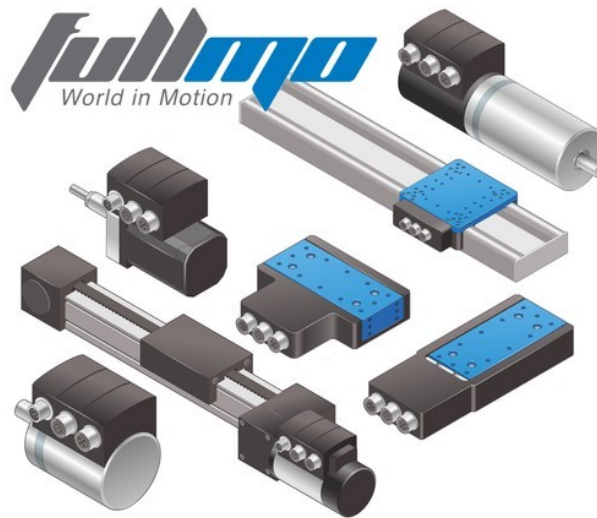




MovingCap Ethernet Antriebe Kurzanleitung

www.fullmo.de

Inhaltsverzeichnis

1 Anwendungsbereich	5
2 Anschlüsse	5
1 MC349 ETH Anschlussbelegung	6
2 MCN23 ETH Anschlussbelegung	8
3 MCFLAT / MCSHORT ETH Anschlussbelegung	10
4 MC6xx ETH Anschlussbelegung	12
3 Anzeigen	14
4 Kurzübersicht Bedienung	15
1 Netzwerkverbindung	15
2 MovingCap WEB - Bedienung über Webbrowser	17
Einfache Funktionsprüfung	17
Positionseinheiten / Skalierung / Nulllage festlegen	18
Dauerhaftes Speichern von Einstellungen	19
3 MovingCap SLCAN - CANopen Cia402 mit Kickdrive	20
4 MovingCap REFGO	21
Befehlsliste	21
RefGo-Beispiel über Weboberfläche	25
RefGo-Beispiel über Terminal	25
5 Betriebsarten und Einstellungen	27
1 CiA 402 Betriebszustand / Controlword	28
2 CiA 402 Betriebsarten / Modes of operation	30
3 Positionierbetrieb / Profile Position Mode	30
4 Drehzahlbetrieb / Velocity Mode	35
5 Referenzierbetrieb / Homing Mode	37
6 Strom-/Drehmoment-/Kraftbegrenzung	39
6 Antriebszustand und Fehlermeldungen	43
1 Statusanzeigen auf der Weboberfläche	43
2 Antriebszustand / CiA 402 Statusword 6041h	44
3 MovingCap Statusregister 1002h	47
4 CiA 402 Fehlermeldungen / Emergency Messages	50
5 CiA 402 Fehlercodes / History	51
6 Istwerte - Position, Geschwindigkeit, Strom, Temperatur	52
7 MovingCap IO	53
1 Eingangsfunktionen	55
Parameterobjekte Eingangsfunktionen 3511h - 351Ah	56
Funktion 1 - Capture Position	57

Funktion 2 - Capture Position	57
Funktion 3 - Counter	58
Funktion 4 - Timed Motion Absolute	59
Funktion 5 - Quick Stop	61
Funktion 6 - Timed Motion Relative	62
Funktion 7 - Set Home / Referencing	63
Funktion 8 - Teach Target	64
Funktion 9 - Software Enable	64
Eingänge simulieren	65
2 Ausgangsfunktionen	66
Parameterobjekte Ausgangsfunktionen 3611h - 3614h	66
Funktion 1 - Statusword 6041h Monitor	67
Funktion 2 - Error Register 1002h Monitor	67
Funktion 3 - Limitüberwachung rechts/links	68
Funktion 4 - Rückmeldung zur Positionierung	69
Funktion 5 - Antrieb ohne Fehler/Warnung	70
Funktion 6 - Rückmeldung Position und Status	70
8 MovingCap CODE - Python	
Programmierung	70
1 Ein einfaches Testprogramm	71
2 Python-Programmsteuerung / Automatischer Start	74
3 Python Benutzerparameter-Objekte	75
4 Beispiele und Dokumentation zu den MovingCap-Befehlen	76
5 Debugging / Fehler finden	79
9 MovingCap RABBIT -	
Highspeed-Positionierung für	
Linearantriebe	81
1 MovingCap RABBIT - Betriebsarten und Einstellungen	82
10 CiA 402 - Allgemeine Übersicht	87

1 Anwendungsbereich

Übersicht Softwarefunktionen und Bedienhinweise zu MovingCap-Kompaktantrieben Ethernet/TCP/IP 12 Vdc, 24Vdc und 48V mit integrierter Steuerung.

Z.B.

- 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 pushTRACK Ethernet TCP/IP (pushTRACK 45S100 - 115S240)



WARNING! Bitte beachten Sie die zu Ihrem Produkt gehörende Fullmo Montageanleitung und Fullmo Einbauerklärung! Für alle in diesem Dokument beschriebenen Bedienvorgänge müssen die Voraussetzungen zu bestimmungsgemäßigem Betrieb gemäß Fullmo Montageanleitung und Fullmo Einbauerklärung vorab hergestellt werden. Der Betrieb ohne Beachtung dieser Voraussetzungen ist untersagt.

HINWEIS: Bitte beachten Sie darüber hinaus spezifische Dokumentationen zu der jeweiligen Antriebsvariante, insbesondere bezüglich Leistungsdaten, Anschlussbelegungen, Parametrierungen, Sonderfunktionen und Optionen wie Bremse.

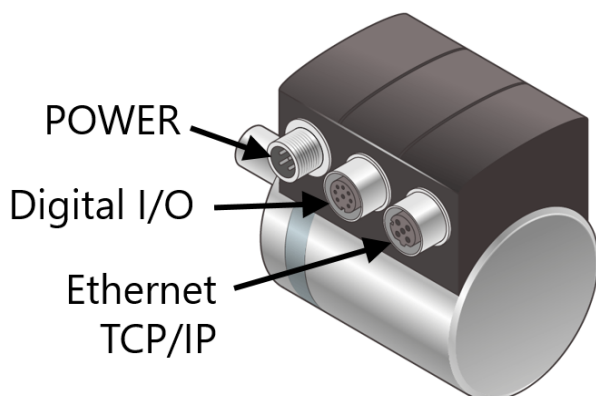
Softwareversion Antriebe: v50.00.08.xx / v50.01.08.xx / v53.02.00.xx / v54.01.08.xx -
Dokumentversion 2024-04-30 Entwurf

2 Anschlüsse

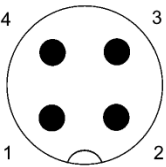
MovingCap Ethernet TCP/IP (ETH) Antriebe besitzen standardmäßig drei M12-Anschlüsse mit unterschiedlicher Codierung und Anschlussbelegung. Nachfolgend finden Sie eine Übersicht über die zu verwendenden Anschlussbelegungen und -kabel.

2.1 MC349 ETH Anschlussbelegung

Übersicht Anschlussbelegung zu MovingCap turnTRACK 349 Ethernet Antriebsvarianten

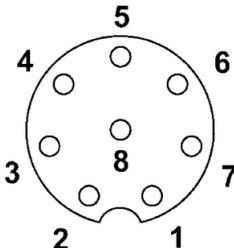


POWER

 MC349 ETH (MovingCap turnTRACK 349 Ethernet) POWER M12 Stecker, 4-polig, Kodierung: A			
Pin	Bezeichnung	Beschreibung	Farbe *1)
1	U_PWR	Leistungsversorgung 24-48 Vdc	braun
2	GND	GND Logik + Power	weiß
3	U_LOGIK	Logikversorgung 24 Vdc	blau
4	(optional HW_EN)	reserviert	schwarz

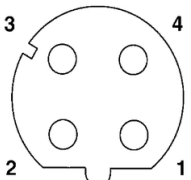
*1) Farben der Einzeladern bei Anschlusskabel Phoenix Contact SAC-4P-M12FS Sensor-/Aktorkabel

DIGITAL I/O

 MC349 ETH (MovingCap turnTRACK 349 Ethernet) Digital I/O M12 Buchse, 8-polig, Kodierung: A			
Pin	Bezeichnung	Beschreibung	Farbe *1)
1	COM U_LOGIK	Ausgang, stellt Logikversorgung U_LOGIK bereit	weiß
2	IN1	Digitaleingang 24 Vdc	braun
3	IN2	Digitaleingang 24 Vdc	grün
4	IN3	Digitaleingang 24 Vdc	gelb
5	IN4	Digitaleingang 24 Vdc	grau
6	OUT1	Digitalausgang 24 Vdc	pink
7	OUT2	Digitalausgang 24 Vdc	blau
8	GND	GND Logik	rot

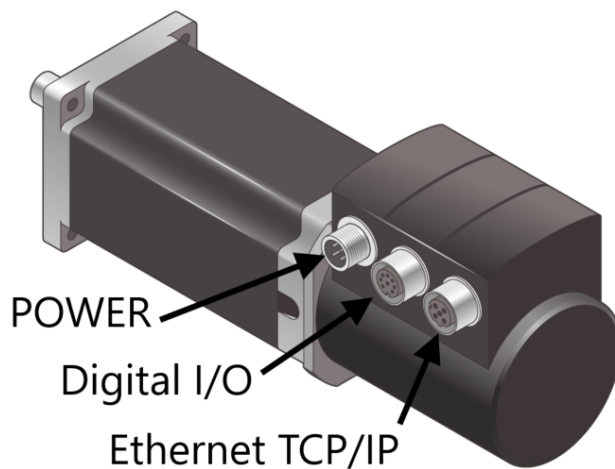
*1) Farben der Einzeladern bei Anschlusskabel Phoenix Contact SAC-8P-M12MS
Sensor/Aktorkabel M12 8-polig

ETHERNET TCP/IP

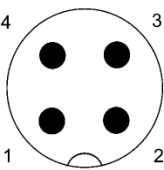
 MC349 ETH (MovingCap turnTRACK 349 Ethernet) Ethernet TCP/IP M12 Buchse, 4-polig, Kodierung: D Daten	
Anschluss für z.B. Phoenix Contact VS-MSD-IP20-93E Netzwirkabel auf RJ45	

2.2 MCN23 ETH Anschlussbelegung

Übersicht Anschlussbelegung zu MovingCap turnTRACK N23 Stepper Ethernet Antriebsvarianten

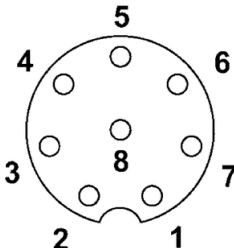


POWER

 MCN23 ETH (MovingCap turnTRACK N23 Stepper Ethernet) POWER M12 Stecker, 4-polig, Kodierung: A			
Pin	Bezeichnung	Beschreibung	Farbe *1)
1	U_PWR	Leistungsversorgung 24-48 Vdc	braun
2	GND	GND Logik + Power	weiß
3	U_LOGIK	Logikversorgung 24 Vdc	blau
4	(optional HW_EN)	reserviert	schwarz

*1) Farben der Einzeladern bei Anschlusskabel Phoenix Contact SAC-4P-M12FS Sensor-/Aktorkabel

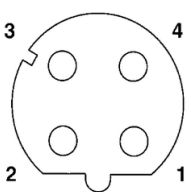
DIGITAL I/O

 MCN23 ETH (MovingCap turnTRACK N23 Stepper Ethernet) Digital I/O M12 Buchse, 8-polig, Kodierung: A			
Pin	Bezeichnung	Beschreibung	Farbe *1)
1	IN7 *2)	Digitaleingang 24 Vdc *2)	weiß
2	IO1	Digitalein/-ausgang 24 Vdc	braun
3	IO2	Digitalein/-ausgang 24 Vdc	grün
4	IO3	Digitalein/-ausgang 24 Vdc	gelb
5	IO4	Digitalein/-ausgang 24 Vdc	grau
6	IN5	Digitaleingang 24 Vdc	pink
7	IN6	Digitaleingang 24 Vdc	blau
8	GND	GND Logik	rot

*1) Farben der Einzeladern bei Anschlusskabel Phoenix Contact SAC-8P-M12MS
Sensor/Aktorkabel M12 8-polig

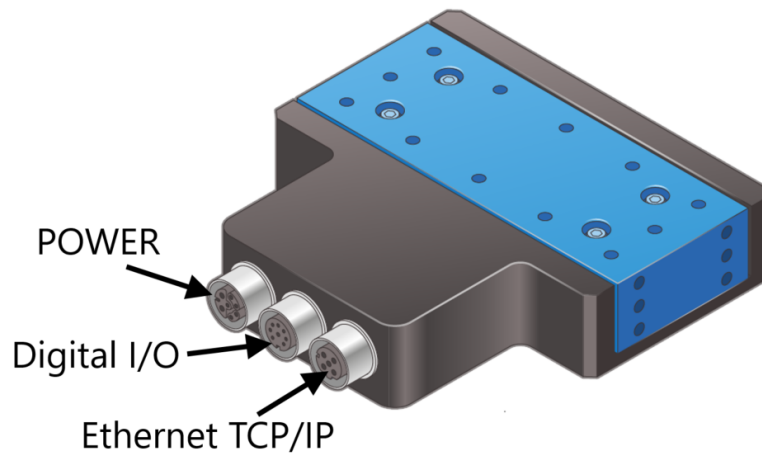
*2) Optional/Auf Anfrage möglich: COM U_LOGIK Ausgang, stellt Logikversorgung U_LOGIK bereit,
wie [MCFLAT](#)¹⁰⁾ und [MC6xx](#)¹²⁾

ETHERNET TCP/IP

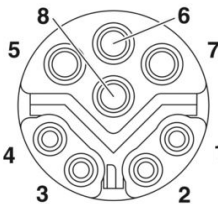
 MCN23 ETH (MovingCap turnTRACK N23 Stepper Ethernet) Ethernet TCP/IP M12 Buchse, 4-polig, Kodierung: D Daten Anschluss für z.B. Phoenix Contact VS-MSD-IP20-93E Netzkabel auf RJ45	
--	--

2.3 MCFLAT / MCSHORT ETH Anschlussbelegung

Übersicht Anschlussbelegung zu MovingCap flatTRACK Ethernet Antriebsvarianten (flatTRACK 100-650, FATrack 200, shortTRACK 046)

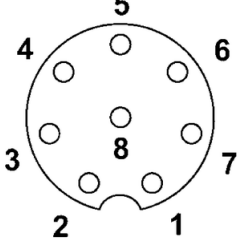


HYBRID POWER + DIGITAL I/O

 MCFLAT ETH (MovingCap flatTRACK Ethernet) MCSHORT ETH (MovingCap shortTRACK Ethernet) POWER + I/O M12 Buchse Hybrid, 8-polig, Kodierung: Y			
Pin	Bezeichnung	Beschreibung	Farbe *1)
1	IN10 (opt. HW_EN)	Digitaleingang 24 Vdc	weiß/orange
2	IO1	Digitalein/-ausgang 24 Vdc	orange
3	IN8	Digitaleingang 24 Vdc	weiß/grün
4	IO2	Digitalein/-ausgang 24 Vdc	grün
5	U_PWR	Leistungsversorgung 24-48 Vdc	blau
6	GND	GND Logik + Power	weiß
7	U_LOGIK	Logikversorgung 24 Vdc	braun
8	IN7	Digitaleingang 24 Vdc	schwarz

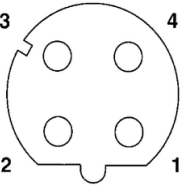
*1) Farben der Einzeladern bei Anschlusskabel Phoenix Contact NBC-M12MSY Hybridkabel M12 8-polig, Y-Kodierung

DIGITAL I/O

 MCFLAT ETH (MovingCap flatTRACK Ethernet) MCSHORT ETH (MovingCap shortTRACK Ethernet) Digital I/O M12 Buchse, 8-polig, Kodierung: A			
Pin	Bezeichnung	Beschreibung	Farbe *2)
1	COM U_LOGIK	Ausgang, stellt Logikversorgung U_LOGIK bereit	weiß
2	IO3	Digitalein/-ausgang 24 Vdc	braun
3	IO4	Digitalein/-ausgang 24 Vdc	grün
4	IO1 (optional)	Digitalein/-ausgang 24 Vdc	gelb
5	IO2 (optional)	Digitalein/-ausgang 24 Vdc	grau
6	IN5	Digitaleingang 24 Vdc	pink
7	IN6	Digitaleingang 24 Vdc	blau
8	GND	GND Logik	rot

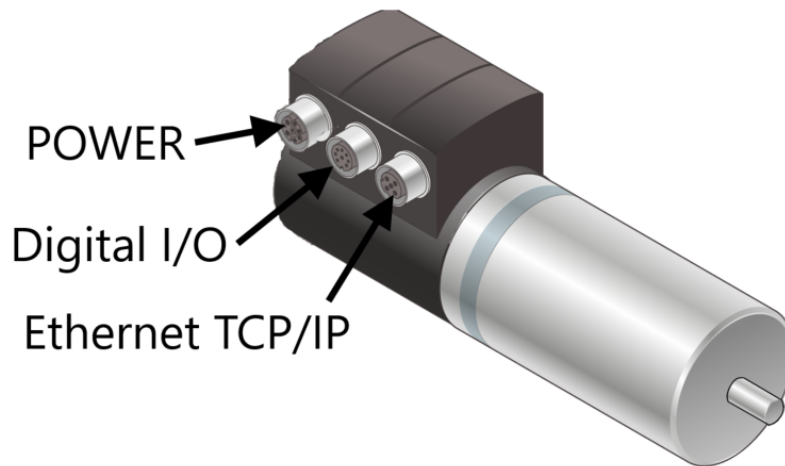
*2) Farben der Einzeladern bei Anschlusskabel Phoenix Contact SAC-8P-M12MS Sensor/Aktorkabel M12 8-polig

ETHERNET TCP/IP

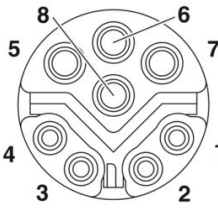
 MCFLAT ETH (MovingCap flatTRACK Ethernet) MCSHORT ETH (MovingCap shortTRACK Ethernet) Ethernet TCP/IP M12 Buchse, 4-polig, Kodierung: D Daten	
Anschluss für z.B. Phoenix Contact VS-MSD-IP20-93E Netzkabel auf RJ45	

2.4 MC6xx ETH Anschlussbelegung

Übersicht Anschlussbelegung zu MovingCap turnTRACK 6xx Ethernet Antriebsvarianten (MC632 ETH / MC634 ETH / MC636 ETH)

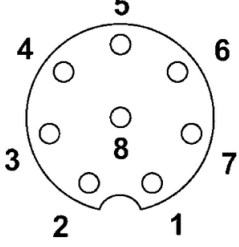


HYBRID POWER + DIGITAL I/O

 MC632 ETH / MC634 ETH / MC636 ETH (MovingCap turnTRACK 6xx Ethernet) POWER + I/O M12 Buchse Hybrid, 8-polig, Kodierung: Y			
Pin	Bezeichnung	Beschreibung	Farbe *1)
1	IN10 (opt. HW_EN)	Digitaleingang 24 Vdc	weiß/orange
2	IO1	Digitalein/-ausgang 24 Vdc	orange
3	IN8	Digitaleingang 24 Vdc	weiß/grün
4	IO2	Digitalein/-ausgang 24 Vdc	grün
5	U_PWR	Leistungsversorgung 24-48 Vdc	blau
6	GND	GND Logik + Power	weiß
7	U_LOGIK	Logikversorgung 24 Vdc	braun
8	IN7	Digitaleingang 24 Vdc	schwarz

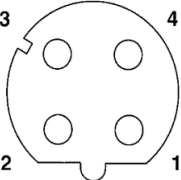
*1) Farben der Einzeladern bei Anschlusskabel Phoenix Contact NBC-M12MSY Hybridkabel M12 8-polig, Y-Kodierung

DIGITAL I/O

 MC632 ETH / MC634 ETH / MC636 ETH (MovingCap turnTRACK 6xx Ethernet) Digital I/O M12 Buchse, 8-polig, Kodierung: A			
Pin	Bezeichnung	Beschreibung	Farbe *2)
1	COM U_LOGIK	Ausgang, stellt Logikversorgung U_LOGIK bereit	weiß
2	IO3	Digitalein/-ausgang 24 Vdc	braun
3	IO4	Digitalein/-ausgang 24 Vdc	grün
4	IO1 (optional)	Digitalein/-ausgang 24 Vdc	gelb
5	IO2 (optional)	Digitalein/-ausgang 24 Vdc	grau
6	IN5	Digitaleingang 24 Vdc	pink
7	IN6	Digitaleingang 24 Vdc	blau
8	GND	GND Logik	rot

*2) Farben der Einzeladern bei Anschlusskabel Phoenix Contact SAC-8P-M12MS Sensor/Aktorkabel M12 8-polig

ETHERNET TCP/IP

 MC632 ETH / MC634 ETH / MC636 ETH (MovingCap turnTRACK 6xx Ethernet) Ethernet TCP/IP M12 Buchse, 4-polig, Kodierung: D Daten	
Anschluss für z.B. Phoenix Contact VS-MSD-IP20-93E Netzwerkkabel auf RJ45	

3 Anzeigen

MovingCap Ethernet TCP/IP (ETH) Antriebe besitzen standardmäßig zwei LED-Anzeigen

LED-Farbe	Bezeichnung	Beschreibung
gelb	LINK/ACT	Verbindungszustand und Aktivität des Netzwerkanschlusses
grün	STATUS	MovingCap IP-Adresse und Status über Morse-Code

MORSE-CODE DEKODIEREN / IP-ADRESSE ERMITTELN

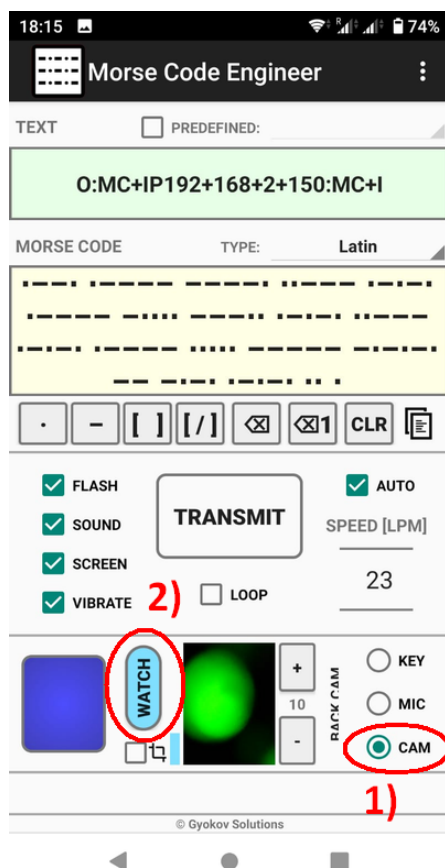
Die grüne Status-LED überträgt u.a. fortlaufend die aktuell eingestellte IP-Adresse und kann wie folgt ausgewertet werden:

- Installieren Sie auf einem Smartphone eine Morse-Code-Decoder App wie "Morse Code Encoder & Decoder" (Noldy Labs) oder "Morse Code Engineer" (GyokovSolutions)
- Starten Sie die Morse-App und filmen Sie mit der Kamera die grüne LED so nahe wie möglich und mit wenig Störlicht ab. Halten Sie das Smartphone möglichst ruhig.
- Warten Sie mindestens 2 Minuten, oder bis Sie zwei aufeinanderfolgende identische Texte dekodiert haben, um sicherzustellen, dass die Dekodierung korrekt war.

Die MovingCap-Morse-Nachricht bei IP-Voreinstellung lautet z.B:

MC+IP192+168+2+150:

(Das Plus-Zeichen ist als Morse-Code deutlich kürzer als ein Dezimalpunkt und wird daher als Trennzeichen verwendet.)



HINWEIS: Falls eine Morse-App bei Ihnen nicht möglich ist, können Sie auch die grüne LED mindestens 2 Minuten abfilmen und uns das Video zur Dekodierung bereitstellen. Unsere Kontaktinformationen finden Sie bei www.fullmo.de

4 Kurzübersicht Bedienung



WARNUNG! Bitte beachten Sie die zu Ihrem Produkt gehörende Fullmo Montageanleitung und Fullmo Einbauerklärung! Für alle in diesem Dokument beschriebenen Bedienvorgänge müssen die Voraussetzungen zu bestimmungsgemäßem Betrieb gemäß Fullmo Montageanleitung und Fullmo Einbauerklärung vorab hergestellt werden. Der Betrieb ohne Beachtung dieser Voraussetzungen ist untersagt.

4.1 Netzwerkverbindung

Der Anschluss an einen PC / Netzwerkschwitch mit Standard RJ45-Netzwerkbuchse erfolgt über ein geeignetes Datenkabel für M12 D-Kodierung, z.B. Phoenix Contact VS-MSD-IP20-93E Netzwerkkabel auf RJ45.

STANDARDEINSTELLUNGEN MOVINGCAP TCP/IP

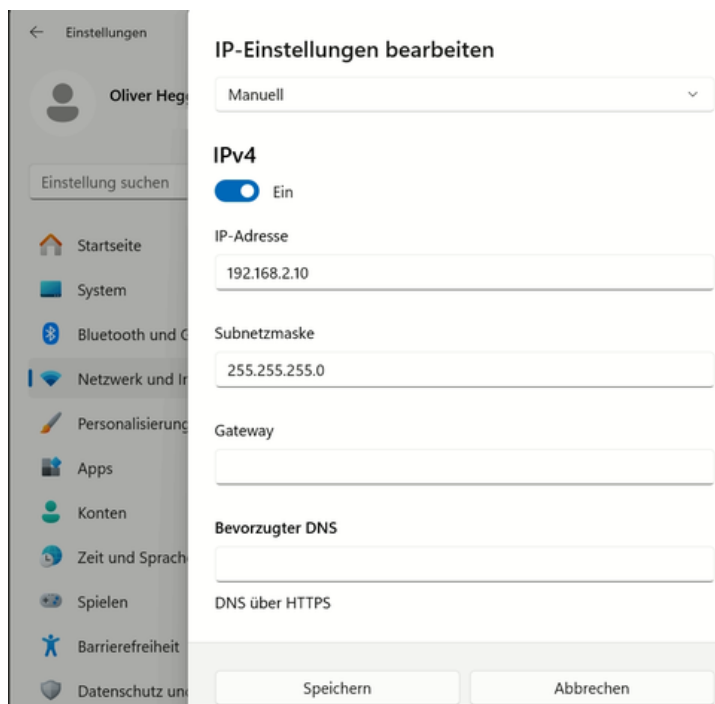
IP Adresse: **192.168.2.150**
Subnetzmaske: **255.255.255.0**
Gateway: -
DHCP: - nicht verfügbar –
DNS: - nicht verfügbar -

HINWEIS: Sollte Ihnen die im MovingCap eingestellte IP-Adresse nicht bekannt sein, können Sie die Adresse wie im Abschnitt [Anzeigen](#)¹⁴⁾ über Morse-Code ermitteln.

VORBEREITUNG DER NETZWERKVERBINDUNG

- Schließen Sie einen USB-nach-Ethernet Adapter an Ihren PC an, oder verwenden Sie den eingebauten RJ45 Ethernetanschluss.
- Ändern Sie die Verbindungseinstellungen bei diesem Anschluss auf ein IP4-Adresse im gleichen Subnetz (jedoch anderer Endnummer) wie die MovingCap-Einstellung, z.B.:
IP Adresse: **192.168.2.10**
Subnetzmaske: **255.255.255.0**
Gateway: -
DNS: -

In Windows 11 kann dies wie folgt aussehen:



4.2 MovingCap WEB - Bedienung über Webbrowser

Kommunikation über: Webbrowser, z.B. Chrome, Edge, Firefox, Opera.
Standard-HTML Verbindung über **TCP-Port 80**.

4.2.1 Einfache Funktionsprüfung

- Öffnen Sie die Weboberfläche durch Eingabe von 192.168.2.150 (bzw. der am MovingCap eingestellten IP-Adresse) in der Adresszeile.

TIPP: Wenn Sie die eingestellte IP-Adresse nicht (mehr) kennen, können Sie diese über die [LED-Anzeige](#) ¹⁴ herausfinden.

- Wählen Sie im Menü **Click & Run** die Funktion **Back & Forth**.
 - Der Antrieb startet einen oszillierenden Betrieb über die über **Target Distance (Move Rel)** festgelegte Strecke.

MC349 Eth
Click&Run
Servo
Command
Network



MC349 Click & Run

Power OFF
Power ON
Stop Motion
Move Absolute
Move Relative

Back & Forth

To (re)set the zero position, go to [Servo](#) and press the "Set Home (Pos=0)" button.
To change how positions and velocity are scaled, go to [Servo](#) and check the "Gear & Axis Configuration".

Positioning Velocity	720
Positioning Acceleration	5000
Positioning Deceleration	1000
Target Pos. (Move Absolute)	960
Target Distance (Move Rel)	1923

MC349	OK	Status	0027h	Oper.	1	Python	0
Live		Word		Mode			
Position	-5473 usr/	Velocity	726 usr/ 363	Temp. °C	-		
inc	-186806	rpm					

4.2.2 Positionseinheiten / Skalierung / Nulllage festlegen

- Wählen Sie das Menü **Servo** und stellen Sie die gewünschte Skalierung bei **Gear & Axis Configuration** wie folgt ein:

Getriebeübersetzung:

Gear In – Umdrehungen an der Motorwelle, z.B. **5** . (Oder **1** falls kein Getriebe vorhanden)

Gear Out – Umdrehungen an der Getriebe-Ausgangswelle, z.B. **1**

Skalierung/Auflösung festlegen:

Feed – Wenn der Motor eine Umdrehung am Getriebeausgang fährt - um wie viel erhöht sich der Positionszähler?

Beispielwerte für "Feed":

360 - wenn Sie am Getriebeausgang einen Drehteller montieren und Positionen in Winkelgrad vorgeben möchten.

100000 - wenn Sie eine Zahnriemenachse mit Antriebsrollenumfang $d=100$ mm einsetzen (z.B. max GmbH MZK 040), und in Mikrometer-Koordinaten arbeiten möchten. Ein [RefGo](#)²⁵-Fahrkommando **G100000** entspricht dann "Fahre auf Absolutposition 100.000 Mikrometer / 100mm", ein Fahrkommando G2500 entspricht "Fahre auf Absolutposition 2,5mm"

MC349 Eth Click&Run Servo Command Network

Deceleration (100 slow - 20000 fast) (*3) 1000

(*1) Target/Profile velocity values are scaled in rpm ("revolutions/minute", default). Or in your "user position units / second", if Gear/Feed settings are used (see below).

(*2) Position values are scaled in motor encoder increments (default). Or in your "user position units", if Gear/Feed settings are used (see below).

(*3) Acceleration/Deceleration values are scaled in $[0.01525 \text{ motor revolutions/s}^2]$.

Gear & Axis Configuration

Gear In revolutions (motor side, e.g. 5) 5

Gear Out revolutions (shaft side, e.g. 1) 1

Feed Constant (e.g. 100 [mm/rev] or 360 [°/rev]) 360

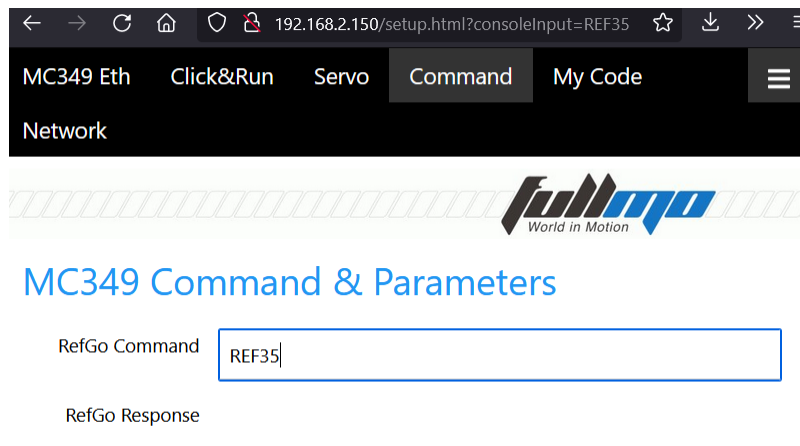
To make the Gear & Axis Configuration permanent, go to [Command](#) and use the button "Store Permanently."

- Antriebe mit Absolut-Wegmesssystem (z.B. MC349_..._AE):
Setzen Sie ggf. die Nulllage neu:

Wählen Sie das Menü Command, und geben Sie bei **RefGo Command** den folgenden Befehl ein:

REF35

("Referenziere mit der CiA 402 Homing-Methode 35 - Neusetzen der Nulllage")



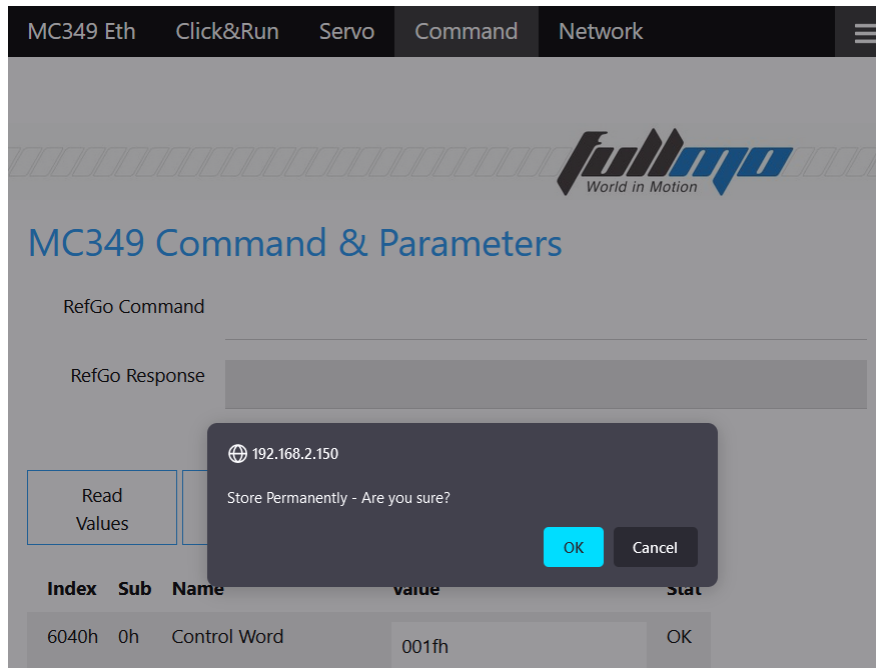
The screenshot shows a web browser window with the address bar displaying `192.168.2.150/setup.html?consoleInput=REF35`. The browser's navigation bar includes back, forward, refresh, and home icons. Below the address bar is a dark navigation menu with tabs labeled "MC349 Eth", "Click&Run", "Servo", "Command" (which is selected), and "My Code". A hamburger menu icon is on the right. Below the navigation menu is a light green banner with the "Fullmo" logo and the tagline "World in Motion". The main content area is titled "MC349 Command & Parameters" in blue. It contains a "RefGo Command" label next to a text input field containing "REF35". Below this is a "RefGo Response" label followed by a horizontal line.

4.2.3 Dauerhaftes Speichern von Einstellungen

Damit Parametereinstellungen auch nach Wegschalten / Wiedereinschalten der Spannungsversorgung erhalten bleiben, müssen Sie über ein gesondertes Kommando im nicht-flüchtigen Parameterspeicher hinterlegt werden.

Gehen Sie wie folgt vor:

- Wählen Sie im Menü **Parameters** die Funktion **Store Permanently**
- Bestätigen Sie die Sicherheitsfrage mit **OK**



4.3 MovingCap SLCAN - CANopen Cia402 mit Kickdrive

Kommunikation über: **TCP Port 15001 / CANopen Node-ID: 50**

Die Kommunikation erfolgt in einem Format ähnlich wie bei CAN-nach-TCP/IP Gateways, z.B. VSCOM NetCAN. Siehe auch "SLCAN ASCII protocol" (www.vscom.de/download/multiio/others/info/VSCAN_Manual.pdf , "ASCII Command Set") oder Python Modul "CAN over Serial / SLCAN" (<https://python-can.readthedocs.io/en/3.3.2/interfaces/slcan.html#>).

Über diese TCP-Verbindung kann MovingCap als CANopen Node mit der **Node-ID 50** wie ein [CiA 402 EtherCAT/CANopen](#) ⁸⁷ Feldbusantrieb parametrisiert und betrieben werden. Diese Betriebsart wird von fullmo Kickdrive und Kickdrive Zero unterstützt.

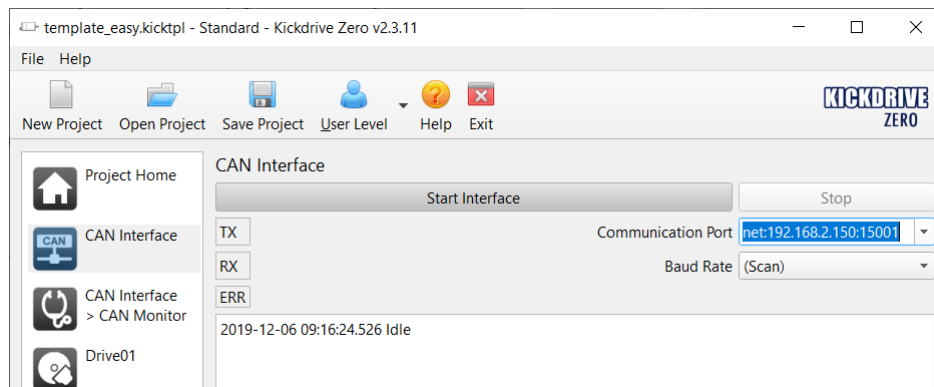
GRUNDINBETRIEBNAHME MOVINGCAP ETH ÜBER WINDOWS-SOFTWARE FULLMO KICKDRIVE ODER KICKDRIVE ZERO:

- Gehen Sie wie in Dokument **fullmoBasicSetupMovingCap349_de.pdf** vor (aus unserem [movingcap.de Service-Portal](http://movingcap.de/Service-Portal), Unterverzeichnis **turnTRACK_MC349_CANopen**), mit folgenden Unterschieden:
 - Der Datenanschluss ist nicht über CAN-Bus, sondern über eine [Ethernet-Netzwerkverbindung](#) ¹⁵.
 - Wählen Sie vor Start des Antriebs im geöffneten Kickdrive-Projekt beim Modul **CAN Interface** die Einstellung
Communication Port = net:192.168.2.150:15001

(bzw. die von Ihnen gewählte IP-Adresse für den Antrieb.)

Baud Rate = 125K

(bzw. beliebig, da für den TCP-Betrieb nicht relevant)



4.4 MovingCap REFGO

Kommunikation über: **TCP-Port 10001**

RefGo ist ein einfaches textbasiertes Protokoll (ASCII-Protokoll) zur Parametrierung und Betrieb des MovingCap.

Befehlsformat (am Beispiel für TP / „Tell Position“):

TP<CR>

Antwortformat:

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

Mit

<CR>= carriage return / ASCII Code dezimal 13 / hexadezimal 0D / kein sichtbares Zeichen!

<LF>= line feed / ASCII Code dezimal 10 / hexadezimal 0A / kein sichtbares Zeichen!

Das Promptzeichen > beendet die Antwort des MovingCap und zeigt an, dass MovingCap für das nächste Kommando bereit ist.

4.4.1 Befehlsliste

Textbefehl	Bezeichnung	Beschreibung und Beispiel	Anmerkungen
G	Go Direct Position	Einschalten des Positionierbetriebs ³⁰⁾ , falls erforderlich, und Fahrt zur angegebenen	*1)

		Absolutposition x: G10000	
GW	Go Way	Einschalten des Positionierbetriebs ^[30] , falls erforderlich,, und relative Bewegung um x: GW-5000	*1) *6)
PO	Set Target Position	Setze Zielposition absolut, um anschließend mit G die Fahrt zu starten P010000 G	*1)
WA	Set Way	Setze Zielposition relativ, um anschließend mit GW die Fahrt zu starten WA-5000 GW	*1)
SM	Stop Motion	Unterbreche die laufende Bewegung über Schnellstop / Quickstop ^[34]	*1)
PWC	Power Continue	Antrieb einschalten und Positionierbetrieb ^[30] aktivieren	*1)
CM	Continue Motion	Setzt eine mit SM Stop Motion unterbrochene Fahrt fort	*7)
REF	Referencing	Ausführen einer Referenzier/Homing ^[37] -Methode, z.B: Nullage setzen: REF35 oder REF37 Anschlagsreferenzierung pos. Richtung: REF-18 Anschlagsreferenzierung neg. Richtung: REF-19	*2)
PQ	Power Quit	Antrieb ausschalten	*1)
OW	CANopen object: write value	Schreibt ein einen CiA 402 ^[87] -Parameter wie z.B. in Positionierbetrieb ^[30] beschrieben RefGo Command: OW6067,0,100 RefGo Response: OW6067,0,100,OK	*3)
OR	CANopen object: read value	Liest einen CiA 402 ^[87] -Parameter wie z.B. in Positionierbetrieb ^[30] beschrieben RefGo Command: OR6067,0 RefGo Response: OR6067,0,100	*3)
TPSR	Tell Process Status Register	Antriebsstatus auslesen RefGo Command: TPSR RefGo Response: 00000106	*1) *4)
TS	Tell Status	Einfachen Antriebsstatus auslesen RefGo Command: TS RefGo Response: 2	*1) *5)
TP	Tell Actual Position	Istposition auslesen RefGo Command: TP RefGo Response: 342344	*1)
SP	Speed	Setze Sollgeschwindigkeit - Profile Velocity im Positionierbetrieb ^[30] , oder Target Velocity im Drehzahlbetrieb ^[35]	*1)

		SP500	
AC	Acceleration	Beschleunigung für Positionier- oder Drehzahlbetrieb vorgeben AC2000	*1) *8)
DE	Deceleration	Bremsrampe für Positionier- oder Drehzahlbetrieb vorgeben AC2000	*8)
JP	Jog Positive	Drehzahlbetrieb im positive Richtung SP500 JP	*1)
JN	Jog Negative	Drehzahlbetrieb im negative Richtung SP500 JN	*1)
RR	Repeat Reverse	Starte oszillierende Fahrt wie über Webseite Back&Forth, ausgehend von der Istposition RR-10000	*1)
RW	Repeat Way	Setze Fahrstrecke für Repeat Reverse, um anschließend mit RR die Fahrt zu starten RW2000 RR	*1)
WT	Wait Repeat	Setze die Wartezeit in den Endpositionen von Repeat Reverse WT5000 RR1000	*1)
SO	Set Output	Setze Digitalausgang x auf High S02	*1)
CO	Clear Output	Setze Digitalausgang x zurück auf Low C02	*1)
TI	Tell Input	Lies den Zustand von Digitaleingang x , oder alle Eingänge RefGo Command: TI3 RefGo Response: 1 RefGo Command: TI RefGo Response: 0005	*1)

Anmerkungen:

*1) Mit Einschränkungen kompatibel zu JennyScience AG XENAX[®] ASCII-Protokoll.

*2) REF ohne Parameter hat keine Funktion, da MovingCap Antriebe nicht zwingend referenziert werden müssen. Um die [Nulllage neu zu setzen](#)^[18] (Actual Position = 0) führen Sie **REF35** oder **REF37** aus. MovingCap unterstützen zusätzlich die herstellereigenspezifische Methode **REF-18** (Anschlagreferenzierung in positiver Richtung) und **REF-19** (neg. Richtung). Hierzu können mit **OW**-Befehle weitere Homing-Parameter gesetzt werden, vgl. [Referenzierbetrieb / Homing Mode](#)^[37]

*3) Schreiben und Lesen von Objektparametern, die gem. [CiA 402 Geräteprofil](#)^[87] und dem zum Antrieb passenden Objektverzeichnis (.xdd Datei) definiert sind. Beispiele:
OW3511,8,120 --> Schreibe in Objekt 3511h.8h den Wert 120 (Input Function 1 Target Position =

120)

OR3511,8 --> Auslesen von Objekt 3511h.8h. Antwort z.B. **OR3511,8,120**

OR6064,0 --> Auslesen von [CiA 402](#)^[87] Standardobjekt 6064h.0h Position Actual Value. Gleiches Resultat wie TP (Tell Position). Antwort z.B. **OR6064,0,13113**

OR6041,0 --> Auslesen von [CiA 402](#)^[87] Standardobjekt [6041h.0h Statusword](#)^[44]. Antwort z.B. **OR6041,0,64,1063**

*4) TPSR Rückgabewerte:

Die TPSR-Antwort ist ein 8-stelliger Hexadezimalwert, die folgende Bits unterstützt.

TPSR Process Status Value	zugehöriger Wert CiA 402 Statusword 6041h ^[44]
BIT0 (Hex 1) - ERROR	Fault = Bit 3
BIT1 – REFERENCE, immer gesetzt	
BIT2 (Hex 4) - IN MOTION	Moving = Bit 8
BIT3 (Hex 8) - IN POSITION	Target Reached = Bit 10
BIT8 (Hex 100) - INVERTER VOLTAGE	Voltage Enabled = Bit 4

Für MovingCap-Neuanwendungen ohne Kompatibilitätsanforderungen zu JennyScience AG XENAX[®] wird empfohlen, direkt das [CiA 402 Statusword 6041h](#)^[44] auszuwerten

*5) TS Rückgabewerte:

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

*6) GW Relativbewegungen:

Die ausgeführte Bewegung kann relativ zur letzten Zielposition (Voreinstellung) oder relativ zur aktuellen Position ausgeführt werden. Dies wird über den CiA 402-Parameter [60F2h.0h Position Option Code](#)^[33] festgelegt.

*7) CM Continue Motion:

Fortsetzen der Fahrt nach einem vorausgehenden **SM** Stop Motion Befehl.
 Positionierbetrieb: Setze die Fahrt zur zuletzt vorgegebenen Position fort.
 Drehzahlbetrieb: Setze die Fahrt mit der zuletzt vorgegebenen Drehzahl fort.

*8) AC Beschleunigungs- und DE Bremswerte / Acceleration / Deceleration:

Die Funktionsweise und Einheiten dieser Parameter können zwischen unterschiedlichen MovingCap-Antriebsvarianten variieren. Bitte beachten Sie die Details und Anmerkungen in [Positionierbetrieb](#)^[30] zu den Parametern **6083h.0h profile acceleration** und **6084h.0h profile deceleration**, sowie die Besonderheiten beim [MovingCap RABBIT Highspeed-Regler](#)^[81].

4.4.2 RefGo-Beispiel über Weboberfläche

- Wählen Sie die Seite / das Menü **Parameters** (lange Bezeichnung "Parameters & Commands")
- Geben Sie bei **RefGo Command** eines der Befehle aus der [Befehlsliste](#)^[21] ein, z.B.

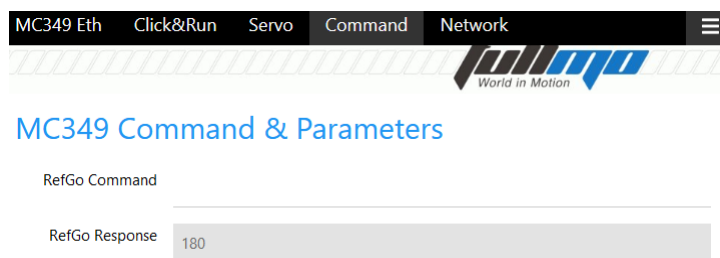
G180

und Bestätigen Sie mit der Eingabetaste

- Das Erreichen der Zielposition können Sie mit

TP

überprüfen - die Antwort sehen Sie in der Zeile **RefGo Response**



4.4.3 RefGo-Beispiel über Terminal

Docklight Scripting ist ein TCP-Terminal, das in einer eingeschränkten Version kostenlos genutzt werden kann und hier zum direkten Download zur Verfügung steht:

https://docklight.de/download/Docklight_Scripting.zip

Ein vorbereitetes Projekt mit den RefGo-Kommandos finden Sie auf movingcap.de, dort nach **Docklight** suchen.

Nachfolgend ein Docklight-Kommunikationsprotokoll mit einem MovingCap turnTRACK 349 Antrieb. Die [PC]-Zeilen sind die Textkommandos, die vom PC als TCP-Client an den MovingCap TCP

Serverport 10001 gesendet werden. Die [mc349]-Zeilen sind die Antworten des MovingCap-Antriebs.

21.10.2021 13:24:52.889 [PC] - REF35<CR> --> *Sende REF*
Kommando mit Terminal

21.10.2021 13:24:52.908 [mc349] - REF35<CR><LF>
 > --> *MovingCap bestätigt*

Befehl und schickt neues Prompt für „bereit für nächsten Befehl“

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] - DE1000<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>

21.10.2021 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> --> Auslesen von Objekt 3511h.08h, aktueller Wert ist 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> --> Schreibe in Objekt 3511h.08h den Wert 120

>

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> --> Erneutes Auslesen von 3511h.08h, aktueller Wert jetzt 120

>

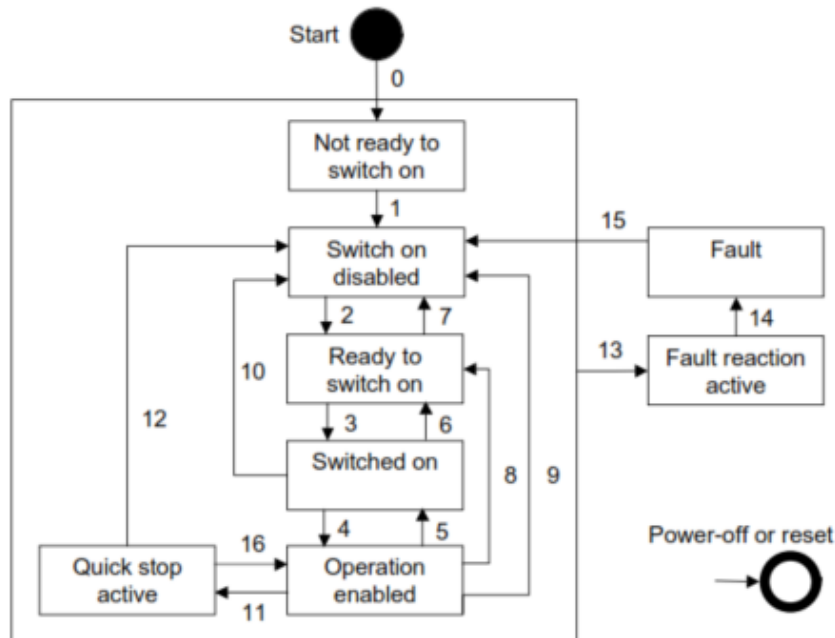
5 Betriebsarten und Einstellungen

MovingCap-Antriebe unterstützen in großem Umfang Betriebsarten, Funktionen und Objekte/Parameter gemäß dem **CiA 402 / DS402** Geräteprofil, sowie die **CiA 402 State Machine** (PDS FSA, Power Drive System - Finite State Automat).

CiA 402, alternativ auch als **DS 402** / Device Standard 402 bezeichnet, ist ein Geräteprofil für Antriebe und Bewegungssteuerung. Es definiert die Betriebsarten und Anwendungsdaten für Frequenzumrichter, Servoregler und -antriebe. Es ist Bestandteil des CANopen-Protokolls, kommt jedoch auch in EtherCAT und POWERLINK-Anwendungen zum Einsatz. Das CiA 402 Geräteprofil ist in der Norm **IEC61800-7-201** spezifiziert als Profil Typ 1 - **CiA402 Drive Profile for power drive systems (PDS)**.

5.1 CiA 402 Betriebszustand / Controlword

Die möglichen Betriebszustände und Zustandswechsel der MovingCap-Antriebe sind gem. der [CiA 402](#) ^[87] State Machine (PDS FSA, Power Drive System - Finite State Automat) definiert:



Die Betriebszustände werden über die folgenden Objekte/Parameter gesteuert:

Index	Subindex	Typ	Zugriff	Name	Beschreibung
6040h	0h	unsigned16	R/W	Controlword	Steuerung der CiA 402 ^[87] State Machine (PDS FSA). Schaltet den Antrieb in den gewünschten Betriebszustand. Siehe Beschreibung unten.
6041h	0h	unsigned16	RO	Statusword	Aktueller Betriebszustand. Siehe Detaillierte Beschreibung zum CiA 402 Statusword 6041h ^[44]

CONTROLWORD 6040H - BETRIEBSZUSTAND ÄNDERN

Bitnr. (0-15)	Bedeutung	Beschreibung
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	halt
9	oms	operation mode specific
10	r	reserved
11-15	ms	manufacturer specific

Gewünschtes Kommando / Änderung des Zustands	Controlword 6040h in Binärdarstellung (x = beliebiger Wert)	Mögliche Übergänge (Transitionen, Pfeile in Diagramm oben)
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
Quick Stop	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

Das Objekt **6060h.0h Modes of operation** wählt die gewünschte Betriebsart.

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
6060h	0h	integer8	R/W	Modes of operation	Betriebsmodus des Antriebs: 0 = aus / kein Modus ausgewählt 1 = Positionierbetrieb / Profile Position Mode 3 = Drehzahlbetrieb / Velocity Mode 6 = Referenzierbetrieb / Homing Mode

5.3 Positionierbetrieb / Profile Position Mode

CIA 402 STANDARDPARAMETER POSITIONIERBETRIEB

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
6060h	0h	integer 8	R/W	modes of operation	1 = Positionierbetrieb / Profile Position Mode
6040h	0h	unsigned 16	R/W	Controlword	Ansteuerung des Positionsbetriebs über CiA 402 Betriebszustände ²⁸ 15 (000Fh) = Switch on + Enable Operation. Bereit für Positionierung 31 (001Fh) = Nächste Zielposition wird übernommen. Jeder aktivierte Fahrt wird vollständig ausgeführt ("Set of Setpoints") 63 (003Fh) = Nächste Zielposition wird direkt

					angefahren. Bereits in Bearbeitung befindliche Fahrten werden abgebrochen / gelöscht.
607Ah	0h	integer 32	R/W	target position [Positionseinheiten ¹⁸ / user defined units] *1)	Zielposition
6083h	0h	unsigned d32	R/W	profile acceleration rotativ: feste Einheit [100 Umdrehungen/min ²] bzw. [0,0277 Umdrehungen/s ²] linear: [1000x Positionseinheiten ¹⁸ /s ²]	Beschleunigungswert Rotativantrieb: Immer bezogen auf Motorumdrehungen/Minute. Keine benutzerdefinierte Skalierung. Linearantrieb: Immer in Positionseinheiten ¹⁸ /s ² .
6084h	0h	unsigned d32	R/W	profile deceleration rotativ: [100 Umdrehungen/min ²] bzw. [0,0277 Umdrehungen/s ²] linear: [1000x Positionseinheiten ¹⁸ /s ²]	Bremswert Skalierung wie bei Beschleunigung
6067h	0h	unsigned d32	R/W	position window [Positionseinheiten ¹⁸ / user defined units] *1)	Positionsfenster +/- der Zielposition für Erreichen des "Target Reached" Zustands. HINWEIS: Hier muss ein größeres Fenster als bei 3403h.03h movingcap position window (siehe unten) vorgegeben werden. Die unterschiedliche Skalierung muss beachtet werden: 6067h.0h ist in "user defined units". 3403h.3h ist immer in internen Inkrementen.
6068h	0h	unsigned d16	R/W	position window time [ms]	Mindestzeit in Millisekunden, in der die Istposition innerhalb des "position window" liegen muss, bevor der "Target Reached" Zustand gesetzt wird.

6081h	0h	unsigned d32	R/W	profile velocity [Positionseinheiten ^[18] /s oder Umdrehungen/min] *2)	Maximalgeschwindigkeit für Positionierbetrieb ("profile position mode")
-------	----	-----------------	-----	---	---

*1) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)^[18] wird die interne Auflösung des Messsystems (Encoderauflösung) verwendet.

*2) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)^[18] wird als Einheit Umdrehung/Minute verwendet.

*3) Bei Rotativantrieben und größeren Bremswerten (z.B. > 1000), sowie größeren bewegten Massen, muss die beim Bremsen erzeugte Energie/Rückspeisung beachtet werden. Ggf. muss die entstehende Rückspeisung mit einem entsprechend dimensionierten Bremsenergie-System aufgefangen werden.

BEISPIEL CIA 402 POSITIONIERBETRIEB

Um den CiA 402 Positionierbetrieb zu testen, kann folgende **RefGo**-Befehlsfolge über [Weboberfläche](#)^[25] oder [Terminal/TCP-Verbindung](#)^[25] ausgeführt werden:

RefGo Command	OW6060,0,1	--> <i>modes of operation = 1 Velocity Mode</i>
RefGo Command	OW6040,0,6	--> <i>shutdown</i>
RefGo Command	OW6040,0,7	--> <i>switch on</i>
RefGo Command	OW6040,0,15	--> <i>switch on + enabled</i>
RefGo Command	OW6081,0,1000	--> <i>profile velocity</i>
RefGo Command	OW607A,0,30000	--> <i>target position</i>
RefGo Command	OW6040,0,31	--> <i>activate single setpoint</i>

Der Antrieb führt jetzt die Fahrt zur Zielposition aus.

... während der Fahrt ...

RefGo Command	OW6041,0	--> <i>Statusword abfragen</i>
RefGo Response	OR6041,0,39	--> <i>Statusword = 0x027 , Ziel noch nicht erreicht</i>

... nach Erreichen des Ziels ...

RefGo Command	OW6041,0	--> <i>Statusword abfragen</i>
---------------	-----------------	--------------------------------

RefGo Response **OR6041,0,1063**
 ist gesetzt, Ziel erreicht

--> Statusword = 0x0427 , "Target Reached" Bit

MOVINGCAP-REGLERPARAMETER: POSITIONIERGENAUIGKEIT

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
3401h	3h	unsigned32	R/W	movingcap position window [increments]	Positioniergenauigkeit des MovingCap-Reglers. Die Einheit ist immer die interne Auflösung des Positionsreglers / Wegmesssystems. (vgl. 6067h.0h target reached position window oben.)

CIA 402 BEGRENZUNG DES POSITIONIERBEREICHS / SOFTLIMIT

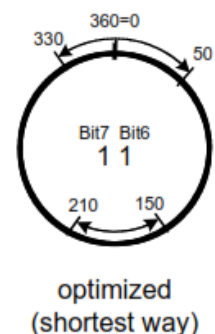
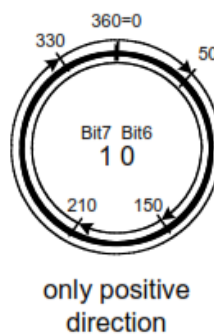
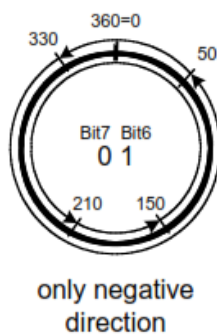
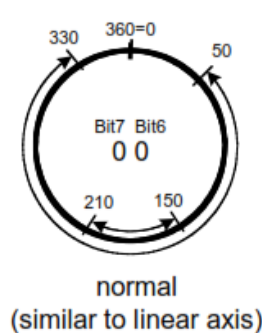
Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
607Dh	1h	integer 32	R/W	min position limit [user-defined units, Positionseinheiten ¹⁸] *1)	minimal mögliche Zielposition für 607Ah.0h target position (s. oben).
607Dh	2h	integer 32	R/W	max position limit [user-defined units, Positionseinheiten ¹⁸] *1)	maximal erlaubte Zielposition für 607Ah.0h target position (s. oben)

*1) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)¹⁸ wird die interne Auflösung des Messsystems (Encoderauflösung) verwendet.

HINWEIS: Wenn beide Werte 607Dh.1h = 0 und 607dh.2h = 0 --> kein Softlimit aktiv / keine Begrenzung des Positionierbereichs.

CIA 402 ROTIERENDE UND RELATIVE BEWEGUNG

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
60F2h	0h	unsigned16	R/W	position option code	Unterstützt Optionen für Relativbewegungen über Bit0 (0001h) und Bit1 (0002h): 60F2h.0h = ...0h - Neue Zielposition ist relativ zur letzten internen Zielposition 60F2h.0h = ...1h - Neue Zielposition ist relativ zu Wert aus Objekt 60FCh.0h Position demand internal value 60F2h.0h = ...2h - Neue Zielposition ist relativ zu Istwert aus Objekt 6064h.0h Position actual value Unterstützt rotierende Koordinatensysteme über die Bit6 (0040h) und Bit7 (0080h): 60F2h.0h = 000xh - Linearbetrieb, keine Fahrt über die min/max position limit hinweg. 60F2h.0h = 004xh - Rotierend, nur negative Richtung. 60F2h.0h = 008xh - Rotierend, nur positive Richtung. 60F2h.0h = 00Cxh - Rotierend, kürzester Fahrweg, auch über die Positionsgrenzen hinweg.



CIA 402 SCHNELLSTOPP / QUICK STOP

MovingCap unterstützt den schnellen Stopp / Schnelles Abschalten über das [CiA 402 Controlword](#) ²⁸.

Das Verhalten bei Aktivieren eines "Quick Stop" wird über folgende Objekte festgelegt:

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
6085h	0h	unsigned32	R/W	quick stop deceleration rotativ: [100 Umdrehungen/min ²] bzw. [0,0277 Umdrehungen/s ²] linear: [1000x Positionseinheiten ¹⁸ /s ²]	Bremswert für Schnellstopp / ("quick Stop")
605Ah	0h	integer16	R/W	quick stop option code	Konfiguration des Schnellstopp- Verhaltens ("quick stop"). MovingCap Antriebe unterstützen folgende Optionen des CiA 402 Standards: 0, 1, 2, 5, 6

605Ah.0h Wert	Definition gem. CiA 402	Beschreibung
0	Disable drive function	Antrieb direkt abschalten
1	Slow down on slow down ramp and transit into switch on disabled	Mit 6084h.0h profile deceleration auf Drehzahl Null bremsen, dann abschalten.
2	Slow down on quick stop ramp and transit into switch on disabled	Mit 6085h.0h quick stop deceleration auf Drehzahl Null bremsen, dann abschalten.
5	Slow down on slow down ramp and stay in quick stop active	Mit 6084h.0h profile deceleration auf Drehzahl Null bremsen, anschließend im Zustand quick stop active ²⁸ bleiben.
6	Slow down on quick stop ramp and stay in quick stop active	Mit 6085h.0h quick stop deceleration auf Drehzahl Null bremsen, anschließend im Zustand quick stop active ²⁸ bleiben.

5.4 Drehzahlbetrieb / Velocity Mode

Für die MovingCap Rotativantriebe ist die [CiA 402 Betriebsart](#)³⁰ "3 - Drehzahlbetrieb" verfügbar, um einen Betrieb mit konstanter Drehzahl und einstellbarer Beschleunigungs- und Bremsrampe zu erhalten.

CIA 402 STANDARDPARAMETER DREHZAHLBETRIEB

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
6060h	0h	integer 8	R/W	modes of operation	3 = Drehzahlbetrieb / Velocity Mode
6040h	0h	unsigned 16	R/W	Controlword	Aktivierung des Geschwindigkeitsbetriebs über den CiA 402 Betriebszustand ^[28] : 15 (000Fh) = Switch on + Enable Operation
6083h	0h	unsigned 32	R/W	profile acceleration rotativ: [100 Umdrehungen/min ²] bzw. [0,0277 Umdrehungen/s ²] linear: [1000x Positionseinheiten ^[18] /s ²]	Beschleunigungswert wie bei Positionierbetrieb ^[30]
6084h	0h	unsigned 32	R/W	profile deceleration rotativ: [100 Umdrehungen/min ²] bzw. [0,0277 Umdrehungen/s ²] linear: [1000x Positionseinheiten ^[18] /s ²]	Bremswert wie bei Positionierbetrieb ^[30]
60FFh	0h	integer 32	R/W	target velocity [Positionseinheiten ^[18] /Sek oder Umdrehungen/min] *1)	Sollgeschwindigkeit

*1) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)^[18] wird als Einheit Umdrehung/Minute verwendet.

BEISPIEL CIA 402 DREHZAHLBETRIEB

Um den CiA 402 Drehzahlbetrieb zu testen, kann folgende **RefGo**-Befehlsfolge über [Weboberfläche](#)^[25] oder [Terminal/TCP-Verbindung](#)^[25] ausgeführt werden:

RefGo Command	OW6060,0,3	--> <i>modes of operation = 3 Velocity Mode</i>
RefGo Command	OW60FF,0,1000	--> <i>target velocity</i>
RefGo Command	OW6040,0,6	--> <i>shutdown</i>
RefGo Command	OW6040,0,7	--> <i>switch on</i>

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

Der Antrieb beschleunigt jetzt bis zur vorgegebenen Solldrehzahl (60FFh.0h target velocity).

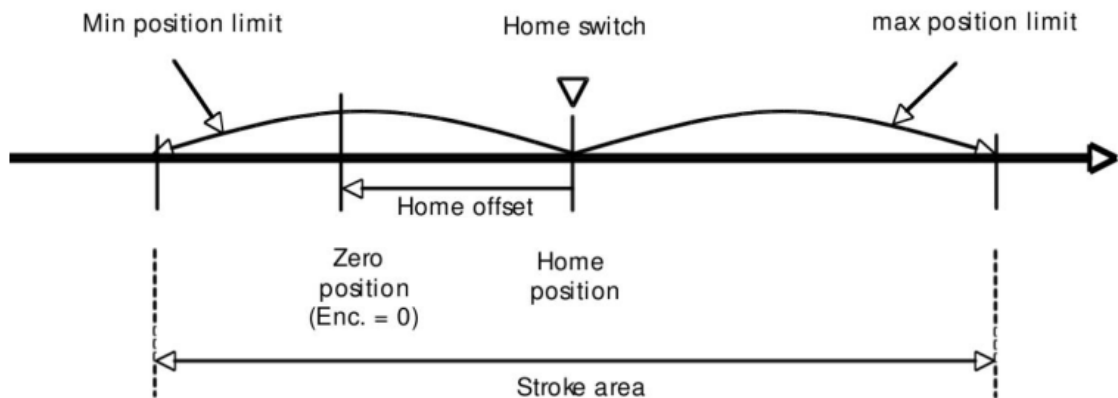
5.5 Referenzierbetrieb / Homing Mode

CIA 402 STANDARDPARAMETER REFERENZIERFAHRT

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
6060h	0h	integer 8	R/W	modes of operation	6 = Referenzierbetrieb / Homing Mode
6040h	0h	unsigned 16	R/W	Controlword	Ansteuerung des Referenzierbetriebs über CiA 402 Betriebszustände ^[28] 15 (000Fh) = Switch on + Enable Operation. Bereit für Referenzierung 31 (001Fh) = Ausführen der Referenzierung
6098h	0h	integer 8	R/W	CiA 402 Homing Method	Gebräuchliche Referenziermethoden: 35 = Aktuelle Position als neue Nullposition setzen. Entspricht RefGo-Kommando ^[27] REF35. -18 = Anschlagsreferenzierung in positive Richtung -19 = Anschlagsreferenzierung in negative Richtung
6099h	1h	unsigned 32	R/W	Homing speed during search for switch [Positionseinheiten] ^[18] / user defined units] *2)	Geschwindigkeit während der Referenzierfahrt / Suche des mechanischen Anschlags
609Ah	0h	unsigned 32	R/W	Homing acceleration rotativ: [100 Umdrehungen/min ²] bzw. [0,0277 Umdrehungen/s ²] linear: [1000x Positionseinheiten ^[18] /s ²]	Beschleunigungswert wie bei Positionierbetrieb
607Ch	0h	integer 32	R/W	Home Offset [user-defined units, Positionseinheiten ^[18]] *1)	Standardwert = 0 Wird als neue aktuelle Position bei Erreichen der Referenzlage / des Anschlags festgelegt.

*1) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)^[18] wird die interne Auflösung des Messsystems (Encoderauflösung) verwendet.

*2) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)^[18] wird als Einheit Umdrehung/Minute verwendet.



5.6 Strom-/Drehmoment-/Kraftbegrenzung

MOVINGCAP-REGLERPARAMETER: STROM-/DREHMOMENT-/KRAFTBEGRENZUNG

Für den Fahrbetrieb mit Vorgabe/Begrenzung des maximalen Drehmoments bzw. der Maximalkraft sind mindestens folgende Objekte einzustellen:

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
6073h	0h	unsigned 16	R/W	max current [0.1% of rated current]	Sollwert für Strom/Drehmoment/Kraftbegrenzung. CiA 402 ^[87] Standardparameter. Einheit: 0.1% des Nennstroms/Nennmoments, d.h. 1000 = Nennmoment, entspricht Nennstrom 6075h.0h.
6075h	0h	unsigned 32	RO	Motor Rated Current [mA]	Nennstrom / Maximalstrom (Kurzzeit). Alle Stromparameter wie 6073h.0h oder 6078h.0h sind relativ zu diesem Wert skaliert. CiA 402 ^[87] Standardparameter.

3401h	1Ah	unsigned 8	R/W	deactivation block detection (Bitwert)	Selektives Deaktivieren der Blockfahrt- und Schleppfehlererkennung. Bitwerte für die Deaktivierung der Blockadeerkennung in unterschiedlichen Bewegungsphasen: 3401h.1ah = 1 - Beschleunigung / acceleration 3401h.1ah = 2 - Abbremsen / deceleration 3401h.1ah = 4 - Konstantgeschw. / constant velocity 3401h.1ah = 8 - Schleppfehlerüberwachung deaktiviert / deactivate following error *1) 3401h.1ah = 15 (hex 0x0F) - alle Block- und Schleppfehler-Überwachungen deaktiviert. *1)
-------	-----	---------------	-----	--	---

*1) Nicht erforderlich / ohne Bedeutung bei MovingCap MC349.

Die Deaktivierung der Blockfahrt-/Schleppfehlererkennung ist erforderlich für kraftbegrenztes Fahren, bei denen die Kraftgrenze die Trajektorie der Fahrt ändert.

Beispiel: Für die Umstellung auf kraftbegrenztes Fahren kann während der Fahrt folgende RefGo-Befehlsfolge über [Weboberfläche](#)^[25] oder [Terminal/TCP-Verbindung](#)^[25] ausgeführt werden:

RefGo Command **OW3401,1A,15** --> *deaktiviert die Blockade- und Schleppfehlererkennung in allen Bewegungsphasen*

RefGo Command **OW6073,0,300** --> *reduziert das Drehmoment auf max. 30% des Nennmoments (Maximales Moment im Stillstand)*

Fährt der Antrieb jetzt auf eine mechanische Begrenzung, wird weiterhin mit 30% der Nennkraft auf den mechanischen Anschlag gedrückt. Es erfolgt kein Abbruch mit [Fehlercode 7121h ERROR_STROKE](#)^[50]. Nach Entfernen der Blockade setzt der Antrieb die Fahrt bis zur Zielposition fort.

ISTWERTE FÜR STROM/DREHMOMENT/KRAFT

Über folgendes Objekt kann die aktuelle Drehmoment/Kraftwirkung ermittelt werden:

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
-------	----------	-----	---------	---------------------	--------------

6078h	0h	integer16	RO	Current actual value [0.1% of rated current]	Istwert des aktuellen Motor(phasen)stroms, proportional zu Drehmoment/Kraft. CiA 402 ⁸⁷ Standardparameter. Einheit: 0.1% des Nennstroms/.moments/-kraft, d.h. 1000 = Nennstrom/moment/kraft
-------	----	-----------	----	---	---

Der Wert ist vorzeichenbehaftet, z.B:

RefGo Command **OR6078,0** --> *aktuellen Motorstrom auslesen*

RefGo Response **OR6078,0,902** --> *ca. 90% vom Nenn-Drehmoment in positive Richtung*

RefGo Command **OR6078,0** --> *aktuellen Motorstrom auslesen*

RefGo Response **OR6078,0,-103** --> *ca. 10% vom Nenn-Drehmoment in negative Richtung*

MC349-REGLERPARAMETER: KRAFTBEGRENZUNG

Die MC349 (MovingCap 349) Motorvariante unterstützt folgende zusätzliche Regelparameter:

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
3401h	15h	unsigned 16	R/W	acceleration torque [0.1 %] *1)	Maximales Drehmoment beim Beschleunigen, in 0.1% des Nennmoments, d.h. 1000 = Nennmoment
3401h	16h	unsigned 16	R/W	deceleration torque [0.1 %] *1)	Maximales Drehmoment beim Abbremsen, in 0.1% des Nennmoments, d.h. 1000 = Nennmoment
3401h	18h	unsigned 16	R/W	stall torque [0.1 %] *1)	Maximales Drehmoment bei Stillstand, in 0.1% des Nennmoments, d.h. 1000 = Nennmoment

HINWEIS: Diese Parameter sind zusätzlich zum CiA 402 Standardparameter **6073h.0h Max Current** wirksam (siehe oben).

MCN23-REGLERPARAMETER: RUHESTROM/KRAFTBEGRENZUNG

Die MCN23 ETH (MovingCap Stepper) Motorvariante unterstützt folgende zusätzliche Regelparameter:

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
3401h	18h	unsigned 16	R/W	stall torque [0.1 %] *1)	Haltemoment / Haltestrom bei Stillstand, in 0.1% des Nennmoments, d.h. 1000 = Nennmoment/Nennstrom

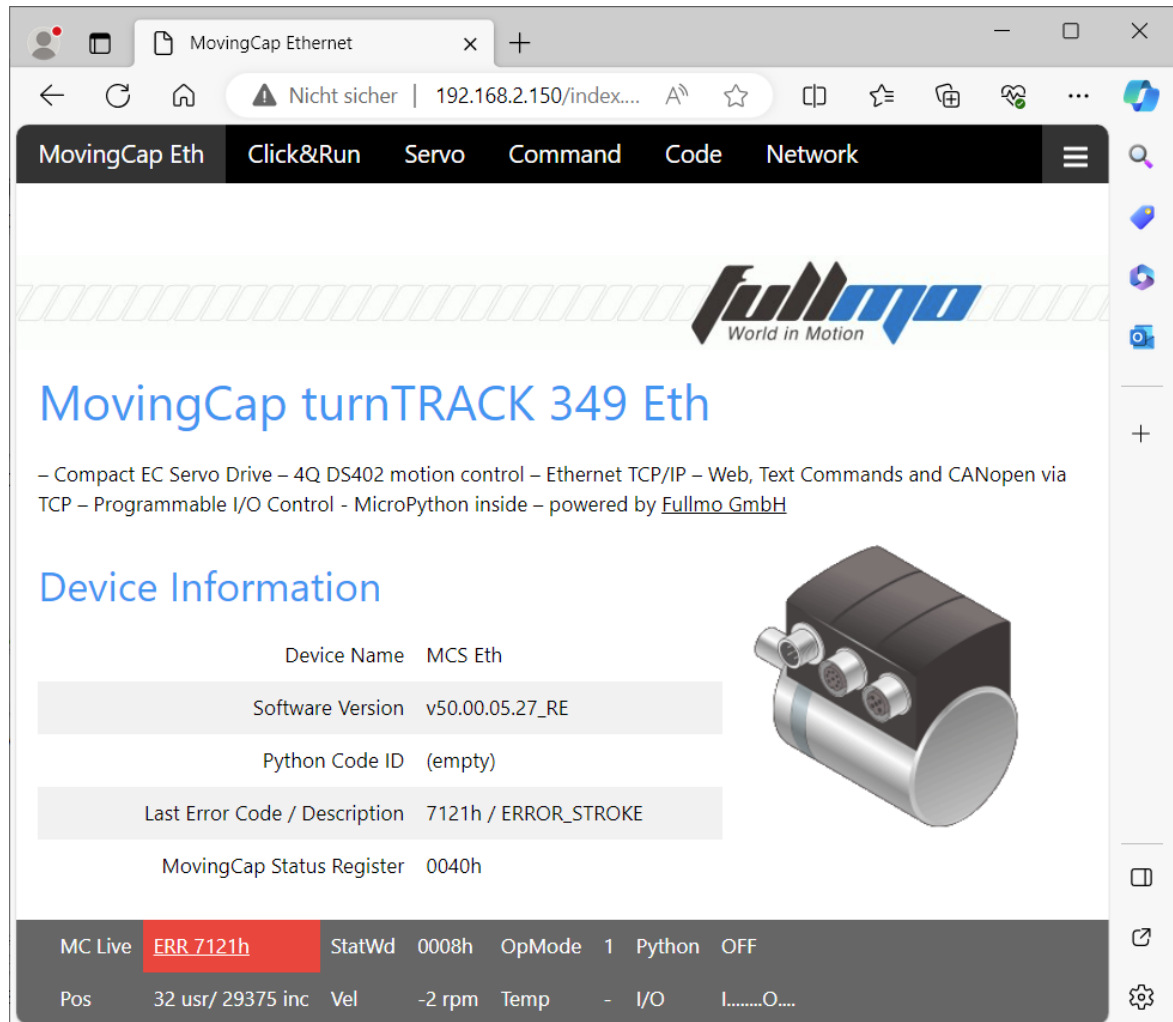
HINWEIS: Die Stromaufnahme / max. Drehmoment während der Fahrt wird durch CiA 402 Standardparameter **6073h.0h Max Current** bestimmt (siehe oben).

HINWEIS: Die Voreinstellung von 3401h.18h stall torque beträgt **100** (entspricht **10% Nennstrom** / Nennmoment). Bei höheren Stillstandsströmen und Dauerbetrieb ist die Wärmeentwicklung zu berücksichtigen. Für Anwendungen mit hohen Haltekräften im Dauerbetrieb empfehlen wir den Einsatz einer MCN23 Variante mit integrierter Haltebremse.

6 Antriebszustand und Fehlermeldungen

6.1 Statusanzeigen auf der Weboberfläche

Die Homepage des Antriebs, sowie die immer sichtbare Fußzeile bieten einen schnellen Überblick über den Antriebszustand:



Bezeichnung	Beispiel	Beschreibung
MC Live	MOV	Allgemeiner Antriebszustand: OFF - Aus RDY - Fahrbereit / In Position MOV - In Bewegung ERR - Fehler, zusammen mit CANopen Fehlercode ⁵⁰ . *1) *2)
StatWd	0427h	Antriebszustand / CiA 402 Statusword 6041h ⁴⁴

OpMode	1	CiA 402 Betriebsart / Modes of operation ^[30]
Python	OFF	OFF - Skript beendet/gestoppt RUN - Skript läuft BRK - Skriptausführung ist unterbrochen/pausiert (zusätzlich) /A - Autostart, siehe Python-Programmsteuerung / Automatischer Start ^[74]
Pos	32 usr/ 29501 inc	usr - Istposition ^[52] in den festgelegten Positionseinheiten ^[18] inc - Istposition in den internen Einheiten der Lagemessung
Vel	106 usr/ 88 rpm	usr - Istgeschwindigkeit ^[52] in Positionseinheiten ^[18] /Sekunde rpm - Istgeschwindigkeit in Umdrehungen/Minute
Temp	46°C	Motor- oder Elektroniktemperatur, nicht in jedem Antrieb verfügbar
I/O	I12.4....O1.3 .	Status Digitale Ein- und Ausgänge. Hinter I folgende die Eingänge, hinter O die Ausgänge. Eine Zahl zeigt an, dass dieser Ein- oder Ausgang aktiv ist, ein Punkt zeigt einen inaktiven I/O. Das Beispiel links zeigt also: Eingänge 1, 2 und 4 sind gesetzt. Ausgänge 1 und 3 sind gesetzt.

*1) Die rote ERR-Fehleranzeige ist ein klickbarer Link und führt auf die Hauptseite zurück. Dort ist bei **Last Error Code / Description** eine Textbeschreibung des Fehlercodes zu sehen. Die Fehlercodes und Beschreibungen entsprechen den [CiA 402 Fehlercodes / Emergency](#)^[50].

*2) Der **Last Error Code** auf der Homepage wird auch nach Beheben des Fehlers weiterhin aufgelistet, um die Diagnose zu erleichtern. Nur wenn in der Fußzeile links unten eine rote **ERRxxxx** Störung angezeigt wird, besteht der Fehler aktuell.

6.2 Antriebszustand / CiA 402 Statusword 6041h

Das Objekt **6041h.0h Statusword** liefert die Informationen zum Antriebszustand gem. der [CiA 402 State Machine](#)^[27].

STATUSWORT 6041H - AKTUELLER ANTRIEBSZUSTAND

Bitnr. (0-15)	15	13	11	10	9	8	7	6	5	4	3	2	1	0
	14	12												

Bedeutung	ms	oms	ila	tr	rm	ms/mv	w	sod	qs	ve	f	oe	so	rtso
------------------	----	-----	-----	----	----	-------	---	-----	----	----	---	----	----	------

Legende:

Bitnr. (0-15)	Bedeutung	Beschreibung
0	rtso	ready to switch on
1	so	switched on
2	oe	operation enabled
3	f	fault
4	ve	voltage enabled
5	qs	quick stop
6	sod	switch on disabled
7	w	warning
8	ms/ mv	manufacturer-specific / MovingCap - "drive moving" *1)
9	rm	remote
10	tr	target reached
11	ila	internal limit active
12,13	oms	operation mode specific
14,15	ms	manufacturer-specific

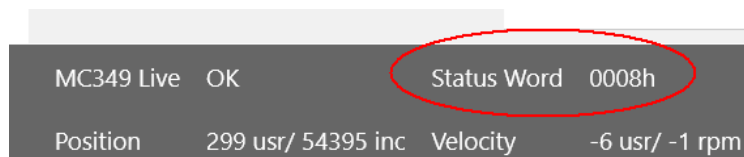
*1) Bit 8 - "drive moving" / Antrieb in Fahrt wird von Softwareversionen ab 12/2024 unterstützt -
Siehe auch Versionshistorie [mc_eth_release_history.txt](#)

Statusword 6041h in Binärdarstellung	PDS FSA Zustand
---	------------------------

(x = beliebiger Wert)	
xxxx xxxx x0xx 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 x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

ANZEIGE / ABFRAGE DES STATUSWORD

Den aktuellen Wert des 6041h Statusword können Sie in der Weboberfläche am unteren Rand sehen:



oder über das folgende [RefGo-Kommando](#) ²⁵ abfragen:

RefGo Command **OW6041,0**

RefGo Response **OR6041,0,1063**

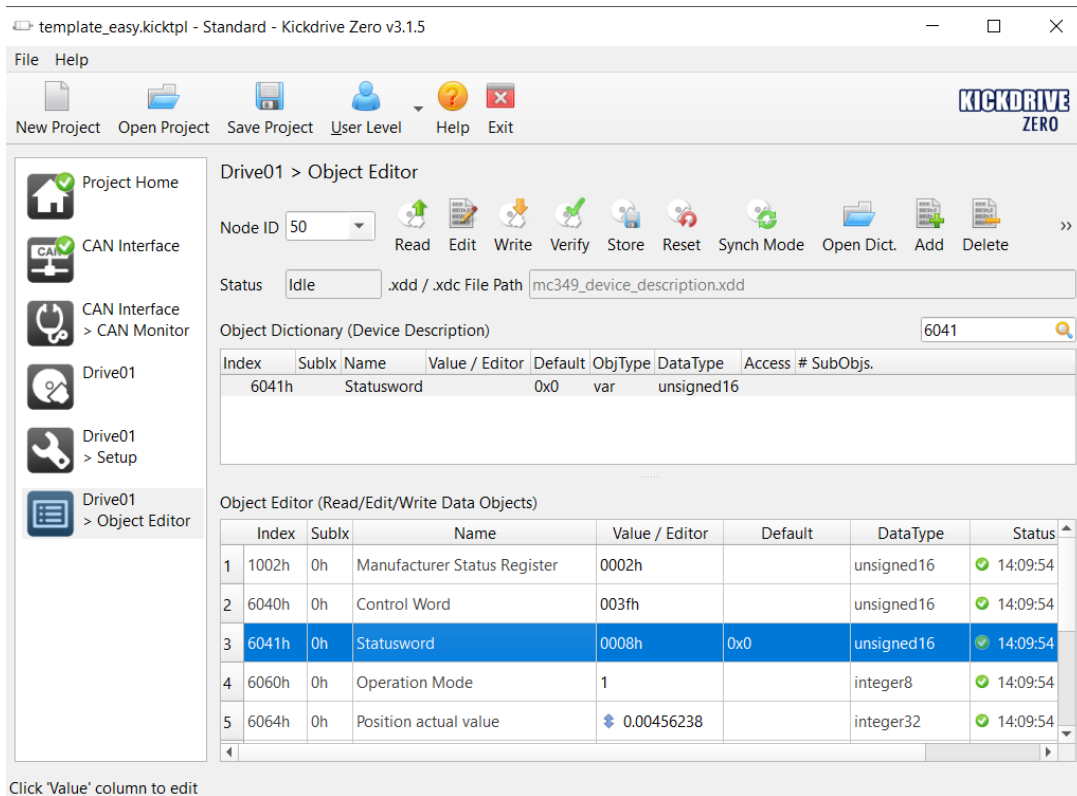
MC349 Command & Parameters

RefGo Command

RefGo Response

OR6041,0,8

oder über den [Kickdrive](#) ²⁰ **Object Editor**:



6.3 MovingCap Statusregister 1002h

Details zu Antriebsfehlern und -warnungen können über das Objekt

1002h.0h Manufacturer Status Register

angezeigt werden. Um den aktuellen Wert auszulesen können Sie über die [Weboberfläche](#)²⁵⁾ im Bereich Parameters eingeben:

RefGo Command **OR1002,0**

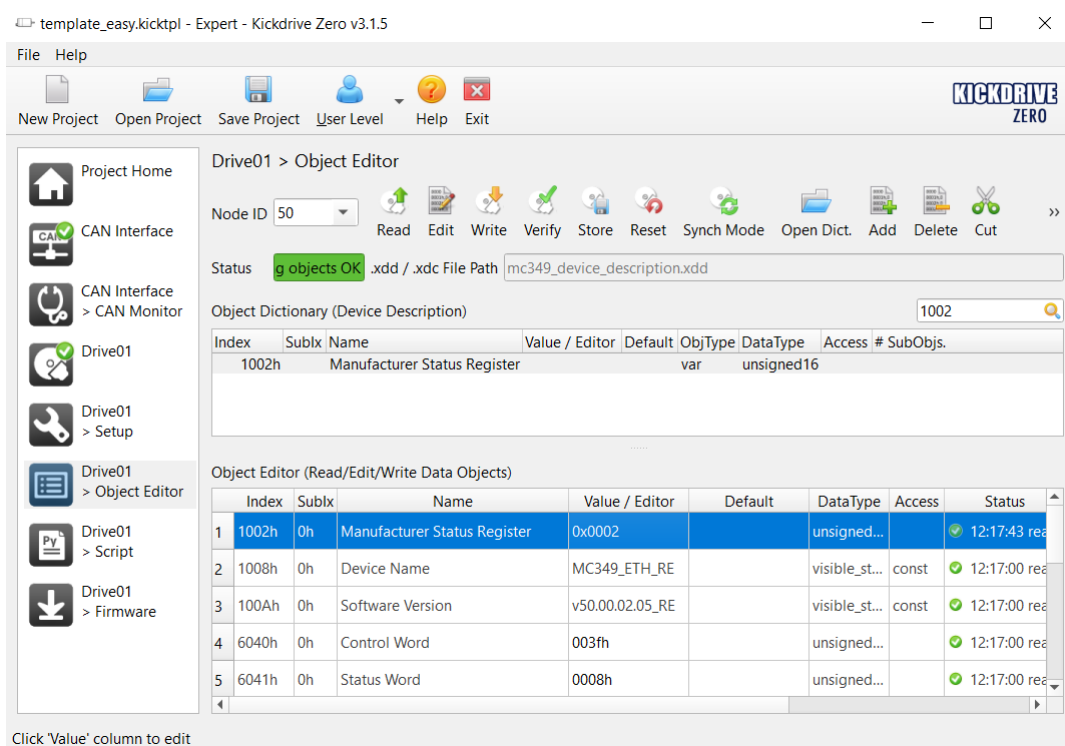
MC349 Command & Parameters

RefGo Command

RefGo Response

OR1002,0,2

oder in [Kickdrive](#)²⁰⁾ den **Object Editor** nutzen:



STATUSREGISTER 1002H.0H - FEHLER

Die folgenden Zustände sind Fehlerzustände. Der Antrieb geht bei Auftreten dieser Fehler in den [CiA 402 Fault Zustand](#)^[28], erzeugt eine [CANopen Emergency Message](#)^[50] und das [Fault-Bit im Statusword](#)^[44] wird gesetzt.

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

STATUSREGISTER 1002H.0H - WARNUNGEN

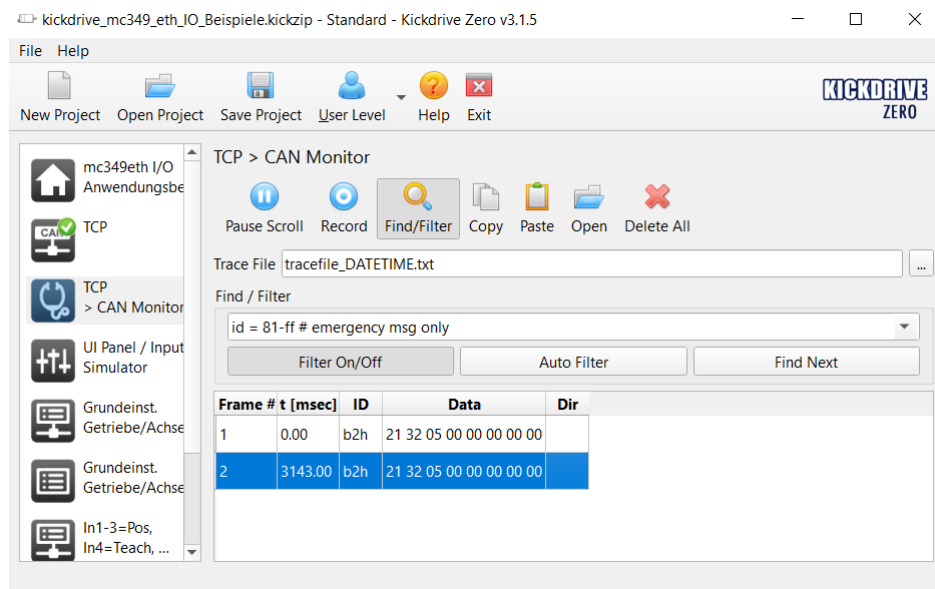
Bitwerte ab Bit16 / 10000h / 65536 dez. sind Warnzustände. Sie führen nicht zu einem Antriebsfehler, die aktuelle Betriebsart wird beibehalten und Fahrbefehle werden weiterhin ausgeführt.

Objekt 1002h.0h Bitwerte (kombinierbar)	Beschreibung
Bit16 / 10000h / 65536 dez.	Warning Sensor *1)

*1) Z.B. Warnzustand eines Multiturn-Encodersystems: Positionsermittlung ist weiterhin möglich, aber es steht eine Warnung an, wie "Battery Low". MovingCap Multiturn-Absolutencoder besitzen eine Lifetime-Batteriepufferung für >10 Jahre - nur im Fehler/Defekt-Fall kann eine Sensorwarnung bereits nach kurzer Betriebszeit ausgelöst werden.

6.4 CiA 402 Fehlernachrichten / Emergency Messages

Über die Kickdrive-Software können Fehlerzustände bzw. der Zeitpunkt des Auftretens genau bestimmt werden. Aktivieren Sie hierzu das **CAN Monitor** Fenster und starten Sie die Kommunikation in Kickdrive, bevor Sie Ihren Test ausführen. Benutzen Sie ggf. die **Find/Filter** Funktion um gezielt nach sog. CANopen Emergency Messages (**id = 81-ff**) zu suchen:



CANOPEN EMERGENCY LISTE / FEHLERCODES

Emergency Fehlercode	Emergency / Fehler-Beschreibung	CAN Monitor Nachricht / Data
2220h / 0x2200 / 8736 dezimal	ERROR_OVER_CURRENT Überstrom	20 22 05 00 00 00 00 00
3211h / 0x3211 / 12817 dezimal	ERROR_OVER_VOLT Überspannung UZK Leistungsverorgung Motor	11 32 05 00 00 00 00 00
3221h / 0x3221 / 12833 dezimal	ERROR_UNDER_VOLT Unterspannung UZK Leistungsverorgung Motor	21 32 05 00 00 00 00 00
4210h / 0x4210 / 16912 dezimal	ERROR_OVER_TEMP Übertemperatur	10 42 09 00 00 00 00 00
4220h / 0x4220 / 16928 dezimal	ERROR_DERATING Derating	20 42 05 00 00 00 00 00

5200h / 0x5200 / 20992 dezimal	ERROR_ACK Acknowledge Error	00 52 09 00 00 00 00 00
7121h / 0x7121 / 28961 dezimal	ERROR_STROKE Blockade / Blockfahrt / Schleppfehler zu groß	21 71 10 00 00 00 00 00
7510h / 0x7510 / 29968 dezimal	ERROR_COMMUNICATION Motorkommunikationsfehler	10 75 10 00 00 00 00 00
FFF1h / 0xFFF1 / 65521 dezimal	ERROR_SENSOR_1 Sensor/Encoderfehler	F1 FF 10 00 00 00 00 00
FFF2h / 0xFFF2 / 65522 dezimal	ERROR_HW_SW_ENABLE HW Enable Fehler (fehlende interne Freigabe) oder SW Enable Fehler (siehe Eingangsfunktion 9)	F2 FF 10 00 00 00 00 00
FFF4h / 0xFFF4 / 65524 dezimal	ERROR_EXT_TORQUE Externe Krafteinwirkung/Externes Moment	F4 FF 10 00 00 00 00 00
FFF5h / 0xFFF5 / 65525 dezimal	ERROR_NEEDS_RESET Ein Spannungsreset ist vor dem Weiterbetrieb erforderlich, z.B. nach erfolgreichem Ausführen eines Software-Updates.	F5 FF 10 00 00 00 00 00

6.5 CiA 402 Fehlercodes / History

Die als [CANopen Emergency](#) ^[50] gemeldeten Fehlercodes können auch über das Objekt **603Fh.0h Error Code** ausgelesen werden. Z.B. können Sie über die [Weboberfläche](#) ^[25] im Bereich Parameters eingeben:

RefGo Command **OR603F,0**

Der angezeigte Rückgabewert ist jedoch immer dezimal und Sie müssen ggf. mit dem (Windows)-Taschenrechner den Wert in Hexadezimal konvertieren, z.B.

RefGo Response **OR603F,0,12833** --> Fehler gem. Tabelle in [CANopen Emergency](#) ^[50] -> 3221h / 12833 dezimal = Unterspannung UZK Leistungsverorgung Motor

Die letzten vier Fehlercodes können über die Objekte **1003h.1h-4h Standard error field** angezeigt werden, z.B.

RefGo Command **OR1003,1**RefGo Response **OR1003,1,12833**

6.6 Istwerte - Position, Geschwindigkeit, Strom, Temperatur

CIA 402 STANDARD-ISTWERTE FÜR POSITION, GESCHWINDIGKEIT UND MOTORSTROM

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
6064h	0h	integer32	RO	Position actual value [Positionseinheit ¹⁸] / user defined units 1*)	Aktuelle Istposition, unter Berücksichtigung der eingestellten Positionseinheiten und Nulllage ¹⁸ , sowie Sondereinstellung für rotierende und relative Bewegung ³³ .
606Ch	0h	integer32	RO	Velocity actual value [Positionseinheit ¹⁸ /Sek oder Umdrehungen/min] 2*)	Aktuelle Geschwindigkeit
6078h	0h	integer16	RO	Current actual value [0.1% of rated current]	Istwert des aktuellen Motor(phasen)stroms, proportional zu Drehmoment/Kraft, siehe auch Strom-/Drehmoment-/Kraftbegrenzung ³⁹ .

*1) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)¹⁸ wird die interne Auflösung des Messsystems (Encoderauflösung) verwendet.

*2) Ohne abweichende Einstellung von [Positionseinheiten/Skalierung](#)¹⁸ wird als Einheit Umdrehung/Minute verwendet.

MOVINGCAP-ISTWERT: TEMPERATUR

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
-------	----------	-----	---------	---------------------	--------------

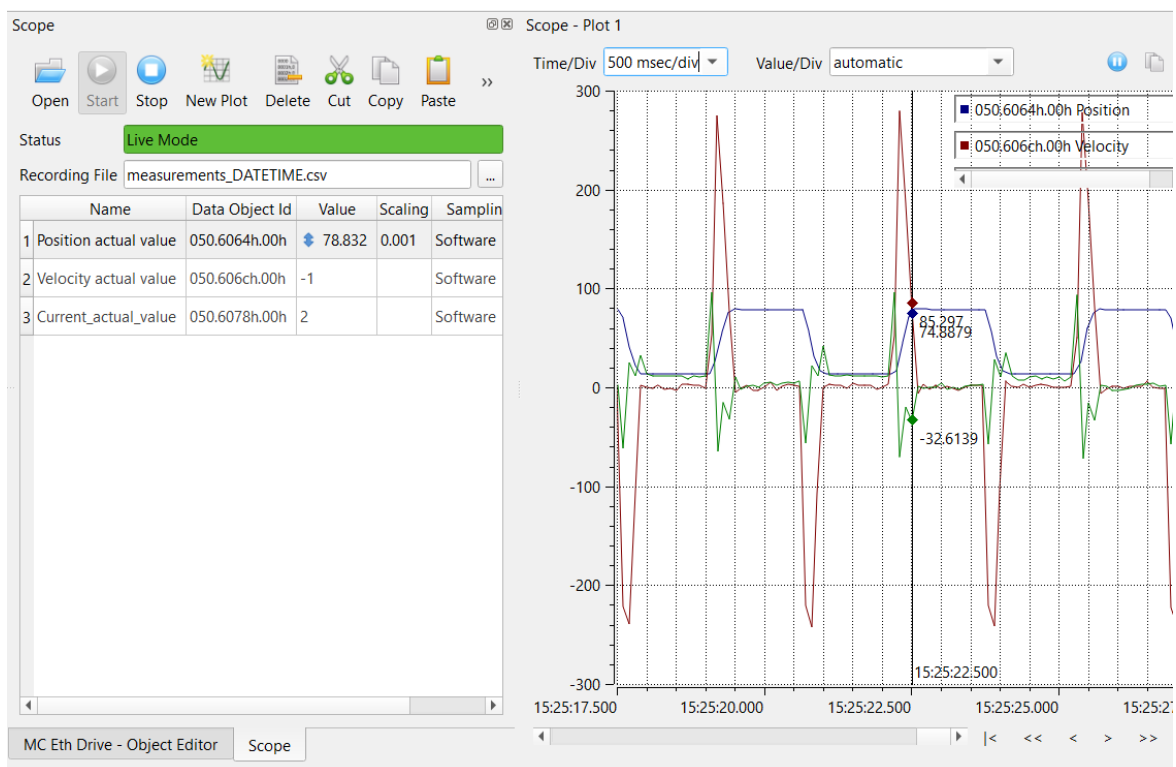
3401h	0Ah	unsigned 8	RO	motor actual temperature [°C]	Aktuelle Motortemperatur. Je nach MovingCap-Variante die Wicklungs- oder Elektronik-Temperatur, bzw. der kombinierte Messwert (Umschaltung zwischen Elektronik- und Motortemperatur im Sekundentakt.)
-------	-----	---------------	----	-------------------------------------	---

Um Istwerte auszulesen können Sie das entsprechende RefGo-Kommando über die [Weboberfläche](#) ²⁵ eingeben, oder an TCP-Port 10001 senden.

Beispiel für das Auslesen des Temperaturwerts:

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

Oder nutzen Sie [MovingCap SLCAN - CANopen CiA 402 mit Kickdrive](#) ²⁰ und das komfortable Scope in Kickdrive mit Aufzeichnungs- und Triggernfunktionen:



7 MovingCap IO

Die MovingCap Ethernet-Antriebe bieten je nach Motorvariante und Optionen

- bis zu 10x digitale Eingänge 24V DC (IN1-IN10)

- bis zu 4x digitale Ausgänge 24V DC (OUT1-4)

Die Anschluss/Pinbelegung finden Sie im Abschnitt [Anschlüsse](#) .

Über die Webseite wird am unteren rechten Rand der aktuelle Zustand der Ein/Ausgänge angezeigt:

MC Live	OFF	StatWd	0040h	OpMode	0	Python	OFF
Pos	1520 inc	Vel	2 rpm	Temp	-	I/O	I.....O.23.

Eine Ziffer hinter "I" bedeutet, dass diese Eingangsnummer (IN..) aktiv ist.

Eine Ziffer hinter "O" bedeutet, dass diese Ausgangsnummer (OUT..) aktiv ist.

In obigem Bild sind Ausgänge OUT2 und OUT3 high, alle Eingänge sind low.

Der Softwarezugriff auf die digitalen Eingänge und Ausgänge erfolgt über die folgenden CiA 402 Objekte des CANopen Objektverzeichnisses:

Index	Subindex	Typ	Zugriff	Beschreibung	Beispiel über Weboberfläche/RefGo 
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 --> an IN2 liegen 24V an (IN2=high)
60FEh	01h	Unsigned16	R/W	Physical Outputs OUT1 = Bit16 (00010000h, Dezimal 65536) OUT2 = Bit17 (00020000h, Dezimal 131072) ...	RefGo Command: OW 60FE,1,65536 RefGo Response: OW 60FE,1,65536,OK --> setze OUT1 = High, (OUT2.. = low) RefGo Command: OW 60FE,1,393216 RefGo Response: OW 60FE,1,393216,OK --> setze OUT2 = High, OUT3 = High (OUT1 bzw. OUT4) = low

HINWEIS: Die Zuordnung der Ein- und Ausgänge auf Objekte 60FDh.00h und 60FEh.01h mit jeweils Bitnummern ≥ 16 ("manufacturer-specific") entspricht der CiA 402-Spezifikation.

7.1 Eingangsfunktionen

Jeder digitale Eingang kann eine der nachfolgend beschriebenen Eingangsfunktionen auslösen.

Für jeden Eingang kann eingestellt werden:

- Welche Funktion bei steigender Flanke (Eingang wechselt von 0V 'Low' auf 24V 'High') ausgelöst wird.
- Welche Funktion bei fallender Flanke (Eingang wechselt von 24V 'High' auf 0V 'Low') ausgelöst wird.

Die Funktionen werden über folgende Parameterobjekte des CANopen Objektverzeichnisses eingestellt:

Objekt 3511h - Reaktion auf IN1

Objekt 3512h - Reaktion auf IN2

Objekt 3513h - Reaktion auf IN3

...

Objekt 3518h - Reaktion auf IN8

Objekt 3519h - Reaktion auf IN9

Objekt 351Ah - Reaktion auf IN10

HINWEIS: Jeder MovingCap Ethernet Antrieb bietet bis zu 10 Programmierobjekte, auch wenn der Motor - je nach Modell und Optionen - weniger als 10 digitale Eingänge als elektrische Schnittstelle anbietet. Die zusätzlichen Objekte können genutzt werden, um weitere Fahrprofile abzulegen, die über das Objekt 3510.01h Virtual Inputs oder das Objekt 3510h.02h Input Fct Command gestartet werden können.

TIPP: Die Eingangsfunktionen können direkt über das Webinterface im Menü **IO IN** konfiguriert werden. Kickdrive-Beispielprojekte mit vorbereiteten Objektlisten und Beispielkonfigurationen finden Sie im [movingcap.de Service-Portal](https://movingcap.de/Service-Portal), dort nach **MovingCap IO** suchen.

7.1.1 Parameterobjekte Eingangsfunktionen 3511h - 351Ah

Jedes der Objekte 3511h - 351Ah besteht aus folgenden Elementen:

Subindex	Typ	Zugriff	Standardfunktion *1)
01h	Unsigned16	R/W	HiByte - Funktionsnummer steigende Flanke LoByte - Funktionsnummer fallende Flanke
02h	Unsigned16	R/W	(reserviert für PDO-Steuerung, derzeit ungenutzt)
03h	Integer32	R/W	Ergebnisregister 1
04h	Integer32	R/W	Ergebnisregister 2
05h	Integer32	R/W	Timer - Zeitverzögerung in Millisekunden für die Ausführung von Funktion "Timed Motion" und "Set Home"
06h	Integer32	R/W	Sollgeschwindigkeit
07h	Integer32	R/W	Profile acceleration / Beschleunigungswert
08h	Integer32	R/W	Zielposition
09h	Integer32	R/W	Profile deceleration / Bremswert
0Ah	Integer32	R/W	Max Current / Drehmomentbegrenzung

*1) Alle Parameter ab Subindex 03h sind als Integer32 ausgeführt, auch wenn dies für die Standardfunktion teilweise nicht erforderlich ist. Bei einzelnen Eingangsfunktionen, z.B. [Funktion 7 - Set Home / Referencing](#) ⁶³ haben einzelne Subindexe andere Bedeutungen.

7.1.2 Funktion 1 - Capture Position

Funktion 1 speichert beim Auslösen die aktuelle Position in Ergebnisregister 1.

Subindex	Typ	Zugriff	Beschreibung
03h	Integer32	R/W	Ergebnisregister 1 : die aktuelle Position beim Erreichen des Ereignisses [user-defined units, Positionseinheiten ₁₈]

KONFIGURATIONSBEISPIELE

Konfiguration	Erklärung
3511h.01h = 0100h	Beim Auftreten einer steigenden Flanke an digitalem Eingang 1 wird die aktuelle Position im Ergebnisregister 1 (Subindex 3) abgelegt.
3512h.01h = 0001h	Beim Auftreten einer fallenden Flanke an digitalem Eingang 2 wird die aktuelle Position im Ergebnisregister 1 (Subindex 3) abgelegt.
3513h.01h = 0102h	Beim Auftreten einer steigenden Flanke an digitalem Eingang 3 wird die aktuelle Position im Ergebnisregister 1 (Subindex 3) abgelegt. Beim Auftreten einer fallenden Flanke an digitalem Eingang 3 wird die aktuelle Position im Ergebnisregister 2 ₅₇ (Subindex 4) abgelegt.

7.1.3 Funktion 2 - Capture Position

Funktion 2 speichert beim Auslösen die aktuelle Position in Ergebnisregister 2.

Subindex	Typ	Zugriff	Beschreibung
04h	Integer32	R/W	Ergebnisregister 2 : die aktuelle Position beim Erreichen des Ereignisses [user-defined units, Positionseinheiten ₁₈]

KONFIGURATIONSBEISPIELE

Konfiguration	Erklärung
3511h.01h = 0200h	Beim Auftreten einer steigenden Flanke (digitaler Eingang 1) wird die aktuelle Position im Ergebnisregister 2 (Subindex 4) abgelegt.
3511h.01h = 0002h	Beim Auftreten einer fallenden Flanke (digitaler Eingang 1) wird die aktuelle Position im Ergebnisregister 2 (Subindex 4) abgelegt.
3511h.01h = 0102h	Beim Auftreten einer steigenden Flanke an digitalem Eingang 1 wird die aktuelle Position im Ergebnisregister 1 (Subindex 3) abgelegt. Beim Auftreten einer fallenden Flanke an digitalem Eingang 1 wird die aktuelle Position im Ergebnisregister 2 (Subindex 4) abgelegt.

7.1.4 Funktion 3 - Counter

Funktion 3 inkrementiert beim Auslösen den Inhalt des Ergebnisregisters 1 um 1. Der Inhalt des Ergebnisregisters 1 kann durch einen schreibenden Zugriff initialisiert werden.

KONFIGURATIONSBEISPIELE

Konfiguration	Erklärung
3511h.01h = 0300h	Beim Auftreten einer steigenden Flanke (digitaler Eingang 1) wird der Inhalt von Ergebnisregister 1 (Subindex 3) um 1 erhöht.
3511h.01h = 0003h	Beim Auftreten einer fallenden Flanke (digitaler Eingang 1) wird der Inhalt von Ergebnisregister 1 (Subindex 3) um 1 erhöht.
3511h.01h = 0303h	Beim Auftreten einer steigenden Flanke (digitaler Eingang 1) wird der Inhalt von Ergebnisregister 1 (Subindex 3) um 1 erhöht. Beim Auftreten einer fallenden Flanke (digitaler Eingang 1) wird der Inhalt von Ergebnisregister 1 (Subindex 3) um 1 erhöht.

7.1.5 Funktion 4 - Timed Motion Absolute

Funktion 4 löst beim Auftreten einen zuvor konfigurierten Fahrbefehl nach Ablauf einer definierten Zeit aus:

- Wechsel in Positionierbetrieb, falls noch nicht in diesem Zustand.
- Anfahren der Zielposition mit der vorgegebenen Drehzahl/Beschleunigung.

Subindex	Typ	Zugriff	Beschreibung
03h	Integer32	R/W	Ergebnisregister 1 : letzte verwendete Zielposition in internen Positionseinheiten [increments] Dieser Wert wird als Vergleichswert für die Ausgangsfunktion 4 - Rückmeldung zur Positionierung ^[69] verwendet.
04h	Integer32	R/W	Ergebnisregister 2 : Anzahl ausgelöste Fahrbefehle
05h	Integer32	R/W	Timer - Zeitverzögerung in Millisekunden
06h	Integer32	R/W	Sollgeschwindigkeit
07h	Integer32	R/W	Profile acceleration / Beschleunigungswert
08h	Integer32	R/W	Zielposition
09h	Integer32	R/W	Profile deceleration / Bremswert
0Ah	Integer32	R/W	Max Current / Drehmomentbegrenzung [1/1000 of rated current], z.B. Einstellwert 250 = 25% des Nennstroms/moment

sowie weitere Einstellungen des Positionsreglers gem. [Einstellungen für Positionierbetrieb](#) ^[30]

KONFIGURATIONSBEISPIEL

Objektwerte	As OW-Befehle über die Webseite	Erklärung
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	Beim Auftreten einer steigenden Flanke (digitaler Eingang 1) wird nach Ablauf einer Sekunde die Position 6300 angefahren. Fahrparameter sind Geschwindigkeit = 500, Beschleunigung = 400, Bremswert = 200 Die OW-Befehle können über die Weboberfläche ^[25] oder Terminal/TCP-Verbindung ^[25] ausgeführt werden.

7.1.6 Funktion 5 - Quick Stop

Funktion 5 aktiviert "Quick Stop" mit folgendem Parameter:

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

Der Antrieb bremst mit der Beschleunigung ab, die in

Objekt **6085h Quick stop deceleration**

festgelegt wurde.

HINWEIS: Um die Positionierung komplett zu deaktivieren, kann [Eingangsfunktion 9 - Software Enable](#)  verwendet werden.

7.1.7 Funktion 6 - Timed Motion Relative

Wie [Funktion 4](#) ^[59], aber relative/fortlaufend addierende Positionierung

Subindex	Typ	Zugriff	Beschreibung
03h	Integer32	R/W	Ergebnisregister 1 : Letzte verwendete Zielposition absolut in internen Positionseinheiten [increments. Dieser Wert wird als Vergleichswert für die Ausgangsfunktion 4 - Rückmeldung zur Positionierung ^[69] verwendet.
04h	Integer32	R/W	Ergebnisregister 2 : Anzahl ausgelöste Fahrbefehle
05h	Integer32	R/W	Zeitverzögerung in Millisekunden
06h	Integer32	R/W	Sollgeschwindigkeit
07h	Integer32	R/W	Profile acceleration / Beschleunigungswert
08h	Integer32	R/W	Zielposition relativ, unter Berücksichtigung von 60F2h.0h position option code ^[33]
09h	Integer32	R/W	Profile deceleration / Bremswert
0Ah	Integer32	R/W	Max Current / Drehmomentbegrenzung [1/1000 of rated current], z.B. Einstellwert 250 = 25% des Nennstroms/moment

HINWEIS: Bei fortlaufenden Relativbewegungen (z.B. immer 180° vorwärts) sollte [60F2h.0h = ...0](#) ^[33] (relativ zur letzten internen Zielposition) verwendet werden. Dadurch ist sichergestellt, dass sich die im Rahmen der Regelgenauigkeit ergebene Differenz zwischen exakter Sollposition und aktueller Istposition nicht zu einem relevanten Summenfehler aufaddiert.

HINWEIS: Nach Einschalten des Positionierbetriebs wird als Ausgangswert für die Relativpositionierung immer die aktuelle Istposition vorgegeben. Das Anfahren einer Grundstellung erfolgt in der Regel über Absolutpositionierung, [Eingangsfunktion 4 - Timed Motion Absolute](#) ^[59].

7.1.8 Funktion 7 - Set Home / Referencing

Referenzierung: Setzen der Nulllage, bzw. Referenzierungsfahrt auf Endlagenschalter oder mechanischen Anschlag.

Subindex	Typ	Zugriff	Standardfunktion *1)
05h	Integer32	R/W	Timer - Zeitverzögerung in Millisekunden für die Ausführung der Referenzierung
06h	Integer32	R/W	Sollgeschwindigkeit
07h	Integer32	R/W	Profile acceleration / Beschleunigungswert
08h	Integer32	R/W	Homing Offset Welcher Positionswert soll nach erfolgreichem Abschluss der Referenzierfahrt als aktuelle Position gesetzt werden? Voreinstellung = 0
09h	Integer32	R/W	CiA 402 Homing Method z.B: 35 = Aktuelle Position als neue Nullposition setzen. Entspricht RefGo  -Kommando REF35 . -18 = Anschlagsreferenzierung in positive Richtung -19 = Anschlagsreferenzierung in negative Richtung
0Ah	Integer32	R/W	Homing Torque / Drehmomentbegrenzung für die Referenzierfahrt [1/1000 of rated current], z.B. Einstellwert 250 = 25% des Nennstroms/moment

7.1.9 Funktion 8 - Teach Target

Wenn der Eingang auf High gesetzt wird, ändert sich das Verhalten aller anderen Eingänge:

- Wenn ein anderer Eingang mit Funktion **4 Timed Motion Absolute** belegt ist, wird bei High-Zustand dieses Kontakts keine Fahrt ausgeführt, sondern die aktuelle Position als neue Zielposition hinterlegt (Teach Target).

7.1.10 Funktion 9 - Software Enable

Funktion 9 kann verwendet werden, um die Applikationserstellung zu vereinfachen und einen Eingangskontakt als Vorbedingung für den [Zustand "Switched On"](#) ²⁸ zu definieren.



WARNUNG! Bei der Funktion "Software Enable" handelt es sich NICHT um eine Sicherheitsfunktion. Sie ist NICHT dazu bestimmt, Sicherheitsfunktionen im Rahmen der Maschinensicherheit umzusetzen.

HINWEIS: Bei fehlendem Software Enable Eingang wird der [Fehlercode](#) ⁵⁰ **FFF2 / ERROR_HW_SW_ENABLE** erzeugt.

Subindex	Typ	Zugriff	Wert / Funktion
01h	Unsigned16	R/W	0900h - Antrieb kann nur in den Zustand "Switched On" geschaltet werden, wenn Eingang = High 0009h - Antrieb kann nur in den Zustand "Switched On" geschaltet werden, wenn Eingang = Low

7.1.11 Eingänge simulieren

Um Eingangsfunktionen zu testen, ohne die elektrischen Eingänge des Antriebs elektrisch mit 0V oder 24V zu beschalten, können Sie folgende Objekte nutzen:

Index	Subindex	Typ	Zugriff	Beschreibung	Beispiel über Weboberfläche/RefGo ²⁵
3510h	01h	Unsigned32	R/W	Virtual Inputs Simulieren von "Eingang INx ist 24V / High". Wertebereich wie bei Objekt 60FD ⁵³, d.h. 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 --> Antrieb "sieht" jetzt für IN2 einen 24V /High Zustand und löst ggf. die festgelegte Eingangsfunktion aus.
3510h	02h	Unsigned16	R/W	Input Function Command Parameterformat YYZZh - Triggert Eingangsfunktion ZZ mit Parametern aus Eingang Nr. YY. z.B. 0304h - Führt "Funktion 4 - Timed Motion Absolute" aus, mit den Fahrparametern, die bei Eingang Nr. 3 / Objekt 3513h gespeichert sind.	RefGo Command: OW 3510h,2,772 RefGo Response: OW 3510h,2,772,OK --> entspricht dem HEX-Zahlenwert 0304h, d.h. es wird wie links beschrieben Funktion 4 ausgeführt.

7.2 Ausgangsfunktionen

Jeder digitalen Ausgang kann für die nachfolgend beschriebenen Ausgangsfunktionen parametrierbar werden.

Die Parametrierung erfolgt über die CANopen Objekte:

Objekt 3611h - Steuerung von OUT1

Objekt 3612h - Steuerung von OUT2

...

Objekt 3614h - Steuerung von OUT4

Subindex	Typ	Zugriff	Beschreibung
1	Unsigned16	R/W	Funktionsnummer
2 .. 6		R/W	(siehe Funktionsbeschreibung der ausgewählten Funktionsnummer)

TIPP: Die Eingangsfunktionen können direkt über das Webinterface im Menü **IO OUT** konfiguriert werden. Kickdrive-Beispielprojekte mit vorbereiteten Objektlisten und Beispielkonfigurationen finden Sie im [movingcap.de Service-Portal](https://movingcap.de/Service-Portal), dort nach **MovingCap IO** suchen.

HINWEIS: Bei gleichzeitiger Verwendung von [Eingangsfunktionen](#)^[55] und Ausgangsfunktionen beachten Sie bitte die IO-Konfiguration am [Anschlussstecker](#)^[5] Ihrer Antriebsvariante. Stellen Sie sicher, dass an einem kombinierten Ein-/Ausgang (IO1, IO2, ...) entweder eine Eingangs- oder eine Ausgangsfunktion genutzt wird, aber nicht beides gemeinsam. Andernfalls führt das Setzen des Ausgangs (z.B. als [Funktion 4 - Rückmeldung zur Positionierung](#)^[69]) zu einer Änderung des entsprechenden Eingangs und damit ggf. auch zum Ausführen einer konfigurierten Eingangsfunktion. Dies ist in der Regel keine sinnvolle Anwendungskonfiguration und führt zu unerwartetem Verhalten.

7.2.1 Parameterobjekte Ausgangsfunktionen 3611h - 3614h

Jedes der Objekte 3611h - 3614h besteht aus folgenden Elementen:

Subindex	Typ	Zugriff	Beschreibung
----------	-----	---------	--------------

1	unsigned16	R/W	Funktionsnummer
2 .. 6	integer32	R/W	(siehe Funktionsbeschreibung der ausgewählten Funktionsnummer)

Enter topic text here.

7.2.2 Funktion 1 - Statusword 6041h Monitor

Überwachung des [CiA 402 Statusworts 6041h](#)⁴⁴.

Setzen des Ausgangs, wenn das/die ausgewählten Bit(s) im Objekt 6041h (CiA 402 Statuswort) gesetzt sind.

Subindex	Bezeichnung	Vorgabe
2	Bitmaske (kombinierbar) gem. CiA 402 Statusword 6041h ⁴⁴ .	

7.2.3 Funktion 2 - Error Register 1002h Monitor

Überwachung des [MovingCap Fehlerregister 1002h](#)⁴⁷.

Setzen des Ausgangs, wenn das/die ausgewählten Bit(s) im Objekt 1002h gesetzt sind.

Subindex	Bezeichnung
2	Bitmaske (kombinierbar), gemäß MovingCap Fehlerregister 1002h ⁴⁷ .

7.2.4 Funktion 3 - Limitüberwachung rechts/links

Aktivierung und Deaktivierung über Positionsbedingung. Die Pulsdauer wird über ein Positionsfenster (right/left position distance) festgelegt.

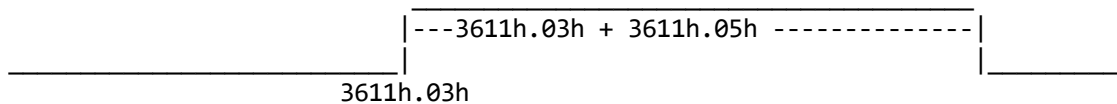
Subindex	Bezeichnung
2	1 = right limit active 2 = left limit active 3 = both
3	right position
4	left position
5	right position distance
6	left position distance

Beispiel :

3611.01h = 3

3611.02h = 1

--> Fahrt in positive Richtung,



7.2.5 Funktion 4 - Rückmeldung zur Positionierung

Bestätigt das Erreichen / Halten der Zielposition. Allgemein oder spezifisch für einen Eingang mit [Eingangsfunktion 4 - "Timed Motion Absolute"](#)^[59]

Index	Subindex	Wert / Beschreibung
3611h-3614h	2	0: letzte Zielposition erreicht (unabhängig von Eingang). Die Vergleichsposition ist hier die Position gem. Position Option Code (Objekt 60F2h.0h)^[33]. Bei 60F2h.0h = 2 ("verwende Istwert") ist diese Einstellung nicht sinnvoll, da diese Bedingung immer erfüllt ist. 1-8: Nummer des Eingangs (IN1-IN8) - Vergleiche mit der Zielposition aus der letzten Fahrt, die durch diesen Eingang ausgelöst wurde. Diese Position kann auch im Register "Result 1" des Eingangs ausgelesen werden. 255: wie 1-8, aber bitcodierte Rückmeldung: Welche IN-Position wird gerade gehalten? Siehe Beispiel unten.
	5	right position window: Zulässige Abweichung der Istposition in positiver Richtung
	6	left position window: Zulässige Abweichung der Istposition in negativer Richtung

HINWEIS: Über [Ausgangsfunktion 6 - Rückmeldung zu Position und Status](#)^[70] kann zusätzlich der Antriebszustand berücksichtigt werden.

BESCHREIBUNG ZU SUBINDEX 2 = 255 BITCODIERTE RÜCKMELDUNG

Für die bitcodierte Ausgabe müssen die Ausgangsfunktionen für **OUT1** und **OUT2** gleich parametrisiert werden:

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

Wenn **IN1** bis **IN3** für [Funktion 4 - Timed Motion Absolute](#)^[59] parametrier sind, können **OUT1** und **OUT2** gemeinsam eine Rückmeldung zur aktuellen Position geben:

OUT1	OUT2	Zustand
low	low	Keine der Positionen aus IN1-3 wird derzeit gehalten
high	low	Die Position aus IN1 wurde erreicht / wird gehalten
low	high	IN2 wurde erreicht / wird gehalten
high	high	IN3 wurde erreicht / wird gehalten

HINWEIS: Falls mehr als 2 Ausgänge verfügbar sind, können OUT1/2/3 wie oben parametrier werden und bis zu IN8 als Rückmeldung geben.

7.2.6 Funktion 5 - Antrieb ohne Fehler/Warnung

Allgemeine Statusrückmeldung:

Ausgang gesetzt / high: Normalzustand, es liegt keine Warnung oder Fehler vor

Ausgang zurückgesetzt / low: Im [CiA 402 Statusword 6041h](#)^[44] ist Bit 7 (Warnung) oder Bit 3 (Fehler) gesetzt, oder das [MovingCap Fehlerregister 1002h](#)^[47] hat mindestens ein Fehlerbit (gesetzt).

7.2.7 Funktion 6 - Rückmeldung Position und Status

Wie [Funktion 4](#)^[69], jedoch zusätzlich mit Überprüfung des Antriebsstatus:

- Antrieb in Positionierbetrieb, ohne [Fehler oder Warnung](#)^[70]
- Letzte Zielposition erreicht, d.h. "target reached" Bit im [Statusword](#)^[44] ist gesetzt

8 MovingCap CODE - Python Programmierung

MovingCap Antriebe mit CODE Unterstützung haben einen eingebauten [Micropython](#)-Interpreter um eigene kleine Programme für autonome Abläufe und Funktionserweiterungen dauerhaft im Antrieb zu hinterlegen.

HINWEIS: Bitte beachten Sie, dass ein im MovingCap dauerhaft hinterlegtes Programm bereits direkt nach Einschalten der Spannungsversorgung den Antrieb in Bewegung setzen kann.

Um den Antrieb dauerhaft zu stoppen, muss das Ablaufprogramm ebenfalls gestoppt werden, z.B. über die nachfolgend beschriebenen Funktionen der Weboberfläche.

HINWEIS: In MovingCap ist derzeit [MicroPython v1.9.4](#) eingebaut, jedoch werden die MicroPython-Module nicht vollständig unterstützt. Insbesondere gibt es derzeit keine Unterstützung für usocket Netzwerkfunktionen. Unterstützt werden unsere dokumentierten [MovingCap-spezifischen Module und Befehle](#) .

8.1 Ein einfaches Testprogramm

BEISPIEL FÜR MOVINGCAP TURNTRACK349 ETHERNET ODER ANDERE DREHANTRIEBE

Hier ist ein einfaches Beispiel für einen Drehmotor, der mit Winkelkoordinaten arbeiten soll (360° entsprechen einer Umdrehung):

```
#id driveSimplePosTest.py 2025-02-04 oh
# This is a simple positioning demo for a MovingCap turnTRACK 349 rotary
drive.

import sys
import mcdrive as mc

def WaitTargetReached():
    # Check statusword for "target reached"
    while (mc.ChkReady() == 0):
        sys.wait(1)
    # only continue if no error
    while (mc.ChkError() != 0):
        sys.wait(1)
    # check digital input IN2 - if low, add a second of pause/delay
    time
    if mc.ChkIn(2) == 0:
        sys.wait(1000)

# 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):
```

```

# set digital output OUT1 to high during move to position 0
mc.SetOut(1)
mc.GoPosAbs(0)
WaitTargetReached()
# reset OUT1 to low
mc.ClearOut(1)
mc.GoPosAbs(1000)
WaitTargetReached()
mc.GoPosAbs(180)
WaitTargetReached()

```

- Speichern Sie dieses Programm in einer .py-Datei, z.B. als **driveSimplePosTest.py** .
- Öffnen Sie in der [MovingCap-Weboberfläche](#) den Bereich **Code (Python)**
- Wählen Sie über **Browse..** diese Datei aus und klicken Sie auf den **Upload** Knopf rechts daneben.

The screenshot shows the MovingCap Web Interface with the 'Code (Python)' tab selected. The left menu includes 'Menu', 'Info', 'Click & Run', 'Servo Startup', 'Commands & Parameters', 'Code (Python)', and 'Network Settings'. The top navigation bar has tabs for 'MovingCap Eth', 'Click&Run', 'Servo', 'Command', 'Code', and 'Network'. The main content area displays a Python script for controlling a motor. The script includes comments and function calls like `mc.SetOut(1)`, `mc.GoPosAbs(0)`, `WaitTargetReached()`, `mc.ClearOut(1)`, `mc.GoPosAbs(1000)`, `WaitTargetReached()`, `mc.GoPosAbs(180)`, and `WaitTargetReached()`. The interface also shows buttons for 'ON', 'OFF [X]', and 'Store Permanently' at the top, and 'Upload' and 'Run' buttons for the script. A status bar at the bottom shows 'MC Live MOV', 'StatWd 0027h', 'OpMode 1 Python RUN', and various position and velocity data.

- Stellen Sie sicher, dass der Antrieb gefahrlos in Betrieb gehen kann und starten Sie das Programm mit dem Knopf **Run**

BEISPIELPROGRAMM FÜR EINE LINEARACHSE MIT VOREINGESTELLTER FAHRBEGRENZUNG (SOFTLIMIT)

Hier ein alternatives Demoprogramm, geeignet für einen Linearantrieb mit Mikrometer-Positionseinheiten. Die unten verwendete Geschwindigkeit `mc.SetPosVel (50000)` entspricht in diesem Fall 5cm/Sekunde.

```
#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)

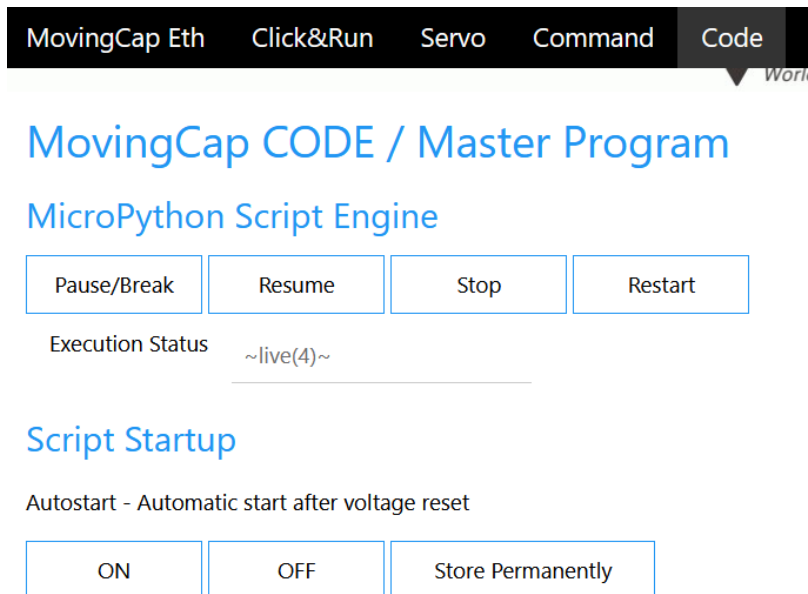
if ((softLimitMin == 0 and softLimitMax == 0) or softLimitMax <=
    (softLimitMin + 1000)):
    print("flatTRACK demo abort. Check softlimit objects 607Dh.1h and
    607D.2h!")
else:
    # initial wait, don't surprise me with immediate movement
    sys.wait(5000)

    # 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()
```

- Speichern Sie dieses Programm wie oben beschrieben in einer .py-Datei und führen Sie in Ihrem MovingCap flatTRACK Linearantrieb aus.

8.2 Python-Programmsteuerung / Automatischer Start

Das geladene Python-Programm kann über die Webseite gesteuert werden:



MicroPython Script Engine

Starten / Stoppen / Unterbrechen des derzeit geladenen Python-Programms

Script Startup

Python-Programm dauerhaft im Antrieb hinterlegen und nach Einschalten der Spannungsversorgung automatisch starten:

- Wählen Sie im Bereich Script Startup / Autostart -> **ON**
- Speichern Sie anschließend diese Einstellung und das Python-Programm dauerhaft mit **Store Permanently**

8.3 Python Benutzerparameter-Objekte

Für MovingCap CODE Python-Anwendungen gibt es frei verwendbare Datenobjekte, die wie alle anderen Antriebsparameter geschrieben / gelesen / permanent gespeichert werden können.

Typ	Index	Subindex-Bereich	Namensmuster	Beschreibung
unsigned 8	340B	01h-0Ah	Python unsigned8 N	8-Bit vorzeichenlose Ganzzahlvariablen (N = 1..10)
integer8	340C	01h-0Ah	Python integer8 N	8-Bit vorzeichenbehaftete Ganzzahlvariablen (N = 1..10)
unsigned 16	340D	01h-0Ah	Python unsigned16 N	16-Bit vorzeichenlose Ganzzahlvariablen (N = 1..10)
integer16	340E	01h-0Ah	Python integer16 N	16-Bit vorzeichenbehaftete Ganzzahlvariablen (N = 1..10)
unsigned 32	340F	01h-0Ah	Python unsigned32 N	32-Bit vorzeichenlose Ganzzahlvariablen (N = 1..10)
integer32	3410	01h-0Ah	Python integer32 N	32-Bit vorzeichenbehaftete Ganzzahlvariablen (N = 1..10)

Ergänzend oder alternativ können für vordefinierte Fahrprofile mit Positionen, Geschwindigkeiten usw. die [Parameterobjekte der Eingangsfunktionen](#)^[56] genutzt werden, auch wenn dort keine auszuführende Eingangsfunktion definiert ist.

HINWEIS: Alle Parameter (CiA 402 Standardparameter oder MovingCap-spezifische Parameterbereiche wie hier beschrieben) können über die

- die [Weboberfläche](#)^[17] und [REFGO](#)^[21]-Befehle **OW** und **OR**,
- den [Kickdrive-Objekteditor](#)^[20],
- oder in CODE/Python über die **mcdrive**-Befehle **WriteObject** und **ReadObject**

geschrieben bzw. gelesen werden.

PERMANENTES SPEICHERN ODER RÜCKKEHR ZUR WERKSEINSTELLUNG ÜBER PROGRAMMCODE

Parameteränderungen werden in einem CiA 301 kompatiblen Gerät zunächst nur im flüchtigen Speicher behalten, und nach Aus/Wiedereinschalten auf die vorher bekannte Einstellung zurückgesetzt.

Für dauerhafte Änderungen des Permanentenspeichers (nichtflüchtiger Speicher / NVM / EEPROM) stehen die CiA 301 Standardobjekte 1010h und 1011h zur Verfügung.

Folgende Operationen für den Permanentenspeicher werden durch Schreiben des entsprechenden Objektwerts ausgelöst:

Index	Sub-index	Typ	Wert (Hex)	Wert (Text)	Beschreibung
1010h	01h	unsigned 32	0x65766173	"save"	Speichert die aktuellen Parametereinstellungen und das Python-Skript dauerhaft im nichtflüchtigen Speicher (NVM/EEPROM). Ausführung über Weboberfläche / RefGo-Kommando: OW1010,1,1702257011 Entspricht dem Weboberflächen-Knopf Store Permanently bzw. der Kickdrive Object Editor-Funktion Store.
1011h	01h	unsigned 32	0x64616663	"load"	Setzt die aktuellen Parameter und den nichtflüchtigen Speicher auf die Werkseinstellung zurück. Löscht das Python-Skript. Ausführung über Weboberfläche / RefGo-Kommando: OW1011,1,1684107116 Entspricht dem Weboberflächen-Knopf Factory Reset bzw. der Kickdrive Object Editor-Funktion Reset.

HINWEIS: Durch das Ausführen von "load" / "Factory Reset" gehen alle anwendungsspezifischen Einstellungen und ggf. auch geänderte Reglereinstellungen verloren. Stellen Sie sicher, dass sie Einstellungsänderungen ggf. vorab gesichert haben, z.B. durch Auslesen über den [Kickdrive-Objekteditor](#).

8.4 Beispiele und Dokumentation zu den MovingCap-Befehlen

TIPP: Alle Dokumentationsdateien (.pyi-Dateien) und Beispielprogramme (.py-Datei) finden Sie im [movingcap.de Service-Portal](#), dort nach **CODE** suchen.

MICROPYTHON-GRUNDBEISPIELE FÜR MOVINGCAP ETH

Beispielprogramm	Erklärung
mcdriveModuleOverview.py	Demonstriert jeden einzelnen Befehl des mcdrive -Moduls in einem funktionsfähigen Testprogramm für den MovingCap-Antrieb
driveSimplePosTest.py	Eine einfache Positionieranwendung

PYTHON-MODULE FÜR MOVINGCAP

Um die MovingCap-spezifischen Antriebs- und Kommunikationsfunktionen zu nutzen, stehen in MovingCap CODE verschiedene Python-Module zur Verfügung. Diese müssen am Anfang Ihres Python-Programms über den **import**-Befehl eingebunden werden.

Python-Modul / Importbefehl	.pyi Dokumentationsdatei *1)	Erklärung
import mcdrive.py oder den Namen abkürzen als: import mcdrive.py as mc	mcdrive.pyi Alternativen in anderer Sprache: de/mcdrive.pyi fr/mcdrive.pyi it/mcdrive.pyi es/mcdrive.pyi	MovingCap-Antriebsfunktionen zum Parametrieren und Bewegen des Antriebs, sowie Schreiben und Lesen von CiA 402 -Objekten. Die zugehörige .pyi-Datei zeigt Definitionen und Beschreibungen aller verfügbaren Befehle. TIPP: Um die Dokumentation in einer anderen Sprache als Englisch zu erhalten, kopieren Sie die mcdrive.pyi-Variante aus dem passenden Unterverzeichnis.
import mcom	mcom.pyi	MovingCap-Kommunikationsfunktionen für Netzwerk. Bietet derzeit eine rudimentäre Möglichkeit, den RefGo TCP Port 10001 mit weiteren/eigenen Kommandos zu erweitern. Geplante Erweiterung: frei programmierbare TCP- und UDP-Verbindungen für applikationsspezifische Protokolle und Funktionen. TIPP: Beispiele im movingcap.de Service-Portal , dort nach refGoExtension suchen.
import refgo	refgo.pyi	Akzeptiert RefGo-Befehle in Textform, als Alternative zum mcdrive -Modul. Beispiel: Mit refgo.cmd("TS") können Sie den Antriebsstatus abfragen. TIPP: Beispiel im movingcap.de Service-Portal , dort nach UsingRefGoCommandsFromPython suchen.

*1) **.pyi**-Datei (Python Interface Definition) zur Unterstützung der Programmerstellung mit einem intelligenten Editor auf dem PC.

TIPP: Verwenden Sie zum Bearbeiten des Python-Programms [Visual Studio Code](#) auf Ihrem PC. Kopieren Sie die gewünschte Sprachvariante der **mcdrive.pyi**-Datei in den gleichen Ordner, in dem Sie Ihr **.py**-Programm erstellen. Visual Studio Code (oder ein anderer cleverer Editor) erkennt das **import mcdrive** Kommando in Ihrem Programm, wertet die dazu passende **mcdrive.pyi**-Datei aus und zeigt Ihnen ab jetzt zu jedem **mcdrive...**-Befehl die zugehörige Dokumentation an:

```

1  #id driveSimplePosTest.py 2023-03-10 oh
2  # This is a simple positioning demo, it assumes a MovingCap turnTRACK
3  # with user defined coordinates, e.g. positioning unit = ° (degrees)
4  import sys
5  import mcdribe as mc
6
7  def ChkReady():
8      # Check statusword for "target reached"
9      while (mc.ChkReady() == 0):
10         sys.wait(1)
11
12     # o (function) def GoPosAbs(targetPos: int) -> Any
13     whi Start new movement to an absolute position.
14
15     • If required, switch to operation mode 1 - profile positioning mode (SetOpMode(1))
16     • Set the new target (see SetTargetPos )
17     • Start positioning using DS402 "single setpoint" mode: the new target position is processed immediately.
18
19     The positioning control uses "6083h.0h profile acceleration" ( SetAcc ) and "6084h.0h profile deceleration" (
20     movement.
21
22     Use ChkReady and ChkError to wait for end of the positioning and detect errors during the run.
23
24     See also 'GoPosRel'
25
26     :param targetPos: new absolute target position [integer32]
27     :type targetPos: int
28     while(1
29         mc.GoPosAbs(0)
30         ChkReady()
31         mc.GoPosAbs(1080)
32         ChkReady()
33         mc.GoPosAbs(360)
34         ChkReady()

```

8.5 Debugging / Fehler finden

PYTHON-AUSGABEN UND FEHLERMELDUNGEN ÜBER UDP PORT 14999

Python-Ausgaben und Fehlermeldungen können über eine Netzwerkverbindung auf UDP Port 14999 erhalten werden.

In Windows können Sie hierzu wie folgt vorgehen:

- Laden Sie [Docklight Scripting](#) herunter und installieren Sie es. Sie benötigen keinen Lizenzschlüssel für einfache Tests / Anzeigen.

- Gehen Sie zum [movingcap.de Service-Portal](https://movingcap.de/Service-Portal), suchen Sie nach **Docklight_MovingCap_ETH_Debug_Output.zip**, laden Sie diese Datei herunter und entpacken Sie sie.
- Starten Sie Docklight Scripting und öffnen Sie das Projekt **Docklight_MovingCap_ETH_Debug_Output.ptp**
- Beachten Sie die in Docklight Scripting angezeigte Projektdokumentation (aus der Datei 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 - Highspeed-Positionierung für Linearantriebe

Für die MovingCap flatTRACK-, FATtrack- und shortTRACK-Baureihe steht ein neuer Positionsregler ("Rabbit Pos Control") zur Verfügung. Dieser nutzt die mechanischen und elektrischen Vorteile der linearen Direktantriebe optimal aus, und wurde für Anwendungen mit hohen Dynamik- und Genauigkeitsanforderungen entworfen.

ÜBERSICHT UND UNTERSCHIEDE DER MOVINGCAP-POSITIONSREGLERSYSTEME

Positionsregler	MovingCap Regler Standard	MovingCap Regler Highspeed ("Rabbit Pos Control")
Produktvariante	MovingCap turnTRACK Ethernet TCP/IP (Rotativantriebe MC349 ETH, MC632/634/636 ETH, MCN23 ETH) MovingCap pushTRACK Ethernet TCP/IP (Kurzhubzylinder pushTRACK 45S100 - 115S240)	MovingCap flatTRACK Ethernet TCP/IP (Lineare Direktantriebe flatTRACK 100-650, FATtrack 200) MovingCap shortTRACK Ethernet TCP/IP (Kompakter linearer Direktantrieb shortTRACK 046)
Regler-Architektur	3-stufige Kaskadenregelung aus PI-Reglern 1. Position 2. Geschwindigkeit 3. Strom/Drehmoment	2-stufige Kaskadenregelung 1. Zustandsregler für Position und Geschwindigkeit + schaltender Integrator für statische Positionsfehlerkorrektur 2. PI Stromregler
Betriebsarten	Positionierbetrieb (Profile Position Mode) Drehzahlbetrieb (Velocity Mode) Referenzierbetrieb (Homing Mode)	Positionierbetrieb (Profile Position Mode) <i>(Drehzahlbetrieb in Vorbereitung)</i>
Regler-Eingangsgroßen	Position oder Geschwindigkeit	Position, Geschwindigkeit und Beschleunigung
Trajektorien-generator	Trapez-Geschwindigkeitsprofil	S-förmiges Geschwindigkeitsprofil

Trajektorien-Parameter	Maximale Geschwindigkeit (Profile Velocity), Beschleunigung (Profile Acceleration), Bremswert (Profile Deceleration)	Maximale Geschwindigkeit (Profile Velocity), Maximale Beschleunigung (Profile Acceleration, gleich wie Bremswert), Ruck / Beschleunigungsänderung (Profile Jerk)
Regler-Einstellung	Voreinstellungen für stabilen Betrieb für die meisten Grundanwendungen Einstellparameter Positionsregelung: Proportionalverstärkung (Kp Pos) Integralverstärkung (Ki Pos) Einstellparameter Geschwindigkeitsregelung: Proportionalverstärkung (Kp Vel) Integralverstärkung (Ki Vel)	Vorgabe der bewegten Nutzlast (Payload) in Gramm [g] Einstellbare Gewichtskompensation / Kompensation von statischen Zusatzkräften Feinabstimmung: Umschaltbare Lineare oder Wurzel-Positionsregelung Einstellbare Kompensation der Motor-Phasen-Unsymmetrie ("Torque Ripple Suppression") Feinabstimmung durch Anpassung der Regelverstärkungen (Pos Gain, Vel Gain, Err Gain)

9.1 MovingCap RABBIT - Betriebsarten und Einstellungen

Der MovingCap Rabbit-Highspeed-Regler unterstützt im derzeitigen Softwarestand den [Positionierbetrieb](#)^[30] und die zugehörigen [CiA 402 Betriebszustände](#)^[28]. Die Unterschiede zum MovingCap Standard-Regler sind wie folgt:

- Nur [Positionierbetrieb](#)^[30]. ([Drehzahlbetrieb](#)^[35] ist derzeit nicht verfügbar.)
- Als Referenziermethode wird nur das [Setzen der Nulllage über REF35 unterstützt](#)^[18].
- Neuer Parameter **60A4h.0h profile jerk** für die zulässige Beschleunigungsänderung / Ruck.

HINWEIS: Es ist noch kein spezifischer Befehl für die Jerk-Einstellung im [Python mcdrive-Modul](#)^[76] vorhanden. Verwenden Sie z.B. `mc.WriteObject(0x60A4h, 0h, 1000000)` um den Jerk-Wert auf 1000000 zu setzen.

HINWEIS: Die Jerk-Einstellung ist auch noch nicht über [IO-Eingangsfunktion 4 Timed Motion Absolute](#) ^[59] oder Eingangsfunktion 6 Timed Motion Relative parametrierbar.

- Beim Schreiben von **6083h.0h profile acceleration** wird **6084h.0h profile deceleration** auf den gleichen Wert gesetzt. Ein nachträglich geänderter Wert für **deceleration** wird nur bei Stop Motion / Schnellstop mit [605Ah.0h quick stop option code 1](#) ^[34] oder **605Ah.0h quick stop option code 5** genutzt.

MOVINGCAP RABBIT TRAJEKTORIENGENERATOR

Für die Fahrprofile bei Punkt-Zu-Punkt Positionierung verwendet MovingCap RABBIT folgende Fahrparameter:

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
607Ah	0h	integer 32	R/W	target position [Positionseinheiten] ^[18] / user defined units]	Zielposition
6081h	0h	unsigned 32	R/W	profile velocity [Positionseinheiten] ^[18] /s]	Maximale Geschwindigkeit
6083h	0h	unsigned 32	R/W	profile acceleration [1000x Positionseinheiten] ^[18] /s ²]	Maximale Beschleunigung
60A4h	0h	unsigned 32	R/W	profile jerk *1) [1000x Positionseinheiten] ^[18] /s ³]	Beschleunigungsänderung / Ruck Empfohlener Startwert ist z.B. 50x größer als die Beschleunigung - dadurch wird innerhalb von 20 Millisekunden die volle Beschleunigung erreicht.

Folgende Randbedingungen werden bei der Berechnung der Trajektorie berücksichtigt:

- Um die vorgegebene Beschleunigungsgrenze einhalten, wird die maximale Geschwindigkeit reduziert, falls der Fahrweg zu kurz ist.
- Um die vorgegebene Beschleunigungsgrenze zu erreichen, wird ein niedriger Jerk/Ruck-Wert ggf. soweit erhöht, dass zumindest eine reine S-Kurve (ohne konstante Beschleunigungsphase) möglich ist.

MOVINGCAP RABBIT REGELPARAMETER

HINWEIS: Für die Regelparameter werden ab Werk stabile Voreinstellungen für Standard-Anwendungen bereitgestellt, es muss jedoch vor Betrieb mindestens Parameter **3101h.04h mass_payload** eingestellt werden.

HINWEIS: Bei Ausführung von **Factory Reset** (z.B. über die **Parameters** Webseite) werden wieder die werksseitigen Voreinstellungen gesetzt und die Payload wird auf 0 zurückgesetzt.

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
3301h	04h	unsigned16	R/W	mass_payload [g]	Additional User/Payload weight [g] Zusätzlich zum Motorschlitten bewegtes Gewicht / Nutzlast in Gramm Dies ist der einzige Parameter, der für jede Anwendung passend eingestellt werden muss. Der eingestellte Wert muss nicht exakt sein, und kann z.B. auch höher gewählt werden, um auf einfachem Weg die Steifigkeit der statischen Lageregelung zu erhöhen.
3301h	01h	unsigned16	R/W	control_mode	Controller Mode / Betriebsmodus des Zustandsreglers: 1 - Rabbit / Hase - lineare Positionsregelung, max. Geschwindigkeit 2 - Hedgehog / Igel - Quadratwurzel-Positionsregelung in der Nähe der Endposition, max. Endpositionsgenauigkeit. Verringern/Verhindern von Überschwinger bei Betrieb mit hohen Lasten / Zusatzkräften / Reibung. HINWEIS: Bei shortTRACK-Antrieben ohne Zusatzmasse instabil. 3 - Float / Schweben - Abstimmungsmodus, nur Schwerkraft- und Asymmetriekorrektur ist aktiv.

3301h	0Eh	integer 16	R/W	deceleration_tuning	Verzögerung/Brems-Tuning [0,1 %], z. B. negative Werte, z.B. "-300" - eine 30% kleiner Bremswert-Vorgabe verwenden, um ein Überschießen der dynamischen Bewegung zu korrigieren. positive Werte, z.B. "300" - 30% mehr Bremskraft verwenden, um eine zu geringe dynamische Bewegung zu korrigieren.
3301h	0Fh	integer 16	R/W	weight_compensation 1*) [g] oder +1/-1	Kraft-Vorsteuerung zur Einbaulage-/Gewichtskompensation: +1 - Schwerkraft zieht den Motorschlitten und die Nutzlast/Payload in positive Richtung. -1 - Schwerkraft zieht in negative Richtung anderer beliebiger Zahlenwert: Kompensation einer Kraft, die dem Gewicht von x Gramm entspricht.
3301h	08h	integer 16	R/W	k2 err gain	Regler-Verstärkung für den integrierten (aufsummierten) verbleibenden Positionsfehler in der Endlage größere Werte - höhere Steifigkeit und Genauigkeit, aber weniger stabil / zunehmende Schwingungsneigung
3301h	06h	integer 16	R/W	k0 pos gain	Regler-Verstärkung für die Positionsdivergenz (linear oder Wurzelfunktion, je nach Betriebsmodus 3301h.01h und Größe der Positionsdivergenz)
3301h	07h	integer 16	R/W	k1 vel gain	Regler-Verstärkung für die Geschwindigkeitsdivergenz größere Werte - Überspringen korrigieren / niederfrequentes Schütteln/Pendeln in der Endlage beheben

					niedrigere Werte - schnellere Resonanz/Brummen in der Endlage beheben
--	--	--	--	--	---

1*) Um 3301h.0Fh **weight_compensation** manuell einzustellen, setzen Sie **3301h.01h control mode = 3 float**, schalten Sie den Positionierbetrieb ein und starten Sie eine Positionsfahrt. Der Antrieb bleibt weiter im 'float'-Betrieb, aber ab jetzt werden die **3301h.0Fh weight_compensation** Werte direkt als Kraft umgesetzt. Nutzen Sie z.B. eine einfache digitale Kofferraumwaage, um die realisierte Kompensationskraft in der Größenordnung zu bestätigen. Die Kraft/Gewichtswerte entsprechen nur ungefähr der tatsächlich realisierten Kraft am Motor und eignen sich nicht für Messanwendungen.

HINWEIS: Als Erweiterung sind zusätzliche Kalibrierungs-/Einstellmöglichkeiten in Planung, um eine genauere Korrelation von Strom- und Kraftwerten realisieren zu können.

MOVINGCAP RABBIT MOTORPARAMETER

Motorparameter werden in der Regel einmalig bei der Antriebsfertigung eingestellt. Sie bleiben auch bei einem **Factory Reset** erhalten und werden nicht auf einen Standardwert zurückgesetzt.

Index	Subindex	Typ	Zugriff	Name [Einheiten]	Beschreibung
3300h	07h	integer 16	R/W	motor_weight 1*) [g]	Das Grundgewicht des beweglichen Motorschlittens
3300h	04h	unsigned 16	R/W	motor_imbalance_amplitude 2*) [0.1% of rated current]	Korrektur der Phasen-Unsymmetrie / "Torque Ripple Suppression" Amplitude/Höhe der Unsymmetrie in 0.1% des Nennstroms. D.h. 10 = 1% des Nennstroms. Entspricht der Skalierung der Strom-/Kraftbegrenzung ³⁹⁾ .
3300h	05h	integer 16	R/W	motor_imbalance_angle 2*) [°]	Der mechanische Winkel des Motors im Bereich (-180°, +180°) relativ zum Einphasungspunkt, bei dem der minimale Motorstrom aufgrund der Unsymmetrie auftritt.

1*) Wie bei der **3301h.04h mass_payload** Einstellung muss der Wert nicht exakt sein. Der Regler berücksichtigt immer die Summe aus **3300h.07h motor_weight** und **3301h.04h mass_payload**. Für flatTRACK und FATtrack-Anwendungen, bei denen der Motorschlitten feststeht und der Fahrweg beweglich ist, wird folgende Einstellung empfohlen: Beibehaltung des ursprünglichen

motor_weight-Werts, und Einstellen von **3301h.04h mass_payload** = Gesamtgewicht des beweglichen Aufbaus incl. MovingCap-Fahrschiene, minus **motor_weight**.

2*) Die Korrekturwerte dienen der zusätzlichen Betriebsoptimierung bei den flatTRACK und FATTrack Motorvarianten. Auch mit Grundeinstellung **3300h.04h motor_imbalance_amplitude = 0** arbeitet der Regler. Der Einfluss des "Torque Ripple" kann sich aber bei schnellen Fahrten und der Endpositionsgenauigkeit bemerkbar machen. Für die manuelle Ermittlung der passenden Einstellungen, benutzen Sie **3301h.01h control mode = 3 float** und schalten Sie den Positionierbetrieb ein. Falls möglich und unter Beachtung von Sicherheitsmaßnahmen, bewegen Sie den Motorschlitten mit der Hand hin und her und achten Sie auf eine sich wiederholende Unregelmäßigkeit/Zugkraft die den Schlitten leicht in bestimmte Pollagen zieht. Experimentieren Sie dann mit einem Startwert, z.B. **3300h.05h motor_imbalance_amplitude = 20** und unterschiedlichen **3300h.05h motor_imbalance_angle** Einstellungen, bis die Unregelmäßigkeit vollständig verschwindet. Für die automatische Durchführung von Kalibrierfahrten und automatische Ermittlung der Einstellwerte sind derzeit [MovingCap CODE - Python](#)^[70] Programme in Entwicklung.

ZUSÄTZLICHE PARAMETER FÜR SONDERANWENDUNGEN

Weitere Sonderparameter und Filter können anwendungsspezifisch eingestellt werden. Entweder über ein entsprechendes Einrichtprojekt für Kickdrive, ein Python-Konfigurationsskript, oder durch OW-Befehle über die [Weboberfläche](#)^[25] oder [Terminal/TCP-Verbindung](#)^[25]. Bitte setzen Sie sich ggf. mit unserem Support in Verbindung.

VOLLSTÄNDIGE GRUND-INITIALISIERUNG, Z.B. NACH REPARATUR/UMBAU

Für die vollständige Neu-Initialisierung der MovingCap Linearantriebe (flatTRACK, shortTRACK, FATTrack) stehen auf unserem [movingcap.de Service-Portal](#), Unterverzeichnis **Init-Projects**, dokumentierte Einrichtprojekte zur Verwendung mit Kickdrive zur Verfügung. Mit diesen Projekten kann auch eine erneute Einphasung / Ermittlung der Motorphasenlage relativ zum Absolutmesssystem durchgeführt werden.



WARNUNG! Über diese Einricht-Projekte werden auch Einstellungen zum Motor-Überlastschutz neu geschrieben. Führen Sie die Grundeinrichtung nur nach Rücksprache mit unserem Support durch.

10 CiA 402 - Allgemeine Übersicht

EINFÜHRUNG

CiA 402, alternativ auch als **DS 402** / Device Standard 402 bezeichnet, ist ein Geräteprofil für Antriebe und Bewegungssteuerung, das von der CAN in Automation (CiA)-Organisation definiert wurde. Es spezifiziert ein universelles Interface und standardisiertes Verhalten zur Steuerung elektrischer Antriebe — wie Servoantriebe, Servoregler, Frequenzumrichter und ähnliche Geräte. Es ist als "CANopen Device Profile" Bestandteil des CANopen-Protokolls, findet jedoch auch in EtherCAT, POWERLINK- und anderen Applikationsprotokollen wie TCP/IP zum Einsatz. Das CiA 402 Geräteprofil ist in der Norm **IEC61800-7-201** spezifiziert als Profil Typ 1 - **CiA402 Drive Profile for power drive systems (PDS)**.

Dieser Abschnitt bietet einen Überblick über die zentralen Aspekte von CiA 402, mit Fokus auf die Antriebszustände / **CiA 402 State Machine**, das Haupt-Controlword (6040h) und Statusword (6041h), gängige Betriebsmodi (6060h) sowie eine Zusammenfassung weiterer wesentlicher Standardobjekte des Objektverzeichnisses / Object Dictionary.

DIE CIA 402 STATE MACHINE

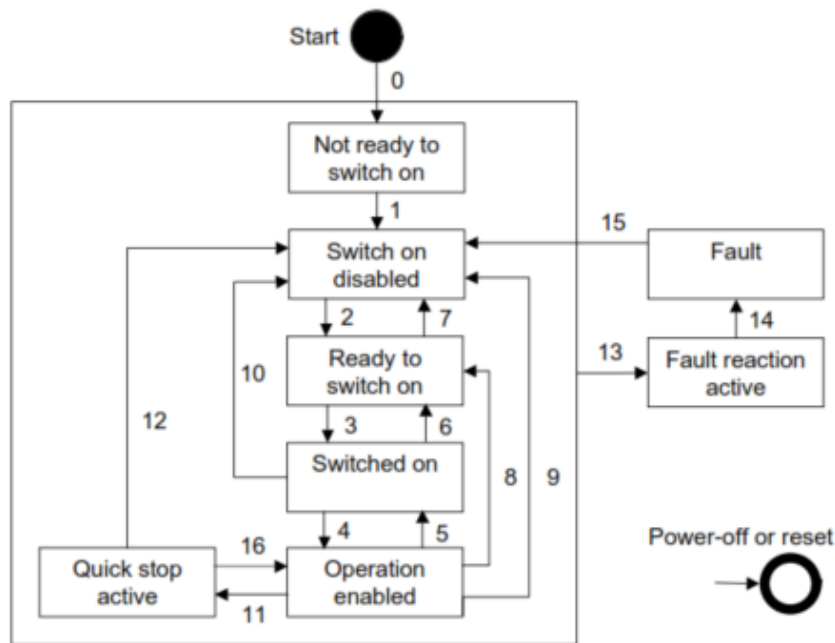
Im Zentrum des CiA 402 Profils steht die State Machine (**PDS FSA, Power Drive System - Finite State Automat**), welche die zulässigen Zustände und Übergänge eines Antriebs definiert. Die State Machine sorgt durch die Definition erlaubter Aktionen und erforderlicher Reaktionen in verschiedenen Betriebszuständen für eine sichere und vorhersehbare Steuerung von Bewegungsachsen.

HAUPTZUSTÄNDE

- **Not Ready to Switch On:** Anfangszustand nach dem Einschalten. Der Antrieb ist nicht bereit, um Bewegungen auszuführen.
- **Switch On Disabled:** Antrieb darf nicht freigeschaltet werden (z.B. nach Fehler oder während Initialisierung).
- **Ready to Switch On:** Interne Prüfungen abgeschlossen; Antrieb kann freigeschaltet werden.
- **Switched On:** Antrieb ist bereit für Enable-Befehl.
- **Operation Enabled:** Der Antrieb kann Bewegungsbefehle ausführen.
- **Quick Stop Active:** Quick Stop ist aktiviert, der Antrieb stoppt so schnell wie möglich.
- **Fault Reaction Active:** Der Antrieb reagiert auf einen erkannten Fehler—er führt entsprechende Sicherheitsmaßnahmen durch.
- **Fault:** Ein Fehlerzustand wurde erkannt; der Antrieb muss zurückgesetzt werden.

ÜBERGÄNGE

Zustandsübergänge werden von Bits im Controlword (6040h) gesteuert, das vom Master an das Gerät gesendet wird. Der Antrieb meldet seinen aktuellen Zustand über das Statusword (6041h) zurück.



Eine detaillierte Beschreibung ist in der CiA 402 Spezifikation enthalten.

CONTROLWORD (6040H OBJEKT)

Das **Controlword** ist das zentrale Steuerobjekt (Index 6040h), mit dem der Master den Zustand und das Verhalten des Antriebs steuert. Es handelt sich um ein 16-Bit-Wort, wobei jedes Bit (bzw. Bitgruppen) bestimmte Aktionen oder Anforderungen am Antrieb auslösen, wie Zustandswechsel oder Bewegungsbefehle.

WICHTIGE BIT-FUNKTIONEN

Bit	Name	Beschreibung
0	Switch On	Fordert das Einschalten des Antriebs an
1	Enable Voltage	Schaltet die interne Spannungsversorgung frei
2	Quick Stop	Fordert schnelles Stoppen der Bewegung an
3	Enable Operation	Vollständige Freischaltung der Achse
7	Fault Reset	Fehler quittieren/zurücksetzen

8	Halt	Bewegung anhalten ohne den Antrieb zu deaktivieren
9	Operation Mode Specific	Wird in bestimmten Betriebsmodi verwendet
10	Reserved	-
11-15	Hersteller-spezifisch	-

Die Kombination und zeitliche Abfolge dieser Bits bestimmen exakt, wie die State Machine zwischen Zuständen wechselt. Zum Beispiel: Für den Wechsel von „Switch On Disabled“ zu „Operation Enabled“ müssen die Bits 0, 1 und 2 gesetzt werden, anschließend Bit 3.

STATUSWORD (6041H OBJEKT)

Das **Statusword** (Index 6041h) ist ein 16-Bit-Wort, das vom Antrieb an den Master gesendet wird und den aktuellen Status signalisiert. Durch das Auslesen des Statuswords erkennt der Master den aktuellen Zustand des Antriebs in der CiA 402 State Machine.

REPRÄSENTATIVE BIT-BEDEUTUNGEN

Bit	Name	Beschreibung
0	Ready to Switch On	Antrieb ist bereit zum Freischalten
1	Switched On	Antrieb ist freigeschaltet
2	Operation Enabled	Antrieb ist für Bewegungen freigegeben
3	Fault	Fehler liegt vor
4	Voltage Enabled	Interne Spannung liegt an
5	Quick Stop	Quick Stop Funktion ist aktiv
6	Switch On Disabled	Antrieb ist deaktiviert
7	Warning	Warnung liegt vor (kein Fehler)
8	Manufacturer Specific	*1)

9	remote	
10	target reached	Position ist innerhalb der Ziel-Toleranz
11	internal limit active	Grenze laut 607D Objekten erreicht
12,13	Operation Mode specific	-
14,15	Hersteller-spezifisch	-

Hinweis:

*1) MovingCap und Festo-Antriebe nutzen dieses Bit für den Zustand „Drive Moving“.

BETRIEBSARTEN / MODES OF OPERATION (6060H OBJEKT)

Operation mode (Index 6060h) bestimmt den Funktionsmodus, in dem der Antrieb arbeitet. Jeder Modus unterstützt ein anderes Steuerparadigma (z.B. Positionsregelung, Geschwindigkeitsregelung, Drehmomentregelung).

6060H MODES OF OPERATION

Wert (Dezimal)	Mode Name	Anwendung
-128 bis -1	Hersteller-spez. Modus	Hersteller-spezifisch
0	Kein Modus zugeordnet	-
1	Profile Position Mode	Fährt Position mit definierten Profil
2	Velocity Mode	(veraltet, meist nicht unterstützt, siehe Mode 3)
3	Profile Velocity Mode	Fährt Geschwindigkeit mit Profilbeschleunigung
4	Profile Torque Mode	Steuert Drehmoment gemäß Profil
6	Homing Mode	Referenzfahrt ("Home"-Position anfahren)

7	Interpolated Position Mode	Fährt nach interpolierten Positionspunkten
8	Cyclic Synchronous Position	Synchrone Positionsregelung (Echtzeit)
9	Cyclic Synchronous Velocity	Synchrone Drehzahlregelung (Echtzeit)
10	Cyclic Synchronous Torque	Synchrones Drehmoment (Echtzeit)

Ein Servoantrieb unterstützt in der Regel nur einige der Betriebsmodi gemäß CiA 402. Häufig verwendete Modi sind

- 1 = Profile Position Mode,
- 3 = Profile Velocity Mode,
- 6 = Homing Mode,
- 8 = Cyclic Synchronous Position Mode.

Der aktuelle Modus kann mit **Modes of operation display** (6061h) ausgelesen werden.

PROFILE POSITION MODE

Profile Position Mode (PPM, Wert 1 für 6060h) ist einer der grundlegenden Positionsregelungsmodi im CiA 402. In diesem Modus bewegt der Antrieb seine Last zu einer Zielposition entlang eines Bewegungsprofils. Dabei können Parameter wie Geschwindigkeit, Beschleunigung und Verzögerung eingestellt werden, was sanfte und kontrollierte Bewegungen ermöglicht. Der Modus eignet sich für Punkt-zu-Punkt-Positionierung, Ablaufbewegungen oder Indizierungsaufgaben.

ABLAUF

1. Betriebsmodus (6060h) auf 1 setzen (Profile Position Mode).
2. **Target position** (607Ah) und gewünschte Profilparameter setzen.
3. **Profile velocity** (6081h), **profile acceleration** (6083h) und **profile deceleration** (6084h) konfigurieren.
4. Mit dem Controlword die Bewegung starten (z.B. durch Setzen des New Set-point Bits gemäß CiA 402).
5. Statusword und **Position actual value** (6064h) überwachen, um die Abwicklung zu bestätigen.

WICHTIGE OBJEKTE FÜR PROFILE POSITIONING

Index	Name	Beschreibung
6060h	Modes of Operation	Auf 1 setzen für Profile Position Mode
607Ah	Target Position	Zielposition
6081h	Profile Velocity	Maximal zulässige Geschwindigkeit
6083h	Profile Acceleration	Beschleunigungsrate
6084h	Profile Deceleration	Verzögerungsrate
6064h	Position Actual Value	Ist-Position
6040h	Controlword	Bewegt auslösen und Status verwalten
6041h	Statusword	Rückmeldung zum Zustand

Optional/zusätzliche Objekte:

- 60F2h: Positioning Option Code (z.B. "relative" oder "absolute")
- 6073h: Max Current – Max. Strom/Drehmoment/Kraft für den Antrieb
- 607Dh.01h/607Dh.02h: Software Position Limit – Software-Endschalter

PROFILE VELOCITY MODE

Profile Velocity Mode (PVM, Wert 3 für 6060h) ermöglicht es, die Achsgeschwindigkeit direkt vorzugeben, wobei profilierte Beschleunigungs- und Verzögerungsrampen eingehalten werden können. Typische Anwendungen finden sich beispielsweise bei Förderbändern, Lüftern oder anderen Systemen, in denen eine konstante Geschwindigkeit gefordert ist.

ABLAUF

1. Betriebsmodus (6060h) auf 3 setzen (Profile Velocity Mode).
2. **Target velocity** (60FFh) sowie gewünschte **profile acceleration/deceleration** (6083h/6084h) setzen.
3. Controlword verwenden, um Start, Stop oder Geschwindigkeitsänderungen auszulösen.

4. **Velocity actual value** (606Ch) und Statusword zur Überwachung auslesen.

WICHTIGE OBJEKTE FÜR PROFILE VELOCITY

Index	Name	Beschreibung
6060h	Modes of Operation	Auf 3 setzen für Profile Velocity Mode
60FFh	Target Velocity	Soll-Geschwindigkeit
6083h	Profile Acceleration	Beschleunigungsrampe
6084h	Profile Deceleration	Verzögerungsrampe
606Ch	Velocity Actual Value	Ist-Geschwindigkeit
6040h	Controlword	Steuerung Start/Stop/Halt
6041h	Statusword	Feedback und Zustandsüberwachung

WICHTIGE OBJECT DICTIONARY EINTRÄGE

Die folgende Tabelle enthält einige wichtige Einträge gemäß CiA 402 Device Profile, die für Antriebssteuerung und -überwachung zentral sind.

Index	Name	Beschreibung	Typ
6040h	Controlword	Steuer- und Zustandswechsel (Master → Antrieb)	unsigned16
6041h	Statusword	Statusrückmeldung (Antrieb → Master)	unsigned16
6060h	Modes of Operation	Setzt Betriebsmodus des Antriebs	integer8
6061h	Modes of Operation Display	Aktueller Betriebsmodus	integer8
607Ah	Target Position	Zielposition für Profile Position Mode	integer32

6064h	Position Actual Value	Positionsrückmeldung	integer32
606Ch	Velocity Actual Value	Geschwindigkeit (Istwert)	integer32
6081h	Profile Velocity	Max. Geschwindigkeit bei Profile Position Mode	unsigned32
6083h	Profile Acceleration	Beschleunigungsrampe	unsigned32
6084h	Profile Deceleration	Verzögerungsrampe	unsigned32
60FFh	Target Velocity	Soll-Geschwindigkeit im Profile Velocity Mode	integer32
6098h	Homing Method	Auswahl Homing-Strategie	integer8
607Dh.01h	Min position limit	Software-Endschalter Minimum	integer32
607Dh.02h	Max position limit	Software-Endschalter Maximum	integer32
6073h	Max Current	Max. Strom/Drehmoment/Kraft	unsigned16
6075h	Motor Rated Current	Nennstrom für Antrieb/Motor	unsigned32
6078h	Current Actual Value	Aktueller Motorstrom	integer16

POSITIONSSKALIERUNG / USER-DEFINED UNITS

Positionsskalierung in CiA 402 kombiniert konfigurierbare Skalierungsparameter mit festen Motor-/Encoder-Eigenschaften, um interne Positionseinheiten (Inkrement) in anwenderspezifische/technische Einheiten umzuwandeln.

MOTOR-/ENCODER-SYSTEMEIGENSCHAFTEN (608FH)

- **608Fh: Position Encoder Resolution**

- 608Fh.01 encoder_increments: Inkremente je Motordrehung

- 608Fh.02 motor_revolutions: Motordrehungen pro Encoderumdrehung
- Beispiel: Bei einem 16-bit Inkremental-Encoder: 608Fh.01 = 65.536, 608Fh.02 = 1 (üblicherweise 1, außer bei nicht 1:1-Getriebe zwischen Motor und Encoder)

HINWEIS:

Diese Werte spiegeln Hardware-Eigenschaften wider und sollten nur bei Hardwarewechsel verändert werden.

KONFIGURIERBARE SKALIERUNG (6091H UND 6092H)

- **6091h: Gear Ratio**

- 6091h.01h motor_revolutions: Anzahl Motordrehungen
- 6091h.02h shaft_revolutions Anzahl Abtriebsumdrehungen
- Beispiel: 5 Motordrehungen pro 2 Abtriebsumdrehungen → 6091h.01h = 5, 6091h.02h = 2

- **6092h: Feed Constant**

- 6092h.01h feed: Technische Einheit pro Abtriebsumdrehung (z.B. mm, µm)
- 6092h.02h shaft_revolutions: Anzahl Abtriebsumdrehungen
- Beispiel: Zahnriemenachse mit 100 mm pro Umdrehung, Skala µm: 6092h.01h feed = 100.000, 6092h.02h = 1

FORMELN ZUR POSITIONSSKALIERUNG

Größe	Formel (sprachlich)	Formel (mit CiA 402-Objekten)
Encoder resolution	$\text{Encoder_resolution} = \frac{\text{encoder_increments}}{\text{motor_rev}}$	608Fh.01h / 608Fh.02h
Gear ratio	$\text{Gear_ratio} = \frac{\text{motor_revolutions}}{\text{shaft_revolutions}}$	6091h.01h / 6091h.02h
Feed constant	$\text{Feed_constant} = \frac{\text{feed}}{\text{shaft_revolutions}}$	6092h.01h / 6092h.02h
Interne Positionseinheit	$\text{Pos_internal} = \frac{\text{Pos_user} \times \text{Encoder_resolution} \times \text{Gear_ratio}}{\text{Feed_constant}}$	$\text{Pos_user} \times \left(\frac{608Fh.01h}{608Fh.02h} \right) \times \left(\frac{6091h.01h}{6091h.02h} \right) / \left(\frac{6092h.01h}{6092h.02h} \right)$

		oder: $\text{Pos_user} \times (608\text{Fh}.01\text{h} \times 6091\text{h}.01\text{h} \times 6092\text{h}.02\text{h}) / (608\text{Fh}.02\text{h} \times 6091\text{h}.02\text{h} \times 6092\text{h}.01\text{h})$
Position als Anwendergröße	$\text{Pos_user} = \text{Pos_internal} \times \text{Feed_constant} / (\text{Encoder_resolution} \times \text{Gear_ratio})$	$\text{Pos_internal} \times (6092\text{h}.01\text{h} / 6092\text{h}.02\text{h}) / [(608\text{Fh}.01\text{h} / 608\text{Fh}.02\text{h}) \times (6091\text{h}.01\text{h} / 6091\text{h}.02\text{h})]$
		oder: $\text{Pos_internal} \times (608\text{Fh}.02\text{h} \times 6091\text{h}.02\text{h} \times 6092\text{h}.01\text{h}) / (608\text{Fh}.01\text{h} \times 6091\text{h}.01\text{h} \times 6092\text{h}.02\text{h})$

BEISPIEL

Gegeben:

- Motor-Encoder = 65.536 Inkremente je Motordrehung (608Fh.01h = 65536, 608Fh.02h = 1)
- Gear ratio: 1 Abtriebsumdrehung pro 5 Motordrehungen (6091h.01h = 5, 6091h.02h = 1)
- Feed constant: 100 mm pro Abtriebsumdrehung (6092h.01h = 100, 6092h.02h = 1)

Anwendereinheit / user-defined unit pro internen Encoder-Inkrement
 = (Feed constant) / (Gear ratio × Encoder resolution)
 = (100 mm/U) / (5 × 65.536 Inkremente/U)
 = 0,00030517578125 [mm / Inkrement]

Beispiel: Bewegung um 80.000 interne Inkremente
 --> $\text{Pos_user} = 80.000 \times 0.00030517578125 \approx 24,41 \text{ mm}$