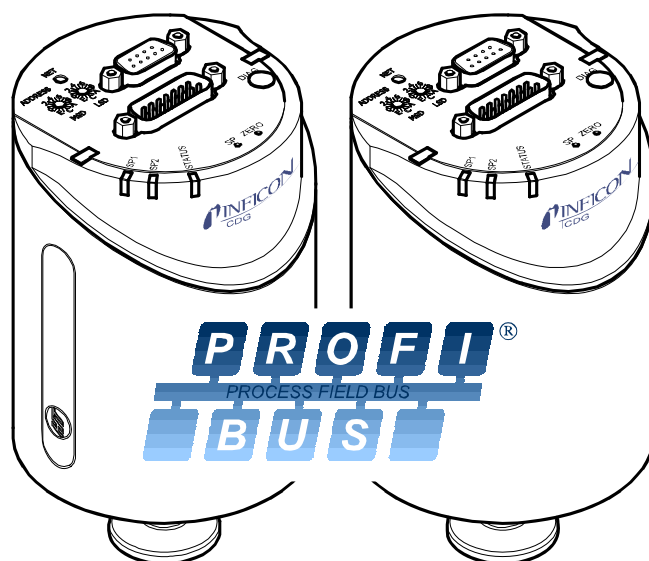


# Profibus

DP/V1 Interface for Capacitance Diaphragm Gauge

CDG045D, CDG100D,  
CDG160D, CDG200D



## General Information

**Caution**

**Caution: data transmission errors**

Any attempt to simultaneously operate the gauge via the RS232C Serial Interface and a Fieldbus interface (DeviceNet or Profibus) or the diagnostic port may result in incorrect data and data transmission errors.

Therefore, it is inadmissible to simultaneously operate the gauge via the RS232C Serial Interface and DeviceNet, Profibus, or the diagnostic port.

## About this Document

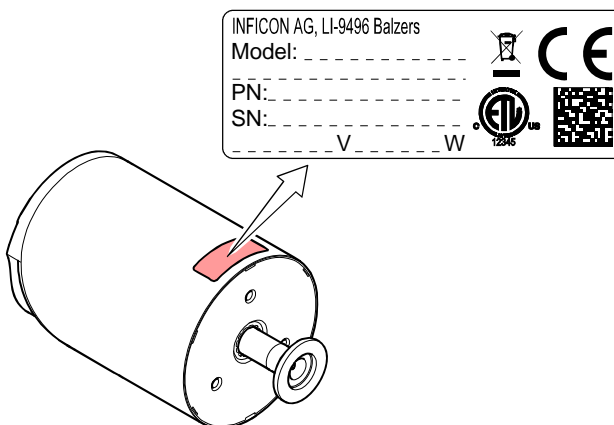
This document describes the functionality and programming of the Profibus interface of the temperature compensated CDG045D, CDG100D, CDG160D and CDG200D gauges.



For safety information on and further technical data of the gauges, please refer to the respective operating manuals (→ [1], [2], [3]).

## Product Identification

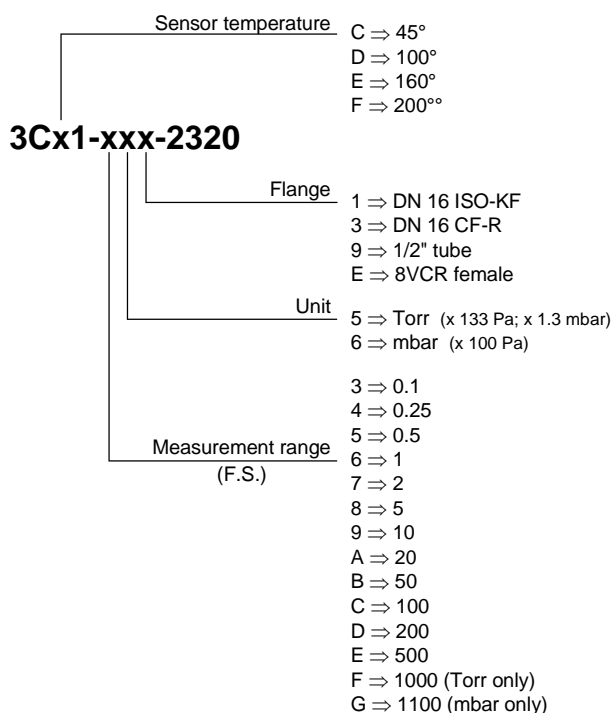
In all communications with INFICON, please specify the information on the product nameplate. For convenient reference copy that information into the space provided below.



## Validity

This document applies to products of the CDG045D, CDG100D, CDG160D and CDG200D series with Profibus interface.

Part numbers of standard products are indicated below. OEM products have other part numbers and different parameter settings (e.g. factory setting of setpoint) as defined in the corresponding ordering information.



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 CDG045D gauges with the DN 16 ISO-KF vacuum connection. They apply to other vacuum connections and to the other gauges by analogy.

We reserve the right to make technical changes without prior notice.

## Trademarks

SEMI® Semiconductor Equipment and Materials International, California

Profibus This document contains Profibus-specific information described and defined in the Profibus standard EN 50170 (→  [5]).

# Contents

General Information	2
About this Document	2
Product Identification	2
Validity	3
Trademarks	3
<b>1 Technical Data</b>	<b>7</b>
<b>2 Power Connection</b>	<b>8</b>
<b>3 Operation</b>	<b>9</b>
3.1 Introduction	9
3.2 Operating Software	9
3.3 Node Address	9
3.4 "NET" LED	9
<b>4 General Data</b>	<b>10</b>
4.1 Data Rate	10
4.2 Ident Number	10
4.3 Configuration Data	10
4.4 User Parameter Data	11
4.5 Types of Communication	11
<b>5 Data Exchange Mode</b>	<b>12</b>
5.1 Acyclic Data Transmission with Profibus DPV1 Functionality	12
5.2 Cyclic Data Telegrams in Data Exchange Mode	13
5.3 Parameter Channel	14
5.3.1 PKW	14
5.3.2 Instruction and response processing	14
5.3.3 Sequence Diagram	16
5.4 Process Data	17
5.4.1 Standard Telegrams	17
5.4.2 Proprietary Telegrams	18
5.4.3 Configuration Data	18
<b>6 Block Model</b>	<b>19</b>
6.1 Block types	19
<b>7 Device Block</b>	<b>20</b>
7.1 Information on the individual Indices	21
7.1.1 Block Type ID (ID 16)	21
7.1.2 Device Type (ID 17)	21
7.1.3 Standard Revision Level (ID 18)	21
7.1.4 Device Manufacturer Identifier (ID 19)	21
7.1.5 Manufacturer Model Number (ID 20)	21
7.1.6 Software or Firmware Revision Level (ID 21)	21
7.1.7 Hardware Revision Level (ID 22)	21
7.1.8 Serial Number (ID 23)	21
7.1.9 Device Configuration (ID 24)	21
7.1.10 Device State (ID 25)	21
7.1.11 Exception Status (ID 26)	22
7.1.12 Exception Detail Alarm (ID 27)	23
7.1.13 Exception Detail Warning (ID 28)	24
7.1.14 Run hours (ID 36)	24
7.1.15 Basic Device Firmware Revision Level (ID 202)	24
7.1.16 Basic Device Hardware Revision Level (ID 203)	24
7.1.17 Common Exception Detail Alarm (ID 204)	24
7.1.18 Device Exception Detail Alarm (ID 205)	24
7.1.19 Manufacturer Exception Detail Alarm (ID 206)	24
7.1.20 Common Exception Detail Warning (ID 207)	24
7.1.21 Device Exception Detail Warning (ID 208)	25
7.1.22 Manufacturer Exception Detail Warning (ID 209)	25
7.2 Device Block Status diagram	25
7.2.1 General Device Block State descriptions	26
7.2.2 Device Block State Command (ID 15)	27
<b>8 Sensor Analog Input Function Blocks</b>	<b>28</b>
8.1 Sensor Analog Input Function Block (Instance 1, CDG)	28
8.1.1 Block Type ID (ID 16)	29
8.1.2 AlarmEnable (ID17)	29
8.1.3 WarningEnable (ID 18)	29

8.1.4	Process Value (ID 19)	29
8.1.5	Status (ID 20)	29
8.1.6	Data Type (ID 21)	29
8.1.7	Data Unit (ID 22)	29
8.1.8	Reading Valid (ID 23)	30
8.1.9	Full Scale (ID 24)	30
8.1.10	Offset-A Data Type (ID 25)	30
8.1.11	Offset-A (ID 26)	30
8.1.12	Alarm Trip Point High (ID 31)	30
8.1.13	Alarm Trip Point Low (ID 32)	30
8.1.14	Alarm Hysteresis (ID 33)	30
8.1.15	Warning Trip Point High (ID 35)	30
8.1.16	Warning Trip Point Low (ID 36)	30
8.1.17	Warning Hysteresis (ID 37)	31
8.1.18	Safe State (ID 39)	31
8.1.19	Safe Value (ID 40)	31
8.1.20	Overrange (ID 44)	31
8.1.21	Underrange (ID 45)	31
8.1.22	Analog Input Block Adjust Command (ID 15)	32
8.2	Sensor Analog Input Function Block (Instance 2, ATM)	33
8.2.1	Block Type ID (ID 16)	33
8.2.2	Process Value (ID 19)	33
8.2.3	Status (ID 20)	33
8.2.4	Data Type (ID 21)	33
8.2.5	Data Unit (ID 22)	34
8.2.6	Reading Valid (ID 23)	34
8.2.7	Full Scale (ID 24)	34
8.2.8	Safe State (ID 39)	34
8.2.9	Safe Value (ID 40)	35
8.2.10	Overrange (ID 44)	35
8.2.11	Underrange (ID 45)	35
8.2.12	Analog Input Block Adjust Command (ID 15)	35
<b>9</b>	<b>OneOfN Analog Input Function Block</b>	<b>36</b>
9.1	Information on the Individual Indices	36
9.1.1	Block behavior (ID 15)	36
9.1.2	Block Type ID (ID 16)	36
9.1.3	Parameters 17...45	36
9.1.4	Channel Instance Selector (ID 46)	36
<b>10</b>	<b>Discrete Output Function Block (Relay 1 and 2)</b>	<b>37</b>
10.1	Information on the Individual Indices	37
10.1.1	Block Type ID (ID 16)	37
10.1.2	Setpoint Value (ID 19)	37
10.1.3	Status (ID 20)	37
10.2	Discrete Output Function Block behavior	38
10.2.1	Block State Command Mapping	38
<b>11</b>	<b>Trip Point Function Block (Relay 1 and 2)</b>	<b>39</b>
11.1	Trip Point Function Block behaviors	39
11.1.1	Setpoint function	39
11.1.2	Atmosphere detection	39
11.2	Trip Point Function Block (Relay 1 and Relay 2)	40
11.2.1	Block Type ID (ID 16)	40
11.2.2	Low Trip Point (ID 19)	40
11.2.3	Low Trip Enable (ID 20)	40
11.2.4	Status (ID 21)	40
11.2.5	Hysteresis (ID 24)	41
11.2.6	Discrete Output Function Block Instance (ID 26)	41
11.2.7	Analog Input Sensor Function Block Instance (ID 27)	41
11.2.8	Data Units (ID 28)	41
11.2.9	Data Type (ID 29)	41
11.2.10	Channel Instance Selector (ID 30)	41
11.2.11	Percentage of ATM (ID 201)	41
11.2.12	High Trip Point (ID 202)	41
11.2.13	High Trip Enable (ID 203)	41
<b>12</b>	<b>Transducer Blocks</b>	<b>42</b>
12.1	CDG Transducer Block (Instance 1)	42
12.1.1	Block Type ID (ID 101)	42
12.1.2	StatusExtension (ID 102)	42
12.1.3	Sensor Alarm (ID 103)	42
12.1.4	Sensor Warning (ID 104)	43

12.1.5 Sensor Temperature (ID 105)	43
12.1.6 Offset (ID 201)	43
12.1.7 Filter (ID 202)	43
12.2 ATM Transducer Block (Instance 2)	44
12.2.1 Block Type ID (ID 101)	44
12.2.2 StatusExtension (ID 102)	44
12.2.3 Sensor Alarm (ID 103)	44
12.2.4 Sensor Warning (ID 104)	44
12.2.5 Analog Input Block Adjust Command (ID 14)	45
<b>Appendix A: Definitions</b>	<b>46</b>
<b>Appendix B: Block Type</b>	<b>49</b>
<b>Appendix C: Literature</b>	<b>50</b>

For cross-references to other documents, the symbol (→  [XY]) is used.

# 1 Technical Data

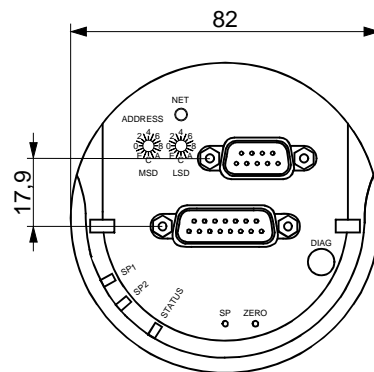


Further technical data → [1], [2], [3].

## Profibus interface

Fieldbus name	Profibus
Standard applied	→  [5]
Communication protocol data format	→  [2], [5]
Interface, physical	RS485
Data rate	≤12 MBaud (→  [2])
Node address	
Local (Adjustable via hexadecimal "ADDRESS", "MSD", "LSD" switches)	00 ... 7D <sub>hex</sub> (0 ... 125 <sub>dec</sub> )
Default setting	01 <sub>hex</sub>
Via Profibus (hexadecimal "ADDRESS" switches set to >7d <sub>hex</sub> (>125 <sub>dec</sub> ))	00 ... 7D <sub>hex</sub> (0 ... 125 <sub>dec</sub> )
Profibus connection	D-Sub, 9 pins, female
Cable	shielded, special Profibus cable (→  8 and  [4])
Cable length, system wiring	according to Profibus specifications (→  [5], [4])

## Dimensions [mm]



## 2 Power Connection

### Making a Profibus Interface Cable

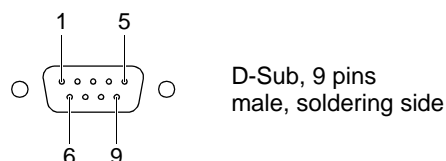
For operating the temperature compensated CDGxxxD gauge via Profibus, an interface cable conforming to the Profibus standard is required.  
If no such cable is available, make one according to the following indications.

Cable type

Only a cable that is suited to Profibus operation may be used (→ [4] and [5]).

Procedure

- 1 Make the Profibus interface cable according to the following indications:

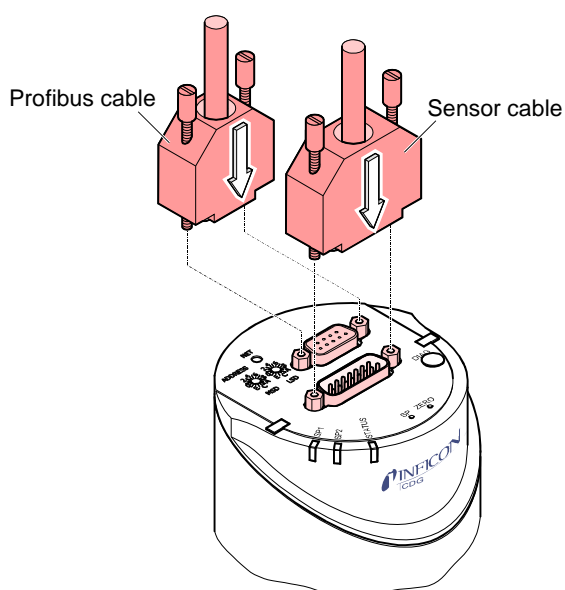


- Pin 1 Do not connect
- Pin 2 Do not connect
- Pin 3 RxD/TxD-P
- Pin 4 CNTR-P 1)
- Pin 5 DGND 2)
- Pin 6 VP 2)
- Pin 7 Not connected internally
- Pin 8 RxD/TxD-N
- Pin 9 Not connected internally

1) Only to be connected if an *optical link* module is used.

2) Only required as line termination for devices at both ends of bus cable (→ [4]).

- 2 Plug the Profibus (and sensor) cable connector into the gauge.



- 3 Lock the Profibus (and sensor) cable connector.

## 3 Operation

### 3.1 Introduction

Via the Profibus interface, the following and further data are exchanged in the standardized Profibus protocol:

- Pressure reading
- Pressure unit (Torr, mbar, Pa)
- Zero adjustment
- Status and error messages
- Status of the switching functions

**Caution**

Caution: data transmission errors

Any attempt to simultaneously operate the gauge via the RS232C Serial Interface and a Fieldbus interface (DeviceNet or Profibus) or the diagnostic port may result in incorrect data and data transmission errors.

Therefore, it is inadmissible to simultaneously operate the gauge via the RS232C Serial Interface and DeviceNet, Profibus, or the diagnostic port.

### 3.2 Operating Software

For operating the gauge via Profibus, prior installation of the CDGxxxD specific GSD file is required on the bus master side. This file can be downloaded from our website ([www.inficon.com](http://www.inficon.com)).

### 3.3 Node Address

The node address is a unique device address on a Profibus network. The gauge can only communicate with the network if its node address has been set properly. The valid address range is 0 ... 125 in decimal form.

In the CDGxxxD, the node address must be set in hexadecimal form (00 ... 7D<sub>hex</sub>). Two rotary switches at the back of the gauge are used for this:

ADDRESS	
	<p>The MSD switch is used to set the high-order address nibble</p> <p>The LSD switch is used to set the low-order address nibble</p> <p>The default node address setting is 01<sub>hex</sub>.</p>

The node address is polled by the firmware when the gauge is switched on. If the setting deviates from the stored value, the new value is taken over into the NVRAM.

If a value >7D<sub>hex</sub> (>125<sub>dec</sub>) is entered, the node address setting currently stored in the device remains valid. However, the address can now be set via the Profibus master with the "Set Slave Address" service. This address setting will be stored in the EEPROM of the gauge.


### 3.4 "NET" LED

THE "NET" LED indicates the network status of the CDGxxxD gauge:


LED	Status
Off	Device is off
Red	Error. The alarm bit in Exception Status is set. See section 7.1.11.
Green/flashing	Device is ok and uses acyclic data traffic. See section 5.1.
Green	Device is ok and uses cyclic data traffic. See section 5.2.

## 4 General Data

### 4.1 Data Rate

The gauge supports all data rates defined in the EN 50170 standard (→  [5]) up to 12 Mbaud. Automatic data rate setting is supported. Alternatively, a fixed data rate can be selected.

### 4.2 Ident Number

The ident number assigned to the gauge by the PNO (→  [4]) is:

Gauge	Ident number (hexadecimal)
CDGxxxD	0B7E

### 4.3 Configuration Data

Depending on the standard telegrams used (→ section “Cyclic Message Telegrams”), the following configuration data have to be transmitted to the gauge during the configuration phase:


M ⇌ S		S ⇌ M							Configuration data
ParamChannel	ParamChannel	Exception Status	OneOfN Status Extension	OneOfN PV Selector	Process Value UINT16	Process Value float	Channel Status	Trip Relay 1-2	
		x			x				0x42,0x82,0x05,0x03
		x				x			0x42,0x84,0x05,0x08
		x	x	x	x				0x44,0x84,0x05,0x05,0x05,0x03
		x	x	x		x			0x44,0x86,0x05,0x05,0x05,0x08
x	x	x			x				0xC4,0x87,0x8A,0x0A,0x0A,0x05,0x03
x	x	x				x			0xC4,0x87,0x8C,0x0A,0x0A,0x05,0x08
x	x	x	x	x	x				0xC6,0x87,0x8C,0x0A,0x0A,0x05,0x05,0x05,0x03
x	x	x	x	x		x			0xC6,0x87,0x8E,0x0A,0x0A,0x05,0x05,0x05,0x08
		x			x		x	x	0x44,0x85,0x05,0x03,0x05,0x0A
		x				x	x	x	0x44,0x87,0x05,0x08,0x05,0x0A
x	x	x			x		x	x	0xC6,0x87,0x8D,0x0A,0x0A,0x05,0x03,0x05,0x0A
x	x	x				x	x	x	0xC6,0x87,0x8D,0x0A,0x0A,0x05,0x03,0x05,0x0A

## 4.4 User Parameter Data

Depending on the pressure unit setting ( $\hat{=}$  data unit), the following configuration string has to be transmitted to the gauge (parameter data in hexadecimal format):

Data string (hex)	Pressure unit
00 00 00	Unit according to ID22 of the "Sensor Analog Input Function Block". See sections 8.1.7 and 8.2.5.
00 00 00 10 01	COUNTS
00 00 00 13 01	Torr
00 00 00 13 08	mbar
00 00 00 13 09	Pascal

## 4.5 Types of Communication

CDGxxxD works according to the Profibus DPV1 specification and can be addressed in cyclic or acyclic data traffic (→  [4]).

Acyclic data traffic should be used to make device or process specific settings such as definition of the Safe Values, Safe States etc. or for reading or writing of rarely used attributes.

Cyclic data traffic is used for continuous exchange of the required process parameter values, i.e. pressure value and status indications. A number of standard telegrams are available for cyclic data traffic. They can be selected according to requirements (→ section "Cyclic Message Telegrams").

## 5 Data Exchange Mode

### 5.1 Acyclic Data Transmission with Profibus DPV1 Functionality

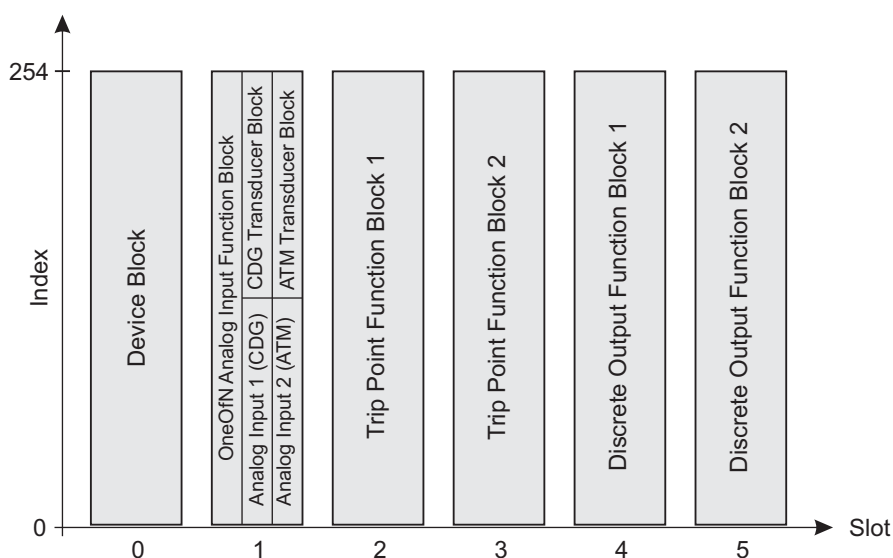
The reading and writing operations defined in Profibus are based on a slot index address scheme. In CDGxxxD, all device functions are organized in the following blocks:

- **Device block.**  
Describes all organizational parameters of the gauge (serial number, manufacturer, software version, ...)
- **OneOfN Analog Input Function Block.**  
Used to determine which function/transducer block parameter set is mapped into the corresponding block address space.
- **Sensor Analog Input Function Block.**  
Describes the function of the pressure presentation.
- **Transducer Block.**  
Describes the physical interface between the gauge and the process.
- **Trip Point Function Block.**  
Used to model the action of the trip point relays.
- **Discrete Output Function Block.**  
Used to control the digital outputs (Trip Function Relays).

Each block is assigned to a separate slot as shown in the table below.

Slot ID	Block	
	Selector = 1	Selector = 2
0	Device Block	
1	OneOfN Analog Input Function Block	
	Analog Input Function Block 1 (CDG)	Analog Input Function Block 2 (ATM)
	CDG Transducer Block	ATM Transducer Block
2	Trip Point Function Block 1	
3	Trip Point Function Block 2	
4	Discrete Output Function Block 1	
5	Discrete Output Function Block 2	

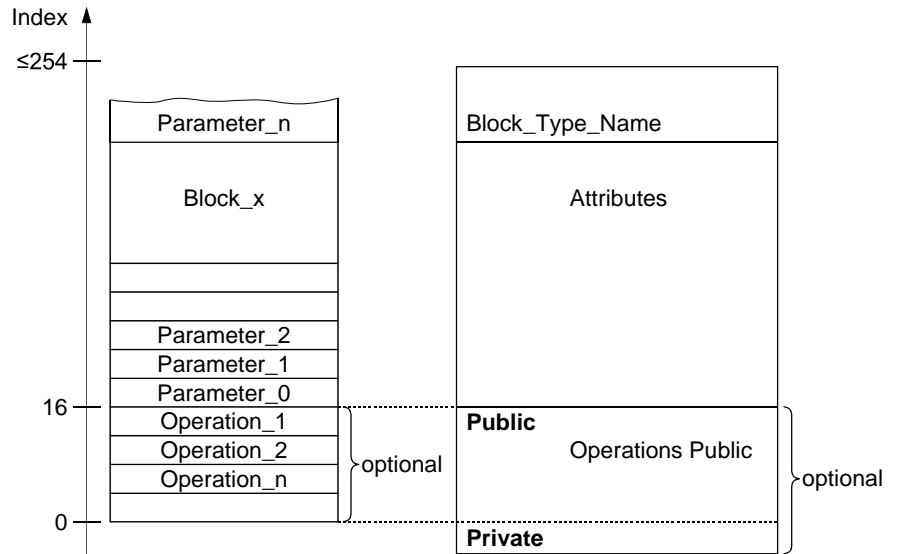
Block, slot and index assignment



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



## 5.2 Cyclic Data Telegrams in Data Exchange Mode

Within the DataExchange mode the DP-master class 1 cyclically transmits and requests data from all configured slaves in the network. This data transfer aims at the fast propagation of measurement data and command data within the system.

The next two figures contain the input and output data of the Profibus device. The data are divided between a parameter and a process data part.

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 5.3 “Parameter channel”.

The process data contain the measurement and command data as described in section 5.4 “Process data”.

Input data

The Input Area (DxIn) as sent in the DataExchange telegram consists of the 8 byte parameter channel and up to 7 byte of process data. The actual amount of process data is determined by the board configuration of the basic device.

Octets 0...7	Octets 8...15
Parameter channel	Process data

Output data

The Output Area (DxOut) as received in the DataExchange telegram contains the 8 byte parameter channel and 7 control bytes in the process part of the telegram.

Octets 0...7	Octets 8...15
Parameter channel	Process data

## 5.3 Parameter Channel

The parameter channel was created as a means to transmit acyclic requests within a cyclic data stream. Within this section a brief introduction into the matter is provided. The following table exhibits the principle structure of the parameter channel.

Octets							Octets							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
PKE		IND	res.	PWE			Standard telegram							

PKE = parameter signature value  
 IND = index within a slot  
 res. = reserved  
 PWE = process value  
 Standard telegram = cyclic telegram

### 5.3.1 PKW

The PKW 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 can't 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 may be transmitted within one instruction or one response. The instructions / responses are coded in the parameter signature word (PKE):

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

AK = Instruction/response signature  
 res. = reserved  
 PNU = Parameter number (bits 10...8 are reserved,  
 bits 7 ... 0 code the slot number of the function block)

### 5.3.2 Instruction and response processing

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

Instruction signature	Function	Response signature	
		Positive	Negative
0	No instruction	0	-
1	Parameter value read	1,2,11	7
2	Parameter value write (word)	1	7
3	Parameter value write (double word)	2	7
10	Parameter value write (byte)	11	7

The right column relates to the tables below and contains the responses according to the 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. At one time only one instruction will be processed.

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 8, the transfer of double words will be done with octet 5...8.

Response signature

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

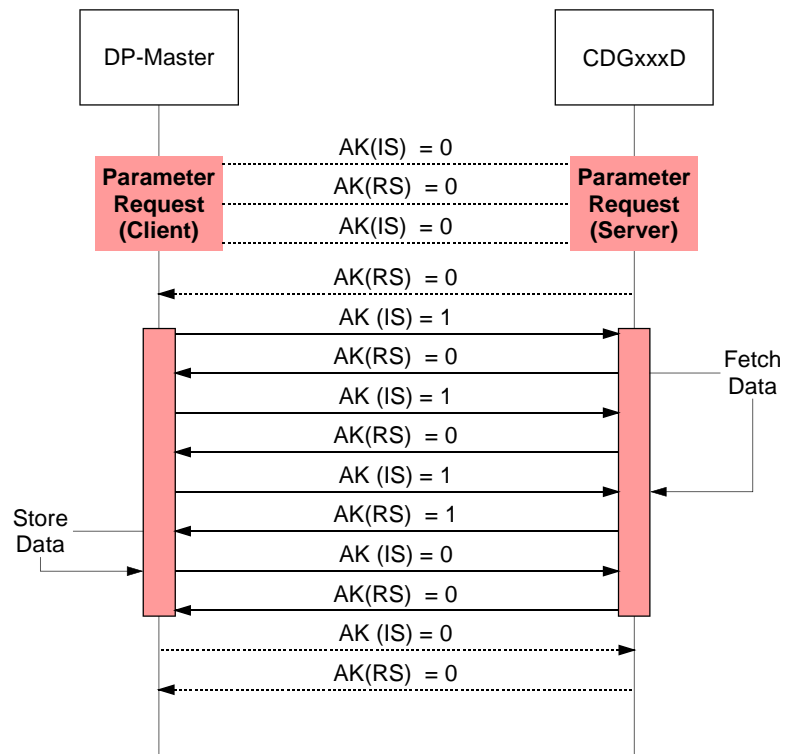
Error code

The slave sends an error message on octet 7 and octet 8 (interpreted as INT16) in case a request cannot be fulfilled.

Error code	Meaning
0	Undefined slot
1	Parameter not changeable
2	Lower or upper value range limit overflow
3	Subindex error
4	No array
5	Data type error
6	Setting not allowed (only resettable)
7	Description element not changeable
8	Reserved
9	Reserved
10	Access group error
11	No operation sovereignty
12	Password error
13	Text not readable in cyclic data transfer
14	Name not readable in cyclic data transfer
15	No text array existent
16	Reserved
17	Instruction not processable due to bad behavior state
18	Other errors
19	Data not readable in cyclic error
20...100	Reserved
201	Already in requested state
202	Object state conflict

### 5.3.3 Sequence Diagram

The following sequence diagram contains an example parameter request by the cyclic DP master in order to illustrate the application of parameter channel.



## 5.4 Process Data

The structure of the process data part of the input and output data area depends on the chosen message telegram. Applicable configurations are described below.

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 gauge can also be operated in such a way that the master does not transmit any output data to the slave.

### 5.4.1 Standard Telegrams

Standard telegram	Master ↔ Slave	Bytes	Meaning
2	M ⇒ S	0...7	Parameter Channel
1	S ⇒ M	0 1...2	Exception Status Process Value UINT16
2	S ⇒ M	0 1...4	Exception Status Process Value float
3	S ⇒ M	0 1 2 3...4	Exception Status OneOfN Status Extension OneOfN PV Selector Process Value UINT16
4	S ⇒ M	0 1 2 3...6	Exception Status OneOfN Status Extension OneOfN PV Selector Process Value float
7	S ⇒ M	0...7 8 9...10	Parameter Channel Exception Status Process Value UINT16
8	S ⇒ M	0...7 8 9...12	Parameter Channel Exception Status Process Value float
9	S ⇒ M	0...7 8 9 10 11...12	Parameter Channel Exception Status OneOfN Status Extension OneOfN PV Selector Process Value UINT16
10	S ⇒ M	0...7 8 9 10 11...14	Parameter Channel Exception Status OneOfN Status Extension OneOfN PV Selector Process Value float

## 5.4.2 Proprietary Telegrams

Proprietary telegram	Master ↔ Slave	Bytes	Meaning
201	S ⇒ M	0	Exception Status
		1...2	Process Value UINT16
		3	Channel Status
		4...5	Trip Relay 1-2
202	S ⇒ M	0	Exception Status
		1...4	Process Value float
		5	Channel Status
		6...7	Trip Relay 1-2
203	S ⇒ M	0...7	Parameter Channel
		8	Exception Status
		9...10	Process Value UINT16
		11	Channel Status
		12...13	Trip Relay 1-2
204	S ⇒ M	0...7	Parameter Channel
		8	Exception Status
		9...12	Process Value UINT16
		13	Channel Status
		14...15	Trip Relay 1-2

## 5.4.3 Configuration Data

The devices support the following standard telegrams:

Standard telegram M ⇒ S	Standard telegram S ⇒ M	Configuration data
-	1	0x42,0x82,0x05,0x03
-	2	0x42,0x84,0x05,0x08
-	3	0x44,0x84,0x05,0x05,0x05,0x03
-	4	0x44,0x86,0x05,0x05,0x05,0x08
2	7	0xC4,0x87,0x8A,0x0A,0x0A,0x05,0x03
2	8	0xC4,0x87,0x8C,0x0A,0x0A,0x05,0x08
2	9	0xC6,0x87,0x8C,0x0A,0x0A,0x05,0x05,0x05,0x03
2	10	0xC6,0x87,0x8E,0x0A,0x0A,0x05,0x05,0x05,0x08
-	201	0x44,0x85,0x05,0x03,0x05,0x0A
-	202	0x44,0x87,0x05,0x08,0x05,0x0A
2	203	0xC6,0x87,0x8D,0x0A,0x0A,0x05,0x03,0x05,0x0A
2	204	0xC6,0x87,0x8D,0x0A,0x0A,0x05,0x03,0x05,0x0A

## 6 Block Model

Data to the CDGxxxD 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. The following communication protocols are defined according to the Profibus DPV1 standard.

MS0	Cyclic data traffic between master class 1 and slave
MS1	Acyclic data traffic between master class 1 and slave
MS2	Acyclic data traffic between master class 2 and slave

In the CDGxxxD, all functions that are made available by the gauge 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.

### 6.1 Block types

The following block types are defined in the CDGxxxD gauge:

Device Block	The Device Block contains all data that are required for describing the device and handling its state (status of Device State Machine). See chapter 7.
Function Block	<p>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.</p> <ul style="list-style-type: none"> <li>• Analog Input Function Block (Instance 1, CDG). See section 8.1.</li> <li>• Analog Input Function Block (Instance 2, ATM). See section 8.2.</li> <li>• OneofN Analog Input Function Block. See chapter 9.</li> <li>• Discrete Output Function Block (Relay 1 and Relay 2). See chapter 10.</li> <li>• Trip Point Function Block (Relay 1 and Relay 2). See chapter 11.</li> </ul>
Transducer Block	<p>The physical, process specific functions or interfaces between the CDGxxxD and the process are represented in transducer blocks.</p> <p>The following transducer blocks are implemented:</p> <ul style="list-style-type: none"> <li>• Capacitance Diaphragm Gauge (CDG) Transducer Block. See section 12.1.</li> <li>• Atmosphere Pressure Sensor (ATM) Transducer Block. See section 12.2.</li> </ul>

## 7 Device Block

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

ID	Name	Structure	Data type	Bytes	Access	Store	See section
15	Device Block State	Simple	Unsigned8	1	1_R/W 2_R/W	N	7.2.2
16	Block Type ID	Simple	Octet string	4	1_R 2_R	N	7.1.1
17	Device Type	Simple	Visible string	8	2_R	N	7.1.2
18	Standard Revision Level	Simple	Visible string	9	2_R	N	7.1.3
19	Device Manufacturer Identifier	Simple	Visible string	20	2_R	N	7.1.4
20	Manufacturer Model Number	Simple	Visible string	20	2_R	N	7.1.5
21	Software or Firmware Revision Level (Profibus Adapter)	Simple	Visible string	8	2_R	N	7.1.6
22	Hardware Revision Level (Profibus Adapter)	Simple	Visible string	8	2_R	N	7.1.7
23	Serial Number	Simple	Visible string	30	2_R	N	7.1.8
24	Device Configuration	Simple	Visible string	50	2_R	N	7.1.9
25	Device State	Simple	Unsigned8	1	0_XI 1_R 2_R	V	7.1.10
26	Exception Status	Simple	Unsigned8	1	0_XI 1_R 2_R	V	7.1.11
27	Exception Detail Alarm	Record	→ below	-	1_R 2_R	V	7.1.12
28	Exception Detail Warning	Record	→ below	-	1_R 2_R	V	7.1.13
36	Run hours	Simple	Unsigned16	2	1_R 2_R	V	7.1.14
202	Basic Device Firmware Revision Level	Simple	Visible string	8	2_R	N	7.1.15
203	Basic Device Hardware Revision Level	Simple	Visible string	8	2_R	N	7.1.16
204	Common Exception Detail Alarm	Array	UINT8	2	1_R 2_R	V	7.1.17
205	Device Exception Detail Alarm	Array	UINT8	2	1_R 2_R	V	7.1.18
206	Manufacturer Exception Detail Alarm	Simple	UNIT8	1	1_R 2_R	V	7.1.19
207	Common Exception Detail Warning	Array	UINT8	2	1_R 2_R	V	7.1.20
208	Device Exception Detail Warning	Array	UINT8	3	1_R 2_R	V	7.1.21
209	Manufacturer Exception Detail Warning	Simple	UINT8	1	1_R 2_R	V	7.1.22

## 7.1 Information on the individual Indices

### 7.1.1 Block Type ID (ID 16)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the “Device Block” is set to “1”. The other defined block types are listed in Appendix B.

### 7.1.2 Device Type (ID 17)

The Device Type parameter identifies the device type which is connected to the field bus via Profibus.

The Device Type of the CDGxxxD gauge is “VPG” which is an acronym for “Vacuum Pressure Gauge”.

### 7.1.3 Standard Revision Level (ID 18)

This parameter describes the version of the “Sensor/Actuator Network Specific Device Model” published by the SEMI<sup>®</sup> (Semiconductor Equipment and Materials International, California), according to which the profile of this device has been developed.

The fixed setting of this parameter is “E54-0997”.

### 7.1.4 Device Manufacturer Identifier (ID 19)

This parameter describes the manufacturer of the device. It is set to “INFICON AG”.

### 7.1.5 Manufacturer Model Number (ID 20)

This parameter provides the part number of the gauge (→ section “Validity”, 3).

### 7.1.6 Software or Firmware Revision Level (ID 21)

This describes the version of the Profibus firmware.

### 7.1.7 Hardware Revision Level (ID 22)

This parameter describes the version of the Profibus adapter hardware.

### 7.1.8 Serial Number (ID 23)

This parameter provides the serial number of the device.

### 7.1.9 Device Configuration (ID 24)

This parameter provides the current device configuration, e.g. “1000 Torr, 45 C”.

### 7.1.10 Device State (ID 25)

This parameter indicates the overall status of the gauge. Due to the structure of the Device State Machine, the following states are possible:

Parameter value	Status
0	Undefined
1	Self testing
2	Idle
3	Self test exception
4	Executing
5	Abort
6	Critical fault
7...50	Reserved by PNO
51...99	Device-specific
100...255	Manufacturer-specific

The device states are described in detail in section “Device Block, Device Behavior”.

### 7.1.11 Exception Status (ID 26)

The Exception Status describes the alarm and warning states of the gauge in an “Extended error output format”.

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, ROM, or EEPROM error.
- ALARM / Warning Device Specific      For device specific errors.
- ALARM / Warning Manufacturer Specific      For errors defined by the manufacturer that are not mentioned in the standard.

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	An error occurred while testing internal hardware, CPU, EPROM, EEPROM, RAM, communication, power supply or real-time system.
1	ALARM, device specific	The sensor alarm bytes (ID 103, CDG Transducer Block) are set to a value different from 0.
2	ALARM, manufacturer specific	An error occurred while testing the internal communication interface, or the communication interface error counter reached the alarm level.
3	-	-
4	WARNING, device common	A warning occurred while testing internal hardware, CPU, EPROM, EEPROM, RAM, communication, power supply or real-time system.
5	WARNING, device specific	The OneOfNStatus Extension (ID 120, OneOfN Vacuum Pressure Gauge Transducer) is set to a value different from 0 and the sensor warning bytes (ID 104 CDG Transducer Block) are set to a value different from 0.
6	WARNING, manufacturer specific	The internal communication interface warning counter reached the warning level.
7	Extended Format	Bit seven is always set because the extended format is used. Otherwise, the definition of Exception Status and Exception Details is vendor specific.

## 7.1.12 Exception Detail Alarm (ID 27)

If one of the bits 0...2 is set in the Exception Status, the current error can be read in the "Exception Detail Alarm" parameter. The "Exception Detail Alarm" parameter consists of a total of 8 bytes that inform on the error status of the gauge.

Due to the use of the extended error output format, these bytes have the following structure:

Byte no	Name	Description	Value
<b>Common Exception Detail Alarm</b>			
0	Size	Number of subsequent bytes used for description of the alarm	2
1	Detail 0	Error message (simple, unsigned8, 1 byte)	See table below
2	Detail 1	Error message (simple, unsigned8, 1 byte)	See table below
<b>Device Exception Detail Alarm</b>			
3	Size	Number of subsequent bytes used for description of the alarm	2
4	Detail	Error message (array, unsigned8, 2 bytes)	Sensor Alarm (ID 103) of CDG Transducer Block
<b>Manufacturer Exception Detail Alarm</b>			
6	Size	Number of subsequent bytes used for description of the alarm	1
7	Detail	Error message (simple, unsigned8, 1 byte)	See table below

### Common Exception Detail Alarm

Bit	Detail 0
0	Internal diagnostic exception
1	Microprocessor exception
2	EPROM exception
3	EPROM exception
4	RAM exception
5	Communications exception
6	Internal real-time exception
7	Calibration expiration

Bit	Detail 1
0	Power supply overcurrent
1	Reserved power supply
2	Power supply output voltage
3	Power supply input voltage
4	Routine maintenance due
5	Notify manufacturer
6	Reset exception
7	Reserved

### Manufacturer Exception Detail Alarm

Bit	Detail
0	Internal communication exception
1	Incompatible software
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

### 7.1.13 Exception Detail Warning (ID 28)

If one of bits 4...6 is set in the Exception Status, the current warning can be read in the parameter "Exception Detail Warning". The Exception Detail Warning parameter consists of a total of 9 bytes that inform on the error status of the gauge. Due to the use of the extended error output format, these bytes have the following structure:

Byte no	Name	Description	Value
<b>Common Exception Detail Warning</b>			
0	Size	Number of subsequent bytes used for description of the alarm	2
1	Detail 0	Error message (simple, unsigned8, 1 byte)	See table above
2	Detail 1	Error message (simple, unsigned8, 1 byte)	See table above
<b>Device Exception Detail Warning</b>			
3	Size	Number of subsequent bytes used for description of the alarm	3
4	Status extension	Status extension (simple, unsigned8, 1 byte)	Status extension (ID 102) of CDG Transducer Block
5	Detail	Error message (array, unsigned8, 2 bytes)	Sensor Alarm (ID 104) of CDG Transducer Block
<b>Manufacturer Exception Detail Warning</b>			
7	Size	Number of subsequent bytes used for description of the alarm	1
8	Detail	Error message (simple, unsigned8, 1 byte)	See table above

### 7.1.14 Run hours (ID 36)

This parameter identifies the number of hours that the basic device has been powered ON. The parameter has a resolution of 1 hour.

### 7.1.15 Basic Device Firmware Revision Level (ID 202)

This proprietary parameter describes the firmware version of the basic device.

### 7.1.16 Basic Device Hardware Revision Level (ID 203)

This proprietary parameter describes the version of the basic device hardware.

### 7.1.17 Common Exception Detail Alarm (ID 204)

This proprietary parameter allows access of the corresponding part (Detail 0 and Detail 1) of the Common Exception Detail Alarm parameter (ID 27). See section 7.1.12.

### 7.1.18 Device Exception Detail Alarm (ID 205)

This proprietary parameter allows access of the corresponding part (Detail) of the Device Exception Detail Alarm parameter (ID 27). See section 7.1.12.

### 7.1.19 Manufacturer Exception Detail Alarm (ID 206)

This proprietary parameter allows access of the corresponding part (Detail) of the Manufacturer Exception Detail Alarm parameter (ID 27). See section 7.1.12.

### 7.1.20 Common Exception Detail Warning (ID 207)

This proprietary parameter allows access of the corresponding part (Detail 0 and Detail 1) of the Common Exception Detail Warning parameter (ID 28). See section 7.1.13.

### 7.1.21 Device Exception Detail Warning (ID 208)

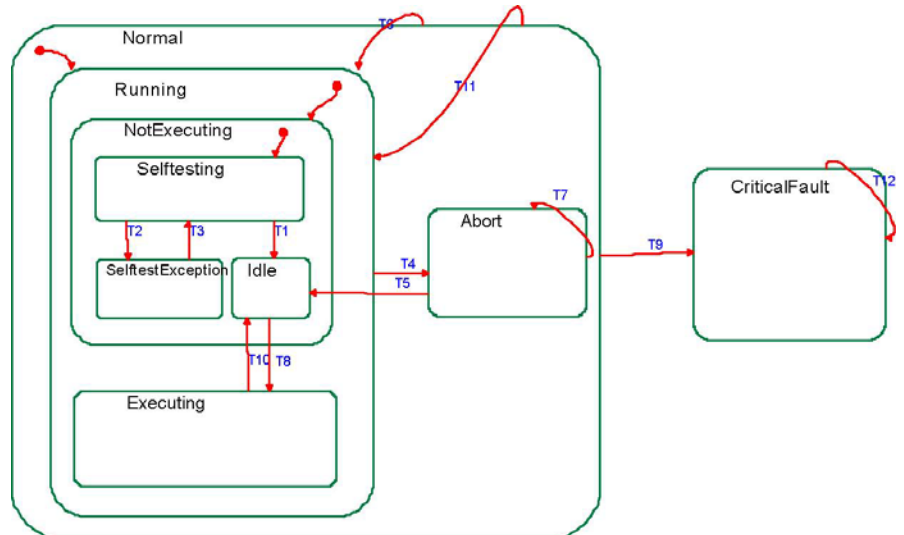
This proprietary parameter allows access of the corresponding part (Status Extension and Detail) of the Device Exception Detail Warning parameter (ID 28). See section 7.1.13.

### 7.1.22 Manufacturer Exception Detail Warning (ID 209)

This proprietary parameter allows access of the corresponding part (Detail) of the Manufacturer Exception Detail Warning parameter (ID 28). See section 7.1.13.

## 7.2 Device Block Status diagram

The CDGxxxD behaves as described in the status diagram below.



After the start, the gauge 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 gauge 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 gauge automatically changes to the EXECUTING status.

Acyclic data traffic

In acyclic data traffic, a START service has to be transmitted to bring the gauge to the EXECUTING status.

## 7.2.1 General Device Block State descriptions

Status name	Description
NORMAL (State set)	The firmware of the device has started. All configured block instances exist. No device internal initialisation and check tasks have been carried out yet.
RUNNING (State set)	This is the entry sub-state to NORMAL. All block instances are initialised. The parameters have appropriate initial or default values (as defined in this profile). Acyclic access to the device is possible (i.e. MS2 services and MS0 diagnosis service (get_diagnosis))
NOTEXECUTING (State set)	This is the entry sub-state to RUNNING. Device is not executing (e.g., it is not performing its device-specific function). No self tests have been carried out after a new start of the device.
SELFTESTING	This is the entry sub-state to NORMAL, RUNNING and NOTEXECUTING. All block instances exist and have been initialised. Device is performing device-specific and device type-specific tests to determine if it is qualified to be running.
IDLE	All blocks and device hardware and software have been initialised and have successfully completed self testing. The device is ready for cyclic data transfer.
SELFTESTEXCEPTION	Object has detected an exception condition during self testing. The details of the exception are stored in the appropriate parameter values of the Device Block.
EXECUTING	Device is executing (e.g., it is performing its device-specific function) its functions according to the purpose of the device. The detailed purposes are described in the certain blocks of the profile. The device performs cyclic data transfer.
ABORT	Device Block instance is in an aborted state. The device-specific functions are not performed properly. If the reason of the abort is not active, the device state can switch automatically back to IDLE state. The cyclic data transfer is not active.
CRITICALFAULT	The Device Block (and device) is in a fault state from which there is no recovery. The result of the device-specific function is bad. The conditions required for exit from a critical fault are outside the scope of this document. The cyclic data transfer is not active.

## 7.2.2 Device Block State Command (ID 15)

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

ID	Name	Structure	Data type	Bytes	Access	Store	Range
15	Device Block State Command	Simple	Unsigned8	1	1_R/W 2_R/W	N	→ below

The device block state method allows the master to force the behavior of the device block state machine (as shown in parameter device state) by setting it to one of the values shown in the table below. Depending of the actual state, some values may not be allowed at a time.

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 $\hat{=}$ 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.
7...127	Reserved by PNO	
128...255	Manufacturer specific	

## 8 Sensor Analog Input Function Blocks

All gauge functions of the CDGxxxD are described in the Analog Input Function Block. Because the gauge includes two measuring systems, there are two Analog Input Function Block Instances:

- Instance 1 represents the Capacitance Diaphragm Gauge (CDG) measuring part of the gauge. See section 8.1.
- Instance 2 represents the Atmosphere Pressure Sensor (ATM) measuring part of the gauge. See section 8.2.

### 8.1 Sensor Analog Input Function Block (Instance 1, CDG)

Instance 1 of the Analog Sensor Input Function Blocks describes the functionality of the Capacitance Diaphragm Gauge (CDG) measuring part of the gauge. The following attributes are supported:

ID	Name	Structure	Data type	Bytes	Access	Store	See section
15	AlBlockAdjust Command	Record	Unsigned8	2	1_R/W 2_R/W	N	8.1.22
16	Block Type ID	Simple	Octet string	4	1_R 2_R	N	8.1.1
17	AlarmEnable	Simple	Boolean	1	1_R/W 2_R/W	N	8.1.2
18	WarningEnable	Simple	Boolean	1	1_R/W 2_R/W	N	8.1.3
19	ProcessValue (PV)	Simple	*)	-	0_XI 1_R 2_R	D	8.1.4
20	Status	Simple	Unsigned8	1	0_XI 1_R 2_R	D	8.1.5
21	Data Type	Simple	Unsigned8	1	2_R/W	N	8.1.6
22	Data Unit	Simple	Unsigned16	2	2_R/W	N	8.1.7
23	Reading Valid	Simple	Boolean	1	1_R 2_R	D	8.1.8
24	Full Scale	Simple	**)	-	1_R 2_R	N	8.1.9
25	Offset-A Data Type	Simple	Unsigned8	1	1_R 2_R	N	8.1.10
26	Offset-A	Simple	**)	-	1_R 2_R		8.1.11
31	Alarm Trip Point High	Simple	**)	-	1_R/W 2_R/W	N	8.1.12
32	Alarm Trip Point Low	Simple	**)	-	1_R/W 2_R/W	N	8.1.13
33	Alarm Hysteresis	Simple	**)	-	1_R/W 2_R/W	N	8.1.14
35	Warning Trip Point High	Simple	**)	-	1_R/W 2_R/W	N	8.1.15
36	Warning Trip Point Low	Simple	**)	-	1_R/W 2_R/W	N	8.1.16
37	Warning Hysteresis	Simple	**)	-	1_R/W 2_R/W	N	8.1.17
39	Safe State	Simple	**)	1	1_R/W 2_R/W	N	8.1.18
40	Safe Value	Simple	**)	-	1_R/W 2_R/W	N	8.1.19
44	Overrange	Simple	**)	-	1_R 2_R	N	8.1.20
45	Underrange	Simple	**)	-	1_R 2_R	N	8.1.21

\*) According to parameter data type    \*\*) According to data type value (ID 21)

### 8.1.1 Block Type ID (ID 16)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the “Sensor Analog Input Function Block” is set to “2”. The other defined block types are listed in Appendix B.

### 8.1.2 AlarmEnable (ID17)

This parameter enables the setting of the Alarm Status Bits (ID 20):

- 0 = disable (Default)
- 1 = enable

### 8.1.3 WarningEnable (ID 18)

This parameter enables the setting of the Warning Status Bits (ID 20):

- 0 = disable (Default)
- 1 = enable

### 8.1.4 Process Value (ID 19)

The process value contains the measurement value of the Diaphragm Gauge transducer instance (CDG), transformed on the basis of 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). This could be the Safe Value, Last Value, Full Scale, Zero, device specific, or vendor specific.

### 8.1.5 Status (ID 20)

This parameter provides the Alarm and Warning State of this block instance according to the following table:

Bit	Definition
0	High Alarm Exception
1	Low Alarm Exception
2	High Warning Exception
3	Low Warning Exception
4	Reserved
5	Reserved
6	Reserved
7	Reserved

### 8.1.6 Data Type (ID 21)

This parameter determines the data type of ProcessValue and all related parameters. Two data types are supported: Float and Integer16.

This parameter can be changed via master class 2 acyclic data transfer only if the device is not in cyclic data transfer with master class 1. The data type is valid for all block instances of the device. This means that all parameters determining the data type will have the same value. A change of one data type parameter changes all others to the same value.

It is possible to adjust the data type during cyclic data transfer start up (setting of configuration data, CFG-String. In this case the data type value is updated with the value requested by the configuration data.

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

Coding	Data type
3	Integer16
8	Float

### 8.1.7 Data Unit (ID 22)

The device supports the four data units described below:

Coding (hex)	Unit
0	COUNTS
4865	Torr
4872	mbar
4873	Pascal



For safety reasons, it is not possible to change the pressure unit while the gauge is cyclically interchanging data with a DP/V0 master.

The data unit setting can only be modified when the gauge is in the IDLE status.

In cyclic data traffic, the data unit must be set in the User Parameter Data. All settings previously made in acyclic data traffic are overwritten (→ section “User Parameter Data”).

If the data unit is set in one instance, that data unit setting applies to all instances. Likewise, the data unit setting made in the User Parameter Data is valid for all instances.

### 8.1.8 Reading Valid (ID 23)

- This parameter indicates that the pressure reading is within a valid range. This means that the following conditions are fulfilled:
- The gauge is in the EXECUTING status (Device Block, ID 25)
- The exception status contains no manufacturer warning or alarm
- The transducer block contains no sensor alarm (ID 103)



If this value is set to zero, the pressure reading is not valid. In such a case, either check Exception Status (ID 26, Device Block) to find out whether there is an error or check the Status Extension (ID 102, Transducer Block) to find out whether the measured value is out of the specified measuring range (overrange or underrange mode).

### 8.1.9 Full Scale (ID 24)

This parameter contains the valid maximum measurement scale of the device in terms of the currently selected data type (ID 21) and data unit (ID 22).

### 8.1.10 Offset-A Data Type (ID 25)

This parameter determines the data type of parameter Offset-A (ID 26).

### 8.1.11 Offset-A (ID 26)

This parameter specifies an amount that is added prior to Gain to derive the ProcessValue.

### 8.1.12 Alarm Trip Point High (ID 31)

This parameter determines the value above which an Alarm Condition will occur. The value must always lie within the boundaries specified for the Safe Value (ID 40). See section 8.1.19.

### 8.1.13 Alarm Trip Point Low (ID 32)

This parameter determines the value below which an Alarm Condition will occur. The value must always lie within the boundaries specified for the Safe Value (ID 40). See section 8.1.19.

### 8.1.14 Alarm Hysteresis (ID 33)

This parameter determines the amount by which the ProcessValue must recover to clear an Alarm Condition. The value must always lie within the boundaries specified for the Safe Value (ID 40). See section 8.1.19.

### 8.1.15 Warning Trip Point High (ID 35)

This parameter determines the Value above which a Warning Condition will occur. The value must always lie within the boundaries specified for the Safe Value (ID 40). See section 8.1.19.

### 8.1.16 Warning Trip Point Low (ID 36)

This parameter determines the Value below which a Warning Condition will occur. The value must always lie within the boundaries specified for the Safe Value (ID 40). See section 8.1.19.

### 8.1.17 Warning Hysteresis (ID 37)

This parameter determines the amount by which the ProcessValue must recover to clear a Warning Condition. The value must always lie within the boundaries specified for the Safe Value (ID 40). See section 8.1.19.

### 8.1.18 Safe State (ID 39)

When the gauge 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 pressure value. You can select among the following Safe State values:

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

### 8.1.19 Safe Value (ID 40)

This parameter specifies the Value to be used if Safe State (ID 39) is set to the Safe Value. The value may be set to any value with respect to the chosen data type and the maximum and minimum values defined below. The data unit conversions of PV apply except for the value "0". To avoid calculation errors and/or underflow and overflow conditions when switching between different data units, the allowable value range for the Safe Value has to be limited to the following values:

Unit	Minimum	Maximum
COUNTS	0	97954
Torr	2.3719E-13	2.2502E+36
mBar	3.1622E-13	3E+36
Pascal	3.1622E-11	3E+38

In addition, "0" is valid for all units. The following rule applies for conversions:

- 0 Pa = 0 mBar = 0 Torr = 0 COUNTS

It is possible that the Safe Value lies outside the measuring range of the sensor.

### 8.1.20 Overrange (ID 44)

This parameter contains the highest valid Process Value (PV) of the device in terms of the currently selected data type (ID 21) and data unit (ID 22).

### 8.1.21 Underrange (ID 45)

This parameter contains the lowest valid Process Value (PV) of the device in terms of the currently selected data type (ID 21) and data unit (ID 22).

### 8.1.22 Analog Input Block Adjust Command (ID 15)

ID	Name	Structure	Data Type	Bytes	Access	Store
15	AIBlockAdjust Command	Record	Unsigned8	2	1_R/W 2_R/W	N

Byte	Name	Structure	Data Type	Bytes	Access	Store
0	Adjust Command	Simple	UINT8	1	1_R/W 2_R/W	N
1	Target Value	Simple	*)	1	1_R/W 2_R/W	N

\*) According to data type value (ID 21)

Adjust Command	Name	Description
0	Zero Adjust	Used to calibrate diaphragm zero pressure value using a target value of actual pressure in current data type (ID 21). This service modifies parameter Offset-A (ID 26).
1...255	Reserved	Reserved, no action

Target value	Description
0	No action
1	Perform requested service
2...255	Reserved, no action

## 8.2 Sensor Analog Input Function Block (Instance 2, ATM)

Instance 2 of the Analog Sensor Input Function Blocks describes the functionality of the Atmosphere Pressure Sensor (ATM) measuring part of the gauge.

The following attributes are supported:

ID	Name	Structure	Data type	Bytes	Access	Store	See section
15	AIBlockAdjust Command	Record	Unsigned8	2	1_R/W 2_R/W	N	8.2.12
16	Block Type ID	Simple	Octet string	4	1_R 2_R	N	8.2.1
19	ProcessValue (PV)	Simple	*)	-	0_XI 1_R 2_R	D	8.2.2
20	Status	Simple	Unsigned8	1	0_XI 1_R 2_R	D	8.2.3
21	Data Type	Simple	Unsigned8	1	2_R/W	N	8.2.4
22	Data Unit	Simple	Unsigned16	2	2_R/W	N	8.2.5
23	Reading Valid	Simple	Boolean	1	1_R 2_R	D	8.2.6
24	Full Scale	Simple	**)	-	1_R 2_R	N	8.2.7
39	Safe State	Simple	**)	1	1_R/W 2_R/W	N	8.2.8
40	Safe Value	Simple	**)	-	1_R/W 2_R/W	N	8.2.9
44	Overrange	Simple	**)	-	1_R 2_R	N	8.2.10
45	Underrange	Simple	**)	-	1_R 2_R	N	8.2.11

\*) According to parameter data type

\*\*\*) According to data type value (ID 21)

### 8.2.1 Block Type ID (ID 16)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the "Sensor Analog Input Function Block" is set to "2". The other defined block types are listed in Appendix B.

### 8.2.2 Process Value (ID 19)

The process value contains the measurement value of the Atmosphere Pressure Sensor (ATM), transformed on the basis of 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). This could be the Safe Value, Last Value, Full Scale, Zero, device specific, or vendor specific.

### 8.2.3 Status (ID 20)

The atmosphere pressure sensor cannot trigger alarms and warnings because the atmosphere pressure changes are too small. Therefore the ATM sensor supports no alarm/warning trip points and consequently this parameter is always set to "0".

### 8.2.4 Data Type (ID 21)

This parameter determines the data type of ProcessValue and all related parameters. Two data types are supported: Float and Integer16.

This parameter can be changed via master class 2 acyclic data transfer only if the device is not in cyclic data transfer with master class 1. The data type is valid for all block instances of the device. This means that all parameters determining the data type will have the same value. A change of one data type parameter changes all others to the same value.

It is possible to adjust the data type during cyclic data transfer start up (setting of configuration data, CFG-String). In this case the data type value is updated with the value requested by the configuration data.

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

Coding	Data type
3	Integer16
8	Float

## 8.2.5 Data Unit (ID 22)

The device supports the four data units described below:

Coding (hex)	Unit
0	COUNTS
4865	Torr
4872	mbar
4873	Pascal



For safety reasons, it is not possible to change the pressure unit while the gauge is cyclically interchanging data with a DP/V0 master.

The data unit setting can only be modified when the gauge is in the IDLE status.

In cyclic data traffic, the data unit must be set in the User Parameter Data. All settings previously made in acyclic data traffic are overwritten (→ section "User Parameter Data").

If the data unit is set in one instance, that data unit setting applies to all instances. Likewise, the data unit setting made in the User Parameter Data is valid for all instances.

## 8.2.6 Reading Valid (ID 23)

This parameter indicates that the pressure reading is within a valid range. This means that the following conditions are fulfilled:

- The gauge is in the EXECUTING status (Device Block, ID 25)
- The exception status contains no manufacturer warning or alarm
- The transducer block contains no sensor alarm (ID 103)



If this value is set to zero, the pressure reading is not valid. In such a case, check the Status Extension (ID 102, Transducer Block) to find out whether the measured value is out of the specified measuring range (overrange or underrange mode).

## 8.2.7 Full Scale (ID 24)

This parameter contains the valid maximum measurement scale of the device in terms of the currently selected data type (ID 21) and data unit (ID 22).

## 8.2.8 Safe State (ID 39)

When the gauge 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 pressure value. You can select among the following Safe State values:

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

### 8.2.9 Safe Value (ID 40)

This parameter specifies the Value to be used if Safe State (ID 39) is set to the Safe Value. The value may be set to any value with respect to the chosen data type and the maximum and minimum values defined below. The data unit conversions of PV apply except for the value "0". To avoid calculation errors and/or underflow and overflow conditions when switching between different data units, the allowable value range for the Safe Value has to be limited to the following values:

Unit	Minimum	Maximum
COUNTS	0	97954
Torr	2.3719E-13	2.2502E+36
mBar	3.1622E-13	3E+36
Pascal	3.1622E-11	3E+38

In addition, "0" is valid for all units. The following rule applies for conversions:

- 0 Pa = 0 mBar = 0 Torr = 0 COUNTS

It is possible that the Safe Value lies outside the measuring range of the sensor.

### 8.2.10 Overrange (ID 44)

This parameter contains the highest valid Process Value (PV) of the device in terms of the currently selected data type (ID 21) and data unit (ID 22).

### 8.2.11 Underrange (ID 45)

This parameter contains the lowest valid Process Value (PV) of the device in terms of the currently selected data type (ID 21) and data unit (ID 22).

### 8.2.12 Analog Input Block Adjust Command (ID 15)

ID	Name	Structure	Data Type	Bytes	Access	Store
15	AIBlockAdjust Command	Record	Unsigned8	2	1_R/W 2_R/W	N

Byte	Name	Structure	Data Type	Bytes	Access	Store
0	Adjust Command	Simple	UINT8	1	1_R/W 2_R/W	N
1	Target Value	Simple	*)	1	1_R/W 2_R/W	N

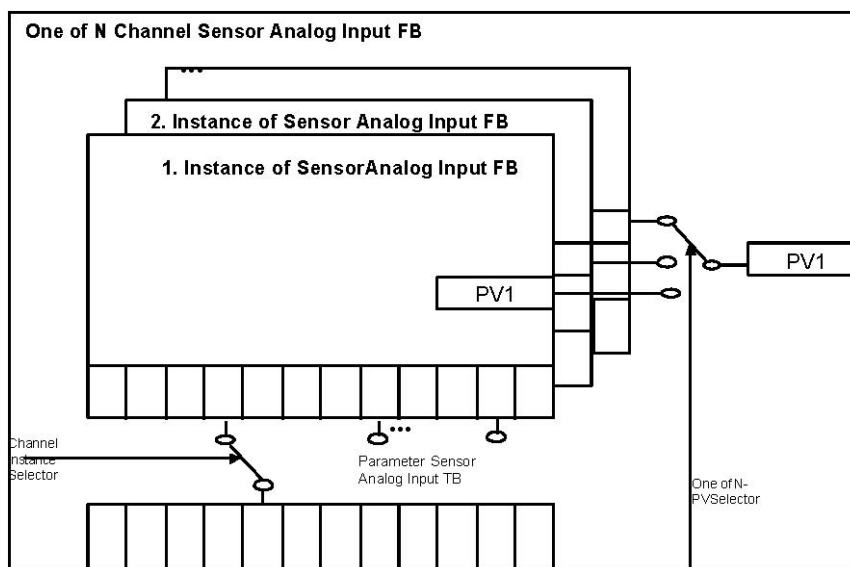
\*) According to data type value (ID 21)

Adjust Command	Name	Description
0..255	Reserved	Reserved, no action

Target value	Description
0..255	Reserved, no action

## 9 OneOfN Analog Input Function Block

The figure below shows the basic design of the analog input function block.



Instance 2 of the Analog Sensor Input Function Blocks describes the functionality of the Atmosphere Pressure Sensor (ATM) measuring part of the gauge.

The following attributes are supported:

ID	Name	Structure	Data type	Bytes	Access	Store	See section
15	Block behavior	Record	Unsigned8	2	1_R/W 2_R/W	N	9.1.1
16	Block Type ID	Simple	Octet string	4	2_R	N	9.1.2
17...45	-	-	-	-	-	-	9.1.3
46	Channel Instance Selector	Simple	Unsigned8	1	1_R/W 2_R/W	N	9.1.4

### 9.1 Information on the Individual Indices

#### 9.1.1 Block behavior (ID 15)

The “One of N Channel Sensor Analog Input Function Block” behavior is identical with the “Sensor Analog Input Function Block” behavior. See section 8.1.22.

#### 9.1.2 Block Type ID (ID 16)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the “One of N Channel Sensor Analog Input Function Block” is set to “3”. The other defined block types are listed in Appendix B.

#### 9.1.3 Parameters 17...45

The parameters 17...45 of the One of N Channel Sensor Analog Input Function Block are identical with the parameters 17...45 of the Sensor Analog Input Function Block. See section 8.1.

#### 9.1.4 Channel Instance Selector (ID 46)

Provides access to the selected parameter instances via the function block parameter interface. The channel instance number starts with 1. The selector determines which function/transducer block parameter set is mapped into the “OneOfN Analog Input Function Block” or “OneOfN Vacuum Gauge Transducer Block” address space.

## 10 Discrete Output Function Block (Relay 1 and 2)

The Discrete Output Function Block is used for switching the relays on and off. The following attributes are supported:

ID	Name	Structure	Data Type	Bytes	Access	Store	See section
15	DOBlockState Command	Simple	Unsigned8	1	1_R/W 2_R/W	N	
16	Block Type ID	Simple	Octet string	1	1_R 2_R	N	10.1.1
19	Setpoint Value	Simple	Unsigned8	1	0_XO 2_W	D	10.1.2
20	Status	Simple	Boolean	1	2_R	D	10.1.3

### 10.1 Information on the Individual Indices

#### 10.1.1 Block Type ID (ID 16)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the "Discrete Output Function Block" is set to "7". The other defined block types are listed in Appendix B.

#### 10.1.2 Setpoint Value (ID 19)

This parameter contains the value which has to be forwarded to the Relay 1:

Setpoint value	Relay
0	Off
1	On
2...255	Off

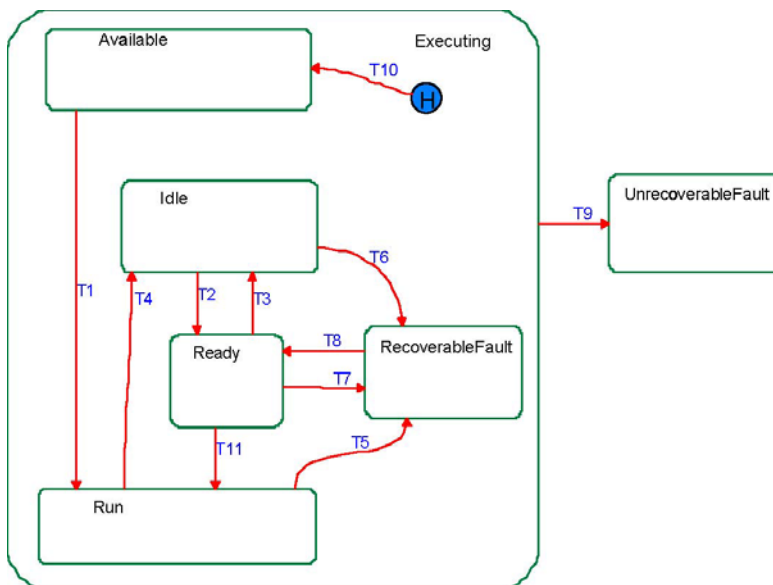
#### 10.1.3 Status (ID 20)

This parameter indicates the state of the Discrete Output Function Block Instance.

Status	Description
0	Ok
1	Error or alarm

## 10.2 Discrete Output Function Block behavior

The figure below shows the state chart of the discrete output function block.



State name	Description
Available	This is the initial state after power on. The Device Block is in Executing state (see section 7.2.1).
Idle	This state cuts the signal flow between the SetpointValue of the certain Discrete Output function block instance and its dedicated underlying actuation hardware. The behavior is according the IdleActionSelector parameter.
Ready	The connection between the SetpointValue parameter of the certain Discrete Output function block instance and its dedicated underlying actuation hardware is recovered. The instance is waiting for new SetpointValue parameter values.
Run	The instance is working in its dedicated way.
Recoverable Fault	There is a temporary fault. The reason is device and manufacturer specific.
Unrecoverable Fault	There is a fault with no automatic return. This fault can force a Critical Fault according to section 7.2.1 or refers to the certain instance only.
Executing	This state is defined in section 7.2.1, i.e. the state machine is working in this device state only.

### 10.2.1 Block State Command Mapping

ID	Name	Structure	Data Type	Bytes	Access	Store
15	DOBlockState Command	Simple	Unsigned8	1	1_R/W 2_R/W	N

A Write service performed to this parameter is interpreted as event to switch the block state according to the state chart.

Parameter value	State
0	Reserved
1	Available
2	Idle
3	Ready
4	Run
5	Recoverable fault
6	Unrecoverable fault
7...127	Reserved
128...255	Manufacturer specific

## 11 Trip Point Function Block (Relay 1 and 2)

The Trip Point Object models the action of trip points for a device, often corresponding to physical outputs (Discrete Output Object). Each Trip Point channel has a pointer (Analog Sensor Input Object Instance ID) to a source Analog Sensor Input Instance.

A trip point value, designated as a High or Low trip point, is compared to the specified instance of an Analog Input Sensor Process Value parameter. This trip point is intended to be used as a process control indicator only, as distinguished from the Analog Input Sensor Function Block Object's Alarm and Warning trip points.

All the necessary calculations and the setting/resetting (triggering) of the relays are done by the basic device itself.

### 11.1 Trip Point Function Block behaviors

Two different functionalities are implemented: Setpoint function and Atmosphere (ATM) detection. The Setpoint function is realized via the Low trip Point function and the ATM detection via the High Trip point function. The two functions are mutually exclusive, only one of them can be active at a time.

The basic device switches to the desired function according to the value „Percentage of ATM“ (ID 201). See section 11.2.11.

#### 11.1.1 Setpoint function

If “Low Trip Enable” (ID 20) is set, the Setpoint function is activated.

The parameter “Percentage of ATM” (ID 201) is automatically set to “0” and therefore the ATM function is disabled. Low Trip Point is compared to the Analog Input Sensor Function Block Instance 1 (CDG) ProcessValue to generate a trip point condition.

Status will be set if the input (ProcessValue) is at or below the Low Trip Point. If the pressure (input) increases above “Low Trip Point + Hysteresis”, the Status will be reset.

#### 11.1.2 Atmosphere detection

This functionality is used to compare the pressure measured by the CDG with the atmospheric pressure.

The parameter “Percentage of ATM” (ID 201) is used to define a value “Percentage of Atmosphere”. This value will be multiplied with the pressure from the Analog Sensor Instance 2 (ATM Sensor). Then this result will be copied into parameter “High Trip Point” (ID 202). If a value other than “0” is written to Percentage of Atmosphere (ID 201), the parameter “Low Trip Enable” (ID 20) is automatically set to “0”. High Trip Point is compared to the Analog Input Sensor Function Block Instance 2 (ATM) ProcessValue to generate a trip point condition.

Status will be set if the input (ProcessValue) value is at or above the High Trip Point. If the pressure (input) decreases below “High Trip Point – Hysteresis”, the Status will be reset.

## 11.2 Trip Point Function Block (Relay 1 and Relay 2)

The following attributes are supported:

ID	Name	Structure	Data type	Bytes	Access	Store	See section
16	Block Type ID	Simple	Octet-String	4	1_R 2_R	N	11.2.1
19	Low Trip Point	Simple	According Data Type value (ID 29)	-	1_R/W 2_R/W	N	11.2.2
20	Low Trip Enable	Simple	Boolean	1	1_R/W 2_R/W	N	11.2.3
21	Status	Simple	Unsigned8	1	0_XI 1_R 2_R	D	11.2.4
24	Hysteresis	Simple	According Data Type value (ID 29)	-	1_R/W 2_R/W	N	11.2.5
26	Discrete Output Function Block Instance	Simple	Unsigned8	1	1_R/W 2_R/W	N	11.2.6
27	Analog Input Sensor Function Block Instance	Simple	Unsigned8	1	1_R/W 2_R/W	N	11.2.7
28	Data Units	Simple	Unsigned16	2	2_R	N	11.2.8
29	Data Type	Simple	Unsigned8	1	2_R	N	11.2.9
30	Channel Instance Selector	Simple	Unsigned8	1	1_R/W 2_R/W	N	11.2.10
201	Percentage of ATM	Simple	Float	4	1_R/W 2_R/W	N	11.2.11
202	High Trip Point	Simple	According Data Type value (ID 29)	-	1_R 2_R	N	11.2.12
203	High Trip Enable	Simple	Boolean	1	1_R 2_R	N	11.2.13

### 11.2.1 Block Type ID (ID 16)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the "Trip Point Function Block" is set to "17". The other defined block types are listed in Appendix B.

### 11.2.2 Low Trip Point (ID 19)

This parameter defines the value at or below which a trip point condition will occur.

### 11.2.3 Low Trip Enable (ID 20)

This parameter enables/disables the Low Trip Point setting.

Parameter value	Description
0	disable Low Trip Point setting
1	enable Low Trip Point setting

### 11.2.4 Status (ID 21)

This parameter contains the state of this object instance. The functionality is described in section 11.1.

Parameter value	Description
0	trip point condition does not exist (unasserted)
1	trip point condition exists (asserted)

### 11.2.5 Hysteresis (ID 24)

The Hysteresis value specifies the amount by which the Analog Input Sensor ProcessValue must recover in order to clear a trip point condition.

For example: A High Trip Point of 100 and a Hysteresis value of 2 will result in a trip point condition being set when the (Analog Input Sensor) ProcessValue goes above 100 and cleared when ProcessValue drops below 98.

Similarly, a Low Trip Point of 100 and a Hysteresis of 2 will result in a trip point condition being set when ProcessValue falls below 100 and cleared when ProcessValue increases above 102.

### 11.2.6 Discrete Output Function Block Instance (ID 26)

This parameter specifies the associated Discrete Output Object Instance.

Parameter value	Description
1	Relay 1
2	Relay 2

### 11.2.7 Analog Input Sensor Function Block Instance (ID 27)

This parameter specifies the Analog Sensor Object Instance from which Value is compared to High or Low Trip Point. This value is fix set to instance 1 (CDG) because the ATM sensor (instance 2) does not support trip points.

### 11.2.8 Data Units (ID 28)

This parameter defines the units of Trip Point, Hysteresis, etc. according to the following table:

Parameter value (hex)	Unit
0	COUNTS
4865	Torr
4872	mBar
4873	Pascal

### 11.2.9 Data Type (ID 29)

This parameter defines the data type of Trip Point, Hysteresis, etc. and all related parameters.

The parameter can be changed via master class 2 acyclic data transfer only if the device is not in cyclic data transfer with master class 1. The data type is valid for all block instances of the device i.e. all parameter determining the data type shall have the same value. A change of one data type parameter changes all other parameters to the same value.

Parameter value	Type
3	Integer16
8	Float (Default)

### 11.2.10 Channel Instance Selector (ID 30)

This parameter provides the access to the selected parameter instance through the function block parameter interface. The channel instance number starts with 1.

### 11.2.11 Percentage of ATM (ID 201)

This parameter specifies the trip point function block behavior. See section 11.1.

Parameter value	Description
0	Setpoint function
> 0	Atmosphere detection

### 11.2.12 High Trip Point (ID 202)

This parameter corresponds to the "Atmosphere detection" function. It specifies the value at or above which a trip point condition will occur. See section 11.1.2.

### 11.2.13 High Trip Enable (ID 203)

This parameter enables/disables the High Trip Point (Atmosphere detection) setting.

Parameter value	Description
0	disable Atmosphere detection setting
1	enable Atmosphere detection setting

## 12 Transducer Blocks

### 12.1 CDG Transducer Block (Instance 1)

ID	Name	Structure	Data type	Bytes	Access	Store	See section
101	Block Type ID	Simple	Octet string	4	1_R 2_R	N	12.1.1
102	Status Extension	Simple	UINT8	1	1_R 2_R	V	12.1.2
103	Sensor Alarm	Array	UINT8	2	1_R 2_R	V	12.1.3
104	Sensor Warning	Array	UINT8	2	1_R 2_R	V	12.1.4
105	Sensor Temperature	Simple	Float	4	1_R 2_R	NV	12.1.5
201	Offset	Simple	*)	-	1_R/W 2_R/W	NV	12.1.6
202	Filter	Simple	UINT8	1	1_R/W 2_R/W	NV	12.1.7

\*) According to data type value (ID 21)

#### 12.1.1 Block Type ID (ID 101)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the “CDG Transducer Block” is set to “14”. The other defined block types are listed in Appendix B.

#### 12.1.2 StatusExtension (ID 102)

This bit mapped byte parameter provides the following additional information of the CDG sensor (Instance 1):

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Res.	Res.	Res.	Res.	Res.	Underrange Exceeded	Overrange Exceeded	Reading Invalid

Res. = Reserved

- Underrange Exceeded  
This bit is set (“1”) if the ProcessValue PV is below the parameter Underrange (ID 45) of the CDG sensor instance. See section 8.1.21.
- Overrange Exceeded  
This bit is set (“1”) if the ProcessValue PV is above the parameter Overrange (ID 44) of the CDG sensor instance. See section 8.1.20.
- Reading invalid  
This bit represents the logical inversion of the parameter “Reading Valid” (ID 23) of the CDG sensor instance. See section 8.1.8.

#### 12.1.3 Sensor Alarm (ID 103)

This parameter contains the following information:

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte 0	0	0	0	0	0	0	0	DF
Byte 1	0	0	0	0	0	OT	EF	0

- DF = Diaphragm Failure  
This bit is always set to “0”.
- OT = Over Temperature of Electronics  
This bit is set to “1” if an Electronics Overtemperature error occurs.
- EF = Electronics Failure  
This bit is set to “1” if a Temperature Sensor error occurs.

### 12.1.4 Sensor Warning (ID 104)

This parameter contains the following information:

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte 0	0	0	0	0	0	0	0	NT
Byte 1	0	0	0	0	0	0	EW	0

- NT = Not At Temperature  
This bit is set to "1" as long as the specified nominal sensor temperature is not reached.
- EW = Electronics Warning  
This bit is set to "1" if a Temperature Sensor error occurs.

### 12.1.5 Sensor Temperature (ID 105)

This parameter contains the nominal temperature in degrees centigrade (Celsius) to which the sensor warms up.

### 12.1.6 Offset (ID 201)

This parameter defines a customer specified offset which will be added to the Offset-A value from the Zero-Adjust Service in terms of the currently selected data type (ID 21) and data unit (ID 22).

### 12.1.7 Filter (ID 202)

This parameter defines the filter function of the basic device.

Filter	Description
0	Dynamic
1	Fast
2	Slow
3...255	Reserved

## 12.2 ATM Transducer Block (Instance 2)

ID	Name	Structure	Data type	Bytes	Access	Store	See section
14	Atmosphere Sensor Block Command	Array	UINT8	2	1_R/W 2_R/W	N	12.2.5
101	Block Type ID	Simple	Octet string	4	1_R 2_R	N	12.2.1
102	Status Extension	Simple	UINT8	1	1_R 2_R	V	12.2.2
103	Sensor Alarm	Array	UINT8	2	1_R 2_R	V	12.2.3
104	Sensor Warning	Array	UINT8	2	1_R 2_R	V	12.2.4

### 12.2.1 Block Type ID (ID 101)

The Block Type ID Parameter contains an ID which describes the block type. The block type ID of the "ATM Transducer Block" is set to "256". The other defined block types are listed in Appendix B.

### 12.2.2 StatusExtension (ID 102)

This bit mapped byte parameter provides the following additional information of the ATM sensor (Instance 2):

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Res.	Res.	Res.	Res.	Res.	Underrange Exceeded	Overrange Exceeded	Reading Invalid

Res. = Reserved

- **Underrange Exceeded**  
This bit is set ("1") if the ProcessValue PV is below the parameter Underrange (ID 45) of the ATM sensor instance. See section 8.2.11.
- **Overrange Exceeded**  
This bit is set ("1") if the ProcessValue PV is above the parameter Overrange (ID 44) of the ATM sensor instance. See section 8.2.10.
- **Reading invalid**  
This bit represents the logical inversion of the parameter "Reading Valid" (ID 23) of the ATM sensor instance. See section 8.2.6.

### 12.2.3 Sensor Alarm (ID 103)

This byte is always "0". (The ATM Sensor does not produce any errors).

### 12.2.4 Sensor Warning (ID 104)

This byte is always "0". (The ATM Sensor does not produce any warnings).

## 12.2.5 Analog Input Block Adjust Command (ID 14)

ID	Name	Structure	Data Type	Bytes	Access	Store
14	Atmosphere Sensor Block Command	Array	UINT8	2	1_R/W 2_R/W	N

Byte	Name	Structure	Data Type	Bytes	Access	Store
0	Transition Command	Simple	UINT8	1	1_R/W 2_R/W	N
1	Target Value	Simple	UINT8	1	1_R/W 2_R/W	N

Transition Command	Name	Description
0...127	Reserved	Reserved, no action
128	User ATM adjust	"User Atmosphere Adjust" command
129...255	Reserved	Reserved, no action

There are no state transitions associated with the invocation of this service. It is therefore mandatory for the user to switch the device into the desired configuration prior to and during execution of this service. This generally means exposing the sensor to a known environment and treating the values read during execution of the services accordingly.

A successful service response indicates that the service was accepted and the application process started.

To perform the "User ATM Adjust Request", vent the gauge to atmosphere and then start the "User Atmosphere Adjust Service" without any target value.

## Appendix A: Definitions

### Data types

Abbreviation	Range	Data type
INT8	$-2^7 \dots (2^7 - 1)$	Integer 1 byte
INT16	$-2^{15} \dots (2^{15} - 1)$	Integer 2 byte
INT32	$-2^{31} \dots (2^{31} - 1)$	Integer 4 byte
UINT8	$0 \dots (2^8 - 1)$	Unsigned integer 1 byte
UINT16	$0 \dots (2^{16} - 1)$	Unsigned integer 2 byte
UINT32	$0 \dots (2^{31} - 1)$	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 ( $\rightarrow$ 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
1/2_R/W	Acyclically readable and writeable by a Master Class 1 and 2
1_R	Acyclically readable by a master Class 1
2_R	Acyclically readable by a master Class 2
1/2_R	Acyclically readable by a master Class 1 and 2
1_W	Acyclically writeable by a master Class 1
2_W	Acyclically writeable by a master Class 2
1/2_W	Acyclically writeable by a master Class 1 and 2
0_XI	Cyclic output data with master Class 1

Excerpts from:  
 “PROFIBUS Profile for SEMI”  
 (→  [4])

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 <i>software functional unit</i> consisting of an interconnected aggregation of <i>function blocks, events and objects</i> , which may be distributed and which may have <i>interfaces</i> with other <i>applications</i> .
Characteristic	An characteristic is a property or characteristic of an <i>entity</i> . (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 <i>aggregate</i> whose elements need not be of the same <i>data type</i> , and each of them is uniquely referenced by an <i>offset identifier</i> .
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	<i>Input data</i> are transmitted from the device to the bus. <i>Output data</i> 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, <i>process</i> , object, concept, association or <i>event</i> .
Function	(1) A specific purpose of an entity. (2) One of a group of actions performed by an entity.
Function Block	A named <i>block</i> 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.
Function Block Application	Application of an automation system performed by a Device Block, Function Block, Transducer Block and accompanied elements.

Excerpts from:  
 "PROFIBUS Profile for SEMI"  
 (cont.)

Term	Meaning
Instance	A set of data related to an invocation of a function block or a class.
Internal Resolution (ir)	The internal resolution is 16383 (3FFF <sub>hex</sub> ) for 100% and -16384 (C000 <sub>hex</sub> ) for -100% of the range.
Mode	Determines the block operating mode and available modes for a block instance.
Object	(IT) A software entity having identity, attributes and behavior.
Parameter	A <i>variable</i> that is given a constant value for a specified <i>application</i> and that may denote the <i>application</i> .
Device Block	A Device Block is a named block. Hardware specific parameters of a field device, which are associated with a resource, are made visible through the Device Block. Similar to transducer blocks, they insulate function blocks from the physical hardware by a set of implementation independent hardware parameters.
Record	A set of <i>data items</i> of different data types treated as a unit.
Resource	A resource is considered to be a logical subdivision within the software (and possibly hardware) structure of a device. Resources have independent control of their operation. The definition of a resource may be modified without affecting other resources within a device. A resource accepts and processes data and/or events from the process and/or communication interfaces and returns data and/or events to the process and/or communication interfaces, as specified by the applications utilizing the resource. An interoperable network view of applications is provided through device resources. Each resource specifies the network visible aspects of one or more local applications (or parts of distributed applications).
Simple Variable	A single variable which is characterized by a defined Data Type.
Substitute Value	In case an optional parameter has not been implemented, the device behaves according to the substitute value for this parameter.
Transducer Block	Transducer Block is a named block. Transducer blocks insulate function blocks from the specifics of I/O devices, such as sensors, actuators, and switches. Transducer blocks control access to I/O devices through a device independent interface defined for use by function blocks. Transducer blocks also perform functions, such as calibration and linearization, on I/O data to convert it to a device independent representation. Their interface to function blocks is defined as one or more implementation independent I/O channels.
Variable	A <i>software</i> entity that may assume any one of a set of values. The values of a variable are usually restricted to a certain data type.

## Appendix B: Block Type

Currently defined Block Types






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 <sup>1)</sup>	13
Diaphragm Gauge <sup>2)</sup>	14
Cold Cathode Ion Gauge	15
Hot Cathode Ion Gauge <sup>3)</sup>	16
Trip Point Function Block	17
OneOfN Vacuum Pressure Gauge Transducer Block	18
Reserved	19 ... $2^8 - 1$
Manufacturer-specific	$2^8 \dots 2^{16} - 1$

<sup>1)</sup> Pirani measuring system

<sup>2)</sup> CDG measuring system

<sup>3)</sup> BA measuring system

## Appendix C: Literature

-  [1] [www.inficon.com](http://www.inficon.com)  
 Operating Manual  
 CDG045D  
 tina51e1  
 INFICON AG, LI-9496 Balzers, Liechtenstein
-  [2] [www.inficon.com](http://www.inficon.com)  
 Operating Manual  
 CDG100D  
 tina52e1  
 INFICON AG, LI-9496 Balzers, Liechtenstein
-  [3] [www.inficon.com](http://www.inficon.com)  
 Operating Manual  
 CDG160D, CDG200D  
 tina53e1  
 INFICON AG, LI-9496 Balzers, Liechtenstein
-  [4] [www.profibus.com](http://www.profibus.com)  
 Profibus user organization
-  [5] European Standard for Profibus EN 50170

Notes

Original: English tira54e1 (2008-06)



tira54e1



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

*www.inficon.com*