Connecting UniOP to K-M Profibus DP

This Technical Note contains all the information required to connect the UniOP panels to a Profibus DP system with a K-M master module PS416-NET-440 and to take advantage from the advanced data access options supported by the UniOP Profibus DP communication driver.

Important: *This Technical Note applies only to the Profibus DP communication driver identified by the name "Profibus DP K-M" associated to the Designer file UPLC111.DLL. To run this communication driver you must have an operator panel of hardware type – 0045 and a communication module type TCM08.*
The operation of the PS416-NET-440 Profibus DP master module requires the Sucosoft S40 Version 1.40 (or later) and the PS416 operating system OS40 version 1.14 (or later).

Contents

1. Introduction

The UniOP panel can be connected to the PS416 PLC via the Profibus DP network by using the K-M Profibus network module PS416-NET-440. 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.

UniOP is always a Slave in a Profibus DP network and it is only able to exchange data with a single Master PLC. To enable UniOP to work, a set of special instructions must be added to the PLC program in the Master PLC. These instructions are needed to process the requests from UniOP.

There are 3 main steps that you need to follow to make UniOP work with Profibus DP:

1) Configure the Profibus DP network with the SyCon software from Hilscher and include UniOP as a slave station.
2) Add the special function calls for UniOP to the Master PLC program.
3) Configure UniOP with the Designer package.

These steps are described in greater detail in the following sections.

2. Configuring UniOP as a Slave Station with SyCon Software by Hilscher

The Master PLC Profibus Network module PS416-NET-440 must be configured to communicate with its Profibus DP slaves. You can do this with the SyCon programming package. This package configures the Profibus DP Network attached to the Master PLC so that it exchanges data with the specified Slaves. With this package you can select different types of Slaves such as MMI, Remote Inputs/Outputs, etc.

2.1 Adding the UniOP GSD File to your System

A Profibus DP Slave type file for UniOP is available for the Profibus DP configuration. The filename is EX9649AX.GSD; this file contains the description of the UniOP panels as Profibus DP Slaves.

To include the file in the system, you have to copy the file to the directory:

\SYCON\FIELDBUS\PROFIBUS\GSD\

This will enable SyCon software to recognise UniOP as an element of the class “MMI”.

2.2 Network Configuration

The basic steps of the Profibus DP configuration are described below.

a) Create a new SyCon project or open an existing project.

b) Configure the Profibus network system using components from the list of the available devices beginning from Master (menu: Insert / Master).

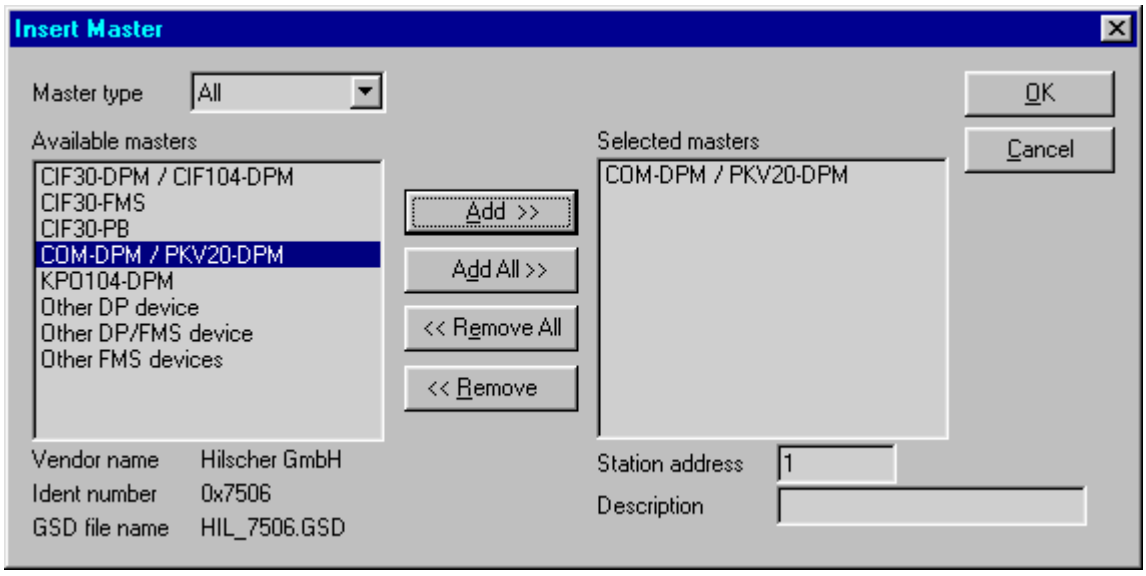

Figure 1 – Insert Master

Figure 1 shows the dialog box for the selection of the DP Master. Select **COM-DPM / PKV20-DPM** from the list.

c) Configure the Profibus DP Master parameters (menu: Settings \ DP Master Settings). Figure 2 shows the DP Master setting dialog box:

- Select the "Controlled release of the communication by the application program" as "Start-up behaviour after system initialisation".
- Enter a value greater than 200 ms for the parameter "Watchdog Time".
- Select "2 KB dual port memory as "Hardware Parameter".
  The "Parameter to process data interface" are determined by the PS416 operating system.

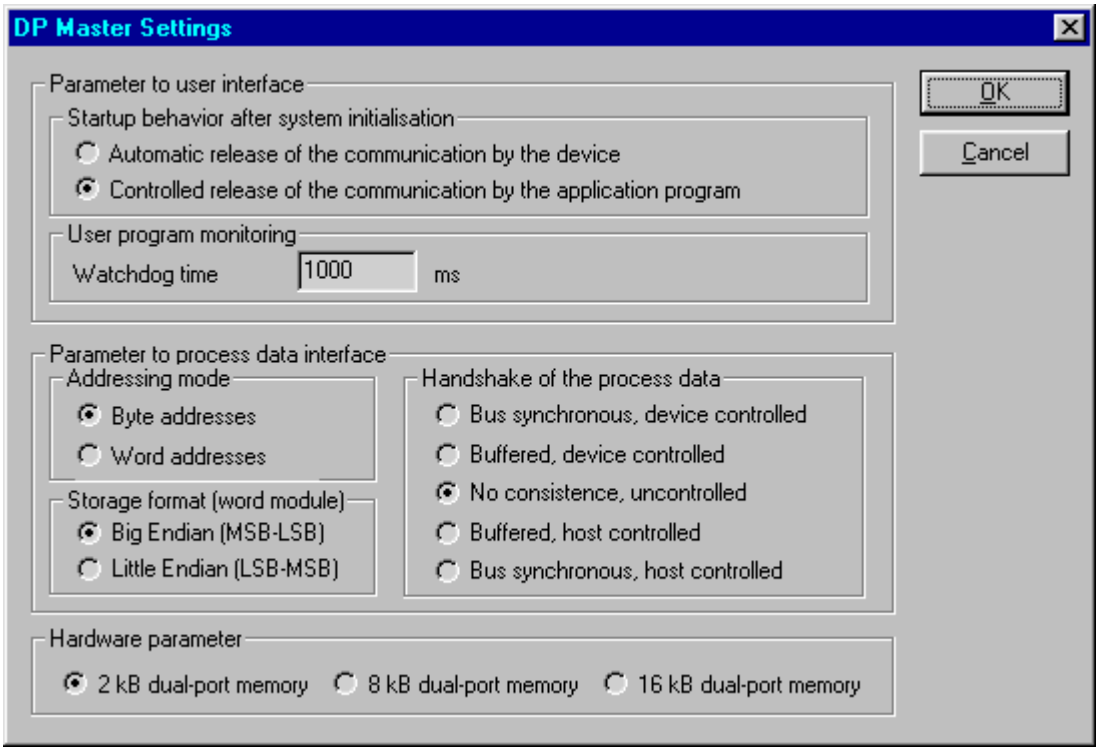

Figure 2 – DP Master Setting

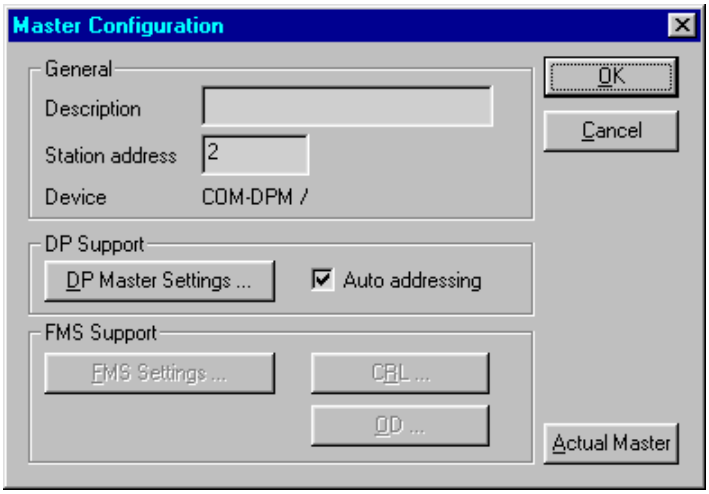

Figure 3 – Master Configuration (right mouse click over Master / Master Configuration)

Figure 3 shows the dialog box with additional Master configuration parameters such as the choice for automatic or manual addressing mode. The flag in this check box informs SyCon to automatically set address I/O for all the slave devices attached to the current Master. If you do not have specific requirements for you system, check 'Auto addressing' mode.

d) Configure the bus parameters (menu: Settings / Bus Parameters).

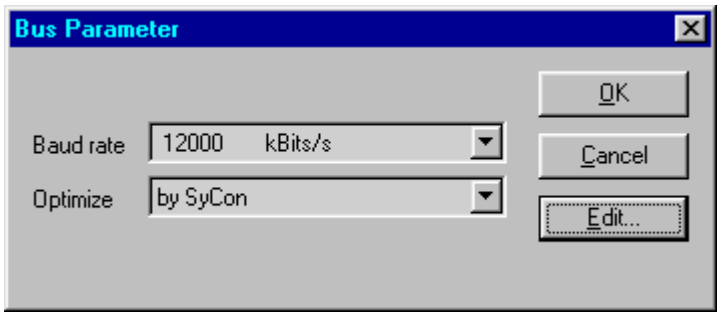

Figure 4 – Bus Parameters

Figure 4 shows the dialog box for the configuration of the bus data rate. Critical bus parameters can be changed clicking the Edit button. Default values of these parameters are correct for most applications.

e) Insert the Profibus DP Slave stations for the desired number of UniOP (menu: Insert / Slave). Figure 5 shows the dialog box that the SyCon software displays when you are adding one UniOP as slave in you Profibus network. In this dialog box you have to assign the Profibus Node Number of the current slave.

Note: *The version V1.14 of OS40 in the PS416 supports up to 24 slave units.*

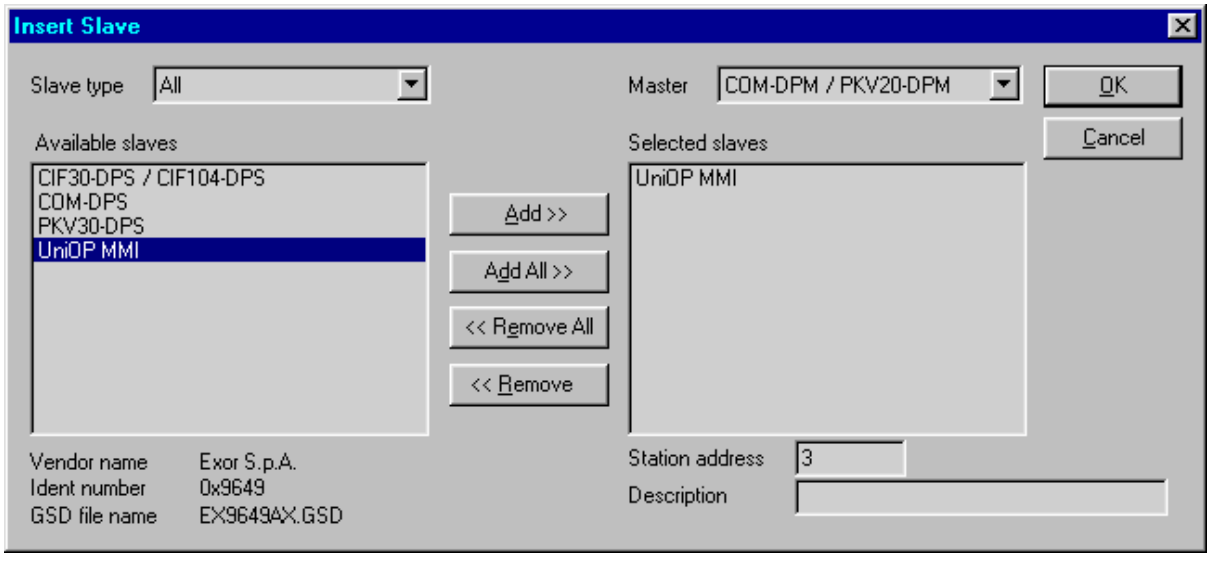

Figure 5 – Insert Slave

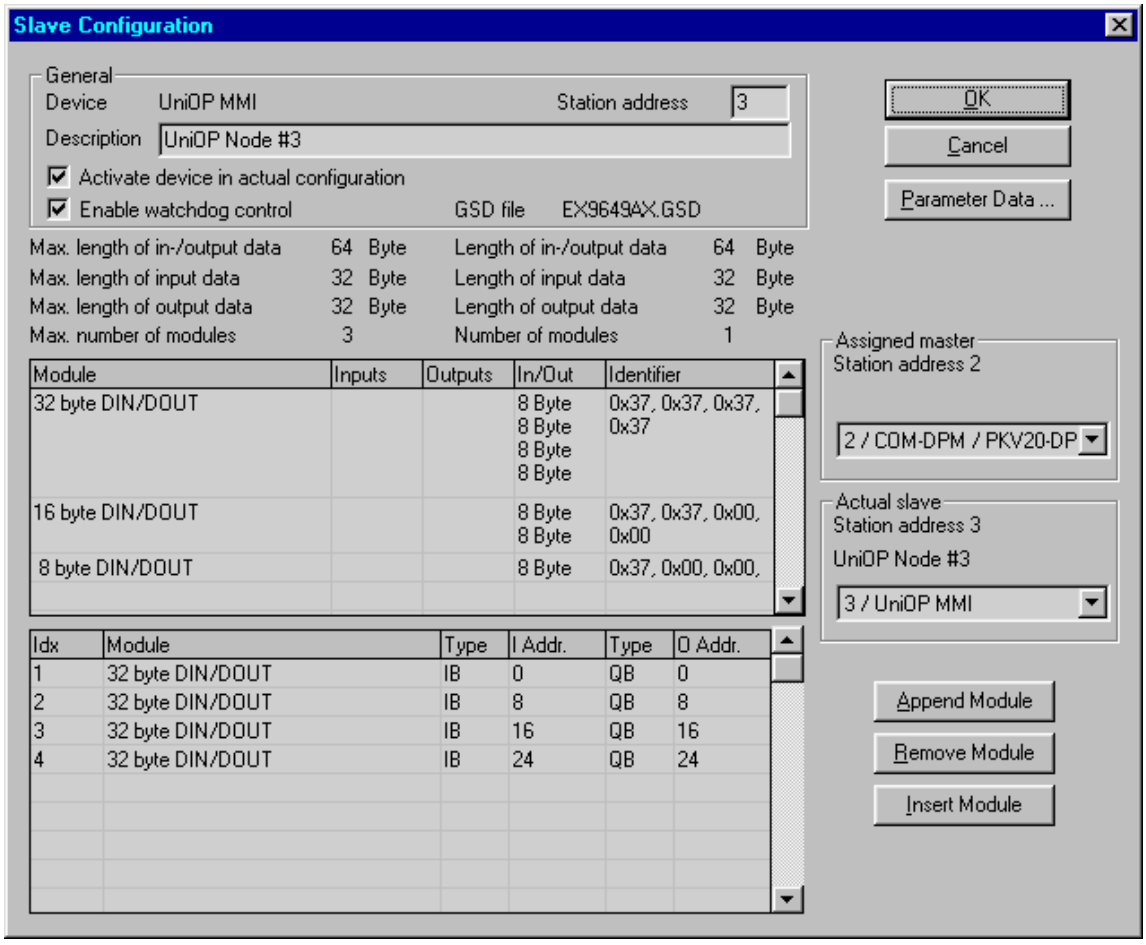

Figure 6 – Slave configuration

Figure 6 shows the dialog box for setting up the slave configuration in the network. The example shows one UniOP MMI configured as Slave Node number 3 with an I/O buffer size of 32 bytes. UniOP panels can work with a Profibus DP image size of 16 bytes or 32 bytes. Using 32 bytes will result, generally, in a higher communication performance at the expense of increased memory usage in the I/O image area.
When Auto addressing mode is enabled (see Figure 3), SyCon will assign the address to the slave in the I/O map and will fill the table located in the lower left part of the dialog box.

f) Download to the Profibus DP master station the complete system configuration.
Once the Profibus system configuration has been completed, the SyCon software produces a map of the network as shown in Figure 7. The example is contains only one MMI as slave.
If Auto addressing mode is selected, no additional operations are required to complete the I/O memory allocation.

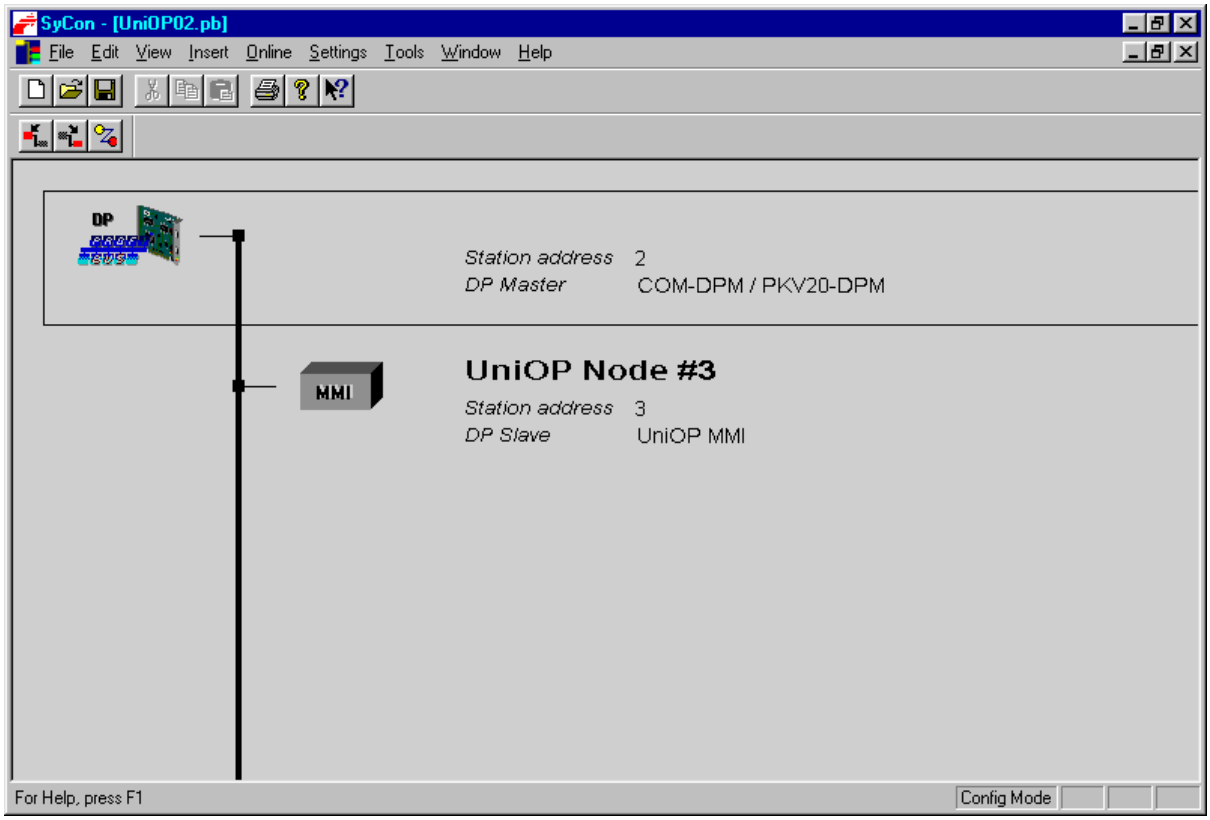

Figure 7 – Profibus network topology in SyCon configurator

At this point the configuration can be downloaded directly to the Profibus DP master via the CFG-Interface of the PS416-NET-440.
The download process can be performed in two ways: using a special communication board installed in your computer or using one of the COM ports of the computer. The COM port must be assigned to the SyCon software as shown in Figure 8; COM 2 serial port is used in the example.

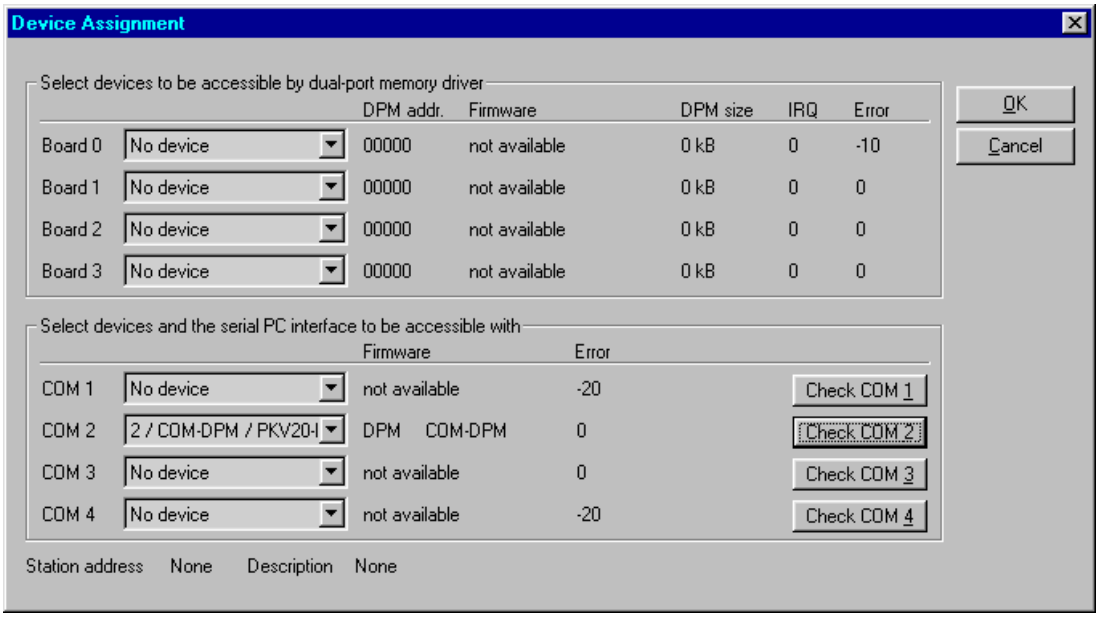

Figure 8 – Device assignment

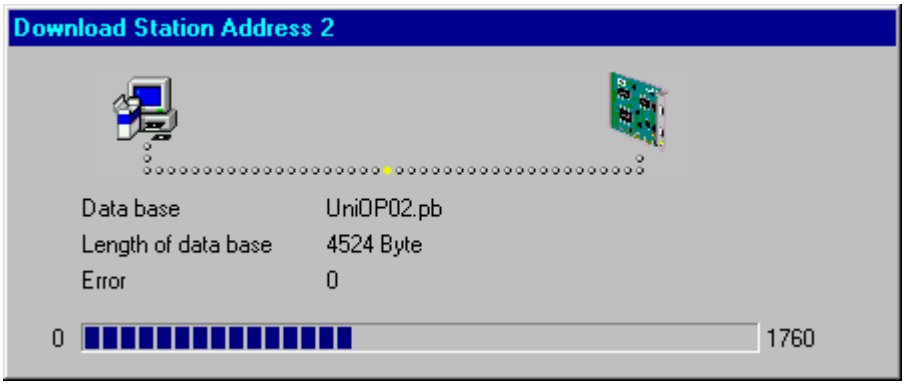

Figure 9 – Download process

During the download process (see Figure 9) the Ready LED in the PS416-NET-440 card blinks.

Note: *If the run LED is on after the end of the download, you have configured the option "Start-up behaviour after system initialisation" as "Automatic release of the communication by the device" (see dialog box in Figure 2). In this case you should modify this option and download the configuration again.*

---

3. Communication Server Program in the PLC

In order for the UniOP to access data resident in the Master PLC, some support from the PLC program is required. It is accomplished by adding to the user's PLC program the appropriate data declaration and function calls. The necessary function block is available in the library provided by the S40 PLC programming package.

The core functionality is provided by one special function block that must be called cyclically in the user's program.
In the Sucosoft S 40 for the PS 416 the function block is MI4netK.
The function block has been originally defined to handle Suconet K communication but it can be used also to support data exchange over the Profibus DP Network.

The value of the input data available from Profibus DP is read out from the DPR at the beginning of the program cycle; the value of the output data is written to the DPR at the end of the cycle.

To process the data for one of the slave UniOP's the manufacturer function block has to be called after the input data has been copied to the input array of the function block. After processing the data the function block will return the response in its output array that will have to be copied to the output buffer of the slave being processed.
The MI4netK function should be called once in each PLC cycle for each of the UniOP configured as slaves. Each time the proper data has to be copied from the input buffer and the response copied to the output buffer.

3.1 Software Configuration

The PS416-NET-440 Profibus module must be configured in the S40 Device Configurator:
- Select the module as "PS416-NET-440" in the option field cards.
- Enter the number of the connected Profibus DP line into the field 'Index'.
- Figure 10 shows the dialog box for the hardware configuration of the PLC rack. The example includes also some digital I/O.

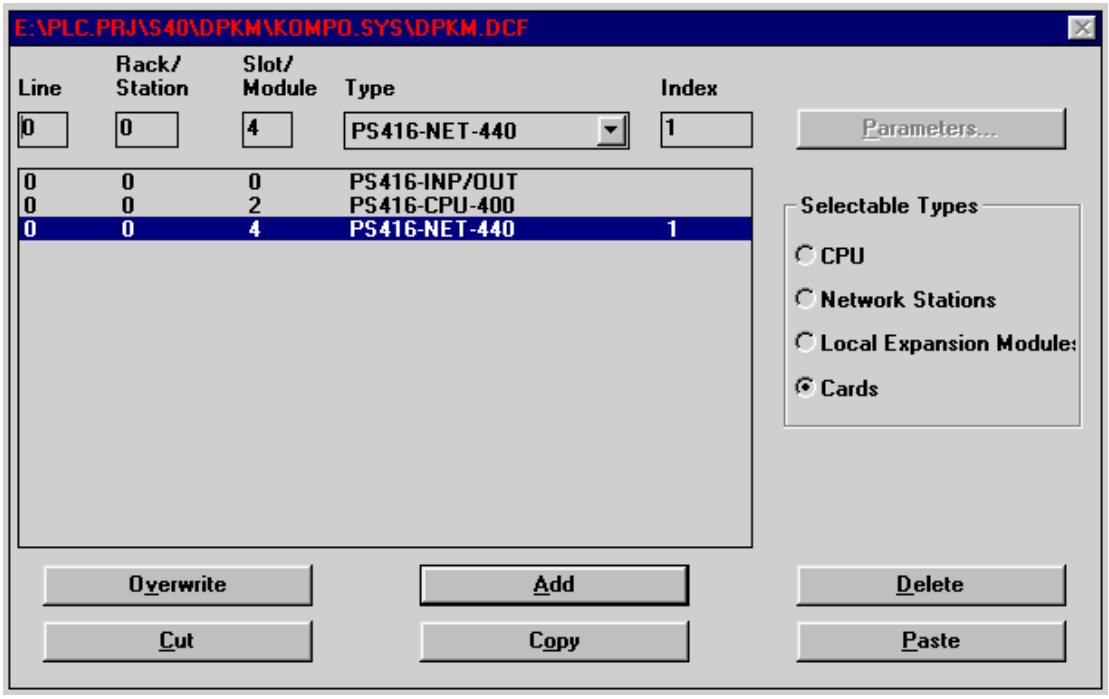

Figure 10 – S40 Device Configurator

Note: *Only line numbers in the range 1 to 9 are allowed. The line number must be clearly related to a line within a configuration (Profibus DP or Suconet-K).*

The PS416 operating system OS40 version V1.14 supports up to 3 NET-440 cards.

Note: *When 3 PS416-NET-440 cards are configured, the plug-in station 20 is blocked for other cards. In case the S40 configuration does not follow this rule, you receive the error report 804B within the application program.*

The PS416-NET-440 is not supported in the expansion rack.

3.2 I/O Access

The Input and Output data buffers of the Profibus DP slave are only accessible as Input or Output in the Master PLC memory. This means that the data buffers of each Profibus Slave device can be reached using an I/O access instruction.
The assignment to the necessary program variable is performed with the AT operator in the declaration part of the application program.
The correct syntax is described below:

```
<Line_Number>.<Participant_Number>.<Mudule_Number>.<Byte>.<Bit>
```

The **<Line_Number>** is the number that is indicated in the S40 Device Configurator for the PS416-NET-440 in the field “Index”.

As **<Participant_Number>** number you must enter the participant address (slave address) which is defined in the DP Configurator.

The **<Module_Number>** is determined by the number that is assigned with the SyCon Software in the column “Idx” of the dialog box “Slave Configuration” (see Figure 6).

When the modules of the participant are serially numbered, it is an I/O participant we are dealing with, the module number is 0.
When the number 1 has been entered in the column "Idx" for all modules, we are dealing with a modular slave; that means that the I/O channels are mounted on modules. In this case the module number results from the position of the module in the slave, starting from 0 and according to the order which is given in the dialog box "Slave Configuration".

3.3 PLC Program example

In this section we provide a simple example of PLC program supporting the simple Profibus DP network defined in the first part of this document.

The complete example program is also available on diskette.

The program fragment below shows the data declaration part.

```
VAR_GLOBAL
      RDB3_00 AT %IB1.3.0.0  : BYTE ;
      RDB3_01 AT %IB1.3.0.1  : BYTE ;
      RDB3_02 AT %IB1.3.0.2  : BYTE ;
      RDB3_03 AT %IB1.3.0.3  : BYTE ;
      RDB3_04 AT %IB1.3.0.4  : BYTE ;
      RDB3_05 AT %IB1.3.0.5  : BYTE ;
      RDB3_06 AT %IB1.3.0.6  : BYTE ;
      RDB3_07 AT %IB1.3.0.7  : BYTE ;
      RDB3_08 AT %IB1.3.0.8  : BYTE ;
      RDB3_09 AT %IB1.3.0.9  : BYTE ;
      RDB3_10 AT %IB1.3.0.10 : BYTE ;
      RDB3_11 AT %IB1.3.0.11 : BYTE ;
      RDB3_12 AT %IB1.3.0.12 : BYTE ;
      RDB3_13 AT %IB1.3.0.13 : BYTE ;
      RDB3_14 AT %IB1.3.0.14 : BYTE ;
      RDB3_15 AT %IB1.3.0.15 : BYTE ;
      RDB3_16 AT %IB1.3.0.16 : BYTE ;
      RDB3_17 AT %IB1.3.0.17 : BYTE ;
      RDB3_18 AT %IB1.3.0.18 : BYTE ;
      RDB3_19 AT %IB1.3.0.19 : BYTE ;
      RDB3_20 AT %IB1.3.0.20 : BYTE ;
      RDB3_21 AT %IB1.3.0.21 : BYTE ;
      RDB3_22 AT %IB1.3.0.22 : BYTE ;
      RDB3_23 AT %IB1.3.0.23 : BYTE ;
      RDB3_24 AT %IB1.3.0.24 : BYTE ;
      RDB3_25 AT %IB1.3.0.25 : BYTE ;
      RDB3_26 AT %IB1.3.0.26 : BYTE ;
      RDB3_27 AT %IB1.3.0.27 : BYTE ;
      RDB3_28 AT %IB1.3.0.28 : BYTE ;
      RDB3_29 AT %IB1.3.0.29 : BYTE ;
      RDB3_30 AT %IB1.3.0.30 : BYTE ;
      RDB3_31 AT %IB1.3.0.31 : BYTE ;
```

```
      SDB3_00 AT %QB1.3.0.0  : BYTE ;
      SDB3_01 AT %QB1.3.0.1  : BYTE ;
      SDB3_02 AT %QB1.3.0.2  : BYTE ;
      SDB3_03 AT %QB1.3.0.3  : BYTE ;
      SDB3_04 AT %QB1.3.0.4  : BYTE ;
      SDB3_05 AT %QB1.3.0.5  : BYTE ;
      SDB3_06 AT %QB1.3.0.6  : BYTE ;
      SDB3_07 AT %QB1.3.0.7  : BYTE ;
      SDB3_08 AT %QB1.3.0.8  : BYTE ;
      SDB3_09 AT %QB1.3.0.9  : BYTE ;
      SDB3_10 AT %QB1.3.0.10 : BYTE ;
      SDB3_11 AT %QB1.3.0.11 : BYTE ;
      SDB3_12 AT %QB1.3.0.12 : BYTE ;
      SDB3_13 AT %QB1.3.0.13 : BYTE ;
      SDB3_14 AT %QB1.3.0.14 : BYTE ;
      SDB3_15 AT %QB1.3.0.15 : BYTE ;
      SDB3_16 AT %QB1.3.0.16 : BYTE ;
      SDB3_17 AT %QB1.3.0.17 : BYTE ;
      SDB3_18 AT %QB1.3.0.18 : BYTE ;
      SDB3_19 AT %QB1.3.0.19 : BYTE ;
      SDB3_20 AT %QB1.3.0.20 : BYTE ;
      SDB3_21 AT %QB1.3.0.21 : BYTE ;
      SDB3_22 AT %QB1.3.0.22 : BYTE ;
      SDB3_23 AT %QB1.3.0.23 : BYTE ;
      SDB3_24 AT %QB1.3.0.24 : BYTE ;
      SDB3_25 AT %QB1.3.0.25 : BYTE ;
      SDB3_26 AT %QB1.3.0.26 : BYTE ;
      SDB3_27 AT %QB1.3.0.27 : BYTE ;
      SDB3_28 AT %QB1.3.0.28 : BYTE ;
      SDB3_29 AT %QB1.3.0.29 : BYTE ;
      SDB3_30 AT %QB1.3.0.30 : BYTE ;
      SDB3_31 AT %QB1.3.0.31 : BYTE ;
END_VAR

VAR
      Read_Data_Area1 : ARRAY[1..78] OF BYTE ;
      Send_Data_Area1 : ARRAY[1..78] OF BYTE ;
      FB_Diagnose1 : BYTE ;
      MI4_Com1: MI4netK ;

END_VAR
```

The communication with the slave number 3 using the standard function blocks is shown below.

```
LD    RDB3_00
ST    Read_Data_Area1[1]
LD    RDB3_01
ST    Read_Data_Area1[2]
LD    RDB3_02
ST    Read_Data_Area1[3]
LD    RDB3_03
ST    Read_Data_Area1[4]
LD    RDB3_04
ST    Read_Data_Area1[5]
LD    RDB3_05
ST    Read_Data_Area1[6]
LD    RDB3_06
```

```
ST    Read_Data_Area1[7]
LD    RDB3_07
ST    Read_Data_Area1[8]
LD    RDB3_08
ST    Read_Data_Area1[9]
LD    RDB3_09
ST    Read_Data_Area1[10]
LD    RDB3_10
ST    Read_Data_Area1[11]
LD    RDB3_11
ST    Read_Data_Area1[12]
LD    RDB3_12
ST    Read_Data_Area1[13]
LD    RDB3_13
ST    Read_Data_Area1[14]
LD    RDB3_14
ST    Read_Data_Area1[15]
LD    RDB3_15
ST    Read_Data_Area1[16]
LD    RDB3_16
ST    Read_Data_Area1[17]
LD    RDB3_17
ST    Read_Data_Area1[18]
LD    RDB3_18
ST    Read_Data_Area1[19]
LD    RDB3_19
ST    Read_Data_Area1[20]
LD    RDB3_20
ST    Read_Data_Area1[21]
LD    RDB3_21
ST    Read_Data_Area1[22]
LD    RDB3_22
ST    Read_Data_Area1[23]
LD    RDB3_23
ST    Read_Data_Area1[24]
LD    RDB3_24
ST    Read_Data_Area1[25]
LD    RDB3_25
ST    Read_Data_Area1[26]
LD    RDB3_26
ST    Read_Data_Area1[27]
LD    RDB3_27
ST    Read_Data_Area1[28]
LD    RDB3_28
ST    Read_Data_Area1[29]
LD    RDB3_29
ST    Read_Data_Area1[30]
LD    RDB3_30
ST    Read_Data_Area1[31]
LD    RDB3_31
ST    Read_Data_Area1[32]

CAL         MI4_Com1(
                  RDB_Area := Read_Data_Area1,
                  SDB_Area := Send_Data_Area1
                  |
                  FB_Diagnose1 :=Status
                  )
```

```
LD    Send_Data_Area1[1]
ST    SDB3_00
LD    Send_Data_Area1[2]
ST    SDB3_01
LD    Send_Data_Area1[3]
ST    SDB3_02
LD    Send_Data_Area1[4]
ST    SDB3_03
LD    Send_Data_Area1[5]
ST    SDB3_04
LD    Send_Data_Area1[6]
ST    SDB3_05
LD    Send_Data_Area1[7]
ST    SDB3_06
LD    Send_Data_Area1[8]
ST    SDB3_07
LD    Send_Data_Area1[9]
ST    SDB3_08
LD    Send_Data_Area1[10]
ST    SDB3_09
LD    Send_Data_Area1[11]
ST    SDB3_10
LD    Send_DAta_Area1[12]
ST    SDB3_11
LD    Send_Data_Area1[13]
ST    SDB3_12
LD    Send_Data_Area1[14]
ST    SDB3_13
LD    Send_Data_Area1[15]
ST    SDB3_14
LD    Send_Data_Area1[16]
ST    SDB3_15
LD    Send_Data_Area1[17]
ST    SDB3_16
LD    Send_Data_Area1[18]
ST    SDB3_17
LD    Send_Data_Area1[19]
ST    SDB3_18
LD    Send_Data_Area1[20]
ST    SDB3_19
LD    Send_Data_Area1[21]
ST    SDB3_20
LD    Send_Data_Area1[22]
ST    SDB3_21
LD    Send_Data_Area1[23]
ST    SDB3_22
LD    Send_Data_Area1[24]
ST    SDB3_23
LD    Send_Data_Area1[25]
ST    SDB3_24
LD    Send_Data_Area1[26]
ST    SDB3_25
LD    Send_Data_Area1[27]
ST    SDB3_26
LD    Send_Data_Area1[28]
ST    SDB3_27
LD    Send_Data_Area1[29]
ST    SDB3_28
LD    Send_Data_Area1[30]
```

```
ST      SDB3_29
LD      Send_Data_Area1[31]
ST      SDB3_30
LD      Send_Data_Area1[32]
ST      SDB3_31
```

Since the option "Controlled release of the communication by the application program" has been selected in the configurator SyCon (see Figure 2), the communication via Profibus DP is not active after the download of the program: the run LED of the PS416-NET-440 is blinking.

Note: *The PS416 operating system will take over the Profibus DP configuration at the first cold start of the application program. For that reason you must stop and start again the PLC application program after transferring a new Profibus DP configuration.*

The data exchange via Profibus DP will start at the transition from ready to run of the PLC. Then the run LED of the PS416-NET-440 will be steady on, provided that at least one of the DP slaves in the configuration is connected.

The data exchange will be stopped at the transition from run to ready of the PS 416 CPU. When this happens the run LED of the PS416-NET-440 will start blinking.

4. Configuring UniOP with Designer for Profibus DP K-M

To configure an operator panel for use with the Profibus DP K-M protocol with the Designer, follow the procedure described in this chapter.

1) Select the option 'Project/Change Communication Driver' and choose "Profibus DP K-M".

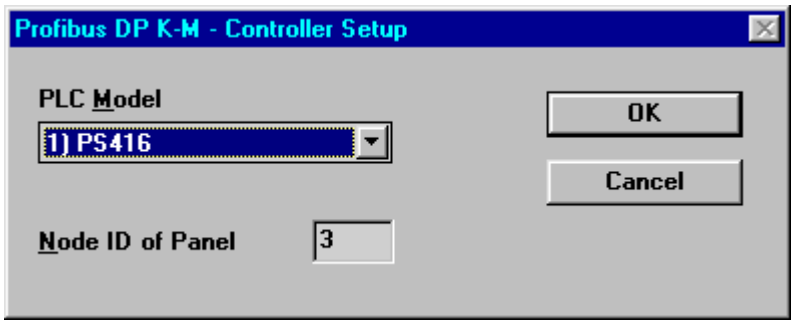

Figure 11 - Controller Setup

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 it's 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.

3) Automatic recognition of the baud rate is supported. Communication speeds up to 12 Mb are supported.

3) The Real Time Clock information in the Reserved Data Area (RDA) is coded in BCD

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

5) The RDA can be freely positioned within the PLC Merker area.

6) The cable to use for the direct connection between the Aux Port of the UniOP panel and the Profibus DP port of the module is CA128.

Note that you can attach more than one UniOP to a single Master PLC.

5. Summary

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

1. Copy the EX9649AX.GSD file to the SyCon GSD subdirectory.
2. Include the NET-440 card as Profibus DP Master in the configuration
3. Include the UniOP panels as Profibus DP slaves in the configuration
4. Download the Profibus DP into the Master card.
5. Include the special function calls to MI4netK in the Master PLC program.
6. Create the project files for the UniOP panels. Make sure that the “Panel Node” specified in the Designer Controller Setup dialog box matches the node number used for the UniOP panel in the SyCon package.
7. Switch the PLC to run mode.

6. Troubleshooting

UniOP doesn't communicate, COM LED blinks and the Communication Status shows COM ERR 05 or COM ERR 09:

1. PLC is in STOP or is not turned on?
2. Cable is not connected or is broken?
3. MI4netK is not called cyclically in the PLC program?
4. Are the data declarations for MI4netK correct?
5. 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?
6. Check if there is another slave station configured with the same address as this one.

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

1. UniOP has been programmed by Designer to access a non-existent data area. Either the RDA or a data item?

Appendix 1 - Communication Error 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:

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 PLC does not accept the request from UniOP. This error occurs if a non-existent data block is accessed.
05	Time-out error to request	This error indicates that the communication line has been broken, the PLC has had a power fail or the PLC is in STOP. This happen also if the master has sent the parameters to UniOP without DP watchdog and the communication cable has been disconnected.
06	Ill formed response	Response data was received from the PLC but the amount of data received was not what was expected.
07	Internal failure	This should never happen. It indicates a hardware problem or software low level problem.
08	DP watchdog timeout	Indicates that the communication line has been broken or the PLC has had a power fail. This happens if the master has sent parameters to UniOP with DP watchdog and the communication cable has been disconnected.
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 or with no cable connected. May also indicate that there are two or more slaves with the same node number.
10	Negative acknowledgement in response	This error occurs when the master PLC accepts the request from UniOP but after processing the request returns a NAK

Appendix 2 - Technical Data and Connection Information

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

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

UniOP is equipped with a standard Profibus connector. A simple point to point connection can be performed with the cable CA128.

Appendix 3 - Requirements and Compatibility

This version of Profibus DP is included in the Designer DLL file UPLC111.DLL. The initial release level is 3.00 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).

A communication module of type TCM08 is required. This driver will not work with any other communication modules.
The type version numbers and status of communication modules mounted in UniOP are displayed in the System Menu.
The UniOP panel must have hardware type –0045 and firmware version number 4.09 or higher to support the TCM08 modules.