Baumer Passion for Sensors

Operating instructions CANopen Pressure Sensor

PDxJ xxxx.xxx.xxxx
CANopen DS 404

Baumer Electric AG
Hummelstrasse 17
Postfach
CH-8501 Frauenfeld
www.baumer.com

We reserve the right to make changes to the technology and design.

Ver. 3.11

1 General

This manual contains important information for the safe and compliant use of the CANopen pressure sensor and must be read before initial start-up.
It was created for personnel trained and qualified in handling electrical equipment.
There are also a short introduction with definitions of CANopen terms and useful notes for properly operating the pressure sensor.

1.1 Safety instructions

- The pressure sensor is a compact, extremely sensitive precision measuring instrument. It is used exclusively to measure pressure and temperature, to process and to supply measured values as CANopen signals for the downstream device. The pressure sensor must only be used for this purpose.
- Correct and safe operation requires proper transport, storage, mounting and careful operation and maintenance.
- Only a specialist may install and mount the pressure sensor.
- Check all electrical connections before using the system for the first time.
- When using the sensors, obey all applicable safety and accident prevention regulations.
- Safety measures must be put in place, both in terms of hardware and software, so that a broken line does not result in undefined states of the automation equipment.
- In the case of systems where a malfunction may cause great damage to property or even to personnel, safety measures must be put in place that ensure a safe operating state in the event of a malfunction.
- You must not operate the pressure sensor outside the specifications (see the data sheet).
- Do not make any mechanical or electrical changes to the sensor.
- Despite the rugged housing, the pressure sensor must not be subjected to any hard impacts.
- Avoid static and dynamic overpressure exceeding 200% of the nominal range.

1.2 Mounting and initial start-up

- For information on mounting and connection to the measuring system, refer to the mounting instructions supplied with the sensor.
- Only perform wiring tasks when no power is applied.
- Do not attach or remove electrical connections that are under power.
- Install the entire system to maximize EMC. The installation environment and the cabling affect the EMC of the pressure sensor. Install the device and the power line separated from one another and at a great distance from lines with high noise levels.
- Connect the pressure sensor to protective ground and use shielded cables. Bond the cable braid to the cable screw fitting.

2 Project engineering

2.1 Maximum system extent

To construct an operational bus, there must be at least one master (or parent system) on the bus. This master may be a PLC controller or a PC with an appropriate CAN board. Every CANopen pressure sensor represents one active CAN node.
One bus string with one master of the CAN network can have a maximum of 127 users. Every user has its own address.

You can find the factory defaults of this sensor in Chapter 6.2 Parameter Handling (save, load default).

You must absolutely comply with the permissible bus and stub line lengths given in Table 1.

The maximum permitted total line length and total stub length
- is dependent on the baud rate and
- can be divided into several segments or individual stubs.

Table 1

Baud rate [Kbit/s]	10	20	50	100	125	250	500	800	1000
Total bus length	5,000 m	3,000 m	1,000 m	500 m	400 m	200 m	75 m	30 m	25 m
Total stub length	1,360 m	875 m	350 m	175 m	140 m	70 m	35 m	20 m	17 m
Individual stub length	270 m	175 m	70 m	35 m	28 m	14 m	7 m	4 m	3 m

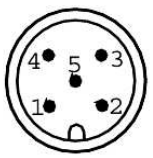
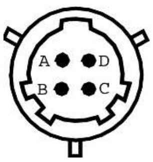
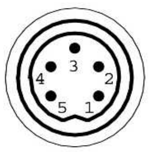
Maximum total bus length (with 120 ohm termination resistor) and maximum stub length (without termination resistor) as a function of the baud rate

3 Connections

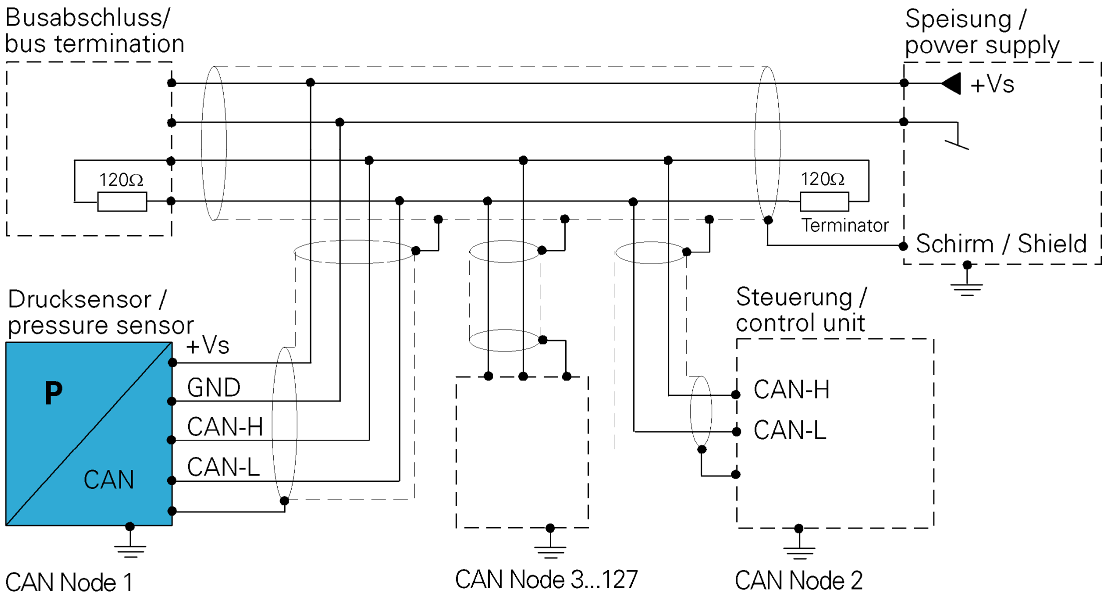
3.1 Electrical connection

Connect the pressure sensor according to the schematic below. Make sure the polarity is correct.
Use shielded cables.
The sensor housing and the cable shield must be grounded. Avoid differences in electrical potential between parts of the system and the measuring chains.

3.1.1 Pin assignment

	**Connector 14C** Pin 1 n.c. Pin 2 +Vs Pin 3 GND Pin 4 CANH Pin 5 CANL housing shield
	**Connector S40** Pin A CANH Pin B +Vs Pin C CANL Pin D GND housing shield
	**Connector S50** Pin 1 n.c. Pin 2 +Vs Pin 3 GND Pin 4 CANH Pin 5 CANL housing shield

3.1.2 Connection diagram



To comply with the PELV requirements according to EN 60204-1 Section 6.4.1, we recommend connecting 0V (GND) to the protective ground at one point in the system.

3.2 Electrical potential conditions

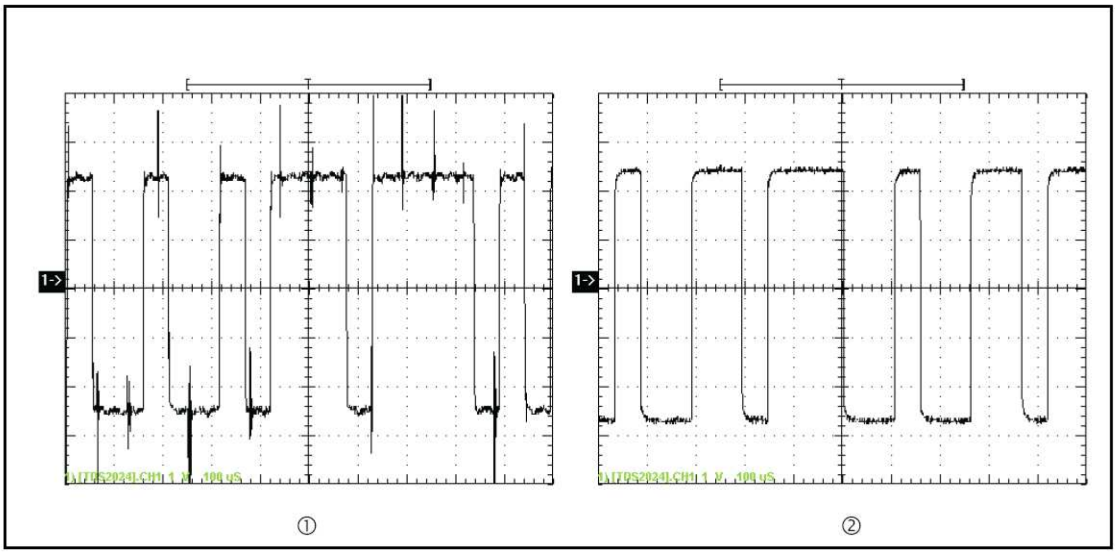
The electrical potential conditions of a CAN bus system with a CANopen pressure sensor are characterized by the following features:

- The CAN bus connection is not potentially isolated from the supply connection
- The individual CANopen pressure sensors are not electrically isolated from the supply voltage
- Every CANopen pressure sensor can be powered separately

Avoid differences in electrical potential by

- connecting every CAN user to the same ground reference potential (PE) of the machine/system via the shortest path with the lowest resistance possible.
- using a potential equalization line between the communications users.
- connecting the ground reference for the machine/system to the main ground via a low resistance path.

Recognizing EMC noise in the signal oscilloscope display



Oscilloscope displays of the CAN signals (1) with and (2) without noise voltage (measuring points: CAN_HIGH to CAN_LOW).

To quantify the noise, measurements with a CAN analyzer are necessary. With this device, important bus parameters such as the bus load or the number of error frames can be determined and more in-depth analyses performed.

3.3 EMC-compatible wiring

EMC (electromagnetic compatibility) is the ability of a device to operate without errors in a specified electromagnetic environment without affecting the environment in an impermissible way.

All CANopen pressure sensors meet these requirements because all sensors have been tested for compliance with the legally prescribed limits (industrial standard).

3.3.1 Grounding inactive metal parts

All inactive metal parts must be bonded over a wide area and via a low impedance path (grounding). This action ensures that there is a uniform reference potential for all elements of the system.

The CANopen pressure sensors are grounded by way of the pressure connector.

3.3.2 Shielding lines

The shield should be grounded, if possible, at both ends using an EMC-compatible shield connection.

3.4 Specification of the CAN lines

The cable that you use to connect the bus users to the CAN bus must comply with the ISO 11898 standard. Consequently, the lines must possess the following electrical characteristics:

Specification of the CAN lines

Bus system total length	< 300 m	< 1,000 m
Cable type	LIYCY 2 x 2 x 0.5 mm² (shielded twisted pair)	CYPIMF 2 x 2 x 0.5 mm² (shielded twisted pair)
Line resistance	≤ 40 Ohm/km	≤ 40 Ohm/km
Capacitance	≤ 130 nF/km	≤ 60 nF/km
Connection	Pair 1 (white/brown): CAN-GND and +Vs Pair 2 (green/yellow): CAN-HIGH and CAN-LOW	

- Only use lines that have an additional pair of conductors for CAN-GND.
- Noise-free bus operation is only possible with a correctly connected CAN-GND.

Connect the bus termination resistors
A 120 ohm termination resistor must be connected at the physical beginning and at the physical end of the bus system.

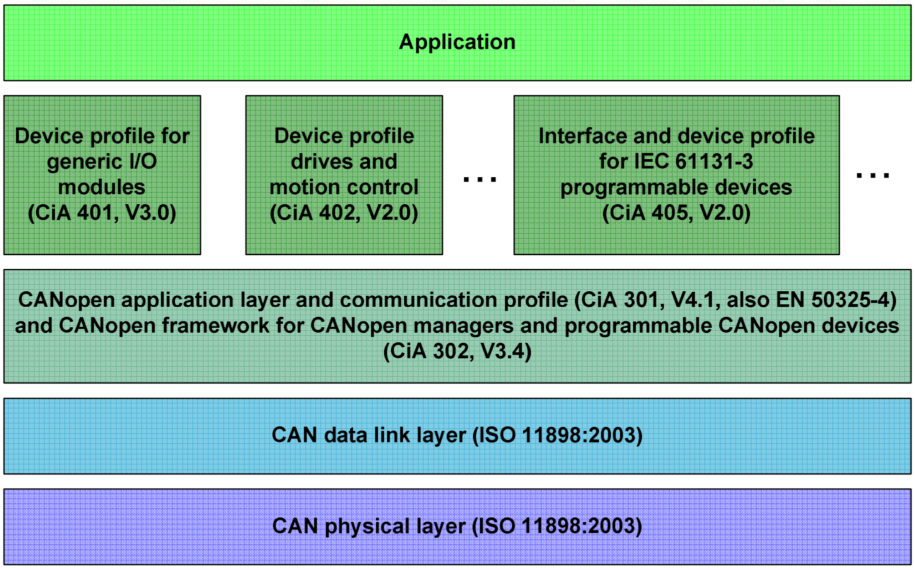
4 CANopen

4.1 Introduction

The CANopen protocol is an open, standardized ISO/OSI Layer 7 protocol based on the Controller Area Network (CAN) application layer. CANopen has been developed, internationally standardized and is maintained by the CAN in Automation (CiA) user organization.
CANopen has the following performance characteristics:

- Transmission of time-critical process data using the producer-consumer principle. Messages may be received by all bus users. They are not given the destination address but rather they have an identifier.
- Standardized device description (data, parameters, functions, programs) in the "Object directory." Access to all objects of a device using the standardized transmission protocol according to the client-server principle.
- Standardized node monitoring (node guarding and heartbeat), fault signaling (emergency messages) and network coordination (network management).
- Standardized system for synchronous operation (synchronization message).
- Standardized function for configuration of the baud rate and the device ID over the bus using LSS.

CANopen consists of a communication profile (controlling communications) and various device profiles for the typical application profiles.



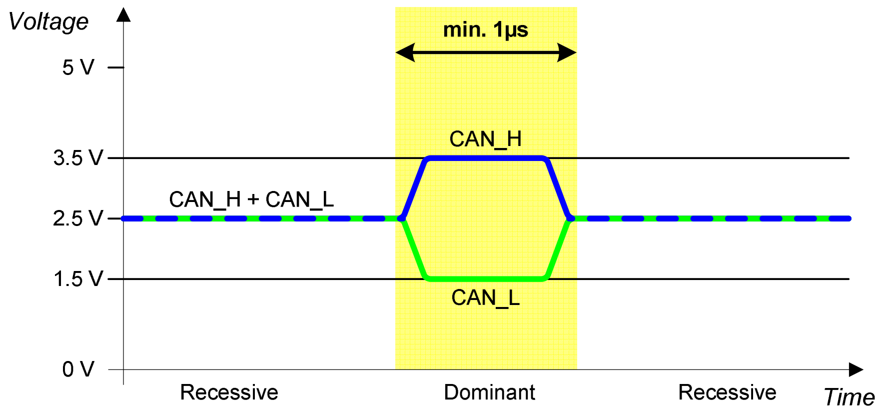
The CANopen communication profile (CiA DS-301) controls the "How" of communications. In this respect, differentiation is made between real-time data and parameter data.
Baumer Process Instrumentation sensors use the DS404 device profile for measurement and control equipment.

4.2 Signals, structure and bus topology

4.2.1 Bus signals

Good electrical noise immunity is achieved, among other measures, in that one bit is transmitted differentially on two lines. The CAN-High and CAN-Low lines contain the inverted and the non-inverted serial data signal.

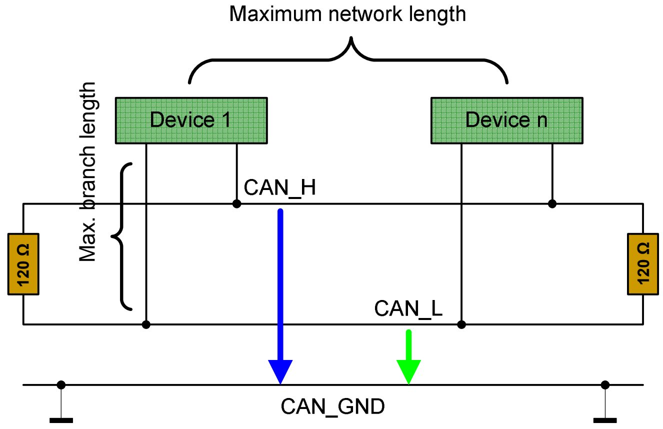
The state having two different levels on CAN-H and CAN-L is known as the dominant state. The state having two equal levels is known as the recessive state.



According to the CAN definition, the dominant state corresponds to a logical zero (bus drivers have an open collector output).
If a node puts a logical zero on the bus, it overwrites the state of a logical one from a different node.

4.2.2 Network topology

The CAN architecture used as a basis defines the physical structures of the CANopen network. This is based on a bus (line) topology. To avoid signal reflections, the ends of the network must be terminated using a termination resistor (120 ohm).
In addition, pay attention to the maximum stub length for connecting the individual network nodes.



The permissible bit rates/line lengths for a CANopen network (CiA 301):

Data rate bus length	Sample point (TQ)	Max. stub length	Accumu-lated stub length
1 Mbit/s 25 m	87,5% (125 ns)	1,5 m	7,5 m
800 kbit/s 50 m	87,5% (125 ns)	2,5 m	12,5 m
500 kbit/s 100 m	87,5% (125 ns)	5,5 m	27,5 m
250 kbit/s 250 m	87,5% (250 ns)	11 m	55 m
125 kbit/s 500 m	87,5% (500 ns)	22 m	110 m
50 kbit/s 1000 m	87,5% (1,25 µs)	55 m	275 m
20 kbit/s 2500 m	87,5% (3,125 µs)	137,5 m	687,5 m
10 kbit/s 5000 m	87,5% (6,25 µs)	275 m	1375 m

Two conditions must exist for a CANopen network to operate without errors:

- All nodes must have the same bit rate
- Each node ID must be unique

The system integrator is responsible for maintaining the same bit rate and the different node IDs.
Baumer sensors come as standard with 125 kBaud and ID = 1. They can be configured using the 2100H and 2101H objects or with the LSS Service (CiA 305).

4.2.3 CAN message structure

A CAN message, also known as a frame, consists of the following seven fields:

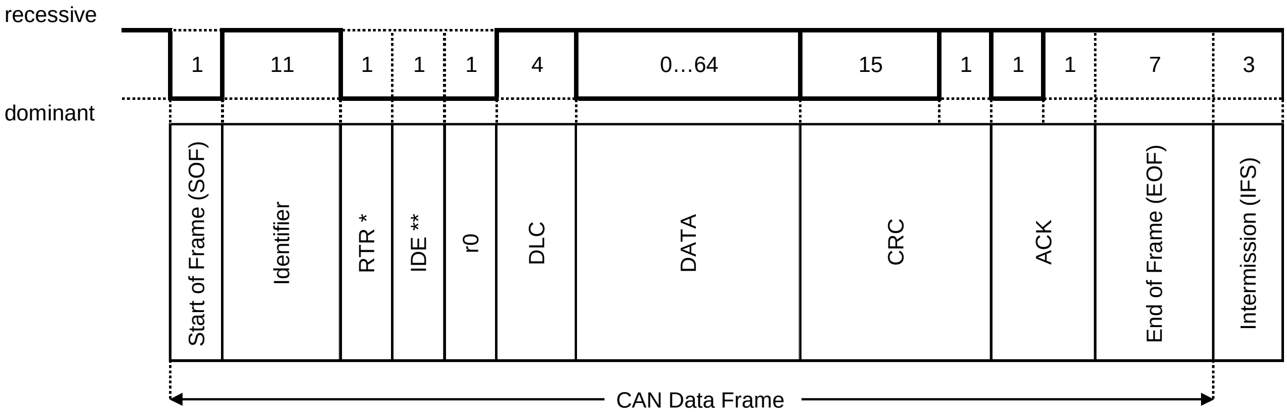
- Start of frame
- Message identifier
- Control bits
- Data (0-8 bytes)
- CRC check bits
- Acknowledge bit
- End of frame

The length of the identifier differentiates the frames:

- Standard Frame (11-bit identifier)
- Extended Frame (29-bit identifier)

Baumer Process Instrumentation only supports Standard Frames.

The figure below shows the structure of a Standard Frame according to the CAN 2.0A standard.



* RTR = 0 => Data Frame
RTR = 1 => Remote Frame

** IDE = 0 => 11Bit Identifier
IDE = 1 => 29 Bit Identifier

- Start of Frame: dominant (logical 0), used for synchronization
- Identifier: information for the receiver and priority information for bus arbitration
- RTR: recessive, differentiates between the data frame (dominant) and the data request frame (recessive)
- IDE: Identifier Extension
- r0: reserved
- DLC: contains length information for the following data
- DATA: contains the data of the frame
- CRC: marks the error code for the preceding data. The CRC checksum is used for detecting errors.
- ACK: contains an acknowledgment from other receivers upon correct reception of the message
- EOF: marks the End of Frame (7 recessive bits)
- IFS: marks the intermission frame space, the time for transmitting a correctly received frame.

4.2.4 Bitwise bus arbitration

During the arbitration phase, it is determined which of the messages undergoing simultaneous arbitration has the highest priority. The message having the lowest value for the message identifier has the highest priority. The arbitration phase comprises the transmission of the message identifier and the RTR bit (Remote Transmission Request bit). If a network node detects a dominant bus level (logical 0) although it connected a recessive level (recessive bit) itself, it stops transmission immediately and transitions to the receiver state because, in this case, a message with a higher priority was obviously transmitted at the same time.

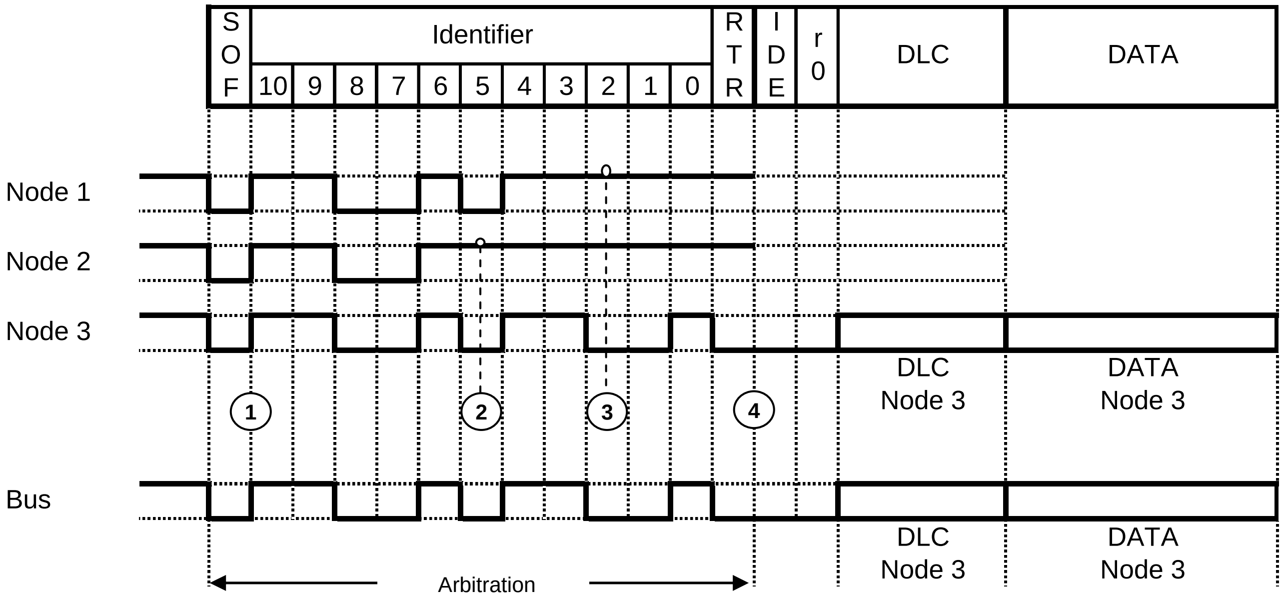


Fig. 1 Principle of bitwise bus arbitration -- Nodes 1, 2 and 3 simultaneously start an arbitration process. At Time 2, Node 2 determines that the bus does not have the recessive level it sent and terminates its arbitration process. At Time 3, Node 1 gives up. At Time 4 (end of the arbitration process), Node 3 transmits its data.

4.2.5 Priority-oriented message transmission

The arbitration process described above guarantees at any time that the message with the highest priority is sent in each case as soon as the bus is free. The priority of the message is specified using the value in the Message identifier. The smaller this value, the higher the message priority. The principle of priority-oriented messages allows a very efficient utilization of the bandwidth available for data transmission. In this way, it is possible to fill the bus 100% with low-priority messages without noticeably delaying the transmission of messages having higher priority. A maximum latency of about 130 µs results for the message having the highest priority at a transmission rate of 1 Mbit/s.

4.2.6 Identifier distribution

As standard, Message identifiers of 11 bits in length are used in communications via CANopen. Thus, the range of 0 to $7FF_H$ is available.
The identifier distribution is designed so that, in one CANopen network, a maximum of 128 devices are present: one NMT master and up to 127 NMT slaves.

Default identifier assignment:

Communication objects	COB-ID(s) hex	Slave-Nodes
NMT node control	000	only receive
Sync	080	only receive
Emergency	080 + NodeID	transmit
TimeStamp	100	only receive
PDO	180 + NodeID 200 + NodeID 280 + NodeID 300 + NodeID 380 + NodeID 400 + NodeID 480 + NodeID 500 + NodeID	1. PDO transmit 1. PDO receive 2. PDO transmit 2. PDO receive 3. PDO transmit 3. PDO receive 4. PDO transmit 4. PDO receive
SDO	580 + NodeID 600 + NodeID	transmit receive
NMT node monitoring (node guarding/heartbeat)	700 + NodeID	transmit
LSS	7E4 7E5	transmit receive

The master in the network is capable of changing the mode of the slaves. Consequently, it controls the CANopen network. For this reason, the master is often also referred to as the CANopen Network Manager. Typically, a CANopen master is implemented using a PLC or a PC. The CANopen slaves can be assigned the addresses from 1 to 127. The device address automatically indicates a number of identifiers that are assigned to this device.

4.3 Objects

The object directory describes the complete functionality of the CANopen devices and is organized in tabular form. The object directory contains not only the standardized data types and objects of the CANopen communication profile and the device profiles but also vendor-specific objects and data types if provided. The entries are addressed using a 16-bit index (row address of the table, a maximum of 65,536 entries) and an 8-bit sub-index (column address of the table, a maximum of 256 entries). This makes it easy to group associated objects. The structure of this CANopen object directory is shown in the following table.

Overview of the entire object directory:

Indexrange	Description
0000h	Reserved
0001h to 025Fh	Data types
0260h to 0FFFh	Reserved
1000h to 1FFFh	Communications profile area
2000h to 5FFFh	Manufacturer specific profile area
6000h to 9FFFh	Standardized profile area
A000h to AFFFh	Network variable
B000h to BFFFh	System variabel
C000h to FFFFh	Reserved

Excerpt of the object region for communication (1000H … 1FFFH)

Indexrange	Description
1000h bis 1029h	general communication objects
1200h to 12FFh	SDO Parameter objects
1300h to 13FFh	CANopen Savety objects
1400h to 1BFFh	PDO Parameter objects
1F00h to 1F11h	SDO Manager objects
1F20h to 1F27h	Configuration Manager objects
1F50h to 1F54h	Program control objects
1F80h to 1F89h	NMT Master objects

4.4 Communication mechanisms

Differentiation is mainly made between two different types of data transmission. The Process Data Objects (PDOs) are used to transmit real-time data or process data and the Service Data Objects (SDOs) allow access to the object directory containing all device settings.
In addition to the standard transmission mechanisms, there are still more communications mechanisms. These are Network Management (NMT), Emergency (EMGY), Node Guarding and Heartbeat.

4.4.1 Process Data Objects (PDOs)

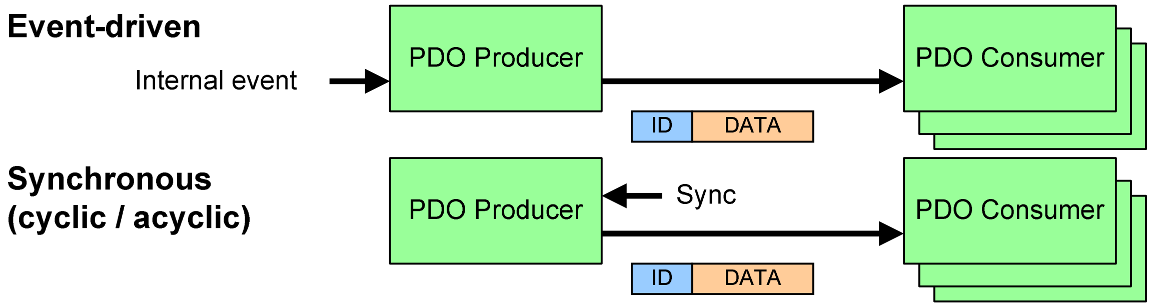
The main task of a CANopen system is exchanging process data.
For the transmission of process data, the protocol overhead is omitted and transmission uses the Producer-Consumer principle. This means that a message sent by a node (the Producer) can be received by all other nodes (the Consumers). This principle is also known as broadcast and represents a very efficient principle of data transmission.
PDO messages are not acknowledged to reduce the bus load as much as possible, primarily during time-critical applications. Consequently, this service is not a query-response mechanism.

The transmission of PDOs is possible only in the Operational state and the transmission packets do not have a fixed data length. The data length of a PDO can range from one to eight bytes.
With regard to the composition of the data packets, both the sender and the receiver must know how to compose or interpret, respectively, the contents. The sender of the PDO can be identified only by the COB ID.
PDO mapping describes the individual process variables transmitted in the data field of a PDO, how they are arranged as well as the data type and length used. The contents and the significance of the transmitted data in a PDO are defined in a PDO mapping list both on the send and the receive ends.

The transmission of process data can be triggered by various events:

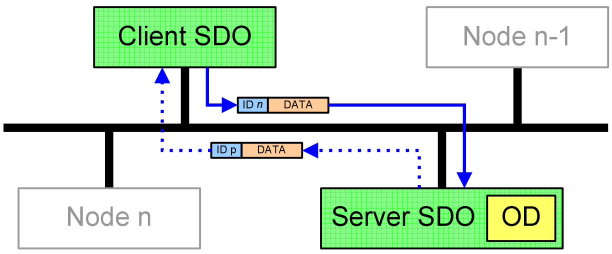
- Event driven
  The transmission of the PDOs is triggered by an internal event of the node. This can occur due to a timer in the device, by exceeding or dropping below a limit or through other internal events.
- Synchronized
  A bus user (usually the master) transmits synchronization messages on the bus. In the case of synchronous transmission, the PDOs are triggered by the received sync message. In this way, it is possible to obtain an instantaneous snapshot (process values at the same time) of the system.
- Request driven
  In this case, a bus user requests processed data using a Remote Transmission Request (RTR). This mechanism is deprecated and not implemented by the pressure sensors.

PDO message structure:

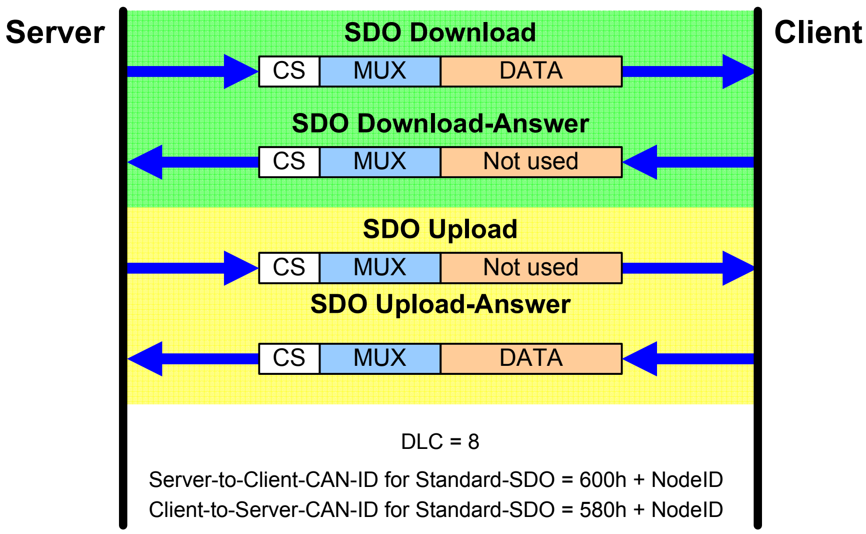


4.4.2 Service Data Objects (SDOs)

Specific communication objects, Service Data Objects (SDOs), are used for direct access to CANopen devices. Entries in the object directory can be read and written using these SDOs. Communication always takes place as a logical 1:1 connection (peer-to-peer) between two nodes (usually, the master is the configuring node and a normal bus user is the node to be configured).
As a result of the direct connection, a response is expected for every request. This can be compared to a connection via radio. Every request must receive a response even if the device is incapable of executing or responding to the request or even if the request itself contains errors. Such a negative response is known as an abort message. In addition to the 4-byte error code (cause of the abort), the abort message contains the object address which was to be accessed.



SDO message structure

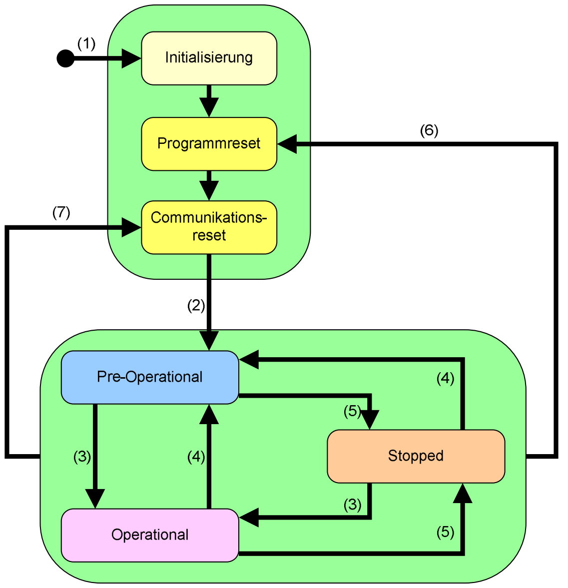


CS = command specifier
MUX = 16-bit Index and 8-Bit Subindex

4.4.3 Network Management (NMT)

In a CANopen network, there is one NMT master and between 1 and 127 NMT slaves.
The NMT master has complete control over all devices and may change the state of these devices.
The NMT messages have the highest priority in a CANopen network and have ID = 0. An NMT command has only two data bytes. The NMT master can control the state of a single slave (e.g., ID = 2) or the entire network (ID = 0).
The states in a CANopen network are usually shown using a state diagram. The following states are possible in a CANopen network:

- Initialization
- Pre-Operational
- Operational
- Stopped



(1) Power on
(2) Automatical alternation to Pre-Operational
(3) Alternation to Operational
(4) Alternation to Pre-Operational
(5) Alternation to Stopped
(6) Reset Application
(7) Communcationsreset

Initialization
The node is in the Initialization state following an NMT reset or a power-on. The device application and communication are initialized in this state. After completing initialization, the node transmits a Boot-up message and switches automatically to the Pre-Operational state.

Pre-Operational
In this state, it is possible to communicate with the node via SDOs. No PDO messages can be sent. This state is primarily used for configuring the CANopen devices.

Operational
In this state, the node is completely ready for operation and can transmit messages on its own.

Stopped
With the exception of Node guarding and Heartbeat messages, the node can send no other messages in this state. Only LSS configuration functions in this state.

Start Remote Node => Transition to Operational Mode

ID	DLC:	Byte1	Byte2
0	2	01h	Node

Node = module address, 0 = all nodes

Stop Remote Node => Transition to Stopped Mode

ID	DLC:	Byte1	Byte2
0	2	02h	Node

Node = module address, 0 = all nodes

Pre-Operational Remote Node => Transition to Pre-Operational Mode

ID	DLC:	Byte1	Byte2
0	2	80h	Node

Node = module address, 0 = all nodes

Reset Node => Software reset of the node

ID	DLC:	Byte1	Byte2
0	2	81h	Node

Node = module address, 0 = all nodes

4.4.4 Emergency (EMGY)

Emergency messages signal errors in a node. The Emergency message contains a code that uniquely identifies the error (defined in DS-301 and in the device profiles).
The Emergency messages are transmitted by the CANopen devices automatically.

Composition of the Emergency message:

Error-code	Error-register	Manufacturer specific error field

Overview of the error codes:

Error code (hex)	Error description
00xx	Errorreset / no error
10xx	General error
2xxx	Current
3xxx	Voltage
4xxx	Temperature
50xx	Device hardware
6xxx	Device software
70xx	Additional modules
8xxx	Monitoring
90xx	External error
F0xx	Additional functions
FFxx	Device specific

Overview of the Error register:

Bit	Cause of error
0	General error
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Device specific
6	Reserved (always 0)
7	Manufacturer specific

At the same time, the Error codes are also written into the Emergency history (object: 1003h).
The COB ID of the Emergency message is contained in object 1014h.

4.4.5 Node guarding and Heartbeat

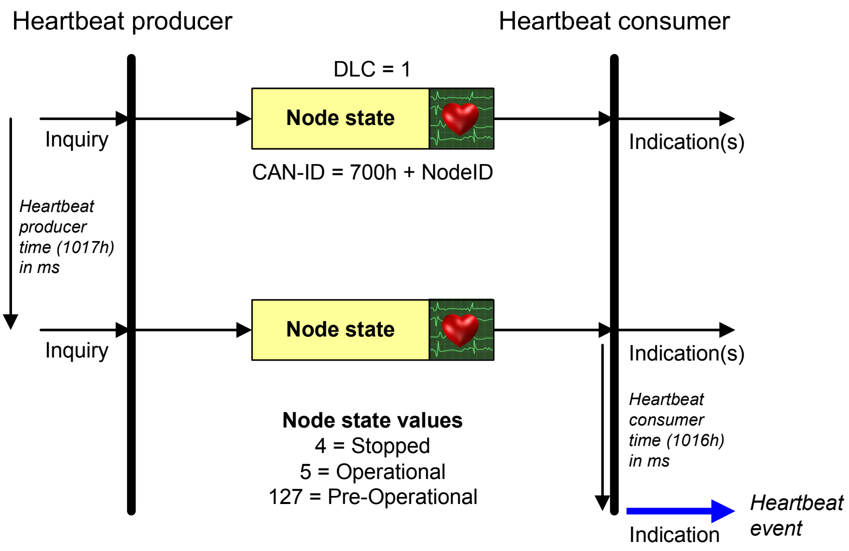
CANopen provides the following capabilities to determine the ability of the network nodes to function:

- Automatic transmission of a heartbeat message by the network nodes (Heartbeat principle)
- Cyclic querying of the node state by the NMT master (Node guarding principle)

With node monitoring according to the Heartbeat principle, every node automatically transmits a message at regular intervals. This message can be monitored by every node in the network. The interval between two heartbeat messages can be set in object 1017h.

With the Node guarding protocol, the NMT master sends messages to the CANopen slaves that then respond within a defined time. The lack of a response can only be detected by the NMT master. If the NMT master fails, the entire network is paralyzed. For this reason, and because of the higher bus load (caused by two CAN messages per monitoring interval), Node guarding has almost completely been replaced by Heartbeat monitoring.

The monitoring message of the nodes contains the COB ID 700h + the node ID of the sender. The only data byte transmitted contains the device state (Pre-Operational, Operational, Stopped) of the sender.



4.5 Additional definitions

4.5.1 Boot-up message

The Boot-up message is the first sign of life from a CANopen device following power-up or a reset. This message signals that the nodes have completed initialization and are transitioning into the Pre-Operational state.

4.5.2 EDS

The Electronic Data Sheet (EDS) describes the functionality of a CANopen device in machine-readable form. These files, in a standardized text format, describe both all supported objects from the object directory of the device, various data about the device and the vendor as well as physical parameters such as the baud rates supported.
Almost all CANopen control systems can read EDS files and make it easier for the system integrator to parameterize the system.

4.5.3 DCF

The Device Configuration File (DCF) uses the EDS file as a basis and also contains the values of each object.
This file can be used for the automatic configuration of CANopen devices.

4.5.4 LSS

The Layer Setting Services (LSS) is a service that can be used to set the ID and the bit rate of a device.
The identifiers $7E4_H$ and $7E5_H$ are reserved for this. The service can be used in a peer-to-peer connection from the master to the device or over the bus.

5 CANopen protocol

5.1 General

These operating instructions reproduce the current state of the implemented functions of the modules (described in the following chapters).

You can obtain more detailed literature from the user organization:
CAN in Automation (CiA)
Kontumazgarten 3
DE-90429 Nürnberg
headquarters@can-cia.org
www.can-cia.org

You do not need any aids for this CANopen pressure sensor to change the identification and the baud rate. You also do not need to open the sensor. The communications parameters can be defined and saved using the software.

You can find the information on the CANopen master in the documentation for the devices you are using.

5.1.1 Boot loader

A boot loader is implemented in the sensor. Upon request, this can be used to update the firmware of the sensor at the customer's location in the CAN network.

5.2 Network Management

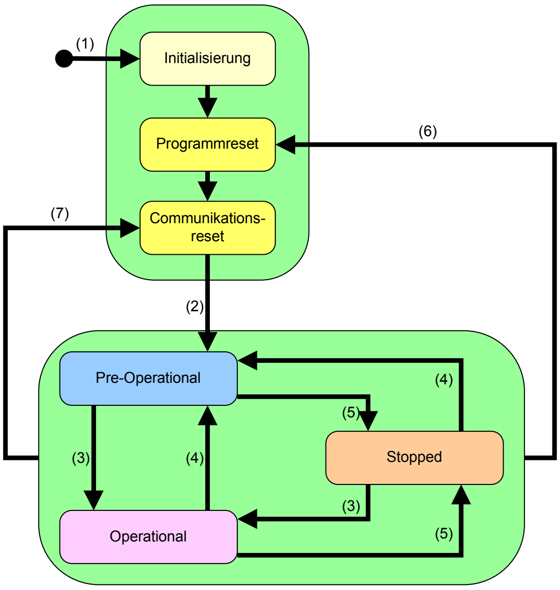
After power is turned on at the CANopen pressure sensor, the sensor responds by sending the CAN Boot-up message. This is a message without data bytes having the COB identifier 1792_D + module ID (700_H + ID).

5.2.1 Predefined connection set

COB ID = Function code (4 bits) + module ID (7 bits)

Object	COB ID (decimal)	COB ID (hex)
Network Management	0	0
Sync	128	80h
Emergency	129 – 255	81h – FFh
PDO1 (tx)	385 – 511	181h – 1FFh
PDO2 (tx)	641 – 767	281h – 2FFh
SDO (tx)	1409 – 1535	581h – 5FFh
SDO (rx)	1537 – 1663	601h – 67Fh
Heartbeat	1793 – 1919	701h – 77Fh

5.2.2 Start procedure



(1) Power on
(2) Automatical alternation to
Pre-Operational
(3) Alternation to Operational
(4) Alternation to Pre-Operational
(5) Alternation to Stopped
(6) Reset Application
(7) Communcationsreset

<u>Initialization</u>
This is the state that a node passes through following power-on. During this phase, the device application and device communications are initialized. Then, the node automatically transitions to the Pre-Operational state.

<u>Pre-Operational</u>
In this state, the node waits for the Operational mode to be enabled. The possible communications are shown in the table below.

<u>Operational</u>
In this state, the CANopen node is completely ready for operation and can transmit messages (PDOs, Emergency) on its own.

Communications possible during the various modes:

	Initialization	**Pre-Operational mode**	**Operational mode**	**Stopped mode**
PDO			X	
SDO		X	X	
Sync indexes			X	
Emergency indexes		X	X	
Boot-up indexes	X			
Network Management		X	X	X
Heartbeat		X	X	X
LSS				X

5.2.3 Start Node

Transition to Operational Mode

ID	DLC	Byte1	Byte2
0	2	01h	Node

Node = module address, 0 = all nodes

5.2.4 Stop Node

Transition to Stopped Mode

ID	DLC	Byte1	Byte2
0	2	02h	Node

Node = module address, 0 = all nodes

5.2.5 Pre-Operational Node

Transition to Pre-Operational Mode

ID	DLC	Byte1	Byte2
0	2	80h	Node

Node = module address, 0 = all nodes

5.2.6 Reset Node

Software reset of the node

ID	DLC	Byte1	Byte2
0	2	81h	Node

Node = module address, 0 = all nodes

Reset Node is corresponding to a Power On Reset.

5.3 Supported Object Overview

The following table is a summary of the supported SDO objects.

Index	16bit index number in hex
Sub-index	Sub-index in hex
Name	Name of objects / Sub-index
Data type	U/I = Unsigned/Integer, value = number of data bits, ARR = array, REC = record
Acc	ro = read only, wo = write only, rw = read & write
Default	Default value used of first initial and load default
PDO mapping	Mapping of object possible, TPDO = Transmit PDO, RPDO = Receive PDO
Page	Further information of the objects

Index	Sub-index	Name	Data type	Acc	Default	PDO mapping	Page
1000	00	Device type	U32	ro	00‘02‘01‘94h	-	28
1001	00	Error register	U8	ro	00h	TPDO	44
1002	00	Calibration date	U32	ro	e.g. 10‘07‘14h (14 July 2010)	-	28
1003		Emergency history	ARR				44
	00	Number of errors	U8	rw	00h	-	
	01	Last error	U32	ro	-	-	
	02-0F	Older errors	U32	ro	-	-	
1005	00	COB-ID SYNC message	U8	ro	80h	-	43
1008	00	Device name	U32	ro	„PDRJ“	-	28
1009	00	Hardware version	U32	ro	e.g. „3.02“	-	29
100A	00	Software version	U32	ro	e.g. „2.02“	-	29
1010		Store parameters	ARR				31
	00	Highest sub-index supported	U8	ro	01h	-	
	01	Save all parameters	U32	rw	„save“	-	
1011		Restore default parameter	ARR				32
	00	Highest sub-index supported	U8	ro	01h	-	
	01	Restore all default parameters	U32	rw	„load“	-	
1014	00	COB-ID Emergency	U32	ro	00‘00‘00‘81h	-	47
1017	00	Producer heartbeat time	U16	rw	00‘00h	-	48
1018		Identity object	REC				29
	00	Highest sub-index supported	U8	ro	04h	-	
	01	Vendor-ID	U32	ro	00‘00‘00‘5Fh	-	
	02	Product code	U32	ro	e.g. 11038931d	-	
	03	Revision number	U32	ro	e.g. 00‘03‘02‘02h	-	
	04	Serial number	U32	ro	e.g. 00000000d	-	
1800		Transmit PDO1 communication	REC				41
	00	Highest sub-index supported	U8	ro	05h	-	
	01	COB-ID and activation TPDO1	U32	rw	40‘00‘01‘81h	-	
	02	Transmission type	U8	rw	FFh	-	
	05	Event timer	U16	rw	03‘E8h	-	
1801		Transmit PDO2 communication	REC				42
	00	Highest sub-index supported	U8	ro	05h	-	
	01	COB-ID and activation TPDO2	U32	rw	40‘00‘02‘81h	-	
	02	Transmission type	U8	rw	02h	-	
	05	Event timer	U16	rw	03‘E8h	-	
1A00		Transmit PDO1 mapping	ARR				42
	00	Number of mapped objects TPDO1	U8	rw	01h	-	
	01	First application object	U32	rw	71‘30‘01‘10h	-	
	02-04	Further application objects	U32	rw	*[2]	-	
1A01		Transmit PDO2 mapping	ARR				43
	00	Number of mapped objects TPDO2	U8	rw	01h	-	
	01	First application object	U32	rw	71‘30‘02‘10h	-	
	02-04	Further application objects	U32	rw	*[2]	-	
2000	00	Averaging time	U16	rw	00‘1Eh	-	38
2100	00	Baud rate	U8	rw	03h	-	39

Index	Sub-index	Name	Data type	Acc	Default	PDO mapping	Page
2101	00	Identification	U8	rw	01h	-	40
6110		Sensor type	ARR				33
	00	Highest sub-index supported	U8	ro	01h	-	
	01	Sensor type	U16	ro	46h	-	
6112		Operating mode	ARR				33
	00	Highest sub-index supported	U8	ro	01h	-	
	01	Operation mode PV1	U8	ro	01h	-	
6131		Physical unit PV	ARR				34
	00	Highest sub-index supported	U8	ro	01h (02h*[1])	-	
	01	Physical unit PV1	U32	ro	00’00‘4E‘00h	-	
	02*[1]	Physical unit PV2	U32	ro	00’00’2D’00h	-	
6132		Decimal digits PV	ARR				34
	00	Highest sub-index supported	U8	ro	01h (02h*[1])	-	
	01	Decimal digits PV1	U8	ro	e.g. 02h	-	
	02*[1]	Decimal digits PV2	U8	ro	e.g. 01h	-	
6150		Status of measurement	ARR				35
	00	Highest sub-index supported	U8	ro	01h (02h*[1])	-	
	01	Status of measurement PV1	U8	ro	e.g. 00h	TPDO	
	02*[1]	Status of measurement PV2	U8	ro	e.g. 00h	TPDO	
7130		Process value 16bit	ARR				35
	00	Highest sub-index supported	U8	ro	01h (02h*[1])	-	
	01	Process value PV1 16bit	I16	ro	e.g. 01‘2Ch	TPDO	
	02*[1]	Process value PV2 16bit	I16	ro	e.g. 00’C8h	TPDO	
7133		Interrupt delta input 16bit	ARR				37
	00	Highest sub-index supported	U8	ro	01h (02h*[1])	-	
	01	Interrupt delta input PV1 16bit	U16	rw	00‘00h	-	
	02*[1]	Interrupt delta input PV2 16bit	U16	rw	00’00h	-	

*[1] Sub-index only available if the sensor as an temperature output (S- and T-types)
*[2] Sub-index is not accessible, accessible for customer mapping only

5.4 SDO-Struktur

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
-	8	CMD	Index		Sub-index	Data bytes			

Procedure	CMD	Remarks
Master request data from slave	40h	
Slave responds	42h	(valid data bytes not specified)
	43h	(4 valid data bytes)
	47h	(3 valid data bytes)
	4Bh	(2 valid data bytes)
	4Fh	(1 valid data bytes)
Master writes to slave	22h	(valid data bytes not specified)
	23h	(4 valid data bytes)
	27h	(3 valid data bytes)
	2Bh	(2 valid data bytes)
	2Fh	(1 valid data bytes)
Slave responds	60h	

In index and data bytes, the lowest byte is transmitted first. In ASCII code the bytes will be transferred legible (first character first).

The range of the communication profile is in the indices 1000h-1FFFh and includes all parameters which concern the CAN network. This range is defined in the same way in all CANopen devices.

The minimum time difference between two SDO messages must not be less than 20ms. Faster SDO communication can put the device into undefined states.

6 Object description

6.1 Standard objects

6.1.1 Device type

Read device type (object 1000h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	00h	10h	00h	94h	01h	02h	00h

Byte 5 + 6: 01’94h = 404d (Device profile number)
Byte 7 + 8: 00’02h (additional information, analog input)

The object 1000h is read only and has no sub-index.

6.1.2 Calibration date

Read calibration date (object 1002h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	02h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	02h	10h	00h	09h	0Bh	14h	00h

Example of calibration date 20.11.09:
Byte 5: 09h Year 09
Byte 6: 0Bh Month 11 (November)
Byte 7: 14h Day 20
Byte 8: reserved

The object 1002h is read only and has no sub-index.

6.1.3 Device name

Read device name (object 1008h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	08h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	08h	10h	00h	50h	44h	52h	4Ah

Example of pressure sensor:
Byte 5 – 8: 50’44’52’4Ah „PDRJ“ in ASCII format

The object 1008h is read only and has no sub-index.

6.1.4 Hardware version

Read hardware version (object 1009h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	09h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	09h	10h	00h	33h	2Eh	30h	32h

Example of hardware version 3.02:
Byte 5 – 8: 33’2E’30’32h „3.02“ in ASCII format

The object 1009h is read only and has no sub-index.

6.1.5 Software version

Read software version (object 100Ah):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	0Ah	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	0Ah	10h	00h	32h	2Eh	30h	32h

Example of software version 2.02:
Byte 5 – 8: 32’2E’30’32h „2.02“ in ASCII format

The object 100Ah is read only and has no sub-index.

6.1.6 Identity object

Structure of object 1018h which contents general information about the device:

Index	Sub-index	Name	Length	Access
1018h	0	Highest sub-index	1 Byte	Read
	1	Vendor-ID	4 Byte	Read
	2	Product code	4 Byte	Read
	3	Revision number	4 Byte	Read
	4	Serial number	4 Byte	Read

Read vendor-ID (sub-index 1):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	01h	5Fh	00h	00h	00h

Byte 5 – 8: 5F’00’00’00h 00’00’00’5F (LSB first) ➔ Baumer Company

Read product code (sub-index 2):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	02h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	02h	44h	0Dh	A8h	00h

Example of product code 11013444:
Byte 5 – 8: 44’0D’A8’00h 00'A8'0D'44 (LSB first) ➔ 11013444 in decimal

Read revision number (sub-index 3):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	03h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	03h	02h	02h	03h	00h

Example of revision number 00030202:
Byte 5 – 8: 02’02’03’00h 00’03’02’02 (LSB first) ➔ 00030202h

Read serial number (Sub-Index 4):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	04h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	04h	7Bh	00h	00h	00h

Example of serial number 123:
Byte 5 – 8: 7B’00’00’00h 00’00’00’7B (LSB first) ➔ 123

The object 1018h is read only.

6.2 Parameter handling (save, load default)

The following changeable objects can be saved in the EEPROM:

Object	Name	Default value	
1017	Producer heartbeat time	00h	Disabled
1800	PDO1 – COB-ID + PDO valid , sub-index 1	40’00’01’xxh	COB-ID = 180h+ID, PDO enabled
	PDO1 – transmission type , sub-Index 2	FFh	Transmit after event timer
	PDO1 – event time , sub-index 5	03’E8h	1000 ms
1801	PDO2 – COB-ID + PDO valid , sub-index 1	40’00’02’xxh	COB-ID = 280h+ID, PDO enabled
	PDO2 – transmission type , sub-index 2	02h	Transmit after 2nd SYNC
	PDO2 – event time , sub-index 5	03’E8h	1000 ms
1A00	Transmit PDO1 mapping, sub-index 1	71’30’01’10h	Index 7130, sub-index 1 mapped
1A01	Transmit PDO2 mapping, sub-index 1	71’30’02’10h (71’30’01’10h)	Index 7130, sub-index 2 mapped only if activated, otherwise sub-index 1
2000	Averaging time	00’30h	30 ms
2100	Baud rate	03h	125 kBaud
2101	Identification (7 bit)	01h	ID 1
7133	Interrupt delta input PV1 16bit	00h	Disabled
	Interrupt delta input PV2 16bit	00h	Disabled

6.2.1 Store parameters

With object 1010h, the current parameters can be stored in EEPROM. The indices which are stored can be seen in the table at the beginning of Section 6.2.

Storage takes place when the “save” message is sent in ASCII code to index 1010h, sub-index 1. The message thus has the following structure:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	10h	10h	01h	73h	61h	76h	65h

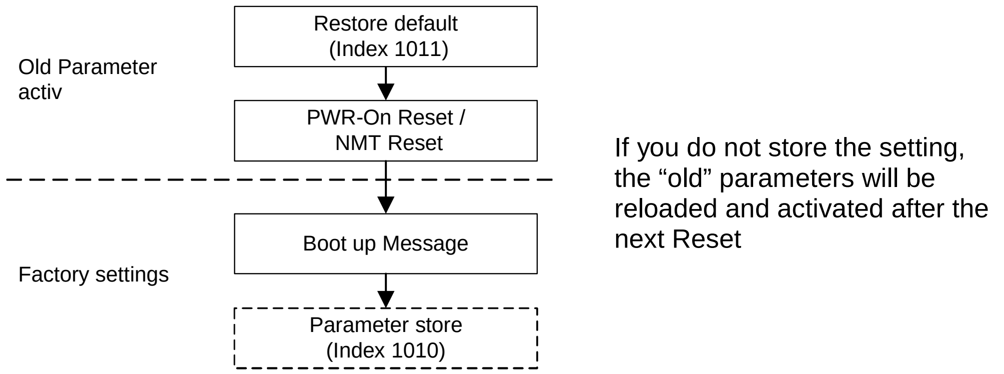
Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	10h	10h	01h	0	0	0	0

6.2.2 Restore default parameters

With object 1011h, the factory settings can be loaded. In this process all parameters, except the communication parameters (Baud rate, Identification), will be set to default values.
For the parameters and the corresponding values, see the start of Section 6.2.

<u>Function mode:</u>



Loading the data takes place when the index 1011h with the “load” message in ASCII code on sub-index 1 is sent. The message thus has the following structure:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	11h	10h	01h	6Ch	6Fh	61h	64h

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	11h	10h	01h	0	0	0	0

Loading the default values means that the values are loaded into RAM. If the values are to remain at the next reset, the parameters must be stored in EEPROM using the index 1010h.

6.3 Device profile specific objects

Setting and interrogating the sensor-specific values. The following indices are supported:

Object	Name
6110	Sensor type
6112	Operating mode
6131	Physical unit PV
6132	Decimal unit PV
6150	Status of measurement
7130	Process value 16bit
7133	Interrupt delta input 16bit

6.3.1 Sensor type

Read sensor type (object 6110h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	10h	61h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	10h	61h	01h	5Ah	00h	0	0

Byte 5:	5Ah	pressure sensor

The object 6110h is read only.

6.3.2 Operation mode

When delivered, the sensor is always in normal mode.

Read operation mode (object 6112h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	12h	61h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	12h	61h	01h	01h	0	0	0

Byte 5:	01h	Acquisition mode
	02h – 09h	reserved
	0Ah	(Adjust mode)

The object 6112h is read only.

6.3.3 Physical unit PV

Physical unit from process value (object 6131h)

Index	Sub-index	Parameter	Length	Access
6131h	0	Highest sub-index	1 Byte	read
	1	Physical unit process value1	2 Byte	read
	2*	Physical unit process value2	2 Byte	read

*only for sensors with temperature output: S- & T-types

Read physical unit from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	31h	61h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	31h	61h	01h	00h	00h	4Eh	00h

Byte 4:	01h	Sub-index 1 (unit pressure)
	02h	Sub-index 2 (unit temperature)
Byte 5 – 8:	00’00’4E’00h	00’4E’00’00 (LSB first) ➔ unit “Bar”
	00’00’4E’FDh	FD’4E’00’00 (LSB first) ➔ unit “mBar”
	00’00’2D’00h	00’2D’00’00 (LSB first) ➔ unit “℃”
	00’00’AC’00h	00’AC’00’00 (LSB first) ➔ unit “℉”
	00’00’05’00h	00’05’00’00 (LSB first) ➔ unit “Kelvin”

Units of the process value are depending of the sensor type and can not be modified

The object 6131h is read only.

6.3.4 Decimal digits PV

Decimal digits from the process value (object 6132h)

Index	Sub-index	Parameter	Length	Access
6132h	0	Highest sub-index	1 Byte	read
	1	Decimal digits process value 1	1 Byte	read
	2*	Decimal digits process value 2	1 Byte	read

*only for sensors with temperature output: S- & T-types

Read decimal digit from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	32h	61h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	32h	61h	01h	02h	0	0	0

Byte 4:	01h	Sub-index 1 (decimal digits pressure)
	02h	Sub-index 2 (decimal digits temperature)
Byte 5:	02h	2 decimal digits

The object 6132h is read only.

6.3.5 Status of measurement

Status of measurement (object 6150h)

Index	Sub-index	Parameter	Length	Access
6150h	0	Highest sub-index	1 Byte	read
	1	Status of measurement PV 1	1 Byte	read
	2*	Status of measurement PV 2	1 Byte	read

*only for sensors with temperature output: S- & T-types

Read status of measurement from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	50h	61h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	50h	61h	01h	00h	0	0	0

Byte 4:	01h	Sub-index 1 (status pressure measurement)
	02h	Sub-index 2 (status temperature measurement)
Byte 5:	00h	actual measurement
	03h	overflow of AD – converter
	05h	underflow of AD – converter

The object 6150h is read only.

6.3.6 Process value 16bit

Process value 16bit (object 7130h)

Index	Sub-index	Parameter	Length	Access
7130h	0	Highest sub-index	1 Byte	read
	1	Request of process value 1	2 Byte	read
	2*	Request of process value 2	2 Byte	read

*only for sensors with temperature output: S- & T-types

Read the 16bit process value from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	30h	71h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	30h	71h	01h	2Ch	01h	0	0

Byte 4:	01h	Sub-index 1 (pressure value)
	02h	Sub-index 2 (temperature value)
Byte 5 + 6:	2C'01h	01'2C (LSB first) ➔ 300
		300 / $10^{\text{Number decimal digits}}$ (6132) + unit(6131) = 3.00Bar

The process value is quoted as 16-bit Integer double complement.

Example of negative temperature:

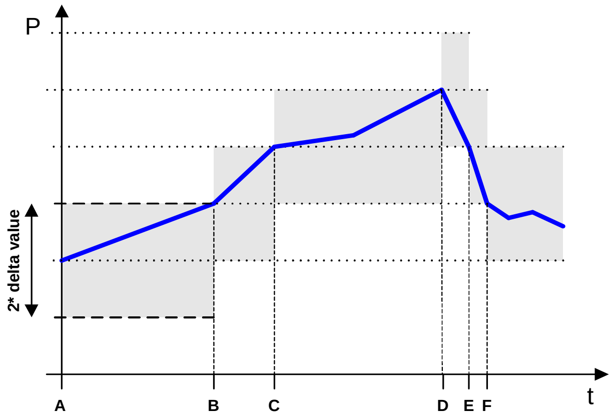
ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	30h	71h	02h	ECh	FFh	0	0

Byte 5 + 6: EC'FFh FF'EC (LSB first) ➔ 65516d = -20
-20 / 10$^{\text{Number decimal digits}}$ (6132) + unit(6131) = -2.0 ℃

The object 7130h is read only.

6.3.7 Interrupt delta input

After the delta function is activated, when the threshold is passed the current process value is sent via PDO and a new threshold value is set. At the next over or undercut the threshold again a PDO will be sent. (Threshold = momentary process value +/- delta value)



A: Activation of delta function via PDO Transmission Type

B-F: Process value is sent via PDO, new threshold value is set

P: Process value
t: Time

With object 7133h (16bit) the interrupt delta input value can be read and written.

Index	Sub-index	Parameter	Length	Access
7133h	0	Highest sub-index	1 Byte	read
	1	Interrupt delta value PV 1	2 Byte	read / write
	2*	Interrupt delta value PV 2	2 Byte	read / write

*only for sensors with temperature output: S- & T-types

The delta function is activated or deactivated via the transmission type of the PDO1 (object 1800h) and PDO2 (object 1801h).

Change interrupt delta input (object 7133h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	33h	71h	01h	F4h	01h	0	0

Interrupt delta input value: 5 Bar $5 * 10^2 = 500 = 01'F4h$
Byte 4: 01h   Sub-index 1 ➔ value for PDO1
Byte 5 + 6: F4'01h   01'F4 (LSB first) ➔ 500

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	33h	71h	01h	0	0	0	0

If the value 0 is written into the object 7133h sub-index 01h, the delta value function will be switched off.

Read interrupt delta input (object 7133h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	33h	71h	01h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	33h	71h	01h	F4h	01h	0	0

Byte 5 + 6: F4'01h   01'F4 (LSB first) ➔ 500 ➔ 5 Bar

6.4 Manufacturer specific objects

The definition of the following objects is manufacturer specific. They are used to set the CANopen pressure sensor. List of supported objects:

Object	Name
2000	Averaging time
2100	Baud rate
2101	Identification

6.4.1 Averaging time

The average time specifies the time in ms over which the measured values are arithmetically averaged. The average time can be set between 0..1000 (0..3E8h).
The sensor internal averaging contains a 32bit buffer. If an averaging time >32ms is selected, just every 2nd measurement is used. With averaging time >64ms just every 3rd value is used, and so on.

Read averaging time (object 2000h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	20h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	00h	20h	00h	1Eh	00h	0	0

Byte 5 + 6: 1E’00h 00’1Eh (LSB first) ➔ 30ms (Default value)

Change of averaging time:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	20h	00h	64h	00h	0	0

Byte 5 + 6: 64’00h 00’64h (LSB first) ➔ 100ms

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	20h	00h	0	0	0	0

6.4.2 Baud rate

The baud rate specifies the speed at which the whole bus is operated. All users must have the same baud rate. The CANopen pressure sensor is shipped with 125kBaud.

Read baud rate (object 2100h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	21h	00h	0	0	0	0

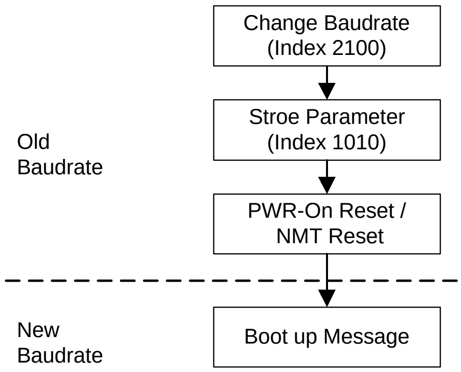
Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	00h	21h	00h	03h	0	0	0

Byte 5:

00h → 10 kBaud
01h → 20 kBaud
02h → 50 kBaud
03h → 125 kBaud
04h → 250 kBaud
05h → 500 kBaud
06h → 800 kBaud
07h → 1000 kBaud

Function mode:
Changing baud rate:



Change of baud rate to 500 kBaud:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	21h	00h	05h	0	0	0

Byte 5: 05h 5 → 500 kBaud

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	21h	00h	0	0	0	0

6.4.3 Identification

The ID of a device in a CANopen network must be unique. Otherwise 2 devices at a time are addressed. The identification can be set from 1 to 127 => 7bit.

Read identification number (object 2101h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	01h	21h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	01h	21h	00h	01h	0	0	0

Byte 5: 01h ID 1 (Default value)

Change of identification to ID 5:

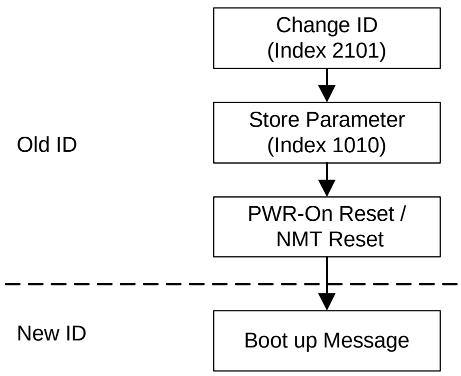
ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	01h	21h	00h	05h	0	0	0

Byte 5: 05h ID 5

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	01h	21h	00h	0	0	0	0

Function mode:



6.5 PDO communication objects

Via communication using PDO (Process Data Objects), it is possible to interrogate specified values of the CANopen pressure sensor simply and quickly.
A variable PDO mapping is implemented in the pressure sensor. The send parameters and the conditions for the PDO sending can be defined by the user. The PDO can be activated by a sync message, after expiration of an event time or by overwriting of a interrupt delta input value.
Communication via PDO is only possible in the operational mode of the sensor.

6.5.1 Transmit PDO 1 communication

Via the object 1800h the settings are made to make it possible to work with Transmit PDOs. PDOs are processed only in the Operational mode of the device. The PDO must be activated.

The PDO should not be requested faster than the corresponding measurement rate.
(1000 measurements / second) > 1 ms

The object has the following structure:

Index	Sub-index	Name	Length	Access
1800h	0	Highest sub-index supported	1 Byte	read
	1	COB-ID and activation TPDO 1	4 Byte	read / write
	2	Transmission type	1 Byte	read / write
	5	Event timer	2 Byte	read / write

Change of transmit PDO from enable to disable (sub-index 1):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	18h	01h	81h	01h	0	40h

Byte 5 + 6:	81'01h	01'81h (LSB first) → 180h + ID 1 (COB-ID)
Byte 8:	40h	PDO enabled and RTR not supported
	C0h	PDO disabled and RTR not supported

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	01h	0	0	0	0

Change of transmission type (sub-index 2):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	18h	02h	FFh	0	0	0

Byte5	01h – F0h	transmit after n^{th} Sync (1 – 240)
	FEh	transmit when interrupt delta input (interrupt delta input value is set with object 7133h)
	FFh	transmit when event timer (Event timer is set with sub-index 05h)

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	02h	0	0	0	0

The event timer defines in which cycle the PDO will be transmitted.
The value range is fixed between 0…2^{16} – 1 = 65535 (ms).

Change of event timer (sub-index 5):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	18h	05h	E8h	03h	0	0

Byte 5 + 6: E8’03h 03’E8h (LSB first) ➔ Event timer 1000 ms (Default value)

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	05h	0	0	0	0

Too small values make no sense, because they overload the bus and the measured values may still not be updated (see start of this section).

The PDO parameters are not stored automatically, must be done with object 1010h.

6.5.2 Transmit PDO 2 communication

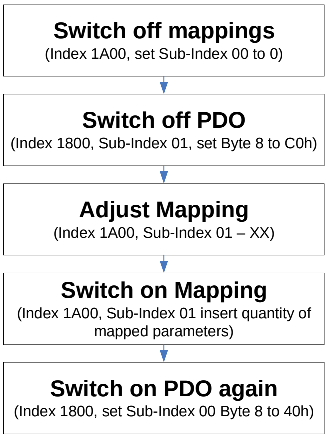
Object 1801h (Transmit PDO communication parameter):
The setting of Transmit PDO2 is made with the object 1801h.
This object is similar to the object 1800h. Detailed information se the previous section in this manual.

6.5.3 Transmit PDO 1 mapping parameter

Via the object 1A00h, the mapping of the Transmit PDO 1 can be interrogated.
The object has the following structure:

Index	Sub-index	Name	Length	Access
1A00h	0	Number of mapped objects TPDO1	1 Byte	read / write
	1	1st application object	4 Byte	read / write
	2	2nd to 4th application object	4 Byte	read / write

How to set the mapping:



The sent PDO 1 has 8 Bytes. This PDO is freely configurable by the user.
The following objects can be chosen:

Index	Sub-index	Name	Length
7130	01	Process value 1 16bit	2 Bytes
7130	02*	Process value 2 16bit	2 Bytes
6150	01	Status of measurement PV 1	1 Byte
6150	02*	Status of measurement PV 2	1 Byte
1001	00	Error register	1 Byte

Mapping of PDO 1 via SDO command:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	1Ah	01h	10h	01h	30h	71h

Example of PDO 1 with process value 16bit:

Byte 4:	01h	1st mapped object in PDO 1
	02h – 04h	2nd to 4th mapped object in PDO 1 (attention: only 8Byte)
Byte 5:	10h	16 Bit data length of the mapped object (2 Byte)
Byte 6:	01h	mapped sub-index 01
Byte 7 + 8:	30’71h	7130 ➔ mapped index

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	1Ah	01h	0	0	0	0

When the PDO 1 is transmitted the message has the following structure:

ID	DLC	Byte1	Byte2*	Byte3*	Byte4*	Byte5*	Byte6*	Byte7*	Byte8*
180h + ID	2	68h	10h						

*Bytes will only be sent when they were mapped in index 1A00.

6.5.4 Transmit PDO 2 mapping

Object 1A01h (Transmit PDO mapping parameter):
The mapping of Transmit PDO2 is made with the object 1A01h.
This object is similar to the object 1A00h. Detailed information se the previous section in this manual.

6.5.5 Sync ID

The sync generation is switched off in the sensor. PDO messages are only sent when the object 180xh sub index 2 is switched on.
With the object 1005h the ID of the sync message can be interrogated. If a sync message with the following ID is on the bus, the PDO can be triggered (compare PDO communication).

Read SYNC COB-ID (object 1005h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	05h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	05h	10h	00h	80h	0	0	0

The ID is defined as 80h. This ensures that the sync messages have a high priority on the CAN bus.

The object 1005h is read only and has no sub-index.

7 Emergency and services

7.1 Error register and history

7.1.1 Error register

Read error register (object 1001h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	01h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	01h	10h	00h	81h	0	0	0

Byte 5: 81h a manufacture error has occurred
Bit 0=1 ➔ error occurred
Bit 7=1 ➔ manufacturer specific error

The object 1001h is read only and has no sub-index.

7.1.2 Emergency History

The object 1003h stores the last 16 error messages (Emergency Messages) which have occurred during operation in the RAM. This means that at the next loss of power or manual deletion the data is deleted. The recording will be deleted if 00h is written on the sub index 0.

The object has the following structure:

Index	**Sub-index**	**Name**	**Length**	**Access**
1003h	0	Number of errors	1 byte	Read/Write
	1...16	Error messages	8 byte	Read

Read number of errors (sub-Index 0):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	03h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	03h	10h	00h	02h	0	0	0

Byte 5: 02h 2 error messages have been recorded

Delete the emergency messages by writing 0 to sub-index 0:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	03h	10h	00h	00h	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	03h	10h	00h	0	0	0	0

Read error message (sub-index 1...16):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	03h	10h	01h	0	0	0	0

Byte 4:	01h	last recorded error message
	02h – 10h	older error messages

Response from CANopen pressure sensor:

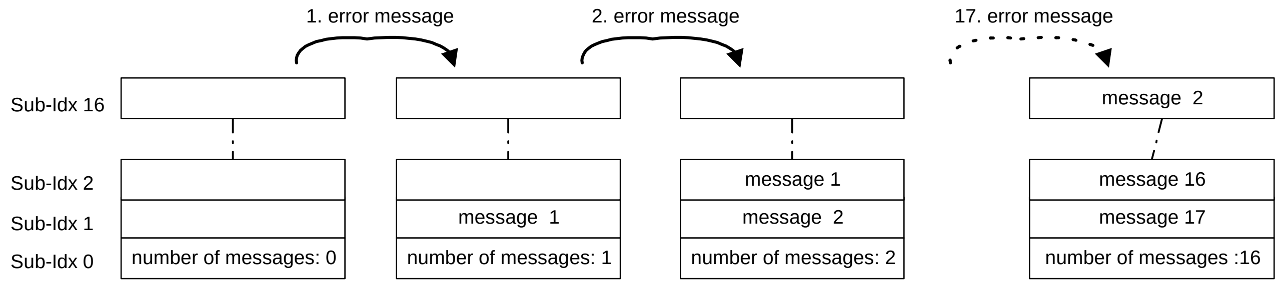
ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	03h	10h	01h	00h	FFh	81h	14h

The error message code is descript in section „Emergency messages“.

Request a sub-index without occurred error the following error message will be received:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	80h	03h	10h	01h	11h	00h	09h	06h

Function mode:



The sensor can store the last 16 error messages. The last message is stored under sub-index 1 and all previous ones are pushed upward by one position.
If the memory is full and a new message appears, it is stored under sub-index 1 and the oldest message (sub-index 16) is pushed out of the memory.

7.2 SDO error messages

In the case of wrong access to an index you receive an error message as the response. An error message has the following structure:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	80h	Index		Sub-index	Abort code			

The ID of the message and the index and sub-index refer to the message which has caused an error.

The error messages can have the following contents:

Abort code	**Meaning**
05 04 00 01h	Client / server command is invalid or unknown
06 01 00 01h	Write only access possible
06 01 00 02h	Read only access possible
06 02 00 00h	Object does not exist
06 04 00 41h	Object cannot be mapped to the PDO.
06 04 00 42h	The number and length of the objects to be mapped would exceed PDO length
06 07 00 10h	Data type does not match, length of service parameter does not match
06 09 00 11h	Sub-index does not exist
06 09 00 30h	Invalid value for parameter
06 09 00 31h	Value of parameter written too high
06 09 00 32h	Value of parameter written too low
08 00 00 20h	Data cannot be saved, signature is invalid
08 00 00 21h	Data cannot be saved, communication with memory component has failed
08 00 00 22h	Data cannot be stored because the memory component is already being accessed
08 00 00 24h	No data available

7.3 Emergency Messages

Emergency messages will be sent from the sensor independently in error case. When an error occurs for the first time an error message will be sent. If the error is eliminated or it is not pending anymore an appropriate error message will be sent.
The last 16 error messages will be saved in the emergency history.

The error messages have the following structure:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
80h + ID	8	Emergency error code		Error register	Manufacturer specific error code				

Following error codes will be supported:

Emergency error code	Error register	Manufact. error code	Description
50’01h	81h	10h	EEPROM read error (hardware error) Boot up: Error will be sent with ID1 and 125kBaud
00’00h	00h	20h	EEPROM write error released Saving of autozero value in EEPROM successful
50’01h	81h	30h	EEPROM write error (hardware error) By saving the autozero value in the EEPROM
00’00h	00h	11h	Internal stain signal has reached valid range
FF’00h	81h	12h	Internal pressure signal value passed maximum limit
FF’00h	81h	14h	Internal pressure signal value passed minimum limit
00’00h	00h	21h	Internal temperature signal reached valid range
FF’00h	81h	22h	Internal temperature signal passed maximum limit
FF’00h	81h	24h	Internal temperature signal passed minimum limit

Object 1014h (COB-ID Emergency object):
When an error occurs the ID will be stored in this index. Concerning this error message the ID can be assigned to a sensor.

Read COB-ID Emergency object (Objekt 1014h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	14	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	14h	10h	00h	81h	0	0	0

Byte 5	81h	ID = 80h + Node ID

The object 1014h is read only and has no sub-index.

7.4 Heartbeat

The CANopen pressure sensor offers the option of the heartbeat protocol. The Heartbeat protocol makes an error control system possible without an interrogation being necessary. The Heartbeat producer transmits the status message cyclically (defined in object 1017h). If the message does not arrive within the defined time, the CAN bus controller can send a corresponding reaction (Network management commands).
The heartbeat time can be adjusted between 1 and 65535 (1ms up to 65.535 seconds).

7.4.1 Producer heartbeat time

Read producer heartbeat time (object 1017h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	17h	10h	00h	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	17h	10h	00h	00h	00h	0	0

Byte 5 + 6: 00’00h 00’00h (LSB first) ➔ disabled (Default value)

Change producer heartbeat time to 1000ms (1000d ➔ 3E8h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	17h	10h	00h	E8h	03h	0	0

Byte 5 + 6: E8’03h 03’E8h (LSB first) ➔ 1000d ➔ 1000ms

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	17h	10h	00h	0	0	0	0

If this time is set to 0 no Heartbeat protocol takes place (default value).

The value can be stored in the EEPROM with the object 1010h.

After the heartbeat protocol is activated, the sensor sends the following messages:

ID	DLC	Byte1
700h + ID	1	04h

Byte 1:	00h	boot up
	04h	stop mode
	05h	operate
	7Fh	pre-operational

7.5 LSS (Layer setting services)

7.5.1 Printed LSS information on the sensor

To use the LSS functionality LSS data, which clearly identifies the sensor, is needed. The data are stored in the identity object (object 1018h). This data is printed on every sensor.

Example of a printed sensor with LSS information:

Baumer

PDAJ A008.14C.S316

Vs	12-33VDC
Output	CANopen DS404
PNom	0-250bar R
Pin 1	n.c.
Pin 2	+Vs
Pin 3	GND
Pin 4	CANH
Pin 5	CANL
CASE	Shield

LSS- Data (dec):

VenID	95
ProdC	163970
RevNr	197121
SerNr	000001

S501 8.07.163970.000001

Swiss made

LSS data are displayed in decimal number system.

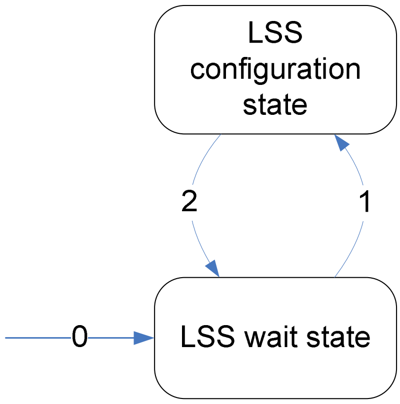
Title:	**Description:**	**Index:**	**Sub-index:**
VenID	Vendor-ID	1018h	01h
ProdC	Product code	1018h	02h
RevNr	Revision number	1018h	03h
SerNr	Serial number	1018h	04h

7.5.2 Address the sensor with LSS

There are two possibilities to communicate with the sensor by LSS. The LSS service supports sensors in an existing network but also sensors with a master.
In both cases the bus has to be brought in stop mode.
In LSS mode the sensors are addressed with ID 7E5h. The respond is with ID 7E4h.

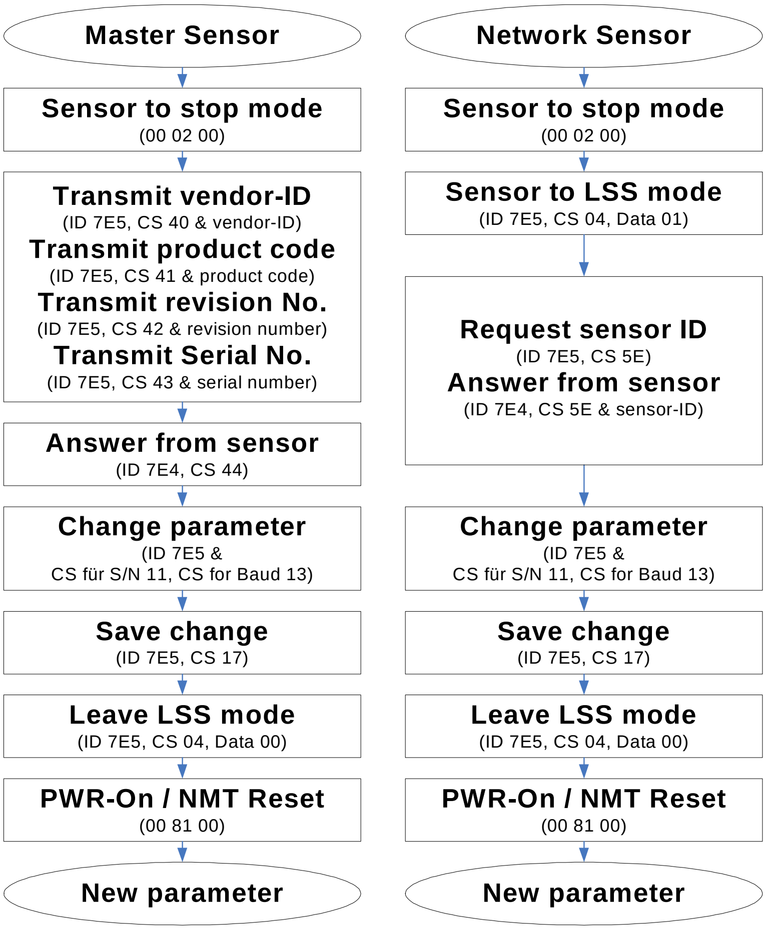
Bring all sensors to stop mode:

ID	DLC	Byte1	Byte2
0	2	02h	00h



0 = automatically enter wait state (sensor in stop mode)
1 = enter to configuration state (cs=04, Data=01)
2 = back to wait state (cs=04, Data=00)

There are different ways to put the sensor in the desired configuration mode (direct connection master-sensor or sensor in network).



7.5.3 Configuration mode direct connection (master sensor)

To use LSS the sensor has to be in the LSS configuration mode.
The sensor can be set into the LSS configuration mode as follows:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	04h	01h	0	0	0	0	0	0

The current set ID can be interrogated to test if the sensor has switched into the configuration mode:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	5Eh	0	0	0	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4	8	5Eh	01h	0	0	0	0	0	0

Example “ID 1“

Byte 2:	01h	accessed sensor has ID 1

If the sensor does not respond, the sensor does not support LSS or the baud rate is incorrect. For safe access the service and its baud rates has to be tested.
Now the ID or the baud rate can be changed.

7.5.4 Configuration mode of a sensor in a network

With the following orders the sensor can be identified:
Transmission of vendor ID:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	40h	5Fh	00h	00h	00h	0	0	0

Vender ID:

Byte 1:	40h	System byte vender ID
Byte 2 – 5:	5F’00’00’00h	00’00’00’5F (LSB first) ➔ Firma Baumer electric

Transmission of product code:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	41h	44h	0Dh	A8h	00h	0	0	0

Product code:

Byte 1:	41h	System byte product code
Byte 2 – 5:	44’0D’A8’00h	00’A8’0D’44 (LSB first) ➔ 11013444 in decimal

Transmission of revision number:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	42h	02h	02h	03h	00h	0	0	0

Revision number:

Byte 1:	42h	System byte revision number
Byte 2 – 5:	02’02’03’00h	00’03’02’02 (LSB first) ➔ 00030202h

Transmission of serial number:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	43h	7Bh	00h	00h	00h	0	0	0

Serial number:

Byte 1:	43h	System byte serial number
Byte 2 – 5:	7B’00’00’00h	00’00’00’7B (LSB first) ➔ 123

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4	8	44h	0	0	0	0	0	0	0

The pressure sensor confirms the identification with the answer. Now the ID and baud rate of the sensor can be changed

7.5.5 Changing ID and baud rate

Set of new ID:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	11h	01h	0	0	0	0	0	0

Example ID = 1:

Byte 2:	01h	ID = 1

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4	8	11h	00h	0	0	0	0	0	0

Byte 2:	00h	ID successfully changed
	01h	ID beyond valid range
	FFh	specific failure

Set new baud rate:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	13h	00h	04h	0	0	0	0	0

Example baud rate = 125kBaud:

Byte 2:	00h	CiA baud rate table
Byte 3:	04h	4 = 125kBau

Baud rates:

0=1Mbaud	3=250kBaud	7=20kBaud
1=800kBaud	4=125kBaud	8=10kBaud
2=500kBaud	6=50kBaud	

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4	8	13h	00h	0	0	0	0	0	0

Byte 2:	00h	Baud rate successfully changed
	01h	Baud rate beyond valid range
	FFh	specific failure

7.5.6 Save settings

To accept the changes the setting must be saved.

Saving LSS setup:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	17h	0	0	0	0	0	0	0

Response from CANopen pressure sensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4	8	17h	00h	0	0	0	0	0	0

Byte 2:	00h	Successfully saved
	01h	Save not possible
	FFh	specific failure

The modifications will be accepted after reset.

7.5.7 Leave LSS Mode

The sensor can be set with the following command to stop mode:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	04h	00h	0	0	0	0	0	0

No respond will be sent to this command.

8 Examples for users with the CANopen protocol

This chapter shows how to easy use a CANopen product. These examples can easy practice with a pressure sensor.

8.1 Read process value with SDO

Initialize of measurement:

1. Request unit (object 61'31h)

Master → Slave

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	31h	61h	01h	0	0	0	0

Slave → Master

ID	DLC	Byte1	Index		Sub.	Unit			
580h + ID	8	43h	31h	61h	01h	00h	00h	4Eh	00h

00'4E'00'00h => Bar

2. Request decimal digits (object 61'32h)

Master → Slave

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	32h	61h	01h	0	0	0	0

Slave → Master

ID	DLC	Byte1	Index		Sub.	Dig.	Byte6	Byte7	Byte8
580h + ID	8	4Fh	32h	61h	01h	02h	0	0	0

02h => 2 decimal digits => Divide with 10^2

Measurement cycle:

Object 71'30h, positive pressure

Master → Slave

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	30h	71h	01h	0	0	0	0

Slave → Master

ID	DLC	Byte1	Index		Sub.	Prozess value		Byte7	Byte8
580h + ID	8	4Bh	30h	71h	01h	2Ch	01h	0	0

01'2Ch => 300_{Dez}

Shift with decimal digits: 300 / 10^2 = 3.00

Add unit: 3.00 => <u>3.00 Bar</u>

Object 71'30h, negative pressure

Master → Slave

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	30h	71h	01h	0	0	0	0

Slave → Master

ID	DLC	Byte1	Index		Sub.	Prozess value		Byte7	Byte8
580h + ID	8	4Bh	30h	71h	01h	D4h	FEh	0	0

FE'D4h => 1111'1110'1101'0100b inv.
- 0000'0001'0010'1011b +1
- 0000'0001'0010'1100b = -300_{Dez}

Shift with decimal digits: -300 / 10^2 = -3.00

Add unit: -3.00 => <u>-3.00 Bar</u>

8.2 Set and request of process value with PDO1

Initialize of measurement:

1. Request unit (object 61'31h)

Master → Slave

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	31h	61h	01h	0	0	0	0

Slave → Master

ID	DLC	Byte1	Index		Sub.	Unit			
580h + ID	8	43h	31h	61h	01h	00h	00h	4Eh	00h

00'4E'00'00h => Bar

2. Request decimal digits (object 61'32h)

Master → Slave

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	32h	61h	01h	0	0	0	0

Slave → Master

ID	DLC	Byte1	Index		Sub.	Dig.	Byte6	Byte7	Byte8
580h + ID	8	4Fh	32h	61h	01h	02h	0	0	0

02h => 2 decimal digits => Divide with 10^2

Configure of PDO settings:

1. Set PDO1 transmission type

Master

ID	DLC	Byte1	Index		Sub.	Type	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	18h	02h	FFh	0	0	0

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	02h	0	0	0	0

FFh => transmit after run down of event timer

2. Set PDO1 event timer (1sec)

Master Slave

ID	DLC	Byte1	Index		Sub.	Event timer		Byte7	Byte8
600h + ID	8	22h	00h	18h	05h	E8h	03h	0	0

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	05h	0	0	0	0

03’E8h => 1000_{Dez} => Event timer = 1000ms

3. Disable PDO1 mapping

Master Slave

ID	DLC	Byte1	Index		Sub.	No.	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	1Ah	00h	00h	0	0	0

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	1Ah	00h	0	0	0	0

4. Disable PDO1 output

Master Slave

ID	DLC	Byte1	Index		Sub.	COB-ID		Byte7	acitve
600h + ID	8	22h	00h	18h	01h	180h + ID		0	C0h

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	01h	0	0	0	0

5. Modify PDO1 mapping

Master Slave

ID	DLC	Byte1	Index		No.	Leng.	Sub.	map. Index	
600h + ID	8	22h	00h	1Ah	01h	10h	01h	30h	71h

ID	DLC	Byte1	Index		No.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	1Ah	01h	0	0	0	0

No.: 01h => 1st mapped object in PDO1

Leng.: 10h => 16 bit data length of mapped object

Sub.: 01h => mapped sub-index = 01h

map. Index: 30’71h => mapped index = 71’30h

6. Enable PDO1 mapping

Master Slave

ID	DLC	Byte1	Index		Sub.	No.	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	1Ah	00h	01h	0	0	0

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	1Ah	00h	0	0	0	0

01h => 1 mapped object with PDO1

7. Enable PDO1 mapping

Master Slave

ID	DLC	Byte1	Index		Sub.	COB-ID		Byte7	acitve
600h + ID	8	22h	00h	18h	01h	180h + ID		0	40h

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	01h	0	0	0	0

Measurement cycle:

Sensor to operational mode

Master Slave

ID	DLC	Byte1	Byte2
0	2	01h	ID

Master Slave

ID	DLC	Prozess value	
180h + ID	2	XXh	YYh

after 1s

ID	DLC	Prozess value	
180h + ID	2	XXh	YYh

after 1s

ID	DLC	Prozess value	
180h + ID	2	XXh	YYh

8.3 Change ID (object 2101 or LSS)

Bus in Stop Mode Object 21'01h	Sensor in network with LSS
Not needed	Master / Slave

LSS – Master → Slave:

ID	DLC	Byte1	Byte2
0	2	02h	0

Set sensor to LSS configuration mode

Object 21'01h: Not needed

LSS – Master / Slave:

ID	DLC	Byte1	Vendor ID				Byte6	Byte7	Byte8
7E5h	8	40h	5Fh	00h	00h	00h	0	0	0

ID	DLC	Byte1	Product code				Byte6	Byte7	Byte8
7E5h	8	41h	AAh	BBh	CCh	DDh	0	0	0

ID	DLC	Byte1	Revision number				Byte6	Byte7	Byte8
7E5h	8	42h	EEh	FFh	GGh	HHh	0	0	0

ID	DLC	Byte1	Serial number				Byte6	Byte7	Byte8
7E5h	8	43h	JJh	KKh	LLh	MMh	0	0	0

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4h	8	44h	0	0	0	0	0	0	0

Product code, revision and serial number are displayed on the sensor and can be read out from object 1018h

Change ID

Object 21'01h – Master / Slave:

ID	DLC	Byte1	Index		Sub.	ID	Byte6	Byte7	Byte8
600h + ID	8	22h	01h	21h	00h	ID	0	0	0

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	01h	21h	00h	0	0	0	0

LSS – Master / Slave:

ID	DLC	Byte1	ID	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5h	8	11h	ID	0	0	0	0	0	0

ID	DLC	Byte1	Error	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4h	8	11h	00h	0	0	0	0	0	0

Store changes

Object 21'01h – Master / Slave:

ID	DLC	Byte1	Index		Sub.	„s“	„a“	„v“	„e“
600h + ID	8	22h	25h	61h	01h	73h	61h	76h	65h

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	25h	61h	01h	0	0	0	0

LSS – Master / Slave:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5h	8	17h	0	0	0	0	0	0	0

ID	DLC	Byte1	Error	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4h	8	17h	00h	0	0	0	0	0	0

Leave configuration mode

Object 21'01h: Not needed

LSS – Master / Slave:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5h	8	04h	00h	0	0	0	0	0	0

Reset or restart sensor
=> CAUTION: Use old ID for reset

Object 21'01h – Master / Slave:

ID	DLC	Byte1	Byte2
0	2	81h	ID

LSS – Master / Slave:

ID	DLC	Byte1	Byte2
0	2	81h	ID

Sensor works with new ID

Boot up message with new ID

Object 21'01h – Master / Slave:

ID	DLC	Status
700 + ID	1	00h

LSS – Master / Slave:

ID	DLC	Status
700 + ID	1	00h

8.4 Change baud rate (object 21’00h or LSS)

Bus to stop mode

Object 21’00h

Not needed

Sensor in network with LSS

Master → Slave

ID	DLC	Byte1	Byte2
0	2	02h	0

Set sensor to LSS configuration mode

Object 21’00h

Not needed

Sensor in network with LSS

Master → Slave

ID	DLC	Byte1	Vendor ID				Byte6	Byte7	Byte8
7E5h	8	40h	5Fh	00h	00h	00h	0	0	0

Master → Slave

ID	DLC	Byte1	Product code				Byte6	Byte7	Byte8
7E5h	8	41h	AAh	BBh	CCh	DDh	0	0	0

Master → Slave

ID	DLC	Byte1	Revision number				Byte6	Byte7	Byte8
7E5h	8	42h	EEh	FFh	GGh	HHh	0	0	0

Master → Slave

ID	DLC	Byte1	Serial number				Byte6	Byte7	Byte8
7E5h	8	43h	JJh	KKh	LLh	MMh	0	0	0

Slave → Master

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4h	8	44h	0	0	0	0	0	0	0

Product code, revision and serial number are displayed on the sensor and can be read out from object 1018h

Change baud rate (to 250kBaud)

Object 21’00h

Master → Slave

ID	DLC	Byte1	Index		Sub.	Baud.	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	21h	00h	04h	0	0	0

Slave → Master

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	21h	00h	0	0	0	0

Baud.: 04h => 250kBaud

Sensor in network with LSS

Master → Slave

ID	DLC	Byte1	Baud.	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5h	8	13h	03h	0	0	0	0	0	0

Slave → Master

ID	DLC	Byte1	Error	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4h	8	13h	00h	0	0	0	0	0	0

Baud.: 03h => 250kBaud

Store changes

Object 21’00h

Master → Slave

ID	DLC	Byte1	Index		Sub.	„s“	„a“	„v“	„e“
600h + ID	8	22h	25h	61h	01h	73h	61h	76h	65h

Slave → Master

ID	DLC	Byte1	Index		Sub.	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	25h	61h	01h	0	0	0	0

Sensor in network with LSS

Master → Slave

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5h	8	17h	0	0	0	0	0	0	0

Slave → Master

ID	DLC	Byte1	Error	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4h	8	17h	00h	0	0	0	0	0	0

Leave configuration mode

Object 21’00h

Not needed

Sensor in network with LSS

Master → Slave

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5h	8	04h	00h	0	0	0	0	0	0

Change baud rate of all bus users

Reset or restart bus

=> CAUTION: Start up with new baud rate

Object 21’00h

Master → Slave

ID	DLC	Byte1	Byte2
0	2	81h	00h

Sensor in network with LSS

Master → Slave

ID	DLC	Byte1	Byte2
0	2	81h	00h

Sensor works with new baud rate

Boot up Message with new baud rate

Object 21’00h

Slave → Master

ID	DLC	Status
700 + ID	1	00h

Sensor in network with LSS

Slave → Master

ID	DLC	Status
700 + ID	1	00h

9 Document revision history

V2.10 Manual before software version 2.00 & revision number 30200h

V3.10 Manual from software version 2.00 & revision number 30200h
Changed to the modified CANopen communication
Add changes from software version 2.00
- LSS service
- Dynamic PDO mapping
- Add 2nd process value (temperature)
- Add bootloader

V3.11 Manual from software version 2.00 & revision number 30200h
Add CANopen introduction and an object table