CIRO PRODUCTS LTD
Caring Beyond Our Products

Motor Controller

Type BCD130-CAN

V40.04

- **for 2-Phase Step Motors**
- **CanOpen DS301**
- **pulse / direction interface**

Technical Documentation

MIDDEX
ELECTRONIC

Version 10061

1. Introduction

The BCD130-CAN motor drivers are speed and direction controllers for 2-phase step motors. The four-quadrant current regulator in connection with resonance dampening allows an exceptionally low noise and very low resonance step motor operation within its entire rpm range.
The position and rpm control is achieved with one of the latest generation of signal processors. The digital signal processing permits easy adaptation and expansion to customer specific needs.
Latest MOS transistors output amplifiers allow a noiseless operation with an excellent efficiency factor.

Position and rpm data is transmitted via CAN bus interface (DS301). These values may be transmitted while the motor is running, without a need to stop the axis. The reference run is approximated to the DC402 profile. The ramp profile, acceleration, motor current, and other data are parametrically represented via SDOs (service data objects).

The unit mounts with two screws to the panel wall.

Several protection functions guarantee a high operation dependability. The units is protected against:
- Short between phases
- Over and under voltage
- Exceeding temperature

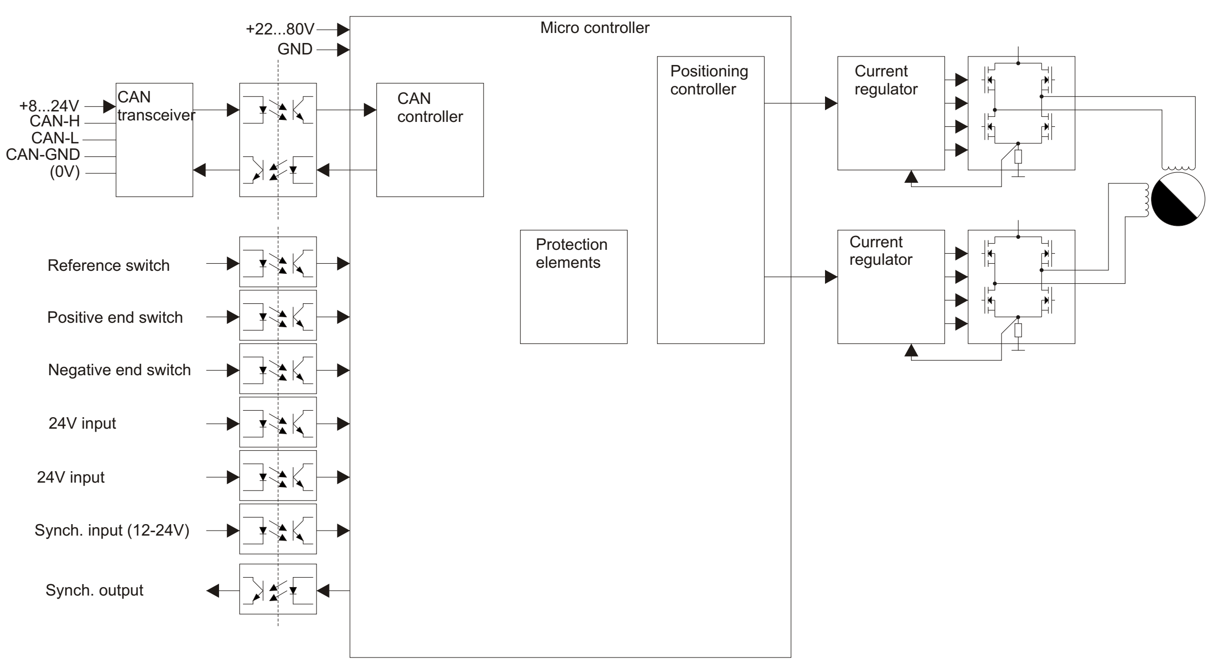
All errors are displayed by LED lights on the front plate.

From version 40.04 positioning and pulse control was added, besides the CanOpen interface.

2. Data Overview

- Supply voltage 22...80VDC
- Motor current up to 6Aeff (momentary 8Aeff)
- Step resolutions up to 25,600 steps per revolution
- Low heat generation due to optimized output amplifiers
- CanOpen interface to DS301, electrically isolated
- Parametric input with SDOs, standard settings however are suitable for many applications.

3. Block Diagram



4. Control Elements and LEDs

Jumper for terminating resistor
Up: terminating resistor off
Down: terminating resistor on

CanOpen interface

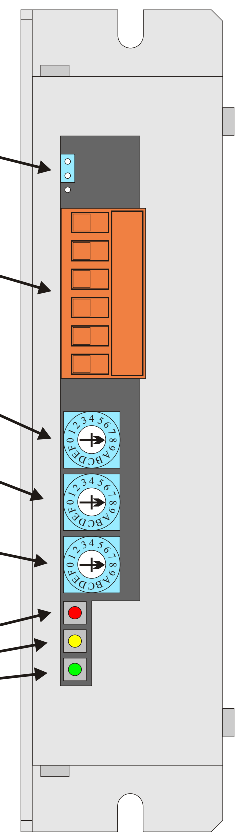
Node ID, Bit4...7
- Acceptable values 0...7 (**X**0h...**X**Fh)

Node ID, Bit0...3
- Acceptable values 0...F (0**X**h...7**X**h)

CanOpen Baud rate

- 0	20kBaud
- 1	50kBaud
- 2	125kBaud
- 3	250kBaud
- 4	500kBaud
- 5	800kBaud
- 6	1MBaud
- 7...F	Pulse / direction interface For step resolution see section 10, page 25

LEDs
- Operational state of CanOpen
- Error state (voltage, current, temperature)
- See description below



4.1 LEDs

CanOpen conditions	LED green	LED yellow	LED red
Power On	off	on	off
Initialization	off	flashing	off
Pre-operational (no errors)	flashing	flashing	off
Pre-operational (over current)	flashing	flashing	on
Pre-operational (under voltage)	flashing	off	flashing
Pre-operational (over voltage)	off	flashing	flashing
Pre-operational (over temperature)	flashing	flashing	flashing
Pre-operational (communication error)	off	off	on
Operational	on	off	off
Stopped	flashing	alternating flashing	off

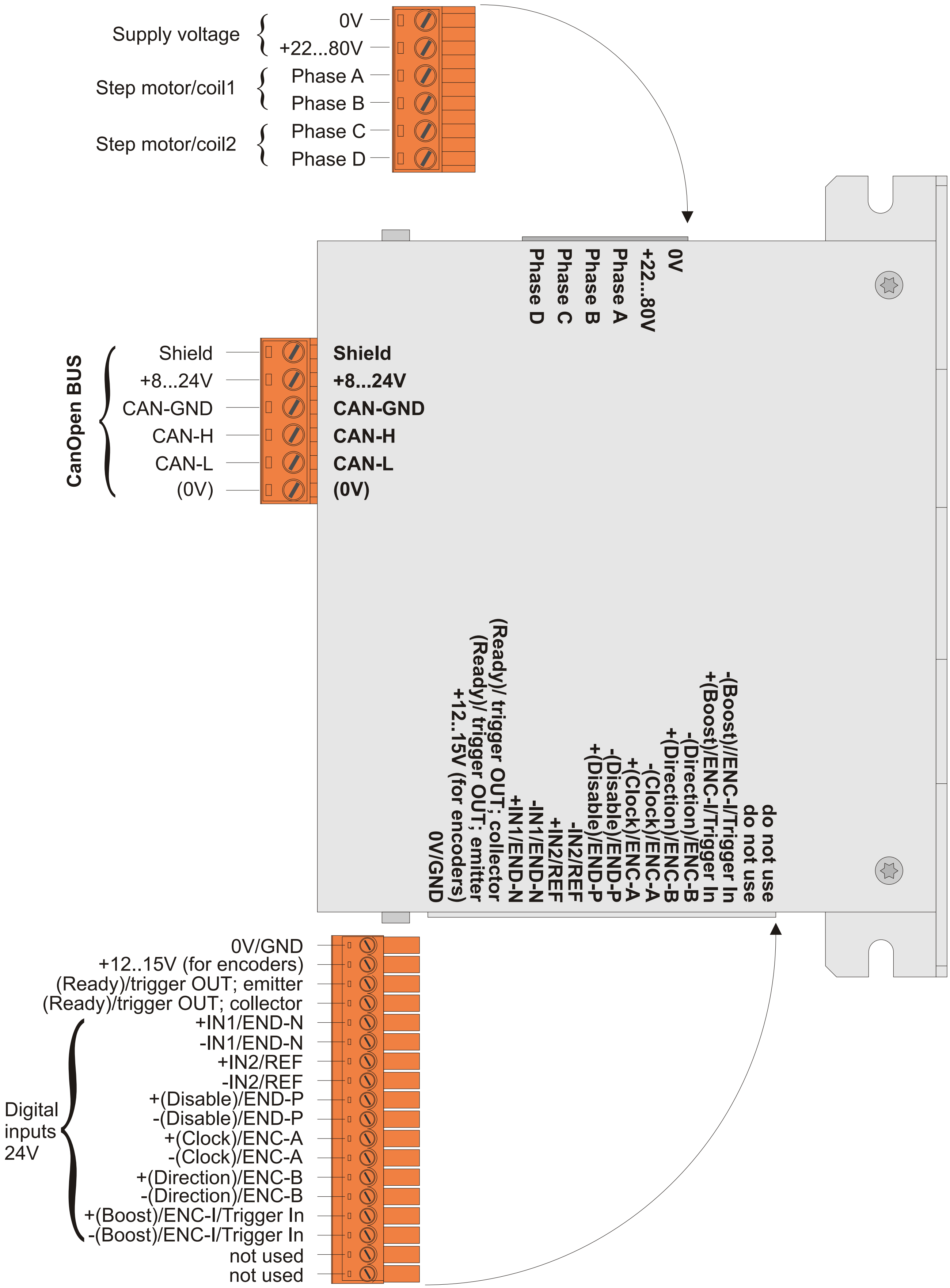
4.2 Address

The node id is selected with the Hex switches Addr-High (upper) and Addr-Low (middle). The range extends from 01h to 7Fh. The selections are acquired after powering the unit up.

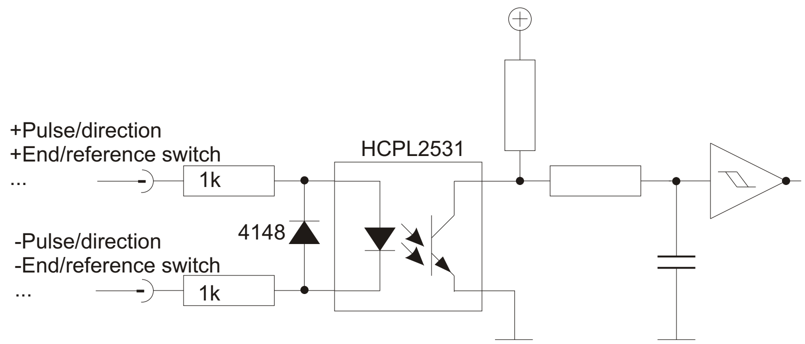
4.2 Baud rate

The Baud rate is selected with the bottom Hex switch. The selection is acquired after powering the unit up.

5. Connections



5.1 Digital inputs



5.1.1 PNP connection (outputs switching to 24V)

The negative input is at 0V/GND at the controller and the positive input is connected with the controller output. This is a common connection with European made controls.

5.1.2 NPN connection (outputs switching to 0V)

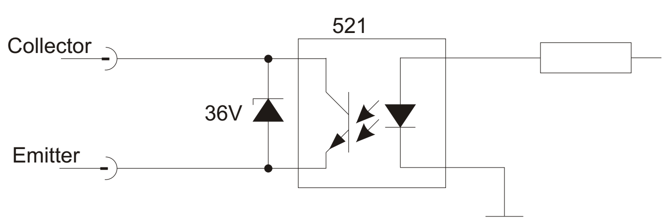
The positive input is at 24V at the controller and the negative input is connected with the controller output. This is a common connection with Asian made controls.

5.1.3 Differential control

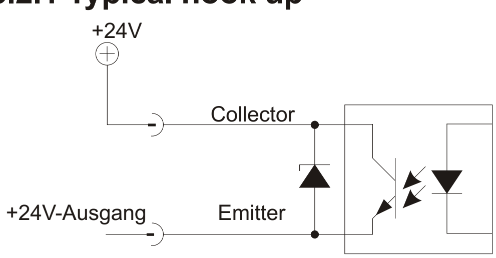
The positive and negative inputs are connected with the corresponding outputs of the controller or encoder. The differential control offers an extremely high interference safety and is used mainly with encoders and sensors.

Please Note: Always connect the positive and the negative input. Both signals are wired together to the controller and hooked up there.

5.2 Digital outputs



5.2.1 Typical hook up

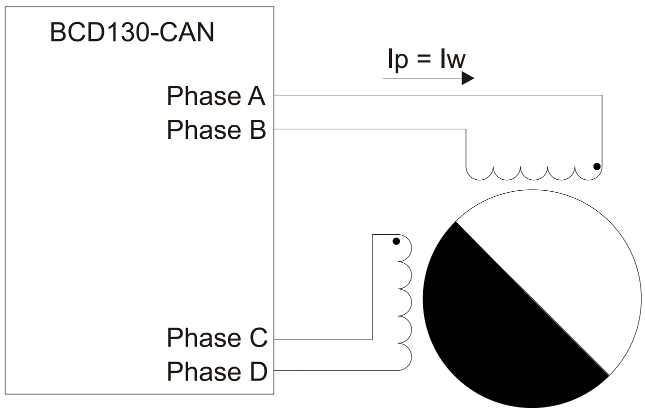


The opto-decoupler outputs at the output are switching potential-free and do not have a conducting connection to the supply voltage of the BCD130-CAN. The voltage connected at the collector will show as output voltage at the emitter.

6. Step Motors

The BCD130-CAN is made for 2-phase step motors with either 4, 6, or 8 leads and a winding inductance of at least 0.75mH. The best results are found normally with motors with 4 or 8 leads.

6.1 Step motors with 4 leads

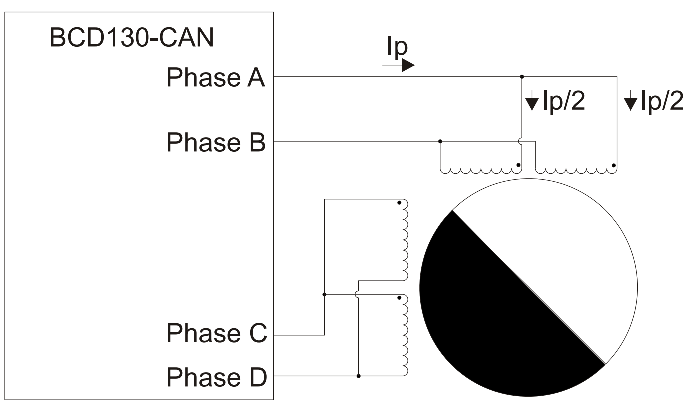


Iw: Nominal current per phase

Ip: Phase current BCD130-CAN = Iw

Example: A step motor with 4 leads is listed with a nominal current of Iw = 4.0A. The BCD130-CAN may be set to a maximum continuous current of up to 4.0A.

6.2 Step motors with 8 leads (parallel)

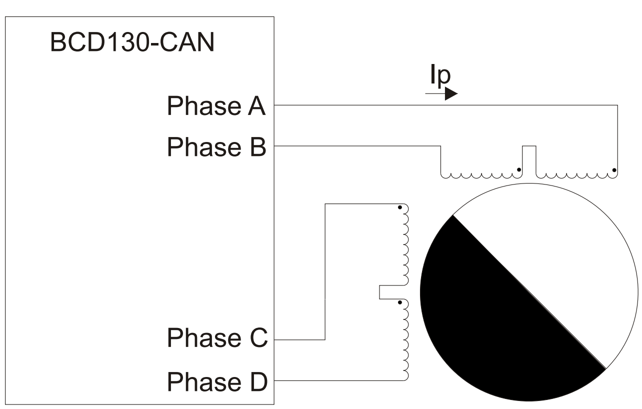


Iw: Nominal current for one coil per phase

Ip: Phase current BCD130-CAN = 1.41 x Iw

Example: For a step motor with 8 leads both coils of a phase are connected in parallel. The maximum continuous current in one coil (Iw) is 4.0A. The BCD130-CAN may be set to a maximum continuous current of up to 5.6A.

6.3 Step motors with 8 leads (serial)

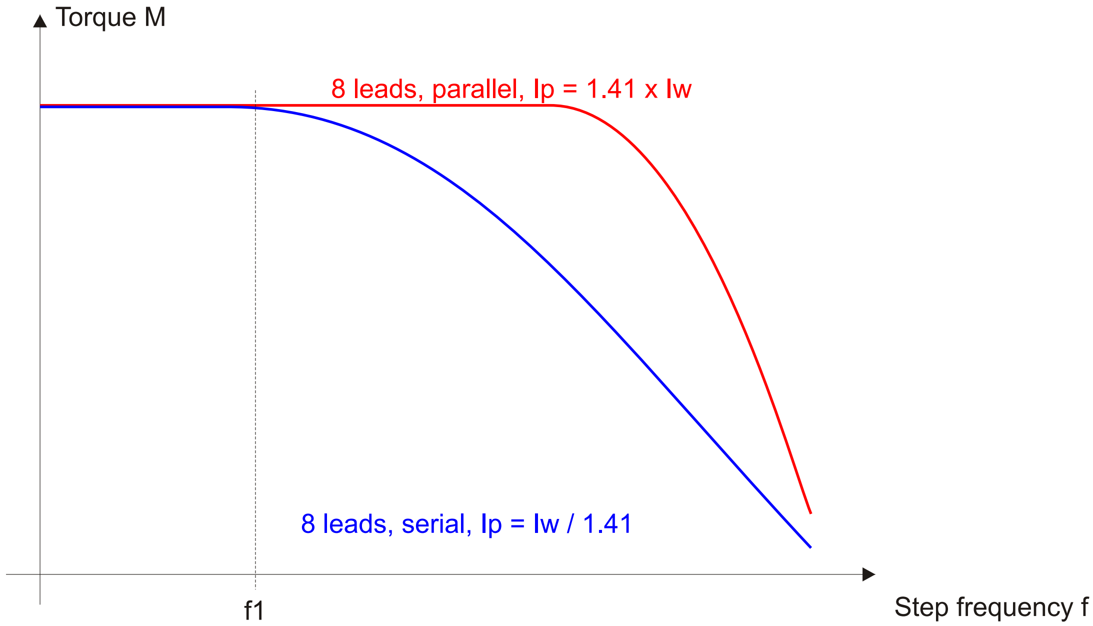


Iw: Nominal current for one coil per phase

Ip: Phase current BCD130-CAN = Iw / 1.41

Example: For a step motor with 8 leads both coils of a phase are connected in series. The maximum continuos current in one coils (Iw) is 4.0A. The BCD130-CAN may be set to a maximum continuous current of up to 2.8A.

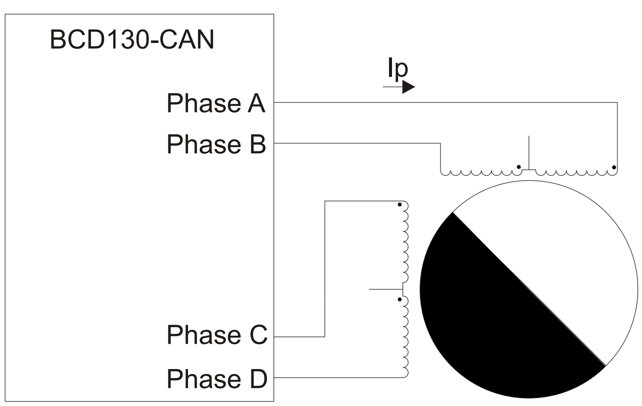
6.4 Parallel or Serial Connection of Motors with 8 Leads



Normally, step motors with 8 leads are connected in parallel. This realizes the maximum possible torque within the whole rpm range. The step driver however has to put out a phase current of 1.41 x Iw.

If the upper rpms past f1 are not used then the two coils of a phase should be connected in series. The step driver in comparison to the parallel connected coils has to deliver only half of the phase current and may be therefore dimensioned smaller.

6.5 Step motors with 6 leads

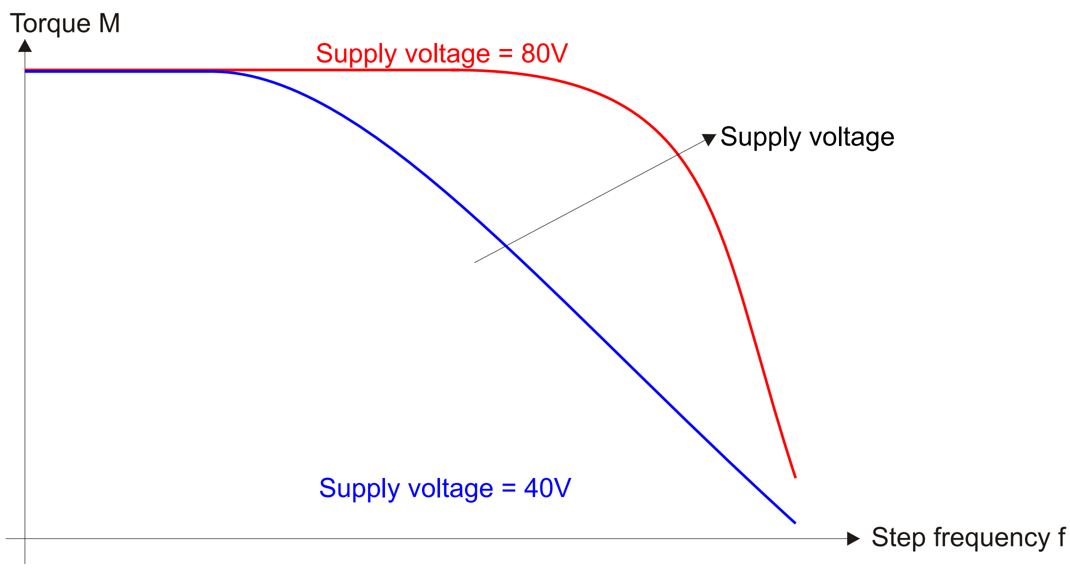


Iw: Nominal current with one coil per phase

Ip: Phase current BCD130-CAN = Iw / 1.41

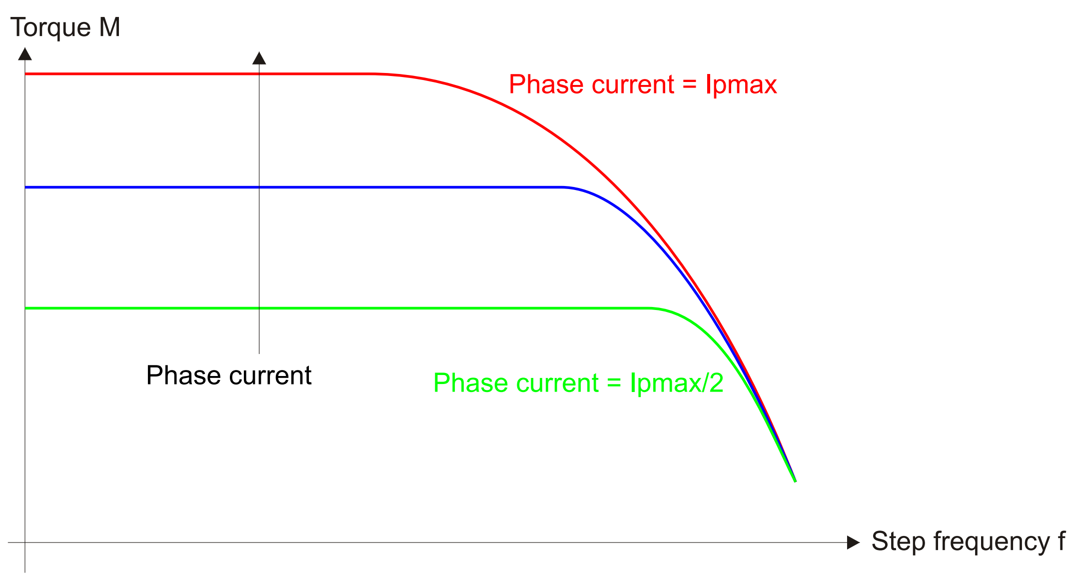
Example: A step motor with 6 leads, both middle connections not connected. The maximum continuous current in one coil is 4.0A. The BCD130-CAN may be set to a maximum continuous current of up to 2.8A.

6.6 Dependence of torque from the supply voltage



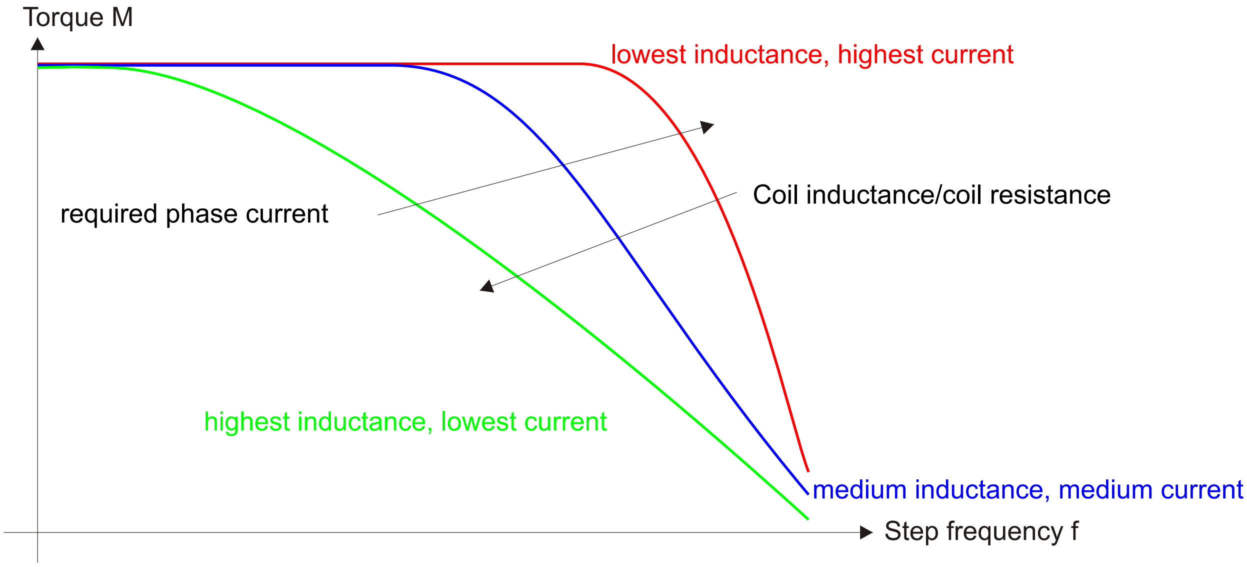
The supply voltage level affects the torque in the upper rpm region.

6.7 Dependence of torque from the phase current



The current level affects the torque in the lower rpm region.

6.8 Same frame size step motors with different windings



Motors with identical frames are often available with different coils. Motors become more dynamic with lower inductance and are able to supply a higher torque at higher rpms. However, the lower the inductance gets the more phase current is needed.

7. CanOpen

Notations:
The address of the CanOpen node (BCD130-CAN) is called the **node address** or **node id** and is set at the driver's front plate..
The address of a Can telegram is called **COB ID**.
The BCD130-CAN is a Can node or also called a CanOpen slave, or simply a slave.
The overriding controller is the CanOpen master, or master.
SDOs are service data objects which are in general transmitted during system start up.
PDOs are process data objects which are cyclic or event driven transmitted.

7.1 Minimum requirements of a master (overriding controller)

A telegram containing 01h 00h 00h 00h 00h 00h 00h 00h and address (COB-ID) 0 is required from the master in order to bring all nodes into the "operational" mode. After the BCD130-CAN received this telegram only the green LED is on.
Now the driver is ready to receive control telegrams with target position, rpm, and command to address (COB ID) = 200h + node address. The node address (1...7Fh) is selectable with two Dip switches on the front plate.

7.2 RPDO1 (controller >> BCD130-CAN)

- Target position : Lead position (byte0...3)
  index: 2001h, sub-index 00h, 4 byte, with sign bit

- Target rpm : Lead rpm (byte4...6)
  index: 2001h, sub-index 01h, 3 byte, without sign bit
  A value from 10...200 000 translates into a step frequency from 1...200 000 Hz
  For command 3 (rpm) a value of 0 (zero) is allowed.

- Command : Command to be executed (byte7)

7.3 TPDO1 (BCD130-CAN >> controller)

- Actual position : Momentary position of drive (byte0...3)
  index: 2002h, sub-index: 00h, 4 byte, with sign bit
- Actual rpm : Momentary rpm / 10 (Byte4...5)
  index: 2002h, sub-index: 01h, 2 byte, without sign bit
  A value from 10...200 000 translates into a step frequency from 10...200 000 Hz
- Executed command : Command currently executed (byte6)
  index: 2002h, sub-index: 02h, 1 byte, without sign bit
- Actual status : Status of command execution (byte7)
  index: 2002h, sub-index: 03h, 1 byte, without sign bit

If the "transmission type" for the TPDO1 (index1800h, sub02h) is set to 255 or 254, the TPDO1 will be transmitted if the executed command or actual status has changed. For example, this is the case when a new command was transmitted (RPDO1) or the command execution was finished.

7.4 RPDO2 (slave/controller >> BCD130-CAN)

The receipt of a RPDO2 with trigger commands via RPDO2 leads to command start. A data length from 0 to 8 bytes is accepted. The received data is not evaluated.

7.5 TPDO1 (BCD130-CAN >> BCD130-CAN)

A TPDO2 is sent when the actual step position changes to the value of the trigger position. The preceding step position may be smaller or larger then the trigger position.
The TPDO2 is sent with a data length of 1 byte.

7.6 Command (index 2001h, sub-index2)

Byte 7 of the RPDO1 transmits the command.

Command 0: **No function, NOP, No Operation**
The active command is not interrupted.

Command 1: **Absolute positioning**
The target position is transferred with byte 0...3.
The maximum step frequency is given in byte 4...6.
The absolute position and step frequency may be changed while the motor is running without changing the command in byte 7.
If the axis contacts the end switch the motor stops with the QickStop ramp.
In all cases the set ramp is followed.

Command 2: **Relative positioning** (based on the actual target position)
The value in byte 0...3 is added to the **target** position.
The maximum step frequency is listed in byte 4...6.
The new target value is written when the command in byte 7 changes to 2. Prior to be able to write a new relative value, the command needs to be changed to a different value (e.g. 0, NOP, No Operation).
If the axis contacts the end switch the motor stops with the QuickStop ramp.
Command 84 is used to change the target position relative to the **actual** position.

Command 3: **Rpm**
The maximum step frequency is set by the byte 4...6. The motor runs with this frequency until another frequency or command is received. The frequency may be changed while the motor is running without the command being changed. The running direction is controlled with the sign bit in byte 0...3. The ramp values are kept.
If the axis contacts the end switch the motor stops with the QickStop ramp. If the count (+/- 1 000 000 000) of the position counter is passed, a bit is set in the condition byte (current status), however, the motor continues to run.
Only the operational sign is used from the data in byte 0...3 (turning direction)
Attention: If a relative positioning in the same direction is executed during the current command run, the overflow bit for the position counter is set. To stop after a defined step count, the command “relative positioning to the current position” (# 84) is used.

Command 4: **Stop**
The motor stops immediately with the defined ramp. The time used to ramp down from the maximum frequency to the start frequency (stand still) corresponds to the acceleration time (index2000; sub-index 5).

Command 5: **QuickStop**
The motor stops immediately with the defined ramp. The time used to ramp down from the maximum frequency to the start frequency (stand still) corresponds to the acceleration time (index2000; sub-index 7). Usually, the delay time for the QuickStop is set shorter then the regular acceleration time.

Command 6: **Reference run**
A reference run is initiated. The actual position is set to 0 if the command was successful. If the command is interrupted, the actual position is immediately set to 0. The motor will continue with the new command. With the command NOP the motor stops and continues to run with the Ref-In frequency to the 0 position.
The reference run is started when the command in byte 7 changes to 6. After a successful reference run, the command value needs to be changed (i.e. 0, NOP, No Operation) before a new run is started.

Command 7: **Motor on**
The motor current is switched on. The target position is set to the actual position so that the motor is held at that position with the holding moment (Standby current).

Command 8: **Motor off**
The motor current is switched off. The reference position is lost. Normally a new reference run is needed.

Command 9: **Set target position**
The actual and target positions are written in byte 0..3.
Should the motor run at the moment it will stop. Then it will assume the set position.
The actual new position is written, when the command in byte 7 changes to 9. If the actual position needs changing again, the prior command value needs to be changed to a different value (i.e. 0, NOP, No Operation).

Command 10: **Immediate stop**
This command corresponds to the QuickStop. This stop is used with servo drives with position feedback to stop with the defined motor current as quickly as possible.

Command 11: **Error reset**
This command is not used with the BCD130-CAN. With servo drives, this command resets a contouring error. This command is accepted due to compatibility reasons, and the bit 0/bit1 are immediately set in the current status.

Command 32: to
Command 47: Reserved for customer specific expansions

Synchronization

Command 48: **Set trigger positions**
A new trigger position is transmitted with byte 0...3. The data in byte 4...6 is not evaluated. The trigger position may also be set via SDO.

Command 49: **Set trigger delay**
A new trigger delay value is transmitted with the byte 0...3. A value larger than 0 delays a command start (in ms) after a valid signal or a RPDO2 is received at the trigger input. The trigger delay may also be changed with a SDO.

Command 55: **Absolute positioning with trigger via RPDO2**
As with command 1, the target position will be set only after a RPDO2 was received and the trigger delay has passed.

Command 56: **Absolute positioning with trigger via digital input (trigger input high)**
As with command 1, the target position will be set only after a high signal is received at the trigger input and the trigger delay has passed.

Command 57: **Absolute positioning with trigger via digital input (trigger input low)**
As with command 1, the target position will be set only after a low signal is received at the trigger input and the trigger delay has passed.

Command 58: **Absolute positioning with trigger via digital input (trigger input, positive edge)**
As with command 1, the target position will be set only if the level at the trigger input changes from low to high and the trigger delay has passed.

Command 59: **Absolute positioning with trigger via digital input (trigger input, negative edge)**
As with command 1, the target position will be set only if the level at the trigger input changes from high to low and the trigger delay has passed.

Command 65: **Relative positioning with trigger via RPDO2**
As with command 2, the target position will be set only after a RPDO2 was received and the trigger delay has passed.

Command 66: **Relative positioning with trigger via digital input (trigger input high)**
As with command 2, the target position will be set only after a high signal is received at the trigger input and the trigger delay has passed.

Command 67: **Relative positioning with trigger via digital input (trigger input low)**
As with command 2, the target position will be set only after a low signal is received at the trigger input and the trigger delay has passed.

Command 68: **Relative positioning with trigger via digital input (trigger input, positive edge)**
As with command 2, the target position will be set only if the level at the trigger input changes from low to high and the trigger delay has passed.

Command 69: **Relative positioning with trigger via digital input (trigger input, negative edge)**
As with command 2, the target position will be set only if the level at the trigger input changes from high to low and the trigger delay has passed.

Command 75: **Rpm with trigger via RPDO2**
As with command 3, the **rpm** will be set only after a RPDO2 was received and the trigger delay has passed.

Command 76: **Rpm with trigger via digital input (trigger input high)**
As with command 3, the rpm will be set only after a high signal is received at the trigger input and the trigger delay has passed.

Command 77: **Rpm with trigger via digital input (trigger input low**
As with command 3, the rpm will be set only after a low signal is received at the trigger input and the trigger delay has passed.

Command 78: **Rpm with trigger via digital input (trigger input, positive edge)**
As with command 3, the rpm will be set only if the level at the trigger input changes from low to high and the trigger delay has passed.

Command 79: **Rpm with trigger via digital input (trigger input, negative edge)**
As with command 3, the rpm will be set only if the level at the trigger input changes from high to low and the trigger delay has passed.

Command 84: **Relative position to actual position**
The value in byte 0...3 is added to the current actual position.
The maximum step frequency is given in byte 4...6
The target position is written when the command in byte 7 changes to 84. Prior to be able to write a new relative value, the command needs to be changed to a different value (e.g. 0, NOP, No Operation).
If the axis contacts the end switch the motor stops with the QickStop ramp.

Command 85 **Relative position to actual position with trigger via RPDO2**
As with command 84, the target position will be set only after a RPDO2 was received and the trigger delay has passed.

Command 86: **Relative position to actual position with trigger via digital input (trigger input high)**
As with command 84, the target position will be set only after a high signal is received at the trigger input and the trigger delay has passed.

Command 87: **Relative position to actual position with trigger via digital input (trigger input low)**
As with command 84, the target position will be set only after a low signal is received at the trigger input and the trigger delay has passed.

Command 88: **Relative position to actual position with trigger via digital input (trigger input, positive edge)**
As with command 84, the target position will be set only if the level at the trigger input changes from low to high and the trigger delay has passed.

Command 89: **Relative position to actual position with trigger via digital input (trigger input, negative edge)**
As with command 84, the target position will be set only if the level at the trigger input changes from high to low and the trigger delay has passed.

7.7 Actual status (index 2002h, sub-index3)

Byte 7 of the TPDO1 returns the actual status back to the controller.

Bit0: The axis executes a command
Bit1: The command execution is finished
Bit2: The axis ran against an end witch
Bit7: Overflow of position counter

In the normal execution of a command first the Bit0 is set. "Actual status" = 1 indicates that the command is in actual execution. For a command with a trigger function the Bit0 is set only after the trigger event (i.e. RPDO2, digital input).
If the command was successfully executed in addition the Bit1 is set. “Actual status” = 3 indicates that the command was executed without error.
If the “actual status” exceeds 3 it indicates a fault.
Bit2 indicates that the motor axis ran against an end switch. A command to move in opposite direction erases Bit2.
Bit3 indicates position counter overflow (>1 000 000 000 or < -1 000 000 000). The actual position therefore does no longer contain a defined value and the reference to the reference position is lost. This may happen when the axis is used in one direction only and if after the command “rpm” a relative positioning in the same direction is executed.

7.8 CanOpen conditions and error management

After switching the supply power on, the Can-node (slave) goes through initialization and reaches the “pre-operational” state. After the Can-node boot-up information is sent, configuration through SDO transfers takes place.
The normal operation mode is reached when the NMT master (controller) declares the Can-node state “operational” (COB-ID = 0; Byte0 = 1; Byte1 = node-ID or 0/all; Byte2-7 = 0). At the same time all errors in the error register (index 1001h; sub-index 0) are erased.

The Can-node status is shown with two LEDs:

If an error occurs the motor current is switched off and the Can-node changes to the “pre-operational” state. The error register (index 1001h, sub-index0) is written. The last five errors are stored in the object register (index 1003h, sub-index 1-5). Index 1003h, sub-index 0 contains the amount of the last errors. The Can-node sends an emergency telegram.

The following errors are recognized:

A) Over current (LED green flashing, yellow flashing, red flashing):
Index 1001h, sub-index 0: Bit 0 (general error) and Bit 1 (current error) are set
Index 1003h, sub-index 1-5 = 2300h

This error resets as soon as the Can-node is brought back to the “operational” state.

B) Voltage error (green and yellow LED flashing together, red alternating flashing):
Index 1001h, sub-index 0: Bit 0 (general error) and Bit 2 (voltage error) are set
Index 1003h, sub-index 1-5 = 3100h

This error resets as soon as the Can-node is brought back to the “operational” state.

C) Over temperature (LED green flashing, yellow flashing, red alternating flashing):
Index 1001h, sub-index 0: Bit 0 (general error) and Bit 3 (temperature error) are set
Index 1003h, sub-index 1-5 = 4200h

This error resets as soon as the Can-node is brought back to the “operational” state.

D) Guarding error (LED green flashing, yellow flashing, red constantly on) :
Index 1001h, sub-index 0: Bit 0 (general error) and Bit 4 (guarding error) are set
Index 1003h, sub-index 1-5 = 8130h

This error resets as soon as the Can-node is brought back to the “operational” state.

Every error triggers an emergency telegram.
The emergency information consists of:

Byte0, Byte1:	EMCY error code from object 1003h, sub-index 1-5
Byte2:	object 1001h
Byte3-7:	0

To get back into the normal operational mode, the master needs to bring the Can-node back into the “operational” state. At the same time all errors in the error registry (index 1001h) are erased.

7.9 Node guarding (slave monitoring)

The master (controller) guards the slave (Can-node). The master regularly sends out remote frames to the slave (COB-id = 700h+node address). The slave responses with a telegram containing its status data. The time delay between two queries from the master is stored in the object register (guard time in ms; index 100Ch, sub-index 0).

7.10 Life guarding (master monitoring)

The slave (Can-node) guards the master (controller). The master regularly sends out remote frames to the slave (COB-id = 700h+node-id). The product from guard time (index 100Ch, sub-index 0, in ms) and life time factor (index 100Dh, sub-index 0) determines the time which may pass between to queries. If this time is exceeded, the Can-node will switch to the “pre-operational” mode and switches the outputs to a predetermined state. At the same time an emergency telegram is sent. Life guarding is started when the “guard time” and the “life time factor” are not equal 0 and the first master query is received by the slave (Can node).
This error resets as soon as the Can-node is brought back to the “operational” state.

7.11 SDOs (communication)

Index	Sub-Index	Type	Default	Function
1000h	0	unsigned 32	0x91010000	Device type, read only
1001h	0	unsigned 8	0x00	Error register, read only Bit0: general errors, is set with every error Bit1: current error, short Bit2: voltage error, under/over voltage Bit3: temperature error, over temperature Bit4: communications error CAN Bit5: unit specific error, is not used Bit6: reserved, always 0 Bit7: manufacturer specific error, is not used
1003h	0	unsigned 8	0	Number of entries, read only
1003h	1-5	unsigned 16	0	Five of the last encountered errors, read only
1005h	0	unsigned 32	0x00000080	Sync Cob-id; read only
1006h	0	unsigned 32	0	Sync Cycle Time = 0; read only; no sync producer
1008h	0	string	"BCD130-CAN"	Device name; segmental transmission, read only
1009h	0	string	"1 0132 210 4 01"	Device hardware, read only
100Ah	0	string	"V40.xx”	Device software, read only
100Ch	0	unsigned 16	0	Guard time; life guarding is supported
100Dh	0	unsigned 8	0	Life time factor
1014h	0	unsigned 32	0x80+node id	Emergency Cob-id, read only
1017h	0	unsigned 16	0	Producer heart beat time = 0, read only, not used
1018h	0	unsigned 8	4	Number of entries, read only
1018h	1	unsigned 32	0x00000000	Vendor id, read only
1018h	2	unsigned 32	0	Product code, read only
1018h	3	unsigned 32	0	Revision number, read only
1018h	4	unsigned 32	0	Serial number, read only
1400h	0	unsigned 8	5	Number of entries, read only
1400h	1	unsigned 32	0x200+node id	RPDO1 Cob-id, read only
1400h	2	unsigned 8	255	RPDO1 transmission type; 0...240, 254, 255(instant)
1400h	3	unsigned 16	0	Inhibit time = 0; read only, not used
1400h	4	unsigned 8	7	not used, read only
1400h	5	unsigned 16	0	Event timer = 0; read only, not used
1401h	0	unsigned 8	5	Number of entries, read only
1401h	1	unsigned 32	0x7F	RPDO2 Cob-id, 1...127(7Fh)
1401h	2	unsigned 8	255	RPDO2 transmission type; read only
1401h	3	unsigned 16	0	Inhibit time = 0; read only, not used
1401h	4	unsigned 8	7	not used, read only
1401h	5	unsigned 16	0	Event timer = 0; read only, not used
1600h	0	unsigned 8	3	Number of entries (RPDO1 mapping)
1600h	1	unsigned 32	0x20010020	RPDO mapping, entry 0, target position (signed 32)
1600h	2	unsigned 32	0x20010118	RPDO mapping, entry 1, target frequency (unsigned 24)
1600h	3	unsigned 32	0x20010208	RPDO mapping, entry 2, command (unsigned 8)
1601h	0	unsigned 8	1	Number of entries (RPDO2 mapping)
1601h	1	unsigned 8	0x00000008	RPDO mapping, entry 0, dummy (signed 8)
1800h	0	unsigned 8	5	Number of entries
1800h	1	unsigned 32	0x180+node id	TPDO1 Cob-id, read only
1800h	2	unsigned 8	255	TPDO1 transmission type; 0: (with sync and changes of transmission data) 1...240: (sync), 254, 255: (event controlled with changes of actual status, executed command, index 2002h (sub02h/03h)
1800h	3	unsigned 16	0	Inhibit time = 0; read only, not used
1800h	4	unsigned 8	7	not used
1800h	5	unsigned 16	1000	Event timer = 0...10000ms 1000 indicates TPDO1 is sent minimum once per second

Index	Sub-Index	Type	Default	Function
1801h	0	unsigned 8	5	Number of entries
1801h	1	unsigned 32	0x7F	TPDO2 Cob-id, 1...127 (7Fh)
1801h	2	unsigned 8	255	TPDO2 transmission type; read only
1801h	3	unsigned 16	0	Inhibit time = 0; read only, not used
1801h	4	unsigned 8	7	not used
1801h	5	unsigned 16	0	Event timer = read only, not used
1A00h	0	unsigned 8	4	Number of entries (TPDO1 mapping)
1A00h	1	unsigned 32	0x20020020	TPDO-mapping, entry 0, actual position (signed 32)
1A00h	2	unsigned 32	0x20020110	TPDO-mapping, entry 1, actual rpm (unsigned 16)
1A00h	3	unsigned 32	0x20020208	TPDO-mapping, entry 2, executed command (unsigned 8)
1A00h	4	unsigned 32	0x20020308	TPDO-mapping, entry 3, actual status (unsigned 8)
1A01h	0	unsigned 8	1	Number of entries (TPDO2 mapping)
1A01h	1	unsigned 8	0x00000008	TPDO-mapping, entry 0, dummy (unsigned 8)

7.12 SDOs (configuration data)

Index	Sub-Index	Type	Default	Function	Range
2000h	0	unsigned 8	16	Number of entries	
2000h	1	unsigned 32	5000	fmin, start frequency	1... 50 000 Hz
2000h	2	unsigned 32	200 000	fmax, maximum frequency	1...200 000Hz
2000h	3	unsigned 32	0	Ramp profile	0: linear 1: sine 2: sine squared 3: sine3 4: exponential
2000h	4	unsigned 32	100	Ramp parameter	sin3: 88...112 (flatter...steeper) exp: 50...500 (flatter...steeper) otherwise: no function
2000h	5	unsigned 32	500	Acceleration time	10...10 000 ms (fmin to fmax)
2000h	6	unsigned 32	8000	Maximum motor current (hardware depending)	8000 equals 8.0Aeff (read only)
2000h	7	unsigned 32	1000	Standby current	100...6000 equals 0,1 ... 6,0A <= Run current !
2000h	8	unsigned 32	2000	Run current	100...6000 equals 0,1 ... 6,0A
2000h	9	unsigned 32	2000	Boost current	100...8000 equals 0,1 ... 8,0A >= Run-Strom!
2000h	10	unsigned 32	100	Resonance dampening	80...120
2000h	11	unsigned 32	17	Mode reference run	17 = reverse, neg. end switch 17...30 from DS402
2000h	12	unsigned 32	5000	Ref-In; step frequency to Ref-/end switch	1 ... 50,000 Hz
2000h	13	unsigned 32	20	Ref-Out; step frequency from the Ref-/end switch	1 ... 50,000 Hz
2000h	14	unsigned 32	-	Supply voltage	0...100 equals 0...100V
2000h	15	unsigned 32	-	Temperature	for internal use only
2000h	16	unsigned 32	10000	Step resolution	400, 800, 1000, 1600, 2000, 3200, 5000, 6400, 10000 12800, 20000, 25600
2000h	17	unsigned 32	50	Delay time for QuickStop	1...10000ms (fmax to fmin)
2000h	18	unsigned 32	0	Invert inputs	0: not inverted 1: invert end switch 2: invert reference switch 3: invert end/reference switch
2000h	48	unsigned 32	2x10E6	Trigger position	+/- 1 000 000 000
2000h	49	unsigned 32	0	Trigger delay	0... 6000ms

7.13 SDOs (user data)

Input values (RPDO)

Index	Sub-Index	Type	Default	Function	Range
2001h	0	signed 32	0	Target position	+/- 1 000 000 000 steps
2001h	1	unsigned24	0	Target frequency	1...200 000 Hz (0 command 3 only, rpm)
2001h	2	unsigned 8	0	Command	see description 7.4, page 10

Output values (TPDO), read only

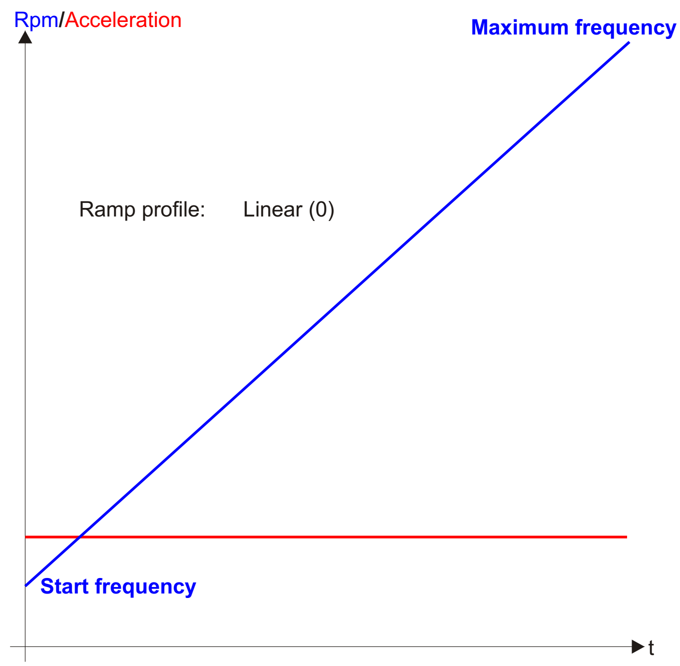
Index	Sub-Index	Type	Default	Function	Range
2002h	0	signed 32	0	Actual position	+/- 1 000 000 000 steps
2002h	1	unsigned 16	0	Actual frequency	1...20 000 equals 10...200 000 Hz
2002h	2	unsigned 8	0	Executed command	
2002h	3	unsigned 8	0	Actual status	

7.14 Configuration

7.14.1 Start frequency (index2000h, sub-index 1), maximum frequency (sub-index 2), ramp profile (sub-index 3), ramp parameter (sub-index 4), acceleration time (sub-index 5)
The motor accelerates within the acceleration time beginning with the start frequency to the maximum frequency. During normal operation the acceleration time is also used for the deceleration of the axis.
The maximum frequency is to be equal or larger then the start frequency. The ramp profile determines the frequency curve during acceleration. The ramp parameter determines the characteristic of the sine3 ramp (3) and the exponential ramp (4).

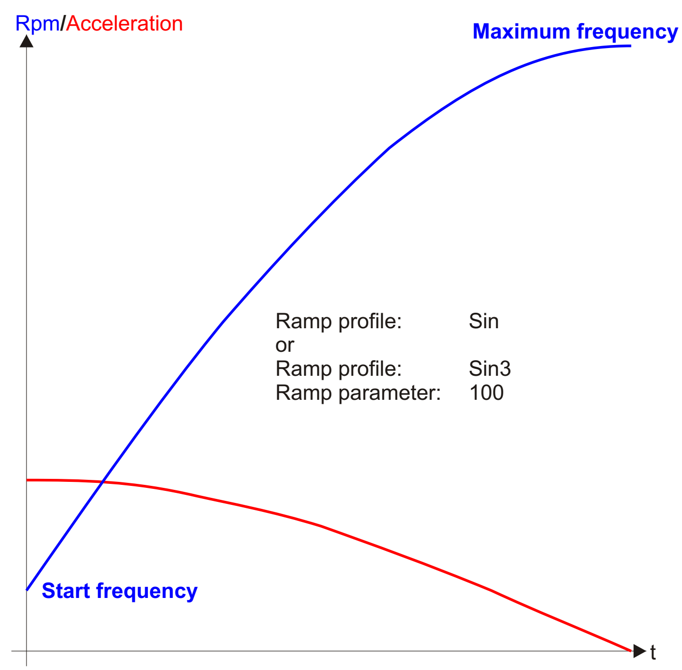
7.14.1.1 Ramp profile: linear (0)

The linear rpm progression with constant acceleration may be used for most standard applications. The acceleration is constant throughout the whole rpm range.



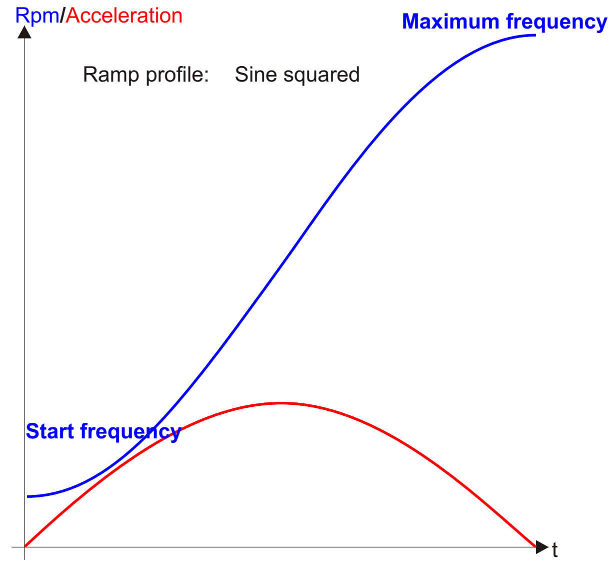
7.14.1.2 Ramp profile: Sine (1)

The sinusoidal rpm progression is used in applications with a high inertia moment, where the wearing-off inertia needs adjustment at higher rpms. The acceleration reduces with increased rpm.



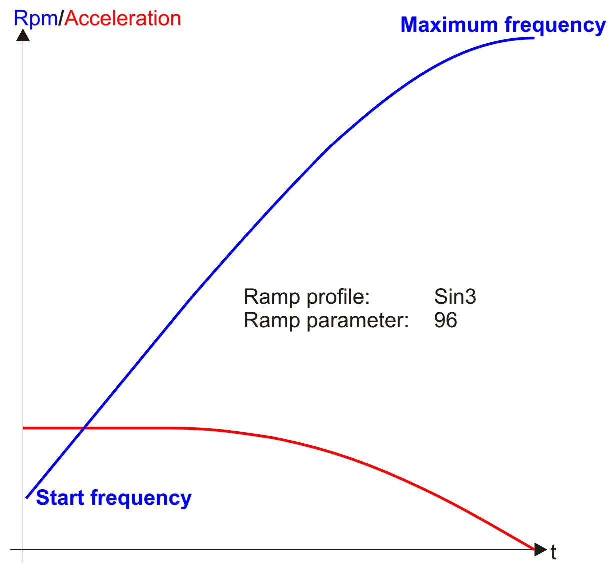
7.14.1.3 Ramp profile: Sine squared (2)

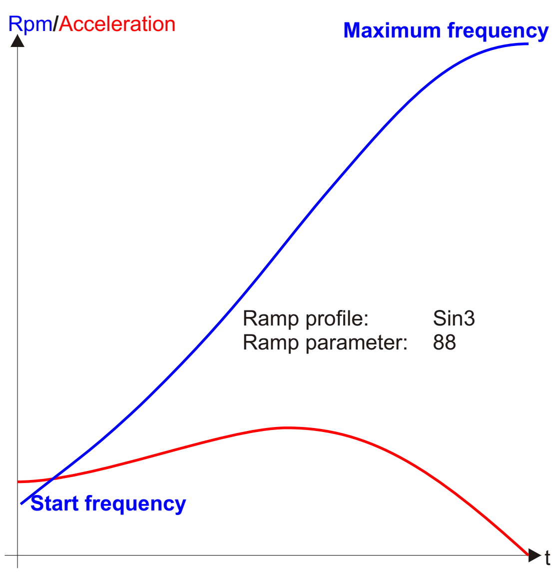
This profile is especially used for applications where a slow and smooth start from standstill is needed. The reduced acceleration in the upper rpm compensates the reduced torque in this area. The acceleration is the highest in the middle of the rpm range.

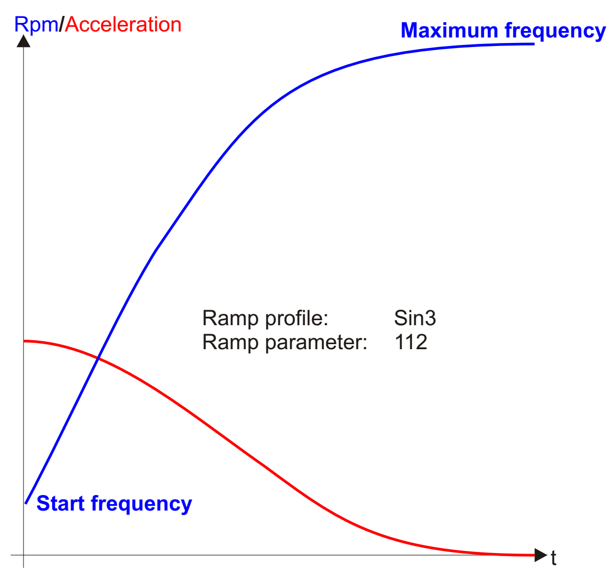


7.13.1.4 Ramp profile: Sine3 (3)

The Sin3 rpm profile allows changing the rpm progression with the ramp parameter. If the ramp parameter is set to 100, the progression is sinusoidal. A value of 93 equals a constant acceleration in the lower rpm range. The acceleration tapers off at higher rpms only.
Practical values lay between 88 and 100.

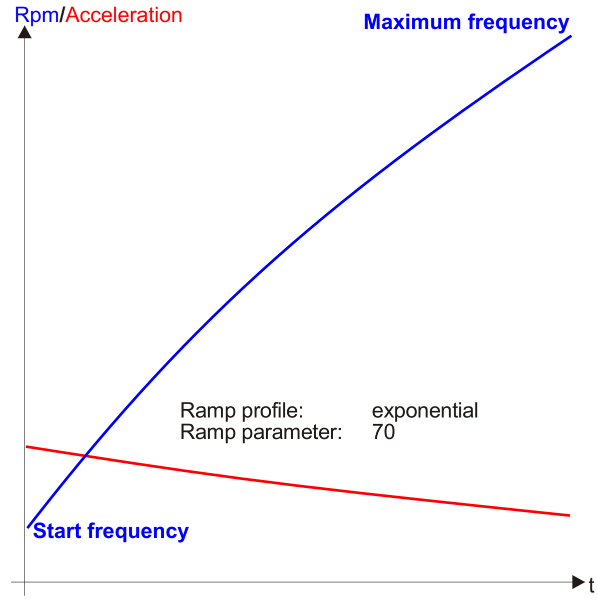


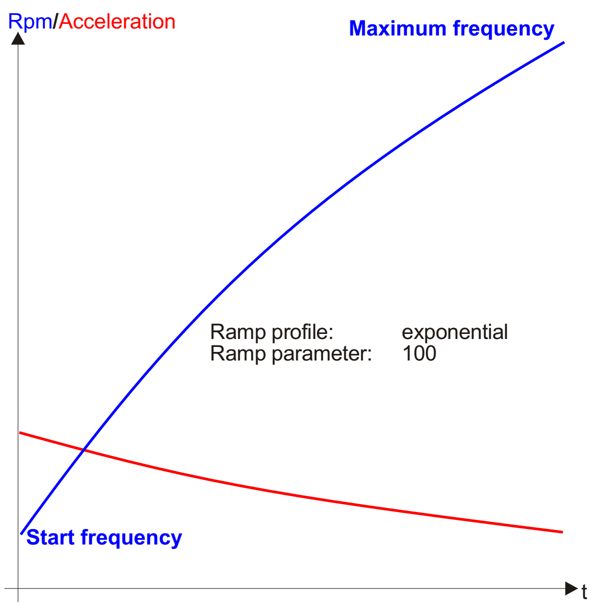


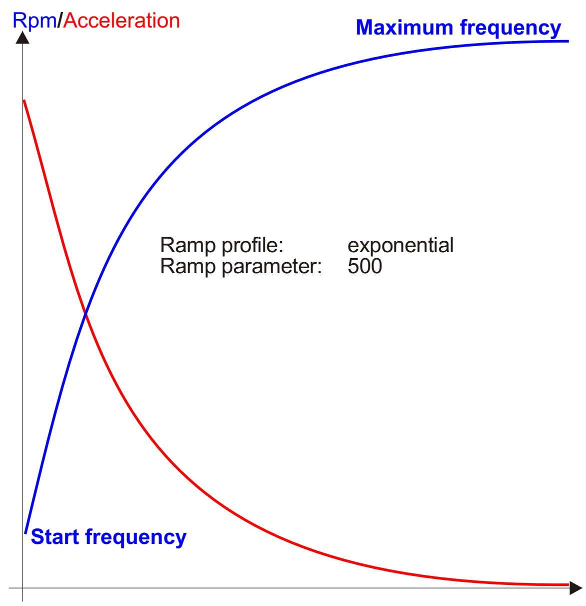


7.14.1.5 Ramp profile: Exponential (4)

The exponential rpm progression allows to change the rpm build-up with the ramp parameter. The acceleration in the lower rpms increases with increasing parameter value. If the parameter is set to 70 (110) the acceleration at the start frequency is about 2 times (3 times) higher then at the maximum frequency.
An exponential ramp parameter value higher then 110 is only meaningful in very few applications.







7.14.2 Maximum motor current (index2000h, sub-index 6)

This parameter is read only and shows the maximum actual phase current for the hardware. The peak value is approximately 1.4 times higher.

7.14.3 Standby current (index2000h, sub-index 7)

At motor standstill the standby current is flowing through the motor coils. Since typically only a small holding torque is needed during standstill, a standby current between 10% and 50% of the run current is normally used. If the standby current should be larger then the run current, the run current value is entered first followed by the standby value.

7.14.4 Run current (index2000h, sub-index 8)

The run current determines the motor current at constant rpm. By setting this parameter it is checked if the standby current is lower than the run current, and the boost current is higher then the run current. The standby current and boost current may be adjusted according to the value of the run current.

7.14.5 Boost current (index2000h, sub-index 9)
The boost current determines the motor current during acceleration and deceleration (braking). If larger inertia moments (mass) are accelerated or decelerated a larger boost current of up to 120% may be set. The drive receives therefore during acceleration and deceleration more torque.
If the boost current should be at a lower value then the run current, adjust first the run current then the boost value.

7.14.6 Resonance dampening (index2000h, sub-index 10)
Quiet and low resonance running of a drive depends largely on the mechanical motor construction. Adjusting the resonance dampening adapts the driver to the motor. Since no such data is found in the motor data sheets, only trial runs will find the best settings. Most of the motors however will run best with settings of 100 to 110. Middex will support you if needed.

7.14.7 Mode reference run (index2000h, sub-index 11)
The type of reference run to the reference and end switches are defined. The reference run is described under 8. page 23.

7.14.8 Ref-In (index2000h, sub-index 12), Ref-Out (index2000h, sub-index 13)
During the reference run the end or reference switch is searched with the step frequency Ref-In. The reference position is then accurately located with the lower Ref-Out step frequency.

7.14.9 Supply voltage (index2000h, sub-index 14)
The value for the supply voltage (in volts) is read only.

7.14.10 Temperature (index2000h, sub-index 15)
The temperature value of the power electronics is read only. The characteristic is non-linear.

7.14.11 Step resolution (index2000h, sub-index 16)
The step resolution is selectable. If an inadmissible value is selected, the value will be set to the maximum step resolution value. Change values only when the motor is at standstill.

7.14.12 Delay time for QuickStop (index2000h, sub-index 17)
The delay time is the time needed to slow down from the maximum frequency to the start frequency. This delay is used only with QuickStop. Normally, the delay time is set shorter than the acceleration time, which is used in normal operation.

7.14.13 Inverting inputs (index2000h, sub-index 18)
The end and reference switches can be inverted. With a value of 0, the switches are not inverted. In normal operation both inputs are at 0V. A run in positive (negative) direction will stop if the input for the positive (negative) end switch will be set to 24V.
With a value of 1 the end switches are inverted. During normal operation both inputs are at 24V. A run in positive (negative) direction will stop if the input for the positive (negative) end switch will be set to 0V.
With a value of 2 the reference switch is inverted.
A value of 3 will invert all end and reference switches.

7.14.14 Trigger position (index2000h, sub-index 48)
If the actual, absolute position is lower then the trigger position, the trigger output is switched off. If the actual, absolute position is larger or equal then the trigger position, the trigger output is switched on. Reaching the trigger position, a TPDO2 is sent.

7.14.15 Trigger delay (index2000h, sub-index 49)
The value defines the time delay in ms after a trigger event before the command is executed. A value of 0 is executed immediately when a signal level is at the trigger input or when a RPDO2 is received.

7.15 Possible responses to SDOs

0x60, index low byte, index high byte, sub-index, 0x00, 0x00, 0x00, 0x00
- Write OK

0x43, index low byte, index high byte, sub-index, data 4 byte (low byte first)
- Read OK, data lengths 4 byte

0x47, index low byte, index high byte, sub-index, data 4 byte (low byte first)
- Read OK, data lengths 3 byte

0x4B, index low byte, index high byte, sub-index, data 4 byte (low byte first)
- Read OK, data lengths 2 byte

0x4F, index low byte, index high byte, sub-index, data 4 byte (low byte first)
- Read OK, data lengths 1 byte

Errors:

0x80, index low byte, index high byte, sub-index, 0x00, 0x00, 0x02, 0x06
- Entry (index) not present

0x80, index low byte, index high byte, sub-index, 0x11, 0x00, 0x09, 0x06
- Sub-index not present

0x80, index low byte, index high byte, sub-index, 0x30, 0x00, 0x09, 0x06
- Parameter value not permissible
- i.e.. rpm outside allowable range

0x80, index low byte, index high byte, sub-index, 0x02, 0x00, 0x01, 0x06
- Write on read only

0x80, index low byte, index high byte, sub-index, 0x00, 0x00, 0x00, 0x08
- General fault
- Wrong distance value (s, e, n)

7.16 Error recovery

Error: **The motor stops shortly after starting.**
Correction:
- Enter a lower value for the starting frequency or
- Increase value for acceleration time or
- Increase the value for the run and boost current or
- The motor torque is too small use a larger motor or
- The axis is binding and repairs are needed.

Error: **The motor turns in the wrong direction.**
Correction:
- Flip the motor connections A and B or
- Flip the motor connections C and D.

Error: **No connection between the bus-master (controller).**
Correction:
- Set address value between 1 and 7F or
- Set Baud rate to the same value as the bus-master.

8. Reference Run

The reference run is based on the CanOpen profile DSP-402 and is started when the command in the RPDO1 changes to 6. The reference run mode is configured with a SDO (index 2000h, sub-index 11(0Bh). The drive runs with the higher speed "Ref-High" (index 2000h, sub-index 12/0Ch) to the reference or end switch and then the zero position is accurately set with the lower speed "Ref-Low" (index 2000h, sub-index 13/0Dh).
The default value of 17 generates a reference run to the negative end switch.

17 Backward reference run until the negative end switch switches on
Axis runs backward with the step frequency "Ref-High" until the negative end switch switches on.
Axis runs forward with the step frequency "Ref-High" until the negative end switch switches on.
Axis runs forward with the step frequency "Ref-High" until the negative end switch switches off.
Axis runs backward with the step frequency "Ref-Low" until the negative end switch switches on.
The actual position is set to (0) zero.

18 Forward reference run until the positive end switch switches on
Axis runs forward with the step frequency "Ref-High" until the positive end switch switches on.
Axis runs backward with the step frequency "Ref-High" until the positive end switch switches on.
Axis runs backward with the step frequency "Ref-High" until the positive end switch switches off.
Axis runs forward with the step frequency "Ref-Low" until the positive end switch switches on.
The actual position is set to (0) zero.

19/20 Forward reference run until the reference switch switches on
Axis runs backward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs forward with the step frequency "Ref-Low" until the reference switch switches on.
Axis runs backward with the step frequency "Ref-Low" until the reference switch switches off.
Axis runs forward with the step frequency "Ref-Low" until the reference switch switches on.
The actual position is set to (0) zero.

21/22 Backward reference run until the reference switch switches on
Axis runs forward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs backward with the step frequency "Ref-High" until the reference switch switches on.
Axis runs forward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs backward with the step frequency "Ref-Low" until the reference switch switches on.
The actual position is set to (0) zero.

23/24 Forward reference run until reference switch switches on (starting in positive direction)
Axis runs forward with the step frequency "Ref-High" until the reference switch or the positive end switch switches on.
Axis runs backward with the step frequency "Ref-High" until the reference switch switches on.
Axis runs backward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs forward with the step frequency "Ref-Low" until the reference switch switches on.
The actual position is set to (0) zero.

25/26 Backward reference run until reference switch switches on (starting in positive direction)
Axis runs forward with the step frequency "Ref-High" until the reference switch or the positive end switch switches on.
Axis runs forward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs backward with the step frequency "Ref-High" until the reference switch switches on.
Axis runs forward with the step frequency "Ref-High" until the reference switch switches on.
Axis runs forward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs backward with the step frequency "Ref-Low" until the reference switch switches on.
The actual position is set to (0) zero.

27/28 Backward reference run until reference switch switches on (start in negative direction)
Axis runs backward with the step frequency "Ref-High" until the reference switch or the negative end switch switches on.
Axis runs forward with the step frequency "Ref-High" until the reference switch switches on.
Axis runs forward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs backward with the step frequency "Ref-Low" until the reference switch switches on.
The actual position is set to (0) zero.

29/30 Forward reference run until reference switch switches on (start in negative direction)
Axis runs backward with the step frequency "Ref-High" until the reference switch or the negative end switch switches on.
Axis runs backward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs forward with the step frequency "Ref-High" until the reference switch switches on.
Axis runs backward with the step frequency "Ref-High" until the reference switch switches on.
Axis runs backward with the step frequency "Ref-High" until the reference switch switches off.
Axis runs forward with the step frequency "Ref-Low" until the reference switch switches on.
The actual position is set to (0) zero.

9. Synchronizing Several Axes

9.1 Synchronizing via in and outputs

At the master axis the absolute position is parametrically set via trigger position (index 2000h, sub-index 48). The trigger position may also be set with the command 48. If the actual position is lower then the trigger position the trigger output is off. If the actual position is larger or equal then the trigger position, the trigger output is switched on.

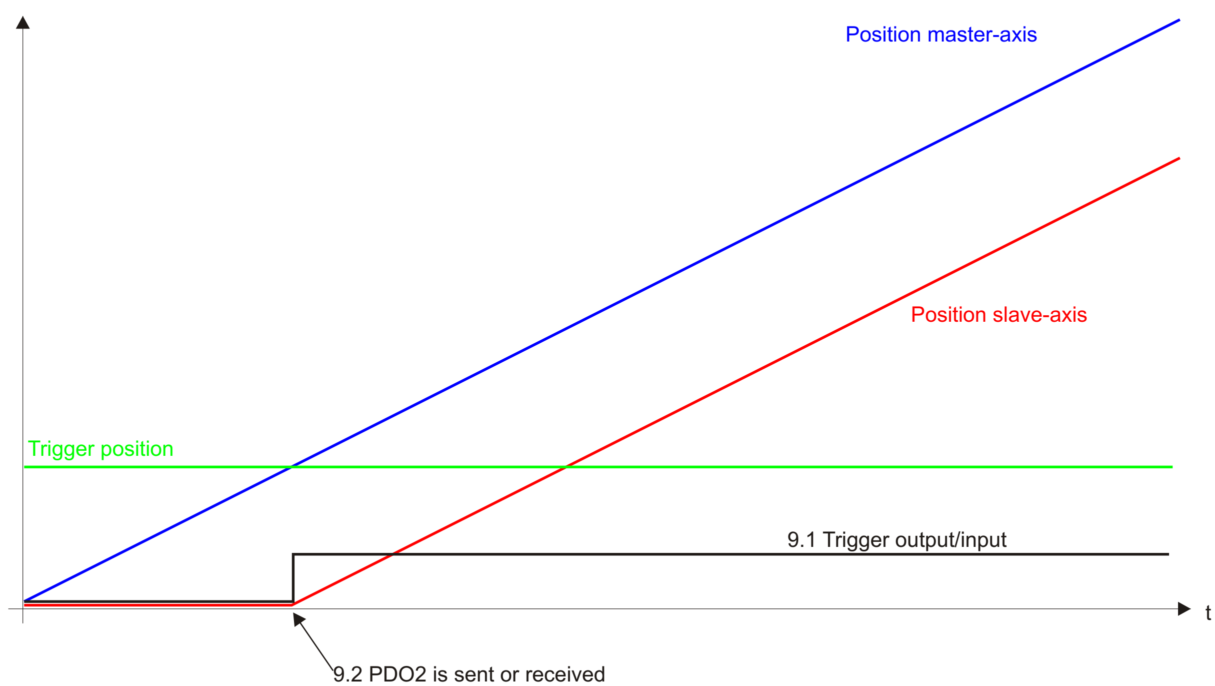
The trigger output of the master axis is connected with the trigger inputs of the slave axes. Alternatively, the trigger inputs may also be connected with a digital output of the controller, with a position sensor, with a proximity switch, or with a light barrier.

The slave axes start their commands in synch with the commands 55, 56, ... from the trigger input. The command execution may be delayed with the trigger delay parameter (index 2000h, sub-index 49) from 0 to 6000ms. The trigger delay may also be set with the command 49.

9.2 Synchronizing via PDO2

At the master axis the absolute position is parametrically set by the trigger position (index 2000h, sub-index 17). The trigger position may also be set with the command 48. If the actual position matches the trigger position a TPDO2 is sent. Alternatively, the TPDO2 may also be sent from the CanOpen master or from another slave. This way several axes may be started in synch.

The command 55.56... starts synchronized position commands at the slave axes when the RPDO2 is received. The command execution may be delayed with the trigger delay parameter (index 2000h, sub-index 49). The trigger delay may also be set with the command 49.



10. Pulse / Direction Interface

10.1 In general

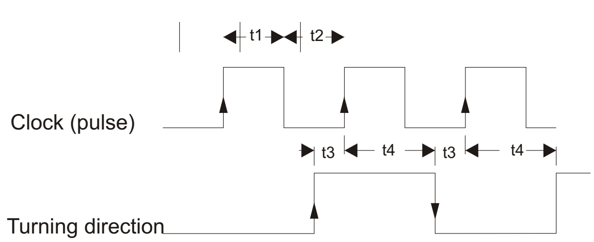
From version 40.04 in addition to the CanOpen interface a pulse and direction interface is included. To activate the pulse/direction interface, turn the lower turn switch past a value of 6 after the unit is powered up. The CanOpen is thereby deactivated.

- The supply voltage and motor is connected as per section 5, page 4
- Section 6, page 6 describes how to hook up different step motors
- The terminals for pulse, direction, boost, disable, and ready are found in section 5 (in parenthesis)
- With each ascending edge at the pulse input the motor turns one step
- Maximum step frequency is 200kHz
- The direction input determines the motor turning direction
- A signal at the disable input switches the motor off
- A voltage on the boost input will boost the motor current by 20%
- Motor standstill uses a reduced current of 50% of the set motor current
- The output ready signals service readiness
- LEDs show the operation status

Phase current (upper turn switch)		**Resonance dampening** (middle turn switch)		**Step resolution** (Lower turn switch)	
Position	Phase current	Position	Resonance dampening	Position	Step resolution
0	1.0Aeff	0	off	0	CanOpen
1	1.33Aeff	1	Stufe1	1	CanOpen
2	1.67Aeff	2	Stufe2	2	CanOpen
3	2.0Aeff	3	Stufe3 (standard)	3	CanOpen
4	2.33Aeff	4	Stufe4	4	CanOpen
5	2.67Aeff	5	Stufe5	5	CanOpen
6	3.0Aeff	6	Stufe6	6	CanOpen
7	3.33Aeff	7	Stufe7	7	400 steps/revolution
8	3.67Aeff	8	Stufe8	8	800 steps/revolution
8	4.0Aeff	9	Stufe9	9	1600 steps/revolution
A	4.33Aeff	A	Stufe10	A	6400 steps/revolution
B	4.67Aeff	B	Stufe11	B	1000 steps/revolution
C	5.0Aeff	C	Stufe12	C	2000 steps/revolution
D	5.33Aeff	D	Stufe13	D	5000 steps/revolution
E	5.67Aeff	E	Stufe14	E	10000 steps/revolution
F	6.0Aeff	F	Stufe15	F	20000 steps/revolution

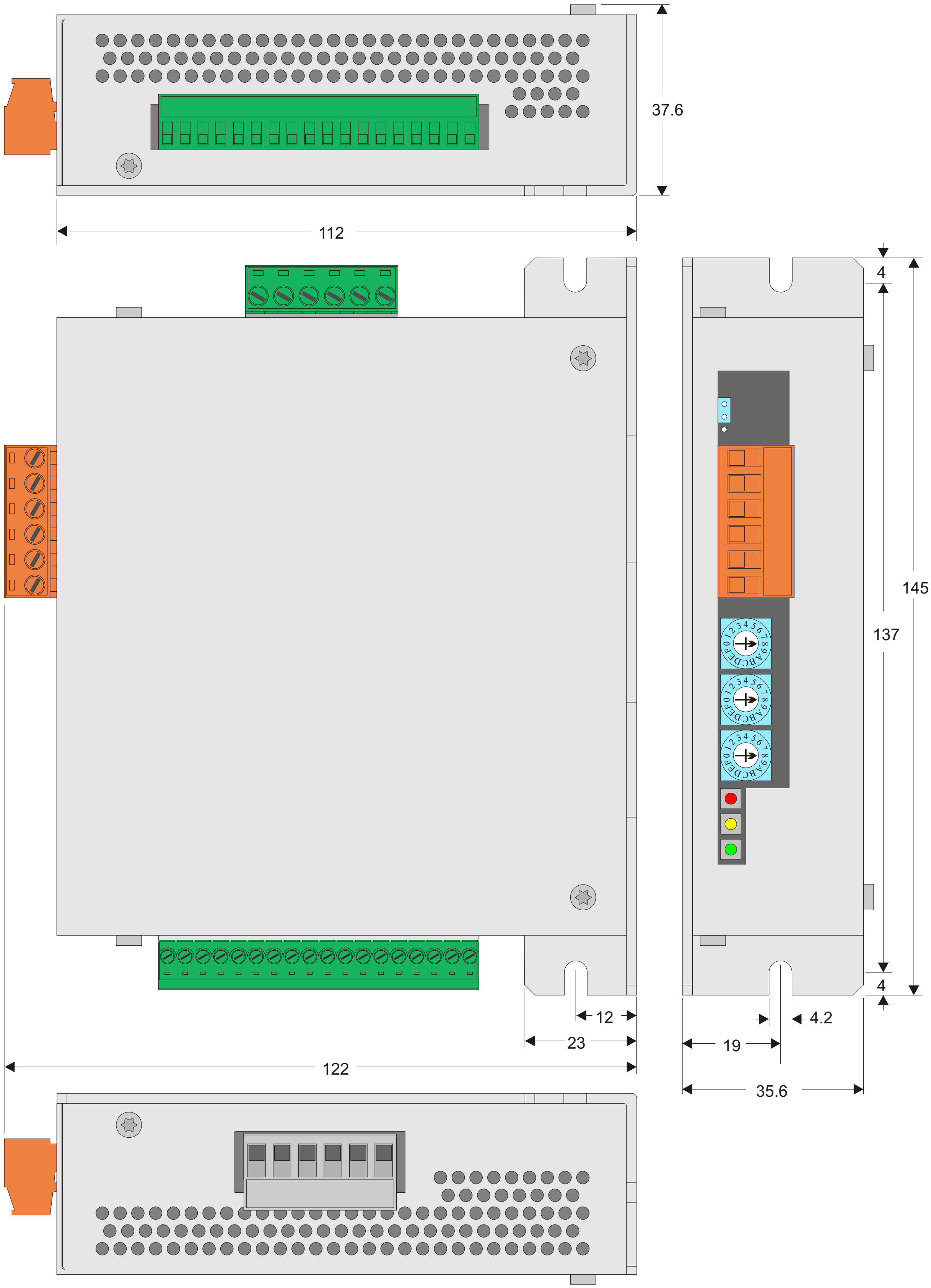
On request other resolutions (200, 400, 500, 800, 1000, 1200, 1600, 2000, 2500, 3200, 4000, 5000, 6400, 10000, 12800, 20000 and more) and any combination of current between 0.5 and 6A even with low quantities are available.

10.2 Signal switching times



t1	:	Pulse width	> 2.5µs
t2	:	Pulse pause	> 2.5µs
t3	:	Set up time direction	> 2.5µs
t4	:	Hold time direction	> 2.5µs
tr	:	Ascending edge	< 0.2µs
tf	:	Descending edge	< 0.2µs

11. Dimensions



12. Versions

V41.00: prototype
V41.01: prototype
V41.02: prototype
V41.03: first series version
V41.04 from 02.03.2010:

1) With a command 3 (rpm) and a rpm target of 0 the motor stops. In earlier versions the motor kept running with the lowest frequency of 1 Hz if a target rpm of 0 was given.
2) The inputs for end and reference switches may be inverted.
3) Cold start with "reset node" and "reset communication".
4) Added pulse and direction interface.

13. Technical Data

Supply power:	22...80VDC (momentary maximum 90VDC)
Motor current:	0,5...6.0Aeff (momentary 8.0Aeff/12Apeak)
Step resolution:	400, 800, 1600, 3200, 6400, 12800, 25600 steps/revolution 1000, 2000, 5000,10000, 20000 steps/revolution
End/reference switches:	Inputs 24V, opto-decoupled
Encoder inputs:	customer specific, on request
Trigger input:	24V delay time = 200...350µs + (0...1/fmax)
Trigger outputs:	opto-decoupled, maximum 30V/10mA delay time maximum 20µs
Status display:	3 LEDs; red, yellow, green
Interface:	CAN, electrically isolated 8...30VDC (at the bus plug required)
Protocol:	CanOpen, DS301
Connectors:	Motor, supply voltage: Phoenix MSTB2,5/6-ST-5,08; 1757051; or similar Phoenix FKC 2,5/6-ST, 5,08; 1873090 (spring, optional) CanOpen: Phoenix MSTB2,5/6-ST-5,08; 1757051; or similar Phoenix FKC 2,5/6-ST, 5,08; 1873090 (spring, optional) Signals: Phoenix MC1,5/18-ST-3,5; 1840528; or similar Phoenix FMC 1,5/18-ST-3.5; 1952429 (spring, optional)
Motors:	2-Phase step motors (bipolar) with 4, 6 or 8 connections
Protection:	- Over/under voltage - Over temperature - Shorted motor connections
Brake resistance:	not installed
Ambient temperature:	0...45°C
Shut-off temperature:	approximately 85°C
Storage temperature:	0...70°C
Humidity:	20-80%, no precipitation
Dimensions:	see drawing page 26
Weight:	approximately. 0.65kg (3.2 lb)