

Das kompakte Universal-Board

Ideal für Robotik- und Experimentierschaltungen



RN-CONTROL Version 1.41

Dieses Board wurde speziell zum Experimentieren mit dem Mikrocontroller entworfen. Es eignet sich auch vorzüglich für den Einstieg in die Bereich Robotik und Mikrocontroller. Aber auch zahlreiche andere Verwendungen sind denkbar, das kostengünstige Board kann auch sehr gut zum realisieren von Projekten eingesetzt werden..

Hunderte von Bastlern haben dieses Board schon aufgebaut und damit den Einstieg in die Mikroelektronik erlernt. Dadurch ist auch schnelle Hilfe bei Soft- und Hardwarefragen in gängigen Foren wie dem Roboternetz.de möglich. Viele Beispielprogramme findet man ebenfalls im Roboternetz, RN-Wissen.de und natürlich Mikrocontroller-Elektronik.de auf welcher das Projekt unter der CC-Lizenz freigegeben wurde.



Ich denke die Leistungsdaten und Features können überzeugen!

Doku aktualisiert am 09.4.2015

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Lizenzhinweis

Dieses Projekt dient vornehmlich für Lehrzwecke und zum Experimentieren. Für den Aufbau sollten ausreichend Elektronik Grundkenntnisse und Kenntnisse bezüglich der Sicherheit (Experimentieren mit Strom und Handhabung gängiger Werkzeuge wie Lötkolben etc.) vorhanden sein. Auf der Seite www.Mikrocontroller-Elektronik.de empfehle ich diesbezüglich noch interessante Literatur mit der man sich dies erarbeiten kann. Weitere Literaturtipps sind auch unter RN-Wissen.de in der Rubrik [Buchvorstellungen](#) zu finden. Für Fragen bezüglich Elektronik und Mikrocontroller empfehle ich das Forum: Roboternetz.de

Um ihnen weitgehende Möglichkeiten zum Nutzen der Schaltung einzuräumen, wurde dieses Projekt jetzt unter die CC-Lizenz gestellt. Sie haben So die Möglichkeit die Schaltung beliebig zu verändern oder weiterzugeben. Lediglich die kommerzielle Weitergaben ist nur mit Genehmigung möglich! Genauere Hinweise finden Sie im Lizenztext.

Dieses Projekt (Schaltung und Projektdateien) steht unter der Creative-Commons-Lizenz Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Lizenziert wurde das Projekt von:

User Frank / www.Roboternetz.de & www.Mikrocontroller-Elektronik.de

Dieser Name und diese Webseiten sind bei der Weitergabe stets deutlich sichtbar zu nennen!

Achtung: Es kann keinerlei Garantie für die Fehlerfreiheit der Schaltung oder anderer Projektdateien übernommen werden! Der Nachbau und Betrieb geschieht auf eigene Gefahr! Jegliche Haftung für Schäden wird ausgeschlossen! Schadensersatzansprüche, gleich aus welchem Rechtsgrund, sind ausgeschlossen.



Dieses Projekt (Schaltung und Projektdateien) steht unter der Creative-Commons-Lizenz Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Lizenziert wurde das Projekt von: User Frank www.Roboternetz.de & www.Mikrocontroller-Elektronik.de

Dieser Name und diese Webseiten sind bei der Weitergabe stets deutlich sichtbar zu nennen!

Über diese Lizenz hinausgehende Erlaubnisse können Sie unter <http://www.mikrocontroller-elektronik.de/> erhalten.

Achtung: Der Nachbau und Betrieb geschieht auf eigene Gefahr! Jegliche Haftung für Schäden wird ausgeschlossen! Es kann keinerlei Garantie für die Fehlerfreiheit der Schaltung oder anderer Projektdateien übernommen werden!

Wichtige Hinweise und Haftungsausschluss



Lesen Sie bitte diese Hinweise, bevor sie dieses Projekt nachbauen bzw. in Betrieb nehmen.

Bestimmungsgemäße Verwendung: Dieses Projekt ist nur Entwicklungsaufgaben, Forschung, Lehrzwecke und Unterricht und Prototypenbau konzipiert! Für die Einhaltung der technischen Vorschriften sind sie selbst verantwortlich.

Sicherheitshinweise

Beim Umgang mit Produkten, die mit elektrischer Spannung in Berührung kommen, müssen die gültigen VDE-Vorschriften beachtet werden, insbesondere VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860.

Werkzeuge dürfen an Geräten, Bauteilen oder Baugruppen nur benutzt werden, wenn sichergestellt ist, dass die Geräte von der Versorgungsspannung getrennt sind und elektrische Ladungen, die in den im Gerät befindlichen Bauteilen gespeichert sind, vorher entladen wurden.

Spannungsführende Kabel oder Leitungen, mit denen das Gerät, das Bauteil oder die Baugruppe verbunden ist, müssen stets auf Isolationsfehler oder Bruchstellen untersucht werden. Bei Feststellen eines Fehlers in der Zuleitung muss das Gerät unverzüglich aus dem Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist. Bei Einsatz von Bauelementen oder Baugruppen muss stets auf die strikte Einhaltung der in der zugehörigen Beschreibung genannten Kenndaten für elektrische Größen hingewiesen werden. Wenn aus einer vorliegenden Beschreibung für den nicht gewerblichen Endverbraucher nicht eindeutig hervorgeht, welche elektrischen Kennwerte für ein Bauteil oder eine Baugruppe gelten, wie eine externe Beschaltung durchzuführen ist oder welche externen Bauteile oder Zusatzgeräte angeschlossen werden dürfen und welche Anschlusswerte diese externen Komponenten haben dürfen, so muss stets ein Fachmann um Auskunft ersucht werden. Es ist vor der Inbetriebnahme eines Gerätes generell zu prüfen, ob dieses Gerät oder Baugruppe grundsätzlich für den Anwendungsfall, für den es verwendet werden soll, geeignet ist!

Im Zweifelsfall sind unbedingt Rückfragen bei Fachleuten, Sachverständigen oder den Herstellern der verwendeten Baugruppen notwendig!

Bitte beachten Sie, dass Bedien- und Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen. Bei Installationen und beim Umgang mit Netzspannung sind unbedingt die VDE-Vorschriften zu beachten. Geräte, die an einer Spannung über 35 V betrieben werden, dürfen nur vom Fachmann angeschlossen werden. In jedem Fall ist zu prüfen, ob der Bausatz oder die Platine für den jeweiligen Anwendungsfall und Einsatzort geeignet ist bzw. eingesetzt werden kann.

Derjenige, der einen Bausatz fertigstellt oder eine Baugruppe durch Erweiterung bzw. Gehäuseeinbau betriebsbereit macht, gilt nach DIN VDE 0869 als Hersteller und ist verpflichtet, bei der Weitergabe des Gerätes alle Begleitpapiere mitzuliefern und auch seinen Namen und Anschrift anzugeben. Geräte, die aus Bausätzen selbst zusammengestellt werden, sind sicherheitstechnisch wie ein industrielles Produkt zu betrachten.

Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und/und Anschlussfehler außerhalb unseres Einflusses liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.

Als hier die Leistungsmerkmale auf einen Blick:

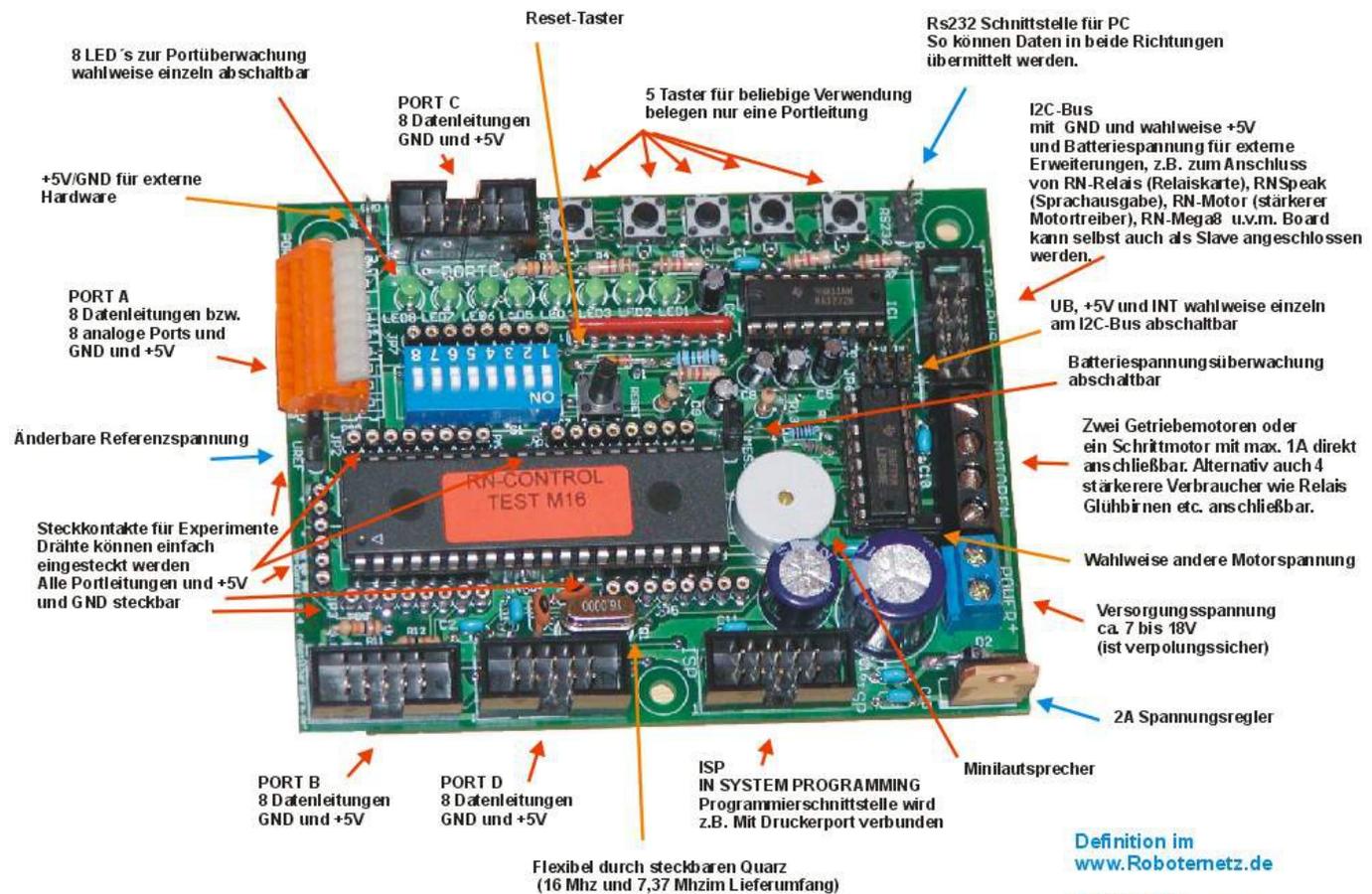
- Wahlweise 7,37 oder 16 Mhz Taktfrequenz (auch beliebiger anderer Quarz denkbar), empfohlen wird 16 Mhz!
- Schneller AVR Mega 16 oder Mega 32 Mikrocontroller (bis zu 32K Speicher, 2K Ram und 1K EEPROM), 32 programmierbare I/O Pins, 8 AD Ports u.v.m.
Alternativ auch ATMega 644 verwendbar
- 8 Leuchtdioden per DIP-Schalter deaktivierbar und anderen Ports per Steckbrücke beliebig zuzuordnen
- alle Portleitungen sind über Stecker nach außen geführt. Die Steckerbelegung entspricht der Roboternetz-Definition als auch der des Atmel Entwicklungsboards STK500
- alle Ports sind zusätzlich über Steckbuchse erreichbar. Ideal zum experimentieren da einfach Drähte (ca. 0,5mm²) eingesteckt werden (kein Löten oder schrauben). So können einfach andere LED's zugeordnet werden oder ein Steckbrett verbunden werden
- Der wichtige Port A (wahlweise 8 digitale oder analoge Ein- o. Ausgänge) ist zusätzlich noch über eine Qualitätssteckklemme mit Hebel herausgeführt
- Motortreiber mit ca. 1A (2x 0,5A) belastbar – für zwei Getriebemotoren oder 1 Schrittmotor. Dieser kann auch für andere Zwecke (Relaisansteuerung, Lämpchen etc.) genutzt oder einfach entfernt werden
- Integrierter programmierbarer Mini-Lautsprecher um Töne auszugeben
- 1 Reset Taster
- 5 Taster für beliebige Verwendung. Sie belegen nur einen analogen Port!
- 5 V Spannungsstabilisierung mit 2 A Belastbarkeit, auch herausgeführt für Erweiterungen
- Eingangsspannung gegen Verpolung geschützt
- RS232 mit normgerechtem Pegelwandler (MAX232) – PC direkt anschließbar
- Batteriespannung kann im Programm abgefragt werden
- ISP – Programmierschnittstelle für übliche AVR-Programmieradapter (10polig)
- **auch per RS232 Programmierbar wenn man Bootloader installiert (siehe in Doku hinten unter BOOTLOADER)**
- Betriebsspannung wahlweise zwischen 7 und 18V (empfohlen 7 bis 14 V) – wahlweise auch höhere Motorspannung bis 24V möglich)
- Sehr kompakt, nur halbes Europaformat nach Roboternetz-Definition (ca. 100x75mm)
- I2C-Bus über die zahlreiche Erweiterungsplatinen anschließbar sind (z.B. Sprachausgabe RN-Speak, Relaisboard RN-Relais, Servoboards, LCD's uvm.)
- Programmierbar in zahlreichen Sprachen, z.B. Basic (BASCOM), C (GCC), AVR-Studio, Assembler, Pascal (viele Entwicklungsumgebungen sind frei im Internet erhältlich)

- Deutsche Doku mit Basic Programmbeispiel
- Kein Starter- oder Applikationsboard notwendig – bereits alles integriert!

Diagramm und Beschreibung der Anschlüsse

RN-CONTROL

V.1.4 kompaktes Low Cost Board mit vielen Leistungsmerkmalen für Roboter, Modellbau und vieles mehr. Ideal auch zum Experimentieren



Definition im
www.Roboternetz.de

Bezugsquelle:
www.robotikhardware.de

Aufbau und Anwendung

Aufbau

Der Aufbau der Schaltung ist durch die vorgefertigte Platine eigentlich völlig problemlos auch von Elektronik-Einsteigern zu bewerkstelligen.

Die Eagle-Dateien zum Projekt können über <http://www.mikrocontroller-elektronik.de/> heruntergeladen werden. Diese können genutzt werden um sich bei einem Platinenhersteller eine Platine anfertigen zu lassen. Auf unserer [Seite](#) nennen wir gerne einige Hersteller die sowas machen.

Durch den Bestückungsdruck und die Bestückungsliste, etwas weiter hinten in dieser Dokumentation, ist der Aufbau unkritisch. Aufgrund moderner Bauteile hält sich die Anzahl der Kleinteile in Grenzen, weshalb die Schaltung meist in ca. 30 bis 60 Minuten aufgebaut ist.

Dennoch einige Anmerkungen zu kleinen Hürden:

1. Das Board verwendet zwei Widerstandsnetzwerke . Auf der Platine ist die mit RN1 und RN2 gekennzeichnet. Da diese Teile Einsteigern noch nicht so bekannt sind, möchte ich darauf hinweisen das diese richtig herum eingelötet werden müssen. Auf den schwarzen Teilen ist auf einer Seite ein Punkt. Dieser Punkt muss auf die Seite wo auf der Platine eine kleine 1 aufgedruckt ist.
2. Sie können frei wählen welchen Quarz Sie einlöten. In der Regel reichen 7,3728 Mhz voll aus, damit ist das Board noch immer schneller als viele vergleichbare Boards dieser Preisklasse, zudem braucht es dann weniger Strom als mit 16 Mhz. Brauchen Sie jedoch die volle Rechenpower, dann ist 16 Mhz sinnvoll. Bei 16 Mhz brauchen Sie jedoch auch einen guten Programmieradapter, billige Lösungen machen hier manchmal Probleme mit der Übertragung. Einige werden wir demnächst auf unserer Seite vorstellen!
Dir krumme Zahl 7,3728 Mhz hat noch einen zweiten Vorteil. Mit dieser Frequenz ist die Baudrate der RS232 ganz exakt, weshalb sich damit höhere Übertragungsgeschwindigkeiten erreichen lassen.
Nachtrag: Seit geraumer Zeit gehört nur noch der 16 Mhz Quarz zum Lieferumfang da dieser zu 99% sowieso immer eingesetzt wurde. Er kann natürlich bei Bedarf durch einen beliebigen anderen zwischen 1 und 16 Mhz ersetzt werden.
3. Achten Sie darauf das die Taster richtig herum eingelötet ist. Richten Sie sich am besten nach dem Foto. Wenn die Platine so vor ihnen liegt das die Beschriftungen T1, T2, T3 usw. auf dem Kopf stehen, dann dürfen die Beine der Tastern nur links und rechts von dem Tastergehäuse zu sehen sein.
4. Erfahrenere Anwender die die Referenzspannung von 2,5V zusammen mit der Batterieüberwachung nutzen wollen und zugleich mehr als 13,4V Eingangsspannung anlegen möchten, wird empfohlen bei R9 einen größeren Widerstand als 22K zu verwenden (z.B. 33k). Einsteiger sollten den Jumper (Kurzschlussstecker) auf UREF immer erst gesteckt lassen (UREF ist dann 5V).
5. Bezüglich der Polung LED's auf den Bestückungsdruck in der Anleitung achten. Da wo die flache Seite der LED ist, muss das kurze Bein (Kathode) eingelötet werden
6. Nicht vergessen das zu den meisten IC's Fassungen mitgeliefert werden, also immer Fassung und nicht IC direkt einlöten

7. Die Polung des Piezo Lautsprechers ist nicht zu beachten, er kann beliebig eingelötet werden

Das waren eigentlich schon die besonderen Punkte die zu beachten sind. Ansonsten natürlich sauber mit einem 15 – 25 W LötKolben alles auf der Unterseite verlöten. Grundkenntnisse beim Löten werden empfohlen.

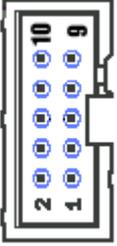
Nach dem Aufbau sollten Sie nochmals alle Lötunkte kontrollieren. Wenn Sie dann Spannung anlegen, dann sollten deutlich weniger als 100mA Strom fließen. Ist der Strom höher, dann deutet das auf ein Lötfehler hin.

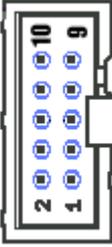
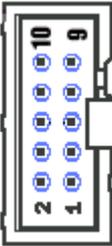
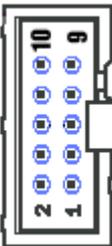
Das mitgelieferte Testprogramm kann dann recht einfach mit Bascom oder Pony übertragen werden. Per Tastendruck können anschließend verschiedene Boardeigenschaften überprüft werden.

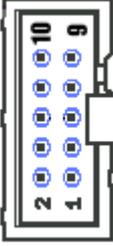
Der Spannungsregler wird beim Betrieb von RN-Control sehr warm, insbesondere wenn alle LED's leuchten und eventuell noch Verbraucher angeschlossen sind. Dies ist jedoch völlig normal und kein Grund zur Besorgnis, der Spannungsregler ist dafür ausgelegt. Er ist im übrigen intern mit einem Hitzeschutz als auch Kurzschlussschutz ausgestattet, er ist also nahezu unzerstörbar.

Lediglich wenn sie Verbraucher anschließen welche sehr viel Strom der 5V Leitung entnehmen, so kann es erforderlich werden ein Kühlkörper anzuschrauben. RN-Control ist so konstruiert das wahlweise ein Kühlkörper montiert werden kann oder aber das der Spannungsregler einfach an eine Gehäusewand geschraubt wird. Alternativ wäre auch ein Metallwinkel denkbar welcher die Wärme auf ein Bodenblech weiterleitet. Diese Kühlmaßnahmen sind wie gesagt nur bei höherer Stromentnahme notwendig, zum Beispiel wenn Sie ein LCD mit Beleuchtung anschließen. Die Beleuchtung kann bei vielen LCD's schon recht viel Strom verbrauchen.

Erläuterung der Anschlüsse, Regler und Kurzschlussbrücken

Anschluss-Bezeichnung	Erläuterung
Port A	<p>Digitaler I/O und analoger Port (PA 0 bis PA7 und ADC0 bis ADC7) Über eine Steckklemme werden hier die 8 Portleitungen PA0 bis PA 7 zur Verfügung gestellt. PA7 befindet sich dabei ganz links und PA0 ganz rechts. Anschlußdrähte können einfach eingesteckt werden, indem man mit einem Kugelschreiber oder Schraubenzieher den oberen weißen Hebel etwas nach unten drückt. Die Ports PA7 bis PA0 können sowohl als normaler I/O-Port (Ein- und Ausgabeport) oder als AD-Port's programmiert werden. Somit könnten also auch bis zu 8 Spannungen quasi gleichzeitig gemessen werden.</p> <p>Ist der Kurzschlussstecker UREF eingesteckt, dann können Spannungen bis zu 5V gemessen werden. Ist UREF offen, dann können Spannungen nur bis 2,5V gemessen werden. Durch geeignete Spannungsteiler kann der Meßbereich natürlich beliebig erhöht werden. Achtung: Die zulässige Höchstspannung 2,5V oder 5V darf am Port nicht überschritten werden, dieses würde den Port zerstören!</p> <p>Vorbelegung: Port PA7 wird auch für die Tastenabfrage genutzt, indem über einen Spannungsteiler verschiedene Spannungen per Tastendruck angelegt werden (siehe Schaltplan). Solange keine Taste gedrückt ist, ist dieser jedoch frei verfügbar. Port PA6 wird über einen Spannungsteiler (22k und 5,1K) zur Batteriespannungsmessung benutzt (siehe Schaltplan und Demoprogramm) wenn der Kurzschlussstecker UMESS eingesteckt ist. Durch entfernen dieses Steckers steht der Port zur freien Verfügung.</p> <p>Der komplette Port steht auch nochmals über die Buchsenleiste JP2 zur Verfügung. Auch dort können Drähte zum experimentieren eingesteckt werden (möglichst 0,6mm²).</p>
Port B 	<p>Digitale I/O Port B (PB0 bis PB 7) Über einen Wannenstecker werden gemäß der Roboternetz-Definition 8 I/O Portleitungen mit Sonderfunktionen als auch GND und +5V bereitgestellt.</p> <p>Die genaue Belegung sieht wie folgt aus:</p> <p>Pin 1 PB0 / T0 / XCL / wird vom Motortreiber für Motor 2 Kanal 1 genutzt Pin 2 PB1 / T1 / wird vom Motortreiber für Motor 2 Kanal 2 genutzt Pin 3 PB2 / AIN0 / INT2 / wird auch vom I2C-Bus genutzt Pin 4 PB3 / AIN1 / OC0 Pin 5 PB4 / SS Pin 6 PB5 / MOSI / wird auch vom ISP Anschluss genutzt Pin 7 PB6 / MISO / wird auch vom ISP Anschluss genutzt Pin 8 PB7 / SCK / wird auch vom ISP Anschluss genutzt Pin 9 GND Pin 10 +5V</p> <p>Durch entfernen des Motortreiber IC's aus der Fassung, steht PB0 und PB1 zur freien Verfügung.</p> <p>Der komplette Port steht auch nochmals über die Buchsenleiste JP3 zur Verfügung. Auch dort können Drähte zum experimentieren eingesteckt werden (möglichst 0,6mm²).</p>

<p>Port C</p> 	<p>Digitale I/O Port C (PC0 bis PC 7) Über einen Wannenstecker werden gemäß der Roboternetz-Definition 8 I/O Portleitungen mit Sonderfunktionen als auch GND und +5V bereitgestellt.</p> <p>Die genaue Belegung sieht wie folgt aus:</p> <p>Pin 1 PC0 / SCL wird vom I2C-Bus genutzt Pin 2 PC1 / SDA wird vom I2C-Bus genutzt Pin 3 PC2 / TCK Pin 4 PC3 / TMS Pin 5 PC4 / TDO Pin 6 PC5 / TDI Pin 7 PC6 / TOSC1 / wird vom Motortreiber für Motor 1 Kanal 1 genutzt Pin 8 PC7 / TOSC2 / wird vom Motortreiber für Motor 1 Kanal 2 genutzt Pin 9 GND Pin 10 +5V</p> <p>Über den 8 fachen DIP Schalter können den Ports auch LED's zugeschaltet werden! Die LED's leuchten wenn Port LOW-Zustand annimmt!</p> <p>Durch entfernen des Motortreiber IC's aus der Fassung, steht PC6 und PC7 zur freien Verfügung. Der komplette Port steht auch nochmals über die Buchsenleiste JP4 zur Verfügung. Auch dort können Drähte zum experimentieren eingesteckt werden (möglichst 0,6mm²).</p>
<p>Port D</p> 	<p>Digitale I/O Port D (PD0 bis PD 7) Über einen Wannenstecker werden gemäß der Roboternetz-Definition 8 I/O Portleitungen mit Sonderfunktionen als auch GND und +5V bereitgestellt.</p> <p>Die genaue Belegung sieht wie folgt aus:</p> <p>Pin 1 PD0 / RXD / wird für RS232 Schnittstelle genutzt Pin 2 PD1 / TXD / wird für RS232 Schnittstelle genutzt Pin 3 PD2 / INT0 Pin 4 PD3 / INT1 Pin 5 PD4 / OC1B / wird für PWM Motor 1 benutzt (Geschwindigkeitsregelung) Pin 6 PD5 / OC1A / wird für PWM Motor 2 benutzt (Geschwindigkeitsregelung) Pin 7 PD6 / ICP Pin 8 PD7 / OC2 Pin 9 GND Pin 10 +5V</p> <p>Durch entfernen des Motortreiber IC's aus der Fassung, steht PD4 und PD5 zur freien Verfügung. Der komplette Port steht auch nochmal über die Buchsenleiste JP5 zur Verfügung. Auch dort können Drähte zum experimentieren eingesteckt werden (möglichst 0,6mm²).</p>
<p>I2C-Bus</p> 	<p>I2C-Bus Über diesen Bus lassen sich zahlreiche Erweiterungen an dieses Board anschließen. Zum Beispiel werden auf der Seite roboternetz.de passende Boards mit Sprachausgabe, Relais, Schrittmotorsteuerung etc. vorgestellt.</p> <p>Aber auch dieses Board kann selbst als Slave-Board, also als Erweiterung an ein anderes Hauptboard angeschlossen werden.</p> <p>Der I2C-Bus benötigt nur 2 Leitungen für alle Funktionen. Entsprechend der Roboternetz-Norm wird hier ein 2x5 poliger Stecker angeschlossen. Die Belegung entspricht exakt der anderer Roboternetz Boards.</p> <p>Pin 1 SCL (Taktleitung) Port PC0 Pin 3 SDA (Datenleitung) Port PC1 Pin 5 +5V Pin 7 +5V Pin 9 Batteriespannung Pin 2,4,6,8 GND Pin 10 INT Diese Leitung kann von allen I2C-Bus Erweiterungen genutzt (Port PB2) werden um den Hauptcontroller darüber zu informieren das sich Daten (z.B. von Sensoren) verändert haben. In diesem Fall wird die Leitung solange auf Masse gelegt bis der entsprechende I2C-Baustein ausgelesen wird.</p> <p>Die Controller muss also immer alle I2C-Bausteine auslesen solange diese Leitung auf Masse liegt. Diese Leitung ist mit Port PD2 verbunden</p> <p>Die PIN's 5,7,9 und 10 können über herausnehmbare Kurzschlussbrücken (Jumper JP6) vom Board getrennt werden. Dies ist zum Beispiel dann notwendig, wenn bereits ein anderes Masterboard die Spannungen auf den Bus legt. Es darf immer nur ein Board die Spannungen bereitstellen.</p>

<p>ISP</p> 	<p>ISP – IN SYSTEM PROGRAMMING Über diesen Anschluß kann der Controller auf dem Sprachboard mit einem Standard ISP-Kabel direkt an einen Parallelport des PC's angeschlossen und programmiert werden. Die Belegung des ISP-Anschlusses ist zu dem weit verbreitetet STK200 Programmier Dongle kompatibel. Ein entsprechender Dongle kann man sich entweder selber basteln (siehe Artikel „ARV Einstieg leicht gemacht“ unter www.roboternetz.de)</p> <p>Pin 1 MOSI Pin 2 VCC Pin 3 Nicht belegt Pin 4 GND Pin 5 RESET Pin 6 GND Pin 7 SCK Pin 8 GND Pin 9 MISO Pin 10 GND</p>
<p>Power</p>	<p>Spannungsversorgung Über diese Schraubklemme wird das Board mit Spannung versorgt. Es reicht eine unstabilierte Gleichspannung von 7 bis 14V aus (max. 18V wenn Kühlkörper verwendet wird) + und – sind auf der Platine markiert. Das Board ist jedoch auch gegen ein verpolen geschützt, so das nichts kaputt geht!</p>
<p>Motoren</p>	<p>Motoren Über diese 4 polige Schraubklemme können zwei Getriebemotoren (jeweils die beiden linken oder rechten Kontakte) oder ein Schrittmotor angeschlossen werden. Der Motortreiber kann jedoch auch für andere Dinge genutzt werden, z.B zum Ansteuern von Relais, Lämpchen etc. verwendet werden. Die Belastbarkeit liegt bei 1A. Sollen größere Motoren angeschlossen werden, so kann z.B. über den I2C-Port eine andere Endstufe angeschlossen werden.</p>
<p>JP6</p>	<p>I2C-Bus Belegung Über drei Kurzschlussstecker können wahlweise die Bateriaespannung (UB), +5V sowie INT mit dem I2C-Bus verbunden werden. Wenn INT nicht benötigt wird, kann man diesen Jumper offen lassen. Somit hat man einen Port zusätzlich frei zur Verfügung Möchte man das Board über den I2C-Bus mit Spannung versorgen, dann kann man UB oder +5V Jumper einstecken. In diesem Fall braucht/darf keine Spannung an dem Power Schraubklemen angelegt werden. Möchte man umgekehrt andere Boards über den I2C-Bus mit Spannung versorgen, dann müssen die Jumper UB und/oder +5V eingesteckt werden. Durch diesen Jumper ist man für alle Fälle gerüstet. Bei älteren RN-Control Versionen mußte man dazu noch das Kabel ändern.</p>
<p>JP8</p>	<p>Über diesen Stecker kann die stabilisierte 5V Logikspannung für Erweiterungen oder Experimente entnommen werden. Wird mehr als 500mA entnommen, so sollte der Spannungsregler mit einem kleinen Kühlkörper versehen werden.</p>
<p>UREF</p>	<p>Referenzspannung Über eine Kurzschlussstecker kann hier die Referenzspannung von 5V eingestellt werden. Wird der Stecker entfernt, so kann an den analogen Ports nur bis 2,5V gemessen werden (jedoch mit höherer Genauigkeit). Sicherheitshalber sollten Sie den Stecker anfangs eingesteckt lassen! Wird er entfernt, so sollte man daran denken das auch die Bateriaespannung über einen analogen Port gemessen wird. Diese darf dann nicht viel höher als 13V sein!</p>
<p>UMOT</p>	<p>Motorspannung Wenn dieser Kurzschlussstecker eingesteckt wird, dann wird die volle Bateriaespannung auch für die Motoren benutzt. Ansonsten könnte man über einen PIN dieses Jumpers auch eine höhere Versorgungsspannung für die Motoren nutzen. Dazu sollte sie aber nochmals in den Schaltplan schau. In der Regel sollte hier ein Jumper eingesteckt sein!</p>
<p>UMESS</p>	<p>Bateriaespannungsmessung Wenn diese Kurzschlussbrücke gesteckt ist, dann wird über Port PA6 die Bateriaespannung überwacht. Ansonsten ist der Port frei!</p>
<p>S1</p>	<p>DIP Schalter Mit einem kleinen Schraubenzieher kann über diesen 8 poligen Schalter den Portleitungen PC0 bis PC7 eine LED zugeschaltet werden. Die LED's leuchten immer dann wenn der Port LOW (0 Pegel) führt. Wenn die LED auf Off geschaltet wird, so kann die LED über eine Drahtbrücke von JP7 mit einem anderen Port verbunden werden.</p>

RS232	<p>PC kompatible RS232 Schnittstelle Über ein Adapterkabel kann die serielle Schnittstelle des PC direkt mit dem Board verbunden werden. Dies ist dann sinnvoll, wenn Fehler in Programmen gesucht . Einfache PRINT Anweisungen werden von einem Terminalprogramm angezeigt. Hier kann Hyperterminal von Windows oder das eingebaute terminalprogramm von Bascom empfohlen werden.</p> <p>Die Belegung ist kompatibel zum Conrad Roboter CCRP5:</p> <p>Pin 1 RX Pin 2 GND Pin 3 TX</p> <p>Ein geeignetes Anschlußkabel kann schnell selbst angefertigt werden.</p>
JP7	<p>Über diese Buchse sind alle LED's ganz einfach durch Einstecken eines Drahtes beschaltbar. Bedenken muß man dabei das die Anode über einen Vorwiderstand immer mit 5V verbunden ist. Die LED leuchtet also nur wenn sie mit der Drahtbrücke auf GND bzw. einen Port mit 0 Pegel gelegt wird.</p>
JP1	Buchsenleiste die dreimal +5V und drei mal GND für Experimente bereitstellt
JP2	Buchsenleiste die den gesamten Port A für steckbare Drähte bereitstellt
JP3	Buchsenleiste die den gesamten Port B für steckbare Drähte bereitstellt
JP4	Buchsenleiste die den gesamten Port C für steckbare Drähte bereitstellt
JP5	Buchsenleiste die den gesamten Port D für steckbare Drähte bereitstellt
TASTER T1 bis T5	<p>Stehen zur freien Verfügung Die Abfrage ist im Demoprogramm beschrieben</p>

Bauteile Bestell- und Bestückungsliste für RN-Control

(Angaben ohne Gewähr)

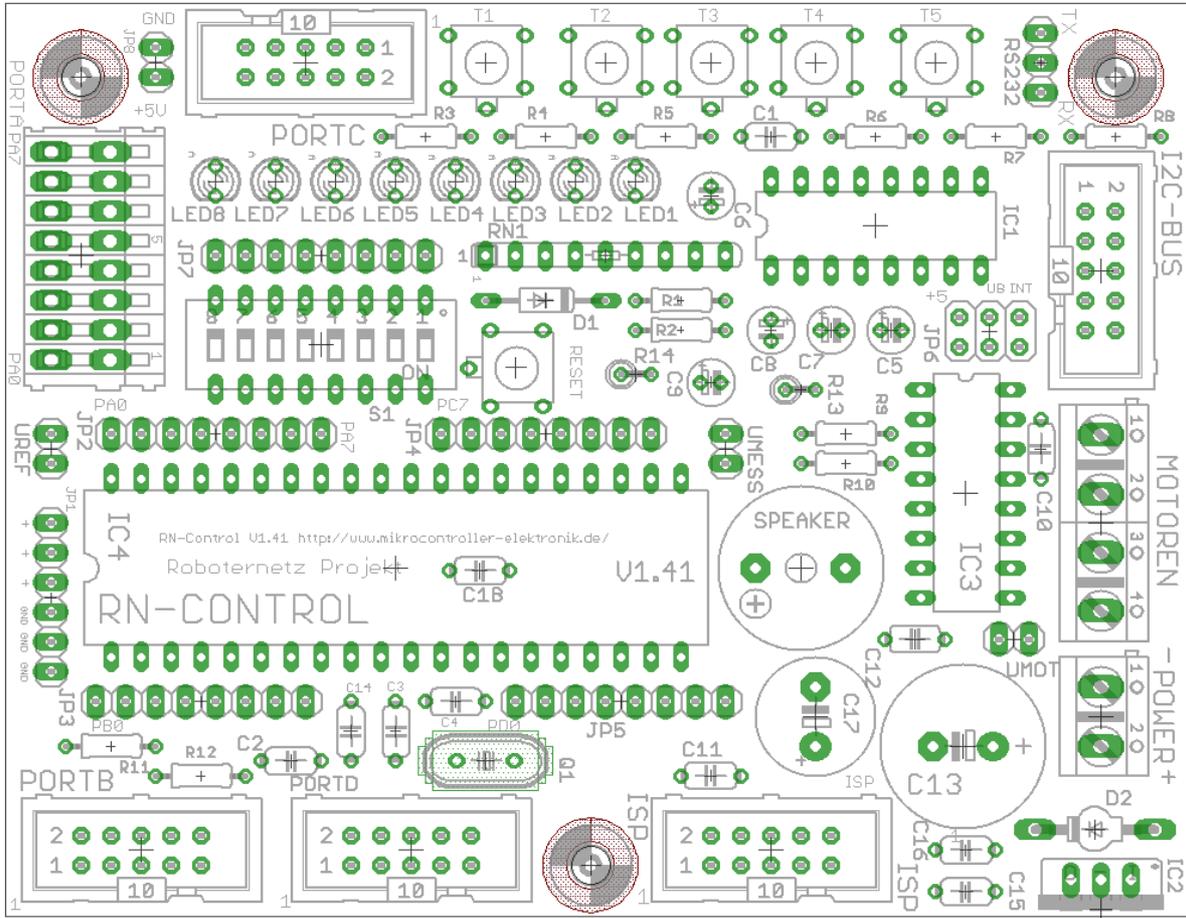
Bauteil	Wert	Beschreibung	Link zu Bezugsquellen auf unserer Projektseite
C1	100n	Keramik Kondensator	KERKO100N
C2	100n	Keramik Kondensator	KERKO100N
C3	22pf	Keramik Kondensator	KERKO-500 22p
C4	22pf	Keramik Kondensator	KERKO-500 22p
C5	4,7uF	Elko	SM 4,7/50RAD
C6	4,7uF	Elko	SM 4,7/50RAD
C7	4,7uF	Elko	SM 4,7/50RAD
C8	4,7uF	Elko	SM 4,7/50RAD
C9	1uF	Elko	SM 1,0/63RAD
C10	100n	Keramik Kondensator	KERKO100N
C11	100n	Keramik Kondensator	KERKO100N
C12	100n	Keramik Kondensator	KERKO100N
C13	1000uF	Elko	RAD 1000/35
C14	100n	Keramik Kondensator	KERKO100N
C15	100n	Keramik Kondensator	KERKO100N
C16	100n	Keramik Kondensator	KERKO100N
C17	220uF	Elko	RAD 220/35
C18	100n	Keramik Kondensator	KERKO100N
D1	1N4148	Diode	1n 4148
D2	BYV27	Diode	BYV 27/200
I2C-BUS	I2C	Wannenbuchse	WSL 10G
IC1	MAX232	RS232 Treiber	MAX 232 CPE
IC2	7805	Spannungsregler	78S05
IC3	L293D	Motortreiber	L 293 D
IC4	MEGA16-P	Atmel Mega 16 oder 32	ATMEGA 16-16
ISP	AVR-ISP	Wannenbuchse	WSL 10G
JP1		Kontaktbuchse (manuell kürzen)	SPL20
JP2		Kontaktbuchse (manuell kürzen)	SPL 20
JP3		Kontaktbuchse (manuell kürzen)	SPL 20
JP4		Kontaktbuchse (manuell kürzen)	SPL 20
JP5		Kontaktbuchse (manuell kürzen)	SPL 20
JP6		Stiftleiste	Stiftl. 2x50g (teilen)
JP7		Kontaktbuchse (manuell kürzen)	SPL 20
JP8		Stiftleiste	LU 2,5 MS2
LED1		Leuchdiode Low	LED 3MM 2MA GN
LED2		Leuchdiode Low	LED 3MM 2MA GN
LED3		Leuchdiode Low	LED 3MM 2MA GN
LED4		Leuchdiode Low	LED 3MM 2MA GN
LED5		Leuchdiode Low	LED 3MM 2MA GN
LED6		Leuchdiode Low	LED 3MM 2MA GN
LED7		Leuchdiode Low	LED 3MM 2MA GN
LED8		Leuchdiode Low	LED 3MM 2MA GN
MOTOREN		Schraublemme 4 polig	AKL 101-04
PORTA		Steckklemme 8 polig	WAGO 233-508
PORTB		Wannenbuchse	WSL 10G
PORTC		Wannenbuchse	WSL 10G
PORTD		Wannenbuchse	WSL 10G
POWER		Schraubklemme 2 polig	AKL 101-02
Q1		Quarz 16 Mhz	16-HC18
R1	100k	Widerstand 100k	1/4W 100k

R2	1k	Widerstand 1k	1/4W 1k
R3	10k	Widerstand 10k	1/4W 10k
R4	1k	Widerstand 1k	1/4W 1k
R5	1k	Widerstand 1k	1/4W 1k
R6	1k	Widerstand 1k	1/4W 1k
R7	1k	Widerstand 1k	1/4W 1k
R8	1k	Widerstand 1k	1/4W 1k
R9	22k	Widerstand 22k	1/4W 22k
R10	5,1k	Widerstand 5,1k	1/4W 5,1k
R11	10k	Widerstand 10k	1/4W 10k
R12	10k	Widerstand 10k	1/4W 10k
R13	10k	Widerstand 10k	1/4W 10k
R14	10k	Widerstand 10k	1/4W 10k
RESET	TASTER3301	Minitaster liegend	TASTER 3301
RN1		Widerstandsnetzwerk	SIL 9-8 1,0k
RS232		Stiftleiste 3 polig	LU 2,5 MS3
S1		DIP Schalter 8 polig	NT08
SPEAKER	F/CM12P	Mini Piezo Lautsprecher	SUMMER EPM 121
T1	TASTER3301	Minitaster liegend	TASTER 3301
T2	TASTER3301	Minitaster liegend	TASTER 3301
T3	TASTER3301	Minitaster liegend	TASTER 3301
T4	TASTER3301	Minitaster liegend	TASTER 3301
T5	TASTER3301	Minitaster liegend	TASTER 3301
UMESS		Stiftleiste	LU 2,5 MS2
UMOT		Stiftleiste	LU 2,5 MS2
UREF		Stiftleiste	LU 2,5 MS2

Achten Sie darauf das zuerst die IC-Fassungen und nicht direkt die IC's eingelötet werden.

Platinen lassen sich über die [Eagle Dateien](#) sehr leicht anfertigen lassen.
Die Eagle Dateien können über die [Projektseite](#) heruntergeladen werden.

Bestückungsplan



Der erste Test

Nachdem Board aufgebaut ist, können wir daran gehen und das Board testen. Zunächst sollten noch einmal alle Jumper (Kurzschlussbrücken) überprüft werden. Für den Ersten Einsteig empfehlen wir folgendes:

- UMOT** - Jumper einstecken (Damit Batteriespannung auch für Motoren verwenden)
- UREF** - Jumper einstecken (Damit 5V Referenzspannung für Spannungsmessungen verwenden)
- UMESS** – Jumper einstecken (Damit wird Batteriespannung über Port PA6 gemessen)

Danach ein möglichst ein Netzteil mit 7 bis 13 V anschließen, da benötigt man noch keinen Kühlkörper. Günstige Netzteile gibt's im Handel ab ca. 5 Euro! Wir empfehlen das 9V Netzteil! Wenn Sie kein finden, schauen sie mal auf die Seite <http://www.mikrocontroller-elektronik.de/>, dort werden wir viele Bezugsquellen für Bauteile, Sensoren usw. verlinken.

Hinweis: Bei der Netzteilspannung sollte man beachten, das auch die Motoren die volle Spannung vertragen. Verträgt Ihr Motor beispielsweise nur 6V, so kann zwar auch ein 12V Netzteil verwendet werden, jedoch muss man dann beim Ansteuern darauf achten das man „nie Vollgas gibt“, also das PWM-Signal entsprechende programmiert. Dies ist im übrigen recht einfach!

Hat das Netzteil den falschen Stecker, einfach abschneiden und Drähte so in die Schraubklemmen einfügen. Auf Polung achten (auch wenn Verpolung das Board nicht beschädigen kann).

Falls Sie ein Messgerät haben, können Sie auch den Strombedarf des Boards checken. Wenn alles korrekt zusammengebaut wurde, muss dieser deutlich niedriger als 100 mA liegen. Bei 16 Mhz und Mega 16 bei ca. 60 mA!. Ein wesentlich höherer Strom würde auf Lötfehler hin deuten. Ein weiterer Test wäre das anfassen des Spannungsreglers 78S05. Er darf warm bis sehr warm werden, aber man darf sich nicht dran verbrennen. Er ist im übrigen gegen Überlastung geschützt!

Stimmt das alles, dann kann man den ISP-Programmieraadapter anschließen. Es eignet sich jeder gängige ISP Dongle der *STK200* / *STK300* kompatibel ist. Bei 16 Mhz machen allerdings billig Lösungen manchmal sehr viele Übertragungsfehler. Hier empfehlen wir einen modernen USB-ISP-Programmer. Hier in der Anleitung verwenden wir teilweise noch einen Programmer der an die Druckerschnittstelle angeschlossen wird, das ist heute aber eigentlich er selten. Per USB geht's aber fast genauso! Über den Programmieraadapter wird nun das Board mit der Druckerschnittstelle ihres PC verbunden.. Dabei ist darauf zu achten das der ISP-Stecker auch in die richtige Wannenbuchse auf dem Board gesteckt wird.

Eine Verwechslung kann den ISP-Programmer oder den Controller zerstören.

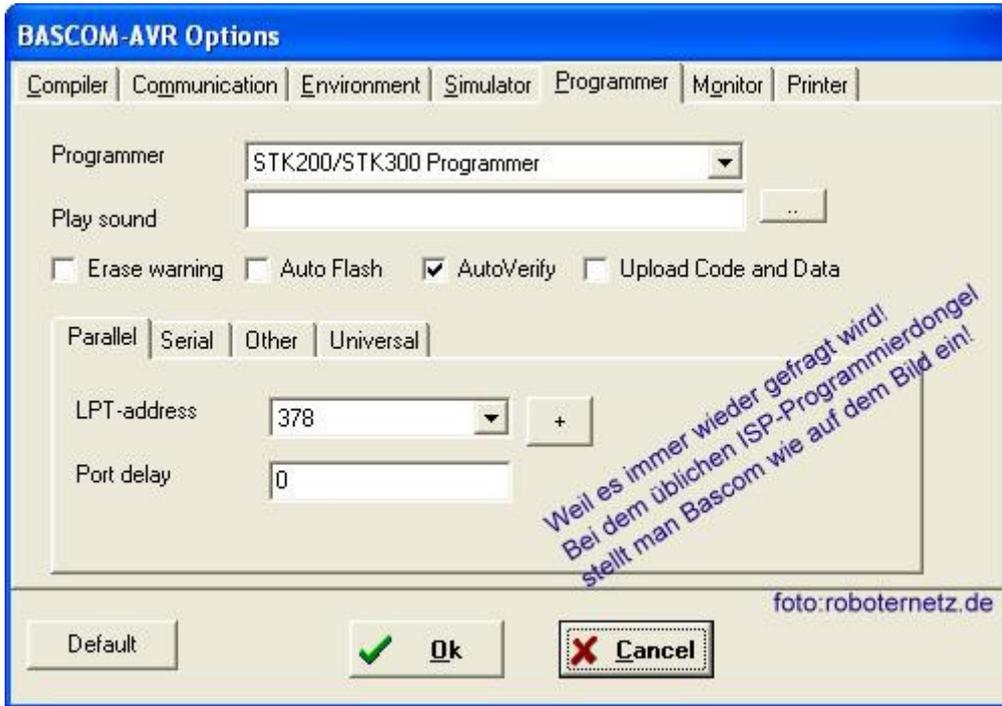
Also unbedingt auf die Beschriftung achten! Falschrum kann man ihn nicht aufstecken.

Ist man kein AVR-Profi, so empfehle ich für den Test des Board's die Entwicklungsumgebung und Programmiersprache Bascom. Ein schneller Basic-Compiler mit hervorragender Benutzeroberfläche. Eine Version die nur in der Programmlänge beschränkt ist (max 2KB) gibt es beim Hersteller MCS Electronics kostenlos. Die genauen Links findet man am besten im Roboternetz-Bascom Forum [http://www.roboternetz.de/community/forums/41-Basic-Programmierung-\(Bascom-Compiler\)](http://www.roboternetz.de/community/forums/41-Basic-Programmierung-(Bascom-Compiler)) Im Roboternetz gibt es ein Unterforum das sich nur mit Bascom beschäftigt. Hier findet man also immer Unterstützung.

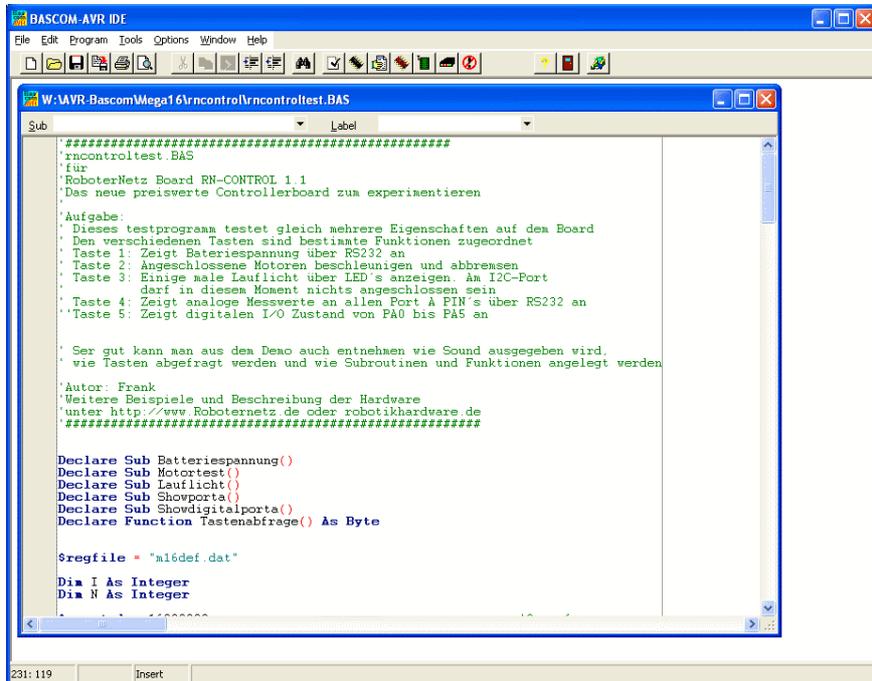
Über oben genannten Link findet man 4 Zip Dateien die man in einem leeren Verzeichnis entpackt. Anschließend kann man das ganze einfach mit SETUP installieren. Anschließend legt man über File/New ein neue Datei an und gibt das nachfolgenden Beispielprogramme ein. Alternativ kann man das ganze Beispielprogramm und viele andere Programme auch auf der Seite <http://www.mikrocontroller-elektronik.de/> finden.

So sieht die Bascom Entwicklungsumgebung aus:

So sollte Bascom eingestellt werden um den Standard ISP-Programmer zu benutzen:



So sieht die Entwicklungsumgebung aus:



In der Bascom Entwicklungsumgebung sollte man nach dem erstmaligen Start über *Options/Compiler/Chip* den Controllertyp anwählen. Bei der üblichen Bestückung müssten sie dort **M16** (für Mega16) zu wählen. Dann muß in einem anderen Karte dieses Dialoges *Programmer* noch der ISP-Adapter gewählt werden. In den meisten Fällen dürfte das der *STK200 / STK300 Programmer* sein.

Damit dürfte das wichtigste passiert sein. Allerdings ist nun noch der Quarz deaktiviert, da der Mega16 generell immer den internen 1 Mhz Takt nutzt. Aber das muss uns für den ersten Test nicht stören. Im Beispiel sollte dann jedoch statt `$crystal = 8000000` oder `16000000` die Anweisung `$crystal = 1000000` stehen. Jetzt kennen Sie auch schon die Anweisung die die Taktfrequenz angibt. Diese ist sehr wichtig damit Zeitabhängige Dinge wie Timerprogrammierung, RS232, Wait-Funktionen usw. korrekt funktionieren.

```
#####
'rncontroltest.BAS
'für
'RoboterNetz Board RN-CONTROL 1.41
'Das neue preiswerte Controllerboard zum experimentieren
'
'Aufgabe:
' Dieses Testprogramm testet gleich mehrere Eigenschaften auf dem Board
' Den verschiedenen Tasten sind bestimmte Funktionen zugeordnet
' Taste 1: Zeigt Batteriespannung über RS232 an
' Taste 2: Angeschlossene Motoren beschleunigen und abbremesen
' Taste 3: Einige male Lauflicht über LED's anzeigen. Am I2C-Bus
'         darf in diesem Moment nichts angeschlossen sein
' Taste 4: Zeigt analoge Messwerte an allen Port A PIN's über RS232 an
''Taste 5: Zeigt digitalen I/O Zustand von PA0 bis PA5 an

' Ser gut kann man aus dem Demo auch entnehmen wie Sound ausgegeben wird,
' wie Tasten abgefragt werden und wie Subroutinen und Funktionen angelegt werden

'Autor: Frank
'Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de oder http://www.mikrocontroller-elektronik.de/
'#####

Declare Sub Batteriespannung()
Declare Sub Motortest()
Declare Sub Lauflicht()
Declare Sub Showporta()
Declare Sub Showdigitalporta()
Declare Function Tastenabfrage() As Byte

$regfile = "m16def.dat"

Dim I As Integer
Dim N As Integer
DIM Ton As Integer

$crystal = 16000000           'Quarzfrequenz
$baud = 9600

Config Adc = Single , Prescaler = Auto           'Für Tastenabfrage und Spannungsmessung

Config Pina.7 = Input           'Für Tastenabfrage
Porta.7 = 1                     'Pullup Widerstand ein

Const Ref = 5 / 1023           'Für Batteriespannungsberechnung

Dim Taste As Byte
Dim Volt As Single

' Für Motorentest
'Ports für linken Motor
Config Pinc.6 = Output           'Linker Motor Kanal 1
Config Pinc.7 = Output           'Linker Motor Kanal 2
Config Pind.4 = Output           'Linker Motor PWM
'Ports für rechten Motor
Config Pinb.0 = Output           'Rechter Motor Kanal 1
Config Pinb.1 = Output           'Rechter Motor Kanal 2
```

```

Config Pind.5 = Output                                'Rechter Motor PWM
Config Timer1 = Pwm , Pwm = 10 , Compare A Pwm = Clear Down , Compare B Pwm = Clear Down
Pwm1a = 0
Pwm1b = 0
Tccr1b = Tccr1b Or &H02                               'Prescaler = 8

I = 0
Sound Portd.7 , 400 , 450                             'BEEP
Sound Portd.7 , 400 , 250                             'BEEP
Sound Portd.7 , 400 , 450                             'BEEP
Print
Print "**** RN-CONTROL 1.1 ****"
Print "Das neue Experimentier- und Roboterboard"
Print
Do

    Taste = Tastenabfrage()
    If Taste <> 0 Then

        Select Case Taste
            Case 1
                Call Batteriespannung                    'Taste 1 Zeigt Bateriaespannung über
RS232 an
            Case 2
                Call Motortest                          'Taste 2 Motoren beschleunigen und
abbremsen
            Case 3
                Call Lauflicht                          'Einige male Lauflicht über LED's
anzeigen. Am I2C-Port darf in diesem Moment nichts angeschlossen sein
            Case 4
                Call Showporta                          'Zeigt Messwerte an allen Port A PIN's
            Case 5
                Call Showdigitalporta                  'Zeigt digitalen I/O Zustand von PA0 bis
PA5 an
        End Select
        Sound Portd.7 , 400 , 500                       'BEEP
    End If

    Waitms 100
Loop
End

'Diese Unterfunktion fragt die Tastatur am analogen Port ab
Function Tastenabfrage() As Byte
Local Ws As Word

    Tastenabfrage = 0
    Ton = 600
    Start Adc
    Ws = Getadc(7)
    If Ws < 1010 Then
        Select Case Ws
            Case 410 To 450
                Tastenabfrage = 1
                Ton = 550
            Case 340 To 380
                Tastenabfrage = 2
                Ton = 500
            Case 265 To 305
                Tastenabfrage = 3
                Ton = 450
            Case 180 To 220
                Tastenabfrage = 4
                Ton = 400
        End Select
    End If
End Function

```

```

        Case 100 To 130
            Tastenabfrage = 5
            Ton = 350
        End Select
        Sound Portd.7 , 400 , Ton           'BEEP
    End If

End Function

'Diese Unterfunktion zeigt Bateriaespannung an
Sub Batteriespannung()
Local W As Word
    Start Adc
    W = Getadc(6)
    Volt = W * Ref
    Volt = Volt * 5.2941
    Print "Die aktuelle Spannung betragt: " ; Volt ; " Volt"
End Sub

'Testet Motoren und Geschwindigkeitsreglung
Sub Motortest()
    'Linker Motor ein
    Portc.6 = 1           'bestimmt Richtung
    Portc.7 = 0           'bestimmt Richtung
    Portd.4 = 1           'Linker Motor EIN

    'Rechter Motor ein
    Portb.0 = 1           'bestimmt Richtung rechter Motor
    Portb.1 = 0           'bestimmt Richtung rechter Motor
    Portd.5 = 1           'rechter Motor EIN

    I = 0
    Do
        Pwmla = I
        Pwmlb = I
        Waitms 40
        I = I + 5
    Loop Until I > 1023

    Wait 1
    Do
        Pwmla = I
        Pwmlb = I
        Waitms 40
        I = I - 5
    Loop Until I < 1
    Pwmla = 0           'Linker Motor aus
    Pwmlb = 0           'rechter Motor aus
End Sub

'Einige male Laufflicht ber LED's anzeigen. Am I2C-Port darf in diesem Moment nichts angeschlossen
sein
Sub Laufflicht()

    Config Portc = Output
    Portd = 0
    For N = 1 To 10
        For I = 0 To 7
            Portc.i = 0
            Waitms 100
            Portc.i = 1
        Next I
    Next N
    Config Portc = Input
End Sub

'Zeigt Die Analogen Messwerte An Port A An
Sub Showporta()
Local Ws As Word

    Config Porta = Input

```

```

    For I = 0 To 5                                     ' Alle internen Pullup Widerständ
ein,bis auf Batteriespannungsmessungsport
        Porta.i = 1
    Next I

    Print
    Print "Ermittelte Messwerte an Port A:"
    For I = 0 To 7                                     ' Alle Eingänge inkl.messen
        Start Adc
        Ws = Getadc(i)
        Volt = Ws * Ref
        Print "Pin " ; I ; " ADC-Wert= " ; Ws ; " bei 5V REF waeren das " ; Volt ; " Volt"
    Next I
End Sub

'Zeigt den Zustand einiger freier I/OI/O von Die Analogen Messwerte An Port A An
Sub Showdigitalporta()
Local Zustand As String
    Config Porta = Input
    For I = 0 To 5                                     ' Alle internen Pullup Widerständ
ein,bis auf Batteriespannungsmessungsport
        Porta.i = 1
    Next I

    Print
    Print "Ermittelter I/O Zustand Port A:"
    For I = 0 To 5                                     ' Alle Eingänge inkl.messen
        If Pina.i = 1 Then
            Zustand = "High"
        Else
            Zustand = "Low"
        End If
        Print "Pin " ; I ; " I/O Zustand= " ; Pina.i ; " " ; Zustand
    Next I
End Sub

```

Das Testprogramm ist etwas länger weil es über eine Reihe von Unterfunktionen über Tastendruck gleich mehrere Dinge testen kann. Dadurch ersparen wir uns das laden unterschiedlicher Testprogramme und lernen so gleich nebenbei wie man Tasten abfragt.

Achtung: In der Prüfversion von Bascom kann diese Version eventuell nicht kompiliert werden, da diese etwas mehr als 2 KB Speicher benötigt. Zum kompilieren müsste das Demo entweder gekürzt oder die Vollversion des Bascom Compilers bestellt werden. Auf der mitgelieferten CD befindet sich jedoch bereits die kompilierte Version, diese läßt sich auch mit dem Demo übertragen.!

Haben sie alles richtig gemacht und oberes Beispiel auch korrekt eingegeben, dann können sie das Programm über das Symbol „Schwarze IC in der Toolbar“ kompilieren. Es darf keine Fehlermeldung kommen. Erscheint eine Fehlermeldung dann klicken sie doppelt darauf, korrigieren sie die fehlerhafte Zeile und kompilieren sie erneut. Ist das Programm fehlerfrei kompiliert, so klicken sie auf das grüne Symbol etwas weiter rechts (soll wohl eine IC-Wechselfassung sein). Anschließend wählen sie Programm in dem erscheinenden Menü. Dadurch gelangen sie in das eingebaute Übertragungsprogramm. Rechts oben müssen Sie nun nochmals M16 für Mega16 wählen (falls noch nicht vorgegeben). Danach klicken sie auch in diesem Dialog auf das gleiche grüne Symbol. Wenn sie alles in Bascom richtig eingestellt haben und die Platine als auch das ISP Kabel in Ordnung ist, dann sollte das Programm in wenigen Sekunden übertragen worden und auch gestartet worden sein.

Den erfolgreichen Start erkennt man an einer kurzen Piepfolge. Drücken Sie nun nochmal zum Test die Reset-Taste in der Mitte des Boards. Die kurze Piepfolge müsste sich wiederholen, denn dadurch wird das Programm nochmals gestartet.

Die Tonfolge klingt vermutlich recht schrecklich und unheimlich lahm. Dies liegt daran das das Programm eigentlich für 8 und 16 Mhz ausgelegt ist. Bei 1 Mhz müsste man bei den Sound Befehlen andere Angaben machen. Da auch die RS232 nicht so sehr gut mit 1 Mhz funktioniert, gehen wir nun gleich dazu über das Board auf 8 (bzw. 7,3728 Mhz) oder 16 Mhz umzustellen.

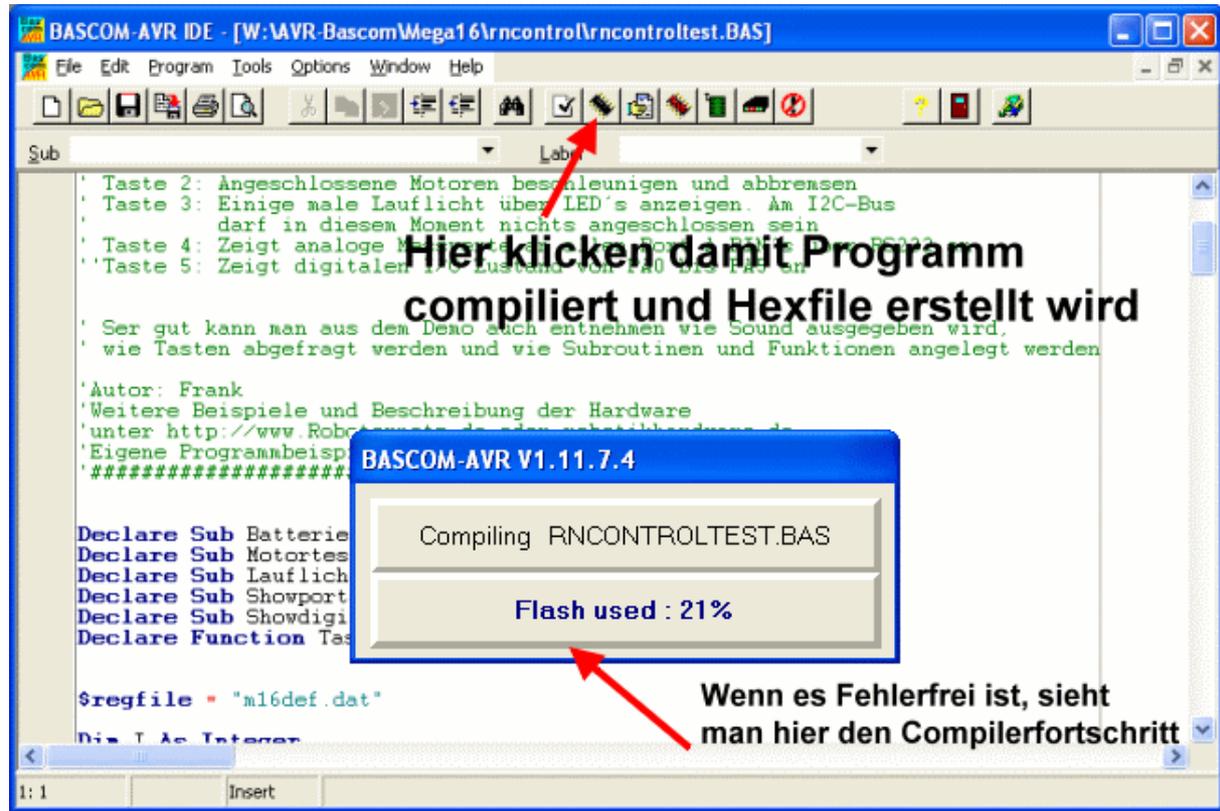
Testprogramm in der Programmiersprache C

Natürlich läßt sich RN-Control auch in C programmieren. Der notwendige Compiler WinAVR wird bereits kostenlos mitgeliefert. Das obere Testprogramm findet man in der Programmiersprache C für RN-Control auf folgender Webseite:

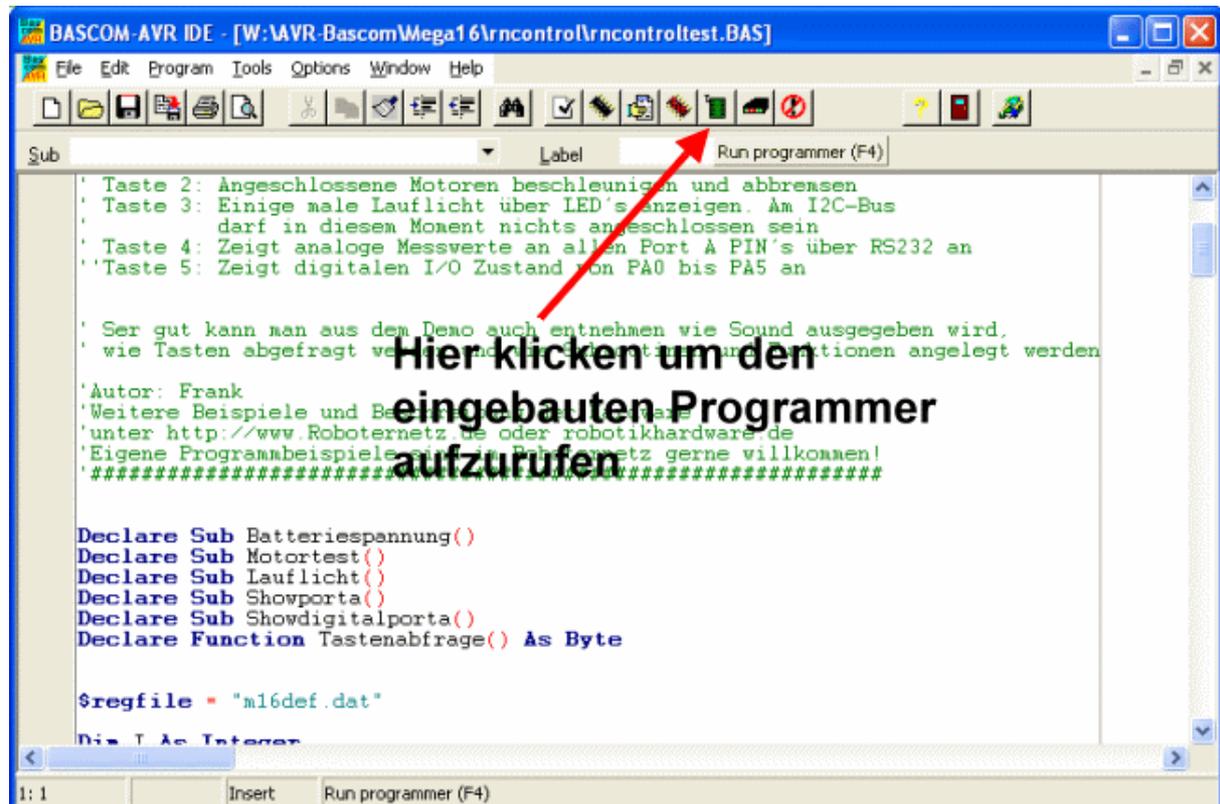
http://www.roboternetz.de/wissen/index.php/RN-Control_Demprogramm_in_C

So ist der Ablauf in der Bascom-Entwicklungsumgebung:

Als erstes Programmcode kompilieren



Nun den eingebauten Programmer aufrufen:



Und nun das Programm in den Chip Übertragen (ISP-Programmierdongel muß angeschlossen sein)

The screenshot shows the AVR ISP STK programmer interface. The 'Auto program chip' button is highlighted with a red arrow. A dialog box titled 'BASCOM-AVR programming status' is open, showing 'Writing ROM' and a progress bar. The background shows a memory table with hexadecimal values.

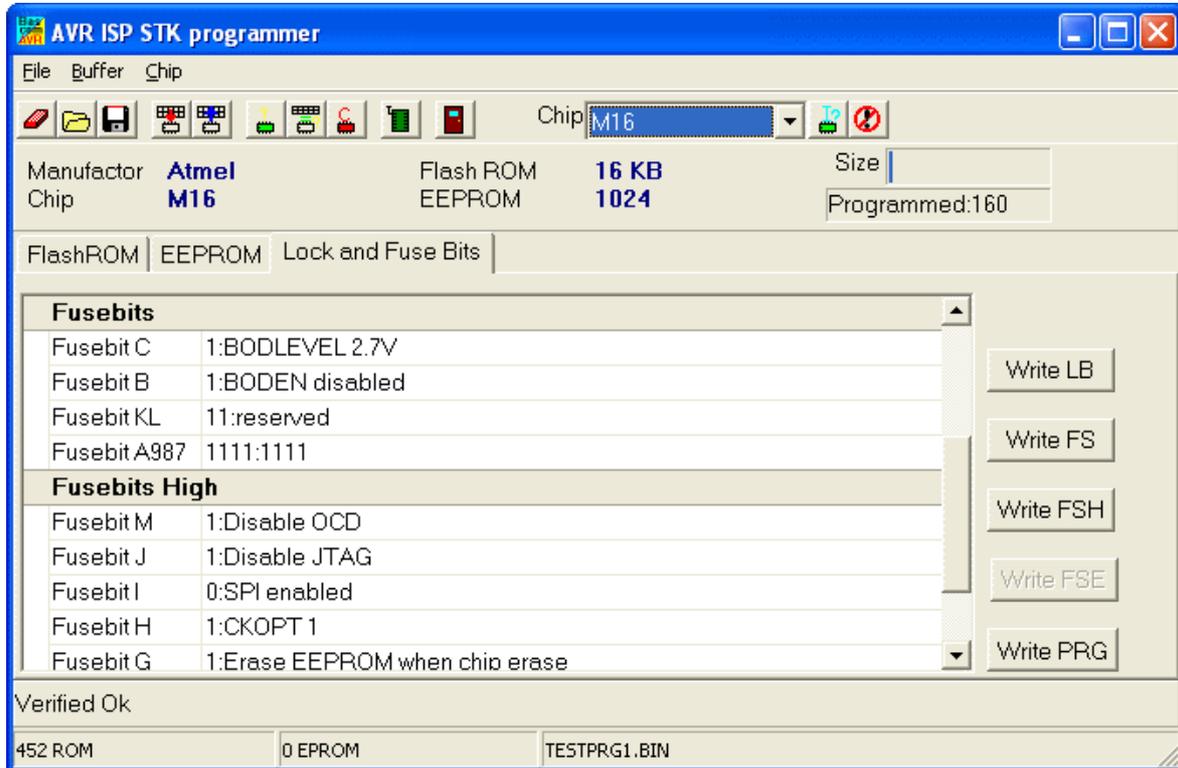
Nun hier klicken um das Programm in den Controller zu übertragen

Nach dem oberen Klick sieht man hier den Übertragungsfortschritt

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	94	0C	00	2A	5											
0010	95	18	00	00	5	18	00	00	95	18	00	00	95	18	00	00
0020	95	18	00	00	5	18	00	00	95	18	00	00	95	18	00	00
0030	95	18	00	00	95	18	00	00	95	18	00	00	95	18	00	00
0040	95	18	00	00	95	18	00	00	95	18	00	00	95	18	00	00
0050	95	18	00	00	E1	8F										
0060	E4	C0	E3	E8	2E	4E										
0070	EF	EE	E0	F3	E6	A0										
0080	E6	87	B9	89	E1	88										
0090	9A	DF	9A	A6	9A	A7										
00A0	BD	8F	E0	80	BD	8E										
00B0	E0	90	BD	99	BD	88										
00C0	E0	80	E0	90	E6	A0	E0	80	93	8D	93	9C	9A	8F	EC	E2
00D0	E0	F1	E9	80	E0	91	97	0								
00E0	94	0F	04	55	2F	48	28	49	E7	B1	9A	85	FF	FA	E0	E0

Nun Fuse-Bits des Mega einstellen

Nachdem das Beispiel gut zu funktionieren scheint, sollte man gleich die Gelegenheit nutzen und den Controller auf die richtige Quarz Taktfrequenz von 8 Mhz oder 16 Mhz umzuschalten. Dies kann man auch in Bascom machen indem man wieder den Programmer über das grüne Symbol aufruft und unten die Seite „Lock and Fuse Bits“ anwählt (siehe Bild).



Das ganze muß bei angeschlossenen Board erfolgen damit die momentan eingestellten Werte erscheinen. Die Werte sollten so eingestellt werden, das sie der oberen Abbildung entsprechen. Hier sollten Sie unbedingt *Fusebit C, B, KL* und *A987* vergleichen und gegebenenfalls ändern. Gleichzeitig sollten Sie an dieser Stelle *Fusebit J* auf *Disable JTAG* einstellen, damit auch die JTAG-Ports frei sind, ansonsten funktioniert auch der Lauflicht –Test im Testprogramm nicht korrekt.

Haben sie die Werte verändert, dann klicken Sie auf **Write FS** und **Write FSH**.

Dadurch sollte der Controller auf 8 Mhz oder 16 Mhz umgeschaltet werden, je nachdem was für einen Quarz sie eingelötet / eingesteckt haben. Benötigt man sehr hohe Geschwindigkeiten bei der RS232 Schnittstelle, dann sind übrigens krumme Quarzwerte günstiger (siehe Roboternetz.de).

Alternativ kann man auch das beliebte Pony-Program verwenden, dazu mehr in dem Artikel „AVR leicht gemacht“ im Roboternetz im unter Artikeln <http://www.roboternetz.de/phpBB2/artikeluebersicht.php>

Jetzt muss noch die Quarzfrequenz im letzten Beispiel wieder mit der Anweisung `$crystal = 8000000` oder halt `$crystal = 16000000` angegeben werden und schon können sie erneut compilieren und das Programm übertragen. Klappt alles genauso wie vorher, dann sollte alles perfekt funktioniert haben.

Achtung: Eine falsche Einstellung der Fuse-Bits kann den Controller unbrauchbar machen!

Nun können wir über die Board Tasten folgende Funktionen auslösen:

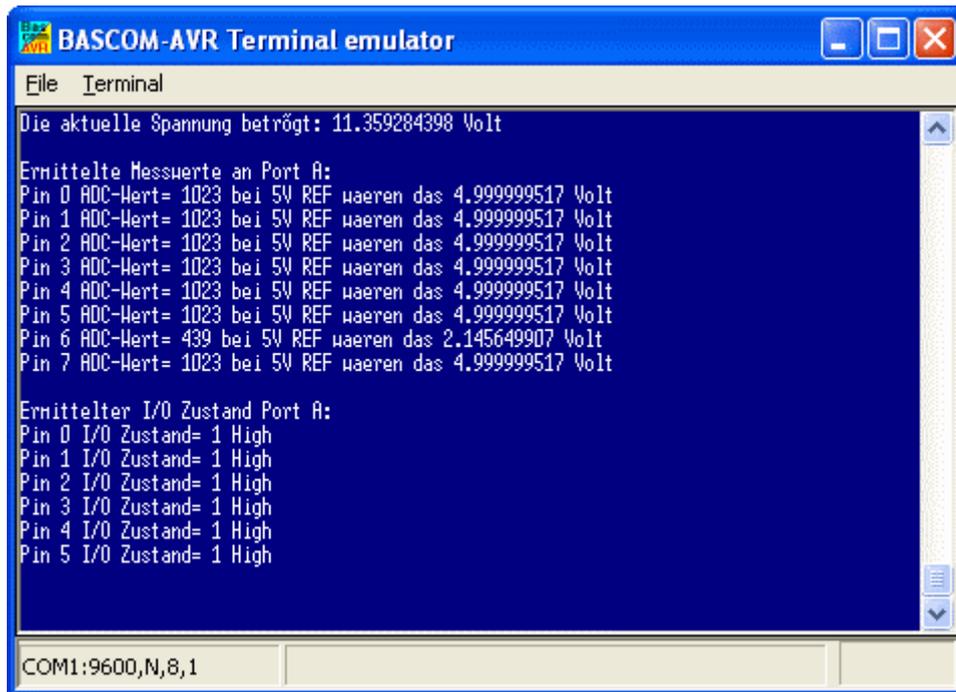
Taste 1	Zeigt Batteriespannung über RS232 an
Taste 2	Angeschlossene Motoren beschleunigen und abbremsen
Taste 3	Einige male Lauflicht über LED's anzeigen. Am I2C-Bus darf in diesem Moment nichts angeschlossen sein
Taste 4	Zeigt analoge Messwerte an allen Port A PIN's über RS232 an
Taste 5	Zeigt digitalen I/O Zustand von PA0 bis PA5 an

RS232

Viele der zuvor genannten Funktionen geben die Daten auf der RS232-Schnittstelle aus. Dazu muß natürlich die RS232-Schnittstelle über ein 3 poliges Kabel mit der 3 poligen Stiftleiste RS232 auf dem Board verbunden werden. Wer zufällig einen Conrad CCRP5 Zuhause haben sollte, der kann dazu das mitgelieferte RS232 Kabel verwenden da unsere RS232 Stiftleiste pinkompatibel ist. Ist dies nicht der Fall kann man sich recht einfach eines selbst löten.

Da in der Mitte GND und an den Außenseiten RX/TX liegt, schadet es in der Regel nicht wenn es mal falsch aufgesteckt werden sollte. Allerdings tut sich dann auch nix ;-). Also im Zweifel auch mal umstecken.

Danach kann man über Menü Tools von Bascom (oder Modem Symbol) einen Terminalemulator starten. Wenn man nun Taste 1, dann 4 und danach 5 klickt, dann sollte es in etwa so auf dem Bildschirm aussehen:



```
BASCOM-AVR Terminal emulator
File Terminal
Die aktuelle Spannung betrögt: 11.359284398 Volt

Ermittelte Messuernte an Port A:
Pin 0 ADC-Wert= 1023 bei 5V REF waeren das 4.999999517 Volt
Pin 1 ADC-Wert= 1023 bei 5V REF waeren das 4.999999517 Volt
Pin 2 ADC-Wert= 1023 bei 5V REF waeren das 4.999999517 Volt
Pin 3 ADC-Wert= 1023 bei 5V REF waeren das 4.999999517 Volt
Pin 4 ADC-Wert= 1023 bei 5V REF waeren das 4.999999517 Volt
Pin 5 ADC-Wert= 1023 bei 5V REF waeren das 4.999999517 Volt
Pin 6 ADC-Wert= 439 bei 5V REF waeren das 2.145649907 Volt
Pin 7 ADC-Wert= 1023 bei 5V REF waeren das 4.999999517 Volt

Ermittelter I/O Zustand Port A:
Pin 0 I/O Zustand= 1 High
Pin 1 I/O Zustand= 1 High
Pin 2 I/O Zustand= 1 High
Pin 3 I/O Zustand= 1 High
Pin 4 I/O Zustand= 1 High
Pin 5 I/O Zustand= 1 High

COM1:9600,N,8,1
```

Kommt garnix, dann ist eventuell das Kabel falsch herum auf die Stiftleiste aufgesteckt worden. Kommen wirre Zeichen, dann ist vermutlich im Terminalprogramm noch keine 9600 Baud eingestellt. Es kann jedoch auch an der falsch eingestellter Quazfrequenz liegen.

Bei 1 Mhz (ohne Quarz) sollte man nach Möglichkeit nicht mehr als 1200 Baud verwenden.

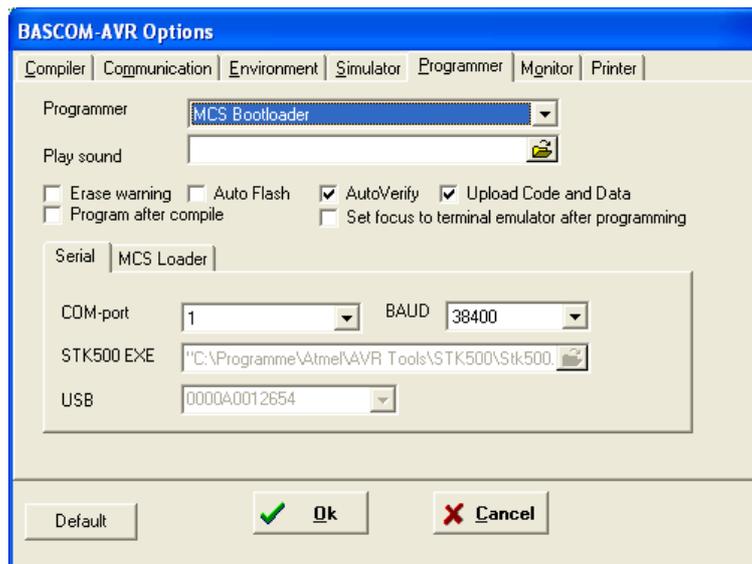
Über Taste 2 kann man die zwei angeschlossenen Motoren kurz anlaufen und wieder abbremesen lassen.

Per RS232 programmieren (Bootloader)

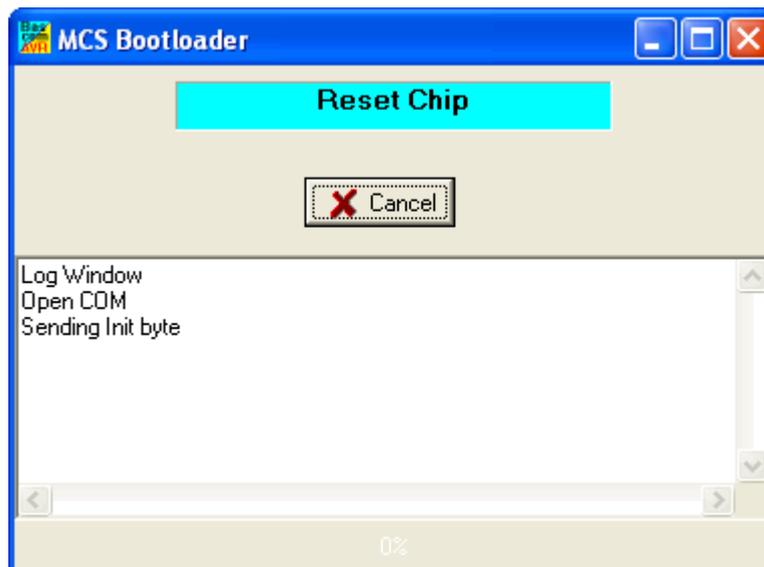
Bislang haben Sie erfahren das RN-Control mit einem ISP-Dongel programmiert werden kann. Neben dem ISP-Dongel der an den Druckerport angeschlossen wird, bieten einige Hertseller auch einen USBISP an. Dieser wird an einen freien USB-Port ihres Rechners angeschlossen und erlaubt eine noch schnellere Programmierung. Der Ablauf ist der gleiche, es muss lediglich in der Bascom Konfiguration der USB-Programmer gewählt werden.

Wenn Sie RN-Control nach dem 5.März 2007 gekauft haben, dann gibt es noch eine dritte Möglichkeit, denn seit diesem Datum wird das Board mit vorinstalliertem Bootloader geliefert. Man erkennt das auch daran das auf dem roten Aufkleber das Wort "BOOTLOADER" auftaucht. Auch diese Version des Boardes kann ganz normal per ISP oder USBISP programmiert werden. Neu hinzugekommen ist jedoch die Programmierung per RS232 ganz ohne ISP-Dongel. Allerdings funktioniert diese Vorgehensweise nur in Bascom Basic.

Dies geht ganz einfach. Dazu müssen Sie in der Konfiguration von Bascom folgendes einstellen.



Wenn nun ein kompiliertes Programm auf das Board RN-Control übertragen werden soll, so klicken Sie wie im letzten Kapitel beschrieben, auf das grüne Symbol "Program Chip". Alternativ können Sie auch die Taste F4 in Bascom betätigen. Im Gegensatz zur ISP-Programmierung kommt nun ein anderer Dialog:



Jetzt müssen Sie nur noch die RESET Taste am Board betätigen und das Programm wird automatisch übertragen und gestartet. Natürlich müssen Sie das RS232-Kabel mit dem dreipoligen Stecker richtig herum am Board angesteckt haben, im Zweifel einfach den dreipoligen Stecker umdrehen.

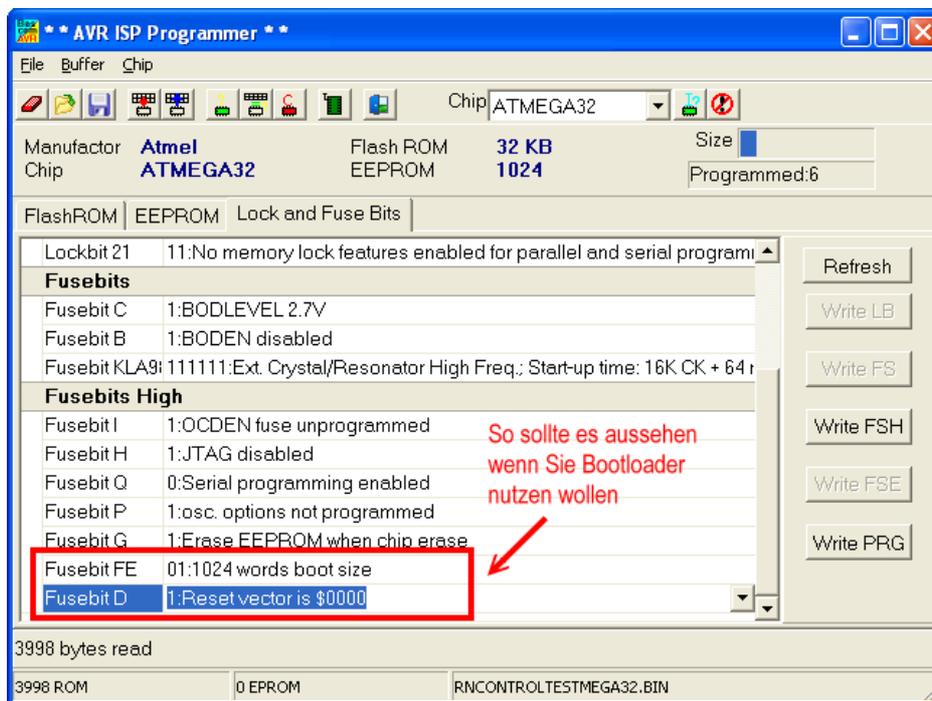
Die Programmierung per RS232 geht ebenfalls sehr fix und ist durchaus recht bequem. Sie können im übrigen auch folgende Programmzeile in Ihrem Quellcode (irgendwo am Anfang) einfügen:

```
$programmer = 13 '13=MCS Bootloader
```

Dadurch können Sie sich das ändern des Programmers in der Bascom Konfiguration sparen, denn so weiss Bascom automatisch das Sie das Programm über die RS232 Schnittstelle übertragen wollen

Leider werden derzeit keine EEPROM sondern nur Flash Daten per Bootloader übertragen. Wenn Ihr Programm keine EEPROM-Data Anweisung enthält (was die Regel ist), dann reicht dies völlig aus. Dennoch würde ich ihnen den Kauf eines ISP und USBISP's noch empfehlen, denn es gibt durchaus Anwendungsfälle wo EEPROM Daten oder auch Fusebit Einstellungen geändert werden sollen, dies geht derzeit mit dem Bootloader per RS232 nicht. Zudem sparen Sie ca. 2000 Byte Speicherplatz wenn sie keinen Bootloader nutzen.

Sollten Sie noch ein RN-Control ohne vorprogrammierten Bootloader besitzen, so kann der Bootloader auch per ISP nachinstalliert werden. Auf der Seite <http://www.mikrocontroller-elektronik.de/> ist er als Hex vorhanden. Wichtig ist jedoch das nach der Installation des Bootloaders noch zwei Fusebit Einstellungen verändert werden. Diese sollten in Bascom dann wie folgt aussehen:



Damit haben Sie den ersten Einstieg erfolgreich abgeschlossen.

Wenn Sie das Demoprogram gründlich studieren werden Sie viele Sachen davon ableiten und in eigenen Programmen verwenden können. Der Mikrocontroller bietet natürlich noch eine ganze Reihe weiterer Features, aber dies alles zu Beschreiben würde ganze Bücher füllen. Daher würde ich Ihnen für den tieferen Einstieg noch folgende Empfehlungen geben:

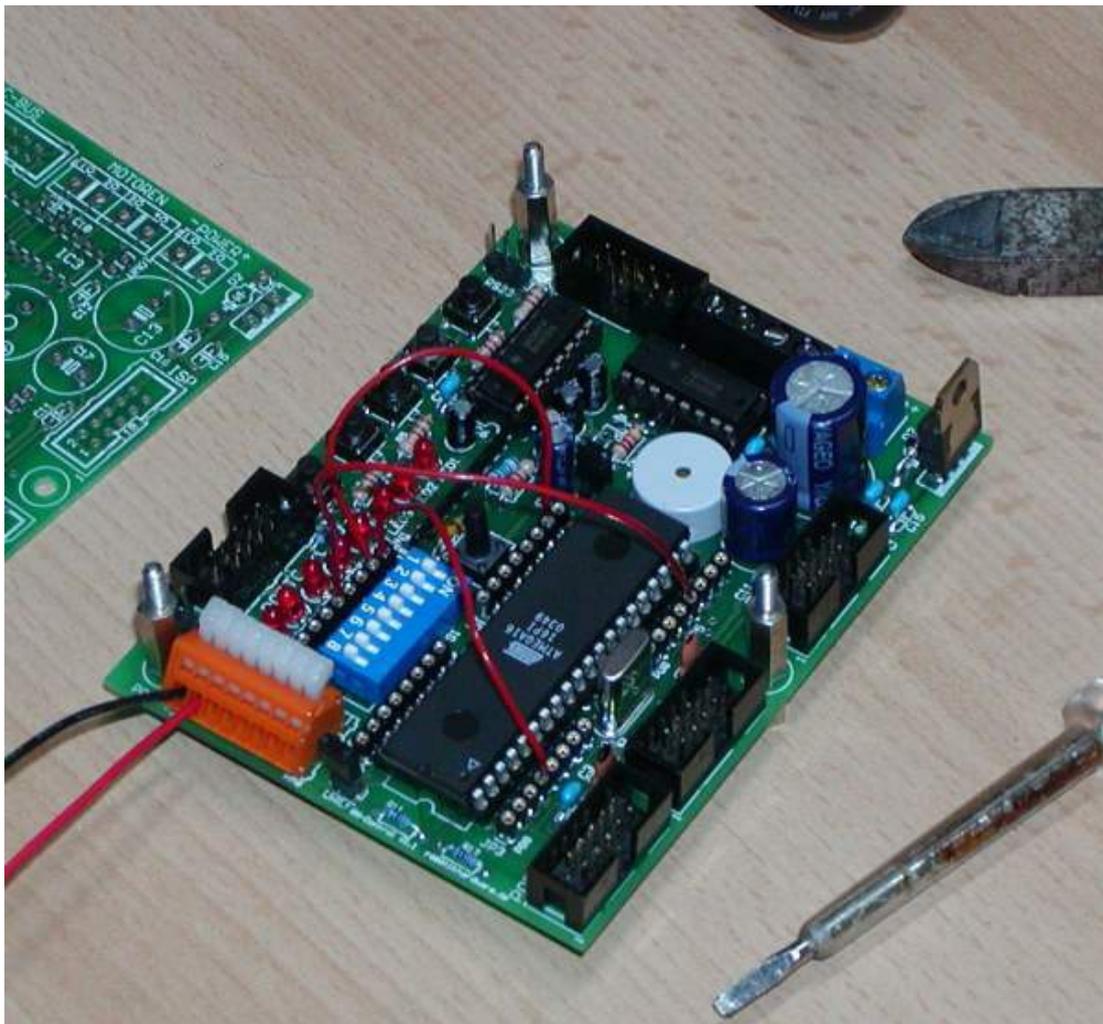
Buch: AVR-Mikrocontroller Lehrbuch von Roland Walter Deutsch	Das Buch führt leicht verständlich in die Welt der AVR-Mikrocontroller ein. Systematisch, Schritt für Schritt, mit der Hochsprache Basic und vielen gut kommentierten Beispiel-Listings. Was auch erwähnt werden muß: Der Stoff ist dicht und das Buch verzichtet auf "Seitenschinderei". Siehe Literaturempfehlungen auf http://www.mikrocontroller-elektronik.de/
Literatur Empfehlungen in RN-Wissen http://www.roboternetz.de	Literatur Empfehlungen und Links unter RN-Wissen.de
	Das inzwischen größte deutsche Forum was sich mit Robotik- und Mikrocontrollern sowie auch der Bascom-Programmierung beschäftigt. Hier findet man schnell Hilfe. Aber auch eigene Beiträge sind willkommen. Hier findet man auch das ausführliche Datenblatt zum Mega 16 und 32 im Download-Bereich. Im Artikelbereich findet man viele Grundlagen und Schaltungen.
http://www.mikrocontroller-elektronik.de/	Das Elektronik Blog-Magazin mit vielen CC-Projekten

Zusammenfassung der wichtigsten Board-Adressen

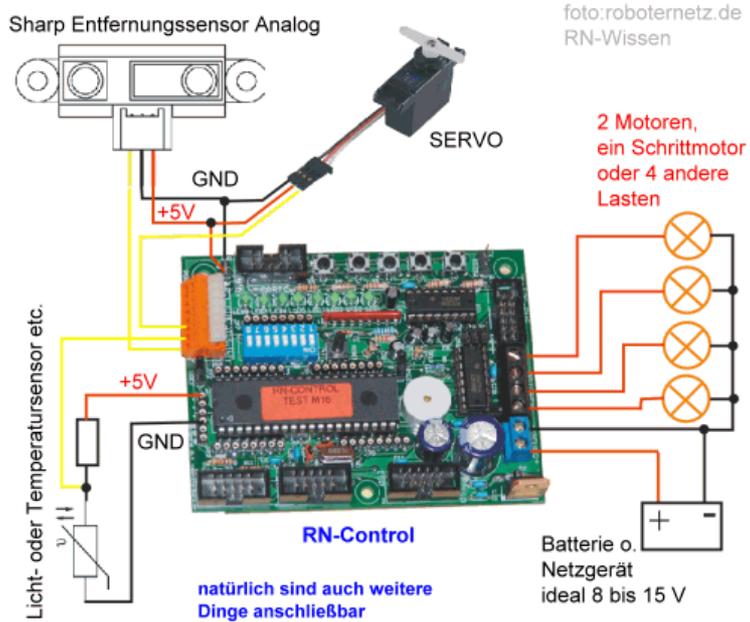
Funktion	Port	Logische Bedeutung und Beispiel
Linker Getriebemotor Ein/ Aus	PD4	PD4 = 0=Aus 1=Ein Bascom-Beispiel: <i>Config Pind.4 = Output</i> <i>Portd.4 = x</i> Im PWM-Mode: Pwm1a = 0 oder Geschwindigkeitl 0-1023
Linker Getriebemotor Drehrichtung	PC6 PC7	Erste Richtung: PC6=0 PC7=1 Zweite Richtung PC6=1 PC7=0 Bascom-Beispiel: <i>Config Pinc.6 = Output</i> <i>Config Pinc.7 = Output</i> <i>Portc.6 = 1</i> <i>Portc.7 = 0</i>
Rechter Getriebemotor Ein/ Aus	PD5	PD5 = 0=Aus 1=Ein Bascom-Beispiel: <i>Config Pind.5 = Output</i> <i>Portd.5 = x</i> Im PWM-Mode: Pwm1b = 0 oder Geschwindigkeitl 0-1023
Rechter Getriebemotor Drehrichtung	PB0 PB1	Erste Richtung: PB=0 PB=1 Zweite Richtung PB=1 PB=0 Bascom-Beispiel: <i>Config Pinb.0 = Output</i> <i>Config Pinb.1 = Output</i> <i>Portb.0 = 1</i> <i>Portb.1 = 0</i>
RS232 TX	PD1	Bascom-Beispiel: Print „Hallo“
RS232 RX	PD0	Bascom-Beispiel: Liest 3 Byte ein: <i>Dim a as Byte, c as integer</i> <i>Inputbin a,c</i> Oder: <i>Input „warte auf zwei zeichen“,x,y</i>
AD Port0 frei	PA0	Bascom-Beispiel: <i>Config Adc = Single , Prescaler = Auto</i> Start Adc <i>W = Getadc(0)</i>
AD Port1 frei	PA1	Wie zuvor
AD Port2 frei	PA2	Wie zuvor
AD Port3 frei	PA3	Wie zuvor
AD-Port4 frei	PA4	Wie zuvor
AD-Port5 frei	PA5	Wie zuvor
AD-Port6 Batteriespannung oder Frei	PA6	<i>Siehe Demo</i>
AD-Port7 Tastenabfrage	PA7	<i>Siehe Demo</i>

I2C BUS SCL	PC0	Bascom-Beispiel: <i>I2cstart</i> <i>I2cwbyte slaveadr</i> <i>I2cwbyte 15</i> <i>I2cstop</i>
I2C BUS SDA	PC1	Bascom-Beispiel: <i>I2cstart</i> <i>I2cwbyte slaveadr</i> <i>I2cwbyte 15</i> <i>I2cstop</i>
I2C BUS INT	PB2	Über Interrupt können hier Änderungen an I2C Erweiterungen gemeldet werden. Wird derzeit nur von wenigen Komponenten unterstützt. Wird in Jp6 kein Jumper eingesteckt, ist diese Leitung frei verfügbar!
ISP (SPI-Interface)	PB7, PB6, PB5	Diese Leitungen werden für Programmieradapter und andere SPI-Erweiterungen genutzt
LED's	PC0 bis PC7	Per DIP-Schalter zuschaltbare LED's
Lautsprecher	PD7	Sound Portd.7 , 400 , 450
Ports die immer frei sind, auch dann wenn alle Funktionen des Boards genutzt werden (Motoren, Sound, I2C, Tasten, Batteriespannungsüberwachung, ISP etc.)	PD2, PD3, PD6, PC2, PC3, PC4, PC5, PB2, PB3, PB4, PA0, PA1, PA2, PA3, PA4, PA5	

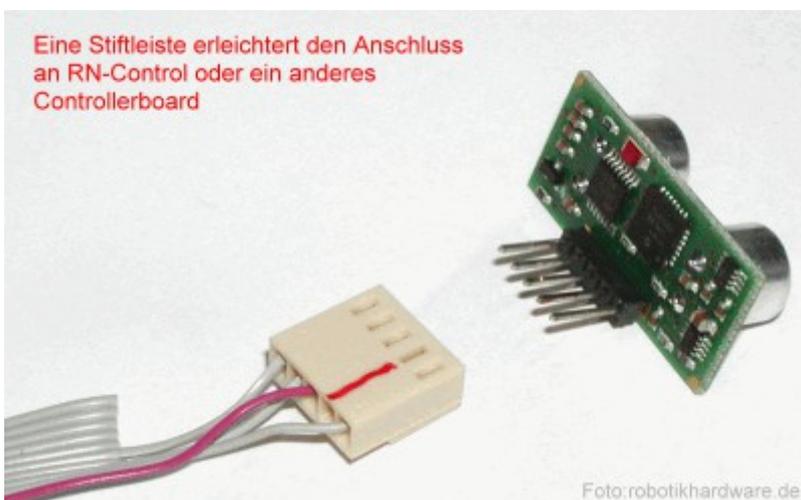
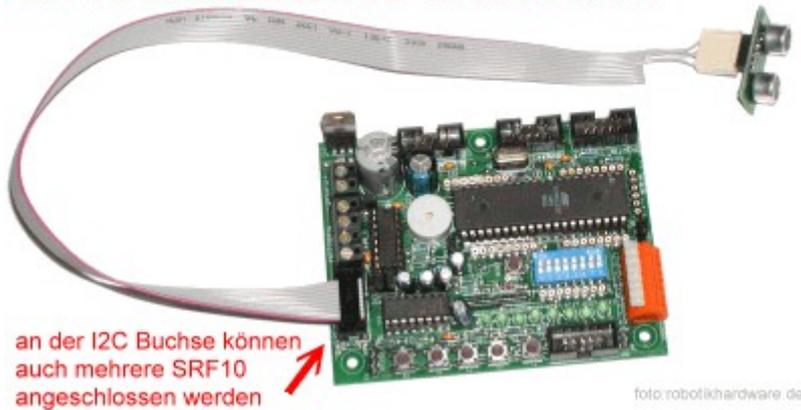
So können auch Drähte einfach eingesteckt werden:



Beispiel wie man Sensoren /Aktoren anschließt

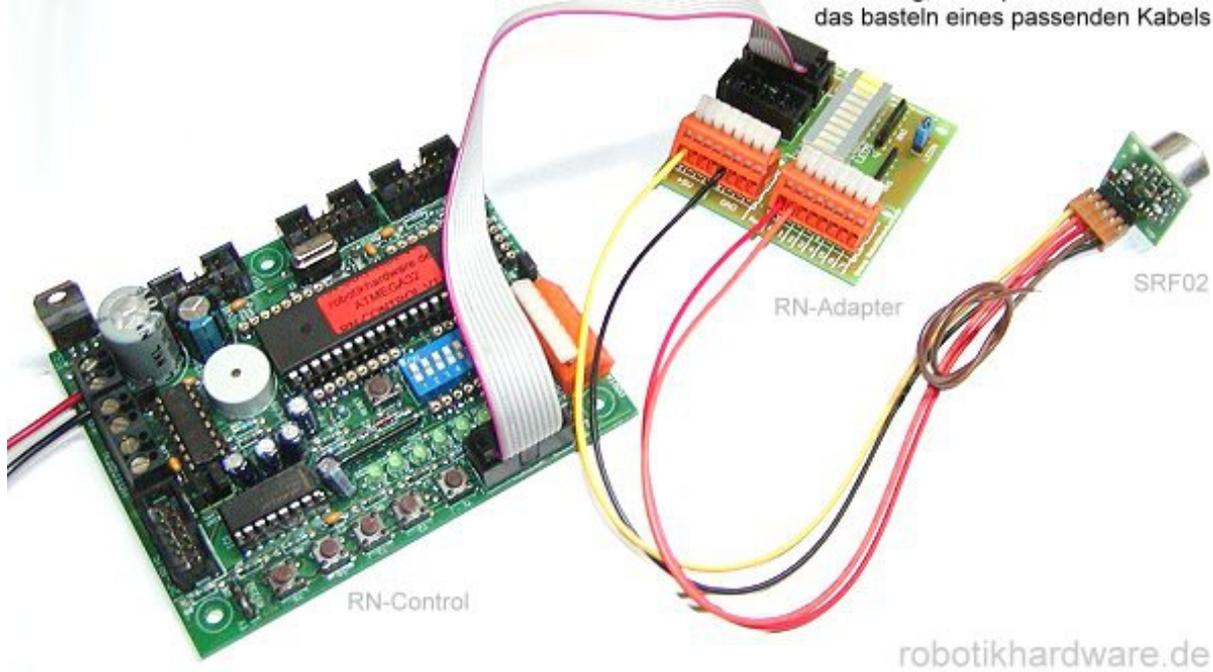


SRF10 Ultraschallsensor an RN-Control



Ultraschall SRF02 an RN-Control

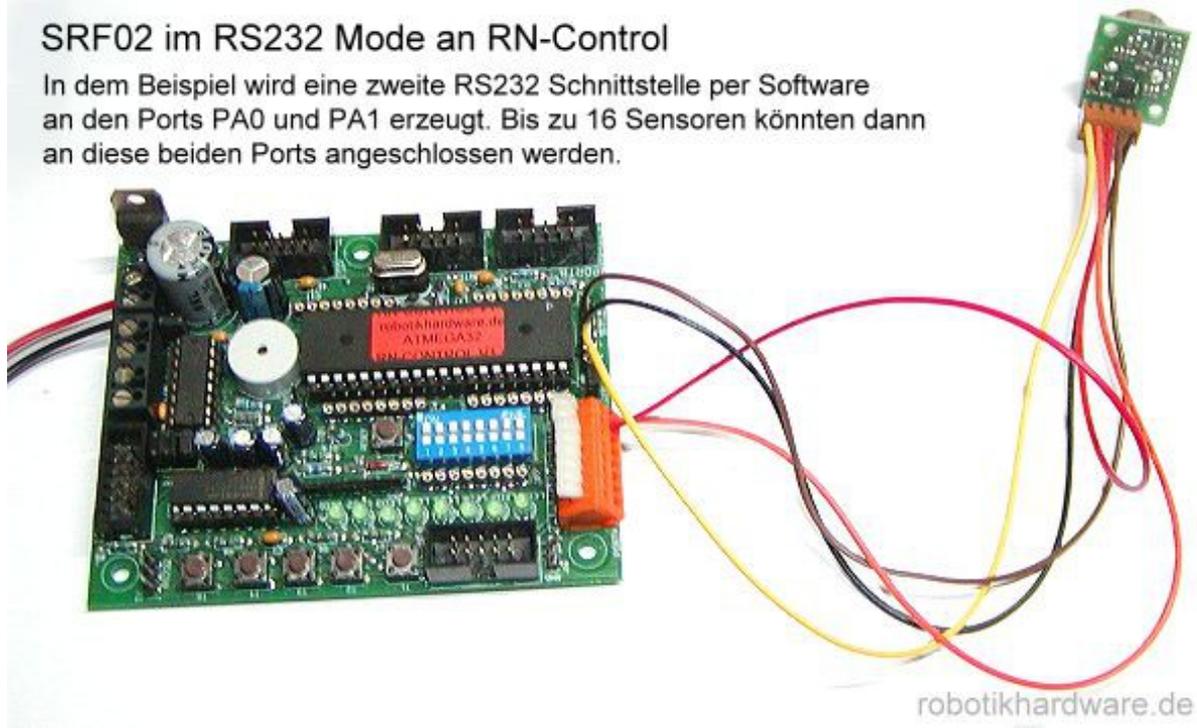
Der Adapter ist nicht unbedingt notwendig, er erspart einem nur das basteln eines passenden Kabels



robotikhardware.de

SRF02 im RS232 Mode an RN-Control

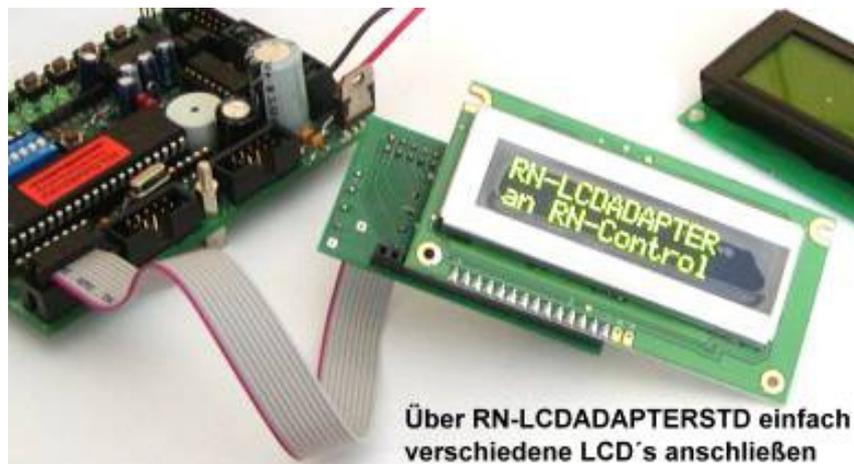
In dem Beispiel wird eine zweite RS232 Schnittstelle per Software an den Ports PA0 und PA1 erzeugt. Bis zu 16 Sensoren könnten dann an diese beiden Ports angeschlossen werden.



robotikhardware.de

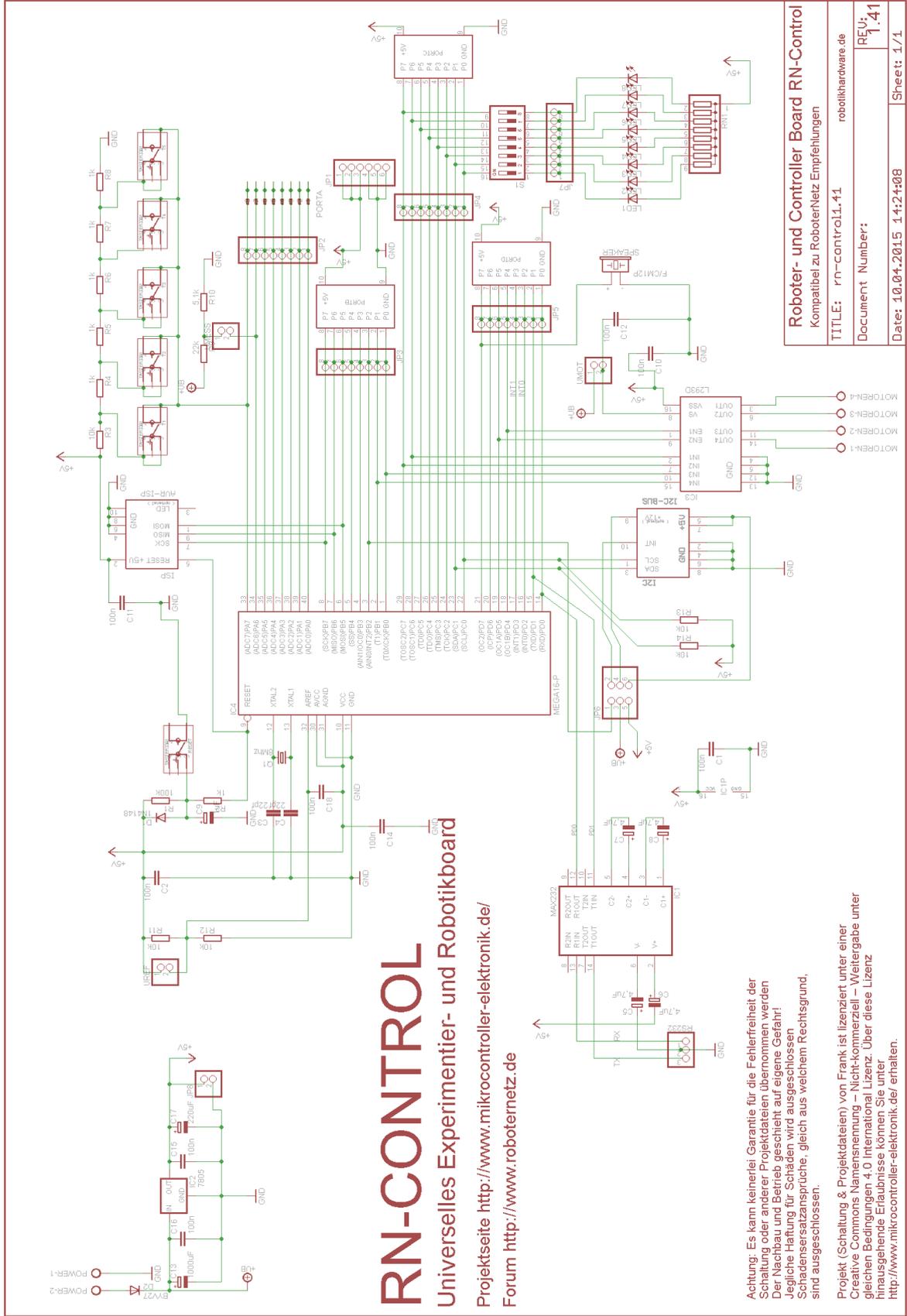
Verschiedene LCD's an RN-Control

Es lassen sich zahlreiche LCD's an RN-Control betreiben. Hier einige LCD's :



Nähere Tips dazu auf Seite <http://www.roboternetz.de/wissen/index.php/RN-Control>
Bezugsquellen für Sensoren finden sie auf [unserer Seite](#)

Schaltplan



RN-CONTROL Universelles Experimentier- und Robotikboard

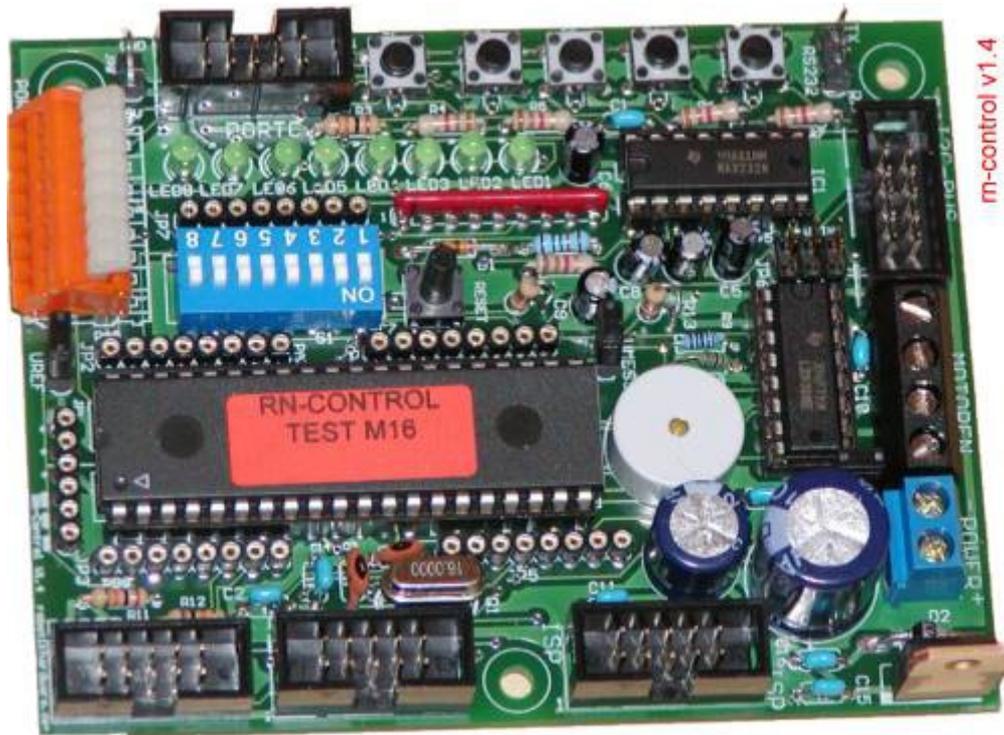
Projektseite <http://www.mikrocontroller-elektronik.de/>
 Forum <http://www.roboternetz.de>

Achtung: Es kann keinerlei Garantie für die Fehlerfreiheit der Schaltung oder anderer Projektdateien übernommen werden. Der Nachbau und Betrieb geschieht auf eigene Gefahr! Jegliche Haftung für Schäden wird ausgeschlossen. Schadensersatzansprüche, gleich aus welchem Rechtsgrund, sind ausgeschlossen.

Projekt (Schaltung & Projektdaten) von Frank ist lizenziert unter einer Creative Commons Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. Über diese Lizenz hinausgehende Erlaubnisse können Sie unter <http://www.mikrocontroller-elektronik.de/> erhalten.

Roboter- und Controller Board RN-Control Kompatibel zu RoboterNetz Empfehlungen	
TITLE: rn-control.1.1	robotikhardware.de
Document Number:	REU: 1.1
Date: 10.04.2015 14:24:08	Sheet: 1/1

Sollte in dieser Doku noch der ein oder andere Fehler drin stecken, so bitte ich um Nachsicht und Hinweise per [Kontaktformular](http://www.mikrocontroller-elektronik.de/Kontaktformular) über <http://www.mikrocontroller-elektronik.de/>



Projekt-Seite zum herunterladen: <http://www.mikrocontroller-elektronik.de/>
Forum-Seite für Fragen: <http://www.roboternetz.de/community/forum.php>

Weiterentwicklungen können gerne ebenfalls auf der [Projektseite](#) unter CC-Lizenz veröffentlicht werden!