

Schrittmotor Steuer- und Endstufenkarte SMSIPC für 6 Mikroschritt-Motore Version 7 (Firmware58)

Inhalt

<u>Erste Inbetriebnahme</u>	<u>2</u>
<u>Motorstromeinstellung</u>	<u>2</u>
<u>Der Adressenschalter</u>	<u>2</u>
<u>Motor- und Schalteranschlüsse</u>	<u>2</u>
<u>Die Bedienung über die Schnittstelle</u>	<u>2</u>
<u>Funktionen</u>	<u>3</u>
<u>Statusauswertung</u>	<u>3</u>
<u>Befehlsbeschreibungen</u>	<u>4</u>
<u>LED-Anzeigen</u>	<u>4</u>
<u>Anschlussbelegungen</u>	<u>5</u>
<u>Anhang:</u>	
<u>Ansteuerbeispiele RS485</u>	<u>6</u>
<u>FAQs (Häufig gestellte Fragen)</u>	<u>8</u>
<u>Platinenabbildungen</u>	<u>9</u>
<u>Vergleich der 1*2A- und 6*1A- Motorkarten</u>	<u>10</u>

Technische Daten:

Größe:	Europakarte 100x160mm, 30mm hoch
Spannungsversorgung:	5V ca. 180mA
Motorspannung:	12V -24V
Ausgangsleistung:	6 Stück 2-Strang Schrittmotor max. 2x 1A
Motorstrom:	In 16 Stufen von 0,1A - 1A; automatische Stromabsenkung
Schnittstelle:	RS 485 full duplex; 19,2 kBaud; Optoisoliert
Befehle:	26, abgesichert mit 8-Bit CRC
Adressbereich:	220 verschiedene mit HEX-Drehschaltern einstellbar
Schrittfrequenz:	31 Hz - 8 kHz
Zählbereich:	999.999 Schritte, Vollschritt bis 16-fach Mikroschritt
Firmwareupdate	Mit kleiner Zusatzplatine im System möglich

In den speziellen Schrittmotorcontrollern auf der Karte sind 6-stellige Zähler realisiert, die die Motorschritte mitzählen. Wird z.B. ein Fahrbefehl von 1000 Schritten gesendet, beschleunigt der Motor mit der eingestellten Beschleunigungsrampe, fährt mit der eingestellten Geschwindigkeit und bremst mit der Rampe ab. Über die Schnittstelle kann mit 26 verschiedenen Befehlen kommuniziert werden. Ein Auszug:

Motor starten/ stoppen
vorwärts/ rückwärts
Schrittweite
Schrittfrequenz lesen/ schreiben
Beschleunigungszeit lesen/ schreiben
Istzähler lesen/ schreiben
Softwareanschläge Minimum und Maximum lesen/ schreiben
Mehrfachschritt und Zielposition schreiben
Status lesen

Für jeden Motor existieren zwei Endschaltereingänge. Außerdem können externe Ein- und Ausgänge angeschlossen werden. Die wichtigsten Parameter werden in einem EEPROM gespeichert..
Folgende Funktionen werden überwacht: Unterspannung; Übertemperatur; Phasenunterbrechung.

Erste Inbetriebnahme:

- 1) Anschluß der Vcc 5V: + an 20a/c; GND an 18a/c, der Motorspannung (12-24V): + an 14a/c GND an 16a/c, des Motors (2 Wicklungen) und der RS485- bzw. RS232 Schnittstelle. Strombedarf 5V ca. 180mA
Die Endschalter müssen auf GND liegen.
- 2) Motorstrom einstellen (vorsichtshalber zuerst auf 1= 100 mA)
- 3) Platinenadresse einstellen (in Hex: 40= 64dez) und Spannungen einschalten.
Nun muss die LED des angeschlossenen Motors dunkel sein.
- 4) Das Testprogramm SMSIPC.exe starten (Bedient Com2).
- 5) Das Programm scannt die Adressen im Bereich 64-80. In dem Programm die Adresse einstellen (64)
- 6) Durch Drücken der Taste „a“ Motor starten.

Ein Kaltstart (Anlegen der Betriebsspannung und initialisieren der Motorcontroller) dauert ca. 5 Sekunden, ein Reset (Laden der Werte aus dem EEPROM) per Befehl 31 oder Taster dauert ca. 0,5 Sekunden.

Ein Schnittstellenwandler von RS232 auf RS485 wurde auch entwickelt (Schaltbild kann angefordert werden).

Hex-Drehschalter für die Motorstromeinstellung

Schalter-Stellung	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Phasenstrom (mA)	0	106	204	293	376	453	525	592	655	713	769	820	869	915	958	1000

Der Motorstrom wird automatisch auf die Hälfte abgesenkt, sobald der Motor steht.

Motor- und Endschalteranschlüsse

Für jeden Motor existiert eine 7-polige Stiftleiste auf der Platine: Vier Pins für die Motorwicklungen und drei Pins für zwei Endschalter und Masse (Seite 6). Die Endschalter müssen auf GND liegen, damit der Motor dreht.

Der Adressenschalter

Die Adresse der Karte wird mit den beiden Hex-Drehschaltern eingestellt: Ist zum Beispiel am High-Schalter eine 4 und am Low-Schalter eine 1 gewählt, bedeutet das 41 in Hex = 65 in Dez.

Der Bereich der Adresse darf nur von 20hex (32dez) bis FFhex (255dez) eingestellt werden, da die Steuerzeichen nicht benutzt werden dürfen.

Die Karte kann mit Adresse 00 in einen Testmodus geschaltet werden. Siehe FAQs am Ende.

Die Bedienung über die Schnittstelle

Parameter: 19200,8,n,1, Kein Handshake, Full Duplex

Ablauf zum Schreiben zur Motorsteuerung:

STX, Adresse, Funktion, Parameter, CRC, EOT

Byte 1 1 1 0-6 3 1

Antwort von der Steuerung:

Es wird eine von fünf möglichen Rückmeldungen gesendet

ACK (06dez) - Befehl erfolgreich ausgeführt

NAK (21dez) - CRC falsch

DC1 (17dez) - ungültiger Befehl, z.B. Befehlsnummer > 139

DC2 (18dez) - ungültiger Parameter, z.B. Schrittfrequenz > 255

DC3 (19dez) - Befehl nicht erfolgreich, z.B. Startbefehl bei Hardwarefehler

Ablauf zum Lesen von der Motorsteuerung:

STX, Adresse, Funktion, CRC, EOT

Byte 1 1 1 3 1

Antwort von der Steuerung:

STX, Adresse, Funktion, Parameter, CRC, EOT

Hinweis: Die Berechnung der CRC wird im Anhang beschrieben.

Schreibfunktionen:

Wert dezimal	Parameter	Wertebereich	Funktion	Mögliche Antwort	
				DC2 ungült. Param.	DC3 nicht erfolgr.
31	keine		Reset		
32	keine		Motor starten		x
33	keine		Motor stoppen		
34	keine		Vorwärts		
35	keine		Rückwärts		
40	keine		Aktuelle Werte ins EEPROM speichern		
41	3 Stellen	001-255	Schrittfrequenz schreiben	x	
42	3 Stellen	001-030	Beschleunigungszeit schreiben	x	
43	6 Stellen	000000-999999	Istzähler schreiben		
44	6 Stellen	000000-999999	Maximum schreiben	x	
45	6 Stellen	000000-999999	Minimum schreiben	x	
46	6 Stellen	000000-999999	Mehrfachschrift schreiben und starten		
51	6 Stellen	000000-999999	Userwert ins EEPROM		x
53	keine		Endstufe ausschalten		
54	keine		Endstufe einschalten		
56	3 Stellen	000000-999999	Zielposition schreiben	x	
57	3 Stellen	000-004	Mikroschrittzahl schreiben	x	

Lesefunktionen:

Wert dezimal	Parameter	Wertebereich	Funktion	Mögliche Antwort
128	keine	001-255	Schrittfrequenz lesen	Bei Auslieferung: 100
129	keine	001-030	Beschleunigungszeit lesen	001
130	keine	000000-999999	Istzähler lesen	010000
131	keine	000000-999999	Maximum lesen	999999
132	keine	000000-999999	Minimum lesen	000000
134	keine	000000-032767	Status lesen	001536
137	keine	000-255	Versionsnummer lesen (hier 052)	052
138	keine	000000-999999	Userwert lesen	123456
140	keine	000000-999999	Zielposition lesen	000000
141	keine	000-004	Mikroschrittzahl lesen	004
142	keine	000-007	Last-Indikator lesen	005

Es werden also zwischen 7 und 13 ASCII-Zeichen für eine Übertragung benötigt.

Die Motorsteuerungen senden nicht selbsttätig, sondern erst nach Anfrage vom Host, um Kollisionen zu vermeiden.

Die Befehlsnummern sind nicht komplett belegt, da die Kompatibilität zu der anderen Schrittmotorkarte gewährleistet sein sollte.

Statusbyte

Als Status wird eine Zahl von 000000 bis 065535 zurückgesendet. Nach dem Umwandeln in hex erhält man zwei Statusbytes. Diese setzen sich wie folgt zusammen:

Byte	Bit	Gelöscht	gesetzt
1	0	Motor steht	Motor dreht
1	1	Vorwärts	Rückwärts
1	2	nicht benutzt (immer 0)	nicht benutzt (immer 0)
1	3	nicht benutzt (immer 0)	nicht benutzt (immer 0)
1	4	nicht benutzt (immer 0)	nicht benutzt (immer 0)
1	5	Endschalter 1 nicht aktiv	Endschalter 1 aktiv
1	6	Endschalter 2 nicht aktiv	Endschalter 2 aktiv
1	7	nicht benutzt (immer 0)	nicht benutzt (immer 0)
2	0	Softwarelimit Low nicht erreicht	Softwarelimit Low erreicht
2	1	Softwarelimit High nicht erreicht	Softwarelimit High erreicht
2	2	Normaltemperatur	Übertemperatur (über 90°)
2	3	Motorspannung 12-28V	Motorspannung unter 12V
2	4	Motor angeschlossen	Phasenunterbrechung
2	5	Keine Hardware-Fehler	Hardware-Fehler war aufgetreten
2	6	nicht benutzt (immer 0)	nicht benutzt (immer 0)
2	7	Kein Überstrom	Überstrom

Die beiden Softwarelimiten werden durch Software eingestellt und wenn der Istzähler diese Werte erreicht, stoppt der Motor und die Bits im Statusbyte werden gesetzt. Die Bits werden durch einen Stoppbefehl gelöscht.

Befehle:

Zu beachten ist: Wenn während einer Motorfahrt der Endschalter geöffnet (angefahren) wird, stoppt der Motor. Der Fahrbefehl wird aber nicht gelöscht sondern nach dem Schließen dreht der Motor weiter. Das bedeutet: wenn ein Fahrbefehl bei geöffnetem Endschalter gesendet wird, wird ein OK zurückgesendet. Ob der Motor dreht, kann aber im Status erkannt werden.

Schrittfrequenz und Beschleunigung

Der Einstellbereich der Schrittfrequenz ist linear und von 001=31 Hz bis 255= ca.8 kHz.

Der Bereich der Rampe geht von 1=langsame Beschleunigung bis 30=schnelle Beschleunigung.

Mikroschritte

Die Zahlen 0-4 bedeuten folgende Schrittzahlen:

000-Vollschritt	(bei den Standardmotoren	200 Schritte/Umdrehung)
001-Halbschritt	(„	400 „
002-Viertelschritt	(„	800 „
003-Achtelschritt	(„	1600 „
004-Sechzehntelschritt	(„	3200 „

EEPROM

Mit dem Befehl 40 werden im EEPROM alle Parameter betriebsspannungsunabhängig gespeichert. Nach einem Reset werden folgende Werte aus dem EEPROM ausgelesen:

- Schrittfrequenz
- Beschleunigung
- Istzähler
- Softwarelimit oben
- Softwarelimit unten
- Mehrfachschritt
- Drehrichtung
- Mikroschritte

Userwert

Der Befehl 51 schreibt eine 6-stellige Zahl in das EEPROM und der Befehl 138 liest diese Zahl und sendet sie zum Rechner.

Anschlag-Erkennung

Wie in der folgenden Anleitung zum Endstufen-IC TMC246 beschrieben, kann eine Anschlagerkennung ohne Endschalter durchgeführt werden. Dazu muss eine Mikroschrittauflösung und eine Geschwindigkeit gewählt werden, so dass der Last-Indikator (Befehl 142) einen Wert zwischen 5 und 7 ergibt. Dann verringert sich der Wert bei einer größeren Belastung des Motors. Die Steuer-Software kann das zum Erkennen verwenden und den Motor stoppen; das geschieht nicht durch die Motorsteuerkarte selbst.

Der Auszug aus der Endstufen-Beschreibung:

Stall Detection

Using the sensorless load measurement

The TMC246 provides a patented sensorless load measurement, which allows a digital read out of the mechanical load on the motor via the serial interface. To get a readout value, just drive the motor using sine commutation and mixed decay switched off. The load measurement then is available as a three bit load indicator during normal motion of the motor. A higher mechanical load on the motor results in a lower readout value. The value is updated once per fullstep.

Since the load detection is based on the motor's back EMF, the readout results depend on several factors:

- Motor velocity: A higher velocity leads to a higher readout value
- Motor resonance: Motor resonances cause a high dynamic load on the motor, and thus measurement may give unsatisfactory results.
- Motor acceleration: Acceleration phases also produce dynamic load on the motor.
- Mixed decay setting: For load measurement mixed decay has to be off for some time before the zero crossing of the coil current. If mixed decay is used, and the mixed decay period is extended towards the zero crossing, the load indicator value decreases.

Implementing sensorless stall detection

The sensorless stall detection typically is used, to detect the reference point without the usage of a switch or photo interrupter. Therefore the actuator is driven to a mechanical stop, e.g. one end point in a spindle type actuator. As soon as the stop is hit, the motor stalls. Without stall detection, this would give an audible humming noise and vibrations, which could damage mechanics.

To get a reliable stall detection, follow these steps:

1. Choose a motor velocity for reference movement. Use a medium velocity which is far enough away from mechanical resonance frequencies. In some applications even motor start / stop frequency or lower may be used. With this the motor can stop within one fullstep if a stall is detected.
2. Use a sine stepping pattern and switch off mixed decay. Monitor the load indicator during movement. It should show a stable readout value in the range 3 to 7 (L_{MOVE}). If the readout is high (>5), the mixed decay portion may be increased, if desired.
3. Choose a threshold value L_{STALL} between 0 and $L_{MOVE} - 1$.
4. Monitor the load indicator during each reference search movement, as soon as the desired velocity is reached. Readout is required at least once per fullstep. If the readout value at one fullstep is below or equal to L_{STALL} , stop the motor.
5. If the motor stops during normal movement without hitting the mechanical stop, decrease L_{STALL} . If the stall condition is not detected at once, when the motor stalls, increase L_{STALL} .

LED-Anzeigen:

Auf der Frontseite ist für jeden Motor eine rote LED angebracht;

Wenn alles in Ordnung ist, ist die LED dunkel. Sie leuchtet, wenn einer der folgenden Fehler vorliegt:

Übertemperatur; Spannung unter 12V; Überstrom; Phasenbruch

Den genauen Grund des Fehlers kann man durch Abfrage der Statusbytes im Stillstand ermitteln.

Während der Motor dreht, blinkt die entsprechende LED.

Außerdem gibt es zwei grüne LEDs für 5V und 24V.

Anschlussbelegungen:

32-polige VG-Leiste

14a	24V Motorversorgung	Betriebsspannungen
14c	(von 12-28V)	
16a	GND (Masse)	
16c		
18a		
18c		
20a	5V Schaltungsversorgung	
20c		
22a	GND iso	RS485- Schnittstelle
22c	(GND für Schnittstelle)	
24a	/RX	
24c	RX	
26a	TX	
26c	/TX	
28a	Ext 1	Anschlüsse für Endschalter oder Relais- Zusatzelektronik
28c	Ext 2	
30a	Ext 3	
30c	Ext 4	
32a	Ext 5	
32c	Ext 6	

6 Stück 7-polige Stiftleisten auf der Platine

1	Phase A	Motor
2	Phase B	
3	Phase B'	
4	Phase A'	
5	GND	Endschalter
6	Endschalter rückwärts	
7	Endschalter vorwärts	

Version mit 6 Stück 9- und 15-polige Sub-D Buchsen auf der Frontplatte

1	Phase A	Motor
2	Phase A'	
3	Phase B	
4	Phase B'	
5	GND	Endschalter
6	Endschalter rückwärts	
7	GND	
8	Endschalter vorwärts	
9-15	nicht belegt	

Belegung am Gamma in Gö und am FRM2

Version mit einer 50-poligen Sub-D Buchse auf der Frontplatte

1	Wicklung 1/A	Motor 1
2	Wicklung 1/B	
3	Wicklung 2/A	
4	Wicklung 2/B	
5	Endschalter 1	
6	Gemeinsame Masse	
7	Endschalter 2	Motor 2
8	Wicklung 1/A	
9	Wicklung 1/B	
10	Wicklung 2/A	
11	Wicklung 2/B	
12	Endschalter 1	
13	Gemeinsame Masse	Versorgung
14	Endschalter 2	
15	+24 V	
16	+24 V	Motor 3
17	+24 V	
18	Wicklung 1/A	
19	Wicklung 1/B	
20	Wicklung 2/A	
21	Wicklung 2/B	
22	Endschalter 1	Motor 4
23	Gemeinsame Masse	
24	Endschalter 2	
25	Wicklung 1/A	
26	Wicklung 1/B	
27	Wicklung 2/A	
28	Wicklung 2/B	Versorgung
29	Endschalter 1	
30	Gemeinsame Masse	
31	Endschalter 2	Motor 5
32	+5V	
33	+5V	
34	Wicklung 1/A	
35	Wicklung 1/B	
36	Wicklung 2/A	
37	Wicklung 2/B	Motor 6
38	Endschalter 1	
39	Gemeinsame Masse	
40	Endschalter 2	
41	Wicklung 1/A	
42	Wicklung 1/B	
43	Wicklung 2/A	Versorgung
44	Wicklung 2/B	
45	Endschalter 1	
46	Gemeinsame Masse	Motor 6
47	Endschalter 2	
48	GND	
49	GND	
50	GND	Versorgung

Version mit einer 44-poligen Sub-D Buchse auf der Frontplatte

1	Wicklung 1/A	Motor 1
2	Wicklung 1/B	
3	Wicklung 2/A	
4	Wicklung 2/B	
5	Endschalter 1	
6	Gemeinsame Masse	
7	Endschalter 2	Motor 2
8	Wicklung 1/A	
9	Wicklung 1/B	
10	Wicklung 2/A	
11	Wicklung 2/B	
12	Endschalter 1	
13	Gemeinsame Masse	Motor 3
14	Endschalter 2	
15	frei	
16	Wicklung 1/A	
17	Wicklung 1/B	
18	Wicklung 2/A	
19	Wicklung 2/B	
20	Endschalter 1	
21	Gemeinsame Masse	Motor 4
22	Endschalter 2	
23	Wicklung 1/A	
24	Wicklung 1/B	
25	Wicklung 2/A	
26	Wicklung 2/B	
27	Endschalter 1	
28	Gemeinsame Masse	Motor 5
29	Endschalter 2	
30	frei	
31	Wicklung 1/A	
32	Wicklung 1/B	
33	Wicklung 2/A	
34	Wicklung 2/B	
35	Endschalter 1	
36	Gemeinsame Masse	Motor 6
37	Endschalter 2	
38	Wicklung 1/A	
39	Wicklung 1/B	
40	Wicklung 2/A	
41	Wicklung 2/B	
42	Endschalter 1	
43	Gemeinsame Masse	
44	Endschalter 2	

Version mit 6 Stück 9-polige Sub-D Buchsen auf der Frontplatte

1	Phase A	Motor
2	Phase A'	
3	Phase B	
4	Phase B'	
5	nicht belegt	Endschalter
6	GND	
7	Endschalter rückwärts	
8	GND	
9	Endschalter vorwärts	

Überprüfen ob diese Belegung am Gamma verwendet wurde.

Belegung am FRM im i/o Rahmen IPC-127.06

1	Phase A	Motor
2	Phase A'	
3	Nicht belegt	Endschalter
4	Endschalter rückwärts	
5	Endschalter vorwärts	
6	Phase B	Motor
7	Phase B'	
8	Nicht belegt	
9	GND	

ANHANG:

CRC (Cyclic Redundancy Check)

Die Checksumme wird in Anlehnung an das MODBUS-Protokoll ausgerechnet; allerdings nur mit 8 Bit. Bei Bedarf kann die EUROTHERM-Modbusbeschreibung angefordert werden.

Ein QuickBasic Unterprogramm als Beispiel:

```
SUB Check8 (wert(), laenge, crc) ' Berechnung der CRC
  crc = 255 ' FFh > CRC Register
  FOR i = 2 TO (laenge) ' STX und EOT nicht auswerten
    byte = wert(i)
    crc = crc XOR byte ' CRC Register XOR next byte > CRC
    n = 0 ' 0>n
n4:    temp = crc
    crc = INT(crc / 2) ' shift CRC right 1 bit
    Carry = temp MOD 2
    IF Carry = 0 THEN GOTO n5 ' Over flow?
    crc = crc XOR 161 ' YES: CRC XOR A1h > CRC
n5:    n = n + 1 ' n+1 > n
    IF n < 8 THEN GOTO n4 ' n > 7 ?
  NEXT i ' Is message complete ?
END SUB
```

Es werden hier aus der Adresse, dem Funktionscode und den Parametern die 8-Bit CRC gebildet und später als 3-Byte ASCII der Nachricht angehängt.

Beispiele zur Ansteuerung mit einem Terminalprogramm

Motorstart

gesendet							
Hex	02	41	20	32	31	33	04
Dez	02	65	32	50	49	51	04
ASCII	STX	A	_	2	1	3	EOT

empfangen
06
06
ACK

STX = Start Of Text
A = Adresse
_ = Leertaste
213 = Ausgerechnete CRC
EOT = End Of Text
ACK = Acknowledge
NAK = Not Acknowledge

Motorstop gesendet STXA!068EOT
Antwort ACK
1000 Schritte gesendet STXA.001000100EOT
Antwort ACK

Es existiert ein QuickBasic und DOS Test- und Bedienprogramm.

Häufig gestellte Fragen

F: Welche Schrittmotore kann man anschließen?

A: Alle 4-Phasen-Schrittmotore, also mit 2 Wicklungen mit maximal 1A pro Wicklung.

Wir setzen verschiedene Schrittmotoren ein, hier ein paar Adressen:

www.ec-motion.de

www.schrittmotor.de

www.nanotec.de

www.slosyn.com

Ansonsten kann man jeden 2-Phasenmotor nehmen; zum Probieren nehmen wir gern einen Ausgebauten aus einer alten Festplatte oder einem Drucker.

F: Wie kann ich die Karte an einen Rechner mit RS232-Schnittstelle anschließen?

A: Es gibt Schnittstellenkonverter zu kaufen oder selbstzubauen.

Wir haben eine kleine Platine entwickelt:

Der Umsetzer besteht nur aus den ICs MAX491 und MAX232 der Firma Maxim um den Differenzpegel zuerst auf TTL und dann auf V24-Pegel zu ändern. Das Schaltbild kann angefordert werden.

So kann man mehrere SMS-Karten mit einem Rechner betreiben.

Falls nur eine Karte angeschlossen werden soll, kann die RS485-Schnittstelle mit einem kleinen Zusatzmodul auf RS232 umgebaut werden (es wird der MAX491 herausgezogen und eine Adapterplatine mit einem MAX232 in den Sockel gesteckt).

F: Was muss angeschlossen werden, damit sich Rechner und Karte verständigen?

A: Es müssen nur die vier Busleitungen und die 5V angeschlossen werden, damit eine Antwort kommt.

F: Die Karte meldet sich nicht, was tun?

A: Nach dem Anlegen der 5V muss die grüne 5V Betriebsspannungs-LED leuchten.

Dann irgendein Terminal-Programm starten (19200,8,n,1) und beide Adressschalter auf 0 stellen (Testmodus).

Jedes Mal wenn man nun den Reset-Taster an der Platine drückt, wird ein String zum Rechner gesendet:

SMS Hardware V7 Software V52

Testmodus IPC 12/2005

Eigene Adresse: 32dez

Wenn das nicht funktioniert, muss man messen: Hat der MAX491 zwischen 7 und 14 eine Spannung von ca. 5V?

Mit einem Terminalprogramm (19200,8,n,1) etwas senden und mit einem Oszi testen, ob an Pin 10 (RXD) vom Prozessor ein Signal ankommt.

Wenn nicht, die drei Optokoppler HCPL7720 raus, dann am Max491 Pin 4 mit 14 brücken und Pin 2 mit 5 brücken.

Jetzt muss das Signal vom Terminal zur Platine wieder zurück an das Terminal geechoet werden.

Ist das OK, stimmt die Verdrahtung.

Wenn nicht, könnten die beiden Leitungen der RX oder TX verdreht sein.

Die Optokoppler wieder einbauen und Prozessor ausbauen. Brücke im Prozessorsockel von 10 nach 11 und Brücke von 40 nach 2.

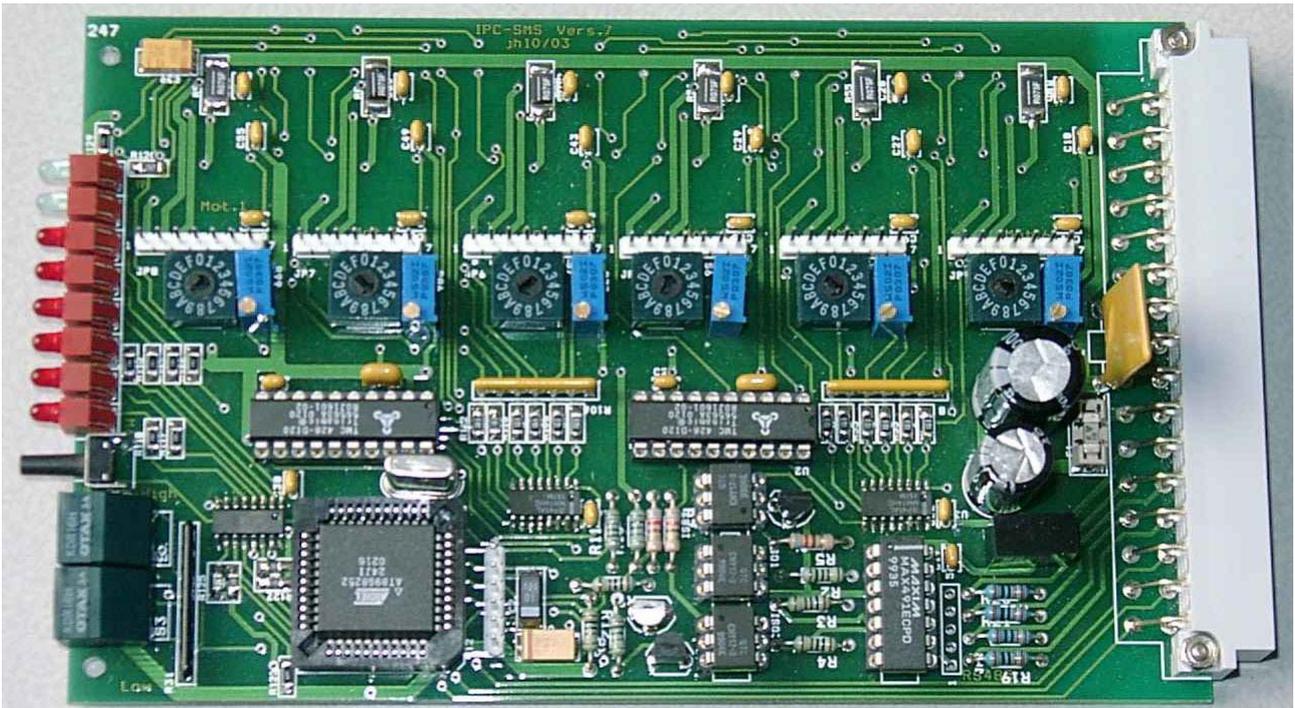
Jetzt muss das Signal vom Terminal zur Platine wieder zurück an das Terminal geechoet werden.

Wenn OK, Prozessor wieder einbauen und nachsehen, ob der Prozessor an Pin 11 (TXD) antwortet. Natürlich muss jetzt das Protokoll (vor allen Dingen die Platinenadresse) stimmen, sonst antwortet die Karte nicht.

F: Welche integrierte Schaltungen werden eingesetzt?

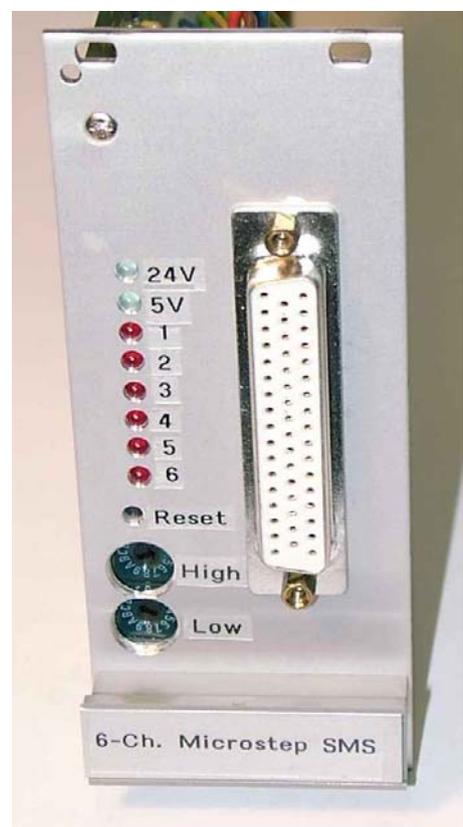
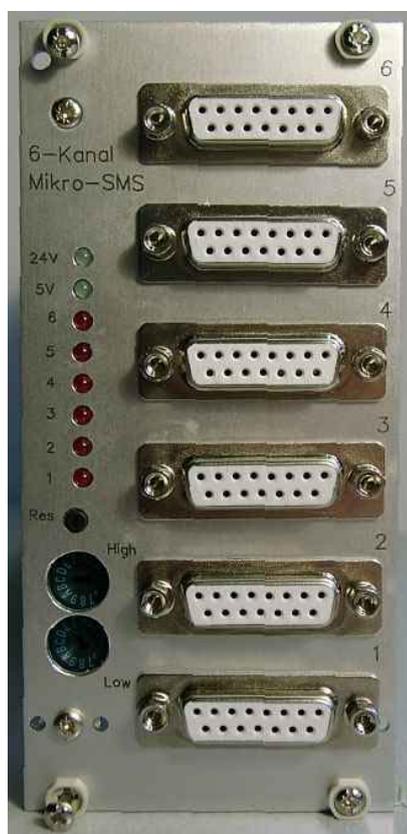
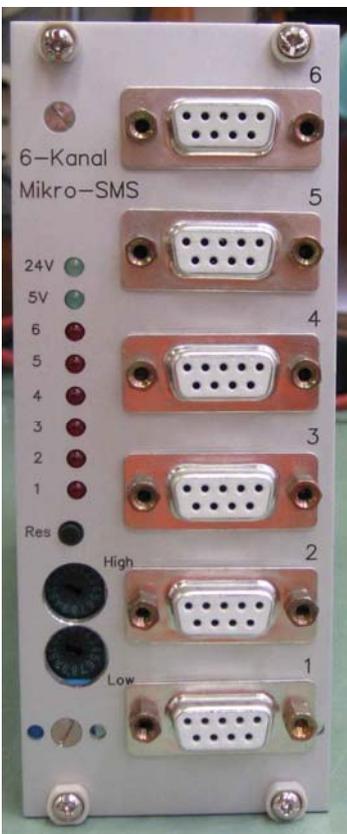
A: 2 x TMC 428 und 6 x TMC246 der Firma Trinamic: www.trinamic.com . Dort gibt es auch Datenblätter und Beschreibungen. Die Steuerung übernimmt ein Atmel 89S8252.

Abbildungen



Bestückungsseite

Am linken Rand sind die Leuchtdioden, der Reset-Taster und die Adressenschalter zu erkennen. Die sechs Drehschalter auf der Platine sind zum Einstellen der Motorströme (siehe Seite 2). Darüber die Steckleisten der Motoranschlüsse.



Verschiedene Frontplattenausführungen

Vergleich der alten Schrittmotorsteuerung für einen 2A Motor mit der neuen Schrittmotorsteuerung für sechs 1A Motore

Wert	Parameter	Wertebereich	Funktion	1x2A	6x1A
31	keine		Reset	X	X
32	keine		Motor starten	X	X
33	keine		Motor stoppen	X	X
34	keine		Vorwärts	X	X
35	keine		Rückwärts	X	X
36	keine		Vollschritt	X	
37	keine		Halbschritt	X	
38	keine		Relais ein	X	
39	keine		Relais aus	X	
40	keine		Aktuelle Werte ins EEPROM speichern	X	X
41	3 Stellen	001-255	Schrittfrequenz schreiben	X	X
42	3 Stellen	001-255	Beschleunigungszeit schreiben	001-255	001-015
43	6 Stellen	000000-999999	Istzähler schreiben	X	X
44	6 Stellen	000000-999999	Maximum schreiben	X	X
45	6 Stellen	000000-999999	Minimum schreiben	X	X
46	6 Stellen	000000-999999	Mehrfachschritt schreiben und starten	X	X
47	3 Stellen	001-255	Referenzpunkt anfahren	X	
48	keine		Phasenunterbrechung testen	X	
49	3 Stellen	000-063	Konfigurationsbyte ins EEPROM	X	
50	3 Stellen	001-004	Rampentabelle auswählen	X	
51	6 Stellen	000000-999999	Userwert ins EEPROM	X	X
52	keine		Mit Bremsrampe abbremsen und stoppen	X	
53	keine		Interne Endstufe aus	X	ab Ver. 5.5
54	keine		Interne Endstufe ein	X	ab Ver. 5.5
55	3 Stellen	000-255	Inhibit-Pause schreiben	X	
56	6 Stellen	000000-999999	Zielposition schreiben		X
57	3 Stellen	000-004	Mikroschritte schreiben		X
58	3 Stellen	000-255	Stop-Pause schreiben	ab Ver. 45	

128	keine	001-255	Schrittfrequenz lesen	X	X
129	keine	001-255	Beschleunigungszeit lesen	001-255	001-015
130	keine	000000-999999	Istzähler lesen	X	X
131	keine	000000-999999	Maximum lesen	X	X
132	keine	000000-999999	Minimum lesen	X	X
133	keine	000000-999999	Mehrfachschritt lesen	X	
134	keine	000000-032767	Status lesen	X	X
135	keine	000-063	Konfigurationsbyte lesen	X	
136	keine	001-004	Rampentabelle lesen	X	
137	keine	000-255	Versionsnummer lesen	X	X
138	keine	000000-999999	Userwert lesen	X	X
139	keine	000-255	Inhibit-Pause lesen	X	
140	keine	000000-999999	Zielposition lesen		X
141	keine	000-004	Mikroschritte lesen		X
142	keine	000-007	Last-Indikator lesen		X
143	keine	000-255	Stop-Pause lesen	ab Ver. 45	

Statusbyte

Byte	Bit	gelöscht	gesetzt	1x2A	6x1A
1	0	Motor steht	Motor dreht	X	X
1	1	Vorwärts	Rückwärts	X	X
1	2	Vollschritt	Halbschritt	X	
1	3	Relais aus	Relais ein	X	
1	4	Inhibit-Eingang nicht aktiv	Inhibit-Eingang aktiv	X	
1	5	Endschalter 1 nicht aktiv	Endschalter 1 aktiv	X	X
1	6	Endschalter 2 nicht aktiv	Endschalter 2 aktiv	X	X
1	7	Referenzschalter nicht aktiv	Referenzschalter aktiv	X	
2	0	Softwarelimit Low nicht erreicht	Softwarelimit Low erreicht	X	X
2	1	Softwarelimit High nicht erreicht	Softwarelimit High erreicht	X	X
2	2	Normaltemperatur	Übertemperatur (über 90°)	X	X
2	3	Motorspannung 20-36V	Motorspannung unter 20V	X	X
2	4	Motor angeschlossen	Phasenunterbrechung	X	X
2	5	Keine Hardware-Fehler	Hardware-Fehler war aufgetreten	X	
2	6	Externe Endstufe	Interne Endstufe	X	
2	7	Kein Überstrom	Überstrom		X

Versionsübersicht

50: Erste Version

51: Anschlag-Erkennung

52: Rampe

55: Endstufe abschaltbar

56: Fehler beseitigt

57: LED-Flackern beseitigt; kleine Änderungen wegen neuem Prozessor

58: Beschleunigungsbereich von 1-30 vergrößert

jh 5/2007

Institut Für Physikalische Chemie
der Universität Göttingen

Jörg Hampe

jhampe@gwdg.de

lr 14.07.2014

Angepasste Phasenbezeichnungen an der 9pol. SUB-D und am Platinenstecker.

Statt „Phase 1“ mit den Anschlüssen A&B (Phase 1A und Phase 1B für eine Wicklung) verwenden wir nun folgende Bezeichnung: Phase A und A' für eine Wicklung.