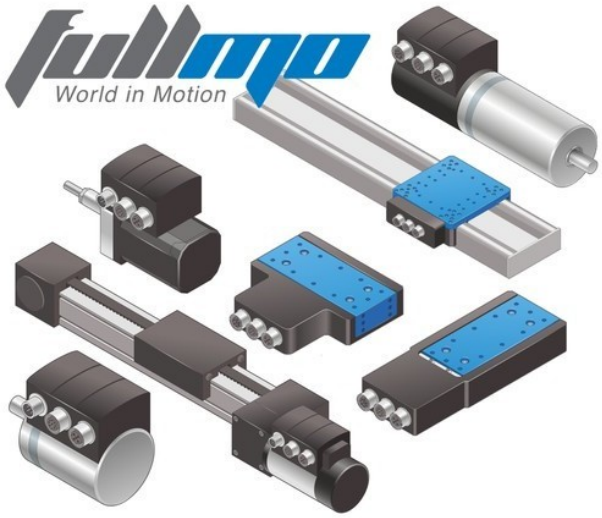


MovingCap Ethernet drives Quick guide

www.fullmo.de

Table of contents

1 Area of application

This manual provides a technical overview of software functions and operating instructions for MovingCap compact drives with integrated controller and Ethernet TCP/IP (ETH) interface for 12Vdc, 24Vdc and 48Vdc supply voltages. e.g.

- MovingCap turnTRACK Ethernet TCP/IP (MC349 ETH, MC632/634/636 ETH, MCN23 ETH)
- MovingCap flatTRACK Ethernet TCP/IP (MCFLAT ETH, flatTRACK 100-650, FATtrack 200)
- MovingCap shortTRACK Ethernet TCP/IP (MCSHORT ETH, shortTRACK 046)
- MovingCap shortTRACK SM Ethernet TCP/IP (shortTRACK 45S100 - 115S240)

WARNING! Please observe the Fullmo installation instructions and Fullmo declaration of incorporation belonging to your product! For all operating procedures described in this document, the requirements for proper operation in accordance with the Fullmo installation instructions and Fullmo declaration of incorporation must be met in advance. Operation without observing these requirements is prohibited.

NOTE: Please also refer to the specific documentation for the respective drive variant, particularly with regard to performance data, connection assignments, parameterizations, special functions and options such as brakes.

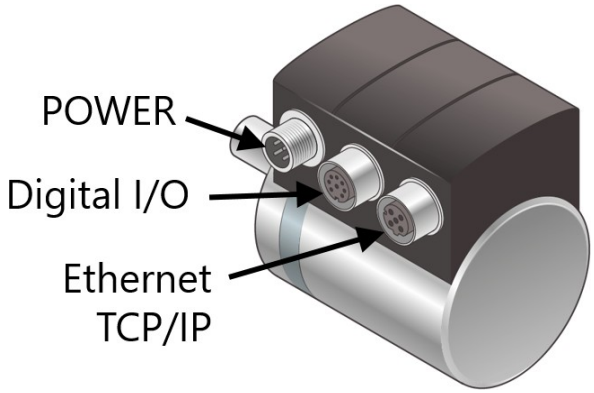
Software version drives: v50.00.07.xx / v50.01.07.xx / v53.02.00.xx / v54.01.07.xx - Document version 2024-11-21 Draft

2 Connections

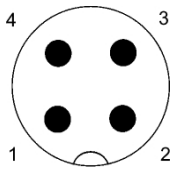
MovingCap Ethernet TCP/IP (ETH) drives have three M12 connections with different coding and pin assignments as standard. Below you will find an overview of the pin assignments and cables to be used.

2.1 MC349 ETH Pin assignment

Overview of pin assignment for MovingCap turnTRACK 349 Ethernet drive variants



POWER

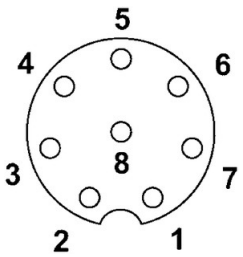


MC349 ETH (MovingCap turnTRACK 349 Ethernet)
POWER
M12 plug, 4-pin, coding: A

Pin	Designation	Description	Color *1)
1	U_PWR	Power supply 24-48 Vdc	brown
2	GND	GND Logic + Power	white
3	U_LOGIK	Logic supply 24 Vdc	blue
4	(optional HW_EN)	reserved	black

*1) Colors of the individual wires for connection cable Phoenix Contact SAC-4P-M12FS sensor/actuator cable

DIGITAL I/O

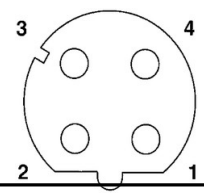


MC349 ETH (MovingCap turnTRACK 349 Ethernet)
Digital I/O
M12 socket, 8-pin, coding: A

Pin	Designation	Description	Color *1)
1	COM U_LOGIK	Output, provides logic supply U_LOGIK	white
2	IN1	Digital input 24 Vdc	brown
3	IN2	Digital input 24 Vdc	green
4	IN3	Digital input 24 Vdc	yellow
5	IN4	Digital input 24 Vdc	gray
6	OUT1	Digital output 24 Vdc	pink
7	OUT2	Digital output 24 Vdc	blue
8	GND	GND Logic	red

*1) Colors of the individual wires for connection cable Phoenix Contact SAC-8P-M12MS sensor/actuator cable M12 8-pin

ETHERNET TCP/IP

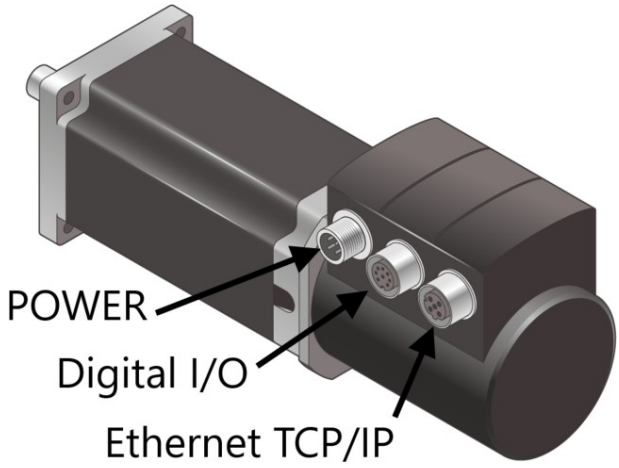


MC349 ETH (MovingCap turnTRACK 349 Ethernet)
Ethernet TCP/IP
M12 socket, 4-pin, coding: D Data

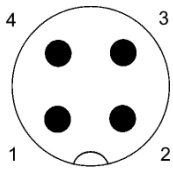
Connection for e.g.
Phoenix Contact VS-MSD-IP20-93E Network cable to RJ45

2.2 MCN23 ETH Pin assignment

Overview of pin assignment for MovingCap turnTRACK N23 Stepper Ethernet drive variants



POWER

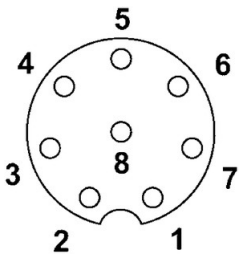


MCN23 ETH (MovingCap turnTRACK N23 Stepper Ethernet)
POWER
M12 plug, 4-pin, coding: A

Pin	Designation	Description	Color *1)
1	U_PWR	Power supply 24-48 Vdc	brown
2	GND	GND Logic + Power	white
3	U_LOGIK	Logic supply 24 Vdc	blue
4	(optional HW_EN)	reserved	black

*1) Colors of the individual wires for connection cable Phoenix Contact SAC-4P-M12FS sensor/actuator cable

DIGITAL I/O

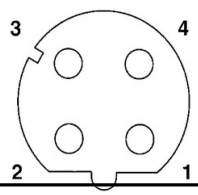


MCN23 ETH (MovingCap turnTRACK N23 Stepper Ethernet)
Digital I/O
M12 socket, 8-pin, coding: A

Pin	Designation	Description	Color *1)
1	COM U_LOGIK	Output, provides logic supply U_LOGIK	white
2	IO1	Digital input/output 24 Vdc	brown
3	IO2	Digital input/output 24 Vdc	green
4	IO3	Digital input/output 24 Vdc	yellow
5	IO4	Digital input/output 24 Vdc	gray
6	IN5	Digital input 24 Vdc	pink
7	IN6	Digital input 24 Vdc	blue
8	GND	GND Logic	red

*1) Colors of the individual wires for connection cable Phoenix Contact SAC-8P-M12MS sensor/actuator cable M12 8-pin

ETHERNET TCP/IP

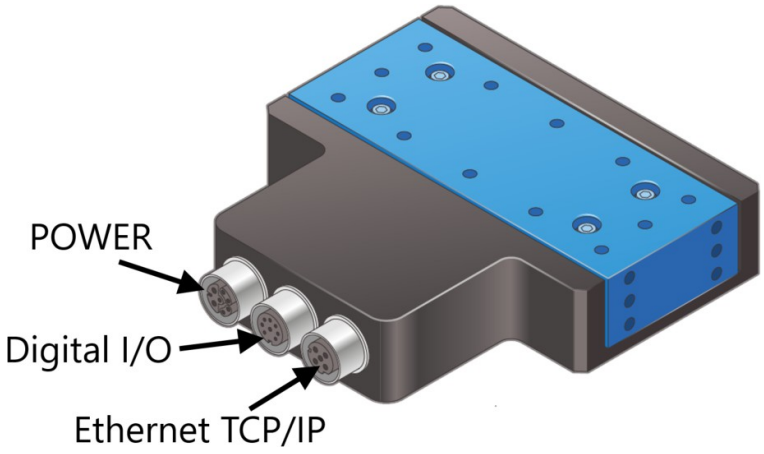


MCN23 ETH (MovingCap turnTRACK N23 Stepper Ethernet)
Ethernet TCP/IP
M12 socket, 4-pin, coding: D Data

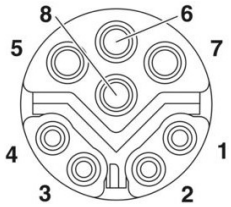
Connection for e.g.
Phoenix Contact VS-MSD-IP20-93E Network cable to RJ45

2.3 MCFLAT / MCSHORT ETH Pin assignment

Overview of pin assignment for MovingCap flatTRACK Ethernet drive variants (flatTRACK 100- 650, FATtrack 200, shortTRACK 046)



HYBRID POWER + DIGITAL I/O

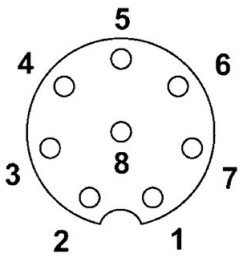


MCFLAT ETH (MovingCap flatTRACK Ethernet)
MCSHORT ETH (MovingCap shortTRACK Ethernet)
POWER + I/O
M12 hybrid socket, 8-pin, coding: Y

Pin	Designation	Description	Color *1)
1	IN10 (opt. HW_EN)	Digital input 24 Vdc	white/orange
2	IO1	Digital input/output 24 Vdc	orange
3	IN8	Digital input 24 Vdc	white/green
4	IO2	Digital input/output 24 Vdc	green
5	U_PWR	Power supply 24-48 Vdc	blue
6	GND	GND Logic + Power	white
7	U_LOGIK	Logic supply 24 Vdc	brown
8	IN7	Digital input 24 Vdc	black

*1) Colors of the individual wires for Phoenix Contact NBC-M12MSY hybrid cable M12 8-pin, Y-coding

DIGITAL I/O

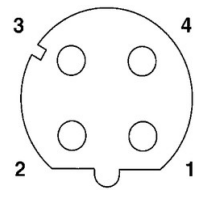


MCFLAT ETH (MovingCap flatTRACK Ethernet)
MCSHORT ETH (MovingCap shortTRACK Ethernet)
Digital I/O
M12 socket, 8-pin, coding: A

Pin	Designation	Description	Color *2)
1	COM U_LOGIK	Output, provides logic supply U_LOGIK	white
2	IO3	Digital input/output 24 Vdc	brown
3	IO4	Digital input/output 24 Vdc	green
4	IO1 (optional)	Digital input/output 24 Vdc	yellow
5	IO2 (optional)	Digital input/output 24 Vdc	gray
6	IN5	Digital input 24 Vdc	pink
7	IN6	Digital input 24 Vdc	blue
8	GND	GND Logic	red

*2) Colors of the individual wires for connection cable Phoenix Contact SAC-8P-M12MS sensor/actuator cable M12 8-pin

ETHERNET TCP/IP

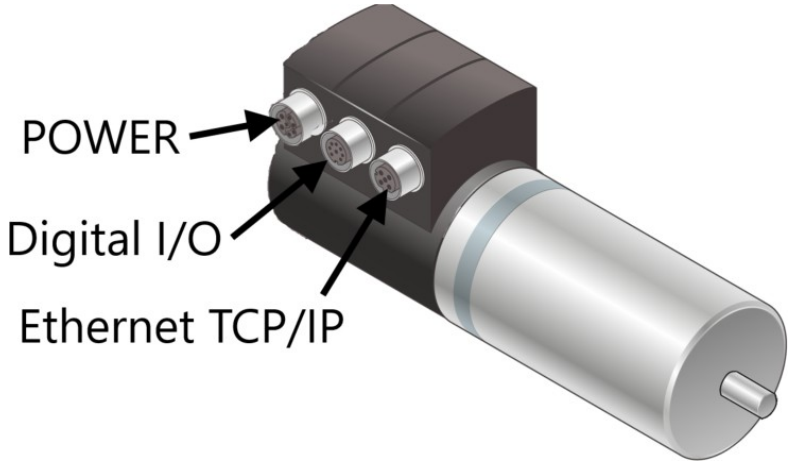


MCFLAT ETH (MovingCap flatTRACK Ethernet)
MCSHORT ETH (MovingCap shortTRACK Ethernet)
Ethernet TCP/IP
M12 socket, 4-pin, coding: D Data

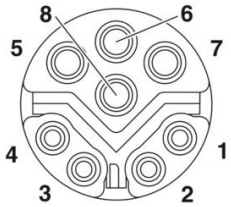
Connection for e.g.
Phoenix Contact VS-MSD-IP20-93E Network cable to RJ45

2.4 MC6xx ETH Pin assignment

Overview of pin assignment for MovingCap turnTRACK 6xx Ethernet drive variants (MC632 ETH / MC634 ETH / MC636 ETH)



HYBRID POWER + DIGITAL I/O

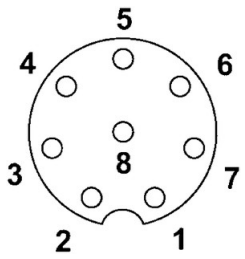


MC632 ETH / MC634 ETH / MC636 ETH (MovingCap turnTRACK
6xx Ethernet)
POWER + I/O
M12 hybrid socket, 8-pin, coding: Y

Pin	Designation	Description	Color *1)
1	IN10 (opt. HW_EN)	Digital input 24 Vdc	white/orange
2	IO1	Digital input/output 24 Vdc	orange
3	IN8	Digital input 24 Vdc	white/green
4	IO2	Digital input/output 24 Vdc	green
5	U_PWR	Power supply 24-48 Vdc	blue
6	GND	GND Logic + Power	white
7	U_LOGIK	Logic supply 24 Vdc	brown
8	IN7	Digital input 24 Vdc	black

*1) Colors of the individual wires for Phoenix Contact NBC-M12MSY hybrid cable M12 8-pin, Y-coding

DIGITAL I/O

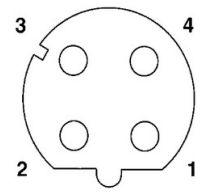


MC632 ETH / MC634 ETH / MC636 ETH (MovingCap turnTRACK 6xx Ethernet)
Digital I/O
M12 socket, 8-pin, coding: A

Pin	Designation	Description	Color *2)
1	COM U_LOGIK	Output, provides logic supply U_LOGIK	white
2	IO3	Digital input/output 24 Vdc	brown
3	IO4	Digital input/output 24 Vdc	green
4	IO1 (optional)	Digital input/output 24 Vdc	yellow
5	IO2 (optional)	Digital input/output 24 Vdc	gray
6	IN5	Digital input 24 Vdc	pink
7	IN6	Digital input 24 Vdc	blue
8	GND	GND Logic	red

*2) Colors of the individual wires for connection cable Phoenix Contact SAC-8P-M12MS sensor/actuator cable M12 8-pin

ETHERNET TCP/IP



MC632 ETH / MC634 ETH / MC636 ETH (MovingCap turnTRACK
6xx Ethernet)
Ethernet TCP/IP
M12 socket, 4-pin, coding: D Data

Connection for e.g.
Phoenix Contact VS-MSD-IP20-93E Network cable to RJ45

3 Display

MovingCap Ethernet TCP/IP (ETH) drives have two LED indicators as standard

LED Color	Designation	Description
yellow	LINK/ACT	Connection status and activity of the network connection
green	STATUS	MovingCap IP address and status via Morse code

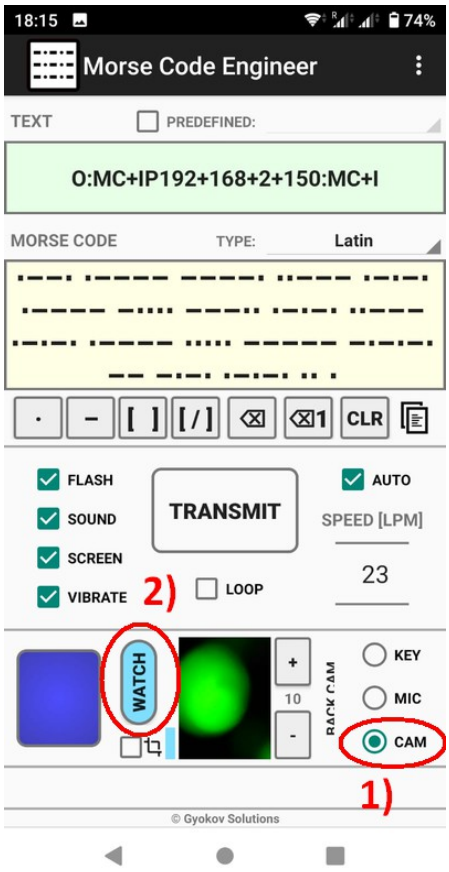
DECODE MORSE CODE / DETERMINE IP ADDRESS

Among other things, the green status LED continuously transmits the currently set IP address and can be evaluated as follows:

- Install a Morse code decoder app such as "Morse Code Encoder & Decoder" (Noldy Labs) or "Morse Code Engineer" (GyokovSolutions) on a smartphone
- Start the Morse app and use the camera to film the green LED as close as possible and with as little interfering light as possible. Hold the smartphone as still as possible.
- Wait at least 2 minutes or until you have decoded two consecutive identical texts to ensure that the decoding was correct.

The MovingCap Morse message with IP default setting is e.g:
MC+IP192+168+2+150:

(As a Morse code, the plus sign is significantly shorter than a decimal point and is therefore used as a separator).



NOTE: If a Morse app is not possible for you, you can also film the green LED for at least 2 minutes and provide us with the video for decoding. You can find our contact information at www.fullmo.de

4 Brief overview of operation

WARNING! Please observe the Fullmo installation instructions and Fullmo declaration of incorporation belonging to your product! For all operating procedures described in this document, the requirements for proper operation in accordance with the Fullmo installation instructions and Fullmo declaration of incorporation must be met in advance. Operation without observing these requirements is prohibited.

4.1 Network connection

The connection to a PC / network switch with standard RJ45 network socket is made using a suitable data cable for M12 D-coding, e.g.Phoenix Contact VS-MSD-IP20-93E network cable to RJ45.

STANDARD SETTINGS MOVINGCAP TCP/IP

IP address: **192.168.2.150**
Subnet mask: **255.255.255.0**
Gateway: -
DHCP: - not available - DNS: -
not available -

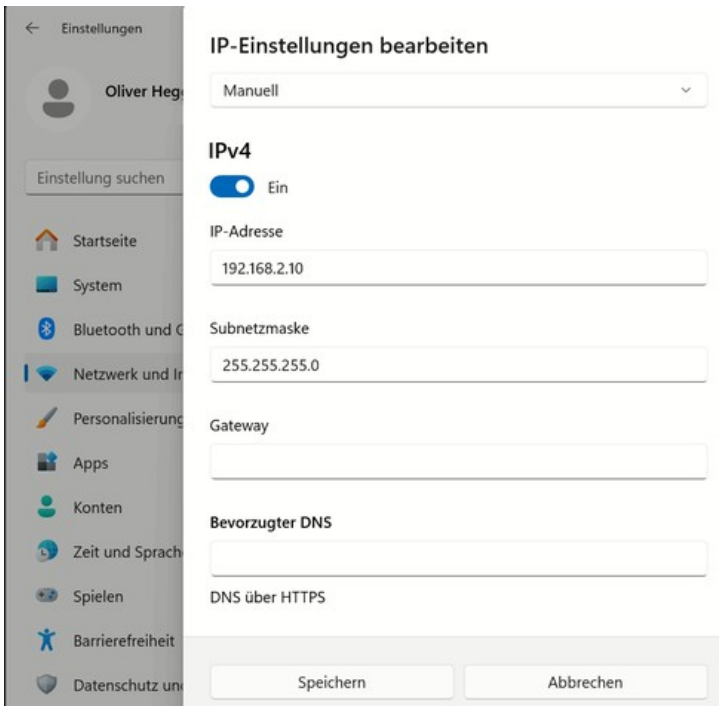
NOTE: If you do not know the IP address set in the MovingCap, you can determine the address as described in the Display 14 using Morse code.

PREPARING THE NETWORK CONNECTION

- Connect a USB-to-Ethernet adapter to your PC or use the built-in RJ45 Ethernet port.

- Change the connection settings for this connection to an IP4 address in the same subnet (but different end number) as the MovingCap setting, e.g:
  IP address: **192.168.2.10**
  Subnet mask: **255.255.255.0**
  Gateway: -
  DNS: -

  In Windows 11 this can look like this:



4.2 MovingCap WEB - operation via web browser

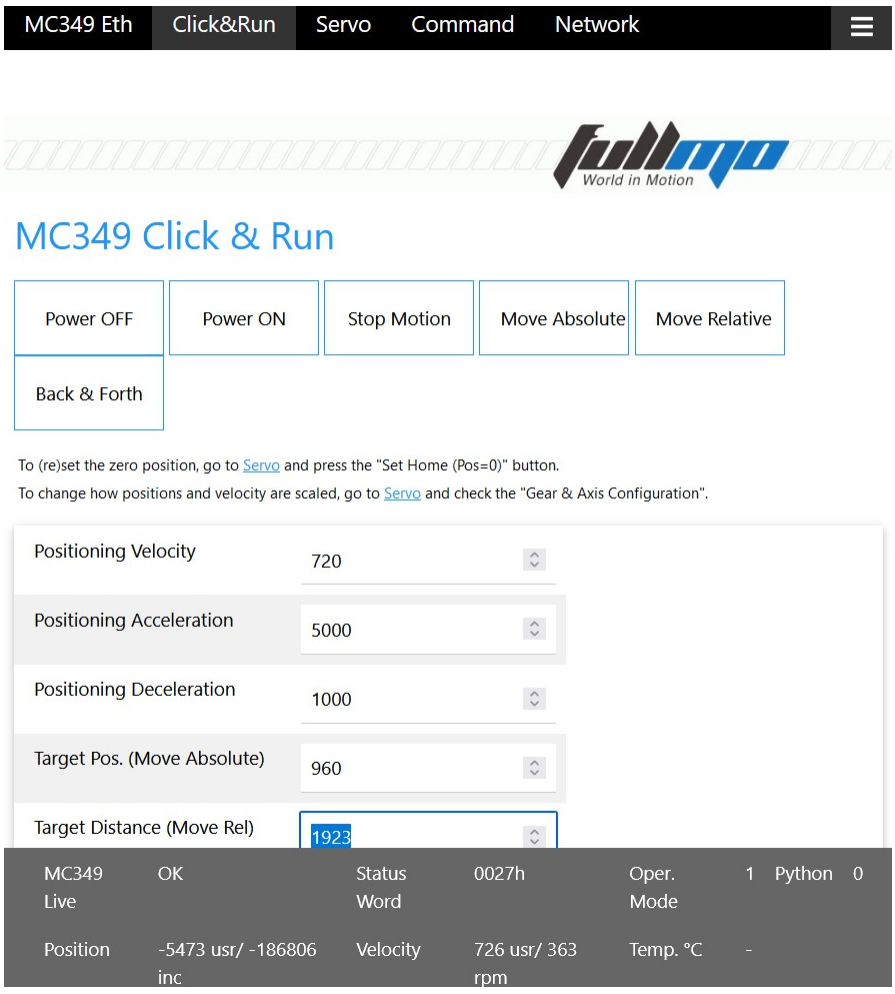
– Use a web browser to open the MovingCap website for configuration and operation, e.g. by entering 192.168.2.150 in the address bar

4.2.1 Simple function test

- Open the web interface by entering 192.168.2.150 (or your set IP address) in the address bar

- Select the **Back & Forth** function in the **Click & Run** menu.

  - The drive starts an oscillating operation over the distance defined via **Target Distance (Move Rel).**



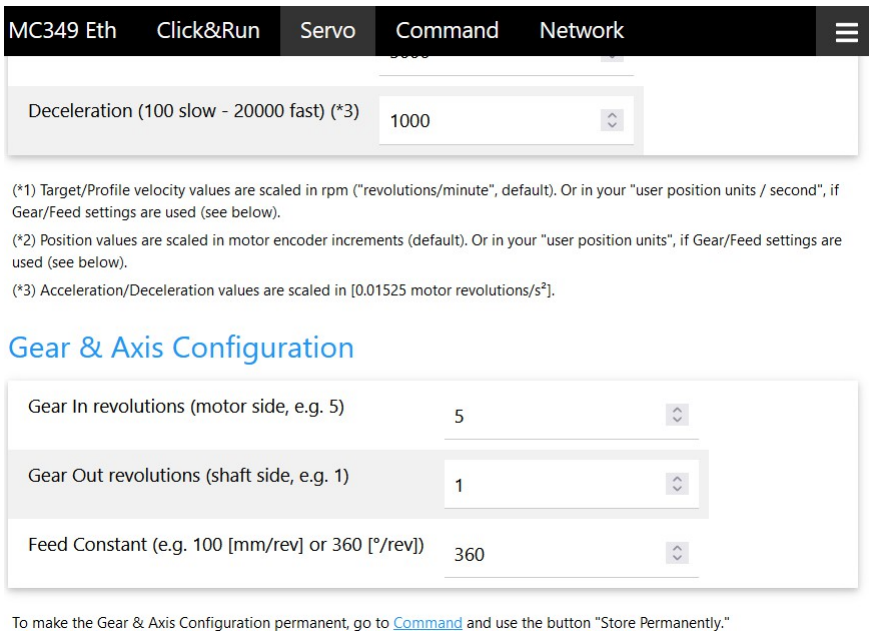
4.2.2 Set position units / scaling / zero position

- Select the **Servo** menu and set the desired scaling in **Gear & Axis Configuration** as follows:

  Gear ratio:
  Gear In - revolutions on the motor shaft, e.g. **5** . (Or **1** if there is no gearbox)
  Gear Out - revolutions at the gearbox output shaft, e.g. **1**

  Set scaling/resolution:
  Feed - If the motor makes one revolution at the gearbox output - by how much does the position counter increase?

  Example values for "Feed":
  360 - if you want to mount a turntable at the gearbox output and specify positions in angular degrees.
  100000 - if you are using a toothed belt axis with drive roller circumference d=100 mm (e.g. max GmbH MZK 040) and want to work in micrometer coordinates. A RefGo 23 move command **G100000** then corresponds to "Move to absolute position 100,000 micrometers / 100mm", a move command G2500 corresponds to "Move to absolute position 2.5mm"

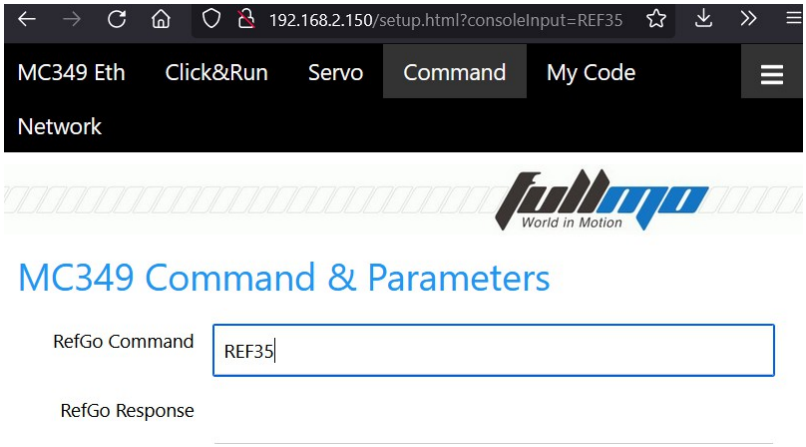


- Actuators with absolute position encoder (e.g. MC349_..._**AE**): Reset the zero position if necessary:

  Select the Command menu and enter the following command for **RefGo Command:**

  REF35

("Referencing with the CiA 402 homing method 35 - resetting the zero position")

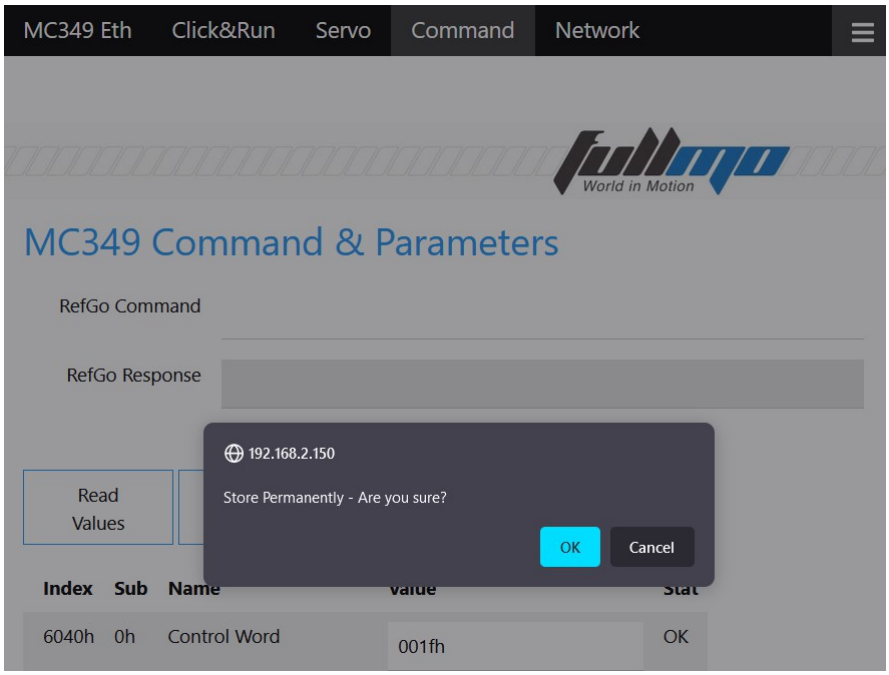


4.2.3 Saving settings permanently

To ensure that parameter settings are retained even after the power supply is switched off/on again, they must be stored in the non-volatile parameter memory using a separate command.

Proceed as follows:

- Select the **Store Permanently** function in the **Parameters** menu
- Confirm the security question with **OK**



4.3 MovingCap SLCAN - CANopen DS402 with kickdrive

Communication via: **TCP Port 15001 / CANopen Node-ID: 50**

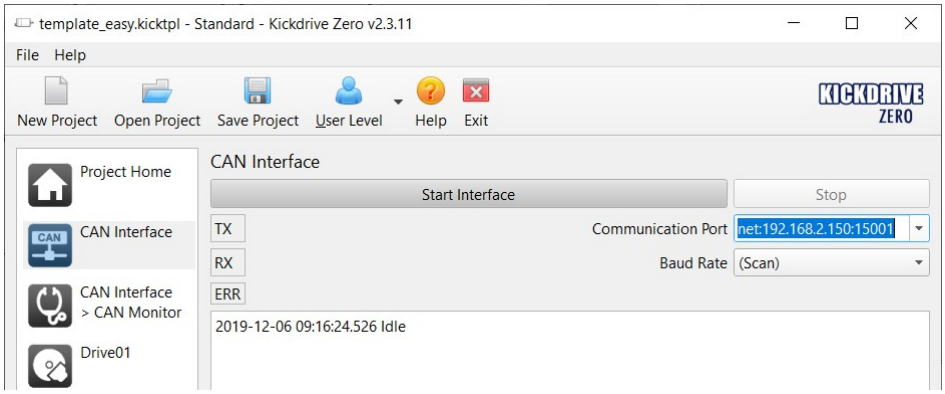
Communication takes place in a format similar to CAN-to-TCP/IP gateways, e.g. VSCOM NetCAN. See also "SLCAN ASCII protocol" (www.vscom.de/download/multiio/others/info/VSCAN_Manual.pdf , "ASCII Command Set") or Python module "CAN over Serial / SLCAN" (https://python-can.readthedocs.io/en/3.3.2/interfaces/slcan.html#).

Via this TCP connection, MovingCap can be parameterized and operated as a CANopen node with **node ID 50** like a CiA 402 CANopen/EtherCAT fieldbus drive. This operating mode is supported by fullmo Kickdrive and Kickdrive Zero.

BASIC COMMISSIONING MOVINGCAP ETH VIA WINDOWS SOFTWARE **FULLMO KICKDRIVE** OR **KICKDRIVE ZERO**:

- Proceed as in document **fullmoBasicSetupMovingCap349_en.pdf** (from our movingcap.de service portal, subdirectory **turnTRACK_MC349_CANopen**), with the following differences:
  - The data connection is not via CAN bus, but via an Ethernet network connection 15.
  - Before starting the drive in the open Kickdrive project, select the **CAN interface** module the setting
    Communication port = net:192.168.2.150:15001

(or the IP address you have selected for the drive).
Baud rate = 125K
(or any, as not relevant for TCP operation)



4.4 MovingCap REFGO

Communication via: **TCP port 10001**

RefGo is a simple text-based protocol (ASCII protocol) for parameterizing and operating the MovingCap.

Command format (using the example for TP / "Tell Position"):

```
TP<CR>
```

Response format:

```
TP<CR><LF>
106019<CR><LF>
>
```

With
<CR>= carriage return / ASCII code decimal 13 / hexadecimal 0D / no visible character!
<LF>= line feed / ASCII code decimal 10 / hexadecimal 0A / no visible character!

The prompt character > ends the MovingCap response and indicates that MovingCap is ready for the next command.

4.4.1 Command list

Text command	Description	Remark
G	Go Direct position	*1)
GW	Go Way	*1) *6)

PO	Set target position	*1)
WA	Set Way	*1)
SM	Stop Motion	*1)
PWC	Power Continue	*1)
REF	Referencing	*1) *2)
PQ	Power Quit	*1)
TPSR	Tell Process Status Register	*1) *4)
TS	Tell Status	*1) *5)
TP	Tell Actual Position	*1)
SP	Speed (Profile Velocity + Target Velocity)	*1)
AC	Acceleration	*1)
EN	Deceleration	
JP	Jog Positive	*1)
JN	Jog Negative	*1)
RR	Repeat Reverse	*1)
RW	Repeat Way	*1)
WT	Wait Repeat	*1)
SO	Set Output	*1)
CO	Clear Output	*1)
TI	Tell Input	*1)
OW	CANopen object: write value	*3)
OR	CANopen object: read value	*3)

Notes:

*1)Compatible with JennyScience AG XENAX® ASCII protocol with restrictions.

*2) REF without parameters only corresponds to PWC, as MovingCap drives do not necessarily have to be referenced. To reset the zero position (Actual Position = 0), execute **REF35** or **REF37**.

Moving Cap also supports the manufacturer-specific method **REF-18** (stop referencing in positive direction) and **REF-19** (negative direction). Additional homing parameters can be set with OW commands for this purpose, see Referencing mode / Homing mode

*3) Write and read object parameters defined in accordance with CiA 402 Servo operation and the object dictionary (**.xdd** file) matching the drive. Examples:

OW3511,8,120 --> Write the value 120 to object 3511h.8h (Input Function 1 Target Position = 120)

OR3511,8 --> Read out object 3511h.8h. Response e.g. **OR3511,8,120**

OR6064,0 --> Readout of CiA 402 standard object 6064h.0h Position Actual Value. Same

Result as TP (tell position). Answer e.g. **OR6064,0,13113**
OR6041,0 --> Readout of CiA 402 standard object 6041h.0h Statusword 42. Response e.g. **OR6041,0,64,1063**

*4) TPSR return values:

TPSR Process Status Value	Associated value Statusword 6041h 42
BIT0 (Hex 1) - ERROR	Fault = Bit 3
BIT1 - REFERENCE, always set	
BIT2 (Hex 4) - IN MOTION	Inverted value from Halt = Bit 8
BIT3 (Hex 8) - IN POSITION	Target Reached = Bit 10
BIT8 (Hex 100) - INVERTER VOLTAGE	Voltage Enabled = Bit 4

*5) TS Return values:

TS Status Value	Description
0	Power Off (see TPSR BIT4 - VOLTAGE)
1	Power On
2	Moving
9	Error (see TPSR BIT0 - ERROR)

*6) GW relative movements:

The movement executed can be relative to the last target position (default setting) or relative to the current position. This is defined via the CiA 402 parameter 60F2h.0h Position Option Code 32 is defined.

4.4.2 RefGo example via web interface

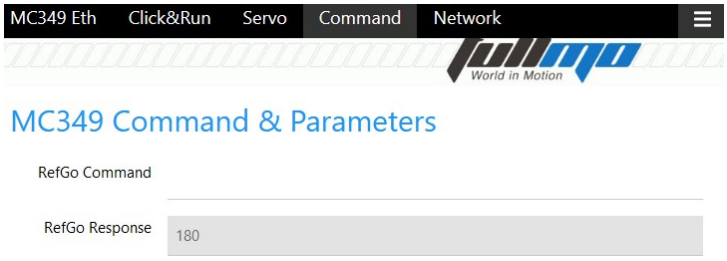
- Select the **Parameters** page / menu (long name "Parameters & Commands")
- For **RefGo Command**, enter one of the commands from the command list 32 command list, e.g.

  G180

  and confirm with the Enter key
- You can confirm that you have reached the target position with

TP

check - you will see the response in the **RefGo Response** line



4.4.3 RefGo example via terminal

Docklight Scripting is a TCP terminal that can be used free of charge in a limited version and is available for direct download here: https://docklight.de/download/Docklight_Scripting.zip

You can find a prepared project with the RefGo commands at movingcap.de , there after Search for **Docklight**.

The following is a docklight communication protocol with a MovingCap turnTRACK 349 drive. The `[PC]` lines are the text commands sent from the PC as TCP client to the MovingCap TCP server port 10001. The `[mc349]` lines are the responses from the MovingCap drive.

```
21.10.2021 13:24:52.889     [PC] - REF35<CR>
```
--> Send REF Command with terminal

```
2021-10-21 13:24:52.908 [mc349] - REF35<CR><LF>
>
```
--> MovingCap confirmed command and sends new prompt for "ready for next command"

```
21.10.2021 13:24:55.520     [PC] - SP720<CR>

21.10.2021 13:24:55.546 [mc349] - SP720<CR><LF>
>
21.10.2021 13:24:56.385     [PC] - AC5000<CR>

21.10.2021 13:24:56.398 [mc349] - AC5000<CR><LF>
>
21.10.2021 13:24:57.342     [PC] - EN1000<CR>

21.10.2021 13:24:57.368 [mc349] - DE1000<CR><LF>
>
21.10.2021 13:24:59.407     [PC] - G360<CR>

21.10.2021 13:24:59.431 [mc349] - G360<CR><LF>
>
21.10.2021 13:25:09.579     [PC] - G-7200<CR>

2021-10-21 13:25:09.601 [mc349] - G-7200<CR><LF>
>
```

```
21.10.2021 13:25:11.685     [PC] - TPSR<CR>

21.10.2021 13:25:11.712 [mc349] - TPSR<CR><LF>
00000006<CR><LF>
>
21.10.2021 13:25:15.181     [PC] - TP<CR>

21.10.2021 13:25:15.207 [mc349] - TP<CR><LF>
-3658<CR><LF>
>
21.10.2021 13:25:20.982     [PC] - TPSR<CR>

21.10.2021 13:25:21.009 [mc349] - TPSR<CR><LF>
0000000a<CR><LF>
>
21.10.2021 13:25:22.840     [PC] - TP<CR>

21.10.2021 13:25:22.866 [mc349] - TP<CR><LF>
-7200<CR><LF>
>
21.10.2021 13:25:26.182     [PC] - PQ<CR>

21.10.2021 13:25:26.210 [mc349] - PQ<CR><LF>
>

21.10.2021 13:53:06.007     [PC] - OR3511.8<CR>

21.10.2021 13:53:06.050 [mc349] - OR3511.8<CR><LF>
OR3511,8,110<CR><LF>            --> Read object 3511h.08h, current value is 110
>

21.10.2021 13:54:36.935     [PC] - OW3511,8,120<CR>

21.10.2021 13:54:37.011 [mc349] - OW3511,8,120<CR><LF>
OW3511,8,120,OK<CR><LF>         --> Write the value 120 to object 3511h.08h
>

21.10.2021 13:55:14.941     [PC] - OR3511,8<CR>

21.10.2021 13:55:15.007 [mc349] - OR3511,8<CR><LF>
OR3511,8,120<CR><LF>            --> Read out 3511h.08h again, current value now 120



>
```

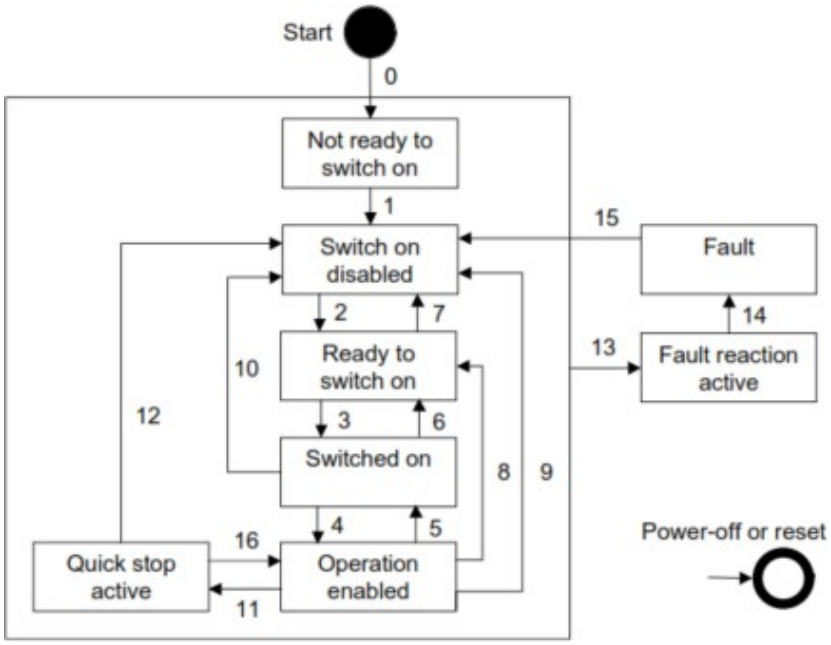
5 Operating modes and settings

MovingCap drives support a wide range of operating modes, functions and objects/parameters in accordance with the **CiA 402** / **DS402** device profile, as well as the **CiA 402 State Machine** (PDS FSA, Power Drive System - Finite State Automat).

CiA 402, alternatively referred to as **DS 402** / Device Standard 402, is a device profile for drives and motion control. It defines the operating modes and application data for frequency inverters, servo controllers and drives. It is part of the CANopen protocol, but is also used in EtherCAT and POWERLINK applications. The CiA 402 device profile is specified in the **IEC61800-7-201** standard as profile type 1 - **CiA402 Drive Profile for power drive systems (PDS)**.

5.1 CiA 402 Operating status / Controlword

The possible operating states and state changes of the MovingCap drives are defined in accordance with the CiA 402 State Machine (PDS FSA, Power Drive System - Finite State Automat):



The operating states are controlled via the following objects/parameters:

Index	Subindex	Type	Access	Name	Description
6040h	0h	unsigned16	R/W	Controlword	Control of the CiA 402 State Machine (PDS FSA). Switches the drive to the desired operating state. See description below.

6041h	0h	unsigned16	RO	Statusword	Current operating status. See detailed description of CiA 402 Statusword 6041h 42

CONTROLWORD 6040H - CHANGE OPERATING STATUS

Bit no. (0-15)	Meaning	Description
0	so	switch on
1	ev	enable voltage
2	qs	quick stop
3	eo	enable operation
4-6	oms	operation mode specific
7	fr	fault reset
8	h	stop
9	oms	operation mode specific
10	r	reserved
11-15	ms	manufacturer specific

Desired command / change of state	Controlword 6040h in binary representation (x = any value)	Possible transitions (arrows in diagram above)
Shutdown	xxxx xx00 0000 x110	2,6,8
Switch on	xxxx xx00 0000 0111	3
Switch on + Enable Operation	xxxx xx00 0000 1111	3 + 4
Disable voltage	xxxx xx00 0000 xx0x	7,9,10,12
Quickstop	xxxx xxxx 0000 x01x	7,10,11

Disable Operation	xxxx xx00 0000 0111	5
Enable Operation	xxxx xx00 0000 1111	4,16
Fault reset	xxxx xx00 1000 xxxx	15

5.2 CiA 402 Betriebsarten / Modes of operation

Object **6060h.0h Modes of operation** selects the desired operating mode.

Index	Subindex	Type	Access	Name [units]	Description
6060h	0h	integer8	R/W	Modes of operation	Operating mode of the drive: 1 = Positioning mode / Profile position mode 3 = Speed mode / Velocity mode 6 = Referencing mode / Homing mode

5.3 Positioning mode / Profile position mode

CIA 402 STANDARD PARAMETERS POSITIONING MODE

Index	Subindex	Type	Access	Name [units]	Description
6060h	0h	integer8	R/W	modes of operation	1 = Positioning mode / Profile position mode
6040h	0h	unsigned16	R/W	Controlword	Control of position mode via CiA 402 Operating states 26 15 (000Fh) = Switch on + Enable

					Operation. Ready for positioning 31 (001Fh) = Next target position is accepted. Each activated movement is executed in full ("Set of Setpoints") 63 (003Fh) = Next target position is approached directly. Trips that are already being processed are canceled / deleted.
607Ah	0h	integer 32	R/W	target position [position units 18 / user defined units] *1)	Target position
6083h	0h	unsigned32	R/W	profile acceleration rotary: fixed unit [100 revolutions/min²] or [0.0277 revolutions/s²] linear: [1000x position units 18 /s²]	Acceleration value Rotary drive: Always referred to motor revolutions/minute. No user-defined scaling. Linear drive: Always in position units 18 /s²
6084h	0h	unsigned d32	R/W	profile deceleration rotary: [100 revolutions/min²] or [0.0277 revolutions/s²] linear: [1000x position units 18 /s²]	Braking value Scaling as for acceleration
6067h	0h	unsigned d32	R/W	position window [position units 18 / user defined units] *1)	Position window +/- the target position for reaching the "Target Reached" status. NOTE: A larger window must be specified here than for 3403h.03h movingcap position window (see below). The different scaling

					must be noted: 6067h.0h is in "user defined units". 3403h.3h is always in internal increments.
6068h	0h	unsigned16	R/W	position window time [ms]	Minimum time in milliseconds in which the actual position must be within the "position window" before the "Target Reached" status is set.
6081h	0h	unsigned32	R/W	profile velocity [position units 18 /s or revolutions/min] *2)	Maximum speed for positioning mode ("profile position mode")
6085h	0h	unsigned32	R/W	quick stop deceleration rotary: [100 revolutions/min²] or [0.0277 revolutions/s²] linear: [1000x position units 18 /s²]	Brake value for quick stop / ("quick Stop")
605Ah	0h	integer 16	R/W	quick stop option code	Configuration of the quick stop behavior ("quick stop") according to The following options of the CiA 402 standard are supported: 0 - Disable drive function / switch off drive

*1) Without different setting of position units/scaling 18 the internal resolution of the measuring system (encoder resolution) is used.

*2) If no other setting is made for position units/scaling 18 the unit used is revolution/minute.

*3) In the case of rotary drives and larger braking values (e.g. > 1000), as well as larger moving masses, the energy/regeneration generated during braking must be taken into account. If necessary, the regenerative energy generated must be absorbed with an appropriately dimensioned braking energy system.

EXAMPLE CIA 402 POSITIONING MODE

To test the CiA 402 positioning mode, the following RefGo **command sequence** can be used via the web interface or terminal/TCP connection can be executed:

RefGo Command **OW6060,0,1** --> *modes of operation = 1 Velocity Mode*

RefGo Command **OW6040,0,6** --> *shutdown*

RefGo Command **OW6040,0,7** --> *switch on*

RefGo CommandOW6040 **,0,15**--> *switch on +*

enabled RefGo CommandOW6081 **,0,1000** --> *profile*

velocity RefGo CommandOW607A **,0,30000** --> *target*

position

RefGo Command **OW6040,0,31** --> *activate single setpoint*

The drive now moves to the target position.

... during the journey ...

RefGo Command **OW6041,0** --> *Query status word*

RefGo Response **OR6041,0,39** --> *Statusword = 0x027 , target not yet reached*

... after reaching the destination ...

RefGo Command **OW6041,0** --> *Query status word*

RefGo ResponseOR6041 **,0,1063** --> *Statusword = 0x0427 , "Target Reached" bit is set, target reached*

MOVINGCAP CONTROLLER PARAMETERS: POSITIONING ACCURACY

Index	Subindex	Type	Access	Name [units]	Description
3401h	3h	unsigned32	R/W	movingcap position window [increments]	Positioning accuracy of the MovingCap controller. The unit is always the internal resolution of the position controller / position measuring system. (see 6067h.0h target reached position window above).

CIA 402 LIMITATION OF THE POSITIONING RANGE / SOFT LIMIT

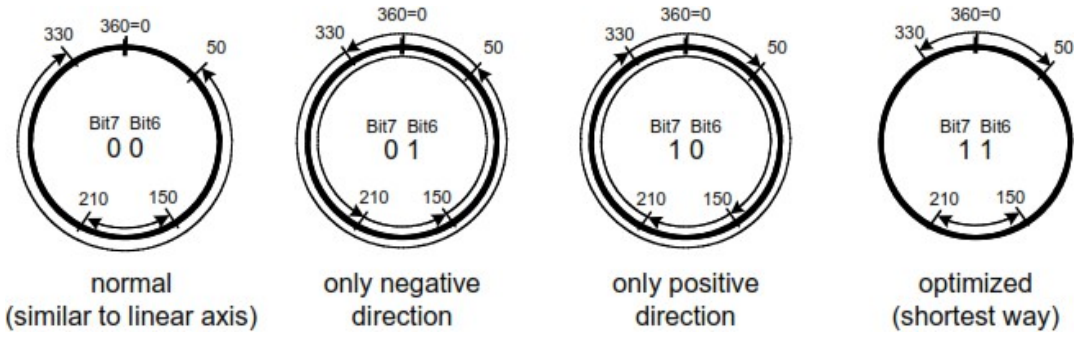
Index	Subindex	Type	Access	Name [units]	Description
607Dh	1h	integer 32	R/W	min position limit [user-defined units, Position units 18] *1)	Minimum possible target position for 607Ah.0h target position (see above).
607Dh	2h	integer 32	R/W	max position limit [user-defined units, Position units 18] *1)	Maximum permitted target position for 607Ah.0h target position (see above)

*1) Without different setting of position units/scaling 18 the internal resolution of the measuring system (encoder resolution) is used.

NOTE: If both values 607Dh.1h = 0 and 607dh.2h = 0 --> no soft limit active / no limitation of the positioning range.

CIA 402 ROTATING AND RELATIVE MOVEMENT

Index	Subindex	Type	Access	Name [units]	Description
60F2h	0h	unsigned16	R/W	position option code	Supports options for relative movements via Bit0 (0001h) and Bit1 (0002h): 60F2h.0h = ...0h - New target position is relative to the last internal target position 60F2h.0h = ...1h - New target position is relative to value from object 60FCh.0h Position demand internal value 60F2h.0h = ...2h - New target position is relative to actual value from object 6064h.0h Position actual value Supports rotating coordinate systems via bit6 (0040h) and bit7 (0080h): 60F2h.0h = 000xh - Linear operation, no movement beyond the min/max position limit. 60F2h.0h = 004xh - Rotating, only negative direction. 60F2h.0h = 008xh - Rotating, only positive direction. 60F2h.0h = 00Cxh - Rotating, shortest travel distance, also beyond the position limits.



CIA 402 SCHNELLSTOPP / QUICK STOP

MovingCap supports quick stop / quick switch-off via the CiA 402 control word .

The behavior when activating a "Quick Stop" is defined via the following objects:

Index	Subindex	Type	Access	Name [units]	Description
6085h	0h	unsigned32	R/W	quick stop deceleration rotary: [100 revolutions/min²] or [0.0277 revolutions/s²] linear: [1000x position units 18 /s²]	Brake value for quick stop / ("quick Stop")
605Ah	0h	integer 16	R/W	quick stop option code	Configuration of the quick stop behavior ("quick stop"). MovingCap drives support the following options of the CiA 402 standard: 0, 1, 2, 5, 6

605Ah.0h Value	Definition according to CiA 402	Description
0	Disable drive function	Switch off drive directly
1	Slow down on slow down ramp and transit into switch on disabled	Brake to zero speed with **6084h.0h profile deceleration**, then switch off.
2	Slow down on quick stop ramp and transit into switch on disabled	Brake to zero speed with **6085h.0h quick stop deceleration**, then switch off.
5	Slow down on slow down ramp and stay in quick stop active	Brake to zero speed with **6084h.0h profile deceleration**, then remain in the quick stop active 26 state.
6	Slow down on quick stop ramp and stay in quick stop active	Brake to zero speed with **6085h.0h quick stop deceleration**, then remain in the quick stop active 26 state.

5.4 Speed mode / Velocity mode

The CiA 402 operating mode "3 - speed operation" is available for the MovingCap rotary drives in order to achieve operation at a constant speed with adjustable acceleration and deceleration ramps.

CIA 402 STANDARD PARAMETERS SPEED OPERATION

Index	Subindex	Type	Access	Name [units]	Description
6060h	0h	integer 8	R/W	modes of operation	3 = Speed mode / Velocity mode
6040h	0h	unsigned16	R/W	Controlword	Activation of speed mode via the CiA 402 Operating state 26 : 15 (000Fh) = Switch on + Enable Operation
6083h	0h	unsigned32	R/W	profile acceleration rotary: [100 revolutions/min²] or [0.0277 revolutions/s²] linear: [1000x position units 18 /s²]	Acceleration value as for positioning mode 28
6084h	0h	unsigned32	R/W	profile deceleration rotary: [100 revolutions/min²] or [0.0277 revolutions/s²] linear: [1000x position units 18 /s²]	Braking value as for positioning mode 28
60FFh	0h	integer 32	R/W	target velocity [position units 18 /sec or revolutions/min] *1)	Target speed

*1) If no other setting is made for position units/scaling 18 the unit used is revolution/minute.

EXAMPLE CIA 402 SPEED OPERATION

To test CiA 402 speed operation, the following RefGo **command sequence** can be used via the web interface 23 or terminal/TCP connection 24 can be executed:

RefGo Command **OW6060,0,3** --> *modes of operation = 3 Velocity Mode*

RefGo Command **OW60FF,0,1000** --> *target velocity*

RefGo Command **OW6040,0,6** --> *shutdown*

RefGo Command **OW6040,0,7** --> *switch on*

RefGo Command **OW6040,0,15** --> *switch on + enabled*

The drive now accelerates up to the specified target speed (60FFh.0h target velocity).

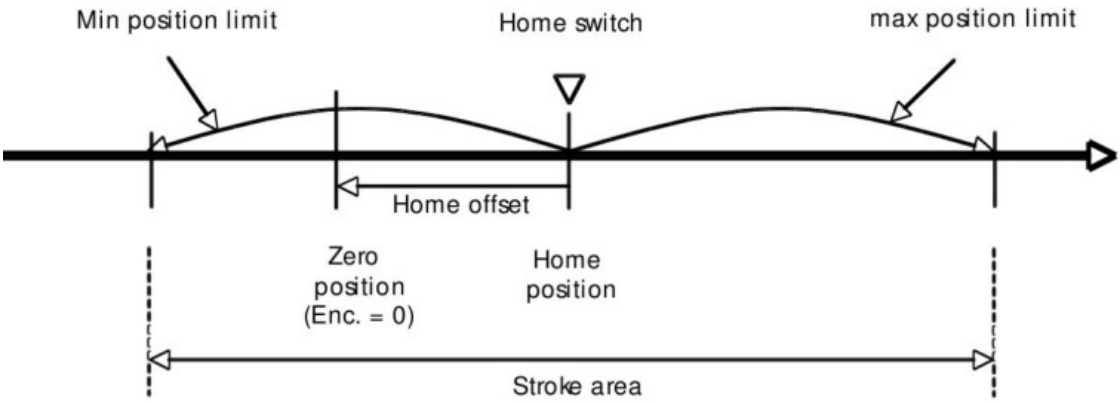
5.5 Referencing mode / Homing mode

CIA 402 STANDARD PARAMETER REFERENCING RUN

Index	Subindex	Type	Access	Name [units]	Description
6060h	0h	integer 8	R/W	modes of operation	6 = Referencing mode / Homing mode
6040h	0h	unsigned16	R/W	Controlword	Control of referencing mode via CiA 402 Operating states 26 15 (000Fh) = Switch on + Enable Operation. Ready for referencing 31 (001Fh) = Execute referencing
6098h	0h	unsigned8	R/W	CiA 402 Homing Method	Common referencing methods: 35 = Set current position as new zero position. Corresponds to RefGo command 21 REF35. -18 = stop referencing in positive direction -19 = stop referencing in negative direction
6099h	1h	unsigned32	R/W	Homing speed during search for switch [position units 18 / user defined units] *2)	Speed during the homing run / search for the mechanical stop
609Ah	0h	unsigned32	R/W	Homing acceleration rotary: [100 revolutions/min²] or [0.0277 revolutions/s²] linear: [1000x position units 18 /s²]	Acceleration value as for positioning mode
607Ch	0h	integer 32	R/W	Home Offset [user-defined units, Position units 18] *1)	Default value = 0 Is defined as the new current position when the reference position / stop is reached.

*1) Without different setting of position units/scaling 18 the internal resolution of the measuring system (encoder resolution) is used.

*2) If no other setting is made for position units/scaling 18 the unit used is revolution/minute.



5.6 Current/torque/force limitation

MOVINGCAP CONTROLLER PARAMETERS: CURRENT/TORQUE/FORCE LIMITATION

For driving mode with specification/limitation of the maximum torque or maximum force, at least the following objects must be set:

Index	Subindex	Type	Access	Name [units]	Description
6073h	0h	unsigned 16	R/W	max current [0.1% of rated current]	Setpoint for torque/force limitation, CiA 402 Standard parameters. Unit: 0.1% of the nominal torque, i.e. 1000 = Nominal torque
3401h	1Ah	unsigned 8	R/W	deactivation block detection (bit value)	Selective deactivation of block travel and following error detection. Bit values for deactivating blockage detection in different movement phases: 3401h.1ah = 1 - Beschleunigung / acceleration 3401h.1ah = 2 - Abbremsen / deceleration 3401h.1ah = 4 - Konstantgeschw. / constant velocity

					3401h.1ah = 8 - Following error monitoring deactivated / deactivate following error*1) 3401h.1ah = 15 (hex 0x0F) - all block and tracking error monitoring deactivated. *1)

*1) Not required / irrelevant for MovingCap MC349.

Deactivation of the block travel/tracking error detection is required for force-limited travel where the force limit changes the trajectory of the travel.

Example: To switch to power-limited driving, the following RefGo command sequence can be issued during the journey via the web interface 24 or terminal/TCP connection can be executed:

RefGo CommandOW3401 **,1A,15** *--> deactivates the blockage and following error detection in all movement phases*

RefGo CommandOW6073 **,0,300** *--> reduces the torque to max. 30% of the nominal torque (maximum torque at standstill)*

If the drive now moves to a mechanical limit, the mechanical stop continues to be pressed with 30% of the nominal force. There is no abort with error code 7121h ERROR_STROKE . After removing the blockage, the drive continues to move to the target position.

ACTUAL VALUES FOR CURRENT/TORQUE/FORCE

The current torque/force effect can be determined via the following object:

Index	Subindex	Type	Access f	Name [units]	Description
6078h	0h	integer16	RO	Current actual value [0.1% of rated current]	Actual value of the current motor (phase) current, proportional to torque/force. CiA 402 Standard parameters. Unit: 0.1% of the rated current/torque/force, i.e. 1000 = rated current/torque/force

The value is signed, e.g:

RefGo Command **OR6078.0** *--> Read out current motor current*

RefGo Response **OR6078,0,992** *--> approx. 10% of the nominal torque in positive Direction*

RefGo Command	**OR6078.0**	*--> Read out current motor current*
RefGo Response *Direction*	**OR6078,0,-1003**	*--> approx. 10% of the nominal torque in negative*

MC349 CONTROLLER PARAMETERS: FORCE LIMITATION

The MC349 (MovingCap 349) motor variant supports the following additional control parameters:

Index	Subindex	Type	Accessf	Name [units]	Description
3401h	15h	unsigned 16	R/W	acceleration torque [0.1 %] *1)	Maximum torque when accelerating, in 0.1% of the nominal torque, i.e. 1000 = nominal torque
3401h	16h	unsigned 16	R/W	deceleration torque [0.1 %] *1)	Maximum torque during braking, in 0.1% of the nominal torque, i.e. 1000 = nominal torque
3401h	18h	unsigned 16	R/W	stall torque [0.1 %] *1)	Maximum torque at standstill, in 0.1% of the nominal torque, i.e. 1000 = nominal torque

NOTE: These parameters are effective in addition to the CiA 402 standard parameter **6073h.0h Max Current** (see above).

MCN23 CONTROLLER PARAMETERS: QUIESCENT CURRENT/FORCE LIMITATION

The MCN23 ETH (MovingCap Stepper) motor variant supports the following additional control parameters:

Index	Subindex	Type	Accessf	Name [units]	Description
3401h	18h	unsigned 16	R/W	stall torque [0.1 %] *1)	Holding torque / holding current at standstill, in 0.1% of the rated torque, i.e. 1000 = rated torque/rated current

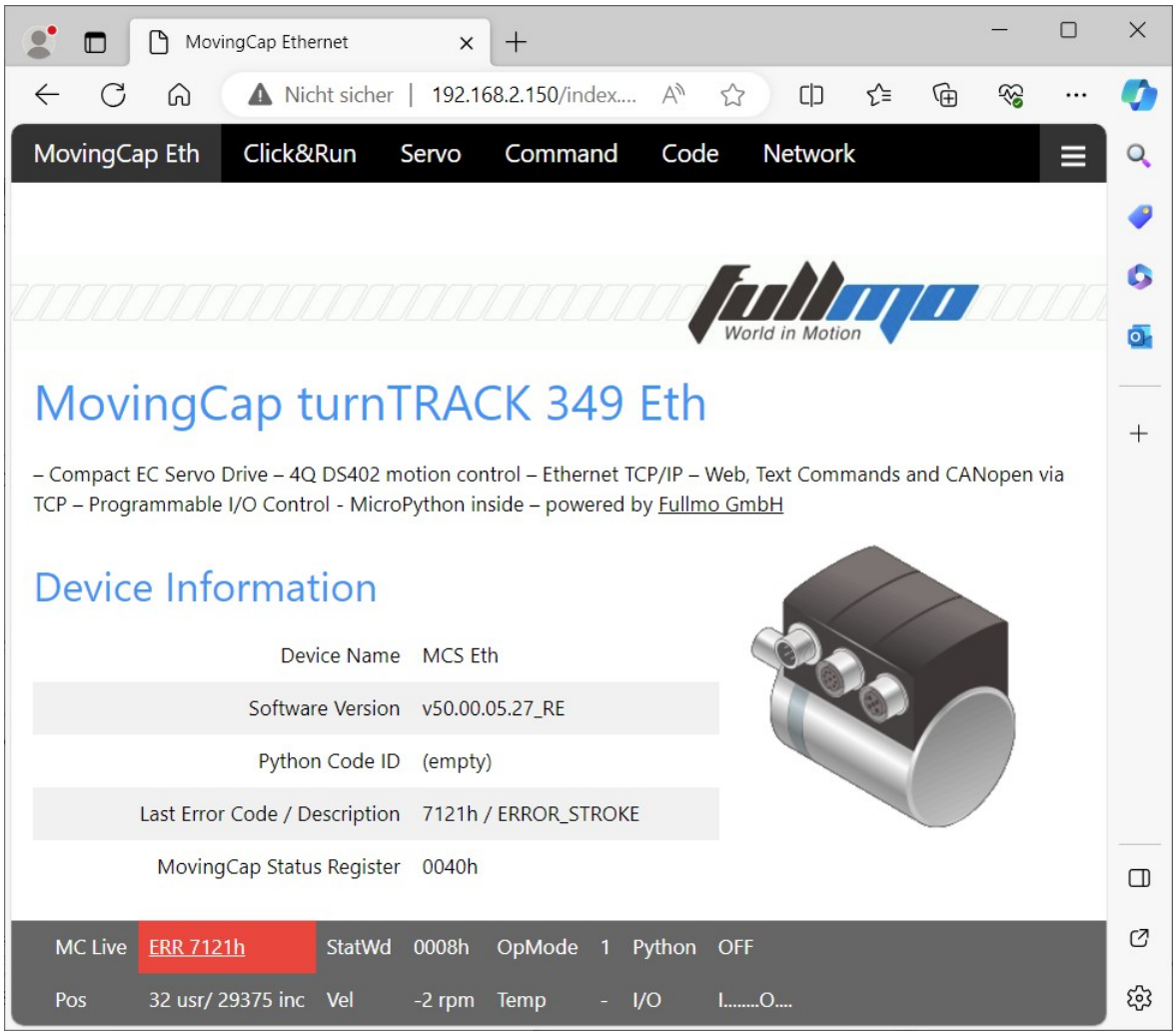
NOTE: The current consumption / max. torque during travel is determined by CiA 402 standard parameter **6073h.0h Max Current** (see above).

NOTE: The default setting of 3401h.18h stall torque is **100** (corresponds to **10% rated current** / rated torque). For higher stall currents and continuous operation, the heat development must be taken into account. For applications with high holding forces in continuous operation, we recommend the use of an MCN23 variant with integrated holding brake.

6 Drive status and error messages

6.1 Status displays on the web interface

The homepage of the drive and the always visible footer provide a quick overview of the drive status:



Designation	Example	Description
MC Live	MOV	General drive status: OFF - Off RDY - Ready to drive / In position

		MOV - In motion ERR error, together with CANopen error code 48 . *1) *2)
StatWd	0427h	Drive status / CiA 402 Statusword 6041h 42
OpMode	1	CiA 402 Betriebsart / Modes of operation 28
Python	OFF	OFF - Script ended/stopped RUN - Script running BRK - Script execution is interrupted/paused (additional) /A - Autostart, see Python program control / Automatic start 71
Pos	32 usr/ 29501 inc	usr - Actual position 50 in the defined position units 18 inc - Actual position in the internal units of the position measurement
Vel	106 usr/ 88 rpm	usr - Actual speed 50 in position units 18 /second rpm - Actual speed in revolutions/minute
Temp	46°C	Motor or electronics temperature, not available in every drive
I/O	I12.4 ....O1.3 .	Status Digital inputs and outputs. The inputs follow after I, the outputs after O. A number indicates that this input or output is active, a dot indicates an inactive I/O. The example on the left shows Inputs 1, 2 and 4 are set. Outputs 1 and 3 are set.

*1) The red ERR error display is a clickable link and leads back to the main page. A text description of the error code can be seen there under **Last Error Code / Description.** The error codes and descriptions correspond to the CiA 402 Error Codes / Emergency 48 .

*2) The **Last Error Code** on the homepage will continue to be listed even after the error has been rectified in order to facilitate diagnosis. Only if a red **ERRxxxx** error is displayed in the footer at the bottom left does the error currently exist.

6.2 Drive status / CiA 402 Statusword 6041h

Object **6041h.0h Statusword** provides information on the drive status in accordance with the CiA 402 State Machine 28 .

STATUS WORD 6041H - CURRENT DRIVE STATUS

Bit no. (0-15)	15 14	13 12	11	10	9	8	7	6	5	4	3	2	1	0
Significance	ms	oms	ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso

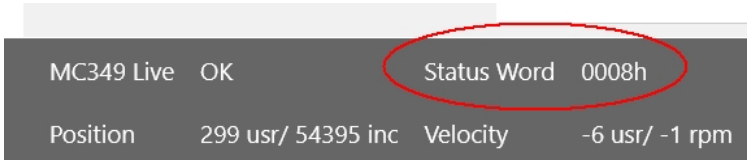
Legend:

Bit no. (0-15)	**Meaning**	**Description**
0	rtso	ready to switch on
1	so	switched on
2	oe	operation enabled
3	f	rotten
4	ve	voltage enabled
5	qs	quick stop
6	sod	switch on disabled
7	w	warning
8	ms	manufacturer-specific
9	rm	remote
10	tr	target reached
11	ila	internal limit active
12,13	oms	operation mode specific
14,15	ms	manufacturer-specific

Statusword 6041h in binary representation (x = any value)	**PDS FSA Status**
xxxx xxxx xOxx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx xOxx 1111	Fault reaction active
xxxx xxxx xOxx 1000	Fault

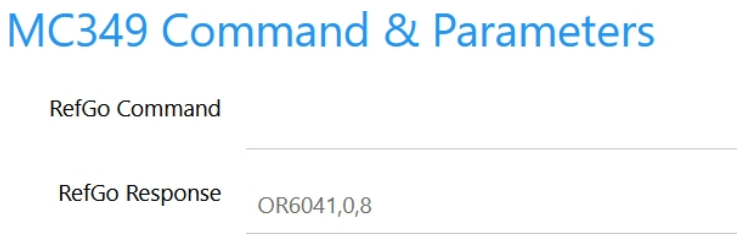
DISPLAY / QUERY OF THE STATUSWORD

You can see the current value of the 6041h status word in the web interface at the bottom:

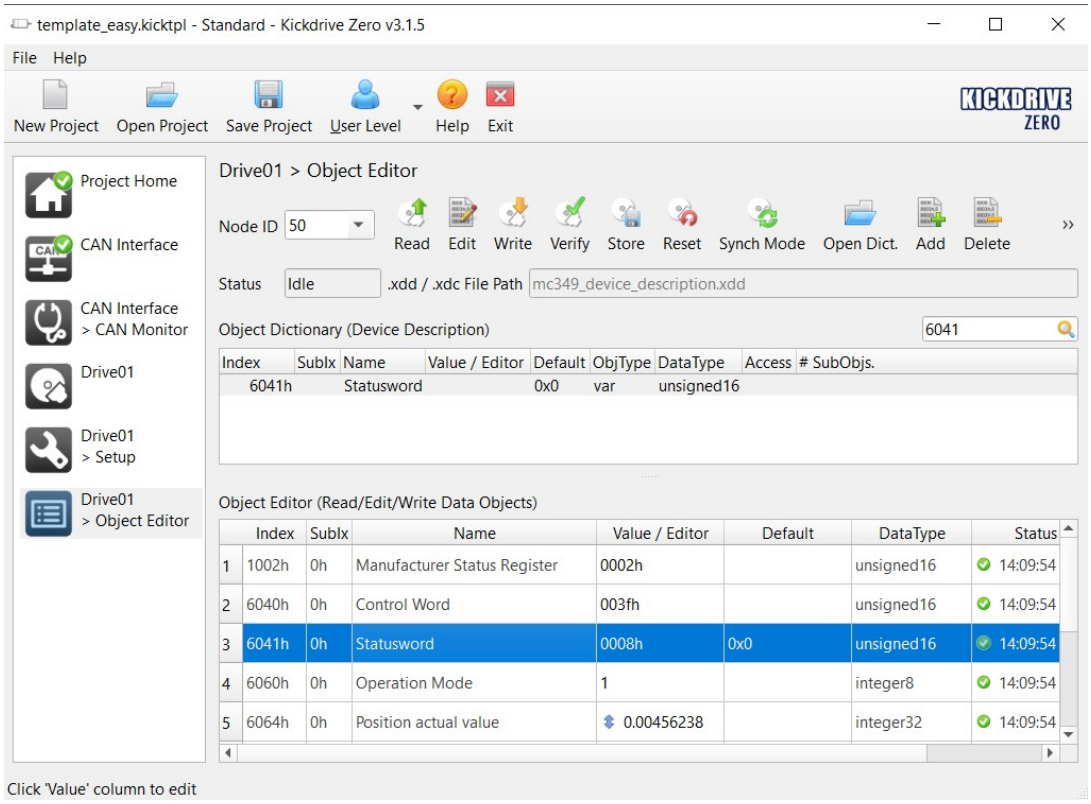


or via the following RefGo command 23 query:

OR6041.0



or via the Kickdrive 20 **Object Editor**:



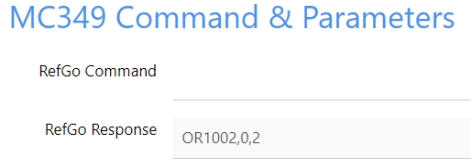
6.3 MovingCap status register 1002h

Details on drive faults and warnings can be accessed via the object

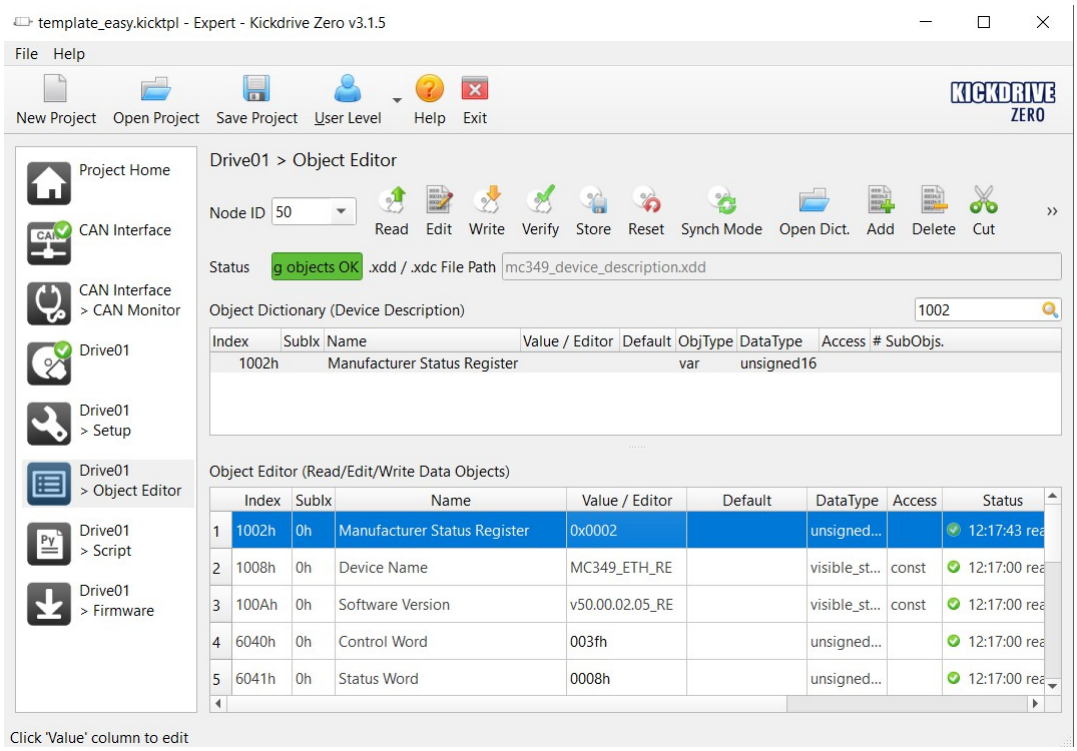
1002h.0h Manufacturer Status Register

are displayed. To read out the current value, you can use the web interface to 23 in the Parameters area:

RefGo Command **OR1002,0**



or in Kickdrive 20 use the **Object Editor**:



STATUS REGISTER 1002H.0H - ERROR

The following states are error states. When these errors occur, the drive switches to the CiA 402 Fault state, generates a CANopen Emergency Message and the Fault bit in the statusword is set.

Object 1002h.0h Bit values (combinable)	Description
Bit0 / 0001h / 1 dec	Error over volt (Uzk)
Bit1 / 0002h / 2 dec	Error under volt (Uzk)
Bit2 / 0004h / 4 dec	Error Ack
Bit3 / 0008h / 8 dec	Error over temp
Bit4 / 0010h / 16 dec	Error I2T / Derating
Bit5 / 0020h / 32 dec	Abort connection
Bit6 / 0040h / 64 dec	Error stroke
Bit7 / 0080h / 128 dec	Error communication
Bit8 / 0100h / 256 dec	Error sensor
Bit9 / 0200h / 512 dec	Error Hardware / Software Enable
Bit11 / 0800h / 2048 dec	Error Over Current
Bit12 / 1000h / 4096 dec.	Error External Force / Torque
Bit13 / 2000h / 8192 dec.	Error Needs Voltage Reset

STATUS REGISTER 1002H.0H - WARNINGS

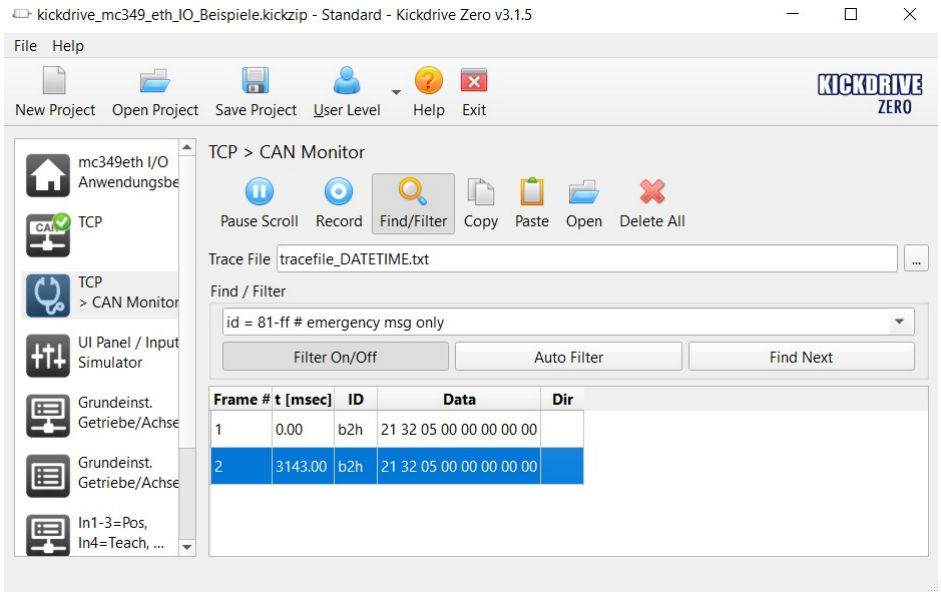
Bit values from bit16 / 10000h / 65536 dec. are warning states. They do not lead to a drive error, the current operating mode is retained and movement commands continue to be executed.

Object 1002h.0h Bit values (combinable)	Description
Bit16 / 10000h / 65536 dec.	Warning sensor *1)

*1) E.g. warning status of a multiturn encoder system: Position detection is still possible, but a warning such as "Battery Low" is pending. MovingCap multiturn absolute encoders have a lifetime battery buffer for >10 years - only in the event of an error/defect can a sensor warning be triggered after a short operating time.

6.4 CiA 402 Error messages / Emergency messages

The Kickdrive software can be used to precisely determine error states and the time at which they occur. To do this, activate the **CAN Monitor** window and start communication in Kickdrive before carrying out your test. If necessary, use the **Find/Filter** function to search specifically for so-called CANopen Emergency Messages (**id = 81-ff**):



CANOPEN EMERGENCY LIST / ERROR CODES

Error code	CAN Message / Data	Description
2220h / 8736 dec	20 22 05 00 00 00 00 00	ERROR_OVER_CURRENT Overcurrent
3211h / 12817 dec	11 32 05 00 00 00 00 00	ERROR_OVER_VOLT Overvoltage UZK Power supply motor
3221h / 12833 dec	21 32 05 00 00 00 00 00	ERROR_UNDER_VOLT Undervoltage UZK Power supply motor
4210h / 16912 dec	10 42 09 00 00 00 00 00	ERROR_OVER_TEMP Overtemperature
4220h / 16928 dec	20 42 05 00 00 00 00 00	ERROR_DERATING Derating
5200h / 20992 dec	00 52 09 00 00 00 00 00	ERROR_ACK Acknowledge Error

7121h / 28961 dec	21 71 10 00 00 00 00 00	ERROR_STROKE Blockage / block travel / drag error too large
7510h / 29968 dec	10 75 10 00 00 00 00 00	ERROR_COMMUNICATION Motor communication error
FFF1h / 65521 dec	F1 FF 10 00 00 00 00 00	ERROR_SENSOR_1 Sensor/encoder error
FFF2h / 65522 dec	F2 FF 10 00 00 00 00 00	ERROR_HW_SW_ENABLE HW Enable error (missing internal enable) or SW Enable error (see input function 9 62)
FFF4h / 65524 dec	F4 FF 10 00 00 00 00 00	ERROR_EXT_TORQUE External force/external moment
FFF5h / 65525 dec	F5 FF 10 00 00 00 00 00	ERROR_NEEDS_RESET A power reset is required before further operation, e.g. after a successful software update.

6.5 CiA 402 Error codes / History

The error codes reported as CANopen Emergency 46 can also be read out via object **603Fh.0h Error Code.** For example, you can use the web interface 68 to in the Parameters area:

RefGo Command **OR603F,0**

However, the displayed return value is always decimal and you may have to convert the value to hexadecimal using the (Windows) calculator, e.g.

RefGo Response OR603F **,0,12833** *--> Error according to table in CANopen Emergency 48 -> 3221h / 12833 decimal = undervoltage UZK power supply motor*

The last four error codes can be displayed via the objects **1003h.1h-4h Standard error field**, e.g.

RefGo Command **OR1003,1**

RefGo Response **OR1003,1,12833**

6.6 Actual values - position, speed, current, temperature

CIA 402 STANDARD ACTUAL VALUES FOR POSITION, SPEED AND MOTOR CURRENT

Index	Subindex	Type	Access	Name [units]	Description
6064h	0h	integer32	RO	Position actual value [position unit en 18 / user defined units] 1*)	Current actual position, taking into account the set position units and zero position 18 , as well as special setting for rotating and relative movement 32 .
606Ch	0h	integer32	RO	Velocity actual value [Position unit en 18 /sec or revolutions/min] 2*)	Current speed
6078h	0h	integer16	RO	Current actual value [0.1% of rated current]	Actual value of the current motor (phase) current, proportional to torque/force, see also Current/torque/force limitation 38 .

*1) Without different setting of position units/scaling 18 the internal resolution of the measuring system (encoder resolution) is used.

*2) If no other setting is made for position units/scaling 18 the unit used is revolution/minute.

MOVINGCAP ACTUAL VALUE: TEMPERATURE

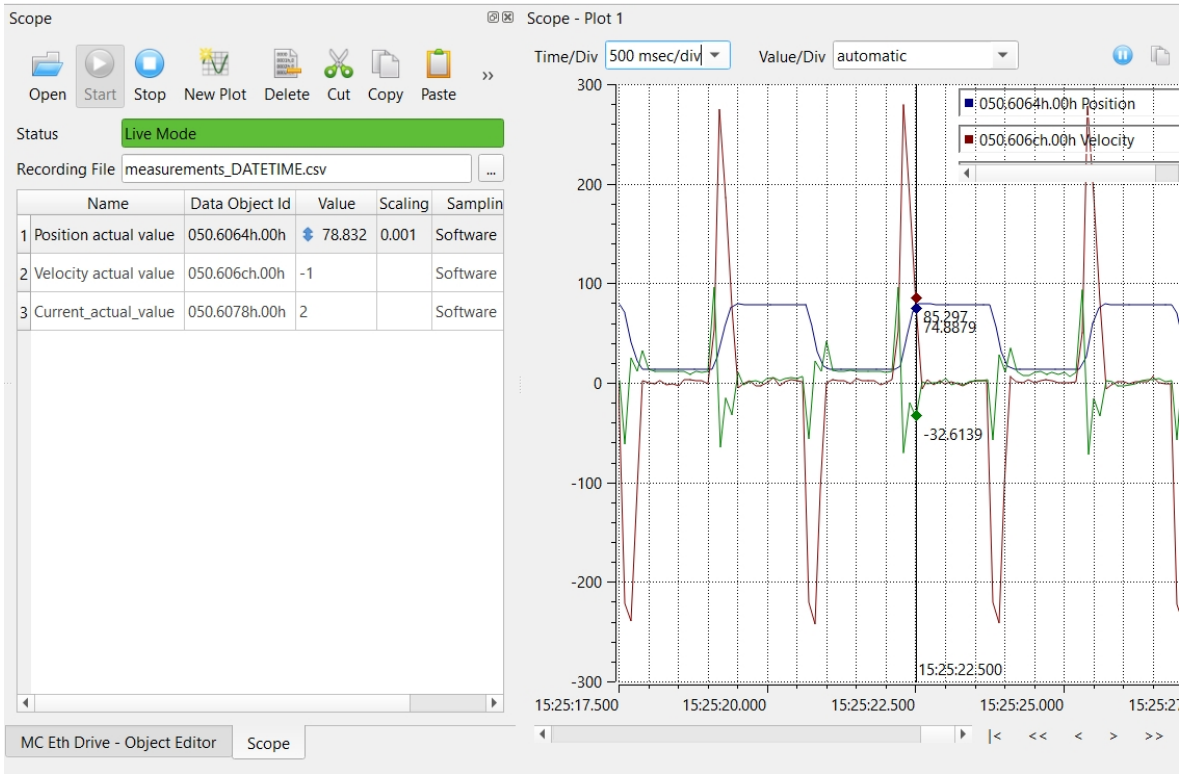
Index	Subindex	Type	Access	Name [units]	Description
3401h	0Ah	unsigned 8	RO	motor actual temperature [°C]	Current motor temperature. Depending on the MovingCap variant, the winding or electronics temperature, or the combined measured value (switchover between electronics and motor temperature every second).

To read out actual values, you can enter the corresponding RefGo command via the web interface 23 or send it to TCP port 10001.

Example for reading the temperature value:

RefGo Command	**OR3401,A**	*--> Object Read 3401h, 0Ah*
RefGo Answer	**OR3401,A,61**	*--> Answer: 61 [°C]*

Or use MovingCap SLCAN - CANopen CiA 402 with Kickdrive 20 and the convenient Scope in Kickdrive with recording and trigger functions:



7 MovingCap IO

Depending on the motor variant and options, the MovingCap Ethernet drives offer

- up to 10x digital inputs 24V DC (IN1-IN10)
- up to 4x digital outputs 24V DC (OUT1-4)

The connection/pin assignment can be found in the Connections 55 section .

The current status of the inputs/outputs is displayed at the bottom right of the web page:

A digit after "I" means that this input number (IN..) is active. A digit after "O" means that this output number (OUT..) is active. In the picture above, outputs OUT2 and OUT3 are high, all inputs are low.

Software access to the digital inputs and outputs is via the following objects in the CANopen object dictionary:

Index	Subindex	Type	Access	Description	Example via web interface/RefGo 23
60FDh	00h	Unsigned16	R	Physical Inputs IN1 = Bit16 (00010000h) IN2 = Bit17 (00020000h) IN3 = Bit18 (00040000h) IN4 = Bit19 (00080000h) ...	RefGo Command: **OR 60FD,0** RefGo Response: **OR 60FD,0,131072** --> 24V are applied to IN2 (IN2=high)
60FEh	01h	Unsigned16	R/W	Physical Outputs OUT1 = Bit16 (00010000h, decimal 65536) OUT2 = Bit17 (00020000h, decimal 131072) ...	RefGo Command: **OW 60FE,1,65536** RefGo Response: **OW 60FE,1,65536,OK** --> set OUT1 = high, (OUT2.. = low) RefGo Command: **OW 60FE,1,393216** RefGo Response: **OW 60FE,1,393216,OK** --> set OUT2 = High, OUT3 = High (OUT1 or OUT4) = low

NOTE: The assignment of the inputs and outputs to objects 60FDh.00h and 60FEh.01h with bit numbers >= 16 ("manufacturer-specific") corresponds to the CiA 402 specification.

7.1 Input functions

Each digital input can trigger one of the input functions described below. The following can be set

for each input:

- Which function is triggered on rising edge (input changes from 0V 'Low' to 24V 'High').

- Which function is triggered on a falling edge (input changes from 24V 'High' to 0V 'Low').

The functions are set via the following parameter objects in the CANopen object dictionary:

Object 3511h - Reaction to IN1
Object 3512h - Reaction to IN2
Object 3513h - Reaction to IN3
...
Object 3518h - Reaction to IN8
Object 3519h - Reaction to IN9
Object 351Ah - Reaction to IN10

NOTE: Each MovingCap Ethernet drive offers up to 10 programming objects, even if the motor - depending on the model and options - offers fewer than 10 digital inputs as an electrical interface. The additional objects can be used to store further travel profiles, which can be started via object 3510.01h Virtual Inputs or object 3510h.02h Input Fct Command.

TIP: Kickdrive sample projects with prepared object lists and sample configurations can be found in the movingcap.de service portal, where you can search for **MovingCap IO**.

7.1.1 Structure of objects 3511h - 351Ah

Each of the objects 3511h - 351Ah consists of the following elements:

Subindex	Type	Access	Standard function *1)
01h	Unsigned16	R/W	HiByte - Function number rising edge LoByte - Function number falling edge
02h	Unsigned16	R/W	HiByte - TxPDO no. rising edge LoByte - TxPDO no. falling edge
03h	Integer32	R/W	Results register 1
04h	Integer32	R/W	Results register 2
05h	Integer32	R/W	Timer - Time delay in milliseconds for the execution of the "Timed Motion" and "Set Home" functions
06h	Integer32	R/W	Target speed
07h	Integer32	R/W	Profile acceleration / acceleration value
08h	Integer32	R/W	Target position
09h	Integer32	R/W	Profile deceleration / Braking value
0Ah	Integer32	R/W	Max Current / Torque limitation

*1) All parameters from subindex 03h are implemented as Integer32, even if this is not required for some of the standard functions. For individual input functions, e.g. function 7 - Set Home / Referencing 61 individual subindexes have different meanings.

7.1.2 Function 1 - Capture position

Function 1 saves the current position in result register 1 when triggered.

Subindex	Type	Access	Description
03h	Integer32	R/W	Result register 1 : the current position when the event is reached [user-defined units, position units 18]

CONFIGURATION EXAMPLES

Configuration	Explanation
3511h.01h = 0100h	When a rising edge occurs at digital input 1, the current position is stored in result register 1 (subindex 3).
3512h.01h = 0001h	When a falling edge occurs at digital input 2, the current position is stored in result register 1 (subindex 3).
3513h.01h = 0102h	When a rising edge occurs at digital input 3, the current position is stored in result register 1 (subindex 3). When a falling edge occurs at digital input 3, the current position is stored in result register 2 55 (subindex 4).

7.1.3 Function 2 - Capture position

Function 2 saves the current position in result register 2 when triggered.

Subindex	Type	Access	Description
04h	Integer32	R/W	Result register 2 : the current position when the event is reached [user-defined units, position units 18]

CONFIGURATION EXAMPLES

Configuration	Explanation
3511h.01h = 0200h	When a rising edge occurs (digital input 1), the current position is stored in result register 2 (subindex 4).
3511h.01h = 0002h	When a falling edge occurs (digital input 1), the current position is stored in result register 2 (subindex 4).
3511h.01h = 0102h	When a rising edge occurs at digital input 1, the current position is stored in result register 1 (subindex 3). When a falling edge occurs at digital input 1, the current position is stored in result register 2 (subindex 4).

7.1.4 Function 3 - Counter

When triggered, function 3 increments the content of result register 1 by 1.The content of result register 1 can be initialized by a write access.

CONFIGURATION EXAMPLES

Configuration	Explanation
3511h.01h = 0300h	If a rising edge occurs (digital input 1), the content of result register 1 (subindex 3) is increased by 1.
3511h.01h = 0003h	If a falling edge occurs (digital input 1), the content of result register 1 (subindex 3) is increased by 1.
3511h.01h = 0303h	If a rising edge occurs (digital input 1), the content of result register 1 (subindex 3) is increased by 1. If a falling edge occurs (digital input 1), the content of result register 1 (subindex 3) is increased by 1.

7.1.5 Function 4 - Timed Motion Absolute

When function 4 occurs, it triggers a previously configured move command after a defined time has elapsed:

- Change to positioning mode if not already in this state.
- Moving to the target position at the specified speed/acceleration.

Subindex	Type	Access	Description
03h	Integer32	R/W	Result register 1 : last used target position in internal position units [increments] This value is used as a comparison value for output function 4 - Positioning feedback 66.
04h	Integer32	R/W	Result register 2 : Number of movement commands triggered
05h	Integer32	R/W	Timer - time delay in milliseconds
06h	Integer32	R/W	Target speed
07h	Integer32	R/W	Profile acceleration / acceleration value
08h	Integer32	R/W	Target position
09h	Integer32	R/W	Profile deceleration / Braking value
0Ah	Integer32	R/W	Max Current / Torque limitation [1/1000 of rated current], e.g. Setting value 250 = 25% of the rated current/torque

and other settings of the position controller in accordance with the settings for positioning mode 28.

CONFIGURATION EXAMPLE

Object values	As OW commands via the website	Explanation
3511h.01h = 0400h 3511h.05h = 1000 3511h.08h = 6300 3511h.06h = 500 3511h.07h = 400 3511h.09h = 200	OW3511,1,1024 OW3511,5,1000 OW3511,8,6300 OW3511,6,500 OW3511,7,400 OW3511,9,200	When a rising edge occurs (digital input 1) after one second the position 6300 approached. Driving parameters are speed = 500, acceleration = 400, braking value = 200 The OW commands can be used via the web interface 23 or terminal/TCP connection 24.

7.1.6 Function 5 - Quick Stop

Function 5 activates "Quick Stop" with the following parameter:

Object **605Ah Quick stop option code = 6 : "**Slow down on quick stop ramp and stay in quick stop active"

The drive decelerates with the acceleration set in object

6085h Quick stop deceleration

was determined.

NOTE: To deactivate positioning completely, input function 9 - Software Enable 62 can be used.

7.1.7 Function 6 - Timed Motion Relative

Like function 4 57 but relative/continuously adding positioning

Subindex	Type	Access	Description
03h	Integer32	R/W	Result register 1 : Last used target position absolute in internal position units [increments. This value is used as a comparison value for output function 4 - Positioning feedback 66.
04h	Integer32	R/W	Result register 2 : Number of movement commands triggered
05h	Integer32	R/W	Time delay in milliseconds
06h	Integer32	R/W	Target speed
07h	Integer32	R/W	Profile acceleration / acceleration value
08h	Integer32	R/W	Target position relative, taking into account 60F2h.0h position option code 32
09h	Integer32	R/W	Profile deceleration / Braking value
0Ah	Integer32	R/W	Max Current / Torque limitation [1/1000 of rated current], e.g. Setting value 250 = 25% of the rated current/torque

NOTE: For continuous relative movements (e.g. always 180° forwards), 60F2h.0h = ...0 32 (relative to the last internal target position) can be used. This ensures that the difference between the exact target position and the current actual position resulting from the control accuracy does not add up to a relevant total error.

NOTE: After positioning mode is switched on, the current actual position is always specified as the output value for relative positioning. A home position is usually approached via absolute positioning, input function 4 - Timed Motion Absolute 57.

7.1.8 Function 7 - Set Home / Referencing

Referencing: Setting the zero position or referencing movement to the limit switch or mechanical stop.

Subindex	Type	Access	Standard function *1)
05h	Integer32	R/W	Timer - Time delay in milliseconds for the execution of referencing
06h	Integer32	R/W	Target speed
07h	Integer32	R/W	Profile acceleration / acceleration value
08h	Integer32	R/W	Homing Offset Which position value should be set as the current position after successful completion of the referencing run? Default setting = 0
09h	Integer32	R/W	CiA 402 Homing Method e.g: 35 = Set current position as new zero position. Corresponds to RefGo 21 command **REF35**. -18 = stop referencing in positive direction -19 = stop referencing in negative direction
0Ah	Integer32	R/W	Homing torque / torque limitation for the homing run [1/1000 of rated current], e.g. setting value 250 = 25% of the rated current/torque

7.1.9 Function 8 - Teach target

If the input is set to High, the behavior of all other inputs changes:

- If another input is assigned function **4 Timed Motion Absolute**, no movement is executed when this contact is in the High state, but the current position is stored as the new target position (Teach Target).

7.1.10 Function 9 - Software Enable

Function 9 can be used to simplify application creation and to define an input contact as a precondition for the "Switched On" state. 26 state.

WARNING! The "Software Enable" function is NOT a safety function. It is NOT intended to implement safety functions in the context of machine safety.

NOTE: If the software enable input is missing, the error code 48 **FFF2** / **ERROR_HW_SW_ENABLE** is generated.

Subindex	Type	Access	Value / Function
01h	Unsigned16	R/W	0900h - Actuator can only be switched to the "Switched On" state if input = High 0009h - Actuator can only be switched to the "Switched On" state if input = Low

7.1.11 Simulate inputs

To test input functions without electrically connecting the electrical inputs of the drive to 0V or 24V, you can use the following objects:

Index	Subindex	Type	Access	Description	Example via web interface/RefGo 23
3510h	01h	Unsigned32	R/W	Virtual Inputs Simulate "Input INx is 24V / High". Value range as for object 60FD 51 i.e. IN1 = Bit16 (00010000h) IN2 = Bit17 (00020000h) IN3 = Bit18 (00040000h) IN4 = Bit19 (00080000h) ...	RefGo Command: **OW 3510h,1,131072** RefGo Response: **OW 3510h,1,131072,OK** --> Actuator now "sees" a 24V /High state for IN2 and triggers the defined input function if necessary.
3510h	02h	Unsigned16	R/W	Input Function Command Parameter format YYZZh - Triggers input function ZZ with parameters from input no. YY. e.g. 0304h - Executes "Function 4 - Timed Motion Absolute" with the movement parameters stored at input no. 3 / object 3513h.	RefGo Command: **OW 3510h,2,772** RefGo Response: **OW 3510h,2,772,OK** --> corresponds to the HEX numerical value 0304h, i.e. function 4 is executed as described on the left.

7.2 Output functions

Each digital output can be parameterized for the output functions described below.

Parameterization is carried out via the CANopen objects:

Object 3611h - Control of OUT1
Object 3612h - Control of OUT2

Subindex	Type	Access	Description
1	Unsigned16	R/W	Function number
2 .. 6		R/W	(see function description of the selected function number)

7.2.1 Function 1 - Statusword 6041h Monitor

Monitoring the CiA 402 status word 6041h 42.

Set the output if the selected bit(s) in object 6041h (CiA 402 status word) are set.

Subindex	Designation	Default
2	Bit mask (combinable) according to CiA 402 Statusword 6041h 42	

7.2.2 Function 2 - Error Register 1002h Monitor

Monitoring of the MovingCap error register 1002h 45

Set the output if the selected bit(s) in object 1002h are set.

Subindex	Designation
2	Bit mask (combinable), according to MovingCap error register 1002h.

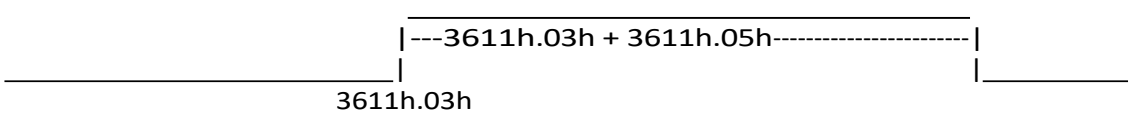
7.2.3 Function 3 - Limit monitoring right/left

Activation and deactivation via position condition. The pulse duration is defined via a position window (right/left position distance).

Subindex	Designation
2	1 = right limit active 2 = left limit active 3 = both
3	right position
4	left position
5	right position distance
6	left position distance

Example :
3611.01h = 3
3611.02h = 1

--> Drive in a positive direction,



7.2.4 Function 4 - Positioning feedback

Confirms that the target position has been reached / held. General or specific for an input with input function 4 - "Timed Motion Absolute" 57

Index	Subindex	Value / Description
3611h-3614h	2	0: last target position reached (independent of input). The comparison position here is the position according to position option code (object 60F2h.0h) 32. With 60F2h.0h = 2 ("use actual value"), this setting is not useful as this condition is always fulfilled. 1-8: Number of the input (IN1-IN8) - Compare with the target position from the last movement that was triggered by this input. This position can also be read out in the "Result 1" register of the input. 255: as 1-8, but bit-coded feedback: Which IN position is currently being held? See example below.
	5	right position window: Permissible deviation of the actual position in positive direction
	6	left position window: Permissible deviation of the actual position in negative direction

NOTE: Output function 6 - Feedback on position and status 67 the actuator status can also be taken into account.

DESCRIPTION OF SUBINDEX 2 = 255 BIT-CODED FEEDBACK

For bit-coded output, the output functions for **OUT1** and **OUT2** must be parameterized in the same way:

Index	Subindex	Value
3611h	1	4
3611h	2	255
3612h	1	4
3612h	2	255

If **IN1** to **IN3** are parameterized for function 4 - Timed Motion Absolute are parameterized, **OUT1** and **OUT2** together to provide feedback on the current position:

OUT1	OUT2	Condition
low	low	None of the positions from IN1-3 are currently held
high	low	The position from IN1 has been reached / is held
low	high	IN2 has been reached / is being held
high	high	IN3 has been reached / is being held

NOTE: If more than 2 outputs are available, OUT1/2/3 can be parameterized as above and provide up to IN8 as feedback.

7.2.5 Function 5 - Drive without error/warning

General status feedback:

Output set / high: Normal state, there is no warning or error

Output reset / low: Set in CiA 402 status word 6041h 42 is bit 7 (warning) or bit 3 has (error), or the MovingCap error register 1002h 45 (set. at least one error bit

7.2.6 Function 6 - Position and status feedback

As function 4 66 but additionally with drive status check:

- Actuator in positioning mode, without error or warning 67
- Last target position reached, i.e. "target reached" bit in the status word 42 is set

8 MovingCap CODE - Python Programming

MovingCap drives with CODE support have a built-in Micropython interpreter to permanently store your own small programs for autonomous sequences and function extensions in the drive.

NOTE: Please note that a program permanently stored in the MovingCap can set the drive in motion as soon as the power supply is switched on.

To stop the drive permanently, the sequence program must also be stopped, e.g. using the web interface functions described below.

NOTE: MovingCap currently has MicroPython v1.9.4 built in, however the MicroPython modules are not fully supported. In particular, there is currently no support for usocket network functions. Our documented MovingCap specific modules and commands are supported 71.

8.1 A simple test program

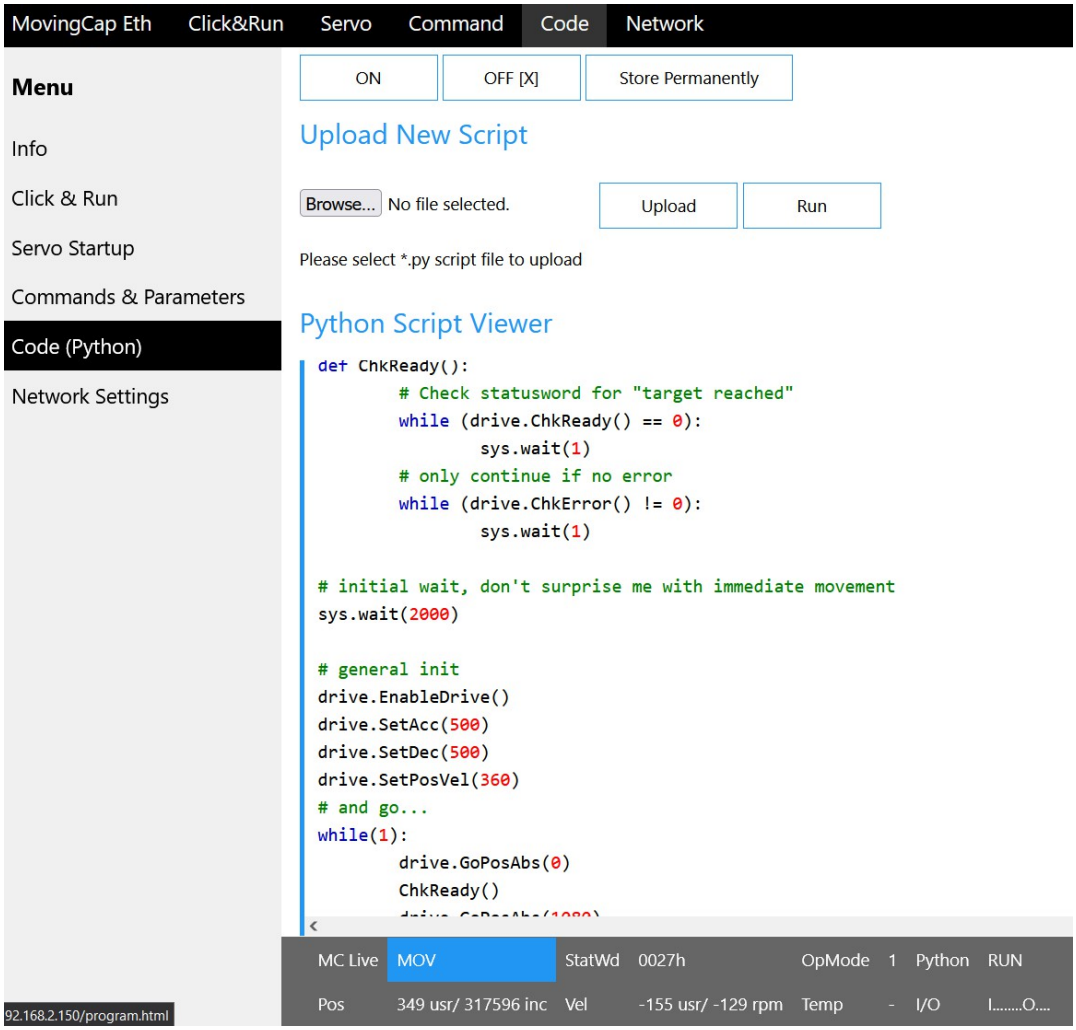
EXAMPLE FOR MOVINGCAP TURNTRACK349 ETHERNET OR OTHER ROTARY ACTUATORS

Here is a simple example of a rotary motor that is to work with angular coordinates (360° correspond to one revolution):

```
#id driveSimplePosTest.py 2024-07-25 oh
# This is a simple positioning demo for a MovingCap turnTRACK 349
# rotary drive.
# it will set the "feed constant" to 360, so one motor (optionally:
gear)
# revolution corresponds to a position change of 360°
import sys
import mcdrive as mc

def ChkReady():
      # Check statusword for "target reached"
      while (mc.ChkReady() == 0):
            sys.wait(1)
      # only continue if no error
      while (mc.ChkError() != 0):
            sys.wait(1)

# initial wait, don't surprise me with immediate movement
sys.wait(2000)

# Set 6092h.01h Feed constant to 360 --> one turn = 360°
mc.WriteObject(0x6092, 0x01, 360)

# general init
mc.EnableDrive()
mc.SetAcc(500)
mc.SetDec(500)
mc.SetPosVel(360)
# and go...
while(1):
      mc.GoPosAbs(0)
      ChkReady()
```

```
mc.GoPosAbs(1080)
ChkReady()
mc.GoPosAbs(360)
ChkReady()
```

- Save this program in a .py file, e.g. as **driveSimplePosTest.py** .
- In the MovingCap web interface, open the **Code (Python)** area
- Select this file via **Browse...** and click on the **Upload** button to the right of it.



- Make sure that the drive can start up safely and start the program with the **Run** button

EXAMPLE PROGRAM FOR A LINEAR AXIS WITH PRESET TRAVEL LIMIT (SOFT LIMIT)

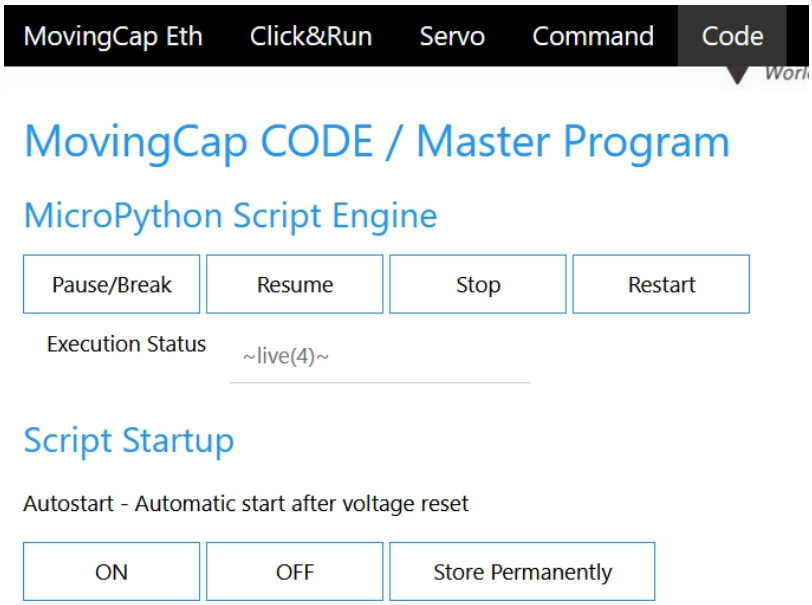
Here is an alternative demo program, suitable for a linear drive with micrometer position units. The speed `mc.SetPosVel(50000)` used below corresponds to 5cm/second in this case.

```
#id driveSoftlimitBackForth.py 2024-07-25 oh
# simple MovingCap flatTRACK application demo
# precondition: softlimit objects 607Dh.1h and 607D.2h set
import sys
import mcdrive as mc

def ChkReady():
      # Check statusword for "target reached"
      while (mc.ChkReady() == 0):
            sys.wait(1)
      # only continue if no error
      while (mc.ChkError() != 0):
            sys.wait(1)

# safety: only run this script if softlimit objects are set up properly
softLimitMin = mc.ReadObject(0x607d, 1)
softLimitMax = mc.ReadObject(0x607d, 2)

if ((softLimitMin == 0 and softLimitMax == 0) or softLimitMax <=
(softLimitMin + 1000)):
      print("flatTRACK demo abort. Check softlimit objects 607Dh.1h and
607D.2h!")
else:
    # initial wait, don't surprise me with immediate movement
    sys.wait(5000)

    # general init
    mc.EnableDrive()
    mc.SetAcc(50)
    mc.SetDec(50)
    mc.SetPosVel(50000)
    # and go...
    while(1):
        mc.GoPosAbs(softLimitMin)
        ChkReady()
        mc.GoPosAbs(softLimitMax)
        ChkReady()
```

- Save this program in a .py file as described above and run it in your MovingCap flatTRACK linear actuator.

8.2 Python program control / automatic start

The loaded Python program can be controlled via the website:



MicroPython Script Engine

Start / stop / interrupt the currently loaded Python program

Script Startup

Store the Python program permanently in the drive and start it automatically after switching on the power supply:

- In the Script Startup / Autostart area, select -> **ON**
- Then save this setting and the Python program permanently with **Store Permanently**

8.3 Examples and documentation for the MovingCap commands

TIP: All documentation files (**.pyi files**) and example programs (**.py file**) can be found in the movingcap.de service portal, search for **CODE** there.

MICROPYTHON BASIC EXAMPLES FOR MOVINGCAP ETH

Example program	Explanation
driveObjectSyntaxTest.py	Demonstrates every single command of the mcdrive **module** in a functional test program for the MovingCap drive
driveSimplePosTest.py	A simple positioning application

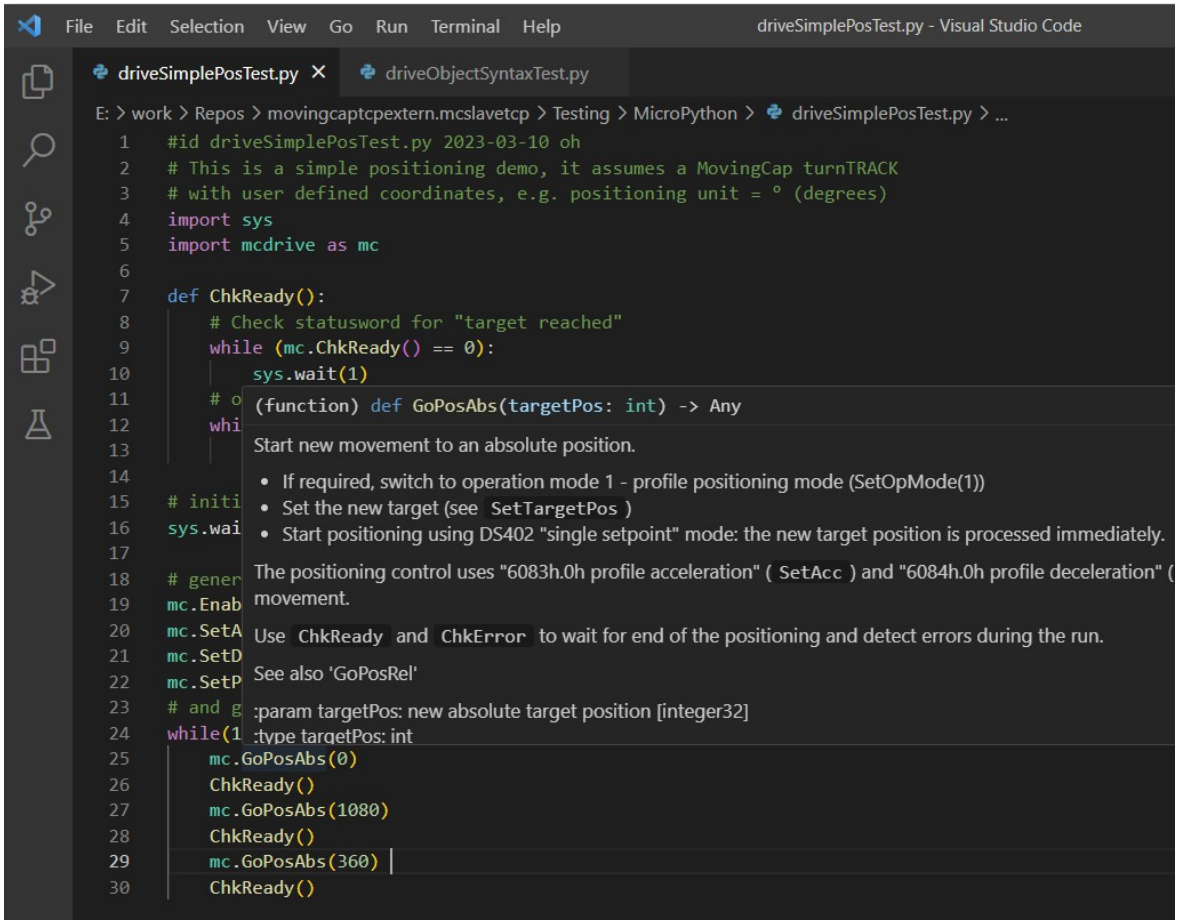
PYTHON MODULES FOR MOVINGCAP

Various Python modules are available in MovingCap CODE to use the MovingCap-specific drive and communication functions. These must be integrated at the start of your Python program using the **import command.**

Python module / import command	.pyi documentation file *1)	Explanation
import mcdrive.py or abbreviate the name as: import mcdrive.py as mc	mcdrive.pyi Alternatives in other languages: en/mcdrive.pyi fr/mcdrive.pyi it/mcdrive.pyi es/mcdrive.pyi	MovingCap drive functions for parameterizing and moving the drive, as well as writing and reading CiA 402 -objects. The associated .pyi file shows definitions and descriptions of all available commands. TIP: To obtain the documentation in a language other than English, copy the mcdrive.pyi version from the appropriate subdirectory.
import mccom	mccom.pyi	MovingCap communication functions for network. Currently offers a rudimentary option to extend the RefGo TCP port 10001 with additional/own commands. Planned extension: freely programmable TCP and UDP connections for application-specific protocols and functions. TIP: Examples in the movingcap.de service portal, search there for **refGoExtension**.
import refgo	refgo.pyi	Accepts RefGo commands in text form, as an alternative to the **mcdrive module**. Example: You can use **refgo.cmd("TS")** to query the drive status. TIP: Example in the movingcap.de service portal, search there for **UsingRefGoCommandsFromPython**.

*1) **.pyi** file (Python Interface Definition) to support program creation with an intelligent editor on the PC.

TIP: Use Visual Studio Code on your PC to edit the Python program. Copy the desired language variant of the **mcdrive.**pyi **file** into the same folder in which you create your **.**py **program.** Visual Studio Code (or another clever editor) recognizes the **import mcdrive** command in your program, evaluates the corresponding **mcdrive.pyi file** and shows you the corresponding documentation for each **mcdrive**... command from now on:



8.4 Debugging / finding errors

PYTHON OUTPUT AND ERROR MESSAGES VIA UDP PORT 14999

Python output and error messages can be received via a network connection on UDP port 14999.

In Windows, you can proceed as follows:

- Download and install Docklight Scripting. You do not need a license key for simple tests / displays.

- Go to the movingcap.de service portal, search for **Docklight_MovingCap_ETH_Debug_Output.zip,** download this file and unzip it**.**

- Start Docklight Scripting and open the project **Docklight_MovingCap_ETH_Debug_Output.ptp**

- Refer to the project documentation displayed in Docklight Scripting (from the file Docklight_MovingCap_ETH_Debug_Output.ptn):

MovingCap ETH Debug Interface via UDP Port 14999

** Click the "---> Connect" Send Sequence to open the connection and send an initial <CR> to the MovingCap.*

A typical initial response could look like this:

12.12.2022 12:55:54.950 [PC] - <CR>

12.12.2022 12:55:54.955 [mc] - SYS_Initialize: The MPFS2 File System is mounted<CR><LF>
TCP/IP Stack: Initialization Ended - success <LF><CR>
Host MOVINGCAP TCP - NBNS disabled<CR><LF>
IP address: 192.168.2.150 <CR><LF>
REFGO - opening server<CR><LF>
code: starting upload...<CR><LF>
code: upload complete, script id: maxTRACK N23 demo rev8<CR><LF>
<CR>
><CR><LF>

From this point on, you can see both Python print() output, but also Python errors in the Docklight communication window:

12.12.2022 12:57:11.074 [mc] - code: compile, script id: maxTRACK N23 demo rev8<LF><CR>
code: execute, script id: maxTRACK N23 demo rev8<LF><CR> Hello World from MovingCap!<CR><LF>

Here is what MovingCap reports after uploading and starting a script that contains an obvious Python error - the amount of spaces in line 22 is incorrect, rep. this line is not properly indented. See the Internet for "Python Indentation" and why this is important:

12.12.2022 12:58:57.769 [mc] - code: starting, script id: maxTRACK N23 demo rev8<LF><CR>

12.12.2022 12:58:57.869 [mc] - code: compile, script id: maxTRACK N23 demo rev8<LF><CR>
Traceback (most recent call last):<CR><LF> File "<stdin>", line 22<CR><LF>
IndentationError: unexpected indent<CR><LF>

9 MovingCap RABBIT - High-speed positioning for linear actuators

A new position controller ("Rabbit Pos Control") is available for the MovingCap flatTRACK, FATtrack and shortTRACK series. This makes optimum use of the mechanical and electrical advantages of the linear direct drives and has been designed for applications with high dynamic and accuracy requirements.

OVERVIEW AND DIFFERENCES BETWEEN THE MOVINGCAP POSITIONER SYSTEMS

Position controller	**MovingCap controller Standard**	**High-speed MovingCap controller ("Rabbit Pos Control")**
Product variant	MovingCap turnTRACK Ethernet TCP/IP (rotary actuators MC349 ETH, MC632/634/636 ETH, MCN23 ETH) MovingCap shortTRACK SM Ethernet TCP/IP (short stroke cylinder shortTRACK 45S100 - 115S240)	MovingCap flatTRACK Ethernet TCP/IP (Linear direct drives flatTRACK 100-650, FATtrack 200) MovingCap shortTRACK Ethernet TCP/IP (Compact linear direct drive shortTRACK 046)
Controller architecture	3-stage cascade control from PI controllers 1. Position 2. Speed 3. Current/torque	2-stage cascade control 1. State controller for position and speed + switching integrator for static position error correction 2. PI current regulator
Operating modes	Positioning mode (Profile Position Mode) Velocity mode (speed mode) Homing mode (homing mode)	Positioning mode (Profile Position Mode) *(speed operation in preparation)*
Controller input variables	Position or speed	Position, speed and acceleration
Trajectory generator	Trapezoidal speed profile	S-shaped speed profile

Trajectory parameters	Maximum speed (profile velocity), Acceleration (Profile Acceleration), Braking value (Profile Deceleration)	Maximum speed (profile velocity), Maximum acceleration (Profile Acceleration, same as braking value), jerk / acceleration change (Profile Jerk)
Controller setting	Default settings for stable operation for most basic applications Position control setting parameters: Proportional gain (Kp Pos) Integral gain (Ki Pos) Speed control setting parameters: Proportional gain (Kp Vel) Integral gain (Ki Vel)	Specification of the moving payload in grams [g] Adjustable weight compensation / compensation of additional static forces Fine tuning: Switchable linear or root position control Adjustable compensation of motor phase asymmetry ("Torque Ripple Suppression") Fine tuning by adjusting the control gains (Pos Gain, Vel Gain, Err Gain)

9.1 MovingCap RABBIT - Operating modes and settings

The current software version of the MovingCap Rabbit high-speed controller supports positioning mode 28 and the associated CiA 402 operating states 26. The differences to the MovingCap standard controller are as follows:

- Positioning mode only 28. (Speed mode 34 is currently not available).
- Only setting the zero position via REF35 is supported as a referencing method 18.
- New parameter **60A4h.0h profile jerk** for the permissible acceleration change / jerk.

  NOTE: There is not yet a specific command for the jerk setting in the Python mcdrive module 71 module. For example, use **mc.WriteObject(0x60A4h, 0h, 1000000)** to set the jerk value to 100000000.

NOTE: The jerk setting cannot yet be parameterized via IO input function 4 Timed Motion Absolute 57 or input function 6 Timed Motion Relative.

- The parameter **6084h.0h profile deceleration** has no influence. (The speed profile in the braking phase is symmetrical to the acceleration phase).

- Quick stop option code 2 is always executed as quick stop is always executed - "Brake to zero speed with **6085h.0h quick stop deceleration**, then switch off."

MOVINGCAP RABBIT TRAJECTORY GENERATOR

MovingCap RABBIT uses the following travel parameters for the travel profiles with point-to-point positioning:

Index	Subindex	Type	Access	Name [units]	Description
607Ah	0h	integer 32	R/W	target position [position units 18 / user defined units]	Target position
6081h	0h	unsigned32	R/W	profile velocity [position units 18 /s]	Maximum speed
6083h	0h	unsigned32	R/W	profile acceleration [1000x Position units 18 /s²]	Maximum acceleration
60A4h	0h	unsigned32	R/W	profile jerk*1) [1000x Position units 18 /s³]	Change in acceleration / jerk Recommended starting value is e.g. 50 times greater than the acceleration - this means that full acceleration is achieved within 20 milliseconds.

The following boundary conditions are taken into account when calculating the trajectory:

- To maintain the specified acceleration limit, the maximum speed is reduced if the travel distance is too short.

- In order to reach the specified acceleration limit, a low jerk value may be increased to such an extent that at least a pure S-curve (without a constant acceleration phase) is possible.

MOVINGCAP RABBIT CONTROL PARAMETERS

NOTE: Stable default settings for standard applications are provided ex works for the control parameters, but at least parameter **3101h.04h mass_payload** must be set before operation.

NOTE: If you perform a **factory reset** (e.g. via the **Parameters** website), the factory default settings are restored and the payload is reset to 0.

Index	Subindex	Type	Access	Name [units]	Description
3301h	**04h**	**unsigned16**	**R/W**	**mass_payload [g]**	**Additional user/payload weight [g] Additional weight/payload moved with the motorized sled in grams** This is the only parameter that needs to be set appropriately for each application. The set value does not have to be exact and can also be set higher, for example, to increase the stiffness of the static position control in a simple way.
3301h	01h	unsigned16	R/W	control_mode	Controller Mode / Operating mode of the status controller: 1 - Rabbit - linear position control, max. speed 2 - Hedgehog / hedgehog - square root position control near the end position, maximum end position accuracy. Reduction/prevention of overshoot during operation with high loads/additional forces / Friction. NOTE: Unstable for shortTRACK drives without additional mass. 3 - Float - tuning mode, only gravity and asymmetry correction is active.

3301h	0Eh	integer 16	R/W	deceleration_tuning	Deceleration/brake tuning [0.1 %], e.g. negative values, e.g. "-300" - use a 30% smaller braking value specification to correct overshooting of the dynamic movement. positive values, e.g. "300" - use 30% more braking force to correct insufficient dynamic movement.
3301h	0Fh	integer 16	R/W	weight_compensation 1*) [g] or +1/-1	Force pilot control for Installation position/weight compensation: +1 - Gravity pulls the motorized sled and the payload in a positive direction. -1 - Gravity pulls in a negative direction any other numerical value: compensation of a force corresponding to the weight of x grams.
3301h	08h	integer 16	R/W	k2 err gain	Controller gain for the integrated (summed) remaining position error in the end position larger values - higher rigidity and accuracy, but less stable / increasing tendency to vibrate
3301h	06h	integer 16	R/W	k0 pos gain	Controller gain for the position difference (linear or root function, depending on operating mode 3301h.01h and size of the position difference)
3301h	07h	integer 16	R/W	k1 vel gain	Controller gain for the speed difference larger values - correct overshoot / eliminate low-frequency shaking/oscillation in the end position

					Lower values - eliminate faster resonance/humming in the end position

1*) To set 3301h.0Fh weight_compensation manually, set **3301h.01h control mode = 3 float**, switch on positioning mode and start a positioning run. The drive remains in 'float' mode, but from now on the **3301h.0Fh weight_compensation** values are converted directly as a force. Use a simple digital luggage scale, for example, to confirm the realized compensation force in the order of magnitude. The force/weight values only correspond approximately to the force actually realized on the motor and are not suitable for measuring applications.

NOTE: Additional calibration/adjustment options are planned as an extension in order to be able to realize a more precise correlation of current and force values.

MOVINGCAP RABBIT MOTOR PARAMETERS

Motor parameters are usually set once during drive production. They are retained even after a **factory reset.**

Index	Subindex	Type	Access	Name [units]	Description
3300h	07h	integer 16	R/W	motor_weight 1*) [g]	The basic weight of the mobile snowmobile
3300h	04h	unsigned16	R/W	motor_imbalance amplitude 2*) [0.1% of rated current]	Correction of phase asymmetry / "Torque ripple suppression" Amplitude/height of the unbalance in 0.1% of the rated current. I.e. 10 = 1% of the rated current. Corresponds to the scaling of the current/force limitation 38 .
3300h	05h	integer 16	R/W	motor_imbalance_angle 2*) [°]	The mechanical angle of the motor in the range (-180°, +180°) relative to the single-phase point at which the minimum motor current due to the asymmetry occurs.

1*) As with the **3301h.04h mass_payload** setting, the value does not have to be exact. The controller always takes into account the sum of **3300h.07h motor_weight** and **3301h.04h mass_payload**. The following setting is recommended for flatTRACK and FATtrack applications where the motor carriage is fixed and the travel path is movable: Retention of the original

motor_weight value, and setting **3301h.04h mass_payload** = total weight of the moving superstructure incl. MovingCap track, minus **motor_weight**.

2*) The correction values are used for additional operating optimization with the flatTRACK and FATtrack motor variants. The controller also works with the basic setting **3300h.04h motor_imbalance_amplitude = 0**. However, the influence of the "torque ripple" can be noticeable during fast movements and the end position accuracy. To determine the appropriate settings manually, use **3301h.01h control mode = 3 float** and switch on positioning mode. If possible and taking safety precautions, move the motorized carriage back and forth by hand and look for a repetitive irregularity/pulling force that pulls the carriage slightly into certain positions. Then experiment with a starting value, e.g. **3300h.05h motor_imbalance_amplitude = 20** and different **3300h.05h motor_imbalance_angle** settings, until the irregularity disappears completely. MovingCap CODE - Python programs are currently under development for the automatic execution of calibration runs and automatic determination of the setting values. 67 programs are currently under development.

ADDITIONAL PARAMETERS FOR SPECIAL APPLICATIONS

Other special parameters and filters can be set for specific applications. Either via a corresponding setup project for Kickdrive, a Python configuration script, or by OW commands via the web interface 23 or terminal/TCP connection 24. Please contact our support if necessary.

COMPLETE BASIC INITIALIZATION, E.G. AFTER REPAIR/CONVERSION

For the complete re-initialization of MovingCap linear drives (flatTRACK, shortTRACK, FATtrack), documented setup projects for use with Kickdrive are available on our movingcap.de service portal, subdirectory **Init-Projects.** These projects can also be used to re-phase / determine the motor phase position relative to the absolute measuring system.

WARNING! These setup projects are also used to rewrite settings for motor overload protection. Only carry out the basic setup after consulting our support team.