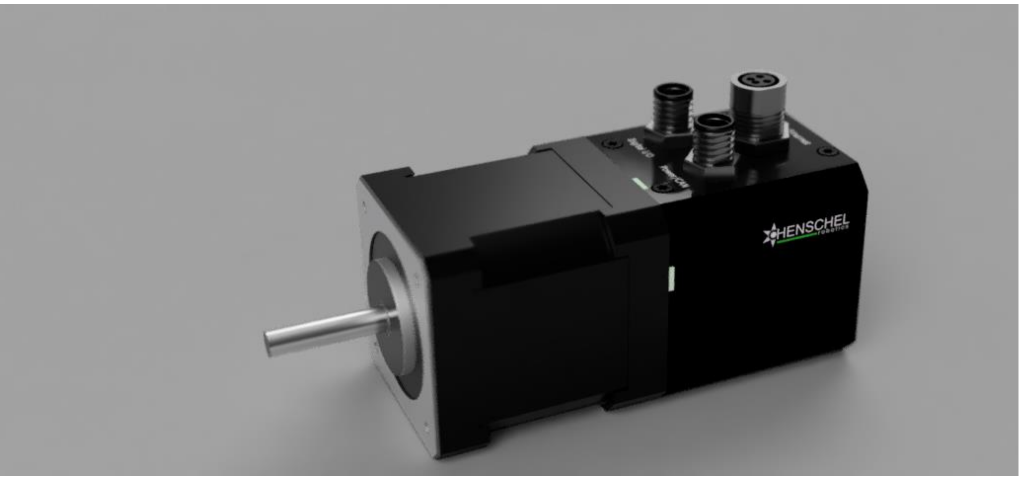


HDrive17/23-ETH-i Servo drive Manual

The Henschel-Robotics "HDrive17-ETH-I" is a compact servo drive with an inbuild webserver. The drive can be controlled by Ethernet. Only a few lines of code are needed to achieve a motion.

The direct drive is based on a high pole count, bipolar stepper motor, controlled by a sophisticated field-oriented control (FOC). The servo drive comes with the following features:

- Closed loop
- High dynamics
- High efficiency
- Fast communication
- Automated internal CAN-Bus to attach multiple slave motors to each HDrive.

The Servomotor can be driven in these three major modes: Position, speed and torque. The position data is acquired by a 14 Bit magnetic encoder system which is factory pre-calibrated. The system position accuracy is +/- 0.1°.

Company

Henschel-Robotics GmbH
Mulchlingerstrasse 67,
CH-8405 Winterthur

Mail: info@henschel-robotics.ch
Web: http://www.henschel-robotics.ch

Purpose of this document

This document aims to commission a HDrive by a technician. Please read this document carefully. The Liability for consequential damage and consequential errors is excluded. The applicable standards and regulations must be observed when installing the device.

Document history

Version	*Date*	*Changes*
0.0	01.09.2020	Document creation
0.1	20.04.2021	Updated Drawings for Nema23-ETH-I Servo drive
0.2	06.10.2021	Updated Homing section regarding Firmware 2.49 and above
1.0	28.10.2021	Final review
1.1	10.04.2022	Corrected Reset Pins for HDrive17-ETH and HDrive17-ETH-I
1.2	05.06.2022	Added new Binary CAN Ticket with additional information like slave-torque and speed

TABLE OF CONTENTS

1 Overview....4

1.1 Preface.... 4

1.2 Variants and order code.... 5

1.3 Certificates 5

2 Commissioning 6

2.1 Mechanical installation 6

2.2 Electrical installation 7

2.3 Configure the drive.... 7

2.4 Pin out 8

2.5 Host network configuration 10

2.6 Communication Interfaces 12

3 How it works....13

3.1 Control concept.... 13

3.2 Motor calibration 15

4 Ethernet Communication 16

4.1 Commands to the drive.... 16

4.2 Messages from the drive to the host computer 23

4.3 Error States.... 27

5 Firmware Upgrade....28

5.1 Website Upgrade.... 28

5.2 Factory Reset.... 28

6 Object Dictionary....29

6.1 Main keys 29

6.2 Sub keys.... 29

7 Technical data....32

7.1 Default values (Factory reset) 32

7.2 Typical electrical values.... 32

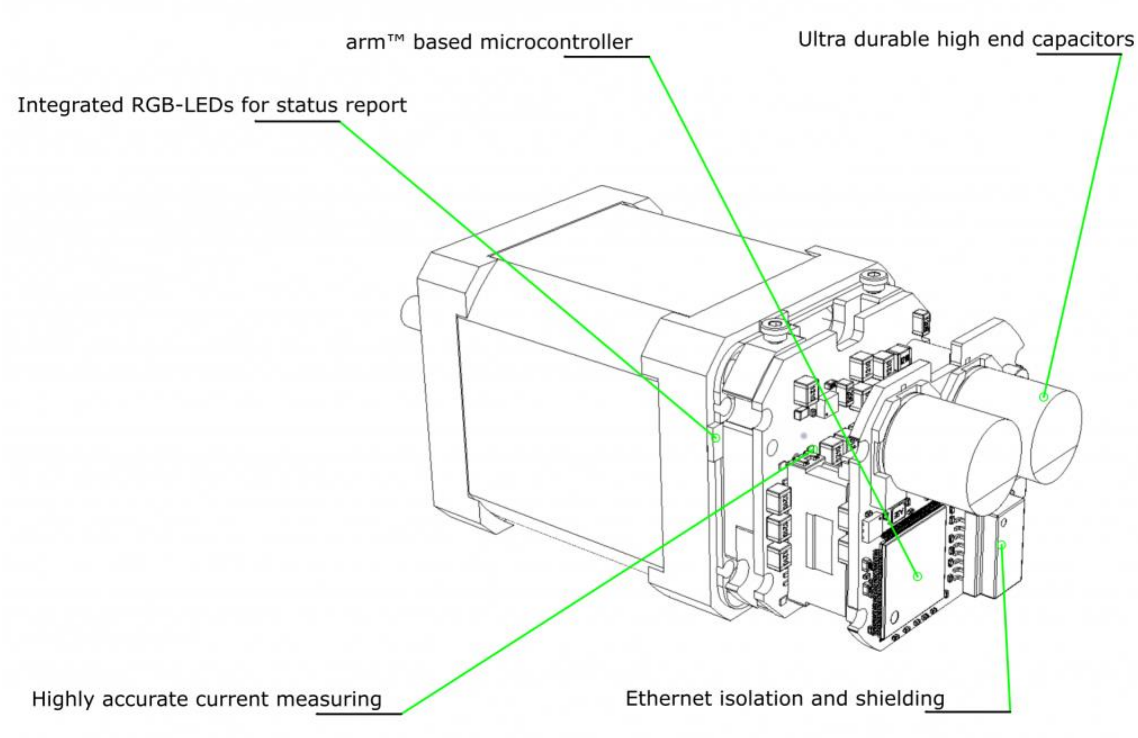
7.3 Digital I/O 33

7.4 Digital output.... 34

7.5 Torque curve, weight, and dimensions 34

1 Overview

1.1 Preface



Picture 1 The HDrive17-ETH-i

The Henschel-Robotics GmbH "HDrive" is an integrated servo motor based on a multi-pole stepper motor. The integration of the drive includes the power unit, the communication module, and a position sensor to acquire the rotor position. The integrated electronics are based on an ARM™ micro-controller and contain, among other functions, a comprehensive axis planner which can work as a position, speed or torque controller.

1.2 Variants and order code

The HDrive is available with or without a gearbox. The product names are as follows:

HDrive**AA**-BBB-**i**-**CCCC-P**

Symbol	*Description*
AA	17 or 23 – flange size of the motor
BBB	ETH for Ethernet variant ETC for EtherCat
CCCC	GP15 or GP27 for planetary gearbox
P	P for precision gearbox empty for standard gearbox

Table 1: Motor variants

1.3 Certificates

The HDrive successfully passed the following EMC standards on May 30, 2017:

Standards	*Description*	*Result*
EN 61000-6-3:2011 IEC 61000-6-3:2011 (ed2.1)	Electromagnetic compatibility (EMC) – Part 6-3: Generic standards - Emission standard for residential, commercial and light-industrial environments	Pass
EN 61000-6-2:2016 IEC 61000-6-2:2016	Electromagnetic compatibility (EMC) – Part 6-2: Generic standards - Immunity for industrial environments	Pass
EN 61326-1:2013 IEC 61326-1:2012 (ed2.0)	Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 1: General requirements	Pass

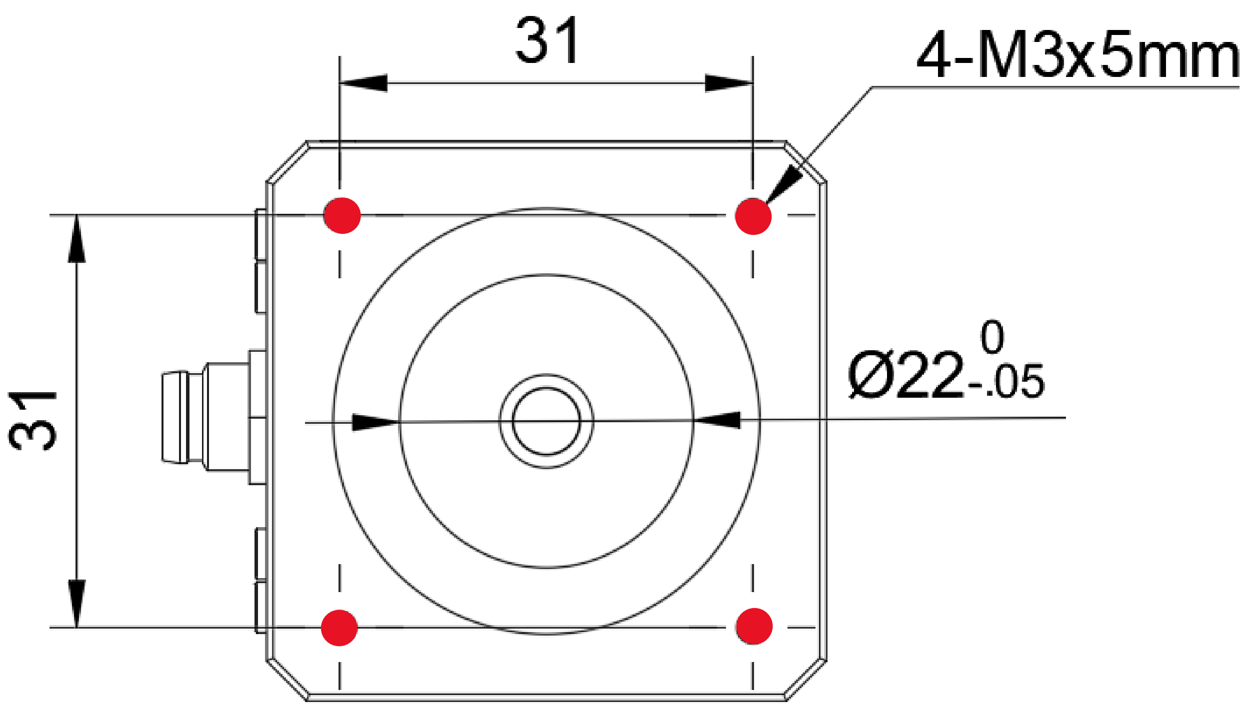
Table 2: EMI certificates

2 Commissioning

This chapter contains information about mechanical and electrical setup to commission a HDrive servo.

2.1 Mechanical installation

The motor should only be attached by the four M3 threads (depth, 4.5mm) on the front (near the motor shaft). Also use the 2mm deep ring with a dimension of $22\ {}^{0}_{-0.05}\ mm$ to contain the centricity of your application to the motor.



Picture 2: Motor flange

2.2 Electrical installation

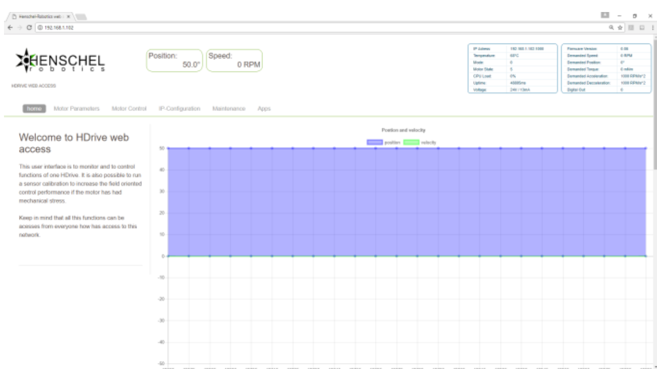
The permissible operating voltage of the HDrive servo motor is 12V to 24V. A capacitor of at least 4.7mF / 63V can be attached close to the motor to temporarily store overvoltage that can arise during braking (generator operation).

CAUTION: Operating voltages of more than 28V, as well as their polarity reversal, can destroy the motor. Never disconnect lines when they are live.

Commissioning must be carried out by technically trained personnel.
In addition, the applicable regulations must be observed.

2.3 Configure the drive

The drive has an integrated web server. All motor data can be viewed in real time using the web browser. In addition, all parameters, such as controller settings, current, acceleration ramp, etc. can be configured. The web browser “Google™ Chrome” or “Apple™ Safari” is recommended for the best visualization.



Picture 1: HDrive17-ETH Web interface

CAUTION: The web GUI is intended to monitor the motor during commissioning. In addition, the Servo-drive can also be operated from the GUI. Make sure that the motor shaft is free to rotate when you use the GUI.

The implemented web server is not protected against unauthorized access. This must be considered at network level.

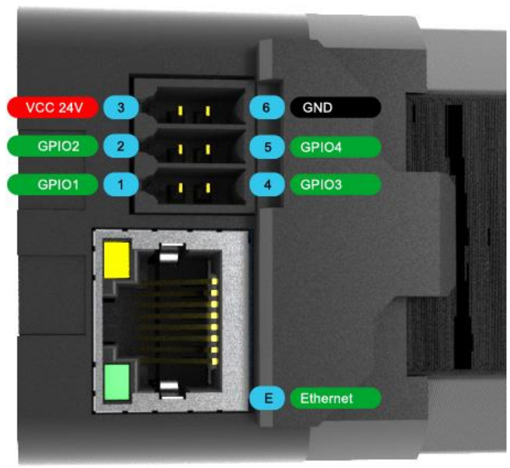
If the web GUI is open in "Motor Control" operating mode, the motor does not respond to external commands

---

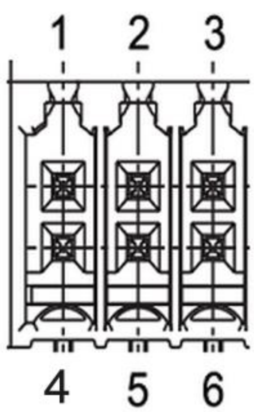
2.4 Pin out

There are several drives available from Henschel-Robotics GmbH. Find below the pin-out of the “HDrive17-ETH”, and the “HDrive17-ETH-**i**”.

2.4.1 HDrive17-ETH



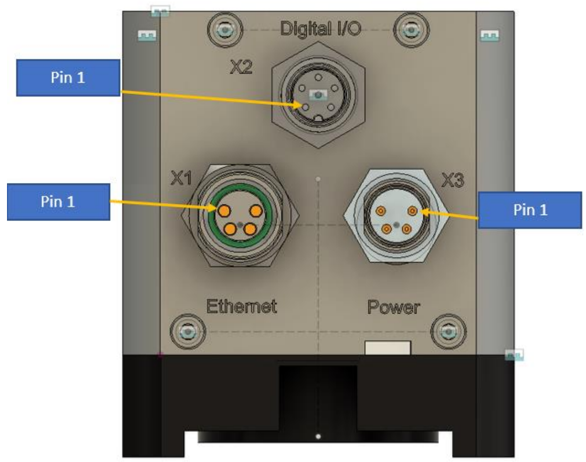
Picture 2: HDrive17-ETH Sideview



PIN	SIGNAL	FUNCTION
1	GPIO 1	Enable Signal, Step/Dir or PWM or Hardware Reset
2	GPIO 2	Step in, PWM-torque in, PWM-Position in, CAN high
3	VCC	Positive Supply voltage (24V)
4	GPIO 3	Limit Switch in, PWM position Out
5	GPIO 4	Dir in, PWM demanded torque out, PWM actual torque out, CAN low
6	GND	GND

Table 3: Connector Pinout

2.4.2 HDrive17-ETH-i



M8	*X3*	*X1*	*X2*
Type	M8 / male	M8 / female	M8 / male
Coding	uncoded	uncoded	A-coded
Description	Power und CAN	Ethernet	Digital I/O
Pin count	4	4	6

Pin function

X3:

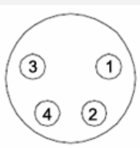
PIN	function
1	VCC
2	CAN_H
3	GND
4	CAN_L

X1:

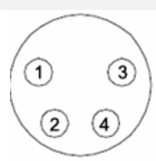
PIN	function
1	ETH TX+
2	ETH RX+
3	ETH RX-
4	ETH TX -

X2:

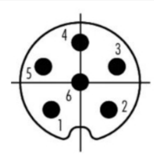
PIN	function
1	VCC (100mA)
2	Digital I/O-A
3	GND
4	Digital I/O-B
5	Digital I/O-C
6	Digital I/O-D



Front view male plug, device side



Front view female plug, device side



Front view female plug, device side

Cable suggestions

X3: e.g.: Beckhoff ZK2020-3200-0xxx

X1: e.g.: Beckhoff ZK1090-3191-0xxx

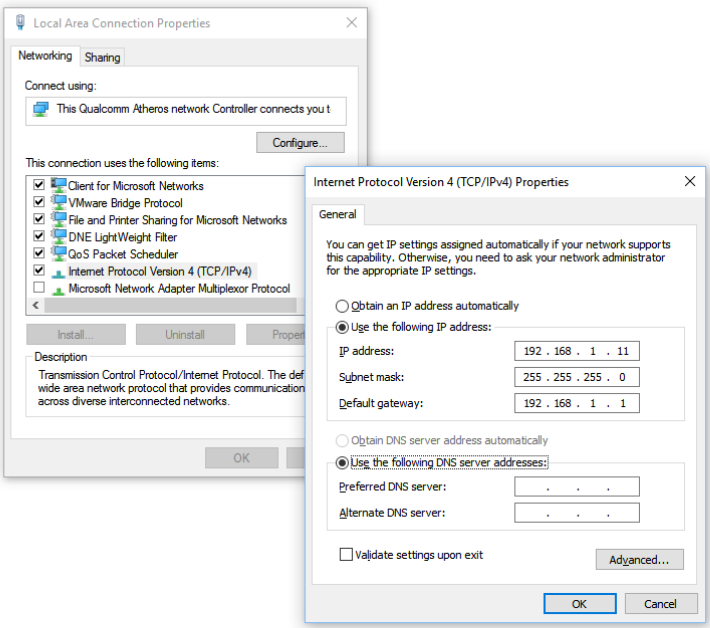
X2: e.g.: Phoenix Contact Art. Nr.: 1522396

2.5 Host network configuration

To configure the servo drive, it is necessary to connect the HDrive to an Ethernet switch or directly to a PC with an Ethernet cable. To establish communication, the host (PC) network adapter must be configured with the same network ID as the motor.

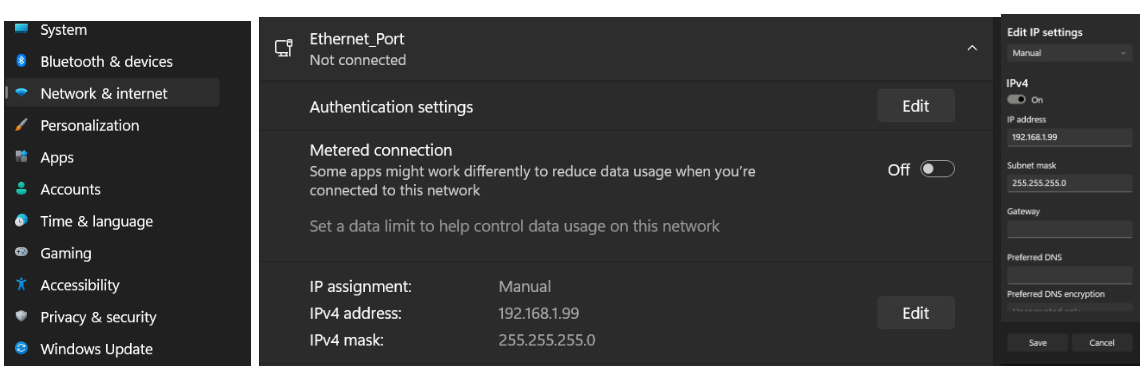
If the motor is brand new or factory reset, it has the IP address 192.168.1.102. The host PC then needs an IP address between 192.168.1.1 and 192.168.1.254. But not the same as the HDrive (192.168.1.102).

A configuration to change the IP address in Microsoft ™ Windows would be as follows:



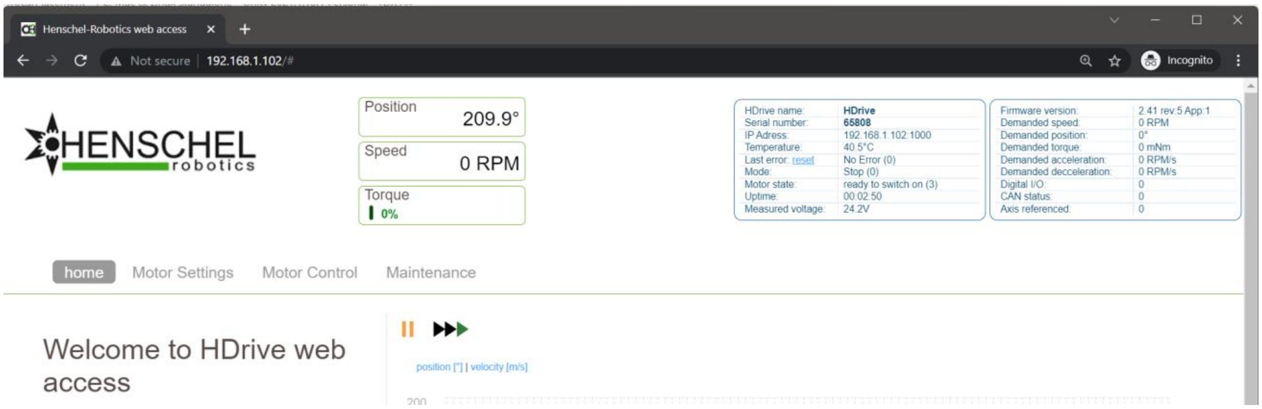
Picture 3: Example to change the IP-Address of a network adapter in Microsoft™ Windows

On Windows11 go to settings, choose “Network & internet”, and click edit on the IPv4 address.



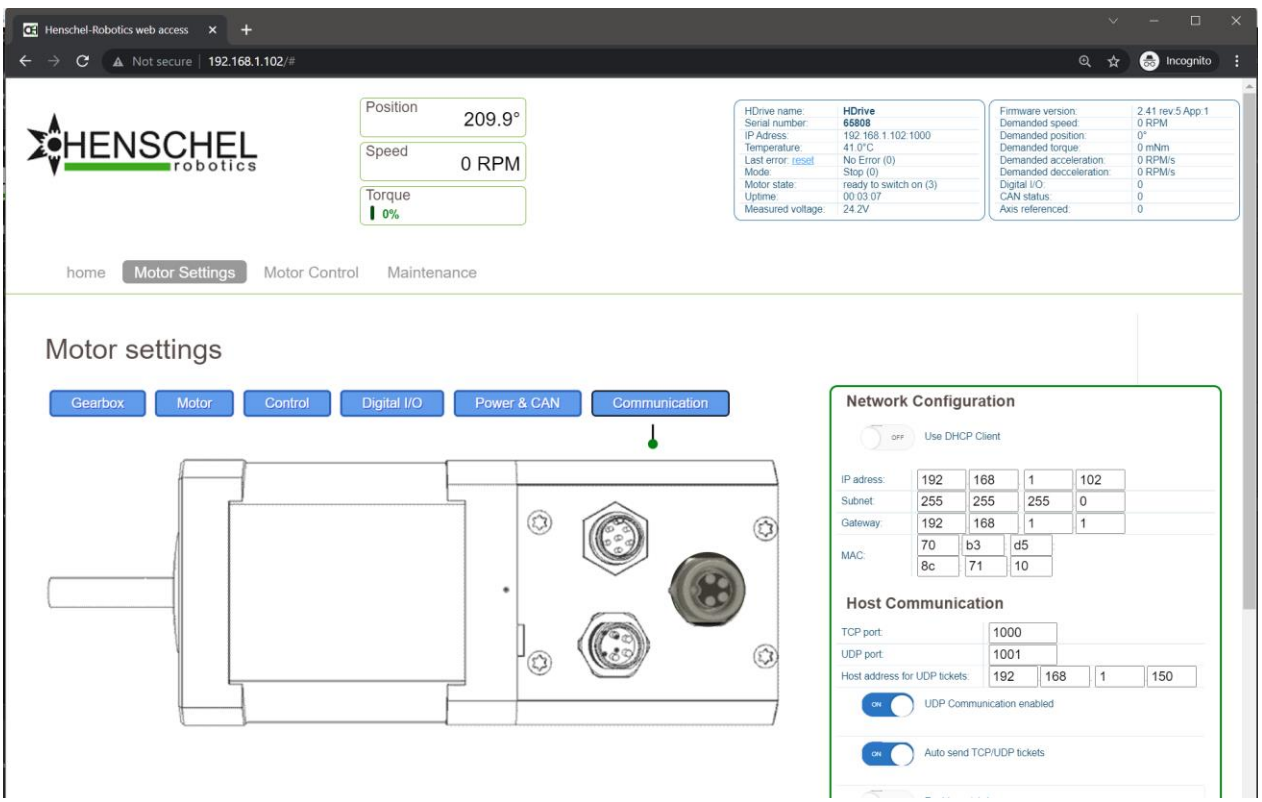
The choose “Manual” mode and enter an IP-Adress e.g. 192.168.1.99

After the host IP address has been changed, then the IP-Address of the drive can be entered in a web browser. The drive is responding with its Web-GUI on port 80 (Standard HTTP).



Picture 4: Example insert the IP-Address into the address bar of your browser

After the first connection the IP-Address of the motor can be changed as required in your network.



Picture 5: Example, change of the IP-Address of the drive

Don't forget to change back your Host-IP after the drive IP has been changed

2.6 Communication Interfaces

Each motor can be operated over a variety of communication channels. These channels can be configured by the Web-interface of the drive.

2.6.1 Ethernet

Ethernet is the main interface to the drive. It is used to configure the drive and cannot be turned off.

2.6.2 Digital I/O

It is possible to also use a digital I/O interface either with 5V or 24V logic. The motor then can be controlled without ethernet in Step/Dir, Torque (PWM), or Position (PWM) mode.

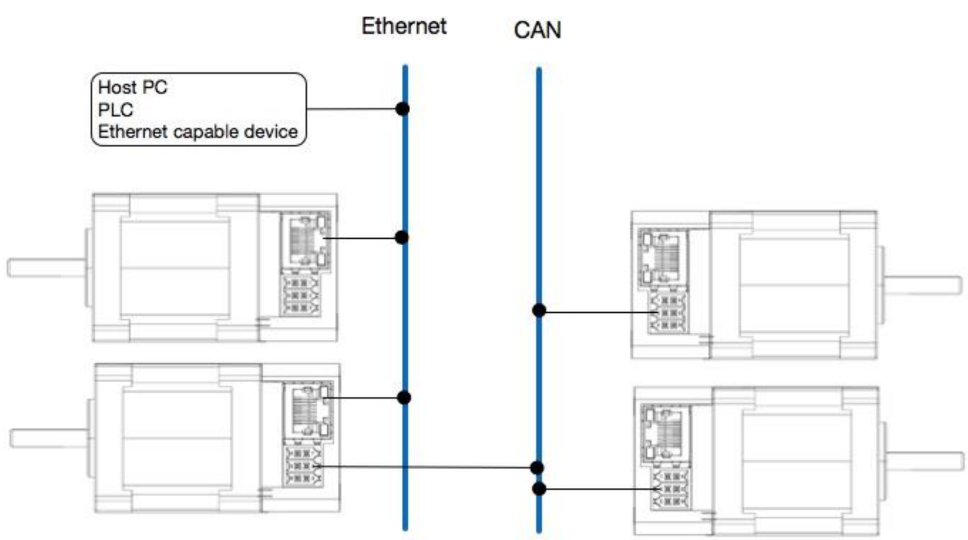
2.6.3 CAN

Every HDrive can be configured as a CAN-Slave or Master. As a Master the drive can control up to 8 additional HDrive servo motors configured as CAN-Slaves. This can simplify the wiring effort for an installation and increases the real-time behaviour to sync multiple drives.

Not all communication channels are fail-safe. In the event of a communication failure, the higher-level controller must therefore handle the error

2.6.4 Bus Topology

In the following example, two HDrive servos are controlled via Ethernet. One of these motors is also configured as a CAN-Master and generates a CAN-Bus. In the example, two further drives configured as CAN-Slaves are connected to the 2 wire CAN-Bus. The high-level control can control all motors without being connected to the CAN bus itself. The slave motors on the CAN-Bus can be controlled by “slave drive commands” through the master motor. For this purpose, each motor configured as a CAN-Salve receives an identification number from 1-9.



Picture 6: Example of operating several HDrive servomotors in a network

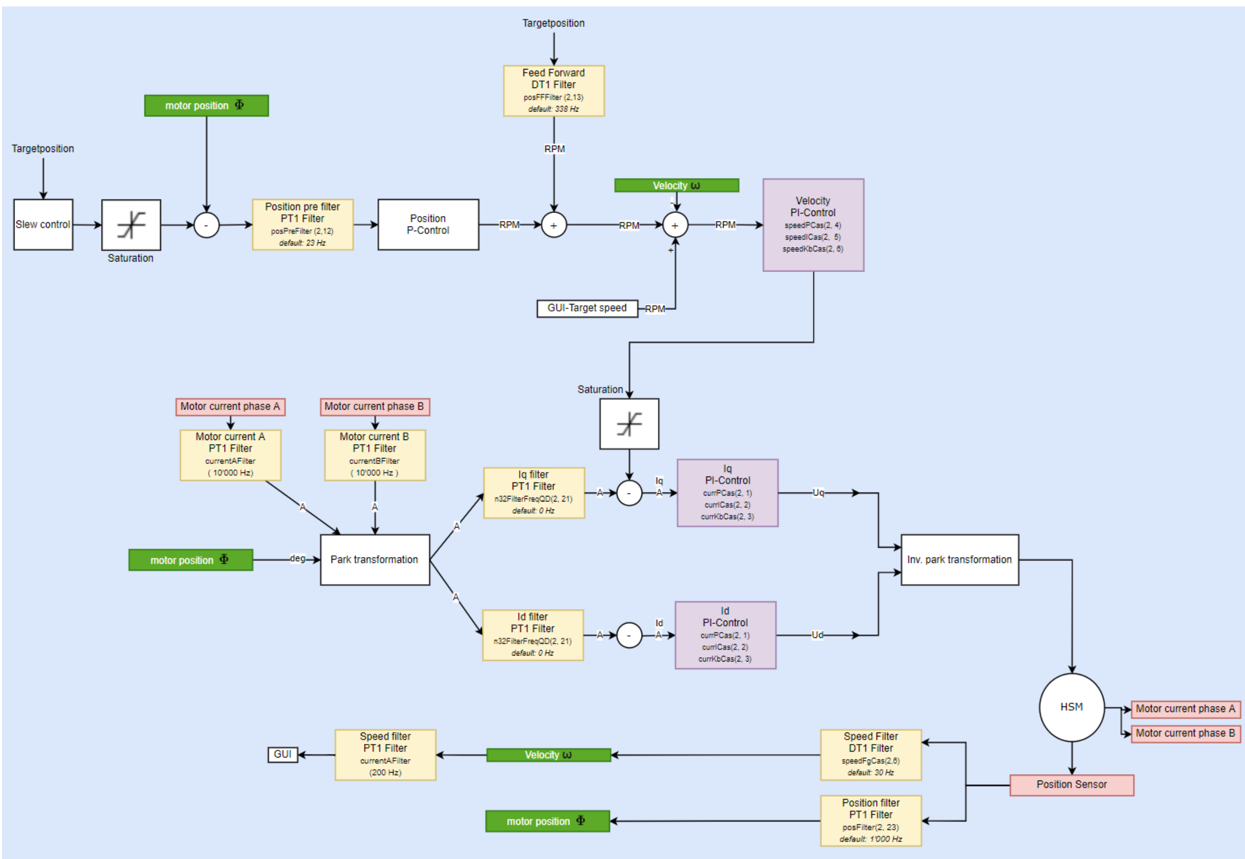
3 How it works

The drive contains several time-discrete control algorithms. With the HDrive the torque, position, speed controllers can be adjusted separately. All control loops are fully documented here.

⚠ *The open and freely parameterizable control architecture must be carefully adjusted to the respective application. It is possible to damage the drive and the attached load if the control parameters are not tuned correctly.*

3.1 Control concept

The used stepper motor is controlled in a field-oriented manner. This means that the torque-generating as well as the reactive current are regulated separately. The control architecture is implemented as followed:



Picture 3: Control architecture

The drive must be tuned for every application attached to the motor. The dynamic performance of the drive is mostly depended on the inertia of the load and their dynamic behaviour. The position controller can be set in a frequency scope through the web GUI. All mandatory control parameters then are getting calculated automatically. It is possible to set the bandwidth of the current, speed and position controller separately.
Always try to keep the frequencies as low as possible to ensure a silent and efficient operation.
There are several predefined frequencies to give you a start point for tuning your application.

The drive can be operated in three different control modes:

- **Current control**
  A target current is set with the "Current" parameter. The current control then checks that this current is maintained. The current regulator works with a cycle of about 20 kHz. In this mode, position or speed controllers are deactivated.
  Use this mode to create a constant torque on the motor shaft.

- **Speed control**
  This mode controls the target speed of the motor shaft. It tries to maintain the selected speed also when the load is changing.

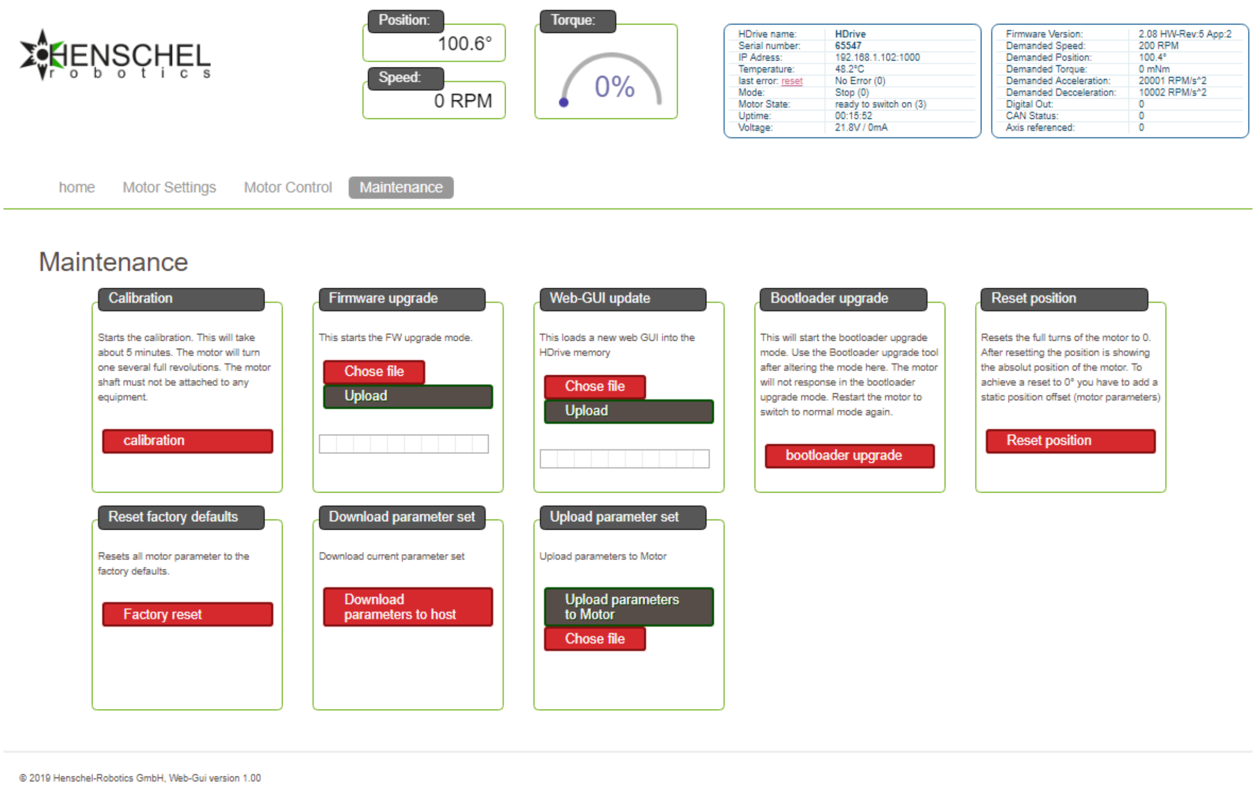
- **Position control**
  The position controller maintain the motor shaft to settle at a selected target position, considering the maximum torque, accelerations and speed.

3.2 Motor calibration

The motor is calibrated in the factory before delivery. A new calibration may be necessary if there are discrepancies in operation, which are manifested, for example by high vibrations.

⚠ *Please remove any load on the motor shaft before starting the calibration.*

The drive can be recalibrated by the WEB GUI. For this, it is necessary that a voltage of 24V is applied to the motor and that the motor shaft can rotate freely, no load attached.



Picture 7: Maintenance Mode

The button “calibration” is listed in the “Maintenance” tab. After activating this mode, the motor makes a few revolutions as a reference run and sets all the necessary parameters. This process takes about 5 minutes. The web GUI must not be used during this time, otherwise the calibration could be interrupted. The Motor is restarting automatically after a successful calibration.

4 Ethernet Communication

In addition to the HTTP web interface, the HDrive also has a TCP and a UDP channel. The UDP channel is only intended to send and not to receive data. Move and configuration commands are always sent by TCP. For the information telegrams from the motor, both the TCP and the UDP protocol can be configured in the web GUI.

For the binary interface it is recommended to use UDP to receive the data, because the UDP packets are not getting fragmentised during transmission over several Network devices.

4.1 Commands to the drive

A target position, target speed or target current can be specified for the motor with the positioning telegram. The internal path planner calculates a path based on the preconfigured accelerations and speeds.

4.1.1 Control-Ticket

Telegram to move to the position 100.0° with a speed of 200rpm, a maximum torque of 200mNm with acceleration and deceleration of 1000RPM / s:

<control pos="1000" speed="200" torque="200" mode="129" acc="1000" decc="1000" />

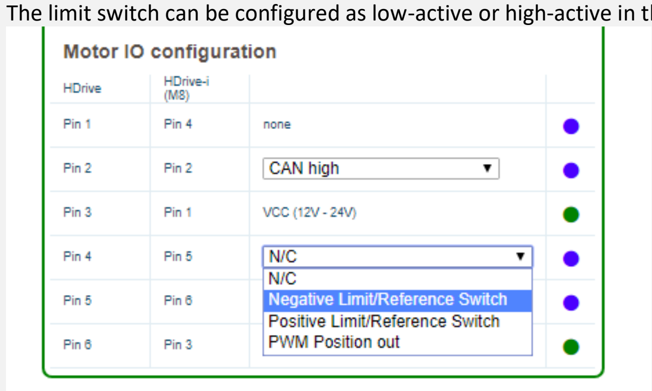
CAUTION: The order of the parameters must be observed. The system is intolerant of spaces in or in front of the values and attributes.

4.1.1.1 Parameter Description Control Ticket

XML Tag	*Range*	*Unit*	*Function*
Pos	±231	1/10 Degree	Target position
Speed	±2000	RPM	Max speed of the internal path planer
Torque	±600	mNm	Max torque
Mode	-1 bis 134	N/A	Operation mode
Acc	1 bis 50’000	RPM/s	Acceleration for internal path planer
Decc	1 bis 50’000	RPM/s	Deceleration for internal path planer

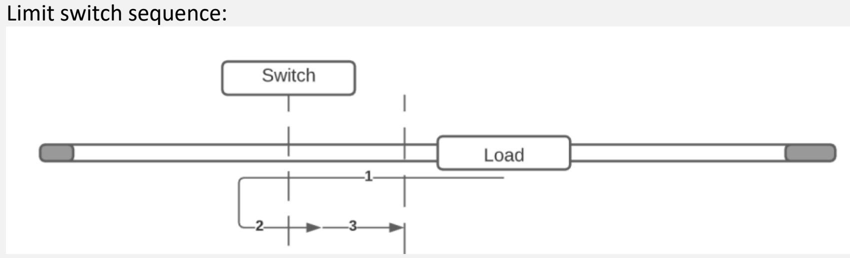
4.1.1.2 Operational modes (Control Ticket)

Mode ID	*Function*	*Description*
-1	Error	The engine has a fault. The error status can be read out in the web GUI or via object 3.4
0	Stop, Motor off	De-energizes the motor, the shaft can be rotated freely
8	Stepper mode	Stepper motor mode, can be used for extremely slow and even movements
9	Motor calibration	For the motor calibration, the shaft must be able to rotate without resistance. The motor then runs several revolutions. A calibration curve is calculated and permanently saved in the drive's EEPROM.
15	Limit Switch Left Advanced	The limit switch can be configured as low-active or high-active in the web GUI:



The speed and the maximum torque for the reference mode is taken from the control ticket with the parameter “speed" and "torque".

Limit switch sequence:



The motor moves to the left until the limit switch is activated for longer than 1 ms (internal debounce).
The motor is driving backwards with 20% of the commanded speed until the switch has been deactivated.
The motor keeps moving until the next index pulse of its internal absolute encoder.

After this procedure the behaviour is like followed:
The operating mode gets changed to mode 20 “Calibration Done”.
The drives position is around ± 1.5° after a successful homing. The accuracy of the position is much better, depending on the used limit switch. Switch to position control mode now to reach any position.

16	Limit Switch Right Advanced	As in Limit Switch Left, only reversed
17	Limit Switch Left Simple	As in Limit Switch Left only without moving to the encoder index pulse
18	Limit Switch Right Simple	As in Limit Switch Left only reversed and without moving to the encoder index pulse
20	Limit Switch Finished	Activated by states 15-18 as soon as the limit switch sequence is done
101	Set Zero	Resets the position to 0

128	Motor current control	The active current of the motor is regulated to a certain value. Thus, a constant torque acts on the motor
129	Motor Position control	The position controller with the activated path planner
130	Motor Speed control	Speed controller with internal path planner
132	Motor Speed control NPP1	Speed controller with deactivated path planner
133	Motor Position control NPP	Position controller with deactivated path planner
134	Motor Position control Stepper-Mode	In this mode the drive behaves like a closed loop stepper motor. In less dynamic applications, the mode can be used for a very smooth and constant run, e.g., when tracking a telescope. When the position is reached, the holding current is automatically reduced to 20% of the nominal current.

4.1.2 Config-Ticket-CAN

To use this configuration ticket, the digital I / O channels must first be set to "CAN-High" and "CAN-Low" in the web GUI. In addition, under the "Communication" tab, under "Communication Host to HDrive Ticket" the ticket "Can Ticket" must be selected.

The CAN master motor and all its slaves can be configured directly with the configuration ticket. The operating mode and the maximum torque of each individual drive can be commanded. As an example, all slave motors can be switched to homing-mode in this way.

[1] NPP = No Path Planer

Description	Range	Unit	Function	XML Tag
Master torque	+/- 600	mNm	Target torque of the CAN-Master motor	m
Master Mode			Mode of the master motor	mm
Slave 1 torque	+/- 600	mNm	Target torque of Slave1	Sl1t
Slave 2 torque	+/- 600	mNm	Target torque of Slave2	Sl2t
Slave 3 torque	+/- 600	mNm	Target torque of Slave3	Sl3t
Slave 4 torque	+/- 600	mNm	Target torque of Slave4	Sl4t
Slave 5 torque	+/- 600	mNm	Target torque of Slave5	Sl5t
Slave 6 torque	+/- 600	mNm	Target torque of Slave6	Sl6t
Slave 7 torque	+/- 600	mNm	Target torque of Slave7	Sl7t
Slave 8 torque	+/- 600	mNm	Target torque of Slave8	Sl8t
Slave 1 mode			Operational mode of Slaves 1	Sl1m
Slave 2 mode			Operational mode of Slaves 2	Sl2m
Slave 3 mode			Operational mode of Slaves 3	Sl3m
Slave 4 mode			Operational mode of Slaves 4	Sl4m
Slave 5 mode			Operational mode of Slaves 5	Sl5m
Slave 6 mode			Operational mode of Slaves 6	Sl6m
Slave 7 mode			Operational mode of Slaves 7	Sl7m
Slave 8 mode			Operational mode of Slaves 8	Sl8m
Slave 1 special function			Special function of Slaves 1	Sl1f
Slave 2 special function			Special function of Slaves 2	Sl2f
Slave 3 special function			Special function of Slaves 3	Sl3f
Slave 4 special function			Special function of Slaves 4	Sl4f
Slave 5 special function			Special function of Slaves 5	Sl5f
Slave 6 special function			Special function of Slaves 6	Sl6f
Slave 7 special function			Special function of Slaves 7	Sl7f
Slave 8 special function			Special function of Slaves 8	Sl8f

CAN Special functions:

XML Tag	Range	Unit	Function
Sf1-8	0 or 5	N/A	0 = no function 5 = Reset last error

The following example sets the operating mode of the master and its first 3 slaves to 133 (position controller without path planner). In addition, the maximum torque of the master is set to 200mNm and that of the slaves to 100mNm. The «special functions» are set to 0 (no function):

```
<canConf m="200" mm="133"
sl1t="100" sl2t="100" sl3t="100" sl4t="0" sl5t="0" sl6t="0" sl7t="0" sl8t="0"
sl1m="133" sl2m="133" sl3m="133" sl4m="0" sl5m="0" sl6m="0" sl7m="0" sl8m="0"
sl1fm="0" sl2f="0" sl3f="0" sl4f="0" sl5f="0" sl6f="0" sl7f="0" sl8f="0" />
```

CAUTION: The sequence of the tickets must not be changed. Every motor must be addressed also if not existing.

4.1.3 Control-Ticket-CAN

With the control ticket, target positions for all motors are sent. These target positions are then approached depending on the operating mode that was previously generated with a “ConfigTicketCan” command.

Description	*Range*	*Unit*	*Function*	*XML Tag*
Position	$\pm 2^{31}$	1/10 degree	Target position of Masters	Pos
Slave pos1	$\pm 2^{31}$	1/10 degree	Target position of Slave1	Sl1
Slave pos2	$\pm 2^{31}$	1/10 degree	Target position of Slave2	Sl2
Slave pos3	$\pm 2^{31}$	1/10 degree	Target position of Slave3	Sl3
Slave pos4	$\pm 2^{31}$	1/10 degree	Target position of Slave4	Sl4
Slave pos5	$\pm 2^{31}$	1/10 degree	Target position of Slave5	Sl5
Slave pos6	$\pm 2^{31}$	1/10 degree	Target position of Slave6	Sl6
Slave pos7	$\pm 2^{31}$	1/10 degree	Target position of Slave7	Sl7
Slave pos8	$\pm 2^{31}$	1/10 degree	Target position of Slave8	Sl8

<canPos pos="1000" sl1="200" sl2="20" sl3="500" sl4="100" sl5="100" sl6="100" sl7="100" sl8="100" />

As an example, a start sequence could look like this:

1. Set operational mode to position control without path planer (133). Set the torque of the master to 200 mNm and 100 mNm for the slaves:
   <canConf m="200" mm="133" sl1t="100" sl2t="100" sl3t="100" sl4t="100" sl5t="100" sl6t="0" sl7t="0" sl8t="0" sl1m="133" sl2m="133" sl3m="133" sl4m="133" sl5m="133" sl6m="0" sl7m="0" sl8m="0" />

2. Command individual target positions:
   <canPos pos="1000" sl1="200" sl2="20" sl3="500" sl4="1000" sl5="1000" sl6="1000" sl7="1000" sl8="1000" />

CAUTION: The sequence of the tickets must not be changed. Every motor must be addressed also if not existing.

4.1.4 Advanced-Config-CAN

With the “advanced config CAN” ticket, profile accelerations and speeds can be modified. These profile data are required in mode 129 «Position Control» or 130 «Speed Control». Or also the set the speed for the homing mode e.g., “Limit switch left advanced” (15).

Description	*Range*	*Unit*	*Function*	*XML-Tag*
Master_speed	Uint16	RPM	Profile speed of master motor	ms
Master_acc	Uint16	RPM/s	Profile acceleration of master motor	ma
Master_decc	Uint16	RPM/s	Profile deceleration of master motor	md
Slave1_speed	Uint16	RPM	Profile speed of Slave 1	S1s
Slave1_acc	Uint16	RPM/s	Profile acceleration of Slave 1	S1a
Slave1_decc	Uint16	RPM/s	Profile deceleration of Slave 1	S1d
Slave2_speed	Uint16	RPM	Profile speed of Slave 2	S2s
Slave2_acc	Uint16	RPM/s	Profile acceleration of Slave 2	S2a
Slave2_decc	Uint16	RPM/s	Profile deceleration of Slave 2	S2d
Slave3_speed	Uint16	RPM	Profile speed of Slave 3	S3s
Slave3_acc	Uint16	RPM/s	Profile acceleration of Slave 3	S3a
Slave3_decc	Uint16	RPM/s	Profile deceleration of Slave 3	S3d
Slave4_speed	Uint16	RPM	Profile speed of Slave 4	S4s
Slave4_acc	Uint16	RPM/s	Profile acceleration of Slave 4	S4a
Slave4_decc	Uint16	RPM/s	Profile deceleration of Slave 4	S4d
Slave5_speed	Uint16	RPM	Profile speed of Slave 5	S5s
Slave5_acc	Uint16	RPM/s	Profile acceleration of Slave 5	S5a
Slave5_decc	Uint16	RPM/s	Profile deceleration of Slave 5	S5d
Slave6_speed	Uint16	RPM	Profile speed of Slave 6	S6s
Slave6_acc	Uint16	RPM/s	Profile acceleration of Slave 6	S6a
Slave6_decc	Uint16	RPM/s	Profile deceleration of Slave 6	S6d
Slave7_speed	Uint16	RPM	Profile speed of Slave 7	S7s
Slave7_acc	Uint16	RPM/s	Profile acceleration of Slave 7	S7a
Slave7_decc	Uint16	RPM/s	Profile deceleration of Slave 7	S7d
Slave8_speed	Uint16	RPM	Profile speed of Slave 8	S8s
Slave8_acc	Uint16	RPM/s	Profile acceleration of Slave 8	S8a
Slave8_decc	Uint16	RPM/s	Profile deceleration of Slave 8	S8d

Example:

The followed command sets the master speed to 500 RPM, Master acceleration to 200 RPM/s and the Master deceleration to 2000 RPM/s. It also sets the velocity to 77 RPM and acceleration as well as deceleration to 1000 RPM/s for the first two slave motors.

```
<canPos
 ms="500" ma="200" md="2000"
 s1s="77" s1a="1000" s1d="1000"
 s2s="77" s2a="1000" s2d="1000"
 s3s="0" s3a="0" s3d="0"
 s4s="0" s4a="0" s4d="0"
 s5s="0" s5a="0" s5d="0"
 s6s="0" s6a="0" s6d="0"
 s7s="0" s7a="0" s7d="0"
 s8s="0" s8a="0" s8d="0" />
```

CAUTION: The sequence of the tickets must not be changed. Every motor must be addressed also if not existing.

4.1.5 System-Ticket

Various system states can be set with the "System" telegram.

XML Tag	*Range*	*Unit*	*Function*
mode	0 - 11	N/A	0 = Firmware Upgrade 1 = Bootloader Upgrade 2 = Position Reset (not permanent) 3 = Factory Reset 4 = Save data to EEPROM 5 = Reset last error 6 = Restart Motor 7 – 9 = reserved

In the following example, the position sensor of the drive is reset to 0 at the current position:

<system mode="2" a="1" b="2" c="3" />

After restarting the motor, the position of the rotor again corresponds to the absolute position added to the offset from the web GUI.

The XML-Tags a, b, and c needs to be 1, 2 and 3 but do not have a meaning yet.

4.1.6 Read and write status objects

XML Tag	*Range*	*Unit*	*Function*
m	0 - 8	N/A	Master Index
S	0 – 70	N/A	Slave Index
V	$\pm 2^{31}$	N/A	Value (in case of writing)

Reading of object 3.4 (last error)
<objRead m="3" s="4" />

Writing of object 3.4 (last error)
<objWrite m="3" s="4" v="0" />

4.2 Messages from the drive to the host computer

As soon as there is a TCP connection, the motor will send its position data via the TCP in the form of an XML-formatted string or in binary, depending on which connection and telegram type was selected in the web GUI.

The transported network data can be reduced in UDP mode.

4.2.1 HDriveTicket

<HDrive Position="300" Speed="0" Torque=*"0" Time="000216" />*

XML Tag	*Range*	*Unit*	*Function*
Position	10 digits	1/10 Degree	Current position of the drive
Speed	8 digits	RPM	Current speed of the drive
Torque	8 digits	mA	The active current, which is directly related to the torque (torque constant)
Time	10 digits	ms	The system time in milliseconds

This ticket is always 82 bytes long.

4.2.2 Binary-Ticket-Short

This telegram is transmitted in binary. This has many advantages in terms of computing time, which is less when interpreting the binary ticket.

Word No.	*Description*	*Bit count*
0	Time	int32
1	Position	int32
2	Speed	int32
3	Torque	int32

The ticket length is 16 Byte.

In binary mode, the transported network data as well as the computing power for interpreting the data can be reduced.

4.2.3 Binary-Ticket

Word No.	*Description*	*Bit count*
0	Time [us]	int32
1	Position	int32
2	Speed	int32
3	CurrentA [mA]	int32
4	CurrentB [mA]	int32
5	Calibration value [inc]	int32
6	Fid [mA]	int32
7	Fiq [mA]	int32
8	0	int32
9	Temperature [1/10°]	int32
10	Motor mode	int32
11	Motor voltage [mV]	int32
12	Demanded Speed	int32
13	Demanded Position	int32
14	Demanded Torque	int32
15	Demanded Acceleration	int32
16	Demanded Deceleration	int32
17	GPIO	int32
18	Actual Motor State	int32
19	Software Version	int32
20	Current (RMS) in mA	int32
21	Temp	int32
22	Temp	int32
23	Slave1 Position	int32
24	Slave 2 Position	int32
25	Slave 3 Position	int32
26	Slave 4 Position	int32
27	Slave 5 Position	int32
28	Slave 6 Position	int32
29	Slave 7 Position	int32
30	Slave 8 Position	int32

Total ticket length: 124 Byte.

Binary-CAN-Ticket

This telegram can only be generated by a CAN bus master and contains all position data of the «slave» motors attached to the CAN bus.

Word No.	*Description*	*Bit count*
0	Time	int32
1	Master Position	int32
2	Slave 1 Position	int32
3	Slave 2 Position	int32
4	Slave 3 Position	int32
5	Slave 4 Position	int32
6	Slave 5 Position	int32
7	Slave 6 Position	int32
8	Slave 7 Position	int32
9	Slave 8 Position	int32
10	Master Mode	int32
11	Slave 1 Mode	int32
12	Slave 2 Mode	int32
13	Slave 3 Mode	int32
14	Slave 4 Mode	int32
15	Slave 5 Mode	int32
16	Slave 6 Mode	int32
17	Slave 7 Mode	int32
18	Slave 8 Mode	int32
19	Master State	int32
20	Slave 1 State	int32
21	Slave 2 State	int32
22	Slave 3 State	int32
23	Slave 4 State	int32
24	Slave 5 State	int32
25	Slave 6 State	int32
26	Slave 7 State	int32
27	Slave 8 State	int32
28	Temp 1	Int32

The ticket length is 116 Byte.

Binary-CAN-Ticket Full

This telegram can only be generated by a CAN bus master and contains all position data of the «slave» motors attached to the CAN bus.

Word No.	*Description*	*Bit count*
0	System Time [us]	int32
1	Master Position	int32
2	Slave 1 Position	int32
3	Slave 2 Position	int32
4	Slave 3 Position	int32
5	Slave 4 Position	int32
6	Slave 5 Position	int32
7	Slave 6 Position	int32
8	Slave 7 Position	int32
9	Slave 8 Position	int32
10	Master Velocity	int32
11	Slave 1 Velocity	int32
12	Slave 2 Velocity	int32
13	Slave 3 Velocity	int32
14	Slave 4 Velocity	int32
15	Slave 5 Velocity	int32
16	Slave 6 Velocity	int32
17	Slave 7 Velocity	int32
18	Slave 8 Velocity	int32
19	Master Torque	int32
20	Slave 1 Torque	int32
21	Slave 2 Torque	int32
22	Slave 3 Torque	int32
23	Slave 4 Torque	int32
24	Slave 5 Torque	int32
25	Slave 6 Torque	int32
26	Slave 7 Torque	int32
27	Slave 8 Torque	int32
28	Master Mode	int32
29	Slave 1 Mode	int32
30	Slave 2 Mode	int32
31	Slave 3 Mode	int32
32	Slave 4 Mode	int32
33	Slave 5 Mode	int32
34	Slave 6 Mode	int32
35	Slave 7 Mode	int32
36	Slave 8 Mode	int32
37	Master State	Int32
38	Slave 1 State	Int32
39	Slave 2 State	Int32
40	Slave 3 State	Int32
41	Slave 4 State	Int32
42	Slave 5 State	Int32
43	Slave 6 State	Int32
44	Slave 7 State	Int32
45	Slave 8 State	Int32
46	Temp1	Int32
47	Temp2	Int32
48	Temp3	Int32

The ticket length is 196 Byte.

4.3 Error States

The last errors are saved directly in the motor and can be read out via the web GUI.

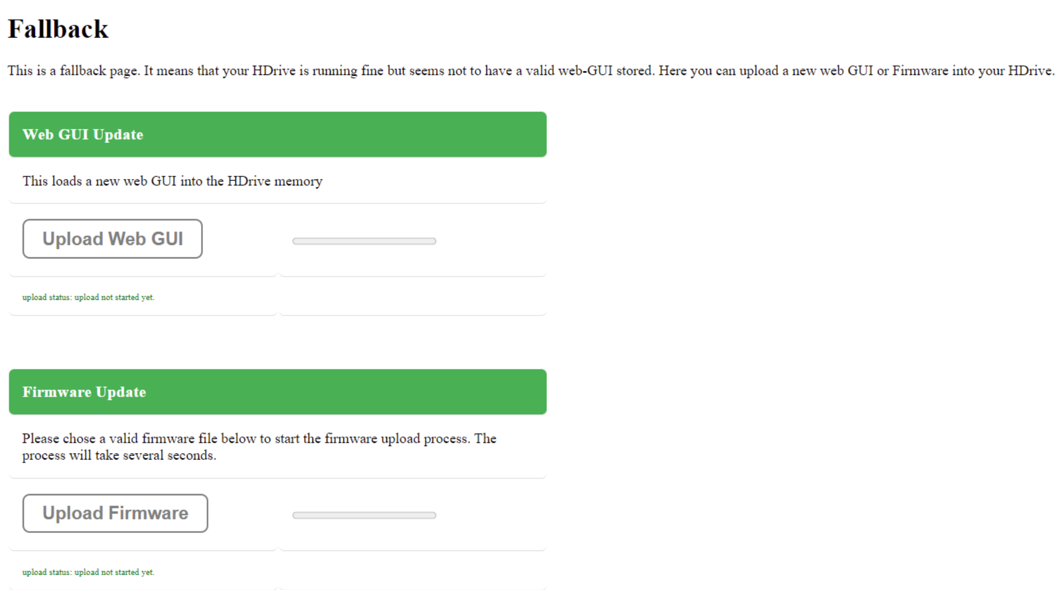
Error No.	*Error Message*	*Error description*
16	over Temperatur	Switches the motor off if the board temperature is more than 85 ° C.
17	under voltage	Switches the motor off and saves all states if the voltage drops below 10V.
18	over voltage	Switches the motor off when the operating voltage is over 30V. The generator operation can cause the voltage to rise sharply, particularly during braking. If necessary, an additional capacitor or a braking resistor must be added to the motor so that the excess power can be temporarily stored or converted into heat.
19	over speed	Only if the motor speed is higher than 2000 RPM, this can lead to an excessive induced voltage. The motor is braked automatically.
20	Positive Software Position Limit	The positive software position limit was recognized and stopped the motor
21	Negative Software Position Limit	The negative software position limit was recognized and stopped the motor
22	Negative Limit Switch Triggered	The negative limit switch was triggered on the referenced axis. Do not configure the digital I / O as a reference switch if this error is not desired.
23	Positive Limit Switch Triggered	The positive limit switch was triggered on the referenced axis. Do not configure the digital I / O as a reference switch if this error is not desired.
25	Limit Switch Timeout	The positive or negative limit switch was not found within 300s.
26	Position Sensor Error	Calibration error. This can be the result of a mechanical defect.
27	Power Stage Error	The power circuit has either an undervoltage or an overtemperature error. The undervoltage is usually the problem of a missing capacitor near this axis
28	Watchdog Timeout	The timeout Error if the watch dog function is switched on and is not reset by the host, it is immediately switching the motor off
50	Limit Switch Minimum Distance to End-Switch	The limit switch is too close to the index pulse of the internal motor encoder. Please move the motor mechanically to fit the minimum requirements (5° space between encoder and Motor internal zero position)

5 Firmware Upgrade

The firmware of the HDrive can be updated. This can be done by the web interface in the "Maintenance" menu. New Firmware files are announced at www.henschel-robotics.ch/Firmware

5.1 Website Upgrade

The website which contains the GUI can be updated separately. This happens in a similar way to the firmware upgrade via the «Maintenance» tab. If there is no GUI at the motor, only a white page appears when entering the IP address.
In any case you can use the address http://192.168.1.102/fallback.html to load a valid GUI from Henschel-Robotics GmbH or a custom one to the drive:



5.2 Factory Reset

If the IP address is unknown or if the motor was programmed incorrectly by mistake, the factory settings can be reloaded. To do this toggle the following **digital I/O Pin:**

HDrive17	*HDrive17-ETH-i*
Pin 1	Pin 4

Connect the pin three times to Vcc (24V) during start up. This must happen within 1 second. It can be done by hand with a little practice. If the reset is successful, the LED blinks fast in white, indicating the bootloader mode.
A new firmware can be loaded then with the web browser.

6 Object Dictionary

The drive's object directory can be read and written. When saving a configuration file in the WebGui, a list consisting of these objects will be written.

An object always contains two keys, the main key and the subkey. These two int32 numbers identify the object entirely.

6.1 Main keys

ID	Function
2	Control parameters such as P gain etc.
4	Communication parameters such as IP address
5	System parameters such as gear reduction, limit switches, etc.
6	Motor values such as inductance, DC resistance, max. Torque etc.

6.2 Sub keys

Each subkey belongs to a main key.

6.2.1 Subkey for master key # 2 - control parameters

ID	Function
0	reserved
1	Current control P
2	Current control I
3	Current control Kb
4	Speed control P
5	Speed control I
6	Speed control Fg
7	Speed control Kb
8	Speed control
10	Position control speed limit
11	Position control Slew rate limit
12	Position pre filter
13	Position FF Filter
14	Bandwidth current controller
15	Bandwidth speed controller
16	Bandwidth position controller
17	DF current control
18	DF speed control
19	DF position control
20	Current filter AB
21	Current filter QD
22	reserved
23	Position filter

6.2.2 Subkeys for Master Key No. 4 - Communication Parameters

ID	*Function*
0	IP-SEGMENT 1
1	IP-SEGMENT 2
2	IP-SEGMENT 3
3	IP-SEGMENT 4
4	HOSTADRESS FOR UDP SEGMENT 1
5	HOSTADRESS FOR UDP SEGMENT 2
6	HOSTADRESS FOR UDP SEGMENT 3
7	HOSTADRESS FOR UDP SEGMENT 4
8	SUBNETMASK SEGMENT 1
9	SUBNETMASK SEGMENT 2
10	SUBNETMASK SEGMENT 3
11	SUBNETMASK SEGMENT 4
12	GATEWAY SEGMENT 1
13	GATEWAY SEGMENT 2
14	GATEWAY SEGMENT 3
15	GATEWAY SEGMENT 4
16	TCP-PORT HOST
17	UDP-PORT HOST
18	COMMUNICATON PREESCALER FOR AUTO SEND TCP/UDP TICKETS
19	UDP-KOMMUNIKATION ENABLE/DISABLE
20	CAN ID
21	RX TICKET (HOST -> HDRIVE)
22	TX TICKET (HDRIVE -> HOST)
23	CAN MASTER
24	CLIENT SLAVE ID 1
25	CLIENT SLAVE ID 2
26	CLIENT SLAVE ID 3
27	CLIENT SLAVE ID 4
28	CLIENT SLAVE ID 5
29	CLIENT SLAVE ID 6
30	CLIENT SLAVE ID 7
31	CLIENT SLAVE ID 8
32	CAN BAUDRATE
33	RESERVED
34	AUTOSEND TCP/UDP TICKETS
35	WATCHDOG ENABLE/DISABLE
36	WATCHDOG TIMEOUT
37	DHCP ENABLE/DISABLE

6.2.3 Subkeys for Master Key No. 5 - System Parameters

ID	*Function*
0	Virtual gearbox numerator
1	Virtual gearbox denumerator
2	Digital I/O 2
3	Digital I/O 4
4	Digital I/O 5
5	Digital I/O 6
6	Digital I/O 7
7	Digital I/O 8
8	Step width (in Step/Dir interface)
9	Step width 2 (in Step/Dir interface)
10	reserved
11	Torque for Step/Dir interface
12	Speed for Step/Dir interface
13	Software Limit Left
14	Software Limit Right
15	Software Limit (1= on, 0=off)
16	Path planer activated
17	Position offset in 1/10°
18	Limit switch high (1) or low (0) active
19	Blink if WebGUI is active (1/0)

6.2.4 Subkey for Master Key # 6 - Engine Values

ID	*Function*
0	DC coil resistance in mOhm
1	Motor inductivity in uH
2	Maximal current in mA
3	Torque constant in mNm/A
4	Rotor and load inertia in g*cm^2
5	Motor damping factor
6	Motor resonance frequency

7 Technical data

7.1 Default values (Factory reset)

DHCP: Off
IP Address: 192.168.1.102
Subnet mask: 255.255.255.0
TCP-Port: 1000
UDP-Port: 1001

7.2 Typical electrical values

7.2.1 Supply

Name	*Function*	*Unit*	*Value*
VCC	Supply voltage	V	12-24
Max current (2s)		A	3
Continuous current		A	2

The HDrive has an integrated reverse polarity protection and can withstand reverse polarity for a short time without damage. If the supply voltage is connected the wrong way, the drive shows no reaction.

7.2.2 Electrical absolute maximum values

Name	*Function*	*Unit*	*Value*
VCC	Supply voltage	V	48 (HW rev 6a)
Max current	Supply current	A	3
Digital In	Digital-in	V	**0 – 24**

7.2.3 Motor characteristics

Name	*Minimum*	*Typical*	*Maximum*	*Unit*
No-load speed	-800		800	U/min
Holding torque	-0.6	-	0.6	Nm
single-turn absolute Encoder		14 Bit		
Absolut accuracy		+/- 0.1 (1 Sigma)		Grad
Torque constant		200		mNm/A

7.2.4 Timing characteristics

Name	*Minimum*	*Typical*	*Maximum*	*Unit*
Current control frequency		19		kHz
Position/Speed control frequency		6		kHz
Start time power up		2		s
Position data from drive (TCP) max.		-	0.5	kHz
Position data from drive (UDP) max.			2	kHz

7.2.5 Environmental conditions

Kind	*function*
Ambient temperature	–10 °C to 40 °C, not condensing

7.2.6 LED, light coding

LED	*Function*
green blinking	Motor ready
green permanent	Power stage switched on
red blinking	Error
red and green fast blinking	Firmware Reset
blue blinking	Drive is accessed by a web browser
blue permanent	Power stage switched on and drive accessed by a browser
yellow blinking	Motor ready but no ethernet connectivity

7.3 Digital I/O

The inputs and outputs on the HDrive can be configured individually. Each GPIO can be defined as an input or output. It is possible to enable a 5 V logic by the Web-GUI, then the inputs can be used to interface with a 5V logic directly.

7.3.1 24V mode

Digital-In Voltage 0…24 VDC
Logic 0 < 4.8 V
Logic 1 > 14.4 V

7.3.2 5V mode

Digital-In Voltage 0…5 VDC
Logic 0 < 0.8 V
Logic 1 > 2.0 V

7.3.3 Digital input

You can use the Inputs as low or high active. Meaning it is possible to short the input to 0 V or to +24V to trigger.

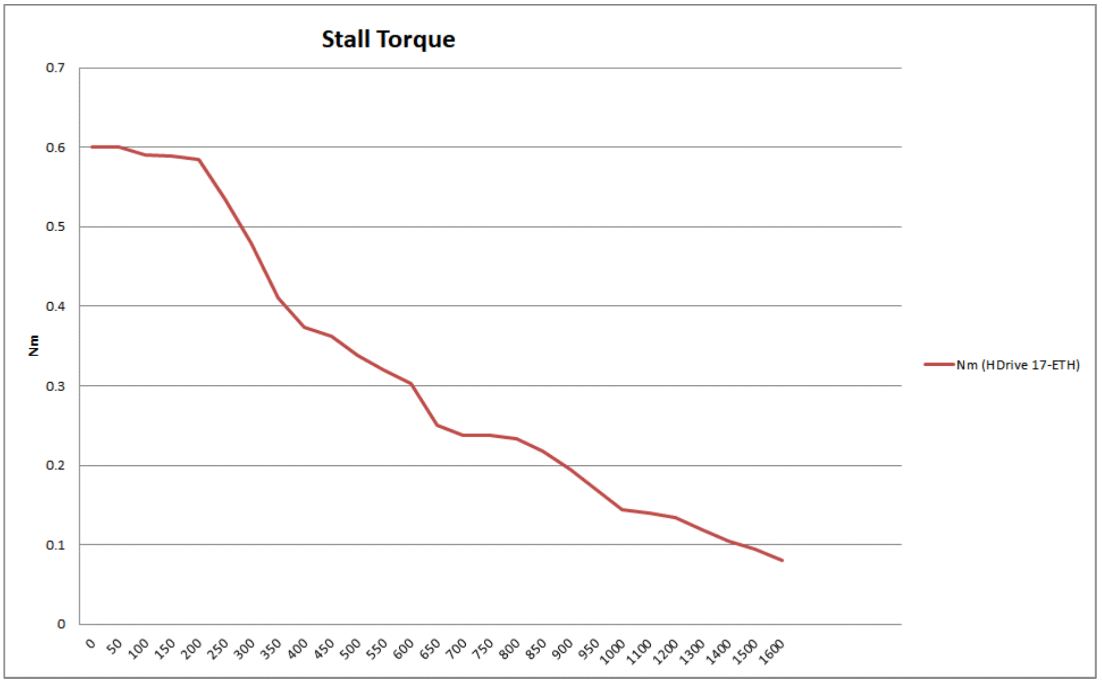
7.4 Digital output

Each Digital I/O can be configured as a digital output. The output is open drain and can drive a load up to 100 mA.
To connect a load, use the 24V to feed your load and connect the GND of the load to the I/O pin of the drive. The open collector transistor then is connecting the GND to the Power supply GND if the I/O is activated.

All logic inputs on the motor are 24V compatible.

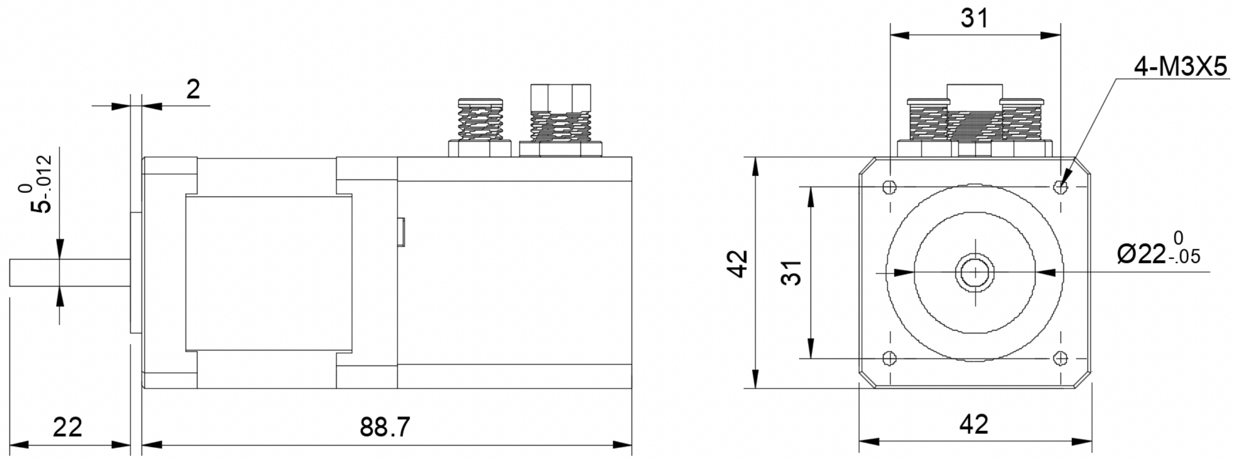
7.5 Torque curve, weight, and dimensions

HDrive17-ETH-I Speed / torque behaviour



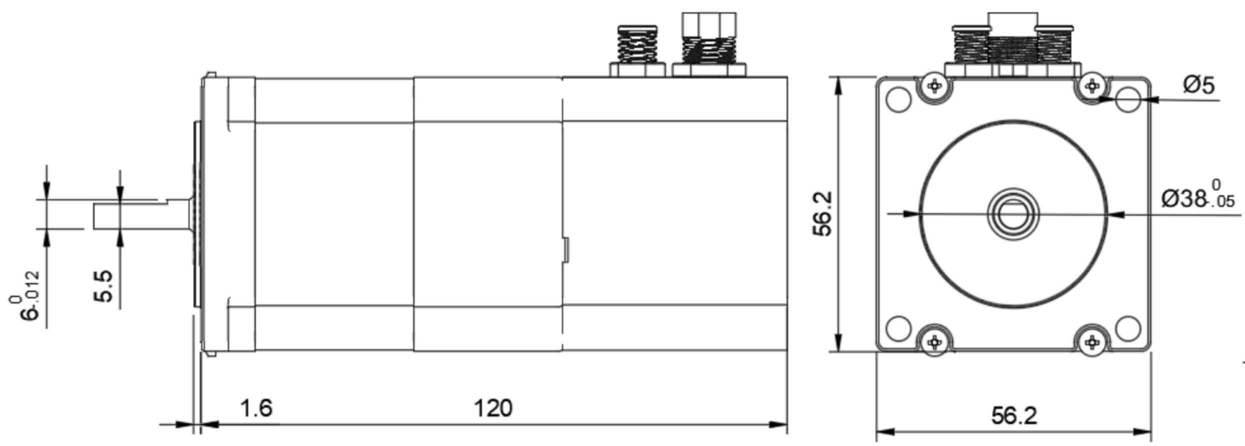
Picture 8: Speed / torque behaviour

7.5.1 Dimensions HDrive17-ETH-i:



Dimensions in mm, weight: 470 grams

7.5.2 Dimensions HDrive23-ETH-i:



Weight: 1250 gram