

---

## Motorsteuerung mit H-Brücke

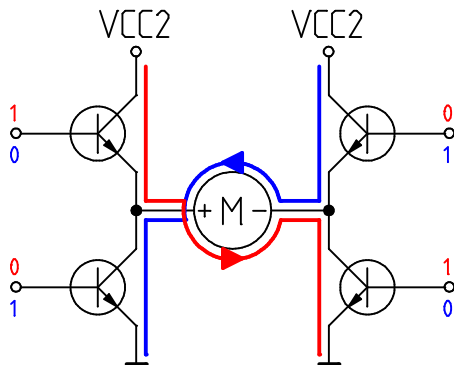
Gunther Zielosko

### 1. Einleitung

Die Steuerung von kleineren DC-Motoren mit Micro-Controllern wie dem BASIC-Tiger® gehört zu den nicht gerade seltenen Anwendungen dieser dienstbaren Geister. Auch in unserer Reihe mit Applikationen für den BASIC-Tiger® nehmen sie einen der vorderen Plätze ein. Erinnern wir uns z.B. an die Experimente mit Gleichstrom-Motoren (Applikation Nr. 019), wo wir Motoren ein- und ausgeschaltet, ihre Drehrichtung gewechselt und sogar die Drehzahl über Pulsweiten-Modulation eingestellt haben. Warum also noch eine neue Variante? Die Erklärung ist einfach, die bisherigen Lösungen haben den Prozessor einfach zu intensiv beschäftigt und die Steuerung mehrerer Motoren (z.B. für Roboter-Projekte!) wäre kaum noch zu realisieren. In der folgenden Applikation werden wir eine clevere Lösung kennenlernen, in der bei moderatem Rechenaufwand und minimalem I/O-Pinbedarf bis zu 62 Motoren unabhängig gesteuert werden können und zwar vorwärts und rückwärts in jeweils 127 Geschwindigkeitsstufen.

### 2. Ein paar Grundlagen zur Motorsteuerung mit H-Brücke

Wir wissen natürlich, daß Motoren in der Regel mit Schaltern **ein- und ausgeschaltet**, ihre **Drehrichtung** ebenfalls mit Schaltern durch Umpolung ihrer Anschlüsse geändert und ihre **Drehzahl** mit Schaltern über das Verhältnis Aus/Ein oder Vorwärts/Rückwärts eingestellt werden. Dabei können diese Schalter mechanische Kontakte, Relais oder Transistoren sein. Auf dem Stand der Technik sind heute integrierte Schaltkreise, die die komplette Logik und die Leistungselemente für die Motor-Ansteuerung enthalten. Am motorseitigen Ende dieser IC's finden wir immer eine sogenannte H-Brücke, die die drei oben genannten Steuerfunktionen realisieren kann. Ihren Namen hat diese Schaltung von ihrer Form, die wie ein „H“ aussieht. Meist besteht diese Brücke aus Bipolartransistoren unterschiedlicher Leitfähigkeiten (pnp und npn), die häufig noch zur besseren Durchsteuerung als Darlington-Transistoren ausgelegt sind, oder aus Leistungs-Mosfets. Die schaltbaren Ströme sind von Typ zu Typ unterschiedlich, ebenso die äußeren Abmessungen und Ausstattungen (Kühlkörper, thermische Abschaltung, eingebaute Freilaufdioden usw.). Vernachlässigt man die Unterschiede, sieht eine solche H-Brücke schematisch so aus wie in Bild 1 gezeigt. Die rote Linie zeigt den Stromweg für die eine Drehrichtung, die blaue Linie den für die andere Drehrichtung. dazu passend die logischen Signale „0“ für einen gesperrten und „1“ für einen durchgeschalteten Transistor. Im gezeichneten Fall sind beide Transistoren vom npn-Typ. Die Betriebsspannung Vcc2 ist die Spannung für den Motor, die meist unabhängig von der Logik ist und auch höher gewählt werden kann als z.B. 5 V. Unser Schaltschema berücksichtigt nicht, daß die oberen und die unteren Transistoren vom Pegel her jeweils anders angesteuert werden müssen – ein richtig brauchbarer Motor-Ansteuer-IC erledigt das natürlich intern.



*Bild 1 Schema einer H-Brücke*

Wie man sieht, kann der Strom in beiden Richtungen durch den Motor fließen. Gefährlich wird es, wenn beide Transistoren eines senkrechten Zweiges gleichzeitig durchgeschaltet sind, so etwas muß die Logik verhindern. Üblicherweise werden die Eingänge in der Applikation oder intern mit Invertern zwangsweise richtig beschaltet.

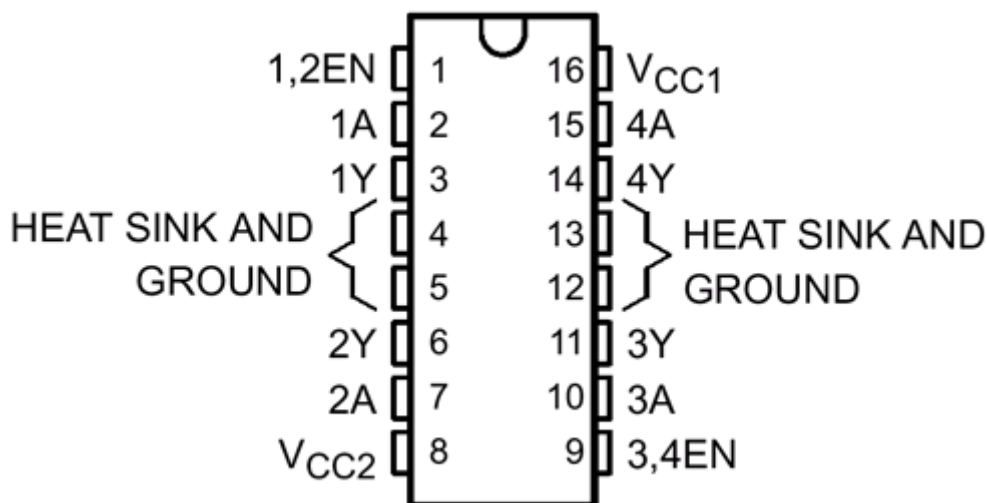
Es gibt eine Vielzahl unterschiedlicher H-Brücken-IC's, hier nur eine kleine Auswahl:

L293, L298, LMD182XX, ZHB67XX, L620X, BA6XXX, TLE5XXX, LB1836M, SN754410

Als Beispiel für die Funktionsweise solcher Bauteile wählen wir den SN754410 für ein paar nähere Informationen aus.

### 3. Der SN754410 von Texas Instruments

Der SN754410 ist zwar ein etwas älterer, von der Handhabung her aber für den Bastler recht gut geeigneter Baustein – er hat ein traditionelles 16-poliges Plastik-DIP-Gehäuse (Bild 2).



*Bild 2 Der Baustein SN754410 kann zwei DC-Motoren unabhängig ansteuern*

Ein Datenblatt ist unter

<http://www-s.ti.com/sc/psheets/slrs007b/slrs007b.pdf>

verfügbar.

Die mit „A“ bezeichneten Anschlüsse sind jeweils Eingänge, die mit „Y“ bezeichneten Anschlüsse sind die korrespondierenden Ausgänge. Für einen DC-Motor benötigt man zwei solche Ein- bzw. Ausgänge. Die mit „EN“ bezeichneten Eingänge können die dazugehörigen Ausgänge und damit den entsprechenden Motor sogar hochohmig abschalten. Eingangsseitig wird der SN754410 mit gängigen Logikpegeln versorgt (TTL oder CMOS). Vcc1 versorgt die interne Logik und kann zwischen 4,5 V und 5,5 V liegen. Vcc2 versorgt nur die Ausgangsstufen und natürlich die Motoren, diese Spannung kann zwischen 4,5 V und 36 V gewählt werden. Jeder Ausgang kann bis zu 1 A Strom für den Motor liefern.

Schaut man sich die Wahrheitstabelle (Tabelle 1) an erkennt man, daß der Ausgang (H = ggf. hohe Spannung, also kein TTL-Signal!) jedes Treibers direkt dem Eingangssignal (logisch H/L, also TTL oder CMOS!) folgt, vorausgesetzt der zugehörige ENABLE-Eingang liegt auf H. Liegt er auf L, ist der Ausgang hochohmig. Der EN-Eingang kontrolliert dabei immer zwei Treiber.

Inputs		Output
1A	1,2EN	1Y
H	H	H
L	H	L
X	L	Z

*Tabelle 1 Wahrheitstabelle des SN754410 (Beispiel Treiber 1)*

Soweit kein Problem, was aber macht man mit den Eingängen? Mit dem SN754410 ist zwar die Teilaufgabe „Leistungstreiber“ einschließlich Thermoschutz, Freilaufdioden usw. gelöst, nicht aber die Ansteuerung mehrerer Motoren mit PWM. Der SN754410 kann so lediglich als Motorschalter (Ein/Aus) und als Umpolschalter verwendet werden. Eine PWM-Steuerung direkt vom BASIC-Tiger<sup>®</sup> aus wäre zwar möglich, allerdings wäre dieser dann weitgehend nur mit diesem Problem beschäftigt – das ist nicht unsere Zielstellung. Hier gibt es eine kleine, aber feine Lösung, ein kompletter Motor-Controller mit serieller Ansteuerung für zwei Motoren.

#### **4. Dual Serial Motor Controller der Firma Pololu**

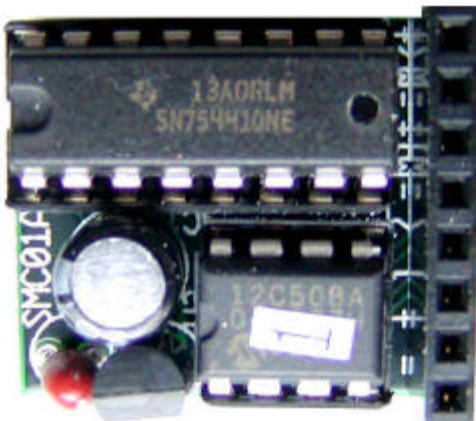
Die US-Firma Pololu bietet für Robotik und ähnliche Projekte kleine Module an, die jeweils 2 DC-Motoren unabhängig voneinander über eine serielle Schnittstelle ansteuern.

<http://www.pololu.com/products/pololu/0101/>  
<http://www.pololu.com/products/pololu/0401/>

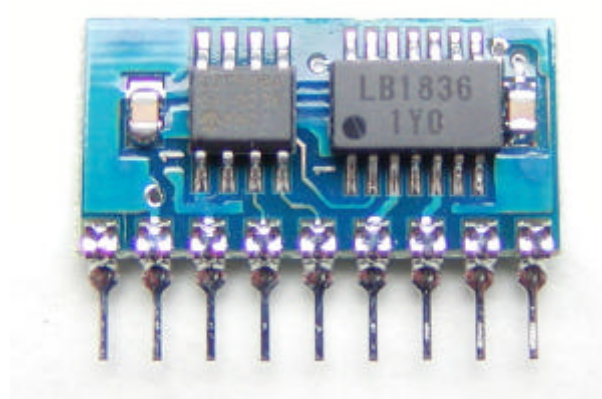
(Dual Serial Motor Controller)

(Micro Serial Motor Controller)

Das Herz dieser Module ist ein kleiner PIC-Controller, der seinerseits einen SN754410 (oder ähnlichen IC, den LB1836M) ansteuert. Mehrere solche Module sind (vom Hersteller entsprechend programmiert!) über eine einzige serielle Schnittstelle ansteuerbar – die Kaskade kann insgesamt bis zu 62 Motoren steuern. Während das erste Modul (Bild 3) als Bausatz für 20 \$ zu haben ist, kann man die zweite Variante als fertige „Mikro“-Ausführung (Bild 4) für 23 \$ bekommen. Beide Module erledigen per Software die Motorauswahl, die Drehzahlsteuerung über PWM und die Umschaltung der Drehrichtung vollkommen selbständig. Die eingestellten Parameter bleiben bis zum RESET oder einem neuen Steuerkommando erhalten, d.h. der Motor dreht sich solange in der eingestellten Weise, bis etwas neues angeordnet wird. Mit dieser feinen Idee ist die Beherrschung von Projekten mit mehreren Motoren über den BASIC-Tiger® ein Kinderspiel.



*Bild 3 Der Bausatz Pololu Dual Serial Motor Controller mit SN754410 (hier als Motor Controller #1 programmiert)*



*Bild 4 Die fertige Miniaturvariante, der Pololu Micro Dual Serial Motor Controller mit LB1836*

Schauen wir uns die Hard- und Software dieser Motor-Controller-Module etwas näher an.

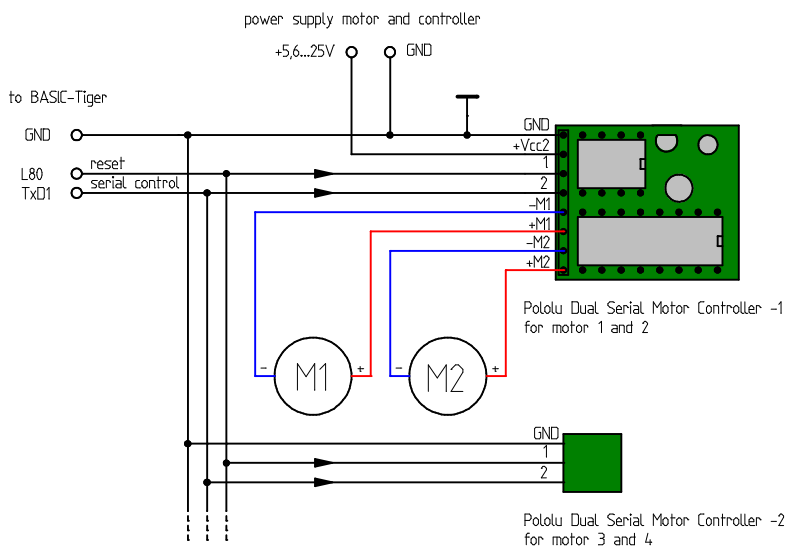
## **5. Die Zusammenschaltung BASIC-Tiger® / Motor-Controller**

Beide Module werden ähnlich angeschlossen, obwohl das eine 8 und das andere 9 Anschlüsse hat. Der Grund ist – das größere Modul besitzt einen eigenen 5 V – Regler, der die beiden IC's direkt aus der (hohen) Motorspannung mit Vcc1 versorgt, während das kleinere Modul jeweils ein eigenes Pin für die Motor- und Logikversorgung mitbringt. Das ist dann vorteilhaft, wenn die Motorspannung unabhängig von der Logikspannung zwischen 1,8 und 9 V sein und die separate 5 V – Spannung aus dem BASIC-Tiger®-System benutzt werden soll. Im Falle des großen Moduls muß standardmäßig die Versorgungsspannung über 5,6 V liegen (damit dessen Regler noch 5 V erzeugen kann), kann dann aber bis zu 25 V gehen (das ist gleichzeitig die Motorspannung).

Der Rest ist einfach. Beide Motoren gehen an die entsprechend bezeichneten Pins der Module, der logische Eingang „reset“ wird direkt an ein Tiger-I/O-Pin oder an 5 V gelegt und der Eingang „serial control input“ an einen seriellen Ausgang des BASIC-Tigers® (TxD0 oder besser TxD1). Da ein bidirektionaler Betrieb nicht erforderlich ist, lassen sich an diesen „Draht“ weitere Motor-Controller anschließen.

**Dabei ist unbedingt zu beachten, daß beide Typen von Pololu-Motor-Controllern am seriellen Eingang TTL-Pegel erwarten. Das ist bei vielen Tiger-Modulen kein Problem. Wenn Sie aber unbedingt BASIC-Tiger® mit eingebauten RS232-Pegelwandlern oder das Plug-and-Play-Lab benutzen wollen, ist ein weiterer externer RS232-Pegelwandler oder eine ähnliche Schaltung vorzusehen (z.B. die aus dem Datenblatt des Motor-Controllers)!**

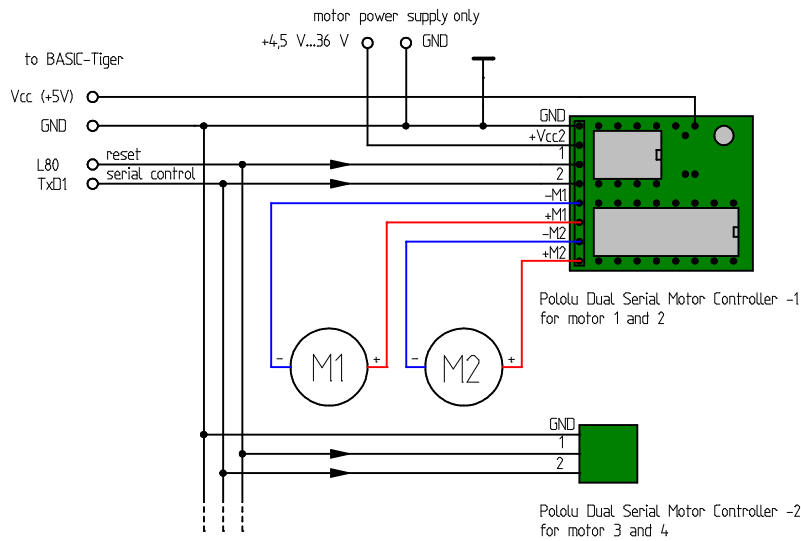
Die fertige Schaltung für die größeren Module sieht dann so aus (Bild 5)



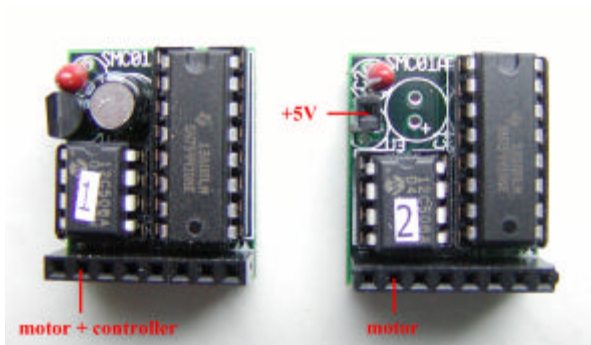
*Bild 5 Zusammenschaltung des originalen Dual Serial Motor Controllers von Pololu mit dem BASIC-Tiger®. Die Stromversorgung der weiteren Module sowie die Anschaltung der Motoren wurde nicht mehr gezeichnet!*

Will man die Vcc des BASIC-Tiger-Systems für den Motorcontroller (Bausatzvariante) nutzen und/oder sollen Motoren mit entweder kleinen oder sehr großen Nennspannungen verwendet werden, können wir ganz auf dessen Regler und seinen primärseitigen Elko verzichten. Damit heben wir die Verbindung beider Stromversorgungen auf. Am Ausgang des (nun nicht eingebauten) Reglers legen wir Vcc (+5V) aus dem BASIC-Tiger-Systems. Die Eingangsspannung Vcc2 ist nun lediglich die Motorspannung, die im Bereich von 4,5...36 V liegen darf. Bild 6 zeigt das Schema dieser Variante und Bild 7 die Ansicht der beiden Modulvarianten.

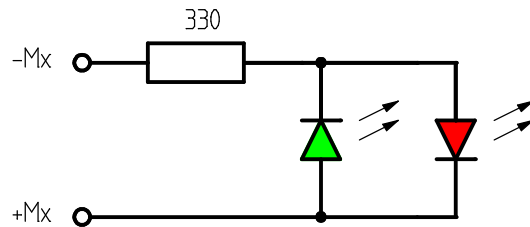
Der Autor hat in das „Ausgangsloch“ des vorgesehenen Spannungsreglers einfach eine zusätzliche Buchse eingelötet, so daß jetzt wieder 2 Spannungen (Vcc1 an diese Zusatzbuchse und Vcc2 an die frühere Kombinationsbuchse) angelegt werden müssen.



*Bild 6 Die unabhängige Stromversorgung der Motoren und die Mitversorgung der Controller aus der Vcc des BASIC-Tigers. Der Regler und sein primärseitiger Elko werden einfach weggelassen.*



*Bild 7 Beide Aufbau-Varianten, links das Original mit kombinierter, rechts mit getrennter Spannungsversorgung für Motoren und Controller*

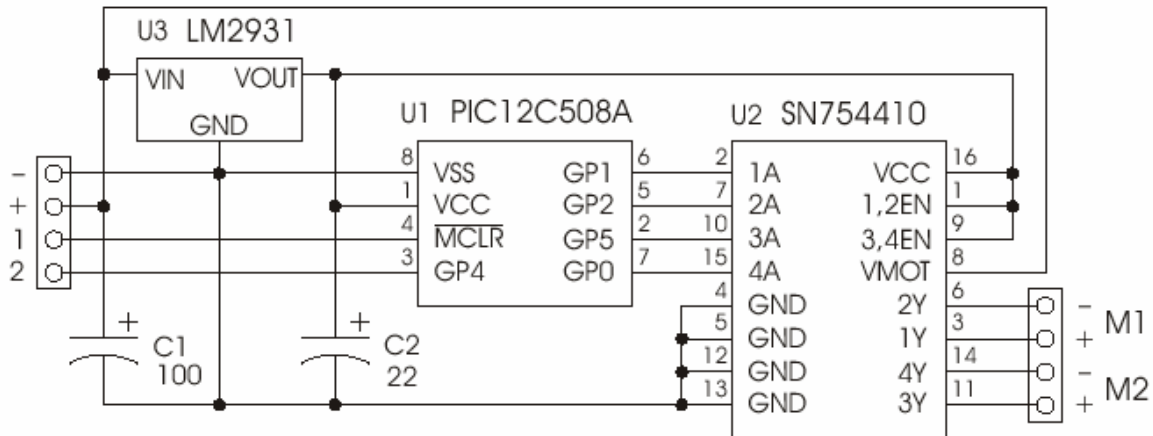


*Bild 8 Eine Indikatorschaltung, die sowohl die eingestellte Drehrichtung als auch die Drehzahl anzeigt*

Übrigens lassen sich mit den Motor-Controllern trotz ihrer Bezeichnung auch ganz andere Dinge steuern, z.B. LED's oder Lampen. So kann man Lichtquellen in Modellen, aber auch anderen Anwendungen auf einfache Weise dimmen. Eine Schaltung nach Bild 8 mit zwei antiparallelen LED's unterschiedlicher Farbe kann man als Indikator für einen Meßwert (negativ oder positiv) nutzen.

Für diejenigen, die tiefer in die Geheimnisse des Pololu Dual Serial Motor Controller mit SN754410 einsteigen oder weiter modifizieren wollen, zeigt Bild 9 die Innenschaltung.





*Bild 9 Die Innenschaltung des Pololu Dual Serial Motor Controller mit SN754410 (Abbildung aus der Aufbau- und Bedienungsanleitung von Pololu)*

## 6. Nun geht's rund – Die Software-Seite der Motorsteuerung

Die serielle Steuerung erfolgt asynchron nach dem RS232-Protokoll (8 Bits, no Parity, 1 Stopbit mit 1200 bis 19200 Baud). Die Baudrate wird durch das Modul automatisch erkannt. In welcher Weise werden nun Daten vom BASIC-Tiger<sup>®</sup> an das Modul geschickt – wir erinnern uns, daß das nur in einer Richtung geht!

Grundsätzlich bestehen alle Befehle für den Motor-Controller aus 4 Bytes mit folgender Struktur:

start byte	device type	motor # and direction	Motor speed
80h	00h	0 <span style="background-color: cyan;">x x x x x x x</span> <span style="background-color: magenta;">x</span> 0	<span style="background-color: green;">x x x x x x x</span>
		motor #	speed
		direction	

*Tabelle 2 Struktur der Steuerkommandos an den Motor-Controller*

Nur das erste Start-Byte „80h“ oder dezimal 128 enthält eine „1“ an der höchsten Bit-Position. Das zweite Byte mit der Bezeichnung „device type“ bedeutet zunächst mal für uns gar nichts, hier können lediglich Motor-Controller von andere Baugruppen unterschieden werden. Wichtiger ist das dritte Byte, das im höchsten Bit immer „0“, in den folgenden 6 Bits die Motor-Nummer und im niederwertigsten Bit die Drehrichtung enthält. Schließlich liegt im 4. Byte in den 7 niederwertigen Bits die Geschwindigkeitsinformation. 00h ... 7Fh, wobei 00h Stillstand des Motors bedeutet. Das Bit 7 dieses 4 Bytes ist ebenfalls immer „0“.

Ist die Bildung der Parameter „direction“ (logisch 0 oder 1) und „speed“ (00h...7Fh oder 0...127 dezimal) noch unkompliziert, muß bei der Motornummer etwas nachgedacht werden.

Die Controller müssen, wie schon erwähnt, vom Hersteller programmiert werden. In den Bestellunterlagen von Pololu findet man da folgende Angaben:

*Dual serial motor controller-1*  
*Responds to motor numbers 0, 1, 2, and 3.*

*Dual serial motor controller-2*  
*Responds to motor numbers 0, 1, 4, and 5.*

*Dual serial motor controller-3*  
*Responds to motor numbers 0, 1, 6, and 7.*

Das ist ein wenig umständlich ausgedrückt, besagt aber einfach folgendes:

- Wird nur ein Controller (für 1 oder 2 Motoren) gebraucht, kann man bei allen Controller-Typen **immer** die Motor-Nummer 0 oder 1 auswählen.
- Werden mehrere Controller an eine serielle Leitung angeschlossen (einfach parallel schalten!), kommen die zweiten Nummern-Paare ins Spiel. Dann kann man mit dem ersten Typ (*Dual serial motor controller-1*) die Motor-Nummern 2 und 3, mit dem zweiten Typ (*Dual serial motor controller-2*) die Motor-Nummern 4 und 5 ansteuern usw. Die Motor-Nummern 0 und 1 werden in diesem Fall nicht mehr benutzt. Braucht man mehr als 6 Motoren, ist eine spezielle Bestellung bei Pololu fällig, da die höheren Nummern nicht auf Vorrat gehalten werden. Die Bestell-Nummern ergeben sich dann aus der Fortführung der oben stehenden Aufstellung. Für die Programmier-Praxis kann man sich merken, daß die Motornummer in der Programmierung immer um 1 erhöht werden muß, der erste Motor bekommt also die Nummer 2, der zweite die Nummer 3, der dritte die Nummer 4 usw.

In unseren Beispielen mit dem BASIC-Tiger® werden bis zu 6 Motoren gleichzeitig verwendet, damit benötigt man drei Controller und zwar die *Dual serial motor controller-1, -2, -3*. Alles klar?

Zwei Beispiele:

Wir haben insgesamt 6 Motoren, nutzen also das erweiterte Motor-Nummern-System und wollen den vierten Motor mit Maximaldrehzahl vorwärts laufen lassen. Wir erinnern uns, daß der erste Controller die Motor-Nummern 2 und 3, der zweite die Nummern 4 und 5 verwaltet usw. Dann hat der vierte Motor in Folge die Nummer 5, so daß das Steuerwort wie folgt aussieht:

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	
80h								00h								0Bh								7Fh							

Wieder 6 Motoren insgesamt, der sechste Motor (also die Nummer 7) soll ganz langsam rückwärts laufen:



---

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1		
80h								00h								0Eh								01h							

Noch ein Wort zum „RESET“ - Pin. Im Normalbetrieb sollte es auf logisch „1“ oder an der Logik-Vcc1 liegen. Legt man das „RESET“ – Pin für mindestens 2 µs auf „0“, werden alle Motoren des betreffenden Motor-Controllers abgeschaltet und der Controller wartet auf das erste serielle Kommando vom BASIC-Tiger®. Sie müssen „RESET“ nicht benutzen, dann sollte der Eingang direkt an die Logik-Vcc1 gelegt werden. Um jedoch ordentliche Startbedingungen zu schaffen, ist beim Programmstart ein RESET immer angebracht.

Das Programm MULTIMOT.TIG steuert insgesamt 6 Motoren nacheinander vom Stillstand schneller werdend in einer Richtung bis zur Maximaldrehzahl und wieder zurück. Nach kurzer Pause wiederholt sich das in der anderen Drehrichtung. Danach kommt der zweite Motor dran usw. Voraussetzung ist, daß drei Controller (Nummern 1, 2, 3!) und ihre sechs Motoren an die seriellen Leitung TxD1 (TTL-Pegel!) angeschlossen werden. Was die einzelnen Motoren tun, wird auf dem Display des BASIC-Tigers® angezeigt, wenn Sie eines angeschlossen haben.

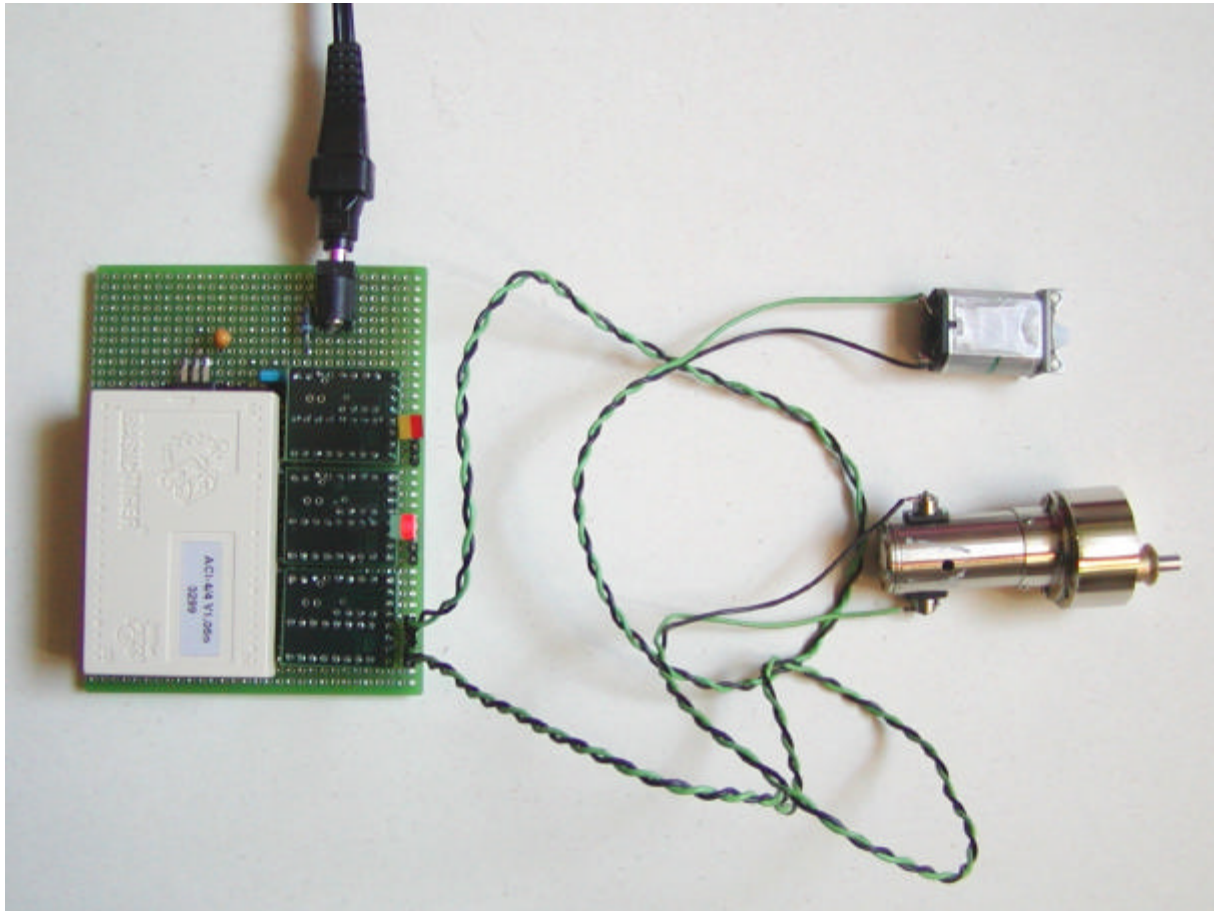
## 7. Neue Möglichkeiten

Durch die einfache, aber komfortable Steuerung vieler Motoren über eine einzige serielle Verbindung sowie deren Betrieb in nur einer Richtung eröffnen sich weitere Möglichkeiten. So kann man diese seriellen Kommandos ganz bequem über Funk oder IrDA verschicken, also drahtlos. Jeder Modellbauer wird sich darüber freuen. Zwar sind moderne Fernsteueranlagen auch ganz schön leistungsfähig, 62 Motoren (oder andere Funktionen, wie, Lampen, LED's usw.) sind aber eher ein Traum. Unsere Lösung kann noch mehr, diese Motoren dürfen auch dezentral betrieben werden, d.h. ein BASIC-Tiger® (z.B. als Fernsteuerung oder Fernbedienung) steuert den Sender und jeweils ein Empfänger steuert eine Gruppe von Motoren. Ideal für Systeme im Haus (Jalousien) oder im Hobbybereich (Fahrzeuge, Roboter).

Doch nicht nur die einfache Einstellung von Drehrichtung und Geschwindigkeit ist interessant, mit der Kombination BASIC-Tiger / Motor-Controller können Modellfahrzeuge z.B. automatisch vorbildgerecht beschleunigen und abbremsen. Genauso verlockend sind Anwendungen, bei denen auf Knopfdruck ein Motor langsam anläuft und in einer vorbestimmten Weise beschleunigt bis zur Maximaldrehzahl. Lassen Sie den Knopf los, beginnt das Spiel von neuem – z.B. rückwärts. Damit können Sie elegant Justierungen durchführen, bei denen es sowohl auf feinfühligere Bewegung (am Anfang) als auch um schnelle Veränderungen über den ganzen Bereich (nach längerem Drücken) ankommt.

Benutzen Sie Encoder oder andere Rückmelde-Sensoren, die Position oder Geschwindigkeit Ihres Antriebes signalisieren! Da der BASIC-Tiger mit der Motorsteuerung kaum belastet ist, sind genug Rechen- und I/O-Recourcen für andere Dinge frei.

Bild 10 zeigt einen Laboraufbau des Autors mit 3 Motor-Controllern für insgesamt 6 Motoren.



*Bild 10 Laboraufbau mit 3 Motor-Controllern. Angeschlossen sind hier zwei unterschiedliche Motoren sowie zwei LED-Kombinationen nach Bild 9*

Noch ein Hinweis:

Für Mehrmotorenprojekte ist die Benutzung eines eigenen Netzteiles für die Motoren in jedem Fall sinnvoll.

Viel Spaß mit der nun wirklich universellen Motorsteuerung!