Tech-note

Control with CODESYS V3

This manual contains detailed information on how to create a control system based on the UniOP platform running the JMobile HMI software. The control is based on the CODESYS V3 software PLC.

Exor International S.p.A.
ptn0405
Ver. 1.03

Contents

Internal PLC (CODESYS V3)

This manual describes the UniOP control system based on the CODESYS V3 PLC software.
The documentation covers:

- CODESYS V3 running on eTOP500 and eTOP600 Series products
- Use of integrated I/O optional modules
- Support of Modbus/TCP and RTU communication
- Use of remote CANopen optional modules

Note: *For specific information regarding the use of CODESYS V3 communication protocol in JMobile related to the communication with CODESYS V3 PLC runtime please refer to the specific CODESYS V3 communication protocol chapter of JMobile Suite help or to the technical note PTN0357 available for download on www.exorint.net .*

Note: *This manual is not intended as CODESYS V3 programming manual, for specific documentation regarding CODESYS V3 Development System please refer to the CODESYS web site www.codesys.com and to and to its on-line help.*

Document code ptn0405
Version 1.03

Control Solutions with UniOP and CODESYS V3

UniOP products can deliver effective HMI and control solutions based on JMobile and CODESYS V3. UniOP eTOP500 and eTOP600 Series HMI products have been designed to include the CODESYS V3 PLC runtime.
The PLC runtime is automatically transferred to the device by JMobile Studio as part of the JMobile Runtime and it is running with the support of the operating system of the device.
CODESYS V3 Development System is required to develop and debug PLC applications and transfer them to the UniOP HMI device.
The CODESYS V3 runtime requires a license activation to be used. CODESYS V3 license is activated by JMobile studio, the license code is univocally matched with the MAC-ID data of the operator panel.

Note: *License activation is required for the operation of CODESYS V3 runtime. CODESYS V3 runtime will communicate with the Development System only after license activation.*

System Configuration

The HMI and control solution based on JMobile and CODESYS V3 can be applied in different configurations.

Compact Stand-alone Controller

The HMI and control system can be used to build very compact standalone systems. Input/output is available using the integrated I/O optional modules.

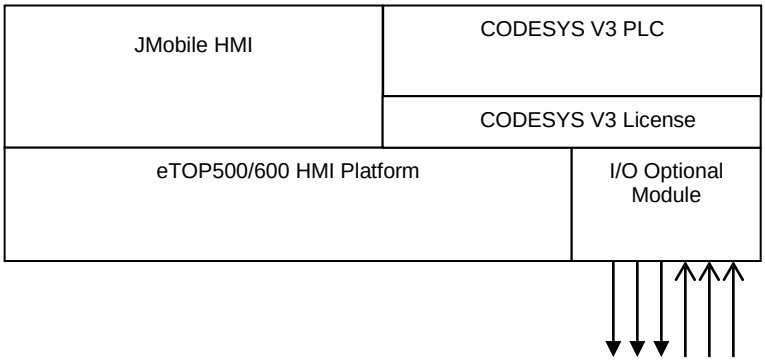


Figure 1

Controller with Remote I/O

A fieldbus interface (either built-in or with an optional module) is available for the HMI and control solutions. Configurations with local and distributed I/O are possible.

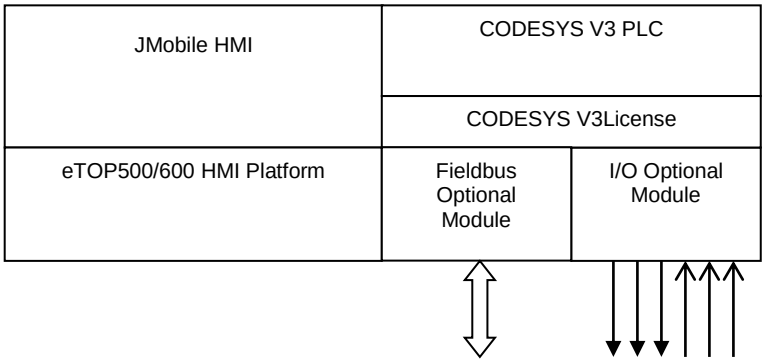


Figure 2

Connectivity in HMI and Control Systems

Even when adding the control option with the CODESYS V3 PLC, the HMI still retain its full communication capabilities based on JMobile communication drivers.

Note: *The communication capabilities over Serial network are limited by the amount of serial interfaces available, each serial interface supports a single communication protocol. Specific optional add-on serial interface modules are available to increment the amount of serial interfaces at disposal.*

Requirements and Limitations

The following firmware and software versions are required to work with the CODESYS V3 PLC Runtime:

	Version
JMobile Studio and Runtime	1.91 SP1 or higher
Operator panels BSP version	UN20: 2.73 or higher UN30: 1.76 or higher UN31: 1.76 or higher
CODESYS V3 Development System	3.5.4 or higher

Table 1

Note: *The operator panels internal PLC is compatible with CODESYS V3.5 Development System only. CODESYS V2.3 Development System is NOT compatible with CODESYS V3 runtime.*

Getting Started

This chapter provides the necessary informations on how to set-up the HMI + CODESYS Plc system. The required operations are listed below and explained in the forthcoming chapters.

1. CODESYS V3 Development System installation
2. Exor CODESYS Package installation
3. JMobile Runtime installation into the operator panel
4. Activation of the CODESYS V3 license into the operator panel
5. Creation of a new PLC project
6. Download plc application
7. Symbol File configuration with CODESYS V3 Development System
8. Communication Setup in JMobile Studio

CODESYS V3 Development System installation

The CODESYS V3 Development System can be downloaded for free from the CODESYS web site at www.codesys.com/download.html
You need to register before you can download the software.

Exor CODESYS Packages installation

A dedicated CODESYS Package is required to allow the standard CODESYS V3 programming software to program control systems based on eTOP500 and eTOP600 Series products.
The EXOR Package is included into JMobile Suite since version 1.91 SP1 under “\CODESYS\V3\” folder and is also available for download from the web site www.exorint.net.
The CODESYS Development System includes a tool called “Package Manager” for the installation of the CODESYS Package. The Package Manager tool can be launched from CODESYS Tools menu, selecting the proper menu item. This tool can be used both for checking the installed Packages and for installing new ones.
To install the Exor CODESYS Package, once opened the Package Manager click on the “Install” button and browse for the file with .package extension, confirming with “Open”.
The Installation procedure will start automatically, the system asks if to perform a Complete or Typical setup, any of these will install all the required files for a complete support of the CODESYS V3 PLC runtime features.
The Package Manager dialog is visible in Figure 3, showing the installed Packages.

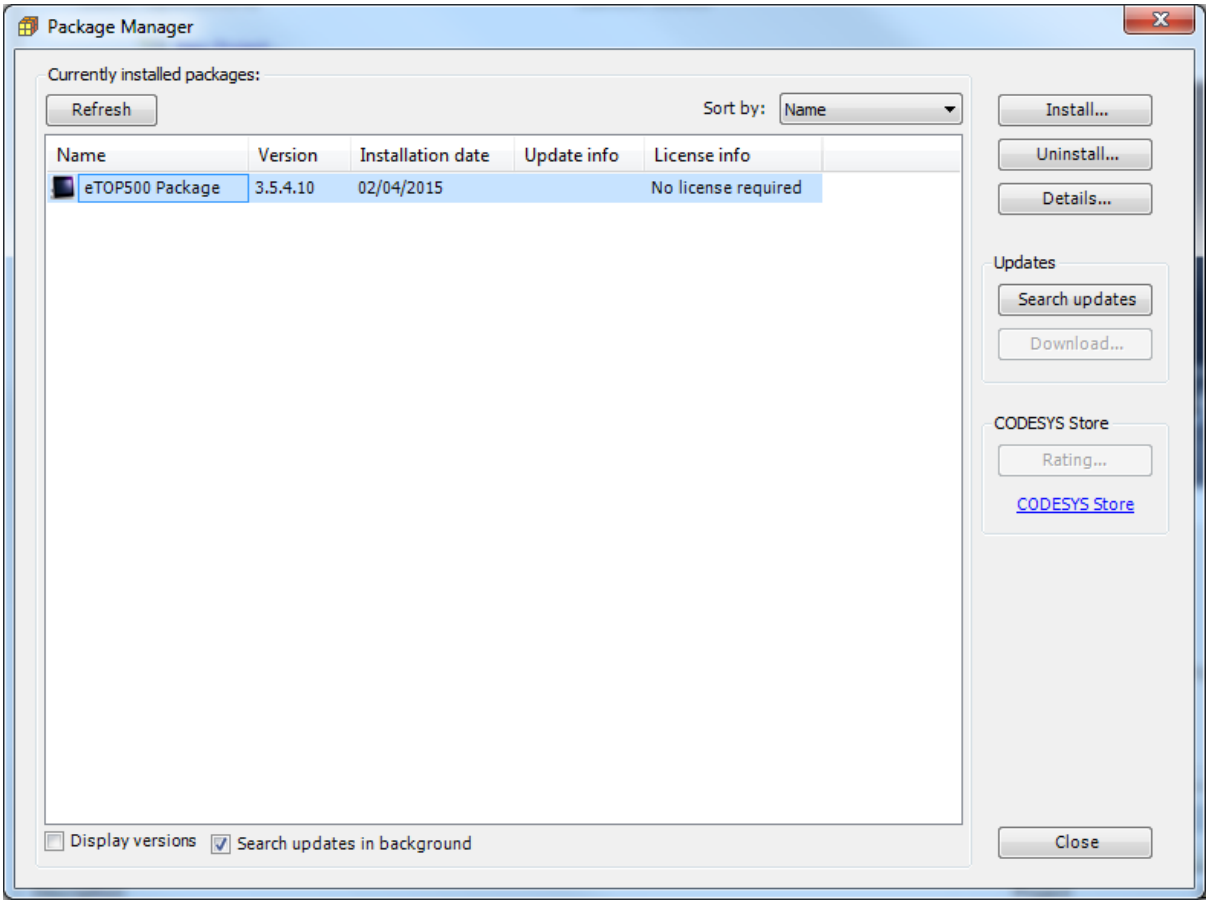


Figure 3

Updating Old CODESYS Packages

When a new version of the CODESYS Package is available, it can be installed to update the Package information accordingly to the latest CODESYS V3 PLC Runtime features.
When a new version of Package is released EXOR always tries to guarantee the complete compatibility with the previous versions.
If the new Package is not fully compatible with the previous version some changes to adapt the PLC application to the new Package may be required. In these cases, the updates to the EXOR Package, will come with proper instructions for the conversion of existing projects.
The Package update procedure is identical to the first installation of the CODESYS Package, please refer to the proper chapter for specific information.

Note: *CODESYS V3 keeps the older Package versions instead of replacing them in order to have all the installed Packages at disposal if required.*

JMobile Runtime installation

The HMI and control system is composed by two main subsystems, the JMobile HMI runtime and the CODESYS V3 PLC runtime.
As the CODESYS PLC runtime is part of the JMobile HMI runtime it is necessary to install the JMobile Runtime on the operator panel to have the PLC runtime running. For further information on installing the runtime, please refer to JMobile Studio help.

Activation of CODESYS V3 license on the operator panel

CODESYS V3 plc runtime license is activated on the operator panel trough an Ethernet connection using JMobile Studio, each license code can be used on a single operator panel, once activated the license is univocally matched with the MAC-ID of the operator panel.
To activate the license follow the below steps:

- In JMobile Studio select Run > Manage Target
- Into the License tab select, from the drop-down menu in the Panel Info section, the IP address of the operator Panel where the license will be activated from the list of operator panels available into the network.
- Type in the Activation keys section the licence code.
- Click on the Activate Panel button.
- Once the system confirms the license to be successfully activated reboot the operator panel to complete the activation procedure.

Note: *The CODESYS V3 license activation procedure requires an active internet connection on the PC.*

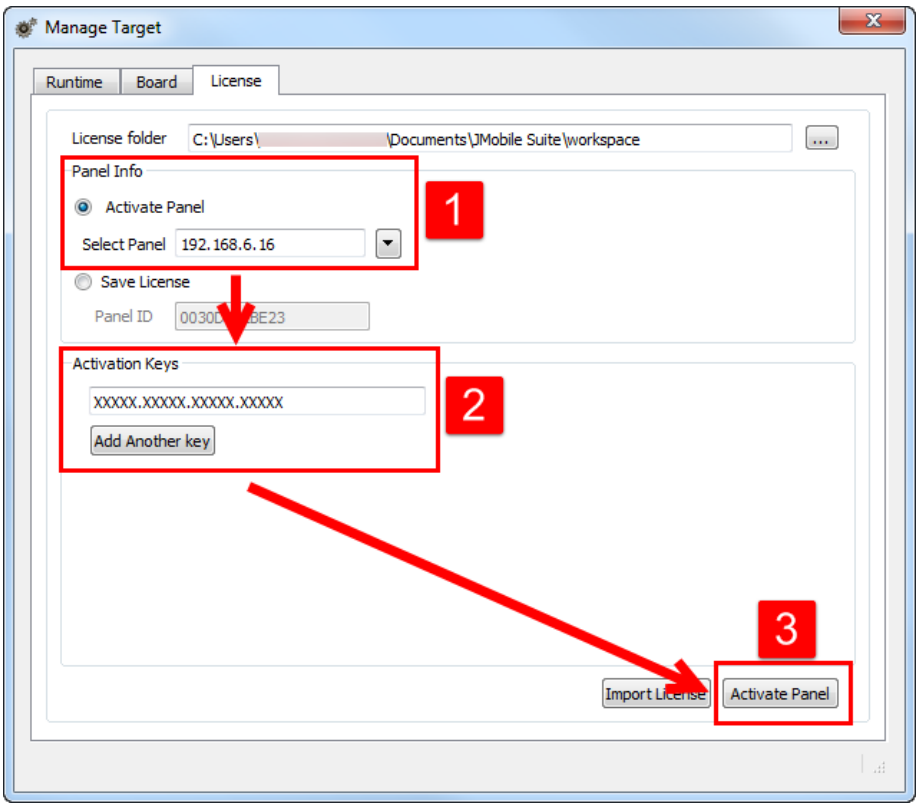


Figure 4

For testing purposes it is possible to use a temporary demo license that will activate the CODESYS V3 plc runtime for 120 minutes. To activate the demo mode enter the license code CODESYS_DEMO.

Note: *CODESYS V3 licenses, once activated cannot be paired with a different device or de-activated.*

In case it is necessary to make sure whether a license has been activated on an HMI it is possible to verify the system log of the operator panel.
From the context menu on the operator panel select the option “Log at boot”, and then reboot the operator panel. At panel restart the Log window will be displayed on screen, if a valid CODESYS V3 license is found from the system the string “CODESYS Module: CODESYS V3 license found: CODESYS V3 is running” will be present among the panel boot logging information.

Creation of a new PLC project

To create a new CODESYS V3 project select File > New Project or click on the icon from the upper tools bar. New Project dialog will be displayed, here, among the available templates, select the “eTOP500 project” template, define then Project Name and Location, then confirm with OK as shown in Figure 5.

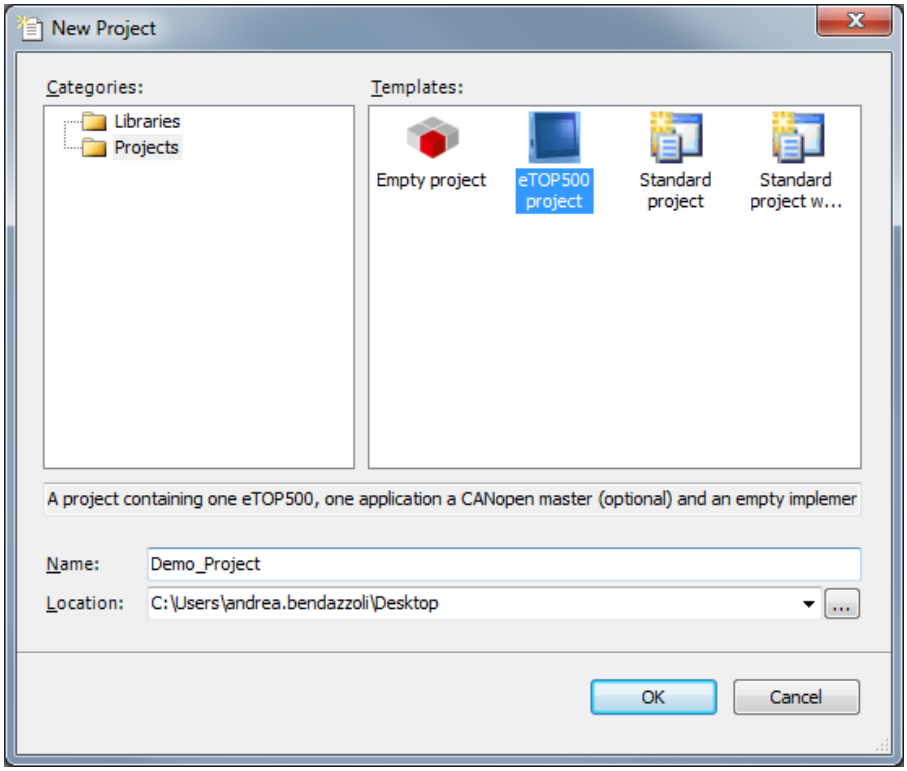


Figure 5

Download PLC application

The selection of the PLC where to download the project must be done from Device communication settings tab before proceeding with the download operation.
Double click on “Device (eTOP500)” in the project tree to display the Device properties in the work area, select the “Communication Settings” tab then click on the “Scan Network…” button.
The Select Device dialog will be displayed, this dialog lists all the compatible devices available in the network, the operator panels are defined as “eTOP500”, select here your device then press on “OK”.

In case more operator panels using CODESYS V3 PLC runtime are present into the network each panel is recognized by a different string between square brackets reported after Device name.
In Figure 6 the string is “0001.B000.29C2”, the last part of the string “29C2” corresponds to the last 2 bytes of the operator panel IP Address in Hex format so, in this case, the corresponding operator panel is the one with IP address xxx.xxx.41.194 as 29Hex corresponds to 41Dec and C2 Hex corresponds to 194 Dec.

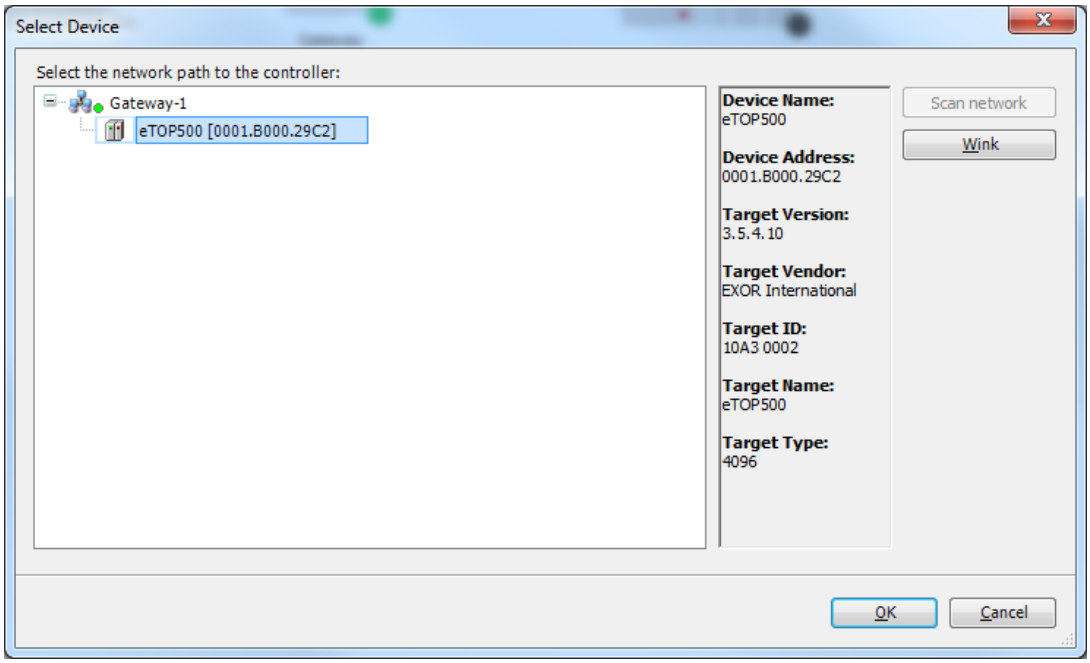


Figure 6

The selected device is then listed in the Communication Settings as shown in Figure 7, the device properties are listed on screen. A green dot over the device graphical representation informs that the device is correctly recognized and available into the network.

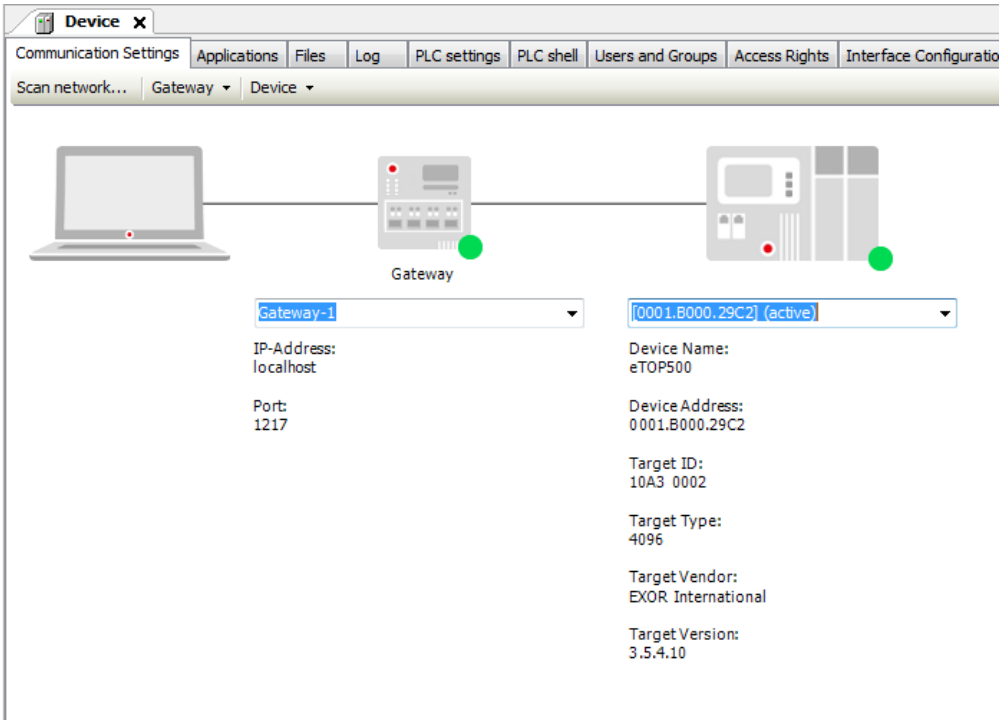


Figure 7

Note: *Communication with the available devices is established through a Gateway, a default Gateway is available, and it is generally not needed to change the standard Gateway settings. For more information about the Gateway set-up please refer to CODESYS V3 documentation.*

Upload PLC Application

The upload of the PLC project is possible only if the project Source has been previously downloaded into the PLC. To download the project source, while Online, select the Online > Source download to connected device command.

To upload a PLC project from the HMI, select the Source upload command from the File menu and select the PLC from the Device list as shown in Figure 8.

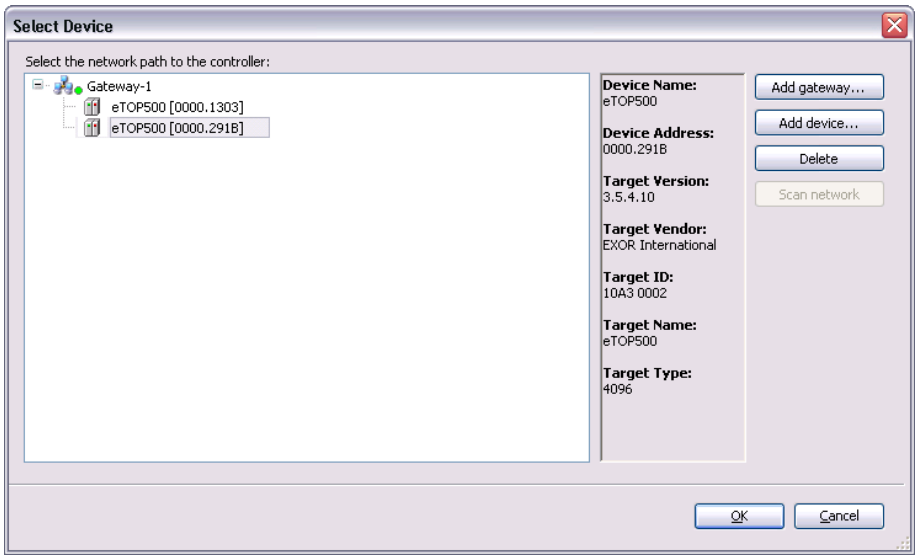


Figure 8

Symbol File configuration with CODESYS V3 Development System

The CODESYS V3 Development System can create a list of all the variables available into the PLC program in form of a file with ".xml" extension.

The variables to be included into the generated xml file must be selected, to get access to the project variables list for the selection is necessary first of all to add the Symbol configuration to the CODESYS project as this project item is not available by default. To add the Symbol configuration right click on the Application voice from the project tree, then into the context menu select Add Object > Symbol configuration. The symbol configuration voice will be added to the project tree as shown in Figure 9.

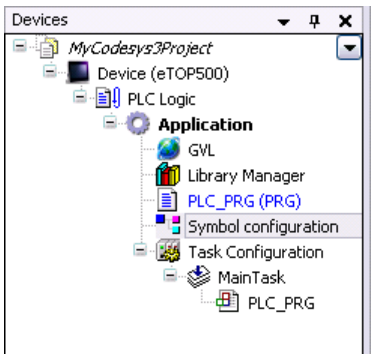


Figure 9

By double clicking on the new project tree item the Symbol configuration will be displayed in page as shown in Figure 10. The Symbol configuration contains a list of all the variables available into the

CODESYS project, single variables or groups of variables can be selected by checking the corresponding voice in the list.

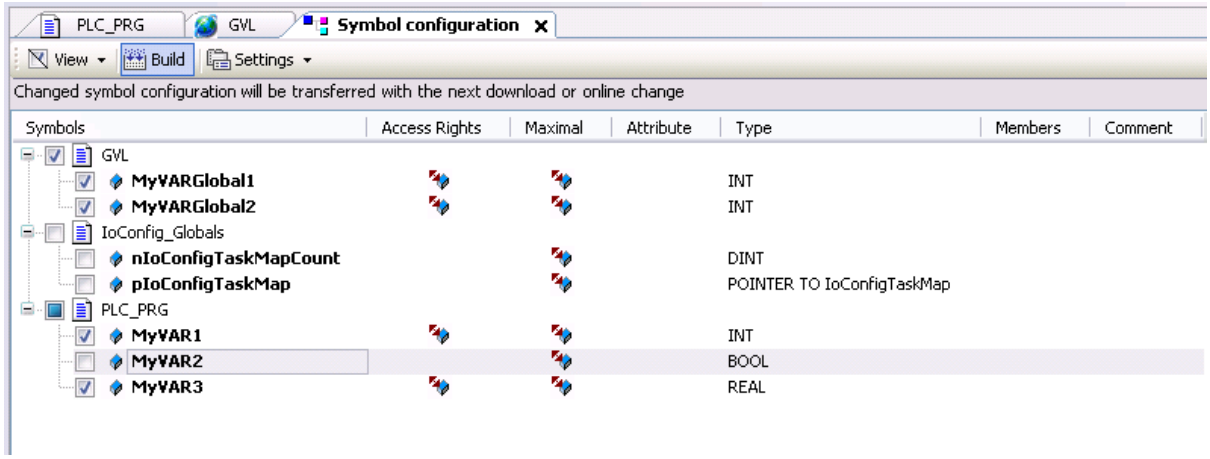


Figure 10

The xml symbol file is generated when the application is Downloaded into the PLC or when the Build > Generate Code command is executed, the file created is stored in the application folder.

Communication Setup in JMobile Studio

JMobile communicates with the internal CODESYS V3 runtime using the CODESYS V3 ETH protocol. Localhost 127.0.0.1 should be entered in the IP Address parameter, this identifies the PLC as internal CODESYS V2 runtime.

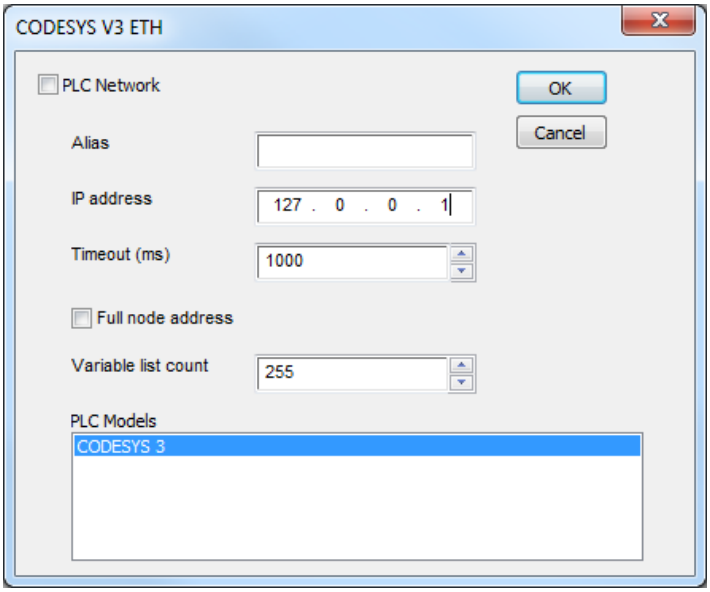


Figure 11

Additional informations regarding the CODESYS V3 ETH driver are available in JMobile help Communication drivers section.

Import Symbol Files in JMobile Studio

JMobile Tag Editor requires direct import of CODESYS V3 symbol file for the definition of the Tags. Use the command “Import Tags” in JMobile Tag Editor to import the symbols generated by the Development System.
Select *.xml as Import Type in the JMobile Tag Import dialog, as shown in Figure 12.

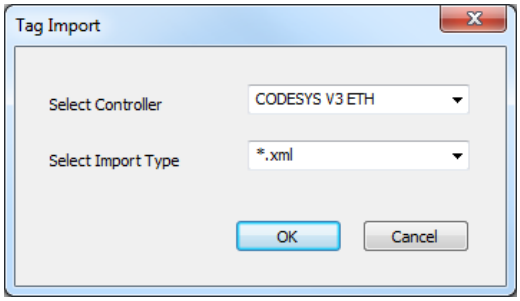


Figure 12

CODESYS V3 Development System generates a new version of the “.xml” file each time the PLC project is built.
CODESYS V3 Ethernet communication driver supports automatic symbol file (SDB) upload from the PLC; the HMI can upload the symbol table from the PLC using the communication protocol.
Any change in the tag offset information due to a new compilation of the PLC program does not require importing again the symbol file.

Symbol file must be imported again when:

- Tags have been renamed
- Tags have changed data format
- New tags have been added.

Communication Diagnostic

JMobile reports communication diagnostic information also for the communication with the internal PLC runtime. To display communication diagnostic information use the proper System Variables or the System Logger. For further informations please refer to JMobile Studio help.

Installing Optional Modules

Optional modules of type PLCM (Communication Modules) and PLIO (I/O modules) can be installed in all eTOP500 and eTOP600 Series HMI products.
The steps to follow for the installation of the optional module are shown in Figure 13 and are the following:

1. If present, remove the sticker covering the plug-in slot
2. Insert the guides of the module into the holes in the enclosure
3. Press the module down into the connector
4. Lock the module using the screw

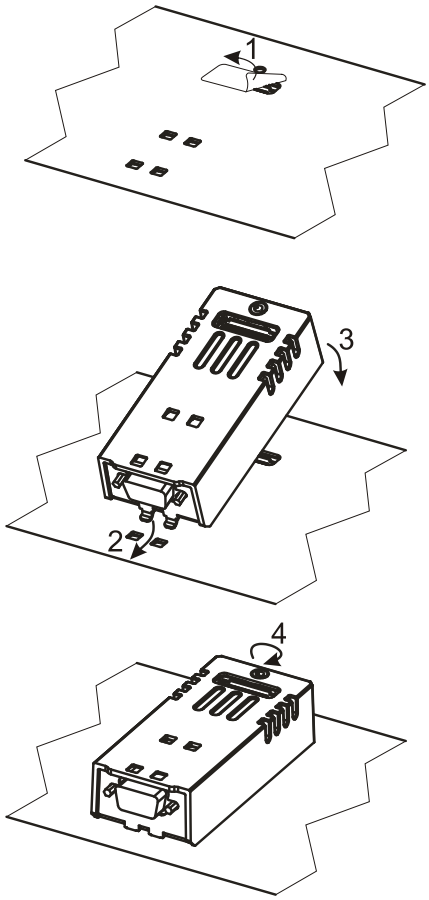


Figure 13

Note: *Do not plug or remove optional modules when the HMI device is powered.*

Figure 14 shows eTOP504 and eTOP507 with option modules PLCM01 mounted.

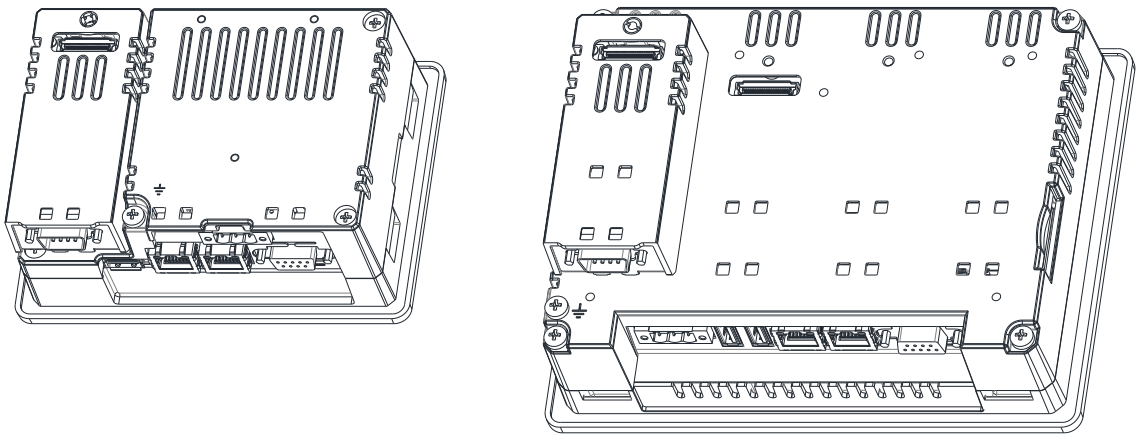


Figure 14

Slot Numbers

Series 500 and serie 600 HMI products have up to 4 slots available for optional modules. Slots are numbered from 1 to 4. Numbering of the slots is shown in Figure 15.

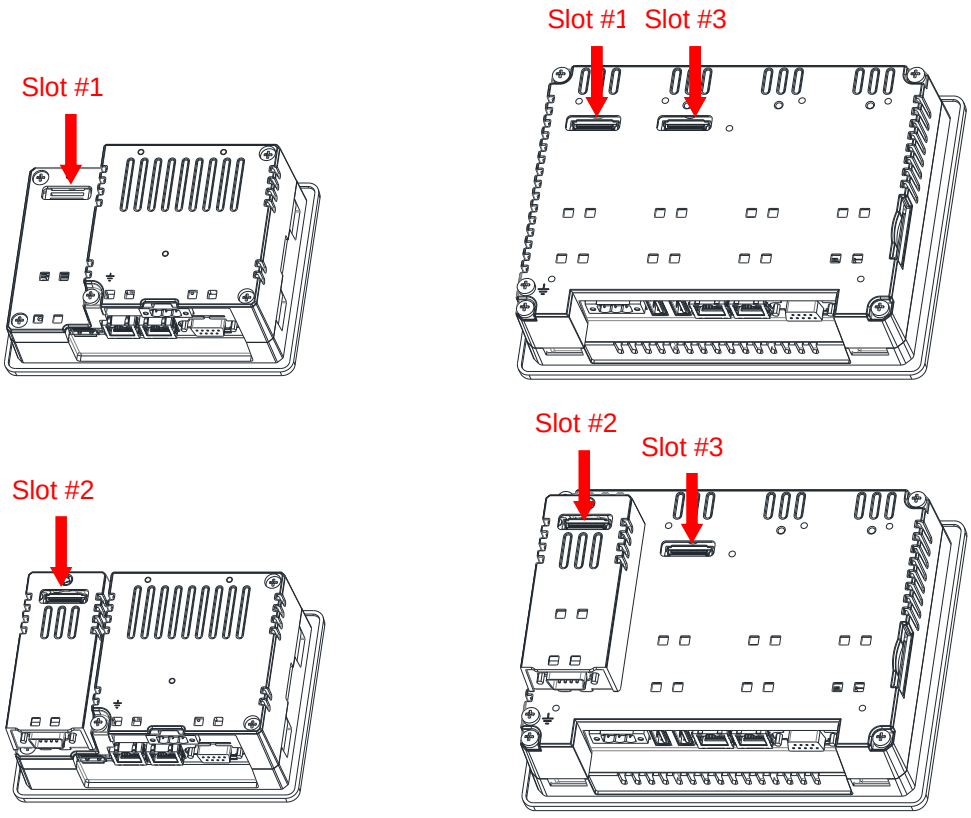


Figure 15

Slot #4 is located on the top of the option module plugged into Slot #3.

Recognizing optional modules installed on the operator panel

If correctly installed on the operator panel and recognized by the system the optional modules are listed into the Plugin List.
The Plugin List (Figure 16) is an item of the System Settings rotating menu.

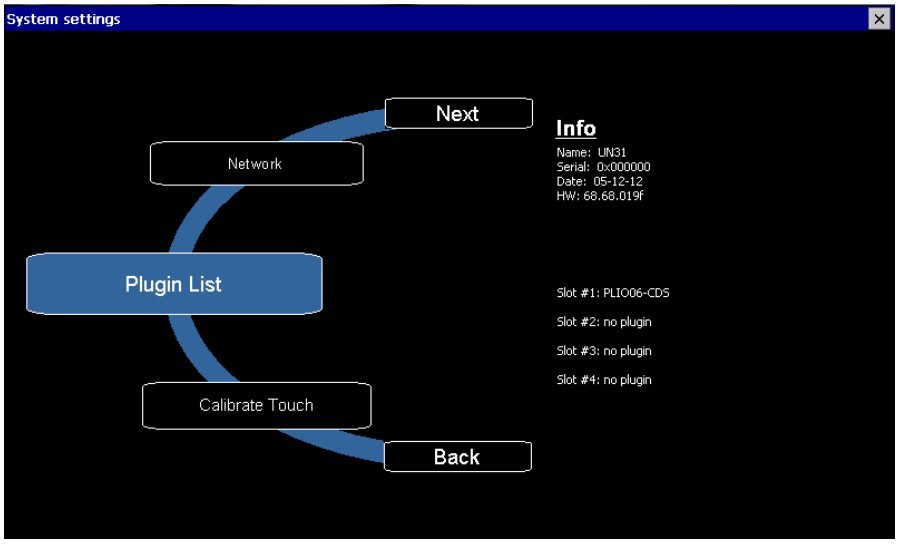


Figure 16

Configuring Integrated I/O

UniOP HMI allows creating control systems with integrated I/O. Integrated I/O systems based on option modules can be easily configured using CODESYS V3 PLC Configuration.

To add an I/O optional module, right click on Device (eTOP500) in the Project tree and select “Add Device”. The Add Device dialog is displayed (Figure 17), the PLIO optional modules are located under the Miscellaneous category, choose the integrated I/O module from the list and click on “Add Device” to add it to the current PLC configuration.

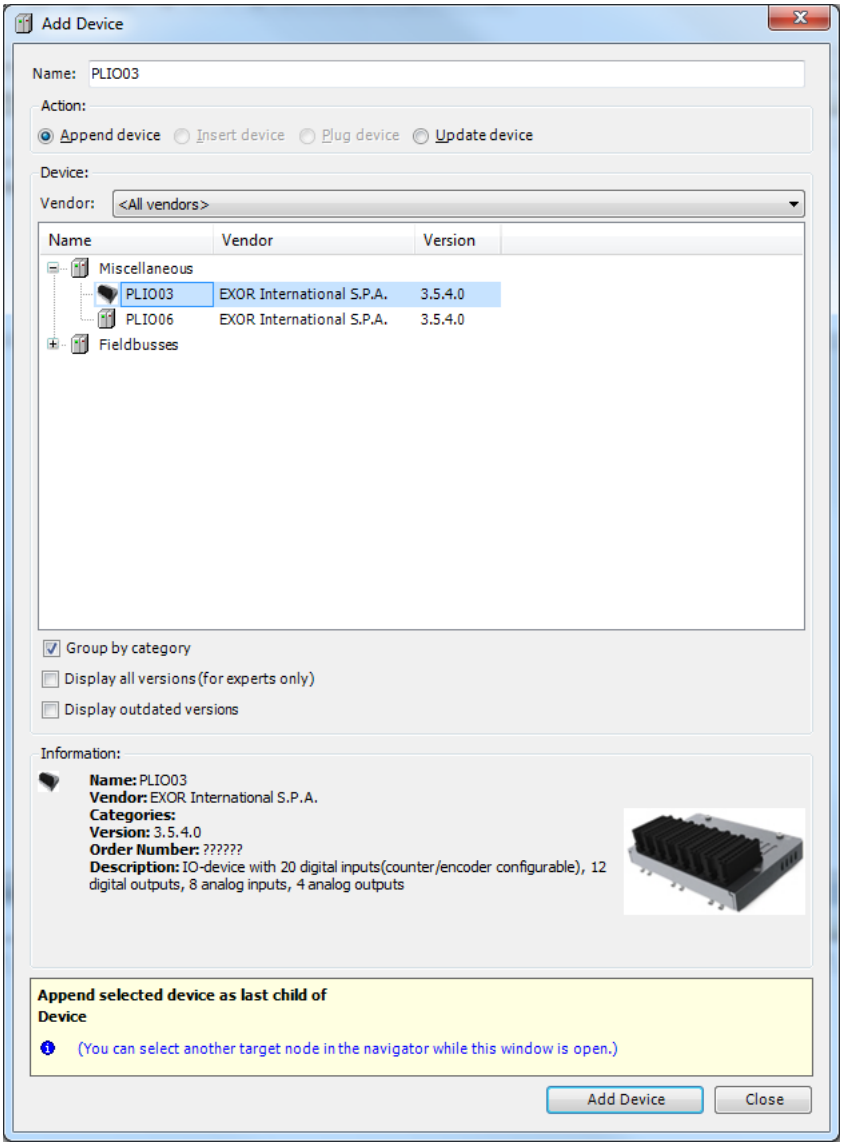


Figure 17

The parameters of all integrated I/O modules are collected in single tab of the PLC Configuration tree.

I/O Modules Diagnostic Address Mapping

When an I/O module is added to the PLC configuration the system automatically creates a Variable of type GetDiagPlio0x into the project, this variable contains the diagnostic information about the module.
The Variable name is assigned by default but can be changed by changing the I/O module name into the PLC configuration. The Variable is reported into the PLIO0x I/O Mapping tab, displayed in the work area by double clicking on I/O Module in the Project tree, as shown in Figure 18.

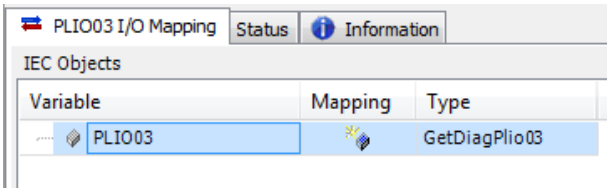


Figure 18

I/O Modules Points Address Mapping

CODESYS V3 allows three different Mapping modes:

- When no specific Mapping Variable is specified the system automatically assigns a physical address to each I/O point.
- A new Variable can be created by double clicking on the icon of the corresponding I/O point and writing the Variable name in the editing box. Such mapped I/O are marked with symbol.
- An existing variable created in the project can be assigned to the I/O point, by double clicking on the icon of the corresponding I/O point and selecting then the Variable from the list of available by clicking on the button. Such mapped I/O are marked with symbol. Using this method the I/O point will not refer anymore to the Physical address assigned by default from the system.

PLIO03

PLIO03 is a multifunction I/O module.
Due to the high level of integration and configurability, PLIO03 offers a “one-board solution” for most typical I/O configurations in simple applications.

PLIO03 specifications are the following:
- 20 optically isolated digital inputs; configurable as counter/encoder channels
- 12 optically isolated digital outputs
- 4 non-isolated (4 differential or 8 single ended channels) 12bit analog inputs configurable for voltage, current, resistance or temperature measurement
- 4 non-isolated 12 bit analog outputs configurable for voltage or current
- 1 dedicated PT100 channel-input for general usage or compensation of thermocouples.

Figure 19 shows a configuration for eTOP504 and eTOP507, using the PLCM01 and PLIO03 module.

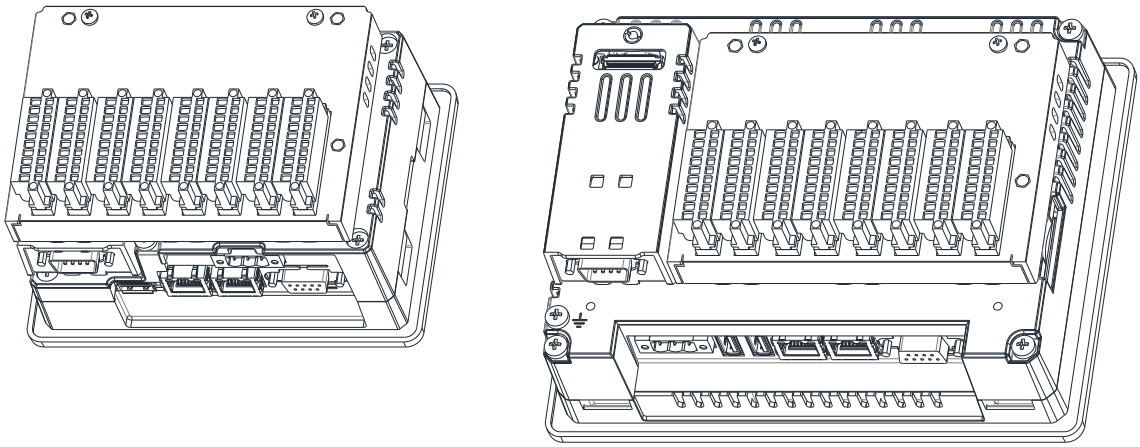


Figure 19

Note: *eTOP504 and eTOP605 panel models require the use of a Bus extender module for the installation of the PLIO03 I/O module.*

After including the PLIO03 module in the PLC configuration, as shown in Figure 20, Specific configuration steps must be performed for different parts of the module.

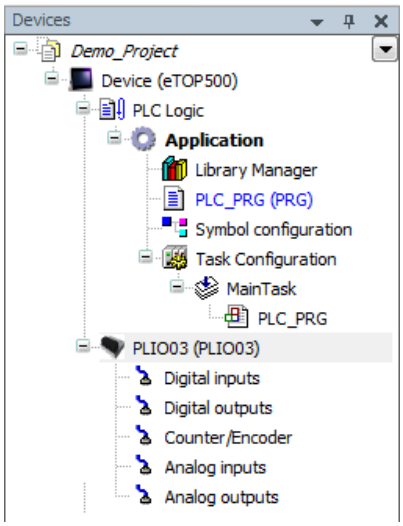


Figure 20

Configuring Digital Inputs

PLIO03 includes 20 programmable digital inputs.

The Mapping configuration of the PLIO03 Digital Inputs (Figure 21) is displayed in the Work area by selecting the Digital Inputs I/O Mapping tab after a double click on PLIO03 > Digital Inputs in the project tree.

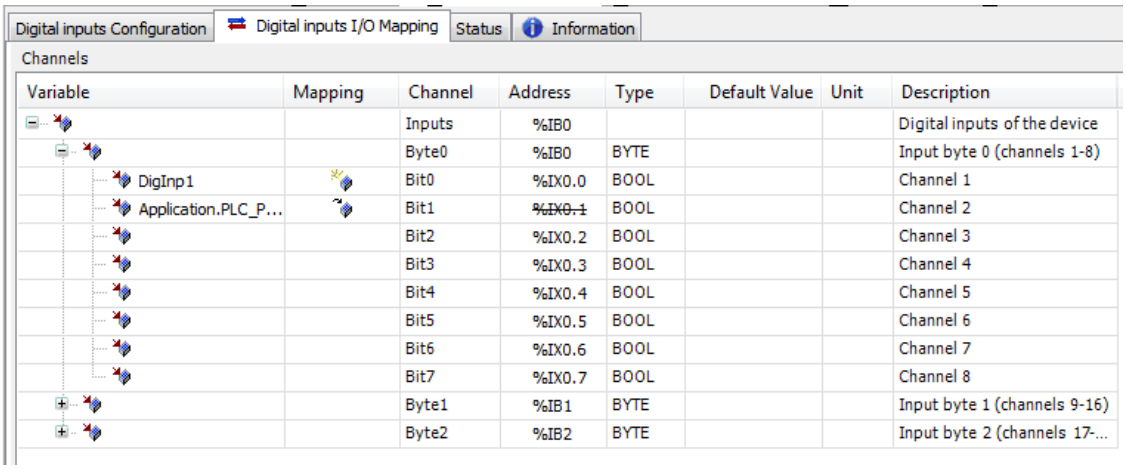


Figure 21

Debounce time parameter of the PLIO03 Digital Inputs (Figure 22) can be set by selecting the Digital Inputs Configuration tab after a double click on PLIO03 > Digital Inputs in the project tree.

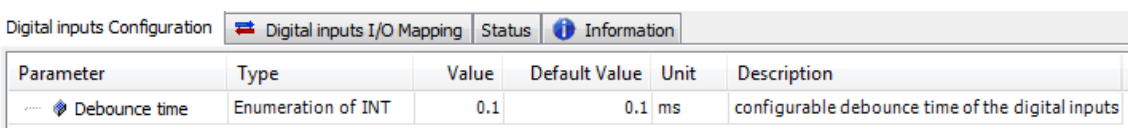
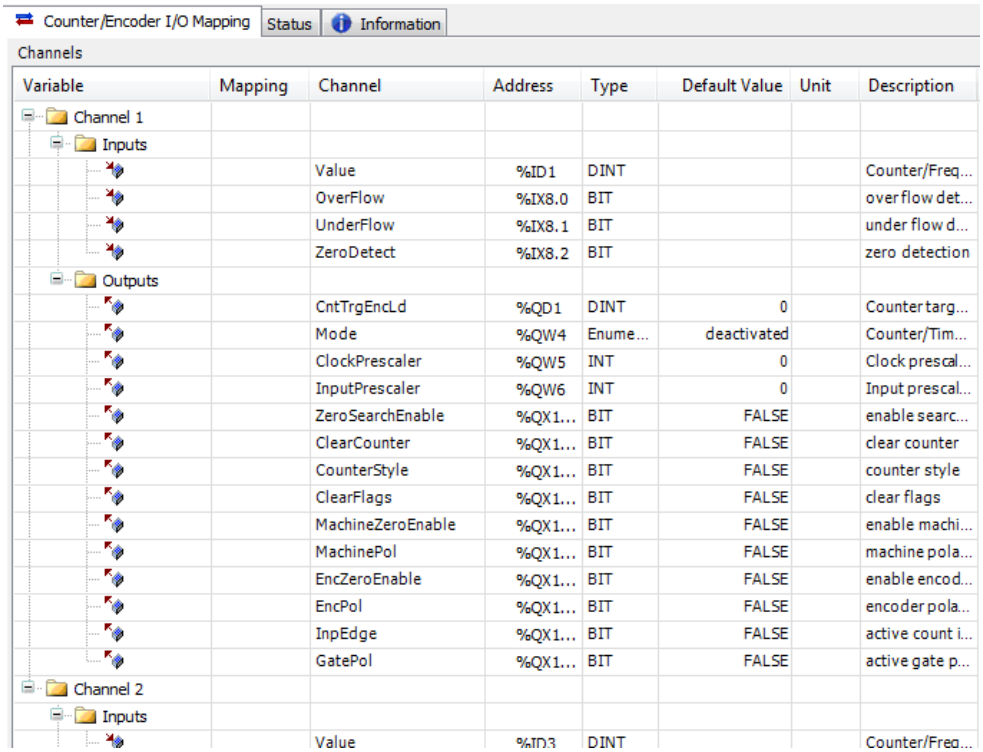


Figure 22

Counter/Encoder Inputs

PLIO03 includes 2 programmable Counter/Encoder channels.
Each Counter/Encoder channel is associated to a group of 4 digital inputs. When a channel is enabled, the associated digital inputs cannot be used as normal inputs. In case all the 2 Counter/Encoder channels are enabled 12 digital inputs are still available for normal operation.
Please refer to PLIO03 hardware manual for detailed specifications and wiring diagrams.

Configuration of Counter/Encoder inputs can be done by selecting the Counter/Encoder I/O Mapping tab after a double click on PLIO03 > Counter/Encoder in the project tree. Figure 23 shows the list of PLIO03 Counter/Encoder parameters as it appears in the CODESYS software.



Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
Channel 1							
Inputs							
		Value	%ID1	DINT			Counter/Freq...
		OverFlow	%IX8.0	BIT			over flow det...
		UnderFlow	%IX8.1	BIT			under flow d...
		ZeroDetect	%IX8.2	BIT			zero detection
Outputs							
		CntTrgEncLd	%QD1	DINT	0		Counter targ...
		Mode	%QW4	Enume...	deactivated		Counter/Tim...
		ClockPrescaler	%QW5	INT	0		Clock prescal...
		InputPrescaler	%QW6	INT	0		Input prescal...
		ZeroSearchEnable	%QX1...	BIT	FALSE		enable searc...
		ClearCounter	%QX1...	BIT	FALSE		clear counter
		CounterStyle	%QX1...	BIT	FALSE		counter style
		ClearFlags	%QX1...	BIT	FALSE		clear flags
		MachineZeroEnable	%QX1...	BIT	FALSE		enable machi...
		MachinePol	%QX1...	BIT	FALSE		machine pola...
		EncZeroEnable	%QX1...	BIT	FALSE		enable encod...
		EncPol	%QX1...	BIT	FALSE		encoder pola...
		InpEdge	%QX1...	BIT	FALSE		active count i...
		GatePol	%QX1...	BIT	FALSE		active gate p...
Channel 2							
Inputs							
		Value	%ID3	DINT			Counter/Freq...

Figure 23

The input signals associated to each Counter/Encoder channel have the following meaning:

Name	Data Type	Description
Value	DINT	Counter/frequency value
UnderFlow	BOOL	Underflow flag
OverFlow	BOOL	Overflow flag
ZeroDetect	BOOL	Zero found flag

Table 2

Board parameters have different options that can be selected at runtime using the PLC program. Description of parameters is given in Table 3.

Name	Data Type	Value	Description
CntTrgEncLd	DINT		Target value in Counter mode
Mode	INT	0	Counter/Encoder not active
		1	Quadrature incremental encoder counter
		2	Normal Counting
		3	Gated Counting
		4	Frequency Measurement
ClockPrescaler	INT		Clock frequency prescaler
InputPrescaler	INT		Input frequency prescaler
ZeroSearchEnable	BOOL		Enable zero search when in encoder mode. When true the counter resets to zero. ZeroFoundF reports when zero has been reached.
ClearCounter	BOOL		Reset counter value when in counter mode
CounterStyle	BOOL		Valid in counter mode, when true the count value resets when the counter reaches the target value
ClearFlags	BOOL		Reset overflow and underflow flags
MachineZeroEnable	BOOL	false	Corresponding terminal block is used as normal digital input
		true	Corresponding terminal block is used as Machine Zero Input
MachinePol	BOOL	false	Means Input is active LOW
		true	Means Input is active HIGH
EncZeroEnable	BOOL	false	Corresponding terminal block is used as normal digital input
		true	Corresponding terminal block is used as Encoder Zero Input
EncPol	BOOL	false	Means Input is active LOW
		true	Means Input is active HIGH
InpEdge	BOOL	false	Active count input edge negative
		true	Input edge positive
GatePol	BOOL	false	Defines signal level for Gate LOW
		true	Signal level for Gate HIGH

Table 3

When the input is configured for use with an incremental encoder, then the Encoder Zero Search procedure is usually performed at power-up. PLC program should start this procedure enabling the Zero Search using the parameterZeroSearchE for the selected channel.

Frequency measurement (Mode 4) is based on the following formula:

$$\text{Frequency}=\frac{\text{Internal Clock Frequency}*\text{Number of counts}*(\text{InputScaler}+1)}{\text{Number of counts Internal Clock}*(\text{ClockPresc}+1)}$$

The I/O driver calculates the frequency that is returned in the input signal Value.
One frequency measurement cycle terminates when there is at least one input count and at least 65536 counts in the internal clock.

As an example, to cover the frequency range 1Hz to 20KHz you have to set:

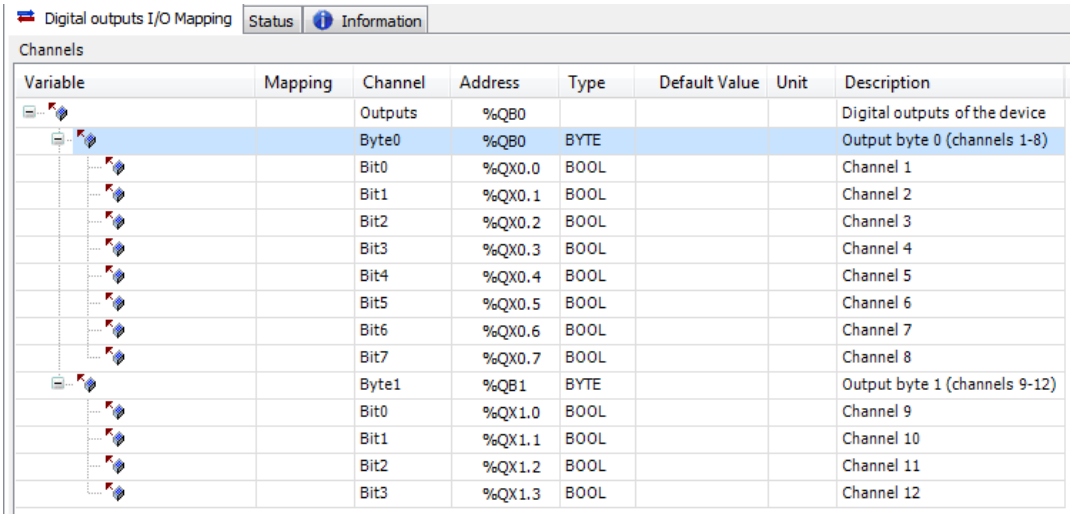
Range	ClockPrescaler	InputPrescaler
1Hz to 2Hz	0	1
2Hz to 20KHz	0	0

Table 4

Configuring Digital Outputs

PLIO03 includes 12 digital outputs.

The Mapping configuration of the PLIO03 Digital Outputs (Figure 24) is displayed in the Work area by selecting the Digital Outputs I/O Mapping tab after a double click on PLIO03 > Digital Outputs in the project tree.



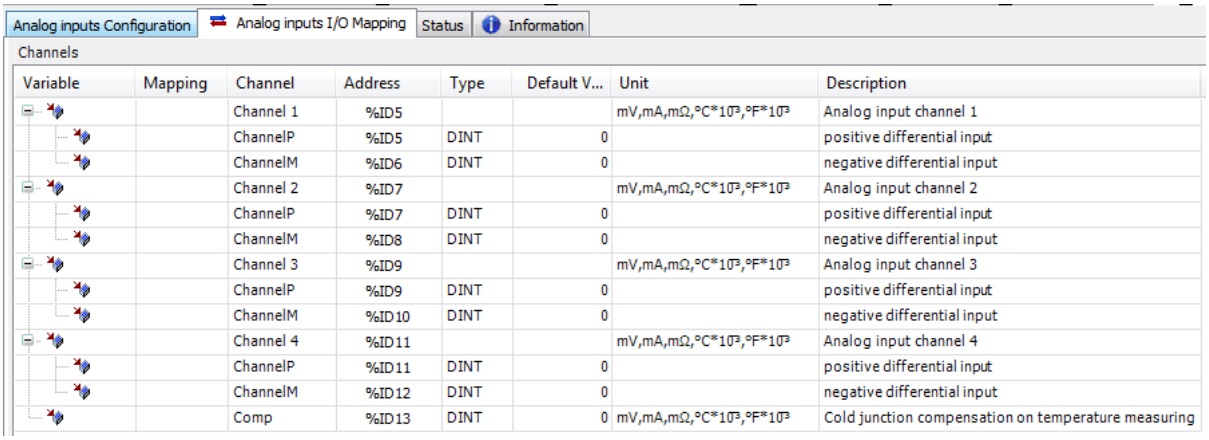
Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Outputs	%QB0				Digital outputs of the device
		Byte0	%QB0	BYTE			Output byte 0 (channels 1-8)
		Bit0	%QX0.0	BOOL			Channel 1
		Bit1	%QX0.1	BOOL			Channel 2
		Bit2	%QX0.2	BOOL			Channel 3
		Bit3	%QX0.3	BOOL			Channel 4
		Bit4	%QX0.4	BOOL			Channel 5
		Bit5	%QX0.5	BOOL			Channel 6
		Bit6	%QX0.6	BOOL			Channel 7
		Bit7	%QX0.7	BOOL			Channel 8
		Byte1	%QB1	BYTE			Output byte 1 (channels 9-12)
		Bit0	%QX1.0	BOOL			Channel 9
		Bit1	%QX1.1	BOOL			Channel 10
		Bit2	%QX1.2	BOOL			Channel 11
		Bit3	%QX1.3	BOOL			Channel 12

Figure 24

Configuring Analog Inputs

PLIO03 includes 4 differential programmable analog input channels plus a dedicated PT100 channel for cold Junction compensation or Temperature measurement.
Each channel can be configured as single-ended voltage measurement resulting in having 8 voltage analog inputs.

The Mapping configuration of the PLIO03 Analog Inputs (Figure 25) is displayed in the Work area by selecting the Analog Inputs I/O Mapping tab after a double click on PLIO03 > Analog Inputs in the project tree. Note that 9 analog values are produced by the module.



Variable	Mapping	Channel	Address	Type	Default V...	Unit	Description
		Channel 1	%ID5			mV,mA,mΩ,°C*10³,°F*10³	Analog input channel 1
		ChannelP	%ID5	DINT	0		positive differential input
		ChannelM	%ID6	DINT	0		negative differential input
		Channel 2	%ID7			mV,mA,mΩ,°C*10³,°F*10³	Analog input channel 2
		ChannelP	%ID7	DINT	0		positive differential input
		ChannelM	%ID8	DINT	0		negative differential input
		Channel 3	%ID9			mV,mA,mΩ,°C*10³,°F*10³	Analog input channel 3
		ChannelP	%ID9	DINT	0		positive differential input
		ChannelM	%ID10	DINT	0		negative differential input
		Channel 4	%ID11			mV,mA,mΩ,°C*10³,°F*10³	Analog input channel 4
		ChannelP	%ID11	DINT	0		positive differential input
		ChannelM	%ID12	DINT	0		negative differential input
		Comp	%ID13	DINT	0	mV,mA,mΩ,°C*10³,°F*10³	Cold junction compensation on temperature measuring

Figure 25

Table 5 shows the different meaning assumed by the 9 values produced by the Analog Input module depending on the measurement mode selected with the configuration parameters.

	Module Signal	Mode			
		Voltage Differential	Voltage Single-ended	Resistance 2/3 wires	Current
1	AI1 Channel P	Value	Value	Value	Value
2	AI1 Channel M	-	Value	-	-
3	AI2 Channel P	Value	Value	Value	Value
4	AI2 Channel M	-	Value	-	-
5	AI3 Channel P	Value	Value	Value	Value
6	AI3 Channel M	-	Value	-	-
7	AI4 Channel P	Value	Value	Value	Value
8	AI4 Channel M	-	Value	-	-
9	COMP	Resistance	-	-	-

Table 5

Channel 9 is an input channel dedicated to cold junction compensation of thermocouples. Please refer to the hardware description manual for additional details.

PLIO03 Analog Input channels have specific parameters that must to be properly configured according to the operation mode requested for each channel: the operating mode and the full-scale range. Configuration of Analog inputs, shown in Figure 26 can be reached by selecting the Analog Inputs Configuration tab after a double click on PLIO03 > Analog Input in the project tree.

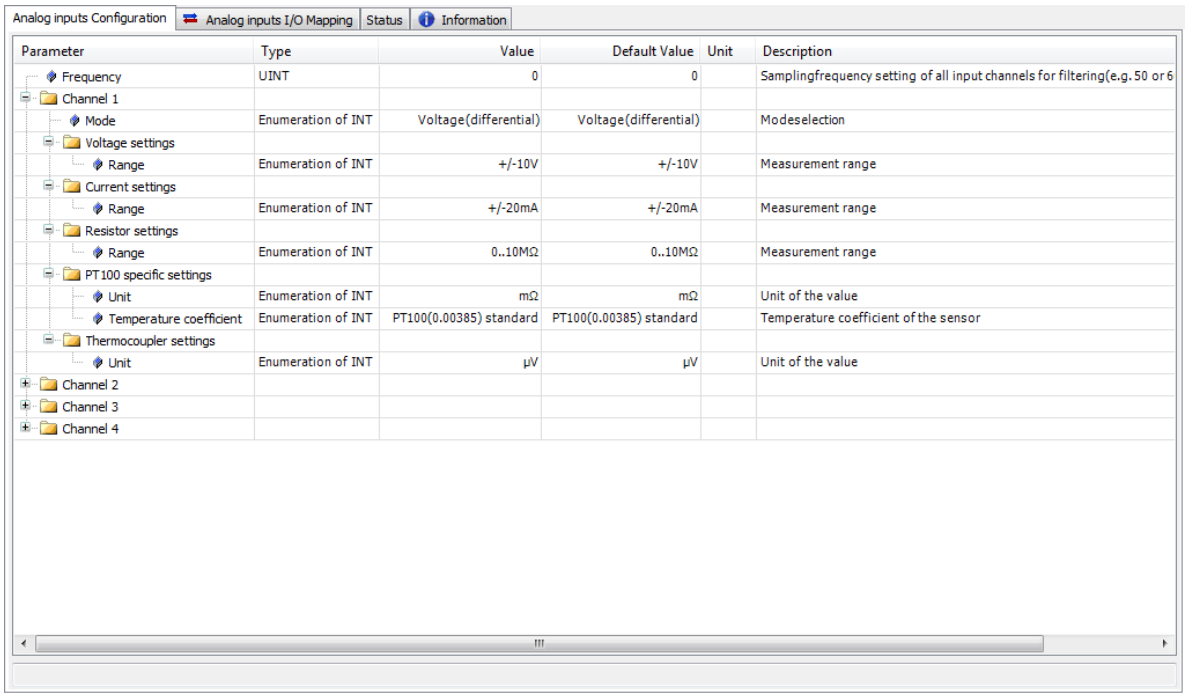


Figure 26

Configuring Channel Measurement mode

Each channel is independently programmable; six different types of measurements can be executed. The type of measure is defined by the Channel Mode setting, Table 6 shows the values that can be configured.

Parameter value	Measurement Mode	Unit
0	Voltage (Differential)	µV
1	Voltage (Single Ended)	µV
2	Current (4-20mA)	µA
3	Resistance (2-wire)	mΩ
4	Resistance (3-wire)	mΩ
5	Resistance (4-wire)	mΩ
6	Thermocoupler (Differential)	µV
7	Thermocoupler (Single Ended)	µV
8	Current (0-20mA)	µA
9	PT100 (2-wire)	mΩ
10	PT100 (3-wire)	mΩ
11	PT100 (4-wire)	mΩ

Table 6

Configuring Channel Full Scale

Measurement range parameter value changes depending on the Channel Mode value set for the Channel, for Channel Mode values from 0 to 5 the Full Scale parameter can be set at programming time. Possible values are reported tables.

Voltage	
Parameter value	**Range**
0	±100mV
1	±500mV
2	±1V
3	±5V
4	±10V
5	0 - 1V
6	0 - 10V

Table 7

Current	
Parameter value	**Range**
0	±2mA
1	±10mA
2	±20mA

Table 8

Resistance	
Parameter value	**Range**
0	0 - 80Ω
1	0 - 400Ω
2	0 - 900Ω
3	0 - 8KΩ
4	0 - 10MΩ
5	0 - 1MΩ

Table 9

For Channel Mode values from 6 to 8 the Range parameter value is set as default by the system.

Parameter value	Measurement Mode	Range
6	Thermocoupler (Differential)	+/- 100mV
7	Thermocoupler (Single Ended)	+/- 100mV
8	Current (0-20mA)	0 - 1V

Table 10

For Channel Mode values from 9 to 11 the Range value is the one reported in Table 11.

Parameter value	Full Scale Value
0	0-157Ω

Table 11

Frequency parameter

The Frequency parameter may be used to synchronize analog measurement to reduce powerline noise. Value 0 means free running measurement, value 50 will take measurements synchronized with powerline at 50Hz. This option is useful with high-gain measurements as in thermocouple channels.

Configuring Analog Outputs

PLIO03 includes 4 programmable analog output channels.

The Mapping configuration of the PLIO03 Analog Outputs (Figure 27) is displayed in the Work area by selecting the Analog Outputs I/O Mapping tab after a double click on PLIO03 > Analog Outputs in the project tree.

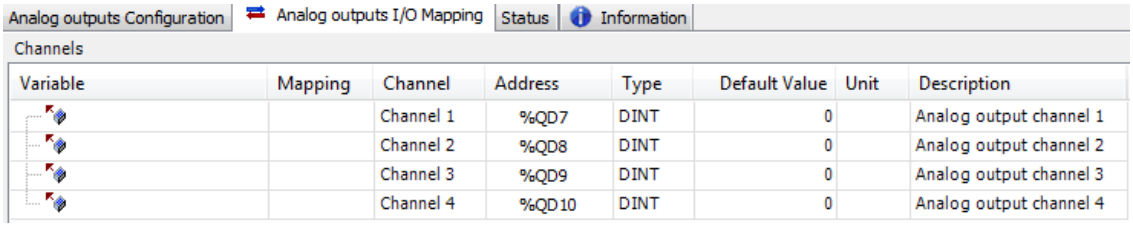


Figure 27

PLIO03 Analog Output channels have specific parameter that must to be properly configured according to the operation mode requested for each channel: the Channel Value. Configuration of Analog Outputs, shown in Figure 28 can be reached by selecting the Analog Outputs Configuration tab after a double click on PLIO03 > Analog Output in the project tree.

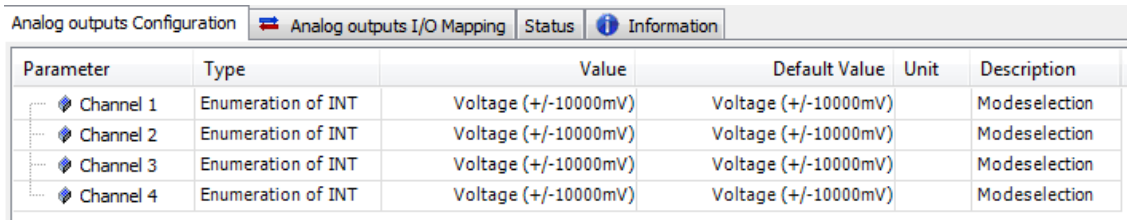


Figure 28

Each of the 4 channels can be independently programmed to be used as voltage or current output. The Channel Value parameter can be configured with the values reported in Table 12.

Parameter value	Value
0	Voltage (+/-10000mV)
1	Current (0..20000µA)

Table 12

The output range is fixed as specified in the table. The value written by the PLC program to the output channels is an integer value between 0 and 10.000 in case of voltage and between 0 and 20.000 in case of current.

PLIO06

PLIO06 is a compact I/O module. It has been designed for creating simple applications with a limited number of digital I/O signals.
PLIO06 specifications are the following:

- 8 optically isolated digital inputs
- 6 optically isolated digital outputs
- 1 relay output

Figure 29 shows a configuration for eTOP504 and eTOP507 using the PLIO06 module.

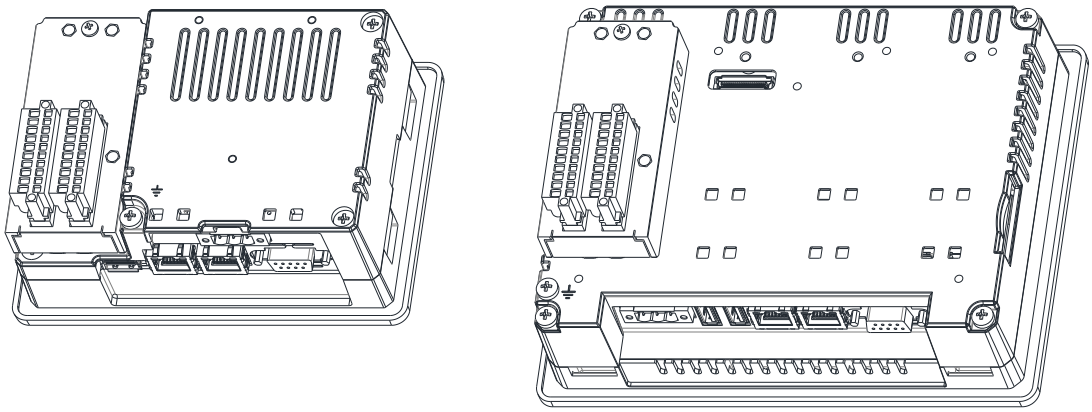


Figure 29

PLIO06 module must be included in the PLC configuration, as shown in Figure.

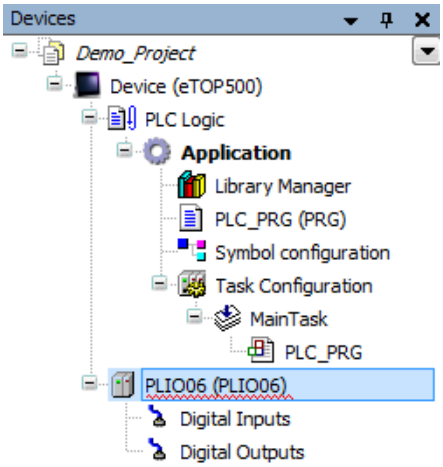


Figure 30

I/O configurations with two PLIO06 are allowed. In such a case you have to specify the Slot number for each module. Configure Slot Number as shown in Figure 31.
PLIO06 must have a Slot Number from 1 to 4. An error will be reported at run-time if the I/O module has not been plugged in the slot specified in the I/O configuration.

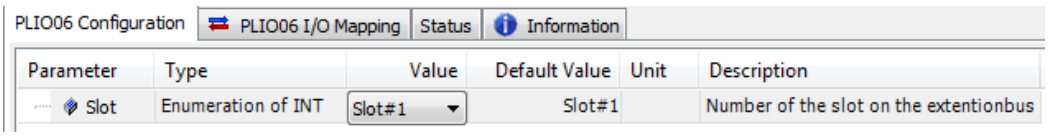
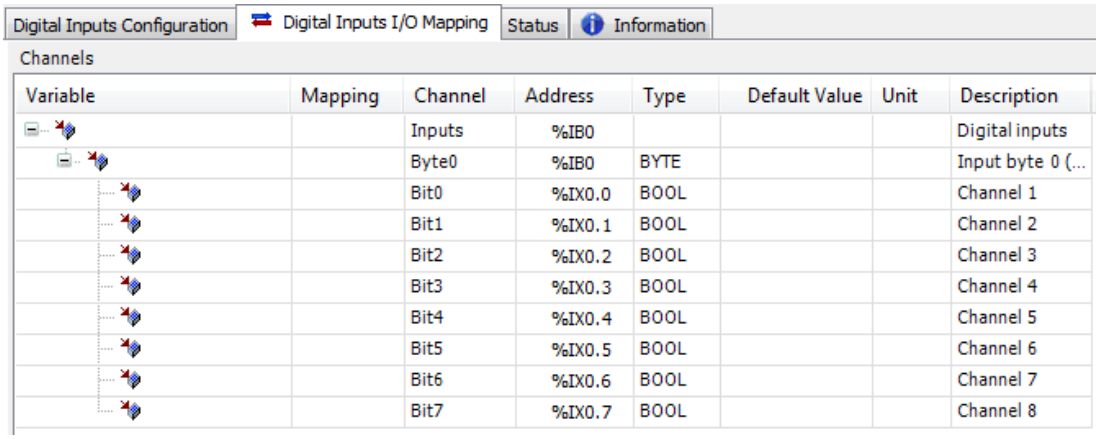


Figure 31

Configuring Digital Inputs

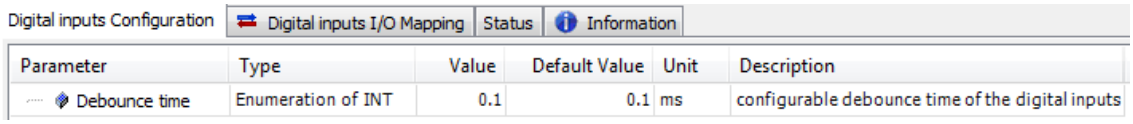
PLIO06 includes 8 digital inputs.
Mapping configuration of the PLIO06 Digital Inputs (Figure 32) is displayed in the Work area by selecting the Digital Inputs I/O Mapping tab after a double click on the PLIO06 > Digital Inputs item in the project tree.



Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Inputs	%IB0				Digital inputs
		Byte0	%IB0	BYTE			Input byte 0 (...
		Bit0	%IX0.0	BOOL			Channel 1
		Bit1	%IX0.1	BOOL			Channel 2
		Bit2	%IX0.2	BOOL			Channel 3
		Bit3	%IX0.3	BOOL			Channel 4
		Bit4	%IX0.4	BOOL			Channel 5
		Bit5	%IX0.5	BOOL			Channel 6
		Bit6	%IX0.6	BOOL			Channel 7
		Bit7	%IX0.7	BOOL			Channel 8

Figure 32

Debounce time parameter of the PLIO06 Digital Inputs (Figure 33) can be set by selecting the Digital Inputs Configuration tab after a double click on the PLIO06 > Digital Inputs item in the project tree.

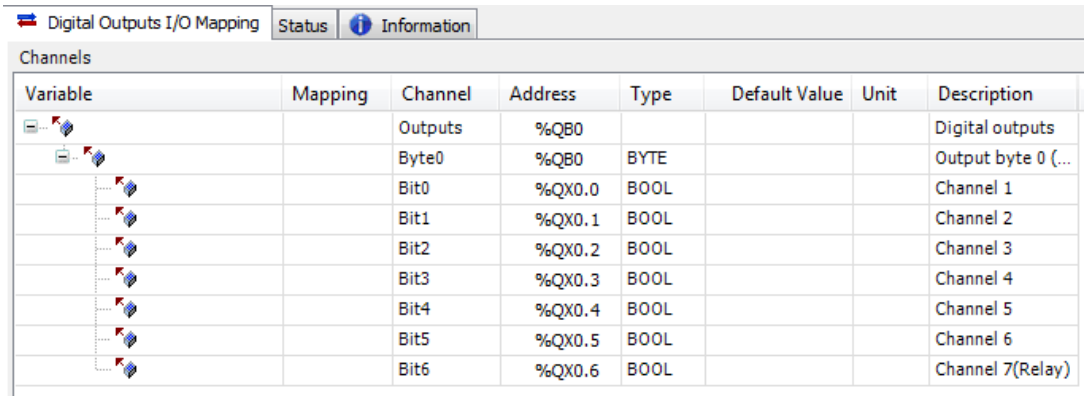


Parameter	Type	Value	Default Value	Unit	Description
Debounce time	Enumeration of INT	0.1	0.1	ms	configurable debounce time of the digital inputs

Figure 33

Configuring Digital Outputs

PLIO06 includes 6 digital outputs plus 1 relay output.
The Mapping configuration of the PLIO06 Digital Outputs (Figure 24) is displayed in the Work area by selecting the Digital Outputs I/O Mapping tab after a double click on the PLIO06 > Digital Outputs item in the project tree.



Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Outputs	%QB0				Digital outputs
		Byte0	%QB0	BYTE			Output byte 0 (...
		Bit0	%QX0.0	BOOL			Channel 1
		Bit1	%QX0.1	BOOL			Channel 2
		Bit2	%QX0.2	BOOL			Channel 3
		Bit3	%QX0.3	BOOL			Channel 4
		Bit4	%QX0.4	BOOL			Channel 5
		Bit5	%QX0.5	BOOL			Channel 6
		Bit6	%QX0.6	BOOL			Channel 7(Relay)

Figure 34

The IOBrdCfg Function Block

Integrated I/O boards requires to be configured at design-time from the I/O specific configuration page, in some cases it is also possible to change the configuration of the I/O board in runtime, to do this the IOBrdCfg function block is provided.

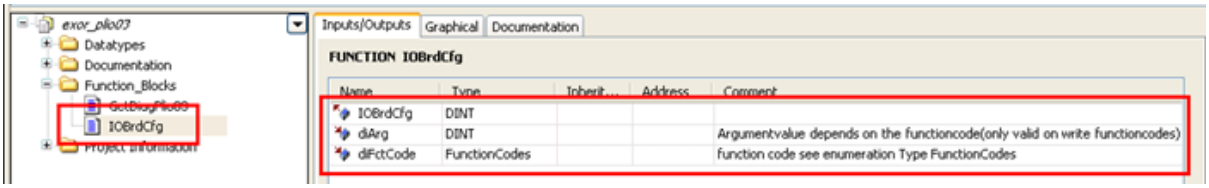


Figure 35

This function block is included in the exor_plio03 library and has the following input parameters:

diFctCode	This parameter is passed to the I/O board; it significance changes from board to board.
diArg	This parameter is passed to the I/O board; its significance varies from board to board.

The function has the following output parameters:

IOBrdCfg	This is the return value; its significance varies from board to board and from FunctionCode to FunctionCode.

Modbus TCP

CODESYS V3 runtime can use the built-in Ethernet interface of the operator panels for the distributed Modbus TCP network. No additional hardware is required.
The system can act as Modbus TCP Master or Slave; both configurations are available at the same time.
To add an Ethernet Modbus TCP interface two steps are required.
Right click on Device (eTOP500) in the Project tree and select “Add Device”. The Add Device dialog is displayed (Figure 17), the Ethernet device is located under the Fieldbuses > Ethernet Adapter category, choose the Ethernet device from the list and click on “Add Device” to add it to the current PLC configuration.

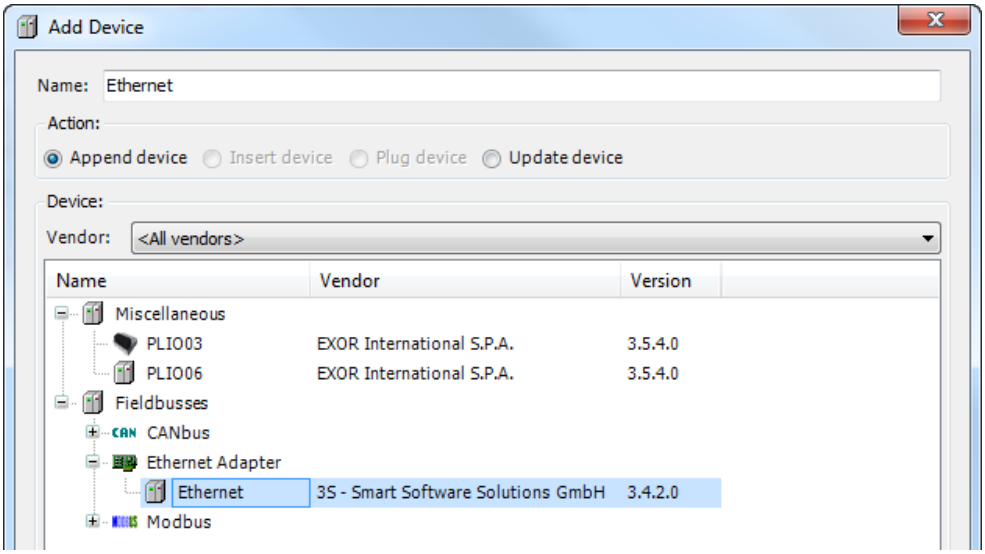


Figure 36

Right click on Ethernet (Ethernet) in the Project tree and select “Add Device”.
Modbus TCP Master and Slave devices are located under the categories Modbus > Modbus TCP Master/Slave (Figure 37); select the required device from the list and click on “Add Device” to add it to the current PLC configuration.

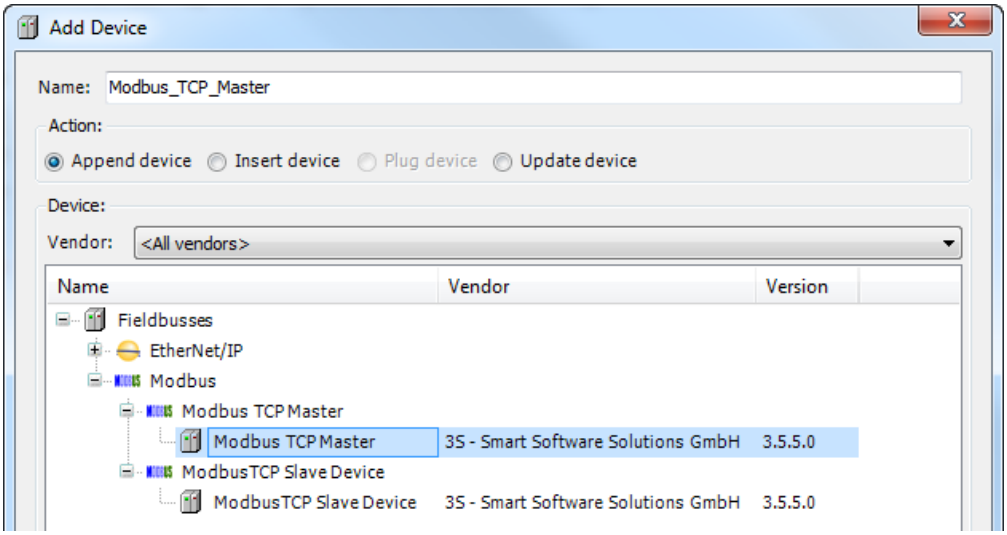


Figure 37

Modbus TCP Master Configuration

Modbus TCP Master configuration (Figure 32) is displayed in the Work area by selecting the Modbus TCP Master Configuration tab after a double click on Ethernet > Modbus TCP Master in the project tree.

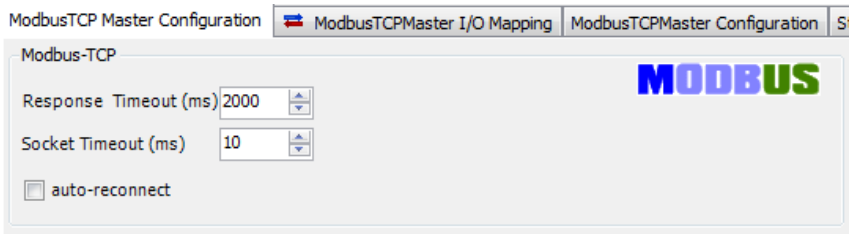


Figure 38

Available parameters are:

Response Timeout (ms) Timeout for Modbus slaves reply, given in milliseconds.

Socket Timeout (ms) Timeout for Socket reply, given in milliseconds.

Auto-reconnect If set auto-confirm error and re-establish TCP connection.

Add and Configure Remote Modbus TCP Slave Devices

To add a remote Modbus TCP Slave device, right click on Ethernet > Modbus TCP Master in the project tree and select “Add Device”. The Modbus TCP Slave devices are located under the category Modbus > Modbus TCP Slave (Figure 39); choose the device from the list and click on “Add Device” to add it to the current PLC configuration.

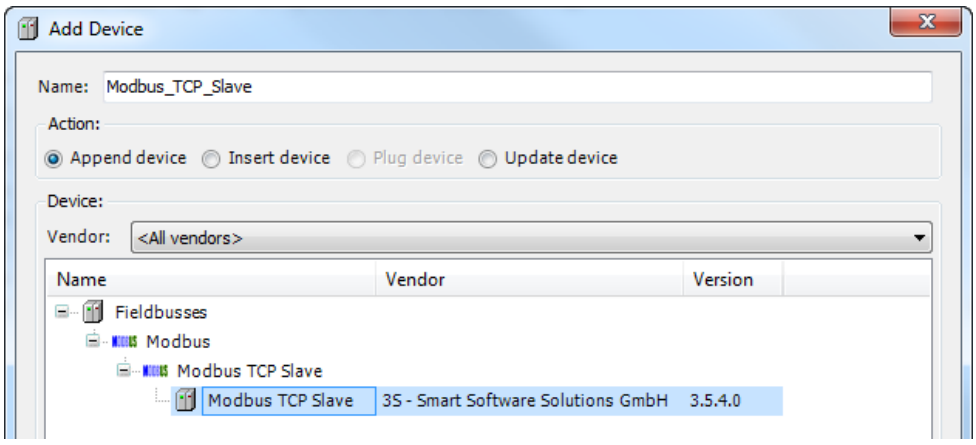


Figure 39

The remote Modbus TCP slave configuration (Figure 40) is displayed in the Work area by selecting the Modbus TCP Slave tab after a double click on Ethernet > Modbus TCP Master > Modbus TCP Slave in the project tree.

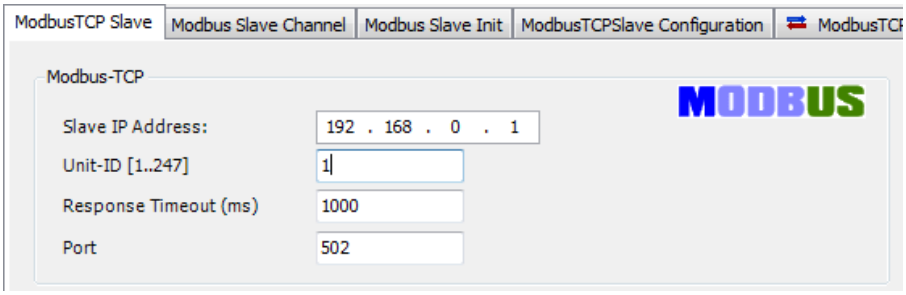


Figure 40

Available parameters are:

Slave IP Address IP Address of the Modbus TCP Slave Device.

Unit-ID Modbus Node ID of the Modbus TCP Slave Device

Response Timeout (ms) Timeout for Modbus slaves reply, given in milliseconds.

Port TCP port used for the communication with the Modbus TCP Slave device.

Configuring Modbus Data Exchange

The Modbus data exchange configuration with the Modbus TCP slave can be done in the Work area by selecting the Modbus Slave Channel tab after a double click on Ethernet > Modbus TCP Master > Modbus TCP Slave in the project tree.
The configuration is based on Channels, for each channel you can configure a Modbus command that will be sent to the Slave. To add a new Channel click on Add Channel button, as shown in Figure 41, the ModbusChannel dialog will be displayed in page.

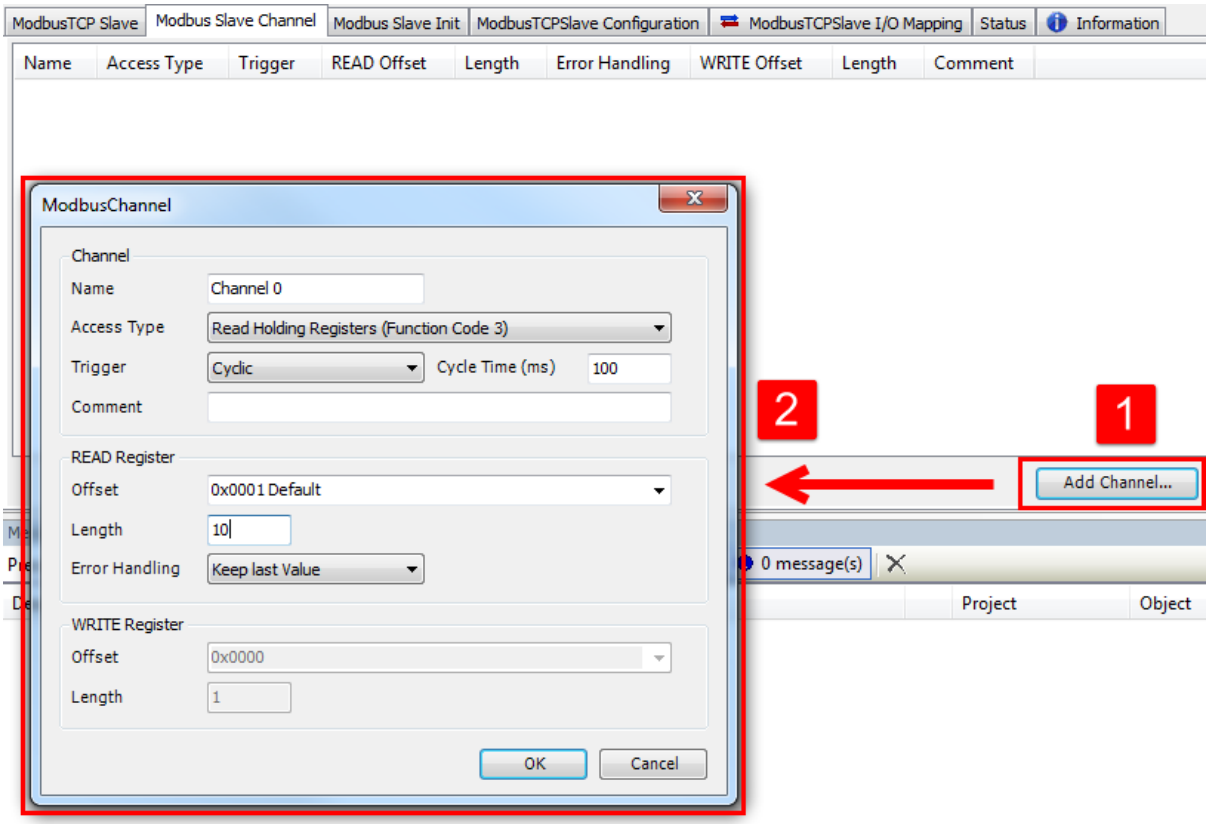


Figure 41

Available parameters are:

Name Channel Name

Access Type Selection of the Modbus command.

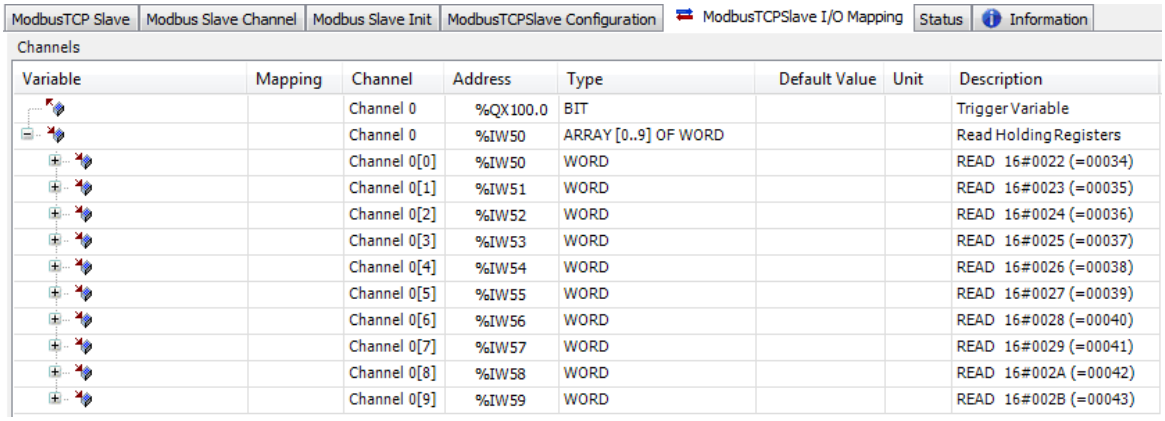
Trigger Specifies is the command must be Cyclic, in this case the Cycle Time (ms) must be specified or Rising Edge, in this case the command is launched on the rising edge event of a bit variable defined into the Modbus TCP Slave I/O Mapping.

Comment User comment if required.

Offset The starting Modbus address

Length Number of registers to be Read/Write

The Mapping configuration of the Modbus TCP Slave I/O (Figure 42) is displayed in the Work area by selecting the Modbus TCP Slave I/O Mapping tab after a double click on Ethernet > Modbus TCP Master > Modbus TCP Slave in the project tree. The Mapping shows a list of all the Modbus Resources read/write in the configured Channels, in case the configured Channel uses a Rising Edge triggered command the Trigger bit is listed into the Mapping.



Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Channel 0	%QX100.0	BIT			Trigger Variable
		Channel 0	%IW50	ARRAY [0..9] OF WORD			Read Holding Registers
		Channel 0[0]	%IW50	WORD			READ 16#0022 (=00034)
		Channel 0[1]	%IW51	WORD			READ 16#0023 (=00035)
		Channel 0[2]	%IW52	WORD			READ 16#0024 (=00036)
		Channel 0[3]	%IW53	WORD			READ 16#0025 (=00037)
		Channel 0[4]	%IW54	WORD			READ 16#0026 (=00038)
		Channel 0[5]	%IW55	WORD			READ 16#0027 (=00039)
		Channel 0[6]	%IW56	WORD			READ 16#0028 (=00040)
		Channel 0[7]	%IW57	WORD			READ 16#0029 (=00041)
		Channel 0[8]	%IW58	WORD			READ 16#002A (=00042)
		Channel 0[9]	%IW59	WORD			READ 16#002B (=00043)

Figure 42

Modbus TCP Slave Configuration

Modbus TCP Slave configuration (Figure 43) is displayed in the Work area by selecting the Modbus TCP tab after a double click on Ethernet > Modbus TCP Slave Device in the project tree.

Note: *When programmed as explained in this chapter the CODESYS V3 PLC will act as a Modbus TCP Slave device. To configure the device for communication with remote Modbus TCP I/O modules please refer to the chapter on Modbus TCP Master configuration.*

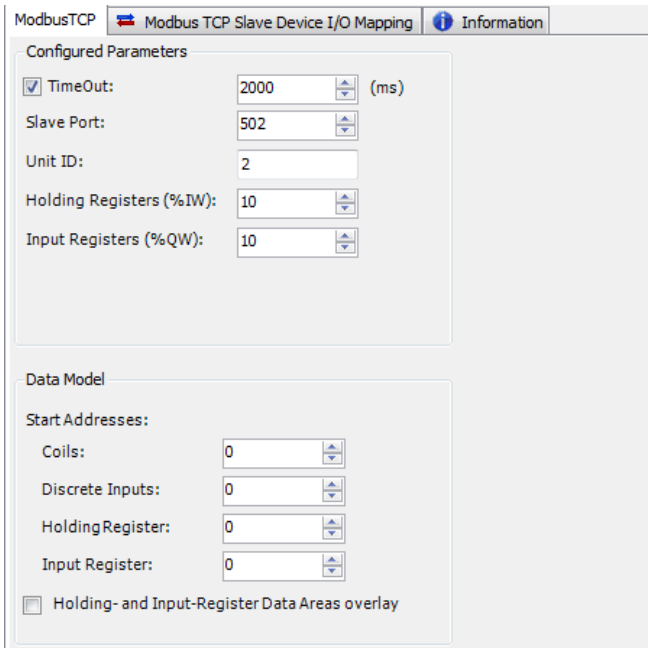


Figure 43

Available parameters are:

Timeout If selected, timeout for Modbus Master queries, given in milliseconds.

Slave port TCP port used for the communication with the Modbus TCP Slave device

Unit-ID Modbus Node ID of the Modbus TCP Slave Device

Holding Registers (%IW)	Number of holding registers available in PLC memory
Input Registers (%QW)	Number of input registers available in PLC memory
Start address	Starting address for Modbus resources in PLC

The Mapping configuration of the Modbus TCP Slave Device (Figure 44) is displayed in the Work area by selecting the Modbus TCP Slave Device I/O Mapping tab after a double click on Ethernet > Modbus TCP Slave Device in the project tree.

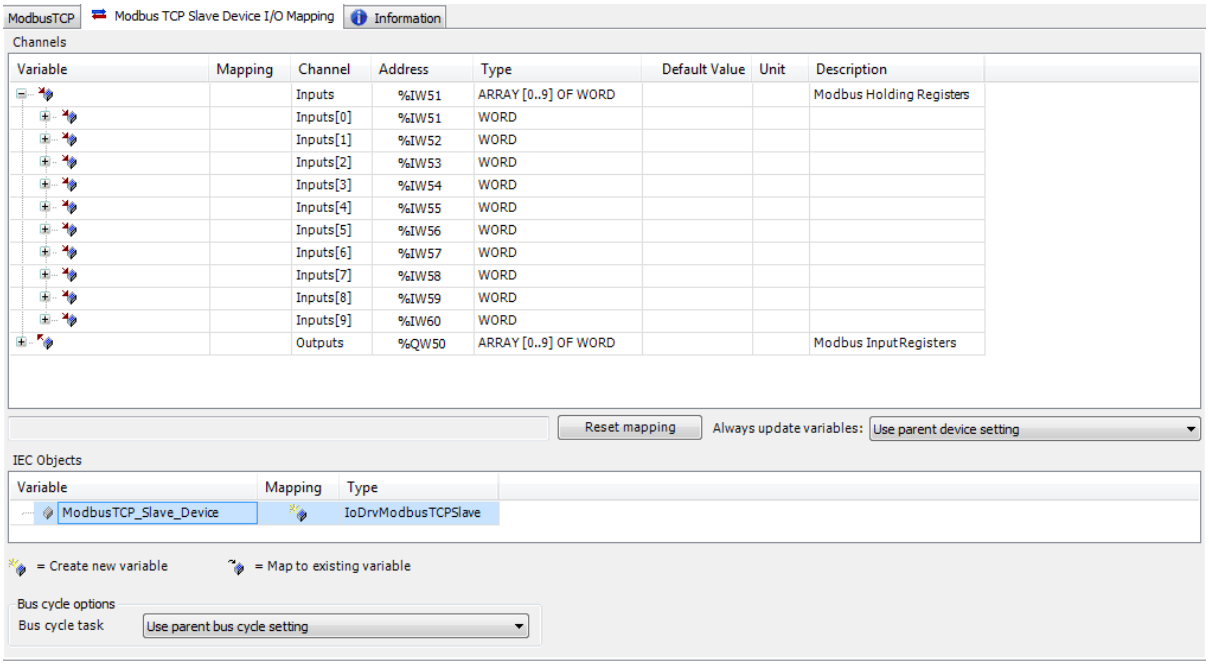


Figure 44

Modbus RTU

CODESYS V3 runtime can use the built-in serial interface of the HMI device for the distributed Modbus RTU network. One single serial interface is available as built-in option for the eTOP500 and eTOP600 Series HMI. With the use of optional plug-in modules it is possible to have up to 3 serial interfaces on the device.
The system can act as Modbus RTU Master or Slave; both configurations are available at the same time.
To add a Modbus RTU interface two consecutive steps are required.
Right click on Device (eTOP500) in the Project tree and select “Add Device”. The Add Device dialog is displayed (Figure 45), the Modbus COM device is located under the Fieldbuses > Modbus > Modbus Serial port category, choose the Modbus COM device from the list and click on “Add Device” to add it to the current PLC configuration.

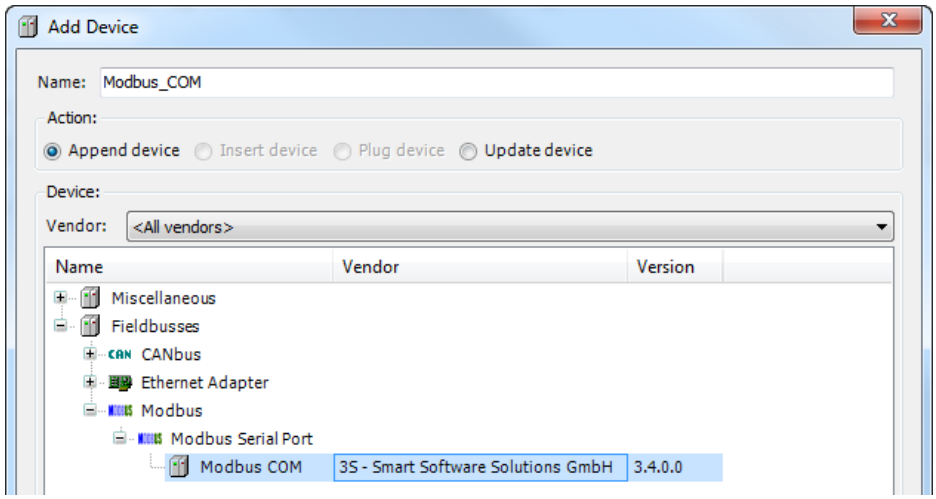


Figure 45

Right click on Modbus COM in the Project tree and select “Add Device”. Modbus RTU Master and Slave devices are located under the Fieldbuses > Modbus > Modbus Serial Master/Device categories (Figure 46), choose the required device from the list and click on “Add Device” to add it to the current PLC configuration.

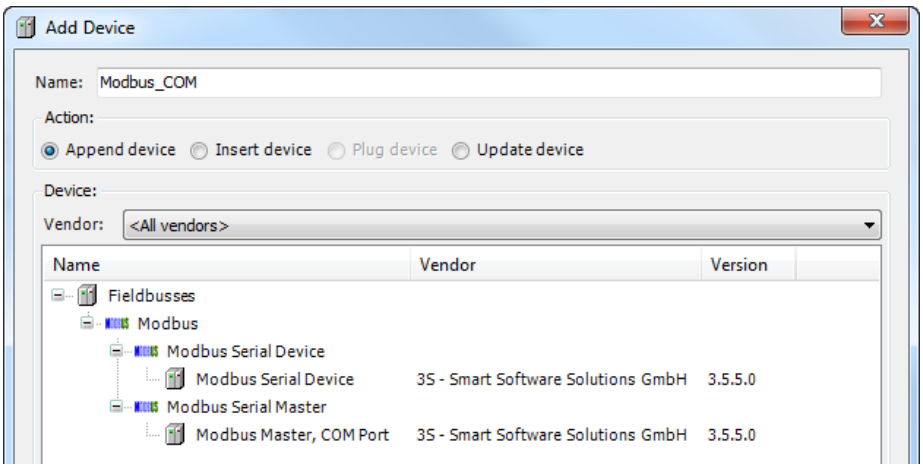


Figure 46

Modbus RTU Serial Port Configuration

Modbus RTU Serial port configuration (Figure 47) is displayed in the Work area by selecting the Modbus Serial Port Configuration tab after a double click on Modbus COM in the project tree.

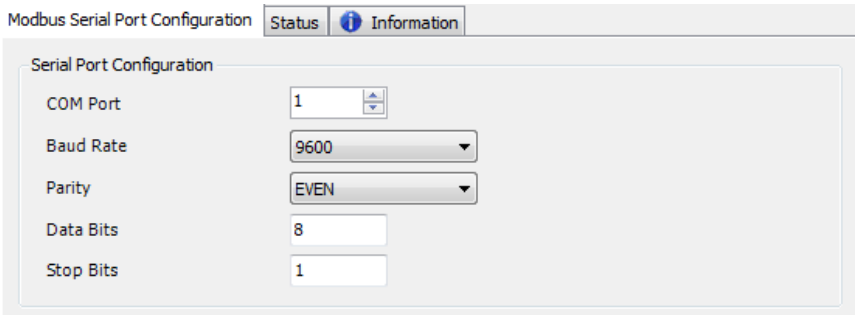


Figure 47

Available parameters are:

COM Port Serial COM Port number (1-4).

Baud Rate Communication Baud Rate.

Parity Communication Parity.

Data Bits Communication Data Bits.

Stop Bits Communication Stop Bits.

The communication mode for the selected serial port (Figure 48) is displayed in the Work area by selecting the Interface Configuration tab after a double click on Device (eTOP500) in the project tree. The Mode of the serial interface parameter is RS232, RS485, RS422.

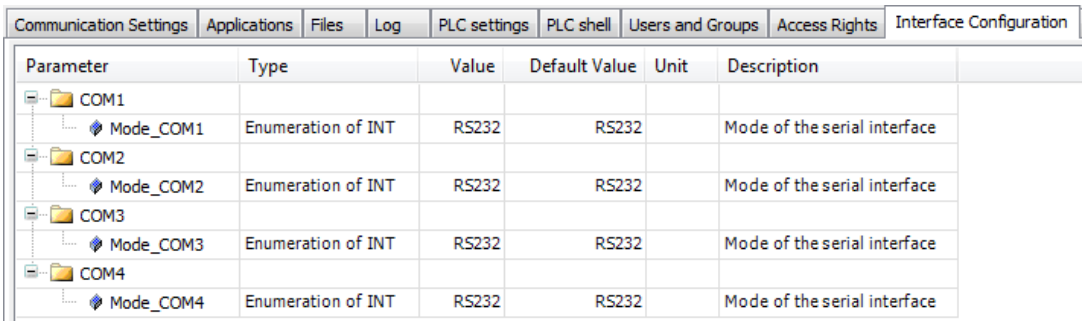


Figure 48

Modbus RTU Master Configuration

Modbus RTU Master configuration (Figure 49) is displayed in the Work area by selecting the Modbus Master Configuration tab after a double click on Modbus COM > Modbus Master COM in the project tree.

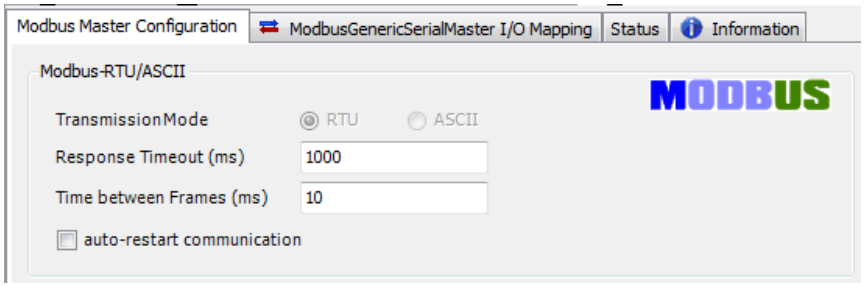


Figure 49

Available parameters are:

Response Timeout (ms) Timeout for Modbus slaves reply, given in milliseconds.

Time between frames (ms) Wait time between Slave reply and next Master query.

Auto-restart communication If set auto-confirm error and re-establish communication.

Add and Configure Remote Modbus RTU Slave Devices

To add a remote Modbus RTU Slave device, right click on Modbus COM > Modbus Master COM in the project tree and select “Add Device”. Modbus RTU Slave devices are located under the category Fieldbuses > Modbus > Modbus Serial Slave (Figure 50), choose the device from the list and click on “Add Device” to add it to the current PLC configuration.

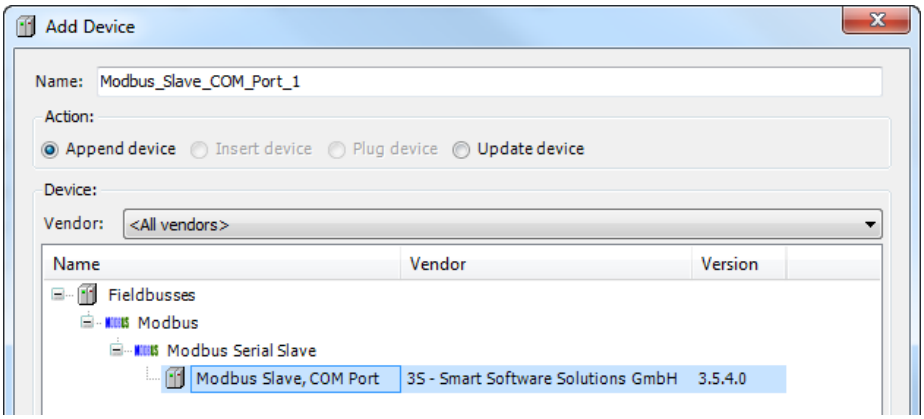


Figure 50

Remote Modbus RTU slave configuration (Figure 51) is displayed in the Work area by selecting the Modbus Slave Configuration tab after a double click on Modbus COM > Modbus Master COM > Modbus Slave COM Port in the project tree.

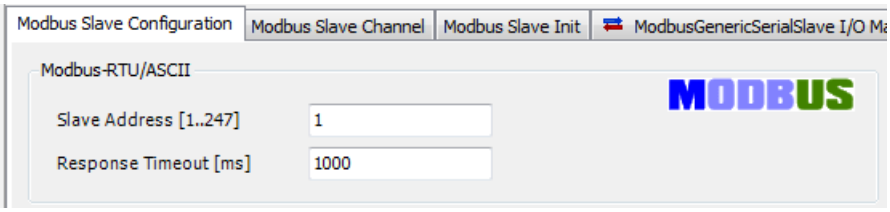


Figure 51

Available parameters are:

Slave Address Modbus Node ID of the Modbus RTU Slave Device

Response Timeout (ms) Timeout for Modbus slaves reply, given in milliseconds.

Configuring Modbus Data Exchange

Modbus data exchange configuration with the Modbus RTU slave can be done in the Work area by selecting the Modbus Slave Channel tab after a double click on Modbus COM > Modbus Master COM > Modbus Slave COM Port in the project tree.
The configuration is based on Channels, for each channel you can configure a Modbus command that will be sent to the Slave. To add a new Channel click on Add Channel button, as shown in Figure 52, the ModbusChannel dialog will be displayed in page, allowing to set-up the channel.

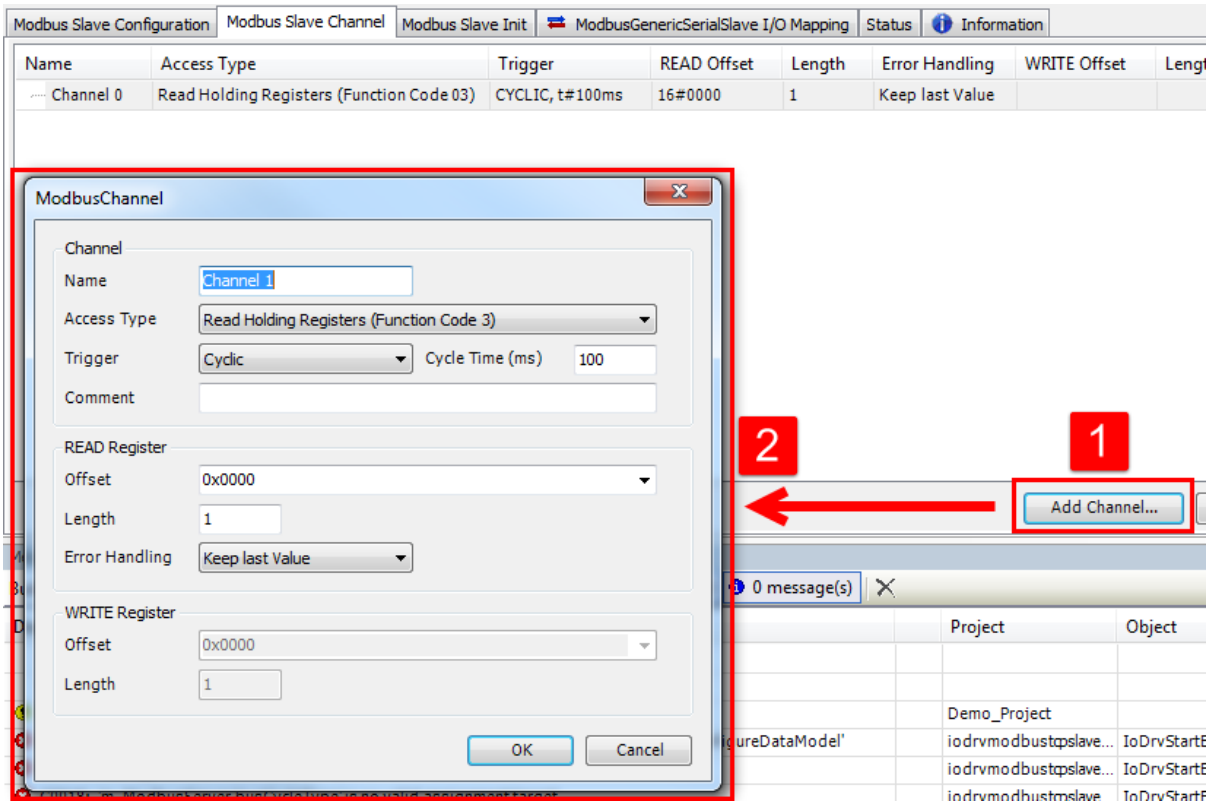
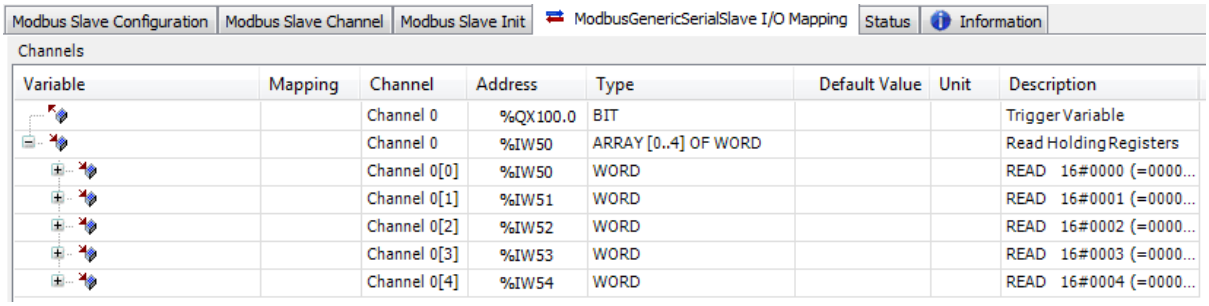


Figure 52

Available parameters are:

Name	Channel Name
Access Type	Selection of the Modbus command.
Trigger	Specifies is the command must be Cyclic, in this case the Cycle Time (ms) must be specified or Rising Edge, in this case the command is launched on the rising edge event of a bit variable defined into the Modbus TCP Slave I/O Mapping.
Comment	User comment if required
Offset	The starting Modbus address
Length	Number of registers to be Read/Write

Mapping configuration of the Modbus RTU Slave (Figure 53) is displayed in the Work area by selecting the Modbus Generic Serial Slave I/O Mapping tab after a double click on Modbus COM > Modbus Master COM > Modbus Slave COM Port in the project tree. Mapping shows a list of all the Modbus Resources read/write in the configured Channels, in case the configured Channel uses a Rising Edge triggered command, the Trigger bit is listed in the Mapping.



Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Channel 0	%QX100.0	BIT			Trigger Variable
		Channel 0	%IW50	ARRAY [0..4] OF WORD			Read Holding Registers
		Channel 0[0]	%IW50	WORD			READ 16#0000 (=0000...
		Channel 0[1]	%IW51	WORD			READ 16#0001 (=0000...
		Channel 0[2]	%IW52	WORD			READ 16#0002 (=0000...
		Channel 0[3]	%IW53	WORD			READ 16#0003 (=0000...
		Channel 0[4]	%IW54	WORD			READ 16#0004 (=0000...

Figure 53

Modbus RTU Slave Configuration

Modbus RTU Slave configuration (Figure 54) is displayed in the Work area by selecting the Modbus Serial Device tab after a double click on Modbus COM > Modbus Serial Device in the project tree.

Note: *When programmed as explained in this chapter the CODESYS V3 PLC will act as a Modbus RTU Slave device. To configure the device for the communication with remote Modbus RTU I/O modules please refer to the chapter Modbus RTU Master configuration.*

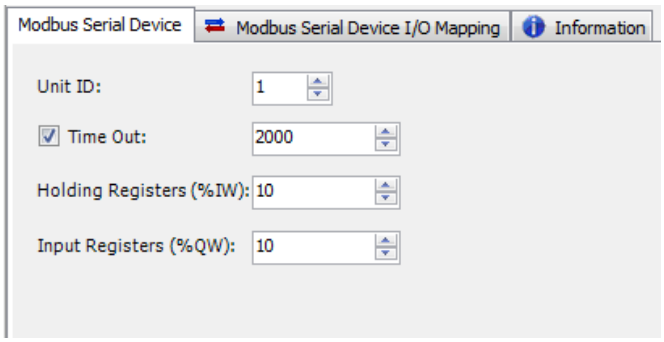


Figure 54

Available parameters are:

Unit-ID	Modbus Node ID of the Modbus TCP Slave Device
Timeout	If selected, timeout for Modbus Master queries, given in milliseconds
Holding Registers (%IW)	Number of holding registers available in PLC memory
Input Registers (%QW)	Number of input registers available in PLC memory

Mapping configuration of the Modbus RTU Slave device (Figure 55) is displayed in the Work area by selecting the Modbus Serial Device I/O Mapping tab after a double click on Modbus COM > Modbus Serial Device in the project tree.

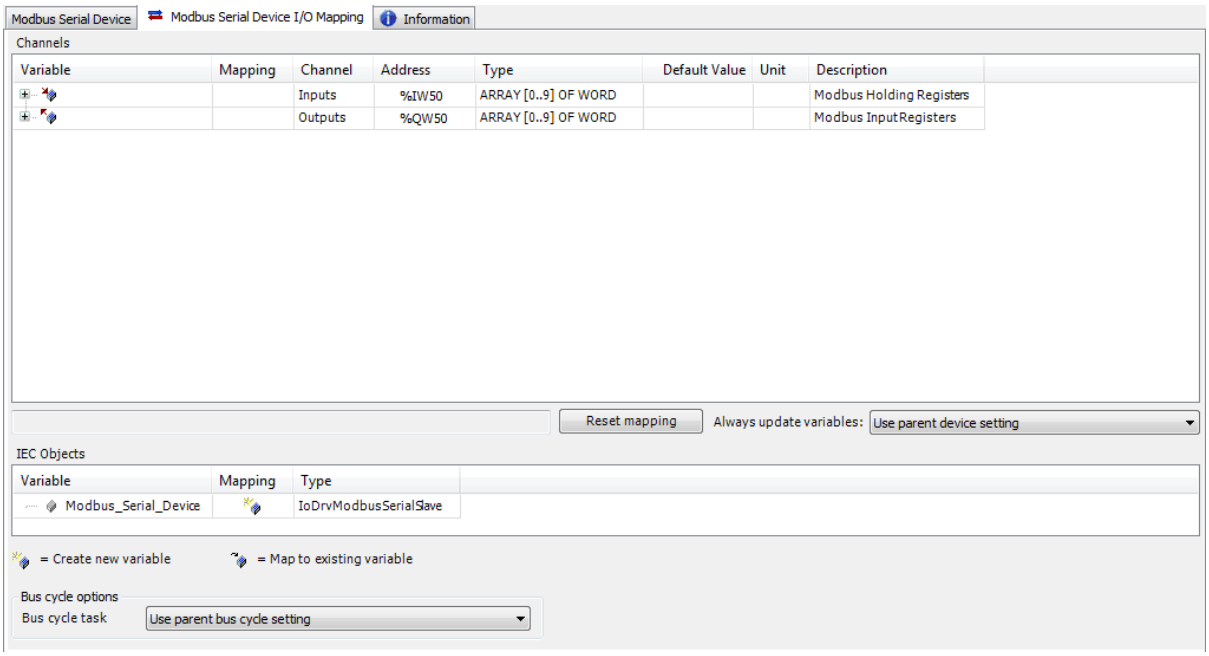


Figure 55

CAN Master

CODESYS V3 runtime can act as a CAN Master to allow the use of distributed CANopen I/O points. To interface the operator panel with CAN network one of the available CAN optional modules must be installed.

Figure 56 shows eTOP504 and eTOP507 with option modules PLCM01 mounted.

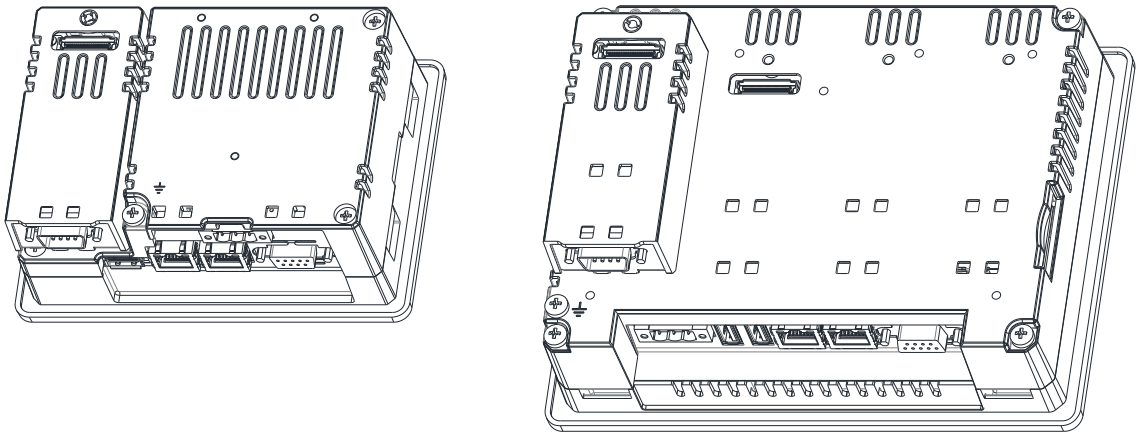


Figure 56

To add a CANopen Master interface two steps are required.
Right click on Device (eTOP500) in the Project tree and select “Add Device”. The Add Device dialog is displayed (Figure 57), the CANbus device is located under the category Fieldbuses > CANbus, choose the CANbus device from the list and click on “Add Device” to add it to the current PLC configuration.

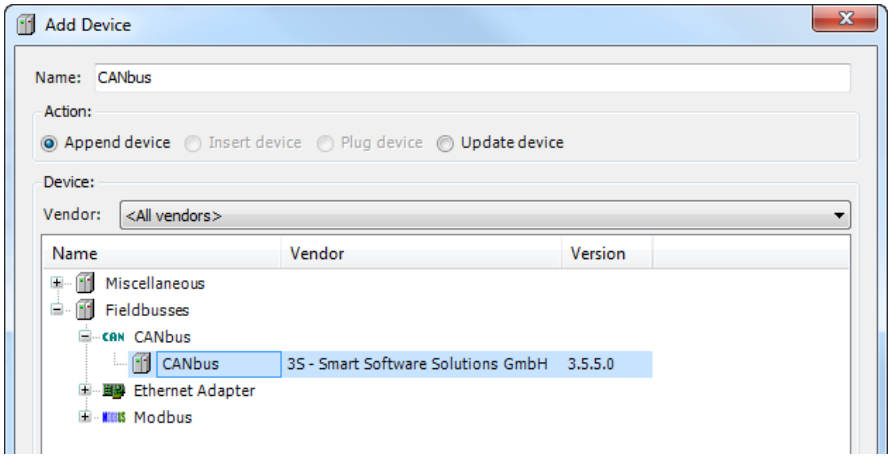


Figure 57

Right click on CANbus in the project tree and select “Add Device”. The CANopen Manager device is located under the category Fieldbuses > CiA CANopen > CiA CANopen Manager (Figure 58), choose the CANopen Manager device from the list and click on “Add Device” to add it to the current PLC configuration.

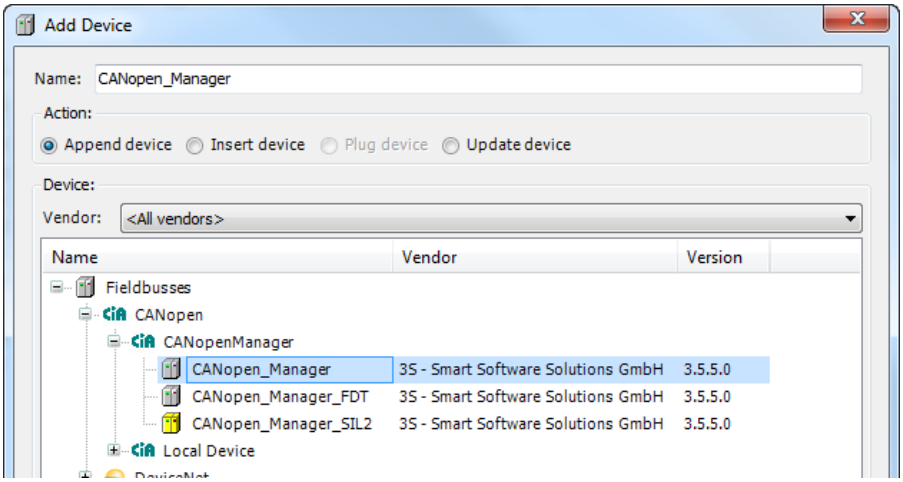


Figure 58

The parameters of the CAN interface are grouped in three tabs accessible on the right part of the PLC Configuration tool when the Can Master element has been added to the configuration tree.

Note: *A complete and detailed description on the configuration of CAN controllers and on the configuration of CAN slave devices is included in the CODESYS User Manual*

CAN Bus Network Configuration

The CAN bus configuration (Figure 59) is displayed in the Work area by selecting the CANbus tab after a double click on CANbus in the project tree.
All operator panel models of type UN30 (1GHz ARM CPU) can support 2 CAN networks.

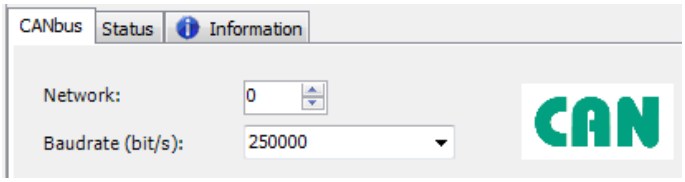


Figure 59

Available parameters are:

Network CAN network number 0 is default, in case of double CAN network configuration Network 1 is used for the second CAN network.

Baudrate (bit/s) CAN network baudrate.

CANopen Master Configuration

The CANopen Master configuration (Figure 60) is displayed in the Work area by selecting the CANopen Manager tab after a double click on CANbus > CANopen Manager in the project tree.

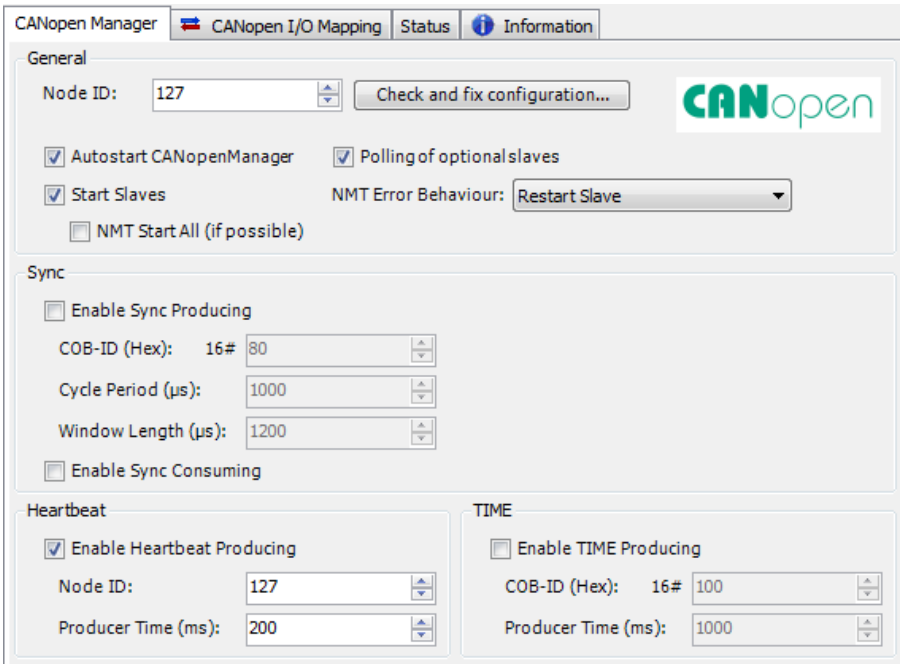


Figure 60

Available parameters are:

Node ID	CAN node number assigned to the CAN master.
Autostart CANopen Manager	If selected, the CANopen Manager starts automatically if all mandatory slaves are ready. If not selected, the manager must be started by the application, using the CiA405 NMT function block for this purpose.
Polling of optional slaves	If an optional slave does not respond during start-up sequence the system polls the slave every second until the slave responds successfully.
Start Slaves	If selected the CAN Master starts the slaves automatically, otherwise the start operation must be done into the application.
NMT Start All	If Start Slaves option is selected it is possible to enable NMT Start All function, this function will start all the slaves at the same time when all the slaves are ready. If not enabled each slave is started separately.
NMT Error Behavior	This option allows to determine the behavior on a guard event, the available options are Restart Slave or Stop Slave
Enable Sync Producing	Enable the sending of Sync telegrams on the CAN bus
COB-ID (Hex)	COB-ID of the Sync message, standard ID is 128 (80 Hex)
Cycle Period (µs)	Time interval between two Sync messages, given in microseconds
Window length (µs)	Length of the time window for synchronous PDOs, given in microseconds
Enable Sync consuming	If selected the Sync messages are supposed to be produced by a different device on the CAN network, the CANopen Manager will receive such messages.
Enable Heartbeat	If selected, the Master sends Heartbeat messages on the CAN network.

producing	
Node ID	CAN Identifier of the Heartbeat messages producer (1-127)
Producer Time (ms)	Time interval between two Heartbeat messages, given in milliseconds.
Enable TIME Producing	If selected the Master sends TIME messages on the CAN network.
COB-ID (Hex)	COB-ID of the TIME messages, default value is 256 (100 Hex)
Producer Time (ms)	Time between two TIME messages, given in milliseconds. Must be a multiple of the task cycle time.

Diagnostic Mapping

When a CAN master device is added to the PLC configuration, the system automatically creates a Variable of type CANOpenManager into the project. This variable contains the diagnostic information about the CAN Master.
The Variable name is assigned by default but can be changed by changing the I/O module name into the PLC configuration. The Variable is reported into the CANopen I/O Mapping tab, displayed in the work area by double clicking on the CANbus > CANopen Manager in the project tree, as shown in Figure 61.

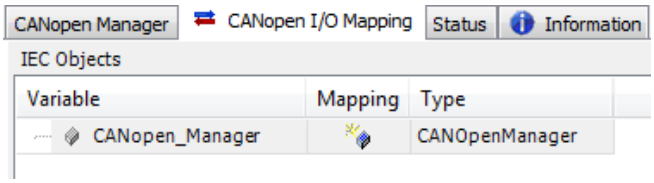


Figure 61

Definition of CAN I/O Slaves in the PLC Configuration.

After the set-up of the CAN Master the structure of available CAN I/O Slaves can be defined.
To add a CANopen I/O Slave, right click on CANopen Manager in the Project tree and select “Add Device”. The Add Device dialog is displayed (Figure 62), the list of available CANopen I/O slaves is located under the Fieldbuses > CiA CANopen > CiA Remote Device category, choose the device from the list and click on “Add Device” to add it to the current PLC configuration.

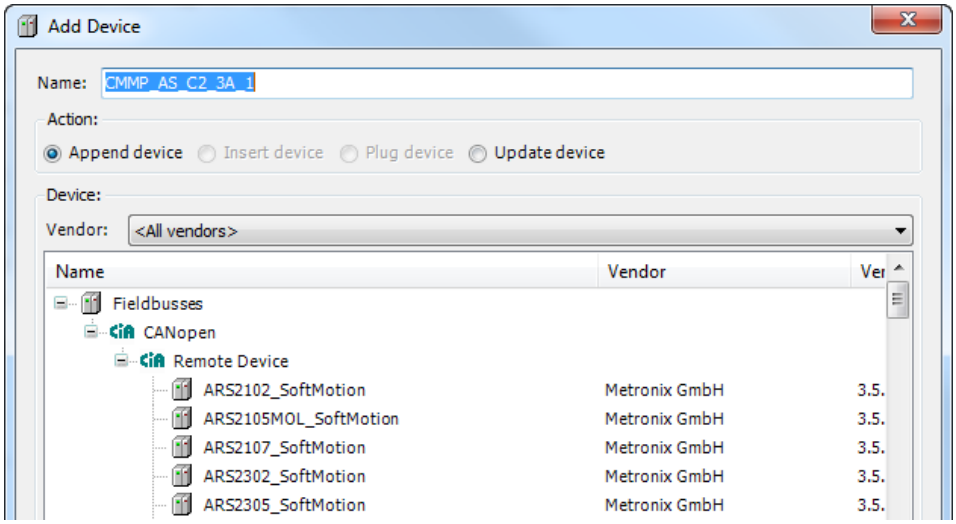


Figure 62

Note: *Installation of third part CAN Slaves into CODESYS V3 requires a specific EDS descriptor file, provided by the CAN Slave manufacturer. Please refer to the CODESYS manual for detailed information regarding the installation of the EDS files.*

Settings for CAN Slaves

The configuration for the CAN slaves has a common part, which is independent from the EDS file. Figure 63 shows the CANopen Remote Device tab of a CAN Slave displayed in the Work area by after a double click on the CANbus > CANopen Manager > CAN SLAVE NAME in the project tree. The parameters shown in Figure 63 can be shown by selecting the Enable Expert Settings option.
.

Note: *For specific information regarding the set-up of the CAN Slaves please refer to the documentation provided from the CAN Slave manufacturer.*

Figure 63

Available parameters are:

Node ID CAN node number of the CAN Slave device (1 – 127).

Create all SDO's When selected the SDO messages for the slave configuration, depending on the PDO mapping are created for all objects.
When not selected the SDO messages for the slave configuration are created only for the modified objects. In this latter case, please make sure the EDS file loaded in CODESYS V3 is matching the hardware device

features, otherwise some required SDO messages will be erroneously skipped.

Enable Sync Producing

If selected this Device send Sync messages on the CAN network. This option is selectable only if the Sync messages production at CANopen Master side is disabled.

No initialization

If selected the sequence of SDO messages required for the device initialization (PDO mapping) will not be created.

Optional device

If selected the current device is considered as optional into the bus.
At start-up the CAN controller will check if it is present applying the following rules:

- If the device is present since start-up and correctly replies to the CANopen mandatory object “Device Type” query (matching the EDS file specification), then it is started. The master will continue with the next device.
- If the device is present since start-up and it does not reply as expected to the “Device type” query, it is not started. The master stops then, reporting a mismatching error in the CAN configuration; if the “Optional device” with not-matching “Device Type” is inserted in the bus after start-up, the master will skip it and continue to scan the other devices.
- If the device is not present since start-up, it is simply skipped. The master will continue with the next device.

Internal Controller Hardware

This chapter describes some implementation-specific issues in the CODESYS V3 runtime developed for use with the Series 500 and 600 HMI products.

The CAN Interface

The PLCM01 CAN option module include a CAN bus interface implemented according to the CAN protocol specifications 2.0 A.
This CAN controller supports only Standard frame format (2.0 A) with bit rates up to 1 Mbit/s.
The following transfer functions have been implemented:

- Transfer rate and timing
- Message framing (Part A)
- Arbitration accordingly to Part A specifications
- Automatic retransmission in case of lost arbitration or error detection
- Acknowledgement
- Message validation
- Error detection and error signaling
- Global Identifier masking (for 11-bit and 29-bit long identifiers)
- Interrupt or data polling driven software supported
- Automatic transfer of data frame (prepared in SDRAM buffer) triggered by one bit setting
- Automatic receive of data packets with the allowed frame identifier
- 32 separated SDRAM memory buffers for data packets having the node corresponding ID
- Fully implemented CAN error fault confinement
- Automatic detection of Bus off state
- Detection of the heavily disturbed CAN bus and warning

Programming the parameter baudRateKbps at the value 0 enables the use of custom timing
The resulting baud rate is calculated using the formula:
Bit frequency = 8 MHz / (Prescaler * (1 + Tsetup + Thold))
Valid values for parameters are:

Prescaler	1 to 64
Tsetup	1 to 8
Thold	1 to 4

Other two parameters can affect the behavior of the CAN controller:
SyncJumpWidth: defines the number of time quanta (8 MHz / Prescaler) allowed to accept a SYNC pulse. Valid values are 1 to 4.
SampleMode: defines the number of times the bit is sampled before is considered valid. Valid values are 0 (1 sample) and 1 (3 samples).

Timer resolution

The resolution of CODESYS V3 timers is 1 millisecond. When a timer value is defined it is internally translated to the corresponding number of milliseconds.
The resolution of the internal Real Time Clock is 1 millisecond allowing the maximum resolution of timers. Note that the execution time of the PLC program may apparently affect the resolution of timers.