

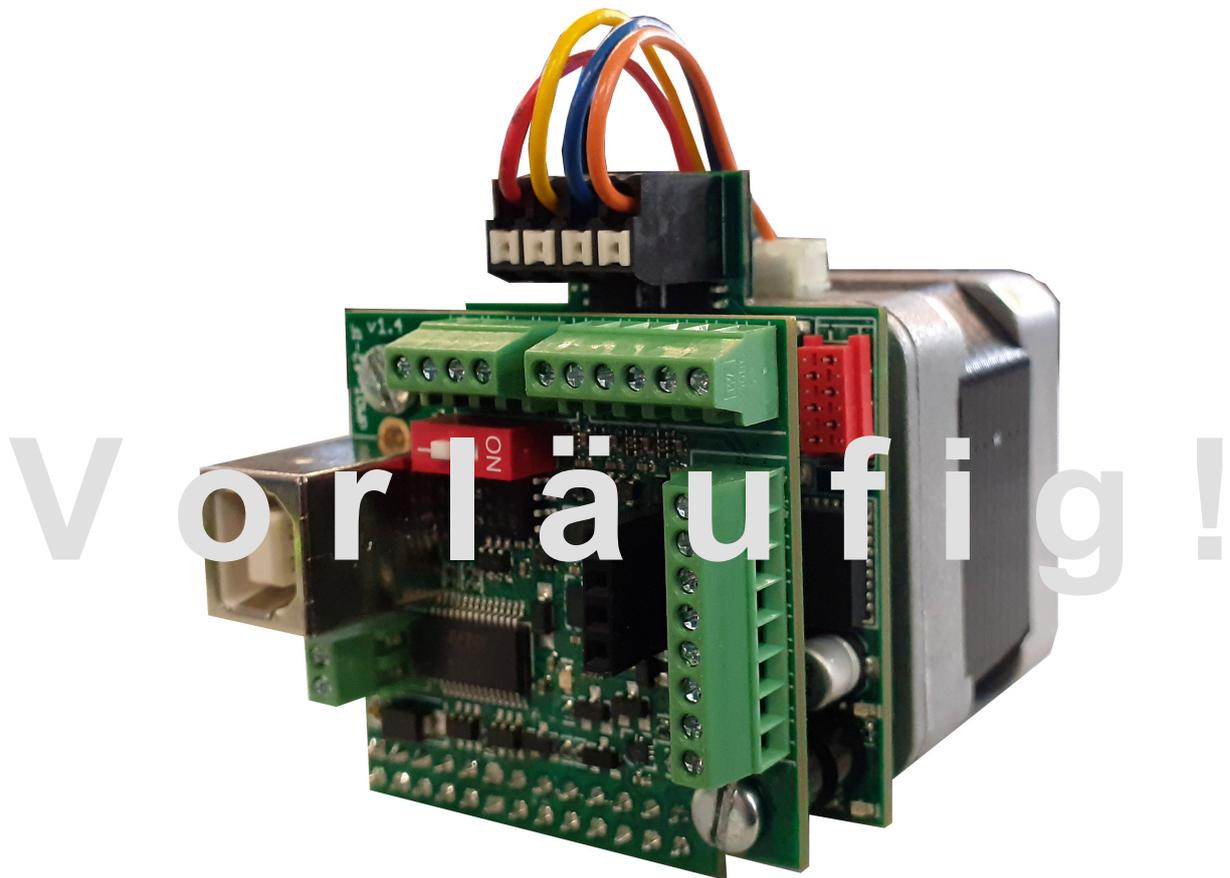
Schrittmotorcontroller

# SMC-dMOT-42

Technisches Handbuch

Hardware-Version 1.3

Firmware-Version 1.067



EMIS GmbH  
Zur Drehscheibe 4  
D-92637 Weiden

[info@emisgmbh.de](mailto:info@emisgmbh.de)  
[www.emisgmbh.de](http://www.emisgmbh.de)

**Die Firma EMIS GmbH behält sich das Recht vor, technische Änderungen und Weiterentwicklungen von Hard- und Software zur Verbesserung der Funktionalität dieses Produktes vorzunehmen.**

**Dieses Handbuch wurde mit der gebotenen Sorgfalt zusammengestellt. Es dient ausschließlich der technischen Beschreibung des Produkts und der Anleitung zur Inbetriebnahme. Die Gewährleistung beinhaltet ausschließlich Reparatur oder Umtausch defekter Geräte, eine Haftung für Folgeschäden und Folgefehler ist ausgeschlossen. Bei der Installation des Gerätes sind die gültigen Normen und Vorschriften zu beachten.**

# Inhalt

## 1. Sicherheitshinweise

## 2. Lieferumfang

## 3. Inbetriebnahme

## 4. Funktionsbeschreibung

## 5. Hardware

### 5.1 Digitale Eingänge

5.1.1 Eingänge für externes Takt- und Richtungssignal

5.1.2 Eingang für Software - NOT-AUS (IN1)

5.1.3 Eingang für Referenzschalter (IN2)

5.1.4 Frei programmierbare Eingänge IN3 bis IN7

5.1.5 Eingang für Fehlerquittierung (IN8)

### 5.2 Digitale Ausgänge

### 5.3 Kommunikations-Interface

5.3.1 USB

5.3.2 RS485

5.3.3 Steckplatz für Bluetooth-Interface

### 5.4 Integrierter Festwertspeicher

### 5.5 Stromversorgung

### 5.6 Anschlussbelegung

## 6. Technische Daten

### 6.1 Steuerung

### 6.2 Schrittmotor

## 7. Zubehör

### 7.1 Bluetooth-Interface

## 8. Software

### 8.1 StandAlone-ProgrammWriter-2d0

## 9. Standalone-Betrieb

# Inhalt

## 10. Befehlsreferenz

- 10.1 Versionsabfrage
- 10.2 Eingänge abfragen
- 10.3 Ausgänge setzen
- 10.4 Startgeschwindigkeit setzen
- 10.5 Arbeitsgeschwindigkeit setzen
- 10.6 Rampendauer setzen
- 10.7 Schrittauflösung setzen
- 10.8 Motorstrom setzen
- 10.9 Referenzfahrt
- 10.10 Offset nach Referenzfahrt setzen
- 10.11 Vektorfahrt
- 10.12 Vektorfahrt mit Rampe abbrechen
- 10.13 Vektorfahrt sofort abbrechen
- 10.14 Positionsabfrage
- 10.15 Wartezeit setzen
- 10.16 Eingang für Software – NOT-AUS aktivieren/deaktivieren
- 10.17 Automatische Referenzfahrt nach NOT-AUS und Fehlerquittierung
- 10.18 Stand-Alone – Ablaufprogramm
  - 10.18.1 Aufbau eines Ablaufprogramms
  - 10.18.2 Befehle für ein Ablaufprogramm

## **1. Sicherheitshinweise**

- Die Elektronik enthält Bauteile, die empfindlich gegen elektrostatische Entladung sind. Beachten Sie die Grundregeln der ESD-Schutzmaßnahmen. Nichtbeachtung kann die Elektronik beschädigen.
- Nehmen Sie keine Verdrahtung an der Elektronik vor, ohne diese vorher von der Stromversorgung zu nehmen. Nichtbeachtung kann die Elektronik beschädigen.
- Vermeiden Sie Kurzschlüsse zwischen den Anschlussleitungen. Nichtbeachtung kann die Elektronik beschädigen.
- Verwenden Sie nur eine zugelassene Stromversorgung. Nichtbeachtung kann die Elektronik beschädigen.
- Achten Sie beim Anschluss der Stromversorgung auf richtige Polarität. Bei Verpolung entsteht ein Kurzschluss zwischen den Versorgungsleitungen über eine Diodenstrecke. Die interne Schmelzsicherung spricht an und muss ersetzt werden. Bei ausreichend hohem Spitzenstrom der Stromversorgung kann zusätzlich die Diode zerstört werden.

## **2. Lieferumfang**

- Schrittmotorcontroller SMC-dMOT-42
- Schrittmotor (Option)
- Zubehör (Option)

## **3. Inbetriebnahme**

Lesen Sie vor der Inbetriebnahme das gesamte Handbuch und beachten Sie die Sicherheitshinweise.

Schließen Sie die Motorleitungen an den entsprechenden Klemmen an, sofern Sie keine vormontierte Kombination erworben haben. Bei der Kombination ist der Motor bereits korrekt angeschlossen.

Schließen Sie einen vorhandenen Referenzschalter an den entsprechenden Klemmen an. Verdrahten Sie die restlichen Eingänge und die Ausgänge nach Bedarf.

Stellen Sie eine USB-Verbindung mittels USB-Kabel (Typ A/B) zum PC her.

Schließen Sie die Stromversorgungsleitungen an den vorgesehenen Klemmen an. Achten Sie dabei unbedingt auf korrekte Polarität und die maximale Versorgungsspannung. Schalten Sie die Stromversorgung ein.

## **4. Funktionsbeschreibung**

### Eigenschaften

- *Schrittmotor bis max. 2,5A Phasenstrom steuerbar*
- *Schrittauflösung 1, 2, 4, 8, 16, 32, 64, 128, 256 Mikroschritte pro Vollschritt*
- *Anschluss für 1 Referenzschalter (Typ: Schließer, Öffner, PNP)*
- *USB-Anschluss 2.0 (Typ B) für PC*
- *10 digitale Eingänge*
- *2 digitale Ausgänge*
- *32 MBit - Festwertspeicher für eigene Programme*
- *Geeignet für Standalone-Anwendungen (Betrieb ohne PC)*
- *Schnittstelle für externes Kommunikationsinterface (Bluetooth) in Vorbereitung*
- *Nur eine Stromversorgung für Logik und Schrittmotoren notwendig*
- *Firmware mit Downloadmanager aktualisierbar (zukunftsicher)*
- *Kundenwünsche realisierbar*

Die Schrittmotorsteuerung SMC-dMOT-42 kann Schrittmotoren mit Phasenströmen von bis zu 2,5A steuern. Die Motorenstufe arbeitet nach dem bipolaren Stromchopperverfahren. Es können bipolare Schrittmotoren mit zwei oder vier Phasen (mit entsprechender Verschaltung) angeschlossen werden.

Als Stromversorgung wird ein Gleichspannungsnetzteil mit einer Spannung von mindestens 12V DC bis maximal 48V DC benötigt. Der erforderliche Strom hängt vom eingestellten Phasenstrom ab. Im Bedarfsfall helfen wir bei der Dimensionierung gerne weiter.

Die Kommunikation zwischen PC und SMC-dMOT-42 erfolgt über USB (Typ B). Nach erstmaligem Einstecken des USB-Kabels wird auf dem PC ein virtueller COM-Port erzeugt und eine COM-Port-Nr. zugewiesen. Über diesen virtuellen COM-Port finden sämtliche Datentransfers zwischen PC und SMC-dMOT-42 statt. Eine RS485-Schnittstelle ist vorhanden, wird in der Firmware allerdings noch nicht unterstützt. Die Unterstützung wird in künftigen Versionen erfolgen.

Der USB-Treiber kann von unserer Homepage unter [www.emisgmbh.de/downloads-treiber/](http://www.emisgmbh.de/downloads-treiber/) oder direkt vom Hersteller des USB-UART-Umsetzers, FTDI, unter [www.ftdichip.com/Drivers/VCP.htm](http://www.ftdichip.com/Drivers/VCP.htm), abgerufen werden. Zur Ansteuerung der SMC-dMOT-42 kann ein Terminal-Programm verwendet werden. In absehbarer Zeit werden wir auf unserer Homepage ein Anwenderprogramm zum Download zur Verfügung stellen. Programme für den Standalone-Betrieb können mit unserer kostenlosen Software „StandAlone-ProgrammWriter-2d0“ in den Festwertspeicher geladen werden.

Den digitalen Eingängen können Schalt- bzw. Steuersignale mit einem Pegel von 0 – 24V DC zugeführt werden.

Am vorgesehenen Eingang kann ein Referenzschalter angeschlossen werden. Damit kann das System zu einer definierten Nullposition gebracht werden.

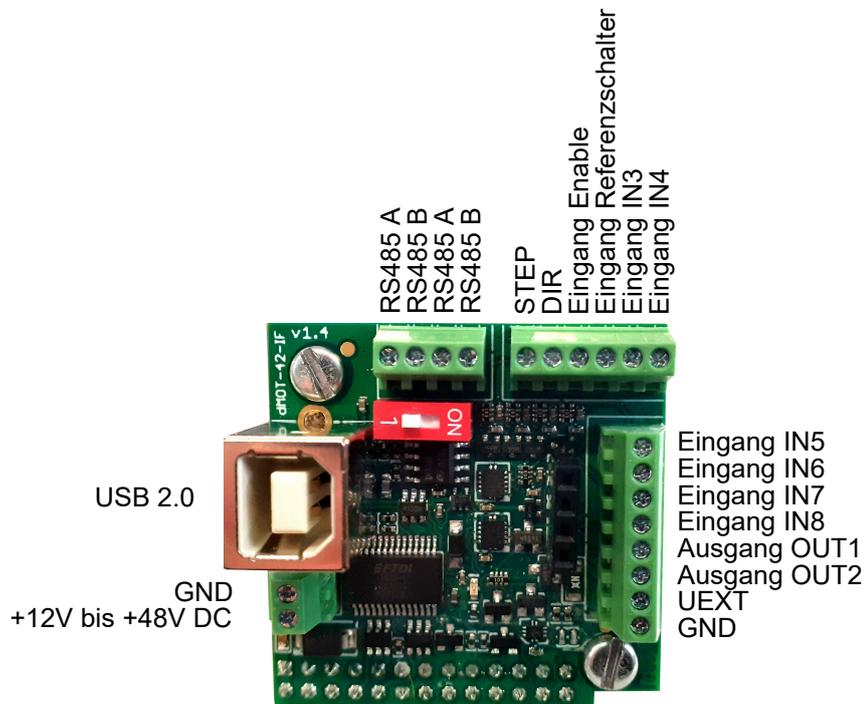
Sofern der Enable-Eingang über die Software aktiviert wird, kann hier eine NOT-AUS-Linie (Öffner) angeschlossen werden. Dann muss an diesem Eingang ein positiver Pegel anliegen (+5V bis +24V), damit die Motorendstufe für Drehbewegungen freigegeben wird. Wird der NOT-AUS-Schalter während einer Motorbewegung getroffen, stoppt die Drehbewegung sofort und die Fehleranzeige blinkt. Dieser Zustand kann nur mit einer Quittierung (positiver Pegel für mindestens 100 ms an Eingang 8) verlassen werden. Soll nach der Quittierung eine automatische Referenzfahrt erfolgen, kann das mit einem bereits vorher platzierten Kommando (Befehlsreferenz 10.17) eingestellt werden. Soll die Quittierung nach einem Fehlerfall automatisch erfolgen, muss am Eingang 8 ein Dauerpegel (+5V bis +24V) angelegt werden.

Es stehen zwei Transistorausgänge mit max. 24V DC Spannung und 100 mA Laststrom zur Verfügung. Werden höhere Leistungen benötigt, sind entsprechende Maßnahmen außerhalb der SMC-dMOT-42 zu treffen. Die Schaltspannung wird der SMC-dMOT-42 an der Klemmleiste von extern zugeführt. Damit ist es möglich die Ausgänge an verschiedene Logiken anzupassen (TTL, CMOS, SPS). Die Befehle zum Schalten der Ausgänge sind in „Befehlsreferenz 10.3“ beschrieben.

Für Standalone-Anwendungen steht ein Festwertspeicher für 10 flankengetriggerte Programme mit jeweils 32 kB zur Verfügung. Diese Programme können den Eingängen 3 bis 7 (jeweils 1 Programm für die positive Flanke und 1 Programm für die negative Flanke) zugewiesen werden. Mehr hierzu können Sie der Beschreibung „Standalone-Betrieb“ unter Punkt 9 entnehmen.

## 5. Hardware

Abb. 1:  
SMC-dMOT-42



### 5.1 Digitale Eingänge

Als High-Pegel werden Spannungen von +5V bis max. +24V erkannt. Spannungen von 0V bis 0,8V werden als Low-Pegel interpretiert.

#### 5.1.1 Eingänge für externes Takt- und Richtungssignal

An diesen Eingängen können Takt- und Richtungssignal einer externen Signalquelle (z.B. SPS, Signalgenerator) angelegt werden. Geschwindigkeit und Drehrichtung werden dann von der externen Signalquelle bestimmt.

### 5.1.2 Eingang für Software - NOT-AUS (IN1)

An diesem Eingang kann eine NOT-AUS-Linie (Öffner) angeschlossen werden, sofern diese Funktion mit dem entsprechenden Kommando freigegeben wurde (siehe Befehlsreferenz 10.16).

### 5.1.3 Eingang für Referenzschalter (IN2)

Als Referenzschalter kann aktuell ein mechanischer oder ein elektronischer Schließer angeschlossen werden. In naher Zukunft kann per Parametereinstellung zwischen Schließer und Öffner gewählt werden. Eine Referenzfahrt erfolgt in negativer Richtung, bis der Referenzschalter getroffen wird. Danach wird dieser freigefahren. Ein nachfolgender Referenz-Offset (definierter Abstand vom Referenzschalter) kann eingestellt werden (siehe Befehlsreferenz 10.10).

### 5.1.4 Frei programmierbare Eingänge IN3 bis IN7

Für diese Eingänge sowohl für die positive Flanke als auch für die negative Flanke ein Ablaufprogramm in den Festwertspeicher hinterlegt werden. Mit der positiven Flanke an einem Eingang löst man z.B. eine dauerhafte Motorbewegung aus und mit der negativen Flanke am selben Eingang stoppt man den Dauerlauf.

Aktuell dürfen sich Ablaufprogramme nicht überschneiden. Wurde z.B. ein Ablaufprogramm mit einem positiven Pegel an Eingang 3 gestartet, kann ein weiteres Ablaufprogramm, das einem anderen Eingang zugeordnet ist, erst dann gestartet werden, wenn Ablaufprogramm 3 beendet ist. In naher Zukunft werden auch Überlappungen von Ablaufprogrammen möglich sein.

### 5.1.5 Eingang für Fehlerquittierung (IN8)

Eingang 8 ist für Fehlerquittierungen reserviert.

Nach Auftreten eines Fehlers, z.B. wenn ein NOT-AUS-Schalter getroffen wurde, blinkt die Fehler-LED und die Steuerung verharrt in dieser Position, bis an Eingang 8 eine positive Quittierung für mindestens 0,1 Sekunden erfolgt. Die Quittierung darf natürlich erst dann erfolgen, wenn die Fehlerursache beseitigt wurde. Falls ein NOT-AUS-Schalter getroffen wurde, kann nach der Quittierung eine Referenzfahrt ausgelöst werden. Die Anlage fährt dann wieder aus dem NOT-AUS-Schalter. Für diese spezielle Situation (NOT-AUS-Schalter getroffen) kann auch ein Automatismus programmiert werden. Hierfür wird Eingang 8 mit einem Dauersignal (+5V bis +24V) belegt und es wird eine „Referenzfahrt nach Fehlerquittierung“ programmiert (siehe Befehlsreferenz 10.17).

## 5.3 Digitale Ausgänge

Die SMC-dMOT-42 stellt zwei Transistor-Ausgänge zur Verfügung. Diese können mit max. 100mA belastet werden. Für höhere Ausgangsströme müssen geeignete Maßnahmen außerhalb der SMC-dMOT-42 durchgeführt werden (z.B. Anschluss eines Lastrelais, das wiederum einen Stromkreis mit höheren Leistungen aktiviert).

Die Transistor-Ausgänge liefern eine Spannung, die der SMC-dMOT-42 von extern am dafür vorgesehenen Kontakt zugeführt wird. Es können Spannungen von +5V bis +24V zugeführt werden. Diese Spannung liegt bei gesetztem Ausgang am entsprechenden Kontakt an. Die Befehle für Setzen und Rücksetzen der Ausgänge entnehmen Sie bitte der „Befehlsreferenz 10.3“.

OUT1: Ausgang 1 → ON = Pegel von UEXT → OFF = 0V

OUT2: Ausgang 2 → ON = Pegel von UEXT → OFF = 0V

UEXT: Externe Spannung für OUT1 und OUT2

GND: Masse

## 5.3 Kommunikations-Interface

### 5.3.1 USB

Das USB-Interface dient zur Anbindung der SMC-dMOT-42 an einen PC.



Abb. 2:  
USB-Interface 2.0 (Typ B)

### 5.3.2 RS485

Über die gezeigten Klemmen kann eine RS485-Kommunikation (Half-Duplex) stattfinden. Ist die SMC-dMOT-42 der letzte Teilnehmer am RS485-Bus, kann mit dem DIP-Schalter der Bus abgeschlossen werden (Stellung ON).

Die RS485-Kommunikation wird aktuell in der Firmware nicht unterstützt.

### 5.3.3 Steckplatz für Bluetooth-Interface

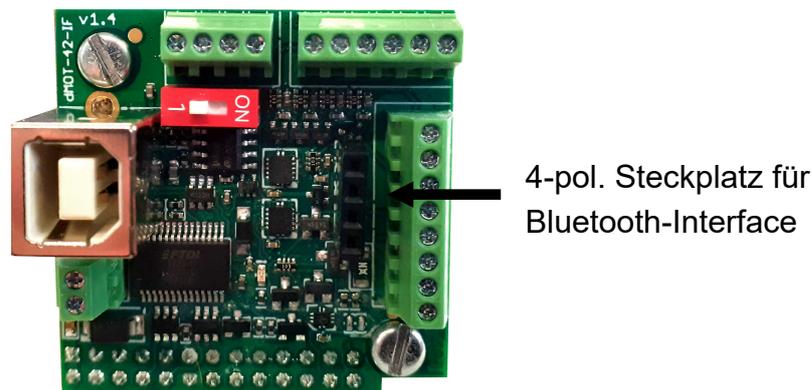


Abb 3: Steckplatz für Bluetooth-Interface

Für die SMC-dMOT-42 ist optional ein Bluetooth-Interface erhältlich.

### 5.4 Integrierter Festwertspeicher

Der integrierte Festwertspeicher kann 10 Programme mit jeweils 32 kB für flankengetriggerte Eingänge (positive und negative Flanke) aufnehmen. Für die Programmierung werden die Befehle aus der Befehlsreferenz verwendet und als einfache Textdatei abgelegt. Mit der kostenlosen Software „StandAlone-ProgrammWriter-2.0“ können die Textdateien, die den Programmablauf enthalten, in den Festwertspeicher der SMC-dMOT-42 übertragen werden.

### 5.5 Stromversorgung

Die SMC-dMOT-42 arbeitet mit einer Versorgungsspannung von 12V DC bis maximal 48V DC. Der benötigte Strom wird maßgeblich vom eingestellten Phasenstrom bestimmt.

Je höher die Versorgungsspannung ist, desto höhere Motorgeschwindigkeiten können erreicht werden. Dies führt jedoch auch zu höheren Verlustleistungen in den Motorendstufen und damit zu einer höheren Erwärmung der Elektronik.

### 5.6 Anschlussbelegung

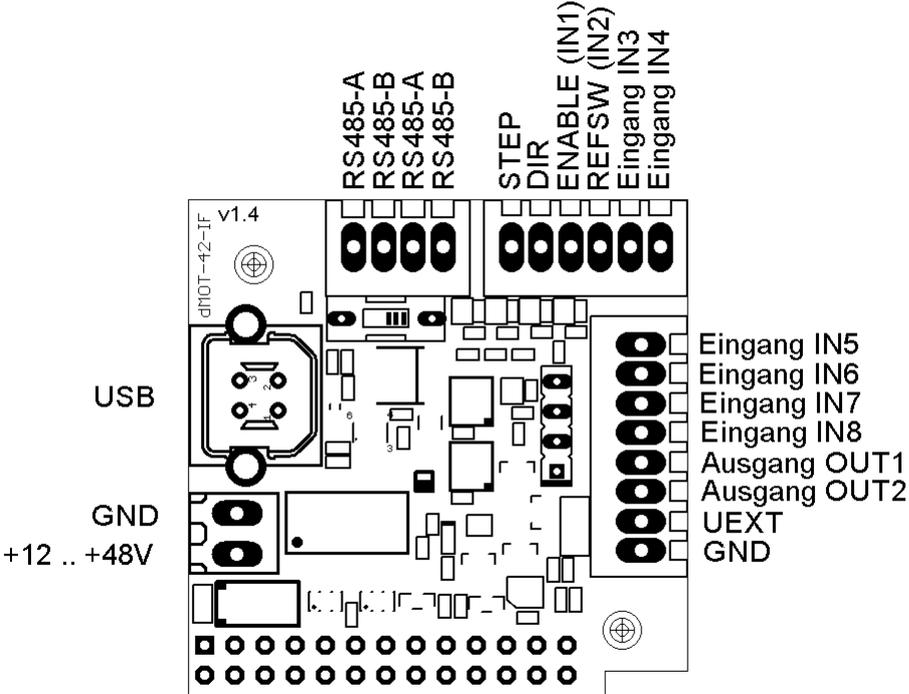


Abb. 4: Anschlussbelegung Interface-Platine

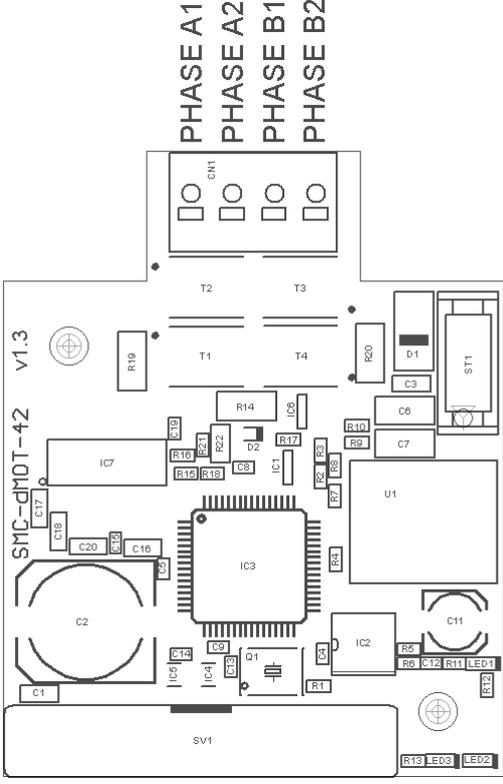


Abb. 5: Anschlussbelegung Motorplatine

## 6. Technische Daten

### 6.1 Steuerung:

Spannungsversorgung	12V DC bis 48V DC / Strom nach Bedarf
Stromaufnahme ohne Motor	30 mA bei 24V DC
Zulässige Umgebungstemperatur	0°C bis +40°C
Zulässige Luftfeuchtigkeit	max. 85%, nicht kondensierend
Digitale Eingänge	10 (High: +5V ... +24V, Low: 0V ... +0,8V)
Digitale Ausgänge	2 (+5V ... +24V, max. 100 mA)
Kommunikationsinterface	USB 2.0 (Typ B), RS485**, Bluetooth
Anzahl Motoranschlüsse	1 Stück, bipolarer Schrittmotor, 2 Phasen
Maximaler Phasenstrom	2 A (100%)   2,5 A Peak (125%)
Automatische Ruhestromreduzierung	einstellbar 0% bis 25% von 2 A
Schrittauflösung	1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256
Anschlüsse für Referenzschalter	1 mech. oder elektronischer Schließer
Festwertspeicher on Board	32 MBit
Abmessungen	42 mm x 53 mm

\* 48V DC absoluter Maximalwert

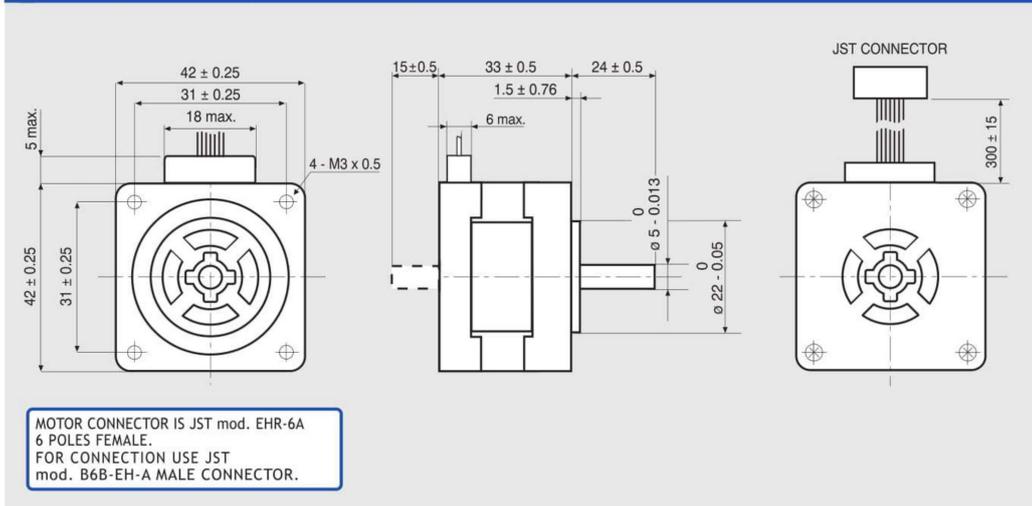
\*\* in Vorbereitung

## 6.2 Schrittmotor:

# 103-H5205-0351

SANYO DENKI  
SANMOTION

### Dimensions (Unit:mm)

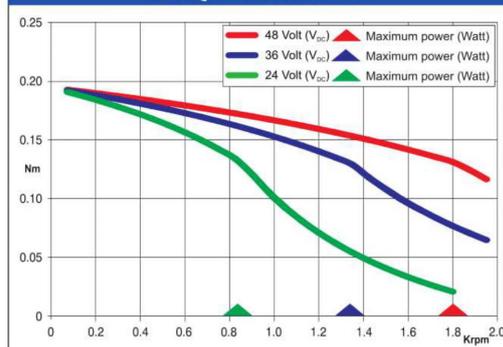


### FEATURES

MODEL	103-H5205-0351 (103-H5205-0312)
BASIC STEP ANGLE	1.8° ± 0.09°
BIPOLAR CURRENT (Amp)	0.7*
UNIPOLAR CURRENT (Amp)	1.0
RESISTANCE (Ohm)	3.6
INDUCTANCE (mH)	3.3
BIPOLAR HOLDING TORQUE (Ncm)	25
UNIPOLAR HOLDING TORQUE (Ncm)	19
ROTOR INERTIA (Kgm <sup>2</sup> × 10 <sup>-7</sup> )	36
THEORETICAL ACCELERATION (rad × sec. <sup>-2</sup> )	69800
BACK E.M.F. (V/Krpm)	37
MASS (Kg)	0.23
LEADS CODE	IV

Codes between brackets refer to double shaft models.

### TORQUE/SPEED CURVE



\*Bipolar series connection.



# EMIS

GmbH

EMIS GmbH  
92637 Weiden  
Zur Drehscheibe 4  
E-Mail: info@emisgmbh.de  
Tel.: 0961 32040

## **7. Zubehör**

### 7.1. Bluetooth-Interface

Mit dem optional erhältlichen Bluetooth-Interface kann die Kommunikation mit der SMC-dMOT-42 drahtlos erfolgen. Als Steuergerät eignet sich jedes Gerät mit Bluetooth-Interface (z.B. Tablet, Smartphone, usw.).

## **8. Software**

Auf unserer Homepage steht in Kürze ein Downloadmanager für Firmware-Updates zur Verfügung. Damit können Sie die SMC-dMOT-42 stets aktualisieren.

Eine Anwender-Software zur Parametrierung und Ansteuerung der SMC-dMOT-42 wird aktuell entwickelt und steht demnächst ebenfalls zur Verfügung.

Des Weiteren ist eine Anwender-Software „StandAlone-ProgrammWriter-2d0“ zur Übertragung von Programmabläufen in den Festwertspeicher verfügbar.

## **9. Standalone-Betrieb**

Die SMC-dMOT-42 kann einen Programmablauf ohne PC abarbeiten. Hierfür ist es notwendig, dass die entsprechenden Kommandos im vorhandenen Festwertspeicher hinterlegt werden.

Die Kommandos können als einfacher Text in einer Textdatei abgelegt werden. Diese Textdatei kann anschließend mit unserem kostenlos zur Verfügung gestelltem Programm „StandAlone-ProgrammWriter-2d0“ in den Festwertspeicher übertragen werden. Alternativ kann zur Übertragung der Kommandos ein einfaches Terminalprogramm (z.B. YAT) verwendet werden. Bei Verwendung eines Terminalprogramms ist zu beachten, dass der Programmheader mit dem Steuerzeichen 02 hex, Programmeinleitung und die

Kommandos mit dem Steuerzeichen 0D hex und das Programmende mit dem Steuerzeichen 03 hex abgeschlossen werden.

Hinterlegt werden können Programmabläufe, die den Eingängen 3 bis 7 zugeordnet werden können. Die entsprechenden Befehle entnehmen Sie bitte der Befehlsreferenz. Nachfolgend wird ein einfaches Beispiel für einen Programmablauf gezeigt.

Ausgelöst wird der Programmablauf mit einem Impuls (mindestens 0,01s) oder Dauersignal am entsprechenden Eingang. Nach der Abarbeitung des Programms befindet sich die SMC-dMOT-42 wieder in Bereitschaft.

Soll ein Programmablauf abgebrochen werden, muss für mindestens 0,01s das Dauersignal am Eingang „Enable“ unterbrochen werden. Zu beachten ist, dass der Enable-Eingang vorher als NOT-AUS-Eingang definiert werden muss.

Ablauf:

1. Programmablauf mit Texteditor erzeugen und im Textformat abspeichern.

Zeilen, die mit einem Hochkomma eingeleitet werden, werden von dem Programm „StandAlone-ProgramWriter-2d0“ als Kommentar interpretiert und nicht in den Festwertspeicher übertragen. Alle Befehle werden mit einem Carriage Return abgeschlossen (nächste Zeile).

Beispiel 1:

```
'Programm schreiben für Eingang 3, positive Flanke, einleiten
*PWIN3h<CR>
IJ3h,1<CR>
'Programmheader schreiben und mit <STX> abschließen
Programm Eingang IN3, positive Flanke: Motordauerlauf<STX>
'Enable-Eingang als NOT-AUS verwenden
##EN,1<CR>
'Automatische Referenzfahrt nach Fehlerquittierung
##HX,1<CR>
'32 Mikroschritte pro Vollschrift
D,x32<CR>
'50% Arbeitsstrom (50% von 4A) und 0% Haltestrom
c,x50,0<CR>
'Startgeschwindigkeit 3200 Mikroschritte/s = 100 Vollschritte pro Sekunde
#S19200<CR>
'Endgeschwindigkeit 19200 Mikroschritte/s = 600 Vollschritte pro Sekunde
#E1,16000<CR>
'Rampe 100 ms (Hochlaufzeit von Start- zur Endgeschwindigkeit)
#R100<CR>
'Motordauerlauf, positive Richtung
L1,xd<CR>
```

Programm-Eingang-3\_positive\_Flanke.txt

## Beispiel 2:

```
'Programm schreiben für Eingang 3, negative Flanke, einleiten
*PWIN3I
IJ3I,1
'Programmheader schreiben und mit STX abschließen
Programm Eingang IN3, negative Flanke: Motor stoppen<STX>
'Motor anhalten mit Rampe
@B
```

Programm-Eingang-3\_negative\_Flanke.txt

2. Textdatei mit „StandAlone-ProgrammWriter-2d0“ an SMC-dMOT-42 senden.

Die Handhabung des Programms „StandAlone-ProgrammWriter-2d0“ entnehmen Sie bitte der Anleitung „StandAlone-ProgrammWriter-2d0-Kurzbeschreibung.pdf“.

3. Programm mit Pegel am Eingang „IN3“ starten.

Wird an Eingang IN3 ein Taster angeschlossen (einseitig an positiver Spannung +5V bis +24V) und dieser betätigt, läuft der Motor solange, bis der Taster wieder losgelassen wird.

Beim Betätigen des Tasters wird ein positiver Pegel an Eingang IN3 gelegt und das Programm „ Programm-Eingang-3\_positive\_Flanke“ gestartet (Motordauerlauf). Beim Loslassen des Tasters wird der Pegel wieder zurückgenommen und das Programm „Programm-Eingang-3\_negative\_Flanke“ gestartet (Motor mit Rampe anhalten). Geschwindigkeit und Schrittauflösung müssen vor Auslösung der Motorbewegung gesetzt werden.

## 10. Befehlsreferenz

Alle Befehle können mit einem Terminalprogramm vom PC an die SMC-dMOT-42 gesendet werden. Die seriellen Parameter des Terminalprogramms sind wie folgt einzustellen:

- Baudrate 115200
- Datenbits 8
- Stoppbits 1
- Handshake Nein

Die Befehle werden mit Carriage Return, im nachfolgenden Text <CR>, abgeschlossen. Die Antworten von der Steuerung werden mit ACK, NAK oder BEL abgeschlossen, abhängig vom gesendeten Befehl und Betriebszustand der Steuerung.

### 10.1 Versionsabfrage

Befehl: @V<CR>  
Antwort: @V SMC-dMOT-v1.064<ACK>  
Rückgabe der Firmware-Version

### 10.2 Eingänge abfragen

Befehl: @I3<CR>  
Nach dem I wird die Eingangsnummer (1-10) angegeben.  
Antwort: @I3 0<ACK>  
Die Antwort zeigt den Logikpegel am abgefragten Eingang.

Befehl: @IG<CR>  
Mit ,G' können sämtliche Eingänge auf einmal abgefragt werden.  
Antwort: @IG --10000000<ACK>  
Die Antwort zeigt die Logikpegel der Eingänge. Im Beispiel führt Eingang 3 High-Pegel (Eingang 1-10 von links nach rechts). Eingang 1 (STEP) und Eingang 2 (DIR) können nicht abgefragt werden, deshalb ,-' an der entsprechenden Position.

### 10.3 Ausgänge setzen

Befehl: `A1,1<CR>`

Nach dem A folgt die Ausgangsnummer (1-7). Nach dem Komma das Setz- bzw. Rücksetzkommando.

Mit „A1,1<CR>“ wird gesetzt, mit „A1,0<CR>“ zurückgesetzt.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen <ACK> quittiert.

### 10.4 Startgeschwindigkeit setzen

Befehl: `#S200<CR>`

Startgeschwindigkeit 200 Mikroschritte pro Sekunde.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen <ACK> quittiert.

### 10.5 Arbeitsgeschwindigkeit setzen

Befehl: `#E1,1000<CR>`

Bei Vektorfahrt Arbeitsgeschwindigkeit 1000 Mikroschritte pro Sekunde.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen <ACK> quittiert.

Befehl: `#E9,400<CR>`

Bei Referenzfahrt Arbeitsgeschwindigkeit 400 Mikroschritte pro Sekunde.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen <ACK> quittiert.

### 10.6 Rampendauer setzen

Befehl: `#R100<CR>`

Rampendauer 100 ms.

Der Hochlauf des Schrittmotors von der Startgeschwindigkeit zur Arbeitsgeschwindigkeit dauert 100 ms. Das Abbremsen auf die Startgeschwindigkeit dauert ebenfalls 100 ms.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen <ACK> quittiert.

## 10.7 Schrittauflösung setzen

Befehl: `D,x32<CR>`

Schrittauflösung X-Motor 32 Mikroschritte. Möglich Schrittauflösungen sind 1, 2, 4, 8, 16, 32, 64, 128, 256 Mikroschritte.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen `<ACK>` quittiert.

## 10.8 Motorstrom setzen

Befehl: `c,x100,20<CR>`

Arbeitsstrom X-Motor 100% vom Nennwert.

Ruhestrom X-Motor 20% vom Nennwert.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen `<ACK>` quittiert.

## 10.9 Referenzfahrt

Befehl: `$Hx<CR>`

Referenzfahrt X-Motor

Antwort: `$Hx<NAK><ACK>`

`<NAK>` bei Beginn der Referenzfahrt,

`<ACK>` nach Beendigung der Referenzfahrt.

## 10.10 Offset nach Referenzfahrt setzen

Befehl: `#Ox,10<CR>`

Nach der X-Referenzfahrt 10 Schritte aus dem Schalter fahren.

Antwort: `<ACK>`

Der Befehl wird mit einem einfachen `<ACK>` quittiert.

## 10.11 Vektorfahrt

Befehl: `L1,x1000<CR>`

Der X-Schrittmotor fährt 1000 Mikroschritte relativ zu seiner aktuellen Position (positive Drehrichtung).

Antwort: `<NAK> <ACK>`

`<NAK>` bei Beginn der Vektorfahrt,

`<ACK>` nach Beendigung der Vektorfahrt.

Befehl: **L1,x-1000<CR>**  
Der X-Schrittmotor fährt 1000 Mikroschritte relativ zu seiner aktuellen Position (negative Drehrichtung).

Antwort: **<NAK> <ACK>**  
<NAK> bei Beginn der Vektorfahrt,  
<ACK> nach Beendigung der Vektorfahrt.

Befehl: **L1,X1000<CR>**  
Der X-Schrittmotor wird zur Absolutposition 1000 Mikroschritte positioniert. Die Vektorlänge und die Drehrichtung werden durch die aktuelle Position bei Befehlsannahme bestimmt.

Antwort: **<NAK> <ACK>**  
<NAK> bei Beginn der Vektorfahrt,  
<ACK> nach Beendigung der Vektorfahrt.

Befehl: **L1,xd<CR>**  
Der X-Schrittmotor läuft solange, bis er mit „@B<CR>“ oder „@S<CR>“ gestoppt wird → Dauerlauffunktion. Ein Minus vor dem „x“ bewegt den Motor in die entgegengesetzte Richtung.

Antwort: **<NAK>**  
<NAK> bei Beginn der Vektorfahrt,

#### 10.12 Vektorfahrt mit Rampe abbrechen

Befehl: **@B<CR>**

Antwort: **@B<ACK><DC1><ACK>**  
Der Befehl wird sofort mit <ACK> und nach Motorstillstand nochmal mit <DC1> und <ACK> quittiert.

#### 10.13 Vektorfahrt sofort abbrechen

Befehl: **@S<CR>**

Antwort: **@RS<ACK><DC1><ACK>**  
Der Befehl wird sofort mit <ACK> und nach Motorstillstand nochmal mit <DC1> und <ACK> quittiert

## 10.14 Positionsabfrage

Befehl: @LX<CR>  
Nach dem @L folgt die Achsangabe, hier immer X.

Antwort: @LX 0<ACK>  
Der Befehl wird mit der aktuellen Position in Mikroschritten und einem <ACK> quittiert.

## 10.15 Wartezeit setzen

Befehl: W500<CR>  
Nach dem W folgt die Wartezeit in Millisekunden, hier 500 ms.

Antwort: <ACK>  
Der Befehl wird mit einem einfachen <ACK> quittiert.

## 10.16 Eingang für Software – NOT-AUS aktivieren/deaktivieren

Befehl: ##EN,1<CR>  
Nach dem „EN,“ folgt die Aktivierung [1] oder Deaktivierung [0] des Enable-Eingangs als NOT-AUS-Eingang.

Antwort: <ACK>  
Der Befehl wird mit einem einfachen <ACK> quittiert.

## 10.17 Automatische Referenzfahrt nach NOT-AUS und Fehlerquittierung

Befehl: ##HX,1<CR>  
Nach dem „HX,“ folgt die Aktivierung [1] oder Deaktivierung [0] einer automatischen Referenzfahrt nach einer Fehlerquittierung.

Antwort: <ACK>  
Der Befehl wird mit einem einfachen <ACK> quittiert.

## 10.18 Stand-Alone – Ablaufprogramm

Für die Übertragung eines Programms in den Festwertspeicher kann das kostenlose Programm „StandAlone-ProgrammWriter-2d0“ verwendet werden. Alternativ kann ein einfaches Terminalprogramm (z.B. YAT) verwendet werden.

### 10.18.1 Aufbau eines Ablaufprogramms

1. Programm schreiben, inklusive Eingangsnummer und Flankenpolarität (siehe oben aufgeführtes Beispiel) einleiten.
2. Programmheader schreiben → hier kann eine kurze Programmbeschreibung erfolgen. Abschluss mit der Zeichenfolge „<STX>“ (siehe

o.a. Beispiel). Bei Verwendung eines Terminalprogramms muss die Zeichenfolge „<STX>“ durch das Steuerzeichen 02 hex ersetzt werden.

3. Gewünschte Befehle einfügen (siehe o.a. Beispiel)
4. Bei Verwendung eines Terminalprogramms muss die Programmierung mit dem Steuerzeichen 03 hex beendet werden. Wird das Programm mit dem Programm „StandAlone-ProgrammWriter-2d0“ übertragen, kann der Abschluss mit dem Steuerzeichen 03 hex entfallen, das wird von dem Programm automatisch eingefügt.

#### 10.18.2 Befehle für ein Ablaufprogramm

Alle Befehle aus der Befehlsreferenz können verwendet werden.