*PBM4-C02*

# PBM4 With CANopen

Last Update: 29 April 2019

# Table of Content

# 1 History

**Tab 1** Document history

| Vers. | Date | Note |
|---|---|---|
| 0.1 | 23 April 2019 | Initial draft version |
| 0.2 | 23 April 2019 | Initial draft version with CoC (chapter Declaration of Conformity) |
| 1.0 | 29 April 2019 | First edition |

# 2 General

The pressure transmitter PBM4 with CANopen measures the physical quantity pressure. The range depends on the sensor which is used in the transmitter and is 25 … 800 bar. The measured value is transmitted on the CAN-Bus with the CANopen protocol. The transmitter takes 1000 samples per second, does filtering and converts the raw value into the output format.
The CAN 2.0B interface is able to run up to a speed of 1 Mbit/sec with 11-bit and 29-bit identifiers.
The CAN protocol complies with the CANopen specification DS301, the pressure transmitter function is presented by the CANopen device profile DS404. The possible configurations can be set with the object dictionary. Heartbeat and emergency messages guarantee high reliability.
With the “Layer setting services” (LSS, DSP305 V2.0), the desired bit rate and node ID can be set easily.

# 3 CAN Interface

The device includes a Full CAN controller specified to CAN 2.0B. The physical layer of the 2-wire interface is specified according to ISO 11898. The wires are protected against short-circuit. By adjusting the rise and fall times of the CAN signals, the noise emission is minimized. The bus termination resistor is not included in the device.

# 4 PBM4 CANopen Specification

## 4.1 Supply Voltage +$U_S$

| | |
|---|---|
| • Supply voltage: | 9 … 36 VDC, protected against reverse polarity |
| • Current consumption at $U_S$ = 24 VDC: | I < 50 mA typical, IMAX < 100 mA |

## 4.2 CAN Interface

| | |
|---|---|
| • Physical layer: | 2-wire interface, 5 V level according to ISO 11898<br>Protected against short-circuit |
| • max. Bit rate: | 1 Mbit/s |
| • Signal rise time: | Bit rate < 125 kbit/s, 12 V/µs (without bus)<br>Bit rate ≥ 125 kbit/s > 24 V/µs (without bus) |
| • Bus termination: | external |
| • Protocol: | CANopen DS301, Device Profile DS404 |

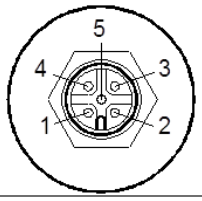
## 4.3 Environment

| | |
|---|---|
| • EMC: | noise emission: according to EN 50 081-2<br>noise immunity: according to EN 50 082-2 |
| • Operating temperature: | -40 … 125°C |
| • Media temperature: | -40 … +150°C |

## 4.4 Connector Pin Assignment (CiA DR303-1)

The pin connection for the used 5 pole M12 connector is shown.

**Tab 2** Pin assignment

| Pin | Assignment | Figure |
|---|---|---|
| 1 | CAN shield, PE |  |
| 2 | +$U_B$, +24 VDC | |
| 3 | GND, 0 V | |
| 4 | CAN_H, CAN+ | |
| 5 | CAN_L, CAN- | |

# 5 CANopen Communication

## 5.1 Summary of the CANopen functions

| | |
|---|---|
| CANopen type: | NMT slave |
| Network bootup: | minimum bootup |
| COB Id placing: | pre-defined connection set, SDO |
| Node ID: | object (specific entry) |
| Bitrate: | object (specific entry) |
| Number of PDOs: | PDO1 synchronous, asynchronous<br>PDO-mapping configurable |
| Emergency message: | supported |
| Heartbeat: | supported |
| Device profile: | DS404 |
| Layer setting services: | supported |
| | |
| Default Settings: | Bitrate 125 kbit/s<br>Node ID 1 |

## 5.2 Object Dictionary: Communication Profile

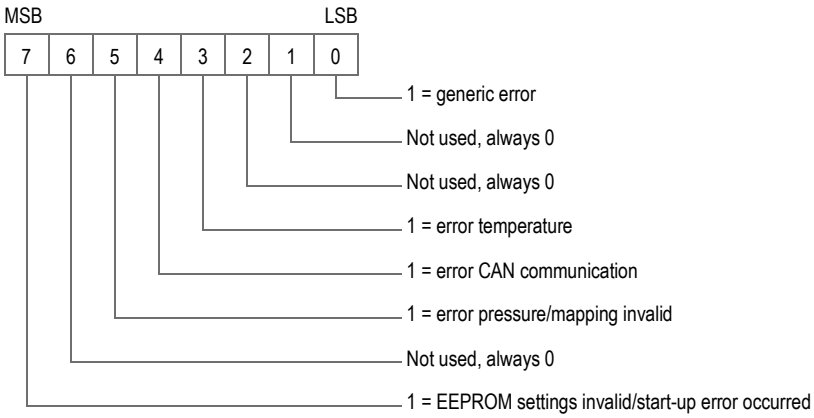
**Tab 3** Communication profile

| Index (HEX) | Sub Index | Name | Type | Access | Default | Comment |
|---|---|---|---|---|---|---|
| 1000 | 00 | Device Type | Unsigned32 | ro | 0x00020194 | DSP404 analog input block |
| 1001 | 00 | Error Register | Unsigned8 | ro | 0x00 | See *Tab 4*, page *9* |
| 1003 | | Pre-defined Error Field | | | | |
| | 00 | Number of errors | Unsigned8 | rw | 0x01 | Write 0: clear errors |
| | 01 | Standard Error Field | Unsigned32 | ro | | See *0*, page *9* |
| 1005 | 00 | COB-ID SYNC | Unsigned32 | rw | 0x80 | |
| 1008 | 00 | Manufacturer Device Name | String | ro | “PBM4-CAN-open” | |
| 1009 | 00 | Manufacturer Hardware Version | String | ro | “x.xxrx” | |
| 100A | 00 | Manufacturer Software Version | String | ro | “x.xxrx” | |
| 1010 | | Store parameters | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Restore default parameters | Unsigned32 | rw | 0x01 | Data will be saved with the command 0x65766173 (ASCII: “save“) |

| Index (HEX) | Sub Index | Name | Type | Ac-cess | Default | Comment |
|---|---|---|---|---|---|---|
| 1011 | | Restore default parame-ters | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Restore default parame-ters | Unsigned32 | rw | 0x01 | Default values will be re-stored with the command 0x64616F6C (ASCII: “load“) Reset of device required |
| 1014 | 00 | COB ID Emergency message | Unsigned32 | ro | 0x81 | 0x00000080 + Node ID |
| 1016 | | Consumer heartbeat time | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Consumer heartbeat time | Unsigned32 | rw | 0 | |
| 1017 | 00 | Producer heartbeat | Unsigned16 | rw | 0 | |
| 1018 | | Identity object | | | | |
| | 00 | Number of entries | Unsinged8 | ro | 4 | |
| | 01 | Vendor ID | Unsigned32 | ro | 0x23D | Baumer Vendor ID |
| | 02 | Product Code | Unsigned32 | ro | | Baumer article number |
| | 03 | Revision number | Unsigned32 | ro | | Firmware version |
| | 04 | Serial number | Unsigned32 | ro | | Baumer internal serial num-ber |
| 1800 | | Transmit PDO1 parame-ter | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 5 | |
| | 01 | COB ID used by PDO | Unsigned32 | rw | 0x181 | (0x00000180 + Node ID) |
| | 02 | Transmission type | Unsigned8 | rw | 0x01 | Only 0x01 (sync) or 0xFF (async) with delta and/or event timer |
| | 03 | Inhibit time | Unsigned16 | rw | 0 | |
| | 04 | Reserved | Unsigned8 | rw | 0 | |
| | 05 | Event timer | Unsinged16 | rw | 0x3E8 | |
| 1A00 | | Transmit PDO1 mapping | | | | |
| | 00 | Number of entries | Unsigned8 | rw | 2 | |
| | 01 | PDO mapping for the 1. application object to be mapped | Unsigned32 | rw | 0x91300120 | Pressure as int32: x91300120<br>Pressure as float32: 0x61300120<br>Status pressure as uint8: 0x61500108<br>Temperature as int32: 0x91300220 |
| | 02 | PDO mapping for the 2. application object to be mapped | Unsigned32 | rw | 0x61500108 | |
| | 03 | PDO mapping for the 3. application object to be mapped | Unsigned32 | rw | 0 | |

| Index (HEX) | Sub Index | Name | Type | Access | Default | Comment |
|---|---|---|---|---|---|---|
| | 04 | PDO mapping for the 4. application object to be mapped | Unsigned32 | rw | 0 | Temperature as float32: 0x61300220<br>Status temperature as uint8: 0x61500208<br>Meaning of status bits (if set):<br>Bit 0: pressure/temperature value invalid<br>Bit 1: positive overload<br>Bit 2: negative overload |
| 1F80 | 00 | NMT start-up | Unsigned32 | rw | 4 | 0x00000004: The NMT master has to start the NMT slave.<br>0x00000008: NMT slave shall enter the NMT state Operational after the NMT state Initialisation autonomously (self-starting). |

**Tab 4** Error register (Index 1001H)




**Tab 5** Standard error field (Index 1003H, 01)

| MSB Bit 31 … 24 | Bit 23 … 16 | Bit 15 … 0 LSB |
|---|---|---|
| Not used, always 0 | Error register (Index 1001H) (see *Tab 4*) | Error Code:<br>0x8100 communication error<br>0x6161 internal software error (EEPROM settings invalid)<br>0x6363 PDO mapping error<br>0x6300 data error (start-up error)<br>0x4000 temperature error<br>0xF011 pressure error |

## 5.3 Object Dictionary: Manufacturer Specific Profile

**Tab 6** Manufacturer specific profile

| Index (HEX) | Sub Index | Name | Type | Access | Default | Comment |
|---|---|---|---|---|---|---|
| 3000 | 00 | Pressure overflow counter | Unsigned16 | rw | 0 | Counter for overpressure |
| 3001 | 00 | Pressure underflow counter | Unsigned16 | rw | 0 | Counter for under-pressure |
| 3002 | 00 | Temperature overflow counter | Unsinged16 | rw | 0 | Counter for over-temperature |
| 3003 | 00 | Temperature underflow counter | Unsigned16 | rw | 0 | Counter for under-temperature |
| 4E00 | | | | | | Available with firmware V1.11r0 and higher |
| | 00 | Number of entries | Unsinged8 | ro | 4 | |
| | 01 | Serial number, part 1 | Unsinged8 | ro | | Part 1, e.g: 11 |
| | 02 | Serial number, part 2 | Unsinged32 | ro | | Part 2, e.g: 223344 |
| | 03 | Serial number, part 3 | Unsinged16 | ro | | Part 3, e.g: 5566 |
| | 04 | Serial number, complete | String | ro | | E.g: 11.223344.5566 |
| 4F00 | 00 | Bit rate | Unsinged8 | rw | 4 | See *0*<br>*Bit* rates<br>Changes take effect after reset node or power on. |
| 4F01 | 00 | Node ID | Unsigned8 | rw | 1 | 1 … 127<br>Changes take effect after reset node or power on. |

The indices 5, 6 and 7 have been changed from firmware release V1.08r6 to V1.09r0 (see OD-entry 0x100A). In the following table the corresponding indices can be selected:

**Tab 7** Bit rates

| Index | [Bit rate] = kbit/s | |
|---|---|---|
| | Through firmware V1.08r6 | Firmware from V1.09r0 |
| 0 | 1000 | 1000 |
| 1 | 800 | 800 |
| 2 | 500 | 500 |
| 3 | 250 | 250 |
| 4 | 125 | 125 |
| 5 | 50 | - |
| 6 | 20 | 50 |
| 7 | - | 20 |
| 100 | 100 | 100 |
| 101 | 40 | 40 |
| 102 | 400 | 400 |

*Hint*: Bit rates 40 kbit/s and 400 kbit/s are only supported by firmware V1.08r0 and higher (see OD-entry 0x100A).

## 5.4 Object Dictionary: Device Profile

**Tab 8** Device profile

| Index (HEX) | Sub Index | Name | Type | Ac-cess | Default | Comment |
|---|---|---|---|---|---|---|
| 6110 | | Ai_Sensor_Type | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_Sensor_Type_1 | Unsigned16 | ro | 0x5A | 90 = pressure sensor |
| | 02 | Ai_Sensor_Type_2 | Unsigned16 | ro | 0x64 | 100 = temperature sensor |
| 6124 | | Ai_Input_Offset | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Ai_Input_Offset_1 | Float32 | rw | 0 | pressure offset; will be added to the current pressure value |
| 6125 | | Ai_Input_Autozero | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Ai_Input_Autozero_1 | Unsigned32 | wo | | Autozero for pressure 0x6F72657A (ASCII: “zero“) |
| 6130 | | Ai_Input_PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_Input_PV_1 | Float32 | ro | | actual pressure value |
| | 02 | Ai_Input_PV_2 | Float32 | ro | | actual temperature value |

| Index (HEX) | Sub Index | Name | Type | Ac-cess | Default | Comment |
|---|---|---|---|---|---|---|
| 6131 | | Ai_Physical_Unit_PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 0x2 | |
| | 01 | Ai_Physical_Unit_PV_1 | Unsigned32 | rw | 0x00220000 | Pressure unit:<br>0x00AB0000: psi<br>0x004E0000: bar<br>0x00220000: Pa<br>Prefixes (only for psi, bar):<br>0xFF__0000: $10^{-1}$ (deci)<br>0xFE__0000: $10^{-2}$ (centi)<br>0xFD__0000: $10^{-3}$ (milli)<br>Hint: also dependents on "decimal digits", Index 0x6132 |
| | 02 | Ai_Physical_Unit_PV_2 | Unsigned32 | rw | 0x002D0000 | Temperature unit:<br>0x002D0000: °C<br>0x00AC0000: °F<br>0x00050000: K |
| 6132 | | Ai_Decimal_Digits_PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_Decimal_Digits_PV_1 | Unsigned8 | rw | 0 | Range depends on the physical unit and prefix:<br>[0…3] with unit psi<br>[0…5] with unit bar<br>[0] with unit Pa |
| | 02 | Ai_Decimal_Digits_PV_2 | Unsigned8 | rw | 0 | [0…5] |
| 6133 | | Ai interrupt delta input PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | PV_1 | Float32 | rw | 0 | Pressure<br>(default 0: disabled) |
| | 02 | PV_2 | Float32 | rw | 0 | Temperature<br>(default 0: disabled) |
| 6134 | | Ai interrupt lower limit input PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | PV_1 | Float32 | rw | -2147483648 | Pressure<br>(-2147483648: disabled) |
| | 02 | PV_2 | Float32 | rw | -2147483648 | Temperature<br>(-2147483648: disabled) |
| 6135 | | Ai interrupt upper limit input PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | PV_1 | Float32 | rw | 2147483520 | Pressure<br>(2147483520: disabled) |
| | 02 | PV_2 | Float32 | rw | 2147483520 | Temperature<br>(2147483520: disabled) |

| Index (HEX) | Sub Index | Name | Type | Ac-cess | Default | Comment |
|---|---|---|---|---|---|---|
| 6148 | | Ai_span_start | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_span_start_1 | Float32 | ro | | Pressure |
| | 02 | Ai_span_start_2 | Float32 | ro | | Temperature |
| 6149 | | Ai_span_end | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_span_end_1 | Float32 | ro | | Pressure |
| | 02 | Ai_span_end_2 | Float32 | ro | | Temperature |
| 6150 | | Ai status | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | PV_1 | Unsigned8 | ro | 0 | (bit set => error)<br>Pressure:<br>Bit 0 = 1: invalid<br>Bit 1 = 1: pos. overload<br>Bit 2 = 1: neg. overload |
| | 02 | PV_2 | Unsigned8 | ro | 0 | (bit set => error)<br>Temperature:<br>Bit 0 = 1: invalid<br>Bit 1 = 1: pos. overload<br>Bit 2 = 1: neg. overload |
| 61A0 | | Ai_filter_type | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Ai_filter_type_1 | Unsigned8 | rw | 1 | 0: no filter<br>1: moving average<br>2: repeating average |
| 61A1 | | Ai_filter_constant | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Ai_filter_constant_1 | Unsigned8 | rw | 30 | |
| 7100 | | Ai_input_FV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_input_FV_1 | Integer16 | ro | | act. ADC pressure value |
| | 02 | Ai_input_FV_2 | Integer16 | ro | | act. ADC temperature value |
| 9124 | | Ai_input_offset | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 1 | |
| | 01 | Ai_input_offset_1 | Integer32 | rw | 0 | Offset for pressure |
| 9130 | | Ai_input_PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_input_PV_1 | Integer32 | ro | | actual pressure value |
| | 02 | Ai_input_PV_2 | Integer32 | ro | | actual temperature value |

| Index (HEX) | Sub Index | Name | Type | Access | Default | Comment |
|---|---|---|---|---|---|---|
| 9133 | | Ai interrupt delta input PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | PV_1 | Integer32 | rw | 0 | Pressure (default 0: disabled) |
| | 02 | PV_2 | Integer32 | rw | 0 | Temperature (default 0: disabled) |
| 9134 | | Ai interrupt lower limit input PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | PV_1 | Integer32 | rw | -2147483648 | Pressure (-2147483648: disabled) |
| | 02 | PV_2 | Integer32 | rw | -2147483648 | Temperature (-2147483648: disabled) |
| 9135 | | Ai interrupt upper limit input PV | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | PV_1 | Integer32 | rw | 2147483520 | Pressure (2147483520: disabled) |
| | 02 | PV_2 | Integer32 | rw | 2147483520 | Temperature (2147483520: disabled) |
| 9148 | | Ai_span_start | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_span_start_1 | Integer32 | ro | | Min. pressure |
| | 02 | Ai_span_start_2 | Integer32 | ro | | Min. operating temperature |
| 9149 | | Ai_span_end | | | | |
| | 00 | Number of entries | Unsigned8 | ro | 2 | |
| | 01 | Ai_span_end_1 | Integer32 | ro | | Max. pressure |
| | 02 | Ai_span_end_2 | Integer32 | ro | | Max. operating temperature |

## 5.5 Configuration of the transmit PDO

This chapter describes the configuration of the transmit PDO. The PBM4 CANopen supports the dynamic mapping with firmware release 1.10r0 and higher (see OD-entry 0x100A). Older firmware versions support only a fixed number of mapped objects (= 2).

*Dynamic mapping*

The PDO configuration is done by the OD entry TPDO1 mapping (index 0x1A00) and its sub-indices. The sub index 1 defines the first value (lower position) transmitted by the PDO. The sub-index 2 defines the second value, the sub index 3 the third and the sub index 4 the fourth value transmitted by the PDO. If not all values are used, the upper sub-indices must be set to 0.

To change the mapping, the following procedure must be observed:

1. Set the “Nr. of mapped objects” (0x1A00/0x00) to 0. => The PDO is deactivated.
2. Set the desired mapping values (0x1A00/0x01…0x04).
3. Set the “Nr. of mapped objects” (0x1A00/0x00) to the desired number of mapping objects.

*Default mapping*

The default values of these sub-indices are 0x91300120 (sub-index 1), 0x61500108 (sub-index 2) and 0 (sub-index 3 and 4):

**Tab 9** Default values

| Name | Index | Sub-index | | | | Value |
|---|---|---|---|---|---|---|
| → TPD01 Mapping – Nr Of Mapped Obj | 0x1a00 | 0x00 | | | | 0x2 |
| → TPD01 Mapping – value 1 | 0x1a00 | 0x01 | | | | 0x91300120 |
| → TPD01 Mapping – value 2 | 0x1a00 | 0x02 | | | | 0x61500108 |
| → TPD01 Mapping – value 3 | 0x1a00 | 0x03 | | | | 0x0 |
| → TPD01 Mapping – value 4 | 0x1a00 | 0x04 | | | | 0x0 |

That means:

The first value which will be sent by the transmit PDO is the value of the OD index 0x9130 with the sub index 0x01 and the length 0x20 bits (→ 0x91300120). It is the pressure value (signed integer 32 bit).

The second value of the transmit PDO is the OD index 0x6150 with the sub index 0x01 and the length 0x08 bits (→ 0x61500108). It is the pressure status (unsigned integer 8 bit).

*Transmit PDO (example)*

So the transmit PDO may look like the following message (example):

**Tab 10** Example transmit PDO

| | ID | DLC | Data | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 6 | Byte 7 |
| TPDO | 0x180 + Node ID | 5 | 0xA0 (pres. LSB) | 0x86 | 0x01 | 0x00 (pres. MSB) | 0x00 (pres. Status | Not available | |

This example shows a pressure value of 100000 (= 0x186A0) and the status 0 (valid).

*Units*

The units of the sent values are defined by the settings of the following object dictionary entries:

- *AI Physical unit PV* (index 0x6131): these specify the physical units of the values of indices 0x6130 and 0x9130.
- *AI Decimal digits PV* (index 0x6132): these specify the decimal digits of the integer values of the index 0x9130. 1 means that the value is multiplied by 10, 2 means multiplied by 100, …

The possible settings can be seen in chapter *5.4 Object Dictionary: Device Profile*, page *11*.

*Hint*

Only the following object dictionary indices are map able:

- index 0x6130, sub index 0x01 (pressure, float32)
- index 0x6130, sub index 0x02 (temperature, float32)
- index 0x6150, sub index 0x01 (pressure status)
- index 0x6150, sub index 0x02 (temperature status)
- index 0x9130, sub index 0x01 (pressure, integer32)
- index 0x9130, sub index 0x02 (temperature, integer32)

*Example mapping:*

To get the pressure value with float 32 bit and the temperature value with signed integer 32 bit, the sub index 1 has to be set to 0x61300120 (0x6130, 0x01, 0x20) and the sub index 2 to 0x91300220 (0x9130, 0x02, 0x20).

## 5.6 Emergency Message

Emergency messages show an internal device error. If the error situation for the de-vice has changed, it will send an emergency message with the current error code.
An error code 0x0000 shows that all errors are removed.
The current error situation could be read out with the object profile entry “Predefined Error Field” index 0x1003, sub index 1.
The COB-ID of an emergency message is shown in the communication profile of the object dictionary, index 0x1014 (= 0x80 + Node ID).

**Tab 11** Construction of the emergency message

| Data | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Error Code LSB | Error Code MSB | Error Register (Index 0x1001) | Not used | | | | |

**Tab 12** Error Codes

| Error Code | Meaning |
|---|---|
| 0x8100 | Error CAN-communication |
| 0x6161 | internal Software-error (EEPROM settings invalid) |
| 0x6363 | PDO-mapping error |
| 0x6300 | Data-error (start-up error) |
| 0x4000 | Error temperature |
| 0xF011 | Error pressure |

# 6 Layer setting services

The PBM4 with CANopen (V1.09r0 and higher) supports the layer setting services. These services and protocols are used to inquire the settings of the LSS address (object 0x1018), the bit rate and the node ID. Also the bit rate and the node ID could be changed by the LSS.

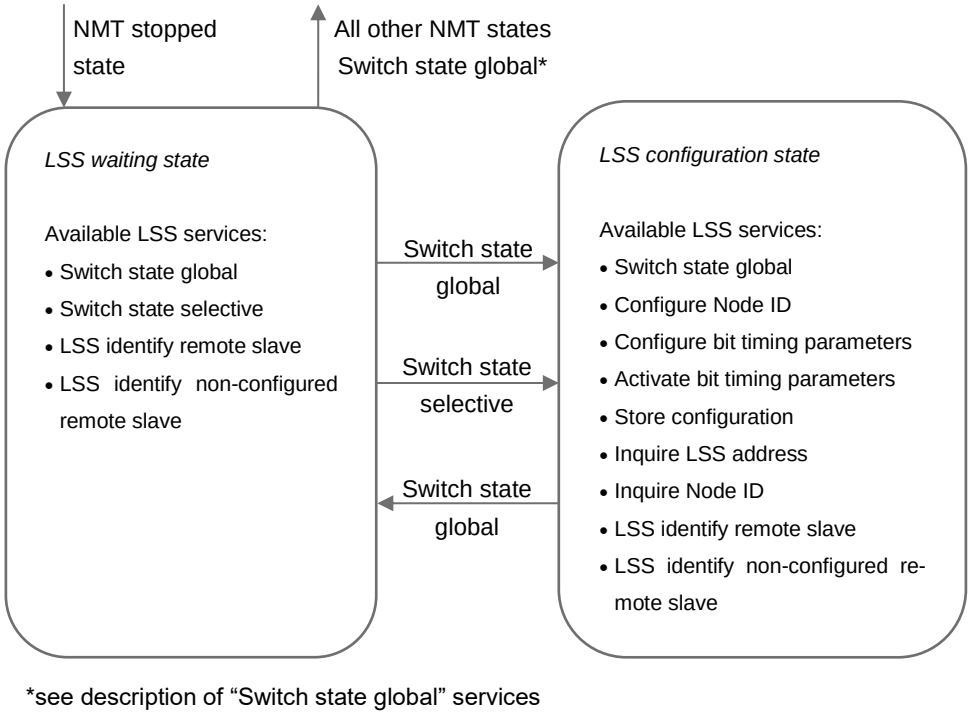



**Fig 1** Layer setting services

Some requirements/hints must be observed when using the LSS
- The producer heartbeat time must be 0 (= default; object 0x1017)
- The PBM4 with CANopen must be in NMT stopped state
- In LSS configuration state, no NMT-command will be executed
- Only a stored bit rate and Node ID will appear in the object dictionary (0x4F00 and 0x4F01)

The LSS address consists of four values:
- Vendor-Id: Object dictionary index 0x1018, sub-index 1: always 0x23D
- Product-code: Object dictionary index 0x1018, sub-index 2: order number of this PBM4 with CANopen (BCD)
- Revision-number: Object dictionary index 0x1018, sub-index 3: software version of this PBM4 with CANopen (BCD)
- Serial-number: Object dictionary index 0x1018, sub-index 4: a unique serial number

## 6.1 Supported services

All services of DSP305 V2.0 can be used. The supported parameters of the services can be found in this section. The CAN identifiers are reserved for LSS:

- 0x7E5 for commands from LSS master
- 0x7E4 for answers from LSS slave

**Tab 13** Switch state global

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | mode | reserved | reserved | reserved | reserved | reserved | reserved |

Command specifier request: 0x04

Mode: 0x00, switches to waiting state

0x01 switches to configuration state

*Hints*

- After storing a new node ID, the “Switch state global” service with the parameter “switches to waiting state” will activate the last stored node ID and it will be used immediately. So after that, the PBM4 with CANopen will transmit the bootup-message and stays in NMT preoperational state.
- Once the LSS configuration state has been left, all not stored data is no longer available.

**Tab 14** Switch state selective

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | data LSB | data | data | data MSB | reserved | reserved | reserved |
| Answer | Command specifier answer | reserved | reserved | reserved | reserved | reserved | reserved | reserved |

Command specifier request: 0x04, vendor-ID

0x41, product-code

0x42, revision-number

0x43, serial-number

Command specifier answer: 0x44

*Hint*

The revision-number can be ignored by using 0.

**Tab 15** Configure node ID

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | node ID | reserved | reserved | reserved | reserved | reserved | reserved |
| Answer | Command specifier answer | error code | spec. error | reserved | reserved | reserved | reserved | reserved |

command specifier request: 0x11
node ID: 1 … 127

command specifier answer: 0x11
error code: 0, protocol successfully completed
1, node ID out of range
spec. error: always 0

**Tab 16** Configure bit timing parameters

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | table selector | reserved | reserved | reserved | reserved | reserved | reserved |
| Answer | Command specifier answer | error code | spec. error | reserved | reserved | reserved | reserved | reserved |

command specifier request: 0x13
table selector: 0, standard CiA bit timing table
0x80, STW bit timing table
table index: *standard CiA bit timing table*
0, 1 Mbit/s
1, 800 kbit/s
2, 500 kbit/s
3, 250 kbit/s
4, 125 kbit/s
5, reserved
6, 50 kbit/s
7, 20 kbit/s
*STW bit timing table*
102, 400 kbit/s
100, 100 kbit/s
101, 40 kbit/s

command specifier answer: 0x13
error code: 0, protocol successfully completed
1, bit timing not supported
spec. error: always 0

**Tab 17** Activate bit timing parameters

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | switch_delay | reserved | reserved | reserved | reserved | reserved | reserved |

command specifier: 0x15
switch_delay: The duration of the two periods of time to wait.
See DSP305 for more details. Unit: milliseconds.

*Hints*

- Only the last saved bit timing will be activated by this service.
- After setting the new bit timing valid, the PBM4 with CANopen will transmit the bootup-message, but also it will stay in NMT stopped state.

**Tab 18** Store configuration

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | reserved | reserved | reserved | reserved | reserved | reserved | reserved |
| Answer | Command specifier answer | error code | spec. error | reserved | reserved | reserved | reserved | reserved |

command specifier request: 0x17
command specifier answer: 0x17
error code: 0, protocol successfully completed
1, node ID out of range
255, see spec. error
spec. error: only with error code 255: 1, nothing to store

**Tab 19** Inquire LSS address

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | reserved | reserved | reserved | reserved | reserved | reserved | reserved |
| Answer | Command specifier answer | data LSB | data | data | data MSB | reserved | reserved | reserved |

command specifier request: 0x5A, vendor-Id
0x5B, product-code
0x5C, revision-number
0x5D, serial-number

command specifier answer: like request
data: requested value

**Tab 20** Inquire node ID

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | reserved | reserved | reserved | reserved | reserved | reserved | reserved |
| Answer | Command specifier answer | node ID | reserved | reserved | reserved | reserved | reserved | reserved |

command specifier request: 0x5E
command specifier answer: 0x5E
node ID: node ID

*Hint*
The return value of the node ID will be the valid and stored value from EEPROM.

**Tab 21** LSS identify remote slave

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | data LSB | data | data | data MSB | reserved | reserved | reserved |
| Answer | Command specifier answer | reserved | reserved | reserved | reserved | reserved | reserved | reserved |

command specifier request: 0x46, vendor-Id
0x47, product-code
0x48, revision-number-low
0x49, revision-number-high
0x4A, serial-number-low
0x4B, serial-number-high
command specifier answer: 0x4F

*Hints*
- The revision-number-low and revision-number-high can be ignored by using 0.
- To identify the slave, the shown order of the requests must be observed.

**Tab 22** LSS identify non-configured remote slave

| | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | Command specifier request | reserved | reserved | reserved | reserved | reserved | reserved | reserved |
| Answer | Command specifier answer | reserved | reserved | reserved | reserved | reserved | reserved | reserved |

command specifier request: 0x4C
command specifier answer: 0x50

## 6.2 LSS example

This example of the usage of the Layer setting services shows the changing of the node ID from 1 to 5 and the changing of the bit rate to 250 kbit/s. Only one slave has to be connected to the CAN bus if using this example.

**Tab 23** LSS example

| No. | Service | CAN-ID | DLC | Dir | Comment | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| | NMT boot-up | 0x701 | 1 | Rx | Boot-up message from slave | | | | |
| | | 0x00 | | | | | | | |
| 1 | NMT stopped | 0x000 | 2 | Tx | Set slave to NMT stopped state | | | | |
| | | 0x02 | 0x00 | | | | | | |
| 2 | Switch state global: LSS configuration state | 0x7E5 | 8 | Tx | Set slave to LSS configuration state | | | | |
| | | 0x04 | 0x01 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| 3 | Configure node ID | 0x7E5 | 8 | Tx | Set new node ID: 5 | | | | |
| | | 0x11 | 0x05 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| | | 0x7E4 | 8 | Rx | Answer: ok | | | | |
| | | 0x11 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| 4 | Configure bit timing parameters | 0x7E5 | 8 | Tx | Set new bit rate: 250 kbit/s | | | | |
| | | 0x13 | 0x00 | 0x03 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| | | 0x7E4 | 8 | Rx | Answer: ok | | | | |
| | | 0x13 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| 5 | Store configuration | 0x7E5 | 8 | Tx | Store the new settings to EEPROM | | | | |
| | | 0x17 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| | | 0x7E4 | 8 | Rx | Answer: ok | | | | |
| | | 0x17 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| 6 | Activate bit timing parameters | 0x7E5 | 8 | Tx | Activate the new bit rate after 100 ms | | | | |
| | | 0x15 | 0x64 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| | | 0x701 | 1 | Rx | Boot-up message from slave with new bit rate (but with old node ID!) | | | | |
| | | 0x00 | | | | | | | |
| 7 | Switch state global: LSS waiting state | 0x7E5 | 8 | Tx | Set all slaves to LSS waiting state | | | | |
| | | 0x04 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |
| | | 0x705 | 1 | Rx | Boot-up message from slave (with new node ID) | | | | |
| | | 0x00 | | | | | | | |

Direction: Tx: message from (NMT/LSS) master
Rx: message from slave

*Hint*
Services no. 6 and 7 can be replaced by a power supply cycle.

# 7 CAN Communication without CANopen Functionality

## 7.1 Basic Configuration

The CAN pressure transmitter can be used without any problems in CAN networks without CANopen functionality. Before using the PBM4 with CANopen, the following basic configurations should be set:

1. Bit rate, default is 125 kbit/s, object 0x4F00
2. Node ID, default is 1, object 0x4F01.
3. The CAN identifier will be created from the node ID (see *Tab 42*, page *29*, chapter *7.5 Reserved CAN Identifiers*). All CAN identifiers are 11 bit identifiers (default setting).
4. Additional settings (phys. unit, etc.) can be found in the object dictionary (*5.4 Object Dictionary: Device Profile*, page *11*) and in *7.4 Change Node Configuration Manually*, page *27*.
5. The new settings are saved with object 0x1010/01. The 0x65766173 (ASCII: “save“) must be entered here. The settings will be saved to non-volatile memory.

## 7.2 Network Operation without CANopen Master

After connecting the transmitter to the supply voltage, the transmitter will send a boot-up message with the CAN identifier 0x700 + Node ID (default 0x701) with one data byte (content = 0) if no error is detected.
If an error is detected the error code (see *5.6 Emergency Message*, page *16*) will be sent together with the CAN identifier.
The pressure transmitter is now in the “Pre_Operational_State”. With the CANopen command “Start_Remote_Node” the pressure transmitter will be activated:

**Tab 24** Start_Remote_Node

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command “Start_Remote_Node | 0x000 | 2 | 0x01 | Node ID or 0x00 (all CANopen members) | | | | | | |

The “Start_Remote_Node” will be answered with a data message (PDO) with the CAN identifier 0x180 + Node ID (default 0x181). Now the CAN transmitter sends cyclically (default setting) PDOs with the pressure value and the status.

**Tab 25** Construction of the message for pressure measurement (PDO)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| PDO message | 0x180 + node ID | 5 | Pressure Signed32 LSB | Pressure Signed32 | Pressure Signed32 | Pressure Signed 32MSB | Pres-sure Status | | | |

The values of the pressure measurement or the temperature measurement can be also read as 32 bit integer or 32 bit float. The choice is done by the PDO mapping and is described in *5.5 Configuration of the transmit PDO*, page *14*.

The values of the pressure measurement or the temperature measurement can be also read from the object dictionary (SDO access) as 32 bit integer or 32 bit float. The floating point format is explained in the appendix (chapter *9.1 Definition of IEEE 32Bit (single precision) floating point numbers*, page *31*). This access is independent of the current operational state of the pressure transmitter. The status provides the following information:

Bit 0: pressure value invalid

Bit 1: positive overload

Bit 2: negative overload

*Pressure measurement*

**Tab 26** Request value of pressure measurement (float32, SDO access)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | 0x600 + node ID | 8 | SDO-request 0x40 | Indes LSB 0x30 | Index MSB 0x61 | Sub-in-dex 0x01 | Not used | | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x43 | Index LSB 0x30 | Index MSB 0x61 | Sub-in-dex 0x01 | Data LSB | Data | Data | Data MSB |

**Tab 27** Request value of pressure measurement (integer32, SDO access)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | 0x600 + node ID | 8 | SDO-request 0x40 | Indes LSB 0x30 | Index MSB 0x91 | Sub-in-dex 0x01 | Not used | | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x43 | Index LSB 0x30 | Index MSB 0x91 | Sub-in-dex 0x01 | Data LSB | Data | Data | Data MSB |

*Temperature measurement*

**Tab 28** Request value of temperature measurement (float32, SDO access)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | 0x600 + node ID | 8 | SDO-request 0x40 | Index LSB 0x30 | Index MSB 0x61 | Sub-index 0x02 | Not used | | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x43 | Index LSB 0x30 | Index MSB 0x61 | Sub-index 0x02 | Data LSB | Data | Data | Data MSB |

**Tab 29** Request value of temperature measurement (integer32, SDO access)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | Byte 7 |
| Command | 0x600 + node ID | 8 | SDO-request 0x40 | Index LSB 0x30 | Index MSB 0x91 | Sub-index 0x02 | Not used | | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x43 | Index LSB 0x30 | Index MSB 0x91 | Sub-index 0x02 | Data LSB | Data | Data | Data MSB |

*SDO abort codes*

If the SDO access fails, the PBM4 with CANopen will answer with a SDO abort code.

**Tab 30** SDO abort codes

| SDO abort code | Meaning |
|---|---|
| 0x06010001 | Attempt to read a write only object. |
| 0x06010002 | Attempt to write a read only object. |
| 0x06020000 | Object does not exist in the object dictionary. |
| 0x06040041 | Object cannot be mapped to the PDO. |
| 0x06040042 | The number and length of the objects to be mapped would exceed PDO length. |
| 0x06040043 | General parameter incompatibility reason. |
| 0x06060000 | Access failed due to an hardware error. |
| 0x06070012 | Data type does not match, length of service parameter too high |
| 0x06070013 | Data type does not match, length of service parameter too low |
| 0x06090011 | Sub-index does not exist. |
| 0x06090030 | Value range of parameter exceeded (only for write access). |
| 0x06090031 | Value of parameter written too high. |
| 0x06090032 | Value of parameter written too low. |

## 7.3 Cyclically Sending

The pressure transmitter PBM4 CANopen is able to send the values of measurements (PDO) cyclic with a programmable time interval.
The event timer is activated by writing 0xFF to the object 0x1800 sub-index 2 (transmission type).
The timer interval is written to the object 0x1800 sub-index 5 (event timer). The value (unsigned16) is set in units of 1 ms. The value range is from 0 ms to 65535 ms. 0 stops the event timer.
Default settings:

- Transmission type: 0xFF (event timer active)
- Event timer: 1000 ms

**Tab 31** Activate event timer (SDO access)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO write 0x2F | Index LSB 0x00 | Index MSB 0x18 | Sub-index 0x02 | Transmission Type 0xFF | Not used | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x00 | Index MSB 0x18 | Sub-index 0x02 | Not used | | | |

**Tab 32** Set event timer (SDO access)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO write 0x2B | Index LSB 0x00 | Index MSB 0x18 | Sub-index 0x05 | Timer LSB | Timer MSB | Not used | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x00 | Index MSB 0x18 | Sub-index 0x05 | Not used | | | |

**Tab 33** Get event timer (SDO access)

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO request 0x40 | Index LSB 0x00 | Index MSB 0x18 | Sub-index 0x05 | | | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x4B | Index LSB 0x00 | Index MSB 0x18 | Sub-index 0x05 | Timer LSB | Timer MSB | Not used | |

Note:
If the device is not configured as self-starting device the message “start_remote_node” must be sent each time after reset or power up.

The pressure transmitter can sample up to 1000 values of measurement per second. The maximum data rate on the CAN bus depends on the bitrate and the maximum workload.

## 7.4 Change Node Configuration Manually

The basic configuration of the pressure transmitter can be manually set through the object dictionary with the addresses 0x4F01 (node ID) and 0x4F00 (CAN bitrate). The new settings are active after a reset.

*Note ID*

**Tab 34** Set node ID

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO write 0x2F | Index LSB 0x01 | Index MSB 0x4F | Sub-index 0x00 | Node ID Byte | Not used | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x01 | Index MSB 0x4F | Sub-index 0x00 | Not used | | | |

**Tab 35** Get node ID

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO write 0x40 | Index LSB 0x01 | Index MSB 0x4F | Sub-index 0x00 | Not used | | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x4F | Index LSB 0x01 | Index MSB 0x4F | Sub-index 0x00 | Node ID Byte | Not used | | |

*CAN bitrate*

**Tab 36** Set CAN bitrate index

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO write 0x2F | Index LSB 0x00 | Index MSB 0x4F | Sub-index 0x00 | Bitrate index byte | Not used | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x00 | Index MSB 0x4F | Sub-index 0x00 | Not used | | | |

**Tab 37** Get CAN bitrate index

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO request 0x40 | Index LSB 0x00 | Index MSB 0x4F | Sub-index 0x00 | Not used | | | |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x4F | Index LSB 0x00 | Index MSB 0x4F | Sub-index 0x00 | Bitrate index byte | Not used | | |

*NMT start-up*

**Tab 38** Activate automatic transition to the “Operational_State”

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO write 0x23 | Index LSB 0x80 | Index MSB 0x1F | Sub-index 0x00 | Data LSB 0x08 | Data 0x00 | Data 0x00 | Data MSB 0x00 |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x80 | Index MSB 0x1F | Sub-index 0x00 | Not used | | | |

**Tab 39** Deactivate automatic transition to the “Operational_State”

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO write 0x23 | Index LSB 0x80 | Index MSB 0x1F | Sub-index 0x00 | Data LSB 0x04 | Data 0x00 | Data 0x00 | Data MSB 0x00 |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x80 | Index MSB 0x1F | Sub-index 0x00 | Not used | | | |

**Tab 40** “save”-command to store all parameters to non-volatile memory

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO request 0x23 | Index Lo 0x10 | Index Hi 0x10 | Sub-index 0x01 | ASCII ‘s’ 0x73 | ASCII ’a’ 0x61 | ASCII ‘v’ 0x76 | ASCII ‘e’ 0x65 |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x10 | Index MSB 0x10 | Sub-index 0x01 | Not used | | | |

**Tab 41** "load"-command to restore all default parameters

| | ID | DLC | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Byte 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Command | 0x600 + node ID | 8 | SDO request 0x23 | Index Lo 0x11 | Index Hi 0x10 | Sub-index 0x01 | ASCII 'l' 0x6C | ASCII 'o' 0x6F | ASCII 'a' 0x61 | ASCII 'd' 0x64 |
| Answer | 0x580 + node ID | 8 | SDO Ack. 0x60 | Index LSB 0x11 | Index MSB 0x10 | Sub-index 0x01 | Not used | | | |

## 7.5 Reserved CAN Identifiers

The following CAN identifiers are reserved by the CAN protocol:

**Tab 42** Reserved CAN identifiers

| CAN-Identifier (11Bit), Hex | Description |
|---|---|
| 0x000 | NMT, network management |
| 0x080 | SYNC, synchronisation message, not used in asynchronous mode (see *5.2 Object Dictionary: Communication Profile*, page 7) |
| 0x080 + Node ID<br>max. range 0x081 … 0x0FF | Emergency message |
| 0x180 + Node ID<br>max. range 0x181 … 0x1FF | PDO1 TX, message with the value of pressure measurement |
| 0x580 + Node ID<br>max. range 0x581 … 0x5FF | SDO TX, CANopen configuration message |
| 0x600 + Node ID<br>max. range 0x601 … 0x67F | SDO RX, CANopen configuration message |
| 0x700 + Node ID<br>max. range 0x701 … 0x77F | CANopen node guarding |

# 8 Extensions

- Device profile DS404
- Heartbeat function
- Different units for the pressure and temperature values available
- Programmable monitoring of the measurement range
- Autozero function
- Offset shift

# 9 Appendix

## 9.1 Definition of IEEE 32Bit (single precision) floating point numbers

According IEEE-754 standard.
Single precision floating point numbers cover a value range of -3.4 * $10^{38}$ … 3.4*$10^{38}$.
32 bit floating point numbers need 4 byte (32 bit) storage memory. The following table shows the IEEE 32 bit implementation of floating point numbers:

**Tab 43** Floating point numbers

| Bit position | 31 | 30 … 23 | 22 … 0 |
|---|---|---|---|
| Function | S | Exponent | Mantissa |

S = sign bit
The value can be calculated with this formula:

$$(-1)^S \cdot 2^{(exponent-127)} \cdot (1 + mantissa)$$

The mantissa starts behind the comma (position 2 – 1). The first number in front of the comma (position 20) is always 1 and will not be stored in the mantissa.

*Example*

Hex: 400C CCEA$_{HEX}$
Binary: 0100 0000 0000 1100 1100 1100 1110 1010$_{BIN}$
Sign bit = 0
Exponent = 10000000$_{BIN}$ = 128$_{DEC}$
Mantissa = 0001100110 0110011101 010$_{BIN}$
= 0*$2^{-1}$ + 0*$2^{-2}$ + 0* $2^{-3}$ + 1*$2^{-4}$ + 1*$2^{-5}$ + … + 1 * $2^{-22}$
= 0.100003481$_{DEC}$

$$400CCCEA_{HEX} = (-1)^0 \cdot 2^{(128-127)} \cdot (1 + 0.100003481) = 2.200007_{DEC}$$

More examples and IEEE-754 definitions:

**Tab 44** Examples and IEEE-754 definitions

| HEX | DEC |
|---|---|
| 00000000$_{HEX}$ | 0.0 |
| 3F800000$_{HEX}$ | 1.0 |
| BF800000$_{HEX}$ | -1.0 |
| FFFFFFFF$_{HEX}$ | Not a Number (NaN) |

## 9.2 CoEdit

The CoEdit is a program for reading and writing the objects of the PBM4 with CANopen. The objects are defined in the EDS file for the pressure transmitter.

## 9.3 References

| | |
|---|---|
| DS301 | Application Layer and Communication Profile |
| DS302-2 | Additional Application Layer Functions<br>Part 2: Network Management |
| DR303-1 | Cabling and Connector Pin Assignment |
| DS404 | Device Profile Measuring Devices and Closed-Loop Controllers |

## 9.4 Definitions

| | |
|---|---|
| COB | Communication Object<br>Data must be sent inside a COB across a CAN network. There exist 2048 different COBs in a CAN network. A COB contains maximal 8 da-ta bytes. |
| LSS | Layer setting services |
| PDO | Process Data Object |
| SDO | Service Data Object |

## 9.5 Declaration of Conformity

Baumer

Passion for Sensors

# EU-Konformitätserklärung
# EU Declaration of Conformity
# Déclaration UE de Conformité

Wir erklären in alleiniger Verantwortung, dass die Produkte, auf die sich diese Erklärung bezieht, die grundlegenden Anforderungen der angegebenen Richtlinie(n) erfüllen und basierend auf den aufgeführten Norm(en) bewertet wurden.

We declare under our sole responsibility that the products to which the present declaration relates comply with the essential requirements of the given directive(s) and have been evaluated on the basis of the listed standard(s).

Nous déclarons sous notre seule responsabilité que les produits auxquels se réfère la présente déclaration sont conformes aux exigences essentielles de la directive/ des directives mentionnée(s) et ont été évalués sur la base de la norme/ des normes listée(s).

**Hersteller**
Manufacturer
Fabricant
Baumer Electric AG

**Bezeichnung**
Description
Description
Elektronische Druckmesstechnik
Electronic pressure measurement
Electronique mesure de pression

**Typ(en) /** Type(s) /Type(s)

PBM4-13.xxxR.[#]xx.xx6x
PBM4-13.xxxR.[@]xx.xx6x/xxxx

[#]: A1...A9

[@]: C1…C9

x = beliebige Zahl oder Buchstabe / any figure or letter / n'importe quel nombre ou lettre

PELV Anforderungen gemäss EN 60204-1: 0VDC (GND) soll an einem Punkt im System mit Schutzerde verbunden werden.
PELV requirements according to EN 60204-1: 0 VDC (GND) is to be connected at one point in the system with protective earth.
Exigences TBTP selon la norme EN 60204-1: 0 VDC (GND) doit être connecté à un point dans le système avec le conducteur de protection.

**Richtlinie(n)**
Directive(s)
Directive(s)
2014/30/EU, 2011/65/EU

**Norm(en)**
Standard(s)
Norme(s)
EN 61000-6-2:2005, EN 61000-6-3:2007+A1:2011, EN 50581:2012

**Ort und Datum**
Place and date
Lieu et date
Frauenfeld, 15.04.2016

**Unterschrift/Name/Funktion**
Signature/name/function
Signature/nom/fonction
Christoph Kluser
Head of Product Segment
Sensor Solutions

Baumer_PBM4_DE-EN-FR_CoC_81152967.docx/su 1/1

**Baumer Electric AG**
P.O. Box
Hummelstrasse 17
CH-8501 Frauenfeld
MwSt-Nr. 192 892

Phone +41 (0)52 728 1122
Fax +41 (0)52 728 1144
sales.ch@baumer.com
www.baumer.com

UBS AG • Clearing 239 • SWIFT/BIC UBSWCHZH80A
(CHF) Konto 239-429 007 02R • IBAN CH64 0023 9239 4290 0702R
(EUR) Konto 239-429 007 62P • IBAN CH11 0023 9239 4290 0762P
(USD) Konto 239-429 007 61G • IBAN CH44 0023 9239 4290 0761G