
Ansteuerung von Servo-Motoren

Gunther Zielosko

1. Grundlagen

Um mit einem Mikrorechnersystem wie dem BASIC-Tiger® auch mechanische Systeme steuern zu können, kann man auf mehrere Konzepte zurückgreifen.

- Antriebe mit Schrittmotoren
- Antriebe mit Gleichstrommotoren
- Antriebe mit Servomotoren

Schrittmotoren werden mit speziellen Signalsequenzen angesteuert, die an separate Spulen dieser Motoren gelegt werden. Damit bewegen sich deren Rotoren immer genau einen "Schritt" weiter, meistens wenige Umdrehungsgrade. Über eine komplizierte Ansteuerung und entsprechende Getriebe kann das System dann bewegt werden.

Gleichstrommotoren können mit Mikrorechnern über entsprechende Verstärkerstufen einmal aus- und eingeschaltet werden, zum anderen kann man, z.B. über Puls-Code-Modulation Drehzahlen einstellen.

Eine ganz andere Technik stellen die sogenannten Servo-Antriebe dar, die unter anderem in der Modellbauwelt (Flugzeug-, Auto- und Schiffsmodelle) verwendet werden. Diese sind bereits für wenige DM zu haben und haben einige bestechende Vorteile für den Hobby-Elektroniker. So arbeiten sie mit niedrigen Spannungen (5 V sind genug), zum anderen verbrauchen sie im Ruhezustand wenig Strom. Ihrem Ursprungszweck entsprechend erzeugen sie eine Drehbewegung von etwa 90-180 Grad an einer stark untersetzten, d.h. sehr kräftigen Achse. Für Bewegungen, die mit solchen Werten auskommen (z.B. Solarelemente der Sonne nachführen, Spiegel kippen, Modelle lenken usw.) sind Servomotoren ideal. Als Beispiel hier ein einfaches Standardservo der Firma Graupner:



Die wichtigsten Bestandteile sind ein kleiner Gleichstrom-Motor mit Getriebe, an dessen Ausgang mechanisch ein Potentiometer gekoppelt ist sowie eine Elektronik, die ein pulslängenmoduliertes Signal auswertet und den Motor entsprechend ansteuert. Im Modellbau gibt es eine standardisierte Ansteuerung dieser Servos. Im Abstand von 20 ms werden Impulse gesendet (bei Funkfernsteuerungen über den Sender), deren Länge von 1,1 ms bis 2,1 ms variiert und damit die jeweilige Stellung des Servos angibt (Linksanschlag, Rechtsanschlag). Diese Eingangsimpulslänge wird mit der Impulslänge eines monostabilen Multivibrators verglichen, das von der Stellung des Potentiometers und damit von der Position des Servos abhängt. Erst wenn die Position des Servos die gleiche interne Impulslänge erzeugt, wie die äußerlich angelegte, hört die Bewegung auf, das Servo hat seine vorgegebene Position erreicht. Alles, was der Anwender eines handelsüblichen Servos noch tun muß, ist die Masse und 5 V anschließen und am Eingang für eine Impulsfolge sorgen, die es "versteht". Üblich sind hier normale TTL-Pegel.

Dies könnte ein Mikrorechner wie der BASIC-Tiger[®] in einfacher Weise tun. Naheliegender wäre hier z.B. ein einfaches BASIC-Programm, das die geforderten Impulse liefert. Leider ist es nicht so einfach, die Impulse dauernd bereitzustellen, ohne den Rechner ständig zu beschäftigen. Er muß ja dauernd die Zeiten für Low und High berechnen und dies ggf. für mehrere Motoren.

Deshalb wurde hier ein anderes Konzept gewählt. Bei handelsüblichen Fernsteuerungen werden zur Impulserzeugung ebenfalls monostabile Multivibratoren benutzt, deren Impulslänge mit einem Potentiometer eingestellt wird (diese sind an die Steuerkuppel der Fernsteuersender angeschlossen). Wir benutzen ebenfalls ein Potentiometer zur Einstellung der Impulslänge, allerdings ein elektronisches. Es gibt einen solchen Baustein von DALLAS Semiconductor, den DS1267, der über ein serielles Interface zwei unabhängige Potentiometer steuert. Ein Datenblatt „aus dem Internet“ bekommt man über:

Datenblatt DS1267: <http://www.dalsemi.com/DocControl/PDFs/1267.pdf>

oder, u.U. auch mit einem kostenlosen Muster bei:

atlantik elektronik GmbH
Fraunhoferstr. 11a
82152 Planegg
Tel.: 089-89505-0

Hier kann man den DS1267 auch beziehen. Den Baustein gibt es in verschiedenen Widerstandsbereichen, für die hier vorgestellte Anwendung ist die 100 k Ω -Version vorgesehen.

Ein besonderer Glücksumstand ist bei diesem IC die serielle Ansteuerung. Günstig ist hierbei, daß ein einmal eingestellter Wert solange im IC bleibt, bis er wieder geändert wird.

2. Das Schaltungskonzept

Da uns der Baustein DS1267 schon einen großen Teil der Arbeit abnimmt, ist der restliche Teil der Schaltung einfach. Es wird ein Taktgenerator für 50 Hz benötigt (hier ein Timer NE555, dieser Takt kann aber auch einfach aus dem Netz generiert werden), jeweils ein monostabiler Multivibrator und eben das elektronische Potentiometer. Beide IC's beinhalten zwei gleiche Komponenten, so daß man ganz einfach zwei Servo-Ansteuerungen erhält. Da der DS1267 auch noch kaskadierbar ist, kann eine nahezu unbegrenzte Anzahl von Servos bedient werden.

Hier noch die Internetadresse für Datenblätter:

Datenblatt NE555: <http://www-s.ti.com/sc/psheets/slfs022/slfs022.pdf>

Datenblatt CD(1)4538: <http://www-s.ti.com/sc/psheets/schs093/schs093.pdf>

Der monostabile Multivibrator CD4538 wird bei einer L/H-Flanke an den Triggereingängen +TR1 bzw. +TR2 (Pins 4 und 12) gesetzt, nach Ablauf der von den RC-Gliedern vorgegebenen Zeit setzt es sich wieder zurück. Die RC-Glieder bestehen aus den Kondensatoren 10 nF und aus jeweils drei aus Einzelwiderständen zusammengesetzten Widerständen (10 k Ω Trimpotentiometer, 91 k Ω Festwiderstand und maximal 100 k Ω des variablen elektronischen Potentiometers). Das eigentliche elektronische Potentiometer ist der Widerstand von 100 k Ω zwischen den Hx- und Lx-Anschlüssen des DS1267 mit dem Abgriff Wx. Die High-Seiten (Hx) werden hier auf Vcc gelegt, die Low-Seiten bleiben offen. Der Widerstandswert wird an den Abgriffen Wx abgenommen. Der Gesamtwiderstand liegt somit zwischen Vcc und den Eingängen des CD4538, wie dies zu dessen ordnungsgemäßer Funktion erforderlich ist. Die oben genannten Werte garantieren, daß bei der vorgegebenen Widerstands-Variation des DS1267 der komplette Bereich des Servos angesteuert werden kann. Der Feinabgleich erfolgt dann mit den Trimpotentiometern so, daß beim Steuerwert 0 die eine Endlage, beim Steuerwert 255 die andere Endlage sicher erreicht wird.

Die externen Komponenten für den Timer müssen ggf. auf 50 Hz abgeglichen werden. Die angegebenen Werte für R und C stellen nur Richtwerte dar.

Noch ein Wort zur Stromversorgung. Die Servos und ihre Motoren sind zwar klein, benötigen aber doch zumindest zeitweise einen erheblichen Strom. Deshalb und wegen der unvermeidlichen Störungen ist es sinnvoll, sie aus einer eigenen 5 V - Quelle zu versorgen. Geeignet sind z.B. NC-Akkus Mignon mit 4 Zellen (4,8 V). Wird ein Netzteil benutzt, kann auch der 50 Hz-Takt dort abgeleitet werden, das spart den Schaltungsteil mit dem Timer NE555. Eine Begrenzung der Amplitude des Taktes sowie eine Begrenzung der Stromversorgung auf 5 V ist unbedingt notwendig.

3. Software

Das Programm Servo01.TIG kann sowohl auf dem Plug-and-Play-Lab als auch in separaten Anwendungen genutzt werden. Es zeigt den Betrieb von zwei Servos mit einfachen Bewegungsabläufen. Der Phantasie des Anwenders sind keine Grenzen gesetzt. So kann ein Meßwert mit einem großen Zeiger angezeigt, eine Videokamera ausgerichtet, ein Modellfahrzeug gesteuert oder ein Roboter auf die Reise geschickt werden. Der eigentliche Ablauf der Bewegung wird im Unterprogramm „servo“ vorgenommen. Das Unterprogramm „senden“ wurde aus dem Buch „Messen, Steuern und Regeln mit dem BASIC-TIGER®-SYSTEM“ von Bernd vom Berg und P. Groppe aus dem Franzis‘ Verlag entnommen und entsprechend angepaßt. Dort kann man auch weitere interessante Informationen zur Ansteuerung des Bausteines DS1267 erhalten, die den Rahmen dieses Applikationsberichtes sprengen würden.

```
'-----
' Programm SERVO01.TIG
'   Das Programm demonstriert den Betrieb von zwei Modell-Servos.
'   Voraussetzung ist eine entsprechende Hardware mit einem Taktgenerator
'   50 Hz (NE555) und einem Dual-Monoflop (CD4538), dessen Haltezeit mit
'   dem digitalen Potentiometer DS1267 vom BASIC-Tiger eingestellt werden
'   kann. Hier werden die Servos nur automatisch langsam in die eine
'   Richtung und schnell in die andere Richtung gestellt. Beliebige
'   Bewegungssteuerungen lassen sich im Unterprogramm servo realisieren.
'   Zum Betrieb des Programms ist nur die spezielle Hardware erforder-
'   lich, alle anderen Komponenten des Plug-and-Play-Labs werden nicht
'   eingesetzt, um die Uebersichtlichkeit nicht zu stoeren. Damit koennen
'   auch stand-alone-Loesungen einfach realisiert werden.
'-----

user_var_strict

byte servo0,servol,n           'die Werte der Servos, Zaehlvariable
long datenwort                 'das komplette Steuerwort fuer DS1267

'-----
' Hauptprogramm
'-----

task main

    dir_pin 8,0,0               'Port 8.0 Ausgang (DQ vom DS1267)
    dir_pin 8,1,0               'Port 8.1 Ausgang (CLK vom DS1267)
    dir_pin 8,2,0               'Port 8.2 Ausgang (RST\ vom DS1267)

Neu:

    call servo                   'Einstellwerte fuer Servos abholen
    goto Neu                     'neu anfangen

end                             'Ende des Hauptprogramms

'-----

' Unterprogramm Servo-Wert einstellen
'   dieses Unterprogramm ist ein Beispiel zum automatischen
'   Betrieb der beiden Servos. Hier koennen beliebige Daten
'   an die Variablen servo0 und servol uebergeben werden.
'-----

sub servo
servo0 = 128                     'Servo0 auf Mittelstellung
servol = 128                     'Servol auf Mittelstellung
call senden                     'Wert fuer Servos an Unterprogramm Senden
WAIT_DURATION 1000              'Nach Einstellung s warten

FOR n = 0 TO 255                 'Zaehlschleife
    servo0 = n                   'Variable Servo0 setzen
    servol = 255 - n             'Variable Servol setzen
    WAIT_DURATION 10             '10 ms warten
    call senden                  'Wert fuer Servos an Unterprogramm Senden
Next                             'naechster Wert
```

```
    WAIT_DURATION 1000                'Wenn Endstellung, 1s warten
end                                    'Ende Unterprogramm
'-----
'
'-----
' Unterprogramm Daten an DS1267 senden
'-----
sub senden
    out 8,00000111b,00000100b        'Start der Dateneübertragung
                                    'RST\ auf HIGH,
                                    'DQ u. CLK auf LOW setzen
    datenwort=((BIT_MIRR(serv0,8))*512)+((BIT_MIRR(serv1,8))*2)
    '17-Bit-Datenwort für DS1267 zusammenstellen:
    'Bit1=SSB (erstes zu übertragendes Bit, hier immer 0),
    'Bit2-Bit9 Wert für serv1, jedoch gespiegelt (Bit2=MSB),
    'Bit10-Bit17 Wert für serv0, jedoch gespiegelt (Bit10=MSB)
    shift_out 8,0,1,datenwort,17     'Ausgabe des Datenwortes über:
                                    'Ausgabeport: 8, mit Datenpin: 8.0
                                    'und Taktpin: 8.1, 17 Bits
    out 8,00000111b,00000000b        'Ende der Dateneübertragung
                                    'RST\, DQ u. CLK auf LOW setzen
end
'-----
```

