Using UniOP with Profibus DP

This Technical Note contains all the information required to connect the UniOP panels to a Profibus DP system with any master system and to take advantage from the flexibility and easy programmability of the UniOP Profibus DP communication driver. This Profibus driver is commonly referred to as Profibus DP Lite.

***Important**: this Technical Note applies only to the Profibus DP communication driver identified by the name 'Profibus DP' and associated to the Designer file D32UPLC97.DLL. To run this communication driver it is required a panel of hardware type –0045 equipped with a communication module type TCM08.*

Contents

Tn097
Ver. 1.06

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.

www.exor-rd.com

1. Introduction

The Profibus DP driver allows a UniOP panel to be connected to any Profibus Master System via the Profibus DP port. This technical note describes the principal points to follow for a successful connection.

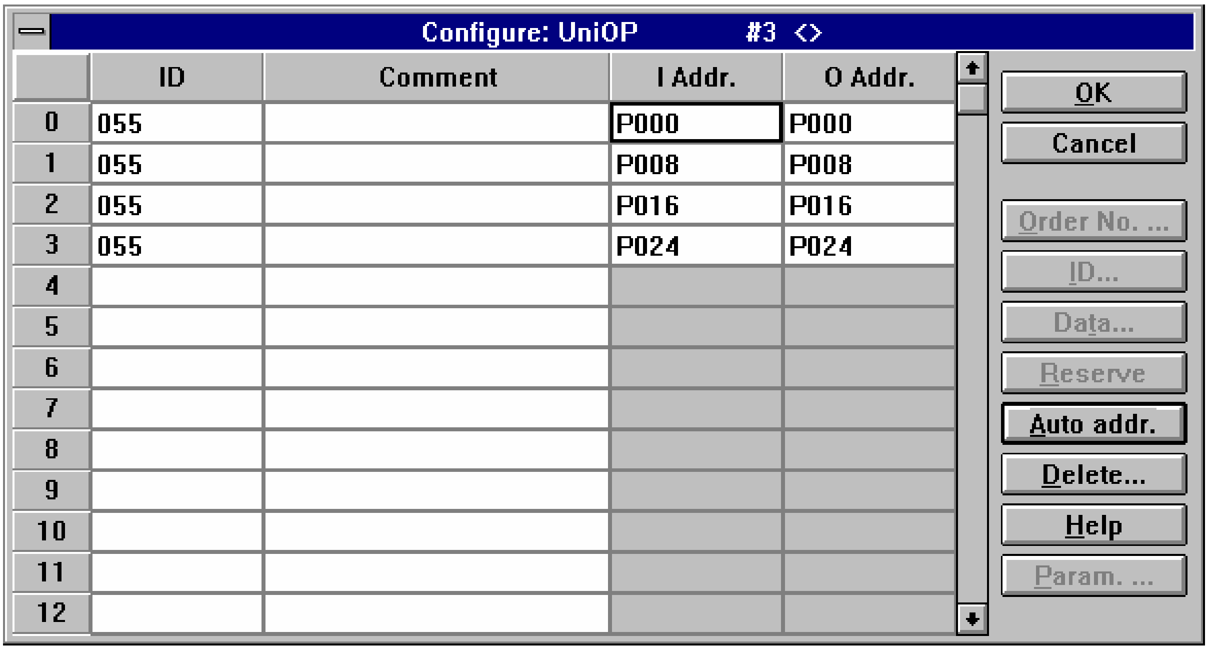
A Profibus DP network can contain multiple nodes. A node in a Profibus DP network can be either a Master or a Slave. The Masters in the network have a group of Slaves assigned to them. A Master is able to exchange data with the Slaves that are under its control. The Master always controls the data exchange with its slaves. The data exchange process to a single Slave involves 2 steps. The first step is the transmission of a request from the Master to the Slave. This request contains the Output data that the Master wants to send to the Slave. The second step is the transmission of the response from the Slave back to the Master. This response contains the Input data that the Slave wants to pass back to the Master.

The Master system can be either a PLC or a personal computer running specialized control software and equipped with a Profibus DP master interface module.
UniOP is always a Slave in a Profibus DP network; it is only able to exchange data with a single Master. With the Profibus DP driver, UniOP can be configured to use 8, 16 or 32 bytes of Input and Output data. The Output data sent from the Master is stored by UniOP in an internal buffer. UniOP can be programmed with the Designer package to display this Output data in whatever way is appropriate. Similarly, UniOP stores the Input data to be sent to the Master in another internal buffer. UniOP allows the Input data to be sent to the Master to be edited by the UniOP user.

2. Configuring UniOP as a Slave

The Master System must be configured to communicate with its Slaves. To configure the Master System you will generally need a software package available from the manufacturer of the Master System. For instance, if the Master System is based on the Siemens Simatic S5, you will use the COM Profibus package, if the Master System is based on the Siemens Simatic S7, you will find the network configuration option already included in the Step 7 software package.
Before the Master configuration software can recognize the UniOP MMI as a Slave, it must be included in the catalog of devices. For this purpose it is available a device description file for UniOP in the standard GSD format. The device description file is EX9649AX.GSD. It must be installed following the instructions of the Master configuration software you are using.

One of the fundamental steps of the configuration of a slave station in a Profibus DP system is the mapping of the slave's I/O buffers in the memory of the Master.
UniOP supports Input / Output buffer sizes of 8, 16 or 32 bytes. UniOP expects that both the Input and the Output areas are configured to the same size, i.e. both 8 bytes, both 16 bytes or both 32 bytes. UniOP will automatically detect the buffer size used by the Master.
The feature generally referred to as Response Monitoring should always be disabled in the Master for UniOP slaves.
You may find in Figure 1 an example of the I/O mapping of a slave in a Simatic S5 system using the Siemens COM Profibus configuration software.



Configure: UniOP #3 <>

	ID	Comment	I Addr.	O Addr.
0	055		P000	P000
1	055		P008	P008
2	055		P016	P016
3	055		P024	P024
4				
5				
6				
7				
8				
9				
10				
11				
12				

Figure 1 – I/O mapping dialog box in Siemens COM Profibus

3. Programming the Master to Process the UniOP Input/Output Data

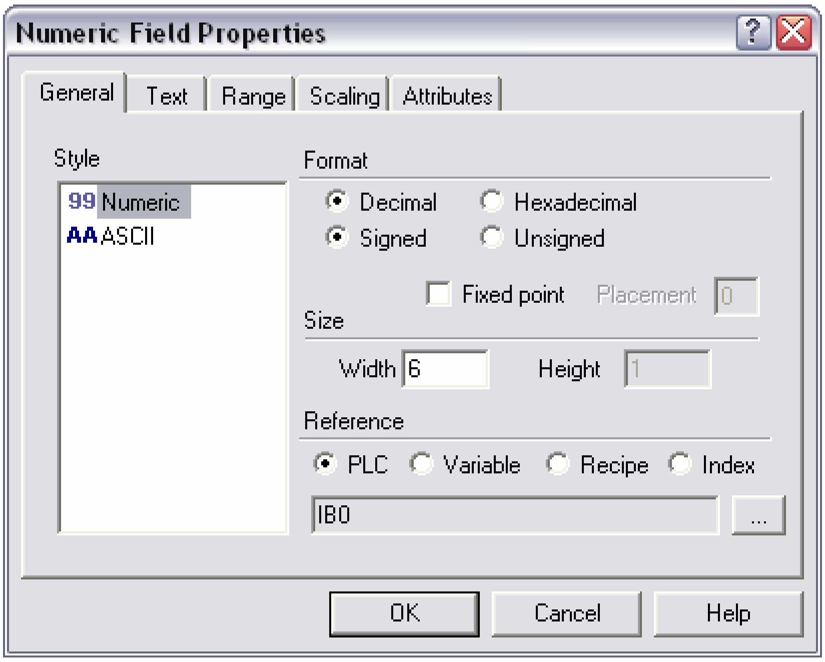
With the Profibus DP driver, UniOP simply exchanges Input and Output data with the Master. It is up to the Master to make sense of this data.

For example, in a simple application of the Profibus DP driver, UniOP could be programmed to set bits in the Input buffer to 1 when keys are pressed on the panel. Using the Macro Editor in the Designer package the F1 key could be programmed such that when pressed it sets bit 1 in the 1st Input buffer byte to 1. Similarly the F2 key could be programmed such that when pressed it sets bit 2 in the 1st Input buffer byte to 1. In this way the PLC program in the Master PLC could monitor the Input data from the UniOP to check the state of the F1 and F2 keys. If the PLC program found that the F1 key was pressed it could perform some action, such as starting a Timer, switching on a motor and so on. If the PLC program found that the F2 key was pressed it could perform some other action. To provide feedback to the UniOP that the key press was recognized, the Master PLC could echo back the data read from the UniOP's Input data to the Output data sent to the UniOP. In this way, for example, the UniOP could be programmed via the RDA Setup dialog box to turn on the LED above the F1 key when bit 1 in the 1st byte of the Output data is set to 1.

This is just a very simple example. Using the features of the Designer package, such as the Macro Editor, the RDA Setup, the Interlock Mailbox and Data Transfer together with your custom PLC program you have the tools to create powerful Profibus DP applications with a very tight interaction between the control program and the MMI.

4. The Define Field Dialog Box for Profibus DP

The Define Field Dialog Box for Profibus DP is shown below:



Style
99 Numeric
AA ASCII

Format
● Decimal ○ Hexadecimal
● Signed ○ Unsigned
□ Fixed point Placement 0

Size
Width 6 Height 1

Reference
● PLC ○ Variable ○ Recipe ○ Index
IB0 ...

Figure 2 – Numeric field properties

1) Designer allows you to access the UniOP “Output Buffer”, the area containing data sent from the PLC, as well as the UniOP “Input Buffer”, the area containing data to be sent to the PLC. The data in the UniOP Output Buffer is read only, while the data in the UniOP Input Buffer is read write. The Address Offset range (in bytes) for these 2 types is from 0 - 31. It should be borne in mind, however, that that Input / Output buffer range configured in the PLC for the panel can be either in the range 0 - 7, 0 - 15 or 0 - 31.

2) In addition to the Input Buffer and the Output Buffer Designer also allows you to access the “Internal Work Buffer” data type. This buffer is purely an internal buffer in the UniOP panel. The UniOP panel sets aside 256 bytes for this buffer. The data in this buffer is neither read from nor written to the PLC. It is purely a work area. It is useful mostly for the RDA and for the Data Transfer. The Data Transfer feature of Designer allows you configure the transfer of data from one area to another either cyclically or based on some trigger. The “Internal Work Buffer” can be used as the source area for a Data Transfer operation and the “Input Buffer” can be used as the target area.

3) The “Input Buffer”, “Output Buffer” and “Internal Work Buffer” data types support the following data formats:

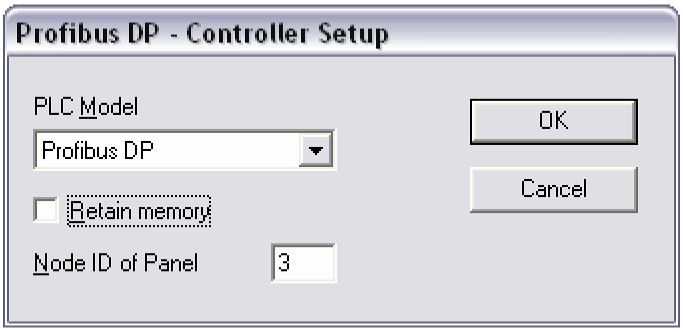
Data format	**Signed**	**Unsigned**
BIT		
BYTE (Bin)	Yes	Yes
BYTE (BCD)	Yes	Yes
WORD (Bin)	Yes	Yes
WORD (BCD)	Yes	Yes
DWORD (Bin)	Yes	Yes
DWORD (BCD)	Yes	Yes
Float	Yes	
WORD INV (Bin)	Yes	Yes
WORD INV (BCD)	Yes	Yes
DWORD INV (Bin)	Yes	Yes
DWORD INV(BCD)	Yes	Yes
Float INV	Yes	

The Word, DWord and IEEE Float formats all use a non-inverted byte order where the low byte comes first and the high byte comes last. All INV formats use an inverted byte order where the high byte comes first and the low byte comes last.

The Float format uses the standard IEEE 754 format where the highest bit is the sign, the next highest 8 bits are the exponent and the remaining 23 bits are the mantissa. The signed Byte, Word and DWord formats use a 2’s complement format.

4. The other supported data type is called “I/O Buffer Length”. Again, this is purely an internal variable. It is a read only variable that allows the currently configured length of the Input / Output buffers in the UniOP to be displayed on the panel. The only supported data format for this data type is Byte.
5. “Node Number Override” is data type which gives to the user a control of the ‘Node ID of the Panel’ at run-time! For more details refer to the chapter 5 ‘Configuring UniOP for use with Profibus DP’.

5. Configuring UniOP for Use with Profibus DP



PLC Model: Profibus DP

□ Retain memory

Node ID of Panel: 3

Figure 3 - Controller Setup

1) To use the Profibus DP protocol with the Designer, follow this procedure: Select the option ‘Project/Change Communication Driver’ and choose ‘Profibus DP’.

2) A UniOP panel in a Profibus DP network is always a Slave that exchanges data with a single Master PLC. Every UniOP in the Profibus DP network must be assigned its own unique Node address. You do this by entering a number in the ‘Node ID of Panel’ edit box in the Controller Setup dialog box.
However, there is a possibility to change the Node ID of the Panel at run time (Refer to the chapter “4. The Define Field Dialog box for Profibus DP”, concerning data type ‘Node Number Override’).
In order to change the UniOP node number it is required a data entry on the ‘Node Number Override’ system variable, and the change will take effect immediatelly at run time. This possibility is introduced starting from driver version 5.10 – 4.09. Earlier version did require a power cycle for the changes to take effect. To restore the use of the original ID programmed in the Designer project, the ‘Node Number Override’ variable should assume value 0 (zero); and UniOP restores immediately the ID number saved in the project.
Node Override information are stored in flash memory.

3) The “Retain memory” check box allows saving in the panel battery backup memory the Input Buffer contents; when enabled the contents of the buffer from panel to DP master is retained between power cycles. This feature is only supported by HMI panels with battery.

4) You do not need to worry about specifying the communication parameters with Designer. UniOP will automatically detect the Baud rate used by the Profibus DP Master.

5) You can attach more than one UniOP to a single Master.

6. The Real Time Clock information in the Reserved Data Area (RDA) is coded in Binary

7. The page number displayed and the page number requested in the RDA are coded in binary.

8. The RDA elements can be positioned within the Input Buffer, Output Buffer and Internal Work Buffer areas.

6. Summary

To get your UniOP up and running as a Profibus DP slave here is a summary of what you need to do:-

1. Install the EX9649AX.GSD file in the Master configuration software package.
2. Configure your Master specifying the UniOP panels as Slaves using the UNIOP MMI type.
3. Create the project files for the UniOP panels. Make sure that the ‘Node ID of Panel’ specified in the Designer Controller Setup dialog matches the node number used for the UniOP panel in the Master configuration.
4. Program the Master to make sense of the UniOP I/O data.
5. Put the Master into RUN mode.

7. Troubleshooting

UniOP doesn’t communicate, COM LED blinks and the Communication Status shows COM ERR 05:

1. Master is in STOP or is not turned on?
2. Cable is not connected or is broken?
3. UniOP has been programmed via the Designer Controller Setup dialog box to have a node number that is the same as another node in the network?

UniOP doesn’t communicate, COM LED blinks and the Communication Status shows COM ERR 04

1. UniOP has been programmed by Designer to access data from the Input or Output buffer beyond the configured range of the Input / Output areas?

Master System fails to go into RUN mode

1. The Master has been configured to have Response Monitoring enabled for the UniOP panels?
2. UniOP has been programmed via the Designer Controller Setup dialog box to have a node number that is the same as another node in the network?

Appendix A. Communication status codes

Current communication status is displayed in the System Menu of the UniOP. Beside the string describing current state of the communication, there is an additional error code representing the last (which may be not the current one) error encountered. The codes are following:

00	No error	
04	Input / Output buffer overflow	Indicates that UniOP attempted to read Input data or to write Output data beyond the range of the currently configured buffers. For example, if the Input and Output buffers are configured to be 8 bytes long then attempting to access the 10th Input byte will result in this error. The solution to this error is to modify the UniOP pages with the Designer package to not access Input or Output data outside the configured range. Alternatively the Master PLC could be reprogrammed to assign a larger Input / Output areas to the UniOP.
05	Low level protocol error	Indicates a Time Out in Data Packet or Status Request. The communication line has been broken or the PLC has had a power fail.
07	Internal failure	This should never happen. It indicates a hardware problem or software low level problem.
08	Communication Time Out	Indicates that the communication line has been broken or the PLC has had a power fail.
09	Failed to enter data exchange	Indicates that the UniOP failed to enter the data exchange state. This happen turning on the UniOP with PLC powered off. May also indicate that there are two or more slaves with the same node number.

Appendix B. Technical Data and Connection Information

The main technical information on the UniOP Profibus DP Slave interface using TCM08 is shown in the table below:

Baud rate	9.6 Kb to 12 Mb
Buffer size	8/16/32 bytes
Slave address	Software configurable
Optical insulation	Yes
Profibus Connector	Standard, 9 pin female sub-D type

The Aux Port is used for the Profibus DP communication. When the TCM08 communication module is mounted, the Aux Port becomes a standard Profibus connector. A simple point to point connection can be performed with the cable CA128.
In all other cases the usage of appropriate bus connectors such as the Siemens 6ES7972-0BA20-0XA0 and 6ES7972-0BB20- 0XA0 is recommended.

Appendix C. Requirements and Compatibility

This version of Profibus DP is included in the Designer DLL file D32UPLC97.DLL.

A communication module of type TCM08 is required. This driver will not work with other communication modules such as the TCM01.
The UniOP panel must have hardware type –0045; firmware version number 4.10 or higher is required to support operation with the TCM08 modules.