

# Objektbeschreibung und Betriebsarten CANopen MovingCap

---

Version 1.1

**Datum:** 02. September 2013  
**Firma:** Fullmo GmbH  
**Autor:** Daniel Wetzel



**Fullmo GmbH**  
Robert-Bosch-Str. 12  
88677 Markdorf

Tel.: +49-(0)7 54 4 / 30 73 39-0  
Fax: +49-(0)7 54 4 / 30 73 39-9  
E-mail: [info@fullmo.de](mailto:info@fullmo.de)

## Inhaltsverzeichnis

<b>1</b>	<b>Controlling the power drive system.....</b>	<b>8</b>
1.1	Übersicht.....	8
1.2	Implementierte Objekte .....	8
1.2.1	Object 6040h: Controlword.....	8
1.2.2	Object 6041h: Statusword .....	9
1.2.3	Object 605Ah: Quick stop option code .....	9
1.2.4	Object 605Bh: Shutdown option code .....	9
1.2.5	Object 605Ch: Disable operation option code .....	9
1.2.6	Object 605Dh: Halt option code.....	9
1.2.7	Object 605Eh: Fault reaction option code.....	9
1.2.8	Object 6060h: Modes of operation.....	9
1.2.9	Object 6061h: Modes of operation display.....	9
<b>2</b>	<b>Profile Position Mode .....</b>	<b>11</b>
2.1	Übersicht.....	11
2.2	ControlWord/StatusWord.....	12
2.2.1	ControlWord.....	12
2.2.2	StatusWord .....	12
2.2.3	Single Setpoint.....	13
2.2.4	Set of Setpoints.....	13
2.3	Implementierte Objekte .....	13
2.3.1	Object 607Ah: Target position .....	13
2.3.2	Object 607Dh: Software position limit.....	13
2.3.3	Object 607Fh: Max profile velocity.....	14
2.3.4	Object 6080h: Max motor speed .....	14
2.3.5	Object 6081h: Profile velocity.....	14
2.3.6	Object 6083h: Profile acceleration.....	14
2.3.7	Object 6084h: Profile deceleration.....	14
2.3.8	Object 6085h: Quick stop deceleration.....	14
2.3.9	Object 60C5h: Max acceleration .....	14
2.3.10	Object 60C6h: Max deceleration .....	14

<b>3</b>	<b>Profile Velocity Mode</b>	<b>15</b>
3.1	Übersicht	15
3.2	ControlWord/StatusWord	16
3.2.1	ControlWord	16
3.2.2	StatusWord	16
3.3	Implementierte Objekte	16
3.3.1	Object 606Bh: Velocity demand value	16
3.3.2	Object 606Ch: Velocity actual value	16
3.3.3	Object 606Dh: Velocity window	17
3.3.4	Object 606Eh: Velocity window time	17
3.3.5	Object 606Fh: Velocity threshold	17
3.3.6	Object 6070h: Velocity threshold time	17
3.3.7	Object 60FFh: Target velocity	17
<b>4</b>	<b>Homing Mode</b>	<b>18</b>
4.1	Übersicht	18
4.2	ControlWord/StatusWord	19
4.2.1	ControlWord	19
4.2.2	StatusWord	19
4.3	Implementierte Objekte	19
4.3.1	Object 607Ch: Home offset	19
4.3.2	Object 6098h: Homing method	19
4.3.3	Object 6099h: Homing speeds	19
4.3.4	Object 609Ah: Homing acceleration	19
4.3.5	Object 60B8h: Touch probe function	20
<b>5</b>	<b>I/O Functions</b>	<b>21</b>
5.1.1	Objekt 0x3611 Ausgangsfunktion 1	21
5.1.2	Objekt 0x3612 Ausgangsfunktion 2	21
<b>6</b>	<b>Position control function</b>	<b>23</b>
6.1	Übersicht	23
6.2	Implementierte Objekte	24
6.2.1	Object 6063h: Position actual internal value	24
6.2.2	Object 6064h: Position actual value	24
6.2.3	Object 6067h: Position window	24
6.2.4	Object 6068h: Position window time	24

6.2.5	Object 60FCh: Position demand internal value.....	24
6.2.6	Object 60F2h: Positioning option code .....	24
<b>7</b>	<b>Herstellerspezifische Objekte.....</b>	<b>25</b>
7.1	Pythoninterpreter.....	25
7.1.1	Funktionen .....	25
7.1.1.1	ReadObjectByte(n, m).....	25
7.1.1.2	ReadObjectDword(n, m).....	25
7.1.1.3	ReadObjectWord(n, m) .....	26
7.1.1.4	SendEmcyMsg().....	26
7.1.1.5	WriteObjectByte(n, m, o).....	26
7.1.1.6	WriteObjectDword(n, m, o).....	27
7.1.1.7	WriteObjectWord(n, m, o).....	27
7.1.2	Pythonvariablen .....	28
7.1.3	8Bit Variablen.....	28
7.1.4	16Bit Variablen.....	28
7.1.5	16Bit Variablen.....	28
7.2	Objekt 0x3401 Regelparameter.....	28
7.2.1	Blockfahrterkennung .....	28
7.2.1.1	Objekt 0x3401 Subindex 1 .....	28
7.2.1.2	Objekt 0x3401 Subindex 2 .....	28
7.2.2	Objekt 0x3401 Subindex 3 .....	28
7.2.3	RS485.....	29
7.2.3.1	Objekt 0x3401 Subindex 4 .....	29
7.2.3.2	Objekt 0x3401 Subindex 5 .....	29
7.2.4	Objekt 0x3401 Subindex 0x15.....	29
7.2.5	Objekt 0x3401 Subindex 0x16.....	29
7.3	Option Codes .....	29
7.3.1	Objekt 0x6059 Subindex 0x01 Stroke option code .....	29
<b>8</b>	<b>Emergency Messages .....</b>	<b>30</b>

## Abbildungsverzeichnis

Abbildung 1: Controlling the power drive system [1] .....	8
Abbildung 2: Profile Position Mode .....	11
Abbildung 3: Profile Velocity Mode .....	15
Abbildung 4: Position reached .....	23

**Tabellenverzeichnis**

Tabelle 1: PPmode ControlWord ..... 12  
Tabelle 2: Objekt 0x3611 ..... 21  
Tabelle 3: Objekt 0x3612 ..... 21  
Tabelle 4: Fehler Codierung ..... 22  
Tabelle 5: Emergency Messages ..... 30

## **Definitionen, Akronyme, Abkürzungen**

## Referenzen



# 1 Controlling the power drive system

## 1.1 Übersicht

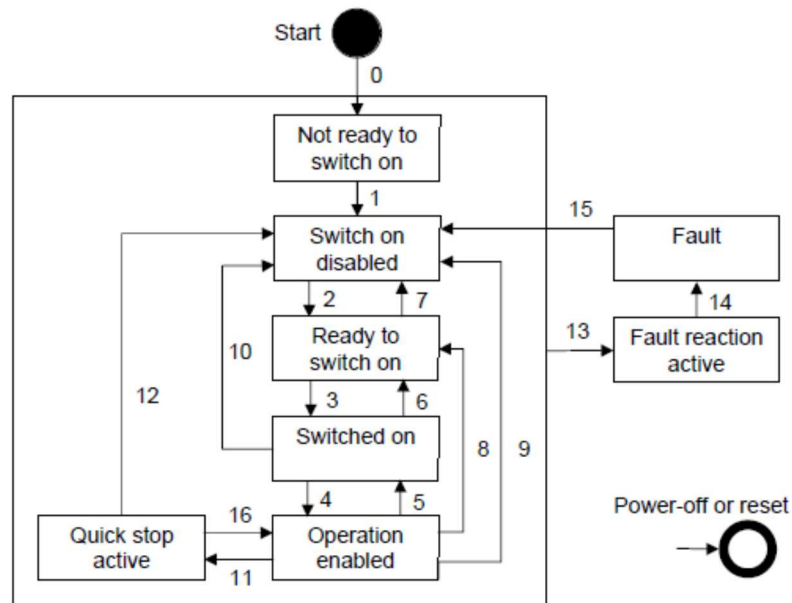


Abbildung 1: Controlling the power drive system [1]

Implementierte Zustände:

- Not ready to switch on
- Switch on disabled
- Ready to switch on
- Switched on
- Operation enabled
- Quick stop active
- Fault
- Fault reaction active

Implementierte Transitionen:

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

## 1.2 Implementierte Objekte

### 1.2.1 Object 6040h: Controlword

Das Objekt wurde für folgende Betriebsarten implementiert:

- Profile Position Mode

- Profile Velocity Mode
- Homing Mode

Es wurden keine herstellerspezifischen Anpassungen vorgenommen.

### **1.2.2 Object 6041h: Statusword**

Das Objekt wurde für folgende Betriebsarten implementiert:

- Profile Position Mode
- Profile Velocity Mode
- Homing Mode

Es wurden keine herstellerspezifischen Anpassungen vorgenommen.

### **1.2.3 Object 605Ah: Quick stop option code**

Das Objekt unterstützt die Option Codes 0 – 8.

Es wurden keine herstellerspezifischen Option Codes erstellt.

### **1.2.4 Object 605Bh: Shutdown option code**

Das Objekt unterstützt die Option Codes 0 – 1.

Es wurden keine herstellerspezifischen Option Codes erstellt.

### **1.2.5 Object 605Ch: Disable operation option code**

Das Objekt wurde unterstützt die Option Codes 0 – 1.

Es wurden keine herstellerspezifischen Option Codes erstellt.

### **1.2.6 Object 605Dh: Halt option code**

Das Objekt unterstützt die Option Codes 0 – 4.

Es wurden keine herstellerspezifischen Option Codes erstellt.

### **1.2.7 Object 605Eh: Fault reaction option code**

Das Objekt unterstützt die Option Codes 0 – 4.

Es wurden keine herstellerspezifischen Option Codes erstellt.

### **1.2.8 Object 6060h: Modes of operation**

Das Objekt unterstützt folgende Modi:

- 1: Profile Position Mode
- 3: Profile Velocity Mode
- 6: Homing Mode
- -1: Interner Service Mode (herstellerspezifisch)

### **1.2.9 Object 6061h: Modes of operation display**

Das Objekt unterstützt folgende Modi:

- 1: Profile Position Mode
- 3: Profile Velocity Mode
- 6: Homing Mode

- -1: Interner Service Mode (herstellerspezifisch)

## 2 Profile Position Mode

### 2.1 Übersicht

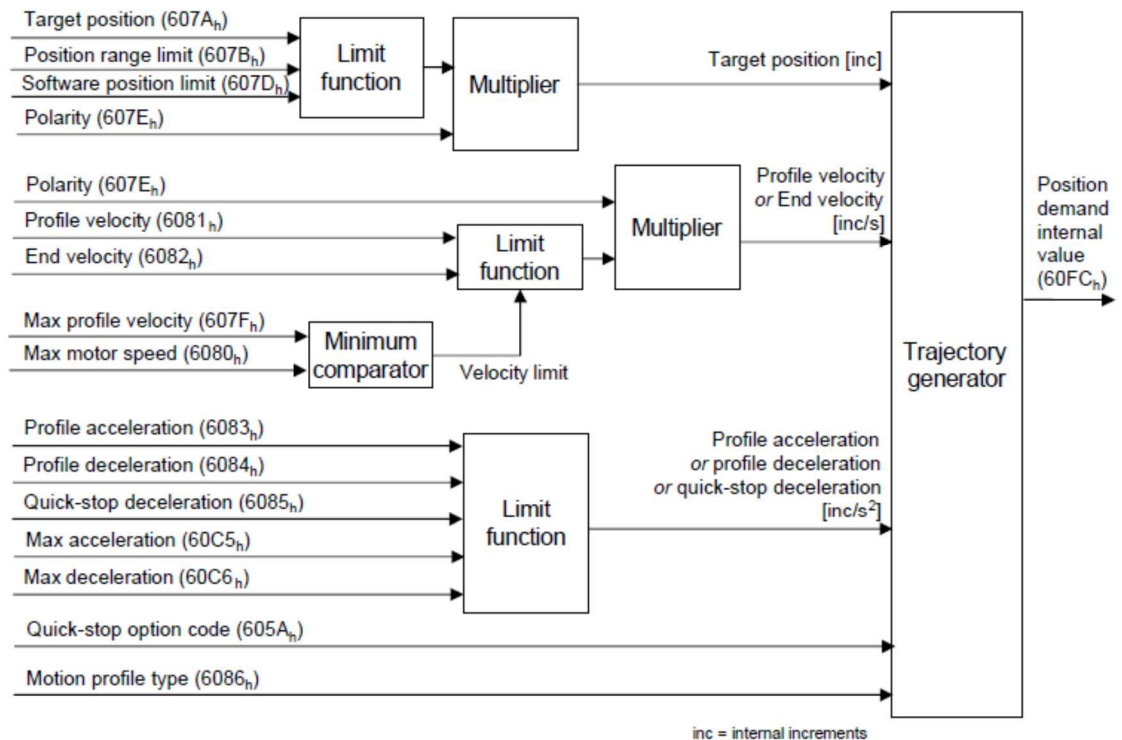


Abbildung 2: Profile Position Mode

Implementierte Objekte:

- Target position
- Software position limit
- Polarity
- Profile velocity
- Max profile velocity
- Max motor speed
- Profile acceleration
- Profile deceleration
- Quick stop deceleration
- Max acceleration
- Max deceleration
- Quick stop option code

Implementierte Funktionsblöcke:

- Limit function
- Minimum comparator
- Trajectory Generator

- Velocity Controller

## 2.2 ControlWord/StatusWord

### 2.2.1 ControlWord

Ansteuerung	Bit 9	Bit 5	Bit 4	Beschreibung
1Fh	0	0	1	Single Setpoint, jede Positionierung wird beendet, bevor der nächste Setpoint aktiv wird. Übertragene Setpoints werden zwischengespeichert (Set of Setpoints)
3Fh	0	1	1	Übertragener Setpoint wird sofort aktiviert, Set of Setpoint Liste wird gelöscht
21Fh	1	0	1	Change on Setpoint

**Tabelle 1: PPmode ControlWord**

#### Bit 6 (abs/rel)

0: absolute Positionierung

1: relative Positionierung

#### Bit 8 (Halt)

1: Positionierung stoppen (Halt)

0: Positionierung fortsetzen (Release from Halt)

### 2.2.2 StatusWord

#### Bit 10 (Target reached)

0: wenn Halt = 0, dann Zielposition nicht erreicht  
wenn Halt = 1, Antrieb bremst

1: wenn Halt = 0, dann Zielposition erreicht  
wenn Halt = 1, Antriebsdrehzahl ist 0

#### Bit 12 (SetPoint acknowledge)

0→1: SetPoint akzeptiert, bzw. zwischengespeichert

1→0 Nächster SetPoint kann angenommen werden

### 2.2.3 Single Setpoint

Jeder Setpoint besteht aus Zielposition und Profilgeschwindigkeit. Für jeden Setpoint kann zwischen relativer und absoluter Positionierung ausgewählt werden. Die Art der relativen Positionierung kann über das Objekt 60F2h festgelegt werden.

### 2.2.4 Set of Setpoints

Die Funktion Set of Setpoints unterstützt die Option Change on Setpoint. Momentan können bis zu 20 Setpoints abgelegt werden. Jeder Setpoint besteht aus Zielposition und Profilgeschwindigkeit. Für jeden Setpoint kann zwischen relativer und absoluter Positionierung ausgewählt werden. Die Art der relativen Positionierung kann über das Objekt 60F2h festgelegt werden.

Die Zahl der Setpoints kann bei Bedarf, ohne großen Aufwand angepasst werden.

## 2.3 Implementierte Objekte

### 2.3.1 Object 607Ah: Target position

Die Eingabe der Zielposition erfolgt in user defined Position Units.

Hierbei gilt:

$$\text{position internal value} = \frac{\text{target position} * \text{position encoder resolution} * \text{gear ration}}{\text{feed constant}}$$

$$\text{feed constant} = \frac{6092h: 1}{6092h: 2}$$

$$\text{encoder resolution} = \frac{608Fh: 1}{608Fh: 2}$$

$$\text{gear ratio} = \frac{6091h: 1}{6091h: 2}$$

### 2.3.2 Object 607Dh: Software position limit

Die Eingabe der Softwareendlagen erfolgt in user defined Position Units. Zielpositionen hinter den Softwareendlagen werden auf die Softwareendlagen begrenzt.

Hierbei gilt:

$$\text{position internal value} = \frac{\text{software position limit} * \text{position encoder resolution} * \text{gear ration}}{\text{feed constant}}$$

$$\text{feed constant} = \frac{6092h: 1}{6092h: 2}$$

$$\text{encoder resolution} = \frac{608Fh: 1}{608Fh: 2}$$

$$\text{gear ratio} = \frac{6091h: 1}{6091h: 2}$$

Werden während einer aktiven Positionierung die Endlagen geändert, wird eine Positionierung auf die Endlagen mit der eingestellten Verzögerung vorgenommen, falls diese überfahren werden sollten. Hierzu werden 3 Bremspunkte berechnet.

- Bremspunkt rechte Endlage
- Bremspunkt linke Endlage
- Bremspunkt Zielposition

### 2.3.3 Object 607Fh: Max profile velocity

Die Eingabe der maximalen Profilgeschwindigkeit erfolgt in U/min.

### 2.3.4 Object 6080h: Max motor speed

Die Eingabe der maximalen Motordrehzahl erfolgt in U/min.

### 2.3.5 Object 6081h: Profile velocity

Die Eingabe der Profilgeschwindigkeit erfolgt in User defined Units.

Die Umrechnung der eingegebenen Einheit in die interne Einheiten erfolgt mithilfe der Objekte 6092h, 608Fh und 6091h.

Hierbei gilt:

$$velocity\ internal\ value = \frac{profile\ velocity * position\ encoder\ resolution * gear\ ration}{feed\ constant}$$

$$feed\ constant = \frac{6092h:1}{6092h:2}$$

$$encoder\ resolution = \frac{608Fh:1}{608Fh:2}$$

$$gear\ ratio = \frac{6091h:1}{6091h:2}$$

### 2.3.6 Object 6083h: Profile acceleration

Die Eingabe der Beschleunigung erfolgt in 0,01525 U/s<sup>2</sup>.

### 2.3.7 Object 6084h: Profile deceleration

Die Eingabe der Verzögerung erfolgt in 0,01525 U/s<sup>2</sup>.

### 2.3.8 Object 6085h: Quick stop deceleration

Die Eingabe der Verzögerung erfolgt in 0,01525 U/s<sup>2</sup>.

### 2.3.9 Object 60C5h: Max acceleration

Die Eingabe der maximalen Beschleunigung erfolgt in 0,01525 U/s<sup>2</sup>.

### 2.3.10 Object 60C6h: Max deceleration

Die Eingabe der maximalen Verzögerung erfolgt in 0,01525 U/s<sup>2</sup>.

## 3 Profile Velocity Mode

### 3.1 Übersicht

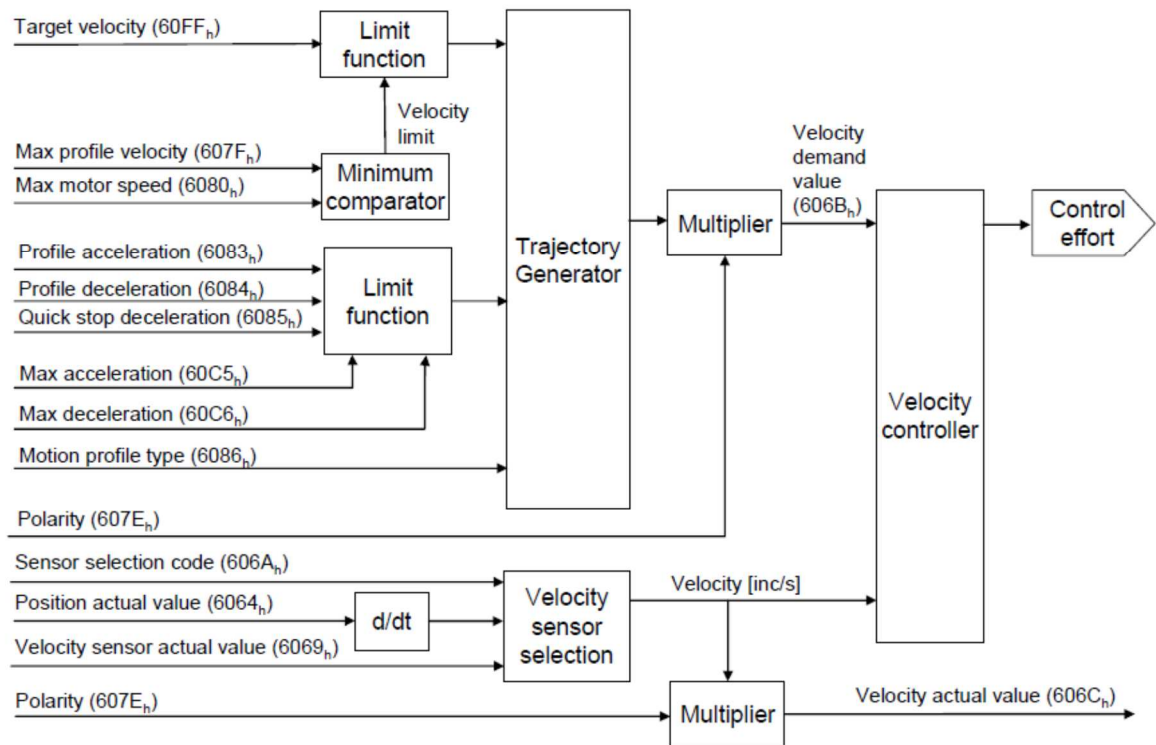


Abbildung 3: Profile Velocity Mode

Implementierte Objekte:

- Target Velocity
- Max profile velocity
- Max motor speed
- Profile acceleration
- Profile deceleration
- Quick stop deceleration
- Max acceleration
- Max deceleration
- Polarity
- Position actual value

Implementierte Funktionsblöcke:

- Limit function
- Minimum comparator
- Trajectory Generator
- Velocity Controller



## 3.2 ControlWord/StatusWord

### 3.2.1 ControlWord

#### Bit 8 (Halt)

- 1: Positionierung stoppen (Halt)
- 0: Positionierung fortsetzen (Release from Halt)

### 3.2.2 StatusWord

#### Bit 10 (Target reached)

- 0: wenn Halt = 0, Soll Drehzahl nicht erreicht  
wenn Halt = 1, Antrieb bremst
- 1: wenn Halt = 0, Soll Drehzahl erreicht  
wenn Halt = 1, Antriebsdrehzahl ist 0

#### Bit 12 (SetPoint acknowledge)

- 0: Ist Drehzahl ungleich 0
- 1: Ist Drehzahl gleich 0

## 3.3 Implementierte Objekte

### 3.3.1 Object 606Bh: Velocity demand value

Ausgang des Trajektorien Generators. Der Wert wird in internen Drehzahleinheiten angegeben.

### 3.3.2 Object 606Ch: Velocity actual value

Aktuelle Drehzahl in user defined units. Hierbei gilt:

$$velocity\ actual\ value = \frac{velocity\ internal\ value * feed\ constant}{encoder\ resolution * gear\ ratio}$$

$$feed\ constant = \frac{6092h:1}{6092h:2}$$

$$encoder\ resolution = \frac{608Fh:1}{608Fh:2}$$

$$gear\ ratio = \frac{6091h:1}{6091h:2}$$

### 3.3.3 Object 606Dh: Velocity window

Das Drehzahlfenster spezifiziert einen Bereich um die Zielgeschwindigkeit.

Befindet sich die aktuelle Ist-Drehzahl nach dem Fahrbefehl - über einen Zeitraum wie in 606Eh definiert - innerhalb des Drehzahlfensters wird das Bit10 im Statuswort gesetzt. Bit10 wird bei jedem neuen Fahrbefehl zurückgesetzt.

### 3.3.4 Object 606Eh: Velocity window time

Zeitfenster in ms.

### 3.3.5 Object 606Fh: Velocity threshold

Überschreitet die aktuelle Drehzahl den Wert des Objekts - über einen Zeitraum wie in 6070h definiert wird das Bit12 im Statuswort auf 0 gesetzt.

Unterschreitet die aktuelle Drehzahl den Wert des Objekts - über einen Zeitraum wie in 6070h definiert wird das Bit12 im Statuswort auf 1 gesetzt.

### 3.3.6 Object 6070h: Velocity threshold time

Zeitfenster in ms.

### 3.3.7 Object 60FFh: Target velocity

Die Eingabe der Solldrehzahl erfolgt in User defined Units.

Hierbei gilt:

$$velocity\ internal\ value = \frac{target\ velocity * position\ encoder\ resolution * gear\ ratio}{feed\ constant}$$

$$feed\ constant = \frac{6092h:1}{6092h:2}$$

$$encoder\ resolution = \frac{608Fh:1}{608Fh:2}$$

$$gear\ ratio = \frac{6091h:1}{6091h:2}$$

Bei Änderung der Solldrehzahl im Zustand OperationEnabled wird ein Fahrbefehl generiert.

## 4 Homing Mode

### 4.1 Übersicht

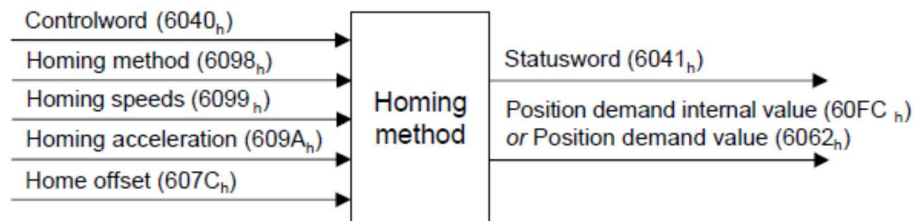


Abbildung 3: Homing Mode

Implementierte Objekte:

- Controlword
- Homing method
- Homing speeds
- Homing acceleration
- Home offset
- Statusword
- Position demand value

Implementierte Homing Methoden:

- Methode 17
- Methode 18
- Methode 19
- Methode 20
- Methode 21
- Methode 22
- Methode 23
- Methode 24
- Methode 25
- Methode 26
- Methode 27
- Methode 28
- Methode 29
- Methode 30
- Methode 35
- Methode -18 (Homing on right block)
- Methode -19 (Homing on left block)

## 4.2 ControlWord/StatusWord

### 4.2.1 ControlWord

#### Bit 4 (Homing operation start)

0: Referenzfahrt nicht starten

1: Referenzfahrt starten

#### Bit 8 (Halt)

1: Referenzfahrt stoppen (Halt)

0: Referenzfahrt freigeben (Release from Halt)

### 4.2.2 StatusWord

Bit 13	Bit 12	Bit 10	Beschreibung
0	0	0	Referenzfahrt aktiv
0	0	1	Referenzfahrt wurde unterbrochen
0	1	0	Na
0	1	1	Referenzfahrt erfolgreich beendet
1	0	0	Fehler während Referenzfahrt
1	0	1	Na

## 4.3 Implementierte Objekte

### 4.3.1 Object 607Ch: Home offset

Bei erfolgreichem Homing wird der gefundenen Home Position der negative Home Offset zugewiesen.

### 4.3.2 Object 6098h: Homing method

Auswahl der Homing Methoden.

### 4.3.3 Object 6099h: Homing speeds

Homing Geschwindigkeit. Der Antrieb fährt nicht automatisch auf Position 0 zurück.

### 4.3.4 Object 609Ah: Homing acceleration

Beschleunig.

#### **4.3.5 Object 60B8h: Touch probe function**

Konfiguration der Sensoren:

Bits 0-3: home switch, mögliche Eingänge 1-9

Bits 4-7: left limit switch, mögliche Eingänge 1-9

Bits 8-11: right limit switch, mögliche Eingänge 1-9

Bits 12-15: reserved

Wert 1 aktiviert Eingang 1 als entsprechenden Sensor.

Wert 2 aktiviert Eingang 2 als entsprechenden Sensor.

...

Wert 9 aktiviert Eingang 9 als entsprechenden Sensor.

## 5 I/O Functions

### 5.1.1 Objekt 0x3611 Ausgangsfunktion 1

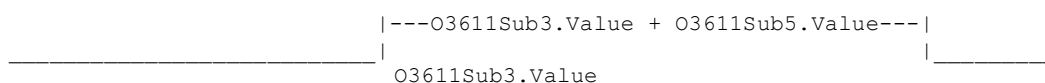
Aktivierung und Deaktivierung von Ausgang 1 über Positionsbedingung.

Die Pulsdauer wird in Inkrementen angegeben.

Index	Subindex	Bezeichnung	Vorgabe
0x3611	0x01	enable/disable right signal	
	0x02	enable/disable left signal	
	0x03	right position	
	0x04	left position	
	0x05	right position distance	
	0x06	left position distance	

**Tabelle 2: Objekt 0x3611**

Beispiel (enable/disable right signal = 1):



### 5.1.2 Objekt 0x3612 Ausgangsfunktion 2

Über das Objekt 0x3612 Subindex 1 kann der Ausgang 3 mit den Bits von Objekt 0x1002 belegt werden.

Index	Subindex	Bezeichnung	Vorgabe
0x3612	0x01	enable/disable output signal	

**Tabelle 3: Objekt 0x3612**

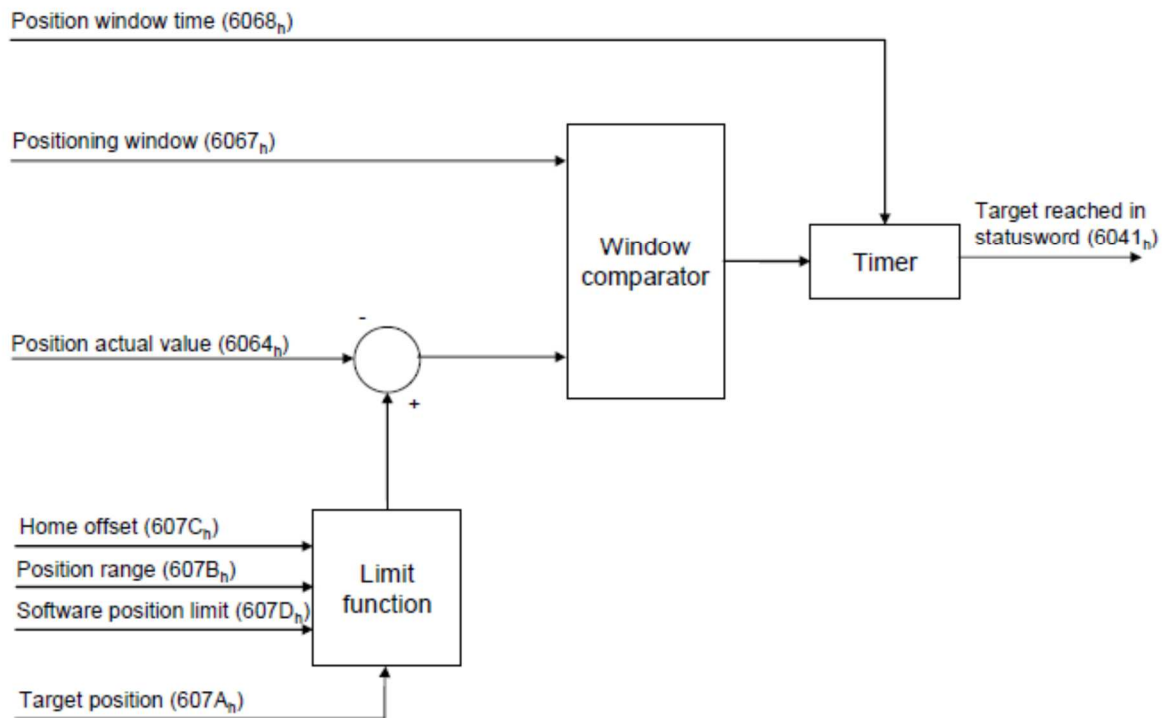
Codierung	Beschreibung
0x0000	Disable out
0x0001	Error over volt (Uzk)
0x0002	Error under volt (Uzk)
0x0004	Error Ack
0x0008	Error over temp
0x0010	Error derating
0x0020	Abort connection

0x0040	Error stroke
0x0080	Error communication
0x0100	Error overcurrent

**Tabelle 4: Fehler Codierung**

## 6 Position control function

### 6.1 Übersicht



**Abbildung 4: Position reached**

Implementierte Objekte:

- Position window time
- Positioning window
- Position actual value
- Home offset
- Software position limit
- Target position

Implementierte Funktionsblöcke:

- Limit function
- Window comparator
- Timer



## 6.2 Implementierte Objekte

### 6.2.1 Object 6063h: Position actual internal value

Aktuelle Position in internen Einheiten.

### 6.2.2 Object 6064h: Position actual value

Aktuelle Position in User defined Units. Hierbei gilt:

$$position\ actual\ value = \frac{position\ internal\ value * feed\ constant}{encoder\ resolution * gear\ ratio}$$

$$feed\ constant = \frac{6092h:1}{6092h:2}$$

$$encoder\ resolution = \frac{608Fh:1}{608Fh:2}$$

$$gear\ ratio = \frac{6091h:1}{6091h:2}$$

### 6.2.3 Object 6067h: Position window

Fenster um die Zielposition. Befindet sich der Antrieb über die Zeit 6068h innerhalb des Zielfensters wird das Bit Target Reached im Statuswort gesetzt.

### 6.2.4 Object 6068h: Position window time

Zeitfenster in ms.

### 6.2.5 Object 60FCh: Position demand internal value

Aktuelle Sollposition des Trajektorien Generators.

### 6.2.6 Object 60F2h: Positioning option code

Unterstützte Optionen:

Bit 1	Bit 0	Definition
0	0	Positioning moves shall be performed relative to the preceding (internal absolute) target position (rsp. relative to 0 if there is no preceding target position) as described in 10.2
0	1	Positioning moves shall be performed relative to the actual position demand value (object 60FC <sub>h</sub> ) – output of the trajectory generator
1	0	Positioning moves shall be performed relative to the position actual value (object 6064 <sub>h</sub> )
1	1	Reserved

## 7 Herstellerspezifische Objekte

### 7.1 Pythoninterpreter

Die Schnittstelle zwischen dem Pythoninterpreter und der Motorsteuerung bildet das Objektverzeichnis. Funktionen aus der Python targetlib ermöglichen den Zugriff auf einzelne Objekte.

Hierfür stehen folgende Funktionen zur Verfügung:

#### 7.1.1 Funktionen

##### 7.1.1.1 ReadObjectByte(n, m)

```
/******  
**  
** Function    : ReadObjectByte  
**  
** Description : Liest einen 8 Bit Objektverteichniseintrag.  
**  
** Parameter  : wIndex   - Index  
**              bySubindex - Subindex  
**              byValue   - Wert  
**  
** Returnvalues : tCopKernel  
**  
*****/
```

##### 7.1.1.2 ReadObjectDword(n, m)

```
/******  
**  
** Function    : ReadObjectWord  
**  
** Description : Liest einen 16 Bit Objektverteichniseintrag.  
**  
** Parameter  : wIndex   - Index  
**              bySubindex - Subindex  
**              wValue   - Wert  
**  
** Returnvalues : tCopKernel
```

\*\*

\*\*\*\*\*/

### 7.1.1.3 ReadObjectWord(n, m)

/\*\*\*\*\*\*

\*\*

\*\* Function : WriteObjectWord

\*\*

\*\* Description : Schreibt einen 16 Bit Objektverteichniseintrag.

\*\*

\*\* Parameter : wIndex - Index

\*\* bySubindex - Subindex

\*\* byValue - Wert

\*\*

\*\* Returnvalues : tCopKernel

\*\*

\*\*\*\*\*/

### 7.1.1.4 SendEmcyMsg()

/\*\*\*\*\*\*

\*\*

\*\* Function : SendEmcyMessage

\*\*

\*\* Description : Sendet die übergebene Fehlernummer über  
den Emergency Kanal.

\*\*

\*\* Parameter : keine

\*\*

\*\* Returnvalues : keine

\*\*

\*\*\*\*\*/

### 7.1.1.5 WriteObjectByte(n, m, o)

/\*\*\*\*\*\*

\*\*

\*\* Function : WriteObjectByte

\*\*

\*\* Description : Schreibt einen 8 Bit Objektverteichniseintrag.

\*\*

\*\* Parameter : wIndex - Index

\*\* bySubindex - Subindex

\*\* byValue - Wert

\*\*

\*\* Returnvalues : tCopKernel

\*\*

\*\*\*\*\*/

#### 7.1.1.6 WriteObjectDword(n, m, o)

\*\*\*\*\*/

\*\*

\*\* Function : WriteObjectDWord

\*\*

\*\* Description : Schreibt einen 32 Bit Objektverteichniseintrag.

\*\*

\*\* Parameter : wIndex - Index

\*\* bySubindex - Subindex

\*\* byValue - Wert

\*\*

\*\* Returnvalues : tCopKernel

\*\*

\*\*\*\*\*/

#### 7.1.1.7 WriteObjectWord(n, m, o)

\*\*\*\*\*/

\*\*

\*\* Function : WriteObjectWord

\*\*

\*\* Description : Schreibt einen 16 Bit Objektverteichniseintrag.

\*\*

\*\* Parameter : wIndex - Index

\*\* bySubindex - Subindex

\*\* byValue - Wert

\*\*

\*\* Returnvalues : tCopKernel

\*\*

\*\*\*\*\*/

### 7.1.2 Pythonvariablen

MovingCap verfügt über einen Satz frei verwendbarer Objekte, die im permanenten Speicher abgelegt werden können (store all parameters).

### 7.1.3 8Bit Variablen

Objekt 0x340B Subindex 1 - 0x0A

Objekt 0x340C Subindex 1 - 0x0A

### 7.1.4 16Bit Variablen

Objekt 0x340D Subindex 1 - 0x0A

Objekt 0x340E Subindex 1 - 0x0A

### 7.1.5 16Bit Variablen

Objekt 0x340F Subindex 1 - 0x0A

Objekt 0x3410 Subindex 1 - 0x0A

## 7.2 Objekt 0x3401 Regelparameter

### 7.2.1 Blockfahrterkennung

Bei der Blockfahrterkennung wird die Zahl der Blockereignisse ausgewertet.

Ein Blockereignis liegt vor, wenn der momentane Phasenstrom größer als der parametrisierte Maximalstrom (Objekt 0x6073) und die momentane Drehzahl kleiner der Minimaldrehzahl für die Blockfahrterkennung ist.

Pro ms kann ein Ereignis gezählt werden. Ist die Zahl der Ereignisse größer als der Maximalwert des Zählers für die Blockfahrterkennung, wird Blockfahrt gemeldet.

#### 7.2.1.1 Objekt 0x3401 Subindex 1

Maximalwert Zähler Blockfahrterkennung

#### 7.2.1.2 Objekt 0x3401 Subindex 2

Minimaldrehzahl Blockfahrterkennung

#### 7.2.2 Objekt 0x3401 Subindex 3

Positionierfenster

### 7.2.3 RS485

MovingCap349 kommuniziert mit dem Motor über eine interne RS485 Schnittstelle.

#### 7.2.3.1 Objekt 0x3401 Subindex 4

Uart Timeout [ms]

Wird innerhalb der spezifizierten Zeit keine Nachricht vom Motor empfangen, wird ein Kommunikationsfehler gemeldet.

#### 7.2.3.2 Objekt 0x3401 Subindex 5

Uart Errors

Zahl der Uart Fehler seit Power Up.

### 7.2.4 Objekt 0x3401 Subindex 0x15

Acceleration Torque

Maximalstrom in der Beschleunigungsrampe in 1/1000 des Nennstroms.

### 7.2.5 Objekt 0x3401 Subindex 0x16

Deceleration Torque

Maximalstrom in der Verzögerungsrampe in 1/1000 des Nennstroms.

## 7.3 Option Codes

### 7.3.1 Objekt 0x6059 Subindex 0x01 Stroke option code

Blockfahrt Option Code

0: Antrieb wechselt in Zustand FAULT

1: Antrieb stoppt, sendet Emergency Message, wechselt nicht in den Zustand FAULT

2: Antrieb führt eine **absolute** Positionierung durch und sendet die entsprechende Emergency Message.

Objekt **0x6059:0x02**: Zielposition [Inkremente]

Objekt **0x6059:0x03**: Geschwindigkeit [U/min]

3: Antrieb führt eine **relative** Positionierung durch und sendet die entsprechende Emergency Message.

Objekt **0x6059:0x02**: Zielposition [Inkremente]

Objekt **0x6059:0x03**: Geschwindigkeit [U/min]

## 8 Emergency Messages

Fehlercode	Beschreibung
3211h	Überspannung Zwischenkreis
3221h	Unterspannung Zwischenkreis
4220h	I <sup>2</sup> t
4210h	Übertemperatur
8160h	CAN
7121h	Blockfahrt
7510h	Motorkommunikationsfehler
5530h	EEProm

**Tabelle 5: Emergency Messages**