

USB-Dual-Schrittmotoren-Interface USI 2

Das USB-Schrittmotoren-Interface ermöglicht die gleichzeitige Ansteuerung von zwei Schrittmotoren mit bis zu 750 mA Strombedarf je Phase, stellt 4 digitale Endschalter-Eingänge und zusätzlich zwei Open-Collector-Schaltausgänge zur Verfügung. Die Ansteuerung erfolgt via USB von einem PC aus, eine dem Bausatz beiliegende Software-Applikation erlaubt den sofortigen Einstieg in eigene Steuerungsanwendungen. Damit steht eine komplette und moderne Steuerungslösung für den Betrieb von kleinen Schrittmotoren zur Verfügung.

Schritt für Schritt ...

In vielen Bereichen der Elektrotechnik/Elektronik, aber vor allem auch im Maschinenbau, kommen für zahlreiche Antriebs- und Positionierungsaufgaben heutzutage Schrittmotoren zum Einsatz. Beispielsweise in Robotern, Druckern, Plottern, Verpackungsmaschinen, Heimelektronikgeräten, Autoradios, CD-Wechslern, ja sogar in Spielzeugen. Der wesentliche Vorteil von Schrittmotoren gegenüber herkömmlichen gleichspannungsgesteuerten Motoren liegt in der zielgenaueren Positionierung und den sehr kurzen Anlauf- und Stoppzeiten.

Selbst im Hobbybereich ist eine eigene Idee zur Anwendung von Schrittmotoren schnell geboren. Auch die mechanische

Realisierung stellt den Hobbyisten selten vor Probleme, denn gerade hier verfügen z. B. Modellbauer über ausreichend Erfah-

Technische Daten:

Spannungsversorgung:	7–20 V DC
Stromaufnahme:	~100 mA (ohne Motor), 750 mA (max. pro Phase)
Motortyp:	unipolar oder bipolar
Betriebsart:	Halbschritt-Modus
Max. Anzahl der Motoren:	2
Endschalter-Eingänge:	4
Schaltausgänge:	2 (Open-Collector/max. 30 V/100 mA)
Gehäuseabmessungen (B x H x T):	64 x 28 x 115 mm
USB-Schnittstellen-Konfiguration:	19.200 Baud, 8 Datenbits, gerade Parität, 1 Stoppbit
Systemvoraussetzung:	PC mit WIN 98/ME/XP/2000 und USB-Schnittstelle

rung. Nur wenn es an die Elektronik mit der Ansteuerung der Schrittmotoren geht, werden viele (potenzielle) Anwender abgeschreckt. Dies ist wohl auch der Hauptgrund, weshalb viele am Thema Interessierte von einem mit einem Schrittmotor realisierbaren Projekt letztlich Abstand nehmen, obwohl sie meist sogar in der Lage wären, die erforderliche Anwendungssoftware selbst zu schreiben.

Denn ein Nachteil dieser Technik ist die relativ aufwändige Ansteuerung des Motors. Hierzu dienen heute entweder Standalone-Mikroprozessorsteuerungen oder an Rechnersysteme angebundene Steuerungs-lösungen.

Problemlöser USB-Interface

Zu letzterer Gattung zählt unser universell einsetzbares Schrittmotoren-Interface.

Das mit einer externen Spannung zu versorgende Interface wird dank implementierter USB-Intelligenz ganz einfach per USB an einen PC angeschlossen und von dort angesteuert. Damit fällt eine wesentliche Hemmschwelle, die umständliche USB-Schnittstellenprogrammierung, weg. Und für den Einstieg gibt es zum Bausatz ein kleines Beispielprogramm sowie die zugehörige Quelltext-Datei in Visual C++, die als Grundlage für die Anbindung eigener Projekte dienen kann.

Das Interface selbst kann gleich zwei Schrittmotoren kleinerer Leistung antreiben, denn in den meisten Anwendungen wird man zwei Motoren auch einsetzen wollen.

Zusätzlich bietet das Interface noch 4 Schalteingänge, die als Endschalter der Anwendungs-konfiguration belegt sind. Und schließlich sind noch zwei Open-Collector-Schaltausgänge vorhanden, die z. B. das Ansteuern von Leuchtdioden oder Relais aus der Steuer-Software möglich machen.

Das Interface kann wahlweise über die Befestigungslöcher der Platine in ein vorhandenes Gerätegehäuse oder in ein passendes, optional erhältliches Gehäuse eingebaut werden.

Als „Hardware-Interface“ der Baugruppe zu den Motoren, Schaltern usw. dient

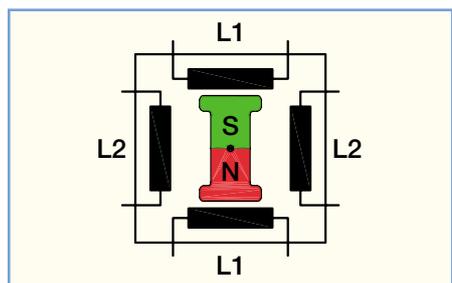


Bild 1: Der prinzipielle Aufbau eines Schrittmotors

Bild 2: Standard-Schrittmotor mit 96 Schritten je Umdrehung (ELV-Best.-Nr. 42-011-04)



eine 25-polige Sub-D-Buchse, über deren Gegenstück die einzelnen Leitungen zu den Motoren usw. bequem anschließbar sind.

Funktion

Schrittmotoren

Die gebräuchlichen Schrittmotoren (auch als Stepper oder Steppmotoren bekannt) bestehen aus einem Permanentmagneten als Rotor und einer geraden Anzahl von Spulen als Elektromagnete im Stator. Die Abbildung 1 zeigt den prinzipiellen Aufbau eines Schrittmotors, bestehend aus dem Rotor (Permanentmagnet in der Mitte) und 4 Spulen. Die Spulen sind gegenüberliegend angeordnet, sie bilden so genannte Polpaare (2 Polpaare). Sie befinden sich im Stator und erzeugen bei Stromfluss ein Magnetfeld. Je nach Stromrichtung wird ein „N“- oder „S“-Magnetfeld erzeugt. Dieses Magnetfeld wirkt auf den Rotor in der Mitte des Schrittmotors, der auf die magnetischen Änderungen reagiert. Ändert sich das Magnetfeld der Statoren durch entsprechende Ansteuerung der Spulen, so dreht sich der Rotor.

Bei dem hier gezeigten vereinfachten Schrittmotor können im Vollschrittbetrieb 4 Schritte pro Umdrehung realisiert werden. Ein Halbschrittbetrieb wird durch entsprechende Halbschritt-Ansteuerung erzeugt, wodurch sich die Schritte auf 8 pro Umdrehung verdoppeln.

Die Anzahl der Polpaare beträgt in industriell gebräuchlichen Schrittmotoren natürlich weit mehr als zwei. Reale Schrittmotoren können dadurch eine Umdrehung in sehr viele Schritte auflösen. Der in Abbildung 2 gezeigte, von ELV angebotene Schrittmotor benötigt beispielsweise 96 Schritte für eine Umdrehung im Halbschritt-Betrieb.

Des Weiteren ist zu erwähnen, dass man zwei Arten von Schrittmotoren unterscheidet. Zum einen gibt es die Gruppe der Bipolar-Motoren. Dieser Motortyp weist ein hohes Drehmoment auf, da durch alle Wicklungen ständig ein Strom fließt. Zum anderen gibt es die Gruppe der Unipolar-Motoren. Diese bieten den Vorteil einer einfacheren Ansteuerung. Mit der USI-2-

Hardware ist eine Ansteuerung beider Motortypen möglich.

Beide Motortypen kennen wiederum zwei verschiedene Betriebsarten, und zwar den Voll- und Halbschrittbetrieb. Das höchste Drehmoment erreicht ein Schrittmotor im Vollschrittbetrieb, wobei allerdings die Schrittauflösung auf volle Schritte begrenzt ist. Benötigt man eine höhere Auflösung, d. h. mehrere Schritte pro Umdrehung, so wählt man den Halbschrittbetrieb, nimmt aber das geringere Drehmoment in Kauf. Die hier vorgestellte USI-2-Hardware arbeitet ebenfalls im Halbschrittbetrieb, um mit dem angeschlossenen Schrittmotor die höchstmögliche Auflösung erreichen zu können. Nur im Halbschrittbetrieb lässt sich der (im Regelfall auf den Motoren beschriftete) minimale Schrittwinkel erreichen. Der in der Abbildung 2 dargestellte Schrittmotor (ELV-Best.-Nr. 42-011-04) erreicht eine Auflösung von 3,75 deg. Diese Bezeichnung besagt, dass der kleinste Schrittwinkel = 3,75 Grad des Vollkreises ist. Dieser Schrittmotor benötigt dadurch 96 Schritte ($360^\circ/3,75^\circ$), um eine gesamte Drehung zu durchlaufen. Ein weiterer von ELV angebotener Schrittmotor (ELV-Best.-Nr. 42-272-72) besitzt eine höhere Auflösung (1,8°/Schritt). Dieser benötigt deshalb 200 Schritte ($360^\circ/1,8^\circ$), um eine gesamte Drehung zu durchlaufen, und ist damit noch genauer positionierbar.

Die hier vorgestellte Schaltung eignet sich natürlich auch zum Ansteuern anderer Schrittmotoren kleinerer Leistung, sofern man die in den technischen Daten des USI 2 gesetzten Bedingungen einhält.

Beschaltung/Anschluss

Die Beschaltung des USB-Schrittmotoren-Interfaces ist vom Anwender je nach geplanter Verwendung individuell zu realisieren, wobei jedoch auch hier das Augenmerk auf die Bedingungen der technischen Daten zu richten ist. An der 25-poligen SUB-D-Buchse lassen sich über einen entsprechenden Stecker die jeweiligen Ein- und Ausgänge abgreifen.

Ein entsprechendes Anwendungsbeispiel ist in Abbildung 3 zu sehen. Die

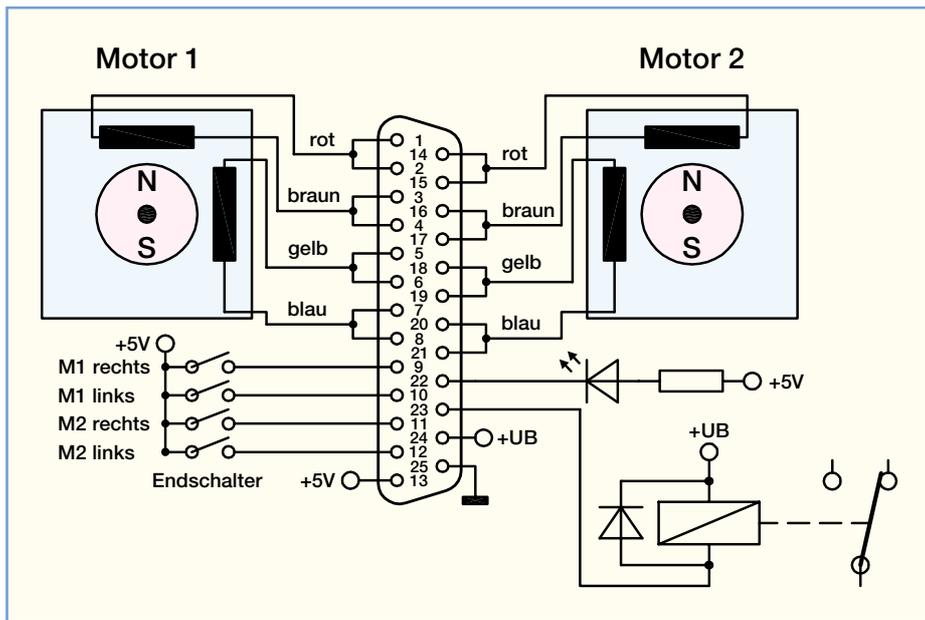


Bild 3: Beschaltungsbeispiel für das Interface

Schrittmotoren sind an die jeweiligen Pins anzuschließen. Wegen der hohen Strombelastung sind die einzelnen Motorleitungen jeweils über 2 Buchsen-Pins geführt. Für die Verwendung des von ELV angebotenen Schrittmotors (Best.-Nr.: 42-011-04) sind die jeweiligen Farben der Anschlussleitungen zur polrichtigen Verdrahtung in der Abbildung 3 angegeben. Bei dem angebotenen Schrittmotor handelt es sich um einen unipolaren Schrittmotor, d. h., dass jede Wicklung einen Mittelabgriff besitzt. Dieser ist jeweils über eine Ader (weiß und schwarz) nach außen geführt. Bei dem Anschluss des Schrittmotors an die USI-2-Hardware sind diese nicht anzuschließen (offen zu lassen).

Soll ein anderer Schrittmotor eingesetzt werden, so ist die Anschlussbelegung des Motors zu ermitteln, da die Farbcodierungen nicht genormt sind. Die zusammengehörenden Leitungspaare sind aber relativ einfach mit einem Ohmmeter zu bestimmen, lediglich die korrekte Polung der Spulen ist durch Funktionstest mittels der Schrittmotorsteuerung zu ermitteln.

Des Weiteren sind auf der USI-2-Hardware noch vier Endabschalter-Eingänge implementiert. Für jeden Motor ist pro Drehrichtung ein Endschalter vorgesehen. Ist einer dieser Schalter geschlossen (Ver-

bindung nach 5 V), so wird der jeweilige Motor gestoppt und kann erst nach Öffnen des Endschalters wieder in dieselbe Richtung drehen.

Als weiteres Leistungsmerkmal bietet das Schrittmotoren-Interface, wie bereits erwähnt, noch zwei Schaltausgänge. Hiermit lassen sich beispielsweise LEDs, Relais usw. ansteuern.

Datenübertragung

Allgemeines

Die USB-Datenübertragung funktioniert in dieser Schaltung ähnlich wie eine serielle Datenübertragung. Sie erfolgt mit einer Baud-Rate von 19.200 Bit/s, 8 Datenbits, gerader Parität und einem Stoppbit. Da eine genauere Erklärung dieser Begriffe und Abkürzung den Rahmen dieses Artikels sprengen würde, wollen wir nicht näher darauf eingehen. Der interessierte Leser kann im „ELVjournal“ 1/04 im Artikel „USB-DA-Wandler“ detaillierte Informationen hierzu nachlesen.

Kommen wir deshalb gleich zur oberen Ebene der Übertragung, dem verwendeten Protokoll. Dieses besteht beim Schrittmotoren-Interface pro Befehl aus mehreren zu übertragenden Bytes, die innerhalb eines Protokollrahmens zu übertragen sind. In Tabelle 1 sind die verwendeten Steuerzeichen abgebildet. Jedes Datenpaket beginnt mit dem Steuerzeichen „STX“ (02hex) und endet mit dem Zeichen „ETX“ (03hex). Dazwischen sind die jeweiligen Parameter zu übertragen. Die folgende Zeile stellt so eine zusammengesetzte Befehlszeile dar:

```
<STX> <Befehl> <Parameter 1>
<Parameter 2> <ETX>
```

Der Empfänger erkennt an diesen Zeichen genau, wann ein Datenpaket beginnt bzw. endet, und weiß somit, wann ein Befehl oder dessen Antwort komplett empfangen wurde. Kommen diese Steuerzeichen jedoch in den Daten vor, so müssen sie entsprechend umgeformt werden, um einen vorzeitigen Neustart bzw. Abbruch der Kommunikation zu verhindern. Die Konvertierung der Zeichen wird entsprechend folgender Auflistung durchgeführt:

```
<STX> → <ENQ> <DC2>
<ETX> → <ENQ> <DC3>
<ENQ> → <ENQ> <NAK>
```

Taucht eines der drei Zeichen „STX“, „ETX“ oder „ENQ“ in den Daten auf, ist es also durch die entsprechende Zeichenfolge auszutauschen. Empfängerseitig wird diese Umsetzung jedoch automatisch wieder rückgängig gemacht, um die eigentlichen Nutzdaten wiederherzustellen. Der entsprechende Befehlsatz des USI 2 ist in Tabelle 2 zu sehen und dort erläutert.

Übertragungsbeispiel

Als Übertragungsbeispiel für die beschriebene Vorgehensweise soll der angeschlossene Motor 1 um 1000 Schritte nach rechts drehen und der Motor 2 um 255 Schritte nach links drehen. Die Schrittmotoren sollen auf höchster Geschwindigkeitsstufe (255) die gewünschten Schritte einstellen. Bei der Übertragung der Parameter lassen sich jeweils nur 8 Bit breite Datenwörter übertragen. Somit ist die gewünschte Schrittzahl (1000 für Motor 1) auf 2 Bytes aufzuteilen. Dies geschieht, indem die Schrittzahl durch 256 geteilt ($1000/256 = 3$ plus Rest 232) und dies als Parameter 2 gesendet wird. Der Restwert wird mit dem Modulo-Operator ($1000\%256 = 232$) ermittelt und als Parameter 1 gesendet. Für das Beispiel ergeben sich die Parameter <232> und <3>. Die weiteren Steuerzeichen sind der Tabelle 2 zu entnehmen. Die gesamte Befehlszeile zur Steuerung von Motor 1 sieht wie folgt aus:

```
<STX> <Steuerzeichen M1_rechts>
<Schritte%256> <Schritte/256>
<ETX>

<STX> <'r'> <232> <3> <ETX>
```

In der unteren Zeile ist zu erkennen, dass als Parameter 2 das Steuerzeichen ETX (0x03) übertragen wird. Dies würde zu einem vorzeitigen Abbrechen der Verbindung führen. Deshalb ist der Befehlsatz wie folgt zu wandeln:

```
<STX> <'r'> <232> <ENQ> <DC3>
<ETX>
```

Tabelle 1: Verwendete Steuerzeichen	
Zeichen	Wert
STX	02hex
ETX	03hex
ENQ	05hex
ACK	06hex
DC2	12hex
DC3	13hex
NAK	15hex

Bedeutung	Steuerzeichen	Parameter 1	Parameter 2	Antwort
M1_links	'l' (0x6C)	(Schritte%256)	(Schritte/256)	jew. ACK/NAK
M1_rechts	'r' (0x72)	(Schritte%256)	(Schritte/256)	jew. ACK/NAK
M2_links	'L' (0x4C)	(Schritte%256)	(Schritte/256)	jew. ACK/NAK
M2_rechts	'R' (0x52)	(Schritte%256)	(Schritte/256)	jew. ACK/NAK
Geschwindigkeitsstufe	'G' (0x47)	0x01..0xff	---	jew. ACK/NAK
Startsignal, damit beide Motoren gleichzeitig starten	'S' (0x53)	---	---	jew. ACK/NAK
Ausgang 1	'o' (0x6F)	'1'=0x31 – setzen '0'=0x30 – rücksetzen	---	jew. ACK/NAK
Ausgang 2	'O' (0x4F)	'1'=0x31 – setzen '0'=0x30 – rücksetzen	---	jew. ACK/NAK
Endschalter-Position (wird nach Einstellung der Schritte von der USI-2-Hardware gesendet)	'E' (0x45)	Endschalter-Position (1 Byte) Endschalter-Byte (0x30)	---	

Die USI-2-Hardware setzt diese empfangenen Daten anschließend wieder in Nutzdaten um, so dass diese die gewünschten 1000 Schritte einstellt.

Die Protokollzusammenstellung für Befehl 2, den Motor 2 um 255 Schritte nach links zu drehen, geschieht prinzipiell wie im vorherigen Beispiel und sieht wie folgt aus:

```
<STX> <Steuerzeichen M2_links>
<Schritte%256> <Schritte/256>
<ETX>

<STX> <'L'> <255> <0> <ETX>
```

Zur Einstellung der höchsten Geschwindigkeitsstufe ist der Befehl wie folgt zusammenzustellen:

```
<STX> <Steuerzeichen Geschwindigkeit> <Geschwindigkeitsstufe>
<ETX>

<STX> <'G'> <255> <ETX>
```

Zum gleichzeitigen Starten der beiden Motoren wurde ein weiteres Steuerzeichen implementiert. Hierfür ist das Steuerzeichen 'S' zu übertragen:

```
<STX> <'S'> <ETX>
```

Eine erfolgreiche Übertragung jeder Befehlszeile wird folgendermaßen quittiert:

```
<STX> <ACK> <ETX>
```

Ein fehlerhafter Empfang einer Befehls-

zeile wird seitens der USI-2-Hardware mit folgendem Protokoll quittiert:

```
<STX> <NAK> <ETX>
```

In diesem Falle ist die Übertragung zu wiederholen.

Nach dem Empfang des Start-Zeichens (Steuerzeichen = 'S') startet die USI-2-Hardware mit der Einstellung der gewünschten Schritte. Sind die Schritte abgearbeitet, so wird von der USI-2-Hardware eine Befehlszeile zurückgesendet. Zum einen ist hieraus zu erkennen, dass das Schrittmotoren-Interface die gewünschten Positionen erreicht hat. Zum anderen werden in diesem Byte die aktuellen Positionen der Endschalter übermittelt. Die empfangene Befehlszeile sieht wie folgt aus:

```
<STX> <'E'> <Endschalter> <ETX>
```

Die Endschalterpositionen (siehe Tabelle 3) werden innerhalb eines Bytes gesendet. Damit dieses Byte nicht in den Bereich der Steuerzeichen (STX, ETX) ragt, wird zu diesem Byte die Zahl 0x30 addiert, welche vor der Auswertung zu subtrahieren ist.

Die Ansteuerung der Open-Collector-Schaltausgänge erfolgt auf gleiche Weise. Das Senden des Parameters <0x31> bewirkt ein Durchschalten des Open-Collector-Schaltausganges. Zum Einschalten des Ausgangs 2 lautet der Befehlssatz:

```
<STX> <'O'> <0x31> <ETX>
```

Zum besseren Verständnis der Ansteuerung des USB-Schrittmotoren-Interface befindet sich auf der dem Bausatz beiliegenden Programmdiskette ein Visual-C++-Beispielprojekt mit zugehörigen Quelltexten.

Weitere Infos und Beispielprojekte zum Umgang mit der USB-Schnittstelle lassen sich auf der Internet-Seite des USB-Chip-Herstellers „www.ftdichip.com“ abrufen.

Schaltung

Die Schaltung des USB-Schrittmotoren-Interface (USI 2) ist übersichtshalber in zwei Abbildungen (Abbildung 4: USB-Umsetzung, Abbildung 5: Schrittmotorsteuerung) dargestellt.

USB-Umsetzung

Das zentrale Element der USB-Umsetzung bildet der USB-Controller (IC3/FTU232AM), der speziell für die Konvertierung

zwischen USB und RS 232 entwickelt wurde. Dieses IC beinhaltet eine Art Mikrocontroller, so dass eine Taktversorgung gewährleistet sein muss. Der interne Oszillator wird mit dem Quarz Q 1 und den Kondensatoren C 14 und C 15 auf eine Frequenz von 6 MHz stabilisiert. Ein definiertes Starten des USB-Controllers wird durch Beschalten des Reset-Pins mit T 1, R 5 bis R 7 und C 10 sichergestellt.

Zur Speicherung der USB-Erkennungsdaten (Vendor-ID, Product-ID, Seriennummer etc.) des USI 2 ist an die „Microwire“-Schnittstelle von IC 3 ein EEPROM vom Typ ELV 04386 (IC 2) angeschlossen.

Neben den beiden Leitungen für die Betriebsspannung besteht der USB aus zwei Datenleitungen (D+, D-). Diese sind jeweils über einen Widerstand zum Leitungsabschluss (R 2, R 4) mit dem USB-Controller IC 3 verbunden. Der Widerstand R 3 dient als Pull-up-Widerstand des USB. Über den definierten „High“-Pegel der D+-Datenleitung erkennt der PC eine angeschlossene USI-2-Hardware. Die Leuchtdioden D 3 und D 4 dienen als Statusanzeige für die Daten-Übertragung, wobei die grüne LED (D 3) einen Sendevorgang und die rote LED (D 4) einen Datenempfang der USI-2-Hardware signalisiert.

Tabelle 3: Endschalterposition

Endschalter	Endschalter-Byte
M1_rechts	xxxx xxx1
M1_links	xxxx xx1x
M2_rechts	xxxx x1xx
M2_links	xxxx 1xxx

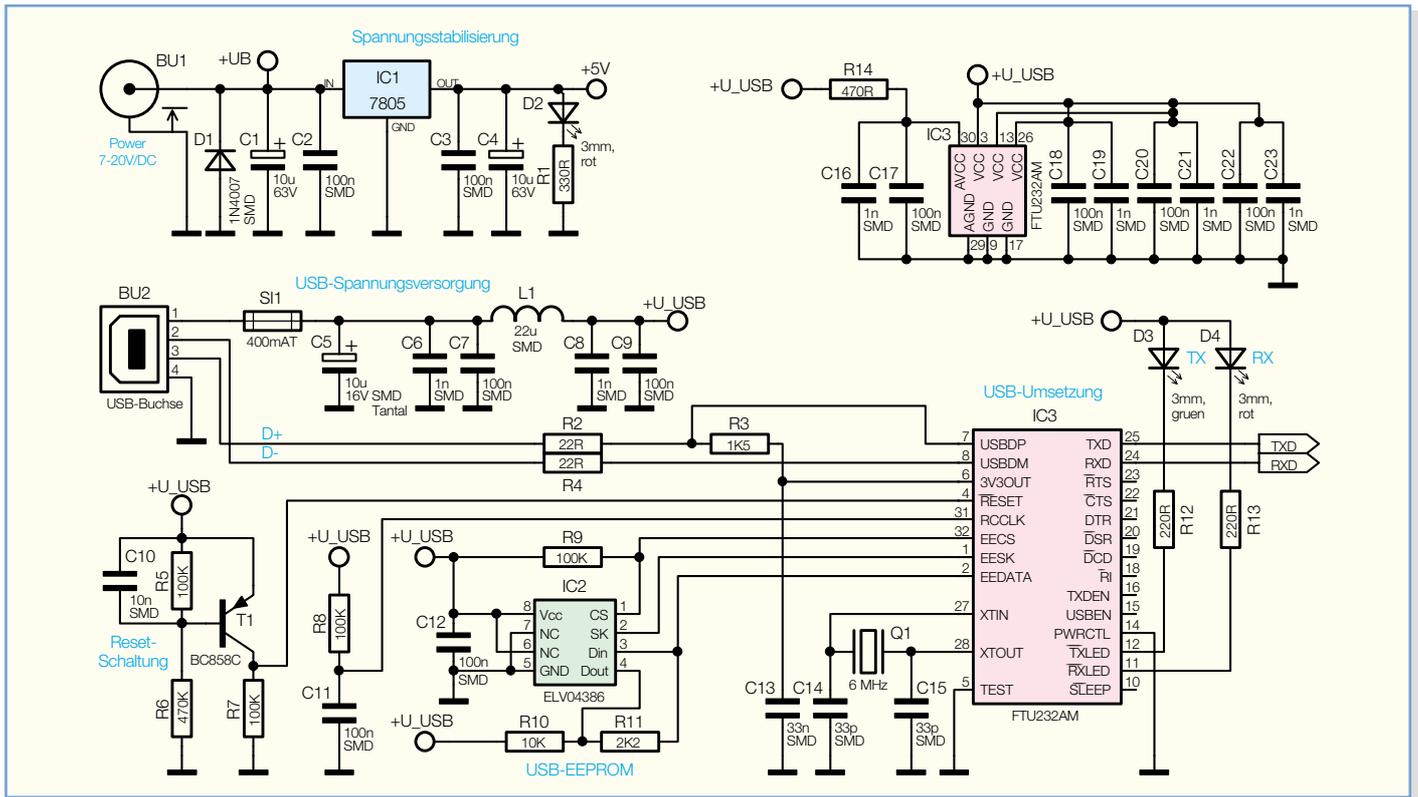


Bild 4: Das Schaltbild des USB-Umsetzers und der Spannungsversorgung

Spannungsversorgung

Die Spannungsversorgung der USB-Umsetzung erfolgt über den USB, damit sich die angeschlossene USI-2-Hardware auch ohne eingeschaltete Versorgungsspannung bei einem PC anmeldet. Der USB stellt eine Spannung von 5 V zur Verfügung. Diese Betriebsspannung gelangt über die USB-Buchse (BU 2/Pin 1 und Pin 4) auf die Schaltung. Die Kondensatoren C 5 bis C 9 und C 16 bis C 23 sowie die Induktivität L 1 dienen zur Stabilisierung dieser Spannung bzw. zur Unterdrückung hochfrequenter Störungen. Die Sicherung SI 1 ist als Überlastungsschutz des USB implementiert.

Die weitere Spannungsversorgung des USI 2 ist mit dem Spannungsregler (IC 1) realisiert. Dieser Spannungsregler hat einen Eingangsspannungsbereich von 7 bis 20 V/DC. Zur Versorgung wird ein geeignetes Steckernetzteil mit Hohlstecker empfohlen, das ausreichend Strom zur Versorgung der angeschlossenen Motoren bereitstellen kann. In Verbindung mit den von ELV angebotenen Schrittmotoren kann ein handelsübliches 12-V-Steckernetzteil verwendet werden. Der Elko C 1 dient zur Glättung der Eingangsspannung. Die Diode D 1 ist als Verpolungsschutz implementiert. Der Elko C 4 und die Kondensatoren C 2 und C 3 dienen zur Stör- bzw. Schwingneigungsunterdrückung von IC 1. Die LED D 2 signalisiert eine eingeschaltete Betriebsspannung.

Schrittmotorsteuerung

Kommen wir nun zur Beschreibung der Schrittmotorsteuerung (Abbildung 5).

Das zentrale Element bildet hier der Mikrocontroller IC 4. Dieser steuert die Schrittmotor-Treiber und -Schaltausgänge an. Weiterhin werden die Endschalter-Eingänge vom Mikrocontroller ausgewertet. Des Weiteren erhält der Mikrocontroller seine Befehle über die vorgeschaltete USB-Umsetzung. Diese setzt ankommende Befehle vom USB-Format in RS-232-Befehle und auch abgehende RS-232-Befehle in das USB-Format um, so dass der Mikrocontroller diese an seinen seriellen Schnittstellen-Pins (RXD und TXD) lesen und ausgeben kann.

Der interne Oszillator des Mikrocontrollers wird durch den Quarz Q 2 und die beiden Kondensatoren C 28 und C 29 auf eine Frequenz von 8 MHz stabilisiert. Die Kondensatoren C 24 bis C 27 dienen zur Störunterdrückung.

Der Programmieradapter PRG 1 und der Widerstand R 15 ermöglichen eine Programmierung des Mikrocontrollers in der Serienproduktion.

Die eigentliche Ansteuerung der Schrittmotoren übernehmen die integrierten Schaltkreise IC 5 und IC 6. Bei diesen Bausteinen handelt es sich um spezielle Treiber-ICs für Schrittmotoren mit integrierter Pulsweitenmodulation zur Strombegrenzung. Diese sind an die Ports PB und PC des Mikrocontrollers (IC 4) ange-

schlossen. Die Widerstände R 16 bis R 27 dienen als „Pull-down-Widerstände“ dieser Datenleitungen und sorgen somit für ein definiertes Potential der Datenleitungen im „nichtaktiven“ Zustand. Die Schrittmotor-Treiber werden vom Mikrocontroller über bestimmte Bitmuster angesteuert.

Mit den Trimpoties R 42 und R 43 lässt sich eine hardwaremäßige Strombegrenzung einstellen. Der Strompegel wird je nach Motor-Typ mit der Spannung an den Eingängen VREF an Pin 11, 15 von IC 5 und IC 6 vorgegeben.

Weiterhin lassen sich mit dem Mikrocontroller vier Endschalter-Positionen auswerten. Für jeden Motor ist pro Richtung ein Endschalter vorgesehen. Wird der jeweilige Endschalter betätigt, so stoppt der Motor und läuft nicht weiter in die vorgegebene Richtung. Die Endschalter-Auswertung wurde mit den Transistoren T 4 bis T 7, den Widerständen R 34 bis R 41 und den Kondensatoren C 30 bis C 33 realisiert. Im Ruhezustand (Endschalter offen/0 V) sperrt der jeweilige Transistor, und an den Mikrocontroller-Ports liegt ein „High“-Pegel an. Im aktiven Zustand (Endschalter geschlossen/5 V) wird der jeweilige Transistor durchgeschaltet und am Mikrocontroller-Pin liegt „Low“-Pegel an. Der Mikrocontroller erkennt dies und stoppt den Schrittmotor. Die Richtung wird erst dann wieder freigegeben, wenn der zugehörige Endschalter-Eingang wieder geöffnet wird.

Mit den Transistoren T 2 und T 3 in Verbindung mit den Widerständen R 28

und R 29 wurden zwei Schaltausgänge realisiert. Mit diesen als „Open-Collector“ geschalteten Ausgängen lassen sich beispielsweise LEDs oder Relais schalten.

Nachbau

Der Nachbau des USB-Schrittmotoren-Interfaces USI 2 erfordert ein wenig Geschick, da die verwendeten Bauelemente fast ausschließlich in SMD-Technik ausgeführt sind, um ein kompaktes Design zu erreichen. Neben einem geregelten Lötcolben mit sehr feiner Spitze, SMD-Lötzinn sowie Entlötlitze sollte auch eine SMD-Pinzette zum Positionieren der kleinen Bauteile nicht fehlen. Auch eine starke und möglichst beleuchtbare Standlupe leistet hier gute Dienste. Der Aufbau erfolgt anhand des Bestückungsdrucks, des Platinenfotos sowie der Stückliste.

Er beginnt mit den ICs 2 bis 6. Diese haben einen sehr geringen Pin-Abstand und sind am besten zu bestücken, wenn ringsum noch keine Bauteile die Lötarbeiten behindern. Beim Bestücken dieser Bauteile ist besonders auf die korrekte Einbaulage zu achten, da nachträgliche Korrekturen nur sehr schwer durchführbar sind und dabei (nicht nur im Hobbylabor) meist Platine und/oder Bauelement beschädigt werden können. Beim USB-Controller (IC 3) und dem Mikrocontroller (IC 4) ist die Pin 1 zugeordnete Ecke entweder angechrägt oder durch eine kreisförmige Ausfräsung des Gehäuses gekennzeichnet, die sich auch im Bestückungsdruck wiederfindet. Die restlichen ICs sind an der Pin 1 zugeordneten Seite abgeflacht bzw. durch eine Gehäusekerbe gekennzeichnet. Bei der Bestückung der ICs wird zunächst jeweils ein Lötpad vorverzinnt, an dem man den zugehörigen Pin zuerst verlötet.

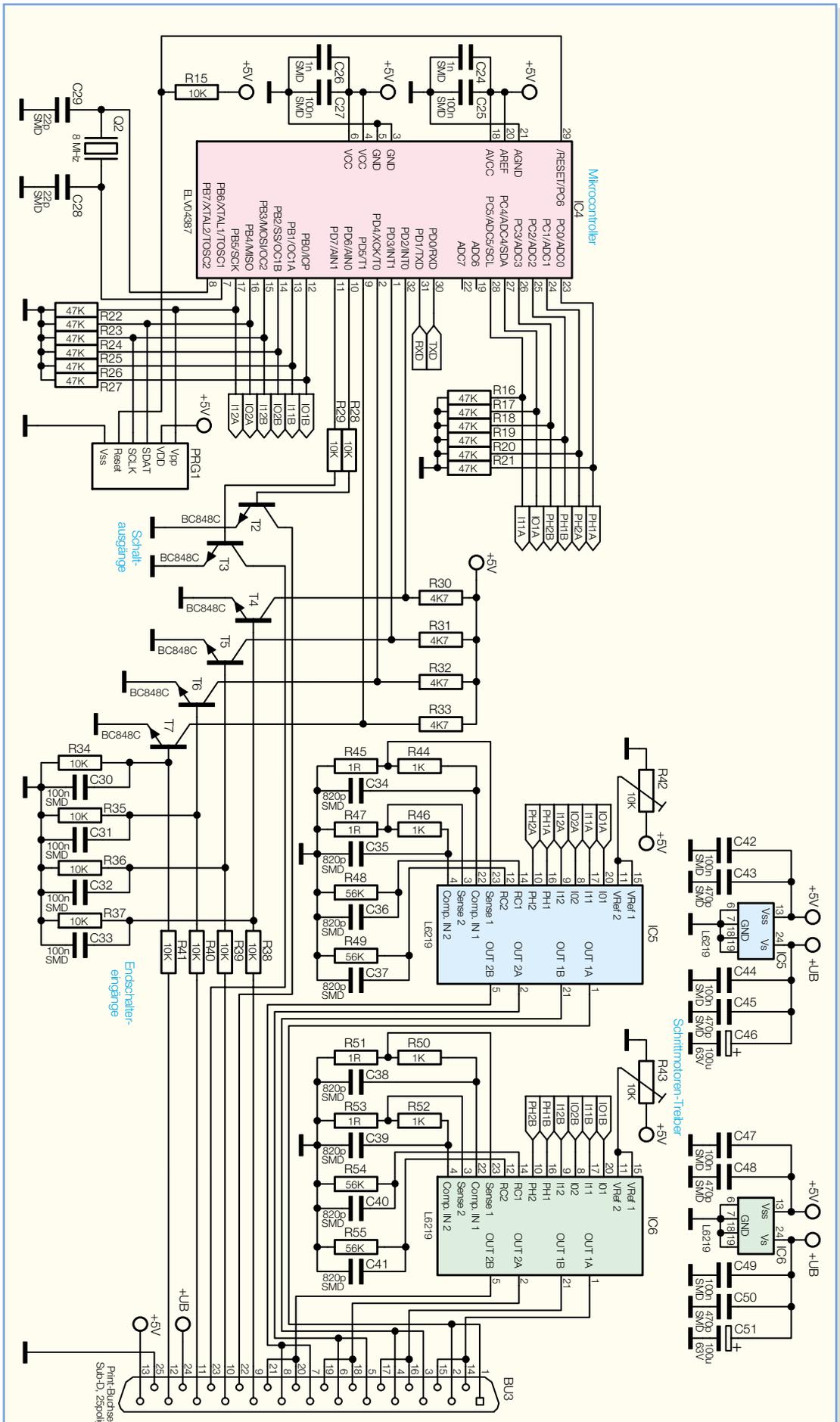


Bild 5: Das Schaltbild der Schrittmotorsteuerung

Stückliste: USB-Schrittmotoren-Interface USI 2

Widerstände:

1 Ω/SMD	R45, R47, R51, R53
22 Ω/SMD	R2, R4
220 Ω/SMD	R12, R13
330 Ω/SMD	R1
470 Ω/SMD	R14
1 kΩ/SMD	R44, R46, R50, R52
1,5 kΩ/SMD	R3
2,2 kΩ/SMD	R11
4,7 kΩ/SMD	R30–R33
10 kΩ/SMD	R10, R15, R28, R29, R34–R41
47 kΩ/SMD	R16–R27
56 kΩ/SMD	R48, R49, R54, R55
100 kΩ/SMD	R5, R7–R9
470 kΩ/SMD	R6
PT10, liegend, 10 kΩ	R42, R43

Kondensatoren:

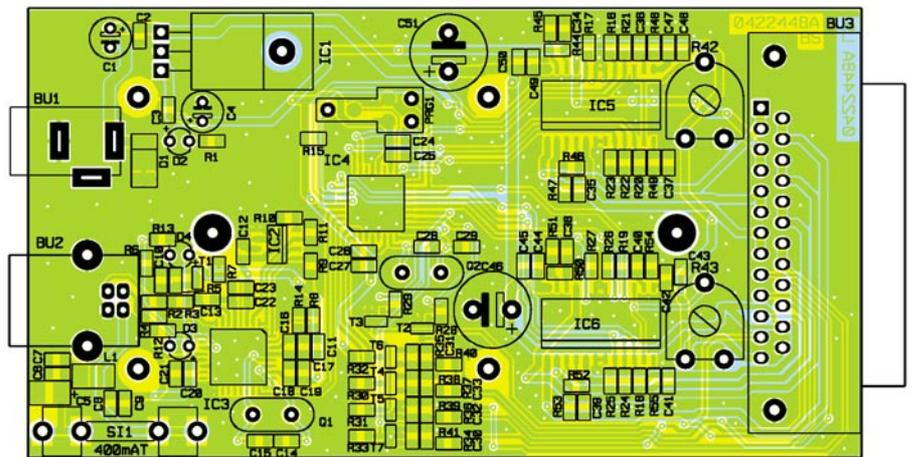
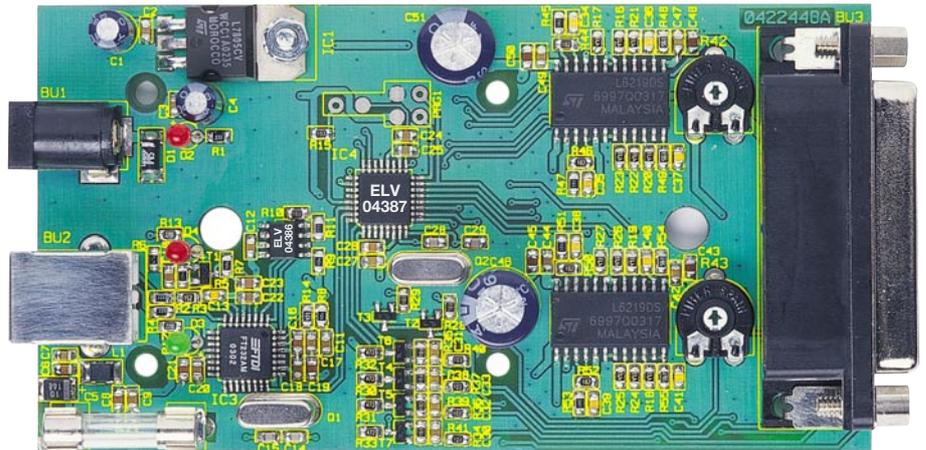
22 pF/SMD	C28, C29
33 pF/SMD	C14, C15
470 pF/SMD	C43, C45, C48, C50
820 pF/SMD	C34–C41
1 nF/SMD	C6, C8, C16, C19, C21, C23, C24, C26
10 nF/SMD	C10
33 nF/SMD	C13
100 nF/SMD	C2, C3, C7, C9, C11, C12, C17, C18, C20, C22, C25, C27, C30–C33, C42, C44, C47, C49
10 µF/16 V/SMD	C5
10 µF/63 V	C1, C4
100 µF/63 V	C46, C51

Halbleiter:

7805	IC1
ELV04386/SMD	IC2
FT8U232AM/SMD	IC3
ELV04387/SMD	IC4
L6219, SMD	IC5, IC6
BC858C	T1
BC848C	T2–T7
SM4007/SMD	D1
LED, 3 mm, rot	D2, D4
LED, 3 mm, grün	D3

Sonstiges:

Quarz, 6 MHz, HC49U	Q1
Quarz, 8 MHz, HC49U	Q2
SMD-Induktivität, 22 µH	L1
Hohlsteckerbuchse, 2,1 mm, print	BU1
USB-B-Buchse, winkelprint	BU2
SUB-D-Buchsenleiste, 9-polig, print	BU3
Sicherung, 0,4 A, träge	SI1
Platinensicherungshalter (2 Hälften), print	SI1
1 Zylinderkopfschraube, M3 x 8 mm	4
1 Kunststoffschrauben, 2,5 x 6 mm	1
1 Mutter, M3	1
1 Fächerscheibe, M3	1
1 3,5"-Diskette USI-2-Software	



Ansicht der fertig bestückten Platine des USB-Dual-Schrittmotoren-Interfaces mit zugehörigem Bestückungsplan

tet. Im Anschluss daran ist ein zweiter Pin an der diagonal gegenüberliegenden Seite zu verlöten. Dabei ist darauf zu achten, dass alle Anschlüsse des ICs auf den zugehörigen Löt pads aufliegen, um spätere Kontaktfehler durch ungenügende Verlotung auszuschließen. Bevor die weiteren Anschlüsse mit der Leiterplatte verlötet werden, ist nochmals die richtige Position zu überprüfen. Nach dem Verlöten aller IC-Pins und sorgfältiger Kontrolle auf Kurzschlüsse (überflüssiges Zinn ggf. mit feiner Entlötlitze absaugen) geht es nun an die weiteren SMD-Komponenten und zwar SMD-Widerstände, -Kondensatoren und -Spule. Hier wird wieder zunächst jeweils ein Löt pad auf der Leiterplatte vorverzinnt, bevor man das Bauteil mit der Pinzette erfasst, positioniert und am vorverzinnten Pad anlötet. Nach der Kontrolle der korrekten Position des Bauteils ist der zweite Anschluss zu verlöten. Die Kondensatoren sollten erst direkt vor dem Bestücken einzeln aus der Verpackung genommen werden, da diese keinen Aufdruck tragen, der über den Wert informiert. Als Nächstes erfolgt die Bestückung des SMD-Elkos C 5. Bei SMD-Elkos ist üblicherweise der +Pol gekennzeichnet. Im

Anschluss daran sind die SMD-Transistoren und -Dioden in gleicher Weise auf der Leiterplatte zu verlöten. Hier ist jedoch besonders auf die richtige Polung zu achten, die sich bei den Transistoren aus der Pin-Konfiguration ergibt. Die Dioden sind durch eine Ring-Markierung an der Kathode gekennzeichnet.

Vor der weiteren Bestückung sind alle SMD-Lötstellen sorgfältig zu kontrollieren, ggf. unter Zuhilfenahme einer starken Lupe. Ist alles in Ordnung, beginnt die Bestückung der konventionell bedrahteten Bauelemente. Bei diesen Bauteilen ist darauf zu achten, dass überstehende Drahtenden auf der Lötseite der Platine mit einem Elektronik-Seitenschneider so abgetrennt werden, dass einerseits die Lötstelle nicht beschädigt wird, andererseits hervorstehende Drahtenden im späteren Betrieb keine Kurzschlüsse hervorrufen können. Die beiden Quarze (Q 1 und Q 2) und die Elkos (C 1, C 4, C 46 und C 51) sind in stehender Position zu bestücken und zu verlöten. Bei den Elkos ist unbedingt auf richtige Polung zu achten, da diese sonst im schlimmsten Fall sogar explodieren könnten. Üblicherweise sind diese am Minuspol gekennzeichnet. Daraufhin sind die Trimmpotis

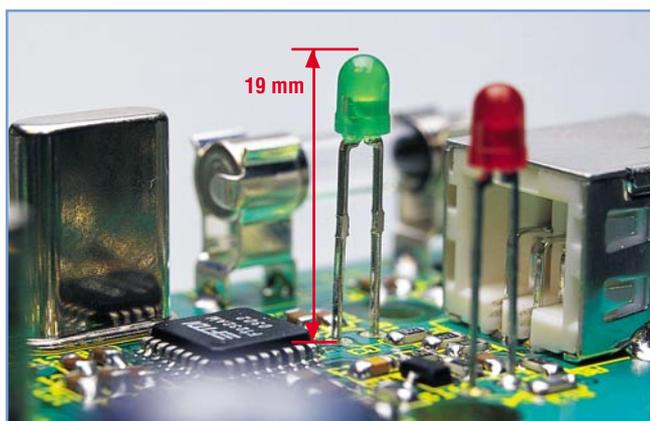


Bild 6: Die LED-Montage im Detail

R 42 und R 43 zu bestücken, deren richtige Bestückungsrichtung sich aus der Pin-Konfiguration ergibt.

Die Buchsen BU 1 bis BU 3 und der Sicherungshalter müssen direkt auf der Leiterplatte aufliegen, bevor die Anschlüsse verlötet werden, da die Lötstellen bei späterer mechanischer Belastung der Buchsen sonst zu stark beansprucht werden.

Nun erfolgt die polrichtige Bestückung der Leuchtdioden, hier ist die Anode (+) durch den längeren Anschlusspin gekennzeichnet. Die LEDs sind so zu bestücken, dass sich die LED-Spitze etwa 19 mm über der Platine befindet, so dass keine Probleme bei der Montage des optional erhältlichen Gehäuses auftreten (siehe Abbildung 6).

Die Bestückung des Spannungsreglers (IC 1) beginnt mit dem Abwinkeln der Anschlusspins sowie der Fixierung des Reglers mittels einer Zylinderkopfschraube (M3 x 8 mm), die von unten in die entsprechende Bohrung gesteckt wird. Von oben wird dann eine Zahnscheibe aufgesetzt und eine M3-Mutter aufgeschraubt. Erst dann erfolgt das Verlöten der Anschlüsse des Reglers. An der Abbildung 7 ist die richtige Bestückung des Spannungsreglers zu erkennen.

Abschließend, nach nochmaliger Über-

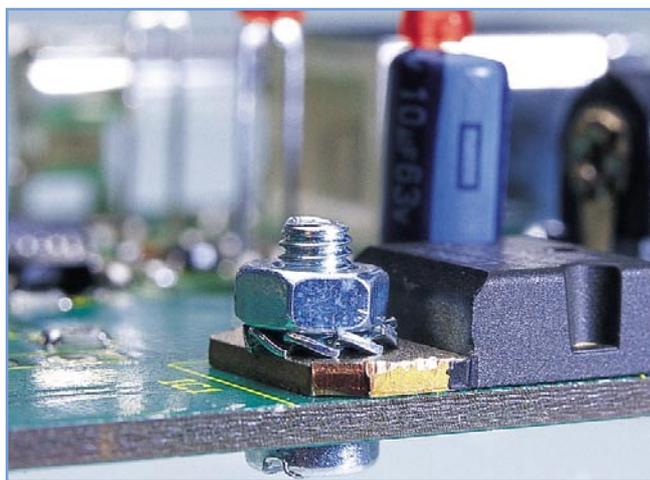


Bild 7: Die Bestückung des Spannungsreglers

prüfung der Leiterplatte auf Bestückungsfehler und Lötzinnbrücken, ist das Gehäuse zu montieren (optional erhältlich). Hierfür wird die Platine in die Gehäuseunterschale eingelegt, so dass die jeweiligen Buchsen in die Aussparungen passen, und dann mit vier selbstschneidenden Schrauben (M2,5 x 6 mm) verschraubt. Daraufhin ist die Gehäuseoberschale aufzulegen und mit zwei weiteren Schrauben (2,2

x 18,5 mm) von der Unterseite zu verschrauben.

Funktionskontrolle/ Inbetriebnahme

Nach Abschluss des Aufbaus ist das USB-Schrittmotoren-Interface mittels des mitgelieferten Testprogramms einfach zu überprüfen. Dazu verbindet man das Interface über ein USB-Verbindungskabel mit einem PC. Dieser erkennt die neu angeschlossene Hardware und verlangt nach kurzer Zeit einen USB-Treiber. Dieser Treiber (ftd2xx.inf) befindet sich auf der mitgelieferten Programmdiskette. Anschließend ist das Testprogramm „usi2_test.exe“ von der Diskette zu starten. In dem daraufhin erscheinenden Dialogfeld (siehe Abbildung 8) kann die Funktion des Schrittmotoren-Interfaces einfach überprüft werden.

Vor Testbeginn ist ein Schrittmotor laut Anschlussplan (siehe Kapitel „Beschalung/Anschluss“) anzuschließen. Außerdem ist eine geeignete Betriebsspannung für den Schrittmotor-Treiber und die Schrittmotoren über BU 1 anzulegen. Im Allgemeinen reicht hierfür ein normales, aber leistungsstarkes Steckernetzteil aus. Besonders effektiv arbeiten hier die modernen Schaltnetzteile, die auch problemlos, dazu bei geringem Volumen und geringer Wärmeentwicklung, die für die Motoren benötigten Ströme bereitstellen können.

Für die Inbetriebnahme ist mit den Trimpotentiometern R 42 und R 43 die individuelle Strombegrenzung einzustellen. Für die Verwendung des ELV-Schrittmotors ist die Mittelstellung der Trimpotentiometer ausreichend.

Im oberen Teil des Testprogramms ist links

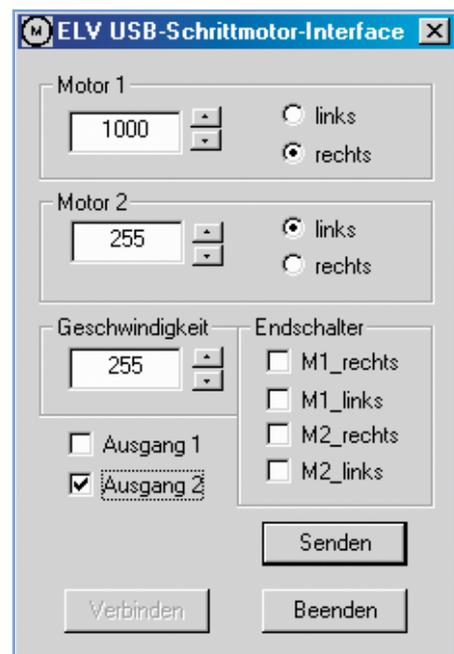


Bild 8: Das Testprogramm zum Schrittmotoren-Interface

die gewünschte Schrittzahl direkt oder mit Hilfe der „Wipp-Tasten“ für Motor 1 einzustellen. Mit der Auswahl „links“ oder „rechts“ lässt sich die gewünschte Drehrichtung auswählen. Die Drehrichtung wurde als Draufsicht auf den Motor und unter Berücksichtigung der vorgegebenen Anschlussverdrahtung festgelegt. Die Verwendung einer anderen Anschlussverdrahtung kann eine andere Drehrichtung nach sich ziehen. In diesem Falle ist die Verdrahtung der Spulen zu tauschen (Motor 1/A mit Motor 1/B und umgekehrt).

Für Motor 2 lassen sich die Einstellungen auf die gleiche Weise vornehmen. Im Eingabefeld „Geschwindigkeit“ lässt sich die Geschwindigkeitsstufe einstellen. Die höchste Stufe (255) bewirkt die schnellste Drehgeschwindigkeit des Schrittmotors.

Mit der Taste „Senden“ werden die eingestellten Daten protokollgemäß zusammengestellt und zur USI-2-Hardware gesendet. Dort wertet der Mikrocontroller die Daten aus und stellt die gewünschten Schritte ein. Ist dieses geschehen, so sendet der Mikrocontroller ein so genanntes „Ende-Flag“ zurück. Mit diesem Flag wird gleichzeitig die aktuelle Position der Endschalter übertragen. Diese werden von der Windows-Software ausgewertet und in dem Feld „Endschalter“ angezeigt.

Durch Setzen der Häkchen vor Ausgang 1 oder Ausgang 2 wird der jeweilige Ausgang geschaltet. Ein Häkchen bedeutet, dass der Open-Collector-Schaltausgang nach „low“ durchschaltet.

Funktionieren alle vorhandenen Features des Schrittmotoren-Interfaces reibungslos, so ist das Modul bereit für eigene Applikationen.

