

Roboter für Lötter, Simulator für Soft-Werker

Spielgefährten

Praxis & Tipps | Praxis .. Uhr

Benjamin Benz, Carl Thiede, Thorsten Thiele

c't-Bots drehen auf der Stelle, spielen mit Kollegen Räuber und Gendarm, sammeln Golfbälle ein und bieten viel Spielraum für eigene Experimente. Wer nicht löten möchte, der erschafft mit c't-Sim virtuelle Roboter, lässt sie gegeneinander antreten und transplantiert ihre Intelligenz vielleicht später bei Freunden in reale c't-Bots.

Mobile Roboter faszinieren ebenso wie Modellautos, nur bieten sie viel mehr Möglichkeiten. Mit wenigen Code-Zeilen bringt man ihnen bei, Hindernissen auszuweichen, Wollmäuse in dunklen Ecken zu jagen oder einem neugierig hinterherzufahren. Beherrscht man erst einmal die Ansteuerung von Motoren und anderen Aktuatoren sowie die Auswertung der Sensoren, steht auch der Lösung komplexer Aufgaben nichts entgegen. Einen Golfball in einem Spielfeld zu suchen, ihn einzufangen und schließlich an einer bestimmten Stelle einzulochen erfordert zwar etwas Tüftelei, muss aber nicht den gut ausgestatteten Forschungslabors vorbehalten bleiben - wie wir mit dem c't-Bot zeigen wollen.



Der c't-Bot bildet eine relativ preiswerte und flexible Plattform für eigene Entwicklungen. Seine Konstruktion ist robust und solide. Das erlaubt vielfältige Erweiterungen sowohl der Mechanik als auch der Elektronik. Dank der runden Grundfläche dreht der c't-Bot auf der Stelle. Er nimmt seine Umgebung über Abstandssensoren wahr und besitzt genug Rechenleistung, um sich mit dem geeigneten Code clever zu verhalten. Im Endausbau soll er autonom Gegenstände wie beispielsweise einen Golfball suchen und von A nach B transportieren. Mit einem Kontroll-PC oder anderen Robotern soll er sich dann per Funk abstimmen.

Da der ganze Roboter nur als Bausatz zu haben ist, entfallen die Montage- und

Bestückungskosten. Insgesamt erhält man so nach ein wenig Lötarbeit für rund 200 Euro einen flexiblen Roboter. Der Aufbau des c't-Bots sollte allen, die halbwegs mit einem LötKolben umgehen können, keine Probleme bereiten. Wir haben ganz bewusst auf SMD-Bauteile verzichtet. Eine Artikelserie wird in den kommenden c't-Ausgaben Stück für Stück den Aufbau, die Inbetriebnahme und die C-Programmierung des c't-Bot darstellen.

Darüber hinaus stellen wir den Simulator c't-Sim und dessen Entwicklung vor. Er bietet auch reinen Soft-Werkern die Möglichkeit mitzuspielen. c't-Sim kommuniziert mit mehreren virtuellen Robotern, gaukelt ihnen eine Umgebung vor und liefert Sensorwerte zurück. Der Code, der auf c't-Sim und auf c't-Bot läuft, ist kompatibel. So lassen sich komplexe Verhaltensmuster für den Roboter bequem am PC testen. Die Oberfläche des in Java geschriebenen Simulators soll aber später auch die c't-Bots kontrollieren. Selbst eine Kombination aus realen und simulierten Robotern ist denkbar.

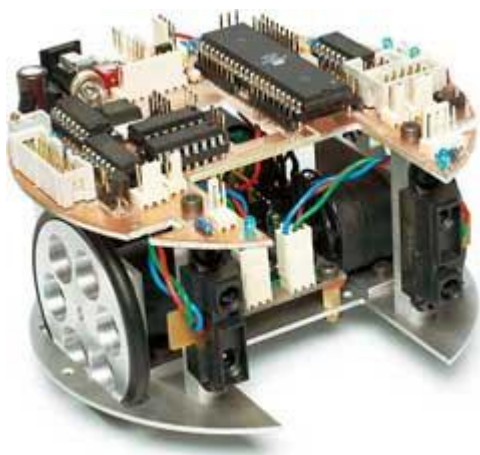
c't-Bot und c't-Sim sind jedoch keine fertigen Spielzeuge aus dem Regal, sondern bieten viel mehr eine Gelegenheit, selbst zu experimentieren. Die Artikel begleiten das Projekt und vermitteln die nötigen Kenntnisse. Sie werden abwechselnd erscheinen. Anregungen, Erweiterungsvorschläge und clevere Programme greifen wir gerne auf oder stellen sie auf die Projektseite [1]. Für Diskussionen rund um den Roboter steht auf heise online ein Leserforum [2] bereit.

Hier widmen wir uns zuerst der mechanischen Konstruktion. Auch das Elektronik-Konzept soll nicht zu kurz kommen. Der nächste Hardware-Artikel befasst sich mit Montage, Inbetriebnahme und Hardware-naher Programmierung der Sensoren sowie Aktuatoren. Alle Platinen, vorbereitete Mechanik und vollständige Teilesätze werden bei eMedia und Segor-Electronics erhältlich sein, sobald der Inbetriebnahme-Artikel erscheint.

Aller Anfang ...

Manövrierfähig soll er sein, der c't-Bot. Daher bietet sich eine Konstruktion mit runder Grundfläche an. Ohne hervorstehende Ecken und Kanten besteht kaum die Gefahr, dass sich das Fahrzeug an Hindernissen verhakt. Verwendet man nur zwei Räder, deren Achsen den Mittelpunkt der runden Grundfläche schneiden, kann der Roboter auf der Stelle drehen. Mit nur zwei Rädern würde der c't-Bot jedoch sofort kippen. Wir haben uns aus Platz- und Kostengründen daher für einen kleinen Gleitpin aus Kunststoff entschieden, der den Roboter hinten abstützt. Das Gewicht der Akkus verhindert ein Kippen nach vorn. Über die meisten Untergründe gleitet der abgerundete Stützpin problemlos hinweg und selbst Fugen zwischen Fliesen oder normaler Teppichboden halten den c't-Bot nicht auf. Konstruktionen mit Rollen oder ein dynamisches Ausgleichen der Kippmomente durch Beschleunigungen wären zwar

auch denkbar, jedoch deutlich aufwendiger.



Die Hauptplatine des Prototyps erstrahlt ohne Lötstoplack noch in kupfernem Glanz, entspricht sonst aber schon fast der finalen Version.

Motoren, Sensoren und Akkus füllen die Grundplatte fast völlig aus. Die Elektronik musste daher in den zweiten Stock ausweichen. Die modulare Bauweise erlaubt es, künftige Erweiterungen auf weiteren Ebenen hinzuzufügen. Die gesamte Grundkonstruktion besteht aus Aluminium und ist stabil genug, um später auch noch Greifer, Kamera und Co. zu tragen.

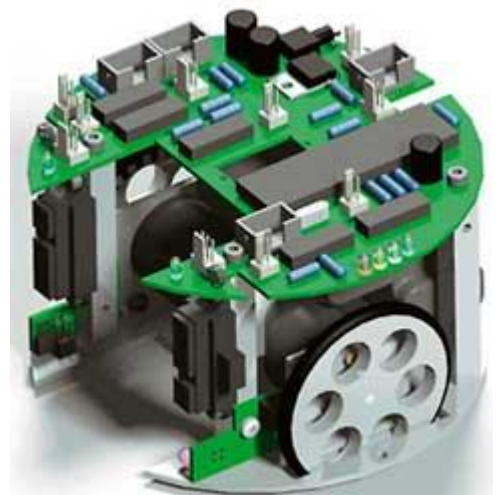
Agil

Zwei Gleichstrommotoren (6 Volt) treiben die beiden Räder an. Die Firma FTB Faulhaber baut die Motoren (2619-006SR) als Flachläufer mit einem eisenlosen Rotor. So entsteht nur ein sehr geringes Trägheitsmoment. Ein Untersetzungsgetriebe (33:1) reduziert die Drehzahl und steigert das Drehmoment. 0,03 Nm reichen aus, um selbst einen mit schweren Büchern beladenen c't-Bot zu bewegen. Bei maximal 151 U/min und einem Raddurchmesser von 57 mm erreicht der unbeladene Roboter eine Geschwindigkeit von bis zu 0,45 m/s (1,6 km/h).

Die Räder, gefräst aus Aluminium, sind genau auf die Motoren abgestimmte Einzelanfertigungen. Sechs große Löcher reduzieren das Gewicht der Räder deutlich und verleihen ihnen ein schickes Design. Ein O-Ring dient als Reifen und sorgt für guten Halt auch auf rutschigen Flächen.

Die runde Grundplatte soll einerseits möglichst klein sein, damit der Roboter auch durch enge Gänge patrouillieren kann. Andererseits erfordern die einzelnen Bauteile auch einiges an Platz. Neben Motoren, Elektronik und Akkus soll auch Raum für Sensoren und Aktuatoren bleiben. Mit einem Durchmesser von 120 mm dreht der c't-Bot auf der Fläche einer CD und kann in einem rechteckigen Ausschnitt später Gegenstände von der Größe eines Golfballes einfangen und transportieren - eine geeignete Mechanik zum Verriegeln des Transportfachs behandelt ein späterer Artikel.

Motoren,
Sensoren



Die Sensorplatinen in der unteren Ebene fehlen dem Prototyp noch.

Sensibel

Auf der Innenseite der Räder kleben Kodierscheiben mit abwechselnd schwarzen und weißen Feldern. Zwei Reflexlichtschranken (CNY70) messen so die Drehgeschwindigkeit der Räder. Diese Rad-Encoder ermöglichen es nicht nur, den Geradeauslauf des Roboters zu kontrollieren, sondern auch seine Position relativ mitzulongen. Allerdings liefern sie nur korrekte Ergebnisse, wenn die Räder keinen Schlupf haben und nicht durchdrehen. Dazu später mehr.

Ohne weitere Sensoren würde der c't-Bot im Dunkeln tappen und seine Umgebung nicht wahrnehmen. Daher liefern zwei Abstandssensoren Informationen über Hindernisse in Vorausrichtung. Eine integrierte Infrarotdiode strahlt modulierte Licht aus, welches das Hindernis auf den Sensor zurück reflektiert. Dort bündelt eine Linse den Lichtstrahl und führt ihn auf einen richtungssensitiven Empfänger (PSD, Position Sensitiv Detector). Der Abstand beeinflusst den Einfallswinkel.

Die Sensoren GP2D12 der Firma Sharp kümmern sich um die gesamte Verarbeitung und liefern eine analoge Spannung, die vom Abstand Sensor-Objekt abhängt. Der Erfassungsbereich liegt ungefähr zwischen zehn und 80 Zentimetern, der Blickwinkel bei rund drei Grad. Die Modulation des IR-Lichts schützt den Sensor relativ gut gegen Streulicht. Stark spiegelnde Flächen, die nicht orthogonal zum Sensor stehen, kann er jedoch nicht erfassen.

Eine einfache schwarze Linie auf dem Boden reicht aus, um den c't-Bot durch den Raum zu lotsen. Mit zwei nach unten gerichteten Lichtschranken (CNY70) kann er diese verfolgen. Zwei weitere Lichtschranken tasten den Boden links und rechts im vordersten Bereich des Roboters ab. Sobald sie das selbst ausgestrahlte Licht - der CNY70 besteht aus einer LED und einem Phototransistor - nicht mehr empfangen, steht der c't-Bot entweder vor einem Abgrund oder einer perfekten schwarzen Fläche. So oder so sollte er dann sicherheitshalber erst einmal anhalten.

Dank zweier Photosensoren kann der c't-Bot beispielsweise den Strahl einer Taschenlampe verfolgen, sich in dunklen Höhlen verstecken oder mit anderen Robotern Räuber und Gendarm spielen. Dabei würde der Räuber eine kleine Lampe tragen und der Gendarm das Licht suchen.

Photosensoren können so dazu dienen, einfache Verhaltensmuster und Reaktionen auf die Umgebung zu testen. Die zwei lichtempfindlichen Widerstände (LDR) sitzen im vorderen Bereich der Platine, räumlich voneinander getrennt. Daher messen die beiden Sensoren die Lichtintensität an zwei verschiedenen Punkten. Es entsteht ein Helligkeitsgradient.

Damit aber noch nicht genug. Um sich besser zu orientieren, soll der Roboter in einer

späteren Ausbaustufe den Sensor einer optischen Maus erhalten. Das erlaubt ihm, seine Position genauer zu verfolgen als mit den Rad-Encodern allein. Als Krönung könnte man dem c't-Bot sogar eine Kamera spendieren. Deren Bilder eignen sich einerseits für Überwachungsvideos, können aber auch die Rohdaten für eine Objektverfolgung und Erkennung liefern. Damit könnte der c't-Bot dann zum Beispiel einen Salzstreuer suchen, in seinem U-förmigen Fach einfangen und direkt zum Teller liefern.

Intelligent

Das Hirn des c't-Bots thront auf der Platine im ersten Stock. Ein Mikrocontroller (Atmel ATmega32) liefert die nötige Rechenleistung, um Motoren anzusteuern, Sensoren auszuwerten und mit der Außenwelt zu kommunizieren. Er führt fast alle Befehle in einem Taktzyklus aus - davon stehen 16 Millionen pro Sekunde zur Verfügung. Eigenen Programmcode nehmen 32 KByte Flash-Speicher auf. 2 KByte SRAM speichern Variablen und 1 KByte EEPROM merkt sich Parameter auch bei abgeschalteter Versorgungsspannung.

Der ATmega32 besitzt insgesamt 32 I/O-Pins, davon hängen acht an einem internen A/D-Umsetzer. Dessen 10-Bit-Auflösung reicht für alle hier besprochenen Sensoren spielend aus.

Sicherlich gibt es noch deutlich schnellere Prozessoren mit mehr Speicher oder I/O-Pins, doch meist nur als SMD-Typen, die der Hobby-Löter nur mühsam verarbeiten kann. Beim c't-Bot behelfen wir uns mit einem Trick und nutzen serielle Schieberegister, um mehr digitale Ausgänge anzusteuern.

Kommunikativ

Mikrocontroller sind nicht darauf ausgelegt, mitzuteilen, was sie gerade tun, woran das Programm festhängt oder welche Werte ein Sensor liefert. Auf der Platine des c't-Bot geben daher acht frei programmierbare LEDs über den Systemstatus Auskunft. Obendrein kann der Roboter ein LC-Text-Display mit 4x20-Zeichen huckepack nehmen. Das sollte reichen, um selbst komplexe Software zu debuggen und den Status des Programmes zu überwachen.

Der c't-Bot horcht auf Befehle, die ihm Herrchen und Frauchen per IR-Fernbedienung (RC5-Code) übermitteln. Man kann ihn so nicht nur lenken, sondern auch Parameter ändern oder verschiedene Subroutinen starten, ohne die Firmware neu zu übertragen. Später soll eine Funkstrecke Kommandos und Statusinformationen zwischen PC und Roboter übertragen.

Kraftvoll

Fünf NiMh-Akkus (Mignon-Zellen) liefern 6 Volt Gleichspannung für die Motoren. Stabile 5 Volt für die Elektronik erzeugt ein Low-Drop-Regler bis zu einer Eingangsspannung von 5,5 Volt. Das deckt den größten Bereich der Entladekurve der NiMh-Zellen ab. Aber selbst danach fällt die Ausgangsspannung nur langsam. Bis rund 4,5 Volt spielt die Elektronik mit. Lediglich die Motoren drehen bei fast vollständig entladenen Akkus etwas langsamer. Batterien sollte man lieber nur im Viererpack einsetzen, da sie pro Zelle 1,5 Volt liefern und sonst die Motoren unnötig strapazieren.

Bei einer Stromaufnahme von rund 250 mA - Basisversion mit laufenden Motoren ohne Display - hält ein Satz Akkus (2300 mAh) rund neun Stunden. Wem das beim Entwickeln nicht reicht, der kann den Roboter auch über ein 6-Volt-Netzteil mit Energie versorgen. Auf einen internen Akkulader haben wir aus Platz- und Kostengründen verzichtet. Die Mignon-Zellen lassen sich aber mit den handelsüblichen Schnellladegeräten zügig wieder befüllen.

In medias res

Nach den allgemeinen Designüberlegungen und der Vorstellung der Kernkomponenten, wollen wir nun Schritt für Schritt die einzelnen Baugruppen detailliert beschreiben. Dieser erste Artikel konzentriert sich dabei auf die Motorsteuerung.

Eine H-Brücke verbindet die Gleichstrommotoren direkt mit der Batteriespannung, die TTL-Pegel des Mikrocontrollers liefern nur die Steuersignale. Da die Laufrichtung von der Polung abhängt, benötigt jeder Motor vier Schalter (siehe Grafik). Dennoch ergeben nur drei Konstellationen Sinn, alle anderen führen zum Kurzschluss: Sind zwei sich diagonal gegenüberstehende Schalter geschlossen und alle anderen offen, so dreht sich der Motor rechts- oder linksherum - je nach Polung. Öffnet man mindestens drei Schalter, steht der Motor still.

Der Baustein L293D vereint gleich zwei solche H-Brücken in einem Chip und kann genug Strom treiben, um die Motoren direkt anzusteuern. Er erwartet pro Motor die Drehrichtung auf zwei Pins - einmal invertiert und einmal direkt - sowie ein Enable-Signal. Nur wenn dieses auf High steht, schaltet er den Motor an. Speist man diesen Eingang mit einem PWM-Signal (Pulsweitenmodulation), so lässt sich auch die Drehgeschwindigkeit regeln.

Ein PWM-Signal besteht aus einem Rechteck mit fester Grundfrequenz. Man verschiebt nur die Anteile der Low- und High-Pegel. Soll der Motor auf voller Geschwindigkeit laufen, so wird t_{low} null und das Signal ist die ganze Zeit high. Zum

Deaktivieren des Motors setzt man t_{high} gleich null. Für eine mittlere Geschwindigkeit können beide Zeiten identisch sein. Es gilt dabei immer: $t_{\text{low}} + t_{\text{high}} = 1/f_{\text{PWM}} =$ konstant. Da die Gleichstrommotoren träge sind, reagieren sie nur auf den zeitlichen Mittelwert der Spannung. Dieser ergibt sich zu:

$$V_{\text{eff}} = V_{\text{In}} \cdot t_{\text{high}} / (t_{\text{high}} + t_{\text{low}})$$

Eine PWM-Steuerung erfordert viel weniger Aufwand als die direkte Regelung der Spannung. Die Timer-Einheit des Controllers erzeugt solche PWM-Signale autonom, ohne dass man Rechenzeit dafür aufwenden muss.

Das war der erste Streich

Mit der Montage der Mechanik, dem Aufbau der Platine und ersten Gehversuchen mit dem Mikrocontroller wird sich der nächste c't-Bot-Artikel genauer befassen. Wer bis dahin nicht nur im Schaltplan schmökern, sondern schon mal in die Mikrocontroller-Programmierung hineinschnuppern möchte, findet bei den Projekten COM-auf-LAN [3] und c't-Netz-Schalter [4] einführende Erklärungen. Da der c't-Bot denselben Prozessortyp nutzt, ähnelt sich auch die Codebasis. Bereits im nächsten Heft bietet der c't-Sim eine Gelegenheit, erste Erfahrungen mit der Programmierung von Roboterverhalten zu sammeln. (**bbe[1]**)

Literatur

[1] **Webseite zum c't-Bot-Projekt**[2]

[2] **Leserforum zum Projekt**[3]

[3] **Webseite zum COM-auf-LAN-Projekt**[4]

[4] **Webseite zum c't-Netz-Schalter-Projekt**[5]

Soft-Link[6]

"Der eigene Roboter"	
Schaltpläne für den c't-Bot und weitere Artikel zu Robotik finden Sie in der c't 2/2006:	
<i>Robotik in der Wissenschaft und Anwendung</i>	S. 124
<i>c't-Bot selbst gebaut</i>	S. 130
<i>Heimroboter im Test</i>	S. 136
<i>Der Roboter als soziales Wesen</i>	S. 144

Sensoren und Aktuatoren				
Typ	Anzahl	Verfahren	Zweck	Basisversion

Sensoren				
CNY70	2	Reflexlichtschranke	Rad-Encoder	+
CNY70	2	Reflexlichtschranke	Liniensensor	+
CNY70	2	Reflexlichtschranke	Abgrunddetektion	+
GP2D12	2	optische Triangulation	Abstandssensor nach vorn	+
LDR	2	Photowiderstand	Helligkeitsmessung	+
IS471F	1	Lichtschranke	Überwachung Transportfach	+
Maussensor	1	integrierte Kamera	Positionsverfolgung	-
Kamera	1		Überwachung, Objektverfolgung	-
Klappensensor	1	opt. oder mag.	Überwachung Klappe	-
Aktuatoren				
Faulhaber 2619	2	Gleichstrommotor	Antrieb	+
Servo	1		Verriegelung Fach	-
Servo	2		Steuerung Kamera	-

URL dieses Artikels:

<http://www.heise.de/-290274>

Links in diesem Artikel:

- [1] <mailto:bbe@ct.de>
- [2] <https://www.heise.de/ct/artikel/c-t-Bot-und-c-t-Sim-284119.html>
- [3] <http://www.heise.de/ct/foren/S-c-t-Bot-und-c-t-Sim/forum-89813/list/>
- [4] <https://www.heise.de/ct/artikel/c-t-COM-auf-LAN-Adapter-284121.html>
- [5] <https://www.heise.de/ct/artikel/Netz-Schalter-284123.html>
- [6] <http://www.heise.de/ct/06/02/links/130.shtml>

Copyright © 2006 Heise Medien