502 074 01

CONDUCTIVITY TRANSMITTER

PD 3270

Manual

GB

December 1994

Proces-Data Silkeborg ApS, Navervej 10, DK-8600 Silkeborg, Denmark, phone +45-86814033, fax +45-86814088

Contents

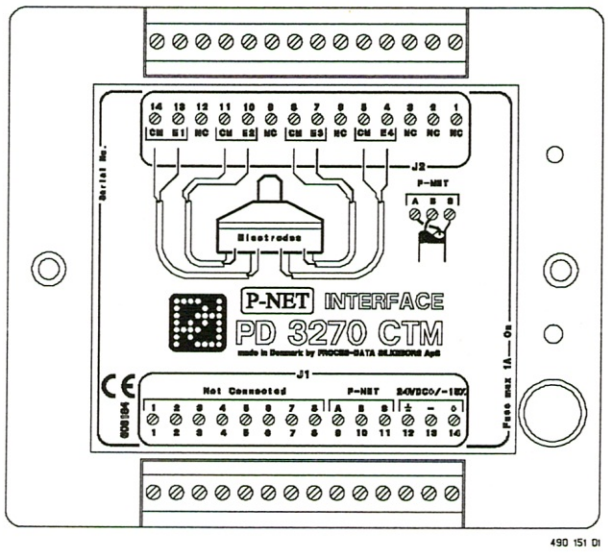
1 General information.

The PD 3270 Conductivity Transmitter is a member of Proces-Data's module series 3000.

The PD 3270 is an intelligent module, provided with 1 Conductivity Channel, and an internal user programmable Calculator Channel.

The P-NET is a field-bus network designed for process control and data collection. The PD 3270 module has been developed for interfacing direct to process signals.

Configuration of the module for the functions required, and communication between the module and a control computer, is carried out via the P-NET.



The unit provides high resolution measurements of a value related to conductivity. The continuous fully integrating non sampling principle offers high noise immunity. The influence of periodic disturbances can be damped with the adjustable averaging function.

The module possesses a programmable calculator channel. With the utilization of a small user programme it is possible to convert the conductivity related measurement value to e.g. a resistance or the actual conductance expressed in SI units.

The compact design and the outstanding environmental specifications for the Conductivity Transmitter makes it an ideal process component in industrial as well as other environments.

1.1 Features.

- 1 Conductivity Channel
- High Resolution non sampling, fully integrating principle
- 1 Programmable Calculator Channel
- Continuous Selftest, which can be monitored through the P-NET.
- P-NET Fieldbus Communication
- Watch Dog Timer
- Rail mounting module (DIN / EN)
- EMC approved (89/336/ECC)

1.2 System description.

The Conductivity Transmitter has four inputs for conductivity measurements. The inputs are summed to minimize influence from momentarily shortcircuited electrodes in mixers etc.

The measurement is not converted into engineering units but to a value related to conductivity.

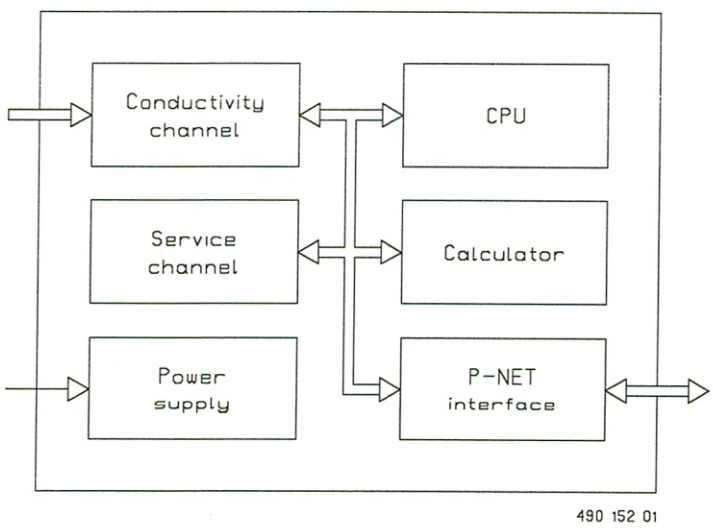
Presetable limit switches combined with the user programmable calculator reduce the basic operations in the central control system. The unit offers comprehensive self-testing features, which enables reporting of connection errors or internal errors.

PD 3270 is approved in compliance with the **EMC-directive no 89/336/ECC**. Test limits are determined by the generic standards **EN 50081-1** for emission and **PrEN 50082-2** for immunity.

PD 3270 is approved in compliance with the **IEC 68-2-6 Test Fc** standard for vibration.

1.3 Channels/registers.

The PD 3270 module contains:



1 Service channel	(channel 0)
1 Conductivity channel	(channel 8)
1 Calculator channel	(channel 9)

A set of 16 variables numbered from 0 - $F, are associated with each channel. For addressing 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 4 on channel 8 needs to be addressed. The SWNo will therefore be $84.

Throughout the manual the variables are depicted as tables.

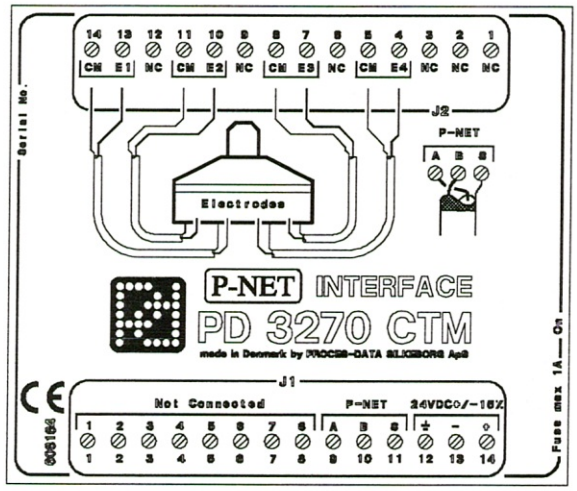
1.4 Connections.

The PD 3270 Conductivity Transmitter is physically designed as a black box, having two 14 pin connectors for screw terminals.The connectors are removable and equipped with a key pin, to avoid reversed connections.

The module wiring should be designed with a maximum of 2 wire connections in each screw terminal.

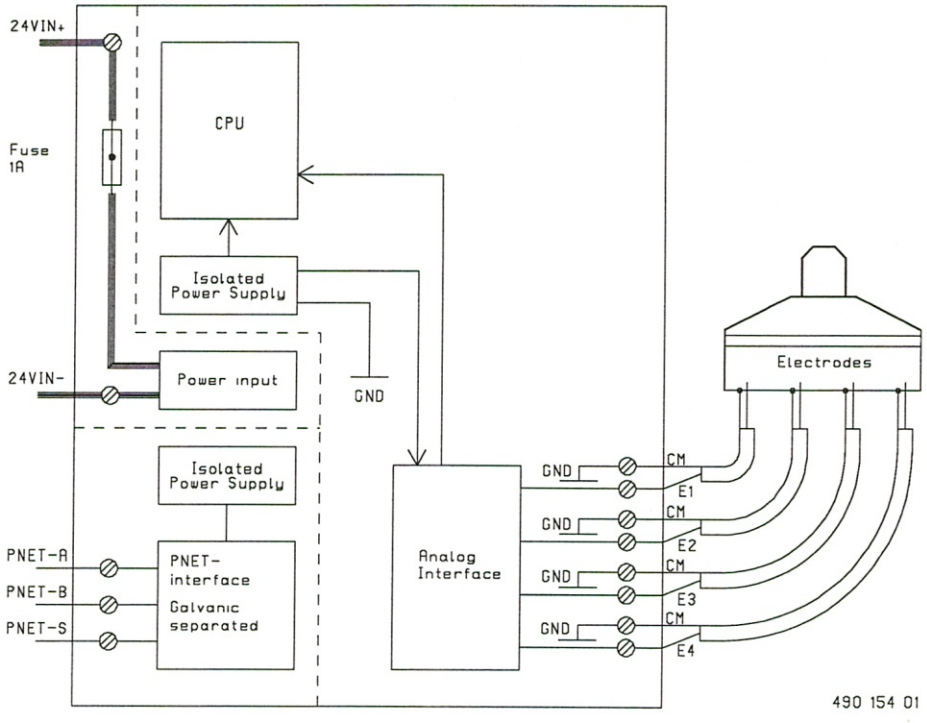
The module has a built in fuse to protect the module.

Connection identities are printed on top of the module.



490 150 01

Hardware diagram, principle.



1.5 Memory types.

The PD 3270 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.

Some variables are stored in both non volatile memory and in volatile memory. The state of the module's WriteEnable register determines whether the contents are changed in both types of memory or only in the volatile type.

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

EEPROM RPW (Read, Protected Write)

The EEPROM is always write protected directly following a reset. By setting WriteEnable to TRUE, the contents of the EEPROM can be changed. The contents of the EEPROM will remain unchanged during and after a power failure.

RAM RPW (Read, Protected Write)

The variable is always write protected following a reset. By setting an input simulation boolean to TRUE, the contents of the RAM can be changed.
NB: This memory type is not utilised in the PD 3270 module.

Read Write

RAM ReadWrite

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

Read Write, Protected BackUp Write

RAM InitEEPROM

The variable is stored in both RAM and EEPROM. After a reset, the variable is copied from EEPROM into RAM. When the variable is changed via P-NET, the value is changed in RAM. If WriteEnable is TRUE, the value is changed in both RAM and EEPROM when the variable is changed via P-NET.

RAM AutoSave

Has the same function as RAM InitEEPROM, with the addition that the contents of RAM are automatically copied to EEPROM, at a frequency of approximately 10 hours.

2 Service channel (channel 0).

PD 3270 contains a service channel containing variables and functions common to the entire module.

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

SWNo	Identifier	Memory type	Read out	Type
0	NumberOfSWNo	PROM Read Only		Integer
1	DeviceID	PROM Read Only	- - - - - - - -	Record
2				
3	Reset	RAM Read Write	Hex	Byte
4	PnetSerialNo	Special function	- - - - - - - -	Record
5				
6				
7	FreeRunTimer	RAM Read Only	Decimal	LongInteger
8	WDTimer	RAM Read Write	Decimal	Real
9	ModuleConfig	EEPROM RPW	- - - - - - - -	Record
A	WDPreset	EEPROM RPW	Decimal	Real
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
```

An example of the field values in the DeviceID record is shown below:

DeviceNumber = 3270
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 of the following type:

```
Record
    PnetNo: BYTE; (* Node Address *)  (* Offset = 0 *)
    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 Nodeaddress.

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 via the P-NET is performed by writing the required node address together with the serial number of the module in question, into the PnetSerialNo at node address $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.
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 = "XXXXXXXPD", is set by **Proces-Data**, and cannot be changed. The seven X's indicate the serial number, and PD is the initials of Proces-Data.

SWNo. 7: FreeRunTimer
FreeRunTimer is a timer, to which internal events are synchronized. The timer is of type Longinteger in 1 /256 Second.

P-NET Watch Dog function

PD 3270 is equipped with a P-NET Watchdog, which switches off all the digital outputs, by clearing OutFlags and Control flags, if P-NET communication ceases. The P-NET watchdog uses SWNo 8 and SWNo A.

SWNo. 8: WDTimer [s]

WDTimer is automatically preset with the value from WDPreset (SWNo A), either each time the module is called via P-NET, or following a power-up or module reset. If the WDTimer reaches zero before it is preset again, the PnetWDRunOut flag will be set, and all the outputs will switch OFF. The timer contains a value in sec.

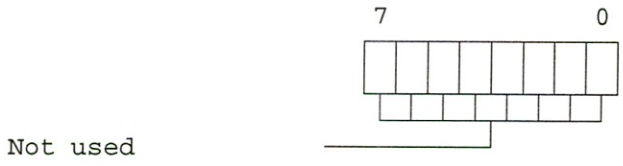
SWNo. 9: ModuleConfig

The variable is a record of the following type:

```
Record
    Enablebit: Array[0..7] Of Boolean;  (* Offset = 0 *)
    Functions: BYTE;                    (* Offset = 1 *)
    Ref_A: BYTE;                        (* Offset = 2 *)
    Ref_B: BYTE;                        (* Offset = 3 *)
End
```

where each field has the following interpretation:

Enablebit :



The watch-dog facility may be switched on and off by means of the field variable Functions as shown below.

ModuleConfig.Functions = 0	watchdog
ModuleConfig.Functions = $10	No watchdog

The Ref_A and Ref_B fields are not utilised in the module.

SWNo A: WDPreset [s]
The maximum allowable time between two calls for the module, before the watch-dog is activated, is defined in seconds, in this register.

SWNo D: WriteEnable
Write protected variables can only be changed when WriteEnable is TRUE ("1"). After reset, WriteEnable is set to FALSE.

After modifying the contents of module EEPROM, WriteEnable should be set FALSE. An EEPROM sum check is calculated each time WriteEnable is changed from "TRUE" to "FALSE". This sum check calculation period is approximately 16 seconds. Consequently, the module should not be reset during this period, otherwise an EEPROM error can occur (see SWNo. F: CommonError).

NB: Writing to EEPROM is limited to 10,000 cycles for each byte, including the sum check bytes.

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, which depends on the channel type.

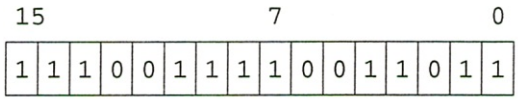
The record for the service channel has the following structure:

```
Record
    ChannelType: WORD;              (* Offset = 0 *)
    Exist: Array[0..15] Of Boolean; (* Offset = 2 *)
    Functions: Array[0..15] Of Boolean; (* Offset = 4 *)
End
```

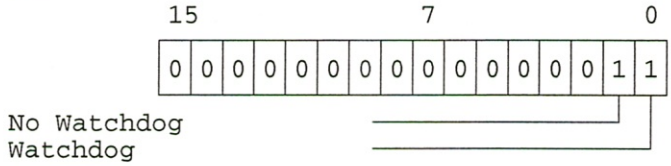
For the service channel, ChType has the following value:

ChannelType = 1

Exist =



Functions =



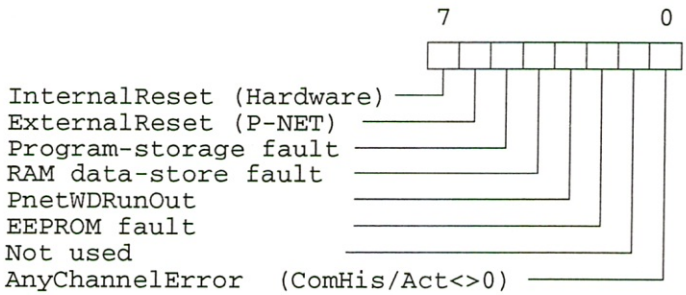
SWNo. F: CommonError

The CommonError variable holds error information on all Channels.

This variable is a record of the following type:

```
Record
    ChError: Record
                His:Array[0..7] Of Boolean;   (* Offset = 0 *)
                Act:Array[0..7] Of Boolean;   (* Offset = 2 *)
            End;
    ComHis:Array [0..$F] Of Boolean;          (* Offset = 4 *)
    ComAct:Array [0..$F] 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 3 PnetWDRunOut is set TRUE if the WDTimer reaches zero and the Watchdog function is switched ON.

Bit 2 EEPROM fault is set to TRUE if the self test finds an error in the data memory (EEPROM). The error may be corrected by setting and resetting WriteEnable (the error will disappear after approx. 16 sec.).

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

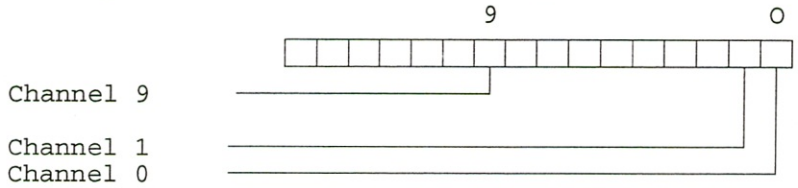
The following function of ChError.His and ChError.Act is analogous in all Channels:

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 in all channels.



3 Conductivity Channel (channel 8).

The conductivity channel uses the structure of a PD 3230 Weight Channel, i.e. the channel contains the same registers and has nearly the same functionality as the PD3230 Weight Channel. It is important for right usage of the Conductivity Channel, that only the registers described in this manual are utilised.

Some of the variables in the Conductivity Transmitter should be set to recommended values, to achieve correct measuring.

The value in the Conductivity register is not converted to engineering units but to a value related to conductivity. The measured value can easily be converted to a conductivity or a resistance by means of a small calculator programme. The input ranges from shortcircuited to disconnected electrodes with resolution centred at 100 Ohm and usable range from 1 Ohm to 10 kOhm.

The module uses a non sampling, full time averaging principle. This principle always suppress 50 and 60 Hz line interference, and is user adjustable to damp the influence from other periodic disturbances.

The channel possess presettable limit switches for the Conductivity variable.

Variables on conductivity channel (channel 8). Channel identifier: **Conductivity**

SWNo	Identifier	Memory type	Read out	Type	
80					
81					
82	Conductivity	RAM Read Write	Decimal	Real	- - -
83					
84	UserReal	RAM Read Write	Decimal	Real	- - -
85	FlagReg	RAM Read Write	Binary	Bit8	- - -
86					
87	HighLevel	RAM Init EEPROM	Decimal	Real	
88	LowLevel	RAM Init EEPROM	Decimal	Real	
89	ChConfig	EEPROM RPW	- - - - - -	Record	
8A	CondResolution	EEPROM RPW	Decimal	Real	- - -
8B	FullScale	EEPROM RPW	Decimal	Real	- - -
8C	ZeroPoint	EEPROM RPW	Decimal	Real	
8D	Maintenance	EEPROM RPW	- - - - - -	Record	- - -
8E	ChType	PROM Read Only	- - - - - -	Record	- - -
8F	ChError	RAM Read Only	Binary	Record	- - -

The following pages contain a modified description of a PD 3230 Weight Channel with comments for conductivity measurements. Some variables and automatic functions should not be used and others should be set to recommended values.

SWNo 82: Conductivity (Conductivity related measurement).
Conductivity holds the fullscale and zeropoint adjusted conductivity related measurement.

Calculation: Conductivity = Input * FullScale - ZeroPoint

The measured value in Conductivity can be converted to engineering units by the expressions:

Measured resistance = 100((FullScale/Conductivity)-1) *[Ω]*

Measured conductivity = 0.01/((FullScale/Conductivity)-1) *[S]*

SWNo 84: UserReal.
If the Conductivity channel is configured as recommended under "ChConfig", this register will not be updated or overwritten by the module. The result of a conversion of the measured value to conductivity or resistance may be placed in this variable.

SWNo 85: FlagReg.


```
7: Not used
6: Not used
5: Not used
4: HighFlag
3: LowFlag
2: Not used
```

HighFlag and LowFlag is set by the HighLevel and LowLevel (see description) limit switches for Conductivity. Operation is not affected by ChConfig.EnableBit.

SWNo 87: HighLevel.
HighLevel is a limit switch for Conductivity with the following function:

IF (Conductivity > HighLevel) AND ChConfig.Enablebit[4]
THEN HighAlarm := True ELSE HighAlarm := False;

IF (Conductivity > HighLevel)
THEN HighFlag := True ELSE HighFlag := False;

SWNo 88: LowLevel.
LowLevel is a limit switch for Conductivity with the following function:

```
IF (Conductivity < LowLevel) AND ChConfig.Enablebit[3]
THEN LowAlarm := True ELSE LowAlarm := False;

IF (Conductivity < LowLevel)
THEN LowFlag := True ELSE LowFlag := False;
```

SWNo 89: ChConfig.
The channel configuration for the conductivity channel is stored in a record with the following type:

```
Record
    Enablebit: Array[0..7] Of Boolean;  (* Offset = 0 *)
    Functions: BYTE;                    (* Offset = 1 *)
    ReadOut: Array[0..7] Of Boolean;    (* Offset = 2 *)
    NoOfSamples: BYTE;                  (* Offset = 3 *)
    BeltRef: BYTE;                      (* Offset = 4 *)
End;
```

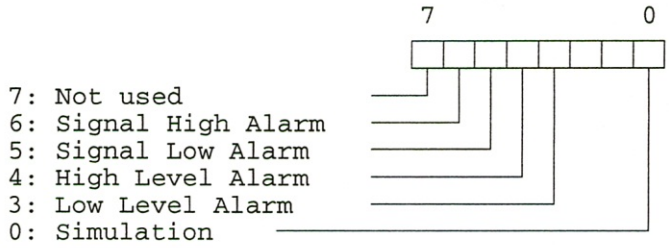
Recommended settings of the Conductivity channel (factory setting):

ChConfig.EnableBit := 011XX000	*Signal Alarm enabled, limit switches selectable*
ChConfig.Functions := $19	*SampleTime: 0.5 s*
ChConfig.ReadOut := 01000000	*Averaging on, rounding off for conductivity*
ChConfig.NoOfSamples := 10	*Average time: 5 s*
ChConfig.BeltRef := 00	*Not relevant*

If the limit switches are selected the wanted limits must be written in the HighLevel and LowLevel variables.

The fields in ChConfig have the following interpretation:

Enablebit:



It is recommended to enable Signal High Alarm and Signal Low Alarm, to achieve an automatic supervision of the connections.

High Level Alarm and Low Level Alarm are user programmable limit switches.

Simulation.
Simulation mode is a tool for testing application programmes or simulation of an active module.

When the module is in simulation mode, values written in the Conductivity variable will not be updated or overwritten by the module. This makes it possible to simulate any value in the variable for test purposes.

The Average function is disabled during simulation. Alarm and error functions maintain the normal function.

Input simulation is enabled by setting ChConfig.EnableBit[0] to TRUE.

Functions:

The functions field is used in a hexadecimal format, where the most significant digit is used to specify the channel functions and the least significant digit is used to specify the time between readouts (SampleTime).

Mode:

Functions = $00 => Channel Disabled
Functions = $1X => Conductivity

SampleTime:

Functions = $X9 => SampleTime = 0.5 s (Calibrated at this value)
Functions = $XA => SampleTime = 1 s (Usable but not calibrated)
Functions = $XB => SampleTime = 2 s (Usable but not calibrated)
Functions = $XC => SampleTime = 5 s (Usable but not calibrated)

If the channel is not used, write "00" in ChConfig.Functions. Otherwise errors may occur.

SampleTime is the time between readouts. A sample represents the averaged measured input since last readout. The module is calibrated at sampletime = 0.5 s, which is also the recommended configuration. Usage of one of the longer sampletimes will damp the influence from a periodic noise signal but it will also result in a small error due to calibration deviations.

ReadOut:

ReadOut bits enables averaging and/or rounding on the main variables. Only bit 6 and bit 2 are relevant for conductivity measurements. The remaining bits should always be set to FALSE ("0").

If averaging of the value in the Conductivity variable is desired the number of samples to be averaged must be inserted in ChConfig.NoOfSamples.

If rounding of the value in the Conductivity variable is desired, the resolution must be inserted in the variable CondResolution.

NoOfSamples:
The number of samples to be averaged must be inserted in NoOfSamples. The value is used for averaged read out of Conductivity. The range for NoOfSamples is 1..32.

The total average time is calculated as SampleTime * NoOfSamples. To damp the influence from vibrations/periodic noise, the total average time should be a multiplum of the vibration/noise period. Note that 50 and 60 Hz is always suppressed by any value of Sample-Time. The best setting is a compromise between speed and noise and can be found by calculation and/or experiments.

SWNo 8A: CondResolution.
This variable offers the facility of rounding the Conductivity variable on SWNo 82 to a valid resolution. If rounding is desired, the resolution must be inserted in CondResolution. The rounding of the variable, is enabled by setting the CondRound bit in ChConfig.ReadOut to TRUE.

SWNo 8B: FullScale.
This variable holds the FullScale value for Conductivity. The value in FullScale corresponds to the reading in Conductivity with shortcircuited electrodes. The module is calibrated to read half the fullscale value with a 100 Ohm resistor on each of the four electrode inputs.

The Service Channel WriteEnable bit must be set TRUE to change the value of FullScale.

SWNo 8C: ZeroPoint.
This variable holds the ZeroPoint value for Conductivity. The value in the ZeroPoint variable should allways be set to zero.

The Service Channel WriteEnable bit must be set TRUE to change ZeroPoint.

SWNo 8D: Maintenance.
The Maintenance variable is used for service management and maintenance purposes, and holds the last date of service and an indication of the type of service.
Date, month, year and category.

Maintenance is a record of the following type:

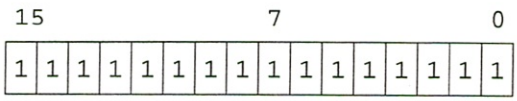
```
Record
    Date: BYTE;      (* Offset = 0 *)
    Month: BYTE;     (* Offset = 1 *)
    Year: BYTE;      (* Offset = 2 *)
    Category: BYTE;  (* Offset = 3 *)
End;
```

SWNo 8E: ChType.
The ChType of the Conductivity Channel described below is identical with the ChType of a PD3230 Weight Channel.

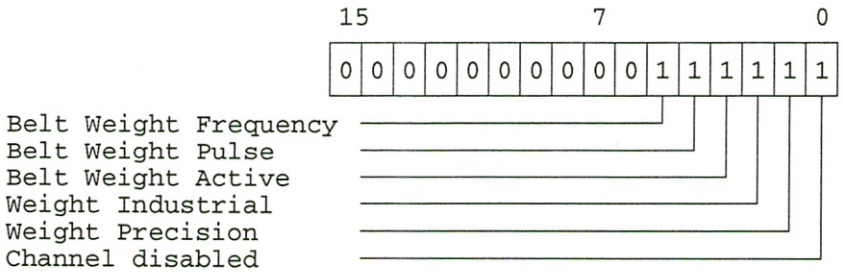
```
Record
    ChannelType: Word;                  (* Offset = 0 *)
    Exist: Array[0..15] Of Boolean;     (* Offset = 2 *)
    Functions: Array[0..15] Of Boolean; (* Offset = 4 *)
    SampleTime: Array[0..15] Of Boolean; (* Offset = 6 *)
End;
```

ChannelType = 10

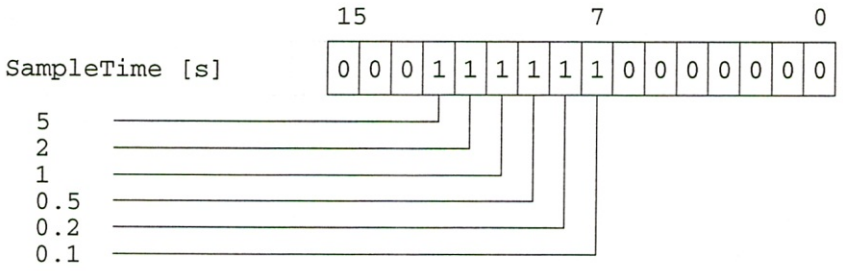
Exist =



Functions =



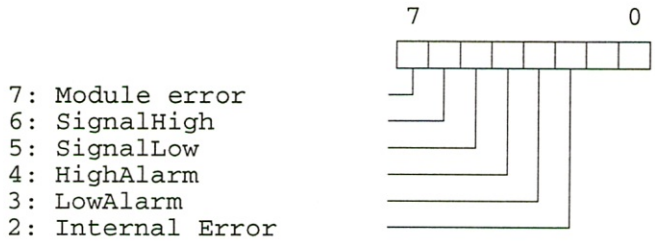
SampleTime:



SWNo 8F: CHError.

Record
His:Array[0..7] Of Boolean;
Act:Array[0..7] Of Boolean;
End;

The 8 bit i ChError.His and ChError.Act has 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 bit are insignificant, because a Module error can lead to random error codes on individual channels (also see "Service channel").

Bit 6 SignalHigh indicates an internal error. It is enabled by setting Ch-Config.Enablebit[6] = TRUE.

Bit 5 SignalLow indicates an internal error. It is enabled by setting ChConfig.Enable-bit[5] = TRUE.

Bit 4 HighAlarm is set if Conductivity > HighLevel and ChConfig.Enablebit[4]= TRUE.

Bit 3 LowAlarm is set if Conductivity < LowLevel and ChConfig.Enablebit[3] = TRUE.

Bit 2 An internal error is indicated. If the module continues to indicate internal error after a reset, the module is likely to require repair.

Note: After an error has disappeared, data can be invalid for a period, depending on the setting of SampleTime and NoOfSamples in ChConfig.

4 Calculator channel (channel 9).

The PD 3270 is able to perform arithmetical and boolean functions by means of the calculator channel. All variables within the module can be used in the expressions.

The calculator has a real type accumulator, a boolean type accumulator, two channel pointers, two index registers and a bit index register. The instruction set includes Move, Compare, Jump, logical operations and arithmetical operations. Some of the instructions can operate either on variables via SWNo (channel no. and register no.) or on immediate values.

It is not possible to write into EEPROM by means of the Calculator programme.

The speed of a calculator programme in a module depends on the configuration of the individual channels and the amount of P-NET communication.

Variables on Calculator channel. Channel identifier: **Calculator**

SWNo	Identifier	Memory type	Read out	Type
90	UniversalA	RAM Auto Save	Decimal	Real
91	UniversalB	RAM Auto Save	Decimal	Real
92	UniversalC	RAM Init EEPROM	Decimal	Real
93	UniversalD	RAM Init EEPROM	Decimal	Real
94	UniversalE	RAM Init EEPROM	Decimal	Real
95	UniversalF	RAM Init EEPROM	Decimal	Real
96	UniversalG	RAM Init EEPROM	Decimal	Real
97	Universal	RAM Read Write	- - - - - -	Array20Real
98	UserTimer	RAM Read Write	Decimal	Real
99	RunEnable	RAM Init EEPROM	Binary	Boolean
9A	LookUp1	EEPROM RPW	- - - - - -	Array20Real
9B	LookUp2	EEPROM RPW	- - - - - -	Array10Real
9C	LookUp3	EEPROM RPW	- - - - - -	Array10Real
9D	ProgramStep	EEPROM RPW	- - - - - -	Array200Word
9E	ChType	PROM Read Only	- - - - - -	Record
9F	IOChError	RAM Read Only	Binary	Record

SWNo 90-96:UniversalA-UniversalG

These variables are universal variables of type real, used by the calculator for input/output values. The first 2 variables are automatically saved in EEPROM at a certain predetermined frequency.

SWNo 97: Universal
This variable is an array which can hold 20 real values. The structure of the Universal variable can be useful to the calculator programmer, for accessing variables within a program loop, by means of an index pointer.

SWNo 98: UserTimer
This register holds a timer variable, which can be used by the calculator program. The timer counts down with a resolution of 1/8 second. The count continues through negative values. The timer register is cleared after a power failure or reset. The maximum value for the timer is approximately 97 days. After an overflow, the timer continues from the maximum value.

SWNo 99: RunEnable
This variable is used to start and stop program execution in the calculator channel. When RunEnable is set to TRUE, the program always restarts from the first instruction line. RunEnable should be set to FALSE before a program is downloaded.

SWNo 9A: LookUp1
This variable is declared as a lookup table with the following format:

```
Coordinate: Record
                X:Real;
                Y:Real;
            End;

LookUp: Array[1..10] of Coordinate;
```

This variable represents a line through 10 pairs of x,y coordinates. LookUp1 performs a function that returns an interpolated Y value when called with an X value. The X coordinates must be in increasing order. For X-values below X1 the function will return the Y1 value and for X-values higher than X10, it returns Y10.

SWNo 9B-9C: LookUp2 - LookUp3
These variables are declared as lookup tables with the following format:

```
Coordinate: Record
                X:Real;
                Y:Real;
            End;

LookUp: Array[1..5] of Coordinate;
```

The variables represents each a line through 5 pairs of x,y coordinates. Each LookUp performs a function that returns an interpolated Y value when called with an X value. The X coordinates must be in increasing order. For X-values below X1 the function will return the Y1 value and for X-values higher than X5, it returns Y5.

SWNo 9D:ProgramStep

This variable is defined as ARRAY[1..200] OF WORD, and holds the calculator program as a number of instructions, operating on various variables. The total number of program steps in a calculator programme is 200.

The execution time for each instruction is approximately 1 ms. By disabling unused automatic functions or channels in the module, the operating speed can be maximized.

Some typical instruction execution times are shown below.

Instruction	**Typ. time**
Move CR2:6,Acc	0.4 ms
Move Acc,CR2:4	0.4 ms
Move 9,CR2	0.4 ms
Sub 123.4	0.8 ms
Add 123.4	0.8 ms
Div 123.4	1.0 ms
Mul 123.4	1.0 ms
LookUp CR2:#A	12.0 ms

Refer to the PD Calculator Assembler Manual (PD no. 502 061) for a list of the available instructions and information on how to programme the calculator.

SWNo 9E: ChType

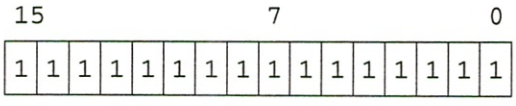
For the calculator channel, ChType is of the following type:

```
Record
    ChannelType: WORD;          (* Offset = 0 *)
    Exist: Array[0..15] Of Boolean;   (* Offset = 2 *)
    NoOfProgramStep: WORD;      (* Offset = 4 *)
end
```

ChType has the following value:

ChannelType = 7

Exist =



NoOfProgramStep = 200

SWNo 9F: ChError

No Error messages will automatically appear in the calculator channel, and therefore ChError normally reads 0. However, the calculator can be programmed to set any of the error bits, to simulate an error condition. This feature can be useful in indicating erroneous data. The register is declared as follows:

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

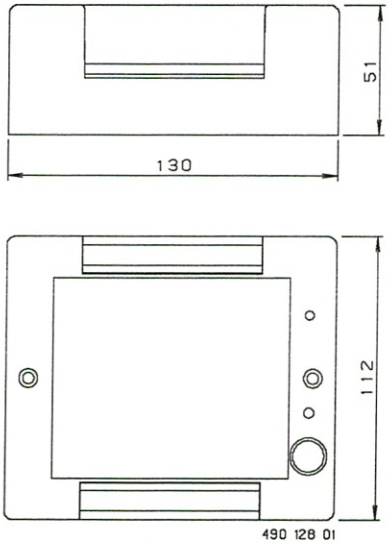
5 Construction, Mechanical.

The PD 3270 module is housed in a black plastic case.
The case measures W x H x D = 130.0 x 112.0 x 50.9 mm (tolerance to DIN 16901).

The module is designed for plugging directly on to a mounting rail (EN 50 022 / DIN 46277). The module incorporates two snap connectors, which provide the terminals for field connection, power and communications.

The module may be DIN rail mounted for a panel mounted configuration, and contained in a sealed box designed for the plant environment. It may be removed for service, without interfering with operational activities on the rest of the network.

Scale drawing (in mm):

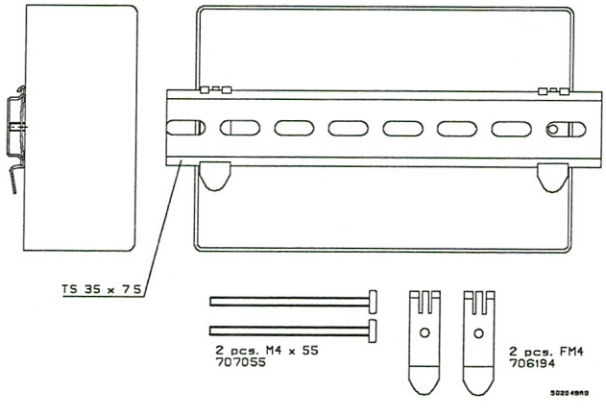


Materials:

Case:	Black NORYL GFN (injection moulded)
Front foil:	Polycarbonate.
Back plate:	Black anodized aluminium.

Weight: 400 gram.

Rail mounting:



6 Specifications.

All electrical characteristics are valid at an ambient temperature -25°C - 70°C, unless otherwise stated.

All specifications are respected in the approved EMI conditions. EMC test specifications for PD 3270 are available in a separate document, PD no. **506 020**.

6.1 Power supply.

Power supply DC:	nom.	24.0 V
	min.	20.0 V
	max.	28.0 V
Ripple:	max.	5%
Power consumption:	typ.	1.2 W
Current at power up:	max.	250 mA
Fuse: 1 A time delay		

6.2 Conductivity input.

Electrode supply, AC:	400 mV (pp)
Electrode frequency:	1200 Hz
Load resistance (each electrode):	0 - ∞ Ohm
Linearity error:	Nonlinear measurement
Measuring time (fully integrating) settings: SampleTime = 0.5 / 1 / 2 / 5 s	
Averaging function settings: NoOfSamples = 1 / 2 / ... / 32 Averaging time (SampleTime * NoOfSamples):	0.5 - 160 s
Accuracy:	± 0,5 % of fullscale

6.3 Ambient Temperature.

Operating temperature:	-25°C - 70°C
Storage temperature:	-40°C - 85°C

6.4 Humidity.

Relative humidity:	max.	95 %

6.5 Approvals.

Compliance with EMC-directive no.:	89/336/ECC
Generic standards for emission:	
Residential, commercial and light industry	EN 50081-1
Industry	PrEN 50081-2
Generic standards for immunity:	
Residential, commercial and light industry	EN 50082-1
Industry	PrEN 50082-2
Vibration (sinusoidal):	IEC 68-2-6 Test Fc

7 Survey of variables in the PD 3270 module.

SWNo	Service channel 0	Conductivity channel 8	Calculator 9
x0	NumberOfSWNo		UniversalA
x1	DeviceID		UniversalB
x2		Conductivity	UniversalC
x3	Reset		UniversalD
x4	PnetSerialNo	UserReal	UniversalE
x5		FlagReg	UniversalF
x6			UniversalG
x7	FreeRunTimer	HighLevel	Universal
x8	WDTimer	LowLevel	UserTimer
x9	ModuleConfig	ChConfig	RunEnable
xA	WDPreset		LookUp1
xB		FullScale	LookUp2
xC		ZeroPoint	LookUp3
xD	WriteEnable	Maintenance	ProgramStep
xE	ChType	ChType	ChType
xF	CommonError	ChError	ChError

490 153 01

Contents