

Profibus-DP Interface Board for Total Pressure Gauge Controller TPG300 and TPG500

IF300P

IF301P



Validity

This document applies to products with part numbers

IO441395	(IF300P, interface board with mechanical relays)
IO441396	(IF301P, interface board with PhotoMOS relays)


The part number (PN) can be taken from the product nameplate.

If not indicated otherwise in the legends, the illustrations in this document correspond to the interface board IF300P. They apply to the interface board IF301P by analogy.

Contents

Validity	2
1 Introduction	2
2 Hardware	3
2.1 Pin Assignment	3
3 Software	4
3.1 Overview	4
3.2 Profibus	4
3.2.1 Configuration	4
3.2.1.1 Bus Parameters	4
3.2.1.2 Slave Parameters	5
3.2.2 Data Exchange	5
3.2.2.1 Description of the Parameter Channel (Input and Output Data)	6
3.2.2.2 Process Data	8
3.2.3 Device Model	8
3.2.3.1 Device Block	8
3.2.3.2 Semantics	9
3.2.4 Function Blocks	10
3.2.4.1 Analog Input Function Blocks	10
3.2.4.2 Threshold Control Function Blocks	11
4 Example	13
4.1 Use of the Parameter Channel	13
5 Address Switches	14
Appendix	15
A: Further Information	15

1 Introduction

This document describes the functionality of the Profibus-DP (→  [1], [2]) interface board IF30xP, a new communication interface for the Total Pressure Gauge Controller TPG300 and TPG500. The communication interface allows to access process and parameter data of the TPG300 and TPG500. As a second line of communication, a set of five relays is used.

A copy of the GSD file can be downloaded from our website (www.inficon.com).

2 Hardware

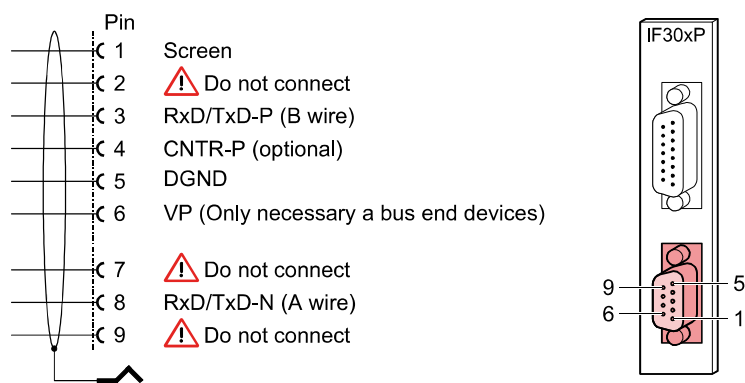
The IF30xP hardware is defined in two parts:

- Profibus interface providing the Profibus functionality and the parameter mapping between the TPG300 / TPG500 parameter protocol and Profibus.
- The analog part of the IF300 interface (relays and attached auxiliary components) is included in the design of the IF30xP. The connection between the TPG300 / TPG500 and the IF30xP is compatible with the existing RS232 interface IF300. On the rear panel, the Profibus signal is available through a D-sub 9 pin connector (female) while the relay signals can be accessed via a D-sub 15-pin connector (male).

2.1 Pin Assignment

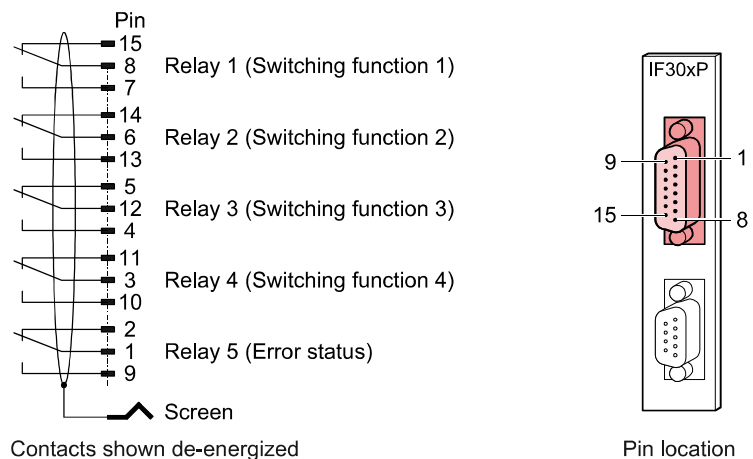
Profibus interface connector

The interface connector on the rear of the IF30xP has the following pin assignment:



Relay connector

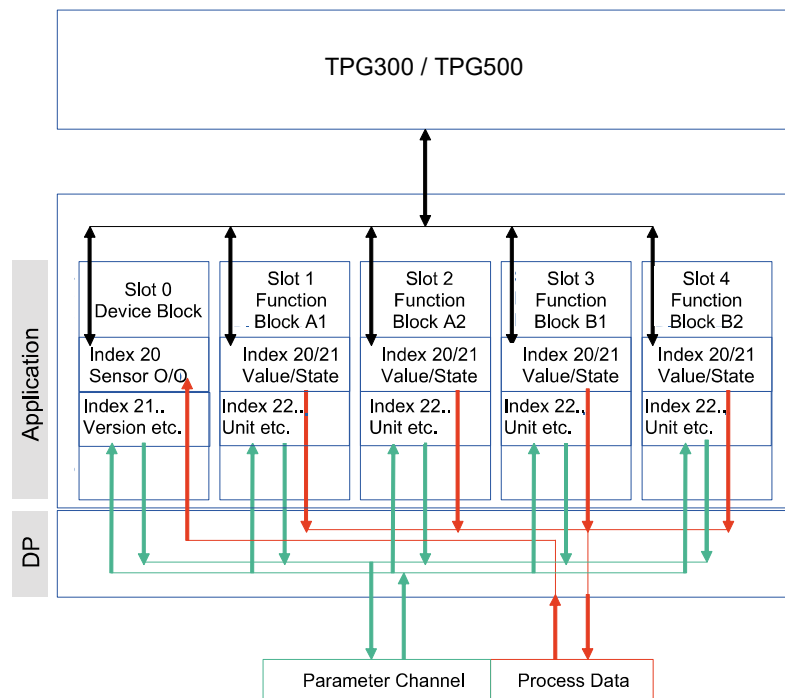
The relay connector on the rear of the IF30xP has the following pin assignment:



3 Software

3.1 Overview

The software provides a mapping between the functions of the TPG300 / TPG500 and Profibus. In this document, the software is called IF30xP application. The basic structure of the application is shown in the following illustration. The address space of the TPG300 / TPG500 is separated into different logical blocks. These blocks allow a Profibus-tailored representation of the functions of the TPG300 / TPG500. The device block contains the properties of the device as a whole, whereas the individual function blocks contain the measuring channel values and their properties.



The data have been divided into a process data and parameter data group. Process data are directly related to the measuring process. They include the measured values and control information by which the master device can determine the basic operation of the device. They must be transmitted continuously between the master device and the TPG300 / TPG500. The process data area of the Data Exchange Telegram is discussed in section "Process Data", § 8. The parameter data group contains all variables which are needed to change or inquire process data. These values usually form a much larger part of the device's address space, but they are only transmitted on request.

To allow access to process data via Profibus-DP/V0, an additional protocol is mapped into the cyclic data stream between the master device and TPG300 / TPG500. The area of the Data Exchange Telegram which contains this mapping is called parameter channel. It is described in section "Description of the Parameter Channel", § 6.

3.2 Profibus

This section describes the properties of the Profibus stack from the point of view of the application software of the IF30xP.

3.2.1 Configuration

3.2.1.1 Bus Parameters

Baud rates

The IF30xP supports all Baud rates defined in the Profibus Standard EN 50170 (→ [1], [2]). The Baud rate is detected by the IF30xP from the data traffic of the connected Profibus network.



For Profibus operation, the Baud rate of the TPG300 / TPG500 itself has to be set to 19200 Baud to achieve a cycle time of 100 ms (→ [3], [4]).

Station address

The station (device) address is set by the user via the dedicated 8 bit address switch on the interface. Additionally, if the address switch is set to 126, the station address may be set by a Profibus master (Set-Slave-Address). The address switches of the IF30xP board can only be adjusted while the board is **not** installed in the TPG300 / TPG500.

3.2.1.2 Slave Parameters

Ident number

The Ident number is assigned by the Profibus User Organization as 05D9_{hex} (→ [1], [2]).

Configuration data

The TPG300 / TPG500 must be classified as modular device in Profibus terms. The following module definitions apply:

Module	Config. Data
Parameter Channel Module (8 byte input, consistency over whole length)	0x97
Parameter Channel Module (8 byte output, consistency over whole length)	0xA7

Module	Config. Data
Channel Module Profibus floating point format (5 byte input, consistency over whole length)	0x94
Channel Module Inficon floating point format (4 byte input, consistency over whole length)	0x93
Device Module (1 byte output, consistency over whole length)	0xA0

A maximum of 5 modules can be chosen. The device module must always be included.

The run-time configuration of any TPG300 / TPG500 must be inquired by the application software before moving into the configuration phase.

Additionally, 2 modules describing the parameter channel must always be included by the user.

User parameters

A vendor specific parameter string is not supported.

3.2.2 Data Exchange

The Data Exchange mode is used for fast cyclic transmission of measuring and command data between the DP-Master and all configured slaves in the network. The following two illustrations contain the input and output data of the IF30xP. The data are divided into a parameter and a process data part. The process data contain the measuring and command data as described in section "Process Data", 8. The parameter data area contains a dedicated channel called parameter channel. It allows the transmission of cyclic requests by specifying an additional protocol within the input and output data area. A concise description can be found in section "Description of the Parameter Channel", 6.

Input data (master's point of view)

The Input Area (DxIn) as sent in the Data Exchange Telegram consists of the 8 byte parameter channel plus 20 byte of process data.

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	24	25	26	27	28
	Parameter channel								Process data												
	PKE	IND	res.															

Output data
(master's point of view)

The Output Area (DxOut) as received in the Data Exchange Telegram contains the 8 byte parameter channel plus one control byte in the process part of the telegram.

Byte	1	2	3	4	5	6	7	8	9
	Parameter channel								Process data
	PKE	IND	res.	PWE				PWE	

3.2.2.1 Description of the Parameter Channel (Input and Output Data)

The parameter channel was created as a means for transmitting acyclic requests within a cyclic data stream. This section provides a brief introduction into the matter. The following illustration shows the basic structure of the parameter channel. **This Parameter channel is called PKW interface.** The PKW interface consists of 8 bytes.

Octets

Byte	1	2	3	4	5	6	7	8
	PKE	IND	res.	PWE				

PKE = parameter signature value
 ≙ read or write command

IND = sub index
 ≙ Index No. of the index you want to read (see Block Model)

res. = reserved

PWE = process value
 ≙ value you want to write or read

The PKW interface allows a read and write access to the parameter space of the slave. Within the PKW mechanism, the master formulates an instruction. The slave processes the instruction and formulates the response.

Instructions and responses cannot be blocked: exactly one instruction is transmitted in one Output Telegram and exactly one response is transmitted in one Input Telegram. Therefore, exactly 4 bytes of user data can be transmitted within one instruction or one response.

The instructions and responses are coded in the parameter signature word (PKE):

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

As described, the PKE consists of 2 bytes:

- Bits 15 ... 12 are named **AK** ≙ instruction/response signature.
- Bits 11 ... 8 are reserved.
- Bits 7 ... 0 define the slot from which you want to read or in which you want to set values.

Instruction and response processing

The instruction and response signatures define which fields of the PKW interface (IND, PWE) have to be processed.

Instruction signature

In case of master ⇒ slave communication the field AK contains the instruction signature.

In case of slave ⇒ master communication the field AK contains the response signature.

Master ⇒ Slave

Instruction signature	Function
0	No instruction
1	Parameter value read
2	Parameter value write (word)
3	Parameter value write (double word)

Slave ⇒ Master (Response Signature)

Positive	Function: Response signature	Negative	Function: Response signature
0	No Response		
1	Parameter value transmitted (word)	7	Instruction not processable (error code)
2	Parameter value transmitted (double word)		
11	Parameter value transmitted (byte)		
1	Parameter value transmitted (word)	7	Instruction not processable (error code)
11	Parameter value transmitted (byte)		
2	Parameter value transmitted (double word)	7	Instruction not processable (error code)

The right column contains the responses to the corresponding instruction in normal case (positive) and in case of an error (negative).

The master transfers an instruction to a slave. It repeats the instruction until it receives a response from the slave. Only one instruction will be processed at a time.

A slave provides the response until the master formulates a new instruction. If the slave codes a parameter value (AK = 1), the response contains the actual value in case of repetitions (cyclic processing).

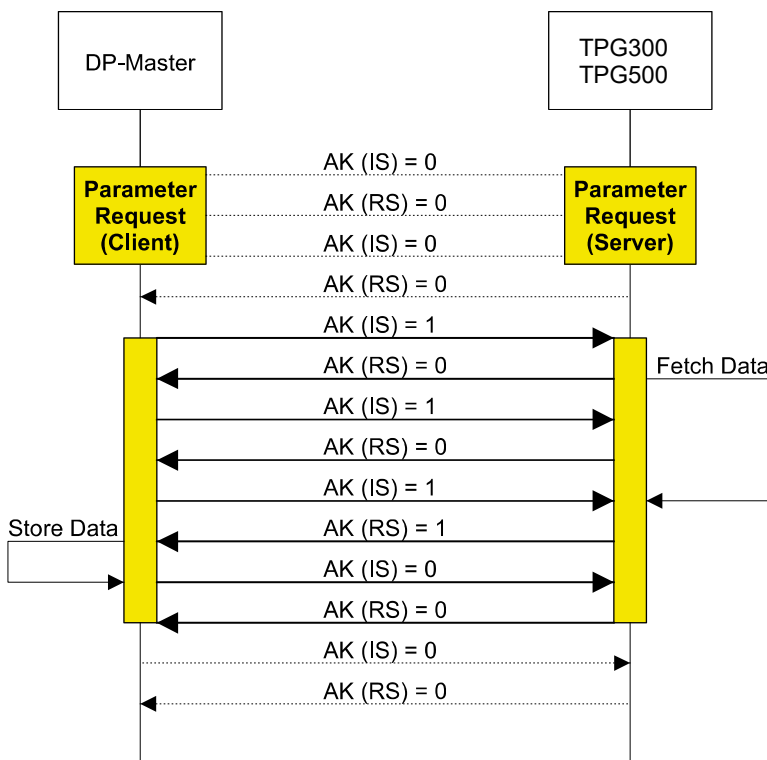
The PWE transfer of word length will be done with octet 7 and octet 8, the transfer of byte length will be done with octet 8, the transfer of double words will be done with octet 5 ... octet 8.

Error message

The slave sends an error message on octets 7 and 8 (interpreted as INT16) if a request can not be fulfilled.

Error code	Semantic
0	Undefined slot
1	Parameter not changeable
2	Lower or upper value range limit overflow
3	Index error
5	Data type error
17	Instruction not allowed in this state

The following sequence diagram is an example of a parameter request by the cyclic DP-Master illustrating the application of the parameter channel.



3.2.2.2 Process Data

The structure of the process data part of the input and output data area is presented in the following illustrations. The TPG300 / TPG500 can host up to four independent measuring channels, depending on the installed sensor boards (→ [3], [4]). Each channel is represented in the input data area by its value and a status byte. A more detailed description of these values can be found in section "Function Blocks", 10.

Input process data
(master's point of view)

Octets	9	10	11	12	13	24	25	26	27	28
	Measured value A1				State A1	Measured value B2				State B2

Output process data
(master's point of view)

Between the DP-Master and the TPG300 / TPG500, only one command byte is transmitted. Its content determines the state of each sensor of the device. The definition of its value range and semantics can be found in the following section "Device Block".

Octets	9
	Sensor On_Off

3.2.3 Device Model

3.2.3.1 Device Block

Device block

Block ID	Instance Number	Slot
0	0	0

Device block parameters

Block ID	Index	Name	Data type	Bytes	Access	Store	Value range	Initial value
0	20	Sensor On_Off	INT8	1	c_w, ac_r			-
0	21	Program version	INT32	1	ac_r			-
0	24	Authorized access code	INT16	2	ac_r, ac_w			-
0	25	Pressure under-range control	OSTRING1	1	ac_r, ac_w			-
0	26	Volatility	OSTRING1	1	ac_r, ac_w			-

ac_r = readable with parameter channel

ac_w = writable with parameter channel

c_w = writable with process data

c_r = readable with process data

3.2.3.2 Semantics

Sensor On_Off

This parameter switches the sensors on and off. The meanings of the bits are listed in the following table.

Bit								Meaning	
7	6	5	4	3	2	1	0		
x	x	x	x	x	x	0	0	No action	Channel A1
x	x	x	x	x	x	0	1	Switch off	
x	x	x	x	x	x	1	0	Automatic	
x	x	x	x	x	x	1	1	Switch on	
x	x	x	x	0	0	x	x	No action	Channel A2
x	x	x	x	0	1	x	x	Switch off	
x	x	x	x	1	0	x	x	Automatic	
x	x	x	x	1	1	x	x	Switch on	
x	x	0	0	x	x	x	x	No action	Channel B1
x	x	0	1	x	x	x	x	Switch off	
x	x	1	0	x	x	x	x	Automatic	
x	x	1	1	x	x	x	x	Switch on	
0	0	x	x	x	x	x	x	No action	Channel B2
0	1	x	x	x	x	x	x	Switch off	
1	0	x	x	x	x	x	x	Automatic	
1	1	x	x	x	x	x	x	Switch on	

Program version

The program version of the TPG300 / TPG500 itself can be read with this attribute. The following coding has been used:

Example: Software version BG509731-A

Via Profibus, you can read only the number 509731-A. The alphanumerical (in this example the A), representing the software revision, is coded to a number, where A $\hat{=}$ 1, B $\hat{=}$ 2, C $\hat{=}$ 3 and so on. Since only 4 bytes are available for data in the parameter channel, each number is coded in one nibble of these 4 bytes.

Software version	5	0	9	7	8	0	-	F
Byte No. of PWE	5		6		7		8	
Value of byte in hex	5	0	9	7	8	0		6
	upper nibble	lower nibble	upper nibble	lower nibble	upper nibble	lower nibble	upper nibble	lower nibble

Authorized access code

Value	Meaning
000	TPG can be operated without a code.
xyxy ¹⁾	TPG can be operated when corresponding code is entered.
9919	TPG can be operated when corresponding code is entered. Code can not be modified.

¹⁾ xx and yy represent "mantissa" and "exponent" values as displayed on the TPG300 / TPG500. Any number is permissible, except "000" and "9919", the "exponent" can have one or two digits (→ [3], [4] for TPG300 / TPG500 operating manuals).

Pressure underrange control

This function is only available for trigger relays, which are assigned to a Cold Cathode Ion Gauge channel.

Value	Semantics of value
0	Underrange is regarded as a valid measured value. The trigger relay remains on.
1	If, after power on, the measured value is within the measuring range for more than 10 sec., the trigger relay may be activated depending on the threshold setting. If the pressure then drops below the underrange limit, the trigger relay remains on. If, after power on, the measured value is within the measuring range for less than 10 sec., the trigger relay is not activated.

Volatility

This parameter controls the volatility of the parameter set written via the parameter channel. By default, these parameters are stored in volatile memory by the TPG300 / TPG500. The parameter controls the non-volatile storage in the TPG300 / TPG500.

Value	Meaning
0	Ready to store
1	Storage requested by user
2	Storage and hotstart requested by user

3.2.4 Function Blocks

3.2.4.1 Analog Input Function Blocks

The device TPG300 / TPG500 offers up to four different measuring channels (depending on the boards used). These channels are considered to be independent functional units called function blocks.

Channel	Block ID	Instance number	Slot
A1	1	0	1
A2	1	1	2
B1	1	2	3
B2	1	3	4

The slot number combined with the index identifies a unique parameter transmitted via the parameter channel.

Block ID	Index	Name	Data type	Bytes	Access	Store	Value range	Initial value
1	20	Measured value	Float or Mantissa and Exponent	4 or 2 and 1	c_r		–	–
1	21	Measurement status	UINT8	1	c_r		–	–
1	22	Pressure unit	UINT8	1	ac_r, ac_w		1 (mbar) 2 (Torr) 3 (Pascal)	–
1	23	Filter time constant	OSTRING1	1	ac_r, ac_w			–

Measurement status

Value	Meaning
0	Measurement data ok
1	Underrange
2	Overrange
3	Error
4	Off
5	No hardware

Pressure unit

The pressure unit parameter is defined in each function block individually. However, when the pressure unit is modified in one function block, the change of the parameter value is automatically taken over by all other function blocks.

Value	Meaning
1	mbar
2	Torr
3	Pascal

Filter time constant

Value bit 1 and 0	Semantics of data
01 bin	Fast
10 bin	Medium
11 bin	Slow

3.2.4.2 Threshold Control Function Blocks

The device TPG300 / TPG500 supplies up to 6 different threshold control channels (depending on the boards used). These channels are considered to be independent functional units called control function blocks.

Control	Block ID	Instance number	Slot
Relay 1	2	0	5
Relay 2	2	1	6
Relay 3	2	2	7
Relay 4	2	3	8
Cold cathode A	2	4	9
Cold cathode B	2	5	10

The slot number combined with the index identifies a single parameter transmitted via the parameter channel.

Block ID	Index	Name	Data type	Bytes	Access	Store	Value range	Initial value
2	20	Threshold validation	INT8	1	ac_r, ac_w			1
2	21	Upper threshold	Float	1	ac_r, ac_w			–
2	22	Lower threshold	Float	1	ac_r, ac_w			–
2	23	Threshold source	UINT8	1	ac_r, ac_w			–
2	24	Setpoint status	UINT8	1	ac_r			–

Threshold validation

This parameter assures the consistency of the threshold parameters (index 21 to 23). If it is set to "0", the threshold parameters can be changed by the Profibus master device user. Otherwise, these parameters can only be read.

Value	Meaning
0	Invalid threshold information – Threshold parameters may be changed and therefore, consistency is not guaranteed.
1	Valid threshold information

To change a threshold value, the following procedure has to be carried out:

- 1 Set index 20 to "0".
- 2 Index 21 ... 23 may be changed.
- 3 Set index 20 to "1".

Switching index 20 from "0" to "1" causes the transmission of the new threshold value set to the TPG300 / TPG500.

Threshold source

The parameter contains the slot number of the analog input function block which controls the relay.

Setpoint status

The setpoint status indicates whether the threshold control was triggered or not.

Value	Meaning
0 _{bin}	Switching function activated
1 _{bin}	Switching function deactivated

Self monitoring for cold cathode gauges

For cold cathode gauges a self monitoring function is implemented which switches off the cold cathode at a user defined pressure (→ [3]).

To activate this function, use a normal setpoint function and set the threshold source to the corresponding cold cathode channel number.

Example

Self monitoring of cold cathode B1 (channel B1, Slot 3) by setting the parameters of Threshold Control Function Block "Cold Cathode B" (Slot 10).

- 1 Set threshold validation to "0":

Byte	1	2	3	4	5	6	7	8	9
	Parameter channel								Process data
	PKE	IND	res.	PWE				PWE	
	20	0A	14	00	00	00	00	00	xx

- 2 Set "Upper Threshold" to the desired value:

Byte	1	2	3	4	5	6	7	8	9
	Parameter channel								Process data
	PKE	IND	res.	PWE				PWE	
	30	0A	15	00	59	03	12	6F	xx

- 3 Set "Lower Threshold" to the desired value:

Byte	1	2	3	4	5	6	7	8	9
	Parameter channel								Process data
	PKE	IND	res.	PWE				PWE	
	30	0A	16	00	3a	83	12	6F	xx

- 4 Set "Threshold Source" to the cold cathode channel B1:

Byte	1	2	3	4	5	6	7	8	9
	Parameter channel								Process data
	PKE	IND	res.	PWE				PWE	
	20	0A	17	00	00	00	00	03	xx

5 Set threshold validation to "1":

Byte	1	2	3	4	5	6	7	8	9
	Parameter channel								Process data
	PKE		IND	res.	PWE			PWE	
	20	0A	14	00	00	00	00	01	xx

4 Example

4.1 Use of the Parameter Channel

The following example shows how to program the parameter channel if resetting the pressure unit to mbar is required.

As shown in section "Threshold Control Function Blocks", § 11, you find the pressure unit in each Analog Input Function Block A1, A2, B1, B2 at index 22.

Function Block A1 is mapped to slot 1. The data type of the pressure unit is UINT8 and the value for Torr is "2".

Thus, you have to write value "2" to index "22" of slot "1".

Octets

yte	1	2	3	4	5	6	7	8
	PKE		IND	res.	PWE			

PKE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				res.				Slot							
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
2 _{hex}				0 _{hex}				0 _{hex}				1 _{hex}			

AK = 2 (parameter value write [word or byte])
 res. = 0
 Slot = 1

IND Index = index 22 = pressure unit 22_{dec} ≅ 16_{hex}

PWE The value for Torr is 02_{dec}. The process value consist of 4 bytes. The transfer of a byte is done with byte No. 8. All other bytes have to be set to 0. Thus, the four bytes of the PWE have the values 0, 0, 0, 2.

Complete instruction string sent from master to slave

Note: Do not forget to add the ninth byte (process data) to the output string.

Octets	1	2	3	4	5	6	7	8	9
Value (hex)	20	01	16	00	00	00	00	02	xx
	PKE		IND	res.	PWE			Process value	
	Parameter Channel								Process data

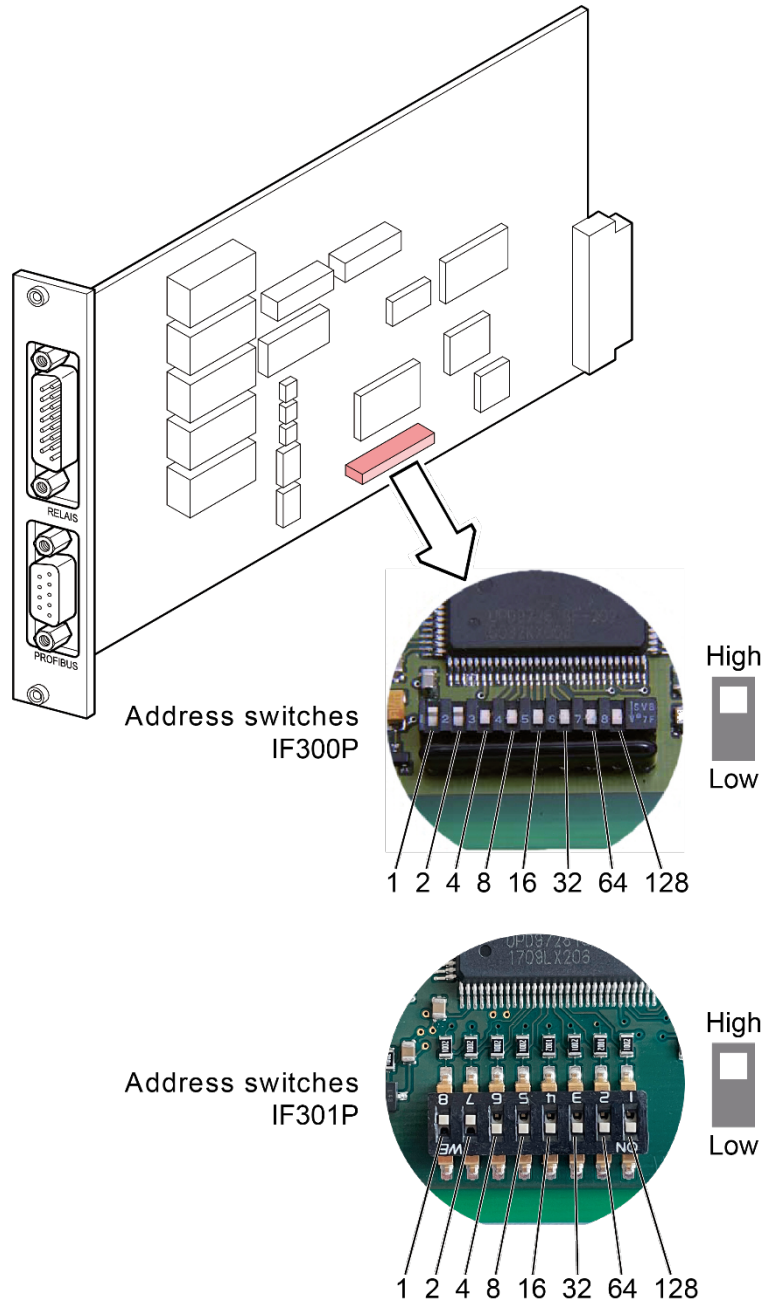
The slave responds to the request above with the following input data string:

Complete instruction string sent from master to slave

Octets	1	2	3	4	5	6	7	8	9	10
Value (hex)	B0	01	16	00	00	00	00	02	xx	xx	xx	xx
	PKE		IND	res.	PWE			Process data Byte 9 ... 28				
	Parameter channel								Process channel			





5 Address Switches

The following illustration shows the location of the address switches:



Appendix

A: Further Information

-  [1] www.profibus.com
Profibus user organization
-  [2] European Standard for Profibus EN 50170
-  [3] Operating Manual
Total Pressure Gauge Controller TPG300
IG9970BEN
INFICON AG, LI-9496 Balzers, Liechtenstein
-  [4] Operating Manual
Total Pressure Gauge Controller TPG500
IG6008BEN
INFICON AG, LI-9496 Balzers, Liechtenstein

Original: English



i 9973ben/ a



*LI-9496 Balzers
Liechtenstein
Tel +423 / 388 3111
Fax +423 / 388 3700
reachus@inficon.com*

www.inficon.com