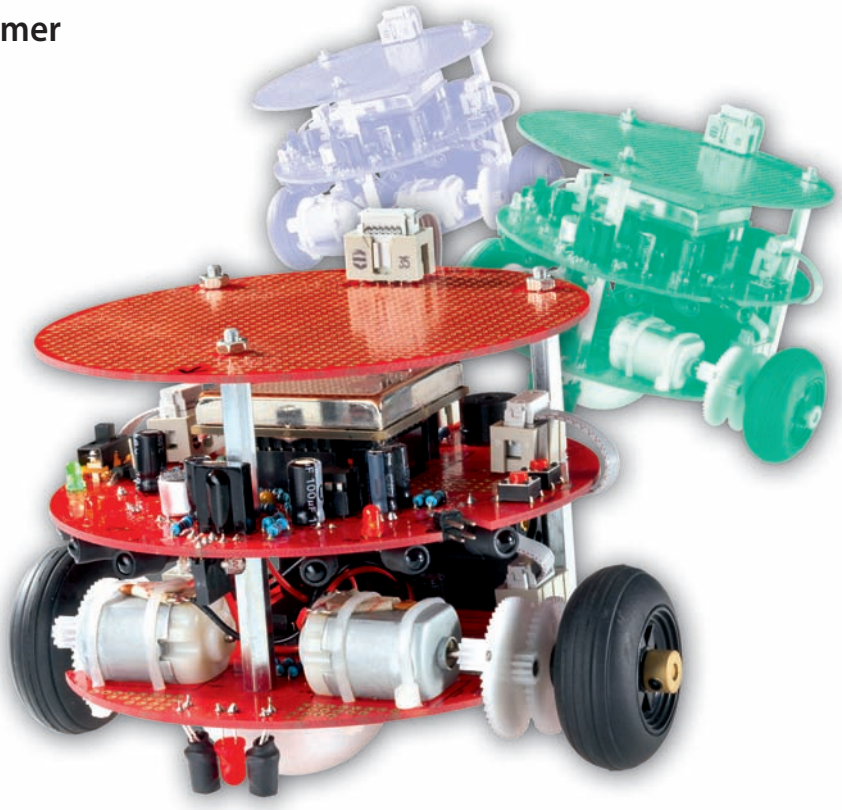
FRANZIS
PC+ELEKTRONIK

Ulli Sommer



PRO-BOT128 selbst bauen und erfolgreich einsetzen

Mikrocontroller-Technik zum Anfassen

Auf CD-ROM:

- IDE (Integrierte Entwicklungsumgebung)
- Alle Beispielprogramme zum Buch in C-Compact & Basic
- PC-Steuerungssoftware für PRO-BOT128 (Open Source)
- Anleitungen zu den Produkten
- Datenblätter

Vorwort

Roboterbauen wird, wie man in den verschiedensten Foren und Fachzeitschriften beobachten kann, in den letzten Jahren immer populärer. Das liegt daran, dass Mikrocontroller und zusätzliche Peripheriebausteine immer günstiger angeboten werden und auch an den Schulen zunehmend die Themen Mikrocontroller, Computer und Robotik unterrichtet werden.

Auch Conrad Electronic bietet seit einigen Jahren unter der Bezeichnung „C-Control" verschiedene Mikrocontroller- und Robotersysteme mit umfangreichem Zubehör zu günstigen Preisen an. Deshalb ist es nicht verwunderlich, dass das neueste Robotersystem *PRO-BOT128* eine C-Control Pro Mega128 als Steuer- und Recheneinheit verwendet. Die C-Control-Pro-Serie ist die jüngste und leistungsstärkste ihrer Klasse und basiert auf einem Atmel-AVR-Mikrocontroller der ATmega-Serie. Der Roboter kann mit der mitgelieferten, sehr luxuriösen und vor allem kostenlosen Entwicklungsumgebung wahlweise in C, Basic und Assembler programmiert werden. Dadurch ist das System für Einsteiger sowie für den erfahrenen Entwickler sehr gut geeignet. Der Einsteiger wird mit Basic seine Ziele und Wünsche schnell verwirklichen können, der Profi hingegen kann in Verbindung mit Assembler das Maximale aus der C-Control Pro „herauskitzeln".

Auf dem Gebiet der Robotik ist aber für viele Einsteiger nicht nur das Programmieren ein Steckenpferd, sondern auch die Elektronik und Mechanik eines Roboters. Für diese Gruppe wird der Roboter auch als Fertiggerät angeboten und man kann sofort mit der Programmierung beginnen. Wer aber dennoch andere Sensoren und Aktoren anbringen möchte, wird trotz allem zum Lötkolben greifen müssen. Um diese Hürden zu nehmen, werden in den kommenden Kapiteln einige Hardware-Erweiterungen mit Nachbauanleitung vorgeschlagen. Natürlich wurden alle Erweiterungen und Experimente mit äußerster Sorgfalt getestet.

Ich wünsche Ihnen viel Spaß beim Lesen und Basteln.

Ulli Sommer, Waidhaus, März 2009

C-Control

Inhaltsverzeichnis

1 CD-ROM zum Buch

Diesem Buch liegt eine CD-ROM bei, die verschiedene Programme, Tools und Beispiele enthält. Die CD-ROM erleichtert das Arbeiten mit diesem Buch. Die hier abgedruckten Beispiele sind auf der CD-ROM enthalten.

1.1 Inhalt der CD-ROM

- C-Control PRO-Entwicklungsumgebung (IDE)
- Originalbeispiele und -anleitungen
- Beispielprogrammcode zum Buch
- Diverse Tools
- Datenblätter
- Schaltpläne

1.2 GPL (General Public License)

Sie können Ihre eigenen Programme selbstverständlich mit anderen Anwendern über das Internet austauschen. Die Beispielprogramme stehen unter der Open-Source-Lizenz *GPL* (General Public License). Daher sind Sie berechtigt, die Programme unter den Bedingungen der GPL zu modifizieren, zu veröffentlichen und anderen Anwendern zur Verfügung zu stellen, sofern Sie Ihre Programme dann ebenfalls unter die GPL-Lizenz stellen.

1.3 Systemvoraussetzung

Um die Programme auf der mitgelieferten CD-ROM nutzen zu können, benötigen Sie einen PC mit mindestens einem Pentium-4-Prozessor oder höher und mindestens 512 MB Arbeitsspeicher ab Windows 2000 aufwärts. Auf eine besondere Grafikkarte kann verzichtet werden, da die Programme keine großen Ansprüche stellen. Dadurch kann man einen Rechner mit Onboard-Grafikkarte ohne Weiteres verwenden. Ein VGA-Monitor mit mindestens 17“ ist empfehlenswert.

Die meisten Programme benötigen die oben genannte Hardware zwar nicht, sollte man sich jedoch auf Visual Basic Express (Visual Studio) einarbeiten, das auf dem Microsoft Framework basiert, stößt man schnell an die Grenzen des PC. Das Arbeiten

wird dann unerträglich langsam und das Kompilieren dauert zu lange, um Spaß daran haben zu können.

Zudem sollten Sie im Besitz eines Internetanschlusses sein, um diverse Updates und die Registrierung der Programme durchführen zu können.

5 Mikrocontroller-Programmierung

Mit der zunehmenden Integration von Halbleiterbauteilen wie Mikroprozessoren hielten Mikrocontroller immer mehr Einzug in die Anwendungsgebiete der Mess-, Steuer- und Regelungstechnik. Aber auch im Hobbybereich wurden die Mikrocontroller immer beliebter. Das liegt zum einen daran, dass heute komplexe, meist analoge Schaltungen durch einfachere digitale Mikrocontroller-Schaltungen ersetzt werden. Ein anderer ausschlaggebender Punkt ist heute das unschlagbare Preis-Leistungs-Verhältnis von Mikrocontrollern.

5.1 Was ist ein Programm?

Ein Programm ist die Beschreibung eines Informationsverarbeitungsprozesses. Im Laufe eines solchen Prozesses wird aus einer Menge von variablen oder konstanten Eingangswerten eine Menge von Ausgangswerten berechnet. Die Ausgangswerte sind entweder selbst Ziel der Informationsgewinnung oder dienen mittelbar zur Reaktion auf die Eingangswerte. Neben den eigentlichen Berechnungen kann ein Programm Anweisungen zum Zugriff auf die Hardware des Computers oder zur Steuerung des Programmflusses enthalten. Ein Programm besteht aus mehreren Zeilen sogenannten Quelltextes. Dabei enthält jede Zeile eine oder mehrere Rechen- oder Steueranweisungen. Außer diesen Anweisungen selbst bestimmt ihre Reihenfolge ganz wesentlich die eingangs beschriebene Informationsverarbeitung. Die Ausführung der den Anweisungen entsprechenden Operationen durch den Steuercomputer erfolgt sequenziell, also nacheinander. Eine Folge von Programmanweisungen mit einem bestimmten Ziel nennt man auch *Algorithmus*.

5.2 Programmierung in Basic

Basic ist eine einfach zu erlernende Programmiersprache. Für die C-Control Pro wurde ein Basic-Dialekt angelehnt an Visual Basic von Microsoft entwickelt, dadurch ist es ein Leichtes für einen Visual-Basic-Programmierer, auch eine Hardware wie z. B. unseren PRO-BOT128 zu programmieren. Basic ist besonders bei Anfängern beliebt und wird im Bereich der Mikrocontroller-Anwendungen, aber auch für professionelle Anwendungen, eingesetzt.

5.2.1 Konzept von Basic

Der hier betrachtete Basic-Dialekt ist modular aufgebaut. Dafür definiert man ein oder mehrere Unterprogramme, die verschiedenste Befehlsfolgen kapseln. Die Abarbeitung der Befehle erfolgt weitestgehend intuitiv. Das bedeutet, dass man den Controller durch einfache Befehle dazu anweist, bestimmte Aufgaben zu erfüllen. Nebenbei bemerkt, gibt es insbesondere für Computer, die mit Windows oder Linux arbeiten, auch objektorientierte Basic-Dialekte, die, im Gegensatz zu den hier gezeigten Programmiersprachen, weniger intuitiv bedient werden.

5.2.2 Vor- und Nachteile von Basic

Ein entscheidender Vorteil von Basic ist, dass ein schneller Einstieg in die Welt der Mikrocontroller ermöglicht wird. Der hier thematisierte Basic-Dialekt ist so ausgelegt, dass er die Hardware optimal und effizient anspricht und der Entwickler trotzdem übersichtlich strukturierten Quellcode schreiben kann.

5.3 Programmierung in C

C ist eine der ältesten und am weitesten verbreiteten Programmiersprachen. Sie ist auf nahezu jeder Plattform implementiert oder verfügbar. Die Programmiersprache C entstand in einer Zeit, in der man versuchte, eine etwas abstraktere Programmierung als die bis dahin üblichen Maschinencode- und Assemblerprogramme einzuführen. C besteht deshalb im Wesentlichen auf einfachen Sprachelementen wie If-Verzweigungen oder einfachen Schleifen und Funktionen.

5.3.1 Konzept von C

Die Programmiersprache C wurde aus zwei wesentlichen Gründen eingeführt: Man suchte nach einer möglichst plattformunabhängigen Programmiersprache, damit man nicht für jede neue Prozessorgeneration komplette Entwicklungen und Programme neu schreiben musste. Außerdem suchte man nach einer Möglichkeit, die sehr maschinennahe Assemblersprache durch eine modulare und übersichtlichere Entwicklungsumgebung zu ersetzen. Das für die C-Control Pro entwickelte C-Compact ist eine „abgespeckte“ Version von ANSI-C. Grund dafür ist, dass der Einsteiger sowie auch der fortgeschrittene Entwickler sehr schnell ans Ziel kommt, ohne große Datenblätter der Mikrocontroller wälzen zu müssen.

5.3.2 Vor- und Nachteile von C

Die Vorteile von C gegenüber Basic sind, dass man mit C-Control Pro C die Grundlagen der C-Programmierung erlernt und dieses Wissen später auch auf anderen Systemen einsetzen kann. Zudem wird C bereits in vielen Schulen gelehrt und daher ist es auch für einen Einsteiger, der den C-Grundkurs in der Aus- oder Weiterbildung besucht, ein Leichtes, die C-Control Pro zu programmieren. Wenn man jedoch die beiden Hochsprachen Basic und C vergleicht, wird Basic in puncto Übersichtlichkeit gewinnen. Auf der C-Control Pro werden beide Programmiersprachen annähernd gleich schnell ausgeführt und durch die Programmierung in Basic entstehen keine Nachteile.

5.4 Programmierung in Assembler

Die C-Control Pro kann seit Anfang 2009 auch in Assembler programmiert werden. Ein Assembler setzt ein Assemblerprogramm in der Regel 1:1 in Maschinencode um, also in Befehle, die direkt vom Mikroprozessor abgearbeitet und ausgeführt werden können. Der Programmierer muss dabei über ein fundiertes Wissen über die Systemarchitektur des Mikroprozessors verfügen. Da es eine Fülle von Registern, Befehlen und Adressierungsarten gibt, fällt es insbesondere Anfängern schwer, sich in die Assemblerprogrammierung einzuarbeiten. Hinzu kommt, dass jede Mikroprozessor-Generation und jeder Typ über einen anderen Befehlssatz verfügen. Selbst jemand, der sich gut mit dem Intel-Assembler für PCs auskennt, benötigt eine lange Einarbeitungszeit, um auch beispielsweise für die C-Control Pro Assemblerprogramme zu entwickeln.

Der große Vorteil von Assembler ist, dass man über nahezu jedes Bit und jede Operation frei verfügen kann. Damit können sehr leistungsstarke und dennoch schlanke Programme entwickelt werden, sofern man sich mit den Spezialbefehlen des Mikroprozessortyps auskennt. Manche Anwendungen lassen sich aus Gründen der Schnelligkeit nur in Assembler verwirklichen. Heute setzt man Assembler häufig nur in kleinen Programmstücken ein, wo ein Höchstmaß an Leistung gefordert wird. Der Aufruf des Programms erfolgt dann aus einer Hochsprache wie C oder Basic. Auch die Auswertung der Ergebnisse des Assemblerprogramms kann von einer Hochsprache übernommen werden.

5.5 Der Interpreter

Auf der C-Control Pro wird das Basic bzw. C-Compact-Anwenderprogramm vom Compiler in Byte-Code übersetzt. Der Symbolcode entspricht dabei einem bestimmten Befehl (z. B. dem Abspeichern eines Messwerts des A/D-Wandlers in eine Variable). Würde man einen Byte-Code-Befehl in die Befehle übersetzen, die der Mikrocontrol-

ler eigentlich verarbeitet (die sogenannten Maschinenbefehle) würde man sehen, dass der Bytecode in der Regel einer Vielzahl von Maschinenbefehlen aufruft. Das ist ein entscheidender Vorteil von Byte-Code-basierenden Programmen, da sie in der Regel weniger Speicher belegen als Programme, die aus Maschinencode bestehen. Hinzu kommt, dass Maschinencode in der Regel viel komplizierter strukturiert ist als Byte-Code. Ein Nachteil von Interpretern ist, dass die Abarbeitung des Codes in der Regel etwas langsamer erfolgt als die Abarbeitung von Maschinenbefehlen. Außerdem kann man mit Maschinenbefehlen viel effizientere Programme schreiben. Für die meisten Anwendungen spielt dieser Nachteil jedoch eine untergeordnete Rolle.

6 Die Hardware der C-Control Pro Mega128

Das Herzstück des C-Control-Pro-Steuercomputers ist ein moderner Ein-Chip-Mikrocontroller (MCU = Micro Controller Unit) der Firma Atmel von der Baureihe ATmega AVR, der in HCMOS-Technologie geliefert wird und vielseitig verwendbar ist.

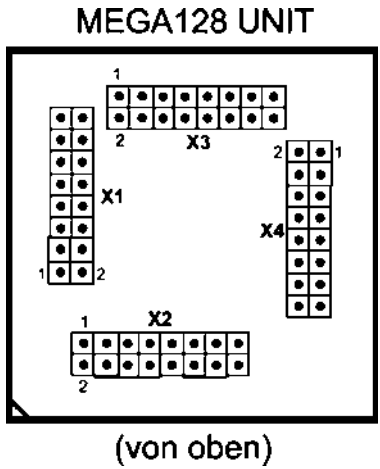


Abb. 6.1: Pinbelegung der C-Control Pro Mega128, die meisten Pins besitzen Doppelbelegungen. Die ausführliche Belegung können Sie in der Port-Übersichtstabelle der Anleitung einsehen.

6.1 Taktfrequenz

Damit die C-Control Pro ein Programm ausführen kann, wird ein Oszillator benötigt. Dieser ist bereits im AVR integriert und benötigt nur noch einen externen Quarz, um zu arbeiten. Getaktet wird die C-Control Pro Mega128 mit 14,7456 MHz. Der Grund für diese „krumme“ Frequenz ist, dass hier der Baudrate-Fehler am kleinsten ist. Der verbaute Mikrocontroller ATmega128 16-AU kann zwar bist 16 MHz betrieben werden, jedoch entsteht dadurch ein größerer Baudrate-Fehler. Mit der verwendeten Quarzfrequenz und dem sehr gut optimierten Interpreter können dadurch bis zu 160.000 Instruktionen pro Sekunde abgearbeitet werden, was für die vorgesehen Anwendungsrahmen in der Steuer-, Mess- und Regelungstechnik ein ausgezeichneter Wert ist.

6.2 Speicher

Der Flash-Speicher umfasst 128 kB, von denen jedoch durch das Betriebssystem (Interpreter, Multithreading usw.) noch etwa 110 kB zur Verfügung stehen. Dieser Speicher reicht in der Regel auch für komplexe Systeme vollkommen aus. Vom Anwender sind ca. 4 kB User-RAM als Variablenspeicher nutzbar. Zudem enthält die Mega128 noch ein 2 kB großes EEPROM, das zur Datenaufzeichnung verwendet werden kann.

6.3 Gehäuse

Auf der Unit selbst befinden sich neben den ATmega128 noch der Quarzoszillator und diverse Abblockkondensatoren sowie die Reset-Beschaltung. Um Stör-Ein-/Ausstrahlung zu vermeiden, wurde der Unit eine Metallhaube aufgesetzt. Das Gehäuseformat wurde zwecks besserer Handhabung auch für den Bastler im Rastermaß 2,54 mm gewählt. Die Unit besitzt vier Präzisionsstiftleisten mit je 2 x 8 Pins. Damit diese nicht verkehrt herum eingesteckt werden können, wurden sie versetzt angeordnet. Das macht eine Verpolung fast unmöglich.

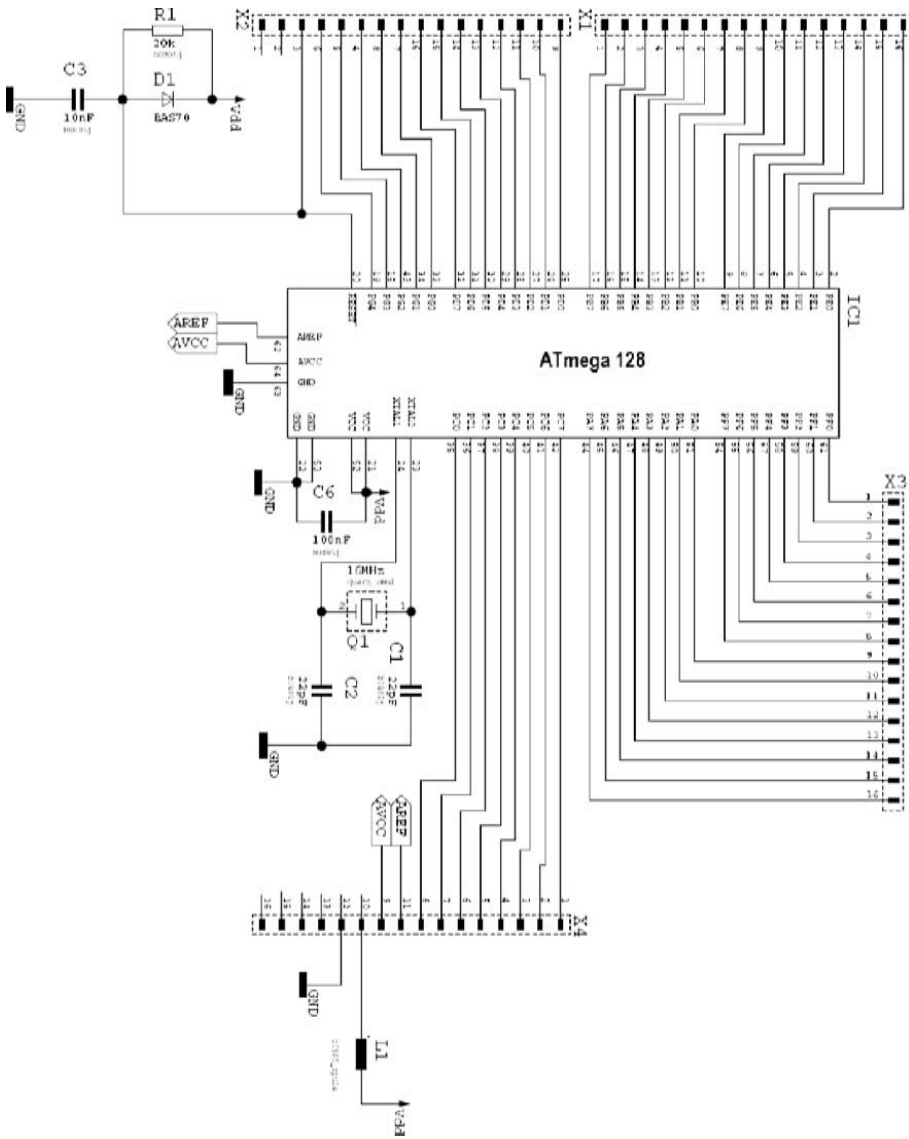


Abb. 6.2: Innenbeschaltung der C-Control Pro Mega128 Unit

6.4 I/O-Ports

Es sind 53 I/O-Ports herausgeführt (Port A bis Port G), die man unabhängig voneinander als Ein- oder Ausgang, als Sonderfunktionen wie PWM, Zählereingang und Spannungsmessung (ADC) frei programmieren kann. Die Ports können in Summe maximal 200 mA liefern bzw. gegen Masse ziehen. Ein einzelner Pin kann maximal +/–20 mA liefern/ableiten. Bei der Beschaltung ist also unbedingt darauf zu achten, dass diese Daten nicht überschritten werden können. Sollte man noch im Experimentierstadium sein, empfiehlt es sich, einen sogenannten *Angstwiderstand* (Strombegrenzungswiderstand auf max. zulässigen Strom) einzubauen, der den Strom bei einem evtl. Kurzschluss auf die Maxima begrenzt.

6.5 ADC (Analog Digital Converter)

Damit wir analoge Spannungen messen und erfassen können, besitzt die C-Control Pro einen internen Analog-Digital-Wandler (ADC = Analog Digital Converter) mit einer Auflösung von 10 Bit. Das heißt für uns, ein „Step" also eine konvertierte Zahl (z. B. 1), hat bei einer ADC-Auflösung von 10 Bit einen analogen Spannungswert von gerundet 0,0025 Volt = 2,5 mV. Beim PRO-BOT128 wird die interne Referenzspannung von 2,56 V verwendet.

Das kann man mit dieser einfachen Formel leicht selbst nachrechnen:

***Ustep** = Uref/Auflösung*
***Ustep** = 2,56 Volt / 1.024 (0 bis 1.023 = 1.024 Steps)*
***Ustep** = 0,0025 Volt*

Will man den digitalen angezeigten Wert wissen, benutzt man am besten folgende Formel:

Anliegende Spannung (ADC) = (Uref/1024) × (ADC-Wert)

6.6 DAC (Digital Analog Converter)

Die meisten in der Natur vorkommenden Signale sind analoger Natur. Daher ist es für eine digital arbeitende Maschine notwendig, die digitalen Werte in analoge Größen umzusetzen, wenn externe Vorgänge beeinflusst werden sollen. Ein anderes Beispiel ist die Ansteuerung einer H-Brücke für eine Motorsteuerung. Hier wird die Motordrehzahl über die Puls-/Pausenverhältnisse geändert. In Anwendungen zur pulsweitenmodulierten Digital-Analog-Umsetzung werden Zeitbasis und Periodenlänge einmalig eingestellt und dann wird nur der Ausgabewert verändert. Alternativ können diese

Ausgänge auch zur Modellbau-Servo-Ansteuerung genutzt werden. Auch eine Tonausgabe ist über die DACs möglich. Wir nutzen beim PRO-BOT128 die PWM-Kanäle zur Ansteuerung der Motoren über eine H-Brücke und zur Takterzeugung für die Infrarot-LEDs sowie zur Tonausgabe über den kleinen Buzzer.

Benötigen wir eine analoge Spannung, muss ein RC-Glied an den Ausgang des PWM-Ports angeschlossen werden, der aus den PWM-Signal eine „quasi“ analoge Spannung formt.

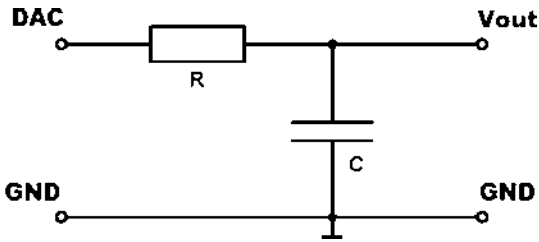


Abb. 6.3: Aufbau eines typischen RC-Glieds für einen PWM-Ausgang

Das RC-Glied berechnet sich folgendermaßen:

$$FG = \frac{1}{2 * Pi * R * C}$$

FG = Grenzfrequenz des PWM-Signals
Pi = 3,14 (grob)
R = Widerstand in Ohm
C = Kondensator in Farad

Das RC-Glied darf in diesem Zustand natürlich nicht sonderlich belastet werden, da sich sonst der Kondensator zu schnell entladen würde und dadurch das RC-Glied überflüssig wäre. Am besten, man setzt noch eine Treiberendstufe hinter das RC-Glied, die dass Signal auf den gewünschten Level anhebt und den erforderlichen Strom abgeben kann.

6.7 UART (Universal Asynchronous Receiver Transmitter)

Die Mega128 ist mit zwei seriellen Hardware-Schnittstellen ausgestattet (UART). Über diese lässt sich die Unit programmieren oder kann zur Datenausgabe auf ein Terminalprogramm oder eine eigene Software verwendet werden. Die Baudrate (Übertragungsgeschwindigkeit) ist wieder in der Software einzustellen und variiert zwischen 1.200 Bps bis 230.400 Bps. Zur Programmierung der Unit werden 38.400 Baud verwendet.

6.8 Interrupt-Eingänge (IRQ)

Es besteht die Möglichkeit, das Programm zu unterbrechen, sobald ein Ereignis an einen IRQ-Pin auftritt. Dies kann eine steigende oder fallende Flanke oder jeder Flankenwechsel sein. Wie der IRQ regieren soll, kann auch hier wieder in der Software festgelegt werden. Wichtig ist zu wissen, dass mit solch einem Eingang das Programm an jeder Stelle unterbrochen werden kann, um in die vorgegebene IRQ-Routine zu springen und den dort angegebenen Code abzuarbeiten. Ein typisches Einsatzgebiet für einen IRQ-Port sind schnelle Zähleingänge oder Notaus-Taster (Schalter). Die IRQ-Routine wird auch *ISR* (Interrupt Service Routine) genannt. Die Interrupt-Pins der Mega128 sind PD.0 bis PD.3 und PE.4 bis PE.7, somit insgesamt 8 IRQ-Eingänge. Diese Möglichkeit nutzt der PRO-BOT128 für den Taster SW2 und um seine Odometer Sensoren auszulesen.

6.9 Der Bootloader

Da der Roboter den seriellen Bootloader-Modus der C-Control Pro verwendet, ist auch ein Boot-Taster am Roboter vorgesehen. Wird der Roboter mit gedrückter Boot-Taste eingeschaltet, springt dieser in die Bootloader-Routine und eine Programmübertragung kann zwischen IDE und Roboter stattfinden.

6.10 Stromversorgung der Mega128

Die Stromversorgung trägt maßgeblich zum sicheren Betrieb der Unit bei. Die C-Control Pro Unit kann mit einer Gleichspannung zwischen 4,0 und 5,5 Volt versorgt werden. Der Roboter besitzt für Akku- oder Batteriebetrieb einen Jumper *JP1,* der bei der Verwendung von Batterien abgezogen werden muss.

7 Das Robotersystem PRO-BOT128

Der Roboter PRO-BOT128 besteht aus drei Platinenebenen, die Platz für den Antrieb, die Controllereinheit mit Sensorik sowie eine Ebene bieten, die für eigene kreative Erweiterungen genutzt werden kann. Inspiriert wurde der Entwickler des PRO-BOT128 vom allseits beliebten *ASURO,* der seit einigen Jahren bereits erfolgreich auf dem Markt ist. Der Antrieb wurde dank dem Einverständnis des DLR (Deutsches Zentrum für Luft- und Raumfahrt) von ASURO übernommen. Im Gegensatz zum ASURO besitzt der PRO-BOT128 wesentlich mehr Sensoren und Erweiterungsmöglichkeiten. Die Programmierung kann in Basic, C oder Assembler geschehen, was auch dem nicht so geübten „Programmierer" dank Basic zugute kommt. Auch im Vergleich des eingesetzten Controllers ist der PRO-BOT128 mit seinen 110 kB Flash ein wahrer Speicherriese. Dank Multithreading-Fähigkeit der C-Control Pro ist es kein Problem, auch sehr komplexe und leistungsfähige Programme zu entwickeln. Der differenzielle Antrieb ermöglicht es, auf der Stelle zu drehen, und benötigt dadurch nur sehr wenig Platz zum Wenden. Das Robotersystem wird einmal als Bausatz oder als Fertiggerät geliefert. Das ermöglicht auch dem handwerklich nicht ganz so begabten Anwender den Zugang in die Robotik. Der PRO-BOT128 wird über vier Mikrobatterien oder Akkus mit Energie versorgt. Zu empfehlen sind aber Akkus, da diese über die eingebaute Ladebuchse gleich wieder aufgeladen werden können, ohne dass sie umständlich wieder aus dem Batteriehalter entfernt werden müssten.

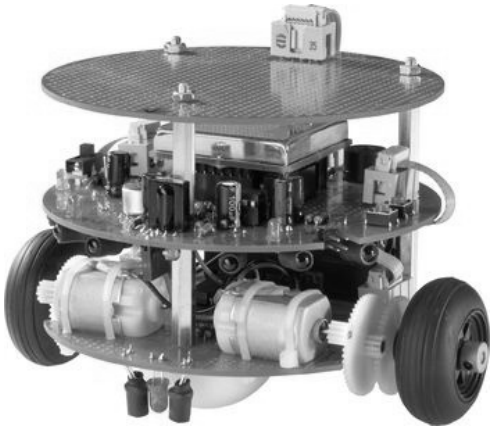


Abb. 7.1: Das Robotersystem PRO-BOT128 – fertig aufgebaut

7.1 Vorhandene Sensoren und Aktuatoren

7.1.1 Sensoren

- 2 Lichtsensoren
- 2 Wegstreckenmesser *Odometer*
- 1 Liniensensor
- 1 berührungsloses Infrarot-Anti-Kollisionssystem (ACS)
- 1 Schallsensor
- 1 Sensor für Betriebsspannung

7.1.2 Aktuatoren

- 2 DC-Elektroantrieb mit stufenloser Geschwindigkeitseinstellung (Vor/Rück)
- 1 Piepser zur Tonausgabe
- 4 Status-LEDs
- 1 Line-LED für Liniensensor

7.2 Erweiterungen

- 64 K I²C-EEPROM
- Experimentierplatine

Achtung! Beim Kauf muss man beachten, dass man nur den Roboter ohne C-Control Pro und Programmierkabel (Voltcraft USB-Programmer) erwirbt. Die beiden Zubehörartikel müssen noch zusätzlich erworben werden. Trotz allem liegt der Preis für den verhältnismäßig gut ausgestatteten Lernroboter weit unter 100 €.

8 Die Elektronik des Roboters

Im folgenden Kapitel wird die Elektronik des Roboters genauer unter die Lupe genommen. Wer den Bausatz erworben hat, wird sofort bemerken, dass kein einziges SMD-Bauteil vorhanden ist. Grund für die THT-Technik (eng. through-hole technology), also die Bestückung mit bedrahteten Bauteilen, ist die leichte Handhabung bei der Bestückung und die einfache Reparaturmöglichkeit. Sollte doch einmal ein Bauteil beim Experimentieren „abrauchen", ist es ein Leichtes, es zu ersetzen. In der mitgelieferten Anleitung werden alle Teile bebildert gezeigt. Im Anhang findet man die Stückliste zur Nachbestellung defekter Bauteile.

8.1 Stromversorgung

Nach dem Schalter SW1 folgt eine Diode, die mit einen Jumper *JP1* überbrückt werden kann. Die Diode sorgt bei abgezogenem Jumper dafür, dass eine Spannung von ca. 0,7 Volt abfällt. Das ist sinnvoll, wenn z. B. Batterien verwendet werden, denn diese besitzen im Neuzustand eine höhere Leerlaufspannung als Akkus. Das ergibt bei vier Batterien in Reihe genau 6 Volt (4 x 1,5 V), was aber für den µC (Mikrocontroller) bereits zu viel ist. Die maximale Betriebsspannung der C-Control Pro beträgt 5,5 Volt. Durch den Einsatz der Diode erhält man an C1 eine Spannung von ca. 5,3 Volt.

Immer wenn Batterien verwendet werden: Den Jumper JP1 abziehen!

Der Kondensator *C1* dient als Puffer (Energiespeicher) für kurzzeitige Stromspitzen. Der Widerstand *R8* dient als Strombegrenzung für die LED (Leuchtdiode) *D5*. Er sorgt dafür, dass maximal 20 mA durch die Diode fließen können. Die beiden Widerstände *R9* und *R10* sind als Spannungsteiler geschaltet, um die Betriebsspannung zu messen. *C2* dient wieder als Puffer, um schnelle Spannungsschwankungen am ADC zu unterbinden. Da wir die Spannung nach der Diode messen, müssen wir zum Messwert die abgefallene Spannung im Programmcode dazuaddieren. Da Dioden wie andere elektronische Bauteile auch einer Toleranz unterliegen, können wir die Spannung auch mit einem Multimeter (Spannungsmesser) ermitteln, indem wir die reale Spannung über die Diode messen. In der Software ist ein Kalibrierfaktor vorgesehen, der sich aus dem Spannungsteiler und den Spannungsabfall über die Diode zusammensetzt.

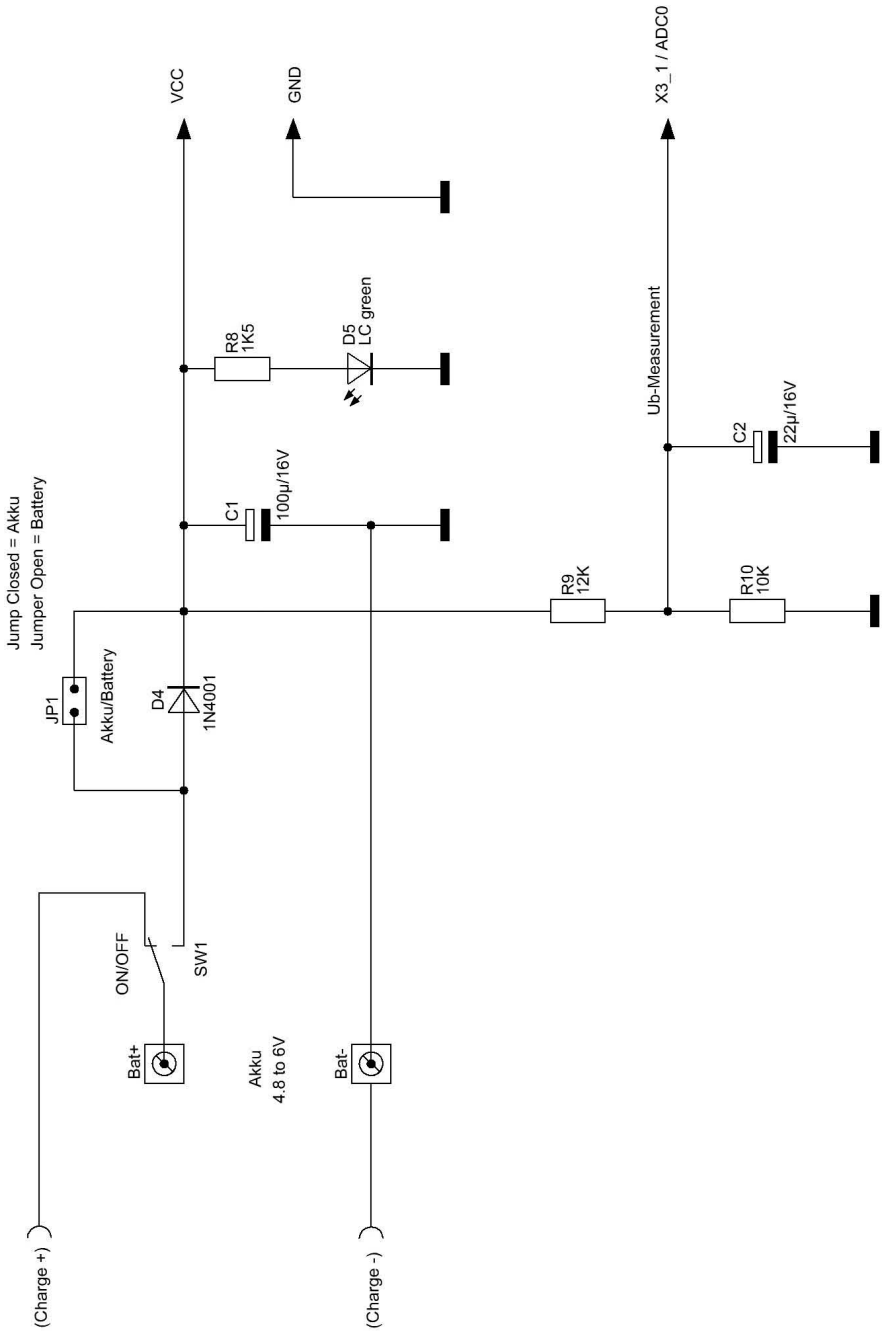


Abb. 8.1: Stromversorgung und Betriebsspannungsmessung

8.2 Controller

Das eigentliche Herzstück des Roboters ist die C-Control Pro Mega128. Sie ist auf der mittleren Ebene des Roboters angeordnet und die Verbindungen zwischen den einzelnen Platinen geschehen mittels Flachbandkabeln, die mit einem 6-poligen Pfostenstecker versehen sind. An der C-Control Pro ist zusätzlich zur Datenspeicherung (Messwerte, Programme usw.) ein I²C-EEPROM mit 8K x 8 angeschlossen. Wenn wir uns die Platinen genauer ansehen, werden wir die kleinen Brücken in der Nähe des EEPROM sehen. Diese dienen zur Adressenänderung des Speichers. Sollten wir ein anderes I²C-Bauteil mit der gleichen Adresse verwenden, können durch das Auftrennen der Leiterbahn andere Adressen vergeben werden (siehe Datenblatt zum EEPROM). Die Widerstände *R22* und *R23* sind die I²C-Bus-Pull-up-Widerstände. Die C-Control Pro wird beim PRO-BOT128 im seriellen Bootloader-Modus betrieben und verwendet daher die UART-Anschlüsse (TxD und RxD) der Unit. Die Bauteile *L1*, *C5* und *C6* dienen zur ADC-Beschaltung und Unterdrückung von Störungen am ADC. Auch die Status-LEDs werden über je einen Widerstand von 1,5 KOhm strombegrenzt.

Der Widerstand *R12* am Mikrolautsprecher dient zur Strombegrenzung des Ausgangsports der Unit. Schöner Nebeneffekt ist, dass die Lautstärke des Minilautsprechers etwas minimiert wird.

8.3 Motortreiber

Die Motoren des Roboters werden über eine elektronische H-Brücke angesteuert. Eine H-Brücke besteht in der Regel aus vier Transistoren (oder MOSFETs), von denen, je nach Drehrichtung des Motors, immer zwei leitend sind. Die unten abgebildete Grafik zeigt, dass man den Motor mit den Schaltern *S1* und *S4* vorwärts und mit den Schaltern *S3* und *S2* rückwärts laufen lassen kann. Im *IC1* ist solch eine H-Brücke, die bis zu 600 mA Strom liefern kann, zweimal integriert. *C4* stützt dabei wieder die Betriebsspannung. Bei zu hochohmigen Akkus oder Batterien kann es vorkommen, dass sich Störungen durch Spannungseinbrüche bemerkbar machen. Sollte das der Fall sein, muss *C4* vergrößert werden (220 bis 2.200 µF). Das IC2 ist ein 4-fach-NAND-Schmitt-Trigger, der für die Aufteilung des PWM-Signals und für die Fahrtrichtungsauswertung zuständig ist. Ziel ist es, über die PWM (Pulsweitenmodulation) die Drehzahl und die Drehrichtung zu beeinflussen. Das PWM-Signal kann bei der aktuellen Software zwischen 1 und 255 eingestellt werden. Bei einer PWM vom 128, was einem Tastverhältnis von 50 % zwischen Signal und Pause entspricht, stoppen die Motoren. Wird das PWM-Signal Richtung 255 erhöht, drehen die Räder vorwärts, bei Verringerung auf kleinere Werte als 128 drehen die Räder rückwärts. Der CD4093 teilt also das PWM-Signal mehr oder weniger für uns auf. Der Widerstand *R4* zieht den Enable-Eingang des L293 auf Masse. Über die Software muss Enable auf +5 V gelegt werden,

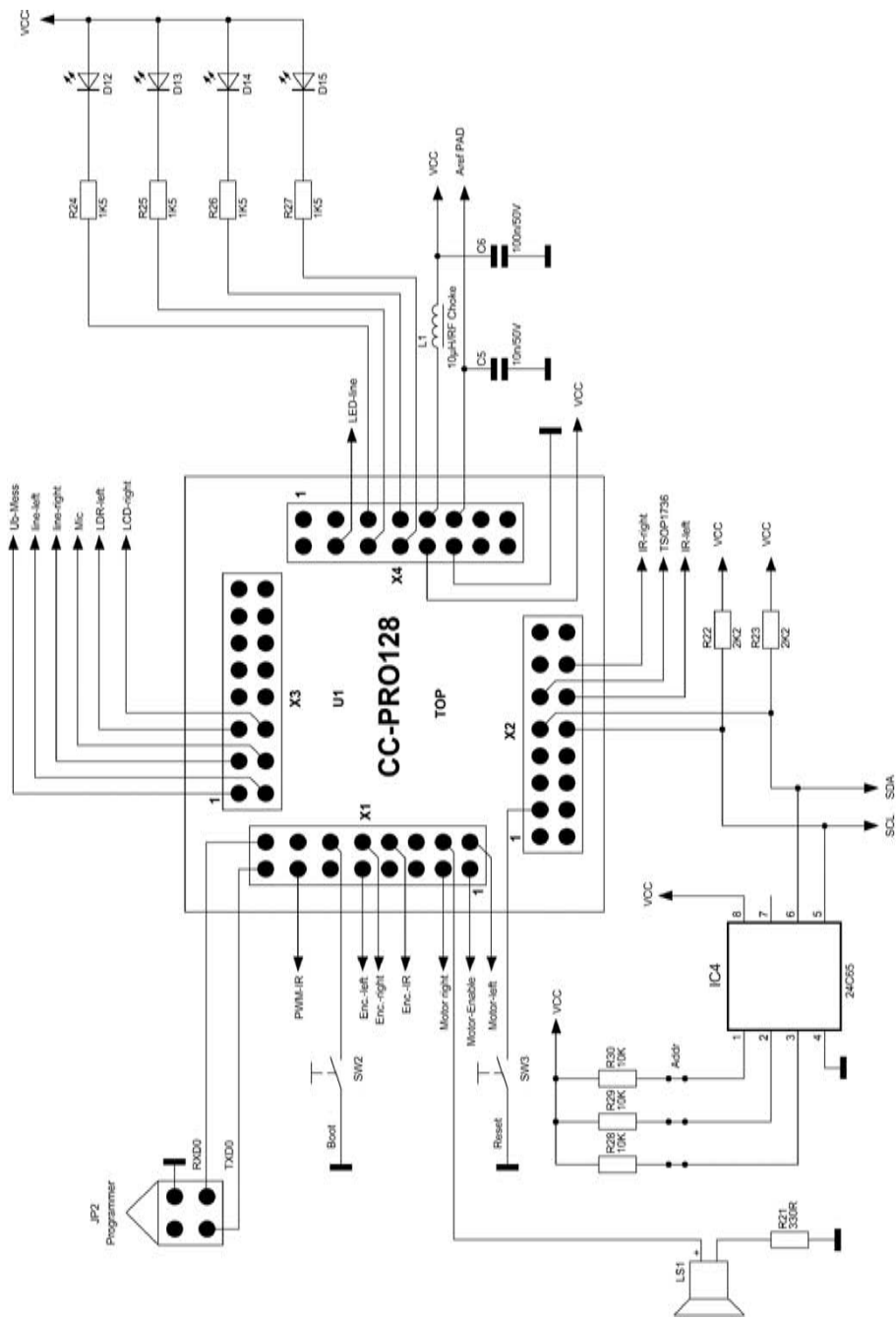


Abb. 8.2: Die C-Control-Pro-Beschaltung im PRO-BOT128

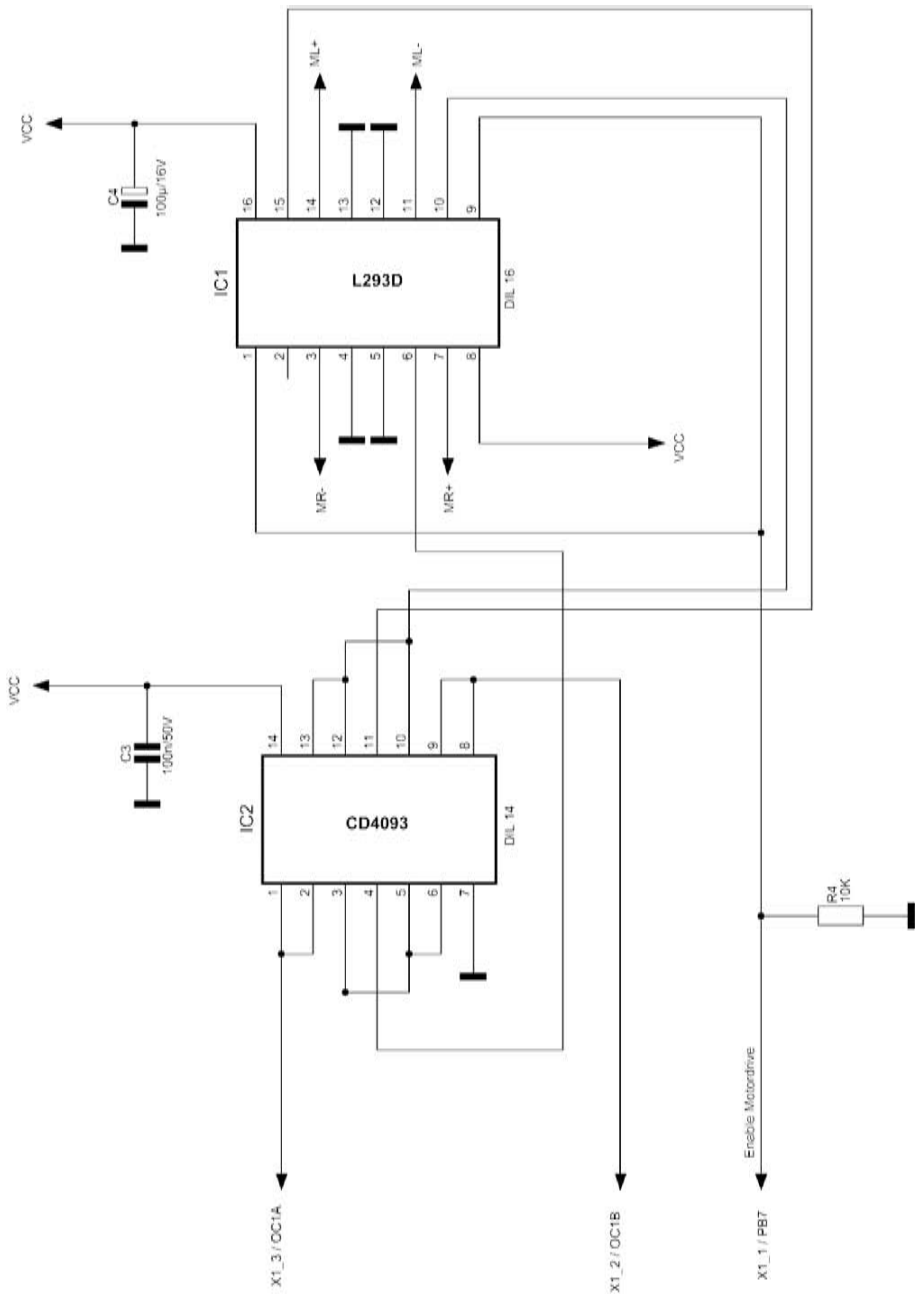


Abb. 8.3: Die Motor-H-Brücke für die Antriebsmotoren des Roboters

um den Motortreiber zu aktivieren. Der Kondensator C3 ist sehr nah am IC2 angebracht. Er dient zur Unterdrückung hochfrequenter Störungen auf der Versorgungsspannung, hervorgerufen durch die Motoren bei der PWM-Ansteuerung.

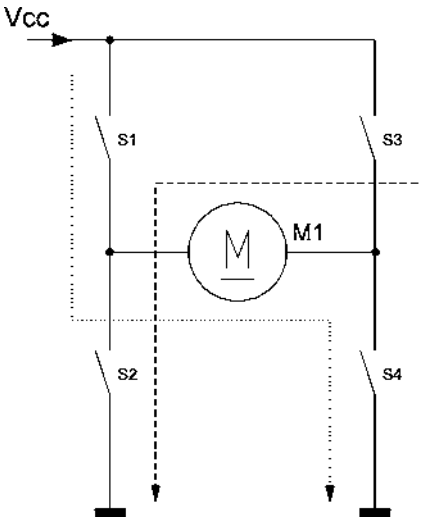


Abb. 8.4: Funktionsschema einer H-Brücke

8.4 Was ist eigentlich PWM?

Die Pulsweitenmodulation *PWM,* auch als *Unterschwingungsverfahren* bekannt, ist eine Modulationsart, bei der elektrischer Strom zwischen zwei Werten wechselt. Dabei wird das Tastverhältnis bei konstanter Frequenz moduliert. Auf unseren Motortreiber bezogen bedeutet das nichts anderes, als dass die C-Control Pro eine konstante Frequenz erzeugt und die Pulse zu Pausenverhältnissen verschoben werden. Eine niedrige Motordrehzahl wird erzeugt, wenn die Einschaltphase kürzer als die Ausschaltphase ist. Werden die Pausen kürzer und die Einschaltphasen länger, steigt die Drehzahl.

8.5 Sensoren, Teil 1

Die Sensoren sind die „Wahrnehmungssinne" des Roboters. Oben links im folgenden Schaltbild erkennt man den *TSOP1736.* Es handelt sich hier um einen Infrarotempfänger (kurz *IR-Empfänger*) mit integrierter Auswerteelektronik. Er empfängt das Infrarotsignal der sechs IR-Dioden *D5*, *D6*, *D7*, *D9*, *D10* und *D11* und gibt am Ausgang nur das aufmodulierte Signal wieder. Die sechs IR-Dioden werden mit einer Trägerfre-

quenz von 36 kHz betrieben, die im Rhythmus ein- und ausgeschaltet werden. Der IR-Empfänger gibt nur den Rhythmus wieder. Der IR-Empfänger und die IR-Dioden bilden das Antikollisionssystem des Roboters zur kontaktlosen Hinderniserkennung. Das funktioniert folgendermaßen: Die IR-Dioden sind in zwei Gruppen zu je drei Dioden aufgeteilt. Die C-Control Pro liefert am Ausgang *OC3A* ein PWM-Signal von 36 kHz. Die Dioden werden nun auf der anderen Seite (Kathode) jeweils kurz nach Masse gelegt, um den Stromkreis zu schließen. Dadurch wird ein kleiner, für uns nicht sichtbarer Lichtblitz ausgelöst. Wäre ein Hindernis in der Nähe, würde dieser Lichtblitz reflektiert, der IR-Empfänger würde das als eine logische Eins erkennen und wir könnten es in der Software als Hindernis interpretieren. Die Dioden sind mechanisch so angebracht, dass sie ca. 140° abdecken. Um die Reichweite des ACS zu ändern, verschiebt man die PWM-Frequenz der IR-Dioden. Das Datenblatt des TSOP zeigt ein Diagramm, das die Empfindlichkeit bezogen auf die Trägerfrequenz beschreibt. Senkt oder erhöht man die PWM-Frequenz um den Bereich von 36 kHz, wird der ACS unempfindlicher.

Später wird noch darauf eingegangen, wie man die ACS-Einheit zur Kommunikation mit dem PC und zur Steuerung mit einer IR-Fernbedienung nutzen kann.

Die Widerstände *R15* und *R17* sind lichtabhängige Widerstände (eng. *Light Dependent Resistor* oder kurz *LDR*) und bilden mit den Widerständen *R14* und *R16* einen lichtabhängigen Spannungsteiler, der an den AD-Ports der C-Control Pro angeschlossen ist. Mithilfe dieser Sensoren kann man die Lichtintensität messen und den Roboter z. B. die hellste oder dunkelste Stelle des Raums suchen lassen.

IC3 ist ein NF-Verstärker-IC, das auch gern als Kopfhörerverstärker und Verstärker für Kleinlautsprecher eingesetzt wird. Bei unserem Roboter wird er aber als Mikrofonverstärker verwendet. An ihm ist ein Kondensatormikrofon angeschlossen, das den „Umgebungslärm“ aufnimmt. Ein Kondensatormikrofon arbeitet etwas anders als ein dynamisches Mikrofon, das ähnlich wie ein Lautsprecher aufgebaut ist und durch Schall eine elektrische Spannung (Magnet bewegt sich in einer Spule) erzeugt. Ein Kondensatormikrofon ist, wie der Name schon sagt, ein Kondensator. Wird das Mikrofon besprochen, bewegen sich die Feldplatten im Mikrofon. Man erkennt im Schaltbild, dass an das Mikrofon noch ein Vorwiderstand *R18* angeschlossen ist. Er dient zur Vorspannungserzeugung, denn ein Kondensatormikrofon kann selbst keine Spannung erzeugen, sondern diese nur in der Frequenz modulieren. Der Kondensator trennt den Eingang des Verstärkers vom Gleichspannungsanteil und liefert am Eingang des IC die relativ geringe Wechselspannung, die nun im IC verstärkt wird. Der Ausgang des IC3 liefert je nach Schallpegel eine Spannung zwischen wenigen mV bis hin zu knapp 2,5 Volt. Angeschlossen ist der Ausgang des IC über einen Spannungsteiler *R19* und *R20* am AD-Port der C-Control Pro. Der Spannungsteiler ist erforderlich, da der IC eine Spannung von bis zu 5 Volt liefern kann. Da wir intern jedoch die Referenzspannung von 2,56 V verwenden, würde der ADC ständig übersteuert werden.

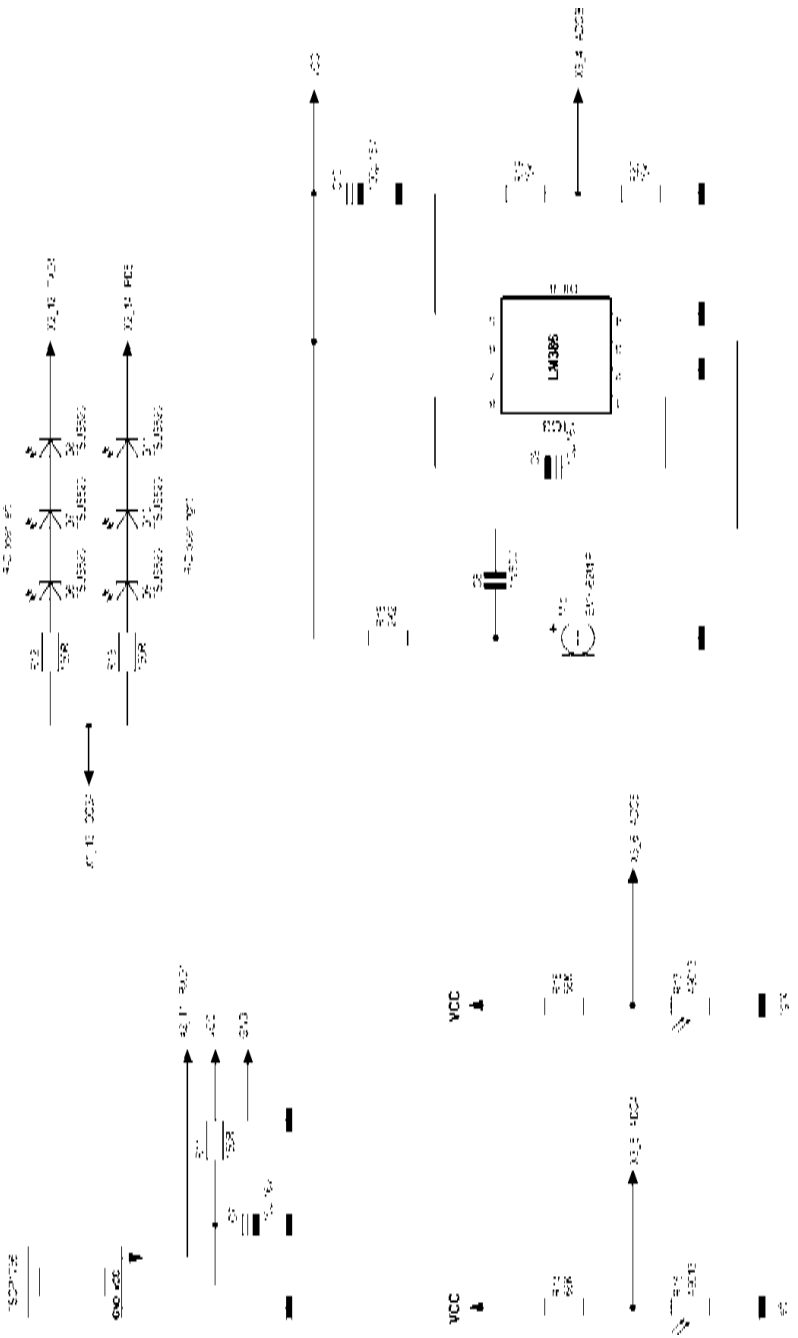


Abb. 8.5: Sensoren, Teil 1 (TSOP12736, ACS, LDRs und Schallerkennung)

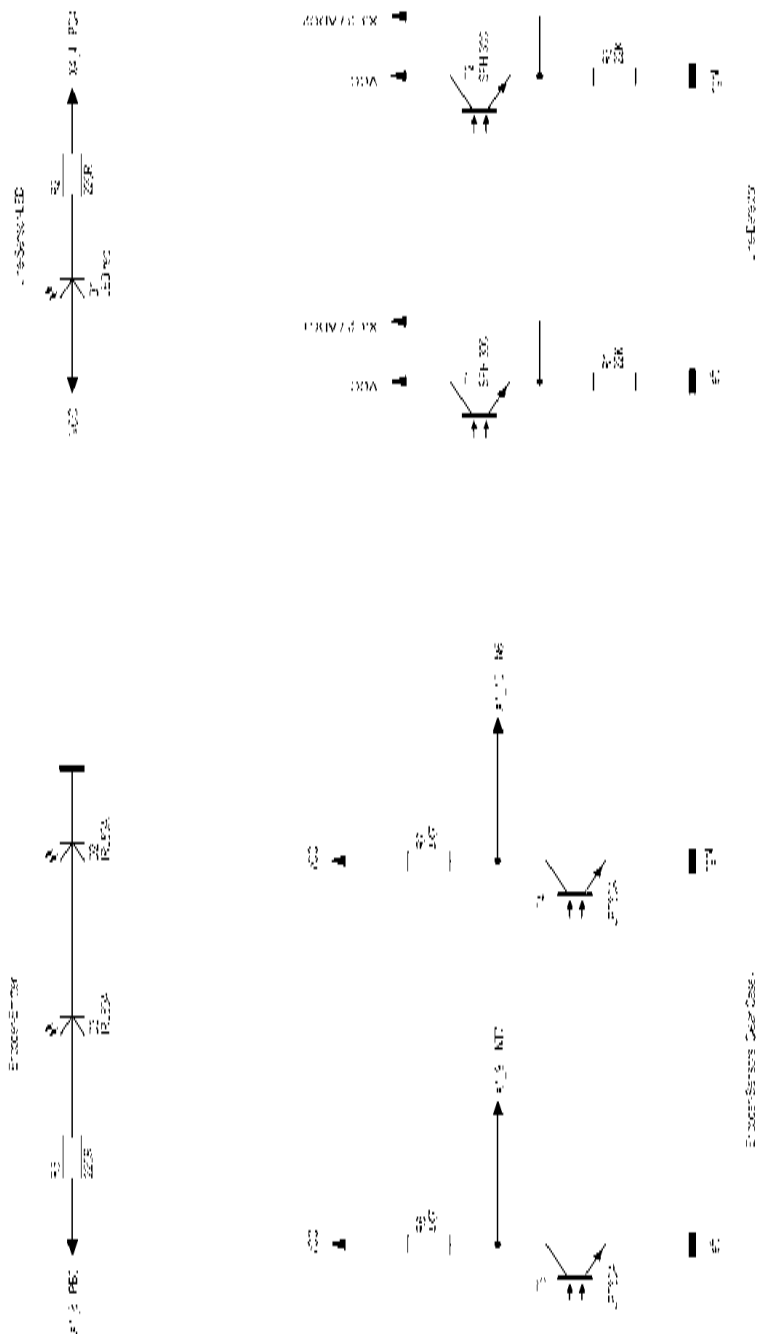


Abb. 8.6: Sensoren, Teil 2 (Odometer, Liniensensor)

8.6 Sensoren, Teil 2

Unsere Rad-Encoder (Odometer, Wegstreckenmessung, ähnlich wie der Kilometerzähler im Auto) arbeitet wie eine Reflexlichtschranke. Die Dioden *D3* und *D2* (Sender) leuchten die Taktscheiben des Antriebs an und die beiden Fototransistoren *T3* und *T4* (Empfänger) empfangen das reflektierte Signal immer dann, wenn ein weißes Feld auftaucht, und schalten auf Masse durch. Die Spannung sinkt dadurch je nach Intensität des IR-Lichts am *INT6* oder *INT7* ab. Die Fototransistoren arbeiten somit wie ein Schalter, der bei Licht geschlossen wird.

Unser Liniensensor arbeitet nach dem gleichen Prinzip. Die rote LED *D1* leuchtet auf den Untergrund, auf dem der Roboter steht. Ändert er seine Reflexionseigenschaften z. B. durch eine schwarze Linie, wird der Fototransistor *T1* oder *T2* eine unterschiedliche Spannung am AD-Port bereitstellen. Hiermit können wir entscheiden, ob wir links oder rechts von der Linie fahren.

8.7 Steckverbindungen

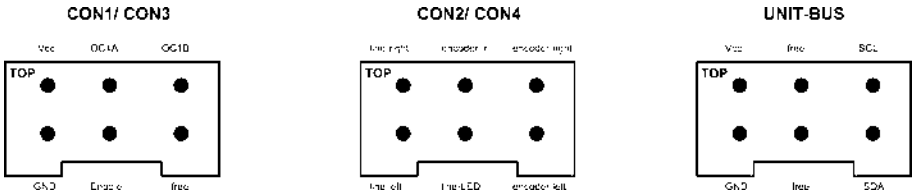


Abb. 8.7: Die Pfostenstecker verbinden die Ebenen untereinander elektrisch.

Die Wannenstecker stellen lediglich die elektrischen Verbindungen zwischen den einzelnen Ebenen her. Beim Zusammenbau sollte man immer darauf achten, dass die Quetschverbinder sauber und gerade zusammengepresst werden.

Sachverzeichnis

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

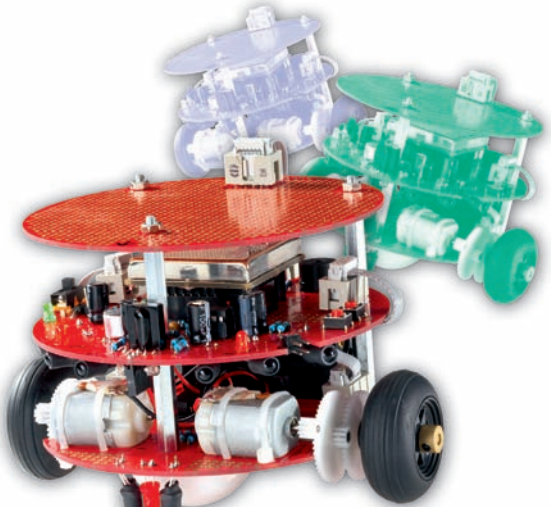
FRANZIS
ELEKTRONIK

Ulli Sommer

PRO-BOT128 selbst bauen und erfolgreich einsetzen

Das als Bausatz und Fertiggerät erhältliche Robotersystem Conrad PRO-BOT128 ist die ideale Grundlage für den Einstieg in die Elektronik, Mechanik und Programmierung. Dieses Robotersystem verwendet als „Gehirn" eine C-Control Pro Mega128, die ausreichend Hardware, Speicher und Geschwindigkeitsressourcen auch für komplexere Aufgaben bereitstellt. Der Roboter kann mit der mitgelieferten kostenlosen Entwicklungsumgebung (IDE) wahlweise in C, Basic und Assembler programmiert werden.

Ziel dieses Buchs ist es, Ihnen die Grundlagen der C-Control-Pro-Programmierung vom Einstieg bis hin zur Programmierung von komplexen Verhaltensweisen wie der Subsumtionsarchitektur zu erläutern und Ihnen die Materie Robotik auf spielerische und experimentelle Weise näherzubringen. Aber auch bereits erfahrene Roboter-Bastler werden durch die vielen Hard- und Softwareerweiterungen, die in diesem Buch beschrieben sind, ihre Freude haben. Viele Werke setzten fundiertes Fachwissen voraus, dieses Buch soll auch Einsteigern das komplexe Zusammenspiel von Mechanik, Elektronik und Software einfach und verständlich nahebringen.



Ein kleiner Ausschnitt aus dem Inhaltsverzeichnis

ISBN 978-3-7723-4117-5

9 783772 341175

Euro **19,95** [D]