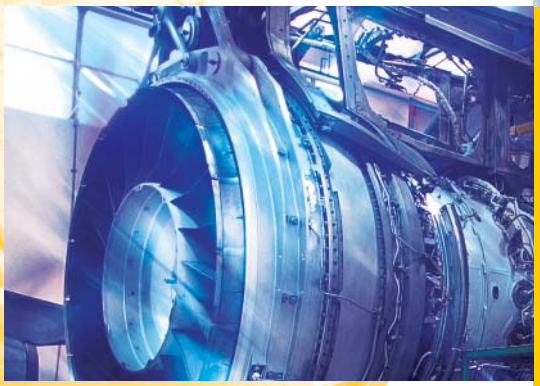
TURCK

Industrial Automation

GETTING STARTED

VT250-57C - HMI PLC with DeviceNet™



Sense it! Connect it! Bus it! Solve it!

Edition 10/2011

Warning!
Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighboring units that are live.
- Follow the engineering instructions of the device concerned.
- Only suitably qualified personnel in accordance with EN 50 110-1/-2 (VDE 0 105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalization. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference do not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60 364-4-41 (VDE 0 100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.
- Emergency stop devices complying with IEC/EN 60 204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency-stop devices must not cause restart.
- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.
- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).
- The electrical installation must be carried out in accordance with the relevant regulations (e. g. with regard to cable cross sections, fuses, PE).
- All work relating to transport, installation, commissioning and maintenance must only be carried out by qualified personnel. (IEC 60 364 and HD 384 and national work safety regulations).
- All shrouds and doors must be kept closed during operation.

Table of contents

1 About this manual

1.1 General

This Getting Started describes the programming of TURCK-HMIs with CoDeSys V3 (3S-Smart Software Solutions GmbH) as well as the visualization with QViS by means of an example using a VT250-57C with DeviceNet™-Master and a BL67-station for DeivceNet™.

1.1.1 Additional documentation

- D301189 "Getting Started - VT250-57P HMI PLC with PROFIBUS-DP"
- D301191 "VT250 - Hardware-description“
- D301195 "Getting Started - VT250-57P HMI PLC with CANopen"
- D301218 "Getting Started - VT250-57x Visualization of Step®-projects“
- D301226 "Getting Started - VT250-57x - QViS with Modbus TCP-driver“
- D301228 "Getting Started - VT250-57x - HMI PLC with Modbus TCP“

1.2 Description of symbols used

Danger

This sign can be found next to all notes that indicate a source of hazards. This can refer to danger to personnel or damage to the system (hardware and software) and to the facility.

This sign means for the operator: work with extreme caution.

Attention

This sign can be found next to all notes that indicate a potential hazard.

This can refer to possible danger to personnel and damages to the system (hardware and software) and to the facility.

Note

This sign can be found next to all general notes that supply important information about one or more operating steps.

These specific notes are intended to make operation easier and avoid unnecessary work due to incorrect operation.

1.3 General

Attention

Please read this section carefully. Safety aspects cannot be left to chance when dealing with electrical equipment.

This manual includes all information necessary for the prescribed use of TURCK HMIs. It has been specially conceived for personnel with the necessary qualifications.

1.3.1 Prescribed use

Appropriate transport, storage, deployment and mounting as well as careful operating and thorough maintenance guarantee the trouble-free and safe operation of these devices.

Danger

The devices described in this manual must be used only in applications prescribed in this manual or in the respective technical descriptions, and only with certified components and devices from third party manufacturers.

1.3.2 Notes concerning planning/ installation of this product

Danger

All respective safety measures and accident protection guidelines must be considered carefully and without exception.

2 Getting Started

2.1 Used hard-/ software

2.1.1 Hardware

- VT250-57x with DeviceNet™-Master

> **Note**
> The VT250-57C only supports the DeviceNet™-master and not the DeviceNet™-slave functionality.

- BL67-GW-DN (MAC-ID 15)
  with:
  - BL67-4DI-PD
  - BL67-2AI-I, with analog sensor Bi5-M18-Li2
  - BL67-8DO-0.5A-P

2.1.2 Software

- CoDeSys 3.4, SP3, Patch 1
- QViS Version 3.1.1.1
- Microsoft®.NET Framework 3.5

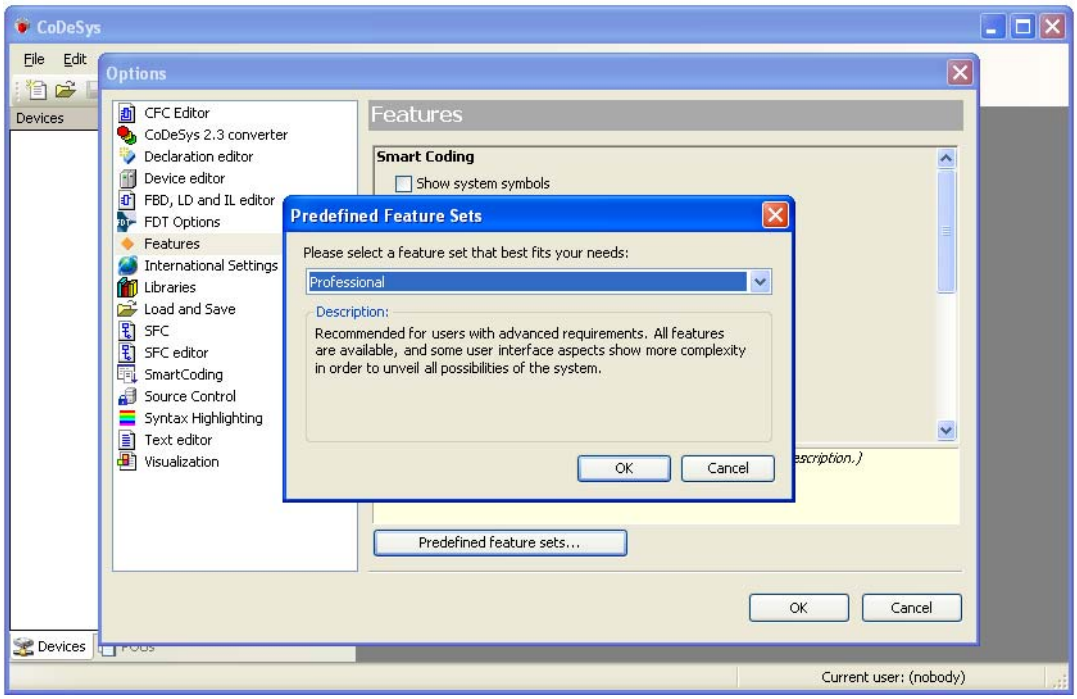
2.2 Network configuration and programming in CoDeSys

Open CoDeSys via "Start → All programs → 3S CoDeSys → CoDeSys → CoDeSys V 3.4 SP3 Patch1".

2.2.1 Predefined feature sets

In this example, CoDeSys is run with the "Professional feature set" not with the "Standard feature set". This setting has influence on different CoDeSys functions and can be changed via "Tools → Options...→ Features → Predefined feature sets...". For further information concerning this topic, please read the CoDeSys online help.

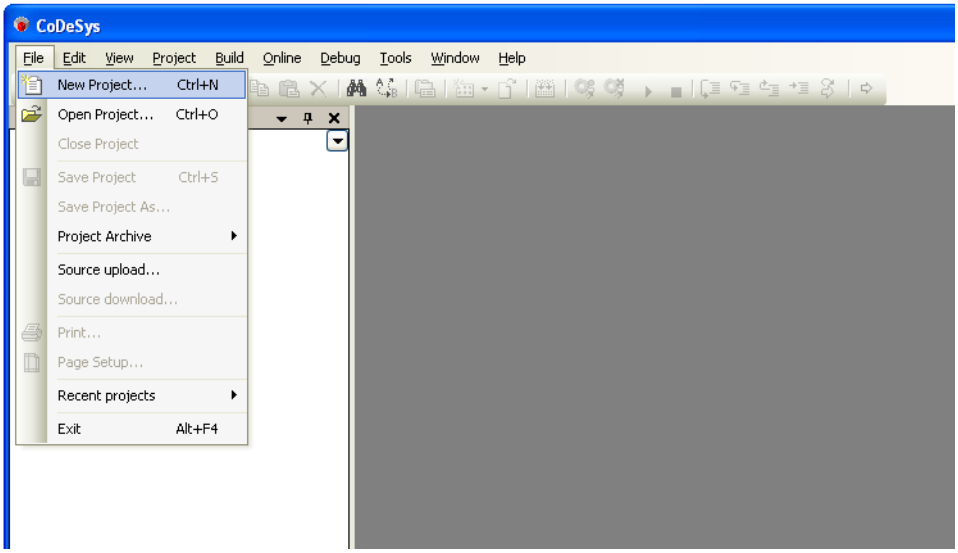
Figure 1: Predefined feature sets



2.2.2 Creating a new project

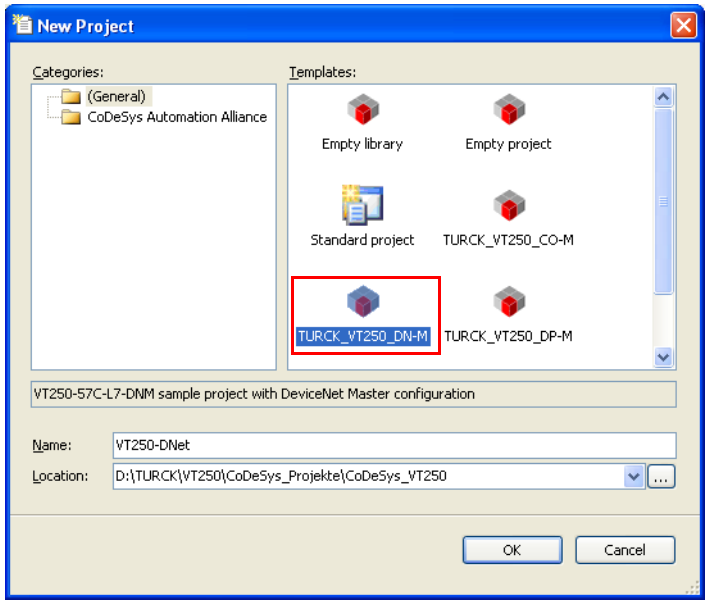
1 Create a new CoDeSys-project using the "File → New project" command.

Figure 2: New project



2 The opened dialog box "New project" shows all project templates installed with CoDeSys. TURCK offers one project template for every fieldbus master (see below Predefined templates/ example project (page 2-5)).

Figure 3: Example DeviceNet™-Master-template



Predefined templates/ example project

The TURCK CoDeSys-installation contains project templates (incl. already defined master, example slave and example program), which are automatically installed together with CoDeSys.

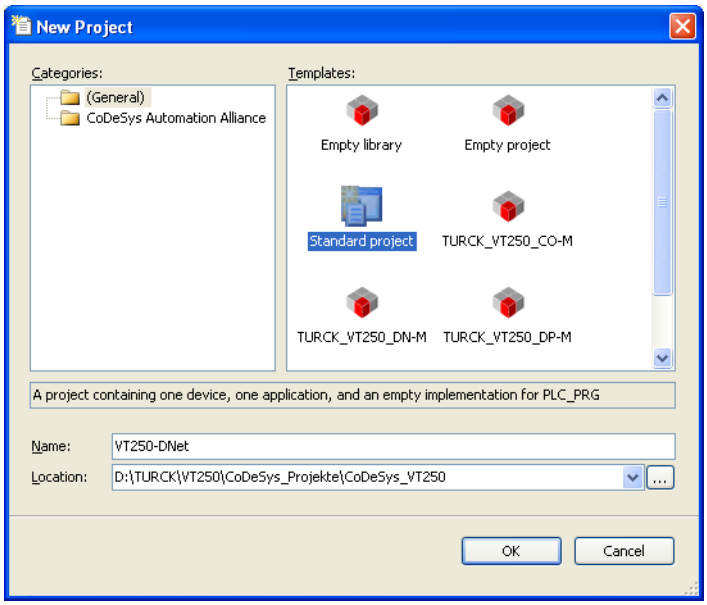
They can be opened by using the corresponding template (e. g. "TURCK_VT250_DN-M) when creating a new project (see chapter 3, Selecting the project templates (page 3-3)).

Note

Please read chapter 3, The CoDeSys example project (page 3-3) for more detailed information concerning the example project.

Creating a new Standard project

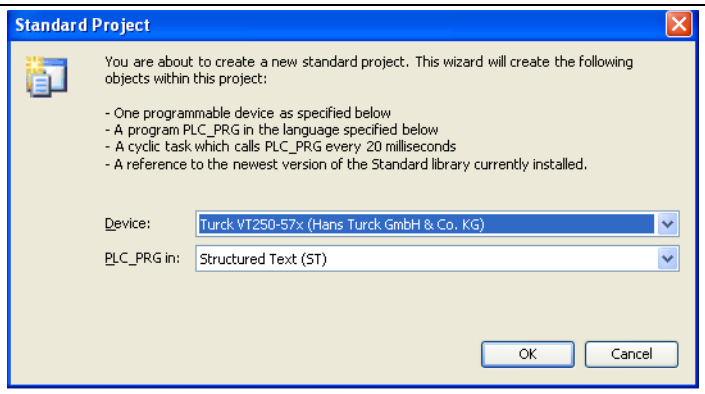
Figure 4: Standard project



3 In the new dialog box "Standard project" the used TURCK VT250-57x (Hans Turck GmbH & Co. KG) is selected as "Device".

4 Please define also your preferred programming language. In this example, Structured Text is used.

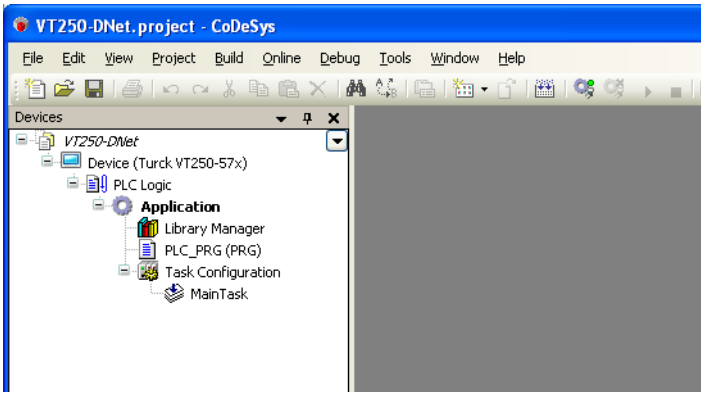
Figure 5: Selection of the VT250-57×



5 The new project is created.

6 In CoDeSys, the project tree is build up as follows:

Figure 6: Project tree



Note

If the window "devices" should not be displayed, it can be activated via "View → Devices".

2.2.3 Defining the communication settings

Double-clicking the "Device VT250-57P" opens the corresponding editors. The communication path (Gateway) to the HMI is defined in the "Communication Settings" tab.

Note

Please observe, that the VT250-57× and your PC on which CoDeSys is running, are nodes of the same Ethernet network! If this is not the case, a communication between the two is not possible. If this is not the case, a communication between the two is not possible.

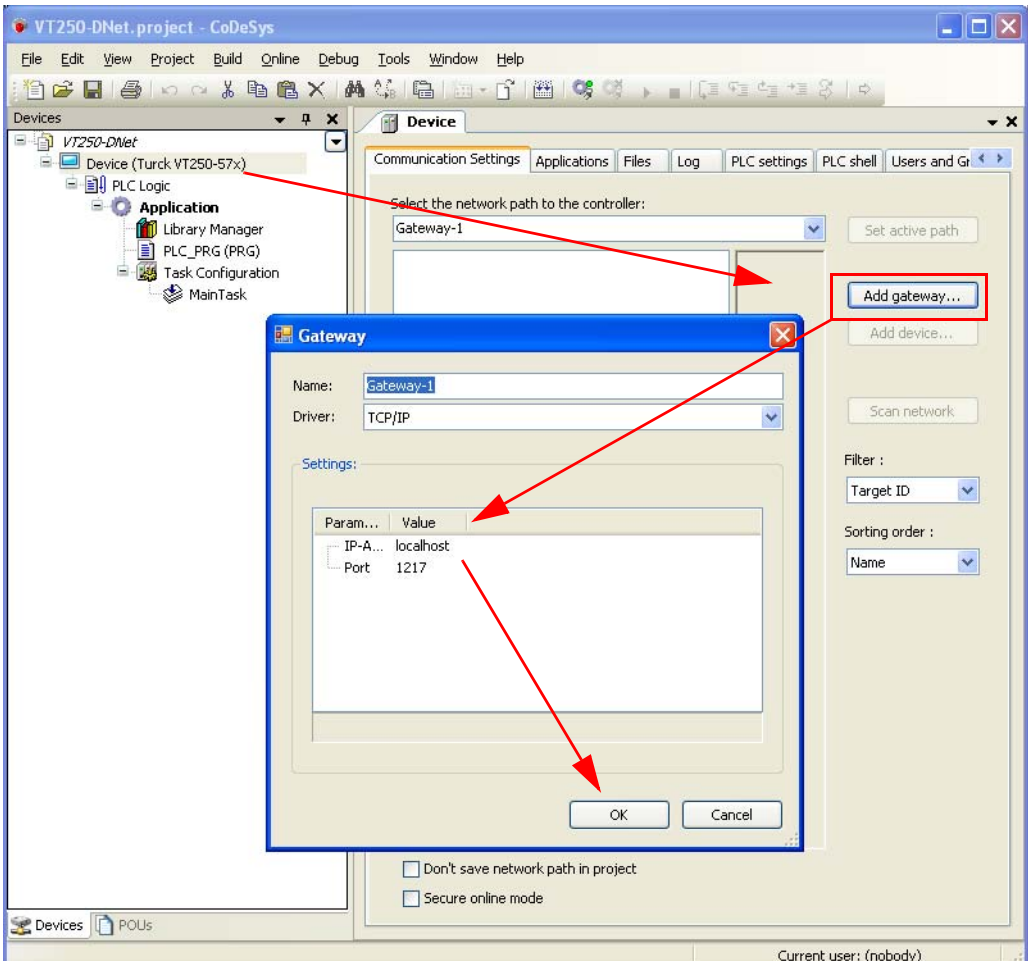
Please use the TURCK IP Address Tool for assigning the IP address to the VT250.
Further information can be found in the hardware manual for the VT250 ((D301191 "VT250 - Hardware-description“).

Gateway definition

1 Use the "Add gateway"-button to open the dialog box "Gateway" and, where necessary, assign a new gateway name.

2 Keep the setting "localhost" or define an IP-address for the gateway instead.
When using the setting "localhost", the CoDeSys-communication-gateway of the PC, on which this CoDeSys-installation is running, is used as programming interface.

Figure 7: Communication settings

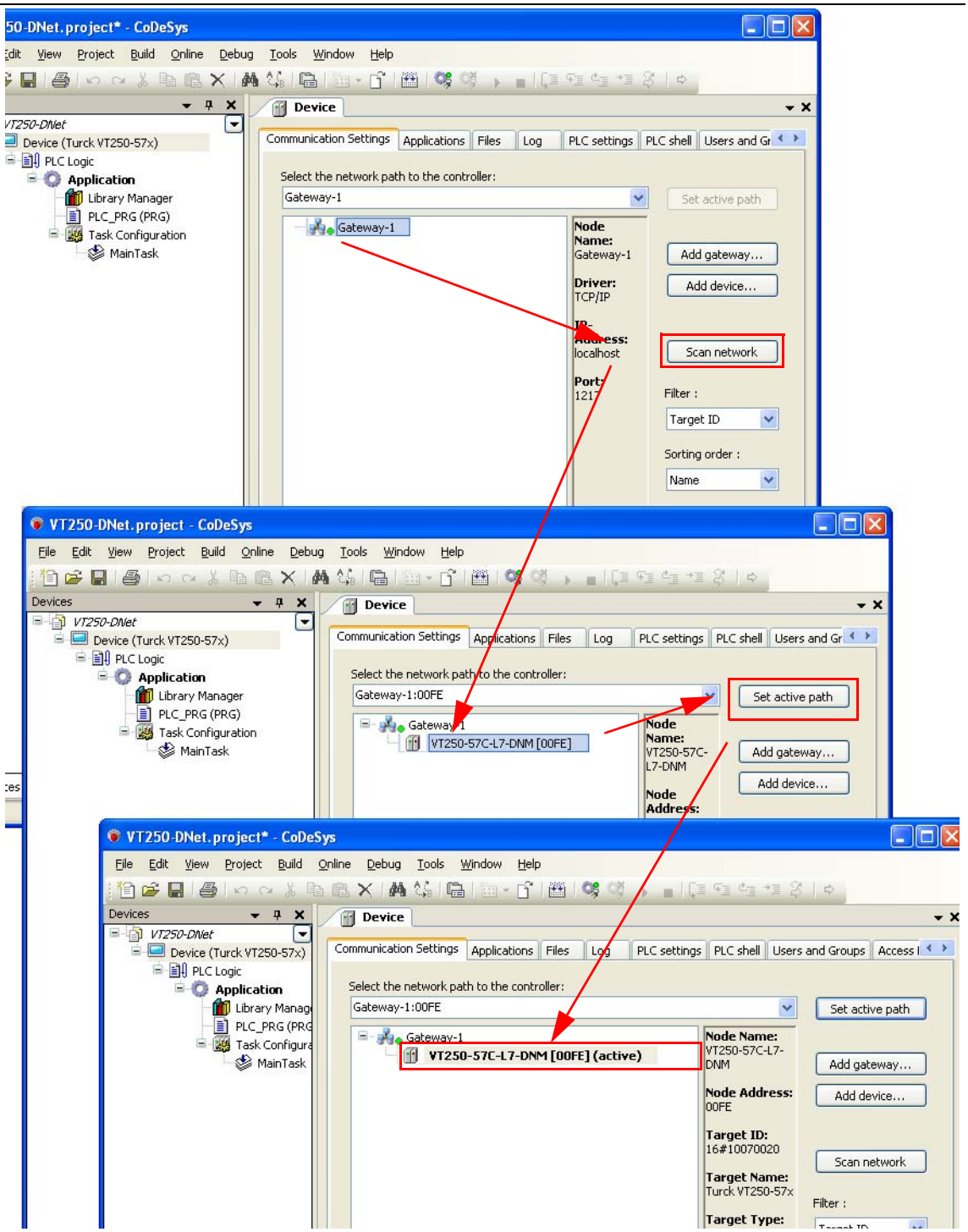


Setting the communication path

1 Mark the gateway and scan the network via the respective button. The connected VT250-57×, in this example the VT250-57C-L7-DNM, is found.

2 Mark the VT250-57x and set the active path using the "Set active path" button.

Figure 8: Setting the communication path



2.2.4 Renaming the application

The operating system rcX uses the 8.3-filename convention.

Therefore, it has to be observed, that the name of the stored application file also corresponds to this convention.

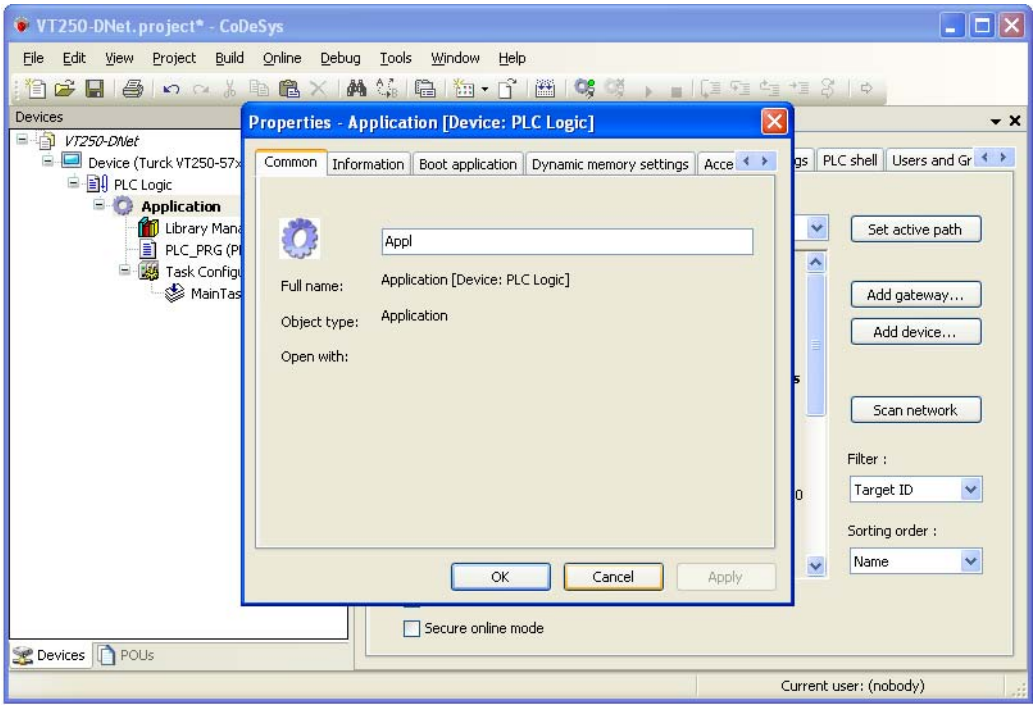
Note

Using an application name, which does not correspond to the 8.3-naming convention, may cause data loss and communication problems between PC and VT250-57×!

1 Open the context menu by right-clicking the "Application" and select "Properties".

2 In the "Common"-tab, define an application name with a maximum of 7 characters and confirm it with OK. The following warning can be confirmed either. The following warning can be confirmed either.

Figure 9: renaming the application



2.2.5 Adding the NetX_CANbus

DeviceNet™-communication is established via the NetX_CANbus from 3S.

1 Open the context menu by right-clicking the entry of the TURCK VT250-57x. In the "add device"-dialog select the NetX_CANbus from 3S under "fieldbusses → CiA CANopen → CiA CANopen CANbus".

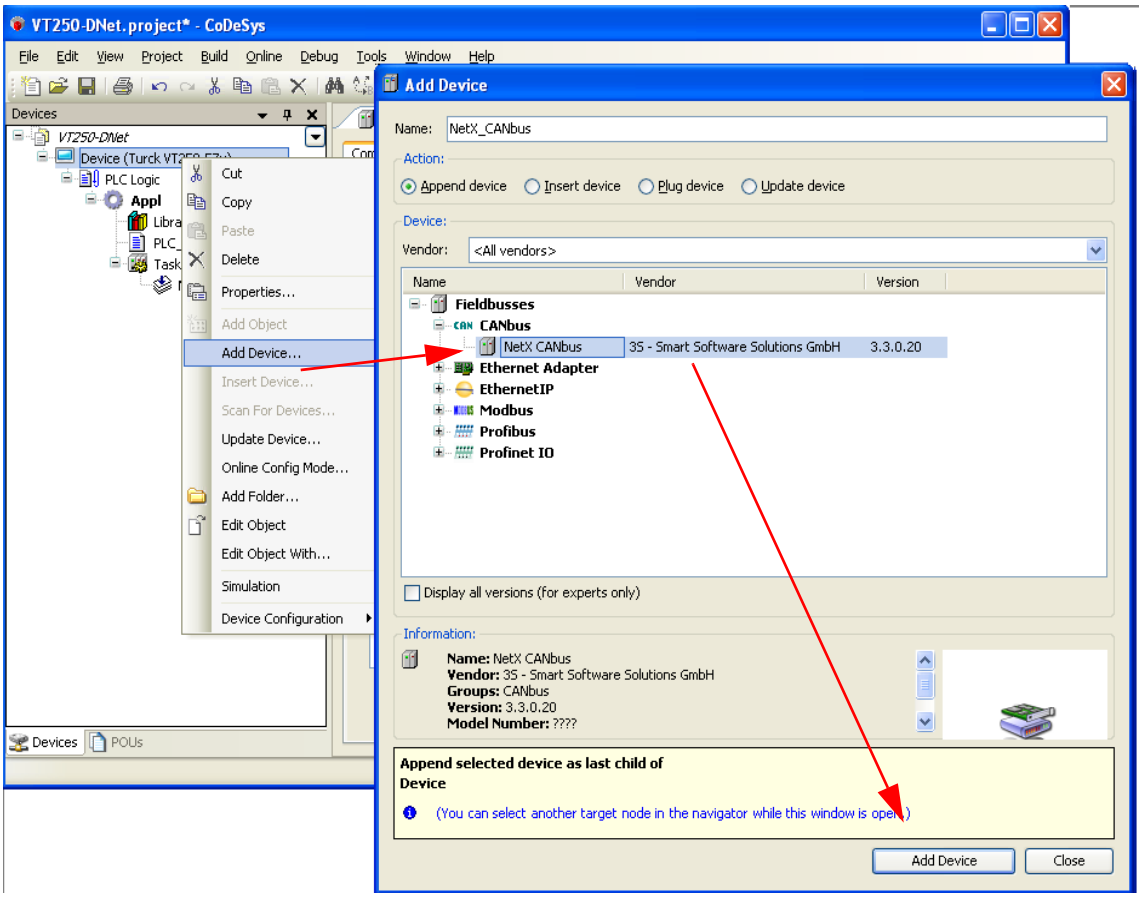


Figure 10: Adding the NetX_CANbus as device

2 Double-clicking the NetX_CANbus-entry in the project tree opens the corresponding editors. Eventually necessary parameterization for the NetX_CANbus (baudrate etc.) can be done in the CANbus-tab.

In this example:

- Baud rate 125 kbps

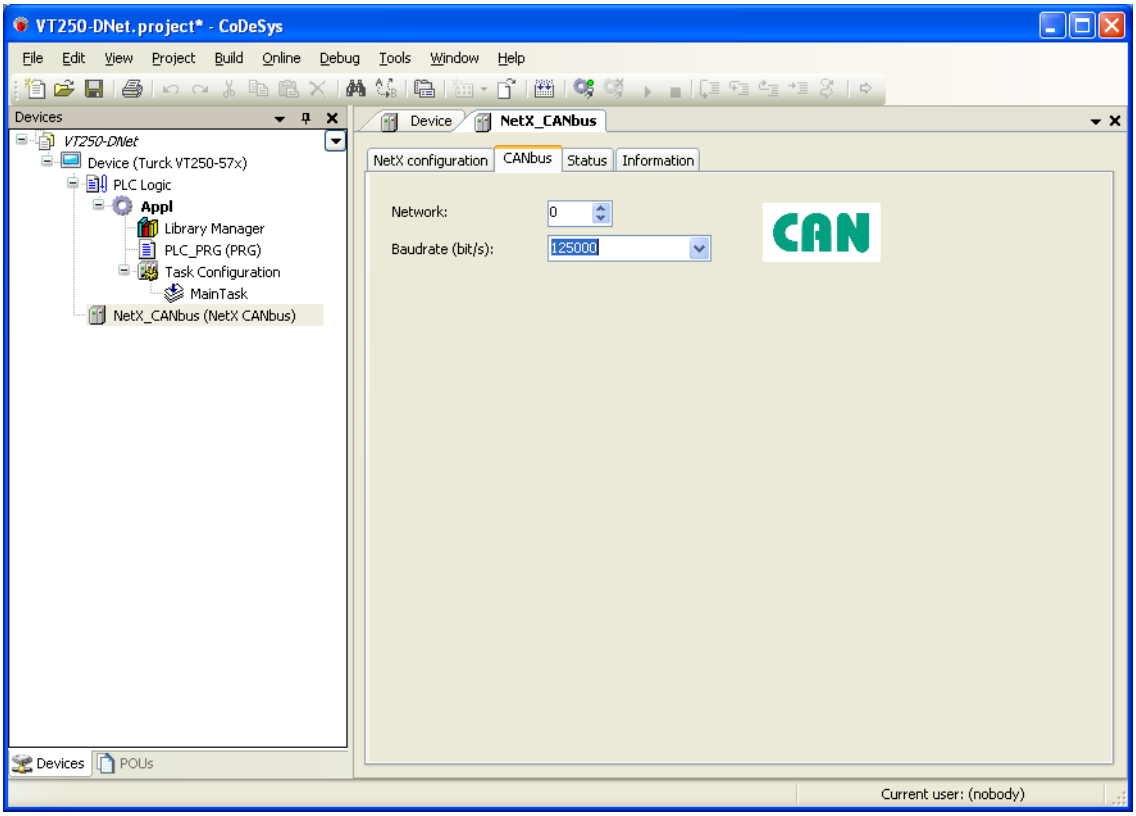


Figure 11: Parameterization of the NetX_CANbus

2.2.6 Adding a DeviceNet™-master

1 Open again the context menu by double-clicking the Device Turck VT250-57x entry. In the dialog "Add Device“, select the DeviceNet™-master under "fieldbusses → DeviceNet → CIFX-DMN" and add it to the project tree.

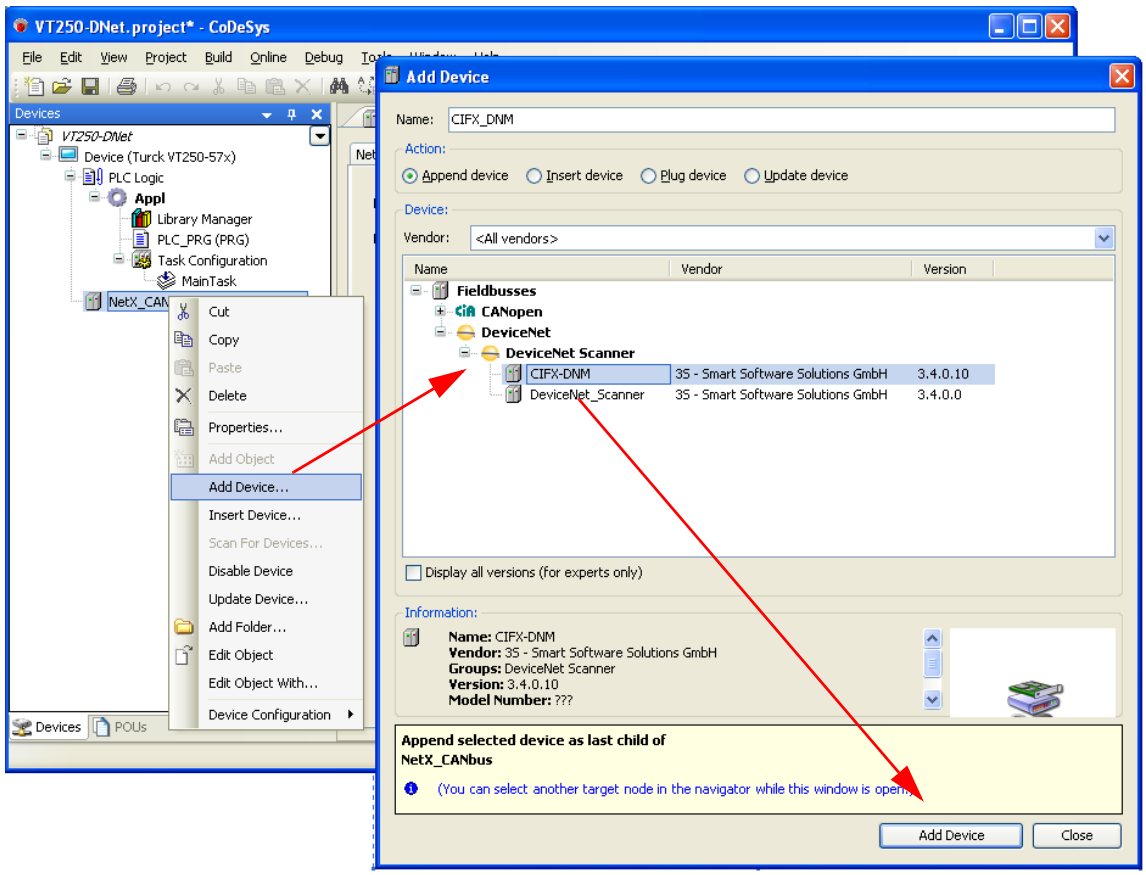


Figure 12: DeviceNet™-the DeviceNet™-master as a device

2 Double-clicking the DeviceNet™-master of the VT 250-57x opens again the respective editors. If necessary, set the DeviceNet™-parameters (MAC-ID etc.) according to your application.

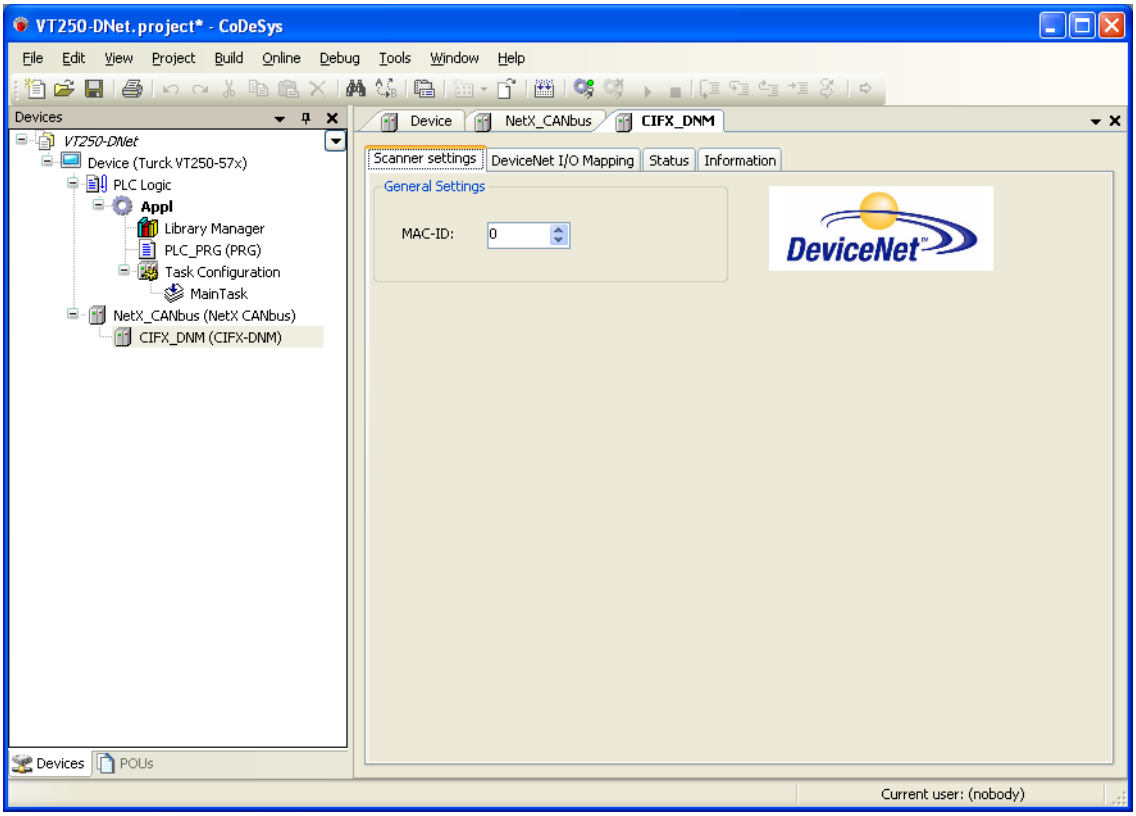


Figure 13: Parameterization of the DeviceNet™-master

2.2.7 Adding DeviceNet™-slaves

DeviceNet™-slaves are added to the DeviceNet™-master in the project tree.

1 A right-click on the DeviceNet™-master opens the context menu. Select "Add device" and add the DeviceNet™-devices to the project tree via "fieldbusses → DeviceNet → DeviceNet Target".

2 If the nodes to be added are not listed in the list of possible DeviceNet™-devices, first of all the corresponding EDS-file has to be added to the CoDeSys "Device Repository".
To do so, please proceed as described in the following.

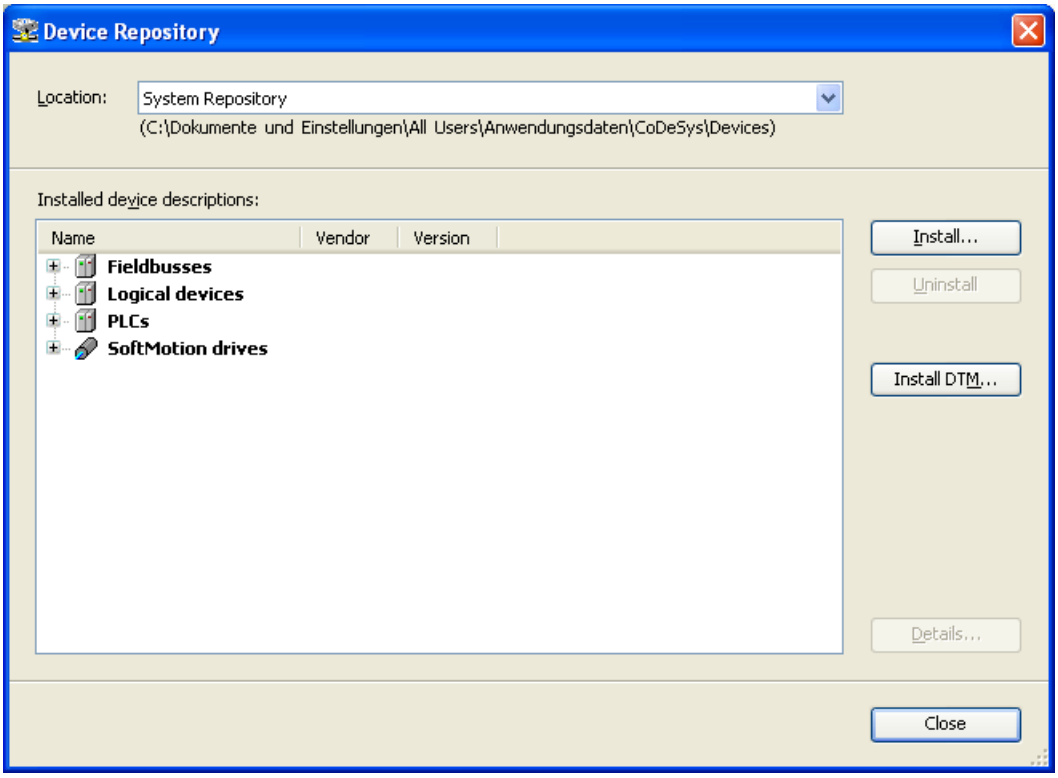
Adding EDS-files to the CoDeSys Device Repository

1 Open the Device Repository via "Tools → Device Repository...".

Note

If CoDeSys is run in Standard mode (see also section „Predefined feature sets“, the GSD files are installed using "Tools → Install device".

Figure 14: Device repository



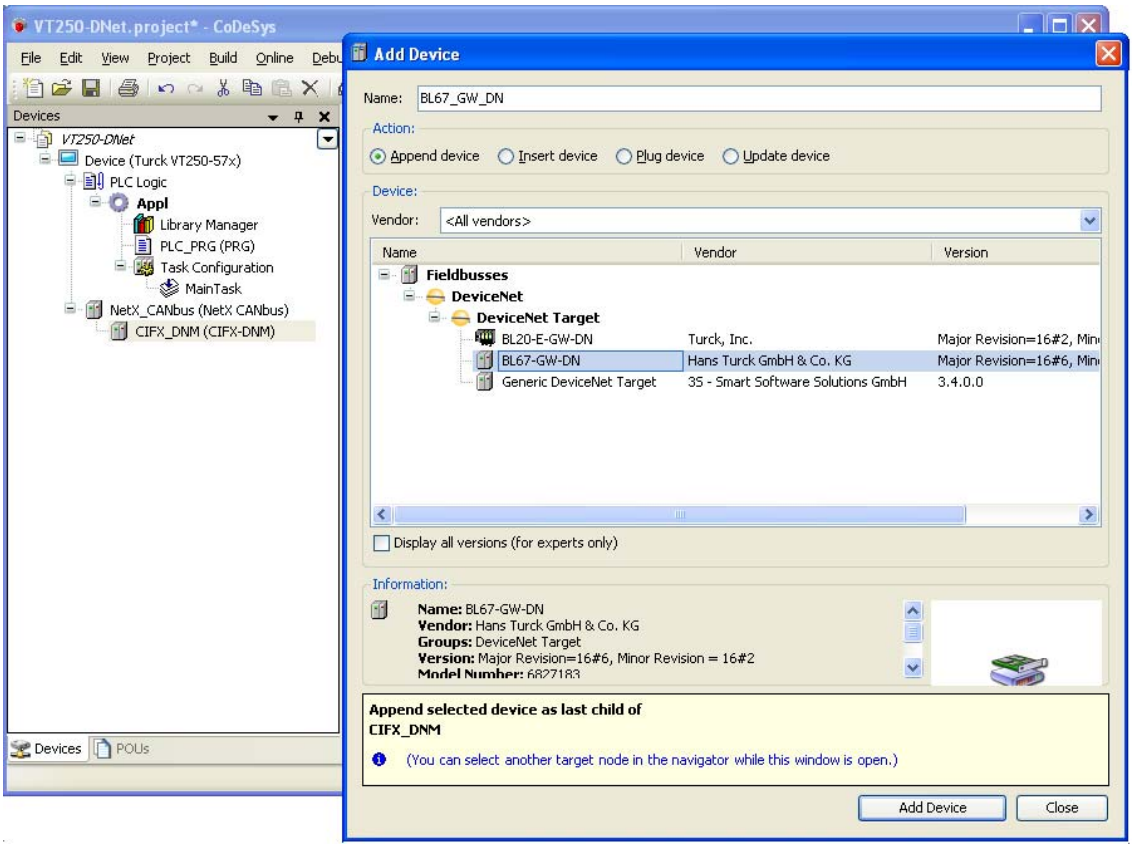
2 The Device Repository contains all currently installed devices. Press "Install", select "EDS and DCF files (*.eds, *.dcf)" under "File type" and search for the files to be installed.

3 After the installation, the installed devices are listed in the "Device Repository" in the "Installed device descriptions"-window under "Fieldbusses → DeviceNet → DeviceNet Target".

Selection of the BL67-gateway

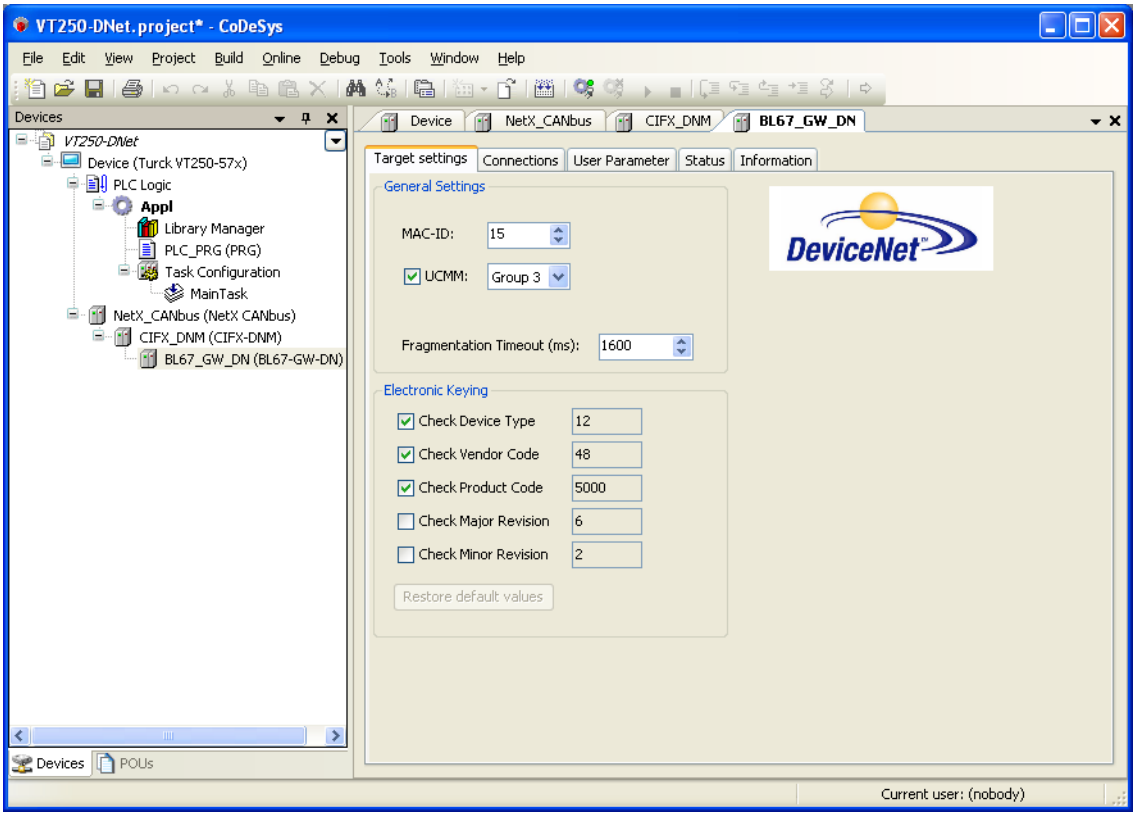
1 Now, add the BL67-gateway (in this example: BL67-GW-DN) to the project by following the description above.

Figure 15: Selection of the BL67-gateway



2 Double-clicking on the gateway-entry in the project tree again opens all corresponding editors. In the "Target settings" tab → "MAC-ID", set the nodes station address (in this example: address **15**) or change other parameter settings, if required, e.g. deactivating the check for Major an Minor Revision (see below).

Figure 16: Example for the parameterization of the gateway



Configuring a BL67-DeviceNet™-station

The configuration of the BL67-station is done by establishing DeviceNet™-connections.

For these connections, the **exact** number of process data for the BL67-station is defined.

In order to determine the process data mapping, use for example the station report function of the I/O-ASSISTANT.

Data mapping

The mapping for the input and output data of a BLxx-DeviceNet™-station depends on it's configuration.

The TURCK-Software "I/O-ASSISTANT" offers the feature of creating a DeviceNet™ report for each BLxx-DeviceNet™-station, which shows the mapping for the respective station.

Example station (see also Seite 2-2):

BL67-GW-DN (MAC-ID 15) with:

- Slot 1: BL67-4DI-PD
- Slot 2: BL67-2AI-I with analog sensor Bi5-M18-Li2
- Slot 3: BL67-8DO-0.5A-P

DeviceNet™-report (I/O-ASSISTANT)

Figure 17: DeviceNet™ report in the I/O-ASSIS-TANT

1. DeviceNet report

1.1. Station description

Station address: 1

Adr./Slot	Name	TAG	Descr.	Data Size In	Data Size Out
Slot 0*	BL67-GW-DN	1/BL67-GW-DN	Term0A	16 bit	16 bit
Slot 1	BL67-4DI-PD	01/BL67-4DI-PD	Term0B	4 bit	0 bit
Slot 2	BL67-2AI-I	02/BL67-2AI-I	Term0C	32 bit	0 bit
Slot 3	BL67-8DO-0.5A-P	03/BL67-8DO-0.5A-P	Term0D	0 bit	8 bit
	Local I/O data incl. status/control			4 Words	2 Words
Total size for in/out data rounded on full words				4 Words	2 Words

*For detailed information about status/control word see online help.

1.2. I/O map for input data

	Byte n+1								Byte n							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Vibrd0*	0A.15	0A.14	0A.13	0A.12	0A.11	0A.10	0A.9	0A.8	0A.7	0A.6	0A.5	0A.4	0A.3	0A.2	0A.1	0A.0
Vibrd1	-	-	-	-	-	-	-	-	-	-	-	-	0B.3	0B.2	0B.1	0B.0
Vibrd2	0C.15	0C.14	0C.13	0C.12	0C.11	0C.10	0C.9	0C.8	0C.7	0C.6	0C.5	0C.4	0C.3	0C.2	0C.1	0C.0
Vibrd3	0C.31	0C.30	0C.29	0C.28	0C.27	0C.26	0C.25	0C.24	0C.23	0C.22	0C.21	0C.20	0C.19	0C.18	0C.17	0C.16

*For detailed information about status/control word see online help.

Process input data: 4 Words

1.3. I/O map for output data

	Byte n+1								Byte n							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Vibrd0*	0A.15	0A.14	0A.13	0A.12	0A.11	0A.10	0A.9	0A.8	0A.7	0A.6	0A.5	0A.4	0A.3	0A.2	0A.1	0A.0
Vibrd1	-	-	-	-	-	-	-	-	0D.7	0D.6	0D.5	0D.4	0D.3	0D.2	0D.1	0D.0

*For detailed information about status/control word see online help.

Process output data: 2 Words

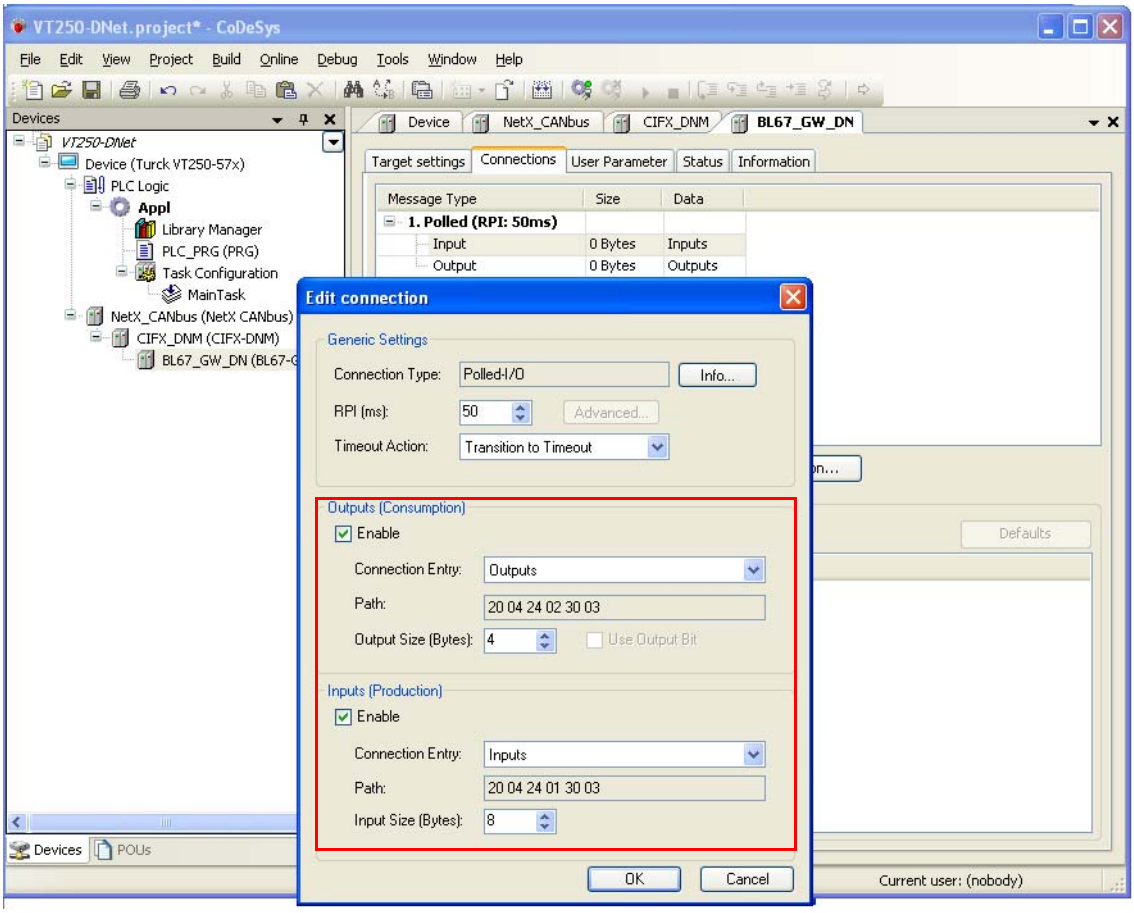
Determined process data width of the example station:

- 4 input words (incl. status word of the gateway)
- 2 output words (incl. control word of the gateway)

These values are entered when establishing a DeviceNet™-connection (see the following page).

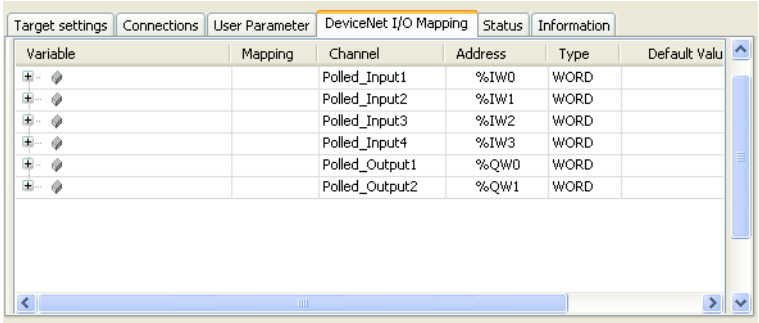
Establishing a DeviceNet™-connection

Figure 18: Establishing the DeviceNet™-connection



3 The "DeviceNet I/O Mapping"-tab shows, only after the definition of the connection, the I/O-mapping for the whole BL67-DeviceNet™-station.

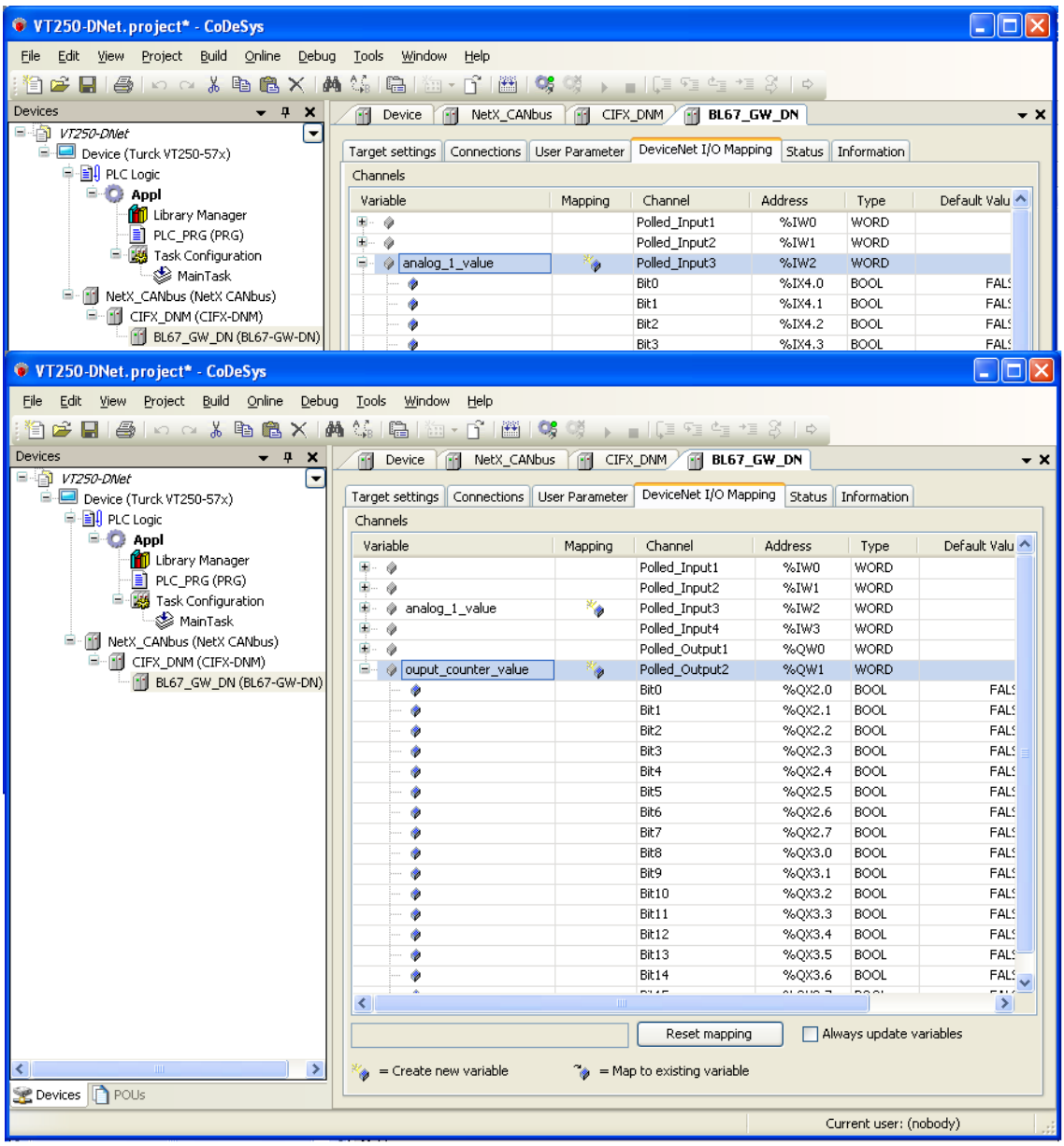
Figure 19: I/O-mapping of the DeviceNet™-station



Global variables

Global variables are defined in the station's I/O-mapping:

Figure 20: Example for the definition of a global variable Variables in the project I/O-mapping



2.2.8 Programming (example program)

The programming is done under PLC-PRG in the project tree. This example is programmed in ST as defined under "Creating a new project".

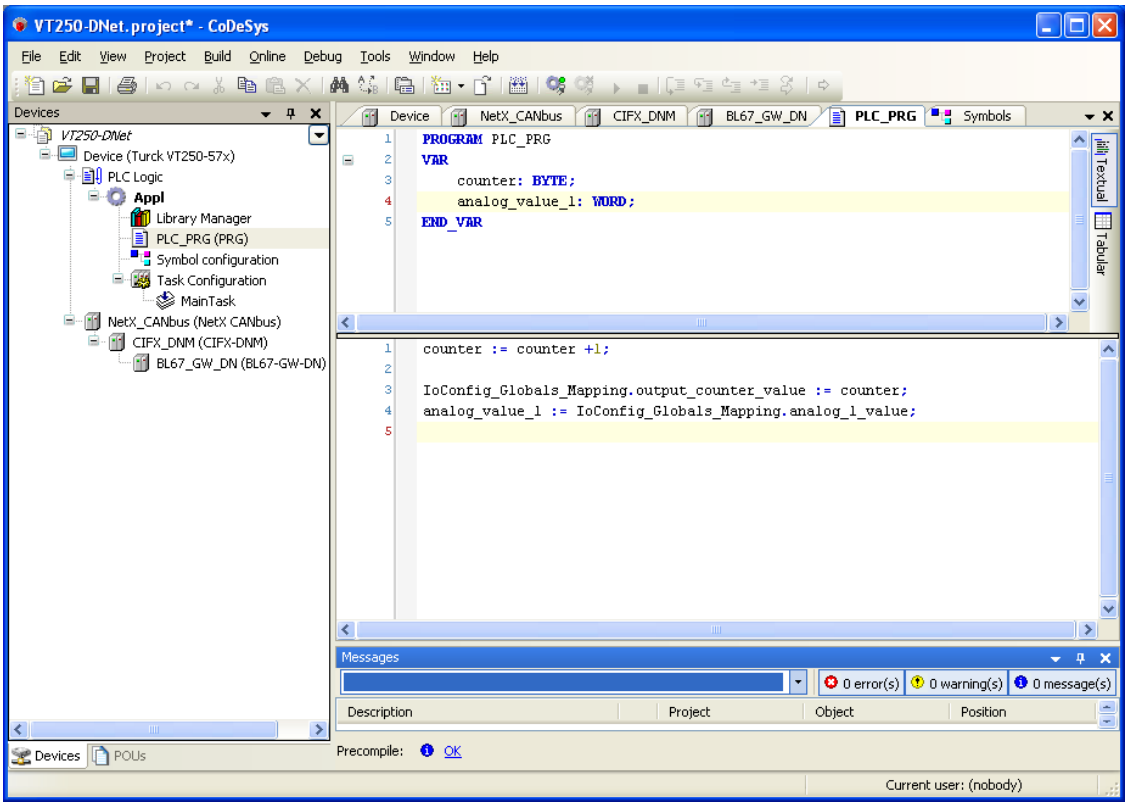
Small example program

1. The counter counts.
2. The counter-value is mirrored to %QW1 (contains the output byte of the BL67-8DO-0.5A-P).
3. The analog value of the sensor at channel 1 ("channel_1_value", global variable, &IW2) of the analog module at slot 2 in the BL67 station is read out.

Note

The status of process values is only shown in the process image if a program refers to them or if the function "Always update variables" in the "DeviceNet I/O Mapping" tab of the gateway is enabled.

Figure 21: Example program

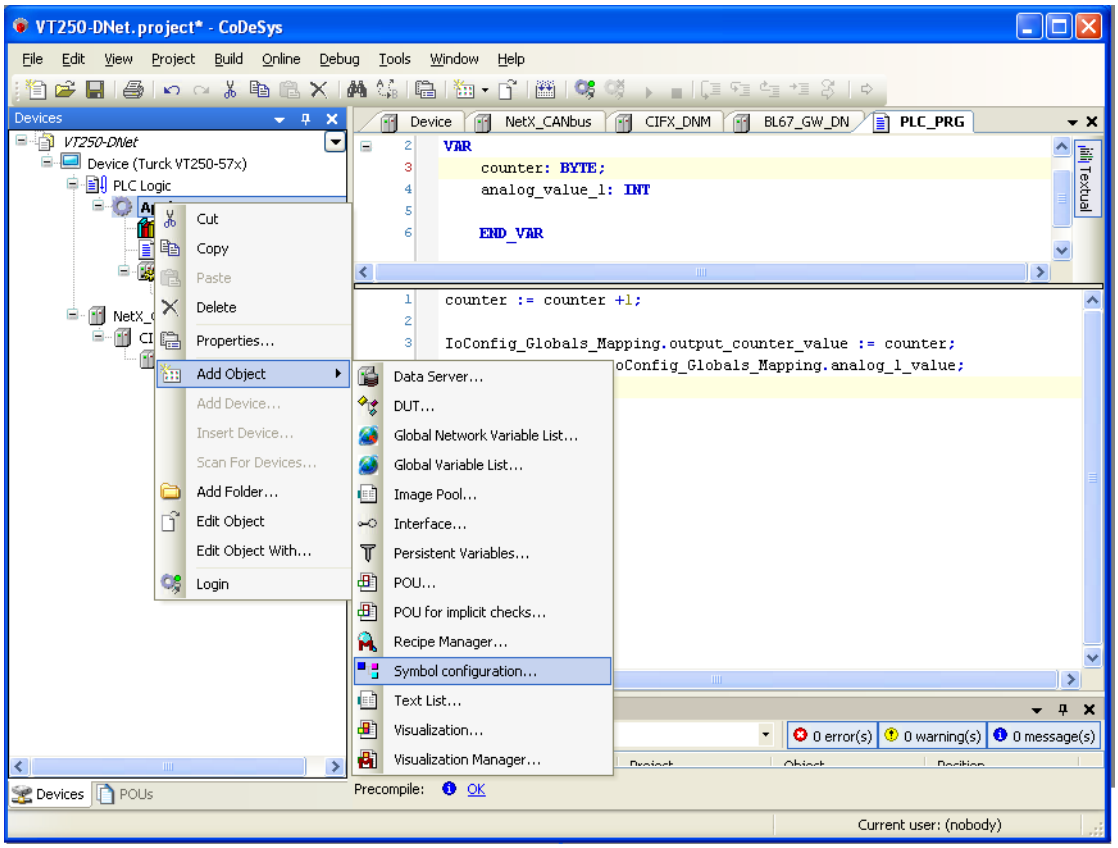


2.2.9 Export of variables

The variables defined in the program have to be exported, for example in order to be used within an visualization.

1 Therefore a symbol configuration has to be added to the project. This is done via right-click on "APPL→ Add object → Symbol configuration". The next dialog can be confirmed.

Figure 22: Add symbol configuration



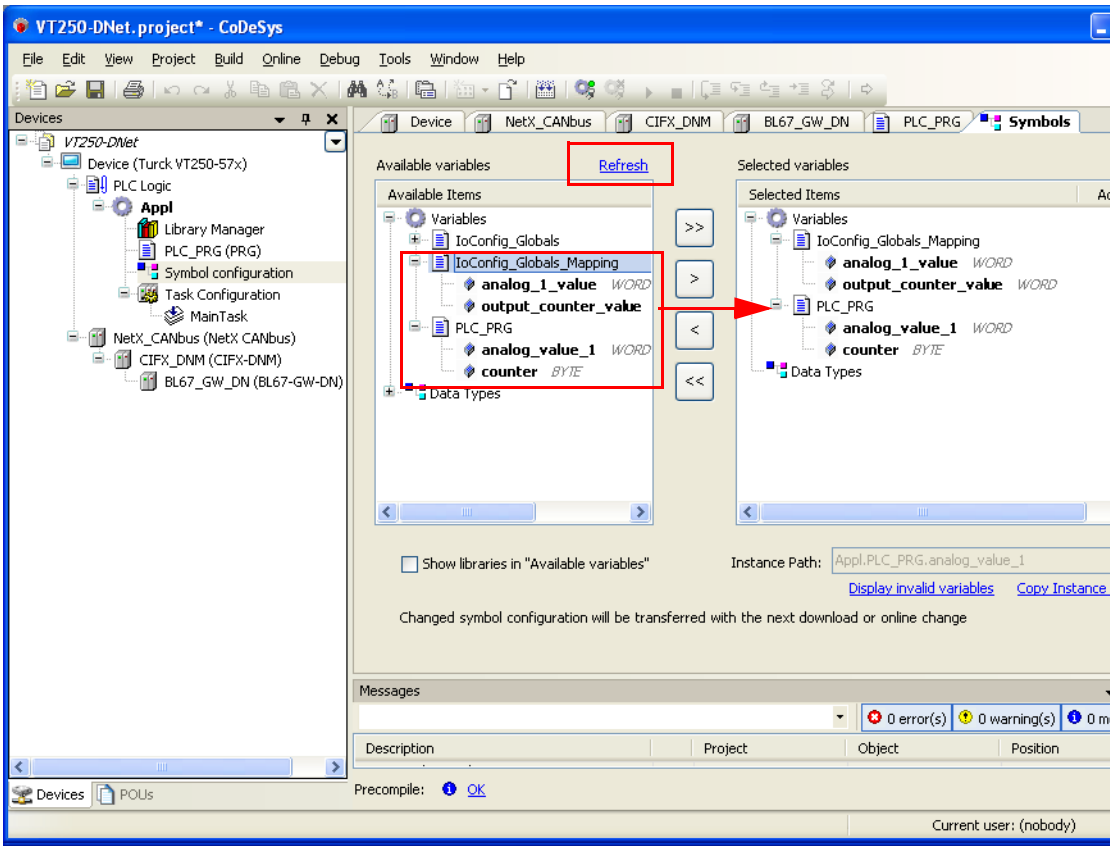
2 The symbol configuration shows all available variables (left side of the window) and allows the definition of all variables to be exported (right side of the window).

3 Mark the variables to be exported and add them to the window "Selected variable" by using the arrow keys.
If any variable is missing, please "Refresh" the variable list.

Note

In order to avoid unnecessary loading of the internal data transfer between CoDeSys and the QViS runtime, only variables which will effectively be used in the visualization project should be added to the symbol configuration.

Figure 23: Variables are selected



4 The *.xml-file of the symbol configuration is built or respectively updated within the project folder with every project download.

Global variable list

The creation of a "Global Variable List" is possible, too:
right-click to "APPL → Add object → Global Variable List".

Define the global variables. The global variables are also automatically exported when building the project if they,have been chosen for export in the symbol configuration. (see also Figure 23:, Variables are selected).

2.2.10 Downloading the project

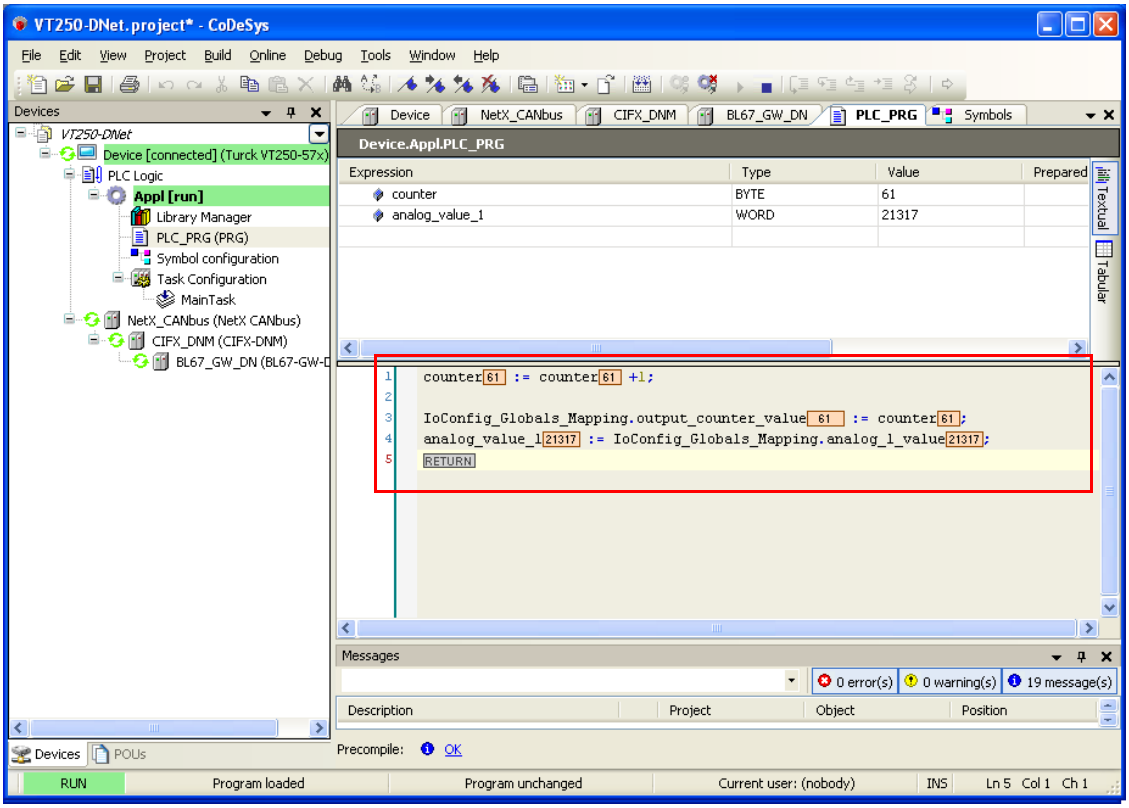
1. Building the project:
   Building the project is done via "Build → 'Rebuild ....".
2. Log-in:
   Log-in: The connection to the device is established via "Online→ Login". The application is downloaded.
3. Create boot application:
   Create boot application: A boot project is created via "Online → Create boot application" and is then directly written into the device

Note

A boot application can only be created in online-mode.

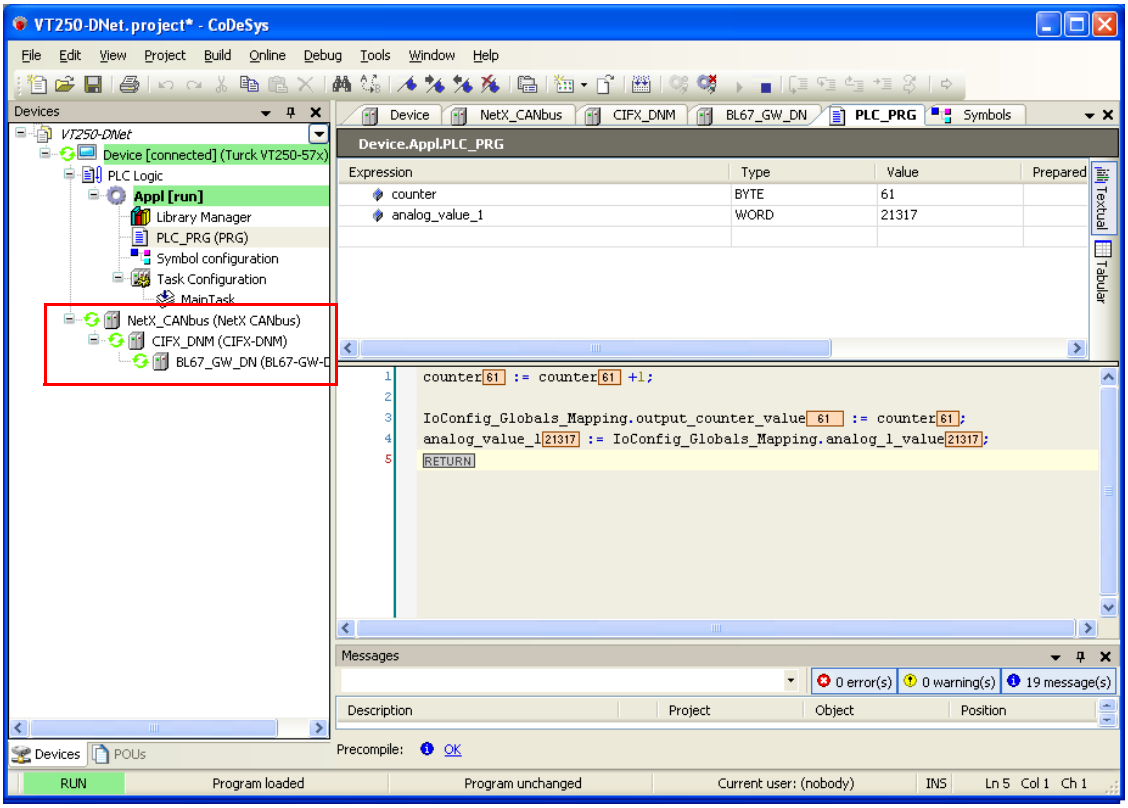
4. Starting the program:
   Start the program using "Online → Start ".
5. The program is running (see PLC_PRG in project tree).

Figure 24: Program running



6 CoDeSys 3 also shows that the configuration of the DeviceNet™ is correct.

Figure 25: DeviceNet™ OK



2.3 Visualization with QViS

2.3.1 QViS example project

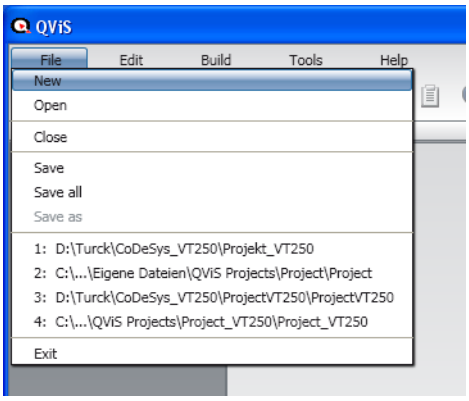
Note

An example project, which explains all important software functions, is installed with the QViS installation. It can be found in the software's program directory "C:\Programs\Hilscher\QViS_3.1.1.x\Resource\Example1_320x240“.

2.3.2 Create a new project

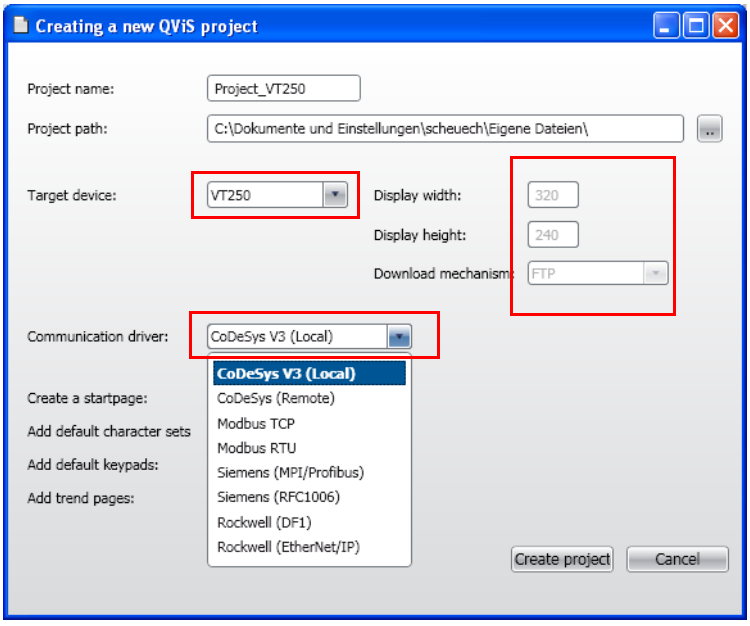
1 Create a new project via "File → New". In the new dialog box, assign a project name and define the storage location of the project.

Figure 26: New project



2 The following settings have to be done:

Figure 27: Project configuration



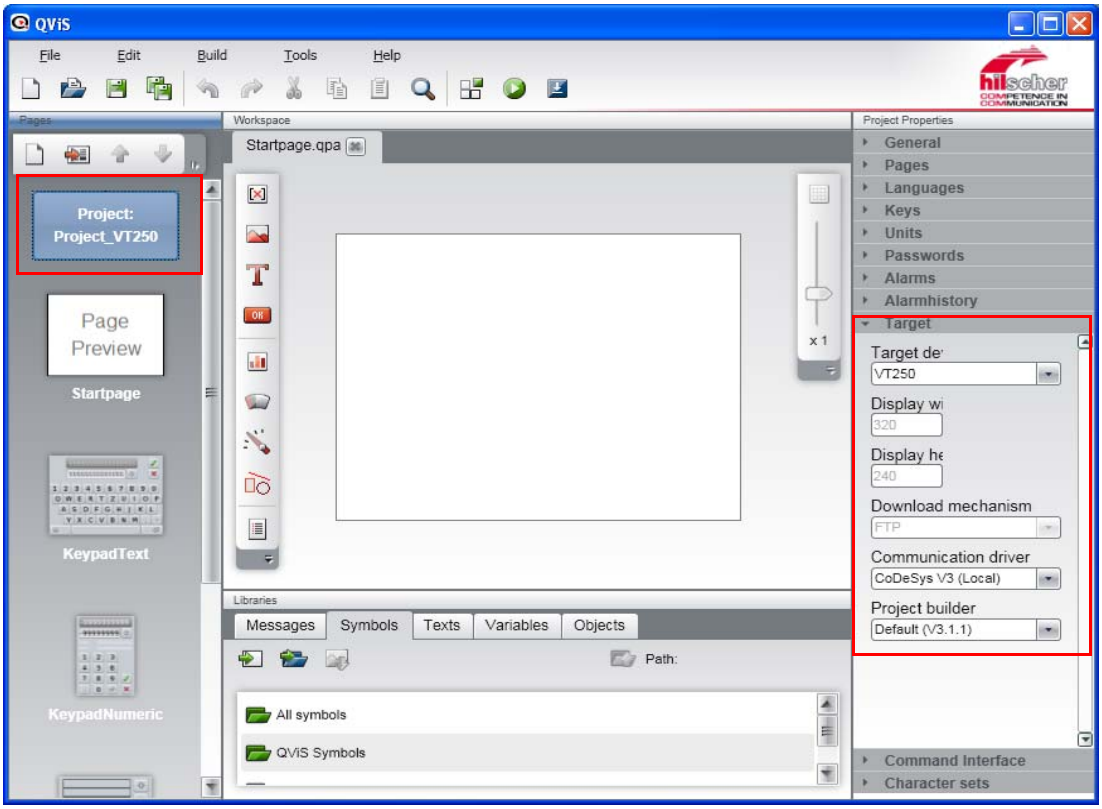
– As "Target device" please select the VT250-57x. The display size is automatically set to 320×240.

– The visualization-download to the HMI is done via FTP-access. This is also automatically set under "Download mechanism“.

– Under "Communication driver" select "CoDeSys".

3 A double-click onto the project opens the project file properties. Under "Target" you can find amongst others the settings done before

Figure 28: Target Device settings



2.3.3 Elements in QViS

Table 1: Elements in QViS

Function	Description
	"Insert a variable" Display of a variable as numeric value or text. The element variable is linked to the imported variables from the PLC program. see Variables (page 2-34)
	"Insert a symbol" – static symbol: figure, e.g. background, logo, etc. which is always shown during runtime – dynamic symbol: linking of several figures to a variable (e.g. for status indication) see Symbols (page 2-31)
	"Insert a text" – static text: text which is always shown during runtime – dynamic text: several texts can be linked to a variable (e.g. for status indication) see Texts (page 2-33)
	"Insert a button" Push-button or switch which can be operated by the user via touch or mouse pointer and then triggers a function. see Buttons (page 2-36)
	"Insert a bargraph" Display of a bar graph. The bargraph is linked to a variable and changes depending on the variable value. see Bargraph etc. (page 2-35)
	"Insert an arc" Display of an arc. The arc is linked to a variable and changes depending on the variable value.
	"Insert a pointer instrument" Display of a pointer instrument. The pointer instrument is linked to a variable and changes depending on the variable value. A scale can be added to the pointer instrument using the ScaleMaker. see Pointer instrument (page 2-37)
	"Insert a vector graphic" Adding vector graphics (line, circle, etc.).
	"Insert a list" Display of: – lists with several variable values – list of alarms or of alarm histories – lists for creating an input window
	"Insert a trend" Graphic display of the trend of variable values.

2.3.4 Objects in QViS

Objects are partial masks which consist of one or several elements.

Compared to elements (variable, symbol, text, etc. see Elements in QViS (page 2-27)), objects have an x/y-position and, if linked to variables, can change their position during runtime.

Object are particularly reasonable if element groups have to be used several times on different pages of the QViS project.

Table 2: Objects in QViS

Function	Description
	"Insert an object" Creating a new object.
	"Insert an existing object" Adding an already made object.

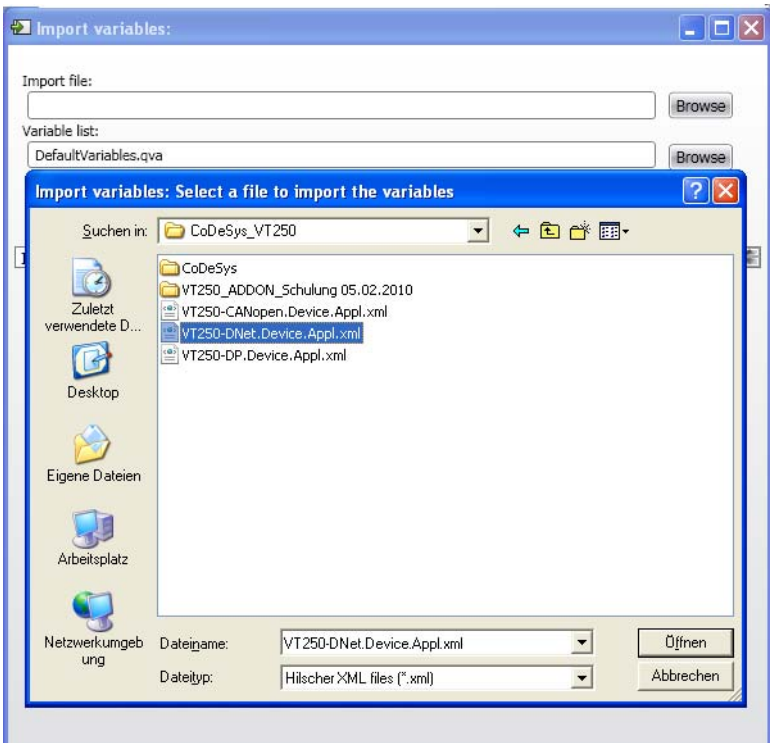
2.3.5 Variable import

In order to have access from the visualization to the variables of the CoDeSys-project created before, the symbol configuration ("project name.Device.Application.name.xml") exported in CoDeSys has to be imported.

1 The import is started via "Tools → Import"

2 Under "Import file" select the exported-file from CoDeSys (here in the example: VT250-DNet.Device.APPL.xml).

Figure 29: Variable import

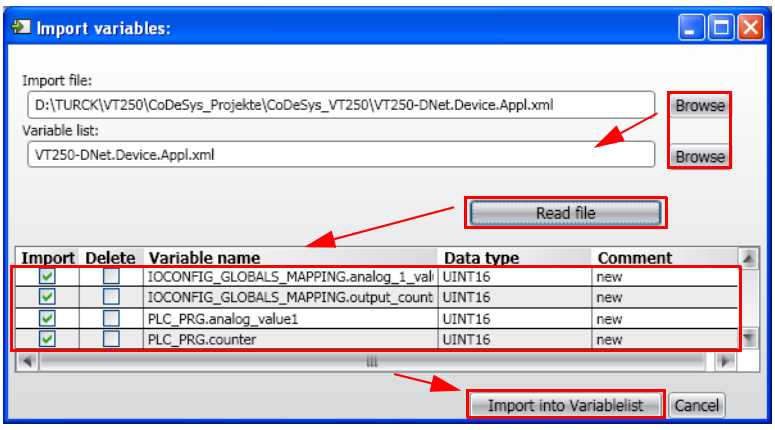


3 Under "Variable list" define the list into which the variables have to be imported (here "DefaultVariables.qva").

4 "Read file" reads in the imported variables.

5 "Import to Variablelist" starts the import of the marked variables into the defined variable list.

Figure 30: Variable import



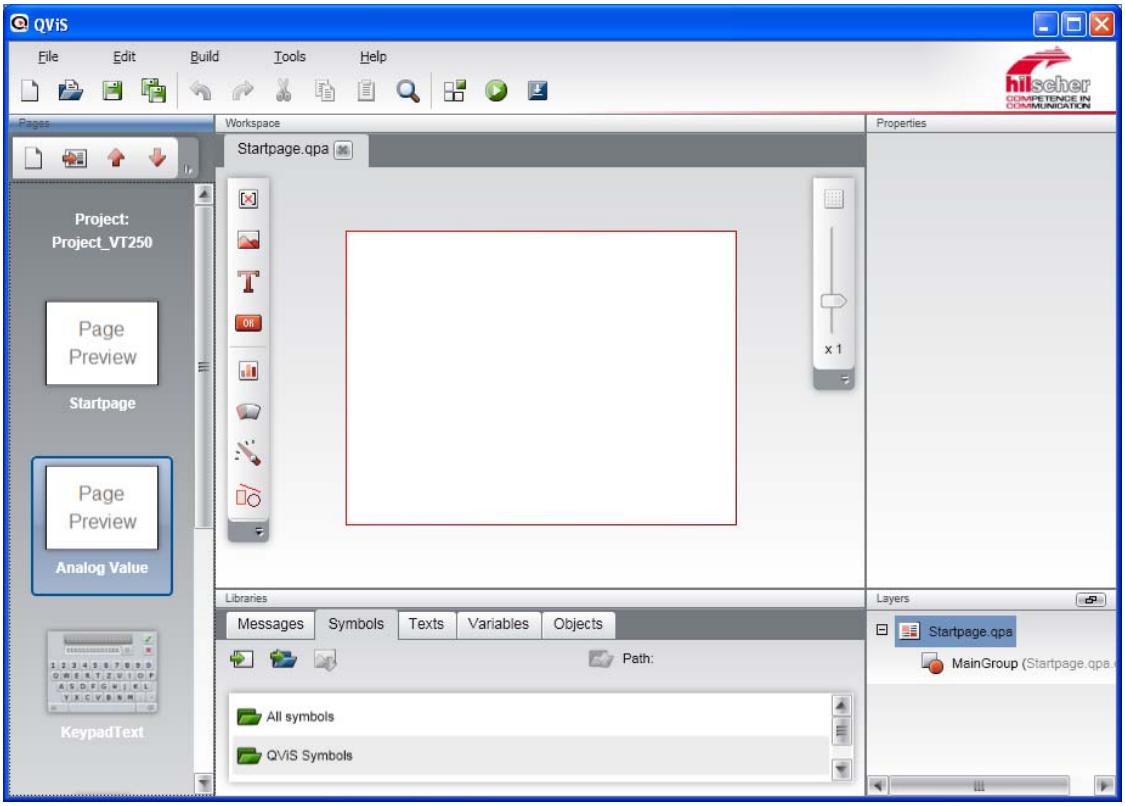
2.3.6 Add pages, rename etc.

Pages are added to the project using the "create new page" button. Via right-click, all pages can be renamed, deleted or removed.

In the example:

A new page is added and named "Analog value".

Figure 31: New page



2.3.7 Visualization

The visualization starts on the first page within the project, here in this example on the "Startpage".

Note

If needed, change the startpage for your project under "Project properties → Start page“.

Symbols

1. In this example, first of all, a background for the single pages of the visualization is imported. To realize this, a static symbol is used.
2. The symbols to be used are added to the project in the "Symbols"-tab within the "Library"-frame using the "Add"-button.

Note

Please observe that QViS only supports symbols in *.bmp-format.
Large symbols (surface and/ or file size) have negative influence on the performance/ the page reproduction of the visualization.

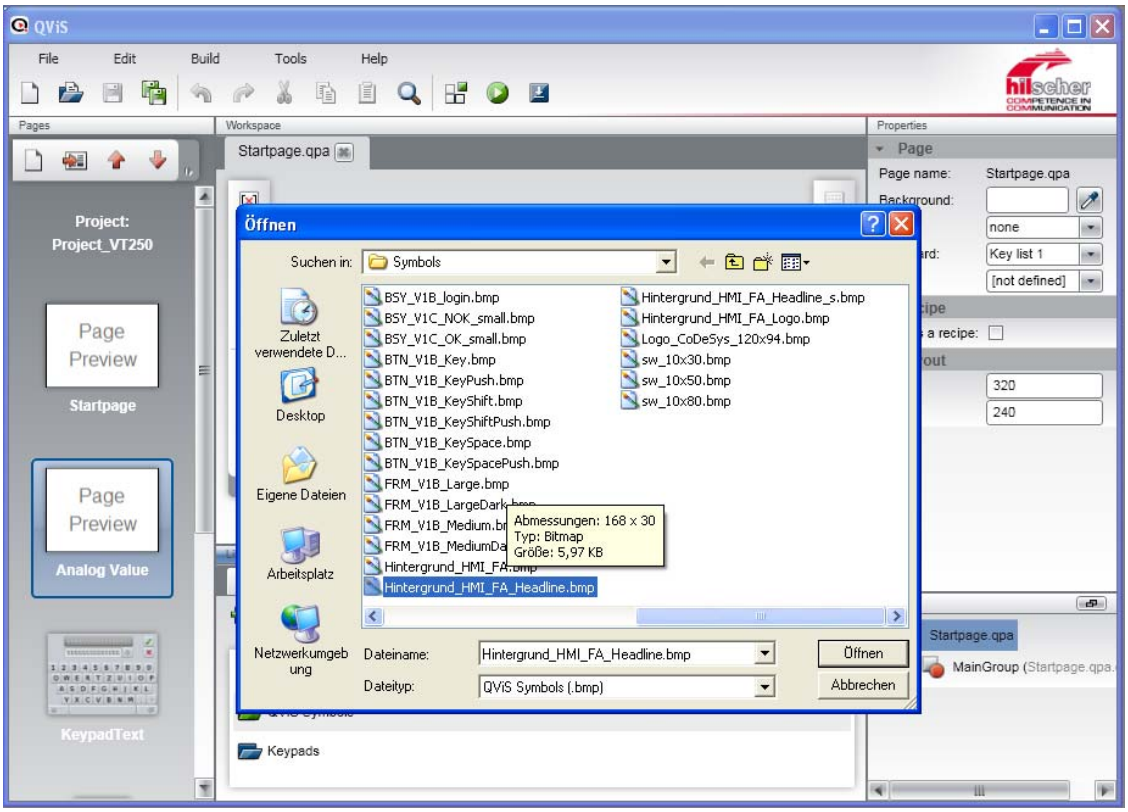
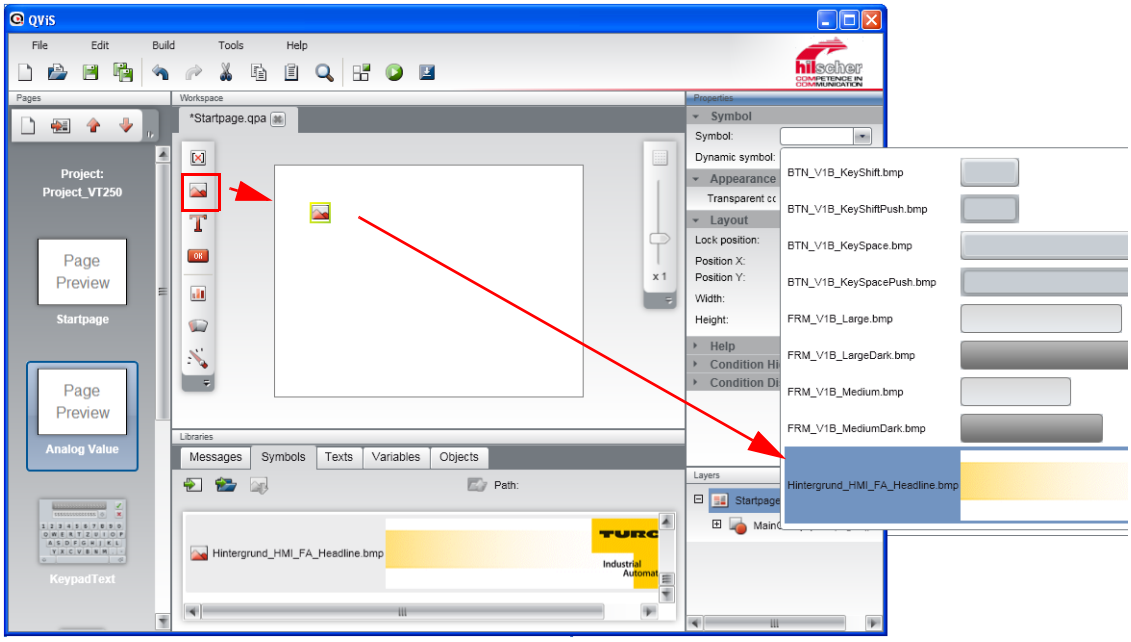


Figure 32: Set background as symbol background as symbol

3 The symbol is placed on the page using the "Insert a symbol"-button and is then defined in the "Properties"-frame.
For the example, the background-file is selected here.

Figure 33: Set background as symbol

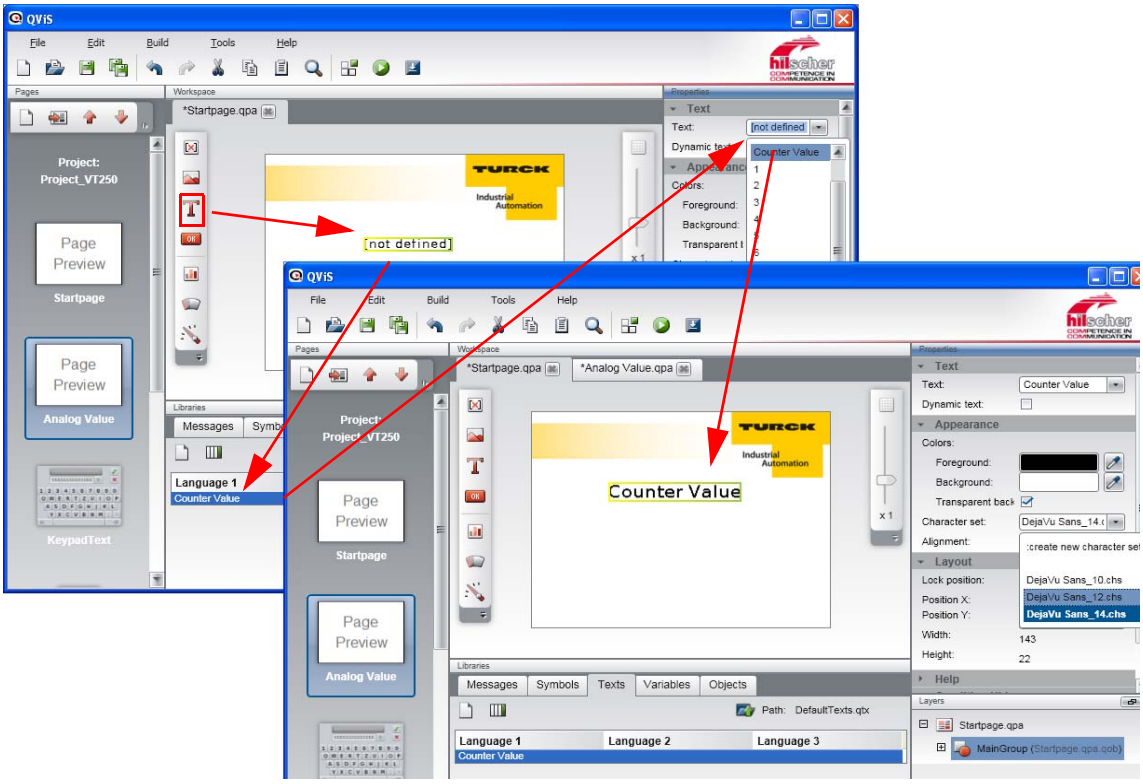


Texts

Texts have to be defined in the "Texts"-tab

1. With "New" a new row is created. The text can then be freely defined for different languages. A double-click onto the respective text makes it possible to edit the entries.
2. The texts are placed on the page using the "Insert a text"-button. In the "Properties"-frame, the text to be shown is assigned to the text-field (here "Counter Value") and properties such as the character set and the position are defined.

Figure 34: Definition of texts

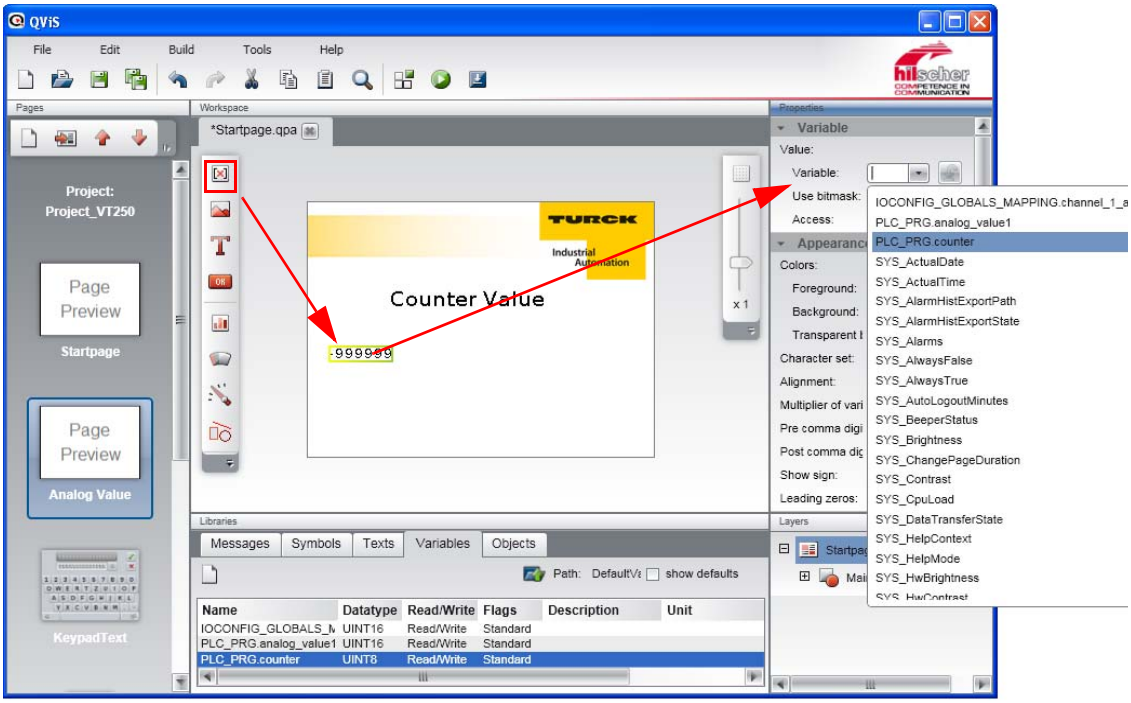


Variables

The imported variables are shown in the "Variables"-tab, if the respective variable list is selected (here "DefaultVariables.qva", see also section "Variable import"section „Variable import“).

1 The variable is placed on the page using the "Insert a variable"-button. In the "Properties"-frame, the variable to be shown is assigned to the variable-field (here "PLC_PRG.Counter") and properties such as the character set and the position can be defined.

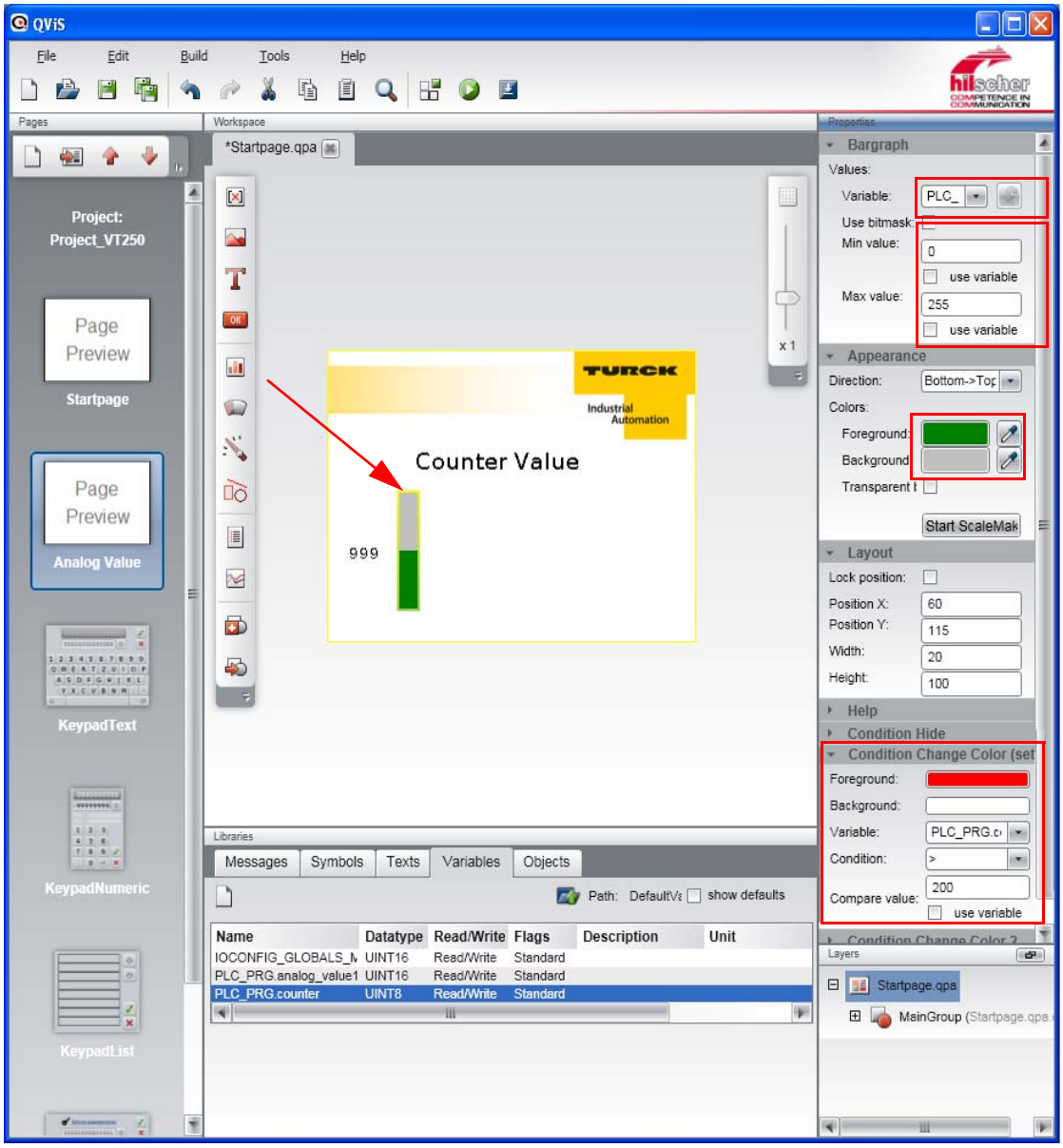
Figure 35: Definition of variables



Bargraph etc.

1 A bargraph is placed on the page by using the "Insert a bargraph"-button. In this example it is used for graphic display of the counter value (variable PRG_PLC.Counter) and is therefore also assigned to this variable. The "Limit minimum" is defined as 0, the "Limit maximum" as 255.

Figure 36: Insert a bargraph



2 Additionally, under the "Properties→ Appearance", the graph's color (here green) can be defined and under "Condition Change Color" it is defined at which value the graph for example changes the color.

Buttons

Buttons are placed on the page using the "insert a button"-button.

Under "Properties", the text, the size, the position etc. can be defined.

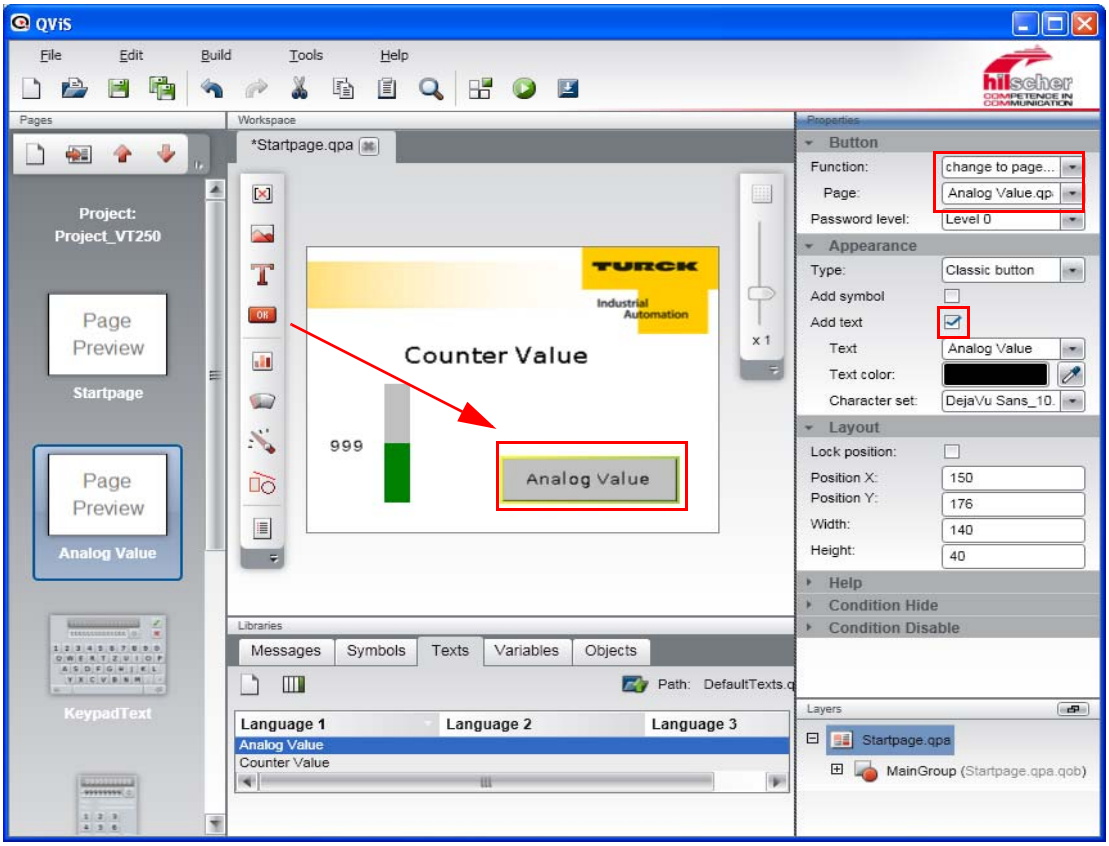
Additionally, the button-function (e.g. change page, setting a variable etc.) is defined.

In the example:

1 On the startpage, a button (size 150×40) is created. It will be used to execute a change to the second page "Analog value".

- function: "change to page“
- Page: "Analog Value“

Figure 37: Insert a button



2 The button is labeled with the text "Analog Value" (for the procedure of defining texts, please read above section „Texts“).

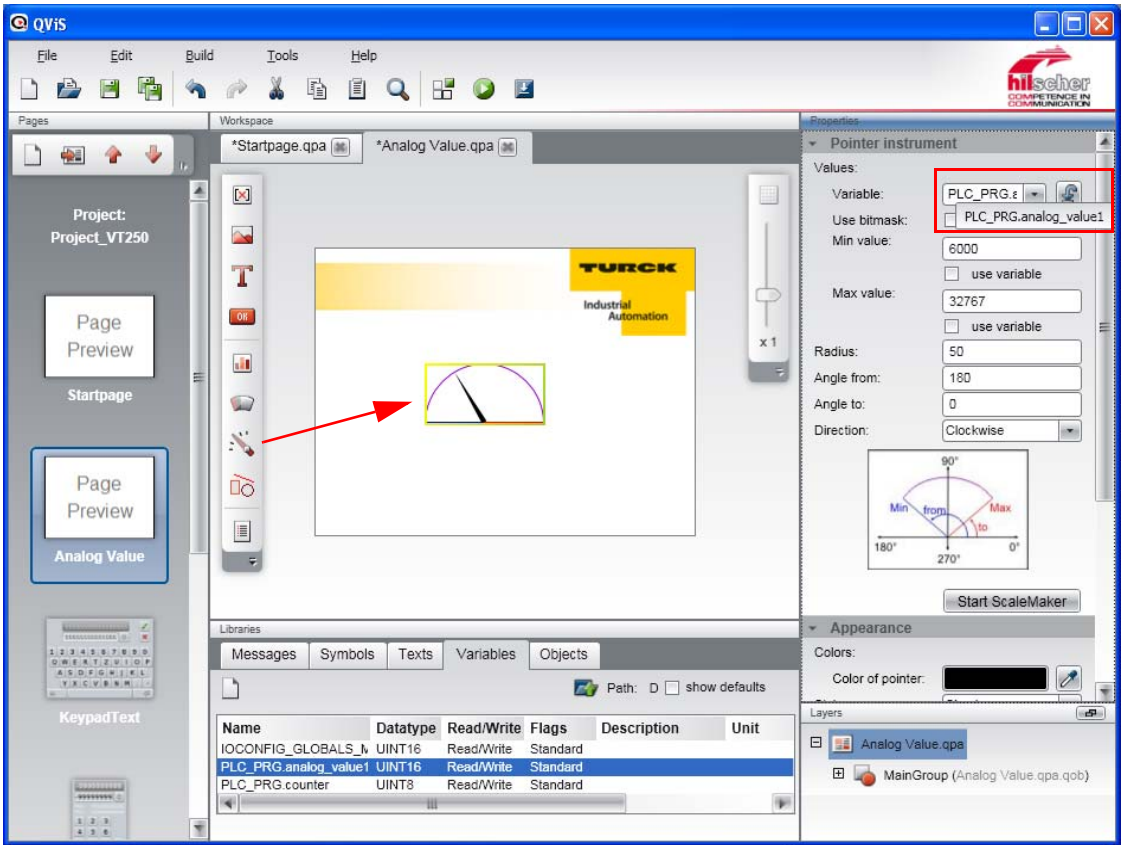
3 To do so, activate the option "add text" under "Properties → Apperarance“ and select the text defined before under "Text".

Pointer instrument

1 On the second page "Analog Value", a pointer instrument is added to display the analog value at channel 1 of the analog module in the station. It is linked to the variable "PLC_PRG.analog.value1“.

2 Depending on the application, the pointer value can be customized by changing the options "Min. value", "Max. value", "Radius", "Angle from" "Angle to" etc. .

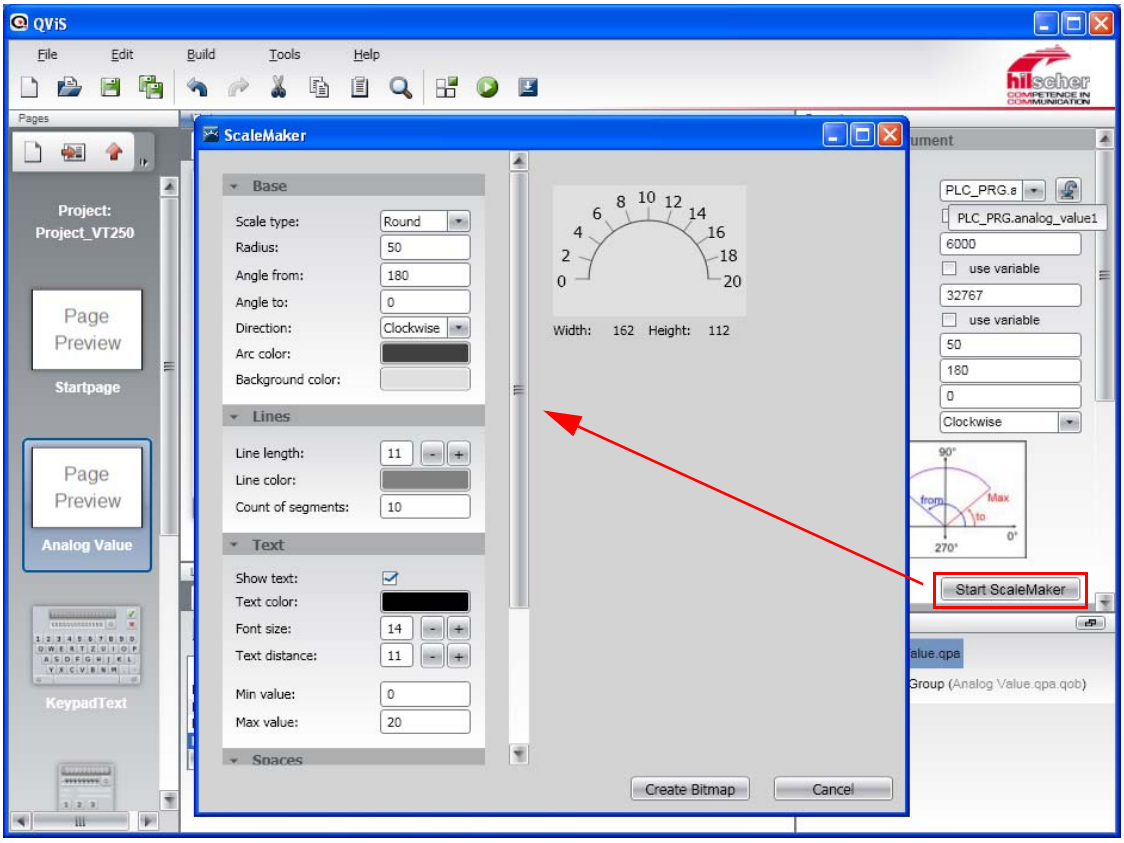
Figure 38: Adding a pointer instrument



3 Additionally, a scale can be added to the pointer. This is done by using "Start Scale Maker“-function.

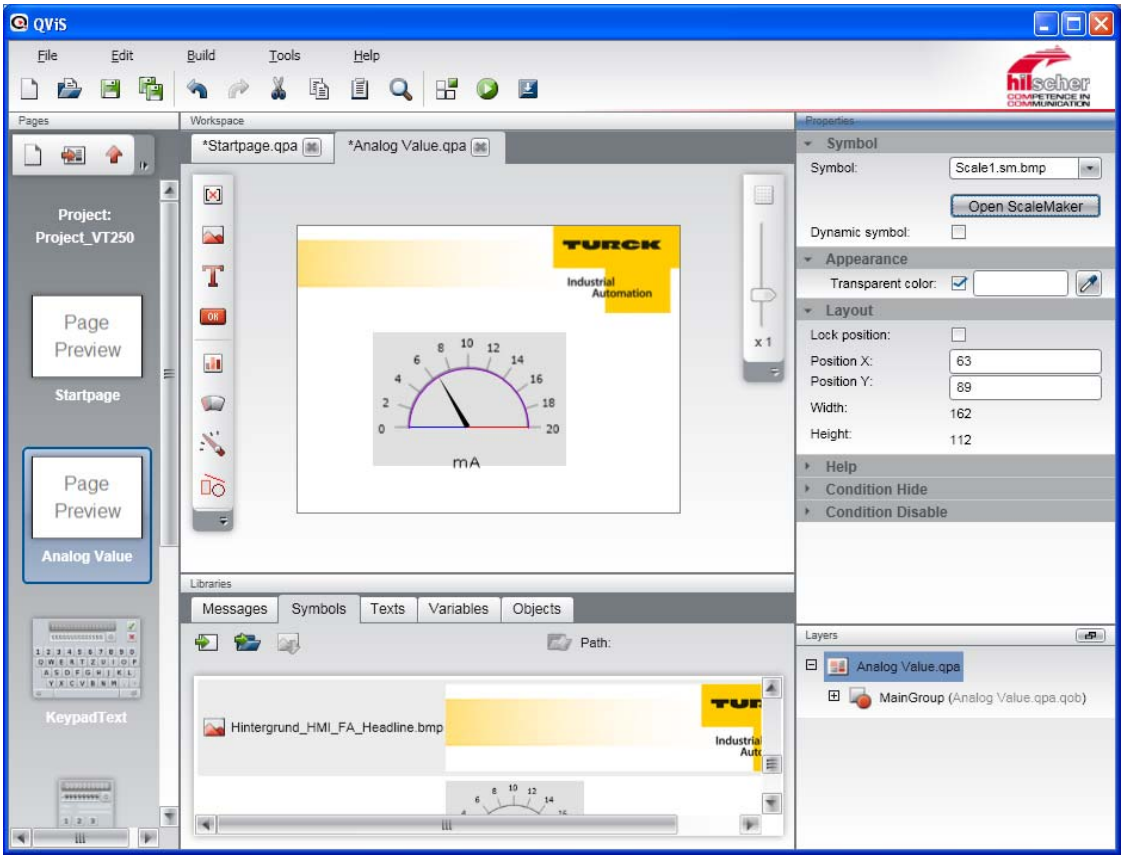
4 Here, enter the values for the needed scale, depending on the application.

Figure 39: Scale Maker, example settings



5 In addition to that, in this example, a further text "mA" is added. To add the text, please proceed as described under Texts (page 2-33).

Figure 40: Display of analog value



"Back“-button

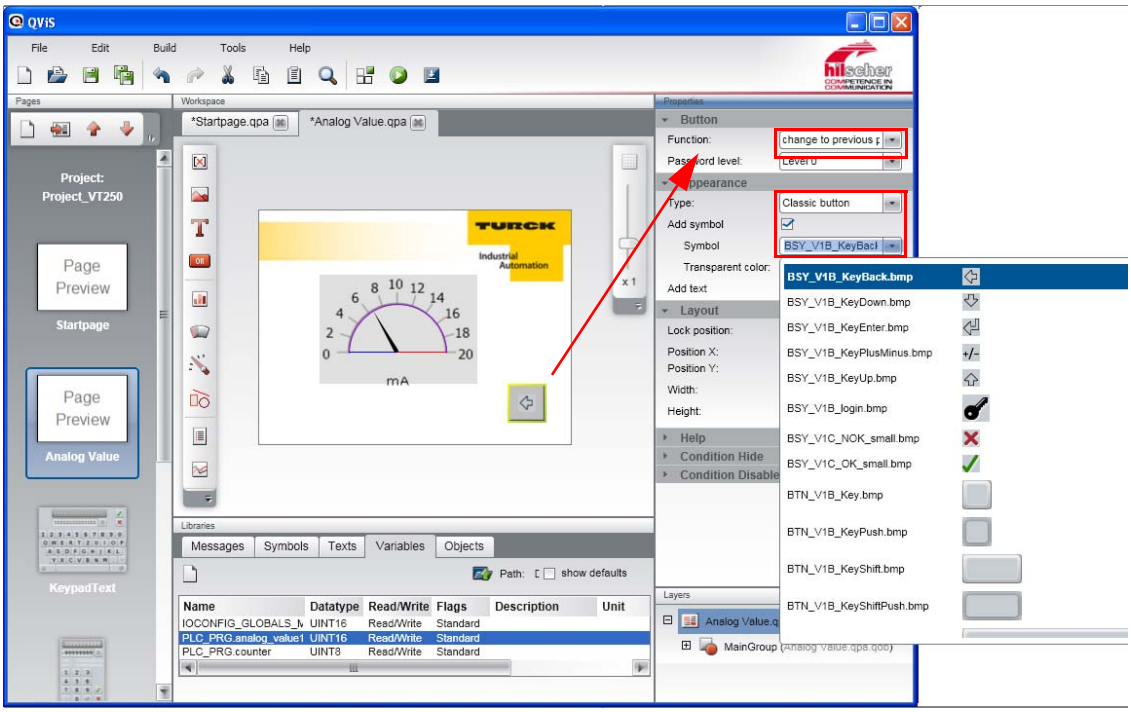
In order to go back from the second project page "Analog value" to the start page, an additional button is added to the example (please proceed as described under Buttons (page 2-36)).

6 The function "change to previous page“ is assigned to the button.

7 In addition to that, a symbol is defined for the button.

8 To do so, please activate the "Add symbol“-function and select the corresponding symbol under "Symbol".

Figure 41: Display of analog value



2.3.8 Compiling the project

Save the project and compile it via the "Build"-button.

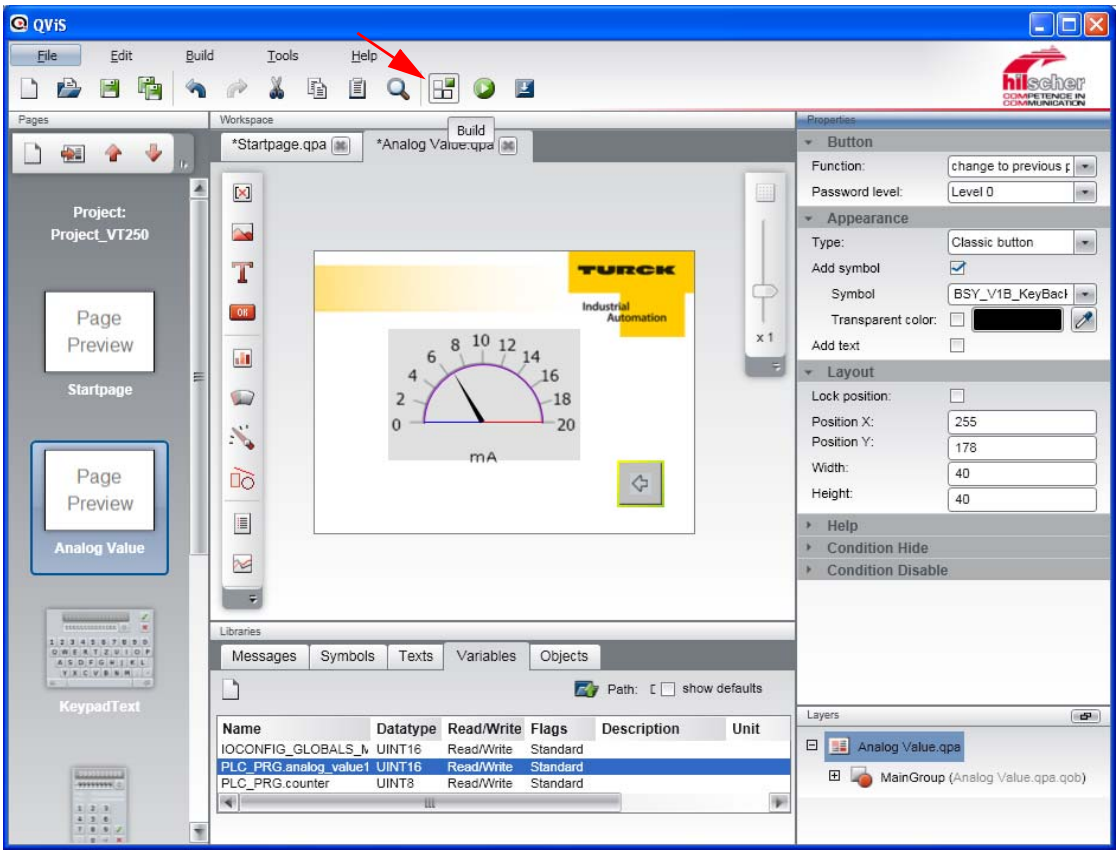
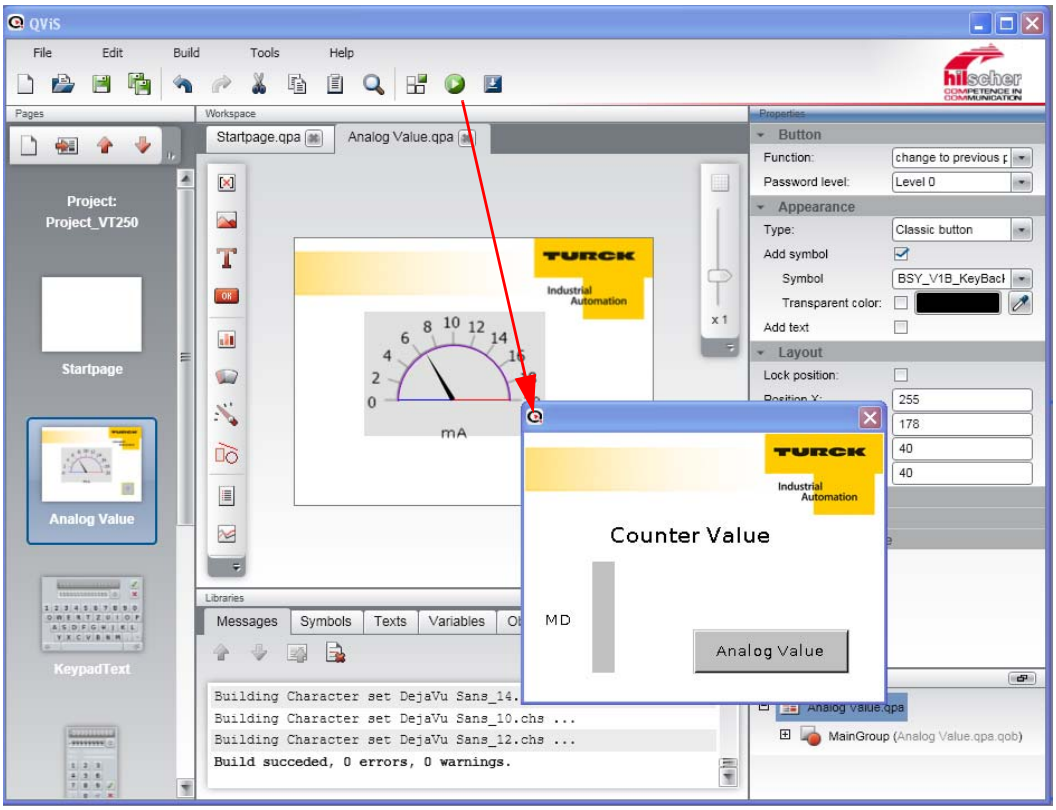


Figure 42: Compiling the project via "Build"

2.3.9 Simulation of the project

After the compilation, several project functions (change of page etc.) can be tested via the simulation.

Figure 43: Simulation



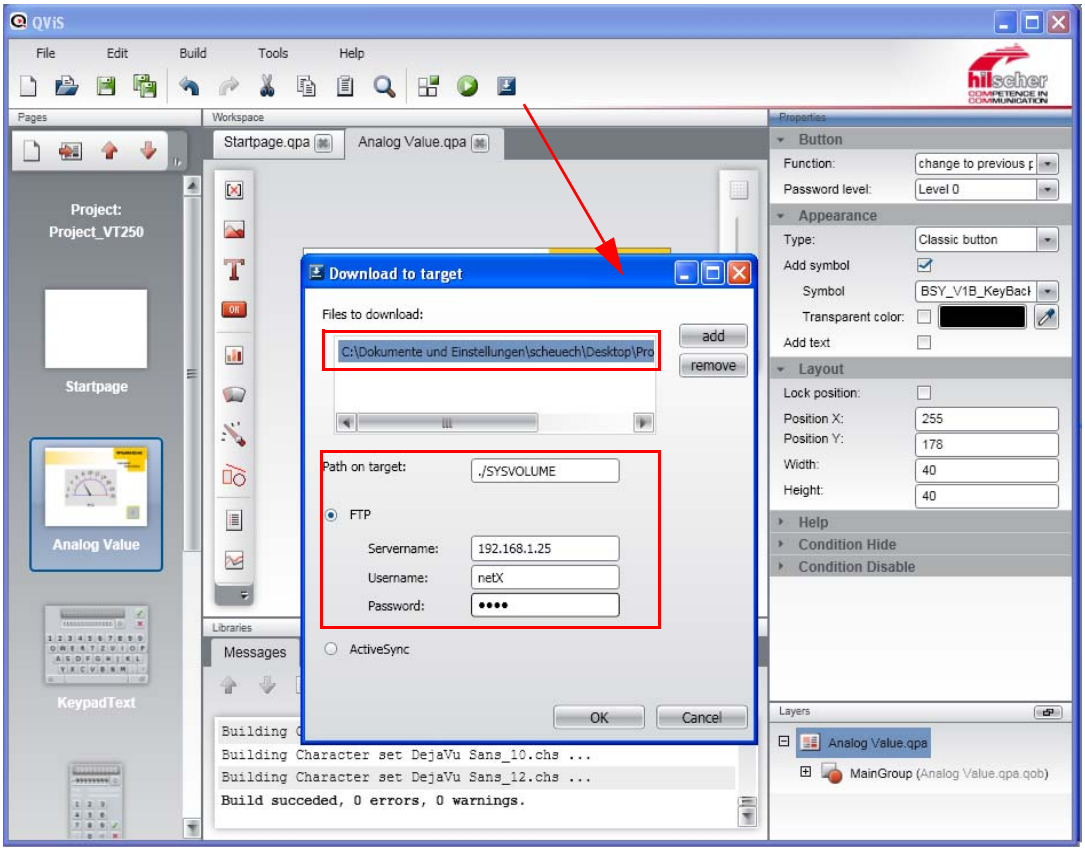
2.3.10 Downloading the project

1 The project is now loaded to the VT250-57x via the "Download"-button.

Note

Please make sure that the selected file is the project file.

Figure 44: Download



2 Mark the download via FTP and enter the following settings
   - Path on target: ./SYSVOLUME
   - Servername: IP-address VT250-57×
     (here: 192.168.1.25, default: 192.168.1.254)
   - Username: netX
   - Password: netX
3 Confirm the settings with OK.
4 The download is started.
5 Confirm the successful project-download with OK in order to execute a restart at the device.
6 The visualization is now running on the VT250-57×.

3 CoDeSys - TURCK-project templates

3.1 General

The CoDeSys version provided by TURCK (to be downloaded from www.turck.de) contains project templates for projects with VT250-HMIs.

They are automatically installed during CoDeSys installation. In addition to that, the installation also contains the necessary configuration files (*.gsd, *.eds, etc.) as well as a QViS example project for the visualization of the CoDeSys project.

Note

Please read chapter 2 of this manual or the manufacturer-documentation provided with the software-tools CoDeSys and QViS (manuals, online help) in order to learn more about the program handling.

3.1.1 CoDeSys project templates

These project templates contain, depending on the HMI used:

- a preconfigured application (see also Renaming the application (page 2-9)) incl.
  - Globale Variablenliste (siehe auch Global variable list (page 2-22)),
  - an example program (see also Programming (example program) (page 2-20)):
  - a symbol configuration (see also Export of variables (page 2-21)) etc.
- an example configuration with the correct fieldbus master and possible IOs (see also Adding a DeviceNet™-master (page 2-12) and Adding DeviceNet™-slaves (page 2-14)).

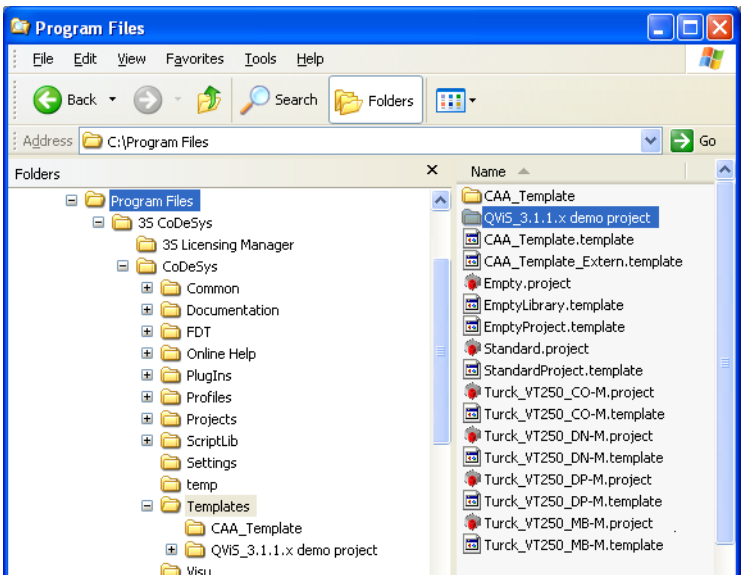
The project templates can be chosen when creating a new project in CoDeSys (see also Selecting the project templates (page 3-3)).

3.1.2 Example visualization in QViS

The QViS example project is also installed during the installation of the TURCK CoDeSys-version.

It can be found in the software's program folder
„..\3S CoDeSys\CoDeSys\Templates\`viS_3.1.1.x demo project“.

Figure 1: Directory of he demo project

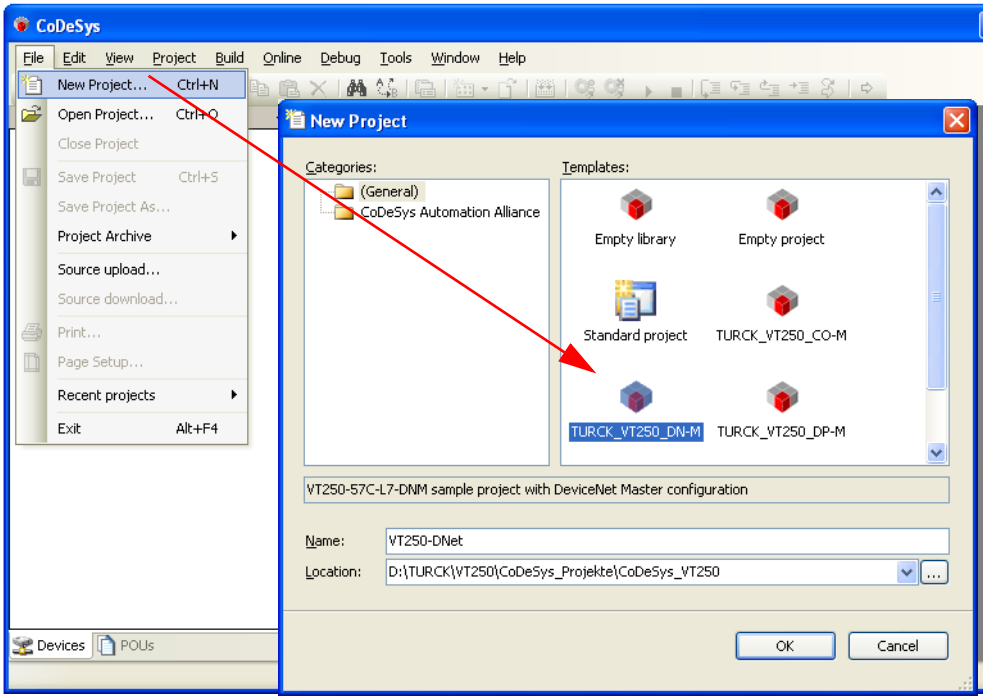


3.2 The CoDeSys example project

3.2.1 Selecting the project templates

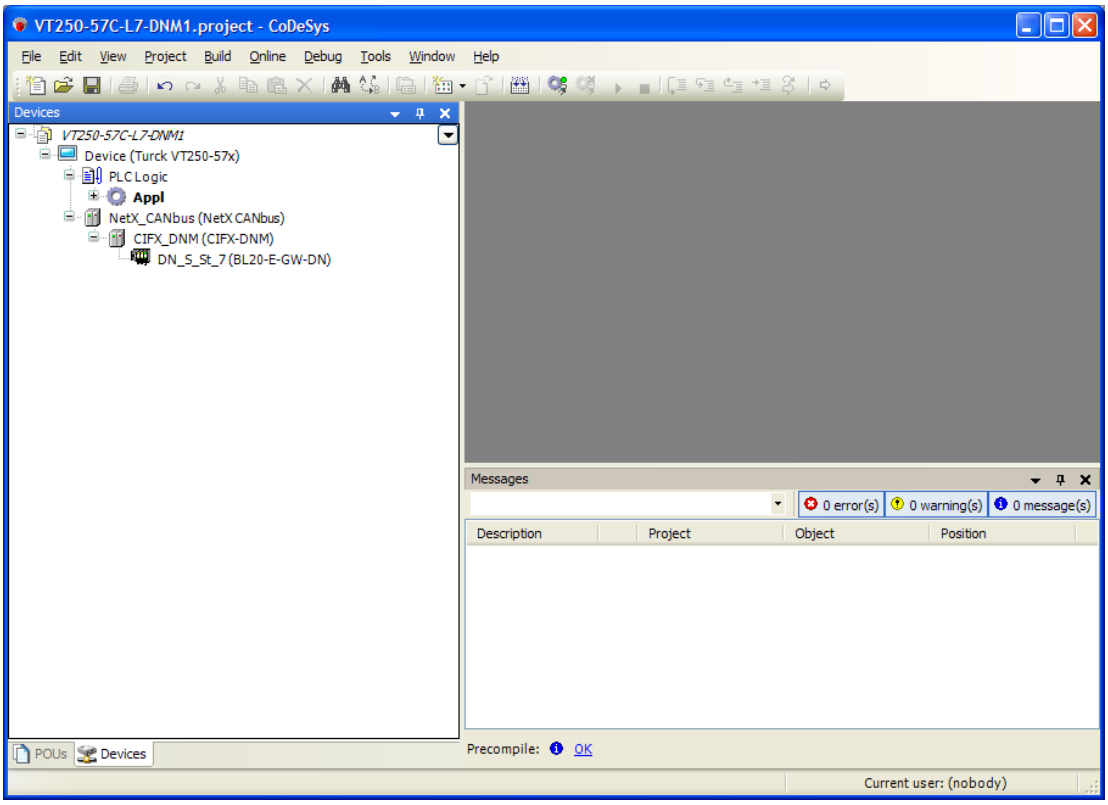
1. Open a new project under "File → New project..." and select the project template depending on the fieldbus used.
2. In this example, an HMI with DeviceNet™-master is used.
3. Rename the example project if necessary.

Figure 2: Creating a new project with template



3.2.2 Project

Figure 3: The example project

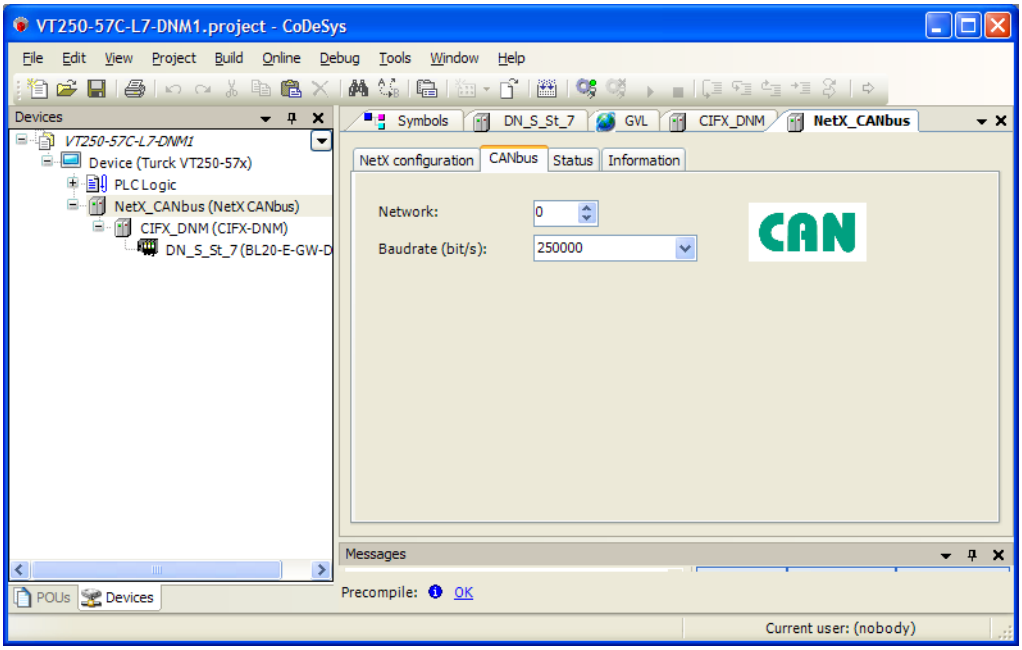


3.2.3 Fieldbus master

The example project contains the master to be used depending on the fieldbus.

For the DeviceNet™-communication, first of all the "NetX_CANbus" is added to the DeviceNet™-project.

Figure 4: Example project, NetX_CANbus



After this, the necessary DeviceNet™-master is added.

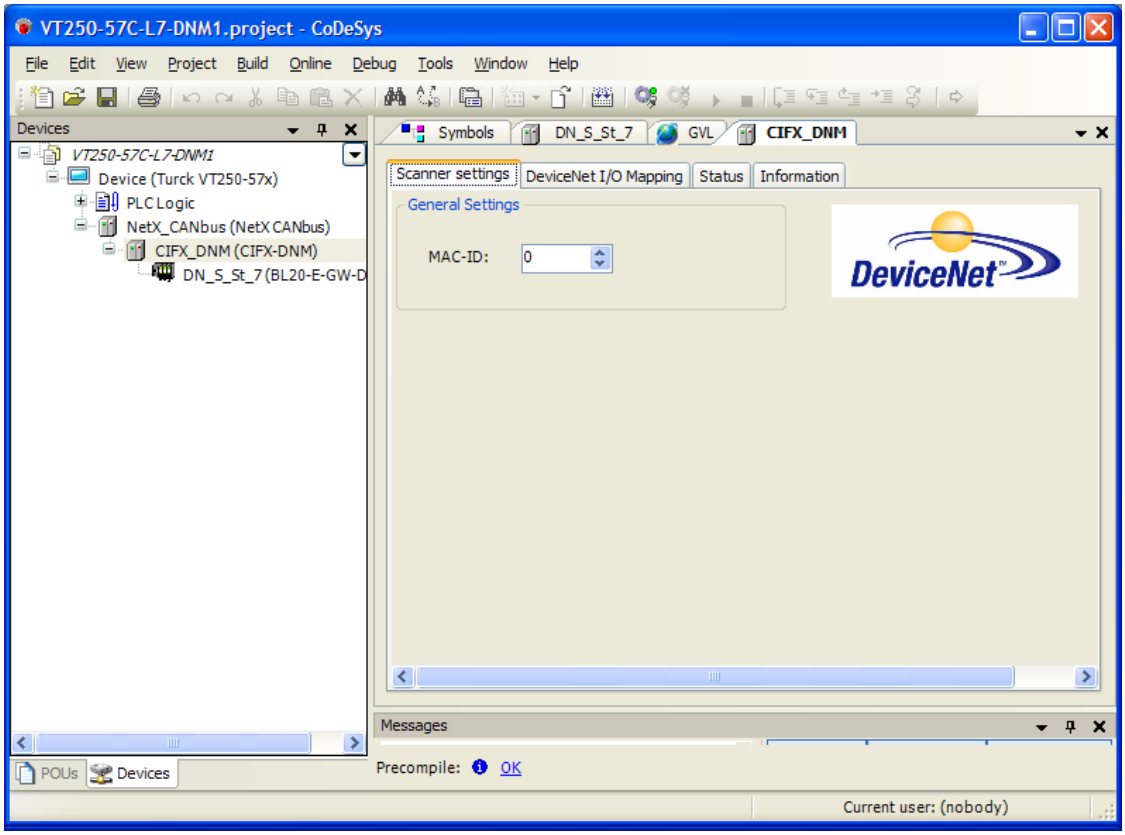


Figure 5: Example project, DeviceNet™-master

3.2.4 Fieldbus slave

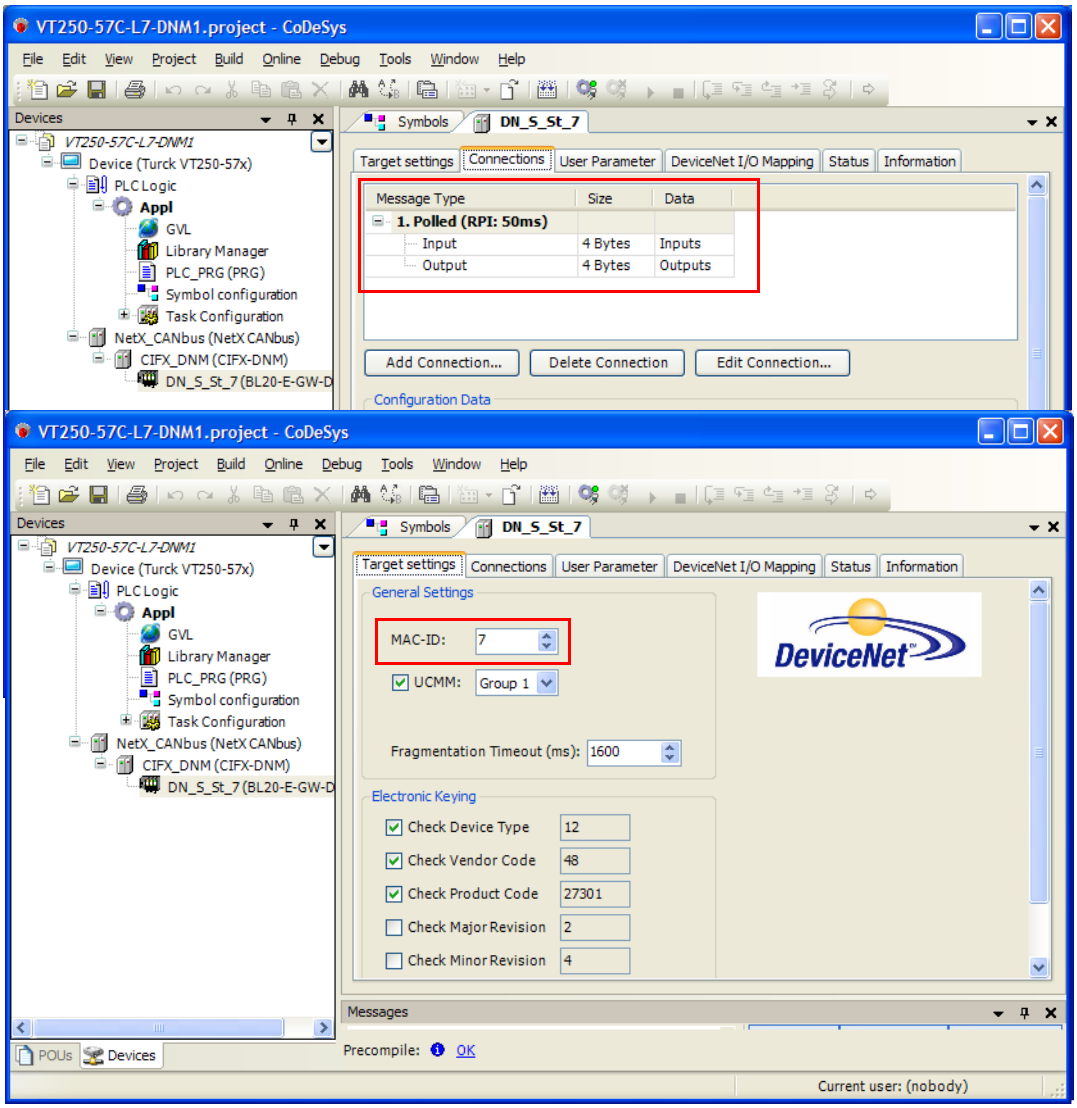
The example projects contains a BL20-ECO gateway for DeviceNet™ (BL20-E-GW-DN) as slave. The process image of the complete physically present station contains 4 byte input and 4 byte output data (incl. status and control word of the gateway).

The physically present station of the example projects is built-up as follows:

- BL20-E-GW-DN
  + BL20-E-8DI-24VDC-P (8-channel digital input module)
  + BL20-E-8DO-24VDC-0.5A-P (8-channel digital output module)

In the example, DeviceNet™-MAC-ID 7 is assigned to the gateway.

Figure 6: Example project, slaves



3.2.5 Variable declaration in the example project

Global variable list (GVL)

The GVL of the example project contains the following variables:

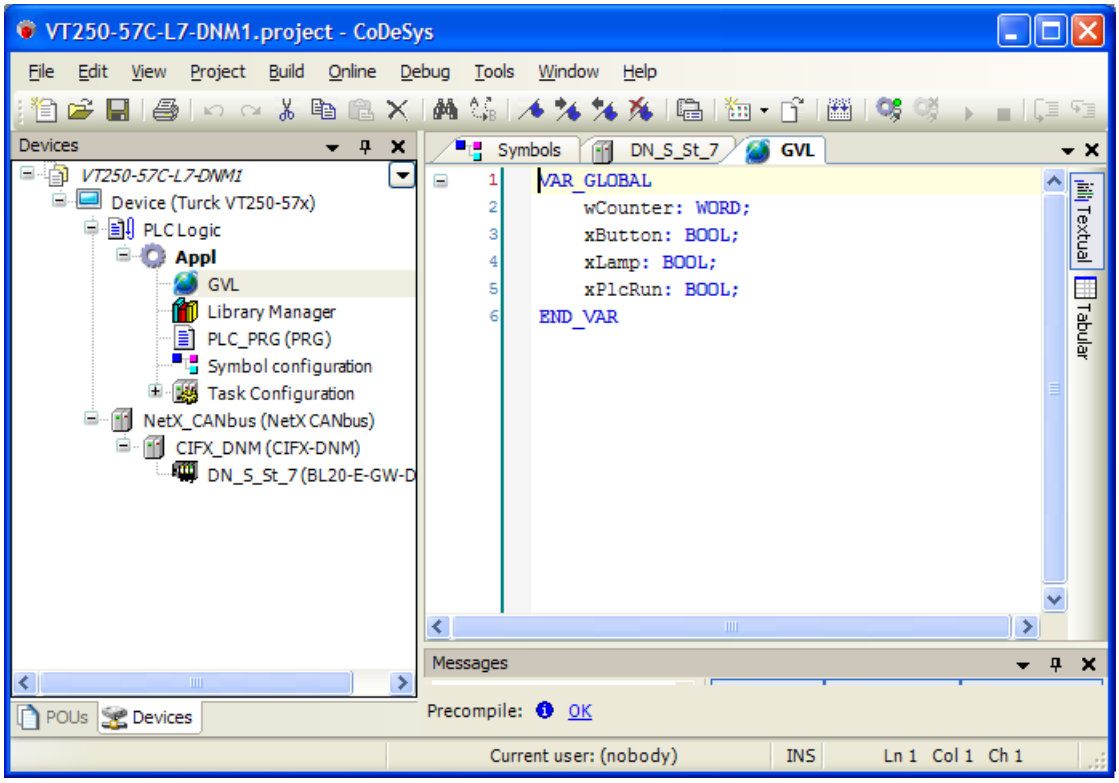


Figure 7: Example project, GVL

Variable-declaration in the I/O Mapping

In addition to the variable declaration in the GVL, the example also contains variables assigned in the station's data image.

Input bits (see Fieldbus slave (page 3-6)):

bit %IX3.0 and %IX3.1 = "xDN_DI_1“ and "xDN_DI_2“

Output bits (seeFieldbus slave (page 3-6)):

bits %QX3.0 and %QX3.1 = "xDN_DO_1“ and "xDN_DO_2“

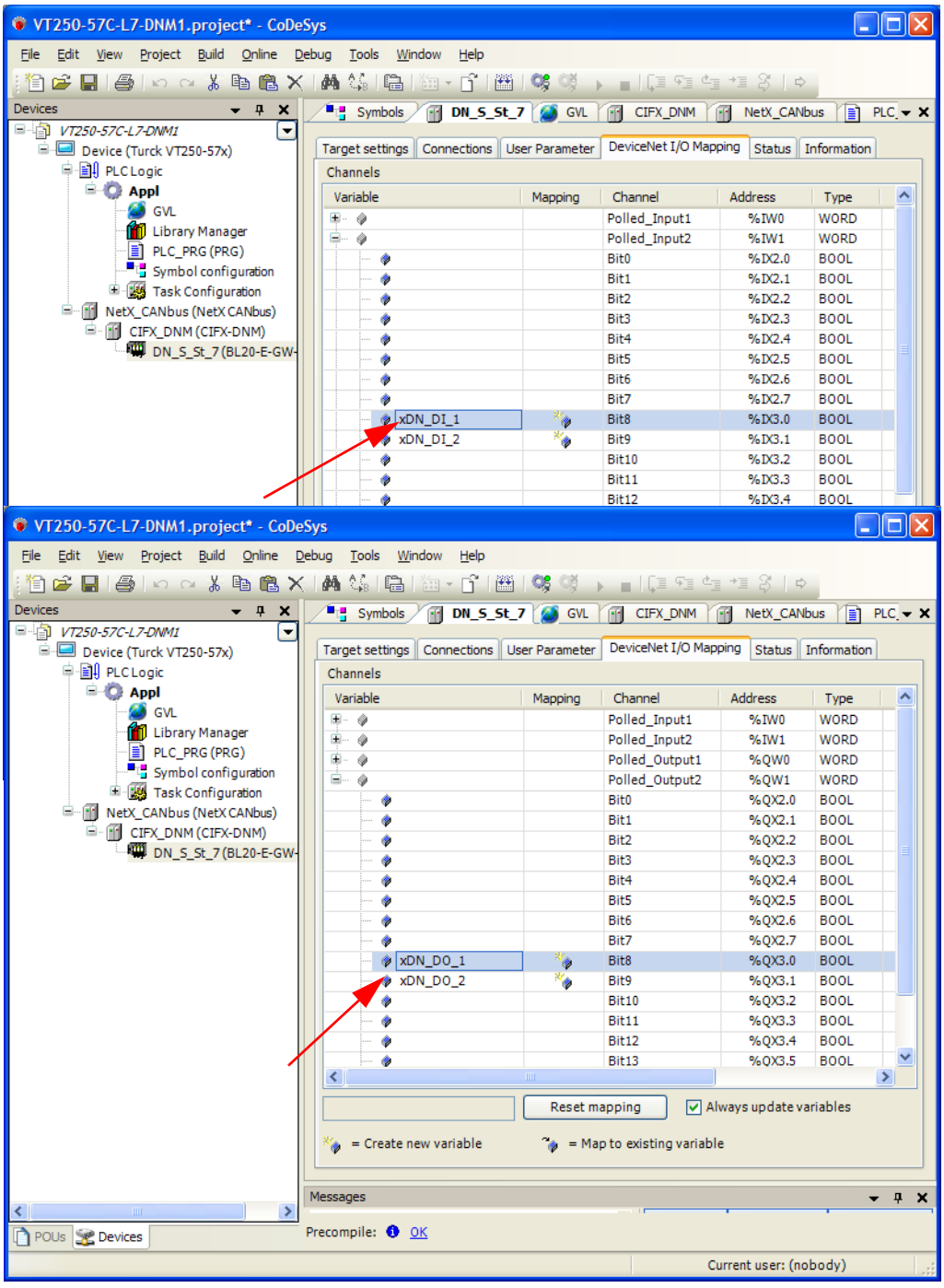


Figure 8: Example project variable-declaration in "I/O Mapping"

3.2.6 Example program (PLC_PRG)

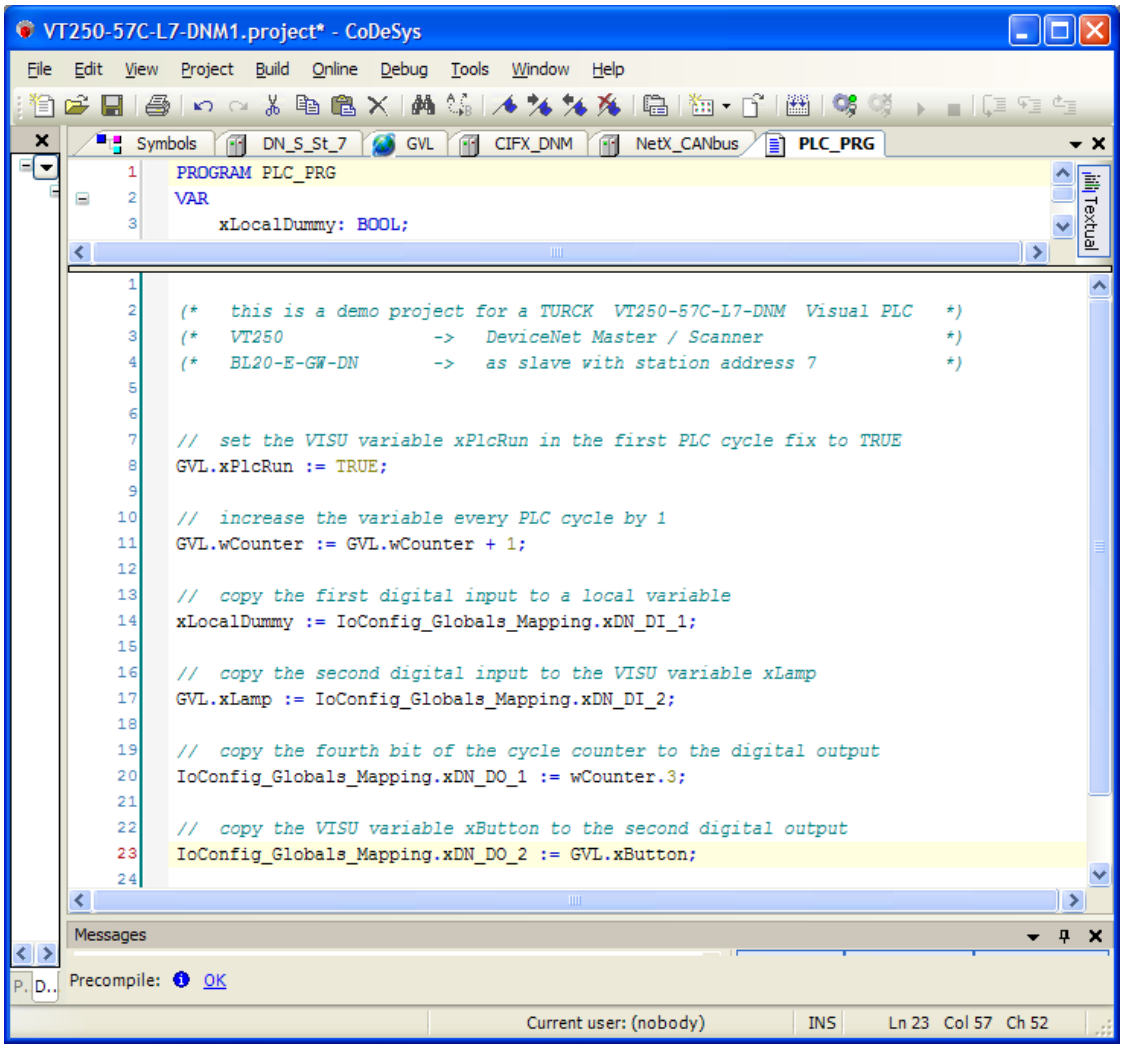


Figure 9: Example program (PLC_PRG)

3.3 The QViS example project

1 Open the example project using "File → Open project...".

Note

Please start QViS first and then open a project. QViS projects cannot be opened via double-click on the project file.

2 The project file QViS_3.1.1.x.qpr“ can be found in the following directory after the installation of the TURCK CoDeSys-version (see also General (page 3-2)).

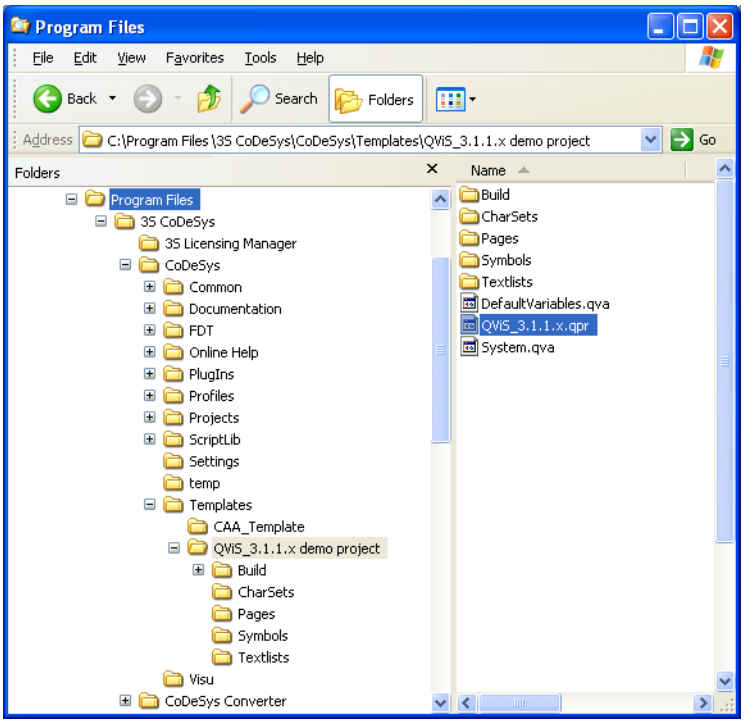
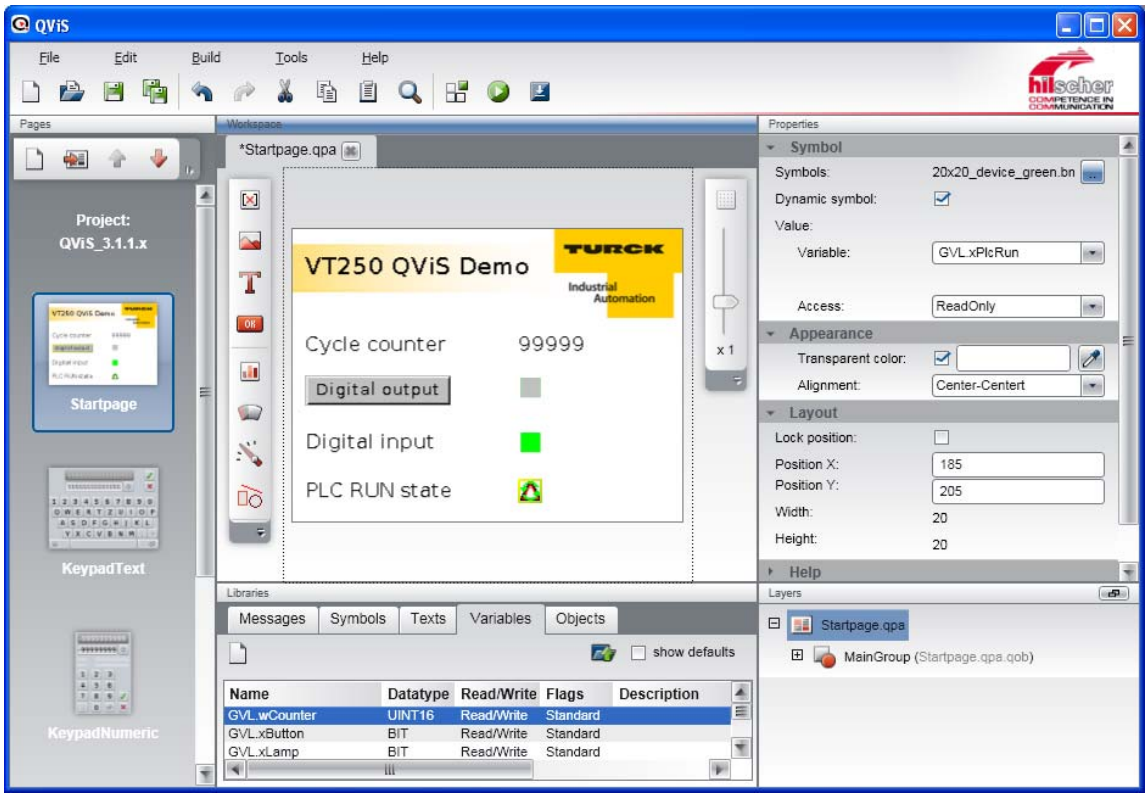


Figure 10: QViS example project, location

Figure 11: QViS example project

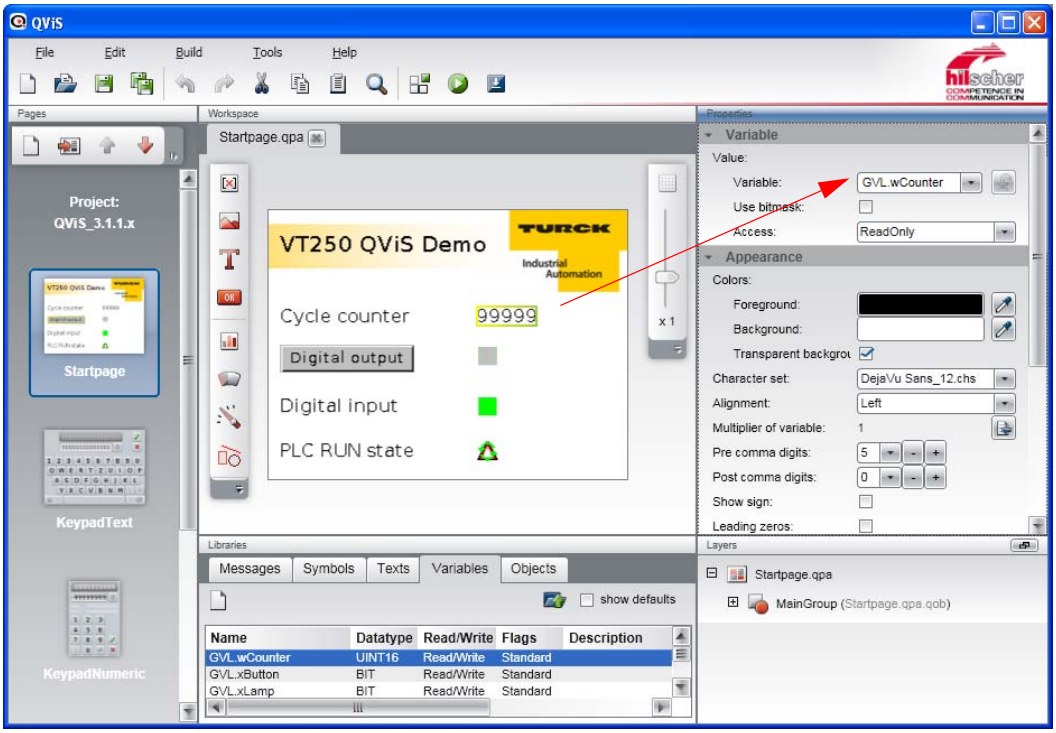


3.3.1 Variable declaration

Please read chapter 2, Visualization with QViS, Variable import (page 2-29) to learn more about the import of program variables to QViS.

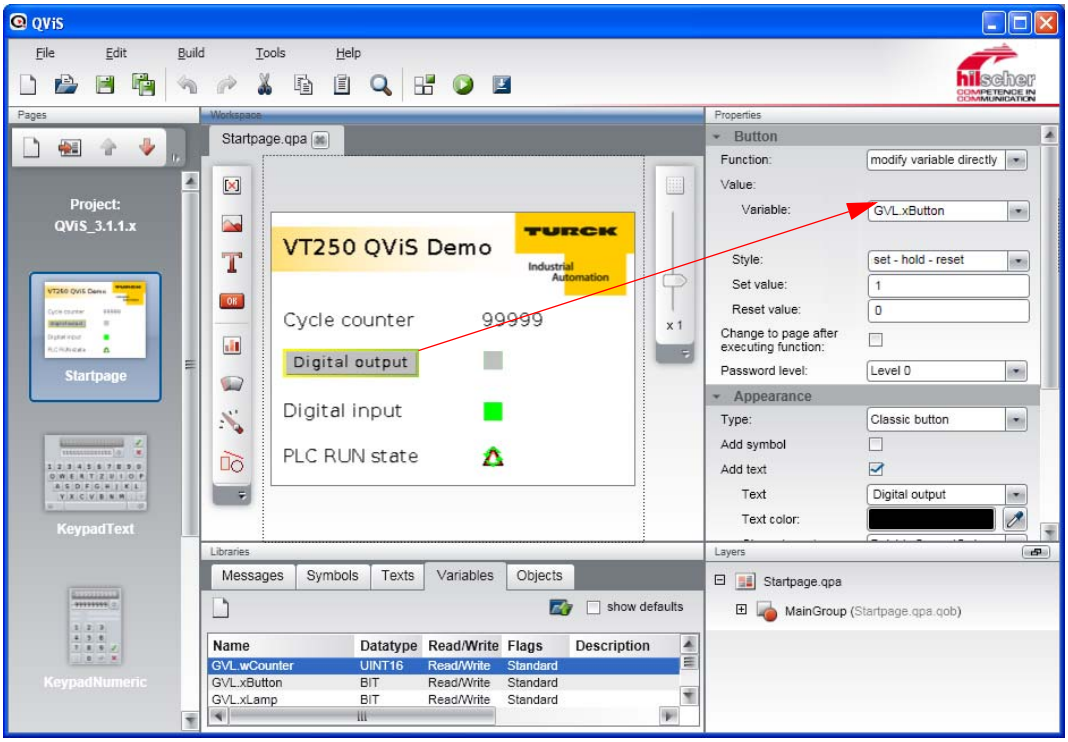
1 The variable "Cycle counter“ is linked to the global CoDeSys-variable GVL.wCounter“.

Figure 12: QViS- "Cycle counter“



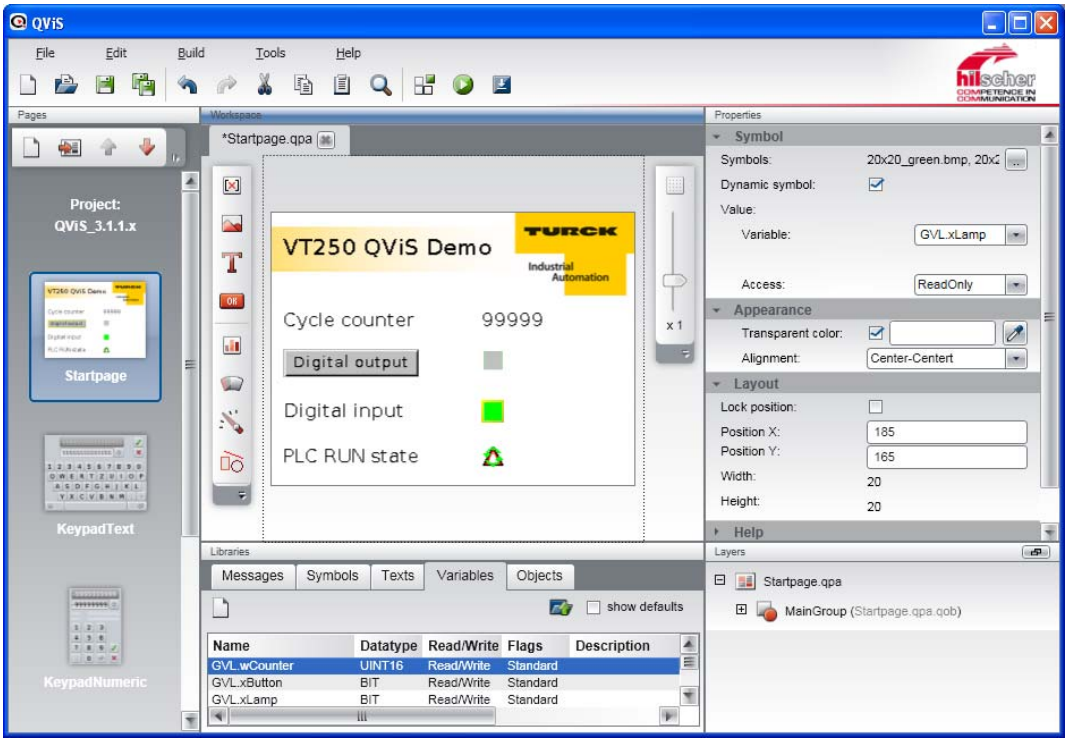
2 The button "Digital output“ is linked to the CoDeSys-variable "GVL.x.Button“. Pressing the button sets the second output of the digital output module (Generic (BL20-8DO)) in the BL20 station.

Figure 13: QViS-button "Digital output“



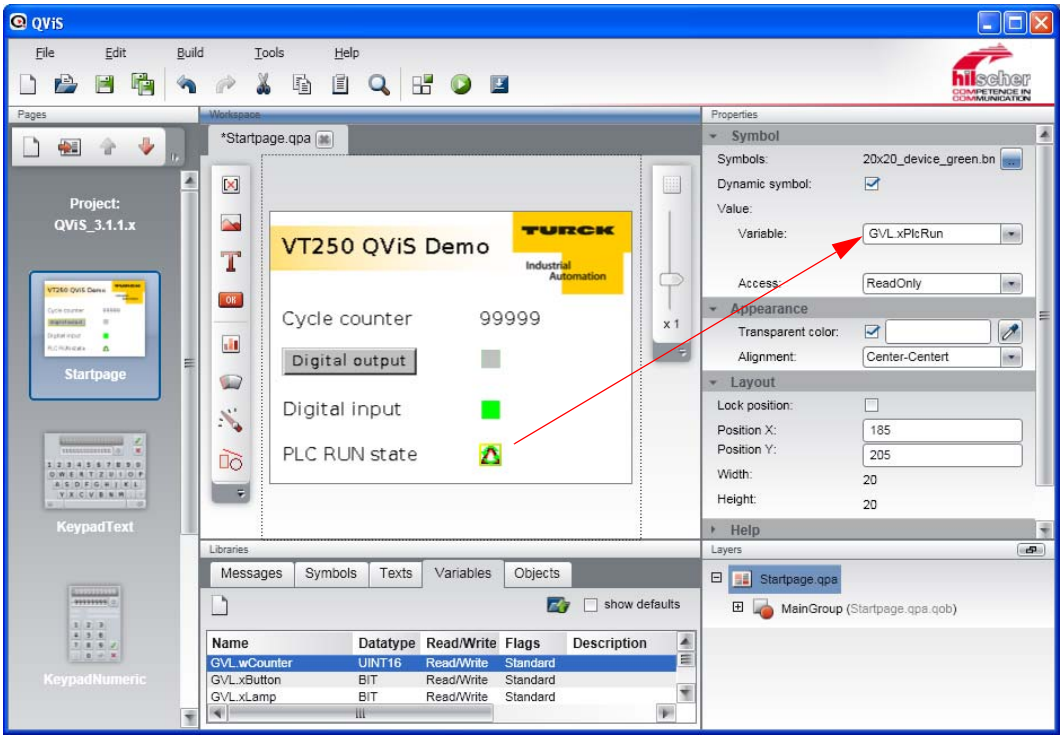
3 The indicator for the "Digital input" is linked to the CoDeSys-variable "GVL-xLamp" and shows an incoming signal at the digital input.

Figure 14: QViS-“Digital input“



4 "PLC RUN state“ is linked to the CoDeSys-variable "GVL.xPlcRun" and displays if the PLC-program is in RUN state or not.

Figure 15: QViS-PLC RUN



4 Index

Index

TURCK

Industrial
Automation

www.turck.com

D301248 1011

Hans Turck GmbH & Co. KG
45472 Mülheim an der Ruhr
Germany
Witzlebenstraße 7
Tel. +49 (0) 208 4952-0
Fax +49 (0) 208 4952-264
E-Mail more@turck.com
Internet www.turck.com