ANTAL ELECTRONIC

Field Bus and Communication Technology

Manual

PDP2CL2

Version 3.08

Höfles 4 • 91322 Gräfenberg
Telephone (+49) 09192/9256-0 • Fax (+49) 09192/9256-78

Important Notice

Antal Electronic reserve the right to make any modifications due to progress in technology to this manual, the software and product described therein without prior notice. No part of this documentation may be reproduced in any way (photocopy, micro film, or any other process) or be processed, copied or distributed to third parties by means of electronic systems without express written permission by Antal Electronic.

All information and technical specifications in this manual were compiled with utmost care. However, Antal Electronics can neither give any guarantee that the information in this manual is correct nor accept any liability, legal or otherwise, for consequences which are due to incorrect information.

All software , hardware and brand names in this manual are property of their respective owners.

Version 3.08
April 2000

Contents

1 Operational Safety

! Only qualified personell are allowed to install and operate this equipment. Qualified personell are those trained and certified to install, and operate electronic equipment with respect to the valid electronic safety standards to date.

! The module must not be installed and wired while in a powered condition.

! To guarantee functionality of the module proper means of transportation, storage and installation must be observed.

! Use power supplies certified according to the national electronic safety standards.

! Proper connection of all power supply and data line connections to be observed

! When transferring the module from a cold environment to a warm one condensation may occur. The module has to be perfectly dry before installation and use. Do not install the module near open water or where high humidity is present.

! No user servicable parts inside. Warranty void if module is opened and/or tampered with.

Chapter 2

2 Introduction

The PDP-2-CL2 bus coupler allows you to transform data from a secondary PROFIBUS-DP-ring into a secondary CAN ring and vice versa.
The PPROFIBUS side is designed as a DP slave. The interfaces correspond to EN 50170 and are galvanically insulated by means of DC/DC converters and opto couplers. A C515C micro controller, supported by a SPC ASIC, is responsible for protocol handling tasks. The DP slave supports the entire DP protocol according to EN 50170. The baud rate from 9.6 kBaud to 12 Mbaud is detected automatically.
The maximum amount of I/O date is 320 byte (40 CAN messages at 8 bytes each). Up to 40 CAN messages can be parameterised in total. A 10 byte window in the process image makes it possible to transfer additional CAN messages.
The communication on the CAN side is based upon Layer 2 exclusively. The CAN bus interface corresponds to ISO/DIS 11898 and is galvanically insulated at 1 kV DC. The interface was constructed with a CAN bus driver module 82C250 and the integrated Basic-CAN-Controller of the C515C micro controller. The serial interface (RS232, RS422, RS485 or TTY) is also galvanically insulated at 1 kV DC.

Chapter 3

3 Hardware

3.1 Display elements and connections

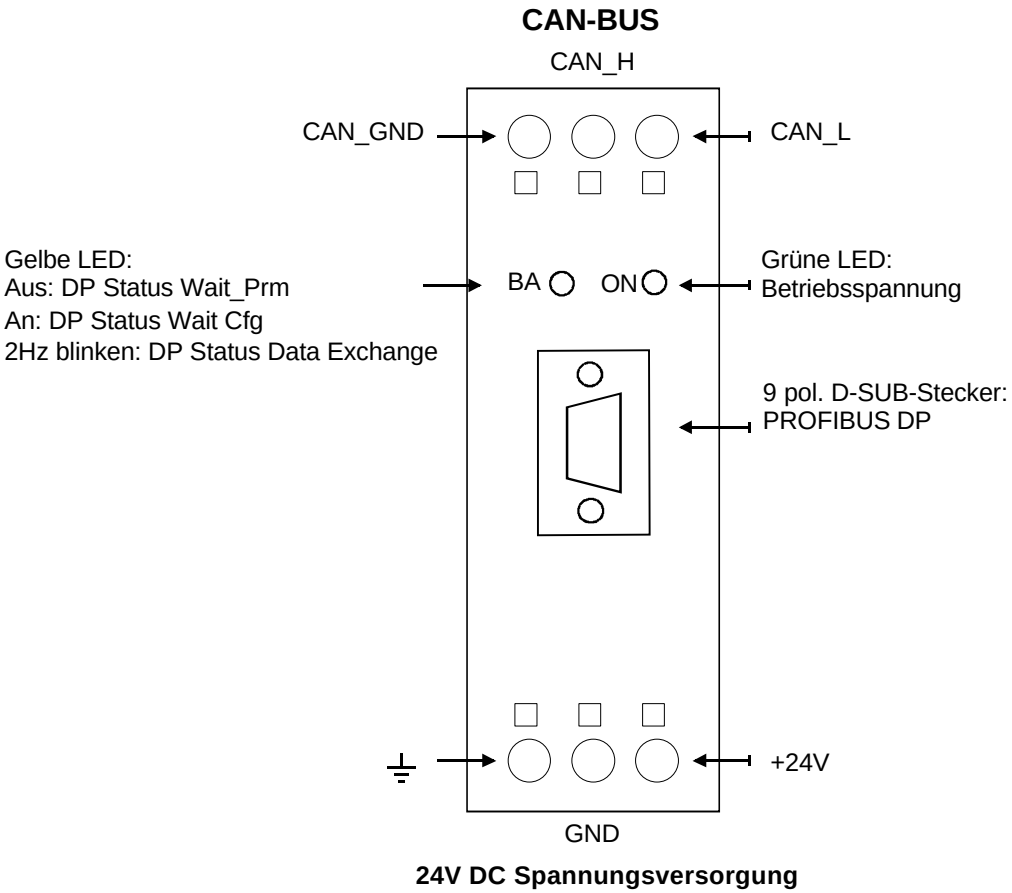


Figure 1 Display Elements PDP-2-CL2

The PDP-2-CL2 uses two LEDs as display elements. The green LED indicates correct power supply, and the yellow LED shows the status of the PROFIBUS module. When both LEDs flash, there is an error (2 Hz flash: error DP initialisation; 05 Hz flash: RAM error). The D-SUB is the connection for the Profibus-DP. The 2 three-pin screw-on clamps supply the voltage and connect to the CAN bus.

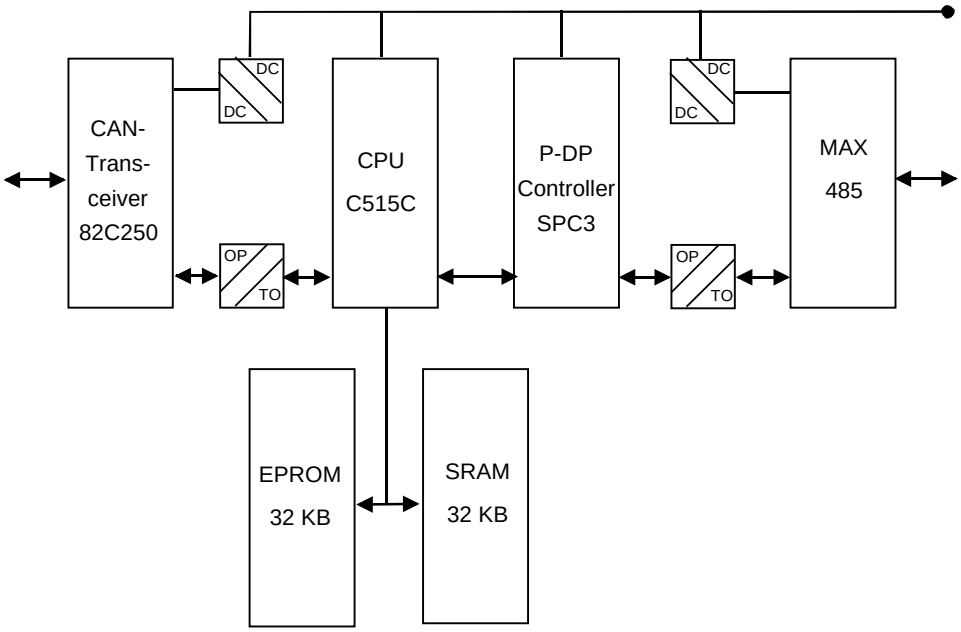


Figure 2: Block diagram PDP2CL2

The CAN bus interface corresponds to ISO/DIS 11898 and is galvanically separated at 1 kV DC.
The CAN bus interface corresponds to ISO/DIS 11898 and is galvanically insulated at 1 kV DC. It was constructed with the CAN bus driver module 82C250 and the integrated Basic-CAN-controller of the C515C micro controller. A SPC3 ASIC by Siemens is in charge of protocol handling. It supports the complete PROFIBUS-DP protocol according to EN 50170. The interface is galvanically insulated via a DC/DC-converter and an opto coupler. Figure 2 shows the block diagram of the PDP2CL2.

3.2 PROFIBUS bus interface

Insulation: 1 kV DC via opto coupler and DC/DC-converter

Transfer rate: 9,6kBit/s ... 12MBit/s

Allocation of the 9 pin D-SUB-plugs

Pin No.	Allocation
1	n.c.
2	n.c.
3	data B (RxD/TXD-P)
4	n.c.
5	DGND (data transfer potential, mass at 5V)
6	VP (voltage supply for terminator resistors)
7	n.c.
8	data A (RxD/TxD-N)
9	n.c.

3.3 DIP-switches

The DIP switch allows you to set the PROFIBUS module address and the CAN baud rate. It can be found on the bottom of the module.

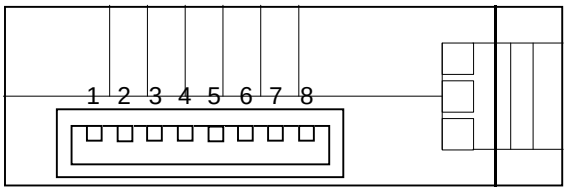


Figure 3 position of DIP switches

Chapter 4

4 Module configuration

4.1 Switch SW1

The DIP switch is used to configure the PROFIBUS module address (node address) and the CAN protocol. Switches 1 to 7 set the node address in the range between 1 to 126. Switch 8 configures the CAN protocol.

Node address:

Module-ID:	1	2	4	8	16	32	64
Switch No.:	1	2	3	4	5	6	7

The settings “0” and “126” are not valid. If the switches are set to these values, the default configuration 1 will be used automatically.

Switch ON means Bit = logical 1
Switch OFF means Bit = logical 0

CAN-Protocol:

Switch S8 determines the relevant CAN protocol. Currently, the CAN layer 2 protocol and the more widely used protocol CANopen defined by the CAN user organisation CiA have been implemented. Further customer-specific protocols can be implemented.

Protocol	**S8**
CAN-Layer-2	off
CANopen	on

Switch ON means Bit = logical 1
Switch OFF means Bit = logical 0

Chapter 5

5 Functional principles

The PDP-2-CL2 bus module allows you to convert data from a PROFIBUS-DP ring into a secondary CAN ring and vice versa. Before data can be exchanged between PROFIBUS and CAN, it is necessary to define the number of CAN messages and the messages themselves during the parameterisation and configuration phase. After successful parameterisation and configuration, data can be transferred.

5.1 Structure of a CAN message

A CAN message consists of a two byte header and 0 to 8 data bytes. The header contains 11 identifier bits, one request-to-receive bit (RTR-bit) and four bit for the data length (L0 ...L3)

CAN-Header

Byte 0:	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
Byte 1:	ID2	ID1	ID0	RTR	L3	L2	L1	L0

Byte 2 - x: Data bytes, depending on data length (max. 8)

5.2 Parameterisation

With the parameterisation message, the master identifies itself and determines the operation mode for the slave. The following CAN parameters are set by means of user-specific parameters (parameterisation demonstrated in an example):

- CAN baud rate
- Number of receiving objects
- Number of sending objects
- All receiving objects (High byte-, Low byte-CAN-Header)
- All sending objects (High byte-, Low byte-CAN-Header)

The sequence of parameters in detail:

0x00: The first parameter byte must always be 0x00; it is only used internally for the Profibus.

CAN-baud rate

The first byte in the parameterisation message configures the baud rate. The following baud rates are supported:

Value (hex)	Baud rate
00	1000 KBit/s
01	500 KBit/s
02	250 KBit/s
03	125 KBit/s
04	100 KBit/s
05	50 KBit/s
06	20 KBit/s
07	10 KBit/s

number of receiving objects

Determines the maximum number of receiving messages.

Number of Sending Objects

Determines the maximum number of sending messages.

all sending objects

- High byte CAN-Header
- Low byte CAN-Header

all receiving objects

- High byte CAN-Header
- Low byte CAN-Header

5.3 configuration

When parameterisation is done, the master has to send a configuration message to the respective slave. The configuration message contains information about the length of input and output data.
The configuration message is composed by the user with the project tool, where it is possible to enter the address range where effective data is stored (see example)

Up to 8 bytes are written in one octet of the configuration message's data unit . It uses as many configuration bytes as are contained in the send/receive buffer plus 4 bytes (handshake in, handshake out, RecObject and TrObject). The objects Handshake_In, Handshake_Out, RecObject and TrObjekt are at the start of the configuration and have to be configured in the correct sequence (see example). If this is not the case, a configuration error will be reported. If the slave detects during check-up that input and output data length do not correspond to each other, it will report a configuration error to the master during the next diagnostic query. In this case, it will not be ready for transfer of effective data.

5.4 Data exchange

After the master has found that were no errors during parameterisation and configuration during the end of the startup phase, data exchange can start. To this end, the PROFIBUS master cyclically sends all data from the parameterised sending identifiers to the PDP2CL2. Sending objects are only transferred to the CAN bus when the corresponding sending bit changes into the handshake area.

On the receiving side, the receiving buffers of the CAN controller are read on a constant basis and entered into the internal buffer of the gateway. Messages are only interpreted only if receiving identifier, message length, and message type are correct.

5.5 The RecObject

ia the RecObject it is possible to receive additional telegrams of a particular group of identifiers.

The CAN telegrams to be received may be defined with an acceptance filter during parameterization.

The acceptance filter will operate on the first byte of the CAN identifier of the CAN telegram (COB-Id).

Of the 11 bit identifier the highest eight (8) bits will pass through the filter and the remaining three (3) bits will be igbnored.

The CAN identifier is as follows:

1st byte:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Highest eight (8) bits of the CAN identifier (Bits 10 .. 3)							

2nd byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Lowest three (3) bits of CAN-Identifiers (Bit 2-0)			RTR-Bit	Data Length Code (DLC)			

Akzeptanz-Filter: CAN-ID Bit Nr.: 10 9 8 7 6 5 4 3
Filter-Bit Nr.: 7 6 5 4 3 2 1 0

The first byte of the acceptance filter defines the acceptance mask and so determines which bits have relevance for the filtratuion.

The second byte of the acceptance filter defines the acceptance code and so determines those bits of the first byte of the CAN identifier that should pass.

Example:

Under CANopen Protocol only SDO1S2M telgrams, that is CAN ID's 0x580 .. 0x5FF, are to be received into the RECObject.
The acceptance filter will be set as follows:

Acceptance Mask (1st byte of the 1st Receive object) 00001111 (0x0F)
Acceptance Code (2nd vyte of the 1st Receive object) 1011XXXX (0xB0)

Also this RecObject has a handshake bit by which it is possible to determine if a CAN telegram was received by the RecObject.
The RecObject is composed of the CAN Header and the eight (8) data bytes.

Layout of the RecObject:
- CAN header high byte
- CAN header low byte
- Data bytes 1 .. 8

5.6 Structure TrObjekt

Additional CAN messages can be defined by means of the TrObject. Like all others, this message is only transferred to the CAN bus if the corresponding handshake bit changes its status. The TrObject consists of the CAN header and 8 data bytes.

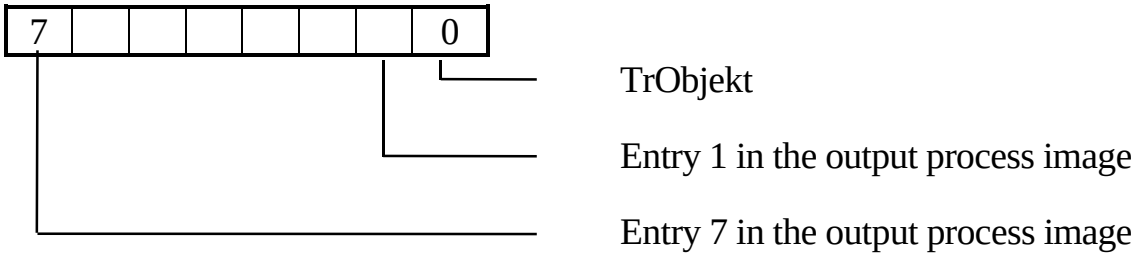
Structure TrObjekt:
High Byte Can-Header
Low Byte Can-Header
Data 1..8

5.7 Handshake

5.7.1 Sending area

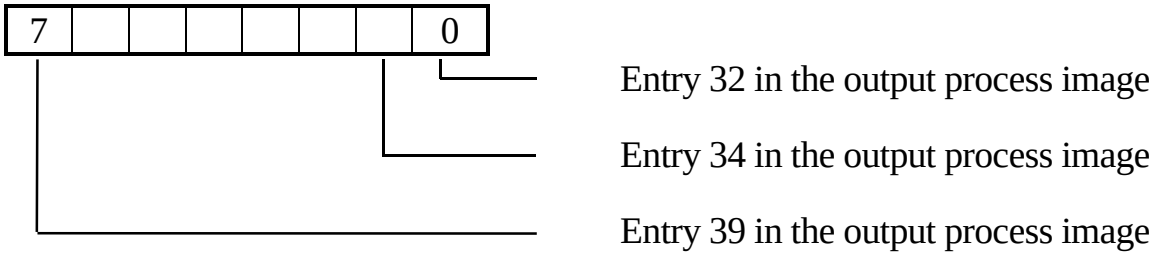
Data is transferred only over the CAN bus if the corresponding bit changes its status.

Byte 0



....

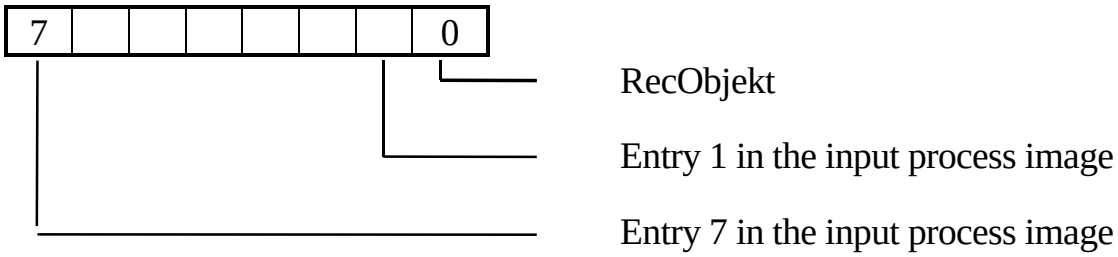
Byte 4



5.7.2 Receiving area

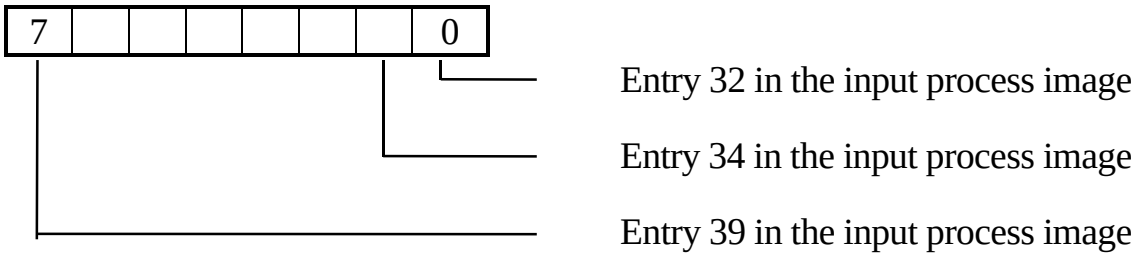
During a change of status, every bit links a received CAN message to the corresponding parameterised identifier in the input process image.

Byte 0



:

Byte 4



5.8 Example

In order to shorten the time required for putting the installation into operation and for troubleshooting, it is advisable to draw up an index list of all modules in the CAN ring, their CAN identifiers, function codes, and byte lengths.

Example A
Three CAN bus nodes are configured in the following way:

1st node
1. digital input module with eight 8Bit input channels
2. digital output module with eight 8Bit output channels
3. Node address set to 1

CAN-HEADER

	Function ID10..ID7	Module-ID ID6..ID0	RTR	Length L3..L0	HighByte	LowByte
Receiving identifier:	0011	0000001	0	1000	0011 0000 30hex	0010 1000 28hex

Sending identifier:	0100	0000001	0	1000	0100 0000 40hex	0010 1000 28hex

2nd node:

1. digital input module with one 8Bit input channels
2. digital output module with one 8Bit output channel
3. node address set to 11

CAN-HEADER

	Function ID10..ID7	Module-ID ID6..ID0	RTR	Length L3..L0	HighByte	LowByte
Receiving identifier:	0011	0001011	0	0001	0011 0001 31hex	0110 0001 61hex
Sending identifier:	0100	0001011	0	0001	0100 0001 41hex	0110 0001 61hex

3rd node:

1. digital input module with eight 8Bit input channels
2. Node address set to 2

CAN-HEADER

	Function ID10..ID7	Module-ID ID6..ID0	RTR	Length L3..L0	HighByte	LowByte
Receiving identifier:	0001	0000010	0	1000	0001 0000 10hex	0100 1000 48hex

4. Window for BROADCAST-message (example)
- Output with 2*8Bit output channels

CAN-HEADER

	Function ID10..ID7	Module-ID ID6..ID0	RTR	Length L3..L0	HighByte	LowByte
Sending identifier:	0000	0000000	0	0010	0000 0000 00hex	0000 0010 02hex

This results in the following index list

		User specific Parameter (hex)	Configuration Data GSD-file
Always 00		00	
CAN Baudrate (500kBaud)		01	
Number of receiving objects		04	
Number of sending objects		03	
RecObjekt (Empfobj 1)	Highbyte	0F	
	Lowbyte	B0	

Receiving Object 1	Highbyte	30	
	Lowbyte	28	DI 8 Byte
Receiving Object 2	Highbyte	31	
	Lowbyte	61	DI 1 Byte
Receiving Object 3	Highbyte	10	
	Lowbyte	48	DI 8 Byte
Send Object 1	Highbyte	00	
	Lowbyte	02	DO 2 Byte
Send Object 2	Highbyte	40	
	Lowbyte	28	DO 8 Byte
Send Object 3	Highbyte	41	
	Lowbyte	61	DO 1 Byte

Telegrams with CAN ID's 0x580 .. 0x5FF are to be received into the RecObject. (Please, refer to the example in Chapter 4.5)

This results in the following configuration data (in hex, separated by comma):

user specific parameters:
00,01,04,03,0F,B0,30,28,31,61,10,48,00,02,40,28,41,61,00,00,00.....

Configuration data:

HANDSHK_IN
HANDSHK_OUT
RecObjekt
TrObjekt
DI 8 Byte
DI 1 Byte
DI 8 Byte
DO 2 Byte
DO 8 Byte
DO 1 Byte

Chapter 6

6 Projects in STEP 7

Proceed as follows:

Copy the GSD file PDP2CL2.GSD to the subdirectory ..\S7DATA\GSD.
Use command “update GSD files” in order to update the hardware list.
Activate Slave PDP2CL2 from the path ‘Other field devices’, ‘others’
The possible configuration Ids will be displayed as in figure 4:Slave modules.
Drag & Drop the PDP2CL2 slave on the Profibus(1)-Net.
Assign the desired address to the slave.

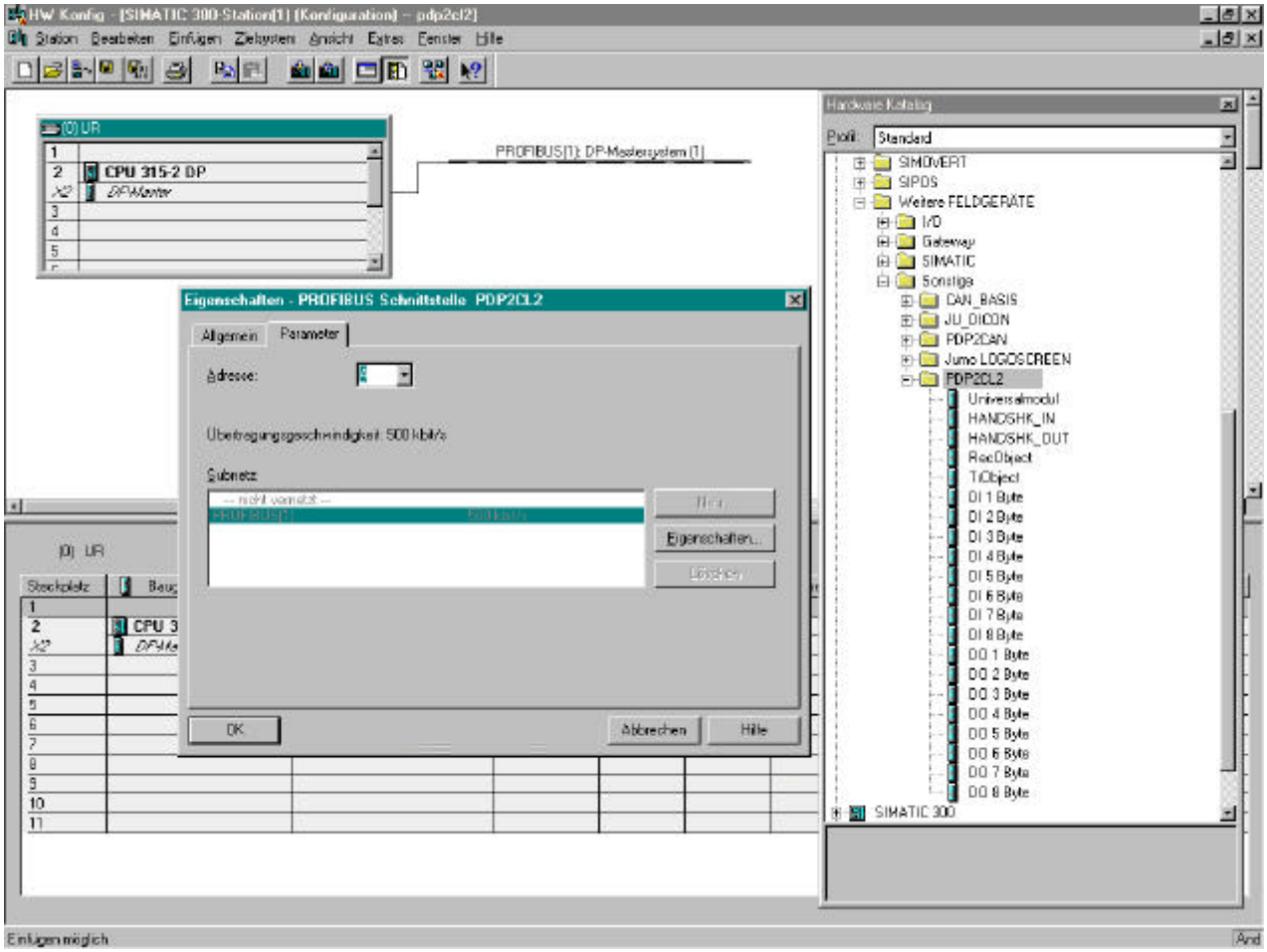


Figure 4: Slave Module

Take the desired PDP2CL2 module from the hardware list and place it (per drag & drop) into the slave's list according to the configuration list.
When selecting the DP slave properties in the parameterisation part, you can carry out general and user-specific parameterisation as shown in figures 5 and 6. Figure 6 also shows a sample parameterisation.

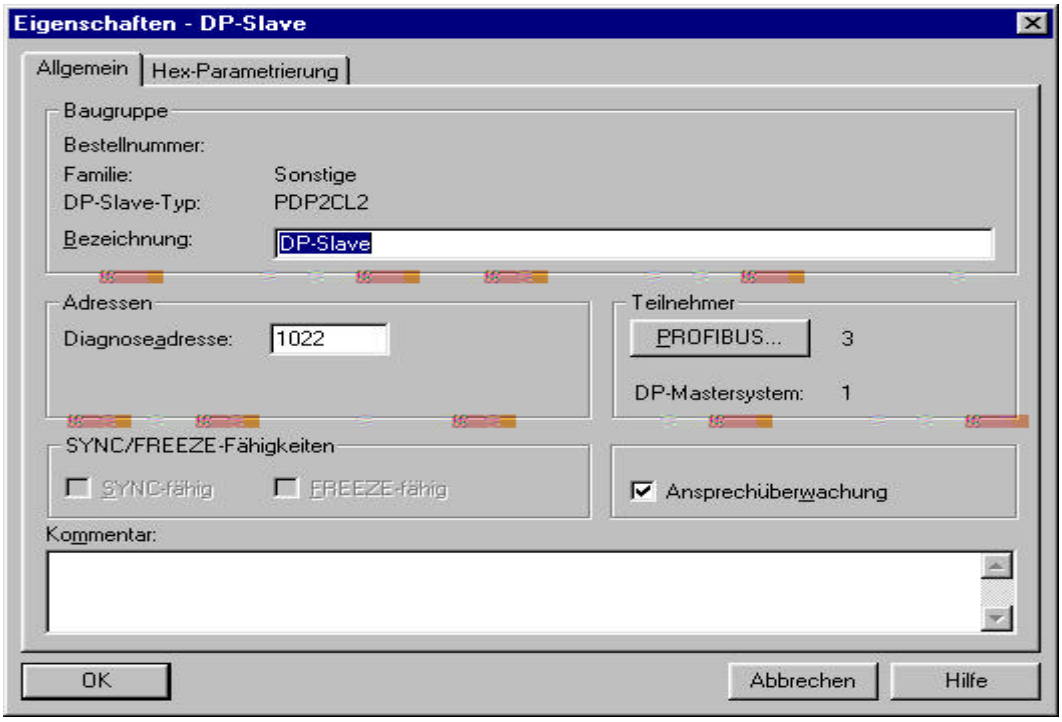


Figure 5: Properties DP slave

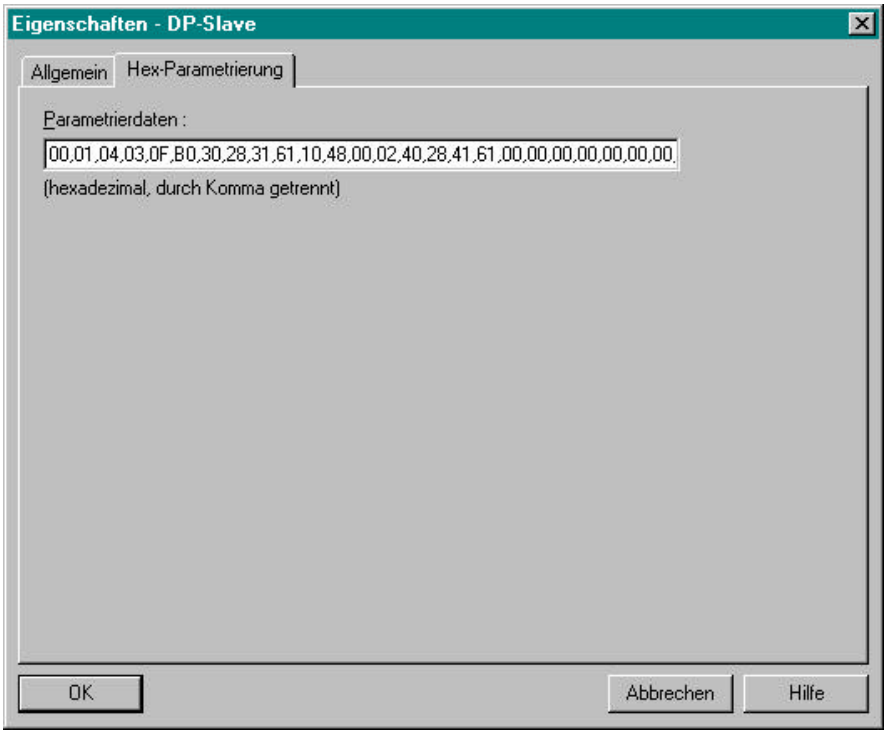


Figure 6: user-specific Parameterisation DP-Slave

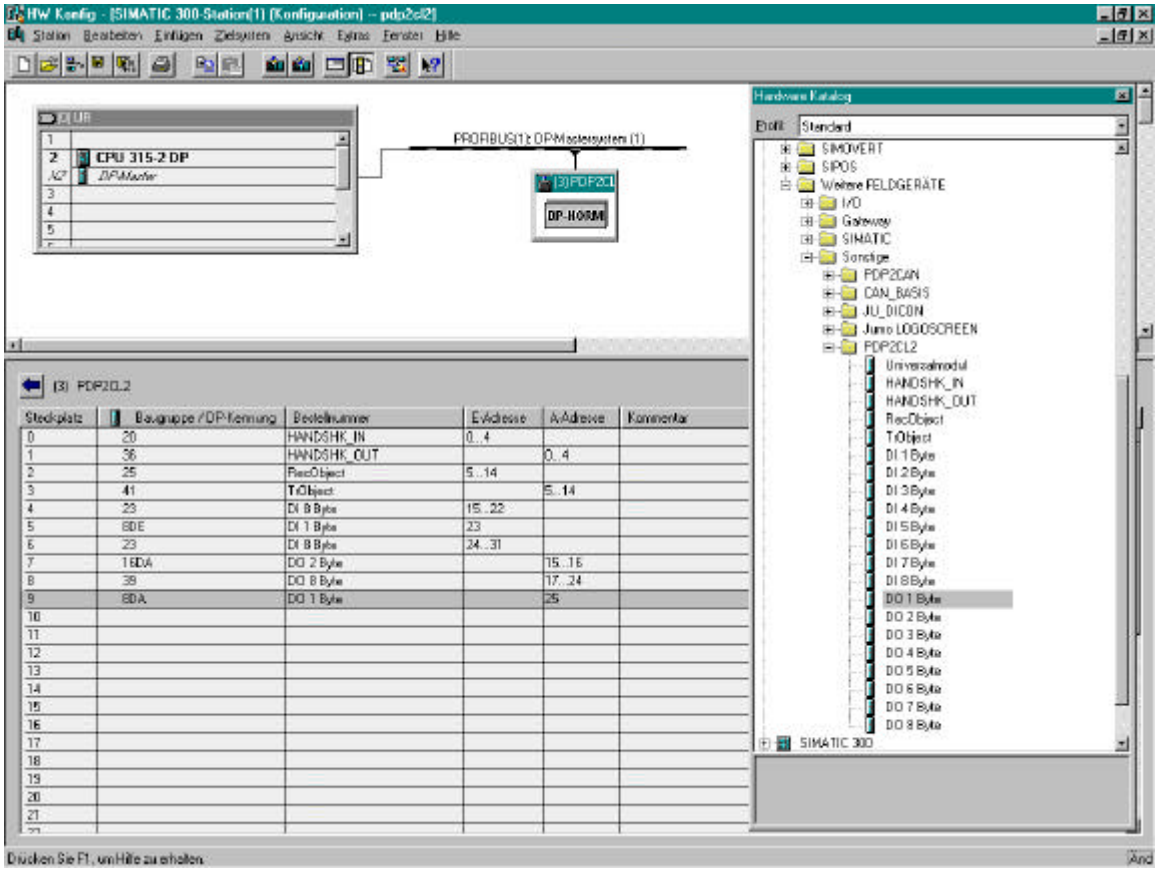


Figure 7: Process Image DP-Slave

7 Specifications

Power supply:	24V DC
Power consumption:	ca. 150mA
Galvanic Insulation:	1 kV DC
Bus speed CAN:	max. 1 MBit
Bus speed Profibus:	ma. 12 MBit
Protection:	IP 20
Dimensions:	Height: 80mm
	width: 23mm
	depth: 90mm
Attachment:	Hat rail

General Warning!

In order to conform to EMV regulations, all data lines must be shielded. This shield must be connected to the earth potential. All earth clamp of our modules must be connected to the earth potential, too.
Antal Electronic will not guarantee compliance to EMV protection measures if these measures are not taken.