R67070.0002 - 07

Kübler

Manual

Encoders with CANopen interface

CANopen®

pulses for automation

Table of Contents

1 Document

This is the English translation of the original manual in German language.

Publisher	Kübler Group, Fritz Kübler GmbH Schubertstraße 47 78054 Villingen-Schwenningen Germany www.kuebler.com
Issue date	03/2024
Copyright	© 2024, Kübler Group, Fritz Kübler GmbH

Text sources
MicroControl Systemhaus für Automatisierung 2003, Einführung in CANopen, Uwe Koppe
CiA Specification CiA ® 406 Draft Standard Proposal, Device profile for encoders

Image sources
CSS Electronics - Intro illustrations, CAN bus, OBD2, J1939 intros
CiA Specification CiA ® 406 Draft Standard Proposal, Device profile for encoders
Beckhoff – Netzwerkmanagement, Heartbeat-Verfahren
Screenshots: Software, Vektor - CANalyzer

Code sources

Legal Notices

2 General Information

Please read this document carefully before working with the product, mounting it or starting it up.

2.1 Target Group

The device may only be planned, mounted, commissioned and serviced by persons having the following qualifications and fulfilling the following conditions:

- Technical training.
- Briefing in the relevant safety guidelines.
- Constant access to this documentation.
- In case of electrical equipment for potentially explosive atmospheres, the specialized personnel needs knowledge about the ignition protection category concept.
- For facilities in potentially explosive atmospheres, the authorized person must comply with the applicable country-specific regulations.

2.2 Symbols used / Classification of the Warnings and Safety instructions

⚠ DANGER	**Classification:** This symbol, together with the signal word **DANGER**, warns against immediately imminent threat to life and health of persons. The non-compliance with this safety instruction will lead to death or severe adverse health effects.
⚠ WARNING	**Classification:** This symbol, together with the signal word **WARNING**, warns against a potential danger to life and health of persons. The non-compliance with this safety instruction may lead to death or severe adverse health effects.
⚠ CAUTION	**Classification:** This symbol, together with the signal word **CAUTION**, warns against a potential danger for the health of persons. The non-compliance with this safety instruction may lead to slight or minor adverse health effects.

ATTENTION	**Classification:** The non-compliance with the **ATTENTION** note may lead to material damage.
NOTICE	**Classification:** Additional information relating to the operation of the product, and hints and recommendations for efficient and trouble-free operation.

2.3 Transport / Storage

Check the delivery immediately upon receipt for possible transport damages. If you do not mount the device immediately, store it preferably in its transport package.

The device must be stored at a dry and dust-free location, in compliance with the technical data, see chapter Technical Data.

3 Product Description

3.1 Technical Data Sendix M36xx, M36xxA, M58xx, M58xxA

Singleturn technology	Magnetic
Multiturn technology	Magnetic, electronic counter, Energy Harvesting
Singleturn resolution (MUR)	Max. 14 bits (default 14 bits)
Multiturn resolution (NDR)	Max. 29 bits (default 18 bits)
Multiturn resolution (TMR)	Max. 32 bits (default 32 bits)
Accuracy	± 1° (over the whole temperature range)
Repeatability	± 0.2°
Data up-to-dateness	2 ms

Mechanical characteristics for the Sendix M36xx, M36xxA encoders

Maximum rotary speed IP65 IP67	 6000 min^{-1}, 3000 min^{-1} (continuous operation) 4000 min^{-1}, 2000 min^{-1} (continuous operation)
Starting torque (at 20 °C) IP65 IP67	 < 0,007 Nm < 0,01 Nm
Permissible shaft load radial axial	 40 N 20 N
Protection level acc. to EN 60529	IP65, IP67
Working temperature range	-40 °C ... +85 °C
Materials Shaft/Hollow shaft Flange Housing Cable	 Stainless steel Aluminum Die-cast zinc PVC
Shock resistance according to EN 60068-2-27	2500 m/s^2, 6 ms
Vibration resistance according to EN 60068-2-6	300 m/s^2, 10 ... 2000 Hz

Mechanical characteristics for the Sendix M58xx M58xxA encoders

Maximum rotary spee IP65	 6000 min^{-1}, 3000 min^{-1} (continuous operation)
Starting torque (at 20 °C) IP65	 < 0,01 Nm
Permissible shaft load radial axial	 80 N 40 N
Protection level acc. to EN 60529	IP65
Working temperature range	-40 °C ... +85 °C
Materials Shaft/Hollow shaft Flange Housing Cable	 Stainless steel Aluminum Die-cast zinc PVC
Shock resistance according to EN 60068-2-27	5000 m/s^2, 6 ms
Vibration resistance according to EN 60068-2-6	300 m/s^2, 10 ... 2000 Hz

Electrical characteristics for the Sendix M36xx, M36xxR, M36xxA, M36xxAR, M58xx, M58xxA encoders

Supply voltage	10 … 30 V DC
Current consumption	Max. 30 mA (without load)

Output	RS485 CANopen CAN High-Speed according to ISO 11898, Basic and Full CAN
Type of connection	Cable or connector
Interface	CANopen Implemented profile versions: Profile DS406 V4.0 LSS DS305 V2.0 Bootloader CiA302 USF
Bootloader	CiA DS 302-3
Baud rate	10 … 1,000 kbit/s, configurable
Node address	1 … 127, configurable
Termination	Configurable

Mechanical characteristics for the Sendix M36xxR encoders

Maximum rotary speed IP67	 4000 min^{-1}, 2000 min^{-1} (continuous operation)
Starting torque (at 20 °C) IP67	 < 0,01 Nm
Permissible shaft load radial axial	 80 N 40 N
Protection level acc. to EN 60529	IP65, IP67, IP69k
Working temperature range	-40 °C ... +85 °C
Materials Shaft/Hollow shaft Flange Housing Cable	 V2A / V4A Stainless steel V4A / Aluminum V4A / Die-cast zinc PVC
Shock resistance according to EN 60068-2-27	5000 m/s^2, 4 ms
Vibration resistance according to EN 60068-2-6	300 m/s^2, 10 ... 2000 Hz

3.2 Technical Data Sendix F36xx

Singleturn technology	Optical
Multiturn technology	Battery-buffered, electronic counter, flash technology
Singleturn resolution (MUR)	Max. 16 bits (default 16 bits)
Multiturn resolution (NDR)	Max. 16 bits
Multiturn resolution (TMR)	Max. 32 bits (default 28 bits)
Accuracy	± 0.0137° (over the whole temperature range)
Data up-to-dateness	5 ms

Mechanical characteristics for the Sendix F36xx encoders

Max. rotational speed IP65 up to 70 °C IP67 up to 70 °C	 12000 min^{-1}, 10000 min^{-1} (continuous operation) 10000 min^{-1}, 8000 min^{-1} (continuous operation)
Starting torque (at 20 °C) IP65 IP67	 < 0.007 Nm < 0.01 Nm
Mass moment of inertia Shaft version Hollow shaft version	 3,0 x 10^{-6} kgm^2 7,5 x 10^{-6} kgm^2 (MT) 6 x 10^{-6} kgm^2 (ST)
Permissible shaft load radial axial	 40 N 20 N
Protection level acc. to EN 60529 Housing side Shaft side	 IP67 IP65, optional IP67
Working temperature range	-40 °C ... +85 °C
Materials Shaft/Hollow shaft Flange Housing Cable	 Stainless steel Aluminum Die-cast zinc PVC
Shock resistance according to EN 60068-2-27	2500 m/s^2, 6 ms
Vibration resistance according to EN 60068-2-6	100 m/s^2, 55 ... 2000 Hz

Electrical characteristics for the Sendix F36xx encoders

Supply voltage	10 … 30 V DC
Current consumption (no load)) 10 … 30 V DC	 max. 80 mA
Supply voltage reverse polarity protection	Yes

Output	RS485 CANopen CAN High-Speed according to ISO 11898, Basic and Full CAN
Interface	Cable or connector
Protocol	CANopen Implemented profile versions: Profile DS406 V3.2 LSS DS305 V2.0 USF
Baud rate	10 … 1000 kbit/s, configurable
Node address	1 … 127, configurable
Termination	Configurable
Data up-to-dateness	5 ms

3.3 Technical Data Sendix 58xx

Singleturn Technologie	Optisch
Multiturn Technologie	Optical, mechanical gear
Singleturn resolution (MUR)	Max. 16 bits (default 13 bits)
Multiturn resolution (NDR)	Max. 12 bits
Multiturn resolution (TMR)	Max. 28 bits (default 25 bits)
Accuracy	± 0.0117° (over the whole temperature range)
Data up-to-dateness	5 ms

Mechanical characteristics for the Sendix 58xx encoders

Maximum rotary speed	
IP65 up to 70 °C	9000 min^{-1}, 7000 min^{-1} (continuous operation)
IP65 up to T_{max}	7000 min^{-1}, 4000 min^{-1} (continuous operation)
IP67 up to 70 °C	8000 min^{-1}, 6000 min^{-1} (continuous operation)
IP67 up to T_{max}	6000 min^{-1}, 3000 min^{-1} (continuous operation)
Starting torque (at 20 °C)	
IP65	< 0.01 Nm
IP67	< 0.05 Nm
Mass moment of inertia	
Shaft version	3,0 x 10^{-6} kg/m^2
Hollow shaft version	7,5 x 10^{-6} kg/m^2 (MT) 6 x 10^{-6} kg/m^2 (ST)
Permissible shaft load	
radial	80 N
axial	40 N
Protection level acc. to EN 60529	
Housing side	IP67
Shaft side	IP65, optional IP67
Working temperature range	-40 °C ... +80 °C
Materials	
Shaft/hollow shaft	Stainless steel
Flange	Aluminum
Housing	Die-cast zinc
Shock resistance according to EN 60068-2-27	2500 m/s^2, 6 ms
Vibration resistance according to EN 60068-2-6	100 m/s^2, 55 ... 2000 Hz

Electrical data for the Sendix 58xx encoders

Supply voltage	10 … 30 V DC
Current consumption (no load)	
10 … 30 V DC	max. 110 mA
Supply voltage reverse polarity protection	Yes

Output	RS485 CANopen CAN High-Speed according to ISO 11898, Basic and Full CAN
Type of connection	Cable or connector
Interface	CANopen Implemented profile versions: Profile DS406 V3.2 LSS DS305 V2.0 USF
Baud rate	10 … 1000 kbit/s, configurable
Node address	1 … 127, configurable
Termination	Configurable

3.4 Technical Data Sendix F58xx

Singleturn technology	Optical
Multiturn technology	Battery-buffered, electronic counter, flash technology
Singleturn resolution (MUR)	
Maximum	16 bit
default	13 bit
Multiturn resolution (NDR)	
Maximum	16 bit
default	12 bit
Total resolution (TMR)	
Maximum	32 bit
default	25 bit
Scaling	Supports USF Supports gear factor
Accuracy	(over the whole temperature range)

Mechanical Characteristics for the Sendix F58xx Encoders

Maximum rotational speed	
IP67 (for short periods– 10 min)	9000 min^{-1}
IP67 (continuous operation)	6000 min^{-1}
Starting torque (at 20 °C)	
IP67	< 0,01 Nm
Mass moment of inertia	
Shaft version	3,0 x 10^{-6} kg·m²
Hollow shaft version	6,0 x 10^{-6} kg·m²
Permissible shaft load	
radial	80 N
axial	40 N
Protection level (acc. to EN 60529)	
Housing side	IP67
Shaft side	IP65 (optional IP67)
Working temperature range	-40°C ... +80°C [-40°F ... +176°F
Materials	
Shaft/hollow shaft	Stainless steel
Flange	Aluminum
Housing	Aluminum
Shock resistance (acc. to EN 60068-2-27)	2500 m/s², 6 ms
Vibration resistance (acc. to EN 60068-2-6)	100 m/s², 55 ... 2000 Hz

Electrical characteristics for the Sendix F58xx encoders

Supply voltage	10 ... 30 V DC
Maximum current consumption	100 mA
Supply voltage reverse polarity protection	Yes

Output	RS485 CANopen CAN High-Speed according to ISO 11898, Basic and Full CAN
Interface	Cable or connector
Protocol	CANopen Implemented profile versions: Profile DS406 V3.2 LSS DS305 V2.0 USF
Baud rate	10 … 1000 kbit/s, configurable
Node address	1 … 127, configurable
Termination	Configurable
Data up-to-dateness	5 ms

3.5 Technical Data Sendix F58xxM

Singleturn technology	Optical
Multiturn technology	Battery-buffered, electronic counter, flash technology
Singleturn resolution (MUR)	Max. 19 bits raw value Max. 16 bits scalable (default 13 bits)
Multiturn resolution (NDR)	Max. 24 bits raw value Max. 16 bits scalable
Multiturn resolution (TMR)	Max. 43 bits raw value and Max. 32 bits scalable (default 25 bits)
Accuracy	± 0.0137° (over the whole temperature range)
Data up-to-dateness	2 ms

Mechanical characteristics for the Sendix F58xxM encoders

Max. rotational speed IP65 up to 85 °C	 9000 min^{-1}, 6000 min^{-1} (continuous operation)
Starting torque (at 20 °C) IP65	 < 0.01 Nm
Mass moment of inertia Hollow shaft version	 6 x 10^{-6} kgm^2 (ST)
Permissible shaft load radial axial	 80 N 40 N
Protection level acc. to EN 60529 Housing side Shaft side	 IP65 IP65
Working temperature range	-40 °C ... +85 °C
Materials Shaft/Hollow shaft Flange Housing Cable	 Stainless steel Aluminum Die-cast zinc PVC
Shock resistance according to EN 60068-2-27	2500 m/s^2, 6 ms
Vibration resistance according to EN 60068-2-6	100 m/s^2, 55 ... 2000 Hz

Electrical characteristics for the Sendix F58xxM encoders

Supply voltage	10 … 30 V DC
Current consumption (no load)) 10 … 30 V DC	 max. 100 mA
Supply voltage reverse polarity protection	Yes

Output	RS485 CANopen CAN High-Speed according to ISO 11898, Basic and Full CAN
Type of connection	Cable or connector
Interface	CANopen Implemented profile versions: Communication CiA301 V4.2.0 Bootloader CiA302 V4.0.2 LSS CiA305 V3.0.0 Encoder profile DS406 V4.1.0
Baud rate	10 … 1000 kbit/s, configurable
Node address	1 … 127, configurable
Termination	Configurable

3.6 Technical Data

Singleturn technology	Optical
Multiturn technology	Optical, mechanical gear
Singleturn resolution (MUR)	Max. 16 bits (default 13 bits)
Multiturn resolution (NDR)	Max. 12 bits
Multiturn resolution (TMR)	Max. 28 bits (default 25 bits)
Accuracy	± 0.0117° (over the whole temperature range)
Data up-to-dateness	5 ms

Electrical characteristics

Supply voltage (depending on the variant)	5 … 30 V DC 10 … 30 V DC 5 V DC (±5 %)
Current consumption* (no load)	max. 100 mA
Protection class according to EN 61140	III (PELV)

*Add an external fuse for protection.

Characteristic values

Maximum rotational speed Shaft version Hollow shaft version	 6000 min^{-1} 3000 min^{-1}
Operating/storage and transport temperature range	-40 °C ... +60 °C [-40 °F … 140 °F]
Protection level according to EN 60529	IP67
Weight	appr. 1,50 kg [3.31 lb]
Installation height	< 2000 m [6562 ft]
Shock resistance according to EN / IEC 60068-2-27	2500 m/s^2 [250 g], 6 ms
Vibration resistance according to EN / IEC 60068-2-6	55 ... 2000 Hz, 100 m/s^2

Output	RS485 CANopen CAN High-Speed according to ISO 11898, Basic and Full CAN
Type of connection	Cable or connector
Interface	CANopen Implemented profile versions: Profile DS406 V3.2 LSS DS305 V2.0 USF
Baud rate	10 … 1000 kbit/s, configurable
Node address	1 … 127, configurable
Termination	Configurable

3.7 Technical Data 71xx

Singleturn technology	Optical
Multiturn technology	Optical, mechanical gear
Singleturn resolution (MUR)	Max. 16 bits (default 13 bits)
Multiturn resolution (NDR)	Max. 12 bits
Multiturn resolution (TMR)	Max. 28 bits (default 25 bits)
Accuracy	± 0.0117° (over the whole temperature range)
Data up-to-dateness	5 ms

Mechanical characteristics 71xx

Maximum rotational speed	6000 min^{-1}
Maximum angular acceleration	$5x10^5$ rad/s²
Operating/storage and transport temperature range	-40 °C ... +60 °C [-40 °F … 140 °F]
Protection level according to EN 60529	IP67
Protection level according to NEMA 250	Type 6
Installation height	< 2000 m [6562 ft]
Shock resistance according to EN / IEC 60068-2-27	1000 m/s^2, 11 ms
Vibration resistance according to EN / IEC 60068-2-6	55 ... 2000 Hz, 100 m/s^2

Electrical characteristics 70xx

Supply voltage (depending on the variant)	10 … 30 V DC
Current consumption (no load) 10 … 30 V DC	 max. 100 mA
Protection class according to EN 61140	III (PELV)

Output	RS485 CANopen CAN High-Speed according to ISO 11898, Basic and Full CAN
Type of connection	Cable or connector
Interface	CANopen Implemented profile versions: Profile DS406 V3.2 LSS DS305 V2.0 USF
Baud rate	10 … 1000 kbit/s, configurable
Node address	1 … 127, configurable
Termination	Configurable

3.8 Supported Standards and Protocols

The CANopen encoders support the CANopen communication profile according to DS301. In addition, device-specific profiles such as the encoder profile DS406 and DS417 (for lift applications) are available. The encoders comply with Encoder Class C2.

The available operating modes are Acyclic Mode, Cyclic Mode or Sync Mode. Moreover, scaling, preset values, limit switch values and many other additional parameters can be programmed via the CANbus. When switching on, all parameters are loaded from a non-volatile memory. These parameters have previously been stored in a zero-voltage secure manner. The output values position, speed, acceleration, etc. can be combined variably as a PDO (variable PDO mapping).

Besides the encoders with bus cover, devices are also available with a connector or with cable outlet, for which the device address and the baud rate can be modified in a software-controlled manner, as all devices support LSS. The models with bus connection cover and integrated T coupler ensure particularly easy installation. Bus and voltage supply are connected by means of M12 connectors. With devices with removable bus cover, the device address can be set by means of two hex rotary switches. An additional DIP witch allows setting the baud rate and a ready-to-activate terminating resistor.

Three LEDs on the rear side signal the operating and error status of the CAN bus and the condition of an internal diagnosis.

4 Installation

NOTICE	**Observe the operation manual** Installation instructions can be found in the relevant operation manual.

4.1 Electrical Installation

4.1.1 General Information for the Connection

ATTENTION	**Destruction of the device** Before connecting or disconnecting the signal cable, always disconnect the power supply and secure it against switching on again.

NOTICE	**General safety instructions** Make sure that the entire system is in a de-energized state during electrical installation.

NOTICE	**No open cable wires** Connect all required cable wires / connectors before commissioning. Insulate individually all unused ends of the output signals to avoid short-circuits. • Electrostatic discharges at the contacts of the connector or at the line ends could damage or destroy the device. Take appropriate precautionary measures.

NOTICE	**Traction relief** Always mount all lines with traction relief.

NOTICE	**Interference susceptibility** Proceed as follows: • Connect the shield to the device housing. • Comply with the maximum cable length for stub lines and for the total length of the bus network. • Check the maximum supply voltage on the device.

ATTENTION	**Wear of the memory module** Only applies to devices that write parameter data via the EEPROM: Avoid too frequent writing of the non-volatile memory.. It is used e.g. when setting a preset value. The memory module is designed for approximately 500,000 write cycles. If the maximum number of write cycles is exceeded, single memory areas may be damaged and errors may occur.

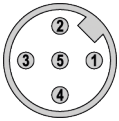
NOTICE	**Use shielded data lines** Use exclusively shielded data lines to comply with the EMC interference immunity requirements in force for interference emissions and external interference.

4.1.2 Connection Legend

+V:	Supply voltage +V DC
0V:	Ground GND (0V)
CAN_H:	Positive CAN Signal (Dominant High)
CAN_L:	Negated CAN-Signal (Dominant Low)
CAN_GND:	CAN-Ground
PH ⏚:	Connector housing (cable shield is applied on the connector housing), protective earth

4.1.3 Terminal Assignment M36, M58

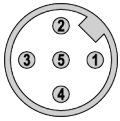
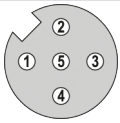
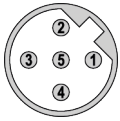
Interface	Type of connection	Cable					
2	M36: 1, 2, A, B M58: 2, B	Signal	+V	0 V	CAN_GND	CAN_H	CAN_L
		Color	BN	WH	GY	GN	YE

Interface	Type of connection	M12 connector, 5-pole						Connector
			Bus IN					
2	M36: 3, 4	Signal	+V	0 V	CAN_GND	CAN_H	CAN_L	
	M58: 4	Pin	2	3	1	4	5	

4.1.4 Terminal Assignment F36xx

Interface	Type of connection	Cable					
2	1, 3, U	Signal	+V	0 V	CAN_GND	CAN_H	CAN_L
		Color	BN	WH	GY	GN	YE

4.1.5 Terminal Assignment 58xx

Interface	Type of connection	Connector						Connector
		2 x M12 (3x M12 with interface 5) connectors, 5-pin						
2, 5	2, F		Bus IN					
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND	
		Pin	3	2	5	4	1	
			Bus OUT					
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND	
		Pin	3	2	5	4	1	
5	2		Incremental track					
		Signal	A	/A	B	/B	0V	
		Pin	1	2	3	4	5	
		1 x M12 connector, 5-pin						
2, 5	E		Bus IN					
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND	
		Pin	3	2	5	4	1	
		2 x M23 connector, 12-pin						
2, 5	J		Bus IN					
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND	
		Pin	10	12	2	7	3	
			Bus OUT					
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND	
		Pin	10	12	2	7	3	
		1 x M23 connector, 12-pin						
2, 5	I		Bus IN					
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND	
		Pin	10	12	2	7	3	
		Sub-D connector, 9-pin						
2, 5	K		Bus IN					
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND	
		Pin	6	9	2	7	3	

Interface	Type of connection	Cable					
2, 5	A, B		Bus IN				
		Signal	0V	+V	CAN_L	CAN_H	CAN_GND
		Wire color	WH	BN	YE	GN	GY

CAN-Bus connection internal terminal strip

NOTICE	**Bus connection with separate voltage supply and PG screwed fitting.**
	The bus cover must be removed to access to the internal terminal strip. Unscrew the two bus cover screws and remove the bus cover from the encoder. When re-tightening, take care to tighten the screws with a torque of 0.5 Nm.

Ansicht in die geöffnete Bushaube / View into the open bus cover

IMG-ID: 58269707

Lead the entering bus cable through the left screwed cable fitting and connect it to the left, orange terminal (CH), terminal (CL) and terminal (CG) (refer to CAN IN connection diagram).

If there are further devices in the bus strand:
Lead the continuing cable through the right screwed cable fitting and connect it to terminal (CG), terminal (CH) and terminal (CL) (refer to CAN OUT connection diagram).

Make sure that the continuous CAN bus is terminated at both ends with a 120 Ohm bus termination resistor between CAN_High (+) and CAN_Low (-).

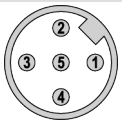
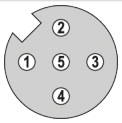
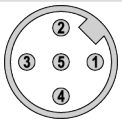
Supply voltage

Guide the encoder supply voltage through the central screwed cable fitting and connect it to the left terminals (+V) and (0V). Apply the cable shield on the screwed cable fitting (refer to CAN IN connection diagram).

Abbreviation	Designation	Direction
CG	CAN Ground	Out
CL	CAN_Low (-)	Out
CH	CAN_High (+)	Out
0V	0 volt supply	Out
+V	+UB supply	Out
CG	CAN Ground	In
CL	CAN_Low (-)	In
CH	CAN_High (+)	In
0V	0 volt supply	In
+V	+UB supply	In

4.1.6 Terminal Assignment F58x8 / S58x8FS3

Interface	Type of connection	Cable (isolate unused wires individually before commissioning)					
2	A, B, L, M	Signal	+V	0 V	CAN_GND	CAN_H	CAN_L
		Color	BN	WH	GY	GN	YE

Interface	Type of connection	M12 connector, 5-pole					Connector
2	F	Bus IN					
		Signal	+V	0 V	CAN_GND	CAN_H	CAN_L
		Pin	2	3	1	4	5
		Bus OUT					
		Signal	+V	0 V	CAN_GND	CAN_H	CAN_L
		Pin	2	3	1	4	5
2	E	Bus IN					
		Signal	+V	0 V	CAN_GND	CAN_H	CAN_L
		Pin	2	3	1	4	5

4.1.7 Terminal Assignment F58xxM

Interface	Type of connection	Cable					
2	L, M	Signal	+V	0 V	CAN_GND	CAN_H	CAN_L
		Color	BN	WH	GY	GN	YE

Interface	Type of connection	Cable									
5	L, M	Signal	+V	0 V	CAN_GND	CAN_H	CAN_L	A	$\overline{A}$	B	$\overline{B}$
		Color	BN	WH	GY	GN	YE	BK	VT	GY-PK	RD-BU

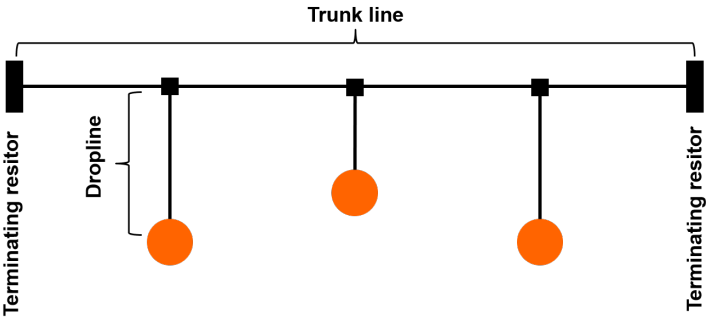
4.1.8 Terminal Assignment 70xx and 71xx

Cable									
Signal:	0 V	+V	CAN_H	CAN_L	CAN_GND	CAN_H	CAN_L	CAN_GND	⏚
Wire labeling	1	2	4	5	6	7	8	9	Shield

4.1.9 CAN network topology

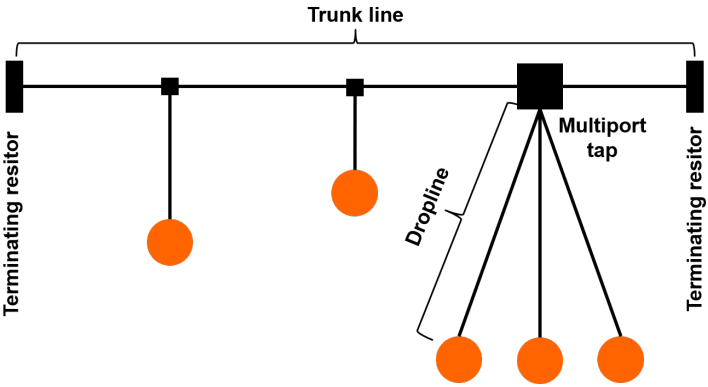
According to DIN ISO 11989, a CAN bus system can basically only have a line bus structure. In variation of this, repeaters or gateways can be used to realize other network topologies.

When designing the network topologies, it must be differentiated between the trunk line and the drop lines.
The trunk line must be terminated at both ends with a 120 ohm resistor. Drop lines may not be terminated with terminal resistors.



IMG-ID: 60283275

Baud rate [kbit/s]	Drop line length [m]	Total length of all drop lines [m]
1000	< 1	< 5
500	< 5	< 25
250	< 10	< 50
125	< 20	< 100
50	< 50	< 250



IMG-ID: 60285707

Baud rate [kbit/s]	Drop line length without multiport [m]	Trunk length without drop lines [m]
1000	< 0.3	< 25
500	< 1.2	< 66
250	< 2.4	< 120
125	< 4.8	< 310

Drop lines must be avoided to the greatest possible extent as they inevitably cause reflections. However, these reflections are uncritical as long as they have entirely decayed before the sampling time. For the bit timing settings chosen in the bus couplers, it can be assumed that this is the case.

5 Commissioning and Operation

DANGER	**Risk of injury due to rotating shafts**
	Hair and loose clothing can be caught by rotating shafts. • Prepare all work as follows: ⇨ Switch the operating voltage off and stop the drive shaft. ⇨ Cover the drive shaft if the operating voltage cannot be switched off.

5.1 Function and Status LED

The device is equipped with three LEDs for displaying status and error messages.

Green = CANopen BUS status

Red = CANopen ERR display

Yellow = Diagnosis

General LED display

Display	LED	Meaning	Error cause	Addition
Bus off		No connection to the master	Data line interruption Wrong baud rate Interchanged data line	Observe the combination with the ERR LED Also off, check the supply voltage
Bus on		Connection to the master Operational state		PDO transfer active
Bus flashing about 250 ms		Connection to the master Pre-operational state		SDO communication
Bus flashing single flash about 1 sec.		Connection to the master Stopped state		SDO communication impossible Only NMT commands
Bus flashing triple flash		Program download		
DIAG off		Device operates error-free		Observe the combination with the Bus status
DIAG flashing		Internal error, temperature overrun, sensor monitoring, single-step error Sensor LED current monitoring		BUS LED green flashing or on: depending on object 1029h

Error LED display after powering

Display	LED	Meaning	Error cause	Addition
ERR off		Device operates error-free		Observe the combination with the BUS LED
ERR on		BUS OFF status	Bus short-circuit or wrong baud rate	
ERR flashing		Connection to the master interrupted Wrong configuration	Combination with BUS state	BUS LED green flashing or on: depending on object 1029h Error behavior
ERR flickering 4 times		Error passive		CAN controllers in Error Passive condition can only signal a detected error with six homogeneous recessive bits.
ERR + Bus alternating quick flashing		Connection / device error LSS active Global mode active	Internal data connection with sensor faulty Sensor faulty Encoder waiting for configuration	Device must be sent back for repair by manufacturer LSS mode
ERR + Bus simultaneous quick flashing (300ms)		Failure of network participant detected	Watchdog error	Device must be sent back for repair by manufacturer
Bus + ERR flickering 4 times		Error Passive in Operational mode		CAN controllers in Error Passive condition can only signal a detected error with six homogeneous recessive bits.

LED combinations during operation

Display	LED	Meaning	Error cause	Addition
ERR + DIAG flashing		Error in Pre-op mode	Temperature overrun sensor monitoring single-step error Sensor LED current monitoring	Device in Pre-operational mode Analyze the emergency messages
ERR + DIAG flashing		Error in Pre-op mode	Temperature overrun sensor monitoring single-step error Sensor LED current monitoring	Device on CAN bus connection with master ok + additional error cause

LED display when switching on the device with SET key pressed - general RESET

Display	LED	Meaning	Error cause	Addition
DIAG flashing fast		Diagnostic mode		Device ready for diagnostic

Definition of the LED states See LED States [▶ 131].

5.2 Quick Start Guide

5.2.1 Default Settings

All parameters are created as objects in CANopen. The most important parameters for CANopen are specified with their default values in the table below:

Index	Name	Default	Note
0x2100	Baud rate	5	250 kbit/s
0x2101	Node number	0x3F	63 decimal
0x2102	CAN bus termination	0	0 = Off *)
0x6000	Scaling	0x04	4 = On**)
0x6001	Measuring units per revolution (MUR)	0x2000	Depending on the type, see Device-Specific Objects [▶ 72]
0x6002	Total measuring range (TMR)	0x2000000	Depending on the type, see Device-Specific Objects [▶ 72]

*) Depending on the device type Object 0x2102 - CAN-bus Termination Off/On [▶ 87]

**) Depending on the device type Object 0x6000 - Operating Parameters [▶ 91]

The following tables indicate the respective factory settings for the single objects.

The original standard values (default values at the delivery) can be loaded again with object 0x1011 and command "load" (0x6C6F6164) (Object 0x1011 - Load factory settings [▶ 79]).

To save the modified parameters in a voltage failure-safe manner, they must imperatively be transferred in the non-volatile memory with object 0x1010 (Object 0x1010 - Save Parameters [▶ 79]).

All data previously saved in the non-volatile memory is overwritten.

5.2.1.1 Mapping Parameters - TPDO1-3

Mapping	TPD01 1800_h	TPDO2 1801_h	TPD03 1802_h
Mapping object	$1A00_h$	$1A01A_h$	$1A02A_h$
Entry	0x60040020	0x60040020	0x60300120
Process	Position	Position	Speed
Object	0x6004	0x6004	0x6030
Subindex	00	00	01
Data length	20_h (32 bits)	20_h (32 bits)	10_h (16 bits)
Transmission type	Asynchronous	Synchronous	Asynchronous

Transmit TPDO 1 (0x1800) Position asynchronous

Default COB-ID is 180 + node number: Example 0x180 + 0x3F = 0x1BF

Message	Byte 0	Byte 1	Byte 2	Byte 3
1BF	Position LSB	XX	XX	Position MSB

Default setting: Transmission type = 255, Event timer 20ms

Transmit TPDO2 (0x1801) Position synchronous

Default COB-ID is 280 + node number: Example 0x280 + 0x3F = 0x2BF

Message	Byte 0	Byte 1	Byte 2	Byte 3
2BF	Position LSB	XX	XX	Position MSB

Default setting: Transmission type = 0x01, Syn mode, Sync at every pulse,
Event timer 0ms

Transmit TPDO3 (0x1802) Speed asynchronous

Default COB-ID is 380 + node number: Example 0x180 + 0x3F = 0x3BF

Message	Byte 0	Byte 1
3BF	Speed LSB	Speed MSB

The value for the speed is signed (Signed INT).

Default setting: Transmission type = 254, Event timer 0ms

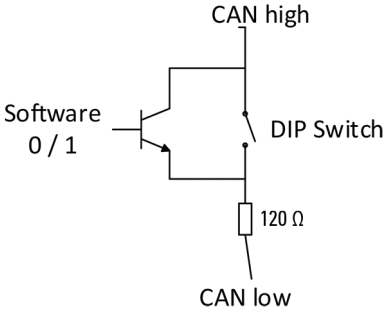
5.2.2 Setting of the Termination

For devices with bus cover, bus termination can be set by hardware via DIP switch 1 on the printed circuit board, for devices without removable bus cover, it can be set via software object 0x2102. The looped-through CAN bus must be terminated at both ends with a 120 Ohm bus termination resistor between CAN+ and CAN-.

Dip switch 1	Bus termination
Off	Off
On	On (120 Ohm)

NOTICE	**Software termination via hardware termination**
	The termination can be switched on and off with object 0x2102. But, if the DIP switch is in ON position, the setting of object 0x2102 has no effect on the termination.

The position of the DIP switch affects the possibility of setting the termination via software.



IMG-ID: 9007199418941579

5.2.3 Setting the Baud Rate

There are various possibilities to set the baud rate:

- Change via object 0x2100
- Change via LSS, see LSS Services DS 305 [▶ 45]
- Change via the DIP switches (only possible for devices with removable bus cover)

NOTICE	**Observe the value of the baud rate.**
	With LSS, the value is exactly reversed with respect to the corresponding baud rate!

The following baud rates are available to the user:

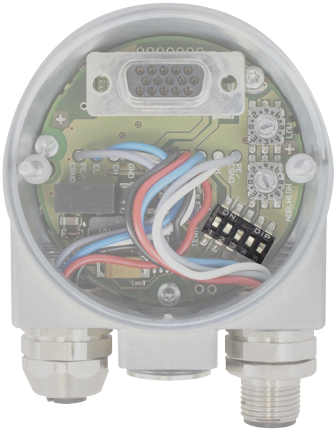
Value via object 0x2100	Value via LSS	Baud rate in kbit/s	Note
0	8	10	
1	7	20	
2	6	50	
3	5	100	No longer recommended by CiA
4	4	125	
5	3	250	
6	2	500	
7	1	800	
8	0	1000	Not supported by all devices

NOTICE	**Note for the corresponding baud rate**
	The selected cycle time must be longer than the bus transmission time so that the PDOs can be transmitted error-free (see object 1800h, subindex 5 - Event timer). For baud rate 10 KBaud: Cycle time at least 14 ms. for baud rate 20 KBaud: Cycle time at least 10 ms. for baud rate 50 KBaud: Cycle time at least 4 ms If cycle time = 0 in the Event timer (i.e. the PDO is transmitted in the event of a value change), the baud rate must be at least 125 KBaud.

The following combinations apply if the baud rate is set via the DIP switches.

Dip switch				Value for object 0x2100	Baud rate in kbit/s
2	**3**	**4**	**5**		
Off	Off	Off	Off	0	12
On	Off	Off	Off	1	20
Off	**On**	Off	Off	2	50
Off	Off	**On**	Off	4	125
On	Off	**On**	Off	5	250
Off	**On**	**On**	Off	6	500
Off	Off	Off	**On**	8	1000

The DIP switches are located on the printed circuit board inside of the removable bus cover.



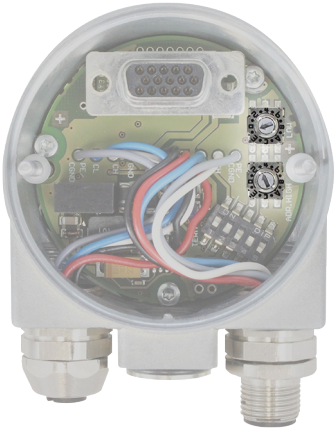
IMG-ID: 59420939

The default setting of the DIP switches is 250 kbit/s.

	1	2	3	4	5
	Termination	Baud rate	Baud rate	Baud rate	Baud rate
On		X		X	
Off	X		X		X

5.2.4 Setting of the Node Number

The node number is set by means of the two address rotary switches.



IMG-ID: 59422859

For devices without bus cover, the node number is set via software on object 0x2101 or via a corresponding LSS service.

Node number 0 is reserved and shall not be used by any node. The resulting node numbers are in the range 1...7Fh hexadecimal (1...127 decimal).

NOTICE	**Taking over a new node number** A new node number is only taken over at the following booting (reset/ power-on) of the device or via a NMT Reset node command. All other settings in the objects table remain retained.

5.2.5 Changing the Mapping Parameters

Procedure for changing the mapping parameters

1. Set the PDO to invalid (0x1800h, subindex 1, bit 31 to "1").
2. Set subindex 0 in the mapping parameter (0x1A00h) to "0".
3. Modify the mapping entries (0x1A00h, SI 1… 8).
4. Set subindex 0 in the mapping parameter to a valid value. The device then checks the consistency of the entries.
5. Set the PDO back to valid (0x1800h, subindex 1, bit 31 to "0").

5.2.6 Reset of the device

There are 2 ways to reset the device to its factory setting:

1. Reset via object 0x1011 - Restore default parameters.
2. Reset via the Set key (for devices with external Set key).

Factory settings via object 0x1011 - Restore default parameters.

The default values can be restored with a specific command. In order to prevent accidental loading of the default values, the command is only carried out when the code word string "LOAD" is entered in this subindex. A RESET node must be performed subsequently.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	11	10	01	6C	6F	61	64
Response	60	11	10	01	00	00	00	00

Factory settings via the Set key

Devices can be reset to factory settings with the integrated Set key. All parameters are reset.

IMG-ID: 158736139

The following steps are required:

1. Switch the device off.
2. When switching it back on, hold the Set key pressed for about 3 seconds, until the DIAG LED flashes.
3. Switch the device off again.

⇒ When it is booted again, all values are reset to the default settings (identical with sending object 0x1011 - Reload parameters).

5.2.7 Performing a Preset

2 steps are required to set the device to a preset value:

1. Set the preset via object 0x6003 – Preset value.
2. Carry out the preset via the Set key (for devices with external Set key).

Saving the preset in object 0x6003 – Preset value.

Describing object 0x6003 allows configuring a preset value. Value = 0 is stored here in the factory.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	03	60	01	00	00	00	00
Response	60	03	60	01	00	00	00	00

Performing a preset with the Set key

The Set key integrated in certain devices allows setting them to the preset value. The position resulting from this depends on the value stored in object 0x6003.

IMG-ID: 158736139

Proceed as follows:

1. Make sure that the device is in Operational or Pre-operational mode.
2. Press the Set key.

⇒ The device now returns the preset position configured in object 0x6003.

5.2.8 Saving the parameters

Command "SAVE“ under 0x1010 subindex 1 - Save all Parameters saves the parameters in the non-volatile memory.

All communication objects, application objects and manufacturer-specific objects are saved under this subindex. This operation requires about 14 ms.

In order to prevent accidental saving, the command is only carried out when the code word string SAVE – converted in Hex – is entered in this subindex. See Object 0x1010 - Save Parameters [▶ 79].

A read access to subindex 1 provides information about the memory functionality

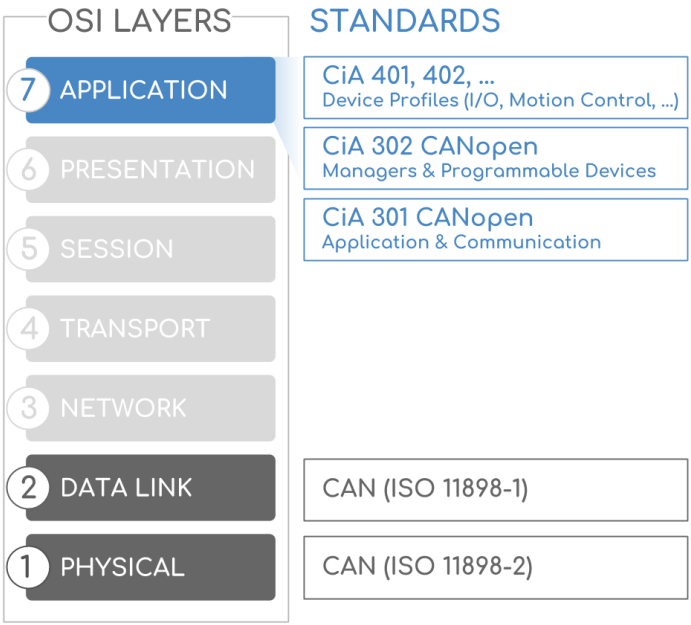
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	10	10	01	73	67	76	65
Response	60	10	10	01	00	00	00	00

5.3 Protocol Features

5.3.1 Interface Description CANopen

The CANopen protocol is a standardized layer 7 protocol for the CAN bus. The CANopen protocol defines on the one hand the "How" of the communication, that is to say with which telegrams (i.e. identifiers) the devices can be addressed. For this purpose, mechanisms for the exchange of process data in real-time, as well as for the transmission of large data volumes or the sending of alarm telegrams are implemented. On the other hand, CANopen defines the "What" of the communication. This means that a parameter is addressed via a defined interface (via the device profile) for setting a device.

These so-called device profiles are organized in the form of a table (object directory). The so-called "communication profile", which allows querying or setting basic device data, is common to all device profiles. This data includes for example the device designation, the hardware and software version, the error status, the used CAN identifiers and many other parameters. The device profiles describe the specific abilities or parameters of a "class" of devices. So far, device profiles have been defined for digital or analog EIA devices, drives, operating devices, sensors and controllers, programmable controls, encoders, medical technology, local public transport, batteries and extrusion systems. Many other profiles are in preparation.



IMG-ID: 103073419

5.3.2 CANopen Communication Profile DS 301

CANopen uses four communication objects (COB) with various features:

1. Process data objects for real-time data (see PDO).
2. Service data objects for parameter and program transmission (see SDO).
3. Network management objects for life-guarding, heartbeat, boot-up etc. (see NMT or also NMO)
4. Special function objects for synchronization, time-stamp, emergency protocol (see SFO).

All device parameters are saved in an object dictionary. This object dictionary contains the description, data type and structure of the parameters, as well as the address (index).

The directory is divided into the following sections:

- Communication profile
- Manufacturer profile
- Device profile

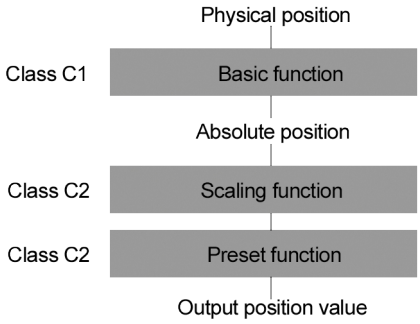
5.3.3 Encoder device profile DS 406

This profile describes a manufacturer-independent and binding specification of the interface for encoders. The profile defines which CANopen functions are used and how they are to be used. This standard allows for an open and manufacturer-independent bus system.

The device profile is structured in two object classes:

- Class C1 describes all basic functions the encoder is to offer.
- Class C2 includes a wide range of extended functions, which have to be supported by the encoders of this class (mandatory) or which are optional.

So devices of class C2 include all C1 and mandatory C2 functions, as well as further manufacturer-specific optional functions. Moreover, an address range, which can receive manufacturer-specific special functions, is defined in the profile.



IMG-ID: 18014398567705611

5.3.4 PDO Mapping

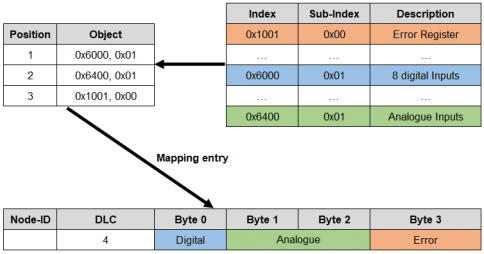
PDO mapping is the image of the application objects (real-time data) from the object dictionary in the process data objects. The CANopen device profiles provide for every device type a default mapping that is suitable for later applications. The default mapping maps the outputs according to their physical sequence in the transmit process data objects.

The current mapping can be read via the corresponding entries in the object dictionary, the so-called mapping tables. The first position of the mapping table (subindex 0) contains the number of mapped objects, which are listed subsequently. The tables for the TxPDOs are in the object dictionary at index 0x1A00h.

5.3.4.1 Dynamic and Variable Mapping

Dynamic / variable mapping allows assigning different PDOs to a communication object (see TPDO). This allows transmitting different objects simultaneously in the same way. Not every object can be mapped. Theoretically, up to 64 objects can be mapped per TPDO.

The user thus has the possibility to compile individually the process data and have it transmitted so that it does not overload the bus or the master. As a general rule, up to five TPDOs are supported.



IMG-ID: 160254347

With the variable PDO mapping, the mapping entries can only be modified in the NMT state Pre-operational.

With the dynamic PDO mapping, the mapping entries can be modified during operation, thus in the NMT state Operational.

5.3.5 Network Management Services

All CANopen devices can be controlled by the network management. A special identifier 0x00 is reserved for this purpose. It is used to send instructions to one or to all devices. The command consists of two bytes. The first byte contains the command code, the second byte the node address of the controlled CANopen device.

Structure of the message of the superordinate controller:

Identifier	Byte 0	Byte 1
0x00	Command	Node address

NOTICE	**Node address Zero** Node address Zero allows activating all nodes in the network simultaneously. The CANopen devices do not acknowledge the NMT instructions.

The "Pre-operational" state is provided for the network configuration phase (SDO). To operate with PDOs, the node must have accepted the "Operational" communication state. As a general rule, the network is started by the host and set in "Operational" condition with the help of the "NMTStart".

Command byte	Description	Mode
0x01	Start_Remote_Node: Switch to Operational	Operational
0x02	Stop_Remote_Node: Switch to Prepared	Pre-operational
0x80	Enter_Pre-operational_State: Switch to Pre-operational	Pre-operational
0x81	Reset_Node: Reset the node	
0x82	Reset_Communication: Reset the communication	

Example

Starting node number 63 (0x3f - Switching to the operating mode)

Identifier	Byte 0	Byte 1
	Command	Node address
0x00	01	3F

5.3.6 LSS Services DS 305

The CiA DSP 305 CANopen Layer Setting Service and Protocol (LSS) were created to read and modify the following parameters via the network:

1. Node address
2. Baud rate
3. LSS address

These abilities increase the "plug-and-play" compatibility of the device and considerably simplify the configuration possibilities. The LSS master is responsible for the configuration of these parameters for one or several slaves in the network.

LSS hardware requirements (LSS address)

All LSS slaves must have a valid object entry in the objects dictionary for identification object 0x1018 in order to be able to carry out a selective configuration of the node. This object consists of the following subindexes:

- Manufacturer ID
- Product code
- Revision number

- Serial number
- LSS master CAN-ID 2021
- LSS slave CAN-ID 2020

Product code, revision number and serial number are set by the manufacturer. The LSS address must be unequivocal in the network.

LSS operative restrictions

In order to ensure trouble-free LSS functionality, all devices in the network must support the LSS services. Furthermore, the following applies:

- There can be only one LSS master.
- All nodes must start with the same baud rate.
- LSS communication can only take place in Stopped mode or in Pre-operational mode.

Use of LSS

For a detailed description of the LSS functionality, see Using the LSS [▶ 119].

5.3.7 Data Transmission

With CANopen, data is transferred using two different communication types (COB=Communication Object) with different characteristics:

- Process Data Objects (PDO - real-time capable)
- Service Data Objects (SDO)

The Process Data Objects (PDO) are used for highly-dynamic exchange of real-time data with a maximum length of 8 bytes (e.g. encoder position, speed, status of the compared positions) . This data is transferred with high priority (low COB identifier). PDOs are broadcast messages and make their real-time data available simultaneously to all desired recipients. PDOs can be mapped. 4 position bytes and 2 speed bytes can be combined in one 8-byte data word.

The Service Data Objects (SDO) provide the communication channel for the transmission of device parameters (e.g. programming of the encoder resolution). As these parameters are transmitted acyclically (e.g. only once when starting the network up), the SDO objects have a low priority (high COB identifier).

5.3.7.1 Process Data Transmission - PDO

Up to three PDO services, PDO1 (tx), PDO2 (tx) and PDO3 (tx) are available for CANopen devices. A PDO transfer can be triggered by various events (see object dictionary index 1800h):

1. Synchronous pulse, triggered by an internal cyclic device timer (event timer) or by a process value change of the sensor data.
2. Synchronous pulse as a response to a SYNC telegram
   A SYNC command prompts all CANopen nodes to save synchronously their values, and to send them subsequently one after the other on the bus according to the set priority)
3. Pulse as a response to an RTR telegram
   Exactly the message with the transmitted identifier is queried per remote frame (recessive RTR bit).

The PDO messages can have the following structure:

COB-ID	Process data in binary code							
11 Bit	Byte 0 $2^7...2^0$	Byte 1 $2^{15}...2^8$	Byte 2 $2^{23}...2^{16}$	Byte 3 $2^{31}...2^{24}$	Byte 4	Byte 5	Byte 6	Byte 7
Default:	Position value							
Typ1:	Position value				Flags [1]			
Typ2:	Position value				Speed [2]		Acceleration [3]	

1) Flags status byte of the working area object 6400h
2) Speed 16-bit word, signed
3) Acceleration 16-bit word, signed

IMG-ID: 72293643

- Transmit PDO 1 is composed (mapped) of the 32-bit values of the position and the state of the Working area register (6400h).
- Transmit PDO 2 is composed of the 32-bit values of the position, 16 bits speed and 16 bits acceleration.
- Transmit PDO 3 is composed of the position as SYNC PDO.

NOTICE	**PDO combinations**
	Any other PDO combination with other objects is also possible, provided the maximum data length of 8 bytes is not exceeded.

5.3.7.2 Service Data Transmission - SDO

DO-COB-ID

The following identifiers are available as a standard for the SDO services:

- SDO (tx) (slave→master): 580h (1408) + node number
- SDO (rx) (master→slave): 600h (1536) + node number

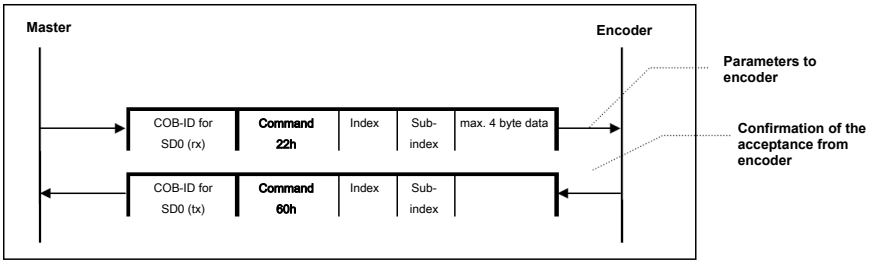
The SDO identifiers cannot be modified.

The command byte describes the type of the SDO message:

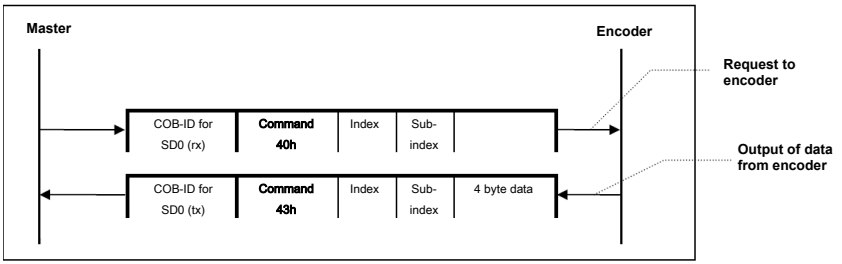
Command	Type	Function
0x23	SDO (rx), Initiate Download Request	Send parameters to the device (max. 4 bytes)
0x27	SDO (rx), Initiate Download Request	Send parameters to the device (max. 3 bytes)
0x2B	SDO (rx), Initiate Download Request	Send parameters to the device (max. 2 bytes)
0x2F	SDO (rx), Initiate Download Request	Send parameters to the device (max. 1 byte)
0x60	SDO (rx), Initiate Download Request	Confirmation of the acceptance to the master
0x40	SDO (rx), Initiate Download Request	Request parameters from the device
0x43	SDO (rx), Initiate Download Request	Parameters to master with data length = 4 bytes (unsigned 32)
0x4B	SDO (rx), Initiate Download Request	Parameters to master with data length = 2 bytes (unsigned 16)
0x4F	SDO (rx), Initiate Download Request	Parameters to master with data length = 1 bytes (unsigned 8)
0x80	SDO (rx), Initiate Download Request	Device sends error code to master

NOTICE	**Error messages**
	In case of an error, an error message (command 80h) replaces the normal acknowledgment (response). The error message includes errors in the communication protocol (e. g. wrong command byte) as well as access errors to the objects dictionary (e. g. wrong index, attempt to write a read-only object, wrong data length). The error codes are described in the CANopen profile (DS 301) or in the device profile (DS 406). See General CANopen Error Codes.

Example of a service data transmission to and from the device



Master transmits parameters to encoder



Master requests parameters from encoder

IMG-ID: 18014398581767051

5.3.7.3 PDO Transmission Types

The PDOs can be transmitted in various ways:

Code (dec.)	Transmission type				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1 … 240	X		X		
241 … 251	reserved				
252			X		X
253				X	X
254				X	
255				X	

Transmission type definition:

0:	After SYNC, but only in the event of a value change since the last SYNC.
1 ... 240:	Send value according to 1. ... 240. SYNC. The number of the transmission type indicates the number of SYNC pulses required to send the PDOs.
252:	SYNC leads to internal saving of the value, but the value must be collected per RTR.
253:	The value is updated and sent upon RTR.
254:	Event timer and/or manufacturer-specific event (for Kübler: value change)
255:	Event timer / Cyclic timer.

Acyclic synchronous

PDOs of transmission type 0 operate synchronously, but not cyclically. A device whose TxPDO is configured for transmission type 0 determines its input data when receiving the SYNC (synchronous process image). It then sends the data, if it corresponds to an event such as for example the change of an input. Transmission type 0 combines the reason for triggering the transmission in an "event-driven" way with the sending or processing moment SYNC reception.

Cyclic synchronous

With transmission type 1 … 240, the PDO is sent cyclically after every n-th SYNC (n=1...240). Since the transmission type can be combined not only in the network, but also in a device, it is possible for example to define a fast cycle for positions (n=1), while for example the temperature data is transmitted in a slower cycle (e.g. n=10). The cycle time (SYNC rate) can be monitored (object 0x1006). In the event of a SYNC failure, the device reacts according to the definition of the device profile and for example switches its outputs in the fault state.

RTR only

Transmission types 252 and 253 apply to process data objects, which are transmitted exclusively upon request by a remote frame. 253 is asynchronous. Data is determined here continuously and sent upon request. This transmission type is not recommended, as some CAN controllers only support partially input data collection. Since some of the CAN controllers respond to remote frames automatically (without requesting previously up-to-date input data), the up-to-dateness of the polled data may possibly be doubtful.

For these reasons, transmission types 252 and 253 will no longer be supported by the Kübler encoders of the M36XX, M58XX, F58XXM series, neither in the future.

Asynchronous

Transmission types 254 and 255 are asynchronous, or also event-driven. For transmission type 254 the event is defined manufacturer-specifically and in the device profile for transmission type 255. In the simplest case, the event is the change of an input value. Therefore every value change is transmitted. The asynchronous transmission type can be coupled with the event timer and thus supplies input data also if no event occurred currently. It must be noted for TT 254 that inhibit time > 100 must be set. Otherwise a CAN overrun error may occur, since the position of the last digit is constantly changing.

5.3.7.4 SDO access

The SDOs allow the direct access to the object dictionary of the CANopen device. This access is simply and clearly laid out. An SDO access is always initiated by the superordinate controller (host), which sends the following to the CANopen device:

- Either a write instruction to modify a parameter of the object dictionary.
- Or a read instruction to read a parameter.

The host receives a response for every instruction. This response either contains the read value or, in the case of a write command, it serves as an acknowledgment. The identification of the message for the CANopen device is ensured by the COB-ID. The structure of the instructions or responses depends on the data type of the object that is to be read or written. 1, 2 or 4 data bytes are sent or received.

5.3.7.4.1 Write Access

Data transfer from the host to the slave

Every access to the object dictionary is checked for validity by the slave. Every write access to non-existing or write-protected objects, or to non-matching data formats is rejected and answered with a corresponding error message.

Request from the host

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CMD	Index LSB ---------- MSB		Subindex	Data LSB ---------- MSB			

CMD determines the direction of the data transmission and the size of the data object:

- 23 hex: Transmission of 4-byte data (bytes 4...7 contain a 32-bit value)
- 2B hex: Transmission of 2-byte data (bytes 4, 5 contain a 16-bit value)
- 2F hex: Transmission of 1-byte data (byte 4 contains a 8-bit value)

Index

- 6-bit value; index of the object to be written (in the object dictionary)

Subindex

- 8-bit value; subindex of the object to be written (in the object dictionary)

Data

- 8-bit, 16-bit or 32-bit value

Response from the slave

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
RES	Index LSB ---------- MSB		Subindex	Reserved			

RES

- 60 hex data successfully transmitted
- 80 hex error, bytes 4 ... 7 contain the error code according to the standard

Index

- 16-bit value, index of the object addressed by the host telegram

Subindex

- 8-bit value, subindex of the object addressed by the host telegram

Reserved

- Not used or error message (depending on RES)

Example

Writing in operating parameter index 6000, subindex 00

Value = 1h (counting direction = ccw)

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2B	00	60	00	01	00	00	00
Response	60	00	60	00	00	00	00	00

5.3.7.4.2 Read access

Data transfer from the slave to the host

Every read access to non-existing objects is responded with an error message.

Request from the host

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CMD	Index LSB ---------- MSB		Subindex	Reserved			

CMD determines the direction of the data transmission

- 40 Hex read access (in any case)

Index

- 16-bit value; index of the object to be read (in the object dictionary)

Subindex

- 8-bit value; subindex of the object to be read (in the object dictionary)

Reserved

- Is not used

Response from the slave

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
RES	Index LSB ---------- MSB		Subindex	Data LSB ---------- MSB			

RES

- 43 hex: Bytes 4 ... 7 contain a 32-bit value
- 4B hex: Bytes 4, 5 contain a 16-bit value
- 4F hex: Byte 4 contain a 8-bit value
- 80 hex: Error, bytes 4 ... 7 contain the error code according to the standard

Index

- 16-bit value, index of the object addressed by the host telegram

Subindex

- 8-bit value, subindex of the object addressed by the host telegram

Data

- Data or error message (depending on RES)

Example

Reading the status word (index 6000, subindex 00)

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	40	00	60	00	00	00	00	00
Response	4B	00	60	00	01	00	00	00

5.3.8 Objects in the Predefined Connection Set

For an easier identifier management, CANopen uses the Predefined Master/Slave Connection Set. Thus, CANopen supports a predefined assignment of message identifiers. All identifiers (CAN-ID or COB-ID) are defined in the object directory with default values. These identifiers can be modified in a customer-specific way by means of SDO access.

The 11-bit identifier consists in a 4-bit function code and a 7-bit node number. The identifiers support:

- An emergency message per node
- Synchronization and timestamp messages
- One SDO connection per device
- The NMT messages for node control and node monitoring
- Up to 4 transmission and 4 receive PDOs per device

A CANopen network allows differentiating up to a maximum of 127 participants. These nodes jointly use the 11-bit identification area. It is thus possible to operate systems including a superordinate control node and up to 127 slave nodes without reconfiguration.

First, a differentiation is made between network-related and device-related functions. A CAN identifier is reserved for each of the network-related functions (e. g. NMT node control). An identification per device is necessary for every device-related functionality (e. g. emergency message, PDOs), as it must be possible to differentiate between the same functions on different devices.

A COB-ID with higher priority is assigned to the important functions. To allow future extensions and for historical reasons, some COB-IDs are not assigned.

The diagram shows the resulting division of the CAN identifier area and the priority assigned to every COB-ID:

NOTICE	**COB-ID prioritization**
	The higher the value of the COB identifier, the lower its priority.

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

SDOs and PDOs are always used in pairs (i.e. to transmit and receive). The rule is that the node transmits on the lower (and thus higher-priority) COB-ID and receives on the higher (and thus lower-priority) COB-ID.

Broadcast (network-wide) objects

Object	Function code	Resulting COB-ID	Communication parameters at index
NMT	0000_b	0 ($000_{h)}$	–
SYNC	0001_b	128 (80_h)	1005_h, 1006_h, 1007_h
TIME STAMP	0010_b	256 (100_h)	1012_h, 1013_h

Peer-to-peer objects

COB	Function code	Resulting COB-IDs
EMCY	0001_b	129 (081_h) - 255 ($0FF_h$)
PDO1 (tx)	0011_b	385 (181_h) - 511 ($1FF_h$)
PDO1 (rx)	0100_b	513 (201_h) - 639 ($27F_h$)
PDO2 (tx)	0101_b	641 (281_h) - 767 ($2FF_h$)
PDO2 (rx)	0110_b	769 (301_h) - 895 ($37F_h$)
PDO3 (tx)	0111_b	897 (381_h) - 1023 ($3FF_h$)
PDO3 (rx)	1000_b	1025 (401_h) - 1151 ($47F_h$)
PDO4 (tx)	1001_b	1153 (481_h) - 1279 ($4FF_h$)
PDO4 (rx)	1010_b	1281 (501_h) - 1407 ($57F_h$)
SDO (tx)	1011_b	1409 (581_h) - 1535 ($5FF_h$)
SDO (rx)	1100_b	1537 (601_h) - 1663 ($67F_h$)
NMT error control	1110_b	1793 (701_h) - 1919 ($77F_h$)

Restricted, reserved objects

COB-ID	Used by COB
0 (000h)	NMT
1 (001h) - 127 (07Fh)	reserved
257 (101h) - 384 (180h)	reserved
1409 (581h) - 1535 (5FFh)	default SDO (tx)
1537 (601h) - 1663 (67Fh)	default SDO (rx)
1760 (6E0h) - 1791 (6FFh)	reserved
1793 (701h) - 1919 (77h)	NMT Error Control
2020 (780h) - 2047 (7FFh)	reserved

5.3.9 Node Guarding and Heartbeat

A CANopen device can be monitored by two mechanisms that can be used alternatively. Monitoring with node guarding or heartbeat makes sense when the participant does not transmit its data cyclically (thus e. g. per event timer).

Node guarding means that the NMT master sends messages to the present CANopen slaves, which must respond to the message within a determined period of time. If a node is faulty, this is only detected by the NMT master. If the NMT master fails, the whole network breaks down.

This is why the heartbeat protocol has been developed, which is available as from version 4.0. With the heartbeat, every node sends its message autonomously on the network. This message can be detected and thus monitored by every network participant (and not only by the master).

5.4 CANopen Object Dictionary

The object dictionary describes the whole range of functions (parameters) of a CANopen device and is organized in the form of a table. The object dictionary not only contains the standardized data types and objects of the CANopen communication profile and the device profiles, but also, if applicable, manufacturer-specific objects and data types. The table shows which objects are supported by which device, or which default value is assigned to the object for the respective device.

The description of the object directory entries is structured as follows:

Index (hex)	Subindex (hex)	Object	Name	Type	Attr.	M/O

Index

16-bit address of the entry

Subindex

8-bit pointer to a subentry

- Is only used with complex data structures (e. g. record, array)
- No subentry Subindex=0

Object

- NULL entry without data
- DOMAIN larger variable volume of data, e. g. program code
- DEFTYPE data types definition, e. g. boolean, float, unsigned16
- DEFSTRUCT definition of a record entry, e. g. PDO mapping structure
- VAR single data value, e. g. boolean, float, unsigned16, string
- ARRAY field with similar data, e. g. unsigned16 data
- RECORD field with arbitrarily mixed data types

Name

Short description of the function

Type

Data type, e. g. boolean, float, unsigned16, integer

Attribute

Specifies the access rights for the object:

- rw read and write access
- ro only read access
- const only read access, value = constant

M/O

- M Mandatory: The object must be implemented in the device
- O Optional: The object must not be implemented in the device

5.4.1 Structure of the object dictionary

The whole object dictionary is subdivided into several areas:

Index range	Use
0x0000	Unused
0x 0001-0x009F	Data types (special case)
0x 00A0-0x0FFF	Reserved
0x 1000-0x1FFF	Communication profile
0x 2000-0x5FFF	Manufacturer-specific area
0x 6000-0x9FFF	Up to 8 standardized device profiles
0x A000-0xAFFF	Process images of IEC61131 devices
0x B000-0xBFFF	Process images of CANopen gateways according to CiA 302-7
0x C000-0xFFFF	Reserved

5.4.2 Communication Objects

Index range	Description
0x1000 to 0x1029	General communication objects
0x1200 to 0x12FF	SDO parameter objects
0x1300 to 0x13FF	CANopen safety objects
0x1400 to 0x1BFF	PDO parameter objects
0x1F00 to 0x1F11	SDO manager objects
0x1F20 to 0x1F27	Configuration manager objects
0x1F50 to 0x1F54	Program control objects
0x1F80 to 0x1F89	NMT master objects

Sendix 58x8, 70x8, 71x8

Objects [hex]	Subindex	Object name	R/W	Data type	5858 5878	5868 5888	7058 7078	7068 7088	7158 7178	7168 7188
1000		DeviceType	CONST	Unsigned32	x	x	x	x	x	x
1001		ErrorRegister	RO	Unsigned8	x	x	x	x	x	x
1002		Manufacturer status	RO	Unsigned32						
1003		Predefined error field	RO	Unsigned32	x	x	x	x	x	x
1005		COB-ID SYNC	RW	Unsigned32	0x80	0x80	0x80	0x80	0x80	0x80
1006		ComCycle Period	RW	Unsigned32	x	x	x	x	x	x
1007		Sync WindowLen	RW	Unsigned32	x	x	x	x	x	x

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
1008		Manufacturer DeviceName	CONST	visible string	x	x	x	x	x	x
1009		Manufacturer Hardware Version	CONST	visible string	x	x	x	x	x	x
100A		Manufacturer Software Version	CONST	visible string	x	x	x	x	x	x
100C		GuardTime	RW	Unsigned16	0x0	0x0	0x0	0x0	0x0	0x0
100D		LifeTimeFactor	RW	Unsigned8	0x0	0x0	0x0	0x0	0x0	0x0
1010		Store Parameters	RW	Unsigned32	x	x	x	x	x	x
	1	Save all Parameters			x	x	x	x	x	x
	2	Save Communication Parameters			x	x	x	x	x	x
	3	Save Application Parameters			x	x	x	x	x	x
	4	Save Manufacturer Parameters			x	x	x	x	x	x
1011		Restore Default Parameters	RW	Unsigned32	x	x	x	x	x	x
	1	Restore AllDefault Parameters			x	x	x	x	x	x

Sendix 58x8, 70x8, 71x8											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
	2	Restore Communication Parameters			x	x	x	x	x	x
	3	Restore Application Parameters			x	x	x	x	x	x
	4	Restore Manufacturer Parameters			x	x	x	x	x	x
1012		COB-ID Time Stamp	RW	Unsigned32	0x100	0x100	0x100	0x100	0x100	0x100
1013		High Resolution Time Stamp	RW	Unsigned32	0x0	0x0	x	x	x	x
1014		COB-ID Emcy	RO	Unsigned32	x	x	x	x	x	x
1015		Inhibit Time Emcy	RW	Unsigned32	x	x	x	x	x	x
1016		Consumer Heartbeat Time	RW	Unsigned32	0x0	0x0	no	no	no	no
	1	Consumer Heartbeat Time_1			0x0	0x0	no	no	no	no
	2	Consumer Heartbeat Time_2			0x0	0x0	no	no	no	no
1017		Producer Heartbeat Time	RW	Unsigned16	0x0	0x0	0x0	0x0	0x0	0x0

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
1018		Identity Object	RO	PDOComPar	x	x	x	x	x	x
	1	Vencor ID			0x13	0x13	0x13	0x13	0x13	0x13
	2	Product Code			x	x	x	x	x	x
	3	Revision Number			x	x	x	x	x	x
	4	Serial Number			x	x	x	x	x	x
1029		Error Behaviour	RW	Unsigned8	x	x	x	x	x	x
	1	Communication Error			0x0	0x0	0x0	0x0	0x0	0x0
	2	Device Specific Error			0x1	0x1	0x1	0x1	0x1	0x1
	3	Manufacturer Error			0x1	0x1	0x1	0x1	0x1	0x1
1800		TxPDO1 Communication Parameter	RW	PDOComPar	x	x	x	x	x	x
	1	COB-ID			0x180 + node number	0x180 + node number	0x180 + node number	0x180 + node number	0x180 + node number	0x180 + node number
	2	Transmission Type			0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	3	InhibitTime			0x0	0x0	0x0	0x0	0x0	0x0
	5	Event Timer			0x0	0x0	0x0	0x0	0x0	0x0
1801		TxPDO2 Communication Parameter	RW	PDOComPar	x	x	x	x	x	x

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
	1	COB-ID			0x280 + node number	0x280 + node number	0x280 + node number	0x280 + node number	0x280 + node number	0x280 + node number
	2	Transmission Type			0x01	0x01	0x01	0x01	0x01	0x01
	3	InhibitTime			0x0	0x0	0x0	0x0	0x0	0x0
	5	Event Timer			0x0	0x0	0x0	0x0	0x0	0x0
1802		TxPDO3 Communication Parameter	RW	PDOComPar	x	x	x	x	x	x
	1	COB-ID			0x380 + node number	0x380 + node number	0x380 + node number	0x380 + node number	0x380 + node number	0x380 + node number
	2	Transmission Type			0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	3	InhibitTime			0x0	0x0	0x0	0x0	0x0	0x0
	5	Event Timer			0x0	0x0	0x0	0x0	0x0	0x0
1803		TxPDO4 Communication Parameter	RW	PDOComPar	no	no	no	x	no	x
	1	COB-ID			no	no	no	x	no	x
	2	Transmission Type			no	no	no	x	no	x
	3	Inhibit Time			no	no	no	x	no	x
	5	Event Timer			no	no	no	x	no	x

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
1804		TxPDO5 Communication Parameter	RW	PDOComPar	no	no	no	no	no	no
	1	COB-ID			no	no	no	no	no	no
	2	Transmission Type			no	no	no	no	no	no
	3	Inhibit Time			no	no	no	no	no	no
	5	Event Timer			no	no	no	no	no	no
1A00		TPDO1 Mapping	RW	PDOMapping	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020
1A01		TPDO2 Mapping	RW	PDOMapping	0x 60040020	0x 60040020	0x 60300110	0x 60300110	0x 60300110	0x 60300110
1A02		TPDO3 Mapping	RW	PDOMapping	0x 60300110	0x 60300110	0x 60040020	0x 60040020	0x 60040020	0x 60040020
1A03		TPDO4 Mapping	RW	PDOMapping	no	no	no	x	no	x
1A04		TPDO5 Mapping	RW	PDOMapping	no	no	no	no	no	no
1F51		Program Control			no	no	no	no	no	no
1F80		NMT Startup	RW		no	no	no	no	no	no

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
1000		Device Type	CONST	Unsigned32	x	x	x	x	0x 000B0196	0x 000B0196	0x 00020196
1001		Error Register	RO	Unsigned8	x	x	x	x	0x00	0x00	x
1002		Manufacturer status	RO	Unsigned32							x
1003		Predefined error field	RO	Unsigned32	x	x	x	x	0x00	0x00	x
1005		COB-ID SYNC	RW	Unsigned32	0x80	0x80	0x80	0x80	0x80	0x80	x
1006		ComCycle Period	RW	Unsigned32	x	x	x	x	x	x	x
1007		SyncWindowLen	RW	Unsigned32	x	x	x	x	x	x	no
1008		Manufacturer DeviceName	CONST	visible string	x	x	x	x	x	x	F58X8M
1009		Manufacturer Hardware Version	CONST	visible string	x	x	x	x	x	x	x
100A		Manufacturer Software Version	CONST	visible string	x	x	x	x	x	x	x
100C		GuardTime	RW	Unsigned16	0x0	0x0	x	x	no	no	no
100D		LifeTime Factor	RW	Unsigned8	0x0	0x0	x	x	no	no	no
1010		Store Parameters	RW	Unsigned32	x	x	x	x	0x 73617665	0x 73617665	0x 73617665
	1	Save all Parameters			x	x	x	x	x	x	x
	2	Save Communication Parameters			x	x	x	x	x	x	x
	3	Save Application Parameters			x	x	x	x	x	x	x

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
	4	Save Manufacturer Parameters			x	x	x	x	x	x	no
1011		RestoreDefault Parameters	RW	Unsigned32	x	x	x	x	0x 6c6f6164	0x 6c6f6164	0x 6c6f6164
	1	Restore AllDefault Parameters			x	x	x	x	x	x	x
	2	Restore Communication Parameters			x	x	x	x	x	x	x
	3	Restore Application Parameters			x	x	x	x	x	x	x
	4	Restore Manufacturer Parameters			x	x	x	x	x	x	no
1012		COB-ID Time Stamp	RW	Unsigned32	no	no	no	0x100	no	no	no
1013		High resolution time stamp	RW	Unsigned32	no	no	no	0x0	no	no	no
1014		COB-ID Emcy	RO	Unsigned32	x	x	0xBE	x	0xBF	0xBF	Node number+0x80
1015		Inhibit Time Emcy	RW	Unsigned32	x	x	0x0	x	x	x	0
1016		Consumer Heartbeat Time	RW	Unsigned32	x	x	x	x	no	no	x
	1	Consumer Heartbeat Time_1			x	x	0x0	x	no	no	0

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
	2	Consumer Heartbeat Time_2			x	x	0x0	x	no	no	0
1017		Producer Heartbeat Time	RW	Unsigned16	0x0	0x0	0x0	0x0	0x0	0x0	0
1018		IdentityObject	RO	PDOComPar	x	x	x	x	x	x	x
	1	Vencor ID			0x13	0x13	0x13	0x13	0x13	0x13	0x13
	2	Product Code			x	x	x	x	x	x	0x 00219000
	3	Revision Number			x	x	x	x	x	x	x
	4	Serial Number			x	x	x	x	x	x	x
1029		Error Behaviour	RW	Unsigned8	x	x	x	x	x	x	x
	1	Communication Error			0x0	0x0	0x0	0x0	0x0	0x0	0x0
	2	Device Specific Error			0x1	0x1	0x0	0x1	0x1	0x1	no
	3	Manufacturer Error			0x1	0x1	0x0	0x1	0x1	0x1	no
1800		TxPDO1 Communication Parameter	RW	PDOComPar	x	x	x	x	x	x	x
	1	COB-ID			0x180 + node number	0x180 + node number	0x180 + node number	0x180 + node number	Node number+ 0x40000180	Node number+ 0x40000180	Node number+ 0x40000180
	2	TransmissionType			0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	3	Inhibit Time			0x0	0x0	0x0	0x0	0x0	0x0	0x0
	5	Event Timer			0x0	0x0	0x0	0x0	0x0	0x0	0x0

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
1801		TxPDO2 Communication Parameter	RW	PDOComPar	x	x	x	x	x	x	x
	1	COB-ID			0x280 + node number	0x280 + node number	0x280 + node number	0x280 + node number	Node number+ 0x40000280	Node number+ 0x40000280	Node number+ 0x40000280
	2	TransmissionType			0x01	0x01	0x01	0x01	0x01	0x01	0x01
	3	Inhibit Time			0x0	0x0	0x0	0x0	0x0	0x0	0x0
	5	Event Timer			0x0	0x0	0x0	0x0	0x0	0x0	0x0
1802		TxPDO3 Communication Parameter	RW	PDOComPar	x	x	x	x	x	x	x
	1	COB-ID			0x380 + node number	0x380 + node number	0x380 + node number	0x380 + node number	Node number+ 0x40000380	Node number+ 0x40000380	Node number+ 0x40000380
	2	TransmissionType			0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
	3	Inhibit Time			0x0	0x0	0x0	0x0	0x0	0x0	0x0
	5	Event Timer			0x0	0x0	0x0	0x0	0x0	0x0	0x0
1803		TxPDO4 Communication Parameter	RW	PDOComPar	x	x	x	no	no	no	x

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
	1	COB-ID			0x480 + node number	0x480 + node number	0x480 + node number	no	no	no	Node number+ 0x40000480
	2	TransmissionType			0xFF	0xFF	0xFF	no	no	no	0xFF
	3	InhibitTime			0x0	0x0	0x0	no	no	no	0x0
	5	Event Timer			0x0	0x0	0x0	no	no	no	0x0
1804		TxPDO5 Communication Parameter	RW	PDOComPar	x	no	no	no	no	no	no
	1	COB-ID			x	no	no	no	no	no	no
	2	TransmissionType			x	no	no	no	no	no	no
	3	Inhibit Time			x	no	no	no	no	no	no
	5	Event Timer			x	no	no	no	no	no	no
1A00		TPDO1 Mapping	RW	PDOMapping	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020
1A01		TPDO2 Mapping	RW	PDOMapping	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020	0x 60040020
1A02		TPDO3 Mapping	RW	PDOMapping	0x 60300110	0x 60300110	0x 60300110	0x 60300110	0x 60300120	0x 60300120	0
1A03		TPDO4 Mapping	RW	PDOMapping	0x 21600020 0x 21600010	0x 21600020 0x 21600010	0x 21600020 0x 21600010		no	no	0
1A04		TPDO5 Mapping	RW	PDOMapping	x	x	no	no	no	no	no

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
1F51		Program Control			no	no	no	no	x	x	x
1F80		NMT Startup	RW		no	no	no	no	no	no	0x0

5.4.3 Manufacturer-Specific Objects

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
2100		Baud Rate	RW	Unsigned8	x	0x5	0x5	x	x	x
2101		Node Number	RW	Unsigned8	x	0x3F	0x3F	x	x	x
2102		CAN Bus Termination	RW	Unsigned8	x	0x00	0x00	x	x	x
2103		Firmware Flash Version	RO	Unsigned16	x	x	x	x	x	x
2105		Save All Bus Parameters	RW	Unsigned32	x	0x 65766173	0x 65766173	0x 65766173	0x 65766173	0x 65766173
2110		Sensor Configuration Structure	RO	Unsigned8		x	x	x	x	x
2120		Sensor Test Data	RW	Unsigned8	x	x	x	x	x	x
2125		Battery Voltage	RO	Unsigned16						
2126		Internal Chip Temperature								
2130		Encoder Measuring Steps	RW	Unsigned16	x	x	x	x	x	x
	1	Speed Calculation Multiplier						0xA	0xA	0xA

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
	2	Speed Calculation Divisor						0xA	0xA	0xA
	3	Speed average value						0xA	0xA	0xA
2140		Customer Memory	RW	Unsigned32	x	x	x	x	x	x
2150		Temperature History	RO	Unsigned16	x	x	x	x	x	x
2160		Position Raw Data	RO	Unsigned32		no	no	no	x	no
2161		Inv. Position Raw Data Value	RO	Unsigned32		no	no	no	no	no
2162		Raw Position CRC Value	RO	Unsigned16	x	no	no	no	no	no

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
2100		Baud Rate	RW	Unsigned8	0x5	0x5	0x5	0x5	0xFF	0xFF	0x05
2101		Node number	RW	Unsigned8	0x3F	0x3F	0x3F	0x3F	0xFF	0xFF	0x3F
2102		CAN bus Termination	RW	Unsigned8	0x1	x	0x1	0x01	0x01	0x01	0x01
2103		Firmware Flash Version	RO	Unsigned16		x	x	x	x	x	x
2105		Save All Bus Parameters	RW	Unsigned32	0x 65766173	0x 65766173	0x 65766173	0x 65766173	0x 65766173	0x 65766173	0x 65766173

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)											
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658(A) M3678(A)**	**M3668 M3688**	**M5858A M5868**	**F5888M**
2110		Sensor Configuration Structure	RO	Unsigned8	x	x	x	no	no	no	no
2120		Sensor Test Data	RW	Unsigned8	x	x	x	no	no	no	no
2125		Battery Voltage	RO	Unsigned16		x	x	no	no	no	x
2126		Internal Chip Temperature									x
2130		Encoder Measuring Steps	RW	Unsigned16	x	no	no	no	no	no	no
	1	Speed Calculation Multiplier									no
	2	Speed Calculation Divisor									no
	3	Speed Average Value									no
2140		Customer Memory	RW	Unsigned32	x	x	x	x	no	no	no
2150		Temperature History	RO	Unsigned16	no	no	no	no	x	x	no
2160		Position Raw Data	RO	Unsigned32	x	x	x	no	no	no	no
2161		Inv. Position Raw Data Value	RO	Unsigned32	no	x	x	no	no	no	no
2162		RawPosition CRC Value	RO	Unsigned16	no	x	x	no	x	x	no

5.4.4 Device-Specific Objects

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
6000		Operating Parameters	RW	Unsigned16	0x04	0x04	0x04	0x04	0x04	0x04
6001		Measuring Units Revolution	RW	Unsigned32	0x2000	0x2000	0x2000	0x2000	0x2000	0x2000
6002		Total Measuring Range	RW	Unsigned32	0x2000000	0x2000000	0x2000000	0x2000000	0x2000000	0x2000000
6003		Preset Value	RW	Unsigned32	0x0	0x0	0x0	0x0	0x0	0x0
6004		Position Value	RO	Unsigned32			x	x	x	x
600B		Position High Res Raw Data Value	RO	Unsigned64						
600C		Position Raw Data Value	RO	Unsigned32						
6030		Speed Value	RO	Unsigned16	x	x	x	x	x	x
6031		Speed Parameter	RO	integer16						
	1	Speed Source Selector								
	2	Speed Integration Time								
	3	Speed Calc Multiplier								
	4	Speed Calc Divisor								
6040		Acceleration Value	RO	Signed16	x	x	x	x	x	x
6041		Acceleration Parameter	RO	integer16						

Sendix 58x8, 70x8, 71x8										
Objects [hex]	**Subindex**	**Object name**	**R/W**	**Data type**	**5858 5878**	**5868 5888**	**7058 7078**	**7068 7088**	**7158 7178**	**7168 7188**
	1	Acceleration Source Selector								
	2	Acceleration Integration Time								
	3	Acceleration Calc Multiplier								
	4	Acceleration Calc Divisor								
6050		Jerk Value	RO	integer16	no	no	x	x	x	x
6200		Cyclic Timer	RW	Unsigned16	0x0	0x0	0x0	0x0	0x0	0x0
6400		Work area state	RO	Unsigned8	x	x	x	x	x	x
6401		Work area low limit	RW	Unsigned32	0x0	0x0	0x0	0x0	0x0	0x0
6402		Work area high limit	RW	Unsigned32	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
6500		Operating status	RO	Unsigned16	x	x	x	x	x	x
6501		Singleturn resolution	RO	Unsigned32	x	x	x	x	x	x
6502		Number of distinguishable revolutions	RO	Unsigned16	x	x	x	x	x	x
6503		Alarms	RO	Unsigned16	x	x	x	x	x	x
6504		Supported Alarms	RO	Unsigned16	x	x	x	x	x	x
6505		Warnings	RO	Unsigned16	x	x	x	x	x	x
6506		Supported Warnings	RO	Unsigned16	x	x	x	x	x	x
6507		Profile and SW Version	RO	Unsigned32	x	x	x	x	x	x

Sendix 58x8, 70x8, 71x8

Objects [hex]	Subindex	Object name	R/W	Data type	5858 5878	5868 5888	7058 7078	7068 7088	7158 7178	7168 7188
6508		Operating time	RO	Unsigned32	x	x	x	x	x	x
6509		Offset Value	RO	Signed32	x	x	x	x	x	x
650A		Module Identification	RO	Signed32	x	x	x	x	x	x
650B		Serial Number	RO	Unsigned32	x	x	x	x	x	x
650D		Absolute Accuracy	RO	Unsigned8	no	no	no	no	no	no
650E		Device Capability	RO	Unsigned32	no	no	no	no	no	no

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)

Objects [hex]	Subindex	Object name	R/W	Data type	F3658 F3678	F3668 F3688 USF	F5868 F5888	M3658 (A) M3678 (A)	M3668 M3688	M5858A	M5868	F5888M
6000		Operating Parameters	RW	Unsigned16	0x00	0x00	0x04	0x00	0x00	0x00	0x00	0x04
6001		Measuring Units Revolution	RW	Unsigned32	0x10000	0x10000	x	0x4000	0x4000	0x4000	0x4000	0x2000
6002		Total Measuring Range	RW	Unsigned32	0x 10000	0x 100000000	x	0x4000	0x 10000000	0x4000	0x 10000000	0x 2000000
6003		Preset Value	RW	Unsigned32	0x0	0x0	x	0x00	0x00	0x00	0x00	0x00
6004		Position Value	RO	Unsigned32	x	x	x	x	x	x	x	x
600B		Position HighRes Raw Data Value	RO	Unsigned64					x	x	x	x

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)												
Objects [hex]	**Sub index**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658 (A) M3678 (A)**	**M3668 M3688**	**M5858A**	**M5868**	**F5888M**
600C		Position Raw Data Value	RO	Unsigned32					x	x	x	x
6030		Speed Value	RO	Unsigned16	x	x	x	x	x	x	x	x
6031		Speed Parameter	RO	integer16		x	x	no	x	x	x	x
	1	Speed Source Selector			0x02	0x02			0x02	0x02	0x02	0x02
	2	Speed Integration Time			0x64	0x64			0x100	0x100	0x100	0x100
	3	Speed Calc Multiplier			0x01	0x01			0x01	0x01	0x01	0x01
	4	Speed Calc Divisor			0x01	0x01			0x01	0x01	0x01	0x01
6040		Acceleration Value	RO	Signed16	x	x	x	no	x	x	x	x
6041		Acceleration Parameter	RO	integer16					x	x	x	x
	1	Acceleration Source Selector							0x02	0x02	0x02	0x02
	2	Acceleration Integration Time							0x100	0x100	0x100	0x100
	3	Acceleration Calc Multiplier							0x01	0x01	0x01	0x01
	4	Acceleration Calc Divisor							0x01	0x01	0x01	0x01
6050		Jerk Value	RO	integer16	no	no	no	x	no	no	no	no

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)												
Objects [hex]	**Sub index**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658 (A) M3678 (A)**	**M3668 M3688**	**M5858A**	**M5868**	**F5888M**
6200		Cyclic Timer	RW	Unsigned16	0x0	0x0	0x0	0x0	0x0	0x0	0x0	no
6400		Work area state	RO	Unsigned8	x	x	x	x	0x0	0x0	0x0	x
6401		Work area low limit	RW	Unsigned32	0x0	0x0	0x0	0x0	0x0	0x0	0x0	x
6402		Work area high limit	RW	Unsigned32	0x 10000	0x 100000000	x	0x4000	0x 10000000	0x1 0000000	0x1 0000000	x
6500		Operating Status	RO	Unsigned16	x	x	x	x	0x0	0x0	0x0	x
6501		Singleturn Resolution	RO	Unsigned32	x	x	x	x	0x4000	0x4000	0x4000	0x80000
6502		Number of distinguishable revolutions	RO	Unsigned16	x	x	x	x	0x 20000000	x	0x 20000000	0x 1000000
6503		Alarms	RO	Unsigned16	x	x	x	x	0x0	0x0	0x0	x
6504		Supported Alarms	RO	Unsigned16	x	x	x	x	0x8001	0x8001	0x8001	0x0001
6505		Warnings	RO	Unsigned16	x	x	x	x	0x00	0x00	0x00	x
6506		Supported Warnings	RO	Unsigned16	x	x	x	x	0xC005	0xC005	0xC005	0x00D6
6507		Profile and SW Version	RO	Unsigned32	x	x	x	x	0x 01060400	0x 01060400	0x 01060400	0x 01000302
6508		Operating Time	RO	Unsigned32	x	x	x	x	x	x	x	x
6509		Offset Value	RO	signed32	x	x	x	x	x	x	x	x
650A		Module Identification	RO	signed32	x	x	x	x	x	x	x	x
650B		Serial Number	RO	Unsigned32	x	x	x	x	x	x	x	x
650D		Absolute Accuracy	RO	Unsigned8	no	x	x	no	0x0A	0x0A	0x0A	no

Sendix F36x8, F58x8, F5888M, M36x8(A), M58x8(A)												
Objects [hex]	**Sub index**	**Object name**	**R/W**	**Data type**	**F3658 F3678**	**F3668 F3688 USF**	**F5868 F5888**	**M3658 (A) M3678 (A)**	**M3668 M3688**	**M5858A**	**M5868**	**F5888M**
650E		Device Capability	RO	Unsigned32	no	3	x	no	0x02	0x02	0x02	x

5.5 Description of the Objects

VAR	Variable
ARRAY	Array of variables
RW	Read/Write
RO	Read only
Const	Constant
Name	Object name
M/O	Mandatory or optional

5.5.1 Objects not mentioned

All objects not mentioned are used for additional information and can be found in the respective CANopen profile.

5.5.2 Object 0x1000 - Device Type

Indicates the device type. Numbers are assigned to every device type.

- 0x00010196 Singleturn encoder
- 0x00020196 Multiturn encoder
- 0x060001A1 Linear measuring system
- 0x0002019A Inclinometer

5.5.3 Object 0x1001 - Error Register

NOTICE	**Temperature error**
	From the time point of view, reading the temperature is an operation that should not be neglected. Therefore, in DC mode, the temperature is only read out continuously from the ASIC when it is part of the process data. In other words, when object 0x2120 is mapped. If DC mode is activated, but object 0x2120 is not mapped, object 0x2120 will show the correct temperature value immediately after switching on, but this value will not be updated in Operational mode. In this case, no occurring temperature error will be displayed in object 0x1001. In the case of FreeRun mode, the temperature is always updated with every bus cycle.

Object 0x1001 is the error register of the device. If an error occurs, it will be displayed, in the event of temperature overshoot or undershoot, directly by this register in bit 3. In this case, bit 0 (generic error) is always set too. The total error code is thus, in the event of inadmissible temperature overshoot / undershoot, 0x09.

In addition, an Emergency message is issued with code 0x4200.

In case of a position error or a "Commissioning diagnostic" error, bit 0 "generic error" and bit 5 "device profile specific error" are set first in this register. Whether position errors or "Commissioning diagnostic" errors have occurred can be found in object 0x6503. Refer to the description of object 0x6503.

5.5.4 Object 0x1008 - Device Name

Displays the device name. In the case of an encoder, the value "Kuebler Sendix Encoder" is displayed.

5.5.5 Object 0x1009 - Hardware version

Returns the hardware version.

5.5.6 Object 0x100A - Software version

Represents the constant value "Va.b.", with a and b representing respectively the numerical values of the major and minor firmware version.

5.5.7 Object 0x1010 - Save Parameters

Command SAVE under subindex 1h (save all parameters) saves the parameters in the non-volatile memory (EEPROM).

All communication objects, application objects and manufacturer-specific objects are saved under this subitem. This operation requires about 14 ms.

In order to prevent accidental saving, the command is only carried out when the code word string SAVE is entered in this subindex.

A read access to subindex 1h provides information about the memory functionality

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	10	10	01	73	61	76	65
Response	60	10	10	01	00	00	00	00

Byte 4: 0x73 (ASCII code for S)

Byte 5: 0x61 (ASCII code for A)

Byte 6: 0x76 (ASCII code for V)

Byte 7: 0x65 (ASCII code for E)

5.5.8 Object 0x1011 - Load factory settings

About the factory settings via object 0x1011 - "Restore default parameters"

The default values can be restored with a specific command. In order to prevent accidental loading of the default values, the command is only carried out when the code word string "LOAD" is entered in this subindex.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	11	10	01	6C	6F	61	64
Response	60	11	10	01	00	00	00	00

Byte 0: 6Ch (ASCII code for "l")

Byte 1: 6Fh (ASCII code for "o")

Byte 2: 61h (ASCII code for "a")

Byte 3: 64h (ASCII code for "d")

A double-click on the "Restore all parameters" opens a dialog as shown in figure 42. After inputting the value 0x64616F6C, which is the Hex signature of the word "load" according to ISO 8859, all user parameters are replaced with the ones from the non-volatile memory.

5.5.9 Object 0x1016 - Heartbeat Consumer Time

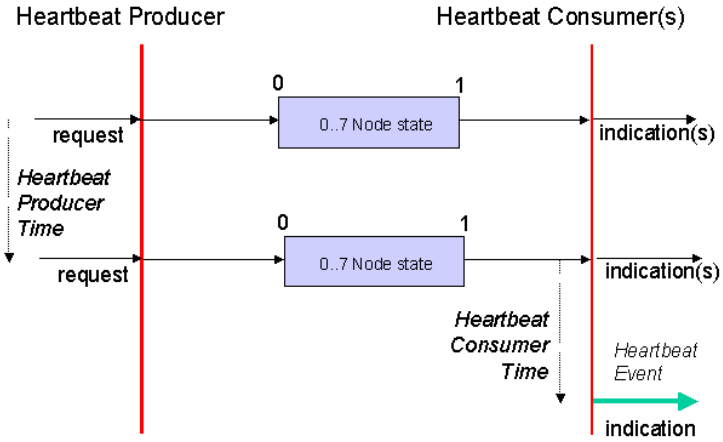
To activate this function, a valid node ID to be monitored must be entered with a corresponding time in object 0x1016, subindex 1 and/or in object 0x1016, subindex 2. The entered time should always be longer than the time of the heartbeat producer to be monitored. This function is active after a bootup cycle, provided the entered data has been saved (Store parameters object 1010h).

Monitoring is started after the first receipt of a heartbeat with the corresponding node address. If a time of 0 ms is entered, the function is inactive.
Valid settings: 1 ms to max. 65535 ms.

Bit	31 (MSB) to 24	23 to 16	15 to 0 (LSB)
Value	0x00 (reserved)	Address of monitored CANopen master	Monitoring time (ms)

2 nodes with node ID and corresponding time setting are supported.

Entries with different times for a node address and node address changes without previously erasing the object with zero entries are answered with an abort code 0604 - 0x0043 (General parameter incompatibility reason).



IMG-ID: 58386571

One or more heartbeat consumers receive the producer message. If, for any reason, this message is missing after the time set for the consumer has elapsed, a heartbeat event is generated.

The heartbeat consumer device activates an emergency message with an error code 8130 Lifeguard or Heartbeat error. According to the setting of the Error behavior object 0x1029, subindex 1, the consumer switches back to Pre-operational state. The behavior is thus determined by object 0x1029, Sub-Index 1 Communication error (0 = switch to Preop, 1 = no state change).

Bit	31 (MSB) to 24	23 to 16	15 to 0 (LSB)
Value	0x00 (reserved)	Node ID	Heartbeat time
Encoded as	-	Unsigned8	Unsigned16

In the event of an error, the following emergency message is generated:

CAN header	rtr	len	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x080 + Node-ID	0	8	30	81	11	00	00	00	00	00

Part of the error frame	Value	Description
Error code	8130	Lifeguard or Heartbeat error
Error register	11	Error register
Manufacturer-specific 1	00	ICLG Error register

A NMT reset node command of the consumer device or a new writing of object 1016h with data will re-activate the supervisor function. However, this activation of the function is only possible if saving has been performed previously with object 0x1010.

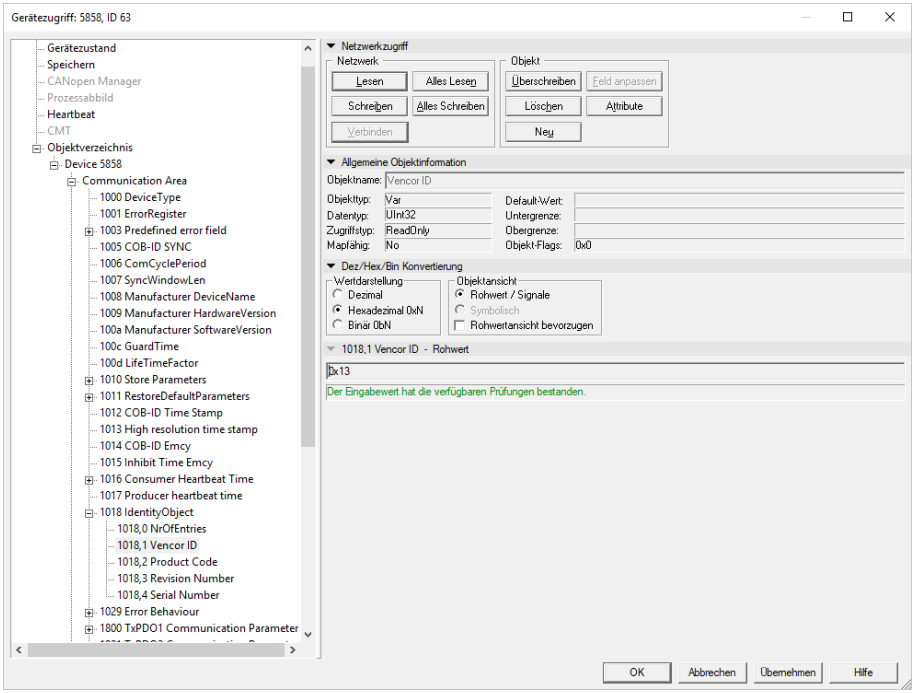
5.5.10 Object 0x1017 - Producer Heartbeat Time

This object defines the heartbeat cycle of the CAN device. If this function is not required, time must be set to 0. This function is activated with a time as from 1 ms (max. 65535 ms).

The originator of the request (heartbeat producer) transmits the message cyclically with the set time. The content of the data byte corresponds to the status of the CAN node (Pre-op, Operational, Stopped).

5.5.11 Object 0x1008 - Identification Object

The Identity object contains information about the manufacturer and the device:



IMG-ID: 58297739

Subindex	Designation	Contents
0x0	Supported Subindexes	4
0x1	Vendor ID	Vendor-ID (0x13) Fritz Kübler GmbH
0x2	Product Code	e. g. 0x58682001 CANopen encoder
0x3	Revision Number	Software revision number (e. g. 102) Subindex 4h: read only
0x4	Serial Number	8-digit serial number of the device

5.5.12 Object 0x1029 - Error Behavior

In case of a serious error, the device should switch automatically to Pre-operational mode. This object allows setting how the device has to behave in case of an error.

The following error classes are covered:

1029h, subindex 1 Communication error

- Bus off status of the CAN interface
- Life guarding event occurred
- Heartbeat monitoring failed

1029h, subindex 2 Device profile-specific

- Sensor error and controller error
- Temperature error

1029h, subindex 3 Manufacturer-specific

- Internal error

Byte 0
$2^7 \ldots 2^0$

The value of the object classes is set up as follows:

Value range 8 bits

0 = Pre-operational mode (only if the Operational mode was active previously)

1 = No mode change

2 = Stopped mode

3 … 127 = reserved

5.5.13 Object 0x1800 ... 0x1804 - TxPDO1-4

The TxPDO objects transmit the mappable PDOs. The following parameters can be set within the TxPDO:

- COB-ID
- Transmission type
- Inhibit time
- Event timer

COB-ID

Describes the identifier and thus also defines the priority on the bus.

If the COB-ID is to be edited, it must first be de-activated. To re-activate it, check whether the device supports RTR or not.

Devices that do not support RTR must be reset to valid with command 0x40000180 + node address.

Devices that support RTR are reset to valid with 0x180 + node address.

Bit	31	30	29	28 … 11	10 … 0
Contents	Valid	RTR	Frame	0x00	11 bit CAN-ID
Explanation	0: Valid 1: Invalid	0: RTR allowed 1: RTR not allowed	0: 11-bit CAN-ID 1: 29-bit CAN-ID	29-bit CAN-ID	11-bit CAN-ID

Transmission type

Indicates the used transmission type. See PDO Transmission Types [▶ 49].

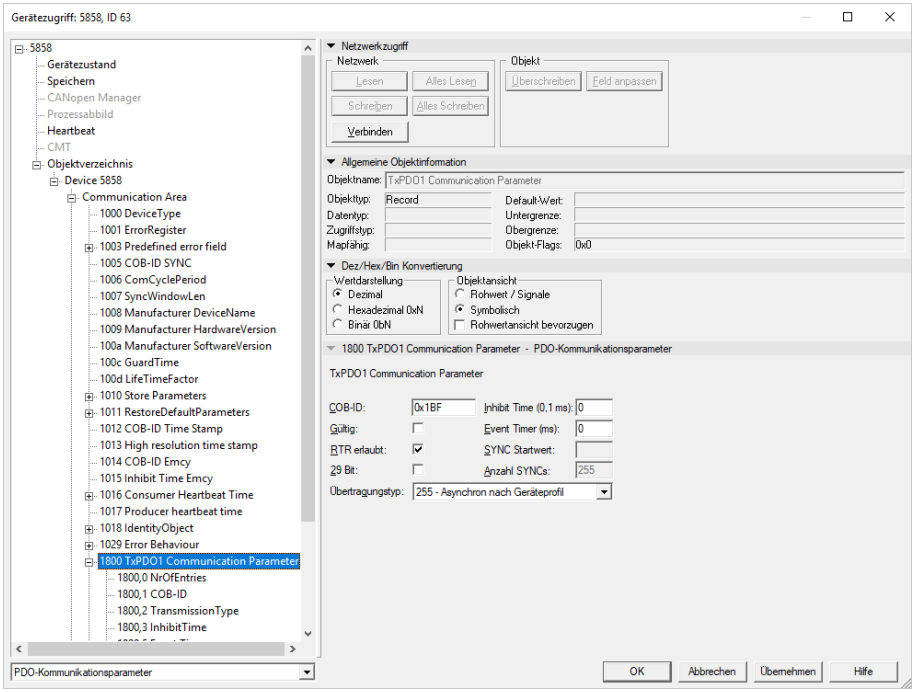
Inhibit time

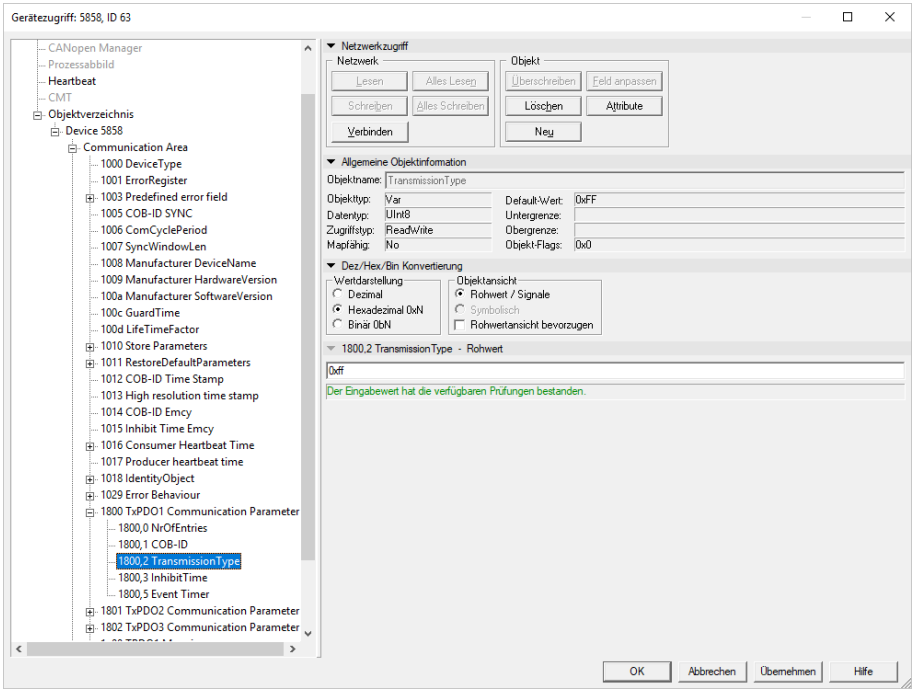
Defines the minimum pause interval between two events transmitted on the bus. This must be observed in particular for transmission type TT 255, as this may limit the bus traffic load.

Event timer

Determines the chronology of the cyclic transmission of an event and is used in combination with TT 254/255.

The value range for the event timer ranges from 0 ms ... 65,535 ms.





IMG-ID: 58275339

5.5.14 Object 0x1A00 … 0x1A04 - TPDO1-5 Mapping

The mapping object is defined in the object dictionary index 0x1A00 to 0x1A04. The subindexes represent the mapped objects. Theoretically, up to 64 objects can be mapped. The user can determine the objects to be transmitted for every available TPDO. See --- FEHLENDER LINK --- and PDO Mapping [▶ 117].

5.5.15 Object 0x1F80 - NMT Startup

The device starts automatically in Pre-operational mode.

If the device is to start directly in Operational mode, this can be set via object 0x1F80.

NOTICE	**Observe the implementation**
	With devices that do not support object 0x1F80, startup in Operational mode can be achieved with object 0x6000. Object 0x6000 - Operating Parameters [▶ 91]

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^{7}…2^{0}$	$2^{15}…2^{8}$	$2^{23}…2^{16}$	$2^{31}…2^{34}$

0x00 = NMT slave must be started by the NMT master.

0x02 = NMT slave starts autonomously in Operational mode.

Bit	Function	Bit = 0	Bit = 1	Default
3	Start in Operational	Disable	Enable	Disable

5.5.16 Object 0x2100 - Baud Rate

This object allows modifying the baud rate via software. As a standard, this value is set to 0xFF. Therefore, the setting with LSS shows a reconfigured node. If the value is set between 0 ... 8 and the parameter is saved using object 0x2105 Save All Bus Parameters, the device will boot with the modified baud rate at the following powering or Reset node and the currently set baud rate will be displayed.

NOTICE	**Observe the value of the baud rate.** With LSS, the value is exactly reversed with respect to the corresponding baud rate!

Data content:

Byte 0
$2^7 \ldots 2^0$

Value via object 0x2100	Value via LSS	Baud rate in kbit/s	Note
0	8	10	
1	7	20	
2	6	50	
3	5	100	No longer recommended by CiA
4	4	125	
5	3	250	
6	2	500	
7	1	800	
8	0	1000	Not supported by all devices

A new node number is only taken over at the following booting (reset/power-on) of the device or via a NMT Reset node command. All other settings in the objects table remain retained.

5.5.17 Object 0x2101 - Node Address

This object allows modifying the node address via software. As a standard, this value is set to 0xFF. Therefore, the setting with LSS shows a reconfigured node. If the value is set between 1 ... 127 and the parameter is saved using object 0x2105 Save All Bus Parameters, the device will boot with the modified node address at the following powering or Reset Node and the currently set address will be displayed.

NOTICE	**Observe the special case of node address 0**
	Node number 0 is reserved and shall not be used by any node

Data content:

Byte 0
$2^7 \ldots 2^0$

Values range 1 … 127 or 0x01 … 7F

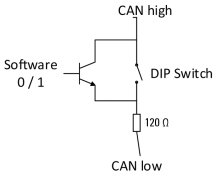
The resulting node numbers are in the range (1 … 127) or 0x01 … 7F. A new node number is only taken over at the following booting (reset/power-on) of the encoder or via a NMT Reset node command. All other settings in the objects table remain retained.

5.5.18 Object 0x2102 - CAN-bus Termination Off/On

This object allows switching the bus termination on via software. For devices without bus cover, this value is set to 1 as a standard: the encoder is terminated. For devices with bus cover and a fixed CAN connection, this value is set to 0 as a standard. The encoder is thus not terminated.

NOTICE	**Danger type and source**
	DIP switches are mounted on the printed circuit board also for encoders without bus cover. These switches are factory-preset.

If the DIP switch is in position "Off", the termination can be activated via CAN object 0x2102. This does not work if the DIP switch is in position "On".



IMG-ID: 9007199418941579

Data content:

Byte 0
$2^7 \ldots 2^0$

Values range 0 … 1

5.5.19 Object 0x2103 - Firmware Flash Version

This object indicates the current firmware version as a 16-bit hexadecimal value. This value is used for checking the current firmware version of the device.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Values range up to 0xFFFF

Example: 0xC47A current firmware

5.5.20 Object 0x2105 - Save All Bus Parameters

This parameter saves the desired bus parameters (objects 0x2100, 0x2101, 0x2102) permanently in the Flash memory. This object serves as an additional protection against an accidental change of the baud rate and node address.

Only targeted saving with parameter SAVE (hexadecimal 0x65766173) will save permanently the bus parameters baud rate, node address and termination. The settings become only active after a power off/on sequence or the NMT command RESET NODE.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{34}$

Values range: SAVE in hexadecimal 0x65766173

Command bytes: 23 05 21 01 73 61 76 65

Response: 60 05 21 01 00 00 00 00 for successful saving

5.5.21 Object 0x2120 - Upper / Lower Limit Temperature Position Sensor

This object 0x2120 subindex 2 and 3 allows setting the upper/lower temperature limit of the encoder as a 8-bit hexadecimal value.

These values are used as triggering thresholds for the emergency message.

Data content:

Byte 0
$2^7 \ldots 2^0$

Values range up to 0x00 … 0xFF

Example: 0x20 corresponds approximately to -32 °C

The following temperatures can be used as a reference:

- -20 °C corresponds to 0x2C
- 0 °C corresponds to 0x40
- 100 °C corresponds to 0xA4

If the temperature exceeds/sinks below these thresholds, an emergency message (see below) and an appropriate reaction are triggered.

Values range: 0x20 … 0xAC

Default setting: 0xA2 Temperature High Limit

5.5.22 Object 0x2125 - Battery voltage

This object shows the current voltage of the battery integrated in the encoder.

This object is only available for encoders with internal battery for the multiturn function. Otherwise the value 0 is issued.

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Values range up to 0x00…0xFFFF

Example: 0x168 corresponds to 3.60V

5.5.23 Object 0x2130 - Measuring Steps

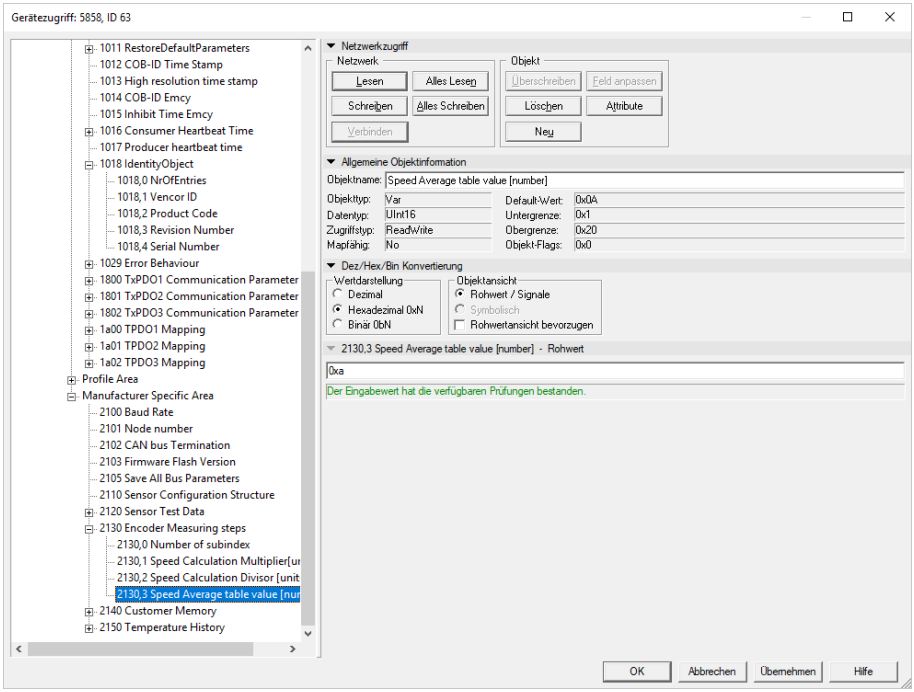
The rotational speed of the encoder shaft is determined as the value difference of two physically unscaled position values with a dynamic time interval of 1 ms, 10 ms or 100 ms. To adapt the speed determination to the actual application, 2 parameterizable objects are available to the user in the manufacturer-specific area. At high speeds, the integration time of the measurement can be reduced, so as to represent accordingly high dynamics.
The number of mean values affects especially the dynamics of the measurement and must be determined specifically for every application.

The accuracy of the measurement depends mainly on the following parameters:

- actual speed
- parameterized resolution / encoder rotation (object 0x6001)
- parameterized number of mean values (object 0x2130,3)
- temporal change of the speed (momentum)

Speed will be calculated according to the following formula:

Speed = (position change % integration time) x unit factor



IMG-ID: 58299659

A parameter under object 0x2130, subindex 2 Speed Measuring Divisor is available as a multiplier for an unit factor. Object 0x2130, subindex 3 Speed Average Value contains the number of measured values for the calculation of the moving speed average. The maximum values range is 1 … 32. Speed is output either as RPM or as a number of steps per second in object 0x6000 bit 13. Parameter object 0x2130, subindex 1 Speed Measuring Multiplier allows defining the periphery of a measuring wheel, e. g. to influence the speed.

NOTICE	**Observe the unit.** Only the speed output with the unit [unit/sec] can be influenced by object 0x2130. The output in RPM is not parameterizable.

5.5.24 Object 0x2140 - User Memory Area

These 4 parameters are a memory area for the user. 4 data words with max. 4 bytes can be saved. This area is not checked for content. Therefore, any format can be saved.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^{7}…2^{0}$	$2^{15}…2^{8}$	$2^{23}…2^{16}$	$2^{31}…2^{34}$

Values range: Numbers, alphanumeric

Default setting: 0

5.5.25 Object 0x2150 - Actual Temperature Position Sensor

This object indicates the current temperature inside of the device as a signed 16-bit hexadecimal value. This value allows checking the current temperature of the device. The temperature value can be mapped as a 16-bit value with the process data. It is updated there all 6 minutes. Accuracy is ± 2 °C. Measurement takes place inside of the encoder electronics.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Values range up to 00 … FFFFh

Example: 0x103 corresponds approximately to 25.9 °C

5.5.26 Object 0x2162 - Raw Value CRC16

A standard CRC16 is created based on current object 2160h Position Raw Data.

CRC-CCITT (CRC-16) x16 + x12 + x5 + 1 (polynomial 0x1021)

The implementation performs a polynomial division when the starting value used is 0x1021. Inverting the first n bits of the data stream corresponds to a polynomial division. A starting value different from 0000... should be preferred, as missing bits within leading zeros would not be detected otherwise in the data stream (as in a usual division, the leading zeros are not considered in a polynomial division).

Starting value (speed value) used here = 0x1021
The polynomial is displayed as a 16-bit hexadecimal value.
This value is used for checking the current position raw data of the device.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Values range up to 0xFFFF

Example: 0x4FA6 current CRC16 of the position raw data.

5.5.27 Object 0x6000 - Operating Parameters

This object must contain the functions for the code sequence, the startup of the diagnostic control and the control of the scaling function, as well as manufacturer-specific parameters.

Bit 0 - Code sequence

0 = increasing for clockwise rotation (cw)

1 = increasing for counter-clockwise rotation (ccw)

Bit 1 - Commissioning diagnostic control (type-dependent)

0 = disable

1 = enable

Bit 2 - Scaling

0 = disable

1 = enable (observe object 0x6001, 0x6002)

Bit 12 - Universal scaling function

0 = disable

1 = enable

NOTICE	**USF functionality**
	For devices with a firmware as from CiA406 v4.1.0, the USF functionality is always activated. Therefore, for these devices, bit 12 has a different assignment. For the devices M36 / M58 MT, Speed format is bit 12, USF is not supported.

Bit 13 - Speed format

0 = revolutions / minute

1 = units / second

Bit 14 - Startup mode

0 = bootup after Pre-operational

1 = bootup after Operational

Bit 15 – Event mode (type-dependent)

0 = output after TPDO 0x1800

1 = output at every position change

Bit	Function	Bit = 0	Bit = 1	Bit numbering
0	Code sequence	CW	CCW	0x0001
1	Commissioning diagnostic control	Disabled	Enabled	0x0002
2	Scaling on	Disabled	Enabled	0x0004
3	Counting direction	Forward	Backward	0x0008
4 … 11	Reserved			
12	Universal Scaling Function - USF	Disabled	Enabled	0x1000
13	Speed format	RPM	Units/sec	0x2000
14	Startup in Operational mode	Disabled	Enabled	0x4000
15	Reserved	Disabled	Enabled	0x8000

The following values apply to products as from CiA406 v4.1.0

Bit	Function	Bit = 0	Bit = 1	Bit numbering	Default
0	Code sequence	CW	CCW	0x0001	CW
1	Reserved				
2	Switch scaling on	Disable	Enable	0x0004	Disable
3..11	Reserved				
12	Speed format	RPM	Units/ sec	0x1000	RPM
13	Reserved				
14	Startup in Operational mode	Disable	Enable	0x4000	Disable
15	Reserved				

Example

You want to perform the following functions:

Code sequence increasing for ccw rotation	0x0001
Scaling function activated	0x0004
Universal Scaling Function - USF activated	0x1000
Operating mode after bootup	0x4000
	0x5005

The summary of the bit numbering (here 0x5005) is the data word that must be written in object 0x6000.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2B	00	60	00	05	50	00	00
Response	60	00	60	00	00	00	00	00

5.5.28 Object 0x6001 - Measuring units per revolution (MUR)

This parameter sets the desired resolution per revolution. The encoder calculates internally the corresponding scaling factor. The scaling factor MURF (by which the physical position value is multiplied) is calculated according to the following formula:
MURF = Measuring steps per revolution 0x6001 / phys. resolution singleturn 0x6501

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^{7}…2^{0}$	$2^{15}…2^{8}$	$2^{23}…2^{16}$	$2^{31}…2^{34}$

Values range: 4 ... maximum physical singleturn resolution

Only valid if scaling 0x6000 is activated.

When changing TMR/MUR, the system also checks the TMR/MUR ratio.

If TMR is set to a value that leads to an invalid ratio, an error message is returned and the new value is rejected. The old value remains active in the device: MUR <= TMR.

5.5.29 Object 0x6002 - Total number of measuring units (TMR)

This parameter sets the total number of measuring steps for singleturn and multiturn.
The maximum physical resolution is multiplied by a factor. The factor is always < 1.
After the scaled total position of the measuring steps, the encoder resets itself to zero.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{34}$

Values range: 4 ... maximum physical resolution ST+MT

NOTICE	**Special case TMR = 32 bits**
	If TMR is to be 32 bits, i. e. the maximum resolution is required, the data content of object 0x6002 = 0! This is a special case.

When changing TMR/MUR, the system also checks the TMR/MUR ratio.
If TMR is set to a value that leads to an invalid ratio, an error message is issued and the new value is rejected.. The old value remains active in the encoder. Only valid if scaling (0x6000 bit 2) is activated.

5.5.30 Object 0x6003 - Preset value

The position value of the encoder is set to the input preset value. This allows e.g. aligning the zero position of the encoder with the zero position of the machine.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{24}$

Values range: 0 ... maximum physical resolution ST+MT

When inputting the preset value, the systems checks automatically whether the point lies within the active scaling or within the total measuring range. Otherwise it rejects the input.

5.5.31 Object 0x6004 - Position value unscaled or scaled

The encoder returns the current position value (possibly multiplied by the scaling factor).

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{34}$

Values range: 0 ... maximum physical resolution ST+MT

The TMR/MUR ratio is active when scaling is active, otherwise the 32-bit raw position of the encoder is output.

5.5.32 Object 0x600B - Position Raw Value High-Resolution

In addition to object 0x6004, the position data can be output as high-resolution raw data. The data is transmitted as a 64-bit logically correct value. This value is used internally as calculation basis.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
$2^{7}…2^{0}$	$2^{15}…2^{8}$	$2^{23}…2^{16}$	$2^{31}…2^{24}$	$2^{39}…2^{32}$	$2^{47}…2^{40}$	$2^{55}…2^{48}$	$2^{63}…2^{56}$

Values range: 0 ... maximum physical resolution ST+MT

5.5.33 Object 0x600C - Position Raw Value

In addition to object 0x6004, the position data can be output as high-resolution raw data. The data is transmitted as a 64-bit logically correct value. This value is used internally as calculation basis. The encoder outputs the current original position value without scaling directly from the device.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^{7}…2^{0}$	$2^{15}…2^{8}$	$2^{23}…2^{16}$	$2^{31}…2^{24}$

Values range: 0 ... maximum physical resolution ST+MT

5.5.34 Object 0x6030 - Speed Value

The encoder returns the current calculated speed (with scaling factor) as a signed 32-bit value. Speed depends on the settings of object 6031h. These values influence the calculation and the result.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^{7}…2^{0}$	$2^{15}…2^{8}$	$2^{23}…2^{16}$	$2^{31}…2^{34}$

Values range: 0 ... +/- maximum speed in RPM (signed value)

In case of values > 12000 RPM, a warning message is emitted and the warning bit "Overspeed bit 0" is set in object Warnings 6505h. Parameters that also influence this object are mentioned in object 6031h.

5.5.35 Object 0x6031 - Speed Parameters

For products as from CANopen CiA406 V4.1.0, the speed is calculated with the following formula:

$$v_{6030_h} = \frac{X_{6004_h} - X_{XXXX_h}}{T_{6031\,02_h} \cdot 10^{-3}} \cdot \frac{C1_{6031\,03_h}}{C2_{6031\,04_h}}$$

IMG-ID: 163727883

A variable integration time of 1 … 100 ms can be entered under 0x6031, subindex 2 (T).

Subindex	Parameter	Areas	Default
Subindex 0	Number of Channels	4	4
Subindex 1	Speed Source Selector	2	2
Subindex 2	Integration Time Value	5 … 2000	100
Subindex 3	Calculation Multiplier	1	1
Subindex 4	Calculation Divisor	1	1

The Speed Source Selector 0x6031, subindex 1 is set as standard to 2 (0x600C Raw Position) and can be modified if necessary. A parameter is available under Object 0x6031,subindex 3 Speed Calculation Multiplier (C1) as a multiplier for a unit factor, or a divisor (C2) can be programmed under Object 0x6031, subindex 4.

The speed output can be influenced with a gear factor via object 0x6031, subindex 3/4. Speed is output either as RPM or as a number of steps per second, the output is set in Object 0x6000 bit 12.

5.5.36 Objekt 0x6040 - Acceleration value

The encoder returns the current calculated acceleration (with the correct sign) as a signed 32-bit value. Acceleration is calculated from the speed changes and therefore also depends indirectly on the settings of object 6031h. Regardless of this, the settings of Object 6041h are determining.. All settings of these values influence the calculation and the result.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{34}$

Values range: 0.... +/- maximum acceleration

Negative values indicate a negative acceleration (speed decreases).

5.5.37 Object 0x6041 - Acceleration Parameters

Subindex	Parameter	Areas	Default
Subindex 0	Number of Channels	4	4
Subindex 1	Speed Source Selector	2	2
Subindex 2	Integration Time Value	5 … 2000	100
Subindex 3	Calculation Multiplier	1	1
Subindex 4	Calculation Divisor	1	1

An average acceleration a is the change over time of speed v and can therefore be described formally from the derivative of speed according to time t, calculating here an average acceleration from the difference Δv of the speeds at 2 different moments Δt (t2-t1).

$$a = \frac{\Delta v}{\Delta t} \quad \text{oder / or} \quad a = \frac{v2 - v1}{t2 - t1}$$

IMG-ID: 18014398673654923

5.5.38 Object 0x6200 - Cycle time

Defines the cycle time with which the current position is transmitted by means of PDO 1 (see Object 1800h).
The timer-controlled output is active as soon as a cycle time >0 is entered.
This object is only present for the purpose of compatibility with older profile versions. Instead of this object, use the Event timer subindex (05h) in the 1st Transmit PDO.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Values range: 0 ... FFFFh (65535) gives the cycle time in milliseconds

5.5.39 Object 0x6400 - Working Area State Register 2 Values

This object contains the current status of the encoder position according to the programmed limits. The flags are set or reset according to the position of both limit values. Comparison with both limit values is performed in "real time" and can be used for real-time positioning or for the limit switch function.

Name	Bit	Value	Definition
Out of range	0	0_b	Position value between minimum and maximum value "Module identification" (Object $650A_h$)
		1_b	Minimum and maximum value ("Module identification", Object $650A_h$) is reached or exceeded
Range overflow	1	0_b	No range overflow
		1_b	Position value higher than the value in "work area high limit" (Object 6402_h)
Range underflow	2	0_b	No range underflow
		1_b	Position value lower than the value in "work area low limit" (Object 6401_h)

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

5.5.40 Object 0x6401 & Object 0x6402 - Upper / Lower Limit Working Area 2 Values

Object 6401h: Working Area Low Limit 2 values

Object 6402h: Working Area High Limit 2 Values

These two parameters set the working area. The status can be signaled within and outside this range by means of flag bytes (object 6400h Working Area State). These area markers can also be used as software limit switches.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{34}$

Values range: 1....maximum physical resolution 4294967295 (232) bits - 1 2 values are available both for the lower and for the upper range.

5.5.41 Object 0x6500 - Working Status

This object displays the status of the programmed settings of object 6000h.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Data content: see Object 0x6000 - Operating Parameters [▶ 91].

5.5.42 Object 0x6502 - Number of Multiturn Revolutions

This object returns the current number of revolutions. This value depends on the encoder type and can have any value from 4096 (12 bits up to 32 bits). This value only affects the number of revolutions. The resolution remains unchanged.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{34}$

Values range: 1 … 4.294.967.296 (32 bits)

5.5.43 Object 0x6503 - Alarms

In addition to the errors signaled by emergency messages, object 0x6503 provides further error messages. As long as the error is present, the corresponding error bit is set to 1.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Bit no.	Designation	Value = 0	Value = 1
Bit 0	Position error	Position value valid	Position error
…	Reserved		
Bit 15	Device error	no error	Error

In both cases, in case of occurrence of an alarm, an emergency message (ID=80h+node number) is transmitted simultaneously with error code 0x1000 (Generic error).

5.5.44 Object 0x6504 - Supported Alarms

This object indicates which alarm messages are supported by the encoder.

Bit	Designation	Bit = 0	Bit = 1
0	Position error	Disable	Enable
1..15	Reserved		

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

Values range see Object 0x6503 - Alarms [▶ 98].

A bit set to 1 means that the alarm message is supported.
Example: Bit 0 = 1 Position error message is supported

5.5.45 Object 0x6505 - Error messages

Warning messages indicate that tolerances of internal encoder parameters have been exceeded. When a warning message is issued, the measured value may still be valid, unlike the alarm messages or the emergency messages. The corresponding warning bit remains set to 1 as long as the tolerance is exceeded or the warning is present.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$

If bit 0 is active, an emergency message (ID=80h+node number) is transmitted simultaneously with error code 4200h (Device-specific).

Bit no.	Designation	Value = 0	Value = 1
Bit 0	Overspeed	none	Exceeded
Bit 1	Reserved		
Bit 2	Watchdog status	System OK	Reset performed
…	Reserved		
Bit 14	Temperature error	Temperature OK	Temperature exceeded
Bit 15	Internal memory error	ok	Error

If bit 15 is active, an emergency message (ID=80h+node number) is transmitted simultaneously with error code 5200h (Device hardware).

5.5.46 Object 0x6506 - Supported Error Messages

This object indicates which warning messages are supported by the encoder (see object 6505h).

Bit	Designation	Bit = 0	Bit = 1
0	Overspeed	Disable	Enable
1 … 13	Reserved		
14	Temperature error	Disable	Enable
15	Reserved		

Data content:

Byte 0	Byte 1
2^7… 2^0	2^{15}…2^8

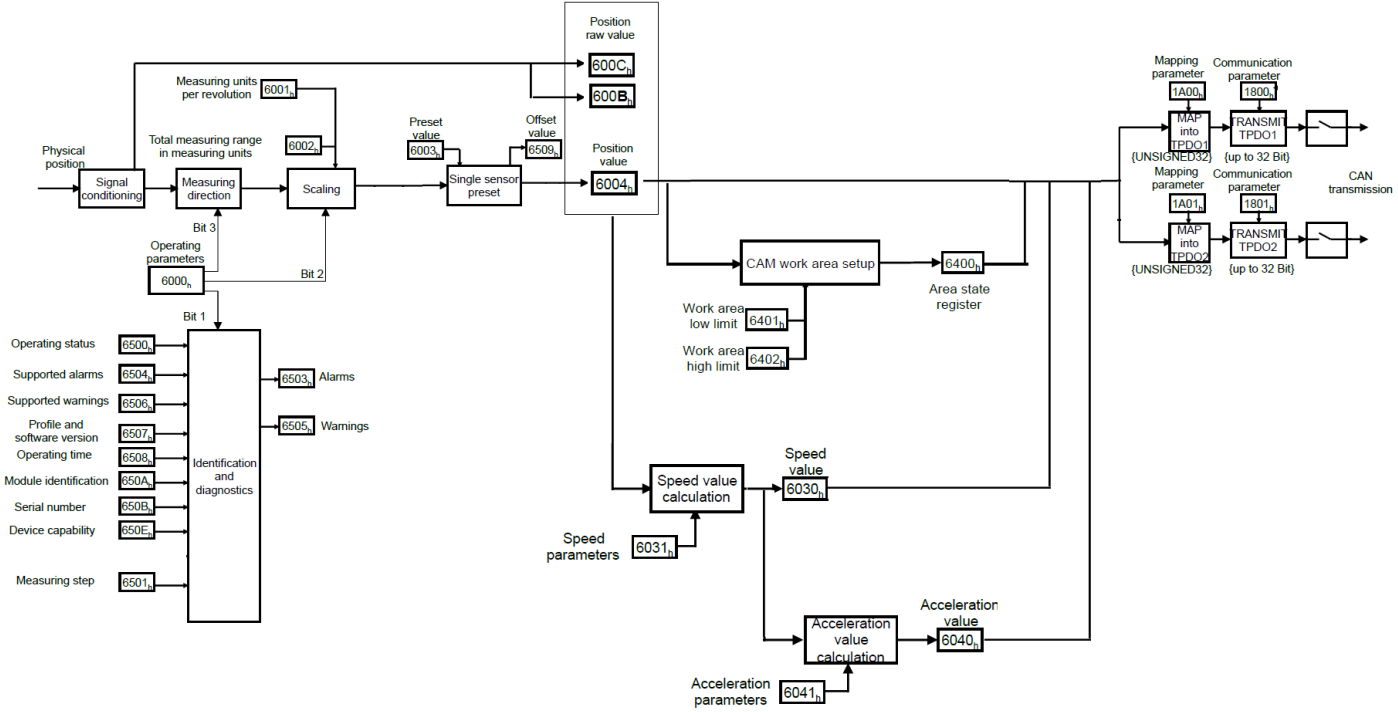
Values range see Object 0x6505 - Error messages [▶ 99].

A bit set to 1 means that the warning message is supported.

5.5.47 Object Process Map F58xxM

The position signal is conditioned immediately after the acquisition and it can thus be issued both as a raw value and as a scaled value taking the preset into account.

The speed and acceleration value, which can be transmitted per TPDO, is calculated from this value.

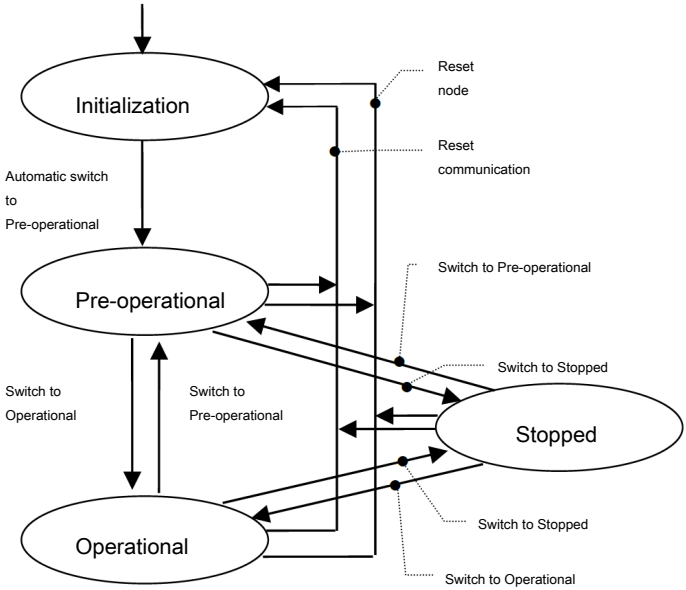


IMG-ID: 160208267

5.6 Network Management

The device supports the simplified network management (minimum boot up) defined in the profile for "minimum capability devices“.

The following status diagram according to DS 301 shows the various node statuses and the corresponding network commands (controlled by the network master via NMT services).



IMG-ID: 109579787

Initialization

After power supply is applied or after a reset, the device is in the original status Initialization. After performing the reset/initialization routines, the node switches automatically to the Pre-operational status. The LEDs display the momentary status.

Pre-Operational

The CAN node can now be addressed by SDO messages or NMT commands under the standard identifier. Then follows the programming of the encoder or communication parameters.

Operational

The node is active. Process values are supplied via the PDOs. All NMT commands can be evaluated.

Prepared or Stopped:

In this status, the node is no longer active, i.e. neither SDO nor PDO communication is possible any longer. The node can be set to Operational or Pre-operational status by means of NMT commands.

5.6.1 NMT Commands

All NMT commands are transferred as an unconfirmed NMT Object. Broadcast allows all participants to recognize the NMT commands.

An NMT Object is structured as follows:

Byte 0	Byte 1
2^7… 2^0	2^{15}… 2^8

COB-ID = 0

Byte 0 = Command byte

Byte 1 = Node number (e.g. 3F or 00 for all participants)

The COB ID of the NMT object is always 0
The node is addressed via the node number. With node number 0 all nodes are addressed.

Command byte	Description
0x01	Start_Remote_Node: Switch to Operational
0x02	Stop_Remote_Node: Switch to Prepared
0x80	Enter_Pre-Operational_State: Switch to Pre-operational
0x81	Reset_Node: Reset the node All parameters of the whole object dictionary are set to their respective default value.
0x82	Reset_Communication: Reset the communication Only the parameters in section Communication profile of the object dictionary are set to their respective default value.

5.7 Examples

5.7.1 Basic Parameterizing

Index 1010
Saving all bus objects with index 1010, subindex 01.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	10	10	01	73	61	76	65
Response	60	10	10	01	00	00	00	00

Index 1011
Loading the factory setting with index 1011, subindex 01.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	11	10	01	6C	6F	61	64
Response	60	11	10	01	00	00	00	00

Index 180x
Changing the transmission mode to Event mode (to 0xFF) with Index 1802, subindex 02.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2F	02	18	02	FF	00	00	00
Response	60	02	18	05	00	00	00	00

Changing the Event timer to 500 ms (0x1F4) with index 1802, subindex 05.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2B	02	18	05	F4	01	00	00
Response	60	02	18	05	00	00	00	00

Index 2100
Setting the baud rate (to 0x05) with index 2100, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2F	00	21	00	05	00	00	00
Response	60	00	21	00	00	00	00	00

0 = 10 kbit/s; 1 = 20 kbit/s; 2 = 50 kbit/s; 4 = 125 kbit/s; 5 = 250 kbit/s; 6 = 500 kbit/s;
8 = 1000 kbit/s

Index 2101
Setting the node address (to 0x3F) with index 2101, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2F	01	21	00	3F	00	00	00
Response	60	01	21	00	00	00	00	00

Index 2102
Switching off the terminating resistor (up to 0x00) with index 2102, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2F	02	21	00	00	00	00	00
Response	60	02	21	00	00	00	00	00

Termination on = 1
Termination off = 0

Index 2105
Saving all bus objects with index 2105, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	05	21	00	73	61	76	65
Response	60	05	21	00	00	00	00	00

Index 6000
Setting the ccw counting direction (to 0x01) with index 6000, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2B	00	60	00	01	00	00	00
Response	60	00	60	00	00	00	00	00

Index 6001
Setting MUR (to 0x1000) with index 6001, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	01	60	00	00	10	00	00
Response	60	01	60	00	00	00	00	00

Index 6002
Setting TMR (to 0x10000) with index 6002, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	02	60	00	00	00	01	00
Response	60	02	60	00	00	00	00	00

Index 6003
Setting the preset value (to 0x00) with index 6003, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	03	60	00	00	00	00	00
Response	60	03	60	00	00	00	00	00

Index 6200
Setting the Event timer (to 0xa) with index 6200, subindex 00.

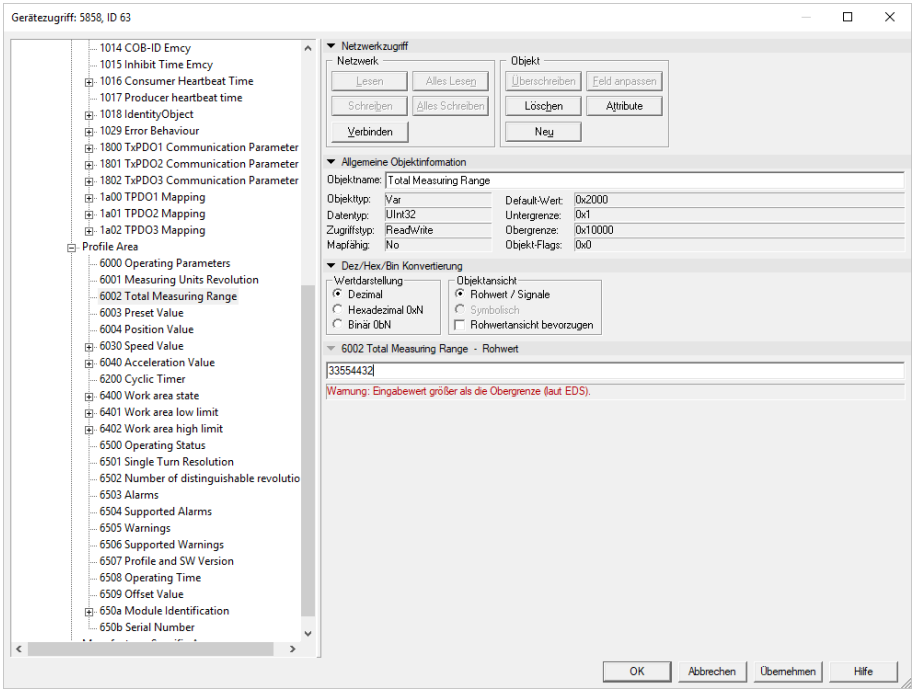
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2B	00	62	00	0A	00	00	00
Response	60	00	62	00	00	00	00	00

5.7.2 Parameterizing a Specific Application

Setting the objects

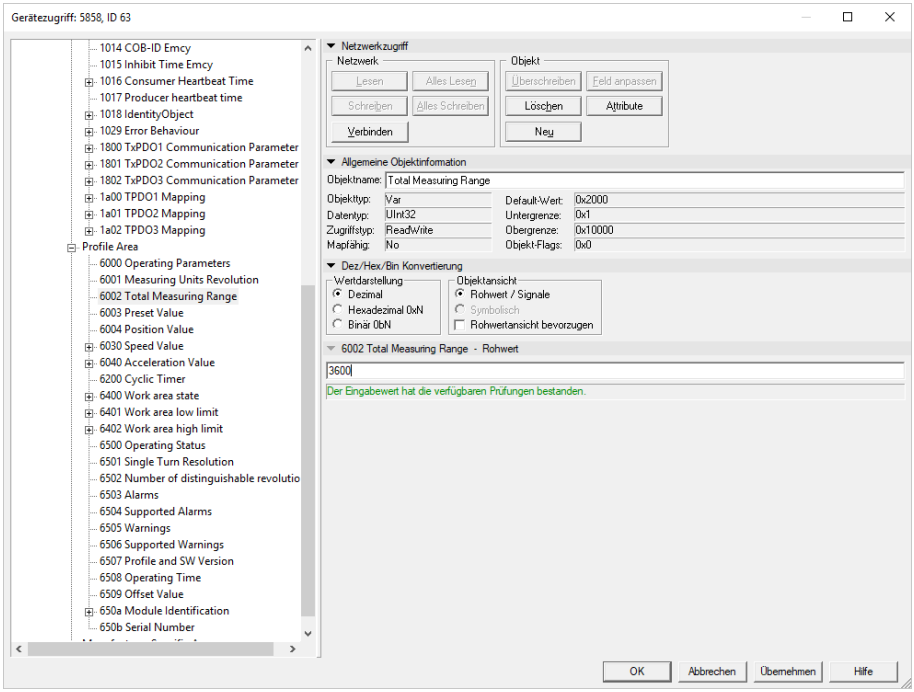
1. Limit the Total Measuring Range to 36000.
2. Set the Measuring Units per Revolution to 3600 steps per revolution.
3. Set the position value to 0.
4. Set TPDO1 (position) so that an event is sent with 10 ms.
5. Set TPDO2 (speed) so that an event is with at 20 ms.
6. Set the Transmission Type TPDO2 to 255 – asynchronous.
7. Reduce the Producer Heartbeat to 500 ms.
   -> The Work Area Limit is 1000 and 35000.
8. Save the new parameters in the non-volatile memory.

Limiting the Total Measuring Range to 36000.



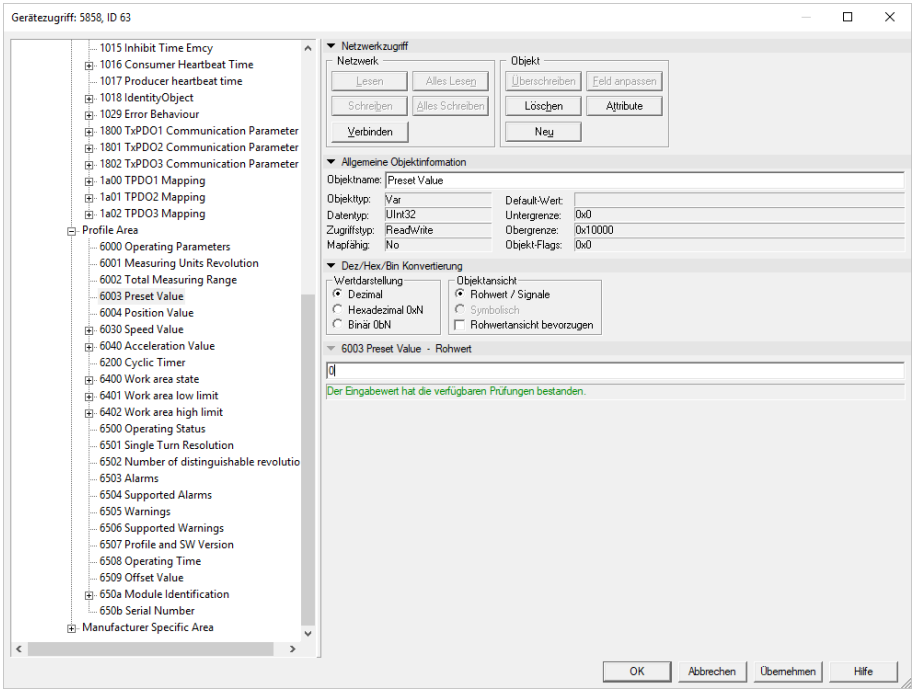
IMG-ID: 58315019

Limiting the Measuring Units per Revolution to 3600



IMG-ID: 58316939

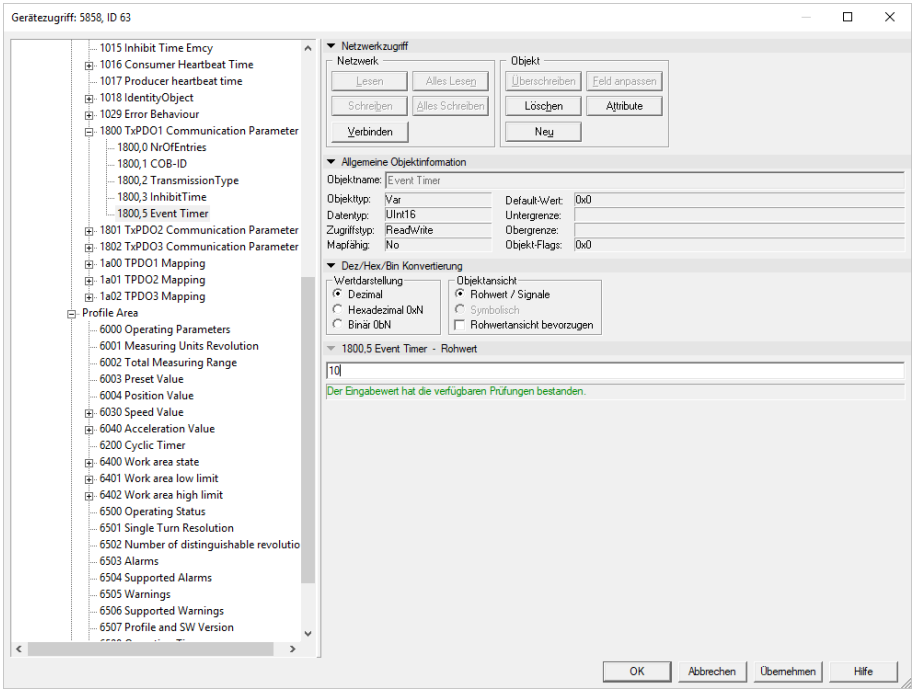
Setting the Preset Value to 0



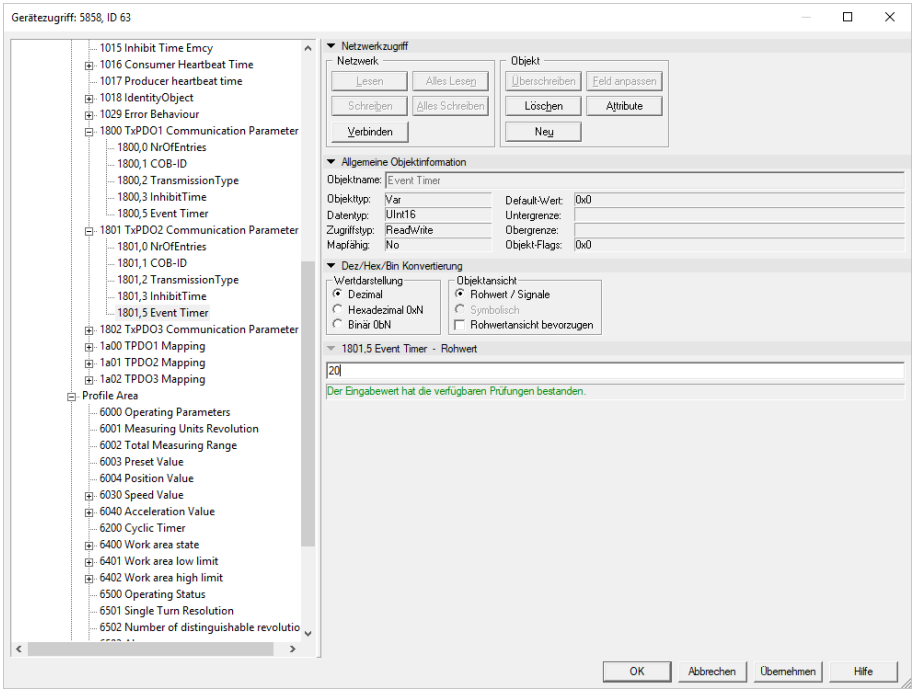
IMG-ID: 58318859

Setting the Transmit Parameters TPDO1 and TPDO2 values

Type 254 means that the event is triggered depending on the application, while number 255 depends on the device profile. In addition, a time-controlled event timer can be used for number 245/255. The value range for the timer ranges from 0 ms ... 65535 ms.

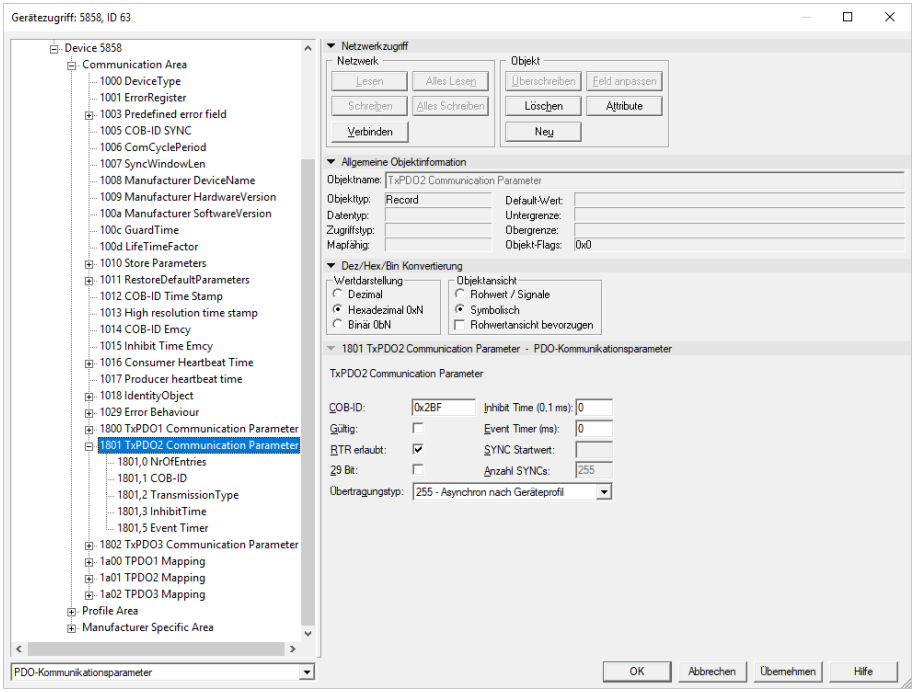


IMG-ID: 58320779



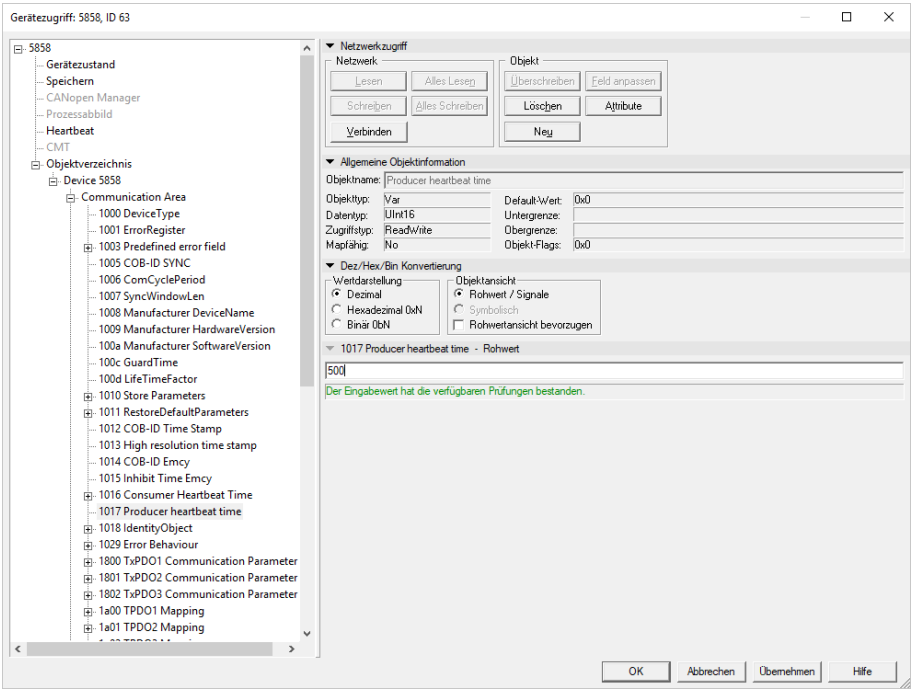
IMG-ID: 58277259

Setting the Transmission Type TPDO2 to 255



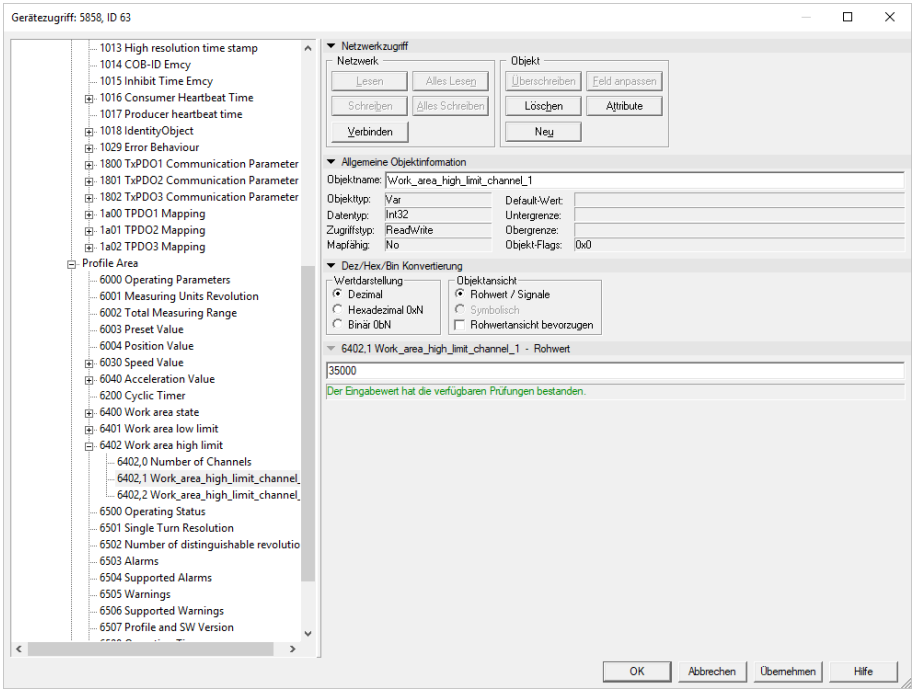
IMG-ID: 58311179

Changing the Producer Heartbeat Time to 500 ms



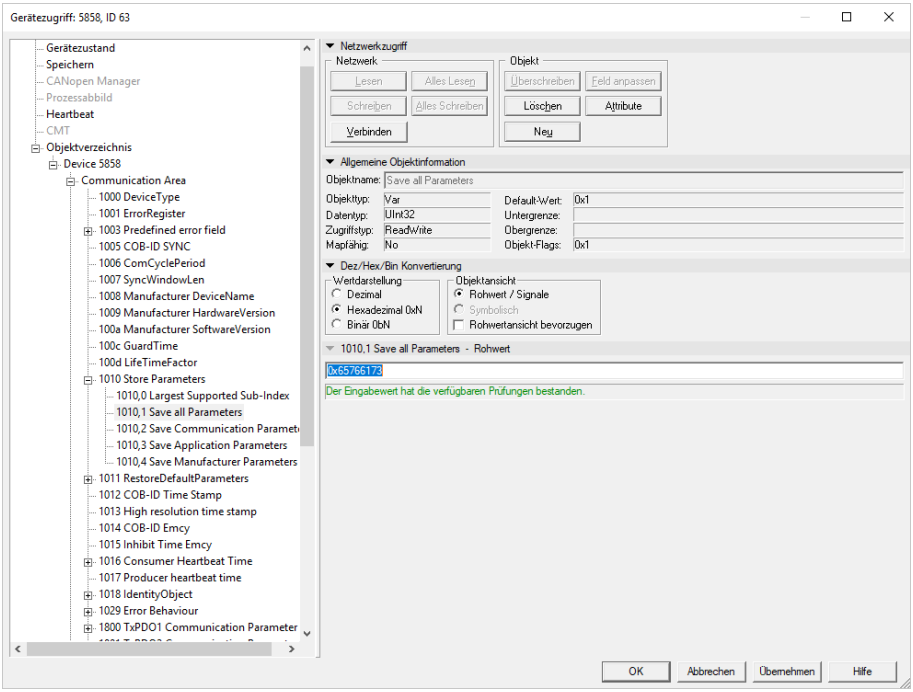
IMG-ID: 58291979

Setting the Work Area Low and High Limit values



IMG-ID: 58293899

Saving all modified parameters in the non-volatile memory - Store Parameters 0x1010



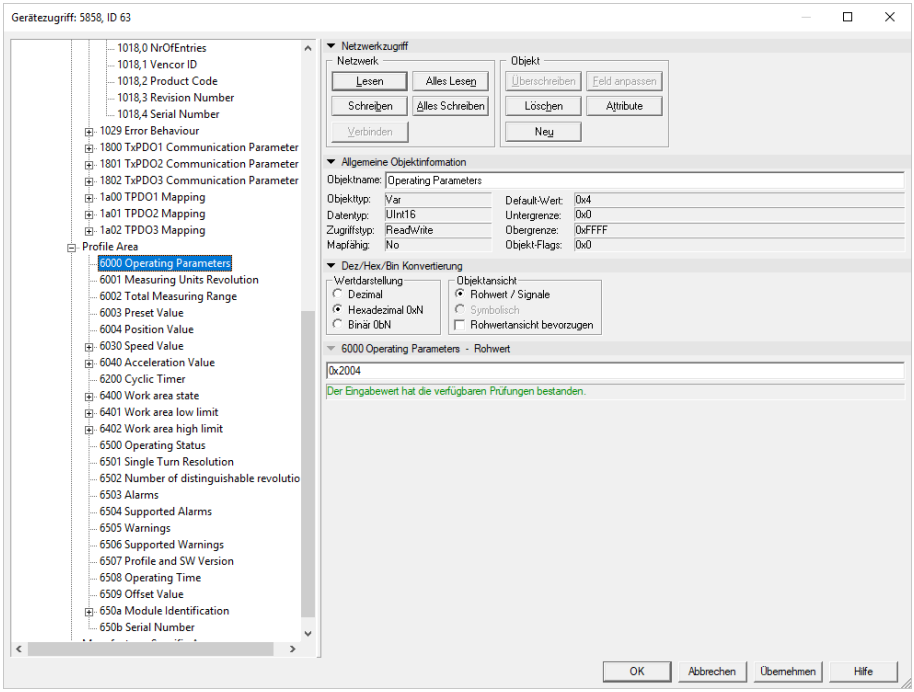
IMG-ID: 58295819

5.7.3 Parameterizing the Speed Output

- Speed display in units/sec
- Number of measured values used for the average value calculation 32
- Speed output factor 20

Output Speed Format: Units/sec

Bit 13 in object 0x6000 must be set to 1.



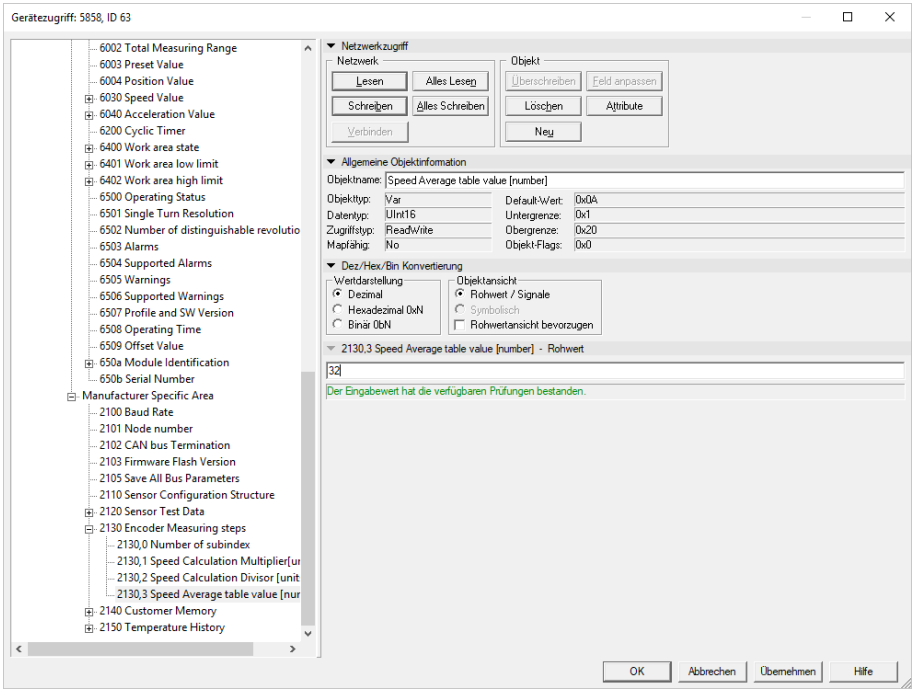
IMG-ID: 58301579

0x2004 means

- Bit 13 = 1 Units/sec
- Bit 2 = 1 Scaling enabled

Number of measured values used for the average value calculation: 32

Contains the number of measured values for calculating the moving speed average value.

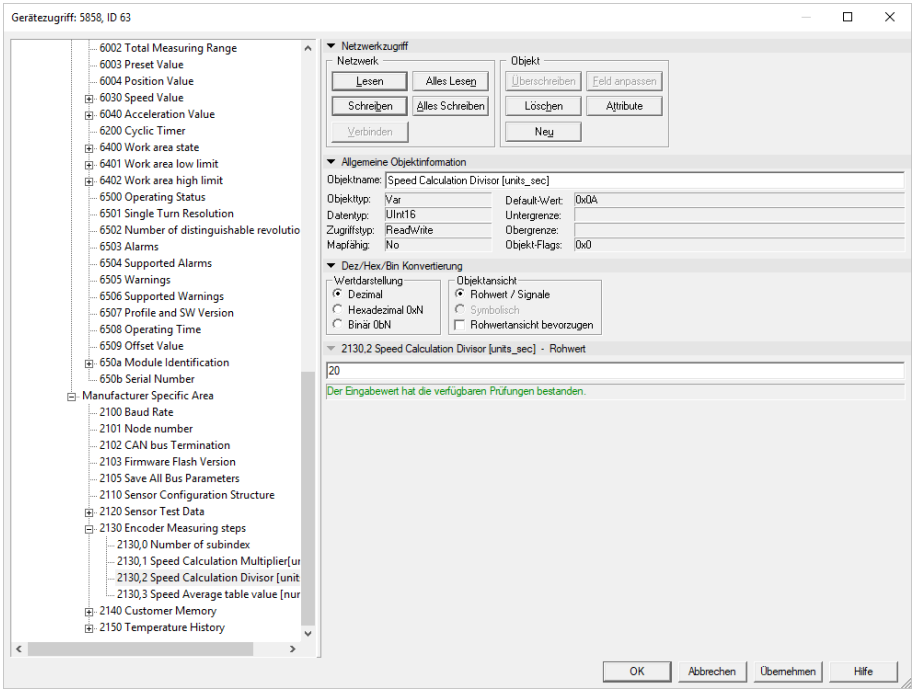


IMG-ID: 58303499

32 = maximum value

Speed output factor: 20

In this example, the speed value is calculated from 20 position values.



IMG-ID: 58305419

5.7.4 PDO Mapping

Example for the representation of position, speed and accuracy in Transmission PDO3.
Proceed as follows to adapt the mapping:

1. Set the PDO to invalid by negating bit 31 in the corresponding COB-ID entry.
2. Set the PDO mapping to invalid by writing 00h in subindex 00h of the corresponding assignment entries.
3. Adapt the desired PDO mapping.
4. Set subindex 00h of the corresponding mapping index to the number of assigned objects.
5. Use bit 31 in the corresponding COB-ID entry to set the PDO back to valid.

Index 1A02
Set the number of entries to 3 (3 objects can be assigned) with index 1A02, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2F	02	1A	00	03	00	00	00
Response	60	02	1A	00	00	00	00	00

Open array with index 1A02, subindex 00.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2F	02	1A	00	00	00	00	00
Response	60	02	1A	00	00	00	00	00

Mapping object 6004 sub 0 with 32-bit data length to representation object 1 with index 1A02, subindex 01.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	02	1A	01	20	00	04	60
Response	60	02	1A	01	00	00	00	00

Mapping object 6030 sub 1 with 16-bit data length to mapping object 2 with index 1A02, subindex 02.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	02	1A	02	10	01	30	60
Response	60	02	1A	02	00	00	00	00

Mapping object 6040 sub 1 with 16-bit data length to mapping object 3 with index 1A02, subindex 03.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	02	1A	03	10	01	40	60
Response	60	02	1A	03	00	00	00	00

Closing array with index 1A02, Subindex 00 to 03.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	2F	02	1A	00	03	00	00	00
Response	60	02	1A	00	00	00	00	00

Example of a change of the mapped object in PDO.

5.7.5 Modifying the COB-ID

Modification of the COB-ID from standard PDO1 (180 + Node-ID) to 0x40B.
Index 1800
ID 11

Invalidating the current COB-ID with index 1800 subindex 01.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	00	18	01	8B	01	00	80
Response	60	00	18	01	00	00	00	00

Writing a new COB-ID with Index 1800, Sub-Index 01 to 40B.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	00	18	01	0B	04	00	80
Response	60	00	18	01	00	00	00	00

Validating the new COB-ID with index 1800 subindex 01.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	00	18	01	0B	04	00	00
Response	60	00	18	01	00	00	00	00

Replacing the COB-ID of TPDO 2 (280 + Node ID) with TPDO 3 (380 + Node ID)

Index 1801
ID 8
Invalidating the current COB-ID with index 1801 subindex 01.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	01	18	01	88	03	00	80
Response	60	01	18	01	00	00	00	00

Overwriting the new COB-ID with index 1801, subindex 01 from 288 to 388.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	01	18	01	88	02	00	80
Response	60	01	18	01	00	00	00	00

Saving all parameters with the command "save".

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	10	10	01	73	61	76	65
Response	60	10	10	01	00	00	00	00

5.7.6 Using the LSS

Exactly two conditions must be met for devices connected to a CANopen network:

- • All devices must have the same baud rate.
- • The node address of every device must be unique within a network.

An existing 1:1 CAN connection to the device is the prerequisite for operation under LSS. Then, a special dialog mode allows modifying the baud rate and the node address. The COB-ID 0x7E5 is used from the master to the slave. The slave answers with COB-ID 0x7E4. LSS messages always have a length of 8 bytes. Unused bytes are reserved and must be filled with 0.

A Switch Mode Global command is sent to switch a device into the LSS configuration mode:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	04	01	00	00	00	00	00	00

NOTICE	**Switch Mode Global is not acknowledged**
	Only a visual check of the LEDs allows determining whether the device has switched to this mode.

All other commands are acknowledged. The response of the slave device always includes also the command byte of the request. Meaning:

- 0 = command accepted.
- 1 = unauthorized node ID.

The other error codes are reserved. The Error extension contains manufacturer-specific information and is displayed at Error code 0xFF.

The next command sent is Inquire node ID:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	5E	00	00	00	00	00	00	00
Response	5E	00	00	00	00	00	00	00

NOTICE	**Inquire node ID**
	If no response from the device is read, the LSS service is not supported or the baud rate is wrong.

The CONFIGURE NODE ID command is used to reconfigure the node address, e. g. to address 2a:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	11	2a	00	00	00	00	00	00
Response	11	00	00	00	00	00	00	00

The baud rate is activated by command CONFIGURE BIT TIMING PARAMETERS. In this case, the baud rate is set to 125 kbit/s:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	13	00	04	00	00	00	00	00
Response	13	00	00	00	00	00	00	00

According to CiA, all baud rates are represented by a specific index:

Index	Baud rate [kbit/s]
0	1000
1	800
2	500
3	250
4	125
5	Reserved
6	50
7	20
8	10

Both network-specific parameters have now been modified. STORE CONFIGURATION allows saving the new parameters:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	17	00	00	00	00	00	00	00
Response	17	00	00	00	00	00	00	00

To conclude the LSS service, the device is switched back from the LSS configuration mode to the Pre-operational mode by command SWITCH MODE GLOBAL:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	04	00	04	00	00	00	00	00

The device performs autonomously a new bootup (reset node). All new settings are then valid.

5.7.7 CANopen Trace

Below, a TRACE of a CANopen communication is to illustrate the execution of the CANopen commands and their responses. Very different settings are performed and subsequently saved. All of the settings of communication objects 0x21xx only become active after a new bootup of the device.

ID	Dir	DLC	Data	Interpretation
73F	Rx	1	00	Bootup message from the encoder
63F	Rx	8	23 11 10 01 6C 6F 61 64	Load factory setting
5BF	Rx	8	60 11 10 01 00 00 00 00	Response from the encoder
0	Rx	2	81 3F	Reset node
73F	Rx	1	00	Bootup message from the encoder
63F	Rx	8	2F 00 21 00 05 00 00 00	Setting to 250 kHz baud rate
5BF	Rx	8	60 00 21 00 00 00 00 00	Response from the encoder
63F	Rx	8	2F 01 21 00 3F 00 00 00	Setting node address to 3f
5BF	Rx	8	60 01 21 00 00 00 00 00	Response from the encoder
63F	Rx	8	2F 02 21 00 00 00 00 00	Termination off
5BF	Rx	8	60 02 21 00 00 00 00 00	Response from the encoder
63F	Rx	8	23 05 21 00 73 61 76 65	Save all bus objects
5BF	Rx	8	60 05 21 00 00 00 00 00	Response from the encoder

✓ Make sure that the parameters have been saved

a) Switch off

b) Switch on

⇨ The encoder restarts with the new bus settings.

ID	Dir	DLC	Data	Interpretation
73F	Rx	1	00	Bootup message from the encoder
63F	Rx	8	2B 00 62 00 0A 00 00 00	Setting event time to 10 ms
5BF	Rx	8	60 00 62 00 00 00 00 00	Response from the encoder
63F		8	2B 00 60 00 01 00 00 00	Setting counting direction to CCW
5BF		8	60 00 60 00 00 00 00 00	Response from the encoder
63F	Rx	8	23 10 10 01 73 61 76 65	Save all
5BF	Rx	8	60 10 10 01 00 00 00 00	Response from the encoder
0	Rx	2	01 3F	Switch to the operating mode
1BF	Rx	4	BA 4C 00 00 00 00	Position values
1BF	Rx	4	BA 4C 00 00 00 00	Position values
1BF	Rx	4	BA 4C 00 00 00 00	Position values
1BF	Rx	4	BA 4C 00 00 00 00	Position values
0	Rx	2	80 3F	Switching to Pre-operational

5.8 Emergency messages, Error and Abort Codes

Emergency objects appear in case of error situations within a CAN network, they are triggered according to the event and transferred on the bus with high priority.

NOTICE	**Triggering of the Emergency objects** An Emergency object is only triggered once per “event”. No new object is generated until the error is corrected. Once the error is corrected, a new Emergency object with content 0 (Error Reset or No Error) is generated and sent on the bus.

An Error message is a high-priority message with the following format:

CAN header	rtr	len	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x080 + Node ID	0	8	Error Code		Error register	Vendor-specific data				

A total of three error codes are transmitted, the "Emergency Error Code", the content of object Error Register 0x1001 and a manufacturer-specific code. The standard CANopen Error codes consist of profiles 301 and DS-406. They are defined as follows:

Error Code	Description
0x00xx	No error (reset error)
0x10xx	Generic error (not defined)
0x20xx	Current fault
0x21xx	Current device input side
0x22xx	Current in the device
0x23xx	Current device output side
0x30xx	Voltage error
0x31xx	Voltage display
0x32xx	Voltage in the device
0x33xx	Output voltage
0x40xx	Temperature
0x41xx	Ambient temperature
0x42xx	Device temperature
0x50xx	Device hardware
0x60xx	Device software
0x61xx	Internal software
0x62xx	User software
0x63xx	Data set
0x70xx	Additional modules
0x80xx	Communication
0x81xx	Communication
0x8110	CAN overrun (objects lost)
0x8120	CAN in Error Passive mode
0x8130	Lifeguard error or heartbeat error
0x8140	Recovered from the bus
0x8150	CAN ID collision
0x82xx	Protocol error
0x8210	PDO not processed because of a length error
0x8220	G.E. length exceeded
0x8230	DAM MPDO not processed target object not available
0x8240	Unexpected SYNC data length
0x8250	RPDO timeout
0x90xx	External error
0xF0xx	Additional functions
0xFFxx	Device-specific

The Error register is defined for 8 states.

Bits	Description
bit 0	generic error
bit 1	current
bit 2	voltage
bit 3	temperature
bit 4	communication error (overrun error state)
bit 5	device profile specific
bit 6	reserved (always 0)
bit 7	manufacturer specific

e. g. 11 = bit 0 and bit 4

5.8.1 Implemented error codes

The table shows an extract of the available error codes. The Emergency messages are sent autonomously by every CANopen device. With the current version of the CANopen communication profile, the transmission of an emergency message can also be switched off.

Error Code	Error Register	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Description	Error type	Product
0xFF00	80	13	5	0	0	0	Sensor Sync. Failure	Critical error	
0x4200	8	20	84	3			System Temperature Error	Self verification	
0x8110	10	0					CAN Overrun Error	Self verification	
0x8120	10	0					CAN Passive Error Mode	Self verification	
0x8130	10	0					Lifeguard or Heartbeat error	Self verification	
0xFF01	80	0	XX	XX	XX	XX	Bootloader no Firmware		
0x5000	1	E1	1				Internal Memory Error NVM	Critical error	
0x5051	1	3	5				Error Multiturn	Critical error	
0x3200							Battery voltage too low		F5888M
0x4200							Internal temperature too high		F5888M
0x5050							Singleturn sensor error		F5888M
0x5051							Multiturn sensor error		F5888M
0x5052							Multi sensor error		F5888M
0x6100							Error internal signal processing		F5888M

5.8.2 Example of an error message

Example of a message in case of overtemperature

CAN header	rtr	len	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x080 + Node ID	0	8	00	42	09	80	56	20	50	2E

Part of the error frame	Value	Description
Error code	4200	Device temperature threshold value exceeded
Error register	09	Error register
Manufacturer-specific 1	80	Sensor Error register
Manufacturer-specific 2	56	Current sensor temperature
Manufacturer-specific 3	20	Sensor current threshold value lower range
Manufacturer-specific 4	50	Sensor current threshold value upper range
Manufacturer-specific 5	2E	Sensor version register

5.8.3 Error behavior

Behavior for uncritical errors class

After correction of the error, a new emergency object with content 0 (error reset or no error) is generated and transmitted via the bus (no system-relevant error).

Behavior for critical errors class

If the error is still present after an Off/On cycle, the device must be sent back for servicing as the position or the configuration is no longer valid.

If the error has disappeared after an Off/On cycle, the validity of position and configuration must be checked.

5.8.4 CANopen Abort Codes

If a CANopen device is addressed incorrectly, it returns a specific ABORT CODE. An ABORT message has the following format:

CAN header	rtr	len	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x080 + Node ID	0	8	80	Object		Subindex	Abort code			

The ABORT message is in all cases indexed with 0x80. The object and its subindex, to which the error refers, are also indicated. Then the ABORT CODE is written in the 4 last bytes. The ABORT CODE can also be read in little-endian format. All ABORT CODES are listed in the table.

Error Code	Description
0x 0503 0000	Toggle bit not changed
0x 0504 0000	SDO protocol timeout
0x 0504 0001	Client / server command specifier invalid or unknown
0x 0504 0002	Invalid block size (only block mode)
0x 0504 0003	Invalid sequence number (only block mode)
0x 0504 0004	CRC error (only block mode)
0x 0504 0005	Out of memory
0x 0601 0000	Unsupported access to an object
0x 0601 0001	Attempt to read a <write-only> object
0x 0601 0002	Attempt to write a <read-only> object
0x 0602 0000	Object does not exist in the object dictionary
0x 0604 0041	Object cannot be mapped to the PDO
0x 0604 0042	The number and length of the objects to be mapped would exceed PDO length
0x 0604 0043	Reason for parameter incompatibility
0x 0604 0047	General internal incompatibility in the device
0x 0606 0000	Access failed due to hardware error
0x 0607 0010	Data type does not match. Service length parameter does not match
0x 0607 0012	Data type does not match. Service time parameter too high
0x 0607 0013	Data type does not match. Service time parameter too low
0x 0609 0011	Subindex not found
0x 0609 0030	Parameter value range exceeded (only for write access)
0x 0609 0031	Value of parameter written too high
0x 0609 0032	Value of parameter written too low
0x 0609 0036	Maximum value is less than minimum value
0x 0800 0000	General error
0x 0800 0020	Data cannot be transferred to or stored in the application
0x 0800 0021	Data cannot be transferred or stored in the application because of local control
0x 0800 0022	Data cannot be transferred or stored in the application because of the current device state
0x 0800 0023	Dynamic generation of the object dictionary fails or no object dictionary present (e.g. object dictionary is generated from file and generation fails because of a file error)

Example of an error code

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request	23	00	50	00	01	00	00	00
Response	80	00	50	00	02	00	01	06

The device sends an error because of the attempt to write object 0x5000, which can only be read.

6 Annex

6.1 Scaling

The usability of the measured values output by the measuring system essentially depends on their scaling. Scaling the measured values presupposes that mathematical operations must be carried out, which, depending on the device type, are integrally or only partly supported. There are basically 3 different scaling types:

1. Binary scaling = Scaling function
2. Non-binary scaling = Universal Scaling function
3. Scaling by means of the gear factor = Gear Factor

6.2 Setting the Baud Rate

The line length must be taken into consideration when defining the topology and the baud rate. The maximum length of the CAN bus is mainly limited by its signal propagation time. The multi-master arbitration process assumes that the signals reach all participants at the same time, i. e. before the sampling within a bit cycle. The signal propagation time in every network component (transceiver, optocoupler, CAN controller) is approximately constant. Therefore the final bus length primarily depends on the baud rate.

Baud rate [kbit/s]	Bus length [m]
1000	< 20
500	< 100
250	< 250
125	< 500
50	< 1000
20	< 2500
10	< 5000

The relevant literature often indicates the value of 40 m for 1 Mbit/s.
This leads to the formula:

- Bit rate * line length < 1 Mbit/s * 40 m

In addition, the following physical framework parameters apply:

- Runtime of a driver stage = 20 ns
- Propagation of the electrical wave on the line = 17 cm/s
- Scanning of the bit at 70% of the bit time.

However, this does not apply to networks with optical electrical isolation of the CAN controllers. In a "worst case" consideration, the value can reduce to 5 m with 1 Mbit/s. However, in practice, a line length of 20 m can be achieved without problems. Repeaters must be used for lengths >1000 m.

Line resistance

The line resistance must not be ignored. A loss on the signal line occurs over the whole length of the line.

In this case, the "worst case" would be when a node transmits at the beginning of the bus line and a node receives at the end of the line.

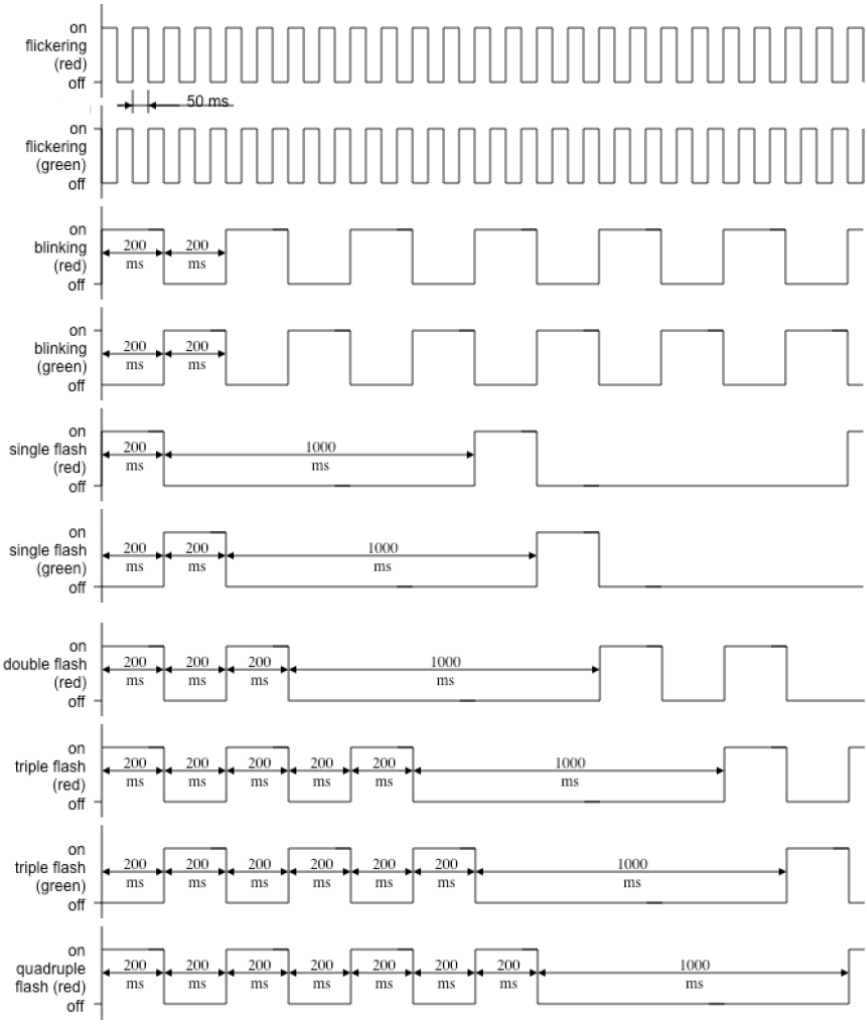
The strength of the differential signal at the receiving node depends on the following factors:

- Generated differential voltage of the transmitting node
- Line resistance RL = L * ρ, with L = line length [m] and ρ = resistance per m [Ohm/m]
- Differential input resistance of the receiving node

6.3 LED States

The LED states are clearly defined by CiA303.

LED status	Description
LED on	The LED must be continuously on.
LED off	The LED must be continuously off.
LED flickering	This must switch the Iso phase on and off and indicate it with a frequency of approximately 10 Hz. LED on for approx. 50 ms and LED off for approx. 50 ms.
LED blinking	This must switch the Iso phase on and off and indicate it with a frequency of approximately 2.5 Hz. LED on for approx. 200 ms followed by LED off for approx. 200 ms.
LED single flash	This signal consists in a short flash (approx. 200 ms) followed by a long off phase (approx 1000 ms).
LED double flash	This signal consists in a sequence of two short flashes (approx. 200 ms) separated by an off phase (approx. 200 ms). This sequence is concluded by a long off phase (approx. 1000 ms).
LED triple flash	This signal consists in a sequence of three short flashes (approx. 200 ms) separated by an off phase (approx. 200 ms). This sequence is concluded by a long off phase (approx. 1000 ms).
LED quadruple flash	This signal consists in a sequence of four short flashes (approx. 200 ms) separated by an off phase (approx. 200 ms). This sequence is concluded by a long off phase (approx. 1000 ms).



IMG-ID: 153433867

6.4 Decimal / Hexadecimal conversion table

Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex
0	*0x0*	51	*0x33*	102	*0x66*	153	*0x99*	204	*0xCC*
1	*0x1*	52	*0x34*	103	*0x67*	154	*0x9A*	205	*0xCD*
2	*0x2*	53	*0x35*	104	*0x68*	155	*0x9B*	206	*0xCE*
3	*0x3*	54	*0x36*	105	*0x69*	156	*0x9C*	207	*0xCF*
4	*0x4*	55	*0x37*	106	*0x6A*	157	*0x9D*	208	*0xD0*
5	*0x5*	56	*0x38*	107	*0x6B*	158	*0x9E*	209	*0xD1*
6	*0x6*	57	*0x39*	108	*0x6C*	159	*0x9F*	210	*0xD2*
7	*0x7*	58	*0x3A*	109	*0x6D*	160	*0xA0*	211	*0xD3*
8	*0x8*	59	*0x3B*	110	*0x6E*	161	*0xA1*	212	*0xD4*
9	*0x9*	60	*0x3C*	111	*0x6F*	162	*0xA2*	213	*0xD5*
10	*0xA*	61	*0x3D*	112	*0x70*	163	*0xA3*	214	*0xD6*
11	*0xB*	62	*0x3E*	113	*0x71*	164	*0xA4*	215	*0xD7*
12	*0xC*	63	*0x3F*	114	*0x72*	165	*0xA5*	216	*0xD8*
13	*0xD*	64	*0x40*	115	*0x73*	166	*0xA6*	217	*0xD9*
14	*0xE*	65	*0x41*	116	*0x74*	167	*0xA7*	218	*0xDA*
15	*0xF*	66	*0x42*	117	*0x75*	168	*0xA8*	219	*0xDB*
16	*0x10*	67	*0x43*	118	*0x76*	169	*0xA9*	220	*0xDC*
17	*0x11*	68	*0x44*	119	*0x77*	170	*0xAA*	221	*0xDD*
18	*0x12*	69	*0x45*	120	*0x78*	171	*0xAB*	222	*0xDE*
19	*0x13*	70	*0x46*	121	*0x79*	172	*0xAC*	223	*0xDF*
20	*0x14*	71	*0x47*	122	*0x7A*	173	*0xAD*	224	*0xE0*
21	*0x15*	72	*0x48*	123	*0x7B*	174	*0xAE*	225	*0xE1*
22	*0x16*	73	*0x49*	124	*0x7C*	175	*0xAF*	226	*0xE2*
23	*0x17*	74	*0x4A*	125	*0x7D*	176	*0xB0*	227	*0xE3*
24	*0x18*	75	*0x4B*	126	*0x7E*	177	*0xB1*	228	*0xE4*
25	*0x19*	76	*0x4C*	127	*0x7F*	178	*0xB2*	229	*0xE5*
26	*0x1A*	77	*0x4D*	128	*0x80*	179	*0xB3*	230	*0xE6*
27	*0x1B*	78	*0x4E*	129	*0x81*	180	*0xB4*	231	*0xE7*
28	*0x1C*	79	*0x4F*	130	*0x82*	181	*0xB5*	232	*0xE8*
29	*0x1D*	80	*0x50*	131	*0x83*	182	*0xB6*	233	*0xE9*
30	*0x1E*	81	*0x51*	132	*0x84*	183	*0xB7*	234	*0xEA*

Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex
31	*0x1F*	82	*0x52*	133	*0x85*	184	*0xB8*	235	*0xEB*
32	*0x20*	83	*0x53*	134	*0x86*	185	*0xB9*	236	*0xEC*
33	*0x21*	84	*0x54*	135	*0x87*	186	*0xBA*	237	*0xED*
34	*0x22*	85	*0x55*	136	*0x88*	187	*0xBB*	238	*0xEE*
35	*0x23*	86	*0x56*	137	*0x89*	188	*0xBC*	239	*0xEF*
36	*0x24*	87	*0x57*	138	*0x8A*	189	*0xBD*	240	*0xF0*
37	*0x25*	88	*0x58*	139	*0x8B*	190	*0xBE*	241	*0xF1*
38	*0x26*	89	*0x59*	140	*0x8C*	191	*0xBF*	242	*0xF2*
39	*0x27*	90	*0x5A*	141	*0x8D*	192	*0xC0*	243	*0xF3*
40	*0x28*	91	*0x5B*	142	*0x8E*	193	*0xC1*	244	*0xF4*
41	*0x29*	92	*0x5C*	143	*0x8F*	194	*0xC2*	245	*0xF5*
42	*0x2A*	93	*0x5D*	144	*0x90*	195	*0xC3*	246	*0xF6*
43	*0x2B*	94	*0x5E*	145	*0x91*	196	*0xC4*	247	*0xF7*
44	*0x2C*	95	*0x5F*	146	*0x92*	197	*0xC5*	248	*0xF8*
45	*0x2D*	96	*0x60*	147	*0x93*	198	*0xC6*	249	*0xF9*
46	*0x2E*	97	*0x61*	148	*0x94*	199	*0xC7*	250	*0xFA*
47	*0x2F*	98	*0x62*	149	*0x95*	200	*0xC8*	251	*0xFB*
48	*0x30*	99	*0x63*	150	*0x96*	201	*0xC9*	252	*0xFC*
49	*0x31*	100	*0x64*	151	*0x97*	202	*0xCA*	253	*0xFD*
50	*0x32*	101	*0x65*	152	*0x98*	203	*0xCB*	254	*0xFE*
								255	*0xFF*

7 Contact

You want to get in touch with us:

Technical advice

For technical advice, analysis or support during installation, Kübler is directly on site with its globally active application team.

Support International (English-speaking)

+49 7720 3903 849
support@kuebler.com

Kübler Germany +49 7720 3903 849
Kübler Australia +61 3 7044 0090
Kübler China +86 10 8471 0818
Kübler France +33 3 89 53 45 45
Kübler India +91 8600 147 280
Kübler Italy +39 0 26 42 33 45
Kübler Austria +43 3322 43723 12
Kübler Poland +48 6 18 49 99 02
Kübler Turkey +90 216 999 9791
Kübler USA +1 855 583 2537

Repair service / RMA-Form

For returns, please pack the product adequately and enclose the completed "Returns Form".

www.kuebler.com/rma

Send your return, specifying the RMA-reference, to the following address.

Kübler Group
Fritz Kübler GmbH

Schubertstraße 47
D-78054 Villingen-Schwenningen
Deutschland

Tel. +49 7720 3903 0
Fax +49 7720 21564

info@kuebler.com
www.kuebler.com

Glossary

CAN

Controller Area Network

CMD

Command

COB

Communication Object. Transport unit in the CAN network (CAN message). Data is sent through the network in a COB.

Default

English for standard, generally used as default value. Factory-preset value of a changeable configuration value.

EEPROM

Electrically erasable programmable read-only memory. Nonvolatile electronic memory elements whose saved information can be erased electrically.

EMC

Electromagnetic compatibility

ID

Identifier. Univocal designation of a CAN message. The identifier determines the priority of the COB in the network.

LSS

Layer Setting Service - Dynamic node number allocation

MUR

Measuring Units per Revolution

NDR

Number of Distinguishable Revolutions

NMO

Network Management Object

NMT

Various tasks are to be performed in a distributed system in connection with the configuration, initialization and monitoring of the network participants. The service element "Network Management (NMT)" defined in CANopen provides this functionality.

PDO

The process data objects (PDO) are the actual means of transport for the transfer of process data (application objects). A PDO is sent by a producer and can be received by one or several consumers.

PELV

Protective Extra Low Voltage. Functional extra-low voltage with electrically safe isolation

RMA

Return Material Authorization, authorization to return material, e.g. in the case of complaints.

SDO

Service data objects (SDO) are used for the confirmed transfer of data of any length between two network participants. Data transfer takes place in client-server mode.

SFO

Special Function Objects

TMR

Total Measuring Range

TPDO

Transmit PDO. A PDO transmitted via a CANopen device.

USF

Universal Scaling Function, a non-binary scaling function (without overflow error)

Kübler Group
Fritz Kübler GmbH
Schubertstr. 47
D-78054 Villingen-Schwenningen
Germany
Phone +49 7720 3903-0
Fax +49 7720 21564
info@kuebler.com
www.kuebler.com