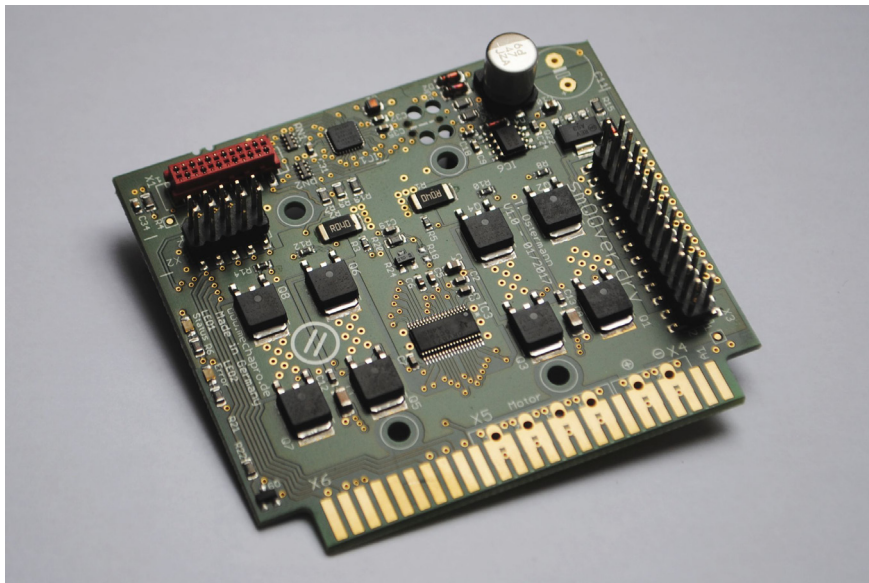


Dokumentation zum
Schrittmotor-Modul smOOVer.drv
Version 1.0, Stand 30.12.2017



Eigenschaften

smOOVer.drv ist ein Schrittmotor-Modul mit einer modernen, hoch effizienten Mikroschritt-Endstufe zur Ansteuerung von 2-phasigen Schrittmotoren. Das Modul wurde zur einfachen Integration einer oder mehrerer leistungsstarker Schrittmotor-Endstufen in eine Schaltung des Anwenders (Zielschaltung) entwickelt. Die Karte kann liegend an den vorgesehenen Befestigungsbohrungen montiert werden, als Steckkarte in einem Slotstecker betrieben werden oder in einer Bestückungsvariante auch als Einschub in einem Bopla Combinorm-Gehäuse eingesetzt werden.

Markenrechtlicher Hinweis

smOOVer[®] ist eine eingetragene Marke der mechapro GmbH

Kontakt:

mechapro GmbH

Im Winkel 34

52146 Würselen

Tel.: +49/2405/47937-20

Fax: +49/2405/47937-23

Mail: info@mechapro.de

Web: www.mechapro.de

Inhalt

Technische Daten	2
Montage.....	3
Anschlüsse.....	3
Inbetriebnahme	6
Betriebsarten	6
Belegung Schieberegister (Modus 3)	10
Strom- und Mikroschritteinstellung (Modus 3).....	11
LEDs	11
Konfigurationsregister.....	12
SPI-Interface.....	14
RS-232 Schnittstelle zur Programmierung.....	14
Firmware-History.....	15
Abmessungen.....	16

Technische Daten

Spannungsversorgung: 12-50V=

Motorstrom: einstellbar 0,25-4,0A (effektiv) in 0,25A Schritten

max. Stromaufnahme: 3,0A

Motorausgang kurzschlussfest

Signalein- und Ausgänge in 5V-Logik, ohne galvanische Trennung

Haftung, EMV-Konformität

Das vorliegende Endstufen-Modul smOOver.drv wurde unter Zugrundelegung aller zum Zeitpunkt der Entwicklung üblichen und bekannten Richtlinien entworfen und sehr aufwendig und sorgfältig getestet. Eine Garantie für fehlerfreie Funktion in der Applikation des Anwenders kann dennoch nicht gegeben werden. mechapro[®] sichert zu, dass smOOver.drv in Verbindung mit geeigneten Steuerungen und geeigneten mechanischen Komponenten im Sinne der Beschreibung und Benutzungsanleitung grundsätzlich für den vorgesehenen Zweck geeignet ist.

Jede Haftung für Folgeschäden oder Schäden aus entgangenem Gewinn, Betriebsunterbrechung, Verlust von Informationen usw. ist ausgeschlossen.

smOOver.drv ist zur Integration in weitere, vom Anwender zu realisierende Schaltungen vorgesehen und somit kein eigenständiges Gerät. Eine CE-Kennzeichnung erfolgt daher nicht. Selbstverständlich wurden bei der Schaltungsentwicklung alle möglichen Maßnahmen für einen EMV-gerechten Aufbau ergriffen.

Montage- und Anschlussmöglichkeiten

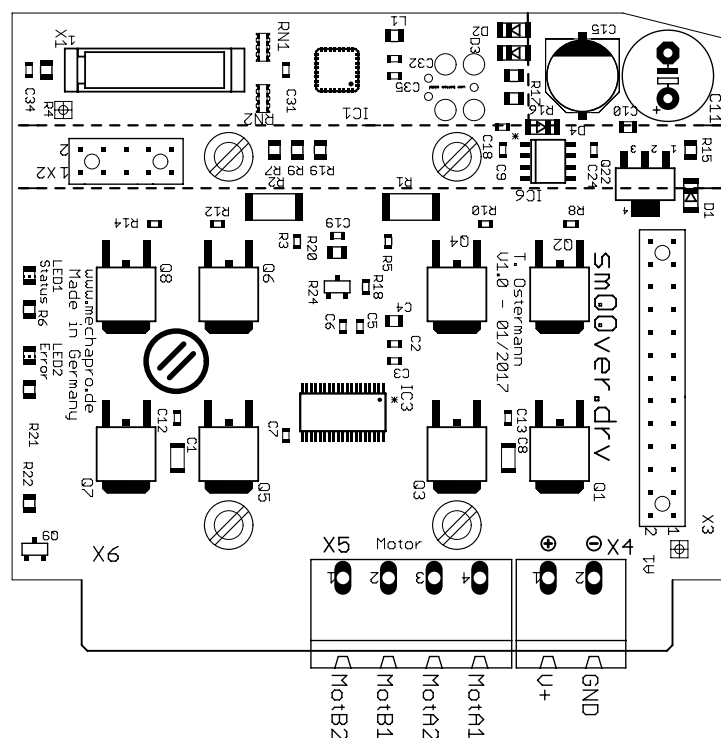
Für die Montage und den elektrischen Anschluss des Moduls sind drei verschiedene Möglichkeiten vorgesehen:

- Verwendung des Slotsteckers: Das Modul wird stehend im Slotstecker montiert (90° zur Basisplatte), alle elektrischen Verbindungen erfolgen über den Kantenstecker X6.
- Sandwich-Montage: Das Modul wird mit seinen Stiftleisten kopfüber in passende Buchsenleisten auf der Trägerplatte gesteckt. Spannungsversorgung und Motorausgangssignale werden über X3 verbunden, alle Steuersignale über X2.
- Einschub in Gehäuse Bopla Combinorm (nur mit optional bestückten Phoenix Combicon-Steckern): Die Verbindung zur Steuerplatine erfolgt über ein Flachbandkabel am Anschluss X1. Die Spannungsversorgung erfolgt über den Stecker X4, der Motor wird an den Stecker X5 angeschlossen.

Hinweis: Die Sandwich-Montage ermöglicht eine besonders niedrige Bauhöhe, erfordert aber bei hohen Strömen eine besondere Betrachtung der Kühlsituation (ggf. Zwangskühlung über erzwungenen Luftstrom, Zirkulation über Bohrungen in der Hauptplatte o.ä.). Die Temperatur des Moduls (aktuelle Temperatur und gespeicherter Maximalwert) kann via SPI ausgelesen werden (siehe Register-Beschreibung).

Anschlüsse

Im Folgenden finden Sie eine Kurzübersicht über die Funktion der verschiedenen Anschlüsse. Anschließend sind die genauen Belegungen der mehrpoligen Steckverbinder wiedergegeben.



- X1 ist die Verbindung der Logiksignale zum smOOver.drv Modul, falls das Modul als Einschub in einem Bopla Combinorm-Gehäuse verwendet wird. Für diese Betriebsart müssen Motor und Versorgungsspannung für das Modul direkt am Modul angeschlossen werden (Bestückungsoption, lieferbar auf Anfrage).
- X2 ist die Stiftleiste für die Logiksignale, alternativ zu X1
- X3 ist die Stiftleiste für Spannungsversorgung und Motoranschluss.
- X4 ist der Stecker zur Versorgung des Endstufen-Moduls (optional).
- X5 ist der Stecker zum Anschluss des Motors (optional).
- X6 ist der Kantenstecker mit allen für den Betrieb erforderlichen Signalen.

X1 I/O-Signale (externes Endstufenmodul)

Pin-Nr.	Signal	Funktion
Pin 1	SCK_Ext	SPI Takt
Pin 2	MISO_Ext	SPI MasterIn, SlaveOut
Pin 3	MOSI_Ext	SPI MasterOut, SlaveIn
Pin 4	!CS_Ext	SPI Chipselect für Betrieb mit externem SPI Master
Pin 5	!CS_SREG	SPI Chipselect für Betrieb mit Schieberegistern (Controller auf smOOver.drv ist SPI Master)
Pin 6	!Error	Open-Collector Ausgang zur Fehler-Signalisierung
Pin 7	/Step	Takteingang. Steigende Flanke löst einen Takt aus, Ruhepegel HIGH
Pin 8	DIR	Drehrichtung
Pin 9	!Disable	Signal zum Abschalten der Endstufe bei low-Pegel
Pin 10	GND	Masse Spannungsversorgung
Pin 11	VCC	Eingang 5V Logikspannung, nur für smOOver.prg!
Pin 12	V+	Versorgungsspannung des Moduls, nicht verwendet. Kann zur Versorgung externer Schaltungsteile verwendet werden.

X2 / I/O-Signale

Pin-Nr.	Signal	Funktion
Pin 1	SCK_Ext	SPI Takt
Pin 2	MISO_Ext	SPI MasterIn, SlaveOut
Pin 3	MOSI_Ext	SPI MasterOut, SlaveIn
Pin 4	!CS_Ext	SPI Chipselect für Betrieb mit externem SPI Master
Pin 5	!CS_SREG	SPI Chipselect für Betrieb mit Schieberegistern (Controller auf smOOver.drv ist SPI Master)
Pin 6	!Error	Open-Collector Ausgang zur Fehler-Signalisierung
Pin 7	/Step	Takteingang. Steigende Flanke löst einen Takt aus, Ruhepegel HIGH
Pin 8	DIR	Drehrichtung
Pin 9	!Disable	Signal zum Abschalten der Endstufe bei low-Pegel
Pin 10	GND	Masse Spannungsversorgung

X3 / Spannungsversorgung Modul

Pin-Nr.	Signal	Funktion
Pin 1-4	V+	Positive Versorgungsspannung
Pin 5-8	MotA1	Ausgang 1 Motorwicklung A
Pin 9-12	MotA2	Ausgang 2 Motorwicklung A
Pin 13-16	MotB1	Ausgang 1 Motorwicklung B
Pin 17-20	MotB2	Ausgang 2 Motorwicklung B
Pin 21-26	GND	Negative Versorgungsspannung (Masse)

X4 / Spannungsversorgung Modul (optional)

Pin-Nr.	Signal	Funktion
Pin 1	V+	Positive Versorgungsspannung
Pin 2	V-	Negative Versorgungsspannung (Masse)

X5 / Motoranschluss (optional)

Pin-Nr.	Signal	Funktion
Pin 4	MotA1	Ausgang 1 Motorwicklung A
Pin 3	MotA2	Ausgang 2 Motorwicklung A
Pin 2	MotB1	Ausgang 1 Motorwicklung B
Pin 1	MotB2	Ausgang 2 Motorwicklung B

X6 / Kantenstecker

Pin-Nr.	Signal	Funktion
A3, A4	GND	Negative Versorgungsspannung (Masse)
B3, B4	GND	Negative Versorgungsspannung (Masse)
A5, A6	V+	Positive Versorgungsspannung
B5, B6	V+	Positive Versorgungsspannung
A8, A9	MOTA1	Ausgang 1 Motorwicklung A
B8, B9	MOTA1	Ausgang 1 Motorwicklung A
A10, A11	MOTA2	Ausgang 2 Motorwicklung A
B10, B11	MOTA2	Ausgang 2 Motorwicklung A
A12, A13	MOTB1	Ausgang 1 Motorwicklung B
B12, B13	MOTB1	Ausgang 1 Motorwicklung B
A14, A15	MOTB2	Ausgang 2 Motorwicklung B
B14, B15	MOTB2	Ausgang 2 Motorwicklung B
A16	!Reset	Reset-Eingang des internen Mikrocontrollers, low-aktiv
B16	VCC	Eingang 5V Logikspannung, nur für smOOver.prg!
A17	!Disable	Signal zum Abschalten der Endstufe bei low-Pegel
B17	GND	Negative Versorgungsspannung (Masse)
A18	/Step	Takteingang. Steigende Flanke löst einen Takt aus, Ruhepegel HIGH
B18	DIR	Drehrichtung
A19	!CS_SREG	SPI Chipselect für Betrieb mit Schieberegistern
B19	!Error	Open-Collector Ausgang zur Fehler-Signalisierung
A20, A21	n.c.	Nicht beschalten
B20, B21	n.c.	Nicht beschalten

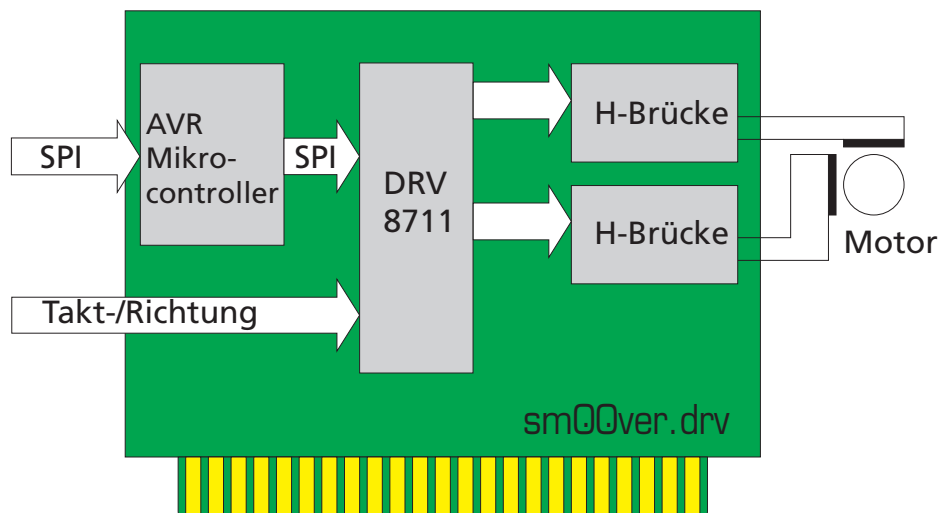
A22	GND	Masse Spannungsversorgung
B22	GND	Masse Spannungsversorgung
A23	Rev_Detect	Liegt auf Masse, wenn ein Modul falsch herum gesteckt wird
B23	Rev_Detect	Liegt auf Masse, wenn ein Modul falsch herum gesteckt wird
A24	MOSI_Ext	SPI MasterOut, SlaveIn
B24	!CS_Ext	SPI Chipselect für Betrieb mit externem SPI Master
A25	SCK_Ext	SPI Takt
B25	MISO_Ext	SPI MasterIn, SlaveOut

Inbetriebnahme

Bei Verwendung des Moduls in der Betriebsart „stand alone“ muss das Modul zunächst mit Hilfe des Programmieradapters smOOver.prg an einem PC mit USB-Anschluss konfiguriert werden (siehe nachfolgende Abschnitte). Bei den anderen möglichen Betriebsarten erfolgt die Konfiguration beim Hochlauf oder zur Laufzeit in der Zielschaltung. Vor dem Einbau des Moduls die Spannungsversorgung abschalten. Das Modul ist nach dem Einstecken in die Zielschaltung grundsätzlich sofort betriebsbereit.

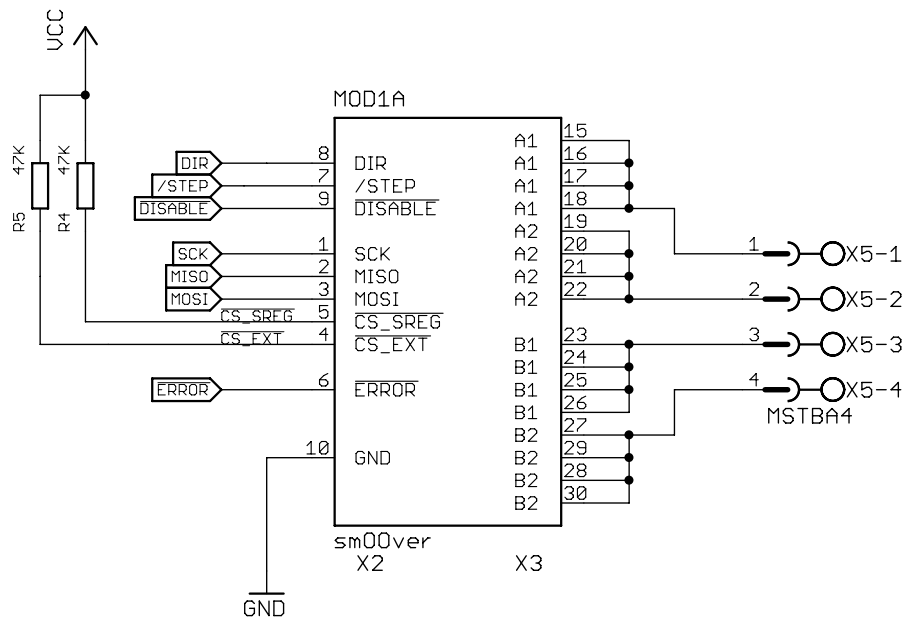
Betriebsarten

Im Folgenden werden die verschiedenen Betriebsarten des Moduls aufgelistet, die dann in den nachfolgenden Abschnitten näher beschrieben werden. In allen Betriebsarten wird die Bewegung des Motors über die Signale Takt, Richtung und Enable (Freigabe) gesteuert. Unterschieden wird je nach Betriebsart, wie die Einstellungen für Motorstrom, Mikroschritt-Auflösung und weitere Konfigurationsoptionen geladen werden. Diese werden via SPI-Bus auf den auf dem Modul verwendeten Treiberchip DRV8711 geladen.



Blockschaltbild smOOver.drv

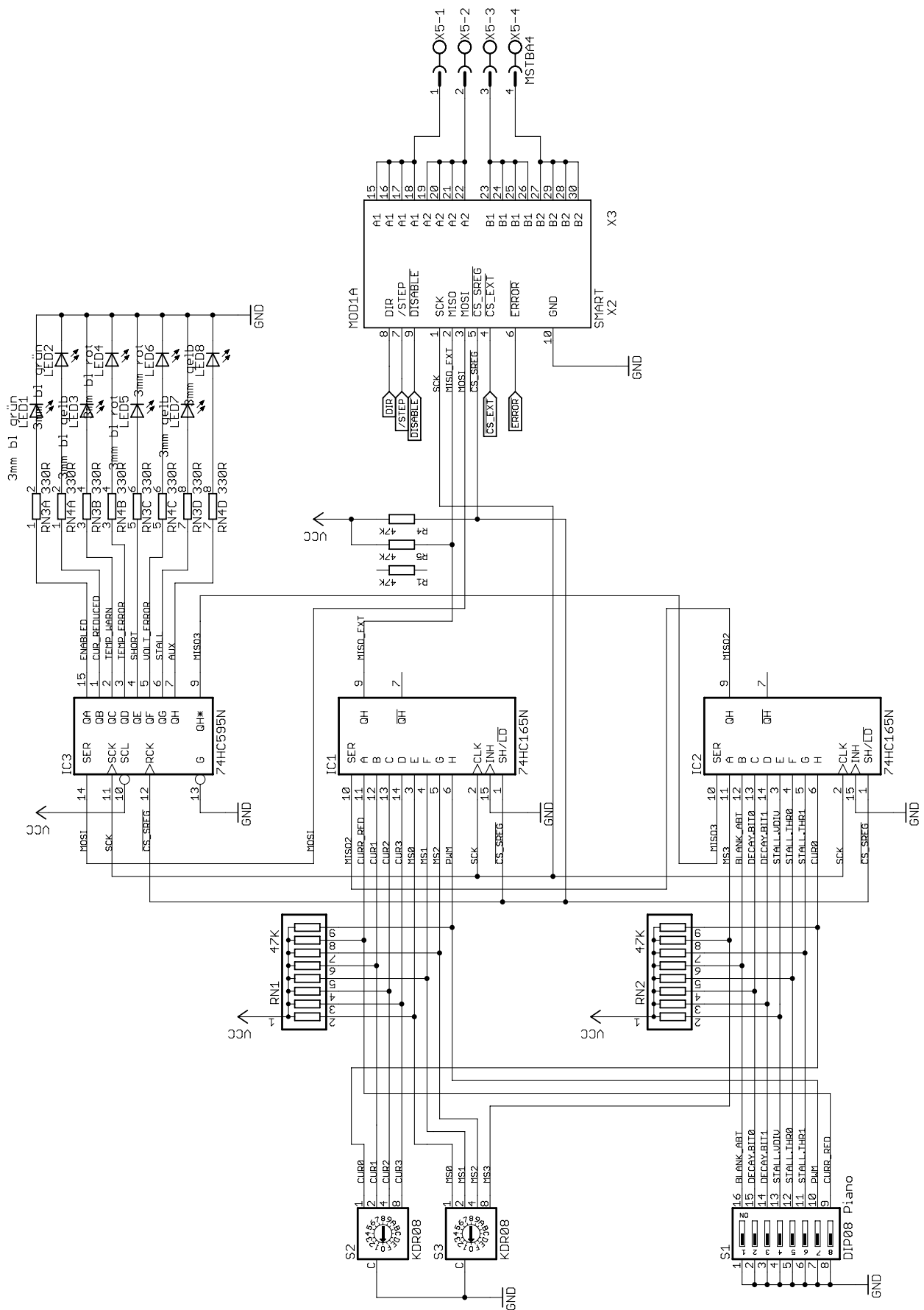
- Stand-alone (Modus 0b11=3): Die SPI-Leitungen werden nicht verwendet und bleiben unbeschaltet (high, ggf. mit Pullup-Widerstand). Beim Hochlauf des Moduls lädt der interne Mikrocontroller die in seinem EEPROM gespeicherten Einstellungen in den Treiber-Chip. Änderungen an der Konfiguration werden außerhalb der Zielschaltung mit dem Programmierinterface smOOver.prg vorgenommen (am PC via Terminal-Programm).



Prinzipschaltbild Stand-alone Betrieb (SCK, MISO und MOSI werden nicht verwendet)

- Schieberegister (Modus 0b11=3): Der Mikrocontroller auf dem Modul fungiert als SPI-Master und lädt die gewünschten Einstellungen aus extern angeschlossenen Schieberegistern. Es können ein oder zwei Eingangsregister (74HC165) und ein oder kein Ausgangsregister (74HC595) angeschlossen werden. Die Belegung der Ein- und Ausgänge ist im nachfolgenden Abschnitt beschrieben. An die Eingangsregister können z.B. DIP-Schalter oder Codierschalter angeschlossen werden, mit denen die Einstellungen zur Laufzeit angepasst werden können. Das Ausgangsregister kann z.B. LEDs zur Statusanzeige ansteuern. Das Evaluierungsboard smOOver.tst veranschaulicht diese Betriebsart und kann verwendet werden, um die Eignung des smOOver.drv Moduls für die Applikation des Kunden zu erproben.

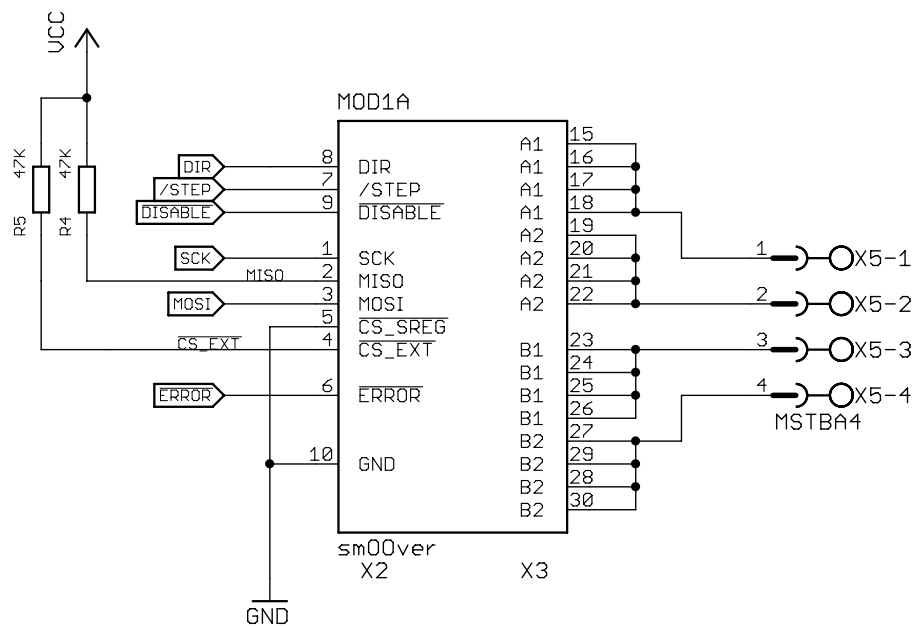
CS_SREG muss beim Hochlauf des Moduls auf high-Pegel liegen, damit die Betriebsart erkannt werden kann. Die Unterscheidung zwischen Stand-alone Betrieb und Betrieb mit externen Schieberegistern wird erst beim Versuch, die externen Register zu lesen, getroffen. Daher haben beide Betriebsarten die gleiche Moduskennung.



Prinzipschaltbild Schieberegister-Modus (MISO und MOSI sind über die Register verbunden)

- Programmierung über smOOver.prg (Modus 0b01=1): Die Betriebsart wird daran erkannt, dass die Signale CS_EXT, CS_SREG, STEP und ENABLE auf Masse gezogen sind. Der interne Controller verwendet dann die Leitungen MISO und MOSI als serielle Schnittstelle (RS-232 mit TTL-Pegeln).
- Externer SPI-Master (Modus 0b10=2): Die Konfiguration des Moduls wird von einem externen Controller übernommen. Er kann lesend und schreibend auf die Register des internen Mikrocontrollers zugreifen, der seinerseits automatisch Änderungen in die Register des Motortreibers überträgt.

CS_SREG muss zur Erkennung fest auf Masse geschaltet werden, /Step muss beim Hochlauf auf high liegen. CS_EXT wird als Chipselect-Signal verwendet und vom Master bei Bedarf auf Masse gezogen.



Prinzschaltbild SPI-Master (CS_SREG auf Masse gezogen)

Belegung Schieberegister

In der Betriebsart „Schieberegister“ stehen die folgenden Ein- und Ausgänge zur Konfiguration des Moduls zur Verfügung.

Nomenklatur: "1" bedeutet Schalter betätigt und entspricht physikalisch 0V, "0" bedeutet Schalter nicht betätigt und entspricht 5V. Zur Bedeutung der Registerwerte bitte das Datenblatt zum Motortreiber TI DRV8711 beachten.

Eingangsregister 1:

Pin-Nr.	Signal	Funktion	Mapping zu Register DRV8711
Bit0	Curr_Red	Stromabsenkung	Automatische Stromabsenkung
Bit1	Cur1	Stromeinstellung	
Bit2	Cur2	Stromeinstellung	
Bit3	Cur3	Stromeinstellung, MSB	
Bit4	MS0	Mikroschritteinstellung, LSB	
Bit5	MS1	Mikroschritteinstellung	
Bit6	MS2	Mikroschritteinstellung	
Bit7	Red_SD	Slow Decay bei Stromabsenkung	0= aus 1= ein

Eingangsregister 2:

Pin-Nr.	Signal	Funktion	Mapping zu Register DRV8711
Bit0	MS3	Mikroschritteinstellung, MSB	
Bit1	Blank.ABT	Register 0x03.8	1=Enable adaptive blanking time
Bit2	Decay.Bit0	Register 0x4.10-8	00=0x001 (slow/mixed decay)
Bit3	Decay.Bit1	Register 0x4.10-8	01=0x011 (all mixed decay) 10=0x100 (slow/auto mixed decay) 11=0x101 (all auto mixed decay)
Bit4	Stall.VDIV	Register 0x05.12-11	0=0x00 (Back EMF /32) 1=0x10 (Back EMF /8)
Bit5	Stall.THR0	Register 0x05.7-0	Stall detect threshold
Bit6	Stall.THR1	Register 0x05.7-0	00=0x20, 01=0x40, 10=0x60, 11=0x80
Bit7	Cur0	Stromeinstellung, LSB	

Ausgangsregister:

Pin-Nr.	Signal	Funktion
Bit0	Enabled	Endstufe eingeschaltet
Bit1	Cur_red	Stromabsenkung aktiv
Bit2	Temp_warn	Warnung Übertemperatur
Bit3	Temp_error	Fehler Übertemperatur, Endstufe wird abgeschaltet
Bit4	Short	Fehler Kurzschluss, Endstufe wird abgeschaltet
Bit5	Volt_error	Fehler Überspannung, Endstufe wird abgeschaltet
Bit6	Stall	Fehler Stall-Detection
Bit7	Aux	Blinkt im Normalbetrieb langsam

Stromeinstellung

Curr3..0	Dezimal	Strom eff. [A]	Strom peak [A]
0000 ^{*)}	0	0,25	0,35
0001	1	0,5	0,71
0010 ^{*)}	2	0,75	1,06
0011	3	1,0	1,41
0100 ^{*)}	4	1,25	1,77
0101	5	1,5	2,12
0110 ^{*)}	6	1,75	2,47
0111	7	2,0	2,83
1000 ^{*)}	8	2,25	3,18
1001	9	2,5	3,54
1010 ^{*)}	10	2,75	3,89
1011	11	3,0	4,24
1100 ^{*)}	12	3,25	4,60
1101	13	3,5	4,95
1110 ^{*)}	14	3,75	5,30
1111	15	4,0	5,66

^{*)}nur mit 2 Eingangsregistern einstellbar. Bei nur einem Register wird CUR0 mit „1“ belegt.

Mikroschritteinstellung

MS3..0	Dezimal	Schrittweite	Mikroschritte/U
0000	0	Vollschritt	200
0001	1	Halbschritt	400
0010	2	1/4	800
0011	3	1/8	1.600
0100	4	1/16	3.200
0101	5	1/32	6.400
0110	6	1/64	12.800
0111	7	1/128	25.600
1000 ^{*)}	8	1/256	51.200

Weitere Werte: undefiniert.

^{*)}nur mit 2 Eingangsregistern einstellbar. Bei nur einem Register wird MS3 mit „0“ belegt.

LEDs

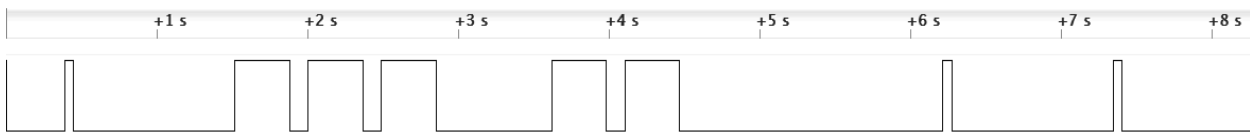
LED1	Status	Endstufe ist aktiviert, der Motor wird bestromt
LED2	Error	Signalisierung von Fehlern

Statusanzeige beim Hochlauf:

Beim Starten des Moduls (nach Power-on oder Reset) werden über LED1 mittels Blinkcode folgende Statusinformationen signalisiert:

- Firmware-Version, Major
- Firmware-Version, Minor
- Modus (erkannte Betriebsart) (0-4)
- Anzahl erkannte Eingangsregister (0-2)
- Anzahl erkannte Ausgangsregister (0-1)

Die Zahlen werden binär codiert ausgegeben, 0 = kurz hell, 1 = lang hell. Zwischen den Positionen wird eine Pause gemacht.



Beispielbild: „0 111 11 ... 0 0“ bedeutet also: Verison 0.7, Modus 3: standalone, Shiftregister: 0 Input Register, 0 Output Register.

Konfigurationsregister

smOOver.drv verwendet die in der Tabelle dargestellten Register zur Konfiguration. Das 7. Bit der Adresse dient der Umschaltung zwischen Lesen (Bit7=0) und schreibenden (Bit7=1) Register-Zugriffen. Bei Zugriffen auf die Register des DRV8711 wird das R/W-Bit an den DRV durchgereicht.

smOOver.drv Konfigurationsregister

Die unteren 12 Bits der Register 0x00 bis 0x07 entsprechen den Registern des Treiberchips DRV8711 und sind im Datenblatt des Chips von Texas Instruments beschrieben. Bit15 dient bei den hervorgehobenen Registern bzw. Registerbereichen als Schreibschutz, um ungewollte Änderungen zu verhindern. Nur wenn Bit15 auf „1“ gesetzt ist, werden in den markierten Bereichen Änderungen übernommen. Das Beschreiben dieser Bereiche wird gleichzeitig dauerhaft im Register „smart_state“ gespeichert.

HINWEIS: Änderungen an den farbig markierten Registerbereichen sind nicht empfohlen und können zum Verlust der Gewährleistung führen.

Die Register 0x08 bis 0x0F steuern das Verhalten des Moduls bzw. speichern relevante Betriebsdaten. Das 7. Bit der Adresse dient der Umschaltung zwischen Lesen (Bit7=0) und schreibenden (Bit7=1) Register-Zugriffen.

Red_cur: Legt den Motorstrom im Stillstand (Stromreduzierung) fest. Eingestellt wird der Strom als %-Wert vom aktuell gültigen Motornennstrom. Bei der Einstellung 100% findet keine Stromabsenkung statt. ¹⁾

Red_time: Legt die Zeit vom letzten Taktpuls bis zur Aktivierung der Stromabsenkung fest. ¹⁾

Soft_Start: Zeit zum Erreichen des Motornennstroms nach dem Einschalten der Endstufe über das Enable-Signal. Softstart vermeidet starke Sprünge des Motors beim Einschalten und eine pulsformige Belastung des speisenden Netzteils. ¹⁾

Temp: Speichert die Temperatur des Moduls. Im oberen Byte wird der Maximalwert abgelegt, im unteren Byte steht der aktuelle Wert zur Verfügung ²⁾. Durch Beschreiben des oberen Bytes wird der Maximalwert zurückgesetzt.

Voltage: Speichert die Versorgungsspannung des Moduls. Im oberen Byte wird der Maximalwert abgelegt, im unteren Byte steht der aktuelle Wert zur Verfügung ²⁾.

BEMF: Speichert die vom DRV8711 ausgegebene BEMF-Spannung. Im oberen Byte wird der Maximalwert abgelegt, im unteren Byte steht der aktuelle Wert zur Verfügung²⁾.

Smart_state: Registern zum Speichern relevanter Betriebsdaten. Im oberen Byte wird die maximale Temperatur des Moduls gespeichert. Im unteren Byte zählt ein Fehler-Zähler auftretende Fehler vom DRV8711 (Kurzschluss, Überspannung etc.). Die untersten drei Bits speichern Änderungen an den Registerbereichen, die Einfluss auf die Verlustleistung des Moduls haben.

Version: Zeigt die Firmware-Version auf dem Modul.

¹⁾ Nicht in der Betriebsart „Externer SPI-Master“. In dieser Betriebsart hat der externe SPI-Master die Kontrolle über das Aktivieren der Endstufe und die Drehmoment-Einstellung (Torque-Register).

²⁾ Der aktuelle Wert im unteren Register steht nur im Modus „Externer SPI-Master“ zur Verfügung. Beim Programmieren über smOOver.prg werden die aktuellen Werte nicht ermittelt, im unteren Byte wird Null zurückgegeben.

SPI-Interface

Das SPI-Interface verwendet den Modus 3 (CPHA=1, CPOL=1 *) bei 125 kHz Taktrate. Zwischen den einzelnen Bytes ist eine Pause von 1ms empfohlen, damit der interne Controller seine Antwort auf die Anfrage vorbereiten kann.

Bytefolge zum Schreiben:

<RW-Bit=1, 7 bit Adresse>, <Datenbyte high>, <Datenbyte low>

Antwort vom smOOver: 0xFF, 0xFF, 0xFF

Bytefolge zum Lesen:

<RW-Bit=0, 7 bit Adresse>, <beliebiges Datenbyte>, <beliebiges Datenbyte>

Antwort vom smOOver: 0xFF, <Datenbyte high>, <Datenbyte low>

*) https://de.wikipedia.org/wiki/Serial_Peripheral_Interface

RS-232 Schnittstelle zur Programmierung

In Verbindung mit dem Programmieradapter smOOver.prg können die Default-Einstellungen von smOOver.driv via USB angepasst werden. Das Interface stellt einen virtuellen COM-Port bereit.

Kommunikationseinstellungen:

9.600 baud/s, 8 bits, 1 Stopbit, ohne Paritätsbit, Zeilenende = „LF“ (Linefeed)

Zahlen können dezimal oder hexadezimal übertragen werden, die Hexdarstellung wird durch ein vorangestelltes „x“ gekennzeichnet.

Register lesen: <Registernummer>?<LF>, z.B. xA? - Antwort: x09=x000a

Register schreiben: <Registernummer>=<Wert><LF>, z.B. xA=x11FF oder 10=23

Firmware-History:

- 1.0 Erste Release-Version.
- 1.1 Default-TORQUE auf 4A geändert.
- 1.2 ENBL-Bit wird erst nach Initialisierung aller Register gesetzt.
- 1.3 Fehler-Status wird mind. 1 Sekunde gehalten, auch wenn Enable vorher zurückgesetzt wird.
- 1.4 Neues Konfig-Bit für Slow Decay während der Stromabsenkung verbessert das Geräuschverhalten. Belegung Schieberegister 1 angepasst. Schreibschutz Byte 0x00 wirkt nur noch auf High-Byte.
- 1.5 Zeitkritische Routinen überarbeitet, um Probleme bei hohen Schrittfrequenzen zu vermeiden.
- 1.6 Fehler beim permanenten Speichern der Register behoben.

Notizen:

Abmessungen:

