

Aufbau und Inbetriebnahme des c't-Bot

Hallo Welt!

Praxis & Tipps | Praxis .. Uhr

Benjamin Benz, Carl Thiede, Thorsten Thiele

Aus einer Tüte mit Elektronik- und Mechanikbauteilen entsteht mit Hilfe eines dampfenden Lötkolbens in wenigen Stunden der c't-Bot. Hat man ihm erst einmal Leben eingehaucht, führt er treu die eingebrannten Programme aus.

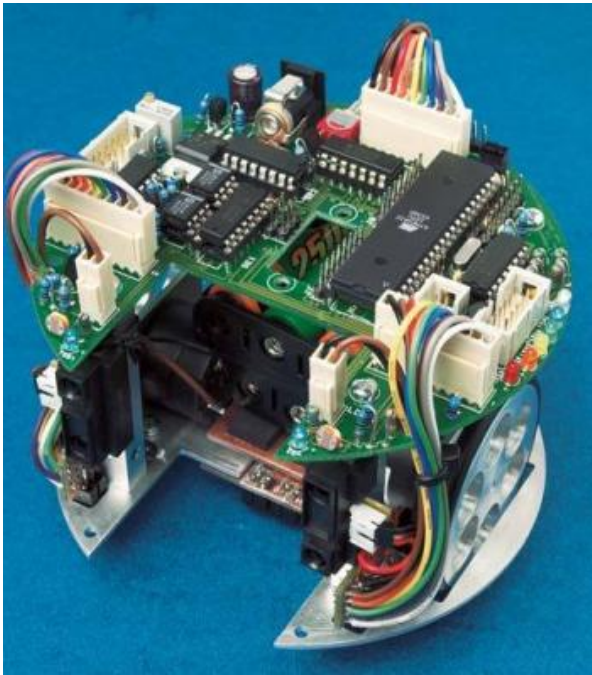
Nachdem sich die ersten beiden Artikel zum c't-Bot mit der prinzipiellen Konstruktion [1] und der Simulation des Roboters [2] beschäftigt haben, ist es nun an der Zeit, den c't-Bot zum Leben zu erwecken. Alle benötigten Platinen bieten eMedia und Segor-electronics für rund 20 Euro. Dazu gibt es bei Segor noch einen Teilesatz mit allen Mechanik- und Elektronikbauteilen für 200 Euro sowie optional Display und Akkus. So erhält man einen Bausatz, der neben den vier Platinen, drei FR4-Platten und allen Elektronikteilen auch das Skelett des Roboters aus Aluminium, die Mechanik sowie die Motoren enthält. Wer ein wenig Zeit mitbringt und den Lötkolben am richtigen Ende anfassen kann, sollte auch ohne große Vorkenntnisse in der Lage sein, den c't-Bot aufzubauen. Für Einsteiger ohne jegliche Löt- und Elektronikerfahrung haben wir auf der Projektseite [3] und unter dem Soft-Link einige Hinweise gesammelt. Auf schwierig zu verarbeitende SMD-Bauteile haben wir ganz bewusst verzichtet. Auch für die Montage der Mechanik reicht übliches Haushaltswerkzeug.



Heißes Eisen

Der Großteil der Elektronik sitzt auf der runden Hauptplatine; für die Sensoren im unteren Stockwerk gibt es kleine Zusatzplatinen. Der Aufbau sollte mit einem Lötkolben mit feiner Spitze und dünnem Lötzinn (0,7 mm) auch Anfängern gelingen.

Die Bestückungspläne geben Aufschluss, wo welches Bauteil wie herum zu montieren ist. Steht seine Bezeichnung in Klammern, so sitzt es nicht auf der Seite mit



Der fertig montierte c't-Bot erkennt mit zwei Infrarotabstandssensoren Hindernisse in Fahrtrichtung. Lichtschranken detektieren Abgründe und ein Maussensor ermittelt die Position.

Bestückungsdruck, sondern auf der Unterseite. Bei den Widerständen (Rxxx) spielt die Einbaurichtung keine Rolle, bei den meisten anderen Teilen jedoch schon. Auf der Projektseite [3] finden sich zusätzliche Detailfotos der bestückten Platinen. Einsteiger finden dort auch Informationen zu einigen Bauteile und eine Anleitung, wie man die Werte der Widerstände ermittelt.

Gegenüber dem ursprünglichen Schaltplan aus [1] haben sich ein paar kleine Änderungen bei der Dimensionierung diverser Widerstände ergeben. Im Zweifelsfall ist die hier abgedruckte Stückliste ausschlaggebend. Ein Schaltplan mit aktualisierten Werten steht auf der **Projektseite[1]** zum Download bereit.

Achtung: Seit Veröffentlichung dieses

Artikels im Jahr 2006 haben sich wiederum einige Verbesserungen und kleine Änderungen ergeben. Bitte beachten Sie daher auch die Hardware-Seiten im **Projekt-Wiki[2]**.

Schritt für Schritt

Der Reflexkoppler U1 auf der Hauptplatine soll später den Schwenkarm zur Verriegelung des Transportfachs detektieren und wird als einziges Bauteil von unten mit der Platine verlötet (Bauteilestempelung in Richtung U10). Alle anderen Bauteile kommen auf die Oberseite (Bestückungsdruck). Die Kabelbäume werden durch die Schlitze in der Platine nach unten geführt. Das vereinfacht die Montage.

Es folgen die Dioden (D1 bis D3); ihr Kennring muss in die vom Bestückungsdruck angegebene Richtung zeigen. Die IC-Sockel besitzen eine Kerbe, die später die korrekte Polung der ICs anzeigt. Mit dem Einsetzen der ICs wartet man jedoch bis ganz zum Schluss.

Die Polung der Keramikkondensatoren C1 bis C4 spielt keine Rolle, ebenso wenig die der Spule (L1) oder des Quarzes (Q1). Der erhält eine Isolierscheibe als Unterlage. Bei den acht Leuchtdioden kennzeichnet das längere Beinchen die Anode (Pluspol). Den Spannungswandler (IC10) knickt man so ab, dass die Kühlfahne direkt auf der Platine

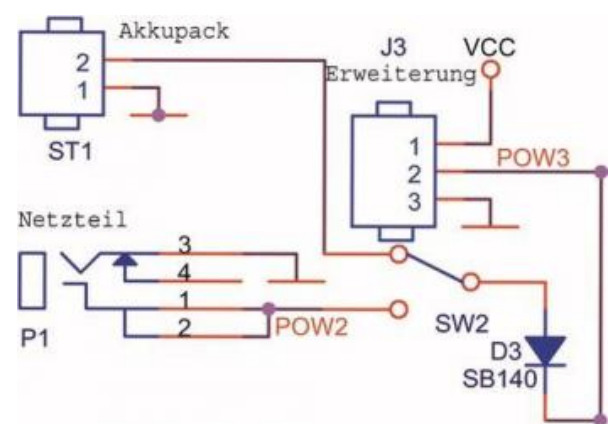
aufliegt.

Die lichtempfindliche abgerundete Seite des IR-Empfängers (IC9) braucht freie Sicht nach hinten. Danach lötet man die Stiftleisten J1 bis J8, SW1, SW2 und BR1 sowie die Transistoren (TR1 bis TR6) ein. Ein Jumper auf BR1 aktiviert später die Hintergrundbeleuchtung des Displays.

Den Platz des Widerstandes R29 nimmt eine Zenerdiode mit 2,4 V ein. Ihr Kennring zeigt in Richtung IC10. Bei den restlichen Widerständen spielt die Ausrichtung keine Rolle, sie haben jedoch nur stehend genug Platz. In welche Richtung die zwei Fotowiderstände (LDR1, LDR2) Licht messen sollen, muss jeder selbst entscheiden. Bei den Steckverbindern ST1 bis ST9 und dem Elektrolytkondensator C5 muss man wieder auf die Polung achten. Nun fehlen nur noch das Potenziometer für den Display-Kontrast POT1 und die DC-Buchse P1.

Erster Test

Bevor man die empfindlichen ICs einsteckt, empfiehlt sich ein Funktionstest der Spannungsversorgung. Dazu hängt man entweder einen Akkupack aus fünf Mignon-Zellen an ST1 oder ein Netzteil mit 6 Volt Gleichspannung an P1. Zeigt ein Voltmeter zwischen Pin 1 und Pin 3 der Leiste J3 5,0 Volt an, arbeitet der Spannungsregler korrekt. Zwischen Pin 2 und Pin 3 liegt hingegen direkt die Eingangsspannung. Achtung, die Belegung des DC-Steckers hat sich gegenüber dem ersten Schaltplan geändert: Masse muss nun außen und nicht mehr innen liegen. Eine Schutzdiode (D3) bewahrt jedoch die Schaltung vor Schäden durch Verpolung. Nach erfolgreichem Spannungstest und Abklemmen der Stromversorgung kann man die verbleibenden ICs einsetzen.



Die Stromversorgung hat sich gegenüber dem ursprünglichen Schaltplan verändert: Masse liegt nun außen an der Buchse und eine Diode schützt die Schaltung.

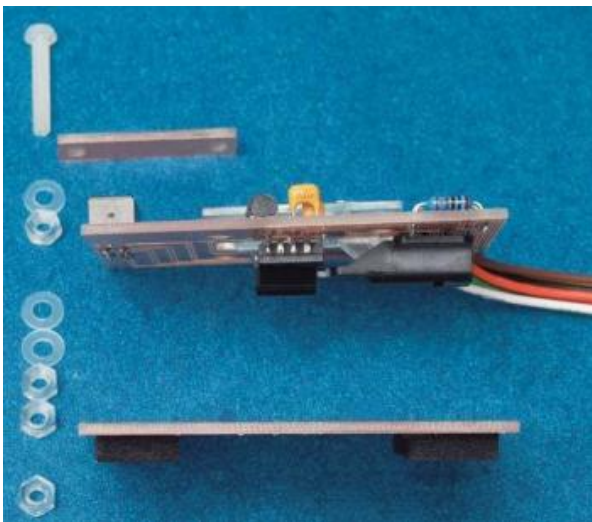
Fühler

Auf den beiden kleinen Sensorplatinen zeigen Bauteile in drei verschiedene Richtungen. Je eine Reflexlichtschranke (U103, U104) beobachtet den Boden und bekommt die Platine zwischen ihre Beinchen gesteckt. Die Stempelung zeigt bei U104 zur Platine hin, bei U103 weg - Achtung, an dieser Stelle stimmt der Bestückungsdruck auf der Platine nicht, der hier abgedruckte jedoch schon. Die anderen beiden (U101,

U102) beobachten die Räder und sitzen auf der Seite mit Bestückungsdruck. Die Leuchtdiode (LED101, rechte Platine) und die Lichtschranke (U105, linke Platine) kommen auf die Unterseite. Die Lichtschranke muss dazu so abgewinkelt werden, dass ihre flache Seite später zur Leuchtdiode blickt. Das Verlöten aller Kabel hebt man sich bis zur Endmontage auf.

Mausbau

Damit der Roboter seine Position verfolgen kann, bekommt er seine eigene optische Maus. Dieses Sandwich aus einer Platine und drei FR4-Platten in drei Ebenen meldet Daten über Bewegungen in X- und Y-Richtung direkt an den Mikrocontroller. Die Kondensatoren (C1, C2) und Widerstände (Rxx) platziert man wie fast alle Bauteile auf der Seite ohne Leiterbahnen. Der Quarz Q1 bekommt wieder eine Isolierscheibe, der Transistor Q2 wird abgewinkelt, sodass seine flache Seite auf der Platine aufliegt. Die Stempelungen der beiden Liniensensoren (U1, U2) muss - anders als auf der Platine aufgedruckt - zu der Seite zeigen, an der auch der Quarz sitzt.



Der optische Maussensor beobachtet mit einer 19x19-Pixel-Kamera den Untergrund und liefert aufbereitete Koordinaten an den Prozessor.

Der Sensor-Chip (U3) kommt - nachdem man die Schutzfolie entfernt hat - von der Leiterbahnseite auf die Platine. Das für die Detektion benötigte Licht liefert die Leuchtdiode (LED1) aus einem Kunststoffträger heraus. Dieser sitzt zusammen mit dem Chip auf einer Seite. Die Linsenplatte kommt auf die andere Seite. Die zwei kleinen Platinenstreifen fixieren sie. Je eine PVC-Mutter und eine Unterlegscheibe sorgen für den richtigen Abstand und verhindern zu hohen Druck auf die Kunststoffteile (siehe Foto S. 209).

Schrauber

Vier Platinen sind bestückt, fast alle Teile haben ihren Platz gefunden. Einer Endmontage des Roboters steht nun nichts mehr im Wege. Es lohnt sich, den ganzen Roboter erst einmal lose zu montieren. Dann kann man die Längen der einzelnen Verbindungskabel bequem ausmessen. Sie sollten dabei möglichst dicht an den Alu-Trägern entlang verlaufen, um die Mechanik nicht zu behindern. Einlöten lassen sie sich jedoch leichter in zerlegtem Zustand.

Die Alu-Grundplatte besitzt bereits alle nötigen Bohrungen und Schlitz für Räder, das Maus-Sandwich, die Motorflansche und die Alu-Träger. Die U-förmige Aussparung in

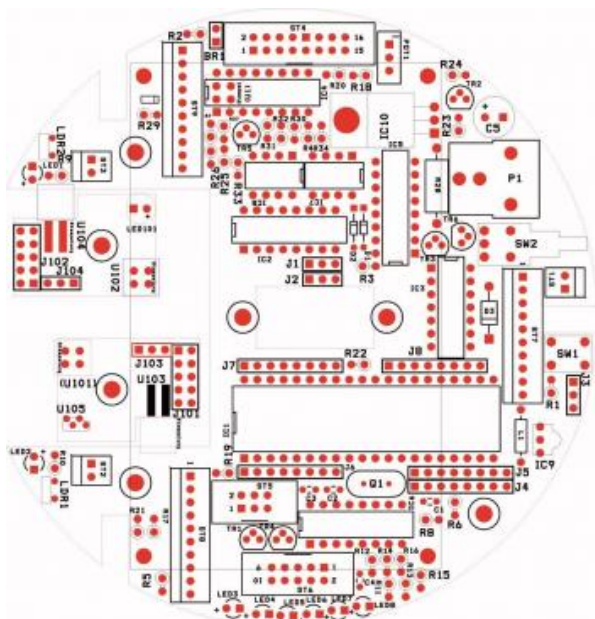
Vorausrichtung dient später dem Roboter als Transportfach. Legt man die Alu-Platte so vor sich, dass die Aussparung von einem weg weist, sollte nur links unten eine Bohrung für den Alu-Träger sein. Rechts fehlt sie, die Grundplatte ist nicht symmetrisch.

Endmontage

Zunächst befestigt man den Gleitpin mit einer Kreuzschlitzschraube, dann kommt das vorbereitete Maus-Sandwich von unten in die Aussparung. Die beiden Lichtschranken zeigen nach vorne; je zwei Unterlegscheiben halten die Platine auf Abstand zum Aluminium. Zwei Kunststoffmuttern pro Schraube fixieren die Konstruktion von oben. Darauf kommt zum Schutz das letzte unbenutzte Stück Platinenmaterial.

Damit der Roboter später seine Räder überwachen kann, klebt man auf die Innenseite der Räder Encoder-Scheiben mit abwechselnd hellen und dunklen Feldern. Bei jedem Übergang zwischen diesen Feldern liefern später die Lichtschranken U101 und U102 ein Signal. Am einfachsten kopiert man die hier abgedruckten Vorlagen auf eine transparente Folie. Wer sie lieber ausdruckt, findet eine Bilddatei auf der Projektseite. Der Schwärzungsgrad eines üblichen Laser-Druckers sollte für die CNY70-Sensoren ausreichen. Zur Not malt man die dunklen Streifen mit einem Folienstift nach.

Hat man die Motoren mit M2-Schrauben an ihren Flanschen befestigt und die Räder aufgesteckt, sichert man diese mit je einer Madenschraube. M3-Schrauben von oben und Muttern von unten verbinden diese Motorblöcke mit der Grundplatte.



Auf der recht vollen Hauptplatine bestückt man am besten zuerst die kleinen Bauteile sowie die Lichtschranke auf der Unterseite. Die ICs kommen zuletzt, nach

Die beiden vorderen Alu-Träger halten die Sensorplatinen und die Abstandssensoren in Position. Deren dreipolige Stecker zeigen nach links beziehungsweise rechts außen, sodass die Kabel nicht in das Transportfach hineinragen. Kurze Kabel verbinden sie mit den Sensorplatinen. Achtung: Die GP2D12-Sensoren verzeihen keine Verpolung. Schaut man direkt in den Stecker hinein und zeigen die Sensoren nach oben, so liegt links Pin 1. Zuerst thront dann die Hauptplatine.

Kabelbaum

Beim Verlegen der einseitig vorkonfektionierten Kabel für ST1 bis ST3

dem Test der Spannungsversorgung.

und ST7 bis ST9 sollte man oberhalb der Hauptplatine genug Spiel lassen, um die Stecker nachher abziehen zu können. Kabelschellen und -binder halten die Leitungen im Zaum. Dank der Schlitze in der Hauptplatine muss kein Kabel über den Umfang der Grundplatte hinausragen - der Roboter kann sich später nicht verheddern. Die Pinbelegung der doppelreihigen Lötfelder auf den Sensorplatinen entspricht der üblichen Zählweise - im Zickzack. Pin 1 (schwarzes Kabel) hat ein eckiges Löt-Pad. Bei beiden Motoren hängt das braune Kabel am Pluspol. Damit laufen sie gegensinnig, aber das korrigiert die Firmware des Mikrocontrollers.

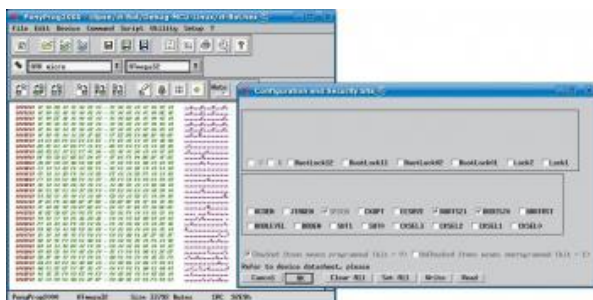
Die beiden Batteriehalter verklebt man zu einem Block, der später auf dem Maus-Sandwich ruht; Moosgummi-Pads und Klettband halten ihn in Position. Elektrisch hängen alle fünf Batterien in Reihe; dazu verbindet man den Pluspol des einen Päckchens mit dem Minuspol des anderen.

Strom marsch!

Der große Moment ist gekommen: Lebt der c't-Bot? Schaltet man den Strom ein und hat noch keine Firmware im Controller, so sollte sich der Roboter auf der Stelle drehen. Der Mikrocontroller hat im Auslieferungszustand alle Pins deaktiviert und das empfindet die H-Brücke L293D als logische Eins. Sie liefert beiden Motoren volle Spannung. Durch deren um 180 Grad gedrehte Montage läuft der linke vorwärts und der rechte rückwärts.

Damit der c't-Bot sich nicht beim Programmieren im Kabel verfängt, bockt man ihn am besten auf zwei Streichholzschachteln auf. Kleine Fehler in der Firmware könnten sonst den Roboter dazu veranlassen, sich von der Tischkante zu stürzen - das Programm startet unmittelbar nach dem Ende der Datenübertragung.

Programmiersation



Beim Programmtransfer in den Mikrocontroller ist es extrem wichtig, die Fuse-Bits passend zu setzen. Sonst schweigt er anschließend womöglich für immer.

Der Flash-Speicher des Mikrocontrollers enthält bei der Auslieferung noch kein Programm. Um ihn mit einer .hex-Datei zu befüllen, benötigt man einen Programmieradapter. Im Rahmen des BlueMP3-Projektes [4] haben wir bereits ein preiswertes Modell zum Selbstbau für die parallele Schnittstelle vorgestellt. Segor bietet jedoch auch einen Adapter für USB an. Das kostenlose Programm PonyProg2000 (siehe Soft-Link) überträgt

dann die Firmware in den Controller. Dabei muss man nicht nur den richtigen Controller (Device/AVR/ATmega32) auswählen, sondern auch auf die korrekte Einstellung der Fuse-Bits (Command/Security and Configuration Bits) achten, sonst verweigert der Controller nachhaltig den Dienst (siehe Screenshot). Je nach Programmieradapter und Entwicklungsumgebung bieten sich auch andere Programme zum Flashen des Controllers an. Auf der Projektseite haben wir einige verlinkt.

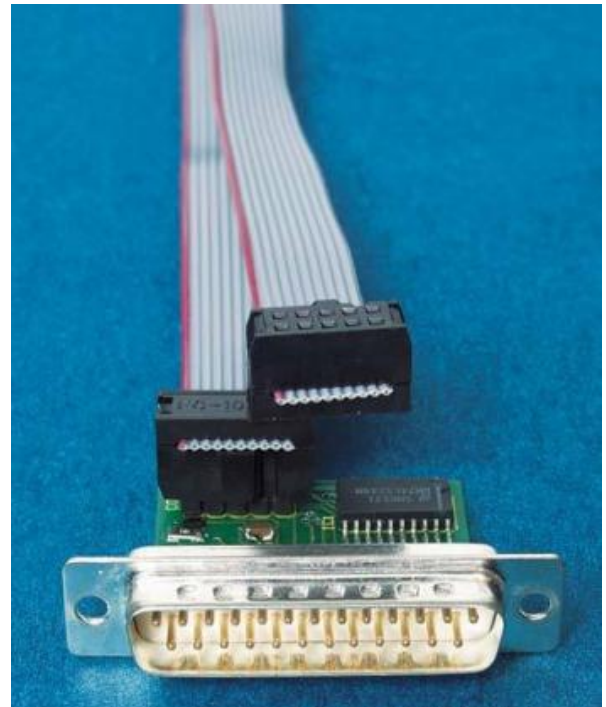
Der c't-Bot darf nur am Programmieradapter (ST6) hängen, solange er über eine eigene Stromversorgung verfügt. Sonst versucht er sich aus der parallelen Schnittstelle zu versorgen und überlastet diese unter Umständen. Den Flash-Vorgang sollte man nicht unterbrechen.

Jagt den Fehlerteufel

Für erste Gehversuche stehen auf der Projektseite mehrere Image-Dateien mit Testprogrammen zum Download bereit (ct-Bot-test-XXX.hex). Diese überprüfen die Hardware des Roboters und zeigen jeweils einen Teilstatus mit den acht LEDs an. Die Funktionstest-Tabelle listet auf, welche Sensorwerte die LEDs im jeweiligen Programm anzeigen. Der Einfachheit halber reagieren die LEDs bei den analogen Signalen jeweils nur auf das höchstwertige Bit des digitalisierten Messwerts. Somit geht beispielsweise die weiße LED an, wenn der Wert des rechten Abstandssensors größer als 511 ist. Dies erlaubt eine rasche Überprüfung, ob die Sensoren funktionieren.

Die Datei ct-Bot-test-motor.hex zeigt dieselben Werte an wie ct-Bot-test-digital.hex, dreht jedoch den Bot nach einem Reset erst ein Stück im Uhrzeigersinn, dann ein Stück anders herum. Wer ein Oszilloskop zur Hand hat, kann an Pin 3 des Steckers J1 ein Rechteck-Signal mit 33 Hz messen.

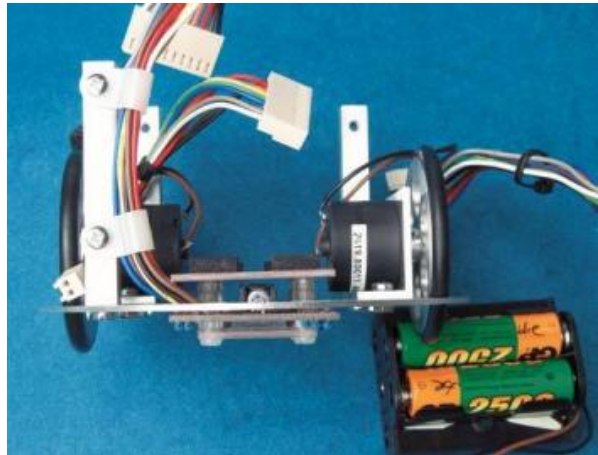
Für eine genauere Betrachtung der Sensorwerte muss man entweder das Testprogramm selbst modifizieren oder ein Display anschließen. Hängt an dem Stecker ST4 ein LC-Display mit 4 x 20 Zeichen, zeigt dieses alle Informationen auf einmal an. Für Display-Besitzer spielt es daher keine Rolle, welches Testprogramm sie verwenden, intern fragen alle drei Varianten alle Sensoren ab.



Ein Programmieradapter für die parallele Druckerschnittstelle überträgt die Firmware in den Roboter.



Auf einem Display für 4 x 20 Zeichen zeigen die Testprogramme Statusinformationen über den c't-Bot an. Das erleichtert die Fehlersuche ungemein.



Die untere Etage des c't-Bot ist den Sensoren und Motoren vorbehalten. Insgesamt vier Lichtschranken und ein optischer Maussensor beobachten den Boden.

Werkzeugkasten

Alle Demo-Programme und Bibliotheken für den c't-Bot sind in C geschrieben und

stehen als OpenSource unter der GNU General Public License (GPL) zur Verfügung. Auf Assembler-Code haben wir der Lesbarkeit und Verständlichkeit des Codes zuliebe verzichtet - auch wenn sich dadurch einiges optimieren ließe. Andere Programmiersprachen wie Basic, Pascal oder sogar Java wären theoretisch auch möglich. Letztere eignet sich aber für die direkte Ansteuerung von Sensoren und Aktuatoren nicht besonders.

Für die ATmega-Prozessoren gibt es eine Portierung des kostenlosen C-Compilers gcc. Obwohl mit WinAVR unter Windows eine komfortable und kostenlose Programmierumgebung für Atmel-Prozessoren rund um diesen Compiler existiert, haben wir uns entschieden, den Code mit Eclipse zu entwickeln. Diese ebenfalls frei erhältliche Umgebung lässt sich mit ein paar Handgriffen so einstellen, dass sie Code für den Mikrocontroller erzeugt und diesen sogar mit einem simulierten Mikrocontroller Schritt für Schritt im Debug-Modus ausführt. Der Vorteil von Eclipse besteht darin, dass sich in einer Entwicklungsumgebung sowohl c't-Bot-Code (Mikrocontroller und PC) als auch der Simulator c't-Sim programmieren lassen. Des Weiteren kann Eclipse direkt auf unser Code-Archiv (CVS) zugreifen und bei Bedarf Änderungen am Code als Patch-Dateien zusammenstellen.

Wenn Sie Ihrem c't-Bot Kunststücke beigebracht haben, die Sie mit anderen teilen möchten, schicken Sie uns einfach eine solche Patch-Datei. Auch um das Einspielen solcher Patches kümmert sich Eclipse.

Wer eigene Erweiterungen zur offiziellen Code-Basis hinzufügen möchte, sollte seinen Patch vorher gut testen und dokumentieren. Ein Patch sollte nach Möglichkeit die Struktur des Codes nicht verändern und sich auf die jeweils aktuelle Version im CVS-Repository beziehen. Damit jeder auch die ihm zustehenden Lorbeeren erhält, verewigt man sich selbst mit Doxygen-konformen @author-Tags in den geänderten Dateien und Funktionen [2]. So erscheint man dann auch in der HTML-Dokumentation auf den Projektseiten. Des Weiteren benötigen wir eine detaillierte Beschreibung des Patches sowie einen vorbereiteten Eintrag für die Changelog-Datei.

Eine Anleitung, wie man Eclipse überredet, Code für Mikrocontroller zu übersetzen und zu debuggen, findet sich auf der Projektseite [3]. Dort bieten wir auch gerne Patches an, bevor wir sie in den offiziellen Code übernehmen. Für Fragen, Anregungen oder Diskussionen um eigene Erweiterungen steht ein Leserforum bereit.

Quellen

Der Code für den Mikrocontroller erweitert den ct-Bot-Zweig der bereits in [2] vorgestellten Quellen um ein MCU-Unterverzeichnis. Die Präprozessor-Direktive `#define MCU` oder ihr Pendant `#define PC` legt fest, ob der Code auf dem realen c't-Bot laufen oder lieber auf dem PC seine Sensorwerte vom Simulator c't-Sim holen soll. In den vorgefertigten Eclipse-Repositories existieren dafür bereits verschiedene Build-Targets (Debug-xx-yyy), die unter anderem diese Direktiven enthalten. Die fertig übersetzten Binär-Dateien (.hex, .elf oder .exe) landen in den zum gewählten Target passenden Unterverzeichnissen (Debug-XXX).

Das Hauptprogramm befindet sich in der Datei ct-bot.c und ist nicht plattformabhängig. Alle Programmteile, die nur den Mikrocontroller betreffen, finden sich im Unterordner mcu, die für den PC im Verzeichnis pc. Weiteren Details des Codes widmet sich der nächste Hardware-Artikel der Serie.

Ungeduldige Naturen finden aber schon jetzt in der Funktion `bot_behave()` (bot_logik.c) die Verhaltenssteuerung des Roboters und in `bot_sens.h` die Namen aller Variablen, die die aktuellen Sensorwerte enthalten. Ihren Inhalt aktualisiert die Firmware automatisch im Hintergrund.

Ein ganz einfaches Programm zur Kollisionsvermeidung könnte wie folgt aussehen:

```
void bot_simple(Behaviour_t *data){
int16 speed_l_col, speed_r_col;
data->speed_l=BOT_SPEED_MAX;
data->speed_r=BOT_SPEED_MAX;
if (sensDistL < COL_NEAR)
speed_r_col=-speed_r-BOT_SPEED_NORMAL;
```

```

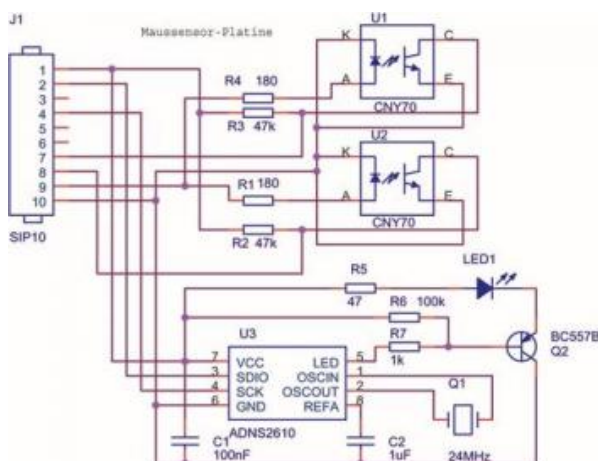
else speed_r_col=0;
if (sensDistR < COL_NEAR)
speed_l_col=-speed_l-BOT_SPEED_FAST;
else speed_l_col=0;
data->speed_l+=speed_l_col;
data->speed_r+=speed_r_col;
}

```

Im Demo-Code findet sich aber auch ein Beispiel für die Steuerung des Roboters per Fernbedienung.

Tuning

Die von uns vorgeschlagene Platzierung des Maussensors stellt einen Kompromiss zwischen Bodenfreiheit und Auflösung dar. Laut Datenblatt sollte die Linse des Maussensors (Kunststoffplatte) nicht weiter als 2,5 mm von der Oberfläche entfernt sein. Um das zu erreichen müsste man die beiden kleinen FR4-Streifen entfernen und die Linse festkleben sowie vielleicht sogar die beiden CNY70-Sensoren entfernen.



Da der Maussensor hinter der Drehachse des Roboters sitzt, lassen sich aus seinen Daten sowohl Translation als auch Rotation des Bots errechnen.

Wer bereit ist, etwas mit den Abständen zu experimentieren, der dürfte die Auflösung des Maussensors noch deutlich steigern können. Allerdings bleibt der Roboter dann auch leichter an Hindernissen hängen. Geländefreunde können die ganze Sensorplatine hingegen auch von oben einsetzen und erlangen so eine deutlich größere Bodenfreiheit.

Auch die Empfindlichkeit der Reflexlichtschranken CNY70 ist ein Eldorado für Tuner. Kleine Eingriffe beeinflussen die Erkennungsdistanz, die Genauigkeit oder die Reaktion auf

verschiedene Materialien. Der Vorwiderstand der LEDs steuert die Lichtmenge - je kleiner der Widerstand ist, desto mehr Licht strahlt der Sensor aus. Mehr Licht führt zu einer größeren Reichweite, aber auch zu einem stärkeren Übersprechen im Nahbereich. Weniger als 75 Ohm sollten die Vorwiderstände (R23 bis R27 auf der Hauptplatine sowie R1 und R4 auf der Mausplatine) nicht haben, da sonst zu viel Strom durch die LEDs fließt. Auch auf der Software-Seite lassen sich noch Umwelteinflüsse wie Streulicht herausrechnen, aber damit wird sich einer der folgenden Artikel beschäftigen.

Im nächsten Hardware-Artikel zum c't-Bot beleuchten wir die Ansteuerung der Sensoren und Aktuatoren des Roboters im Detail. Des Weiteren steht noch eine Mechanik zur Verriegelung des Transportfaches sowie eine ganze Reihe von Software-Themen auf unserer Roadmap. Dazu gehört unter anderem die Auswertung der Sensoren oder die Positionsberechnung des Roboters über den Maussensor aber auch Kartografie. Später wird es die Möglichkeit geben, das eigene Robotikkönnen mit anderen zu messen. (**bbe[3]**)

Literatur

[1] **Benjamin Benz, Carl Thiede, Thorsten Thiele, Spielgefährten, Roboter für Lötter, Simulator für Soft-Werker[4]**

[2] **Benjamin Benz, Peter König, Virtuelle Spielgefährten, Simulator für c't-Bots[5]**

[3] **Webseite zum c't-Bot-Projekt[6]**

[4] Dr. Till Harbaum, Blue Connection, Stereoton drahtlos per Bluetooth übertragen, Teil 2, c't 9/04 S. 208

[5] **Webseite zum c't-Netz-Schalter-Projekt[7]**

Soft-Link[8]

LC-Display	
Sensor	Kürzel
Distanz ¹	
Linie ¹	L
Licht ¹	P
Kante ¹	K
Rad-Encoder ¹	R
Transportfach	T
Fehler	F
Klappe	K
Maus ²	M
IR-Fernbedienung	I
¹ Der erste Wert bezieht sich auf den linken, der zweite auf den rechten Sensor ² Der erste Wert steht für die X-, der zweite für die Y-Komponente	

Funktionstest

LED	ct-Bot-test-analog.hex	ct-Bot-test-digital.hex	ct-Bot-test-motor.hex
rechts	Distanz rechts	Rad-Encoder rechts	Rad-Encoder rechts
links	Distanz links	Rad-Encoder links	Rad-Encoder links
rot	Linie links	Transportfach	Transportfach
orange	Linie rechts	Fehler	Fehler
gelb	Licht links	Klappe	Klappe
grün	Licht rechts	Maus dx	Maus dx
türkis	Kante links	Maus dy	Maus dy
weiß	Kante rechts	IR-Fernbedienung	IR-Fernbedienung

Stückliste	
Mechanikteile	
Grundplatte	ct-Robot/Grundplatte
Motorflansch	ct-Robot/Motorflansch
Getriebemotor	ct-Robot/Motor
Rad	ct-Robot/Rad
Reifen	ct-Robot/Reifen
Gleiter	ct-Robot/Gleiter
Alu-Träger	ct-Robot/Alu-Träger
Batteriehalter	3 x Mignon + 2 x Mignon
Hauptplatine	
C1, C2	22 pF
C3, C4	100 nF
C5	100 µF
D1, D2	1N4148
D3	SB140
IC1	ATmega32-16PU
IC2	L293D
IC3	74HC14N
IC4, IC5, IC6	74HC595N
IC7, IC8	LM311N
IC9	TSOP34836
IC10	L4940V5
J1, 2, 3, 4, 5, 6, 7, 8	Stiftleiste
LDR1, LDR2	MPY54C569
LED1, LED2, LED3	LED blau

LED4	LED rot
LED5	LED orange
LED6	LED gelb
LED7	LED grün
LED8	LED weiß
L1	100uH-SMCC
POT1	67W 0,5 kΩ
P1	DCBU2,1-PR
Q1	Q 16,0-LP
R1	10 kΩ
R2	20 Ω
R3, 4, 6, 8	4,7 kΩ
R5, 7, 29, 30, 31	47 kΩ
R9-R16	160 Ω
R17, R18, R32	39 kΩ
R19,R20	6,2 kΩ
R21,R22	470 kΩ
R23-26	180 Ω
R27	100 Ω
R28	6,8 Ω 2W
R30, R31	47 kΩ
R33	1 kΩ
R34	5,1 kΩ
R29 (siehe Text)	ZD2V4 0,5W
ST1-ST3, ST7-ST9	(Stecker+Buchse mit Kabel)
ST4, ST5, ST6	Stiftwanne
SW1	Taster
SW2	Kippschalter
TR1-TR6	BS250
U1	CNY70
Sensorplatinen links/rechts	
U101-U104	CNY70
U105	IS471F
LED101	LD274-3
Abstandssensor	GP2D12 + Kabel
Maussensor-Platine	

C1	100 nF
C2	1 µF
LED1	HLMP-ED80-K0T00
Klammer für LED	HDNS-2200
Linseplatte	HDNS-2100
Q1	Q24,0-LP/GW
Q2	BC557B
R1, R4	180 Ω
R2, R3	47 kΩ
R5	100 Ω
R6	100 kΩ
R7	1 kΩ
U1, U2	CNY70
U3	ADNS2610

URL dieses Artikels:

<http://www.heise.de/-290314>

Links in diesem Artikel:

- [1] <http://www.ct-bot.de>
- [2] <http://www.heise.de/ct/projekte/machmit/ctbot/wiki/ct-Bot-Hardware>
- [3] <mailto:bbe@ct.de>
- [4] <https://www.heise.de/ct/artikel/Spielgefaehrten-290274.html>
- [5] <https://www.heise.de/ct/artikel/Virtuelle-Spielgefaehrten-290294.html>
- [6] <https://www.heise.de/ct/artikel/c-t-Bot-und-c-t-Sim-284119.html>
- [7] <https://www.heise.de/ct/artikel/Netz-Schalter-284123.html>
- [8] <http://www.heise.de/ct/06/04/links/208.shtml>

Copyright © 2006 Heise Medien