EXOR
Tech-note

Control with UniOP - CoDeSys

This manual contains detailed information for using the CoDeSys softlogic system from the company 3S as a software PLC in the HMIcontrol solution for the UniOP family of HMI products.

Sitek S.p.A.
Tn200
Ver. 1.10

Contents

1 Overview

This manual describes the UniOP HMIcontrol system based on the CoDeSys control software.
The documentation covers:
- SCM03-C, SCM05-C, SCM11-C and SCM12-C control modules factory-loaded with CoDeSys run-time,
- UniOP local I/O subsystems
- CANopen distributed I/O interface (for SCM03-C, SCM05-C and SCM12-C only).

This manual is not intended to be a reference for CoDeSys programming. If you need a CoDeSys programming manual please refer to the appropriate documentation.

1.1 UniOP and CoDeSys Control Solutions

At the core of HMIcontrol there are the SCM03-C, SCM05-C, SCM11-C or SCM12-C PLC modules. Control modules based on CoDeSys are compatible with **all** UniOP panels with firmware 5.12 or higher.

SCM03-C	CANopen master interface
SCM05-C	CANopen master interface
SCM11-C	Ethernet interface, supports also basic TCP/IP Ethernet connectivity for UniOP
SCM12-C	Ethernet interface, supports also basic TCP/IP Ethernet connectivity for UniOP and CANopen master interface

The SCM03-C/SCM11-C/SCM12-C modules physical memory configuration is described in Table 1. The table refers to CoDeSys firmware version V4.01K and above.

	Data Memory	**Code Memory**	
SCM03-C	96KB	160KB	
SCM05-C	1MB	1MB	
SCM11-C	1MB	316KB	
SCM12-C	1MB	**HW v1.0**	**HW v1.1**
		316KB	1MB

Table 1

They all have a typical scan time measured for the execution of BOOL instructions written in IL language of 350 $\mu s / K$.

CoDeSys is the complete development environment for the SCM03-C/ SCM05-C SCM11-C/SCM12-C PLC modules. CoDeSys offers the PLC programmer a simple approach to the powerful IEC languages.

The original documentation of CoDeSys will be installed when the software is installed. The available documentation includes a clear and detailed presentation of the software and contains also a useful "First Steps with CoDeSys" that should be your first guide in getting confidence with the system.

The main technical data of the SCM control modules are shown in the table.

CPU	32 bit MIPS RISC processor
Clock speed	24Mhz
Flash memory	1 MB (512 KB reserved)
SDRAM memory	SCM03-C 2MB SCM05-C 8MB SCM11-C 8MB SCM12-C 8MB
CPU supervision (Reset, Watchdog)	Yes
Interface	SCM03-C CAN interface with optical isolation SCM05-C CAN interface with optical isolation SCM11-C Ethernet 10BASE-T SCM12-C Ethernet 10BASE-T and CAN interface with optical isolation

Table 2

1.2 System Configurations

You can apply the HMIcontrol systems in different configurations. The possible configurations are the same as the ones already available for the ISaGRAF-based controllers.

1.2.1 Compact Stand-alone Controller

The HMIcontrol system can be used to build very compact standalone HMI and PLC systems. Input/output is available via the UIM internal modules.

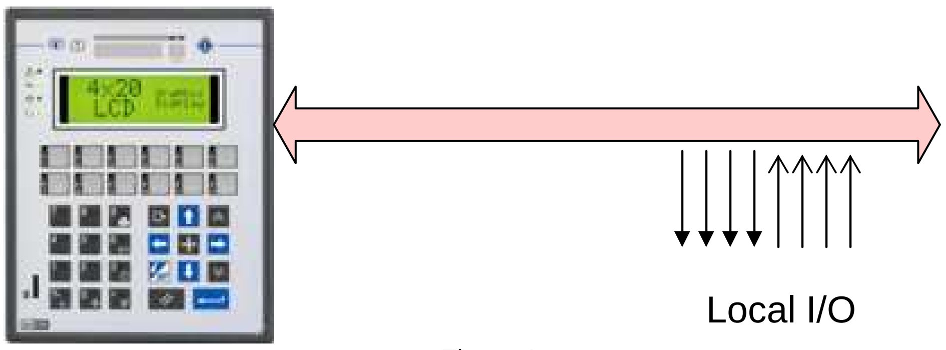


Figure 1

1.2.2 Controller with Distributed I/O

A built-in fieldbus interface is provided with the HMIcontrol modules. Configurations with local and distributed I/O are possible.

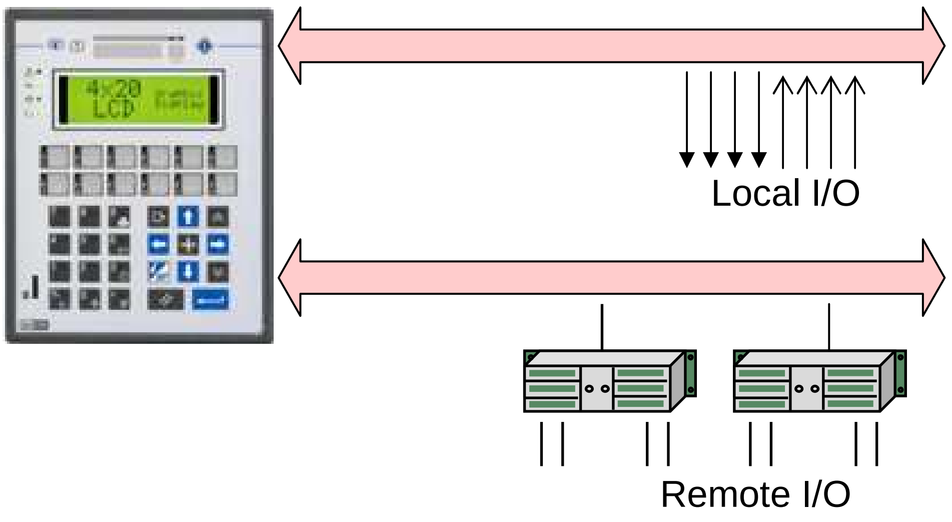


Figure 2

Note: *this configuration is only possible with SCM03-C, SCM05-C and SCM12-C.*

1.2.3 HMIcontrol Connectivity

An HMIcontrol system still offers the same connection capabilities of the UniOP products. Control capability can be combined with connection to a conventional controller (PLC).

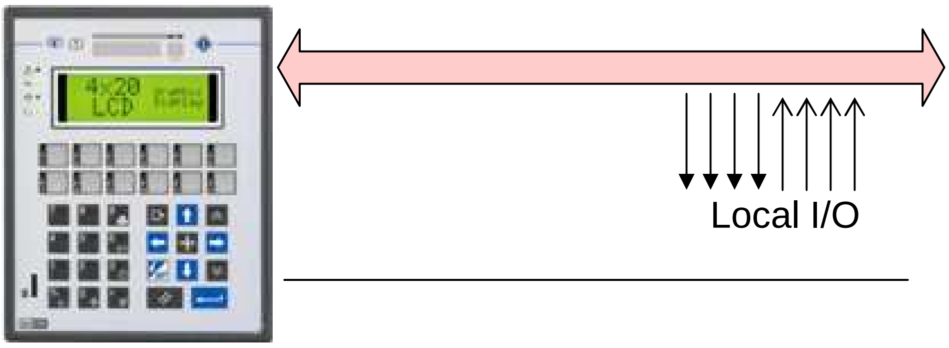


Figure 3

1.3 Requirements and Limitations

The following firmware and software versions are required to work with the CoDeSys HMIcontrol systems:

	Version
UniOP Designer	6.03 or higher
UniOP Firmware	5.12 or higher
CoDeSys	2.3.2 or higher

Table 3

Hardware types –0045 and –0050 have only one slot available for the communication or control module and, if one SCM module is used, it is no longer possible to plug one additional TCM module. Special products equipped with the double module communication board (TCA11 option) are available for special projects.
Use of local I/O is limited to certain UniOP models.
The current implementation of CoDeSys on SCM11-C/SCM12-C does not include drivers for Ethernet distributed I/O.

Note: *CoDeSys Workbench 3 is not supported.*

2 Getting Started

The following chapter describes the basic steps to follow in order to get HMIcontrol running on the SCM03-C/ SCM05-C/SCM11-C/SCM12-C control modules.

2.1 Installing the SCM Control Module

The SCM Control Module must be installed in the socket available for TCM Communication Modules. The Control Module must be installed with the panel powered off. Once the power supply is connected again the UniOP firmware will recognize the new module.

Follow this procedure to install the module in the operator panel:
1. Turn off the operator panel.
2. Using a screwdriver, unscrew (not completely) the two screws “A” fixing the rear cover (shown in Figure 4).
3. Remove the rear cover.
4. Plug the module in the slot with red connectors and make sure the connectors are properly latched.
5. Re-install the rear cover.
6. Fix the screws “A”
7. Stick in the area “B” the label indicating the type of module that has just been plugged in.

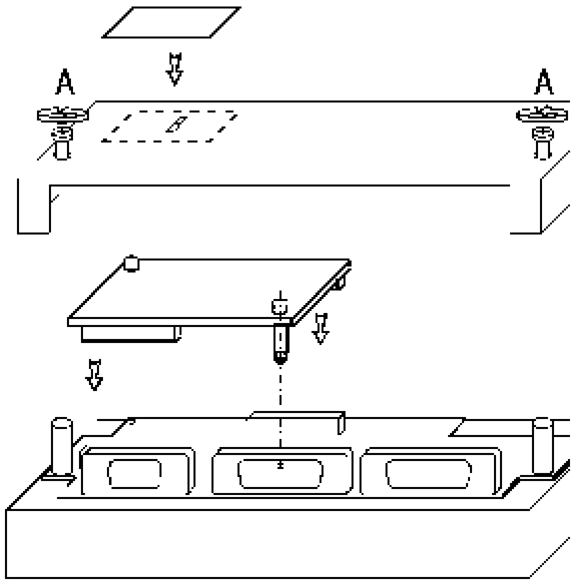


Figure 4

2.2 Control Module Diagnostic

The System Menu of UniOP provides some basic diagnostic information on the operation of the communication and control modules.
To view the diagnostic information:
1. Make sure the operator panel is in Operation Mode
2. Recall the System Menu
3. Scroll down the display to show the bottom row of the page
4. The diagnostic information will be shown as in the example below:

`SCM03 0 H160 X130 OK`	
`SCM03`	type of module
`0 H160 X130`	internal version codes
`OK`	confirms the correct installation of the control module.

5. SCM firmware version can be read directly in system menu; when the SCMxx line is highlighted by the cursor, press the right arrow key to display the version information.

Since essentially the same hardware is used for both ISaGRAF and CoDeSys internal controllers, a proper version numbering scheme has been defined to identify the type of the Internal Controller installed in the module.
The Table 4 shows the rules.

	ISaGRAF	**CoDeSys**	**TCP/IP only**
SCM03	1.xxY	4.xxY	-
SCM05	1.xxY	4.xxY	-
SCM11	2.xxY	4.xxY	1.xxY
SCM12	2.xxY	4.xxY	-

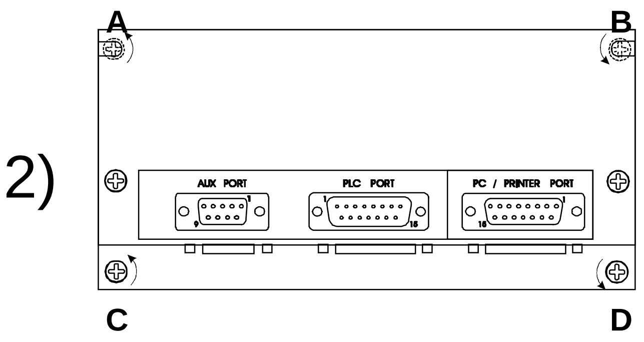
Table 4

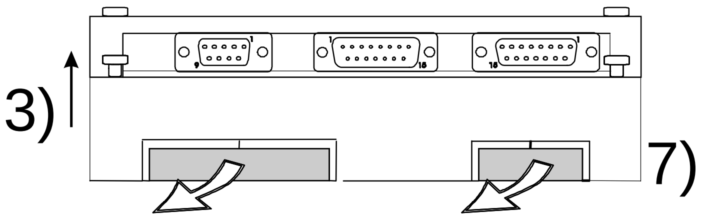
In the above table the “xx” stands for a version number between 01 and 99. The number is normally incremented at each revision of the firmware that includes new features or any major enhancements. “Y” stands for a letter between A and Z. The letter is changed to identify any minor modification in the firmware such as bug fixing.

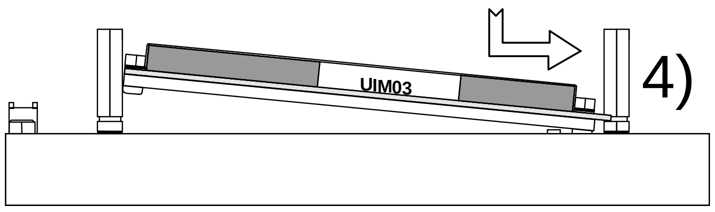
Versions 1.xxY identify the old SCM11 modules with no iPLC support.
SCM11-C and SCM12-C modules always include the TCP/IP software usable for all Ethernet based UniOP protocols.
SCM03-C and SCM05-C modules always include the CAN interface.

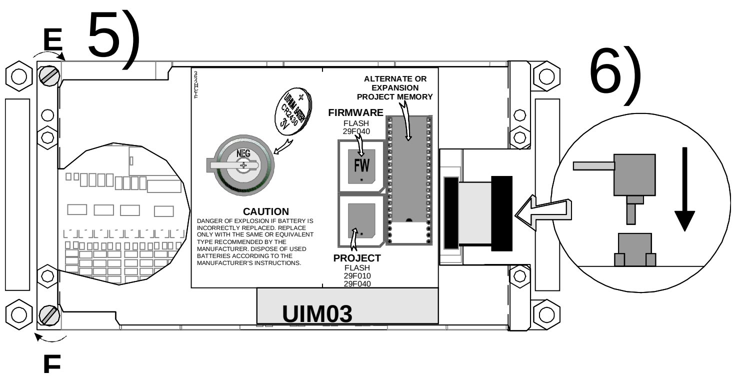
Internal I/O modules can only be installed in operator panels with hardware type –0050.
Follow the procedure described below to install an internal I/O module.

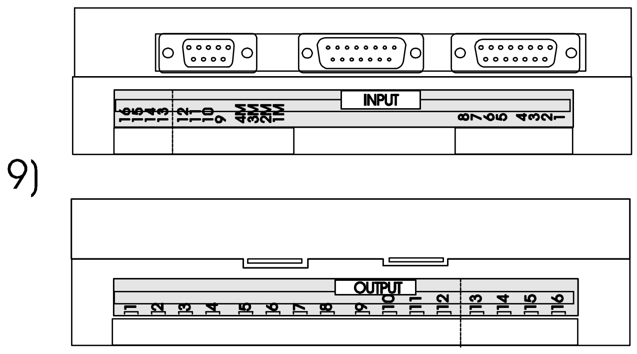
1) Turn off the operator panel and remove all cables.
2) Unscrew (but not remove) with the crosshead screwdriver the four screws A, B, C, and D.
3) Remove the rear cover.
4) Insert the UIM03 module
5) Fix the UIM03 module with the two screws E and F.
6) Plug the UIM03 internal flat cable connector in to the red connector and make sure they are properly latched.
7) Remove the side protection inserts from the rear box.
8) Replace the rear cover, and fix the screws A, B, C, D.
9) Stick the labels indicating the pin assignment.


2)
A
B
C
D
AUX PORT
PLC PORT
PC / PRINTER PORT


3)
7)


4)
UIM03


5)
6)
E
F
ALTERNATE OR EXPANSION PROJECT MEMORY
FIRMWARE
FLASH 29F040
FW
NEG
CAUTION
DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. REPLACE ONLY WITH THE SAME OR EQUIVALENT TYPE RECOMMENDED BY THE MANUFACTURER. DISPOSE OF USED BATTERIES ACCORDING TO THE MANUFACTURER'S INSTRUCTIONS.
PROJECT
FLASH 29F010 29F040
UIM03


9)
INPUT
OUTPUT

2.3 I/O Module Diagnostic

The System Menu of UniOP provides some basic diagnostic information on the operation of the internal I/O modules.
To view the diagnostic information:

1. Make sure the operator panel is in Operation Mode
2. Recall the System Menu
3. Scroll down the display to show the bottom row of the page
4. The diagnostic information will be shown as in the example below:

`UIM03 0 H160 X130 OK`

`UIM03`	type of module
`0 H160 X130`	internal version codes
`OK`	confirms the correct installation of the control module.

2.4 Installing the CoDeSys Programming System

The CoDeSys programming software can be downloaded for free from the 3S web site at the following address:

http://www.3s-software.com/index.shtml/download

The download page is accessible only after a free registration that will be processed by 3S within a couple of days but normally faster.

A special ‘target module’ is required to allow the standard CoDeSys package to program systems based on SCM03-C, SCM05-C, SCM11-C or SCM12-C.
The UniOP Target-Package-Support required to support the SCM03-C, SCM05-C, SCM11-C and SCM12-C hardware is available for download from the www.uniop.com web site in the Document area under the CoDeSys folder.
The Target support is provided as a zip file that can be un-zipped in any directory respecting the internal folder structure.
The CoDeSys installation package includes a tool for Target installation.
The tool is called “Install Target” and can be used both for checking the installed target support packages and for installing new ones. To install the SCM Target Support Package just click “Open” and locate in the hard disk the file with extension “.tnf” that was supplied with the zip package. The target tool does the rest of the installation automatically.
The Install Target tool can be found in the CoDeSys program group that you will find in the Start Menu after the installation of the software.

The Target configuration for SCM03-C modules is called:
EXOR/SITEK CoDeSys for SCM03 (eCos/Mips)

The Target configuration for SCM11-C, SCM05-C and SCM12-C modules is called:
EXOR/SITEK CoDeSys for SCM1x and SCM05 (eCos/Mips)

The Install Target tool is shown in Figure 5.

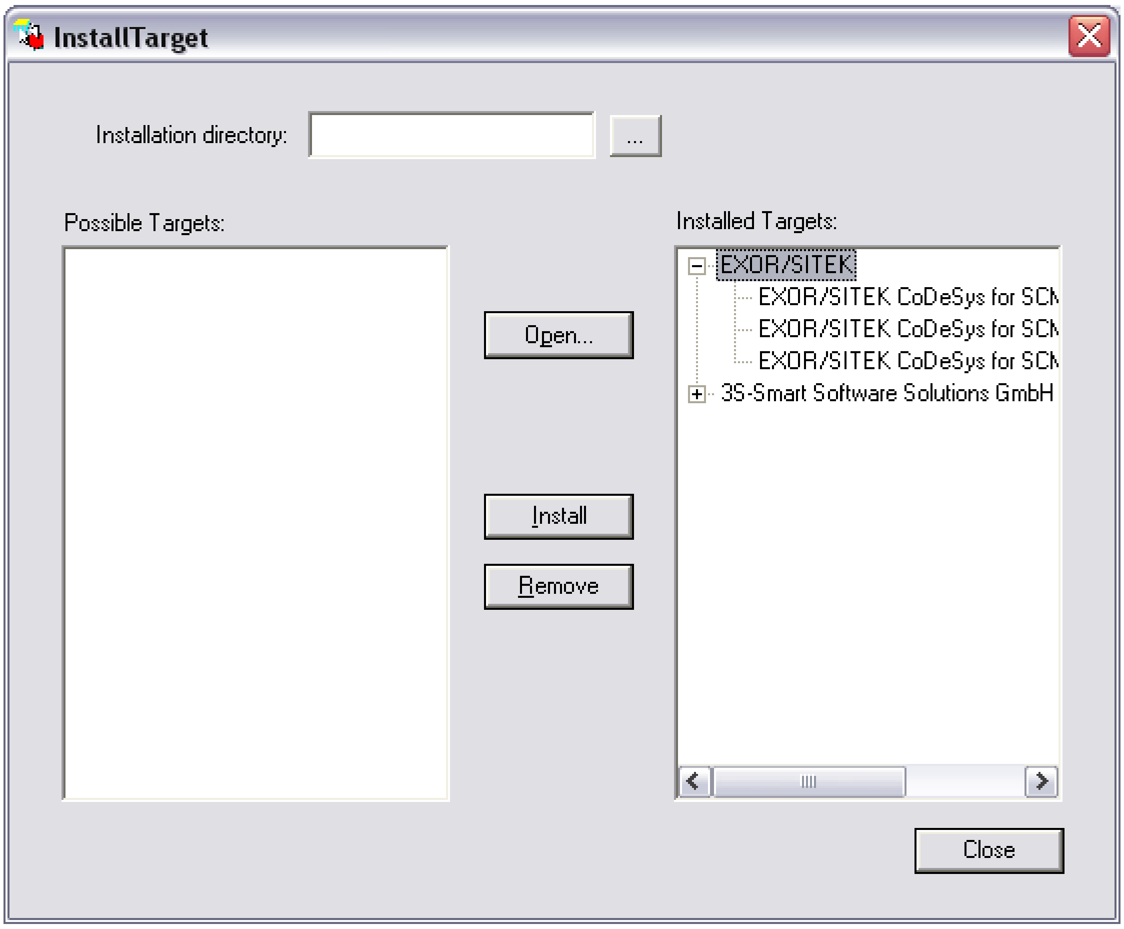

InstallTarget
Installation directory:
...
Possible Targets:
Installed Targets:
EXOR/SITEK
EXOR/SITEK CoDeSys for SCM
EXOR/SITEK CoDeSys for SCM
EXOR/SITEK CoDeSys for SCM
3S-Smart Software Solutions GmbH
Open...
Install
Remove
Close

Figure 5

2.4.1 Updating old Target Support Packages

Target Support Package should be always aligned with the CoDeSys firmware version installed on the SCM module.

SCM modules equipped with CoDeSys firmware version V4.01F requires target support package version V1.02 or greater.

Any new version of the CoDeSys “Target Support Package” can be installed over the existing one; this would keep the support for the older modules, still requiring old versions of the support package.
Start the “Install target” tool from the 3S CoDeSys program group, click “Open”, select the .TNF file from the “Target Support Package” folder, click on “EXOR/SITEK” and click “Install”.

Note: *The first official and releasable CoDeSys firmware for SCM03-C and SCM11-C modules is the V4.01F. The first official and releasable CoDeSys firmware for SCM12-C modules is the V4.01M. The first official and releasable CoDeSys firmware for SCM05-C modules is the V4.02P.*

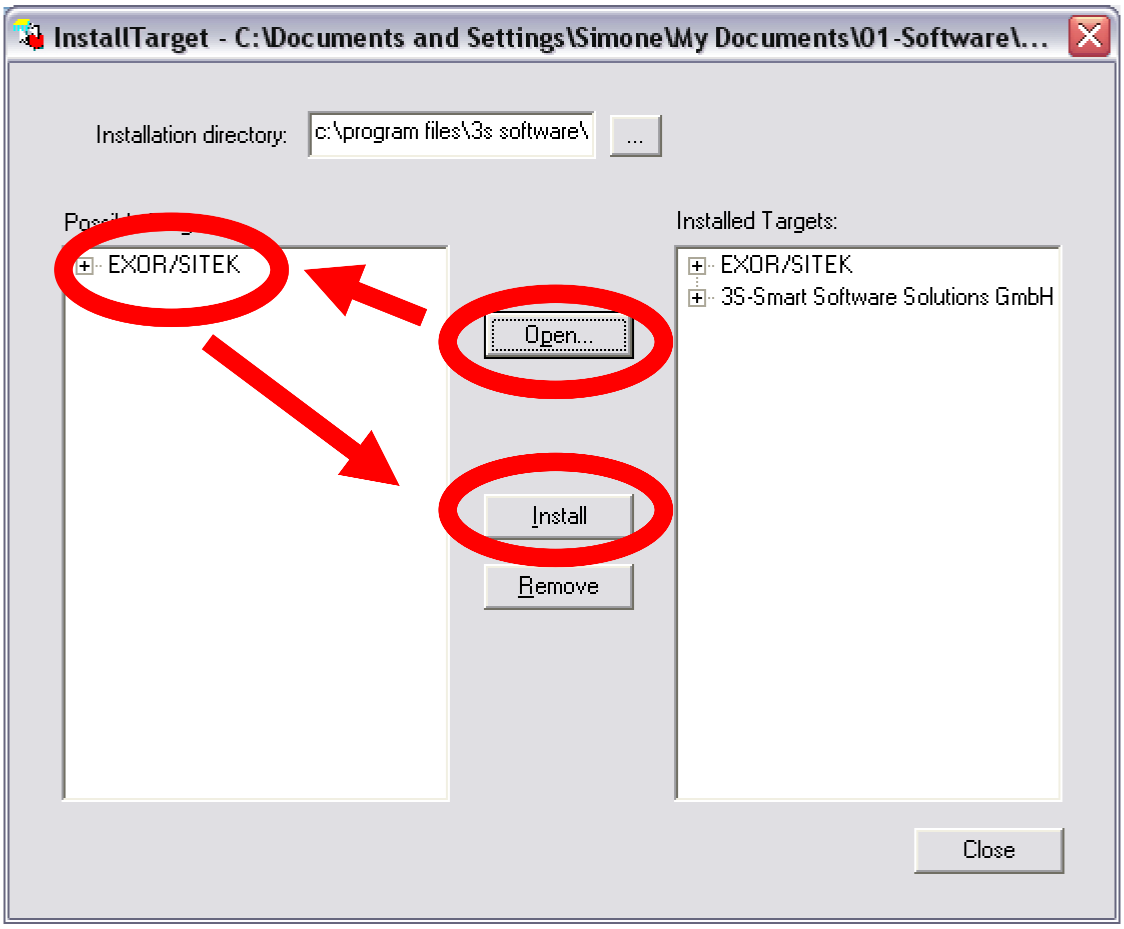

InstallTarget - C:\Documents and Settings\Simone\My Documents\01-Software\...
Installation directory:
c:\program files\3s software\
...
Possible Targets:
EXOR/SITEK
Open...
Install
Remove
Installed Targets:
EXOR/SITEK
3S-Smart Software Solutions GmbH
Close

Figure 6

Depending on the presence of old support packages, the software will ask to overwrite existing settings; please confirm to overwrite them.
After the new Support Package has been installed the conversion of existing projects may require some steps depending on the level of compatibility of the new Support Package in compare with the old one. The conversion is done automatically by the CoDeSys programming software when opening the old application. Any update of the “Target Support Package” will come together with proper instruction for existing projects conversion, if needed.

2.5 Setting-up the Communication

The HMIcontrol system is composed by two subsystems, the UniOP Operator Panel and the SCM Internal Controller Module.
Programming both subsystems will be via the same serial communication link.
The SCM11-C and SCM12-C modules offer, as a more efficient alternative, an Ethernet interface that can be used for:
- Programming the internal PLC,
- communication with an external controller using a UniOP Ethernet communication protocol and finally
- programming UniOP.

The SCM05-C modules, when installed on FW70 units using firmware version v5.90A or higher, offer the possibility to be programmed using the panel integrated Ethernet interface.

Note: *SCM05-C supports programming via onboard Ethernet interface on FW70 units with firmware version v5.90A and above ONLY when panel is in Operation Mode.*

The set-up of the CoDeSys communication is described in the following chapters.

2.5.1 Setting-up the Port for the CoDeSys Programming Software

Select "Online\Communication Parameters" in the CoDeSys programming software. The "Communication Parameters" dialog will appear as shown in Figure 7.
The first time this dialog is opened, the user will be requested to specify the channel for the connection with the PLC. Connection channels can be created with the "New…" button.

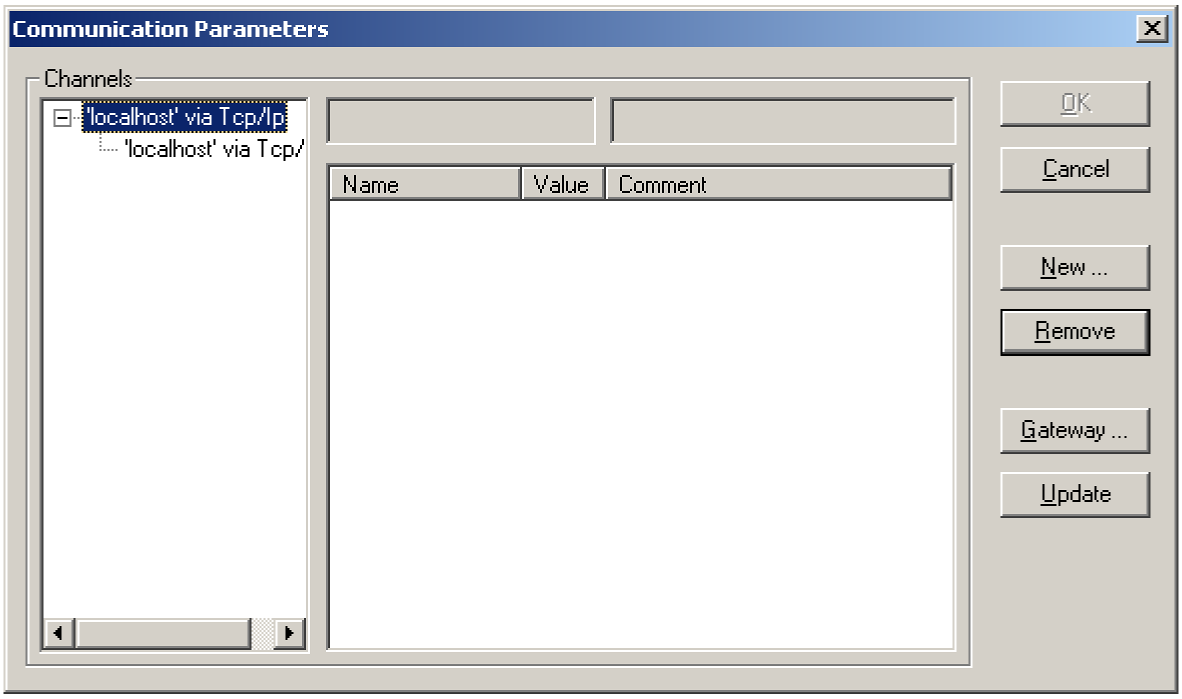
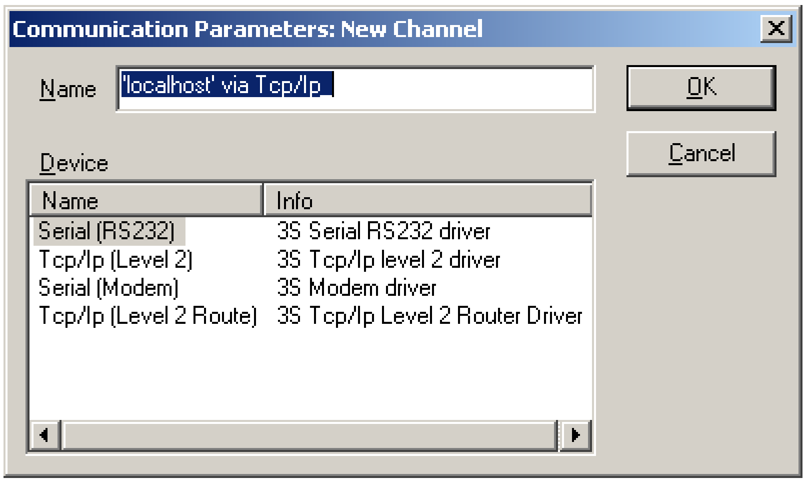


Figure 7

When defining a new channel, the type and all the relevant parameters can be defined in the dialog box shown in Figure 8.

The following options are possible:

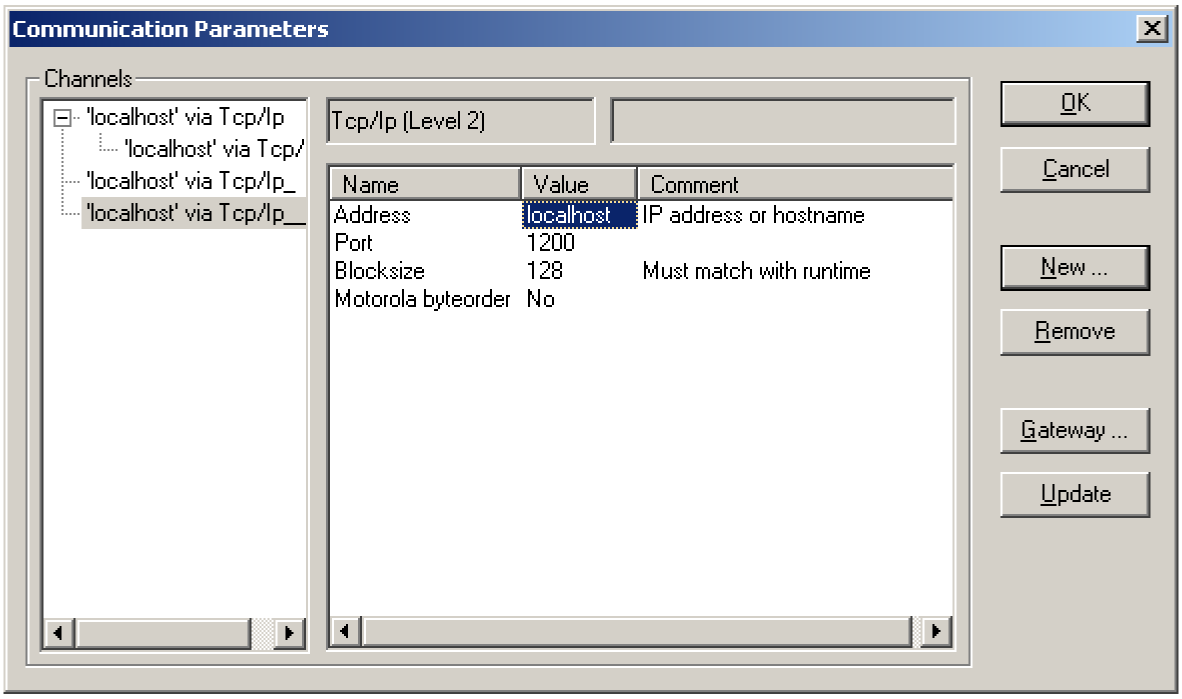
SCM03-C	Serial (RS232)
SCM05-C	Serial (RS232)
	TCP/IP (Level 2) (refer to chapter 2.5)
SCM11-C	Serial (RS232)
SCM12-C	TCP/IP (Level 2)



Name	Info
Serial (RS232)	3S Serial RS232 driver
Tcp/Ip (Level 2)	3S Tcp/Ip level 2 driver
Serial (Modem)	3S Modem driver
Tcp/Ip (Level 2 Route)	3S Tcp/Ip Level 2 Router Driver

Figure 8

Default parameters for both serial port connection and Ethernet connection are correct for operation with UniOP.
When defining the driver for Ethernet connection, the “Address” parameter shown in Figure 9 must contain the IP address assigned to the SCM05-C/SCM11-C/SCM12-C module.



Name	Value	Comment
Address	localhost	IP address or hostname
Port	1200	
Blocksize	128	Must match with runtime
Motorola byteorder	No	

Figure 9

2.5.1.1 Setting the IP address in SCM05-C/SCM11-C/SCM12-C Control Modules

Ethernet programming for SCM11-C/SCM12-C modules is possible only if the UniOP panel is equipped with firmware version V5.20 or greater. Ethernet programming for SCM05-C is possible only with UniOP panels firmware type FW70 and firmware version V5.90A or above.
Setting the IP address of the module follows the standard rules for UniOP IP address setup.

2.5.2 Communicating with the Internal Controller

When UniOP does not contain a valid project, it stays in Configuration Mode; in this situation the PLC port is assigned by default to the Internal Controller.
To program the Internal Controller when no project is loaded in the panel, use the normal UniOP programming cable connected to the PLC port of the panel.
If the operator panel has no PC/Printer Port, then the PC/PLC Port will be used also for communication with the Designer software. The Internal Controller will be programmable only after a valid project has been transferred and the panel is in Operation Mode.
If the SCM05-C/SCM11-C/SCM12-C module is used, then the Ethernet port is always available for communication with the PLC run-time system (for SCM05-C please refer to chapter 2.5.1.1)
When a valid project is present in UniOP and the panel is in Operation Mode, the System Menu will allow the user selecting the mode of communication for the Internal Controller using the following procedure:

- recall the System Menu in the panel,
- use the Up/Down arrow keys to scroll the menu lines until PC/Printer or PLC are highlighted,
- use the Left/Right arrow keys to change the assignment of the selected port.

Two options related to the Internal Controller are available for each UniOP port:

Application Mode	ports should always be assigned to the Internal Controller in Application Mode if it has to be used for normal operations such as application downloading and debugging
Service Mode	Service Mode is reserved for special Internal Controller maintenance and should not be used

The Port is assigned to the Internal Controller in Application Mode when the corresponding row of the System Menu displays the text “**Application**”. This message is reduced to “**A**” for displays with 20 characters per row.
When the PC/Printer port is not assigned to the Internal Controller, it reports the printer status as usual.
When the PLC port is not assigned to Internal Controller and the Designer project does not use an external controller, the System Menu contains the string “**NOT IN USE**” in the PLC row. In case an external controller is used, the PLC row reports the communication error code as usual.

Note *any modification to the port assignment done in System Menu becomes effective after you exit the menu.*

Communication with the Internal Controller is possible both when the panel is in Configuration Mode and when it is in Operation Mode.
Standard UniOP programming cables CA2 or CA114 can be used to connect the CoDeSys software UniOP. A gender changer may be required to connect to the UniOP PLC port.

2.5.2.1 Limitations

There are some limitations in the configurations available for programming the Internal Controller. This chapter provides an overview.

1. If UniOP contains a valid project that uses the PLC port to communicate with an external controller and it is in Operation Mode, then communication with the Internal Controller is not allowed through the PLC Port, because it is already assigned to the PLC communication.

2. If UniOP contains a valid project configured to work with Remote Passthrough, the communication with the Internal Controller through the PC/Printer Port is not allowed. The PC/Printer port is already assigned to wait for incoming commands for the Remote Passthrough operation.

3. If UniOP contains a valid project configured to use the UniNet network and the PC/Printer port is assigned to network communication, the same UniOP port cannot be used to communicate with the UniOP Internal Controller. A similar consideration applies in case the PLC port is used as network port: communication with the PLC is not allowed through the same port.

4. If UniOP contains a valid project where the external controller is configured with a protocol that requires a TCM module, then the Internal Controller may not work properly. Operations with external controllers that require Ethernet interface via SCM11-C/SCM12-C are instead always allowed.

The table below summarizes the most common cases in the connection with the Internal Controller.

UniOP Mode	Communication Ports	Selection in System Menu	Connect Internal Controller to
Configuration	PC/Printer+PLC	-	PLC Port
	PC/PLC	-	Programmable only in Operation Mode
Operation	PC/Printer+PLC	PLC: Application	PLC Port
	PC/Printer+PLC	Printer: Application	PC/Printer Port
	PC/PLC	Printer: Application	PC/PLC Port

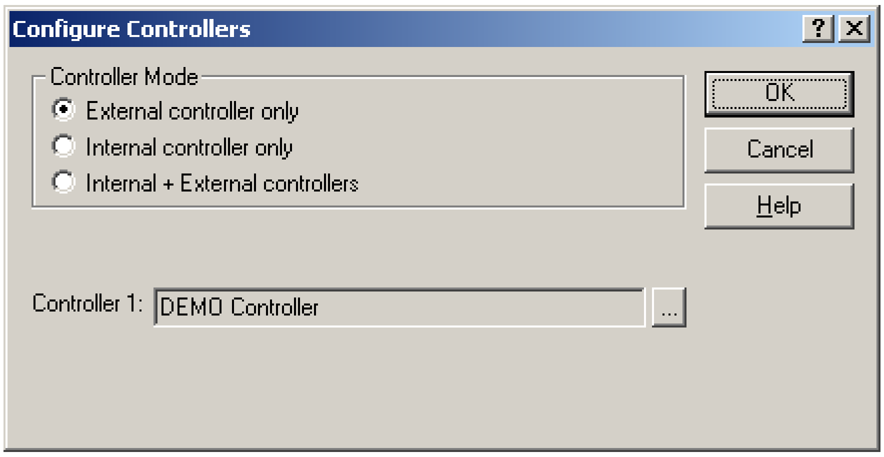
Table 5

3 HMI Programming

UniOP Designer software version 6.03 or higher is required to program the HMI panel equipped with the SCM03-C/SCM05-C /SCM11-C/SCM12-C modules when the use of the Internal Controller is required.

3.1 Enabling the Internal Controller CoDeSys

UniOP supports three different modes of operation of the Internal Controller.
The mode of operation can be selected in the dialog box "Project\Configure Controllers" as shown in Figure 10.



Controller Mode
- [✓ External controller only]
- Internal controller only
- Internal + External controllers

Controller 1: DEMO Controller

Figure 10

3.1.1 No Internal Controller

If you select "External Controller Only" from the "Configure Controllers" dialog box, the UniOP HMI panels will not activate the internal controller. UniOP will communicate to an external controller using the communication driver selected from the list of drivers displayed when you click on the reference button.
This option can be chosen also when an SCM03-C/ SCM05-C/SCM11-C/SCM12-C Control Module is present in the system. In this case the Control Module will not be activated.

3.1.2 Stand Alone Operation

Selecting "Internal Controller Only" from the "Configure Controllers" dialog box will activate the Internal Controller in the HMIcontrol system. UniOP will not communicate to an external controller via the PLC Port. HMI projects will only reference the Internal Controller.

Note *If the Designer project is configured to use the Internal Controller and SCM03-C / SCM05-C / SCM11-C / SCM12-C module is not installed on the panel, the project will not run properly. Additionally, using a TCMxx module with a project configured to use the Internal Controller may result in an unexpected behavior.*

3.1.3 Operation with External Controller

Operation “Internal + External Controller” is the most general configuration. UniOP is connected to an external controller via the PLC port or Ethernet port and the Internal Controller is also activated.
The Internal Controller will work independently of the communication with the external controller.
When operation with external controller has been selected, the Designer will always present the Source Selection dialog box when entering communication references for all dynamic data items. The Source Selection dialog box lets the user choose the location of the requested data.

3.1.4 Selecting the Internal Controller Type

After selecting one of the modes including Internal Controller in the “Configure Controllers” dialog box, Designer will prompt you for the Internal Controller type selection showing the dialog box of Figure 11.

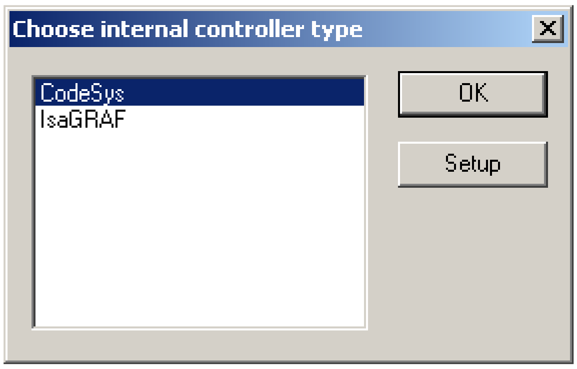


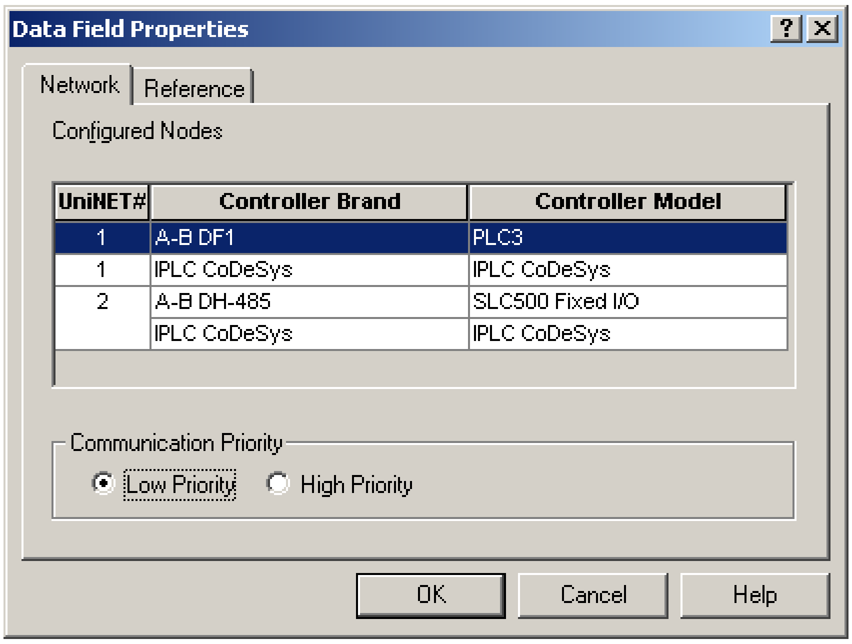
Figure 11

Note *Use SCM03, SCM05 and SCM11 for ISaGRAF. Use SCM03-C, SCM05-C, SCM11-C or SCM12-C for CoDeSys.*

3.1.5 UniNet and the Internal Controller

Internal Controllers are compatible with the UniNet network. The data of the Internal Controller running in any operator panel configured as UniNet server is available to all clients in the network.

Internal Controllers appear in the list of available data sources in the Source Selection dialog box. See figure below.


Data Field Properties

Network | Reference

Configured Nodes

UniNET#	Controller Brand	Controller Model
1	A-B DF1	PLC3
1	IPLC CoDeSys	IPLC CoDeSys
2	A-B DH-485	SLC500 Fixed I/O
	IPLC CoDeSys	IPLC CoDeSys

Communication Priority

[✓ Low Priority] High Priority

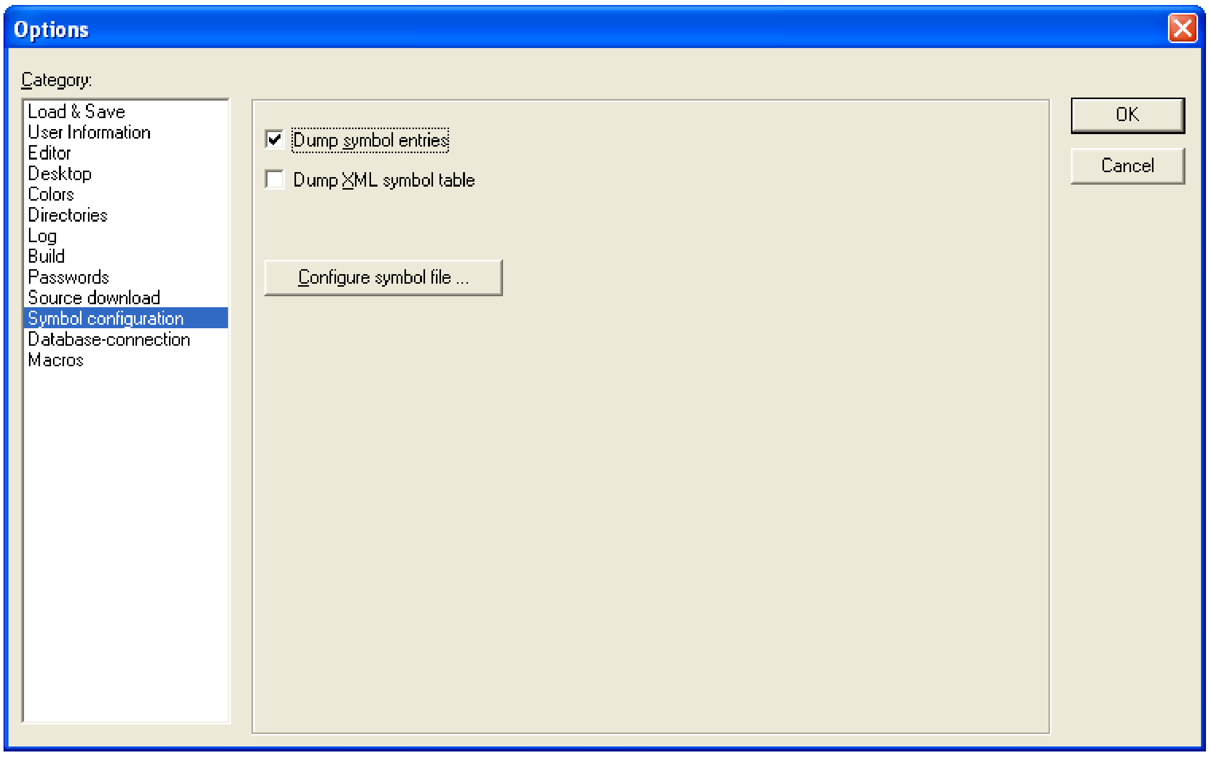
Figure 12

3.2 The Tag Editor

The Designer Tag Editor supports direct import of the tag file generated by the CoDeSys programming software.

3.2.1 Importing a Tag File

The CoDeSys programming software saves a list of all the names used into the PLC program in a file with extension “.sym”. This file is stored in the application folder.
The software creates the “.sym” file only if the option “Dump symbol entries” is selected in the CoDeSys Option, under “Symbol Configuration” as shown in Figure 13.
You may need eventually to check the configuration of the symbol file in order to be sure that symbols are created for all variables in all POUs. Please refer to CoDeSys documentation for additional information.



Options

Category:

- Load & Save
- User Information
- Editor
- Desktop
- Colors
- Directories
- Log
- Build
- Passwords
- Source download
- Symbol configuration
- Database-connection
- Macros

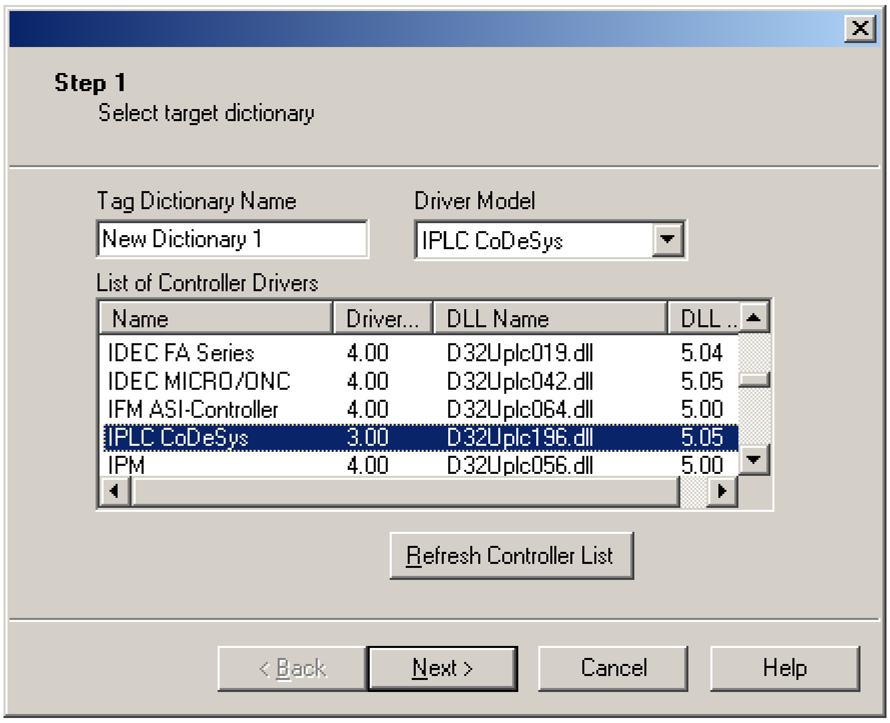
[✓] Dump symbol entries

[ ] Dump XML symbol table

Figure 13

A new version of the ".sym" file is created each time the project is built. Symbol files should be re-imported in Designer Tag Editor to update the Designer's tag list every time they are updated.

The CoDeSys symbol file can be imported in Designer selecting the "Import tags" command from the "File" menu of the Tag Editor. The first step of the import process is shown in Figure 14; in the list of the available controllers the CoDeSys is listed as "iPLC CoDeSys".



Step 1
Select target dictionary

Tag Dictionary Name: New Dictionary 1

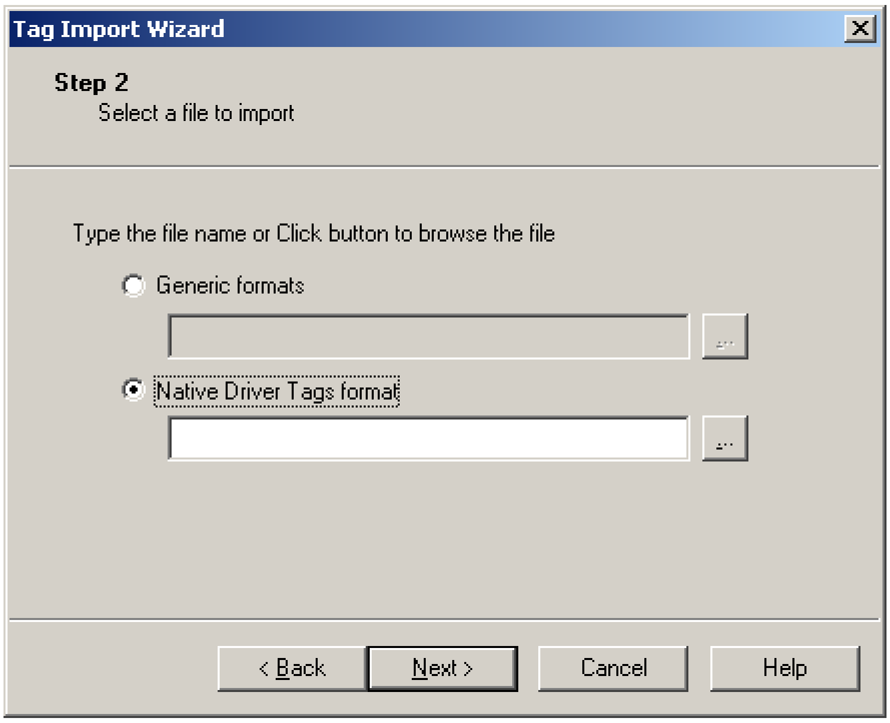
Driver Model: IPLC CoDeSys

List of Controller Drivers

Name	Driver...	DLL Name	DLL ..
IDEC FA Series	4.00	D32Uplc019.dll	5.04
IDEC MICRO/ONC	4.00	D32Uplc042.dll	5.05
IFM ASI-Controller	4.00	D32Uplc064.dll	5.00
IPLC CoDeSys	3.00	D32Uplc196.dll	5.05
IPM	4.00	D32Uplc056.dll	5.00

Figure 14

The second step is shown in Figure 15.



Type the file name or Click button to browse the file

( ) Generic formats

(●) Native Driver Tags format

Figure 15

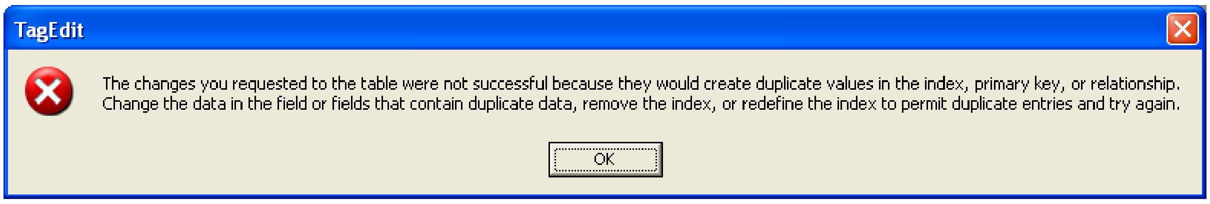
Support for CoDeSys native tag format is provided selecting the “Native Driver Tags format” radio button.

Any new set of tags imported in Tag Editor after the first one will be imported as a new Dictionary. The new Dictionary should be properly linked to the Designer project when enabling the Tag Support under “Project\Configure Tag Dictionaries…”.

Note: *New Tags can be ONLY created starting from CoDeSys programming software. The Designer tag database must not be changed from within Designer.*

Note: When the “**Clean all**” command is executed in the CoDeSys programming software, all the absolute tag addresses are re-calculated by the CoDeSys compiler and the tag file needs to be imported again in the Designer Tag Editor. Failure to do so will result in errors accessing the PLC data

Note: Duplicate Entry in *.SYM file is not supported, Tag Editor will warning you (Figure 16)



The changes you requested to the table were not successful because they would create duplicate values in the index, primary key, or relationship. Change the data in the field or fields that contain duplicate data, remove the index, or redefine the index to permit duplicate entries and try again.

Figure 16

Check *.SYM file if contains duplicate Tag names and chapter.3.2.2.

3.2.2 Tags configuration for UIM05/UIM06

The UIM05 and UIM06 boards support Global Variables to configure I/O channels. Selecting “Export variables object” for “Global Variables” and highlighting “PLC configuration” (Figure 17) may result in duplicate entries in *.SYM file (Figure 18) for the UIM05 and UIM06 Analog points.

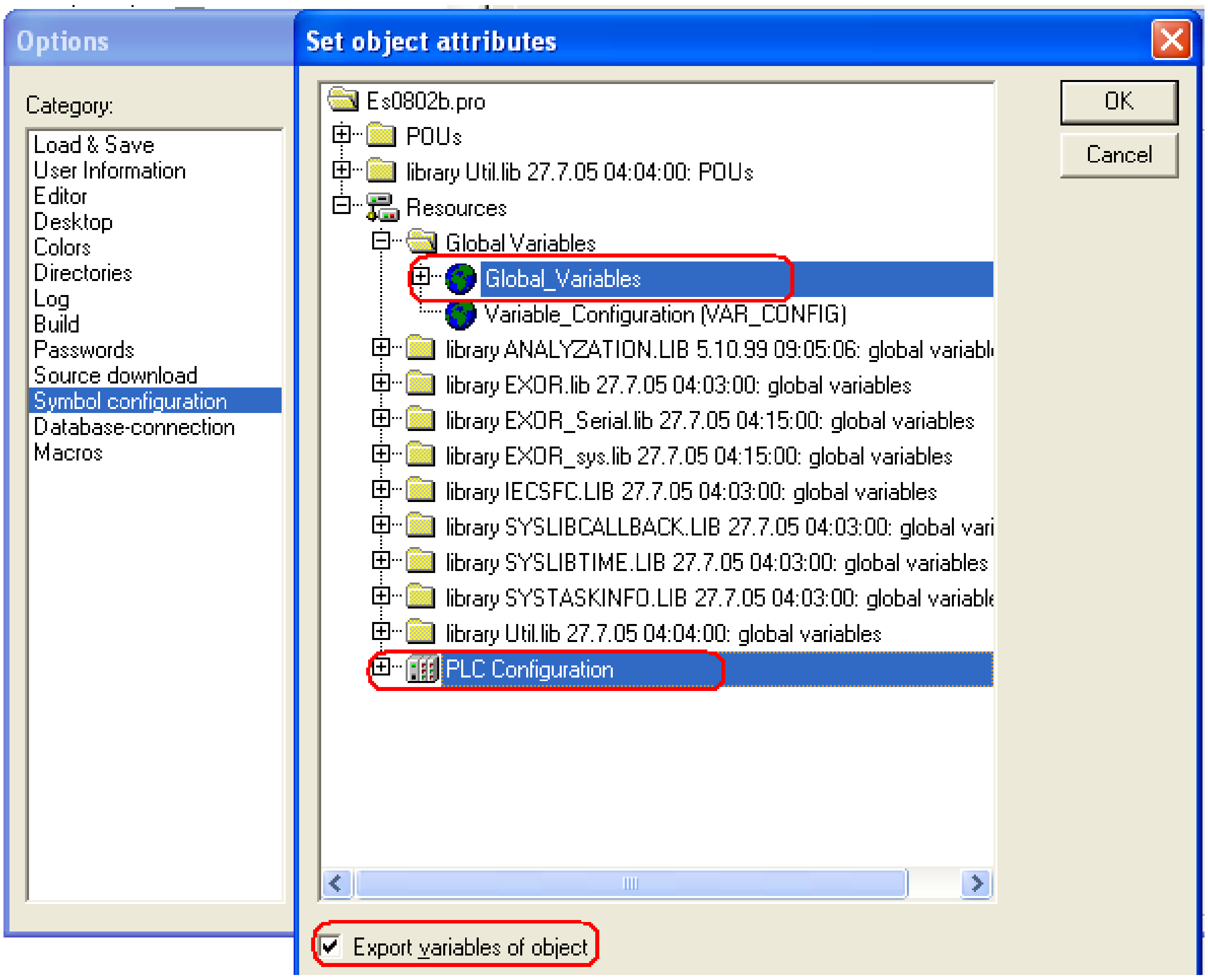
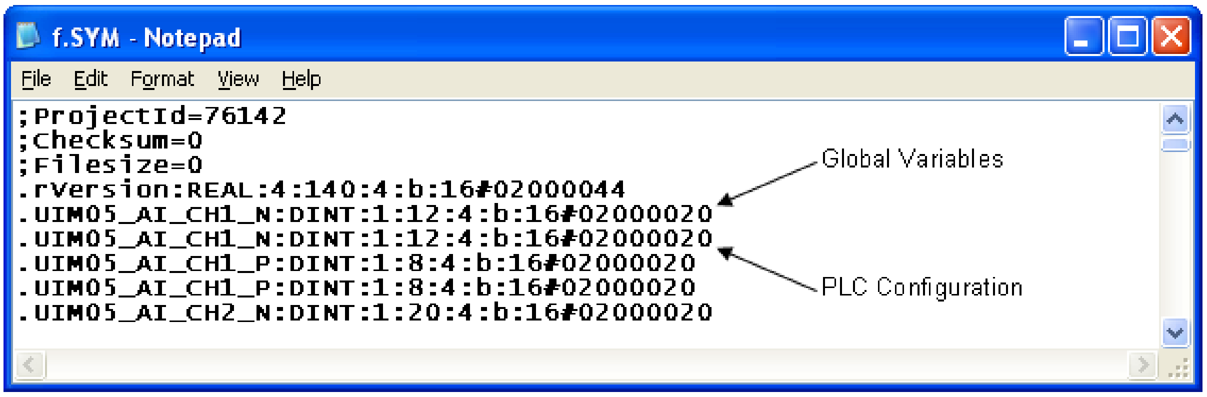

Options
Category:
Load & Save
User Information
Editor
Desktop
Colors
Directories
Log
Build
Passwords
Source download
Symbol configuration
Database-connection
Macros
Set object attributes
OK
Cancel
Es0802b.pro
POUs
library Util.lib 27.7.05 04:04:00: POUs
Resources
Global Variables
Global_Variables
Variable_Configuration (VAR_CONFIG)
library ANALYZATION.LIB 5.10.99 09:05:06: global variabl
library EXOR.lib 27.7.05 04:03:00: global variables
library EXOR_Serial.lib 27.7.05 04:15:00: global variables
library EXOR_sys.lib 27.7.05 04:15:00: global variables
library IECSFC.LIB 27.7.05 04:03:00: global variables
library SYSLIBCALLBACK.LIB 27.7.05 04:03:00: global vari
library SYSLIBTIME.LIB 27.7.05 04:03:00: global variables
library SYSTASKINFO.LIB 27.7.05 04:03:00: global variabl
library Util.lib 27.7.05 04:04:00: global variables
PLC Configuration
Export variables of object

Figure 17



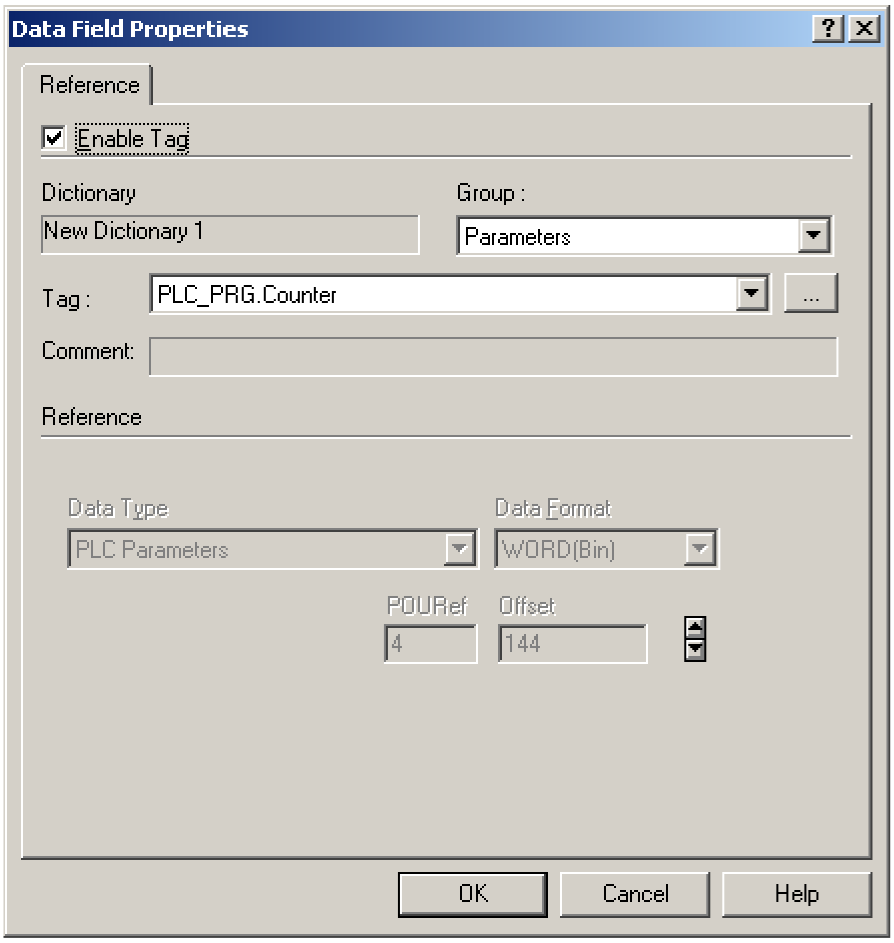
```
;ProjectId=76142
;Checksum=0
;Filesize=0
.rVersion:REAL:4:140:4:b:16#02000044
.UIM05_AI_CH1_N:DINT:1:12:4:b:16#02000020
.UIM05_AI_CH1_N:DINT:1:12:4:b:16#02000020
.UIM05_AI_CH1_P:DINT:1:8:4:b:16#02000020
.UIM05_AI_CH1_P:DINT:1:8:4:b:16#02000020
.UIM05_AI_CH2_N:DINT:1:20:4:b:16#02000020
```

Figure 18

We suggest to remove from “PLC configuration” the entries for the Analog channels and export again the symbol file.

3.2.3 Source Selection Dialog Box

If the Internal Controller set-up uses only the internal PLC, the “Define Field” dialog box immediately appears and includes only the CoDeSys variables as shown in Figure 19.


Data Field Properties

Reference

☑ Enable Tag

Dictionary: New Dictionary 1

Group : Parameters

Tag : PLC_PRG.Counter

Comment:

Reference

Data Type: PLC Parameters

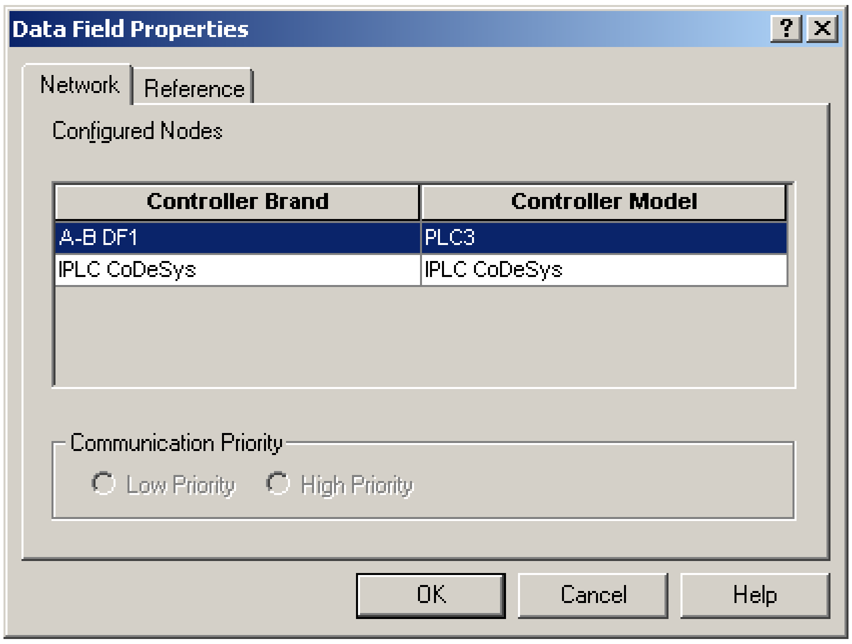
Data Format: WORD(Bin)

POURef: 4

Offset: 144

Figure 19

If the Internal Controller set-up is configured to use a combination of internal PLC and an external controller, the “Network” tab allows the source selection of the variable to be added as shown in Figure 20.


Data Field Properties

Network | Reference

Configured Nodes

Controller Brand	Controller Model
A-B DF1	PLC3
IPLC CoDeSys	IPLC CoDeSys

Communication Priority

○ Low Priority ○ High Priority

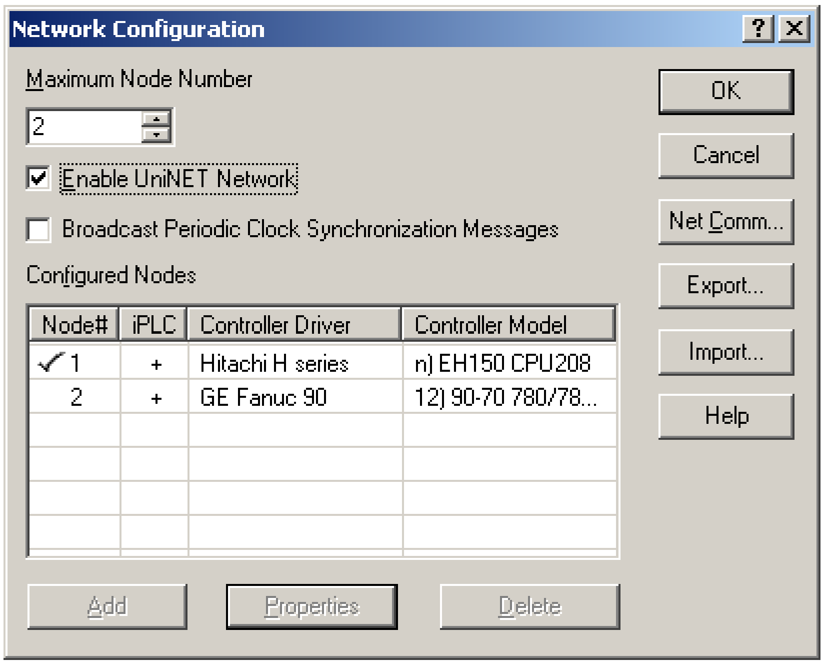
Figure 20

3.2.4 Using the Internal Controller with UniNet

If the UniNET network has been enabled and any of the UniNET servers includes an Internal Controller, then the Internal Controllers appears in the list of available data sources in the Network tab of the Data Field Properties.

An example is given in the rest of this chapter
The UniNET network has been configured according to Figure 21; the network has two panels both configured as Server and both configured to use their internal PLC, the “Network” tab of the Define Field dialog box will appear as shown in Figure 22.

There are four possible sources for a reference to be added:
- the external controller connected to the UniNET node 1 (Hitachi H series)
- the internal controller of the panel that has UniNET node 1
- the external controller connected to the UniNET node 2 (GE Fanuc 90)
- the internal controller of the panel that has UniNET node 2



Network Configuration

Maximum Node Number: 2

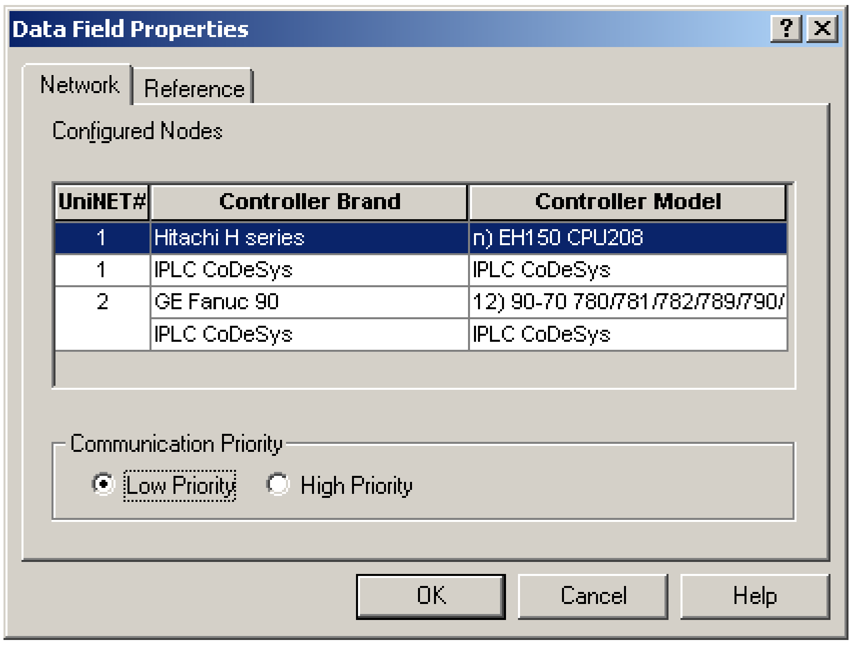
[✓] Enable UniNET Network

[ ] Broadcast Periodic Clock Synchronization Messages

Configured Nodes

Node#	iPLC	Controller Driver	Controller Model
✓ 1	+	Hitachi H series	n) EH150 CPU208
2	+	GE Fanuc 90	12) 90-70 780/78...

Figure 21



Data Field Properties

Network | Reference

Configured Nodes

UniNET#	Controller Brand	Controller Model
1	Hitachi H series	n) EH150 CPU208
1	IPLC CoDeSys	IPLC CoDeSys
2	GE Fanuc 90	12) 90-70 780/781/782/789/790/
	IPLC CoDeSys	IPLC CoDeSys

Communication Priority: [● Low Priority] ○ High Priority

Figure 22

3.2.5 Data Field Dialog Box for the Internal Controller

The “Data Field” dialog box for an Internal Controller data item is show in Figure 23.

The “Enable Tag” checkbox allows browsing the Tag list created or imported in the Designer project. See chapter 3.2 for detailed description of the Tag Editor.

In a CoDeSys system the following data types are available:

- PLC Memory
- PLC Input

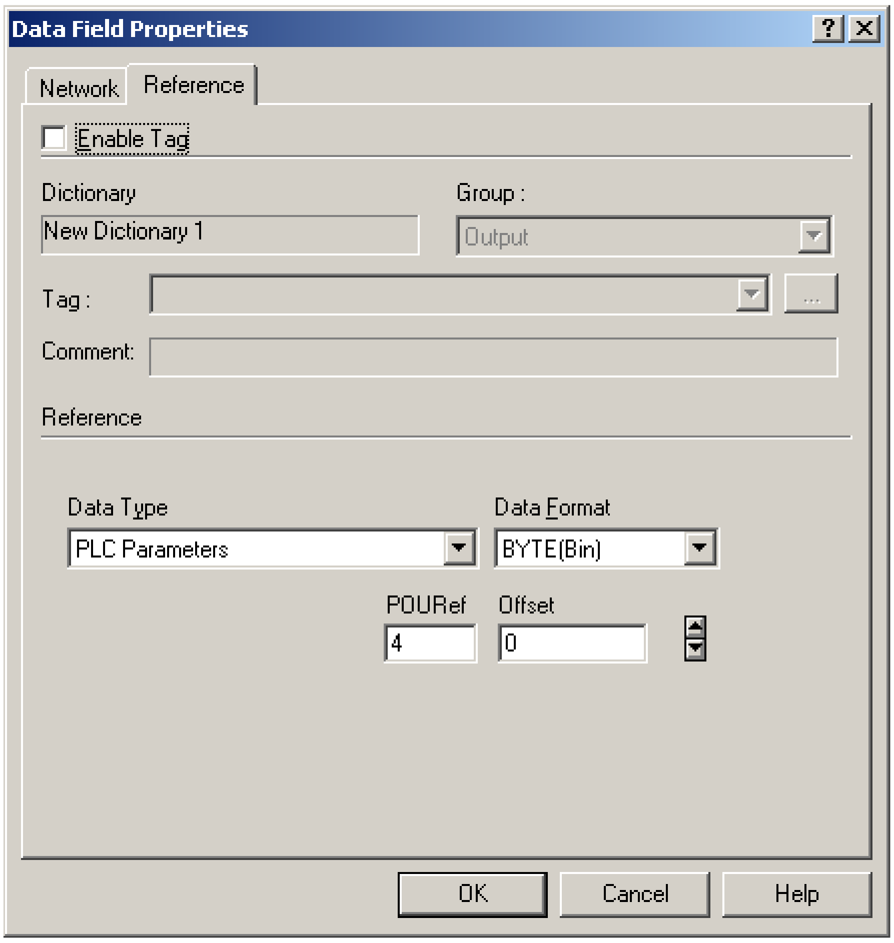
- PLC Output
- PLC Retain
- PLC Parameter
- Internal Panel Memory

The CoDeSys user program data is divided into “segments” as shown in the table below.

Segment		Type
0	%M	Memory
1	%I	Input
2	%Q	Output

Table 6

If the option “Retain in own segment” is on, retain variables are in segment 3.
Global and POU (**P**rogram **O**rganization **U**nit) local variables without direct address are in the subsequent segments, starting with segment 4. If the option “Retain in own segment” is off, they start at segment 3.
The reference to variables in the CoDeSys system consists of “POUref” (the segment), Offset and size. Detailed description will be given in the following sections.

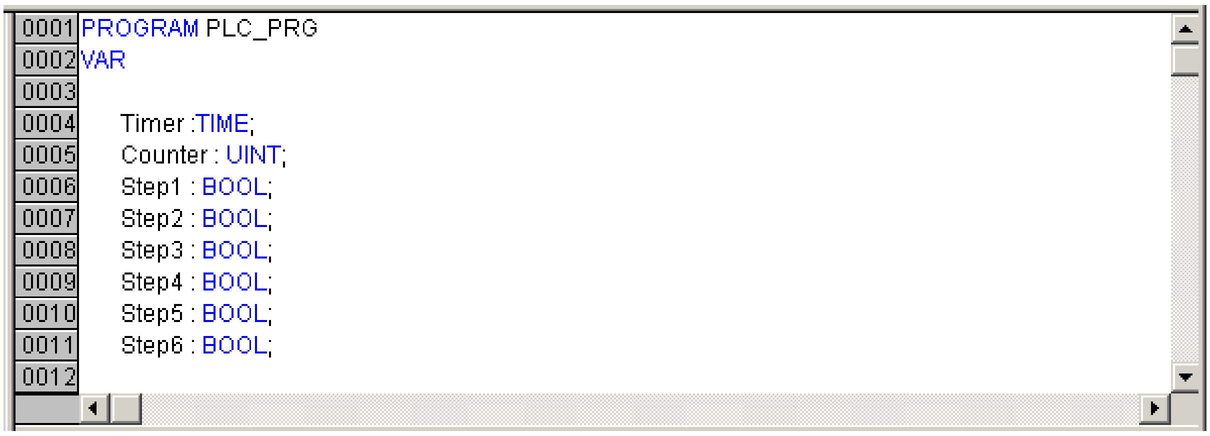

Data Field Properties

Enable Tag: ☐
Dictionary: New Dictionary 1
Group: Output
Tag:
Comment:

Reference
Data Type: PLC Parameters
Data Format: BYTE(Bin)
POURef: 4
Offset: 0

Figure 23

All variables used in a CoDeSys program must be declared in the “Declaration Editor”. The CoDeSys “Declaration Editor” is shown in Figure 24.
The Declaration Editor is used to declare variables of POUs and global variables, for data type declarations, and in the Watch and Receipt Manager

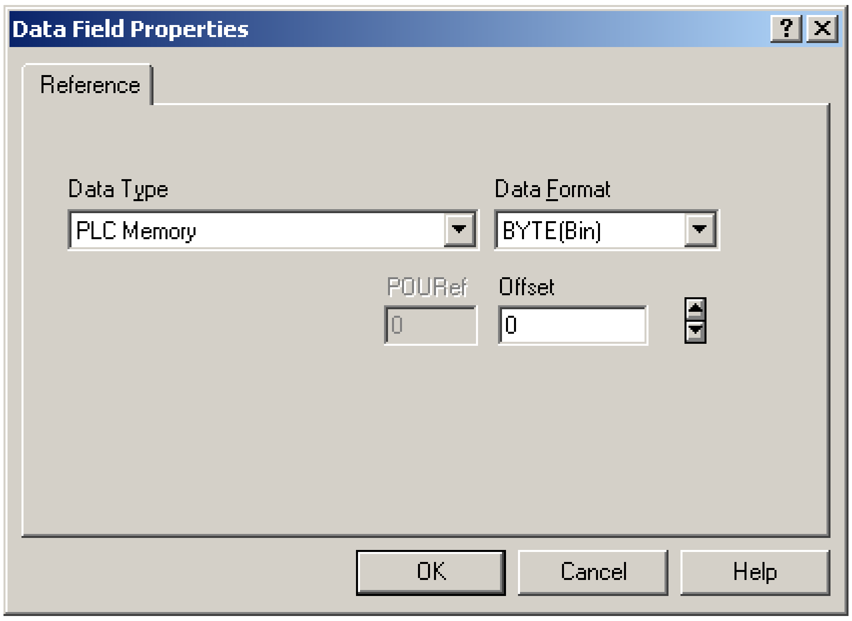


```
PROGRAM PLC_PRG
VAR

    Timer :TIME;
    Counter : UINT;
    Step1 : BOOL;
    Step2 : BOOL;
    Step3 : BOOL;
    Step4 : BOOL;
    Step5 : BOOL;
    Step6 : BOOL;

```

Figure 24

3.2.6 PLC Memory

Variable of type “PLC Memory” refer to the flag memory area in the PLC.



Data Field Properties

Reference

Data Type: PLC Memory

Data Format: BYTE(Bin)

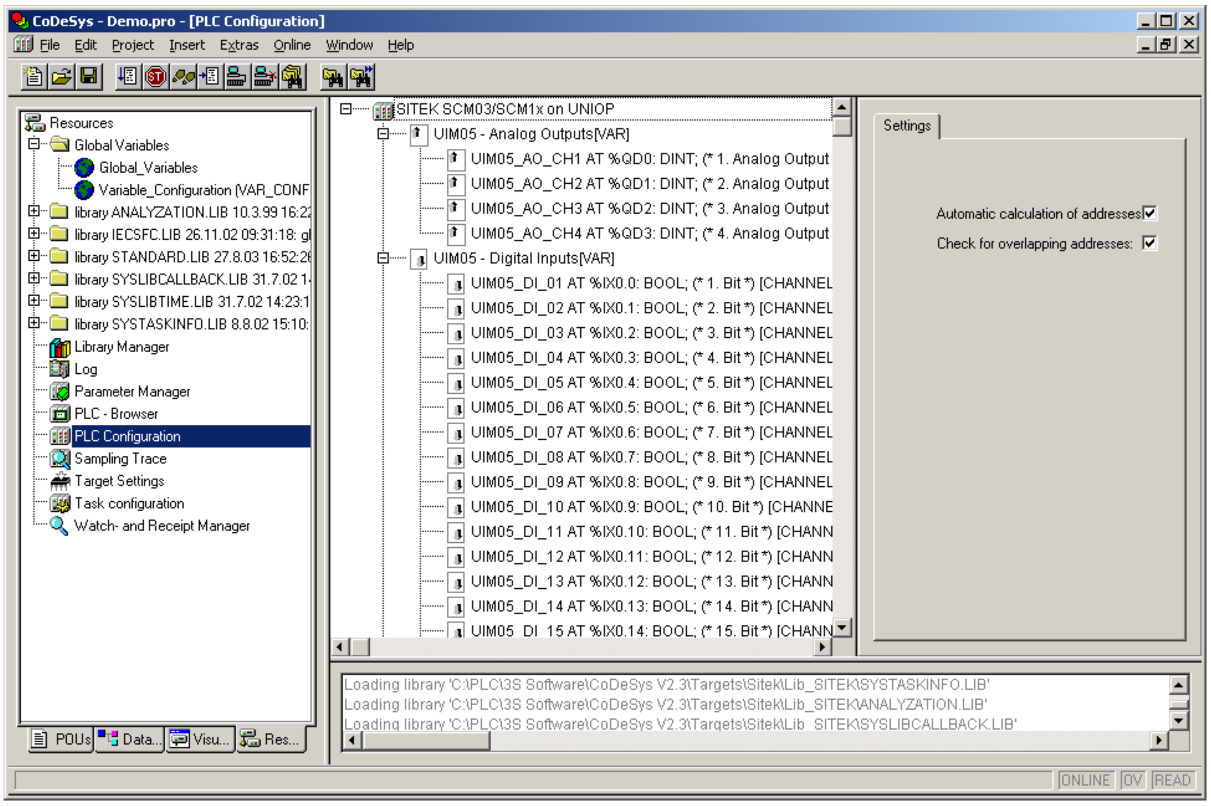
POURef: 0

Offset: 0

Figure 25

3.2.7 PLC Input and PLC Output

Variable of type “PLC Input” refer to the variables configured as “Input variables” in the “PLC Configuration” tool. The “PLC Configuration” tool can be opened from the “Resources” tree in the CoDeSys programming software.
The CoDeSys “PLC Configuration” window is shown in Figure 26.


CoDeSys - Demo.pro - [PLC Configuration]
File Edit Project Insert Extras Online Window Help
Settings
POUs Data... Visu... Res...
ONLINE OV READ

```
Resources
  Global Variables
    Global_Variables
    Variable_Configuration (VAR_CONF
  library ANALYZATION.LIB 10.3.99 16:2
  library IECSFC.LIB 26.11.02 09:31:18: gl
  library STANDARD.LIB 27.8.03 16:52:2
  library SYSLIBCALLBACK.LIB 31.7.02 1
  library SYSLIBTIME.LIB 31.7.02 14:23:1
  library SYSTASKINFO.LIB 8.8.02 15:10:
  Library Manager
  Log
  Parameter Manager
  PLC - Browser
  PLC Configuration
  Sampling Trace
  Target Settings
  Task configuration
  Watch- and Receipt Manager
```

```
SITEK SCM03/SCM1x on UNIOP
  UIM05 - Analog Outputs[VAR]
    UIM05_AO_CH1 AT %QD0: DINT; (* 1. Analog Output
    UIM05_AO_CH2 AT %QD1: DINT; (* 2. Analog Output
    UIM05_AO_CH3 AT %QD2: DINT; (* 3. Analog Output
    UIM05_AO_CH4 AT %QD3: DINT; (* 4. Analog Output
  UIM05 - Digital Inputs[VAR]
    UIM05_DI_01 AT %IX0.0: BOOL; (* 1. Bit *) [CHANNEL
    UIM05_DI_02 AT %IX0.1: BOOL; (* 2. Bit *) [CHANNEL
    UIM05_DI_03 AT %IX0.2: BOOL; (* 3. Bit *) [CHANNEL
    UIM05_DI_04 AT %IX0.3: BOOL; (* 4. Bit *) [CHANNEL
    UIM05_DI_05 AT %IX0.4: BOOL; (* 5. Bit *) [CHANNEL
    UIM05_DI_06 AT %IX0.5: BOOL; (* 6. Bit *) [CHANNEL
    UIM05_DI_07 AT %IX0.6: BOOL; (* 7. Bit *) [CHANNEL
    UIM05_DI_08 AT %IX0.7: BOOL; (* 8. Bit *) [CHANNEL
    UIM05_DI_09 AT %IX0.8: BOOL; (* 9. Bit *) [CHANNEL
    UIM05_DI_10 AT %IX0.9: BOOL; (* 10. Bit *) [CHANNE
    UIM05_DI_11 AT %IX0.10: BOOL; (* 11. Bit *) [CHANN
    UIM05_DI_12 AT %IX0.11: BOOL; (* 12. Bit *) [CHANN
    UIM05_DI_13 AT %IX0.12: BOOL; (* 13. Bit *) [CHANN
    UIM05_DI_14 AT %IX0.13: BOOL; (* 14. Bit *) [CHANN
    UIM05_DI_15 AT %IX0.14: BOOL; (* 15. Bit *) [CHANN
```

Automatic calculation of addresses ✓
Check for overlapping addresses: ✓

```
Loading library 'C:\PLC\3S Software\CoDeSys V2.3\Targets\Sitek\Lib_SITEK\SYSTASKINFO.LIB'
Loading library 'C:\PLC\3S Software\CoDeSys V2.3\Targets\Sitek\Lib_SITEK\ANALYZATION.LIB'
Loading library 'C:\PLC\3S Software\CoDeSys V2.3\Targets\Sitek\Lib_SITEK\SYSLIBCALLBACK.LIB'
```

Figure 26

Input and Output points can be addresses either using tags or pointing directly to them (direct addressing mode). Designer can specify 4 different Data Types for Input and Output, depending on the size of the element you want to address.

The data types for Input are:

PLC Input (X)	to address bit elements
PLC Input (B)	to address byte elements
PLC Input (W)	to address word elements
PLC Input (D)	to address double word elements

The data types for Output are:

PLC Output (X)	to address bit elements
PLC Output (B)	to address byte elements
PLC Output (W)	to address word elements
PLC Output (D)	to address double word elements

The different data types must be used depending on the PC Configuration built into the CoDeSys PLC program; the mnemonics are compatible.

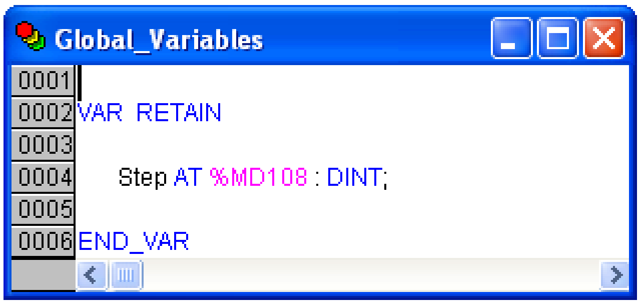
3.2.8 PLC Retain

Variables of type "PLC Retain" refer to CoDeSys variables declared in the "Declaration Editor" in the section enclosed between the keywords VAR_RETAIN and END_VAR.
These variables maintain their value when the controller is powered off and even after an uncontrolled shutdown of the controller.

The content of retain variables is saved when the device is turned off and restored at the following power-up.

Note *UniOP firmware can allocate memory for retentive variables ONLY when the panel is in Operation Mode.*

Note *Using "Memory" addresses for Retain variables is not allowed (Figure 27)*



```
VAR RETAIN

    Step AT %MD108 : DINT;

END_VAR
```

Figure 27

There is a limit to the maximum number of retentive variables that can be defined. The current implementation will support a maximum of up to 2048 bytes.

At programming time it will be responsibility of the programmer to ensure that the maximum amount of available memory will not be exceeded. When compiling the project, the CoDeSys software will use the specific Target Settings information to check if the total amount of retentive variables has been exceeded.

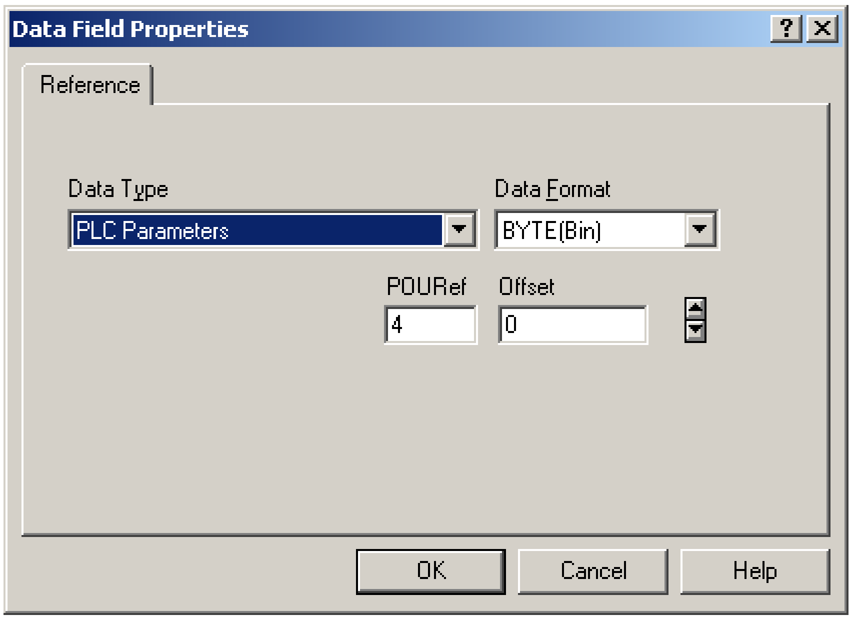
As SCM has no on-board battery backup, removing the SCM controller from the unit will result in losing the information of the retentive memories.

The content of retentive memories will also be lost in the following cases:
- a new project file has been downloaded to UniOP
- a new firmware has been downloaded to UniOP
- the SCM module is moved from one UniOP panel to another
- a new PLC program is downloaded to the SCM

Note: *Retentive Memory mechanism is activated only 2 power cycles after CoDeSys project download or UniOP firmware download.*

3.2.9 PLC Parameter

The data type "PLC Parameter" refers to all POU local and global variables. The POUref parameter is editable depending on the fact that the retain variables are allocated in a separate segment.
Target settings for SCM03-C / SCM05-C / SCM11-C / SCM12-C uses always by default a separate segment (the number 3) for all the retain variables.
Retain settings are in the "Target Settings" dialog box as shown in Figure 29. They are fixed and can not be changed by the user.


Data Field Properties

Reference

Data Type: PLC Parameters

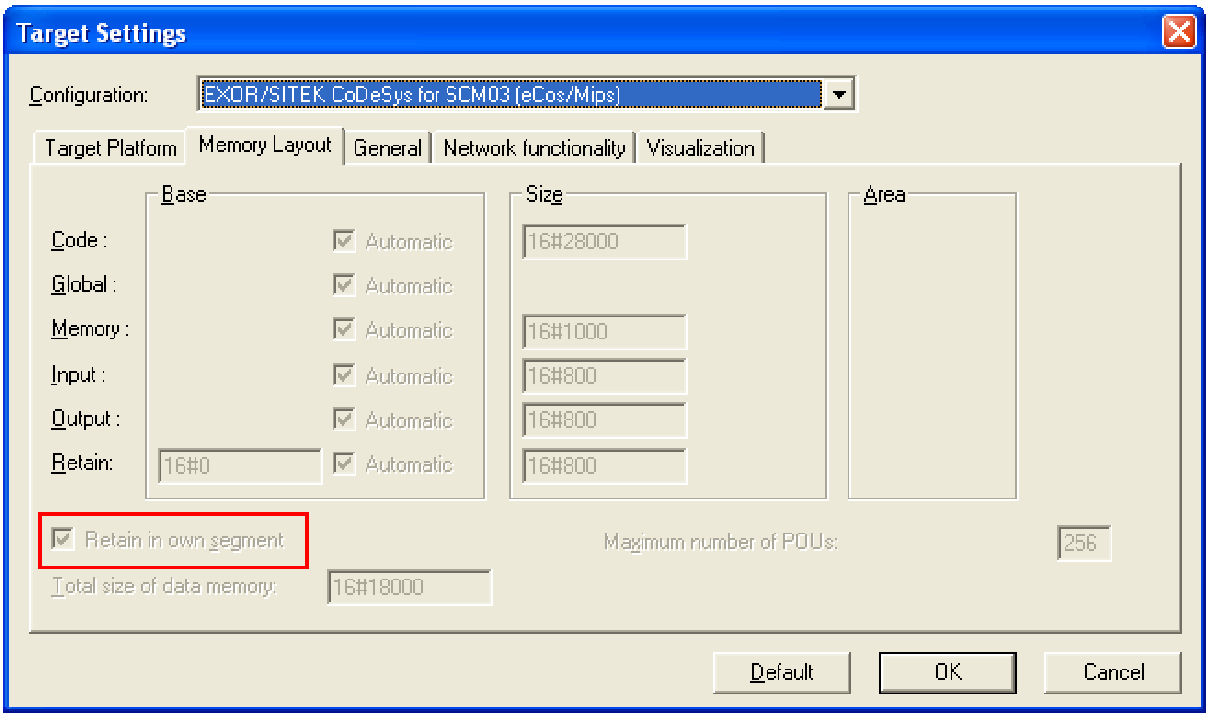
Data Format: BYTE(Bin)

POURef: 4

Offset: 0

OK | Cancel | Help

Figure 28


Target Settings

Configuration: EXOR/SITEK CoDeSys for SCM03 (eCos/Mips)

Target Platform | Memory Layout | General | Network functionality | Visualization

	Base		Size	Area
Code :		✓ Automatic	16#28000	
Global :		✓ Automatic		
Memory :		✓ Automatic	16#1000	
Input :		✓ Automatic	16#800	
Output :		✓ Automatic	16#800	
Retain:	16#0	✓ Automatic	16#800	

✓ Retain in own segment

Total size of data memory: 16#18000

Maximum number of POUs: 256

Default | OK | Cancel

Figure 29

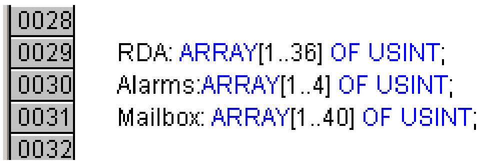
3.2.10 Internal Panel Memory

The variables of type Internal Panel Memory refer to an internal memory structure, located in the UniOP panel. Accessing this memory does not involve any communication with the iPLC controller memory.

3.3 Using the RDA

The Reserved Data Area can be configured in the memory of the Internal Controller.

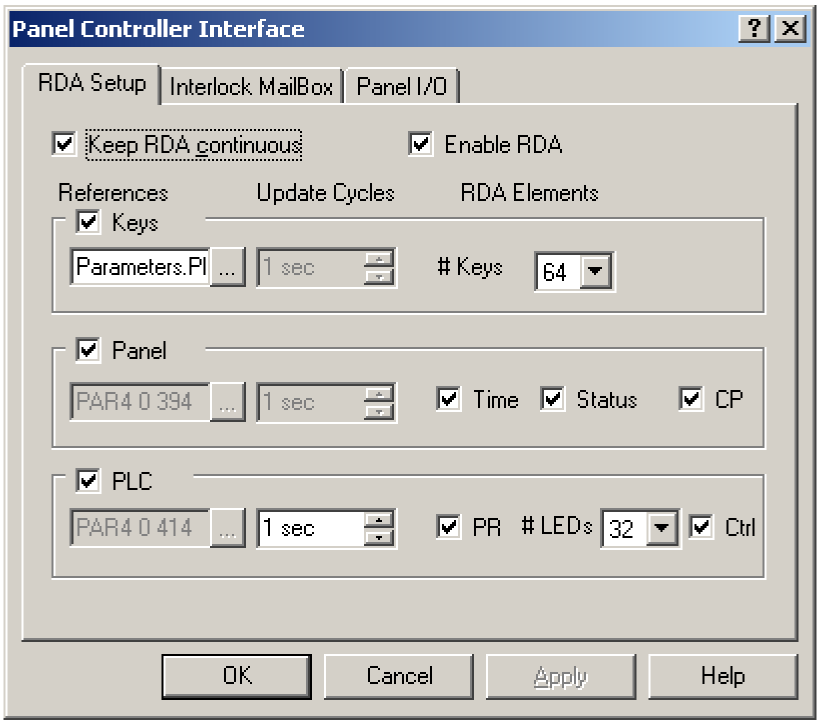
To use the RDA a certain number of Tag's with contiguous addresses must be configured in the CoDeSys program.
The easiest way to declare a list of variables in CoDeSys, which can be considered "contiguous", is to use an array as shown in Figure 30 and address the RDA using the tags.



```
RDA: ARRAY[1..36] OF USINT;
Alarms:ARRAY[1..4] OF USINT;
Mailbox: ARRAY[1..40] OF USINT;
```

Figure 30

The array must contain bytes elements declared in CoDeSys as unsigned short integer (USINT).
The absolute address into the controller memory segment of a variable declared in CoDeSys is only visible in the symbol file created by the programming software at compile time.
The CoDeSys array structure ensures that all its elements have contiguous addresses; the first element of the array can be used as offset reference for the RDA area.



☑ Keep RDA continuous ☑ Enable RDA

References / Update Cycles / RDA Elements

☑ Keys: Parameters.PI ... 1 sec # Keys 64

☑ Panel: PAR4 0 394 ... 1 sec ☑ Time ☑ Status ☑ CP

☑ PLC: PAR4 0 414 ... 1 sec ☑ PR # LEDs 32 ☑ Ctrl

Figure 31

If the "Keep RDA Contiguous" check box is enabled, Designer calculates the proper address of the RDA segments, showing the absolute memory address into the PLC memory.

Considering the example of variable declaration shown in Figure 30, the Keyboard area is mapped as shown in the Figure 32.

Tag	RDA[4]								RDA[3]								RDA[2]								RDA[1]							
Bit	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
	F32	F31	F30	F29	F28	F27	F26	F25	F24	F23	F22	F21	F20	F19	F18	F17	F16	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1

Figure 32 – The Keyboard Area

Tag	RDA[8]								RDA[7]								RDA[6]								RDA[5]							
Bit	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
	Day								Month								Reserved								Day of week							
	Minutes								Seconds								Year								Hour							

Figure 33 – The Panel Area

Tag	RDA[12]								RDA[11]								RDA[10]								RDA[9]							
Bit	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
	L16	L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	PR (Page Request)															

Figure 34 – The PLC Area

3.4 Programming the Mailbox

The Mailbox can be configured in the Internal Controller memory area using an array of bytes (USINT). To the mailbox should be reserved an array of minimum 40 bytes in length.

Tag	Mailbox[4]								Mailbox[3]								Mailbox[2]								Mailbox[1]							
Bit	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
	Command/Response Word																Status Word															
	Parameter 1																Parameter 0															
	Parameter 3																Parameter 2															
	....																.....															

Figure 35 – The Mailbox

3.5 Alarms

The Alarm area in the Internal Controller memory is organized as bytes. An array of bytes (USINT) can be configured to handle Alarms.
Alarm bits are organized according to Figure 36.

Tag	Alarm[4]								Alarm[3]								Alarm[2]								Alarm[1]							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A32	A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1

Figure 36 – The Alarm Area

3.6 Alternative Approach to RDA, Alarms and Mailbox

Instead of using arrays the Controller interface areas can be defined statically into the iPLC memory declaring all variables as global with their corresponding names.

As an example, the RDA Keys area can be declared as follows:

```
(* RDA : Keys area *)
        RDA_Keys1                      AT %MW0: WORD;
        RDA_Keys2                      AT %MW1: WORD;
        RDA_Keys3                      AT %MW2: WORD;
        RDA_Keys4                      AT %MW3: WORD;
```

The RDA Panel area can be declared as follows:

```
(* RDA : panel area *)
      RDA_RTC_DayOfWeek              AT %MB8: BYTE;
      RDA_RTC_Month                  AT %MB10: BYTE;
      RDA_RTC_Day              AT %MB11: BYTE;
      RDA_RTC_Hour                   AT %MB12: BYTE;
      RDA_RTC_Year                   AT %MB13: BYTE;
```

This means that the individual bits in the Status word can be identified as follows:

```
      RDA_S0                         AT %MX8.0:BOOL;
      RDA_S1                         AT %MX8.1:BOOL;
      RDA_S2                         AT %MX8.2:BOOL;
      RDA_S3                         AT %MX8.3:BOOL;
      RDA_S4                         AT %MX8.4:BOOL;
      RDA_S5                         AT %MX8.5:BOOL;
      RDA_S6                         AT %MX8.6:BOOL;
      RDA_S7                         AT %MX8.7:BOOL;
      RDA_S8                         AT %MX8.8:BOOL;
      RDA_S13                        AT %MX8.13:BOOL;
      RDA_S14                        AT %MX8.14:BOOL;
      RDA_S15                        AT %MX8.15:BOOL;
```

A Similar approach is valid also for the Mailbox area:

```
(* MailBox mapping *)

      MB_Status                      AT %MW21:WORD;
      MB_CmdResp                     AT %MW22:WORD;
      MB_Param00                     AT %MW23:WORD;
      MB_Param01                     AT %MW24:WORD;
      MB_Param02                     AT %MW25:WORD;
      MB_Param03                     AT %MW26:WORD;
      MB_Param04                     AT %MW27:WORD;
      MB_Param05                     AT %MW28:WORD;
      MB_Param06                     AT %MW29:WORD;
      MB_Param07                     AT %MW30:WORD;
      MB_Param08                     AT %MW31:WORD;
      MB_Param09                     AT %MW32:WORD;
      MB_Param10                     AT %MW33:WORD;
      MB_Param11                     AT %MW34:WORD;
      MB_Param12                     AT %MW35:WORD;
      MB_Param13                     AT %MW36:WORD;
      MB_Param14                     AT %MW37:WORD;
      MB_Param15                     AT %MW38:WORD;
      MB_Param16                     AT %MW39:WORD;
      MB_Param17                     AT %MW40:WORD;
```

The main advantage of this approach would be the natural possibility to overlay the variables definition and to give to each word, byte or bit the proper tag for a proper reference into the PLC program and later into the Designer project.
This approach requires of course accepting that a fixed memory area in the controller memory is allocated and reserved to UniOP RDA.

3.7 Transferring Data with the Internal Controller

The UniOP Data Transfer function can be used to copy data from an external controller to the CoDeSys memory. Data can also be copied from the internal CoDeSys memory to the external controller memory.

The Data Transfer process has different options based on the different Data Format of the data involved in the copy process.
Source and target tag data format should be always compatible.

In case more than one variable needs to be copied using data transfer, the physical memory address of all the elements must be contiguous. The easiest way to obtain this is in CoDeSys is configuring an array.

Before starting a copy operation the Data Transfer module checks the byte order convention used by the Source and the Target addresses. In UniOP the so-called Intel data format (little endian) is considered not inverted; the Motorola format (big endian) is considered inverted.
When Source and Target are both inverted or not inverted the Data Transfer module does not apply any transformation.
If Source and Target have different byte ordering, the Data Transfer module applies a byte swap according to the rules explained in Figure 37.

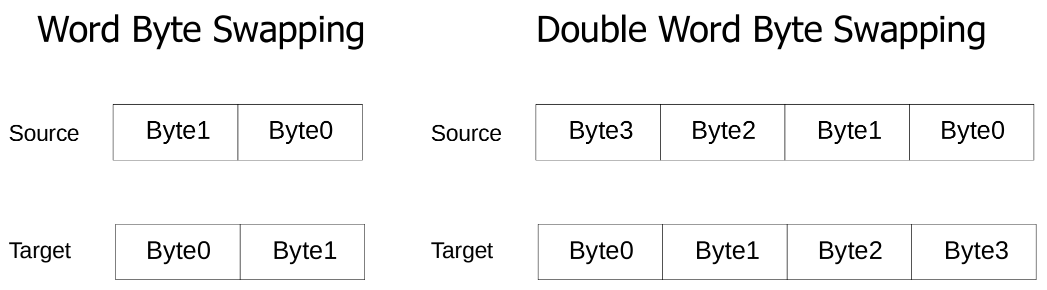


Figure 37

3.8 UniOP Communication Diagnostic

UniOP provides some useful communication diagnostic information in the
System Menu. This is available also for the internal controller. When the communication error status LED is available, it will also include status information for the Internal Controller.

There are various cases, depending on the system configuration:

1) The Designer project is configured to use an external controller along with the SCM application, the UniOP COMM LED blinks when an error occurs in the communication link with the external controller or when the error is detected in the link with SCM. In case of a communication error with the external controller, the System Menu provides the communication error code as usual.
2) The Designer project is configured to use only the Internal Controller. In this case the Communication status LED blinks when a communication error with the SCM occurs. The UniOP System Menu does not provide any information about the nature of the communication error. Communication error with the internal PLC should never appear.

Note *UniOP will communicate with the CoDeSys Internal Controller even if the PLC application is not running. This is different from internal controllers based on ISaGRAF where no communication is possible when the PLC runtime is stopped.*

4 Using the Internal Controller

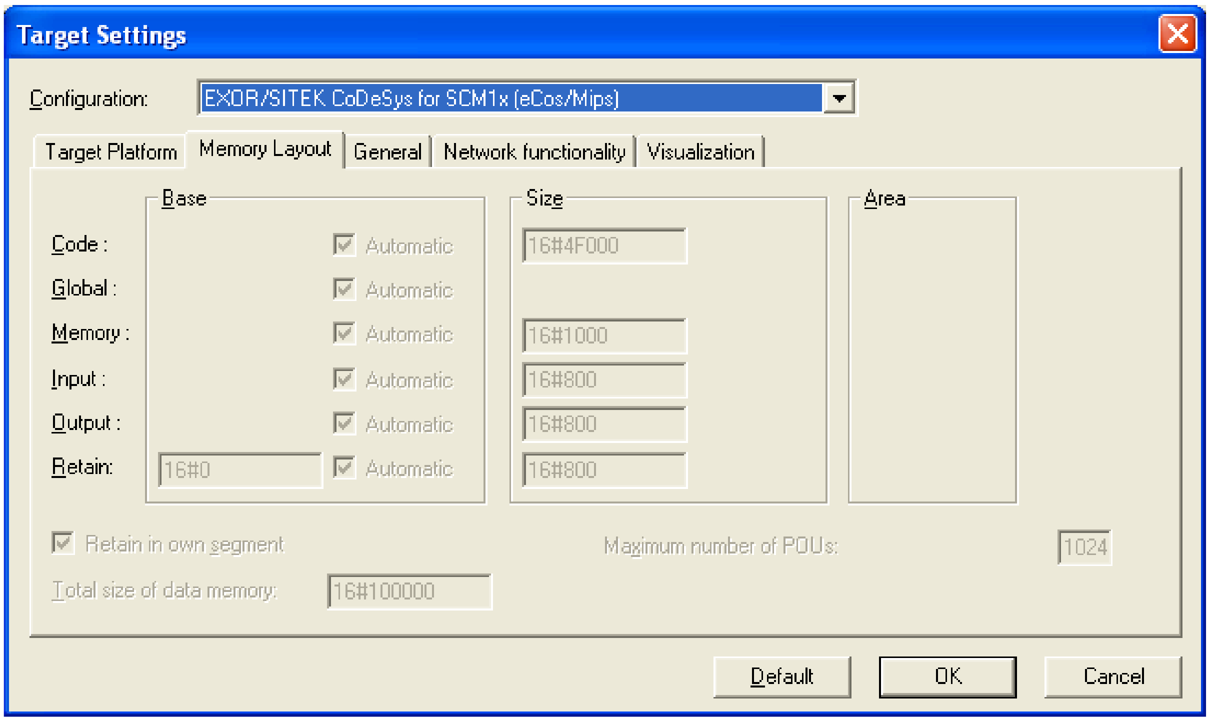
The UniOP Internal Controller is fully compatible with the CoDeSys standard.
The description of the CoDeSys programming tool is contained in the CoDeSys manual distributed with the package; please refer to this documentation for detailed information.

4.1 Target Settings

Target Settings are accessible from the "Resources" tag of the CoDeSys programming software.
Please note that most of the options are fixed and cannot be changed by the user.

This chapter includes the basic explanation of the options available in the EXOR/SITEK Target Settings.

A complete explanation of all the options is available in the CoDeSys online help. The tools provided by 3S to the OEMs allow a high level of customization of the Target Settings dialog depending on the specific hardware requirements and capabilities. Most of the options described in the on line help are not available in the EXOR/SITEK Target Support Package. This is not an error but the result of the OEM decision to keep the interface as simple as possible leaving to the user the possibility to control only some of the most important options.



Configuration: EXOR/SITEK CoDeSys for SCM1x (eCos/Mips)

	Base		Size	Area
Code :		☑ Automatic	16#4F000	
Global :		☑ Automatic		
Memory :		☑ Automatic	16#1000	
Input :		☑ Automatic	16#800	
Output :		☑ Automatic	16#800	
Retain:	16#0	☑ Automatic	16#800	

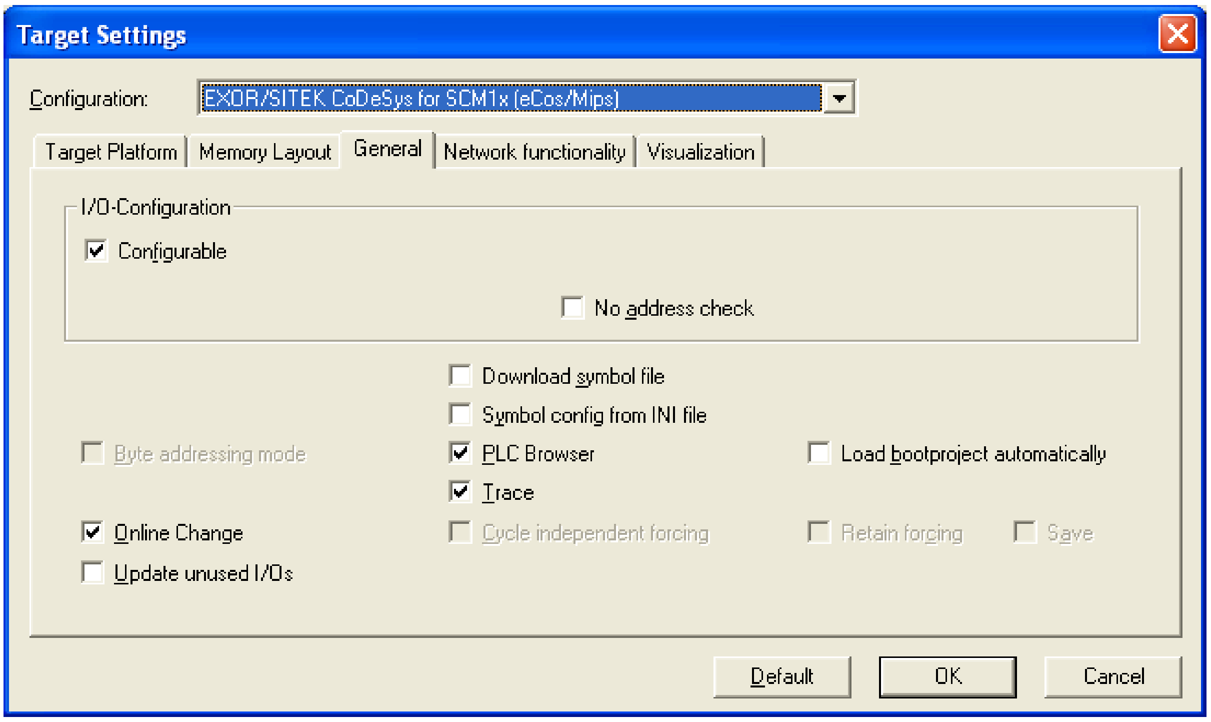
☑ Retain in own segment

Maximum number of POUs: 1024

Total size of data memory: 16#100000

Figure 38

Figure 38 shows the "Memory Layout" tab of the Target Settings.
The "Maximum number of POUs" specifies the max number of POU allowed in a project.
The maximum number of POUs supported by the SCM03-C, SCM05-C, SCM11-C and SCM12-C modules is 1024.
The user can specify here any number; at compilation time CoDeSys will verify that the actual number of POU defined in the project does not exceed the value specified in target settings.
Please note that at download time, if the number of POUs used is greater than 1024, it will be not possible to download the project to the target.


Target Settings

Configuration: EXOR/SITEK CoDeSys for SCM1x (eCos/Mips)

Target Platform | Memory Layout | General | Network functionality | Visualization

I/O-Configuration
- [✓] Configurable
- [ ] No address check

- [ ] Download symbol file
- [ ] Symbol config from INI file
- [ ] Byte addressing mode
- [✓] PLC Browser
- [ ] Load bootproject automatically
- [✓] Trace
- [✓] Online Change
- [ ] Cycle independent forcing
- [ ] Retain forcing
- [ ] Save
- [ ] Update unused I/Os

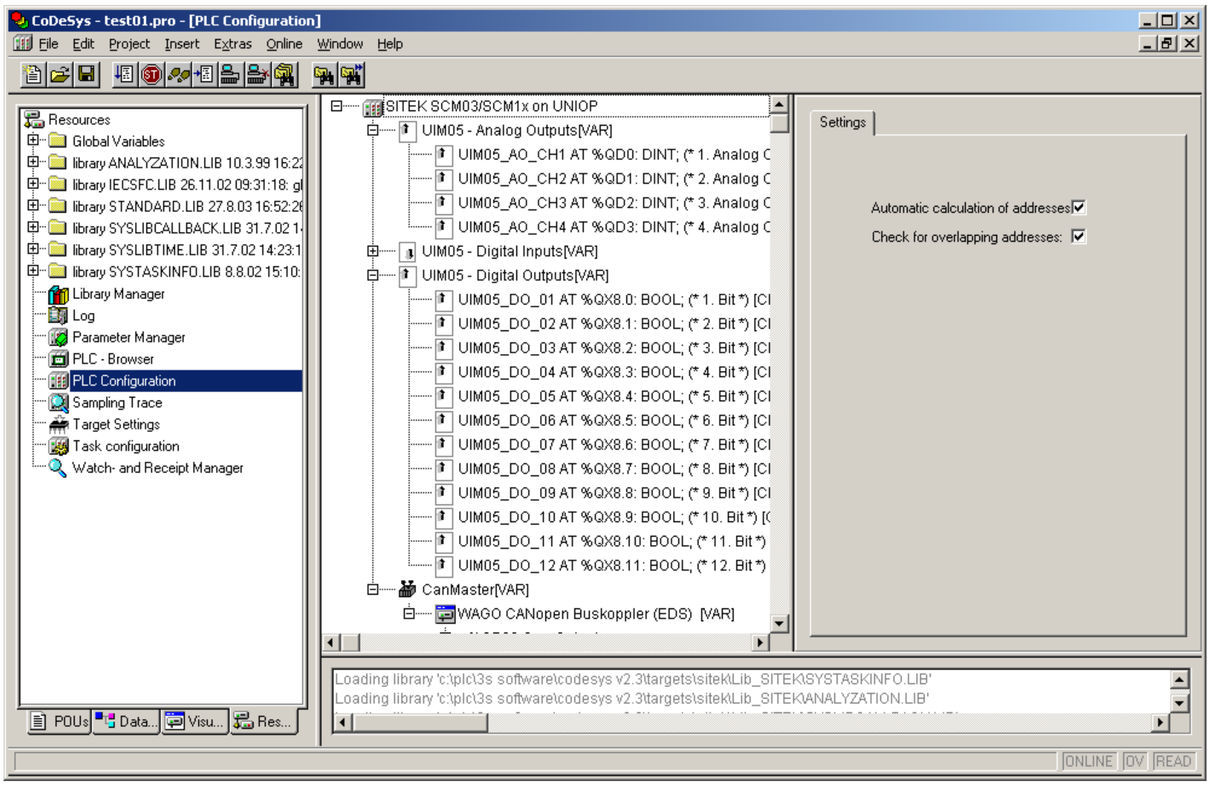
Figure 39

Figure 39 shows the “General” tab and the customizable options available for the EXOR/SITEK target.

Network functionality (Network Global Variables) is supported starting from SCM firmware version V4.01K (SCM11-C and SCM12-C only). Visualization capabilities are at the moment not supported. The corresponding tab in Target Setting do not contain any information.

4.2 Setting-up the PLC Configuration

The PLC configuration of the Internal Controller must be defined in the “PLC Configuration” tool available under the “Resources” tag as shown in Figure 40.



CoDeSys - test01.pro - [PLC Configuration]

File Edit Project Insert Extras Online Window Help

```
Resources
  Global Variables
  library ANALYZATION.LIB 10.3.99 16:22
  library IECSFC.LIB 26.11.02 09:31:18: gl
  library STANDARD.LIB 27.8.03 16:52:26
  library SYSLIBCALLBACK.LIB 31.7.02 1
  library SYSLIBTIME.LIB 31.7.02 14:23:1
  library SYSTASKINFO.LIB 8.8.02 15:10:
  Library Manager
  Log
  Parameter Manager
  PLC - Browser
  PLC Configuration
  Sampling Trace
  Target Settings
  Task configuration
  Watch- and Receipt Manager
```

```
SITEK SCM03/SCM1x on UNIOP
  UIM05 - Analog Outputs[VAR]
    UIM05_AO_CH1 AT %QD0: DINT; (* 1. Analog C
    UIM05_AO_CH2 AT %QD1: DINT; (* 2. Analog C
    UIM05_AO_CH3 AT %QD2: DINT; (* 3. Analog C
    UIM05_AO_CH4 AT %QD3: DINT; (* 4. Analog C
  UIM05 - Digital Inputs[VAR]
  UIM05 - Digital Outputs[VAR]
    UIM05_DO_01 AT %QX8.0: BOOL; (* 1. Bit *) [CI
    UIM05_DO_02 AT %QX8.1: BOOL; (* 2. Bit *) [CI
    UIM05_DO_03 AT %QX8.2: BOOL; (* 3. Bit *) [CI
    UIM05_DO_04 AT %QX8.3: BOOL; (* 4. Bit *) [CI
    UIM05_DO_05 AT %QX8.4: BOOL; (* 5. Bit *) [CI
    UIM05_DO_06 AT %QX8.5: BOOL; (* 6. Bit *) [CI
    UIM05_DO_07 AT %QX8.6: BOOL; (* 7. Bit *) [CI
    UIM05_DO_08 AT %QX8.7: BOOL; (* 8. Bit *) [CI
    UIM05_DO_09 AT %QX8.8: BOOL; (* 9. Bit *) [CI
    UIM05_DO_10 AT %QX8.9: BOOL; (* 10. Bit *) [(
    UIM05_DO_11 AT %QX8.10: BOOL; (* 11. Bit *)
    UIM05_DO_12 AT %QX8.11: BOOL; (* 12. Bit *)
  CanMaster[VAR]
    WAGO CANopen Buskoppler (EDS)  [VAR]
```

Settings

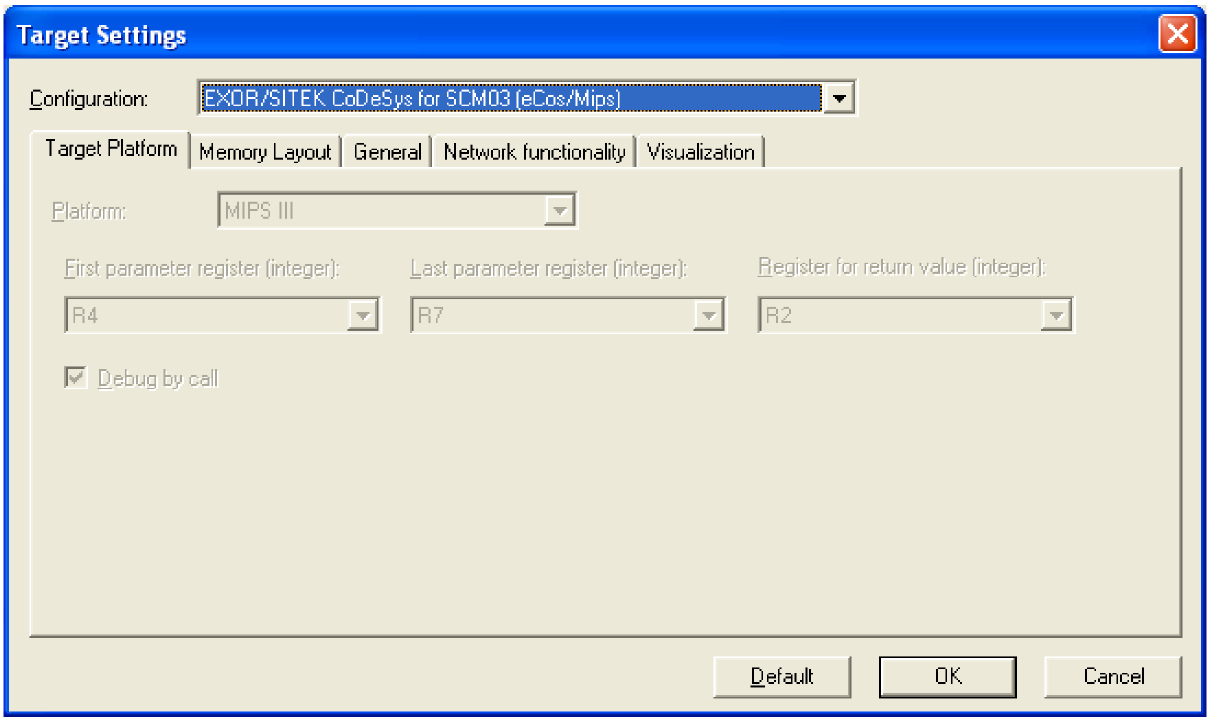
Automatic calculation of addresses ☑

Check for overlapping addresses: ☑

```
Loading library 'c:\plc\3s software\codesys v2.3\targets\sitek\Lib_SITEK\SYSTASKINFO.LIB'
Loading library 'c:\plc\3s software\codesys v2.3\targets\sitek\Lib_SITEK\ANALYZATION.LIB'
```

Figure 40

To obtain a valid PLC Configuration, the target settings must be properly configured.
CoDeSys implementation for UniOP SCM modules, corresponds to the target “CoDeSys for eCos/Mips on SCM03/SCM1x SITEK S.p.A.”, as shown in Figure 41.



Target Settings

Configuration: EXOR/SITEK CoDeSys for SCM03 (eCos/Mips)

Target Platform | Memory Layout | General | Network functionality | Visualization

Platform: MIPS III

First parameter register (integer): R4

Last parameter register (integer): R7

Register for return value (integer): R2

☑ Debug by call

Figure 41

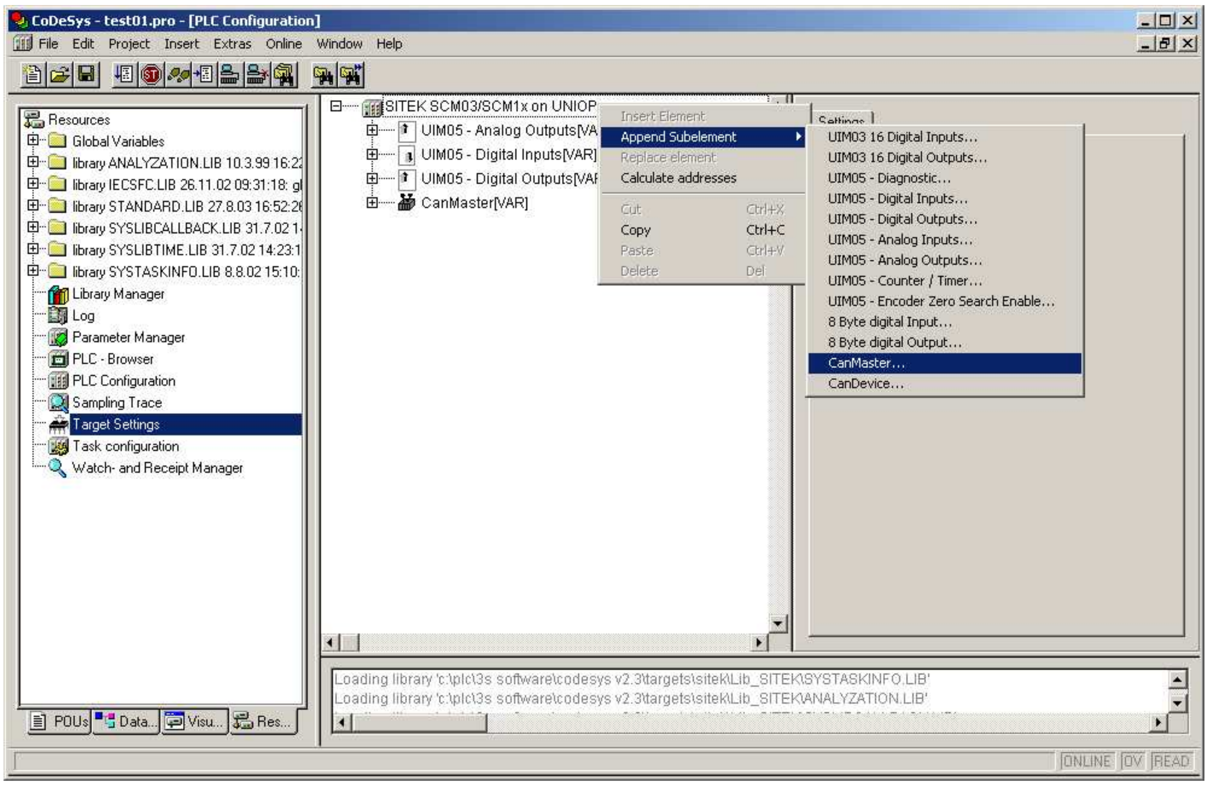
The correct selection of the target ensures a proper configuration of the CoDeSys programming software environment.

4.2.1 Configuring CANopen Distributed I/O

Distributed I/O systems based on the CANopen fieldbus interface of the SCM03-C can be easily configured using the I/O Connection tool.

Note: *SCM11-C modules DO NOT have the CANopen interface; present chapter refers ONLY to SCM03-C, SCM03-C and SCM12-C modules*

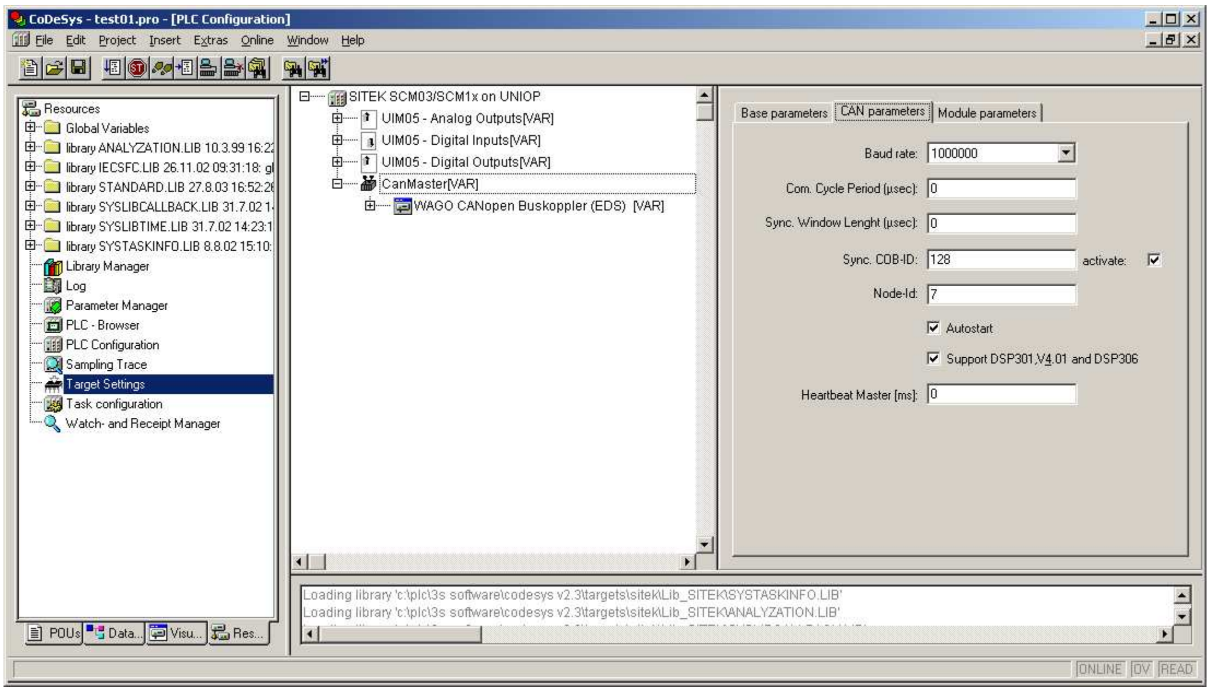
The Internal Controller CANopen master interface must be configured adding the “CanMaster” device in the PLC Configuration tool. Right-click over the root of the tree as shown in Figure 42 to select the device from a list of available boards.


CoDeSys - test01.pro - [PLC Configuration]
File Edit Project Insert Extras Online Window Help
Resources
Global Variables
library ANALYZATION.LIB 10.3.99 16:2
library IECSFC.LIB 26.11.02 09:31:18: gl
library STANDARD.LIB 27.8.03 16:52:2
library SYSLIBCALLBACK.LIB 31.7.02 1
library SYSLIBTIME.LIB 31.7.02 14:23:1
library SYSTASKINFO.LIB 8.8.02 15:10:
Library Manager
Log
Parameter Manager
PLC - Browser
PLC Configuration
Sampling Trace
Target Settings
Task configuration
Watch- and Receipt Manager
POUs
Data...
Visu...
Res...
SITEK SCM03/SCM1x on UNIOP
UIM05 - Analog Outputs[VA
UIM05 - Digital Inputs[VAR]
UIM05 - Digital Outputs[VAI
CanMaster[VAR]
Insert Element
Append Subelement
Replace element
Calculate addresses
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Settings
UIM03 16 Digital Inputs...
UIM03 16 Digital Outputs...
UIM05 - Diagnostic...
UIM05 - Digital Inputs...
UIM05 - Digital Outputs...
UIM05 - Analog Inputs...
UIM05 - Analog Outputs...
UIM05 - Counter / Timer...
UIM05 - Encoder Zero Search Enable...
8 Byte digital Input...
8 Byte digital Output...
CanMaster...
CanDevice...
ONLINE
OV
READ

```
Loading library 'c:\plc\3s software\codesys v2.3\targets\sitek\Lib_SITEK\SYSTASKINFO.LIB'
Loading library 'c:\plc\3s software\codesys v2.3\targets\sitek\Lib_SITEK\ANALYZATION.LIB'
```

Figure 42

The parameters of the CAN controller are configurable in the rightmost part of the PLC Configuration dialog, once the CAN Master device has been selected in the tree with one mouse click. The configuration window is shown in Figure 43.



CoDeSys - test01.pro - [PLC Configuration]

File Edit Project Insert Extras Online Window Help

Resources
- Global Variables
- library ANALYZATION.LIB 10.3.99 16:2
- library IECSFC.LIB 26.11.02 09:31:18: g
- library STANDARD.LIB 27.8.03 16:52:2
- library SYSLIBCALLBACK.LIB 31.7.02 1
- library SYSLIBTIME.LIB 31.7.02 14:23:1
- library SYSTASKINFO.LIB 8.8.02 15:10:
- Library Manager
- Log
- Parameter Manager
- PLC - Browser
- PLC Configuration
- Sampling Trace
- Target Settings
- Task configuration
- Watch- and Receipt Manager

SITEK SCM03/SCM1x on UNIOP
- UIM05 - Analog Outputs[VAR]
- UIM05 - Digital Inputs[VAR]
- UIM05 - Digital Outputs[VAR]
- CanMaster[VAR]
  - WAGO CANopen Buskoppler (EDS) [VAR]

Base parameters | CAN parameters | Module parameters

Baud rate:	1000000
Com. Cycle Period (µsec):	0
Sync. Window Lenght (µsec):	0
Sync. COB-ID:	128 activate: ☑
Node-Id:	7
	☑ Autostart
	☑ Support DSP301,V4.01 and DSP306
Heartbeat Master [ms]:	0

```
Loading library 'c:\plc\3s software\codesys v2.3\targets\sitek\Lib_SITEK\SYSTASKINFO.LIB'
Loading library 'c:\plc\3s software\codesys v2.3\targets\sitek\Lib_SITEK\ANALYZATION.LIB'
```

POUs Data... Visu... Res...

ONLINE OV READ

Figure 43

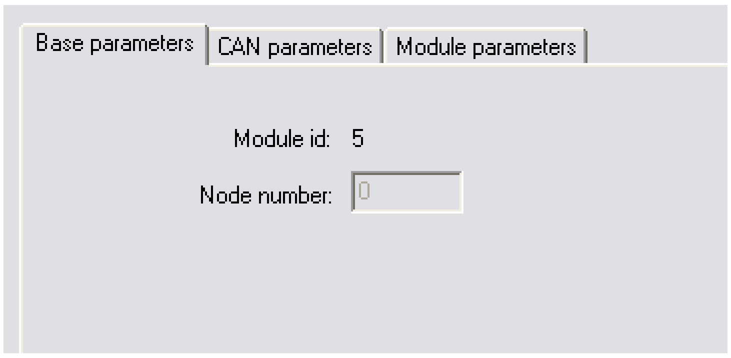
A complete and detailed description on how the CAN controller should be configured and about the configuration of CAN slave devices is included in the CoDeSys User Manual in chapter “6.5.7 – Configuration of CAN Modules”.
The next chapter provides a description of the CanMaster parameters in connection with the SCM03-C hardware and EXOR CoDeSys implementation.

4.2.2 Configuring the CANopen controller

The parameters of the CAN interface are grouped in three tabs accessible on the right part of the PLC Configuration tool when the CanMaster board is selected.

4.2.2.1 Base parameters

Figure 44 shows the “Base Parameters” tab.



Base parameters | CAN parameters | Module parameters

Module id:	5
Node number:	0

Figure 44

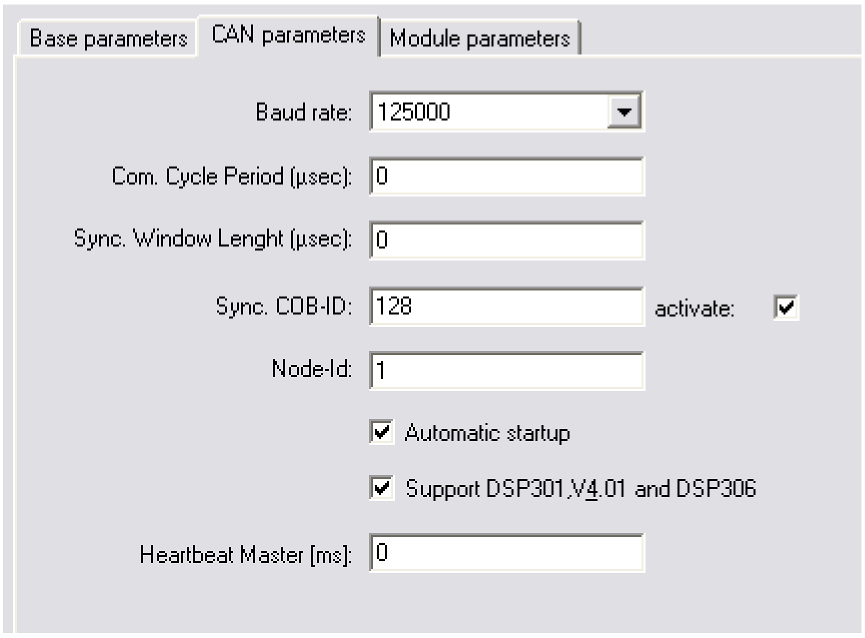
The parameters have the following meaning:

Module id	CoDeSys internal identifier used to recognize the board; it is a read only parameter displayed just for user information

Node number	CoDeSys internal identifier assigned by the programming software depending on the order of the board in the PLC Configuration layout; it is a read only parameter displayed just for user information

4.2.2.2 CAN parameters

Figure 45 shows the “CAN Parameters” tab; it contains all the parameters related to the bus configuration.



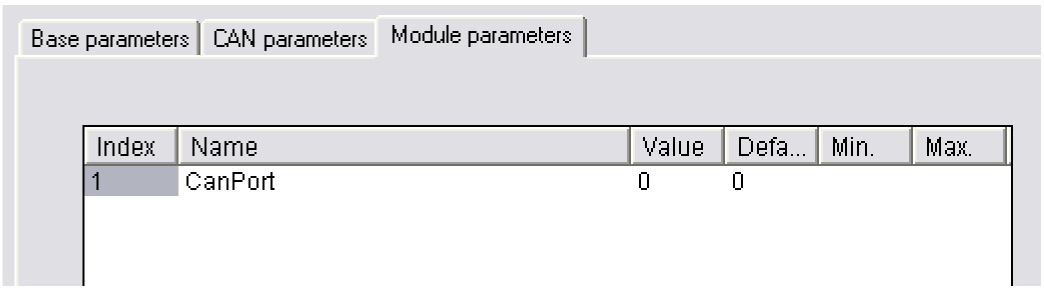
Baud rate: 125000
Com. Cycle Period (µsec): 0
Sync. Window Lenght (µsec): 0
Sync. COB-ID: 128 activate: ✓
Node-Id: 1
✓ Automatic startup
✓ Support DSP301,V4.01 and DSP306
Heartbeat Master [ms]: 0

Figure 45

Baud rate	baud rate settings for the bus speed
Com. Cycle Period	specifies the communication cycle interval related to the PDO messages exchange; if set to 0, the default value for communication cycle is set equal to 20ms; note that this is not an inferior limit; minimum allowed value is in fact 1ms
Sync. Window Len.	not supported
Sync. COB-ID	the default value (128) reflect the CiA standard for CANopen; the COB-ID can be changed depending on installation requirements
Activate	allows to specify whenever the Sync should be used or not
Node-Id	node number assigned to the CAN master; even if it can be specified, it is actually not used since the CoDeSys implementation on SCM modules does not support the SDO server functionality
Automatic Startup	specifies if the NMT startup command should be sent automatically to all configured slaves at power up or not
Support DSP301…	enables the controls for the heartbeat and of the Modular Slaves; Modular Slaves is a concept defined by CiA and refers to the usage of a bus coupler that allows connecting several combination of modules (please refer to CiA documentation for further details)
Heartbeat…	not supported

4.2.2.3 Module Parameters

Figure 46 shows the “Module Parameters” tab. It includes the controls to enable CoDeSys to handle CAN controllers when the hardware offers this possibility.
Current implementation of CoDeSys and SCM related hardware does not support this functionality.

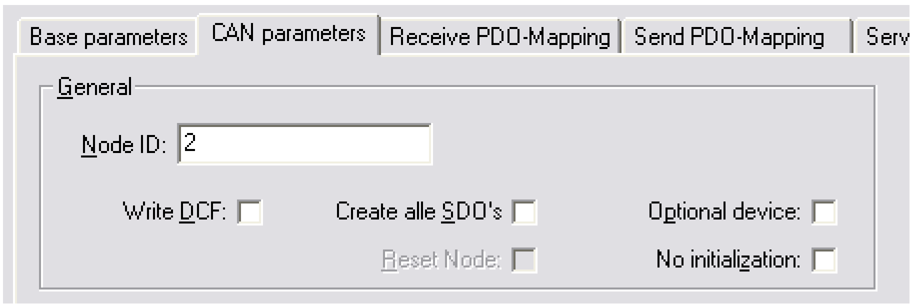


Index	Name	Value	Defa...	Min.	Max.
1	CanPort	0	0		

Figure 46

4.2.3 Settings for CAN Slaves

The configuration for the slaves has a common part, which is independent from the EDS. Figure 47 shows the “CAN Parameters” tab of a generic CAN Slave.



Node ID: 2
Write DCF: ☐
Create alle SDO's ☐
Optional device: ☐
Reset Node: ☐
No initialization: ☐

Figure 47

The meaning of the parameters is the following:

Create all SDO’s	specifies whenever the SDO messages for the slave configuration, dependant on the PDO mapping, should be created for all objects (option activated) or just for the modified objects (option not activated); in this second case, please make sure the EDS file loaded in CoDeSys is matching the hardware device features, otherwise some required SDO messages will be erroneously skipped
No initialization	if activated the sequence of SDO messages required for the device initialization (PDO mapping) will not be created
Optional device	if activated the current device is considered as optional into the bus; at start-up the CAN controller will check if it is present appling the following rules: - if the device is present since start-up and it correctly replies to the CANopen mandatory object “Device Type” query (matching the eds 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.

Note: *“Optinal device” option is supported by Target version V4.01L and greater*

4.3 PLC Programming

The CoDeSys software is based on the IEC 61131-3 standard. It includes the 5 standard programming languages defined in the IEC 61131-3 model:

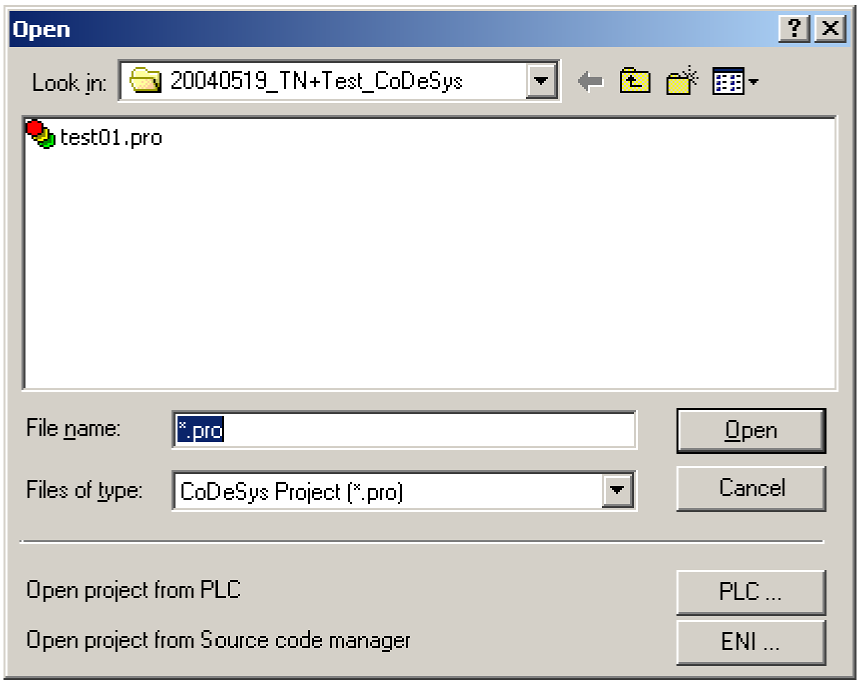
- **SFC**: Status Flow Chart
- **FBD**: Function Block Diagram
- **LD**: Ladder Diagram
- **ST**: Structured Text
- **IL**: Instruction List

In addition the **CFC** Continuous Function Chart language has been included.

Please refer to the CoDeSys documentation for all the necessary information and details about programming languages.

4.4 PLC Project Upload

PLC project Upload is supported in the UniOP implementation.
To upload a project from the SCM module, select the command Open from the File menu and click on the "PLC" button to specify the source of the open action.


Open

Look in: 20040519_TN+Test_CoDeSys

test01.pro

File name: *.pro

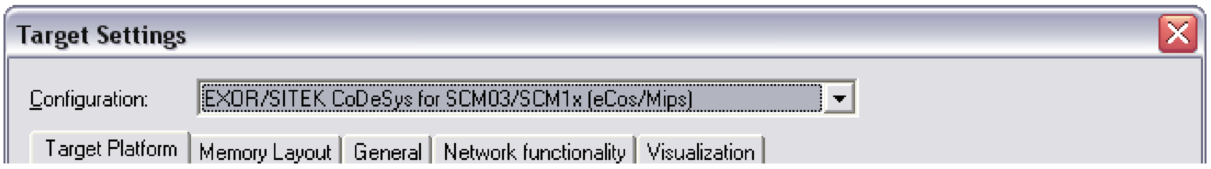
Files of type: CoDeSys Project (*.pro)

Open project from PLC

Open project from Source code manager

Figure 48

When opening a program from PLC, CoDeSys requires the specification of the target settings; they should be as shown in Figure 49.


Target Settings

Configuration: EXOR/SITEK CoDeSys for SCM03/SCM1x (eCos/Mips)

Figure 49

Project upload can be executed only if the project source code has been downloaded to the target device.

The “Source code download” command is available form the “Online” menu of the CoDeSys programming software.

4.5 Programming the CANopen Interface

Connection to special CANopen devices may require direct access to some CAN commands. Function blocks are available for this purpose. This chapter describes the most important cases.

Note: *SCM11-C modules DO NOT have the CANopen interface; this chapter refers ONLY to SCM03-C, SCM05-C and SCM12-C modules*

4.5.1 Access to Remote Variables Using SDO Protocol

The SDO protocol can be used to access any remote variable, defined according to the CANopen standard. The SCM03-C CANopen master works as SDO client and remote nodes are servers.
While in PDO mode the transmission is normally cyclical and automatic, in SDO mode the data exchange is normally single-shot. Each session can normally transfer only one data item. This means that the SDO protocol is much slower than the PDO protocol.

Two Function Blocks are available to configure communication via the SDO protocol.

EXOR_CAN_SDO_RD	read remote variables
EXOR_CAN_SDO_WR	write remote variables

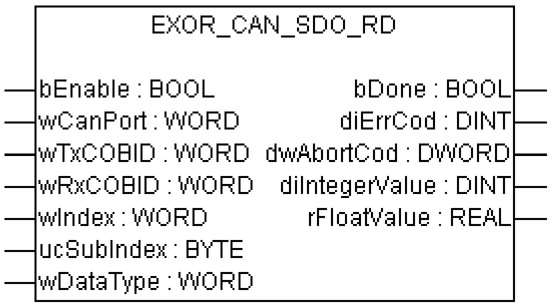


Figure 50

The parameters for the EXOR_CAN_SDO_RD function block are:

wIndex	address of CAN object inside the remote node, as defined by the manufacturer
ucSubindex	address of single variable inside the object
wDataType	one of the data types supported by CAN.
wCanPort	identifies the CAN controller channel
wTxCOBID	specifies the COD ID of the transmit SDO
wRxCOBID	specifies the COD ID of the receiving SDO
bEnable	enable bit

Returned values are:

bDone	flag indicating the operation is executed.
diErrCod	error code of the operation. It is generated by the client and it is valid only after operation has been completed. Value 0 means a successful operation.
dwAbortCode	code sent by the server in case the operation is aborted. It is returned by the function block as received from the remote device, so refer to CAN standard definition or to specific technical description of the server (remote device).
diIntegerValue	value of the read variable in integer format
rFloatValue	value of the read variable in float format

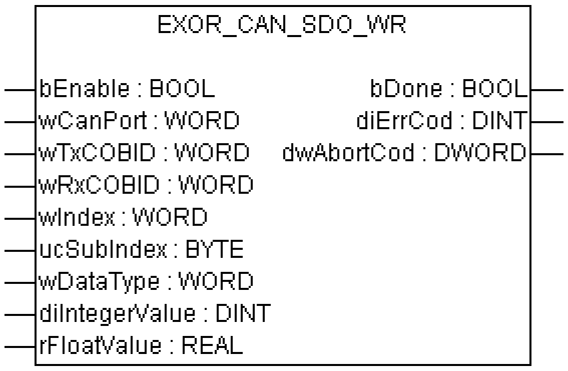


Figure 51

The parameters for the EXOR_CAN_SDO_WR function block are:

bEnable	enable bit
wCanPort	identifies the CAN controller channel
wTxCOBID	specifies the COD ID of the transmit SDO
wRxCOBID	specifies the COD ID of the receiving SDO
wIndex	address of the CAN object inside the remote node, as defined by the manufacturer
ucSubindex	address of single variable inside the object
wDataType	one of the data types supported by CAN.
diIntegerValue	is the value to be written in integer format
rFloatValue	is the value to be written in float format

Returned values are:

bDone	is a flag indicating the operation is executed.
diErrCod	error code of the operation. It is generated by the client and it is valid only after operation is completed. Value 0 means a successful operation.
dwAbortCode	is the code sent by the server in case the operation is aborted. It is reported as received, so refer to CAN standard definition or to specific technical description of the server.

The possible values for the abort code (see return value dwAbortCod) are showed in the table below.

0503 0000h	Toggle bit not alternated.
0504 0000h	SDO protocol timed out.
0504 0001h	Client/server command specifier not valid or unknown.
0504 0002h	Invalid block size (block mode only).
0504 0003h	Invalid sequence number (block mode only).
0504 0004h	CRC error (block mode only).
0504 0005h	Out of memory.
0601 0000h	Unsupported access to an object.
0601 0001h	Attempt to read a write only object.
0601 0002h	Attempt to write a read only object.
0602 0000h	Object does not exist in the object dictionary.
0604 0041h	Object cannot be mapped to the PDO.
0604 0042h	The number and length of the objects to be mapped would exceed PDO length.
0604 0043h	General parameter incompatibility reason.
0604 0047h	General internal incompatibility in the device.
0606 0000h	Access failed due to an hardware error.
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist.
0609 0030h	Value range of parameter exceeded (only for write access).
0609 0031h	Value of parameter written too high.
0609 0032h	Value of parameter written too low.
0609 0036h	Maximum value is less than minimum value.
0800 0000h	general error
0800 0020h	Data cannot be transferred or stored to the application.
0800 0021h	Data cannot be transferred or stored to the application because of local control.
0800 0022h	Data cannot be transferred or stored to the application because of the present device state.
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error).

Allowed values for data type (refer to input parameter wDataType) are listed below.

0001	BOOLEAN
0002	INTEGER8
0003	INTEGER16
0004	INTEGER32
0005	UNSIGNED8
0006	UNSIGNED16
0007	UNSIGNED32
0008	REAL32

5 Internal Controller Hardware Manual

This chapter describes some implementation-specific issues in the CoDeSys kernel developed for use with the SCM03-C controller module.

5.1 The CAN Interface

The SCM03-C/ SCM05-C/ SCM12-C controller modules 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

Bit timing is done in a sophisticated way to allow fine-tuning of real systems in case of communication problems.
Programming the parameter baudRateKbps at 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). It is currently not used in SCM03-C where the bus line is always sampled once.

5.2 Watchdog

The SCM controller modules includes a watchdog circuit that forces the restart of the module in case of software crashes. The watchdog timer is enabled when the PLC program is running; the control time is fixed and is approximately 1.6 seconds.
The PLC program can re-trigger the watchdog timer calling a special function called WDRESET.

Note *Calling the function WDRESET in a program loop can cause a deadlock condition that will not be resolved by the watchdog system.*

5.3 Timer resolution

The resolution of CoDeSys 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.

5.4 CAN Master Operate Functions (EXOR_CAN_IO_CTRL)

To control special features of CAN master, the Function "EXOR_CAN_IO_CTRL" shown hereafter is used.
This function can act on CanMaster Module, one of the connected CAN slaves or one of the Tx or Rx PDOs of the slaves, depending on the parameters, as explained below.

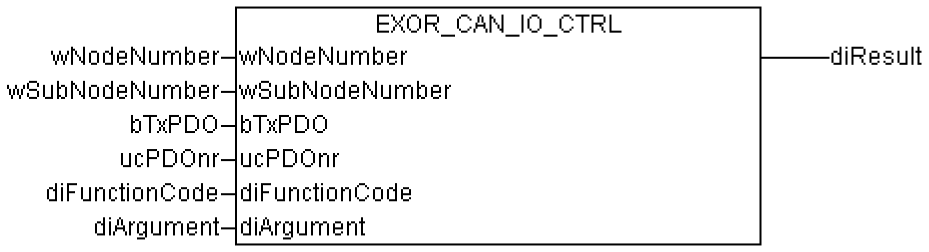


Figure 52

This function can be found in "EXOR_CANopen.lib" file.

The input parameters have the following meaning:

wNodeNumber	Each board in the CoDeSys PLC Configuration Editor has a progressive Node Number starting from zero, this is assigned by CoDeSys and can be visible in the board "Base Parameters". This parameter identifies the board to which the "diFunctionCode" and "diArgument" parameters will be passed to. This must be the Node Number of a CanMaster board.
wSubNodeNumber	This is the Node Number of the Sub Element, i.e. the CAN Slave Node appended to the CanMaster to which the IO_CTRL is directed to. If this value is set to 0xFFFF, then the IO_CTRL is directed to the CanMaster Module

	itself.
bTxPDO	This parameter selects if the IO_CTRL is directed to a Tx or Rx PDO of the selected CAN Slave Node. If TRUE then TX PDO is selected. It's unused if parameter ucPDOnr is set to 0xFF.
ucPDOnr	This parameter selects to which PDO the IO_CTRL is directed to. Zero is the first PDO defined for the selected CAN Slave Node. If this parameter is set to 0xFF, then the IO_CTRL is directed to the CAN Slave Node itself
diFunctionCode	This parameters selects the Function to direct to the selected item (Functions Codes are listed in the tables below)
diArgument	This is the parameter of the Function, its meaning varies from Function to Function.

The Operate Function returns the following output:

diIoCtrlResult	This is the return value, its meaning varies depending on the selected function.

The following tables show the various Functions Codes for EXOR_CAN_IO_CTRL, depending if the IO_CTRL is directed to CanMaster, CAN Slave Node, Rx PDO or Tx PDO, a different set of Functions is possible.

EXOR_CAN_IO_CTRL Functions Codes List Functions Code Directed to CanMaster Module itself (wSubNodeNumber == 0xFFFF)		
Function Code	**Argument**	**Description**
1	Any	Return the QuickStatus, i.e. the global status: 0 means all ok, otherwise the error code (for a list of error codes please see below).
2	Any	All the errors are collected in an Error FIFO, so that no error event is lost. This operate function fetches one error from the Error FIFO. The 32 bits integer returned contains the error code (see the list below) in the lower 16 bits and the NodeId in the higher 16 bits. If there are no error events it returns 0.

Table 7

EXOR_CAN_IO_CTRL Functions Codes List Functions Code Directed to CAN Slave Node (ucPDOnr == 0xFF)		
Function Code	**Argument**	**Description**
0	Any	Returns the current status of the device to which the specified I/O variable is connected. 0 means no errors, otherwise the error code is reported (see error codes list below).

Table 8

EXOR_CAN_IO_CTRL Functions Codes List Functions Code Directed to TX PDOs (bTxPDO TRUE)		
Function Code	**Argument**	**Description**
5	Val	Sets the transmission frequency (expressed in number of Scansions), by default a PDO is sent every scansion.
7	Val	Set new CAN ID (11 bits) and LENGHT of PDO. The passed Value must contain in lower 16 bits the new CAN ID, in upper 16 bits the new LENGTH.

Table 9

EXOR_CAN_IO_CTRL Functions Codes List Functions Code Directed to RX PDOs (bTxPDO FALSE)		
Function Code	**Argument**	**Description**
3	Any	The number of received PDOs since last call to this operate is returned.
7	Val	Set new CAN ID (11 bits) of PDO. The passed Value must contain the new CAN ID.
8	Val	Forces to send the selected PDO; PDO selection depends form the **ucPDOnr** parameter.

Table 10

5.5 Error Codes

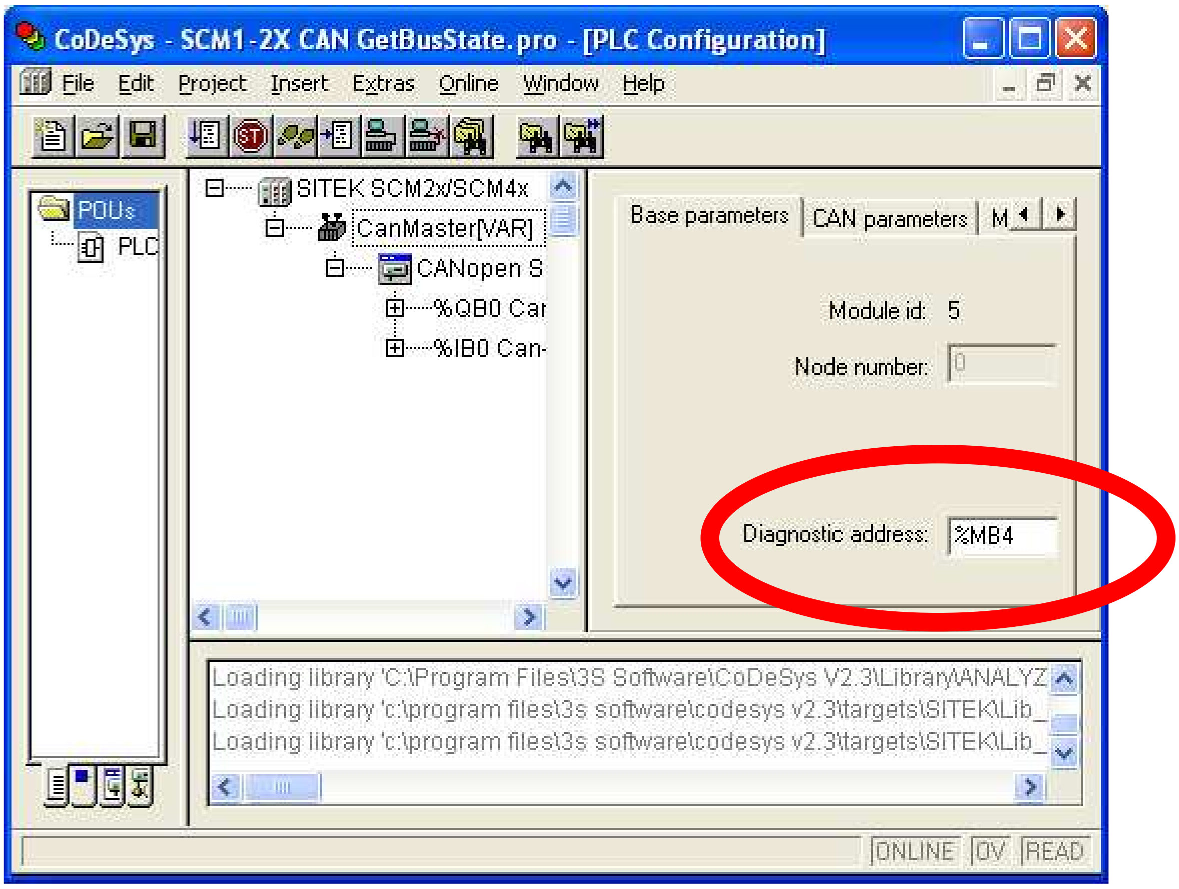
The CoDeSys programming software allows to show in its status bar some diagnostic message related to the PLC operation. The following table contains a list of all the possible error codes. Errors marked as “Fatal” are those preventing the operation of the CANopen I/O driver.

0	OK, no errors
1	EXOR.CANopen.CANCFG: (Fatal) [SlotNr] Invalid baudrate
2	EXOR.CANopen: (Fatal) [SlotNr] Invalid board (perhaps an old board version ?)
3	EXOR.CANopen: (Fatal) [SlotNr] Too many boards defined
4	EXOR.CANopen: (Fatal) [SlotNr] Missing Configuration board (it should be before any CANopen board)
5	EXOR.CANopen: (Fatal) [SlotNr] Invalid Node ID (1..127)
6	EXOR.CANopen: (Fatal) [SlotNr] Invalid PDO Length (0..8)
7	EXOR.CANopen: Node [NodeId] has Guarding Error (toggling bit or status incorrect)
8	EXOR.CANopen: Node [NodeId] is Dead (Node does not reply)
9	EXOR.CANopen: Node [NodeId] restarted (Node is alive again)
10	EXOR.CANopen.CANM????: (Fatal) [SlotNr] Missing preceding CANMICFG or CANMOCFG board (see tech note)
11	EXOR.CANopen.CANM????: (Fatal) [SlotNr] Invalid Offset
12	EXOR.CANopen.CANSDO??: Function Block CANSDORD/WR: Too many SDO/PDO
13	EXOR.CANopen.CANSDO??: Function Block CANSDORD/WR: Invalid parameter/s
14	EXOR.CANopen.CANSDO??: Function Block CANSDORD/WR: Invalid reply from remote SDO server
15	EXOR.CANopen.CANSDORD: Function Block CANSDORD: Returned size differs from requested size
16	EXOR.CANopen.CANSDO??: Function Block CANSDORD/WR: No reply timeout
17	EXOR.CANopen: CAN Error Bus Offstate
18	EXOR.CANopen: CAN Error STAT_ERR_PASSIV
19	EXOR.CANopen: CAN Error STAT_WARN
20	EXOR.CANopen: CAN Error STAT_STUFF_ERR
21	EXOR.CANopen: CAN Error STAT_FORM_ERR
22	EXOR.CANopen: CAN Error STAT_ACK_ERR
23	EXOR.CANopen: CAN Error STAT_BIT_ERR
24	EXOR.CANopen: CAN Error STAT_CRC_ERR
25	EXOR.CANopn: CANOPEN_ERR_SCAN_TOO_FAST: value of Com.CyclePeriod too low or problems on CANbus

Error code 25
In the case the value of Com.CyclePeriod is too low and not all of the PDOs of the previous scan have been transmitted when a new scan is started, the new scan will only send the SYNC message, while new PDOs and NodeGuarding messages will not be transmitted. This is to ensure that all previous PDOs can be transmitted. New scans will only transmit SYNC message until all of the PDOs have been transmitted.

5.6 CAN Master Diagnostic Information

When adding a “CAN Master” board in the PLC Configuration, an area of 144 bytes is allocated starting at the indicated diagnostic address, as shown in the next picture:

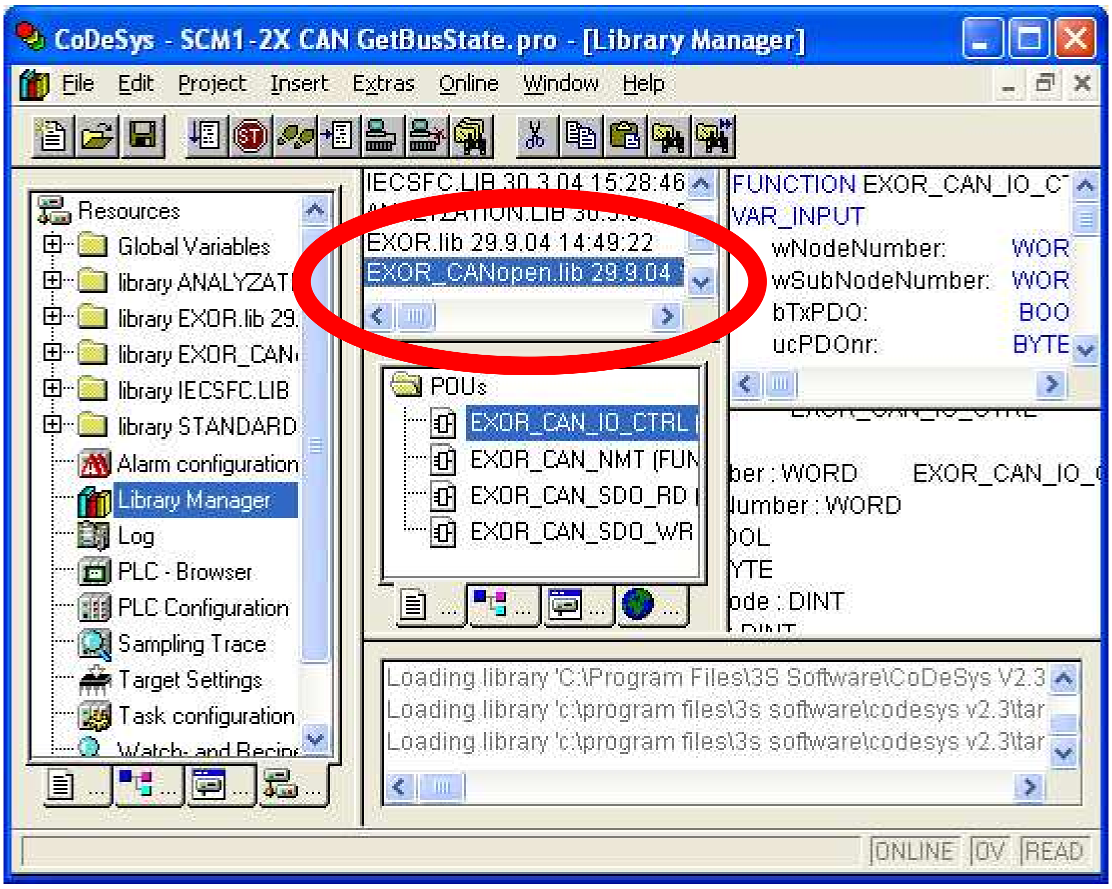

CoDeSys - SCM1-2X CAN GetBusState.pro - [PLC Configuration]
File Edit Project Insert Extras Online Window Help
POUs
PLC
SITEK SCM2x/SCM4x
CanMaster[VAR]
CANopen S
%QB0 Car
%IB0 Can-
Base parameters
CAN parameters
M
ONLINE
OV
READ

Module id: 5
Node number: 0
Diagnostic address: %MB4

```
Loading library 'C:\Program Files\3S Software\CoDeSys V2.3\Library\ANALYZ
Loading library 'c:\program files\3s software\codesys v2.3\targets\SITEK\Lib_
Loading library 'c:\program files\3s software\codesys v2.3\targets\SITEK\Lib_
```

Figure 53

When the “CAN Master” board is added to your project, AFTER first compilation, the library “EXOR_CANopen.lib” is automatically included, as shown in the next picture:



```
FUNCTION EXOR_CAN_IO_C
VAR_INPUT
    wNodeNumber:        WOR
    wSubNodeNumber:     WOR
    bTxPDO:             BOO
    ucPDOnr:            BYTE
```

```
Loading library 'C:\Program Files\3S Software\CoDeSys V2.3
Loading library 'c:\program files\3s software\codesys v2.3\tar
Loading library 'c:\program files\3s software\codesys v2.3\tar
```

Figure 54

Inside this library the **GetBusState user data type** is defined, as reported below (detailed description of the various fields is given further below):

```
TYPE GETBUSSTATE :
STRUCT

        BOLDENABLE : BOOL;
        ENABLE: BOOL;
        DRIVERNAME:POINTER TO STRING;
        DEVICENUMBER:INT;
        READY:BYTE;
        STATE:INT;
        EXTENDEDINFO:ARRAY[0..129] OF BYTE;
END_STRUCT
END_TYPE
```

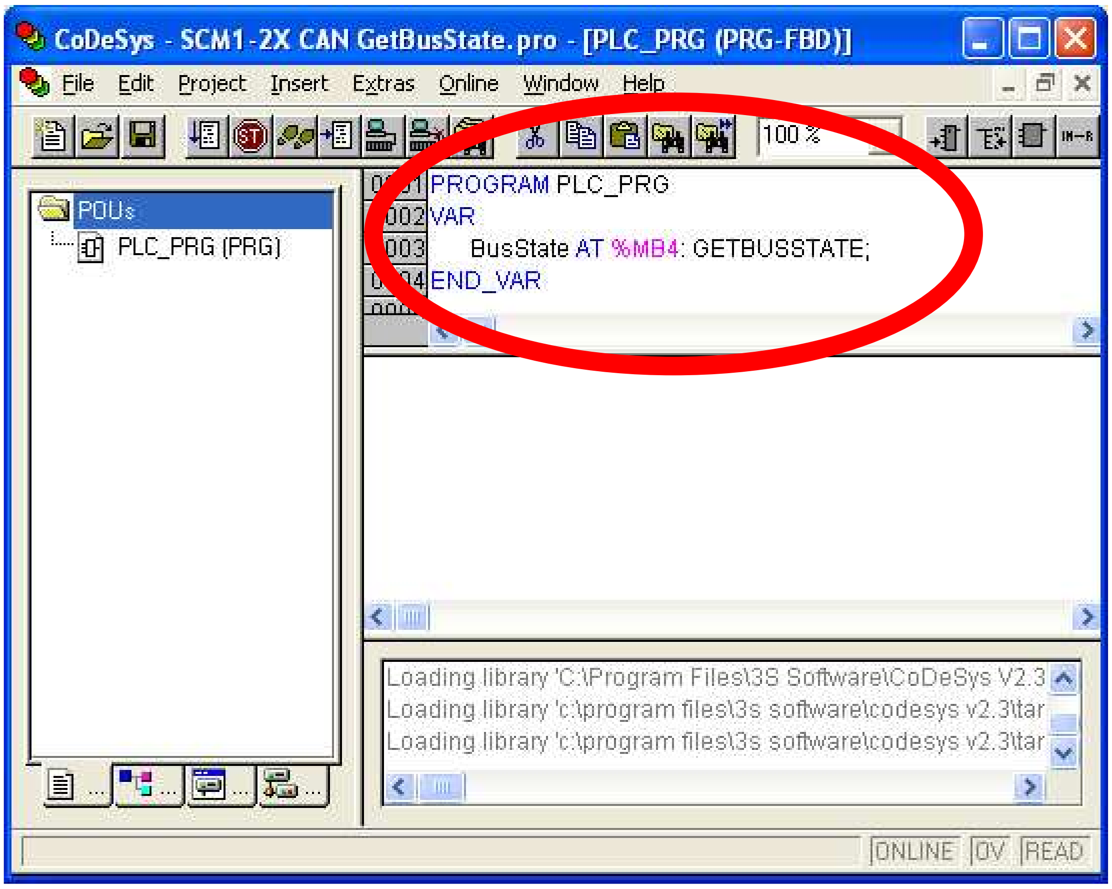
where:

BOLDENABLE	Always TRUE
ENABLE	Always TRUE
DRIVERNAME	"CANopen Master"
DEVICENUMBER	CanPort choosen in the CANMaster configuration
READY	TRUE if running
STATE	Quick Status: 0 (zero) means OK, other values are error codes, see “ERROR CODES” table
EXTENDEDINFO	Elements 0, 128 and 129 are not used. Elements 1 to 127 reports the state of nodes 1 to 127.

	Meaning of the bits of each byte is: Bit 0: Bus module exists in PLC configuration. Bit 1: Bus module is available in bus system. Bit 2: Bus module reports error. Bit 3: Node is initialized and without errors (i.e. it's OFF during initialization and configuration).

Table 11

In the user application can be included a variable of this type which must point at the "Diagnostic Address" of the "CAN Master" board, as shown in Figure 55.


CoDeSys - SCM1-2X CAN GetBusState.pro - [PLC_PRG (PRG-FBD)]
File Edit Project Insert Extras Online Window Help
100 %
POUs
PLC_PRG (PRG)
ONLINE OV READ

```
PROGRAM PLC_PRG
VAR
    BusState AT %MB4: GETBUSSTATE;
END_VAR
```

```
Loading library 'C:\Program Files\3S Software\CoDeSys V2.3
Loading library 'c:\program files\3s software\codesys v2.3\tar
Loading library 'c:\program files\3s software\codesys v2.3\tar
```

Figure 55

This "GetBusState" user data type complies to the CoDeSys standard method of getting diagnostic information of fieldbuses I/O as explained in the CoDeSys help.