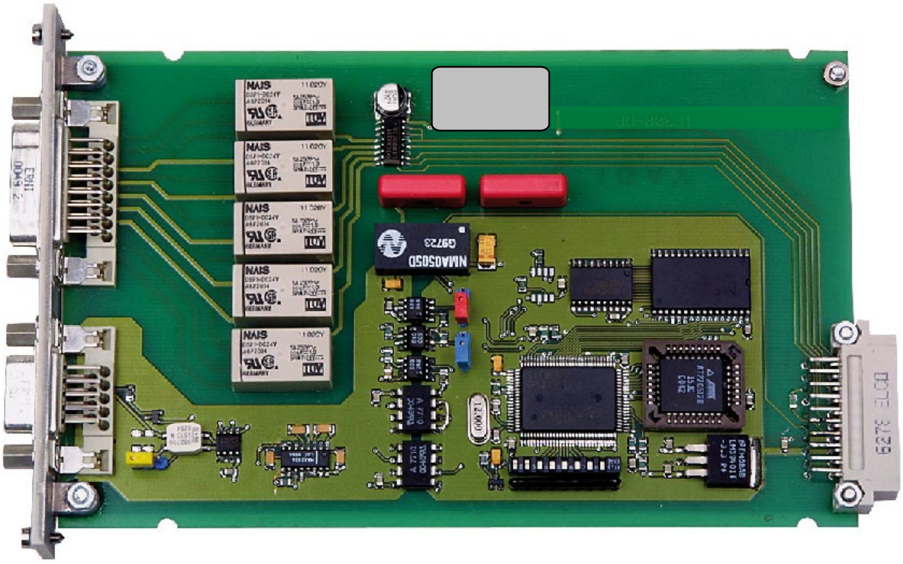


# COMMUNICATION PROTOCOL

EN

Original

# IF 400 P

Profibus-DP Interface Board for Ionization Gauge Controller IMG 400

BG 5530 BEN / A (2019-07)

PFEIFFER VACUUM

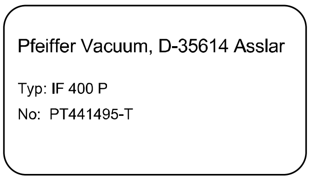
## About this Document

This document describes the functionality and programming of the Profibus interface IMG 400, designed to be used with the Ionisation Gauge Controller IMG 400.

For safety information and technical data of the controller, please refer to the IMG 400 operating manual (→ 🕮 [3]).

## Product Identification

In all communications with Pfeiffer Vacuum, please specify the information on the product nameplate (position of product nameplate → illustration on 🗎 41).




## Validity

This document applies to products with part number

PT 441 495 -T

The part number (No:) can be taken from the product nameplate.

## Trademark

SEMI® Semiconductor Equipment and Materials International, California

# Contents



For cross-references within this document, the symbol (→ 🗎 XY) is used; for cross-references to further documents listed under "Literature", use is made of the symbol (→ 🕮 [Z]).

# 1 Introduction

The communication interface IF 400 P allows to access process and parameter data of the IMG 400 via the Profibus-DP. As a second line of communication, a set of five relays is made available on the board.

Profibus-DP definitions and further information → 🕮 [1], [2].

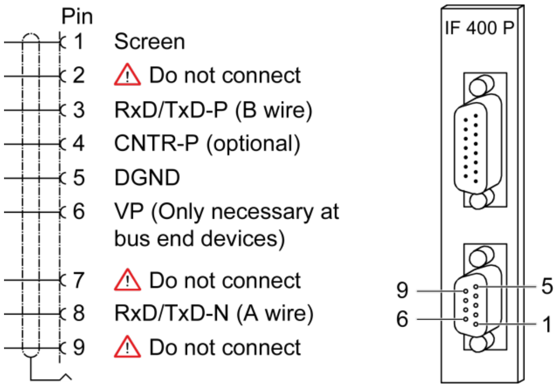
# 2 Hardware, Pin Assignment

The IF 400 P hardware is defined in two parts.

- Profibus interface providing the Profibus functionality and the parameter mapping between the IMG 400 parameter protocol and Profibus.
- Relays section (5 on board relays) and attached auxiliary circuitry is included in the design of the IF 400 P.

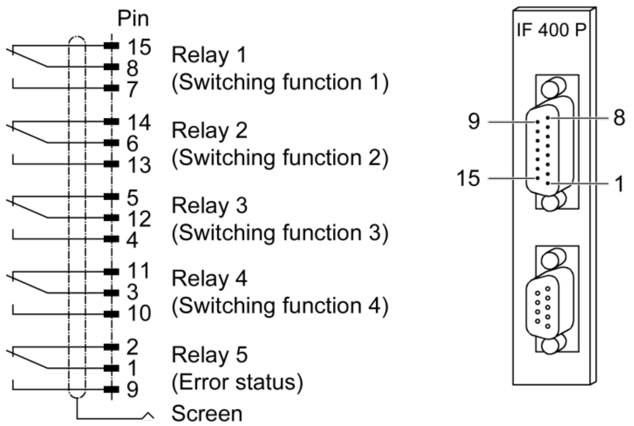
Profibus interface connector

The interface connector on the rear of the IF 400 P has the following pin assignment:



Relay connector

The relay connector on the rear of the IF 400 P has the following pin assignment:




Contacts shown de-energized

# 3 Installation

The IF 400 P interface option is installed and configured by the end-user.

## 3.1 Installing the IF 400 P Interface Board

**Skilled personnel**

All work described in this document may only be carried out by persons who have suitable technical training and the necessary experience or who have been instructed by the end-user of the product.

**WARNING**

Static electricity.

Static electricity can damage electronic components.

Do not touch the electronic components unless you have taken the appropriate protective measures. Store product in its antistatic bag.

Required tool

Screwdriver (size 2)

Procedure

❶ Switch off the IMG 400.

Wait one minute.

❷ Unscrew and remove cover plate "L" on the rear of the IMG 400 (→ 🕮 [3]).

❸ Set the device address on the IF 400 P board (→ 🗎 8).

❹ Insert IF 400 P board.

❺ Thighten screws on the rear panel of the board.

❻ Connect Profibus cable (and relays cable, if applicable) and secure locking screws of the connector.

☞ For correct operation, ensure that the screws of the plug-in modules are tightened.

## 3.2 Preparing the IMG 400 for Profibus Operation

For Profibus operation, the serial interface inside the IMG 400 has to be configured to communicate with the IF 400 P option. The following interface parameter settings are required in connection with the IF 400 P:

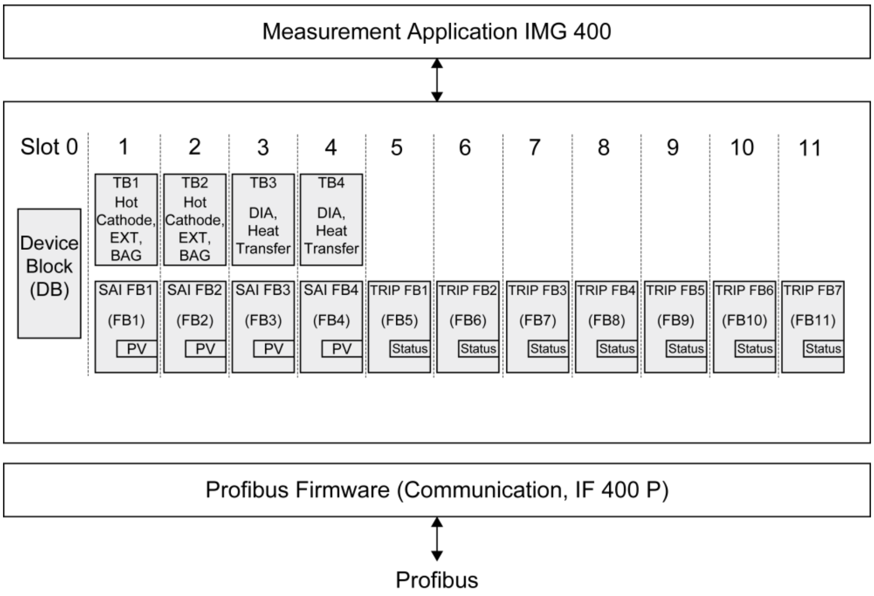
| Parameter | Value |
|---|---|
| Number of data bits | 8 |
| Parity bit | none |
| Stop bit | 1 |
| Data transmission rate | 9600 Baud |

The appropriate programming procedure is described in 📖 [3].

# 4 Software

## 4.1 Device Model

The diagram shown below exhibits the general structure of the implementation.




| Where: | | | | |
|---|---|---|---|---|
| | TBx | = | Transducer Block x | → 🗎 26, 28, 29 |
| | DB | = | Device Block | → 🗎 16 |
| | Hot Cathode | = | Hot Cathode ionization gauge | → 🕮 [3] |
| | EXT | = | Extractor gauge | → 🕮 [3] |
| | BAG | = | Bayard-Alpert gauge | → 🕮 [3] |
| | DIA | = | Capacitance Diaphragm gauge | → 🕮 [3] |
| | Heat Transfer | = | Heat Transfer gauge (Pirani) | → 🕮 [3] |
| | SAI FBx | = | Sensor Analog Input Function Block x | → 🗎 21 |
| | PV | = | Process Value | → 🗎 21 |
| | Trip FBx | = | Trip Function Block x | → 🗎 24 |

Measurement channels

The four measurement channels (input channels) of the IMG 400 can be connected to different types of gauges simultaneously, however, not every type can be connected to any channel (→ Appendix A).

☞ As the type of gauge is only known at the start-up time of the software the actual type of each transducer block is determined at that time.

The four measurement channels (input channels) are mapped into the respective transducer and function blocks.

Switching functions (Trip point functions)

A total of seven switching functions / Trip points with associated relays are available.

- 2 Relays located on the main board of the IMG 400 (Relays 1 and 2, → 🕮 [3]
- 5 Relays located on the IF 400 P board (Relay 3 … 7, → 🗎 5)

Relays are represented by one trip point function block each.

# 4.2 General Device Data

## 4.2.1 Data Transmission Rate

The device supports all data transmission rates (up to 12 MBaud) specified in IEC 61158 / IEC 61784 (→ 🕮 [2]). The data transmission rate used by the Profibus master is automatically detected by the device.

## 4.2.2 Device Address

For an unambiguous identification of the device on the bus, a device address is required.

The device address is set by the user via the dedicated 8 bit address switches located on the IF 400 P board (→ Appendix F).

Additionally, if the address switch is set to 126Dec, the device address may be set by a Profibus master (Set-Slave-Address). The address switches of the IF 400 P board can only be adjusted while the board is not installed in the IMG 400.

## 4.2.3 Ident Number

The Ident Number assigned to the IMG 400 by the PNO (→ 🕮 [1]).

| Device | Ident Number (hex) |
|---|---|
| IMG 400 | 0x0947 |

## 4.2.4 Configuration Data

Depending on the standard telegrams used (→ 🗎 13), the following configuration data have to be transmitted to the device during the configuration phase:

| Configuration | Standard telegram Master ⇒ Slave | Standard telegram Slave ⇒ Master | Configuration data (hex) |
|---|---|---|---|
| 1 | – | 5 | 0x45, 0x88, 0x05, 0x03, 0x03, 0x03, 0x03 |
| 2 | – | 6 | 0x45, 0x90, 0x05, 0x08, 0x08, 0x08, 0x08 |
| 3 | 1 | 5 | 0xC7, 0x81, 0x88, 0x05, 0x05, 0x05, 0x03, 0x03, 0x03, 0x03 |
| 4 | 1 | 6 | 0xC7, 0x81, 0x90, 0x05, 0x05, 0x05, 0x08, 0x08, 0x08, 0x08 |
| 5 | 2 | 11 | 0xC7, 0x87, 0x90, 0x0A, 0x0A, 0x05, 0x03, 0x03, 0x03, 0x03 |
| 6 | 2 | 12 | 0xC7, 0x87, 0x98, 0x0A, 0x0A, 0x05, 0x08, 0x08, 0x08, 0x08 |
| 7 | 3 | 11 | 0xC9, 0x89, 0x90, 0x0A, 0x05, 0x05, 0x0A, 0x05, 0x03, 0x03, 0x03, 0x03 |
| 8 | 3 | 12 | 0xC9, 0x89, 0x98, 0x0A, 0x05, 0x05, 0x0A, 0x05, 0x08, 0x08, 0x08, 0x08 |
| 9 | – | 201 | 0x47, 0x91, 0x05, 0x03, 0x03, 0x03, 0x03, 0x0A, 0x0A |
| 10 | – | 202 | 0x47, 0x99, 0x05, 0x08, 0x08, 0x08, 0x08, 0x0A, 0x0A |
| 11 | 1 | 201 | 0xC9, 0x81, 0x91, 0x05, 0x05, 0x05, 0x03, 0x03, 0x03, 0x03, 0x0A, 0x0A |
| 12 | 1 | 202 | 0xC9, 0x81, 0x99, 0x05, 0x05, 0x05, 0x08, 0x08, 0x08, 0x08, 0x0A, 0x0A |
| 13 | 2 | 203 | 0xC9, 0x87, 0x99, 0x0A, 0x0A, 0x05, 0x03, 0x03, 0x03, 0x03, 0x0A, 0x0A |
| 14 | 2 | 204 | 0xC9, 0x87, 0xA1, 0x0A, 0x0A, 0x05, 0x08, 0x08, 0x08, 0x08, 0x0A, 0x0A |
| 15 | 3 | 203 | 0xCB, 0x89, 0x99, 0x0A, 0x05, 0x05, 0x0A, 0x05, 0x03, 0x03, 0x03, 0x03, 0x0A, 0x0A |
| 16 | 3 | 204 | 0xCB, 0x89, 0xA1, 0x0A, 0x05, 0x05, 0x0A, 0x05, 0x08, 0x08, 0x08, 0x08, 0x0A, 0x0A |

## 4.2.5 User Parameter

The user data part of the parameter string contains a data unit information. The unit is coded as described on 🗎 22 (ID 22).

## 4.3 Data Exchange Mode

### 4.3.1 Data Transmission

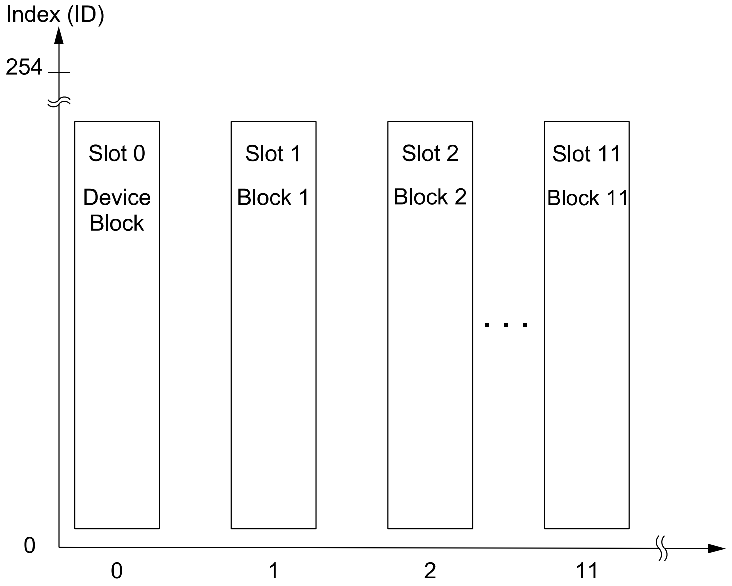
The reading and writing operations defined in the Profibus are based on a slot index address scheme. In the IMG 400, all device functions are organized in the following blocks:

- A device block describing all organizational parameters of the device (serial number, manufacturer, software version, status, exceptions …)
- An Analog Sensor Function Block describing the function of the pressure presentation
- An Analog Sensor Transducer Block describing the physical interface between the IMG 400 and the process (emission current …)
- Trip Function Blocks describing all relevant parameters, status and functionality of switching functions (setpoints …).

Each block is assigned to a separate slot. The exact assignment Block ⇒ Slot ⇒ Index is described in section "Block Model" (→ 🗎 14).

The Device Block is assigned to Slot 0, the Analog Transducer Function Blocks to Slot 1 … 4 and the Trip Point Function Blocks to Slot 5 … 11.

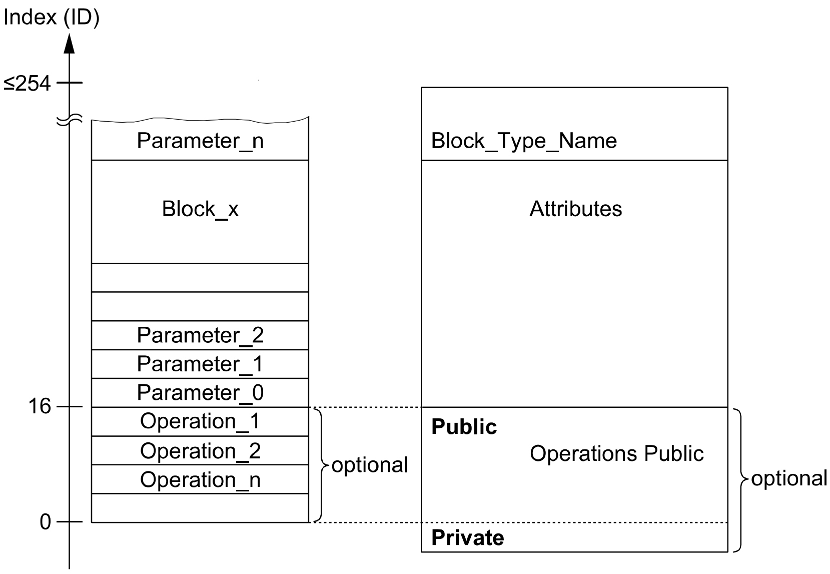
Block, slot and index assignment




There are 254 indices per slot. The indices can have a width of 255 bytes. All values that can be accessed via Profibus have to be mirrored to one of these slots / indices.

The parameters are generally numbered in ascending order, starting with index 16. Services such as "Degas On" or "Full scale" are numbered in descending order, starting with index 15.

Assignment of the block elements to the slot indices




## 4.3.2 Structure of the Cyclic Data Telegrams in Data Exchange Mode

In Data Exchange mode, the DP master class 1 cyclically transmits data from and to all slaves that are connected to the bus.

In this document, data transmitted from the slave to the master are called "input data" and data transmitted from the master to the slave are called "output data".

The input and output data of the IMG 400 have two logic parts:

1) the parameter channel
2) the process data channel

There are a number of standard telegrams, consisting of:

a) the parameter channel only
b) the process data channel only
c) both, the parameter and process data channel

The parameter channels allow masters to access device specific parameters that are not part of the normal cyclic data telegram.

**Input data**

The input data (transmitted by the IMG 400) consist of the 8 bytes of the parameter channel (if there is a parameter channel in the standard telegram) and of 9 … 26 bytes of process data depending on the selected standard telegram.

| Low | | | | | | | | | | | | | | High |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte | | | | | | | | Byte | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Parameter channel | | | | | | | | Process data and status | | | | | | |
| PKE | | ID | res. | PWE | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| Where: | PKE | = | Parameter Signature Value | Reading or writing command and definition of the slot |
| | ID | = | Index | Index to be read (→ "Block Model") |
| | res. | = | reserved | |
| | PWE | = | Process Value | Value to be read or written |

**Output data**

The output data (transmitted by the master) consist of 8 bytes of the parameter channel or, if there is no parameter channel in the standard telegram, of 0 bytes.

| Low | | | | | | | High |
|---|---|---|---|---|---|---|---|
| Byte | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Parameter channel | | | | | | | |
| PKE | | ID | res. | PWE | | | |

## Parameter Channel

The structure of the parameter channel is described in the table below.

The parameter channel (called PKW Interface hereinafter) consists of 8 bytes.

| Low | | | | | | | High |
|---|---|---|---|---|---|---|---|
| Byte | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| PKE | | ID | res. | PWE | | | |

The PKW Interface allows reading and writing of slave parameters with a maximum data length of 4 bytes. Strings cannot be read.

The slave generates exactly one response per instruction transmitted by the master. The instruction and response cannot be blocked. This means that exactly one instruction per output telegram can be transmitted to the slave and that exactly one response per input telegram can be transmitted to the master. 4 bytes of actual data can thus be transmitted at a time.

**PKE Parameter, Signature Value**

| High | | | | | | | | | | | | | | | Low |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AK | | | | res. | | | | Slot | | | | | | | |

Where:

| Bits | Meaning |
|---|---|
| 15 … 12 | AK = Instruction/response signature |
| 11 … 8 | Reserved |
| 7 … 0 | Define the slot from which data are read or onto which a value is to be written |

**Instruction signature**

In Master ⇒ Slave communication, the AK field contains the instruction signature of the master.

In Slave ⇒ Master communication, the AK field contains the instruction signature of the slave.

| AK | Function<br>Master ⇒ Slave<br>(Instruction signature) | AK normal | Function<br>Slave ⇒ Master<br>(Response signature) | AK error |
|---|---|---|---|---|
| 0 | No instruction | 0 | No response | |
| 1 | Read parameter value | 1<br><br>2<br><br>11 | Transmit parameter value (word)<br>Transmit parameter value (double word)<br>Transmit parameter value (byte) | 7 [1)] |
| 2 | Write parameter value (data type: word) | 1 | Transmit parameter value (word) | 7 [1)] |
| 3 | Write parameter value (data type: double word) | 2 | Transmit parameter value (double word) | 7 [1)] |
| 10 | Write parameter value (data type: byte) | 11 | Transmit parameter value (byte) | 7 [1)] |

[1)] Instruction cannot be executed (error code)

On the left of the table, the instruction signatures of the master are listed according to their function. On the right of the table, the corresponding normal responses (AK Normal) and error codes (AK Error) transmitted by the slave are listed.

**Instruction – response sequence**

1) The master transmits an instruction to the slave and repeats that instruction until it receives a response from the slave.
2) The slave keeps transmitting the response to the instruction until the master transmits a new instruction.
3) The master marks the end of the first instruction cycle by setting AK to zero. Only after that, a new instruction/response cycle may be started.

**PWE Parameter, Process Value**

The PWE represents the data element to be transmitted.

If a byte is to be transmitted, that byte has to be in position 8 of the parameter channel.

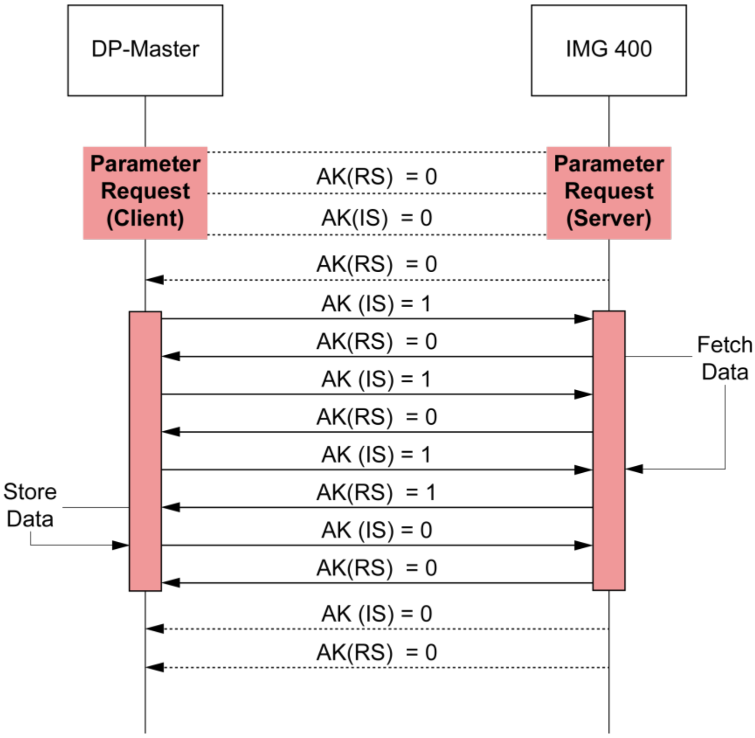
Integers are transmitted with bytes 7 and 8. Double integer and float values are transmitted with bytes 5 … 8.

**Error Code (Error Message)**

In the event of a transmission error (AK response signature = 7), the slave transmits an error code in byte positions 7 and 8 (data type: INT16).

| Error code | Meaning (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 |
| 18 | Other errors |
| 201 | Already in requested state |
| 202 | Object state conflict |

The following diagram shows an example of a data request from a master to an IMG 400 via parameter channel.




### 4.3.3 Cyclic Message Telegrams

The message telegrams listed below are implemented in the IMG 400. They can be selected according to requirements. When selecting a message telegram, ascertain what output format of the measured value (integer/float) is required and whether a parameter channel is needed or not. The IMG 400 can also be operated in such a way that the master does not transmit any output data to the slave.

Standard Telegrams

| Standard Telegram | Master ⇔ Slave | Octets | Meaning | Data origin |
|---|---|---|---|---|
| 1 | M ⇒ S | 0 | Transition Command | TBx (ID 14) [4)] |
| | | 1 | Transition Command Value for Hot Ion Gauges | TBx (ID 14) [4)] |
| 2 | M ⇒ S | 0 … 7 | Parameter Channel | |
| 3 | M ⇒ S | 0 … 7 | Parameter Channel | |
| | | 8 | Transition Command | TBx (ID 14) [4)] |
| | | 9 | Transition Command Value for Hot Ion Gauges | TBx (ID 14) [4)] |
| 5 | S ⇒ M | 0 | Exception Status | DB (ID 26) [1)] |
| | | 1 … 8 | Multi Process Value | FB (ID 19) [2)] |
| 6 | S ⇒ M | 0 | Exception Status | DB (ID 26) [1)] |
| | | 1 … 16 | Multi Process Value | FB (ID 19) [3)] |
| 11 | S ⇒ M | 0 … 7 | Parameter Channel | |
| | | 8 | Exception Status | DB (ID 26) [1)] |
| | | 9 … 16 | Multi Process Value | FB (ID 19) [2)] |
| 12 | S ⇒ M | 0 … 7 | Parameter Channel | |
| | | 8 | Exception Status | DB (ID 26) [1)] |
| | | 9 … 24 | Multi Process Value | FB (ID 19) [3)] |

Proprietary Telegrams

| Proprietary Telegram | Master ⇔ Slave | Octets | Meaning | Data origin |
|---|---|---|---|---|
| 201 | S ⇒ M | 0 | Exception Status | DB (ID 26) [1)] |
| | | 1 … 8 | Multi Process Value | FB (ID 19) [2)] |
| | | 9 … 10 | Channel Status | DB (ID 204) [1)] |
| | | 11 … 17 | Trip Relays 1 … 7 | FB (ID 21) → table 4, 🗎 24 |
| 202 | S ⇒ M | 0 | Exception Status | DB (ID 26) [1)] |
| | | 1 … 16 | Multi Process Value | FB (ID 19) [3)] |
| | | 17 … 18 | Channel Status | DB (ID 204) [1)] |
| | | 19 … 25 | Trip Relays 1 … 7 | FB (ID 21) → table 4, 🗎 24 |
| 203 | S ⇒ M | 0 … 7 | Parameter Channel | |
| | | 8 | Exception Status | DB (ID 26) [1)] |
| | | 9 … 16 | Multi Process Value | FB (ID 19) [2)] |
| | | 17 … 18 | Channel Status | DB (ID 204) [1)] |
| | | 19 … 25 | Trip Relays 1 … 7 | FB (ID 21) → table 4, 🗎 24 |
| 204 | S ⇒ M | 0 … 7 | Parameter Channel | |
| | | 8 | Exception Status | DB (ID 26) [1)] |
| | | 9 … 24 | Multi Process Value | FB (ID 19) [3)] |
| | | 25 … 26 | Channel Status | DB (ID 204) [1)] |
| | | 27 … 33 | Trip Relays 1 … 7 | FB (ID 21) → table 4, 🗎 24 |

[1)] → Device Block overview, table 1, 🗎 16.

[2)] UINT16 (4 × 2 bytes), → Sensor Analog Input Function Block, table 3, 🗎 21

[3)] Float, (4 × 4 bytes), → Sensor Analog Input Function Block, table 3, 🗎 21

[4)] UINT8, (2 × 1 byte), → Transducer Block, 🗎 26

## 4.4 Block Model

Data to the IMG 400 can be transmitted by means of a number of communication protocols and corresponding masters. Profibus defines a master class 1 as normal control unit of the slave (typically a PLC) and a master class 2 as configuration and service unit.

| | |
|---|---|
| MS0 | Cyclic data traffic between master class 1 and slave |
| MSAC1 | Acyclic data traffic between master class 1 and slave |
| MSAC2 | Acyclic data traffic between master class 2 and slave |

With the IMG 400, all functions that are made available by the interface via Profibus are organized in blocks. Access to the individual parameters of the blocks is possible via acyclic services or, for byte, integer and float values, also in cyclic data traffic via the parameter channel.

## Block types

The following block types are defined in the IMG 400.

Device Block

The Device Block contains all data that are required for describing the device and handling its state (status of device state machine).

Transducer Block

The physical, process specific functions or interfaces between the IMG 400 and the process such as current and voltage values are represented in transducer blocks

The following transducer blocks are implemented:

- Transducer Block Channel 1 and 2
  Hot cathode ion gauges
- Transducer Block Channel 3 and 4
  Type A transducers, heat transfer vacuum gauges
  Type B transducers, diaphragm gauges

Function Block

Application specific values such as pressure values that result from or can be calculated from the values of the transducer block are represented in the function blocks.

The following function blocks are implemented:

- Sensor Analog Input Function Block 1 … 4 (channel 1 … 4)
- Trip Function Block 5 … 11 (relays 1 … 7)

### 4.4.1 Device Block (DB)

(Slot 0)

Device Block overview
(Table 1)

The following table lists the services and parameters integrated in the Device Block (→ Appendix B for definitions and abbreviations).

| ID | Name | Structure | Data type | Bytes | Access | Store |
|---|---|---|---|---|---|---|
| 15 | Device Block State | Simple | Unsigned8 | 1 | 1_R/W<br>2_R/W | |
| 16 | Block Type ID | Simple | Octet string | 4 | 1_R<br>2_R | N |
| 25 | Device State | Simple | Unsigned8 | 1 | 1_R<br>2_R | V |
| 26 | Exception Status | Simple | Unsigned8 | 1 | 0_XI<br>1_R<br>2_R | V |
| 203 | Key Lock | Simple | Unsigned8 | 1 | 1_R/W<br>2_R/W | V |
| 204 | Channel Status | Simple | Octet string | 2 | 0_XI<br>1_R<br>2_R | V |
| 205 | Reset Error Condition | Simple | Unsigned8 | 1 | 1_W<br>2_W | V |

Block Type
(ID 16)

The Block Type ID parameter contains the ID of the block as defined in 🕮 [1]). Currently defined Block Type IDs are listed in Appendix C.

Device State
(ID 25)

The parameter contains the overall state of the device as determined by internal events and the commands sent by the master to invoke the Device Block State Command. The possible states are described under the explanation of Device Block State.

| Parameter value | State |
|---|---|
| 0 | Undefined |
| 1 | Self testing |
| 2 | Idle |
| 3 | Self test exception |
| 4 | Executing |
| 5 | Abort |
| 6 | Critical fault |
| 100 | Init |

Exception Status
(ID 26)

The Exception Status describes the alarm and warning statuses of the gauge in an "Expanded error output format" (Bit 7 = "1"), → 🕮 [1].

A difference is made between warnings and errors.

Alarms and errors are divided into three groups (→ sections "Exception Detail Alarm" and "Exception Detail Warning" for details):

- ALARM / Warning Device Common — For errors that occur independently of the type of device used, e.g. supply error, RAM or EEPROM error.
- ALARM / Warning Device Specific — For device specific errors, e.g. Hot cathode power supply error.
- ALARM / Warning Manufacturer Specific — For errors defined by the manufacturer that are not mentioned in the standard e.g. power supply error

In each of the above groups, there are several error or warning conditions. The individual fields are presented in the "Exception Detail Alarm" and "Exception Detail Warning". If an error message occurs in "Exception Detail Alarm" or "Exception Detail Warning", the corresponding bit is set in the Exception Status. Therefore, if bits 0 … 6 of the Exception Status are on "0" there is no warning message pending.

If a bit is set, the actual error can be read in the corresponding group.

The Exception Status is output in cyclic data and informs on the current error status using only one byte. If an error occurs, the current error status can be read via acyclic services or in cyclic data exchange via the parameter channel. This ensures that while the current error status is always available in the cyclic data, no unnecessary data overhead is transmitted.

| Bit | Function | Trigger |
|---|---|---|
| 0 | ALARM/device-common | At least one of bits 0, 2, 3, 4, or 5 in table 3 (Exception Detail 0) is ≠ 0<br>or a serial communication alarm occurred. |
| 1 | ALARM/device-specific | The sensor alarm bytes (ID 103, Transducer 1, 2, 3 or 4) are set to some value ≠ 0. |
| 2 | ALARM/manufacturer-specific | A power supply error or warning has occurred.<br>A ionization gauge supply error or warning has occurred. |
| 4 | WARNING/device-common | At least one of bits 0, 2, 3, 4, or 5 in table 3 (Exception Detail 0) is ≠ 0<br>or a serial communication alarm occurred. |
| 5 | WARNING/device-specific | The Status Extension (ID 102) in Transducer 1, 2, 3 or 4 is set to some value ≠ 0. |
| 6 | WARNING/manufacturer-specific | A power supply error or warning has occurred.<br>An ionization gauge supply error or warning has occurred. |
| 7 | Extended Format | Bit seven is always set. |

Keylock
(ID 203)

This parameter allows the Profibus user to enable or disable the keys on the IMG 400 concerning Profibus accessible parameters.

| Keylock | Coding |
|---|---|
| Off | 0 |
| Para_Lock | 1 |
| Profi_Lock | 2 |
| Full_Lock | 3 |

(For detailed information on keylock → 🕮 [3].)

Channel Status
(ID 204)

The parameter contains the status of each measurement channel. It may be used by the controller application to determine the validity of each channel value. The value of each channel nibble is determined by the respective bit values in the Status Extension Byte of each channel Transducer Block.

**Common Exception Detail 1 (table 2)**

| Bit | Channel | Meaning | Value |
|---|---|---|---|
| 0 | 1 | Reading Invalid | → 🗎 26 (ID 102) |
| 1 | | Overrange | |
| 2 | | Underrange | |
| 3 | | Reserved | 0 |
| 4 | 2 | Reading Invalid | → 🗎 26 (ID 102) |
| 5 | | Overrange | |
| 6 | | Underrange | |
| 7 | | Reserved | 0 |
| 8 | 3 | Reading Invalid | → 🗎 26 (ID 102) |
| 9 | | Overrange | |
| 10 | | Underrange | |
| 11 | | Reserved | 0 |
| 12 | 4 | Reading Invalid | → 🗎 26 (ID 102) |
| 13 | | Overrange | |
| 14 | | Underrange | |
| 15 | | Reserved | 0 |

Reset Error Condition
(ID 205)

The parameter allows the Profibus user to reset alarm and warning states.

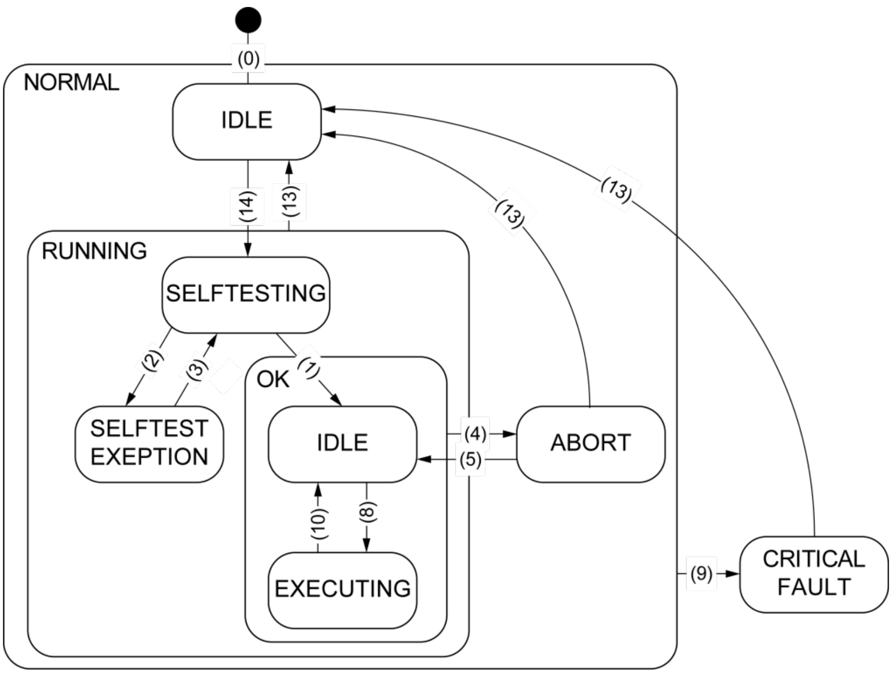
| Bit | Function | Description |
|---|---|---|
| 0 | Reset general power supply alarms | Power supply err |
| 1 | Reset general power supply warnings | Power supply warnings |
| 2 | Reset hot cathode gauge supply alarms | Hot cathode gauge supply alarms |
| 3 | Reset hot cathode gauge supply warnings | Hot cathode gauge supply alarms |
| 4 | Reset all general power supply alarms and hot cathode gauge supply alarms and switch on power supplies of channel 3 and 4 again | |
| 5 | Reset error flag | Error flag was set by:<br>• $p > p_{max.}$<br>• $p > p_{user}$<br>• EMIS key pressed (emergency off) on IMG 400 (→ 🕮 [3]).<br>(→ also tables 5 and 6) |
| 6 | not used | — |
| 7 | Reset all alarms and warnings and clear error buffer | (Alarms and warnings → description of bit … 5)<br>If no valid error exists, the IMG 400 display will no longer show "ERROR xx" (→ 🕮 [3]). |

Where: bit value 0 = no action
bit value 1 = reset command

Device Block State (ID 15)

The IMG 400 behaves as described in the status diagram below.

**State diagram**




After the start, the device independently runs through the INIT and SELFTESTING status and eventually changes to the IDLE status (if there is no error) or to the SELFTEST EXCEPTION status (if there is a device error).

When data traffic is taken up, a difference has to be made between cyclic and acyclic data traffic.

**Cyclic data traffic**

As soon as cyclic data interchange is taken up, the device automatically changes to the EXECUTING status.

**Acyclic data traffic**

In acyclic data traffic, a START service has to be transmitted to bring the device to the EXECUTING status (write 0x04 into ID 15).

**Transition table**

| Transition | Start state | End state |
|---|---|---|
| 0 | Start | INIT |
| 1 | SELFTESTING | IDLE |
| 2 | SELFTESTING | SELFTEST EXCEPTION |
| 3 | SELFTEST EXCEPTION | SELFTESTING |
| 4 | OK | ABORT |
| 5 | ABORT | IDLE |
| 8 | IDLE | EXECUTING |
| 9 | NORMAL | CRITICAL FAULT |
| 10 | EXECUTING | IDLE |
| 13 | RUNNING, ABORT, CRITICAL FAULT | INIT |
| 14 | INIT | SELFTESTING |

**Device status table**

| State | Coding | Description |
|---|---|---|
| INIT | 100 | Block initialization, waiting for AYT response from IMG 400. |
| SELFTESTING | 1 | ROM- and RAM test, evaluation internal communication and EEPROM communication. |
| SELFTEST EXCEPTION | 3 | Internal communication problem discovered during self test. |
| IDLE | 2 | All blocks and device hardware and software have been initialized and have successfully completed self testing. The device is ready for cyclic data transfer. |
| EXECUTING | 4 | Device is executing (e.g., it is performing its device-specific function) its functions according to the purpose of the device. |
| ABORT | 5 | Device Block instance is in an aborted state. The device-specific function are not performed properly. |
| CRITICAL FAULT | 6 | The Device Block (and device) are in a fault state from which there is no recovery. The results of the device-specific functions are bad. |

| Set | Members | Description |
|---|---|---|
| NORMAL | INIT, SELFTESTING, SELFTEST EXCEPTION, IDLE, EXECUTING, ABORT | The set of normal states |
| RUNNING | SELFTESTING, SELFTEST EXCEPTION, IDLE, EXECUTING, | Set of states where device is initialized and no critical faults are detected. |
| OK | IDLE, EXECUTING | Set of states where device is tested and all internal conditions for execution are given. |

**Device Block, State Command**

There are a number of special commands for bringing the device into a status it does not automatically go to.

| ID value | Name | Description |
|---|---|---|
| 0 | Inactive | No action |
| 1 | Reset | Used for reinitializing the device. |
| 2 | Abort | Brings the device to the ABORT status. |
| 3 | Recover | Used for bringing the device from the ABORT status into the Recovered State IDLE. |
| 4 | Execute | Brings the unit to the EXECUTING status, in which the gauge functions normally. As soon as cyclic data traffic is initialized, this status command is executed automatically. |
| 5 | Stop | Brings the gauge to the IDLE status. |
| 6 | Perform Diagnostic | Stops the running activity and starts SELFTEST. |

### 4.4.2 Sensor Analog Input Function Blocks (FB) (Channel 1 … 4) (Table 3)

(Slots 1 … 4 = channels 1 … 4)

| ID | Name | Structure | Data Type | Bytes | Access | Store |
|---|---|---|---|---|---|---|
| 15 | Block Adjust Command | Record | → Semantics | | 2R/W | |
| 16 | Block Type ID | Record | → Semantics | | 2R | N |
| 19 | ProcessValue (PV) | Simple | According Parameter Data Type | – | 0_XI 1_R 2_R | D |
| 20 | Status | Simple | Unsigned8 | 1 | 1_R 2_R | D |
| 21 | Data Type | Simple | Unsigned8 | 1 | 2_R 2_W | N |
| 22 | Data Units | Simple | Unsigned16 | 2 | 2_R | N |
| 23 | Reading Valid | Simple | Boolean | 1 | 1_R 2_R | D |
| 24 | Full Scale | Simple | According Data Type value (parameter 21) | – | 1_R 2_R | N |
| 39 | Safe State | Simple | Unsigned8 | 1 | 1_R 1_W 2_R 2_W | N |
| 40 | Safe Value | Simple | According Data Type value (parameter 21) | – | 1_R 1_W 2_R 2_W | N |
| 44 | Overrange | Simple | According Data Type value (parameter 21) | – | 1_R 2_R | N |
| 45 | Underrange | Simple | According Data Type value (parameter 21) | – | 1_R 2_R | N |

Block Type ID
(ID 16)

The Block Type ID parameter contains the unique ID (2) of the block as defined in 🕮 [1]). Currently defined Block Type IDs are listed in Appendix C.

Process Value (PV)
(ID 19)

The process value contains the measurement value of the Transducer Block dedicated to the channel in the chosen data unit. If the device is not in state EXECUTING (→ ID 25, Device Block) the Process value is set to the value specified by the Safe State (→ ID 39).

Status
(ID 20)

The parameter is always set to "0".

Data Type
(ID 21)

The data type can be changed from Float to Integer16. After startup it is set to the value stored in the EEPROM.

| Data Type | Coding |
|---|---|
| Integer16 | 3 |
| Float | 8 |

Changing the data type of one instance sets the data types of all instances to the same value.
The default factory setting is to "Float".

If the device is in cyclic data exchange, the data type cannot be changed. The first change option is with the settings of the configuration data (choice of standard telegram) during the startup of the cyclic communication.

This method overrules any setup made before, in order to give the priority to cyclic data exchange. The acyclic change may take place using MSAC2 services → 🗎**14** and 🕮 [1]).

Data Unit
(ID 22)

The data unit states the measurement unit of the measurement values.

The device supports the data units described below:

| Data Unit | Coding |
|---|---|
| COUNTS | 0x1001 |
| Torr | 0x1301 |
| mTorr (Micron) | 0x1302 |
| hPa | 0x1308 |
| Pa (Pascal) | 0x1309 |

For safety reasons the data unit may not be changed if the device is in cyclic data exchange with a DP/V0 master. Changing the data unit has to be done with User Parameter during the configuration phase.
The conversion rules from the basic unit hPa are listed below:

**Data Unit Conversion**

Data units can be converted using the rules below:

$$\text{COUNTS} = 4000 \times [\log_{10} (PV_{hPa} / \text{hPa}) + 12.5]$$

$$PV_{Torr} = 0.75006168\ PV_{hPa}$$

$$PV_{Micron} = 10^3 \times PV_{Torr}$$

$$PV_{Pa} = 100 \times PV_{hPa}$$

Reading Valid
(ID 23)

The parameter contains "1" if the device state is set to "EXECUTING" (Device Block, ID 25), the exception status contains no manufacturer warning or alarm and both transducer blocks contain no sensor alarm (ID 103). In all other cases it must be set to "0".

Full Scale
(ID 24)

The parameter contains the valid maximum value of the attached gauge in terms of the currently selected data unit and data type (→ appendix of 🕮 [3]).

**Safe State (ID 39)**

When the device is not in the EXECUTING status (ID 25, Device Block) or if there is a device error, a value defined by Safe State is output as process value. You can select among:

- "0"
- Full scale
- Last valid value
- Safe Value (user-definable in ID 40)

| Safe State | Coding | PV behavior |
|---|---|---|
| Zero | 0 | The Process Value (measured value ID 19) is set to 0. |
| Full Scale | 1 | The Process Value (measured value ID 19) is set to the full scale value (ID 24). |
| Hold Last Value | 2 | The Process Value is set to the last valid value obtained in the EXECUTING status. |
| Use Safe Value | 3 | The Process Value (measured value ID 19) is set to the Safe Value (ID 40). |

**Safe Value (ID 40)**

The Safe Value is the value output with the Process Value parameter (ID 19) when an error occurs or the device goes to the NOT EXECUTING status. If this value is set to zero, it will remain on zero when the data unit is changed.

**Overrange (ID 44)**

Contains the highest valid PV depending on the attached gauge in terms of the unit specified by the parameter "Data Unit" (→ appendix of 🕮 [3]).

**Underrange (ID 45)**

Contains the lowest valid PV depending on the attached gauge in terms of the unit specified by the parameter "Data Unit" (→ appendix of 🕮 [3]).

**Analog Input Block Adjust Command (ID 15)**

| Byte | Name | Structure | Data Type | Bytes | Access |
|---|---|---|---|---|---|
| 0 | State Command | Simple | UINT8 | 1 | 1 / 2_R/W |
| 1 | Target Value | Simple | UINT8 | 1 | 1 / 2_R/W |

| State Command | Name | Description |
|---|---|---|
| 0 | Zero Adjust | Used to calibrate the measurement channel of this instance depending on the transmitted target value. Differing from the profile definition the target value is set as explained in the next table. |

| Target Value | Description |
|---|---|
| 0 | Disable offset (only channel 3 and 4 with CDG sensor) |
| 1 | Start new offset calculation and enable offset (only for Ioni and CDG sensor). |

### 4.4.3 Trip Function Blocks (Relays 1 … 7) (Table 4)

(Slots 5 … 11 = relays 1 … 7)

The Trip relays 1 and 2 are located on the IMG 400 main board. Relays 3 … 7 are located on the IF 400 P (→ 🗎 5 and 🕮 [3]).

| ID | Name | Struct | Data Type | Bytes | Access | Store |
|---|---|---|---|---|---|---|
| 16 | Block Type ID | Record | → Semantics | | 2R | |
| 19 | Low Trip Point | Simple | According Data Type value (ID 29) | – | 1_R/W<br>2_R/W | N |
| 20 | Low Trip Enable | Simple | Boolean | 1 | 1_R/W<br>2_R/W | N |
| 21 | Status | Simple | Unsigned8 | 1 | 0_XI<br>1_R<br>2_R | V |
| 24 | Hysteresis | Simple | According Data Type value (ID 29) | – | 1_R/W<br>2_R/W | N |
| 27 | Analog Input Sensor Function Block Instance | Simple | Unsigned8 | 1 | 1_R/W<br>2_R/W | N |
| 28 | Data Unit | Simple | Unsigned16 | 2 | 2_R | N |
| 29 | Data Type | Simple | Unsigned8 | 1 | 2_R | N |
| 201 | Low Trip Upper Threshold | Simple | According Data Type value (ID 29) | 1 | 1_R/W<br>2_R/W | N |

Block Type ID (ID 16)

The Block Type ID parameter contains the unique ID (17) of the block as defined in 🕮 [1]). Currently defined Block Type IDs are listed in Appendix C.

Low Trip Point (ID 19)

The Value below which a trip point condition will occur.

☞ Relays are activated if pressure falls below the trip point.

Low Trip Enable (ID 20)

Enables or disables the Low Trip Point function.

Status (ID 21)

The state of the relay connected to this instance.

Hysteresis (ID 24)

Determines the amount by which the Sensor Analog Instance value must recover to clear a trip point condition. It can be set directly by writing to the parameter. If the parameter "Low Trip Upper Threshold" is set, it can be recalculated as follows:

| | | |
|---|---|---|
| ID 24 | = | ID 201 – ID 19 |

Analog Input Sensor Function Block Instance (ID 27)

Defines the reference to the measurement channel assigned to this instance.

| Measurement channel | Value |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |

Data Unit
(ID 28)

The device supports the data units described below:

| Unit | Value |
|---|---|
| COUNTS | 0x1001 |
| Torr | 0x1301 |
| mTorr (Micron) | 0x1302 |
| hPa | 0x1308 |
| Pa (Pascal) | 0x1309 |

For safety reasons the data unit may not be changed if the device is in cyclic data exchange with a DP/V0 master. Changing the data unit has to be done with User Parameter during the configuration phase.
The conversion rules from the basic unit hPa are listed below:

**Data Unit Conversion**

Data units can be converted using the rules below:

$$COUNTS = 4000 \times [\log_{10} (PV_{hPa} / hPa) + 12.5]$$

$$PV_{Torr} = 0.75006168\ PV_{hPa}$$

$$PV_{Micron} = 10^3 \times PV_{Torr}$$

$$PV_{Pa} = 100 \times PV_{hPa}$$

Data Type
(ID 29)

The data type can be changed from Float to Integer16. After startup it is set to the value stored in the EEPROM.

| Data Type | Coding |
|---|---|
| Integer16 | 3 |
| Float | 8 |

Changing the data type of one instance sets the data types of all instances to the same value.
The default factory setting is to "Float".

If the device is in cyclic data exchange, the data type cannot be changed. The first change option is with the settings of the configuration data (choice of standard telegram) during the startup of the cyclic communication.

This method overrules any setup made before, in order to give the priority to cyclic data exchange. The acyclic change may take place using MSAC2 services → 🗎14 and 🕮 [1]).

Low Trip Upper Threshold
(ID 201)

Determines upper threshold above which the Sensor Analog Instance Value must recover to clear a trip point condition. It can be set directly by writing to the parameter. If the parameter "Hysteresis" is set it is recalculated as follows:

ID 201 = ID 19 + ID 24

### 4.4.4 Transducer Block Channel 1 and 2 Hot Cathode Ion Gauge (IONI)

The transducer is invoked regardless of the attached gauge for channel 1 and 2.

| ID | Name | Structure | Data Type | Bytes | Access | Store | Initial value |
|---|---|---|---|---|---|---|---|
| 14 | Hot Cathode Block State Command | REC | | 2 | 1_R/W 2_R/W | | FALSE |
| 101 | Block-Type | Simple | Octet-String | 4 | 2_R | | 16 |
| 102 | Status Extension | Simple | UINT8 | 1 | 1_R 2_R | V | 0 |
| 103 | Sensor Alarm | Struct | Array of 2 Bytes | 2 | 1_R 2_R | V | 0 |
| 104 | Sensor Warning | Struct | Array of 2 Bytes | 2 | 1_R 2_R | V | 0 |
| 105 | Emission Status | Simple | BOOL | 1 | 1_R/W 2_R/W | V | 0 |
| 107 | Emission Current | Simple | Float | 4 | 1_R/W 2_R/W | NV | 0 |
| 110 | Degas Status | Simple | BOOL | 1 | 1_R 2_R | V | 0 |

Block Type (ID 101)

The Block Type parameter contains the unique ID (16) of the block as defined in 🕮 [1]). Currently defined Block Type IDs are listed in Appendix C.

Status Extension (ID 102)

The parameter contains the following information:

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Underrange Exceeded 1) | Overrange Exceeded 2) | Reading Invalid 3) |

1) Underrange Exceeded:
The parameter is set to "1", if the sensor value is below the parameter Underrange (ID 45) of the AI FB of the concerned channel (1 or 2).

2) Overrange Exceeded:
The parameter is set to "1", if the sensor is above the parameter Overrange (ID 44) of the AI FB of the concerned channel (1 or 2).

3) Reading invalid:
The parameter is set to "1", if the Reading Valid parameter (ID 23) of the AI FB of the concerned channel (1 or 2) is set to "0" or if the measurement channel represented by this transducer:

- is not connected to a gauge
- has not switched its gauge on.

Sensor Alarm
(ID 103)
(Table 5)

The parameter contains the following information:

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | 0 | Environment Failure [3] | Overpressure Emission off [2] | 0 | Electronics Failure [1] | 0 |

1) The parameter is set to "1", if the byte value of the Manufacturer Exception Alarm Detail 2 or 3 is ≠ "0".

2) The parameter is set to "1", if pressure p > pmax. or p > puser (→ 🕮 [3]). In case of an Emission Off caused by overpressure, the user has to acknowledge the reason for the Emission Off situation. The acknowledgement is carried out by the service "Clear Emission Off Alarm". After receiving this service, the device changes to the Emission Off state (→ table "Supported Commands", 🗎 28).

3) The parameter is set to "1", if the emission has been switched off via keyboard of the IMG 400 (→🕮 [3]).

Sensor Warning
(ID 104)
(Table 6)

The parameter contains the following information:

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | 0 | 0 | Pressure too high for Degas [2] | 0 | Electronics Warning [1] | 0 |

1) The parameter is set to "1", if the byte value of the Manufacturer Exception Warning Detail 2 or 3 is ≠ "0".

2) The parameter is set to "1", if the pressure rises above the specified maximum pressure allowed to start a degas cycle. If the pressure falls below the specified value, bit 3 is reset to "0".

Emission Status
(ID 105)

| Emission Status | Value |
|---|---|
| Off | 0 |
| On | 1 |

Emission Current
(ID 107)

| Emission Current [mA] | Value |
|---|---|
| 0 | 0 |
| 0.1 | 1 |
| 1 | 2 |
| 1.6 | 3 |
| 10 | 4 |
| 45 | 5 |
| 90 | 6 |

Degas Status
(ID 110)

| Degas Status | Value |
|---|---|
| Off | 0 |
| On | 1 |

Hot Cathode Block State Command
(ID 14)

| ID | Name | Struct | Data Type | Bytes | Access | Store | Range |
|---|---|---|---|---|---|---|---|
| 14 | Hot Cathode Block State Command | Record | → below | 2 | 1_R/W<br>2_R/W | N | → below |
| | Transition Command Value | simple | UINT8 | 1 | | | → below |

**Supported Commands**

| Transition Command | Name | Description |
|---|---|---|
| 0 | NOP | No action |
| 1 | Set Degas State | Activates / deactivates degas mode according to the target value:<br>Target Value: 0 ⇒ switch degas off<br>Target Value: 1 ⇒ switch degas on<br>The command will return an error if the pressure is not below:<br>• $1x10^{-4}$ hPa for BAG gauges<br>• $1x10^{-5}$ hPa for EXT gauges<br>Degas will be terminated automatically by the device itself, if not stopped before by the "switch degas off" command. |
| 3 | Set Emission State | Turns emission on or off according to the target value:<br>• Target value: 0 ⇒ switch filament off<br>• Target value: 1 ⇒ switch filament on |
| 4 | Clear Emission Off Alarm | Resets the sensor alarm parameter: Overpressure Off Value<br>• Target value: 0 ⇒ no action<br>• Target value: 1 ⇒ clear Emission Off Alarm. |

## 4.4.5 Transducer Block Channel 3 and 4 (Type A Transducers, Heat Transfer Vacuum Gauges)

☞ The Transducer Block is invoked if a gauge of the type heat transfer vacuum gauge (e.g. TPR) could be identified on the concerned channel.

| ID | Name | Structure | DataType | Bytes | Access | Store |
|---|---|---|---|---|---|---|
| 101 | Block Type | Simple | Octet-String | 4 | 2_R | |
| 102 | Status Extension | Simple | UINT8 | 1 | 1_R<br>2_R | V |
| 103 | Sensor Alarm | Struct | Array of Byte 2 | 2 | 1_R<br>2_R | V |
| 104 | Sensor Warning | Struct | Array of Byte 2 | 2 | 1_R<br>2_R | V |

Block Type (ID 101)

The Block-Type parameter contains the unique ID (13) of the block as defined in 🕮 [1]). Currently defined Block Type IDs are listed in Appendix C.

Status Extension
(ID 102)

The parameter contains the following information:

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Underrange Exceeded [1)] | Overrange Exceeded [2)] | Reading Invalid [3)] |

1) Underrange Exceeded:
The parameter is set to "1", if the sensor value is below the parameter Underrange (ID 45) of the AI FB of the concerned channel (3 or 4).

2) Overrange Exceeded:
The parameter is set to "1", if the sensor value is above the parameter Overrange (ID 44) of the AI FB of the concerned channel (3 or 4).

3) Reading invalid:
The parameter is set to "1", if the Reading Valid parameter (ID 23) of the AI FB instance 3 is set to "0" or if the measurement channel represented by this transducer

- is not connected to a gauge
- has not switched its gauge on.

Sensor Alarm
(ID 103)

The parameter contains the following information:

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | 0 | 0 | 0 | 0 | Electronics Failure [1)] | 0 |

1) The parameter is set to "1", if bit 0 or bit 1 of the Manufacturer Exception Alarm Detail 1 byte is set to "1".

Sensor Warning
(ID 104)

The parameter contains the following information:

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | 0 | 0 | 0 | 0 | Electronics Warning [1)] | 0 |

1) The parameter is set to "1", if bit 0 or bit 1 of the Manufacturer Exception Warning Detail 1 byte is set to "1".

### 4.4.6 Transducer Block Channel 3 and 4 (Type B Transducers, Diaphragm Gauges)

☞ The Transducer Block is invoked, if a gauge of type CMR (Diaphragm gauge) was identified on the concerned channel.

| ID | Name | Structure | DataType | Bytes | Access | Store | Initial Value |
|---|---|---|---|---|---|---|---|
| 101 | Block Type | Simple | Octet-String | 4 | 2_R | | 14 |
| 102 | Status Extension | Simple | UINT8 | 1 | 1_R<br>2_R | V | 0 |
| 103 | Sensor Alarm | Struct | Array of 2 Bytes | 2 | 1_R<br>2_R | V | 0 |
| 104 | Sensor Warning | Struct | Array of 2 Bytes | 2 | 1_R<br>2_R | V | 0 |

Block Type
(ID 101)

The Block-Type parameter contains the unique ID (14) of the block as defined in 🕮 [1]). Currently defined Block Type IDs are listed in Appendix C.

Status Extension
(ID 102)

The parameter contains the following information:

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Underrange Exceeded [1] | Overrange Exceeded [2] | Reading Invalid [3] |

[1] Underrange Exceeded:
The parameter is set to "1", if the sensor is below the parameter Underrange (ID 45) of the AI FB of the concerned channel (3 or 4).

[2] Overrange Exceeded:
The parameter is set to "1", if the sensor value is above the parameter Overrange (ID 44) of the AI FB of the concerned channel (3 or 4).

[3] Reading invalid:
The parameter is set to "1", if the Reading Valid parameter (ID 23) of the AI FB of the concerned channel is set to "0" or if the measurement channel represented by this transducer

- is not connected to a gauge
- has not switched its gauge on.

Sensor Alarm
(ID 103)

The parameter contains the following information:

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | 0 | 0 | 0 | 0 | Electronics Failure [1] | 0 |

[1] The parameter is set to "1", if bit 0 or bit 1 of the Manufacturer Exception Alarm Detail 1 byte is set to "1".

Sensor Warning
(ID 104)

The parameter contains the following information:

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | 0 | 0 | 0 | 0 | Electronics Warning [1] | 0 |

[1] The parameter is set to "1", if bit 0 or bit 1 of the Manufacturer Exception Warning Detail 1 byte is set to "1".

# 5 Maintenance

## 5.1 Installing / Replacing the Profibus Firmware EPROM

If a new firmware becomes available, the corresponding EPROM on the IF 400 P can be changed by the end-user.

The transmitted telegrams are located, without exception, within the following frameworks:

**Skilled personnel**

The plug-in board may only be installed by persons who have suitable technical training and the necessary experience or who have been instructed by the end-user of the product.

**WARNING**

Static electricity.

Static electricity can damage electronic components.

Do not touch the electronic components unless you have taken the appropriate protective measures. Store product in its antistatic bag.

Required tools

- Screwdriver (size 2)
- Removal tool for 32 pin PLCC integrated circuits

Procedure

❶ Switch off the IMG 400

Wait one minute.

❷ Remove Profibus cable (and relays cable, if applicable) from IF 400 P

❸ Unscrew the rear panel of the IF 400 P and slide out the module

❹ Remove old EPROM from its socket with the PLCC handling tool

❺ Unpack new EPROM and check version number printed on label (→ 🗎 41)

❻ Install new EPROM carefully. Ensure correct orientation of the EPROM in the PLCC socket (→ 🗎 41)

❼ Re-install the IF 400 P in the IMG 400 following the procedure described on 🗎 5, starting with step ❸.

# 6 Appendix

## A: Gauge – Channel Configuration

Permitted channel configurations

| Channel → Configuration ↓ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | BAG | | | |
| 2 | EXT | | | |
| 3 | | EXT | | |
| 4 | | BAG | | |
| 5 | | | CMR | |
| 6 | | | TPR | |
| 7 | | | | TPR |
| 8 | | | | CMR |
| 9 | BAG | BAG | | |
| 10 | BAG | EXT | | |
| 11 | EXT | BAG | | |
| 12 | EXT | EXT | | |
| 13 | BAG | | TPR | |
| 14 | BAG | | CMR | |
| 15 | EXT | | TPR | |
| 16 | EXT | | CMR | |
| 17 | BAG | | | TPR |
| 18 | BAG | | | CMR |
| 19 | EXT | | | TPR |
| 20 | EXT | | | CMR |
| 21 | | BAG | TPR | |
| 22 | | BAG | CMR | |
| 23 | | EXT | TPR | |
| 24 | | EXT | CMR | |
| 25 | | BAG | | TPR |
| 26 | | BAG | | CMR |
| 27 | | EXT | | TPR |
| 28 | | EXT | | CMR |
| 29 | BAG | BAG | TPR | |
| 30 | BAG | BAG | CMR | |
| 31 | BAG | EXT | TPR | |
| 32 | BAG | EXT | CMR | |
| 33 | EXT | BAG | TPR | |
| 34 | EXT | BAG | CMR | |
| 35 | EXT | EXT | TPR | |
| 36 | EXT | EXT | CMR | |
| 37 | BAG | BAG | | TPR |
| 38 | BAG | BAG | | CMR |
| 39 | BAG | EXT | | TPR |
| 40 | BAG | EXT | | CMR |

(continued)

(Table "Permitted channel configurations" concluded)

| Channel → Configuration ↓ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 41 | EXT | BAG | | TPR |
| 42 | EXT | BAG | | CMR |
| 43 | EXT | EXT | | TPR |
| 44 | EXT | EXT | | CMR |
| 45 | BAG | | TPR | TPR |
| 46 | BAG | | TPR | CMR |
| 47 | BAG | | CMR | TPR |
| 48 | BAG | | CMR | CMR |
| 49 | EXT | | TPR | TPR |
| 50 | EXT | | TPR | CMR |
| 51 | EXT | | CMR | TPR |
| 52 | EXT | | CMR | CMR |
| 53 | | BAG | TPR | TPR |
| 54 | | BAG | TPR | CMR |
| 55 | | BAG | CMR | TPR |
| 56 | | BAG | CMR | CMR |
| 57 | | EXT | TPR | TPR |
| 58 | | EXT | TPR | CMR |
| 59 | | EXT | CMR | TPR |
| 60 | | EXT | CMR | CMR |
| 61 | BAG | BAG | TPR | TPR |
| 62 | BAG | BAG | TPR | CMR |
| 63 | BAG | BAG | CMR | TPR |
| 64 | BAG | BAG | CMR | CMR |
| 65 | BAG | EXT | TPR | TPR |
| 66 | BAG | EXT | TPR | CMR |
| 67 | BAG | EXT | CMR | TPR |
| 68 | BAG | EXT | CMR | CMR |
| 69 | EXT | BAG | TPR | TPR |
| 70 | EXT | BAG | TPR | CMR |
| 71 | EXT | BAG | CMR | TPR |
| 72 | EXT | BAG | CMR | CMR |
| 73 | EXT | EXT | TPR | TPR |
| 74 | EXT | EXT | TPR | CMR |
| 75 | EXT | EXT | CMR | TPR |
| 76 | EXT | EXT | CMR | CMR |

Where: BAG = Bayard Alpert hot cathode gauge
CMR = Capacitance diaphragm gauge
EXT = Extractor ionization gauge
TPR = Pirani standard gauge

☞ Empty cells in the table above contain place holders (→ following table).

Invalid transducer configurations

☞ The transducer types written in the following table serve as place holders if no gauge is connected. Their "Reading invalid" parameter is always set to "1".
Empty cells in the table below may contain valid gauge types (→ previous table).

| Slot → Configuration ↓ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | | IONI | PIRANI | PIRANI |
| 2 | | IONI | PIRANI | PIRANI |
| 3 | IONI | | PIRANI | PIRANI |
| 4 | IONI | | PIRANI | PIRANI |
| 5 | IONI | IONI | | PIRANI |
| 6 | IONI | IONI | | PIRANI |
| 7 | IONI | IONI | PIRANI | |
| 8 | IONI | IONI | PIRANI | |
| 9 | | | PIRANI | PIRANI |
| 10 | | | PIRANI | PIRANI |
| 11 | | | PIRANI | PIRANI |
| 12 | | | PIRANI | PIRANI |
| 13 | | IONI | | PIRANI |
| 14 | | IONI | | PIRANI |
| 15 | | IONI | | PIRANI |
| 16 | | IONI | | PIRANI |
| 17 | | IONI | PIRANI | |
| 18 | | IONI | PIRANI | |
| 19 | | IONI | PIRANI | |
| 20 | | IONI | PIRANI | |
| 21 | IONI | | | PIRANI |
| 22 | IONI | | | PIRANI |
| 23 | IONI | | | PIRANI |
| 24 | IONI | | | PIRANI |
| 25 | IONI | | PIRANI | |
| 26 | IONI | | PIRANI | |
| 27 | IONI | | PIRANI | |
| 28 | IONI | | PIRANI | |
| 29 | | | | PIRANI |
| 30 | | | | PIRANI |
| 31 | | | | PIRANI |
| 32 | | | | PIRANI |
| 33 | | | | PIRANI |
| 34 | | | | PIRANI |
| 35 | | | | PIRANI |
| 36 | | | | PIRANI |
| 37 | | | PIRANI | |
| 38 | | | PIRANI | |
| 39 | | | PIRANI | |
| 40 | | | PIRANI | |

(continued)

(Table "Invalid transducer configurations" concluded)

| Slot → Configuration ↓ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 41 | | | PIRANI | |
| 42 | | | PIRANI | |
| 43 | | | PIRANI | |
| 44 | | | PIRANI | |
| 45 | | IONI | | |
| 46 | | IONI | | |
| 47 | | IONI | | |
| 48 | | IONI | | |
| 49 | | IONI | | |
| 50 | | IONI | | |
| 51 | | IONI | | |
| 52 | | IONI | | |
| 53 | IONI | | | |
| 54 | IONI | | | |
| 55 | IONI | | | |
| 56 | IONI | | | |
| 57 | IONI | | | |
| 58 | IONI | | | |
| 59 | IONI | | | |
| 60 | IONI | | | |
| 61 | | | | |
| 62 | | | | |
| 63 | | | | |
| 64 | | | | |
| 65 | | | | |
| 66 | | | | |
| 67 | | | | |
| 68 | | | | |
| 69 | | | | |
| 70 | | | | |
| 71 | | | | |
| 72 | | | | |
| 73 | | | | |
| 74 | | | | |
| 75 | | | | |
| 76 | | | | |

# B: Definitions

Data types

| Abbreviation | Range | Data type |
|---|---|---|
| INT8 | $-2^7 \ldots (2^7 - 1)$ | Integer 1 byte |
| INT16 | $-2^{15} \ldots (2^{15} - 1)$ | Integer 2 byte |
| INT32 | $-2^{31} \ldots (2^{31} - 1)$ | Integer 4 byte |
| UINT8 | $0 \ldots (2^8 - 1)$ | Unsigned integer 1 byte |
| UINT16 | $0 \ldots (2^{16} - 1)$ | Unsigned integer 2 byte |
| UINT32 | $0 \ldots (2^{31} - 1)$ | Unsigned integer 4 byte |
| FLOAT | $\pm 3.402 \times 10^{38}$ | Floating Point, IEEE 754 Short Real Number, 4 byte |
| VSTRING(n) | | ISO 646 and ISO 2375 |
| OSTRING(n) | | Octet string |

Definitions

| Term | Meaning |
|---|---|
| Byte | Number of bytes used by a data structure (integer value) |
| Store | This parameter defines whether the values are stored in non-volatile memory (→ store characteristics) |
| Default | Manufacturer-defined value |

Store characteristics

| Abbreviation | Meaning |
|---|---|
| V | "Volatile": Value is not saved to the RAM or EEPROM and is lost in the event of a power failure |
| N | "Nonvolatile": Value is saved to the RAM or EEPROM and is not lost in the event of a power failure |

Data access

| Abbreviation | Meaning |
|---|---|
| 1_R/W | Acyclically readable and writeable by a Master Class 1 |
| 2_R/W | Acyclically readable and writeable by a Master Class 2 |
| 1_R | Acyclically readable by a master Class 1 |
| 2_R | Acyclically readable by a master Class 2 |
| 1_W | Acyclically writeable by a master Class 1 |
| 2_W | Acyclically writeable by a master Class 2 |
| 0_XI | Cyclic output data with master Class 1 |

## Definitions from the Profibus standard

The following table explains terms used in connection with the Profibus.

| Term | Meaning |
| --- | --- |
| Alert Elements | Alert Elements are used to communicate notification messages from slave to master when warnings, alarms or events are detected. |
| Application | A *software functional unit* consisting of an interconnected aggregation of *function blocks, events and objects*, which may be distributed and which may have *interfaces* with other *applications*. |
| Characteristic | An characteristic is a property or characteristic of an *entity*. (Au) In block applications a block interface is defined by input/output parameters. These parameters have characteristics called parameter characteristics. Examples are access rights and identification names. (IT) The UML defines characteristics as a feature within a classifier that describes a range of values that instances of the classifier may hold. It is a property of a class instance (object). |
| Block (Block Instance) | A logical processing unit of software comprising an individual, named copy of the block and associated parameters specified by a block type, which persists from one invocation of the block to the next. Concept similar to the class/ object approach, but well suited to the automation requirements. |
| Class | (IT) A class represents a template for several objects and describes how these objects are structured internally. Objects of the same class have the same definition both for their operations and for their information structures. |
| Configuration (of a system/device) | A step in system design: selecting functional units, assigning their locations and identifiers and defining their interconnections. |
| Data Structure | An *aggregate* whose elements need not be of the same *data type*, and each of them is uniquely referenced by an *offset identifier*. |
| Data Type | A data item with certain characteristics and permissible operations on that data, e.g. INT8. |
| Device | A physical entity capable of performing one or more specified functions in a particular context and delimited by its interfaces. |
| Direction of Data | *Input data* are transmitted from the device to the bus. *Output data* are transmitted from the bus to the device. |
| Direction of Flow | A positive set point causes a flow from P to A. |
| Entity | A particular thing, such as a person, place, *process*, object, concept, association or *event*. |
| Function | (1) A specific purpose of an entity. (2) One of a group of actions performed by an entity. |
| Function Block | A named block consisting of one or more input, output and contained parameters. Function blocks represent the basic automation functions performed by an application which is as independent as possible from the specifics of I/O devices and the network. Each function block processes input parameters according to a specified algorithm and an internal set of contained parameters. They produce output parameters that are available for use within the same function block application or by other function block applications. |

## C: Block Type

Currently defined Block Type IDs

| Block Name | Block Type ID |
|---|---|
| Device Block | 1 |
| Sensor Analog Input Function Block | 2 |
| One of N Channel Sensor Analog Input Function Block | 3 |
| Multi Channel Sensor Analog Input Function Block | 4 |
| Discrete Input Function Block | 5 |
| Actuation Analog Output Function Block | 6 |
| Discrete Output Function Block | 7 |
| Analog Output Function Block | 8 |
| Single Stage Controller Function Block | 9 |
| Gas Calibration Transducer Block | 10 |
| Flow Transducer Block | 11 |
| Sensor Analog Input Ambient Temperature Transducer Block | 12 |
| Heat Transfer Vacuum Gauge [1)] | 13 |
| Diaphragm Gauge | 14 |
| Cold Cathode Ion Gauge | 15 |
| Hot Cathode Ion Gauge [2)] | 16 |
| Trip Point Function Block | 17 |
| Reserved | 18 … $2^8 – 1$ |
| Manufacturer-specific | $2^8$ … $2^{16} – 1$ |

1) Pirani measuring system

2) BAG/EXT measuring system

## D: Example Using the Inputs and Outputs

| Parameter channel (example) | Byte 1 Low byte (100) | Byte 2 (101) | Byte 3 (102) | Byte 4 (103) | Byte 5 (104) | Byte 6 (105) | Byte 7 (106) | Byte 8 High byte (107) |
|---|---|---|---|---|---|---|---|---|
| | **PKE** *) | | **Index** | **Res.** | **PWE** | | | |
| | Bit position 15 … 0 (AK / res. / Slot) | | | | Low byte | | | High byte |
| | 100.7-100.4 | 101.7-101.0 | Byte 102 | Byte 103 | Byte 104 | Byte 105 | Byte 106 | Byte 107 |
| Output | 0001_0000 | 0000_0100 | 22 | – | | | | |
| Meaning | Read | Channel 4 | Data unit | – | 0 | 0 | 0 | 0 |
| Input | 0001_0000 | 0000_0100 | 22 | – | 0 | 0 | 13 | 08 |
| Meaning | Read | Channel 4 | Data unit | – | 0 | 0 | hPa | |

*) PKE

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

# E: Examples Parameter Channel

The following example shows how to program the parameter channel.

For this sample the Configuration number 8 is used. The Input data is set up with the parameter channel followed by exception status and channel values in float. The output data is set up with parameter channel followed by transition command and transition command value.

Input data

| Low | | | | | | | | | | | | | | High |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte | | | | | | | | Byte | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | … | 25 |
| Parameter channel | | | | | | | | Process data and status | | | | | | |
| PKE | | ID | res. | PWE | | | | ES | Process Value 1 | | | | … | PV4 |

Output data

| Low | | | | | | | | | High |
|---|---|---|---|---|---|---|---|---|---|
| Byte | | | | | | | | Byte | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Parameter channel | | | | | | | | Process | |
| PKE | | ID | res. | PWE | | | | TC | TV |

Where:
ES = Exception Status
PV = Process Value
TC = Transition Command
TV = Transition Command Value

## Emission control using Transition Command

Using the Transition Parameter the emission can be controlled:

Emission on:
Output data

| Byte | | | | | | | | Byte | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Parameter channel | | | | | | | | Process | |
| PKE | | ID | res. | PWE | | | | TC | TV |
| xx | xx | xx | xx | xx | xx | xx | xx | 03h | 01h |

Emission off:
Output data

| Byte | | | | | | | | Byte | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Parameter channel | | | | | | | | Process | |
| PKE | | ID | res. | PWE | | | | TC | TV |
| xx | xx | xx | xx | xx | xx | xx | xx | 03h | 00h |

No action:
Output data

| Byte | | | | | | | | Byte | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Parameter channel | | | | | | | | Process | |
| PKE | | ID | res. | PWE | | | | TC | TV |
| xx | xx | xx | xx | xx | xx | xx | xx | 00h | 00h |

## Emission control using Parameter Channel

Using Parameter Channel the emission can be controlled.

Read Emission Status from Transducer Block 2(02h), ID 105(69h):
Output data

| Byte | | | | | | | | Byte | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Parameter channel | | | | | | | | Process | |
| PKE | | ID | res. | PWE | | | | TC | TV |
| 10h | 02h | 69h | 00h | 00h | 00h | 00h | 00h | 00h<br>xx | 00h<br>xx |

Input data

| Byte | | | | | | | | Byte | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | … | 25 |
| Parameter channel | | | | | | | | Process data and status | | | | | | |
| PKE | | ID | res. | PWE | | | | ES | Process Value 1 | | | | … | PV4 |
| B0h | 02h | 69h | 00h | 00h | 00h | 00h | 00h | xx | xx | xx | xx | xx | xx | xx |

Turn emission on writing to Transducer Block 2(02h), ID 105(69h):
Output data

| Byte | | | | | | | | Byte | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Parameter channel | | | | | | | | Process | |
| PKE | | ID | res. | PWE | | | | TC | TV |
| A0h | 02h | 69h | 00h | 00h | 00h | 00h | 01 | 00h | 00h |

Input data

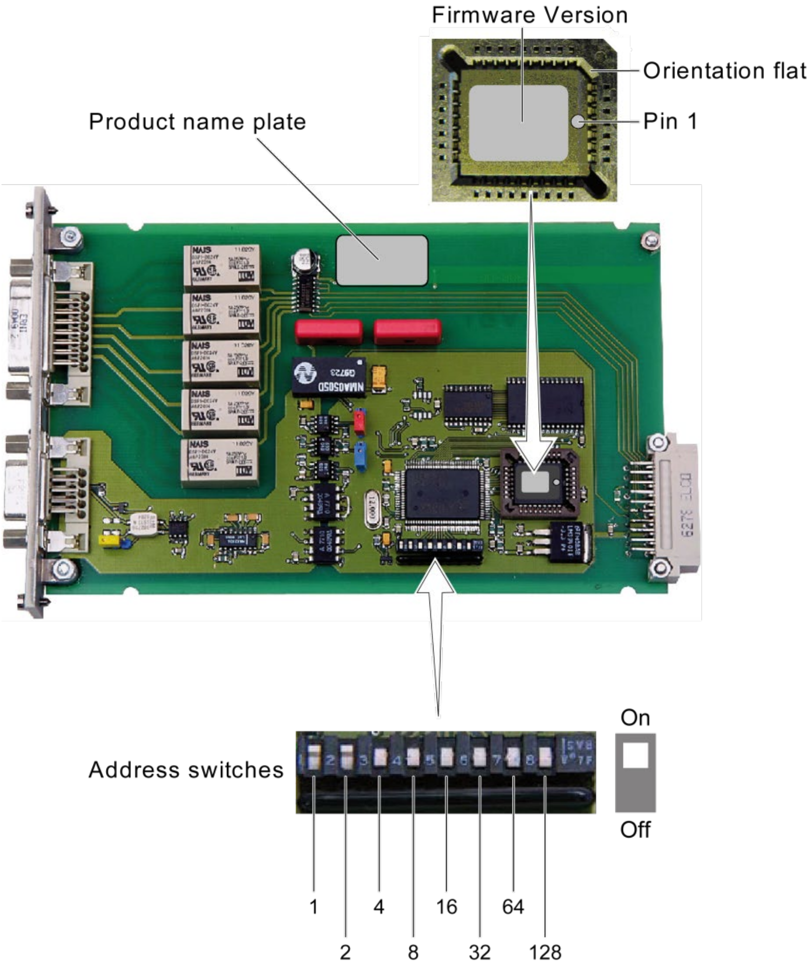
| Byte | | | | | | | | Byte | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | … | 25 |
| Parameter channel | | | | | | | | Process data and status | | | | | | |
| PKE | | ID | res. | PWE | | | | ES | Process Value 1 | | | | … | PV4 |
| B0h | 02h | 69h | 00h | 00h | 00h | 00h | 01h | xx | xx | xx | xx | xx | xx | xx |

No action:
Output data

| Byte | | | | | | | | Byte | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Parameter channel | | | | | | | | Process | |
| PKE | | ID | res. | PWE | | | | TC | TV |
| 00h | 02h<br>xx | 69h<br>xx | 00h<br>xx | 00h<br>xx | 00h<br>xx | 00h<br>xx | 01h<br>xx | 00h | 00h<br>xx |

# F: Board Layout

The following illustration shows the location of the device address switches, EPROM socket for the Profibus firmware and the product name plate on the IF 400 P board.



# G: Literature


📖 [1] www.profibus.com
Profibus user organization

📖 [2] IEC 61158 Type 3 elements: Industrial communication networks – Fieldbus specifications
IEC 61784: Industrial communication networks – Fieldbus profiles

📖 [3] www.pfeiffer-vacuum.com
Operating Instructions
Ionization Gauge Controller IMG 400
BG 5520 BDE (German)
BG 5520 BEN (English)
Pfeiffer Vacuum GmbH, D–35614 Asslar, Germany

Notes

## VACUUM SOLUTIONS FROM A SINGLE SOURCE

**Pfeiffer Vacuum stands for innovative and custom vacuum solutions worldwide, technological perfection, competent advice and reliable service.**

## COMPLETE RANGE OF PRODUCTS

**From a single component to complex systems:**
**We are the only supplier of vacuum technology that provides a complete product portfolio.**

## COMPETENCE IN THEORY AND PRACTICE

**Benefit from our know-how and our portfolio of training opportunities!**
**We can support you with your plant layout and provide first-class on-site-service worldwide.**

Are you looking for a
perfect vacuum solution?
Please contact us:

Pfeiffer Vacuum GmbH
Headquarters • Germany
T +49 6441 802-0
info@pfeiffer-vacuum.de

www.pfeiffer-vacuum.com

PFEIFFER VACUUM

bg5530ben/a