RTCON SPECIALIZED SOFTWARE
USER GUIDE

Document version 5.1.10

R&D Company Vector LLC

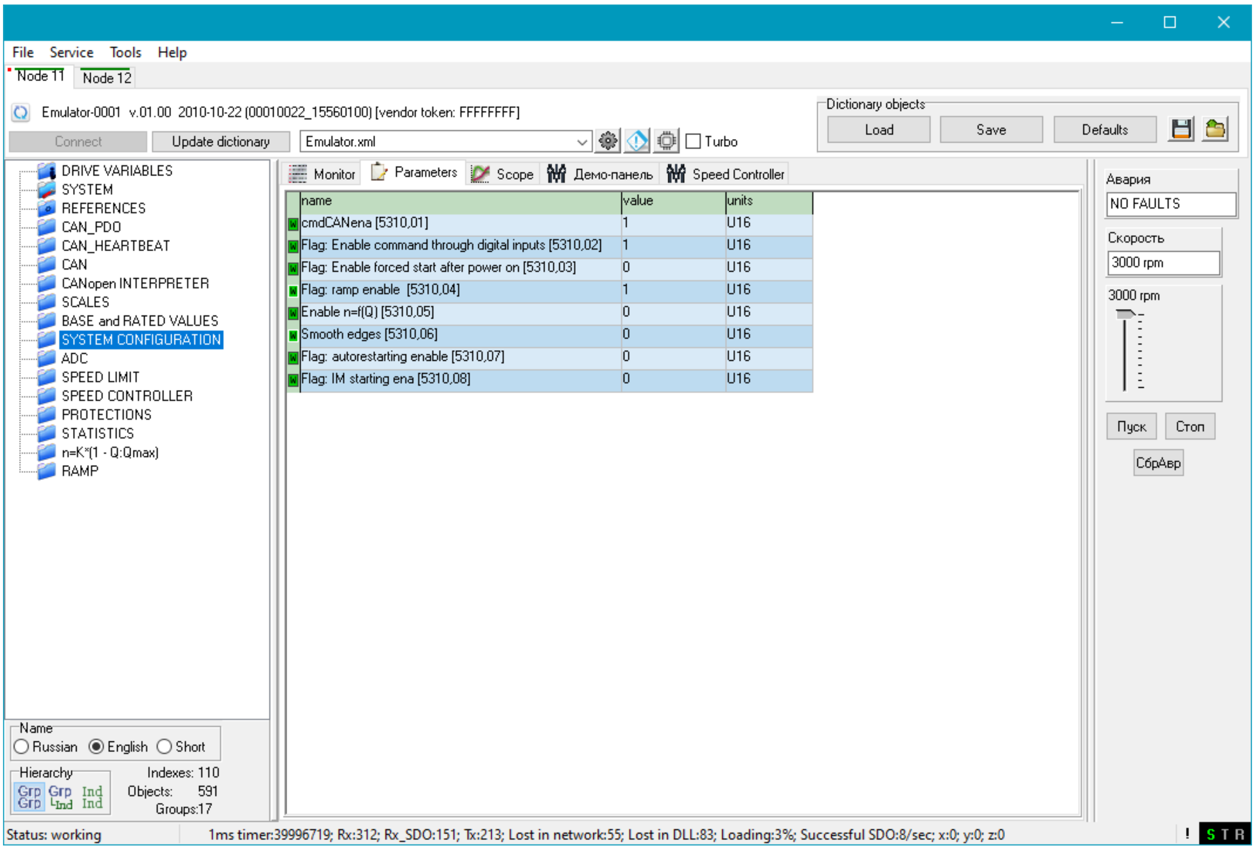
Moscow 2022

Contents

1. Introduction

Control interface RTCON is a set of software tools for monitoring and adjusting CANopen-compatible devices using a personal computer (PC) connected via the physical CAN interface.

This document contains a description of RTCON v.5.1.10 operation and the basic principles of its use.



Emulator-0001 v.01.00 2010-10-22 (00010022_15560100) [vendor token: FFFFFFFF]

Emulator.xml

name	value	units
cmdCANena [5310,01]	1	U16
Flag: Enable command through digital inputs [5310,02]	1	U16
Flag: Enable forced start after power on [5310,03]	0	U16
Flag: ramp enable [5310,04]	1	U16
Enable n=f(Q) [5310,05]	0	U16
Smooth edges [5310,06]	0	U16
Flag: autorestarting enable [5310,07]	0	U16
Flag: IM starting ena [5310,08]	0	U16

Авария: NO FAULTS

Скорость: 3000 rpm

Name: ○ Russian ● English ○ Short

Indexes: 110
Objects: 591
Groups:17

Status: working

1ms timer:39996719; Rx:312; Rx_SDO:151; Tx:213; Lost in network:55; Lost in DLL:83; Loading:3%; Successful SDO:8/sec; x:0; y:0; z:0

Fig. 1.1 General view of the RTCON main window when connected to the network device

2. Features

RTCON main functional features:

In operation conditions:

1. "On the fly" connecting to a running device without the need to stop (when the connection cables meet the requirements, USB/CAN adapter is able to operate, and the RTCON program settings are right).
2. "On the fly" operation without having to stop the device software operating in a real time mode.
3. Connecting to any available CANopen network node, if a group of devices connected to a local network. The number of devices in the network should be less than 127.
4. Automatic adjustment of RTCON parameters updating frequency based on the current traffic in CAN network to prevent network from overloading.

In parameters displaying and editing:

1. Displaying of the device parameter values (object dictionary) in a tree structure, easy to read.
2. The ability to switch the tree structure displaying mode:
   - parameters grouping by indexes;
   - parameters grouping by groups;
   - parameters grouping by groups and indexes together.
3. The ability to change parameter names displaying mode:
   - full names (long text);
   - short names (matching texts on converter control panel display);
   - English names.
4. Device parameters browsing and editing in real time mode.
5. Saving edited parameters to a device Non-Volatile RAM (NVRAM) / recovery of previous settings from device NVRAM (if modified parameters saving isn't needed) / restore default settings from the device flash memory.
6. Saving the values of any selected group of parameters to a file. Transfer a group of parameter values stored in the file to other devices – replication of the optimal settings to objects with the same operation conditions.
7. The ability to view more information about each parameter quickly. Displaying parameter in the different numerical systems (decimal, binary, hexadecimal). Displaying of minimum, maximum and default value, short text of the control panel display, etc.

In parameters monitoring (MONITOR function):

1. Forming of the required for serviceman list of observed variables (selection of observed variables) to save the list to a file.
2. Viewing the values of observed variables in real time.
3. Updating values of observed variables with the maximum possible frequency.

In graphic displaying of dynamic processes (SCOPE function):

1. Simultaneous recording and graphical displaying in function of time of up to 16 observed variables with an adjustable scale of the axes.
2. Ability to save graphics in "bmp" graphic file for later analysis and documentation.
3. Ability to saving the oscilloscope data to a file that is compatible with Microsoft® Excel®.
4. Automatic saving of real-time graph files in internal format, with the possibility of later viewing, analysis, and resave (bmp, Excel®) in the additional waveforms display software.

5. Ability of recording rapidly changing signals, including the tracking of the instantaneous values of the motor phase currents, voltages, etc. (the function is only available for devices supporting it):
   - the ability to synchronize the process of recording with various events, pre-defined in the CANopen device (start, stop, fault, etc.);
   - the ability to save prehistory when process recording is synchronized with some events (the feature is useful when recording emergency events, because user is able to track the processes that take place just before the fault);
   - the ability to select any parameter as trigger to automatically record waveforms when the value of parameter changes (this feature must be supported by the device software too).

In operational control capabilities:

1. Direct device control via RTCON using the operational control panel, which is a “virtual control panel” located on the right side of application main window.
2. The ability to create additional control panels as tabs. For these panels, you can set a background image.
3. The ability to design required “virtual control panel” with the control buttons (“Start” and “Stop” for example), operation mode selectors, frequency/process variables sliders directly in the RTCON.
4. Assigning controls names and binding controls to actions performed by them in the operational control window.
5. Current device status displaying on the virtual control panel.
6. Device real time clock current status displaying. Real time clock setting.
7. Device fault journal browsing. Working with device fault journal – browsing, clearing, saving information to a file on the hard disk.

In capabilities of adaptation to different device types:

1. For each device type (frequency converter, pumps group control station, etc.) serviceman makes a one-time setup for RTCON, then parameters are stored in the device profile file.
2. Automatic detection of the device type and its software version, and application of an appropriate settings profile (if it exists).

3. Minimum requirements

The equipment needed for RTCON operation:

1. A computer with operating system “Windows XP” or higher, which has a free USB port.
2. USB/CAN interface converter with galvanic isolation (adapter) – a standard device, supplied by a number of companies, for example, USB-CANmodul by SYS TEC Electronic or CAN-bus-USB by “MARATHON” (the list of supported devices, see below).
3. A cable to connect the adapter to converter CAN-bus connector.

4. Supported adapters

This version of RTCON supports following USB/CAN adapters:

- USB-CANmodul. SYS TEC Electronic. www.systec-electronic.com.
- USB-CANmodul1. SYS TEC Electronic. www.systec-electronic.com.
- CAN-bus-USB interface. MARATHON. www.marathon.ru.
- ZigBee – wireless interface. For information, please contact RTCON vendor.
- RS – it is permissible to connect to the device via a serial port or USB-RS adapter if the target device includes the appropriate driver.
- Emulator. You can choose the emulator (EMULATOR) as a hardware device in the communication module settings. In this case, RTCON will create two virtual devices communicating via CANopen. This option is needed to explore the RTCON without connecting to a real device. The emulator contains a set of parameters that are similar to one of the real frequency converters. However, the emulator does not perform the reaction to changes in parameters. The exceptions are some service settings required for operation of the scope, faults bank, etc. In the ADC group of parameters emulator simulates measurement of phase currents, which can be observed in the RTCON scope.

5. Setup

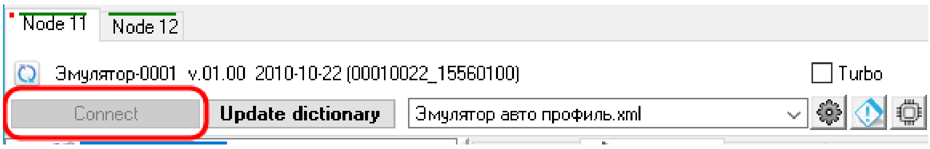
To install RTCON, follow these steps:

1. Copy RTCON folder with program files from a CD or flash drive into the work directory of the computer (any convenient directory).
2. Install the driver for your USB/CAN adapter (see list of supported devices in para. 4), following the instructions of the adapter manufacturer.
3. Connect the USB/CAN adapter to the computer.
4. Start RTCON program (RTCON.exe file) from the work folder on your computer’s hard drive.
5. If your computer has a firewall, a question about RTCON program access to the network may occur. RTCON provides a network interface (TCP server) to transfer CAN network traffic to third-party programs if necessary. If this feature is not required, you can deny access to the network and disable TCP server function in RTCON settings.
6. In the menu, select ***Service/Communication module parameters***. Select the adapter is used from the ***Communication module*** drop-down list.
7. Click ***Ok*** and restart the program.

Installation is complete.

6. Work methods

6.1. Connecting to a device



Эмулятор-0001 v.01.00 2010-10-22 (00010022_15560100)

Эмулятор авто профиль.xml

Fig. 6.1 Connecting to present network devices

After starting the program, after some time required for the detection of devices (a few seconds) in the upper part of the window appear tabs with nodes available for connection (see note). Below is written the name of the device that is associated in profile settings to the device type and software version (SW). Version and creation date (day.month.year) of software are also dispalyed. Inside the parenthesis are displayed: "Product Code" (parameter 1018.02) and "Revision Number" (parameter 1018.03) in hexadecimal format.

For connection you need to click the ***Connect*** button. First time you connect the dictionary loading will start. It will be displayed by the download process progress bar. When the dictionary is downloaded, you can begin working with the device. The downloaded dictionary will be automatically saved on your computer's hard disk, so on subsequent connections to this or a similar device connection will be fast and the dictionary download will not be initiated. If the device object dictionary has been changed, it is necessary to update the device dictionary file by clicking the ***Update dictionary*** button.

Note

If the program does not detect the device, check the network settings, make sure the USB/CAN adapter is connected properly, and the correct driver from the manufacturer is installed for it.

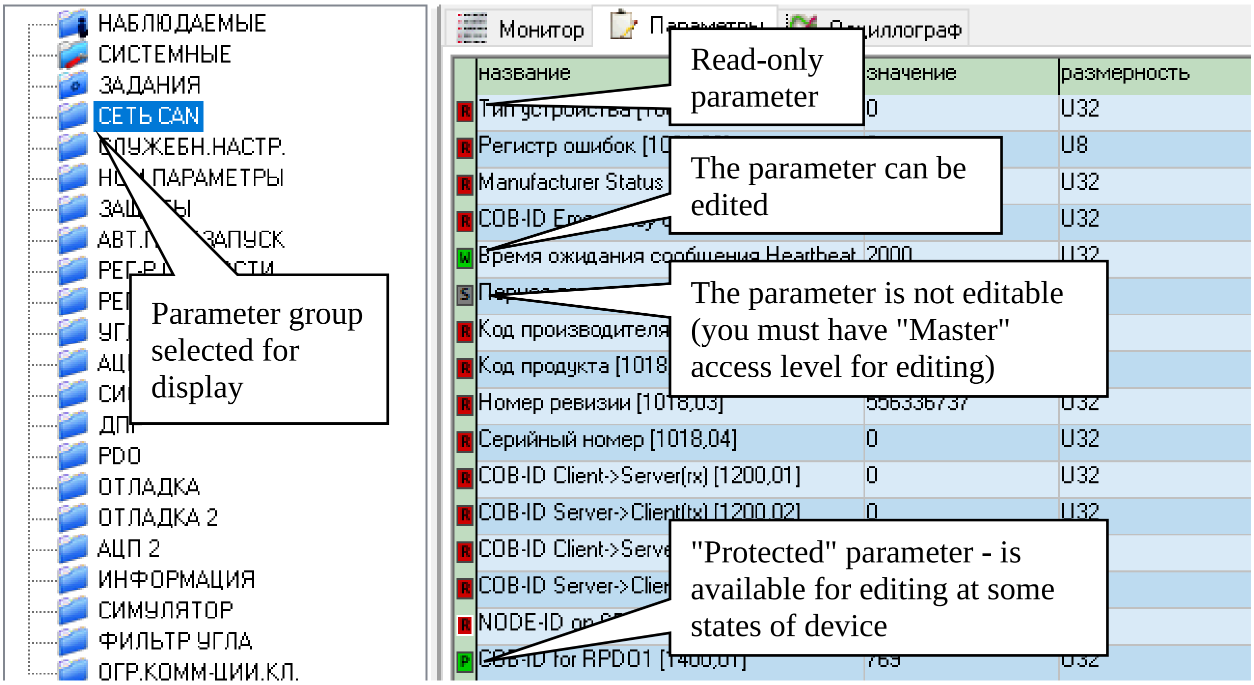
6.2. Parameters editing

To change the value of any element of the dictionary, select a specific section in the tree on the left, and then click the value of the variable – see Fig. 6.2. You can write a new value in the field, and then press the Enter key – after this the new value will be immediately sent to the device.

Availability for editing is shown as a colored box in the list, to the left of the parameter name:

R	Read-only parameter
W	The parameter is available for editing at the access level «Service» or «Master»*
P	«Protected» parameter – is available for editing only in certain modes of operation of the device at the access level «Service» or «Master»*
S	«Secret» parameter – is available for editing at the access level «Master»*
W S P	The parameter is not editable due to low access level*

* Note: see section 7.3 "Access level" below.



Read-only parameter

The parameter can be edited

The parameter is not editable (you must have "Master" access level for editing)

"Protected" parameter - is available for editing at some states of device

Parameter group selected for display

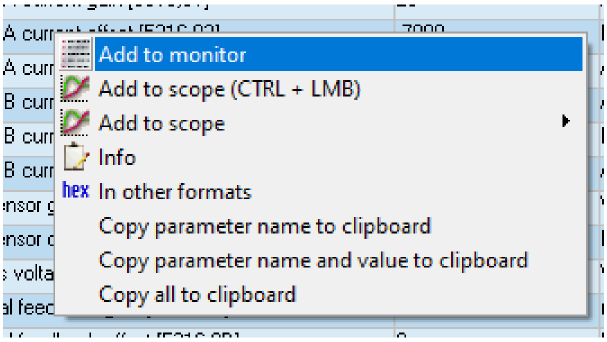
Fig. 6.2 Parameter editing tab

Group names, parameter names and associated enums (like «Yes»/«No») are defined by the unique identifiers that the RTCON application loads from the device when updating the object dictionary. The texts themselves are stored not in the device, but in the file «coTextBase.xml» in the root folder of the application. This file is generated by the application-editor of the dictionary COODEdit. If the RTCON application is connected to a device whose dictionary has a non-zero ***vendor token*** at address 2600h.01h (a parameter of type UINT32, for example, with the value 0xD467395A), then the text file «coTextBase_D467395A.xml» will be used (not «coTextBase.xml»). This functionality can be useful if there are devices from different manufacturers in the CANopen network (these devices will have completely different texts).

6.3. Monitor

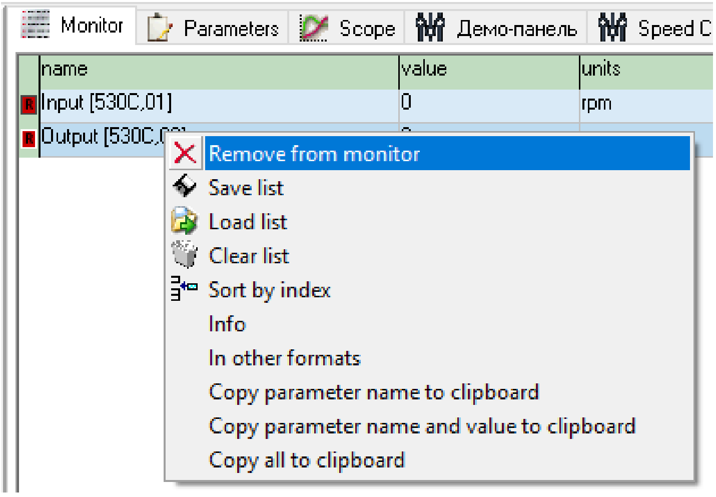
Monitor tab allows you to group interesting object dictionary variables from different groups in one window.

To add a variable to the monitor right-click on the interesting variable in the dictionary and select ***Add to monitor***.


Add to monitor
Add to scope (CTRL + LMB)
Add to scope
Info
hex In other formats
Copy parameter name to clipboard
Copy parameter name and value to clipboard
Copy all to clipboard

Removing of the variable can be performed in monitor tab via the context menu.

The variables in the monitor are arranged in a list that can be saved to a file and downloaded from file. There are corresponding context menu items for this purpose – see Fig. 6.3



name	value	units
Input [530C.01]	0	rpm
Output [530C.0		

Fig. 6.3 Context menu operations available in "Monitor" tab

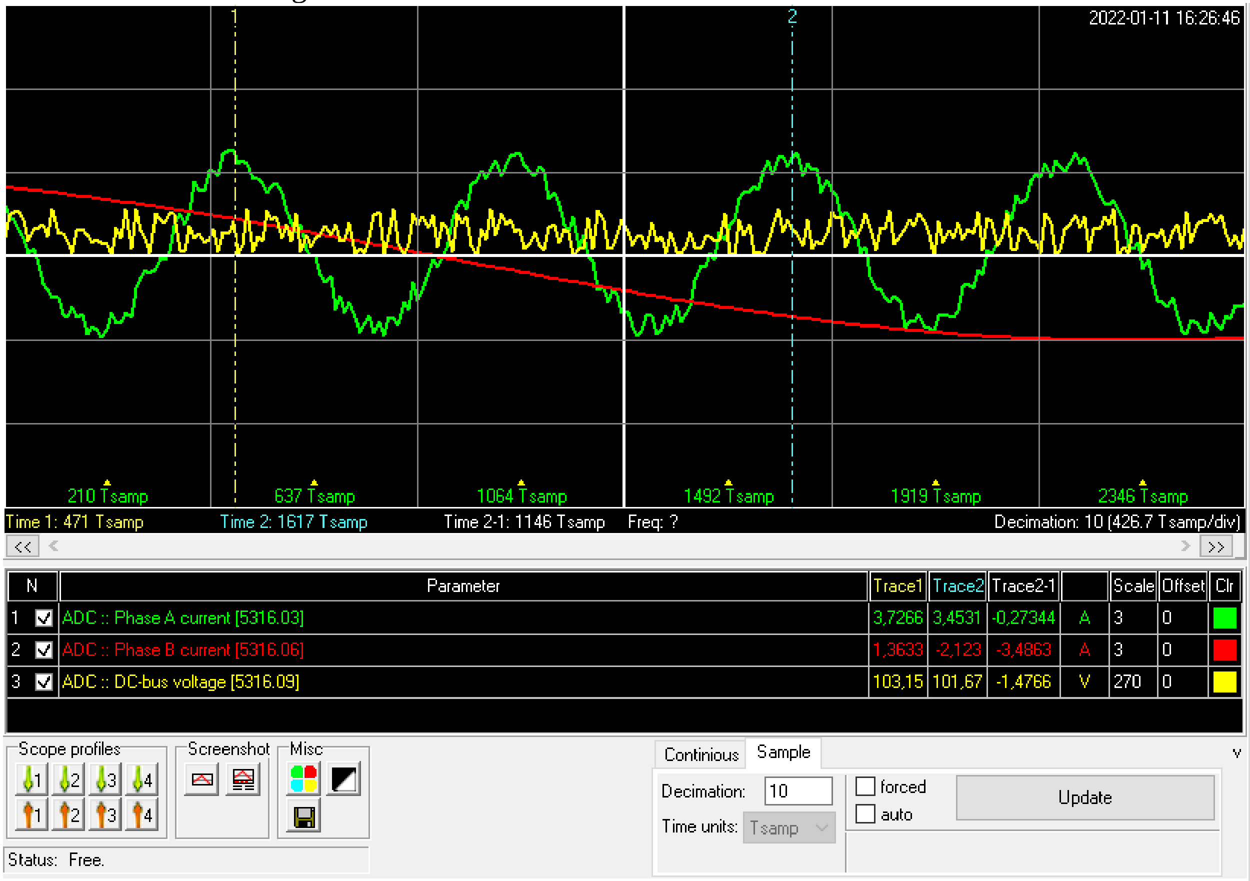
❗ **Attention.**

When saving a file only the list of variables is saved, without their values. Thus, no changes of variables in the device are going to happen when downloading a list. If you want to make changes, it must be done manually, by clicking the left mouse button on the interesting value, setting a new value and pressing Enter key. To transfer the values of the device parameters use ***Parameters transfer***.

6.4. Scope

6.4.1. General

This program module is intended for monitoring the transient processes over time. Scope tab exterior is shown in Fig. 6.4.



Time 1: 471 Tsamp   Time 2: 1617 Tsamp   Time 2-1: 1146 Tsamp   Freq: ?   Decimation: 10 (426.7 Tsamp/div)

N		Parameter	Trace1	Trace2	Trace2-1		Scale	Offset	Clr
1	☑	ADC :: Phase A current [5316.03]	3,7266	3,4531	-0,27344	A	3	0	🟩
2	☑	ADC :: Phase B current [5316.06]	1,3633	-2,123	-3,4863	A	3	0	🟥
3	☑	ADC :: DC-bus voltage [5316.09]	103,15	101,67	-1,4766	V	270	0	🟨

Decimation: 10

Time units: Tsamp

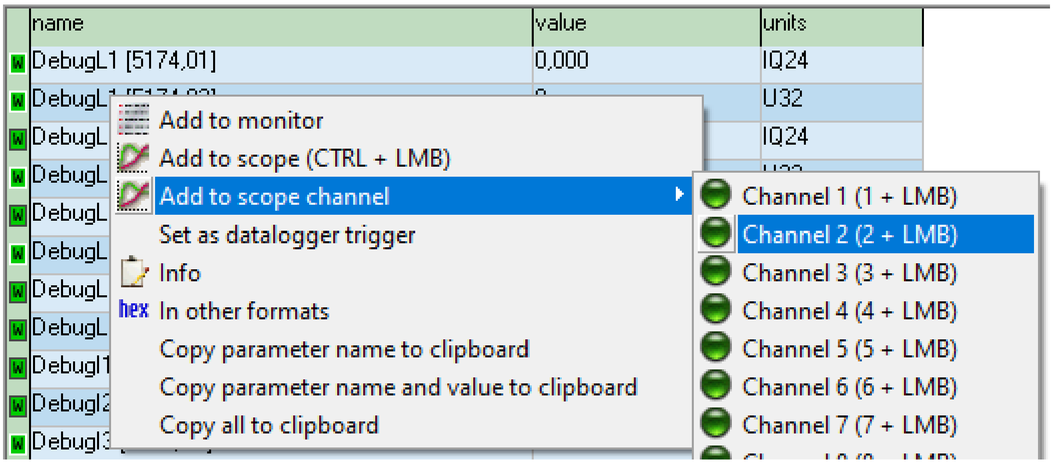
□ forced

□ auto

Status: Free.

Fig. 6.4 Scope window

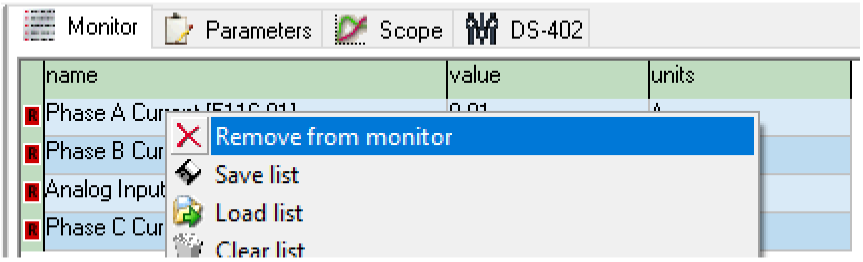
There are up to 16 measurement channels. You can add any variable from the object dictionary to each channel. To do this, right-click on the interesting variable in the dictionary and select either the ***Add to scope*** command (variable will be added to the next free channel), or the ***Add to scope channel*** **→** ***Channel N*** command (variable will be added to the selected channel N). In this case variable is automatically added to the monitor because these two modules are internally linked.



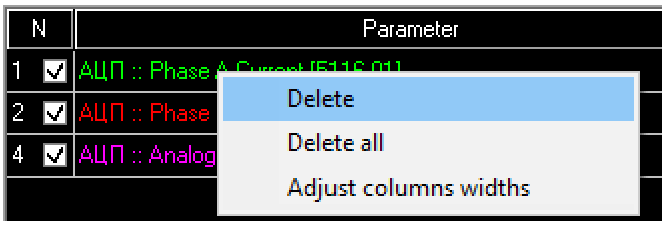
From context menu you can also set any variable as a trigger to automatic record waveforms when the value of variable changes (the device software must support this function, see section 6.4.3.2 “How to record an oscillogram on value change using trigger”).

There is a “quick” add mode: if at the moment of pressing the left mouse button on the parameter name the key 1 ... 9, A ... F or 0 is also pressed, then this variable will be added to the scope channel 1 ... 9, 10 ... 15 or 16, respectively.

A variable from the scope can be deleted through the context menu either on the ***Monitor*** tab:

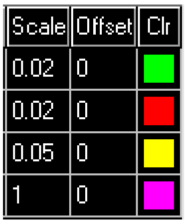


or directly on the ***Scope*** tab:



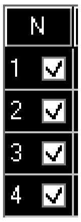
There are two fundamentally different modes: ***Continuous*** and ***Sample*** mode. To switch modes and adjust each mode two tabs are provided at the bottom right of the window (see. Fig. 6.4).

You can change the graph scale (you may set negative value), zero offset and the color individually for each channel. For these purposes, the corresponding cells in the table are used (the editing mode is activated by clicking the left mouse button on the cell):



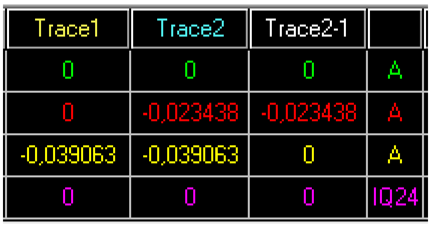
Scale	Offset	Clr
0.02	0	🟩
0.02	0	🟥
0.05	0	🟨
1	0	🟪

To enable/disable the display of the graph, there are checkboxes in the first column of the table:



N	
1	☑
2	☑
3	☑
4	☑

To monitor specific values of variables on graphs, in addition to the grid, you can use the cursors. To do this, click on the graphs area, and the corresponding values will appear in the table:



Trace1	Trace2	Trace2-1	
0	0	0	A
0	-0,023438	-0,023438	A
-0,039063	-0,039063	0	A
0	0	0	IQ24

You can use two cursors: one for the right and one for the left mouse button. If the device supports natural time units (see section 6.4.3 «Sample operation mode») then the frequency corresponding to the time difference between cursors will be displayed at the bottom of the window with graphs.

Button brings up a dialog where you can export currently displayed waveform to a file. You can save waveforms either in graphic formats BMP/PNG, or as an array of points in csv format, which is supported by a variety of office software packages such as Microsoft® Excel and OpenOffice.org Calc. When the scope is in ***Sample*** mode, the suggested filename is the time of ***Update*** button click or the time the trigger was fired (see section 6.4.3.2).

Buttons and are used to quickly save and restore all the settings of the scope channels. You can save up to 4 profiles.

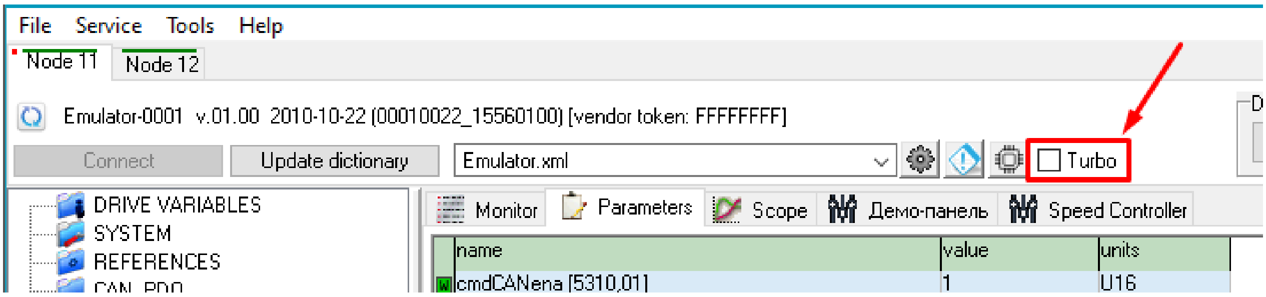
Button automatically assigns default colors to all channels.

Button inverts the colors of the entire scope window.

Buttons make “screenshot” of scope window with different details. The screenshot will be copied to clipboard.

6.4.2. Continuous operation mode

The mode is intended to observe slowly changing variables, such as pressure or frequency for a long time (minutes, hours or even days) in real time. In this mode variables polling is carried out approximately four times per second, the data is automatically written to a file and displayed on the screen. It is possible to scroll written process along the time axis using the scroll bar, and change the scale on the time axis on ***Continuous*** mode tab. At the bottom of the graph window the absolute time in hours:minutes:seconds format is displayed. A maximum interval of one day can be displayed on the time axis – at 00:00 a new file with a new date will automatically open, and recording will be continued to the new file. It is possible to accelerate the variables polling to the maximum possible frequency by activating the “turbo” mode by checking the appropriate checkbox:



Emulator-0001 v.01.00 2010-10-22 (00010022_15560100) [vendor token: FFFFFFFF]

Emulator.xml

□ Turbo

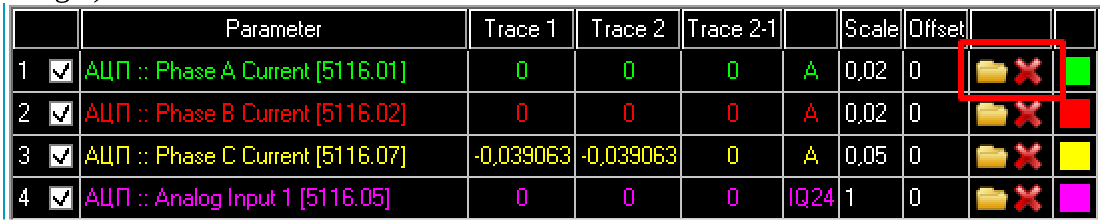
name	value	units
cmdCANena [5310,01]	1	U16

In this case, the frequency of data polling in a continuous scope mode rises to 10-20 Hz, but all the other features of the program may not work properly (changing a variable can possibly not work for the first time, the dictionary update will be decelerated and so on). It is recommended to disable this option, as soon as the waveforms viewing with a maximum sampling frequency is no longer required. For even higher polling frequency the sample mode should be used as described below.

You may use mouse wheel to horizontal scroll the scope window (press «Shift» key to increase the speed of scrolling).

All waveforms in a continuous scope mode are recorded to the binary files at **RTCON\scope\date\variable_name.ocf** file and can be re-opened using a special viewer, which is available via the **Tools → Viewer of ocf-files with oscillograms** menu.

This viewer is also available as an executable file "UniOSViewer.exe" inside the folder of RTCON application. The interface of this program is generally similar to the oscilloscope described in this document. Opening/closing files is carried out using the buttons (marked with red rectangle):



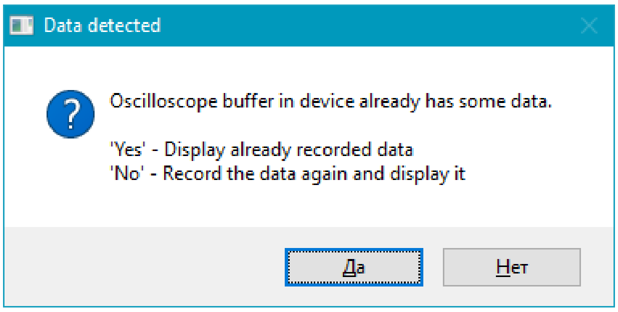
		Parameter	Trace 1	Trace 2	Trace 2-1		Scale	Offset		
1	☑	АЦП :: Phase A Current [5116.01]	0	0	0	A	0,02	0	📂 ✖	🟩
2	☑	АЦП :: Phase B Current [5116.02]	0	0	0	A	0,02	0	📂 ✖	🟥
3	☑	АЦП :: Phase C Current [5116.07]	-0,039063	-0,039063	0	A	0,05	0	📂 ✖	🟨
4	☑	АЦП :: Analog Input 1 [5116.05]	0	0	0	IQ24	1	0	📂 ✖	🟪

Viewer also has its own file browser.

6.4.3. Sample operation mode

This mode can be activated by clicking on the appropriate tab under the condition that the variables addresses are properly defined in device profile settings to work in this mode. Good for displaying fast processes, such as phase currents, various transients, etc.

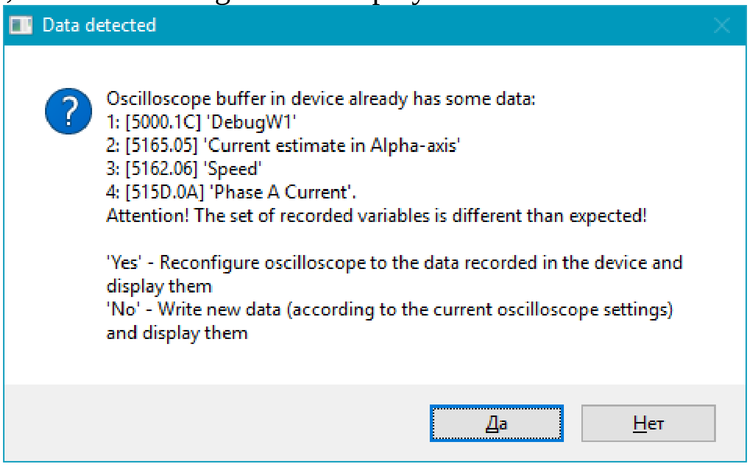
Understanding of the principles of its operation is desirable for a more successful work with this scope mode. This section describes specific software implementation, and if there is no special interest in the work subtleties, you can skip it. ***Sample*** mode is called so because the scope screen displays only some part of the whole process, depending on the length of the array to write data in a particular device. It is clear that the transmission rate of the CAN interface is not enough to transmit fast processes data in real time. Therefore, the target device must use the data buffer (actually an array of points) to record the entire process before the buffer is transmitted to the PC for displaying on the scope. There are up to 16 buffers (channels) of 256 points each (the specific number of channels is determined by the software of the target device). Recording of points can be made no more often than every period of the control system sampling (due to controller calculating speed limitations). Usually this frequency is 10-80 kHz depending on software version. For less frequent points recording there is a ***Divider***, which indicates how many times will be period of points recording increased relative to the sampling period. Divider possible value is limited by 255 – at higher values use of this mode is no longer appropriate, and you can go to the ***Continuous*** mode. At 10 kHz sampling frequency and with the ***Divider*** close to the maximum, the time of the process recording is already significant (6-7 seconds), which obliges the scope to wait for the completion of a process recording, and only then begin to download the data. In order to monitor the data load status, the field ***Status*** is made. It displays the current state of the scope, allowing the user to assess the situation more correctly. Often it is necessary to synchronize the start of the recording process with some event, such as motor start or stop, to watch the transient process. It is quite difficult to press the ***Update*** button manually at the right time, therefore, an automatic data writing start is provided inside the target device. In other words, the control system programmer should foresee the most important events itself and paste buffer recording start trigger there. As a rule, this should be done at least for start and stop of the motor, as well as for some of the emergency conditions. Then, when you click on the ***Update*** button, the program will show a message that the data has already been recorded, and will offer to download it. And the channels and the variables that have been set at the last time will be recorded. Thus, if you want to view, for example, the transition process of starting the motor, you need to set the desired variables in scope, switch it to the sample operating mode, click the ***Update*** button (the controller will set the necessary variables, divider and so on.), and then start the motor. Now, if you re-click ***Update*** button, the program should offer to download the recorded data, and you can agree this:


Data detected

Oscilloscope buffer in device already has some data.

'Yes' - Display already recorded data
'No' - Record the data again and display it

If the set of variables recorded in the device differs from the expected set by the current oscilloscope settings, another warning will be displayed:


Data detected

Oscilloscope buffer in device already has some data:
1: [5000.1C] 'DebugW1'
2: [5165.05] 'Current estimate in Alpha-axis'
3: [5162.06] 'Speed'
4: [515D.0A] 'Phase A Current'.
Attention! The set of recorded variables is different than expected!

'Yes' - Reconfigure oscilloscope to the data recorded in the device and display them
'No' - Write new data (according to the current oscilloscope settings) and display them

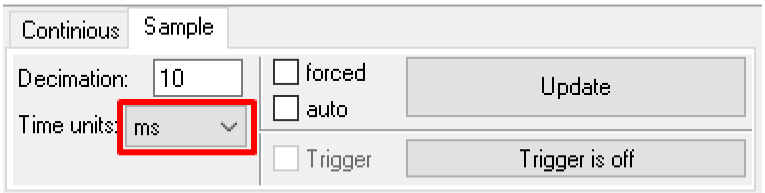
Divider allows you to specify how many times you need to increase the scale of the time axis. For more detailed description see above.

Update button sends a request to the device about necessity of new data write, after that the data is downloaded and displayed as a graph. You can also press the F5 key on your keyboard to refresh the data.

Auto update – after the end of the current recorded process data load, a request for record of a new data will be sent, and the cycle will be repeated.

Forced – regardless of whether there is already a data recorded in the controller system or not, pressing the ***Update*** button will start the recording of a new waveform without any additional confirmations.

If there is an object of type UINT32 in the device dictionary that contains the sampling rate in Hz, then on the «Sample» tab it is possible to select the time units:


Decimation: 10
Time units: ms
□ forced
□ auto
□ Trigger

The address of «sampling rate» object must be defined in the device profile settings (see section 7.4.2 below).

If the corresponding option is enabled in the program settings (see section 7.1), then after clicking the ***Update*** button, the time of data loading from the device is displayed in the upper right corner of the oscillograms window. If the data in the device's buffer appeared after the trigger was fired (see section 6.4.3.2), then after clicking the ***Update*** button, the trigger firing time will be displayed.

6.4.3.1. How to record an oscillogram corresponding to a certain event

Here we will consider the issue how to launch the data logger to record oscillograms in the "sampling" mode for certain events.

As stated above, in the "sampling" mode you can see the contents of data buffers that are located in the memory of the device controller (1-16 channels, 256 points per channel).

To start writing data to these buffers, you need to run a so-called "data logger" – a special software module in the controller that operates at a high frequency (usually in ADC interrupt service routine), which captures the values of the desired variables and saves them to the buffer with the specified decimation. Data logger can be launched in two ways:

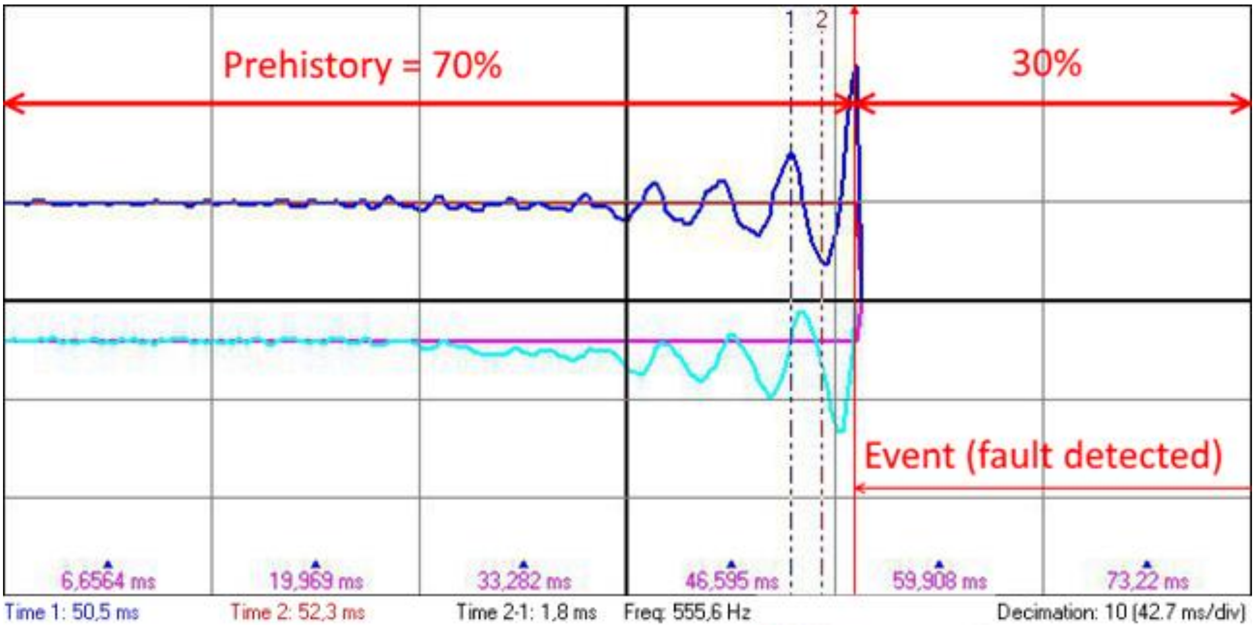
- Manually
  By pressing the «Update» button in the scope window of RTCON.
- Automatically
  In this case, the controller software implements the logic of automatic start of the data logger upon certain events in the control system.

The set of events for which automatic launch of the data logger is possible is "hardcoded" in the controller firmware and is determined by the state of the bit flags of two parameters in the **'Data Logger'** group: **StartBitL [5150.07]** and **StartBitH [5150.08]**. For example, "hardcoded" events may be such as:

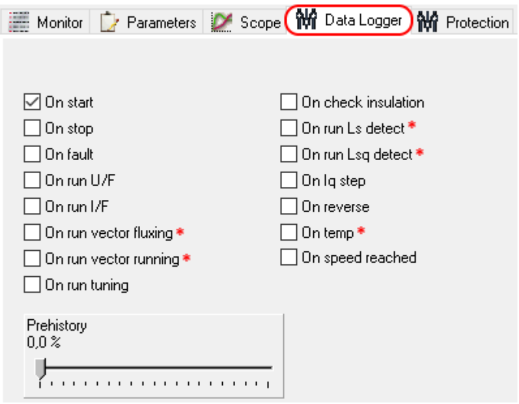
StartBitL [5150.07]	StartBitH [5150.08]
bit 0: on start command bit 1: on stop command bit 2: on fault detected bit 3: on enter to U/F state bit 4: on enter to I/F state bit 5: reserved bit 6: reserved bit 7: on enter to controller tuning state	bit 0: on enter to insulation check state bit 1: reserved bit 2: reserved bit 3: on Iq current becomes greater than 20% of maximum FOC current bit 4: on reverse command bit 5: reserved bit 6: on speed reached the reference value minus 30 rpm bit 7: reserved

In this case, for example, if you set bit #2 of StartBitL to «one» (00000100b) then the data logger will start on any fault detected in the control system.

Using the **'Data Logger'** → **'Prehistory [5150.09]'** parameter (in different devices, the address of this parameter may differ) you can also set the "depth" of the data prehistory, i.e. the number (in percentage terms) of data points before the event:



For the convenience of the user, the settings of the data logger can be placed on a separate control panel. For table of **StartBitL [5150.07]** and **StartBitH [5150.08]** bit association, given above, this control panel may look like this:



☑ On start
□ On stop
□ On fault
□ On run U/F
□ On run I/F
□ On run vector fluxing *
□ On run vector running *
□ On run tuning

□ On check insulation
□ On run Ls detect *
□ On run Lsq detect *
□ On Iq step
□ On reverse
□ On temp *
□ On speed reached

Prehistory
0,0 %

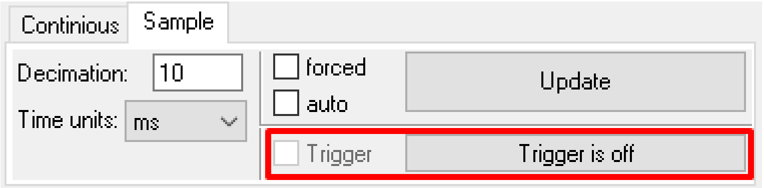
How such additional panels are created is described in the section 7.4.3.

The waveforms may also be automatically recorded when the value of some parameter-trigger changes. This function is described in the next section.

6.4.3.2. How to record an oscillogram on value change using trigger

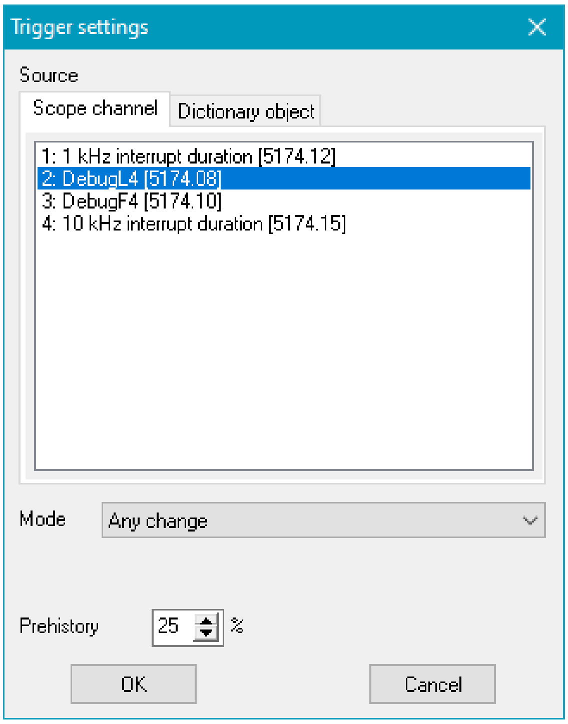
The *trigger* function implements the automatic record of waveforms when the value of some parameter (trigger-variable) changes.

The *trigger* function is only available if the device software implements corresponding trigger logic and the device profile configured properly (see section 7.4.2 “Profile main options setting”). In this case the trigger settings button and the trigger activation checkbox will be shown on the *Sample* tab:



Decimation: 10
Time units: ms
□ forced
□ auto
□ Trigger

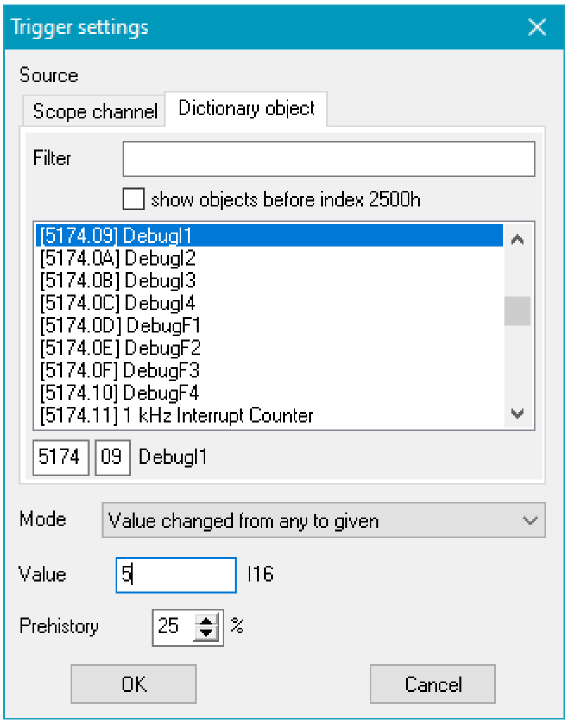
If you press the trigger settings button the window with two tabs appears:



1: 1 kHz interrupt duration [5174.12]
2: DebugL4 [5174.08]
3: DebugF4 [5174.10]
4: 10 kHz interrupt duration [5174.15]

Mode: Any change

Prehistory: 25 %



Filter:

□ show objects before index 2500h

[5174.09] DebugI1
[5174.0A] DebugI2
[5174.0B] DebugI3
[5174.0C] DebugI4
[5174.0D] DebugF1
[5174.0E] DebugF2
[5174.0F] DebugF3
[5174.10] DebugF4
[5174.11] 1 kHz Interrupt Counter

5174 09 DebugI1

Mode: Value changed from any to given

Value: 5 I16

Prehistory: 25 %

If the “Scope channel” tab is active (left figure) then you can select the scope channel as source variable for trigger.

If the “Dictionary object” tab is active (right figure) then you can select any dictionary object as source variable for trigger. You can select an object either by double-clicking in the list of objects, or by entering its index/sub-index in the appropriate text boxes below the list. You can also filter the objects in the list by name/address entering the text in the “Filter” text box. By default the service objects before index 2500h is not included in the list. If you want to see them you should check the corresponding checkbox.

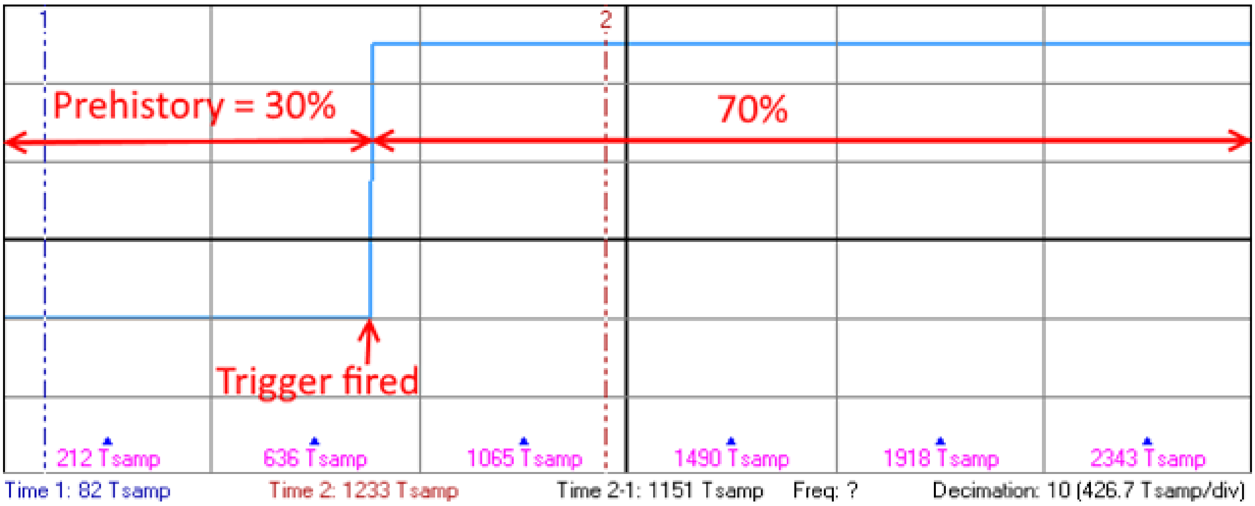
The modes of trigger function are listed in the following table:

Mode	Description (condition at which the trigger will fire)
Trigger off	
Any change	The value of trigger-variable has changed
Rising	The value of trigger-variable has increased
Falling	The value of trigger-variable has decreased
Value changed from any to given	The value of trigger-variable has changed from any to the given. Example: from -10 to 5, or from 15 to 5. (hereinafter, we will consider “5” as the given value/threshold)
Value changed from given to any	The value of trigger-variable has changed from the given to any. Example: from 5 to -10, or from 5 to 15.
Rising above value	The value of trigger-variable has increased above the given threshold (>). Example: the value has increased from -10 to 6.

Mode	Description (condition at which the trigger will fire)
Rising to value	The value of trigger-variable has increased to the given (but not above). Example: from -10 to 5.
Rising from value	The value of trigger-variable has increased from the given. Example: from 5 to 15.
Falling below value	The value of trigger-variable has decreased below the given threshold (<). Example: the value has decreased from 15 to 4.
Falling to value	The value of trigger-variable has decreased to the given (but not below). Example: from 15 to 5.
Falling from value	The value of trigger-variable has decreased from the given. Example: from 5 to -10.
Leaving range	The value of trigger-variable has left the given range (minimum…maximum)
Entering range	The value of trigger-variable has entered the given range [minimum…maximum]

Depending on the selected mode the appropriate text boxes will be shown to enter the given value or range limits.

In the "Prehistory" field you can also set the "depth" of the waveforms prehistory, i.e. the number of data points (in percentage terms) before the trigger is fired. For example:



Pressing "OK: button saves the trigger settings for the current session of the application (until you close the RTCON). In this case, the trigger source is determined by which of the two tabs was active at the time the "OK" button was pressed. After closing the window, the trigger settings button displays the address and the name of the source variable. To save the trigger settings to the file you should use buttons 1/2/3/4 which save scope settings (see section 6.4.1 "General").

To activate a trigger, check the checkbox to the left of the trigger settings button. In case of successful activation, the status will be changed to "Trigger is activated". In this state, you can update the waveform via the "Update" button. When the trigger is fired, the status will be changed to "Trigger has fired" and the text on the trigger settings button will blink. To download the recorded waveform, click the "Update" button and when the answer appears, press "Yes" button. After clicking the “Update” button, the trigger firing time will be displayed in the upper right corner of the oscillograms window.

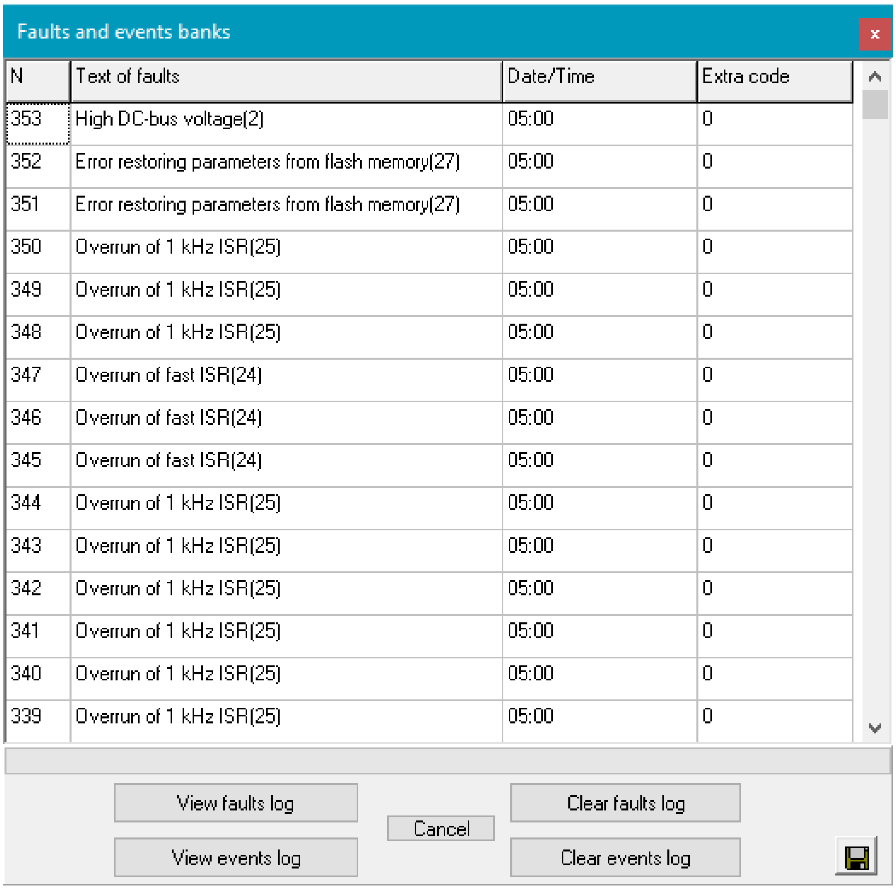
When the trigger is activating or fired, an error may occur. In this case, the status will be changed to “Trigger fault”. If you hover the mouse over this status text, a pop-up window will appear with a description of the error:

Trigger error description	Cause
Error during the activation of trigger	a) The dictionary has no objects to configure the trigger. b) Data transmission errors over the network (for example, due to a high level of interference).
Trigger has fired but the data wasn't recorded	When the trigger fired, the datalogger was not in the loop recording mode (so called mode #2).

To reset the trigger error you should press the “Update” button and then activate the trigger again.

6.5. Faults bank, events bank

Faults and events banks (logs) allow you to view a list of the last fifty occurred faults or events. It is called from the menu ***Service* → *Faults and events banks***. The window is shown in Fig. 6.5.

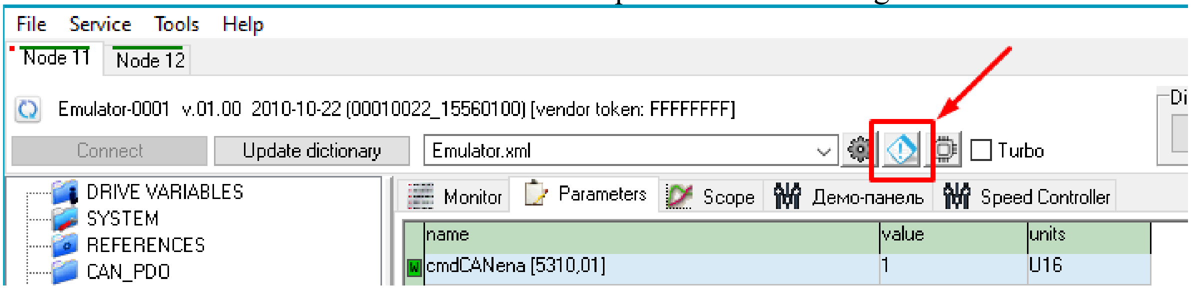


N	Text of faults	Date/Time	Extra code
353	High DC-bus voltage(2)	05:00	0
352	Error restoring parameters from flash memory(27)	05:00	0
351	Error restoring parameters from flash memory(27)	05:00	0
350	Overrun of 1 kHz ISR(25)	05:00	0
349	Overrun of 1 kHz ISR(25)	05:00	0
348	Overrun of 1 kHz ISR(25)	05:00	0
347	Overrun of fast ISR(24)	05:00	0
346	Overrun of fast ISR(24)	05:00	0
345	Overrun of fast ISR(24)	05:00	0
344	Overrun of 1 kHz ISR(25)	05:00	0
343	Overrun of 1 kHz ISR(25)	05:00	0
342	Overrun of 1 kHz ISR(25)	05:00	0
341	Overrun of 1 kHz ISR(25)	05:00	0
340	Overrun of 1 kHz ISR(25)	05:00	0
339	Overrun of 1 kHz ISR(25)	05:00	0

Fig. 6.5 Fault and event log window

To download the latest faults list click button ***View faults log***. To download the latest events list click button ***View events log***. A floppy disk icon button allows you to save the list to a text file and ***Clear*** buttons clear respective banks, by removing all the records from the device. The first column shows the absolute number of fault/event, further follows the text, where the numeric code of the text is indicated in parentheses. ***Date/time*** can be displayed in two different formats, depending on the presence of the real time clock on the device. In case of their absence this column displays the short time format of hours:minutes, or else a long time format is used, containing the full date and time. The last column has a special purpose and can be used at the discretion of the programmer, for example, to display the number of the pump on which the fault occurred.

There is also a button in main window to “quick” view faults log:



Emulator-0001 v.01.00 2010-10-22 (00010022_15560100) [vendor token: FFFFFFFF]

Emulator.xml

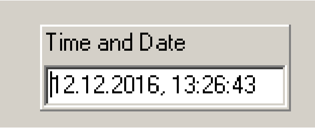
name	value	units
cmdCANena [5310,01]	1	U16

Note

The function is only available in case of fault journal presence in the device and the correct specified **control object binding settings.**

6.6. Real time clock

If the device has a real time clock, it is possible to control its work. For this purpose it is possible to create special editable control on control panel (see section 7.4.3 below):

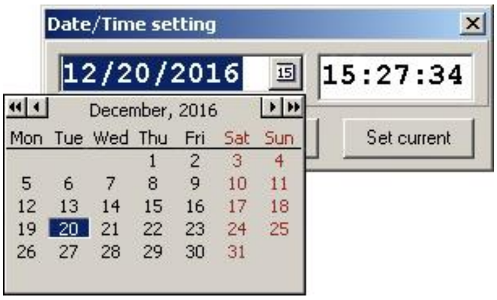


Time and Date

12.12.2016, 13:26:43

This control shows the time and date currently set on the device.

For correcting you need to click on the control and set the desired values at the appeared window.



12/20/2016 15:27:34

December, 2016

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

The time is displayed dynamically (it is seen as the time “goes”). If seconds do not change, or the inscription is absent, then most likely, the clock on this device is not installed, or communication parameters settings may be set incorrectly.

Note

The panel is only available if a real time clock is present in the device and the correct specified **control object binding settings.**

6.7. Parameters transfer

6.7.1. Features

Parameters transfer function is a HW/SW service for transfer settings between different CANopen devices. It has the following features:

- **saving device parameters to your computer’s hard drive:**
    - the ability to specify a file name and folder for saving at the users discretion;
    - the ability to update the entire device dictionary or only the device parameters values before saving to a file;
    - displaying of a device parameters list in a tree-like structure with the ability to change settings such as parameters grouping (by groups/ groups and indexes/ by indexes);
    - switching between the parameters text displaying modes (full/ short);
    - selection of a particular set of parameters/group of parameters from the list to transfer.
- **loading device parameters from your computer’s hard drive:**
    - the ability to select a file for loading from any directory;
    - selection of a particular set of parameters/group of parameters from the list to transfer;
    - displaying of a device parameters list in a tree-like structure with the ability to change settings such as parameters grouping (by groups/ groups and indexes/ by indexes);
    - switching between the parameters text displaying modes (full/ short);
    - the comparison of context (format/text) between parameters selected for loading and device parameters before loading. Automatic cancellation of the parameter loading in case of different context.
    - Displaying of log of all operations, canceled or failed loads and loading progress.

! **Note**

Loading device parameters from file to device is available only at the «Service» or «Master» access level (see section 7.3 «Access level» below).

6.7.2. Using techniques

6.7.2.1. Saving device parameters to your computer hard drive

To save the device parameters to your computer's hard drive, perform the following steps:

1. If RTCON is not running, start it. After starting the program a list of devices present in the CAN network should be formed – see Fig. 6.6:

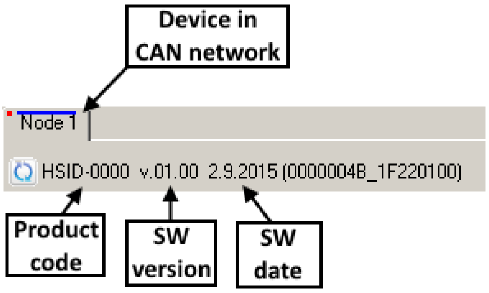
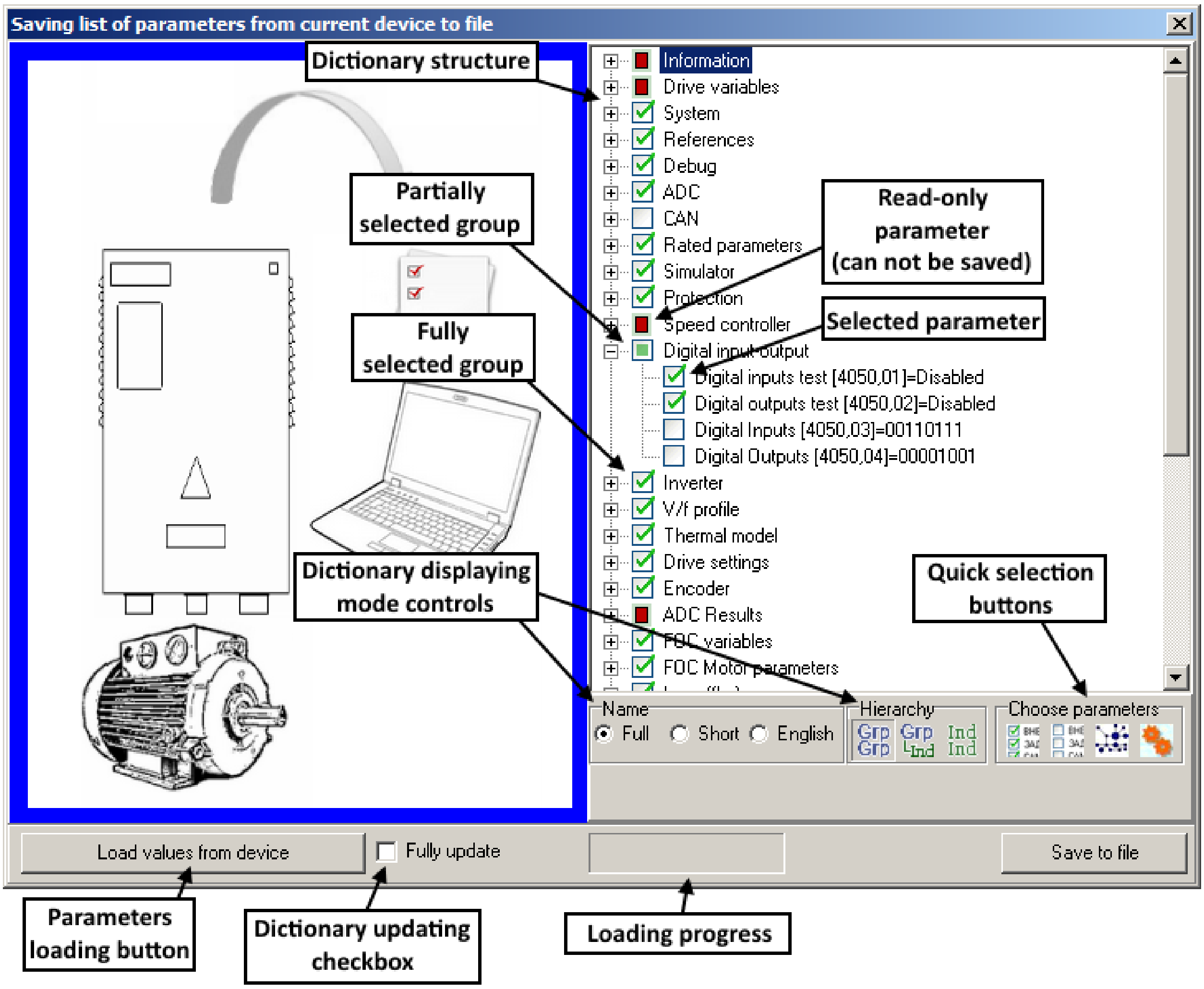


Fig. 6.6 List of devices in the network

   If the list of devices is not formed, it is necessary to check the quality and accuracy of USB/CAN adapter connection, check compliance of the adapter and communication module parameters in ***Menu/Service/Communication module parameters***.
2. Connect to the interesting device.
3. To call the device parameters saving window, select ***Menu/Service/Parameters transfer/Save to file***. The appropriate window opens – see Fig. 6.7.
4. Before you save the parameters to a file you need to download parameters from the device. To do this, click on ***Load values from device***. Pay attention to ***Fully update*** checkbox – see Fig. 6.7. If the box is checked, then when you click on the values loading button this will download the device dictionary completely – structure, formats, texts, etc. If the box is checked, then click on the ***Load values from device*** will cause download of the parameters values only. Thus, if the user is sure that the current dictionary is identical to the device dictionary, he can uncheck the box to speed up parameters loading, otherwise it's better to check the box.
5. After downloading of the device parameters the dictionary structure is displayed in a tree view – see Fig. 6.7.



- 🟥 Information
- 🟥 Drive variables
- ☑ System
- ☑ References
- ☑ Debug
- ☑ ADC
- □ CAN
- ☑ Rated parameters
- ☑ Simulator
- ☑ Protection
- 🟥 Speed controller
- 🟩 Digital input output
  - ☑ Digital inputs test [4050.01]=Disabled
  - ☑ Digital outputs test [4050.02]=Disabled
  - □ Digital Inputs [4050.03]=00110111
  - □ Digital Outputs [4050.04]=00001001
- ☑ Inverter
- ☑ V/f profile
- ☑ Thermal model
- ☑ Drive settings
- ☑ Encoder
- 🟥 ADC Results
- ☑ FOC variables
- ☑ FOC Motor parameters

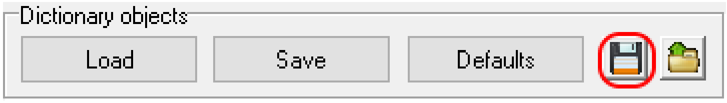
Name: ● Full ○ Short ○ English

□ Fully update

Fig. 6.7 Parameters saving window

6. Select parameters for saving. The selection is performed by right-clicking on the required element or group of parameters. It should be noted that regardless of the user-selected parameters the whole dictionary of the device is saved to your hard drive. Therefore, you can select the required parameters, at the phase of loading parameters to another device.
7. After selection of all required elements, click ***Save to file*** button. A standard dialog to save the files to your computer opens. Choose the necessary directory, enter the name and save the file.

To quick save all user parameters (with indexes 2500h and above) you may use button 💾 in the main window of application:



6.7.2.2. Loading settings from a file on the computer hard disk to the device

To load the device parameters from your computer to the device, perform the following steps:

1. If RTCON is not running, start it. After starting the program a list of devices present in the CAN network should be formed – see Fig. 6.6.
   If the list of devices is not formed, it is necessary to check the quality and accuracy of USB/CAN adapter connection, check compliance of the adapter and communication module parameters in ***Menu/Service/Communication module parameters***.
2. Connect to the interesting device.
3. To access the device parameters load window, select ***Menu/Service/Parameters transfer/Load from file***. The appropriate window opens – see Fig. 6.8.
4. Select the file from which you want to load parameters – click the button. A standard dialog to open the file opens.
5. The dictionary loaded from the specified file is displayed in the left pane of the parameters loading window – see Fig. 6.8.

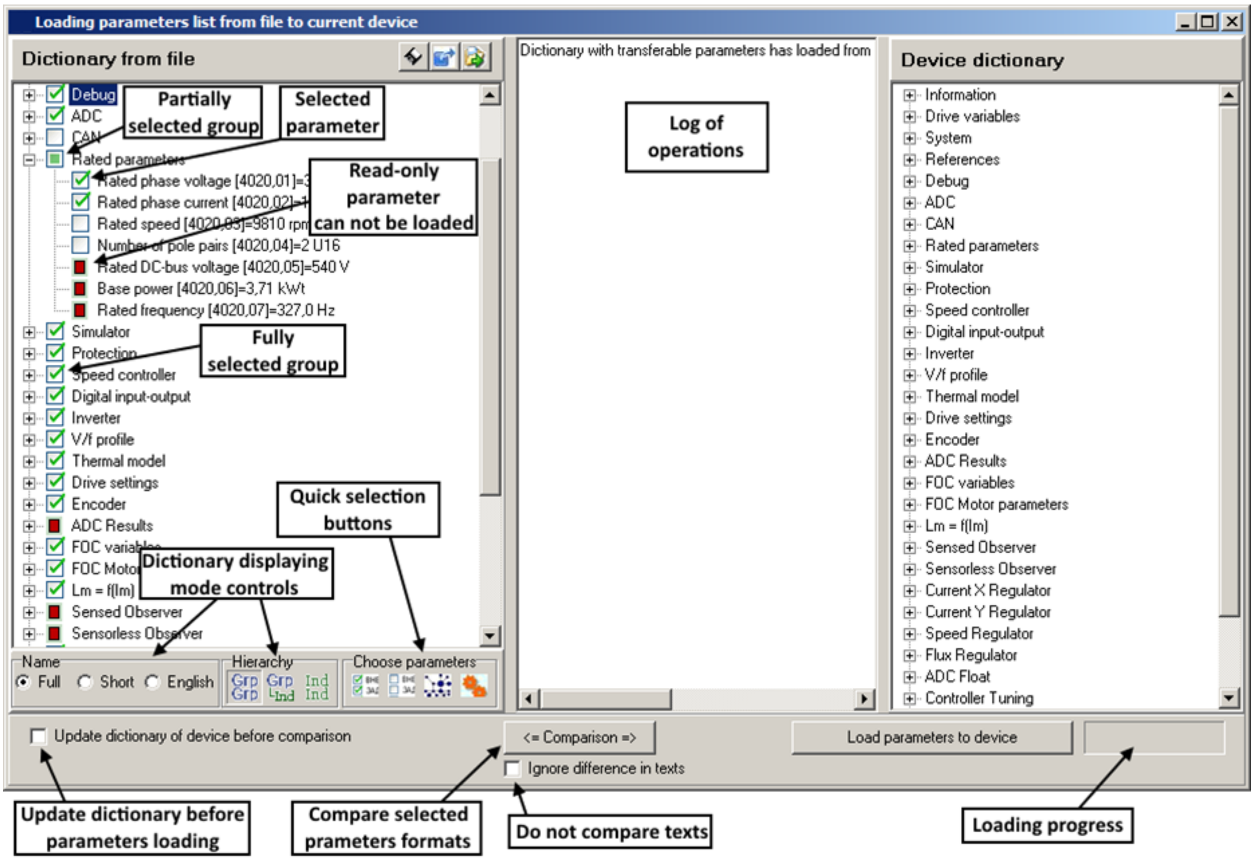


Fig. 6.8 Parameters loading window

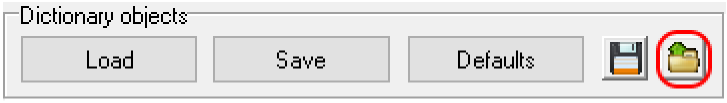
At this time, the tree will already contain checked boxes for parameters selected for saving/loading during saving of the file to your hard drive.

6. If necessary, modify the list of parameters for loading to the device. To do this, right-click to check or uncheck boxes next to the corresponding elements of the dictionary.
7. Next it is necessary to check the possibility of loading the selected parameters to the current device. Click ***Comparison*** button to do this. Pay attention to ***Update dictionary of device before comparison*** checkbox – see Fig. 6.8. If the box is checked, then clicking on the comparison button will update the device dictionary – structure, formats, texts, etc. If the box is unchecked, then comparison with the current dictionary will be performed.

Thus, if the user is sure that the current dictionary is identical to the device dictionary, he can uncheck the box to speed up parameters comparison, otherwise it's better to check the box.

8. After comparing the information will appear in the log window whether all the selected parameters can be loaded to the device or why they can't, etc.
9. After reviewing the comparison log, if the user still intend to load parameters to the device it is necessary to click the ***Load parameters to device*** button. When loading the parameters the user must clearly understand what he is doing and should be familiar with the safety precautions in this manual, because this can lead to unpredictable results – for more details see paragraph 6.7.2.3.
10. User can save loaded and selected (on the left panel) parameters to new file by clicking to button .
11. Loaded parameters can be exported as plain text by clicking to button .
12. Close the load window to complete the parameters transfer process.

To quick load all parameters from file to device you may use button in the main window of application:



Note

Loading device parameters from file to device is available only at the «Service» or «Master» access level (see section 7.3 «Access level» below).

6.7.2.3. Parameters transfer function precautions

The unreasonable use of parameters transfer function can lead to unpredictable results. Consider the example of improper use of this function.

Example:

The user decides to transfer settings from one drive (already configured) to another, while the configured drive is in operation. The user connects to a running drive via RTCON and saves the drive settings to a file. Then he connects to unconfigured drive, loads the appropriate parameters file, clicks the button to choose to download all parameters except communications, compares dictionaries and clicks the ***Load parameters to device*** button. It seems that the user didn't do anything dangerous, but it is not so – as a result of such actions the drive could start operation immediately after parameters loading! This is due to the fact that there was a parameter *510B.01 sm_ctrl.sm_state* in the drive dictionary – which is responsible for the state of the control system main state machine – for details, see the instructions for the electric drive. Because user had saved the settings from a running electric drive, a state other than "STOP" appears in this parameter. Therefore, downloading settings to another drive in further, the user is actually shifted it into operation mode different from the "STOP", which is dangerous and can cause serious hazard. The right action: before downloading the settings the user had to uncheck the box for loading of parameter *510B.01 sm_ctrl.sm_state*.

To avoid such situations, here are some guidelines for using parameters transfer function:

- It is advisable to save the parameters to a file at the stopped equipment. If it is not possible to stop the drive, you can save parameters, but be sure to specify that the settings taken from the operating equipment in the file name or in a different location to take it into account when loading them to another device.

- It is advisable to load the settings from the file to the device without high voltage power supply.
- The user performing parameters loading from the file to the device, should understand the purpose of all loaded parameters.
- If only a specific list of parameters is necessary to transfer you should not transfer all the rest at the same time. The transfer of unknown to user parameters can lead to undesirable results.
- Be sure to read the instructions to the equipment, on which you going to perform the parameters transfer.

6.8. Programming

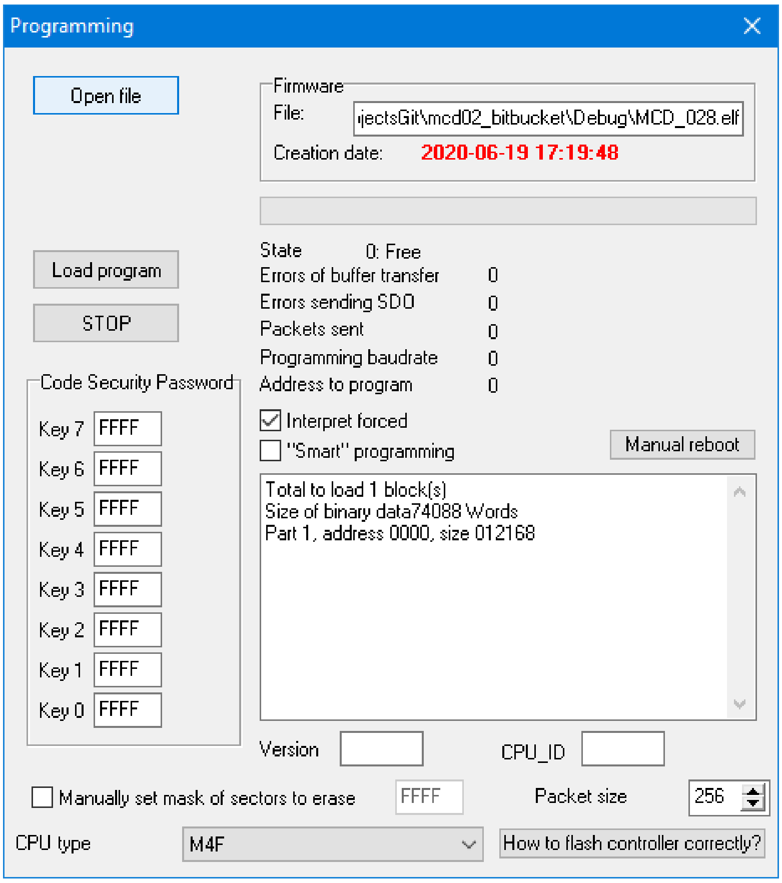
The programming function allows you to update the firmware of the microcontroller device via CANopen network. The new software is provided by the developer of the device. Programming window opens from the ***Service/Programming*** menu where you should select appropriate version of flash programmer.

Note

The programming function is available only at the «Service» or «Master» access level (see section 7.3 «Access level» below).

6.8.1. Flash programmer v1

This version of flash programmer is intended for one-core CPU like TMS320 F24x, F281x, F28335 (firmware file has extension «out») and ARM Cortex M4F (firmware file has extension «elf»).



File: ijectsGit\mcd02_bitbucket\Debug\MCD_028.elf
Creation date: 2020-06-19 17:19:48

State 0: Free
Errors of buffer transfer 0
Errors sending SDO 0
Packets sent 0
Programming baudrate 0
Address to program 0

☑ Interpret forced
□ "Smart" programming

Key 7 FFFF
Key 6 FFFF
Key 5 FFFF
Key 4 FFFF
Key 3 FFFF
Key 2 FFFF
Key 1 FFFF
Key 0 FFFF

```
Total to load 1 block(s)
Size of binary data74088 Words
Part 1, address 0000, size 012168
```

Version
CPU_ID
□ Manually set mask of sectors to erase FFFF
Packet size 256
CPU type M4F

Fig. 6.9 Flash programmer v1

To program flash, follow these steps:

1. Update device dictionary by clicking ***Update dictionary*** button.
2. Save device parameters to a file on your computer.
3. Open programming window via menu Service → Programming → Flash programmer v1.
4. Change processor type if necessary*:
   - Texas Instruments 24x/28x (firmware file has extension «out»);
   - ARM, core M4F (firmware file has extension «elf»).
5. If necessary, set the mask of sectors to erase (depends on device software), by default - FFFF**.
6. If necessary, specify «code security password» for unlocking the processor (depends on device software), by default - FFFF (password not used).

7. Open file with new firmware.
8. Click ***Load program*** button. If programming is successfully completed, the corresponding message “Programming completed” will be displayed, then the device will reboot. Be careful with the selection of an updated software file and do not turn off the device before programming is finished. If an error occurs during programming, try to restart RTCON and update software again. Do not power down the device! If power is switched off before successful completion of the programming, the device will be unable to update via the CAN network and will not operate. It will be necessary to call a developer company specialist, open the device to access the connector on the controller and update software via JTAG programmer.
9. Refresh the device dictionary after programming.
10. Load settings from the file on the computer to which you saved the settings in step 2.
11. Save parameters in the controller’s non-volatile RAM: ***Save parameters*** button is on the top right in the RTCON main window.
12. Reboot the device. This can be done by either a power reset or a software means: most of devices have a reboot parameter in the dictionary where it needs to write “1” to reboot the device: for example, *System.Reset CPU*.
13. Update the dictionary again.

The programming process is over.

If the option “Smart programming” is checked then the "full workflow" of the programming (steps 2 – 13 above) will be automatically executed.

The short version of the instructions for programming is available on the button ***How to flash controller correctly?*** in RTCON programming window.

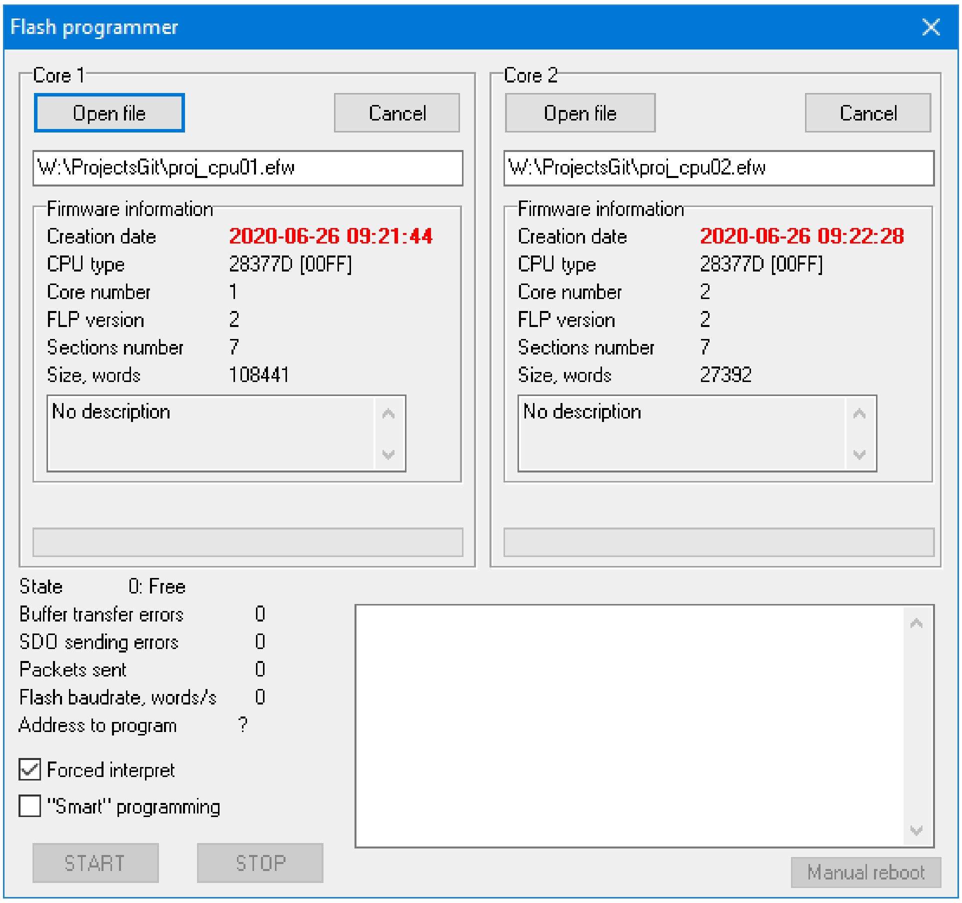
A string with firmware date is colorized. The color depends on build date of the firmware file. If the file is more than 2 hours old, the color will be red. Otherwise the color will become closer to green. For example, if the file was built 1 hour ago, the color will be yellow; if the file was built 10 minutes ago, the color will be green. This helps to keep attention to the files during project development.

* Note: the type of processor is also set in device profile (see section 7.4.2 below). When you open ***Programming*** window the type of processor (and extension of firmware file name) will be set automatically based on information from the profile.

** Note: if check box «Manually set mask of sectors to erase» is cleared, then parameter 2702.00 of type U16, which should contain the mask of the sectors (it is initialized by the device software), is requested before the firmware loading. If this parameter in the device dictionary is missed or zero, then the mask FFFF is used.
For a project with a bootloader, a FFFE mask is usually used, without a bootloader - FFFF.

6.8.2. Flash programmer v2

This version of flash programmer is intended for two-core CPU like F28377D/28379D and one-core CPU like F28377S/28379S, firmware file has extension «efw».



Core 1

W:\ProjectsGit\proj_cpu01.efw

Firmware information

Creation date	2020-06-26 09:21:44
CPU type	28377D [00FF]
Core number	1
FLP version	2
Sections number	7
Size, words	108441

No description

Core 2

W:\ProjectsGit\proj_cpu02.efw

Firmware information

Creation date	2020-06-26 09:22:28
CPU type	28377D [00FF]
Core number	2
FLP version	2
Sections number	7
Size, words	27392

No description

State	0: Free
Buffer transfer errors	0
SDO sending errors	0
Packets sent	0
Flash baudrate, words/s	0
Address to program	?

☑ Forced interpret

□ "Smart" programming

Fig. 6.10 Flash programmer v2

To program flash, follow these steps:

1. Update device dictionary by clicking ***Update dictionary*** button.
2. Save device parameters to a file on your computer.
3. Open programming window via menu Service → Programming → Flash programmer v2.
4. Open firmware efw-file(s) for appropriate core (you may flash two cores at once).
5. Click ***START*** button. If programming is successfully completed, the corresponding message "Programming completed" will be displayed, then the device will reboot. Be careful with the selection of an updated software file and do not turn off the device before programming is finished. If an error occurs during programming, try to restart RTCON and update software again. Do not power down the device! If power is switched off before successful completion of the programming, the device will be unable to update via the CAN network and will not operate. It will be necessary to call a developer company specialist, open the device to access the connector on the controller and update software via JTAG programmer.
6. Refresh the device dictionary after programming.
7. Load settings from the file on the computer to which you saved the settings in step 2.
8. Save parameters in the controller's non-volatile RAM: ***Save parameters*** button is on the top right in the RTCON main window.
9. Reboot the device. This can be done by either a power reset or a software means: most of devices have a reboot parameter in the dictionary where it needs to write "1" to reboot the device: for example, *System.Reset CPU*.
10. Update the dictionary again.

A string with firmware date is colorized. The color depends on build date of the firmware file. If the file is more than 2 hours old, the color will be red. Otherwise the color will become closer to green. For example, if the file was built 1 hour ago, the color will be yellow; if the file was built

10 minutes ago, the color will be green. This helps to keep attention to the files during project development.

If the option “Smart programming” is checked then the "full workflow" of the programming (steps 2 – 10 above) will be automatically executed.

6.8.3. CAN-bootloader activation

Menu ***«Service»*** → ***«Programming»*** → ***«CAN-bootloader activation»*** opens window to activate CAN-bootloader.

This function is intended for devices with embedded CAN-bootloader. The main function of such a bootloader is to restore the operability of the device via the CAN network (without direct access to device) after unsuccessful firmware, for example, due to a power failure.

When the device is turned on, the bootloader listens to the CAN network for 100 ms at a speed of 125 kbps waiting for a special message. If such a message appears, the bootloader delegates control to the programmer module, which becomes visible in the CAN network as a node with name "CAN_prog". You can connect to this node in a standard way and flash the software.

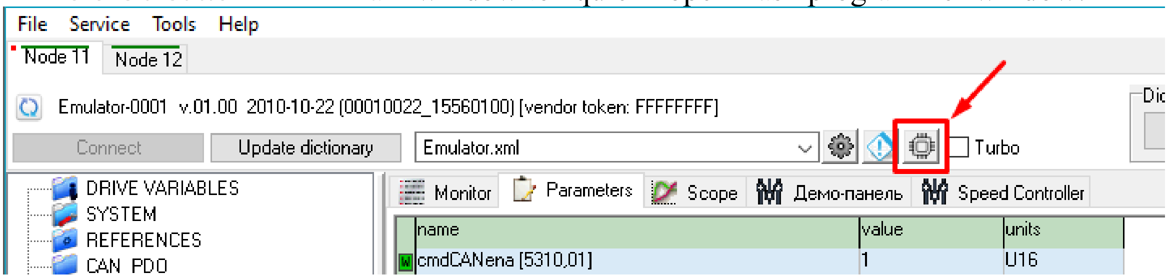
Since the bootloader only listens to the CAN line and can't set confirmation (acknowledge), then for successful message delivery you need to have two CAN-devices. As a second device, you can use, for example, the second channel of CAN-bus-USB adapter (Marathon) with the "canwise" utility running on PC.

Follow the next steps to activate the CAN bootloader:

- Check that the data transfer rate is set to 125 kbps in the communication module settings.
- Make sure that there is at least one more device in the CAN network besides the target device.
- Press the "Start" button below - a special message will be broadcast into the CAN network (with period of 50 ms).
- Reboot the target device (via reset button or power off/on).
- In case of successful activation of the bootloader, the device named "CAN_prog" should appear in the list of available nodes.
- Close the activation window.
- Connect to the "CAN_prog" node and flash it as usual.

6.8.4. «Quick» way to open flash programmer

There is a button in main window to “quick” open flash programmer window:



Emulator-0001 v.01.00 2010-10-22 (00010022_15560100) [vendor token: FFFFFFFF]

name	value	units
cmdCANena [5310.01]	1	U16

When you press this button the application tries automatically detect appropriate version of flash programmer as follows:

a) Using CPU type from device’s profile (see section 7.4.2). If it’s not specified then:

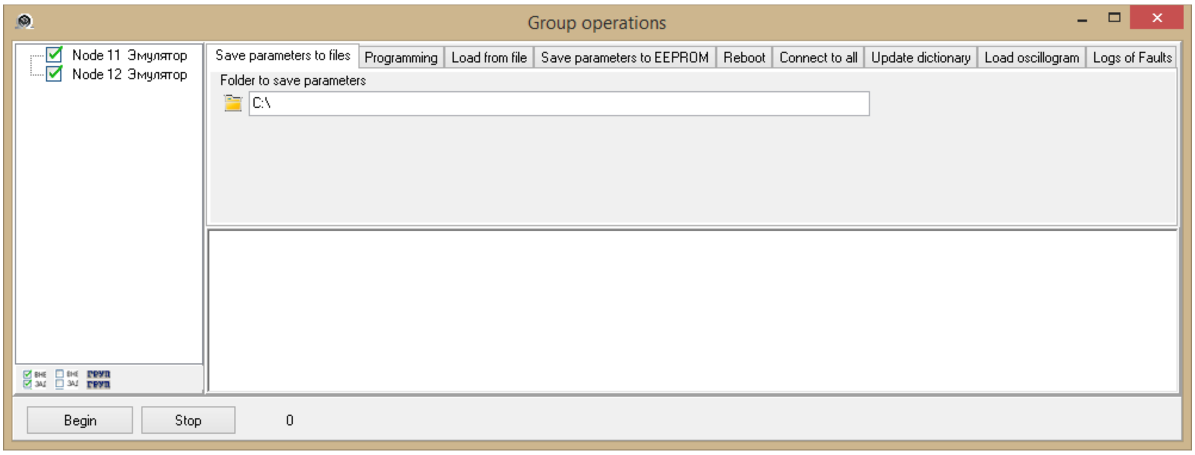
b) Reading flash programmer version from object 2701.0E. If this parameter is equal to 0 or 1 then programmer v1 will be used, if it is equal to 2 then programmer v2 will be used. Otherwise:

c) Reading CPU type from object 2701.0C. If CPU type is unknown then:

d) The user will be prompted to select appropriate version of flash programmer.

6.9. Group operations

The "Group operations" function, which can be launched in the Service menu, allows you to perform the same operations with several devices on the network at once. The window for such an operation is shown in the figure below:



Folder to save parameters

C:\

User selects which group operation he wants to perform by selecting a corresponding tab, sets the required operation settings, selects the necessary devices in the list on the left (all devices on the network are selected by default), and then starts the operation with the "Begin" button. This functionality is implemented in a "script" way, by simulating user actions with the RTCON interface. After the start of the group operation, additional windows for parameter transfer, programming, etc. can be opened (depending on the type of operation). Do not interfere with the work process (for example, by closing windows, etc.) – this will disrupt the group operation progress.

A wide range of operations are supported:

- **Save parameters to files** – allows you to save the parameters from the devices as files to the PC using the path specified by the user. To perform the operation, the function **Service / Parameters transfer / Save to file**, called automatically for all selected devices is used.
- **Programming** – allows you to program (flash) all selected devices with one version of the software selected by user. The **Service / Programming** function is used.
  This operation is available only at the «Service» or «Master» access level (see section 7.3 «Access level» below).
- **Load from file** – allows you to load the values of the parameters that were previously stored in files on the computer into devices. There are two options. The first option is to **load from single file to all devices**. In this case, the user selects the path to the file with the parameters. The second option is to **load automatically using appropriate file with parameters from the folder**. In this case, the user specifies the path to the directory where the parameters files for the different devices represented on the network are located. RTCON automatically analyzes devices on the network and files in the directory, and tries to determine which file should be loaded to each device by the device node number and name. The matching list is displayed to user for verification. If it is correct, the user clicks "Begin". If there are several options for the same device with different dates, the function tries to find the file with the latest date. To perform the operation, the function **Service / Parameters transfer / Load from file**, called automatically for all selected devices is used.
  This operation is available only at the «Service» or «Master» access level (see section 7.3 «Access level» below).

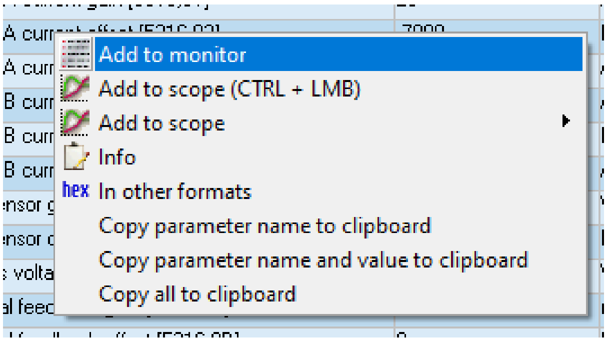
- **Save parameters to EEPROM** – sends each device a command to store the parameter values in the non-volatile memory of the parameters, i.e. the "Save parameters" button on the right side of the top of the main program window is clicked for selected devices.
- **Reboot** – sends each device a command to reboot. To perform the operation, the programmer's ("CANopen programmer") functionality is used, which, in addition to updating the software, has a command to reboot the device.
- **Connect to all** – connects to all devices on the network, i.e. the "Connect" button is clicked sequentially.
- **Update dictionary** – updates the dictionary for all devices, pre-checking whether it needs to be updated. The dictionary will be updated only for those devices that have the dictionary hash mismatching the hash of the already downloaded dictionary.
- **Load oscillogramm** – the operation loads the oscilloscope variables profile to all selected nodes, selects the "sampling" mode and leaves the oscilloscope running.
- **Logs of faults** – downloads fault logs from all selected devices and saves the result in different files on the selected path.

6.10. Logging selected parameters to a file

Ability to save the flow of selected parameters values is implemented via RTCON Monitor function.

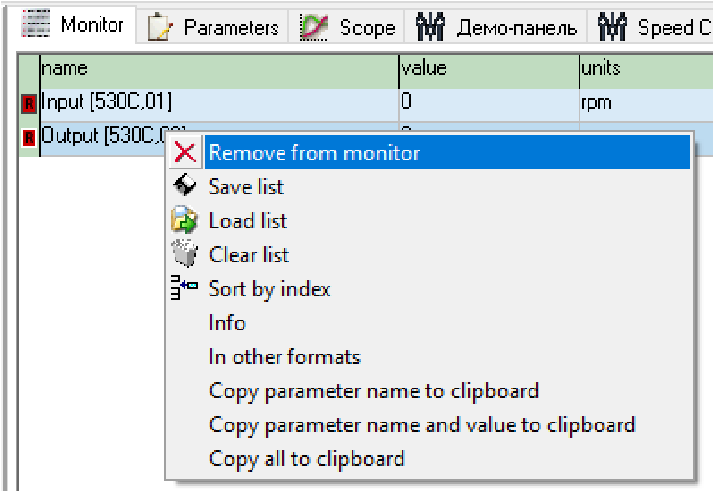
Monitor tab allows you to group interesting object dictionary variables in one window.

To add a variable to the monitor right-click on the interesting variable in the dictionary and select ***Add to monitor***.


Add to monitor
Add to scope (CTRL + LMB)
Add to scope
Info
hex In other formats
Copy parameter name to clipboard
Copy parameter name and value to clipboard
Copy all to clipboard

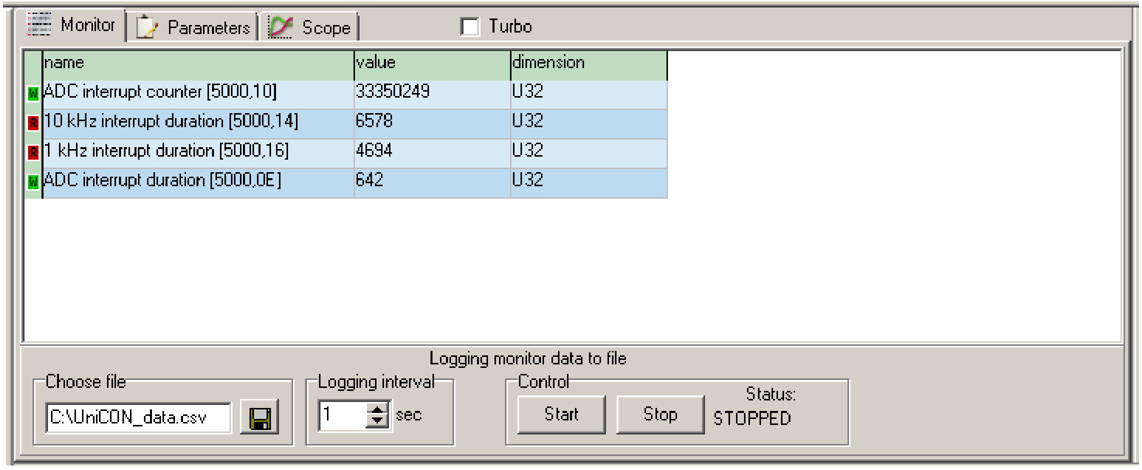
Removal of a variable is performed on the ***Monitor*** tab from the context menu.

The variables in the monitor are arranged in a list that can be saved to a file and loaded from a file. For this purpose there are appropriate context menu items.



name	value	units
Input [530C,01]	0	rpm
Output [530C,0	0	

In addition to displaying of the selected parameter list on the screen function, you can activate the logging of the selected parameters to a file on the bottom of the monitor tab window.



	name	value	dimension
W	ADC interrupt counter [5000,10]	33350249	U32
R	10 kHz interrupt duration [5000,14]	6578	U32
R	1 kHz interrupt duration [5000,16]	4694	U32
W	ADC interrupt duration [5000,0E]	642	U32

Logging monitor data to file

Choose file: C:\UniCON_data.csv

Logging interval: 1 sec

Control

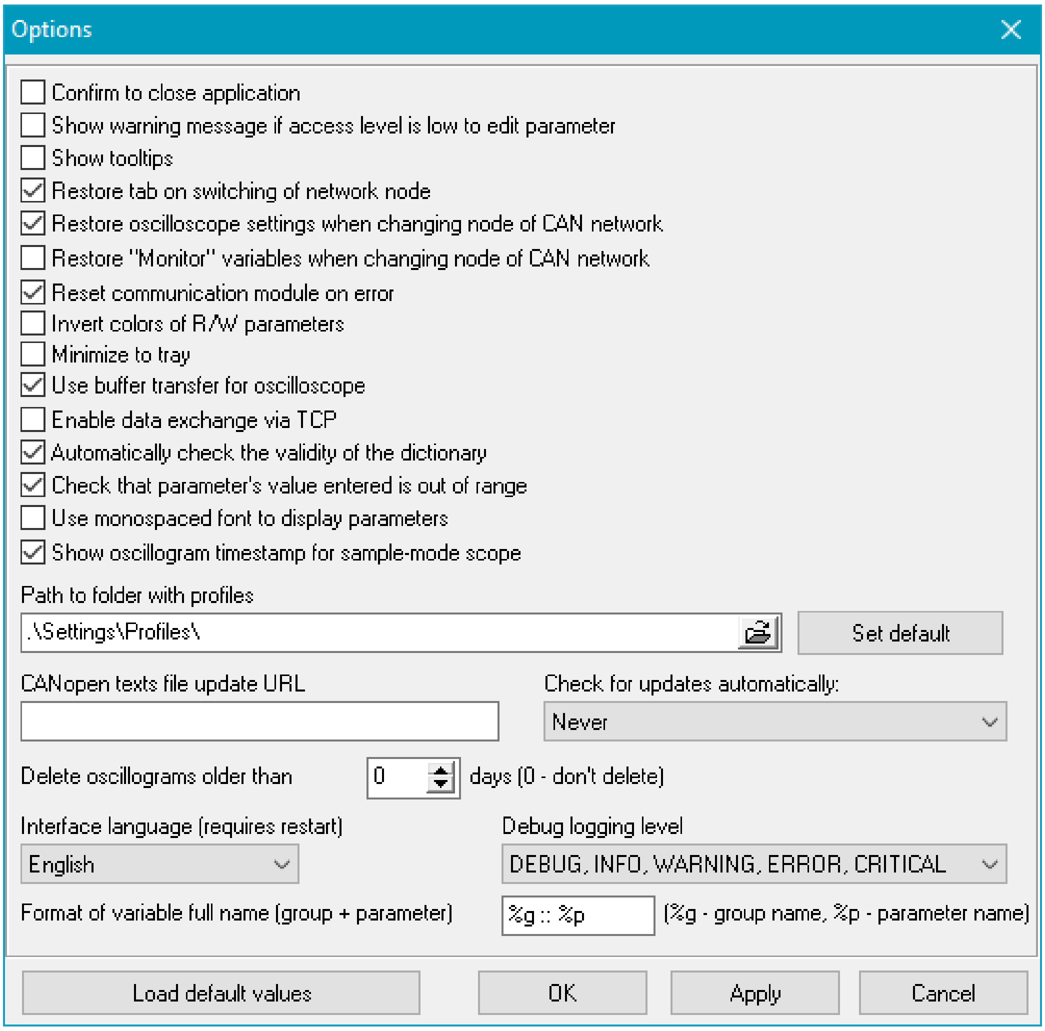
Status: STOPPED

The data is saved in a .csv file, which is supported by a variety of office software packages, such as Microsoft® Excel and OpenOffice.org Calc. Name and path for saving can be selected by 💾 button. Values of all the variables of the monitor, as well as the time stamp are recorded in the lines of the file with period selected in the ***Logging interval***. Writing to a file is initiated by the ***Start*** button. ***Stop*** button stops the recording. If the file already exists, the data will be recorded to its end. When you start recording a “cap” of the table with the names of logged variables is placed to a file in addition to the data itself.

7. Options/ Interface

7.1. Options

Options window opens from ***Service/Options***.



☐ Confirm to close application
☐ Show warning message if access level is low to edit parameter
☐ Show tooltips
☑ Restore tab on switching of network node
☑ Restore oscilloscope settings when changing node of CAN network
☐ Restore "Monitor" variables when changing node of CAN network
☑ Reset communication module on error
☐ Invert colors of R/W parameters
☐ Minimize to tray
☑ Use buffer transfer for oscilloscope
☐ Enable data exchange via TCP
☑ Automatically check the validity of the dictionary
☑ Check that parameter's value entered is out of range
☐ Use monospaced font to display parameters
☑ Show oscillogram timestamp for sample-mode scope

Path to folder with profiles: .\Settings\Profiles\ [Set default]

CANopen texts file update URL:

Check for updates automatically: Never

Delete oscillograms older than 0 days (0 - don't delete)

Interface language (requires restart): English

Debug logging level: DEBUG, INFO, WARNING, ERROR, CRITICAL

Format of variable full name (group + parameter): %g :: %p (%g - group name, %p - parameter name)

Confirm to close application, ***Show tooltips*** and ***Interface language*** options do not require explanation.

Option ***Show warning message if access level is low to edit parameter*** defines whether a warning message will be displayed if you try to edit parameter but your access level is low for this operation.

Option ***Restore tab on switching of network node*** defines whether the current active tab (Monitor / Parameters / Scope) will be saved/restored when user switch to other network node.

Option ***Restore oscilloscope settings when changing node of CAN network*** defines whether individual oscilloscope settings for each network node will be saved/restored. If the option is OFF then when you switch to another network node, its oscilloscope settings will be reset (variables and scales/offsets will have to be set again).

Option ***Restore «Monitor» variables when changing node of CAN network*** defines whether the list of variables on «Monitor» tab for each network node will be saved/restored. If the option is OFF then when you switch to another network node, its «Monitor» tab will be cleared.

Restart communication module on error allows the program to attempt to reinitialize the USB-CAN adapter; if the driver generated an error for some reason (the USB connector has dropped out, adapter hangs, etc.). Since the adapter driver version changes and different adapters are supported, this option works in different ways, sometimes incorrectly. You should make a decision about its use depending on the situation.

Invert colors of R/W parameters changes color (green and red) for the read/write parameters to a more neutral.

Minimize to tray allows the program window to minimize to system tray icon instead of the regular minimize.

Use buffer transfer for oscilloscope allows using the option of packet transmission via the CANopen to download oscillograms in the ***Sample*** mode. If for some reason there is a problem with loading the oscillograms, you can try to disable this option. In this case waveform data will be downloaded point-by-point as usually.

Enable data exchange via TCP allows RTCON to use a network connection to exchange CAN network traffic with other programs. For this RTCON starts a TCP server – opens the network port number 5000 and awaits client connections. Currently sending of all CAN network messages via TCP/IP protocol is supported only. RTCON automatically begins to send all network traffic to each connected client in following format:

u32	package data
u32	id
u8	package length
u32	timestamp
u8	control field

If the network exchange is not required, you can disable this option.

Automatically check the validity of the dictionary allows RTCON to use object dictionary hash to check whether the structure of the object dictionary in the device corresponds the structure of the dictionary in RTCON, previously downloaded and saved to a file. If it is not so, the user will be prompted to refresh the dictionary (***Update dictionary*** button will blink).

Show oscillogram timestamp for sample-mode scope enables displaying the time of data download from the device in the upper right corner of the oscillogram window (if scope is in the "Sample" mode). If data recording to the device's buffer is initiated manually by the "Update" button, the time of pressing this button is actually displayed. If the data in the device's buffer appeared after the trigger was fired (see section 6.4.3.2), then the trigger firing time will be displayed.

Path to folder with profiles allows you to specify the path where custom device profiles (control panels) created by the user will be stored. For more information about profiles, see chap. 7.4. Pressing button «Set default» will set default path.

CANopen texts file update URL allows to download new file "coTextBase.xml" with texts from the remote server (this file will be saved to application folder overwriting the existing one). It is supposed that no authentication is needed to get the remote file. The update can be initialized via menu ***Service/Update file with CANopen texts***. Example of URL: http://SERVER/PATH/coTextBase.xml

Check for updates automatically allows to setup the period of automatic update of CANopen texts file.

Delete oscillograms older than N days allows to automatically (every time you start the application) delete old oscillogram files automatically saved to the RTCON\scope folder. Zero value disables this feature.

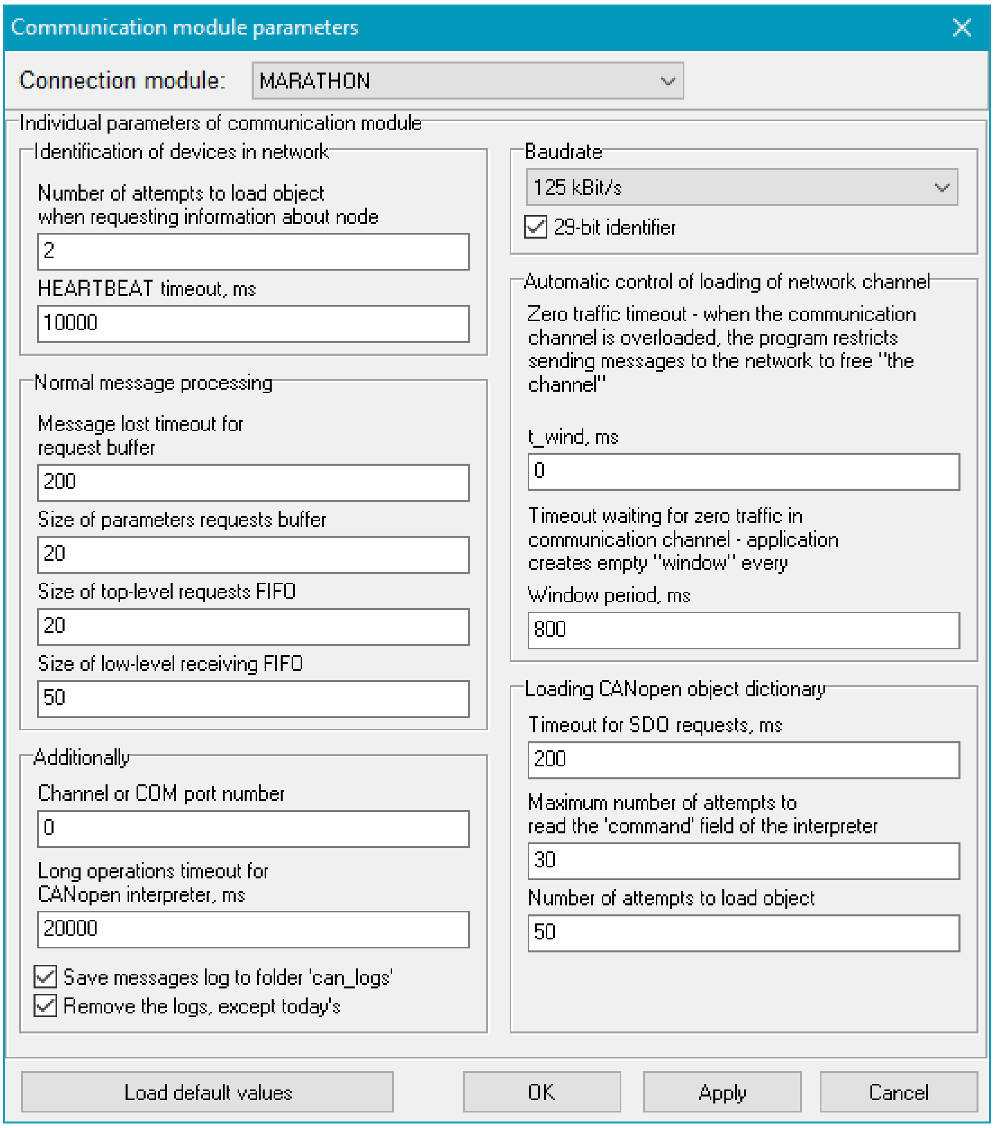
Debug logging level option allows you to control the verbosity of debug logging. A debug log is a log file that is automatically generated in the «debug_logs» folder each time the application is started. Various diagnostic messages are written to this file in text form. The following table describes the logging levels, starting with the lowest:

Logging disabled	No debug log is generated
CRITICAL	Only critical errors that cause the application to crash are logged
ERROR, CRITICAL	The log also stores errors that occur during operation and do not lead to application crash
WARNING, ERROR, CRITICAL	The log also stores warnings
INFO, WARNING, ERROR, CRITICAL	The log also stores informational messages, for example, user actions in the graphical interface of the application (pressing buttons, selecting menu commands, etc.)
DEBUG, INFO, WARNING, ERROR, CRITICAL	The log also stores low-level debug messages

! **Attention.** Debug log files older than five days are automatically deleted.

Format of variable full name (group + parameter) defines the format of full variable name in the scope variables table and ‘Copy parameter name…’ context menu. You may use %g and %p specifiers to insert the name of group and the name of parameter respectively. The name of parameter includes it’s address [index.subindex].

7.2. Communication module parameters



Communication module parameters

Connection module: MARATHON

Individual parameters of communication module

Identification of devices in network

Number of attempts to load object when requesting information about node: 2

HEARTBEAT timeout, ms: 10000

Normal message processing

Message lost timeout for request buffer: 200

Size of parameters requests buffer: 20

Size of top-level requests FIFO: 20

Size of low-level receiving FIFO: 50

Additionally

Channel or COM port number: 0

Long operations timeout for CANopen interpreter, ms: 20000

☑ Save messages log to folder 'can_logs'

☑ Remove the logs, except today's

Baudrate: 125 kBit/s

☑ 29-bit identifier

Automatic control of loading of network channel

Zero traffic timeout - when the communication channel is overloaded, the program restricts sending messages to the network to free "the channel"

t_wind, ms: 0

Timeout waiting for zero traffic in communication channel - application creates empty "window" every

Window period, ms: 800

Loading CANopen object dictionary

Timeout for SDO requests, ms: 200

Maximum number of attempts to read the 'command' field of the interpreter: 30

Number of attempts to load object: 50

7.2.1. General parameters setting

Network Settings allow you to select the desired communication module, set the speed of its operation, and also define the advanced settings for communication module driver connection with the RTCON.

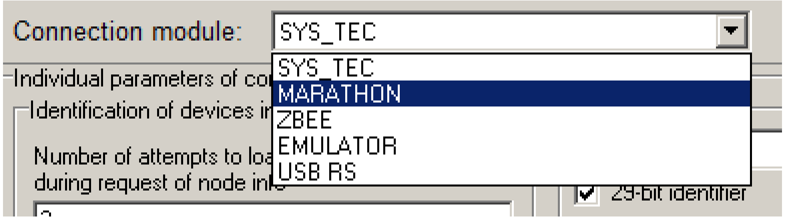
To customize network interface parameters, click ***Communication module parameters*** in the ***Service*** menu.

❗ **Attention.**

It is not recommended to change the default advanced settings if there is no confidence in the correctness of the changes!

To **select the needed communication module**:

1. Click the ***Communication module*** dropdown list.
2. Select the communication module you are using from the list:



Connection module: SYS_TEC

SYS_TEC
MARATHON
ZBEE
EMULATOR
USB RS
Individual parameters of co
Identification of devices in
Number of attempts to loa
during request of node in
29-bit identifier

3. To apply the setting, click on ***Apply*** or ***OK*** button.

To **restore all network settings to defaults**:

1. Click ***Load default values*** button.
2. To apply the setting, click on ***Apply*** or ***OK*** button.

7.2.2. Advanced parameters setting

7.2.2.1. Setting the baud rate

The baud rate of the adapter you are using physical channel can be selected from the ***Baudrate*** dropdown list.

Note

Baud rate of your adapter must be equal to the baud rate in the network you want to connect. The recommended baud rate is **125 kb/sec**.

7.2.2.2. Size of low level FIFO on receiving

Specifies the size of the FIFO type (“first in – first out”) buffer where messages received from the network are stored. Do not define too big or too small buffer size.

If the buffer is too large, it may lead to the accumulation of the message queue (e.g., the program does not have time to process incoming messages). Large queue will lead to long time delays between the request arrival and its processing by the program, which is unacceptable.

If the buffer is too small, this may lead to its overflow by the received messages, which in its turn will cause the loss of subsequent messages. It is also not desirable.

It is recommended not to change the value of this field.

7.2.2.3. Number of attempts to load “item” during request of node info

When a new node appears in the network program asks for its identification information. This parameter defines how much attempts to request this information must the program perform. The value is set based on the condition of node – program communication channel reliability.

7.2.2.4. Size of high level FIFO requests

Specifies the size of the FIFO type (“first in – first out”) buffer where upper level control system requests are stored. This buffer generates a request queue, which is handled by the communication module interface driver. The size of this buffer should not be too large or too small.

If the buffer is too large, it may lead to the accumulation of the request queue. Large queue will lead to long time delays between the request arrival to the communication module interface driver and sending of the request to the network, which is unacceptable.

If the buffer is too small, it can lead to its overflow by received from the upper level system requests (for example, if the program issues requests in blocks that exceed the size of the buffer size). It is permissible, but creates difficulties for upper level program in requests forming. This is also undesirable.

It is recommended not to change the value of this field.

7.2.2.5. Size of buffer for parameters requests

After interface driver retrieves next request from the requests FIFO the request is decrypted and sent to the network. To filter the messages received from the network interface driver needs to know what messages you want to accept, and which ones you don’t. For this purpose, the driver has a special buffer containing a list of active requests (an active request is a request that was sent to the network, but the response to it is not received yet).

Obviously, the larger the buffer, the greater number of messages can be handled simultaneously. However, too large buffer may cause a heavy load of the communication channel, which is extremely undesirable in responsible networks.

It is recommended to change this setting down from the default value only (down to 1).

❗ **Attention.**

If the ZigBee module is selected as an adapter, this parameter must be set to 1.

7.2.2.6. Message lost timeout for request buffer

Defines the time for which the interface driver will wait for a response from the network to the request sent. If the timeout is over, then the request is removed from the requests buffer to clear space for other upper level system requests.

The timeout should be chosen for reasons of maximum request fulfillment time of the addressed node plus sending and software processing costs plus a small margin for unforeseen delays, but no more than that, since the timeout value directly affects the interface driver bandwidth. A default value set by the developer is recommended.

7.2.2.7. Number of attempts to load item (when loading dictionary)

Downloading of the device object dictionary is complex and quite long operation. During this operation the network passes up to several thousand messages. Obviously, the algorithm of the dictionary downloading should be resistant to possible interference occurring on the line. This parameter is a part of the noise protection algorithm. It determines the number of times the interface driver will try to download the parameter in case of failure on the line. Obviously, the higher the value of this parameter, the more resistant to interference dictionary download process is. However, if the channel is bad, a large value of this parameter will increase the dictionary downloading time.

7.2.2.8. Maximum number of attempts to load interpreter field “command”

The node command interpreter is used when downloading node object dictionary. However, it is possible that the node interpreter is busy – for example, when a control panel (connected via CAN) operates with the node and some actions are performed on this panel. This setting determines how many times the interface driver is requesting access to the interpreter – if access is denied, then the program will display a message that interpreter is not available, and will offer to capture it “forcibly” – to the user’s discretion. It makes no sense to set this parameter very high – it will lead to increased delays. It is recommended to set it for reasons of maximum interpreter command execution time – defined by developers.

7.2.2.9. SDO request timeout

It is similar to the timeout for a request buffer (**Message lost timeout for request buffer**), but affects dictionary downloading only.

7.2.2.10. HEARTBEAT timeout

Each device in the CANopen network generates HEARTBEAT message. So node tells the rest of devices that it is “alive”. The program needs to know how often they send messages of this type for building a list of the devices present in the network. This parameter indicates the period of Heartbeat messages sending to the network. If the communication channel is heavily loaded, it makes sense to increase this parameter, because low-priority Heartbeat messages can wait for their turn to be sent to the network for a long time.

7.2.2.11. Long operations timeout for CANopen interpreter

The parameter defines the maximum time for the driver interface to wait for confirmation of interpreter command completion. The value should be a bit more than time of the longest interpreter operation fulfillment (saving controller parameters to non-volatile RAM can reach several seconds).

7.2.2.12. Channel or COM port number

Determines which channel of the communication module or which COM port of the PC is used to communicate with the device.

For example, for the two-channel communication module "MARATHON" you can set the values 0 or 1.

For a USB connection, you must specify the number of the COM port corresponding to connected USB device (you can see it in Device Manager of operating system).

7.2.2.1. Save messages log to folder “can_logs”

All communication can be logged to the log-file in a text form. If enabled, all messages will be recorded in the log file with the current date in “can_logs” directory at the root catalogue of the RTCON.

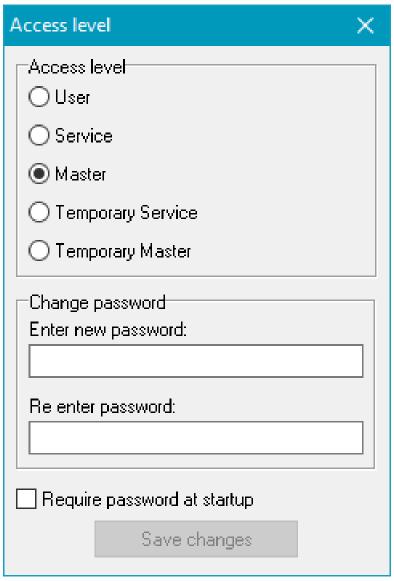
❗ Attention.

The size of a log-file for one day of operation with active network communication can reach several gigabytes.

7.2.2.1. Remove the logs, except today’s

If the option is set, all log-files except for the file with the current date will be automatically removed. This is useful when the logging option is selected, because the size of the log files is very large, and they are rarely required (in the rare cases that require a detailed analysis of what happened etc.).

7.3. Access level



Access level

Access level

○ User

○ Service

● Master

○ Temporary Service

○ Temporary Master

Change password

Enter new password:

Re enter password:

□ Require password at startup

This dialog box is intended to change the access level or change the password of “Service” or “Master”.

The ability to edit a parameter depends on its type (symbol inside the box, to the left of parameter name — see section 6.2 “Parameters editing” above) and the current access level:

Symbol	Type of parameter	Access level		
		User	Service	Master
R	Read-only	X	X	X
W	General editable	X	√	√
P	«Protected» editable	X	√	√
S	«Secret» editable	X	X	√

X – editing is not possible

√ – editing is possible

If the access level is insufficient for editing the parameter, then the box is gray: W otherwise, it is green: W.

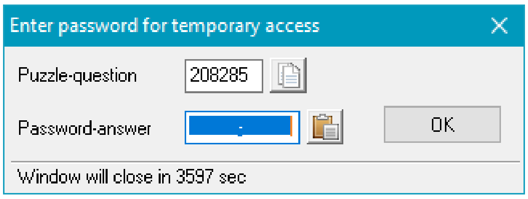
As you can see, at the “User” access level you can’t edit parameters at all. In addition, for the "User" the following functions are not available:

- programming the device (flash new firmware);
- loading parameters from file to the device;
- interaction with elements on the control panels (pressing buttons, modifying the values, moving sliders, toggling checkboxes – see section 7.4.3 “Control panels” below);
- entering to the device test mode (via “Service” → “Testing” menu).

When you try to change access level to "Service" or "Master", the current password will be requested.

If the option “Require password at startup” is enabled then the "Service" or "Master" password will be requested each time the application is launched.

“Temporary Service” and “Temporary Master” access levels are used if it is necessary to enable the high access level for a certain period of time. When you try to switch to this “temporary” access level, a window with a puzzle-question appears:



Enter password for temporary access

Puzzle-question: 208285

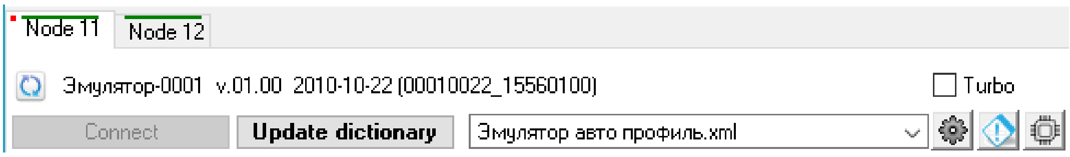
Password-answer:

Window will close in 3597 sec

The user must report this 6-digit number to the staff of the authorized organization responsible for setting up the device. In response, he will be informed of a 10-digit password-answer, which must be entered in the appropriate field followed by clicking "OK" button. This password will encode the duration of the temporary access level. If the correct password is entered, then the temporary access level is activated, and the time remaining until the end of this mode will be displayed in the application’s window title.

7.4. Control panel interface and profiles

Device profile defines:
1) Structure and functions of all elements of the main control panel and additional panels (on tabs);
2) Objects for working with faults and events banks;
3) Objects for working with scope in sample mode;
4) Objects for working with equipment testing algorithms.



Эмулятор-0001 v.01.00 2010-10-22 (00010022_15560100)

□ Turbo

Эмулятор авто профиль.xml

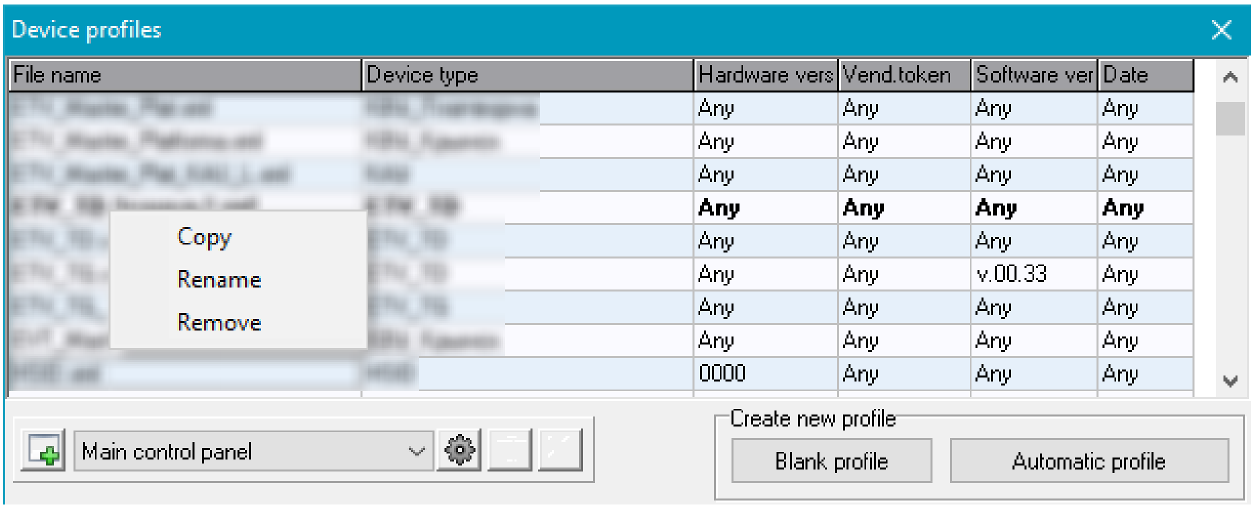
The device profile is bound to the device full name. The name is displayed above the ***Connect*** button and consists of the device type, hardware version, software version and software date. The values of the variables "Product code" (1018.02) and "Revision number" (1018.03) are displayed in brackets in hexadecimal form. Several profiles can be bound to the same device. Profile is selected from the ***profile*** list (to the right of the ***Update dictionary*** button).

7.4.1. Saving and deleting device profiles

Profile for a new device can be created in four different ways:
1) Copy an existing profile from the other, the most similar device, and then change the identification criteria for the new device (***Device type***, ***Hardware version***, ***Sostware version*** and ***Date***).
2) Create a new blank profile by clicking ***Blank profile*** button, and then fully customize it yourself.
3) Create a new profile by ***Automatic profile*** button. You should be connected to the device for which the profile is created. In this case, RTCON will try to automatically fill in all the profile elements, in particular, by the presence of the corresponding objects in the device dictionary:
5150/5150 – objects for working with a datalogger (oscilloscope);
2506 – objects for working with the faults log.
A control panel with elements determined by the presence of the following objects in the dictionary is also automatically created:
2502 – buttons for operational control commands (Start, Stop, etc.);
2532.01 – field for displaying faults texts;
250F.00 – field for displaying/editing the current time of device’s real time clock.

If the device dictionary contains a group #3 "References", then a field and a slider for editing a reference are created on the control panel for the first object of this group.

4) Find the profile for device of the same type, figure out criteria because of which it does not suit to the current device (e.g., different software date) and check ***Any*** box for this criterion. Then, this profile will be fit for such devices with any SW date.

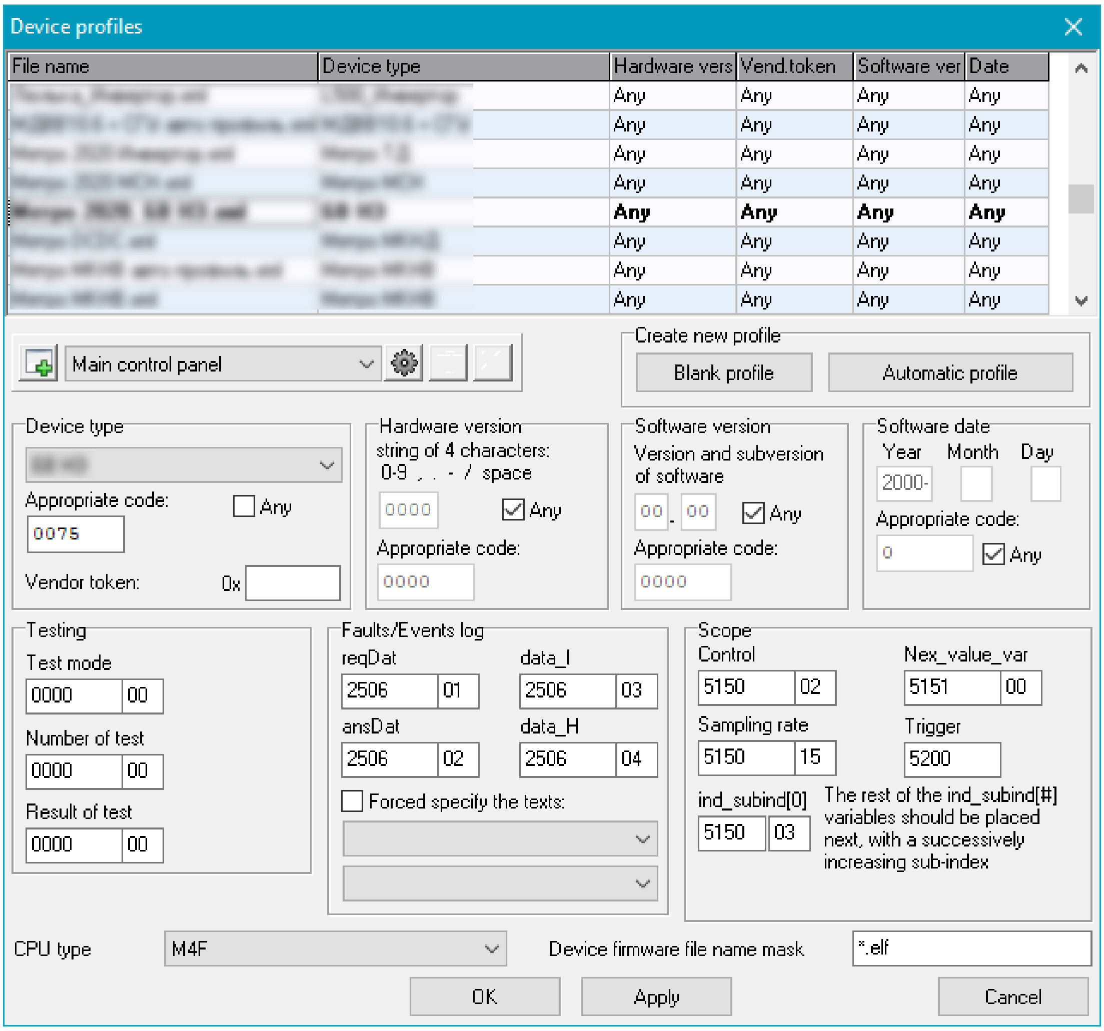


File name	Device type	Hardware vers	Vend.token	Software ver	Date
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	**Any**	**Any**	**Any**	**Any**
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	v.00.33	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	0000	Any	Any	Any

To copy, rename and delete the profile a context menu with the corresponding commands is used.

7.4.2. Profile main options setting

To work with profiles, open the ***Service*** menu and select ***Device profiles*** there.



Device profiles

File name	Device type	Hardware vers	Vend.token	Software ver	Date
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	**[redacted]**	**Any**	**Any**	**Any**	**Any**
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any
[redacted]	[redacted]	Any	Any	Any	Any

Main control panel

Create new profile: Blank profile | Automatic profile

Device type: [redacted]
Appropriate code: 0075 □ Any
Vendor token: 0x

Hardware version
string of 4 characters:
0-9 , . - / space
0000 ☑ Any
Appropriate code: 0000

Software version
Version and subversion of software
00 . 00 ☑ Any
Appropriate code: 0000

Software date
Year: 2000- Month: Day:
Appropriate code: 0 ☑ Any

Testing
Test mode: 0000 00
Number of test: 0000 00
Result of test: 0000 00

Faults/Events log
reqDat: 2506 01
data_l: 2506 03
ansDat: 2506 02
data_H: 2506 04
□ Forced specify the texts:

Scope
Control: 5150 02
Nex_value_var: 5151 00
Sampling rate: 5150 15
Trigger: 5200
ind_subind[0]: 5150 03
The rest of the ind_subind[#] variables should be placed next, with a successively increasing sub-index

CPU type: M4F
Device firmware file name mask: *.elf

OK | Apply | Cancel

Each profile is linked to the file located in the directory *RTCON\Settings\Profiles*. There is a list of these files at the top of the window where the file name is the name of the profile. In addition, it indicates which device fits this profile. Profile suits the device, if all four criteria match: the ***Device type***, ***Hardware version***, ***Software version*** and ***Date***. Using the mouse, you can select any profile whose settings you want to view/edit. These four criteria are given in the profile settings window (under the list of profiles):

Vendor token is an optional parameter, which is an 8-character hexadecimal number (for example: D467395A). This token is used to uniquely identify the device's text file. If the token is defined, the RTCON application will try to use the text file «coTextBase_D467395A.xml» (not «coTextBase.xml», used when there is no token) when configuring the profile of this device. In this case, for the correct display of parameter names and text enumerations in the device's dictionary at address 2600h.01h there should be an object of type UINT32 containing this token.

Device type is selected from the list. Each device type corresponds to a certain code. If the desired device is not in the list for some reason, you can set the code manually. You can also make a profile that is suitable for any device type by checking ***Any*** checkbox.

Hardware version of the device is a string of four characters. The numbers from 0 to 9 and special symbols: comma “,”, point “.”, hyphen “-“ slash “/”, and space “ ”. Similar to device type, each line corresponds to a specific code. Hardware version also can be set to ***Any*** by corresponding checkbox.

Software version consists of a major version number and a minor version number, separated by dot.

Date consists of a year, month and day. The year can't be set less than 2000, but the month and day may have zero values for compatibility with older devices, where the date is not set. All of these fields are collected into a single date code value, which can be set manually below.

Profile is suitable to the device only if all four criteria match. If the criterion is set to any (by the corresponding checkbox), it is ignored when checking.

Next in the profile window are the following settings: ***Testing***, ***Faults/Events log*** and ***Scope***. In all these fields are entered hexadecimal addresses of the dictionary objects through which these modules operate. The address consists of four-digit "index" and the two-digit "subindex" of the object. These numbers are the same for all devices in most cases and can be refined from the developer of a particular device.

Faults/events logs, in contrast to other modules include selecting of a text enumerations (***Forced specify the textst***). For some older devices, you need to manually specify the enumerations that are used as fault texts and events texts. For example, if fault texts do not appear at the opening of the faults bank from RTCON menu, you need to choose the necessary enumeration with these texts in described setting.

If the addresses are not configured or are configured incorrectly, the module will not operate.

CPU type – this setting is required for the correct operation of processor firmware programming (see section 6.8 above).

Device firmware file name mask – optional setting that restricts the name of file with device firmware (see paragraph 6.8 above). This option reduces the chance of accidentally loading the wrong firmware. The mask must match the format used in OS «Microsoft Windows ®». For example, if using the mask: «ETV_TD_v * .out» then files with the names «ETV_TD_v1.out», «ETV_TD_v2.5.out», etc. will be allowed to load.

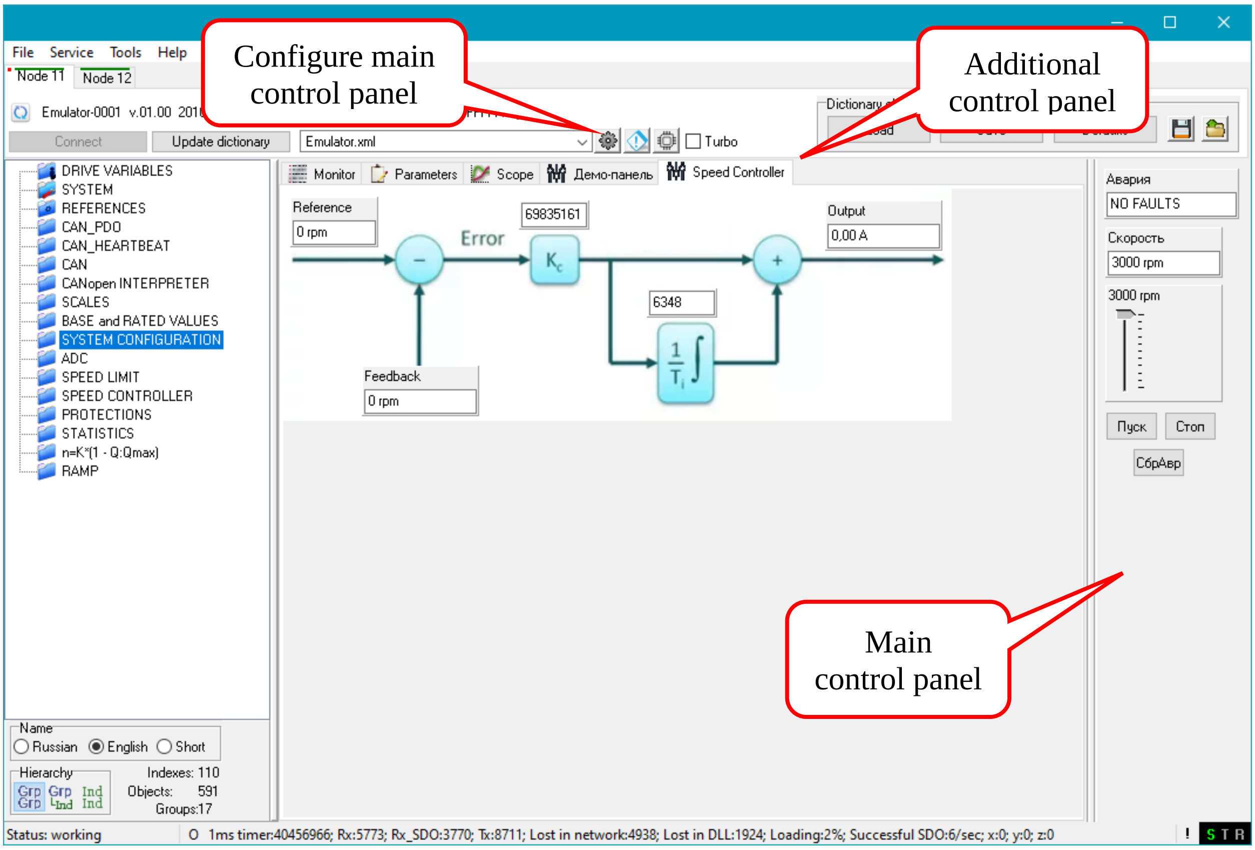
To save changes, you need to click ***OK*** or ***Apply*** button (***OK*** button will close the window after saving). If you select a different profile from the list without saving settings for the current, all changes will be lost.

7.4.3. Control panels

Binding variables of the dictionary to control panel window graphic elements allows you to conveniently display the current status of the device and send operational control commands.

There are two types of control panels – “Main” and “Additional”.

Main control panel is displayed on the right of the main window. ***Additional*** panels are displayed as tabs to the right of the “Scope” tab:



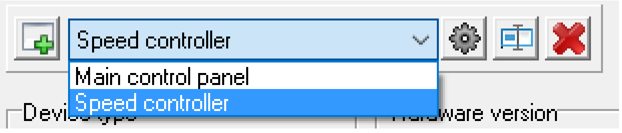
Configure main control panel

Additional control panel

Main control panel

Status: working O 1ms timer:40456966; Rx:5773; Rx_SDO:3770; Tx:8711; Lost in network:4938; Lost in DLL:1924; Loading:2%; Successful SDO:6/sec; x:0; y:0; z:0

These panels can be managed through the device profiles setup window (see section 7.4.2 “Profile main options setting” above):



The functions of buttons:

➕ – create additional control panel;

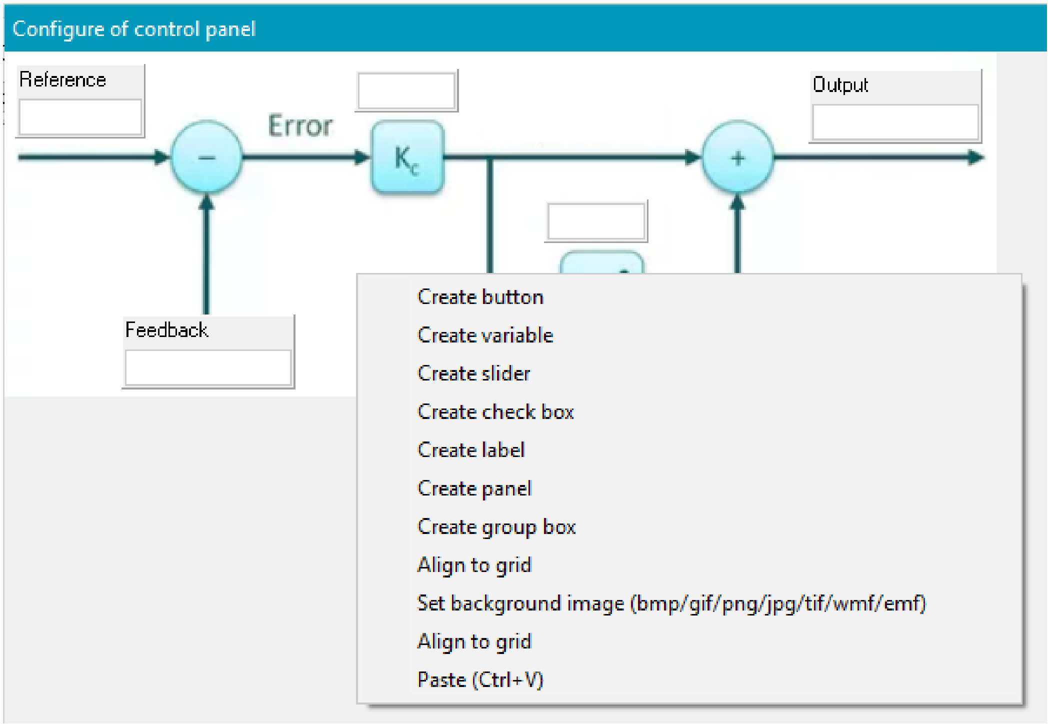
⚙ – configure control panel;

⎘ – rename control panel;

✖ – remove control panel.

The ***main*** control panel can not be renamed or deleted.

When the button ⚙ is clicked, the control panel layout editor will appear:



The following controls are available:

Button – allows, when you click on it, to send the specified value to the parameter. For example, the "Start" button can send the value 1 to the parameter corresponding for starting the device.

Variable – Displays any parameter of the object dictionary. It could be like speed, voltage, and current alarm or warning. If you want to, the variable can be allowed to change.

Slider – allows you to set the value of a variable using the mouse.

Check box – allows to view/change any bit in any integer parameter.

Label – can display either static text or the value of any parameter.

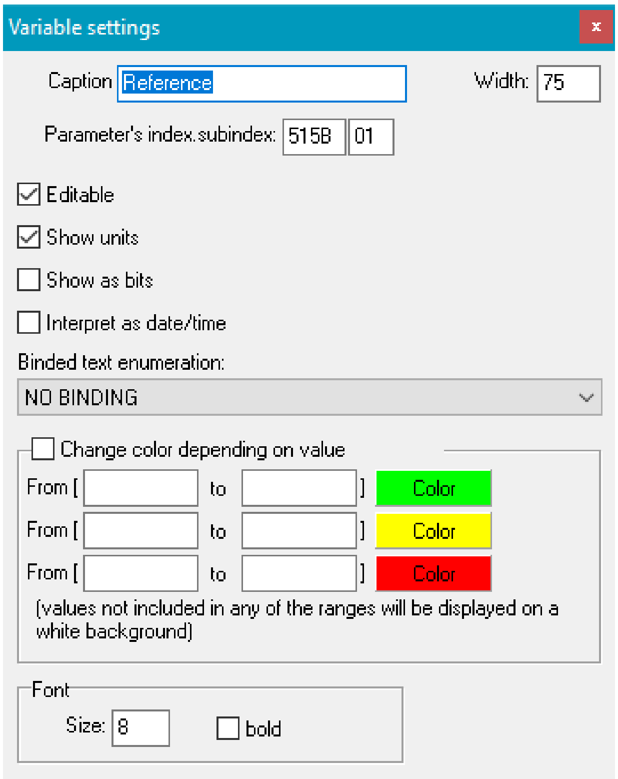
Panel – is a "container" inside which you can place other controls grouped according to some common feature. You can also adjust the accessibility/visibility of the panel depending on the value of any parameter. For the panel, you can also select a background image from the file.

Group – is a "container" with an optional text title, inside which you can place other controls, grouped by any common criterion.

For additional control panels, you can set the background image selected from the file.

To create any control element, call context menu by clicking right mouse button on a free space.

After creating an element you need to configure it – bind it to the desired object of the dictionary, set the name, size, etc. To do this, you need to click the left mouse button on created element, the settings window will open. For example, variable settings window:



Variable settings ×

Caption: Reference   Width: 75

Parameter's index.subindex: 515B 01

☑ Editable

☑ Show units

□ Show as bits

□ Interpret as date/time

Binded text enumeration:

NO BINDING

□ Change color depending on value

From [to] 🟩 Color

From [to] 🟨 Color

From [to] 🟥 Color

(values not included in any of the ranges will be displayed on a white background)

Font

Size: 8   □ bold

Caption – the text displayed as the variable header.
Width – here you can set the element width in pixels.
Parameter's index.subindex – is address of the dictionary object, which is bound to the variable.
Editable – option allowing editing of variable.
Show units – option to display units next to the value.
Show as bits – option to display the integer value of variable in binary format.
Interpret as date/time – option to interpret the value of variable as encoded date/time; it will be decoded and displayed as date/time string.
Binded text enumeration – if selected, instead of the numeric value of the variable the corresponding text from the selected text enumeration will be displayed.
Change color depending on value – this option allows to change the background color of variable depending on it's value. There are three ranges available to customize. The boundaries of the range are inclusive. This feature can be used for integer variables, IQ-variables and for variables which are displayed as text enumeration.
Font – here you can specify the size and style of font.

The control settings are applied either immediately (for example, font "bold" style), or by pressing the "Enter" key (for example, width). To close the settings window, press the "cross" in the upper right corner, or press the "Esc" key on the keyboard.

To move items on the control panel use mouse dragging: hold down the left mouse button and move it. You can also select any object by clicking the left mouse button while holding down the «Ctrl» key and then move it using WASD keys. If the "Shift" key is pressed, then instead of moving there will be the element resizing.

The selected object can be copied/pasted.

❗ **Note**

The ability to interact with elements on the control panel (pressing buttons, entering values, moving "sliders" and switching "flags") depends on the type of parameter to which the element is binded, and the current access level (see section 7.3 "Access level" above). For example, if a button is associated with "secret" parameter ("S"), and the current access level is "Service", then such a button will be disabled ("grayed"), since the parameters of type "S" can be edited only by "Master".