Type 8412

RTD temperature probe with CANopen output

Operating instructions

bürkert
FLUID CONTROL SYSTEMS

V2.00/EN/00743950/2022-04-04

Factory setting

Baud rate: 500 kbaud
see Chapter 5.1 for setting

Node ID:
for 7412: 125
see Chapter 5.2 for setting

Content

Content

1.1 Typographical conventions

Warning signs

Danger

This symbol is used when there may be **danger to personnel** if the instructions are ignored or not followed correctly.

Caution

This symbol is used when there may be **damage to equipment or data** if the instructions are ignored or not followed correctly.

Note signs

Note

This symbol is used when your **special attention** is drawn to a remark.

Reference

This symbol refers to **further information** in other chapters.

abc[1]

Footnote

Footnotes are remarks that **refer to specific points** in the text. Footnotes consist of two parts:

A marker in the text and the footnote text.

The markers in the text are arranged as continuous superscript numbers.

The footnote text (in smaller typeface) is placed at the bottom of the page and starts with a superscript number.

*****

Action

This symbol indicates that an **action to be performed** is described.

The individual steps are marked by this asterisk, for example:

* Connect plug

1 Introduction

1.2 Preface

Please read these Operating Instructions before commissioning the instrument. Keep the manual in a place that is accessible to all users at all times.

Please assist us to improve these operating instructions, where necessary.

All necessary settings are described in this manual. However, if any difficulties should still arise during start-up, you are asked not to carry out any unauthorized manipulations on the unit. You could endanger your rights under the instrument warranty!

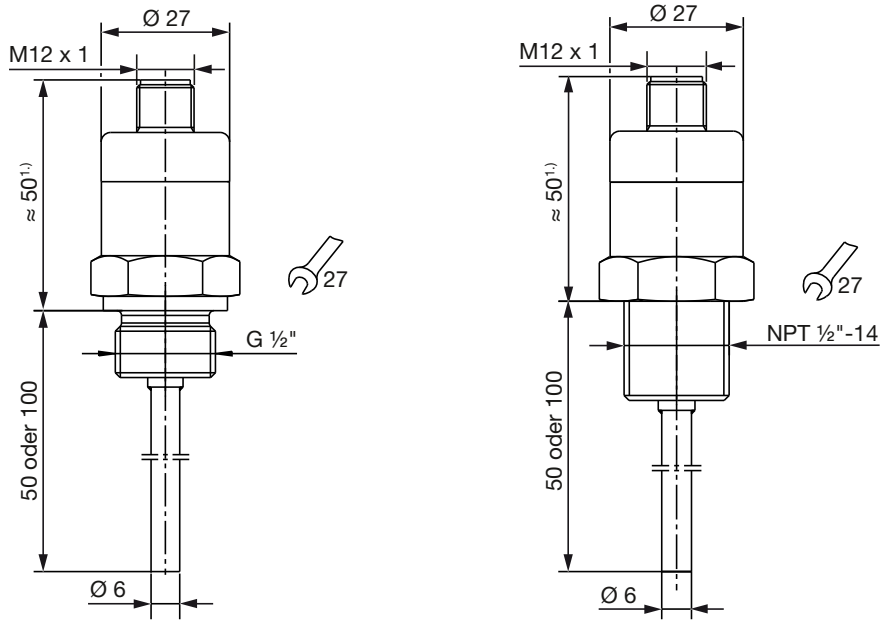
Please contact the nearest subsidiary or the head office in such a case.

1.3 Brief description

RTD temperature probes are the preferred choice for temperature measurement in liquids and gases. An important factor for selecting this installation type is its ability to form a reliable seal in the event of underpressure and overpressure. Areas of application include, among others, medical technology, mechanical engineering, drive technology, commercial vehicles, and railroad systems.

The measuring insert is equipped with a Pt1000 temperature sensor according to DIN EN 60751:2009/IEC 60751:2008, class B, as a standard feature. The measured temperature value is digitalized, linearized, and made available for further processing via the CANopen Interface (CAN slave). A large number of additional useful functions are achieved with the DS 404 device profile. All settings can be made using standard CANopen software tools.

1.4 Dimensions

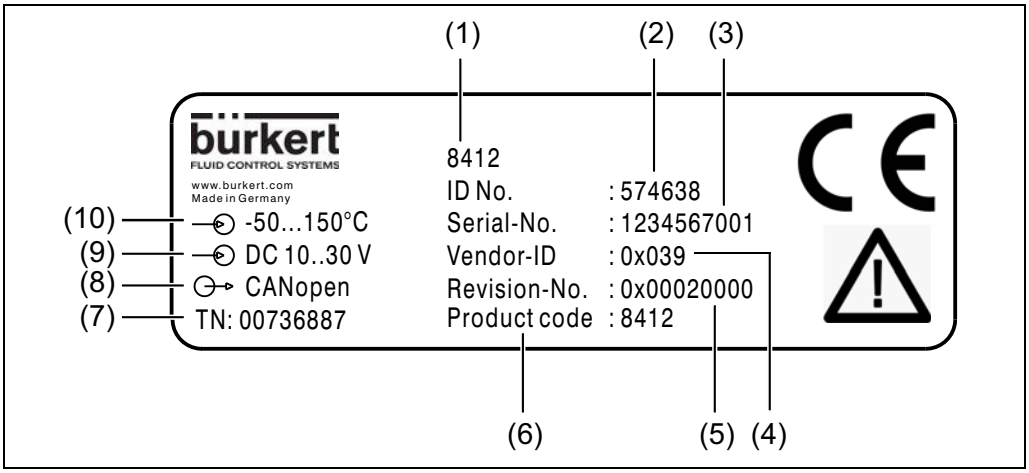


The total height is increased by the height of the socket and cable used.

2.1 Nameplate

Position

The nameplate is located at the housing surface.



(1)	Device type no.	(2)	Device ID no.
(3)	Device serial no.	(4)	Manufacturer ID no. for CANopen devices
(5)	Device revision no.	(6)	Standardized product designation
(7)	TN	(8)	Digital interface
(9)	Voltage supply, for more in-depth information, see „Technical Date“	(10)	Input

Device ID no.

The device ID number uniquely identifies an article and, together with the device type no., determines the selected device variant.

TN

Internal no.

Device type no.

The device type no. can help to localize the associated device description file (EDS) as part of the file name.

Load EDS:

1. Go to web page https://country.burkert.com/
2. Select your country
3. Click on Continue to website
4. Confirm or change cookie settings
5. Enter the device type number, e.g. 8412 (see device nameplate) in the search field
6. Click on the first result of the search
7. In the area Software download the ZIP file DeviceDescription
8. Unpack the ZIP file
9. Identify and select the required EDS file by device type no.

The EDS file is now available for use with the CANopen configuration tool to

2 Identifying the device version

The EDS file is now available for use with the CANopen configuration tool to configure and verify the device. This can be used to configure and check the device.

Date of manufacture

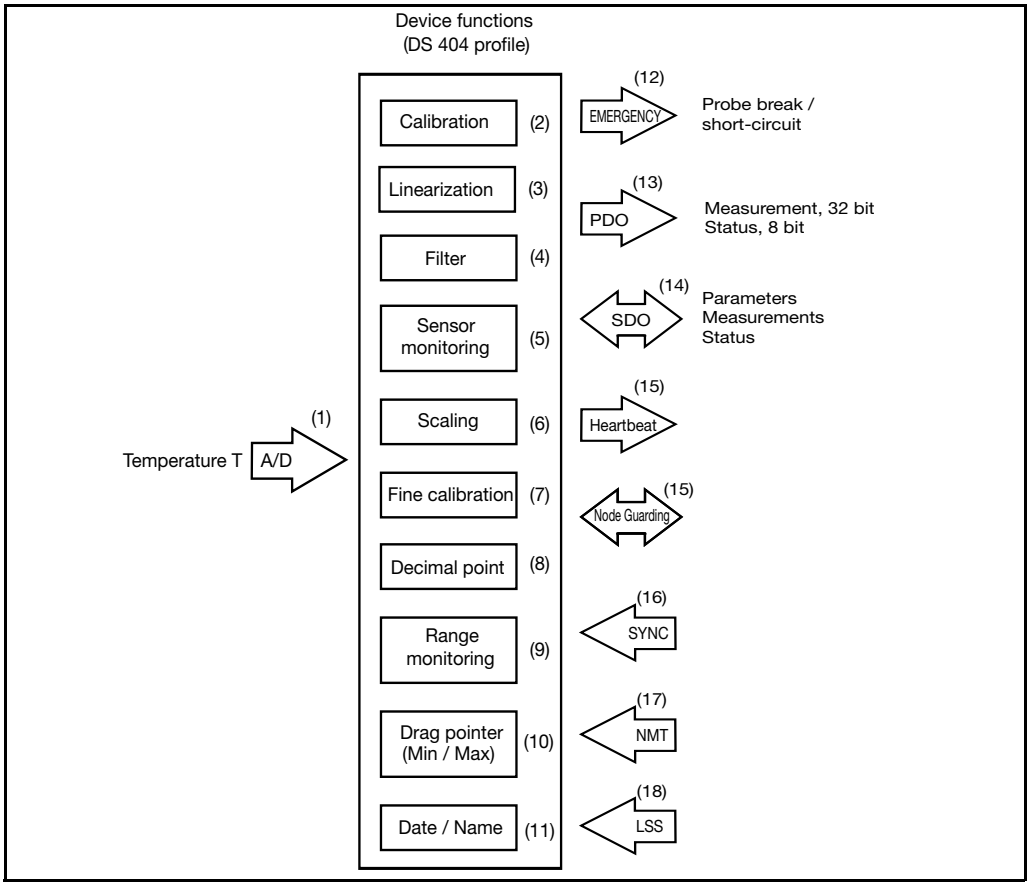
The device's date of manufacture (year and calendar week) is part of the fabrication number. Digits 12 to 15 denote the year of manufacture and the calendar week.

3.1 Application

Transmitters are used for acquiring pressures or temperatures in liquid or gaseous media.

The measurements from the pressure or temperature sensors are digitized and made available for further processing via the CANopen. Several useful extra functions are implemented through the DS 404 device profile. All settings can be made using standard CANopen software tools.

3.2 Block diagram



3.2.1 Operation

(1) The analog signal from the pressure cell or the temperature sensor is digitized.

(2) The pressure or temperature signal is digitally calibrated at the factory.

(3) The temperature signal is linearized.

(4) Undesirable signal fluctuations can be suppressed through the (adjustable) filter constant.

(5) The sensor monitoring facility continuously checks the correct performance of the sensor signal and triggers high-priority emergency telegrams in the event of an error.

(6) The measurement can be scaled to any dimensional unit (or in % of range).

(7) Fine calibration features an autozeroing function (with pressure sensors only) and a freely adjustable shift of the characteristic (offset).

(8) The measurements are output with a freely selectable decimal place.

(9) Range monitoring features freely selectable upper and lower limits. The result is output as a status byte together with the measurement value in the PDO telegram.

(10) The drag pointer function stores the minimum and maximum measured value.

(11) Date and name of the last servicing action can be stored.

(12) An emergency telegram is triggered in the event of a sensor fault.

(13) The PDO telegram contains the 32-bit measurement and the 8-bit status. The measurement that is output can be controlled by means of different trigger conditions.

(14) Parameters can be set through SDO telegrams, and measurements and status can be requested.

(15) The heartbeat signal or Node Guarding[1] can be used to additionally monitor the transmitter functions.

(16) Measurement transmission can additionally be controlled by using the Sync command.

(17) The NMT telegrams serve to control the operating status of the transmitter.

(18) The CAN node ID and CAN baud rate is set via LSS or SDO, according to choice.

[1] Node Guarding is only available for transmitters with sensor.

3.3 Setup program

All instrument parameters, see Chapter 8 “Object dictionary”, page 27, can be accessed via the CANopen object dictionary (EDS file) and can be set using standard CANopen software tools. An appropriate EDS file is available for all device types. The file is downloadable free of charge from the Bürkert home page www.burkert.com using the product typ number 8412 in the search field.

4 Installation

4.1 Electrical connection

Earth the instrument at the pressure connection.
The bus ends must be provided with a line termination.
See Chapter 4 “Installation” / “Line termination”, page 13.

Bus cable

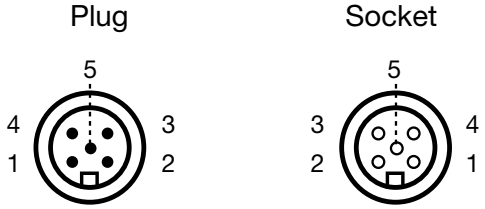
- the bus specifications to ISO 11 898 must be observed
- cable diameter 6 to 12 mm
- conductor cross-section up to 1.5 mm^2 per core
- signal cables must be routed separately from cables with voltages above 60 V
- use cables with twisted cores
- avoid the vicinity of electrical installations, or use screened cables

Connection

Connection		Terminal assignment
		M12 connector
Supply voltage DC 10 to 30 V	CAN_V+ CAN_GND	2 3
CANopen	screen CAN_H CAN_L	1 4 5

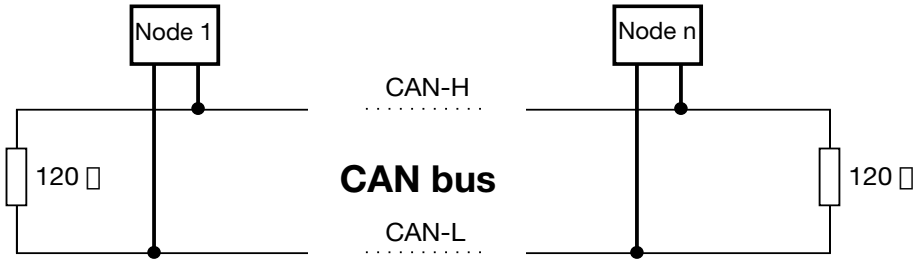
Circular connector

M12 x1; 5-pole to IEC 60 947-5-2



Line termination

The CAN bus has a linear topology. Each end of the bus must be terminated with a 120 Ω resistor, to avoid signal reflections and, as a consequence, transmission problems.



5 Commissioning

5.1 Setting the CAN baud rate

General

The baud rate is set to 500 kbaud ex-factory.

The CAN baud rate can be set both via SDO telegrams (object dictionary) and LSS.

Setting via SDO

The CAN baud rate can be reprogrammed via the CANopen object dictionary, index 0x2001.
This setting will only be accepted as the new CAN baud rate after resetting the transmitter.

CAN baud rate [kbaud]	Max. bus length [m]	Entry in object dictionary 0x2001
1000	25	0
800	100	1
500	100	2
250	250	3
200	250	99
125	500	4
100	500	98
50	1000	6
20	2500	7

Setting via LSS

The transmitters support the LSS standard (Layer Setting Services) as per DSP-305, V1.1.
This can be used to set the baud rate and node ID for the entire plant in a standardized manner.
The LSS address consists of four elements, which are indicated on the nameplate: Vendor-ID, Product code, Revision-No., Serial-No.
The latest setup tools from different manufacturers can also be used to operate this function.
As an alternative, the baud rate and node ID are also settable via SDO (see above).

5.2 Setting the node ID

General

Ex-factory, the node ID is preset as follows:

for 8412: 125

The node ID can be set both via SDO telegrams (object dictionary) and LSS.

Each node ID may only be allocated once on the bus.

Setting via SDO

The node ID can be reprogrammed via the CANopen object dictionary, index 0x2000, thereby enabling all transmitters of a plant, for instance, to be programmed to new node IDs from a central CAN terminal.

This setting will only be accepted after resetting the transmitter.

Setting via LSS

The transmitters support the LSS standard (Layer Setting Services) as per DSP-305, V1.1.
This can be used to set the baud rate and node ID for the entire plant in a standardized manner.
The LSS address consists of four elements, which are indicated on the nameplate: Vendor-ID, Product code, Revision-No., Serial-No.
The latest setup tools from different manufacturers can also be used to operate this function.
As an alternative, the baud rate and node ID are also settable via SDO (see above).

6 CANopen function

6.1 Overview of communication functions

Communication profile

The CAN interface communication functions correspond to the CANopen communication profile DS-301.

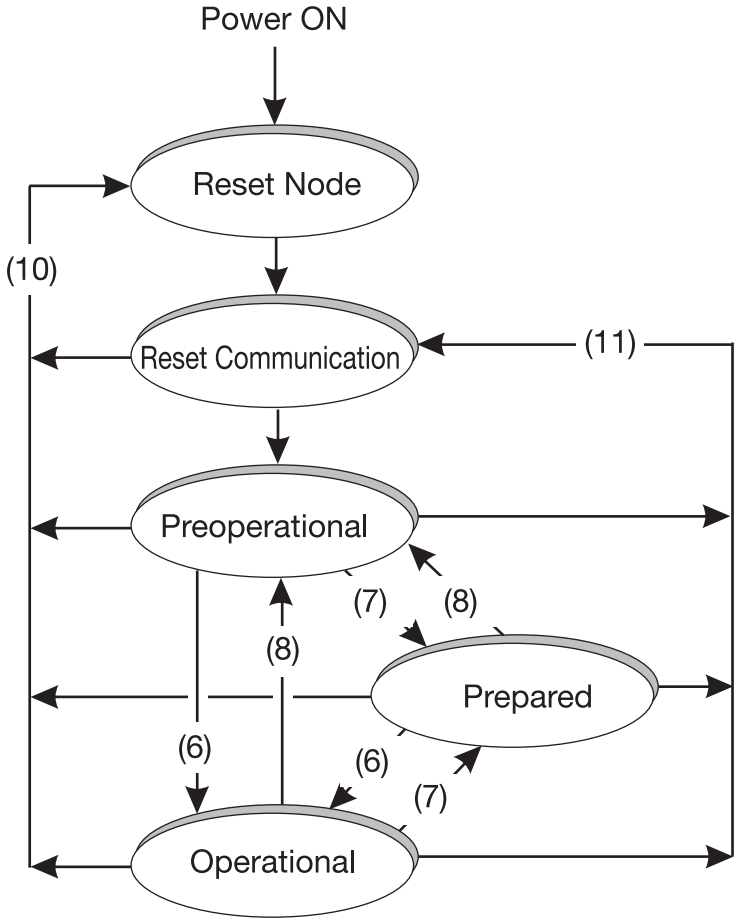
Objects

Data exchange with CANopen devices takes place in the form of objects.
The table below contains the supported objects; these will be explained in the sections that follow.

Object	CAN identifier	Function	Note
NMT	0	network management	bus master is the sender
SYNC	0x80	PDO synchronization	bus master is the sender
EMERGENCY	0x80 + node ID	alarm message	
TPDO 1	0x180 + node ID	measurement 1 and status	identifier changeable via object dictionary 0x1800,1
TPDO 5	inactive	measurement 2 and status	identifier changeable via object dictionary 0x1804,1 only with twin probe
SDO (tx)	0x580 + node ID	access to parameters (object dictionary)	slave (8412) to master
SDO (rx)	0x600 + node ID	access to parameters (object dictionary)	master to slave (8412)
Heartbeat	0x700 + node ID	device monitoring	cyclic “sign of life”
Bootup	0x700 + node ID	device monitoring	once, after power ON
LSS(tx)	0x7E4 = 2020	setting of baud rate, or node ID	slave (8412) to master
LSS(rx)	0x7E5 = 2021	setting of baud rate, or node ID	master to slave (8412)

6.2 NMT

The transmitters support both the CANopen minimum bootup and the auto-operational bootup.



NMT user data

Network management command	Network management of object data	
	Byte 1 Command specifier	Byte 2 Node ID
Node start (6)	0x01	0 —127 (0 = all devices)
Node stop (7)	0x02	
Enter preoperational state (8)	0x80	
Reset node (10)	0x81	
Reset communication (11)	0x82	

Settings for NMT

Boot mode	**State after power ON**	**Setting of object 0x1F80**
Minimum bootup	Preoperational	0xC
Auto-operational bootup	Operational	0x8[1]

[1] Factory setting

6 CANopen function

6.3 Sync

The PDOs of the transmitter can be configured as “synchronous”. When a Sync object has been received, the corresponding PDO will be transmitted.

Settings for Sync

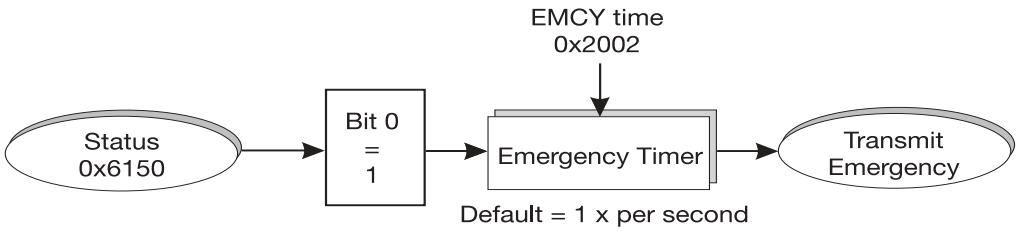
The PDO transmission type can be switched between synchronous (controlled by the master) and asynchronous (event-controlled) in the object dictionary (0x1800,2 or 0x1804,2).

Factory setting: event-controlled (=0xFF)

Transmission type	**Setting the object 0x1800,2 (for PDO1) 0x1804,2 (for PDO5)**
asynchronous	0xFF
synchronous	0x01

6.4 Emergency

In the event of a sensor short-circuit or sensor break, the transmitters will send a high-priority emergency object (EMCY).
In this case the telegram is repeated cyclically. The cycle time can be set.



EMCY user data (8 bytes)

Sensor break

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5030 h (hardware) 2 bytes		00000001 1 byte	1 or 2 (chann.) 1 byte	00000001 1 byte	not used		

Sensor short-circuit

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5030 h (hardware) 2 bytes		00000001 1 byte	1 or 2 (chann.) 1 byte	00000010 1 byte	not used		

Reset error

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0000 h (hardware) 2 bytes		00000000 1 byte	1 or 2 (chann.) 1 byte	xxxxxxxx 1 byte	not used		

Setting for Emergency

Factory setting: once per second (= 1000 msec)

EMCY time	Setting the object 0x2002
milliseconds	0 — 65535 (0 = not repeated)

6.5 PDO

1 or 2 transmit PDO(s) (process data object) are available for the measurements.

The setting for mapping (0x1A00) of the PDO user data is fixed to 0x9130 (measured value in INT32 format, up to revision no. 0x10000) or 0x6130 (measured value in float format, from revision no. 0x20000) and to 0x6150 (status byte).
The calculation of these values is described in chapter 7 “Device function”, page 25.

PDO user data (5 bytes)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
4 Byte 0x9130 = Measurement value INT32 (up to Rev.-No. 0x10000) 0x6130 = Measurement value Float (from to Rev.-No. 0x20000)				0x6150 1 byte status bit 2, 1, 0	not used not included in transmission		

Status bit 0 = Sensor faulty (sensor monitoring)

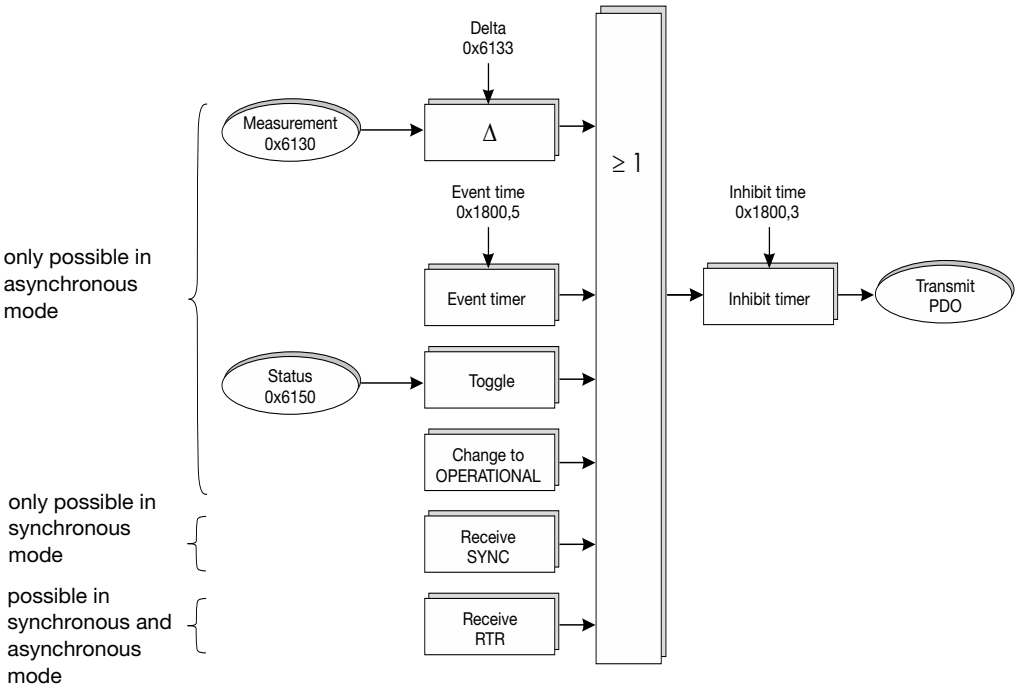
If bit 0 is set, the measurement transmitted in the PDO is invalid!

Status bit 1 = overrange (measurement range monitoring)
Status bit 2 = underrange (measurement range monitoring)

6 CANopen function

PDO output control

The graphic below shows possible events that will lead to a PDO telegram being transmitted. The setting options are described below. For calculating the measurement and status, chapter 7 “Device function”, page 25.



Mode chapter 6.3 “Sync”, page 18

Settings for PDO output

Delta:
If a measurement modification exceeds the set value, a PDO is sent.
Factory setting: 1

Delta	**Setting the object** **0x6133,1 (for PDO 1)** **0x6133,2 (for PDO 5)**
Float value	(0 = inactive)

Event time (cyclic transmission):
When the set event time has elapsed, a PDO is sent.
Factory setting: once per second (= 1000 msec).

Event time	**Setting the object** **0x1800,5 (for PDO 1)** **0x1804,5 (for PDO 5)**
milliseconds	0 — 65535 (0 = inactive)

Toggle:
A PDO is sent for each change of the measurement status.

Operational:
When changing to the “Operational” status, a PDO is sent once.

Sync:
If the transmission type has been configured as “synchronous”, a PDO is sent on receipt of the Sync object.
Description chapter 6.3 “Sync”, page 18.

RTR (Remote Transmission Request):
If requested by a PDO recipient, a PDO is sent.

Inhibit time

The transmission of a PDO is suppressed before the set inhibit time has elapsed. This will reduce the load on the bus and prevent it from being overloaded.
Factory setting: 0 (= inactive)

Inhibit time	**Setting the object 0x1800,3 (for PDO 1) 0x1804,3 (for PDO 5)**
0.1 milliseconds	0 — 65535 in 1/10 msec (0 = inactive) Example: 1000 = 100 msec

6.6 SDO

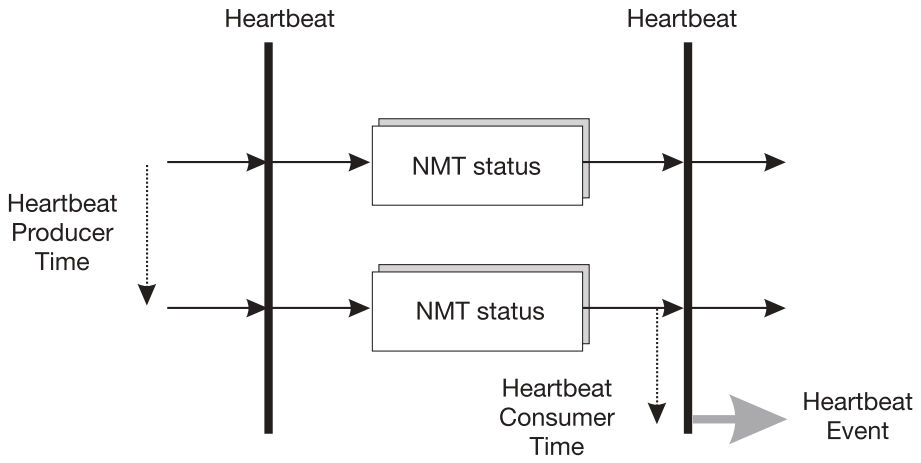
The service data object (SDO) is used for accessing the object dictionary (transmitter parameters). Using the SDO, it is possible to gain read or write access to the object dictionary.

For a description of all objects: chapter 8 “Object dictionary”, page 27.

6 CANopen function

6.7 Heartbeat

The Heartbeat object signals the presence of a transmitter, thereby ensuring system reliability. It provides a simple alternative to the Node Guarding protocol (chapter 6.8 “Node Guarding”, page 23).



Heartbeat user data

The heartbeat message (heartbeat event) consists of one byte. In this byte, the NMT status of the internal status machine is coded as follows:

Bootup:	0
Stopped:	4
Operational:	5
Preoperational:	127

Settings for heartbeat

The configuration as a heartbeat sender takes place via the heartbeat producer time in the object dictionary (0x1017).

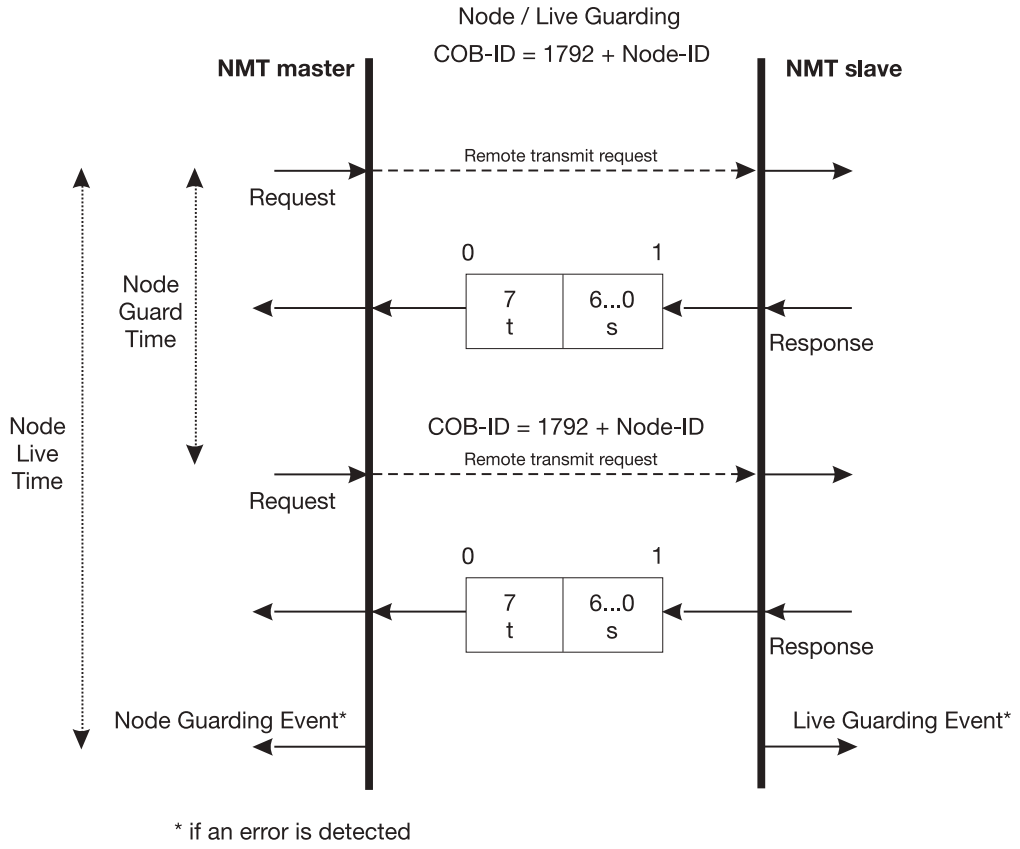
The Heartbeat and Node Guarding functions can only be activated one at a time, but never simultaneously.

Factory setting: HeartbeatTime = 500 msec

Heartbeat timer	**Setting the object 0x1017**
milliseconds	0 — 65535 0 = inactive

6.8 Node Guarding

The Node Guarding object provides an alternative to the Heartbeat object (chapter 6.7 “Heartbeat”, page 22).
It indicates the presence of a transmitter, thereby ensuring system reliability. Unlike Heartbeat, in the case of Node Guarding the NMT master (usually a PLC) sends a request, which is answered by the NMT slave (here: the CANtrans transmitter).
The structure of the Node Guarding response is similar to that of the Heartbeat protocol. The only difference is that it contains an additional toggle bit that changes between 0 and 1 for consecutive messages.



* if an error is detected

s: status of the NMT slave
- 4: STOPPED
- 5: OPERATIONAL
- 127: PREOPERATIONAL

t: toggle bit

Node Guarding user data

The Node Guarding message contains one byte, consisting of the toggle bit t and the NMT status s, which is coded as follows:

Bootup:	0
Stopped:	4
Operational:	5
Preoperational:	127

Settings for Node Guarding

The settings for the Node Guarding slave are made in the object directory, via the parameters Guard Time (0x100C) and Live Time Factor (0x100D).

The Node Guarding slave calculates its own live time to be the product of these two parameters. If the transmitter does not receive a Node Guarding request within the live time, the Live Time Guarding Event is initiated and the transmitter adopts the "Preoperational" state.

If Guard Time **or** Live Time Factor has the value 0, then Live Time = 0 and no Live Guarding Event is initiated. The NMT slave, however, will still answer any NMT request by the NMT master.

If Guard Time **and** Live Time Factor have the value 0 (factory setting), Node Guarding is not active.

The Node Guarding and Heartbeat functions can only be activated one at a time, but never simultaneously.

Guard Time	**Setting the object 0x100C**
milliseconds	0 — 65535 (0 = inactive)

Live Time Factor	**Setting the object 0x100D**
Factor	0 — 255 (0 = inactive)

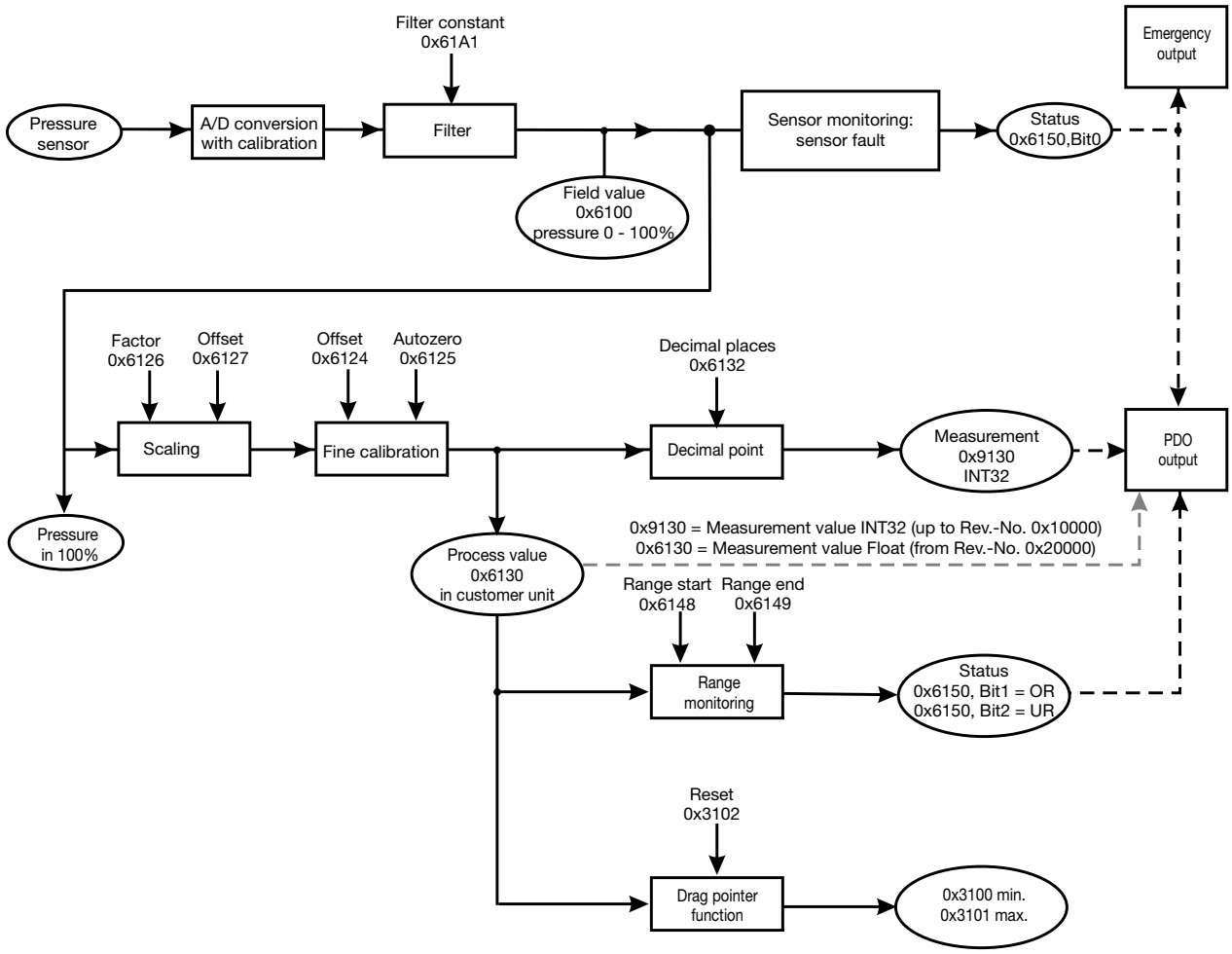
6.9 LSS

The transmitters support the LSS standard (Layer Setting Services) as per DSP-305, V1.1.
This can be used to set the baud rate and node ID for the entire plant in a standardized manner.
The LSS address consists of four elements, which are indicated on the nameplate: Vendor-ID, Product code, Revision-No., Serial-No.
The latest setup tools from different manufacturers can also be used to operate this function.
As an alternative, the baud rate and node ID are also settable via objects in the object dictionary.
See chapter 5.1 “Setting the CAN baud rate”, page 14;
See chapter 5.2 “Setting the node ID”, page 15.

7.1 Device profile

The transmitters operate according to the CANopen device profile DS-404 “Measuring Devices and Closed-Loop Controllers”. The graphics below show the signal flow of the measurement through the transmitter functions. Some functions can be set by the user.
The setting options are described in
⇨ Chapter 8 “Object dictionary”, page 27.

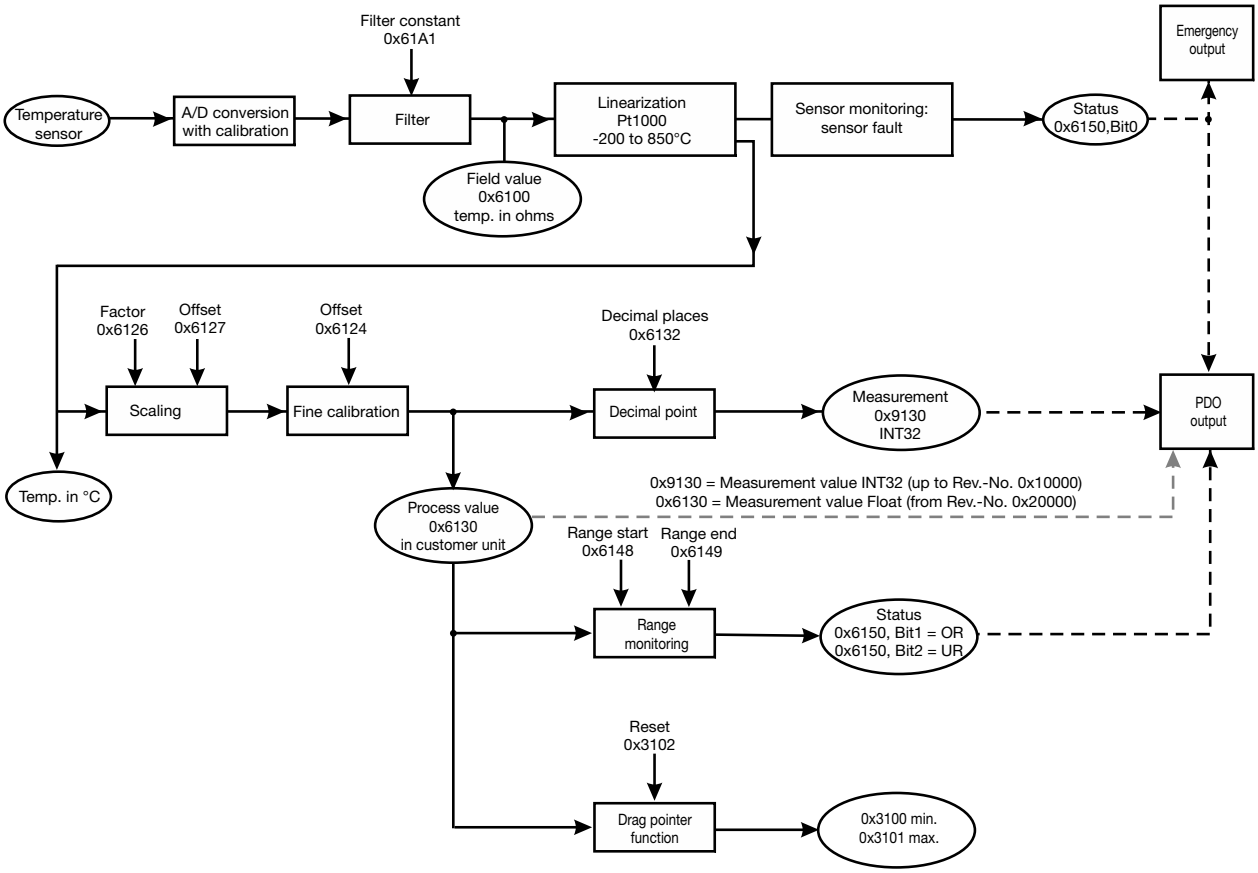
7.2 Data flow: pressure channel



The calculation for the pressure channel is processed every 1.0 msec.

7 Device function

7.3 Data flow: temperature channel



The calculation of the temperature channel is processed every 250 msec.

8.1 Overview

The entire object dictionary is available as an EDS file, thereby enabling all CANopen-compatible configuration programs to be used for installation and parameterization. For this reason, a setup program for these devices is not supplied.

The most important setting parameters are summerized below, together with their possible values.

All objects can be read, or written, with SDO telegrams. This object dictionary is valid for all transmitter variants. Depending on the device, some objects have 1 or 2 subindices. Accordingly, the 8412, for example, only has the sub-index 1 = temperature channel.

For all device types, the corresponding EDS file is downloadable free of charge from the Bürkert home page www.burkert.com.

Index	Sub-index	Format	Access	Name	Description	Value
0x1017	-	UINT16	RW	Heartbeat Producer Time	Time for cyclic transmission of a “sign of life”	0 — 65535 msec 0 = inactive ex-factory: 500
0x100C	-	UINT16	RW	Guard Time	Time factor for monitoring Node Guarding	0 — 65535 msec 0 = inactive ex-factory: 0
0x100D	-	UINT8	RW	Live Time Factor	Multiplier for monitoring Node Guarding	0 — 255 0 = inactive ex-factory: 0
0x1800	-			PDO 1 Communication Parameter	Controls the transmission conditions for 1st PDO	
	0x01	UINT32	RW[a]	COB-ID	ID with which the PDO is transmitted	0x180 — 0x57F bit 0x800000000 set = PDO inactive ex-factory: 0x180+Node-ID
	0x02	UINT8	RW[a]	Transmission Type	Transmission mode	0x01 = synchronous 0xFF = event-controlled ex-factory: 0xFF
	0x03	UINT16	RW[a]	Inhibit Time	Do not transmit before time has elapsed	0 — 65535 (x 0.1 msec) ex-factory: 0 = inactive
	0x05	UINT16	RW[a]	Event Time	Time for cyclic transmission	0 — 65535 msec 0 = inactive ex-factory: 1000 msec

8 Object dictionary

Index	Sub-index	Format	Access	Name	Description	Value
0x1804	-			PDO 5 Communication Parameter	Controls the transmission conditions for 2nd PDO (on devices with 2 sensors)	
0x1F80	-	UINT32	RW	NMT Startup	Bootmode, see Chapter 6.2 “NMT”, page 17	0xC "Preoperational" 0x8 "Operational" ex-factory: 0x8
0x2000	-	UINT8	RW[a]	Node-ID	Setting the node address via SDO (also possible via LSS)	1 — 127 ex-factory: 123 (PT) ex-factory: 124 (P) ex-factory: 125 (T) ex-factory: 126 (TT)
0x2001	-	UINT8	RW[a]	Baud rate	Setting the baud rate via SDO (also possible via LSS)	0 = 1 Mbaud 1 = 800 kbaud 2 = 500 kbaud 3 = 250 kbaud 99 = 200 kbaud 4 = 125 kbaud 98 = 100 kbaud 6 = 50 kbaud 7 = 20 kbaud ex-factory: 2
0x2002	-	UINT16	RW[a]	EMCY_Time	Time for cyclic transmission of error messages	0 — 65535 msec 0 = once ex-fact. 1000 msec
0x3100	0x01	float	RO	AI PV Min 1	Drag pointer minimum value	
	0x02	float	RO	AI PV Min 2	as subindex 0x01, for devices with 2 sensors	
0x3101	0x01	float	RO	AI PV Max 1	Drag pointer maximum value	
	0x02	float	RO	AI PV Max 2	as subindex 0x01, for devices with 2 sensors	
0x3102	0x01	UINT32	WO	AI Reset Min-Max 1	Reset drag pointers 0x3100 and 0x3101	Reset with "roeb" = 0x62656F72
	0x02	UINT32	WO	AI Reset Min-Max 2	as subindex 0x01, for devices with 2 sensors	
0x3400	-	String (4)	RW	AI Customer Date	any text, 4 bytes, e.g. date	ex-factory: "0003"

Index	Sub-index	Format	Access	Name	Description	Value
0x3401	-	String (4)	RW	AI Customer Name	any text, 4 bytes, e.g. name	ex-factory: "8412"
0x6124	0x01	float	RW	AI Offset 1	Customer fine calibration	ex-factory: 0
	0x02	float	RW	AI Offset 2	as subindex 0x01, for devices with 2 sensors	
0x6125	0x01	UINT32	WO	AI Autozero	With pressure sensors only: show current pressure as zero, alters object 0x6124,1	Set to zero with “zero” = 0x6F72657A
0x6126	0x01	float	RW	AI Scaling Factor 1	Scaling of factor	ex-factory: 1[b] e.g. 0.1, to show pressure not as 0 — 100% but as 0 — 10 bar; or e.g. 1.8, to show temperature not in °C but in °F.
	0x02	float	RW	AI Scaling Factor 2	as subindex 0x01, for devices with 2 sensors	
0x6127	0x01	float	RW	AI Scaling Offset 1	Scaling offset	ex-factory: 0[b] (up to Rev.-No. 0x10000) ex-factory: 273.15 (from Rev.-No. 0x20000), e.g. 0.0, to show pressure not as 0 — 100% but as 0 —10 bar; or e.g. 32, to show temperature not in °C but in °F.
	0x02	float	RW	AI Scaling Offset 2	as subindex 0x01, for devices with 2 sensors	
0x6130	0x01	float	RO	AI Input PV float 1	Process value as float (for readout via SDO; as also in PDO from Rev.-No. 0x200000)	
	0x02	float	RO	AI Input PV float 2	as subindex 0x01, for devices with 2 sensors	

8 Object dictionary

Index	Sub-index	Format	Access	Name	Description	Value
0x6132	0x01	UINT8	RW	AI Decimal Digits 1	Decimal places for fixed-point representation as INT 32 as in PDO	0 — 3 ex-factory: 1 Example, pressure: **0** => 0 — 100 = 0 — 100% **1** => 0 — 1000 = 0 — 100.0% **2** => 0 — 10000 = 0 — 100.00% Example, temp.: **0** => 19 = 19°C **1** => 197 = 19.7°C **2** => 1973 = 19.73°C
	0x02	UINT8	RW	AI Decimal Digits 2	as subindex 0x01, for devices with 2 sensors	
0x6133	0x01	float	RW	AI Interrupt Delta Input PV1	Delta value for event-controlled PDO transmission	ex-factory: 1.0[b] (0 = inactive)
	0x02	float	RW	AI Interrupt Delta Input PV2	as subindex 0x01, for devices with 2 sensors	
0x6148	0x01	float	RW	AI Span Begin 1	Start of range monitoring	ex-factory: 0 (P sensor)[b] ex-factory: -50 (T-Sensor up to Rev.-No. 0x10000) ex-factory: 223.15 (T-Sensor from Rev.-No. 0x20000)
	0x02	float	RW	AI Span Begin 2	as subindex 0x01, for devices with 2 sensors	
0x6149	0x01	float	RW	AI Span End 1	End of range monitoring	ex-factory: 100 (P sensor)[b] ex-factory: 450 (T sensor up to Rev.-No. 0x10000) ex-factory: 723.15 (T sensor from Rev.-No. 0x20000
	0x02	float	RW	AI Span End 2	as subindex 0x01, for devices with 2 sensors	

Index	Sub-index	Format	Access	Name	Description	Value
0x6150	0x01	UINT8	RO	AI State 1	Error status (as also in PDO) bit 0 = sensor faulty bit 1 = overrange (value > object 0x6149) bit 2 = underrange (value < object 0x6148)	
	0x02	UINT8	RO	AI State 2	as subindex 0x01, for devices with 2 sensors	
0x61A1	0x01	UINT8	RW	AI Filter Constant 1	Filter constant of floating average-value filter	1...255 ex-factory: 0 = inactive
	0x02	UINT8	RW	AI Filter Constant 2	as subindex 0x01, for devices with 2 sensors	
0x9130	0x01	INT32	RO	AI PV32Bit1	Process value as Int32 (as also in PDO up to Rev.-No. 0x10000)	
	0x02	INT32	RO	AI PV32Bit2	as subindex 0x01, for devices with 2 sensors	

[a] The parameter alteration becomes only effective after a hardware reset, after the NMT command “Reset Communication” or after “Reset Node”, see Chapter 6.2 “NMT”, page 17!

[b] As of Rev.-No. 0x20000 is factory-scaled to Kelvin unit according to the device measuring range.

9 Programming examples

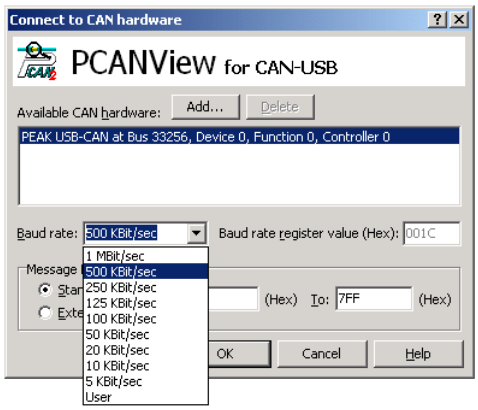
9.1 General

You can compile simple CAN messages yourself and transmit them to the individual CAN devices by using the free PCANView program (supplied by Peak, www.peak-system.com).

9.2 Function

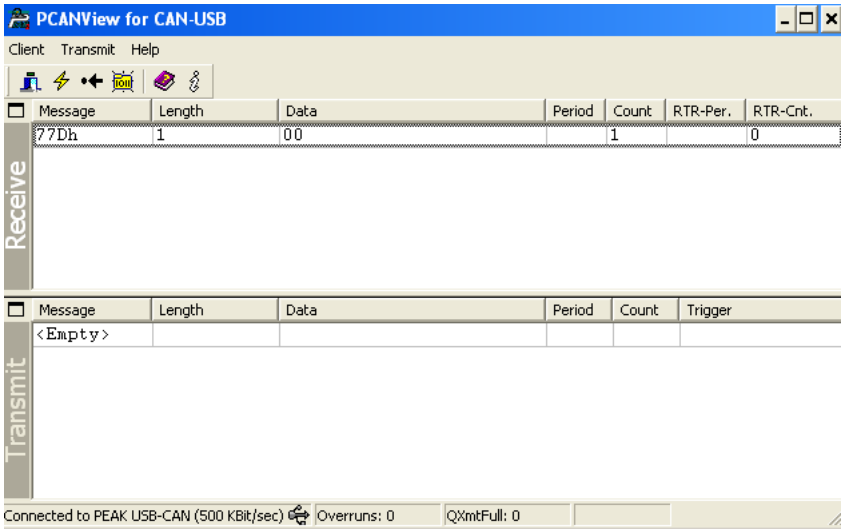
To start with, you will be asked to select the baud rate. It can be set by choosing one of the values displayed in the program window.

The default setting for the transmitters as delivered is 500kbit/sec.

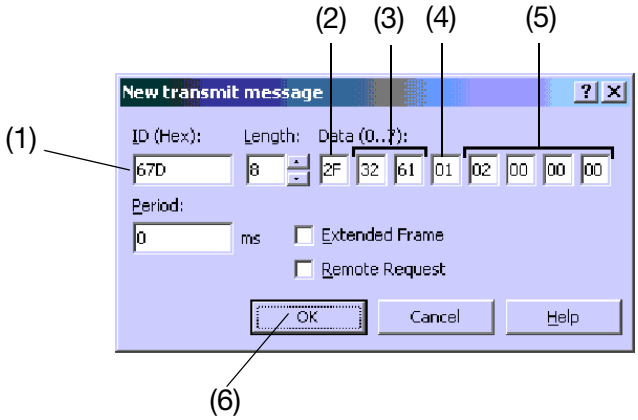


9.3 Testing the connection

After switching on the transmitter (power-on), you will see a message (bootup message) in the *Receive* field, which is transmitted for test purposes by all CANopen devices after switch-on.



The program then gives you the option of entering CAN messages via the *Transmit* folder, in the sub-item *New transmit message*. The following window appears:



For an overview of the communication functions see Chapter 6.1 “Overview of communication functions”, page 17.

The **ID (Hex)** (1) determines the telegram type (PDO, SDO or LSS), the address and the priority of the message. The lowest ID has the highest priority in the case of CAN telegrams.

The fields **Data (0..7)** contain the user data of the CAN telegram in hexadecimal format. Please note the following arrangement:

The data field (2) contains the control byte. Here you can define whether the CAN device should be read out or written to. At the same time, you can also define the type of value here. The following parameters are possible:

Read:	0x40
Write an 8-bit value:	0x2F
Write a 16-bit value:	0x2B
Write a 32-bit value:	0x22

The next two bytes (3) specify the object index (Chapter 7), whereby it is absolutely essential to write the Low byte first and then the High byte. The object index 0x6132 has been entered in the screenshot above, by way of example.

The byte (4) specifies the 8-bit subindex, which can also be taken from the table in Chapter 7. The value 00 is entered here for objects without a subindex.

The last 4 bytes (5) contain object values that are read or written. As a rule, the Low byte must also be entered first here. The byte fields that are not required are filled with the value 00. Some examples will be given below.

The data telegram created in this way is transmitted to the CAN device by clicking **OK** (6).

The transmitted CAN message is logged in the “Transmit” field and listed.
The CAN response of the transmitter is logged in the “Transmit” field and listed.

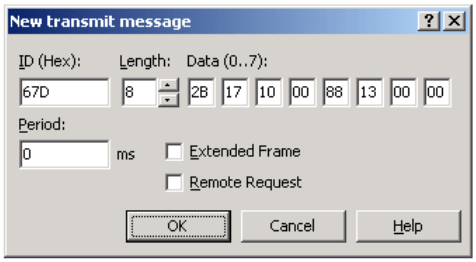
Wrong entries may result in uncontrollable behavior!

9 Programming examples

9.4 Heartbeat Producer Time

(see Chapter 6.7 “Heartbeat”, page 22)

Alteration of the time for the cyclic transmission of a sign of life at 5000msec intervals (1388hex)



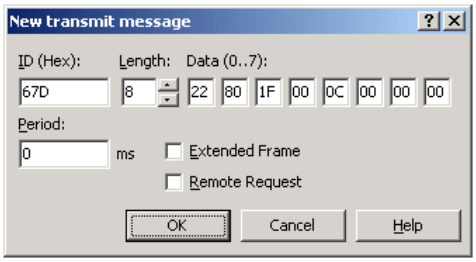
Node ID: 125_{dec}
COP ID: $67D_{hex}$
Object index: 1017_{hex}
Subindex: 00_{hex}
Value: 1388_{hex}

9.5 Boot mode “Minimum bootup”

(see Chapter 6.2 “NMT”, page 17)

After switch-on, the transmitter should adopt the preoperational state.

A change of boot mode only becomes effective after a reset !

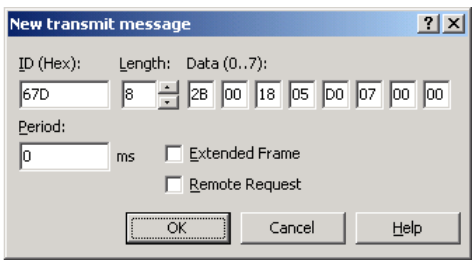


Node ID: 125_{dec}
COP ID: $67D_{hex}$
Object index: $1F80_{hex}$
Subindex: 00_{hex}
Value: $0C_{hex}$

9.6 Event time

(see Chapter 6.5 “PDO”, page 19)

Set the time for cyclic measurement transmission to 2000msec (7D0hex)



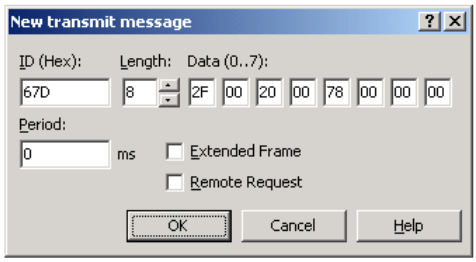
Node ID: 125_{dec}
COP ID: $67D_{hex}$
Object index: 1800_{hex}
Subindex: 05_{hex}
Value: $7D0_{hex}$

9.7 Setting the node ID

(see Chapter 5.2 “Setting the node ID”, page 15)

Set node address to the value 120 (78hex) via SDO

A change of the node ID only becomes effective after a reset !



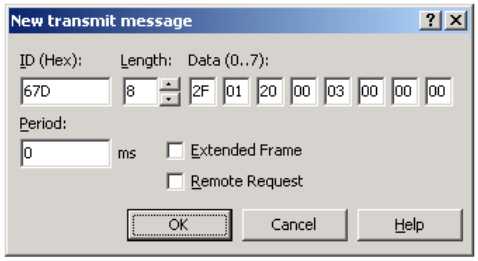
Node ID:	125_{dec}
COP ID:	$67D_{hex}$
Object index:	2000_{hex}
Subindex:	00_{hex}
Value:	78_{hex}

9.8 Setting the baud rate

(see Chapter 5.1 “Setting the CAN baud rate”, page 14)

Set the baud rate to the value 3 = 250kbaud (03hex) via SDO.

A change of the baud rate only becomes effective after a reset !

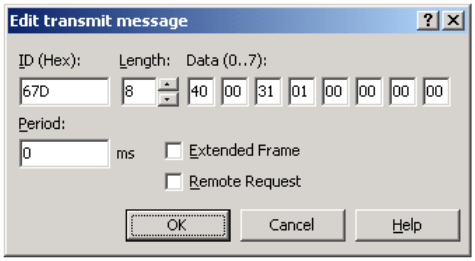


Node ID:	125_{dec}
COP ID:	$67D_{hex}$
Object index:	2001_{hex}
Subindex:	00_{hex}
Value:	03_{hex}

9.9 Reading out the minimum value

(see Chapter 7.2 “Data flow: pressure channel”, page 25)

Readout of the smallest value that was registered.



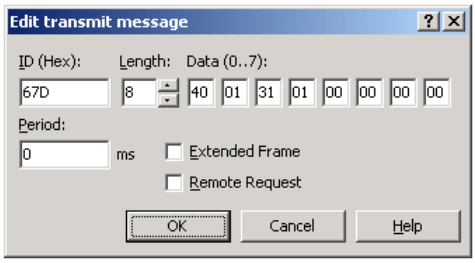
Node ID:	125_{dec}
COP ID:	$67D_{hex}$
Object index:	3100_{hex}
Subindex:	01_{hex}
Value:	*Read procedure*

9 Programming examples

9.10 Reading out the maximum value

(see Chapter 7.2 “Data flow: pressure channel”, page 25)

Readout of the largest value that was registered.

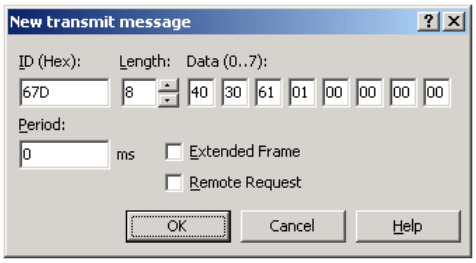


Node ID:	125_{dec}
COP ID:	$67D_{hex}$
Object index:	3101_{hex}
Subindex:	01_{hex}
Value:	*Read procedure*

9.11 Reading out the measurement in “Float” format

(see Chapter 7.2 “Data flow: pressure channel”, page 25)

Read measurement as “Float” (4-byte value) via SDO.



Node ID:	125_{dec}
COP ID:	$67D_{hex}$
Object index:	6130_{hex}
Subindex:	01_{hex}
Value:	*Read procedure*

Bürkert SAS
Rue du Giessen
F-67220 TRIEMBACH-AU-VAL