Tech-note

# Connecting UniOP to Beckhoff ADS Ethernet

Connecting UniOP to Beckhoff controllers via ADS Ethernet.

**Sitek S.p.A.**
**ptn0303**
**Ver. 1.0**

**Copyright © 2009 Sitek S.p.A. – Verona, Italy**
Subject to change without notice
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.
Third-party brands and names are the property of their respective owners.
www.exor-rd.com

# Contents

1 Introduction ....4

2 Setting up UniOP ....4
2.1 The Controller Setup ....4
2.1.1 Access to Multiple Controllers ....5
2.2 Designer Panel Setup ....6
2.3 Tag Editor ....7
2.3.1 Importing the Tag File ....9
2.3.2 The Define Field Dialog Box ....11
2.3.3 Downloading the Project ....12

3 RDA, Alarms, Mailbox and Data Transfer ....13
3.1 Using the RDA ....13
3.2 Programming the Mailbox ....14
3.3 Alarms ....15
3.4 Transferring Data with the Internal Controller ....15

Appendix A. Communication Error Codes ....16

Appendix B. Requirements and Compatibility ....17

# 1 Introduction

The protocol Beckhoff ADS Ethernet is implemented in the D32UPLC167.DLL driver file.
The protocol has been implemented according to the published Beckhoff ADS specifications. Implementation on the ADS Ethernet protocol for access to Beckhoff controller data is based on the information published by Beckhoff.

# 2 Setting up UniOP

To create a UniOP application for Beckhoff ADS Ethernet, select the driver “Beckhoff ADS ETH” from the list of available communication drivers in the Change Controller Driver… dialog box.

***Note:*** *Designer 6.07 (01) or higher is required for Beckhoff ADS ETH protocol.*

## 2.1 The Controller Setup

The system can be composed by one or more Beckhoff systems, each of them equipped with proper Ethernet interface card. Each Ethernet card has its own IP address. Controller setup is divided in two parts:

**Souce – Panel**
Source – Panel contains the AMS Net ID assigned to the HMI, the first 4 bytes of the AMS Net ID must be the same of the IP address choose for UniOP in Designer Panel Setup.

**Target - Controller**
You can enter the IP address of the Beckhoff controller Ethernet card you what to connect and its proper AMS Net ID.

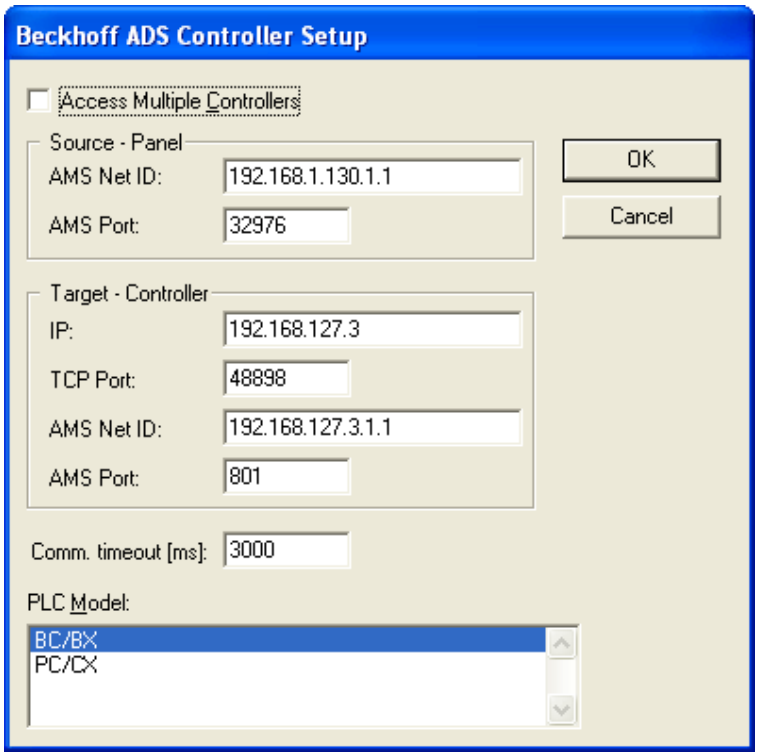


Figure 1

Figure 1**Error! Reference source not found.** shows the Controller Setup dialog configured for accessing to only one controller. The AMS Net ID assigned to the UniOP is 192.168.1.13.1.1. The PLC Ethernet card has the IP address 192.168.127.3, its AMS Net ID is 192.168.127.3.1.1

***Note:*** *For a correct communication with the device, please select the appropriate PLC model family.*

### 2.1.1 Access to Multiple Controllers

In the case the complete control system is composed of more than one Beckhoff Controller, the “Access Multiple Controllers” option can be used to configure the Designer project to access to each individual controller. In this case the IP address specified in the “PLC Setup” dialog must be different for each Ethernet board.

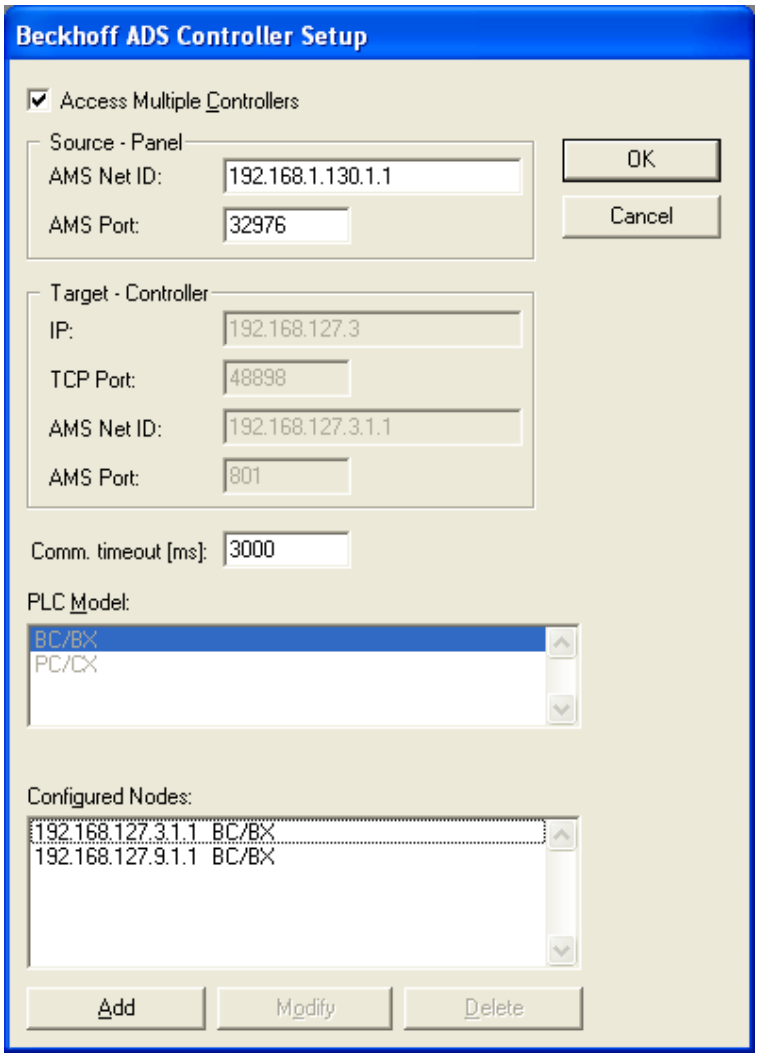


Figure 2

Figure 2 shows the case of a system with two Beckhoff controllers, the two Ethernet cards have IP address respectively 192.168.127.3 and 192.168.127.9.The same is for AMS Net ID, 192.168.127.3.1.1 for the first controller in the list and 192.168.127.9.1.1 for the second.

## 2.2 Designer Panel Setup

The operator panel must have an IP address to connect to Ethernet.
The IP address of the panel must be entered in Panel Setup dialog box in the tab “External Devices”. Enter the IP address in the “Ethernet board” field as shown in Figure 3 below.
IP address for UniOP can also be assigned as fixed from Configuration Mode. Please see Designer online help for further information about setting panel IP address.

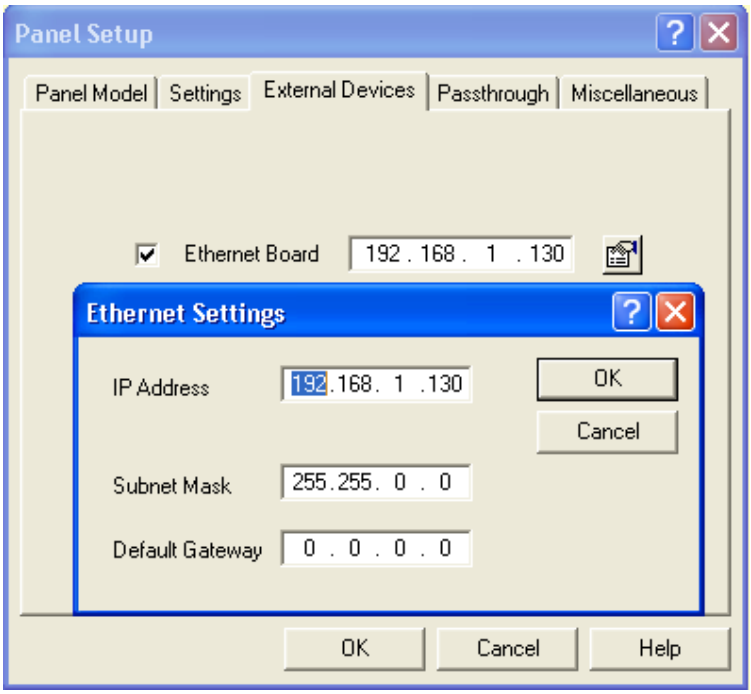


Figure 3

***Note:*** *The IP address must be the same with the first 4 bytes assigned to the AMS Net ID in Controller Setup, Source – Panel.*

## 2.3 Tag Editor

The data in the Beckhoff system is based on tags.
The organization of the internal memory of the controller is not fixed but it is configured by the user at development time. Each data item can be identified by a string called “tag”.
The TwinCAT development environment generates the list of tags created for each controller in the configuration of the application.
The project in the panel must refer to the tag names assigned in the TwinCAT PLC Control programming software at development time. The Designer Tag Editor supports direct import of the tag file generated by the Beckhoff software.

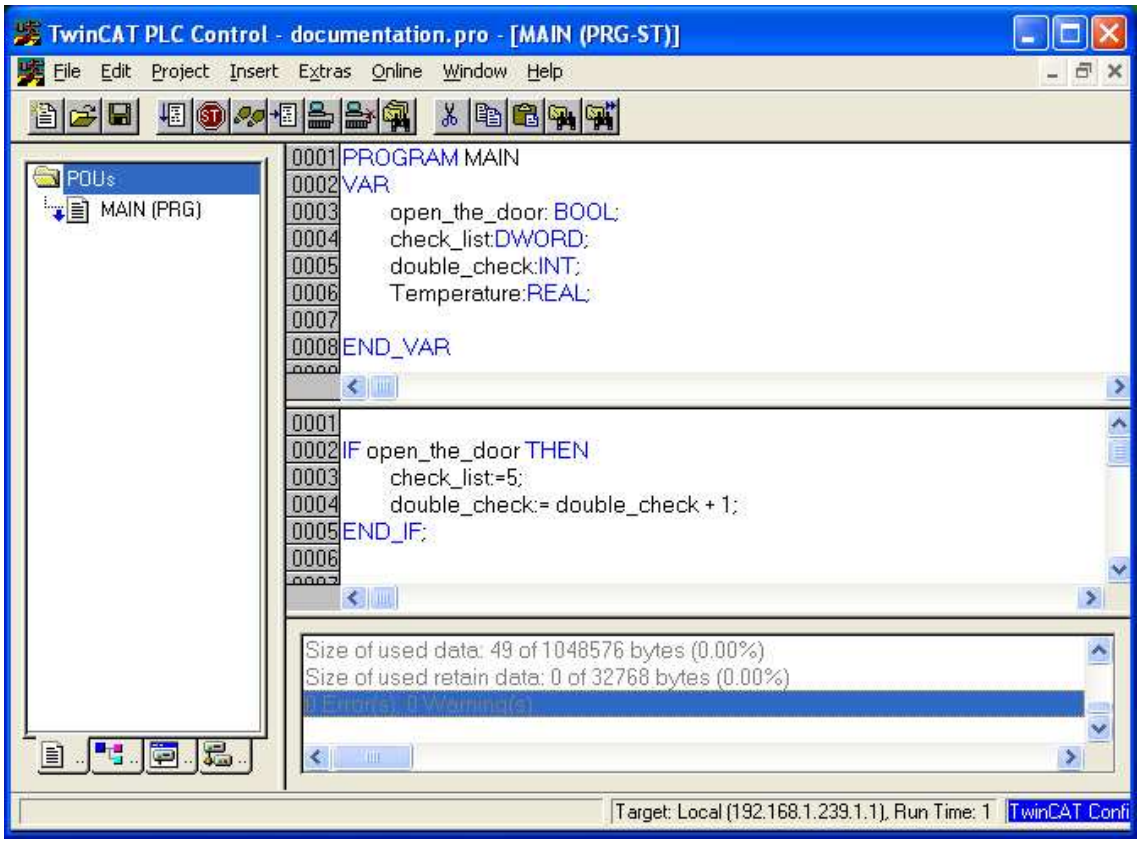


Figure 4

Figure 4 shows an application created for one Controller.
To export the tags defined for the selected controller, press on **Project** – **Build**" as shown in Figure 5.
The TwinCAT PLC Control software will create a file in TPY format.

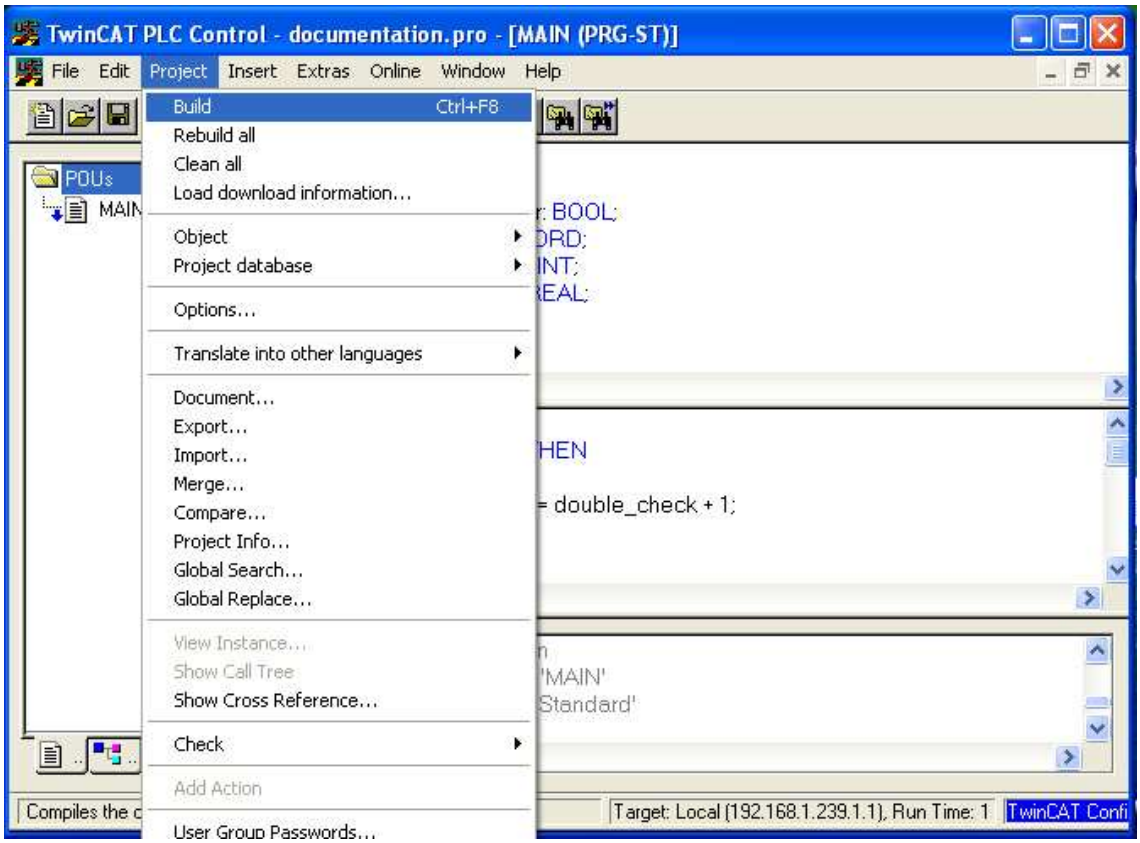


Figure 5

### 2.3.1 Importing the Tag File

The tag file exported from the TwinCAT PLC Controller software can be easily imported in Designer Tag Editor selecting the “Import tags” command from the “File” menu of the Tag Editor. The first step of the import process is shown in Figure 6; in the list of the available controllers the Beckhoff ADS ETH is listed.

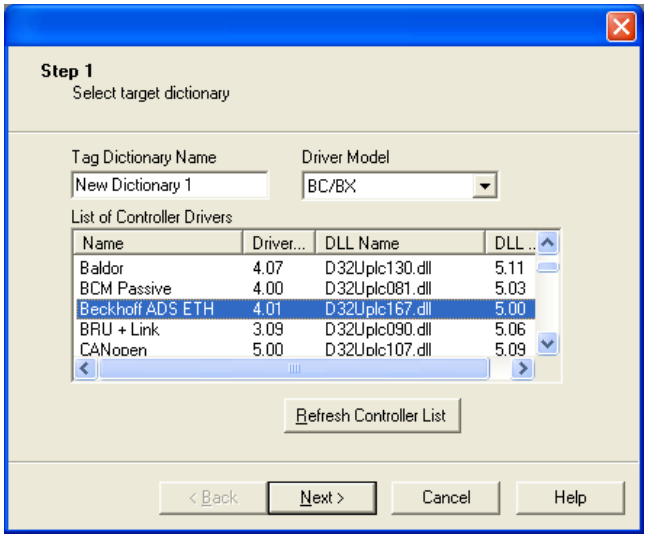


Figure 6

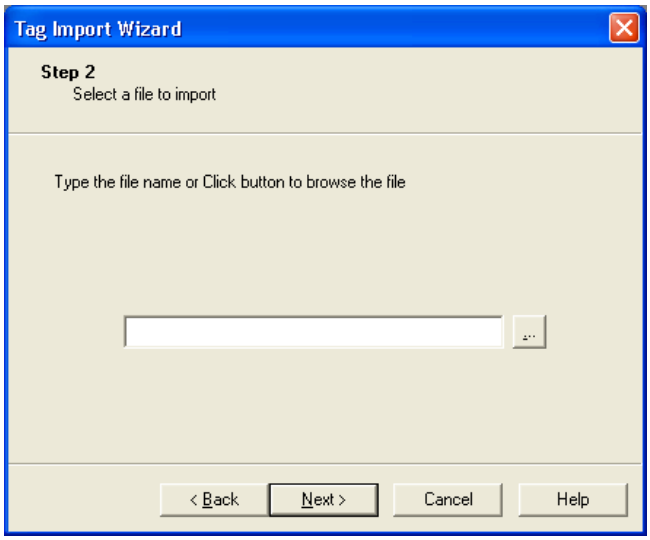


Figure 7

If more than one controller is present in the configuration, Designer project needs to be properly configured.
As the TwinCAT PLC Controller software does not include in the tag file any information on the controller owning the exported tag list, the Designer Tag Editor needs to associate explicitly the tags to the appropriate processor. The selection dialog is shown in Figure 8 below.

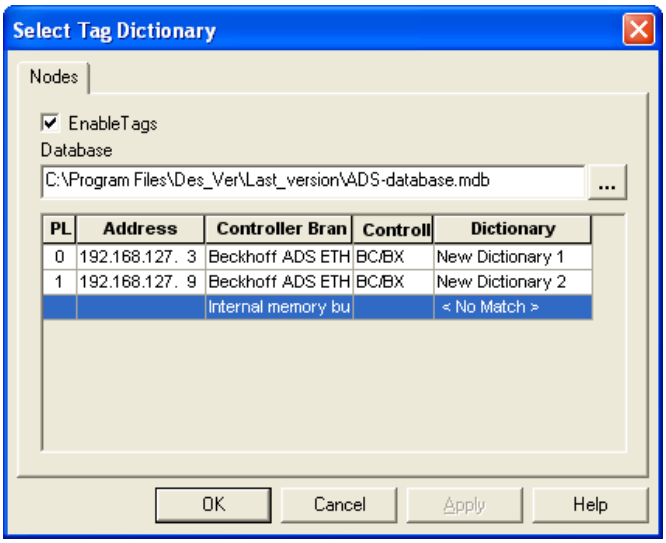


Figure 8

In case the Designer project must be configured to connect to more than one controller, two Dictionaries must be present into the Tag Editor database.

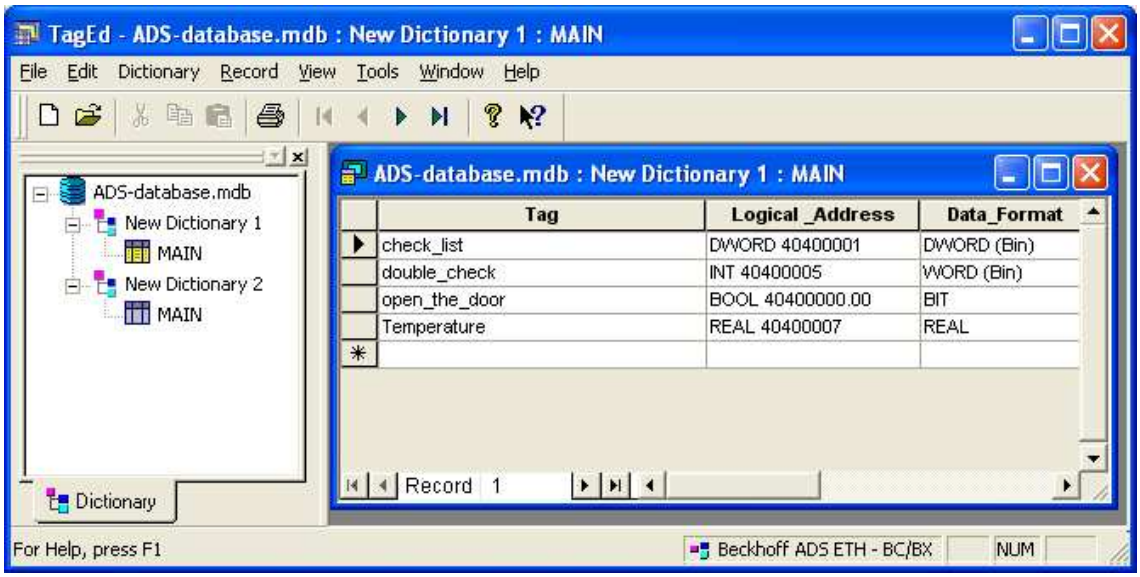


Figure 9

Figure 10 shows the case of two controllers. The first PLC with IP address 192.168.127.3 and the second PLC with IP address 192.168.127.9.

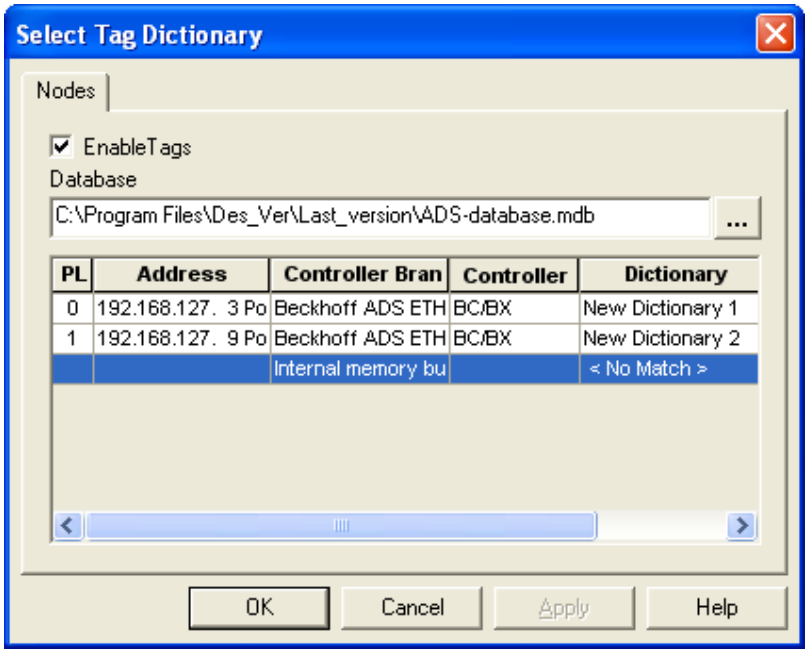


Figure 10

### 2.3.2 The Define Field Dialog Box

When working with Designer software you can create data fields connected directly to any of the controller tags present in the dictionary.
Figure 11 shows the Define Field dialog box. The “Enable Tag” button is available to enter the name of the item.

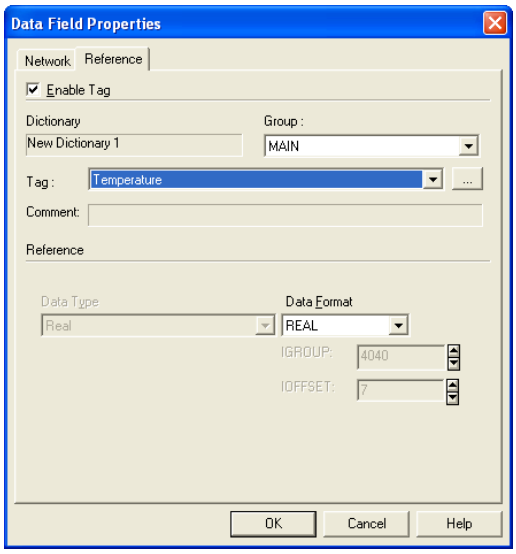


Figure 11

Figure 11 shows the Define field dialog box automatically filled by Designer after a tag item has been selected.

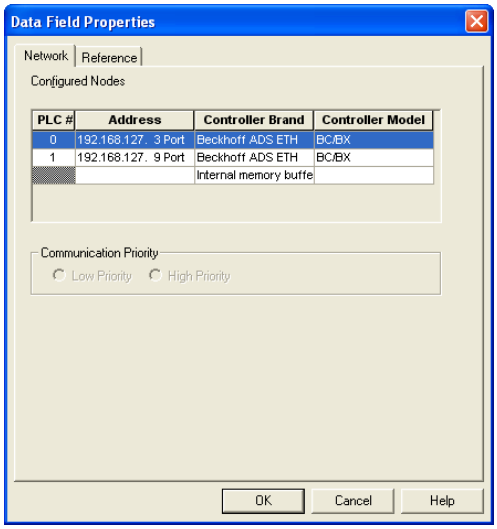


Figure 12

The “PLC Slave ID” specifies the controller that contains the requested data item.

### 2.3.3 Downloading the Project

Since the UniOP protocols needs for communication with the controller the name of the tags used in the project, the tag items have to be downloaded to the panel. Before starting the project download the “Download tags (New)” options available from the Transfer advanced menu has to be checked ( normally is checked by default ). See Figure 13.

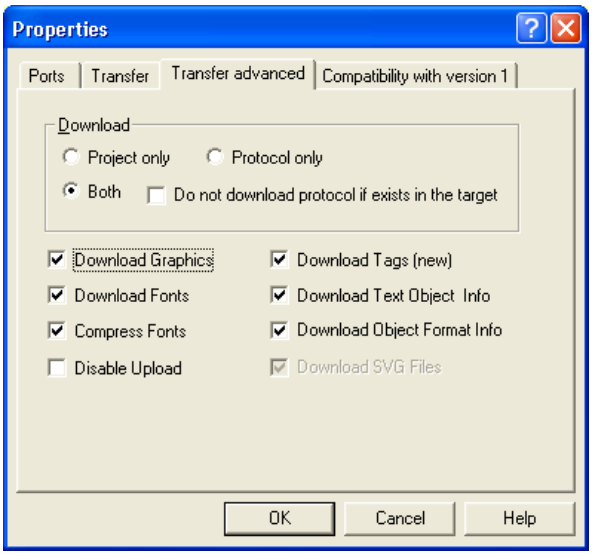


Figure 13

***Note:*** *Upload from panel of Designer projects using Tags is not supported*

# 3 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 Integer 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.

## 3.1 Using the RDA

The Reserved Data Area can be configured in the memory of PLC.
To use the RDA a certain number of Tag's with contiguous addresses must be configured in the TwinCAT program.
The easiest way to declare a list of variables in TwinCAT, which can be considered "contiguous", is to use an array as shown in Figure 14 and address the RDA using the tags.

```
0028
0029  RDA: ARRAY[1..36] OF USINT;
0030  Alarms:ARRAY[1..4] OF USINT;
0031  Mailbox: ARRAY[1..40] OF USINT;
0032
```

Figure 14

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

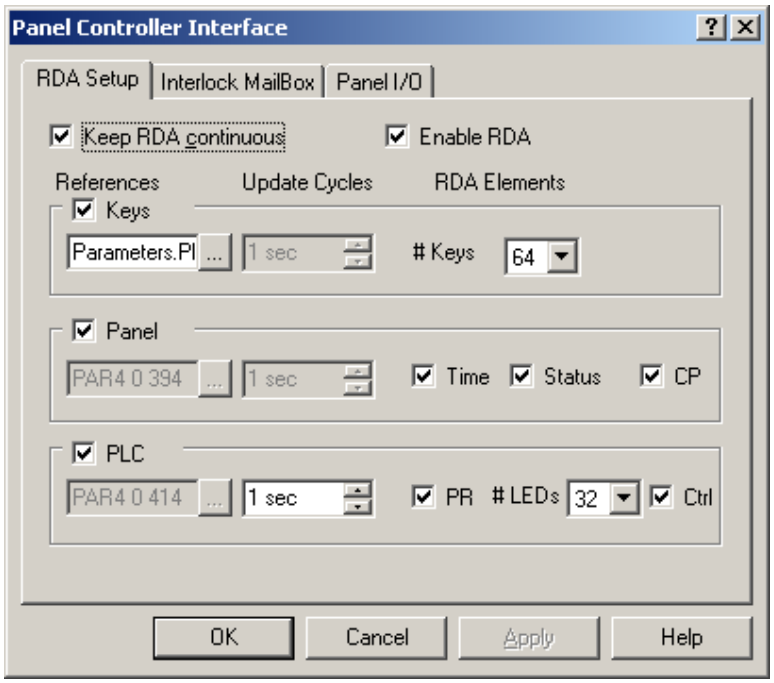


Figure 15

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

Considering the example of variable declaration shown in Figure 14, the Keyboard area is mapped as shown in the Figure 16.

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

Figure 16 – The Keyboard Area

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

Figure 17 – The Panel Area

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

Figure 18 – The PLC Area

## 3.2 Programming the Mailbox

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

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

Figure 19 – The Mailbox

## 3.3 Alarms

The Alarm area in the Controller memory is organized as bytes. An array of bytes (USINT) can be configured to handle Alarms.
Alarm bits are organized according to Figure 20.

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

Figure 20 – The Alarm Area

## 3.4 Transferring Data with the Internal Controller

The UniOP Data Transfer function can be used to copy data from an external controller to another controller.

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

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

Before starting a copy operation the Data Transfer module checks the byte order convention used by the Source and the Target addresses. In UniOP the so-called Intel data format (little-endian) is considered not inverted; the Motorola format (big-endian) is considered inverted.
When Source and Target are both inverted (or not inverted) the Data Transfer module does not apply any transformation.
If Source and Target have different byte ordering, the Data Transfer module applies a byte swap according to the rules explained in Figure 21.

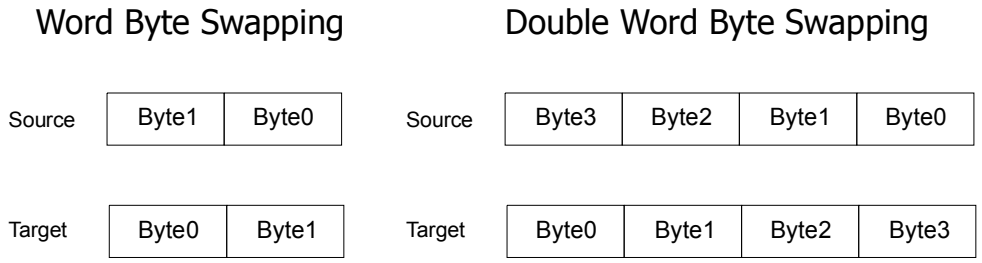


Figure 21

# Appendix A. Communication Error Codes

Current communication status is displayed in the System Menu of the UniOP.
A message and a numeric error code describe the error status.
The message reports the current communication status. The number shows the code of the current communication error or, if the communication is correct, the code of the last error encountered. When the error code 0 is shown, it means there have been no communication errors since this system start-up.

| Code | Description | Notes |
|---|---|---|
| **00** | No error | There are no communication errors and there have been no errors since start-up. |
| **04** | Negative acknowledgement to request | The request sent to the PLC was not valid and the PLC responded with a NAK. This can occur because:<br>a) the request was trying to access a tag that doesn't actually exist in the PLC<br>b) the request was trying to access an array element beyond the maximum array size for the tag<br>c) the request was trying to write a value to the tag outside the allowed range for the tag type<br>d) the request was trying to write a value to a read only type |
| **05** | Failed to create a connection with the Controller | The Forward Open request failed. Perhaps the PLC has too many open connections |
| **06** | Unexpected response length | Means that the response frame was received from the PLC but the frame length was not what was expected. |
| **08** | Failed to decode the PLC response or the PLC did not respond to the request within the timeout interval | It is the typical timeout error caused by a disconnected communication cable or tags not downloaded with the Designer project. |
| **10** | Unexpected response format | Indicates that the response frame was received from the PLC but the frame format was not what was expected. |
| **12** | Unexpected response data length | Indicates that the response frame was received from the PLC but the length of its data part was not what was expected. |
| **16** | Low level protocol error | Should never happen. Contact the manufacturer. |

# Appendix B. Requirements and Compatibility

This communication driver is included in the Designer DLL file D32UPLC167.DLL.

The initial release level is 4.02 for the communication driver and 5.00 for the DLL (both version numbers can be seen in the Change Controller Driver dialog box of the Designer software).

Designer software version 6.07 (01) or higher is required for supporting tags.

UniOP requires the optional module SCM11 or SCM12 to support TCP/IP communication.
The SCM11/12 modules require the ETADxx adapter to convert the 9-pin connector of the AUX port to the standard Ethernet 10BASE-T RJ45 connector. The adapter is delivered with the communication module.

Panels with integrated Ethernet interface are already enabled to run this communication driver.

The supported Ethernet physical layer is 10BASE-T and 10/100BASE-T on units with integrated Ethernet interface with exception of eTOP02C and .eTOP04C where the Ethernet port is 100BASE-T only.

Direct Access is not supported.