EXOR

Tech-note

Connecting UniOP to CoDeSys Controllers via Ethernet

This Technical Note contains the information required to connect UniOP panels to controllers based on the 3S CoDeSys PLC system using Ethernet communication.

Contents

Tn242
Ver. 1.02

The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind.

www.uniop.com

1 Introduction

To create a Designer project for communication with controller based on CoDeSys ETH, select the driver “CoDeSys ETH” from the list of available communication drivers in the Configure Controller dialog box.

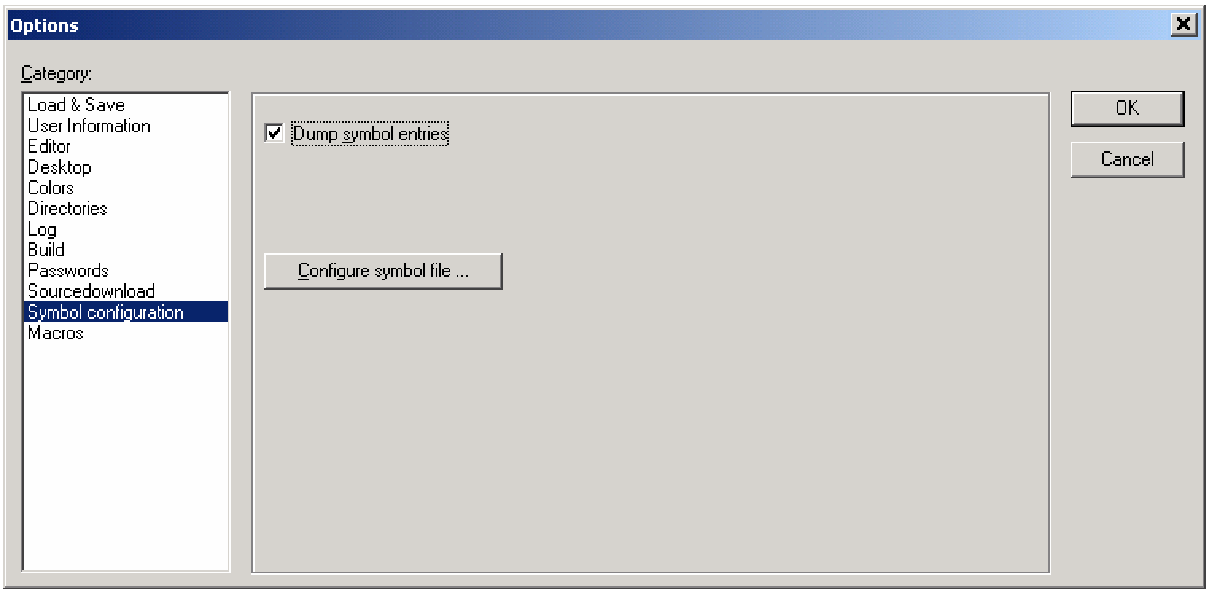
***Note**: In case the CoDeSys ETH is not shown in the list, make sure that the file D32uplc212.dll is present in the Designer folder and then press the Refresh button.*

Starting from the version 5.02 – 4.01 the CoDeSys Ethernet communication driver supports two types of CoDeSys TCP protocols: TCP/IP Level 2 (this is the protocol originally supported) and TCP/IP Level 4.
Starting from the version 5.03 – 4.03 the CoDeSys Ethernet communication driver supports also TCP/IP Level 2 Route.
Selection can be done in Controller Setup as explained later in this document.

2 CoDeSys Parameters and Designer Tags

This chapter describes how to import CoDeSys PLC parameter information into Designer using the TagEditor.

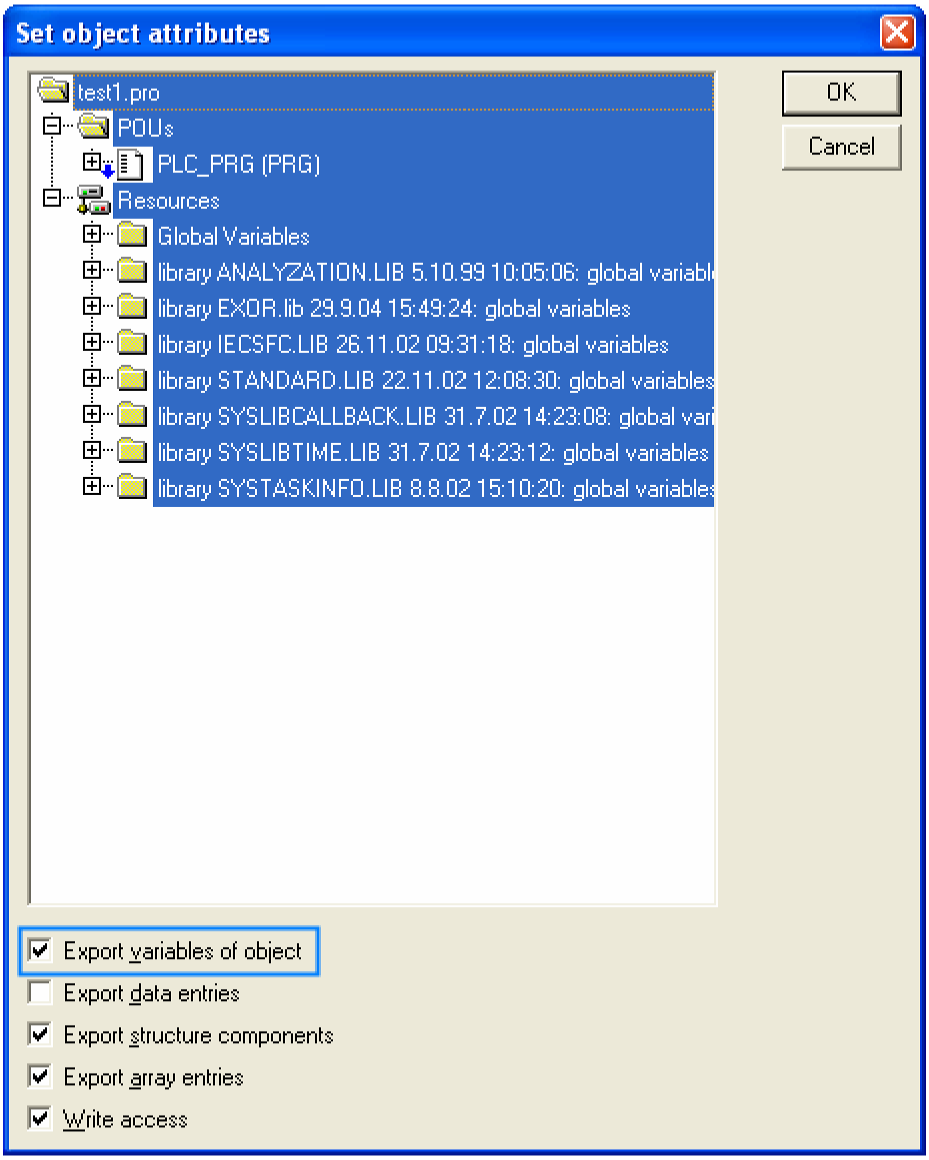
When configuring PLC using the manufacturer’s configuration software, make sure to enable Symbol file creation (file with .SYM extension). It can be done under the CoDeSys programming software, by selecting “Project\Option\Symbol configuration” and mark the check box “Dump symbol entries” as shown in the picture below.



☑ Dump symbol entries

Figure 1 – Symbol configuration

Click then on the “Configure symbol file…” button and make sure the “Export variables of object” check box is marked as shown in the following picture.
We recommend to un-check the check box and mark it again to be sure about the proper settings.



Set object attributes

test1.pro
- POUs
  - PLC_PRG (PRG)
- Resources
  - Global Variables
  - library ANALYZATION.LIB 5.10.99 10:05:06: global variabl
  - library EXOR.lib 29.9.04 15:49:24: global variables
  - library IECSFC.LIB 26.11.02 09:31:18: global variables
  - library STANDARD.LIB 22.11.02 12:08:30: global variables
  - library SYSLIBCALLBACK.LIB 31.7.02 14:23:08: global vari
  - library SYSLIBTIME.LIB 31.7.02 14:23:12: global variables
  - library SYSTASKINFO.LIB 8.8.02 15:10:20: global variables

☑ Export variables of object

☐ Export data entries

☑ Export structure components

☑ Export array entries

☑ Write access

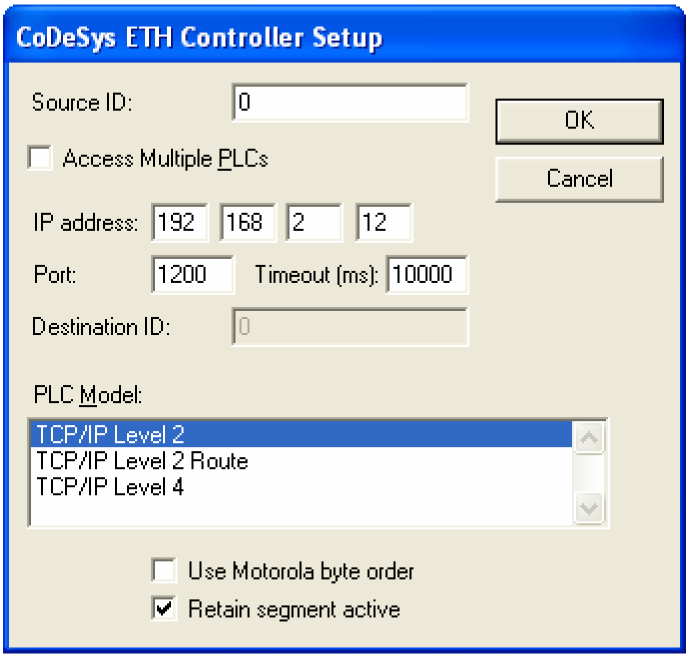
Figure 2

TagEditor can be used to import parameters contained in the file exported by the CoDeSys software. TagEditor will prepare the Tag Database (.MDB file) which can be used in standard way from Designer’s UI to import Tags. For more details on how to import MDB file, please check the Designer manual.

3 Setting-up the Communication with Designer

3.1 Controller Setup

Figure 3 below shows the “Controller Setup...” dialog box for the CoDeSys ETH driver.



Source ID: 0

☐ Access Multiple PLCs

IP address: 192 168 2 12

Port: 1200 Timeout (ms): 10000

Destination ID: 0

PLC Model:

- TCP/IP Level 2
- TCP/IP Level 2 Route
- TCP/IP Level 4

☐ Use Motorola byte order

☑ Retain segment active

Figure 3 – Controller Setup Dialog Box

IP address
Ethernet IP address of the controller connected to the UniOP

Port Number
This parameter allows changing the port number used for the communication. Default value for this parameter is set to 1200 and it corresponds to the default setting of CoDeSys-based controllers.

PLC Model
Shows a list with the available protocol variants. Please make sure you check what protocol variant is supported by the CoDeSys run-time you want to connect.

Use Motorola byte order
Defines the byte order that will be used by UniOP’s communication driver when sending communication frames to the PLC.

This is not applied to the byte order of the Data *Received From/Sent To* controller! To obtain the correct Data byte order, make sure to use appropriate “Data Format”, for instance: WORD or WORD INV. (Data Format can be selected in Data Field Properties dialog box described in chapter 4. Also, check chapter 5 for more details about Data Formats).

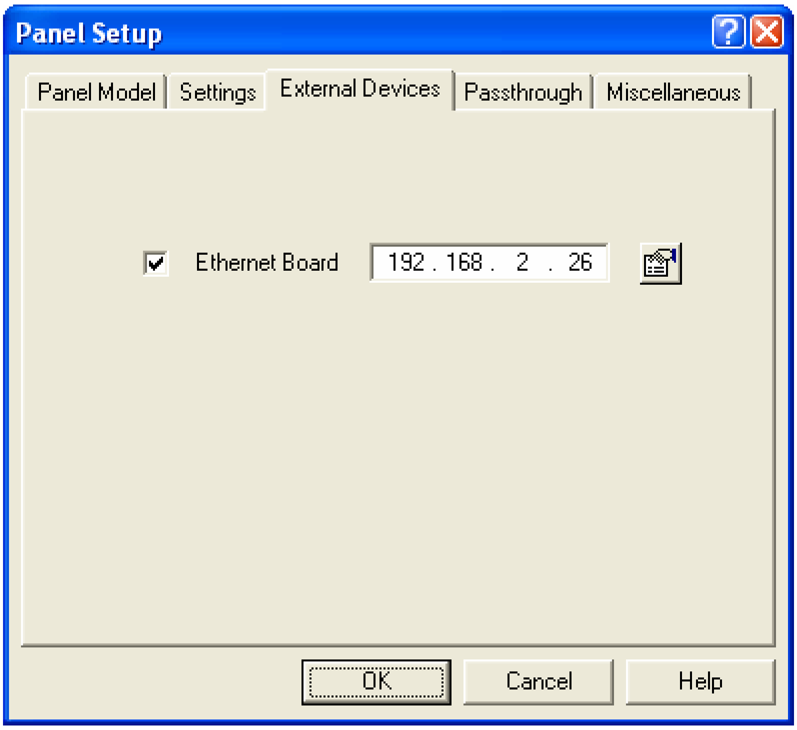
Retain segment active
Activates or deactivates usage of the retain segment defined in POURef 3.

The user must be aware that when this option is checked, no automatic reallocation of already defined fields which uses POURef 3 segment is done. So, in the moment of checking this option, a warning message is displayed.

Access Multiple PLC’s	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check “Access Multiple PLC’s” checkbox and enter IP Address for all controllers.
Source ID &Destination ID	Source and Destination ID are available only when TCP/IP Level 2 Route is selected in Controller Setup. The Destination ID is the node of the PLC and allows the protocol to reads variable in a sub-network. The ID is used to read variables when multiple PLCs are connected in a sub-network (serial network) but only one of it have the Ethernet interface. See chapter for

3.2 Panel Setup

The HMI panel must have a unique IP address assigned to it. The IP address for the device may be entered in the Panel Setup dialog box under the tab “External Devices”. Enter the IP address in the field “Ethernet Board” as shown in Figure 3 below.
Alternative methods of specifying the panel’s IP address are also available.

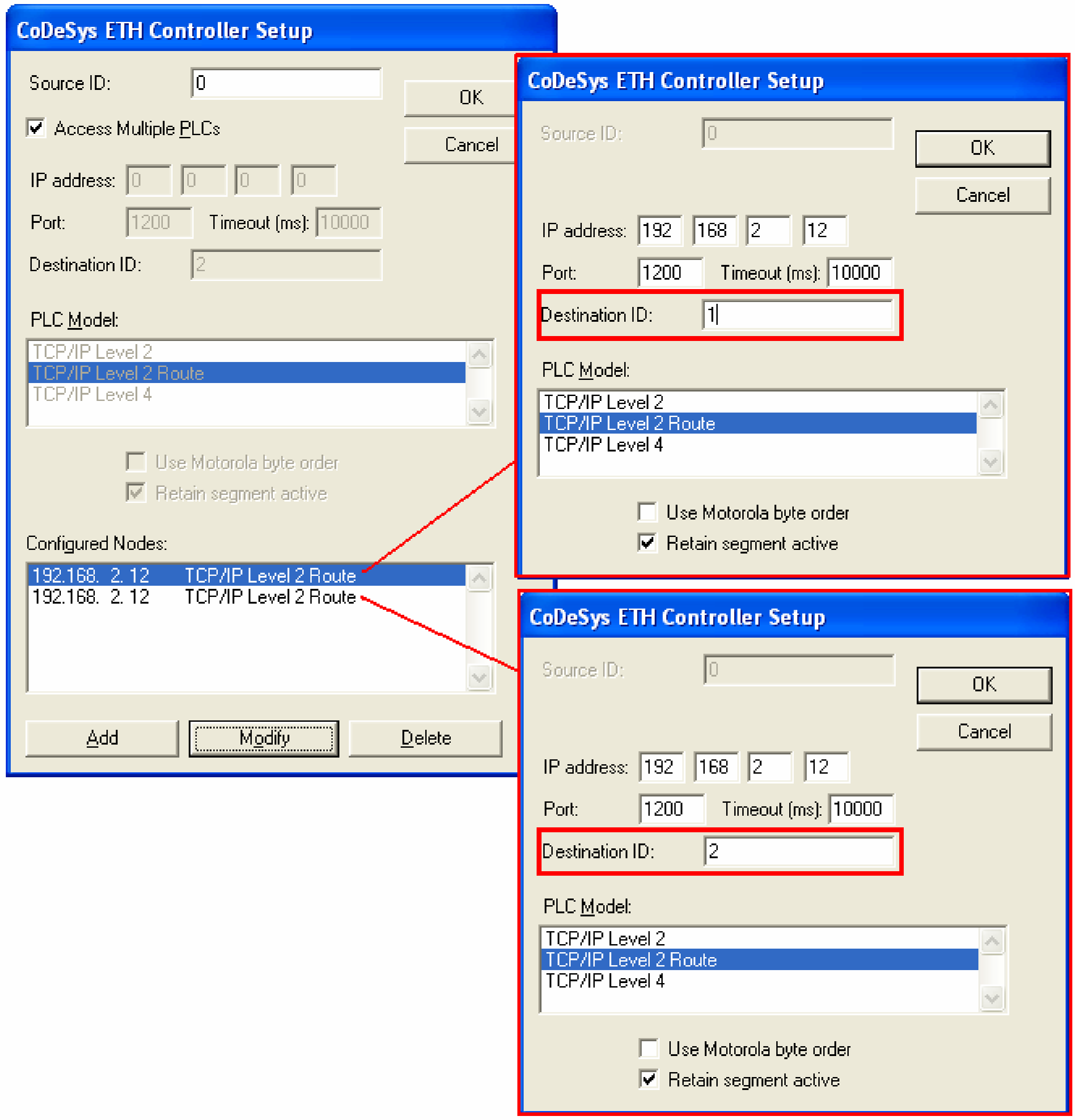


☑ Ethernet Board 192 . 168 . 2 . 26

Figure 4 – Define panel IP Address

3.3 Access Multiple Controllers, Source ID and Destination ID

The Source ID and Destination ID are used only with “TCP/IP Level 2 Route” controller model.
The typical situation in which Source ID and Destination ID are required, is a sub network made of controllers, interconnected between them, and having one unique Ethernet interface with one unique IP. The “Destination ID” is the controller identifier in the sub-network.
Figure 5 shows an example.



CoDeSys ETH Controller Setup

Source ID: 0

☑ Access Multiple PLCs

IP address: 0 0 0 0

Port: 1200 Timeout (ms): 10000

Destination ID: 2

PLC Model:
TCP/IP Level 2
TCP/IP Level 2 Route
TCP/IP Level 4

□ Use Motorola byte order

☑ Retain segment active

Configured Nodes:

```
192.168.  2. 12      TCP/IP Level 2 Route
192.168.  2. 12      TCP/IP Level 2 Route
```

CoDeSys ETH Controller Setup

Source ID: 0

IP address: 192 168 2 12

Port: 1200 Timeout (ms): 10000

Destination ID: 1

PLC Model:
TCP/IP Level 2
TCP/IP Level 2 Route
TCP/IP Level 4

□ Use Motorola byte order

☑ Retain segment active

CoDeSys ETH Controller Setup

Source ID: 0

IP address: 192 168 2 12

Port: 1200 Timeout (ms): 10000

Destination ID: 2

PLC Model:
TCP/IP Level 2
TCP/IP Level 2 Route
TCP/IP Level 4

□ Use Motorola byte order

☑ Retain segment active

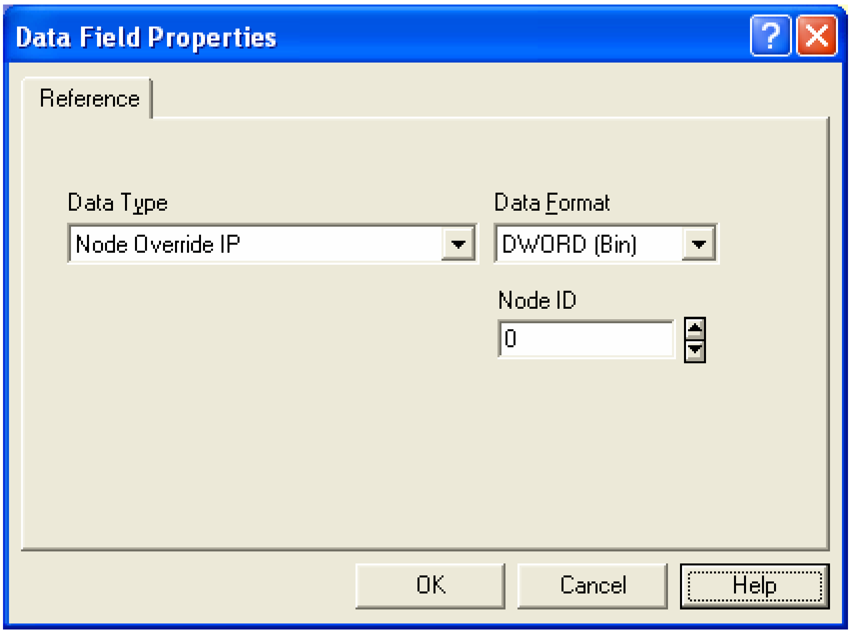
Figure 5

4 Node Override IP

The Node Override IP is a function that allows changing in run time the IP addresses of any of the controllers configured in the PLC network.

Note: *the Node Override IP function requires driver version 5.04 – 4.04 or higher*

The “Node Override IP” is an internal variable of the communication driver; the Data Field Properties dialog box is shown in Figure 6. Each of the devices configured in the PLC network has it own Node Override IP variable. The node override variables can be used in the Designer project as normal numeric items; they have a default value matching the IP address assigned to individual controllers in the PLC network configuration. The value of the Node Override variables can be changed at run time to connect different IP addresses.



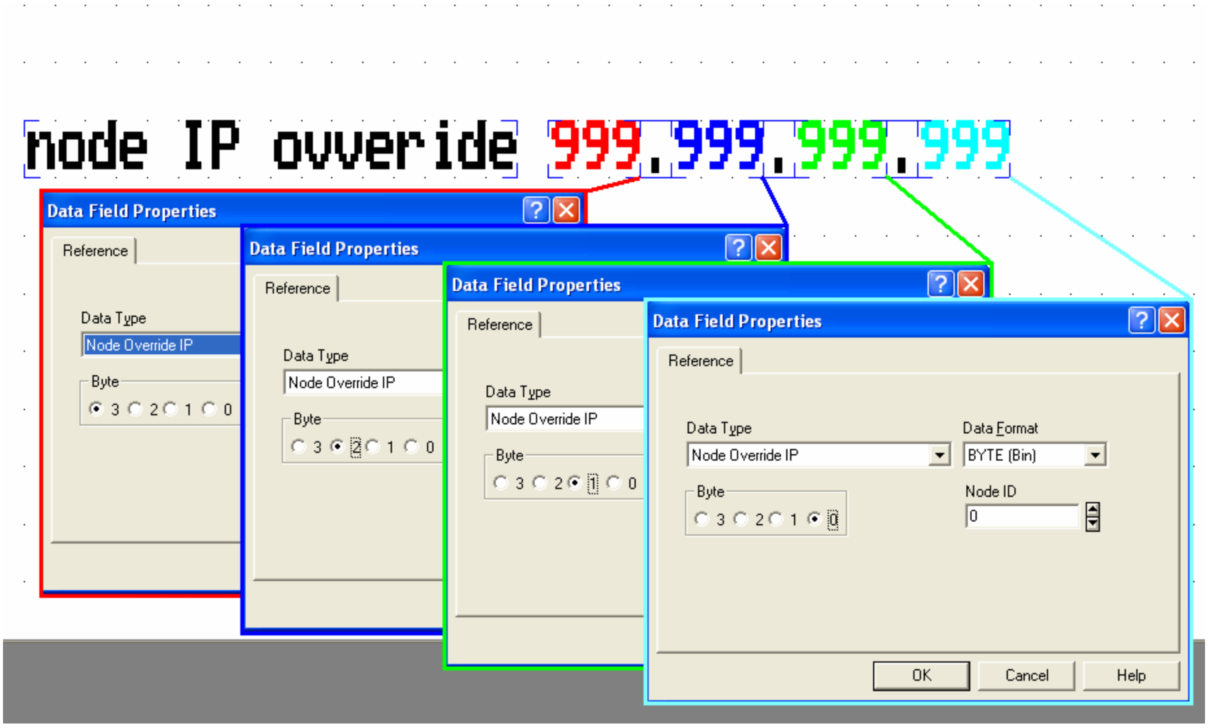
Data Type: Node Override IP

Data Format: DWORD (Bin)

Node ID: 0

Figure 6 – The Data Field Properties dialog box

The Node Override IP object is a double word; the 4 bytes correspond to the 4 bytes of the IP address as explained in Figure7.



Data Type: Node Override IP — Byte: ● 3 ○ 2 ○ 1 ○ 0

Data Type: Node Override IP — Byte: ○ 3 ● 2 ○ 1 ○ 0

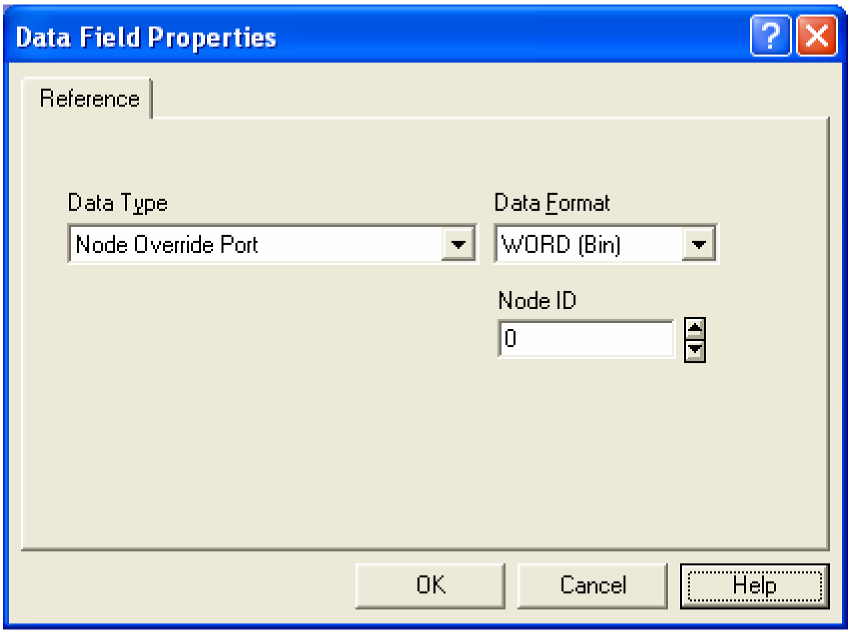
Data Type: Node Override IP — Byte: ○ 3 ○ 2 ● 1 ○ 0

Data Type: Node Override IP — Data Format: BYTE (Bin) — Byte: ○ 3 ○ 2 ○ 1 ● 0 — Node ID: 0

Figure7 – Designer project example showing the bytes of the IP address

4.1.1 Node Override Port

“Node Override Port” allows changing at run time the Port number of a controller configured in Controller Setup. This object has a default value set to 1200 (Figure8).



Data Field Properties

Reference

Data Type: Node Override Port

Data Format: WORD (Bin)

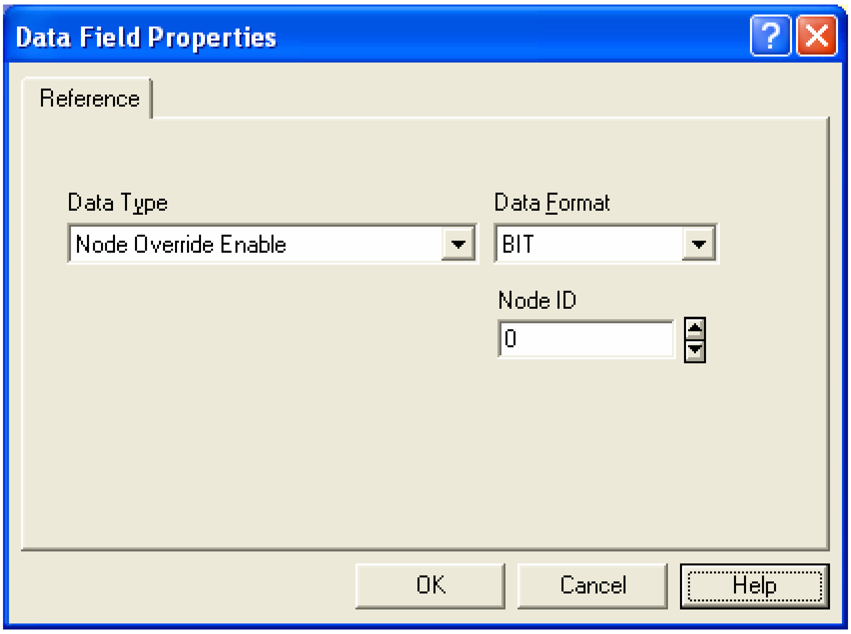
Node ID: 0

Figure8 – The Data Field Properties dialog box

Similarly to the “Node Override IP”, each controller configured in the PLC network, has its own “Node Override Port”.

4.2 Node Override Enable

The “Node Override Enable” is an internal variable of the communication driver which is required to enable/disable the IP and Port override as programmed in the “Node IP Override” and “Node Override port” variables. The Data Field properties dialog is shown in Figure 9.
Each controller configured in the PLC network has its own “Node Override Enable” variable.
When the “Node Override Enable” is set to “true”, it activates the use by the communication protocol of the controller target IP specified by the “Node Override IP” and “Node Override port” variables.
When the “Node Override enable” is set to “false” the target IP and “Node Override port” for the controller is the one specified in the project at programming time.



Data Type: Node Override Enable

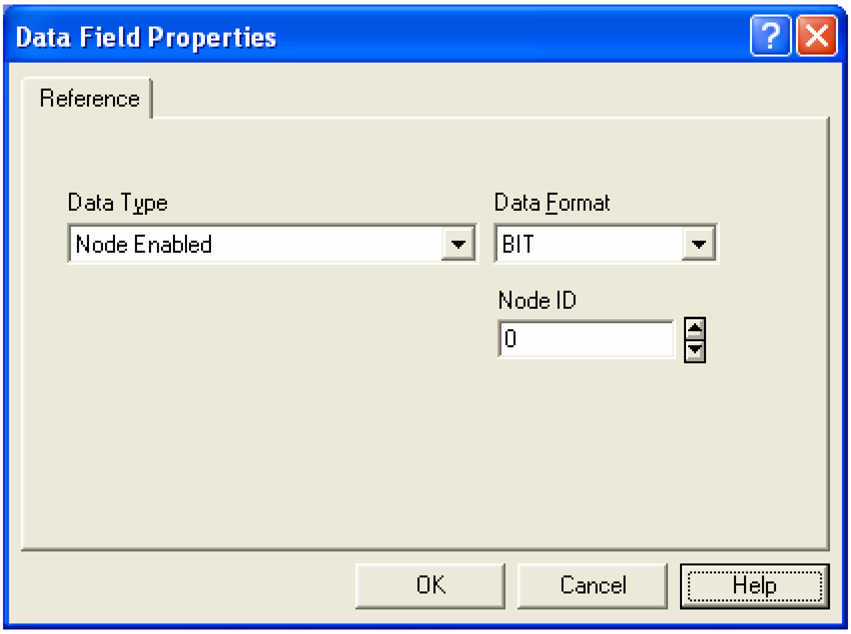
Data Format: BIT

Node ID: 0

Figure 9 – The Data Field Properties dialog box

5 Node Enable

The "Node Enable" variable allows disabling the connection to a certain controller. This can be useful when the Designer project is configured for several nodes and some of them are not physically present in the network for any reason. After disabling the communication with those nodes, the rest of the project will operated properly without the performance being influenced by the missing nodes (Figure 10).



Data Type: Node Enabled

Data Format: BIT

Node ID: 0

Figure 10 – The Data Field Properties dialog box

Note: *all protocol system variables "Node Override IP", "Node Port override", "Node Override enable", and "Node enable" are stored in RAM memory with battery back-up.*

6 The Data Field Properties Dialog Box

The CoDeSys ETH Data Field Properties dialog box is shown in the figure below.

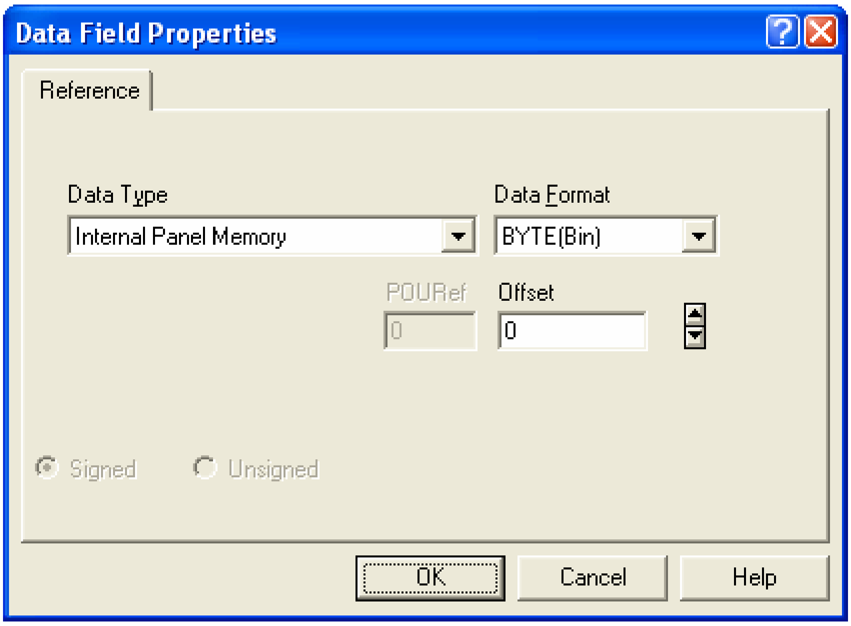


Figure 4 – The Data Field Properties dialog box

The "Data Type" combo box is used to select the memory area to be referenced.
The following memory areas can be selected:

Name	Offset range
Internal Panel Memory	0-511
PLC Memory	0-2147483646
PLC Input	0-65535
PLC Output	0-65535
PLC Retain	0-65535
PLC Parameters	0-65535

Internal Panel Memory
This is a memory area internal to the UniOP panel and does not have any relation with the controller's memory. Any access to this area does not generate any message on the serial line.

PLC Memory, PLC Input, PLC Output, PLC Retain
This is the controller's general-purpose memory. Code number (POURef) for PLC Retain is adjustable in Controller Setup.

PLC Parameters
In order to access PLC Parameters, the user needs to import parameters from the symbol file, as described in previous chapter.

7 Data Formats

The CoDeSys ETH communication driver handles standard as well controller-specific data formats, such as those needed for the TIME and POINTER data types.

All standard data types are available both in ‘normal’ and ‘inverted’ low-level data storage. This will make possible to connect to controllers that are based both on high-endian and little-endian data formats.

Some comments on special data types are given below.

1) Designer provides the special Data Format “TIME” and “TIME INV”. Values read from the PLC will be displayed in the following format: 1h2m3s4ms.
Data entry for TIME variables can be done only using macro command “STEP” available in Designer Macro Editor, with appropriate step sizes:

	Use step size:
hours	3600000
minutes	60000
seconds	1000
milliseconds	1

2) Designer will treat POINTER type data as DWORD when presenting its Data Format in the user interface.

Please remind that special data formats do not support scaling and cannot be used in UniNet clients.

8 RDA, Alarms, Mailbox and Data Transfer

All available data types can be used as reference for RDA, Alarms, Mailbox and Data Transfer.
NOTE: Data Type PLC Parameters can be used for such purpose only if PLC’s memory is configured in appropriate way. To check this, see Designer manual and appropriate memory mappings for each of these purposes.

Appendix A. Communication Error Codes

Current communication status is displayed on the system page of the UniOP. Beside the string, describing current state of the communication (OFF, ON, ERR), there is an additional error code representing the last (which may be not the current one) error encountered. The codes are:

Code	Description	Notes
00	No error	There are no communication errors and there have been no errors since start-up.
04	Negative ACK	NAK returned by the controller while UniOP is waiting for ACK, or invalid data in frame.
05	Time out (receiving)	No response received form the controller within the timeout period.
06	Response error	Wrong response telegram structure.
07	General communication error	General unknown communication error.
09	Send Timeout	Timeout when sending data frame to controller
12	Checksum error	Wrong checksum received from the controller.