

Programmieren mit der BASIC-stamp 2

Teil 4: Hindernislauf

Als Teilnehmer im Straßenverkehr wissen Sie, daß zwei Objekte niemals zur gleichen Zeit den gleichen Raum einnehmen sollten. Was für den Straßenverkehr richtig ist, gilt um so mehr für Roboter. Ein autonomer mobiler Roboter muß sich selbst vor Kollisionen mit Hindernissen schützen, ob es sich nun um feststehende wie eine Mauer oder einen Pfahl oder um mobile wie einen Hund, einen Menschen oder einen anderen Roboter handelt.

Glücklicherweise benötigen Roboter kein aufwendiges optisches System, um Hindernisse zu erfassen, sie kommen mit viel einfacheren Mitteln aus. Manche Roboter nutzen Radar, Sonar oder Sodar (wenn sich der Roboter an Land und nicht unter Wasser bewegt). Eine noch einfachere Variante stellt Infrarot-Licht dar, das den Weg ausleuchtet, von einem Hindernis reflektiert und vom Roboter erkannt und ausgewertet wird.

Wir wollen in diesem vierten Teil der Artikelserie zeigen, wie man den BoE-Bot mit Infrarot-Sensoren ausstattet und wie der Roboter deren Informationen nutzt, um Hindernissen aus dem Weg zu gehen. In diesem Zusammenhang wird auch erläutert, wie die BASIC-stamp mit einfachen mathematischen Mitteln Zufallszahlen erzeugen kann. Die für die Anwendung in **Bild 1** nötigen Teile sind in der Stückliste aufgeführt.

IR-GRUNDLAGEN

In der Theorie scheint es sehr einfach, ein Objekt mit infrarotem Licht zu erfassen. Eine Infrarot-Quelle (eine IR-LED) leuchtet in die Bewegungsrichtung des Roboters, ein in gleicher Richtung ausgerichteter Detektor erkennt von einem Hindernis reflektiertes IR-Licht. In der Praxis sieht die Angelegenheit ein wenig komplizierter aus. Bei dieser vereinfachten Vorgehensweise würde der Detektor stark vom "natürlichen" infraroten Umgebungslicht abgelenkt.

Um den Roboter vor solchen falschen Signalen zu schützen, werden Detektoren eingesetzt, die nicht auf einfaches, sondern nur auf mit einer bestimmten Frequenz moduliertes IR-Licht reagieren. Die Quelle muß dazu das IR-Licht in genau dieser Frequenz modulieren. Bei vielen AV-Geräten kommen Fernbedienungen mit einer

Modulationsfrequenz von 38 kHz zum Einsatz, so daß geeignete IR-Empfänger gut erhältlich und billig sind.

Die BASIC-stamp ist leider nicht multitasking-fähig, kann zwar die IR-Quelle moduliert ansteuern, aber nicht gleichzeitig das Signal des Detektors empfangen. Deshalb ist eine externe Modulationsschaltung für die IR-LEDs notwendig.

Eine gute Idee ist es, nicht nur eine, sondern zwei LED/Detektor-Kombinationen anzubringen, und zwar eine an der linken und die andere an der rechten Seite des Chassis. Dies erlaubt es, die Position eines Hindernisses relativ zum Roboter zu erfassen. Wenn nur ein Detektor reagiert, so befindet sich das Hindernis auf der entsprechenden Seite, liefern beide Detektoren ein Signal, so liegt es genau in Fahrtrichtung.

DER OSZILLATOR

IR-Leuchtdioden sind Allerwelts-Bauteile und unterscheiden sich weder in Aussehen noch in der Behandlung von normalen LEDs. Um ihr Licht zu modulieren, kann ein Oszillator mit einem 555-Timer verwendet werden. Der Timer steuert die LEDs an, wenn sein Reset-Eingang High (nicht aktiv) ist. Darauf nimmt Port 5 der BASIC-stamp Einfluß. Das Ausgangssignal des Timers wird außer zur LED auch zu Port 6 geleitet. Dies ist für den normalen Betrieb nicht nötig, sondern gestattet es, den Oszillator hinreichend genau auf die Frequenz von 38 kHz einzustellen.

Um den Timer zu kalibrieren, lädt man das Programm in **Listing 1** und startet es. Die Frequenz wird auf dem PC-Monitor in Hertz angezeigt. Am Poti läßt sich die Frequenz auf den gewünschten Wert 38.000 einstellen. Der 555 ist zwar nicht besonders stabil, so daß der Wert im Betrieb leicht abweichen kann, man sollte sich aber den-

noch die Mühe geben, die 38 kHz unter den üblichen Betriebsbedingungen möglichst genau zu treffen.

Ob eine IR-LED funktioniert, ist natürlich nicht so einfach zu erkennen wie bei einer "normalen" LED. Mißt man die Schwellenspannung der (arbeitenden) LED, sollte das Voltmeter etwa 1,2 V anzeigen. Liegt die Spannung dagegen bei 5 V, könnte die IR-LED verpolt angeschlossen sein (die Anode mit langem Anschlußdraht gehört direkt an die BASIC-stamp).

Alternativ läßt sich eine Plastik-Karte im IR-empfindlicher Beschichtung verwenden, wie sie von Service-Technikern zur Überprüfung von Fernbedienungen gerne eingesetzt wird und bisweilen im Fernsehfachhandel erhältlich ist. Eine elektronische Variante stellt der IR-Fernbedienungstester dar, der im Elektor-Halbleiterheft 1998 beschrieben wurde.

DER IR-DETEKTOR

Sind die IR-LEDs einsatzbereit, werden die IR-Detektoren wie in **Bild 2** zu sehen angeschlossen. Die Detektoren gleichen gewöhnlichen Transistoren mit einer Beule an der Seite. Diese Beule ist die IR-sensitive Zone. Trotz ihrer Ähnlichkeit mit Transistoren beherbergen die Bauteile eine komplette integrierte Schaltung mit Demodulator, Filter, Verstärker und Schaltstufe. Der Ausgang führt eine logische null, wenn der Detektor ein IR-Signal erfaßt. Über die beiden anderen Anschlüsse erhält der Detektor seine Betriebsspannung. Neben dem in der Stückliste genannten IR-Detektor kann problemlos auch ein SFH505/6 von Siemens oder ein Exemplar der ISU-Serie von Sharp eingesetzt werden.

Die optimale Montage der LEDs ist ein wenig knifflig und von der genauen Konstruktion des Roboterchassis abhängig. Wichtig ist, daß die LED genau nach vorne und möglichst wenig zur Seite leuchtet. Es gibt zwar IR-LEDs mit definiert schmaler Abstrahlcharakteristik, wenn Sie nicht darüber Bescheid wissen, reicht es aus, einen wenige Millimeter langen Schrumpfschlauch über die LED zu schieben, um die Abstrahlung zur Seite zu unterbinden.

Ähnliches gilt auch für den Detektor, aber hier funktioniert mangels Befestigungsmöglichkeit der Trick mit dem Schrumpfschlauch nicht. Deshalb sollte der Detektor so weit wie möglich von der LED entfernt befestigt und wie in **Bild 2** zu sehen leicht gegen die LED-Richtung verdreht werden. Um die optimale Positionierung von LED und Detektor herauszufinden, nutzt man das Programm in **Listing 2**. Wenn sich kein Hindernis vor dem BoE-Bot befindet und die LEDs nicht seitlich abstrahlen, zeigt sich auf dem Monitor

Stückliste

- 1 einsatzbereiter BoE-Bot
- 1 Kondensator 3300 µF/16 V
- 1 Poti 500 Ω
- 1 Kondensator 10 nF
- 2 Kondensatoren 100 nF
- 2 Widerstände 1 kΩ
- 2 Widerstände 470 Ω
- 1 Widerstand 470 Ω
- 1 Timer-IC 555
- 2 Infrarot-Empfänger (z.B. Panasonic 4602)
- 2 IR-LEDs (z.B. LD271)

weder die Meldung "left" noch "right". Sind auch die Detektoren optimal eingestellt, so kann man den BoE-Bot in Bewegung setzen. Dazu dient das Programm in **Listing 3**, das anhand der Sensor-Signale den vom PC-Anschluß befreiten Roboter steuert. Entdeckt der Roboter mit dem linken oder rechten Sensor (oder beiden) ein Hindernis auf seinem Weg, wendet er, um einen Zusammenstoß zu vermeiden und fährt in die entgegengesetzte Richtung weiter.

ETWAS KOMPLIZIERTER

Auf diese Art und Weise würde der BoE-Bot sich immer nur auf gerader Linie zwischen zwei Hindernissen bewegen. Ein etwas komplexeres Steuerprogramm bietet **Listing 4**. Wie bei der einfachen Steuerung bewegt sich der Roboter geradeaus, bis ein Hindernis auf seinem Weg erscheint. Dann aber: Liegt das Hindernis auf der rechten Seite, dreht der BoE-Bot nach links ab, befindet es sich dagegen links, fährt der Roboter nach rechts weiter. Wenn beide Detektoren das Hindernis nahezu gleichzeitig erfassen, das Hindernis sich also genau in Fahrtrichtung befindet, bewegt sich der Roboter zurück und dreht dann nach links oder rechts ab.

Das Programm verwendet an mehreren Stellen Zufallszahlen. Der Zufall entscheidet einerseits, in welche Richtung der BoE-Bot bei einem frontalen Hindernis abdreht, zum anderen, mit welcher Geschwindigkeit es nach einem Richtungswechsel weitergeht.

EIN ZUFÄLLIGER BEFEHL

Die BASIC-stamp kennt den Befehl RANDOM, um auf den ersten Blick zufällige Zahlen zu erzeugen. In Wirklichkeit verknüpft der Befehl jeden Wert einer Variablen mit einer festgelegten Zahl. Gleiche Variablenwerte haben stets gleiche Zufallszahlen zur Folge. Deshalb spricht man von einer Pseudo-Zufallszahl.

In unserem Fall schadet das aber nichts. Ist die erzeugte Zahl auch nicht

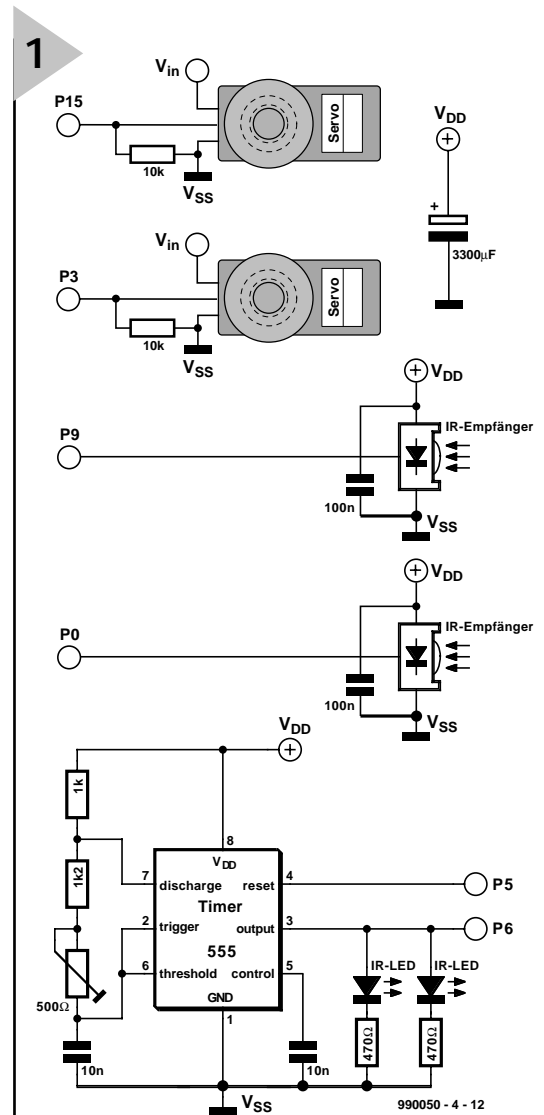


Bild 1. Die komplette Schaltung

zufällig, so doch wenigstens unvorhersehbar. Das Programm addiert zur Random-Variablen jedesmal eine eins, wenn die Hauptschleife *sense* durchlaufen wird. Die unvorhersehbare Zahl wird nach *back*: übernommen, in der über die zukünftige Richtung entschieden wird.

Da die Zufallszahl den gesamten möglichen Bereich der Zahlen umfaßt, sollte die RANDOM- immer eine Word-Variable sein. Der Wertebereich kann anschließend mit den Operatoren AND (&) oder mod (//) eingeschränkt werden. Um zum Beispiel den Wert auf 15 (%1111 in binär) zu beschränken, enthält das Programm die Zeilen

```
RANDOM rnd
X=rnd & %1111
```

Gebräuchlicher ist aber die Beschränkung des Wertes mit dem mod-Operator (//). Der Operator gibt den Rest

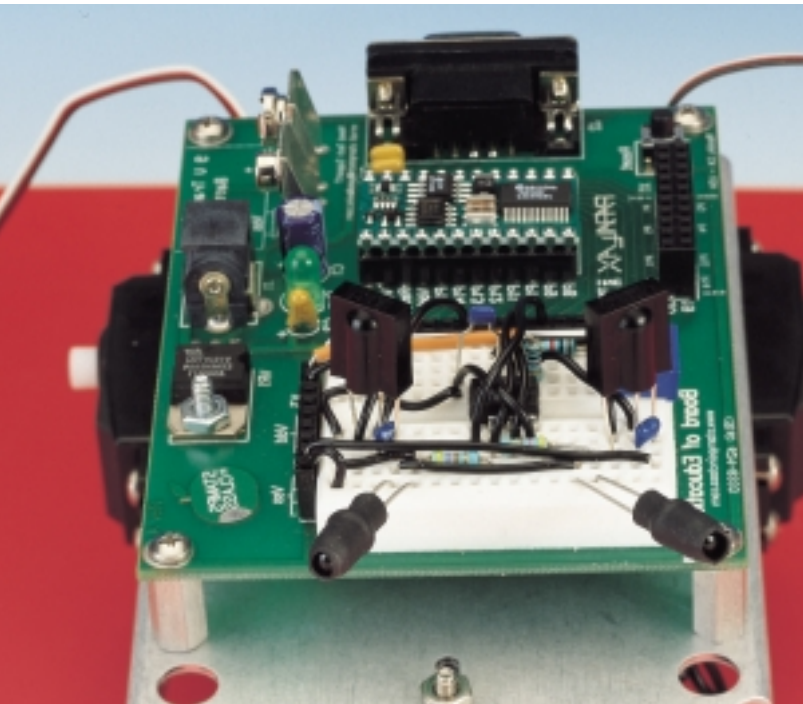


Bild 2. Vorschlag zur LED- und Sensor-Positionierung

einer Division von Integer-Zahlen an. So ergibt die Division $11/3$ einen

Rest von 2 ($11/3 = 3$ Rest 2). Um den maximalen Wert einer Zufallszahl auf 100 festzulegen, schreibt man

```
RANDOM rnd
X=rnd // 100
```

Mit diesen einfachen Gleichungen lassen sich scheinbar zufällige Zahlen in jedem gewünschten Bereich generieren. Achten Sie darauf, den Bereich der Variablen *rnd* selbst nicht zu beschränken, ansonsten ergibt sich nur eine sehr begrenzte Anzahl von Zufalls-werten, die erzeugt werden können.

MATHEMATISCHE AUSDRÜCKE

An vielen Stellen findet man im Pro-

gramm Ausdrücke, die mehr als einen einzelnen mathematischen Operator enthalten. Im Gegensatz zu den üblichen Konventionen gibt es in PBASIC keine Hierarchie der Operatoren, der Ausdruck wird streng von links nach rechts abgearbeitet. Schreibt man beispielsweise

```
5+2*3
```

so lautet das Ergebnis 21 und nicht 11, wie es den mathematischen Regeln entsprechen würde.

Glücklicherweise bietet die BASIC-stamp 2 Klammer-Operatoren, mit denen bestimmte Terme zusammengefaßt werden können. In PBASIC würde der obige Ausdruck korrekt

```
5+(2*3)
```

lauten, oder viel einfacher

```
2*3+5
```

Listing 1. Kalibrieren des 555-Timers

```
freq var word
high 5 `Oszillator an
start:
  count 6, 100, freq
  debug dec 5 freq*10, cr
  goto start
```

Listing 2. Hilfsprogramm zur Ausrichtung der Sensoren

```
high 5 `Oszillator an
top:
if in9=1 then nol eft
debug "left"
nol eft:
debug 9 `Schri tt
if in0=1 then nori ght
debug "right"
nori ght
debug cr
goto top
```

Ziehen wir ein etwas komplizierteres Beispiel aus dem Listing 4 heran. Im Unterprogramm *left*: generiert das Programm eine Zufallszahl und durchläuft eine Schleife.

```
for i=1 to delay +
(rnd&3*5) `Zufal lsbewegung
```

Die Anzahl der Schleifenwiederholungen beträgt $delay + (rnd \& 3 * 5)$, wobei *delay* eine Konstante mit dem Wert 10 ist. Die Zufallszahl reicht von 0 bis 15, da 3 als maximaler Wert der Zufallszahl mit 5 multipliziert wird. Die Zahl der Durchläufe kann also 10, 15, 20 oder 25 betragen.

FEHLERQUELLEN

Sollte sich der BoE-Bot nicht oder nicht wie gewünscht bewegen, können mehrere Fehler die Ursache sein:

Listing 3. Ein Programm zur einfachen Bewegungs-Steuerung

```
`Roboter mit Kollision-Vermeidung - Williams
i var word `Schleifenzähler
right_IR var in0 `rechter IR-Empfänger
left_IR var in9 `linker IR-Empfänger
right_servo con 3 `rechter Servo
left_servo con 15 `linker Servo
IR_out con 5 `38 kHz freigeben
delay con 10 `Motor-Zykluszeit
center con 750
speed con 100

high IR_out `IR einschalten
pause 50

sense:
```

```
if left_IR=0 or right_IR=0 then turn
`kein Hindernis: geradeaus
forward:
```

```
for i=1 to delay*2
  pul sout left_servo, center-speed
  pul sout right_servo, center+speed
  pause 20
next
goto sense
```

```
turn:
  pause 50
  for i=1 to delay*5
    pul sout left_servo, center-speed
    pul sout right_servo, center-speed
    pause 20
  next
  goto sense
```

Listing 4. Ein Programm zur komplexen Bewegungs-Steuerung

```
' Roboter mit Kollision-Vermeidung - Williams
rnd var word      ' Zufallsgenerator
i var word        ' Schleifenzähler
right_IR var in0  ' rechter IR-Empfänger
left_IR var in9   ' linker IR-Empfänger
right_servo con 3 ' rechter Servo
left_servo con 15 ' linker Servo
IR_out con 5      ' 38 kHz freigeben
delay con 10      ' Motor-Zykluszeit
center con 750
speed con 100

high IR_out      ' IR einschalten
pause 50

sense:
  rnd=rnd+1      ' Zufallszahl erzeugen
  ' Hindernis zentral: umdrehen
  if left_IR=0 and right_IR=0 then back
  ' Hindernis rechts oder links:
  ' nach links oder rechts
  if left_IR=0 then right
  if right_IR=0 then left
  ' kein Hindernis: geradeaus
  forward:

  for i=1 to delay*2
    pul sout left_servo, center-speed
    pul sout right_servo, center+speed
    pause 20
  next
  goto sense

back:
  pause 50
  for i=1 to delay*3
    pul sout left_servo, center+speed
    pul sout right_servo, center-speed
    pause 20
  next
  ' nach umdrehen zufällig nach
  ' links oder rechts
  random rnd
  if rnd&1 then right

left:
  pause 50
  random rnd
  ' Zufallsgeschwindigkeit
  for i=1 to delay + (rnd&3*5)
    pul sout left_servo, center-speed
    pul sout right_servo, center-speed
    pause 20
  next
  goto sense

right:
  pause 50
  random rnd
  ' Zufallsgeschwindigkeit
  for i=1 to delay + (rnd&3*5)
    pul sout left_servo, center+speed
    pul sout right_servo, center+speed
    pause 20
  next
  goto sense
```

- Die IR-LEDs und/oder die Sensoren sind nicht richtig ausgerichtet, sind nicht weit genug voneinander entfernt oder befinden sich nicht weit genug vorne am Chassis.

- Die seitliche Abstrahlung der LEDs ist nicht ausreichend unterbunden.

- Der 3300- μ F-Elko ist nicht oder nicht richtig angeschlossen, so daß Störungen der Servos die BASIC-stamp während des Betriebs zurücksetzen. Um dies zu überprüfen, schließt man den PC an, gibt zu Beginn des Programms die Zeile

```
DEBUG "RESET", CR
```

ein, lädt und startet das modifizierte Programm. Die Meldung RESET sollte nur einmal auf dem Monitor erscheinen, bei mehrfacher Meldung dagegen setzen Störungen die BASIC-stamp zurück.

- Sollte der Roboter auf ein Hindernis überhaupt nicht reagieren, ist eventuell die Verdrahtung der LEDs oder der Empfänger nicht in Ordnung. Überprüfen Sie die Funktion der LED-Ansteuerung, indem Sie die IR-LEDs durch "sichtbare" ersetzen. Die Sendefrequenz läßt sich mit der BASIC-

stamp oder einem Oszilloskop ermitteln.

- Reagiert der Roboter auf zu weit entfernte Hindernisse, ist die Empfindlichkeit des Systems zu hoch. Das Problem läßt sich durch ein paar Experi-

mente mit der Platzierung und Ausrichtung von LEDs und Detektoren beheben. Auch kann man die IR-LEDs mit größeren Vorwiderständen (zum Beispiel 2 k Ω) ausstatten, um die Ausgangsleistung zu verringern.

(990054-4)rg

BoE-Bot als Kettenfahrzeug

Das Original-Chassis ist nicht nur bei den in Elektor inserierenden Firmen Geist und Stippler erhältlich, sondern auch in einer Modifikation. Nach den Parallax-Bauplänen fertigt nämlich die Firma

Ivanka Bangert
Medizinische Geräte
Ehrenbergweg 8
72379 Hechingen
Tel. 07471/3793

ein Aluminium-Chassis an, das statt mit Rädern mit (Kunststoff-) Raupenkettens ausgestattet ist. Das Chassis verfügt über fertig abgeglichene Servos, Mikroschalter mit langen, bis an die Chassiskanten reichenden Schaltbügeln, einen Piezo-Summer und eine Batteriehalterung für vier Mignonzellen. Die Bauteile sind schon verkabelt, so daß lediglich das Board of Education angeschlossen werden muß. Das komplette Chassis ist zum Preis von 297 DM erhältlich.

