



Pressure sensor, model MH-4-CAN

© 02/2022 WIKA Alexander Wiegand SE & Co. KG  
All rights reserved. / Alle Rechte vorbehalten.  
WIKA® is a registered trademark in various countries.  
WIKA® ist eine geschützte Marke in verschiedenen Ländern.

Prior to starting any work, read the installation instructions!  
Keep for later use!

Vor Beginn aller Arbeiten Einbauanleitung lesen!  
Zum späteren Gebrauch aufbewahren!

# Contents

<b>1. Model MH-4-CAN</b>	<b>6</b>
1.1 General information . . . . .	.6
1.2 Features . . . . .	.6
1.3 CANopen feature summary . . . . .	.6
1.4 Supported bitrates . . . . .	.6
1.5 CAN frames . . . . .	.6
<b>2. Quick start guide</b>	<b>7</b>
2.1 Quick start introduction . . . . .	.7
2.2 Connecting and powering the device . . . . .	.7
2.3 Configuring the device . . . . .	.7
2.4 Starting the device . . . . .	.8
2.5 Service Data Object (SDO) . . . . .	.8
2.5.1 SDO read object . . . . .	.8
2.5.2 SDO write object . . . . .	.9
<b>3. CANopen communication</b>	<b>9</b>
3.1 CANopen introduction . . . . .	.9
3.2 Device profile 404. . . . .	10
3.3 Service data object (SDO) . . . . .	10
3.3.1 SDO read object . . . . .	11
3.3.2 SDO write object . . . . .	11
3.3.3 Abort SDO Transfer . . . . .	12
3.3.4 SDO Abort Codes . . . . .	12
3.4 Transmit process data object (TPDO) . . . . .	13
3.4.1 TPDO mapping . . . . .	13
3.4.2 Changing the TPDO mapping . . . . .	14
3.4.3 TPDO transmission types . . . . .	15
3.5 Network management (NMT) . . . . .	16
3.5.1 Module Control Protocol . . . . .	16
3.5.2 Error Control Services . . . . .	17
3.5.3 Bootup Service . . . . .	17
3.6 Emergency object (EMCY). . . . .	17
<b>4. The Object Dictionary</b>	<b>18</b>
4.1 Object Dictionary – Communication Profile Area . . . . .	18
4.1.1 Object 1000h: Device Type . . . . .	18
4.1.2 Object 1001h: Error Register . . . . .	18
4.1.3 Object 1003h: Predefined Error Field . . . . .	18
4.1.4 Object 1005h: COB-ID SYNC . . . . .	20
4.1.5 Object 1008h: Manufacturer Device Name . . . . .	21

4.1.6	Object 1009h: Manufacturer Hardware Version . . . . .	21
4.1.7	Object 100Ah: Manufacturer Software Version . . . . .	21
4.1.8	Object 1010h: Store Parameter Field . . . . .	21
4.1.9	Object 1011h: Restore Default Parameters . . . . .	23
4.1.10	Object 1014h: COB ID EMCY . . . . .	24
4.1.11	Object 1015h: Inhibit Time Emergency . . . . .	25
4.1.12	Object 1017h: Producer Heartbeat Time . . . . .	25
4.1.13	Object 1018h: Identity Object . . . . .	25
4.1.14	Object 1800h to 1801h: Transmit PDO Communication Parameter . . . . .	27
4.1.15	Object 1A00h to 1A01h: Transmit PDO Mapping Parameter . . . . .	28
4.1.16	Object 1F80h: NMT Startup . . . . .	31
4.2	Object Dictionary – Device Profile Area . . . . .	32
4.2.1	Object 6110h: AI Sensor Type . . . . .	32
4.2.2	Object 6112h: AI Operating Mode . . . . .	33
4.2.3	Object 6125h: AI Autozero . . . . .	34
4.2.4	Object 6126h: AI Scaling Factor . . . . .	34
4.2.5	Object 6127h: AI Scaling Offset . . . . .	35
4.2.6	Object 6130h: AI Input PV . . . . .	36
4.2.7	Object 6131h: AI Physical Unit PV . . . . .	37
4.2.8	Object 6132h: AI Decimal Digits PV . . . . .	39
4.2.9	Object 6133h: AI Interrupt Delta Input PV . . . . .	40
4.2.10	Object 6134h: AI Interrupt Lower Limit Input PV . . . . .	41
4.2.11	Object 6135h: AI AI Interrupt Upper Limit Input PV . . . . .	42
4.2.12	Object 6136h: AI Interrupt Hysteresis Input PV . . . . .	42
4.2.13	Object 6148h: AI Span Start . . . . .	43
4.2.14	Object 6149h: AI Span End . . . . .	44
4.2.15	Object 6150h: AI Status . . . . .	45
4.2.16	Object 61A0h: AI Filter Type . . . . .	47
4.2.17	Object 61A1h: AI Filter Constant . . . . .	47
4.2.18	Object 7100h: AI Input FV . . . . .	48
4.2.19	Object 7130h: AI Input PV . . . . .	49
4.2.20	Object 7133h: AI Interrupt Delta Input PV . . . . .	50
4.2.21	Object 7134h: AI Interrupt Lower Limit Input PV . . . . .	51
4.2.22	Object 7135h: AI Interrupt Upper Limit Input PV . . . . .	52
4.2.23	Object 7136h: AI Interrupt Hysteresis Input PV . . . . .	53
4.2.24	Object 7148h: AI Span Start . . . . .	53
4.2.25	Object 7149h: AI Span End . . . . .	54
4.2.26	Object 9130h: AI Input PV . . . . .	55
4.2.27	Object 9133h: AI Interrupt Delta Input PV . . . . .	56
4.2.28	Object 9134h: AI Interrupt Lower Limit Input PV . . . . .	57
4.2.29	Object 9135h: AI Interrupt Upper Limit Input PV . . . . .	58
4.2.30	Object 9136h: AI Interrupt Hysteresis Input PV . . . . .	59

4.2.31	Object 9148h: AI Span Start . . . . .	60
4.2.32	Object 9149h: AI Span End . . . . .	61
4.3	Object Dictionary – Manufacturer Specific Area. . . . .	62
4.3.1	Object 2010h: Measuring Range Start PV . . . . .	62
4.3.2	Object 2011h: Measuring Range End PV . . . . .	63
4.3.3	Object 2012h: Original Physical Unit PV . . . . .	64
4.3.4	Object 2020h: Minimum Value PV . . . . .	65
4.3.5	Object 2021h: Maximum Value PV . . . . .	66
4.3.6	Object 2022h: Reset Min Max Value PV . . . . .	67
4.3.7	Object 2023h: Damping PV . . . . .	68
4.3.8	Object 2320h: Node-ID . . . . .	69
4.3.9	Object 2321h: Bitrate . . . . .	70
<b>5.</b>	<b>Emergency Object (EMCY)</b>	<b>70</b>
<b>6.</b>	<b>References</b>	<b>74</b>
<b>7.</b>	<b>Change log</b>	<b>74</b>

---

## 1. Model MH-4-CAN

### 1.1 General information

The MH-4-CAN is a pressure sensor providing measurements through a CANopen interface implementing device profile DS404.

### 1.2 Features

Sample rate (pressure): 640  $\mu$ s

Settling time: 2 ms

### 1.3 CANopen feature summary

CANopen feature summary	
CANopen type	NMT Slave
Bitrate	From 20 kBit/s to 1 Mbit/s
Number of transmit PDOs	2
PDO Mapping	Dynamic
PDO Triggering Events	Synchronous, asynchronous
Heartbeat Protocol	Supported
Emergency Messages	Supported
Device Profile	CiA 404
Layer Setting Services (LSS)	Supported
Default settings	
Node-ID	1
Bitrate	250 kBit/s

### 1.4 Supported bitrates

- 1.000 kBit/s
- 800 kBit/s
- 500 kBit/s
- 250 kBit/s
- 125 kBit/s
- 50 kBit/s
- 20 kBit/s
- Automatic bit rate detection

### 1.5 CAN frames

The pressure transmitter supports standard CAN frames with 11-bit identifier. Extended frames with 29-bit identifier are not supported but tolerated.

## 2. Quick start guide

### 2.1 Quick start introduction

This chapter describes setting the sensor up using a simple CAN-Software without specialized CANopen support. The sensor may be configured entirely using raw CAN telegrams. We recommend to use a software with build-in CANopen support.

Please do note this is only a couple of possible configurations, for full details please refer to the official CANopen specification.

### 2.2 Connecting and powering the device

First setup the sensor:

- Connect the CAN interface to your PC (e.g. PEAK PCAN-USB, Ixxat USB-to-CAN) and to the sensor and start the CAN-Software (e.g. PEAK PCAN-View, Ixxat canAnalyser3 Mini). Ensure you use the correct bitrate and Node-ID of the pressure transmitter.
- When the MH-4 is powered up it sends a Boot-Up message

ID	DLC	Byte0
700h + Node-ID	1	00h

The sensor is now in PRE-OPERATIONAL mode and ready to be configured by SDO (if necessary).

### 2.3 Configuring the device

The sensor is now ready to be configured and started. Choose one of the following transmission methods:

- Acyclic synchronous transmission (transmission type = 0)  
Triggered when the SYNC message is received and one of the mapped process data has changed its value after the last transmission.
- Cyclic synchronous transmission (transmission type = 1 ... 240) (default = 1)  
Triggered when a SYNC message is received (type = 1), each second SYNC message received (type = 2), etc.
- Asynchronous event/timer triggered
  - Timer triggered (transmission type = 254)  
Adjustable in ms (via Object 1800h.5)
  - Event triggered (transmission type = 254/255)  
If the measured value falls below or exceeds an adjustable limit.  
If the measured value has changed by more than a delta value compared to the last transferred measured value.

In order to change the transmission type object 1800h.2 must be changed to the appropriate transmission type using SDO write access (see 2.5).

## 2. Quick start guide

### 2.4 Starting the device

The configured device must now be started by the master:

- In order to send PDO-messages the sensor has to be set to OPERATIONAL mode  
Master transmits:

ID	DLC	Byte0	Byte1
00h (0 = all nodes)	2	01h	Node-ID

The sensor is now in OPERATIONAL mode

- Depending on the previous selected transmission type the master needs to send the SYNC object:

Master transmits:

ID	DLC
80h	0

The received data now looks like (depending on configuration):

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4
180h + Node-ID	5	6130h subindex 1				6150h subindex 1

This data is called TPDO.

### 2.5 Service Data Object (SDO)

#### 2.5.1 SDO read object

The SDO read object to read an dictionary entry:

Master transmits:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
600h + Node-ID	8	CS = 40h	LSB	MSB	Subindex	00h	00h	00h	00h
			Index						

MH-4 CAN replies:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
580h + Node-ID	8	CS	LSB	MSB	Subindex	LSB			MSB
			Index						



CS	Number of valid bytes
4Fh	1 (Byte4)
4Bh	2 (Byte4 - Byte5)
47h	3 (Byte4 - Byte6)
43h	4 (Byte4 - Byte7)

### 2.5.2 SDO write object

The SDO read object to write data to an dictionary entry:

Master transmits:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
600h + Node-ID	8	CS	LSB	MSB	Subindex	LSB			MSB
			Index			Data			

CS	Number of valid bytes
2Fh	1 (Byte4)
2Bh	2 (Byte4 - Byte5)
27h	3 (Byte4 - Byte6)
23h	4 (Byte4 - Byte7)

MH-4 CAN replies:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
580h + Node-ID	8	CS = 60h	LSB	MSB	Subindex	00h	00h	00h	00h
			Index						

Other values for CS than 60h are abort codes, indicating a failed SDO write (see 3.3).

## 3. CANopen communication

### 3.1 CANopen introduction

CANopen is a communication protocol using the CAN bus to provide sensor access and measurements.

All CANopen data is stored in the CANopen dictionary. CANopen supports a couple of different objects to access the object dictionary entries:

- **SDO (Service Data Object):** Used to access any object of the object dictionary. Offers read and write functionality. Usually used to indentify and configure the sensor.

## 3. CANopen communication

- **TPDO (Transmit Process Data Object):** Provides status and measurement information with a low communication overhead. Usually used to retrieve the measured data.

EN

The CANopen dictionary contains all data the sensor provides. It consists of multiple entries with the following properties (in excerpts):

- **Index:** The 16bit object index, usually written in hexadecimal, e.g.: 1000h
- **Subindex:** Each object may contain up to 255 subobjects. If no subobjects are supported a subindex of 0 is used. In case of existing subobjects subindex 0 provides a UNSIGEND8 value indicating the number of provided subobjects.
- **Data Type:** The data type, e.g.: UNSIGEND32 (unsigned 32 bit integer)
- **Access:** The permitted dictionary access, e.g.: RW (ReadWrite), RO (Read Only)
- **Default Value:** The default value of an object after CANopen device initialization
- **PDO Mapping:** When yes the object may be transmitted by TPDO (depending on TPDO configuration, see 3.4.1)

In the following document objects and subobjects are referenced by following nomenclature: "object.subobject" (e.g. 1A00h.01 means object index 1A00h, subobject 01h). All indices are given as hex numbers.

CANopen devices implement different device profiles, which provide different functionality and object entries.

### 3.2 Device profile 404

The sensor implements the CANopen device profile 404 for measuring devices. This adds a couple of object dictionary entries, ranging from 6110h to 9149h.

The measurement values are called analog input process values. Mainly they are provided through the entries 6130h, 7130h and 9130h (Analog input PV).

A couple of objects, like the analog input PV exist in 3 ways, distinguished by the first character:

- 6xxxh The resulting data type is REAL32 (IEE 754)
- 7xxxh The resulting data type is INTEGER16
- 9xxxh The resulting data type is INTEGER32

Entries like these are often referred to in a form like x130h, which means 'any kind of analog input'

### 3.3 Service data object (SDO)

Service Data Objects are used to access the entries of the transmitters object dictionary. Therefore the dictionary entries are accessed by index and subindex.

Therefore the master sends a specific request message followed by an reply from the sensor.

## 3. CANopen communication

EN

### 3.3.1 SDO read object

The SDO read object to read an dictionary entry:

Master transmits:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
600h + Node-ID	8	CS = 40h	LSB	MSB	Subindex	00h	00h	00h	00h
			Index						

MH-4 CAN replies:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
580h + Node-ID	8	CS	LSB	MSB	Subindex	LSB			MSB
			Index		Data				

CS	Number of valid bytes
4Fh	1 (Byte4)
4Bh	2 (Byte4 - Byte5)
47h	3 (Byte4 - Byte6)
43h	4 (Byte4 - Byte7)

### 3.3.2 SDO write object

The SDO read object to write data to an dictionary entry:

Master transmits:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
600h + Node-ID	8	CS = 60h	LSB	MSB	Subindex	LSB			MSB
			Index		Data				

CS	Number of valid bytes
2Fh	1 (Byte4)
2Bh	2 (Byte4 - Byte5)
27h	3 (Byte4 - Byte6)
23h	4 (Byte4 - Byte7)

## 3. CANopen communication

MH-4 CAN replies:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
580h + Node-ID	8	CS = 60h	LSB	MSB	Subindex	00h	00h	00h	00h
			Index						

Other values for CS than 60h are abort codes, indicating a failed SDO write (see 3.3).

### 3.3.3 Abort SDO Transfer

If an error occurred while reading or writing an object, the transmitter answers:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
580h + Node-ID	8	CS = 80h	LSB	MSB	Subindex	LSB			
			Index			Abort Code			

### 3.3.4 SDO Abort Codes

Possible SDO abort codes are:

Abort code	Description
0503 0000h	Toggle bit not altered
0504 0000h	SDO protocol timed out
0504 0001h	Client/server command specifier not valid or unknown
0504 0002h	Invalid block size (block mode only)
0504 0003h	Invalid block sequence number (block mode only)
0504 0004h	Invalid block CRC value (block mode only)
0504 0005h	Out of memory
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write only object
0601 0002h	Attempt to write a read only object
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	The number and length of the objects to be mapped would exceed PDO length
0604 0043h	General parameter incompatibility reason
0604 0047h	General internal incompatibility in the device
0606 0000h	Access failed due to an hardware error
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0030h	Invalid value for parameter (download only)

## 3. CANopen communication

EN

Abort code	Description
0609 0031h	Value of parameter written too high (download only)
0609 0032h	Value of parameter written too low (download only)
0609 0036h	Maximum value is less than minimum value
060a 0023h	Resource not available: SDO connection
0800 0000h	General error
0800 0020h	Data cannot be transferred or stored to the application
0800 0021h	Data cannot be transferred or stored to the application because of local control
0800 0022h	Data cannot be transferred or stored to the application because of the present device state
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error).
0800 0024h	No data available

### 3.4 Transmit process data object (TPDO)

Transmit process data objects are used by the sensor to frequently transmit measured data with low protocol overhead. A TPDO may contain up to 8 bytes of measurement or status data. The data mapped to the TPDO may be changed through the TPDO mapping.

All TPDO transmission requires the sensor to be in NMT state 'OPERATIONAL' (see 3.4).

TPDO data is transmitted using the CAN-ID: 1800h.1 (COB-ID) + Node-ID, data length as big as needed by the data.

#### 3.4.1 TPDO mapping

Object entries marked as 'mapable' may be mapped into either TPDO. The TPDO mapping may be accessed through the object 1A00h and 1A01h (the first one defines the content of TPDO1 the latter of TPDO2).

Subindex 0 contains the number of used entries (up to 8, excluding subindex 0).

The other subindex contain the object dictionary entries index and subindex encoded as a 32bit integer. The information is encoded as follows:

Bits 24..31	Bits 16..23	Bits 8..15	Bits 0..7
Index (high byte)	Index (low byte)	Subindex	Data length in bits

Please note, that the data length needs to match the entries data size from the object dictionary.

### 3. CANopen communication

#### Example:

the value 0x61300120 will map the object 6130h (AI input PV) subindex 01 (pressure value as REAL32) to the PDO. Since the size of REAL32 is 32 bit, data length needs to be set to 0x20h.

EN

When mapping multiple entries to an TPDO, the data is mapped from subindex 1 to 8.

#### Example:

The PDO1 with the mapping:

1800h.0 = 2

1800h.1 = 0x61300120

1800h.2 = 0x61500108

would look like:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	
180h + Node-ID	5	LSB				MSB	
		6130h subindex 1					
						6150h subindex 1	

#### 3.4.2 Changing the TPDO mapping

In order to change the TPDO mapping a couple of steps need to be executed in the right order:

- Set TPDO invalid by switching bit 31 in the COB-ID (e.g. 1800h.1)
- Set TPDO mapping invalid by setting the mapping index (e.g. 1A00h.0) to 0
- Change TPDO mapping entries
- Set TPDO mapping index top number of used entries
- Set TPDO valid by switching bit 31 back.

#### Example:

- Set TPDO invalid by setting bit "valid" to 1b (1800h.1)

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
601h	8	23h	00h	18h	01h	81h	01h	00h	C0h
			Index			Data			

- Disable mapping by setting index 1A00.0 to 00h

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
601h	8	2Fh	00h	1Ah	00h	00h	00h	00h	00h
			Index			Data			

## 3. CANopen communication

- Modify mapping by changing the values of the corresponding sub-indices, e.g. change mapped object to 6130.1

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
601h	8	23h	00h	1Ah	01h	20h	01h	30h	61h
Index					Data				

- Enable mapping by setting sub-index 00h to the number of mapped objects (1).

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
601h	8	2Fh	00h	1Ah	00h	01h	00h	00h	00h
Index					Data				

- Set TPDO valid by setting bit “valid” to 0b (1800.1)

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
601h	8	23h	00h	18h	01h	81h	01h	00h	40h
Index					Data				

### 3.4.3 TPDO transmission types

The PDO transmission types can be configured via objects 1800h/1801h subindex 2.

#### Synchronous transmission (cyclic, acyclic)

A transmission type of 0 means that the PDO is triggered acyclic when a SYNC message is received and one of the mapped process data has changed its value after the last transmission.

A transmission type of n (1 ... 240) means that the PDO is triggered cyclic with every n-th SYNC message.

#### Asynchronous transmission

A transmission type of 254 means that the PDO is triggered after the event timer elapses. The event timer can be configured via objects 1800h/1801h subindex 5.

A transmission type of 254/255 means that the message will be sent if the measured value falls below or exceeds an adjustable limit and also if the measured value has changed by more than a delta value compared to the last transferred measured value. The interrupt limits can be configured via entries x133h/x134h/x135h/x136h.

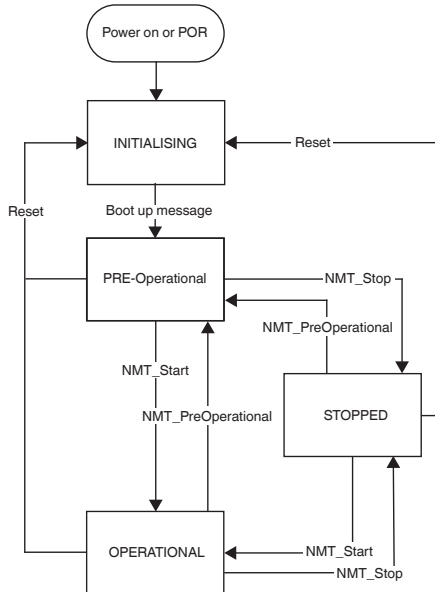
# 3. CANopen communication

## 3.5 Network management (NMT)

Every CANopen device has an internal state machine as shown in figure 1:

NMT State machine (simplified):

EN



After POR all devices enter the state 'INITIALISING'. The initial state is left automatically after finishing the initialization. Leaving the initialization state is indicated by sending the bootup message (see 3.2).

After leaving 'INITIALISING' the device entered the Pre-Operational state. The PRE-OPERATIONAL is a kind of idle state, this is the goto state to configure the device. In order to obtain TPDOs from the sensor, the sensor must be started via NMT command from the master (See 3.5.1)).

It is possible to configure the sensor to enter the state Operational without NMT command, for more information please refer to 4.1.16 (Object 1F80h).

In case of an EMCY the device stays in its current state.

### 3.5.1 Module Control Protocol

NMT Master request:

ID	DLC	Byte0	Byte1
00h	2	CS	Node-ID (0 = all nodes)



## 3. CANopen communication

EN

CS: NMT command specifier  
NMT services are unconfirmed.

CS	
01h	Start Remote Node
02h	Stop Remote Node
80h	Enter Pre-Operational
81h	Reset Node
82h	Reset Communication

### 3.5.2 Error Control Services

Through Error control services the NMT detects failures in a CAN-based.

The heartbeat mechanism for a device is established through cyclically transmitting a message by a heartbeat producer. One or more devices in the network are aware of this heartbeat message. If the heartbeat cycle fails for the heartbeat producer the local application on the heartbeat consumer will be informed about that event. (see 4.1.12 (object heartbeat producer)).

### 3.5.3 Bootup Service

Through this service, the NMT slave indicates that a local state transition occurred from the state 'INITIALISING' to the state 'PRE-OPERATIONAL'. The protocol uses the same identifier as the error control protocols.

ID	DLC	Byte0
700h + Node-ID	1	00h

### 3.6 Emergency object (EMCY)

In case of an EMCY the sensor sends an EMCY object:

Sensor transmits:

ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
80h + Node-ID	8	LSB	MSB	Error register (1001h)	Subindex	Manufacturer specific error field			
		Emergency Error Code							

For detailed error behaviour description and supported error codes please refer to chapter 5.

### 4. The Object Dictionary

#### 4.1 Object Dictionary – Communication Profile Area

EN

##### 4.1.1 Object 1000h: Device Type

The device type describes the type of device and its functionality. The lower 16-bit field describes the device profile number (404) and the upper 16-bit field contains additional information (analogue input block, Manufacturer-specific PDO Mapping).

<b>Index</b>	1000h
<b>Parameter Name</b>	Device Type
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>Default Value</b>	0x00820194
<b>PDO Mapping</b>	No

##### 4.1.2 Object 1001h: Error Register

The error register is a field of 8 bits, each for a certain error type. If an error occurs the corresponding bit is set.

Supported error bits:

- Bit 0: Generic Error (always set at any error situation)
- Bit 3: Temperature
- Bit 5: Device profile specific, always 0b for DS 404
- Bit 4: Communication error (overrun, error state)
- Bit 7: Manufacture specific

The detailed information of each error is described in 6 Emergency Object (EMCY)

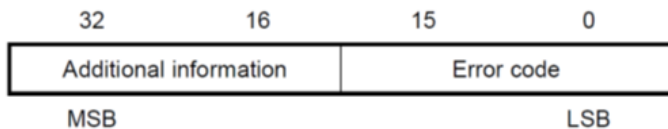
<b>Index</b>	1001h
<b>Parameter Name</b>	Error Register
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>PDO Mapping</b>	No

##### 4.1.3 Object 1003h: Predefined Error Field

This object holds errors that have occurred on the transmitter and have been signalled via Emergency Object. It represents an error history containing up to 4 errors. For a list of possible error entries please refer to chapter 5.

## 4. The Object Dictionary

In case that any error is into the history the values have the following meaning:



EN

For detailed description of Error code see 6 Emergency Object (EMCY).

The Bits 16 to 23 (1 Byte) is Byte 4 in the EMCY message

The Bits 24 to 31 (1 Byte) is Byte 3 in the EMCY message

Writing to sub index 0 deletes the entire error history.

<b>Index</b>	1003h
<b>Parameter Name</b>	Predefined Error Field
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of Errors
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	rw
<b>Default Value</b>	0
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Standard Error Field
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>Default Value</b>	0
<b>PDO Mapping</b>	No

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Standard Error Field_2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>Default Value</b>	0
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	3
<b>Parameter Name</b>	Standard Error Field_3
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>Default Value</b>	0
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	4
<b>Parameter Name</b>	Standard Error Field_4
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>Default Value</b>	0
<b>PDO Mapping</b>	No

### 4.1.4 Object 1005h: COB-ID SYNC

COB-ID of the Synchronization object.

<b>Index</b>	1005h
<b>Parameter Name</b>	COB ID SYNC
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Lower Limit</b>	0x00000001
<b>Access</b>	rw
<b>Default Value</b>	0x00000080
<b>PDO Mapping</b>	No

14547342.01 03/2022 EN

### 4.1.5 Object 1008h: Manufacturer Device Name

Contains the device name.

<b>Index</b>	1008
<b>Parameter Name</b>	Manufacturer Device Name
<b>Object Type</b>	VAR
<b>Data Type</b>	VISIBLE_STRING
<b>Access</b>	const
<b>Default Value</b>	MH-4-CAN
<b>PDO Mapping</b>	No

### 4.1.6 Object 1009h: Manufacturer Hardware Version

Contains the transmitter hardware version.

<b>Index</b>	1009h
<b>Parameter Name</b>	Manufacturer Hardware Version
<b>Object Type</b>	VAR
<b>Data Type</b>	VISIBLE_STRING
<b>Access</b>	const
<b>PDO Mapping</b>	No

### 4.1.7 Object 100Ah: Manufacturer Software Version

Contains the transmitter software version.

<b>Index</b>	100Ah
<b>Parameter Name</b>	Manufacturer Software Version
<b>Object Type</b>	VAR
<b>Data Type</b>	VISIBLE_STRING
<b>Access</b>	const
<b>PDO Mapping</b>	No

### 4.1.8 Object 1010h: Store Parameter Field

This entry supports saving of parameters in non-volatile memory.

With a read access the device provides information about its saving capabilities.

- Sub-index 1: All parameters
- Sub-index 2: Communication Parameters
- Sub-index 3: Application parameters

For saving the signature “save” (0x65766173) must be written.

## 4. The Object Dictionary

EN

<b>Index</b>	1010h
<b>Parameter Name</b>	Store Parameter Field
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Lower Limit</b>	0x0
<b>Higher Limit</b>	0x7F
<b>Access</b>	ro
<b>Default Value</b>	4
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Save all Parameters
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Save Communication Parameters
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	3
<b>Parameter Name</b>	Save Application Parameters
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

14547342.01 03/2022 EN

### 4.1.9 Object 1011h: Restore Default Parameters

This entry supports restoring of default parameters.

With a read access the transmitter provides information about its capabilities to restore these values.

- Sub-index 1: All parameters
- Sub-index 2: Communication Parameters
- Sub-index 3: Application parameters

For restoring the signature “load” (0x64616f6c) must be written.

<b>Index</b>	1011h
<b>Parameter Name</b>	Restore Default Parameters
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Lower Limit</b>	0x0
<b>Higher Limit</b>	0x7F
<b>Access</b>	ro
<b>Default Value</b>	4
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Restore all Default Parameters
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Restore Communication Default Parameters
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	3
<b>Parameter Name</b>	Restore Application Default Parameters
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.1.10 Object 1014h: COB ID EMCY

COB-ID used for emergency message (Emergency Producer).

Please do note, that storing all parameters (1010h.1, Save all parameters) will overwrite the cob-ID with 80h+Node-ID. In order to save a COB-ID different from this pattern use 1010h.2 (Save communication parameters) instead.

<b>Index</b>	1014h
<b>Parameter Name</b>	COB ID EMCY
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Lower Limit</b>	0x00000001
<b>Higher Limit</b>	0xFFFFFFFF
<b>Access</b>	ro
<b>Default Value</b>	\$NODEID+0x80
<b>PDO Mapping</b>	No



### 4.1.11 Object 1015h: Inhibit Time Emergency

The inhibit time emergency defines the inhibit time for the EMCY Message.

<b>Index</b>	1015h
<b>Parameter Name</b>	Inhibit Time Emergency
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED16
<b>Access</b>	rw
<b>Default Value</b>	0
<b>PDO Mapping</b>	No

### 4.1.12 Object 1017h: Producer Heartbeat Time

The producer heartbeat time defines the cycle time of the heartbeat. If the time is 0 it is not used.

The time has to be a multiple of 1 millisecond.

<b>Index</b>	1017h
<b>Parameter Name</b>	Producer Heartbeat Time
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED16
<b>Access</b>	rw
<b>Default Value</b>	0
<b>PDO Mapping</b>	No

### 4.1.13 Object 1018h: Identity Object

This object contains general information about the device.

- Sub-index 1: CANopen Vendor ID
- Sub-index 2: Manufacturer specific product code
- Sub-index 3: Revision number. Bit 31-16 is the major, Bit 15-0 the minor revision number.
- Sub-index 4: Manufacturer specific serial number.

<b>Index</b>	1018h
<b>Parameter Name</b>	Identity Object
<b>Object Type</b>	RECORD

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Lower Limit</b>	1
<b>Higher Limit</b>	4
<b>Access</b>	ro
<b>Default Value</b>	4
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Vendor Id
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>Default Value</b>	71
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Product Code
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	3
<b>Parameter Name</b>	Revision number
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	4
<b>Parameter Name</b>	Serial number
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.1.14 Object 1800h to 1801h: Transmit PDO Communication Parameter

This object contains the communication parameters for the PDOs the CANopen device is able to transmit.

- Sub-index 0: Number of PDO-parameters implemented
- Sub-index 1: COB-ID
- Sub-index 2: Transmission type
- Sub-index 5: Event timer

Please do note, that storing all parameters (1010h.1, Save all parameters) will overwrite the cob-ID with 180h+Node-ID. In order to save a COB-ID different from this pattern use 1010h.2 (Save communication parameters) instead.

<b>Index</b>	1800h to 1801h
<b>Parameter Name</b>	Transmit PDO Communication Parameter
<b>Object Type</b>	RECORD

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	5
<b>PDO Mapping</b>	No

## 4. The Object Dictionary

EN

Sub-Index	1
Parameter Name	COB ID
Object Type	VAR
Data Type	UNSIGNED32
Lower Limit	0x00000001
Higher Limit	0xFFFFFFFF
Access	rw
Default Value	\$NODEID+0x40000180 (1800h) \$NODEID+0x40000280 (1801h)
PDO Mapping	No

Sub-Index	2
Parameter Name	Transmission Type
Object Type	VAR
Data Type	UNSIGNED8
Lower Limit	0
Higher Limit	255
Access	rw
Default Value	1 (1800h) 254 (1801h)
PDO Mapping	No

Sub-Index	5
Parameter Name	Event Timer
Object Type	VAR
Data Type	UNSIGNED16
Access	rw
Default Value	0
PDO Mapping	No

### 4.1.15 Object 1A00h to 1A01h: Transmit PDO Mapping Parameter

Contains the mapping for the PDOs the transmitter is able to transmit.

The sub-index 0h contains the number of valid entries within the mapping record.

This number of entries is also the number of the application variables which shall be transmitted with the corresponding PDO.

14547342.01 03/2022 EN

## 4. The Object Dictionary

EN

The sub-indices from 1h to number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

All three values are hexa-decimal coded. The length entry contains the length of the object in bits (1 ... 40h).

Possible entries for PDO mapping:

- 0x6130 01 20 – Process Value Pressure as float32
- 0x7130 01 10 – Process Value Pressure as int16
- 0x9130 01 20 – Process Value Pressure as int32
- 0x6150 01 08 – Analog Input Status Pressure as uint8
- 0x6130 02 20 – Process Value Temperature as float32
- 0x7130 02 10 – Process Value Temperature as int16
- 0x9130 02 20 – Process Value Temperature as int32
- 0x6150 02 08 - Analog Input Status Temperature as uint8

<b>Index</b>	1A00h to 1A01h
<b>Parameter Name</b>	Transmit PDO Mapping Parameter
<b>Object Type</b>	RECORD

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Lower Limit</b>	0
<b>Higher Limit</b>	8
<b>Access</b>	rw
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	PDO Mapping Entry
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>Default Value</b>	0x61300120 (1A00h) 0x61300220 (1A01h)
<b>PDO Mapping</b>	No

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	2
<b>Parameter Name</b>	PDO Mapping Entry_2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>Default Value</b>	0x61500108 (1A00h) 0x61500208 (1A01h)
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	3
<b>Parameter Name</b>	PDO Mapping Entry_3
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	4
<b>Parameter Name</b>	PDO Mapping Entry_4
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	5
<b>Parameter Name</b>	PDO Mapping Entry_5
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	6
<b>Parameter Name</b>	PDO Mapping Entry_6
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	7
<b>Parameter Name</b>	PDO Mapping Entry_7
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	8
<b>Parameter Name</b>	PDO Mapping Entry_8
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.1.16 Object 1F80h: NMT Startup

This object configures the startup behavior of a CANopen device.

Allowed values:

- 0000 0000h: The NMT master shall start the NMT slave.
- 0000 0008h: The NMT master shall not start the NMT slave and the application may start the NMT slave.

<b>Index</b>	1F80h
<b>Parameter Name</b>	NMTStartup
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

# 4. The Object Dictionary

## 4.2 Object Dictionary – Device Profile Area

### 4.2.1 Object 6110h: AI Sensor Type

Specifies the type of sensor, which is connected to the analogue input.

- Sub-index 1: 90 = Pressure transducer
- Sub-index 2: 100 = Temperature transducer

EN

<b>Index</b>	6110h
<b>Parameter Name</b>	AI Sensor Type
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Sensor Type 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED16
<b>Access</b>	ro
<b>Default Value</b>	90
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Sensor Type 2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED16
<b>Access</b>	ro
<b>Default Value</b>	100
<b>PDO Mapping</b>	No



### 4.2.2 Object 6112h: AI Operating Mode

This object indicates the operating modes of the analogue input channels:

- 0: Channel off (not operating)
- 1: Normal Operation

<b>Index</b>	6112h
<b>Parameter Name</b>	AI Operating Mode
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Operating Mode 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Operating Mode 2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

## 4. The Object Dictionary

### 4.2.3 Object 6125h: AI Autozero

If the signature “zero” (0x6F72657A) is written to this object, the transmitter will autozero the actual attached process value.

The resulting offset value can be readout via object AI Scaling Offset 6127h.

EN

<b>Index</b>	6125h
<b>Parameter Name</b>	AI Autozero
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Autozero 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	wo
<b>PDO Mapping</b>	No

### 4.2.4 Object 6126h: AI Scaling Factor

This object offers the possibility to configure a scaling factor, which has effect on the process values.

Please note: depending on the settings of object 6132h (AI Decimal Digits PV), object 6131h (AI Physical Unit PV), scaling settings and the current value of object 7100h (AI Input FV) the value for object 7130h might reach its maximum, although the measured value is within the measurement range. In this case it is needed to switch to object 6130h (output as REAL32) or 9130h (output as INTEGER32).

<b>Index</b>	6126h
<b>Parameter Name</b>	AI Scaling Factor
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Scaling Factor 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Scaling Factor 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.2.5 Object 6127h: AI Scaling Offset

This object offers the possibility to configure an offset, which has effect on the process values.

The offset value must be specified as a floating point value in the actual configured physical unit (see object 6131h).

Please note: depending on the settings of object 6132h (AI Decimal Digits PV), object 6131h (AI Physical Unit PV), scaling settings and the current value of object 7100h (AI Input FV) the value for object 7130h might reach its maximum, although the measured value is within the measurement range. In this case it is needed to switch to object 6130h (output as REAL32) or 9130h (output as INTEGER32).

<b>Index</b>	6127h
<b>Parameter Name</b>	AI Scaling Offset
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Scaling Offset 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Scaling Offset 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.6 Object 6130h: AI Input PV

The analog input process value in REAL32 format.

Subindex 1 contains the pressure measurement value

Subindex 2 contains the temperature measurement value

<b>Index</b>	6130h
<b>Parameter Name</b>	AI Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

### 4.2.7 Object 6131h: AI Physical Unit PV

This object provides the physical unit of the transmitter.  
Any change of the physical unit has influence on the signal processing.

The pressure channel (analog input channel 1) provides the following units:

- 0x004E0000 = bar
- 0x00AB0000 = psi
- 0x06220000 = MPa

The temperature channel (analog input channel 2) provides the following units:

- 0x002D0000 = °C
- 0x00AC0000 = °F
- 0x00050000 = K

## 4. The Object Dictionary

Please note: depending on the settings of object 6132h (AI Decimal Digits PV), object 6131h (AI Physical Unit PV), scaling settings and the current value of object 7100h (AI Input FV) the value for object 7130h might reach its maximum, although the measured value is within the measurement range. In this case it is needed to switch to object 6130h (output as REAL32) or 9130h (output as INTEGER32).

EN

<b>Index</b>	6131h
<b>Parameter Name</b>	AI Physical Unit PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Physical Unit PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No
<b>Value Range</b>	0x004E0000 = bar 0x00AB0000 = psi 0x06220000 = MPa

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Physical Unit PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No
<b>Value Range</b>	0x002D0000 = °C 0x00AC0000 = °F 0x00050000 = K

### 4.2.8 Object 6132h: AI Decimal Digits PV

This object describes the number of decimal digits for interpreting the process values.

Example 1:

- Pressure range = 0..10 bar / decimal digits = 2 / physical unit = „bar“  
→ A process value of 924 corresponds to 9.24 bar.

Example 2:

- Temperature range = -40 ... 125°C / decimal digits = 1 / physical unit = „°C“  
→ A process value of 284 corresponds to 28.4 °C.

Please note: depending on the settings of object 6132h (AI Decimal Digits PV), object 6131h (AI Physical Unit PV), scaling settings and the current value of object 7100h (AI Input FV) the value for object 7130h might reach its maximum, although the measured value is within the measurement range. In this case it is needed to switch to object 6130h (output as REAL32) or 9130h (output as INTEGER32).

<b>Index</b>	6132h
<b>Parameter Name</b>	AI Decimal Digits PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Decimal Digits PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	rw
<b>PDO Mapping</b>	No
<b>Value Range</b>	0 - 5

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Decimal Digits PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	rw
<b>Default Value</b>	1
<b>PDO Mapping</b>	No
<b>Value Range</b>	0 - 5

### 4.2.9 Object 6133h: AI Interrupt Delta Input PV

Indicates the set delta on the analog input until an interrupt is triggered. If the change in PV is equal or greater delta input PV since the last PDO transmission, the PDO will be transmitted again.

If set to 0 the delta input PV interrupt is disabled.

<b>Index</b>	6133h
<b>Parameter Name</b>	AI Interrupt Delta Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Interrupt Delta Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No



<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Interrupt Delta Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.10 Object 6134h: AI Interrupt Lower Limit Input PV

Indicates the lower limit at which an interrupt is triggered.

<b>Index</b>	6134h
<b>Parameter Name</b>	AI Interrupt Lower Limit Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Interrupt Lower Limit Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Interrupt Lower Limit Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

## 4. The Object Dictionary

### 4.2.11 Object 6135h: AI AI Interrupt Upper Limit Input PV

Indicates the upper limit at which an interrupt is triggered.

EN

<b>Index</b>	6135h
<b>Parameter Name</b>	AI Interrupt Upper Limit Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Interrupt Upper Limit Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Interrupt Upper Limit Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.12 Object 6136h: AI Interrupt Hysteresis Input PV

Indicates the hysteresis on the analog input at which an interrupt is triggered.

<b>Index</b>	6136h
<b>Parameter Name</b>	AI Interrupt Hysteresis Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Interrupt Hysteresis Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Interrupt Hysteresis Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.13 Object 6148h: AI Span Start

This object indicates the lower limit of the expected PVs. PVs, which are lower than this limit, are marked as negative overloaded (see also object 6150h).

<b>Index</b>	6148h
<b>Parameter Name</b>	AI Span Start
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Span Start 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Span Start 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.2.14 Object 6149h: AI Span End

This object indicates the upper limit of the expected PVs. PVs, which are exceeding this limit, are marked as positive overloaded (see also object 6150h).

<b>Index</b>	6149h
<b>Parameter Name</b>	AI Span End
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Span End 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Span End 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.2.15 Object 6150h: AI Status

This object provides the status of the analog input channels as defined in the following:

- Bit 0: Not valid
- Bit 1: Positive overload
- Bit 2: Negative overload

The combination of bit 1 and bit 2 is not possible.

The following limits apply to the overload Limits:

- Positive overload: Measuring range end + 3.125% full scale
- Negative overload: Measuring range start -3.125% full scale

If the overload limits are reached, the following actions will be performed:

- EMCY message will be send: F001h – User Input Overload (see chapter 5)
- AI Input PVs will be set to the following values
  - 6130h: NAN (not a number)
  - 7130h: -32768 (minimum of int16)
  - 9130h: -2147483648 (minimum of int32)

## 4. The Object Dictionary

Example AI Status:

- 0x00: No failure
- 0x03: Positive overload and the AI Input PVs are not valid
- 0x05: Negative overload and the AI Input PVs are not valid

EN

<b>Index</b>	6150h
<b>Parameter Name</b>	AI Status
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Status 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Status 2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.2.16 Object 61A0h: AI Filter Type

Indicates the type of filter used for calculation. Following values are specified:

- 0: No filter applied
- 1: Moving average filter
- 2: Repeating average filter
- 3: Lowpass Bessel filter (not supported by sensor)
- 4: Lowpass butterworth filter (not supported by sensor)
- 5 to 99: reserved
- 100 to 255: manufacturer specific (not in use)

<b>Index</b>	61A0h
<b>Parameter Name</b>	AI Filter Type
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Filter Type 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Lower Limit</b>	0
<b>Higher Limit</b>	2
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.17 Object 61A1h: AI Filter Constant

Indicates the number of steps used for filter calculation. For more information please refer to the CiA documents.

<b>Index</b>	61A1h
<b>Parameter Name</b>	AI Filter Constant
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Filter Constant 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.18 Object 7100h: AI Input FV

The measured analog input raw values, ranging from 0 (measuring range start) to 10,000 (measuring range end).

Subindex 1 contains the field value of the measured pressure

Subindex 2 contains the field value of the measured temperature

<b>Index</b>	7100h
<b>Parameter Name</b>	AI Input FV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No



<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Input FV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Input FV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

### 4.2.19 Object 7130h: AI Input PV

The analog input process value in INTEGER16 format.

Subindex 1 contains the pressure measurement value

Subindex 2 contains the temperature measurement value

Please note: depending on the settings of object 6132h (AI Decimal Digits PV), object 6131h (AI Physical Unit PV), scaling settings and the current value of object 7100h (AI Input FV) the value for object 7130h might reach its maximum, although the measured value is within the measurement range. In this case it is needed to switch to object 6130h (output as REAL32) or 9130h (output as INTEGER32).

<b>Index</b>	7130h
<b>Parameter Name</b>	AI Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

### 4.2.20 Object 7133h: AI Interrupt Delta Input PV

Indicates the set delta on the analog input until an interrupt is triggered. If the change in PV is equal or greater delta input PV since the last PDO transmission, the PDO will be transmitted again.

If set to 0 the delta input PV interrupt is disabled.

<b>Index</b>	7133h
<b>Parameter Name</b>	AI Interrupt Delta Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Interrupt Delta Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Interrupt Delta Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.21 Object 7134h: AI Interrupt Lower Limit Input PV

Indicates the lower limit at which an interrupt is triggered.

<b>Index</b>	7134h
<b>Parameter Name</b>	AI Interrupt Lower Limit Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Interrupt Lower Limit Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Interrupt Lower Limit Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.22 Object 7135h: AI Interrupt Upper Limit Input PV

Indicates the upper limit at which an interrupt is triggered.

<b>Index</b>	7135h
<b>Parameter Name</b>	AI Interrupt Upper Limit Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Interrupt Upper Limit Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Interrupt Upper Limit Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.23 Object 7136h: AI Interrupt Hysteresis Input PV

Indicates the hysteresis on the analog input at which an interrupt is triggered.

<b>Index</b>	7136h
<b>Parameter Name</b>	AI Interrupt Hysteresis Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Interrupt Hysteresis Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Interrupt Hysteresis Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.24 Object 7148h: AI Span Start

This object indicates the lower limit of the expected PVs. PVs, which are lower than this limit, are marked as negative overloaded (see also object 6150h).

<b>Index</b>	7148h
<b>Parameter Name</b>	AI Span Start
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Span Start 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Span Start 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.2.25 Object 7149h: AI Span End

This object indicates the upper limit of the expected PVs. PVs, which are exceeding this limit, are marked as positive overloaded (see also object 6150h).

<b>Index</b>	7149h
<b>Parameter Name</b>	AI Span End
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Span End 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Span End 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.2.26 Object 9130h: AI Input PV

The analog input process value in INTEGER32 format.

Subindex 1 contains the pressure measurement value

Subindex 2 contains the temperature measurement value

<b>Index</b>	9130h
<b>Parameter Name</b>	AI Input PV
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER16
<b>Access</b>	ro
<b>PDO Mapping</b>	Yes

### 4.2.27 Object 9133h: AI Interrupt Delta Input PV

Indicates the set delta on the analog input until an interrupt is triggered. If the change in PV is equal or greater delta input PV since the last PDO transmission, the PDO will be transmitted again.

If set to 0 the delta input PV interrupt is disabled.

<b>Index</b>	9133h
<b>Parameter Name</b>	AI Interrupt Delta Input PV
<b>Object Type</b>	ARRAY



<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Interrupt Delta Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Interrupt Delta Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.28 Object 9134h: AI Interrupt Lower Limit Input PV

Indicates the lower limit at which an interrupt is triggered.

<b>Index</b>	9134h
<b>Parameter Name</b>	AI Interrupt Lower Limit Input PV
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Interrupt Lower Limit Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Interrupt Lower Limit Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.29 Object 9135h: AI Interrupt Upper Limit Input PV

Indicates the upper limit at which an interrupt is triggered.

<b>Index</b>	9135h
<b>Parameter Name</b>	AI Interrupt Upper Limit Input PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Interrupt Upper Limit Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Interrupt Upper Limit Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.30 Object 9136h: AI Interrupt Hysteresis Input PV

Indicates the hysteresis on the analog input at which an interrupt is triggered.

<b>Index</b>	9136h
<b>Parameter Name</b>	AI Interrupt Hysteresis Input PV
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Interrupt Hysteresis Input PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Interrupt Hysteresis Input PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.2.31 Object 9148h: AI Span Start

This object indicates the lower limit of the expected PVs. PVs, which are lower than this limit, are marked as negative overloaded (see also object 6150h).

<b>Index</b>	9148h
<b>Parameter Name</b>	AI Span Start
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Span Start 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Span Start 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.2.32 Object 9149h: AI Span End

This object indicates the upper limit of the expected PVs. PVs, which are exceeding this limit, are marked as positive overloaded (see also object 6150h).

<b>Index</b>	9149h
<b>Parameter Name</b>	AI Span End
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	AI Span End 1
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	AI Span End 2
<b>Object Type</b>	VAR
<b>Data Type</b>	INTEGER32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.3 Object Dictionary – Manufacturer Specific Area

#### 4.3.1 Object 2010h: Measuring Range Start PV

This object provides the measuring range start of the transmitter in the corresponding original physical unit (see 2012h). The value can also be found on the transmitter device type label.

<b>Index</b>	2010h
<b>Parameter Name</b>	Measuring Range Start PV
<b>Object Type</b>	ARRAY

14547342.01 03/2022 EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Measuring Range Start PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Measuring Range Start PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.3.2 Object 2011h: Measuring Range End PV

This object provides the measuring range end of the transmitter in the corresponding original physical unit (see 2012h). The value can also be found on the transmitter type label.

<b>Index</b>	2011h
<b>Parameter Name</b>	Measuring Range End PV
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Measuring Range End PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Measuring Range End PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.3.3 Object 2012h: Original Physical Unit PV

This object provides the original physical unit of the transmitter. The value can also be found on the device type label.

For more details on the physical unit refer to 4.2.7 (Physical unit AI PV)

<b>Index</b>	2012h
<b>Parameter Name</b>	Original Physical Unit PV
<b>Object Type</b>	ARRAY



<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Original Physical Unit PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Original Physical Unit PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.3.4 Object 2020h: Minimum Value PV

This object provides the minimum process value of the AI Input PV (see 6130h) in the corresponding physical unit (see 6131h) since power up.

<b>Index</b>	2020h
<b>Parameter Name</b>	Minimum Value PV
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Minimum Value PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Minimum Value PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.3.5 Object 2021h: Maximum Value PV

This object provides the maximum process value of the AI Input PV (see 6130h) in the corresponding physical unit (see 6131h) since power up.

<b>Index</b>	2021h
<b>Parameter Name</b>	Maximum Value PV
<b>Object Type</b>	ARRAY

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Maximum Value PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Maximum Value PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	REAL32
<b>Access</b>	ro
<b>PDO Mapping</b>	No

### 4.3.6 Object 2022h: Reset Min Max Value PV

This object resets the Minimum Maximum PV Counter (see 2020h / 2021h). The signature “zero” ('z' = 0x7A; 'e' = 0x65; 'r' = 0x72; 'o' = 0x6F) needs to be add to the transmitted value to reset the specific PV Counter.

<b>Index</b>	2022h
<b>Parameter Name</b>	Reset Min Max Value PV
<b>Object Type</b>	ARRAY

## 4. The Object Dictionary

EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	2
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Reset Min Max Value PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	wo
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	2
<b>Parameter Name</b>	Reset Min Max Value PV 2
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	wo
<b>PDO Mapping</b>	No

### 4.3.7 Object 2023h: Damping PV

This object is a helper function for setting the filter constant (see 61A1h) for a given damping value in milliseconds, in case of an enabled moving average filter (see 61A0h).

The damping value is defined as a time interval between the step change of the measurement signal and the point in time at which the output signal deviates by no more than 1% tolerance from the value of its final condition.

<b>Index</b>	2023h
<b>Parameter Name</b>	Damping PV
<b>Object Type</b>	ARRAY

14547342.01 03/2022 EN

<b>Sub-Index</b>	0
<b>Parameter Name</b>	Number of entries
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED8
<b>Access</b>	ro
<b>Default Value</b>	1
<b>PDO Mapping</b>	No

<b>Sub-Index</b>	1
<b>Parameter Name</b>	Damping PV 1
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED16
<b>Access</b>	wo
<b>PDO Mapping</b>	No

### 4.3.8 Object 2320h: Node-ID

In addition to the LSS Services, the Node-ID can also be configured via this object. To prevent accidental adjustment, the signature “set” (‘s’ = 0x73; ‘e’ = 0x65; ‘t’ = 0x74) needs to be add to the transmitted value.

Example:

To set the Node-ID to 3, the following Value needs to be written to the object:  
0x74657303

Changes take effect after executing the Store Parameters object 1010h and a following Reset All Nodes or power on reset. Restoring the default parameters via Object 1011h has no effect to the Node-ID, that means the Node-ID remains untouched

<b>Index</b>	2320h
<b>Parameter Name</b>	Node-ID
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

### 4.3.9 Object 2321h: Bitrate

In addition to the LSS Services, the bitrate can also be configured via this object. To prevent accidental adjustment, the signature “set” ('s' = 0x73; 'e' = 0x65; 't' = 0x74) needs to be added to the transmitted value.

EN

Example:

To set the bitrate to 250 kBit/s, the following Value needs to be written to the object: 0x74657303

Changes take effect after executing the Store Parameters object 1010h and a following Reset All Nodes or power on reset. Restoring the default parameters via Object 1011h has no effect to the Node-ID, that means the Node-ID remains untouched.

Following baudrate settings are supported:

- 0: 1MBaud
- 1: 800 kBaud
- 2: 500 kBaud
- 3: 250 kBaud
- 4: 125 kBaud
- 5: not supported
- 6: 50 kBaud
- 7: 20 kBaud
- 8: not supported
- 9: Automatic bitrate detection

Please do note, in case of automatic bitrate detection the object 2321h will contain the detected bitrate code (instead of 9 for automatic bitrate detection).

<b>Index</b>	2321h
<b>Parameter Name</b>	Bitrate
<b>Object Type</b>	VAR
<b>Data Type</b>	UNSIGNED32
<b>Access</b>	rw
<b>PDO Mapping</b>	No

## 5. Emergency Object (EMCY)

The emergency objects (EMCY) are automatically sent when an error occurs. An EMCY object is transmitted only one time per “error event”. As long as the error is pending, no new EMCY message with the same Emergency Error Code will be sent. In case that an Emergency Error Code is sent the correspondent Bit in the Error Register (1001h) will be set.

## 5. Emergency Object (EMCY)

EN

The following Emergency Error Codes are supported:

Emergency Error Code	Description	Affected Bit Error Register (1001h)
0000h	Error Reset or No Error	
1000h	Generic Error	Bit 7 (Manufacture specific)
4200h	Device Temperature	Bit 3 (Temperature)
5030h	Sensor Fault	Bit 7 (Manufacture specific)
8100h	Communication – generic	Bit 4 (Communication error)
8110h	CAN overrun (objects lost)	Bit 4 (Communication error)
8120h	CAN in error passive mode	Bit 4 (Communication error)
8140h	CAN recovered from bus off	Bit 4 (Communication error)
8150h	CAN-ID collision	Bit 4 (Communication error)
8240h	Unexpected SYNC data length	Bit 4 (Communication error)
F001h	User Input Overload	Bit 7 (Manufacture specific)

Hint: Bit 0 of Error Register (1001h) signals that any error is occurred. That means if any other bit of the register is set (e.g. Bit 7 (Manufacture specific Error)) the Bit 0 (Generic Error) is also set.

Error Register:

Contains the current contents of object 1001h Error Register

### Detailed Emergency error code description:

#### ■ 1000h – Generic Error

Internal device error during start up or runtime. Please restart the device (power off → power on). If the error is still present, please exchange the device.

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
0	0	0	0	0

Manufacturer specific Error Fields are 0.

#### ■ 4200h – Device Temperature

The Device Temperature is too low (-50°C) or high (+110°C). Please check the environmental conditions

Byte3	Byte4	Byte5	Byte6	Byte7
2 (AI Channel Number)	AI status (6150h)	0	0	0

Manufacturer specific error field:

## 5. Emergency Object (EMCY)

### ■ 5030h – Sensor Fault

The pressure sensor is defect. Please exchange the device.

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
1 (AI Channel Number)	AI status (6150h)	0	0	0

### ■ 8100h – Communication – generic

The CAN Controller goes into the warning state. Please check the bus connection, baudrate, termination, etc.

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
CAN error state (see table below)	CAN error type (see table below)	CAN error receive counter	CAN error transmit counter	0

### ■ 8110h – CAN overrun (objects lost)

Not all CAN messages could be received because of an buffer overflow

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
0	0	0	0	0

### ■ 8120h – CAN in error passive mode

The CAN Controller entered the passive state. Please check the bus connection, baudrate, termination, etc.

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
CAN error state (see table below)	CAN error type (see table below)	CAN error receive counter	CAN error transmit counter	0

### ■ 8140h - CAN recovered from bus off

The CAN controller was in the bus off state and is now again active. Please check the bus connection, baudrate, termination, etc.

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
0	0	0	0	0



## 5. Emergency Object (EMCY)

EN

### ■ 8150h - CAN-ID collision

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
Number of message buffer	0	0	0	0

### ■ 8240h - Unexpected SYNC data length

The data length code of the SYNC message is wrong or have a not expected value

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
0	0	0	0	0

### ■ F001h – User Input Overload

The pressure at the device is too high or too low (exceeding  $\pm 3,125\%$  of measuring range). Please check your system. A too high/low pressure can damage the device or the measurement accuracy can not be guaranteed longer more – also if the overload is no longer present.

Manufacturer specific error field:

Byte3	Byte4	Byte5	Byte6	Byte7
1 (AI Channel Number)	AI status (6150h)	0	0	0

### CAN error states:

CAN error state number	Description
0	CAN controller is in STOPPED mode
1	CAN controller is in SLEEP mode
2	CAN controller has errors detected (ACTIVE mode)
3	CAN controller has errors detected (ACTIVE mode, but warning level is reached)
4	CAN controller has errors detected (PASSIVE mode)
5	CAN controller went into Bus-Off
10	General failure of physical layer detected
11	Fault on CAN-H detected (Low Speed CAN)
12	Fault on CAN-L detected (Low Speed CAN)

**CAN error types:**

CAN error type number	Description
0	No error
1	Bit 0 error
2	Bit 1 error
3	Stuff bit error
4	Format error
5	CRC error
6	Acknowledge error

EN

**6. References**

<b>CiA 301 V 4.2.0</b>	CANopen application layer and communication profile
<b>CiA 303-1 V1.9.0</b>	CANopen Recommendation: Cabling and Connector Pin Assignment
<b>CiA 303-2 V1.5.0</b>	CANopen Recommendation: Representation of SI units and prefixes
<b>CiA 404-1 V2.1.0</b>	Device profile for measuring devices and closed-loop controllers Part 1: Generic objects and generic PDO mapping
<b>CiA 305 V3.0.0</b>	Layer setting services (LSS) and protocols

**7. Change log**

Document version	Reason for change	Date
1.0	Initial release	



WIKA subsidiaries worldwide can be found online at [www.wika.com](http://www.wika.com).  
WIKA-Niederlassungen weltweit finden Sie online unter [www.wika.de](http://www.wika.de).



**WIKAI Alexander Wiegand SE & Co. KG**

Alexander-Wiegand-Strasse 30

63911 Klingenberg • Germany

Tel. +49 9372 132-0

Fax +49 9372 132-406

[info@wika.de](mailto:info@wika.de)

[www.wika.de](http://www.wika.de)