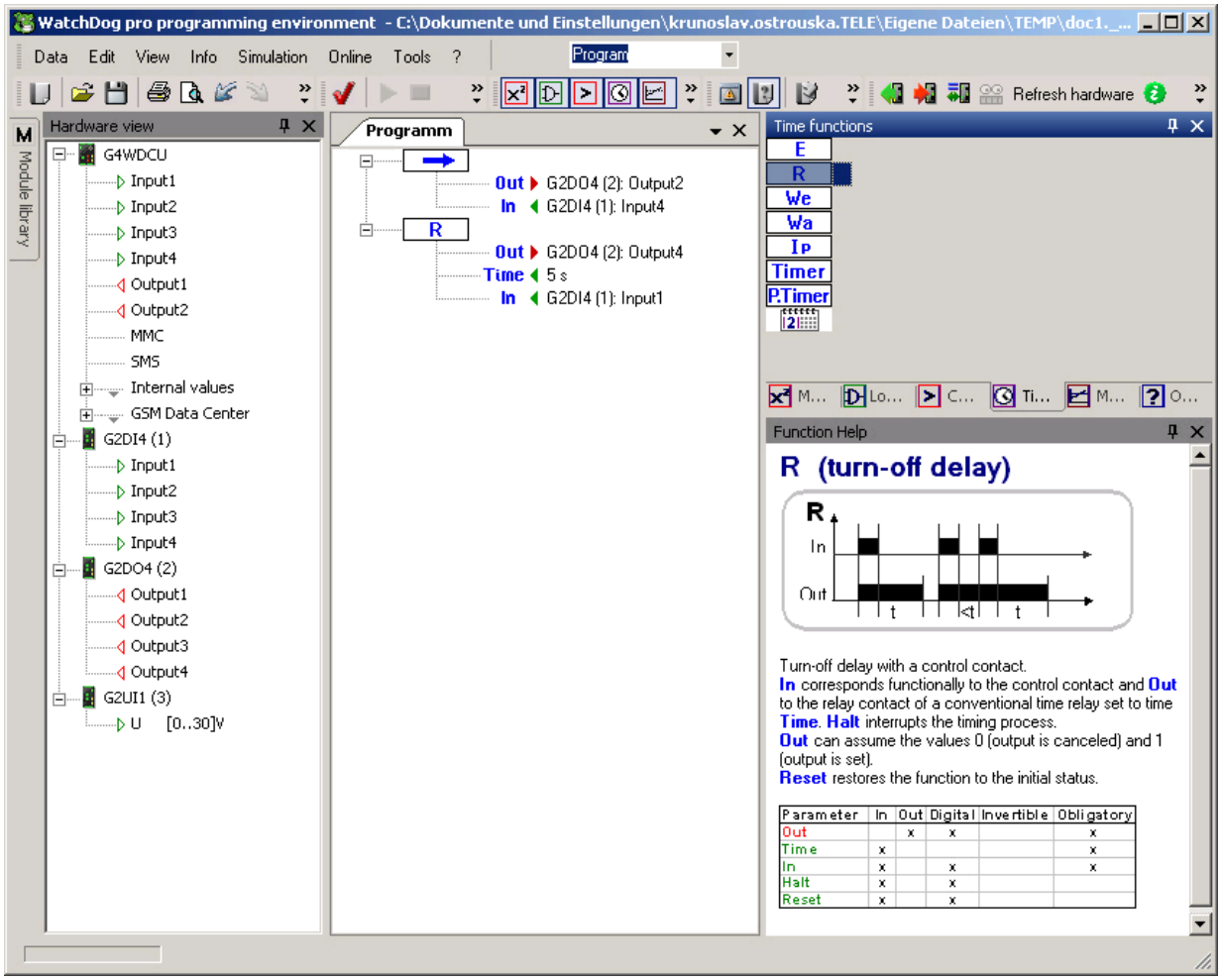
PROGRAMMING ENVIRONMENT WATCHDOG PRO

tele

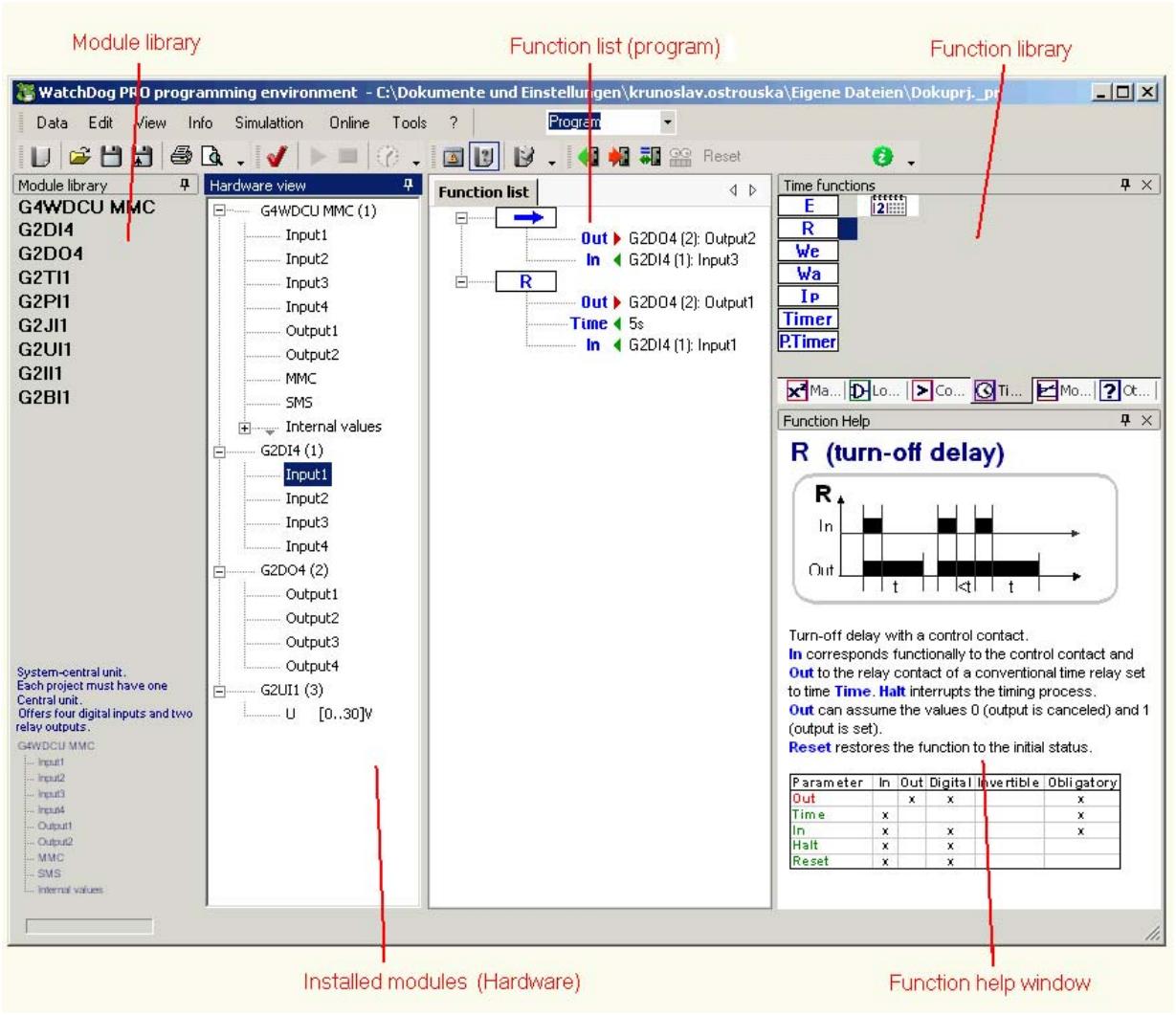
Technik Braucht Kontrolle

1. Overview

WatchDog pro Programming Environment is a program by means of which the WatchDog pro systems can be configured and their parameters set.
The latest version of the program can be downloaded from TELE internet page (www.tele-online.com).



1.1 Main features



1.2 Setting the hardware configuration

Compile the list of modules that are to comprise the finished control system.
Specify the module addresses and input configuration.

1.3 Creating an application

The system's response to different input statuses is defined using the program.
Input values can be converted, compared with each other and linked. Six libraries having different functions are available to users for this.

1.4 Simulating the system

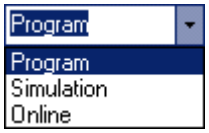
A project can be simulated using Programming Environment.
A separate simulation mode of Programming Environment has been provided for this.
The modules' input values can be changed and output and interim values observed while the program is running in simulation mode.
It is thus possible to check how the control functions even before any hardware has been obtained and installed.

1.5 Online system monitoring

In the online mode it is possible to observe the control status, program running, the measured values and the values of the registers.
The status of the inputs and outputs and of the registers can be controlled from the PC while the central unit is running (forcing).

1.6 Three operating modes of Programming Environment

To enhance the clarity of WatchDog pro Programming Environment during all project phases, the various functions have been spread over three operating modes:



Programming mode,
Simulation mode and
Online mode.

The operating mode can be changed over in the menu bar with the aid of the mode-control field:

The hardware configuration and function list can only be changed in the programming mode. No changes can be made to the program in the simulation or online mode.
Some online instructions (such as Read hardware) that may be helpful in creating the program are also available in the programming mode.

The application is simulated in the simulation mode.*
In the online mode it is possible to exchange data with the control and to control the system. All instructions that operate with the hardware will be rendered visible here or released for use.

* The application will first be checked on changeover to the simulation or online mode. Changeover to another mode will be prevented until any errors have been corrected.

1.7 Installation

The software requires Microsoft .NET Run Time Environment version V2.0, which will have to be installed if necessary.
Installation can be done for single or for all users.
When the software is launched, the program will check if the file extension '._PR' (project file) has been linked to Programming Environment and ask whether the entry should be made in the registry file so that the project file can also be opened in Explorer by double-clicking.
It will also ask for a user name, organisation and the product key.

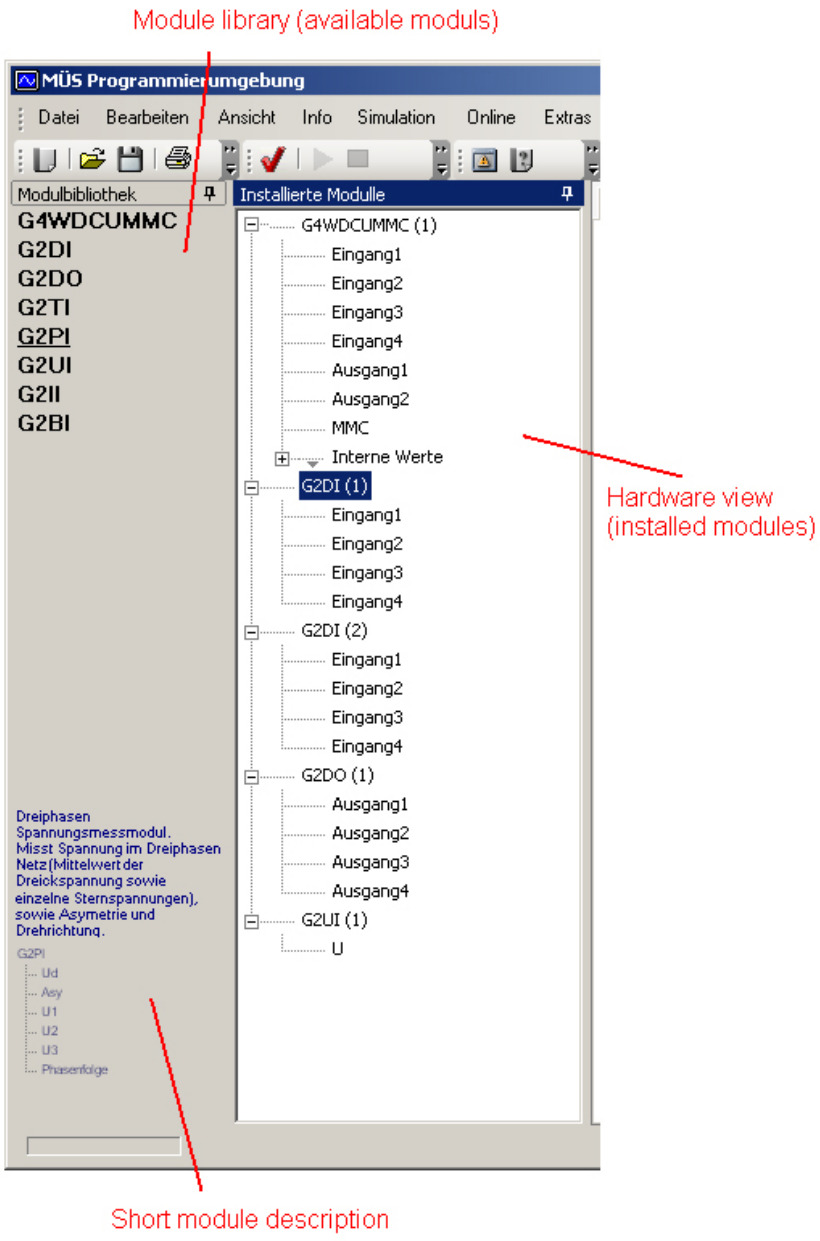
User rights (Windows 2000, Windows XP and Vista):
Main-user or administrator rights are needed on the local PC for installing and operating the software. People who only have "User" group rights can neither install WatchDog pro Programming Environment nor call it up.

1.8 System requirements

Operating system:	Windows 98/ME/2000/XP/Vista with .NET Framework V2.0400 400
Processor:	400 MHz
Main memory:	64 MB RAM
Interface:	RS232*

* A USB/RS232 converter can also be used. No connection can be set up to the WatchDog pro system without an interface, although the software can otherwise be used without any limitations (program can be created and simulated).

2. Setting the hardware configuration



A WatchDog pro system consists of a central unit and several expansion modules.
The basic system consists of a standalone central unit and its inputs and outputs that can be used in the program.
Maximum system capacity is limited by the power consumption of the individual expansion modules and the cycle time required.
Up to 99 expansion modules of the same type (such as power-measuring modules) can be connected to the central unit's bus (local or remote BUS).

The WatchDog pro modules used in the system are listed in the hardware view. In the programming mode, additional modules can be added to the system from the module library. The hardware configuration can also be read out from the connected system.
The module addresses can be entered in the hardware view and the expansion modules' measuring ranges set there.

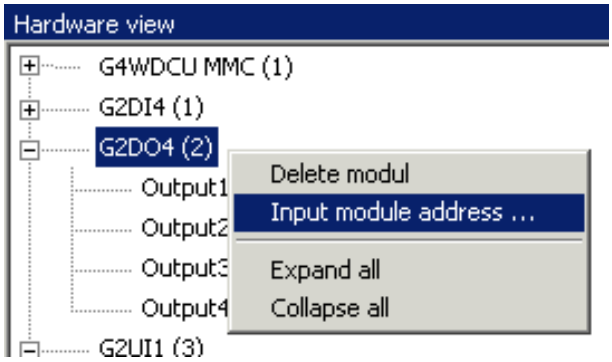
2.1 Detecting hardware

The hardware configuration can be read out from an existing system. The expansion modules detected when this is done will then be displayed in the hardware view.

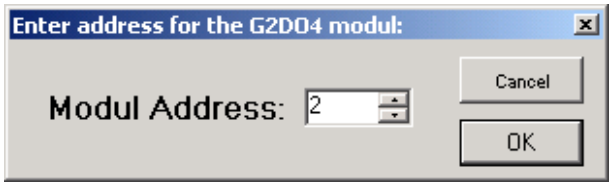
The instruction used for this is Refresh hardware (see Refresh hardware)

2.2 Adding a module and specifying the module address

A new WatchDog pro module can be added to the project from the module library. This is done by dragging the module into the Hardware view list or double-clicking on the module.
The module library window is only active in the programming mode.
The module is automatically assigned the next free bus address (appears in parentheses next to the module name).



The module address can be changed over to the value actually set on the module using the context menu entry Enter module address.



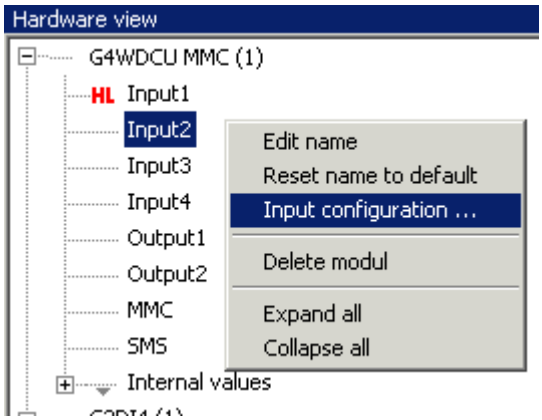
The central unit always has the module address 1 (only one CU possible in the system).

2.3 Configuring the digital inputs

The digital inputs of the central unit (G4WDCU) and digital input modules (G2DI4) can operate in three modes:

- Standard (the status of the digital input is sent through at the time of interrogation),
- High Latched (the input notes if the input was activated between two interrogations),
- Low Latched (the input notes if the input was deactivated between two interrogations).

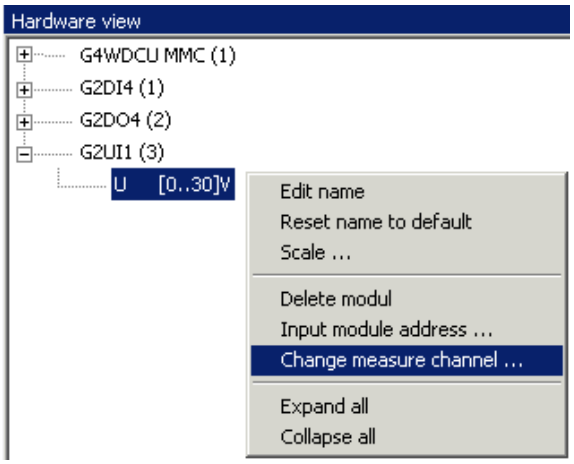
This can be set for each input using the context menu item Configuration of input.

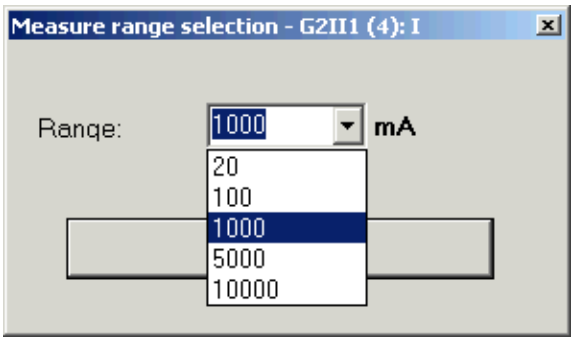


2.4 Configuring the analog inputs

Three measuring ranges are usually available (cannot be used simultaneously) in the case of the single-phase voltage and current expansion modules (G2UI1 and G2II1).
Here, the measuring range must be selected where the measurement variable is actually applied (is physically connected), because no meaningful values will be measured at the wrong measuring input.

So that the measuring input can be set, activate Change measuring range... in the module's context menu and select the required measuring range in the measuring range selection dialog.





The measuring-range dialog offers a list of measuring ranges that are possible across all module variants.
This means that measuring ranges will also be listed there which a specific type of module does not necessarily need (many expansion modules come in a variety of versions having different input ranges).

The G2II1 (single-phase current-measuring module), for instance, is available in a 5A and 10A version each having three measuring inputs: 20mA, 1A, 5A and 100mA, 1A, 10A.

The control cannot be started if a measuring range has been set that does not exist on the expansion module. The control will report the critical error ("measuring range not found!") and assume the error condition.

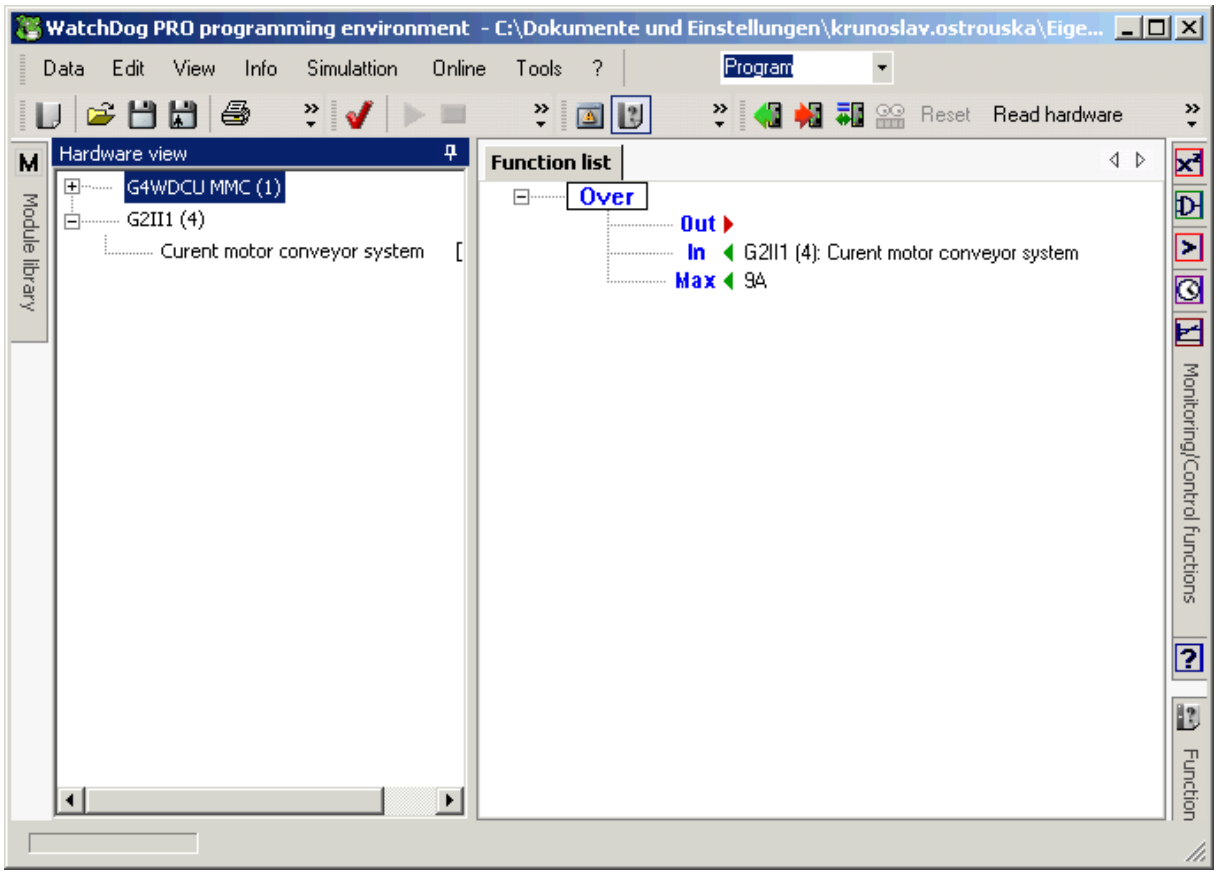
The measuring input of the G2UI1 and G2II1 (single-phase current and voltage) can be scaled to any value range. The unit of measurement can also be flexibly changed.
Scaling is useful when variables are measured via measuring transducers. The measurement variable's real value can be used and not the value supplied by the measuring transducer.
The scaling dialog can be accessed from the input's context menu.

The current transformer coefficient can be entered in the case of the G2JI1 (3-phase current) and G2BI1 (power) modules. That will then be applied to all measurement variables of those modules (to all three phases of the G2JI1 and to the power of the G2BI1, too).
The current transformer coefficient dialog can be accessed from the module's context menu.

2.5 Renaming the hardware parameters

The expansion modules' inputs and outputs can be renamed at any time. You do this using the context menu entry Edit name.
This allows the various parameters to be given names relevant to your application.

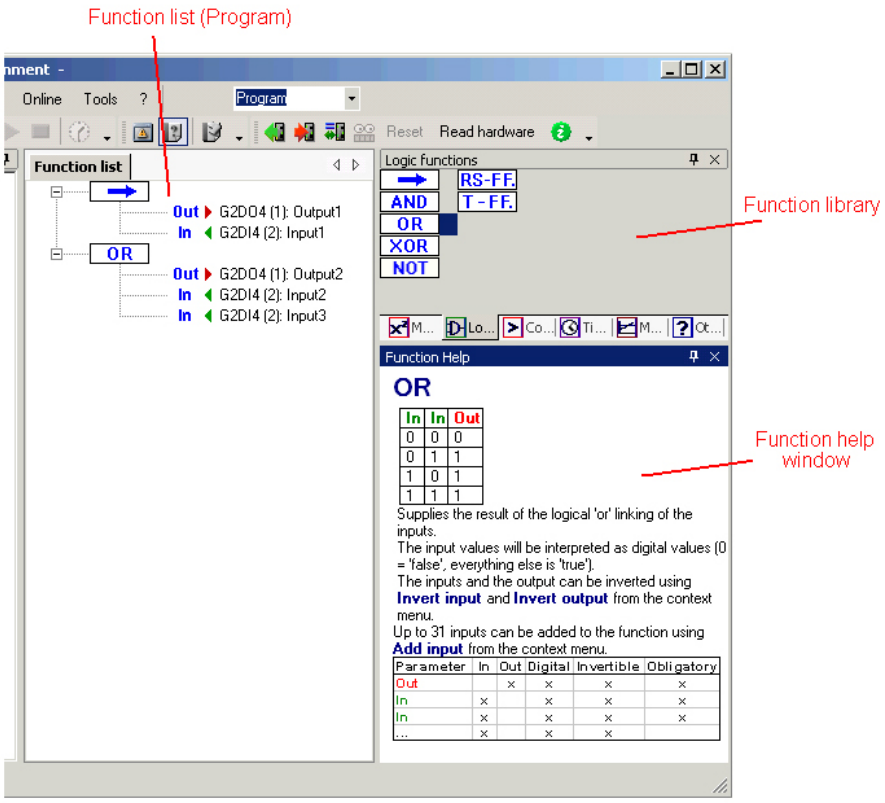
For example, instead of the 'I' of a G2II1 current-measuring module, you can use the name "Current motor conveyor system".



That name will then be used in every function link, making the program much easier to read.

You can reset the changed parameter names to the default at any time using Reset name.

3. Creating a program



This is where the control's response to the input values is defined.
The functions comprising the program are shown (listed) in the 'Function list' window.
All actions and conversions can be implemented using different kinds of functions.

This kind of programming was developed specifically for inexperienced users unfamiliar with typical control-related terms such as instruction list, contact plan, and function plan.
The type of programming used here will allow anyone to create the program for a WatchDog pro system.
Developing the applications will be all the less problematic for those who have experience of IEC601131 or higher languages.
This type of programming is not suitable for complex tasks, though. It lacks certain items such as function call-ups, arrays, character strings, etc., that can often be found in the instruction list and which make it easier to create complex programs.

The program flow is cyclic (typical for controls), meaning the program will be executed right from top to bottom (one program cycle). This process will keep being repeated while the central unit's mode selector switch is still set to 'Run'.
The input values are read at the start of each program cycle. The results are written to the outputs after the program run.
The cycle time is the sum of the times needed for reading the inputs, executing the program, and writing to the outputs. This value is important because it describes the control's response time. The cycle time is also important for the controllability of a system controlled using a PID controller.

3.1 Adding a function

There are six different function libraries:

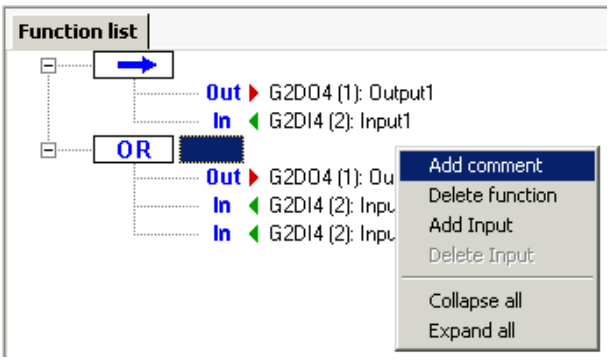
- Arithmetic (basic operations, modulo, increment, decrement, minimum, maximum, absolute value and rounding)
- Comparisons
- Logic (And, Or, Exclusive Or and Negation)
- Time functions (tripping and turn-off delay, single-shot-leading edge and trailing edge, flasher and timer)
- Monitoring and control functions (minimum, maximum and window monitoring, start-up monitoring, PID controller)
- Other functions (multiplexer, edge forming, counter function, function for safe condition and program-flow control functions such is conditional jump).

The library windows are only active in the programming mode and can be activated and deactivated in the View menu or Function library toolbar.

In the programming mode you can add a new function (to the program) by either double-clicking on the function in the library or dragging the function from the library window into the Function list.

You can change the sequence of the functions in the program at any time by dragging the function to the position you want
The functions or parameters can also be copied using the mouse (drag the function(s)/parameter(s) with the Ctrl key pressed).

The buffer can also be used for editing the function list. A function or parameter from the function list can be copied to the buffer along with the parameter and pasted in again afterwards (see menu item Edit).



A comment can be added to each function (context menu entry Add comment).

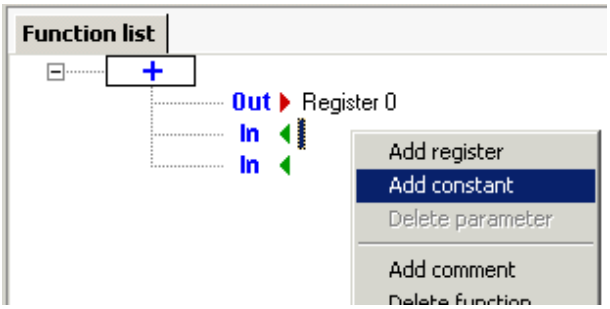
An editable text field where a description or notes can be added will then appear next to the function.

3.2 Assigning parameters

After inserting a new function into the function list you can specify the function parameters.

- The hardware values can be copied to the function list from the hardware view by dragging them with the mouse.
  It should be noted that it is not possible to use an input value at the function's output or an output value at its input.

- The constants are entered using the context menu item Insert constant.
  That will open a text field that can be edited.



The constant can be changed at any time.* Text can be entered again by double-clicking on the constant or using the context menu item Edit.

- Registers can be used at virtually all function inputs and outputs.

  You can produce a new register using the context menu item Insert register.

The 'Variable tracking' function has been integrated into Programming Environment (View menu) to make the parameters easier to work with. When Variable tracking is active, Programming Environment will highlight the marked parameter in color wherever it occurs.

* Changes to the programming are only permitted in the programming mode.

3.3 Constants

The constant values can be entered with prefixes and units.
It is thus possible to use values such as 750 µA or 4.72 kW as function parameters.

The following SI units will be recognized:

Unit	Name	Calculations
A	Ampere	= V/Ohm = W/V = VA/V
Ohm	Ohm	= V/A
s	Second	= 1/Hz= J/W = J/VA= Wb/V
°C	°Celsius	
V	Volt	= A/Ohm= W/A= VA/A
W, VA	Watt, VoltAmpere	W = VA= V*A= J/s *
Hz	Hertz	= 1/s
m	meter	= J/N
bar	Bar	
Pa	Pascal	
T	Tesla	
J	Joule	= Nm= Ws
H	Henry	= Wb/A= Ohm*s
Wb	Webber	= Vs
N	Newton	= J/m
l	Litre	
kg	Kilogram	
%, Number	-	

* W and VA are equivalent. No comparisons will result in error messages or warnings. Addition and subtraction applied to mixed values containing W and VA will produce VA values.

For all functions where a comparison takes place (Over, Under, Window), the input values must be of the same magnitude (using the same unit). An error will otherwise be reported when the syntax is checked.
If the units are not used, that may result in warnings such as:
"The input value is not compatible with the input. A value having the unit XY is expected".
As this is only a warning and not an error, it will not, despite the warning, prevent you from transferring the created application to the central unit or simulating it.

All kinds of prefixes can also be used for the constants.
These prefixes are recognized as follows:

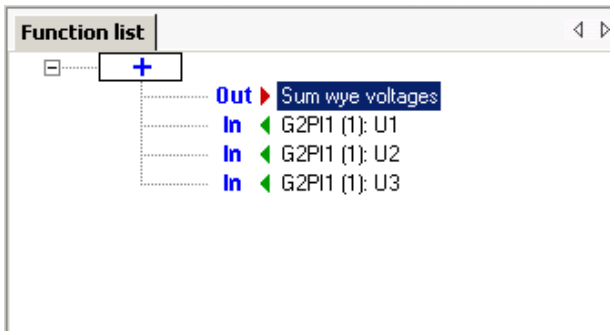
SI prefix	Name	Value
p	Pico	10^{-12}
n	Nano	10^{-9}
µ	Micro	10^{-6}
m	Mili	10^{-3}
c	Centi	10^{-2}
d	Deci	10^{-1}
h	Hekto	10^{-2}
K	Kilo	10^{-3}
M	Mega	10^{-6}
G	Giga	10^{-9}
T	Tera	10^{-12}

Both the units and the prefixes exist only in the programming environment. Values without units are used for computing in the central unit.
More complex unit conversions such as:

1 H = 1 V*s/A
or
1 W = 1 A^2*Ohm are not supported.

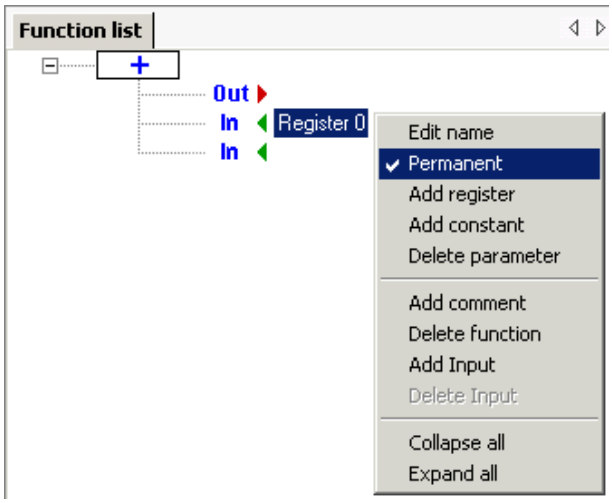
3.4 Registers

A new register is produced using the context menu item Insert register.
It will be called 'Register X' (X = consecutive number) and can be renamed at any time by double-clicking on the register name or the context menu entry Edit name.



A certain number of registers can be configured as non-resetting on voltage failure (permanent register). The values of permanent registers will remain stored in the non-powered condition and be available to the system after rebooting.

To configure a register as non-resetting on voltage failure, you must activate the Permanent option in the register's context menu.



The values of permanent registers can be transferred to the project by means of the Read permanent register entry in the Online menu.
The permanent register values will be transferred automatically when the program is read from the central unit.

Only a limited number of permanent registers are available.
Information about the register resources used (number, availability) is fed out in the Project info window. The 'Project info' window can be called up from the Info menu or using the icon in the toolbar.
See Windows and menus/Info.

3.5 Number format

The floating decimal point form of numerical notation was chosen for WatchDog pro to minimize the complexity of program development for the user.
All data is computed * and fed out using numbers with a floating decimal point (16Bit mantissa and 8 bit exponent with base 10, = $\pm 3.2760 * 10^{\pm 131}$). That means you no longer have to worry about the types of variables.
The floating decimal point format offers a very extensive number range with somewhat less accuracy (compared to float in the PC world):

Number range: $\pm 3.2760 * 10^{\pm 131}$
Accuracy: 16 bits or:
[-32760..32760] with total accuracy,
±32760 to $\pm 3.2760 * 10^{\pm 131}$ with 99.969% accuracy.

In most other control languages you need to differentiate between bool, short, int, long, float, double, unsigned short, unsigned int, etc. However, having to differentiate in this way would increase proneness to error and impact negatively on legibility. The wide variety of types is one of the factors that make programming a difficult and abstract task, a job for experts.

Not so with WatchDog pro.
The uniform data format also makes programming much easier.

There are, though, some situations where the characteristics of the number format still have to be taken into account. It is the accuracy of number interpreting that usually gives rise to a situation like this, and it will arise in WatchDog pro just as with other controls when 'floating-point' types of numbers are used in computing.

If, say, you want to add or subtract low numbers to or from high numbers, then the low number will not be reflected in the result because the result cannot be shown with sufficient accuracy.

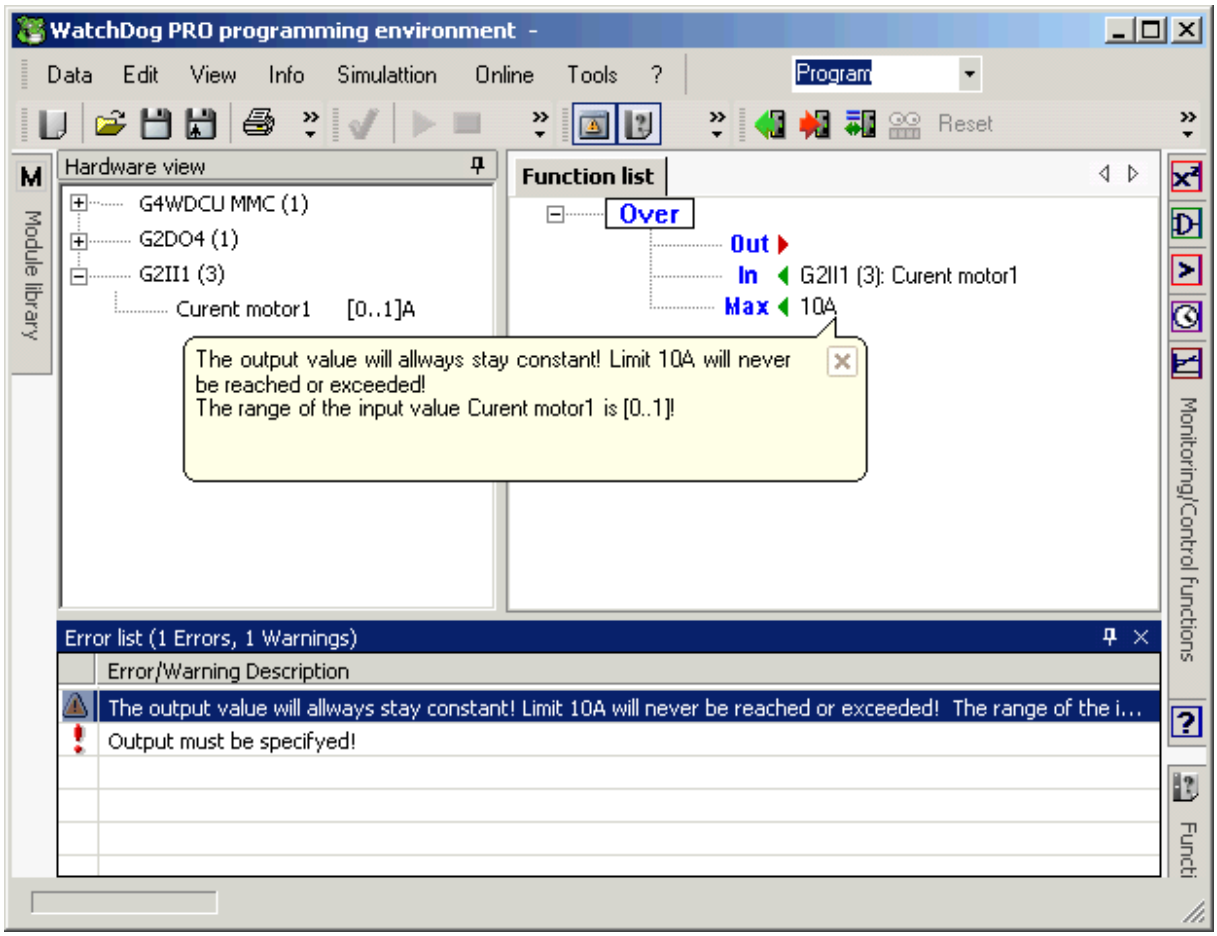
WatchDog pro offers some functions which in certain situations can make this less of a problem.
There are cases where one is to be added to or subtracted from a number that can become very large. If the computation concerned is carried out using + or -, then the value cannot exceed the limit of ±32760. That is because of the loss of data due to the lower accuracy of the float format.

The INC, DEC and Counter functions operate internally using a number format with a greater degree of accuracy. With the INC, DEC and Counter functions the integers can be processed with no loss of data in the $\pm 2.1474 * 10^9$ range (32 bits).
It must, though, always be borne in mind that the accuracy of these functions will be reduced in the result (still a floating decimal point number).

* Many functions compute internally with a greater degree of accuracy (see below).

3.6 Checking the program

Once you have created the program it can be checked by Programming Environment (using).



Programming Environment will check the program automatically on changeover to the simulation or online mode or before it is transferred to the central unit.
All reports produced during the check will be fed out in the error list.
If you click on the report, then a window containing a detailed description will be fed out along with the actual location of the error.

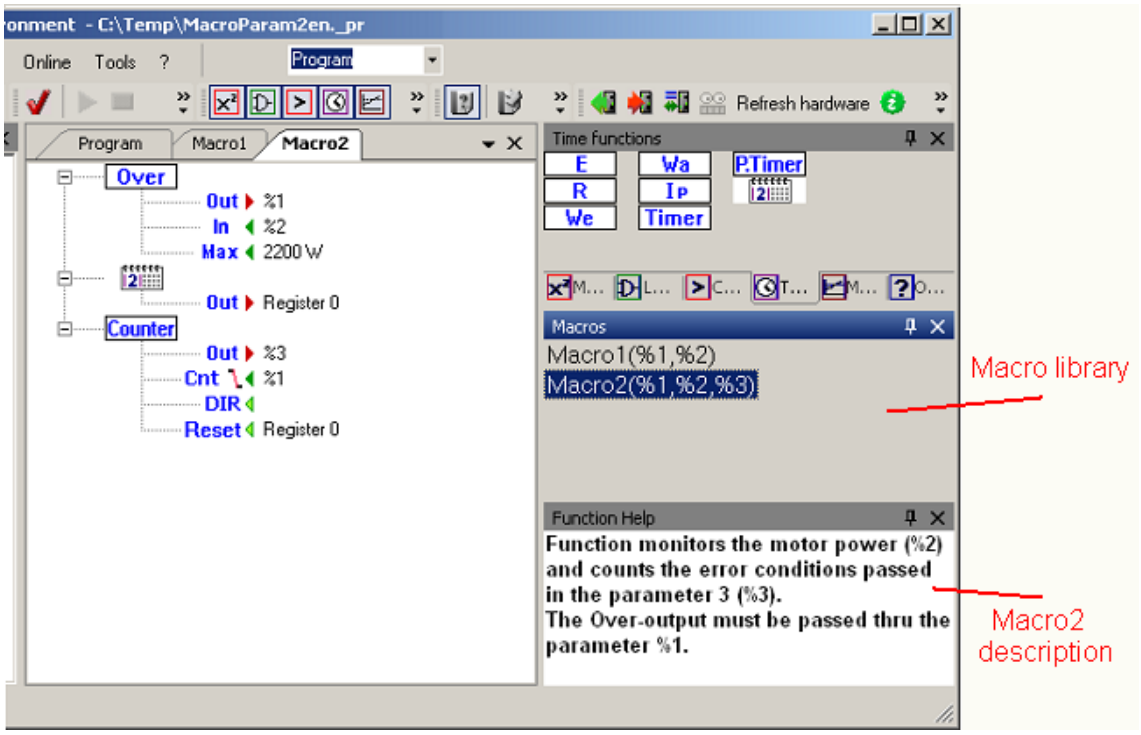
The reports are as a basic rule assigned to an error or a warning message.
The error messages can be identified by ! and require corrective measures.
A faulty program can be neither simulated nor transferred to the central unit. Changing over to the simulation or online mode will therefore be prevented.
The warnings ⚠ indicate where there may be a logic error in the program. Although warnings do not demand corrective measures, a check is nonetheless recommended. The program can still be simulated or transferred to the central unit with warnings active.

3.7 Self defined functions (macros)

Function groups that appear multiple times in programs can be moved to macros and called as self-defined functions in other program sections.

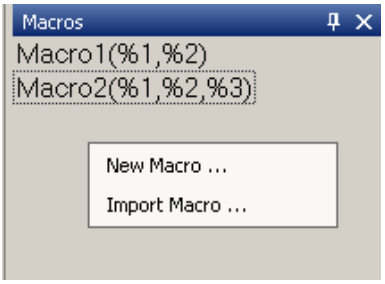
These functions can be passed input and output parameters.
You can assign descriptions to the function itself and the individual parameters.
The macros can be exported and imported into other projects.

There are no restrictions on the use of library functions in macros except for the S-Window function where the display of data in the graph (simulated or online) is not possible because the macro may be invoked multiple times and for this reason is not clear as to which instance is desired. The S-Window can always be used in macros with different input parameters but always with identical limits.

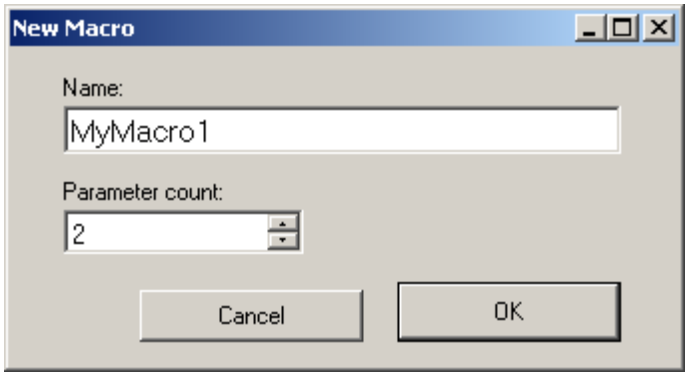


3.8 Create macro

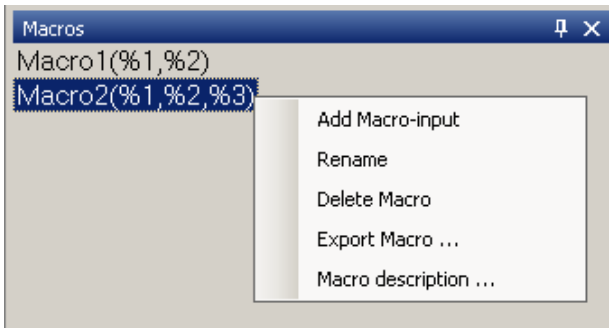
A new macro can be created with New macro... (execute in the menu or in the Macro Library Window context menu).

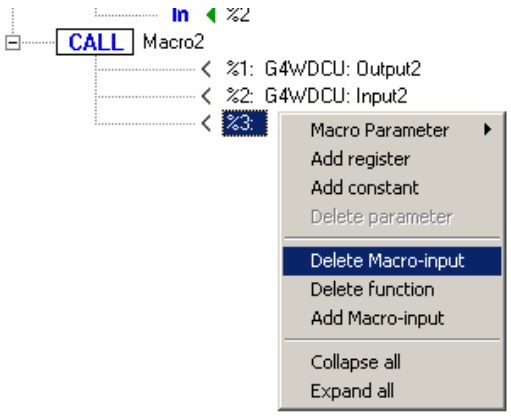


A dialog window with two entry fields appears.
The name of the macro is to be entered in the first window, and the initial number of parameters is to be specified in the second.

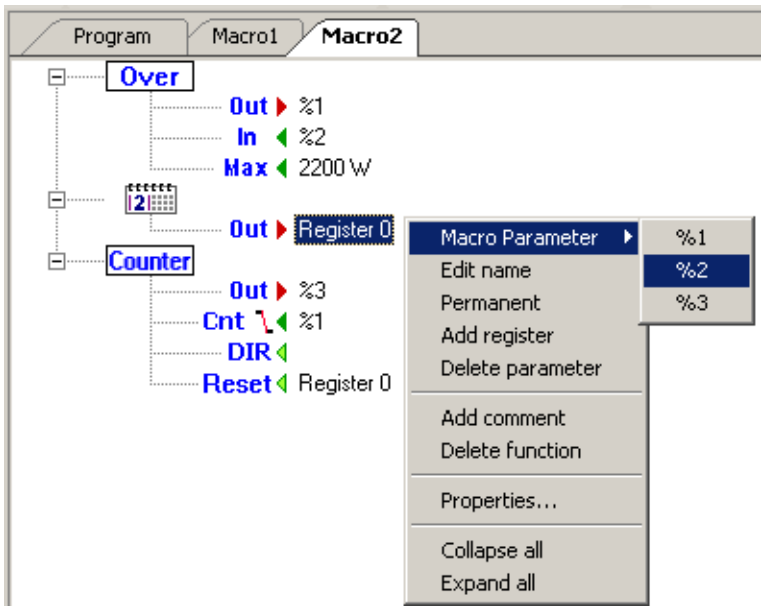


The macro can be renamed later.
More parameters can be added to the macro, or existing ones can be deleted (see also the context menu of the macro call in the program window).



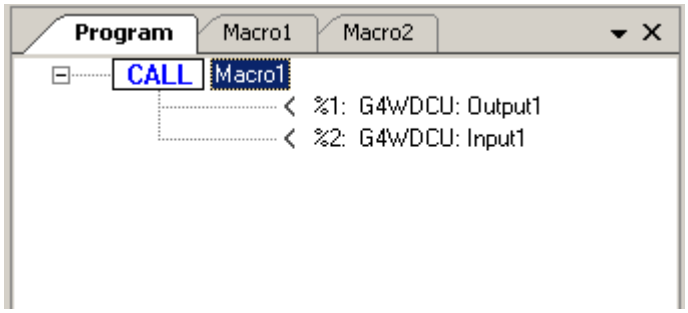


The macro parameters start with percent signs and are numbered consecutively.
They can be used as inputs and outputs in all functions in the macro.
The plausibility check will be conducted during the syntax check.



3.9 Utilizing macros

The macro can be inserted into the program by double clicking in the macro library window or by dragging and dropping.



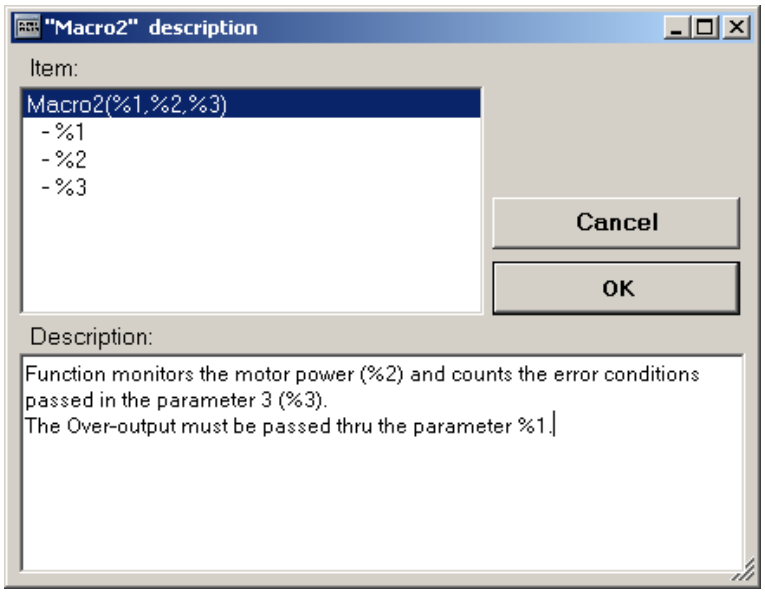
Input and output variables can now be passed to the macro inputs.
The CU data logger outputs MMC and SMS cannot be passed in macros!
The parameter plausibility check will otherwise be conducted during the syntax check.

Circular Call:
Through the use of the macros, a situation can arise where a macro calls itself or calls itself through multiple macros (macro1 calls macro 2, which in turn calls macro1).
This is not permitted and leads to a "circular call" error during the syntax check.

Note: Every macro call generates a complete copy of all functions and locally defined registers contained in the macro.
While reading the project from the CU, all macros are lost and the contained functions are included in the main program.

3.10 Describing the macro and the parameters

A description of the macro and its parameters can often be useful and increases the readability of the program.
The descriptions can be entered in the dialog window, which can be invoked in the macro library window with the context menu option Macro description...

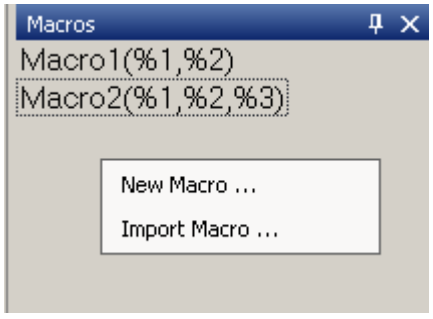


The object that is to be described can be selected in the list box above, and the description can be edited in the text box below.
The function description will be displayed in the function help window if the macro is selected and also appears in the printout.

3.11 Exporting and importing the macro functions

Self-generated macro functions can be exported as necessary out of the project in which they are defined.
The function Export macro... opens the File Saving dialog to save the macro file with the "._mc" file extension.

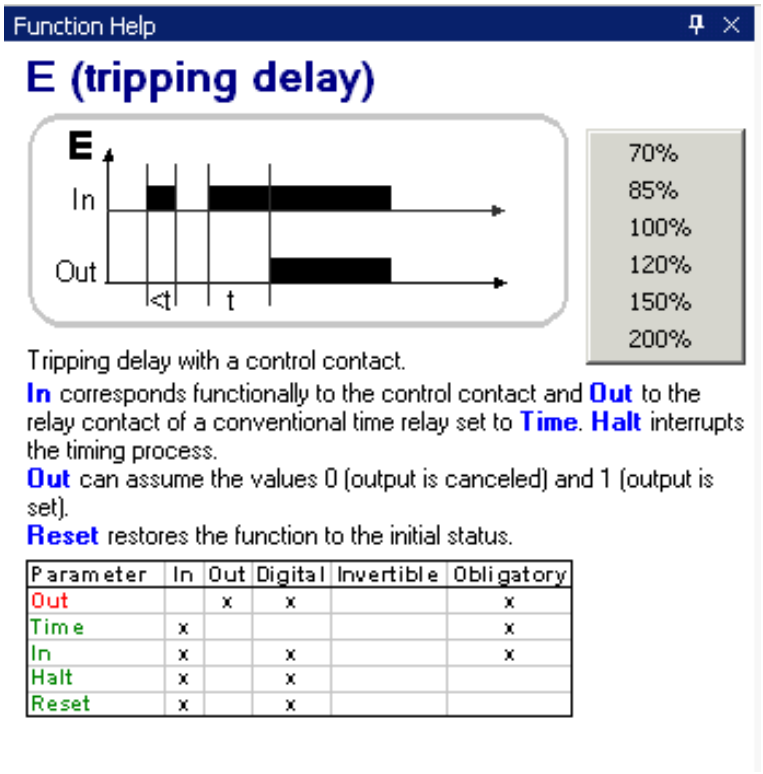
The exported macro can be imported into another project with the Import macro... command. It becomes part of the project. All changes to the function in the project do not affect the macro file.



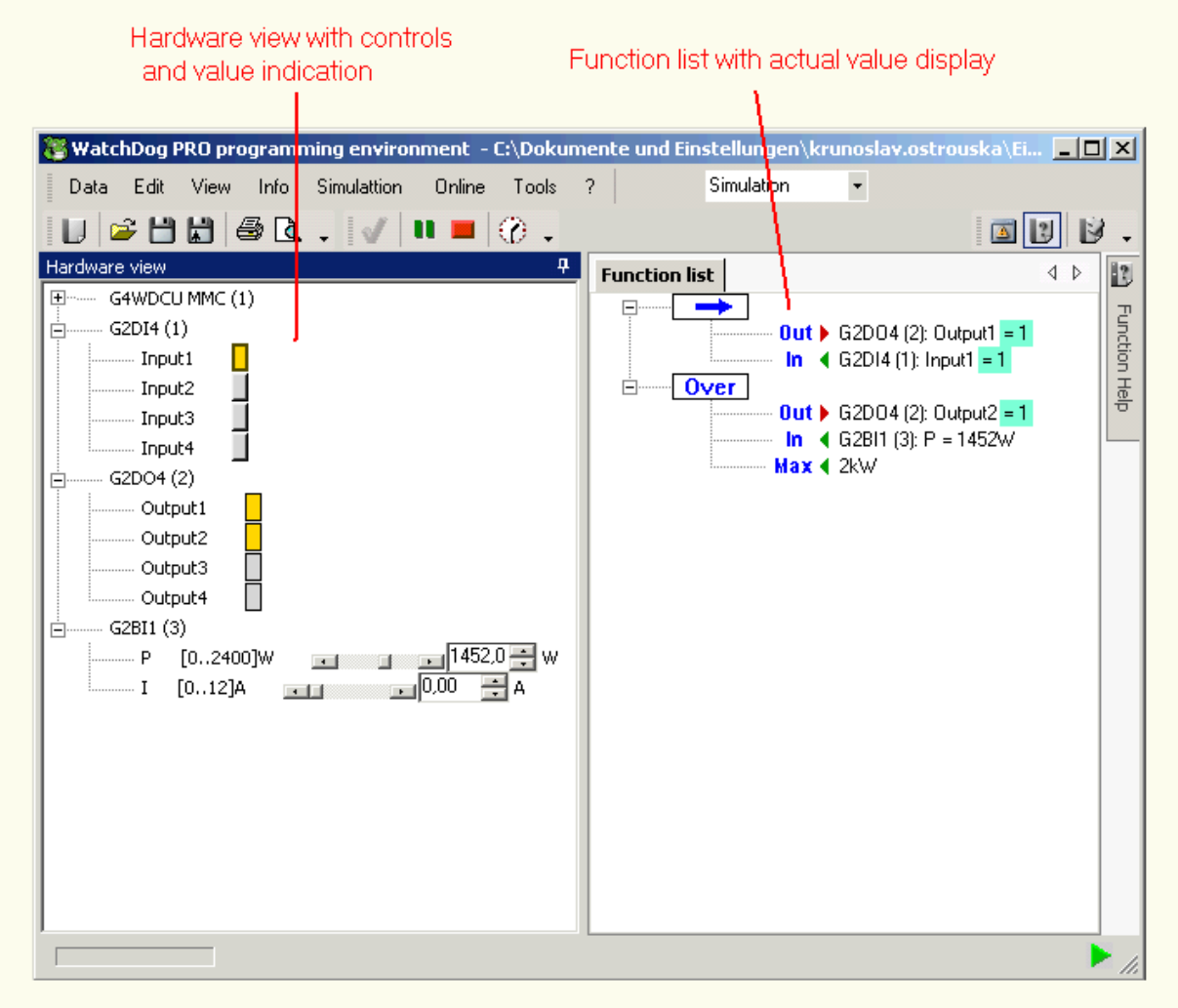
3.12 Function help

Interactive help with the currently marked function is displayed in the Function help window. This window can be activated in the menu or View toolbar  .

The size of the view can be adjusted. To do this, right-click once in the function help window:



4. Simulating



Programming Environment offers you the option to subject the WatchDog pro system that has been created to a virtual test. With flexibly settable input values, you will be able to see how the program that has been produced will react.
Programming Environment is changed over to the simulation mode for this purpose.

Alongside the hardware inputs, control elements are displayed with which the input values can be set, while the hardware outputs and registers are given fields showing the actual values.

The program can no longer be changed in the simulation mode.

4.1 Controlling simulation

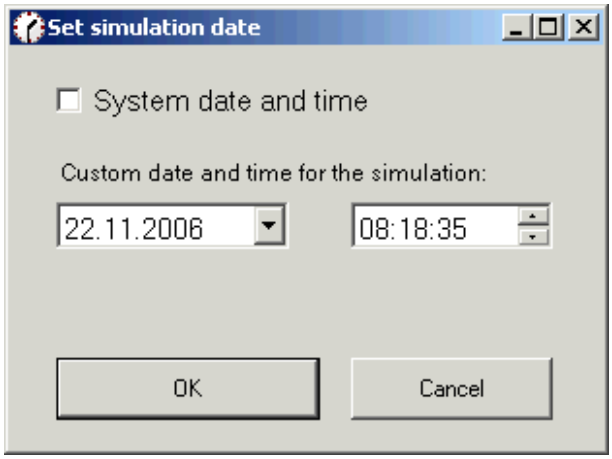
When you have changed Programming Environment over to the simulation mode, the Start button that launches the program run becomes active in the toolbar.
The program you have created will then be executed cyclically, just as in WatchDog pro's central unit. The cycle time is fixed at 100 ms. That is not the same as the real cycle time and is the only thing setting this run apart from a 'normal' program run.

Tip: The menu item Simulation/Start is also active in the programming and online modes. Clicking on it will switch over to the simulation mode and launch the simulation.

You can pause the program run using the Pause  menu option/button. Start will resume it.

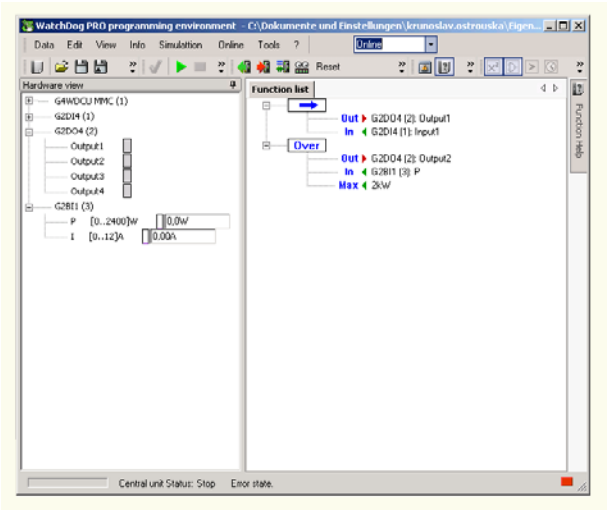
Stop will cancel the program run and reset the values.

If you have used the calendar function in your program, then changing the time and date within the simulation can yield some useful results.
To do this, activate the menu item Simulation/Set date and time for the simulation... or the same button  in the Control toolbar. Deactivating the 'Accept system time' option will cause the input fields for date and time to become active. Here you can set the time required for simulation.



The date and time can be changed while the program is running (take care with negative time changes).

5. Online mode



In the online mode the connection is set up to WatchDog pro's central unit, whose status is continuously interrogated and fed out (in the status bar or as an error message).
It makes it possible to control the central unit as well as to display the values of the registers and measured values of the connected WatchDog pro modules.

Only in the online mode is it possible to control the central unit using Start, Stop, and Reset instructions or observe the program and hardware values using Watch.
Forcing the values in the control is also only possible in the online mode (the context menu of the parameters only contains the Force option in the online mode).

Some online instructions can only be accessed in the programming mode:

- Read program from the central unit
- Write program to the central unit
- Compare program
- Read permanent register values
- Read hardware
- Display central unit values (status, firmware version etc., no program values!)
- Set date and time in the central unit
- Set PIN code

Any 'online' instructions that change the central unit's status will only be accepted by it if its mode selector switch is set to RSto (Remote Stop) or RRun (Remote Run).
Only the central unit's Remote Mode will allow the operator to issue instructions or change the program in the central unit.

All reading instructions such as Read program, Compare program or Watch (observe values) can be executed independently of the position of the central unit's mode selector switch.

5.1 Start, Stop and Reset

These three instructions control the central unit's program run. The central unit's mode selector switch must be on one of the Remote settings (RRun or RStop) for these instructions to be carried out.

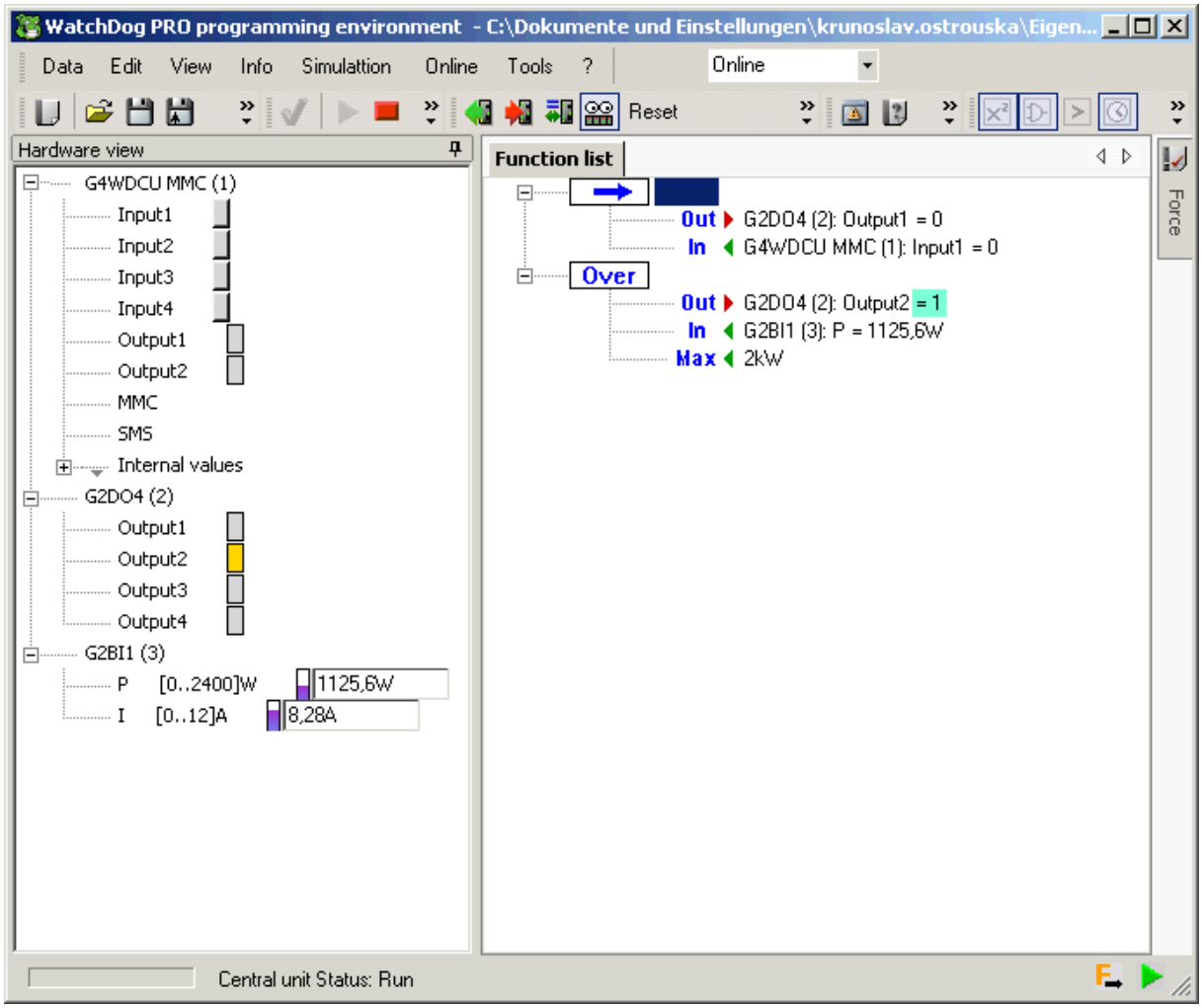
The program run is launched using Start  .
Stop  will halt it and reset all the register values, except those of the permanent registers.
Reset will reset the central unit. The central unit will then be re-initialized and assume the status set using the mode selector on the front of the central unit. The permanent registers will remain as they were on Stop.
The 'RESET' button on the central unit can also be used for re-initializing.

The central unit's actual status (run/stop) is displayed in the status bar.

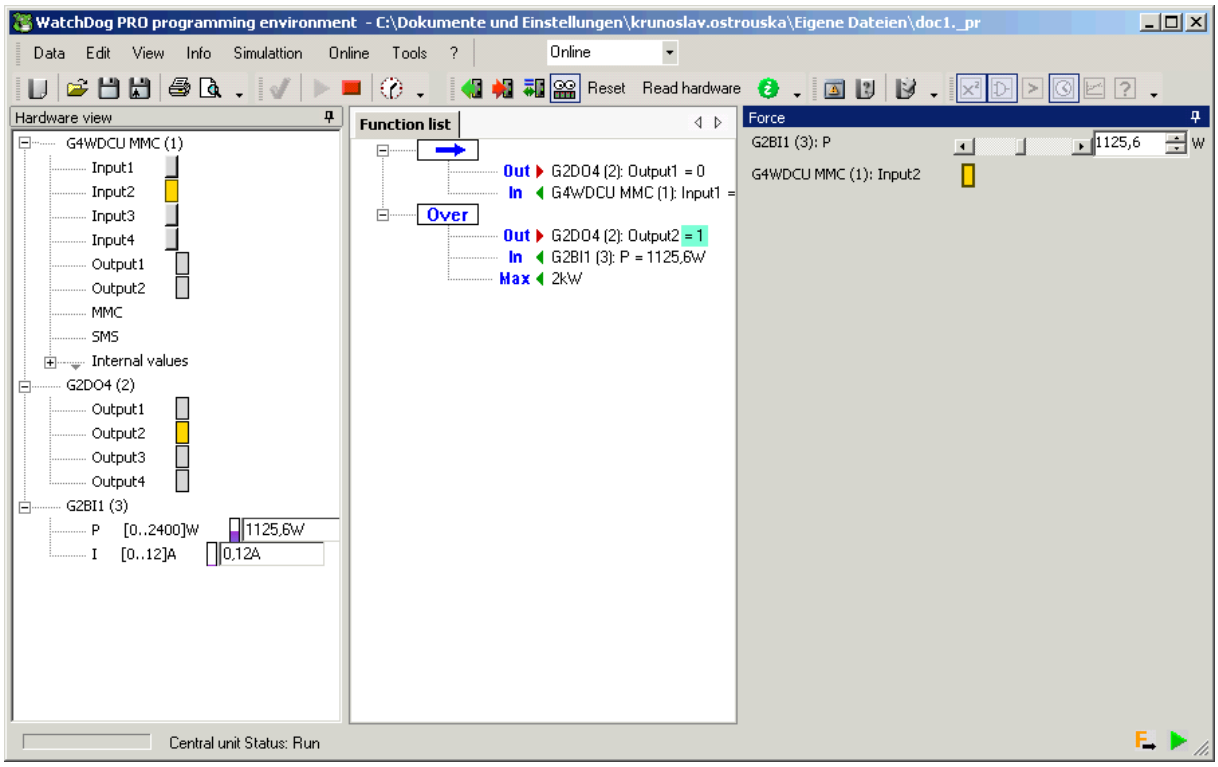
5.2 Watch

When Watch has been activated  , all hardware values as well as those of the registers will be continuously read by the central unit and displayed with the aid of Programming Environment. That will give you an overview of the input values actually measured and of the register values.

Watch can be deactivated again using the same toolbar button or the menu option.



5.3 Force values



While the system is being tested (during power-on, for instance), or during troubleshooting following a fault, you often want to be able to manipulate certain measured values so the control's reactions can be checked. That is usually very complex, if not impossible.
It was for this purpose that forcing was implemented.
It allows you to set any input or output value of the control to any values within the variable's value range, then observe the control's response.
For example: The motor current is 1.3A. You can use Programming Environment to 'force' it to 5A and observe how the control reacts on deactivation.

To force a parameter, activate the Force option in the parameter's context menu (that can be done in the online mode both in the hardware view and in the function list).

When you no longer wish to force a parameter, deactivate the parameter's Force function context menu. Alternatively, you can also activate the context menu option Remove entry from the force list in the Force window.

Up to ten values can be forced simultaneously.

5.4 Read program from the central unit

The program stored in the central unit is read out using Read program from the central unit . The program can then be edited or archived.
This will be the only way you can alter the program if you no longer have the original project file. The default texts will be used as the program read out of the central unit does not contain the names of the variables.

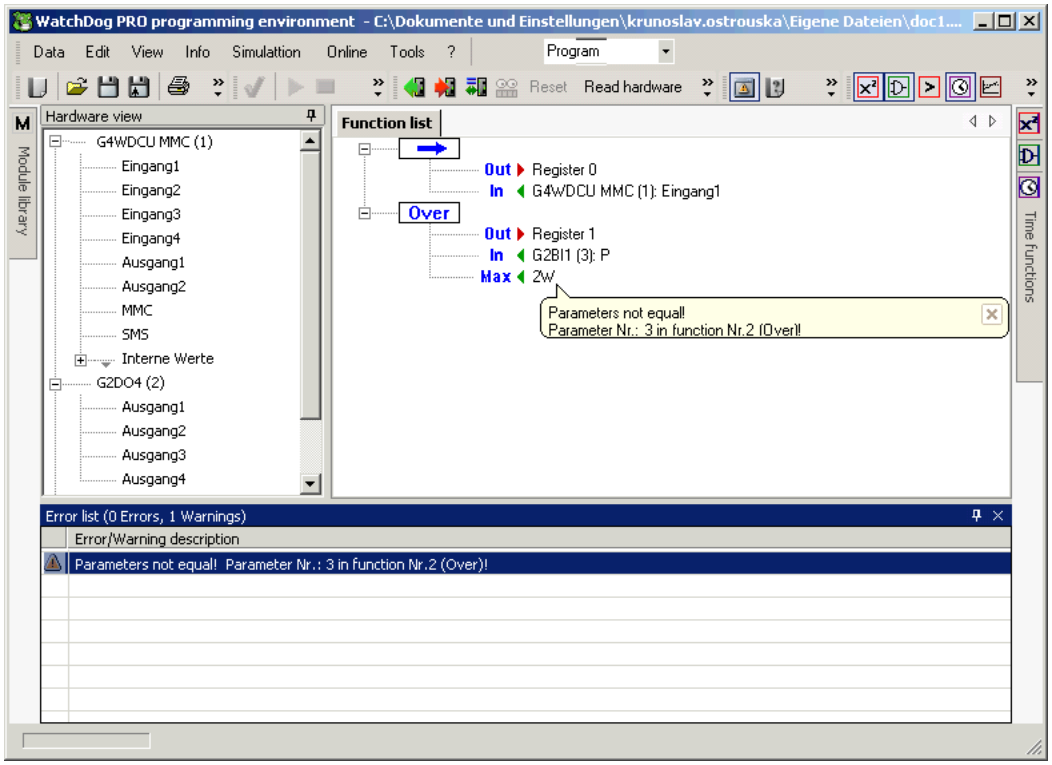
5.5 Write program to the central unit

You can transfer the finished program to the central unit using the instruction . The program is checked before being transferred. It cannot be transferred if found to contain errors. A program already stored in the central unit will be overwritten.

If permanent registers are used and you want to read out their values from the control, the values can either continue being used or reset to zero.

5.6 Compare program

This option is available to you if you want to know if the program in the central unit matches the open project's program.
The open project will first be checked and the program then read out from the central unit. Only then will they be compared with each other.
After this comparison a summary (same/different) will be displayed. If the two programs are not the same, their differences will be listed in the Error list window.



If any differences are detected in the hardware configuration, the comparison that has started cannot be resumed after the error has been reported.

5.7 Read permanent register values

The values of the permanent registers are read from the control and transferred to the project.

The permanent registers will be transferred automatically if the program is read from the central unit using Read program from the central unit  .

5.8 Refresh hardware

This option lets you update the hardware configuration of the WatchDog pro project.

The linked central unit will search for connected modules.
When the search has been completed, additional hardware modules will be added to the hardware list. The configuration of the existent modules will be reviewed and, if needed, the input channel will be set to, on the hardware, existent value.
For the non existant modules the user will be asked to delete or to leeve it in project.

5.9 Display central unit values

This option  opens the window in which different values within the control are displayed (not to be confused with Watch). As with any online-mode activity, the PC has to be linked to the central unit. The displayed information is updated once a second.

The information is spread over four registers.
The System values register shows the position of the central unit's mode selector switch and the central unit's actual status (Run/Stop). Also displayed here are the central unit's temperature, voltages, BUS currents, and the measured maximum cycle time.
The maximum cycle time can only be determined while the program is running. It will be reset each time the central unit is stopped.

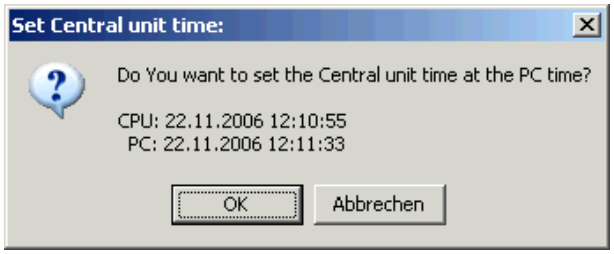
The Version register displays the software version and serial number of the central unit.

The information about the MMC used is displayed in the MMC slot register, along with the maximum recording rate and number of lost datalogger entries - so only error messages specifically for the flash card and datalogger.

The Status register displays any errors that may be present.
Errors are logged in a log file that can be displayed using Show log. Purge log will clear the contents of this file.

5.10 Set date and time in the central unit

This option lets you set the central unit's time and date to the PC's time/date.



5.11 Set PIN code

The central unit can be protected with a PIN code.
The default PIN code is 0000. The central unit is open when this PIN code is in use. The unit can be read from at any time without entering a PIN and written to with a new application.
If you want to prevent this you can use another PIN code for the central unit. The central unit will then be protected against unauthorized access.

Select Set PIN code from the Online menu to open the PIN dialog where CU PIN code can be set. Trying to control a PIN protected CU or performe data exchange, require the input of the correct code in a pop-up window. Entering right PIN Code unlocks the CU, which stays unlocked until next CU reset.

If you have forgotten the PIN code for your central unit you can overwrite the old PIN code. For this you will need the master PIN code available from TELE on request.

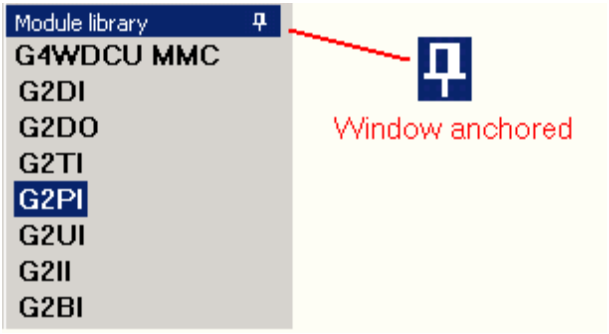
6. The windows

Programming Environment contains several windows, most of which can be configured. The appearance of individual windows sometimes depends on the mode in which the environment happens to be operating. Quite a few are only visible in specific modes.

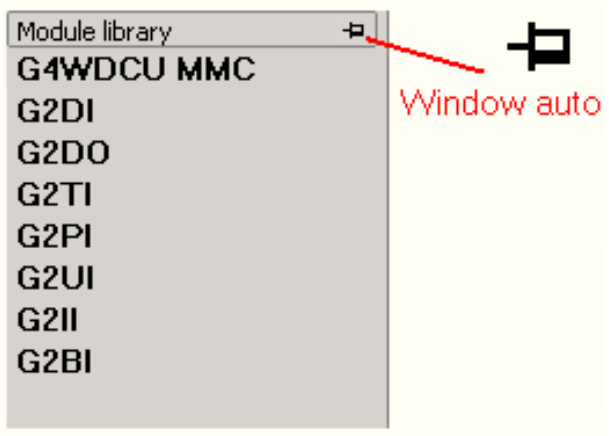
What applies to all windows is that they can be moved to any position you want. Most can be anchored along any edges or remain unanchored in Programming Environment.
Any number of windows can be anchored to an edge. In the default setting, the function libraries (six windows), for instance, are all anchored together along the right-hand edge of the window.

To anchor a window somewhere else, drag the main window bar to the required edge. A frame outline will appear there showing you the position the window will occupy when you release the left mouse button. If you just leave the window anywhere at random, it will remain unanchored (if that status is possible for the window concerned). Double-clicking on the main bar of the window will release its anchorage or re-position it to its original location.

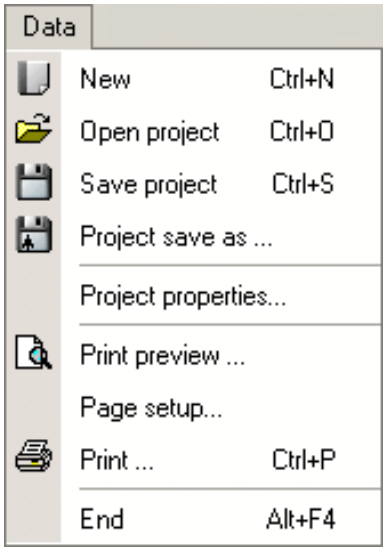
By clicking on the anchor icon in the main window bar, any anchored window can be changed over to the auto mode (icon changes to). The window will then be minimized into its docking edge. Pointing the mouse cursor at the window bar will open the window again.



That will free up space for frequently used windows, while hidden widows can still be accessed at any time.

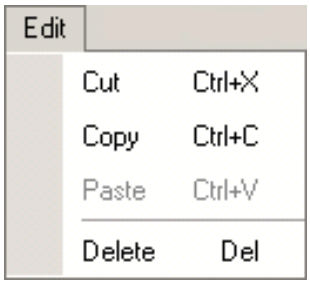


6.1 File menu



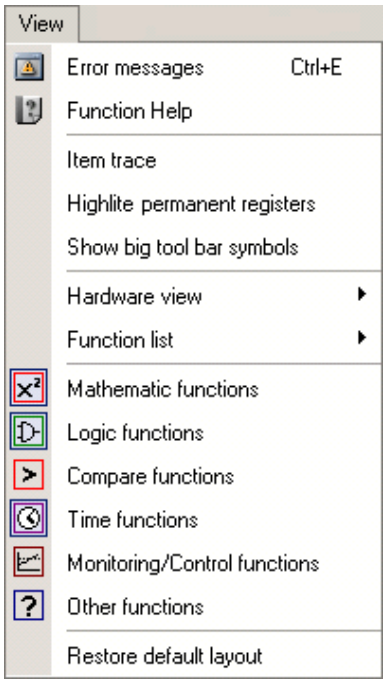
- New generates a new blank project.
- You can load a previously saved project using Open project.
- You can save the current project with Save project and Save project as... .
- Project properties... opens the project properties window.
- The title, version, editor, company name and corporate logo can be specified in the General register. This information will be used in printouts.
- Information that is to be printed out can be specified in the Printout details register.
- Print preview shows how the printed page will appear, you use Page setup to define the print format and, finally, you use Print to print out the project information.

6.2 Edit menu



Edit menu contains four instructions for the buffer.

6.3 View menu



- All these windows can be displayed and masked out as and when required using Error window, Function help and Function library windows.

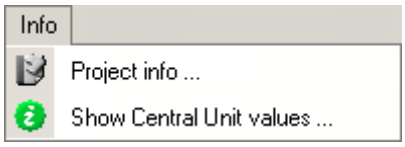
- When the Variable tracking option is active, the marked variable will be highlighted wherever it is used in the function list and hardware view.
  That means, for example, that if you mark input1 of the central unit in the program list or hardware view, the text will be given a bright green background wherever input1 appears in the program.

- The Large icons option allows you to change the size of the toolbar icons.

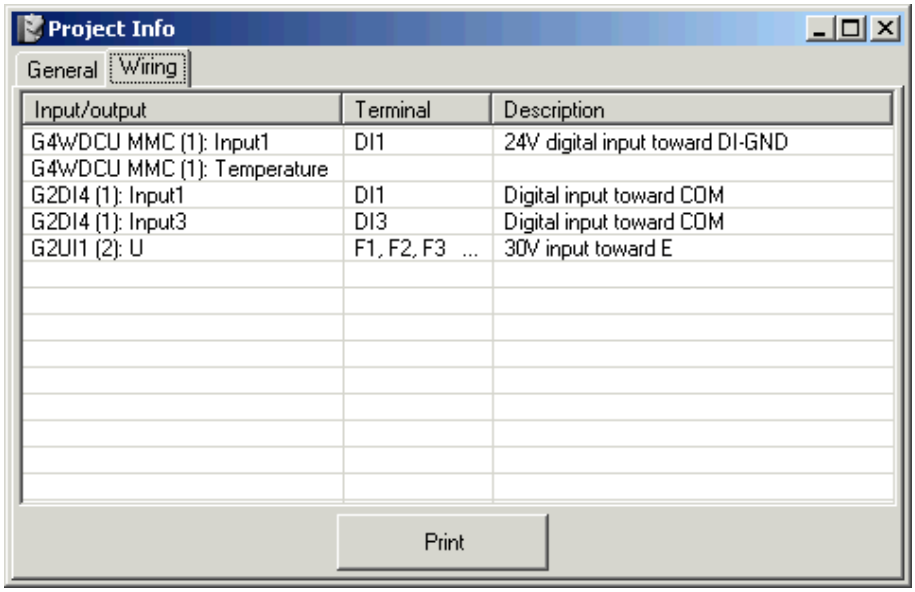
- Reset window layout resets the positions of all windows and of the toolbars to the default settings.

- The Hardware view and Function list menus contain the entries Minimize all and Restore all, with which you can minimize or maximize all the modules and function links in these windows.

6.4 Info menu

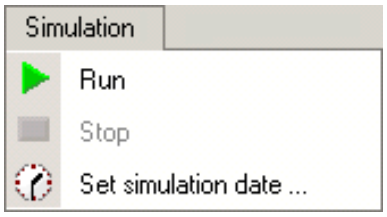


Project info, opens the project info window.
The General register contains information about the register and hardware resources used (number, availability) and the computed cycle time.
The Wiring diagram register is where all the hardware inputs and outputs used are listed along with their terminal reference and a description:



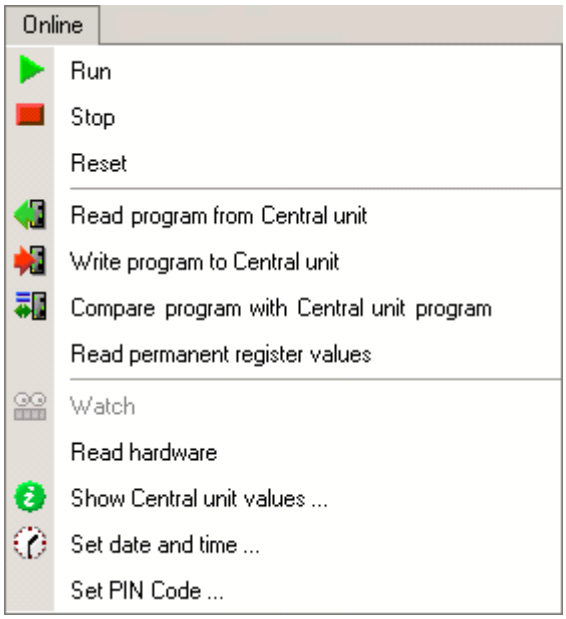
Display central unit values, can also be activated in the Online menu and displays internal data of the central unit (see Display central unit values).

6.5 Simulation menu



Contains options for controlling the simulation (see Controlling simulation and Simulating).

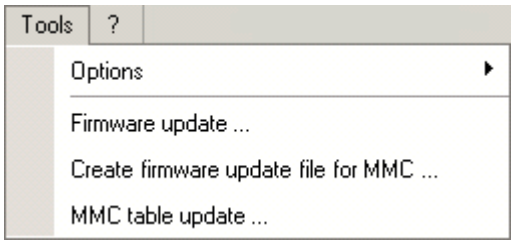
6.6 Online menu



All instructions that control the online mode are grouped here.

- Start, Stop and Reset **Reset**
- Read program from the central unit
- Write program to the central unit
- Compare program
- Read permanent register values
- Watch
- Read hardware
- Display central unit values (status, firmware version etc., no program values!)
- Set date and time in the central unit
- Set PIN code

6.7 Tools menu



The Tools menu contains:

- Update firmware...
- Create firmware file for updating MMC
- Update MMC table…
- Options.

- The central unit's firmware can be updated as and when required using Update firmware... .
- The firmware can be written to the MMC using Create firmware file for updating MMC The central unit's firmware can then be updated later using this.
- The table of the multimedia cards can be updated using Update MMC table... .

IMPORTANT NOTE: Updating the central unit's firmware is a critical process. There is a risk of a total central-unit outage if the power supply is interrupted during updating.

There are four submenus in the Options:

- Communication... activates a dialog field in which you can specify the serial interface or Ethernet server (DNS name or IP address) used for the link to the control. The required communication type must then be activated (e.g. The "Communication via RS232" checkbox).
- The Confirm delete option can be deactivated to suppress the query "Do you really want to delete the module(/function)?" that appears when various objects are being deleted.
- Development Environment's language can be specified using the Language menu entry.
- Product key, user name and the name of the organisation can be specified using the Product key ... menu entry.

The 'same as Windows' option is active as the default setting

7. Functions

WatchDog pro offers numerous functions available in six different function libraries:

- Mathematical (basic operations, modulo, increment, decrement, minimum, maximum, absolute value and rounding)
- Comparisons
- Logic (And, Or, Exclusive Or, Negation and RS flip-flop)
- Time functions (tripping and turn-off delay, single-shot-leading edge and trailing edge, flasher and timer)
- Monitoring and control functions (minimum, maximum and window monitoring, start-up monitoring, PID controller)
- Other functions (multiplexer, edge forming, counter function and program-flow control functions such is conditional jump).

As well as with their name, the function parameters are also provided with a colored direction sign indicating the use of the linked variable. That means an input is also visually distinguishable from an output.

Some functions also have some optional parameters as well as mandatory ones. The default setting is for the optional parameters to be masked out; they can be called up from the context menu as and when required. For example, most time functions have Halt and Reset as optional parameters.

Some digital inputs respond to edges. Examples of these are the Trg input, Edge and the DataL. (datalogger) function.
The edge to which the input responds is shown graphically for these inputs:

- 

Rising (= leading) edge
- 

Falling (= trailing) edge
- 

Both edges

The required signal edge can be selected from the context menu under Edge.

7.1 Assignments

The input value In is assigned to the output value Out. The parameters can be any type.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
In	×				×

7.2 Addition

Out supplies the result of adding all input values In.

Up to 31 inputs can be added to the function using Add input from the context menu.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
In	×				×
In	×				×
...	×				

7.3 Subtraction

Out supplies the result of X - Y.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
X	×				×
Y	×				×

7.4 Multiplication

Out supplies the result of multiplying all input values In.

Up to 31 inputs can be added to the function using Add input from the context menu.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
In	×				×
In	×				×
...	×				

7.5 Division

Out supplies the result of X divided by Y.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
X	×				×
Y	×				×

7.6 INC (increment)

One will be added to the InOut variable if input EN is true.
The input value at InOut will be transformed into a whole number (decimal places cut off).
Input EN can be inverted using Invert input from the context menu.

Value range:
[-32760..32760] with total accuracy,
±32760 to ±2147483648 with 99.969% accuracy due to the float number format.
In the case of ±2,147,483,648 there will be an overflow (2147483647+1= -2147483648 or -2147483648-1=+2147483647)

Parameter	In	Out	Digital	Invertible	Obligatory
InOut	×	×			×
EN	×		×	×	×

7.7 DEC (decrement)

One will be subtracted from the InOut variable if input EN is true (not equal to 0).
The input value at InOut will be transformed into a whole number (decimal places cut off).
Input EN can be inverted using Invert input from the context menu.

Value range:
[-32760..32760] with total accuracy,
±32760 to ±2147483648 with 99.969% accuracy due to the float number format.
In the case of ±2,147,483,648 there will be an overflow (2147483647+1= -2147483648 or -2147483648-1=+2147483647)!

Parameter	In	Out	Digital	Invertible	Obligatory
InOut	×	×			×
EN	×		×	×	×

7.8 Min (minimum)

Supplies the lowest of the input values In at output Out.

Up to 31 inputs can be added to the function using Add input from the context menu.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
In	×				×
In	×				×
...	×				

7.9 Max (maximum)

Supplies the highest of the input values In at output Out.

Up to 31 inputs can be added to the function using Add input from the context menu.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
In	×				×
In	×				×
...	×				

7.10 Modulo

Out supplies the remainder of the integral division of X by Y.
The function is valid for numbers in the range ±2147483648 (±2^31).
The result will not be correct for higher input values.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
X	×				×
Y	×				×

7.11 Trunc (Truncate)

The Out result of the Trunc function is the integral portion of the float parameter In (decimal places will be cut off).

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
In	×				×

7.12 Round

The Out result of the Round function is a whole number closest to the value of the float parameter In.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
In	×				×

7.13 | x | (absolute value)

The Out result of this function represents the absolute value of the input variable X (negative numbers will be transformed into positive numbers).

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×			×
X	×				×

7.14 AND

In	In	Out
0	0	0
0	1	0
1	0	0
1	1	1

Supplies the result of the logical 'and' linking of the inputs.
The input values will be interpreted as digital values (0 = 'false', everything else is 'true').

The inputs and the output can be inverted using Invert input and Invert output from the context menu.

Up to 31 inputs can be added to the function using Add input from the context menu.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×	×	×	×
In	×		×	×	×
In	×		×	×	×
...	×		×	×	

7.15 OR

In	In	Out
0	0	0
0	1	1
1	0	1
1	1	1

Supplies the result of the logical 'or' linking of both inputs In.
The input values will be interpreted as digital values (0 = 'false', everything else is 'true').

The inputs and the output can be inverted using Invert input and Invert output from the context menu.

Up to 31 inputs can be added to the function using Add input from the context menu.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×	×	×	×
In	×		×	×	×
In	×		×	×	×
...	×		×	×	

7.16 XOR

In	In	Out
0	0	0
0	1	1
1	0	1
1	1	0

Output Out supplies the result of the logical 'exclusive or' linking of inputs In.
The input values will be interpreted as digital values (0 = 'false', everything else is 'true').

The inputs and the output can be inverted using Invert input and Invert output from the context menu.

Up to 31 inputs can be added to the function using Add input from the context menu.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×	×	×	×
In	×		×	×	×
In	×		×	×	×
...	×		×	×	

7.17 NOT

In	Out
0	1
1	0

Out supplies the logically inverted value of In.
In will be interpreted as a digital value (0 = 'false', everything else is 'true').

Parameter	In	Out	Digital	Invertible	Obligatory
Out		×	×		×
In	×		×		×

7.18 RS-FF (RS Flipflop)

Set	Reset	Out
0	0	Out $_{-1}$
1	0	1 (Set)
0	1	0 (Reset)
1	1	0 (Reset)

The RS flip-flop is characterized by having two possible stable conditions: Set and reset. To set the flip-flop, a 1 must be applied to the Set input. The flip-flop will then remain in the set condition until a 1 is again applied to the Reset input. The Reset input predominates.
The input values will be interpreted as digital values (0 = 'false', everything else is 'true').
The output can be inverted using Invert output from the context menu.
0 for 'false' and 1 for 'true' is fed out at output Out.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X	X	X
Set	X		X		X
Reset	X		X		

7.19 T-FF (T Flipflop)

A T flip-flop changes the status of the output Out whenever there is a leading edge at the input In. It can be used as a latching relay.
As in the case of other edge-triggered inputs, no edge will be detected if the input was already activated on start.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
In	X		X		X

7.20 X==Y (Equality)

Compares the input values X and Y and supplies 1 at output Out in the event of equality, otherwise 0.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
X	X				X
Y	X				X

7.21 X!=Y (Inequality)

Compares the input values X and Y and supplies 1 at output Out in the event of inequality, otherwise 0.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
X	X				X
Y	X				X

7.22 X>Y (Is greater than)

Compares the input values and supplies 1 at output Out if X>Y, otherwise 0.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
X	x				x
Y	x				x

7.23 X>=Y (Is greater than or equal to)

Compares the input values and supplies 1 at output Out if X>Y or X=Y, otherwise 0.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
X	x				x
Y	x				x

7.24 X<Y (Is less than)

Compares the input values and supplies 1 at output Out if X<Y, otherwise 0.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
X	x				x
Y	x				x

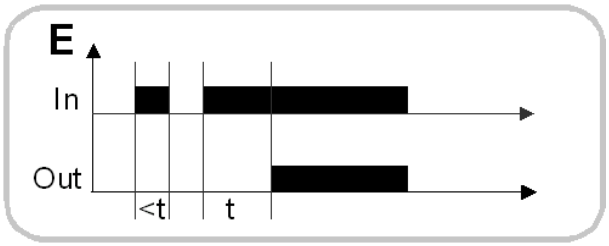
7.25 X<=Y (Is less than or equal to)

Compares the input values and supplies 1 at output Out if X<Y or X=Y, otherwise 0.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
X	x				x
Y	x				x

7.26 E (tripping delay)

Tripping delay with a control contact.



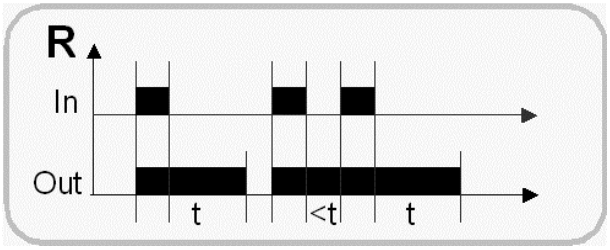
In corresponds functionally to the control contact and Out to the relay contact of a conventional time relay set to time Time. Halt interrupts the timing process.
Out can assume the values 0 (output is canceled) and 1 (output is set).
Reset restores the function to the initial status.
In, Halt and Reset are active when the value is not equal to 0.

The Halt and Reset inputs are optional and are masked out as the default setting. They can be displayed using Show Halt and Reset inputs.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
Time	x				x
In	x		x		x
Halt	x		x		
Reset	x		x		

7.27 R (turn-off delay)

Turn-off delay with a control contact.



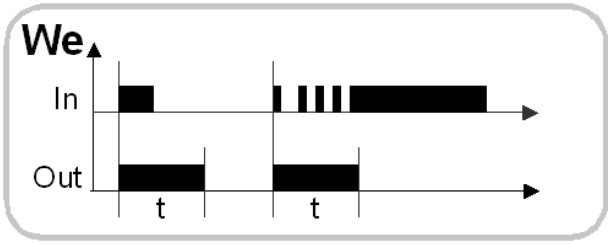
In corresponds functionally to the control contact and Out to the relay contact of a conventional time relay set to time Time. Halt interrupts the timing process.
Out can assume the values 0 (output is canceled) and 1 (output is set).
Reset restores the function to the initial status.
In, Halt and Reset are active when the value is not equal to 0.

The Halt and Reset inputs are optional and are masked out as the default setting. They can be displayed using Show Halt and Reset inputs.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
Time	x				x
In	x		x		x
Halt	x		x		
Reset	x		x		

7.28 We (single-shot leading edge)

Single-shot leading edge with a control contact.



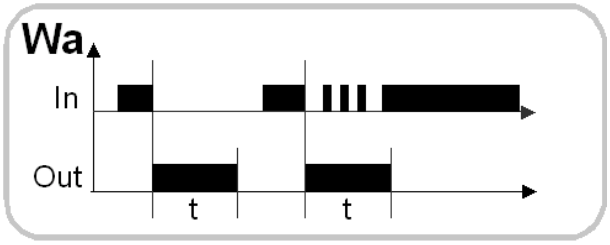
In corresponds functionally to the control contact and Out to the relay contact of a conventional time relay set to time Time. Halt interrupts the timing process.
Out can assume the values 0 (output is canceled) and 1 (output is set).
Reset restores the function to the initial status
In, Halt and Reset are active when the value is not equal to 0.
Single-shot leading edge means that output Out is set for the Time (t) (1) after the positive edge of the control contact In. The function is not retriggerable.

The Halt and Reset inputs are optional and are masked out as the default setting. They can be displayed using Show Halt and Reset inputs.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
Time	x				x
In	x		x		x
Halt	x		x		
Reset	x		x		

7.29 Wa (single-shot trailing edge)

Single-shot trailing edge with a control contact.



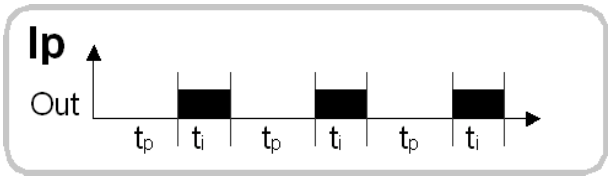
In corresponds functionally to the control contact and Out to the relay contact of a conventional time relay set to time Time. Halt interrupts the timing process.
Out can assume the values 0 (output is canceled) and 1 (output is set).
Reset restores the function to the initial status.
In, Halt and Reset are active when the value is not equal to 0.
Single-shot trailing edge means that output Out is set for the Time (t) (1) after the negative edge of the control contact. The function is not retriggerable.

The Halt and Reset inputs are optional and are masked out as the default setting. They can be displayed using Show Halt and Reset inputs.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
Time	x				x
In	x		x		x
Halt	x		x		
Reset	x		x		

7.30 lp (asymmetric flasher with pause start)

The Pause time and Pulse time can be set separately.



Out can assume the values 0 (output is canceled) and 1 (output is set).
Halt interrupts the timing process. Reset restores the function to the initial status.
Halt and Reset are active when the value is not equal to 0.

The Halt and Reset inputs are optional and are masked out as the default setting. They can be displayed using Show Halt and Reset inputs.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
Pause	x				x
Pulse	x				x
Halt	x		x		
Reset	x		x		

7.31 Timer

The parameter Out is incremented by the value 0.1 every 100 ms while input EN (Enable) is active. Input EN can be inverted.
Output Out indicates the expired time in seconds using the floating decimal point form of numerical notation.
The time overflows at 429496729.6 seconds (~13.6 years).
The optional Reset input resets the counter to zero.
The time is derived from the internal quartz oscillator (not from the RTC) and has its accuracy (see Hardware Manual).

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
EN	X		X	X	X
Reset	X		X		

7.32 P.Timer (measure operating time)

The parameter Out is incremented by the value 0.1 every 100 ms while input EN (Enable) is active. Input EN can be inverted.
Output Out indicates the expired time in seconds using the floating decimal point form of numerical notation.
The only difference compared to the Timer function is that the P.Timer function uses a permanent register for the internal counter's storage. That ensures that the counter reading (of the hours-run meter, for example) will be retained following a power outage or after resetting.
The time overflows at 429496729.6 seconds (~13.6 years).
The optional Reset input resets the counter to zero.
The time is derived from the internal quartz oscillator (not from the RTC) and has its accuracy.

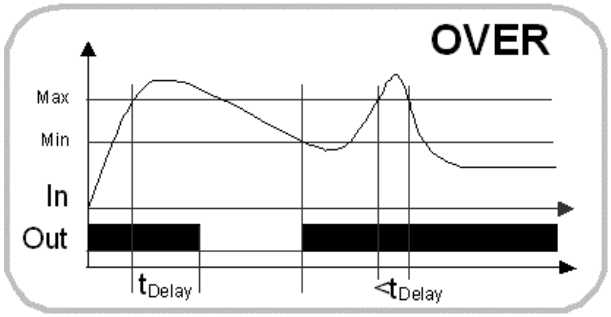
Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
EN	X		X	X	X
Reset	X		X		

7.33 Digital time switch

At a specific instant, this function feeds out a pulse lasting one cycle time at the digital output Out (0 = canceled, 1 = set). The instant is defined in the function properties dialog (function context menu). The internal RTC (Real-Time Clock) serves as the source for the time; it offers a resolution of 1 sec. 

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X

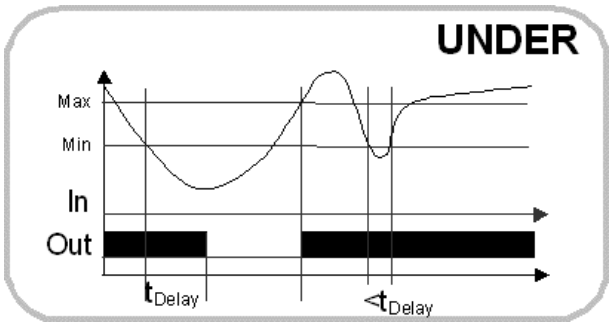
7.34 OVER (maximum monitoring)



The optional En (Enable) input activates the function and starts the optional start-up suppression time. The Out output is always activated during the start-up suppression time.
The output is immediately deactivated when En is deactivated.
Output Out will remain set (1) while the input value is less than the Max threshold value. If the input value In exceeds the Max threshold, the delay will be started and the output will remain set until this time has expired. The output will be canceled IMMEDIATELY if no delay has been specified.
The output will remain canceled until the measured value falls below the Min threshold value. The MIN threshold will be dynamically set to 98% of the Max threshold if no value was specified for the Min threshold.
A fixed delay of 400 ms will be effective on changeover to the OK condition.
Delay and StartT are indicated in seconds; resolution 0.1 sec.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
In	X				X
Max	X				X
Min	X				
Delay	X				
Start T	X				
En	X		X	X	

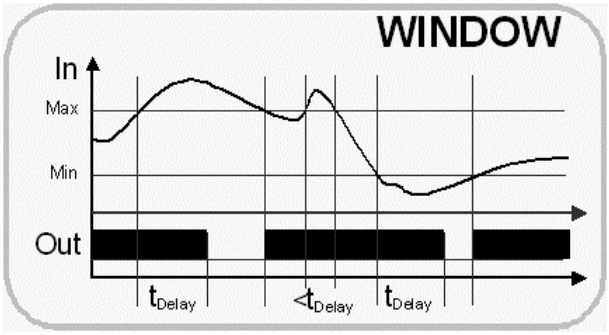
7.35 UNDER (minimum monitoring)



The optional En (Enable) input activates the function and starts the optional start-up suppression time. The Out output is always activated during the start-up suppression time (StartT).
The output is immediately deactivated when En is deactivated.
Output Out will remain set (1) while the input value In exceeds the Min threshold value. If the input value falls below the Min threshold, the delay will be started and the output will remain set until this time has expired. The output will be canceled IMMEDIATELY if no delay has been specified.
The output will remain canceled until the measured value exceeds the Max threshold value. The Max threshold will be dynamically set to 102% of the Min- threshold if no value was specified for the Max threshold.
A fixed delay of 400 ms will be effective on changeover to the OK condition.
Delay and StartT are indicated in seconds; resolution 0.1 sec.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
In	X				X
Min	X				X
Max	X				
Delay	X				
Start T	X				
En	X		X	X	

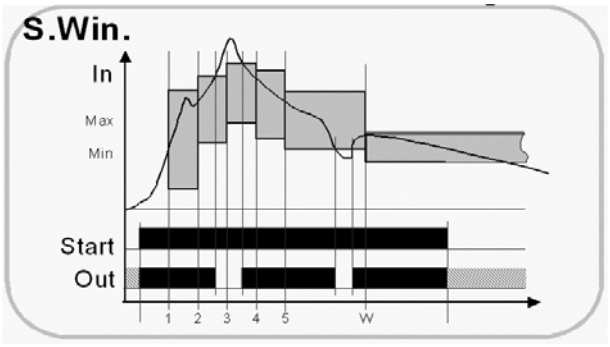
7.36 WINDOW (window monitoring)



The optional En (Enable) input activates the function and starts the optional start-up suppression time. The Out output is always activated during the start-up suppression time (StartT).
The output is immediately deactivated when En is deactivated.
Output Out will remain set (1) while the input value In is between Min and Max. The delay will be started when the window is exited and the output will remain set until this time has expired. The output will be canceled IMMEDIATELY (Out = 0) if no delay has been specified.
The output will remain canceled until the measured value has attained a value within the window. A fixed delay of 400 ms will be effective on changeover to the OK condition.
Delay and StartT are indicated in seconds; resolution 0.1 sec.
Both, Min and Max are working with fix 2% threshold.

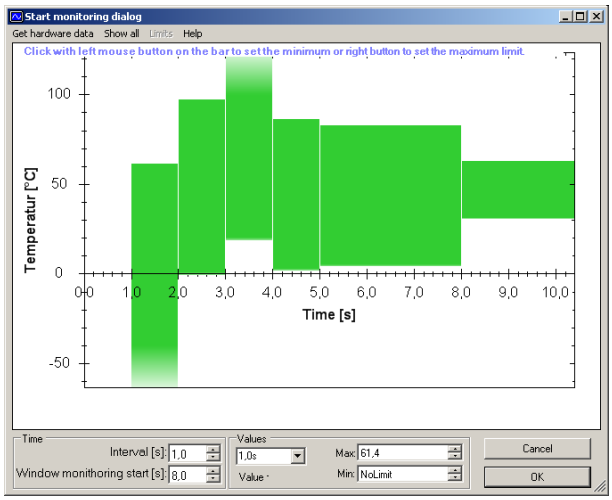
Parameter	In	Out	Digital	Invertible	Obligatory
Out		x	x		x
In	x				x
Min	x				x
Max	x				x
Delay	x				
StartT	x				
En	x		x	x	

7.37 SWindow (start-up monitoring)



This function is started with a leading edge at the Start input (even if Start was already active during start-up). Output Out will be set at least by the time of the first qualifying point.
Out can assume the values 0 (output is canceled) and 1 (output is set).
The input variable In is constantly monitored from the first qualifying point onward (window monitoring).
New pairs of threshold values remaining valid until the next qualifying point are transferred at each of the six qualifying points (in the graph: 1, 2, 3, 4, 5, W).

The behavior of the output, if the Start is inactive, can be specified in the S.Window context menu using Set output for inactivity.
The default setting for the output is 0 in the S.Window function's inactive status (deactivated).

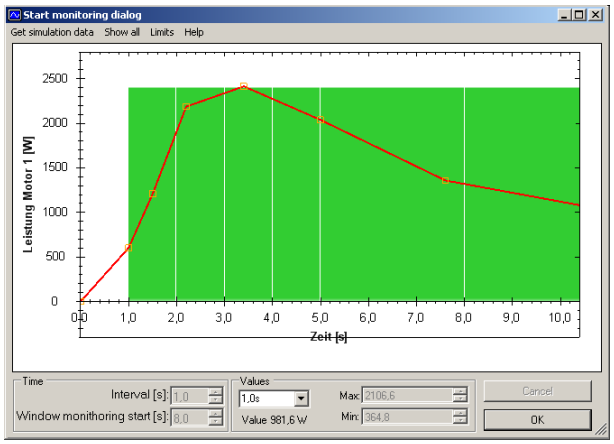


The function's behavior can be set in the Start-up monitoring dialog window, which is opened using the context menu S.Wind properties... .
It is possible to select both the time interval between two qualifying points (min. 1 sec.) and the instant at which window monitoring starts.
The OK value ranges are shown in the dialog by green bars. The ends of the bars are the thresholds.
The minimum can be set by left-clicking in the graph.
The maximum can be set by right-clicking.
You can set the thresholds more precisely in the 'Values' control-field group by first selecting the time instant (the required threshold value), then changing the associated Min and Max thresholds.
The threshold value can also be deactivated if it does not have to be checked. To do this, click in the graph with the relevant mouse button outside the input variable's value range.
The green bar will then show continuous shading for this threshold value, representing the inactive threshold value.

Display the curve of the value and set the threshold values according to the curve

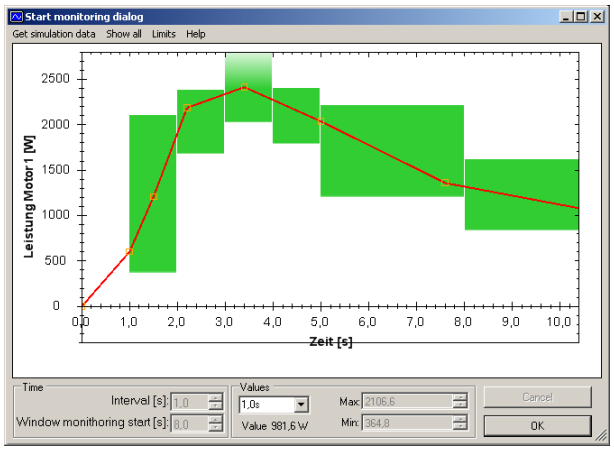
The central unit will accept the actually measured values of the input variable when the function start-up monitoring function is activated.
These values can be read out from the central unit and displayed in the graph. That will make it easier to define the threshold values.

In the online mode the S.Window function is given the context menu entry Show curve... displaying the start-up monitoring dialog with the curve that has been recorded. The function parameters cannot be changed directly here (only using the Threshold values menu, see further below).
The curve can also be read in in the programming mode using the menu item Read data from the central unit.
The rectangular markings on the curve represent the measured values. The red line interpolates between the measured values.



Using Show all in the start-up dialog, you can set the time axis in such a way that all recorded values will be displayed.
Use Zoom in to re-zoom the time axis to the range containing the threshold values.

The Threshold values menu entry becomes active when the recorded curve has been read in. It has several submenus that can be used to automatically set the threshold values relative to the recorded curve.
The same percentages can be used for both threshold values (e.g. +-20%), or Other... can be used to set the percentages for minimum and maximum separately and to any values.
Programming Environment will be changed over automatically to the programming mode so that the threshold values can be changed. The individual values can only be post-edited in the programming mode.



Parameter	In	Out	Digital	Invertible	Obligatory
Out		X	X		X
In	X				X
Start	X		X		X

7.38 PID controller

The DV (desired value) and PV (process value = actual value) inputs and the Out output are interfaces to the process.
The Kp (proportional component), Tn (reset time) and Tv (derivative time) inputs specify the PID control parameters.
The inputs Min and Max limits the output value (Anti Wind-up).
The controller's sampling time is the same as the cycle time. To ensure controllability, the system response time should be about an order of magnitude longer than the cycle time.
Output Out supplies the manipulated variable.

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x			x
DV	x				x
PV	x				x
Kp	x				x
Tn	x				x
Tv	x				x
Min	x				
Max	x				

7.39 Mux (multiplexer)

Output Out is assigned one of the parameters X or Y as a function of the value of the Select input.
Examples: 0 at the Select input -> Out = X; 1 at the Select input -> Out = Y
Select is evaluated as a whole unsigned number (0.7 or -5 is same as 0).

Parameter	In	Out	Digital	Invertible	Obligatory
Out		x			x
Select	x				x
X	x				x
Y	x				x

7.40 Edge (edge detection)

Detects edges at the Trg input and feeds out a pulse lasting one cycle time at the Out output whenever there is an edge.
Leading, trailing and both edges can be selected in the input's context menu.

Parameter	In	Out	Digital	Invertible	Obligatory	Edge
Out		x	x		x	
Trg	x		x		x	leading, trailing, both

7.41 Counter

Counts the edges at the count input Cnt and supplies the result at the output Out.
With a right click on the Cnt input the option to select between leading or trailing edge as trigger for increment or decrement is selectable.
The counting direction can be defined using the optional DIR input:

DIR=0 count down,
DIR=1 (not equal to 0) count up.

The default setting is for the function to count up.

The optional Reset input resets the counter to zero.

The output value is modifyed only by the inputs Cnt, DIR and Reset. It is not possible to preset the counter to another value than zero.

Counting range:
[-32760..32760] with total accuracy,
±32760 to ±2147483648 with 99.969% accuracy due to the float number format.
In the case of ±2,147,483,648 there will be an overflow (2147483647+1= -2147483648 or -2147483648-1=+2147483647)!

Parameter	In	Out	Digital	Invertible	Obligatory	Edge
Out		x			x	
Cnt	x		x		x	leqading, trailing, both
DIR	x		x			
Reset	x		x			

7.42 P.Counter (permanent Counter)

Counts the edges at the count input Cnt and supplies the result at the output Out.
The only difference compared to the Counter function is that the P.Counter function uses a permanent register for the internal counter's storage. That ensures that the counter reading will be retained following a power outage or after resetting.

With a right click on the Cnt input the option to select between leading or trailing edge as trigger for increment or decrement is selectable.
The counting direction can be defined using the optional DIR input:

DIR=0 count down,
DIR=1 (not equal to 0) count up.

The default setting is for the function to count up.

The optional Reset input resets the counter to zero.

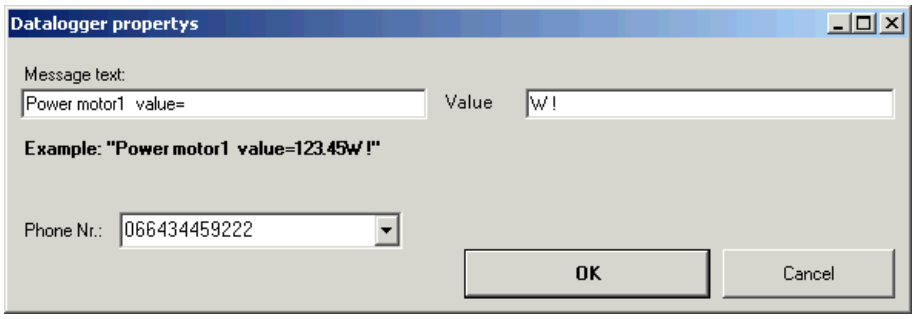
The output value is modifyed only by the inputs Cnt, DIR and Reset. It is not possible to preset the counter to another value than zero.

Counting range:
[-32760..32760] with total accuracy,
±32760 to ±2147483648 with 99.969% accuracy due to the float number format.
In the case of ±2,147,483,648 there will be an overflow (2147483647+1= -2147483648 or -2147483648-1=+2147483647)!

Parameter	In	Out	Digital	Invertible	Obligatory	Edge
Out		x			x	
Cnt	x		x		x	leqading, trailing, both
DIR	x		x			
Reset	x		x			

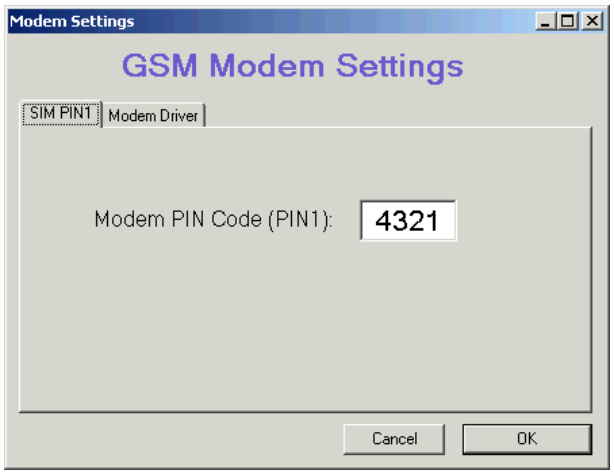
7.43 DataL. (datalogger)

The datalogger logs the variable value being applied to the In input or just simple text message if there is no variable specified, when the trigger Trg is active. The trigger input is edge-controlled and can react to a leading edge, a trailing edge or both (Edge option in the Trg input's context menu). The MMC output (causes storage on the flash card) of the MMC-enabled central unit or the SMS output of the CU can function as the output of the datalogger function D.Sink (data sink). The text that is also logged each time data is logged must be defined in the Properties dialog. The same applies to the SMS recipient's phone number.



The logging rate can be up to 150 values in 3 seconds. An overflow entry is created each time the buffer overflows.

For sending an SMS, it is necessary to indicate the SIM card's PIN number and the modem type (Modem settings ... context menu of the CU's SMS output).



Parameter	In	Out	Digital	Invertible	Obligatory	Edge
D.Sink		X			X	
Trg	X		X		X	leading, trailing, both
In	X					

7.44 JMP (conditional jump)

A branch will be made to the relevant label if the condition (Cond.) is logically 'true' (not equal to 0); the program will otherwise be resumed unchanged.
Only the forward branches are possible. The associated label can only be positioned after the jump instruction.

Parameter	In	Out	Digital	Invertible	Obligatory
Cond.	x		x	x	x
Label					x

7.45 Safe (safe condition)

Puts WatchDog pro's outputs into the safe condition (deactivated) when the Enable input is logically true.
Everything else fed out by the program will then be overwritten!
The field bus outputs of the G2FBC will not be affected.

Parameter	In	Out	Digital	Invertible	Obligatory
Enable	x		x	x	x