Control with UniOP

This manual contains detailed information on the new HMIcontrol option available in UniOP family of HMI products. HMIcontrol is a totally new concept of integrating a state-of-the-art HMI system and a powerful softlogic controller.

Contents

1. Overview

All UniOP HMI products can be transformed into HMIcontrol products adding an optional control module. HMIcontrol is the new term describing the integration of the UniOP family of HMI products with a powerful IEC-1131 softlogic controller.
A UniOP panel running the control software can still be connected to a normal PLC over a serial connection. It is also possible to transfer information between the internal controller and the PLC connected externally.
Turning a UniOP into HMIcontrol is as simple as plugging a control module in the unit.

This manual describes the HMIcontrol system. It covers the SCM03 control module with the ISaGRAF softlogic system, the UniOP local I/O subsystems and the CANopen distributed I/O interface.
This manual is not intended to be a manual for ISaGRAF programming. Refer to the appropriate documentation for this.

1.1 UniOP and Control Product Description

The brain of the HMIcontrol is the new SCM03 PLC module. It is a small module designed to plug in the same socket as the TCMxx communication modules.
It is compatible with **all** UniOP panels with firmware 4.24 or higher.
The SCM03 is powered by a high performance 32 bit MIPS microprocessor and includes a CANopen bus interface.
The SCM03 has 128 KB of Flash memory for PLC program storage and has a typical scan time of 3 ms/K.
The control logic is a full implementation of the IEC1131-3 specification. Advanced features such as SFC programming are included. Flow chart programming is also supported, although it is not part of the IEC standard.

CPU	32 bit MIPS RISC processor
Clock speed	24Mhz
Flash memory	1 MB
SDRAM memory	2 MB
CPU supervision (Reset, Watchdog)	Yes
Interface	CAN interface with optical isolation

Table 1

1.2 System Configurations

You can apply the HMIcontrol systems in different configurations. The possible configurations

1.2.1 Compact Stand-alone Controller

The HMIcontrol system can be used to build very compact standalone HMI and PLC system. I/O is provided via the internal UIM modules.

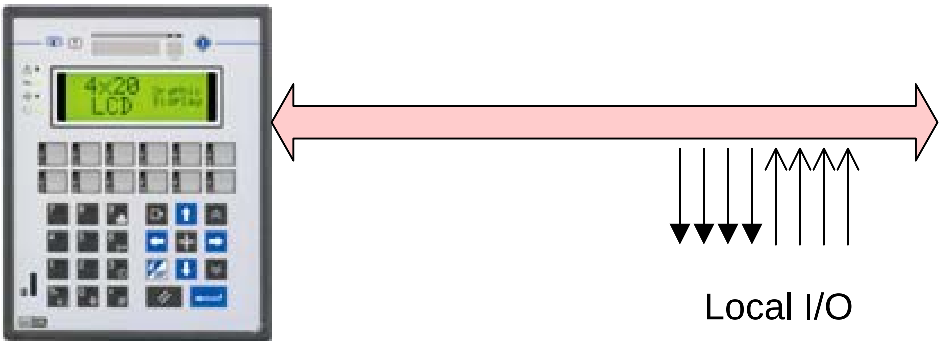


Figure 1

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.

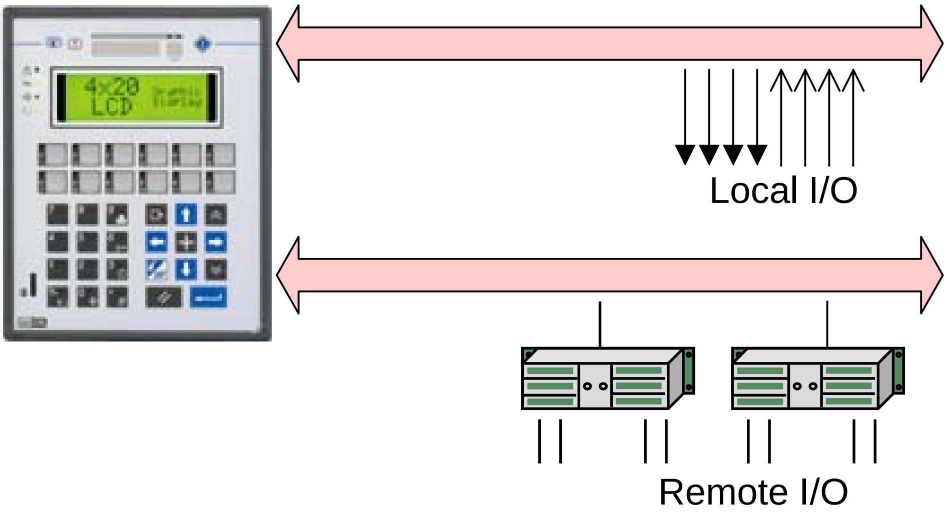


Figure 2

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).

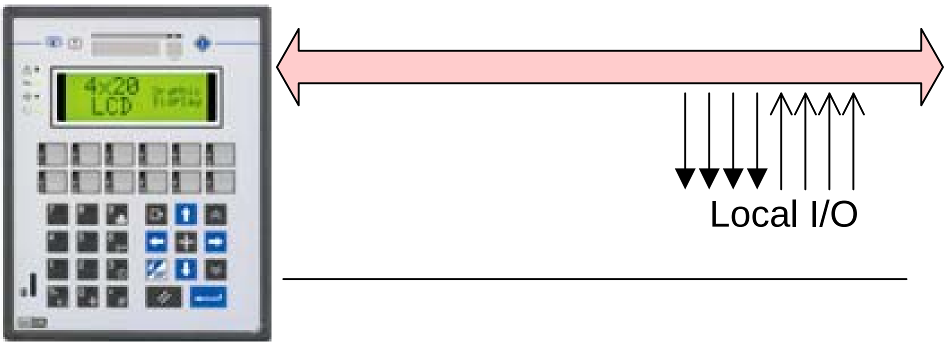


Figure 3

1.3 Technical Specifications

The table below reports the main technical data of the SCM03 control module running the ISaGRAF software.

Max Memory for Application Program	128KB
Max Number of Variables	65535 Boolean 65535 Analog (32bit) 65535 Timer 65535 Messages 4095 FB Instances
Max Number of SFC Steps	4095
Max Number of Function Blocks	255 IEC FB 255 IEC Functions 255 C FB 255 C Functions
Max Number of I/O Boards	256
Max Number of I/O Signals	128 per Board
Execution Speed (average)	Boolean AND 3 ms/K Integer ADD 3 ms/K Float ADD 10 ms/K

Table 2

1.4 Requirements and Limitations

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

	Version
UniOP Designer	5.07 or higher
UniOP Firmware	4.24 or higher
ISaGRAF	3.30 or higher

Table 3

Hardware types –0045 and –0050 have only one slot available and if one SCM module is used, it is no longer possible to plug one TCM module.
Use of local I/O is limited to certain UniOP models. Complete list will be in Appendix.

2. Getting Started

The following chapter describes the basic steps to follow in order to get running the HMIcontrol system using the SCM03 Control Module.

2.1 Installing the SCM03 Control Module

The SCM03 Control Module must be installed in the socket available for normal 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 recognise the new module.

The procedure to install the module is the following:

1. Turn off the operator panel.
2. Unscrew (not completely) with a screwdriver the two screws “A” (shown in Figure 4) attaching the rear cover.
3. Remove the rear cover.
4. Plug the module in the red connectors and make sure they 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.

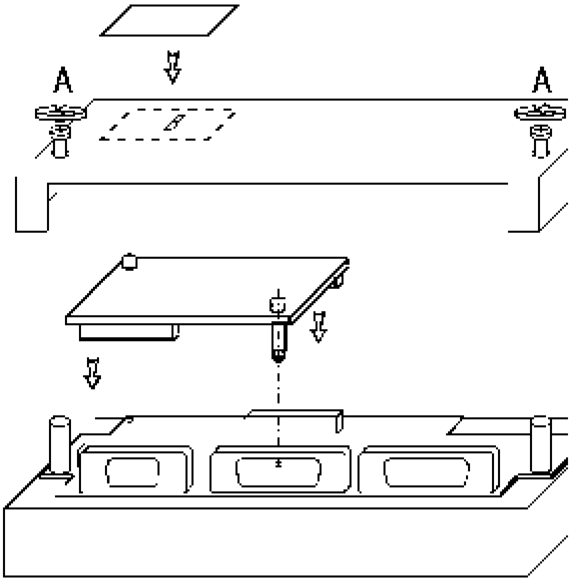


Figure 4

2.2 Basic Control Module Diagnostic

The System Menu of UniOP provides some 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:

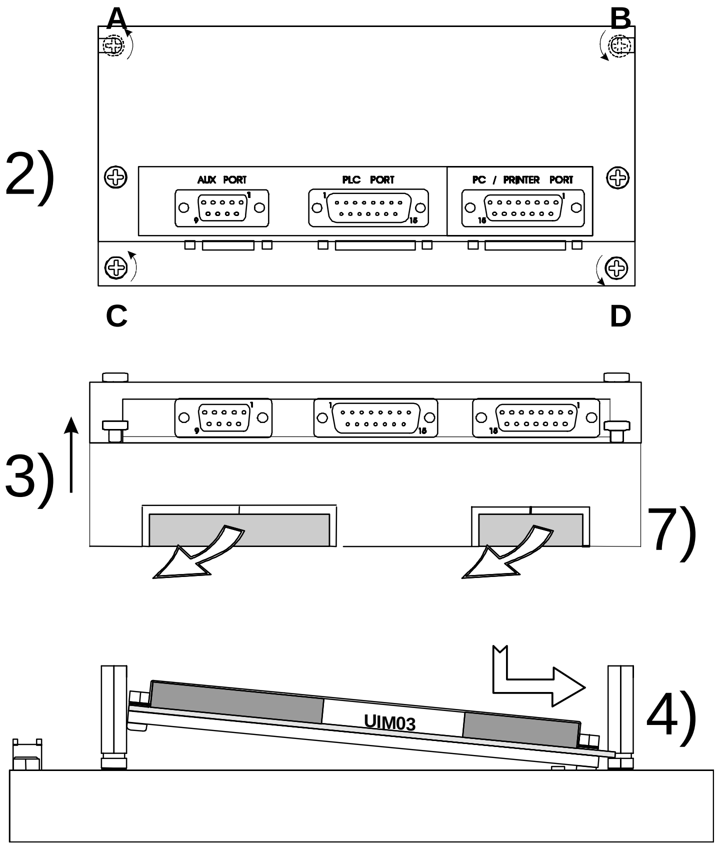
```
SCM03 0 H160 X130 OK
```

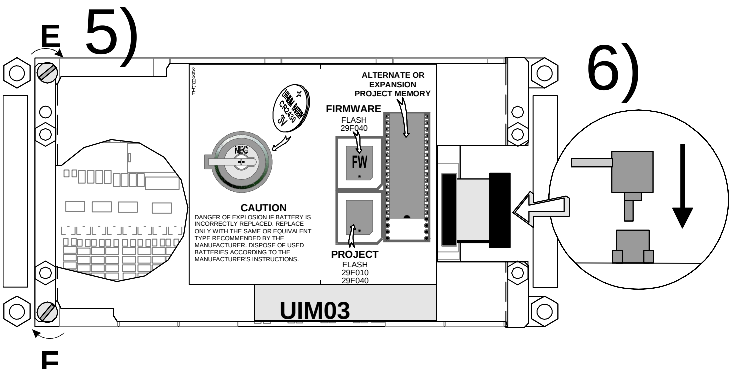
`SCM03`	type of module
`0 H160 X130`	internal version codes
`OK`	confirms the correct installation of the control module.

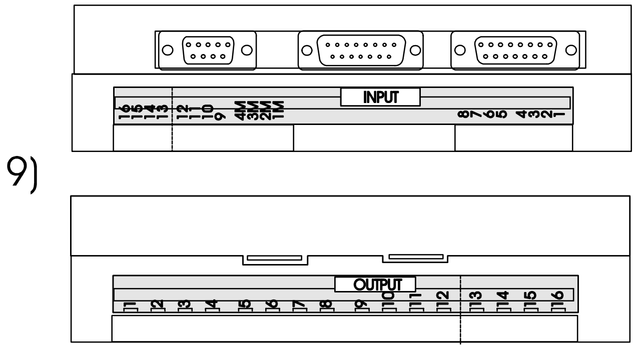
2.3 Installing an I/O Module

Internal I/O modules can only be installed if the operator panel has 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, 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 UIM03 pins assignment.




5)
6)
E
F
ALTERNATE OR
EXPANSION
PROJECT MEMORY
FIRMWARE
FLASH
29F040
FW
LITHIUM BATTERY
CR2430
3V
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



2.4 Basic I/O Module Diagnostic

The System Menu of UniOP provides some diagnostic information on the operation of the communication and control modules.
To view the diagnostic information:

5. Make sure the operator panel is in Operation Mode
6. Recall the System Menu
7. Scroll down the display to show the bottom row of the page
8. 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.

2.5 Installing the ISaGRAF Workbench

The ISaGRAF Workbench version 3.3 or later must be used to program the ISaGRAF controller in the HMIcontrol system.
ISaGRAF comes with an installation procedure that installs the package and all necessary libraries.

The UniOP I/O libraries and the UniOP symbol translator are installed automatically by the standard installation procedure. They will have to be installed manually if a standard ISaGRAF package is used or a new I/O module is to be added to the list. How to use a standard ISaGRAF package is described later in this technical note.

2.5.1 Hardware and Software Requirements for ISaGRAF Workbench

The ISaGRAF Workbench can be installed on any computer meeting the minimum qualification for Windows 3.1. However, the following minimum hardware is recommended for application development:

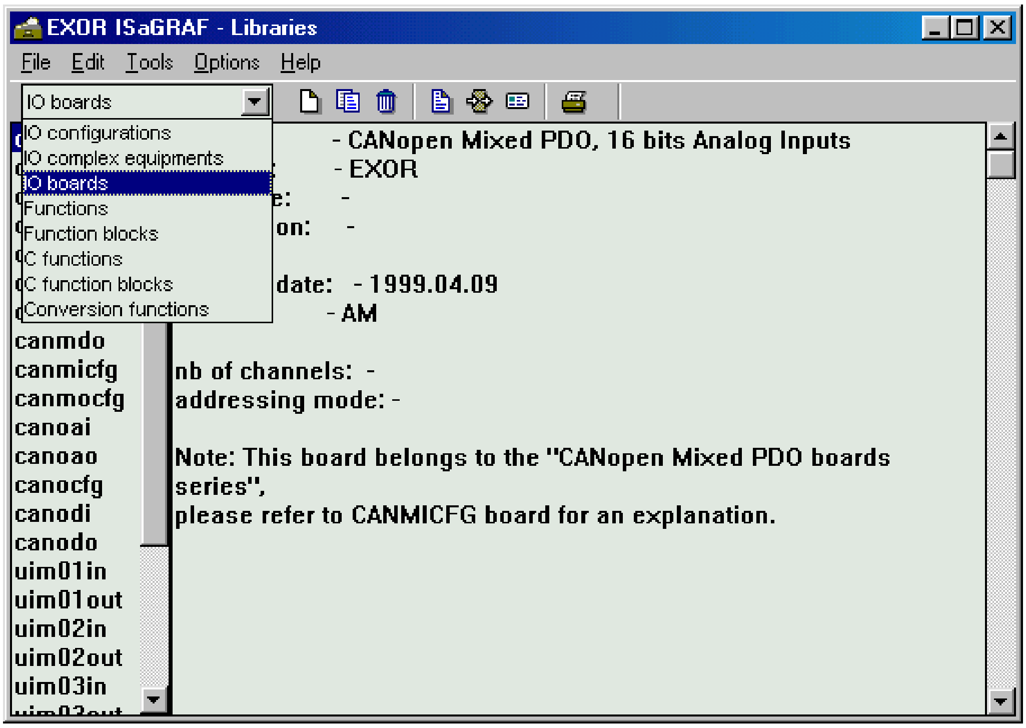
- A PC with a Pentium processor
- 16MB memory
- One 3.5” (1.44MB) disk drive
- One HD with at least 20MB of available space
- A graphic VGA or SVGA adapter and compatible monitor
- A mouse (required for graphic development tools)
- A parallel port (required for the hardware protection key)

2.5.2 Using a Standard ISaGRAF Package

The UniOP Internal Controller can be also programmed with an ISaGRAF package other than the tool provided by EXOR.
In case you need to use a standard ISaGRAF package, the installation of the UniOP libraries for the I/O boards and the tag converter must be done manually.
Additionally the file **ISATXT.ERR** should be copied from the EXOR Installation CD to the EXE sub-directory of the standard ISaGRAF installation tree.

2.5.2.1 Installing the UniOP I/O Library

The EXORARK directory provides all the libraries for I/O boards that have to be installed in a standard ISaGRAF package. They are stored in a “archived” format and have to be restored with the Libraries tool.
Select ‘Libraries’ from the Tools menu of the Project Management software. The Libraries tool is shown in Figure 5.
From the selection list on the left part of the dialog box select ‘I/O boards’ to restore the UniOP boards.



canmdo
canmicfg
canmocfg
canoai
canoao
canocfg
canodi
canodo
uim01in
uim01out
uim02in
uim02out
uim03in

- CANopen Mixed PDO, 16 bits Analog Inputs
- EXOR
e: -
on: -

date: - 1999.04.09
- AM

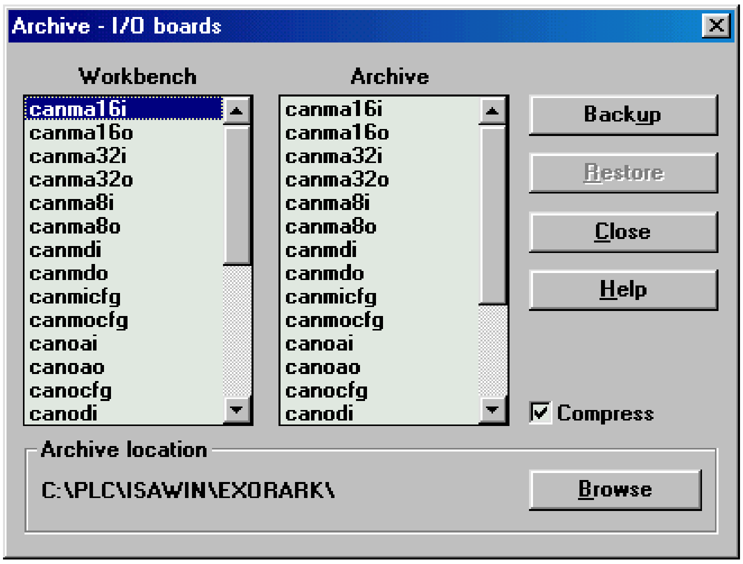
nb of channels: -
addressing mode: -

Note: This board belongs to the "CANopen Mixed PDO boards series",
please refer to CANMICFG board for an explanation.

Figure 5

Select 'Archive' from the Tools menu. The dialog box shown if Figure 6.
The list on the right (Workbench) represents the I/O boards already installed; the list on the right (Archived) shows the I/O board available in compressed format that can be restored and later used in the Workbench.
To restore all the EXOR I/O boards, select all the items in the Archive list and click on the "Restore" button.

The Archive location where the I/O board definition files are stored can be easily reached by clicking on the Browse button.



Workbench	Archive
canma16i	canma16i
canma16o	canma16o
canma32i	canma32i
canma32o	canma32o
canma8i	canma8i
canma8o	canma8o
canmdi	canmdi
canmdo	canmdo
canmicfg	canmicfg
canmocfg	canmocfg
canoai	canoai
canoao	canoao
canocfg	canocfg
canodi	canodi

☑ Compress

Archive location
C:\PLC\ISAWIN\EXORARK\

Figure 6

2.5.2.2 Installing the Tag Generator

The Tag generator for UniOP Designer consists in the file **UDTMTAG.DLL**. This file has to be manually copied in the …\ISAWIN\COM\ directory of the ISaGRAF package.

2.6 Setting-up the Communication

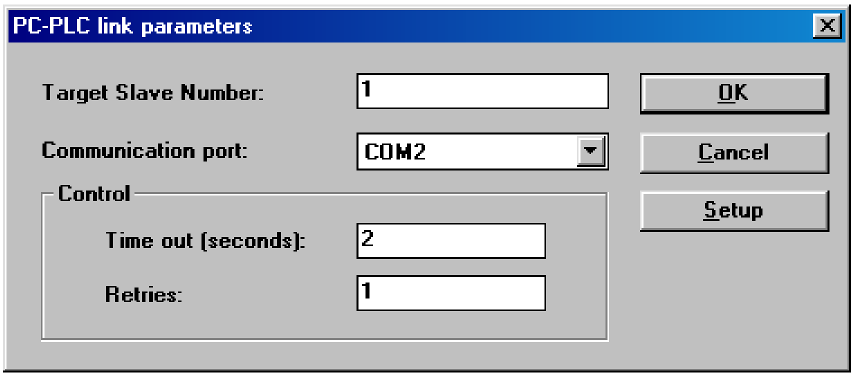
The HMIcontrol system contains two subsystems, the UniOP Operator Panel and the SCM Internal Controller Module. Programming both subsystems will be via the same serial communication link.
The Designer software will be used to program the HMI in the conventional way through the PC/Printer Port.
The ISaGRAF Workbench can communicate with the controller connecting either to the PC/Printer Port or to the PLC Port of the UniOP.
The ISaGRAF Workbench communication set-up is described below.

2.6.1 Selecting the Communication Port for the ISaGRAF Workbench

In the ISaGRAF Program manager, the 'Link Setup' option of the 'Debug' menu, shown below, contains the following parameters:

Target Slave Number it is the communication address assigned to the controller.

Communication port it is the COM port in the PC assigned for the communication between the ISaGRAF Workbench and the HMIcontrol unit.



PC-PLC link parameters

Target Slave Number: 1

Communication port: COM2

Control

Time out (seconds): 2

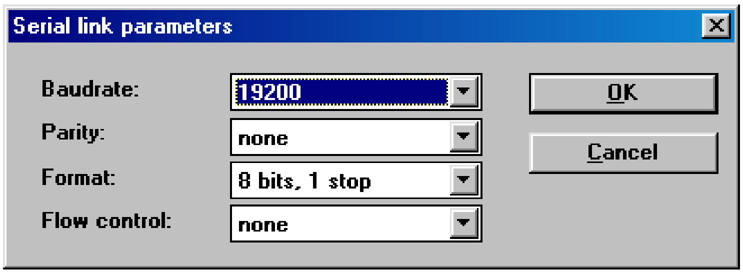
Retries: 1

Figure 7

Click on 'Setup' to see the ISaGRAF Workbench Serial link parameters.

2.6.2 ISaGRAF Workbench Communication Set-up

The default ISaGRAF Serial link parameters are shown in the figure below.



Serial link parameters

Baudrate: 19200

Parity: none

Format: 8 bits, 1 stop

Flow control: none

Figure 8

Note *Do not change the ISaGRAF default communication parameters since they are a constant in the HMIcontrol system.*

2.6.3 Communicating with the Internal Controller

The ISaGRAF Workbench can communicate with the SCM03 controller connecting to the UniOP PLC Port or to the PC/Printer Port.

When UniOP does not contain a valid project, it will remain in Configuration Mode; in this situation the PLC port is assigned by default to the Internal Controller. To program the Internal Controller, before loading a project, the PLC programming cable should be connected to the PLC port.
If the operator panel has no PC/Printer Port, then the PC/PLC Port will be used 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.

When a valid project is present in UniOP and the panel is in Operation Mode, the System Menu will let the user choose the mode of the communication port for the Internal Controller.
Once the System Menu has been displayed on the panel, use the Up/Down arrow keys to scroll to the menu lines where PC/Printer or PLC are highlighted. Now use the Left/Right arrow keys to change the assignment of the selected port.

Two options are available for each UniOP port regarding the Internal Controller: the **Application Mode** and the **Service Mode**.

A port should always be assigned to the Internal Controller in **Application Mode** if it has to be used for normal operations such as ISaGRAF Project downloading and debugging.
The **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 in the PLC row the string “**NOT IN USE**”. 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 when you leave the menu.*

Communication with the UniOP 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 ISaGRAF Workbench to the UniOP Internal Controller.
A gender changer may be required to connect to the UniOP PLC port.

2.6.3.1 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, which 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: Workbench communication 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.

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

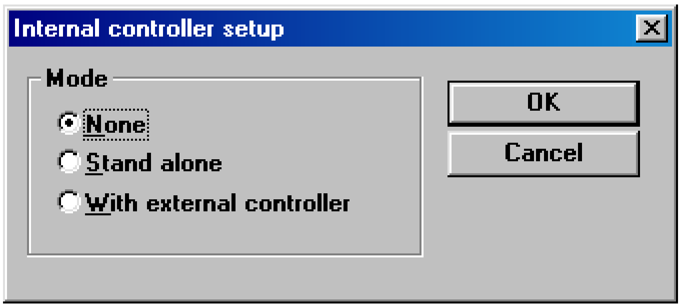
Table 4

3. HMI Programming

UniOP Designer software version 5.07 or greater has to be used to program the HMI panel equipped with the SCM03 module, when the use of the Internal Controller is required.
If the Internal Controller is not used, a previous version of Designer is still compatible with the HMI part of the UniOP.

3.1 Enabling the Internal Controller

UniOP supports three different modes of operation of the Internal Controller.
The mode of operation can be selected from the dialog box ‘Options/Internal Controller Setup…’ shown in figure.



Internal controller setup

Mode

● None
○ Stand alone
○ With external controller

Figure 9

3.1.1 No Internal Controller

If you select ‘None’ from the Internal Controller Setup dialog box, the UniOP HMI panels will not activate the internal controller. UniOP will communicate to an external controller using the communication driver selected in ‘Change Controller Driver…’.
This option can be chosen also when an SCM03 Control Module is present in the system. In this case the Control Module will not be activated.

3.1.2 Stand Alone Operation

Selecting ‘Stand Alone’ from the Internal Controller Setup 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 the SCM03 module is not installed on the panel, the project will not run properly. In addition, using a TCMxx module with a project configured to use the Internal Controller may result in an unexpected behavior.*

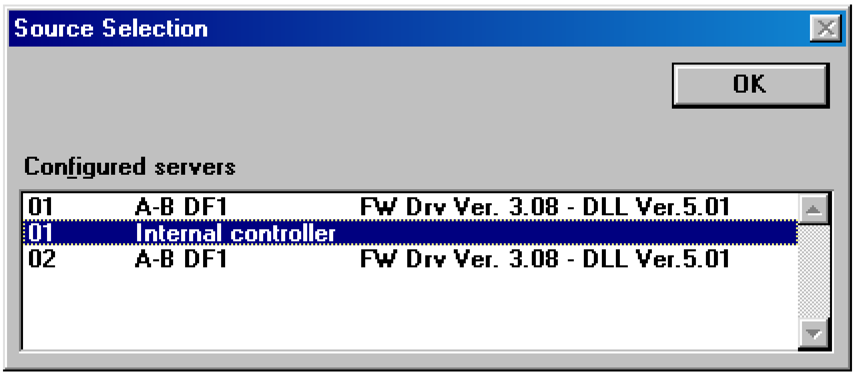
3.1.3 Operation with External Controller

Operation 'With External Controller' is the most general configuration. UniOP is connected to an External Controller via the PLC 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 a communication address. The Source Selection dialog box lets the user choose the location of the requested data.

3.1.4 UniNet and the Internal Controller

The Internal Controller is 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.



Configured servers

01	A-B DF1	FW Drv Ver. 3.08 - DLL Ver.5.01
01	Internal controller	
02	A-B DF1	FW Drv Ver. 3.08 - DLL Ver.5.01

Figure 10

3.2 The Tag Editor

The Tag Editor is a tool integrated with Designer accessible from the Project Menu once the support for the internal controller has been enabled.
Any defined tag in the Editor will allow mnemonic reference to PLC variables.

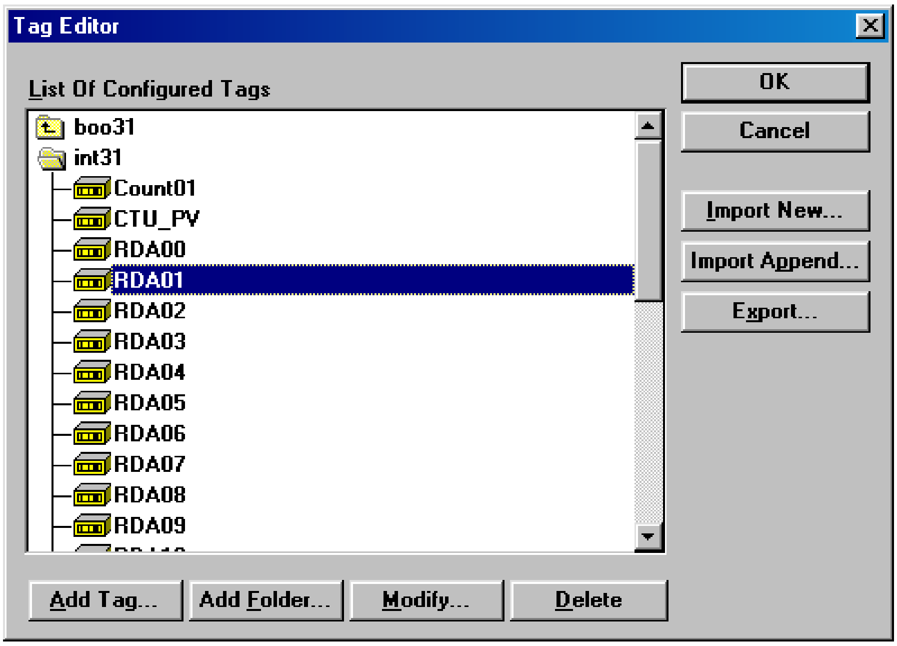


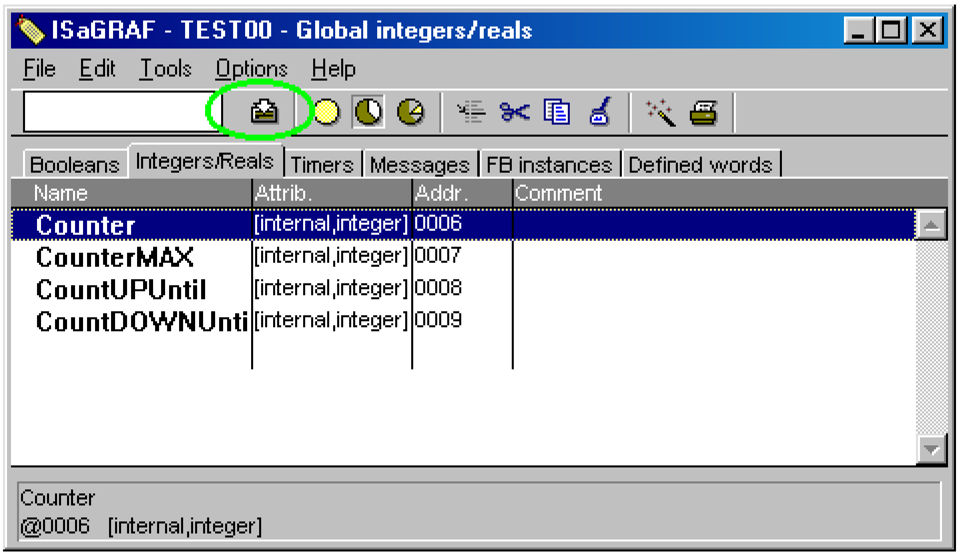
Figure 11

The tag's representation in the tag list is organised in different folders. In case of imported files, folders are created by default, according with the variable data type.
The folder structure can be completely rearranged by the user according with a different logic structure rather being organised by data types.

3.2.1 Importing a Tag File

ISaGRAF for EXOR programming package has an integrated system that provides automatic generation of a binary file formatted with a given syntax, which contains the Tag definition directly interpretable from Designer.

Each time the Dictionary is saved in the ISaGRAF software, a corresponding file, with extension **.utg**, is created (or replaced) in the current ISaGRAF Application folder.



Name	Attrib.	Addr.	Comment
Counter	[internal,integer]	0006	
CounterMAX	[internal,integer]	0007	
CountUPUntil	[internal,integer]	0008	
CountDOWNUnti	[internal,integer]	0009	

Counter
@0006 [internal,integer]

Figure 12

To import the generated tag list into the corresponding Designer project, select the “Import New…” function in the tag editor. The tags will be imported in the Tag Editor by overwriting all the existing identifiers both previously imported and created by the user.

If the file to be imported represents only an additional set of existing tags, the “Append New…” function should be used. The new tag list will be attached to the existing one.

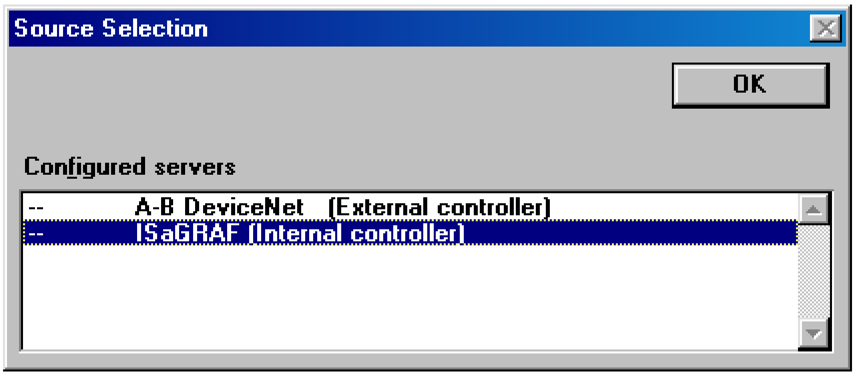
3.2.2 Exporting a Tag File

An exiting collection of tags can be easily exported in a file in case the tags have to be ported to a different Designer project. The “Export” function will create a binary file with the definition of all the existing tags. The format of the Designer Tag File is described later in this technical note.

3.2.3 Creating New Tags

Tags in a project can be both imported from a description file generated by the PLC Programming Software, or created manually.

In case of manual definition of a Tag, just use the “Add Tag…” button; Designer will show the dialog box reported Figure 13 where it has to be selected the Source Reference for the current new tag.



Source Selection

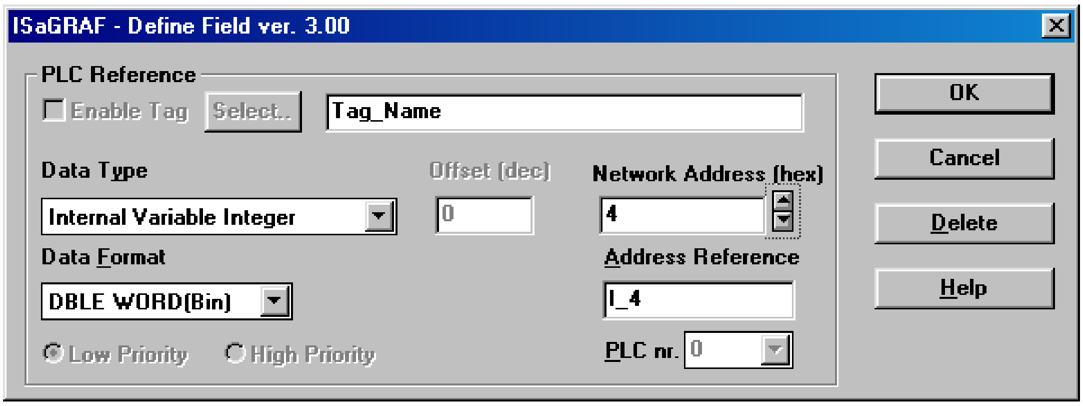
Configured servers

-- A-B DeviceNet (External controller)
-- ISaGRAF (Internal controller)

Figure 13

The dialog box in Figure 13 is displayed only if the Internal Controller is used in combination with an External Controller.

In case the Internal Controller is used with an External PLC and the current tag is defined for ISaGRAF, Designer will then display the Define Field dialog box shown in Figure 14 where the Tag name can be attached to a certain controller reference.



ISaGRAF - Define Field ver. 3.00

PLC Reference

□ Enable Tag   Tag_Name

Data Type: Internal Variable Integer

Offset (dec): 0

Network Address (hex): 4

Data Format: DBLE WORD(Bin)

Address Reference: I_4

● Low Priority ○ High Priority

PLC nr. 0

Figure 14

Tag definition for an External Controller is allowed only if the External PLC supports itself Tag definition. Even if the External Controller does not support Tag Definition its name will be displayed in the Source Selection dialog box. Tag definition for a similar controller can not be applied.
Please refer to EXOR Technical Documentation for further information about drivers that support Tag Definition.

3.2.4 Creating Tag Folders

The organization inside the Tag Editor is done in Folders. As previously described, as result of an Import operation of an **.utg** file created by ISaGRAF, the folders are automatically created to match the different variable types.
According to a better user structure new folders can be created as main folder or as sub-folder like shown in Figure 15.

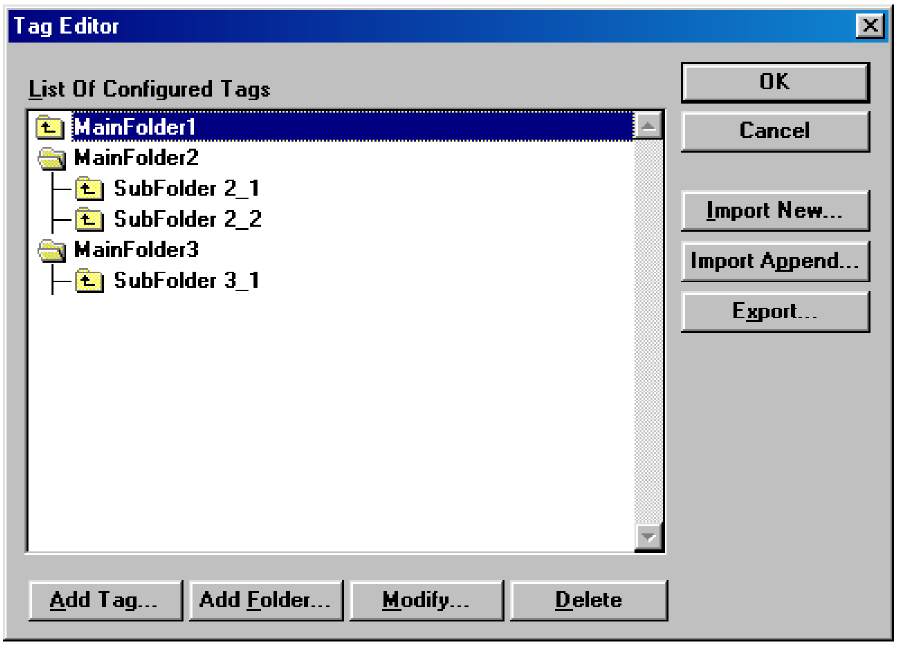


Figure 15

3.2.5 Modifying a Tag or a Folder Name

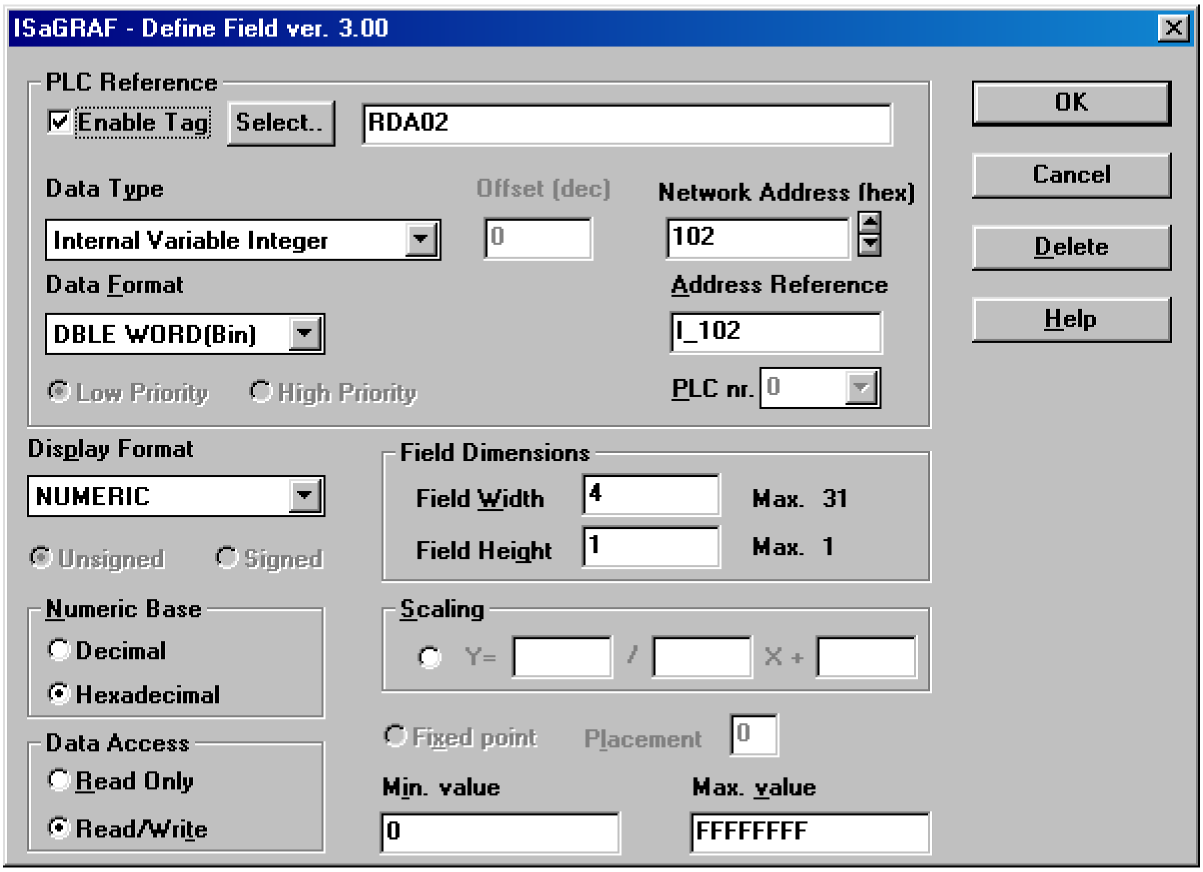
An existing Tag can be modified by double clicking on the Tag name or by selecting the Tag and then using the button “Modify…”.
An existing Folder Name can be modified by double clicking on the Folder name or by selecting the Folder using the “Modify…” button.

3.3 Defining Controller References

The Internal PLC installed on a UniOP panel is conceptually speaking a special UniNet node. Depending on the Internal Controller set-up configured in the dialog box shown in Figure 10 the Define Field dialog box will appear in different mode, but always according to the UniNet configuration style.

3.3.1 Source Selection Dialog Box

If the Internal Controller set-up uses only the internal PLC, the “Define Field” dialog box immediately appears and it shows only the ISaGRAF variables like shown in Figure 16.



ISaGRAF - Define Field ver. 3.00

PLC Reference
☑ Enable Tag: RDA02

Data Type: Internal Variable Integer
Offset (dec): 0
Network Address (hex): 102
Data Format: DBLE WORD(Bin)
Address Reference: I_102
● Low Priority ○ High Priority
PLC nr.: 0

Display Format: NUMERIC
● Unsigned ○ Signed

Field Dimensions
Field Width: 4 Max. 31
Field Height: 1 Max. 1

Numeric Base
○ Decimal
● Hexadecimal

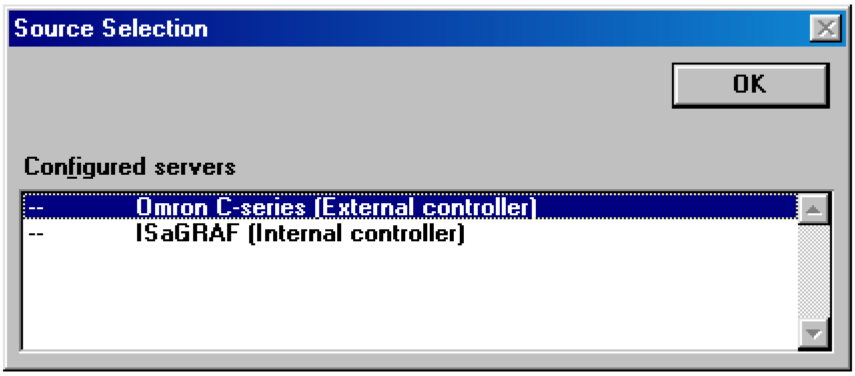
Scaling
○ Y= / X +

Data Access
○ Read Only
● Read/Write

○ Fixed point Placement: 0
Min. value: 0
Max. value: FFFFFFFF

Figure 16

If the Internal Controller set-up is configured to use the combination between the internal PLC and an external controller, the “Source Selection2 dialog box will allow the source selection of the variable to be added like shown in Figure 17.



Configured servers

-- Omron C-series (External controller)
-- ISaGRAF (Internal controller)

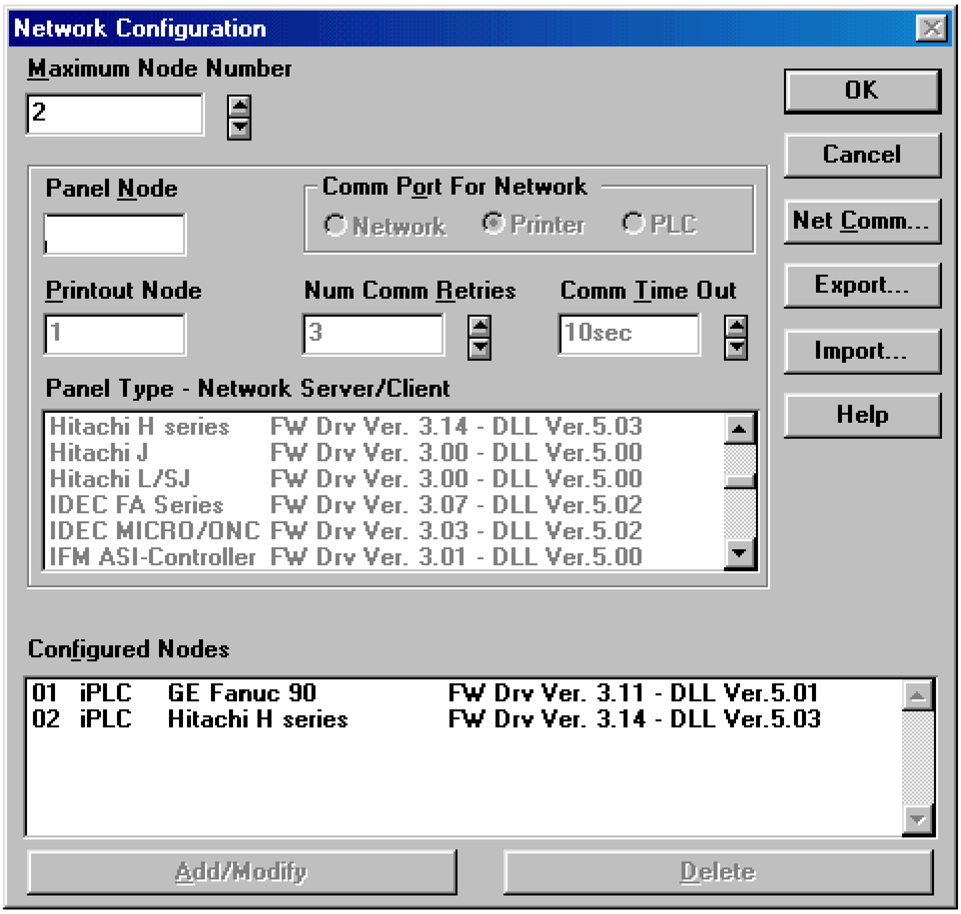
Figure 17

3.3.2 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 Source Selection dialog box.
If the UniNet network has been configured according to Figure 18 where the network has two panels both configured as Server and both configured to use their internal PLC, the “Source Selection” dialog box will appear as shown in Figure 19.

There are four possible sources for a reference to be added:

- a reference from the external controller connected to the UniNet node 1 (GE Fanuc 90)
- a reference from the internal controller of the panel that has UniNet node 1
- a reference from the external controller connected to the UniNet node 2 (Hitachi H series)
- a reference from the internal controller of the panel that has UniNet node 2



Network Configuration

Maximum Node Number: 2

Panel Node:

Comm Port For Network: ○ Network ● Printer ○ PLC

Printout Node: 1

Num Comm Retries: 3

Comm Time Out: 10sec

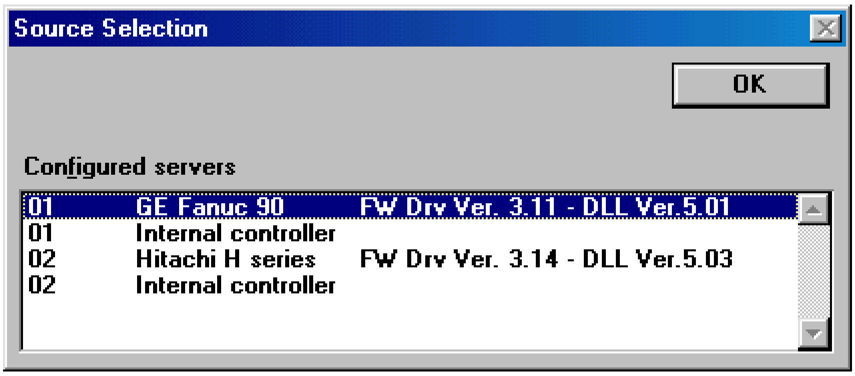
Panel Type - Network Server/Client

Hitachi H series	FW Drv Ver. 3.14 - DLL Ver.5.03
Hitachi J	FW Drv Ver. 3.00 - DLL Ver.5.00
Hitachi L/SJ	FW Drv Ver. 3.00 - DLL Ver.5.00
IDEC FA Series	FW Drv Ver. 3.07 - DLL Ver.5.02
IDEC MICRO/ONC	FW Drv Ver. 3.03 - DLL Ver.5.02
IFM ASI-Controller	FW Drv Ver. 3.01 - DLL Ver.5.00

Configured Nodes

01	iPLC	GE Fanuc 90	FW Drv Ver. 3.11 - DLL Ver.5.01
02	iPLC	Hitachi H series	FW Drv Ver. 3.14 - DLL Ver.5.03

Figure 18



Source Selection

Configured servers

01	GE Fanuc 90	FW Drv Ver. 3.11 - DLL Ver.5.01
01	Internal controller	
02	Hitachi H series	FW Drv Ver. 3.14 - DLL Ver.5.03
02	Internal controller	

Figure 19

3.3.3 Data Field Dialog Box for the Internal Controller

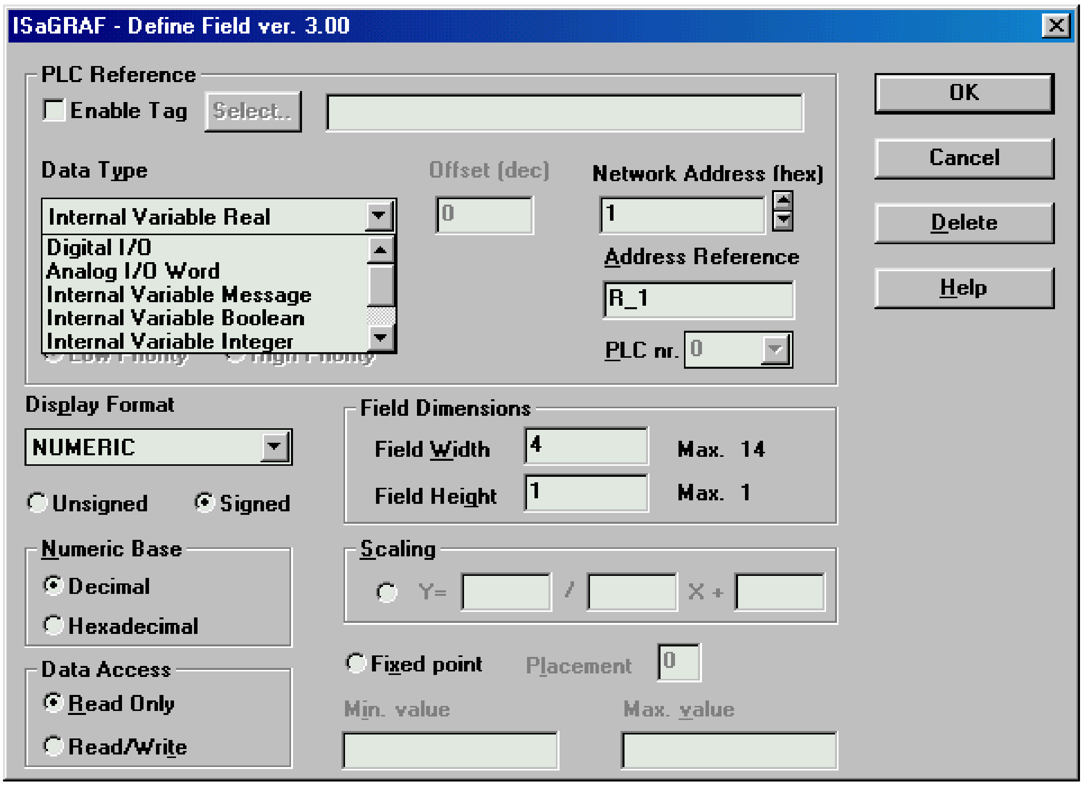
The “Data Field” dialog box for an Internal Controller variable is show in Figure 20.

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

The available Data Types are:

- Digital I/O
- Analog I/O Word
- Internal Variable Message
- Internal Variable Boolean
- Internal Variable Integer
- Internal Variable Real
- Internal Variable Timer

A detailed description for each of them will be presented in the following sections.



ISaGRAF - Define Field ver. 3.00

PLC Reference
□ Enable Tag

Data Type: Internal Variable Real
Digital I/O
Analog I/O Word
Internal Variable Message
Internal Variable Boolean
Internal Variable Integer

Offset (dec): 0
Network Address (hex): 1
Address Reference: R_1
PLC nr.: 0

Display Format: NUMERIC
○ Unsigned ● Signed

Field Dimensions
Field Width: 4 Max. 14
Field Height: 1 Max. 1

Scaling
○ Y= / X +

Numeric Base
● Decimal
○ Hexadecimal

Data Access
● Read Only
○ Read/Write

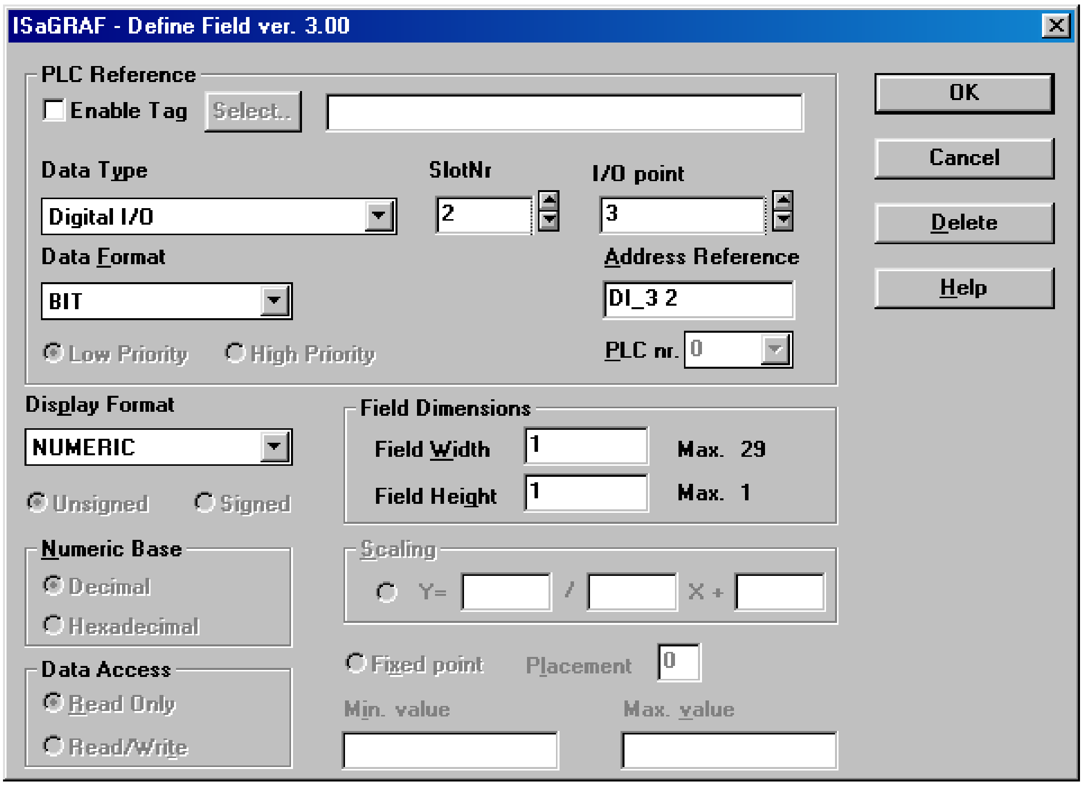
○ Fixed point Placement: 0
Min. value
Max. value

Figure 20

3.3.4 Digital I/O

Digital I/O variables allow to refer directly to the configured I/O points in the ISaGRAF I/O connection tool.

Note *Digital I/O variables are of Read Only type. The Output variables can only be controlled or forced by the ISaGRAF Workbench in the Debug Mode.*



ISaGRAF - Define Field ver. 3.00

PLC Reference
□ Enable Tag   Select..

Data Type: Digital I/O
SlotNr: 2
I/O point: 3
Data Format: BIT
Address Reference: DI_3 2
● Low Priority ○ High Priority
PLC nr. 0

Display Format: NUMERIC
● Unsigned ○ Signed

Field Dimensions
Field Width: 1   Max. 29
Field Height: 1   Max. 1

Numeric Base
● Decimal
○ Hexadecimal

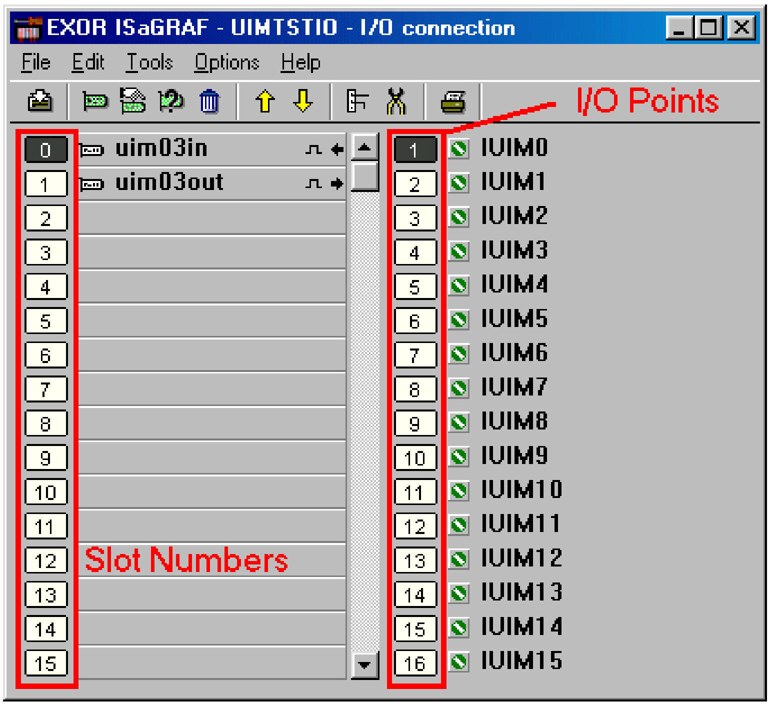
Scaling
○ Y= / X +

Data Access
● Read Only
○ Read/Write

○ Fixed point   Placement 0
Min. value   Max. value

Figure 21

Figure 21 shows the Designer dialog box for these of fields and Figure 22 shows the corresponding information in the ISaGRAF I/O connection tool.



EXOR ISaGRAF - UIMTSTIO - I/O connection

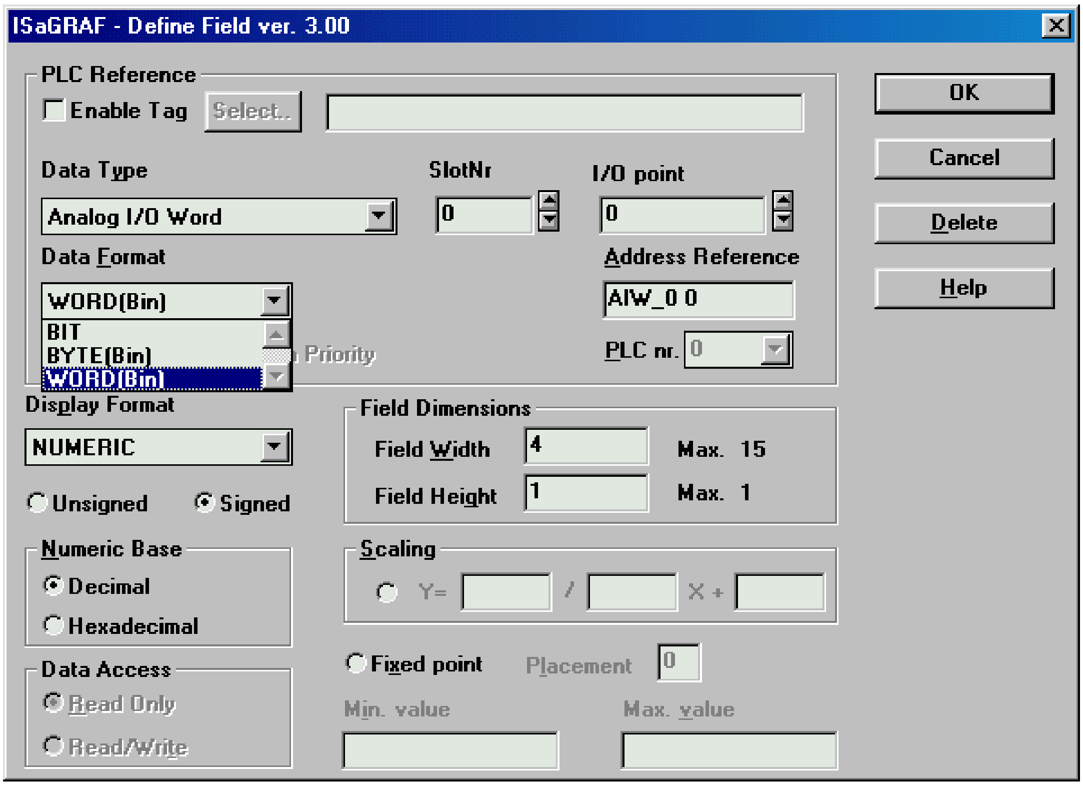
0	uim03in	1	IUIM0
1	uim03out	2	IUIM1
2		3	IUIM2
3		4	IUIM3
4		5	IUIM4
5		6	IUIM5
6		7	IUIM6
7		8	IUIM7
8		9	IUIM8
9		10	IUIM9
10		11	IUIM10
11		12	IUIM11
12		13	IUIM12
13		14	IUIM13
14		15	IUIM14
15		16	IUIM15

Figure 22

3.3.5 Analog I/O Word

Analog I/O variables allow to refer directly to the configured Analog I/O variables in the ISaGRAF I/O connection tool. Analog I/O Word data Type can be referred with 3 different Data Types: Word, Byte and Bit.

Note *Analog I/O variables are of Read Only type. The Output variables can only be controlled or forced by the ISaGRAF Workbench in the Debug Mode.*



ISaGRAF - Define Field ver. 3.00

PLC Reference: □ Enable Tag

Data Type: Analog I/O Word

SlotNr: 0

I/O point: 0

Data Format: WORD(Bin) (list: BIT, BYTE(Bin), WORD(Bin))

Address Reference: AIW_0 0

PLC nr. 0

Display Format: NUMERIC

○ Unsigned ● Signed

Numeric Base: ● Decimal ○ Hexadecimal

Data Access: ● Read Only ○ Read/Write

Field Dimensions: Field Width 4 Max. 15; Field Height 1 Max. 1

Scaling: ○ Y= / X +

○ Fixed point Placement 0

Min. value Max. value

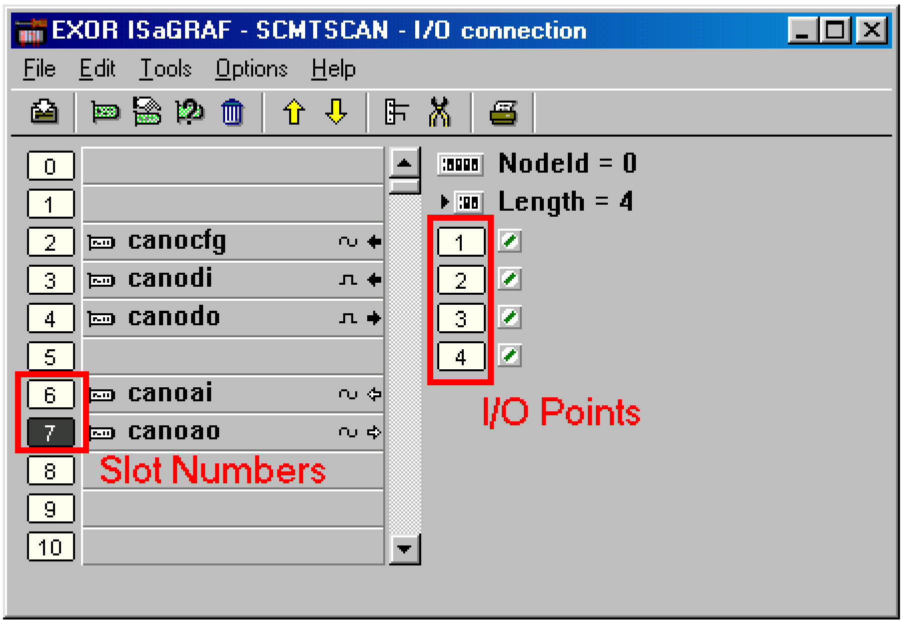
Figure 23

Figure 23 shows the Designer dialog box for these of fields; Figure 24 shows the corresponding information under the ISaGRAF I/O connection tool.

Depending on the Data Format, the I/O point box should be filled with the corresponding information.

If the Data Format is set to BYTE, the I/O point box will show byte index; if the Data Format is set to WORD the I/O point box will show word index; if the Data Format is set to BIT the I/O box contains bit index.

Figure 24 shows the Slot Numbers and the I/O Points as they are displayed in the ISaGRAF I/O Connection tool. The example shows CANopen I/O's.



EXOR ISaGRAF - SCMTSCAN - I/O connection

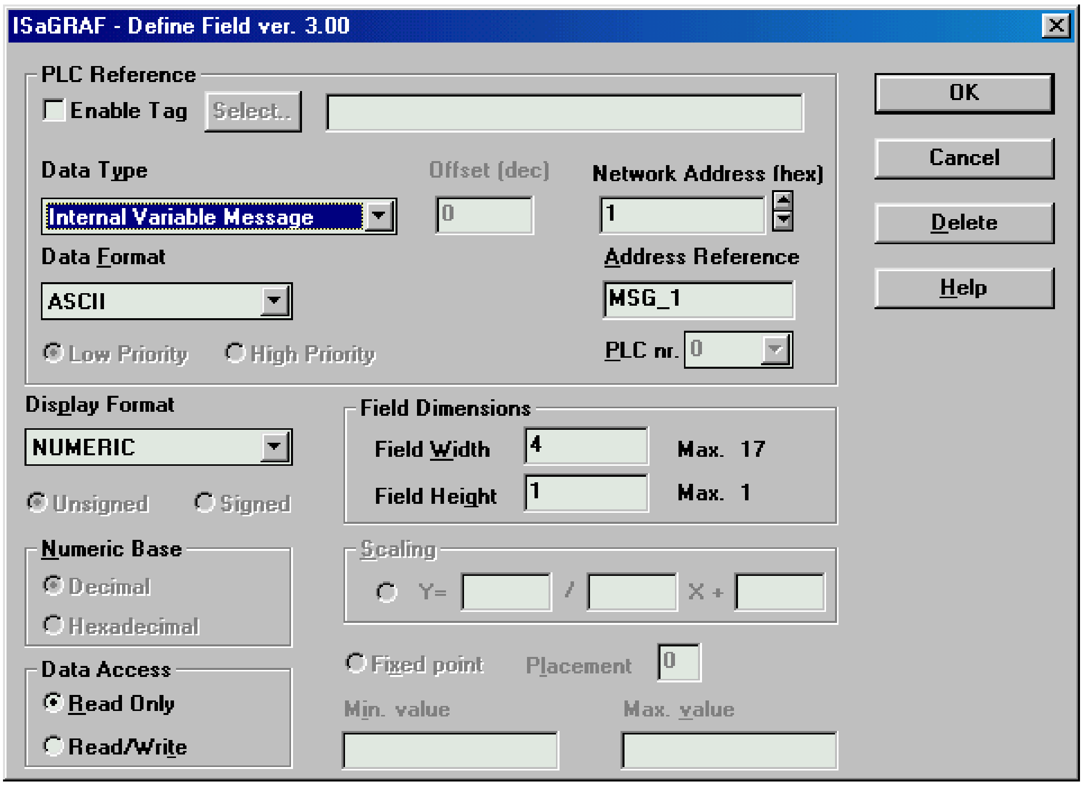
0
1
2 canocfg
3 canodi
4 canodo
5
6 canoai
7 canoao
8
9
10

NodeId = 0
Length = 4
1
2
3
4

Figure 24

3.3.6 Internal Variable Message

Internal Variable Message allow to refer to ISaGRAF variables defined of Message type (see Figure 25) and to which a certain Network Address is assigned.



ISaGRAF - Define Field ver. 3.00

PLC Reference
☐ Enable Tag

Data Type: Internal Variable Message
Offset (dec): 0
Network Address (hex): 1
Data Format: ASCII
Address Reference: MSG_1
● Low Priority ○ High Priority
PLC nr. 0

Display Format: NUMERIC
● Unsigned ○ Signed

Field Dimensions
Field Width: 4 Max. 17
Field Height: 1 Max. 1

Numeric Base
● Decimal
○ Hexadecimal

Scaling
○ Y= / X +

Data Access
● Read Only
○ Read/Write

○ Fixed point Placement 0
Min. value Max. value

Figure 25

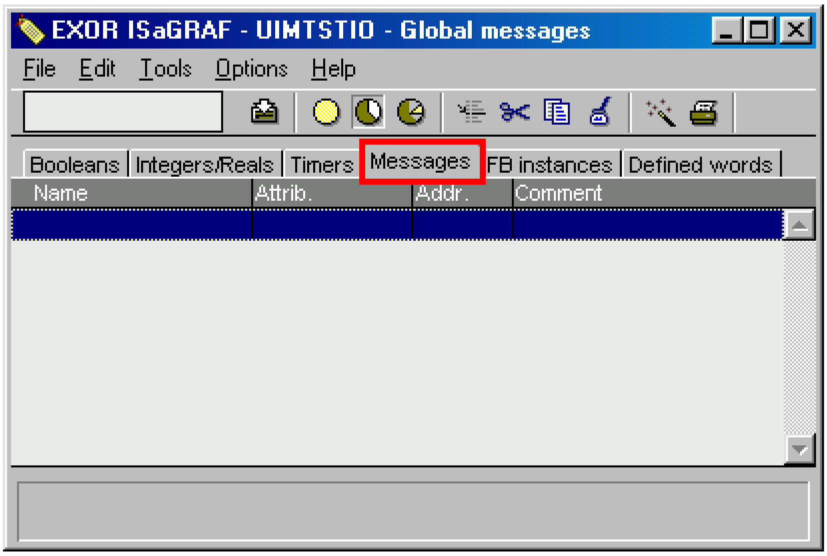
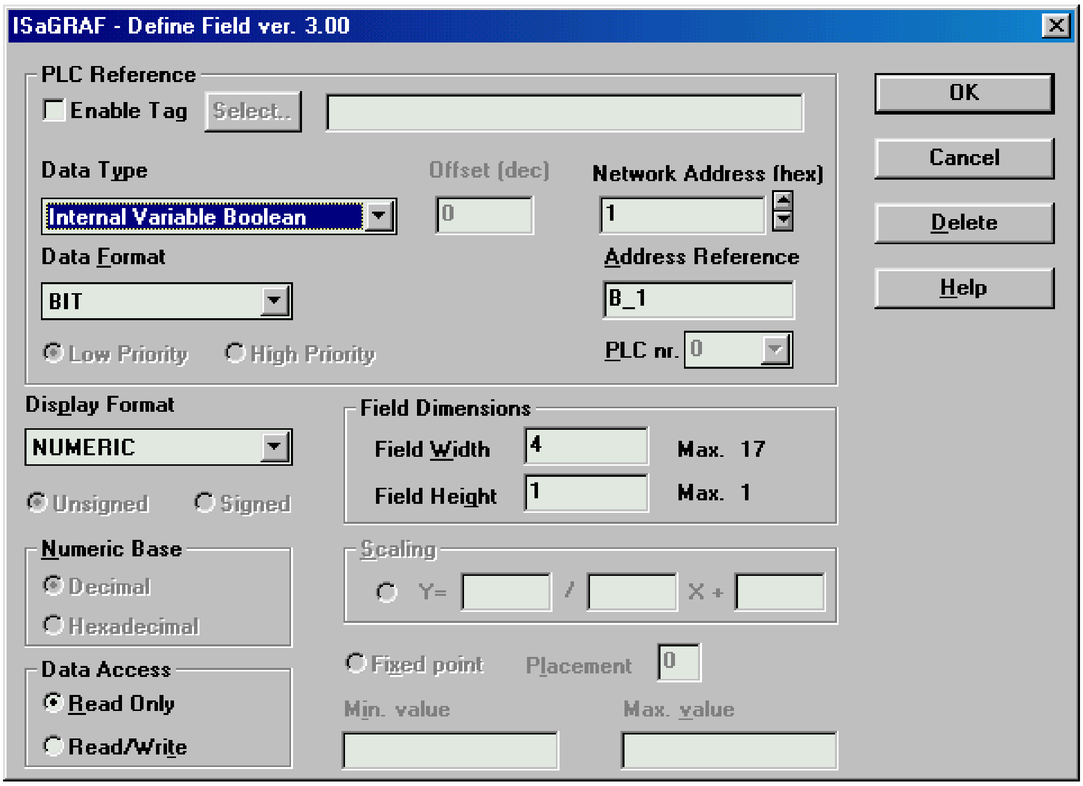


Figure 26

3.3.7 Internal Variable Boolean

Internal Variable Boolean Data type allows to refer to ISaGRAF internal variables declared as Boolean (see Figure 28) and to which a certain Network address is assigned. If a new variable is created in ISaGRAF but is not assigned a network address, it will not be visible in the list of configured tags. This is true for all internal variables type.



ISaGRAF - Define Field ver. 3.00

PLC Reference
☐ Enable Tag

Data Type: Internal Variable Boolean
Offset (dec): 0
Network Address (hex): 1
Data Format: BIT
Address Reference: B_1
● Low Priority ○ High Priority
PLC nr.: 0

Display Format: NUMERIC
● Unsigned ○ Signed

Field Dimensions
Field Width: 4 Max. 17
Field Height: 1 Max. 1

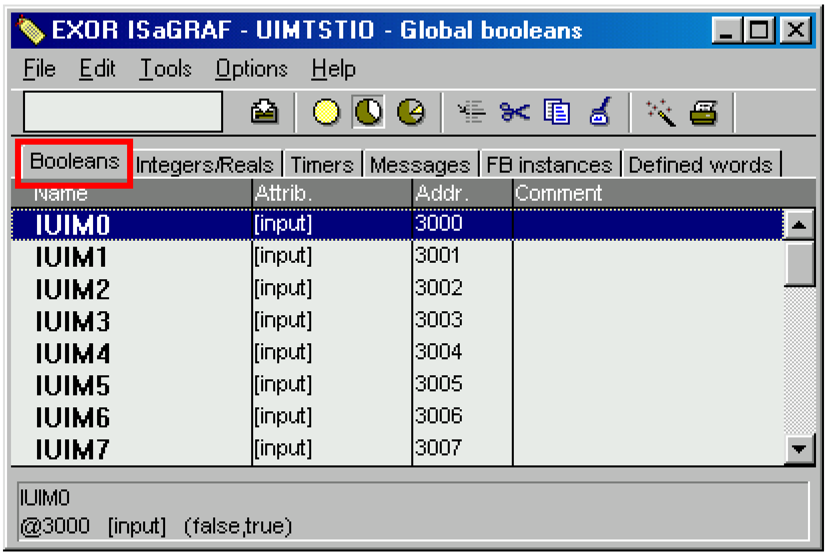
Numeric Base
● Decimal
○ Hexadecimal

Scaling
○ Y= / X +

Data Access
● Read Only
○ Read/Write

○ Fixed point Placement: 0
Min. value
Max. value

Figure 27



EXOR ISaGRAF - UIMTSTIO - Global booleans

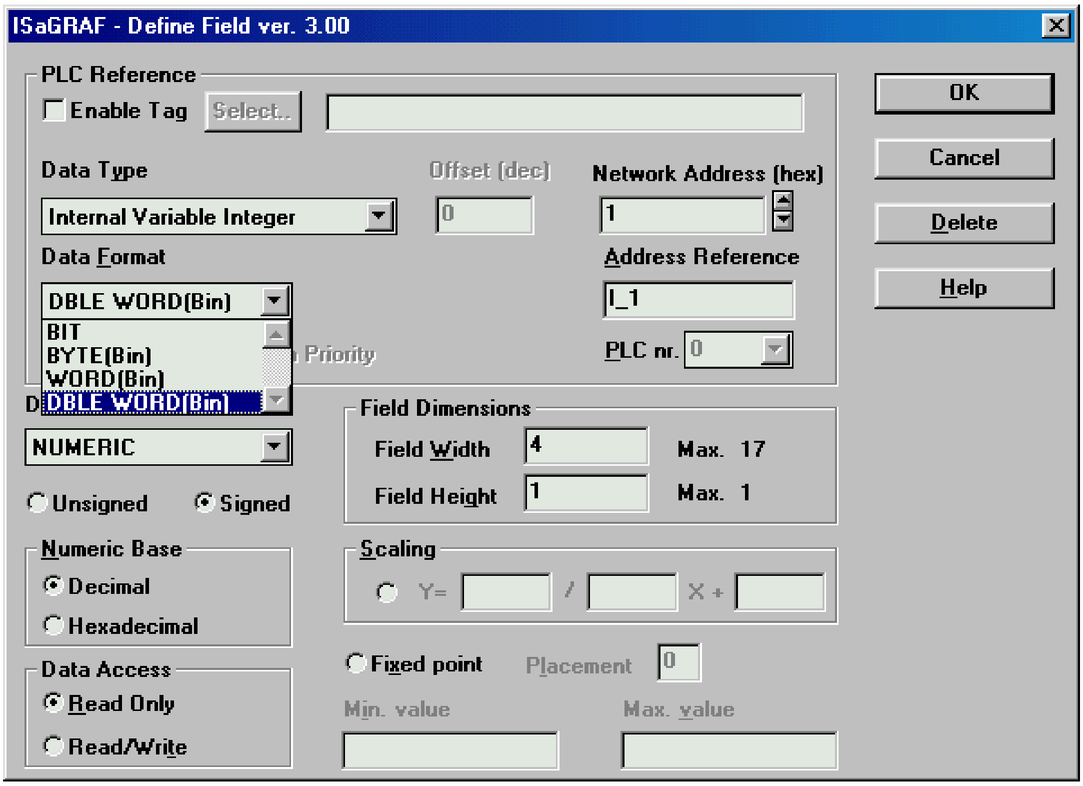
Name	Attrib.	Addr.	Comment
IUIM0	[input]	3000	
IUIM1	[input]	3001	
IUIM2	[input]	3002	
IUIM3	[input]	3003	
IUIM4	[input]	3004	
IUIM5	[input]	3005	
IUIM6	[input]	3006	
IUIM7	[input]	3007	

IUIM0
@3000 [input] (false,true)

Figure 28

3.3.8 Internal Variable Integer

Internal Variable Integer Data Type allows referencing ISaGRAF variables declared as Integer (see Figure 30) and with a certain Network Address.



ISaGRAF - Define Field ver. 3.00

PLC Reference
□ Enable Tag

Data Type: Internal Variable Integer
Offset (dec): 0
Network Address (hex): 1
Data Format: DBLE WORD(Bin)
Address Reference: I_1
PLC nr. 0
NUMERIC
○ Unsigned ● Signed

Field Dimensions
Field Width 4 Max. 17
Field Height 1 Max. 1

Numeric Base
● Decimal
○ Hexadecimal

Scaling
○ Y= / X +

Data Access
● Read Only
○ Read/Write

○ Fixed point Placement 0
Min. value Max. value

Figure 29

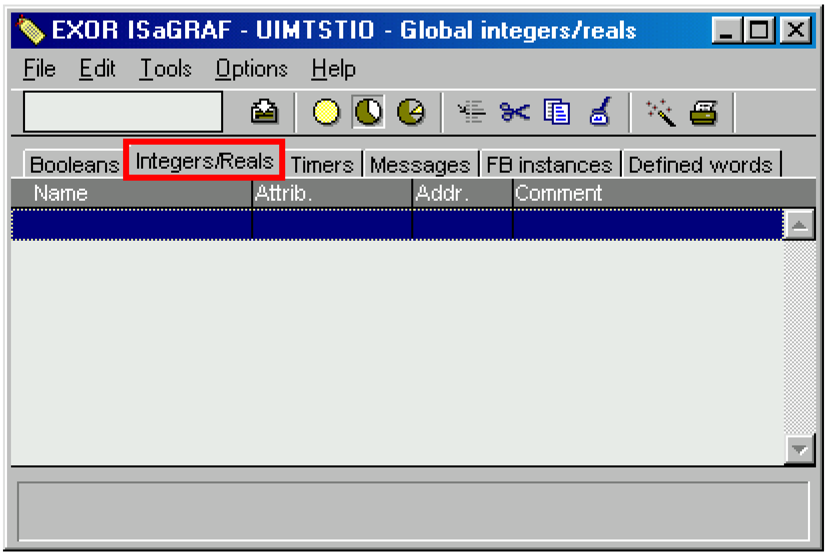
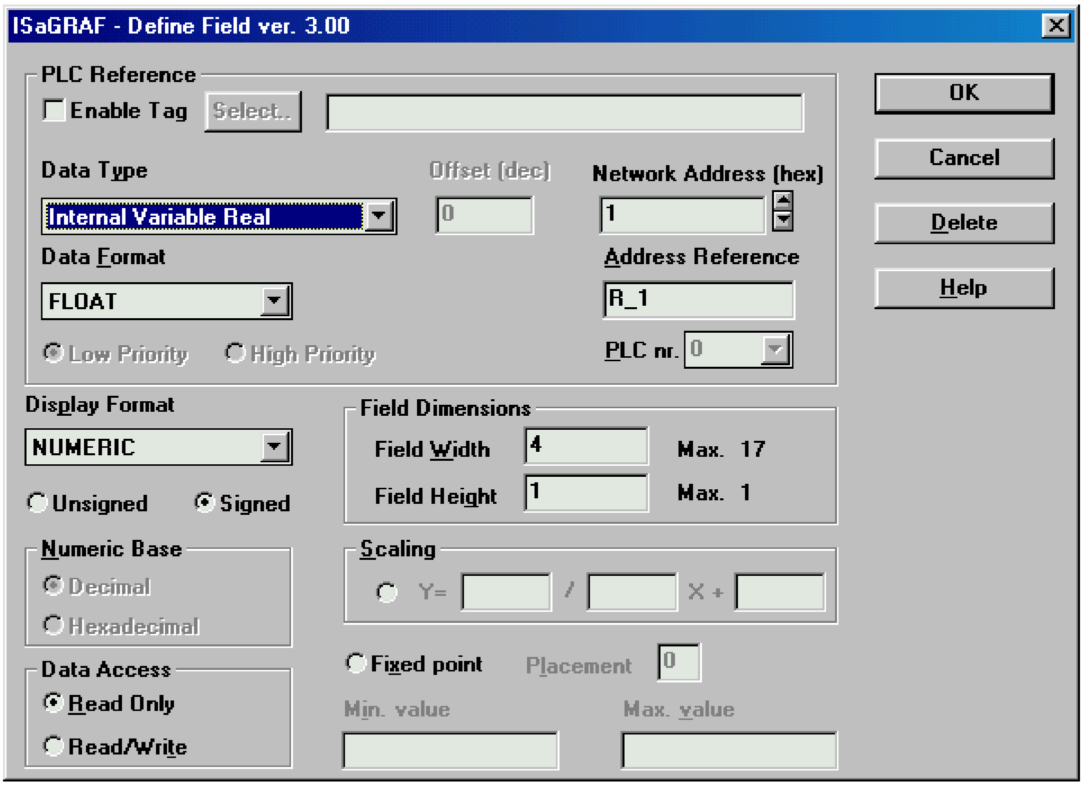


Figure 30

3.3.9 Internal Variable Real

Internal Variable Real Data Type allows referencing to ISaGRAF internal variable defined as Real. The floating-point format used by UniOP and ISaGRAF is the standard IEEE 754.


ISaGRAF - Define Field ver. 3.00

PLC Reference
□ Enable Tag

Data Type: Internal Variable Real
Offset (dec): 0
Network Address (hex): 1
Data Format: FLOAT
Address Reference: R_1
● Low Priority ○ High Priority
PLC nr.: 0

Display Format: NUMERIC
○ Unsigned ● Signed

Field Dimensions
Field Width: 4 Max. 17
Field Height: 1 Max. 1

Numeric Base
● Decimal
○ Hexadecimal

Scaling
○ Y= / X +

Data Access
● Read Only
○ Read/Write

○ Fixed point Placement 0
Min. value
Max. value

Figure 31

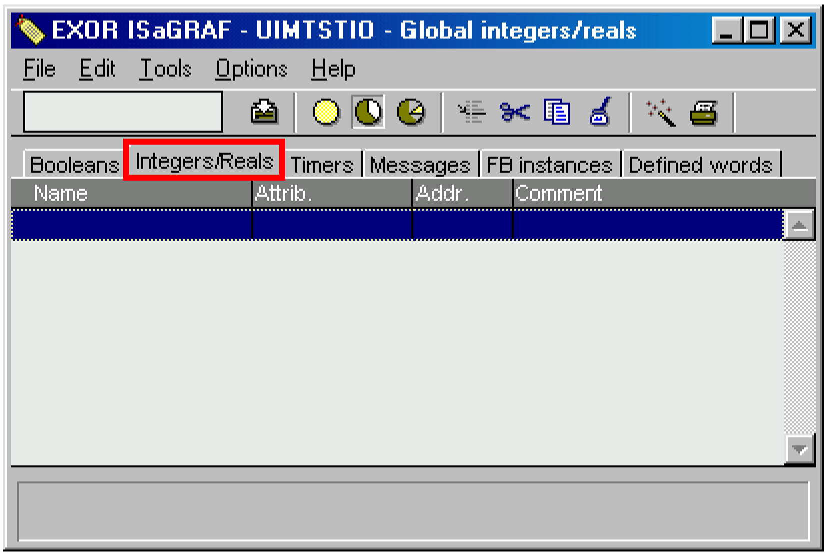
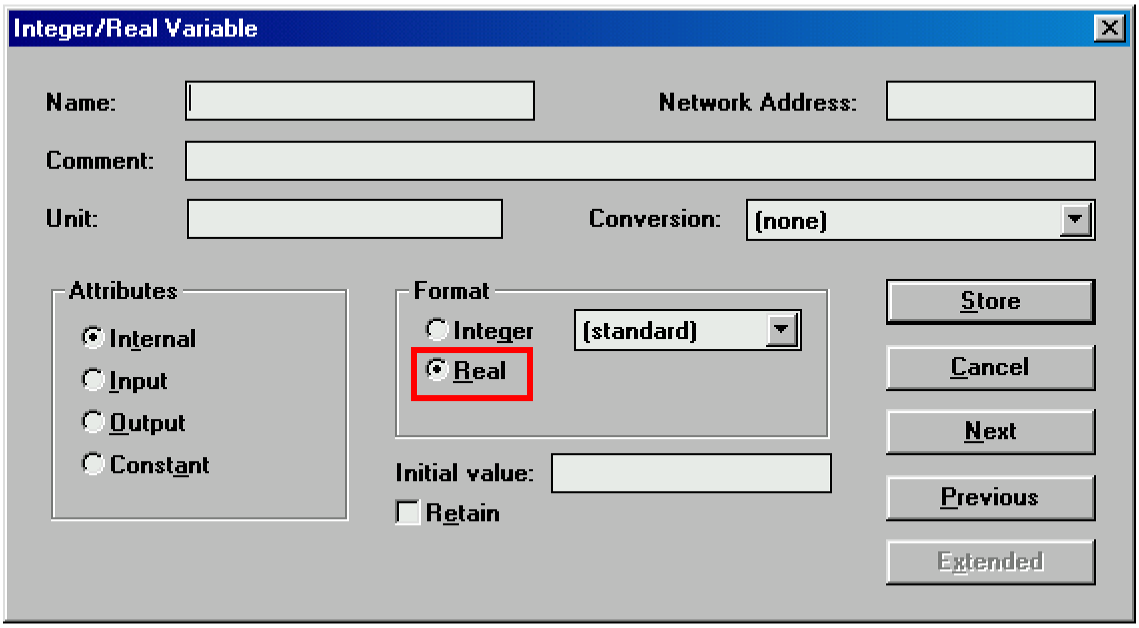


Figure 32



Attributes: ● Internal ○ Input ○ Output ○ Constant

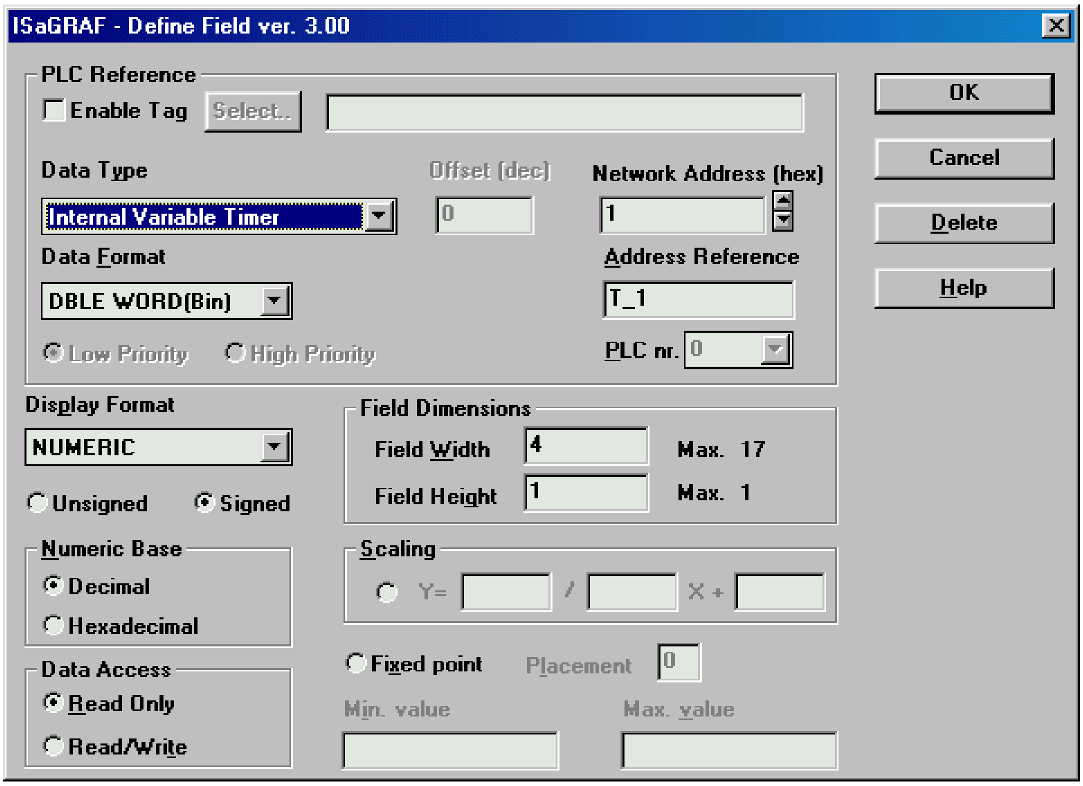
Format: ○ Integer (standard) ● Real

□ Retain

Figure 33

3.3.10 Internal Variable Timer

Internal Variable Timer allows referencing to ISaGRAF internal variables declared as Timers (see Figure 35) in the ISaGRAF dictionary.



ISaGRAF - Define Field ver. 3.00

PLC Reference: □ Enable Tag

Data Type: Internal Variable Timer

Offset (dec): 0

Network Address (hex): 1

Data Format: DBLE WORD(Bin)

Address Reference: T_1

● Low Priority ○ High Priority

PLC nr. 0

Display Format: NUMERIC

○ Unsigned ● Signed

Field Dimensions

Field Width 4 Max. 17

Field Height 1 Max. 1

Numeric Base: ● Decimal ○ Hexadecimal

Scaling: ○ Y= / X +

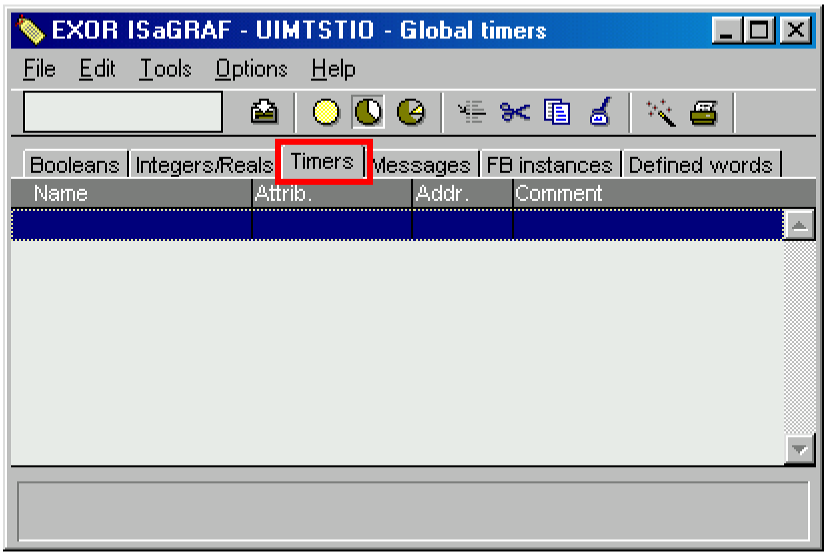
Data Access: ● Read Only ○ Read/Write

○ Fixed point Placement 0

Min. value

Max. value

Figure 34



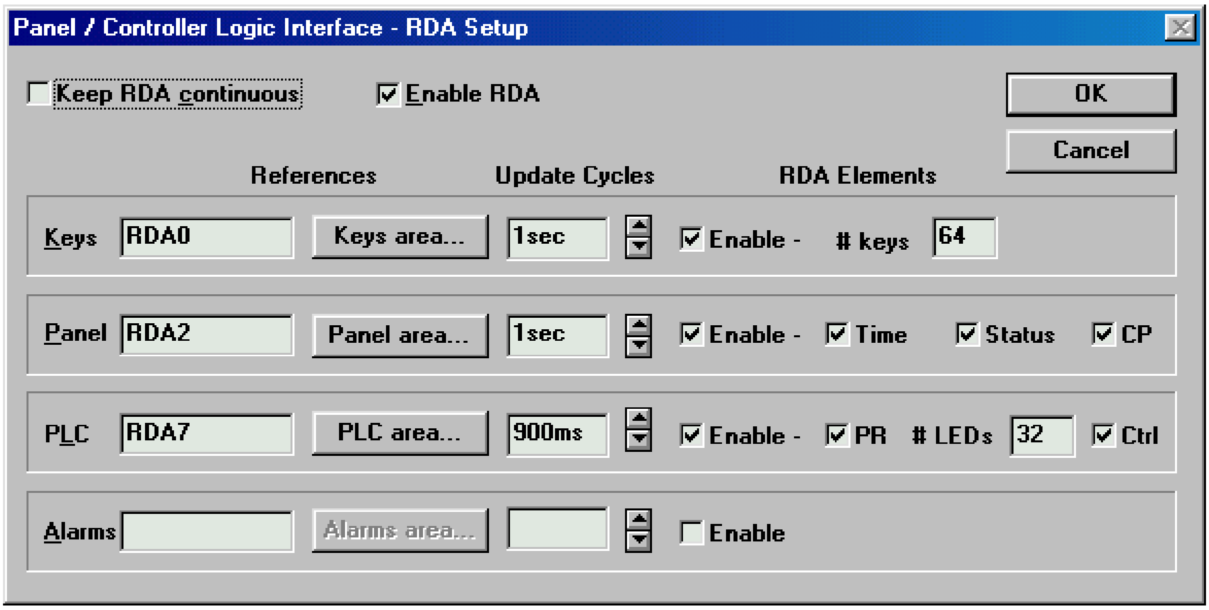
EXOR ISaGRAF - UIMTSTIO - Global timers

Figure 35

3.4 Using the RDA

The Reserved Data Area memory can be configured in the memory of the Internal Controller.
To use the RDA a certain number of Tag's with contiguous network address must be configured in the ISaGRAF project.

Each component of the RDA (Keys, Panel, PLC areas) must be contiguous. Different areas can be placed in different Internal Controller memory types; i.e. they can start from any network address, but from this point the necessary number of variables with contiguous network address have to be configured in the ISaGRAF project.



Panel / Controller Logic Interface - RDA Setup

□ Keep RDA continuous ☑ Enable RDA

	References		Update Cycles	RDA Elements
Keys	RDA0	Keys area...	1sec	☑ Enable - # keys 64
Panel	RDA2	Panel area...	1sec	☑ Enable - ☑ Time ☑ Status ☑ CP
PLC	RDA7	PLC area...	900ms	☑ Enable - ☑ PR # LEDs 32 ☑ Ctrl
Alarms		Alarms area...		□ Enable

Figure 36

Note *the 'Keep RDA continuous' option is not supported. The checkbox should be always left not checked.*

The ISaGRAF memory organisation is based on double words. UniOP memory is organised a in word-oriented mode. The RDA start address must point to a double word object; the UniOP words will be packed as described in the figures below.

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
	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
	K16	K15	K14	K13	K12	K11	K10	K9	K8	K7	K6	K5	+/-	.	9	8	7	6	5	4	3	2	1	0	K4	K3	K2	K1	Down	Up	Right	Left

Figure 37 – The Keyboard Area

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
	Day								Month								Reserved								Day of week							
	Minutes								Seconds								Year								Hour							
	CP (Current Page)																S15	S14	S13					S8	S7	S6	S5	S4	S3	S2	S1	S0

Figure 38 – The Panel Area

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
	L16	L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	PR (Page Request)															
	C15													C2	C1	C0	L32	L31	L30	L29	L28	L27	L26	L25	L24	L23	L22	L21	L20	L19	L18	L17

Figure 39 – The PLC Area

3.5 Programming the Mailbox

The Mailbox can be configured in the Internal Controller memory area. Each ISaGRAF network address corresponds, in the Internal Controller memory, to a double word. To the mailbox should be reserved a minimum number of 10 contiguous network addresses. Starting address of the mailbox should be set using Word Data format.
Words inside the mailbox areas can be referred with Tag using Word data format and contiguous network address. The following table shows how the Mailbox words are mapped to the ISaGRAF double words.

<table>
<tr><th>Bit</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td></td><td colspan="16">Command/Response Word</td><td colspan="16">Status Word</td></tr>
<tr><td></td><td colspan="16">Parameter 1</td><td colspan="16">Parameter 0</td></tr>
<tr><td></td><td colspan="16">Parameter 3</td><td colspan="16">Parameter 2</td></tr>
<tr><td></td><td colspan="16">….</td><td colspan="16">…..</td></tr>
<tr><td></td><td colspan="16">Parameter 17</td><td colspan="16">Parameter 16</td></tr>
</table>

Figure 40 – The Mailbox

3.6 Alarms

The Alarm area in the Internal Controller memory is organised as double words. A certain number of Tags should be configured and associated to contiguous network address.
The ISaGRAF memory is organised in double words; UniOP Alarm memory is organised in words; two UniOP words are merged in one ISaGRAF double word.
The least significant bit of the ISaGRAF double word corresponds to Alarm 1; the most significant bit of the ISaGRAF double word corresponds to Alarm 32.

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
									...	...	...	...	A52	A51	A50	A49	A48	A47	A46	A45	A44	A43	A42	A41	A40	A39	A38	A37	A36	A35	A34	A33

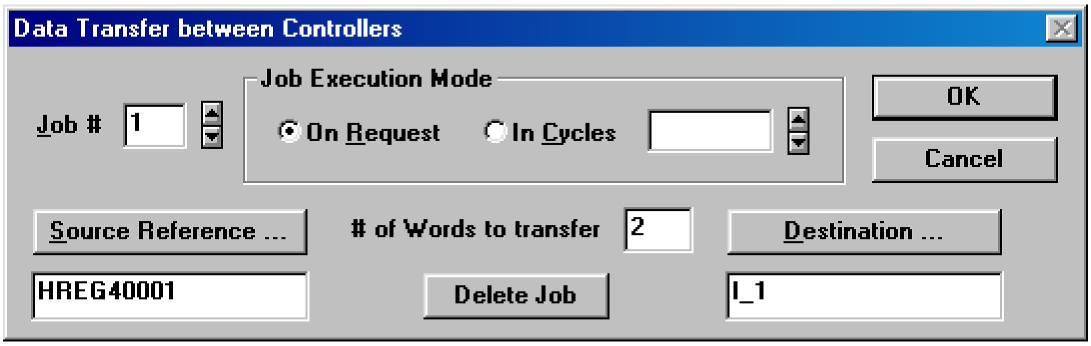
Figure 41 – The Alarm Area

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 Internal ISaGRAF memory. Data can also be copied from the internal ISaGRAF memory to the external controller memory.
To transfer data between controllers the ISaGRAF variables involved in the copy process must have a defined, unique network address.

The Data Transfer process has different options based on the different Data Format of the variable memories involved in the copy process.
The internal ISaGRAF memory organization is based on double word location memory and it should be taken in consideration when the Source or Target address is defined.

When a block of ISaGRAF variables is going to be transferred, the network address for ALL the variables involved in the copy process, as target or source, must be defined.
The number of configured words to transfer should match the number of ISaGRAF variables define with contiguous address of the same Data Format.
If this condition is not respected the Data Transfer will not be executed.
The ISaGRAF variables are copied one next to the other depending on the Network Address order.



Job # 1

Job Execution Mode: ● On Request ○ In Cycles

# of Words to transfer 2

HREG40001

I_1

Figure 42

Before starting a copy operation the Data Transfer tool checks the byte order convention used by the Source and the Target addresses. In UniOP an 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 tool does not apply any transformation.
If Source and Target have different byte ordering, the Data Transfer tool applies a byte swap according to the rules explained in Figure 43.

Word Byte Swapping

Source	Byte1	Byte0
Target	Byte0	Byte1

Double Word Byte Swapping

Source	Byte3	Byte2	Byte1	Byte0
Target	Byte0	Byte1	Byte2	Byte3

Figure 43

3.8 UniOP Communication Diagnostic

UniOP provides some useful communication diagnostic information for the Internal Controller in the System Menu. 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 only the Internal Controller. In this case the COMM LED blinks when a communication error with the SCM03 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 except when the ISaGRAF Workbench stops the Internal Controller.

Note *UniOP will not communicate with the ISaGRAF Internal Controller if the PLC application is not running.*

If a UniOP variable is linked to a non-existing network address, no communication error will be displayed.

2) The Designer project is configured to use an External Controller at the same time as the SCM03 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 SCM03.
In case of a communication error with the External Controller, the System Menu provides the communication error code as usual.

4. Using the Internal Controller

The UniOP Internal Controller is an ISaGRAF based Softlogic system fully compatible with the ISaGRAF standard.
The complete description of the ISaGRAF programming tool is contained in the on-line manual distributed together with the ISaGRAF package; please refer to this documentation for a complete description.

4.1 The ISaGRAF Softlogic System

An ISaGRAF project is divided into several programming units called programs. The programs of the project are linked together in a tree-like architecture. Programs can be described using any of SFC, FC (Flow Chart), FBD, LD, ST or IL graphic or literal languages.
The ISaGRAF Language reference contains all the User's information about project architecture and program/sub-programs execution.

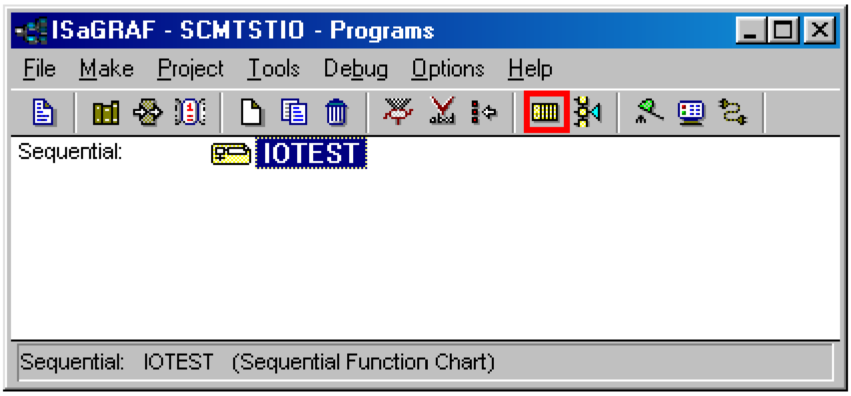
ISaGRAF Programs can be written both with reference to only internal PLC memory and with reference to an external I/O configuration.
Current implementation of HMIcontrol allows the use of integrated digital or analog I/O and remote CANopen I/O modules.

4.2 Setting-up the I/O Configuration

The I/O configuration of the Internal Controller must be defined before compiling an application, which refers to any I/O variables.

The "Edit" menu contains basic commands to define the selected board (set-up its parameters), and to connect I/O variables to its channels.

The ISaGRAF variables defined in the dictionary can be directly connected to the I/O points of the boards configured in the "I/O Connection". The I/O Connection toll is accessible from the ISaGRAF Program window by clicking on the icon "I/O Connection". See Figure 44.


ISaGRAF - SCMTSTIO - Programs
File Make Project Tools Debug Options Help

Sequential: IOTEST

Sequential: IOTEST (Sequential Function Chart)

Figure 44

Before connecting I/O variables to a board, the board identification must be entered. A library of pre-defined boards is available on the ISaGRAF workbench. EXOR delivers the ISaGRAF package with a complete set of pre-compiled and tested I/O library elements for the local and remote I/O modules. Library elements for new I/O devices can also be compiled when needed.

The "Edit / Set Board/Equipment" command is used to set-up board identification. This command can be used to select either a single board, or complex I/O equipment from the ISaGRAF library. It is also possible to double click on a slot to set the corresponding board or equipment.

The list on the left shows the rack of the target machine, with board slots. A slot can be free or used by I/O boards or complex equipment. An order number identifies each slot. The rack may contain up to 255 boards. The list on the right shows the board's parameters and the variables connected on the selected board. A board may have up to 128 I/O channels. The total number of single I/O boards (including single equipment and boards of complex equipment) cannot exceed 255.

All the channels of a single board have the same type (Boolean, integer/real or message) and direction (input or output). Real and integer variables are not distinguished during I/O connection. A complex I/O equipment represents an I/O device with channels of different types or directions. A complex I/O equipment is represented as a list of single I/O boards. It uses only one slot in the rack list.

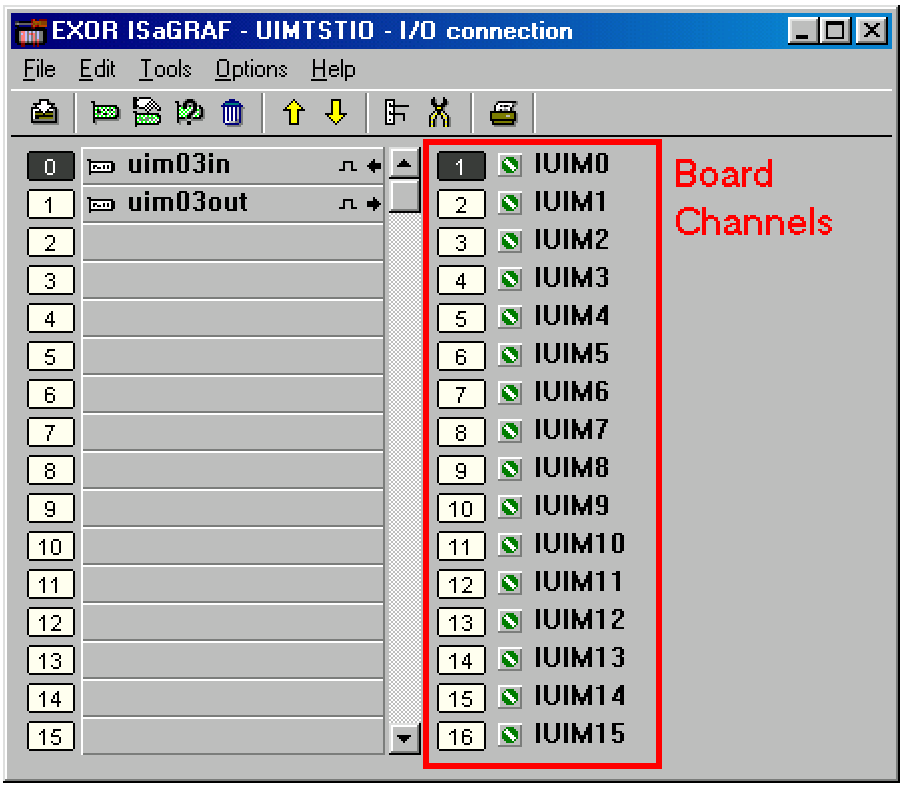
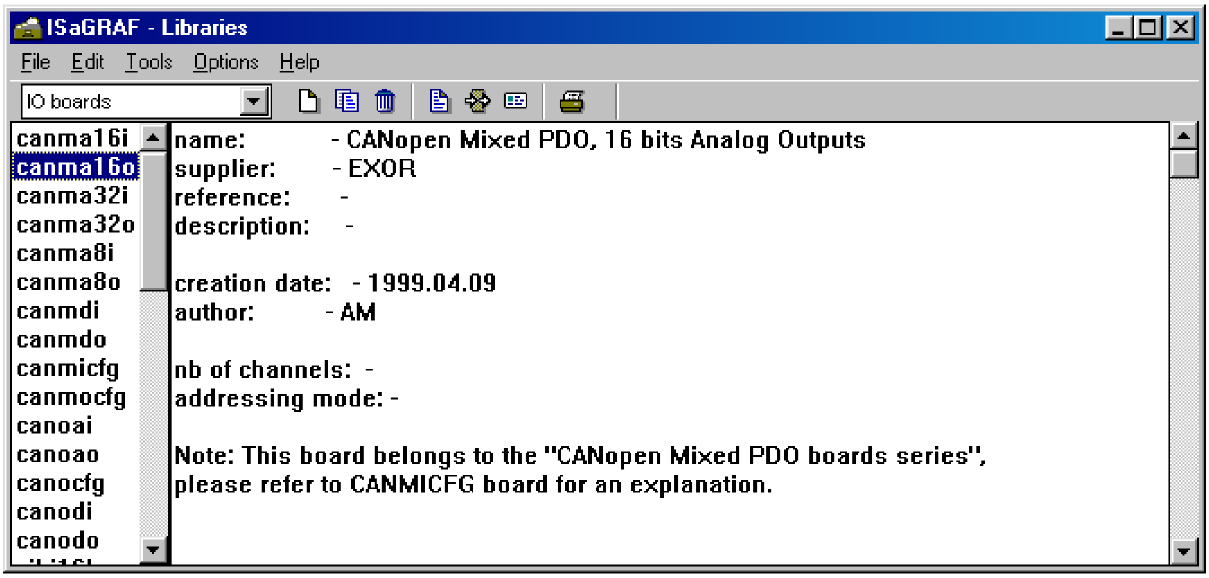


Figure 45

4.2.1 Installing and browsing the Library

The additional library entries for the I/O modules used by the internal PLC are automatically installed by the Designer-ISaGRAF set-up procedure.
The “Library” tool is accessible from the “Project Management” window under the “Tools” menu.

The ISaGRAF Library tool allows browsing the installed set of boards, equipment and function blocks; additionally the Backup/Restore functions are available under then "Tools" menu. Libraries tool window is shown in Figure 46.



name: - CANopen Mixed PDO, 16 bits Analog Outputs
supplier: - EXOR
reference: -
description: -

creation date: - 1999.04.09
author: - AM

nb of channels: -
addressing mode: -

Note: This board belongs to the "CANopen Mixed PDO boards series", please refer to CANMICFG board for an explanation.

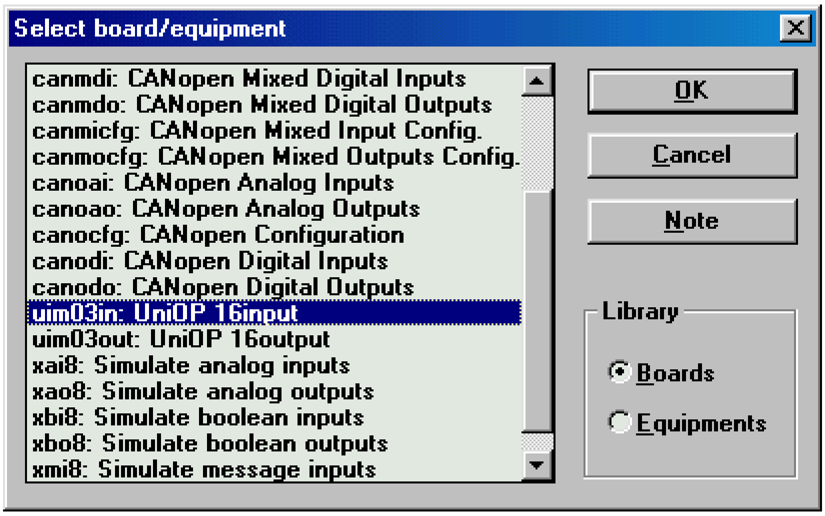
Figure 46

Note *When a new library, board or function is restored, the "Verify (compile)" function available under the "File" menu must be used to compile the newly added element in the current environment.*

For each board, function or equipment available in the EXOR ISaGRAF system, an on-line technical note is available. It includes a brief description of the board and the detailed explanation of all the channels.

4.2.2 Configuring Local I/O Modules

The EXOR ISaGRAF programming package includes in the library the complete set of available UniOP I/O boards.
The "Select board/equipment" dialog box is shown in Figure 47. Click on the "Note" button to recall the on-line description of the selected board.
UniOP local I/O modules are listed with the prefix name "**uim**".



canmdi: CANopen Mixed Digital Inputs
canmdo: CANopen Mixed Digital Outputs
canmicfg: CANopen Mixed Input Config.
canmocfg: CANopen Mixed Outputs Config.
canoai: CANopen Analog Inputs
canoao: CANopen Analog Outputs
canocfg: CANopen Configuration
canodi: CANopen Digital Inputs
canodo: CANopen Digital Outputs
uim03in: UniOP 16input
uim03out: UniOP 16output
xai8: Simulate analog inputs
xao8: Simulate analog outputs
xbi8: Simulate boolean inputs
xbo8: Simulate boolean outputs
xmi8: Simulate message inputs

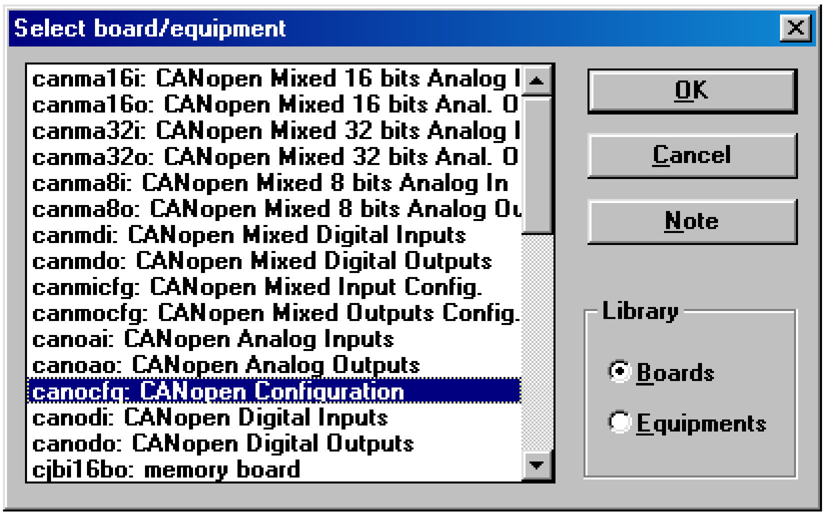
● Boards
○ Equipments

Figure 47

4.2.3 Configuring CANopen Distributed I/O

Distributed I/O systems based on the CANopen fieldbus interface of the SCM03 can be easily configured using the I/O Connection tool.
Several configuration options are available as described in this chapter.

The Internal Controller CANopen master interface must be configured using the **canocfg** board.
Select the **canocfg** from the list in the Select board/equipment dialog as shown in the figure below.



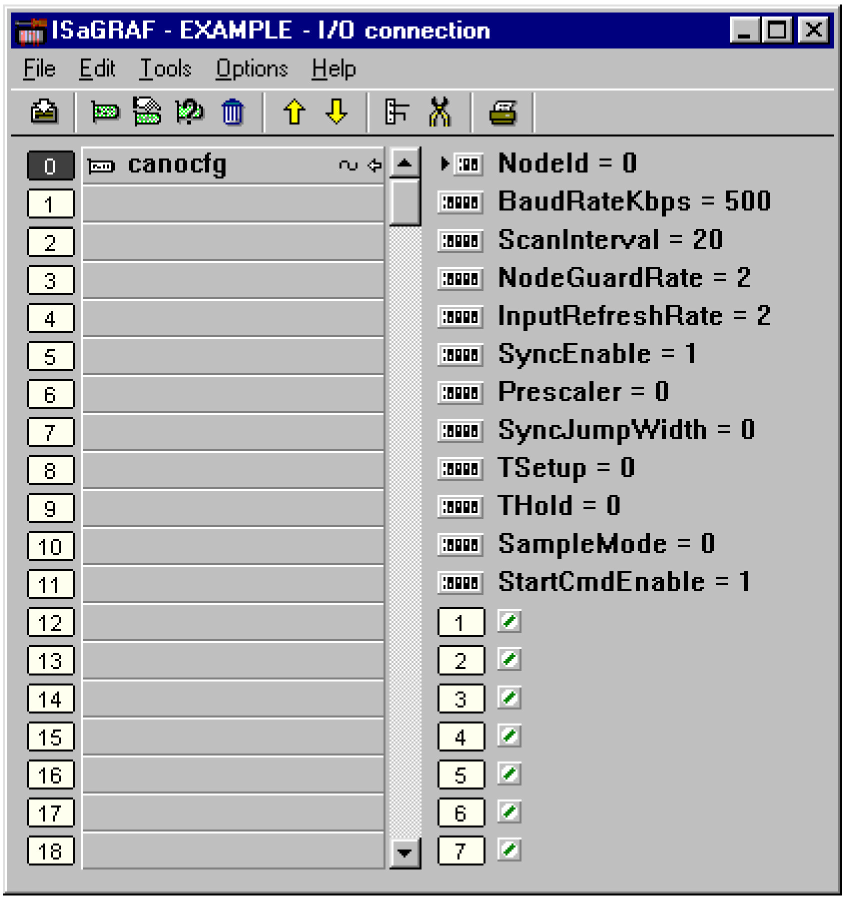
canma16i: CANopen Mixed 16 bits Analog I
canma16o: CANopen Mixed 16 bits Anal. O
canma32i: CANopen Mixed 32 bits Analog I
canma32o: CANopen Mixed 32 bits Anal. O
canma8i: CANopen Mixed 8 bits Analog In
canma8o: CANopen Mixed 8 bits Analog Ou
canmdi: CANopen Mixed Digital Inputs
canmdo: CANopen Mixed Digital Outputs
canmicfg: CANopen Mixed Input Config.
canmocfg: CANopen Mixed Outputs Config.
canoai: CANopen Analog Inputs
canoao: CANopen Analog Outputs
canocfg: CANopen Configuration
canodi: CANopen Digital Inputs
canodo: CANopen Digital Outputs
cjbi16bo: memory board

● Boards
○ Equipments

Figure 48

Note *The **canocfg** board must always be the first CAN board in the I/O rack.*

The **canocfg** board has several parameters that must be configured for proper operation.


ISaGRAF - EXAMPLE - I/O connection
File Edit Tools Options Help
0 canocfg
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
1
2
3
4
5
6
7

NodeId = 0
BaudRateKbps = 500
ScanInterval = 20
NodeGuardRate = 2
InputRefreshRate = 2
SyncEnable = 1
Prescaler = 0
SyncJumpWidth = 0
TSetup = 0
THold = 0
SampleMode = 0
StartCmdEnable = 1

Figure 49

NodeID is the node number of the master in the CANopen network. It is currently not used, but it is provided for future master to master communications

BaudRateKbps is the speed of CAN network. According to the CAN bus specifications the possible values are 10K, 20K, 50K, 100K, 125K, 250K, 500K, 800K, 1000K. If the BaudRateKbps parameter is defined, an internal table defines the network timing parameters. If custom settings are needed for the timing, the BaudRateKbps parameter must be set to 0, and appropriate values must be entered for Prescaler, SyncJumpWidth, Tsetup, Thold and SampleMode. You will find more details about CAN timing another section of this manual.

ScanInterval defines how frequently the CAN master will perform update operations on the bus. It is defined in milliseconds. Messages sent by the master will be based on the ticks of this counter, while asynchronous incoming messages may be received at any moment.

NodeGuardRate defines how frequently the master will manage the Node Guarding. It is a safety feature used to check if remote nodes are alive and operational; remote nodes may use the NodeGuarding message to manage the Life Guarding option, i.e. to know if the master is alive. NodeGuardRate is defined in ScanInterval units. The value 0 will disable Node Guarding.

InputRefreshRate defines how frequently the master will update input objects. Normally remote nodes are programmed to send Input objects every time input values are changing. In some cases it is not possible or, for safer operation, a forced Refresh is preferred. InputRefreshRate is defined in ScanInterval units. The value 0 means to disable Input Refresh.

SyncEnable defines whether the master will send the SYNC message at each scan cycle. The SYNC message is used in the CANopen network to synchronize inputs and outputs located in different remote nodes. If SYNC is enabled all inputs received will be sampled at the same time and all the outputs sent will be activated at the same time. The choice of using SYNC must be done in the early design phase and all the nodes must be set-up properly.

StartCmdEnable defines whether the master will send the NMT Start Command message at start-up and in case of guarding errors.

Prescaler
SyncJumpWidth
Tsetup
Thold
SampleMode **t**his set of parameters is used to set the timing of CAN bus in a custom way. Normally the default parameters programmed in the SCM03 module are correct, but in some extreme conditions it might be necessary to set a custom timing.

The **canocfg** board includes 7 virtual input points, useful for diagnostic purposes:

QuickStatus contains the global status of the CANopen controller. The error codes are shown in the table below.

Code	Severity	Description
0		No errors
1	fatal	Invalid baudrate
2	fatal	Invalid board (perhaps an old board version ?)
3	fatal	Too many boards defined
4	fatal	Missing Configuration board (it must be defined before any CANopen board)
5	fatal	Invalid Node ID (1..127)
6	fatal	Invalid PDO Length (0..8)
7		Node has Guarding Error (toggling bit or status incorrect)
8		Node is Dead (Node does not reply)
9		Node restarted (Node is alive again)
10	fatal	Missing preceding CANMICFG or CANMOCFG board
11	fatal	Invalid Offset
12		Function Block CANSDORD/WR: too many SDO/PDO's
13		Function Block CANSDORD/WR: invalid parameter/s
14		Function Block CANSDORD/WR: invalid reply from remote SDO server
15		Function Block CANSDORD: returned size differs from requested size
16		Function Block CANSDORD/WR: no reply timeout
17		CAN Error Bus Offstate
18		CAN Error STAT_ERR_PASSIV
19		CAN Error STAT_WARN
20		CAN Error STAT_STUFF_ERR
21		CAN Error STAT_FORM_ERR
22		CAN Error STAT_ACK_ERR
23		CAN Error STAT_BIT_ERR
24		CAN Error STAT_CRC_ERR

Table 5

Baudrate[Kbps] contains the actual baud rate (it can be useful when you specify 0 in the BaudRate parameter, i.e. a custom setting of baudrate is used)

Prescaler contains the currently used value of Prescaler

SyncJumpWidth contains the currently used value of SyncJumpWidth

TSetup contains the currently used value of Tsetup

THold contains the currently used value of Thold

SampleMode contains the currently used value of SampleMode

4.2.3.1 Standard CANopen I/O Boards

Standard CANopen I/O may be configured using four different boards:

- **canodi** digital inputs
- **canodo** digital output
- **canoai** analog input
- **canoao** analog output

For each board configured the master defines a PDO message. Digital I/O's are assigned to 1st PDO and Analog I/O's to 2nd PDO. PDO addressing is predefined by the CANopen standard:

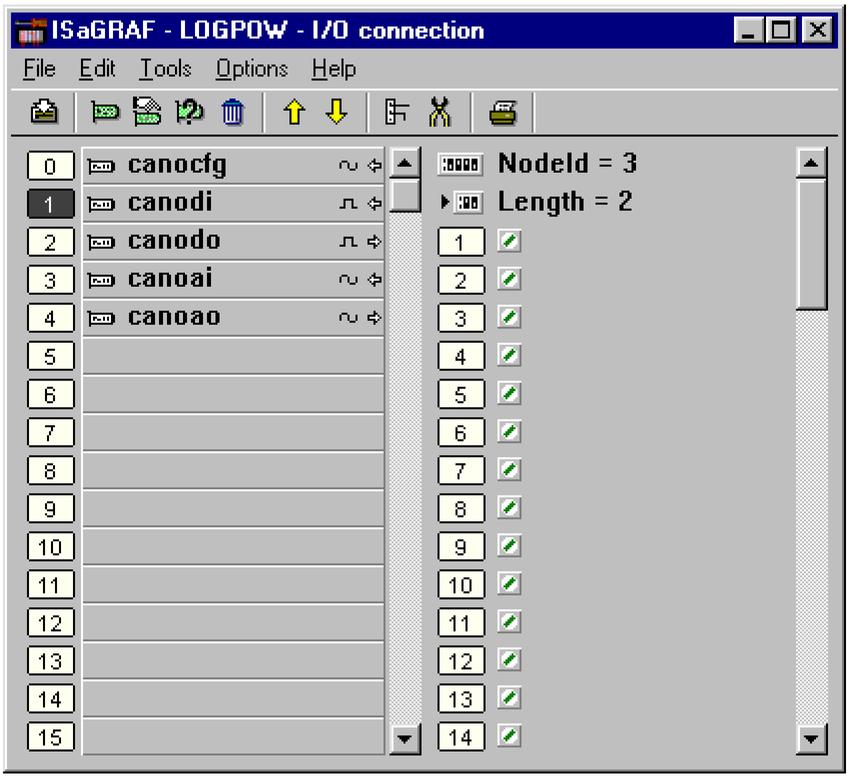
Digital input	180h + NodeID
Digital output	200h + NodeID
Analog input	280h + NodeID
Analog Output	300h + NodeID

Table 6

Two parameters must be defined for each board:

NodeID is the node number (in the range 1 to 127) assigned to the remote node where the I/O is located

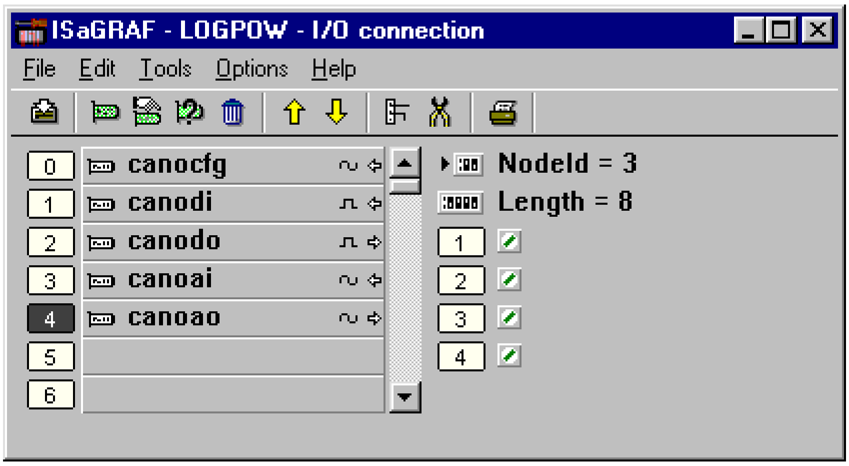
Length is the number of data bytes (from 1 to 8). For Digital I/O each I/O point is transferred in one bit; a PDO can contain up to 64 I/O values. Analog I/O points use two bytes for each channel; a PDO can contain up to 4 values



NodeId = 3
Length = 2

Figure 50

In this example the **canodi** board defines 2*8 = 16 inputs located in the remote node #3.


ISaGRAF - LOGPOW - I/O connection
File Edit Tools Options Help
0 canocfg
1 canodi
2 canodo
3 canoai
4 canoao
5
6
1
2
3
4

NodeId = 3
Length = 8

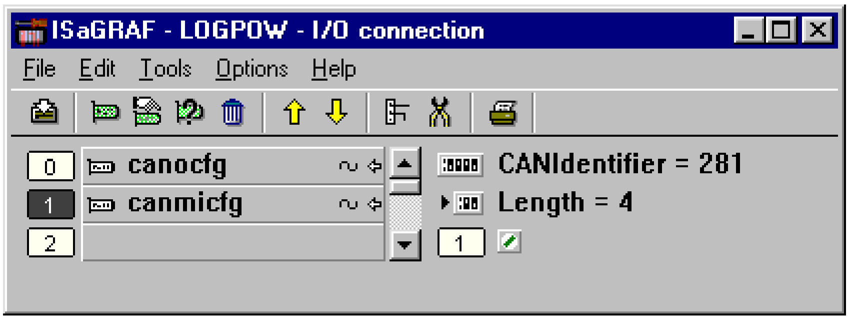
Figure 51

In this example the **canoao** board defines 4 analog outputs located in the remote node #3.

4.2.3.2 CANopen Mixed Boards

The CANopen Mixed PDO boards allow to define PDO objects containing a mix of different data types; such objects cannot be defined using standard I/O boards. For example, you could have a special sensor that accommodates in the same PDO Boolean and integer values. Examples of such type of PDO are typically encoders, motor drives, etc.
To define this type of PDO you have first to include a CANopen mixed configuration board: two boards are available:

- **canmicfg** input PDO
- **canmocfg** output PDO


ISaGRAF - LOGPOW - I/O connection
File Edit Tools Options Help
0 canocfg
1 canmicfg
2
1

CANIdentifier = 281
Length = 4

Figure 52

Two parameters must be defined:

CANidentifier is a valid CAN identifier as specified by the manufacturer of the remote device. Note that the value must be entered in hexadecimal.

Length is the total number of data bytes contained in the message.

The board will return in the first I/O point the status of the object (the same coding introduced for the **canocfg** board has been used).

The definition of the configuration board itself is sufficient to enable the data exchange between the controller and the remote device. Data is stored in the controller in internal buffers of appropriate length.
To access the data in the buffers with different data types special I/O boards are available. They must be defined just after the configuration board for the mixed-mode device. To enhance readability of the I/O configuration, you should leave blank slots before and after the group of "CANopen Mixed PDO boards" that make up your special mixed PDO.

The access boards for input PDOs are:

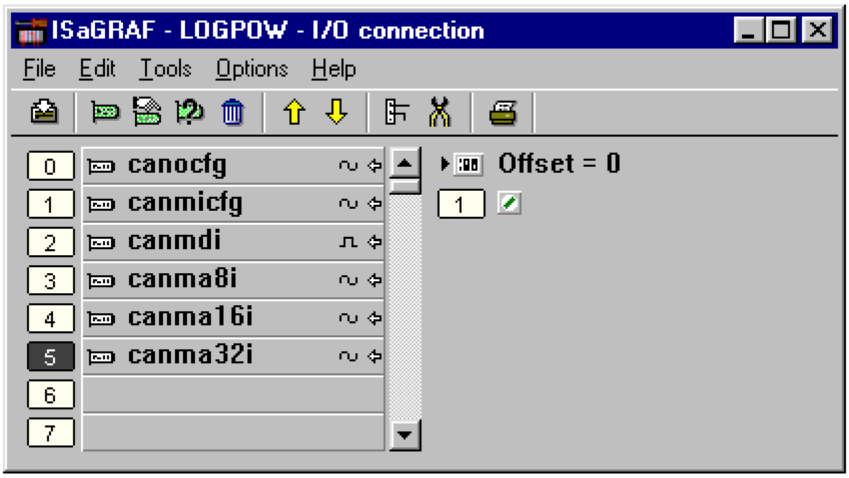
- **canmdi** mixed PDO digital inputs
- **canma8i** mixed PDO 8 bit analog input
- **canma16i** mixed PDO 16 bit analog input
- **canma32i** mixed PDO 32 bit analog input

The access boards for output PDOs are:

- **canmdo** mixed PDO digital outputs
- **canma8o** mixed PDO 8 bit analog output
- **canma16o** mixed PDO 16 bit analog output
- **canma32o** mixed PDO 32 bit analog output

For each board one parameter must be defined:

Offset position of data in the buffer


ISaGRAF - LOGPOW - I/O connection
File Edit Tools Options Help

0 canocfg
1 canmicfg
2 canmdi
3 canma8i
4 canma16i
5 canma32i
6
7

Offset = 0
1

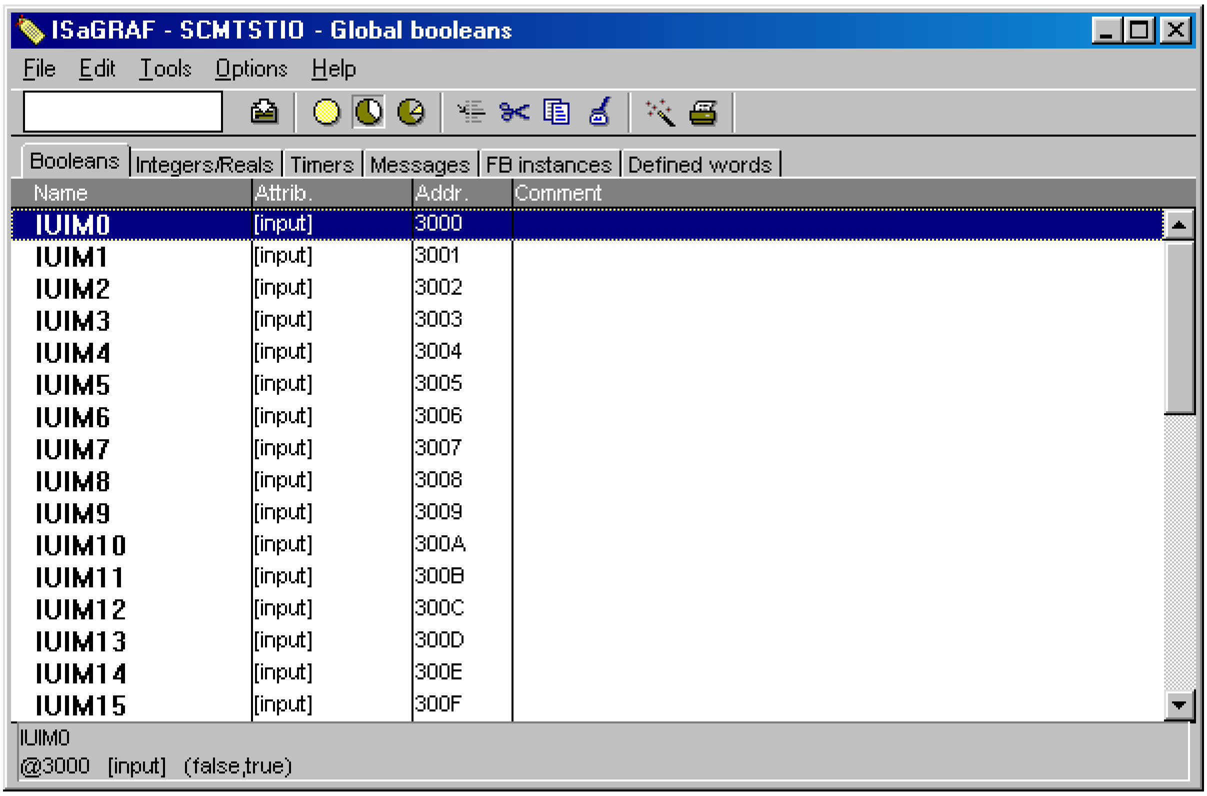
Figure 53

Note that the same data can be seen in different ways at the same time.

4.3 Assigning Network Addresses to Tags

The ISaGRAF Target provides a simple way to refer to its internal variables via a Modbus message structure.

The internal memory organisation is completely under the control of the ISaGRAF compiler and the user variables defined in the dictionary do not have a fixed address in memory. To refer to internal variables with an external device the Network address option has been introduced.
A Network Address is an ‘handle’ that can be used by an external device to request to the ISaGRAF kernel the value of a data item.
Network addresses are optional. Only variables with a non-zero network address can be spied by an external system (the UniOP HMI in this case) at run time. More generally, the network address provides an identifying mechanism for each run time communication system that cannot handle symbolic names. A network address may be entered for each variable when the variable is created or modified.



Name	Attrib.	Addr.	Comment
IUIM0	[input]	3000	
IUIM1	[input]	3001	
IUIM2	[input]	3002	
IUIM3	[input]	3003	
IUIM4	[input]	3004	
IUIM5	[input]	3005	
IUIM6	[input]	3006	
IUIM7	[input]	3007	
IUIM8	[input]	3008	
IUIM9	[input]	3009	
IUIM10	[input]	300A	
IUIM11	[input]	300B	
IUIM12	[input]	300C	
IUIM13	[input]	300D	
IUIM14	[input]	300E	
IUIM15	[input]	300F	

IUIM0
@3000 [input] (false,true)

Figure 54

The "Tools / Renumber addresses" command allows the user to set-up network addresses of an entire group of variables. When this command is used, it operates on the group of variables currently selected on the list. Entering a hexadecimal base address (address for the first variable of the group) results in network addresses of all the variables in the group being set with consecutive addresses. Entering a null base address resets to zero the network address of all the selected variables.

Note *The current version of EXOR ISaGRAF supports only the hexadecimal notation for the value of the Network Address.*

Network addresses for ISaGRAF variables can also be assigned with the "Tools / Modbus SCADA addressing map" tool, which allows to quickly create a Modbus virtual map with all the variables of the application.
You may refer to the ISaGRAF documentation for further details.

4.4 PLC Programming

The ISaGRAF package is based on the IEC 1131-3 model for Programming Languages for PLCs.
It integrates the 5 standard programming languages defined in the IEC 1131-3 model:

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

In addition the **FC** Flow Chart language is included for building decision diagrams.

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

4.5 PLC Project Upload

Upload functionality is a well-documented feature of the ISaGRAF Soft PLC available starting from Target version V3.22.
The SCM03 production starts by using ISaGRAF target version V3.30; this means that upload can be used according to the limitation described in the ISaGRAF documentation.

4.6 User Function Blocks

There are several useful function blocks included in the ISaGRAF kernel of the SCM03 controller module. List and description of these function blocks is available in a separate document.

4.7 Programming the CANopen Interface

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

4.7.1 Sending NMT Commands

CANopen NMT Commands can be sent through the CANONMT Function.
Using NMT services you can initialize, start, monitor, reset or stop nodes. All nodes are regarded as NMT slaves. Its Node ID uniquely identifies an NMT Slave in the network; the Node ID is a number in the range of 1 to127.
The CANONMT function block has two parameters:

Command Code the following values are available:

Command Code	
1	Start
2	Stop
128	Enter PRE-OPERATIONAL
129	Reset_Node
130	Reset_Communication

Table 7

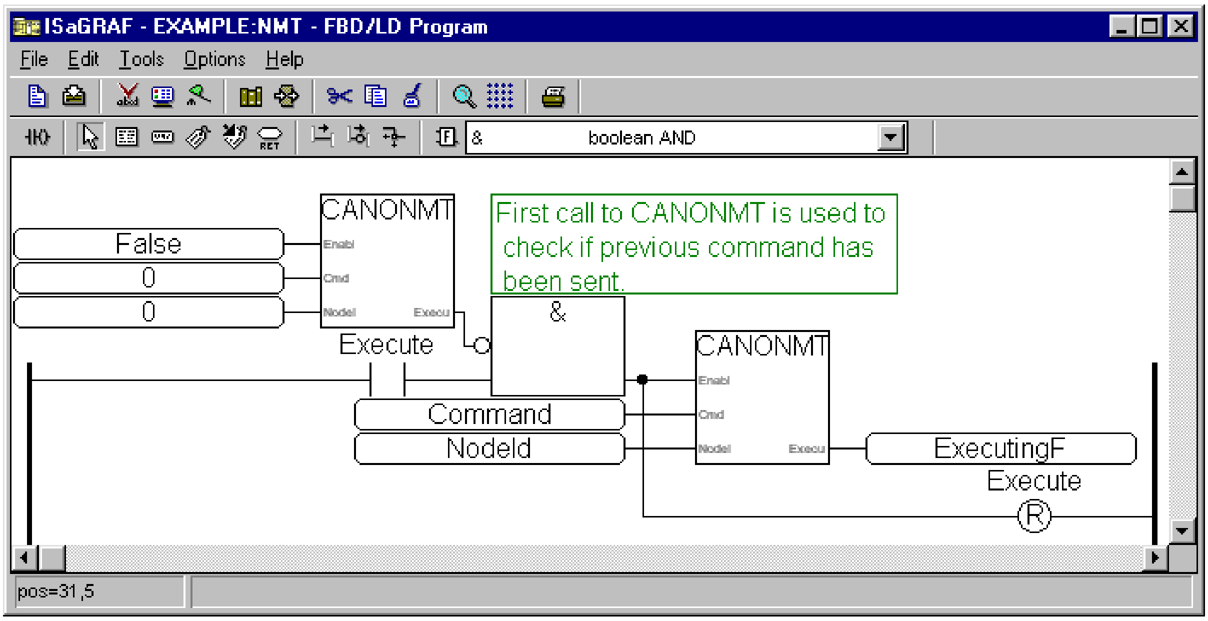
NodeId is the number of the destination Node or 0 if the command is directed to all nodes (broadcast).

Returned values are:

ExecutingF is a flag indicating the operation is still being executed. The value 1 means that the command has not been sent yet. The value 0 means that transmission has been completed and a new command can be sent.

Please refer to the CIA Draft specification DS301 for more information.

The figure below shows an example of using the CANONMT command.



First call to CANONMT is used to check if previous command has been sent.

Figure 55

4.7.2 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 CANopen master works as SDO client and remote nodes are servers.
While in PDO mode the transmission is normally cyclical and automatic, in SDO the data exchange is normally single-shot. Each session is normally used to 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.

- CANSDORD read remote variables
- CANSDOWR write remote variables

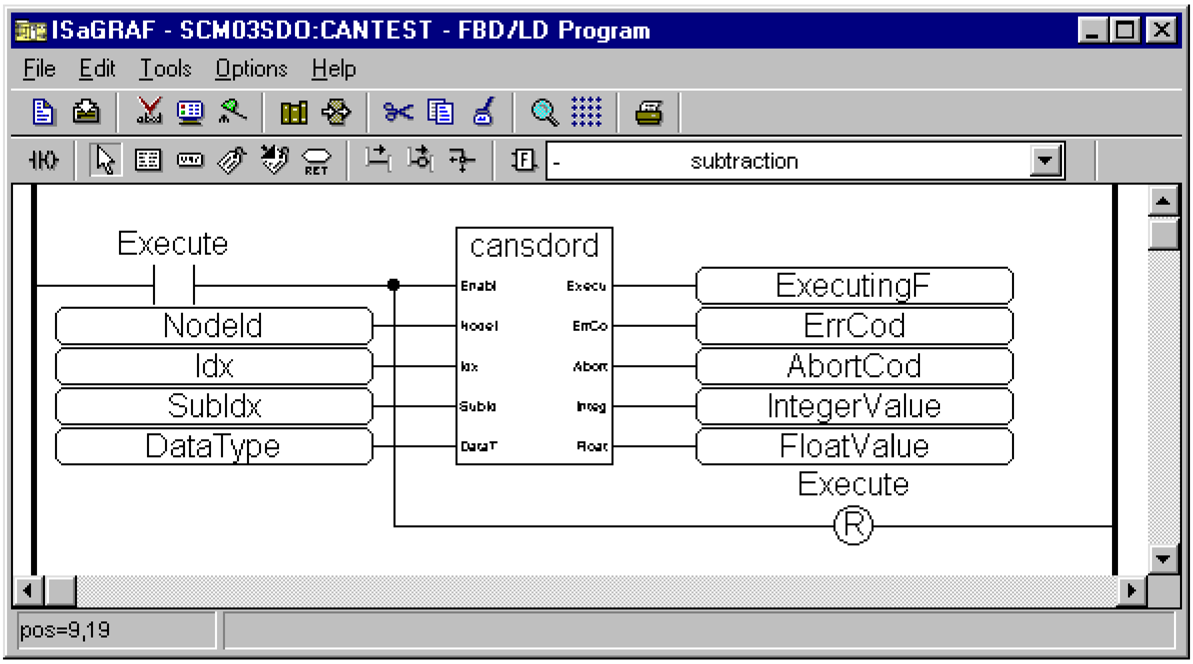


Figure 56

The parameters for the CANSDORD function block are:

NodeID remote node number

Index address of CAN object inside the remote node, as defined by the manufacturer

Subindex address of single variable inside the object

DataType one of the data types supported by CAN.

Returned values are:

ExecutingF is a flag indicating the operation is still being executed.

ErrCod 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.

AbortCode is the 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).

IntegerValue is the value of the read variable in integer format

FloatValue is the value of the read variable in float format

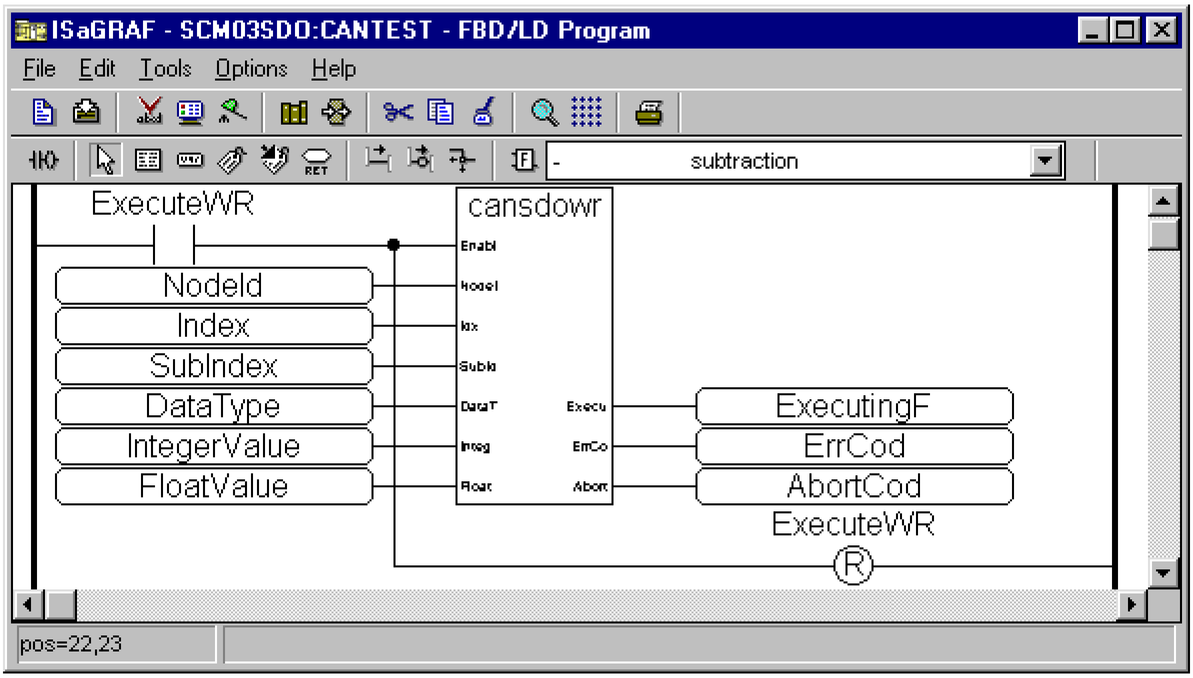


Figure 57

The parameters for the CANSDOWR function block are:

NodeID remote node number

Index address of the CAN object inside the remote node, as defined by the manufacturer

Subindex address of single variable inside the object

DataType one of the data types supported by CAN.

IntegerValue is the value to be written in integer format

FloatValue is the value to be written in float format

Returned values are:

ExecutingF is a flag indicating the operation is still being executed.

ErrCod 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.

AbortCode 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.

5. Internal Controller Hardware Manual

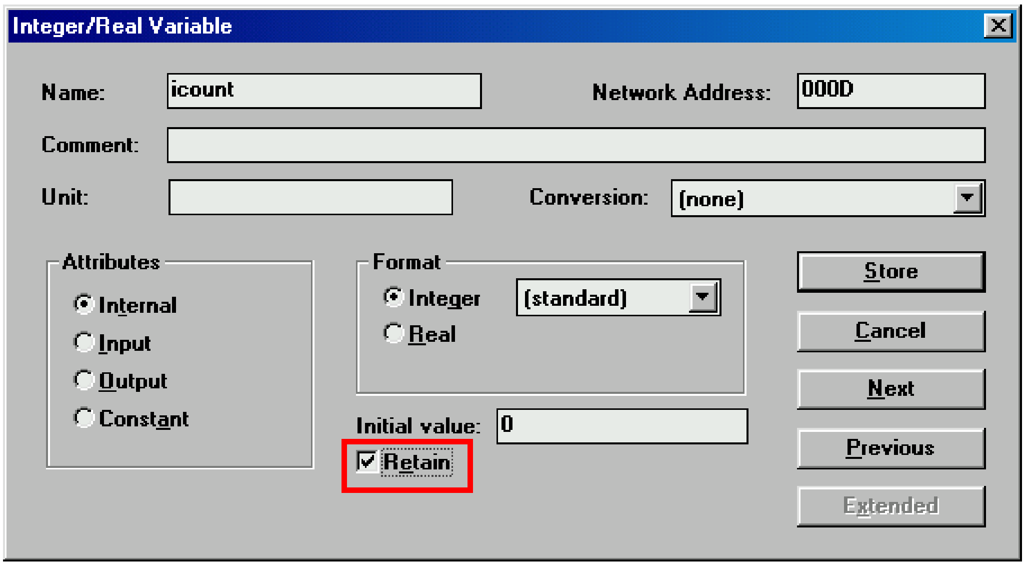
This chapter will describe some implementation-specific issues in the ISaGRAF kernel developed for use with the SCM03 controller module.

5.1 Retentive Memory

The ISaGRAF programming environment allows the user to define variables as **retentive**. The content of retentive 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.*

To configure a variable as retentive, just select the checkbox in the variable definition dialog box of the Tag Dictionary. See figure below.



Integer/Real Variable

Name: icount

Network Address: 000D

Comment:

Unit:

Conversion: (none)

Attributes: ● Internal ○ Input ○ Output ○ Constant

Format: ● Integer (standard) ○ Real

Initial value: 0

☑ Retain

Figure 58

There is a limit to the maximum number of retentive variables that can be defined. The current implementation will support up to a maximum of 2048 bytes.
The memory requirement of individual variables is shown in the table below:

Data type	**Bytes used**
Boolean	1
Integer/Real	4
Timer	5
Message	256

Table 8

As SCM03 has no on-board battery backup, removing the SCM03 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 SCM03 module is moved from one UniOP panel to another

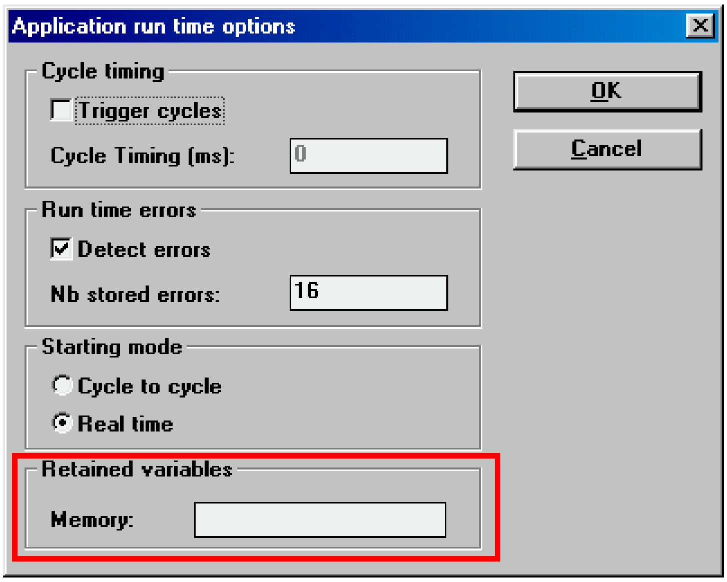
In the case a new firmware version is loaded into the UniOP panel, you will have to cycle power to times before the retentive data can be stored correctly.

In the current implementation the ISaGRAF package does not provide information on the total amount of variables defined as retentive. It will be responsibility of the programmer to ensure that the maximum amount of available memory will not be exceeded.

Some security checks are performed at power-up, to prevent that wrong data is restored on PLC variables: PLC project CRC, Checksum of the total area and total length of the area are controlled. Due to those checks it is impossible to restore data once the project is updated, once the SCM03 board is moved from one UniOP to another or once a new firmware version is loaded into UniOP.

Note: *Retentive Memory mechanism is activated only after 2 power cycles after ISaGRAF project download. After the first download the project starts reporting that "memory can not be allocated"; at the second power cycle the system will start to operate.*

It is **not** necessary to use the 'Retained variables' option in the 'Application run time options' dialog box of the ISaGRAF Workbench. Entering information in this field will have no effect.



Application run time options

Cycle timing

☐ Trigger cycles

Cycle Timing (ms): 0

Run time errors

☑ Detect errors

Nb stored errors: 16

Starting mode

○ Cycle to cycle

● Real time

Retained variables

Memory:

Figure 59

5.2 The CAN Interface

The SCM03 controller module includes 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 signalling
- 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 behaviour 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 where the bus line is always sampled once.

5.3 Watchdog

The SCM03 controller module 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 retrigger 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.*

5.4 Timer resolution

The resolution of ISaGRAF 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.

Appendix A. Internal Controller Features: a Reference Table

Not all features are available in any UniOP model.
For instance, local I/O is only possible in units with hardware extension –0050.
The table below describes the current status of availability of the new features in all UniOP models.

Model	SCM03	Local I/O	
MD00R-02	Yes	-	
MD00R-04	Yes	-	
MD00G-04	Yes	-	
MD03R-02	Yes	Yes	
MD03R-04	Yes	Yes	
MD02F-02	Yes	-	
MD02R-04	Yes	Yes	
CP01R-04	Yes	-	
CP02R-04	Yes	-	
CP01F-02	Yes	-	
CP02F-02	Yes	-	
CP04F-04	Yes	-	
CP10G-04	Yes	Yes	
CP11G-04	Yes	Yes	
CP12G-04	Yes	Yes	
CP13G-04	Yes	Yes	
EF-02	Yes	-	
EF-04	Yes	-	
ER-04	Yes	-	
MKDF-02	Yes	-	
MKDF-04	Yes	-	
MKDR-04	Yes	-	
MKDR-05	Yes	-	
MKDG-06	Yes	Yes	
MKDG-07	Yes	Yes	

Model	SCM03	Local I/O	
ER-16	Yes	-	
EL-16N	Yes	-	
ER-25	Yes	-	
ER-VGA	Yes	-	
EL-25	Yes	-	
EL-VGA	Yes	-	
BKDR-16	Yes	-	
BKDC-16	Yes	-	
BKDL-16	Yes	-	
BKDR-16T	Yes	-	
BKDC-16T	Yes	-	
BKDL-16T	Yes	-	
BKDR-46	Yes	-	
BKDC-46	Yes	-	
BKDL-46	Yes	-	
MKDL-16N	Yes	-	
MKDR-25	Yes	-	
MKDL-25	Yes	-	
MKDR-VGA	Yes	-	
MKDC-VGA	Yes	-	
MKDL-VGA	Yes	-	
MKDT-VGA	Yes	-	
MKDR-VGA-T	Yes	-	
MKDC-VGA-T	Yes	-	
MKDL-VGA-T	Yes	-	
MKDT-VGA-T	Yes	-	
ERT-16	Yes	-	
ECT-16	Yes	-	
ELT-16	Yes	-	
ER-25T	Yes	-	
EL-25T	Yes	-	
ERT-VGA	Yes	-	
ECT-VGA	Yes	-	
ELT-VGA	Yes	-	
ETT-VGA	Yes	-	

Table 9