PC P-NET INTERFACE

PD 3930

Manual

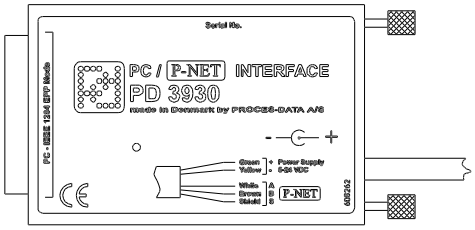
GB *April 1998*

PROCES-DATA A/S, Navervej 8, 8600 Silkeborg, Denmark, phone +45-87 200 300, fax + 45-87 200 301

Contents

1 General information

The PD 3930 is a communication module, providing an interface from the P-NET Fieldbus to a PC. The module is connected to the PC by means of a standard Bi-directional, ECP or EPP parallel interface port on the PC.



490288 01

1.1 Features

- Integrates PC's running VIGO 4.00 or higher, with P-NET.
- Fully configurable via P-NET.
- Baud rates 1200, 9600 or 76800.
- Galvanically isolated P-NET port.
- P-NET Fieldbus Communication, European standard EN 50170, Vol. 1.
- Multi-master P-NET interface.
- PC parallel port interface.
- IP50 mounting box.
- EMC approved (89/336/EEC)

1.2 System description

The PD 3930 PC P-NET interface module provides an interface between the P-NET Fieldbus and a standard PC running VIGO version 4.00 or higher. It is connected to P-NET via a shielded 4-wire twisted pair cable, and has a built in DSUB/25 male connector for connecting to a parallel port on the PC.

Power to the module is supplied either by means of a mini jack connector, or via the twisted pair cable also used for P-NET connection. If supplied by means of the mini jack connector, power might be able to be supplied from the PS2 connector for mouse or keyboard on the PC, provided the PC is able to deliver 350 mA at 5V (400 mA at power up).

The PD 3930 is connected to the PC via a standard IEEE 1284 parallel port. It can be plugged directly into the parallel port, or by means of an IEEE 1284 approved cable with a maximum length of 1.5 m.

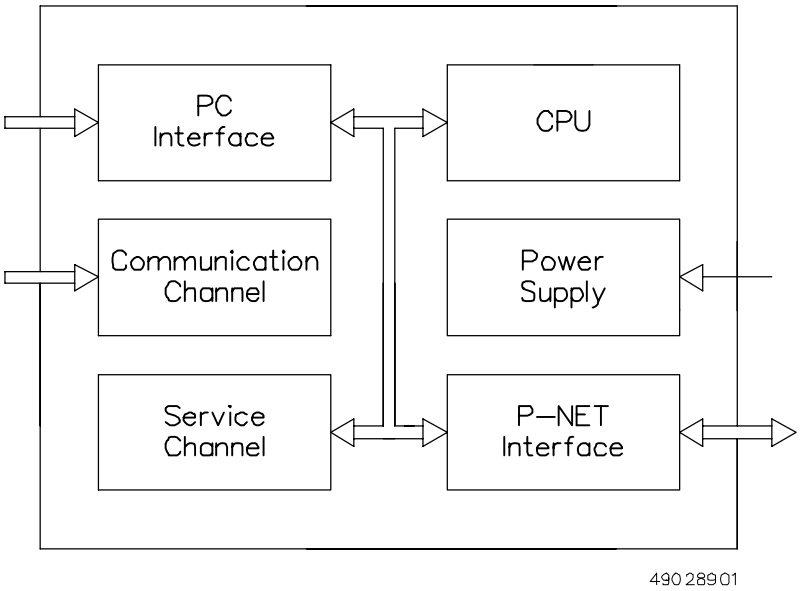
The PD 3930 can act as a P-NET master and as a P-NET slave. The module has 2 channels, - the Service Channel (channel 0) and the Communication Channel (channel 1).

The Service channel, which is mandatory in all P-NET modules, holds general information about the module, such as ID, serial number, P-NET node address and error status of the complete module.
The Communication channel holds all data applicable to the P-NET port set up (P-NET node

address, baud rate etc.). The Communication channel also holds the error status of the channel. Errors reported include configuration errors, communication errors and module errors.

1.3 Channels/registers

The PD 3930 module contains:



1 Service Channel (channel 0)
1 Communication Channel (channel 1)

A set of 16 variables, numbered from 0 - $F, are associated with each channel. To address a variable within a particular channel, a logical address called a SoftWire Number (SWNo), is used. The SWNo is calculated as: (channel number ∗ $10) + variable number within the channel.

Example: Variable 9 on channel 1 needs to be addressed.
The SWNo will therefore be $19.

Throughout the manual, the variables are depicted within tables. The variable names are standard identifiers, as defined in Process-Pascal.

1.4 Memory types

The PD 3930 stores data in different types of memory, depending on the value of a control variable following a reset or a power failure, and the state of write protection.

The following memory types are listed in the channel definition tables.

Read Only

PROM ReadOnly
: The PROM is always write protected and can never be changed.

RAM ReadOnly
: The variable is stored in RAM and is only accessible for Reading.

Read, Protected Write

RAM RPW (Read, Protected Write)
: Data stored in this type of memory is always write protected directly following a reset. By setting WriteEnable to TRUE, the contents of the RAM can be changed. The contents of the RAM will remain unchanged, even after a power failure.

Read Write

RAM ReadWrite
: The variable can be changed instantly. After reset or a power failure, it's value is set to zero.

2 Service channel (channel 0)

Variables on Service channel (channel 0). Channel identifier: **Service**

SWNo	Identifier	Memory type	Read out	Type
0	NumberOfSWNo	PROM Read Only	Decimal	Integer
1	DeviceID	PROM Read Only	- - - - - - -	Record
2				
3	Reset	RAM Read Write	Hex	Byte
4	PnetSerialNo	Special function	- - - - - - -	Record
5				
6				
7				
8				
9				
A				
B				
C				
D	WriteEnable	RAM Read Write	Binary	Boolean
E	ChType	PROM Read Only	- - - - - - -	Record
F	CommonError	RAM Read Write	- - - - - - -	Record

SWNo 0: NumberOfSWNo
This variable holds the highest SWNo in the module

SWNo 1: DeviceID
The purpose of this record is to be able to identify the device. The record includes a registered manufacturer number, the type number of the module and a string, identifying the manufacturer.

The record is of the following type:

```
Record
    DeviceNumber: Word;          (* Offset = 0 *)
    ProgramVersion: Word;        (* Offset = 2 *)
    ManufacturerNo: Word;        (* Offset = 4 *)
    Manufacturer: String[20];    (* Offset = 6 *)
end
```

The field values in the DeviceID record is shown below:

```
DeviceNumber = 3930
ProgramVersion= 100              (the first version)
ManufacturerNo = 1
Manufacturer = Proces-Data DK
```

SWNo 3: Reset
By writing $FF to SWNo 3, the module performs a reset, and ExternalReset in CommonError SWNo $F is set TRUE.

SWNo 4: PnetSerialNo
This Variable is a record having the following structure:

```
Record
    PnetNo: Byte;          (* Offset = 0 *)    (* Node Address *)
    NoOfMasters: Byte;     (* Offset = 1 *)
    SerialNo: String[20];  (* Offset = 2 *)
end
```

The serial number is used for service purposes and as a 'key' to setting the module's P-NET Node Address, NoOfMasters and ErrorcheckMethod from the P-NET connection.

A special function is included for identifying a module connected to a network containing many other modules, having the same or unknown node addresses, and to enable a change of the node address via the P-NET.

Setting a new Node Address, NoOfMasters and ErrorcheckMethod via the P-NET is performed by writing the required Node Address, NoOfMasters and ErrorcheckMethod together with the serial number of the module in question, into the PnetSerialNo at node address 126 ($7E), calling all modules. All modules on the P-NET will receive the message, but only the module with the transmitted serial number will store the P-NET Node Address, NoOfMasters and ErrorcheckMethod. The most significant bit (bit 7) in NoOfMasters defines the ErrorcheckMethod to be used. If the most significant bit is a “1", the ErrorcheckMethod to be used is “Normal”, and if the most significant bit is a “0" the ErrorcheckMethod is set to “Reduced”.

When Node Address, NoOfMasters and ErrorcheckMethod are changed, they will simultaneously be changed in the appropriate registers in the communication channel.

An attempt to write data to node address $7E will give no reply. Consequently the calling master must disable the generation of a transmission error when addressing this node.

In the module, the SerialNo = "XXXXXXXXPD", is set by **PROCES-DATA**, and cannot be changed. The eight X`s indicate the serial number, and PD is the initials of PROCES-DATA.

SWNo $D: WriteEnable
Write protected variables can only be changed when WriteEnable is TRUE. After reset, WriteEnable is set to FALSE.

SWNo $E: ChType

Each channel in an interface module is described in an individual ChType variable. This is a Record, consisting of a unique number for the channel type and a TRUE boolean value for each of the registers which are represented within a channel. The register number in a channel, corresponds to the index number in the boolean array. In addition to these fields, various other fields can be found in the record, the contents of which depend on the channel type.

The record for the service channel has the following structure:

```
Record
    ChannelType:    WORD;       (* Offset = 0 *)
    Exist:          Bit16;      (* Offset = 2 *)
    Functions:      Bit16;      (* Offset = 4 *)
end
```

For the service channel, ChType has the following value:

ChannelType = 1

Exist =

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	0	0	0	0	1	1	0	1	1

Functions =

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

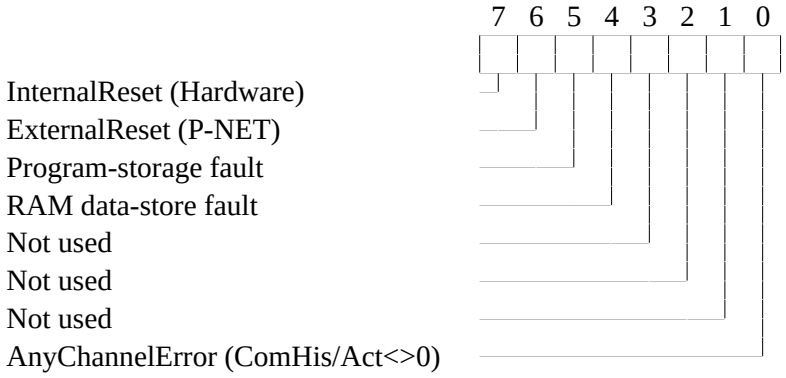
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
No Watchdog
Watchdog

SWNo $F: CommonError

The CommonError variable holds error information on all Channels. This variable is a record having the following structure:

```
Record
    ChError: Record
                His:Array[0..7] of Boolean;     (* Offset = 0 *)
                Act:Array[0..7] of Boolean;     (* Offset = 2 *)
            End;
    ComHis8:Array [0..7] of Boolean;            (* Offset = 4 *)
    ComAct8:Array [0..7] of Boolean;            (* Offset = 6 *)
End
```

The 8 bits in ChError.His and ChError.Act have the following meaning:



Bit 7 InternalReset is set TRUE, if a reset is caused by a power failure, or if the power has been disconnected.

Bit 6 ExternalReset is set TRUE, if a reset is caused by writing $FF to SWNo 3, Reset, via P-NET.

Bit 5 Program-storage fault is set TRUE, if the self test finds an error in the program memory (PROM).

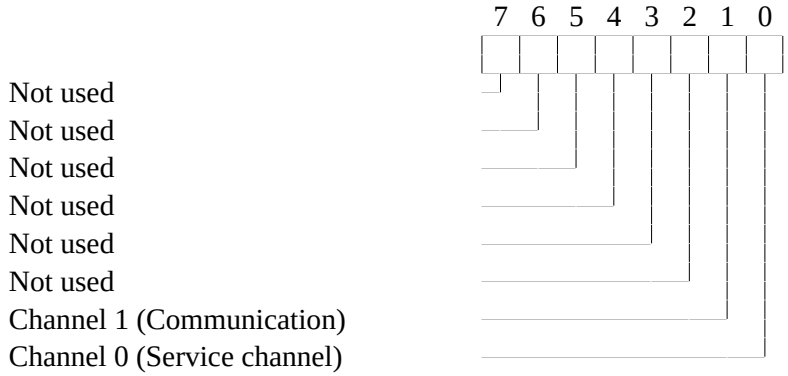
Bit 4 RAM data-store fault is set TRUE, if the self test finds an error in the data memory (RAM).

Bit 0 AnyChannelError = 1, means that an error, or an unacknowledged error, exists in one or more channels.

The following function of ChError.His and ChError.Act is analogous for all Channels:

1 When an error occurs the corresponding bits in ChError.Act and ChError.His is set.
2 When the error disappears the corresponding bit is reset in ChError.Act.
3 After reading ChError.His, ChError.Act is copied to ChError.His.
4 Transmission responses from a module will include the Actual Data Error bit (DataError) set TRUE if ChError.Act <> 0.
5 The Historical Data Error bit (GeneralError) will be set TRUE in all responses from the module if ChError.His <> 0.

ComHis and ComAct are unique fields in the service channel, and hold an error status relating to all channels, where the bit number corresponds to the channel number. Each Channel has an error register, ChError. If ChError.His in a particular channel is<>0, the corresponding bit is set in ComHis. If ChError.Act in a particular channel is<>0, the corresponding bit is set in ComAct in the service channel. If the error disappears (ChError. Act = 0), the corresponding bit in ComAct is automatically cleared.
If the channels become error free, individual bits in ComHis will be cleared when reading ChError in each of the channels.
ComHis:=0 performs a special function, equivalent to reading all ChErrors.His in all channels.



3 Communication channel (channel 1)

Variables on Communication channel Channel identifier: **Port_1**

SWNo	Identifier	Memory type	Read out	Type
10				
11				
12				
13				
14				
15	NodeParam	RAM RPW	- - - - - - - -	Record
16	Baudrate	RAM RPW	Decimal	LongInteger
17				
18				
19	ChConfig	RAM RPW	- - - - - - - -	Record
1A				
1B				
1C				
1D				
1E	ChType	PROM ReadOnly	- - - - - - - -	Record
1F	CHError	RAM ReadOnly	Binary	Record

SWNo $15: NodeParam

```
Record
      PNETNo          : BYTE;
      NoOfMasters     : BYTE;
end
```

PNETNo defines the P-NET number (Node address) of the module.

NoOfMasters defines the number of masters for the module.

The values of these registers may also be changed via SWNo $04. When this is done, the values are always transferred to the RAM, regardless of the state of WriteEnable.

SWNo $16: BaudRate

This variable selects the baud rate for the module. If an illegal baud rate is selected, i.e. the baud rate is not implemented, an error code is generated in ChError. BaudRate can hold the following values: 76800, 9600, 1200

SWNo $19: ChConfig
This variable selects the mode of operation for the channel. The ChConfig variable is a record having the following structure:

```
Record
    Enablebit   : Bit8;          (* Offset = 0 *)
    Functions   : BYTE;          (* Offset = 1 *)
    Ref_A       : BYTE;          (* Offset = 2 *)
    Ref_B       : BYTE;          (* Offset = 3 *)
end
```

where each field is interpreted as follows:

Enablebit is not used.

Functions:
The Function field holds the following:
Functions = $00 => Channel disabled
Functions = $01 => P-NET with reduced errorcheck
Functions = $02 => P-NET with normal errorcheck

Ref_A is not used.

Ref_B is not used.

Every time a write operation is performed to either NodeParam (SWNo $15), BaudRate (SWNo $16) or ChConfig (SWNo $19), the P-NET module is initialised, using the new contents of these variables.

SWNo $1E: ChType
For the Communication Channel, ChType has the following form:

```
Record
    ChannelType: WORD;           (* Offset = 0 *)
    Exist: Bit16;                (* Offset = 2 *)
    Functions: Bit16;            (* Offset = 4 *)
end
```

For the communication channel, ChType has the following value:
ChannelType = 14

Exist =

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0

Functions =

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

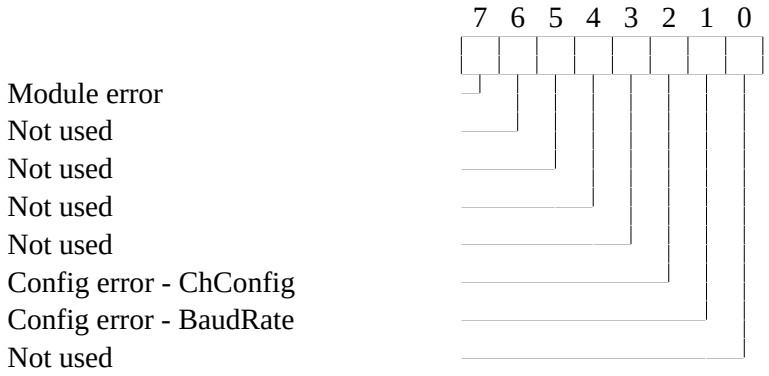
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
Not used
DatamodeInOut
DatamodeOut
DatamodeIn
P-NET Normal error
P-NET Reduced error
Can be disabled

When selection of un-supported Baudrates, protocols etc. is attempted, configuration errors will be generated in ChError.

SWNo 1F: ChError

```
Record
    His:Array[0..7] of Boolean;     (* Offset = 0 *)
    Act:Array[0..7] of Boolean;     (* Offset = 2 *)
End
```

The 8 bits in ChError.His and ChError.Act have the following meaning. When an error occurs, the corresponding bit is set in both ChError.His and ChError.Act. When the error disappears, the bit is cleared in ChError.Act.



Bit 7 **Module error.** If this bit is set, the rest of the bits have no meaning, because a module error can cause random error codes within the individual channels (see also "Service channel").

Bit 2 **Config error - ChConfig** is generated when an illegal configuration of the ChConfig variable is attempted. The error is generated in ChError.His and in ChError.Act. The error is cleared in ChError.Act by writing a legal configuration to the ChConfig variable.

Bit 1 **Config error - BaudRate** is generated when an illegal BaudRate selection is attempted. The error is generated in ChError.His and in ChError.Act. The error is cleared in ChError.Act by writing a legal BaudRate value to the BaudRate variable.

4 Installation

Before the PD 3930 module is connected to a PC, the following should be noted:

1: Never plug or unplug the PD 3930 module when the PC is turned ON.

2: To communicate, the parallel port in the PC (normally LPT1) should be set to Bi-directional, ECP or EPP mode (NOT printer-mode). This is normally done by means of the C-MOS BIOS setup facility in the PC. This setup is entered by pressing a key during startup of the PC - the specific key can differ from PC to PC. If the PC does not which key to press on the screen during startup, try <ESC>, <DEL>, <F1> or <F2>. To enable communication, an interrupt must be allocated to the parallel port.

3: The PC software driver for the PD 3930 P-NET module, is installed along with the VIGO system version 4.00 or higher, or from a separate installation floppy disk.

4: Once the software is installed, parameters, such as node address etc. can be set up from the PC using the VIGO system, or the parameters can be set up from another node connected to P-NET.

If the parameters in the PD 3930 module are changed from another P-NET node, the values are automatically transferred to the parameters shown in VIGO.

Note, that when the power to the module is turned off, the parameters in the module are lost. But as the values are saved in the PC under the VIGO system, they will be restored next time the module is used from VIGO.

5: The VIGO system is provided with 3 different drivers for the PD 3930 module, which apply to LPT1, LPT2 and LPT3. Therefore, up to 3 modules can be connected to the same PC.

6: Variables in the PD 3930 module (for example Softwire $16, Baudrate) can be accessed from another P-NET node as variables in any other node. But they can also be accessed from the PC, as if they were placed in a “normal” P-NET node. This is done by defining a node in the MIB (refer to the VIGO user manual) of type PD 3930. This node must have the same Node Address as the PD 3930 module. When this node is accessed from for example the MONITOR, the PD 3930 will respond as a “normal” P-NET node, but communicating on the P-NET field bus.

5 Mechanical specifications

The PD 3930 module is housed in a black aluminium case.
The Module is designed for plugging directly into the parallel port of a PC. The module is fixed with the two finger-screws.

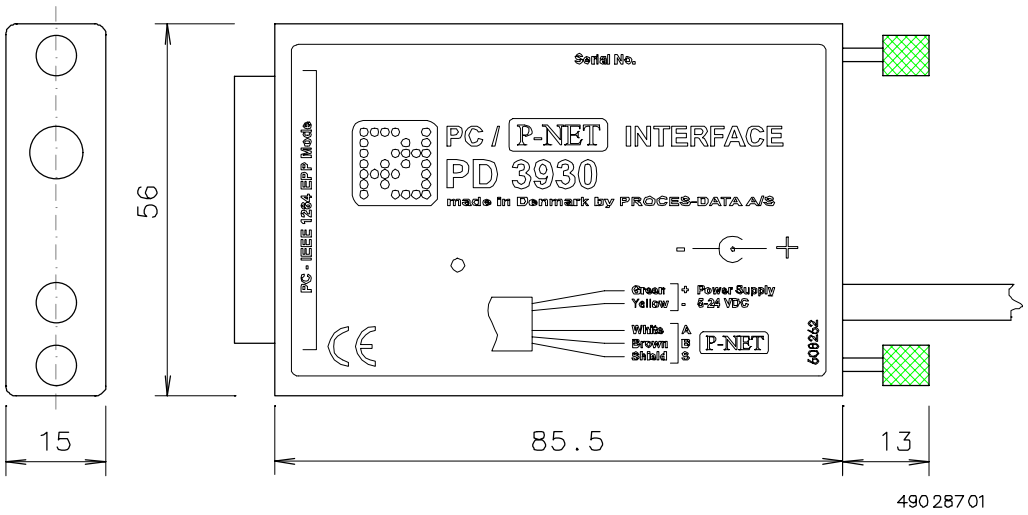
5.1 Materials

Case:	Black anodized aluminum
Front foil:	Polycarbonate
Weight:	150 gram

5.2 Sealing

IP 50

Scale drawing in mm:



6 Electrical specifications

All electrical characteristics are valid at an ambient temperature of -25°C to 70°C, unless otherwise is specified.

All specifications are valid within the range of approved EMI conditions. EMC test specifications for the PD 3930 are available in a separate document. The document is available from PD on request.

6.1 Power supply

Power supply DC:	nom. 24.0 V
	min. 5.0V
	max. 28.0 V
Ripple:	max.5 %
Power consumption:	max.1.6 W
Current at 5V supply:	350 mA
Current at power up:	max.400 mA

6.2 P-NET

Galvanically isolated

6.3 LED indicator

The green LED indicator will illuminate if the power connections are correct.

6.4 PC interface

IEEE 1284

6.5 Interface cable

length: 3 m

Colour code:

24V+ : Green
24V- : Yellow
P-NET A : White
P-NET B : Brown
P-NET S : Shield

6.6 Mini-jack power connector

The mini-jack next to the interface-cable, can also be used to supply the module with power. When the mini-jack is in use, the green wire (24V+) in the interface-cable is disabled.

6.6	Ambient temperature	
	Operating temperature:	-25 °C to +70 °C
	Storage temperature:	-40 °C to +85 °C
6.7	Humidity	
	Relative humidity:	max.95 %
6.8	Approvals	
	Compliance with EMC-directive no.:	89/336/EEC
	Generic standards for emission:	
	Residential, commercial and light industry	EN 50081-1
	Industry	EN 50081-2
	Generic standards for immunity:	
	Residential, commercial and light industry	EN 50082-1
	Industry	EN 50082-2