User Manual

Positioning motor *FlexiDrive* MSBA with CANopen

Software Revision (Firmware Version) 2.00.xx

Baumer Electric AG
Hummelstrasse 17
8501 Frauenfeld · Switzerland
Phone +41 52 728 11 22
Fax +41 52 728 11 44
E-Mail: sales.ch@baumer.com
www.baumer.com

04.05.2015
Subject to technical and design modification
Errors and omissions excepted

Baumer_MSBA_CANopen_V2 00 xx rev04_MA_EN.docx

Contents

Disclaimer of liability
The present manual was compiled with utmost care, errors and omissions reserved. For this reason Baumer rejects any liability for the information compiled in the present manual.
Baumer nor the author will accept any liability for direct or indirect damages resulting from the use of the present information.

At any time we should be pleased receiving your comments and proposals for further improvement of the present document.

Created by:
Baumer IVO GmbH & Co. KG
Villingen-Schwenningen, Germany

1 Introduction

1.1 Product assignment

Table 1: Product assignment.

Product family	Product	Product code	EDS file	Interface
FlexiDrive	MSBA	102h	MSBA_xxC.eds	CANopen

1.2 Function overview

- Intelligent positioning motor as decentralized CAN bus participant
- Absolute multiturn position detection (3 Bit singleturn, 29 bit multiturn
- 2Q-controller (generator mode is possible)
- Building set made of planetary, spur or worm gears
- 2 digital inputs for Jogging, hardware limit switch or homing
    - triggered by standard position sensor
    - various functions programmable (Object 2111h-0Dh)
- Operating modes (Object 6060h):
    - positioning
    - homing
- Additional functions
    - software limit switches to limit moving path (Object 607Dh)
    - hardware limit switches (Object 2111h-0Dh)
    - set target value (Object 607Ah)
    - position controller
    - standstill detection (Object 6041h)
    - backlash compensation against gear or spindle play (Object 2111h-01h)
    - torque / current limit (Object 6073h)
    - linear- or rotational axis (Object 608Ah)
    - direction of rotation can be reversed (Object 607Eh)
    - stand-alone / Jogging operation without CAN bus and programming (Object 2111h-13h)
    - comfortable Homing functions (Object 6098h)
        - set actual position (Object 2111h-11h)
        - move to block
        - measuring method
    - status and error diagnosis via CAN bus (Object 1003h and 603Fh)
    - temperature monitoring with warning message and drive shut-down in case of error (Object 2114h)
    - Over- and undervoltage monitoring
    - Boot-up current limitation
    - Bootloader for loading of new firmware
    - LED status display
- Communication via CANopen
    - SDO communication (read / write parameters)
    - PDO communication (drive control, send status and position)
    - Galvanic isolation of the fieldbus communication for optimal resistance against disturbances
- Quick start-up
    - adaptation of only little number parameters
    - fieldbus connection can be looped through
    - PC software CANmaster for programming and start-up (see CD-ROM Art. Nr. 10147362)
- Removable cover for access of integrated switches to select node-ID, baud rate and terminating resistor
- Non-volatile memory (EEPROM) for default and user data (object 2300h)

1.3 Safety and operating instructions

Supplementary information

- This manual is intended as a supplement to already existing documentation (catalogue, product information and assembly instructions).
- Please read this manual carefully before initial appliance of the device.

Intended purpose of the device

- This motor is a precision measurement and actuator device. It is intended to adjust angular positions of axles and spindles and to prepare and supply measured position values as electrical output signals for follow-on device systems. The motor may be used for this purpose only.

Commissioning

- The motor may be installed by suitably qualified experts only.
- Observe all operating instructions of the machine manufacturer.

Safety remarks

- Prior to commissioning of the equipment, check all electrical connections.
- If installation, electrical connection or any other work performed at the motor or at the equipment is not correctly executed, this can result in a malfunction or failure of the motor.
- Steps must be taken to exclude any risk of personal injury, damage to the plant or to the operating equipment as a result of motor failure or malfunction by providing suitable safety precautions.
- Motor must not be operated outside the specified limited values (see detailed product documentation).

Failure to comply with the safety remarks can result in malfunctions, personal injury or damage to property.

Transport and storage

- Transport or store motor in original packaging only.
- Never drop motor or expose it to strong vibrations.

Assembly

- Avoid impacts or shocks on housing or shaft.
- Avoid any twist or torsion on the housing.
- Do not open the motor or perform any mechanical changes.

Failure to comply with the safety remarks can result in malfunctions, personal injury or damage to property.

Electrical Commissioning

- Do not perform any electrical changes at the motor.
- Do not carry out any wiring work when the motor is connected to supply voltage.
- Never plug or unplug the electrical connection when the motor is connected to supply voltage.
- Ensure that the entire plant is installed in line with EMC requirements. The installation environment and wiring affects the electromagnetic compatibility of the motor. Install the motor and supply cables separately or at a long distance from cables with high interference emissions (frequency converters, contactors, etc.)
- Where working with devices having high interference emissions, make available a separate power supply for the motor.
- Completely shield the motor housing and connecting cable.
- Connect the motor to the protective earth (PE) conductor by using shielded cable. The braided shield must be connected to the cable gland or plug. Ideally, aim at bilateral connection to protective earth (PE), the housing via the mechanical assembly, the cable shield via the downstream connected devices. In case of earth loop problems, earth on one side only as a minimum requirement.

Failure to observe these instructions can result in malfunctions, material damage or personal injury.

2 Communication via CANopen

2.1 General information

The CAN bus (Controller Area Network) was developed by Bosch and Intel to enable fast and economic communication in automotive applications. Today, the CANopen bus is one of the widest used fieldbus interfaces in industrial automation. The user association “CAN in Automation” (CIA) defines and releases standardized regulations for this fieldbus.

CAN bus properties

- Data rate up to 1 Mbit/s with network extension up to 25 m
- Network terminated on both sides.
- Network connection with twisted-pair cables.
- Real time capability: Defined maximum waiting time for high-priority messages.
- Theoretically up to 127 bus participants or network nodes are possible on one bus, but limited through drivers it is only 32.
- Data consistency across network is ensured. Damaged messages are notified as faulty for all network nodes.
- Message-oriented communication: The message is identified by a message identifier. All network nodes use this identifier to check if the message is sent to them.
- Broadcasting and multicasting: All network nodes receive each message simultaneously. Therefore synchronization is possible.
- Multi-Master capability: Each bus participant is able to independently transmit and receive data without depending on priority settings from the master. Each bus participant is able to start its message when the bus is not busy. When messages are sent at the same time, the message with highest priority prevails.
- Prioritization of messages: The message priority is set by the identifier. This ensures a fast transmission of important messages.
- Residual error probability: Safety procedures in the network reduce the probability of an undiscovered data transmission error below 10^{-11}. This leads to a 100 per cent safe communication.
- Function monitoring: Localization of faulty or stopped network nodes. The CAN protocol contains a network node monitoring function. The function of faulty network nodes is restricted, or they are completely uncoupled from the network.
- Data transmission with a short error recovery time: By using several error detection mechanisms, falsified messages are detected with high probability. If an error is detected the message transmission is automatically repeated.

In the CANopen bus architecture several network participants are connected via a bus cable. Each network participant can receive and transmit messages. The data are transmitted serially between the network participants.

Examples for network participants or CANopen bus devices are:

- Automation devices such as PLC or PC
- Input and output modules
- Drive control systems
- Analysis devices, such as a CAN-monitor
- Control and input devices as Human Machine Interfaces (HMI)
- Sensors and actuators

2.2 CANopen basics

Under technical management of the Steinbeis Transfer Center for Automation, the CANopen profile was developed on the base of Layer 7 specification of CAL (CAN Application Layer). In comparison with CAL, CANopen only contains the functions suitable for this application. CANopen thus represents only a partial function of CAL optimized for applications, to support a simplified system structure and the use of simplified devices. CANopen is optimized for fast data exchange in real time systems. The organization CAN in Automation (CiA) is responsible for the applicable standards of the relevant profiles.

CANopen enables:

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

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

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

All device and communication parameters are subdivided into an object directory. An object directory contains the name of the object, data type, number of sub-indexes, structure of the parameters and the address (refer to chapter object description). This object directory is subdivided into three different parts:

- Communication profile
- Device profile
- Manufacturer-specific profile

2.2.1 Communication profile

Communication between network users and master is established with object directories and objects. The objects are addressed via a 16-bit index. The CANopen communication profile DS-301 sets the standard for various communication objects. They are divided into several groups:

- Process data objects PDO for real time transmission of process data
- Service data objects SDO for read/ write access to the object directory
- Objects for synchronization and error display of CAN users:
  SYNC object (synchronization object) for synchronization of network users
  EMCY object (emergency object) for error display of a device or its peripherals
- Network management (NMT) for initialization and network control
- Layer Setting Services (LSS) for configuring by serial or revision number within an existing network

2.2.2 CANopen message structure

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

Structure of the 11-bit COB-ID:

Function code	Node-ID
4-bit function code	7-bit node-ID

The function code provides information on type and priority of message. A low COB-ID corresponds to a high message priority.

Broadcast message:

Function code	COB-ID
NMT	0
SYNC	80h

Peer-to-peer message:

Function code	COB-ID
EMERGENCY	80h + Node-ID
T_PDO1 (tx)[1]	180h + Node-ID
R_PDO1 (rx)[1]	200h + Node-ID
T_PDO2 (tx)[1]	280h + Node-ID
R_PDO2 (rx)[1]	300h + Node-ID
SDO (tx)[1]	580h + Node-ID
SDO (rx)[1]	600h + Node-ID
Nodeguard, Heartbeat	700h + Node-ID
LSS (tx) [1]	7E4h
LSS (rx) [1]	7E5h

[1](tx) and (rx) according to view from drive.

For details about adjustment of node-ID of bus participant, see following chapters.

A CAN telegram consists of the COB-ID and up to 8 data bytes. For details, see following chapters.

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Xxx	x	xx	xx	xx	xx	xx	xx	xx

DLC (data length code) stands for the length of the telegram.

2.2.3 Service data object communication (SDO)

The service data objects (SDO) correspond to the standards of the protocol DSP-402. It is possible to access an object via index and sub-index. The data can be requested or written into the object.

An SDO telegram consists of a command byte, two object bytes, a sub-index byte and up to four data bytes. It is composed as follows:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3

The COB-ID of an SDO is composed as follows:

Communication master to slave: 600h + Node-ID
Communication slave to master: 580h + Node-ID

Table 2: SDO command description. The data length code (DLC) contains the telegram length. The command byte defines if data are read or written and how many data bytes are attached

SDO command	Description	DLC	Remark
22h	Download request	Max. 4 byte	Master sends parameter to drive
23h	Download request	4 byte	Master sends parameter to drive
2Bh	Download request	2 byte	Master sends parameter to drive
2Fh	Download request	1 byte	Master sends parameter to drive
…			
60h	Download response	-	Confirms receipt to master
40h	Upload request	-	Master requests parameter from drive
…			
42h	Upload response	Max. 4 byte	Parameters to master with max. 4 byte
43h	Upload response	4 byte	Parameters to master
4Bh	Upload response	2 byte	Parameters to master
4Fh	Upload response	1 byte	Parameters to master
…			
80h	Abort message	-	Drive sends abort code to master

An SDO abort message indicates an error in CAN communication. The corresponding SDO command is 80h. Object and sub-index are those of the requested object. The SDO abort code is indicated in byte 4...7 (Byte 7 = MSB).

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
580h+Node-ID	8	80h	Object L	Object H	Subindex	ErrByte 0	ErrByte 1	ErrByte 2	ErrByte 3

For SDO abort code definition, see chapter 5 error diagnosis.

SDO communication examples

Request of a drive value from the master. A frequently used request is the status request from master to slave (see Object 6041h).

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	40h	41h	60h	0	x	x	x	x

Slave response to master. The status length is 4 byte (see Object 6041h).

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	43h	41h	60h	0	a	b	c	d

Write a value from master to slave. The slave is controlled by controlword (see Object 6040h).

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	22h	40h	60h	0	a	b	x	x

Slave response to master on writing a value.

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

2.2.4 Receive process data object communication (PDO)

Receive process data objects (R_PDO) are used to exchange real time data such as a new target or the control word. R_PDO can be received by the drive synchronously or asynchronously. The drive supports static R_PDO mapping with R_PDO1 (control word) and R_PDO2 (control word and target position).

Synchronous communication

With synchronous communication the process data are received by the drive at the same time with the SYNC telegram. In order to set drive to synchronous R_PDO communication, write a value between 1 and F0h into Object 1400-2h or 1401-2h. With value = 3, the R_PDO is received with every third SYNC-telegram.
SYNC telegram:

Byte 0	Byte 1
COB-ID = 80	0

Asynchronous communication

With asynchronous communication the R_PDO are overtaken by the drive directly after receipt. In order to set drive to asynchronous R_PDO communication, write value = FEh into Object 1400-2h and 1401-2h.

1400h-02h	Description
FEh	R_PDO are overtaken immediately after receipt
01h	R_PDO are overtaken after the next SYNC

R_PDO1 (Control word)

Telegram:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
200h+Node-ID	2	xx	xx	-	-	-	-	-	-

COB-ID : 200h + Node-ID
DLC : Length 2 Data byte
Byte 0..1 : new control word (Object 6040h Controlword)

R_PDO2 (Control word and target position)

Telegram:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
300h+Node-ID	6	xx	xx	xx	xx	xx	xx	-	-

COB-ID : 300h + Node-ID
DLC : Length 6 Data byte
Byte 0..1 : new control word (Object 6040h Controlword)
Byte 5..2 : new target (Object 607Ah Target position)

2.2.5 Transmit process data object communication

Transmit process data objects (T_PDO) are used to exchange real time data of process data such as drive status or drive position from master to drive. T_PDO message sending can be set in synchronous or asynchronous (cyclic) mode

Synchronous communication

With synchronous communication the T_PDO are requested by the master by the SYNC telegram. In order to send synchronous process data, write value = 1…F0h into Object 1800-2h or 1801-2h. With value = 3, a T_PDO response is sent to every third SYNC-telegram. The Object 2800h or 2801h define if a message is sent only after a change of the object.

SYNC telegram:

Byte 0	Byte 1
COB-ID = 80	0

Asynchronous (cyclic) communication

In order to set drive to cyclic T_PDO communication, write value = FEh into Object 1800-2h and 1801-2h. The cycle time in milliseconds is defined in Object 1800-05h and 1801-05h. The minimum cycle time is 1 millisecond. For value = 0, no T_PDO are sent and this function is deactivated. In Object 2800h and 2801h can be defined, if T_PDO sending should be done only on change of parameter. For value = 0, cyclic sending is activated as described above. For value = 1, a cyclic check is done for parameter change and the parameter value is sent, if parameter has changed. If no change occurred, the parameter will not be sent. For value = 4, the T_PDO is sent 4 times in the pre-defined cycle.
Alternatively the communication type FFh can be set in Object 1800-02h and 1801-02h respectively. If so, the T_PDO is sent immediately after the parameter has changed. Additionally Object 1800-05h and 1801-05h allow to configure a cyclic transmission which takes place also if the parameters do not change. In order to prevent bus over load in case of very frequent changes it is proposed to establish a silent period in Object 1800-03h and 1801-03h. This is the waiting time between two T_PDO.

Table 3: Examples for T_PDO1 communication settings.

1800h resp. 1801h			2800h resp. 2801h	Description
Subindex 02h	Subindex 03h	Subindex 05h		
FEh	xxx	10 ms	0h	Cyclic sending of T_PDO1 every 10 ms
FEh	xxx	10 ms	1h	Every 10ms, the T_PDO is sent once (if a change occurred)
FEh	xxx	10 ms	2h	Every 10ms, the T_PDO is sent twice (if a change occurred)
FEh	xxx	0 ms	x	Switch off T_PDO sending
FFh	5 ms (32h)	0 ms (0h)	xxx	T_PDO is sent immediately after parameter change. Afterwards no T_PDO is sent for 5 ms.
FFh	5 m (32h)	10 ms (0Ah)	xxx	T_PDO is sent immediately after parameter change. If it does not change the data is sent every 10 ms. Afterwards no T_PDO is sent for 5 ms.
03h	xxx	xxx	00h	T_PDO is sent on every third SYNC telegram
03h	xxx	xxx	2Bh	T_PDO is sent on every third SYNC telegram, but in total only 43 times (43 =2Bh).

T_PDO1 (Status)

Telegram structure:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
180h+Node-ID	x	xx	xx	xx	-	xx	xx	xx	xx

COB-ID	: 180h + node-ID
DLC	: 2, 3 or 8 data byte (can be adjusted in Object 2110h Version control)
Byte 0…1	: Statusword
Byte 2	: Additional Statusword (if this option is set in Object 2110h Version control)
Byte 3	: Never contains information
Byte 4…7	: Indication of the digital hardware inputs (if this option is set in Object 2110h Version control)

Table 4: Statusword (Object 6041h) is transmitted in byte 0 and byte 1 of T_PDO1 telegram.

Bit	Description
0	Ready to switch on
1	SwitchOn enabled
2	Operation enabled
3	Error active
4	Voltage enabled
5	QuickStop active
6	SwitchOn disabled
7	Warning active
8	CalibrationOK
9	Not used (permanently 1)
10	Target position reached
11	Internal software limit switch active
12	Drive moving
13	Not used (permanently 0)
14	HomingOK
15	External hardware limit switch active

Table 5: Additional statusword content in byte 2 of T_PDO1 telegram for transfer of internal drive status. Per default the additional statusword content is activated. Standard length of T_PDO1 telegram is 3 byte. Deactivation of additional statusword content is done in Object Version control (Object 2110h, bit 1).

Bit	Byte2	Description
SELF_TEST	00	Selftest after initialization
READY	01	After successful initialization -> Ready
POSITIONING_READY	02	SwitchOn, EnableVoltage, QuickStop, EnableOperation and Positioning -> PositioningReady
HOMING_READY	03	SwitchOn, EnableVoltage, QuickStop, EnableOperation and Homing -> HomingReady
POSITIONING_MOVING	04	Switch on drive (bit4 of controlword) -> PositioningMove
POSITIONING_FIRSTTARGET	05	Intermediate target position in case of active backlash compensation.
POSITIONING_ENDTARGET	06	Set target position
HOMING_MOVING1	07	
HOMING_MOVING2	08	
HOMING_SET	09	
HOMING_STEP1	10 (0Ah)	Move manually
HOMING_STEP2	11 (0Bh)	Move manually
HOMING_STEP3	12 (0Ch)	Move manually
HOMING_STEP4	13 (0Dh)	Move manually
QUICK_STOP_ACTIVE_1	15 (0Fh)	Bit [0:3] of controlword results in QuickStop. Target position is set inactive.
QUICK_STOP_ACTIVE_2	16 (10h)	Read position
QUICK_STOP_ACTIVE_3	17 (11h)	Set target = position. Target position is set inactive.
HALT_ACTIVE	18 (12h)	Bit [8] of controlword results in intermediate stop. Target position stays active.
ERROR_DIAGNOSTIC	19 (13h)	In case of an error, any mode is changed to ErrorDiagnostic
COMU	20 (14h)	State for internal communication
WAIT_TIME_UNBLOCKING	29 (1Dh)	State between block and unblocking
MOVE_UNBLOCKING	30 (1Eh)	Unblocking movement by distance s (Free referencing distance, Object 2111-1Eh)

Table 6: Additional statusword content in byte 4...7 of T_PDO1 telegram for transfer of status of digital hardware input. Activation is done in Object Version control (Object 2110h, bit 12). If active, the total T_PDO1 telegram length is 8 byte. Byte 3 never contains information.

Bit	Description
0..15	Not defined
16	Monitor bit for digital input 1
17	Monitor bit for digital input 2
18..31	Not defined

T_PDO2 (Position)

Telegram structure:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
280h+Node-ID	x	xx	xx	xx	xx	xx	xx	xx	xx

COB-ID : 280h + node-ID
DLC : 4 or 8 data byte (activate in Object 2110h Version control)
Byte 0…3 : actual position in user defined unit (e.g. micron, see Object 6064h)
Byte 4…7 : contains the actual current value in mA (Object 6078h Current actual value) or actual velocity (Object 606Ch Velocity actual value), if activated in Object 2110h bit 13 or 14 respectively.

2.2.6 Emergency service

In case of a device or bus error, the drive sends an emergency message (EMCY message):

COB-ID	DLC	Byte0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80h+Node-ID	8	Error code		Error register 1001h	Error code additional information		Not used		Not used

For error code and error register definition, see chapter 5 error diagnosis.

2.2.7 Network management and NMT state machine

After initialization, drive is in Pre-Operational mode (see Figure 1). In this mode, SDO can be received or sent. In order to start PDO communication, master must send the NMT command “Start Remote Node”. Then, the drive is in Operational mode. Now, the desired PDO are sent. Additionally, SDO can be received and sent.

After stopping with NMT command “Stop Remote Node”, the drive is in Stopped/Prepared mode. In this mode, no SDO or PDO communication is possible any more. Only NMT communication, LSS, Heartbeat or Node Guarding is possible then. The drive can be initialized by the NMT command “Reset Remote Node”. Afterwards, the drive is again in Pre-Operational mode.

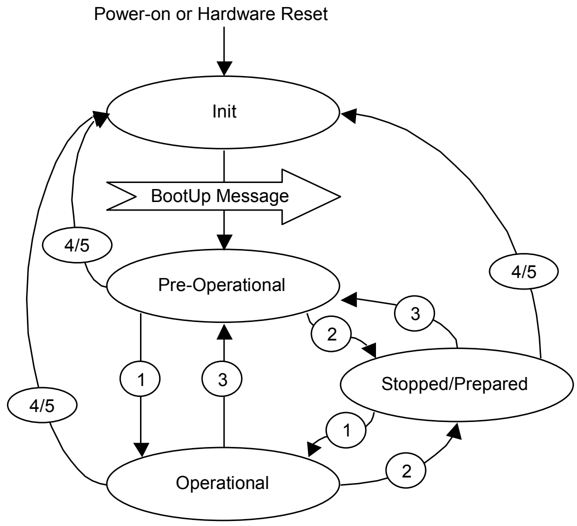


Figure 1: NMT state machine.

By using the network management (NMT) services, bus participants can be initialized, started or stopped. In addition, NMT services for communication supervision exist.

Description of NMT commands

All NMT commands are transmitted as unconfirmed objects and have a structure as follows:

Byte 0	Byte 1	Byte 2
COB-ID = 0	Command byte	Node-ID

The COB-ID for NMT commands is always zero for maximum priority. The node-ID of the desired bus participant is transmitted in byte 2 of the NMT command. By using node-ID zero, all bus participants are desired (Broadcast message).

Table 7: NMT command byte.

Command byte	Description	Jump in NMT state machine (see Figure 1)
01h	Start Remote Node	1
02h	Stop Remote Node	2
80h	Enter Pre-Operational Mode	3
81h, 82h	Reset Remote Node	4, 5

NMT state change

Init

After initialization, the drive appears on to the CAN bus with a BootUp message. Then, the drive changes automatically into pre-operational mode. The BootUp message has the following structure:

The COB-ID of the BootUp Messages is 700h plus the Node-ID:

COB-ID	Byte 0
700h+Node-ID	00h

Start remote node (1)

With the start command, the drive is switched to operational mode.

COB-ID	Command Byte	Node-ID
0	1h	0..127

Stop remote node (2)

With the stop command, the drive is switched to Stopped/Prepared mode.

COB-ID	Command Byte	Node-ID
0	2h	0..127

Enter Pre-Operational mode (3)

Change to Pre-Operational mode.

COB-ID	Command Byte	Node-ID
0	80h	0..127

Reset remote node (4) or reset communication (5)

With the reset command, the drive is re-initialized.

COB-ID	Command Byte	Node-ID
0	81h	0..127

2.2.8 Node guarding and life guarding

The node guarding protocol is used to monitor NMT slaves by the NMT master. Node guarding is activated by setting bit 5 in Object Version control (2110h). After node guarding activation, the NMT slave appears in the database of the NMT master which contains the NMT states of all present NMT slaves. By node guarding protocol, it is checked, if all NMT slaves work correctly or if a NMT slave has stopped communication. At the same time, every NMT slave can check, if the NT master is still active.

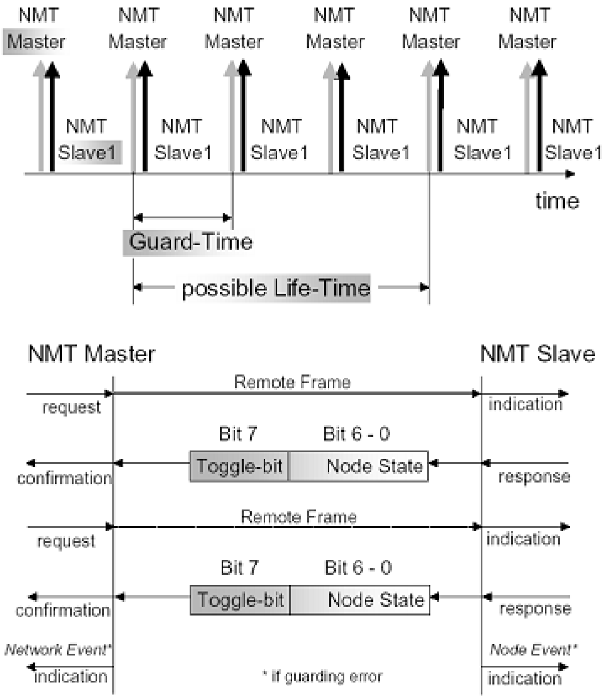


Figure 2: Node guarding and life guarding.

The NMT master starts the monitoring service by cyclic sending of remote frames to the NMT slave. The time interval between remote frames is called guard time (see Figure 2 and Object 100Ch). The guard time can be different for each individual NMT slave. The NMT slave response contains the individual NMT slave status. In this way, the NMT master can check, if the NMT slave is in the correct NMT mode and can handle an occurring error.
The node life time of an NMT slave is given by the guard time multiplied by the lifetime factor (Object 100Dh). The node life time can be different for each individual NMT slave. If an NMT slave does not receive a remote frame during the node life time, a life guarding event is triggered. The error handling in case of a life guarding event is defined in Object Error behaviour - Communication error (Object 1029h, subindex 01h).
According to CAN user organization CiA, monitoring protocol heartbeat should be preferred instead of node guarding in order to reduce bus load.

Example for node guarding protocol:

COB-ID	Data / Remote	Byte 0
700h+Node-ID	r	00h (0d)
700h+Node-ID	d	FFh (255d)
700h+Node-ID	r	00h (0d)
700h+Node-ID	d	7Fh (127d)

Possible states for NMT participants:

0:	BootUp event
4:	Stopped/Prepared
5:	Operational
127:	Pre-Operational

The lower 7 bits correspond to 127d or 7Fh. This means, that the drive is in Pre-Operational mode.

2.2.9 Heartbeat protocol

For new applications, it is strongly recommended to use the heartbeat monitoring protocol. Heartbeat is activated in Object Version control (2110h) by setting bit5 to zero. Then, a heartbeat producer sends a cyclic heartbeat message (see Figure 3). One or more heartbeat consumer can receive this heartbeat message. If one of the cyclic heartbeat messages is missing, a heartbeat event is triggered. The error handling in case of a heartbeat event is defined in Object Error behaviour - Communication error (Object 1029h, subindex 01h). As an example, the slave (drive) can be defined as consumer, which stops in case of a missing heartbeat producer message from the master (PLC). At the same time, the master monitors the slave either via heartbeat or via SDO responses.

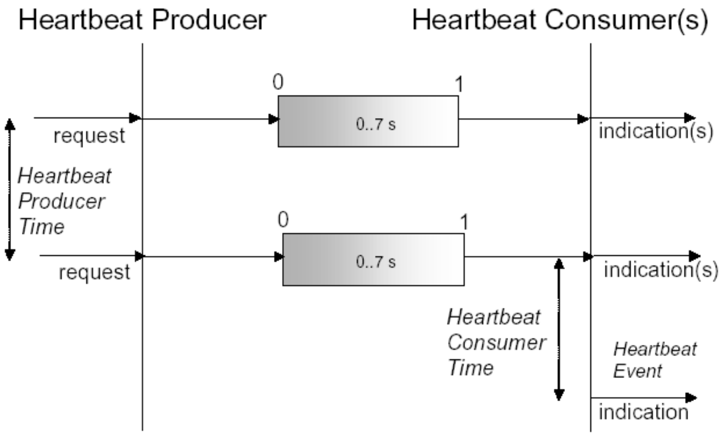


Figure 3: Heartbeat protocol.

Heartbeat protocol example:

COB-ID	Data/Remote	Byte 0
700h+Node-ID	d	7Fh (127d)

Heartbeat messages consist of COB-ID and a byte to transmit the NMT state:

0:	BootUp message
4:	Stopped/Prepared mode
5:	Operational mode
127:	Pre-Operational, i.e. drive is in Pre-Operational mode (7Fh = 127).

2.2.10 Layer Setting Services (LSS)

LSS is used to connect to bus participants which are on the same bus with the same node-ID. (see procedures in Layer Setting Services and Protocol, CiA Draft Standard Proposal 305).
Every bus participant with LSS has a unique serial number for addressing and initialization. Subsequently, node-ID and baudrate can be set to desired values. LSS can be carried out only in Stopped/Prepared mode.

Telegram structure

COB-ID

Master → drive : 2021 = 7E5h
Master ← drive : 2020 = 7E4h
After the COB-ID follow the LSS Command Specifier and up to seven data byte:

COB-ID	Command Specifier	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6

Switch mode global

7E5h →	04h	Mode	reserved

Mode: : 0 → Operating mode
1 → Configuration mode

Switch mode selective

By using the following procedure, a specific bus participant can be addressed:

7E5h →	40h	Vendor-ID	reserved

7E5h →	41h	Product code	reserved

7E5h →	42h	Revision number	reserved

7E5h →	43h	Serial number	reserved

7E4h ←	44h	Mode	reserved

Vendor-ID : 5Fh
Product code : product code of bus participant
Revision number : revision number of bus participant
Serial number : serial number of bus participant
Mode : Answer of bus participant is new mode (0=Operating mode; 1= Configuration mode)

Set node-ID

7E5h →	11h	Node-ID	reserved

7E4h ←	11h	Error code	Specific error	reserved

Node-ID : new node-ID of bus participant
Error code : 0=OK; 1=node-ID outside of allowed range; 2..254=reserved; 255=specific error
Specific error : if error code=255 → application specific error code

Set baudrate

7E5h →	13h	Table Sel	Table Ind	reserved

7E4h ←	13h	Error code	Specific error	reserved

Table Sel : select baudrate table (default = 0: standard CiA baudrate table)
Table Ind : baudrate index in baudrate table (see Table 8).
Error code : 0=OK; 1= baudrate outside of allowed range; 2..254=reserved; 255=specific error
Specific error : if error code=255 → application specific error code

Table 8: Standard CiA baudrate table.

Baudrate [kBit/s]	10	20	50	100	125	250	500	800	1000
Index (LSS) according CiA-Table	8h	7h	6h	5h	4h	3h	2h	1h	0h
Index (definition in Object 2100h)	0h	1h	2h	3h	4h	5h	6h	7h	8h

Alternatively, the baudrate can be changed in Object 2100h. This object contains as well an index corresponding to the baudrate and not eh baudrate itself.
NOTE: Index for Object 2100h and for LSS are different.

Save configuration parameters

By using the following procedure, the configuration parameters are saved to EEPROM:

7E5h →	17h	reserved

7E4h ←	17h	Error code	Specific error	reserved

Error code : 0=OK;1=save not supported; 2=access error;3..254=reserved;255=specific error
Specific error : if error code=255 → application specific error code

Activate new Bit Timing

The new Bit Timing parameters are activated by Command Specifier 21:

7E5h →	15h	16 Bit Switch Delay	reserved

Switch delay : time delay of drive reset in milliseconds,
after which bus participant responds with new baudrate.

Request vendor-ID

Request vendor-ID of a selected bus participant:

7E5h →	5Ah	reserved

7E4h ←	5Ah	32 Bit Vendor-ID	reserved

Vendor-ID : = 5Fh

Request product code

Request product code of a selected bus participant:

7E5h →	5Bh	reserved

7E4h ←	5Bh	Product code	reserved

Product code : Manufacturer specific product code

Request revision number

Request revision number of a selected bus participant:

7E5h →	5Ch	reserved

7E4h ←	5Ch	32 Bit Revision number	reserved

Revision number : actual revision number

Request serial number

Request serial number of a selected bus participant:

7E5h →	5Dh	reserved

7E4h ←	5Dh	32 Bit Serial number	reserved

Serial number : actual serial number

Request parameter range

In order to search for bus participants with parameters within a specific range, send the following objects:

7E5h →	46h	Vendor-ID	reserved

7E5h →	47h	Product code	reserved

7E5h →	48h	Revision number LOW	reserved
7E5h →	49h	Revision number HIGH	reserved

7E5h →	4Ah	Serial number LOW	reserved
7E5h →	4Bh	Serial number HIGH	reserved

Every bus participants with the corresponding parameters responds with the following message:

7E4h ←	4Fh	reserved

2.3 State machine for positioning / homing

During drive start-up, the NMT state machine is activated. In the NMT states Pre-Operational and Operational, the following state machine is valid for Positioning mode and Homing mode (see Object 6060h).

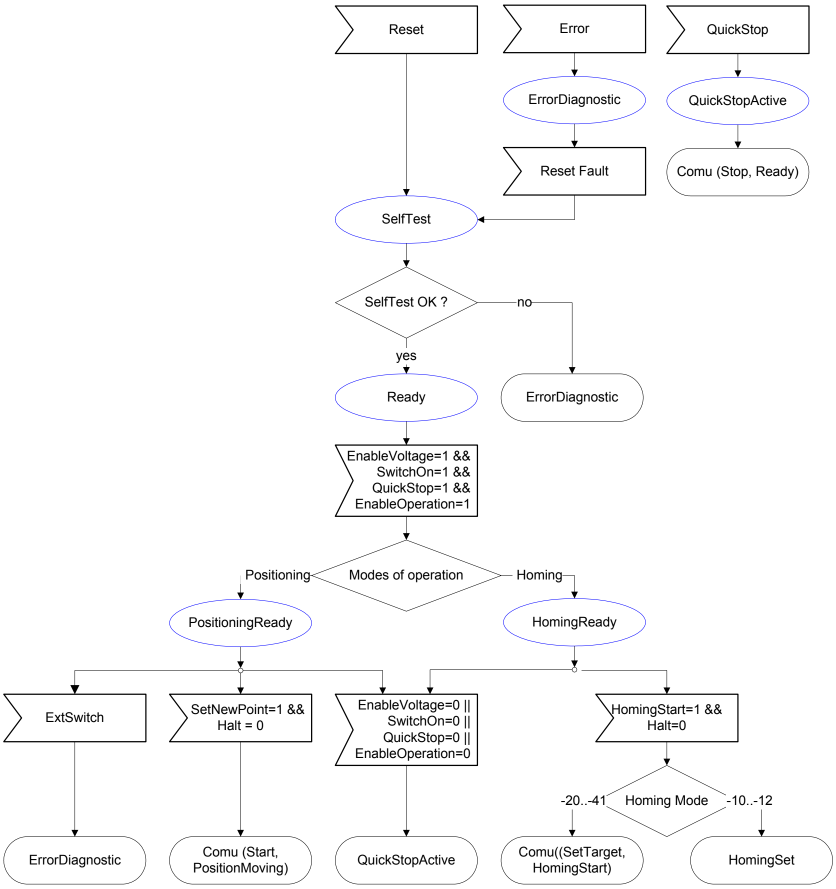


Figure 4: State machine for Positioning mode and Homing mode.

Selftest

After drive reset, a selftest is carried out. During this selftest, a reset is sent to the regulator. Afterwards, all regulator parameters are checked for correctness and plausibility. After successful selftest, the drive switches to Ready mode.

Ready mode

The Ready mode is the basic drive mode. This mode is changed only on master request. In Ready mode, all parameters can be read or set.

PositioningReady mode

Change from Ready mode to PositioningReady mode by:

1. Set Object 6060h (Modes of operation) to 1 = Positioning
2. Set Object 6040h (Controlword) to 000Fh
   SwitchOn, EnableVoltage, QuickStop and EnableOperation = 1

In PositioningReady mode, the drive is ready to carry out positioning tasks and waits only for a start signal via Controlword. QuickStop and triggering of digital hardware inputs is processed separately.

HomingReady mode

Change from Ready mode to HomingReady mode by:

1. Set Object 6060h (Modes of operation) to 6 = Homing
2. Set Object 6040h (Controlword) to 000Fh
   SwitchOn, EnableVoltage, QuickStop and EnableOperation = 1

In HomingReady mode, the Homing method is selected in Object 6098h (Homing method). By Controlword, Bit 4 (HomingOperationStart), the drive is changed to HomingMove mode.

QuickStop active

QuickStop is no static mode. If the drive changes to QuickStop, an internal QuickStop command is immediately sent to the regulator. Afterwards, the drive changes to Ready Mode. If an error is present, the drive subsequently changes to ErrorDiagnostic mode.

ErrorDiagnostic

In ErrorDiagnostic mode, the drive sends an EMCY message (Error code see Table 17) and waits for further instructions from CAN master. In this mode, all parameters can be read or set. After Error reset (Controlword), the drive changes to Selftest mode. In this mode, the regulator receives an error reset as well.

PositioningMoving

The regulator always has the actual target position. Therefore, the start command is sent and the drive changes to PositioningMoving mode.

Events:

1.	Stop command via CAN	-> send QuickStop to regulator
2.	Drive not moving	-> target position reached
3.	External hardware limit switch active	-> send QuickStop to regulator

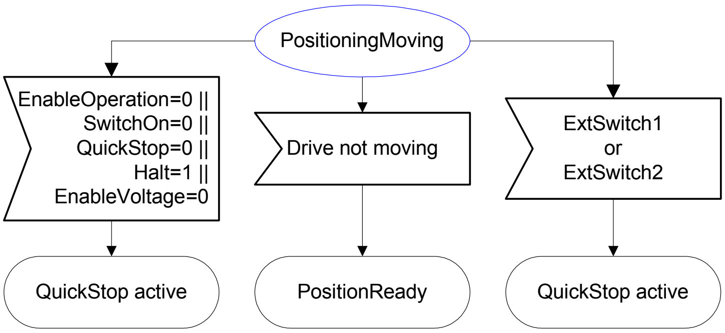


Figure 5: PositioningMoving Mode.

Homing

For Homing, two procedures are possible. In HomingSet, only position values can be set and stored without drive movement. In HomingMoving, the drive can move.

HomingMoving

New target position is set. Afterwards, the start command is sent to the regulator and the drive changes to HomingMoving Mode.

Events:

1.	Stop command via CAN	-> send QuickStop to regulator
2.	Drive not moving	-> target position reached
3.	External hardware limit switch active	-> send QuickStop to regulator

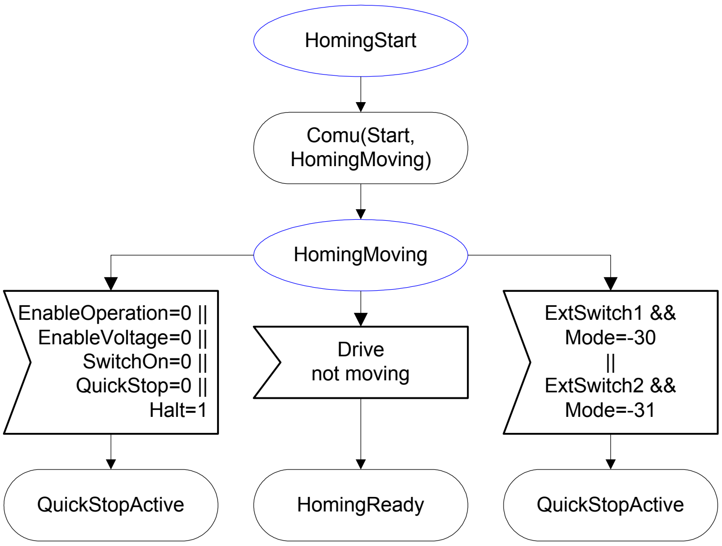

HomingStart
Comu(Start, HomingMoving)
HomingMoving
EnableOperation=0 || EnableVoltage=0 || SwitchOn=0 || QuickStop=0 || Halt=1
Drive not moving
ExtSwitch1 && Mode=-30 || ExtSwitch2 && Mode=-31
QuickStopActive
HomingReady
QuickStopActive

Figure 6: HomingMoving.

HomingSet

In HomingSet Mode, only parameters are set (e.g. position of software limit switches).

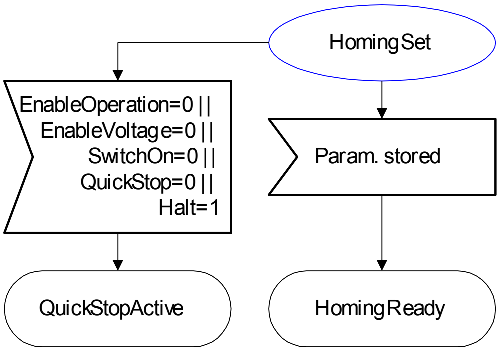

HomingSet
EnableOperation=0 || EnableVoltage=0 || SwitchOn=0 || QuickStop=0 || Halt=1
Param. stored
QuickStopActive
HomingReady

Figure 7: HomingSet.

Read status or position

In all drive modes, status or actual position is updated every 1 ms.

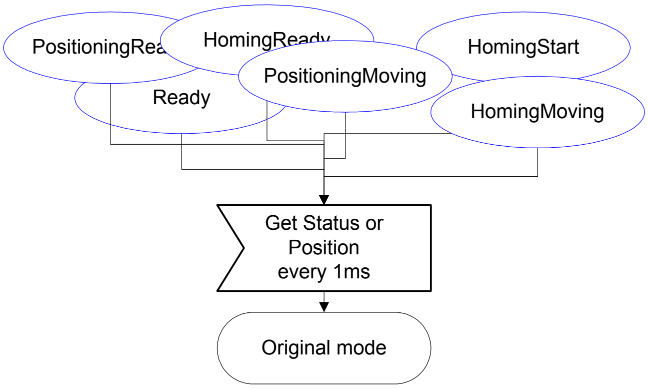


Figure 8: Read status or actual position.

Controlword

Mode changes are controlled via controlword.
The master sends the following controlword to activate the drive: **xxxx xxx0 0xxx 1111**
The master sends the following controlword to start the drive: **xxxx xxx0 0xx1 1111**

Table 9: Controlword (Object 6040h). Bit 13...15 not defined.

Bit	Description
0	SwitchOn
1	EnableVoltage
2	QuickStop
3	EnableOperation
4	StartAction (move to target)
5	Change set immediately
6	Relative/absolute movement
7	Error reset
8	Halt
11	Jogging+
12	Jogging-

Examples for mode changes:

Action	Controlword it					Description
	7	3	2	1	0	
SwitchOn	0	**1**	**1**	**1**	**1**	Ready → PositioningReady or HomingReady (depends on Object 6060h)
DisableOperation	0	**0**	1	1	1	PositioningReady or HomingReady → Ready
DisableVoltage	0	1	1	**0**	1	PositioningReady or HomingReady → Ready
SwitchOff	0	1	1	1	**0**	PositioningReady or HomingReady → Ready
QuickStop	0	**1**	**0**	1	1	PositioningReady or HomingReady → QuickStop active If drive is moving, controlword command is processed separately. It can be changed directly to OperationEnabled, if QuickStop bit set back to 1.

2.4 Drive operation

2.4.1 Read out drive status

Depending on the NMT mode, the drive status can be read out via SDO (Object 6041h) or can be requested cyclic or synchronous via T_PDO1.

COB-ID	DLC	Byte 0	Byte 1	Byte 2
180h+Node-ID	3	xx	xx	xx

2.4.2 Read out actual position

The effective position can be read out via SDO or can be requested cyclic or synchronous via T_PDO2.

Master request for actual drive position (Object 6064h):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	40h	64h	60h	0	0	0	0	0

Drive response with actual position (abcd):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	43h	64h	60h	0	a	b	c	d

As an alternative, it is possible to transfer position information, e.g. in case of a position change, by process data objects (T_PDO). By default, T_PDO2 is set to position (Object 6064h). Here, synchronous and asynchronous T_PDO communication can be selected (Object 1400h).

T_PDO2 telegram structure

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3
280h+Node-ID	4	Ww	Xx	Yy	Zz

Byte1…4 contains position

2.4.3 Move to target position by SDO (with positioning task)

Set target position (Object 607Ah):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Ah	60h	0	x	x	x	x

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Ah	60h	0	0	0	0	0

Write start into controlword (Object 6040h):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	1Fh	0	0	0

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

The drive starts to move.

2.4.4 Stop positioning task

Write stop into controlword (Object 6040h):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	1Fh	01h	0	0

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

The drive stops.

2.4.5 Move to target position by R_PDO (with positioning task)

NMT start:

COB-ID	DLC	Byte 0	Byte 1
0	2	01h	Node-ID

Response T_PDO1 (statusword Object 6041h):

COB-ID	DLC	Byte 0	Byte 1	Byte 2
180h+Node-ID	3	21h	0Ah	01h

Response T_PDO2 (actual position Object 6064h):

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3
280h+Node-ID	4	Ww	Xx	Yy	Zz

Set drive to PositioningReady Mode by R_PDO1 (controlword Object 6040h):

COB-ID	DLC	Byte 0	Byte 1
200h+Node-ID	2	0Fh	00h

Response T_PDO1 (statusword Object 6041h):

COB-ID	DLC	Byte 0	Byte 1	Byte 2
180h+Node-ID	3	37h	0Ah	02h

Set controlword and target position by R_PDO2 (Object 6040h, 607Ah):

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
300h+Node-ID	6	1Fh	00h	x	x	x	x

Send only SYNC telegram (if R_PDO2 is set to synchronous communication)

COB-ID	DLC
80h	0

The drive starts to move.

2.5 Command synchronization between master and drive

For CAN bus communication, the following demands apply:

- Minimum bus load (therefore, cyclic communication should be avoided, except Nodeguarding).
- Fast execution of start commands in order to achieve short cycle times.

In case of CANopen, a positioning task is started by sending a controlword (R_PDO or SDO, Object 6040h) from the master to the drive. For transfer of status information, T_PDO is used (statusword, Object 6041h). The statusword contains all drive status information (drive moving, target position reached, etc.). During a positioning task, master and drive must be synchronized by using the statusword. For minimum bus load, the drive sends statusword only in case of changes.

For command synchronization, the following is implemented in the drive:

- A T_PDO is sent after every status change.
- Every start command creates a status change.

Both topics are explained in the following. For command synchronization between master and drive, it is important, that any communication sequence is correct and complete.

A T_PDO is sent after every status change

The drive has an adjustable cycle with cycle time T (transmit PDO2 event timer, Object 1801h-05h) for sending of T_PDO. Usually, the drive sends the actual status which is valid at PDO sending time (Object 6041h). However, it is important that the drive sends a T_PDO, even if the drive status is changed for a short moment within a cycle and this change disappears before the next T_PDO cycle (see Figure 9). In order to allow efficient master-drive command synchronization, this implemented. If no status change has occurred, no T_PDO is sent.

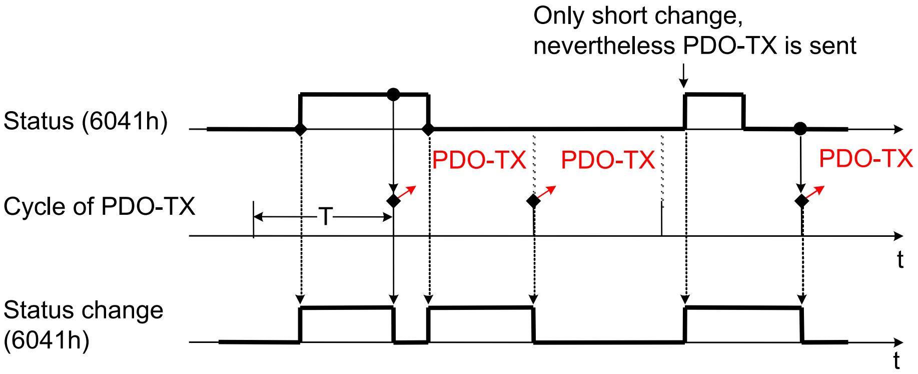


Figure 9: A T_PDO is sent after every status change, even after short changes.

Every start command creates a status change

If the master sends a positioning task to a drive with a target position far away from the actual drive position, a clear status change occurs and the status bit 10 „Target position reached“ is set subsequently after reaching target position.
However, if target position is identical with actual position, no status change would occur. This could lead to an incorrect master-drive synchronization. In order to ensure a correct command synchronization in such cases, the drive status bit 10 „Target position“ is forced for a time Tmin to zero after receiving a start command. This is true for all Homing procedures as well. Here, the respective status bits are set for Tmin to zero and after finishing set to one again. In this way, it is ensured, that for all commands a status change is generated and a corresponding T_PDO is sent.

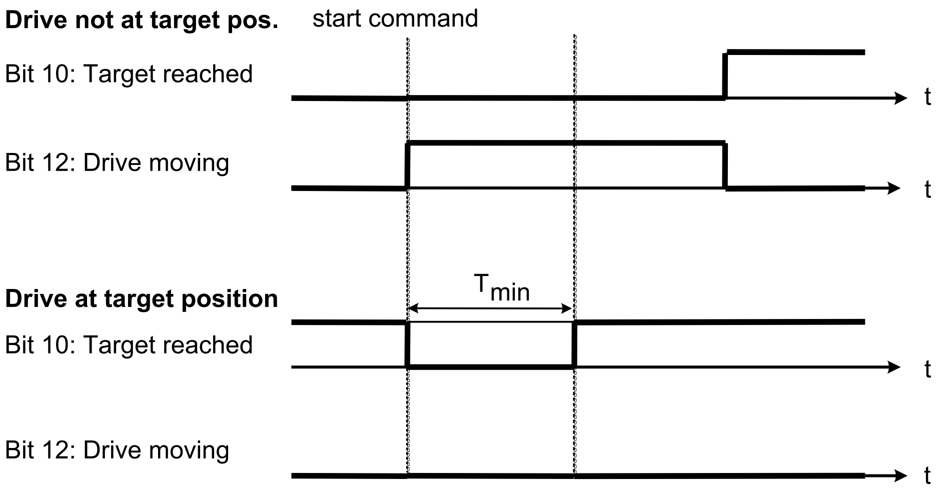


Figure 10: Every start command creates a status change (applies as well for Homing).

Correct communication sequence with up to two SDO requests of statusword 6041h

An undefined situation can occur, if the master sends a start command R_PDO by accident at the same time as a drive status change occurs with corresponding sending of a T_PDO (Figure 11). In this case, the master receives an invalid drive status which is from the past and does not correspond to the active master command and therefore must be ignored.
In order to suppress unwanted T_PDO messages, they must be blocked as soon as the master sends a start command (Ignore T_PDO = 1). Subsequently, the master asks for the drive status by 1. SDO request. As soon as the master receives the 1. SDO response, the message blocking is cancelled (Ignore T_PDO = 0). Then, all received T_PDO messages are evaluated again. If the 1. SDO response contains the status „Target position reached = 1“, the position task is correctly finished. If a T_PDO message which is received after cancelling of the message blocking contains the status „Target position reached = 1“,the position task is correctly finished as well.
However, a T_PDO message with high priority can overtake 1. SDO response and therefore will be blocked. (see Figure 11). In this case, the position task can be only correctly finished by a second SDO request with subsequent SDO response. The above applies as well for all Homing procedures.

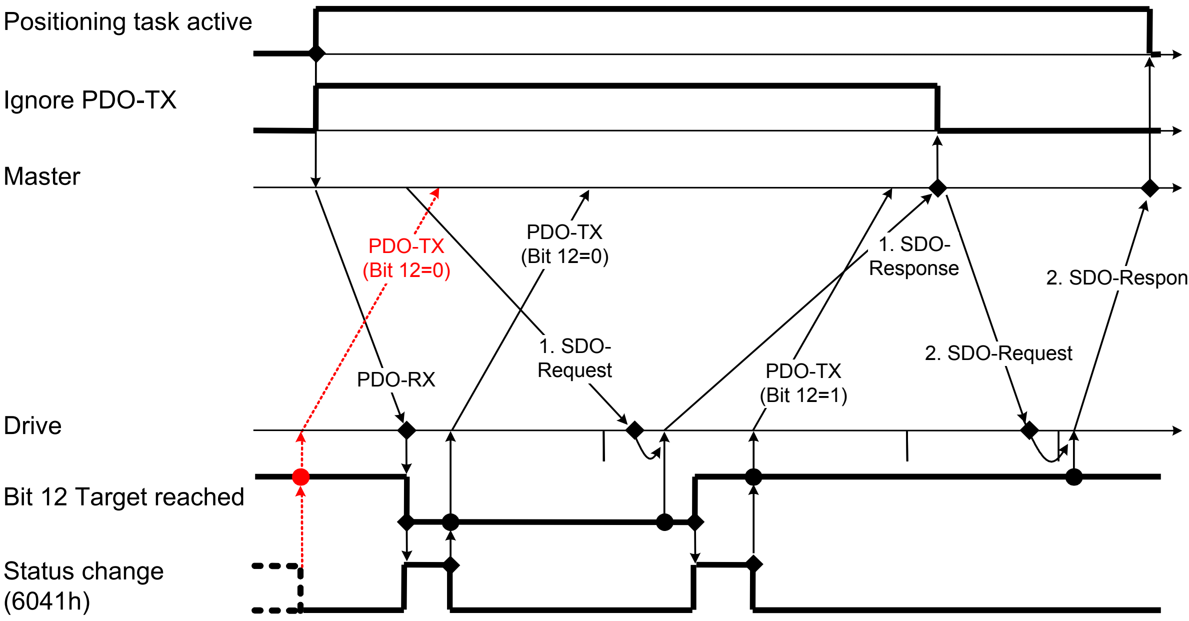


Figure 11: Correct communication sequence with up to two SDO requests of statusword 6041h.

3 Object description

According to CiA (CAN in Automation), three object classes exist:

- **Standard objects**
  1000h, 1001h, 1018h
- **Manufacturer specific objects**
  2000h - 5FFFh
- **Device specific objects**
  All other objects in 1000h - 1FFFh, 6000h - FFFFh

If the CAN master sends an invalid command to the drive, the drive sends as response an Abort Message with Abort Code (see Table 18). Any change in objects with write access can be saved non-volatile in drive EEPROM. After power-on of drive, all parameters are loaded automatically from this EEPROM.

Table 10: Summary of all supported objects. Save actual object configuration by using Object 1010h. Load default parameters by using Object 1011h. For further details, see chapter 4.2.

Object	Object number in hexadecimal (example 1000h)
Name	---
Format	Variable format of object: U/I = unsigned / integer, number=number of bits, ARR = array, REC = record
Access	ro = read only, wo = write only, rw = read / write
Default	Default value of parameter after first initialization or after load default parameter command (Object 1011h)
Save	Save or load of parameters (EEPROM) by objects 1010h and 1011h, subindex 1, 2, 3 or 4
Description	Description of object options

Object	Name	Format	Access	Default	Save	Description
1000h	Device type	U32	ro	00020192h 00h 02h 0192h	-	Device type according to CiA Motor type = 0 (DC Motor) Drive type =2 (ServoDrive) Profile number = 192h = 402
1001h	Error register	U8	ro	0h	-	Error register which contains class of active error (Table 17). Bit 2=1 Voltage error Bit 3=1 Temperature error Bit 4=1 CAN bus communication error Bit 5=1 Device specific error
1003h	Predefined error field	ARR			-	Shift register of last 8 specific errors or warnings
00h	Number of Errors	U8	rw	0h	-	-
01h	Last entry	U32	ro		-	Last occurred error (see Table 17 or Object 603Fh). Error code / Description 0000h No error 2310h Continuous over current 3110h Error over voltage bus 3111h Error over voltage power electronics 3120h Error under voltage bus 3121h Error under voltage power electronics 4210h Warning / error over temperature (see additional information byte 3…4 in EMCY message and Table 17) 5441h Warning min. hardware limit switch active 5442h Warning max. hardware limit switch active 5530h Warning memory (EEPROM) 6010h Warning software (Watchdog) 7121h Warning motor blocked (see additional information byte 3…4 in EMCY message and table 17)

Object	Name	Format	Access	Default	Save	Description
						7320h Error encoder 7510h Error internal communication 8110h Warning CAN bus communication 8130h Warning lifeguard or heartbeat 8500h Warning Position control FF00h Warning Data valid multiturn (DVMT) FF02h Error positioning timeout FF05h Warning deblocking active FF06h Error deblocking timeout FF10h Warning minimum software limit switch active FF11h Warning maximum software limit switch active
...	...	...	...	...	-	...
08h	First entry	U32	ro		-	See subindex 01h description
1005h	SYNC COB-ID	U32	rw	80h	1, 2	COB-ID of SYNC object
1008h	Device name	U32	ro	MSBA	-	Drive name in ASCII (example: MSBA = „ABSM“ = 41h 42h 53h 4Dh)
1009h	-	U32	ro	factory	-	Not assigned
100Ah	Software version	U32	ro	factory	-	Subversion of overall firmware version of drive in ASCII. This subversion corresponds to correction of software errors. For overall firmware version, see Object 1018h-3h. Example: 0017 = 17
100Ch	Guard time	U16	rw	0h	1, 2	Guard time for node guarding (see Figure 2).
100Dh	Life time factor	U8	rw	0h	1, 2	Life time factor for node guarding (see Figure 2). The node lifetime is given by guard time multiplied with life time factor. In case of an error, e.g. after elapse of node lifetime, a life guarding event is activated. The error behaviour is defined in Object Error behaviour (Object 1029h). NOTE: Activate node guarding protocol in Object Version control (Object 2110h, bit 5).
1010h	Store parameters	ARR				Save actual drive parameter. Column Save of this table gives for each Object the required subindex for save parameter operation
00h	Largest subindex	U8	ro	4h		-
01h	Store all parameters	U32	rw			Save all drive parameters to EEPROM by writing save = „evas“ = 65h 76h 61h 73h
02h	Store communication parameters	U32	rw			Save communication parameters to EEPROM by writing save = „evas“ = 65h 76h 61h 73h
03h	Store application parameters	U32	rw			Save application parameters to EEPROM by writing save = „evas“ = 65h 76h 61h 73h
04h	Store manufacturer specific parameters	U32	rw			Save manufacturer specific parameters to EEPROM by writing save = „evas“ = 65h 76h 61h 73h
1011h	Restore parameters	ARR				Load default drive parameter. Column Save of this table gives for each object the required subindex for load default parameter operation.
00h	Largest subindex	U8	ro	4h		-
01h	Restore all parameters	U32	rw			Load all drive parameters from EEPROM by writing load = „daol“ = 64h 61h 6Fh 6Ch
02h	Restore communication parameters	U32	rw			Load communication parameters from EEPROM by writing load = „daol“ = 64h 61h 6Fh 6Ch

Object	Name	Format	Access	Default	Save	Description
03h	Restore application parameters	U32	rw			Load application parameters from EEPROM by writing load = „daol“ = 64h 61h 6Fh 6Ch
04h	Restore manufacturer specific parameters	U32	rw			Load manufacturer specific parameters to EEPROM by writing load = „daol“ = 64h 61h 6Fh 6Ch
1014h	Emergency COB-ID	U32	rw	81h	1, 2	COB-ID of Emergency object EMCY COB-ID = 80h + Node-ID
1016h	Consumer heartbeat	ARR				
00h	Largest subindex	U8	ro	1h		-
01h	Consumer heartbeat time	U32	rw	00010000h 0001h 0000h	1, 2	Consumer heartbeat time interval (see Figure 3). Bit16..23: Node- ID Bit0..15: Consumer heartbeat time interval in milliseconds NOTE: Activate heartbeat protocol in Object Version control (Object 2110h, bit 5).
1017h	Producer heartbeat time	U16	rw	0h	1, 2	Producer heartbeat time interval in milliseconds
1018h	Identity object	REC				Manufacturer and device identification for LSS
00h	Largest subindex	U8	ro	4h		-
01h	Vendor-ID	U32	ro	5Fh		Number for manufacturer identification according to CiA. (Baumer Electric AG = 5Fh)
02h	Product code	U32	ro	00000102h		Number for identification of drive (MSBA = 102h)
03h	Revision number	U32	ro	factory		Firmware version number of drive (overall firmware version). Example: Firmware version 1.00 = 0001h’0000h. Main index and subindex correspond to drive functions (1.00.xx). The last index (xx) corresponds to a correction of software errors and is saved in Object 100Ah Software version. This firmware version number is used for LSS identification of drive and must be identical with manual version and with revision number (RevNr) printed on drive label.
04h	Serial number	U32	ro	factory		Unique drive serial number
1029h	Error behaviour	ARR				Error handling of NMT communication errors.
00h	Largest subindex	U8	ro	1h		-
01h	Communication error	U8	rw	1h	1, 2	Drive behaviour after communication error: 0h = Change to Pre-Operational mode 1h = No mode change 2h = Change to Stopped/Prepared mode 3h = Reset of drive
1400h	Receive PDO1 parameter	REC				
00h	Largest subindex	U8	ro	2h		-
01h	COB-ID	U32	rw	201h	1, 2	PDO-ID = 200h + Node-ID
02h	R_PDO1 type	U8	rw	FEh	1, 2	1h..F0h = synchronous operation. With value n the R_PDO1 is received with every n-th SYNC telegram. FEh = cyclic (asynchronous) operation. R_PDO1 are received immediately.
1401h	Receive PDO2 parameter	REC				

Object	Name	Format	Access	Default	Save	Description
00h	Largest subindex	U8	ro	2h		-
01h	COB-ID	U32	rw	301h	1, 2	PDO-ID = 300h + Node-ID
02h	R_PDO2 type	U8	rw	FEh	1, 2	1h..F0h = synchronous operation. With value n the R_PDO2 is received with every n-th SYNC telegram. FEh = cyclic (asynchronous) operation. R_PDO2 are received immediately.
1600h	Receive PDO1 mapping	ARR				
00h	Number of mapped objects in R_PDO1	U8	ro	1h		Number of mapped objects in R_PDO1
01h	R_PDO1 content 1	U32	ro	60400010h		Control word. Read only (CiA recommendation is read / write)
1601h	Receive PDO2 mapping	ARR				
00h	Number of mapped objects in R_PDO2	U8	ro	2h		Number of mapped objects in R_PDO2
01h	R_PDO2 content 1	U32	ro	60400010h		Control word. Read only (CiA recommendation is read / write)
02h	R_PDO2 content 2	U32	ro	607A0020h		Target position. Read only (CiA recommendation is read / write)
1800h	Transmit PDO1 parameter	REC				
00h	Largest subindex	U8	ro	5h		-
01h	COB-ID	U32	rw	181h	1, 2	PDO-ID = 180h + Node-ID
02h	Transmission type	U8	rw	FEh	1, 2	1h..F0h = synchronous operation. With value n the T_PDO is sent on every n-th SYNC telegram, depending on object 2800h PDO1 add-on. FEh = asynchronous operation (manufacturer specific). T_PDO1 are sent with cycle time defined in sub-index 05h, depending on object 2800h PDO1 add-on. FFh = asynchronous operation (device profile specific). T_PDO1 are sent after a change of data and with respect to sub-index 03h Inhibit time and cycle time sub-index 05h Event timer (also without change). Object 2800h PDO1 add-on has no influence
03h	Inhibit time	U16	rw	0h	1, 2	Wait time after sending of a T_PDO1 in 100 micro seconds. Waiting time is active if transmission type (sub-index 02h) is FFh. <0Ah = no wait time >=0Ah = wait time >= 1 ms (use multiple values of 0Ah)
05h	Event timer	U16	rw	203h	1, 2	0h = Transmit PDO1 are not sent (Transmission Type FEh) or only after change (Transmission Type FFh) >0h = Cycle time in milli seconds
1801h	Transmit PDO2 Parameter	REC				
00h	Largest subindex	U8	ro	5h		-
01h	COB-ID	U32	rw	281h	1, 2	PDO-ID = 280h + Node-ID

Object	Name	Format	Access	Default	Save	Description
02h	Transmission type	U8	rw	FEh	1, 2	1h..F0h = synchronous operation. With value n the T_PDO is sent on every n-th SYNC telegram, depending on object 2801h PDO2 add-on. FEh = asynchronous operation (manufacturer specific). T_PDO2 are sent with cycle time defined in sub-index 05h, depending on object 2801h PDO2 add-on. FFh = asynchronous operation (device profile specific). T_PDO2 are sent after a change of data and with respect to sub-index 03h Inhibit time and cycle time sub-index 05h Event timer (also without change). Object 2801h PDO2 add-on has no influence
03h	Inhibit time	U16	rw	0h	1, 2	Wait time after sending of a T_PDO2 in 100 micro seconds. Waiting time is active if transmission type (sub-index 02h) is FFh. <0Ah = no wait time >=0Ah = wait time >= 1 ms (use multiple values of 0Ah)
05h	Event timer	U16	rw	104h	1, 2	0h = Transmit PDO2 are not sent (Transmission Type FEh) or only after change (Transmission Type FFh) >0h = Cycle time in milli seconds
1A00h	Transmit PDO1 mapping	ARR				
00h	Number of mapped objects in T_PDO1	U8	ro	2h		Number of mapped objects in T_PDO1
01h	T_PDO1 content 1	U32	ro	60410010h		Read only (CiA recommendation is read / write)
02h	T_PDO1 content 2	U32	ro	60FD0020h		Read only (CiA recommendation is read / write)
1A01h	Transmit PDO2 mapping	ARR				
00h	Number of mapped objects in T_PDO2	U8	ro	1h		Number of mapped objects in T_PDO2
01h	T_PDO2 content 1	U32	ro	60640020h		Read only (CiA recommendation is read / write)
2100h	Baudrate	U8	rw	2h	1, 2	Drive baudrate for bus communication (see Table 12). 01h = 20 kBit/s 02h = 50 kBit/s 03h = 100 kBit/s 04h = 125 kBit/s 05h = 250 kBit/s 06h = 500 kBit/s 07h = 800 kBit/s 08h = 1000 kBit/s NOTE: After setting a new baudrate, drive must be initialized e.g. by NMT reset or by switching drive off and on.
2101h	Node-ID	U8	rw	1h	1, 2	Node-ID (slave address) of drive on CAN bus. Possible values are 1...127. NOTE: After setting a new node-ID, drive must be initialized e.g. by NMT reset or by switching drive off and on.
2102h	Node-ID offset	U8	rw	0h		Offset for drive node-ID. Possible values are 0..126. Node-ID offset is added to node-ID and activated immediately after setting. Node-ID offset cannot saved permanently and is set back to 0h after initialization e.g. by switching drive off and on. To confirm node-ID offset, drive send a download response with new node-ID (see SDO commands in Table 2). NOTE: Node-ID offset cannot be saved.

Object	Name	Format	Access	Default	Save	Description
2110h	Version control	U32	rw	0Ah	1, 3	Bit 1: T_PDO1 standard content 0 = T_PDO1 contains statusword with 2 byte length 1 = T_PDO1 contains statusword and additional statusword content with 3 byte total length (default). Bit 3: Drive behavior in case of BusOFF 0 = drive withdraws itself from bus after BusOFF 1 = automatic drive initialization after BusOFF(default) Bit 5: 0 = Heartbeat (default) 1 = Node guarding Bit 9: Service operation (Jogging possible without CAN bus) 0 = inactive (default) 1 = active Bit 10: Emergency message during Homing 0 = active (default) 1 = inactive Bit 11: Emergency message of software limit switches 0 = active (default) 1 = inactive Bit 12: T_PDO1 additional content: Send hardware input state (Object 60FDh) in addition to T_PDO1 standard content (8 byte total length) 0 = inactive (default) 1 = active Bit 13: T_PDO2 additional content: Send Velocity actual value (Object 606Ch) in addition to T_PDO2 standard content (8 byte total length). Restriction: Either Bit13 or Bit14 can be can be activated. 0 = inactive (default) 1 = active Bit 14: T_PDO2 additional content: Send Current actual value (Object 6078h) in addition to T_PDO2 standard content (8 byte total length). Restriction: Either Bit13 or Bit14 can be can be activated. 0 = inactive (default) 1 = active Bit 18: Automatic unblocking: For description see chapter 4.8 Automatic unblocking. 0 = inactive (default) 1 = active Bit 19: Current limiting. 0 = inactive (default) 1 = active Bit 21: Behaviour when motor blocked. 0 = Motor blocked handled as a Warning (default) 1 = Motor blocked handled as an Error
2111h	Positioning parameter	REC				Parameters for positioning task
00h	Largest subindex	U8	ro	13h		-
01h	Backlash compensation	U8	rw	0h	1, 3	Function to compensate spindle or gear play. The direction for target position approach can be defined. If necessary, the drive goes by backlash delta over the target position and then subsequently moves back directly to the target position. 00h = move directly to target position 01h = move to target position from lower position values 02h = move to target position from higher position values
02h	Backlash delta	U32	rw	64h	1, 3	Parameter in user defined position unit. For backlash compensation, the drive goes by backlash delta over the target position before it moves back directly to the target position.

Object	Name	Format	Access	Default	Save	Description
03h	Positioning timeout	U32	rw	0h	1, 3	Timeout for positioning tasks in seconds: If a positioning task takes more than Positioning timeout, it is terminated and an EMCY message with error code FF02h is activated. 0h = Timeout inactive >0h = Timeout in seconds
0Bh	Time of jogging start speed	U32	rw	3E8h	1, 3	Time interval for jogging start speed while jogging operation in milliseconds
0Ch	Jogging start speed	U32	rw	1Eh	1, 3	Start speed in jogging operation in user defined speed unit (see Object 608Ch).
0Dh	Digital input 1	U8	rw	0h	1, 3	Digital hardware input 1 for drive control (see controlword Object 6040h). Input is HIGH active, signal inverting by addition of 80h. Parameter / description: 0h = Digital input inactive 01h = SwitchOn 02h = EnableVoltage 03h = QuickStop 04h = EnableOperation 05h = StartAction 06h = Change set immediately 07h = Relative/absolute 08h = Error reset 09h = Halt 0Ah = Jogging+ (within software limit switches, if active) 0Bh = Jogging- (within software limit switches, if active) 10h = Status: Operation enabled (01h..04h) 12h = Referencing with preset value (Object 2111h, subindex 11h). After referencing, this parameter is set back to 0h. 13h = Jogging+ (with warning beyond software limit switches) 14h = Jogging- (with warning beyond software limit switches) 15h = maximum external hardware limit switch 16h = minimum external hardware limit switch 17h = not defined NOTE: If parameter >0, the drive is directly controlled by the input. In this case, the master loose partially control on the CAN bus.
0Eh	Digital input 2	U8	rw	0h	1, 3	Digital hardware input 2. Description see Digital input 1.
11h	Reference position	I32	rw	0h	1, 3	Preset value for referencing of drive position. For details about referencing procedures, see chapter 4.3 Referencing position and Object 6098h Homing method. For Homing methods E2h, E1h, D8h, D7h, the reference position overwrites Object 6064h Position actual value during referencing procedure.
13h	Simulation controlword	U16	rw	0h	1, 3	Controlword, which is automatically set, if no bus connection is present (e.g. for stand-alone-operation of drive without fieldbus).
1Dh	Service speed	U32	rw	1Dh	1, 3	Speed in user defined speed unit for service operation of drive without fieldbus connection.
1Eh	Free referencing distance	U32	rw	64h	1, 3	Free drive movement path in user defined position unit e.g. for referencing to block (Free referencing path).
2112h	Statistics	REC	rw	0h		Statistical information for drive operation
00h	Largest subindex	U8	ro	5h		-
01h	Number of watchdog events	U16	ro	0h	1, 4	Number of processor watchdog events

Object	Name	Format	Access	Default	Save	Description
02h	Number of position warnings	U16	rw	0h	1, 4	Number of Warning Position events (Table 18). Suppressing of warning can be activated in Object 2110h, Bit 20. Warning counter is not deactivated. Reset of warning counter by overwriting object with zero.
05h	Electronics temperature	U16	ro	-		Actual temperature of power electronics in °C
06h	i2t overload Level	U16	ro	0		Actual i2t Overload Level in [%], range of value 0..99
2113h	End test	REC				
00h	Largest Subindex	U8	ro	33h		-
20h	Break away current	U16	ro	factory provided		Unblocking current in mA for free run after referencing to block
27h	Position target range	I16	rw	4h	1,4	When the actual position is within the target position ± Position target range the target position window is reached.
31h	-	U32	ro	factory provided	-	No assigned
33h	Bootloader version	U8	ro	factory provided	-	Version of Bootloader Example: Version 5 = 05h
2114h	Warning temperature	U16	rw	50h	1, 3	Temperature limit in °C for activation of over temperature warning. If the electronics temperature (Object 2112h-05h) exceeds this limit, an EMCY message is activated (Object 1003h und 603Fh, error code 4210h-0000h). The drive can be operated further (50h = 80°C). In case of a temperature error, the drive is stopped an set to ErrorDiagnostic mode (error code 4210h-0001h). Error is acknowledged (error reset) by controlword (Object 6040h).
2300h	Customer EEPROM	ARR				Memory range for user data
00h	Largest subindex	U8	ro	7h		-
01h	Data0	U16	rw	0h	1, 4	-
...	...	...	...	...	...	...
07h	Data6	U16	rw	0h	1, 4	-
2800h	PDO1 add-on	U8	rw	1h	1, 2	0 = send PDO1 cyclic 1 = send PDO1 only in case of change n = send PDO1 only in case of change (n times, n : 2...255)
2801h	PDO2 add-on	U8	rw	1h	1, 2	0 = send PDO2 cyclic 1 = send PDO2 only in case of change n = send PDO2 only in case of change (n times, n : 2...255)
603Fh	Error code	U16	ro	0h		Last occurred error (Table 18, Object 1003h).
6040h	Controlword	U16	rw	0h		Controlword (see Table 9) Bit 0 = SwitchOn Bit 1 = EnableVoltage Bit 2 = QuickStop Bit 3 = EnableOperation Bit 4 = StartAction (move to target position) Bit 5 = Change set immediately Bit 6 = Relative/absolute movement Bit 7 = Error reset Bit 8 = Halt Bit 9…10 not defined Bit 11 = Jogging+ Bit 12 = Jogging- Bit 13...15 not defined

Object	Name	Format	Access	Default	Save	Description
6041h	Statusword	U16	ro	0h		Statusword (see Table 5) Bit 0 = Ready to switch on Bit 1 = SwitchOn enabled Bit 2 = Operation enabled Bit 3 = Error active Bit 4 = Voltage enabled Bit 5 = QuickStop Bit 6 = SwitchOn disabled Bit 7 = Warning active Bit 8 = CalibrationOK Bit 9 = Not used (permanently 1) Bit 10 = Target position reached Bit 11 = Internal software limit switch active Bit 12 = Drive moving Bit 13 = EEPROM access Bit 14 = HomingOK Bit 15 = External hardware limit switch active NOTE: HomingOK and CalibrationOK are set to zero as soon as a gear or position parameter is changed (see Object 6098h).
605Dh	Halt option code	I16	rw	2h	1, 3	Behaviour of the actuator after halt command (controlword Bit 8) 0h = Controller is deactivated, actuator is coasting 1h = Controlled quick stop (after standstill the controller is deactivated) 2h = according to 1h
6060h	Modes of operation	I8	wo	1h		Operating mode of drive (see mode diagram in Figure 4) 01h = Positioning mode 06h = Homing mode
6061h	Modes of operation display	I8	ro	1h		Operating mode of drive (see mode diagram in Figure 4) 01h = Positioning mode 06h = Homing mode
6063h	Position actual steps	I32	ro	-		Actual drive position in encoder steps
6064h	Position actual value	I32	ro	-		Actual drive position in user defined position unit (see Object 608Ah). Value is calculated from actual position in encoder steps multiplied by position factor. Position actual value = Position actual steps * Position factor NOTE: If position dimension is encoder steps, then position factor = 1. Gear ratio or Feed constant are not taken into account in this case. Position actual value is automatically saved when the drive is switched off or if the supply voltage is lost and is immediately available after switching on the drive. If an error occurred while saving the position, e.g. due to drive in movement, a position warning is activated (error code FF04h).
6068h	Position window time	U16	rw	0h	1,3	Time motor needs to stay in target position window, before motor control is switched off. 0h = function deactivated >0h = function activated, time in milliseconds
606Ch	Velocity actual value	I32	ro	-		Actual drive speed in user defined speed unit (see object 608Ch). Value is calculated from actual motor speed in revolutions per second multiplied with the velocity factor. Velocity actual value = motor speed * Velocity factor If speed dimension is encoder steps per second, then Velocity factor = 4. Gear ratio or Feed constant are not taken into account in this case.

Object	Name	Format	Access	Default	Save	Description
6073h	Maximum current	U16	rw	factory	1, 3	User adjustable i2t limitation for motor current to limit torque and protect overload. 00h = maximum current limit xxh = maximum motor current in mA Actual current is determined by calculation an average over 8 current values (every 12 milliseconds). If the current limit is exceeded, the active positioning task is terminated, an EMCY message is activated and the drive is set to ErrorDiagnostic. NOTE: Factory setting depends on motor type (see chapter Technical specifications and datasheet). The factory setting corresponds to the maximum value for the adjustment range of Maximum current and Object 6510h-7h setting. To enable maximum starting torque, the actual motor current can reach for short times the Maximum current of the factory setting, even if a lower Maximum current is adjusted by user.
6078h	Current actual value	I16	ro	-		Actual value of current in mA
607Ah	Target position	I32	rw	0h	1, 3	Drive target position in user defined position unit (see Object 608Ah) Bit 6 in controlword defines, if target position is absolute or relative to actual position. Intermediate position for backlash compensation are determined automatically (if function active). If new target position lies outside of software limit switches, drive sends an Abort message (0609003h = value outside of allowed range).
607Dh	Software position limit	ARR				Position of software limit switches. These positions can be directly defined or set by Homing procedure (see Object 6098h). The software limit switches can be deactivated by setting the subindices 01h and 02h to the same value.
00h	Largest subindex	U8	ro	2h		-
01h	Minimum software position limit	I32	rw	0h	1, 3	Position of minimum software limit switch in user defined position unit (see Object 608Ah)
02h	Maximum software position limit	I32	rw	0h	1, 3	Position of maximum software limit switch in user defined position unit (see Object 608Ah)
607Eh	Polarity	U8	rw	0h	1, 3	Direction of rotation of motor shaft for increasing position values (view to motor shaft from gear side) Bit7= 0: Rotation in counter-clockwise direction (CCW) = 1: Rotation in clockwise direction (CW) Bit0..6: not defined NOTE: After change of direction of rotation, positions of software limit switches must be mirrored (exchange the position value of the upper and lower software limit switch and change the prefix of both). For spur gears, every gear step leads to an inversion of direction of rotation of output shaft. Changing the polarity changes the prefix of the position actual value (object 6064h). HomingOK and CalibrationOK are set to zero as soon as parameter is changed (see Object 6098h).
607Fh	Max profile velocity	U32	rw	factory	1, 3	Maximum allowed speed in user defined speed unit (see Object 608Ch). Max profile velocity results from maximum motor speed. Maximum value is set by factory and cannot be exceeded.

Object	Name	Format	Access	Default	Save	Description
6081h	Profile velocity	U32	rw	factory	1, 3	Target speed in user defined speed unit (see Object 608Ch). For actual speed, see Object 606Ch). NOTE: Profile velocity must be smaller than max profile velocity (607Fh). If not, an abort message is activated (see SDO abort code in Table 19)
6089h	Position notation	I8	rw	FDh	1, 3	Notation of user defined position unit. FAh = mikro FBh = 10 mikro FCh = 100 mikro FDh = milli FEh = 10 milli FFh = 100 milli 00h = 1 (none) 01h = 10 02h = 100 03h = kilo NOTE: Valid for linear movement with length unit only (Position dimension = 01h). HomingOK and CalibrationOK are set to zero as soon as parameter is changed (see Object 6098h).
608Ah	Position dimension	U8	rw	01h	1, 3	Dimension of user defined position unit. Length unit for linear movement 01h = meter Angular unit for rotational movement (motor/gear output) 00h = revolution ACh = encoder step 10h = radian 41h = degree 42h = angular minute 43h = angular second The user defined position unit is given by multiplication of Position notation with Position dimension millimeter (valid for linear movement only). Example: Position unit in millimeters (Position notation = FDh and Position dimension = 01h). NOTE: HomingOK and CalibrationOK are set to zero as soon as parameter is changed (see Object 6098h).
608Bh	Velocity notation	I8	rw	00h	1, 3	Notation of user defined speed unit. FAh = mikro FBh = 10 mikro FCh = 100 mikro FDh = milli FEh = 10 milli FFh = 100 milli 00h = 1 (none) 01h = 10 02h = 100 03h = kilo NOTE: Valid for linear movement with length unit only.
608Ch	Velocity dimension	U8	rw	A3h	1, 3	Dimension of user defined speed unit. Length unit for linear movement A6h = meters per second A7h = meters per minute A8h = meters per hour Angular unit for rotational movement (motor/gear output) 00h = encoder steps per second A3h = revolutions per second A4h = revolutions per minute A5h = revolutions per hour The user defined speed unit is given by multiplication of Velocity notation with Velocity dimension (valid for linear movement only). Example: Speed unit in millimeters / second (Velocity notation = FDh and Velocity dimension = A6h).

Object	Name	Format	Access	Default	Save	Description
608Fh	Position encoder resolution	ARR				Single-turn resolution of quasi-absolute encoder in steps per revolution. The multi-turn resolution is 18 bits or 262'144 revolutions. NOTE: HomingOK and CalibrationOK are set to zero as soon as parameter is changed (see Object 6098h).
00h	Largest subindex	U8	ro	2h		-
01h	Encoder increments	U32	rw	4h	1, 3	Number of encoder steps per motor shaft revolution (counter)
02h	Encoder revolutions	U32	rw	1h	1, 3	Number of encoder or motor revolutions (enumerator)
6091h	Gear ratio	ARR				Gear ratio i in revolutions of motor shaft per revolutions of gear output shaft. Gear ratio = Motor revolutions / Gear shaft revolutions Notation as fraction number, since only integer number are allowed. Example: i = 61,25 : 1 = 6125 / 100. NOTE: HomingOK and CalibrationOK are set to zero as soon as parameter is changed (see Object 6098h).
00h	Largest subindex	U8	ro	2h		-
01h	Motor revolutions	U32	rw	1h	1, 3	Number of revolutions of motor shaft (counter)
02h	Gear shaft revolutions	U32	rw	1h	1, 3	Number of revolutions of gear output shaft (denominator)
6092h	Feed constant	ARR				Calculation factor to convert revolutions of motor or of gear output shaft into movement on user side. Feed constant = Feed / Spindle shaft revolutions. For linear movements with spindle, Feed constant corresponds to the spindle gauge. Notation as fraction number, since only integer number are permitted. Example: Spindle gauge 2,5 mm / revolution = 1000 mm / 400 revolution = 1 meter / 400 revolution. NOTE: The length unit for Feed is always meters. The setting of Position notation (6089h) and Position dimension (608Ah) is not taken into account. HomingOK and CalibrationOK are set to zero as soon as parameter is changed (see Object 6098h).
00h	Largest subindex	U8	ro	2h		-
01h	Feed	U32	rw	1h	1, 3	Path in meters (counter) NOTE: The length unit for Feed is always meters.
02h	Spindle shaft revolutions	U32	rw	1h	1, 3	Number of revolutions (denominator)
6093h	Position factor	ARR				Calculation factor to convert Position actual steps (in encoder steps) in Position actual value (in user defined position unit, see Object 608Ah). NOTE: For Position dimension encoder steps, position factor is always 1. For position conversion, Gear ratio and Feed constant are no taken into account in this case.
00h	Largest subindex	U8	ro	2h		-
01h	Numerator	U32	ro	1h		
02h	Feed constant	U32	ro	1h		
6094h	Velocity factor	ARR				Calculation factor to convert motor rotational speed (in revolutions per second) in Velocity actual value (in user defined speed unit, see Object 608Ch).
00h	Largest subindex	U8	ro	2h		-
01h	Numerator	U32	ro	1h		

Object	Name	Format	Access	Default	Save	Description
02h	Divisor	U32	ro	1h		
6098h	Homing method	I8	rw	0h		Methods for referencing of drive position. After finishing referencing procedure, the reference position is saved (Object 11h). For details, see chapter 4.3. Parameter / description 00h No referencing procedure active F6h Set actual position (Object 6064h) as minimum software limit switch (Object 607Dh-1h) F5h Set actual position (Object 6064h) as maximum software limit switch (Object 607Dh-2h) F4h Referencing with preset value: Actual position (Object 6064h) is overwritten with target position (Object 607Ah). Subsequently, bit 14 (HomingOK) is set automatically in statusword (Object 6041h). F3h Referencing with measuring method (Figure 16): Function can be used to calibrate the position of a reference switch which is difficult to access: Actual position (Object 6064h) is automatically overwritten with target position (Object 607Ah). In addition, reference position (Object 2111h-11h) is corrected by measuring offset. Subsequently, bit 8 (CalibrationOK) is set automatically in statusword (Object 6041h). NOTE: Possible only, if in statusword (Object 6041h) the bit 14 (HomingOK) is already set. ECh Move to minimum software limit switch EBh Move to maximum software limit switch EAh Move to position zero E2h Referencing to minimum reference switch (Figure 14): Move to minimum reference switch. As soon as minimum reference switch is activated, the actual position (Object 6064h) is automatically overwritten with the saved reference position (Object 2111h-11h). Subsequently, bit 14 (HomingOK) is set automatically in statusword (Object 6041h). NOTE: Possible only, if Input 1 or Input 2 (Object 0Dh or 0Eh) are set to Referencing with preset value (12h). E1h Referencing to maximum reference switch (Figure 14): Description analog description E2h referencing to minimum reference switch. D8h Referencing to minimum block (Figure 15): Move to minimum block. As soon as motor is blocked, automatically overwrite actual position (Object 6064h) with saved reference position (Object 2111h-11h). Subsequently, bit 14 (HomingOK) is set automatically in statusword (Object 6041h). D7h Referencing to maximum block (Figure 15). Description analog description D8h referencing to minimum block. After finishing referencing procedure, Object Homing method is automatically set back to 0h. NOTE: HomingOK and CalibrationOK are automatically set to zero if drive is set to Homing mode or if one of the following position calculation parameters is changed: Polarity 607Eh, Position notation 6089h, Position dimension 608Ah, Position encoder resolution 608Fh, Gear ratio 6091h, Feed constant 6092h (see chapter 4.3.5).
6099h	Homing speed	ARR				Speed for referencing procedure
00h	Largest subindex	U8	ro	2h		-
01h	Speed during search for switch	U32	rw	0Ah	1, 3	Speed in user defined speed unit while referencing to block or to limit switch.

Object	Name	Format	Access	Default	Save	Description
02h	Speed during search for zero position	U32	rw	0Ah	1, 3	Speed in user defined speed unit for movement to position zero during referencing procedure.
60FDh	Digital input monitor	U32	ro	00000000h		Object to monitor status of external hardware inputs Bit0..15 not defined Bit16 Monitor bit for digital input 1 (Object 2111h-0Dh) Bit17 Monitor bit for digital input 2 (Object 2111h-0Eh) Bit18..31not defined
6402h	Motor type	U16	rw	0001h		Motor type (0001h = DC motor with PWM control)
6410h	Motor data	ARR				Motor parameters
00h	Largest subindex	U8	ro	2h		-
01h	Motor operating time	U32	ro		1, 4	Operating time of motor in seconds. Value is stored every 6 minutes.
02h	Software version controller	U32	ro	factory		Version of regulator processor firmware (example: Version 1.0 = 0001h 0000h)
6502h	Drive modes	U32	ro	00000021h		Bit0 Positioning mode Bit5 Homing mode
6510h	Drive data	ARR				Drive parameters
00h	Largest subindex	U8	ro	4		-
01h	Drive operating time	U32	ro		1, 4	Operating time of drive in seconds
03h	Number of positioning tasks	U32	ro		1, 4	Number of positioning tasks. Value is stored every 6 minutes.
04h	Number of homing tasks	U32	ro		1, 4	Number of executed Homing procedures and software limit switch definitions. Value is stored every 6 minutes.
05h	Number of saving tasks	U23	ro		1, 4	Number of save to EEPROM procedures. Value is stored every 6 minutes.
06h	Error temperature	U16	ro	6Eh		Temperature limit in °C for activation of over temperature error (6Eh = 110°C). If electronics temperature (Object 2112h-05h) exceeds Error temperature, the drive stops and an EMCY message is activated (Object 1003h und 603Fh). Subsequently, the drive changes to Error Diagnostic (Error code 4210h-0001h). Acknowledge of temperature error (Error reset) by controlword (Object 6040h). The Error temperature for over temperature error is defined by factory and cannot be changed.
07h	Max motor current	U16	ro	factory	1, 4	Maximum allowed motor current in mA. This current is set by factory and cannot be changed. This current defines the maximum current which can be adjusted in Object 6073h.
08h	Min position	S32	ro			Minimum permitted position in user defined position unit (see Object 608Ah)
09h	Max position	S32	ro			Maximum permitted position in user defined position unit (see Object 608Ah)
0Ah	Max velocity	U32	ro			Maximum permitted speed in user defined speed unit (see Object 608Ch).

4 Quick start-up

4.1 Adjust node-ID and baudrate

After correctly connecting and switching on the drive, a BootUp message occurs on the CAN bus. Now, the drive must be configured according to application.

Adjust node-ID and baudrate via switches

Node-ID and baudrate can be adjusted via switches. Switch settings are dominant. Factory default setting is switches in setting 00h. Node-ID and baudrate are then defined by parameter settings in Object 2100h and 2101h or in EEPROM.

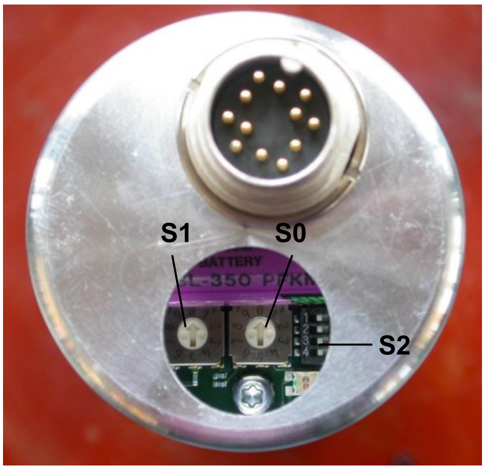


Figure 12: Drive backside with rotary switch S1 (left hand side) and S0 (right hand side) for adjustment of node-ID, DIP switch S2 for adjustment of baudrate and terminating resistor.

Table 11: Adjustment of node-ID via decimal rotary switch S1 and S0 (assignment, see Figure 13). Factory default setting is 00d. In this case, node-ID and baudrate are defined by Object 2101h and 2100h and the DIP-switch is inactive (Table 13).

Dec rotary switch position S1 (left)	Dec rotary switch position S0 (right)	Node-ID
0	0	Object 2101h
0	1...9	1...9
1	0...9	10...19
2	0...9	20...29
3	0...9	30...39
4	0...9	40...49
5	0...9	50...59
6	0...9	60...69
7	0...9	70...79
8	0...9	80...89
9	0...9	90...99

Table 12: Adjustment of baudrate and terminating resistor via DIP-switch. Factory default setting is S0...S3=OFF. DIP-switch is active only, if adjustment of decimal rotary switches S1 und S0 is >00d.

Position S0	Position S1	Position S2	Position S3	Baudrate [kBit/s]	Terminating resistor
OFF	OFF	OFF	-	10	-
ON	OFF	OFF	-	20	-
OFF	ON	OFF	-	50	-
ON	ON	OFF	-	125	-
OFF	OFF	ON	-	250	-
ON	OFF	ON	-	500	-
OFF	ON	ON	-	800	-
ON	ON	ON	-	1000	-
-	-	-	OFF	-	open
-	-	-	ON	-	120 Ω

After next initialization, the drive appears on the bus with new node-ID and baudrate.
NOTE: Baudrate of master needs to be adapted as well to connect to drive.

Adjustment of node-ID and baudrate via EEPROM

Adjustment of node-ID in EEPROM is done in Object 2101h. This is possible only, if all adjustment switches are in factory default setting.

Example: Adjustment of node-ID to 23h:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	2Fh	01h	21h	0h	23h	Xx	Xx	Xx

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	01h	21h	0h	0h	0h	0h	0h

Table 13: Adjustment of node-ID and baudrate in EEPROM via Object 2100h. Here, only an index is written as parameter into object, not the effective baudrate (only possible, if rotary switches in factory default setting 00).
NOTE: Index for Object 2100h and LSS are different.

Baudrate [kBit/s]	10	20	50	100	125	250	500	800	1000
Index (definition in Object 2100h)	0h	1h	2h	3h	4h	5h	6h	7h	8h
Index (LSS) according to CiA table	8h	7h	6h	5h	4h	3h	2h	1h	0h

Example: Adjustment of baudrate to 250 kBit/s (5h):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	2Fh	00h	21h	0h	05h	Xx	Xx	Xx

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	00h	21h	0h	0h	0h	0h	0h

After adjustment, settings must be saved via Object 1010h into non-volatile memory (EEPROM). In order to prevent unintended save, write message "save“ into subindex 1.

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	10h	10h	01h	73 's’	61 'a’	76 'v’	65 'e’

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	10h	10h	01h	0h	0h	0h	0h

After next initialization, the drive appears on the bus with new node-ID and baudrate.
NOTE: Baudrate of master needs to be adapted as well to connect to drive.

Adjustment of node-ID and baudrate via LSS

If several bus participants have the same node-ID and baudrate, they can be adapted via LSS. LSS uses parameters such as product code, revision number, vendor-ID and unique serial number to identify and configure bus participants (see 0).

After next initialization, the drive appears on the bus with new node-ID and baudrate.
NOTE: Baudrate of master needs to be adapted as well to connect to drive.

4.2 Read and write objects

To read and write an object (SDO), always two telegrams are exchanged between master and drive.

Write object

First, the master sends the new object value. Subsequently, the drive sends a response as confirmation.

Master sends new object value (wxyz):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Dh	60h	1h	w	x	y	z

Drive response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Dh	60h	1h	0	0	0	0

Read object

First, the master sends a request for the desired value. Then, the drive sends the requested value.

Master request:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	40h	41h	60h	0	x	x	x	x

Drive response (abcd):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	43h	41h	60h	0	a	b	c	d

Save parameters to EEPROM

Parameters can be saved to EEPROM by using Object 1010h. For that purpose, write save into Object 1010h, subindex 1, byte 5 to 8. For further details, see Object description in Table 10.

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600h+Node-ID	8	23	10	10	01	73 (= s)	61 (= a)	76 (= v)	65 (= e)

Load default parameters from EEPROM

Parameters can be loaded from EEPROM by using Object 1011h. For that purpose, write load into Object 1011h, subindex 1, byte 5 to 8. For further details, see Object description in Table 10.

NOTE: Actual values in RAM are overwritten in this case.

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600h+Node-ID	8	23	11	10	01	6C (= l)	6F (= o)	61 (= a)	64 (= d)

4.3 Configure drive parameters

4.3.1 Notation, dimension and gear

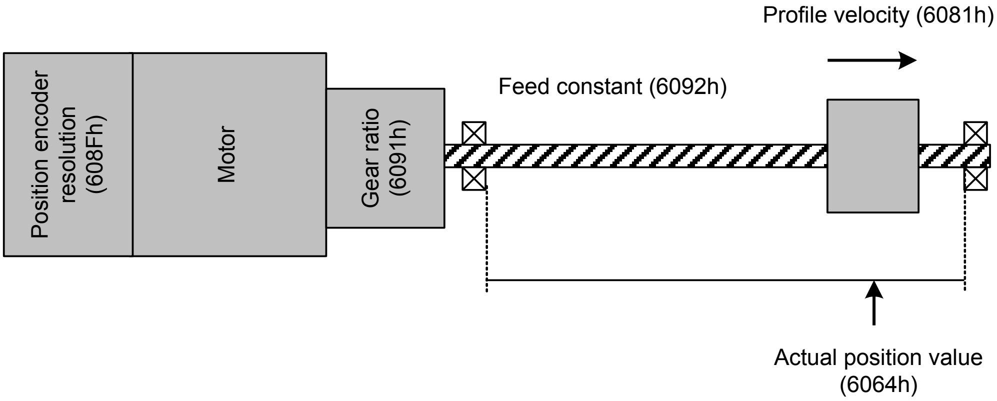


Figure 13: Setting of drive configuration.

User defined position unit (Position notation and Position dimension)

The user defined position unit is composed of Position notation (Object 6089h) and Position dimension (Object 608Ah). Position notation corresponds to a prefactor, e.g. for linear movement micro (10^{-6}), milli (10^{-3}), 1 (10^{0}) or kilo (10^{3}). For rotational movement, only Position notation = 1 is useful.
Position dimension corresponds to the effective position unit, e.g. for linear movement meters and for rotational movement encoder steps or degrees. The user defined position unit is given by multiplication:

$$\text{User defined position unit} = \text{Position notation} * \text{Position dimension}$$

$$\text{Example: } [\text{Position}] = \text{milli} * \text{meter} = \text{millimeter}$$

After definition of user defined position unit, all position objects for drive control (except Object 6063h) refer to this unit and are internally converted to encoder steps while taking into account Gear ratio and Feed constant.

The above principle applies as well for the user defined speed unit.

Position encoder resolution (Object 608Fh)

This object contains 2 sub indices. Subindex 1 contains the number of encoder steps. Subindex 2 contains the number of revolutions. The MSQA encoder has 4 steps per motor revolution, i.e. position encoder resolution = 4.

$$\text{Position encoder resolution} = \frac{\text{Encoder increments}}{\text{Motor revolutions}}$$

Gear ratio (Object 6091h)

This object contains in sub indices 1 and 2 the gear transmission ratio as a fraction with an integer number as counter and as denominator. For a gear ratio of 61,25, subindex 1 contains value 6125 and subindex 2 contains value 100. 6125/100 = 61,25

$$\text{Gear ratio} = \frac{\text{Motor revolutions}}{\text{Gear shaft revolutions}}$$

Feed constant (Object 6092h)

This object contains in subindex 1 and 2 spindle or user gear data. For a spindle with a gradient of 2,5 mm per revolution, 400 revolutions are required for one meter of linear movement.

NOTE: The length unit for Feed is always meters. The setting of Position notation (6089h) and Position dimension (608Ah) is not taken into account.

$$\text{Feed constant} = \frac{\text{Feed}}{\text{Spindle shaft revolutions}}$$

$$\text{Example: Feed constant} = \frac{1\text{m}}{400\text{U}} = \frac{1000\text{mm}}{400\text{U}} = 2{,}5\frac{\text{mm}}{\text{U}}$$

Position factor (Object 6093h)

$$\text{Position factor} = \text{Position notation} \cdot \text{Position dimension} \cdot \text{Encoder resolution} \cdot \frac{\text{Gear ratio}}{\text{Feed constant}}$$

$$\text{Example: Position factor} = \frac{1}{1} \cdot \frac{1\text{m}}{1000\text{mm}} \cdot \frac{\frac{4\text{ steps}}{1\text{rev}} \cdot \frac{6125}{100}}{\frac{1\text{m}}{400\text{rev}}} = 98\frac{\text{steps}}{\text{mm}}$$

Position actual value (Object 6064h)

$$\text{Position actual value} = \frac{\text{Position actual steps}}{\text{Position factor}}$$

$$\text{Example: } 100\text{mm (application side)} \Leftrightarrow 9'800 \text{ encoder steps (drive)}$$

$$\text{Example: Maximum value for Position actual value} = \frac{2^{31}-1}{98}\text{mm} = 21'913'098\text{mm}$$

NOTE: For Position dimension encoder steps, position factor is always 1. For position conversion, Gear ratio and Feed constant are no taken into account in this case. By using the above formula, the maximum Position actual value can be calculated from the maximum number of Positions actual steps (2^{31}-1) and the position factor (see example).

User defined speed unit (Velocity notation and Velocity dimension)

The user defined speed unit is composed of Velocity notation (Object 6089h) and Velocity dimension (Object 608Ah). Velocity notation corresponds to a prefactor, e.g. for linear movement micro (10^{-6}), milli (10^{-3}), 1 (10^{0}) or kilo (10^{3}). For rotational movement, only Velocity notation = 1 is useful.
Velocity dimension corresponds to the effective speed unit, e.g. for linear movement meters per second and for rotational movement encoder steps per second or degrees per second. The user defined speed unit is given by multiplication:

$$\text{User defined speed unit} = \text{Velocity notation} * \text{Velocity dimension}$$

$$\text{Example: } [\text{Speed}] = \text{milli} * \text{meter} / \text{s} = \text{millimeter} / \text{s}$$

After definition of user defined speed unit, all speed objects for drive control refer to this unit and are internally converted to encoder steps per second while taking into account Gear ratio and Feed constant.

Velocity factor (Object 6094h)

$$\text{Velocity factor} = \text{Velocity notation} \cdot \text{Velocity dimension} \cdot \text{Encoder resolution} \cdot \frac{\text{Gear ratio}}{\text{Feed constant}}$$

$$\text{Example: Velocity factor} = \frac{1}{60} \cdot \frac{1}{1000} \cdot \frac{\frac{4}{1} \cdot \frac{6125}{100}}{\frac{1}{400}} = 0.408$$

Velocity actual value (Object 606Ch) und Profile velocity (Object 6081h)

$$\text{Velocity actual value} = \frac{\text{Motor speed in U/s}}{\text{Velocity factor}} \qquad \text{Profile velocity} = \frac{\text{Motor speed in U/s}}{\text{Velocity factor}}$$

$$73\,\text{mm/min (application side)} \Leftrightarrow 30\,\text{U/s (drive side)}$$

NOTE: For velocity dimension encoder steps per second, automatic setting is Velocity factor = 1/4. For position conversion, Gear ratio and Feed constant are no taken into account in this case.

4.3.2 Direction of rotation (Polarity, Object 607Eh)

This object defines the direction of rotation of the motor shaft for increasing positions values.

4.3.3 Current and speed

Drive torque limitation is done by limiting the maximum motor current (Object 6073h). A value between 0 mA and the maximum motor current (see Object 6510h-7h) is possible. Value 0 corresponds to the maximum current, which is given by the factory default setting. Values between 1 mA and the maximum current lead to a limitation of drive torque.

Velocity is given in user defined speed unit. Velocity dimension and notation (Objects 608Bh, 608Ch) as well as position encoder resolution, gear ratio and feed settings are taken into account for conversion (Objects 608Fh, 6091h, 6092h).

4.3.4 i2t-overload protection

The i2t monitoring protects the motor from thermally overloading. This is especially useful in applications in which the drive is operated repeatedly and for a limited time above the S1 rated current.

The motor power loss equivalent is integrated over time and displayed in parameter i2t level (Object 2112h-06h). In case on an i2t level ≥100% the motor is thermally overloaded. As a result state changes to ErrorDiagnostic (error Continuous over current, Error Code 2310h) and motor control is aborted.

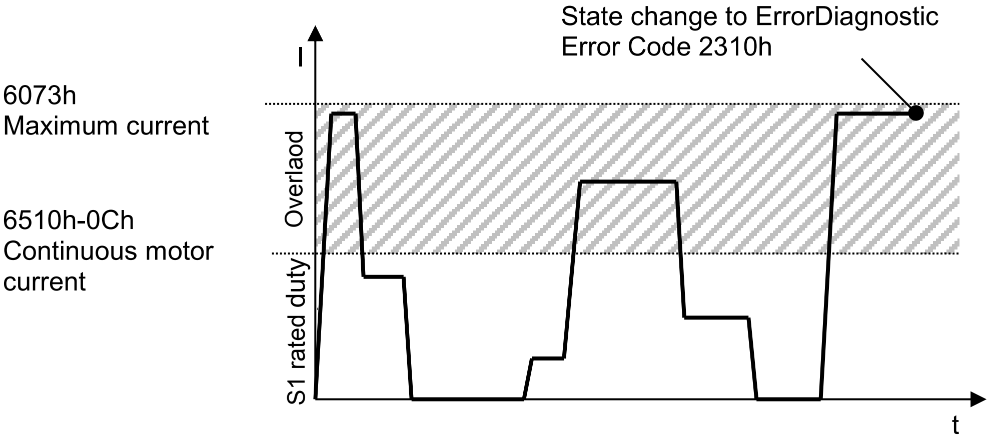


Figure 14: Example of current behavior that leads to i2t overload.

4.3.5 Current limiting

With activating current limiting the controller does not allow any current bigger than the maximum current defined in object 6073h. If the actuator faces increasing load, a constant torque is applied. If the actuator is blocked by the load a warning motor blocked (Object 7121h 0001h) is set.

Current limiting activated

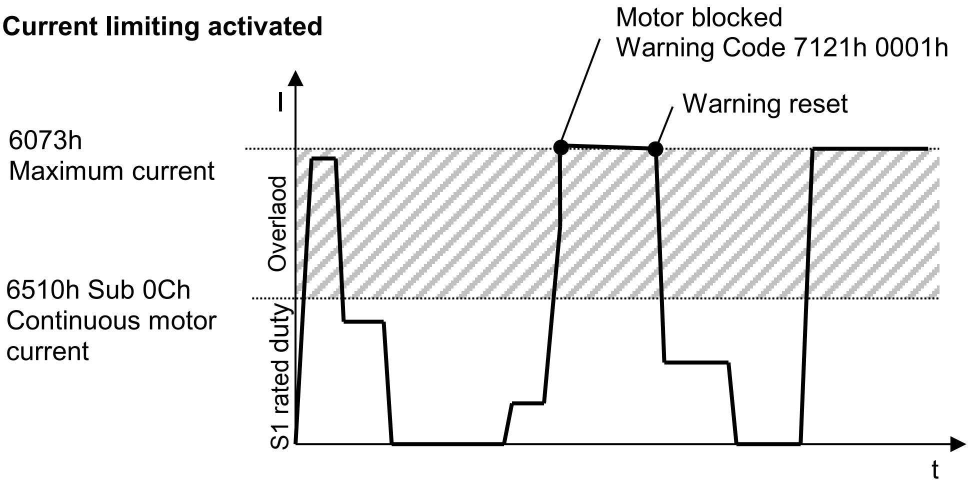


Figure 15: Example of current behavior that leads to current limiting warning.

NOTE: With current limit deactivated the lifetime of brushed DC-motor is expected to be reduced.

With current limiting deactivated the motor control is aborted in case the motor current in exceeding maximum current for more than typ. 100ms. As a result error Motor blocked (Error Code 7121h 0000h) is set.

Current limiting deactivated

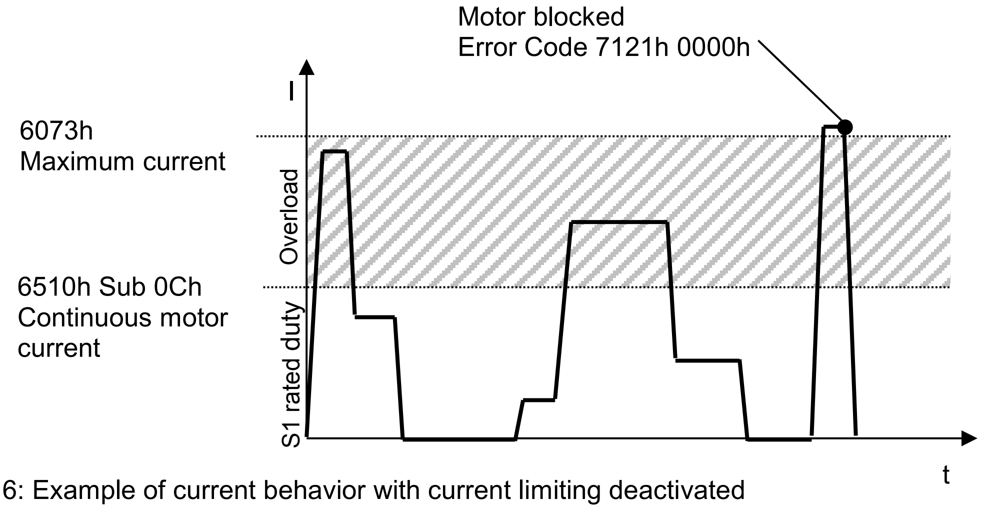


Figure 16: Example of current behavior with current limiting deactivated

Per Default current limiting is deactivated (see Version control Object 2110h Bit 19).

4.3.6 Positioning timeout (Object 2111-03h)

This object contains the maximum allowed time for a positioning task in milliseconds (if >0), before an EMCY message is activated.

4.3.7 Backlash compensation (Object 2111-01h)

Object 2111h subindex 1 contains options how the drive should move to compensate spindle or gear play. The target position can be approached either directly, from lower or from higher position values. Additionally, in subindex 2 is defined, how far the drive exceeds the target position for backlash compensation.

Example: The actual position of drive is 1000. The target position is 2000. If option “move directly to target position” is active, then drive moves to 2000 directly from lower position values. If option “approach target position from higher position values” is active, then drive moves first to intermediate position 2100 (for backlash delta 100) and subsequently from higher position values to position 2000.

Table 14: Drive parameter examples.

Parameter	Object	Subindex	Value	Description
Position notation	6089h	0	FDh	milli (*10^{-3})
Position dimension	608Ah	0	01h	meter (linear movement)
Velocity notation	608Bh	0	FDh	milli (*10^{-3})
Velocity dimension	608Ch	0	A6h	meter / min
Encoder increments	608Fh	1	04h	8 encoder steps per revolution
Motor revolutions	608Fh	2	01h	
Gear ratio – motor revolutions	6091h	1	17EDh	61,25 = 6125 / 100
Gear ratio – shaft revolutions	6091h	2	64h	
Feed constant - feed	6092h	1	01h	2,5 mm / revolution
Feed constant – shaft revolutions	6092h	2	190h	= 1000 mm / 400 revolutions = 1 Meter / 400 revolutions

4.3.8 Save parameters

After parameter adjustment, settings must be saved via Object 1010h into non-volatile memory (EEPROM). In order to prevent unintended save, write message "save“ into subindex 1.

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	10h	10h	1h	73 ‚s’	61 ‚a’	76 ‚v’	65 ‚e’

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	10h	10h	1h	0	0	0	0

4.4 Referencing position (homing)

For correct operation, the drive position needs to be adjusted with respect to a reference point in the machine or application. Usually, this is required only one time after drive installation. After switching on, the drive is in Ready Mode. Referencing is done in Homing mode with Object 6098h. Mode change is done by:

1. Set Object 6060h (Modes of operation) to 6=Homing
2. Set Object 6040h (Controlword) to 000Fh
   SwitchOn, EnableVoltage, QuickStop and EnableOperation = 1

Move to machine reference point

The drive offers several convenient procedure for referencing of drive position:

1. Cautious approach to machine reference point by Jogging, e.g. by control via digital hardware inputs
    a. Define digital hardware inputs in Object 2111h, subindex 2 as input Jogging+/-.
2. Cautious approach to machine reference point by target position via CAN bus (relative / absolute)
    a. Define target position in user defined position unit (see Object 607Ah)
    b. Set relative or absolute movement in Object 6040h, bit 6
    c. Move drive by Object 6040h, bit 4 (HomingStart)
3. Move drive to block (block as machine reference point)
    a. Set current limit to minimum in Object 6073h
    b. Move drive by Object 6098h, value –40, -41
    c. NOTE: Due to gear ratio, large forces may occur.
4. Move drive to reference switch (position of reference switch as reference point)
    a. Mount external reference switch in machine at reference point
    b. Define digital hardware input in Object 2111h, subindex 2 as hardware limit switch
    c. Move drive by Object 6098h, value –30, -31, until reference switch becomes active
    d. After reference position is found and saved, external reference switch can be dismounted

Set reference position directly

NOTE: Direct setting of reference position can be dangerous. If an incorrect reference position is entered, the software limit switches are without effect. Then, the drive can hit a barrier or block and damage can occur.

For this reason, CiA has defined a safety procedure. In this safety procedure, first the reference position to be set is written into the target position. Subsequently, the target position is written to the actual drive position by using Object 6098, Value –12.

Example: Set drive position at actual position to reference position 100 mm.

1. Set drive to HomingReady mode
2. Set Object 607Ah (target position) to 100 (in this example, drive operates mm)
3. Select referencing procedure in Object 6098 (Homing method) to –12
4. Start referencing with Object 6040h (controlword) = 001Fh

Table 15: Overview of referencing methods (Object 6098h) in HomingReady mode.

Object 6098h		Description
0	0h	No referencing procedure active
-10	F6h	Set actual position as minimum software limit switch
-11	F5h	Set actual position as maximum software limit switch
-12	F4h	Referencing with preset value
-13	F3h	Referencing with measuring method
-20	ECh	Move to minimum software limit switch
-21	EBh	Move to maximum software limit switch
-22	EAh	Move to position zero
-30	E2h	Referencing to minimum reference switch
-31	E1h	Referencing to maximum reference switch
-40	D8h	Referencing to minimum block
-41	D7h	Referencing to maximum block

By start command, the desired selection can be activated and started. In PositioningReady mode, the target position can be set and a positioning task can be activated (see 2.4.3).

4.4.1 Set position or referencing position directly

Set drive to HomingReady mode:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0
600h+Node-ID	5	2Fh	60h	60h	0	06h

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	60h	60h	0	0	0	0	0

Write command 0Fh into controlword:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	0Fh	0	0	0

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Set Homing method: Referencing with preset value: Write target position to actual position.

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0
600h+Node-ID	5	2Fh	98h	60h	0	F4h

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	98h	60h	0	0	0	0	0

Set target position (or 0):

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Ah	60h	0	x	x	x	x

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Ah	60h	0	0	0	0	0

Write start to controlword:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	1Fh	0	0	0

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Bit 14 HomingOK in statusword (Object 6041h) is set to 1 as soon as actual position is set to new value.

Set drive to PositioningReady mode:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0
600h+Node-ID	5	2Fh	60h	60h	0	01h

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	60h	60h	0	0	0	0	0

Write command 0Fh to controlword:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	0Fh	0	0	0

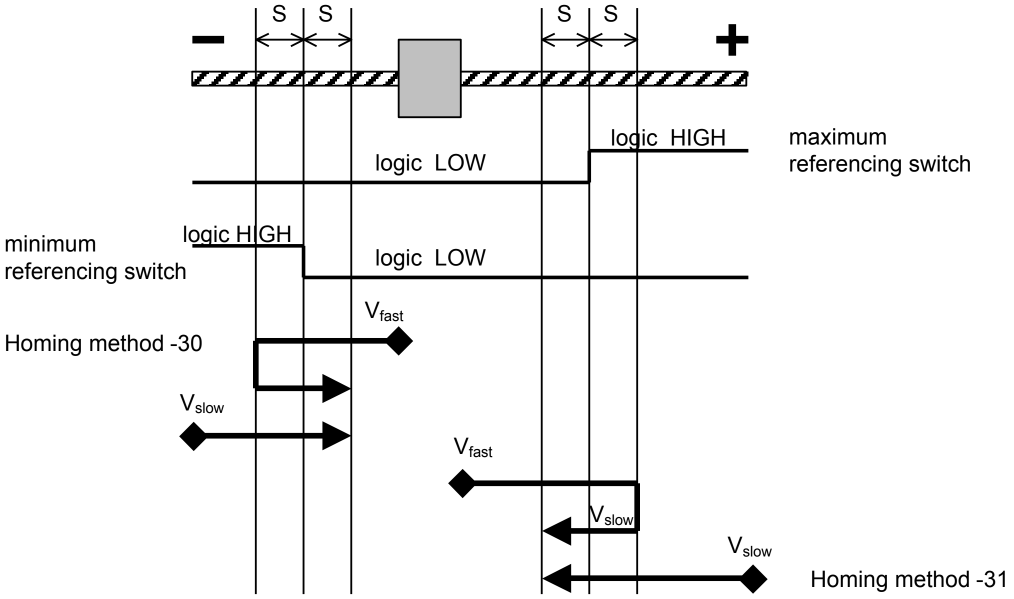
Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Now, the drive is in PositioningReady mode and is ready for positioning tasks.

4.4.2 Referencing with referencing switch

The reference switch position is indicated by a change of switch state from logic HIGH to logic LOW. Upon this change, the reference position (Object 2111h-11h) is automatically written to the actual position (Object 6064h). The referencing procedure is finished after automatic setting of bit 14 HomingOK in statusword 6041h.



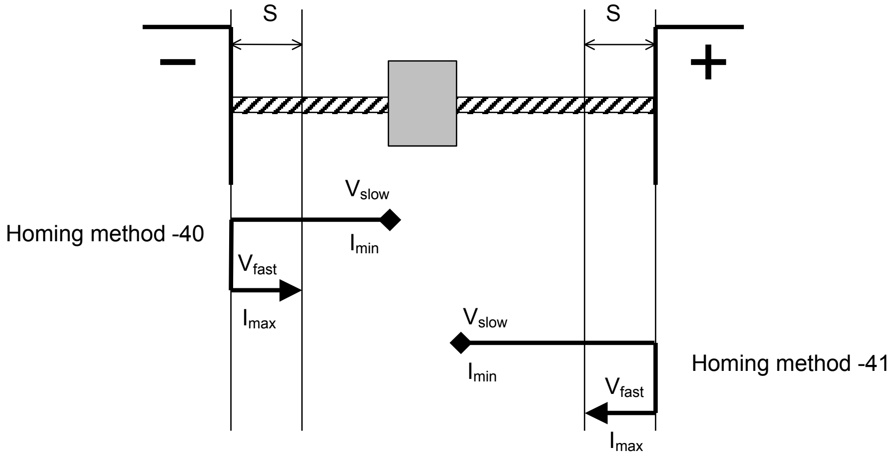
S: Free referencing distance (Object 2111-1Eh)
V_{fast} Search for referencing switch with profile velocity (Object 6081h)

Figure 17: Referencing to minimum or maximum referencing switch.

NOTE: Only one digital input can be configured as a reference switch.

4.4.3 Referencing to block

The drive moves with Homing speed to the block. Upon blocking of the motor, the reference position (Object 2111h-11h) is automatically written to the actual position (Object 6064h). Subsequently, the drive performs a free movement with maximum current by the path S. The referencing procedure is finished after automatic setting of bit 14 HomingOK in statusword 6041h.



Search block with
V_{slow} slow speed (Homing speed, Object 6099-1h)
I_{min} minimum current (Max current, Object 6073h)

Free movement with
V_{fast} maximum speed (Profile velocity, Object 6081h)
I_{max} maximum current (Max current, Object 6073h)
S: Free path (Free referencing distance, Object 2111-1Eh)

Figure 18: Referencing on block in negative or in positive direction.

4.4.4 Referencing with measurement method

Before using this Homing method, ensure drive referencing. Therefore, bit 14 in statusword (Object 6041) HomingOK must be set already. During this procedure, first measuring offset is determined. Subsequently, the actual position is overwritten with the target position. Additionally, the reference position is corrected by the measuring offset. The reference procedure is finished after automatic setting of bit 8 in statusword (6041h) CalibrationOK.

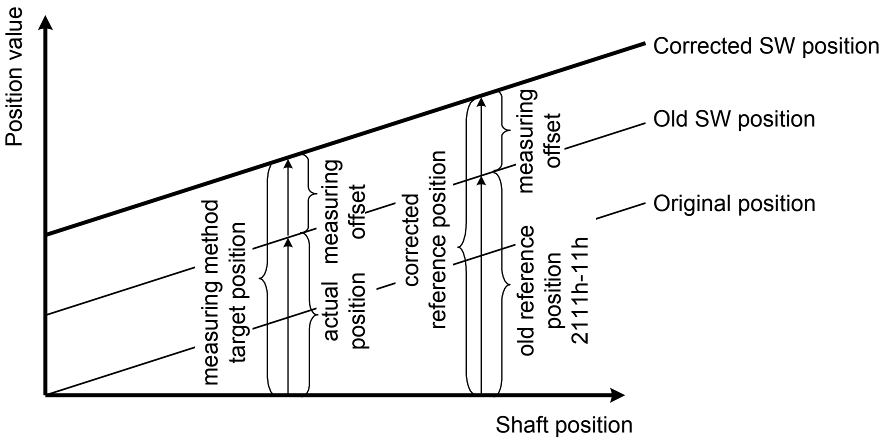

Position value
Corrected SW position
Old SW position
Original position
measuring method target position
actual measuring position
measuring offset
corrected reference position
old reference position 2111h-11h
measuring offset
Shaft position

Figure 19: Referencing with measuring method.

4.4.5 Reset status bits HomingOK and CalibrationOK

The status bits HomingOk (bit 14) and CalibrationOK (bit 8) are reset to zero by the following actions:

- Loss of supply voltage during a positioning task. If separated supply voltage connection is used and supply voltage remains connected to +VsE, HomingOK and CalibrationOK are not reset.
- Drive movement in service operation without bus communication.
- After starting the following Homing methods:

Object 6098h		Description
-12	F4h	Referencing with preset value (overwrite actual position with target position)
-13	F3h	Referencing with measuring method
-30	E2h	Referencing to minimum referencing switch (move to minimum referencing switch)
-31	E1h	Referencing to maximum referencing switch (move to maximum referencing switch)
-40	D8h	Referencing to minimum block (move to minimum block)
-41	D7h	Referencing to maximum block (move to maximum block)

- After overwriting the following objects:

Object	Description
607Eh-00h	Polarity
6091h-01h	Gear - Motor revolutions
6091h-02h	Gear - Gear shaft revolutions
6089h-00hh	Position notation
608Ah-00h	Position dimension
608Fh-01h	Position encoder resolution - Encoder increments
608Fh-02h	Position encoder resolution - Encoder revolutions
6092h-01h	Feed constant - Feed
6092h-02h	Feed constant - Spindle shaft revolutions

4.5 Set software limit positions

After referencing, software limit switch positions can be defined. These positions are always related to the actual reference and the offset value. If referencing or offset value is shifted, software limit switch positions are shifted as well.

Move drive to software limit switch position

The drive can be moved to the desired position for a software limit switch by Jogging or by carrying out appropriate positioning tasks via CAN bus.

Set software limit switch position

After moving the drive to the desired position for a software position limit, this position can be assigned to the minimum or maximum software limit switch position by using Object 6098h, value –10 or –11.

Set software limit switch position directly

Software limit switch positions can be directly set in Object 607Dh, subindex 1 or 2.

Set position for minimum software limit switch directly:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Dh	60h	1	x	x	x	x

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Dh	60h	1	0	0	0	0

Set position for maximum software limit switch directly:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Dh	60h	2	x	x	x	x

Response:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Dh	60h	2	0	0	0	0

4.6 Perform positioning tasks

After parameter definition, a change from Ready Mode to PositioningReady mode must be performed, before the drive can carry out positioning tasks:

1. Set Object 6060h (Modes of operation) to 1 = Positioning mode
2. Set Object 6040h (Controlword) to 000Fh
   SwitchOn, EnableVoltage, QuickStop and EnableOperation = 1

Subsequently, a target position can be defined in Object 607Ah. Relative or absolute movement is defined in bit 6 of Object 6040h. Additional setting of bit 4 in Object 6040h initiates the positioning task. The drive is stopped by setting bit 8 in Object 6040h (intermediate stop). Subsequent restart of the positioning task is done by reset of bit 8. Then, the drive continues movement to the originally defined target position.

4.7 Stop of motor control after positioning

Stop of the motor control after positioning can increase lifetime of the brushed DC-motor.
This functionality prevents an automatic readjusting if the position is moved out of the target position window.

After staying in the target position window for a time defined in position window time (object 6068h), status bit 10 (Target reached) is set and the motor control is switched off. No further automatic readjustment of position is then performed.

Per default this functionality is deactivated (Object 6068h = 0h). The function is activated, with setting Position window time > 0 (Object 6068h > 0h).

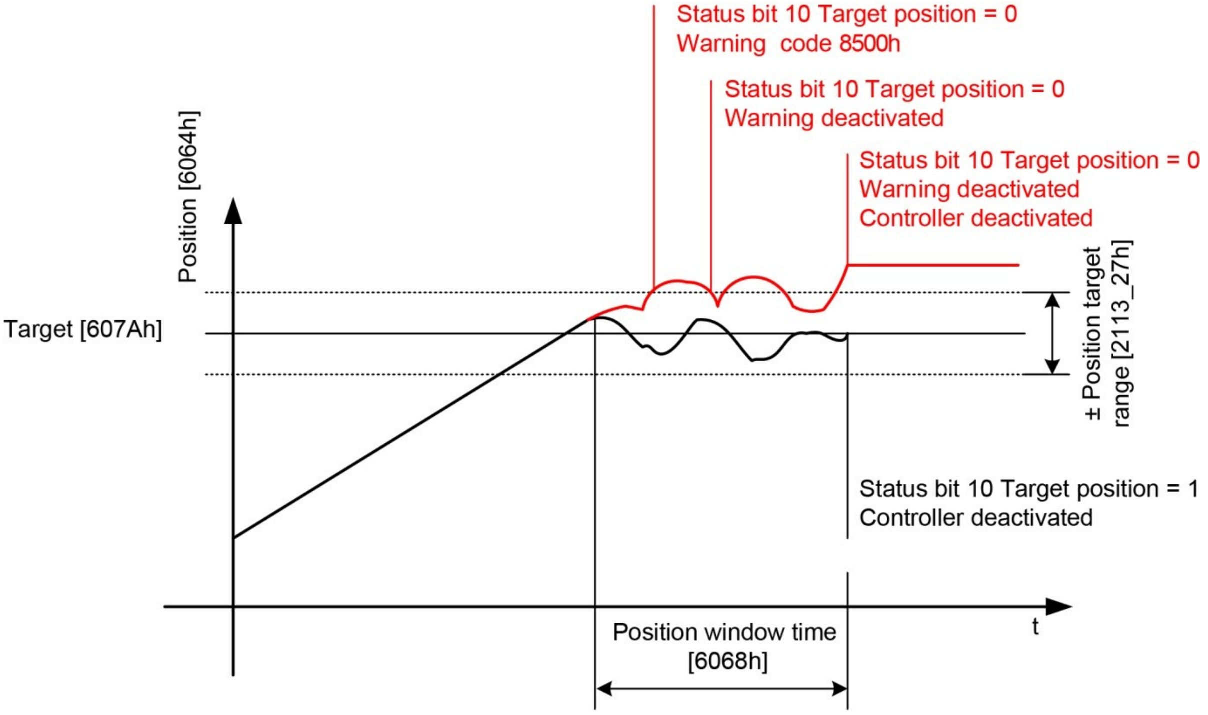

Status bit 10 Target position = 0
Warning code 8500h
Status bit 10 Target position = 0
Warning deactivated
Status bit 10 Target position = 0
Warning deactivated
Controller deactivated
Position [6064h]
Target [607Ah]
± Position target range [2113_27h]
Status bit 10 Target position = 1
Controller deactivated
Position window time [6068h]
t

Figure 20: Behavior of the actuator if moved out of position target range.

a) Before actuator switch-off :
   - Warning: position outside target window (ErrorCode 8500h)
   - Actuator moves back into target window (no position window time restart)
   - If the target is not achieved prior to the position window time having expired, status bit 10 (Target reached) remains at 0, error warning “outside target window (ErrorCode 8500h) disappears and actuator will switch off.

b) After actuator switch-off:
   - Actuator stops moving inside the target window
   - Status bit 10 (Target reached) reset to 0

4.8 Automatic unblocking

The function is to recognize motor block and start autonomous unblocking.
Motor block will be recognized if:

- Actual current is exceeding the max. current in object 6073h. Actual current is subject to a defined filter time in order to prevent premature unblocking.
- Actual velocity (object 606Ch Velocity actual value) drops below 2 rev/sec

After motor block recognition:

1. Setting bit 7 in the status word (Warning active) will give a block warning (Error Code FF05h).
2. Actuator unblocks and travels for free referencing distance “S” (object 2111h -1Eh) in the opposite direction at breakaway current (object 2113h-20h). For safety reasons, presence of current is verified throughout the entire travel to identify and stop repeated blocking. The function’s sensitivity can be defined in the Current Time Filter (object 2111h-05). For safety aspects, the filter time should be kept as short as possible.
   The unblocking time exceeding the unblocking timeout (object 2111h-04h), will abort the unblocking operation and activate an EMCY warning (Error Code FF06h).
   After unblocking and having completed the free referencing travel, a position controller warning (Error Code 8500h) is given in the position window time (object 6968h) > 0. Once the Position window time has expired, bit 7 in the status word is reset to 0 (warning inactive). The warning will disappear once the position window time = 0 and the unblocking operation has been completed.

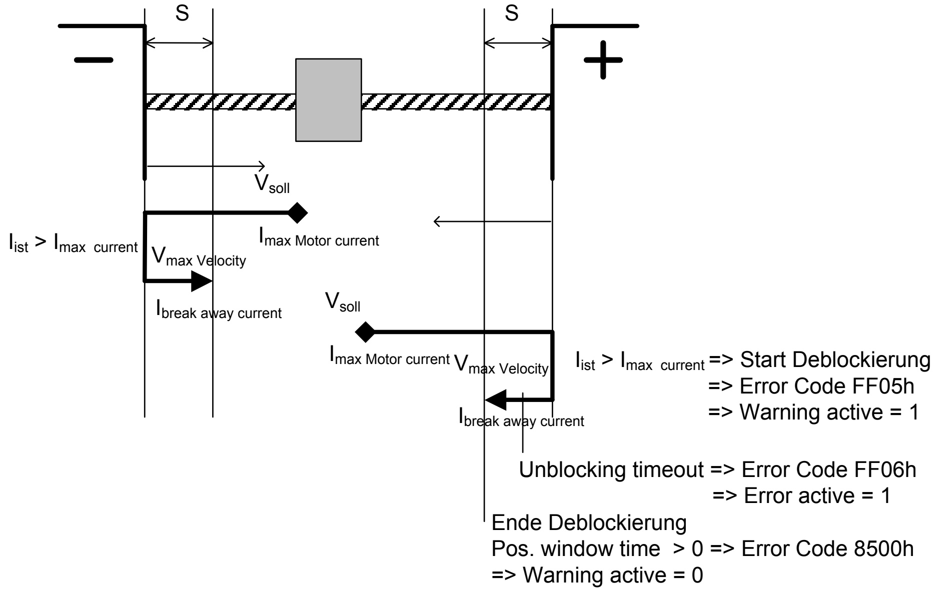
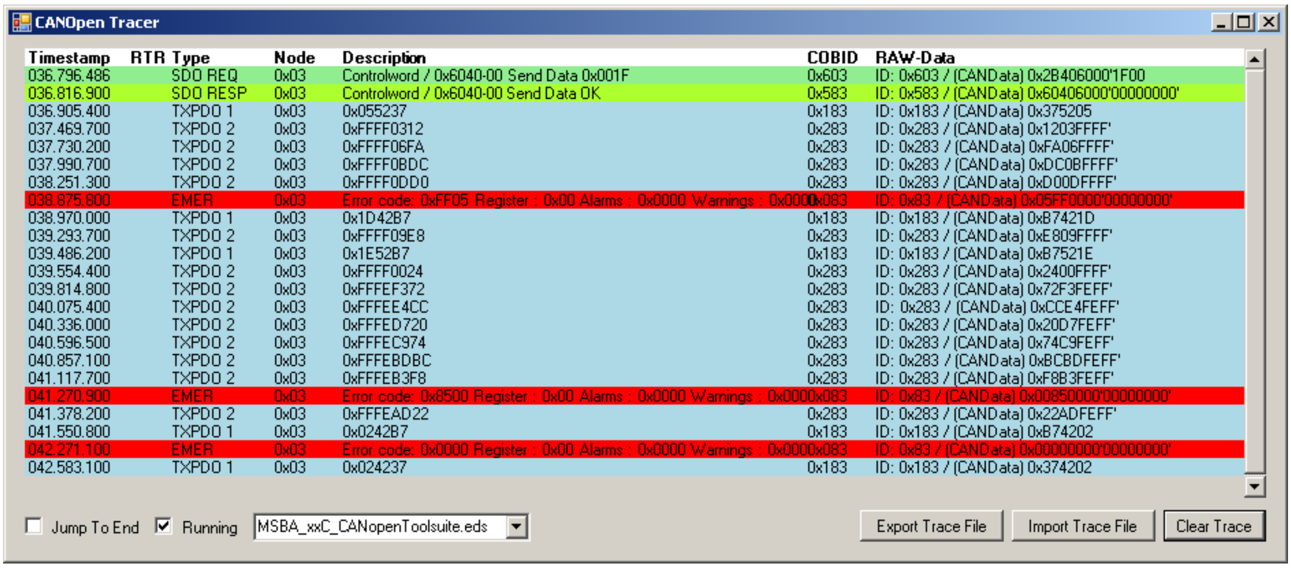

S
S
−
+
V_{soll}
I_{ist} > $I_{max\ current}$
$I_{max\ Motor\ current}$
$V_{max\ Velocity}$
$I_{break\ away\ current}$
V_{soll}
$I_{max\ Motor\ current}$
$V_{max\ Velocity}$
I_{ist} > $I_{max\ current}$ => Start Deblockierung
=> Error Code FF05h
=> Warning active = 1
$I_{break\ away\ current}$
Unblocking timeout => Error Code FF06h
=> Error active = 1
Ende Deblockierung
Pos. window time > 0 => Error Code 8500h
=> Warning active = 0

Figure 1: Unblocking

Activation of automatic unblocking:
    Version control Object 2110h Bit[18] = 0x00040000
Recognition of motor block:
    I_{ist} > $I_{max\ current}$, (Object 6073h)
    Velocity <2 U/sec, (Velocity actual value, Object 606Ch)
Unblocking:
    Max velocity $V_{max\ velocity}$, Object 2113h-0Eh default
    Break away current $I_{break\ away\ current}$ -, Object 2113h-20h default
    Free referencing distance S, Object 2111-1Eh
    Unblocking timeout in ms, Object 2111-4h
    Current time filter measuring rate in ms, Object 2111-5h



Timestamp	RTR	Type	Node	Description	COBID	RAW-Data
036.796.486		SDO REQ	0x03	Controlword / 0x6040-00 Send Data 0x001F	0x603	ID: 0x603 / (CANData) 0x2B406000'1F00
036.816.900		SDO RESP	0x03	Controlword / 0x6040-00 Send Data OK	0x583	ID: 0x583 / (CANData) 0x60406000'00000000'
036.905.400		TXPDO 1	0x03	0x055237	0x183	ID: 0x183 / (CANData) 0x375205
037.469.700		TXPDO 2	0x03	0xFFFF0312	0x283	ID: 0x283 / (CANData) 0x1203FFFF'
037.730.200		TXPDO 2	0x03	0xFFFF06FA	0x283	ID: 0x283 / (CANData) 0xFA06FFFF'
037.990.700		TXPDO 2	0x03	0xFFFF0BDC	0x283	ID: 0x283 / (CANData) 0xDC0BFFFF'
038.251.300		TXPDO 2	0x03	0xFFFF0DD0	0x283	ID: 0x283 / (CANData) 0xD00DFFFF'
038.875.800		EMER	0x03	Error code: 0xFF05 Register : 0x00 Alarms : 0x0000 Warnings : 0x0000	0x083	ID: 0x83 / (CANData) 0x05FF0000'00000000'
038.970.000		TXPDO 1	0x03	0x1D42B7	0x183	ID: 0x183 / (CANData) 0xB7421D
039.293.700		TXPDO 2	0x03	0xFFFF09E8	0x283	ID: 0x283 / (CANData) 0xE809FFFF'
039.486.200		TXPDO 1	0x03	0x1E52B7	0x183	ID: 0x183 / (CANData) 0xB7521E
039.554.400		TXPDO 2	0x03	0xFFFF0024	0x283	ID: 0x283 / (CANData) 0x2400FFFF'
039.814.800		TXPDO 2	0x03	0xFFFEF372	0x283	ID: 0x283 / (CANData) 0x72F3FEFF'
040.075.400		TXPDO 2	0x03	0xFFFEE4CC	0x283	ID: 0x283 / (CANData) 0xCCE4FEFF'
040.336.000		TXPDO 2	0x03	0xFFFED720	0x283	ID: 0x283 / (CANData) 0x20D7FEFF'
040.596.500		TXPDO 2	0x03	0xFFFEC974	0x283	ID: 0x283 / (CANData) 0x74C9FEFF'
040.857.100		TXPDO 2	0x03	0xFFFEBDBC	0x283	ID: 0x283 / (CANData) 0xBCBDFEFF'
041.117.700		TXPDO 2	0x03	0xFFFEB3F8	0x283	ID: 0x283 / (CANData) 0xF8B3FEFF'
041.270.900		EMER	0x03	Error code: 0x8500 Register : 0x00 Alarms : 0x0000 Warnings : 0x0000	0x083	ID: 0x83 / (CANData) 0x00850000'00000000'
041.378.200		TXPDO 2	0x03	0xFFFEAD22	0x283	ID: 0x283 / (CANData) 0x22ADFEFF'
041.550.800		TXPDO 1	0x03	0x0242B7	0x183	ID: 0x183 / (CANData) 0xB74202
042.271.100		EMER	0x03	Error code: 0x0000 Register : 0x00 Alarms : 0x0000 Warnings : 0x0000	0x083	ID: 0x83 / (CANData) 0x00000000'00000000'
042.583.100		TXPDO 1	0x03	0x024237	0x183	ID: 0x183 / (CANData) 0x374202

□ Jump To End ☑ Running MSBA_xxC_CANopenToolsuite.eds

Figure 2: CANopen Tracer blocking log file

5 Error diagnosis

5.1 Fieldbus error

1. If drive does not respond to CAN bus communication, check all electrical connections.

   If electrical connections are correct, check CAN bus operation. For this purpose, use a CAN monitor to display and trace CAN bus communication telegrams.

2. After shutdown and power up of drive supply voltage, drive must send a BootUp message.

   If no BootUp message occurs, check if baudrate of drive and baudrate of CAN monitor are identical.

3. If you have difficulties to connect to a bus participant, check node-ID and baudrate.

   The baudrate must be set identical for all bus participants. The node-ID must be set between 1 and 127. Every bus participant must have a unique node-ID, i.e. more than one bus participant with the same node-ID is not allowed.

5.2 LED status indication

On the backside of the actuator a Duo LED indicates the status (below the sealing cap)

- Permanent green light indicates operation without any warning or errors.
- Flashing green light indicates NMT-status Pre-Operational.
- Permanent orange light indicates bootloader status.
- Flashing red light indicates an active warning.
- Permanent red light indicates an active error.

For detailed interpretation of active errors or warnings it is recommended to check the corresponding objects 1003h or 603Fh.

5.3 Drive error

The drive offers several objects and message for monitoring of drive status or error states:

- Error register (Object 1001h): This object is a register which contains the actual drive error status.
- Object 1003h: In this object, the last eight error and warning codes are saved.
- Object Emergency (80h + Node-ID): High priority error message of a bus participant with Error code and Error register.
- SDO Abort Message: In case of incorrect SDO communication, SDO response contains SDO Abort Code.

Table 16: Error register (Object 1001h) in Emergency message byte 2 (byte 3...7 not used).

Error register	Description
Bit 2 = 1	Voltage error
Bit 3 = 1	Temperature error
Bit 4 = 1	CAN bus communication error
Bit 5 = 1	Device specific error

Table 17: Error code definition for Emergency message byte 0...1 and byte 3...4 (see chapter Emergency service). For last error, see Object 603Fh. For last 8 errors, see Object 1003h.

Error code Byte 0...1	Add. inform. Byte 3...4	Error / warning	Description / action
0000h	0000h	No Error	-
2310h	0000h	Continuous over current	i2t level ≥100% the motor is thermally overloaded
3110h	0000h	Error over voltage bus	Check supply voltage.
3111h	0000h	Error over voltage power electronics	Check supply voltage.
3120h	0000h	Error under voltage bus	Message during drive shutdown. Actual position is saved immediately. If under voltage occurs during active positioning task, a position warning (Error code FF04h) is activated and the bits CalibrationOK and HomingOK are reset.
3121h	0000h	Error under voltage power electronics	Message during shutdown. No motor movement possible. CAN bus communication possible as long as supply voltage at +VsE.
4210h	0000h	Error over temperature	No further drive operation possible.
4210h	0001h	Warning over temperature	Further drive operation possible.
5441h	0000h	Warning minimum hardware limit switch active	Drive movement possible only in positive direction.
5442h	0000h	Warning maximum hardware limit switch active	Drive movement possible only in negative direction.
5530h	0000h	Warning memory (EEPROM)	Error during save or write to memory procedure. Repeat procedure. If warning repeats to occur, check of drive in factory necessary.
6010h	0000h	Warning firmware (watchdog)	Internal drive reset. Further drive operation possible. For safety reasons, check drive parameters. If warning repeats to occur, check of drive in factory necessary.
6320h		object 6093h could not be calculated → enter different positioning parameters	
7121h	0000h	Error motor blocked	Motor eventually blocked, check load, check current limit (Object 6073h).
7121h	0001h	Warning motor blocked	Motor eventually blocked, check load, check current limit (Object 6073h).
7305h		Encoder error	
7320h	0000h	Error encoder	Check of drive in factory necessary.
7510h	0000h	Error internal communication	In case of repeated appearance of the error the actuator needs to be sent back to factory for inspection
8110h	0000h	Warning CAN bus communication warning	Communication data are eventually.
8130h	0000h	Error lifeguard or heartbeat	Check cables. Check bus master function.
8500h	0000h	Warning position controller	Outside target position window

Table 18: SDO abort code definition (see chapter service data communication). In case of an SDO communication error, an abort message is sent, which contains the SDO abort code.

SDO abort code	Description
05040001h	Command byte not supported
06010000h	Incorrect access to Object
06010001h	Read access to write only
06010002h	Write access to read only
06020000h	Object not supported
06090011h	Subindex not supported
06090030h	Value outside of allowed range
08000020h	Incorrect signature during save or load of default parameters
08000021h	Save procedure not finished
08000022h	Target position not set, since drive is moving

5.4 Automatic error reset

The actuator can be operated without PLC by using the digital inputs. In this case it might be required to reset errors appearing at power losses at +VsM or motor blockings. Therefore Error under voltage power electronics (Error Code 3121h) and Error motor blocked (Error Code 7121) can be automatically reset, providing there is no producer heartbeats received. This feature must be activated in version control (Object 2110h) Bit 9: Service operation (Jogging possible without CAN bus). Further Consumer Heartbeat must be configured (with ID of PLC).

6 Wiring for optimum EMC

Note CAN cable specifications

Table 19: Specification of CAN cables.

Total length bus system	< 300 m	< 1000 m
Cable type	LIYCY 2 x 2 x 0,5 mm² (twisted pair with shield is optimum)	CYPIMF 2 x 2 x 0,5 mm² (twisted pair with shield is optimum)
Wire resistance	≤ 40 Ohm/km	≤ 40 Ohm/km
Capacity	≤ 130 nF/km	≤ 60 nF/km
Assignment	Pair 1 (white / brown): CAN-GND and +Vs (for drives, +Vs on separate motor cable only) Pair 2 (green / yellow): CAN-HIGH and CAN-LOW	

- Use only cables with additional cable pair for CAN-GND.
- Disturbance free bus communication is possible with correctly connected CAN-GND.
- Use only shielded cables with shield connected to connecting nut on both sides.

Connect terminating resistors
A terminating resistor of 120 Ohm is required at physical beginning and at physical end of the bus system.

Ensure correct shielding of bus and motor cable

- Connect shield of bus cable on both ends to connector housing or connector nut.
- Connect shield of motor cable on both ends to connector housing. If this is not possible, connect shield to connector on drive side (see Fig. 18).

If a proper earth connection over the drive housing can not be ensured, connect cable shield to earth. If a cable is connected to earth on both ends, guarantee potential equalization by a separate potential equalization cable with sufficient cross section to avoid a ground loop.

Note maximum bus and stub cable length according to Table 21

The maximum allowed total bus cable length and stub line length

- Depend on baudrate
- And can be divided into several segments and single stub cables.

Table 20: Maximum total bus length (with terminating resistor) and maximum stub length (without terminating resistor) as a function of baudrate.

Baudrate [kBit/s]	10	20	50	100	125	250	500	800	1000
Total bus length	5000 m	3000 m	1000 m	500 m	400 m	200 m	75 m	30 m	25 m
Total stub length	1360 m	875 m	350 m	175 m	140 m	70 m	35 m	20 m	17 m
Single stub length	270 m	175 m	70 m	35 m	28 m	14 m	7 m	4 m	3 m

The maximum cable length of a segment according to Table 22 depends on

- used wire cross section and
- number of bus communication participants.

Table 21: Maximum segment length as a function of number of bus communication participants and wire cross section.

Number of bus communication participants	Wire cross section		
	0,75 mm²	0,5 mm²	0,25 mm²
32	550 m	360 m	200 m
64	470 m	310 m	170 m
100	410 m	270 m	150 m

- Divide the maximum total bus length according to Table 21 and use repeaters, if you have to realize a larger segment length than allowed in Table 22.
- The use of repeaters require a reduction of maximum bus length as given in Table 21, which depends on the repeater type. Each repeater requires a typical reduction of maximum total bus length by 30 m.
- The maximum total cable length with repeaters is given by length in Table 21 reduced by the sum of all cable reductions due to repeaters.
  LBus max. = Lmax. (Tab. 21) - (∑ cable length reduction * number of repeaters)
- A repeater is recommended for service interface for disturbance free connection in running bus operation or as measuring interface for galvanic separation of programming device.

Avoid potential difference between CAN network nodes

Potential differences (reference to PE) between nodes of CAN network can cause communication disturbance or can damage CAN bus participants.

Avoid potential differences by

- Connecting each CAN bus participant on the shortest and minimum resistance path to the same earth potential (PE) of the machine/system.
- Using a potential compensation cable between bus communication participants.
- realizing low resistance contact between overall earth and earth contact of machine/system.

Identify EMC disturbances in the signal oscilloscope picture

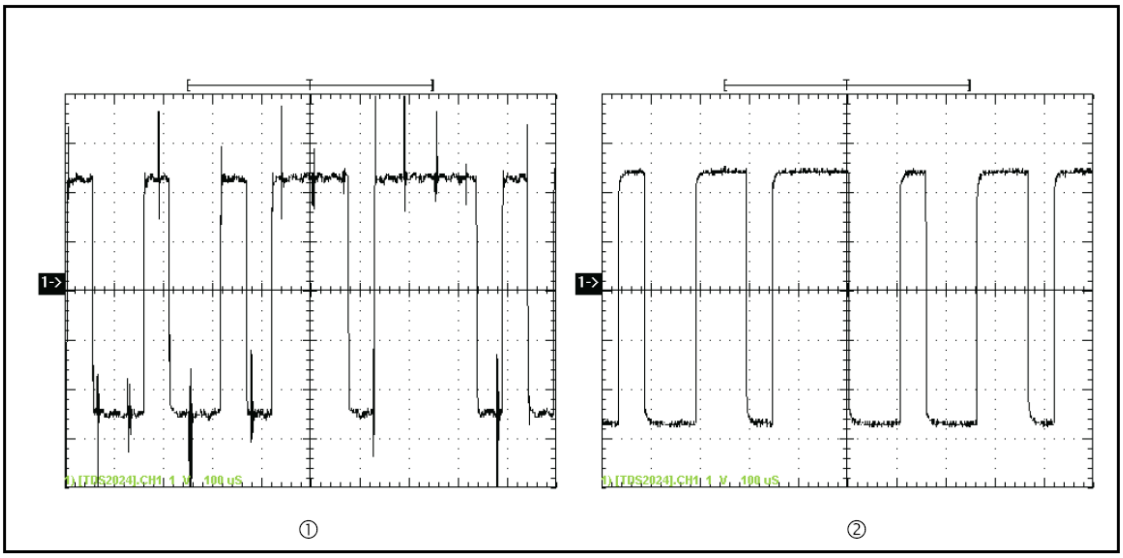


Figure 23: Oscilloscope pictures of CAN signals (1) with and (2) without disturbance peaks (measured points: CAN_HIGH to CAN_LOW).

For quantification of disturbances, measurement with CAN analyzer is required. By using a CAN analyzer, important bus parameters such as bus load or number of error frames can be detected and analyzed.

7 Technical data

7.1 Electrical and mechanical data

See corresponding datasheet.

7.2 Terminal assignment

See corresponding datasheet or product label.

7.3 Block diagram and external inputs

The drive offers two ways to connect the voltage supply. (Figure 20). If the voltage supply is connected to +VsM, the drive is fully working. If the supply voltage connected to +VsM is switched off due to an emergency-stop, the positioning control and fieldbus interface still work, if the voltage supply is additionally connected to +VsE. Drive movement is excluded by hardware in this case.

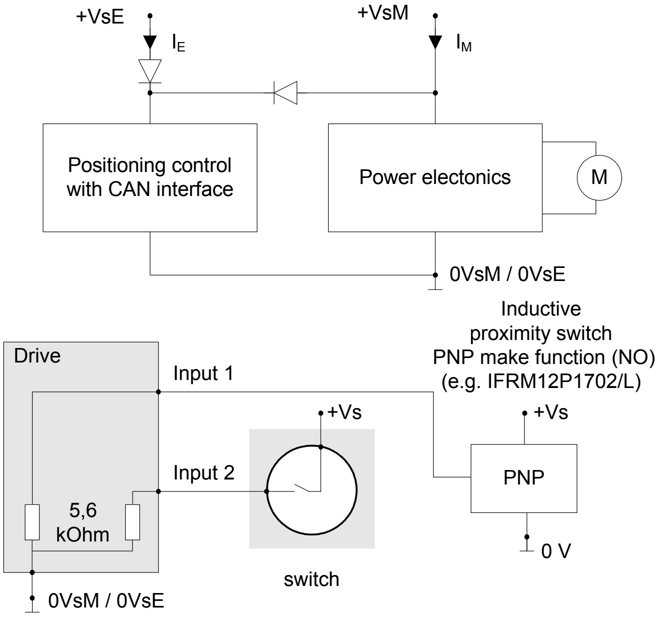

+VsE
I_E
+VsM
I_M
Positioning control with CAN interface
Power electonics
M
0VsM / 0VsE
Inductive proximity switch PNP make function (NO) (e.g. IFRM12P1702/L)
Drive
Input 1
+Vs
+Vs
Input 2
PNP
5,6 kOhm
0 V
switch
0VsM / 0VsE

Figure 24: Block diagram of drive and example for connecting a sensor or switch to external inputs.

In addition, the drive offers two programmable digital inputs (Object 2111h-0Dh) to connect standard position sensors. Possible programming functions are e.g.:

- Activate Jogging (Jogging+, Jogging-)
- Hardware limit switch or referencing switch
- Control of controlword for e.g. access to traversing blocks (Object 6040h)

7.4 Dimensions

See corresponding datasheet.

7.5 Command sequence examples

Table 24: Command sequence to identify drive.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Upload Request (Get Software version)	100Ah	00000000h	SDO Upload Response	xxxxxxxxh
SDO Upload Request (Get Identity object, Product code)	1018h-02h	00000000h	SDO Upload Response	00000101h
SDO Upload Request (Get Identity object, Revision number)	1018h-03h	00000000h	SDO Upload Response	xxxxxxxxh
SDO Upload Request (Get Identity object, Serial number)	1018h-04h	00000000h	SDO Upload Response	xxxxxxxxh

Table 25: Command sequence to set application parameters.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Download Request (Set Maximum current)	6073h	03E8h	SDO Download Response	00000000h
SDO Download Request (Set Polarity)	607Eh	00h	SDO Download Response	00000000h
SDO Download Request (Set Position notation)	6089h	FDh	SDO Download Response	00000000h
SDO Download Request (Set Position dimension)	608Ah	01h	SDO Download Response	00000000h
SDO Download Request (Set Velocity notation)	608Bh	00h	SDO Download Response	00000000h
SDO Download Request (Set Velocity dimension)	608Ch	A3h	SDO Download Response	00000000h
SDO Download Request (Set Position encoder resolution, Encoder increments)	608Fh-01h	00000004h	SDO Download Response	00000000h
SDO Download Request (Set Position encoder resolution, Encoder revolutions)	608Fh-02h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Gear ratio, Motor revolutions)	6091h-01h	00000012h	SDO Download Response	00000000h
SDO Download Request (Set Gear ratio, Gear shaft revolutions)	6091h-02h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Feed constant, Feed)	6092h-01h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Feed constant, Spindle shaft revolutions)	6092h-02h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Software position limit, Minimum software position limit)	607Dh-01h	00000000h	SDO Download Response	00000000h
SDO Download Request (Set Software position limit, Maximum software position limit)	607Dh-02h	00000000h	SDO Download Response	00000000h
SDO Download Request (Set Positioning parameter, Backlash compensation)	2111h-01h	00h	SDO Download Response	00000000h
SDO Download Request (Set Positioning parameter, Backlash delta)	2111h-02h	00000064h	SDO Download Response	00000000h
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h

Table 26: Command sequence to set communication parameters.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Download Request (Set Receive PDO1 parameter, R_PDO1 type)	1400h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Receive PDO2 parameter, R_PDO2 type)	1401h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO1 parameter, T_PDO2 type)	1800h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO1 parameter, T_PDO1 Event Timer)	1800h-05h	0203h	SDO Download Response	00000000h
SDO Download Request (Set T_PDO1 add-on)	2800h	01h	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO2 parameter, T_PDO2 type)	1801h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO2 parameter, T_PDO2 Event Timer)	1801h-05h	0100h	SDO Download Response	00000000h
SDO Download Request (Set T_PDO2 add-on)	2801h	01h	SDO Download Response	00000000h
SOD Download Request (Set Consumer heartbeat time)	1016h-01h	yyyyzzzzh	SDO Download Response	00000000h
SDO Download Request (Set Producer heartbeat time)	1017h	xxxxh	SDO Download Response	00000000h
SDO Download Request (Store communication parameters)	1010h-02h	65766173h	SDO Download Response	00000000h

Table 27: Switch from Ready mode to PositioningReady mode (only one time after power-on or reset of drive).

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Table 28: Jogging in positive direction (drive in PositioningReady mode).

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Download Request (Set Controlword)	6040h	081Fh	SDO Download Response	00000000h

Table 29: Jogging in negative direction (drive in PositioningReady mode).

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Download Request (Set Controlword)	6040h	101Fh	SDO Download Response	00000000h

Table 30: Positioning via SDO (drive in PositioningReady mode).

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
NMT Cmd (Start Node, only once after power-on / reset)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Target position)	607Ah	xxxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	4237h
-			T_PDO1	5237h
-			T_PDO2	xxxxxxxxh
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	5237h
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4637h

Table 31: Positioning via R_PDO (drive in PositioningReady mode).

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
NMT Cmd (Start Node, only once after power-on / reset)		01h	(T_PDO1, TPDO2)	
R_PDO2 (Set Target position + Controlword)		xxxxxxxxh 001Fh	T_PDO1	5237h
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	4237h
-			T_PDO2	xxxxxxxxh
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	5237h
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4637h

Table 32: Stop positioning task.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Download Request (Set Controlword)	6040h	011Fh	SDO Download Response	00000000h

Table 33: Error acknowledge.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
SDO Download Request (Set Controlword)	6040h	008Fh	SDO Download Response	00000000h

Table 34: Referencing with preset value.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	F4h	SDO Download Response	00000000h
SDO Download Request (Set Target position)	607Ah	xxxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			(T_PDO1, T_PD02)	
-			T_PDO2	xxxxxxxxh
-			T_PDO1	4237h
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Table 35: Referencing to referencing switch in negative direction.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Positioning parameter, Input1)	2111h-0Dh	12h	SDO Download Response	00000000h
SDO Download Request (Set Positioning parameter, Reference Position)	2111h-11h	xxxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	E2h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			T_PDO1	1237h
-			T_PDO2	xxxxxxxxh
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4237h
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Table 36: Referencing to block in negative direction.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Positioning parameter, Reference Position)	2111h-11h	xxxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Maximum current)	6073h	01F4h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	D8h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			T_PDO1	1237h
-			T_PDO2	xxxxxxxxh
-			(T_PDO1, T_PDO2)	
-			Emergency Message	7121h
-			Emergency Message	0000h
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4237h
-			T_PDO2	xxxxxxxxh
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Table 37: Referencing with measuring method.

Master PLD to slave MSQA	Object	Value	Slave MSQA to master PLD	Value
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Target position)	607Ah	xxxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	F3h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			T_PDO1	4337h
-			T_PDO2	xxxxxxxxh
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h