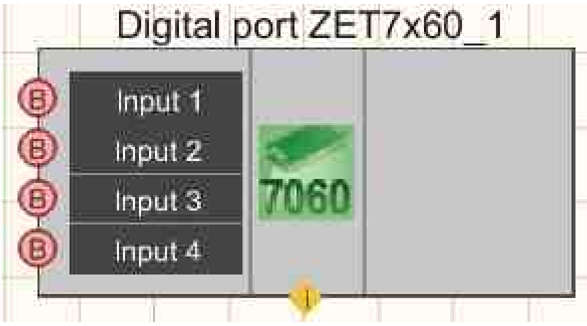
- LONG serialNumber - Set the serial number of the device (number).
- FLOAT FrequencyA - Set the frequency of the generator signal A, Hz (number).
- FLOAT LevelA - Set the signal level A of the generator, V (number).
- FLOAT ShiftA - Set the offset of the generator signal A relative to zero, V (number).
- VARIANT_BOOL StatusA - Set Enable\Disable the status of operation A of the generator:
  - true - the A generator is enabled;
  - false - the A generator is disabled.
- FLOAT FrequencyB - Set the frequency of the signal B of the generator, Hz (number).
- FLOAT LevelB - Set the signal level B of the generator, V (number).
- FLOAT ShiftB - Set the offset of the signal B of the generator relative to zero, V (number).
- VARIANT_BOOL StatusB - Set Enable\Disable the status of operation B of the generator:
  - true - the B generator is enabled;
  - false - the B generator is disabled.
- FLOAT Phase - Set the phase of the signal B of the generator relative to the signal A of the generator, degr (number).

30.16.Digital port control module ZET7x60

The component "**Digital port control module ZET7x60**" is used digital port / logical analyzer with RS-485 or CAN interface. It is used for the control of the connected device. The component is used in control and automation systems.

Appearance of the component:

Developer interface	Operator interface

<table>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:
- Input 1 - control the status of channel 1.
- Input 2 - control the status of channel 2.
- Input 3 - control the status of channel 3.
- Input 4 - control the status of channel 4.

➢Output:
Doesn't have

The port is triggered by a command received via the interface.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- SerialNum (0) - set the serial number of the device.

Questions and answers:
Question: Please help with the operation of the ZET 7160 digital port. Work in the SCADA. The first 2 channels are input, the second 2 channels are output.
example. On the first sales channel, the voltage is 5 volts, at the output in the SCADA configuration Zet 7 xxx to “current measured value, unit.» there is no reaction.
At the input, we sell physical quantities, voltage, and at the output in SCAD we must also receive voltage in digital form.
In the "Device Manager" everything is configured accordingly. Specify where I should take the signal "Multichannel oscilloscope"?
Another such moment: 4 signals were displayed in the “Device Manager”, as I understand that I use 1 and 2 as inputs, and 3 and 4 as outputs? Then it's not clear how to configure them in the device manager? Because if I set 2 input and 2 output signals in signal 1, then this is automatically displayed in the rest of the signals.

Answer:

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

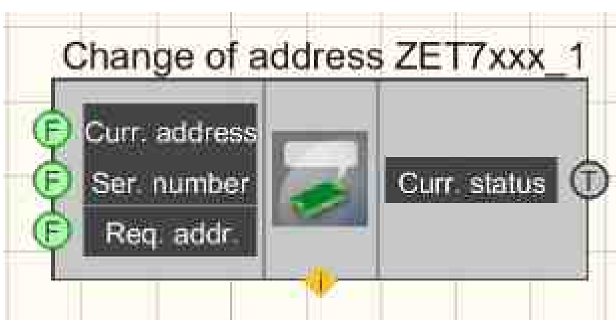
➢**Custom private properties (the default value is shown in parentheses):**

•BSTR SerialNumber - Set the serial number of the device (string).

30.17.Change of address ZET7xxx

The component "**Change of address ZET7xxx**" is allows to change addresses of Zet7xxx sensors.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Curr. address - set the current address of the device.
- Ser. number - set the serial number of the interface converter.
- Req. addr. - set the required device address.

➢ Output:

- Curr. status - set the current status of the address change.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- CurrNode (0) - Set the current node number.
- SerialNum (0) - Set the serial number of the interface converter
- NeedNode (0) - Set the required node address.
- Channel 1 Name (Channel_1) - Set the name of the first channel.
- Channel 2 Name (Channel_2) - Set the name of the second channel.
- Channel 3 Name (Channel_3) - Set the name of the third channel.
- Channel 4 Name (Channel_4) - Set the name of the fourth channel.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:

- Curr. address - Set the current address of the device (from the minimum allowable value of this device to the maximum allowable value of this device).
- Ser. number - Set the serial number of the interface converter
- Req. addr - Set the required address of the device (from the minimum allowable value of this device to the maximum allowable value of this device).
-

➢Custom common properties:

- Are available by the link below. 154

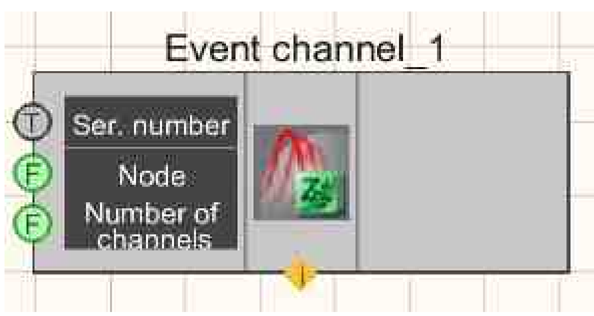
➢Custom private properties (the default value is shown in parentheses):

- LONG CurrNode - Set the current node address (number).
- LONG SerialNum - Set the serial number of the interface converter (number).
- LONG NeedNode - Set required node address (number).
- BSTR Channel1Name - Set the name of the first channel (string).
- BSTR Channel2Name - Set the name of the second channel (string).
- BSTR Channel3Name - Set the name of the third channel (string).
- BSTR Channel4Name - Set the name of the fourth channel (string).

30.18.Event channel ZET7xxx

The component **Event channel ZET7xxx** is designed to read data from event channels of ZET7xxx modules.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Ser. number - set the serial number of the interface board.
- Node - set the node number of the connected device.
- Number of channels - Set the number of event channels.

➢Output:

Doesn't have

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- Ser_number (0) — Set the serial number of the interface board.
- Num_node (0) — Set the node number.
- Quan (0) — Set the number of channels.
- WinSignal (False) — high resolution signal sampling (signal window).
- ChannelActivity(False) — Set the display the activity status of the parametric channel. There can be several parametric channels in one parametric device, they can be either on or off.
- NumPrevEvents (0) — Set the number of recent events to display. With it, you can customize the display of the number of events after starting the project.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Ser. number - set the serial number of the interface board.
- Node - set the node number of the connected device.
- Number of channels - set the number of event channels.

➤**Custom common properties:**

➤Total properties (environment):

- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR Ser_number - Set the serial number of the interface board (string).
- LONG Num_node - Set the node number (number).
- LONG Quan - Set and read the number of channels (from 0 to 5).

VARIANT_BOOL WinSignal - High resolution signal sampling setting (signal window):

- true - the signal window is enabled;
  false - the signal window is disabled.

- VARIANT_BOOL ChannelActivity - Set the display of parametric channel status activity. There can be several parametric channels in one parametric device, they can be either on or off..

true - parametric channel status is enabled;
false - parametric channel status disabled.

- LONG NumPrevEvents - Set the number of recent events to display (number).

30.19. List of ZET7xxx devices

The component "**List of ZET7xxx devices**" is intended for reading the list of devices of ZET7xxx modules.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢ Output:

- Outputs - channels of ZET 7 xxx devices are set .
- Devices – set a list of serial numbers of ZET 7 xxx devices .

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- DeviceType - set the device type.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

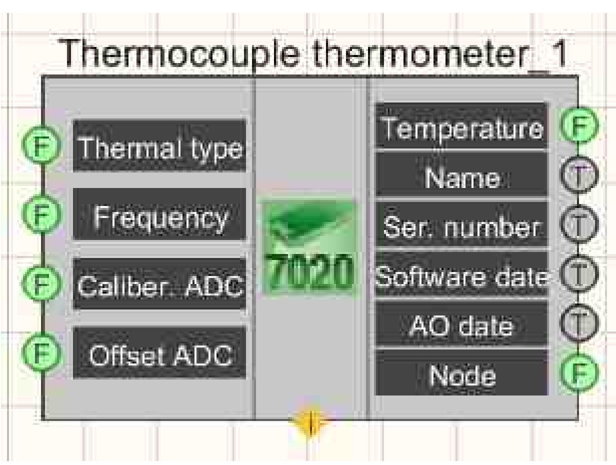
•BSTR DeviceType - Set the type of device (string).

30.20.TRC thermometer ZET7x20

The component **"TRC thermometer ZET7x20"** is used for temperature measurements in liquid, gaseous and granular environments that are non-aggressive to the TRC package. Application spheres: heat energy, chemical industry, metallurgy. The component is used together with intelligent temperature sensors with RS-485 interface or CAN.

The component is used for intelligent thermocouple thermometers: ZET7020, ZET 7120.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Thermal type - the type of thermocouple to be connected.
- Frequency - output signal frequency, Hz.
- Caliber. ADC - set the calibration constant is set in the ADC device.
- ADC offset - set the calibration offset in the ADC device.

➢ Output:
- Temperature - current temperature of the measuring channel.
- Name - set the current name of the measuring channel of the ZET 7020 sensor.
- Ser. number - set the current serial number of the ZET 7020 sensor.
- Software date - set the current date of creation of the ZET7020 sensor software..
- AO date - set the current date of creation of the ZET7020 hardware.
- Node - set the current modbus address of ZET 7020 device.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- Type (Voltage) - set the type of sensor to be connected to the thermocouple thermometer:
  1. Voltage;
  2. Resistor;
  3. Reference junction;
  4. Type R;
  5. Type J;
  6. Type T;
  7. Type E;
  8. Type K;
  9. Type N;
  10. Type A1;
  11. Type A2;
  12. Type A3;
  13. Type L;
  14. Type M.
- freq (1) - set the frequency, Hz
- calibrADC (1) - set the ADC gain calibration correction.
- offsetADC (0) - set the voltage offset, V.
- channelVal (0) - set the current value of the sensor connected to the thermometer thermocouple, units.
- sensortype (ZET_7020) - set the device type: ZET _7020; ZET_7120.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Thermal type - Value of the type of thermocouple to be connected (from 1 to 14).
- Frequency - Polling frequency value (1 or 4), Hz.
- Caliber. ADC - The value of the calibration constant in the ADC device (any number).
- ADC offset - The value of the calibration offset to the ADC device (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

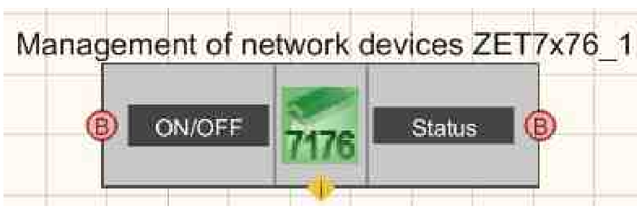
- BSTR Type - Set the type of sensor to be connected to the thermocouple thermometer: Voltage; Resistor; Reference junction; Type R; Type S; Type B; Type J; Type T; Type E; Type K; Type N; Type A1; Type A2; Type A3; Type L; Type M (string).
- BSTR freq - Set the frequency, Hz: 1 Hz; 4 Hz (string).
- FLOAT calibrADC - Set the calibration correction for the ADC gain (number).
- FLOAT offsetADC - Set the voltage offset, V (number).
- FLOAT channelVal - Set the current value of the sensor connected to the thermocouple thermometer (in units).
- BSTR sensorType - Set the device type: ZET _7020; ZET _7120 (string).

30.21.Management of network devices ZET7x76

The component "**Management of network devices ZET7x76**" is used for connection of measuring networks based on digital sensors with RS-485 or CAN interface to PC via Ethernet network.

ZETLab package contains drivers for **ZET7x76 modul** . The ***ZETLab*** data server automatically determines the presence of devices on the R -485 or CAN line, reads the name of the measuring channel, the unit of measurement, the upper and lower range of acceptable parameter values, and continuously reads the measuring channel data and forms a continuous synchronized channel in the ***ZETLab environment*** . The ***ZETLab*** data server scans the line for all available addresses and, in the event of a new device, dynamically connects a new channel to the system. When a new **ZET7x76 device is detected,** ***ZETLab*** data server also connects all channels on the go. Thus, the line can be serviced without interrupting the work process through other channels, i.e. the system allows "hot" replacement of elements.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- ON/OFF - device status control.

➢ Output:

- Status - set the status of the device.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- serialNumber (0) - set the serial number of the device.
- State (false) - device status (ON/OFF).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

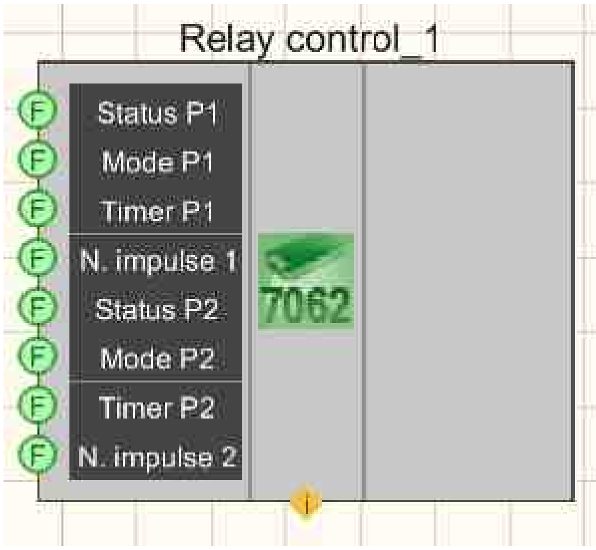
- BSTR SerialNumber - Set the serial number of the device (string).
  VARIANT_BOOL status - Set the status of the device:
- true - The status of the device is enabled;
  false - The status of the device is disabled.

30.22.Relay control ZET7062

The component "**Relay control ZET7062**" is used for relay control. Settings: by default. The data is saved to flash-memory. The component is used to set initial configuration of the relay;

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:

- Status P1 - relay status - changes status once in start mode.
- Mode P1 - start mode - single status change.
- Timer P1 - status change - by time.
- N. impulse P1 - number of impulses - how many times the relay will change its status in the timer mode (-1 - infinitely).
- Status P2 - relay status - changes status once during start mode.
- Mode P2 - start mode - single status change.
- Timer P2 - status change - by time.
- N. impulse P2 - number of impulses - how many times the relay will change its status in the timer mode (-1 - infinitely).

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- statusr1 (A) - status of relay 1 (A / B).
- modestartr1(Start) - start mode of relay 1 (Start/Timer).
- timerR1 (0) - Relay 1 start timer, s.
- impulseNumR1 (0) - Number of relay 1 impulses, (-1) - permanent.
- statusr2 (A) - status of relay 2 (A / B).

- modestartr2(Start) - start mode of relay 2 (Start/Timer).
- timerR2 (0) - Relay 2 start timer, s.
- impulseNumR2 (0) - Number of relay 2 impulses, (-1) - permanent.
- serialNumH (0) - set the high part of the serial number.
- serialNumL (0) - set the high part of the serial number.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Status P1 - Status value of relay 1 - changes status once in start mode (A / B).
- Mode P1 - The value of the relay 1 start mode is a single status change (Start/Timer).
- Timer P1 - The value of the status change of relay 1 over time, s (any number).
- N. impulse P1 - The value of the number of impulses - how many times relay 1 will change its status in the timer mode (-1 - infinite) (any number).
- Status P2 - Status value of relay 2 - changes status once in start mode (A / B).
- Mode P2 - The value of the relay 2 start mode is a single status change (Start/Timer).
- Timer P2 - The value of the status change of relay 2 over time, s (any number).
- N. impulse P2 - The value of the number of impulses - how many times relay 2 will change its status in the timer mode (-1 - infinite) (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR StateR1 - Set the status of relay 1: A ; B (string).
- BSTR ModeStartR1 - Set the relay start mode 1: Start; Timer (string).
- FLOAT timerR1 - Set the relay 1 start timer, s (number).
- FLOAT pulseNumR1 - Set the number of impulses for relay 1, (-1) - permanent (number).
- BSTR StateR2 - Set the status of relay 2: A ; B (string).
- BSTR ModeStartR2 - Set the start mode of relay 2: Start; Timer (string).

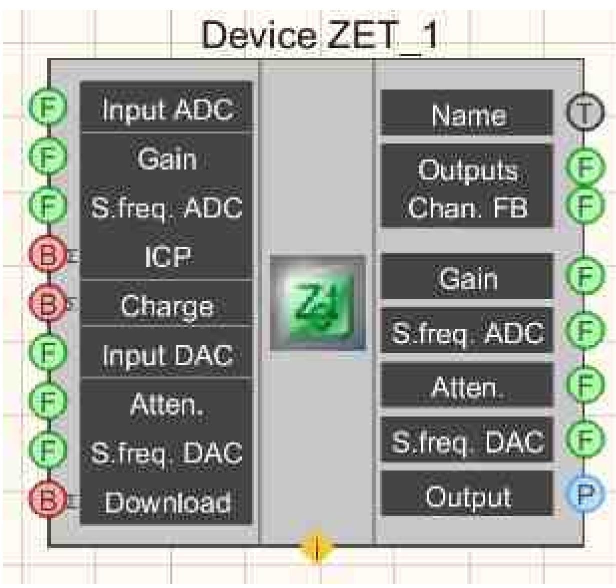
•FLOAT timerR2 - Set the relay 2 start timer, s (number).

•FLOAT pulseNumR2 - Set the number of impulses for relay 2, (-1) - permanent (number).
•LONG serialNumH - Set the high part of the serial number (number).
•LONG serialNumL - Set the low part of the serial number (number).

30.23.Device ZET

The component "**Device ZET**" is used for connection and setting parameters of ZET devices.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
➢ Input ADC - Set the current ADC channel.
• Gain - Set the channel gain.

- S.freq. ADC - Set the sampling frequency of the ADC.
- ICP - Set power supply via ICP per channel.
- Charge - Set to turn on the charge amplifier per channel.
- ➢ Input DAC - Set the current DAC channel.
- ➢ Atten. - Set the attenuation coefficient for the current DAC channel.
- S.freq. DAC - Set the sampling frequency of the DAC.
- Download - Device settings update is installed.
-

➢ Output:

- Name - The name of the device.
- Channels - the number of ADC channels of the device.
- Chan. FB - number of channels with device feedback.
- Gain - the gain factor is set for the current ADC channel.
- S.freq. ADC - Set the current sampling frequency of the ADC.
- Atten. - Set the current attenuation factor for the current DAC channel.
- S.freq. DAC - Set the current sampling frequency of the DAC is set.
- Output - the selected channel.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Device (0) - set the serial number of the device.
- DeviceType - set the device type: ZET-0xx-U2; ZET-017-xx; ZET-110; ZET-210; ZET-220; ZET-230; ZET-048.
- BuiltInGeneratorStatus (False) - Set the running status of the internal generator (if supported).
- BuiltInGeneratorFreq (1000.0) - Set the frequency of the Sine signal of the internal generator, Hz (if supported).
- BuiltInGeneratorLevel (1.0) - Set the level of the Sine signal of the internal generator, V (if supported).
- BuiltInGeneratorShift (0.0) - Set the offset of the DC component of the Sine signal of the internal generator, V (if supported).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input ADC - Set the current ADC channel for which you want to measure the required values (from 0 to (number of channels - 1)).
- Gain - Gain value of the measuring channel (from 1 to the maximum gain) .
- S.freq. ADC - Set the ADC sampling frequency(from the minimum ADC sampling frequency to the maximum ADC sampling frequency).
- Input DAC - Set the current DAC channel for which you want to measure the required values (from 0 to (number of channels - 1)).
- Atten. - Set the attenuation coefficient for the current DAC channel.
- S.freq. DAC - Set the DAC sampling frequency(from the minimum DAC sampling frequency to the maximum DAC sampling frequency).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR Device - Set the serial number of the selected device (string).
- BSTR DeviceType - Set the device type: ZET-0xx-U2; ZET-017-xx; ZET-110; ZET-210; ZET-220; ZET-230; ZET-048; (string).
- VARIANT_BOOL BuiltInGeneratorStatus - Set the frequency of the internal generator's Sine signal, Hz (if supported).
- FLOAT BuiltInGeneratorFreq - Set the frequency of the internal generator's Sine signal, Hz (if supported).
- FLOAT BuiltInGeneratorLevel - Set the internal generator sine signal level, V (if supported).
- FLOAT BuiltInGeneratorShift - Set the internal generator sine wave DC offset, V (if supported).

➢**Methods:**

- DOUBLE Amplify(LONG channum) - Returns the ADC channel gain factor.
- void Amplify(LONG channum, DOUBLE newVal) - Set the channel number and ADC channel gain, where:
  channum - new channel number;
  newVal - new channel gain value .
- DOUBLE FreqADC() - Returns the sampling frequency of the ADC .
- void FreqADC(DOUBLE newVal) - Set the sampling frequency of the ADC, where: newVal is the new value of the sampling frequency.

- DOUBLE FreqDAC() - Returns the sampling frequency of the DAC .
- void FreqDAC(DOUBLE newVal) - Set the sampling frequency of the DAC, where newVal is the new value of the sampling frequency.
- DOUBLE Atten(LONG channum) - Returns to the attenuation coefficient for the DAC channel.
- void Atten(LONG channum, DOUBLE newVal) - Set the channel number and DAC attenuation factor, where:

    channum - new channel number;
    newVal - new channel gain value .

30.24.Digital generator ZET7060g

The component "**Digital generator ZET7060g**" is used for generation of impulse signals with frequency range up to 100 mHz;

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:

- Frequency - generator signal frequency.
- Duty cycle - Set the duty cycle of the impulse signal.
- On\Off - turn the digital generator on and off.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- Frequency (240) - Frequency of generated impulses, Hz.
- Porosity (0.000960) - Duty cycle of generated impulses.
- Enable (False) - Enable/disable digital generator.
- SerialNumber (0x0000000000000000) - Serial number of the connected device, if only one ZET 7060 G module is connected to the computer, then the serial number can be omitted.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Frequency - Set the generator signal frequency, Hz (from 0.01 to 1000000 Hz).
- Duty cycle - Set the value of impulse signal duty cycle (from 0 to 1).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FLOAT Frequency - Set the frequency of generated pulses, Hz.
- FLOAT Porosity - Set the duty cycle of the generated pulses (number).

VARIANT_BOOL Enable - Set Enable\Disable the generator:

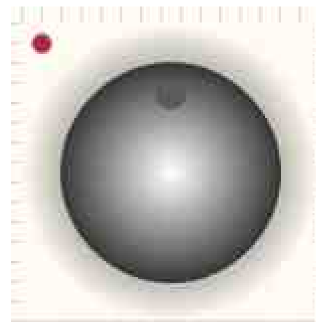
- true - the generator is enabled;
  false - the generator is disabled.

- BSTR SerialNumber - Set the serial number of the device (string).

30.25.Encoder ZET7060E

The component "**Encoder ZET7060E**" is used for angular and linear displacement, shaft rotation velocity, acceleration and position measurements. The data is transferred by RS-485 interface (Modbus protocol);

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:
- Input - a communication channel with a real encoder.

➢Output:
- The value is a control signal that the user specifies.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- BackColor (ffffff) - Set the background color of the selector.
- GradientBackColor (373737) - Set the color of the selector.
- selectorform(Sphere) - determines the form of the selector:
- 1. Cylinder;
- 2. Sphere.
- CursorForm (Circle) - Set the form of the cursor:
  1. Triangle;
  2. Circle.
- CursorColor (005eff) - Determines the color of the cursor (for the "With fill in" style).
- cursorstyle(Gradient) - Set the cursor style:

1. With fill in;
2. Gradient.

- shadeshade (Dark) - a shadow is set:
  1. No;
  2. Dark;
  3. Light.
- LinearVisible (true) - set the visibility of the ruler (True / False).
- Linearstyle(Different lengths) - the style of the ruler is determined:
  1. One length;
  2. Different length.
- LinearColor (a0a0a4) - set the color of the line.
- ValueOnLap (1) - Determines the change in value when turning 360 degrees.
- CurrentValue (0) - Set the Current value.
- StepMove (true) - Determines the movement in steps (On/Off).
- ValueInStep (0.05) - step discreteness is set (cannot be more than ValueOnLap / 8).
- MaxOutputValue (10,0000) - Set the maximum output value.
- MinOutputValue (-10.0000) - Set the minimum output value.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

•LONG BackColor - Set the background color of the selector (number).
•LONG GradientBackColor - Set the color of the selector (number).
•BSTR SelectorForm - Set selector form: Cylinder; Sphere (string).
•BSTR CursorForm - Set the cursor form: Triangle; Circle (string).
•LONG CursorColor - Set cursor color (for style \"With fill in\ (number).
•BSTR CursorStyle - Set cursor style: With fill in; Gradient (string).
•BSTR Shade - Shade setting: No; Dark; Light (string).
VARIANT_BOOL LinearVisible - Set ruler visibility:

- true - display of ruler visibility is enabled;
  false - display of ruler visibility is disabled.

- BSTR LinearStyle - Set the ruler style: One length; Different length (string).
- LONG LinearColor - Set the color of the line (number).
- FLOAT ValueOnLap - Set value to change when rotated 360 degrees (number).
- FLOAT CurrentValue - Set the current value (number).

VARIANT_BOOL StepMove - Set the movement in steps:

- true - enabled step by step.

false - disabled step by step.

- FLOAT ValueInStep - Set the step increment (from 0.01 to ValueOnLap / 8).
- FLOAT MaxOutputValue - Set the maximum output value (number).
- FLOAT MinOutputValue - Set the minimum output value (number).

Глава 31.File

Components of "File" clause are used for operations with files in SCADA-system ZETView:

- Select catalogue - allows the operator to select a folder and set corresponding directory for the components. Thus, the operator may assign a folder for automated saving of the reports.
- Select the file - is used for selecting the file name, shows the file directory.
- Calculation of hash - is used for to verify the integrity and authenticity of files by calculating the checksum.
- Recording in a file – allows to record data to a file in the course of project operation. Thus, it is possible to use dynamic data, that do not relate to project operation.
- Report - is used for recording various values to a file in accordance with user template.
- Search files in the directory- allows to find the files in particular category.
- Data base – allows to work with SQL data bases in SCADA-system ZETView.
- Access point (APN) - is used to select the current user (operator, engineer, etc. with corresponding access rights). The name of the selected user can be saved to database and be used for personal informational notification within the framework of complex systems, or be saved in reports made in the course of usual activities.
- Run program by name – allows to run programs (or files) from the project.
- Reading from a file - allows to read data from files in the course of project operation and to get access to dynamic data, that is not related to project operation.
- ZETLab directories – it allows to set ZETLab main directory, directories for recorded signals, data processing results, tooltips files, etc. This component

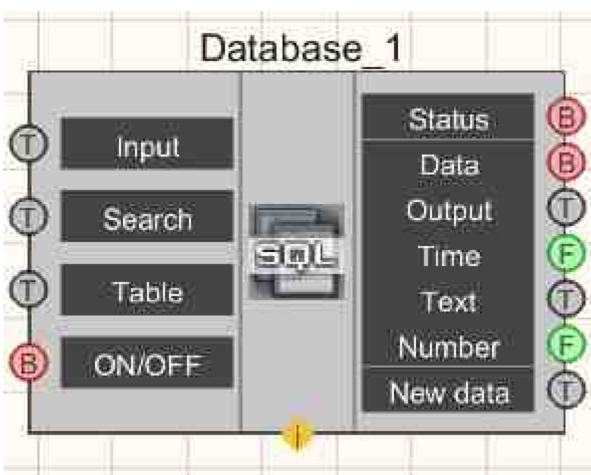
contributes to convenience of using standard ZETLab directories; it is used for report purposes, file operations, etc.

31.1.Database

The component "**Database**" - is used for allows you to work with SQL or MariaDB databases

There is more than one database option available for developers. MySQL is one of the most widely used databases across the world, but it is by no means the only one. Developed in C/C++, MySQL is free and open-source, and as such has made great headway
However, during the acquisition of Sun Microsystems by Oracle, some of the senior engineers who were working on the development of MySQL felt that there was a conflict of interest between MySQL and Oracle's commercial database - Oracle Database Server.
As a result, these engineers created a fork of the MySQL code base and started their own organization. This is how MariaDB was born.
Today, both databases are very popular and are used extensively by the developer community. MySQL is ranked second among the relational databases and second overall (Oracle database being first). MariaDB is behind - ninth among the relational databases and fourteenth overall.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢Input:

- Input - used to add a line to the database.
- Search - allows you to search for records in the database.
- Table - set the name in the table.
- ON / OFF - connect to the database / disconnect from the database.

➢Output:

- Status - connection status.
- Data - a new piece of data.
- Output - the value of the column.
- Time - timestamps.
- Text - the text of the selected line.
- Number - the number of the selected line.
- New data - new data.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- IP - IP address of the computer on which the database is installed.
- DSN - the name of the DSN data source.
- Database - the name of the database to which you want to connect.
- User - user (name entered during MySQL installation).
- Password - user password.
- Table - the table to which we are connecting.
- Delimiter (|) - text delimiter for database columns (fields).
- Column - the column whose values will be in the output.
- Timestamp - a column with timestamps.
- ErrorKey1 - Error 1 keyword.
- ErrorKey2 - Error 2 keyword.
- ErrorKey3 - Error 3 keyword.
- WarningKey1 - warning keyword 1.
- WarningKey2 - warning keyword 2.
- WarningKey3 - warning keyword 3.
- ErrorKey 1 Color (ff 0000) - error highlight color 1.
- ErrorKey 2 Color (ff 0000) - error highlighting color 2.
- ErrorKey 3 Color (ff 0000) - error highlight color 3.
- WarningKey 1 Color (ffff 00) - highlight color for warning 1.
- WarningKey 2 Color (ffff 00) - Warning 2 highlight color.
- WarningKey 3 Color (ffff 00) - highlight color for warning 3,
- Limit (100) - record limit.
- SearchLimit (1000) - search limit.

- PanelVisible (true) - visibility of the control panel.
- PrimaryKeys - primary keys.
- ShowPrimaryKeys - show primary keys.
- PrintEnabled (true) - the ability to print the report.
- ExcelEnabled (true) - the ability to export data to Excel .
- ColumnWidths - column width.
- AutoUpdate (true) - automatic data update.
- Servertype (MySQL) - type server of database.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values. Support for Firebird DBMS.

Setting:

➢Input:

- Input - The input value is used to add a row to the database (text).
- Search - The search value allows you to search for records in the database (text).
- Table - The table value sets the name in the table (text).

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR IP - Set the IP address of the computer on which the database is installed (string).
- BSTR DSN - Set the name of the DSN data source (string).
- BSTR Database - Set and replace the name of the database to which you want to connect (string).
- BSTR User - User setting (name entered during MySQL installation) (string).
- BSTR Password - Set user password (string).
- BSTR Table - Setting the name of the table to which we are connecting (string).
- BSTR Delimiter - Set the text delimiter of the columns (fields) of the database (string).
- BSTR Column - Set the column whose values will be output (string).
- BSTR Timestamp - Set the timestamp column (string).
- BSTR ErrorKey 1 - Set the error keyword 1 (string).

- BSTR ErrorKey 2 - Set the error keyword 2 (string).
- BSTR ErrorKey 3 - Set the error keyword 3 (string).
- BSTR WarningKey 1 - Set warning keyword 1 (string).
- BSTR WarningKey 2 - Set warning keyword 2 (string).
- BSTR WarningKey 3 - Set warning keyword 3 (string).
- LONG ErrorKey 1 Color - Sets the highlight color for error 1 (number).
- LONG ErrorKey 2 Color - Set the highlight color for error 2 (number).
- LONG ErrorKey 3 Color - Sets the highlight color for error 3 (number).
- LONG WarningKey 1 Color - Sets the highlight color for warning 1 (number).
- LONG WarningKey 2 Color - Sets the highlight color for warning 2 (number).
- LONG WarningKey 3 Color - Sets the highlight color for warning 3 (number).
- LONG Limit - Set the limit of entries (number).
- LONG SearchLimit - Set search limit (number).
- VARIANT_BOOL PanelVisible - Set the visibility of the control panel:
  true - the output value is given to display the control panel;
  false - the output value is to not show the control panel.
- BSTR PrimaryKeys - Set the primary key (string).
- VARIANT_BOOL ShowPrimaryKeys - Set the display of primary keys:
  true - the output value is given to display the primary keys;
  false - the output value is returned not to display primary keys.
- VARIANT_BOOL PrintEnable - Set the ability to print the report:
  true - output value of report printing capability;
  false - the output value of not being able to print the report.
- VARIANT_BOOL ExcelEnable - Set the ability to upload data to Excel :
  true - the output value of the ability to export data to Excel ;
  false - the output value of not being able to upload data to Excel .
- BSTR ColumnWidths - Set the width of the columns (string).
- VARIANT_BOOL AutoUpdate - Set automatic data update:
  true - the ability to automatically update data is enabled;
  false - the ability to automatically update data is disabled.
- BSTR ServerType - Set the database server support: MySQL / MariaDB ; Firebird / Interbase (string).

Installing and setting a MySQL database

Contents

On the PC, run the installer file “mariadb-10.0.5-win32.msi” or “mariadb-10.0.5-winx64.msi” (depending on the bitness of the installed version of Windows OS), then in the “MariaDB (1)” window (Fig. 1.1 1169) activate "Next".

Fig. 1.1 "MariaDB(1)" window

In the "MariaDB (2)" window (Fig. 1.2 1169) activate "Next".

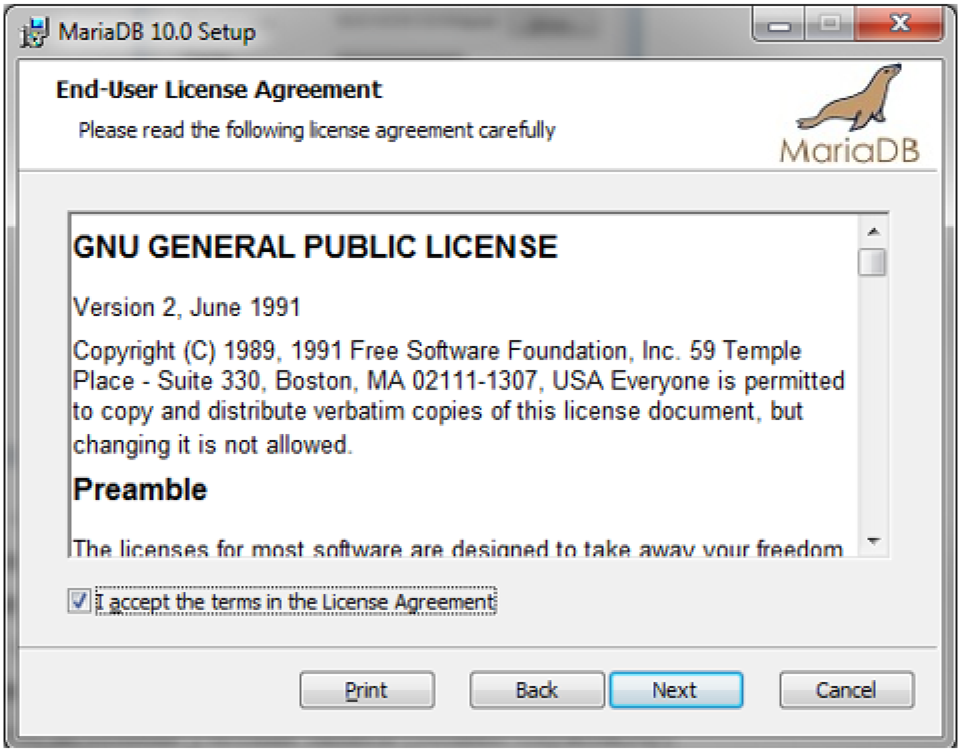


Fig. 1.2 "MariaDB(2)" window

In the "MariaDB (3)" window (Fig. 1.3 1169) activate "Next".

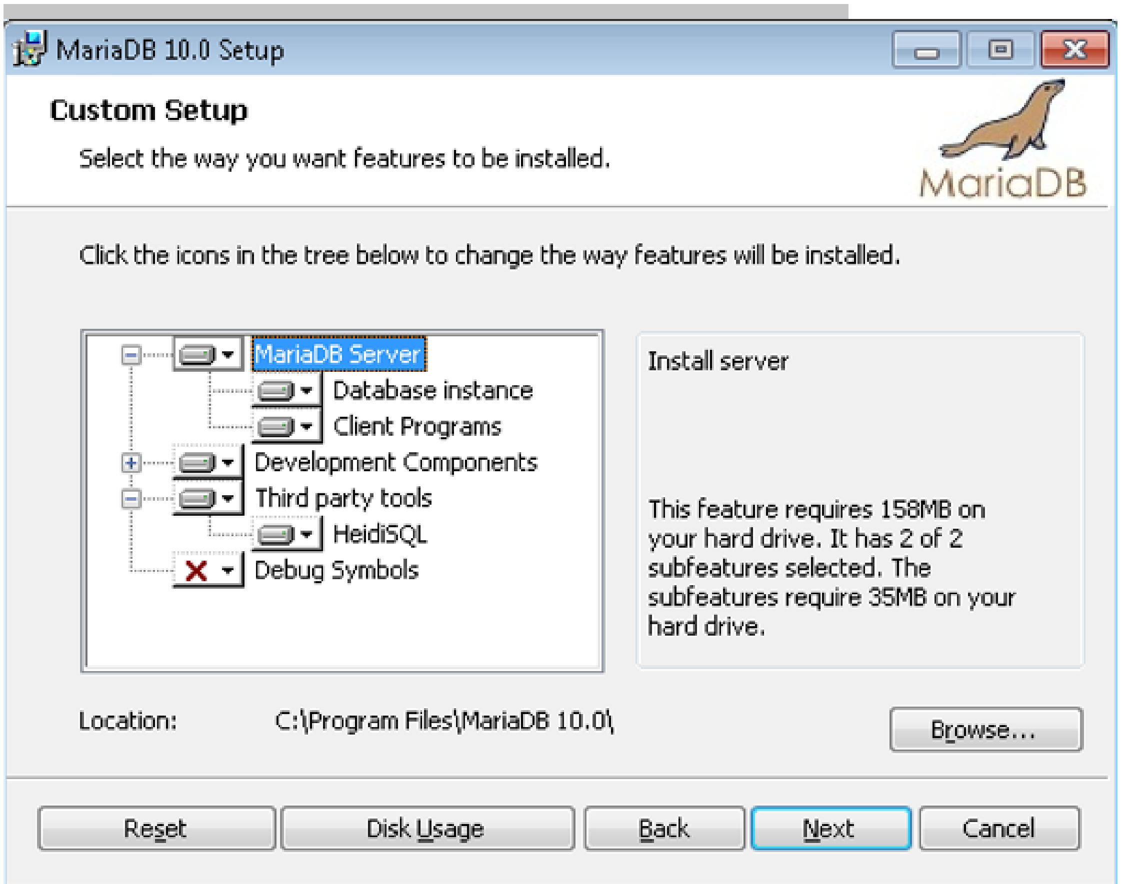


Fig. 1.3 "MariaDB(3)" window

In the “MariaDB (4)” window (Fig. 1.4 1169) in the “New root password” field, enter the password for root, and then activate “Next”.

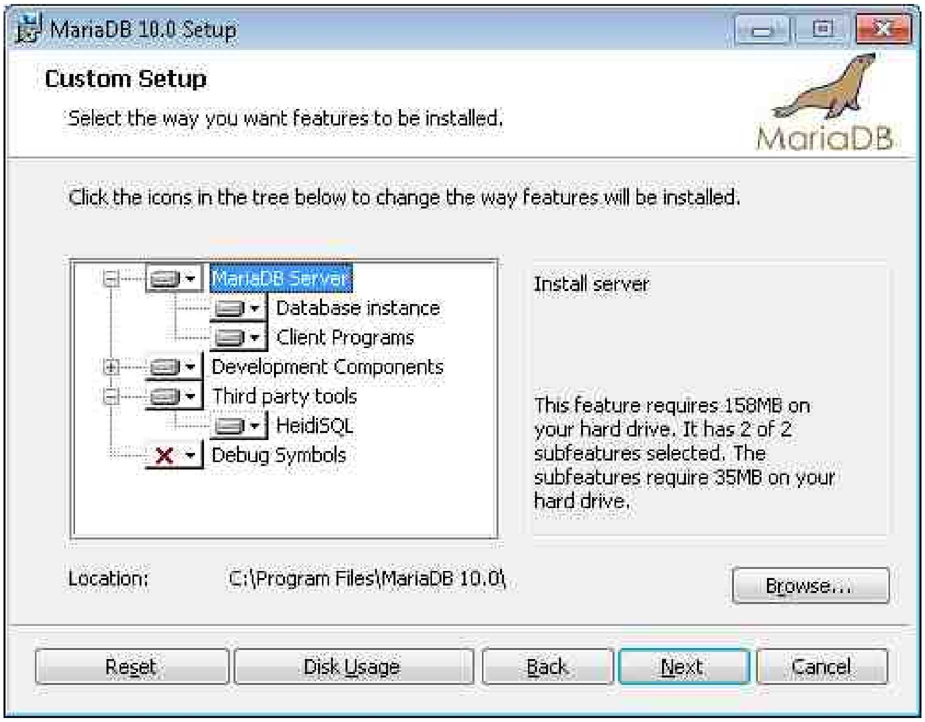


Fig. 1.4 "MariaDB (4)" window

In the "MariaDB (5)" window (Fig. 1.5 1169) in the "New root password" field, enter the password for root, and then activate "Next".

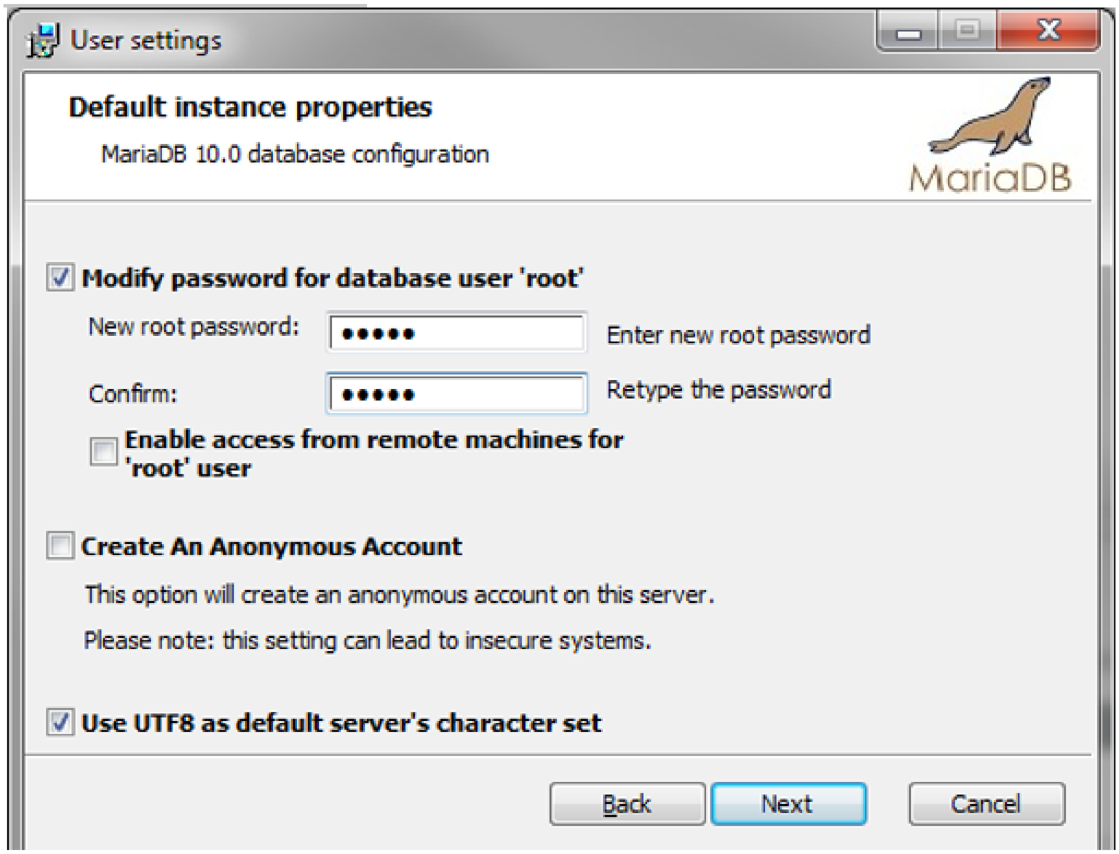


Fig. 1.5. MariaDB (5) window

MariaDB (6)" window, enter the field values in accordance with the example shown (Fig. 1.6 1169), and then activate " Next ".

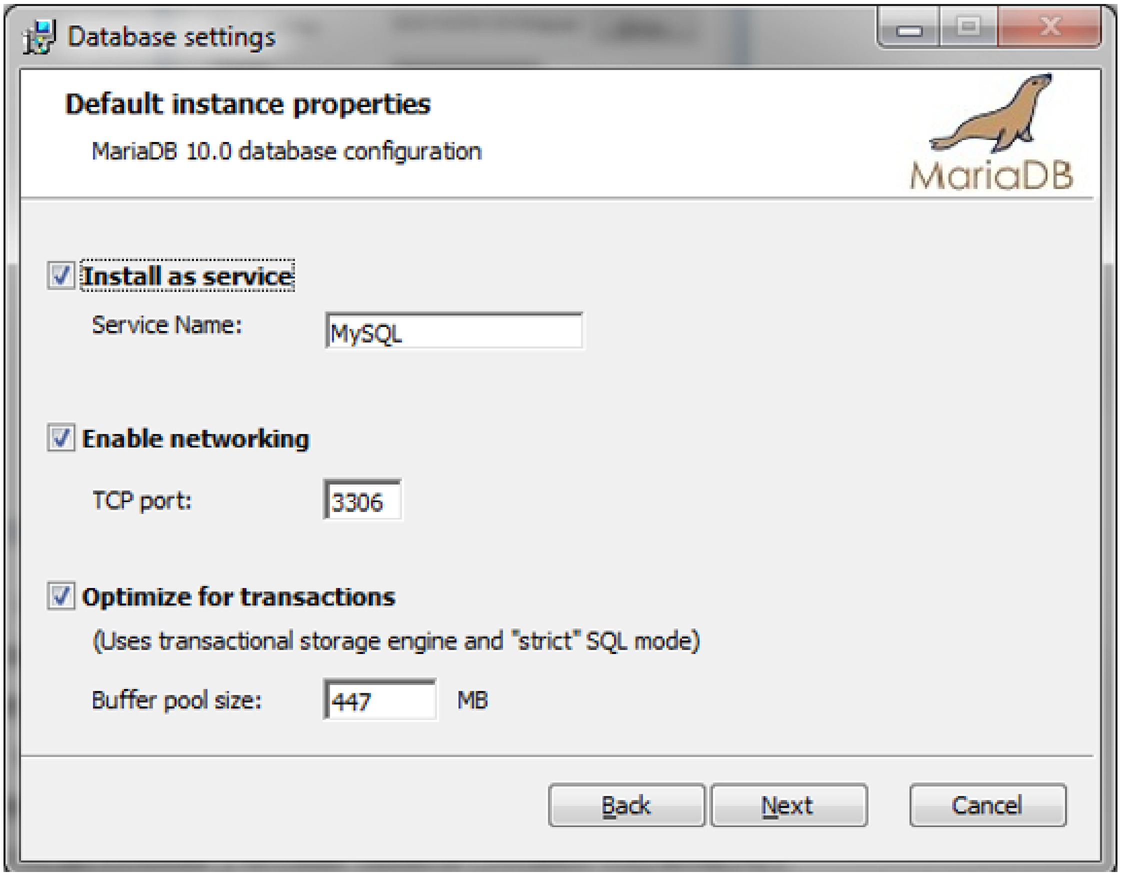


Fig. 1.6. MariaDB (6) window

In the "MariaDB (7)" window (Fig. 1.7 1169) activate "Next".

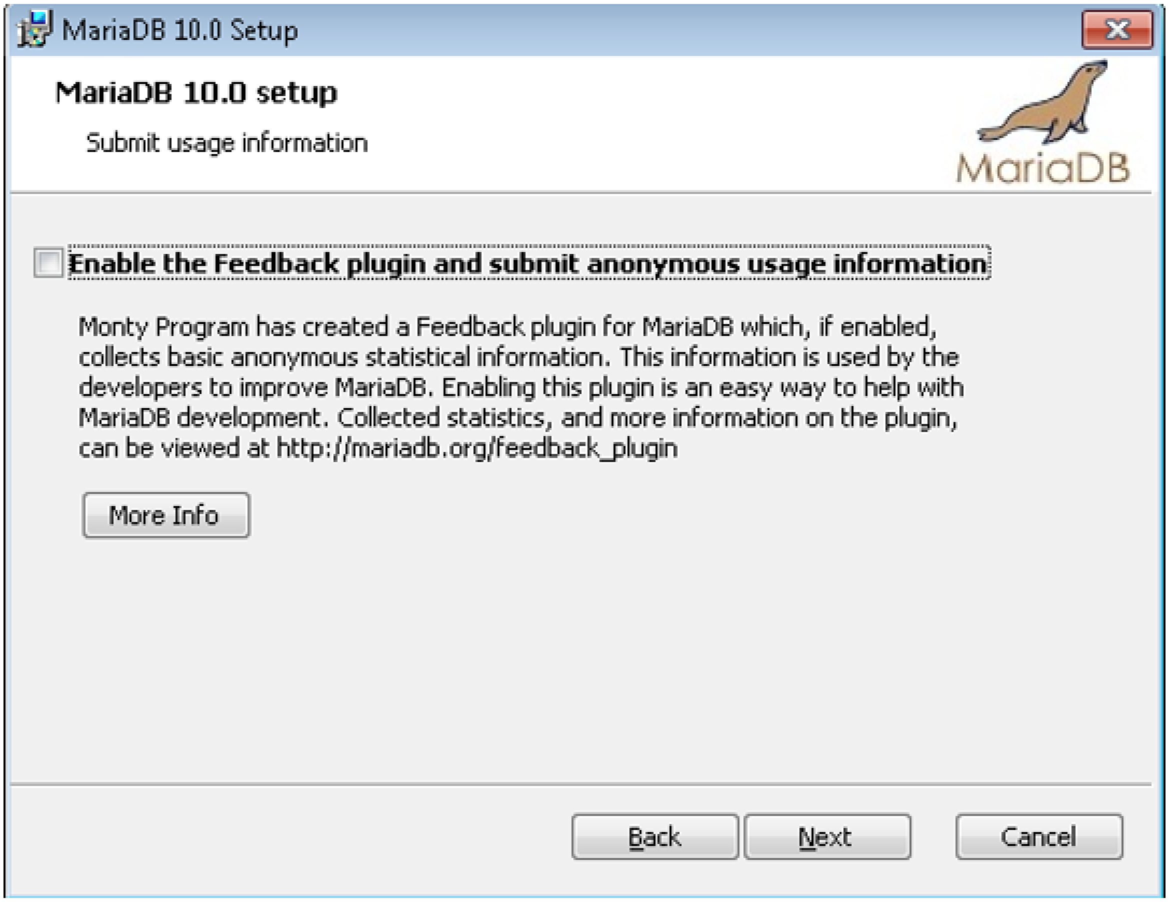


Fig. 1.7 "MariaDB (7)" window

In the "MariaDB (8)" window (Fig. 1.8 1169) activate "Install".

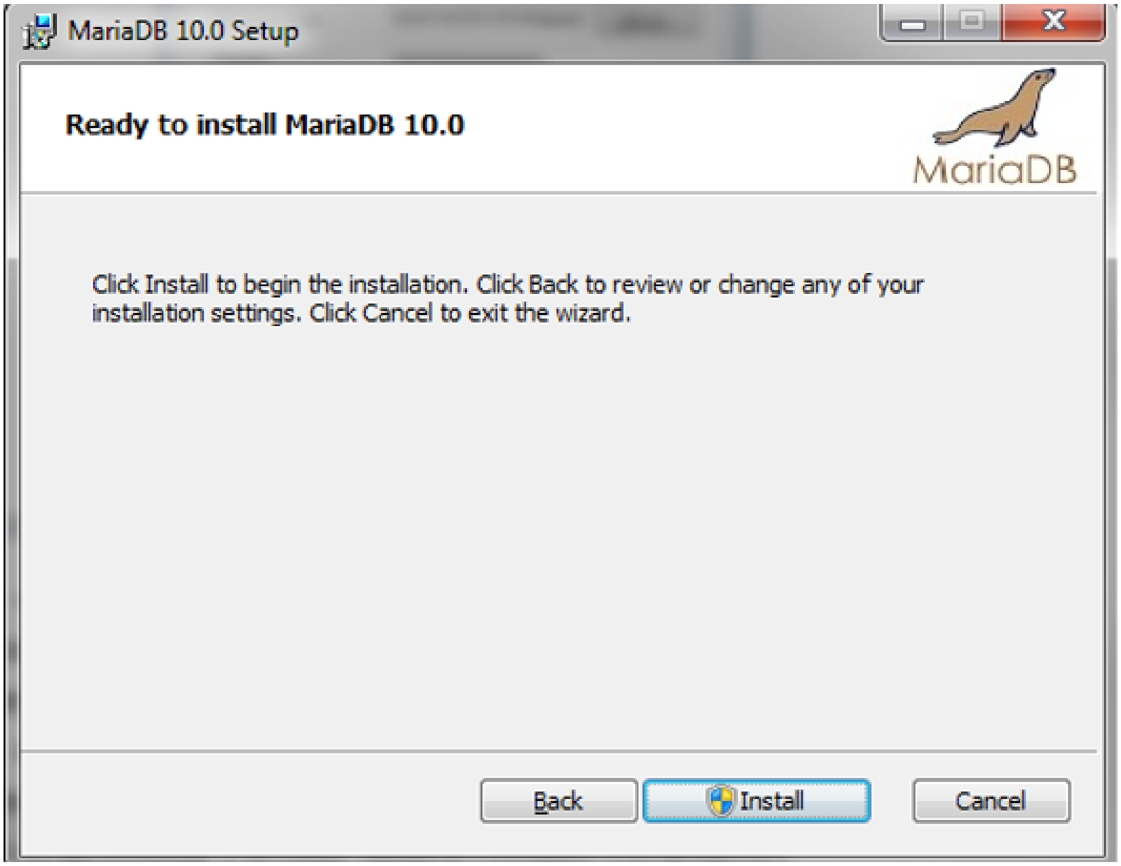


Fig. 1.8 "MariaDB (8)" window

In the "MariaDB (9)" window (Fig. 1.9 1169) activate "Finish".

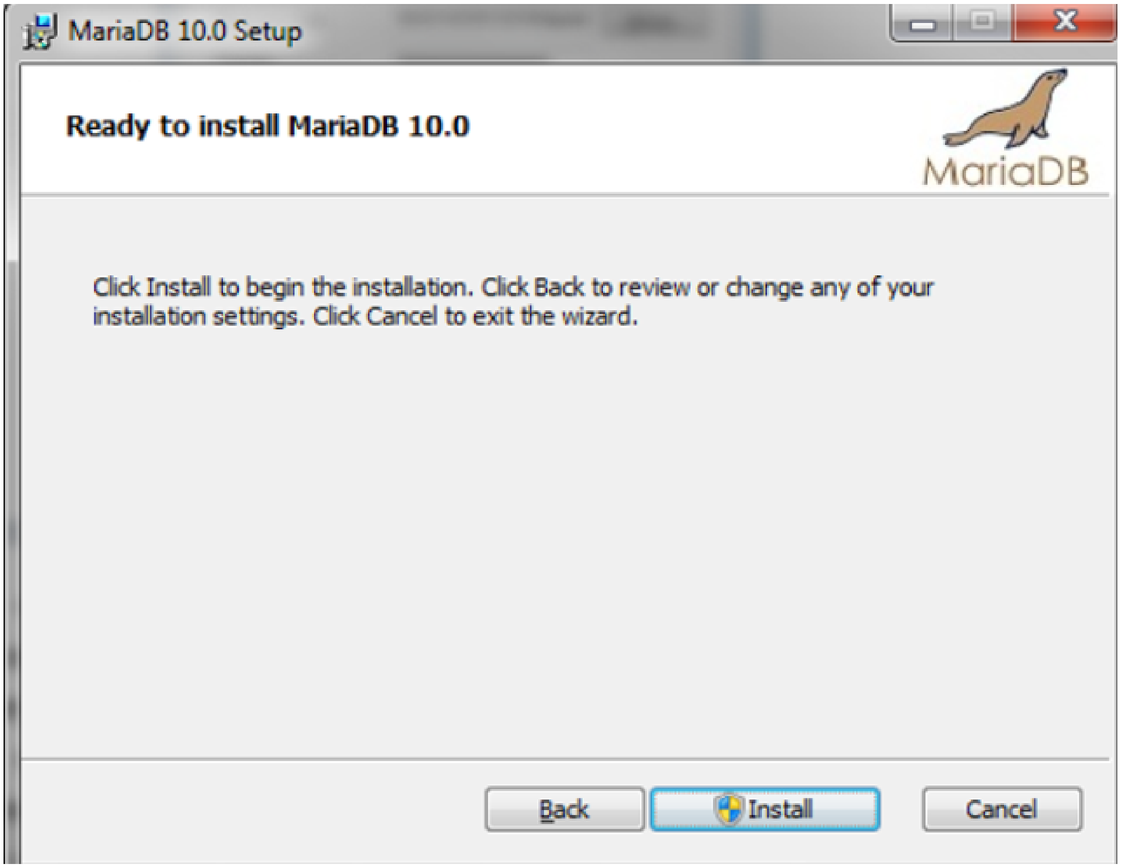


Fig. 1.9 "MariaDB (9)" window

Run the installer file “mysql-connector-odbc-5.1.11-win32.msi” on the PC, and then activate “Next” in the window “MySQL Connector/ODBC 5.1 (1)” (Fig. 5.82). 1178Attention: the installer file " mysql - connector - odbc -5.1.11- win 32. msi " should be run for both 32-bit and 64-bit versions of the operating system

Fig. 1.10 "MySQL Connector/ODBC 5.1 (1)" window

In the "MySQL Connector / ODBC 5.1 (2)" window (Fig. 1.11 1178) select "I accept the terms in the license agreement" and then activate "Next".

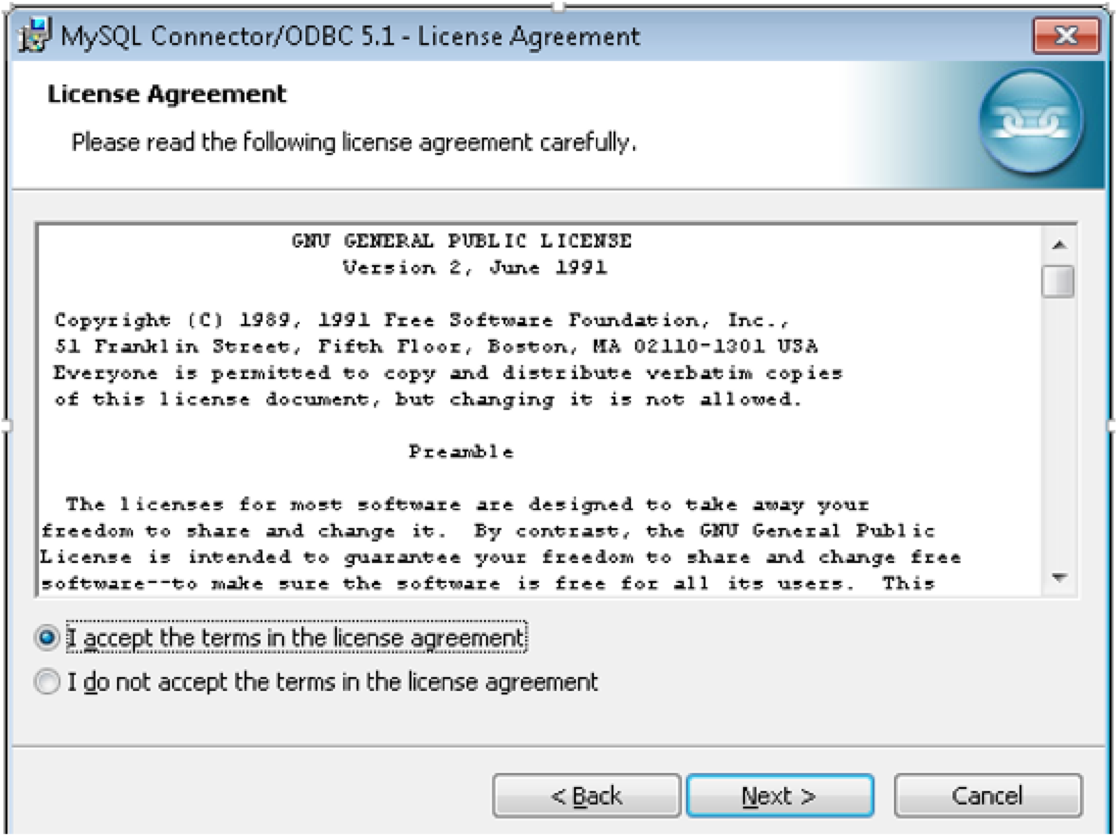


Fig. 1.11 "MySQL Connector/ODBC 5.1 (2)" window

In the "MySQL Connector / ODBC 5.1 (3)" window (Fig. 1.12 1178) select "Typical" and then activate "Next".

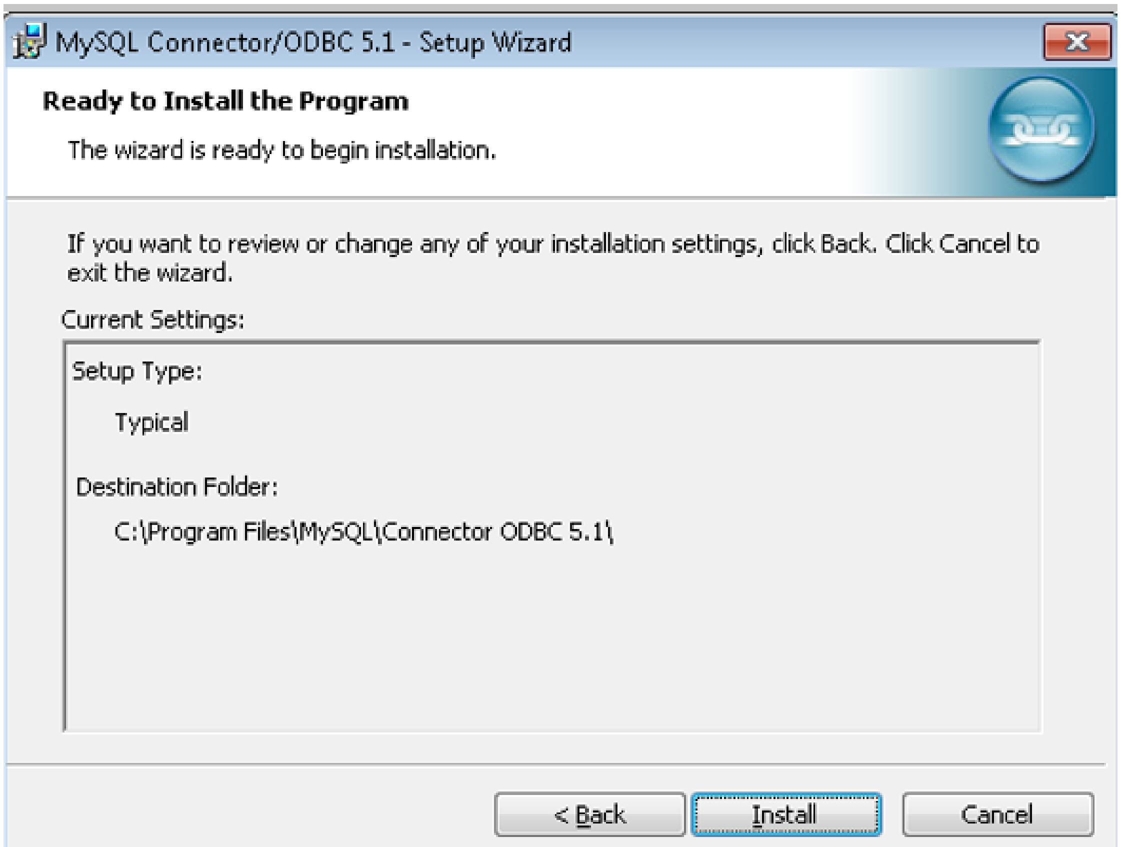


Fig. 1.12 "MySQL Connector/ODBC 5.1 (3)" window

In the "MySQL Connector/ODBC 5.1 (4)" window (Fig. 1.13 1178) activate "Install".

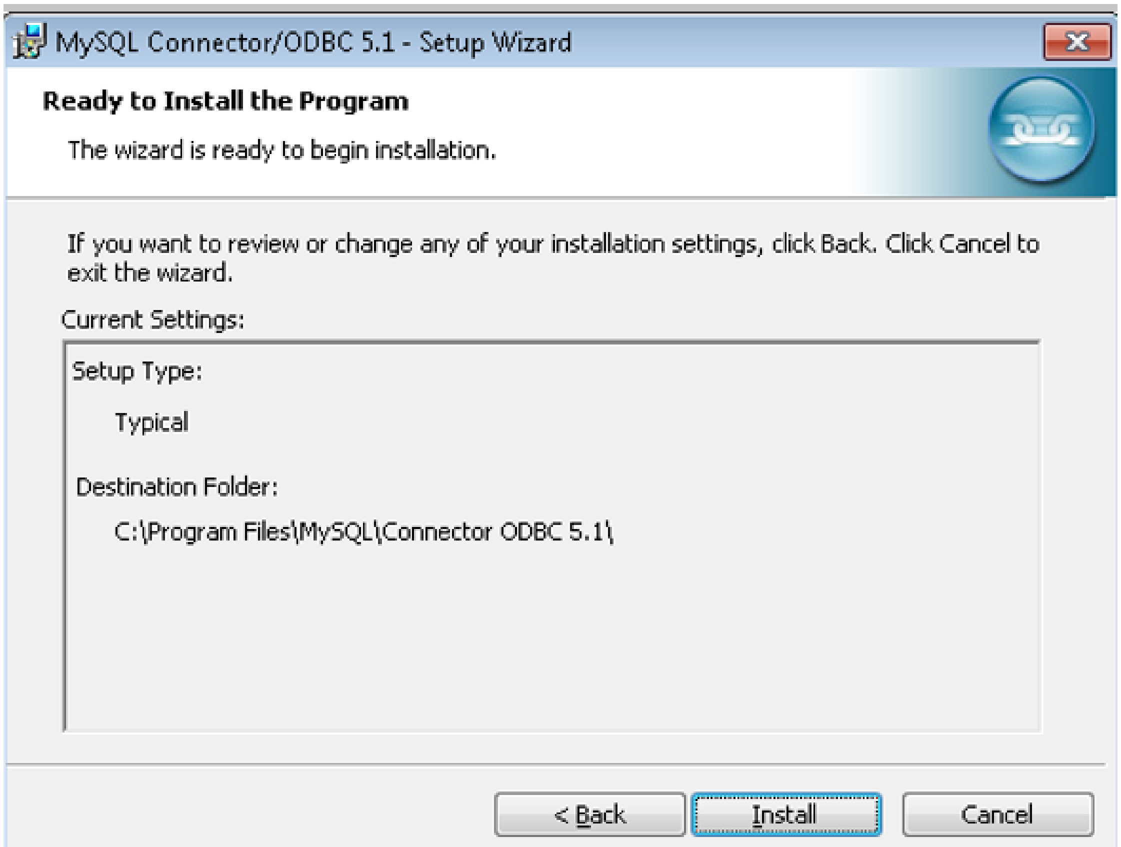


Fig. 1.13 "MySQL Connector/ODBC 5.1 (4)" window

In the "MySQL Connector/ODBC 5.1 (5)" window (Fig. 1.14 1178) activate "Finish".

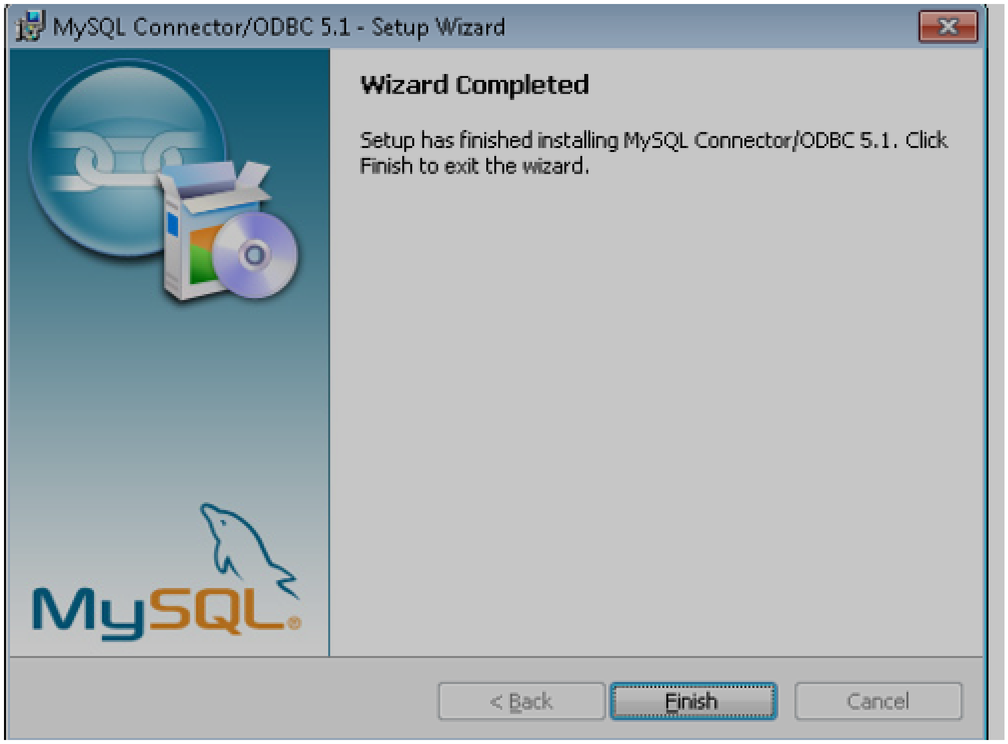


Fig. 1.14. MySQL Connector/ODBC 5.1 (5) window

From the system window "Run" (called by the command " Win + R ") run the program " odbcad 32.exe " (Fig. 1.15 1178), and in case of using a 64-bit version of the operating system, run the program " odbcad 32.exe " from the directory C :\ Windows \ SysWOW64 \.

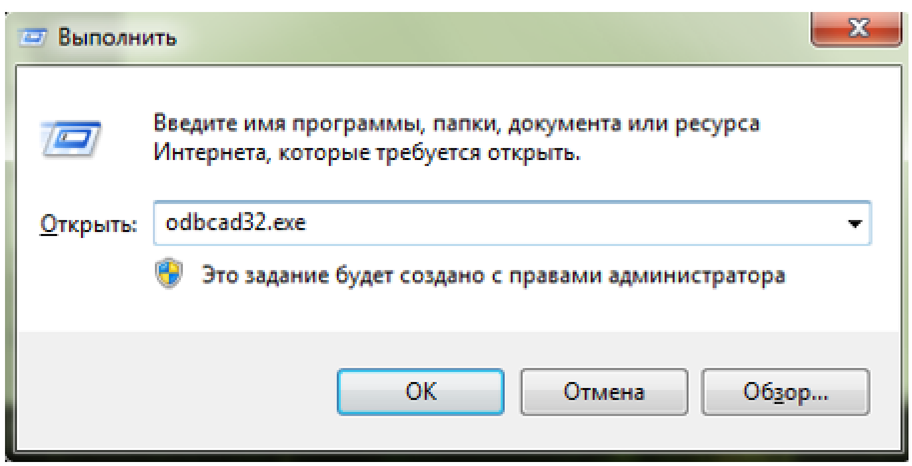


Fig. 1.15 Launching the program " odbcad 32.exe "

ODBC Data Source Administrator " on the tab " System DSN "Make sure that the driver" MySQL ODBC 5.1 Driver ". If this driver is missing, then select the "Add" command (Fig. 1.16 1178).

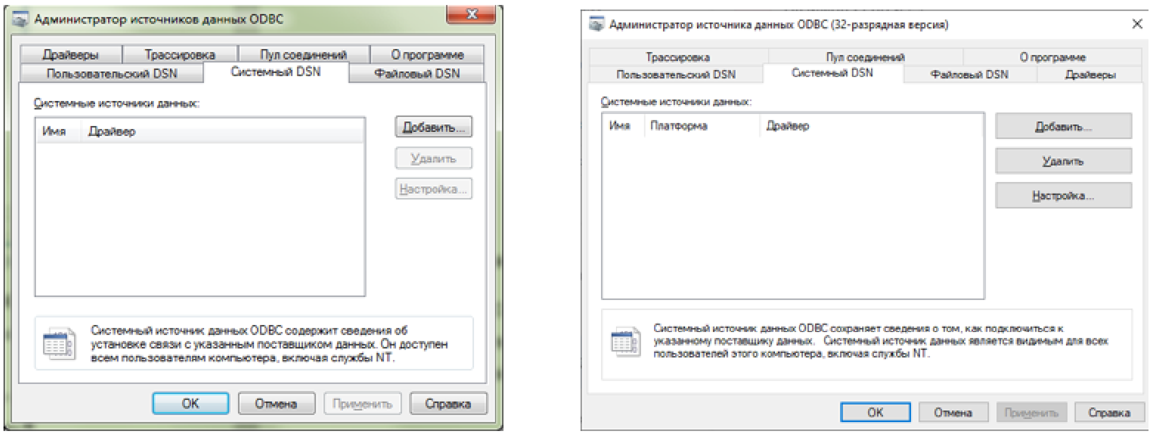


Fig. 1.16 ODBC Data Source Administrator program window

In the "Create a new data source" window, select the " MySQL" driver ODBC 5.1 Driver " and click "Finish" (Fig. 1.17 1178).

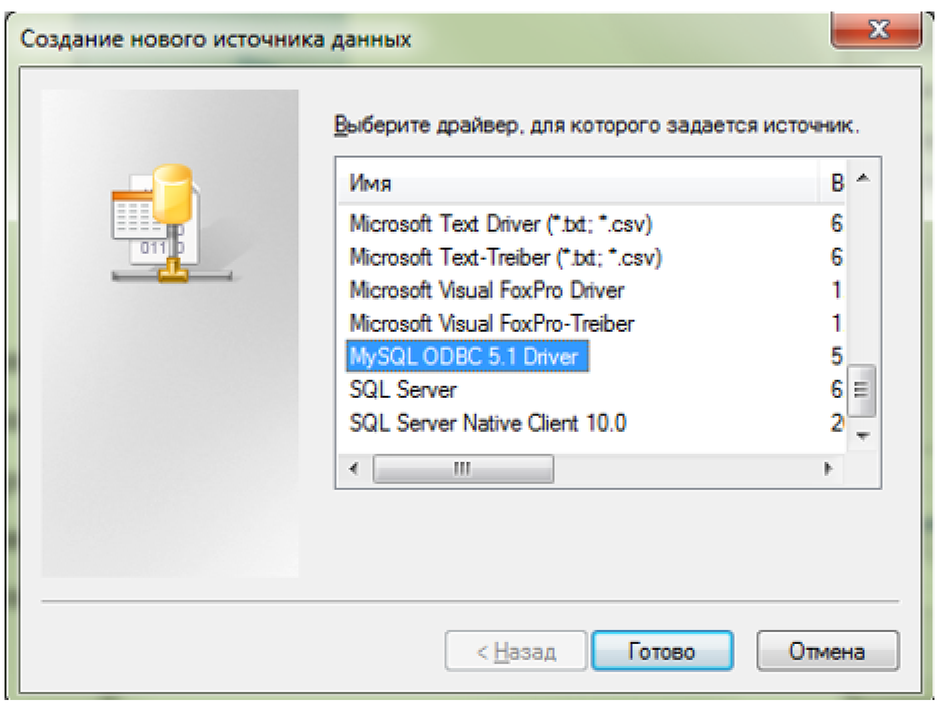


Fig. 1.17 Selecting the MySQL ODBC 5.1 Driver

In the opened MySQL Connector / ODBC Data Source Configuration » to configure the driver in accordance with Fig. 1.18 1178.

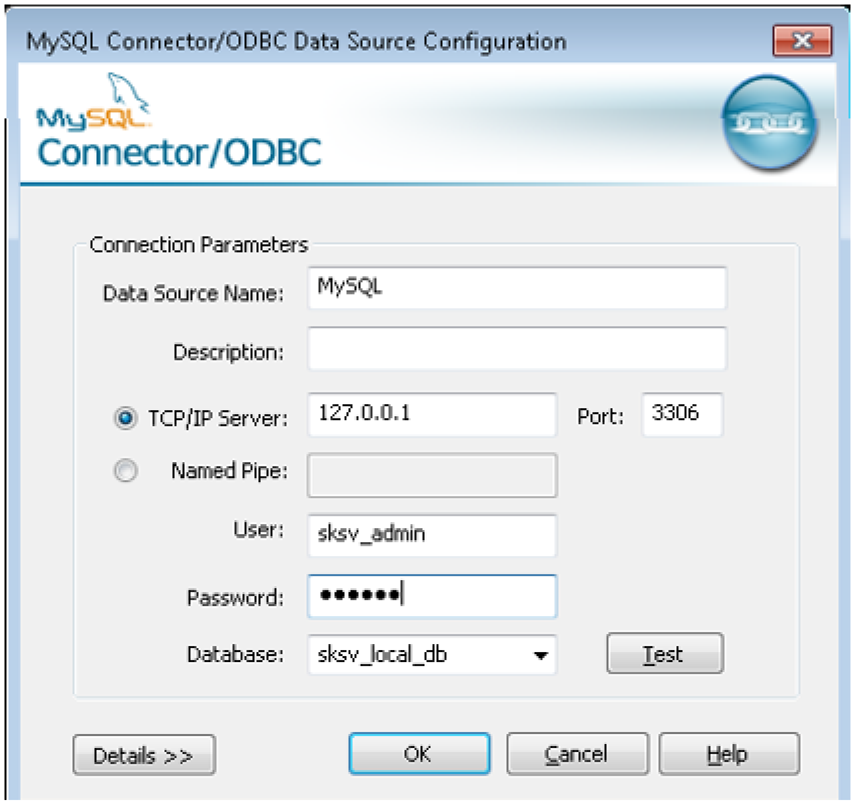


Fig. 1.18 MySQL driver setup ODBC 5.1 Driver »

To check the correctness of the settings made, click the "Test" button. If the test is successful, the program will display a message (Fig. 1.19 1178).

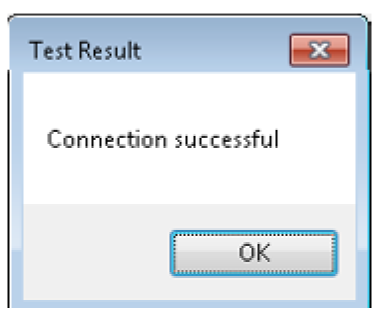


Fig. 1.19 Success message

To save the settings, click the "OK", "Apply" buttons.

When using a 64-bit version of the operating system, the above steps are performed in the same way, only the odbcad32.exe application is launched from the C:\Windows\SysWOW64\ directory.

Installing and setting a MySQL database

Contents

MySQL binaries and installer can be downloaded from https :// dev . mysql . com / downloads / mysql /5.1. html 1186 The instructions in this article are based on MySQL version 5.1 Community Edition installed with the MSI Installer for Windows .

Run the installer and select the installation option. For most cases, a typical installation will suffice:

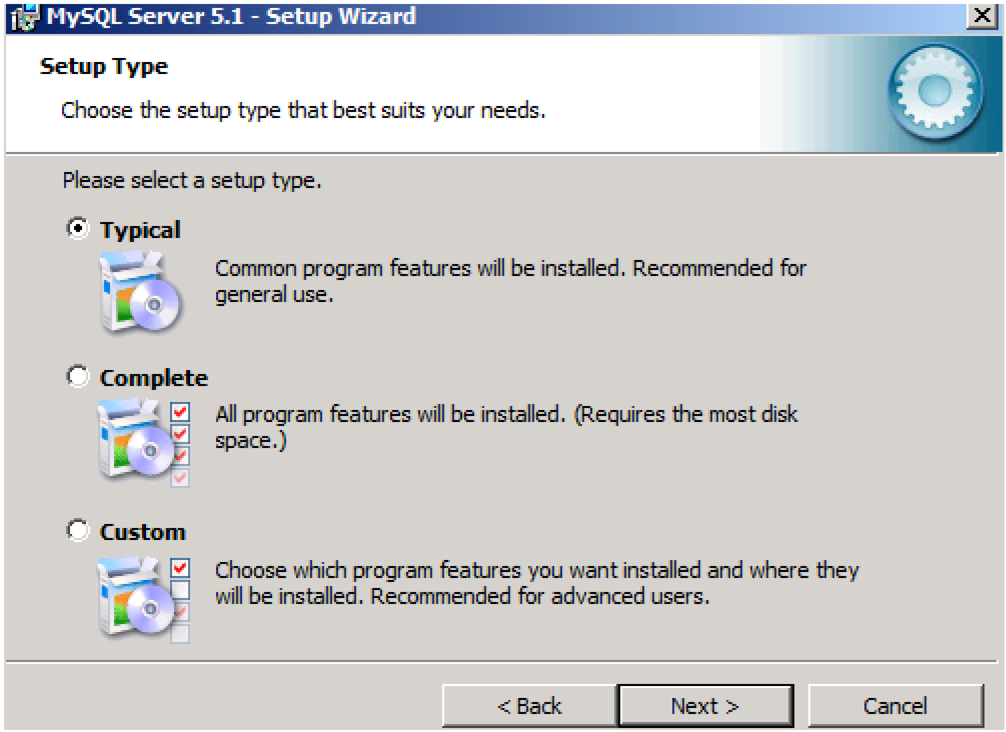


Fig. 1.1 MySQL (1) window

However, if you want to manage the installation of components or use an installation path other than the default one, select the Custom option (Configurable) *Fig. 1.2 Window «MySQL (2)»* window.

After the installation is complete, select the "Configure MySQL Server now" checkbox. The MySQL Server Instance Setup Wizard opens, which will help you configure the MySQL instance.

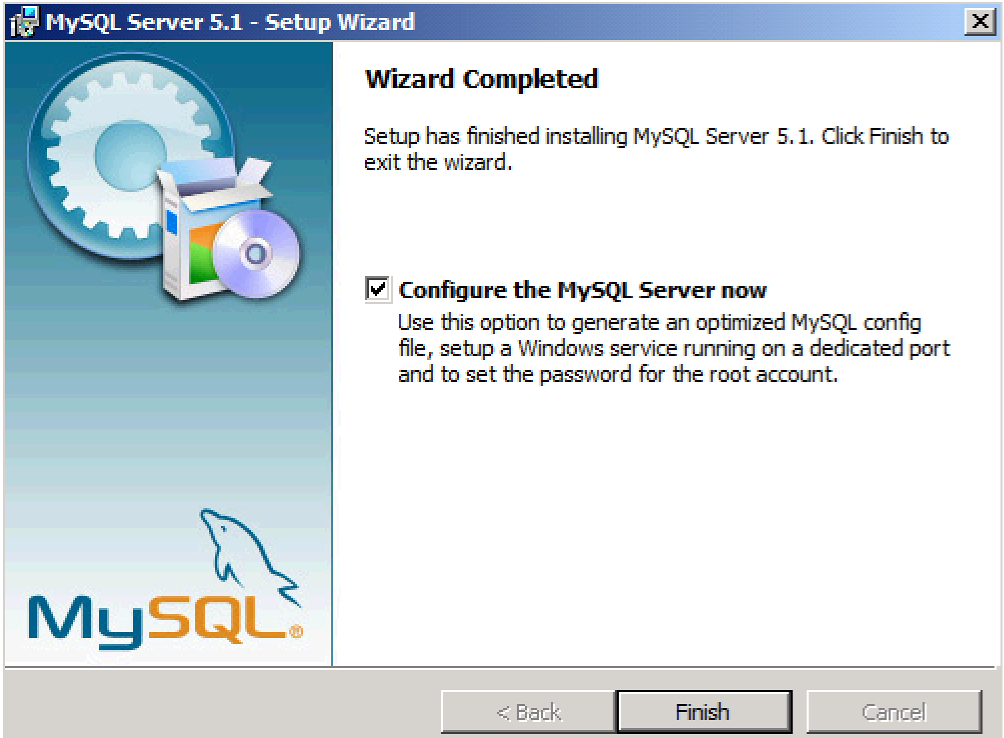


Fig. 1.2 «MySQL (2) window»

Complete the following steps in the MySQL Server Instance Configuration Wizard to optimize your MySQL configuration for the tasks you expect to perform.

On the first page of the wizard, select "Detailed configuration" *Fig. 1.3 MySQL (3)* window:

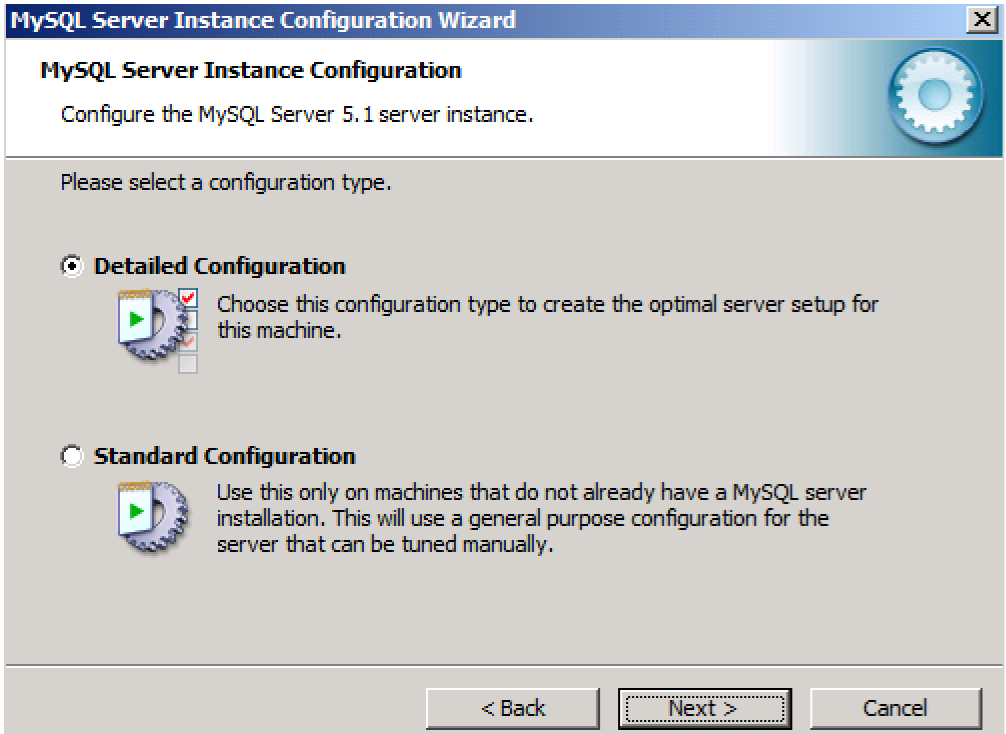


Fig. 1.3 MySQL (3) window

On the next page, select the server type option *Fig. 1.4 MySQL (4)* window:

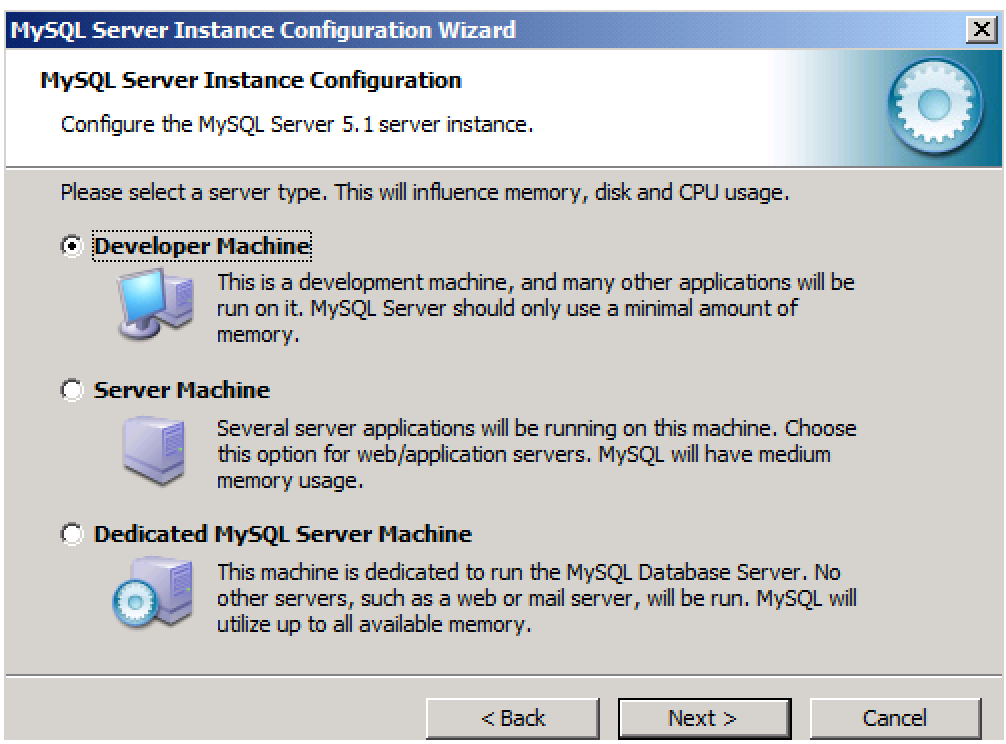


Fig. 1.4 MySQL (4) window

Select the Database Usage option *Fig. 1.5 MySQL (5)* window:

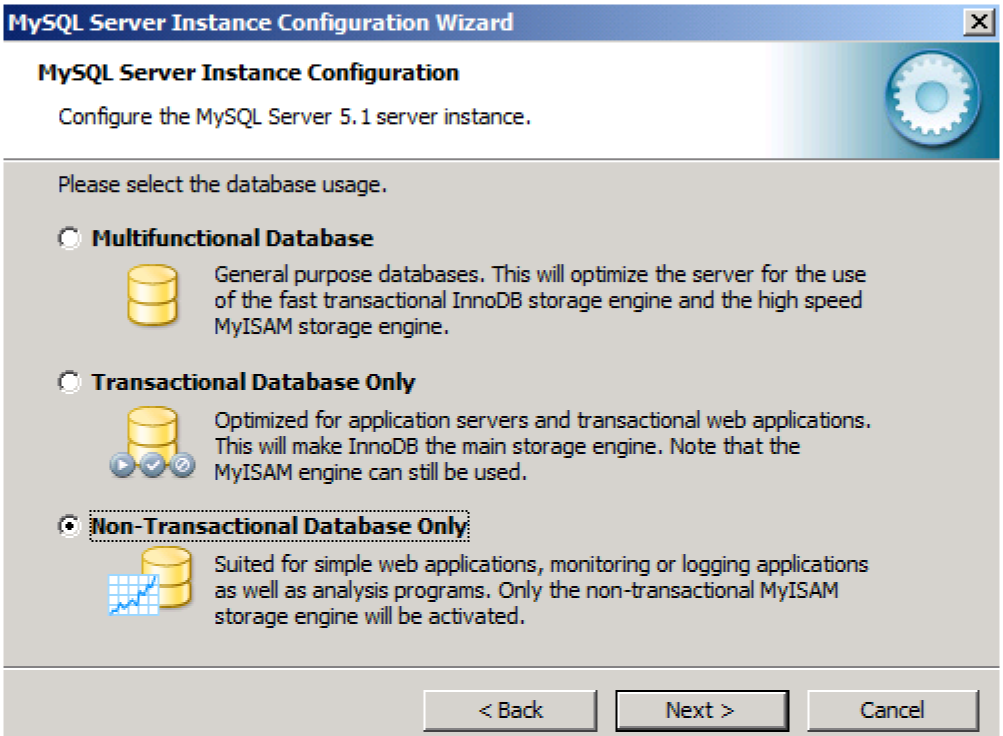


Fig. 1.5 MySQL (5)

The database usage settings determine what type of database storage engine is used on the server:

- MyISAM - Optimized for high performance SELECT operations . It has low overhead in terms of memory usage and disk usage, but at the expense of not supporting transactions.
- InnoDB - Provides full ACID transaction capabilities , but at the cost of more aggressive disk space and memory usage

For a detailed comparison of these subDs, see the MySQL Storage Engine Architecture section 1188. As a general recommendation. If web applications on the server require multi-factor transactions, advanced isolation levels and row-level locking, foreign key constraints, or other requirements for ACID functionality , use InnoDB . Otherwise, use MyISAM

Then select the number of concurrent connections to the server *Fig. 1.6 MySQL (6)* window:

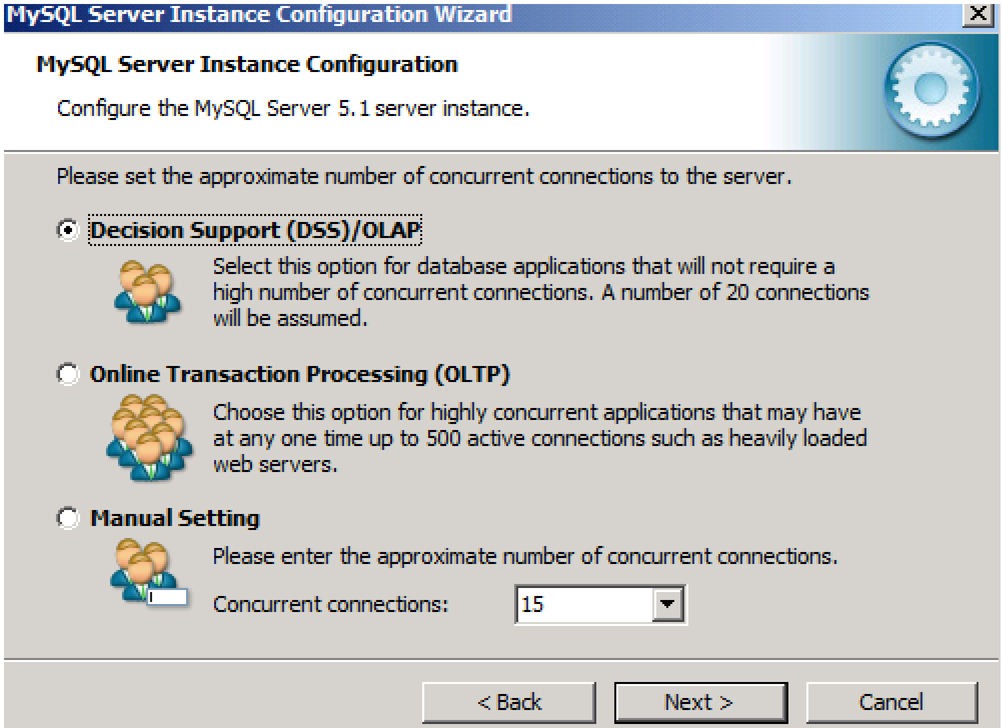


Fig. 1.6 MySQL (6)

On the next page, select the network settings *Fig. 1.7 MySQL (7)* window:

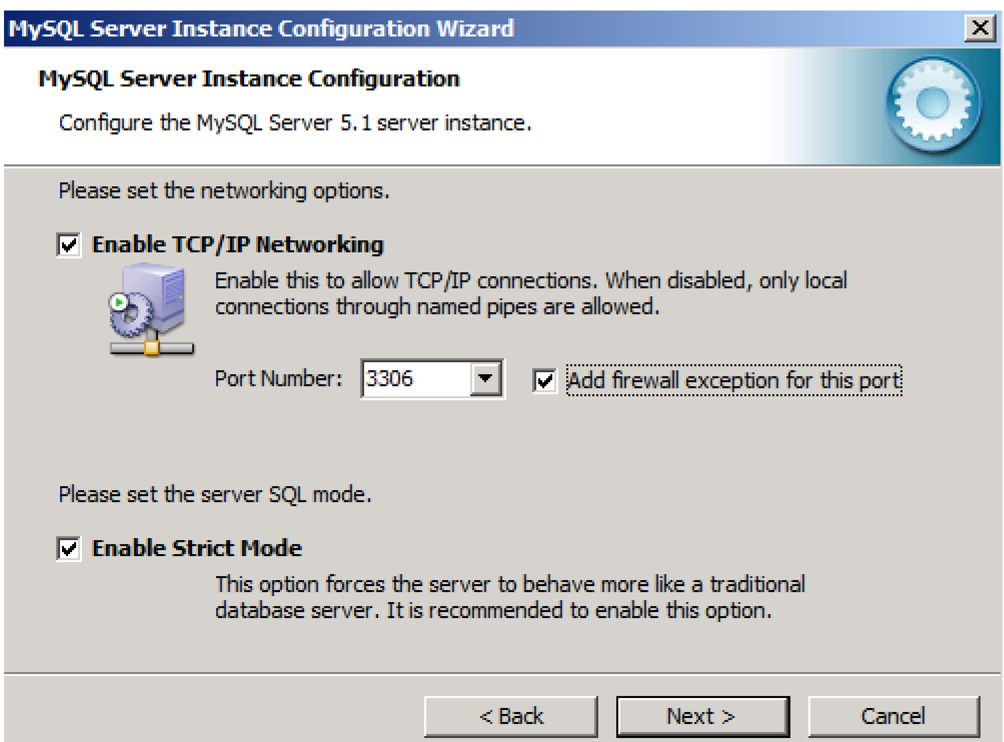


Fig. 1.7 MySQL (7) window

If you have mysql and a webserver on the same machine, you may not enable TCP / IP networking and use named pipes instead. Note that some PHP applications may require a TCP connection to MySQL . Check the application's documentation to see if it supports named pipe connection to MySQL .

Select the default encoding used when creating new databases *Fig. 1.8 MySQL (8)* window:

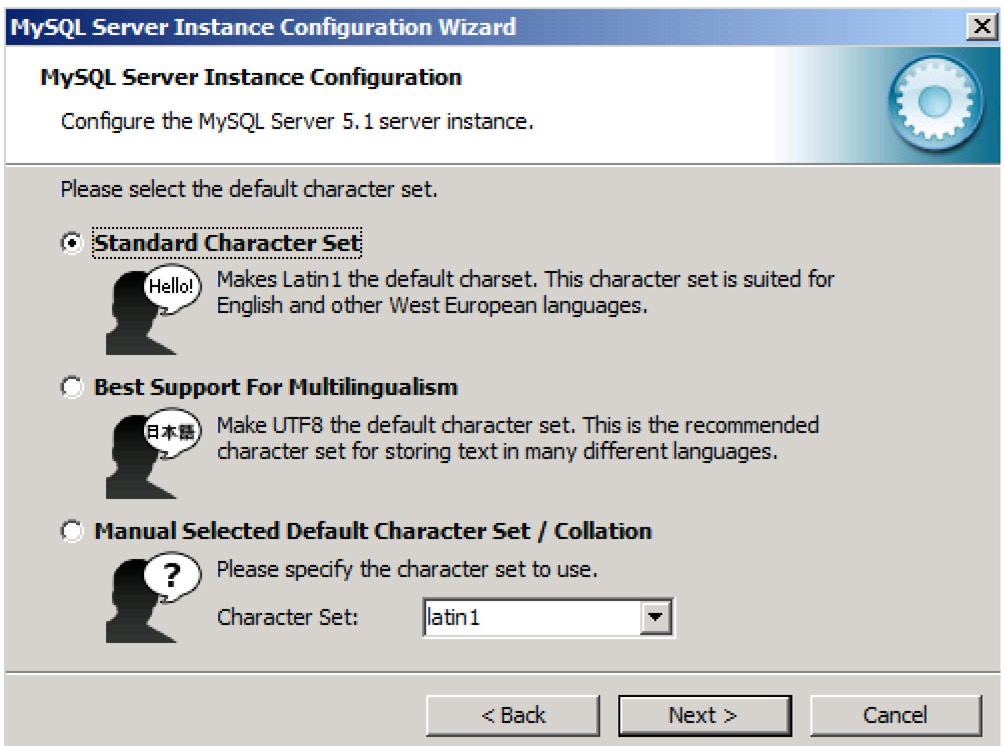


Fig. 1.8 MySQL (8) window

Then make sure MySQL will be configured as a windows service *Fig. 1.9 MySQL (9)* window:

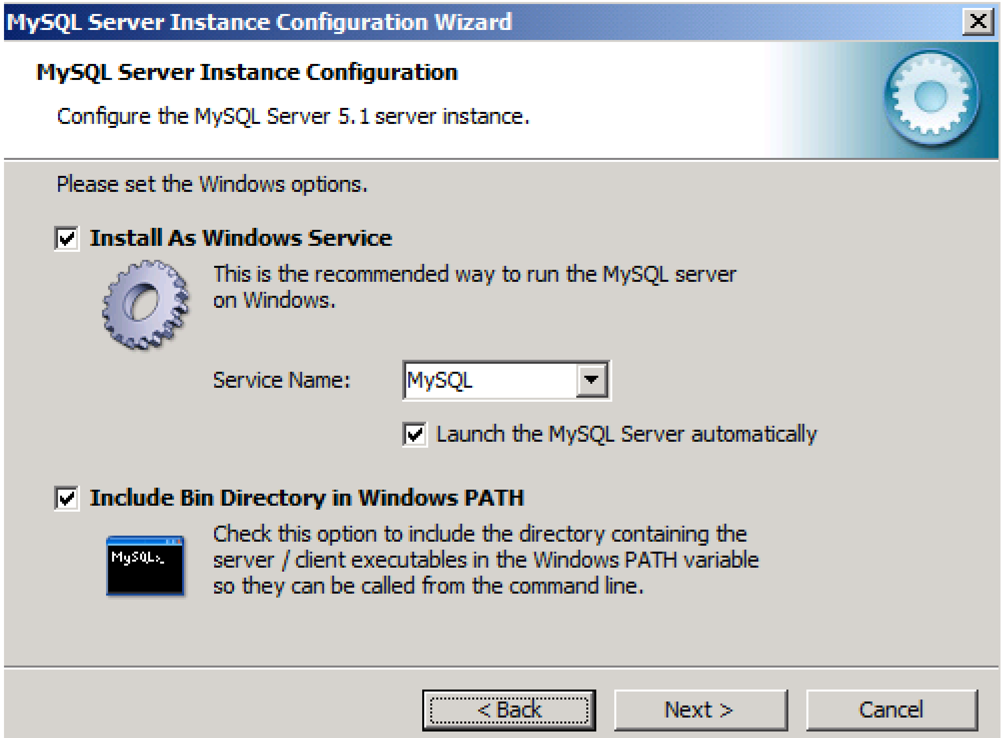


Fig. 1.9 MySQL (9) window

Optionally, you can add a bin directory MySQL to Windows environment variable PATH . This makes it easier to run MySQL tools from the command line.

Finally, provide a password for the database administrator account, which is called root in MySQL . Make sure "Create Anonymous Account" is unchecked *Fig. 1.10 MySQL (10)* window:

Fig. 1.10 MySQL (10) window

On the next page, click the "Run" button to apply all configuration options and start the MySQL service *Fig. 1.11 MySQL (11)* window:

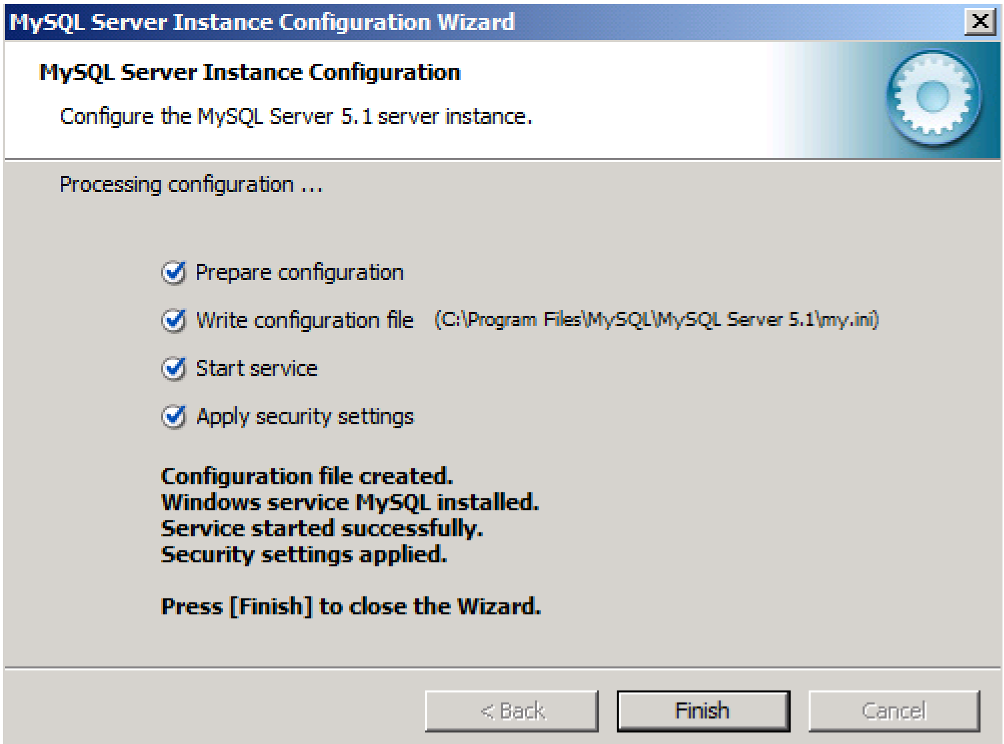


Fig. 1.11 MySQL (11) window

You can now login to MySQL by opening a command prompt window and typing the following command *Fig. 1.12 MySQL (12) window*:

Консоль Копировать

```
mysql -u root -p
Enter password: ******
```

Fig. 1.12 MySQL (12) window

If MySQL is configured correctly, the MySQL prompt will be displayed *Fig. 1.13 " MySQL (1)" window*:

Консоль Копировать

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 3
Server Version 5.1.32-community MySQL Community Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Fig. 1.13 "MySQL (13)" window

Run the installer file “mysql-connector-odbc-5.1.11-win32.msi” on the PC, and then activate “Next” in the window “MySQL Connector/ODBC 5.1 (1)” (Fig. 5.82). 1196Attention: the installer file " mysql - connector - odbc -5.1.11- win 32. msi " should be run for both 32-bit and 64-bit versions of the operating system

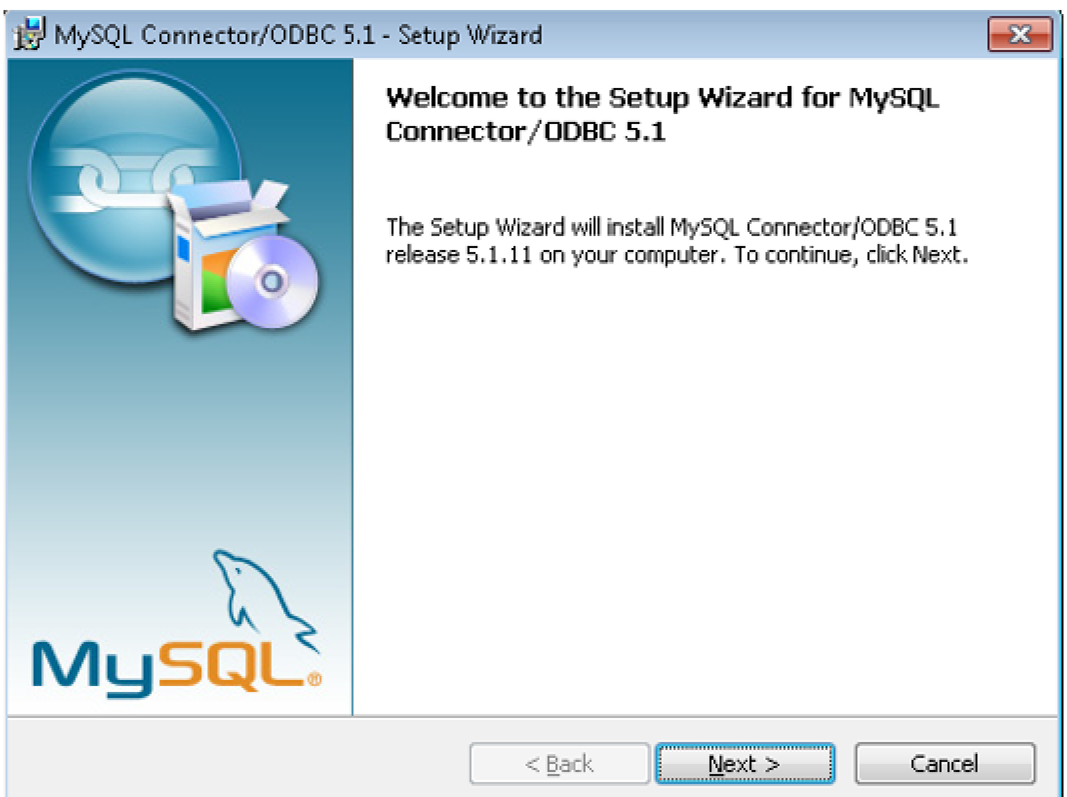


Fig. 1.10 "MySQL Connector/ODBC 5.1 (1)" window

In the "MySQL Connector / ODBC 5.1 (2)" window (Fig. 1.11 1196) select "I accept the terms in the license agreement" and then activate "Next".

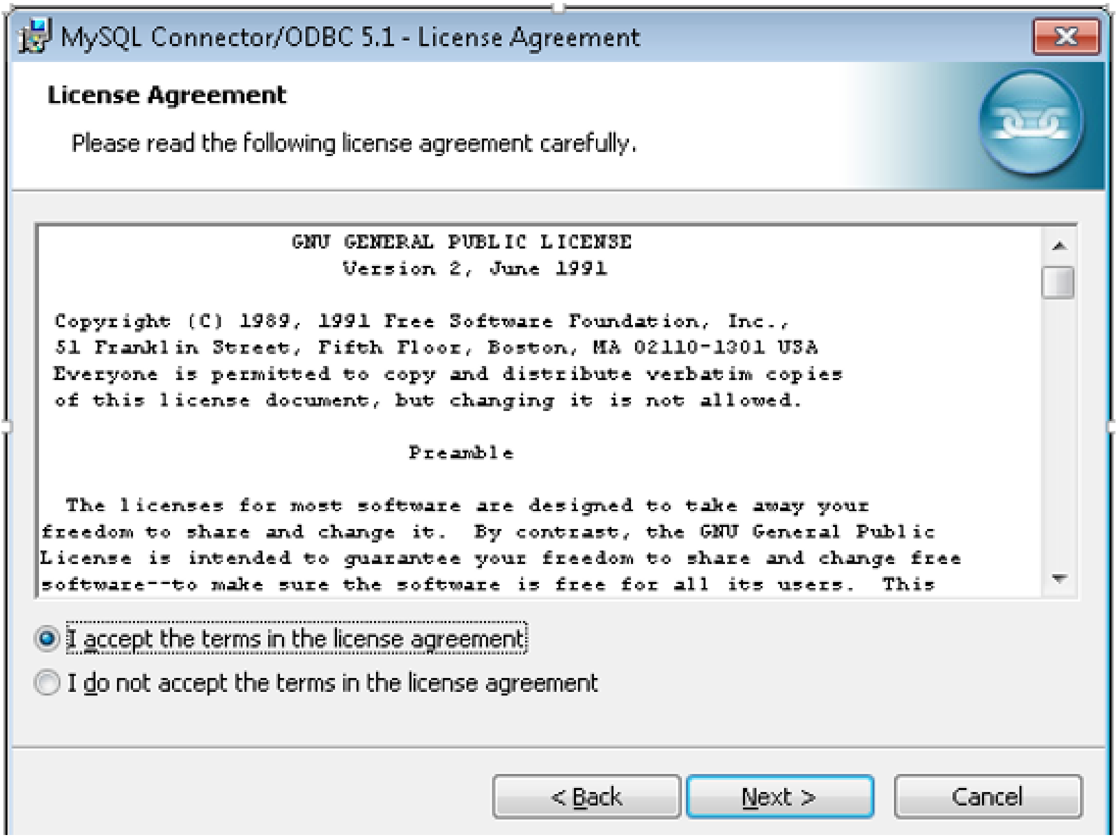


Fig. 1.11 "MySQL Connector/ODBC 5.1 (2)" window

In the "MySQL Connector / ODBC 5.1 (3)" window (Fig. 1.12 1196) select "Typical" and then activate "Next".

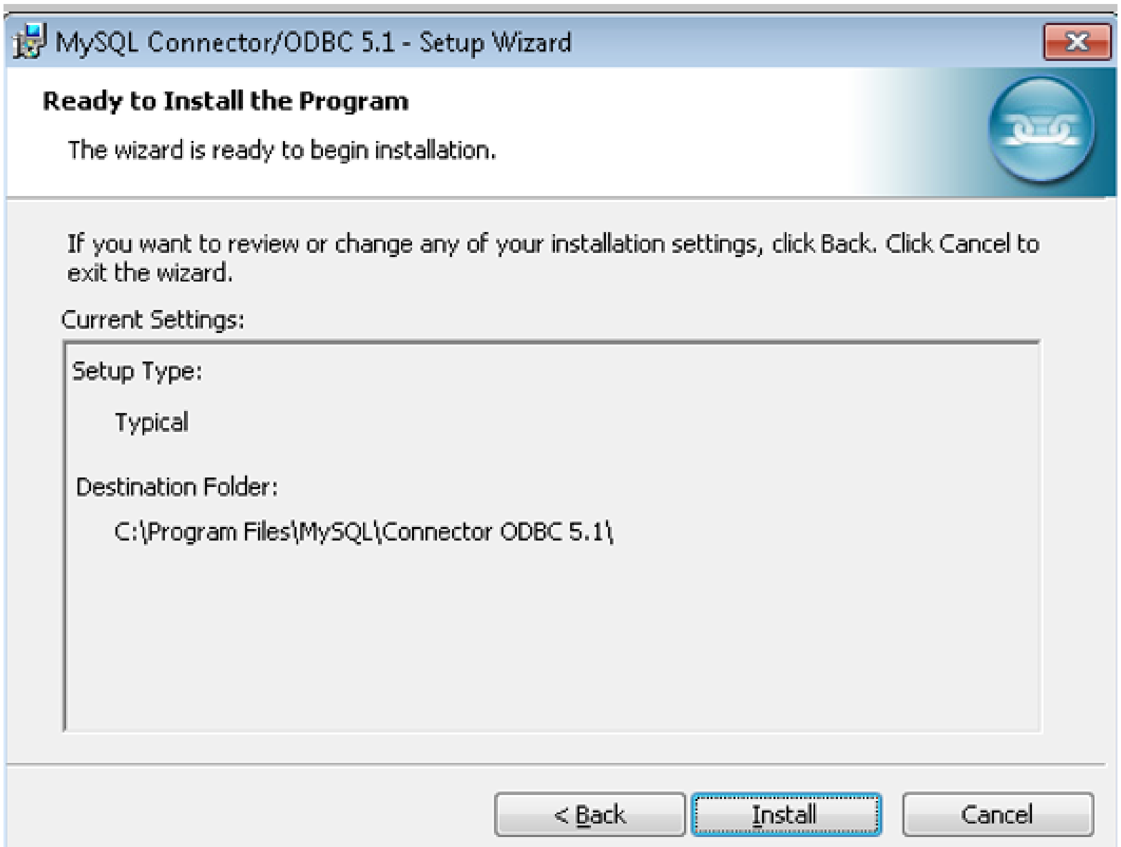


Fig. 1.12 "MySQL Connector/ODBC 5.1 (3)" window

In the "MySQL Connector/ODBC 5.1 (4)" window (Fig. 1.13 1196) activate "Install".

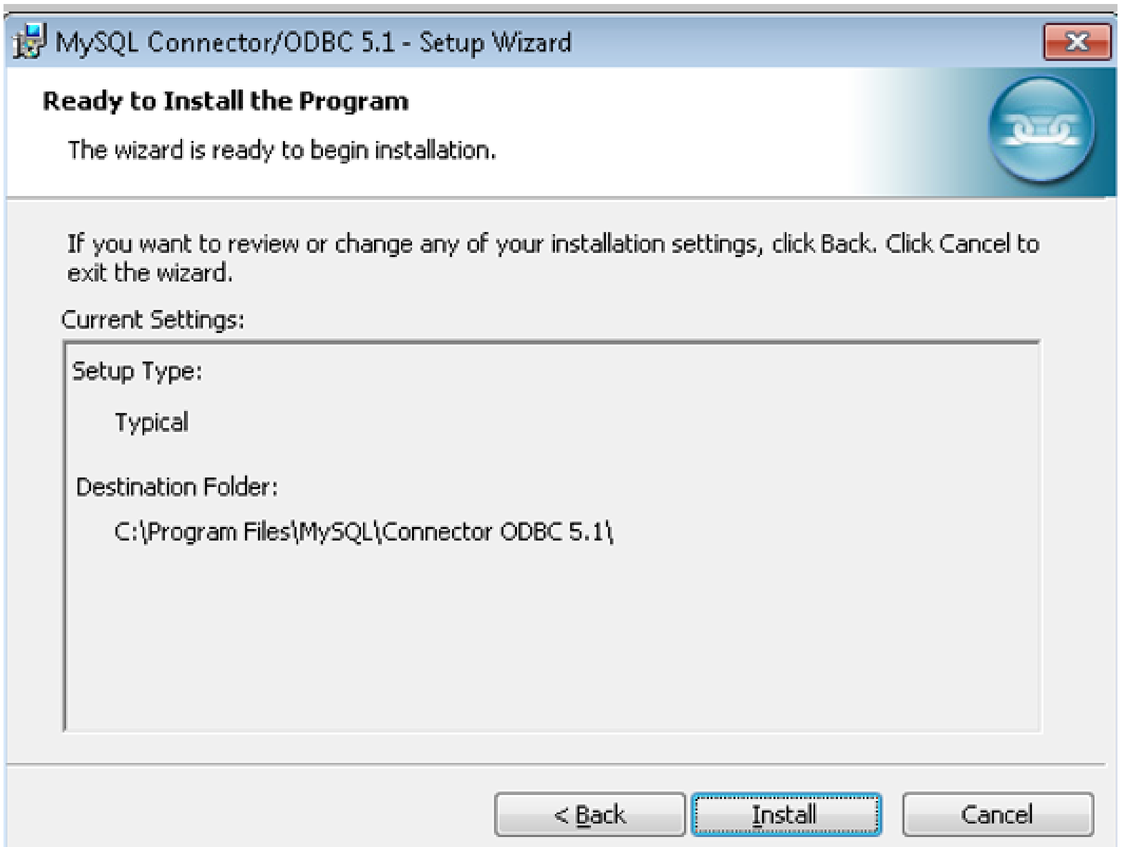


Fig. 1.13 "MySQL Connector/ODBC 5.1 (4)" window

In the "MySQL Connector/ODBC 5.1 (5)" window (Fig. 1.14 1196) activate "Finish".

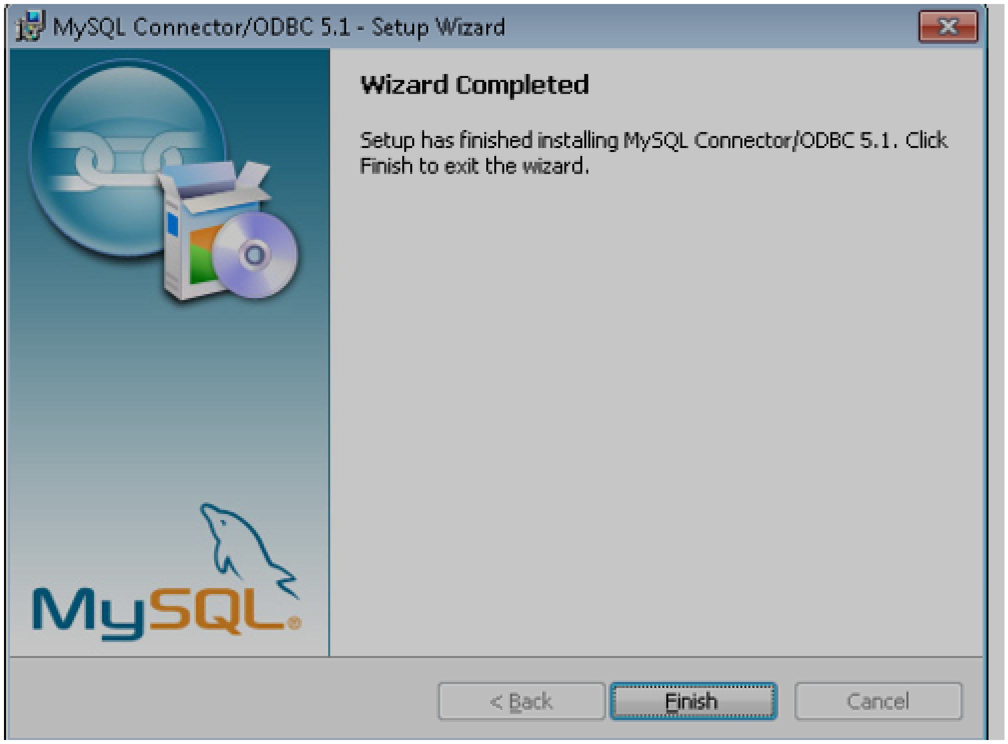


Fig. 1.14. MySQL Connector/ODBC 5.1 (5) window

From the system window "Run" (called by the command " Win + R ") run the program " odbcad 32.exe " (Fig. 1.15 1196), and in case of using a 64-bit version of the operating system, run the program " odbcad 32.exe " from the directory C :\ Windows \ SysWOW64 \.

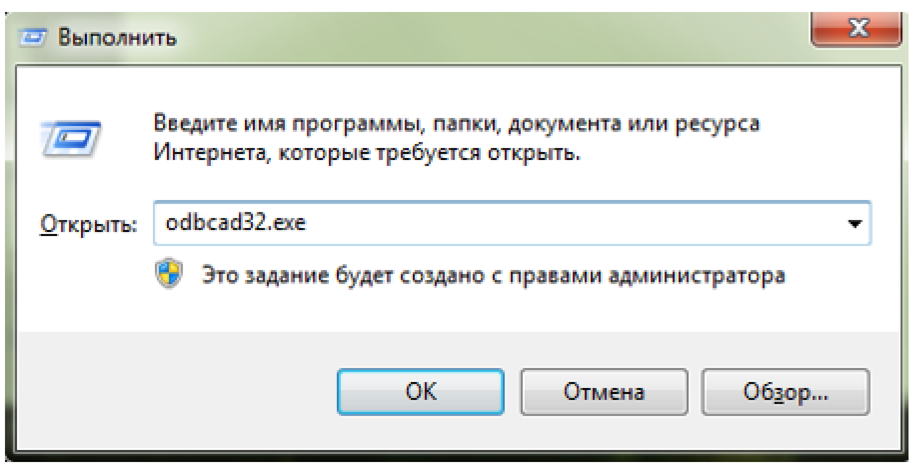


Fig. 1.15 Launching the program " odbcad 32.exe "

ODBC Data Source Administrator " on the tab " System DSN "Make sure that the driver" MySQL ODBC 5.1 Driver ". If this driver is missing, then select the "Add" command (Fig. 1.16 1196).

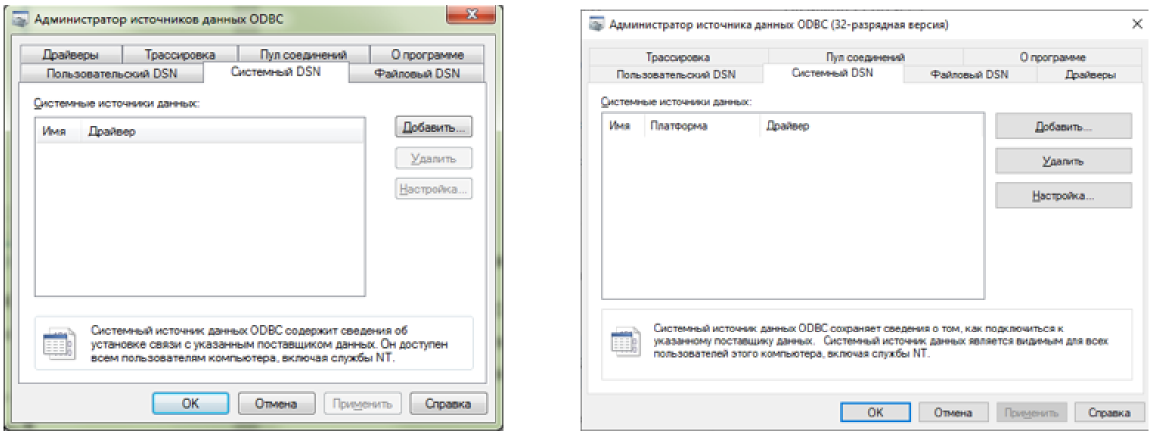


Fig. 1.16 ODBC Data Source Administrator program window

In the "Create a new data source" window, select the " MySQL" driver ODBC 5.1 Driver " and click "Finish" (Fig. 1.17 1196).

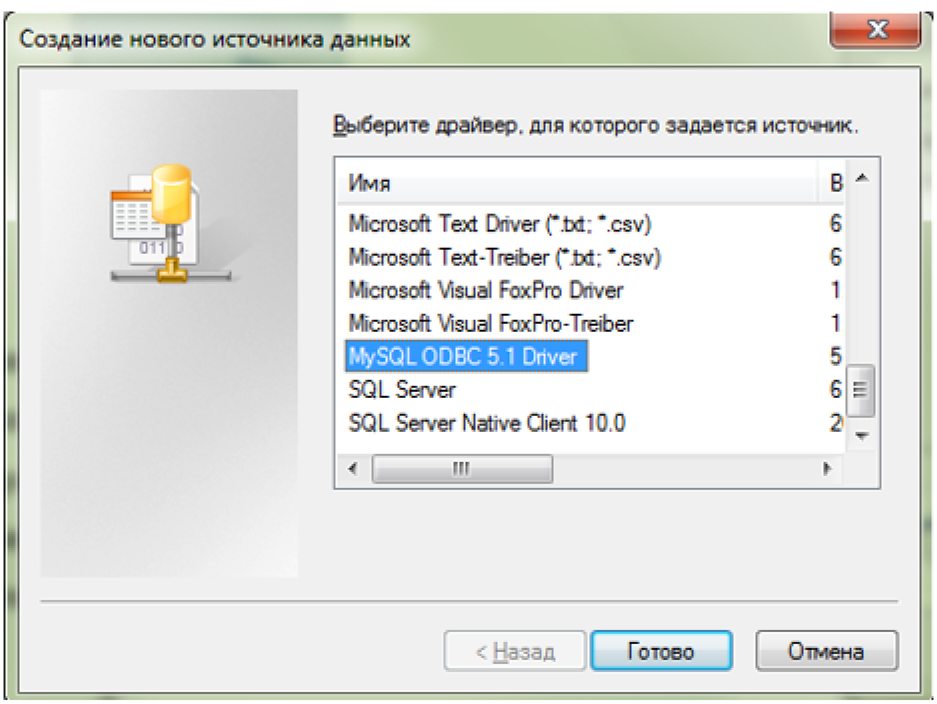


Fig. 1.17 Selecting the MySQL ODBC 5.1 Driver

In the opened MySQL Connector / ODBC Data Source Configuration » to configure the driver in accordance with Fig. 1.18 1196.

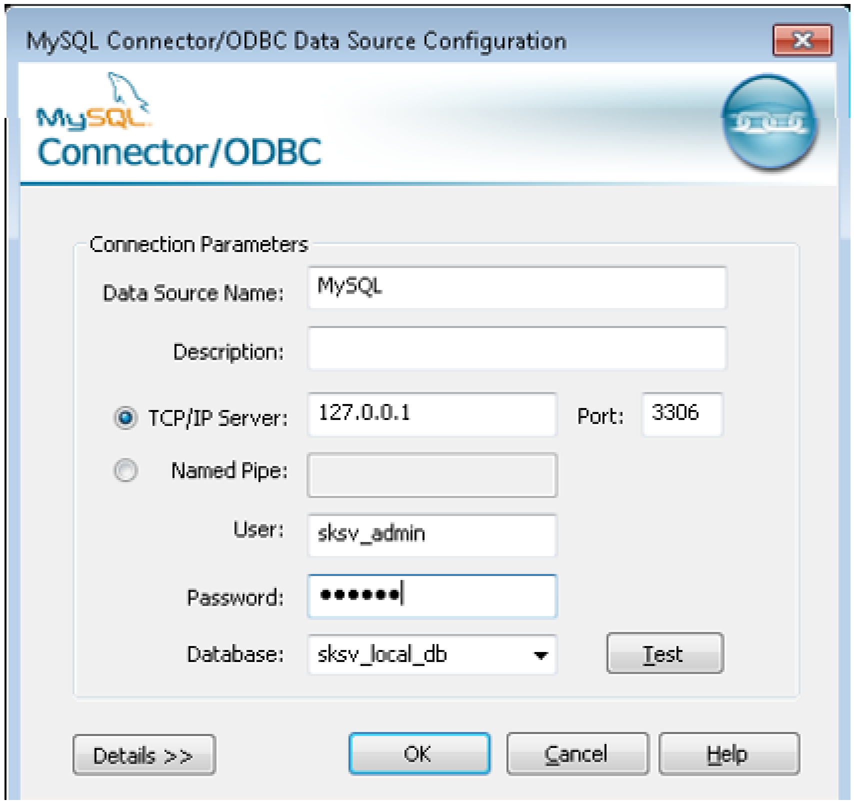


Fig. 1.18 MySQL driver setup ODBC 5.1 Driver »

To check the correctness of the settings made, click the "Test" button. If the test is successful, the program will display a message (Fig. 1.19 1196).

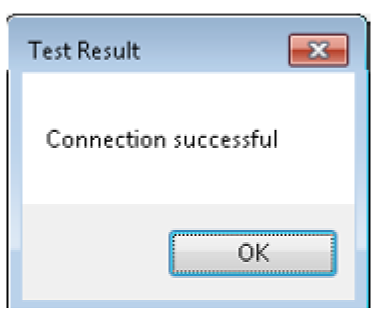


Fig. 1.19 Success message

To save the settings, click the "OK", "Apply" buttons.

When using a 64-bit version of the operating system, the above steps are performed in the same way, only the odbcad32.exe application is launched from the C:\Windows\SysWOW64\ directory.

Creating a database, creating a database table, entering fields

Contents

Launch the MariaDB program by activating (double-clicking) the shortcut on the Windows desktop with the “ HS ” logo (Fig. 1.1 1203).

Fig. 1.1 Label " HeidiSQL "

In the "Session Manager" window that opens (Fig. 1.2 1203), activate "Create" to create a new session and then set a name for the session.

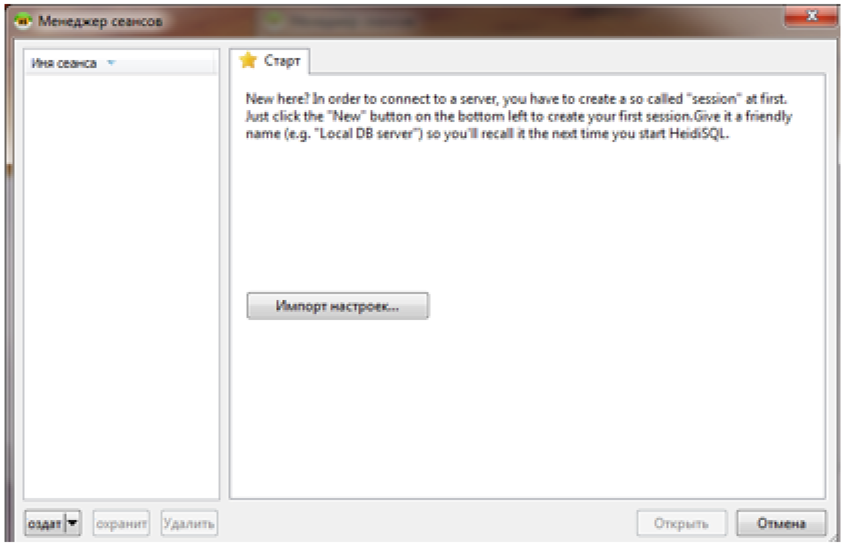


Fig. 1.2 "Session Manager" window

In the created session, enter the password for root and click the "Open" button.

In the "Request" tab (Fig. 1.3 1203), copy and paste the text below:

```
CREATE DATABASE IF NOT EXISTS `local_db` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `local_db`;

CREATE TABLE IF NOT EXISTS `events` (
  `#` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `##` int(11) unsigned NOT NULL,
   `Date and time`  timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
`Event` varchar(128) NOT NULL,
`Source` varchar(128) NOT NULL,
`Operator` varchar(64) NOT NULL,
`Note` varchar(256) NOT NULL,
  PRIMARY KEY (`#`,`##`),
  UNIQUE KEY `#_UNIQUE` (`#`,`##`)
) ENGINE=InnoDB AUTO_INCREMENT=42075 DEFAULT CHARSET=utf8;

GRANT ALL PRIVILEGES ON sksv_local_db.* TO 'admin'@'%' IDENTIFIED BY '111111' WITH GRANT OPTION;

GRANT REPLICATION SLAVE ON sksv_local_db.* TO 'replica'@'%' IDENTIFIED BY '111111' WITH GRANT OPTION;
```

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD("password" );
SET PASSWORD FOR 'root'@'NPS-XXX' = PASSWORD("password" );
SET PASSWORD FOR 'root'@'164.0.0.1' = PASSWORD("password" );
SET PASSWORD FOR 'root'@':1' = PASSWORD("password" );
SET PASSWORD FOR 'admin'@'%' = PASSWORD("password" );
FLUSH PRIVILEGES;
```

Note: when configuring, you must replace the text "password" and " XX - XXX " with the actual values for this computer.

Then activate the "Run" symbol in the program window, or " F 9" on the keyboard.

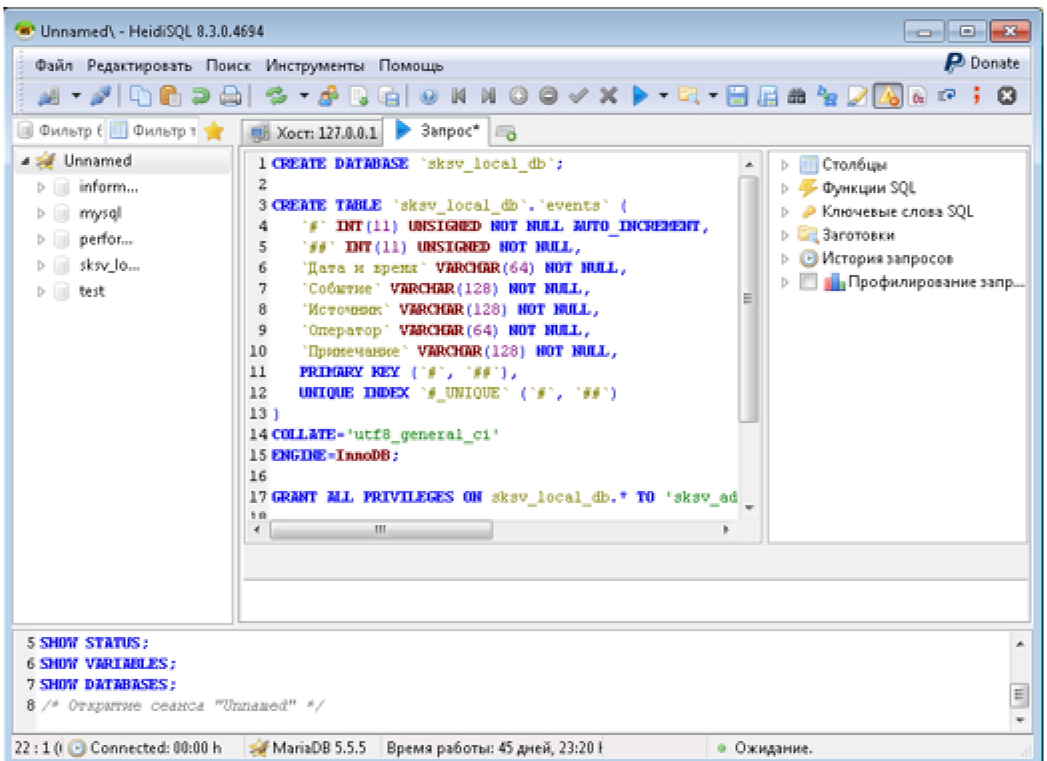


Fig. 1.3 Request tab

Close the MariaDB program, when a message appears asking you to save the file, select the "No" option.

Open the configuration file "my . ini " located in the directory:

For 32-bit operating system - C:\ Program Files \ MariaDB 10.0\ data

For 64-bit operating system - C:\ Program Files (x 86)\ MariaDB 10.0\ data

In the configuration file "my . ini " copy and paste the text below.

```
[mysqld]
datadir=C:/Program Files/MariaDB 10.0/data
port=3306
sql_mode="STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION"
default_storage_engine=innodb
innodb_buffer_pool_size=254M
innodb_log_file_size=50M
character-set-server=utf8
server-id=XXX
log-bin = C:/Program Files/MariaDB 10.0/data/binlog
replicate-do-db = sksv_local_db
[client]
port=3306
```

Note:

In the config file " my . ini " in the [mysqld] section , you must specify a unique identifier for this seismic station (server - id =XXX), where XXX is an identifier corresponding to the sequence number of the seismic station.

In the config file " my . ini " in the [mysqld] section, you must specify a name for the log:

For 32-bit OS: log-bin = C:/Program Files/MariaDB 10.0/data/binlog

For 64-bit OS: log-bin = C:/Program Files (x86)/MariaDB 10.0/data/binlog

In the config file " my . ini " in the [mysqld] section, you must specify the database to synchronize (do - db = local _ db).

Attention! If the slashes in the file " my . ini " will be in the opposite direction, then the MySQL service will not start .

As a result of the changes, the file my . ini will have an approximate form, as shown in the example (Fig. 1.4 1206).

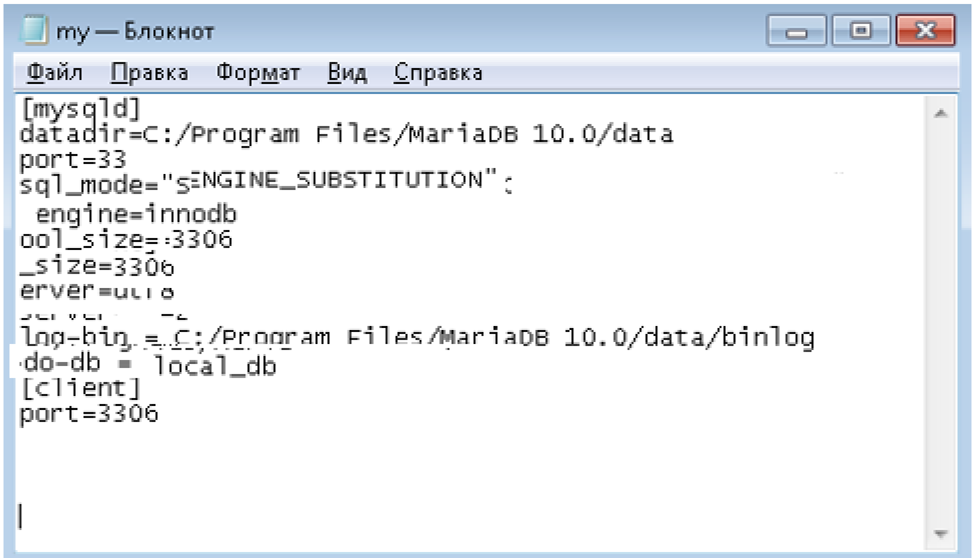


Fig. 1.4 "Configuration file" xx . ini "

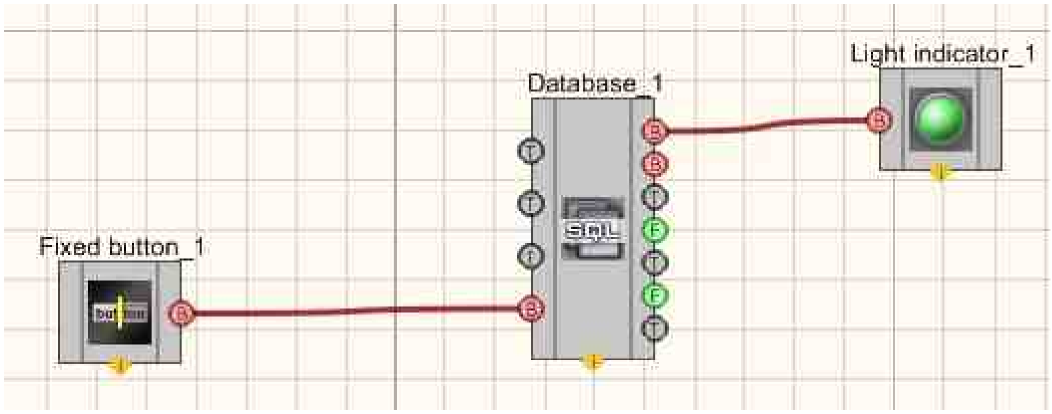
Make sure you have enough disk space for binary logs.

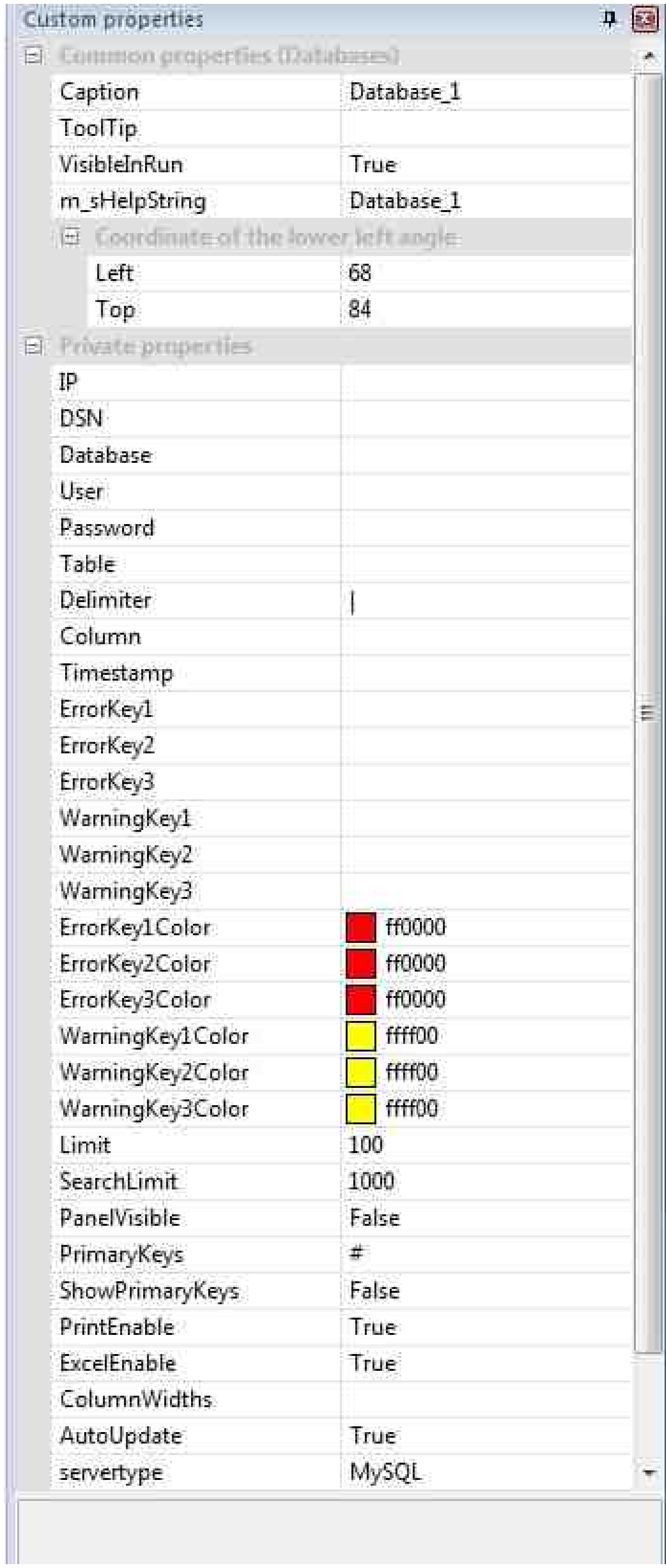
Add the replica user under whose rights replication will be performed. The " replication" privilege will suffice. slave ". To do this, you need to execute the request " SLAVE ON * TO ' rep '@'%' IDE BY '1'".

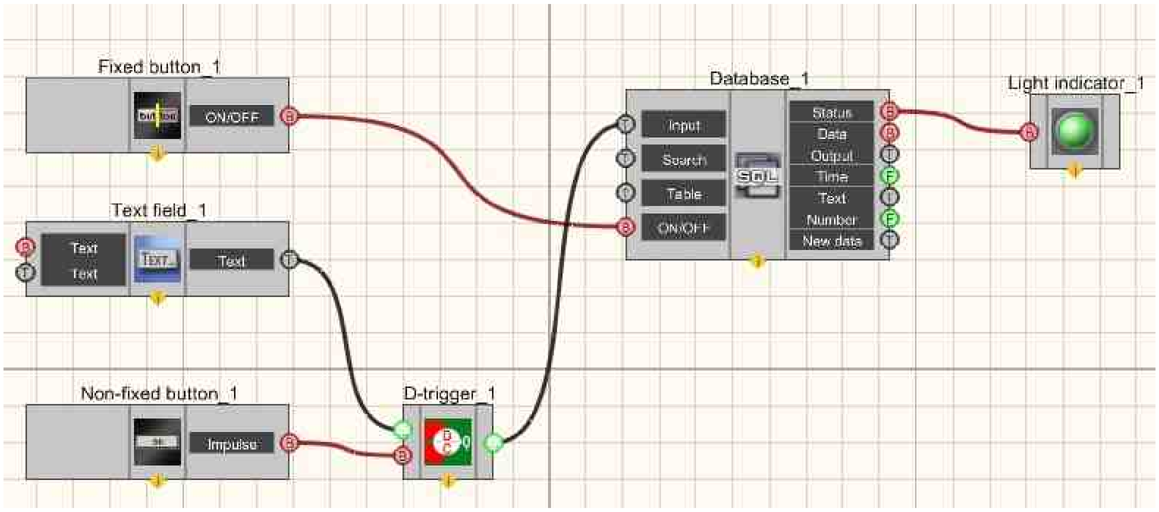
Reload the MySQL service ;

Execute the query " SHOW MASTER STATUS ". If the synchronization setup was successful, then the request returns a message containing an approximate view:

- File: mysql-bin.021451;
- Position: 104;
- Binlog_Do_DB;
- Binlog_Ignore_DB.


Light indicator_1
Database_1
Fixed button_1
SQL


Custom properties
Common properties (Databases)
Caption Database_1
ToolTip
VisibleInRun True
m_sHelpString Database_1
Coordinate of the lower left angle
Left 68
Top 84
Private properties
IP
DSN
Database
User
Password
Table
Delimiter |
Column
Timestamp
ErrorKey1
ErrorKey2
ErrorKey3
WarningKey1
WarningKey2
WarningKey3
ErrorKey1Color ff0000
ErrorKey2Color ff0000
ErrorKey3Color ff0000
WarningKey1Color ffff00
WarningKey2Color ffff00
WarningKey3Color ffff00
Limit 100
SearchLimit 1000
PanelVisible False
PrimaryKeys #
ShowPrimaryKeys False
PrintEnable True
ExcelEnable True
ColumnWidths
AutoUpdate True
servertype MySQL



Adding a new entry to the "Database" from ZETView

To add a new record to the Database with which the connection is established, it is necessary to submit a text string to the input of the "Database" component in which the Delimeter (by default "|") will divide the data by table columns (by fields).

A simple example is shown in Fig. 1.7.1 and 1.7.2.

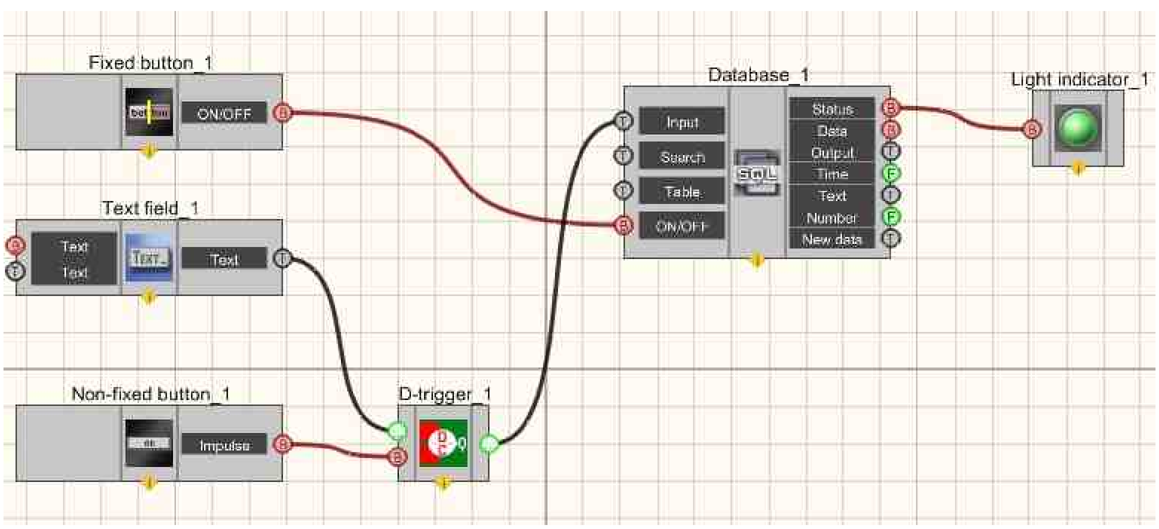


Fig. 1.7.1 "Adding a new entry (line) Type of designer.

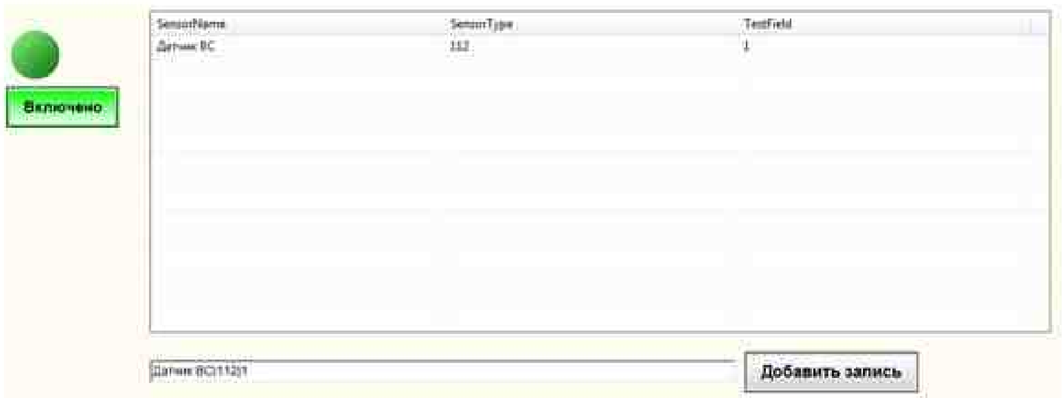


Fig. 1.7.2 "Adding a new entry (row) Type of operator.

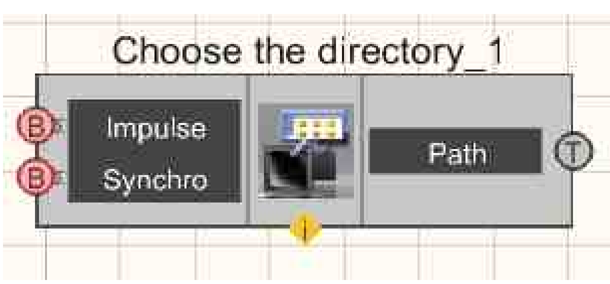
This project has a "Text field" in which we find a line with separators. This line gets to the input of the "D-trigger" that saves the entered line and the "Non-commit button" by clicking on which, the line saved in the "D-trigger" gets to the input of the "Database" component, after which it is entered as a new record in the database to which the connection is made.

The example shows that an entry about the BC type 112 sensor marked 1 got into the database.

31.2.Choose the directory

The component "**Choose the directory**" is used for allows the operator to select a folder and set corresponding directory for the components. Thus, the operator may assign a folder for automated saving of the reports.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - when a control pulse arrives, it opens the specified directory and sends data to the output.
- Synchro - send directory name.

➢ Output:

- Path - getting the directory name.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

FolderName (C :\) - Set the name of the directory.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

- Are available by the link below. 154

➢**Adjustable custom properties:**

- BSTR FolderName - Set the name of the directory (string).

Example

Project in SCADA ZETView

When the button is pressed without fixing, the Choose the directory 1211 component receives a control pulse as an input and prompts you to select a directory, then passes the received value from the "Path" output, in this example the result is simply displayed in a text label. Usually this component is used in conjunction with the "Report" component to select a directory for saving reports.

Project operation result

View the example in ZetView

31.3.Select the file

The component "**Select the file**" is used for selecting the file name, shows the file directory.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - opens File Explorer to select a file.
- Synchro - passes the path to the file to the output of the component.

➢ Output:

- File - the path to the selected file is received.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FileName - Set the file name.
- Type - set the type: Open; Save.
- Filter - set the filter.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:
- Are available by the link below. 154

➢**Adjustable custom properties:**

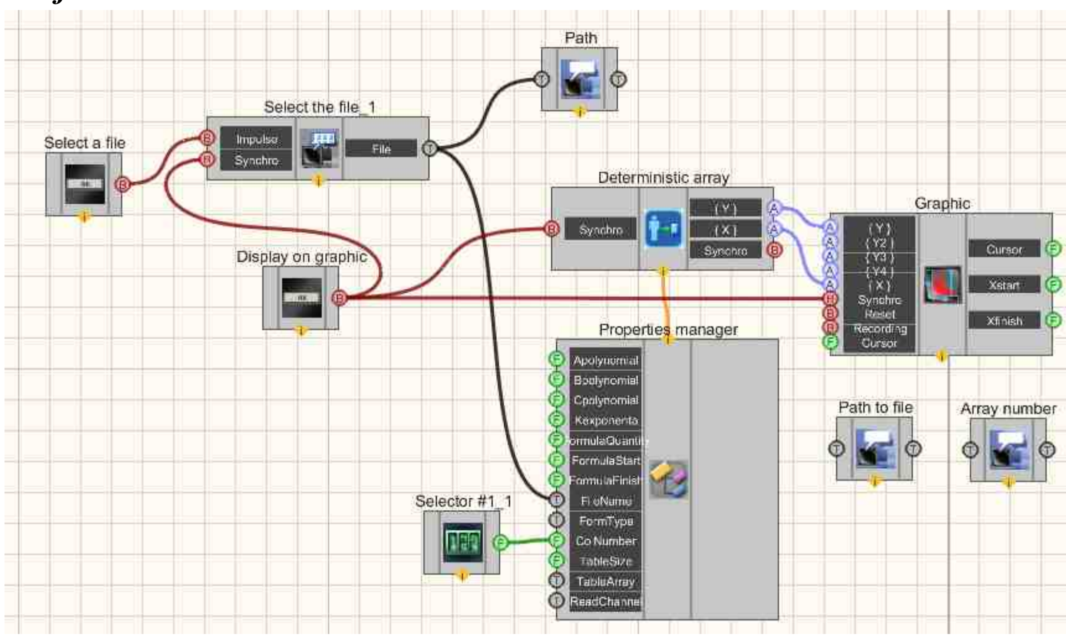
- BSTR FileName - Set the file name (string).
- BSTR Type - set the type: Open; Save (string).
- BSTR Filter - set the filter (string).

Note:
Added **Custom private properties** Filter. You can add file permissions to it with a space.
For example for the string: dtu dtx txt
In the file selection dialog box, there will be a selection by the following extensions . dtu, . dtx *. txt.
If the string is empty, the filter will be disabled (File type: All files(*.*))
If you need to add "All files(*.*)" in addition to the specified ones, you should record in a line, for example: dtu .

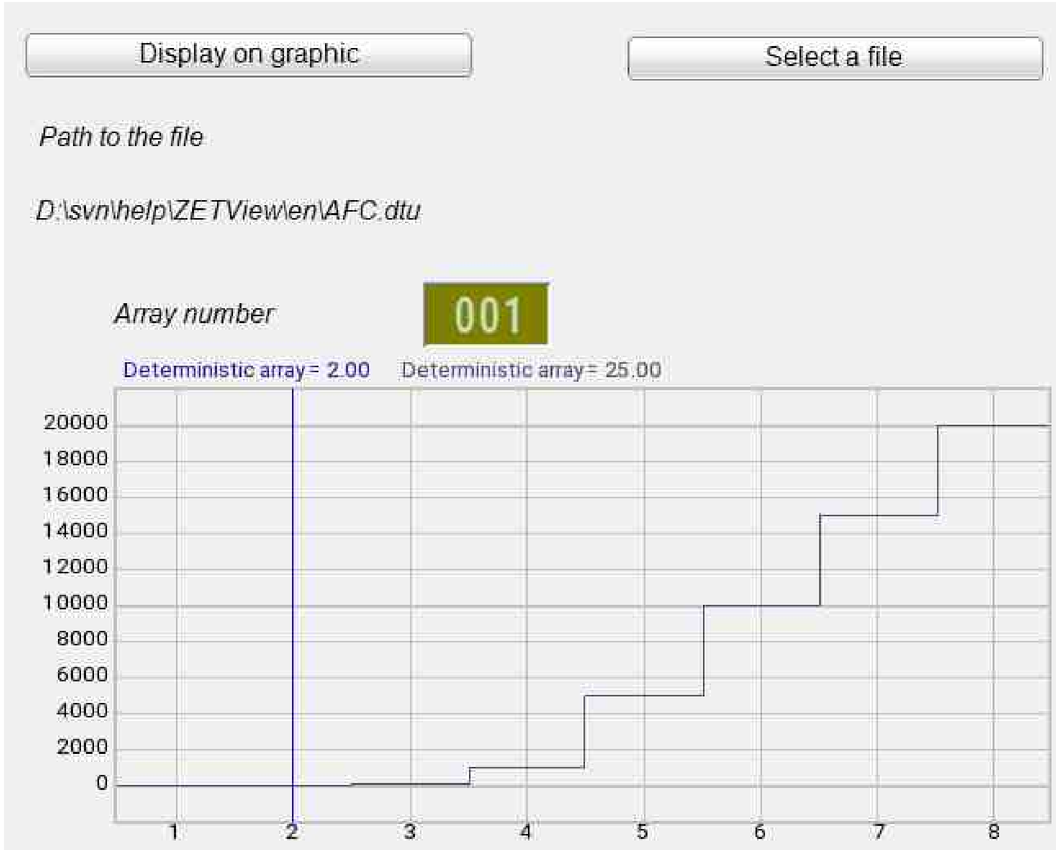
Custom private properties

Example No. 1

Project in SCADA ZETView



Project operation result

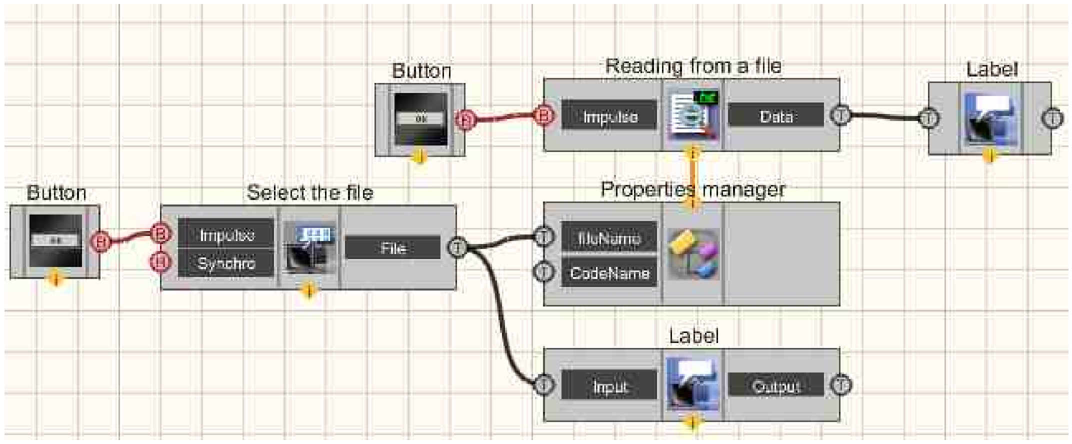


View the example in ZetView

Example No. 2

Project in SCADA ZETView

In this example, an arbitrary txt file is used, which is not included in the example. When running the example, to see the result of its work, you must select any txt file on your computer using the " Open " button, the file path will be displayed below. And then click the "Read File" button. The result may differ from this example, depending on the contents of the selected .txt file.

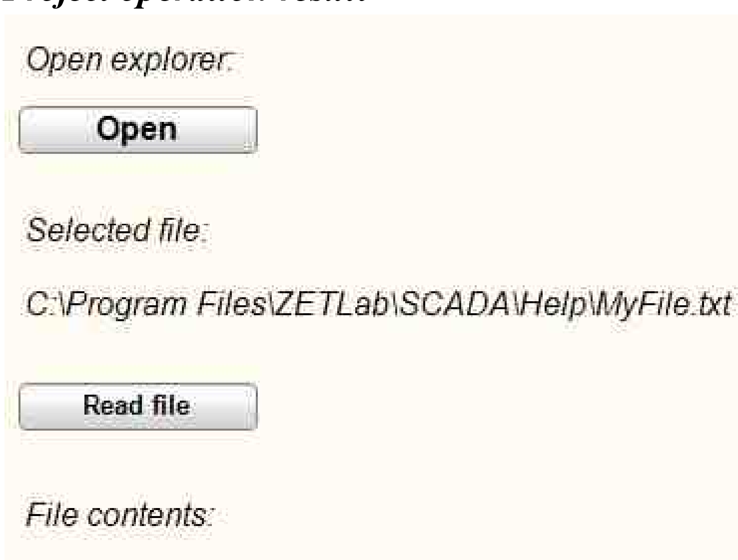


This diagram shows the operation of the Select the file 1213.

When the Non-fixed button 694 is pressed, a impulse is sent to the input of the Select the file 1213, as a result of which the explorer opens to select a file. The path to the Select the file 1213 is displayed with the Label 1087. Also, a string containing the path to the file is sent to the Property Manager 999 connected to the Read from file 1213 component. When receiving a control pulse from a Non-fixed button 694, this component reads the contents of the selected file and outputs the result using the Labels 1087.

For this example, you can use any txt file.

Project operation result

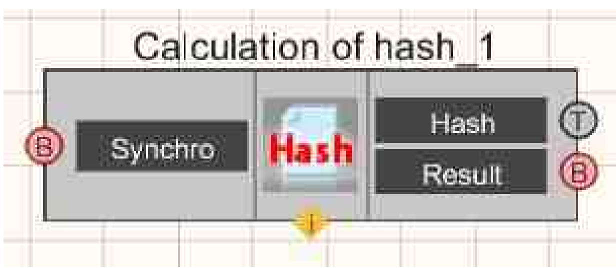


View the example in ZetView

31.4.Calculation of hash

The component "**Calculation of hash**" is used for to verify the integrity and authenticity of files by calculating the checksum.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Synchro - sends a signal to calculate the hash function.

➢ Output:
- Hash - a string with a hash sum.
- Result - the result of the check.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FileName - Set the filename to read the hash.
- HashType - Set the type of hash function to be calculated: MD 5.
- UseSalt - Set salt for MD 5.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

- Are available by the link below. 154

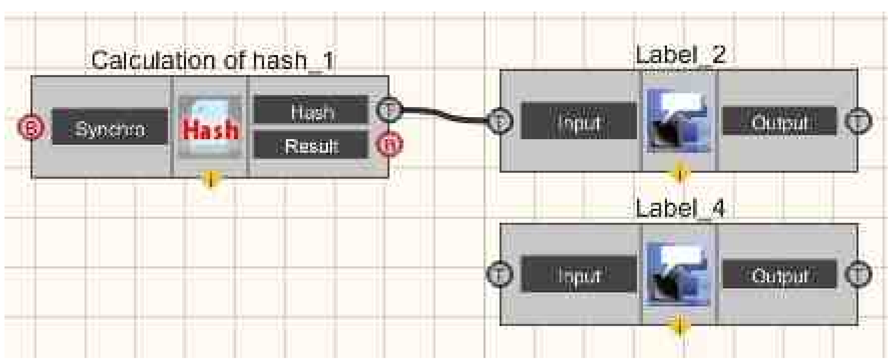
➢**Adjustable custom properties:**

- BSTR FileName - Set the file name for the hash calculation (string). .
- BSTR HashType - Set the type of hash function to be calculated: MD 5 (string). .
- BSTR UseSalt - Set salt for MD 5 (string). .

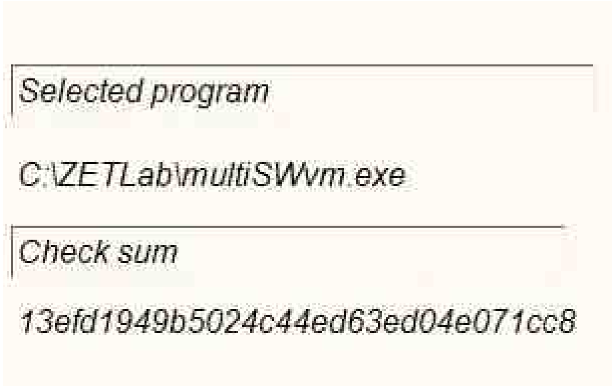
Example

Project in SCADA ZETView

In this example, an arbitrary program from the ZETLab composition is used . When you run the example, to see the result of its work.



Project operation result

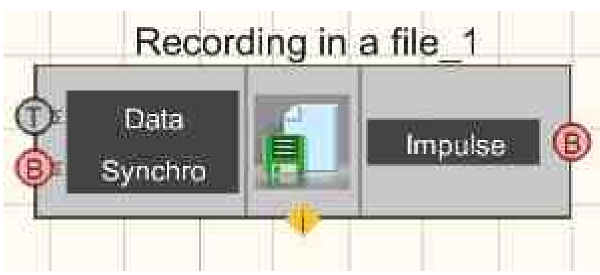


View the example in ZetView

31.5.Recording in a file

The component **"Recording in a file"** allows to record data to a file in the course of project operation. Thus, it is possible to use dynamic data, that do not relate to project operation.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Data - data to be written to the file.
- Synchro - a command to record to a file.

➢ Output:
- Impulse - informs about the end of writing to the file.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- fileName - Set the file name to record to the file.
- typeMode - Set the operating mode: By change; By impulse.
- writeMode - Set the record mode: Overwrite; Add to end.
- Encoding - Set file encode: UNICODE ; ANSI ; UTF -8; UTF -16

➢**Methods:**

- void writeFile() - record to a file.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:

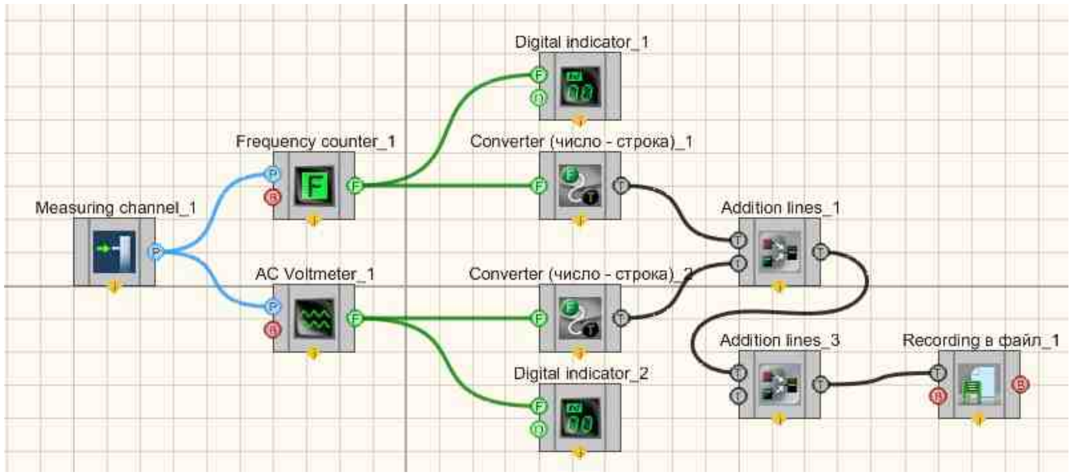
- Are available by the link below. 154

➢**Adjustable custom properties:**

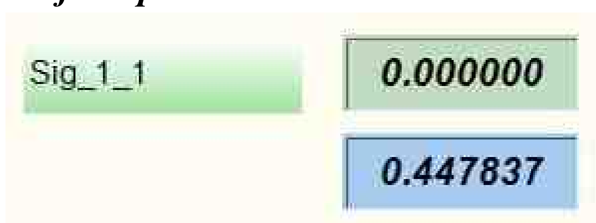
- BSTR fileName - Set the filename to be written to the file (string).
- BSTR typeMode - Set the operating mode: By change; By impulse (string). .
- BSTR writeMode - Set the record operation mode: Overwrite; Add to end; (string). .
- BSTR Encoding - Set file encode: UNICODE ; ANSI ; UTF -8; UTF -16; (string).

Example

Project in SCADA ZETView



Project operation result



The data received in the file:

Note: If you apply a Signal to the component, then in the text document the frequency will be indicated in the first column, and its amplitude in the second.

View the example in ZetView

31.6.Run program by name

The component **"Run program by name"** is used for allows to run programs (or files) from the project. .

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - runs the specified file.
- Output:
- Impulse - the program informs about the completion of work.

If "1" comes to the "Impulse" input, then the file is launched in a new process, if "0" comes to the input, then the previously launched files with the specified name are closed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- fileName - the name of the file to run.
- fileParametr - the parameter for the file to be started.

➢Methods:

- StarProcessAndWait() - Start the program and wait for it to finish.

This component is designed to run programs. To do this, in the component's " fileName " property, you must specify the full name of the program, i.e. path and name with extension " exe ", " bat " or " cmd ". The " fileParameter " property specifies the command line of the process to be launched. For ZetLab software programs, it is not necessary to set the path. Upon completion of the ZetView project with this component, the running program will be closed. If the program is completed before the completion of the project, then a logic one pulse will be issued to the output pin of the component. The accuracy of the time of issuing a pulse to the output contact is ± 1 s.

In addition, the component allows you to run files with registered extensions for processing. If the path to this file is not specified, the component will assume that the specified file is located in the current directory (folder). So, for example: when setting the " fileName " property to the text " test .

txt ", the Notepad program (or another program specified for this user as processing files with the extension " txt ") will be launched, into which the file will be loaded from the current folder " test .txt " . At the same time, upon completion of the project, the program that processes this file will not be closed. Also, no signal will be given to the output pin of the component if the processing program is closed before the completion of the ZetView project.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

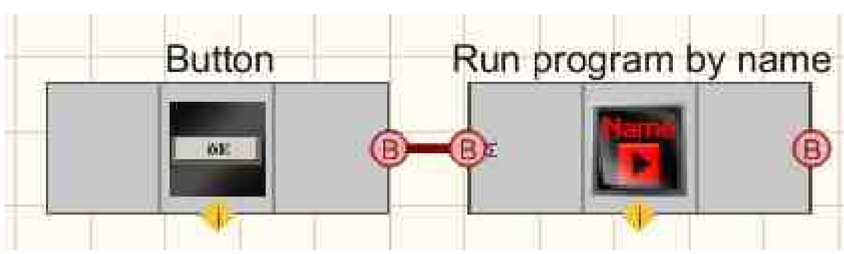
- BSTR fileName - Set the name of the file to run (string).
- BSTR fileParametr - Set the parameter to the file to be launched (string).

➢**Methods:**

- StarProcessAndWait() - Start the program and wait for it to finish.

Example No. 1

Project in SCADA ZETView



Project operation result

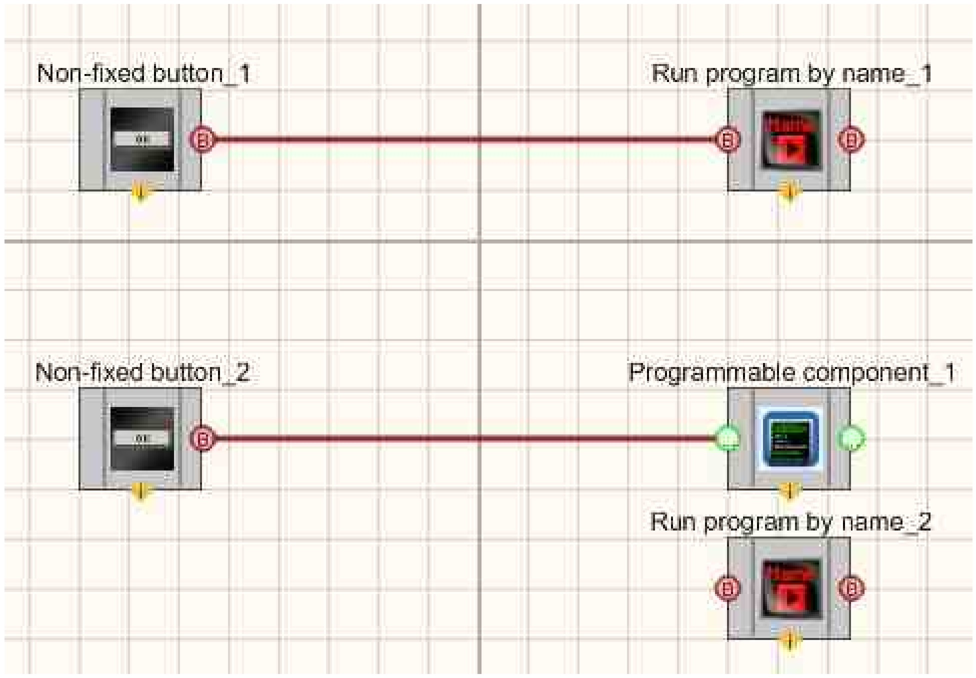
In the Run program by name 1222 component, you must correctly specify the directory and path to the executable program.

When the Non-fixed button 694 is pressed, a pulse is sent to the input of the Run program by name 1222 component, as a result of which the program will be launched by name.

The path to the executable program may be different when running by default, which may cause the example to work incorrectly.

Example No. 2

Project in SCADA ZETView



Project operation result

In the Run program by name 1222 component, you must correctly specify the directory and path to the executable program.

When the Non-fixed button 694 is pressed, a pulse is sent to the input of the Run program by name 1222 component, as a result of which the program will be launched by name.

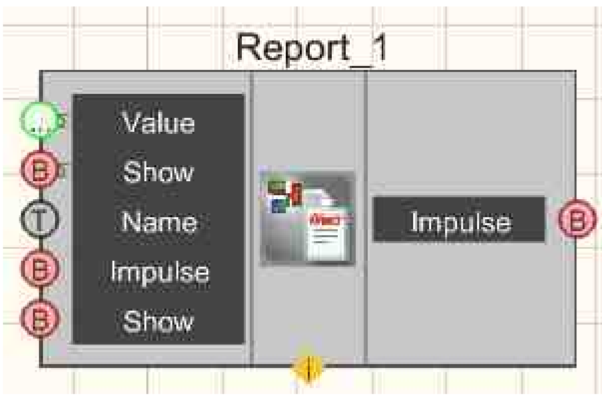
The path to the executable program may be different when running by default, which may cause the example to work incorrectly.

View the example in ZetView

31.7.Report

The component **"Report"** is used for recording various values to a file in accordance with user template.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Value - record the value to the report. Components are connected to this contact, the values of which are required to be included in the report. Values are accumulated until the report is saved, so the last values from each connected component are written to the file. A contact can form multiple connections. Contact interface "Any connection type"
- Show - When the value "1" is received by this contact, the current values are displayed.
- Name - When the "Autoname" flag is set in the report settings window, the created file, when saved, is assigned the value that came to this input as a name.
- Impulse - When the value "1" is received by this contact, the accumulated values are saved to a file.
- Show - When a value of "1" is received by this contact, the file of the last saved report is opened.

➢ Output:

- Impulse - this contact sends the value "1" to the connected components at the end of the report generation. The report generation time depends on the amount of data that needs to be written to the file. In addition, most projects generate reports automatically. Thus, to increase the information content, a report creation indicator can be added to the project.

Custom common properties:

➤Total properties (environment):

- [Are available by the link below.](154)

➤**Custom private properties (the default value is shown in parentheses):**

- FileTemplateName - the name of the template file for the report. The template file extension must match the extension of the future report. Those. if you set the reporttype property : " Microsoft Word (. doc)", then the report template file must have the extension *. doc . Adding the ability to select a template with the . xlsm extension
- Show (false) - show the report after it has been generated.
- replace unused identifiers in the report. Since the templates use the data entry form of the &1_ Massiv 1/& type (then such inscriptions are replaced by the values recorded in the report), then if, for example, the template file is designed for five values, but gets four, then the remaining blank value will be replaced by a dash. If the Replace property is set to " False ", then the extra lines will not be replaced. See fig.

This is how the report template file looks like, in *.doc format

1	&1_massiv1/&
2	&2_massiv1/&
3	&3_massiv1/&
4	&4_massiv1/&
5	&5_massiv1/&

This is how the report file looks like in *.doc format if the Replace property is set to True

1	23.86
2	13.55
3	14.18
4	27.11
5	-

This is how the report file looks like, in *. doc if the Replace property is set to False

1	23.86
2	13.55
3	14.18
4	27.11
5	&5_massiv1/&

- Reset (false) - zeroing the base of fixed values. This property is used when the report file needs to be overwritten. For example, during the first recording, 40 values were written to the report, after that the report was regenerated into the same file, writing 20 values to it. If the Reset value was set to " True " when generating the report , ZETView will delete all previous values and record new ones in their place. If the value of the property is set to " False " then the data will be simply replaced, the results of the value of which did not change will remain untouched.
- Precision (0) - number of decimal places to report floating point values.
- AutoName (false) - automatic assignment of a report name.
- FolderName - the name of the directory where automatically generated reports will be placed.
- reporttype (Notepad (. txt)) - report type:
- 1. Notepad (.txt).
- 2. Microsoft Word (.doc).
- 3. Microsoft Word (.docx).
- 4. Microsoft Excel (.xls/.xlsm).
- 5. Microsoft Excel (.xlsx).
- 6. Results file (.dtu).
- 7. HTML file (.html).

- BufferVar - label for inserting information from the clipboard into the report when generating the report.
- delimiter (Comma) - delimiter of integer and fractional parts.
- 1. Comma.
- 2. Point.
- ImageFile - The name of the image file to replace the image in the report. C property that can be set to a string with the name of the image file (or a list of image files separated by ";"). Reports in EXCEL - X and WORD - X formats should, in order, replace their images with the ones specified by the user.
- ReplaceZero - Replace null values with a special character.

Note:

Report types "2. Microsoft Word (.doc) " and "4. Microsoft excel (. xls /. xlsm) " can also work with templates in *. docx and *. xlsx formats , but they require installed Microsoft programs office Word and Excel .
Report types "3. Microsoft Word (.docx) " and "5. Microsoft Excel (. xlsx)" work only with templates in *. docx and *. xlsx formats , but they do not require the mandatory presence of Microsoft programs office .

Working with the report:
The template file must contain strings like "& helpstring /&", which, after saving the report, will be replaced with values from the components with the corresponding " m _ sHelpString " field. The "&" and "/&" characters are required to highlight replacement strings.
To save the text to the report, in the " m _ sHelpString " field of the component that transmits the string, specify the "&" symbol before the keyword.
To save an array of numbers or strings to a report in a template, you must specify a set of strings of the form "& i _ helpstring /&", where i is the number of the element in the stored array, starting from the 1st.
To save numbers to a report, you just need to specify the corresponding keyword in the component.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Value - record a value to the report. Components are connected to this contact, the values of which are required to be included in the report. Values are accumulated until the report is saved, so the last values from each connected component are written to the file. A contact can form multiple connections (any type).
- Name - When the "Autoname" flag is set in the report settings window, the created file, when saved, is assigned the value that came to this input (text) as a name.

➢Custom common properties:

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

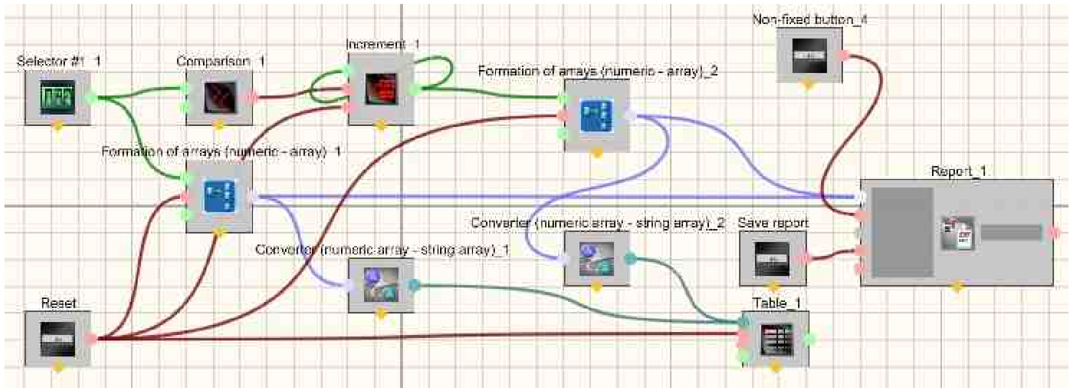
- BSTR FileTemplateName - Set the name of the template file for the report (string). The extension of the template file must match the extension of the future report. Those. if you set the reporttype property : " Microsoft Word (. doc).", then the report template file must have the extension *. doc .
- VARIANT_BOOL Show - Set the report at the end of its generation:
- true - the report is enabled at the end of its generation;
- false - the report is disabled when it is generated.
- VARIANT_BOOL Replace - Set and replace unused identifiers in the report:
- true - enabled the replacement of unused identifiers in the report;
- false - disabled the replacement of unused identifiers in the report.
- VARIANT_BOOL Reset - Set and Reset the base of fixed values:
- true - zeroing of fixed values base is enabled;
- false - zeroing of fixed values base is disabled.
- LONG Precision - Set the number of decimal places to record floating point values to the report (from 0 to 6).
- VARIANT_BOOL AutoName - Set and read automatic report naming:
- true - automatic assignment of the report name is enabled;
- false - automatic naming of the report is disabled.
- BSTR FolderName - Set the name of the directory where automatically generated reports will be placed (string).
- BSTR ReportType - Set report type: Notepad (.txt) ; Microsoft Word (.doc) ; Microsoft Word (.docx) ; Microsoft Excel (.xls) ; Microsoft Excel (.xlsx) ; Results file (.dtu) ; HTML file (.html) (string).
- BSTR BufferVar - Set the label for pasting information from the clipboard into the report when generating the report (string).
- BSTR ReplaceZero - Set to replace zero values with a special character (string).

➢Methods:

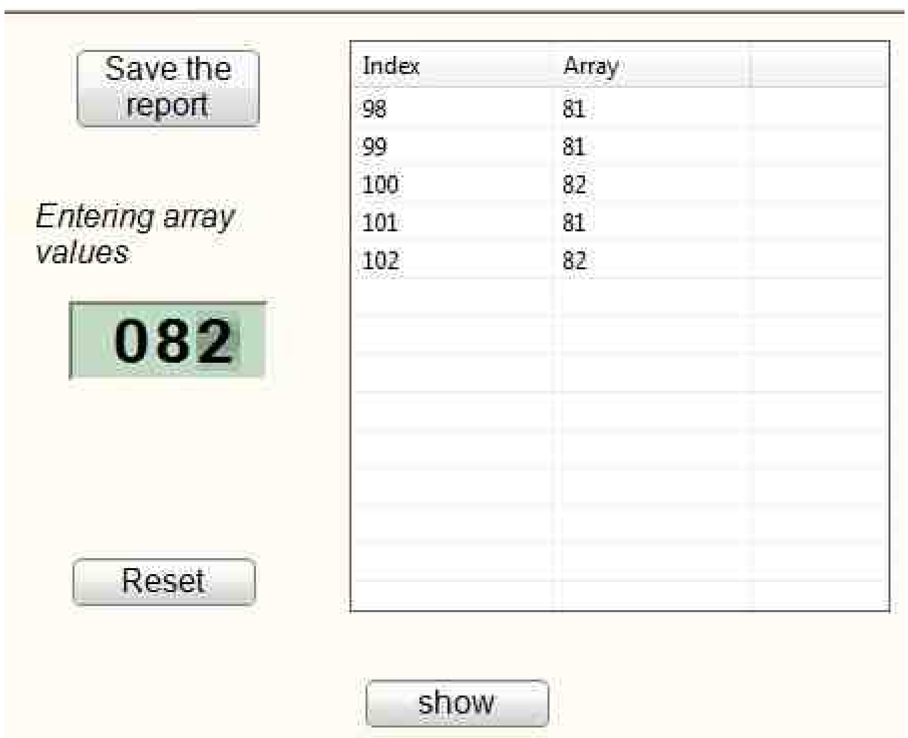
- void PasteFromBuffer(LPCTSTR name) - Add information from the clipboard to the report.
- void ShowReport(void) - Show the report.
- void ClearReport(void) - Clear the database.

Example

Project in SCADA ZETView



Project operation result

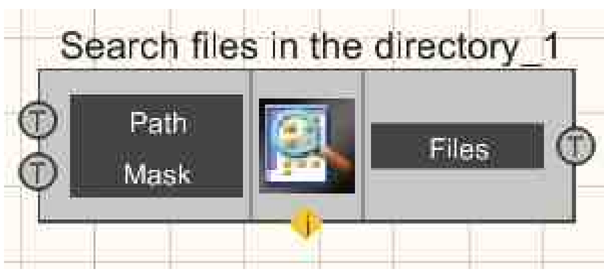


View the example in ZetView

31.8.Search files in the directory

The component **"Search files in the directory"** allows to find the files in particular category.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Path - the name of the directory.
- Mask - mask for searching in the directory. This input is supplied with textual information about the extension of the files to be found. In the format: "*txt" or "*.txt", without quotes.

➢ Output:

- Files - the found files are listed (separated by ;), not just the file name is indicated, but the absolute path to each found file.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- DirectionName (c :\) - the name of the directory in which files will be searched.
- FileExtension (*. txt) - extension of files to be searched. Specified in *txt or *.txt format.
- AddFolderName (true) - add folder name to file name.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Path - The Path value contains the name of the directory (text).
- Mask - The Mask value is used to search in the format: "* txt " or "*. txt ", without quotes.

➢**Custom common properties:**

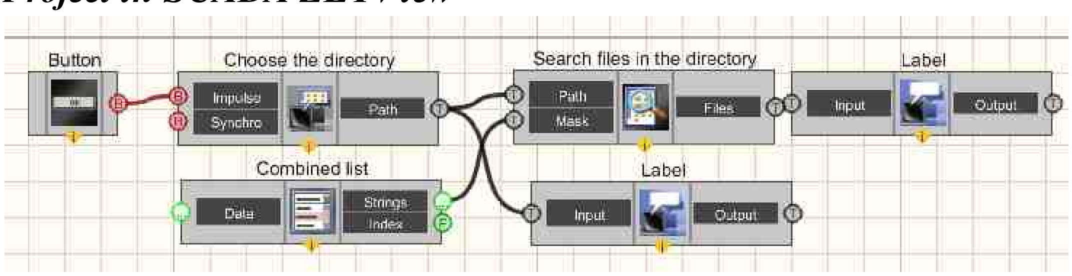
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

- BSTR DirecrionName - Set the name of the directory in which files will be searched (string).
- BSTR FileExtension - Set the extension of the files to be searched (string).
- VARIANT_BOOL AddFolderName - Set to add the folder name to the file name:
  true - the output value is given when adding the folder name to the file name;
  false - no output is produced when appending a folder name to a file name..

Example

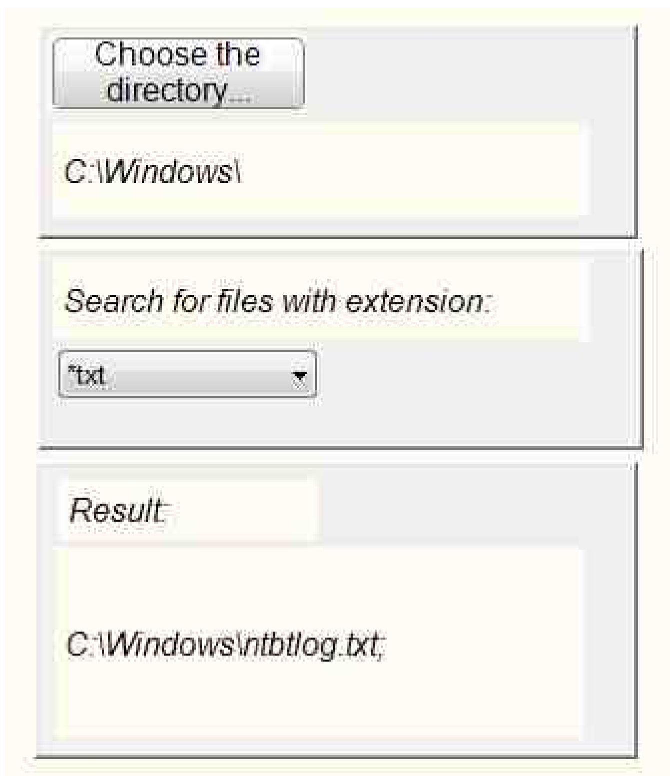
Project in SCADA ZETView



In this project, the user selects the directory and extension of the searched files, after which all the files found by the mask in the specified directory are displayed on the screen. The directory selection is carried out by pressing the momentary button with the inscription Choose the directory 1232. The button sends a control pulse to the Choose the directory 1232 component. Clicking the button will open the standard Windows

Explorer in a separate window and prompt the user to select a directory to search. When the directory is selected press the "OK" button of the explorer. Then the path to the selected directory will appear in the text label. As can be seen from the diagram, the path received by the Choose the directory 1232 component is passed further as a text string to the Label 1087 component, which displays the path selected by the user, and to the Search files in the directory 1232 component, indicating in which directory to search. Also, a combo box is connected to the Search files in the directory 1232 component, which allows the user to select the required file extension from the list. The search result is displayed on a text label.

Project operation result

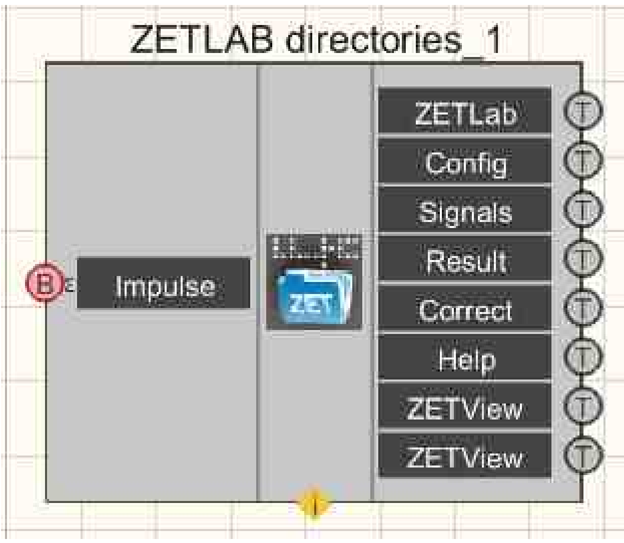


View the example in ZetView

31.9.ZETLab directories

The component "ZETLab directories" it allows to set ZETLab main directory, directories for recorded signals, data processing results, tooltips files, etc. This component contributes to convenience of using standard ZETLab directories; it is used for report purposes, file operations, etc.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Impulse - impulse to read paths ZETLab.

➢ Output:

- ZETLab - ZETlab root directory.
- Config - let to the folder with the current list of devices.
- Signals - path to the folder with recorded signals.
- Result - path to the folder where processing results are stored.
- Correct - path to a folder containing custom corrections.
- Help - path to ZETLab help files.
- ZETView - ZETView directory.
- ZETView - ZETView Help.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

Doesn't have.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

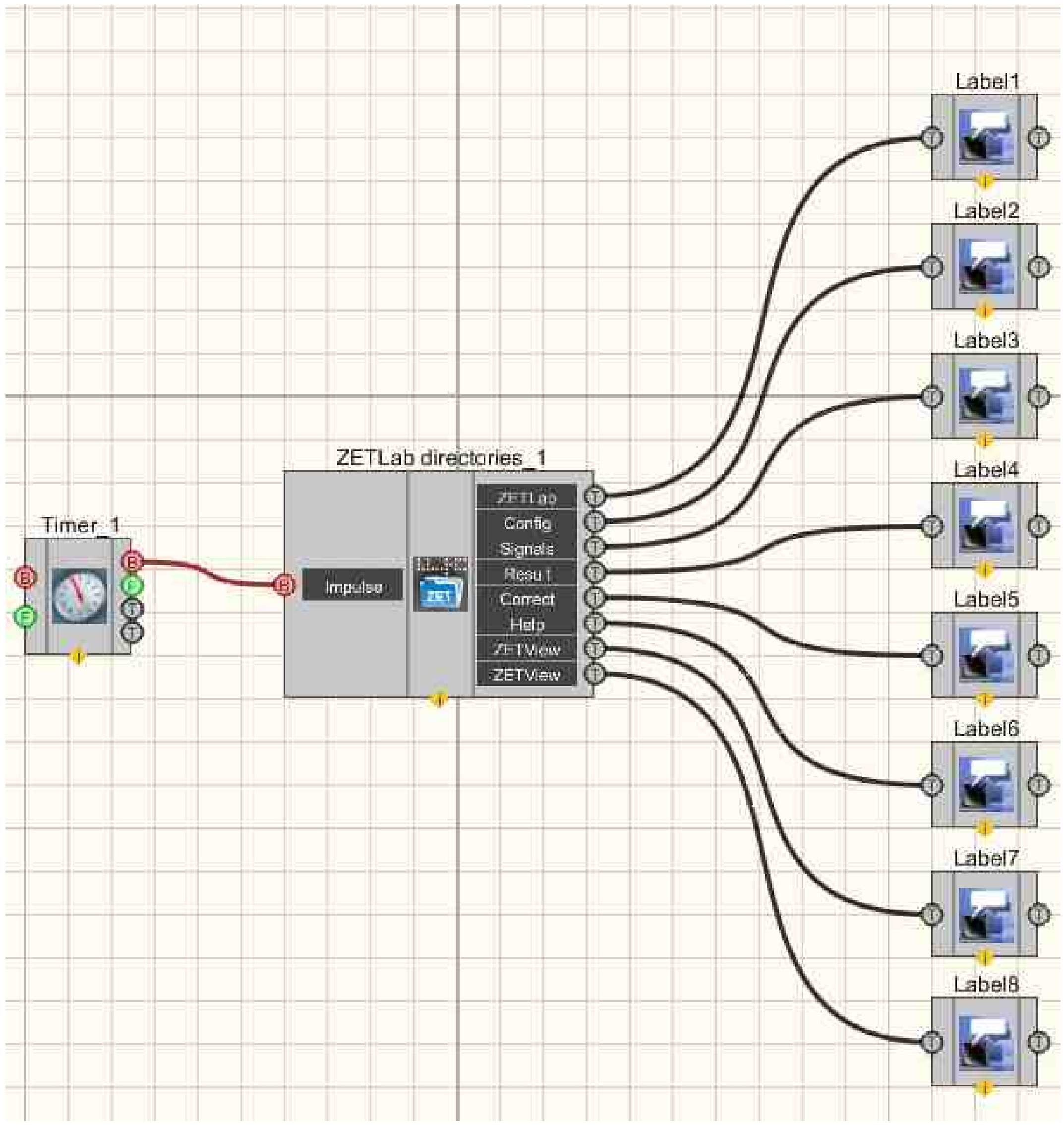
Setting:

➢**Custom common properties:**

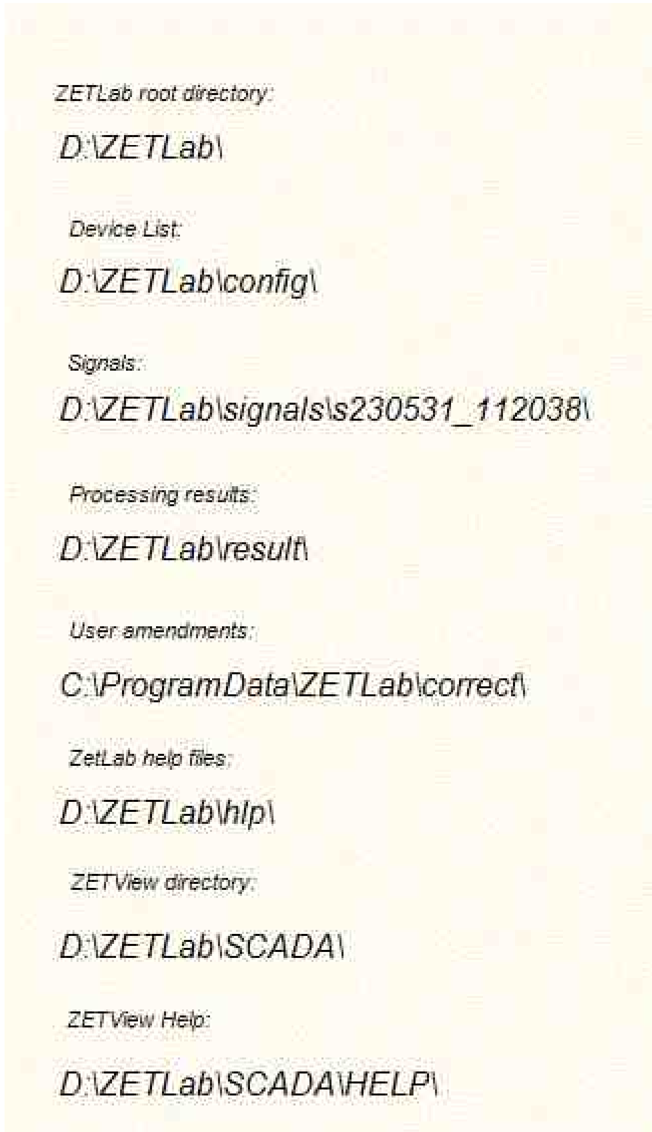
- Are available by the link below. 154

Example

Project in SCADA ZETView



Project operation result



View the example in ZetView

31.10.Structural recording

The "**Structural recording**" and "**Structural reading**" components are designed to Signals recording and Play recorded data transmitted between the components of the Scada-system ZETVIEW.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➤Input:
- Data - input data is being received.
- ON\OFF - enable / disable structure recording.

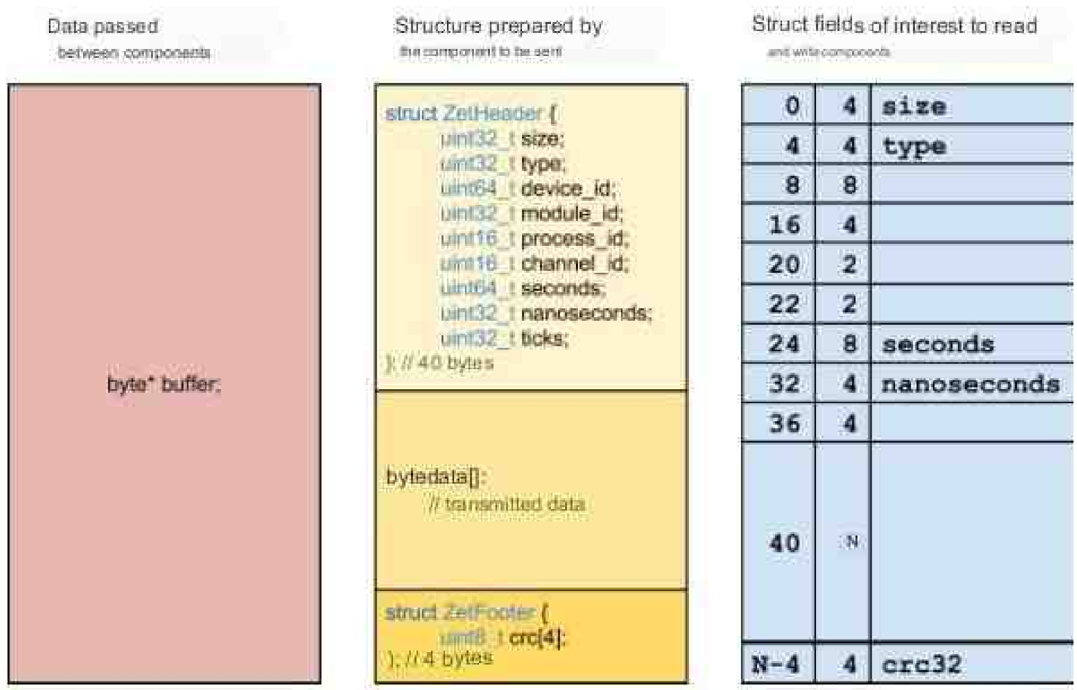
Custom common properties:
➤Total properties (environment):
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**
- FileName - the name of the files to record the structure.
- WriteMethod (Add to end) - set and read the recording mode: Add to the end; Overwrite.
- WriteSQLite (false) - record structures to SQLite file.

Algorithm of the component "Structural reading"

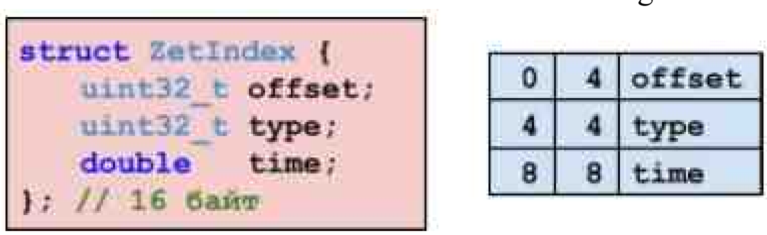
The structure of the recording files:

Structures coming to the input of the "Structural recording" component are recorded to the file as a stream of bytes.
The structures to be passed must conform to the following schemes:



1. size - the size of the transferred structure,
2. type - structure descriptor number registered in the corresponding document,
3. seconds - time in UNIX format (integer number of seconds since January 1, 1970),
4. nanoseconds - number of nanoseconds,
5. crc 32 - checksum calculated over the first N -4 bytes, that is, over the entire structure, not including the checksum bytes.

The index file must consist of structures of the following form:



1. offset - offset relative to the beginning of the file, indicating the beginning of the structure corresponding to the index,
2. type - structure descriptor number registered in the corresponding document,
3. time - time of getting the structure according to server time in DATE format (number of days since midnight December 30, 1899).

The algorithm of the "Structural recording" component operation:

1. When starting the project, the component must open the file specified in the properties for writing. The component must also create (or open an already created)

index-file for writing. The index file must be located in the same directory and have the name “record_file_name” + “record_file_extension” with the extension “. IDX ".

2. When the project is stopped, the component should close the files.
3. When the component is enabled, the recording thread should start. When the component is turned off, the recording stream must be closed. While the project is not running, the thread does not start.
4. When the next structure arrives at the input of the component, the structure must be written to the end of the queue so as not to slow down the work of the entire scada. Also, upon admission, the next indexing structure is calculated. The received structure and its index must lie in the queue as a pair.
5. During the next cycle of the record stream, the structure and its index must be extracted from the head of the queue and written to the appropriate files. If there is a suspicion of an error in Set the structure (the size does not match), then the structure and index should be removed from the queue without writing to a file.
6. If the queue is empty, then the thread should be idle.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Data - The data value contains the input data to be written to the structure (of any type).

➤**Custom common properties:**

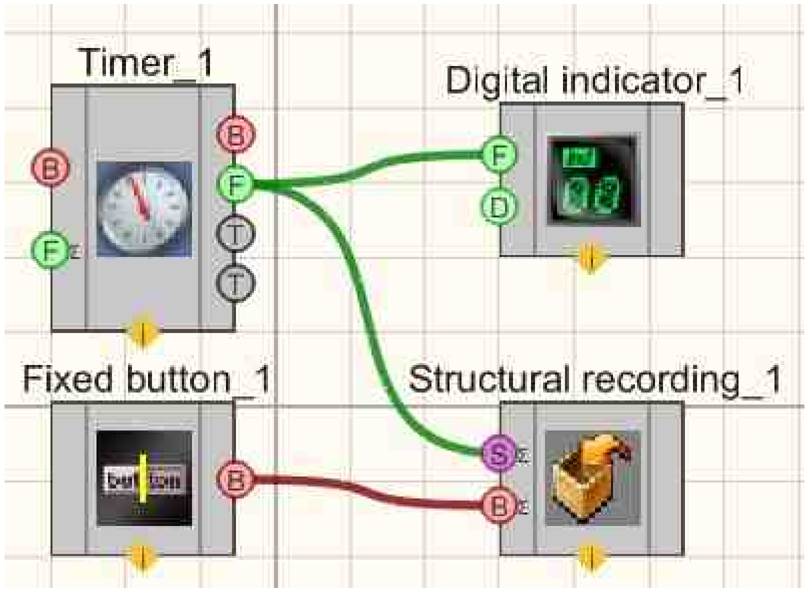
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR FileName - Set the file name for writing structures to a file in an arbitrary *.* format. (string).
- BSTR WriteMethod (Add to end) - Set and read the recording mode: Add to the end; Overwrite(string).
- BSTR WriteSQLite (false) - Set the record structures to SQLite file (string).

Example

Project in SCADA ZETView



Project operation result

31.11.Structural reading

The "**Structural recording**" and "**Structural reading**" components are designed to Signals recording and Play recorded data transmitted between the components of the Scada-system ZETVIEW.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- ON\OFF - enable / disable structure recording.
- Synchro - External impulses for synchronizing the reading of structures.

➢ Output:

- Data - output data is coming.
- Time - The timer for which structures are recorded.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- FileName - the name of the files for reading the structure in the format of writing to the structure.
- ReadMode (By impulse) - set and read of the read start mode: By impulse; By change.
- SynchroMode (Uniform feeding) - set and read the synchronization method: Time of structures; External impulses; Uniform feeding.
- StartTime - set and read the start time of the structure (the time the structure was recorded).
- FinishTime - set and read the end time of the structure (the time the structure was recorded).
- SortStructureTime (false) - set and read c sorting of read structures by time from structure header.
- FilterProgramID - set and read the filter by program IDs.
- FilterModulesID - set and read of the filter by program module identifiers.
- FilterProgramNumber - set and read the filter by program numbers.
- FilterModuleNumber - set and read the filter by program module numbers.
- FilterTypeStructure - set and read the filter by structure types.

Algorithm of the component “Structural reading”

1. When starting the project, the component must open the file with the recorded structures and the index file. In the properties, the name of the file with the recorded structures is set, and the name of the index file is determined automatically by adding the extension “. IDX ".
2. When the project is stopped, the component should close the files.
3. When you turn on the project, you need to start the reading flow. When the component is turned off, the flow should stop.
4. The pointer to the current reading location must remain unchanged when the component is turned on or off. When the project is stopped, the pointer must be reset to the beginning of the file.
5. The reader thread must first read the index file, then read the file with the written structures. The next read index and the next read structure must correspond to each other, that is, the structure must start from the place specified in the offset field of the index structure. If the structures do not match, both files must be read in the same direction until the first time match is found.
6. Sending read structures must be synchronized with the server time. When the component is turned on for the first time, it is necessary to remember the current server time and then subtract it during calculations. The send time should be the value of the time field of the index structure. The first time read from the index structure must also be remembered and subtracted from the next times in the indexes so that data can be sent in sync at any time and at any server speed. Until the time specified in this field arrives, the thread must be in standby mode.
7. When it is time to send the next structure, you need to send the structure.
8. When the end of one of the files is reached, the component should stop.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

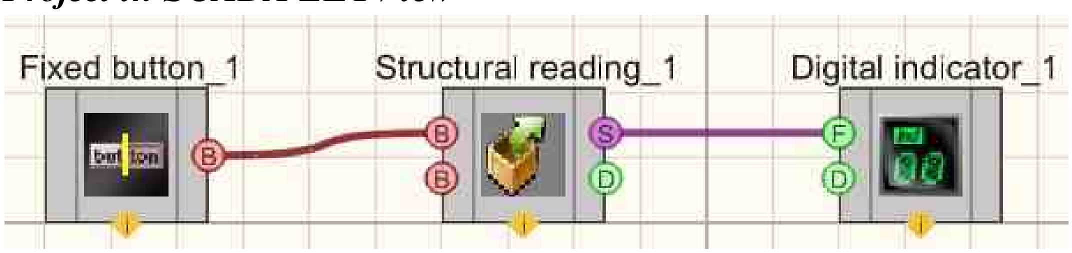
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- BSTR FileName - the file name for reading the structure in the format of writing to the structure (string).
- BSTR ReadMode (By impulse) - set and read the start reading mode: On impulse; By change (string).
- BSTR SynchroMode (Uniform feeding) - set and read the synchronization method: Time in structures; External impulses; Uniform feeding (string).
- BSTR StartTime - set and read the start time of the structure (the time the structure was recorded) (string).
- BSTR FinishTime - set and read the end time of the structure (the time the structure was recorded) (string).
- BSTR SortStructureTime (false) - set and read sorting of read structures by time from structure header (string).
- BSTR FilterProgramID - set and read the filter by program IDs (string).
- BSTR FilterModulesID - set and read the filter by program module identifiers (string).
- BSTR FilterProgramNumber - set and read the filter by program numbers (string).
- BSTR FilterModuleNumber - set and read the filter by program module numbers (string).
- BSTR FilterTypeStructure - set and read the filter by structure types (string).

Example

Project in SCADA ZETView



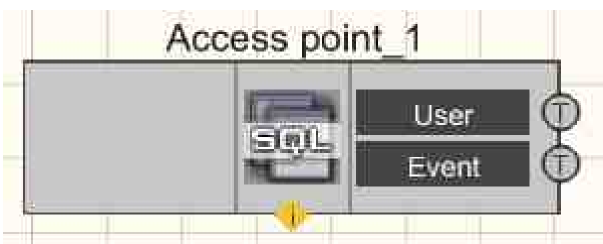
Project operation result

View the example in ZetView

31.12.Access point

The component **"Access point"** is used to select the current user (operator, engineer, etc. with corresponding access rights). The name of the selected user can be saved to database and be used for personal informational notification within the framework of complex systems, or be saved in reports made in the course of usual activities.

Appearance of the component:

Developer interface	Operator interface
	

Setting:

➢ Output:
- User - user data is set;
- Event - event data is set.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- IP - IP address of the computer on which the database is installed.
- DSN is the name of the DSN data source .
- Database - the name of the database to which you want to connect.
- TableUsers - table to which we connect users.

- TableGroups - table to which we connect user groups.
- User - user (name entered during MySQL installation).
- Password - user password.
- NeedLogin (false) - authorization window at startup.
- EnableExit (true) - enable the ability to exit.
- ShowUsers (false) - show a list of users on login
- ComplexPassword (true) - Require a complex password.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Custom common properties:**

- Are available by the link below. 154

➢**Adjustable custom properties:**

- BSTR IP - Set the IP address of the server (string).
- BSTR DSN - Set data source (DSN) (string).
- BSTR Database - Set the name of the database (string).
- BSTR TableUsers - Set the user table name (string).
- BSTR TableGroups - Set the group table name (string).
- BSTR User - Set username (string)..
- BSTR Password - Set user Set the authorization window at startup:
  true - the authorization window at startup is enabled;
  false - the authorization window at startup is disabled.
- VARIANT_BOOL EnableExit - Set the exit prohibition property:
  true - the exit prohibition property is enabled;
  false - the exit prohibition property is disabled.
- VARIANT_BOOL ShowUsers - Set to show a list of users on login:
  true - The state to show a list of users at login is enabled;
  false - The state to show a list of users at login is disabled.
- VARIANT_BOOL ComplexPassword - Set to require a complex password:
  true - State to require complex password;
  false - State not to require complex password.

Operation algorithm of the “Access point” component

The component is based on the MySQL database. The MySQL database is configured on a local or remote machine with the following options:

- ✓ The number of fields is 4, the first field is an integer type INT, the rest are character VARCHAR(64).
- ✓ Field No. 1 key index field, -
- ✓ Field No. 2 - The group to which the user belongs (possibly a position),
- ✓ Field No. 3 - Username,
- ✓ Field number 4 - User password.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
#	INT(10)	✓	✓	✓		✓		✓
Name	VARCHAR(64)		✓					
Group	VARCHAR(64)		✓					
Password	VARCHAR(64)		✓					

Explanation of the settings: RK - flag of belonging to the main key, NN - non-null field, must be filled in, UQ - index uniqueness flag, UN - flag that the value is an unsigned integer type, AI - auto-increment.

Setting up databases is not covered in this document. As a rule, databases are configured by an administrator. The MYSQL database can be configured in console mode with specialized commands in SQL syntax, or in a specialized graphical shell in which most of the actions are implemented using a graphical user interface. MySQL Workbench is an example of such software.

#	Name	Group	Password
1	Ivanov A.A.	Operator	*******
2	Kovalev A.I.	APCS Engineer	*******
3	Ustinov A.Yu.	APCS Engineer	*******
NULL	NULL		NULL

For the correct operation of the component, it is necessary to create a database and configure the component itself to access it by specifying the following settings:

IP - The IP address of the computer, if it is a local computer, you can enter localhost or specify 127.0.0.1

DSN - database source name

Database - the name of the database

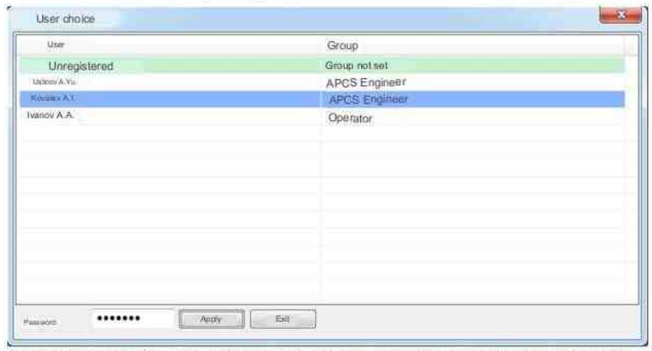
Username of the database administrator

Password - password of the database administrator

Table - the name of the table, which lists the names of users with the indication of the Group (position) and password.

If there is no database, the Unregistered user without specifying the Group will always be used.

Clicking the Register button establishes a connection to the Database specified in the component settings. At



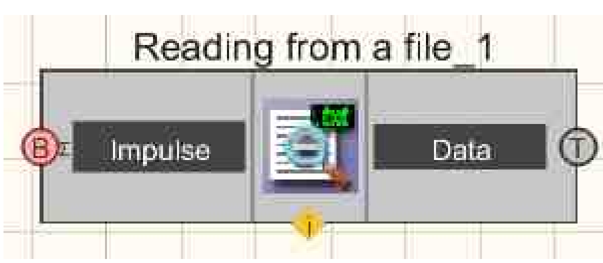
pressing the Exit button will automatically switch to an unregistered user, without the need to select this item from the table.

When you select a user, you will be prompted to enter their password. The currently selected user is indicated in green. Blue cursor.

31.13.Reading from a file

The component **"Reading from a file"** allows to read data from files in the course of project operation and to get access to dynamic data, that is not related to project operation.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Impulse - open the specified file and reads the data, then sends the data to the output.

➢ Output:
- Data - a text string is received from the specified txt file.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- FileName - the name of the file.
- codename (UTF-8) - encoding (UTF-8/ASCII).

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:
- Are available by the link below. 154

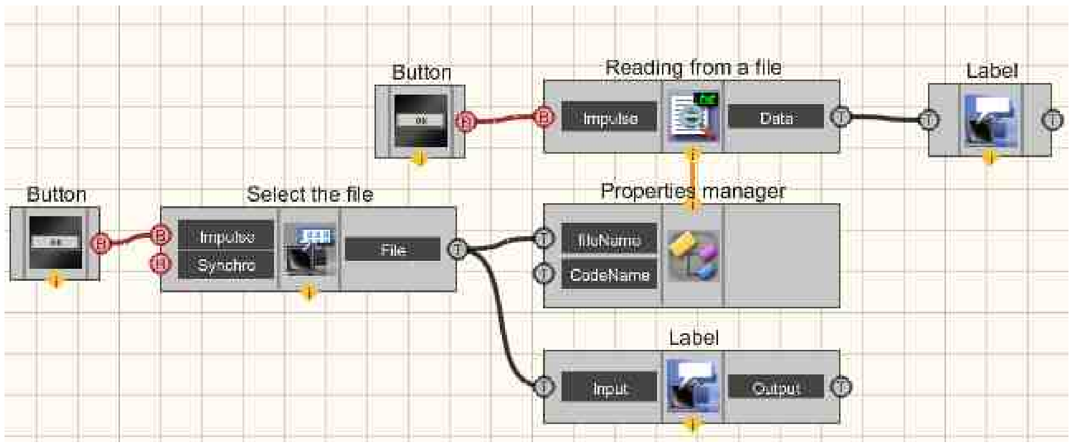
➢**Adjustable custom properties:**

- BSTR fileName - Set file name (string).
- BSTR CodeName - Set encoding: ASCII ; UTF -8 (string).

Example

Project in SCADA ZETView

In this example, an arbitrary txt file is used, which is not included in the example. When running the example, to see the result of its work, you must select any txt file on your computer using the " Open " button, the file path will be displayed below. And then click the "Read File" button. The result may differ from this example, depending on the contents of the selected .txt file.



This diagram shows the operation of the Select the file 1249.

When the Non-fixed button 694 is pressed, a impulse is sent to the input of the Select the file 1249, as a result of which the explorer opens to select a file. The path to the Select the file 1249 is displayed with the Label 1087. Also, a string containing the path to the file is sent to the Property Manager 999 connected to the Read from file 1249 component. When receiving a control pulse from a Non-fixed button 694, this component reads the contents of the selected file and outputs the result using the Labels 1087.

For this example, you can use any txt file.

Project operation result

View the example in ZetView

31.14.Reading system log

The component "**Reading system log**" is a component. They contain information about events taking place in the computer. This information is useful for diagnosing problems, especially on servers. Designed for viewing and managing event logs. This is an indispensable tool for monitoring the performance of the system and troubleshooting problems that have arisen. The Windows service that manages event logging is called the "Event journal". In the event that it is running, Windows writes important data to the logs.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

Doesn't have.

➢ Output:

- Date/Time - Enters the Date/Time of the event.
- Event - the text string of the event is received from the Windows system log.
- ID - the ID of the event arrives.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- SourceName (Application) - the name of the message source.
- Path - read source: Application; System.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values..

Setting:

➢Custom common properties:

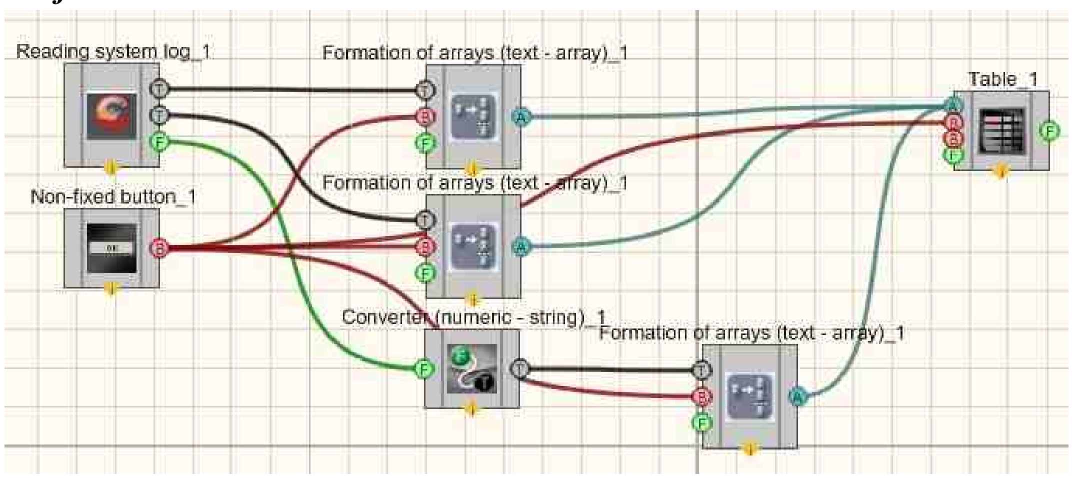
- Are available by the link below. 154

➢Adjustable custom properties:

- BSTR SourceName - Set the name of the message source" (string). .
- BSTR Path - Set read source (Application): Application ; system (string). .

Example

Project in SCADA ZETView



Project operation result

View the example in ZetView

Глава 32.Digital port

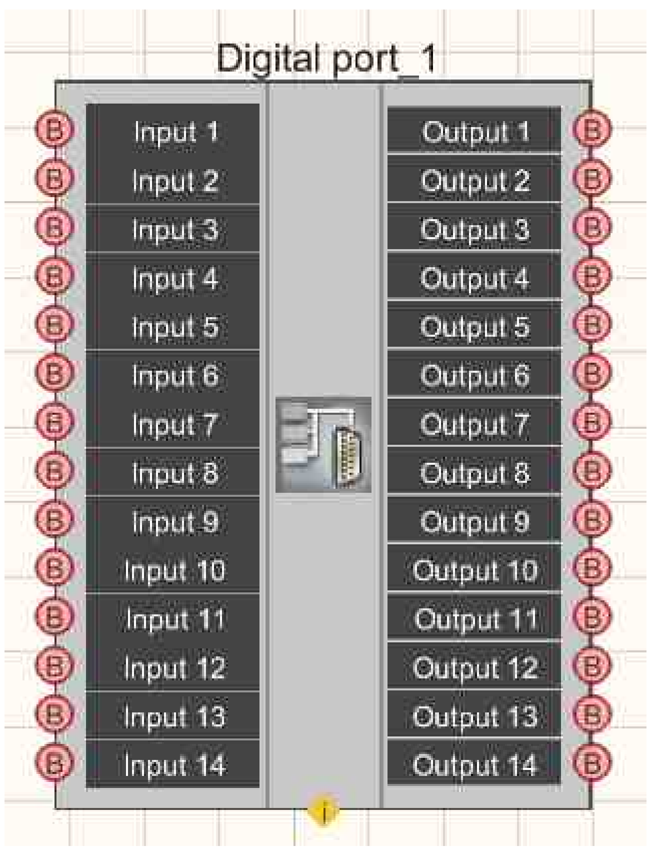
32.1.Digital port

The component **"Digital port"** allows you to work with discrete input and output signals.
The contacts that correspond to 0 in the mask are in a status ready to receive data.
The contacts that correspond in mask 1 are in a status ready for data transmission.

Note: when the mask is 0, then the output of the contact will be logic "1" (3.3V), and when the mask is 1, then the output of the contact will be logic "0".
ZETLAB Software User Manual "page 288 "1.2.3 Digital Port"

In order for Input X to receive and react to a digital signal as an input, it is necessary to select the Device Manager item in ZETLab in the Service group of programs and set the port type to Input or Output (receiving or output). "Digital port" component with a fixed number of pins equal to 14.

Appearance of the component:

<table>
<tr><th>Developer interface</th><th>Operator interface</th></tr>
<tr><td>
</td><td>Doesn't have</td></tr>
</table>

Setting:

➢Input:
- Input 1 - input channel, which receives a discrete signal.
- ...
- Input 14 - input channel, which receives a discrete signal.

➢ Output:
- Output 1 - output channel, which, depending on the input signal, appears high or low logic levels.
- ...
- Output 14 - output channel, which, depending on the input signal, appears high or low logic levels.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- Device (0) - the serial number of the device is set.
- OutputMask (65535) - digital port output mask.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Custom common properties:
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- LONG Device - set the serial number of the device is set. (number).
- LONG OutputMask - Set the digital port output mask (number).

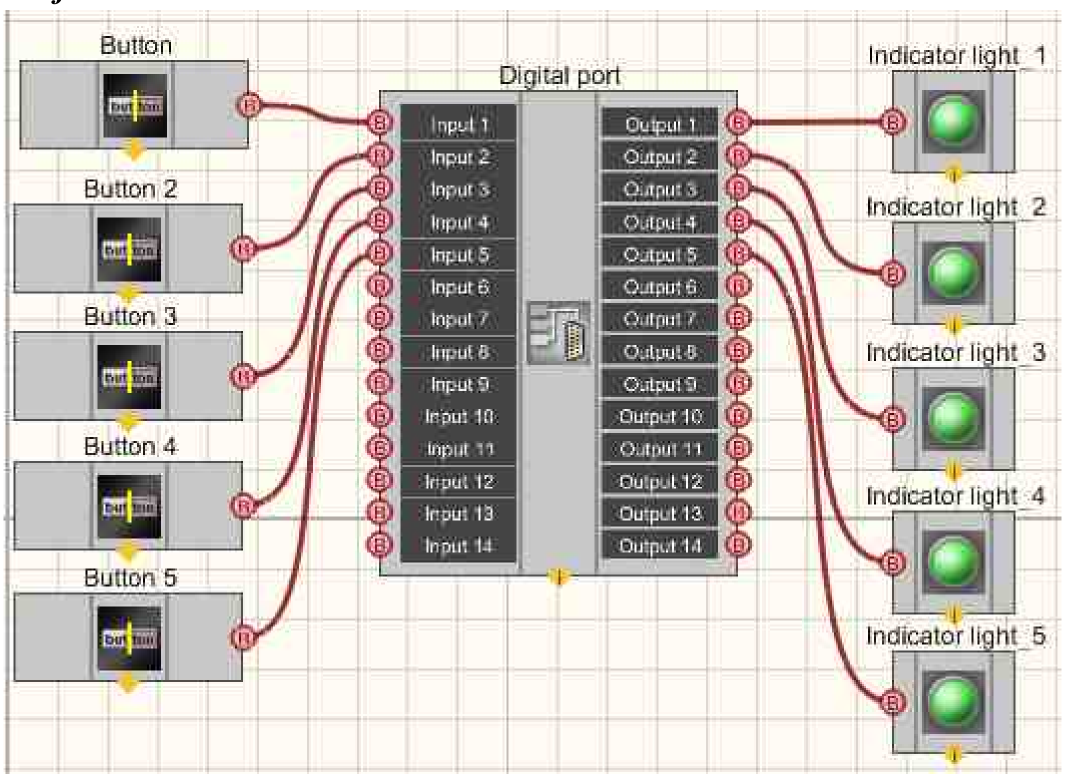
➢Methods:

- void SetInd(LONG num, VARIANT_BOOL status) - Set the status of a digital pin.

- void SetAll(void) - Set the logic-one status of all digital pins.
- void ResetAll(void) - Set the logic zero status of all digital pins.
- void InvertPin(LONG num) - Set the digital pin to be inverted.

Example

Project in SCADA ZETView



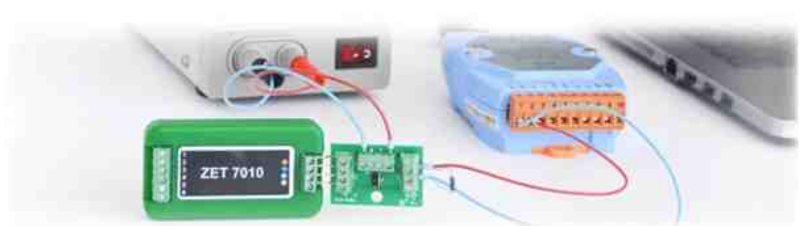
This project shows the operation of the Digital port 1255 component is a bridge between a digital input and an output. The Fixed button 691 is used you to control the status of the input. The Light indicator 669 indicates the status of the input.

Project operation result

View the example in ZetView

Глава 33.OPC

Connecting ZETSENSOR to third-party systems



How do I integrate a specific type of measurement equipment into an existing measurement system? One of the questions that torments more than one generation of engineers.

This task did not go unnoticed for our company and the solution to this issue is ZET 7XXX intelligent sensors with the ability to connect to third-party systems.
Briefly about the main thing

The main requirement for implementing a new meter in distributed systems is the availability of a universal interface for data exchange. The most widely used open industrial network standard is still the Modbus protocol, which can be used for data transmission via serial communication lines RS-

485, RS-422, RS-232, as well as TCP/IP (Modbus TCP) networks. One of the advantages of Modbus is that there is no need for special interface controllers, simple software implementation and elegant operating principles. And the most reliable and secure protocol is the CAN protocol used in the field of industrial automation.

Since ZETSENSOR devices also use Modbus and CAN protocols, and data transmission is organized via RS-485 and CAN interfaces, our intelligent sensors are perfectly suited for automating measurement processes on an industrial scale.

General description and characteristics of devices

- ability to work with third-party systems;
- supply voltage 9-24 V;
- communication interfaces RS-485, CAN;
- Modbus RTU and CAN protocols.
- operating speed 57,600;
- parity is enabled.
- checking parity for oddness.
- number of digits: 8.
- floating-point numeric data.

Features of smart sensor management

The industry-standard Modbus protocol simplifies working with smart sensors by allowing data to be output and read using a large number of third-party Modbus programs.

Information output is a function of writing a number in float format to a device with a given address, simplifying the output of information to the limit. The sensors can also be used in most SCADA systems.

Connecting at the hardware level

Fig. 1. Connecting the ZET 7010 smart sensor to a third-party system

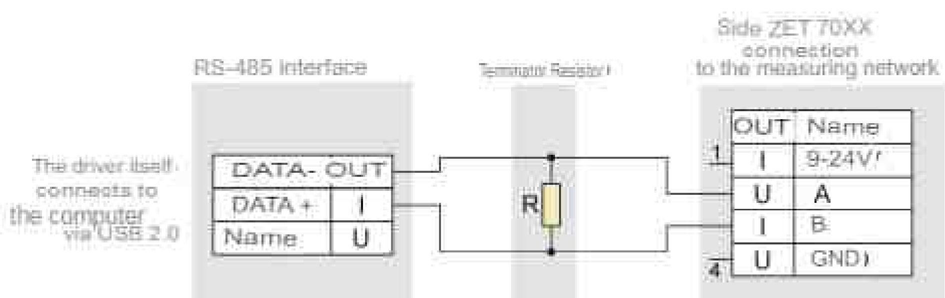
Ability to connect modules with the RS-485 interface to third-party systems

As mentioned above, ZETSENSOR devices implement an open Modbus communication protocol with a standard set of commands.

The master device is a controller, whose main task is to provide read/write operations, and manage the structure of data placement. The controller has a finite number of inputs and outputs, and sensors connected to them. Its programming, diagnostics and maintenance are performed by connected personal computers or laptops for this purpose.

I / O modules, namely ZETSENSOR smart sensors, are the slave devices of the industrial network. It is possible to connect the sensors directly to the controller using standard data reception/transmission protocols via RS-232 and RS-485 interfaces, or using interface converters.

The protocol reads and writes data to the controller registers. The data required for working with the device is stored in its internal memory in C structures. To read these structures, the standard Modbus read commands (Read Holding Registers and Read Input Registers) are used. Reading is performed from the null address.



The ability to connect modules with a CAN interface to third-party systems

CAN real-time protocol, used for communication of intelligent sensors in distributed industrial networks and performs data transmission with a very high degree of reliability and security.

Similarly, the Modbus protocol in CAN applies the "master / slave" type of communication between the transmitting and receiving device. CAN controller equipment must be used as the master, and ZETSENSOR sensors with a CAN communication interface are the slaves.

Connecting at the software level

Transfer results to third-party systems using the ZETVIEW SCADA component "OPC Server"

The OPC server is widely used in industrial automation. It provides data exchange between the client program and physical devices.

The ZETLAB software provides three types of programs that organize interaction over OPC:

1. The ModbusOPC programModbusOPCsupports data exchange over the Modbus protocol. When such a device is connected to a computer, the ZETLAB ModbusOPC program,which is a full-fledged OPC server, is automatically launched. A block diagram that allows you to understand the principle of data exchange is shown in Fig. 2.
2.

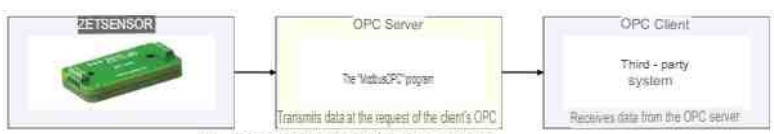


Fig. 2. Block diagram describing data exchange

2. The OPC server "ZET. OPC" from ZETLAB is intended for connecting hardware to third-party software, if it meets the OPC standard (Fig. 3).

Fig. 3. Operation of the "Data exchange with OPC" component and the ZETLAB OPC server

3. ZETVIEW SCADA component "OPC Client" - data exchange over OPC. A simplified SCADA project that demonstrates the ability of various components to communicate with OPC to provide calculation results to other systems.

Our OPC server allows, in addition to data exchange with SCADA, to perform the following useful functions:

- create a hierarchical representation of tag names.
- observe tag values.

Also, in accordance with the standard, the OPC server is automatically registered in the Windows registry during installationWindows. The server is started in the same way as any other program or automatically from the client program.

The client program and the OPC server can be installed on the same computer or on different computers on the Ethernet network. Thanks to DCOM technology, which uses Remote Procedure Call (RPCRemote Procedure Call), any OPC client can access any OPC server from any computer. For example, the SCADA in Figure 4 can access the I / O module via the path indicated by the dashed line in Figure 4. PCs and controllers in this architecture can work with different industrial networks.

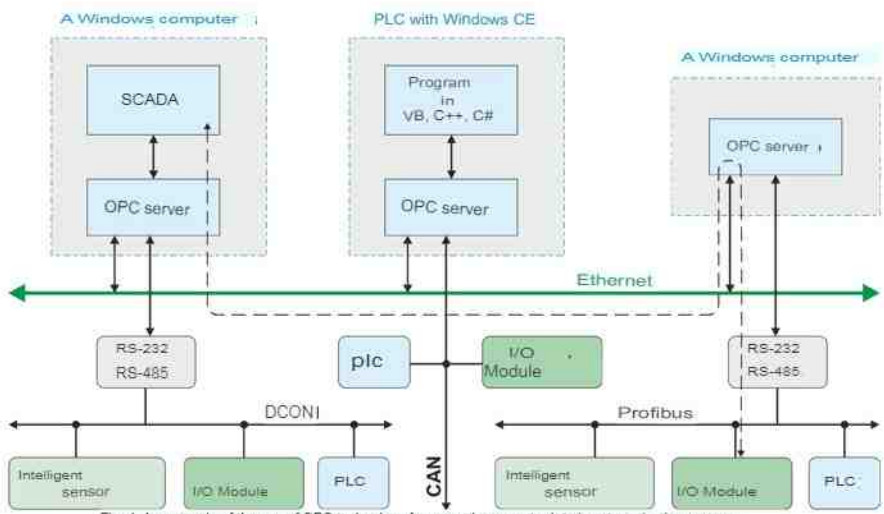


Fig. 4. An example of the use of OPC technology for network access to data in vetomatization systems

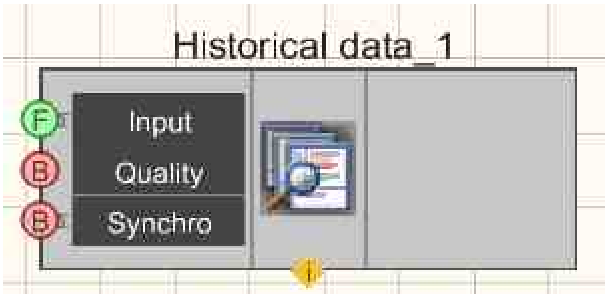
When using equipment from different manufacturers, several OPC servers from different manufacturers can be installed on the computer (controller), but do not forget that the OPC server monopolizes the COM port of the PC (since it continuously updates data), so the number of ports must be equal to the number of OPC servers.

33.1.Historical data

The component "**Digital port**" Historical data is a component. Designed to provide OPC data to clients from ZET70XX series devices.

Appearance of the component:

Developer interface	Operator interface

	Doesn't have

Setting:

➢Input:
- Input - adding a value to the History Server.
- Quality - the quality label of the added value.
- Synchro - record data.

➢ Output:

Doesn't have.

Custom common properties:

➢Total properties (environment):
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):
- IP (164.0.0.1) - IP address of the server.
- DSN is the name of the DSN data source.
- Database - the name of the database to which you want to connect.
- User - user (name entered during MySQL installation).
- Password - user password.
- Table - the table to which we are connecting.
- type (Auto) - synchronization type:
  1. Auto
  2. Manually.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Input - The input number to add value to the History Server (any number).

➢**Custom common properties:**

- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

- BSTR IP - Set the IP address of the computer on which the database is installed (string).
- BSTR DSN - Set the name of the DSN data source (string).
- BSTR Database - Set and replace the name of the database to connect to (string).
- BSTR User - User setting (name entered during MySQL installation) (string).
- BSTR Password - Set the user's password (string).
- BSTR Table - Set the name of the table to which we are connecting (string).
- BSTR Type - Set the type of synchronization: Auto; Manually (string).

33.2.Exchange with OPC

The component "**Exchange with OPC**" from third-party manufacturers using OPC client-server technology.
Allows you to transfer data and quality simultaneously to several OPC servers in different tags.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Tag - data to record to the tag.
- Quality - a mark of the quality of the input data.

➢ Output:

- Output - reading a tag.
- Quality - The label for the quality of the output.

Custom common properties:

➢Total properties (environment):

- Are available by the link below. 154

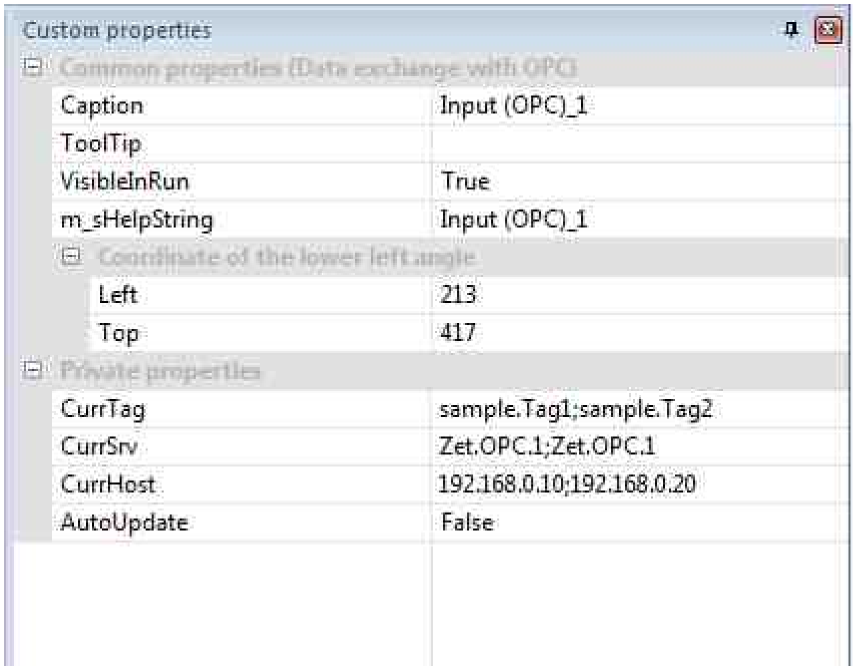
➢Custom private properties (the default value is shown in parentheses):

- CurrTag - tag name.
- CurrSrv - server name.
- CurrHost - host name.
- AutoUpdate (false) - give data constantly.

Algorithm of the component "Exchange with OPC"

In private properties, you can specify several hosts, servers and tags at the same time (the number of parameters must be the same) separated by a semicolon.

Example. Here the data is placed on the ZET OPC server on hosts 164.168.0.10 and 164.168.0.20 in the sample tags. Tag 1 and sample Tag 2.



To demonstrate the operation of the ModbusOPC server, it is enough:

1) Connect the device70 XX to the computer.

2) Run the ZETView SCADA system on another computer and place the “Exchange with OPC ” component on the form

2) Open the component properties window and enter the parameters:

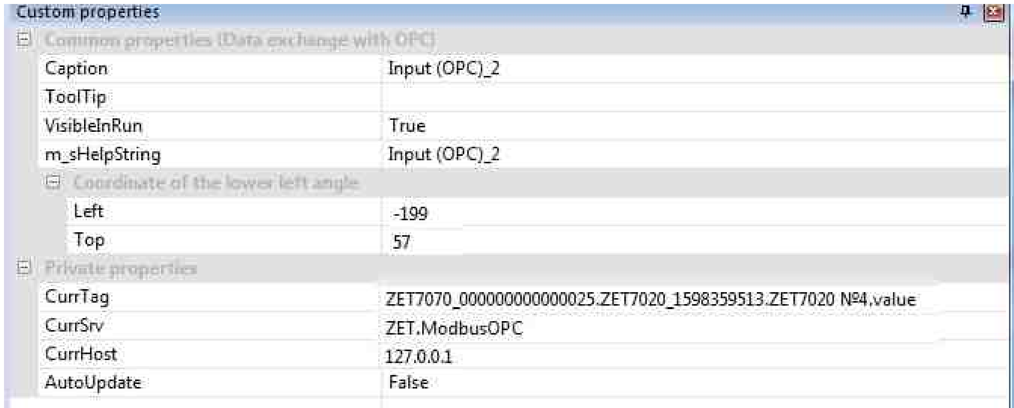
- CurrTag is the full path to the OPC tag.

  For example: ZET7070_000000000000273.ZET7021_268764889.Temperature1.value

- CurrSrv – OPC server software identifier – ZET.ModbusOPC

- CurrHost - enter the name of the remote computer to which the device is connected.

  For example: 164.0.0.1



3) Connect the necessary elements to the output of the “Exchange with OPC ” component (for example, a digital indicator, an array builder, a light indicator, etc.).
4) After launching the ZETView project , it will connect to the OPC server and data exchange starts.

Note:

Fixed the work of the dialog for working with OPC servers.
Tab "Working environment" - button "OCR tags".
There are two servers on the computer
HOST "192.168.0.56"
SERVER 1 " AP . OPCDAServer "
SERVER 2" AP > OPCDAServer .1"
We connect, disconnect, add and edit tags using the context menu, edit the "Exchange with OPC" component.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Tag - The value of the data tag to record to the tag (any type).

➤**Custom common properties:**

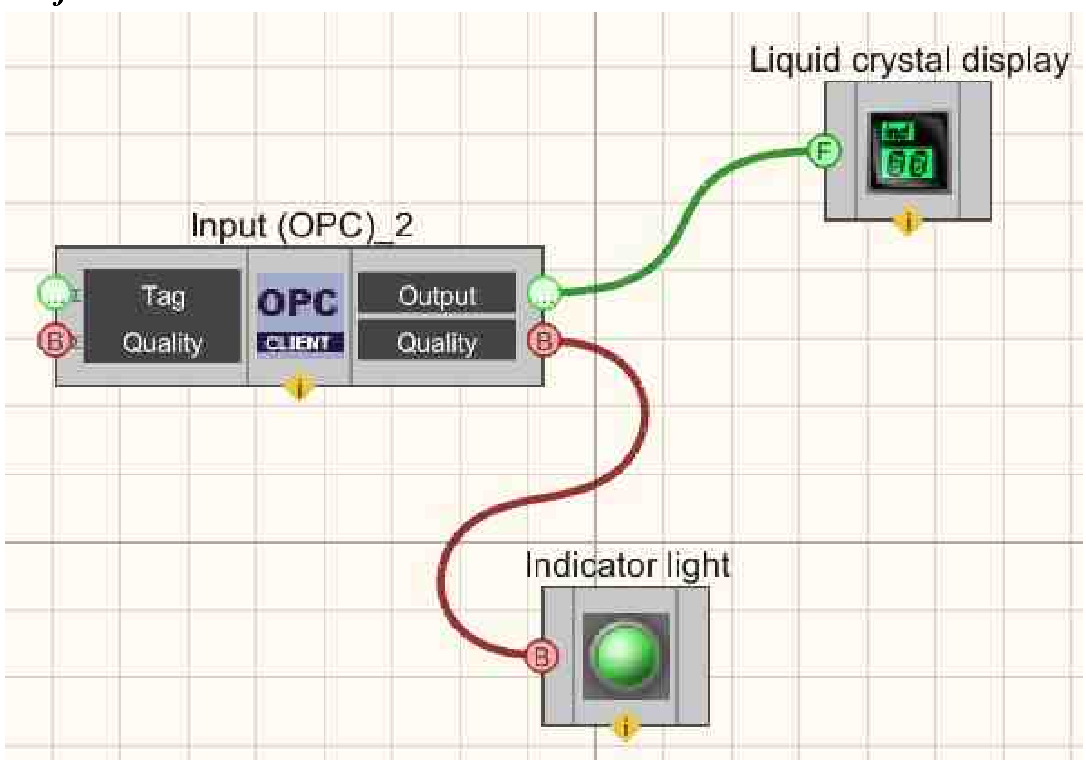
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR CurrTag - Set the tag name (string).
- BSTR CurrSrv - Set the name of the OPC server (string).
- BSTR CurrHost - Set the IP address or computer name (string).
- VARIANT_BOOL AutoUpdate - Set to give data constantly:
  true - the status to give data is always on;
  false - the status to give data is always off.

Example

Project in SCADA ZETView



Project operation result

View the example in ZetView

33.3.Exchange with OPC UA

The component "**Exchange with OPC UA**" designed to connect controllers and converter modules from third-party manufacturers using OPC UA client-server technology.
Allows you to transfer data and quality simultaneously to several OPC servers in different tags.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Tag - data to record to the tag.
- Quality - the quality of the input data.

➢ Output:

- Data - data value from OP U.A.
- Quality - the quality of the output data.

Custom common properties:

➤Total properties (environment):

- Are available by the link below. 154

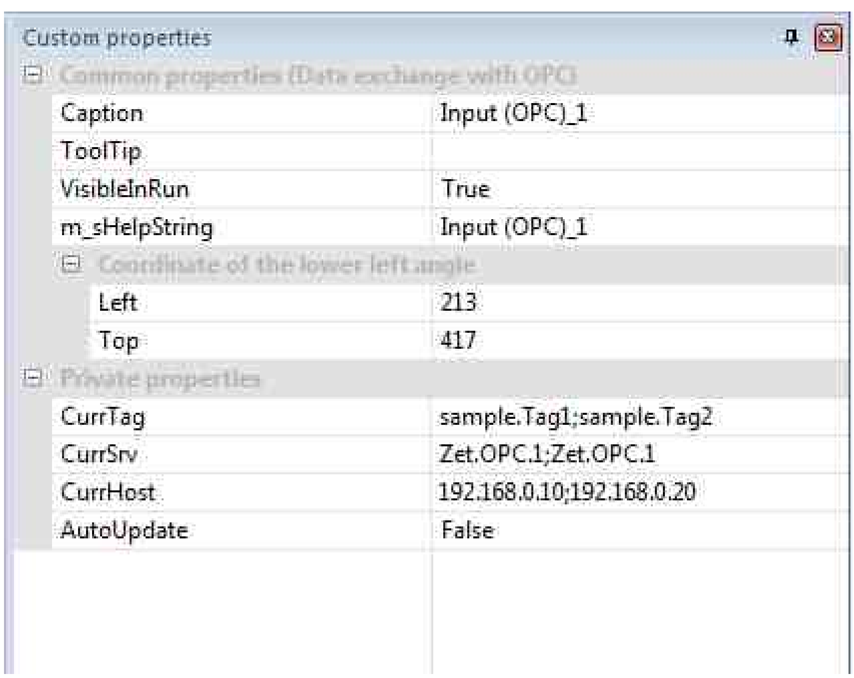
➤Custom private properties (the default value is shown in parentheses):

- Endpoint (localhost) - IP address or computer name.
- NamespaceIndex (0) - address index or computer name.
- NodeType(Numeric) - node type.
- NodeId(Server.ServerStatus.BuildInfo.ProductUri) - name of the node.

Operation algorithm of the component “Exchange with OPC UA"

In private properties, you can specify several hosts, servers and tags at the same time (the number of parameters must be the same) separated by a semicolon.

Example. Here the data is placed on the ZET OPC UA server on hosts 164.168.0.10 and 164.168.0.20 in the sample tags. Tag 1 and sample Tag 2.



To demonstrate the operation of the ModbusOPC UA server, it is enough:

1) Connect the device70 XX to the computer.

2) Run the ZETView SCADA system on another computer and place the “Exchange with OPC UA ” component on the form

2) Open the component properties window and enter the parameters:

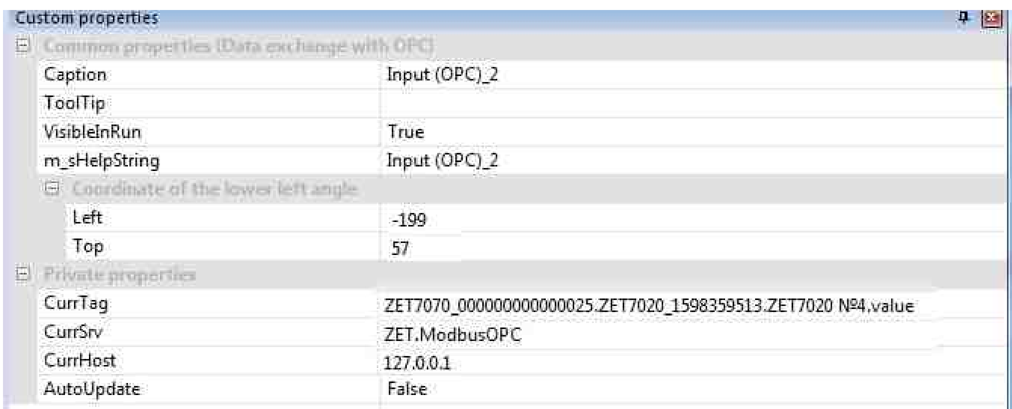
- CurrTag is the full path to the OPC UA tag.

  For example: ZET7070_000000000000273.ZET7021_268764889.Temperature1.value

- CurrSrv – OPC UA server software identifier – ZET.ModbusOPC

- CurrHost - enter the name of the remote computer to which the device is connected.

  For example: 164.0.0.1



3) Connect the necessary elements to the output of the “Exchange with OPC OPC UA” component (for example, a digital indicator, an array builder, a light indicator, etc.).
4) After launching the ZETView project , it will connect to the OPC UA server and data exchange starts.

Note:

Fixed the work of the dialog for working with OPCOPC UA servers.
Tab "Working environment" - button "OCR tags".
There are two servers on the computer
HOST "192.168.0.56"
SERVER 1 " AP . OPCDAServer "
SERVER 2" AP > OPCDAServer .1"
We connect, disconnect, add and edit tags using the context menu, edit the "Exchange with OPC UA" component.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➤**Input:**

- Data - Value to record to OPC UA (any type).

➤**Custom common properties:**

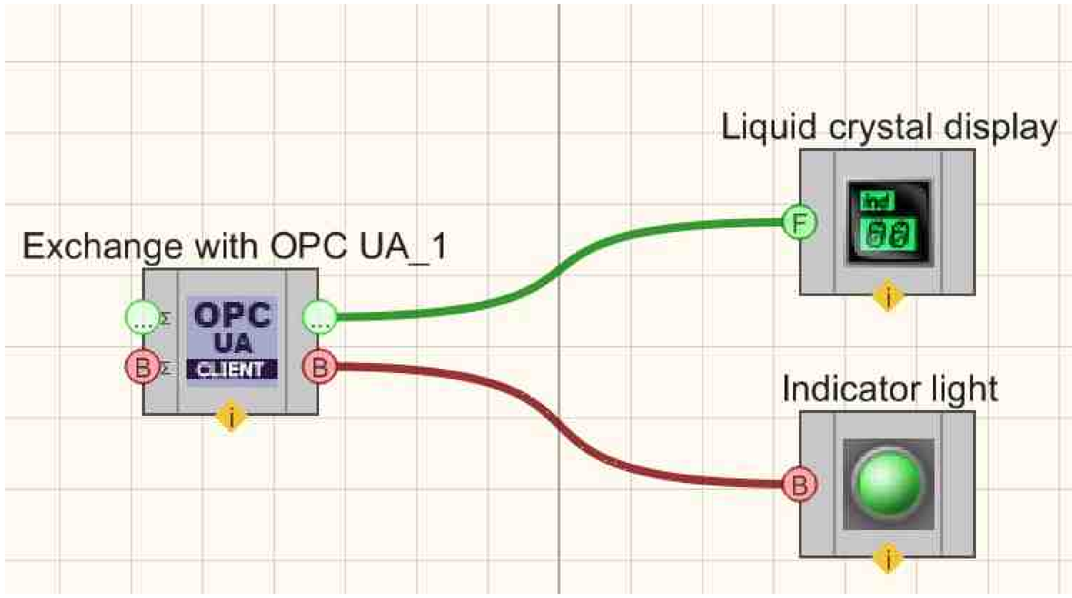
- Are available by the link below. 154

➤**Custom private properties (the default value is shown in parentheses):**

- BSTR Host - Set the IP address or computer name (string).
- BSTR Port - Set the OPC port number Server UA (string).
- BSTR Node - Set the name of the node (string).

Example

Project in SCADA ZETView



Project operation result

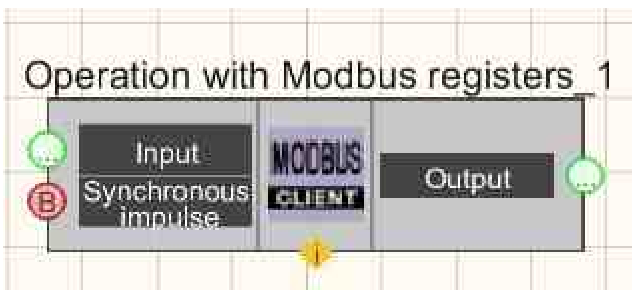
View the example in ZetView

Глава 34.Modbus

34.1.Operation with Modbus registers

The component "**Operation with Modbus registers**" designed to connect controllers and converter modules from third-party manufacturers using Modbus client-server technology;
Allows you to transfer data simultaneously to several Modbus servers.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:
- Input - the data of the input value.
- Synchronous impulse - the impulse on which the server is being polled and output to the output

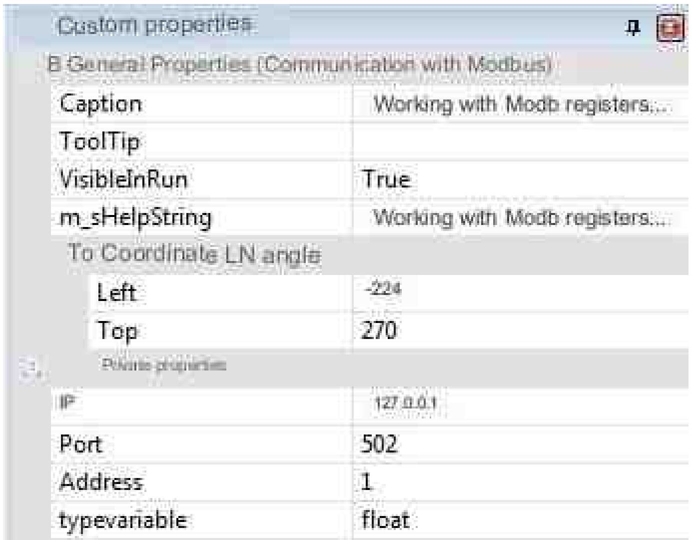
➢Output
- Output - read modbus.

Custom property:
➢Custom common properties:
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- IP - the name of the IP address.
- Port - the name of the port.
- Address - name of the address of the data register.
- TypeVariable - variable type for receiving data.
- Slave(1) - device address

In private properties, you can specify the IP address of the server, port, and address of the data register.

Example . Here the data is put on Modbus c IP 164.0.0.1, port 502, data register address 0, float variable .



Programming

When using a component in a script and a programmable component (script), it is necessary to take into account the ranges of values supplied to the input contacts of the component, the ranges of values of the component properties, as well as the ranges of values of the parameters of the component methods.

Setting:

➢**Input:**

- Input - Data value to pass modbas (any type).

➢**Custom common properties:**

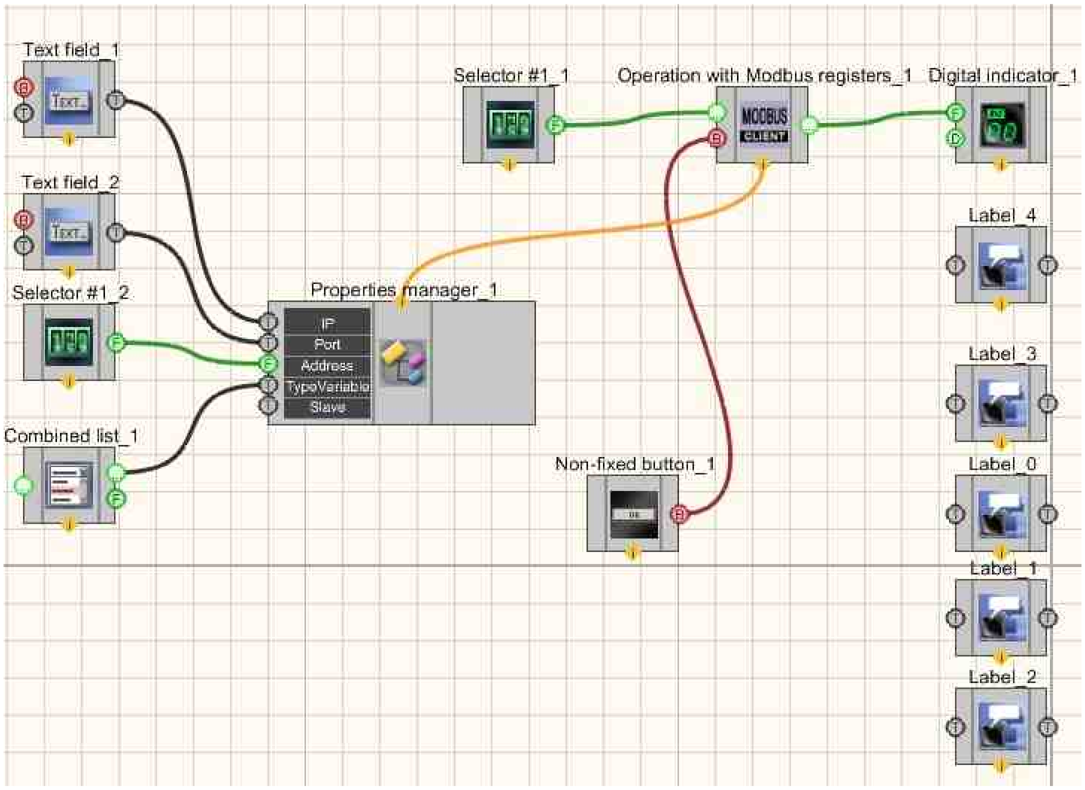
- Are available by the link below. 154

➢**Custom Custom private properties (the default value is shown in parentheses):**

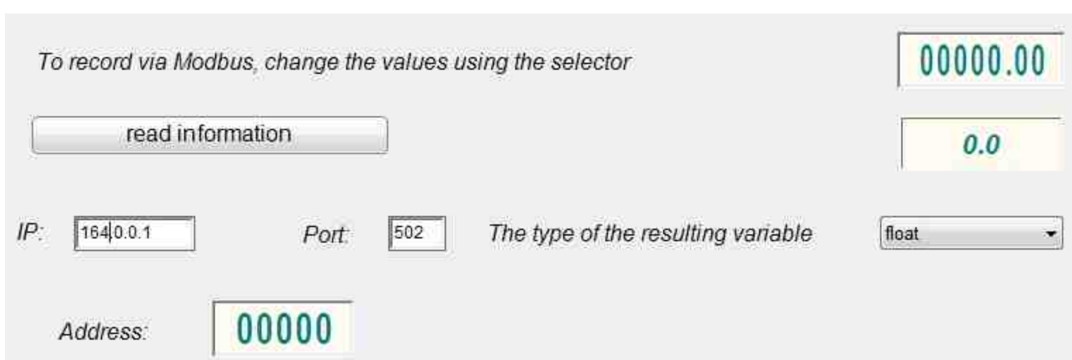
- BSTR IP - Set the IP address (string).
- BSTR Port - Set the port number (string).
- BSTR Address - Set the address of the data register (string).
- BSTR TypeVariable - Set the type of variable to receive data: bool ; float ; double (string).
- BSTR Slave(1) - Set device address.

Пример

Project in SCADA ZETView



Project operation results



View the example in ZetView

Глава 35.Excel

Components of "Excel" section are used for adding Excel-document to the project, creating a link to a particular sheet of Excel-file and data recording to a particular cell:

- Excel document – component for adding Excel-document to the project, output of the component contains a link to Excel-file.
- Excel page establishes connection to a particular sheet of Excel-file, has additional function of printing a page.
- Excel cell – is used for data recording to a particular cell.

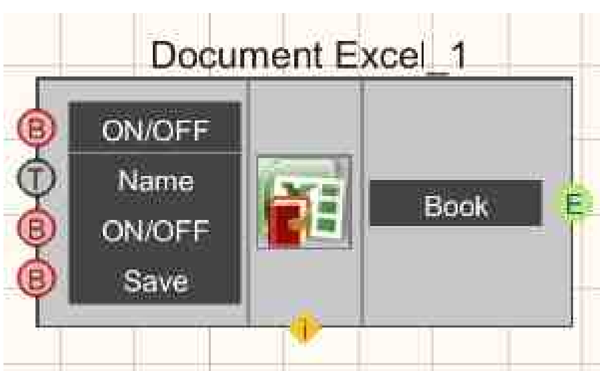
Components of "Excel" section are not depicted in operator mode and can be added to the project only in developer mode.

35.1.Excel Document

The component **"Excel Document"** component for adding Excel-document to the project, output of the component contains a link to Excel-file.

Components of "Excel" section are not depicted in operator mode and can be added to the project only in developer mode.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- ON\OFF - connects/disconnects to the specified Excel file.
- Name - the name of the document to connect to.
- ON\OFF - enable or disable the display of the edited Excel file.
- Save - when a control impulse arrives at this contact, it saves the connected document.
-

➢ Output:

- Book - a pointer to an Excel file is received.

Custom common properties:

➢Total properties (environment):

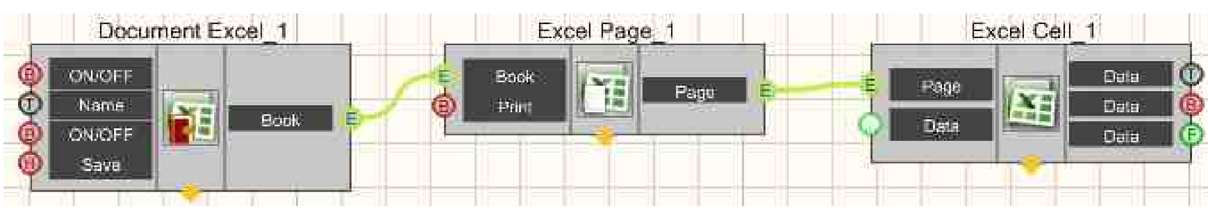
- Are available by the link below. 154

➢Custom private properties (the default value is shown in parentheses):

- ExcelDocFileName - the name of the document to which we connect, if not specified, a blank document is created.
- ExcelDocVisible (false) - a parameter that Set the visibility of the edited excel file. Accepts the values true / false. If the parameter value = true , ZETView will contact MSExcel and open the specified file.
- Password_read - a string containing the password to open the specified Excel file.
- Password_write - a string containing a password to record data to the specified Excel file.

Description:

To record data to an Excel file, it is used in conjunction with the Excel Page 1283 and " Excel Cell 1289 components. You need to connect the components in the following sequence:



This component is more of a building block than a stand-alone component. Required when working with excel files. Although the Excel Document 1277, Excel Page, 1283 and Excel Cell 1289 components have the same type of input and output pins, the data coming from these pins is different. The pointer to the book comes from the Excel Document 1277 component, and the pointer to the page comes from the Excel Page1283 component. Therefore, you cannot connect Excel Document1277 and Excel Cell1289 directly.

The component works stably with the Microsoft version office Excel 2007.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢Input:
- Name - Text title of the document to which you want to connect (text).

➢Custom common properties:
- Are available by the link below.154

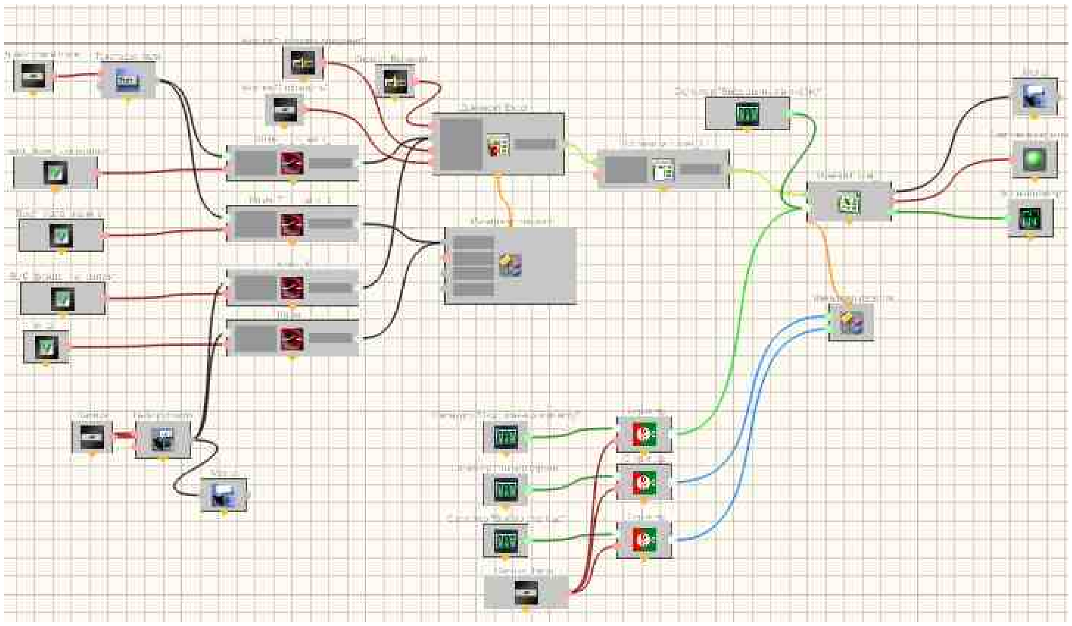
➢Custom private properties (the default value is shown in parentheses):

- BSTR ExcelDocFileName - Set the name of the document to connect to, if not specified, a blank document is created (string).
- VARIANT_BOOL ExcelDocVisible - Set to show or hide the Excel program :
  true - the output value is given when the Show Excel program is enabled ;
  false - no output value is produced when you enable hide Excel
- BSTR Password_read - Set a password to open a file (string).
- BSTR Password_write - Set a password for recorded data to a file (string).

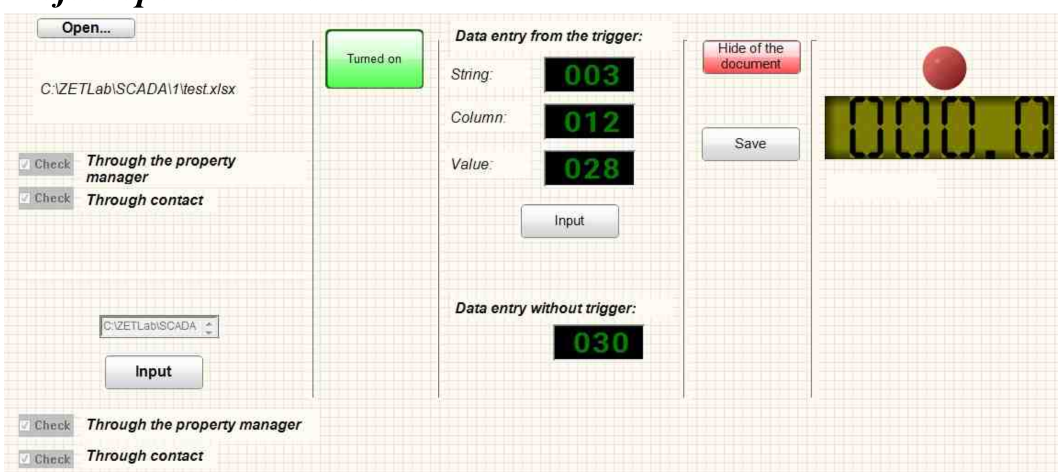
Note: ZETExcelDocument.osx if the file specified in its properties is not found, it will try to find it in the project's exe directory. if it is not there, it will display a message that the file was not found in such and such a place, and if you click "OK", a filedialog will be called with the choice of a file to open. First you need to create and put in the selected folder a

Example No. 1

Project in SCADA ZETView



Project operation result



View the example in ZetView

Example No. 2

To generate a report in an Excel file using ZETView, 3 components from the Excel tab are used :

1. Excel document;

2. Excel page;
3. Excel cell.

For more detailed information on the operation of these components, select the component you are interested in and press F 1.

Report save algorithm

1. First you need to specify the path to the file for the report. It can be the full path to the file or just its name. If you specify an incomplete path, the program will look for the file in the directory where the running project is located. If the file does not exist in the specified location, it will be created automatically.

2. Getting started with a document, first you need to open it by setting the value "True " to the first input of the Excel Document component .

3. The cell where the user wants to record data must then be selected by selecting a page in the document using the SheetNumber private property of the Excel Page component. Then, using the private properties of the " Excel Cell " component " Line " and " Col ", select the row and column with the cell in which the information will be written.

4. To record data to a cell, it is necessary to submit the text to the second input of the Excel Cell component, which will be written to it. After that, if necessary, you can save the changes in the document by applying an impulse to the 4th input of the Excel Document component or do it later when closing the file.

5. When all the necessary operations with the document are completed, if necessary, save the changes in the document (by applying a impulse to the 4th input of the Excel Document component) and you need to close it by applying the value False to the first input of the Excel Document component.

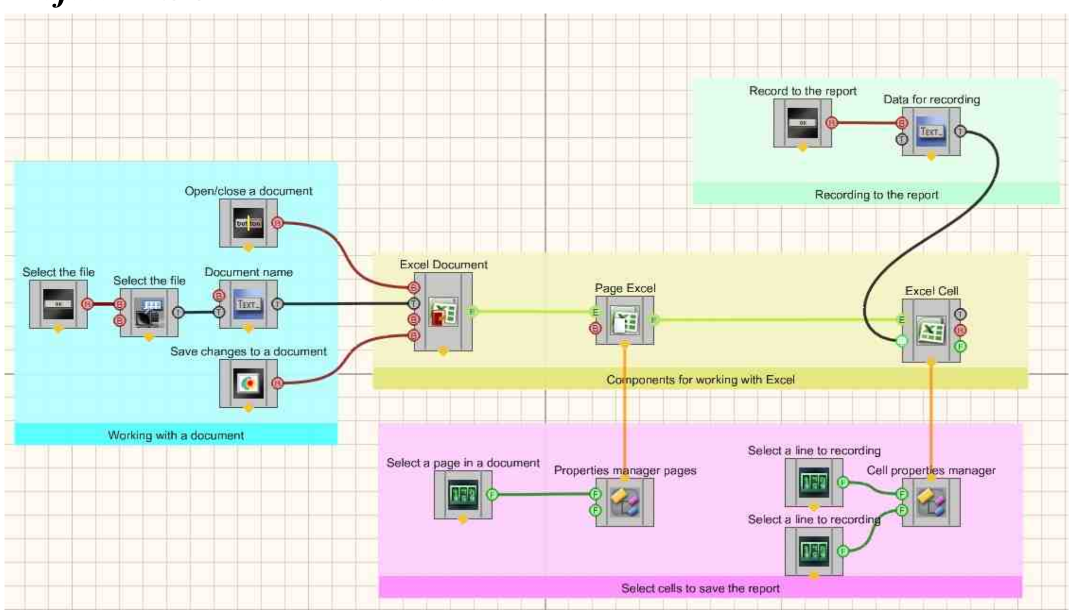
Notes

1. The file can be opened in hidden mode by changing the ExcelDocVisible property to False in the properties of the Excel Document component.

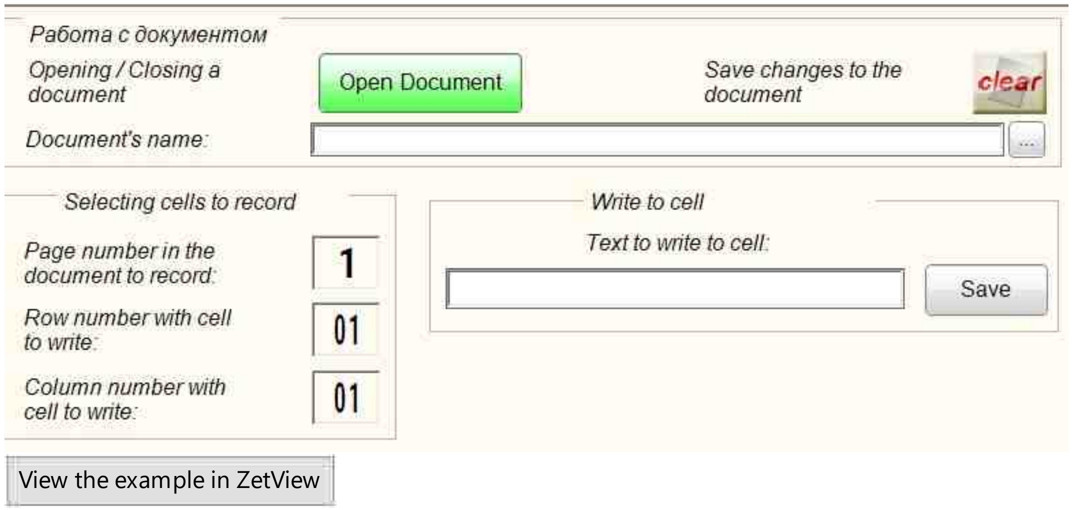
2. The report file can be generated in advance (template) by filling in the required fields and adding the necessary macros / formatting to the document itself, and then enter the data into the cells. To do this, before opening the document, set the document name as the template name and before saving changes in it, change the document name to the one under which you want to save the report. A new file will be created with your report.

3.

Project in SCADA ZETView



Project operation result



35.2.Excel Page

The component "**Excel Page**" establishes connection to a particular sheet of Excel-file, has additional function of printing a page.

Components of "Excel" section are not depicted in operator mode and can be added to the project only in developer mode.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Book - gets a pointer to an Excel file.

- Print - sends the specified page for printing. The default printer in the system is used.

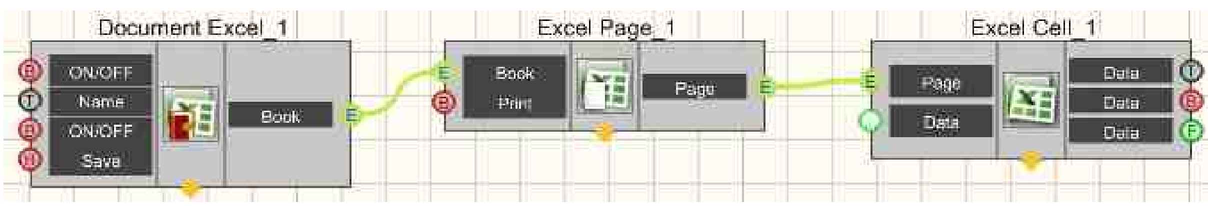
➢ Output:
- Page - Passe a pointer to an Excel page.

Custom common properties:
➢Total properties (environment):
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**
- SheetNumber (1) - page number.
- Print _ Copies (0) - number of copies to print.

Description:

To record data to an Excel file, it is used in conjunction with the Excel Page 1283 and " Excel Cell 1289 components. You need to connect the components in the following sequence:



This component is more of a building block than a stand-alone component. Required when working with excel files. Although the Excel Document 1283, Excel Page, 1283 and Excel Cell 1289 components have the same type of input and output pins, the data coming from these pins is different. The pointer to the book comes from the Excel Document 1283 component, and the pointer to the page comes from the Excel Page 1283 component. Therefore, you cannot connect Excel Document 1283 and Excel Cell 1289 directly.

The component works stably with the Microsoft version office Excel 2007.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the

component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**

- Book - gets a pointer to an Excel file.

➢**Custom common properties:**

- Are available by the link below. 154

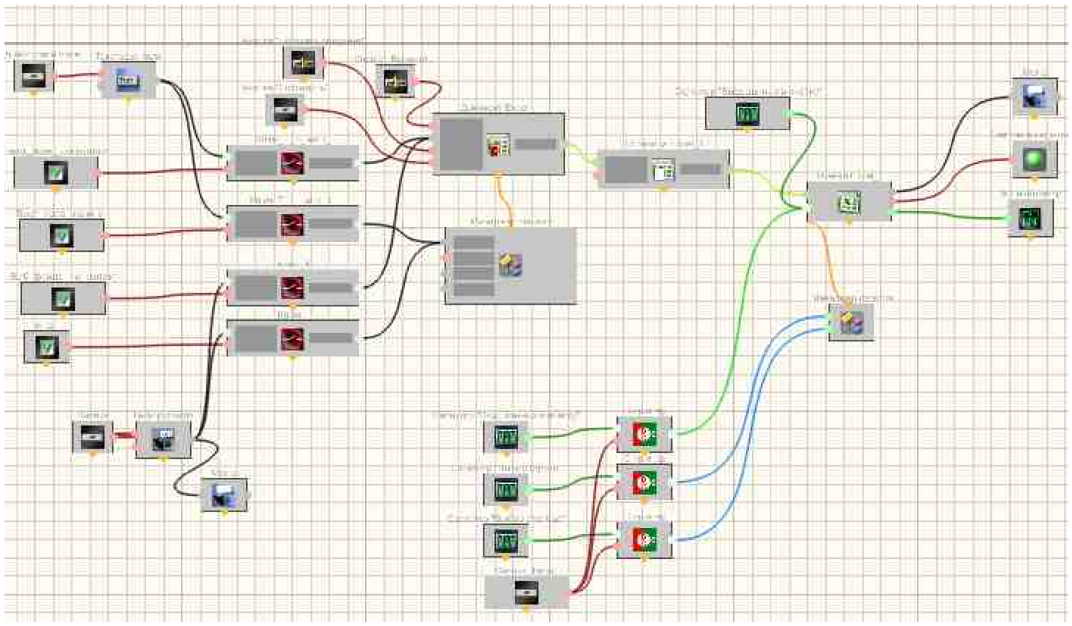
➢**Custom private properties (the default value is shown in parentheses):**

- SheetNumber (1) - page number.
- Print _ Copies (0) - number of copies to print.

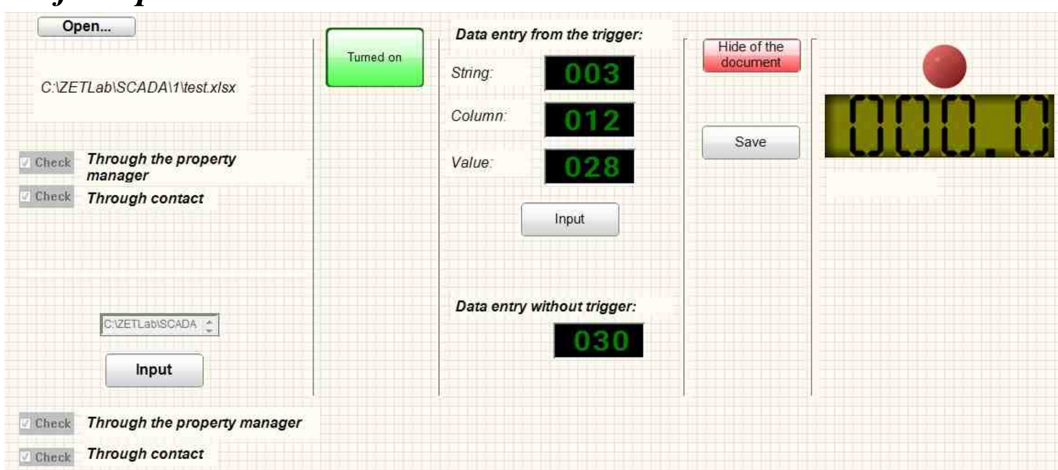
Note: ZETExcelDocument .osx if the file specified in its properties is not found, it will try to find it in the project's exe directory. if it is not there, it will display a message that the file was not found in such and such a place, and if you click "OK", a filedialog will be called with the choice of a file to open. First you need to create and put in the selected folder a file with an extension with the name and extension *. xlsx .

Example No. 1

Project in SCADA ZETView



Project operation result



View the example in ZetView

Example No. 2

To generate a report in an Excel file using ZETView, 3 components from the Excel tab are used :

1. Excel document;
2. Excel page;

3. Excel cell.

For more detailed information on the operation of these components, select the component you are interested in and press F 1.

Report save algorithm

1. First you need to specify the path to the file for the report. It can be the full path to the file or just its name. If you specify an incomplete path, the program will look for the file in the directory where the running project is located. If the file does not exist in the specified location, it will be created automatically.

2. Getting started with a document, first you need to open it by setting the value “True ” to the first input of the Excel Document component .

3. The cell where the user wants to record data must then be selected by selecting a page in the document using the SheetNumber private property of the Excel Page component. Then, using the private properties of the “ Excel Cell ” component “ Line ” and “ Col ”, select the row and column with the cell in which the information will be written.

4. To record data to a cell, it is necessary to submit the text to the second input of the Excel Cell component, which will be written to it. After that, if necessary, you can save the changes in the document by applying an impulse to the 4th input of the Excel Document component or do it later when closing the file.

5. When all the necessary operations with the document are completed, if necessary, save the changes in the document (by applying a impulse to the 4th input of the Excel Document component) and you need to close it by applying the value False to the first input of the Excel Document component.

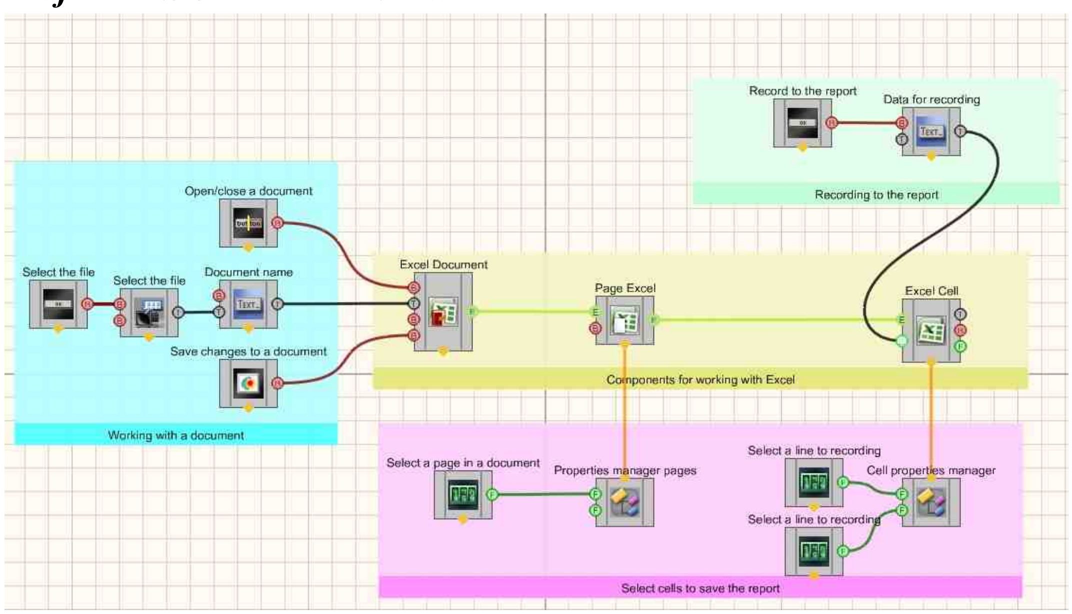
Notes

1. The file can be opened in hidden mode by changing the ExcelDocVisible property to False in the properties of the Excel Document component.

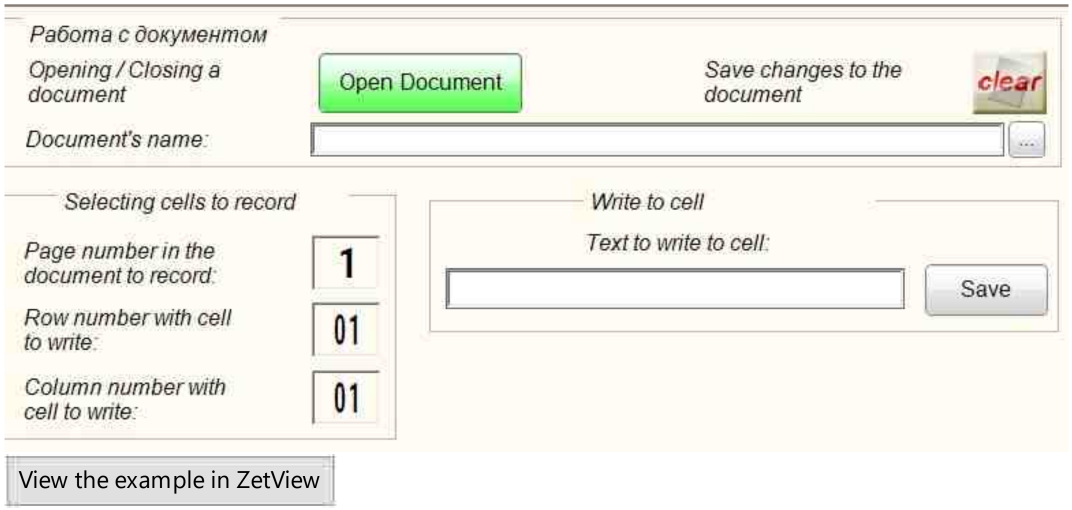
2. The report file can be generated in advance (template) by filling in the required fields and adding the necessary macros / formatting to the document itself, and then enter the data into the cells. To do this, before opening the document, set the document name as the template name and before saving changes in it, change the document name to the one under which you want to save the report. A new file will be created with your report.

3.

Project in SCADA ZETView



Project operation result

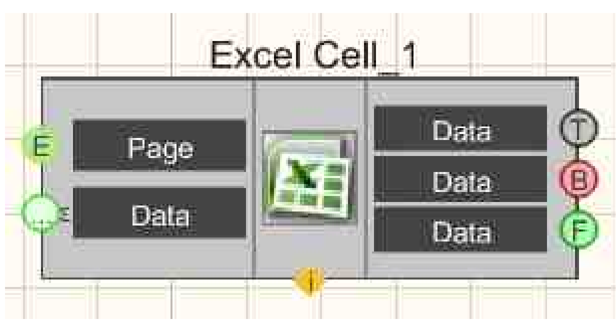


35.3.Excel Cell

The component "**Excel Cell**" is used for data recording to a particular cell.

Components of "Excel" section are not depicted in operator mode and can be added to the project only in developer mode.

Appearance of the component:

Developer interface	Operator interface
	Doesn't have

Setting:

➢Input:

- Page - a pointer to the Excel page, in the cells of which values will be written.
- Data - data of any type that will be written to the specified cell.

➢ Output:

- Data - the data in the cell (text) is displayed.
- Data - displays the data in the cell (bool).
- Data - the data in the cell (float) is displayed.

Custom common properties:

➢Total properties (environment):

- Are available by the link below.[154]

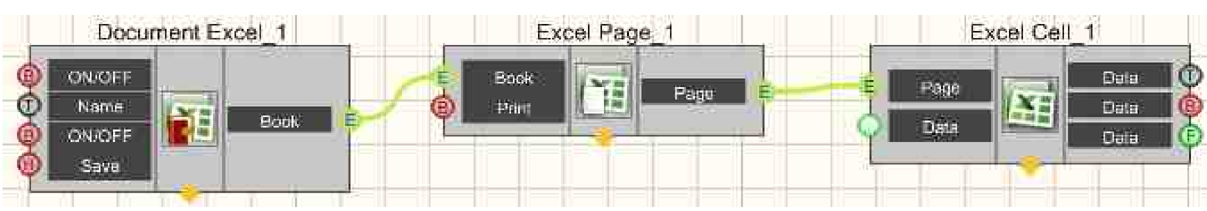
➢Custom private properties (the default value is shown in parentheses):

- Line (1) - line number to record to.
- Col (1) - the number of the column to record to.

Description:

This component has some feature of work. If the component receives 3 parameters at the same time: the row number, the column number and the value to be written, then regardless of the number of changes in the row number or column number, the component will not go to the desired cell until it receives the value for the "Data" input. That is, if you record the value "3" in a cell (1,2), and then without changing the value change the cell number, to Example (2,3), nothing will change. But if, let's say, the value 4 is given to the "Data" input, then the component will record to the specified excel file in the cell (2,3), or in the one that is specified, the value 4.

The Excel Cell[1289] components is specific and is a building block, not a stand-alone component. To record data to an Excel file, you must use this component in conjunction with the Excel Page[1283] and Excel Document[1277] component. You need to connect the components in the following sequence:



Although the Excel Document[1277], Excel Page, [1283]and Excel Cell[1289] components have the same type of input and output pins, the data coming from these pins is different. The pointer to the book comes from the Excel Document[1277] component, and the pointer to the page comes from the Excel Page[1283] component. Therefore, you cannot connect Excel Document[1277] and Excel Cell [1289]directly.

Note : ZETExcelDocument .osx if the file specified in its properties is not found, it will try to find it in the project's exe directory. if it is not there, it will display a message that the file was not found in such and such a place, and if you click "OK", a filedialog will be called with the choice of a file to open. First you need to create and put in the selected folder a file with an extension with the name and extension *. xlsx.

Programming

When using the present component in a script it is necessary to take into consideration the range of the values that are sent to the input contacts of the component, component properties values range as well as the range of component's methods parameters values.

Setting:

➢**Input:**
- Page - a pointer to the Excel page, in the cells of which values will be written.
- Data - data of any type that will be written to the specified cell.

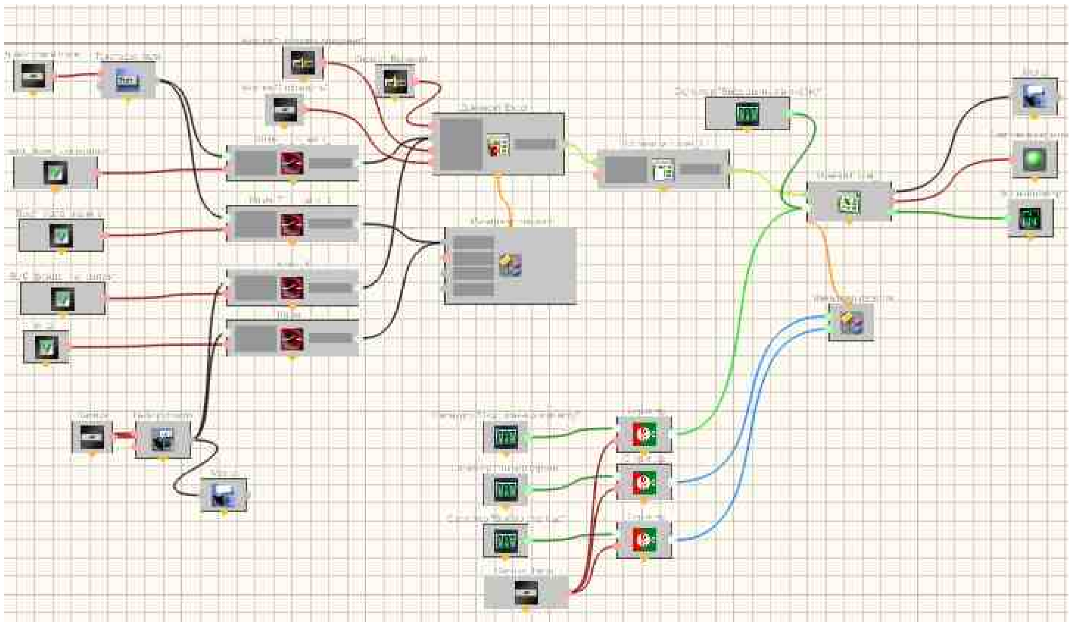
➢**Custom common properties:**
- Are available by the link below. 154

➢**Custom private properties (the default value is shown in parentheses):**

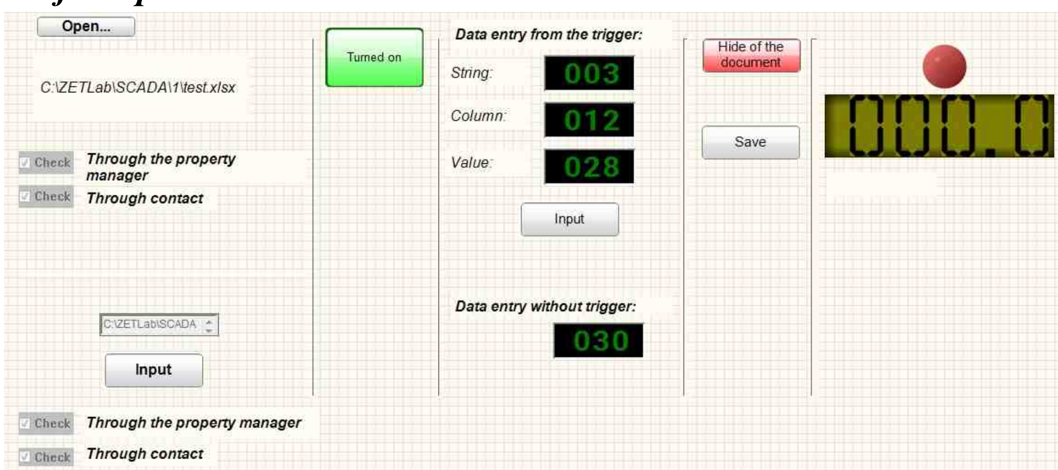
- Line (1) - line number to record to.
- Col (1) - the number of the column to record to.

Example No. 1

Project in SCADA ZETView



Project operation result



View the example in ZetView

Example No. 2

To generate a report in an Excel file using ZETView, 3 components from the Excel tab are used :

1. Excel document;
2. Excel page;

3. Excel cell.

For more detailed information on the operation of these components, select the component you are interested in and press F 1.

Report save algorithm

1. First you need to specify the path to the file for the report. It can be the full path to the file or just its name. If you specify an incomplete path, the program will look for the file in the directory where the running project is located. If the file does not exist in the specified location, it will be created automatically.

2. Getting started with a document, first you need to open it by setting the value “True ” to the first input of the Excel Document component .

3. The cell where the user wants to record data must then be selected by selecting a page in the document using the SheetNumber private property of the Excel Page component. Then, using the private properties of the “ Excel Cell ” component “ Line ” and “ Col ”, select the row and column with the cell in which the information will be written.

4. To record data to a cell, it is necessary to submit the text to the second input of the Excel Cell component, which will be written to it. After that, if necessary, you can save the changes in the document by applying an impulse to the 4th input of the Excel Document component or do it later when closing the file.

5. When all the necessary operations with the document are completed, if necessary, save the changes in the document (by applying a impulse to the 4th input of the Excel Document component) and you need to close it by applying the value False to the first input of the Excel Document component.

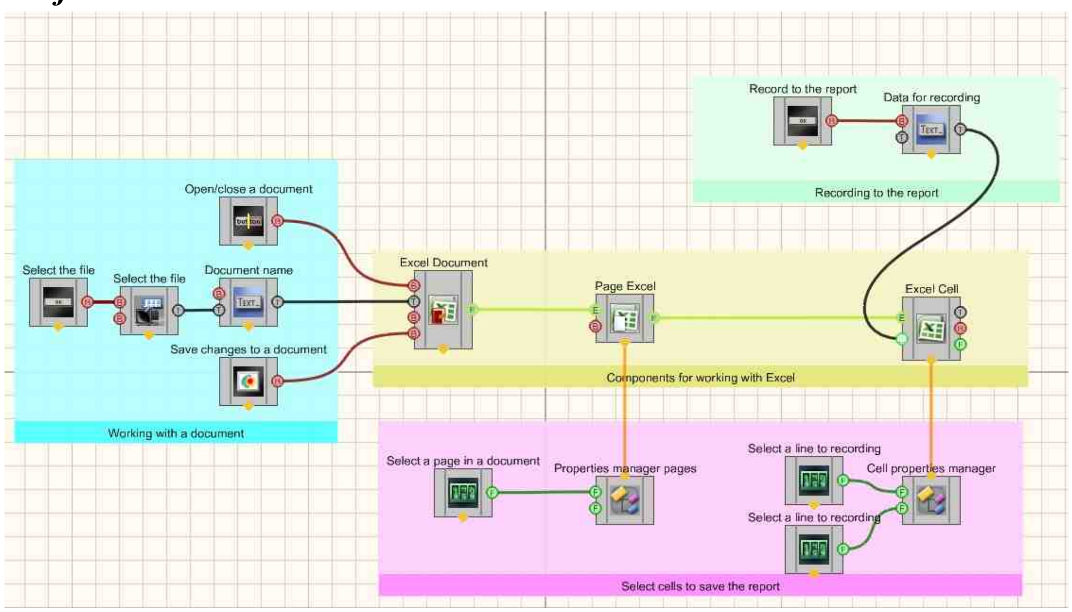
Notes

1. The file can be opened in hidden mode by changing the ExcelDocVisible property to False in the properties of the Excel Document component.

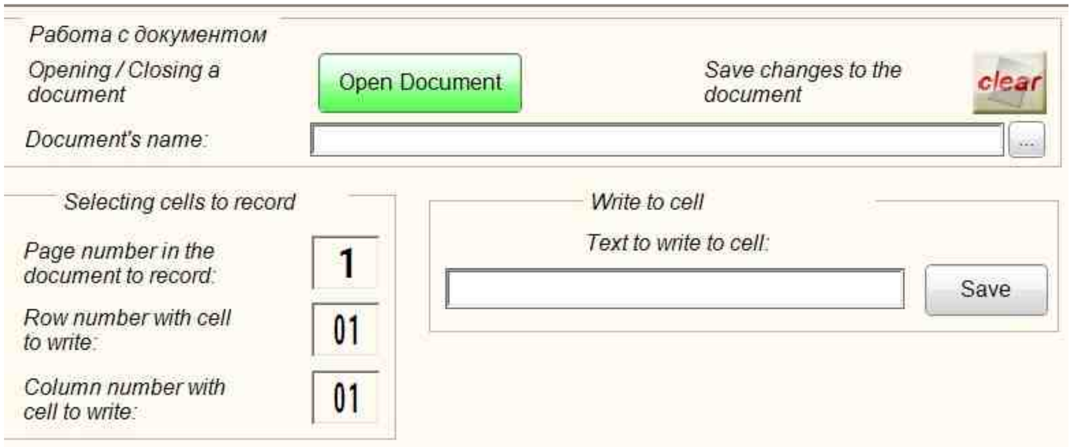
2. The report file can be generated in advance (template) by filling in the required fields and adding the necessary macros / formatting to the document itself, and then enter the data into the cells. To do this, before opening the document, set the document name as the template name and before saving changes in it, change the document name to the one under which you want to save the report. A new file will be created with your report.

3.

Project in SCADA ZETView



Project operation result



View the example in ZetView

Part 5. ZETView projects available in ZETLab

Глава 1.Metrology

1.1.AFR measurement at logarithmic sweep by frequency (AC)

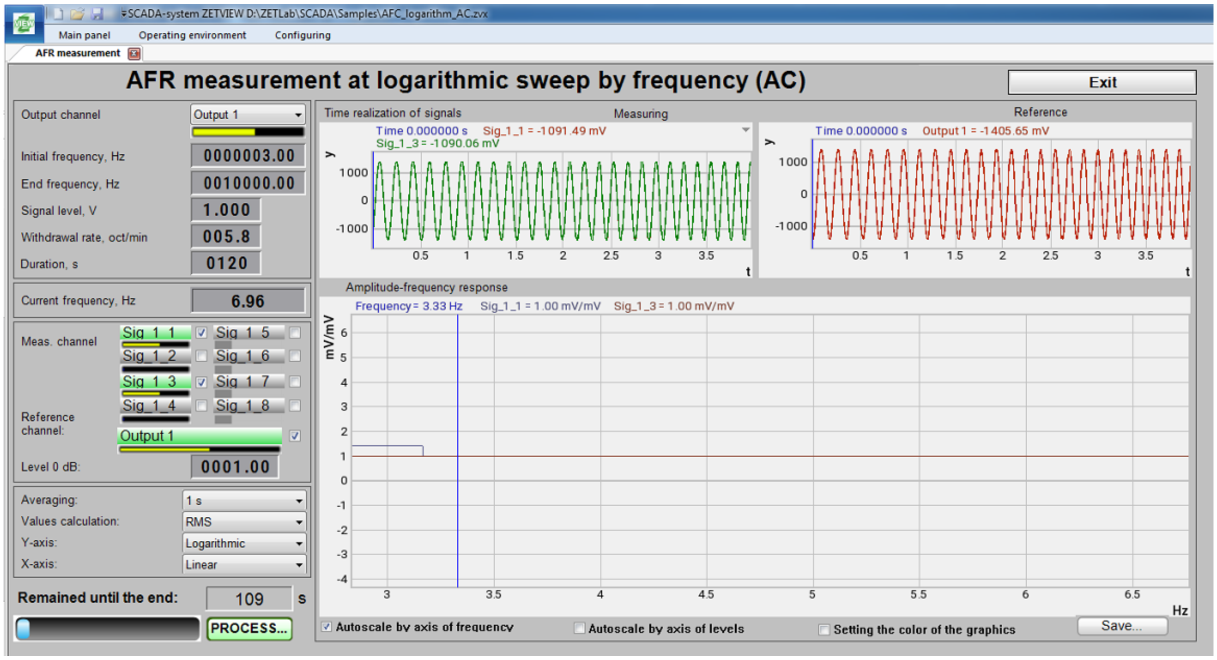
The program "**AFR measurement at logarithmic sweep by frequency (AC)"** is designed to measure the amplitude-frequency characteristic and can be used for:

:

- **AFR measurement** of primary transducers by the electric method or on a shaker,
- **AFR measurement** of monomorphic piezoelectric elements,
- **AFR measurement** of shaker,
- etc.

When **AFR measurement**, a signal with the specified parameters is formed at the output of the generator. On the input channels of the spectrum analyzer, the response of the tested path is recorded. Measurement results and recorded signals are displayed in the program in real time.

The measurement results are saved in a file format *.dtu, which is opened in the **Results viewing** program in graphical and numerical form.



AFR measurement at logarithmic sweep by frequency (AC)

The program **AFR measurement at logarithmic sweep by frequency (AC)** is implemented in the **ZETVIEW** SCADA system **.** When you start the program **AFR measurement at logarithmic sweep by frequency (AC)** in the hidden mode, the programs **Signals generator** (one copy) and **AC Voltmeter** (from 1 to 9 copies - according to the number of measurement channels) are launched. To remove the frequency response, a frequency-modulated signal with a logarithmic (**LogFM**) frequency sweep is used.

Taking measurements

To carry out measurements it is necessary:

- select oscillator channel - if several spectrum analyzers are connected to the computer, it is necessary to select a master oscillator;
- set the start and end frequency of measurements - it should be remembered that measurements in the set frequency range will be carried out with the same data averaging value, so measurements, in Example , in the range from 0.1 to 1000 Hz are best done in 2 stages: in the range from 0.1 to 20 Hz with an average of 10 seconds and in the range from 10 to 1000 Hz with an average of 0.1 s;
- set the generator signal level - in the process of measurements, a signal with a given voltage (RMS) with a serrated changing frequency from the initial value to the final value will be generated at the generator output;
- set frequency response rate - determines the frequency change rate of the generator signal;
- the duration of the measurements will be calculated automatically according to the values of the initial and final frequencies and the frequency response rate. When changing the duration value, the frequency response rate value will be automatically recalculated.

Note : when changing the start or end frequency, the duration of the measurements changes, when the duration changes, the frequency response rate changes, when the speed changes, the duration of the measurements changes.
Also, before starting the measurements, you should select the measuring channel - one or more - in the program **AFR measurement at logarithmic sweep by frequency (AC).** Measurements can be carried out on 8 channels simultaneously.
To perform relative measurements, you should also select the reference channel and set the flag in the "Reference channel" line.
Note:

- when carrying out *absolute measurements,* the measurement result is the signal voltage along the measuring channel
- when carrying out *relative measurements* , the measurement result is the ratio of the signal voltage along the measuring channel to the signal voltage along the reference channel.

Level 0 dB Set the zero level when displaying results in dB. in Example , when carrying out relative measurements with the *0 dB level parameter* equal to 1, the measurement result $U_{meas} / U_{op} = 1$ will be recorded on the graphic at point 0.
AFR measurement at logarithmic sweep by frequency (AC) program allows you to take measurements with an average of 0.1, 1 or 10 seconds, calculating the RMS, peak or amplitude value of the signal, presenting the result in a linear or logarithmic scale along the Y axis .
In the process of measurements, the time realizations of signals on the measuring and reference channels are displayed (on the oscillograms from above), the current frequency and the time remaining until the end of the measurements (on the indicators on the left) and a graphic of the frequency response is plotted.
Measurement results can be saved to a file for viewing in display programs (in graphical and numerical form) and printing.

Hardware support

The program for **AFR measurement at linear sweep by frequency (AC)** is supplied with spectrum analyzers (for Example, **ZET 017-U2, ZET 017-U4, ZET 017-U8, A19) in the presence of SCADA ZETVIEW.**

Supported hardware

The program "**AFR measurement at logarithmic sweep by frequency (AC)**" is part of the following software:
ZETLAB_ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB_VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times. In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example, using an arithmometer, or as a reference channel signal in the Strain-gauge

meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet, the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.1.1) **ZETLab control panels** press the button **AFR measurement at logarithmic sweep by frequency (AC)**.

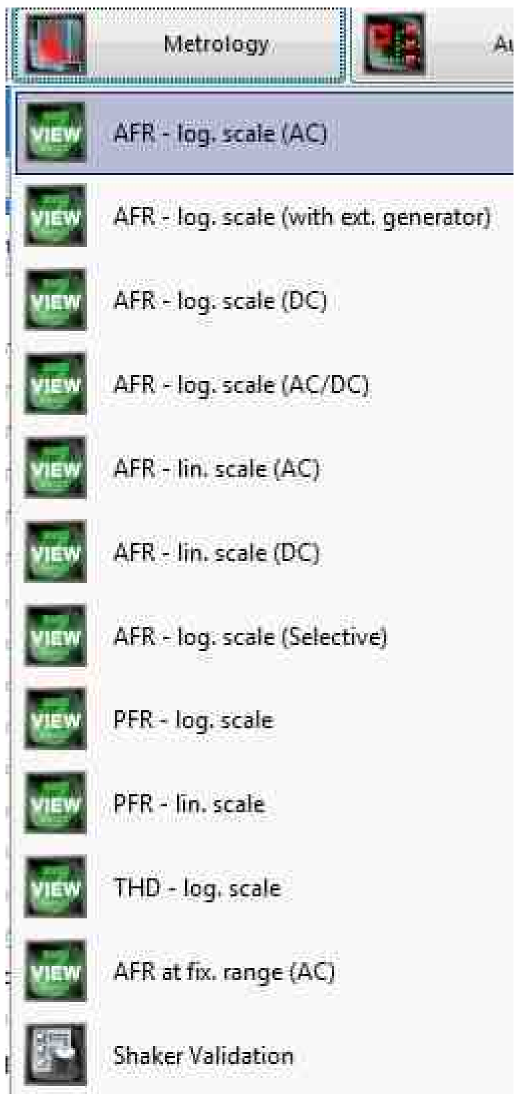


Fig. 5.1.1. List of menu programs "Metrology"

In the opened window of the program **AFR measurement at logarithmic sweep by frequency (AC)** (Fig. 5.1.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to such a value that acceleration and displacement do not exceed the maximum allowable values.);
- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

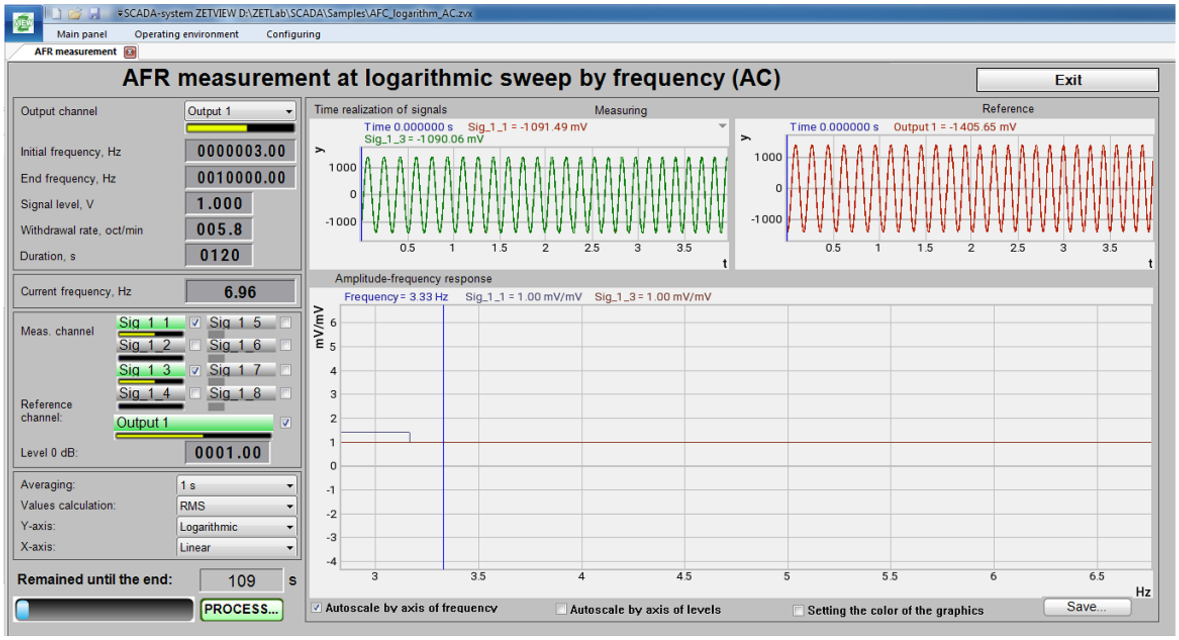
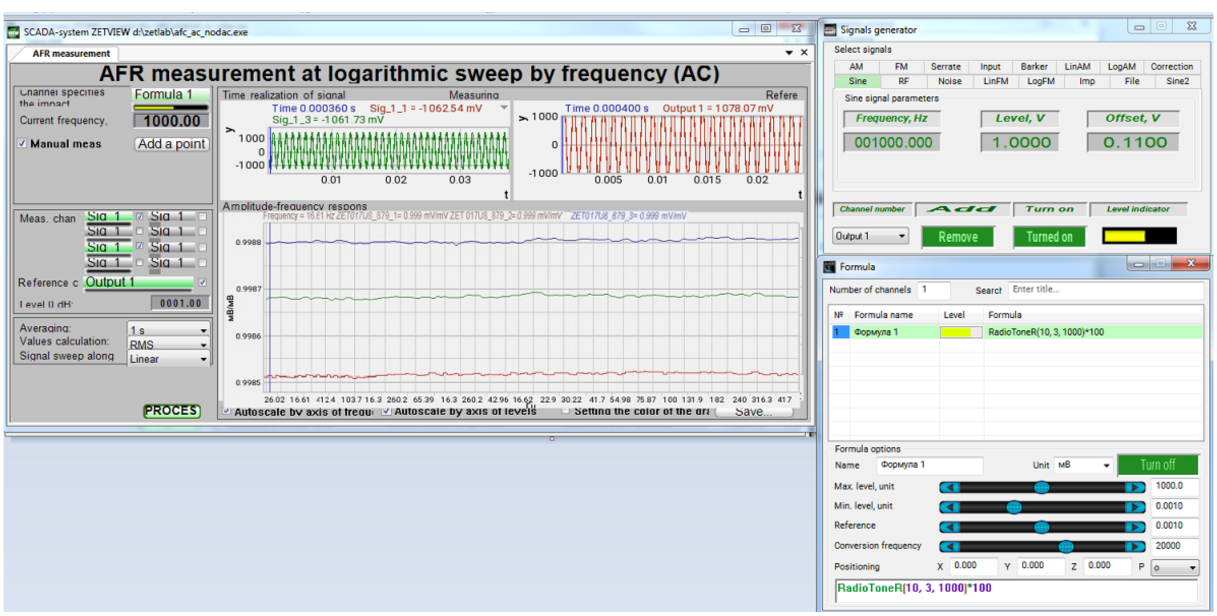


Fig. 5.1.2. Window of the program "AFR measurement at logarithmic sweep by frequency (AC)". The project is in the process of reading the frequency response with a logarithmic sweep.

1.2.AFR - log. scale (with ext. generator)

The program " **AFR - log. scale (with ext. generator)** is designed to measure the frequency response using an external generator as a master device, and not a generator that runs in the Zetlab software . For example, this is very important when taking the frequency response of a shaker during its certification, where the shaker is controlled by an external generator, and certification is carried out on equipment manufactured by ETMS LLC.



AFR - log. scale (with ext. generator)

The program **AFR - log. scale (with ext. generator)** is implemented in the **ZETVIEW** SCADA system **.** When you start the program **AFR - log. scale (with ext. generator)** in the hidden mode, the programs **Signals generator** (one copy) and **AC Voltmeter** (from 1 to 9 copies - according to the number of measurement channels) are launched. To remove the frequency response, a frequency-modulated signal with a logarithmic (**LogFM**) frequency sweep is used.

Taking measurements

To carry out measurements it is necessary:

- select oscillator channel - if several spectrum analyzers are connected to the computer, it is necessary to select a master oscillator;
- set the start and end frequency of measurements - it should be remembered that measurements in the set frequency range will be carried out with the same data averaging value, so measurements, in Example , in the range from 0.1 to 1000 Hz are best done in 2 stages: in the range from 0.1 to 20 Hz with an average of 10 seconds and in the range from 10 to 1000 Hz with an average of 0.1 s;
- set the generator signal level - in the process of measurements, a signal with a given voltage (RMS) with a serrated changing frequency from the initial value to the final value will be generated at the generator output;
- set frequency response rate - determines the frequency change rate of the generator signal;
- the duration of the measurements will be calculated automatically according to the values of the initial and final frequencies and the frequency response rate. When changing the duration value, the frequency response rate value will be automatically recalculated.

Note : when changing the start or end frequency, the duration of the measurements changes, when the duration changes, the frequency response rate changes, when the speed changes, the duration of the measurements changes.

Also, before starting the measurements, you should select the measuring channel - one or more - in the program **AFR - log. scale (with ext. generator).** Measurements can be carried out on 8 channels simultaneously.

To perform relative measurements, you should also select the reference channel and set the flag in the "Reference channel" line.

Note:

- when carrying out *absolute measurements,* the measurement result is the signal voltage along the measuring channel
- when carrying out *relative measurements* , the measurement result is the ratio of the signal voltage along the measuring channel to the signal voltage along the reference channel.

Level 0 dB Set the zero level when displaying results in dB. in Example , when carrying out relative measurements with the *0 dB level parameter* equal to 1, the measurement result $U_{meas} / U_{op} = 1$ will be recorded on the graphic at point 0.

AFR - log. scale (with ext. generator) program allows you to take measurements with an average of 0.1, 1 or 10 seconds, calculating the RMS, peak or amplitude value of the signal, presenting the result in a linear or logarithmic scale along the Y axis .

In the process of measurements, the time realizations of signals on the measuring and reference channels are displayed (on the oscillograms from above), the current frequency and the time remaining until the end of the measurements (on the indicators on the left) and a graphic of the frequency response is plotted.
Measurement results can be saved to a file for viewing in display programs (in graphical and numerical form) and printing.

Hardware support

The program for **AFR - log. scale (with ext. generator)** is supplied with spectrum analyzers (for Example, **ZET 017-U2, ZET 017-U4, ZET 017-U8, A19) in the presence of SCADA ZETVIEW.**

Supported hardware

The program " **AFR - log. scale (with ext. generator)**" is includes the following software:
ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times . In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.2.1) **ZETLab control panels** press the button **- AFR - log. scale (with external generator)**.

Fig. 5.2.1. List of menu programs "Metrology"

In the opened window of the program **AFR - log. scale (with ext. generator)** (Fig. 5.2.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to

such a value that acceleration and displacement do not exceed the maximum allowable values.);

- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

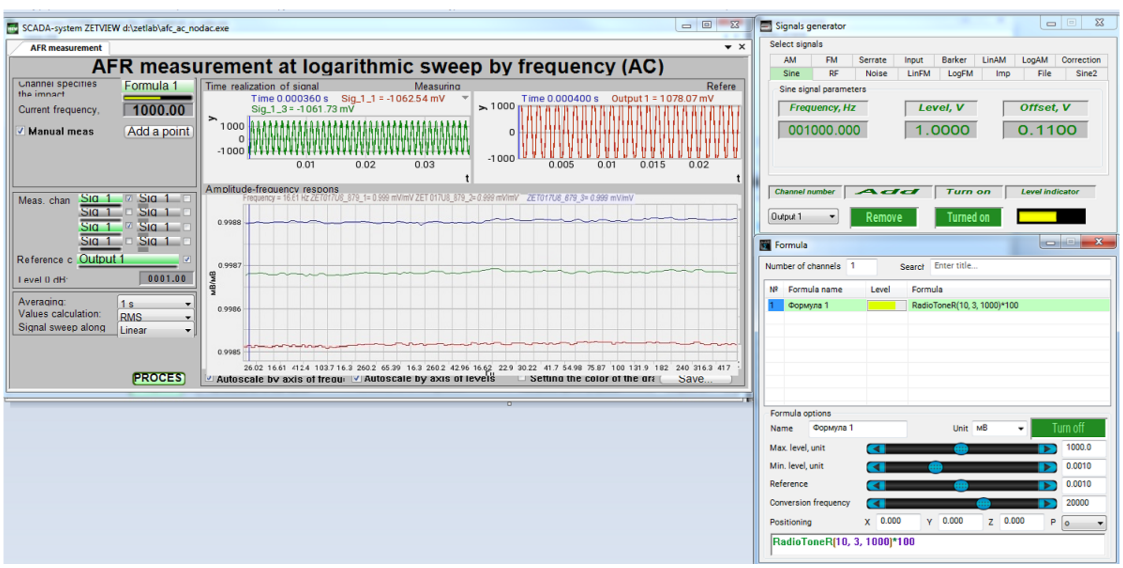
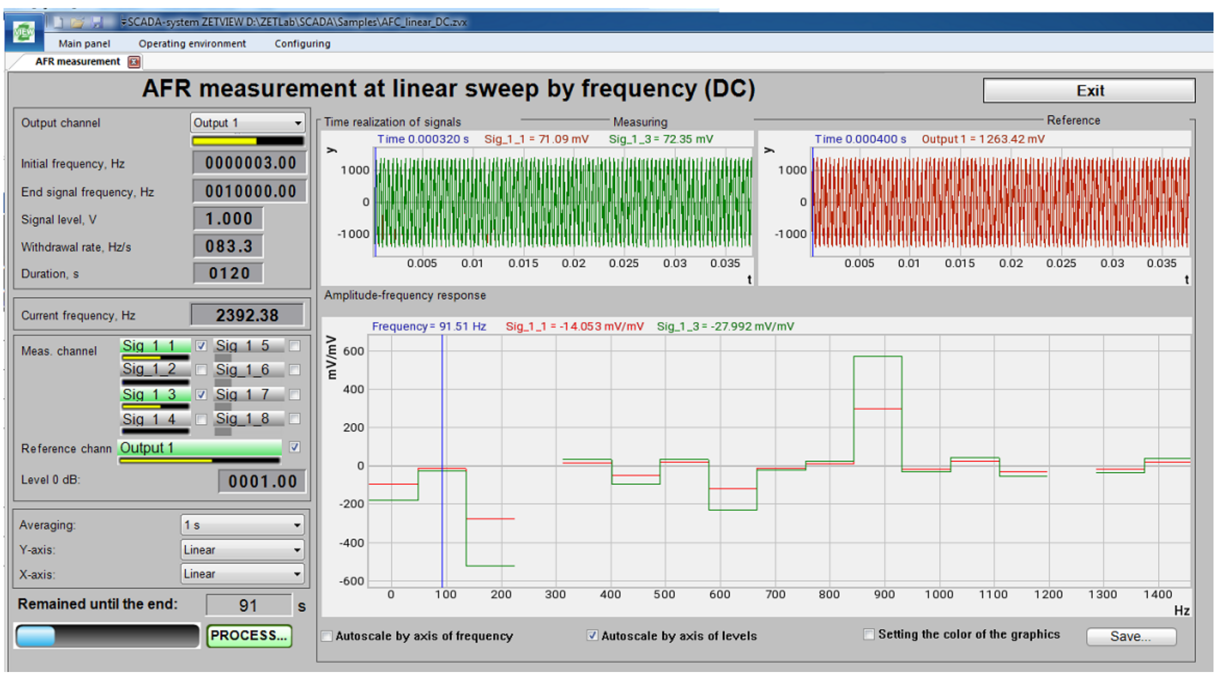


Fig. 5.2.2. Window of the program "AFR - log. scale (with ext. generator)". The project is in the process of reading the frequency response with a logarithmic sweep.

1.3.AFR measurement at logarithmic sweep by frequency (DC)

The program " **AFR measurement at logarithmic sweep by frequency (DC)** " is designed to determine the amplitude-frequency characteristic when measuring the amplitude of the constant component of the signal. Measurements can be carried out on 8 channels simultaneously.
For measurements, certified equipment is used - spectrum analyzer type **ZET017** or type **A19** and SCADA **ZETVIEW.**
The spectrum analyzer's built-in generator is used to reproduce the test signal (a Sine signal with a serrated varying frequency). This signal is applied to the input of the device under test or to the input of the test equipment (depending on the measurement scheme). The signals under test are connected to the input channels of the spectrum analyzer. Signal parameters are measured with a DC voltmeter. The graphic plots the amplitude-frequency characteristic in absolute or relative units.

Measurement results are saved in a file format . *dtu* , which is opened in the **Results viewing** program in graphical and numerical form



AFR measurement at logarithmic sweep by frequency (DC).

The program **AFR measurement at logarithmic sweep by frequency (DC)** is implemented in the **ZETVIEW** SCADA system **.** When you start the program **AFR measurement at logarithmic sweep by frequency (DC)** in the hidden mode, the programs **Signals generator** (one copy) and **AC Voltmeter** (from 1 to 9 copies - according to the number of measurement channels) are launched. To remove the frequency response, a frequency-modulated signal with a logarithmic (**LogFM**) frequency sweep is used.

Taking measurements

To carry out measurements it is necessary:

- select oscillator channel - if several spectrum analyzers are connected to the computer, it is necessary to select a master oscillator;
- set the start and end frequency of measurements - it should be remembered that measurements in the set frequency range will be carried out with the same data averaging value, so measurements, in Example , in the range from 0.1 to 1000 Hz are best done in 2 stages: in the range from 0.1 to 20 Hz with an average of 10 seconds and in the range from 10 to 1000 Hz with an average of 0.1 s;
- set the generator signal level - in the process of measurements, a signal with a given voltage (RMS) with a serrated changing frequency from the initial value to the final value will be generated at the generator output;
- set frequency response rate - determines the frequency change rate of the generator signal;
- the duration of the measurements will be calculated automatically according to the values of the initial and final frequencies and the frequency response rate. When changing the duration value, the frequency response rate value will be automatically recalculated.

Note : when changing the start or end frequency, the duration of the measurements changes, when the duration changes, the frequency response rate changes, when the speed changes, the duration of the measurements changes.

Also, before starting the measurements, you should select the measuring channel - one or more - in the program **AFR measurement at logarithmic sweep by frequency (DC).** Measurements can be carried out on 8 channels simultaneously.

To perform relative measurements, you should also select the reference channel and set the flag in the "Reference channel" line.

Note:

- when carrying out *absolute measurements,* the measurement result is the signal voltage along the measuring channel
- when carrying out *relative measurements* , the measurement result is the ratio of the signal voltage along the measuring channel to the signal voltage along the reference channel.

Level 0 dB Set the zero level when displaying results in dB. in Example , when carrying out relative measurements with the *0 dB level parameter* equal to 1, the measurement result $U_{meas} / U_{op} =1$ will be recorded on the graphic at point 0.

AFR measurement at logarithmic sweep by frequency (DC) program allows you to take measurements with an average of 0.1, 1 or 10 seconds, calculating the RMS, peak or amplitude value of the signal, presenting the result in a linear or logarithmic scale along the Y axis .

In the process of measurements, the time realizations of signals on the measuring and reference channels are displayed (on the oscillograms from above), the current frequency and the time remaining until the end of the measurements (on the indicators on the left) and a graphic of the frequency response is plotted.

Measurement results can be saved to a file for viewing in display programs (in graphical and numerical form) and printing.

Hardware support

The program for **AFR measurement at logarithmic sweep by frequency (DC)** is supplied with spectrum analyzers (for Example, **ZET 017-U2, ZET 017-U4, ZET 017-U8, A19) in the presence of SCADA ZETVIEW.**

Supported hardware

The program "**AFR measurement at logarithmic sweep by frequency (DC)**" is a part of the following software:
ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.3.1) **ZETLab control panels** press the button **AFR measurement at logarithmic sweep by frequency (DC)**.

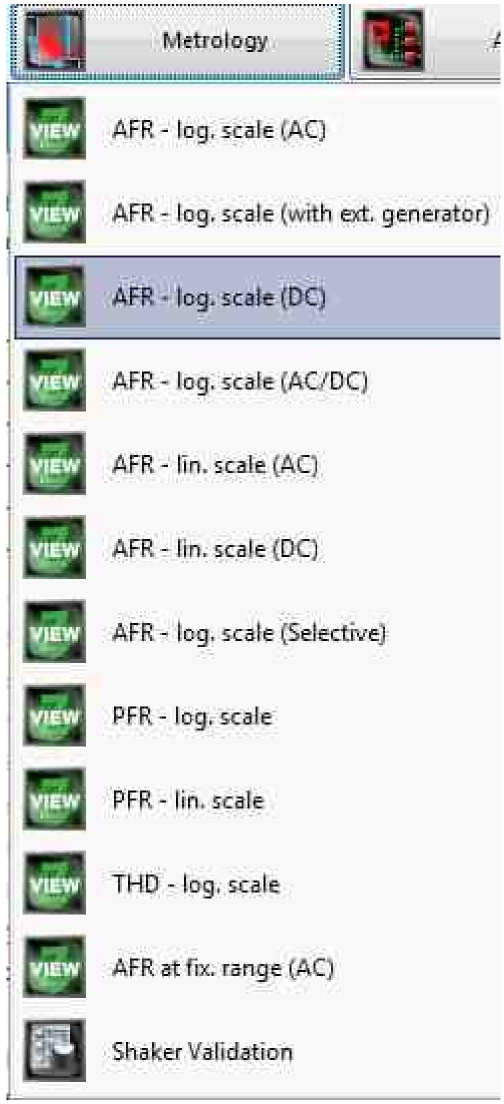


Fig. 5.3.1. List of menu programs "Metrology"

In the opened window of the program **AFR measurement at logarithmic sweep by frequency (DC)** (Fig. 5.3.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to

such a value that acceleration and displacement do not exceed the maximum allowable values.);

- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

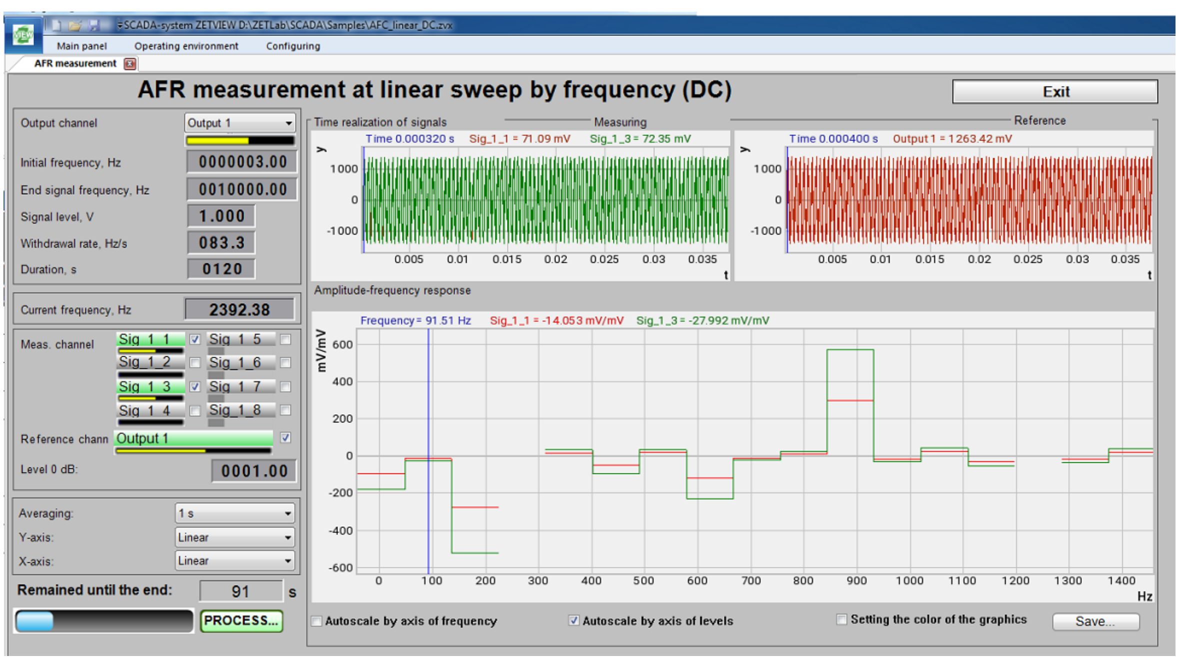
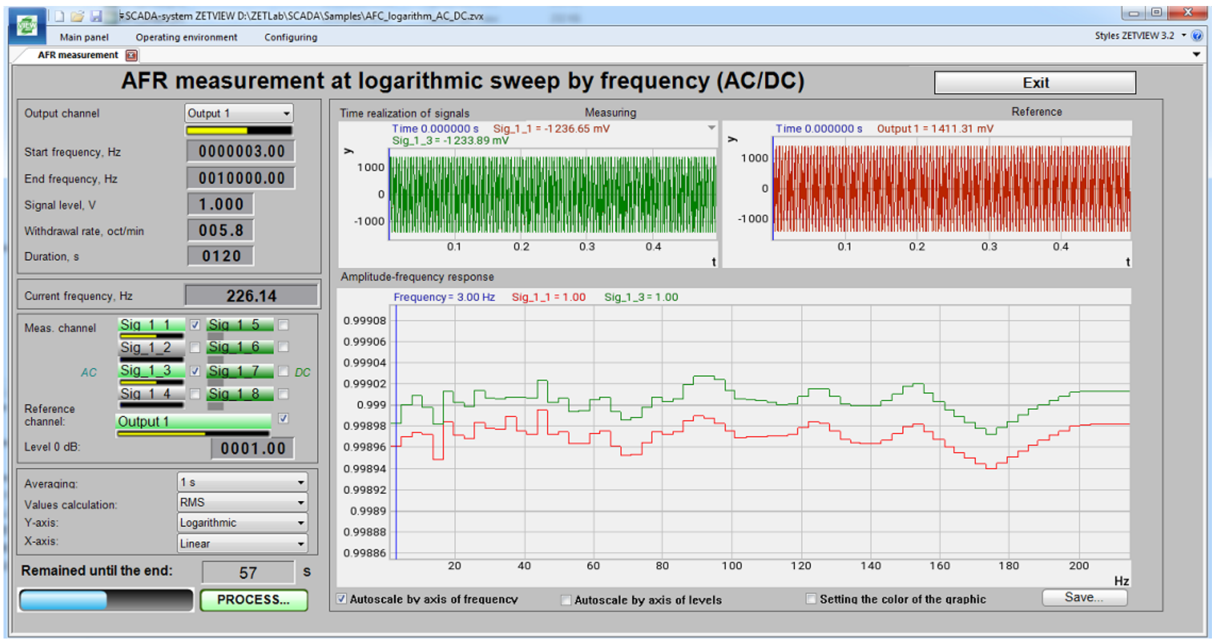


Fig. 5.3.2. Window of the program "**AFR measurement at logarithmic sweep by frequency (DC)**". The project is in the process of reading the frequency response with a logarithmic sweep.

1.4.AFR measurement at logarithmic sweep by frequency (AC/DC)

The program "**AFR measurement at logarithmic sweep by frequency (AC/DC)**" is designed to determine the amplitude-frequency response by frequency for 4 channels with a DC voltmeter and for 4 channels with an AC voltmeter.

Measurement results are saved in a file format *.dtu* , which is opened in the **Results viewing** program in graphical and numerical form



AFR measurement at logarithmic sweep by frequency (AC/DC)

The program **AFR measurement at logarithmic sweep by frequency (AC/DC)** is implemented in the **ZETVIEW** SCADA system **.** When you start the program **AFR measurement at logarithmic sweep by frequency (AC/DC)** in the hidden mode, the programs **Signals generator** (one copy) and **AC Voltmeter** (from 1 to 9 copies - according to the number of measurement channels) are launched. To remove the frequency response, a frequency-modulated signal with a logarithmic (**LogFM**) frequency sweep is used.

Taking measurements

To carry out measurements it is necessary:

- select oscillator channel - if several spectrum analyzers are connected to the computer, it is necessary to select a master oscillator;

- set the start and end frequency of measurements - it should be remembered that measurements in the set frequency range will be carried out with the same data averaging value, so measurements, in Example , in the range from 0.1 to 1000 Hz are best done in 2 stages: in the range from 0.1 to 20 Hz with an average of 10 seconds and in the range from 10 to 1000 Hz with an average of 0.1 s;
- set the generator signal level - in the process of measurements, a signal with a given voltage (RMS) with a serrated changing frequency from the initial value to the final value will be generated at the generator output;
- set frequency response rate - determines the frequency change rate of the generator signal;
- the duration of the measurements will be calculated automatically according to the values of the initial and final frequencies and the frequency response rate. When changing the duration value, the frequency response rate value will be automatically recalculated.

Note : when changing the start or end frequency, the duration of the measurements changes, when the duration changes, the frequency response rate changes, when the speed changes, the duration of the measurements changes.

Also, before starting the measurements, you should select the measuring channel - one or more - in the program **AFR measurement at logarithmic sweep by frequency (AC/DC).** Measurements can be carried out on 8 channels simultaneously.

To perform relative measurements, you should also select the reference channel and set the flag in the "Reference channel" line.

Note:

- when carrying out *absolute measurements,* the measurement result is the signal voltage along the measuring channel
- when carrying out *relative measurements* , the measurement result is the ratio of the signal voltage along the measuring channel to the signal voltage along the reference channel.

Level 0 dB Set the zero level when displaying results in dB. in Example , when carrying out relative measurements with the *0 dB level parameter* equal to 1, the measurement result $U_{meas} / U_{op} = 1$ will be recorded on the graphic at point 0.

AFR measurement at logarithmic sweep by frequency (AC/DC) program allows you to take measurements with an average of 0.1, 1 or 10 seconds, calculating the RMS, peak or amplitude value of the signal, presenting the result in a linear or logarithmic scale along the Y axis .

In the process of measurements, the time realizations of signals on the measuring and reference channels are displayed (on the oscillograms from above), the current frequency and the time remaining until the end of the measurements (on the indicators on the left) and a graphic of the frequency response is plotted.

Measurement results can be saved to a file for viewing in display programs (in graphical and numerical form) and printing.

Hardware support

The program for **AFR measurement at logarithmic sweep by frequency (AC/DC)** is supplied with spectrum analyzers (for Example, **ZET 017-U2, ZET 017-U4, ZET 017-U8, A19) in the presence of SCADA ZETVIEW.**

Supported hardware

The program "**AFR measurement at logarithmic sweep by frequency (AC/DC)**" is part of the following software:
ZETLAB_ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB_VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times. In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example, using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet, the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.4.1) **ZETLab control panels** press the button **AFR measurement at logarithmic sweep by frequency (AC/DC)**.

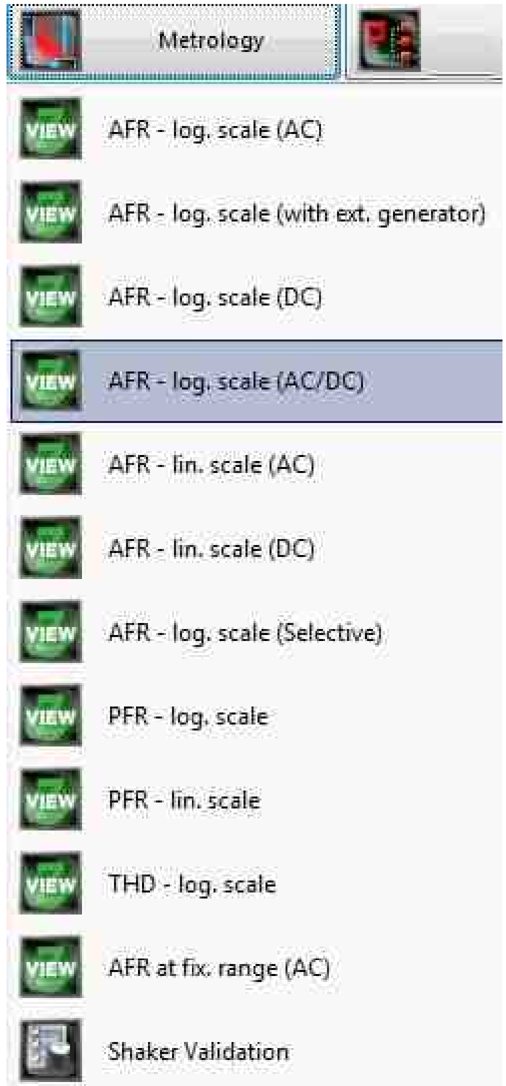


Fig. 5.4.1. List of menu programs "Metrology"

In the opened window of the program **AFR measurement at logarithmic sweep by frequency (AC/DC)** (Fig. 5.4.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to

such a value that acceleration and displacement do not exceed the maximum allowable values.);

- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

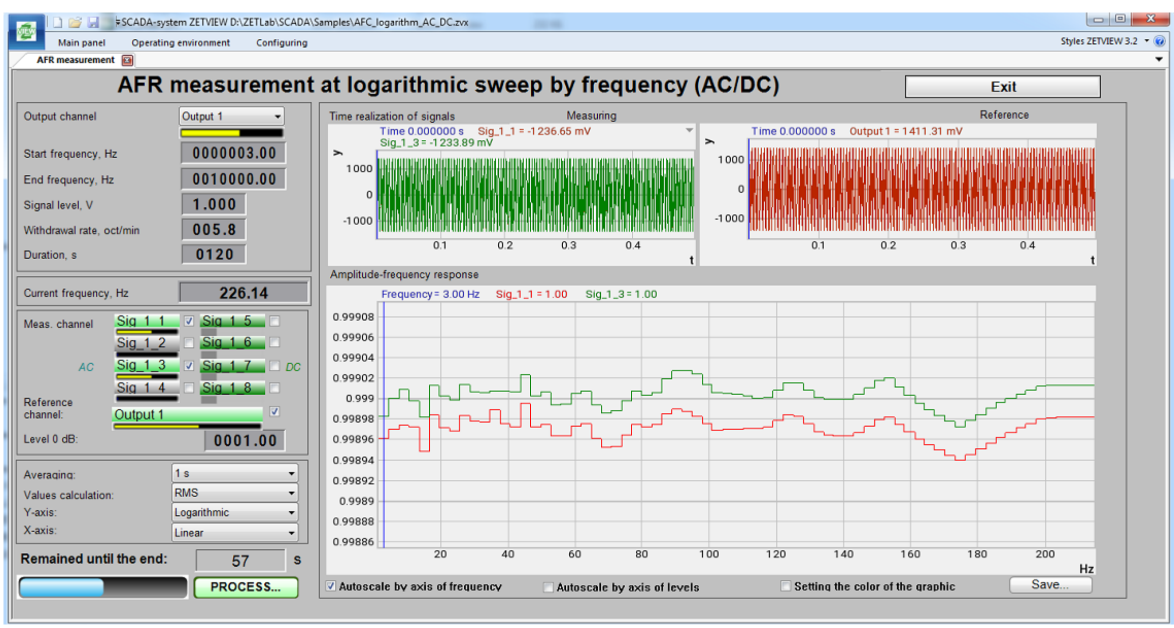


Fig. 5.4.2. Window of the program "**AFR measurement at logarithmic sweep by frequency (AC/DC)**". The project is in the process of reading the frequency response with a logarithmic sweep.

1.5.AFR measurement at linear sweep by frequency (AC)

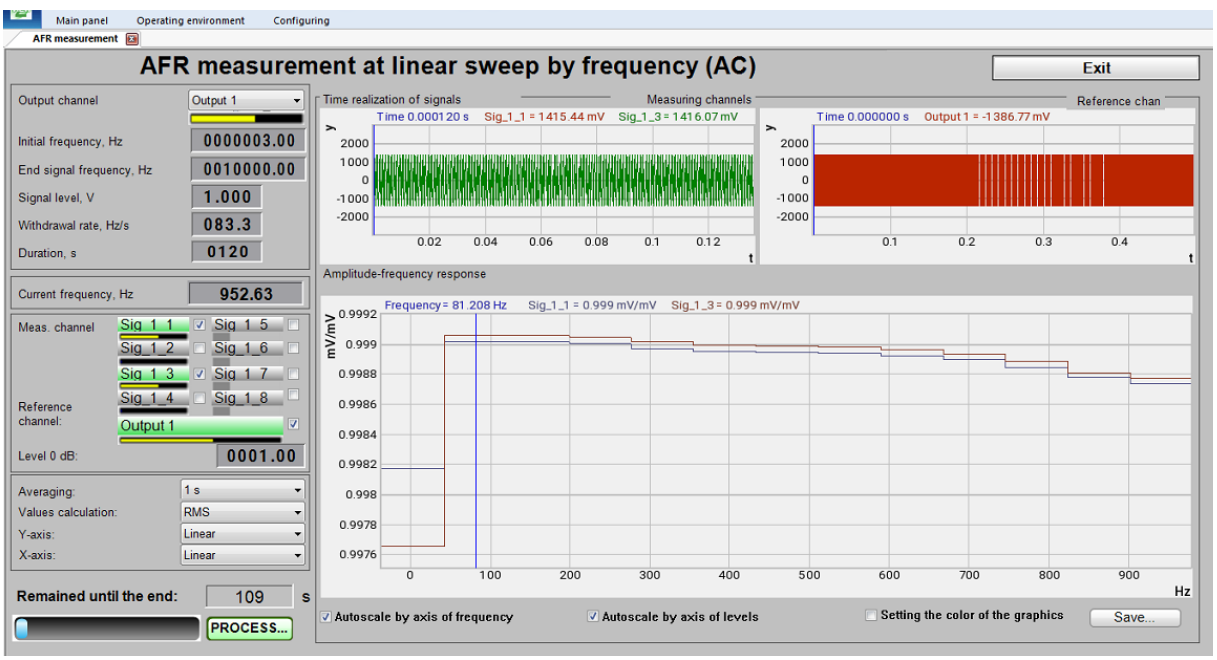
The program "**AFR measurement at linear sweep by frequency (AC)"** is designed to measure the amplitude-frequency characteristic and can be used for:

:

- **AFR measurement** of primary transducers by the electric method or on a shaker,
- **AFR measurement** of monomorphic piezoelectric elements,
- **AFR measurement** of shaker,
- etc.

When **AFR measurement**, a signal with the specified parameters is formed at the output of the generator. On the input channels of the spectrum analyzer, the response of the tested path is recorded. Measurement results and recorded signals are displayed in the program in real time.

The measurement results are saved in a file format *.dtu, which is opened in the **Results viewing** program in graphical and numerical form.



AFR measurement at linear sweep by frequency (AC)

The program **AFR measurement at linear sweep by frequency (AC)** in the hidden mode, the programs **Signals generator** (one copy) and **AC Voltmeter** (from 1 to 9 copies - according to the number of measurement channels) are launched. To remove the frequency response, a frequency-modulated signal with a linear **(LinFM)** frequency sweep is used.

Taking measurements

To carry out measurements it is necessary:

- select oscillator channel - if several spectrum analyzers are connected to the computer, it is necessary to select a master oscillator;
- set the start and end frequency of measurements - it should be remembered that measurements in the set frequency range will be carried out with the same data averaging value, so measurements, in Example , in the range from 0.1 to 1000 Hz are best done in 2 stages: in the range from 0.1 to 20 Hz with an average of 10 seconds and in the range from 10 to 1000 Hz with an average of 0.1 s;
- set the generator signal level - in the process of measurements, a signal with a given voltage (RMS) with a serrated changing frequency from the initial value to the final value will be generated at the generator output;
- set frequency response rate - determines the frequency change rate of the generator signal;
- the duration of the measurements will be calculated automatically according to the values of the initial and final frequencies and the frequency response rate. When changing the duration value, the frequency response rate value will be automatically recalculated.

Note : when changing the start or end frequency, the duration of the measurements changes, when the duration changes, the frequency response rate changes, when the speed changes, the duration of the measurements changes.

Also, before starting the measurements, you should select the measuring channel - one or more - in the program **AFR measurement at linear sweep by frequency (AC).** Measurements can be carried out on 8 channels simultaneously.

To perform relative measurements, you should also select the reference channel and set the flag in the "Reference channel" line.

Note:

- when carrying out *absolute measurements,* the measurement result is the signal voltage along the measuring channel
- when carrying out *relative measurements* , the measurement result is the ratio of the signal voltage along the measuring channel to the signal voltage along the reference channel.

Level 0 dB Set the zero level when displaying results in dB. in Example , when carrying out relative measurements with the *0 dB level parameter* equal to 1, the measurement result $U_{meas} / U_{op} =1$ will be recorded on the graphic at point 0.

AFR measurement at linear sweep by frequency (AC) program allows you to take measurements with an average of 0.1, 1 or 10 seconds, calculating the RMS, peak or amplitude value of the signal, presenting the result in a linear or logarithmic scale along the Y axis .

In the process of measurements, the time realizations of signals on the measuring and reference channels are displayed (on the oscillograms from above), the current frequency and the time remaining until the end of the measurements (on the indicators on the left) and a graphic of the frequency response is plotted.

Measurement results can be saved to a file for viewing in display programs (in graphical and numerical form) and printing.

Hardware support

The program for **AFR measurement at linear sweep by frequency (AC)** is supplied with spectrum analyzers (for Example, **ZET 017-U2, ZET 017-U4, ZET 017-U8, A19) in the presence of SCADA ZETVIEW.**

Supported hardware

The program "**AFR measurement at linear sweep by frequency (AC)**" is part of the following software:
ZETLAB_ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB_VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times. In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example, using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet, the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface and an example of how the program works

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.5.1) **ZETLab control panels** press the button **AFR measurement at linear sweep by frequency (DC)**.

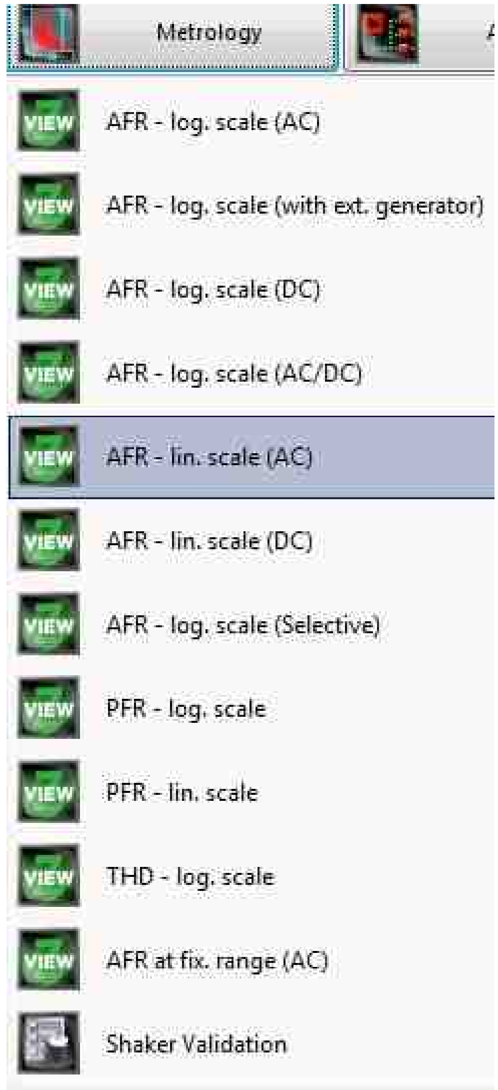


Fig. 5.5.1. List of menu programs "Metrology"

In the opened window of the program **AFR measurement at linear sweep by frequency (DC)** (Fig. 5.5.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to

such a value that acceleration and displacement do not exceed the maximum allowable values.);

- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

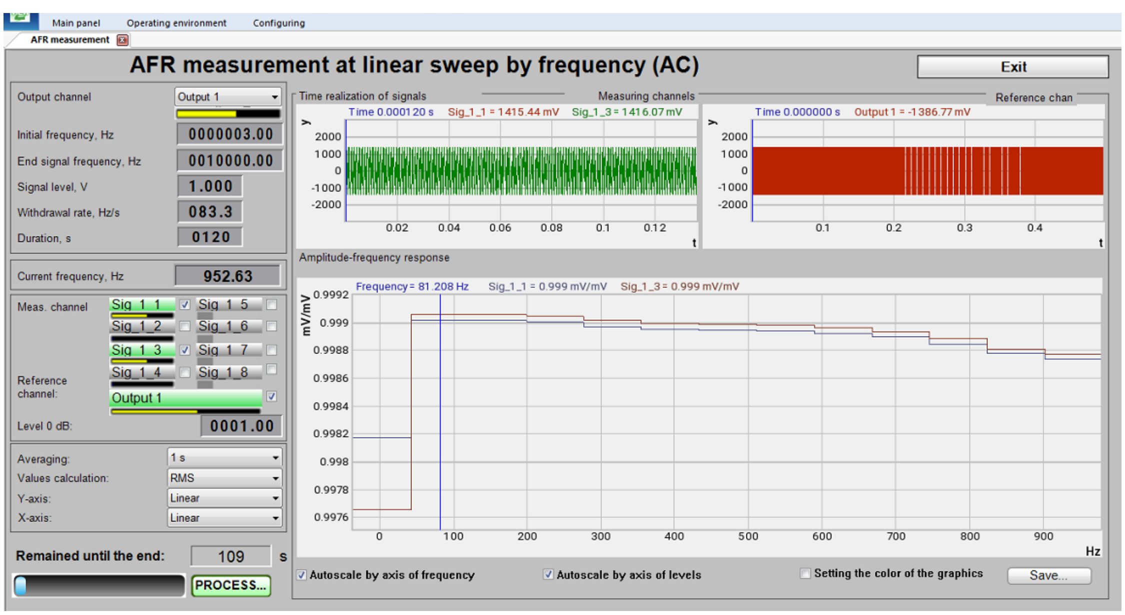


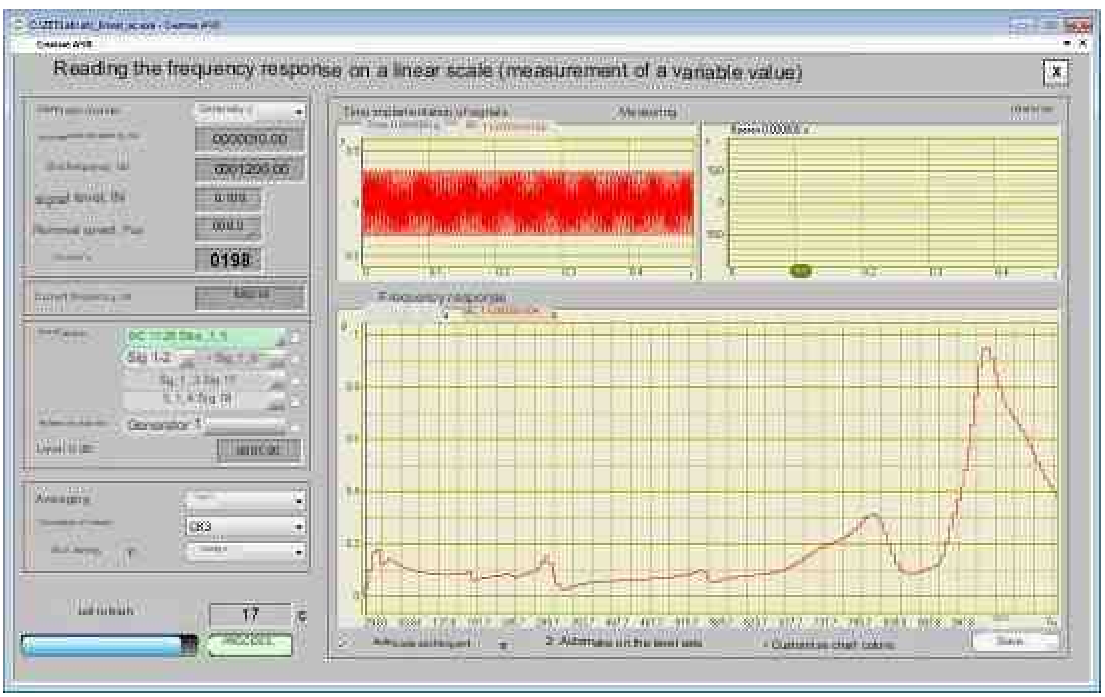
Fig. 5.5.2. Window of the program "**AFR measurement at linear sweep by frequency (DC)**". The project is in the process of reading the frequency response with a Linear sweep.

Example. Determination of the resonant frequency of the suspension

Resonant frequencies are determined by the frequency response of acceleration or displacement taken at a constant value of the excitation parameter. At the same time, the excitation parameter maintained constant is set so that acceleration and displacement do not exceed the maximum permissible values, and for electromechanical shakers, the displacement should be minimal.

The resonant frequency of the suspension corresponds to the first acceleration peak at least 1.5 times higher than the acceleration at a frequency of 400 Hz for electrodynamic shakers; at a frequency equal to approximately fhfor electromechanical shakers and at a frequency equal to approximately (fh * fb) 1/2 - for other types of shakers.

To measure the frequency response parameters, the **AFR measurement at log. sweep by frequency (AC)** programs are designed and **AFR measurement at lin. sweep by frequency (AC)** (the programs are implemented in SCADA ZETVIEW). Fig. 5.4.3 shows the appearance of the program with the frequency response of the shaker being tested already removed.



5.4.3 Appearance of the program with the frequency response of the shaker being tested already removed.

To remove the frequency response of the **Vibration generator system**, you need to set the frequency range, the output level from the generator that will be fed to the vibrating stand, and the signal type: linear-frequency modulated with linear (LinCHM) or logarithmic (LogCHM) frequency sweep. The program allows you to record the frequency response at 8 controlled points simultaneously. The frequency response can be displayed on either a linear or logarithmic scale.

In our case, we used a signal with a linear frequency sweep (LinCHM signal) from 20 Hz to 2.5 kHz. Thanks to the convenient properties of the Grid graphic component used in the frequency response measurement program, the results of the frequency response measurement can be written to a results file with the DTU extension. You can open the results file using the **Results Viewing** (see Fig. 5.4.4).

Fig. 5.4.4 You can view the results file using the Results **Viewer program**.

There is also another method for estimating the resonant frequencies not only of shaker, but also of various structures, accessories for attaching products to a shaker, etc.This method is based on the use of modal analysis to determine the dynamic characteristics of structures based on measurement results and analysis of forced mechanical vibrations. To do this, you need to run the Modal Analysis program included in ZETLAB.

For modal analysis, you must use an impact hammer (see Fig. 5.4.5). The impact hammer is a structure very similar to a conventional hammer, but only a vibration sensor is attached to the end of the digital sensor. The handle ends with a cable and a connector for connecting to the input channels of spectrum analyzers.

Fig. 5.4.5 Impact hammer.

To obtain a shock effect, you can also use **the Shock impulse Generator program**, which allows you to reproduce Vibration generator system Sine, triangular and rectangular impulses with a given duration and amplitude on the vibration system.

On Fig. 5.4.6 shows the main window **of the Modal Analysis program** after playing a shock using **the Shock impulse Generator program**. The impulse form is shown in the upper graphic. The lower graphic shows the shape of the signal received from the control vibration sensor installed on the shaker. Synchronization and stopping of graphics by time occurs automatically, and the threshold for triggering synchronization is configured at the user's request.

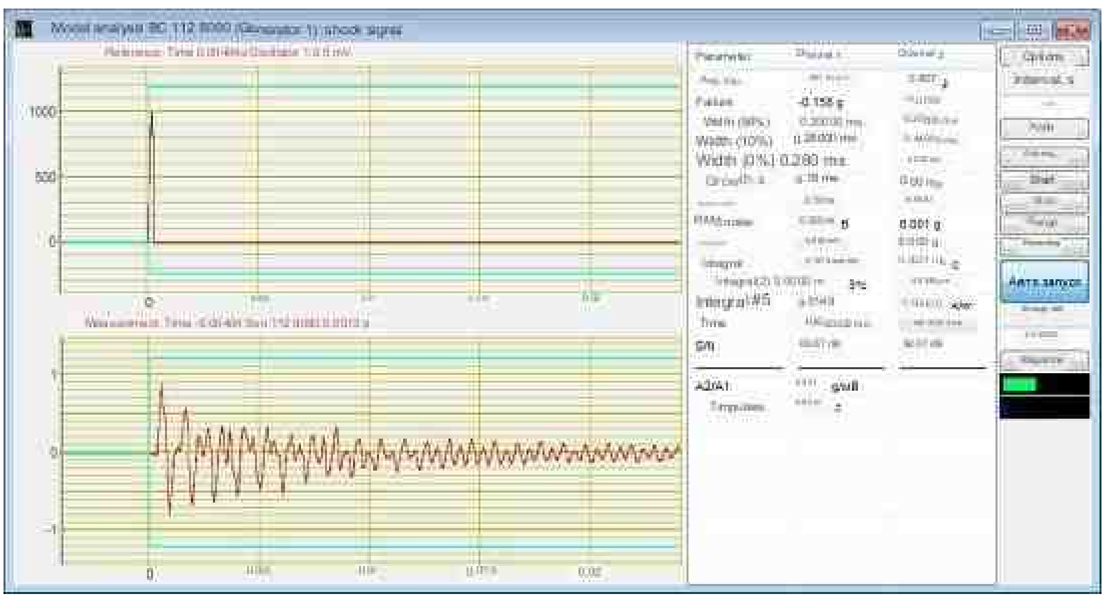


Fig. 5.4.6 Main window **of the Modal Analysis program**

When you click on **the Spectrum button** in **the Modal Analysis program**, an additional window opens (Fig. 5.4.7), showing the shape of the narrow-band spectrum of the shock impulse (upper graphic) and the signal from the control vibration sensor (lower graph). The right part **of the Spectrum window** shows the calculated values of Q-factors — decay decrements. The larger the attenuation decrement, the longer the residual phenomenon at the controlled point from the impact. This value will be useful for design departments that can make design changes during development (refinement or modernization) to reduce low-frequency or high-frequency resonances, and shift them beyond the operating frequency range.

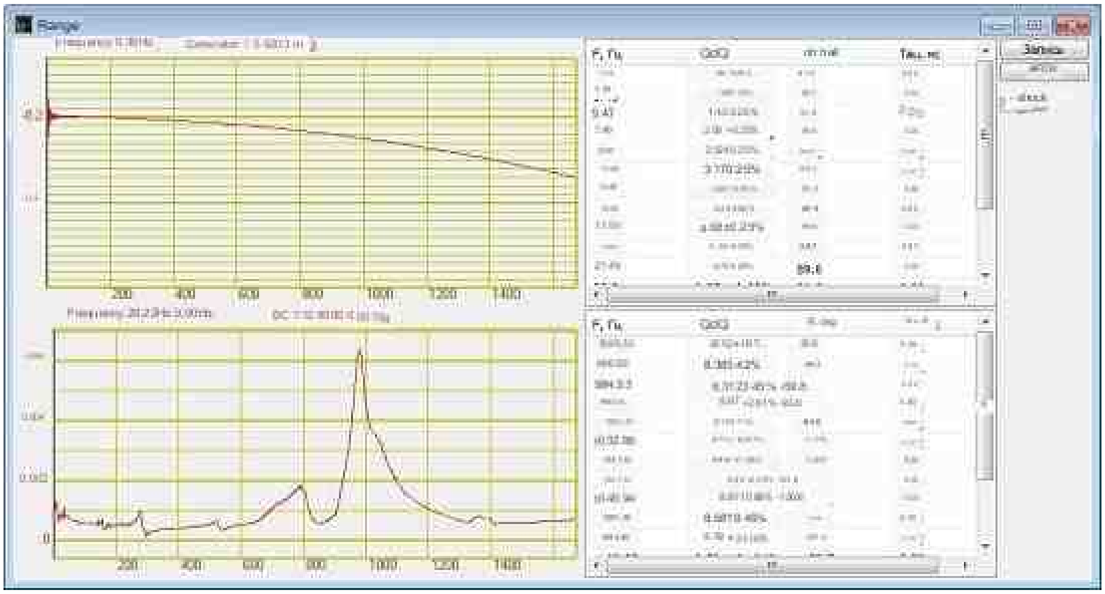


Fig. 5.4.7 An additional window opens Spectrum

By performing a visual and numerical comparison, you can make sure that the "pictures" in Fig.5.4.3 the lower graphic in Fig.5.4.7 coincide, i.e. the values of resonant frequencies obtained in two ways coincide: when using the frequency response measurement program **and the Modal Analysis** program. This allows us to speak about the applicability of the two methods and the reliability of the results in the study of the frequency characteristics of shakers.

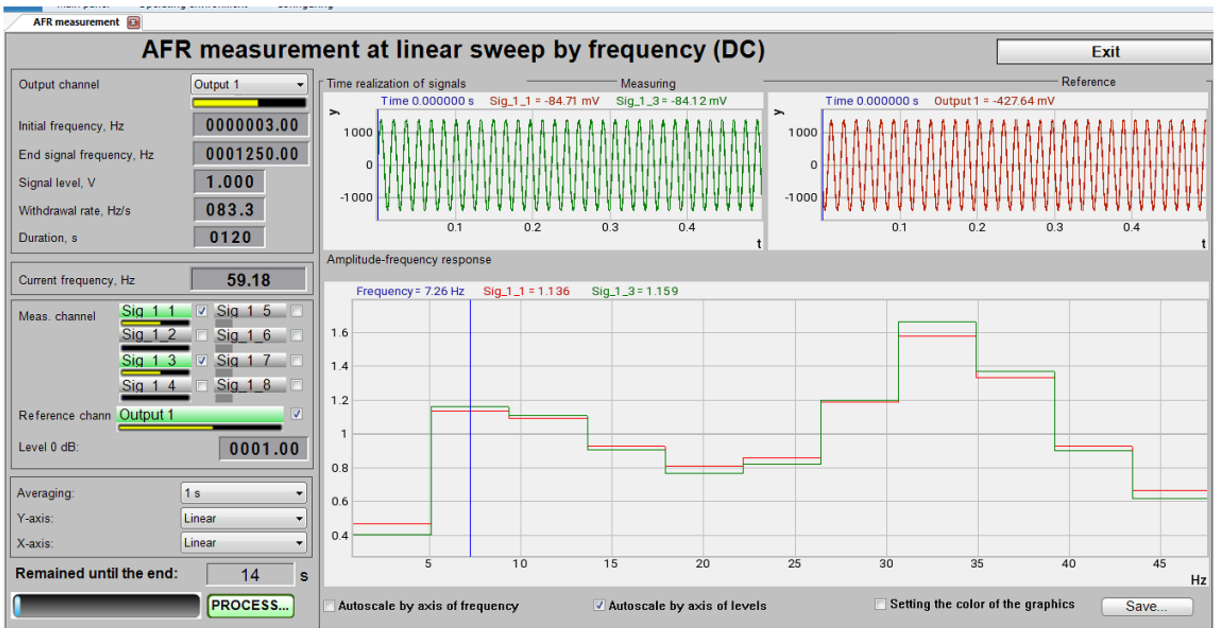
As can be seen from Figures 5.4.4 and 5.4.7, the graphs for different methods of estimating resonant frequencies coincide. The method using an shock hammer allows you to quickly measure the resonant characteristics not only of shakers, but also of any other structures. Just one blow with a hammer — and the picture is ready — there is a working frequency range and resonant frequencies of the studied structure.

1.6.AFR measurement at linear sweep by frequency (DC)

The program "**AFR measurement at linear sweep by frequency (DC)**" is designed to determine the amplitude-frequency characteristic when measuring the amplitude of the constant component of the signal. Measurements can be carried out on 8 channels simultaneously.
For measurements, certified equipment is used - spectrum analyzer type **ZET017** or type **A19** and SCADA **ZETVIEW.**
The spectrum analyzer's built-in generator is used to reproduce the test signal (a Sine signal with a serrated varying frequency). This signal is applied to the input of the device under test or to the input of the test equipment (depending on the measurement scheme). The signals under test are connected

to the input channels of the spectrum analyzer. Signal parameters are measured with a DC voltmeter. The graphic plots the amplitude-frequency characteristic in absolute or relative units.

Measurement results are saved in a file format . *dtu* , which is opened in the **Results viewing** program in graphical and numerical form.



AFR measurement at linear sweep by frequency (DC)

The program **AFR measurement at linear sweep by frequency (DC)** is implemented in the **ZETVIEW** SCADA system **.** When you start the program **AFR measurement at linear sweep by frequency (DC)** in the hidden mode, the programs **Signals generator** (one copy) and **AC Voltmeter** (from 1 to 9 copies - according to the number of measurement channels) are launched. To remove the frequency response, a frequency-modulated signal with a logarithmic (**LogFM**) frequency sweep is used.

Taking measurements

To carry out measurements it is necessary:

- select oscillator channel - if several spectrum analyzers are connected to the computer, it is necessary to select a master oscillator;
- set the start and end frequency of measurements - it should be remembered that measurements in the set frequency range will be carried out with the same data averaging value, so measurements, in Example , in the range from 0.1 to 1000 Hz are best done in 2 stages: in the range from 0.1 to 20 Hz with an average of 10 seconds and in the range from 10 to 1000 Hz with an average of 0.1 s;
- set the generator signal level - in the process of measurements, a signal with a given voltage (RMS) with a serrated changing frequency from the initial value to the final value will be generated at the generator output;

- set frequency response rate - determines the frequency change rate of the generator signal;
- the duration of the measurements will be calculated automatically according to the values of the initial and final frequencies and the frequency response rate. When changing the duration value, the frequency response rate value will be automatically recalculated.

Note : when changing the start or end frequency, the duration of the measurements changes, when the duration changes, the frequency response rate changes, when the speed changes, the duration of the measurements changes.

Also, before starting the measurements, you should select the measuring channel - one or more - in the program **AFR measurement at linear sweep by frequency (DC).** Measurements can be carried out on 8 channels simultaneously.

To perform relative measurements, you should also select the reference channel and set the flag in the "Reference channel" line.

Note:

- when carrying out *absolute measurements,* the measurement result is the signal voltage along the measuring channel
- when carrying out *relative measurements* , the measurement result is the ratio of the signal voltage along the measuring channel to the signal voltage along the reference channel.

Level 0 dB Set the zero level when displaying results in dB. in Example , when carrying out relative measurements with the *0 dB level parameter* equal to 1, the measurement result $U_{meas} / U_{op} = 1$ will be recorded on the graphic at point 0.

AFR measurement at linear sweep by frequency (DC) program allows you to take measurements with an average of 0.1, 1 or 10 seconds, calculating the RMS, peak or amplitude value of the signal, presenting the result in a linear or logarithmic scale along the Y axis .

In the process of measurements, the time realizations of signals on the measuring and reference channels are displayed (on the oscillograms from above), the current frequency and the time remaining until the end of the measurements (on the indicators on the left) and a graphic of the frequency response is plotted.

Measurement results can be saved to a file for viewing in display programs (in graphical and numerical form) and printing.

Hardware support

The program for **AFR measurement at linear sweep by frequency (DC)** is supplied with spectrum analyzers (for Example, **ZET 017-U2, ZET 017-U4, ZET 017-U8, A19) in the presence of SCADA ZETVIEW.**

Supported hardware

The program "**AFR measurement at linear sweep by frequency (DC)**" is a part of the following software:

ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately

ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.6.1) **ZETLab control panels** press the button **AFR measurement at linear sweep by frequency (DC)**.

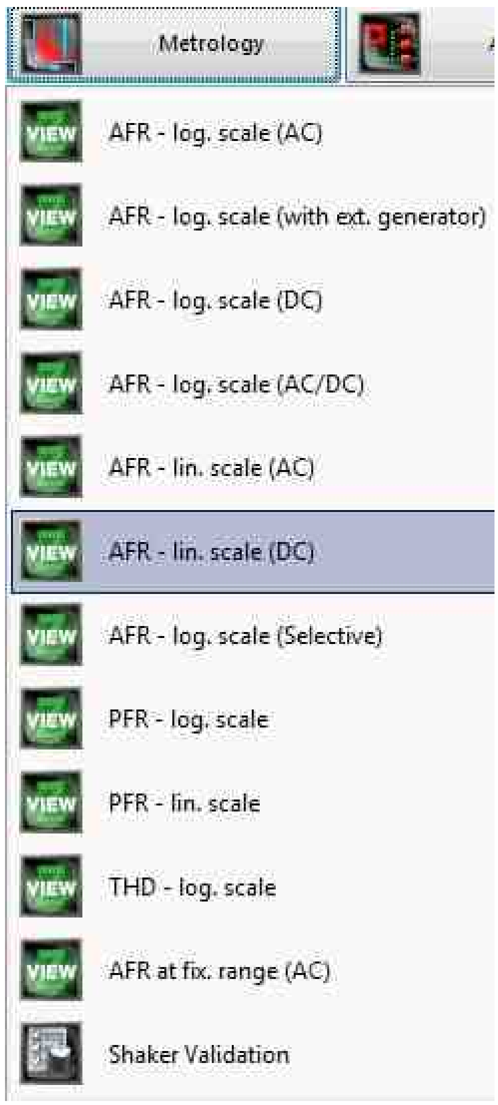


Fig. 5.6.1. List of menu programs "Metrology".

In the opened window of the program **AFR measurement at linear sweep by frequency (DC)** (Fig. 5.6.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to such a value that acceleration and displacement do not exceed the maximum allowable values.);
- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

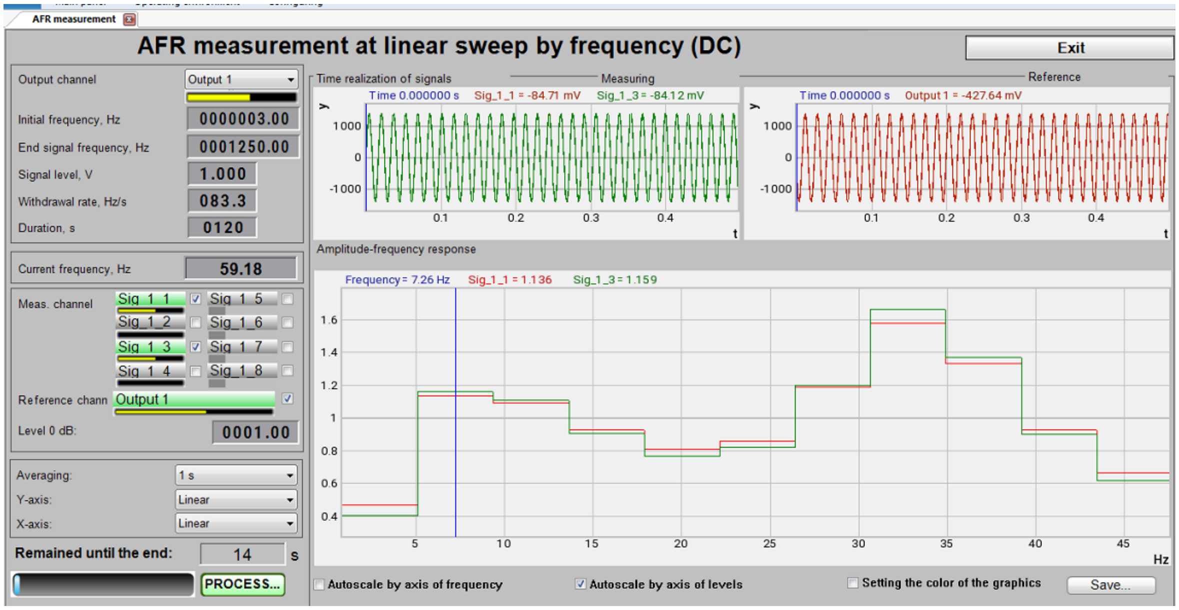


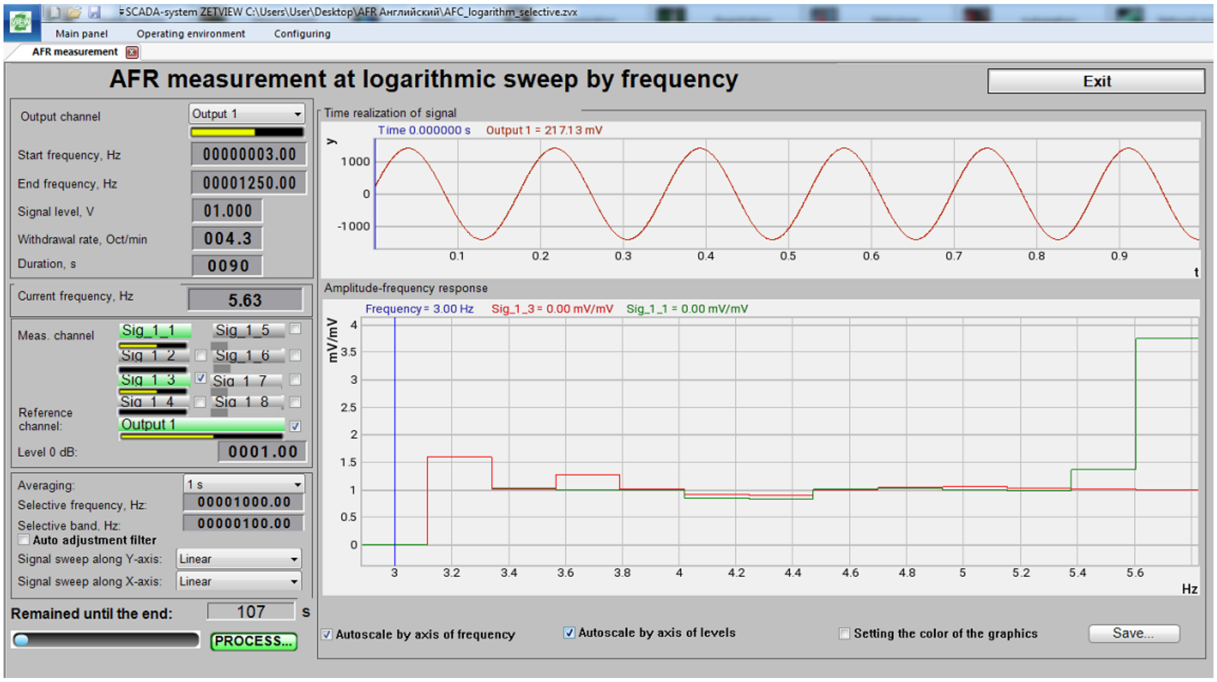
Fig. 5.6.2. Window of the program "**AFR measurement at linear sweep by frequency (DC)**". The project is in the process of reading the frequency response with a linear sweep.

1.7.AFR - log. scale (Selective)

The program "**AFR - log. scale (Selective)**" is designed to determine the amplitude-frequency characteristics using a selective voltmeter. Measurements with a selective voltmeter make it possible to neutralize the contribution of a random component to the resulting signal, thus ensuring high measurement accuracy in conditions of strong interference.

The selective voltmeter used in the frequency response removal program makes measurements in a narrow frequency band (set by users or auto-tuned). Measurements can be carried out on several channels simultaneously (up to 8).
A Sine signal generator is also used to measure the frequency response. The frequency of the generator varies in a given range according to the logarithmic law.

Measurement results are saved in a file format . *dtu* , which is opened in the **Results viewing** program in graphical and numerical form.



AFR - log. scale (Selective)

The program **AFR - log. scale (Selective)** in the hidden mode, the programs **Signals generator** (one copy) and **Selective voltmeter** (from 1 to 9 copies - according to the number of measurement channels) are launched. To remove the frequency response, a frequency-modulated signal with a linear **(LogFM)** frequency sweep is used.

Taking measurements

To carry out measurements it is necessary:

- select oscillator channel - if several spectrum analyzers are connected to the computer, it is necessary to select a master oscillator;
- set the start and end frequency of measurements - it should be remembered that measurements in the set frequency range will be carried out with the same data averaging value, so measurements, in Example , in the range from 0.1 to 1000 Hz are best done in 2 stages: in the range from 0.1 to 20 Hz with an average of 10 seconds and in the range from 10 to 1000 Hz with an average of 0.1 s;
- set the generator signal level - in the process of measurements, a signal with a given voltage (RMS) with a serrated changing frequency from the initial value to the final value will be generated at the generator output;
- set frequency response rate - determines the frequency change rate of the generator signal;
- the duration of the measurements will be calculated automatically according to the values of the initial and final frequencies and the frequency response rate. When changing the duration value, the frequency response rate value will be automatically recalculated.

Note : when changing the start or end frequency, the duration of the measurements changes, when the duration changes, the frequency response rate changes, when the speed changes, the duration of the measurements changes.

Also, before starting the measurements, you should select the measuring channel - one or more - in the program **AFR - log. scale (Selective).** Measurements can be carried out on 8 channels simultaneously.
To perform relative measurements, you should also select the reference channel and set the flag in the "Reference channel" line.
Note:

- when carrying out *absolute measurements,* the measurement result is the signal voltage along the measuring channel
- when carrying out *relative measurements* , the measurement result is the ratio of the signal voltage along the measuring channel to the signal voltage along the reference channel.

Level 0 dB Set the zero level when displaying results in dB. in Example , when carrying out relative measurements with the *0 dB level parameter* equal to 1, the measurement result $U_{meas} / U_{op} = 1$ will be recorded on the graphic at point 0.
The program **AFR - log. scale (Selective)** program allows you to take measurements with an average of 0.1, 1 or 10 seconds, calculating the RMS, peak or amplitude value of the signal, presenting the result in a linear or logarithmic scale along the Y or X-axises.
In the process of measurements, the time realizations of signals on the measuring and reference channels are displayed (on the oscillograms from above), the current frequency and the time remaining until the end of the measurements (on the indicators on the left) and a graphic of the frequency response is plotted.
Measurement results can be saved to a file for viewing in display programs (in graphical and numerical form) and printing.

Hardware support

The program for **AFR measurement at linear sweep by frequency (AC)** is supplied with spectrum analyzers (for Example, **ZET 017-U2, ZET 017-U4, ZET 017-U8, A19) in the presence of SCADA ZETVIEW.**

Supported hardware

The program "**AFR measurement at linear sweep by frequency (DC)**" is a part of the following software:
ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.7.1) **ZETLab control panels** press the button **AFR - log. scale (Selective)**.

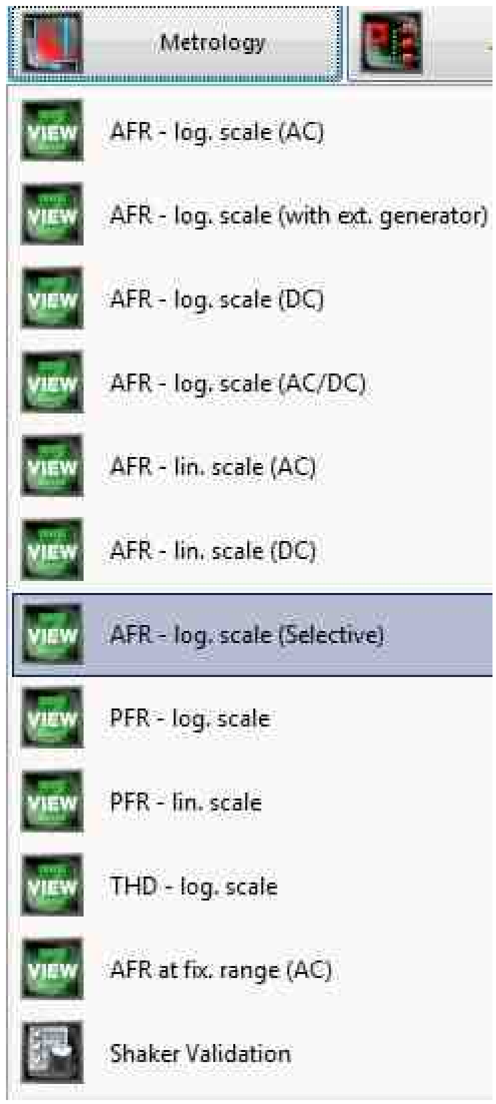


Fig. 5.7.1. List of menu programs "Metrology".

In the opened window of the program **AFR - log. scale (Selective)** (Fig. 5.6.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to such a value that acceleration and displacement do not exceed the maximum allowable values.);
- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the selector **"Selective frequency, Hz"** - selective frequency range in which the frequency response will be taken;
- in the selector **"Selective band, Hz"** - selective band of the frequency range in which the frequency response will be taken;
- flag **"Selective filter auto-tuning** " - is used for auto-tuning of the selective filter.
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

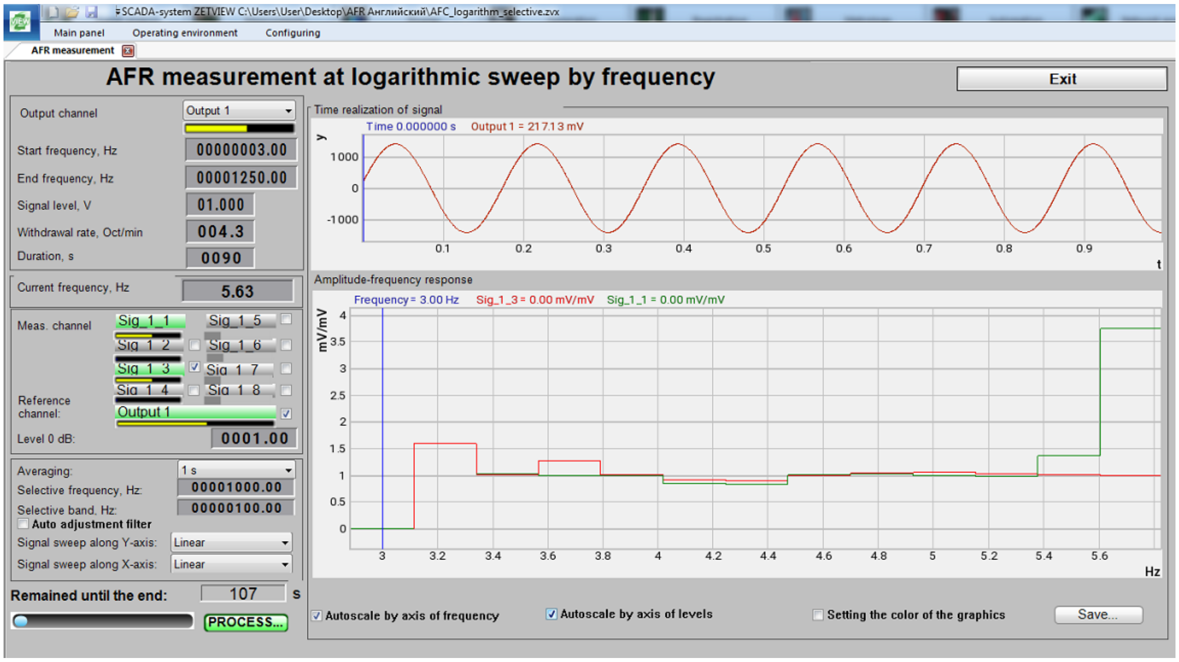


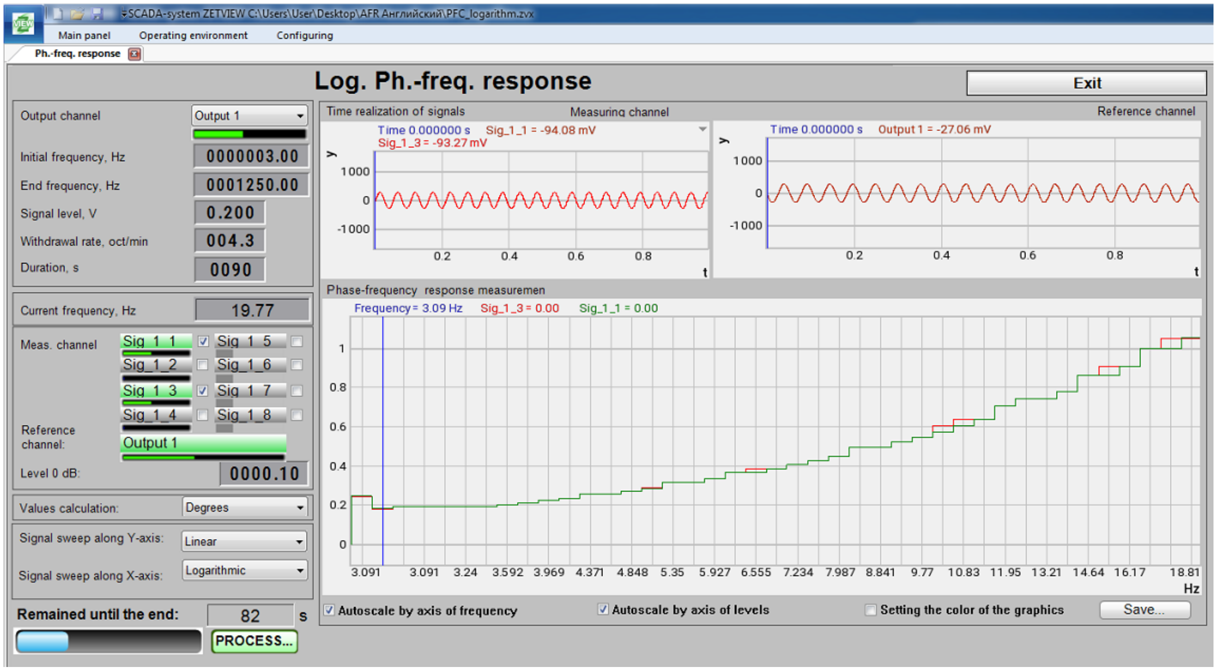
Fig. 5.7.2. Window of the program "**AFR - log. scale (Selective)**". The project is in the process of reading the frequency response with a linear sweep.

1.8.Log. Ph.-freq. response

The program "**Log. Phase-frequency response"** is designed to determine the phase response in a given frequency range. Measurements can be carried out on 8 channels simultaneously.

The PFC removal program **on a logarithmic scale** uses a Sine signal generator, which gives a testing effect, and phase meters that determine the phase difference between the measuring and reference channels. A plot of phase difference versus signal frequency is displayed in the "Phase response" field. The program provides the following options: absolute or relative measurements, various scales of graphics, setting the initial and final frequencies, as well as the speed of taking and the duration of measurements, measurements in degrees or radians. Additionally, during the measurement process, an oscillogram of the generator signal, the time until the end of the measurement and the current frequency are displayed.

Measurement results are saved in a file format . *dtu* , which is opened in the **Results viewing** program in graphical and numerical form.



Log. Phase-frequency response

Supported hardware

The program "**Log. Phase-frequency response**" is a part of the following software:
ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.8.1) **ZETLab control panels** press the button **Log. Phase-frequency response**.

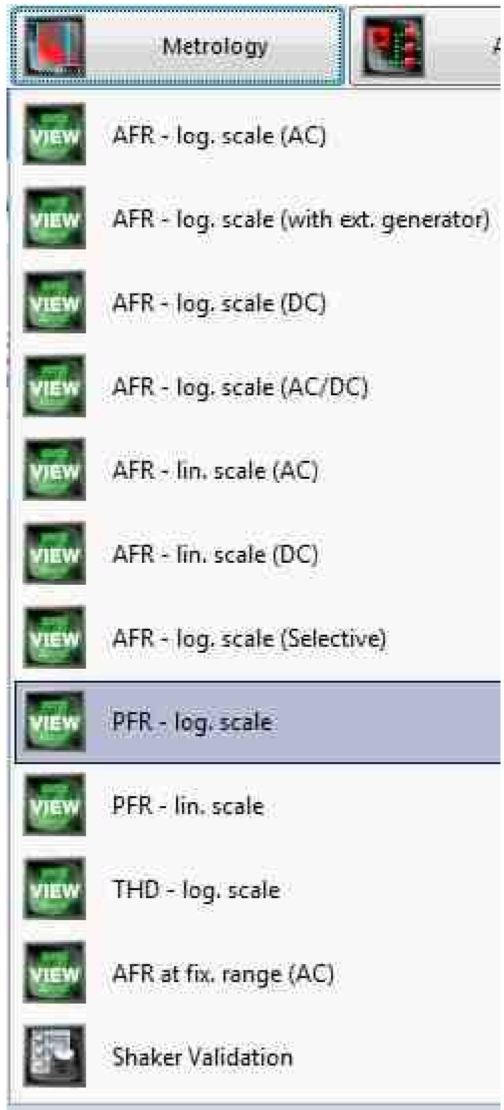


Fig. 5.8.1. List of menu programs "Metrology".

In the opened window of the program **Log. Phase-frequency response** (Fig. 5.8.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to

such a value that acceleration and displacement do not exceed the maximum allowable values.);

- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

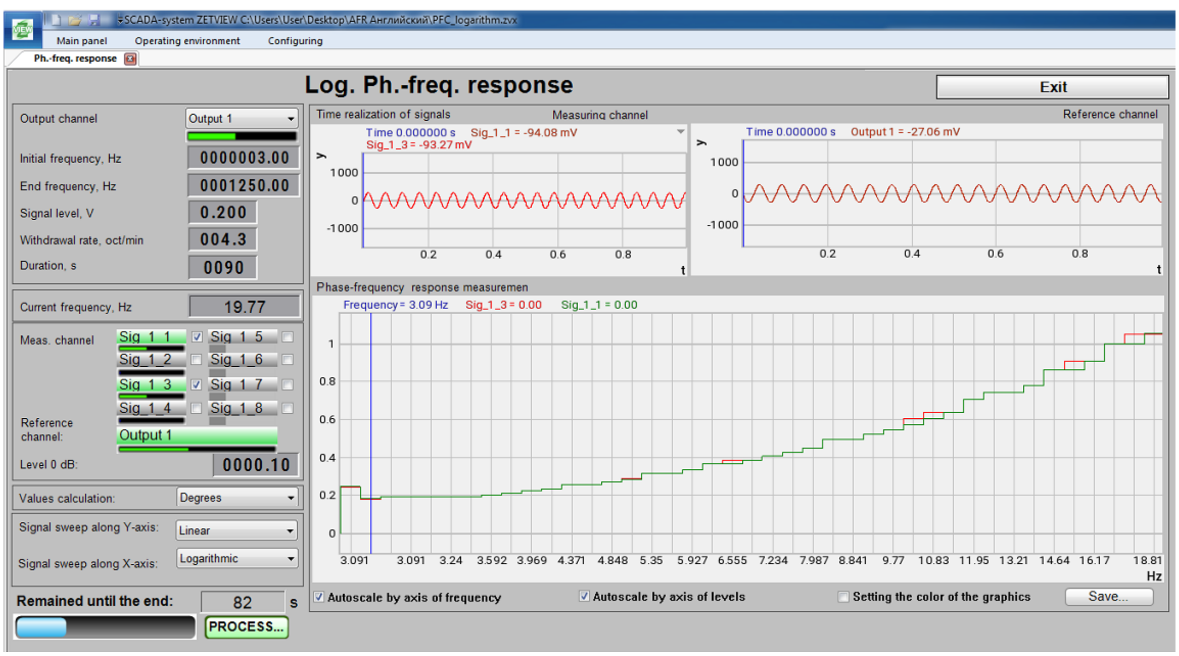


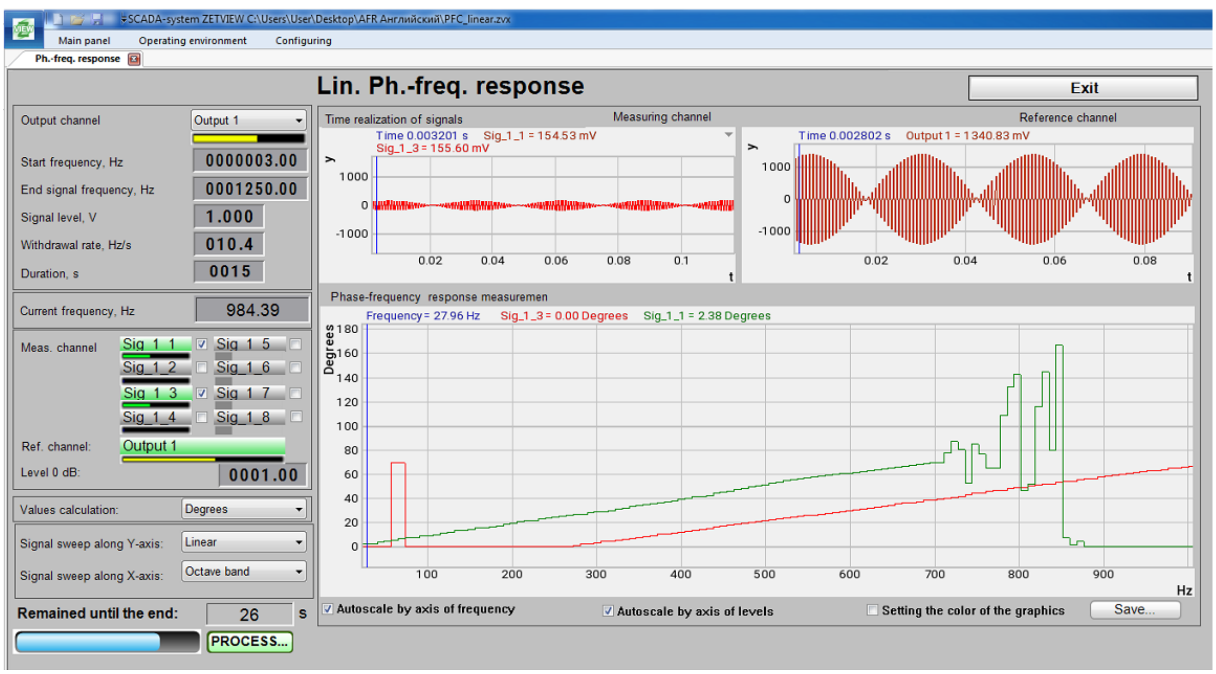
Fig. 5.8.2. Window of the program "**Log. Phase-frequency response**". The project is in the process of reading the frequency response with a linear sweep.

1.9.Lin. Ph.-freq. response

The program " **Lin. Phase-frequency response (PFC)**" is a part of the following software:
ZETLAB ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
· ZETLAB VIBRO - the software supplied with the shaker control system, the ZETView option is included.

The PFC removal program **on a linear scale** uses a Sine signal generator, which gives a testing effect, and phase meters that determine the phase difference between the measuring and reference channels. A plot of phase difference versus signal frequency is displayed in the "Phase response" field. The program provides the following options: absolute or relative measurements, various scales of graphics, setting the initial and final frequencies, as well as the speed of taking and the duration of measurements, measurements in degrees or radians. Additionally, during the measurement process, an oscillogram of the generator signal, the time until the end of the measurement and the current frequency are displayed.

Measurement results are saved in a file format . *dtu* , which is opened in the **Results viewing** program in graphical and numerical form.



Lin. Phase-frequency response (PFC)

Supported hardware

The program "**Log. Phase-frequency response**" is a part of the following software:
ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.9.1) **ZETLab control panels** press the button **Lin. Phase-frequency response (PFR)**.

Fig. 5.9.1. List of menu programs "Metrology"

In the opened window of the program **Lin. Phase-frequency response (PFR)** (Fig. 5.9.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to such a value that acceleration and displacement do not exceed the maximum allowable values.);
- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Calculation of values"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.

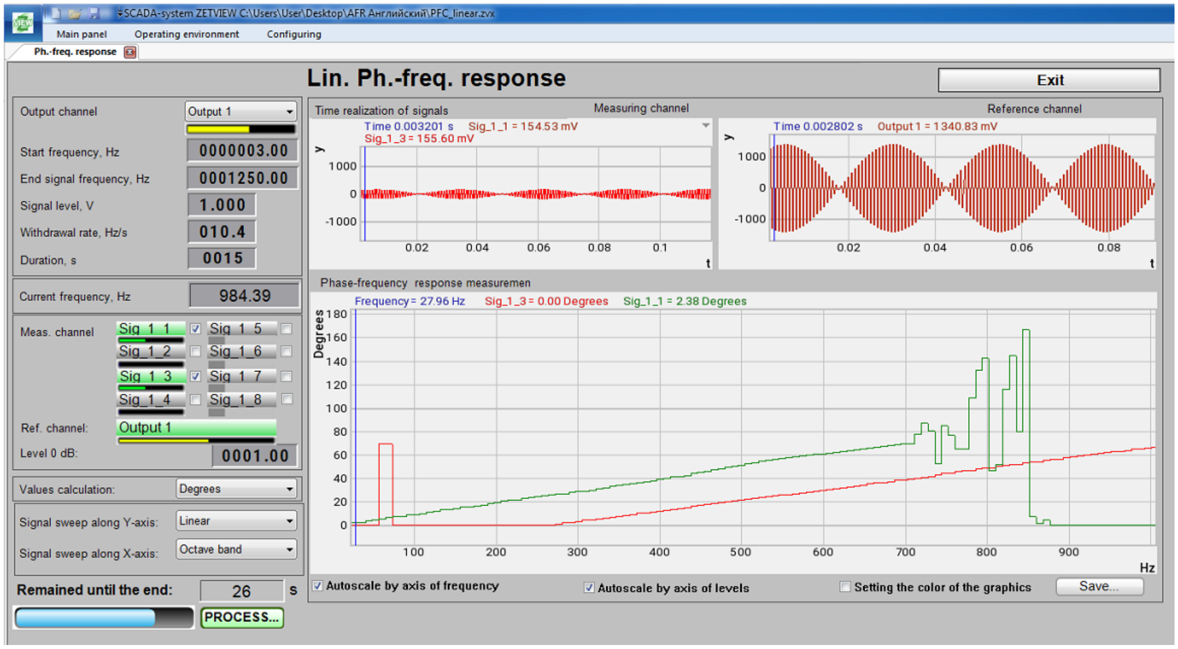


Fig. 5.9.2. Window of the program "**Log. Phase-frequency response (PFR)**". The project is in the process of reading the frequency response with a linear sweep.

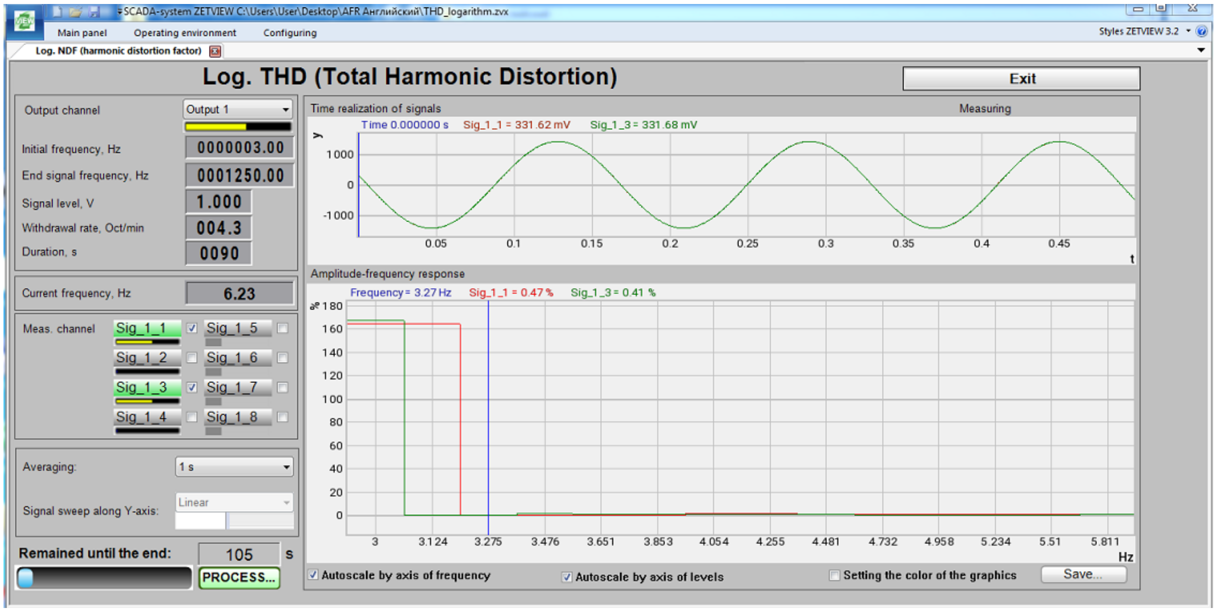
1.10.Log. THD (Total Harmonic Distortion)

The program "**Log. THD (Total Harmonic Distortion)**" is designed to determine the coefficient of non-linear distortion in the selected frequency range.

For measurements, certified equipment is used - spectrum analyzer type **ZET 017** or type **A19** and SCADA **ZETVIEW**.

For measurements, certified equipment is used - spectrum analyzer type **ZET 017** or type **A19** and SCADA **ZETVIEW**.
A Sine signal with a given amplitude is formed at the output of the generator. The frequency of the signal changes in a given range at a given rate on a logarithmic scale. The signal from the generator is fed to the input of the device under test or test equipment. The signal under test is connected to the input of the measuring device and the **Harmonic Distortion Analysis** is carried out. The result of the measurements is displayed as a graphic of the dependence of the **Total Harmonic Distortion** (THD) on the frequency. Additionally, during the measurement process, the graphic of the generator signal, the current frequency, and the time until the end of the measurements are displayed.

The measurement results are saved in a file format *.dtu*, which is opened in the **Results viewing** program in graphical and numerical form.



Log. THD (Total Harmonic Distortion) vs frequency

Supported hardware

The program "**Log. THD (Total Harmonic Distortion)**" is a part of the following software:
ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.10.1) **ZETLab control panels** press the button **Log. THD (Total Harmonic Distortion)**.

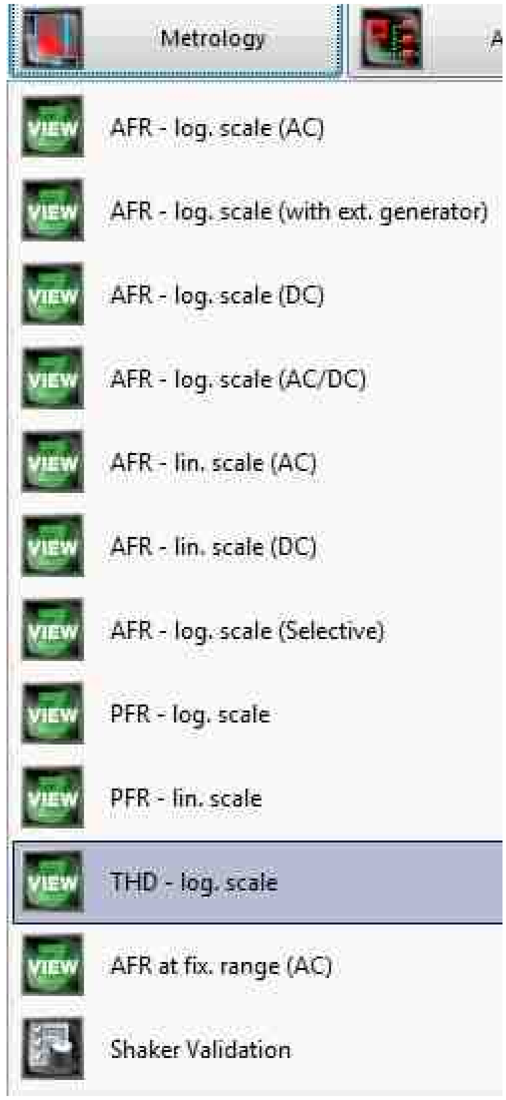


Fig. 5.10.1. List of menu programs "Metrology".

In the opened window of the program **Log. THD (Total Harmonic Distortion)** (Fig. 5.10.2) you need to set:

- in the drop-down list **"Generator channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the selector **"Start frequency, Hz"** - the initial frequency of the frequency range in which the frequency response will be taken;
- in the selector **"End frequency, Hz"** - the final frequency of the frequency range in which the frequency response will be taken;
- in the window **"Signal level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to

such a value that acceleration and displacement do not exceed the maximum allowable values.);

- in the window **"Duration, s"** – sweep duration in seconds;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list **"Averaging"** - the time interval for which one point on the chart will be considered;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

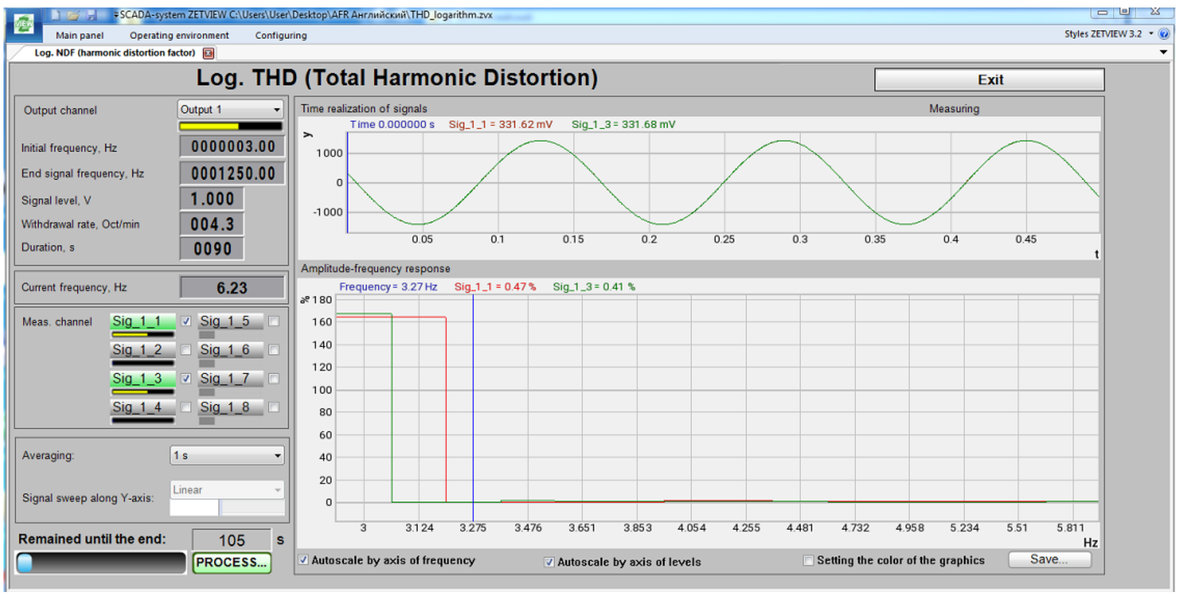


Fig. 5.10.2. Window of the program "**Log. Phase-frequency response (PFR)**". The project is in the process of reading the frequency response with a linear sweep.

1.11.AFR measurement fixed frequency range (AC)

The program "**AFR measurement fixed frequency range (AC)"** is designed to measure the amplitude-frequency characteristic and can be used for:

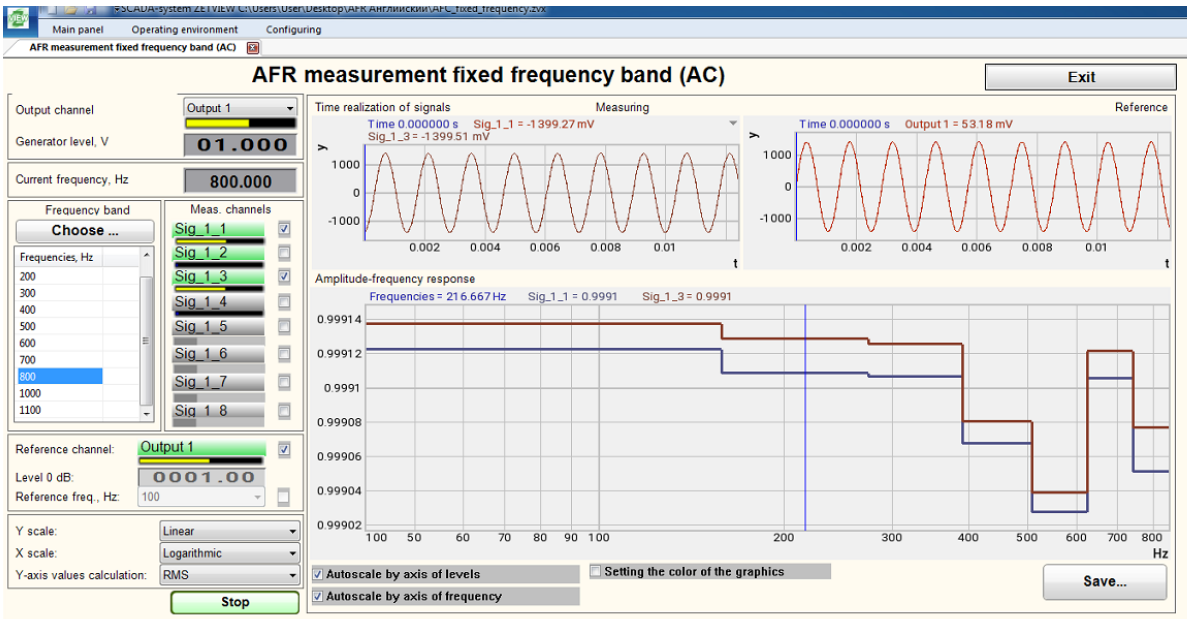
:

- **AFR measurement** of primary transducers by the electric method or on a shaker,
- **AFR measurement** of monomorphic piezoelectric elements,
- **AFR measurement** of shaker,

- etc.

When **AFR measurement**, a signal with the specified parameters is formed at the output of the generator. On the input channels of the spectrum analyzer, the response of the tested path is recorded. Measurement results and recorded signals are displayed in the program in real time.

The measurement results are saved in a file format *.dtu, which is opened in the **Results viewing** program in graphical and numerical form.



AFR measurement fixed frequency range (AC)

Supported hardware

The program "**AFR measurement fixed frequency range (AC)**" is a part of the following software:
ZETLAB _ ANALIZ - software supplied with spectrum analyzers, ZETView option sold separately
ZETLAB _ VIBRO - software supplied with shaker control system, ZETView option included.

ZETLAB is manifested by more than 10 times In this case, the generated signal fully complies with the specified parameters. Therefore, with such settings, the virtual signal cannot be used for calculations, in Example , using an arithmometer, or as a reference channel signal in the Strain-gauge meter program. If the use of a virtual channel is necessary, the DAC sampling frequency must be lowered or the ADC sampling frequency must be increased.
Note 2: When connecting devices to a PC via Ethernet , the sampling frequency of the DAC should not exceed 100 kHz. If the device has multiple DAC outputs, then the DAC sampling frequency should not exceed 100/ N kHz, where N is the number of generators involved.

Program interface

To start the program for removing the frequency response of acceleration, it is necessary from the **Metrology menu** (Fig. 5.11.1) **ZETLab control panels** press the button **AFR measurement fixed frequency range (AC)**.

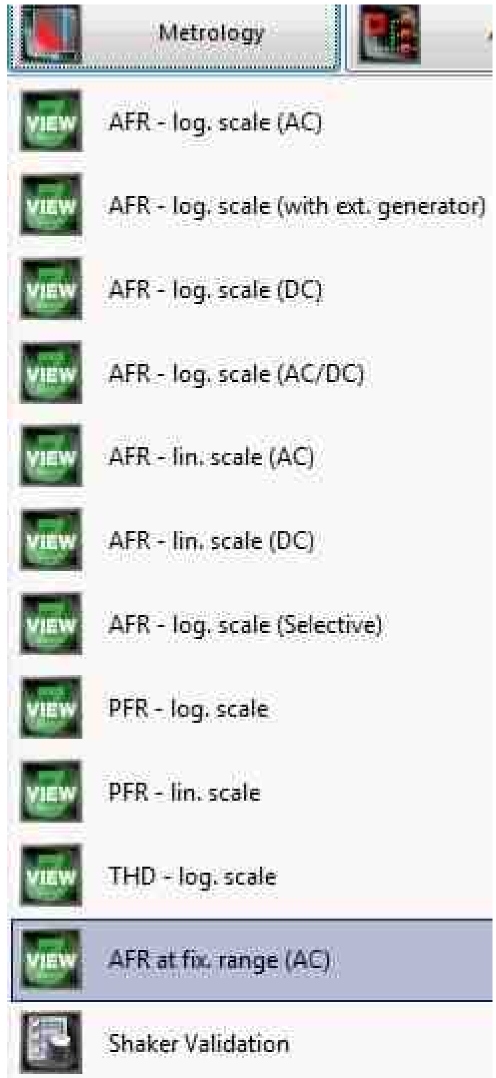


Fig. 5.11.1. List of menu programs "Metrology".

In the opened window of the program **AFR measurement fixed frequency range (AC)** (Fig. 5.11.2) you need to set:

- in the drop-down list **"Output channel"** - the generator channel that will be used when measuring the frequency response (by default, this is the generator launched by the program for measuring the frequency response);
- in the window **"Generator level, V"** – voltage of the output signal in volts from the analyzer generator, which will be supplied to the shaker (the output level remains constant and is set to

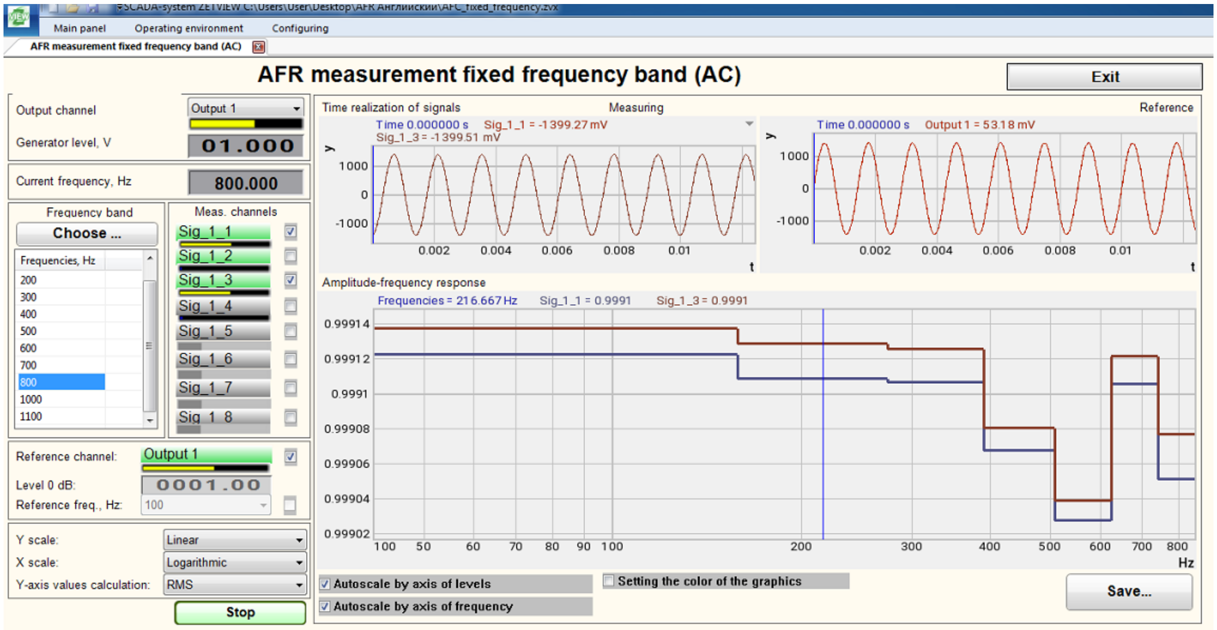
such a value that acceleration and displacement do not exceed the maximum allowable values.);

- in the selector **"Current frequency, Hz"** - the current frequency in which the frequency response will be taken;
- in the drop-down list "**Measuring channels**" - select the measuring channels and set the flag for enabling the channel in the project;
- in the drop-down list "**Frequency band**" - select the Frequency range presented from the selected directory Choose...;
- in the drop-down list "**Reference channel**" - select the reference channel and set the flag for enabling the channel in the project;
- in the "**Level 0, dB**" selector – the level equal to 0, dB is set;
- in the drop-down list "**Reference frequency, Hz**" - the choice becomes available if the Y Decibel scale is selected from the list, set the flag and select the frequency with which the project will work, and not be executed at all frequencies from the list;
- in the drop-down list **"Sweep along the Y axis "** - presentation of measurement results in a linear or logarithmic scale, or linearly in decibels.

To select the channels for which the frequency response will be measured, you must check the boxes next to the elements for selecting channels and select the channel. Once selected, the element will change its color to green.

- in the drop-down list " **Sweep along the X axis"** - presentation of measurement results along X in a uniform or logarithmic scale.
- in the drop-down list **"Y axis values calculation"** - select the measurement method: "RMS" for measuring root mean square values, "Amplitude" for measuring amplitude values or "Peak" for measuring peak values;

After setting up the program, you must click the **"Start"** button located in the lower left corner of the program. At the end of the measurement, the program will stop, and the acceleration frequency response curve will be displayed in the graphic field.



AFR measurement fixed frequency range (AC)

Глава 2.Multi-channel measurements

1) SCADA project "Multi-channel measuring system" is supplied as a separate installation file (the project requires ZETVIEW SCADA system 1348).

2) The SCADA-project "Multi-channel monitoring system" is supplied as a separate installation file (the project requires the ZETVIEW SCADA system 1348).

3) SCADA-project "Multi-channel strain gauge" is supplied as a separate installation file (the project requires the SCADA system ZETVIEW 1348).

2.1.Multi-channel measurements

ZETLAB users often need to take measurements on several channels at once, for example, take DC voltmeter readings from a large number of sensors at the same time. In addition to monitoring the readings, the user may need statistics on the readings of each sensor, and there may also be a need to save the readings to a separate file for data post-processing. For the convenience of the user, a special project was developed called "**Multi-channel tensometriy**" and "**Multi-channel measuring system**", which allows you to cope with all these tasks

Fig. 1. Project location "**Multi-channel tensometriy**" and "**Multi-channel measuring system**"

For the convenience of the user, the project is divided into 4 tabs: current values: measurement settings, temperature recorder, speed recorder.

a) Current values. On this tab, the user can select the number of channels with which the project will work, select the channels themselves, view the instrument readings at the moment and save them to a file for post-processing. The appearance of the "Current values" tab is shown in Fig. 2.

The project is universal, there is no distinction between speed and temperature sensors. The number of channels is up to 48. After selecting a sensor, the corresponding unit of measurement will appear next to the value window.

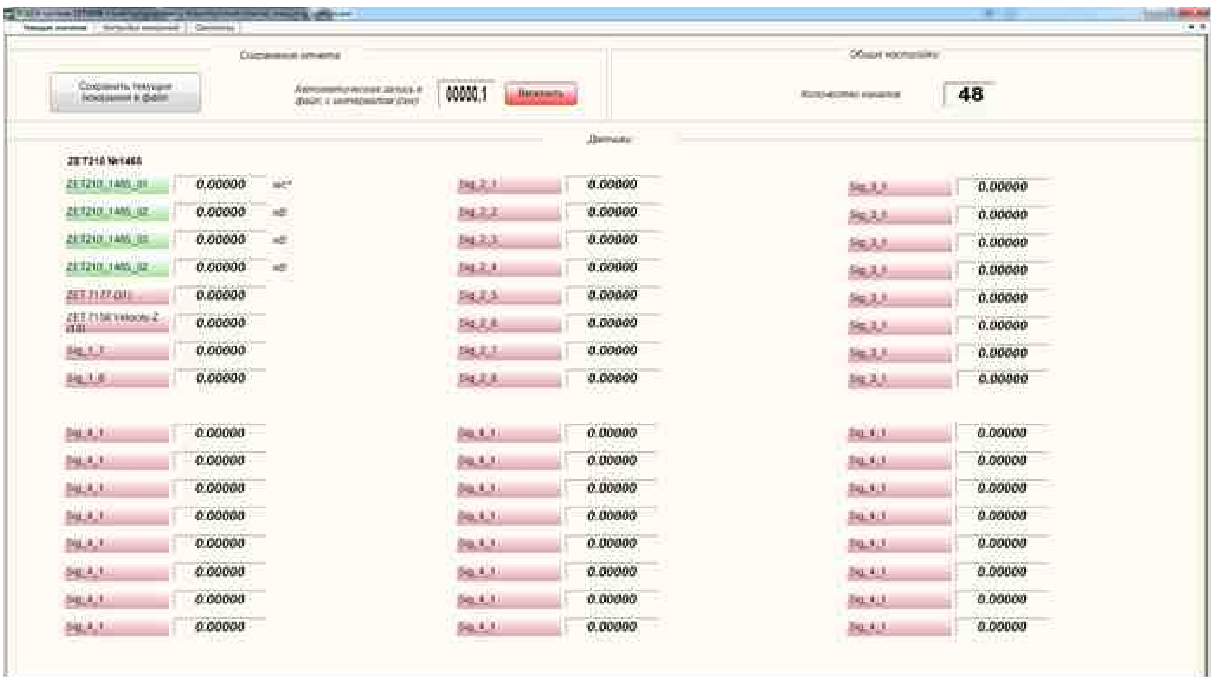


Fig. 2 Current values tab

b) Measurement setup. On this tab, the user can configure several measuring devices at the same time. The appearance of the "Measurement settings" tab is shown in fig. 3.

To work with sensors, you must select them in the "Current values" window. Then, in the "Measurement settings" window, tick off the sensors whose readings we want to record, select the desired averaging, make sure that the operation status is "on", and then click on "Apply for selected" (Fig. 3).

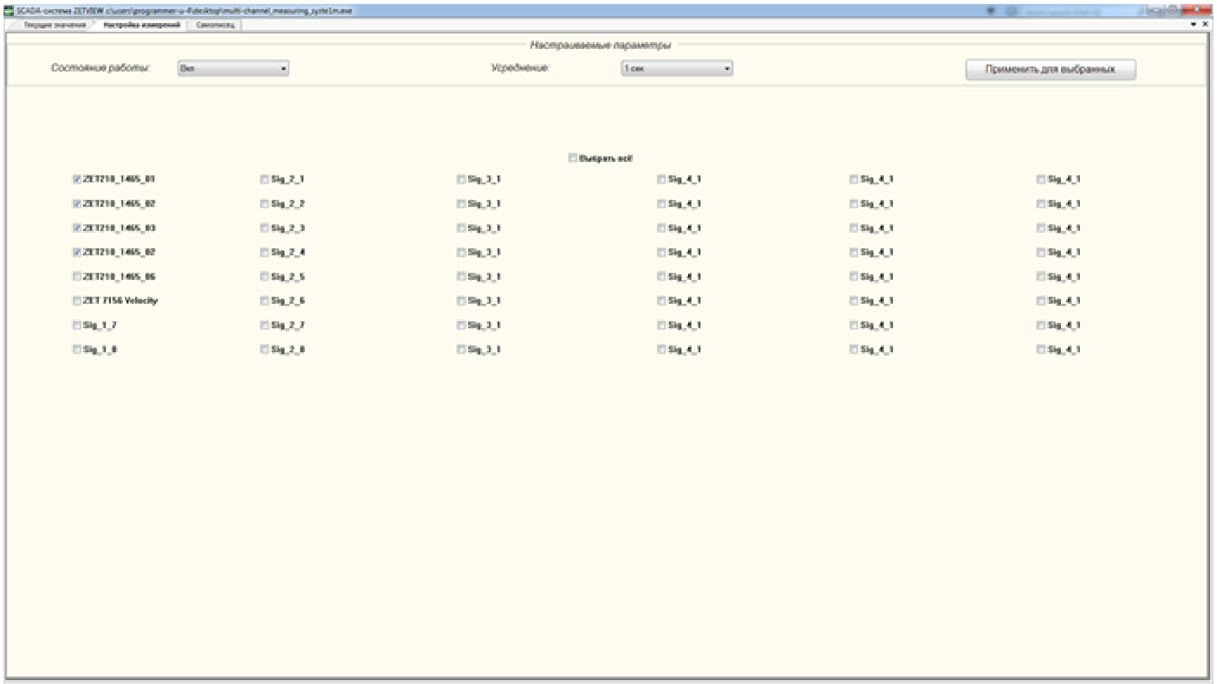


Fig. 3. Tab “Measurement settings”

c) Temperature recorder and speed recorder. These tabs are designed so that the user can track changes in instrument readings depending on time. The data is displayed as a graph and can be saved to a file for post-processing.

One recorder. The minimum value of the data update period is 0.05 s. The recorder records data on the sensors selected in the "Measurement settings" window (Fig. 4).

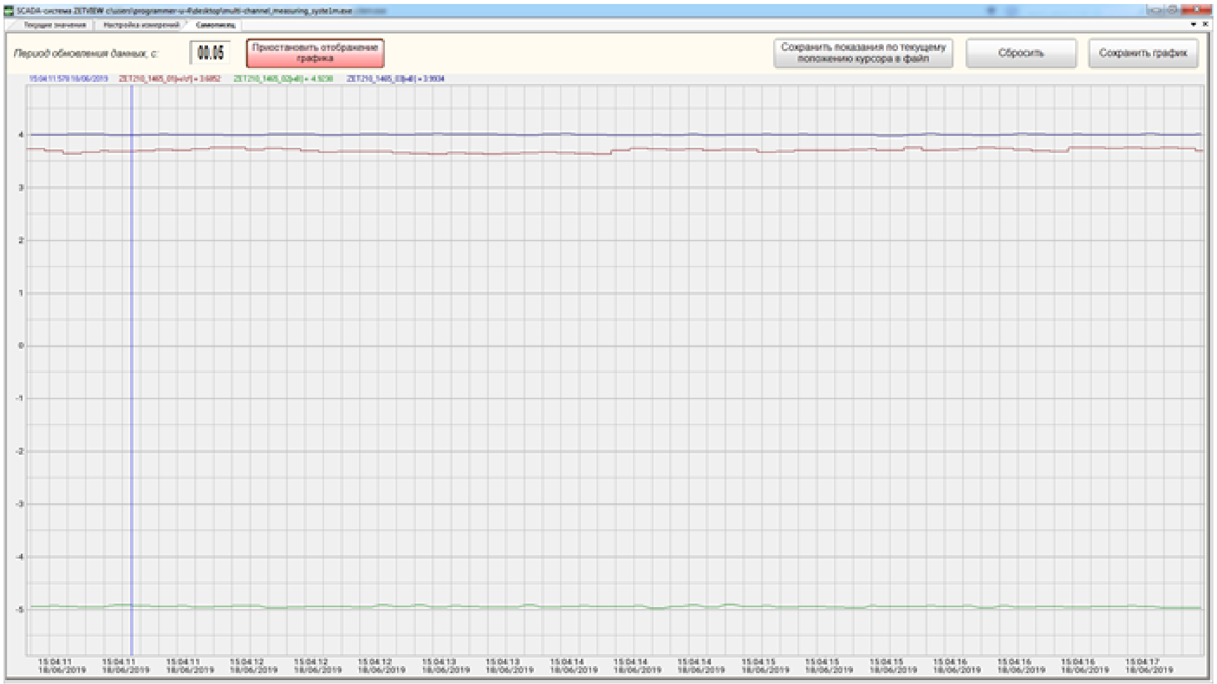


Fig. 4 The Multi-channel recorder records data on the sensors selected in the “Measurement settings” window

Changes from the old project:

1) There were two groups of sensors (speed and temperature), now a universal project;

2) There were two recorders, now one;

3) Added a unit of measurement by the sensor in the "Current values" window, the "Recorder" window, as well as when outputting information to a file;

4) In the "Recorder" window, the "Data update period" value has been corrected. Previously, the minimum value was displayed as zero, now it is exactly defined (0.05 s).

An example of working with the project can be viewed in our videos at the links:
[Automation of the process of certification of the heat and cold chamber](https://www.youtube.com/watch?v=zCdzq-W6P6A)
https://www.youtube.com/watch?v=zCdzq-W6P6A

[Summary of the Climate Chamber Qualification Process](https://www.youtube.com/watch?v=NTuqxcYlKPw)
https://www.youtube.com/watch?v=NTuqxcYlKPw

You can also upgrade and customize this project depending on your needs, the file with the SCADA project is located at: "\ZETLab\SCADA\Samples\Multi-channel measuring system .zvx".

2.2.Multi-channel tensometry

for processing 128 measuring channels

A **Multi-channel tensometry** based on **the strain gauge ZET 017-T has been developed**, which allows you to simultaneously process up to 128 measuring channels!
To set up the measuring channels and carry out measurements, the program **Multi-channel tensometry is used**, which is part of the **ZETLAB software**

Terms and definitions

The main terms and definitions are listed in the table (Table 61 1354).

Table 61

Accelerogram	Recording displacement, velocity, or acceleration as a time function.
Accelerometer	Primary converter (sensor) generating an electrical signal proportional to the registered acceleration.
Amplitude	The largest (by module) instantaneous values determining the signal for the averaging period.
Amplitude-frequency response (AFR)	The dependence of the amplitude of steady-state oscillations of the output signal of a certain system on the frequency of its input harmonic signal. Amplitude-frequency response is one of the types of "frequency response" of the system along with the phase-frequency response (PFR) and Amplitude-phase-frequency response (AFC).
Antiresonance	The frequency at which the response to the control signal (oscillator) is decreasing sharply (very small). Do not install sensors that will be assigned the Monitoring status (feedback channel) in areas of the test object with large antiresonances. If there are high value antiresonances, you can use multipoint control (by selecting "by average" or "by maximum" control mode in the Pre-Test) for several sensors with Monitoring status whose antiresonances are inconsistent in frequency.
Blades of gas turbine engines(SDS)	This is a separate component that makes up the turbine section of a gas turbine or steam turbine <159.>. The blades are responsible for extracting energy from high-temperature gas under high pressure generated by the combustion chamber. Turbine blades are often the limiting element of gas turbines.
Fast Fourier transform (FFT) spectrum analyser	Is now being used increasingly to improve performance reduce costs in RF design, electronics manufacturing test, service, repair. With increasing use of wireless technology used in the electronic circuit design of electronic devices, improved performance from spectrum analyzers is growing in importance.

	As the name suggests the FFT spectrum analyzer is an item of RF test equipment that uses Fourier analysis and digital signal processing techniques to provide spectrum analysis.
Cross-Spectrum CPB (Constant Percentage Bandwidth) analysis	The program is intended for automated measurement of total harmonic distortion parameters and analysis of the form of the signals sent to the input channels of FFT Spectrum analyzers.
Gas turbine blades (SDS)	This is a separate component that makes up the turbine section of a gas turbine or steam turbine <159.>. The vanes are responsible for extracting energy from the high-temperature, high-pressure gas produced by the combustion chamber. Turbine blades are often the limiting element of gas turbines.
Gas turbine engines (GTE)	This is an engine that works by heating compressed gas, which in turn is fed to the shaft of a gas turbine. Compressed atmospheric air from the compressor enters the combustion chamber, then fuel is supplied there, which, when burned, forms a large amount of combustion products under high pressure. Then, in the gas turbine, the energy of the gaseous products of combustion is converted into mechanical work due to the rotation of the blades by the gas jet, part of which is spent on compressing the air in the compressor. The rest of the work is transferred to the driven unit. The work consumed by this unit is the useful work of the engine. Gas turbine engines have the highest specific power among internal combustion engines, up to 6 kW/kg. Anything that burns can be used as fuel, from gasoline to crushed coal.
Vibration generator system	The equipment including a Shaker with a power amplifier.
Imaginary reference point	A reference point to which a certain signal is assigned, produced by vibration signals from several verification points (with measuring channels in the Control status) and used to control the test mode (multipoint control) so that to meet the test requirements
Reproducibility	The proximity of measurement results of the same unidad with the same value carried out by different methods, with different primary converters (sensors), by different operators, in different testing laboratories, at different times, the interval between which is significantly longer than the time of one measurement.
Averaging time	The time interval for sampling instantaneous signal from the recorded stream to the instantaneous value array for further processing of the array.

Boundary points	The points used for building vibration test profiles for Sine and Random.
Effective vibration	Vibration characterized by signal from a sensor installed at the reference point.
Decibel (dB)	The unit of measurement of a physical quantity relative to the selected reference value, expressed as the logarithm lg (based on 10) of the ratio of the physical quantity value to the reference value. In the ZETLab vibration control system, the reference value is equal to one; therefore, for converting values in linear physical quantities "x" to dB, the formula is: dB = 20lg(x), and in the case of physical quantities with the power unidad "x2", the formula is: dB=10lg(x2).
Dynamic range of the measuring channel	It is defined as the ratio of the maximum level of recorded signals to the minimum recorded level. The theoretical limit for a 24-bit ADC is 140 dB, but the actual dynamic range is reduced due to interference and distortion in the system.
Dynamic range of the control signal	It is defined as the ratio of the maximum value of the signal generated on the control channel to its minimum value. For Sine mode, if the control signal changes from 1 mV to 10 V, the dynamic range is 10000 times = 80 dB. For Random (BRV) mode, the maximum value and minimum value of the control signal are measured by the power spectral density. The dynamic range of the vibrating system in whole is determined not only by the dynamic range of the VCS's DAC controller, but in any particular test may be limited to other factors, such as noise level at the Shaker table (recorded without control signal), a dynamic range of the vibratory installation, the maximum allowable vibration level in testing, etc.
Duration of the shock impulse	The time interval from the beginning to the end of the shock impulse which is a strong part of the accelerograms.
Q factor, and Quality factor	In physics and engineering, the quality factor or Q factor is a dimensionless parameter that describes how under-damped an oscillator or resonator is. It is defined as the ratio of the initial energy stored in the resonator to the energy lost in one radian of the cycle of oscillation. Q factor is alternatively defined as the ratio of a resonator's center frequency to its bandwidth when subject to an oscillating driving force. These two definitions give numerically similar, but not identical, results. Higher Q indicates a lower rate of energy loss and the oscillations die out more slowly. A pendulum suspended from a high-quality bearing, oscillating in air, has a high Q, while a pendulum immersed in oil has a

	low one. Resonators with high quality factors have low damping, so that they ring or vibrate longer.
Units	You can connect sensors to the VCS controllers inputs to record various physical quantities, such as acceleration (m/s^2, mm/s^2, g), displacement (m, mm, micron), velocity (m/s, mm/s), therefore, to obtain valid results, for the measurement channels set units which correspond to the types of sensors being connected. *Note: Units of measurement for primary converters (accelerometers) are listed in their respective datasheets.*
Strong part of the accelerogram	For a classic shock: a part of the accelerogram between two points in time, when the signal reaches 10% of the peak value for the first time and when it falls below this level for the last time. For a vibration shock: a part of the accelerogram between two points in time, when the signal reaches 25% of the peak value for the first time and when it falls below this level for the last time
Measuring channel (control/tracking/viewing)	The input channel (ADC channel) of the VCS controller with connected primary converter used for vibration tests. Measuring channels can be assigned the Control, Tracking, and Viewing status during vibration testing. The Control status determines that data from the measuring channel are used for generating a control signal, including an emergency stop of vibration tests upon exceeding the thresholds defined on the Profile tab of the test profile editor window. The tracking status indicates that measurement channel data are used to initiate an emergency stop of vibration tests when the thresholds defined on the Stop tab of the test profile editor window are exceeded. The Viewing status indicates that measurement channel data are not involved in the vibration tests control and are only used for visualization of the recorded signals.
Instrumental error	A set of errors introduced by both analog devices connected to the controller inputs and the VCS controller itself.
True spectral density of acceleration	The spectral density of acceleration affecting the specimen under test.
Feedback channel	The control system channel is used for: signal digitization at the reference point, signal processing and conversion of the processed signal into analog format to feed to the power amplifier of the Shaker.

Control channel	The VCS controller's oscillator channel used for generating the control signal.
Total harmonic distortion, or THD	The total harmonic distortion, or THD, of a signal is a measurement of the harmonic distortion present and is defined as the ratio of the RMS of all high harmonic components to the RMS of the fundamental frequency harmonica. The coefficient is dimensionless, but is usually multiplied by 100% to obtain a percentage value.
Emergency stop button	The button located to the right on the front panel of the VCS controller and intended for emergency stop (STOP mode) of transmitting control signal to the Shaker.
VCS controller	ZET 024 or ZET 028 model devices provide one output control channel (DAC) and, respectively, four or eight measurement channels (ADC).
Reference point	One of the verification points (with the measuring channel with Control status), the signal from which is used to control the test mode (single-point control) in a way to meet the test requirements.
Correction	Procedure for minimizing the error in play recorded signals the acceleration spectral density.
Maximum control voltage	Voltage threshold at the control channel (oscillator) output of the VCS controller.
Instantaneous value of the signal	The signal amplitude value registered for a single ADC count.
Multipoint control	Control by signals averaged by analog method or other suitable mean, recorded by measuring channels from vibration sensors installed at several verification points.
Observed acceleration spectral density	Visualized acceleration spectral density on the VCS monitor, including instrumental error, random error, and offset.
Normative technical documentation (NTD)	State standard, enterprise standard, specifications, technical descriptions, regulations and other documentation fixing the requirements for specimen quality.
Test Object	A specimen subjected to vibration tests.
Single-point control	Control by a signal recorded by the measuring channel from the vibration sensor installed at the reference point, to keep the specified vibration level at this point.

Cutoff of the drive signal	Limiting the maximum drive signal at the level determined by the peak factor value.
Primary converters	Sensors converting various physical quantities (acceleration, velocity, displacement, deformation, temperature, etc.) into an electrical signal proportional to the effect of the physical quantity.
Peak factor	The ratio of the peak value to RMS value of the signal.
Measure of inaccuracy in reproducing the acceleration spectral density	The difference between the specified acceleration spectral density and the acceleration spectral density of the control signal.
Transverse vibration	Vibration acting in a direction other than the specified direction (usually defined in two orthogonal axes in a plane perpendicular to the specified direction of movement. Please note that the transverse vibration must be measured close to the attachment points.
Preferred directions of vibration action	Three mutually orthogonal directions chosen to ensure the maximum probability of damage to the test object in case of vibration exposure in these directions.
Pred-Test	A program that provides registration of amplitude-frequency characteristics, as well as other parameters necessary for testing, on all available measuring channels of the SUV by registering response signals from the measuring channels to the impact generated by the control channel in the form of a noise signal in the established frequency range with a low level of spectral acceleration density. As a result, the program automatically generates recommendations and indicates possible mistakes made in preparation for the tests.
Verification point	The sensor installation points (with measuring channels with Tracking status) on the attachment device, vibration table or test object, located as close as possible to the attachment points of the test object (rigid connection) and used to monitor compliance with the test requirements.
Vibration test profile	Defines a profile required by the test conditions, which must be provided during vibration tests by generating a required signal through the control channel. For tests with Random and Sine, the profile is determined in the frequency domain, and for tests in shock mode - in the time domain.
The frequency resolution	The width of the frequency increment interval in the acceleration spectral density vie, Hz.

Recording	Processing a set of readings (recorded in measuring channels at regular intervals) using the fast Fourier transform algorithm.
Control mode (by one, by average, by maximum)	There are three control modes used as a basis for generating control signal: in "by one" mode, the control signal is generated based on data recorded in a single control channel. In the "by average" mode, the control signal is generated based on the average values recorded in a group of channels selected for control. In the "by maximum" mode, the control signal is generated based on the maximum values recorded in a group of channels selected for control. The "by average" mode and "by maximum" mode refer to multipoint control.
STOP mode	In this mode, the emergency stop button on the right pane of the front panel of the VCS controller is pressed.
RPM (Rounds Per Minutes)	is an instrument for measuring the speed of rotation on a shaft or on a disk, as in an engine or other machine.
Resonance	The frequency at which the response to the control signal (oscillator) increases sharply (very high). When examining the specimen's fatigue characteristics, exposure to resonant frequencies is used.
Results of the Pre-Test	A list of parameters saved based on the results of the Pre-Test and relevant until the next one or until the expiration of the time of day in which the Pre-Test was conducted.
Profile segment	A section of the vibration test profile bounded by adjacent frequency boundary points.
Control signal	Output voltage of the control channel (oscillator) of the VCS controller used to excite the Shaker.
Vibration control system (VCS)	Is a hardware and software system used for generation of the signals applied to the input of the shaker amplifier in compliance with the set test profile. The system is also used for recording of the response from the transducers installed at the moving part of the system and at a sample under test.
Signal attenuation rate	When stopping vibration tests, the control signal (oscillator) must attenuate smoothly, otherwise the test object may be subjected to shock. The control signal strength reduction can be selected from 20 dB/s to 60 dB/s.
Random inaccuracy	Estimation error the acceleration spectral density that varies from one measurement to another and is caused by finite time of signal averaging and the finite filter bandwidth.

High-frequency roll-off	A section of the acceleration spectral density at frequencies higher than the upper limit of the effective test frequency range.
Low-frequency roll-off	A section of the acceleration spectral density at frequencies lower than the lower limit of the effective test frequency range.
Acceleration spectral density (ASD)	Frequency function defined as the limiting ratio of the mean square value of the acceleration signal after it passes through a narrow-band filter whose geometric mean frequency coincides with the specified one, to the filter bandwidth as the bandwidth tends to zero and the averaging time to infinity.
The acceleration spectral density of the control signal	Acceleration spectral density of a signal measured at a reference point (real or imaginary).
Signal root-mean-square (RMS)	The square root of the sum of squares of instantaneous signal values recorded during averaging.
Standard deviation	The characteristic of a random time signal that is consistent with the RMS value for a vibration signal
Static degree of freedom	A value that characterizes the properties of estimating the acceleration spectral density obtained by random samples with time averaging method, and depends on the frequency resolution and time of averaging.
Small arms and cannon shock (SACS)	For testing specimens for resistance to random vibration in the mode SACS.
Steady-state accuracy	The ratio of true acceleration spectral density to the observed one.
Response measurement point	Sensor installation points (with measuring channels with Viewing status) on the test object, the signals from which are not involved in the vibration test control, but used only for examining its frequency response.
Attachment point	A part of the test object which is in contact with the attachment device or vibration table in the place where it is usually attached during operation.- If a device used during the operation of the test object is used for testing, the attachment point is determined on this device rather than on the test object.
Control by the maximum value	A method for determining the signal in multipoint control by selecting the maximum value of the controlled parameter for each frequency component at least in two verification points whose measuring channels are assigned the "Monitoring" status.

Control by average value	Method for determining the signal for multipoint control by averaging each frequency component at least in two verification points whose measuring channels are assigned the "Monitoring" status.
Acceleration	A vector value determining the degree of velocity change over time.
Acceleration of gravity	Acceleration of gravity is rounded to the closest integer, i.e. up to 10 m/s^2.
Averaging (linear/exponential)	The time interval during which instantaneous signal values are sampled from the recorded data stream to the array for further array processing. It is used to improve statistical accuracy or suppress interferences. In case of linear averaging, each data element contributes the same amount to the average value. Linear averaging is usually used for limited time intervals, since for large time intervals, the last added values actually no longer affect the resulting averaged value. In case of exponential averaging, each last averaged value has a greater weight than those involved in the averaging earlier, so it can be used at infinite intervals. The average value will dynamically reflect the influence of the new recorded values involved in averaging, and the influence of the previous ones will decrease as they age. The degree of exponential averaging is determined by a weighting factor calculated as reciprocal value of the number of averaging.
Phase frequency response (PFR)	The dependence of the phase difference between the output and input signals on the frequency of the signal. For a linear electrical circuit, the dependence of the phase shift between harmonic oscillations at the output and input of this circuit on the frequency of harmonic oscillations at the input. Frequency response is often used to evaluate phase distortions of the shape of a complex signal caused by an unequal delay in time of its individual harmonic components during their passage through the circuit.
CSM (Central Statistical Moment)	The moments of a random variable are some functions that numerically describe the characteristics of the distribution of a given random variable. There are initial, central, absolute, centrally absolute and factorial moments. The central moment corresponds to the variance of the distribution, it shows the spread around the average value (that is, around the mathematical expectation or the first moment).
Frequency	The number of vibrations or cycles per unit of time. Unit of measurement is Hz.

Sampling rate	The number of sample values of the signal per unit of time (per second) when the signal is represented digitally.
Sampling frequency (sampling)	In relation to the measuring channels, it refers to the number of analog-to-digital conversions per second for each recorded measuring channel, in relation to the control signal, it refers to the number of digital-to-analog conversions per second when generating the control signal. The ZETLAB programs processing a digital signal require a data array from a set of recorded instantaneous values of the processed signal amplitude accumulated during averaging, and the frequency of recording instantaneous values is determined by the sampling frequency. Thus, the higher the sampling frequency, the larger the array becomes at the same averaging time. The accuracy of the measurement results is directly related to whether the averaging time and sampling frequency are properly set. The best measurement results are achieved when providing the required level of detail without unnecessary redundancy. For VCS with max. 48 channels, the sample rates are set to 25 kHz for measuring channels and 50 kHz for control channels. For VCS with 49 to 160 channels, the sample rate values are set to 2.5 kHz for measuring channels and 5 kHz for control channels.
Resonance frequency	The frequency value typical for an object susceptible to vibration, at which the following is recorded: increase in vibration amplitude of the object and the difference between the vibration effect phase and oscillation phase of the object equal to 90 degrees.
Frequency range for testing	The range between the lower and upper limit in the frequency domain defined in the test profile.
Number of degrees of freedom	Indicates the number of independent variables used in calculating the average value. It is used in averaging for the Random control. Each averaging adds two degrees of freedom. The more degrees of freedom, the more accurately the spectral power density of the broadband signal is calculated.
The peak width at -3 dB	The bandwidth between two frequency response points located at 0.708 of its maximum value, assuming that the frequency response in this bandwidth describes a single resonance peak.
Random	The signal generated on the control channel (when testing Random) is noise randomly distributed over a wide range in the frequency range.
Stage of tests	A test program element occupying a line in the schedule table.

Effective test frequency range	The range between the lower and upper limit in the frequency domain defined in the test profile. Remember that beyond the effective frequency range, there are also signal components due to lack of sharp drop in the acceleration spectral density curve at the profile boundaries.

ZETLab

Contact information

LLC "Electronic technologies and metrological systems"

Please notify us using any communication of your convenience on any issues and faults occurring during ZETLab software installation and operation.

Manufacturer's address: 14 Konstruktora Lukina str. build.12, Zelenograd, 124460, Moscow, Russia.

GPS COORDINATES 56.008067, 37.153907

Telephone/fax: +7 (495) 739-39-19 (Multichannel)
Technical support: INFO@ZETLAB.COM for issues relating to purchase of standard products
Web site: ZETLAB@ZETLAB.COM information on Company products

OFFICE HOURS Mon-Fri: 9 a.m.–6 p.m (MSK time)

For issues relating to publications and advertising proposals, please, contact:

reklama@zetlab.com

Technical support

Should you have any questions regarding equipment selection, use and maintenance, you can contact us by E-mail or in the forum of our website. Our specialist will provide you with informational support.

In order to receive information concerning equipment operation from our specialists, you should prepare a list of the source data. Taking into consideration the source data volume, it is better to send it by E-mail. It is hardly reasonable to try to submit this information over the phone.

We need the following details:

• your name and contacts;

• name and serial number of the instrument;

• information about your PC (processor, memory, video card) and operating system;

• ZETLAB software version and configuration. Date of the previous software update;

• the program settings – sampling frequency, number of channels, amplification ratios, co-phase and differential channels;

• external connection scheme – text description, technical drawing, photo of the connected device or a schematic drawing;

• contact contact numbers, connections length, type of cable used: shielded, twisted pair;

• signal sources used: inner impedance levels;

• Evaluation of signal levels at the device input, signal type used (specific signal parameters (if any) – impulse, sine, random, periodical, frequency band width);

• Operating environment of the instrument (laboratory, manufacturing facility);

• Describe grounding chains of the PC, grounding of signal sources – if they are used, describe them;

• It is also necessary to describe the interference factors – cross-channel mixing or any other negative effects together with some quantitative characteristics! It is also desirable to attach several print-screens.

In the case if you provide our technical specialist with this source information, it will allow us to provide you with the necessary information as soon as possible!

Back Cover