Baumer

Passion for Sensors

Manual

Spindle position displays NM17x with CANopen®

Version V1.0.2
Valid from software version V1.07

Baumer Germany GmbH & Co. KG
Bodenseeallee 7
DE-78333 Stockach
www.baumer.com

01.24 · 174.02.081/4

Disclaimer

This manual was created very carefully. However, errors cannot be entirely ruled out at all times. Therefore, Baumer does not assume any type of liability for the information contained in this manual. Baumer or the author are not liable under any circumstances for any direct or indirect damage resulting from the application of this information.

We always welcome any suggestions that serve the improvement of this manual.

1 Introduction

1.1 Scope of delivery

Before commissioning, please verify the completeness of the delivery.
Depending on the version and order, the scope of delivery may include:

- Spindle position display
- Description files
- Mounting instructions
- Manual

1.2 Product assignment

Product	Product code	Device Name	EDS file	Product family
NM170	0x70	NM170 Standard	NM170.eds	Spindle position display
NM172	0x71	NM172 Standard	NM172.eds	Spindle position display
NM174	0x72	NM174 Standard	NM174.eds	Spindle position display

2 General information

2.1 Instructions for safety and operation

Intended use

- The spindle position display (SPA) is a precision measurement device that serves the detection of positions. It provides measured values as electronic output signals for the subsequent device. The device may only be used for this purpose. Unless specifically labeled, this product may not be used in explosive environments.
- Endangering persons, damage to the system or operational facilities due to the failure or error of the SPA must be ruled out by suitable safety measures.

Staff qualification

- Installation and mounting of the SPA may only be carried out by an electrics and precision engineering specialist.
- The operating instructions of the machine manufacturer must be observed.

Safety instructions

- Prior to commissioning the system, all electric connections must be checked.
- Improper mounting, electrical connection or other work on the SPA and the system may lead to malfunction or failure of the spindle position display.
- Endangering persons, damage to the system or damage to operational facilities due to the failure or error of the SPA must be ruled out by suitable safety measures.
- The SPA may not be operated outside the limit values (see additional documentation).
- Non-observance of the safety instruction may lead to malfunction, material and personal damage!

Mounting

- Avoid blows or shocks to the housing and hollow shaft.
- Do not wire the housing.
- Do not open or mechanically alter the SPA.

Preventive maintenance

- The spindle position display is maintenance free and may not be opened or altered mechanically or electrically.

Electrical commissioning

- Do not carry out wiring work while the system is live
- Do not attach or remove the electrical connections while the system is live
- Install the entire system considering the electromagnetic compatibility. The installation environment and wiring affect the electromagnetic compatibility of the SPA. Install the SPA and its supply lines physically separated or at a large distance from lines with a large interference level (frequency converters, contactors, etc.).
- If there are ultimate consumers with high interference levels provide a separate power supply for the SPA
- The max. electrical load via the connection lines must be observed. See product information.
- With bus cable extensions, the braid must be connected to pin 1.

Earthing concept

- To ensure maximum protection of the EMC and thus error-free operation, the connection to protective earth (PE) should be carried out as shown in the following figure.

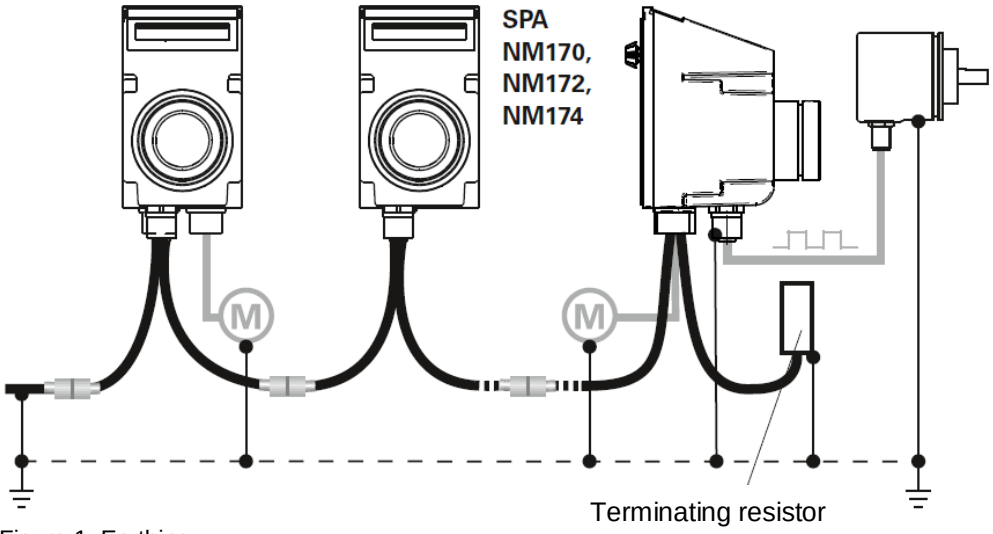


Figure 1: Earthing

Disposal

- The SPA contains electronic components and a battery. In case of disposal the local environmental specifications must be observed.

Additional information

- The manual complements other documentation (e.g. data sheet or mounting instructions).

2.2 General description

Mounting of the spindle position display (SPA) is by attaching the hollow shaft to the end of the spindle shaft or according fastening bolts for the remote display NM174 (Figure 2: A and B).

The hollow shaft is connected with a frictional connection via a hexagon socket head screw to the spindle. The SPA is therefore in an overhung position and is secured against rotation at the back of the housing via an included expanding pin (Figure 2: C and D).

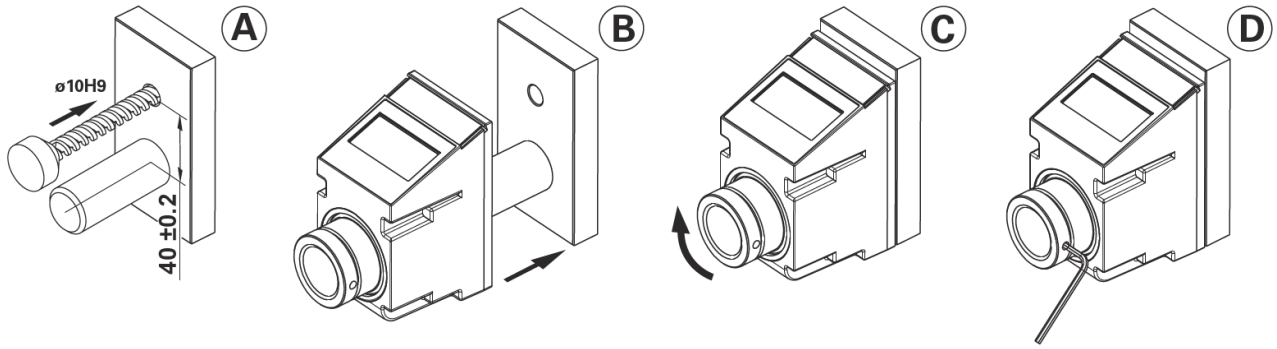


Figure 2: Mounting

The spindle position displays NM170 and NM172 feature an absolute multi-turn measuring system. This way, the position can also be recorded when the power is switched off for several revolutions of the spindle. The position points are not lost even if there is a power interruption (at least 10 years).

The remote display NM174 does not contain an internal measuring system. An external sensor with an SSI interface is connected linear or rotational via an 8-pin M12 sensor.

The current position value is displayed in the two-line (2 x 6-digit, numerical) back-lit LED display as the actual value . At the same time, the same display can include the provided set value (e.g. by the master). Two directional arrows in the display inform the machine operator the rotational direction to which the spindle should be adjusted to match the actual value to the set value. If the actual value matches the set value within the tolerance range, the set value disappears.

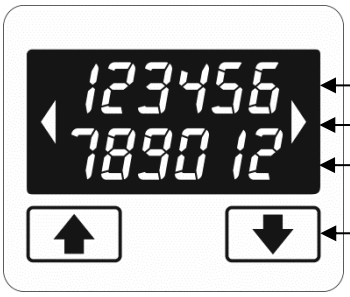
The display can be rotated by 180° via the parameter programming, for example on the master, to allow the vertical installation of the device. The spindle displays are connected via cables with assembled male connectors and socket plugs. The power supply for all connected SPAs is via a single cable.

Display points:
Including the positions after the decimal point, in the positive range 6-digit values and in the negative range 5-digit values can be displayed (the 6th position from the right includes the minus sign).
If the value is above the maximum actual value of e.g. 9999.99 or below the minimum value of -999.99, then pre-zeros are shown. At the same time, NM172 and NM174 display the error message *Er 01* or *Er 02* (breach of the end stop) at the latest upon reaching the maximum or minimum value.

NM170

Display "set value"
Arrows for rotational direction for adjustment of the set value/actual value
Display "actual value

NM172, NM174



Display "set value"
Arrows for rotational direction for adjustment of the set value/actual value
Display "actual value"

The NM172 and NM174 can be connected to a motor via the 12-pin cable. The two keys can be used to manually control the motor. It is possibly to automatically control the motor via the‘ master. The power supply for the motor is separate.

2.3 Boot phase

When connecting the operating voltage, the device identification data is shown for the first 5 seconds on the display. During this boot phase, the device is already fully operational. All read and write operations can be carried out via the CAN interface. Similarly, automatic position indication with NM172 and NM174 is already possible.

The following information is displayed in sequence, here shown on an NM170 as an example:

Phase 1: Display duration 0 … 2.5 seconds

Device denomination: here NM**170**.**A0**1

Firmware version of the STM controller (see Section 3.6 block circuit diagram)

Phase 2: Display duration 2.5 ... 5 seconds

Node ID

Phase 3: Normal display after 5 seconds. Boot phase is completed.

Display "set value"

Display "actual value "

3 Function descriptions

3.1 Addressing (allocation of the node ID)

To be able to communicate on the CAN bus, each device must have a clearly defined node ID.
The node ID is allocated via one of the following addressing mechanisms:

- Addressing an individual SPA via the SDO object 0x2101 (SDO addressing).
- Addressing individual SPAs via layer setting services (LSS addressing).
- Device specific addressing

3.1.1 SDO addressing

This addressing is primarily used when the node ID of the SPA is known.
See description of the object 0x2101 in Section 4.4.5.26 Node ID
For this addressing, the SPA must either be the only device operated on the Master or already have a clearly defined node ID within a CANopen network, which should then be changed.

3.1.2 LSS addressing

A new node ID can be allocated to a random spindle position display (SPA) connected to the network with an unknown node ID via the LSS (Layer Setting Service) protocol.

Addressing takes place via the device data of the object 0x1018 (identity object). With these four specification, each device in the network can be clearly identified as each SPA has a unique serial number.
This way, any number of SPAs can be connected to a bus system with the same node ID and then initialized via LSS.
For more information about the topic of LSS, see Section *4.3.9 Layer Setting Services (LSS)*

For the following example, the four parameters of the identity object for an SPA NM172 are presumed

Object	Name	Value
0x1018:01	Vendor ID	0x000000EC
0x1018:02	Product Code	0x00000071
0x1018:03	Revision number	0x00010001
0x1018:04	Serial number	0x4D52F567

The following steps must be implemented:

- Set the SPA to the Operational or Stop mode (here stop mode) [No. 1]
- Switch to configuration mode [No. 2]
- Transmit the four entries of object 0x1018 in sequence [No. 3-6]
- After the confirmation of the SPA has taken place [No. 7], send desired address [No. 8]
- After the confirmation of the SPA has taken place [No. 9], store the new address [No. 10]

Example of LSS addressing:

No	Dir	COB ID	DLC	Data (hex)	Comment / Process
1	→	000	2	02 00	Stop Communication
2	→	7E5	2	04 01	Switch mode global: Configuration
3	→	7E5	8	40 EC 00 00 00 00 00 00	Vendor ID
4	→	7E5	8	41 71 00 00 00 00 00 00	Product Code
5	→	7E5	8	42 01 00 01 00 00 00 00	Revision number
6	→	7E5	8	43 67 F5 52 4D 00 00 00	Serial number
7	←	7E4	8	44 00 00 00 00 00 00 00	Confirmation of the SPA

No	Dir	COB ID	DLC	Data (hex)	Comment / Process
8	→	7E5	2	11 25	Transmit the desired address (here 25)
9	←	7E4	8	11 00 00 00 00 00 00 00	Confirmation of the SPA
10	→	7E5	2	17 00	Save command. Address is acquired
11	←	7E4	8	17 00 00 00 00 00 00 00	Confirmation of the SPA

→ Master transmits
← SPA responds

3.1.3 Device specific addressing

All SPAs present in the network are addressed simultaneously (broadcast) and the desired address (node ID) is shown in the own display. Address acquisition takes place on a random SPA by turning the spindle or pushing a key.

This is enhanced LSS addressing (see above) that is not specified according to CiA, but constitutes manufacturer-specific addressing.

For the broadcast transmission, instead of the identification data that differ from one SPA to another, the broadcast values 0xFFFFFFFF are transmitted. The four parameters of the identity object are transmitted with the following
data:

Object	Name	Value
0x1018:01	Vendor ID	0x000000EC
0x1018:02	Broadcast Product Code	0xFFFFFFFF
0x1018:03	Broadcast Revision Number	0xFFFFFFFF
0x1018:04	Broadcast Serial Number	0xFFFFFFFF

The LSS protocol (COB ID 7E4h and 7E5h) is used for the transmission.

Protocol sequence for the allocation of node IDs via broadcast function:

No	Dir	COB ID	DLC	Data (hex)	Comment / Process
1	→	000	2	02 00	Stop Communication
2	→	7E5	2	04 01	Switch mode global: Configuration
3	→	7E5	8	40 EC 00 00 00 00 00 00	Vendor ID
4	→	7E5	8	41 FF FF FF FF 00 00 00	Broadcast Product Code
5	→	7E5	8	42 FF FF FF FF 00 00 00	Broadcast Revision Number
6	→	7E5	8	43 FF FF FF FF 00 00 00	Broadcast Serial Number
					All SPA switch over to address mode
7	→	7E5	8	11 02 00 00 00 00 00 00	Set first node ID to 02
					All SPAs display address 02 (top line)
					Spindle on SPA 02 is rotated -> acquisition
8	←	7E4	8	11 00 00 00 00 00 00 00	Command Response: No Error
					Node ID is stored non-volatile
9	→	7E5	8	11 **03** 00 00 00 00 00 00	Set next node ID to 03
					All SPAs display address 03 (top line)
					Spindle on SPA 03 is rotated -> acquisition
10	←	7E4	8	11 00 00 00 00 00 00 00	Command Response: No Error
				:	Allocate additional addresses (same as No. 9-10)
				:	
xx	→	000	2	81 00	Reset node

→ Master transmits
← SPA responds

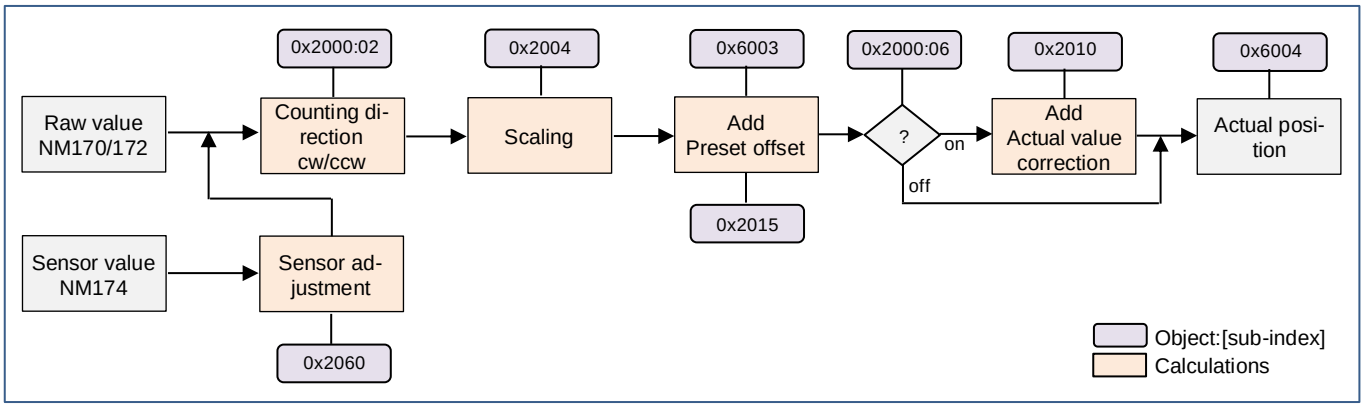
3.2 Setting the SPA to the correct dimension

Prior to its first use, the spindle position display (SPA) must be adjusted to the dimensional conditions of the place of installation. Carry out the following steps in the stated sequence to prevent subsequent corrections.

The following table lists all SDO objects involved in the position calculation. The *Priority* column states the sequence of the recommended parameterization. Starting with priority 1. With the same priority the sequence is arbitrary.

SDO object	Name	Priority
0x2000:02	Counting direction (cw/ccw)	1
0x2000:06	Actual value correction on/off	1
0x2060:01-03	Sensor parameters (only for NM174)	1
0x2004:00	Scaling	2
0x6003:00	Preset	3
0x2010:00	actual value correction (value)	4
0x2015:00	Reset preset offset (parameter 6)	-
0x6004:00	Read actual position	-

The following graph shows the dependency of the position calculation regarding the individual objects:



3.3 Position indication

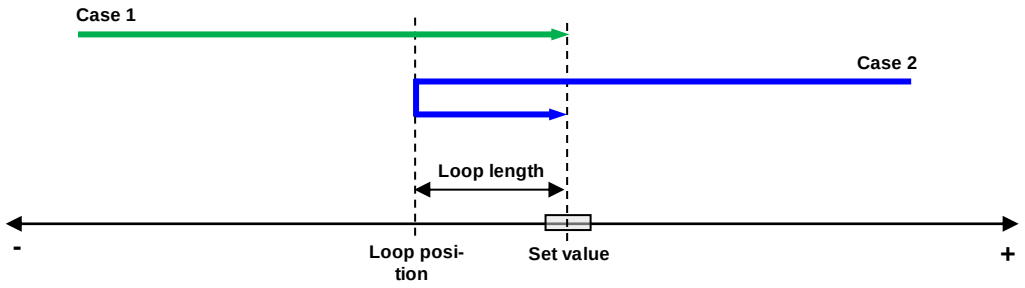
The main task of the spindle position display (SPA) is to position the spindle on a set position. This can be carried out by manually turning the spindle with the NM170 or motor-actuated with NM172 and NM174. Manual position indication is possible with all Multicon devices.

3.3.1 Loop position indication

When using the SPA on spindles or transmissions there is a problem of spindle or transmission play. Therefore a position must always be approached from the same side. The approach side and loop length can be adjusted. See object 0x2000:01 and 0x2002:00.

The following example applies to the default setting. The position indication direction (object 0x2000:01) is parameterized to 0 = UP. The new position should be approached in the positive (increasing) direction.

Case 1: New position (set value) > current actual position
Case 2: New position (set value) < current actual position



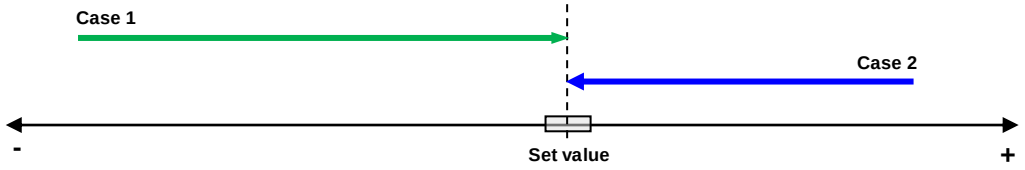
3.3.2 Position indication without loop

If no play compensation is required, the new position can always be approached directly. All that is required is to set the loop length (object 0x2002:00) to 0.
The settings in the objects:

- 0x2000:01 Position indication direction
- 0x2047:01 loop time

are ignored.

Case 1: New position (set value) > current actual position
Case 2: New position (set value) < current actual position



3.3.3 Status flags

Only NM172, NM174

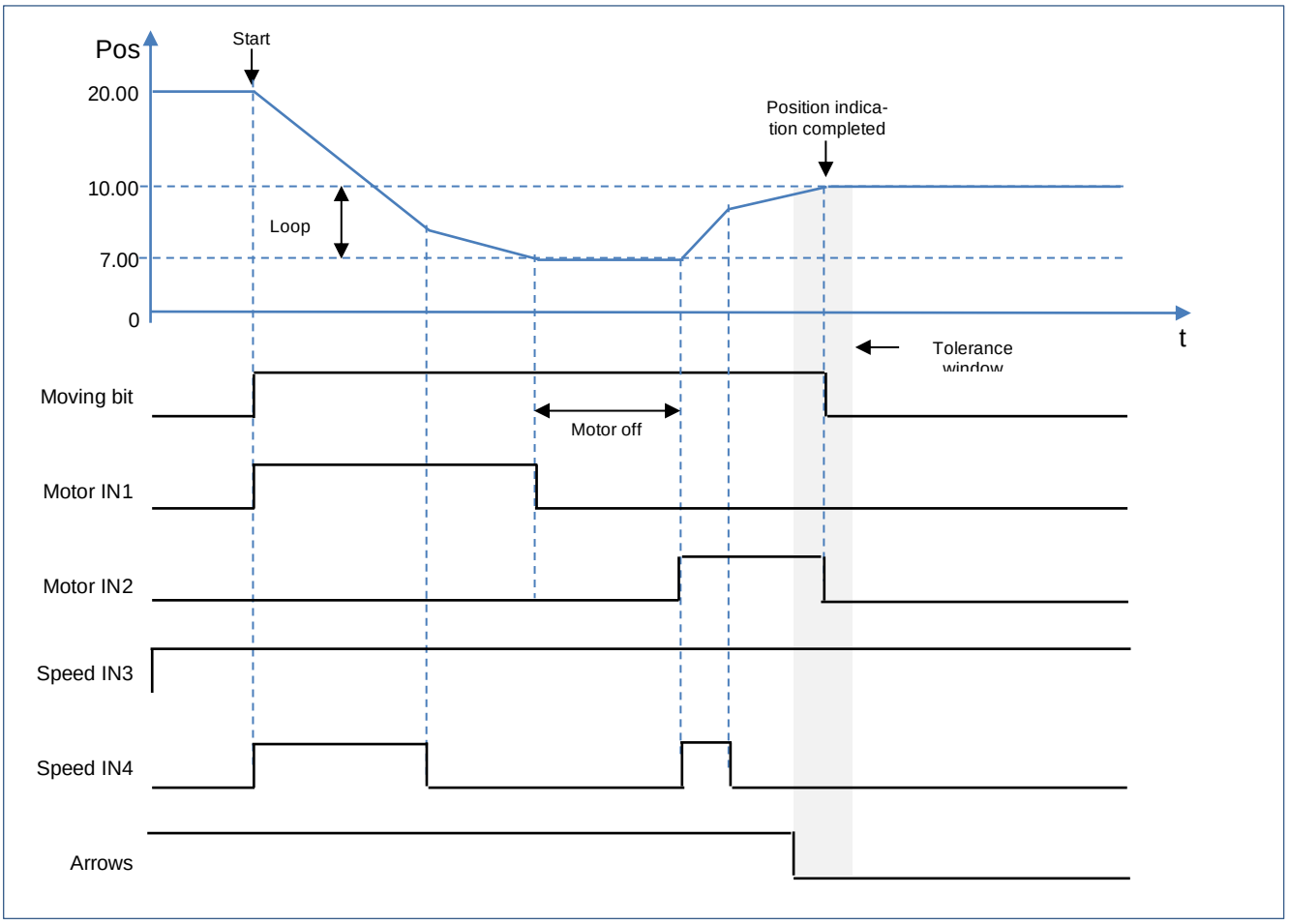
During motor-actuated position indication, status flags provide the current status of the position indication. The status flags are transmitted cyclically via TxPDO1 and TxPDO2 or can be read out via SDO object 0x204E:01.

The following status flags are relevant for this:

- Bit 0: Moving Bit
- Bit 8-9: Motor speed level (speed) (IN4, IN3)
- Bit 10-11: Motor direction signal (IN2, IN1)
- Bit 12: Arrows active (visible on the display. permanent or blinking)
- Bit 15: In position

The following diagram shows the status of these status flags and the time sequence during a position indication with loop drive:

Initial situation: actual value = 20.00; set value = 10.00; loop = 3.00; other parameters = default



Note: the diagram is not true to scale

Movingbit during clamping:

With a defined clamping time of > 0, the moving bit is set before and after the positioning operation during the clamping time delay. Hence, the moving bit is set from the point in time of clamping disabled (opened) until clamping re-enabled (closed).

3.3.4 Objects that affect position indication

The following table lists the SDO objects that directly influence position indication:

Object:[sub-index]	Name	Description
0x2000:01	Position indication direction	Determines the side from which the target position is approached
0x2000:01	Arrows	Inform the operator of the rotational direction with manual position indication
0x2002:00	Loop length for spindle play compensation	Indicates the loop value in millimeters that should be used for the play compensation.
0x2003:00	Tolerance window	The size of the tolerance window for the set value. If the actual position is within this tolerance window, the SPA provides the status information "in position."
0x2042:01	Creep speed	Relative value before the loop value (if with loop drive) and before the end position at which the speed switches from fast to creep speed
0x2042:02	Crawl speed	Relative value before the loop value (if with loop drive) and before the end position at which the speed switches to crawl speed
0x2042:03	Switch off value	Relative value that allows stopping the motor before ot after the target position. This compensates potential overdrive or withdrawal of the motor after the motor is stopped. If 0 then the motor stops at the target position.
0x2047:01	Loop time	Retention time of the motor at the reversal point of the loop drive
0x2047:03	Clamping	Regulates the delayed start and stop of the motor when using an external clamping device for the drive shaft.

3.3.5 Abandoning automatic position indication

Automatic positioning is distinguished by the fact that the CAN interface prompts the motor to either drive independently to a new set value or with the *press* function without a specific goal to move until a stop command is issued.

For operating and safety reasons, there are different actions that stop an active position indication process. The following table lists all cases:

Action	Description
Stop motor command	Object 0x204D:01 or control bit 14 in RxPDO2 are the official commands for stopping an active position indication.
Determine a new set value	If a new set value is specified during a position indication via object 0x2020 or RxPDO2, the ongoing position indication stops. If a command with motor release is issued, the position indication for the new set value is started immediately after the stop of the ongoing position indication.
Key activation	If the device key is briefly activated during automatic position indication, the position indication stops. Similar to an EMEREGNCY OFF key.
Change of status	In case of a change of status to a lower level, a position indication stops. Please note: Position indication is possible in the operational and pre-operational status. There is no stop when changing from pre-operational to operational.
Heartbeat event	If the SPA detects a heartbeat error, then there is a communication problem. An active position indication is abandoned.
Reset node	If the reset node is carried out, the device is restarted and a position indication terminated.
Power interruption	After a power interruption, a position indication not previously completed is not automatically concluded. The position indication must be restarted.

3.3.6 Jog function (microstep)

This function, which is also known as microstep, allows the movement of a shaft by a previously specified number of steps. The function is started by a brief push of a key (< 400ms).
Starting via the external inputs "F" and "G" is also possible.

The following table lists all objects that affect the jog function:

Object	Denotation	Selection	Description
0x2040:03	Activate jog	Up Down Ever Off	This objects blocks or releases the individual keys and the external inputs for implementation of the jog function. See also the object description.
0x2040:01	Key allocation	Up Down	Interchanges the key allocation. If "Down" is selected, the left key assumes the function of the right key, and vice versa.
0x2000:08	External inputs	Key Slow Middle Fast	In the setting "Key", the jog function can be implemented via the external inputs as well as the keys. In the other settings, the jog function via the external inputs is deactivated .
0x2046:00	step length	0..9999	Number of steps with a brief pressing of a key. The movement (up/down) is determined by the key. Note: The step length is affected by object 0x2042:03.
0x2042:03	Switch off value	0..9999	A switch off value not equal to 0 directly affects the step length. The following dependency applies: If the switch off value >= step length, the following applies: $Step\ length_{internal} = 1$ In all other instances the following applies: $Step\ length_{Internal} = step\ length - switch\text{-}off\ value$ Legend: step length$_{Internal}$ is the value of the actual movement.

The following table shows the effect of the parameters *jog activation* and parameter *key allocation*:

Parameter setting		Effect with brief key activation	
Activation 0x2040:03	Key allocation 0x2040:01	Left key Input "G"	Right key Input "F"
Up	Up	-	Pos +
Up	Down	Pos +	-
Down	Up	Pos -	-
Down	Down	-	Pos +
Ever	Up	Pos -	Pos +
Ever	Down	Pos +	Pos -
Off	Up	-	-
Off	Down	-	-

Pos + : Position indication in the positive direction.
Pos + : Position indication in the negative direction
- : no function

3.4 Operation types of the external inputs

The operation types consist of different positioning types that are partially controlled via the CAN interface but primarily via the external inputs.

The individual operating types are set via the object 0x2041:03. The following settings are available as operation types:

0 Function standard / oscillate
1 Function Cycle
2 Function ExternPosi
3 OFF

3.4.1 Oscillate

The function Oscillate allows in the NM172 and NM174 constant position indication between two previously set positions D1 and D2 (see object 0x2045).

Enable the function "Oscillate"
For the function "oscillate" to be used, it must be enabled (object 0x2041:03 = 0).
The following selection options are available:

0 Function standard / oscillate
1 Function Cycle
2 Function ExternPosi
3 OFF

After delivery and reset to default values, the "oscillate" function is already selected, but not yet active. The individual functions cannot be used simultaneously. Switching between functions is possible at any time.

Parameterization of the reverse functions D1 and D2
Parameterization of the reverse functions D1 and D2 is via object 0x2045.

Starting and ending the "oscillate function"
"Oscillating" can be activated and deactivated by an external switch or the CAN interface. Please note that the external switch is prioritized.

a) Activation via an external switch:
The oscillation can be activated and deactivated by an external rotary switch. This switch is connected to the input pins "F" and "G" of the motor cable.

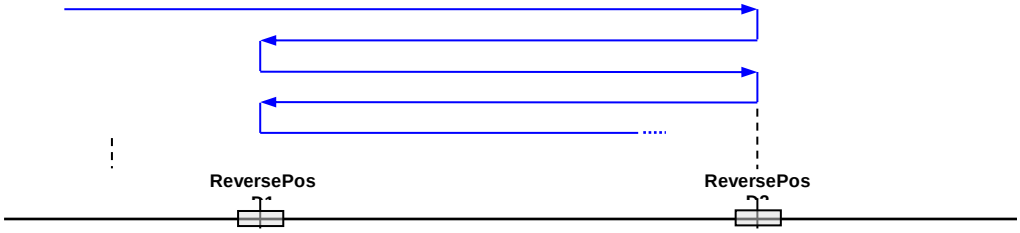
PIN G	PIN F	Function
0	0	Display normal/no oscillation
0	1	Manual movement of the shaft via an external input/no oscillation
1	0	Manual movement of the shaft via an external input/no oscillation
1	**1**	**Oscillation active**

If the inputs are operated individually, e.g via external keys, the shaft can be moved manually according to the "ext. inputs" programming in object 0x2000:08. The oscillating function is only activated in the setting "both inputs active."

b) Activation via the CAN interface:
The oscillation function can also be started and stopped via a drive command (object 0x204D:02).
If an operation type switch is connected to the external inputs "F" and "G", it must be set to "both inputs inactive." Otherwise, activation or deactivation is not possible via the interface. The operation type switch is prioritized. Reading out the status remains possible.

Start the oscillating function: object 0x204D:02 = 1

Graphical representation of the oscillation function



The individual position indications on D1 and D2 take place the same way as position indication on a profile value. Prior to reaching the reversal point every time, the speed is switched to creep/crawl speed. The positions themselves are approached directly without a loop drive.

Ending the oscillation function via:

- the CAN interface: set object 0x204D:03 = 0
- the external inputs: set external inputs "F" and "G" to inactive

After termination of the oscillation function, the SPA automatically indicates the position according to the set value (object 0x2020:01 or 0x2020:02). This set value can be newly parameterized via the CAN interface while the oscillation function is active. This setting then serves as the current set value position.

3.4.2 Cycle

The function Cycle allows in the NM172 and NM174 position indication at two previously set positions (D1 and D2) via the external inputs F and G. If input F is activated (wing), then the position indication is for position D1. Similarly, if input G is activated, then the position indication is for position D2 (wing).
Parameters D1 and D2 are defined via object 0x2045.

Enable the function "Cycle"
For the function "cycle" to be used, it must be enabled (object 0x2041:03 = 1)
The following selection options are available:

0 Function standard / oscillate
1 Function Cycle
2 Function ExternPosi
3 OFF

After delivery and reset to the default values (object 0x2015) the function "cycle" is deactivated. The individual functions cannot be used simultaneously. Switching between functions is possible at any time, however.

Process of position indication via "cycle"
When activating an input (wing), the position value of D1 or D2 is written as the new set value in the top display and the position indication started. The position indication is carried out even if the respective input is deactivated. After the completion of the position indication, the SPA stays at this position and waits for new actions. The position indication as such takes place in the same way as the general position indication based on a set value position.

Note: If the concerned input stays active (only 1 input, not both), then no other position indications are possible. The SDO commands specify set value (object 0x2020) and the drive commands (object 0x204D) are ignored.

Status message
If the function "cycle" is selected and input "F" or "G" activated, the status bit 3 "cycle active" is set. The bit responds with the status of input F or G.
The status bits can be read out via the SDO object 0x204E:01 and are also returned cyclically in TxPDO2.

Key functions in the "cycle" mode

PIN G	PIN F	Cycle active	Function
0	0	No	No position indication. Shaft remains stopped at the current position
0	1	Yes	Position indication at position D1
1	0	Yes	Position indication at position D2
1	1	No	No position indication. Shaft remains stopped at the current position

Debouncing input F and G
The debouncing time of the inputs depends on the program cycle time and is in the range of 390..394 ms.

3.4.3 ExternPosi

The function "ExternPosi" allows in the NM172 and NM174 position indication of the previously set position D3 via the external input F. If input F is activated then the position indication with the signal wing is for position D3.

Enable the Function ExternPosi

For the function "ExternPosi" to be used, it must be enabled (object 0x2041:03 = 2).
The following selection options are available:

0 Function standard / oscillate
1 Function Cycle
2 Function ExternPosi
3 OFF

After delivery and reset to the default values (object 0x2015) the function "ExternPosi" is deactivated. The individual functions cannot be used simultaneously. Switching between functions is possible at any time.

Parameterization of the position value D3

The position value D3 is set or read out via the object 0x2045:03. Storage is only in the RAM. In case of a power failure, the position value is lost. After switching on the device the position value D3 = 0.00 is preset and blocked. When position D3 is blocked, position indication is not possible via the input. The block is cancelled when a value is written to object 0x2045:03.
A transmitted value can be overwritten by a new value at all times, even if no position indication took place for this value.

Note: The position value can be parameterized a random number of times, as it is not stored in EEPROM.

Process of position indication via "ExternPosi"

position value D3 is set to the desired position via the CAN interface through object 0x2045:03.
When activating the input "F" (wing), the position value of D3 is written as the new set value in the top display and the position indication started. The position indication is carried out as long as the input is activated. After the completion of the position indication, the SPA stays at this position and waits for new actions. If the input is deactivated again prior to reaching the target value, the position indication stops.
Upon renewed activation of the input the position indication only continues if D3 has is parameterized first. The position indication as such takes place in the same way as the general position indication based on a set value position.

Note: If the input stays active, then no other position indications are possible. The SDO commands specify set value (object 0x2020) and the drive commands (object 0x204D) are ignored.
Similarly, setting a new value D3 via object 0x2045:03 is blocked.
The command motor stop via object 0x204D:01 is possible at any time.

Status message

If the function "ExternPosi" is selected and input "F" activated, the status bit 4 "ExternPosi active" is set. The bit responds with the status of input F.
The status bits can be read out via the SDO object 0x204E:01 and are also returned cyclically in TxPDO2.

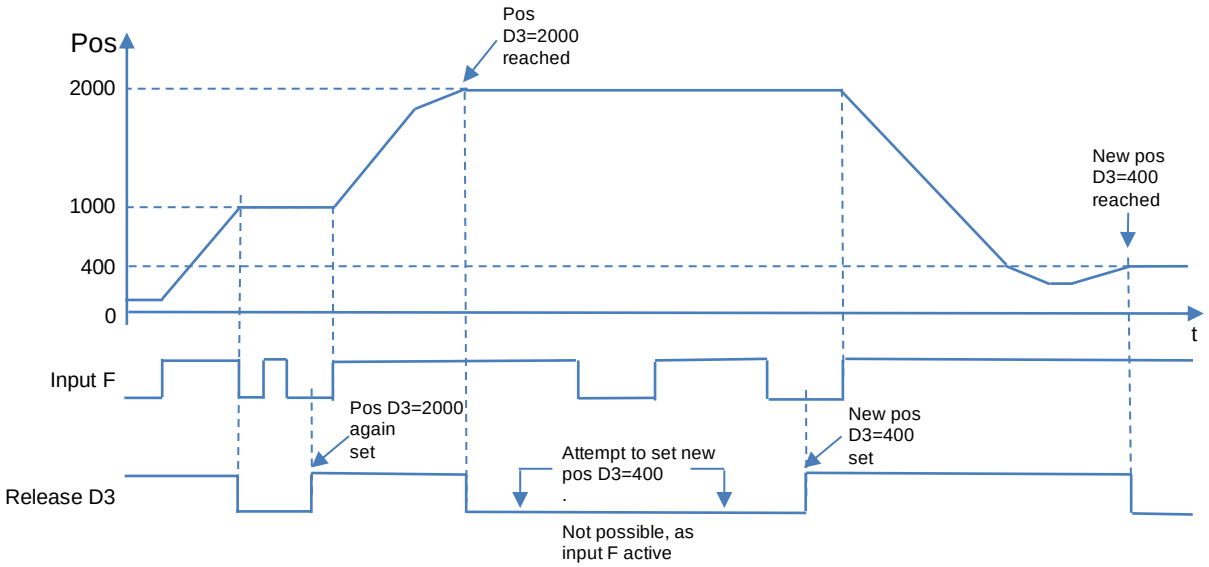
Input function in the mode "ExternPosi"

PIN G	PIN F	Function
x	0	No position indication
x	1	Position indication at position D3 as long as the input is active (must be enabled)

Input G has no effect on this operation type.

Debouncing input F
The debouncing time of the input depends on the program cycle time and is therefore in the range of 16..20 ms.

Example of position indication with several modes:
In this example D3 = 2000 is parameterized at the outset. Following a start via input F, position indication commences until input F is deactivated. The release is also cancelled. Before continuing the position indication, position D3 must be transmitted again to obtain a new release. Following a restart, position 2000 is reached. The release is cancelled again upon reaching the goal. Now a new value D3 = 400 is transmitted and a new position indication is carried out for it.



3.4.4 Off

With NM172 and NM174, the function "Off" deactivates the function of the inputs F and G.
The status of the external inputs is ignored. This selection is identical to selection 0 with the difference that the function "oscillation" is no longer possible via the external inputs.

Enable the function "Off"
For the function "Off" to be used, it must be enabled (object 0x2041:03 = 3)
The following selection options are available:

0 Function standard / oscillate
1 Function Cycle
2 Function ExternPosi
3 OFF

3.5 Data storage

The parameters relevant for the SPA are stored in different internal storage areas. These are:

- RAM Data is lost in case of a power interruption.
- Flash Maximum storage cycles 100,000 (storage for STM controllers)
- EEPROM Maximum storage cycles 1,000,000 (storage for MSP controllers)

The storage areas for the individual parameters are stated in the object description or the object overview.

Data storage process:
Note: The non-volatile storage in Flash or in EEPROM takes place in different ways:

EEPROM:
Parameters that are stored in the EEPROM are always automatically written in the EEPROM immediately after the writing of a parameter. Additional separate storage is not necessary.

Flash:
The parameters that are stored in the Flash storage are initially only stored in the internal RAM. The parameters are transmitted to the Flash storage via the save object 0x1010. If power is interrupted before the save object is carried out, the data previously stored in the Flash becomes valid again.

Setting data to the factory setting:
There are also two different commands available for resetting the parameters to the default values (factory settings).

EEPROM: via object 0x2015
Flash: via object 0x1011

3.6 Block diagram

The SPA contains two microcontrollers:

- **MSP** Power-saving main controller that processes the actual functions of the SPA
- **STM** Interface controller (CAN bus). Creates the connection to the master.

In this document, the two controllers are called MSP and STM.
Figure 1 shows the internal set up for the devices NM170, NM172 and NM174. Please note that not all blocks are contained in all devices.

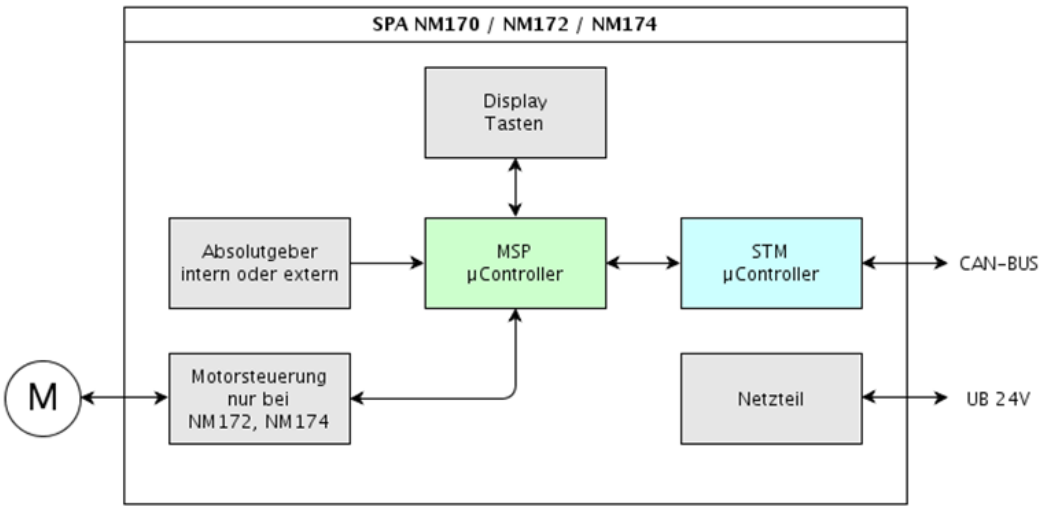


Figure 1: Block diagram overall system

3.7 Timing

3.7.1 Timing - Internal interface

The SPA contains two microcontrollers. The CAN bus data is exchanged between these two controllers MSP and STM (see block diagram Section 3.6) via a serial interface.

The two diagrams 1 and 2 show the cyclical transmission of the data between the two microcontrollers STM and MSP. Communication takes place with a fixed time rhythm of 10ms.
In each cycle, the process data (PD), consisting of *position value*, *status bits,* and *error messages*, is transmitted from the MSP to the STM (diagram 1).
If there is new SDO data (SD), it is transmitted in the gap between two PD transfers from the STM to the MSP (Diagram 2). One SDO command is transmitted per cycle.

Diagram 1: Transmission of the process data (PD) from MSP to STM32.

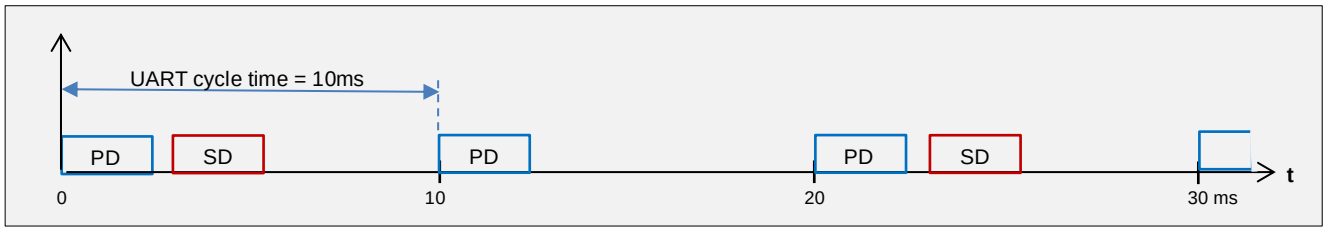


Diagram 2: Additional transmission of the SDO data (SD) to the MSP

In Diagram 2, two exemplary SDO commands that were received approximately 15ms after each other via the CAN bus, are transmitted to the MSP.

SDO Processing (SD):

SDO commands can be transmitted to or requested from the SPA at any speed. The received data is temporarily stored in the Controller STM and can also be read out immediately.

Transmission to the Controller MSP is via a cycle of 10ms (see above). Therefore, a bottleneck may occur if many SDO write commands are transmitted to the same SPA shortly after each other with 125k Baud, for example. Reading SDO commands are not affected by this.

The transmission is handled via a FIFO ring storage with max. 50 entries. If the FIFO is full, the transmission of all entries requires 50 * 10ms = 500ms. The transmission of the 50 SDO commands via CAN bus can be completed after 160ms already at 125k Baud.

4 CAN Bus and CANopen Communication

The CAN bus (CAN: Controller Area Network) was originally developed by Bosch and Intel for the quick cost-effective data transmission in the vehicle technology sector. The CAN bus is today also used in industrial automation.
The CAN bus is a fieldbus (the standards are determined by the CAN in Automation (CiA) group) with which devices, actuators and sensors of various manufacturers communicate with each other.

4.1 CAN bus characteristics

- Data rate of 1MBaud with a network range of up to 25 m
- Closed network on both sides
- The bus medium consists of cables that comply with the standard ISO11898.
- Real-time capacity: defined maximum holding time for high-priority messages.
- Bus participants: in theory 127, physically 120 dependent on the CAN driver.
- Terminating resistance 120 Ω at the beginning and end of the bus system required.
- Ensures data consistency across the network. Defective messages are identified as faulty for all network nodes.
- Message-based communication
  The message is marked with an identifier. Based on the identifier, all network nodes verify whether the message is relevant to them.
- Broadcasting, multicasting
  All network nodes receive every message simultaneously. Therefore, synchronization is possible.
- Multi-master capacity
  Each participant in the field bus can independently transmit and receive data without having to be dependent on the priority of the Masters. Anyone can initiate a message if the bus is not occupied. If messages are transmitted simultaneously, the participant with the highest priority prevails.
- Prioritization of messages
  The identifier determines the priority of the message. This way, important messages can be quickly transmitted via the bus.
- Residual error probability
  Safety processes within the network reduce the probability of an undetected erroneous data transmission to less than 10^{-11}. In effect,100% security of the transmission can be assumed.
- Function monitoring
  Localization of faulty or failed stations. The CAN protocol includes function monitoring of network nodes. Network nodes that are faulty are limited in function or entirely disconnected from the network.
- Data transmission with short error delivery time
  Thanks to several error detection mechanisms, biased messages are detected with a high probability. If an error is detected, the message is automatically retransmitted.

Within the CAN bus, several network participants are connected to each other via a bus cable. Each network participant can transmit and receive messages. The data among the network participants is transmitted serially.

Examples of network participants for CAN bus devices include:

- Automation devices, e.g. SPS
- PCs
- Input/output modules
- Operator controls
- Analyses devices, e.g. a CAN monitor
- Operating and input devices as human machine interface (HMI)
- Sensors and actuators

4.2 CANopen

The CANopen profile was developed under the technical supervision of the Steinbeis Transfer Center for Automation based on the layer 7 specification CAL (CAN Application Layer). Compared to CAL, CANopen only contains the functions suitable for this application. CANopen is therefore a segment of CAL that is optimized for the application, thus allowing simplified system set up and the use of simplified devices. CANopen is optimized for quick data exchange in real-time systems.
The organization CAN in Automation (CiA) is responsible for the applicable standards of the respective profiles.

CANopen allows:

- Easy access to all device and communication parameters
- Synchronization of several devices
- Automatic configuration of the network
- Cyclical and event-controlled process data traffic

CANopen consists of four communication objects (COB) with different characteristics:

- Process data objects for real-time data (PDO)
- Service data objects for parameter and program transmission (SDO)
- Network management (NMT, heartbeat)
- Pre-defined objects (for synchronization, emergency message)

All device and communication parameters are organized in an object directory. An object includes the name of the object, data type, number of sub indexes, structure of the parameters, and the address. According to CiA, this object directory is divided into three different parts: communication profile, device profile, and a manufacturer-specific profile. (see object directory)

4.3 CANopen Communication

4.3.1 Communication profile

The communication between the network participants and the Master (PC/control) is via object directories and objects. The objects are addressed via a 16-bit index. The CANopen communication profile DS 301 standardizes the different communication objects. Accordingly, they are divided into several groups:

- Process data objects (PDO) for the real-time transmission of process data
- Service data objects (SDO) for the read and write access to the object directory
- Objects for the synchronization and error display of CAN participants:
  SYNC object (synchronization object) for the synchronization of network participants
  EMCY object (emergency object) to indicate the fault of a device or its periphery
- Network management (NMT) for initialization and network control
- Layer Setting Services LSS for configuration via the serial number, revisions number, etc. within an existing network.

4.3.2 CANopen message structure

The first part of a message is the COB ID (identifier).

COB ID										
11	10	9	8	7	6	5	4	3	2	1
4 bit function code				7 bit node ID						

The function code provides information about the type of message and the priority
The lower the COB ID, the higher the priority of the message.

Broadcast messages:

Function code	COB ID
NMT	0
SYNC	0x80

Peer-to-peer messages:

Function code	COB ID
Emergency	0x080 + Node ID
PDO1 (tx)[1)]	0x180 + Node ID
PDO1 (rx)[1)]	0x200 + Node ID
PDO2 (tx)[1)]	0x280 + Node ID
PDO2 (rx)[1)]	0x300 + Node ID
SDO (tx)[1)]	0x580 + Node ID
SDO (rx)[1)]	0x600 + Node ID
Heartbeat	0x700 + Node ID
LSS (tx)[1)]	0x7E4
LSS (rx)[1)]	0x7E5

[1)] (tx) and (rx) from the viewpoint of the sensor

The node ID can be freely selected between 1 and 127 via the CANopen bus..
The sensors are transmitted with node ID 127 and a baud rate of 125kBit/s.
Changes are implemented via the service data object 0x2101 or object 0x2100 or via LSS.

A CAN telegram consists of the COB ID, the data length (DLC) and up to 8 bytes of data:

COB ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
xxx	x	xx	xx	xx	xx	xx	xx	xx	xx

The precise telegrams will be presented in greater detail later.

4.3.3 Service data communication (SDO)

The service data objects comply with the standards of CiA. An object can be accessed via the index and sub index. The data can be requested or written in the object if required.

4.3.3.1 General information about SDO

Structure of an **SDO telegram**:

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3

COB ID	An SDO COB ID is structured as follows: Master → SPA: 0x600 + Node ID SPA ← Master: 0x580 + Node ID
DLC	(Data length code) denotes the length of the telegram. It is structured as follows:. 1 byte command + 2 byte object + 1 byte sub index + number of data bytes (0..4)
Command	The commando byte determines whether data is read or set and how many data bytes are involved. See
Object	16-bit object index of the SDO-object (object L and object H)
Sub index	Sub index of the SDO object
Data x	The object data to be transmitted. Data length is variable from 0..4 byte.

Command	Description	Data length	Explanation
0x22	Download request	Max 4 byte	send parameters to SPA
0x23	Download request	4 byte	send parameters to SPA
0x2B	Download request	2 byte	send parameters to SPA
0x2F	Download request	1 byte	send parameters to SPA
0x60	Download response	-	Confirmation of the acquisition by the Master
0x40	Upload request	-	Request parameters from SPA
0x42	Upload response	max 4 byte	parameters to Master with max. 4 byte
0x43	Upload response	4 byte	parameters to Master with 4 byte
0x4B	Upload response	2 byte	parameters to Master with 2 byte
0x4F	Upload response	1 byte	parameters to Master with 1 byte
0x80	Abort message	-	SPA reports error code to Master

4.3.3.2 Abort Message

An Abort Message indicates an error in the CAN communication. The SDO command byte is 80h. Object and sub index are those of the desired object. The error code is found in byte 5..8.

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x580 + Node ID	8	0x80	Object L	Object H	Sub index	ErrByte 0	ErrByte 1	ErrByte 2	ErrByte 3

Byte 8..5 results in the SDO abort message (byte 8 = MSB).
The following messages are supported:

0x05040001 : Command byte is not supported
0x06010000 : Faulty access to an object
0x06010001 : Read access to Write Only
0x06010002 : Write access to Read Only
0x06020000 : Object not supported
0x06060000: SDO hardware or firmware error (internal overflow, overload)
0x06090011 : Sub index is not supported
0x06090030 : value beyond the limits
0x06090031 : Value too large
0x08000000 : General error
0x08000020 : Faulty storage signature ("save")
0x08000021 : Data cannot be stored
0x08000024 : No data available

4.3.3.3 SDO Examples

Example 1: Request of the actual position by the Master from the Slave.
Object = 0x6004; sub index = 0; Actual position = 500000 = 0x**0007A120** (values in hundredths of mm)

Request of the master:

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x600+Node ID	8	0x40	0x04	0x60	0	x	x	x	x

Response of the Slave. Slave provides the actual position (here 0x0007A120)

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x580+Node ID	8	0x43	0x04	0x60	0	0x20	0xA1	0x07	0x00

Example 2: Writing the tolerance window value by the Master to the Slave
Object = 0x2003; sub index = 0; value for tolerance windows = 320d = 0x00000140

Writing of the Master:

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x600+Node ID	8	0x23	0x03	0x20	0	0x40	0x01	0x00	0x00

Response of the Slave to the writing of a value

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x580+Node ID	8	0x60	0x03	0x20	0	0	0	0	0

Example 3: Reading the node ID of the Slave
Object = 0x2101; Sub index = 0; Node ID = 127d = 0x7F

Writing of the Master:

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x600+Node ID	8	0x40	0x01	0x21	0	x	x	x	x

Response of the Slave. Slave transmits the node ID = 7F as a 1 byte value in return (as command = 0x4F).

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x580+Node ID	8	0x4F	0x01	0x21	0	0x7F	0	0	0

Example 4: Writing the parameter display rotate from the Master to Slave
Object = 0x2000; sub index = 5; parameter = 1 (inverted display)

Writing of the Master:

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x600+Node ID	8	0x23	0x00	0x20	0x05	0x01	0x00	0x00	0x00

Response of the Slave to the writing of a value

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x580+Node ID	8	0x60	0x00	0x20	0x05	0	0	0	0

Example 5: Writing an invalid parameter
Writing the parameter display rotate by the Master to the Slave
Object = 0x2000; sub index = 5; parameter = 2. parameter is beyond the permissible range.

Writing of the Master:

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x600+Node ID	8	0x23	0x00	0x20	0x05	0x02	0x00	0x00	0x00

Response of the Slave to the writing of the invalid value

COB ID	DLC	Command	Object L	Object H	Sub index	Data 0	Data 1	Data 2	Data 3
0x580+Node ID	8	**0x80**	0x00	0x20	0x05	0x30	0x00	0x09	0x06

The Slave responds with an abort message (Command = 0x80).
Error code = 0x06090030: value beyond the limits

4.3.4 Process data communication

4.3.4.1 General description

Process data objects serve the exchange of process data in real time, for example the position value or operational mode. PDOs are transmitted synchronous or cyclical (asynchronous).

Synchronous
To transmit the process data synchronous, a value between 1 and 0xF0 (=240) must be written in object 0x1800 sub index 2. If value 3 is entered, the PDO is transmitted to every third sync telegram. With value 1 it will be transmitted to every sync telegram.
In synchronous operation, the PDO is requested by the Master via the sync telegram:

COB ID	DLC	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
0x80	1	0	-	-	-	-	-	-	-

Cyclical (asynchronous)
If the PDOs are to be transmitted cyclically then the value 0xFE or 0xFF should be written in object 0x1800 sub index 2. In addition, in the same object sub index 5 the cycle time must be entered in milliseconds. The entered time is rounded up to 1ms. If the value 0ms is stored the PDOs are not transmitted. The function is switched off.

The following table sums up the different transmission modes of the PDOs:

Object 0x1800		**Brief description**
Sub index 2	**Sub index 5**	
FEh	3ms	Cyclical transmission every 3 ms
FEh	0ms	PDO transmission switched off
3	xxx	Send with every third Sync telegram

Structure of a PDO telegram:
The basic structure is identical in all used PDOs.

COB ID	DLC	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8

COB ID	A PDO COB-ID is structured as follows: TxPDO1: 0x180 + Node ID RxPDO1: 0x201 [1] (per default to 0x201 for SPAs in the network) TxPDO2: 0x280 + Node ID RxPDO2: 0x300 + Node ID
DLC	(Data length code) denotes the length of the telegram. This consists of the transmitted data bytes data 1 to data 8
Data n	The PDO data to be transmitted. The data length is fixed at 8 byte for all used PDOs.

[1] RxPDO1 is a special case. This PDO is used as a broadcast channel for SDO telegrams. This means that an SDO command is packed into the PDO and can then be sent via a send telegram to all participants in the network. Therefore, the COB ID must be identical on all devices and is therefore parameterized to 0x201 per default. The setting remains unchanged if the node ID is changed. However, it is possible to change it directly.

4.3.4.2 RxPDO1

Receive PDO1 is a PDO for the simultaneous transmission (broadcast) of individual SDO commands to all connected SPAs. The COB ID of the RxPDO1 (object 0x1400:01) is fixed at 0x201 for all SPAs. This way, all devices receive the PDO message and carry it out.

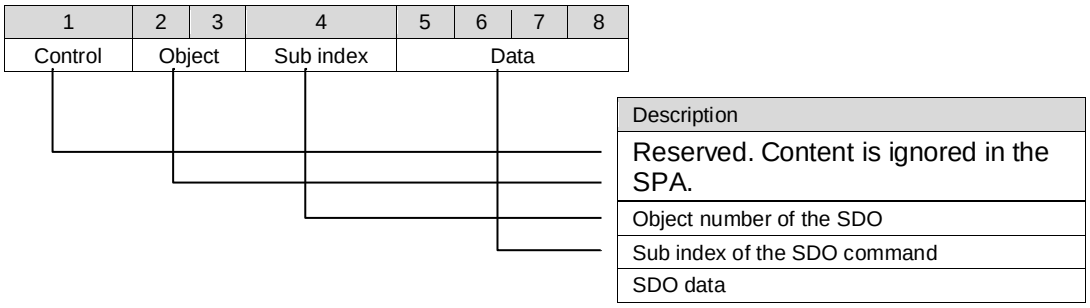
List of SDO commands:
The following lists shows the SDO commands, that can be transmitted as a broadcast via RxPDO1.

Object	Sub index	Description
0x2001	0	Conversion mm/inch
0x2008	0	Display device address
0x204D	1	Drive command [1)]
0x204D	2	Oscillating mode start/stop [1)]
0x2015	0	Restore (reset) MSP
0x6003	0	Preset

[1)] only for NM172, NM174

When there is an attempt to transmit an SDO command other than those listed above, the SPA replies with the emergency message 0x8220 sub-code 01 (see section 6.1).

Data RxPDO1:



A received SDO command is carried out immediately upon receipt. The setting in object 0x1400:02 is ignored. The implementation, as standard for PDOs, only in the *Operational* status.

Broadcast example:
The following example sends the SDO command 0x2001 for converting the display to the inch format. Values in a hexadecimal format:

1	2	3	4	5	6	7	8
Control	Object		Sub index	Data			
00	**01**	**20**	**00**	**01**	**00**	**00**	**00**

4.3.4.3 RxPDO2

Receive PDO2 is an object for the control of a position indicator. It transmits the set value, a potential actual value correction and a control word for the control of different position indicators.

Data RxPDO2:

1	2	3	4	5	6	7	8
Set value				Actual value correction		Control	

Set value	Description
-99999..999999	The set value is the set position to which SPA should be repositioned. The set value is related to the center of the shaft of the used tool.

Actual value correction	Description
-32767..32767 -32768	The actual value correction (offset) moves the actual value (bottom display line) by the stated value.The set value shown on the display (top display line) is offset against this offset value. Calculation see below: depending on on the set value and actual value correction. The received offset value (16 bit) overwrites the value in object 0x2010 (the top 16 bit are deleted). This value serves as a switch of the actual value correction from 16 bit to 32 bit. If -32768 is received, the value in object 0x2010 is no longer replaced by this value. The offset value available in object 0x2010 will be used. If a change of the offset value is required, the object 0x2010 must now be set separately by the corresponding SDO command before transmitting the RxPDO2. As soon as an actual value correction > -32768 is transmitted in RxPDO2 this value will once again be written in object 0x2010. In TxPDO2 the value received here is always transmitted back. See also 4.3.4.5 TxPDO2 and Section 4.4.5.10 actual value correction.

Control bits	Description
0	1: Enable. The concerned command is carried out. Note: Motor stop is always carried out, even when enable = 0.
1	1: With release. Motor starts immediately. If bit 9=1 then: 0: clamping open 1: clamping closes
2	1: Without spindle compensation
3..4	01b..11b: Speed for command cutting in and pressing
5..6	00b: No function 01b: Write set value at position value D3 (object 0x2045:03) 10b: reserved 11b: reserved
7	1: Press right. With default setting motor drives to the right, counting upwards.
8	1: Press left. With default setting motor drives to the left, counting downwards
9	Clamping on/off. only enabled in operating mode *with clamping* (object 0x2047:03)
10..13	Reserved
14	1: Motor stop. Motor is stopped. New set value, actual value correction and the other control bits are ignored.
15	Toggle bit. Only for Masters to enforce a change of the PDO data for the automatic sending of an identical data set. Ignored by the SPA.

Jog function (bit 7,8):
Jog functionality enabledy using two buttons. One-time transmissions are get the same reaction as when pressing button and hold.

Motor stops at stop command (bit 14) in the control register.
Automatic motor stop when reaching the final stop positions.

In operating mode "with clamping", travel usin the buttons and time-relayed clamping enable/disable is performed before motor start respectively after motor stop.

The two parameters *counting direction* (object 0x2000:02) and *direction of motor rotation* (object 0x2040:02) invert the of motor's rotational direction. Any parameter change should be in standstill/I dle state since in a running jog function the change is directly adopted for the motor's direction of rotation.

The Jog function ignores both setpoint and actual value correction. Consequently, objects 0x2010 and 0x2020 will remain unchanged..

Execute Jog function will interrupt any another, previously started but still active positioning operation

Function clamping on/off (bit 9):
Only enabled with active clamping feature. Clamping enable is in object 0x2047:03. In operating mode "clamping disabled" command (bit 9) has no effect. In this case no error message will be created either.

Command (bit 9) is ignored in a running positioning operation (automatic or manual via button). The moving bit is being set. Please see object 0x204E:01 for the moving bit function.

Clamping open and close is immediately adopted if moving bit = 0.

Function stop motor (bit 14):
Motor stops immediately with bit 14 being set in the control register. The remaining bits can be used as requested. Setpoint and actual value correction, if set, will be ignored.

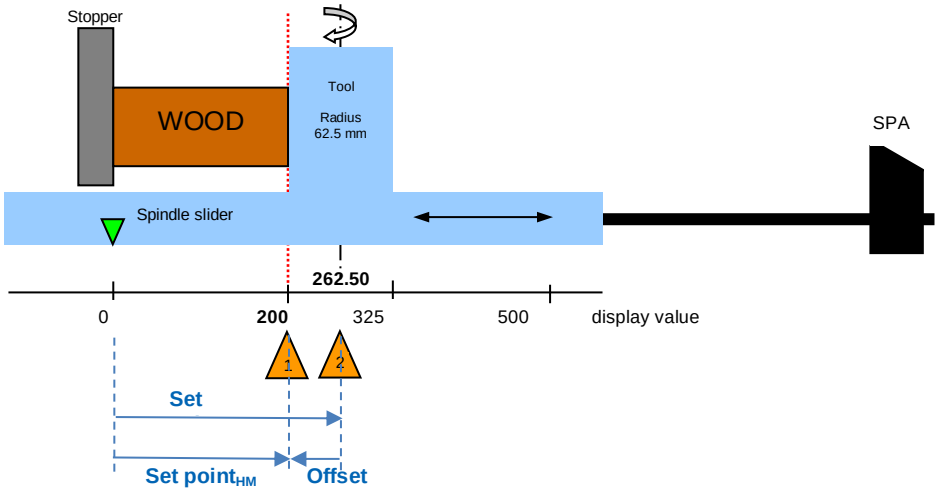
Consequently, function *Stop motor* via PDO can be used for EMERGENCY STOP.

Permissible control bit combinations:
Not all control bit combinations are possible. In case of invalid combinations the emergency message: 0x8230 is sent (see Section 6.1).

The following table lists all possible functions:

No.	Control bits													Function
	15	14	13..10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	No function. Bite combination is permissible. PDO is ignored by the SPA. No emergency message is sent either.
1	X	0	X	0	0	0	0	1	X	X	X	X	1	Writing the set value in the position value for the function ExtPosi D3 (0x2045:03) + actual value correction
2	X	0	X	0	0	0	1	0	X	X	X	X	1	reserved
3	X	0	X	0	0	0	1	1	X	X	X	X	1	reserved
4	X	0	X	0	0	0	0	0	X	X	1	1	1	Set value setting directly with motor release and position indication without loop drive (0x2020:05) + actual value correction
5	X	0	X	0	0	0	0	0	0	0	0	0	1	Standard set value setting without release (0x2020:01) + actual value correction
6	X	0	X	0	0	0	0	0	0	0	0	1	1	Standard set value setting with release (0x2020:02) + actual value correction
7	X	0	X	0	0	0	0	0	0	1	0	0	1	Set value cutting in without release (0x2020:06) + Slow speed for cutting in function (0x2020:08) + actual value correction
8	X	0	X	0	0	0	0	0	1	0	0	0	1	Set value cutting in without release (0x2020:06) + Medium speed for cutting in function (0x2020:08) + actual value correction
9	X	0	X	0	0	0	0	0	1	1	0	0	1	Set value cutting in without release (0x2020:06) + Fast speed for cutting in function (0x2020:08) + actual value correction
10	X	0	X	0	0	0	0	0	0	1	0	1	1	Set value cutting in with release (0x2020:07) + Slow speed for cutting in function (0x2020:08) + actual value correction
11	X	0	X	0	0	0	0	0	1	0	0	1	1	Set value cutting in with release (0x2020:07) + Medium speed for cutting in function (0x2020:08) + actual value correction
12	X	0	X	0	0	0	0	0	1	1	0	1	1	Set value cutting in with release (0x2020:07) + Fast speed for cutting in function (0x2020:08) + actual value correction
13	X	0	X	0	0	1	0	0	0	1	0	0	1	Pressing right at slow speed
14	X	0	X	0	0	1	0	0	1	0	0	0	1	Pressing right at medium speed
15	X	0	X	0	0	1	0	0	1	1	0	0	1	Pressing right at high speed
16	X	0	X	0	1	0	0	0	0	1	0	0	1	Pressing left at slow speed
17	X	0	X	0	1	0	0	0	1	0	0	0	1	Pressing left at medium speed
18	X	0	X	0	1	0	0	0	1	1	0	0	1	Pressing left at high speed
19	X	0	X	1	0	0	0	0	0	0	0	0	1	Clamping disable (open)
20	X	0	X	1	0	0	0	0	0	0	0	1	1	Clamping enable (close)
21	X	1	X	X	X	X	X	X	X	X	X	X	X	Stop motor

Depending on the set value and actual value correction:



Formula for calculating the set point$_{HM}$:

$$\text{Set point}_{HM} = \text{Set point}_{AM} + \text{Offset} = 262{,}50\ \text{mm} + (-62{,}50\ \text{mm}) = 200\ \text{mm}$$

Legend:
Set point$_{AM}$: Dimension from the center of the shaft (bradcast set value)
Set point$_{HM}$: Wood dimension (set value shown in the display)
Offset: Actual value correction

4.3.4.4 TxPDO1

TxPDO1 is deactivated upon delivery (factory setting) and deactivated after a *Restore Parameter*. The release is via object 0x1800 sub index 1.

Data TxPDO1:

1	2	3	4	5	6	7	8
Position value Object 0x6004:00				Status Object 0x204E:01		Errors Object 0x204E:02	

4.3.4.5 TxPDO2

TxPDO2 is always active upon delivery (factory setting) and after a *Restore Parameter*.

Data TxPDO2:

1	2	3	4	5	6	7	8
Position value Object 0x6004:00				Actual value correction Object 0x2501:04		Status object 0x204E:01	

Actual value correction:
Returns the actual value correction received from RxPDO2 or object 0x2010. The actual value correction received via RxPDO2 is transmitted to object 0x2010.
However, object 0x2010 is a 32-bit object. In this case only 16 bit can be transmitted.
For values > 32767 or < -32768 in object 0x2010 the respective greatest (32767) or smallest 16-bit value (-32768) is transmitted. This indicates overflow in the positive or negative region. It is then mandatory to read object 0x2010 to receive the correct actual value correction.
See also Section 4.3.4.3 RxPDO2 and Section 4.4.5.10 Actual value correction

4.3.5 Emergency service

Internal device errors or bus problems actuate an emergency message.
For structure and messages see Section 6.1

4.3.6 Network management services (NMT)

Network management can be divided into two groups:
With the NMT services for **device control**, the bus participants can be initialized, started, and stopped.
In addition, there are NMT services for **connection monitoring**

Description of the NMT commands
The commands are transmitted as unconfirmed objects and are structured as follows:

COB ID	Byte 1	Byte 2
0	Command byte	Node number

COB ID for NMT commands is always zero. The node ID is transmitted to byte 2 of the NMT command.

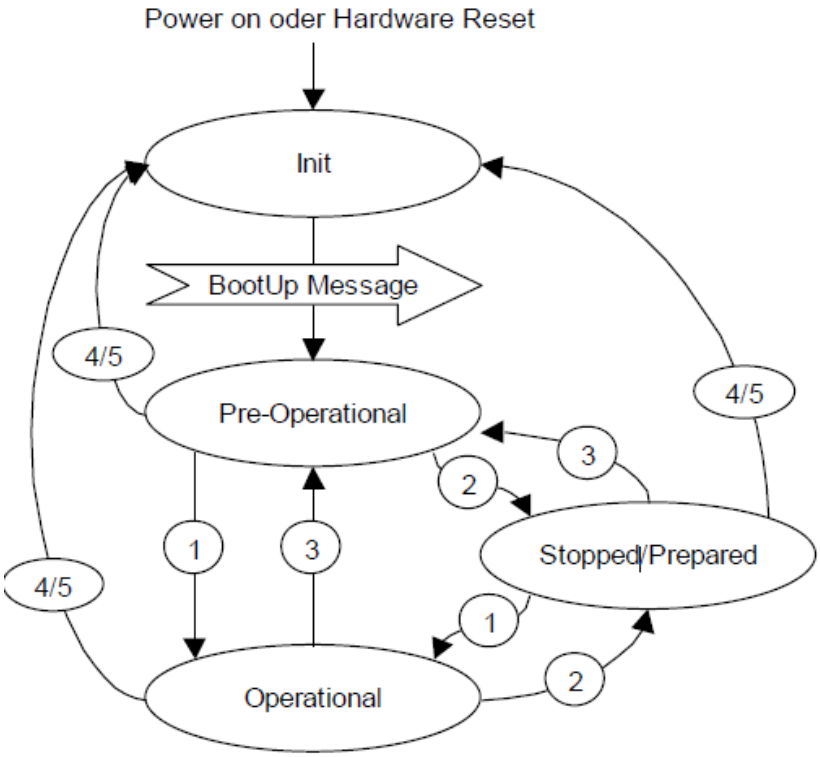
Command byte

Command byte	Description	In state event drawing
0x01	Start remote node	1
0x02	Stop remote node	2
0x80	Enter pre-operational mode	3
0x81, 0x82	Reset remote node	4, 5

The **node number** corresponds to the node ID of the desired participant. Node number = 0 addresses all participants.

4.3.7 NMT State Event

Following initialization, the spindle position display is in the pre-operational mode. In this mode, it is possible to read and write SDO parameters. To request PDO parameters, the spindle position display must first be set to the operational mode.



4.3.7.1 The different NMT modes

Init Mode

Following initialization, the spindle position display contacts the CAN bus with a BootUp message.
Then the spindle position display automatically assumes the pre-operational mode.
The COB ID of the BootUp message consists of 700h and the node ID.

COB ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x700 +node ID	1	0x00	-	-	-	-	-	-	-

Pre-operational mode

In the pre-operational mode, it is possible to read and write SDO.

Operational Mode

In the operational mode, the spindle position display sends the desired PDOs. In addition, reading and writing SDOs is possible.

Stopped or Prepared Mode

In the Stopped Mode only NMT communication is possible. SDO parameters cannot be read or set.

Note: LSS communication, for example for addressing, is possible in all modes.

4.3.7.2 Mode Change

Mode changes can be carried out separately for individual devices (node ID = 1..127) or simultaneously for all devices included in the network (node ID = 0).

Start Remote Node (1)

The start command sets the spindle position display to the operational mode.

COB ID	DLC	Command byte	Node ID	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0	2	**0x01**	0..127	-	-	-	-	-	-

Stop Remote Node (2)

The stop command sets the spindle position display to the stopped or prepared mode

COB ID	DLC	Command byte	Node ID	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0	2	**0x02**	0..127	-	-	-	-	-	-

Enter Pre-Operational Mode (3)

Switch to pre-operational mode.

COB ID	DLC	Command byte	Node ID	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0	2	**0x80**	0..127	-	-	-	-	-	-

Reset Remote Node (4) or Reset Communication (5)

With the reset command the SPA is newly initialized.
Reset Remote Node (4):

COB ID	DLC	Command byte	Node ID	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0	2	**0x81**	0..127	-	-	-	-	-	-

Reset communication (5):

COB ID	DLC	Command byte	Node ID	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0	2	**0x82**	0..127	-	-	-	-	-	-

4.3.8 Life Guarding

4.3.8.1 Heartbeat Protocol - General Information

The heartbeat protocol is used for monitoring the individual bus participants or the Master. The heartbeat function recognizes if one or more bus participants are switched off or defect and therefore no interface communication is taking place.

A "heartbeat producer" cyclically produces a heartbeat message. One or more "heartbeat consumers" can receive this heartbeat message. If the heartbeat message fails, the Master or the Slave, depending on the heartbeat variation, may assume a secure mode or potentially initiate measures for ramification.

4.3.8.2 Producer Heartbeat

The Master monitors the mode of the SPA through the producer heartbeat. As a heartbeat producer, the SPA sends its NMT status cyclically. The cycle time of the heartbeat message is set via the object 0x1017. A cycle time of 0 deactivates the heartbeat protocol. In the default setting the producer heartbeat is deactivated.

COB ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x700 + Node ID	1	**NMT mode**	-	-	-	-	-	-	-

NMT mode	Description
0	BootUp event
4	Stopped
5	Operational
127	Pre-operational

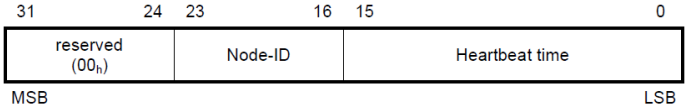
4.3.8.3 Consumer heartbeat

With the consumer heartbeat the SPA expects an external heartbeat protocol in a specific cycle period. The time interval in which a heartbeat is expected can be entered into object 0x1016:01.

If the SPA does not receive a heartbeat protocol during the time interval, a "heartbeat event" is actuated. The behavior in case of error is defined in the object 0x1029:01 "Communication Error".
Any active automatic position indication is stopped. An emergency message 0x8130 heartbeat error is sent one time.

Monitoring of the consumer heartbeat in the device takes place as soon as the first heartbeat protocol is received. From this time one, the subsequent protocols must be received within the set time interval.

Object 0x1016:01:



Heartbeat time: Heartbeat interval in ms.
Node ID: Node ID of the heartbeat protocol, which is received and evaluated.
This node ID may differ from the device node ID. This way, the
same node ID can be used multiple ways and thus link it with the same heartbeat message.

For more information, see section 4.4.4.12.

4.3.9 Layer Setting Services (LSS)

4.3.9.1 General

For unique addressing, devices require a node number (node ID) in the range from 1 to 127 and an identical baud rate for all devices. The layer setting services (LSS) defined in CiA-305 allow allocation of the node number and baud rate can be dynamically allocated via CANopen.

As standard, the SPA is transmitted with the node ID 127 and baud rate of 125 kBaud. Several SPAs can be connected to a bus system with the same node ID. LSS is now used to address the individual SPAs. For a detailed example of addressing (allocation node ID) via LSS, see Section 3.1 Addressing.

LSS can be implemented in the **Stopped Mode, PreOp Mode a**nd **Operational Mode**.

←

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

COB ID	Master → SPA : 0x7E5 SPA ← Master : 0x7E4
DLC	(Data length code) denotes the length of the telegram. It is structured as follows: 1 byte CS + n byte data. Data length varies according to the LSS command specifier (CS)
CS	LSS command specifier
Byte n	LSS data. 1..7 Data bytes.

4.3.9.2 LSS Switch Modes

To reconfigure individual devices, they must be set to the configuration mode. This takes place with the *Switch Mode* commands.

Switch Mode Global

Changes the mode of all devices in the network. In practical application, this is only useful for exiting the configuration mode and if there is only 1 device in the network.

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	2	0x04	Mode	-	-	-	-	-	-

Mode: 0 switches the device to the operation mode (standard operation mode)
1 → switches the device to the configuration mode

Switch Mode Selective

Changes the mode of only one device in the network. This takes place with the complete LSS address, consisting of : *vendor ID*, *product code*, *revision number*, *serial number* (object 0x1018).

With the following sequence a very specific SPA can be addressed in the bus system

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	8	0x40	Vendor ID				0	0	0
0x7E5	8	0x41	Product code				0	0	0
0x7E5	8	0x42	Revision number				0	0	0
0x7E5	8	0x43	Serial number				0	0	0
0x7E4	8	0x44	Mode	0	0	0	0	0	0

Vendor ID: 0xEC for Baumer products
Product code: Internal product code for the concerned SPA
Revision number: Current revision number of the SPA
Serial number: unique, continuous serial number
Mode: Reply of the SPA is the new mode (0= operation mode; 1=configuration mode)

4.3.9.3 Setting the node ID and bit rate (baud rate)

If the device is in the configuration mode, the node ID and baud rate can be newly set via LSS commands.

Setting the node ID

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	2	0x11	Node ID	-	-	-	-	-	-
0x7E4	8	0x11	Error code	0	0	0	0	0	0

Node ID: New node ID of the SPA
ErrorCode: Feedback of the SPA
0: OK (no error)
1: Node ID outside the range
2..254: reserved
255: Application-specific error code in the other bytes

Setting BitTiming (baud rate):

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	2	0x13	TableSel	TableInd	-	-	-	-	-
0x7E4	8	0x13	Error code	0	0	0	0	0	0

TableSel: Selects the bit rate table:
0: Standard CiA bit timing table
1..127 : reserved for CiA
128..255: reserved for manufacturer-specific tables (none defined)
TableInd: Bitrate entry in select table (see table below)
ErrorCode: Feedback of the SPA
0: OK (no error)
1: Bit rate outside the range
2..254: reserved
255: Application-specific error code in the other bytes

Bit rate table for TableSel 0 (standard CiA bit timing table):

Baud rate	**Table index**
1000 k baud	0
800 k baud	1
500 k baud	2
250 k baud	3
125 k baud	4
100 k baud	5
50 k baud	6
20 k baud	7
10 k baud	8
Automatic bit rate detection	**9 (not implemented)**

Storing the configuration parameters

This protocol stores the transmitted configuration parameters.

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	2	0x17	0x00	-	-	-	-	-	-
0x7E4	8	0x17	Error code	0	0	0	0	0	0

Error code: Feedback of the SPA
- 0: OK (no error)
- 1: Storage not supported
- 2: Access error
- 3..254: reserved
- 255: Application-specific error code in the other bytes

4.3.9.4 Activate bit rate parameters

The new bit rate parameters are activated with the command specifier (Cs) 0x15.

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	3	0x15	Switch Delay		-	-	-	-	-

Switch Delay: Delay in the reset in the Slave in ms
After the delay period, the SPA signs up with the new baud rate.

4.3.9.5 Request LSS address

For a selected SPA, the following protocols deliver the individual parameters *vendor ID*, *product code*, *revision number*, *serial number* of the LSS address in object 0x1018.

Vendor ID:

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	1	0x5A	-	-	-	-	-	-	-
0x7E4	8	0x5A	Vendor ID (always 0xEC)				0	0	0

Product code:

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	1	0x5B	-	-	-	-	-	-	-
0x7E4	8	0x5B	Product Code				0	0	0

Revision number:

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	1	0x5C	-	-	-	-	-	-	-
0x7E4	8	0x5C	Revision number				0	0	0

Serial number:

COB ID	DLC	CS	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x7E5	1	0x5D	-	-	-	-	-	-	-
0x7E4	8	0x5D	Serial number				0	0	0

Serial number: unique, continuous serial number of the SPA

4.4 Object directory

According to CiA (CAN in Automation) the objects are divided into three groups:

- **Communication objects**
  0x1000 – 0x1FFF
- **Manufacturer-specific objects**
  0x2000 – 0x5FFF
- **Device profile specific objects**
  0x6000 – 0x9FFF

The following tables show a summary of all SDO objects supported by the SDA.
The table columns are structured as follows:

Column headlines	Description
Object	Object number in hex
Sub index	Sub index of the respective object in hex
Name	Denotation of the object
Format	uint/int = unsigned/signed integer, number = number of bits, string = string of characters
Access	ro = read only, wo = write only, rw = read write
Default	Default value at the first init or restore
Measurement range	permissible value range of the respective parameter
Save	RAM/FLASH/EEPROM = Storage location in which parameters are stored
Description	Link for the description of the concerned object

4.4.1 Object overview – communication objects

Object Sub index	Name	Format	Access	Default	Measurement range	Save	Description Section/link
0x1000	Device type	uint32	ro	0x20196	0x00020196	-	4.4.4.1
0x1001	Error register	uint8	ro	0x00	0..uint8	-	4.4.4.2
0x1002	Manufacturer status register	uint32	ro	-	0..uint16	-	4.4.4.3
0x1003	Pre-defined error field						4.4.4.4
00h	Number of emergency messages	uint8	rw	0	0 .. 8	-	4.4.4.4
01h-08h	Occurred emergency messages	uint32	ro	-	-	-	4.4.4.4
0x1005	Sync COB ID	uint32	rw	0x00	0..uint32	FLASH	4.4.4.5
0x1008	Manufacturer device name	string	ro	-	Fixed text	-	4.4.4.6
0x1009	Manufacturer hardware version	string	ro	-	Fixed text	-	4.4.4.7
0x100A	Manufacturer software version	string	ro	-	Fixed text	-	4.4.4.8
0x1010	Store parameters						4.4.4.9
00h	Largest sub index	uint8	ro	0x02	2	-	4.4.4.9
01h	Save all parameters	uint32	rw	1	0x65766173	FLASH	4.4.4.9
02h	Save communication parameters	uint32	rw	1	0x65766173	FLASH	4.4.4.9
0x1011	Restore parameters						4.4.4.10
00h	Largest sub index	uint8	ro	0x02	2	-	4.4.4.10
01h	Restore all parameters	uint32	rw	1	0x64616F6C	FLASH	4.4.4.10
02h	Restore communication parameters	uint32	rw	1	0x64616F6C	FLASH	4.4.4.10
0x1014	Emergency COB ID	uint32	rw	0x80 + Node ID	0x81 .. 0x00FF	FLASH	4.4.4.11
0x1016	Consumer heartbeat time						4.4.4.12
00h	Largest sub index	uint8	ro	0x01	1	-	4.4.4.12
01h	Consumer heartbeat time	uint32	rw	0	0..uinit32	FLASH	4.4.4.12
0x1017	Producer heartbeat time	uint16	rw	0	0..65535	FLASH	4.4.4.13
0x1018	Identity object						4.4.4.14
00h	Largest sub index	uint8	ro	0x04	4	-	4.4.4.14
01h	Vendor ID	uint32	ro	0xEC	0x000000EC	-	4.4.4.14
02h	Product Code	uint32	ro	at the factory	0x70 or 0x71 or 0x72	-	4.4.4.14
03h	Revision number	uint32	ro	at the factory	1..0xFFFFFFFF	-	4.4.4.14
04h	Serial number	uint32	ro	at the factory	1..0xFFFFFFFF	-	4.4.4.14
0x1029	Error behavior						4.4.4.15
00h	Largest sub index	uint8	ro	0x02	2	-	4.4.4.15
01h	Communication error	uint8	rw	0	0..2	FLASH	4.4.4.15
02h	Manufacturer specific error	uint8	rw	1	0..2	FLASH	4.4.4.15
0x1400	Receive PDO1 parameter						4.4.4.16
00h	Largest sub index	uint8	ro	0x02	2	-	4.4.4.16
01h	COB ID	uint32	rw	0x40000201	0x40000200 + Node ID	FLASH	4.4.4.16
02h	Transmission type	uint8	rw	0xFF	0xFF	-	4.4.4.16
0x1401	Receive PDO2 parameter						4.4.4.17
00h	Largest sub index	uint8	ro	0x04	4	-	4.4.4.17
01h	COB ID	uint32	rw	0x4000037F	0x40000300 + Node ID	FLASH	4.4.4.17
02h	Transmission type	uint32	rw	0xFF	0xFF	FLASH	4.4.4.17
0x1600	Receive PDO1 mapping						4.4.4.18
00h	Largest sub index	uint8	ro	0x04	4	-	4.4.4.18
01h	Broadcast control	uint32	ro	0x25000108	0x25000108	-	4.4.4.18
02h	Broadcast object	uint32	ro	0x25000210	0x25000210	-	4.4.4.18
03h	Broadcast sub index	uint32	ro	0x25000308	0x25000308	-	4.4.4.18
04h	Broadcast data	uint32	ro	0x25000420	0x25000420	-	4.4.4.18
0x1601	Receive PDO2 mapping						4.4.4.19
00h	Largest sub index	uint8	ro	0x03	3	-	4.4.4.19
01h	Position indicator– set value	uint32	ro	0x25010120	0x25010120	-	4.4.4.19
02h	Position indicator - actual value correction	uint32	ro	0x25010210	0x25010210	-	4.4.4.19
03h	Position indicator – control	uint32	ro	0x25010310	0x25010310	-	4.4.4.19

Object Sub index	Name	Format	Access	Default	Measurement range	Save	Description Section/link
0x1800	Transmit PDO1 parameters						4.4.4.20
00h	Largest sub index	uint8	ro	0x05	5	-	4.4.4.20
01h	COB ID	uint32	rw	0xC00001FF	0xC0000180 + node ID	FLASH	4.4.4.20
02h	Transmission type	uint8	rw	0xFF	0x01 .. 0xF0, 0xFE, 0xFF	FLASH	4.4.4.20
03h	Inhibit time	uint16	rw	80	0..65535	FLASH	4.4.4.20
05h	Event time	uint16	rw	1000	0..65535	FLASH	4.4.4.20
0x1801	Transmit PDO2 parameters						4.4.4.21
00h	Largest sub index	uint8	ro	0x05	5	-	4.4.4.21
01h	COB ID	uint32	rw	0x400002FF	0x40000280 + Node ID	FLASH	4.4.4.21
02h	Transmission type	uint8	rw	0xFF	0x01 .. 0xF0, 0xFE, 0xFF	FLASH	4.4.4.21
03h	Inhibit time	uint16	rw	80	0..65535	FLASH	4.4.4.21
05h	Event time	uint16	rw	1	0..65535	FLASH	4.4.4.21
0x1A00	Transmit PDO2 mapping						4.4.4.22
00h	Largest sub index	uint8	ro	0x03	3	-	4.4.4.22
01h	Object 0x6004 (position)	uint32	ro	0x60040020	0x60040020	-	4.4.4.22
02h	Object 0x204E:01 (SPA status)	uint32	ro	0x204E0110	0x204E0110	-	4.4.4.22
03h	Object 0x204E:02 (SPA errors)	uint32	ro	0x204E0210	0x204E0210	-	4.4.4.22
0x1A01	Transmit PDO1 mapping						4.4.4.23
00h	Largest sub index	uint8	ro	0x03	3	-	4.4.4.23
01h	Object 0x6004 (position)	uint32	ro	0x60040020	0x60040020	-	4.4.4.23
02h	Object 0x2501:04 (actual value correction)	uint32	ro	0x25010410	0x25010410	-	4.4.4.23
03h	Object 0x204E:01 (SPA status)	uint32	ro	0x204E0110	0x204E0110	-	4.4.4.23
0x1F80	NMT startup	uint32	rw	0	0; 8	FLASH	4.4.4.24

4.4.2 Object overview – manufacturer-specific objects

Object Sub index	Name	Format	Access	Default	Measurement range	Save	Description Section/link
0x2000	General basic settings						4.4.5.1
00h	Largest sub index	uint8	ro	08h	8	-	4.4.5.1
01h	Position indication direction	uint32	rw	0	0..1	EEPROM	4.4.5.1
02h	Counting direction	uint32	rw	0	0..1	EEPROM	4.4.5.1
03h	Arrows	uint32	rw	0	0..5	EEPROM	4.4.5.1
04h	Rounded actual value	uint32	rw	0	0..1	EEPROM	4.4.5.1
05h	Rotate display	uint32	rw	0	0..1	EEPROM	4.4.5.1
06h	Actual value correction off/on	uint32	rw	1	0..1	EEPROM	4.4.5.1
07h	Hide set value	uint32	rw	0	0..2	EEPROM	4.4.5.1
08h	External inputs	uint32	rw	0	0..3	EEPROM	4.4.5.1
0x2001	Length unit mm/inch	uint32	rw	0	0..1	EEPROM	4.4.5.2
0x2002	Loop for spindle play compensation	uint32	rw	100	0..9999	EEPROM	4.4.5.3
0x2003	Tolerance window	uint32	rw	5	0..9999	EEPROM	4.4.5.4
0x2004	Scaling (spindle gradient)	uint32	rw	1736111	0..99999999	EEPROM	4.4.5.5
0x2005	Special parameters						4.4.5.6
00h	Largest sub index	uint8	ro	03h	3	-	4.4.5.6
01h	Filter width	uint32	rw	100	0..2303	EEPROM	4.4.5.6
02h	Filter number	uint32	rw	5 / 2	0..99	EEPROM	4.4.5.6
03h	LCD digit on/off	uint32	rw	0	0..1	EEPROM	4.4.5.6
0x2006	Device data MSP controller						4.4.5.7
00h	Largest sub index	uint8	ro	02h	2	-	4.4.5.7
01h	Version	uint32	ro	-	104 or higher	-	4.4.5.7
02h	Device type	uint32	ro	-	fixed value	-	4.4.5.7
0x2007	Display message	string	rw	"123456 7890AB"	String with 12 characters	RAM	4.4.5.8
0x2008	Display address	uint32	wo	-	0..1	-	4.4.5.9
0x2010	Actual value correction	int32	rw	0	-99999..999999	EEPROM	4.4.5.10
0x2015	Reset SPA	uint32	wo	-	1..6	EEPROM	4.4.5.11
0x2020	Read write set value						4.4.5.12
00h	Largest sub index	uint8	ro	08h	8	-	4.4.5.12
01h	Set value (S)	int32	rw	-	-99999..999999	EEPROM	4.4.5.12
02h	Set value (SF) with motor release	int32	rw	-	-99999..999999	EEPROM	4.4.5.12
03h	Set value direct (SD)	int32	rw	0	-99999..999999	RAM	4.4.5.12
04h	Set value direct (SDF) with motor release	int32	rw	0	-99999..999999	RAM	4.4.5.12
05h	Set value direct (SEF) with motor release, without play compensation	int32	rw	0	-99999..999999	RAM	4.4.5.12
06h	Set value cutting in (SL)	int32	rw	0	-99999..999999	RAM	4.4.5.12
07h	Set value cutting in (SLF) with motor release	int32	rw	0	-99999..999999	RAM	4.4.5.12
08h	Speed for function Cutting in	uint32	rw	0	0..2	EEPROM	4.4.5.12
0x2040	Motor basic settings						4.4.5.13
00h	Largest sub index	uint8	ro	04h	4	-	4.4.5.13
01h	Key allocation	uint32	rw	0	0..1	EEPROM	4.4.5.13
02h	Motor rotation direction	uint32	rw	0	0..1	EEPROM	4.4.5.13
03h	Activate jog	uint32	rw	0	0..3	EEPROM	4.4.5.13
04h	Group setting	uint32	rw	0	0..7	EEPROM	4.4.5.13
0x2041	Expanded motor settings						4.4.5.14
00h	Largest sub index	uint8	ro	04h	4	EEPROM	4.4.5.14
01h	Collision on/off	uint32	rw	0	0..1	EEPROM	4.4.5.14
02h	Dynamic pre-stop	uint32	rw	0	0..1	EEPROM	4.4.5.14
03h	Input modes	uint32	rw	0	0..3	EEPROM	4.4.5.14
04h	Key modes	uint32	rw	0	0..3	EEPROM	4.4.5.14
0x2042	Speed switching points						4.4.5.15
00h	Largest sub index	uint8	ro	03h	3	-	4.4.5.15
01h	Creep speed	uint32	rw	0	0..9999	EEPROM	4.4.5.15
02h	Crawl speed	uint32	rw	150	0..9999	EEPROM	4.4.5.15
03h	Switch off value	int32	Rw	0	-9999..9999	EEPROM	4.4.5.15

Object Sub index	Name	Format	Access	Default	Measurement range	Save	Description Section/link
0x2043	End stops						4.4.5.16
00h	Largest sub index	uint8	ro	02h	2	-	4.4.5.16
01h	Min	int32	rw	-99999	-99999--999998	EEPROM	4.4.5.16
02h	Max	int32	rw	999999	-99998--999999	EEPROM	4.4.5.16
0x2044	Collision						4.4.5.17
00h	Largest sub index	uint8	ro	02h	2	-	4.4.5.17
01h	Min	int32	rw	-99999	-99999--999998	EEPROM	4.4.5.17
02h	Max	int32	rw	999999	-99998--999999	EEPROM	4.4.5.17
0x2045	Special set values D1, D2, D3						4.4.5.18
00h	Largest sub index	uint8	ro	03h	3	-	4.4.5.18
01h	Set value D1	int32	rw	-99999	-99999--999999	RAM	4.4.5.18
02h	Set value D2	int32	rw	999999	-99999--999999	RAM	4.4.5.18
03h	Set value D3	int32	rw	0	-99999--999999	RAM	4.4.5.18
0x2046	Step width for jog function	uint32	rw	1	0..9999	EEPROM	4.4.5.19
0x2047	Motor system times						4.4.5.20
00h	Largest sub index	uint8	ro	03h	3	-	4.4.5.20
01h	Loop time	uint32	rw	10	1..999	EEPROM	4.4.5.20
02h	Contouring error time	uint32	rw	30	1..999	EEPROM	4.4.5.20
03h	Clamping	uint32	rw	0	0..999	EEPROM	4.4.5.20
0x2048	Open / close clamping	uint8	rw	-	0..2	-	4.4.5.21
0x204D	Drive command						4.4.5.21
00h	Largest sub index	uint8	ro	02h	2		4.4.5.21
01h	Position indication start/stop	uint8	rw	-	0..8	-	4.4.5.21
02h	Oscillation start/stop	uint8	rw	-	0..1	-	4.4.5.21
0x204E	Read device mode data						4.4.5.23
00h	Largest sub index	uint8	ro	02h	2	-	4.4.5.23
01h	Read SPA status	uint16	ro	-	0..uint16	-	4.4.5.23
02h	Read SPA error	uint16	ro	-	0..uint16	-	4.4.5.23
0x2060	Sensor parameters						4.4.5.24
00h	Largest sub index	uint8	ro	03h	3	-	4.4.5.24
01h	SSI value resolution	uint32	rw	25	12..32	EEPROM	4.4.5.24
02h	Code	uint32	rw	1	0..1	EEPROM	4.4.5.24
03h	Sensor type	uint32	rw	0	0..4; 9	EEPROM	4.4.5.24
0x2100	Baud rate	uint8	rw	5	0..8	FLASH	4.4.5.25
0x2101	Node ID	uint8	rw	127	1..127	FLASH	4.4.5.26
0x2500	PDO broadcast						4.4.5.27
00h	Largest sub index	uint 8	rw	04h	4	-	4.4.5.27
01h	Broadcast control register	uint8	rw	0	0..255	-	4.4.5.27
02h	Broadcast object number	uint16	rw	0	0..uint16	-	4.4.5.27
03h	Broadcast sub index	uint8	rw	0	0..255	-	4.4.5.27
04h	Broadcast data	uint32	rw	0	0..uint32	-	4.4.5.27
0x2501	PDO set value						4.4.5.28
00h	Largest sub index	uint 8	ro	04h	4	-	4.4.5.28
01h	PDO set value – set value	uint32	rw	0	0.. uint32	-	4.4.5.28
02h	PDO set value - actual value correction	uint16	rw	0	0.. uint16	-	4.4.5.28
03h	PDO set value - control register	uint16	rw	0	0.. uint16	-	4.4.5.28
04h	PDO set value - actual value correction	uint16	rw	0	0.. uint16	-	4.4.5.28

4.4.3 Object overview – device profile specific objects

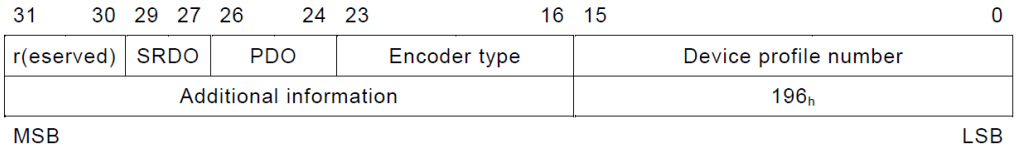
Object Sub index	Name	Format	Access	Default	Measurement range	Save	Description Section/link
0x6003	Preset	int32	rw	-	-99999..999999	EEPROM	4.4.6.1
0x6004	Position (actual value)	int32	ro	-	-99999..999999	-	4.4.6.2

4.4.4 Detailed object list - communication objects

4.4.4.1 Object 0x1000 – device type

Object 0x1000 transmits the device type back.

Structure of the data field:



Sub index	0x00
Description	Device type
Access	ro
Data type	UNSIGNED 32
Value range	0x00020196
Default	0x00020196
Storage	no
PDO mapping	no
Function	The device type consists of the following two parts: Encoder type: 0x02 = 2 (multiturn absolute rotary encoder) Device profile number: 0x0196 = 406 (encoder profile)

4.4.4.2 Object 0x1001 – Error Register

Object 0x1001 transmits the contents of the error register back

Sub index	0x00
Description	Error register
Access	ro
Data type	UNSIGNED 8
Value range	Bit 0: generic error Bit 1..3: reserved Bit 4: communication error Bit 5..6: reserved Bit 7: manufacturer specific error. Is 1, when in object 0x204E at least one error is set.
Default	0
Storage	-
PDO mapping	no
Function	Three error messages according to profile CiA 406 V4.0.2 Section 7.2.3 are implemented. If "manufacturer specific error" (bit 7) is active, then there is an applicative error of the SPA. These are the errors that are also shown on the display. The detailed error code can be read out via object 0x204E:02.

4.4.4.3 Object 0x1002 – Manufacturer Status Register

Object 1002h issues the status. The content of this register provides information about the functional status of the SPA. This status can also be read via object 0x204E:01 or via TxPDO1 and TxPDO2.

Note: NM170 does not contain object 0x204E. This status can only be read via object 0x1002. NM170 only transmits the information 'arrows active' (bit 12) back.

Sub index	0x00
Description	Manufacturer Status Register of the SPA Function
Access	ro
Data type	UNSIGNED 32
Value range	0 .. UNSIGNED 32
Default	-
Storage	-
PDO mapping	no
Valid for	NM170, NM172, NM174
Function	For a description of the individual status flags see Object 0x204E Sub index 1

4.4.4.4 Object 0x1003 – Pre-defined Error Field

The 8 most recent error situations are archived in object 1003h. The possible error situations are described in Section 6.1 CANopen Emergency Messages.

The entry under sub index 0 states the number of stored errors.
The most current error situation is always stored in sub index 1. The position of previous error messages is always moved one sub index further.

In case of a power interruption or a reset node, the entire error list is deleted.
Similarly, the error list can also be deleted by writing the value 0 at sub index 0.

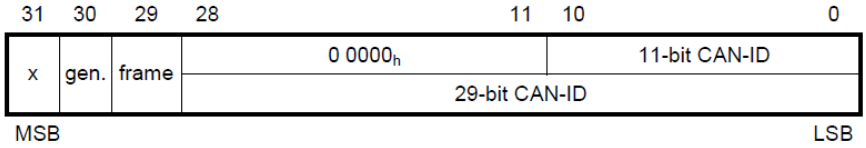
Sub index	0x00
Description	Number of stored emergency messages
Access	rw
Data type	UNSIGNED 8
Value range	Read: 0 .. 8; Write: only 0
Default	0
Storage	-
PDO mapping	no

Sub index	0x01 – 0x08
Description	Occurred emergency messages
Access	ro
Data type	UNSIGNED 32
Value range	0 .. UNSIGNED 32
Default	-
Storage	-
PDO mapping	no

4.4.4.5 Object 0x1005 – Sync COB ID

The Sync COB ID is defined via object 0x1005

Structure of the COB I



Sub index	0x00
Description	Sync COB ID
Access	rw
Data type	UNSIGNED 32
Value range	-
Default	0x00000080
Storage	Flash
PDO mapping	no
Function	Defines the COB ID of the synchronization object (SYNC) Bit 31 not defined (bit is ignored) Bit 30 1=sensor generates SYNC messages, 0=does not generate SYNC messages Bit 29 1=29 Bit SYNC COB ID (CAN 2.0B), 0=11 Bit SYNC CAN ID (CAN 2.0A) Bit 28..0 29 Bit SYNC CAN ID Bit 10..0 11 Bit SYNC CAN ID

4.4.4.6 Object 0x1008 – Device Name

The manufacturer specific device name is read out via object 0x1005

Sub index	0x00
Description	Manufacturer device name
Access	ro
Data type	VISIBLE_STRING (14 characters)
Value range	"NM170 Standard" "NM172 Standard" "NM174 Standard"
Default	depending on the used SPA
Storage	no
PDO mapping	no

4.4.4.7 Object 0x1009 – Hardware Version

The manufacturer specific hardware version is read out via object 0x1009

Sub index	0x00
Description	Manufacturer hardware version
Access	ro
Data type	VISIBLE_STRING (2 characters)
Value range	/A or higher
Default	depending on the used SPA
Storage	no
PDO mapping	no

4.4.4.8 Object 0x100A – Software Version

The manufacturer specific software version is read out via object 0x100A

Sub index	0x00
Description	Manufacturer software version
Access	ro
Data type	VISIBLE_STRING (4 characters)
Value range	1.05 or higher (complies with V1.0.5)
Default	depending on the used SPA
Storage	no
PDO mapping	no

4.4.4.9 Object 0x1010 – Store Parameters

Storage of the CAN interface parameters (except for baud rate and node ID) into the non-volatile storage (flash) is initiated via object 0x1010. These are parameter objects that offer "flash" as a storage option.
All applicative SPA parameters are stored into the internal EEPROM directly when writing the respective parameter. For these objects,"EEPROM (immediately)" is available as a storage option.

To prevent unintentional storing, the message "save" must be written in sub index 1 or 2.

Data format of the "save" signature

MSB			LSB
e	v	a	s
65_h	76_h	61_h	73_h

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	Save all parameters
Access	rw
Data type	UNSIGNED 32
Value range	read: 0x00000001 (the other bits are reserved) write: 0x65766173 (signature. corresponds to "save" in ASCII)
Default	0x00000001
Storage	FLASH
PDO mapping	no
Function	Stores all modifiable parameters of the standard objects (index range 0x1000 to 1FFF) in the flash storage. Note: The parameter range of sub index 1 is identical to index 2 (save communication parameters).

Sub index	0x02
Description	Save communication parameters
Access	rw
Data type	UNSIGNED 32
Value range	read: 0x00000001 (the other bits are reserved) write: 0x65766173 (signature. corresponds to "save" in ASCII)
Default	0x00000001
Storage	FLASH
PDO mapping	no
Function	Stores all modifiable parameters of the standard objects (index range 0x1000 to 1FFF) in the flash storage. Note: The parameter range of sub index 2 is identical to sub index 1 (save all parameters).

4.4.4.10 Object 0x1011 – Restore Parameters

Object 0x1011 resets the CAN interface parameters (including the baud rate and node ID) to the factory settings (default values). However, the default values are only active following a network failure or "reset node".
All applicative SPA parameters are reset to the factory settings via object 0x2015. For these objects,"EEPROM (immediately)" is available as a storage option.

To prevent unintentional restoring, the message "load" must be written in sub index 1 or 2.

Data format of the "load" signature

MSB			LSB
d	a	o	l
64_h	61_h	$6F_h$	$6C_h$

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

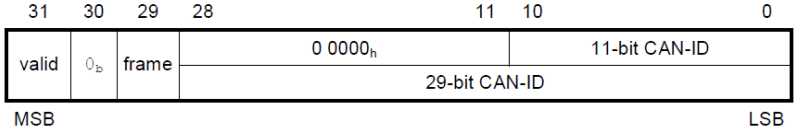
Sub index	0x01
Description	Restore all parameters
Access	rw
Data type	UNSIGNED 32
Value range	read: 0x00000001 (the other bits are reserved) write: 0x64616F6C (signature. corresponds to "load" in ASCII)
Default	0x00000001
Storage	FLASH
PDO mapping	no
Function	Sets the parameters of the standard objects (index range 0x1000 to 1FFF) and the communication parameters 0x2100 and 0x2101 to the default values. Note: The parameter ranges of sub index 1 and 2 are identical in these SPAs.

Sub index	0x02
Description	Restore all parameters
Access	rw
Data type	UNSIGNED 32
Value range	read: 0x00000001 (the other bits are reserved) write: 0x64616F6C (signature. corresponds to "load" in ASCII)
Default	0x00000001
Storage	FLASH
PDO mapping	no
Function	Sets the parameters of the standard objects (index range 0x1000 to 1FFF) and the communication parameters 0x2100 and 0x2101 to the default values. Note: The parameter ranges of sub index 1 and 2 are identical in these SPAs.

4.4.4.11 Object 0x1014 – Emergency COB ID

The COB ID for the emergency messages is defined via object 0x1014.

Structure of the emergency COB ID



Bit(s)	Value	Description
valid	0_b	EMCY exists / is valid
	1_b	EMCY does not exist / is not valid
30	0_b	reserved (always 0_b)
frame	0_b	11-bit CAN-ID valid (CAN base frame)
	1_b	29-bit CAN-ID valid (CAN extended frame)
29-bit CAN-ID	x	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	x	11-bit CAN-ID of the CAN base frame

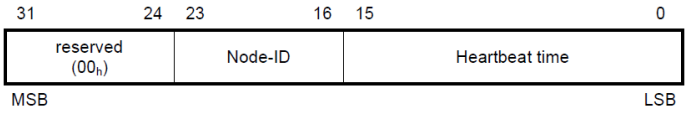
Sub index	0x00
Description	Emergency COB ID
Access	rw
Data type	UNSIGNED 32
Value range	0x00000081 .. 0x000000FF
Default	0x00000080 + Node ID
Storage	FLASH
PDO mapping	no
Function	In case of an according alarm, the SPA sends an emergency message with the COB ID set here.

4.4.4.12 Object 0x1016 – Consumer Heartbeat Time

The consumer heartbeat time object must display the expected heartbeat cycle times. Monitoring of the heartbeat producer commences after receipt of the first heartbeat.
If the heartbeat time is 0 or the node ID 0 or larger than 127, the corresponding object entry is not used. The heartbeat time must be given in multiples of 1ms.

Note: the time of the consumer heartbeat should be greater than the corresponding producer heartbeat. Prior to the receipt of the first heartbeat the status of the heartbeat is not known.

Structure of the RW register:



Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	1
Default	1
PDO mapping	no

Sub index	0x01
Description	Consumer heartbeat time
Access	rw
Data type	UNSIGNED 32
Value range	UNSIGNED 32
Default	0
Storage	FLASH
PDO mapping	no
Function	See heartbeat protocol 4.3.8

4.4.4.13 Object 0x1017 – Producer Heartbeat Time

Object 0x1017 defines the cycle time for the producer heartbeat.
The value 0 deactivates the producer heartbeat.

Sub index	0x00
Description	Producer heartbeat time
Access	rw
Data type	UNSIGNED 16
Value range	0..0xFFFF
Default	0
Storage	FLASH
PDO mapping	no
Function	The time in which the SPA carries out a heartbeat. The heartbeat time is given in multiples of 1ms. See heartbeat protocol in Section 4.3.8

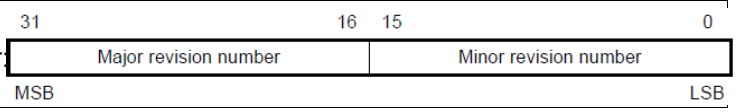
4.4.4.14 Object 0x1018 – Identity Object

With object 0x1018 the identity of the device is defined.

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	4
Default	4
PDO mapping	no

Sub index	0x01
Description	Vendor ID
Access	ro
Data type	UNSIGNED 32
Value range	0x000000EC (online at www.can-cia.de)
Default	0x000000EC
Storage	-
PDO mapping	no
Function	Vendor ID for Baumer Germany GmbH & Co. KG assigned by CiA

Sub index	0x02
Description	Product code
Access	ro
Data type	UNSIGNED 32
Value range	0x70 or 0x71 or 0x72
Default	depending on the used SPA
Storage	-
PDO mapping	no
Function	Describes the device type within the Multicon product range: 0x70 = NM170 0x71 = NM172 0x72 = NM174

Sub index	0x03
Description	Revision number
Access	ro
Data type	UNSIGNED 32
Value range	0..0xFFFFFFFF
Default	0x00010001 or higher
Storage	-
PDO mapping	no
Function	Structure of the revision number: 

Sub index	0x04
Description	Serial number
Access	ro
Data type	UNSIGNED 32
Value range	0..0xFFFFFFFF
Default	0x00010001 or higher
Storage	-
PDO mapping	no
Function	Unique, continuous serial number of the SPA. The SPA can be clearly identified via this number.

4.4.4.15 Object 0x1029 – Error Behavior Object

If a serious device error occurs during the NMT operational mode, i.e. an according flag is set in the error register object 0x1001m the CANopen device must switch to the mode shown here or remain in its previous mode, depending on the parameterization.

The following actions are available:

Value	Description
0	Switching to the pre-operational NMT mode if currently in the operational mode
1	No change of the NMT mode
2	Switch to the stopped NMT mode

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	Communication error
Access	rw
Data type	UNSIGNED 8
Value range	0..2
Default	0
Storage	FLASH
PDO mapping	no
Function	Defines the behavior of the SPA only when a "communication error" occurs due to a heartbeat error. With the bus warning and bus passive bus errors no device status change is carried out.

Sub index	0x02
Description	Manufacturer specific error
Access	rw
Data type	UNSIGNED 8
Value range	0..2
Default	1
Storage	FLASH
PDO mapping	no
Function	Defines the behavior of the SPA when a "manufacturer specific error" occurs due to a heart-beat error.

4.4.4.16 Object 0x1400 – Receive PDO1 - Parameter

Object 0x1400 defines the setting parameters for the receive process data object 1 (RxPDO1)

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	COB ID
Access	rw
Data type	UNSIGNED 32
Value range	0x40000200 + Node ID
Default	0x40000201
Storage	FLASH
PDO mapping	no
Function	RxPDO1 is used as broadcast for the transmission of SDO commands. Therefore, for all SPAs, in the factory settings the COB ID is defined at 0x201. This way, all devices receive the broadcast information and carry it out. This COB ID remains unchanged even with a change of the node ID.

Sub index	0x02
Description	Transmission type
Access	rw
Data type	UNSIGNED 8
Value range	0xFF
Default	0xFF
Storage	FLASH
PDO mapping	no
Function	PDO is event-driven. This means that the PDO can be received at any time. The CANopen device immediately carries out the broadcast operation.

4.4.4.17 Object 0x1401 – Receive PDO2 - Parameter

Object 0x1401 defines the setting parameters for the receive process data object 2 (RxPDO2)

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	COB ID
Access	rw
Data type	UNSIGNED 32
Value range	0x40000300 + Node ID
Default	0x4000037F
Storage	FLASH
PDO mapping	no
Function	In case of a change of the node ID, the COB ID is automatically adjusted to the new node ID.

Sub index	0x03
Description	Transmission type
Access	rw
Data type	UNSIGNED 8
Value range	0xFF
Default	0xFF
Storage	FLASH
PDO mapping	no
Function	PDO is event-driven. This means that the PDO can be received at any time. The CANopen device immediately implements the establishment of a set value and the potential starting of the motor.

4.4.4.18 Object 0x1600 – Receive PDO1 - Mapping

Object 0x1600 defines the mapping for the receive process data object 1 (RxPDO1)
The PDO serves the broadcast transmission of select SDO commands.

Structure of RxPDO1:

1	2	3	4	5	6	7	8
Control	Object		Sub index	Data			

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	4
Default	4
PDO mapping	no

Sub index	0x01
Description	Broadcast control
Access	ro
Data type	UNSIGNED 32
Value range	0x25000108
Default	0x25000108
Storage	-
PDO mapping	no
Function	The 1st byte (control) of the RxPDO1 is allocated to object 0x2500:01.

Sub index	0x02
Description	Broadcast object
Access	ro
Data type	UNSIGNED 32
Value range	0x25000210
Default	0x25000210
Storage	-
PDO mapping	no
Function	Byte 2 and 3 (object) of the RxPDO1 are allocated to object 0x2500:02.

Sub index	0x03
Description	Broadcast sub index
Access	ro
Data type	UNSIGNED 32
Value range	0x25000308
Default	0x25000308
Storage	-
PDO mapping	no
Function	Byte 4 (sub index) of the RxPDO1 is allocated to object 0x2500:03.

Sub index	0x04
Description	Broadcast data
Access	ro
Data type	UNSIGNED 32
Value range	0x25000420
Default	0x25000420
Storage	-
PDO mapping	no
Function	Bytes 4-8 (data) of the RxPDO1 are allocated to object 0x2500:04.

4.4.4.19 Object 0x1601 – Receive PDO2 - Mapping

Object 0x1601 defines the mapping for the receive process data object 2 (RxPDO2)
The PDO serves the position indication of the SPA. The following parameters are transmitted.

Structure of RxPDO2:

1	2	3	4	5	6	7	8
Set value				Actual value correction		Control	

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	Position indicator– set value
Access	ro
Data type	UNSIGNED 32
Value range	0x25010120
Default	0x25010120
Storage	-
PDO mapping	no
Function	Bytes 1-4 (set value) of the RxPDO2 are allocated to object 0x2501:01.

Sub index	0x02
Description	Position indicator - actual value correction
Access	ro
Data type	UNSIGNED 32
Value range	0x25010210
Default	0x25010210
Storage	-
PDO mapping	no
Function	Bytes 5-6 (set value) of the RxPDO2 are allocated to object 0x2501:02.

Sub index	0x03
Description	Position indicator – control
Access	ro
Data type	UNSIGNED 32
Value range	0x25010310
Default	0x25010310
Storage	-
PDO mapping	no
Function	Bytes 7-8 (control) of the RxPDO2 are allocated to object 0x2501:03.

4.4.4.20 Object 0x1800 – Transmit PDO1 - Parameter

Object 0x1800 defines the mapping for the transmit process data object 1 (TxPDO1)

Sub index	0x00
Description	Highest supported sub index
Access	ro
Data type	UNSIGNED 8
Value range	5
Default	5
PDO mapping	no

Sub index	0x01
Description	COB ID
Access	rw
Data type	UNSIGNED 32
Value range	0xC0000180 + node ID
Default	0xC00001FF TxPDO1 is deactivated after resetting to factory settings (highest bit set).
Storage	FLASH
PDO mapping	no
Function	In case of a change of the node ID, the COB ID is automatically adjusted to the new node ID.

Sub index	0x02
Description	Transmission type
Access	rw
Data type	UNSIGNED 8
Value range	0x01 .. 0xF0, 0xFE, 0xFF
Default	0xFF
Storage	FLASH
PDO mapping	no
Function	The following functions of the PDO can be set: 0x00: PDO has a non-cyclical character. 0x01..n..0xF0: PDO has a synchronous character. Each nth SYNC telegram receives the PDO. 0xFE: PDO has an asynchronous character. PDOs are sent cyclically depending on the event timer. 0xFF: PDO has an asynchronous character. PDO transmits with a change of data.

Sub index	0x03
Description	Inhibit time
Access	rw
Data type	UNSIGNED 16
Value range	0..UNSIGNED16
Default	80
Storage	FLASH
PDO mapping	no
Function	The inhibit time is the minimum interval for the PDO transmission, when the transmission type is set to 0xFE and 0xFF. The value is defined as a multiple of 1ms. Value 0 deactivates the inhibit time.

Sub index	0x05
Description	Event time (cycle time in ms)
Access	rw
Data type	UNSIGNED 16
Value range	0.. UNSIGNED 16
Default	1000
Storage	FLASH
PDO mapping	no
Function	Event timer for process data object 0: Cyclical transmission switched off 1..n..65535: Repeat time of the cyclical transmission is n ms.

4.4.4.21 Object 0x1801 – Transmit PDO2 - Parameter

Object 0x1801 defines the mapping for the transmit process data object 2 (TxPDO2)

Sub index	0x00
Description	Highest supported sub index
Access	ro
Data type	UNSIGNED 8
Value range	5
Default	5
PDO mapping	no

Sub index	0x01
Description	COB ID
Access	rw
Data type	UNSIGNED 32
Value range	0x40000280 + Node ID
Default	0x400002FF
Storage	FLASH
PDO mapping	no
Function	In case of a change of the node ID, the COB ID is automatically adjusted to the new node ID.

Sub index	0x02
Description	Transmission type
Access	rw
Data type	UNSIGNED 8
Value range	0x01 .. 0xF0, 0xFE, 0xFF
Default	0xFF
Storage	FLASH
PDO mapping	no
Function	The following functions of the PDO can be set: 0x00: PDO has a non-cyclical character. 0x01..n..0xF0: PDO has a synchronous character. Each nth SYNC telegram receives the PDO. 0xFE: PDO has an asynchronous character. PDOs are sent cyclically depending on the event timer. 0xFF: PDO has an asynchronous character. PDO transmits with a change of data. **Note**: even if an event time is set, a PDO is <u>not</u> transmitted based on this event time.

Sub index	0x03
Description	Inhibit time
Access	rw
Data type	UNSIGNED 16
Value range	0..UNSIGNED16
Default	80
Storage	FLASH
PDO mapping	no
Function	The inhibit time is the minimum interval for the PDO transmission, when the transmission type is set to 0xFE and 0xFF. The value is defined as a multiple of 1ms. Value 0 deactivates the inhibit time.

Sub index	0x05
Description	Event time (cycle time in ms)
Access	rw
Data type	UNSIGNED 16
Value range	0.. UNSIGNED 16
Default	1
Storage	FLASH
PDO mapping	no
Function	Event timer for process data object 0: cyclical transmission switched off 1..n..65535: Repeat time of the cyclical transmission is n ms.

4.4.4.22 Object 0x1A00 – Transmit PDO1 - mapping

Object 0x1A00 defines the mapping for the transmit process data object 1 (TxPDO1)

Structure TxPDO1:

1	2	3	4	5	6	7	8
Position value				Status		Errors	

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	Mapping for object 0x6004 (position)
Access	ro
Data type	UNSIGNED 32
Value range	0x60040020
Default	0x60040020
Storage	-
PDO mapping	no
Function	The position value is allocated to bytes 1-4 of the PDO

Sub index	0x02
Description	Mapping for object 0x204E:01 (SPA status)
Access	ro
Data type	UNSIGNED 32
Value range	0x204E0110
Default	0x204E0110
Storage	-
PDO mapping	no
Function	The status flags are allocated to bytes 5-6 of the PDO.

Sub index	0x03
Description	Mapping for object 0x204E:02 (SPA errors)
Access	ro
Data type	UNSIGNED 32
Value range	0x204E0210
Default	0x204E0210
Storage	-
PDO mapping	no
Function	The error flags are allocated to bytes 7-8 of the PDO.

4.4.4.23 Object 0x1A01 – Transmit PDO1 - mapping

Object 0x1A01 defines the mapping for the transmit process data object 2 (TxPDO2)

Structure TxPDO2:

1	2	3	4	5	6	7	8
Position value				Actual value correction		Status	

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	Mapping for object 0x6004 (position)
Access	ro
Data type	UNSIGNED 32
Value range	0x60040020
Default	0x60040020
Storage	-
PDO mapping	no
Function	The position value is allocated to bytes 1-4 of the PDO

Sub index	0x02
Description	Mapping for object 0x2501:04 (actual value correction)
Access	ro
Data type	UNSIGNED 32
Value range	0x25010410
Default	0x25010410
Storage	-
PDO mapping	no
Function	The status flags are allocated to bytes 5-6 of the PDO. If object 0x2000:06 = actual value correction off, always returns 0 even hen object 0x2010 (offset for actual value correction) is unequal 0.

Sub index	0x03
Description	Mapping for object 0x204E:01 (SPA status)
Access	ro
Data type	UNSIGNED 32
Value range	0x204E0110
Default	0x204E0110
Storage	-
PDO mapping	no
Function	The status flags are allocated to bytes 7-8 of the PDO.

4.4.4.24 Object 0x1F80 – NMT startup

This object (described in DS-302 part 2) defines whether the device is automatically in the OPERATIONAL mode following a reset. For this, 0x08 must be written in the object. Parameter 0x00 sets the automatic OPERATIONAL mode to inactive.

Sub index	0x00
Description	NMT startup
Access	rw
Data type	UNSIGNED 32
Value range	See function
Default	0
Storage	FLASH
PDO mapping	no
Function	The following settings are possible 0 = NMT Slave must be started by the NMT Master 8 = NMT Slave automatically switches to the operational NMT mode(self-starting) When writing with an invalid data value the SPA replies with the abort message 0x06090030. The parameter set here is only implemented following a restart or reset node. The parameter must be stored via the object 0x1010 before a power interruption or reset node.

4.4.5 Detailed object list – Manufacturer-specific objects

4.4.5.1 Object 0x2000 – General basic settings

The object contains general basic settings that are equally present in all SPAs.
The following parameters are writable and readable:

- Position indication direction
- Counting direction
- Arrows
- Rounded actual value
- Rotate display
- Actual value correction on/off
- Hide set value
- External inputs

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	8
Default	8
PDO mapping	no

Sub index	0x01
Description	Position indication direction.
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM
PDO mapping	no
Function	The parameter determines the side from which the target position is approached. To compensate a potential play during exact position indications, e.g of gearwheels, joints, spindles it is necessary to carry out a play compensation. This play compensation is accomplished by always approaching the target position from the same direction (position indication direction). This same position indication direction is ensured via a so-called loop drive. The target position is not directly approached. It is first traversed by a defined distance and then the position indication carried out from the other direction. This defined distance can be set via object 0x2002:00. Whether a loop drive must take place is indicated by the directional arrows on the display. If the directional arrow blinks, a spindle play compensation must be carried out. When the reversal point is reached, the arrow direction changes and the arrow no longer blinks. If the target position can be reached directly, a non-blinking arrow appears immediately. The arrows only go out in the target, i.e. in the programmed tolerance window when a require play compensation has been carried out. See also object 0x2002. The following settings are possible: 0: Up: Position is directly reached when the target position > actual position; otherwise play compensation is carried out 1: Down: Position is directly reached if target position < actual position; otherwise play compensation is carried out

Sub index	0x02
Description	Counting direction.
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	The actual value counting method "ascending" or "descending" is allocated to the spindle direction of rotation. The following settings are possible: 0: Up: If the spindle direction of rotation is to the right, the actual value changes in an ascending manner 1: Down: If the spindle direction of rotation is to the right, the actual value changes in a descending manner

Sub index	0x03
Description	Arrows
Access	rw
Data type	UNSIGNED 32
Value range	0..3 (NM170) 0..5 (NM172, NM174)
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	With this parameter, arrows can be inserted in the display. The arrows are setting aids that signal to the operator in which direction the new setting should be carried out (right or left, larger or smaller). A blinking arrow signals that the target position cannot be reached directly. Play compensation must be carried out first. The following settings are possible: 0: Up With actual value < set point, right arrow inserted; actual value > target value, left arrow inserted. 1: Down Same as "Up", but with a reversed arrow direction 2: Uni When actual value ≠ set value, both arrows are always inserted. No loop drive 3: Off arrows are always masked. No loop drive 4: Uni loop [*1)] Same as 02h Uni but with loop drive 5: Off Loop [*1)] Same as 03h Off but with loop drive [1)] only for NM172, NM174

Sub index	0x04
Description	Rounded actual value
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	If the actual value is within the tolerance window but is not identical to the set value, the value is rounded to the set value 3 seconds after the spindle comes to a halt. This rounding only takes place on the display. When reading out the actual value (object 0x6004) you receive the actual non-rounded value. If a key is pressed, the true actual value is displayed again. If the position is still within the tolerance window, the value is displayed rounded again after 3 seconds. Note: In the operating mode "´Hide set value = Ever“ the set value is deactivated. The SPA serves as a pure actual value display. The rounding function is ignored. The following settings are possible: 0: Off Actual value is not rounded 1: On Actual value is rounded

Sub index	0x05
Description	Rotate display
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	With this parameter, the presentation in the display can be rotated by 180°. With standard mounting the display is on top and the keys are on the bottom. The following settings are possible: 0: Off Display is readable with standard mounting, i.e . the display is on top, the keys are on the bottom. 1: On Display is readable with inverted mounting.

Sub index	0x06
Description	Actual value correction off/on
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	1
Storage	EEPROM (immediately)
PDO mapping	no
Function	Via object 0x2010 an offset value can be programmed that is added to the actual value. Whether or not the offset value is included in the calculation can be selected via this parameter. 0: Off Actual value correction is deactivated. If an offset value was previously transmitted, it is not added to the current actual value and set value. 1: On Actual value correction released. The offset value is immediately added to the current actual value and added when a new set point is set.

Sub index	0x07
Description	Hide set value
Access	rw
Data type	UNSIGNED 32
Value range	0..2
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	This parameter regulates the presentation of the set point in the top display line. The set value is closely associated with the arrows. If no set value is displayed, the arrows are also masked. The following settings are possible: 0: On Set value is displayed if set value ≠ actual value 1: Off Set value is always displayed even when set value = actual value. Additionally, the set value ≠ actual value 2: Ever The set value and the arrows are always masked

Sub index	0x08
Description	External inputs
Access	rw
Data type	UNSIGNED 32
Value range	0..3
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	This consists of the basic functions of the external inputs. Additional functions are defined in object 0x2041:03. For these functions to be active here, object 0x2041:03 = 0 must be set. The following settings are possible: 0: Key Function identical to the keys on the SPA 1: Slow When actuated, motor always drives at slow speed 2: Middle When actuated, motor always drives at medium speed (only when 3 speeds are available, otherwise fast speed) 3: Fast When actuated, the motor always drives at fast speed The external cables are integrated as follows in the motor cable: **Pin assignment** F External input 1 G External input 2

4.4.5.2 Object 0x2001 – Length unit (mm/inch)

The length unit of the SPA is specified via object 0x2001.

Sub index	0x00
Description	Length unit
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no Setting the length unit is also possible via RxPDO1 (broadcast).
Function	Internally, the SPA uses only millimeter values. The set value and all other length-related parameters must be generally converted to millimeters. The length unit parameter only affects the display. The conversion from mm to inch takes place in the spindle position display when the position values are displayed. Similarly, when reading length values, including object 0x6004, millimeter values are transmitted back. The following settings are possible: 0: mm 1: Inch

4.4.5.3 Object 0x2002 – Loop for Spindle Play Compensation

With object 0x2002 the length of the loop travel of a spindle play compensation is specified.

Sub index	0x00
Description	Loop value for spindle play compensation in mm
Access	rw
Data type	UNSIGNED 32
Value range	0..9999 (unit 1/100 mm)
Default	100
Storage	EEPROM (immediately)
PDO mapping	no
Function	The stated value acts in relation to the specified set value. In case of a loop drive, an absolute loop position is calculated in the SPA. This is dependent on the parameter position indication direction object 0x2000:01.

4.4.5.4 Object 0x2003 – Tolerance window

With object 0x2003 the dimension of the tolerance window for the current set value is determined.

Sub index	0x00
Description	Value for tolerance window (Tol) in mm
Access	rw
Data type	UNSIGNED 32
Value range	0..9999 (unit 1/100 mm)
Default	5
Storage	EEPROM (immediately)
PDO mapping	no
Function	The absolute tolerance window is derived from the following formula: Tolerance window$_{Abs}$ = (Set value– Tol) .. (Set value + Tol) If, following successful position indication, the actual position is within the tolerance window, then the status bit "InPosition" is set to 1 (see object 0x204E:01).

4.4.5.5 Object 0x2004 – Scaling of the spindle gradient

With object 0x2004 the scaling required to achieve a specific spindle gradient per revolution is specified.

Sub index	0x00
Description	Scaling value for spindle gradient.
Access	rw
Data type	UNSIGNED 32
Value range	00000000..99999999 (1 digit before and 7 digits after the decimal point)
Default	1736111 (equals 0.1736111)
Storage	EEPROM (immediately)
PDO mapping	no
Function	The scaling value is calculated as follows for the different devices: $\text{Skalierung} = \frac{\text{Spindelsteigung}}{\text{Auflösung pro Umdrehung}} * 10.000.000$ Transmission of the scaling value: The calculated scaling value is multiplied by 10,000,000 and transmitted as an integer (UNSIGNED 32).

Example calculation for NM170, NM172:

Resolution per revolution: 2304 steps/revolution
Spindle gradient: 4.00 mm

$$\text{Skalierung} = \frac{\text{Spindelsteigung}}{\text{Auflösung pro Umdrehung}} * 10000000 = \frac{400}{2304} * 10000000 = 0{,}1736111 * 10000000 = \mathbf{1736111}$$

Example calculation for NM174:

Resolution of the rotary encoder: 4096 steps/revolution ≙ 40.96 mm with scaling of 1.0
Spindle gradient: 4.00 mm

$$\text{Skalierung} = \frac{\text{Spindelsteigung}}{\text{Auflösung pro Umdrehung}} * 10000000 = \frac{400}{4096} * 10000000 = 0{,}0976563 * 10000000 = \mathbf{976563}$$

If an external sensor, such as e.g. a length measuring device, transmits the position values already in a resolution of 1/100, the scaling value is set to 1.0. Scaling = 1.0 * 10000000 = **10000000**.

4.4.5.6 Object 0x2005 – Special Parameters

With object 0x2005 the following special parameters are specified:

- Filter parameters
- LCD digit on/off

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	Filter width (maximum permissible position jump in steps)
Access	rw
Data type	UNSIGNED 32
Value range	0..2303
Default	100
Storage	EEPROM (immediately)
PDO mapping	no
Function	The filter width is the permissible difference leap in steps, at which filtering begins. If there are position leaps > the difference, then this position value is filtered out. The filter width stated here refers to the non-scaled steps of the sensor and not the scaled value shown on the display. NM170, NM172: To 2304 steps/revolution NM174: To the resolution of the used external sensor

Sub index	0x02
Description	Filter number
Access	rw
Data type	UNSIGNED 32
Value range	0..99
Default	5 (NM170, NM172) 2 (NM174)
Storage	EEPROM (immediately)
PDO mapping	no
Function	Number of position leaps to be filtered. If the leap difference remains constant across the parameterized number of filters, filtering ceases. The most recently read sensor value is accepted again as a valid position value. If the number of filters = 5, then the 6th position leap is acquired as the valid value. The internal counter is reset to 0. If a valid value is read with position leaps prior to the application of the filter counter, this value is used and the internal counter reset to 0. **Note:** The EMC interference resistance is maintained if the parameter is set to >= 2.

Sub index	0x03
Description	Mask LCD digit right
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	With this parameter, the respective right LCD digit in both the top and bottom display line can be switched on or off. This way, the display is in tenths of millimeters. The following parameter settings are possible: 0: Digit on 1: Digit off

4.4.5.7 Object 0x2006 – Device data MSP controller

With object 0x2006 the following device data of the internal µ-controller MSP can be read out:

- Version of the MSP controller firmware
- Device type (reserved for later special versions)

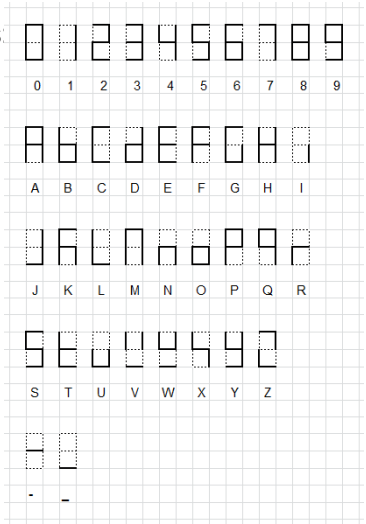
Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	Version of the MSP firmware
Access	ro
Data type	UNSIGNED 32
Value range	105 or higher
Default	no default value (version hard coded)
Storage	EEPROM (immediately)
PDO mapping	no
Function	Corresponds to the version number that is shown on the display during the boot process. Presentation on the display is with a decimal point and two digits after the decimal point (e.g. 1.05). However, the presentation when reading over this object is without decimal point (e.g. 105).

Sub index	0x02
Description	Device type MSP firmware
Access	ro
Data type	UNSIGNED 32
Value range	fixed value
Default	no default value (device type is hard coded)
Storage	EEPROM (immediately)
PDO mapping	no
Function	The different devices such as NM170, NM172, NM174 each contain their own firmware. This object provides the concerned device type in an encoded form: 0x8081 = NM170 standard version 0x8281 = NM172 standard version 0x8481 = NM174 standard version

4.4.5.8 Object 0x2007 – Display Message

With object 0x2007 alphanumeric characters are written in the two display lines of the SPA.

Sub index	0x00
Description	Display message.
Access	rw
Data type	VISIBLE_STRING (UNSIGNED_8[12])
Value range	String with 12 characters
Default	"1234567890AB"
Storage	no (only in the internal RAM)
PDO mapping	no
Function	The object allows writing the characters defined below in the two display lines. 12 characters must always be transmitted. If the data length is not equal to 12, the characters are ignored and acknowledged with the abort code: 0x06090030 "value outside the limits". Permissible are alphanumeric characters including the minus sign and underscore. Transmitted characters that cannot be shown on the 7-segment display are replaced by blanks. Alpha characters must be transmitted as capital letters. Presentable characters  Switch to the normal mode: By writing a random object, power off/on, or a reset node.

4.4.5.9 Object 0x2008 – Display device address (Node ID)

With object 0x2008 the node ID is shown on the display of the SPA.

Sub index	0x00
Description	Show the node ID on the display
Access	wo
Data type	UNSIGNED 32
Value range	0..1
Default	-
Storage	no
PDO mapping	no Display of the node ID is also possible via RxPDO1 (broadcast).
Function	The object switches the display and shows the node ID in the bottom display line. The top display line is deactivated (6 blank spaces). The following settings are possible: 0: Standard display 1: Node ID is displayed

4.4.5.10 Object 0x2010 – Actual Value Correction

With object 0x2010 an offset for moving the actual value can be set.

Sub index	0x00
Description	Offset for actual value correction
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Function	The offset value is added to the actual value. This is an absolute (not cumulative) offset value. The actual value correction function must be enabled in object 0x2000:06. It then applies: $\text{IstwertAnz} = \text{IstwertAbs} + \text{PresetOffset} + \text{Istwertkorrektur}$ Legend: Actual value display: Actual value that is shown on the display and read out via the interface Actual value absolute: Internal absolute actual value (after scaling) PresetOffset: Offset value created via object 0x6003 The entered value is transmitted back in TxPDO2. The TxPDO2 data format for the actual value correction is limited to16 bit. If values > 32767 or < -32768 are entered here, in TxPDO the maximum possible value 32767 or the minimum possible value -32768 is transmitted back. Therefore, these two values indicate a positive or negative overflow. In this case it is mandatory to use object 2010 to receive the precise value for the actual value correction. See also sections 4.3.4.3 RxPDO2 and 4.3.4.5 TxPDO2

4.4.5.11 Object 0x2015 – SPA Resets

With object 0x2015 different resets of SPA functions can be carried out.

Sub index	0x00
Description	SPA reset function
Access	wo
Data type	UNSIGNED 32
Value range	1..6
Default	-
Storage	EEPROM (immediately)
PDO mapping	No Resets are also possible via RxPDO1 (broadcast).
Function	The reset functions that are executable here relate to SPA functions that are implemented in the µ-controller MSP. The interface parameters are set to default with the object 0x1011. The following reset functions are possible: 1: Setting the SPA parameters to default values 2: Setting the multi-turn counter to 4096. Equals the value 0 3: All parameters and offsets are set to default values 4: Reset actual value offset 5: Controller reset MSP 6: Reset preset offset Immediately after the reset, the changed parameters and offsets are automatically stored in the EEPROM. Storage via object 0x1010 is not required.

4.4.5.12 Object 0x2020 – Set value (target position)

With object 0x2020 set values for different position indication functions can be specified. After a set value is written, in case of release, the arrows for the position indication are accordingly placed. Position indication to the specified target value can then be carried out manually or automatically. Some of the following set value commands immediately initiate the motor-actuated position indication. With the other commands the start is initiated with the object 0x204D:01.

The following functions are possible:

- Standard set value (S)
- Standard set value with immediate motor release (SF)
- Set value dimension (SD)
- Set value dimension with immediate motor release (SDF)
- Set value for direct position indication without loop drive and immediate motor release (SEF)
- Set value for cutting in (SL) function
- Set value for the cutting in function with immediate motor release (SLF)
- Speed for cutting in function

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	8
Default	8
PDO mapping	no

Sub index	0x01
Description	Standard set value (S)
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	none - last programmed value (at delivery 1000)
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM170, NM172, NM174
Function	The transmitted set value is stored in the RAM and the non-volatile storage (EEPROM). Following a power interruption, this standard set value is always active. Motor-actuated position indication is enabled via object 0x204D:01.

Sub index	0x02
Description	Standard set value with immediate motor release (SF)
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	none - last programmed value (at delivery 1000)
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	Same as sub index 0x01, however, the motor release is activated after writing the set value. Motor-actuated positioning takes place immediately.

Sub index	0x03
Description	Set value dimension (SD)
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	0
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	This set point command is used for a single position indication. The transmitted set value is only stored in the RAM. After a power interruption the set value in sub index 0x03 is the same as the set value in sub index 0x01. Motor-actuated position indication is enabled via object 0x204D:01.

Sub index	0x04
Description	Set value dimension with immediate motor release (SDF)
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	0
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	Same as sub index 0x03, however, the motor release is activated after writing the set value. Motor-actuated positioning takes place immediately.

Sub index	0x05
Description	Set value for direct position indication without loop drive and immediate motor release (SEF)
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	0
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	Same as sub index 04h, but position indication is carried out without a potential loop drive. Motor-actuated positioning takes place immediately.

Sub index	0x06
Description	Set value for cutting in (SL) function
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	0
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	For the cutting in function the position indication is implemented with the speed selected in 0x2020:08. Switching to crawl speed prior to the target drive remains in place. After writing in this subindext, manual positioning operations via button are performed at this speed The speed is maintained until: - setting a new set point. - jog via PDO is being performed. - clamping via PDO is being opened resp. closed. - in the event of voltage interrupt. The transmitted set value is only stored in the RAM. After a power interruption the set value in sub index 0x06 is the same as the set value in sub index 0x01. Motor-actuated positioning is enabled via object 0x204D:01.

Sub index	0x07
Description	Set value for the cutting in function with immediate motor release (SLF)
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	0
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	Same as sub index 0x06, however, the motor release is activated after writing the set value. Motor-actuated positioning takes place immediately.

Sub index	0x08
Description	Speed for cutting in function
Access	rw
Data type	UNSIGNED 32
Value range	0..2
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	Setting the positioning speed for the cutting in function. For the functionality see sub index 0x06 and 0x07. The following speeds can be set: 0: slow speed 1: medium speed 2: fast speed

4.4.5.13 Object 0x2040 – Motor - Basic Settings

With object 0x2040 the basic settings of the motor are specified:
The following parameters are writable and readable:

- Key allocation
- Motor rotation direction
- Activate jog (microstep)
- Group setting

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	4
Default	4
PDO mapping	no

Sub index	0x01
Description	Key allocation
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	Allocation of the device front keys regarding the motor control right/left. The following settings are possible: 0: Up Left key pressed = shaft turns left (ccw), descending count Right key pressed = shaft turned right (cw), ascending count 1: Down Left key pressed = shaft turns right (cw), ascending count Right key pressed = shaft turns to the left (ccw), descending counting The rotational directions stated here apply when the object *0x2000:02 counting direction* and *0x2040:02 motor rotational direction* are set to default values.

Sub index	0x02
Description	Motor rotation direction
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The output signals of the motor control for rotating left (pin B) and rotating right (pin C) are switched. The following settings are possible: 0: Up Standard direction of rotation 1: Down Inverted direction of rotation

Sub index	0x03
Description	Activate jog (microstep)
Access	rw
Data type	UNSIGNED 32
Value range	0..3
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The jog function allows the driving of the motor by a defined number of steps by briefly pressing a key. With a key pressure > 400 ms the motor runs continuously as long as the key remains pressed. Shorter key activation is implemented as a jog. The step widths are set via the object 0x2046. Note: the concerned step is always carried out completely and cannot be restarted or stopped by pressing a key again while the motor is carrying out the position indication. Stopping via the interface (object 0x204D:01) is possible. While the motor is rotating, the moving bit is set to 1 (see object 0x204E:01). The following settings are possible: 0: Up The jog function is enabled for the ascending counting direction 1: Down The jog function is enabled for the descending counting direction 2: Ever The jog function is enabled in both directions 3: Off The jog function is deactivated

Sub index	0x04
Description	Group setting
Access	rw
Data type	UNSIGNED 32
Value range	0..7
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The individual SPAs can be divided into 8 different groups. Position indication can be carried out for each group separately at different times via a broadcast start command (see RxPDO1). The grouping is used, for example, to prevent collisions with other shafts. The following settings are possible: 0: Group 1 1: Group 2 2: Group 3 : : 7: Group 8

4.4.5.14 Object 0x2041 – Motor - Enhanced Settings

With object 0x2041 special enhanced motor settings are specified:
The following parameters are writable and readable:

- Collision on/off
- Dynamic pre-stop on/off
- Input modes (Oscillation, Cycle, ExtPosi, off)
- Key modes

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	4
Default	4
PDO mapping	no

Sub index	0x01
Description	Collision on/off
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The parameter switches the collision function on or off. The collision is similar to the end stop MIN or end stop MAX. The following settings are possible: 0: On Collision is active 1: Off Collision is deactivated

Sub index	0x02
Description	Dynamic pre-stop
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The parameter activates or deactivates the dynamic pre-stop. If the parameter is set to On, after each position indication the dynamic pre-stop is calculated, which then directly affects the switch off point object 0x2042:03. During this, the ½-difference between the target point (set value) and the actual target position to the most recent switch-off position is added or subtracted. The following settings are possible: 0: Off Dynamic pre-stop deactivated 1: On Dynamic pre-stop activated

Sub index	0x03
Description	Input modes
Access	rw
Data type	UNSIGNED 32
Value range	0..3
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The parameter specifies the functions of the external inputs "F" and "G" (in the motor cable). The following settings are possible: 0: Standard Standard functions or the function "Oscillation" is active 1: Cycle Function "Cycle" is active 2: ExtPosi Function "ExternPosi" is active 3: Off External inputs are not active

Sub index	0x04
Description	Key modes
Access	rw
Data type	UNSIGNED 32
Value range	0..3
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The parameter specifies the functions of the device keys. The functions here are comparable with the setting functions external inputs in object 0x2000. However, they refer to the keys instead of the external inputs. The following settings are possible: 0: Standard Standard function of the keys. Jog function as well as slow and fast speed (switchable by briefly releasing the key) 1: Slow When actuated the motor rotates generally at slow speed 2: Middle When actuated the motor rotates generally at medium speed 3: Fast When actuated the motor rotates generally at fast speed

4.4.5.15 Object 0x2042 – Motor - Speed switching points

Prior to reaching the set value, the motor speed can be slowed down via 2 speeds (creep and crawl speed), to achieve a precise position indication at the set value.
It is also possible to set the switch-off point of the motor to a relative value before or after the set value.

The individual switching points are given relative to the set value. The actual absolute switching point can therefore lie to the left or right of the set point, depending on the setting of the positioning direction (object 0x2000:01).
The switch-off point can also be parameterized with a negative value. With negative values the actual absolute switch-off point must be inside the tolerance window.

The speed switching points affect the end stops (object 0x2043) and the collisions (object 0x2044).

The following parameters are writable and readable:

- Creep speed
- Crawl speed
- Switch off value

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	Creep speed
Access	rw
Data type	UNSIGNED 32
Value range	0..9999 (unit: hundredths of millimeters)
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	At this position, the SPA switches from the fast speed to the creep speed. With the setting creep speed <= crawl speed the creep speed is deactivated.

Sub index	0x02
Description	Crawl speed
Access	rw
Data type	UNSIGNED 32
Value range	0..9999 (unit: hundredths of millimeters)
Default	150
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	At this position, the SPA switches from the creep speed to the crawl speed.

Sub index	0x03
Description	Switch off value
Access	rw
Data type	SIGNED 32
Value range	-9999..9999 (unit: hundredths of millimeters)
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	At this position, the SPA stops the motor.

4.4.5.16 Object 0x2043 – Motor - End Stops

A minimum value (MIN) and maximum value (MAX) are available as end stops. The end stops limit the driving range of the motor. The speed switching points (see object 0x2042) also affect the end stops.

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	End stop MIN
Access	rw
Data type	SIGNED 32
Value range	-99999..999998 (unit: hundredths of millimeters)
Default	-99999
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The motor stops at this position. Falling below this position is not possible motor-actuated (automatically and via keys). If the motor is moved below the MIN end stop by manual rotation, motor-actuated moving in the ascending direction is possible.

Sub index	0x02
Description	End stop MAX
Access	rw
Data type	SIGNED 32
Value range	-99998..999999 (unit: hundredths of millimeters)
Default	999999
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174

Function	The motor stops at this position. Exceeding this position is not possible motor-actuated (automatically and via keys). If the MAX end stop is exceeded by manual rotation, motor-actuated driving in the descending direction is possible.

4.4.5.17 Object 0x2044 – Motor - Collisions

A minimal value (MIN) and a maximum value (MAX) are available as collision values. Similar to the end stops, the collision values limit the driving range of the motor. Collisions act as additional driving range limitations in addition to the specified fixed end stops.
The speed switching points (see object 0x2042) also affect the collisions.

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	Collision MIN
Access	rw
Data type	SIGNED 32
Value range	-99999..999998 (unit: hundredths of millimeters)
Default	-99999
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The motor stops at this position. Falling below this position is not possible motor-actuated (automatically and via keys). If the MIN collision is not reached by manual rotation, motor-actuated driving in the ascending direction is possible.

Sub index	0x02
Description	Collision MAX
Access	rw
Data type	SIGNED 32
Value range	-99998..999999 (unit: hundredths of millimeters)
Default	999999
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	The motor stops at this position. Exceeding this position is not possible motor-actuated (automatically and via keys). If the MAX end stop is exceeded by manual rotation, motor-actuated moving in the descending direction is possible.

4.4.5.18 Object 0x2045 – Motor – Special set values D1, D2, D3

Special set value specifications D1, D2 and D3 are available for the operating modes *Oscillation, Cycle*, and *ExtPosi*. Depending on the operating mode, these set values have different functions.
Below is a brief overview:

Oscillate: In the operating mode Oscillate a lower reversal point (D1) and an upper reversal point (D2) are available.

Cycle: In the cycle operating mode D1 and D2 are used as two fixed target values, whose positions are indicated motor-actuated via the two external entrances.

ExtPosi: In the operating mode ExtPosi the position is indicated in D3 when the external input "F" is active.

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	Set value D1
Access	rw
Data type	SIGNED 32
Value range	-99999..999999 (unit: hundredths of millimeters)
Default	-99999
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	The set value has the following function in the individual operational modes: Oscillation mode: Set value for the lower reversal point Cycle: Set value for the ext. input "F" ExtPosi: Not used

Sub index	0x02
Description	Set value D2
Access	rw
Data type	SIGNED 32
Value range	-99999..999999 (unit: hundredths of millimeters)
Default	999999
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	The set value has the following function in the individual operational modes: Oscillation mode: Set value for the upper reversal point Cycle: Set value for the ext. input "G" ExtPosi: Not used

Sub index	0x03
Description	Set value D3
Access	rw
Data type	SIGNED 32
Value range	-99999..999999 (unit: hundredths of millimeters)
Default	0
Storage	RAM
PDO mapping	no
Valid for	NM172, NM174
Function	The set value has the following function in the individual operational modes: Oscillation mode: Not used Cycle: Not used ExtPosi: Set value for ExtPosi. Position indication via external input "F"

4.4.5.19 Object 0x2046 – Motor – Step Width for the Jog Function

With object 0x2046 the step width for the function jog (micro step) is defined.
The jog function must be enabled in object 0x2040:03 . Writing and reading to this object is always possible.
See also the function description Section: 3.3.6 Jog function (microstep)

Sub index	0x00
Description	Step width for jog function
Access	rw
Data type	UNSIGNED 32
Value range	0..9999
Default	1
Storage	EEPROM (immediately)
PDO mapping	no
Function	If the jog function is enabled the motor moves by the value stated here when briefly pressing a key. With programming 0, the SPA uses the step width 50.

4.4.5.20 Object 0x2047 – Motor – System times

With object 0x2047 the system times of a position indication are defined. These are holding times or delay times.

The value range for the times extends from 0.1 s to 99.9 s, tolerance: ±7% for the shortest time; < 1‰ for the longest time. For 1s approx. 1%.

The following parameters are writable and readable:

- Loop time (holding time at the loop reversal point)
- Contouring error (time until the motor stops, when motor signals are issued, but the shaft does not turn)
- Clamping (time delay between opening/closing of the clamping and start/stop of the motor)

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	Loop time
Access	rw
Data type	UNSIGNED 32
Value range	1..999 (unit: tenths of seconds)
Default	10 (equals 1 second)
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	Holding time at the reversal point of a loop drive. Note: The moving bit (see object 0x204E:01) remains set during the holding time.

Sub index	0x02
Description	Contouring error time
Access	rw
Data type	UNSIGNED 32
Value range	1..999 (unit: tenths of seconds)
Default	30 (equals 3 seconds)
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	Timeout time for motor signals, if the shaft does not turn after the start of the motor. At the end of the time period the motor is stopped and the error message *Er 3* is displayed.

Sub index	0x03
Description	Clamping
Access	rw
Data type	UNSIGNED 32
Value range	0..999 (unit: tenths of seconds)
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM172, NM174
Function	Time delay between the release/activation of clamping or hold brake before the start of the motor/after the stop of the motor. The following operating modes are possible: 000: Clamping deactivated 001 - 799: Brake handling via clamping **8**00: Permanent motor holding torque without brake handling via clamping **8**01 - **8**99: Permanent motor holding torque with brake handling via clamping 900 Special case: Function same as setting 000 **9**01 - **9**99: Short-term motor holding torque with brake handling via clamping The clamping is controlled via the speed signals of the motor IN3 (pin L) and IN4 (pin D). With IN3 = IN4 = 0, the clamping is activated. Motor cannot be moved. If at least 1 speed output is activated, the clamping is deactivated (open). With the motor holding torque, the two motor control signals for driving right/driving left (IN1, IN2) are activated simultaneously (high level). The motor switches to the hold mode. This is a position control that keeps the motor in the current position.

4.4.5.21 Object 0x2048 – Motor – manual clamping on/off

Object 0x2048 is for manual clamping on/off and only enabled if in object 0x2047:03 a clamping time of 001 - 799 has been defined.

Subindex	0x00
Description	clamping off/on
Access	rw
Data type	UNSIGNED 8
Value Range	Read: 0 .. 2; Write: 0 .. 1
Default	-
Save	no
PDO-Mapping	no
Function	Only enabled with active clamping (see above). The parameters have different meanings in read and write. The following parameters / replies are defined: Read: 0: Clamping feature disabled. Enable via object 0x2047:03 1: Clamping not active (open) 2: Clamping active (closed) Write: 0: deactivate clamping (open) 1: activate clamping (close) Remarks: In mode without clamping (0x2047:03 = 0) the following applies: Read: always 0 Write: without function (command will be ignored) In mode with clamping the following applies: A command for clamping on or off received during a positioing operation (moving bit active) will be ignored. If clamping opened by the function and is followedby a manual or automated positioning operation clamping is automatically re-activated after the positiong operation.

Note:
Open clamping is automatically closed if
- setting a new setpoint without approval in object 0x2020.
- setting a new setpoint without approval via RxPDO

Upon a corresponding motor start command clamping is opened and remains open until the positioning operation has been finalized. Then clamping is closed after to the clamping time defined in object 0x2047:03.

4.4.5.22 Object 0x204D – motor – drive command

With object 0x204D automatic motor-actuated position indication is started or stopped.

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	Position indication start/stop (drive)
Access	rw
Data type	UNSIGNED 8
Value range	0..8
Default	-
Storage	-
PDO mapping	No The drive command is also possible via RxPDO1 (broadcast).
Valid for	NM172, NM174
Function	This command stops or starts motor-actuated position indication. The motor is only enabled if the group number supplied here is defined in the target SPA. See also object 0x2040:04. In addition, active position indication (motor moving) can be stopped. The following parameters are possible: 0: Start release is deactivated, motor stops 1: Start release for SPA group 1 **:** **:** 8: Start release for SPA group 8

Sub index	0x02
Description	Oscillation start/stop
Access	rw
Data type	UNSIGNED 8
Value range	0..8
Default	-
Storage	-
PDO mapping	No The drive command is also possible via RxPDO1 (broadcast).
Valid for	NM172, NM174
Function	This command stops or starts the oscillation function. The oscillation function can only be started or stopped via the interface if the external operation mode is set to 0 = no oscillation. The following parameters are possible: 0: End oscillation 1: Switch on oscillation

Sub index	0x03
Description	Start pressing
Access	rw
Data type	UNSIGNED 8
Value range	11..13; 21..23
Default	0 (only readable)
Storage	-
PDO mapping	No
Valid for	NM172, NM174
Function	This command starts the motor at a predefined speed in the predefined direction without targeting a predefined position. The specified set value is ignored. Thus, this function resembles the moving of the shaft via the device keys or the external inputs. Following a start, the shaft moves at the selected speed in the selected direction until: • An end stop is reached • The motor stopped via the stop command (object 0x204D:01) • An error message is shown on the display • A reset node is carried out • A power interruption occurs The following transmission parameters are possible: 11: Motor rotates at slow speed to the right, counting upwards 12: Motor rotates at medium speed to the right, counting upwards 13: Motor rotates at fast speed to the right, counting upwards 21: Motor rotates at slow speed to the left, counting downwards 22: Motor rotates at medium speed to the left, counting downwards 23: Motor rotates at fast speed to the left, counting downwards Reading the object: If the function pressing is inactive, 0 is returned. If the function pressing is active, the active state 11..13, 21..23 is transmitted back. Note: With parameters 12,13, 22 or 23 prior to reaching the end stop or collision value, the speed is reduced according to the setting in object 0x2042.

4.4.5.23 Object 0x204E – Motor – Reading Status and Error Messages

With object 0x204E the status modes and error messages (errors) of the SPA can be read out.

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	2
Default	2
PDO mapping	no

Sub index	0x01
Description	SPA status
Access	ro
Data type	UNSIGNED 16
Value range	-
Default	-
Storage	-
PDO mapping	yes
Valid for	NM172, NM174
Function	The status information of the SPA can be read out via the object. The following status information is true if the according Bit=1 is set. **Bit Function** 0: Moving bit. Is set while the motor moves including the loop break If clamping is activated, the bit is set as long as the clamping is open. 1: Manual position indication stop. Is set if during automatic position indication a key is pressed on the SPA. The flag remains set until a new set value is transmitted, a start command or power interruption occurs. 2: Oscillation mode is active. The bit is set as soon as the operating mode switch is set to oscillation function. 3: Cycle active. Is set as soon as the operating mode switch is set to position "01" or "10". Bit is deleted with setting "00" and "11". Function "Cycle" must previously be enabled via object 0x2041:03, otherwise the bit is always 0. 4: ExternPosi active. Is set when input F is active. Function "ExternPosi" must previously be enabled via object 0x2041:03, otherwise the bit is always 0. 5: Left key activated. 6: Right key activated. 7: Always 0. 8..9: Motor speed level (IN4, IN3). 10..11: Motor direction signal (IN2, IN1). 12: Arrows active (visible on the display. permanent or blinking). 13. Reserved (always 0). 14: Error bit. Errors are reported in object 0x204E:02. 15: In position. Actual value is within the tolerance window. If the flag is not set, even though the shown position is within the tolerance window, the position was not approached correctly.

Sub index	0x02
Description	SPA error messages
Access	ro
Data type	UNSIGNED 16
Value range	-
Default	-
Storage	-
PDO mapping	yes
Valid for	NM172, NM174
Function	With the object the error messages of the SPA that are also shown on the display can be read out. The following errors are active if the according bit =1 is set. **Bit Function** 0: Err 1 - End stop MAX violated 1: Err 2 - End stop MIN violated 2: Err 3 - Device shaft does not rotate despite drive signal 3: Err 4 - Motor fault (over current) 4: Err 5 - Target window not reached 5: Err 6 - Directional error, spindle is rotating in the wrong direction 6: Reserved (always 0) 7: Always 0 8: Err 8 - Set value > End stop MAX (motor does not move) 9: Err 9 - Set value < End stop MIN (motor does not move) 10: Err A - End stop collision violated. For MIN and MAX 11: Err b - Set value is between end stop and collision. 12: Err C - No encoder signal. Position cannot be read 13: Err d - Encoder reports signal. Position is wrong. Actual value = 0 14: Reserved (always 0) 15: Always 0

4.4.5.24 Object 0x2060 – Sensor parameters

Object 0x2060 contains the parameters for adjusting an external SSI sensor to the SPA.

The following parameters are writable and readable:

- Resolution in bit of the SSI value
- Code of the position indication value (binary or gray)
- Sensor type

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	3
Default	3
PDO mapping	no

Sub index	0x01
Description	SSI value data width
Access	rw
Data type	UNSIGNED 32
Value range	12..32 (unit: bit)
Default	25
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM174
Function	Number of transmitted data bits of the complete position value.

Sub index	0x02
Description	Code
Access	rw
Data type	UNSIGNED 32
Value range	0..1
Default	1
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM174
Function	Transmission code of the position value. 0: Binary 1: Gray

Sub index	0x03
Description	Sensor type
Access	rw
Data type	UNSIGNED 32
Value range	0..4; 9
Default	0
Storage	EEPROM (immediately)
PDO mapping	no
Valid for	NM174
Function	Sensor type defines different sensors for the functions length measurement or angel display. Not every sensor can be used. The sensor must always use an SSI interface. The sensors from selection parameter 2 are devices with special additional functions that must also be present. 0: Length measurement with rotating sensor (encoder) 1: Length measurement with linear sensor (length measuring device) 2: Angle display with single-turn rotary encoder left-justified SSI version 3: Dunker motor with AEM 65 incl. evaluation of the battery status. 4: Turck Li200 with evaluation of the transmitted error bits 9: Angle display with single-turn rotary encoder and right-justified SSI version (25 bit).

4.4.5.25 Object 0x2100 – Baud rate CAN bus

With object 0x2100 the baud rate for the SPA on the CAN bus is specified.

Sub index	0x00
Description	Baud rate
Access	rw
Data type	UNSIGNED 8
Value range	0..8
Default	5
Storage	Flash (immediately)
PDO mapping	no
Function	After setting the baud rate, the new baud rate is saved in the flash without implementation of the save parameter (object 0x1010:02). However, the new baud rate only takes effect after a power interruption or the NMT command reset node. The following baud rates are possible: 0=10 kBit/s 1=20 kBit/s 2=50 kBit/s 3=100 kBit/s 4=125 kBit/s 5=250 kBit/s 6=500 kBit/s 7=800 kBit/s 8=1000 kBit/s

4.4.5.26 Object 0x2101 – Node ID

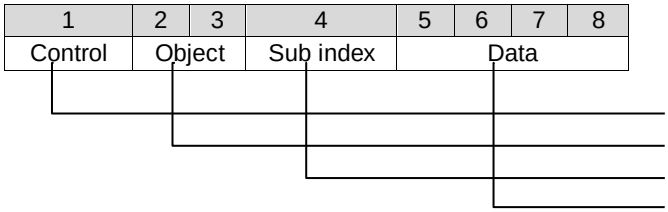
With object 0x2101 the node ID for the SPA is specified.
Note: The default setting is defined at 127.

Sub index	0x00
Description	Node ID
Access	rw
Data type	UNSIGNED 8
Value range	1..127
Default	127
Storage	Flash (immediately)
PDO mapping	no
Function	The node ID is saved in the flash without implementation of the save parameter (object 0x1010). However, the new node ID only takes effect after a power interruption or the NMT command reset node.

4.4.5.27 Object 0x2500 – PDO Broadcast Object

The object contains the data that was received via the Receive PDO1. Reading the object returns the 4 data blocks of the most recent broadcast command. Writing to this object is possible but does not have a functional effect. The broadcast function can only be implemented via RxPDO1. Following a power interruption, all index directories are set to 0. See also the function description Section: 4.3.4.2.

RXPDO1



Index	Sub index
0x2500	1
0x2500	2
0x2500	3
0x2500	4

Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	4
Default	4
PDO mapping	no

Sub index	0x01
Description	Broadcast - control register (reserved)
Access	rw
Data type	UNSIGNED 8
Value range	0..255
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Reserved for subsequent extensions. Transmitted values are ignored by the SPA.

Sub index	0x02
Description	Broadcast – SDO object number
Access	rw
Data type	UNSIGNED 16
Value range	0x2001: Length unit (mm/Inch) 0x2008: Show device address 0x204D: Drive command 0x2015: SPA resets 0x6003: Carry out preset
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Contains the object number of the SDO command that is carried out as a broadcast.

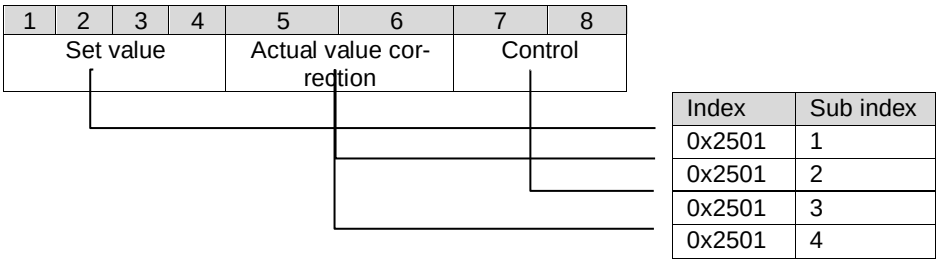
Sub index	0x03
Description	Broadcast – SDO sub index
Access	rw
Data type	UNSIGNED 8
Value range	0..UNSIGNED 8 (sub index associated with the SDO object)
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Contains the sub index of the object number of the SDO command that is carried out as a broadcast.

Sub index	0x04
Description	Broadcast – SDO data
Access	rw
Data type	UNSIGNED 32
Value range	0..UNSIGNED 32
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Contains the data of the SDO command that is carried out as a broadcast.

4.4.5.28 Object 0x2501 – PDO Set Value Object

The object contains the data that was received via the Receive PDO2. Reading the object returns the 3 data blocks of the most recent PDO set value command. Writing to this object is possible but does not have a functional effect. The PDO set value function can only be implemented via RxPDO2. Following a power interruption, all index directories are set to 0. See also the function description Section 4.3.4.3

RXPDO2



Sub index	0x00
Description	Number of entries
Access	ro
Data type	UNSIGNED 8
Value range	4
Default	4
PDO mapping	no

Sub index	0x01
Description	PDO set value object – set value
Access	rw
Data type	UNSIGNED 32
Value range	0..UNSIGNED 32
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Contains the set value of the PDO position indication

Sub index	0x02
Description	PDO set value object – actual value correction (mapping writing to RxPDO2)
Access	rw
Data type	UNSIGNED 16
Value range	0..UNSIGNED 16
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Contains the value for the actual value correction of a PDO position indication.

Sub index	0x03
Description	PDO set value object - control register
Access	rw
Data type	UNSIGNED 16
Value range	0..UNSIGNED 16
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Contains the value for the control register of a PDO position indication.

Sub index	0x04
Description	PDO set value object – actual value correction (mapping reading to RxPDO2)
Access	rw
Data type	UNSIGNED 16
Value range	0..UNSIGNED 16
Default	0
Storage	-
PDO mapping	yes
Valid for	NM170, NM172, NM174
Function	Contains the value for the actual value correction of a PDO position indication. **Note:** This object is identical to sub index 02h (a copy). An object cannot be simultaneously mapped to a PDO in reading and in writing. Therefore there is this copy.

4.4.6 Detailed object list – Device profile specific objects

4.4.6.1 Object 0x6003 – Implement Preset

The actual value of the SPA is specified via object 0x6003.
Writing the preset value results in the non-volatile specification of the actual position to this value.

The SPA determines an offset value "preset offset" for the actual absolute physical position. When calculating the preset offset the actual value (object 0x2010) is taken into account in order for the actual value at the time of the preset implementation is equal to the preset value specified here.

Sub index	0x00
Description	Implement preset
Access	rw
Data type	SIGNED 32
Value range	-99999..999999
Default	-
Storage	EEPROM (immediately)
PDO mapping	no Setting the preset is also possible via RxPDO1 (broadcast).
Function	For technical reasons, the preset offset is determined without scaling. Therefore in rare instances rounding errors may occur, which result in the actual value being shown 1 LSB smaller or larger to the programmed preset value. The shown actual value is structured as follows: Actual value display = Actual value absolute + Preset offset + Actual value correction Actual value display: Actual value that is shown on the display and read out via the CAN interface. Actual value absolute: Internal absolute actual value (after scaling). Preset offset: Offset value created via object 0x6003. Actual value correction: Offset value that can be specified in addition to object 0x2010. However, the function must be enabled via object 0x2000:06. Otherwise it applies that: Actual value correction = 0.

4.4.6.2 Object 0x6004 – Position value (actual value)

The actual value of the SPA is read out via object 0x6004.

Sub index	0x00
Description	Position value (actual value)
Access	ro
Data type	SIGNED 32
Value range	-99999..999999
Default	-
Storage	-
PDO mapping	yes
Function	The position value is always read out in 1/100 mm (also when set to inch). **Dependency on Actual Value Correction:** Calculation of the position depends on the specification of the actual value correction in object 0x2000:06. Calculation is based on the following formulas: Object 0x2000:06 = 0: $Position_{6004} = Position_{Dis}$ Object 0x2000:06 = 1: $Position_{6004} = Position_{Dis} - Offset$ Offset: Value for actual value correction. See object 0x2010 $Position_{Dis}$: Position value shown on the display

5 Diagnostic tools

5.1 Error Diagnosis Fieldbus Communication

- If the spindle position display cannot be addressed via the CANopen bus, you should check the connections first.

If the connections are Ok, then you should test the function of the field bus next. This requires a CAN monitor that records the CANopen communication and presents the telegrams.

- Now the SPA should issue a BootUp message when the power supply is switched on and off.

If no BootUp message appears, check whether the baud rates of the SPA, the CAN monitor and the bus system are matching.

- If you have problems setting up a connection to a participant, check the node number and the baud rate.

Within a CAN network all participants (Master and Slaves) must be configured to the same baud rate. The node number (node ID) must be between 1 and 127. Each bus participant must be clearly defined with a node ID. This means that under no circumstances should the same node ID be allocated several times.

5.2 Error Diagnosis via Fieldbus

The spindle position display contains several objects and messages that described the status or error situations of the spindle position display:

- Object 0x1001: This object is an error register for the error status of the device.
- Object emergency (80h + node ID): High priority error message of a participant with error code and error register.
- SDO abort message: If the SDO communication does not proceed correctly, the SDO reply contains an abort code.
- Status register. Errors that are shown on the display.

Object 0x1001 Error Register
In this register the occurrence of a device error and its type are shown.
See the separate object description in Section 4.4.4.2.

Object Emergency
Error message of a participant.

SDO Abort Message
If SDO communication is not trouble free, the SDO reply is an abort code:
See overview: CANopen SDO Abort Messages

Status Register
If functional errors occur during operation (not interface errors), they are shown on the display (see Section 6.3.2). At the same time, but 14 is set in the status register. The status register is transmitted via TxPDO1 and TxPDO2. In addition, bit 7 is set in the error register object 0x1001.
The precise cause of the error can be read out via object 0x204E:02. The status register can also be read out via the SDO from object 0x204E:01. See also Section 4.4.5.23.

6 Messages

6.1 CANopen Emergency Messages

COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80h+Node ID	8	**Error code**		Object 0x1001	Manufacturer specific				
					Not used		Not used		**Sub code**

Emergency messages of the CANopen interface

Error code	Sub code	Description
81 00	00	CAN communication error
81 10	00	CAN buffer overflow
81 20	00	CAN Passive
81 30	00	Heartbeat error
81 40	00	BUS Off Recover
81 50	00	CAN ID collision

Emergency messages of the application

Error code	Sub code	Description
82 10	01	RxPDO does not exist.
82 20	01	Object not found in the implementation table.
82 20	02	Transmission storage area (FIFO) for the transmission of data to the MSP µ-controller too small.
82 20	03	Data type not correct.
82 30	01	RxPDO2: Invalid value in the register control.
83 01	01	FIFO storage for communication among the internal controllers too small.
83 10	01	Data (position, status, errors) could not be read several times
83 10	02	Error occurred during the data transmission to the MSP µ-controller. No valid reply received.
83 10	03	Faulty data received while reading the MSP µ-controller.
83 10	04	Data (position, status, errors) could not be read more than 3 times.
83 10	05	Unknown status of the application status machine occurred.
83 20	01	Buffer overflow during transmission to the MSP µ-controller.
83 20	02	Motor stop command could not be transmitted.
83 20	03	Addressing commands could not be transmitted to the MSP µ-controller.
83 20	04	Unknown status of the application status machine during the action "write data to MSP" occurred.

6.2 CANopen SDO Abort Messages

Abort code	Meaning
05040001h	Command byte is not supported.
06010000h	Faulty access to an object.
06010001h	Read access to Write Only.
06010002h	Write access to Read Only.
06020000h	Object not supported.
06060000h	SDO hardware error or SDO firmware error (internal overflows, overload).
06090011h	Sub index is not supported.
06090030h	Value beyond the limits.
06090031h	Value too large.
08000000h	General error.
08000020h	Faulty storage signature ("save").
08000021h	Data cannot be stored.
08000024h	No data available.

6.3 SPA error messages

6.3.1 Overview

Error code	Description
Er 1	In the SPA the end stop MAX was violated through manual moving via the keys.
Er 2	In the SPA the end stop MIN was violated through manual moving via the keys.
Er3	The SPA issues a motor drive signal. Yet the hollow shaft of the SPA does not move.
Er4	There is a motor fault in the SPA.
Er5	The target window was not reached in the SPA.
Er6	Directional error. Spindle position indication is taking place in the wrong direction.
Er8	Transmitted set value or DIM value above the MAX end stop. Note: consider the loop travel.
Er9	Transmitted set value or DIM value below the MIN end stop. Note: consider the loop travel.
Er A	In the SPA the end stop collision MIN or MAX was violated through manual moving via the keys.
Er b	Transmitted set value is outside the specified collision values. Note: consider the loop travel.
Er C	In the remote display NM174 there is a fault in the SSI sensor. Cable break or not connected.
Er d	In the remote display NM174 there is a fault in the SSI sensor.
dEF9Eb	In the remote display NM174 there is a fault in the SSI encoder Dunker AEM65.

These error messages can also be read out via the CAN interface. There are two options for this:

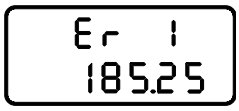
- via SDO object 0x204E:02
- via TxPDO1 and TxPDO2

If there is at least one error message, then bit 7 "manufacturer-specific error" is also set in object 0x1001. For correction of these errors see the following Section.

6.3.2 Presentation in the Display

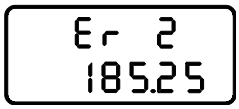
The applicative error messages of the SPA are shown on the display flashing in the top line. The error message and the current display (usually the set value) are shown alternately.

Valid for: NM142, NM144



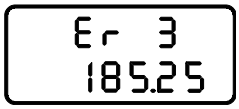
In the SPA the end stop MAX was violated through manual moving via the keys.

Error correction:
Move the SPA to the valid position area.



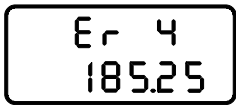
In the SPA the end stop MIN was violated through manual moving via the keys.

Error correction:
Move the SPA to the valid position area



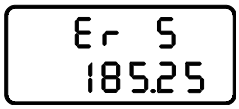
The SPA issues a motor drive signal. Yet the hollow shaft of the SPA does not move.

Error correction:
Check the function of the motor. Check the wiring and mechanics from the motor via the shaft to the SPA.



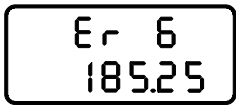
There is a motor fault in the SPA.
This error is shown when an automatic or manual drive command (start command object 0x204D via CAN interface or key activation) is actuated. If an error Er 04 occurs, the motor signals "left"/"right" are immediately deactivated. The error message itself remains active for approx. 5 seconds. Reading out the error message via the interface is possible within these 5 seconds. In the event of short-term motor error (< 1 sec) the error message appears on the display for about 1 second.

Error correction:
Amend motor error.



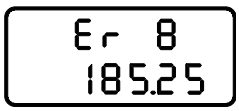
The target window was not reached in the SPA.

Error correction:
Select a larger tolerance window. Set the switch to creep speed earlier
.



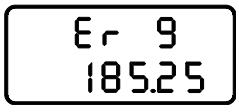
Directional error. Spindle position indication is taking place in the wrong direction.

Error correction:
Programming of the SPA (e.g. check motor rotational direction or cable connection to the motor)



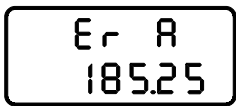
The transmitted set value or DIM value is above the MAX end stop.
Note: consider the loop travel.

Error correction:
Transmit new valid position value.



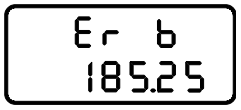
The transmitted set value or DIM value is below the MIN end stop.
Note: consider the loop travel.

Error correction:
Transmit new valid position value.



In the SPA the end stop collision MIN or MAX was violated through manual moving via the keys.

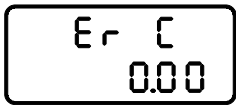
Error correction:
Move the SPA to the valid position area.



The transmitted set value is outside the specified collision values.
Note: consider the loop travel.

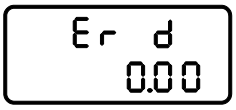
Error correction:
Transmit new valid set value.

Only applicable to: NM174



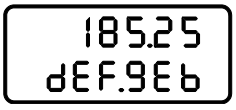
In the remote display NM174 there is a fault in the SSI sensor.
This error is shown when no SSI encoder is connected, a cable break is present.
While the error is active, the outputs for motor signals IN1 to IN2 are inactive. Actuating driving via the keys or automatically is not possible.

Error correction:
Correct error of the SSI rotary encoder



In the remote display NM174 there is a fault in the SSI sensor *Turck Li200*.
This error is shown when the connected sensor reports an error.
While the error is active, the outputs for motor signals IN1 to IN2 are inactive. Actuating driving via the keys or automatically is not possible.

Error correction:
Correct error of the SSI rotary encoder.



In the remote display NM174 there is a fault in the SSI encoder *Dunker AEM65*.
This error is displayed when the battery of the AEM65 is empty.
While the error is active, the outputs for motor signals IN1 to IN2 are inactive. Actuating driving via the keys or automatically is not possible.

Error correction:
Correct error of the SSI rotary encoder.

Sequence of the error display with multiple errors

If several errors occur simultaneously, only one error can be displayed at a time. The following list shows which error is displayed:

High priority

Er C	Er d	Er 4	Er 1	Er 2	Er3	Er5	Er 6	Er 8	Er 9	ErA	Er b

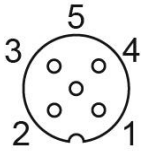
low priority

7 Connection assignment and commissioning

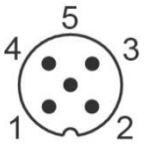
7.1 Electrical connection

7.1.1 Supply and interface cable

M12 socket, 5-pin- A-coded

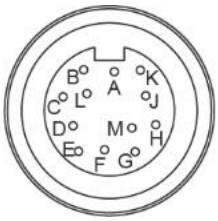
Pin	Assignment	Description	M12 socket
1	Shield	Cable shield	
2	+Vs	Operating voltage	
3	GND	Ground connection for +Vs	
4	CAN-H	CAN Bus Signal (dominant High)	
5	CAN-L	CAN Bus Signal (dominant low)	

M12 connector, 5-pin – A coded

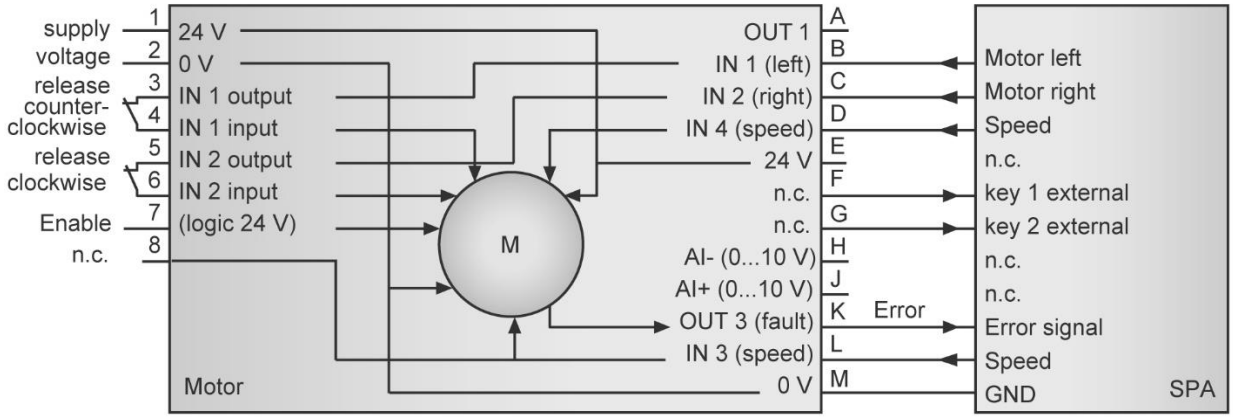
Pin	Assignment	Description	M12 connector
1	Shield	Cable shield	
2	+Vs	Operating voltage	
3	GND	Ground connection for +Vs	
4	CAN-H	CAN Bus Signal (dominant High)	
5	CAN-L	CAN Bus Signal (dominant low)	

7.1.2 Motor cable (only NM172, NM174)

M16 socket, 12-pin

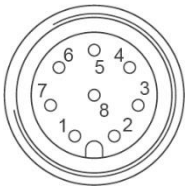
Pin	Assignment	Description	M16 socket
A	n.c	Not consigned	
B	IN1	Motor rotation left	
C	IN2	Motor rotation right	
D	IN4	Speed 2	
E	n.c.	Not consigned	
F	KEY 1	Key 1 external	
G	KEY 2	Key 2 external	
H	n.c.	Not consigned	
J	n.c.	Not consigned	
K	OUT	Motor malfunction	
L	IN3	Speed 1	
M	GND	Ground	

Motor circuit sketch



7.1.3 Sensor connector (only NM174)

M12 connector, 8-pin – SSI sensor

Pin	Assignment	Description	M12 connector
1	GND	Ground	
2	+Vs	Power supply for sensor	
3	Clock +	SSI clock signal +	
4	Clock -	SSI clock signal -	
5	Data +	SSI data signal +	
6	Data -	SSI data signal -	
7	n.c.	Not consigned	
8	n.c.	Not consigned	

8 List of abbreviations

CiA
CAN in Automation: the international users' and manufacturers' group for the promotion and standardization of the CAN fieldbus protocol technology.

COB ID
Communication Object Identifier

CS
LSS command specifier

DLC
Data Length Code. Contains the number of the data bytes of the CAN telegram to be transmitted.

EMCY
Communication object: time stamp and error messages

EMC
Electromagnetic compatibility

LSS
Layer Setting Service

MSP
Micro-controller in the SPA that contains the actual SPA functions.

NMT
Communication object: network management for the finite automation control and hub monitoring

PDO
Communication object: process data object. Transmission channel for cyclical process data

SDO
Communication object: service data object. Transmission channel for parameter data

SPA
Spindle position indicator (NM170, NM172 or NM174)

PLC
Programmable Logic Controller

STM
Micro-controller in the SPA that sets up the connection to the outside to the upstream control via the bus interface CANopen.

SYNC
Communication object: synchronization object