BAUR
ANTRIEBE
UND
REGELUNGEN

Operator Manual for Power Drive Controller

pdcx85

Optional Heat Sink

Revision: 37/2009 Subject to change without prior notice

TABLE OF CONTENTS

Product Features

- Power drive plus controller

- For 2 phase and 3 phase stepper motors (microstep), DC-motors or Brushless-DC (EC) motors

- Motion and control functions

- Supply: 24...80(130) Volt, up to 5(10)Amps

- Front panel parameter setting requires no additional input device

- Multiple protection functions
  Over current, temperature, voltage, motion monitoring, ...

- User-Interface (galvanic isolated)
  Pulse/Direction, RS232/RS485, SPI, CANopen (optional)

- Peripheral-Interface (galvanic isolated)
  2xlimit switch, reference switch, stop input
  Ready output, 1x output: operating mode dependent

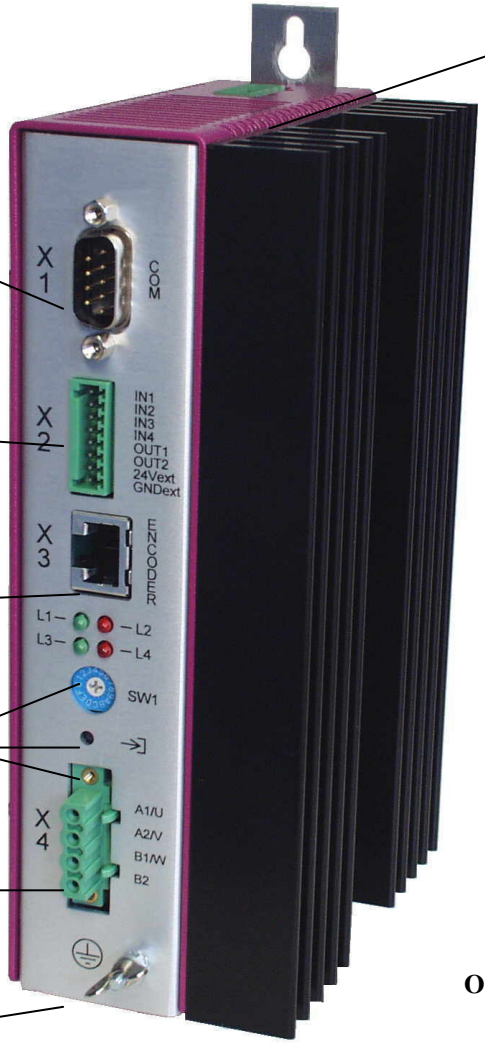
- Encoder-Interface (galvanic isolated)
  Incremental-Encoder, HALL-Sensors

- Status LEDs

- Alle connections with removable screw terminals

- Compact metal housing, wallmount, optional DIN-rail clip

Placement of the Interfaces



Power Supply
Power drive plus controller

User-Interface X1 (optional)
Pulse and direction
RS232
RS485 (planned)
SPI/SSI
CANopen

Peripheral-Interface X2
Limit switch left/right
Reference switch
Stop input
Ready output
1x free output

Encoder-Interface X3
Incremental encoder
HALL-sensors

Settings
Diagnostics
Parameter setting

Motor connection X4
2-phase or 3-phase stepper motor
DC-servomotor
Brushless DC-motor (EC-motor)

Earth grounding

Option: Heat Sink

Ordering Number Key

pdcx85.11xxxxx

Power Drive & Controller

Series Motor Type*
- 0 DC Motor
- 1 BLDC (EC) Motor
- 2 2 Phase Stepper motor
- 3 3 Phase Stepper motor

Series Nominal Voltage **
- 8 24...80Vdc

Series Nominal Current
- 5 1,25...5Amps

Signal Voltage
- 1 24V signal input voltage

Housing Version
- 1 IP20 (with Hood)

Cooling
- 0 no Heat sink
- 1 with Heat sink

Mounting
- 0 Wallmount
- 1 DIN-rail mounting clip

User Interface
- 0 without
- 1 RS232
- 2 RS485 (planned)
- 3 ---
- 4 CANopen SUI (Simple User Interface)
- 5 PULSEE/DIRECTION

Motor Voltage Range
- 0 Nominal motor voltage 24...80Vdc
- 1 Higher motor voltage 60...130Vdc

Motor Current Range
- 0 Nominal motor current 1,25...5Amps
- 1 Higher motor current 2,5...10Amps

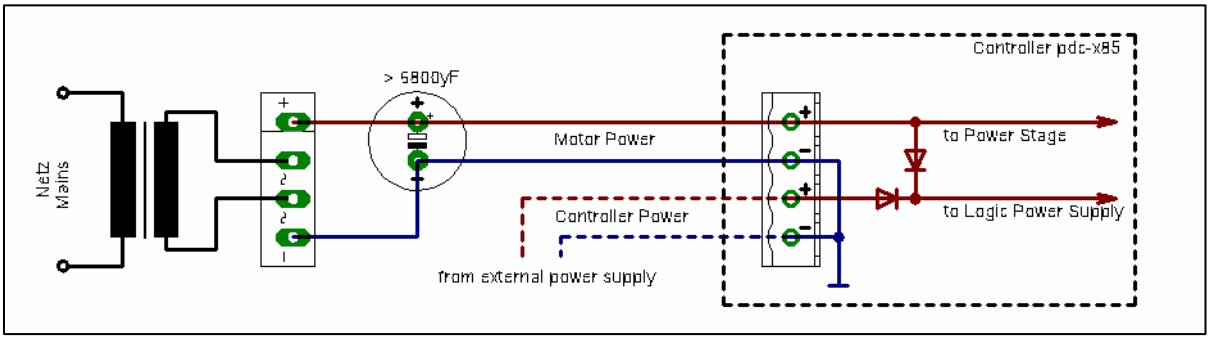
Accessories (separate order items)

STS.02	Interface connector set: Supply voltage, X1-User, X2-Peripheral and X4-Motor
STS.03	SUB-D 9 pole terminating resistor female/male with 120 Ohm
HS.00	Heat sink (recommended for motor current 6A and higher)
DINH.00	Din rail mounting clip
KD.00	Nullmodem cable 2 Meter female/female SUB D 9pole (! only in combination with RS232 user interface)
KM*m*	Motor cable 2x2 0.75mm^2 twisted pair with shield *m* = length in meter 01, 02, 05, 10 (example 2 Meter motor cable = KM02)
KE.01	Encoder cable with RJ45 connector for interface X3. Second end open. 2.5 meter long.
KE.02	Encoder cable with RJ45 connector for interface X3. Second end open. 5 meter long.
DOKU	DIN-A5 paper bound operator manual. Contains pdcx85 and IOEXT.

* DC, EC-and 2 phase or 3 phase stepper motors upon request. We can assist with motor sizing and drive calculation.
** PFC power supply ps400 (400 Watt 80Vdc or 130Vdc) upon request

Wiring: (general)

Power Supply:



Normally the pdc-x85 is fully supplied by the motor voltage. The controller however must be supplied separately if it is necessary for safety reasons to disable the motor without disabling the controller. This allows to continue a process after a motor e-stop.

It must be guaranteed that the power supply is equipped with a suitable charge capacitor of at least 6800µF, so that the kinetic feed back braking energy doesn't let the supply voltage exceed its limits

Excessively high motor voltages can damage the power drive.

An active device internal ballast circuit eliminates over-voltages in the limit range to a certain degree.

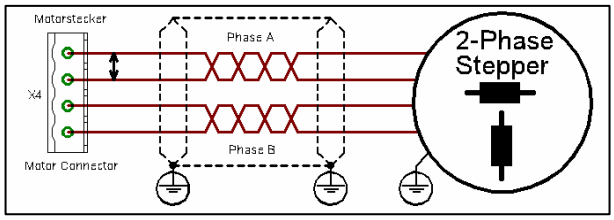
However, at high motor currents and with high inertial load driven, it is necessary to use an adequate power supply that doesn't let the output voltage exceed the max. limit.

Under no circumstances the motor supply voltage must be connected life to the drive because it can be possible that the charge current of the internal capacitors can destroy the fuse. Proper function is guaranteed if the full operating voltage is reached during a quarter mains cycle (5ms @ 50Hz).

Never connect live supply voltage wires to the terminals, because the sudden charge current of the internal electrolytic capacitors can destroy the internal fuses

! Check for correct polarity

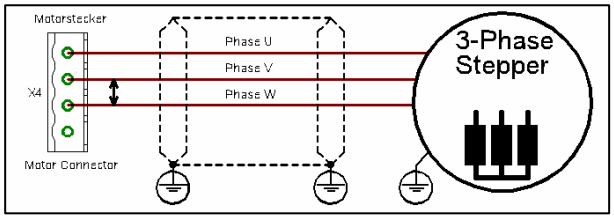
Motor Wiring: 2-Phase Stepper Motor X4



To reverse the sense of rotation simply swap the two wires of Phase A

8 lead motors have a winding pair for each winding A and winding B. These pairs can be connected in parallel or in series. The parallel circuit is advantageous at higher speeds. The series circuit results in a higher torque with less motor current.

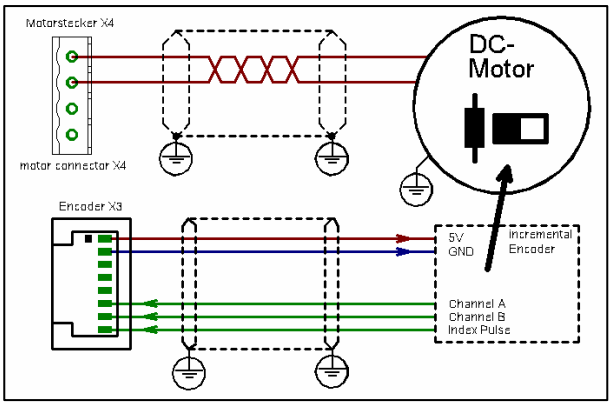
Motor Wiring: 3-Phase Stepper Motor X4



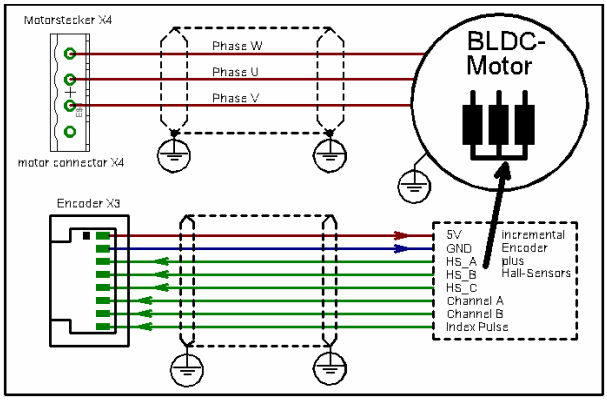
To reverse the sense of rotation simply swap Phase V and Phase W

Warning: Never connect or disconnect life energized motor connectors

Motor Wiring: DC-Motor X4

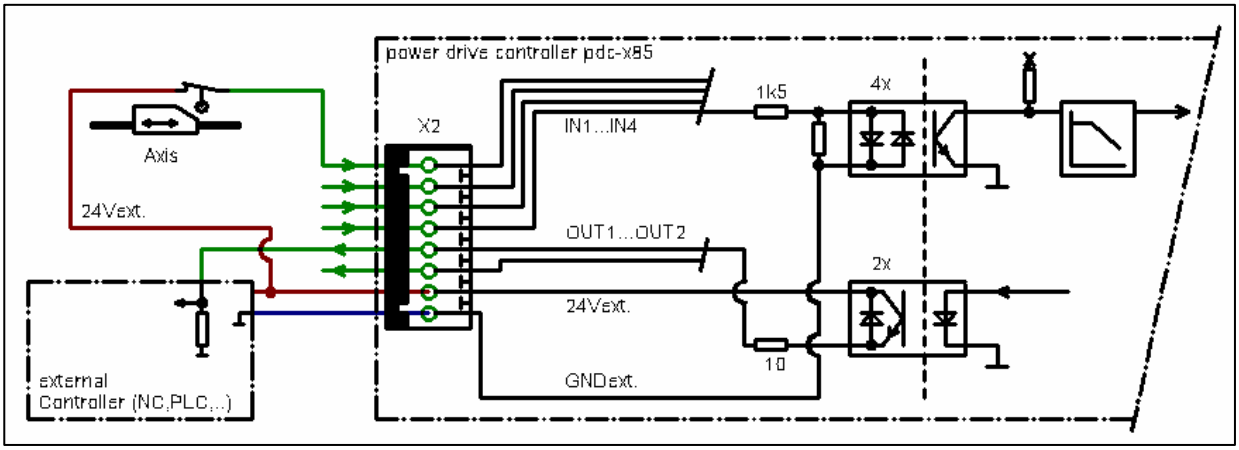


Motor Wiring: Brushless DC-Motor X4



Warning: Never connect or disconnect life energized motor connectors

Wiring: Peripheral Interface X2



The peripheral interface X2 is mainly used to connect axis signals such as limit switches, reference switch and stop-input. The inputs are AC active towards the ground potential ”GNDext“. The allows to chose between high active and low active inputs.

High active: -> GNDext= low (normal case)

Low active: -> GNDext= high

Two outputs are available. Output ”OUT1“ is the standard assigned output for controller readiness. The outputs are generally high-active. That means that the ready output is high if the controller is OK. Supply voltage for the outputs must be provided externally by a power supply or a third part controller that is connected to the outputs.

All signals are galvanic isolated.

Operation, Settings

General Information:
To define the operating characteristics of a controller parameters or settings are necessary. Via a user menu these settings can simply be entered directly at the controller front panel. No external devices such as a PC are necessary. The settings are the stored in the machine parameter set and become immediately active after parameter input.

To store the parameter settings permanently in the controller it is necessary to store ("flash") them to retain memory (see parameter list parameter-code 9E12)

Front panel parameter input

Each parameter is defined with at least 3 input values. These are Index 1 and Index 2 as well as Parameter-Value 1 and Parameter-Value 2

- Index 1: Defines the parameter group
- Index 2: Sub-index for additional parameters within the parameter group (Index 1)
- Parameter 1: Value input as single digit or as a multi-digit #number
- Parameter 2: Value input as single digit

The method yields high flexibility wit a minimum of operating elements and its handling is still transparent . Without external devices (e.g. PC) all basic settings can be entered. In theory 256 parameters can be set.

The below listed front panel elements are available for all operator settings:

- 4x LED Name: L1...4
- 1x HEX-switch SW1
- 1x Enter button ->]

The parameter input procedure is monitored. A flashing L1 indicated parameter setting errors. The new input value is then ignored. This error condition must be reset before attempting to reenter a parameter.

Parameter input:

Step 1:
o Press enter button ->] for more than 2 seconds
to enter the parameter input menu
L1 flashes to indicate that Index 1 must now be entered at SW1
(Index 1)= Parameter group
o Store Index 1 with ->] => now L2 flashes

Step 2:
o L2 flashes to indicate that Index 2 must now be entered at SW1
(Index 2)= Sub-index of Index 1 for additional parameters within the parameter group (Index 1)
o Store Index 2 with ->] => now L3 flashes

Step 3:
o L3 flashes to indicate that Parameter-Value 1 must now be entered at SW1
(Parameter Value 1) represents now a value, that must be entered as a single digit or as a multi-digit #number
o Single digit input:
Store Parameter-Value 1 with ->] =>
now L4 flashes. It must be continued with step 4 even if Parameter-Value 2 is not required.

o Multi-digit #number input
Each digit is stored with ->]. L4 flashes once to indicate acknowledgement. For a negative number enter an 'F' first. Entering an 'A...E' terminates the input routine. Step 4 is not performed.
Multi-digit number inputs are marked with a #.
Number input example: -4711
F4711C ,F'= minus, ,C'= return

Step 4:
o L4 flashes to indicate that Parameter-Value 2 must now be entered at SW1
(Parameter Value 2) represents a value as well
o Store Parameter Value 2 with ->] =>
The parameter input now accomplished, the controller returns from this menu level
for a new parameter input see Step1

If none or only one parameter value is required it is still necessary to perform Step 3 and Step 4 (simply click through, parameter values are ignored).
Step 4 is not performed if Parameter Value 1 is a multi-digit #number.

The default values are shaded.
Non shaded values must be entered in any case.

Parameter list

Index 1	Index 2	Value 1	Value 2	Description
0	---	---	---	Not used
1	0	0...2	---	STEPS: Steps per revolution) 1000, 2000, 4000 stepper motor only ! input as first parameter
	1	#0...F	---	Phase current at stepper motor [mA] 1000 #500...10000 Peak current at DC-Motor
	2	#0...F	---	Current limitation at DC-Motor [mA] 6000 #500...10000
	3	0...5	---	Current reduction to [%] (stepper motor only) 100, 90, 80 70, 60, 50
	4	0...6	---	Encoder increments 0, 50, 100, 200, 250, 500, 1000 Stepper Motor: 0= no encoder evaluation BLDC-Motor: 0= velocity mode only with hall-sensors >0 external encoder is used (minimal 500 steps/rev) selection mandatory
	5	1...4	0...1	Signal inverting Value 1 -> IN1...IN4 Value 2 -> 0= not, 1= inverted
	6	#0...F	---	Gear factor: Numerator ! not active, when XFER > 0 #1...1000
	7	#0...F	---	Gear factor: Denominator ! not active, when XFER > 0 #1...1000
	8	#0...F	---	XFER: Scaling factor: [mm] per motor revolution #0...100 0 = unit is step -> STEPS/revolution
	9	0...3	---	UNIT: Measuring unit 1/1, 1/10, 1/100, 1/1000
2	0	0...F	---	Position controller Kp [%] 0, 10, 20, 30
	1	0...F	---	Ki [%] 40, 50, 60, 70
	2	0...F	---	Kd [%] 80, 90, 100, 110
	3	0...F	---	Auto Tuning (not yet implemented) 125, 150, 175, 200
	4	---	---	RESET -> all values reset to 100%
	5	0...F	---	Current controller: Kp [%]
	6	0...F	---	Ki [%]
	7	0...F	---	Auto Tuning (not yet implemented)
	8	---	---	RESET -> values reset to 100%
3	0	0...8	---	Baud rate: RS232, RS485 1200, 2400, 4800, 9600, 19200, 38400, 38400, 38400 CANopen 10k, 20k, 50k 100k, 125k 250k, 500k, 800k, 1M
	1	0	---	Format: 8Bit, no parity, 1 stop
		1		8Bit, even parity, 1 stop
		2		8Bit, odd parity, 1 stop
	2	#0…F	---	Device address 0 RS232 127 CANopen #0...127
4	0	#0...F	---	Acceleration: [‰] of max. value 800 Hz/ms Step-Motor 20 #1...1000 1000 Hz/ms DC-Motor
	1			Deceleration: [‰] (only for velocity profiles) >100% stops immediately 20 #1...1000
	2			Start velocity [‰] of 200/40kHz at DC/SM #0....100
	3			End velocity [‰] of 200/40kHz at DC/SM 25 #1..1000
	4			Velocity: Search limit switch [‰] of 200/40kHz at DC/SM 50 #1..1000
	5			Velocity: Clear limit switch [‰]of 200/40kHz at DC/SM 10 #1...100
	6			Acceleration [Hz/ms] for stepper motor #1...800 DC-Motor #1...1000
	7			End velocity [Hz] for stepper motor #4...40000 DC-Motor #1000...200000
5	---	---	---	not used
6	0...15	#0...F	---	Velocity-variables 0...7 [%] of 200/40kHz at DC/SM #25 1...100
7	0...15	#0...F	---	Positioning-variables 0...7 #0 +/- 10.000.000
8	0...7	#0...F	---	User-variables 0...7 #0 +/- 32.765
9	E	1	2	Flash save machine parameters

Specific Parameter Information

Step resolution
The step resolution mainly defines the running performance of the motor. Set bevore XFER and UNIT !

Running performance:
☹ less than 1000, 😐1000, ☺ more than 1000

Resonances and positioning behavior of stepper motors
The motor resonance behavior and with it the running performance can be positively influenced by increasing the step resolution. However, practical studies revealed that resolutions greater 4000 steps per revolution didn't produce significant improvement because

a.) the stepper motor is not able to deliver the step accuracy due to friction and other forces and
b.) the resonance behavior can not be further improved

Current setting

The motor current setting at stepper motors is the sum vector current. This is he geometric sum Imotor= $\sqrt{(Ia^2+Ib^2)}$ of both phase currents Ia and Ib.

In general only as much current should be set as actually is required for the application. Too high motor currents results in unnecessary losses in motor and drive and do not improve the drive.

However, it is recommended to set the nominal motor current if the application focuses on a small step angle error.

At higher step pulse rates the motor current reduces because of the motor inductance. The nominal current can no longer flow in he windings causing consequential torque loss (see motor torque curve diagrams from manufactures)

Automatic current reduction
In operating modes with pauses between movements it is useful to activate the current reduction. The motor current is reduced to approx. 60% of the set motor current. The losses in motor and drive are reduced as could be seen in following table:

current reduction	0%	to	60%
losses	100%		36%
motor torque	100%		60%

! Current reduction reduces holding torque. Assure the resulting holding torque is acceptable for your application.

The current reduction is activated, if the pulse input is inactive for more then approx. 2 seconds.

With the next pulse at the pulse input, the nominal current is set again. The time to full motor current depends on motor type and the motor voltage.

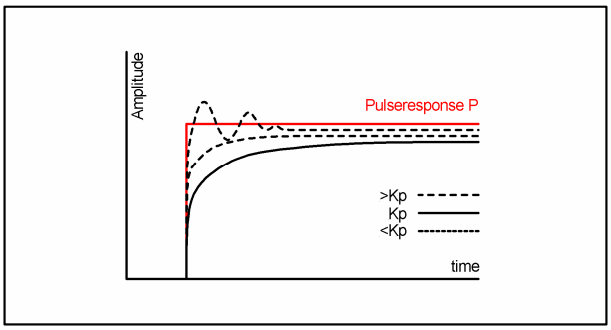
The current reduction must be activated at motor currents over 6A

Control settings:

Control parameter setting of the positioning controller and especially the current control must be done with great care.

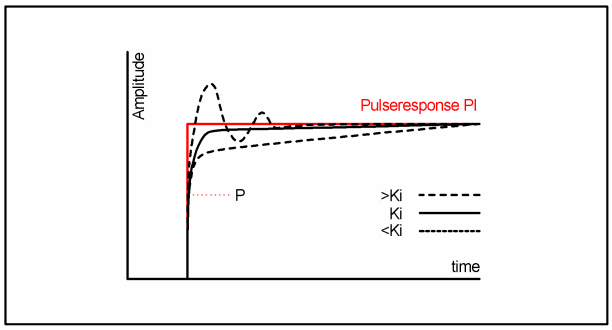
The default values are set so that a large application spectrum can be covered under normal circumstances. Modification can be done as a percentage within a given range.

Behavior of the proportional gain Kp:



A high Kp increases the stiffness of the system and results basically in a smaller offset. If Kp iss et too high the systems tends to ring which means a tendency to vibrate at higher frequency. With an optimal Kp setting a small and dampened overshoot is accepted.
The above graphic shows that a true P controller always has an offset. This is usually unwanted in drive technology. For this reason always PI-controllers are used. Its integral gain eliminates the offset after a certain time. A PI-controller is used to control the motor current.

Behavior of the integral gain Ki:

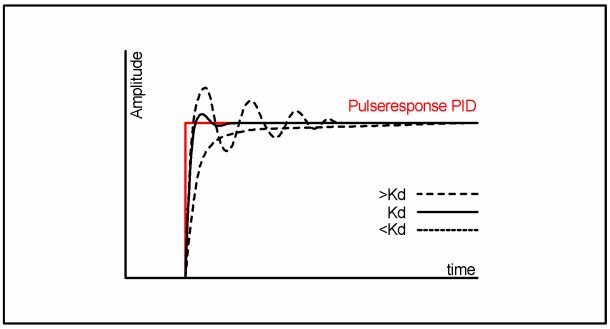


If a small Ki factor is set, it takes longer for the offset to get towards zero. On the other hand it is important that Ki doesn't dominate Kp. This results in a "Winding-Up" behavior, i.e. the system starts to vibrate continuously with lower frequency as when is "ringing". For this reason the following setting guideline should be observed:

First set the max. possible Kp with little or no Ki. Then set only as much Ki as absolutely necessary for the application.

An integral gain decreases the system dynamic. To still be able to get high dynamic and system stiffness (high Kp and Ki) a dampening element must be implemented which eliminates the tendency to vibrate. For this the PID-controller is used. It has the differential gain Kd as additional control parameter. For the positioning controller a PID-controller is used.

Behavior of the differential gain Kd:



With a Kd gain set too low the system tends to vibrate because for this controller type usually high Kp and Ki gains are set. Too much Kd results causes rough running performance and is noticed as "hissing noise". Furthermore high dampening prevents from reaching the target position quickly.

Because of the dependence of Kp, Ki and Kd an iterative procedure fort he controller setting is necessary. For this the following procedure is recommended to set the PID-controller:

1. Set moderates Kd (Default value)
2. Set moderates Kp (Default value)
3. Set moderates Ki Default value)
4. Increase Kp one step
5. Increase Kd one step
6. Repeat Step 4 and 5 as long as no tendency to vibrate (Kp) an no "hissing noise" (Kd) is observed. If the drive overshoots the target position prioritize Kd against Kp, possibly also reduce Kp
7. Only set as much Ki gain as necessary

Practical application experience revealed that the default values are practical in most cases.

Encoder evaluation:

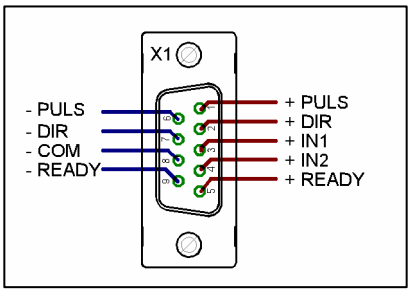
Stepper Motor:
Encoder increments= 0: Rotation monitoring disabled
Encoder increments > 0: The rotation monitoring is active. To do so the increment count per encoder revolution must be entered.

BLDC- or EC-Motor:
Encoder increments= 0: In velocity mode the speed feed back is sensed only via the hall sensors.
Encoder increments > 0:External encoder is used for position and speed feed back.

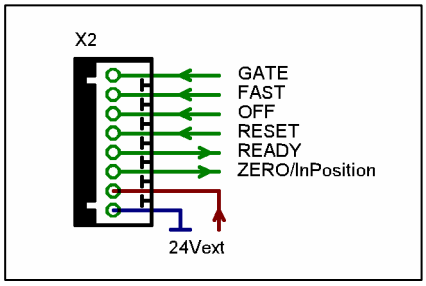
For all other operating modes the declared encoder increments must be greater than zero.

Operating mode: Pulse and Direction

Connection: User-Interface



Connection: Peripheral-Interface



Signal description

PULSE:
A step is executed with each positive signal edge. The power drive exclusively reacts on signal edges. In case of an active current reduction and pulse pauses greater than approx. 2s, the motor current is reduced accordingly.

DIR: (Direction)
The direction signal defines the sense of motor rotation. The signal must only be changed at motor stand still.

IN1: (not used)

IN2: (not used)

COM: (0 Volt input for IN1, IN2)

READY:
This output is switched on when the drive is functional. If fatal faults occur the output is switched to high impedance. Fatal faults can be: over current, over-temperature, lag error too high, etc.
This condition is latched and can only be reset with an active "RESET-Signal" or by pressing the front panel button.
The power drive indicates READY approx. 200ms after power supply is stable connected.

GATE:
The power drive ignores all input pulses if the input GATE is activated. With this function it is possible to operate multiple power drives from one pulse source. Only those motors turn that are not inhibited by the gate input.

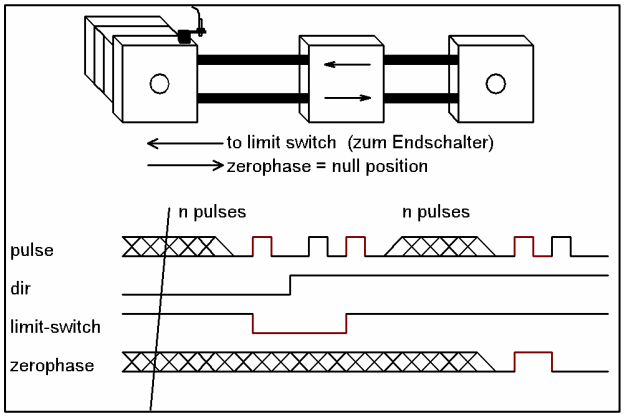
FAST:
Activating the input reduces the step resolution to ¼. Therefore the motor velocity is 4 times higher.
With active FAST signal the output ZERO can no longer be guaranteed.

OFF: (Disable drive)
When active, the motor current is switched to zero. The step counter retains its current value.

RESET (Reset fault)
Change from fault condition to operating condition. the drive switches to fault state for the following conditions e.g. motor current too high (short circuit), over-temperature etc. The power drive switches off and the corresponding LED indicated this state. The ready output is switched to high impedance. Activating the RESET input clears the fault state. The rotor position is undefined.

InPosition (only for DC-motors)
The InPosition output is set after the motor has reached the target position.

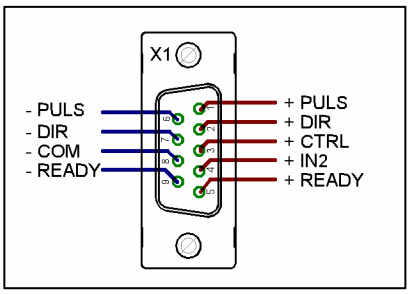
ZERO: (only for stepper motor)



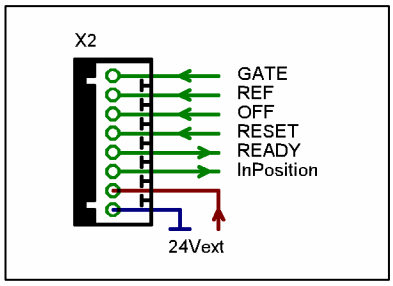
The output ZERO can be used to exactly and repeatedly find the reference point. At first carefully move towards the limit switch, then reverse the direction and move until ZERO phase is active. Be sure, the ZERO phase doesn't coincide with the limit switch hysteresis and perhaps adjust the limit switch.

Operating mode: Multiturn and Direction

Connection: User-Interface



Connection: Peripheral-Interface



General Information

This operating mode allows to move a defined distance or rotation repeatedly with a single start signal (pulse). Distance/angle can set via the ratio 1000*gear numerator/gear denominator. The gear factor, acceleration, end-frequency etc. can be set via front panel parameter input (see parameter list)
Example:
Step resolution: 4000 steps per revolution
Gear factor: Numerator= 2, Denominator= 1
Start(Pulse) => Motor rotates ½ revolution

Signal description

PULSE (START):
With each pulse the motor is started. The motion function depends here of the control inputs CTRL and DIR.

DIR: (Direction)
The direction signal defines the motor sense of rotation. The signal must only be changes at stand still. With the PULSE signal the direction signal change is executed.

CTRL: Operating mode selection
Usually a mechanical axis must first be homed with a reference move to set its zero position. This is done by activating the CTRL input.

With PULSE the reference move is started. The motor runs until the input REF at the peripheral interface becomes active. This position is then the zero position.

IN2: (not used)

COM: (0 Volt input for CTRL, IN2)

READY: (Ready)
This output is switched on when the drive is functional. If fatal faults occur the output is switched to high impedance. Fatal faults can be: over current, over-temperature, lag error too high, etc.
This condition is latched and can only be reset with an active ”RESET-Signal“ or by pressing the front panel button.

GATE: (Tor)
The power drive ignores all input pulses if the input GATE is activated. With this function it is possible to operate multiple power drives from one pulse source. Only those motors turn that are not inhibited by the gate input.

REF: (Reference switch NO)
Connection for the reference switch

OFF: (Disable drive)
When active, the motor current is switched to zero. The step counter retains its current value.

RESET (Reset fault)
Change from fault condition to operating condition. the drive switches to fault state for the following conditions e.g. motor current too high (short circuit), over-temperature etc. The power drive switches off and the corresponding LED indicated this state. The ready output is switched to high impedance. Activating the RESET input clears the fault state. The rotor position is undefined.

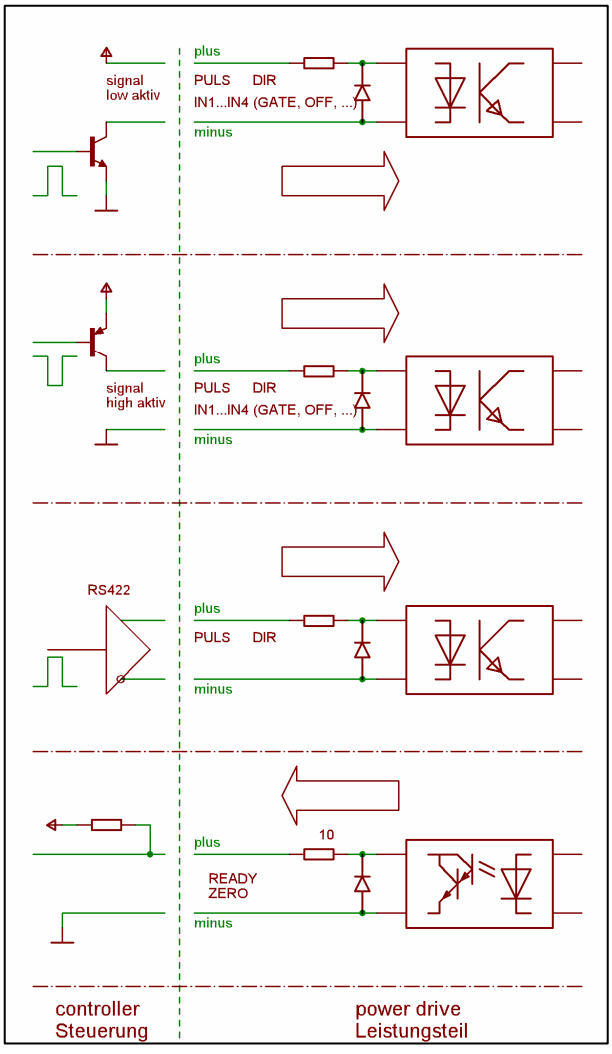
Depending on the step resolution and the direction signal, the ZEROPHASE signal is set at multiple input pulses as shown in the following table.

Steps/rev.:	ZEROPHASE after pulses
4000	80
2000	40
1000	20

The default resolution is generally 4000 Steps per revolution

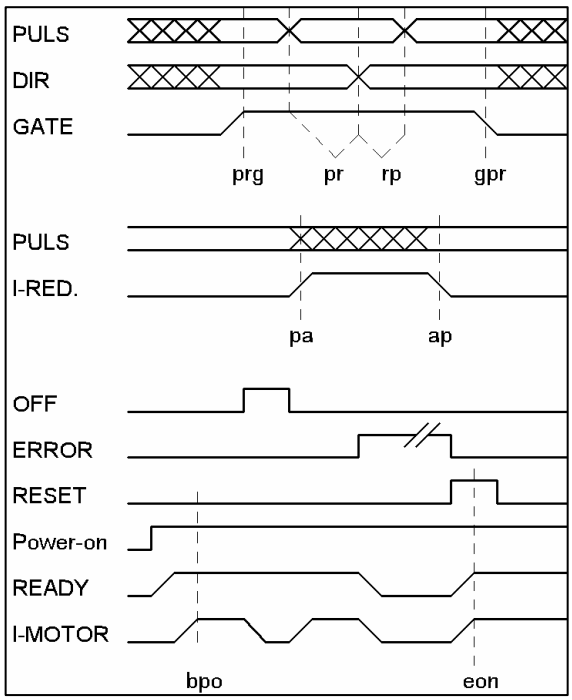
With FAST the output ZERO is no longer guaranteed.

Wiring examples:



The signal interface is completely isolated by opto-couplers. To have a wide flexibility, both inputs plus and minus of the opto-couplers are available. So its easy to drive the signal interface with high-, low- or RS422 active signals.

Timing:



! Pulse slope:	**max.**	**2µs**
! Pulse width:	**min.**	**5µs**
! Pulse pause:	**min.**	**5µs**

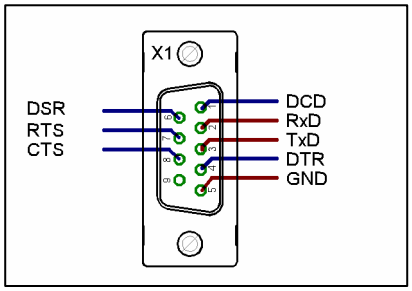
prg:	> 5µs	Gate active after Pulse/Direction
gpr:	> 1ms	Pulse/Direction active after Gate
pr:	> 5µs	Pulse before Direction
rp:	> 5µs	Pulse after Direction
pa:	< 2s	Current reduction active after Pulse
ap:	< 1ms	Nominal current after Pulse
bpo:	< 1s	Ready after Power-On
eon:	< 10ms	Ready after RESET

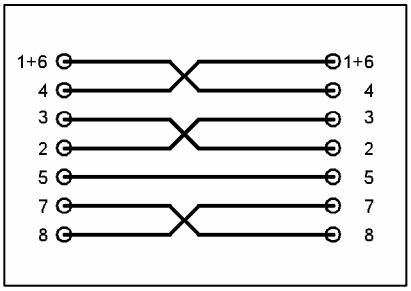
Status display L1...L4:

L1:	Over voltage	Ballast circuit active
L2:	Temperature warning	Cooling fan is switched on
L3:	with stepper motor:	Current reduction is active
	with DC-Motor:	Motor is in position
L4:	permanent on	Run Mode
	flashing slowly	Stop Mode
Lx	flashing rapidly	Error Mode

Operating mode: RS232 Serial

Connection: User-Interface:





Null-Modem Cable wiring direct connection to PC COM port

The Signals DCD,DSR and DTR are device internally connected.
Output RTS is always active (high, => send ok)
The data transfer can be stopped with Input CTS.

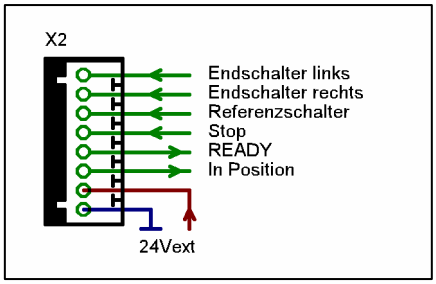
Minimal cable: RxD, TxD, GND, shield

First a buffer stores all received characters. This buffer is big enough to store an entire command string. Received character are immediately processed after reception. The characters Line-Feed [LF], Tabulator [HT] and Blank [SP] are always ignored.

A simple bus structure can be built with the RS232-interface. For this, only the internal TxD output is switched active whose device address matches the command address. All devices connected to the RS232-BUS are generally receiving, but only the device carries out commands whose address matches the command address. With this a very simple 1 out on n selection can be made.

Furthermore there is the possibility to utilize a gateway (relatively expensive) xBUS <-> RS232-BUS to connect multiple devices an operated them with a 1 out of n selection.

Connection: Peripheral-Interface:



Signal description

Limit switch left/right:
Limit switches are used to mechanically reference the axis. Usually this is done with a reference movement to define the zero position after the machine is switched on. Normally limit switches are normally closed (NC) contact versions. This means when moving towards the limit switch it is deactivated. For this the input signal must be inverted (see parameter 15xx)

Reference switch:
For the most accurate zero point definition an additional reference switch can be used. For a normally close (NC) type the signal must be inverted (see parameter 15xx)

Stop-Input:
With the Stop-Input the motor motion can asynchronously be stopped externally. This is also generally interpreted by the controller as a fault.

READY: (Ready)
This output is switched on when the drive is functional. If fatal faults occur the output is switched to high impedance. Fatal faults can be: over current, over-temperature, lag error too high, etc.
This condition is latched and can only be reset with an active "RESET-Signal" or by pressing the front panel button.
The power drive indicates READY approx. 200ms after power supply is stable connected.

In Position:
Point-to-point mode:
The output In Position is set after the axis has reached its target position.

Velocity profile mode:
The output In Position is set after the motor has reached the set velocity.

Syntax RS232-Interface:
- ASCII formatted command string
- Small and caps are allowed
- All parameter values are in decimal format
- All instruction are entered with CR

Examples: ← stands for CR (hex 0D)

Correct	MREL4000← mrel4000← MREL-1234← CLRE←

Functionality: RS232-Serial-Mode:

At power up it can be possible that the host controller is ready later and therefore errors are detected on the interface. These errors must be deleted with the command sequence CR, CLRE and CR.

Erroneous command strings are ignored, however the error remains set and must be reset with the command "CLRE" or with front panel push button.

Reference move:

Per direction there are four available reference modes. Movement towards limit switch, additionally to reference switch or index pulse or to reference switch (e.g. with indexing tables)

At first there is a movement in direction towards the limit switch if it is not active. After reaching the limit switch the drives stops and reverses direction. The limit switch is cleared with low speed until the switch is deactivated (reversing). If it is necessary to move towards an additional reference switch or index pulse a corresponding reference movement must be added. The position at the end of the reference movement is set as zero. It is absolutely necessary to add a little offset so that the limit switch doesn't trigger to quickly under normal operating condition.

Example:

Command:	**Comment:**
MREF0	Start reference move left
STAT	Request status x-times
MREL40	Limit switch offset
PSET0	Set position to zero

The search and clearing speed of the reference movement are set with the machine parameters "44" and "45".
(see parameter list)

While the reference move is active, the limit switches are monitored if they trigger within a certain window or if the assignment is correct.
The following faults are recognized:
- Wrong limit switch active
- No limit switch edge trigger found when clearing 4000 steps
- No reference switch edge trigger found when clearing within 40000 Steps
- No index pulse edge trigger found when clearing within 5000 Steps

Sequence faults:

Sequence faults can occur under the following circumstances:
- New motion command or ramping commands while current movement is still active
- Limit switch triggered
- External stop-input active
- Lag error or under voltage

After fault reset the movement can be continued with "MCONT"

Status display L1...L4:

L1:	Over voltage	Ballast circuit active
L2:	Temperature warning	Cooling fan is switched on
L3:	with stepper motor:	Current reduction is active
	with DC-Motor:	Motor is in position
L4:	permanent on	Run Mode
	flashing slowly	Stop Mode
Lx	flashing rapidly	Error Mode

Command list

Name	Argument range	Unit	Description
MABS	-99.999.999...99.999.999 ?	steps or USER-Units	Absolute movement sends back position mabs=xxxx CR
MREL	-99.999.999...99.999.999 ?	steps or USER-Units	Relative movement (absolute position is not changed) sends back position mrel=xxxx CR
MREF	0	---	Reference move left towards limit switch
	1	---	Reference move left towards limit switch plus reference switch
	2	---	Reference move left towards limit switch plus encoder index pulse
	3	---	Reference move left towards reference switch (e.g. at indexing tables)
	4	---	Reference move right towards limit switch
	5	---	Reference move right towards limit switch plus reference switch
	6	---	Reference move right towards limit switch plus encoder index pulse
	7	---	Reference move right toward reference switch (e.g. at indexing tables)
MCON	---	---	Continue interrupted movement
MVEL	-40.000… 40.000	steps/sec	Velocity profile at stepper motor resolution
	-200.000…200.000	steps/sec	Velocity profile at DC/BLDC-Motor with external encoder
	-3.000…3.000	rot/min	Velocity profile at DC/BLDC-Motor without external encoder (hallsens)
ACC	1...800 =ACC*STEPS/XFER/1000	steps/msec² USER-Units	Acceleration stepper motor
	1…1.000 =ACC*4.000/XFER	steps/msec² USER-Units	Acceleration DC motor
VSTART	1...16.000 =VSTART*STEPS/XFER	steps/sec USER-Units	Start velocity at stepper motor
	1000…16.000 =VSTART*4.000/XFER	steps/sec USER-Units	Start velocity at DC-motor
VEND	1...40.000 =VEND*STEPS/XFER	steps/sec USER-Units	End velocity at stepper motor
	1000...200.000 =VEND*4000/XFER	steps/sec USER-Units	End velocity at DC-motor
STEPS	1.000, 2.000, 4.000	steps/rev.	Step resolution per revolution (stepper motor only) set bevore XFER and UNIT
PLIM	1…99.999.999	steps	Max. absolute positioning range
PSET	-99.999.999...99.999.999	steps	Set current position to zero
IPHASE	0...10.000	mA	Stepper motor: Amplitude of the phase current 0 -> OFF DC-motor: Peak current
ILIM	1.000...10.000	mA	Motor current limitation active after 2s, (DC-motor only)
IRED	10...100	%	Current reduction active after 2s (stepper motor only) at 100% no reduction
STOP	---	---	Stop movement (can be continued with MCON)
STAT	---	---	Request status -> response -> no fault: ACK status byte CR - StatusByte.Bit0: Position not reached, if 1 - StatusByte.Bit1: Reference move active, if 1 - StatusByte.Bit2: Velocity-Mode: Velocity reached, if 1 - StatusByte.Bit6: Current reduction active, if 1 - StatusByte.Bit7: always 1 Fault: NAK Fault Byte CR (Fault no.: see table)
ENCP	---	---	Encoder Feedback encp=xxxx CR
CLRE	---	---	Clear fault
ADR	0...127	---	Device address -> only device matching machine data address is active Address 0 is default and always active
GNUM	0...1.000	---	Gear factor: Numerator ! not active, when XFER > 0
GDEN	1...1.000	---	Gear factor: Denominator ! not active, when XFER > 0
XFER	0...100	---	Scaling factor [mm] per rotor revolution at 0: -> step unit= STEPS/rev.
UNIT	1,10,100,1.000	---	Measuring unit: reciprocal input, => 10 corresponds to 1/10

Specific command list information

Use of XFER and UNIT:
XFER and UNIT are scaling factors to be able to enter movement distances not in steps but in SI-units. This correspond to parameter STEPS per revolution.

STEPS must be set before XFER and UNIT

Example:
Toothed belt axis with 70mm Hub pro motor revolution, => XFER= 70 is available.
Positioning increment should be 1/10 mm. For this UNIT is set with the reciprocal value of the unit, in this case UNIT= 10 (at 1/100 -> 100)
Now 30mm should be moved. Since the unit was set to 1/10mm, the following command must be transmitted: MREL300

Condition:
XFER * UNIT must be less or equal to parameter STEPS

If XFER is zero, no formatting is done. In this case the transmitted position is interpreted in steps and multiplied with the gear factor. GNUM / GDEN.

For simple setup and test configuration files for the Windows ® application Hyper-Terminal® can be downloaded from the website www.baur-motion-control.de. When loading the file Hyper-Terminal® is started with the required parameters for RS232 communication with the PDC controller.

Attention: COM1 is set in File „pdc_rs232_com1.ht“ as PC interface and COM4 is set in the file „pdc_rs232_com4.ht. Based on the used PC hardware it might be necessary to set other COM ports. This can simply be done in Hyper-Terminal® menu point FILE / SETTINGS.

Using the ADR address command :
Even though not common with the RS232 interface it is possible to operate multiple devices on the bus. For this each device must be equipped with it's own device address. (see machine data parameter setting) With the command ADR an address is transmitted. Each device now compares the received address with the one in the machine parameters. The device with the matching address becomes active the others don't.

Using the ADR address command :
Even though not common with the RS232 interface it is possible to operate multiple devices on the bus. For this each device must be equipped with it's own device address. (see machine data parameter setting) With the command ADR an address is transmitted. Each device now compares the received address with the one in the machine parameters. The device with the matching address becomes active the others don't.

Example:
The are 5 devices connected on the bus, that have the respective addresses 1...5 assigned.

Command	Comment
ADR1	Device 1 is immediately active
MREL40000	Motion command to device 1
ADR2	Device 2 is immediately active
MREL200	Motion command to device 2
ADR5	Device 5 is immediately active
MREL3000	Motion command to device 5
ADR1	switch back to device 1
STAT	and request status
.....	etc.:

Use of MVEL
At BLDC-Motor the operating mode depends on the rotation encoder parameter 1.4.n
Encoder increments n= 0: The velocity feed back is only sensed via the hall sensors. The argument is normalized in rotations per minute. Arguments over 300 rpm are practical.
Encoder increments n > 0: The velocity feed back is sensed via an external rotary encoder. The argument is normalized in steps per second

Use of MREL? and MABS?

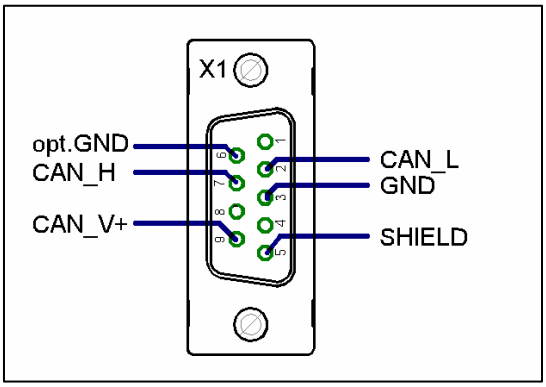
Relative movement does not change absolute position.
Before relative movement the encoder is set to zero

In general, the position request must be done in the way the last position mode was active. So, last movement was MABSxxx, request: MABS? Similar at relative motion: MRELxxx, request MREL?

Last Motion	Request	Mack Message
relative	MREL?	relative position
relative	MABS?	last absolute position
absolute	MREL?	actual absolute position
absolute	MABS?	actual absolute position

Operating mode: CANopen

Connection: User-Interface:



The signals GND and opt.GND are connected device internally.

The bus ends must be terminated with a 120 Ohm resistor (available as an accessory)

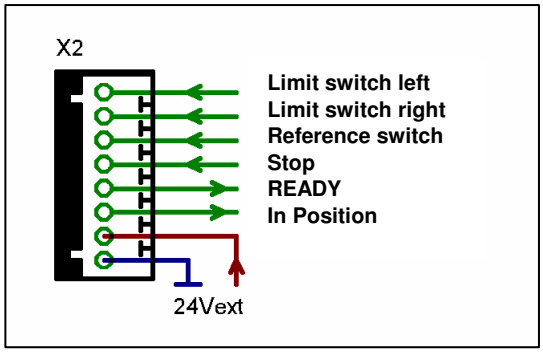
The pdc-x85 is a CANopen slave device
Object directory and EDS can be downloaded from the web site www.baur-motion-control.de

With the CANopen the pdc-x85 can be very simply controlled with a CANopen Master. For simplicity only a few objects were implemented. This was realized with a command structure ,analog to the RS232 interface, that is using a command with an according parameter value. This significantly reduces the effort to define the parameter and with it the set-up is reduced as well. Furthermore the object directory remains unchanged and doesn't have to be newly adapted every time there are new commands added to the list. The defined and standardized DS301 communication objects [predefined connection set] are sufficient for CANopen SUI.
The following communication objects are available:

COB	Node-ID	COB-IDs
NMT	---	0
SYNC	---	80_h
TIME	---	100_h
EMCY	1…127	081_h…$0FF_h$
PDO1 (tx)	1…127	181_h…$1FF_h$
PDO1 (rx)	1…127	201_h…$27F_h$
PDO2 (tx)	1…127	281_h…$2FF_h$
PDO2 (rx)	1…127	301_h…$37F_h$
SDO (tx)	1…127	581_h…$5FF_h$
SDO (rx)	1…127	601_h…$67F_h$
NMT (error)	1...127	701_h…$77F_h$

not used with CANopen SUI

Connection: Peripheral-Interface:



Signal description

Limit switch left/right:
Limit switches are used to mechanically reference the axis. Usually this is done with a reference movement to define the zero position after the machine is switched on. Normally limit switches are normally closed (NC) contact versions. This means when moving towards the limit switch it is deactivated. For this the input signal must be inverted (see parameter 15xx)

Reference switch:
For the most accurate zero point definition an additional reference switch can be used. For a normally close (NC) type the signal must be inverted (see parameter 15xx)

Stop-Input:
With the Stop-Input the motor motion can asynchronously be stopped externally. This is also generally interpreted by the controller as a fault.

READY: (Ready)
This output is switched on when the drive is functional. If fatal faults occur the output is switched to high impedance. Fatal faults can be: over current, over-temperature, lag error too high, etc.
This condition is latched and can only be reset with an active "RESET-Signal" or by pressing the front panel button.
The power drive indicates READY approx. 200ms after power supply is stable connected.

In Position:
Point-to-point mode:
The output In Position is set after the axis has reached its target position.

Velocity profile mode:
The output In Position is set after the motor has reached the set velocity.

Objects: Pdc-x85 with CANopen

To control the pdc-x85, only the following 8 objects are relevant:

- CAN-ErrorRegister [Index 1001]
- CAN-Control [Index 2000]
- CAN-Status [Index 2001]
- CAN-Command [Index 2002]
- CAN-Argument [Index 2003]
- ERROR-ID [Index 2004]
- CAN-DataId [Index 2005]
- CAN-DeviceData [Index 2006]
- CAN-UserExtin [Index 2007]

CAN-Control [Index 2000]

15 ... 9	8	7	6 ... 0
	Halt	Fault Reset	

Only bits 7 and 8 are assigned and relevant in the CAN-Control.

Fault Reset:
If the pdc-x85 is in the fault state (indicated in the CAN-status word or in the CAN-ErrorRegister), the fault state can be reset by setting the „Fault Reset“. This can also be done with the command No.16(hex).

Halt:
A movement can be interrupted with the bit Halt. This can also be done with the command No.03(hex)

! With HALT active, there is no positioning possible.

CAN-Status [Index 2001]

15.....11	10	9.....6	5	4	3	2….0
x	Target reached	x	Halt	x	Error	x

The bits marked with x are not defined and can have any state.

A changed StatusWord is automatically transmitted with the communication object PDO1(tx).

Target reached:
Is set after the drive has reached the set (target) position.

Halt:
Is set if the drive decelerates (Target position = 0) or if the drive is stopped (Target position = 1)

! HALT must be reset explicitly with the CAN-ControlWord

Error:
Indicates a general fault if 1 and can be reset with „Fault Reset“ in the CAN-ControlWord or with a command.

CAN-Command [Index 2002]
CAN-Argument [Index 2003]

The actual pdc-x85 command is specified with the object CAN-Command. Depending on the command also a parameter is required, that is transmitted in the CAN-Argument. The objects are received in PDO2(rx). A command list can be found in this chapter under „Command List“.

ERROR-ID [Index 2004]

The object ERROR-ID contains the manufacturer specific fault codes.
The fault group 1 to 4 is mapped in the high byte. The higher the number the more severe the fault is. Fault group 1 is more interpreted as a warning, fault group 4 in contrast is a fatal error that switched off the motor.

The low byte contains an error number, that specifies the error more detailed. An error number table and the error description can be found at the end of the chapter fault states.

The object ERROR-ID can be read with the SDO0 [Index 2004].

CAN-DataId [Index 2005]
CAN-DeviceData [Index 2006]

Data send back to the master after request are specified through the CAN-DataId. The Data itself is stored in CAN-DeviceData.
This structure is similar to CAN-Command und CAN-Argument. Therefore a very flexible and scaleable communication is possible.

CAN-UserExtin [Index 2007]
Reserved for User-Extensions.

CAN-ErrorRegister [Index 1001]

7	6.....1	0
Manufacturer specific	X	General Fault

The bits marked with x are not defined and can have any state.

The bit “general fault“ is set independently of the occurred error as a global fault indication. At device internal manufacturer specific fault additionally bit 7 is set.

Error flags can only be reset with the bit “FaultReset“ in the CAN-ControlWord or with the command No.16(hex).

Object mapping with CANopen

Usually CANopen transmits process data with PDOs (Process Data Objects). From the view of the pdc-x85 a differentiation is done between Send- [PDO(tx)] and Receive-PDOs [PDO(rx)]. A PDO can contain up to a maximum of 8 byte user data. The mapping defines at which byte position the user data is located. Transmitted data is always in the Little Endian Format, i.e. for a multiple byte length value always the LSB (least significant byte) is transmitted first.

PDO-Mapping of the pdc-x85

	2 Byte
PDO1 (rx)	CAN-Control

	2 Byte	2 Byte
PDO1 (tx)	CAN-Status	CAN-UserExtin

	2	1	4
PDO2 (rx)	CAN-Control	Command	Argument

	2 Byte	1 Byte	4 Byte
PDO2 (tx)	CAN-Status	CAN-DataId	CAN-DeviceData

CANopen communication objects

Definitions:
- all number values are in Hex format
- e.g. Node-ID 20 hex (equals 32 in decimal)
- from the view of the pdc-x85 Receive Transmit

NMT-Services for device control

COB-ID	Command	Node-ID

COB-ID:
- for device services always 0, therefore highest priority

Command:

NR.	Command	Changes state to
- 01	Start Remote Node	operational
- 02	Stop Remote Node	Stop communication
- 80	Enter Pre-operational	pre-operational

Node-ID

- 0	Broadcast, to all slaves simultaneously
- n	to slave with Node-ID (Address) n e.g. 20_h

NMT-Services Response

COB-ID	State

COB-ID:
- 700_h + Node-ID here 720_h

State:	Description:
- 0	Boot up
- 4	Stopped
- 5	Operational
- $7F_h$	Pre-operational

The pdc-x85 sends approximately every second a heart beat, that indicates the CANopen state of the communication.

Example of a power up sequence for pdc-x85

pdc-x85 is switched on

pdc-x85 indicates "Boot up"

COB-ID	State
720_h	0

and then
pdc-x85 heart beat indicates "pre-operational"

COB-ID	State
720_h	$7F_h$

In this state SDOs can be used to set device parameters

Master sends command "Start Remote Node"

COB-ID	Command	Node-ID
0	01	20_h

pdc-x85 heart beat indicates "operational"

COB-ID	State
720_h	5

The pdc-x85 is now fully operational via CANopen

SDO Communication Objects

Data direction: SDO-Client to pdc-x85

COB-ID	CCD	Index	SubIndex	Data
620_h	x	x	x	x
	1	2	1	4 Byte

CCD: Command Code

- 23_h - 27_h - $2B_h$ - $2F_h$	4 Byte write 3 Byte write 2 Byte write 1 Byte write	SDO write request
- 40_h		SDO read request

Data direction: pdc-x85 to SDO-Client

COB-ID	CCD	Index	SubIndex	Data
$5A0_h$	x	x	x	x
	1	2	1	4 Byte

CCD: Command Code

- 60_h - 80_h	write successful Fehler	SDO write response
- 43_h - 47_h - $4B_h$ - $4F_h$	4 Byte read 3 Byte read 2 Byte read 1 Byte read	SDO read response

Example: ERROR-ID [Index 2004] read

Read request Index 2004

COB-ID	CCD	Index	SubIndex
620_h	40_h	0420_h	00

Response to read request

COB-ID	CCD	Index	SubIndex	Data
$5A0_h$	$4B_h$	0420_h	00	Value

Example: **CAN-Control** [Index 2000] write

Value 4711_h write to index 2000

COB-ID	CCD	Index	SubIndex	Data
620_h	$2B_h$	0020_h	00	1147_h

Response after write access

COB-ID	CCD	Index	SubIndex	Data
$5A0_h$	60_h	0020_h	00	---

PDO Communication Objects

Data direction: PDO-Client to pdc-x85

COB-ID	Control	Command	Argument
320_h	X	x	x
	2	1	4 Byte

The pdc-x85 is completely controlled with PDO2 (rx). The CAN-Control [Index 2000], CAN-Command [Index 2002] and CAN-Argument [Index 2003] are mapped in the sequence as indicated above.

The pdc-x85 is automatically sent with PDO1(tx) after each change of the CAN-Status. In contrast to polling with the master, the CAN-Bus is only additionally when the CAN-Status register changes.

The command list and description can be found in the command list of this chapter.

! The command is immediately executed

Data Direction: pdc-x85 to PDO-Client

COB-ID	Status	DataId	Data
$2A0_h$	x	x	x
	2	1	4 Byte

Via PDO2(tx) the pdc-x85 sends Data back to the Master. The StatusWord [Index 2001] is part of the PDO2(tx) in general. The data send by the PDO2(tx) are identified by object DataID [Index 2005]. The Data itself are stored in object DeviceData [Index 2006]. The structure is the same as by PDO2(rx).

Example: Commands to pdc-x85

COB-ID	Control	Command	Argument

Set acceleration 1000_d for reference move

320_h	0_h	09_h	$3E8_h$

Set start velocity 1_d for reference move

320_h	0_h	$0A_h$	1_h

Note: Reference switch search speed and clearing speed from the switch are set with the machine parameters 44 and 45.

Start reference move towards left limit switch

320_h	0_h	05_h	0_h

wait until pdc-x85 indicates positioning move finished in the CAN-Status Word (Bit10) via PDO1(tx). (happens automatically at a change of CAN-Status Word)

COB-ID	CAN-Status	CAN-UserExtin
$1A0_h$	0400_h	0000_h

Set end velocity 5000_d for positioning move

320_h	0_h	$0B_h$	1388_h

to clear limit switch with 100_d increments

320_h	0_h	02_h	64_h

wait until target position reached

COB-ID	CAN-Status	CAN-UserExtin
$1A0_h$	0400_h	0000_h

set this position to zero

320_h	0_h	07_h	---

move absolute to position 4000_d

320_h	0_h	01_h	$FA0_h$

…
usw

Decimal Value $(_d)$ Hex Value $(_h)$

Example: send position back to master

COB-ID	Control	Command	Argument

send actual position back

320h	0	17_h	---

if actual or last motion was absolute, PDO2(tx) will be send as follows:

data direction: pdc-x85 to master

COB-ID	Status	DataId	DeviceData
$2A0_h$	x	1_h	x_h
	2	1	4 Byte

if DataId= 1, the data send is a absolute position

or

if actual or last motion was relative, PDO2(tx) will be send as follows:

data direction: pdc-x85 to master

COB-ID	Status	DataId	DeviceData
$2A0_h$	x	2_h	x_h
	2	1	4 Byte

if DataId= 2, the data send is a relative position

.

CANopen command list

Com.No.	Argument Range [dec]	Unit	Description
01_h	-99.999.999...99.999.999	steps	Absolute movement
02_h	-99.999.999...99.999.999	steps	Relative movement (absolute Position is not changed)
03_h	---	---	Interrupt movement (can be continued with 04_h)
04_h	---	---	Continue interrupted movement
05_h	0	---	Reference move left towards limit switch
	1	---	Reference move left towards limit switch plus reference switch
	2	---	Reference move left towards limit switch plus encoder-index-pulse
	3	---	Reference move left towards reference switch (at indexing tables)
	4	---	Reference move right towards limit switch
	5	---	Reference move right towards limit switch plus reference switch
	6	---	Reference move right towards limit switch plus encoder-index-pulse
	7	---	Reference move right auf Reference switch (at indexing tables)
06_h	1…99.999.999	steps	Maximum absolute positioning range
07_h	---	---	Current position is defined as zero
08_h	-40.000…40.000	steps/sec	Velocity profile for stepper motor resolution
	-200.00…200.000	steps/sec	Velocity profile for DC-Motor
09_h	1…800	steps/msec²	Acceleration Stepper-Motor
	1…1.000	steps/msec²	Acceleration DC-Motor
$0A_h$	1…16.000	steps/sec	Start velocity for stepper motor
	1…20.000	steps/sec	Start velocity for DC-motor
$0B_h$	1…40.000	steps/sec	End velocity for stepper motor
	1…200.000	steps/sec	End velocity for DC-motor
$0C_h$	1.000, 2.000, 4.000	steps/rev.	STEPS: Step resolution per revolution
$0D_h$	0...10.000	mA	Stepper motor: Phase current amplitude 0 -> OFF DC-motor: Peak current
$0E_h$	10...100	%	Current reduction active after 2s (stepper motor only) at 100% no reduction
$0F_h$	1.000...10.000	mA	Motor current limitation active after 2s, (DC-Motor only)
10_h	---	---	not used
11_h	0...1.000	---	Gear factor: Numerator ! not active, when XFER > 0
12_h	1...1.000	---	Gear factor: Denominator ! not active, when XFER > 0
13_h	0...1.000	---	Scaling factor XFER [mm] pro rotor-revolution at 0: -> step unit = STEPS/revolution
14_h	1,10,100,1.000	---	Scaling factor: UNIT reciprocal input, => 10 equal 1/10
15_h	---	---	Request status -> response via PDO1(tx) CAN-StatusWord In case of an error. the exact specification can be read via SDO-service and object ERROR-ID [Index 2004]
16_h	---	---	Clear fault
17_h	---	---	send back actual position to the master PDO2(tx) - when absolute position: CAN-DataId [Index 2005]= 1 CAN-DeviceData [Index 2006]= abs. Position - when relative position: CAN-DataId [Index 2005]= 2 CAN-DeviceData [Index 2006]= rel. Position

Specific Command List Information

The following commands must only be sent at stand still. (CAN-StatusWord.Bit10 = 1)

Command No.	Description
01	Absolute movement
02	Relative movement
05	Reference move
07	Set position to zero
09	Acceleration
0A	Start velocity
0B	End velocity
0C	Step resolution

Using XFER(Nr.13_h) and UNIT(Nr14_h):
XFER and UNIT are scaling factors, that can be used to enter positioning distances in SI-units instead of steps. They relate to a step resolution of parameter STEPS per revolution.

STEPS must set before XFER and UNIT

Example:
Toothed belt axis with 70mm stroke per motor revolution, => XFER= 70
1/10 mm increments should be the resolution. For this, the reciprocal value of UNIT is entered, UNIT= 10 (or at 1/100 -> 100)
Now 30mm should be moved. Since 1/10mm was set as the unit, the following command must be sent:
Move relative 300 1/10mm $12C_h$

COB-ID	Control	Command	Argument
320_h	0	02_h	$012C_h$

Condition:
XFER * UNIT must be less or equal parameter STEPS

If XFER is zero, no scaling is done. The transmitted position is then interpreted in steps and processed with the gear factor. GNUM / GDEN

Using MREL? and MABS?:
relative movement does not change absolute position

Fault handling:
If a fault occurs first the general fault flag is set in the CAN-Status Word.Bit3 [Index 2001] and in the CAN-ErrorRegister [Index 1001].

The fault state is immediately sent to the master because the CAN-Status Word is automatically sent via PDO1(tx) when a change occurs.

If more detailed fault information is needed it must be requested with an SDO service read request by sending a read request on the error identification register ERROR-ID [Index 2004]. Following that, the fault must be explicitly deleted.

A table with error numbers and their description can be found in the chapter fault states .

Example:

Fault occured, CAN-Statusword is sent

COB-ID	CAN-Status	CAN-UserExtin
$1A0_h$	0008_h	0000_h
		value always 0

Read request: ERROR-ID [Index 2004]

COB-ID	CCD	Index	SubIndex
620_h	40_h	0420_h	00

Response to read request

COB-ID	CCD	Index	SubIndex	Data
$5A0_h$	$4B_h$	0420_h	00	ERROR-ID

Depending on the fault state the master initiates actions accordingly. In any case the fault must be explicitly deleted. This can be done immediately with the PDO1(rx) ControlWord

COB-ID	CAN-Control
220_h	0080_h

or with the command No.16_h.

COB-ID	Control	Command	Argument
320_h	0	16_h	---

Error states: general

A fault is indicated by L4 not being continuously on and LEDs L1...L4 flashing rapidly. With this the corresponding LED L1...L4 indicated the according fault group. Where group 4 indicates a more severe fault than group 3 and so on.

If the controller via a serial interface (RS232, RS485, CANopen) a fault byte can be sent, that reveals more detailed information about the fault. There is a number assigned to each fault (see table fault numbers). The fault group however is not sent.

Error groups starting at level 2 are seen as fatal, i.e. the motor is possibly switched off and the corresponding outputs such as Ready etc. are deactivated.

An occurring fault state remains latched. This fault state can only be reset manually by the user and by pressing the front panel push button **->]** or with master-controller (e.g. with the command „CLRE“ at the RS232 Interface)

Error group 1: Operating problems

Display: L1, L2, L3, L4 = flash, off, off, off
Causes:
- Communication fault (Interface),
- Instruction unknown, parameter fault

Error group 2: Start up problems
Display: L1, L2, L3, L4 = off, flash, off, off
Causes:
- Low voltage, over-temperature
- Limit switch triggered, stop-signal active
- Lag error, sequence error, etc.

Error group 3: Hardware problems
Display: L1, L2, L3, L4 = off, off, flash, off
Causes:
- Limit switch not found, wrong assigned,
- No index pulse, no reference signal
- Over current, short circuit in the drive stage
- Encoder problems, hall sensor problems

Error group 4: System problems
Display: L1, L2, L3, L4 = off, off, off, flash
Causes:
- ROM-, RAM-, stack, etc.

Table of Fault Numbers:

No.	Assignment:	Cause:
00	no fault	
01	---	---
02	System fault	---
03	RAM fault	---
04	ROM fault	---
05	Program fault	Error in sequence program
06	Parameter fault	Machine data fault
07...10	---	---
11...13	intern	---
14...19	---	---
20	Power drive defective	Short circuit in power drive
21	Over current	Motor defective or blocked
22	Hall-Sensor	Faulty wiring, broken cable, defective sensor, EMI -problems
23	Hex-Schalter	Mechanically defective
24	Input IN1	Doesn’t switch
25	Input IN2	Doesn’t switch
26	Input IN3	Doesn’t switch
27	Input IN4	Doesn’t switch
28	Input IN1...IN4	Wrong assignment, i.e. at limit switch- reference switch or stop input
29	Index pulse (Encoder)	Doesn’t switch
30	Encoder	Wiring problem, broken cable, encoder defective
31...39	---	---

40	Command buffer overrun	no CR detected within 16 characters, wrong interface format
41	Command unknown	Wrong instruction, noise on cable
42	Parameter fault^^	Parameter range exceeded
43	---	---
44	Command overrun	Previous instruction must be finished first
45	Send error	---
46	Receive error	Interface format, noise on cable
47...49	---	---
50...59	---	---
60	Over temperature	Insufficient cooling Current reduction active? Optional heat sink ?, Forced air cooling ?
61	Lag error	Offset too high between set and actual position Control parameter setting ? more Kp, more integral gain Supply voltage too low, End velocity too high Acceleration forces too high (ramp parameter), process forces
62	Low voltage	Power supply under powered Motor voltage was switched off
63	---	---
64	WatchDog	Safety timer trigger time expired (controller hangs up)
65...66	---	---
67	Program download	Sequential program could not be loaded
68	Movement towards limit switch	only allowed at reference move, otherwise
69	Stop-Input was active	Asynchronous external interruption
70...79	---	---
80...89	---	---
90...99	---	---

Trouble shooting:

Motor has no holding torque despite present supply voltage
- Defective power drive fuse
-The motor voltage is below 24Vdc
-Signal inputs “off” is active
-The over temperature monitoring is still active
- A non-valid step resolution is selected

No motor holding torque but supply voltage is OK
- Defective power drive fuse
-The motor voltage is below 24Vdc
-Signal inputs “off” is active

Immediate fault indication after power on
- Defective power drive
- Motor short circuit
- The temperature is still over 70 degrees Celsius

Sudden “crackling“ noises in the motor
- Low motor supply voltage (<24Volt)
- Supply volage cable cross section is too small

The motor doesn’t reach the set speed but starts
- The motor voltage is too low
- The motor current was set too low
- The acceleration ramp was set too high
- Peak torques withing the movement distance
- Motor wires are too long or too small cross section
- Power supply is not powerful enough
- Control parameters not optimized

The motor “loses“ steps and drifts
- (Pulse/Direction) signal amplitudes are too low
- Signal cable noise is too high (shielded cables?)
-The wiring concept is not optimal (system ground)
- The mechanical shaft coupling has play

Motor vibrates
- Start/Stop-frequency too high
- Motor windings are connected wrong
- Encoder or hall sensors wired wrong
- Broken cable
- low step frequency at fullstep and no load connected
- The motor current is set too low

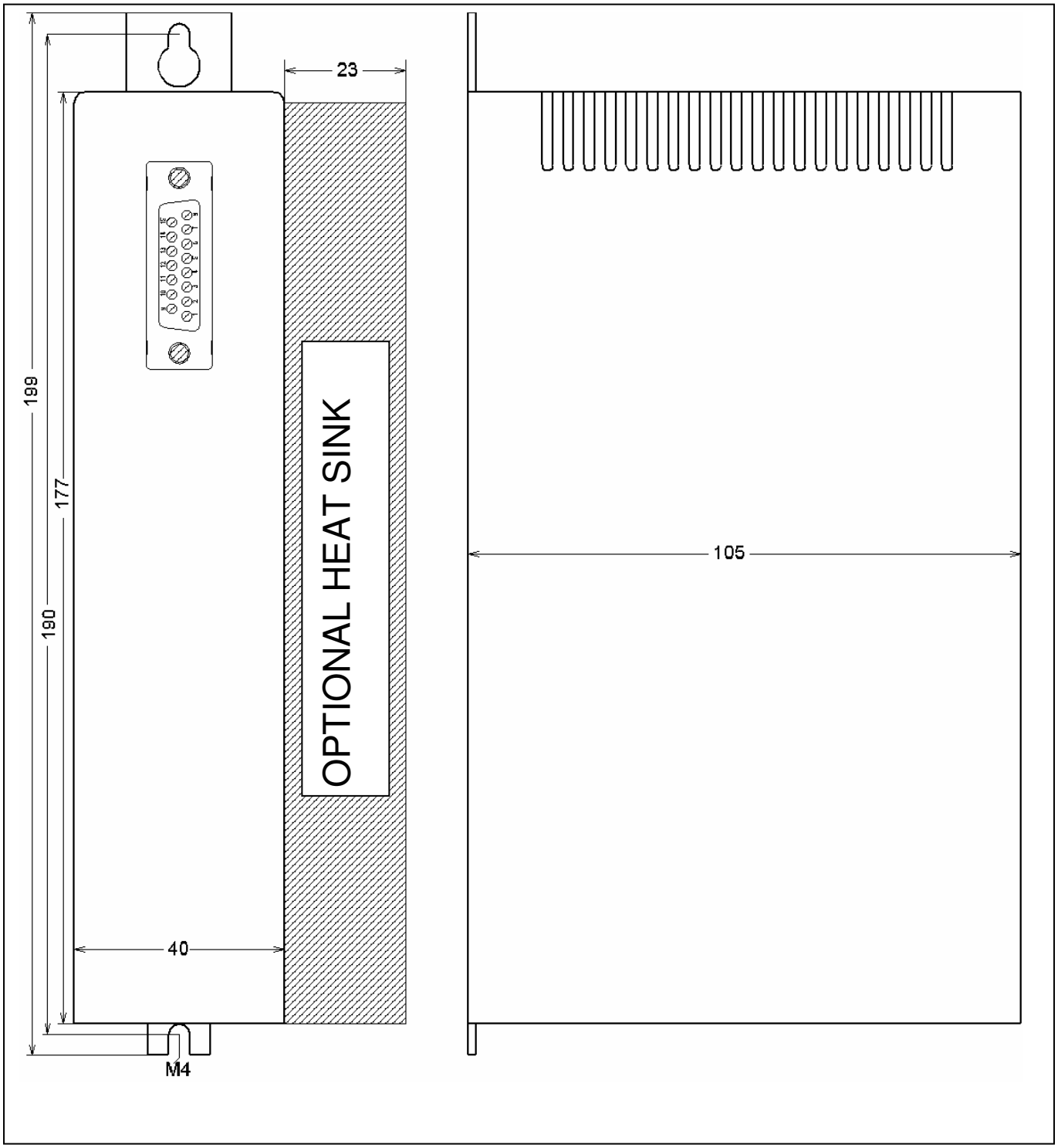
The motor is hot
Up to 85 ° Celsius should be no problem

Poor step angle accuracy in micro step operation
- Motor inductance is too high
- Motor is operated far below nominal current
- Motor is operated above nominal current

Noise at certain step positions
- Motor voltage too high at low motor current
- Motor inductance is too low

Immediate fault message after motion command
- Limit switch is active
- Lag error, encoder problem

Dimensions



Technical Specifications

Supply:	Type:	80V, 130V
absolute max. voltage:		85V, 135V
minimum voltage:		24V, 60V
recommended voltage:		80V, 130V
Voltage ripple:		< 2,0Vss
Inrush current:		< 4,0A
Fusing:		5,0Amt
Power supply capacitor:		>6800µF
Supply cable cross section:		0,75mm²
Distance to power supply capacitor		<0,3m

Motor connection:		
Cable cross section:	<4A	>0,75mm²
	>4A	>1,00mm²
Cable length:		<10m

Interface X1:	galvanic isolated	
Pulse, Direction, IN1, IN2		
Input voltage	low:	< 1,0V
	high:	>3,5V
	max.:	28V
Input current:	max.:	16mA
Pulse length:		>5µs
Pulse pause:		>5µs
Pulse edge:		<2µs
Ready		
Switching voltage:		<30V
Impedance:		<15 Ohm
Switching current :		< 50 mA
Load:		non-reactive

Interface X2:	galvanic isolated	
IN1...IN4		
Input voltage	low:	< 1,0V
	high:	>3,5V
	max.:	28V
Input current:	max.:	18mA
Pulse length:		>1ms
Pulse pause:		>1ms
OUT1, OUT2		
Switching voltage:		<30V
Impedance:		<15 Ohm
Switching current :		< 50 mA
Load:		non-reactive

Interface X3:	galvanic isolated
Encoder supply	5V, max. 70mA
Encoder A, B, I	TTL-level
	max. 100kHz
Hall-Sensors	5V, >13mA

Temperature monitoring:	
Cooling fan automatic active:	> ca. 60°
Temperature safety shut off:	> ca. 70 °

Environmental conditions:	
Operating temperature range:	0...40°C
Storage temperature range:	-10...60°C
Protection class:	IP30
Pollution degree:	2, no condensation
Weight:	0,5kg

EMI / Noise suppression:

Noise field strength:	EN55011B
ESD:	4kV
Burst:	IEC-Level 4 / 2,5kHz

GENERAL INSTALLATION REQUIREMENTS

The device housing[1] must be grounded separately. In most cases a wing nut on the front panel or another grounding connection is available. Each component must be grounded wit a separate grounding wire at a central ”grounding point“. This is usually the machine bed or a grounding rail inside the electrical cabinet.

Before installation and setup make sure that the required drive power is sufficient for your application and that the maximum values are not exceeded.

Mounting orientation is vertical, make sure air intake[1] and cooling slots are not blocked.

Only shielded motor cable must be installed. For identical potential between motor flange and power drive)short distance) the shield is grounded on both ends. Otherwise it is recommended to ground only the device end and that the shield on the motor end is ground connected galvanically isolated via a capacitor.

In general the ground potential difference must be in the range of only a couple mV.

For symmetrical motor cables such as with 2 phase steppers twisted pair wires are recommended per circuit.

Signal cables must also be shielded. Twisted pair wires are recommended per circuit.

The ground potential common point should be located directly at the housing or the mounting point of the power drive.

Signal cable and motor cable must be separated. Long parallel cable installation must be avoided. Cable crossings (if necessary) should be installed vertically.

Check all device settings for validity..

SAFETY AND PROTECTION REQUIREMENTS

The installation of the device must only be conducted by an educated, trained and experienced expert (electro). The local guidelines for safety, installation of electrical and mechanical systems and EMI must be observed.

Unintended operation and faulty installation of the device can lead to personal injury (incl. the possibility of death)
and the device as well as other external components can be damaged or an excessive pollution of the environment can occur.

Operation is only permitted with the mounted housing[2]. Because of eventually present high voltage the device must not be opened (also not after a long period of idle time). Make sure children have no direct access to the device.

No technical modifications of the device are permitted.

The device housing[3] must be grounded separately. In most cases a wing nut on the front panel or another grounding connection is available. The device must be grounded prior to the installation.

Under no circumstances live of functional connectors must be removed or connected. All installations must be conducted in the powerless de-energized state.

Device operation in damp, humid environment or with present spray water is not permitted.

[1] if available

[2] not with open frame (only PCBs)

[3] if available