

# Das kompakte Board

Ideal für Atmel AVR Mega8,168 oder 328



## RN-MEGA 8 Plus V1.5

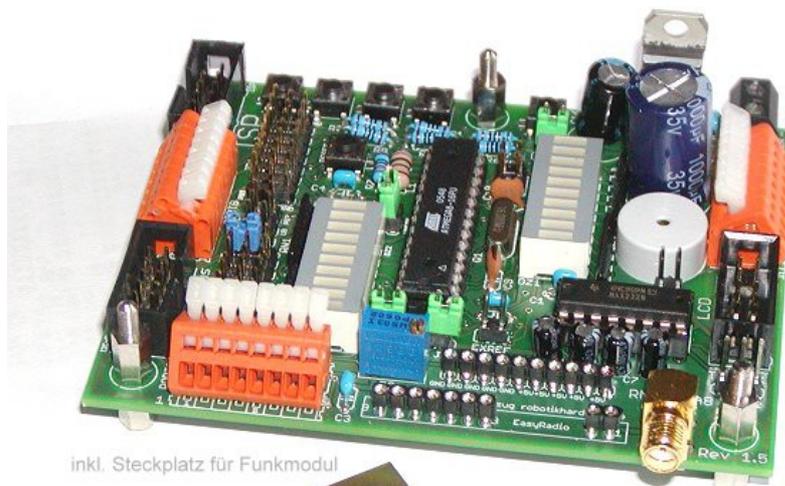
Dieses Board wurde speziell zum Experimentieren mit dem Mikrocontroller AT-MEGA 8, 168 oder 328 (bekannt vom Arduino Uno) und andere pincompatible Controller entworfen. .

Der Controller Mega8 als auch Mega168 eignet sich wegen seines günstigen Preises als auch geringen Baugröße für zahlreiche Aufgaben, bei denen ein Mega16 oder Mega32 oder gar ein Raspberry Pi völlig überdimensioniert oder einfach zu groß wäre. Im Bereich Robotik kann ein Avr wie der Mega 8 oder 328 ideal auch als Co-Controller für Servosteuerung, Motorsteuerung, Display Ausgabe, Sensorüberwachung und viele mehr eingesetzt werden. Oft sind kaum externe Bauteile notwendig.

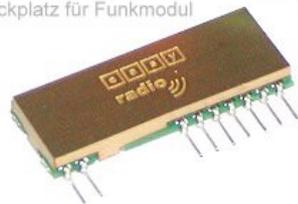
Um den Controller jedoch für eine spezielle Aufgabe programmieren zu können, bedarf es einer Entwicklungsumgebung die quasi alle Ports steckbar zugänglich macht, also auch die visuelle Überwachung der Port-Zustände erlaubt.

Für diese Aufgabe ist RN-MEGA8Plus ideal! Ganze 20 Ports können gleichzeitig visuell über Leuchtbalken überwacht werden. Nahezu alle Ports sind über einfache Steckklemmen erreichbar. Zudem verfügt das Board über einen genormten LCD-Display-, I2C-Bus-, RS232-, Servo- und ISP-Anschluss.

Eine Besonderheit von RN-MEGA8Plus ist der Steckplatz für ein EasyRadio Funkmodul. Dadurch wird das Board funkkompatibel zu unserem Projekt RN-Mega128Funk, RN-Steuerung und RN-Funk. Somit können Daten mit anderen Boards oder PC per Funk ausgetauscht werden. Das Funkmodul ist optional im Handel beziehbar – es wird nur eingesteckt und man kann loslegen.



inkl. Steckplatz für Funkmodul



RN-Mega8Plus

Experimentierboard für AVR  
ATMEGA8 und ATMEGA168

[robotikhardware.de](http://robotikhardware.de)

**Ich denke die Leistungsdaten und Features können überzeugen!**

**Anleitung aktualisiert am 20.4.2015**

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](http://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](http://RN-Wissen.de) in der Rubrik [Buchvorstellungen](#) zu finden. Für Fragen bezüglich Elektronik und Mikrocontroller empfehle ich das Forum: [Roboternetz.de](http://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](http://www.Roboternetz.de) & [www.Mikrocontroller-Elektronik.de](http://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](http://www.Roboternetz.de) & [www.Mikrocontroller-Elektronik.de](http://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 Zweifelsfalle 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/oder Anschlussfehler außerhalb unseres Einflusses liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.

## Als hier die Leistungsmerkmale auf einen Blick:

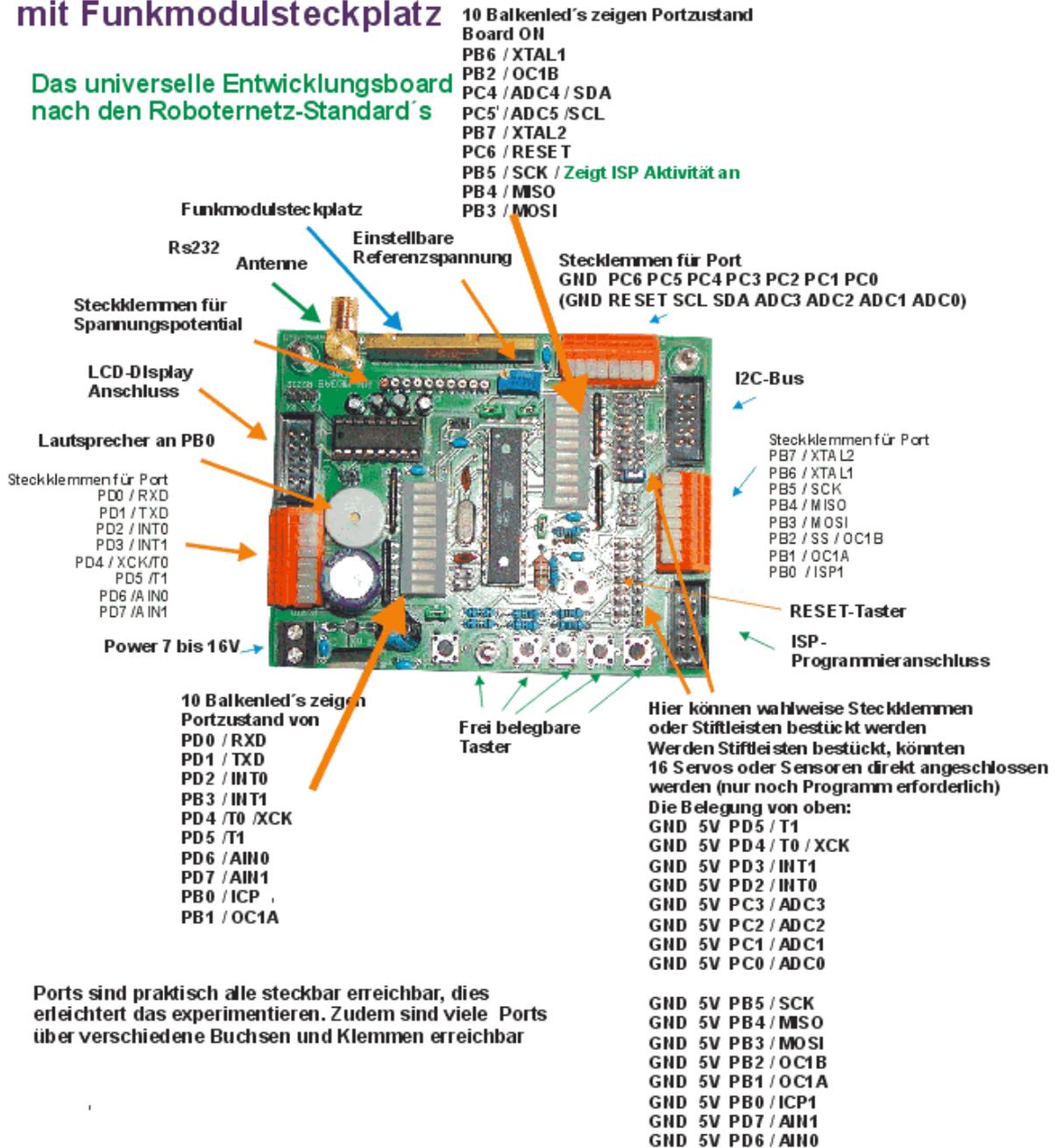
- Beliebiger Quarz kann bestückt werden
- Schneller AVR Mega 8 Mikrocontroller (bis zu 8K Flash Speicher, 1K Ram und 0,5K EEPROM) (optional auch Mega168 / 328 oder pinkompatibler AVR )
- 23 programmierbare I/O Pins, 6 AD Ports u.v.m.
- 20 Leuchtdioden über zwei Balkenanzeigen auf dem Board – über Jumper deaktivierbar
- [Steckplatz für ein EasyRadio Funkmodul](#)
- fast alle Portleitungen sind über Steckklemmen mit Hebelmechanik herausgeführt. Kein löten mehr, Drähte und Litze kann einfach eingeklemmt werden
- Alle Stecker RS232, ISP, LCD, I2C-BUS entsprechen der vereinbarten [Roboternetz-Definition](#)
- Potentiale (GND und +5V) sind mehrfach über Steckbuchsen erreichbar. Ideal zum experimentieren da einfach Drähte (ca. 0,5mm<sup>2</sup>) eingesteckt werden (kein Löten oder schrauben).
- Integrierter programmierbarer Mini-Lautsprecher um Töne auszugeben
- Ein Reset Taster
- 5 Taster für beliebige Verwendung. Sie belegen nur einen analogen Port!
- 5 V Spannungsstabilisierung mit 2 A Belastbarkeit (bei Kühlung), auch herausgeführt für Erweiterungen
- Eingangsspannung gegen Verpolung geschützt
- ISP-Programmer wird über Diode versorgt (da einige Programmer dies nicht eingebaut haben)
- RS232 mit normgerechtem Pegelwandler (MAX232) – PC direkt anschließbar
- Batteriespannung kann im Programm abgefragt werden
- ISP – Programmierschnittstelle für übliche AVR-Programmieradapter (10polig)
- Betriebsspannung wahlweise zwischen 7 und 16V (empfohlen 8 bis 14 V)
- Sehr kompakt, nur halbes Europaformat nach [Roboternetz-Definition](#) (ca. 100x78mm)
- I2C-Bus über die zahlreiche Erweiterungsplatinen anschließbar sind (z.B. Sprachausgabe RN-Speak, Relaisboard RN-Relais, Servoboards, LCD's, RN-Control uvm.).
- RN-MEGA8PLUS kann auch selbst als Slave Erweiterung für anderes Board (z.B. RN-Control) dienen
- Programmierbar in zahlreichen Sprachen, z.B. Basic ([BASCUM](#) Compiler, eingeschränkt bis 2K wird mitgeliefert), C (C-Compiler GCC wird mitgeliefert), Assembler, Pascal
- Deutsche Doku mit Basic Programmbeispiel
- Eagle Dateien frei zum Download
- Kein Starter- oder Applikationsboard notwendig – bereits alles integriert!

## Diagramm und Beschreibung der Anschlüsse

# RN-MEGA 8PLUS

## mit Funkmodulsteckplatz

Das universelle Entwicklungsboard nach den Roboternetz-Standard's



Es ist zu empfehlen diese Seite auszudrucken, da hier sehr gut die Portbelegungen ersichtlich sind.

# Aufbau und Anwendung

## Aufbau

Über die frei downloadbare Eagle Dateien lässt sich bei einem Platinenhersteller leicht eine eigene Platine für dieses Projekt bestellen. Einige Platinenanbieter werden auf unserer Seite <http://www.mikrocontroller-elektronik.de/> genannt! Der Aufbau der Schaltung ist durch ein vorgefertigte Platine eigentlich völlig problemlos auch von Elektronik-Einsteigern zu bewerkstelligen. Durch den Bestückungsdruck und die Bestückungsliste, etwas weiter hinten in dieser Dokumentation, ist der Aufbau recht unkritisch. Aufgrund moderner Bauteile hält sich die Anzahl der Kleinteile in Grenzen, weshalb die Schaltung meist in ca. 45 bis 90 Minuten aufgebaut ist.

### Dennoch einige Anmerkungen zu kleinen Hürden:

1. Das Board verwendet vier Widerstandsnetzwerke. Auf der Platine sind diese mit RN1 bis RN4 gekennzeichnet. Da diese Teile Einsteigern noch nicht so bekannt sind, möchte ich darauf hinweisen dass 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 (Frequenz) 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. In dieser Anleitung nutzen wir noch weitgehend einen Programmer für einen Druckerport, natürlich können Sie aber auch einen moderneren USB-Programmer nutzen. Am besten schauen Sie auch dazu auf unsere Seite <http://www.mikrocontroller-elektronik.de/>, dort werden wir einige vorstellen und testen. Sie können aber auch im [Roboternetz](#) andere User nach den entsprechenden Erfahrungen mit Programmern befragen!

Die 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.

**Bei den Beispielprogrammen wird in der Regel von einem eingelöteten 16 Mhz Quarz ausgegangen.**

3. Die Taster richtig herum einsetzen. Wir verwenden 4 polige Taster, das fünfte Loch auf der Platine wird nicht benötigt. Die Taster passen genau in die Löcher, die Pin's müssen nicht verbogen werden sondern der Taster muss nur mit leichtem Druck in die Platine eingedrückt werden.
4. Über den Poti R2 (Spannungsteiler) wird die Referenzspannung festgelegt. Diese ist später wichtig für das messen von analogen Spannungen. Über diesen Spannungsteiler kann diese beliebig eingestellt werden. Sie sollte zwischen 2,5 und 5V liegen und kann an der Stiftleiste JP2 gemessen werden. Wurde diese korrekt eingestellt, wird ein Jumper auf EXREF gesteckt. Alternativ kann man den Jumper auch weglassen und die Referenzspannung per Software programmieren. Ist man sich unsicher (Einsteiger), lässt man den Jumper erst mal weg, die Poti-Einstellung ist dann belanglos.
5. Das Board nutzt den Analog-Wandler des Controllers auch zur Batteriespannungsüberwachung. Soll diese Funktion über Jumper "UMESSPORT" aktiviert werden, dann ist darauf zu achten das auch der Spannungsteiler R9 und R10 richtig dimensioniert ist. Die hier abfallende Spannung darf nicht höher als die zuvor erwähnte Referenzspannung sein.  
Einige Beispiele:  
R9=22 k  
R10= 5,1K  
Bei diesen Werten würde bei 13V Eingangsspannung 2,5V abfallen. Wenn man eine Referenzspannung von 2,5V eingestellt hat, dann sollte die Eingangsspannung nicht höher sein, da ansonsten der analoge Port beschädigt werden kann.

**Diese Werte werden in den Beispielen verwendet. Werden andere Werte eingesetzt, so muß das Beispielprogramm angepaßt werden.**

R9=22 k

R10= 2,2K

Bei diesen Werten dürfte Eingangsspannung bis ca. 27V hoch sein, um die Referenzspannung von 2,5V zu erreichen

Die abfallende Spannung berechnet sich so:

$I = \text{Eingangsspannung} / (R9 + R10)$

$U_{\text{ref}} = I * R10$

6. Zu beachten ist noch das die Balken-LED-Anzeigen nicht falsch herum eingelötet werden. Die Anode ist auf der Seite wo die Beschriftung KINGBRIGHT steht. Wenn man den Bestückungsplan so vor sich liegen hat, das der Ein-/Ausschalter rechts ist, dann muß bei der linken Balkenanzeige die Schrift auch links sein und bei der rechten Balkenanzeige muß die Schrift rechts sein. Im Zweifel kann man auch mit einem Multimeter Anode und Kathode bestimmen
7. RN-Mega8Plus verfügt über einen Platz für ein **Funkmodul**. Wird dieses nicht benötigt, läßt man den Platz einfach unbelegt. Man kann ein solches Funkmodul wahlweise direkt einlöten oder in eine Buchsenleiste stecken. Ich empfehle eine 7 und 2 polige Buchsenleiste einzulöten. Eine solche liegt dem Bausatz bei. Da diese im Handel länger sind, muss diese mit einer Zange oder kleinen Säge (Laubsäge) auf die entsprechende Pinzahl gekürzt werden.
8. Soll statt des vorhandenen RS232 Anschlusses ein Funkmodul eingesetzt werden, so ist das IC1 (MAX232) nicht mehr notwendig und muss aus der Fassung gezogen werden.
9. Auf der Platine ist bei dem Piezo Lautsprecher die Polarität "+" gekennzeichnet. Bei dem mitgelieferten Lautsprecher müssen Sie dies nicht beachten, der Piezo Lautsprecher kann beliebig herum 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.

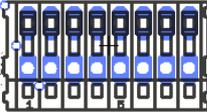
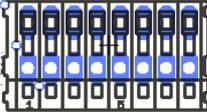
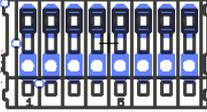
Nach dem Aufbau sollten Sie nochmals alle Lötunkte kontrollieren. Wenn Sie dann Spannung anlegen, dann sollten deutlich weniger als 100mA Strom fließen. Ist der Strom höher, dann deutet das auf ein Lötfehler hin.

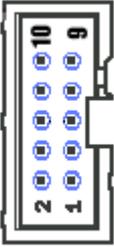
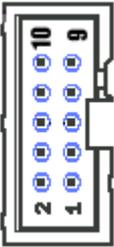
Das mitgelieferte Testprogramm kann dann recht einfach mit Bascom oder Pony übertragen werden. Per Tastendruck können anschließend verschiedene Boardeigenschaften überprüft werden.

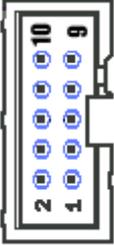
Gibt es Probleme bei der Übertragung des Programmes, so empfehle ich folgende Hilfeseite:

[http://www.roboternetz.de/wissen/index.php/RN-Board\\_FAQ-Seite](http://www.roboternetz.de/wissen/index.php/RN-Board_FAQ-Seite)

## Erläuterung der Anschlüsse, Regler und Kurzschlussbrücken

Anschluss- Bezeichnung	Erläuterung
<p><b>Port B</b></p> 	<p><b>Digitaler I/O Port</b> Über eine Steckleiste werden hier die 8 Portleitungen PB0 bis PB8 zur Verfügung gestellt. Anschlussdrähte können einfach eingesteckt werden, indem man mit einem Kugelschreiber oder Schraubenzieher den oberen weißen Hebel etwas nach unten drückt.</p> <p>Die genaue Belegung ist auch aus dem Diagramm weiter vorne zu entnehmen</p> <p>Die Belegung sieht wie folgt aus (von links nach rechts von vorne gesehen):</p> <p><b>Pin 1</b> PB0 / ISP1  <b>Pin 2</b> PB1 / OC1A  <b>Pin 3</b> PB2 / SS / OC1B  <b>Pin 4</b> PB3 / MOSI  <b>Pin 5</b> PB4 / MISO  <b>Pin 6</b> PB5 / SCK  <b>Pin 7</b> PB6 / XTAL1  <b>Pin 8</b> PB7 / XTAL2</p> <p>Einige der Ports werden auch für den Quarz und die ISP-Programmierschnittstelle genutzt. Wird ISP jedoch gerade nicht benutzt als auch der interne Quarzoszillator verwendet, dann sind diese Ports alle frei.</p>
<p><b>Port C</b></p> 	<p><b>Digitaler I/O Port und analoger Port</b> Über eine Steckleiste werden hier 7 Portleitungen PC0 bis PC6 zur Verfügung gestellt. Anschlussdrähte können einfach eingesteckt werden, indem man mit einem Kugelschreiber oder Schraubenzieher den oberen weißen Hebel etwas nach unten drückt.</p> <p>Die genaue Belegung ist auch aus dem Diagramm weiter vorne zu entnehmen</p> <p>Die Belegung sieht wie folgt aus (von links nach rechts von vorne gesehen):</p> <p><b>Pin 1</b> PC0 / Analoger Port 0  <b>Pin 2</b> PC1 / Analoger Port 1  <b>Pin 3</b> PC2 / Analoger Port 2  <b>Pin 4</b> PC3 / Analoger Port 3  <b>Pin 5</b> PC4 / I2C-Bus SDA  <b>Pin 6</b> PC5 / I2C-Bus SCL  <b>Pin 7</b> PC6 / RESET  <b>Pin 8</b> GND</p> <p>Einige der Ports werden auch für den I2C-Bus als auch den Controller Reset genutzt.</p>
<p><b>Port D</b></p> 	<p><b>Digitaler I/O Port</b> Über eine Steckleiste werden hier die 8 Portleitungen PD0 bis PD8 zur Verfügung gestellt. Anschlussdrähte können einfach eingesteckt werden, indem man mit einem Kugelschreiber oder Schraubenzieher den oberen weißen Hebel etwas nach unten drückt.</p> <p>Die genaue Belegung ist auch aus dem Diagramm weiter vorne zu entnehmen</p> <p>Die Belegung sieht wie folgt aus (von links nach rechts von vorne gesehen):</p> <p><b>Pin 1</b> PD0 / RXD  <b>Pin 2</b> PD1 / TXD  <b>Pin 3</b> PD2 / INT0  <b>Pin 4</b> PD3 / INT1  <b>Pin 5</b> PD4 / XCK / T0  <b>Pin 6</b> PD5 / T1  <b>Pin 7</b> PD6 / AIN0  <b>Pin 8</b> PD7 / AIN1</p>

<p><b>DISPLAY</b></p> 	<p>Einige der Ports werden auch für die RS232 Schnittstelle benutzt</p> <p><b>LCD Display-Anschluß nach Roboternetznorm</b> Über diesen 10 poligen Wannenstecker können gemäß der Roboternetz-Definition nahezu alle üblichen LCD-Display´s angeschlossen werden. Wahlweise auch Displays mit Beleuchtung, in dem Fall kann bei manchen Displays der Adpater .RN-LCDAdapter (siehe Robotikhardware.de) nützlich sein.</p> <p>Durch die einheitliche RN-Definition kann dieser Anschluss auch für viele andere Aufgaben genutzt werden. Passend wäre auch der RN-Adapter.</p> <p>Die genaue Belegung sieht wie folgt aus:</p> <p>Pin 1 DB7 PD5 Pin 2 DB6 PD4 Pin 3 DB5 PD3 Pin 4 DB4 PD2 Pin 5 EN2 PD7 (wird nur bei manchen LCDs benötigt) Pin 6 EN PC3 Pin 7 R/W PD6 Pin 8 RS PC2 Pin 9 GND Pin 10 +5V</p> <p>Bitte daran denken das diese Ports auch an den Steckklemen und an dem ISP-Wannenstecker genutzt werden. Da aber selten alle Pin´s / Ports bei einem Display verwendet werden, stehen die anderen auch bei angeschlossenem Display zur Verfügung</p>
<p><b>I2C-Bus</b></p> 	<p><b>I2C-Bus</b> Über diesen Bus lassen sich zahlreiche Erweiterungen an dieses Board anschließen. Zum Beispiel werden auf der Seite <a href="http://www.mikrocontroller-elektronik.de/">http://www.mikrocontroller-elektronik.de/</a> passende Projekte mit Sprachausgabe, Relais, Schrittmotorsteuerung etc. vorgestellt. Aber auch dieses Board kann selbst als Slave-Board, also als Erweiterung an ein anderes Hauptboard (wie z.B. RN-Control) angeschlossen werden. Der I2C-Bus benötigt nur 2 Leitungen für alle Funktionen. Entsprechend der Roboternetz-Definition wird hier ein 2x5 poliger Stecker angeschlossen. Die Belegung entspricht exakt der anderer Roboternetz Boards.</p> <p><b>Pin 1</b> SCL (Taktleitung) <b>Pin 3</b> SDA (Datenleitung) <b>Pin 5</b> +5V <b>Pin 7</b> +5V <b>Pin 9</b> Batteriespannung <b>Pin 2,4,6,8</b> GND <b>Pin 10</b> INT Diese Leitung kann von allen I2C-Bus Erweiterungen genutzt 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. 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) 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>Hinweis: Im Gegensatz zu RN-Control sind bei RN-MEGA noch keine Pullup-Widerstände für den I2C-Bus eingebaut. Wenn also noch kein anderes Board am I2C-Bus diese Pullup-Widerstände eingebaut hat, dann muss dies noch erfolgen. Dazu verbindet man einen 10K Widerstand mit +5V und dem Pin SDA. Zudem ein 10K Widerstand zwischen +5V und SCL.</p>
<p><b>ISP</b></p>	<p><b>ISP – IN SYSTEM PROGRAMMING</b> Über diesen Anschluss kann der <b>Controller auf dem Sprachboard</b> mit einem Standard ISP-</p>

	<p>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 „ISP-Programmieradapter“ unter <a href="http://www.roboternetz.de">www.roboternetz.de</a>).</p> <p><b>Pin 1</b> MOSI <b>Pin 2</b> VCC <b>Pin 3</b> Nicht belegt <b>Pin 4</b> GND <b>Pin 5</b> RESET <b>Pin 6</b> GND <b>Pin 7</b> SCK <b>Pin 8</b> GND <b>Pin 9</b> MISO <b>Pin 10</b> GND</p> <p>Achtung, beim Einstecken des ISP-Steckers darauf achten das dieser nicht versehentlich in den I2C-Bus gesteckt wird, dadurch kann Controller oder Programmieradapter beschädigt werden.</p>
Power	<p><b>Spannungsversorgung</b> Über diese Schraubklemme wird das Board mit Spannung versorgt. Es reicht eine unstabilisierte Gleichspannung von 7 bis 14V aus (max. 20V wenn ein Kühlkörper verwendet wird) + und – sind auf der Platine markiert. Das Board ist jedoch auch gegen ein verpolen geschützt, so das nichts kaputt geht!</p>
JPI2C	<p><b>I2C-Bus Belegung</b> Über drei Kurzschlussstecker können wahlweise die Batteriespannung (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 Schraubklemmen 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. Liegt am Power Stecker jedoch eine Spannung an und versorgt den I2C-Bus bereits ein anderes Board mit Spannung, dann dürfen diese Jumper (bis auf INT) nicht eingesteckt werden.</p>
POW5	<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>
UMESS	<p><b>Batteriespannungsmessung</b> Wenn diese Kurzschlussbrücke gesteckt ist, dann wird über Port PC1 die Batteriespannung überwacht. Ansonsten ist der Port frei!. Beachten Sie dazu die Hinweise zur Referenzspannung weiter vorne in der Anleitung.</p>
Antenne	<p><b>Funkmodul Antenne</b> Wird ein Funkmodul eingesteckt, kann hier entweder eine SMA-Antenne direkt angeschlossen werden. Alternativ kann man auch einfach ein kleines Stück Draht anlöten</p>
RS232	<p><b>PC kompatible RS232 Schnittstelle</b> Über ein Adapterkabel kann die serielle Schnittstelle des PC direkt mit dem Board verbunden werden. Dies ist dann sinnvoll, wenn ein Fehler in einem Programmen gesucht wird. 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 Roboternetz-Standard, als auch zum Conrad Roboter CCRP5:</p> <p><b>Pin 1</b> RX <b>Pin 2</b> GND <b>Pin 3</b> TX</p>

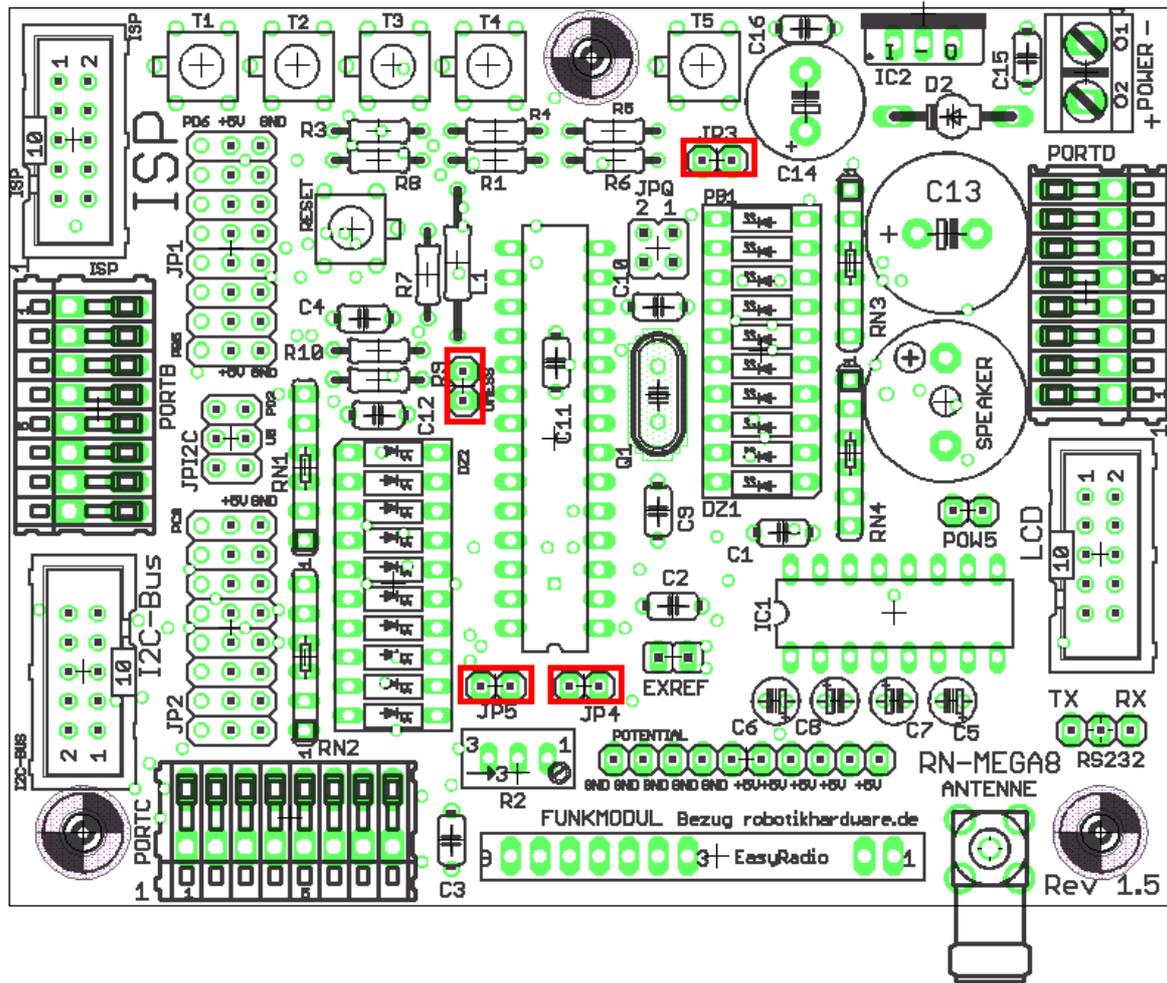
	Ein geeignetes Anschlusskabel kann schnell selbst angefertigt werden !
JP1	<b>Universelle PIN's</b> Hier können wahlweise drei 8 fache einpolige Stiftleisten oder 3 Buchsenleisten eingelötet werden. Man kann auch eine 2x8 und eine 1x8 polige Stiftleiste einlöten. Daraus würden sich dann 8 Servo kompatible Stecker ergeben. Es können also 8 handelsübliche Modellbauservos direkt angeschlossen werden. Mit einer geeigneten Software können sich diese dann natürlich auch steuern. In Verbindung mit JP2 können an das Board also 16 Servos direkt angeschlossen werden. Natürlich werden dadurch andere Ports belegt, so das unter Umständen z.B. das LCD-Display nicht mehr betrieben werden kann.
JP2	<b>Universelle PIN's</b> Hier können wahlweise drei 8 fache einpolige Stiftleisten oder 3 Buchsenleisten eingelötet werden. Man kann auch eine 2x8 und eine 1x8 polige Stiftleiste einlöten. Daraus würden sich dann 8 Servo kompatible Stecker ergeben. Es können also 8 handelsübliche Modellbauservos direkt angeschlossen werden. Mit einer geeigneten Software können sich diese dann natürlich auch steuern. In Verbindung mit JP2 können an das Board also 16 Servos direkt angeschlossen werden. Natürlich werden dadurch andere Ports belegt, so das unter Umständen z.B. das LCD-Display nicht mehr betrieben werden kann.
JP3	Deaktiviert die komplette Balkenanzeige DZ1 - Kann sinnvoll sein um Strom zu sparen.
JP4	Deaktiviert die ein Teil der Balkenanzeige DZ2 - Kann sinnvoll sein um Strom zu sparen.
JP5	Deaktiviert die ein Teil der Balkenanzeige DZ2 - Kann sinnvoll sein um Strom zu sparen.
JPQ	Wird kein Quarz genutzt, kann man mit diesen Jumpers die Portleitungen auf die Steckklemme Port B legen
Potential	Buchsenleiste die fünf mal GND und fünf mal +5V bereitstellt Hier können bei Experimenten einfach Drähte (0,5cm <sup>2</sup> ) eingesteckt werden
TASTER T1 bis T5	Stehen zur freien Verfügung Die Abfrage ist im Demoprogramm beschrieben

## **Bauteile Bestell- und Bestückungsliste für RN-MEGA8PLUS**

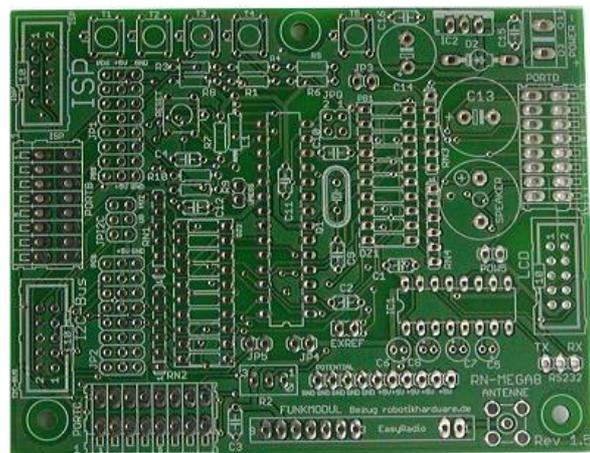
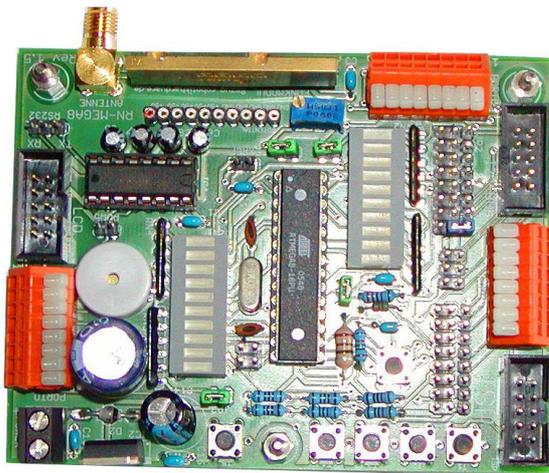
Die aktuelle Bestückungsliste aller Bauteile sowie Tipps für Bezugsquellen finden Sie bei uns auf der Projektseite <http://www.mikrocontroller-elektronik.de/>

**Platine kann man sich über die Eagle-Dateien selbst herstellen lassen  
Frei zum Download über die Seite <http://www.mikrocontroller-elektronik.de/>**

# Bestückungsplan



**ROT** sind die Jumper-Grundeinstellungen markiert



## Der erste Test

Nachdem Board aufgebaut ist, können wir daran gehen und das Board testen. Zunächst sollten noch einmal alle Jumper (Kurzschlussbrücken) überprüft werden. Stecken Sie dazu die Jumper entsprechend dem Bestückungsplan, auf der vorhergehenden Seite, ein.

Danach ein möglichst ein Netzteil mit 8 bis 13 V anschließen, da benötigt man noch keinen Kühlkörper. Ideal sind z.B. 9V oder 12V Netzgeräte. Günstige Steckernetzteile gibt es überall im Elektronikhandel!

Hat das Netzteil den falschen Stecker, einfach abschneiden und Drähte so in die Schraubklemmen einfügen. Auf Polung achten (auch wenn eine 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. 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 und Überhitzung 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.

Über den Programmieradapter wird nun das Board mit der Druckerschnittstelle ihres PC (oder bei neueren Programmern mit USB-Port) verbunden.. Dabei ist darauf zu achten das der ISP-Stecker auch in die richtige Wannenbuchse auf dem Board gesteckt wird. Also unbedingt auf die Beschriftung achten! Falsch herum kann man ihn nicht aufstecken.

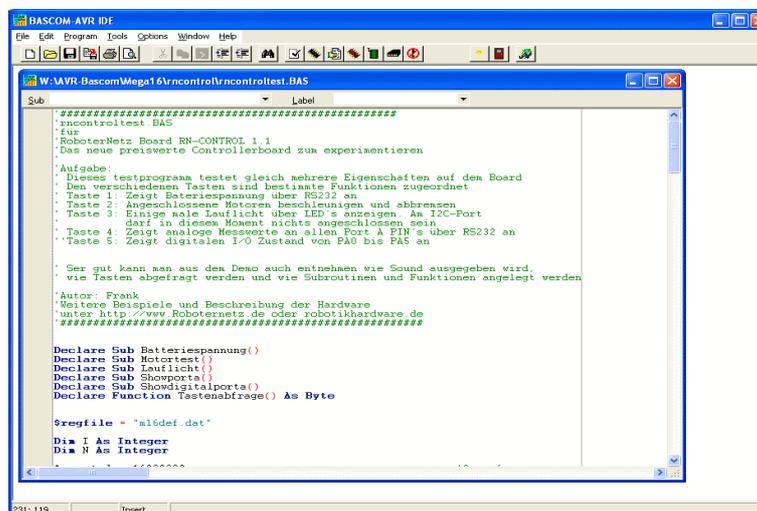
Ist man kein AVR-Profi, so empfehle ich für den Test des Board´s die Entwicklungsumgebung und Programmiersprache Bascom. Ein schneller Basic-Compiler mit hervorragender Benutzeroberfläche. Eine Version die nur in der Programmlänge beschränkt ist (max 4KB) gibt es beim Hersteller MCS Electronics kostenlos. Die genauen Links findet man am besten im Roboternetz-Bascom Forum

<http://www.roboternetz.de/community/forums/12-Atmel-AVR-Controller-und-Programmierung>

Eine Testversion ist auch über den Hersteller kostenlos erhältlich. Alternativ kann man das Board natürlich auch in C programmieren, zum Beispiel mit dem GCC Compiler oder AVR-Studio!  
Im Roboternetz gibt es ein Unterforum das sich nur mit Bascom beschäftigt. Hier findet man also immer Unterstützung.

Den Bascom Compiler kann man das ganze einfach über ein SETUP-Programm installieren. Anschließend legt man über File/New ein neue Datei an und gibt das nachfolgenden Beispielprogramme. ein. Alternativ kann man das ganze Beispielprogramm auch auf unserer Projektseite <http://www.mikrocontroller-elektronik.de/> herunterladen.

So sieht die Bascom Entwicklungsumgebung aus:



```
#####
rncntrltest.BAS
Für
  Roboternetz Board RN-CONTROL 1.1
  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 abbremsen
  Taste 3: Einige male Lauflicht über LED's anzeigen. Am I2C-Port
  dort 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

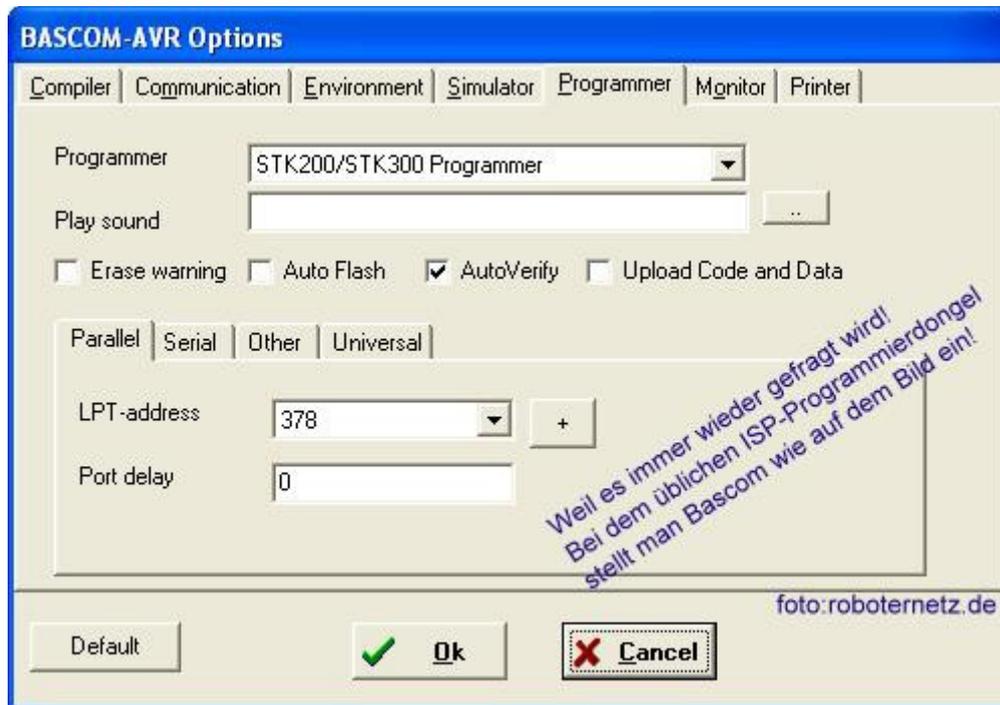
Autor: Frank
Weitere Beispiele und Beschreibung der Hardware
unter http://www.Roboternetz.de oder robotikhardware.de
#####

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
```

In der Bascom Entwicklungsumgebung sollte man nach dem erstmaligen Start über *Options/Compiler/Chip* den Controllertyp anwählen. Bei diesem Board müssten sie dort **M8** (für Mega 8) 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 Mega8 generell immer den internen 1 Mhz Takt nutzt. Aber das muss uns für den ersten Test nicht stören. Im Beispiel sollte dann jedoch statt `$crystal = 16000000` die Anweisung `$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.

```
#####
'rnmega8plustest.bas
'für
'RoboterNetz Board RN-MEGA8PLUS nur für Version 1.5 der Platine
'Das neue preiswerte Entwicklungsboard 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 und LCD an
' Taste 2: Gibt per Zufall einen Lottozahlentip ab - 6 aus 49
' Taste 3: Einige male Lauflicht über ein Balken-LED's
'           (2 Ports bleiben an, da dies rs232 Ports sind)
'           Bei einer LED kommt es zu leichtem Knacksen, da der Piepser diesen Port ebenfalls nutzt
' Taste 4: Ein einfaches Sirengeräusch
' Taste 5: Zeigt Copyriht auf LCD Display

' Jumper EXREF (Externe Referenzspannung) darf niucht gesteckt sein, da dieses
' Demo die Referenzspannung per Software auf 5V legt
' Bei veränderter Referenzspannung muss Tastenabfrage angepaßt werden

' Als LCD wird hier ein EADIP LCD mit Adapter RN-LCDAdapter genutzt
' es können natürlich auch andere angepasst werden

' Sehr 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/
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
#####
```

```
Declare Sub Batteriespannung()
Declare Sub Lauflicht()
Declare Function Tastenabfrage() As Byte
Declare Sub Textauflcdausgeben()
Declare Sub Lottozahlen()
Declare Sub Sirene()

$regfile = "m8def.dat"
$hwstack = 82
$framesize = 68
$swstack = 68

$crystal = 16000000           'Quarzfrequenz
'$crystal = 7372800         'Quarzfrequenz
$baud = 9600
Config Scl = Portc.5         'Ports fuer IIC-Bus
Config Sda = Portc.4

Config Adc = Single , Prescaler = Auto , Reference = Avcc
                        'Für Tastenabfrage und Spannungsmessung
Config Pinc.0 = Input       'Für Tastenabfrage
Portc.0 = 1                 'Pullup Widerstand ein

Start Adc
Const Ref = 5 / 1023        'Für Batteriespannungsberechnung
                        'Bei einer anderen Referenzspannung muss die 2.5 durch entsprechend ersetzt werden
                        'Die Referenzspannung wird durch den Jumper JPREF bzw. Spindeltrimmer festgelegt

Dim I As Integer
Dim N As Integer
Dim Ton As Integer
Dim Ws As Word              'Wird für Messung bei Tastenabfrage verwendet
Dim Adc_ohne_taste As Word

Dim Taste As Byte
Dim Volt As Single

' Falls ein LCD Display angeschlossen ist
Dim __lcdno As Bit
Config Lcdpin = Pin , E = Portc.3 , E2 = Portd.7 , Rs = Portc.2 , Db4 = Portd.2 , Db5 = Portd.3 ,
Db6 = Portd.4 , Db7 = Portd.5
Config Lcd = 20 * 4a , Chipset = Ks077
Config Lcdbus = 4
Config Pind.6 = Output      'RW=0
Portd.6 = 0
Config Pind.7 = Output
Portd.7 = 1                'LCD Licht ein

Initlcd
Cursor Off
Cls

Wait 1

I2cinit                    'Falls I2C benutzt wird

I = 0
Sound Portb.0 , 400 , 450   'BEEP
Sound Portb.0 , 400 , 250   'BEEP
Sound Portb.0 , 400 , 450   'BEEP

Print

Call Textauflcdausgeben

Adc_ohne_taste = Getadc(0)   'Messe Spannung wenn keine Taste gedrückt ist
Adc_ohne_taste = Adc_ohne_taste - 2 'für Toleranz abziehen

Do
    Taste = Tastenabfrage()
    If Taste <> 0 Then
```

```
Select Case Taste
  Case 1
    Call Batteriespannung           'Taste 1 Zeigt Bateriaespannung über RS232 an
  Case 2
    Call Lottozahlen               'Gibt Lottozahlen-Tip (ohne Gewähr) über rs232 und lcd
                                   'Bei Gewinn bitte mit Autor Frank teilen :-)
  Case 3
    Call Lauflicht                 'Einige male Lauflicht über Balken LED's anzeigen
  Case 4
    Call Sirene
  Case 5
    Call Textauflcdausgeben
End Select

    Sound Portb.0 , 400 , 500      'BEEP
End If
Waitms 100
Loop
End

'Diese Unterfunktion fragt die Tastatur am analogen Port ab
Function Tastenabfrage() As Byte
  Tastenabfrage = 0
  Ton = 600
  Ws = Getadc(0)
  ' Print "ws=" ; Ws 'Auskommentieren wenn man Tastenabfrage an andere Referenzspannung anpassen
  möchte
  If Ws < Adc_ohne_taste Then
    Select Case Ws
      Case 400 To 440
        Tastenabfrage = 1
        Ton = 550
      Case 330 To 370
        Tastenabfrage = 2
        Ton = 500
      Case 260 To 300
        Tastenabfrage = 3
        Ton = 450
      Case 180 To 230
        Tastenabfrage = 4
        Ton = 400
      Case 100 To 130
        Tastenabfrage = 5
        Ton = 350
      Case Else
        Print "Bitte Tastenabfrage an Messert ws=" ; Ws ; " anpassen!"
    End Select
    Sound Portb.0 , 400 , Ton      'BEEP
  End If
End Function

'Diese Unterfunktion zeigt Bateriaespannung an
Sub Batteriespannung()
Local W As Word
  Start Adc
  W = Getadc(1)
  Volt = W * Ref
  Volt = Volt * 5.3137             'Spannungsteiler (R9+R10)/R9
  Print "Die aktuelle Spannung beträgt: " ; Volt ; " Volt"
  Print "Rev: " ; Ref ; " W=" ; W

  Cls
  Locate 1 , 1
  Lcd "Spannung:" ; Fusing(volt , "#.##") ; " Volt"
End Sub

'Dies Funktion gibt auf einem angeschlossenen LCD-Display (2x27 Zeichen)
'eine Copyright Meldung aus
Sub Textauflcdausgeben()
  Cls
  Home
  Locate 1 , 1
  Lcd "Experimentierboard"
```

```
Locate 2 , 1
Lcd "RN-MEGA8PLUS V1.5"
Locate 3 , 1
Lcd "Roboternetz Standard"
Locate 4 , 1
Lcd "robotikhardware.de"

'Fuer RS232
Print "**** RN-MEGA8PLUS V 1.5 ****"
Print "Das neue Entwicklungsboard mit Funkmodul Steckplatz von www.robotikhardware.de"
End Sub

'Einige male Lauflicht über Balkenanzeige laufen lassen
Sub Lauflicht()

    Config Portd = Output
    Config Pinb.0 = Output
    Config Pinb.1 = Output

    Portd = 255
    Portb.0 = 1
    Portb.1 = 1
    For N = 1 To 10
        For I = 0 To 9
            Select Case I
                Case 0 To 7 :
                    Portd.i = 0
                    Waitms 150
                    Portd.i = 1
                Case 8:
                    Portb.0 = 0
                    Waitms 150
                    Portb.0 = 1
                Case 9:
                    Portb.1 = 0
                    Waitms 150
                    Portb.1 = 1
            End Select
        Next I
    Next N
    Config Portd = Input
    Config Pinb.0 = Input
    Config Pinb.1 = Input
    !Call 0                                     'Führt RESET aus
End Sub

Sub Lottozahlen()
    Print "Lottozahlen: ";
    Cls
    Home
    Lcd "Lottozahlentip-ohne Gewaehr):" "
    Locate 2 , 1
    For I = 1 To 6
        N = Rnd(48)
        N = N + 1
        Print " " ; N ;
        Lcd " " ; N;
    Next I
    Print
End Sub

Sub Sirene()
    For N = 1 To 5
        For I = 500 To 900
            Sound Portb.0 , 5 , I
        Next I
        For I = 900 To 500 Step -1
            Sound Portb.0 , 5 , I
        Next I
    Next N
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 Prüfverson von Bascom kann diese Version eventuell nicht compiliert werden, da diese etwas mehr als 4 KB Speicher benötigt. Zum compilieren müsste das Demo entweder gekürzt oder die Vollversion des Bascom Compilers bestellt werden. Auf unserer Projektseite <http://www.mikrocontroller-elektronik.de/> befindet sich jedoch auch eine bereits compilierte Version, diese lässt sich auch mit dem Demo übertragen.!

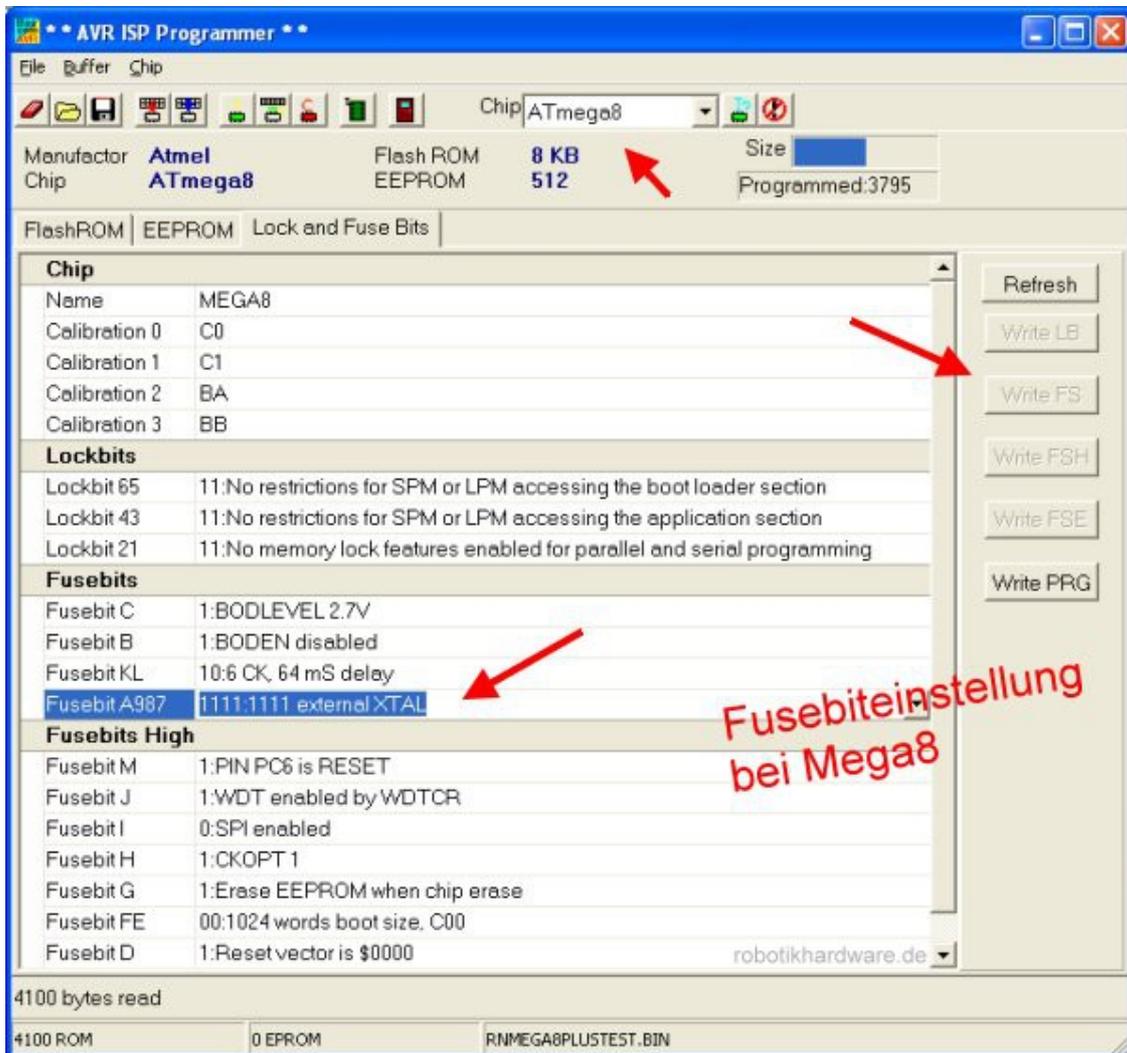
Haben sie alles richtig gemacht und oberes Beispiel auch korrekt eingegeben, dann können sie das Programm über das Symbol „Schwarze IC in der Toolbar“ compilieren. Es darf keine Fehlermeldung kommen. Erscheint eine Fehlermeldung dann klicken sie doppelt darauf, korrigieren sie die fehlerhafte Zeile und compilieren sie erneut. Ist das Programm fehlerfrei compiliert, 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 M8 für Mega8 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. Die ISP Übertragung ist manchmal etwas empfindlich gegenüber Störungen, so das diese eventuell wegen einem Fehler wiederholt werden muß. Dies ist aber kein Grund zur Besorgnis, das ist bei den AVR's leider so.

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.

## 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 Mega8 erfolgen damit die momentan eingestellten Werte erscheinen. Die Werte vom Fusebit A987 sollten so eingestellt werden, das sie der oberen Abbildung entsprechen. Haben sie die Werte verändert, dann klicken Sie auf **Write FS**.

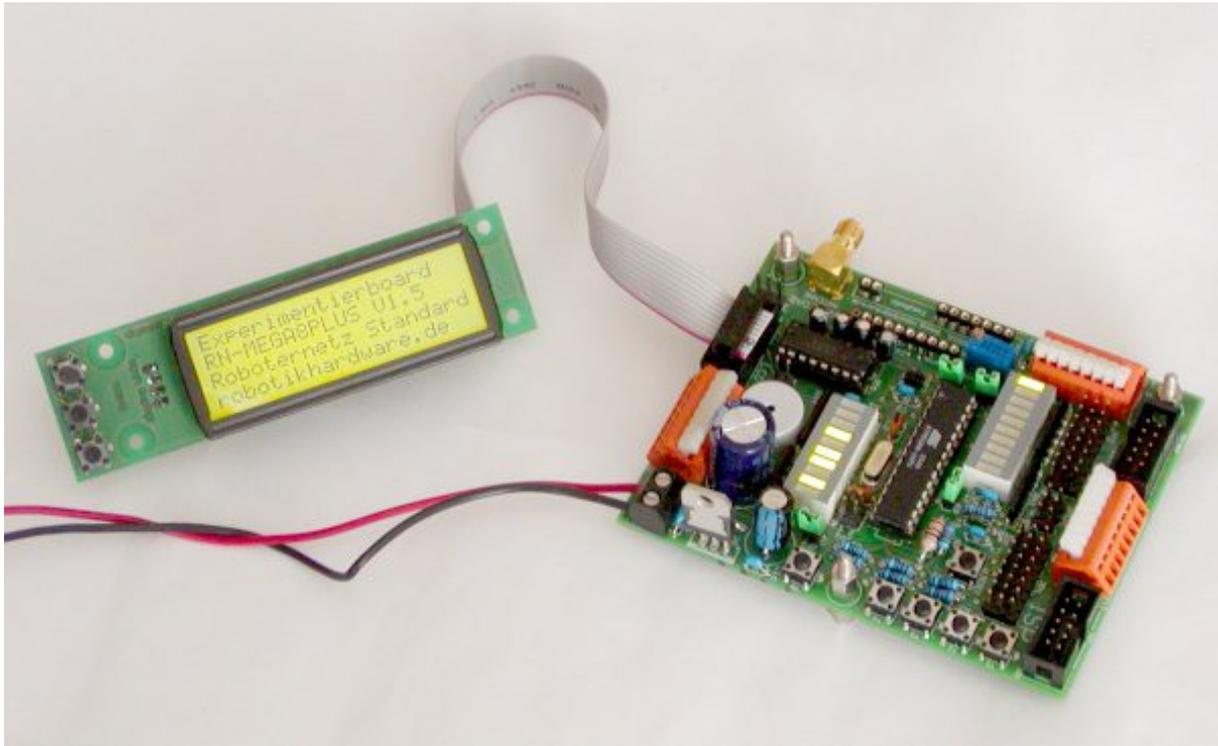
Dadurch sollte der Controller auf den externen Quarz umgeschaltet werden, die Taktfrequenz entspricht dann der Quarzfrequenz. Benötigt man sehr hohe Geschwindigkeiten bei der RS232 Schnittstelle, dann sind übrigens krumme Quarzwerte günstiger (siehe Roboternetz.de). Allerdings für die übliche Baudrate von 9600 Baud kann ohne Probleme auch der 16 Mhz Quarz genutzt werden.

Weitere Tipps hier: [http://www.roboternetz.de/wissen/index.php/RN-Board\\_FAQ-Seite](http://www.roboternetz.de/wissen/index.php/RN-Board_FAQ-Seite)

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!

Das Demoprogramm kann wahlweise mit LCD, Terminalprogramm oder beidem ausgetestet werden. Als LCD wurde hier das EADIP LCD mit RN-LCDAdapter über ein 10 poliges Kabel angeschlossen.



Nun können wir über die Board Tasten folgende Funktionen auslösen:

Taste 1	Zeigt Batteriespannung über RS232 und LCD an
Taste 2	Gibt per Zufall einen Lottozahlentip ab - 6 aus 49 über RS232 und LCD an
Taste 3	Einige male Lauflicht über ein Balken-LED's (2 Ports bleiben an, da dies rs232 Ports sind) Bei einer LED kommt es zu leichtem Knacksen, da der Piepser diesen Port ebenfalls nutzt
Taste 4	Ein einfaches Sirengeräusch
Taste 5	Zeigt den Hersteller und Produktnamen auf dem LCD Display an

# RS232 Kabel oder Funkverbindung

## RS232 Kabel

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 oder 5 oder auch RESET klickt, sollte im Terminalfenster was erscheinen. Hat man ein LCD-Display angeschlossen, dann erscheint dort ebenfalls etwas!

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.

## Funkmodul statt RS232 KABEL

Statt dem RS232 Kabel kann in RN-Mega8Plus auch einfach ein Funkmodul (EASYPAD) eingesteckt werden. Alle Daten werden dann statt über Kabel über Funk ausgetauscht. Wahlweise wieder mit PC oder einem anderen Board wie RN-Mega8Plus, RN-Mega128Funk, RN-Funk, RN-Steuerung usw. Sollen die Daten mit dem PC ausgetauscht werden, so sollte das Projekt RN-Funk als Gegenstelle am PC angeschlossen werden.

Die Software braucht in der Regel nicht oder nur gering geändert werden wenn man statt Kabel das Funkmodul nutzt. Wichtig ist nur das IC1 (MAX232) aus der Fassung genommen wird, wenn ein Funkmodul eingesetzt wird. Die Kabelschnittstelle wird durch das Funkmodul ersetzt, also steht dann nicht mehr bereit.

Um das Funkmodul zu konfigurieren kann beispielsweise folgendes Bascom Programm genutzt werden. Hier sieht man auch wie man das Funkmodul anspricht. Am Ende des Programms wird in einer Schleife dauerhaft gesendet, so das man die Funkverbindung gut testen kann.

```
'#####  
'rnmega8plusfunkmodulbefehl.bas  
'für  
'RoboterNetz Board RN-MEGA8PLUS nur für Version 1.5 der Platine  
'Das neue preiswerte Entwicklungsboard zum experimentieren  
'  
'Aufgabe:  
' Dieses Testprogramm testet das Funkmodul  
' Es werden einige Grundeinstellungen vorgenommen (Konfiguration)  
  
' Als LCD wird hier ein EADIP LCD mit Adapter RN-LCDAdapter genutzt  
' es können natürlich auch andere angepasst werden  
' Ohne LCD geht es auch, jedoch sieht man dann Diagnose nicht  
  
' Achtung: Bei eingestecktem Funkmodul muss IC1 (MAX232)  
'          aus Fassung entfernt werden  
  
'Autor: Frank  
'Weitere Beispiele und Beschreibung der Hardware  
'unter http://www.Roboternetz.de oder http://www.mikrocontroller-elektronik.de/  
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!  
'#####  
  
$regfile = "m8def.dat"  
$hwstack = 82  
$framesize = 68  
$swstack = 68  
  
$crystal = 16000000  
$crystal = 7372800  
$baud = 19200  
  
'Quarzfrequenz  
'Quarzfrequenz
```

```
Config Scl = Portc.5 'Ports fuer IIC-Bus
Config Sda = Portc.4

Declare Sub Rn_funkmodulversion();
Declare Function Rn_sendefunkbefehl(byval Funkbefehl As String) As Byte
Declare Function Rn_empfange_funkstring(dat As String) As Byte

Dim I As Word
Dim S As String * 90
Dim Rueck As Byte

' Falls ein LCD Display angeschlossen ist
Dim ___lcdno As Bit
Config Lcdpin = Pin , E = Portc.3 , E2 = Portd.7 , Rs = Portc.2 , Db4 = Portd.2 , Db5 =
Portd.3 , Db6 = Portd.4 , Db7 = Portd.5
Config Lcd = 20 * 4a , Chipset = Ks077
Config Lcdbus = 4
Config Pind.6 = Output 'RW=0
Portd.6 = 0
Config Pind.7 = Output
Portd.7 = 1 'LCD Licht ein

Initlcd
Cursor Off
Cls
Wait 1
Sound Portb.0 , 400 , 450 'BEEP
Sound Portb.0 , 400 , 250 'BEEP
Sound Portb.0 , 400 , 450 'BEEP

'RS232 Buffern
Config Serialin = Buffered , Size = 120
Config Serialout = Buffered , Size = 120
Enable Interrupts

Print
Print "**** RN-MEGA8PLUS V 1.5 ****"
Print "Das neue Entwicklungsboard mit Funkmodul Steckplatz von www.robotikhardware.de"
Lcd "RN-MEGA8PLUS V 1.5"
Locate 2 , 1
Lcd "Funkmodul testen"

I = 0

For I = 15 To 0 Step -1
    Locate 3 , 1
    Lcd "Countdown:" ; I ; " "
    Wait 1
Next I

Wait 1
Cls
Locate 3 , 1
Lcd "Firmware abrufen"
Rueck = Rn_sendefunkbefehl( "ER_CMD#T3")
If Rueck <> 0 Then
    Locate 4 , 1
    Lcd "ERROR"
End If
Rueck = Rn_empfange_funkstring(s)
Locate 4 , 1
Lcd "V " ; S

Wait 3

'
' Wait 1
' Print #2 , "Schaltet auf 9600 Baud"
' Rueck = Rn_sendefunkbefehl( "ER_CMD#U3")
' Baud = 9600

Wait 2
```

```
Cls
Locate 3 , 1
Lcd "19200 Baud einstellen"
Rueck = Rn_sendefunkbefehl( "ER_CMD#U4")
Baud = 19200

'
  Wait 1
'
  Print #2 , "Schaltet auf niedrigste Sendeleistung 1 mW"
'
  Rueck = Rn_sendefunkbefehl( "ER_CMD#P0")

  Wait 2
  Cls
  Locate 3 , 1
  Lcd "Sendeleistung 10mW"
  Rueck = Rn_sendefunkbefehl( "ER_CMD#P9")
  If Rueck <> 0 Then
    Locate 4 , 1
    Lcd "ERROR"
  End If

  Wait 2
  Cls
  Locate 3 , 1
  Lcd "Kanal 0 einstellen"
  Rueck = Rn_sendefunkbefehl( "ER_CMD#C0")
  If Rueck <> 0 Then
    Locate 4 , 1
    Lcd "ERROR"
  End If

  Wait 2
  Cls
  Locate 3 , 1
  Lcd "Sende in Schleife"
  Wait 3

  Do
    Locate 4 , 1
    Lcd "Sende HALLO:" ; I ; "    "
    Print "Hallo:" ; I
    I = I + 1
    Wait 1
  Loop

End

'Sendet EasyRadio Befehl und prüft ob korrekt empfangen
'Rückgabe: 0=Erfolgreich 1=Fehler
Function Rn_sendefunkbefehl(byval Funkbefehl As String) As Byte
Local Sech As String * 20
Local Z As Byte

Print Funkbefehl;
Waitms 20
Sech = ""
Do
  Z = Inkey()
  Sech = Sech + Chr(z)
Loop Until Z = 0

If Left(sech , 9) = Funkbefehl Then
  Print "ACK";
  Rn_sendefunkbefehl = 0
Else
  Rn_sendefunkbefehl = 1
End If
End Function
```

```
'Empfängt Datenstring (Holt alles aus dem Eingabebuffer)
'Rückgabe: 0=Erfolgreich 1=Fehler
Function Rn_empfange_funkstring(dat As String) As Byte
Local Z As Byte

    Waitms 20
    Dat = ""
    Do
        Z = Inkey()
        If Z <> 0 Then Dat = Dat + Chr(z)
    Loop Until Z = 0
    Rn_empfange_funkstring = 0
End Function
```

**Damit haben Sie den ersten Einstieg erfolgreich abgeschlossen.**

Wenn Sie das Demoprogramm 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 einiges an Literatur empfehlen. Sie finden passende Buchempfehlungen auf unserer Projektseite <http://www.mikrocontroller-elektronik.de/>

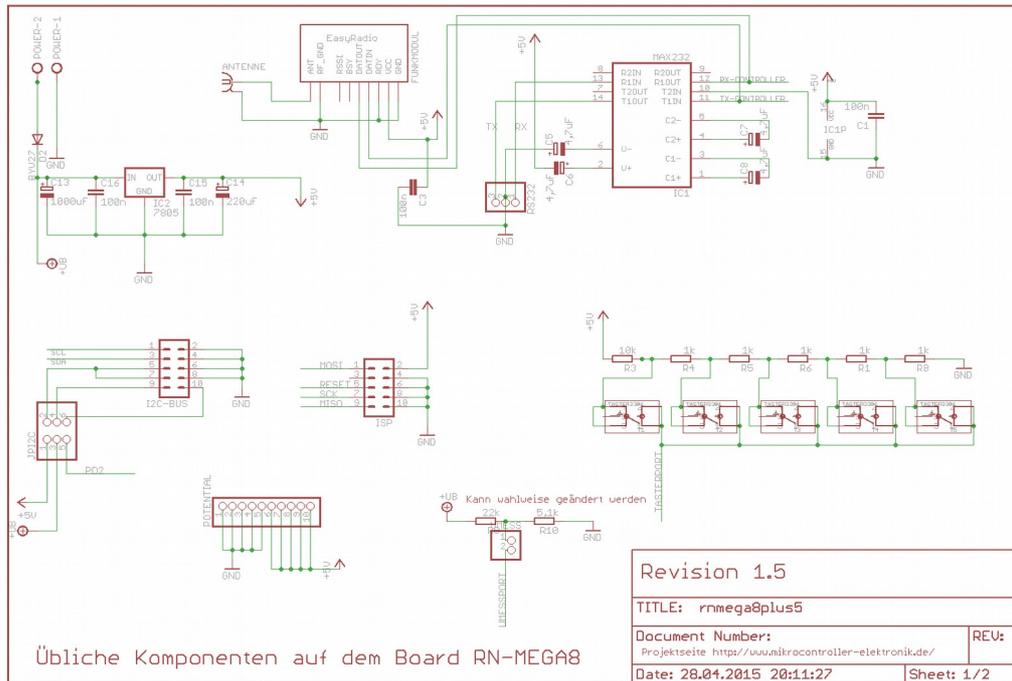
**Projekt-Seite zum herunterladen:** <http://www.mikrocontroller-elektronik.de/>

**Forum-Seite für Fragen:** <http://www.roboternetz.de/community/forum.php>

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

# Schaltplan Teil 1

Auf unserer Webseite <http://www.mikrocontroller-elektronik.de/> können Sie den Schaltplan im größeren Format herunterladen, wahlweise auch die Eagle Dateien.



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

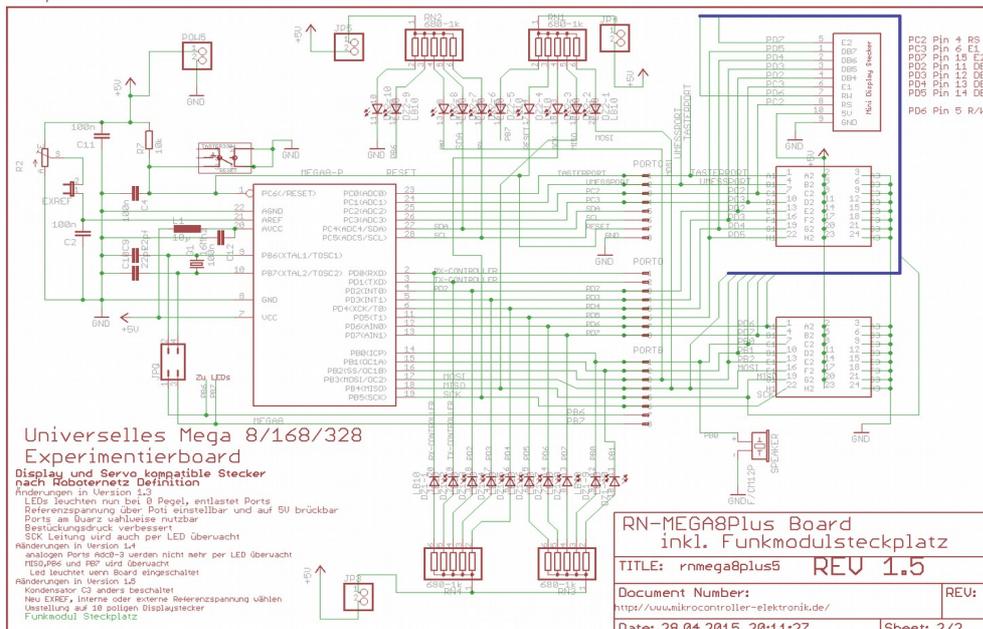
Forum <http://www.roboternetz.de>

Projekt (Schaltung & Projektdateien) 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.

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.

# Schaltplan Teil 2

<http://www.mikrocontroller-elektronik.de/>



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

Forum <http://www.roboternetz.de>

Projekt (Schaltung & Projektdateien) 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.

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.

Sollte in dieser Doku noch der ein oder andere Fehler drin stecken, so bitte ich um Nachsicht und Hinweise per Mail an den Entwickler. Also immer mal im Download Bereich auf der Projektseite nach der Versionsnummer der Doku schauen. Ergänzungen sind denkbar!



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](http://www.Roboternetz.de) & [www.Mikrocontroller-Elektronik.de](http://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!