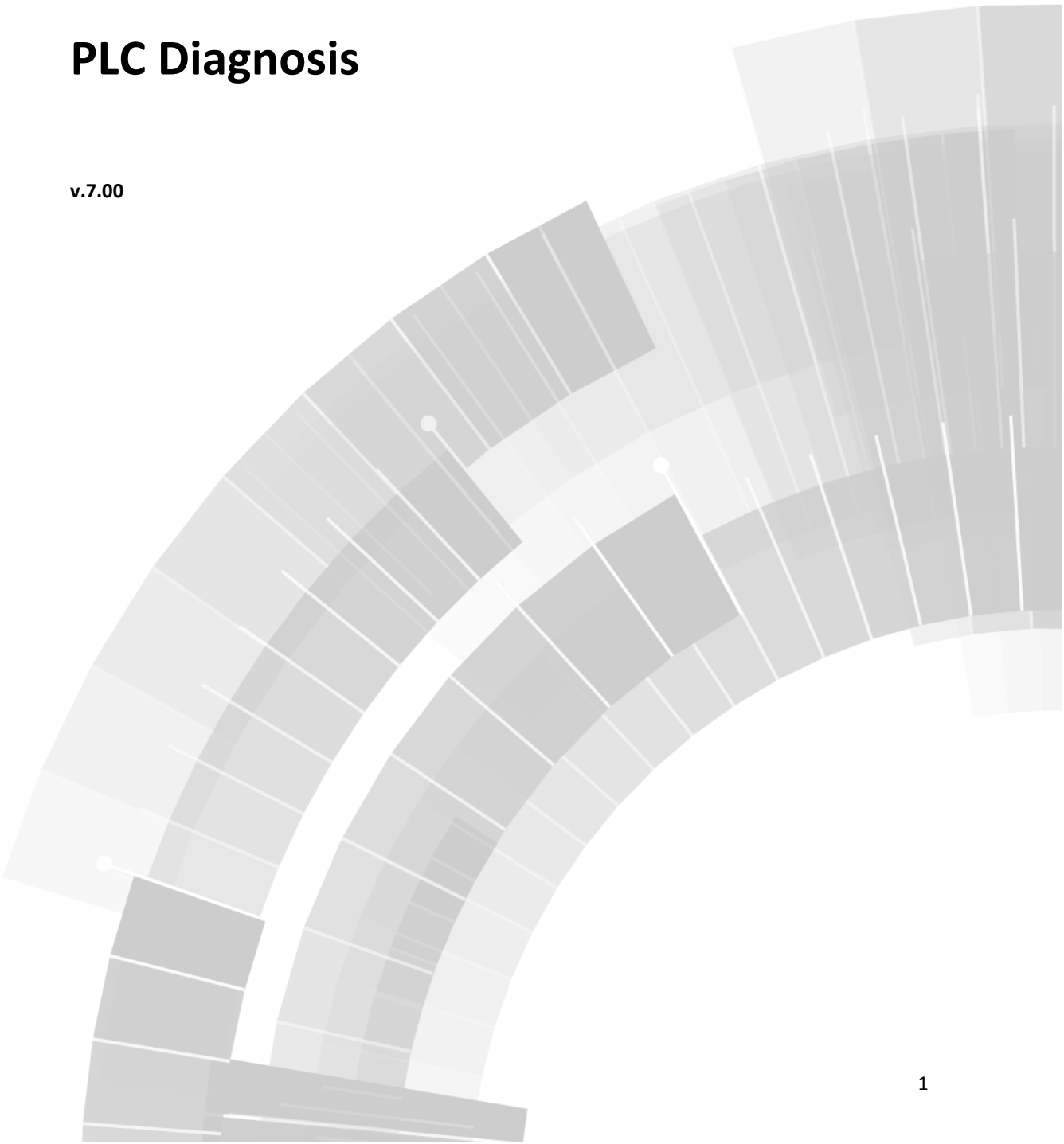
COPA-DATA
do it your way

zenon manual

PLC Diagnosis

v.7.00



Contents

1. Welcome to COPA-DATA help

GENERAL HELP

If you miss any information in this help chapter or have any suggestions for additions, please feel free to contact us via e-mail: documentation@copadata.com (mailto:documentation@copadata.com).

PROJECT SUPPORT

If you have concrete questions relating to your project, please feel free to contact the support team via e-mail: support@copadata.com (mailto:support@copadata.com)

LICENSES AND MODULES

If you realize that you need additional licenses or modules, please feel free to contact the sales team via e-mail: sales@copadata.com (mailto:sales@copadata.com)

2. PLC Diagnosis

With the PLC Diagnosis program details of a PLC can be displayed directly in a screen of the control system. Depending on the PLC different functionalities are available. At the moment the S7 Graph (on page 5) functional chart analysis for the display of functional chart information of S7 PLCs and the zenon Logic Monitoring Viewer (on page 17) for the display of all programs being executed in the zenon Logic (on page 17)Runtime (Soft PLC) are available.

In the Runtime the currently active steps from the PLC are displayed graphically. Alarm information can be directly transferred to the alarm administration.

License information

Must be licensed for Editor and Runtime (single-user, Server, Standby and Client).
available for zenon Operator and under Windows CE as a licensed module.

3. S7 Graph

You have the possibility to create a screen of the type S7 Graph in zenon. In this screen you can display the functional chart, the active step, the name of the functional chart, the number of the functional chart etc. of a S7 control with S7 Graph programming.

3.1 Changeover from ActiveX CD_Graph7Diag.ocx to screen of type S7 Graph

The functionality of the ActiveX CD_Graph7Diag.ocx has been taken over by the screen S7 Graph and the function Analyze S7 Graph heuristics. In order to change from ActiveX to the screen and the function or to display S7 Graph in zenon, carry out the following steps:

1. If you have not already done it, carry out the S7 Graph import so that the required variables and files are created.
2. Create screen of the type S7 Graph.
3. Create screen switch functions to screen S7 Graph. Enter the respective variables and files (see also Filter for screen switch (on page 13)).
4. If you need the heuristics, create the function Analyze S7 Graph heuristics and enter the respective variables and files (see also Function Analyze S7 graph heuristics (on page 15)).

CHANGES COMPARED TO THE ACTIVEX

- Buttons LD and FBD are now distinct control elements
- Heuristics is detached from the actual display. Thus heuristics needs a function of its own.

- Button forward and back make it possible to navigate step-by-step through the step detail view.

3.2 Requirements for the automatic import

Even though importing is extensively automatic, some preparatory work has to be done in the S7 Graph project in order to make the import of functional charts possible in zenon. The following prerequisites are necessary:

1. DATA BLOCK SETTINGS

You must make sure that the configuration of the data block is correct.

Make the following settings in tab **`Translate/Save`**:

Parameters	Entry
`FB Parameter`	user defined
`Interface description`	Individual structures / Load in AS
`Run capability`	Standard FC necessary
`Chain properties`	Criteria analysis data in DB

Make the following settings in tab **`Message`**:

Parameters	Entry
`Message handling`	Message with ALARM_SQ / ALARM_S

2. GENERATE SOURCES

For each functional chart to be imported a current source must be generated. This is done in the S7 Graph editor via *File -> Generate source*. Take care that the name of the source has to match the FB name of the functional chart. So for a functional chart in the function block 12 the name of the source must be FB12 (or fb12).

3. INSTANCE DB NUMBER

In order to make the allocation of the function block of the functional chart to the according instance data block (Instance-DB) possible, you may not change the default setting (DB name = FB name).

3.3 Import data from S7 Graph project

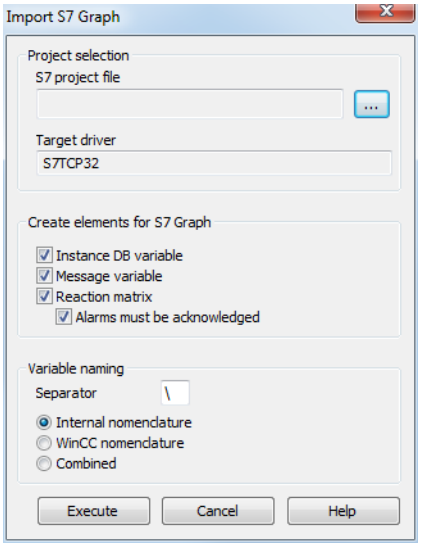
Right-click entry **Variables** in the project manager. Select menu entry **Extended import/export** from the context menu and **Import S7 Graph project...**. You can also reach the context menu by selecting the node **Variables** and right-clicking on the project manager detail view.

Select the appropriate driver that is used for communication with the PLC in the following dialog and confirm your selection with **OK**.

> **Attention**
>
> *Before the import in Step 7 or PC S7 everything must be translated for all objects to be adopted in zenon.*

IMPORT S7 GRAPH PROJECT

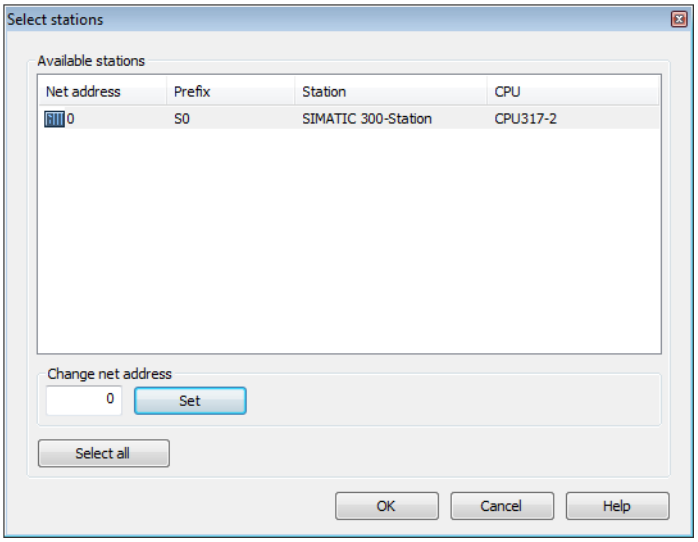
> **Info**
>
> *The settings you make in this dialog are saved. When you open the dialog again, all settings you made are displayed again.*



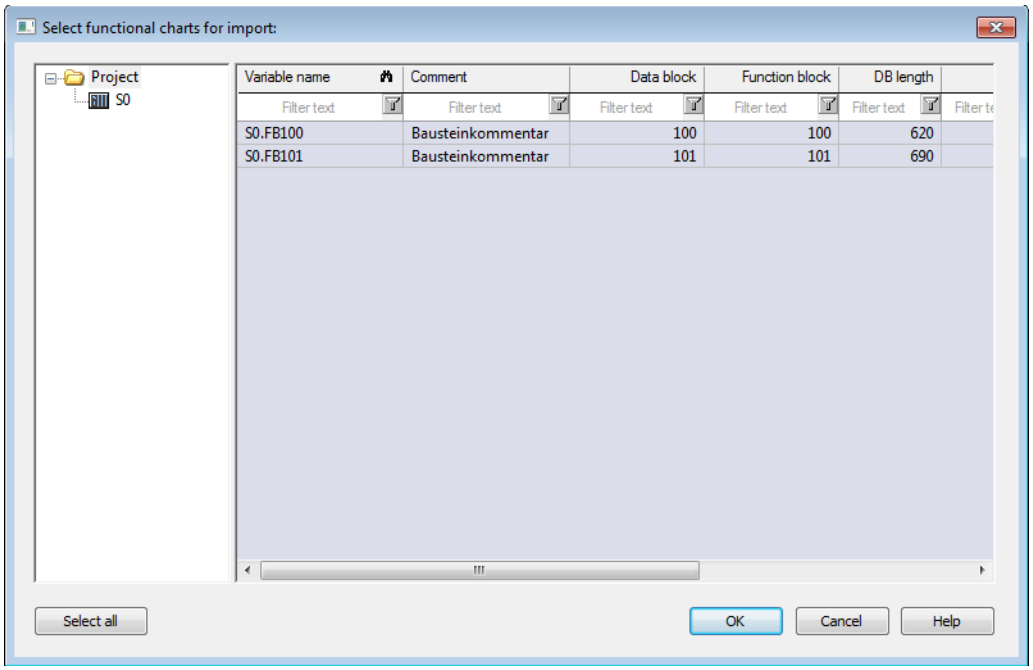
The following settings are available.

Parameters	Description
Project selection	
S7 project file	Select the desired project file (*.s7p).
Target driver	The driver you have selected before is displayed there. You cannot change it there.
Create elements for S7 Graph	
Instance DB variable	Creates variables of the type USINT which depicts the instance data block of the functional chart. All status information is communicated with the help of this variable.
Message variable	Creates an internal variable of type STRING which displays the status of the heuristics.
Reaction matrix	Creates a numeric reaction matrix. The reaction matrix is used for linking with the Alarm-S variable. With this they cause an alarm in zenon. The import of alarms for S7 Graph is carried out via PDIAG Import Wizard.
Alarms must be acknowledged	If you activate this checkbox, all alarms created by the reaction matrix must be acknowledged.
Variable naming	
Separator	Define the character which will be positioned between the nomenclature and the name of the functional chart.
Internal nomenclature	Uses the internal nomenclature for the created objects. Example: S0 (for net address = 0)
WinCC nomenclature	Uses the WinCC nomenclature for the created objects. Example: S7 program
Combined	Uses both the internal and the WinCC nomenclature for the names of the created objects. A dot is used as a separator.

Confirm the dialog with **Execute**.



Select the desired stations (CPU) in the following dialog. The dialog supports multi-select. You can select several functional charts at once if you press and hold `Ctrl` or Shift while making your selection. Enter the net address for the communication. Confirm this dialog with **OK**. An analysis of the S7 project is carried out. Depending on the size of the project, it can take some time.



IN the next dialog select the functional charts which you want to import. The dialog supports multi-select. You can select several functional charts at once if you press and hold `Ctrl` or Shift while making your selection. If no source is found for a functional chart, it is displayed in the **Comment** with the entry **!! Source missing !!**. These functional charts cannot be imported. Confirm the dialog with **OK**. The import is carried out and the respective objects are created in zenon. The source of the functional chart

(*.gr7) and the symbolism of the functional chart (*.seq) are imported as well. The are saved in the zenon Editor under `Files -> Others -> S7 Graph`.

3.3.1 Language switch

Texts in the S7 Graph chart can be engineered as language changeable. In zenon the language changeability is usually marked by a preceded `@`. In the SIMATIC manager it is however not possible to create keywords with preceded `@`. Therefore the language change is carried out via text comparison:

1. in the language table the corresponding elements are entered without the @
2. all dynamic fields of the screen are searched and it is tried to find a translation for:
   - **`Information: Name of the active sequential function chart`**
   - **`Information: Name of the active step`**
   - all texts from **`Graphical Display: Active steps`**
   - all texts from **`Graphical Display: Whole functional chart`**
   - all texts of **`Information: Symbol table`**
3. static fields must have the key character `@` as prefix and an entry in the language table must be created for:
   - **`Step back`**
   - **`Step forward`**
   - **`Switch LD/FBD`**

The display of the texts for functional charts and transitions and the texts in the symbol table take place automatically if the corresponding entries exist in **`.GR7`** or **`.SEQ`**.

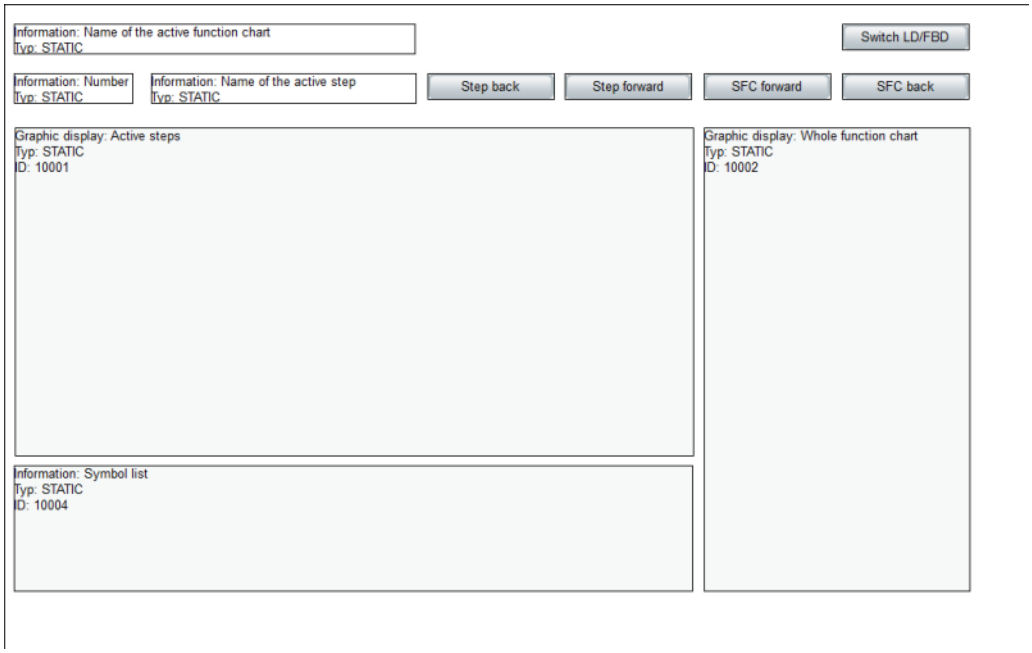
You can read details on language switch in the Language switch chapter.

3.4 Screen S7 Graph

The screen of the type S7 Graph supersedes the functionality of the ActiveX CD_Graph7Diag.ocx.

To create the screen:

1. select New screen
2. select **S7 Graph** from the the drop-down list
3. name the screen
4. add the default elements via menu item *Control elements -> Add template*
5. add additional needed elements via sub items of menu item *control elements* if necessary

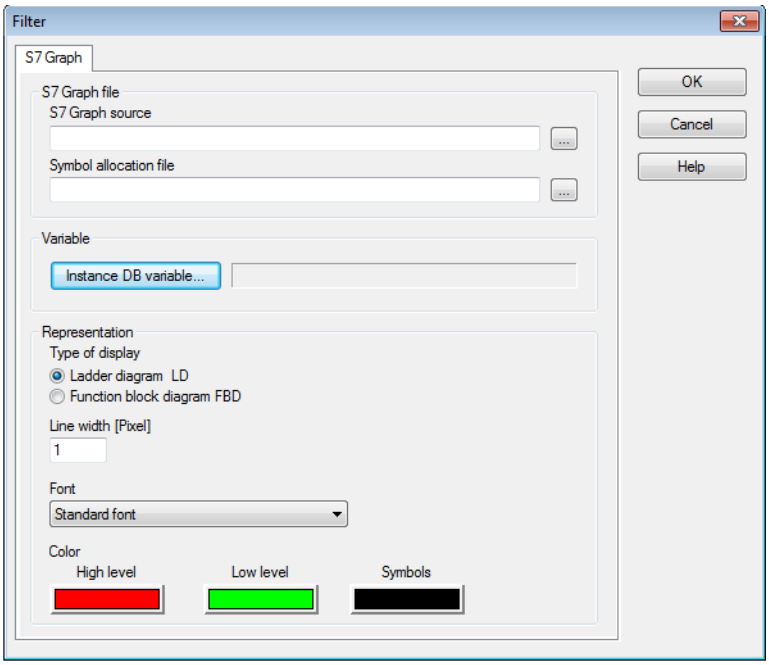


Parameters	Description
Add template	Opens the dialog for selecting a template for the screen type. Templates are shipped together with zenon and can also be created by the user. Templates add pre-defined control elements to pre-defined locations in the screen. Elements that are not necessary can also be removed individually once they have been created. Additional elements are selected from the drop-down list and dragged onto the screen. Elements can be moved on the screen and arranged individually.
Graphic display	Elements for the graphic display.
Active steps	Display of the active step or steps from the engineered functional chart.
Whole functional chart (not default)	General view of the functional chart. Displays all engineered functional charts. Is controlled via **SFC forward** and **SFC back**.
Information	Elements for informational purpose.
Name of the active functional chart.	Shows the name of the active functional chart.
Name of the active step	Shows the name of the active functional chart.
Number of the active step	Shows the number of the active functional chart.
Symbol table	Displays in two columns the address and the corresponding symbol name.
PLC	Elements for the control.
Switch LD/FBD	With the help of this button you can switch between notations LD and FBD.
Step forward (not default)	In the Runtime moves one step forward.
Step back (not default)	In the Runtime moves one step back.
SFC forward (not default)	In the Runtime in view **Whole functional chart** navigates one SFC forward if there are several SFCs in a function block.

SFC back (not default)	In the Runtime in view **Whole functional chart** navigates one SFC back if there are several SFCs in a function block.

3.5 Filter for screen switch

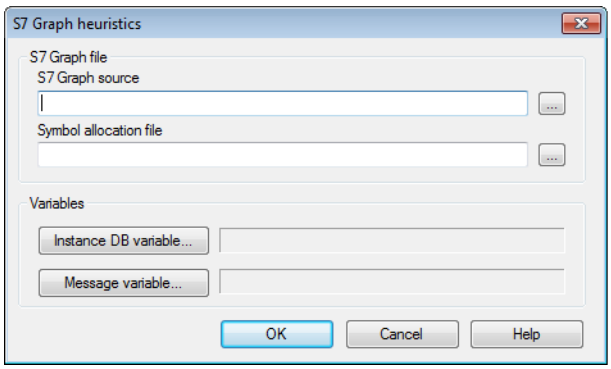
If you use function screen switch to the screen S7 Graph, the following dialog is displayed.



Parameters	Description
S7 Graph files	
S7 graph source	Select the the file (*.gr7) which has been generated by the S7 Graph import. This file contains the full logics (steps, transitions, surveillances etc.).
Symbol allocation file	Select the the file (*.seq) which has been generated by the S7 Graph import. This file contains the assignment of absolute operands (equals physical address) to symbolic operands or symbolism (equals logical identifier). Example: Input "E 1.0 = pushbutton 1"
Variable	
Instance DB variable	zenon variable which displays the instance data block of the functional chart. All status information is communicated with the help of this variable.
Representation	
Representation type	Select if you would like the display as a contact plan (COP) or as a function plan (FUP). If you configure the **Switch COP/FUP button in your screen**, you can switch between the two display types in Runtime.
Line width [Pixel]	Define the line width in pixels of the connection line between the elements.
Font	Select the desired font from the drop-down list for all labels in screen S7 Graph.
Colors	Define the colors for the high level and low level states and for the symbols. Click on the respective colored rectangle in order to change the color.

3.6 Function Analyze S7 Graph heuristics

This function makes it possible to carry out the S7 Graph heuristics without the screen S7 Graph being active. Thus for example an analysis can be carried out by a S7 PDiag error message via an Alarm S variable.



The following properties are available.

Parameters	Description
S7 Graph files	
S7 graph source	Select the the file (*.gr7) which has been generated by the S7 Graph import. This file contains the full logics (steps, transitions, surveillances etc.).
Symbol allocation file	Select the the file (*.seq) which has been generated by the S7 Graph import. This file contains the assignment of absolute operands (equals physical address) to symbolic operands or symbolism (equals logical identifier). Example: Input "E 1.0 = pushbutton 1"
Variables	
Instance DB variable	zenon variable which displays the instance data block of the functional chart. All status information is communicated with the help of this variable.
Message variable	Variable of type STRING which displays the status text of the heuristics.

3.7 Description of the operands analysis

In order to make clear, which step is disturbed and which operand in the transitions is responsible, the message variable is filled with a telling message. This message is generated in the following steps:

3.7.1 1. Finding the disturbed step

If the sent alarm variable is of the type UDINT, it contains the number of the disturbed step in case of an error. Thsi can be used for further evaluations. If only a BOOL variable is sent, the first actie step, which is disturbed at the moment, is used.

3.7.2 2. Selection of the processed transition

If in the disturbed step there are several transitions to following steps, only the transition with the lowest transition number is evaluated.

3.7.3 3. Finding the missing operands

The disturb cause which was detected in this way is displayed with the help of the message variable. The content of this variable is updated when you execute function Analyze S7 Graph heuristics.

In order to find the missing operands in the defined transition the condition tree is processed hierarchically. This can be demonstarted best in the FUP display (see example illustration belo). The following cases are distinguished on finding an operator:

AND

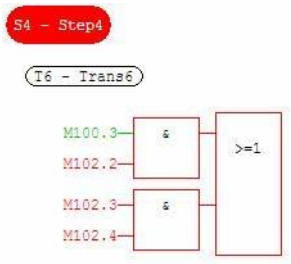
All paths are traced, which at the time of the evaluation are not fulfilled (i.e. red in the illustration).

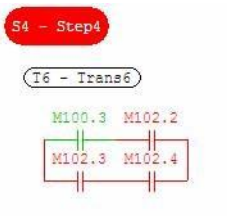
OR

For each path thefulfilment level, the proportion of fulfilled inputs and the total number of inputs, is calculated. Then only those paths are traced, that have the highest fulfilment level. If thee are several paths with the same fulfilment level, the first one is used.

With this method a possible small but nevertheless - with high probability - meaningful number of operands is found, which can be important for the next step.

In the example illustration this method would only find the operand M102.2, because this is the smallest possible number of operands, which is necessary to change the disturbed transition to undisturbed.





4. zenon Logic Monitoring Viewer

> **Info**
>
> *This tool visually displays and evaluates zenon Logic programs. With the Monitoring Viewer actions in the zenon Logic Runtime can directly be displayed in a screen in zenon.*

4.1 Requirements

As of zenon 6.20 SP3 the ActiveX X5Monitoring.ocx which carries out the zenon Logic program display is installed in the zenon program folder and registered in the operating system. For earlier zenon versions you must carry out the installation and registration manually.

At the moment this module is only available for the PC version of zenon.

A licence for the PLC Analysis is needed for using the zenon Logic Monitoring Viewer.

The Wizard zenon LogicMonitoring Builder has to be executed in the zenon Logic Workbench, before the zenon Logic Monitoring Viewer can display the current program code. The Wizard is executed in the zenon Logic Workbench under Tools - Create Monitoring Application.

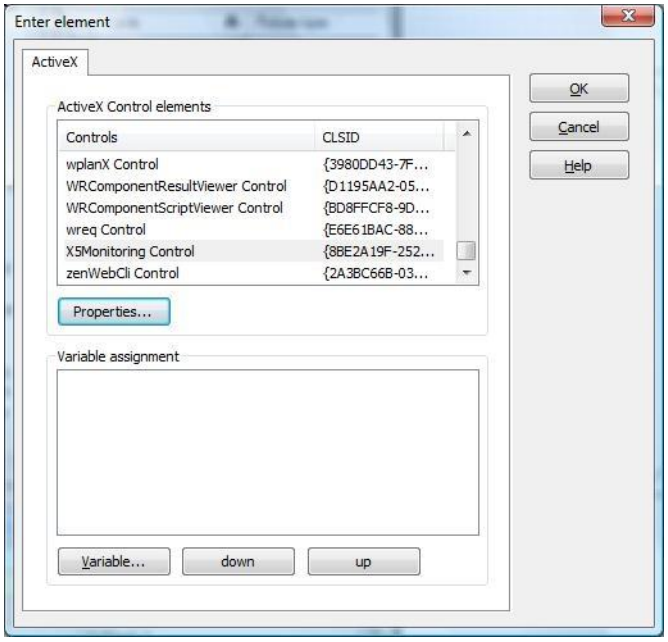
This Wizard has to be licensed.

Different proeprties can be changed in the Wizard, e.g. the authorization for viiewing programs or forcing variables can be defined (locked, password protected, open).

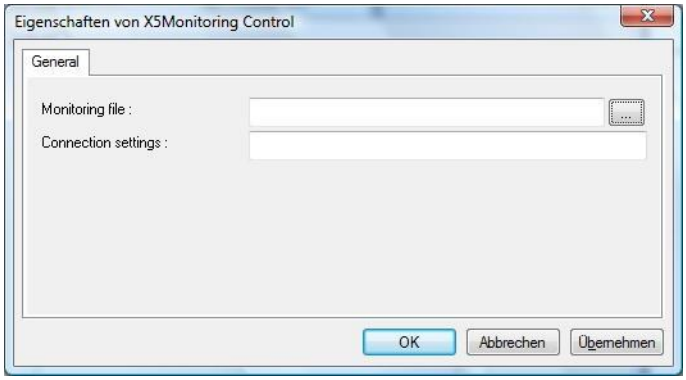
The Wizard creates a <ProjektName>.K5m file and save it in the zenon project folder.

4.2 Integration of ActiveX in a zenon screen

In the zenon Editor select the dynamic element ActiveX and draw it in a zenon screen in the desired size. Now a dialog for the selection of the ActoveX control opens. Select X5Monitoring Control and click on Properties.



In the following dialog enter the path of the file, which has been created with the Monitoring Builder and set the communication parameters:



Info

If now paths have been changed, while the Monitoring Builder Wizards was running, the monitoring file is in the SQL directory of the zenon project. Find out the GUID (e.g. by checking the zenon Logic project properties in zenon) and browse for the *.K5m file. You can normally find it in the standard path.

C:\ProgramData\COPA-DATA\SQL\<GUID>/FILES/straton/<zenon Logic Project Name>/<zenon Logic Project Name.K5m>.

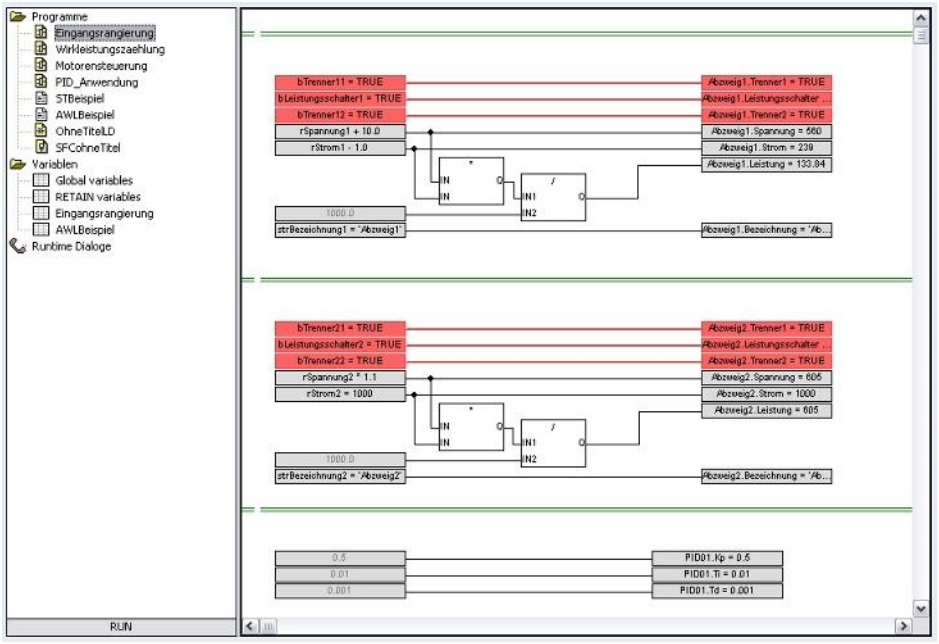
Info

For the Connection settings only use IP addresses and no computer names! If the zenon Logic Runtime runs on the same computer as zenon, enter 127.0.0.1 for the localhost.

Separated with a colon (:) enter the Runtime port of the zenon Logic Runtime after the IP address. Usually it is 1200.

4.3 Display the ActiveX in a zenon screen during runtime

Confirm the two dialog with OK. After creating the zenon Runtime files and starting the zenon and zenon Logic Runtime, the PLC code is displayed in the according zenon screen:



Depending on the authorization defined in the Monitoring Builder Wizard programs can only be seen or also values can be forced.