

Leistungsstarkes AVR Microcontrollermodul neuester Generation
 100 Pin`s, mit USB / ISP / I2C u.v.m. für unzählige Aufgaben. Steckbar in eigene Boards
 auf Lochrasterplatinen.



RN-MEGA 2560 Modul V1.0

Dieses Board wurde für Anwender, Bastler und Entwickler konzipiert die bereits Erfahrungen mit AVR-Controllern haben oder denen die kleineren AVR's wie Mega8/Mega32 oder Mega128 noch nicht ganz ausreichen bzw. zu wenig Reserven bieten.

Dieses Modul nutzt den neusten Controllerchip ATmega2560. Er dürfte derzeit ohne Frage einer der leistungsfähigsten AVR Controller überhaupt sein. Er bietet zum Beispiel wesentlich mehr Funktionen, Speicher als der recht bekannte Mega128. Dennoch ist dieses Modul genauso einfach zu programmieren wie auch die kleineren Controller die zum Beispiel in RN-Control, RN-Mega128Funk, RN-Mega8 usw. genutzt werden. Ein Umstieg von einem der genannten Controllerboards oder AVR's ist demnach problemlos, Programme in Hochsprachen wie Basic müssen gar nicht bzw. nur kaum verändert werden.

Dieses Modul stellt nahezu alle 100 Pin`s dieses leistungsstarken Controllers über zwei Standardbuchsen (zwei mal 2x25 Buchsen=100 Pin) bereit. Der Kontaktabstand beträgt 2,54 mm, so das dieses Modul bequem in eigene Boards oder auf Lochrasterplatinen gesteckt werden kann, sogar Lochrasterplatinen mit 3er Reihen sind passend. Passende Stiftleisten als auch USB Kabel werden gewöhnlich mitgeliefert (siehe Shop Lieferumfang).

Das Modul beinhaltet bereits einen RS232 Pegelwandler als auch einen USB-Baustein. Es kann somit direkt an die RS232 oder an den USB-Anschluss eines PC's angeschlossen werden, wahlweise auch beides gleichzeitig.

Ein ISP-Programmierschlüssel ist bereits als 10 polige Standardbuchse (RN-Definition) auf dem Board. Eine Signal – Leuchtdiode und Stromsparsfunktion wurde ebenfalls integriert.

Trotz all dieser Features, ist dieses Modul mit 63x43mm unglaublich klein, leicht und flach. Erreicht wurde dies durch modernste SMD-Bauteile und moderne Serienfertigung. Das Modul wird bereits voll bestückt geliefert. Je nach Ausführung sind die Stiftleisten bereits integriert oder können je nach Vorhaben selbst eingelötet werden (siehe dazu im Shop / Lieferumfang).

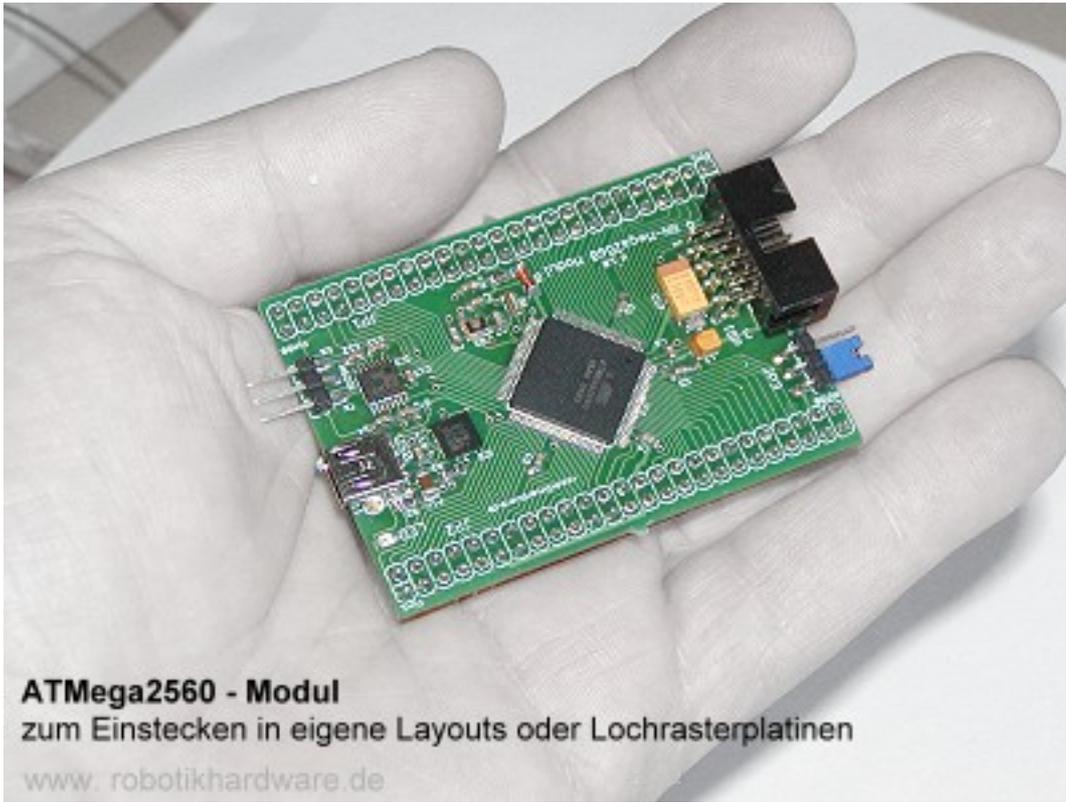


Wir denken die Leistungsdaten und Features werden jeden überzeugen!

Als hier die Leistungsmerkmale des Controllers und Moduls auf einen Blick:

- Leistungsstarker Controller **ATMega2560** (eine der wohl leistungsfähigsten AVR-Controller überhaupt), **100 Pin's am Gehäuse**,
- **256.000 Bytes Flash Speicher**
- 8000 Byte RAM
- 4000 Bytes EEPROM
- RAM erweiterbar auf 64K
- zwei 8 Bit Timer
- vier 16 Bit Timer
- vier 8 Bit PWM Kanäle
- **zwölf PWM Kanäle (programmierbar 2 bis 16 Bit)**
- 16 analoge Eingänge (Spannungsmessung)
- 86 programmierbare I/O Leitungen
- **vier TTL UARTS (RX/TX)** – 2 frei, einer für USB, einer für RS232)
- RS232 Anschluss (PC-Pegel) PC kann direkt über das übliche dreipolige RS232 Kabel angeschlossen werden.
- RS232 – Treiber auf dem Board schaltet automatisch in Stromsparmmodus wenn keine Daten übertragen werden – aktiviert sich auch wieder bei Datenübertragung. Manuell kann RS232 Treiber auch ganz deaktiviert werden.
- **USB-Anschluss** (Standard USB-Kabel mit MiniUSB-Stecker passt sofort). Treiber für Windows und Linux im Lieferumfang
- ISP-Programmierschluss – Standard Programmieradapter (wie unser ISP-Dongel) passen sofort.
- I2C-Bus über Stiftleiste herausgeführt (zum Anschluss zahlreicher Erweiterungen, Sensoren etc.)
- SPI-Bus über Stiftleiste herausgeführt
- 16 Mhz Quarz auf dem Board integriert
- Uhrenquarz ebenfalls auf dem Board integriert (für Echtzeituhr)
- eine programmierbare Leuchtdiode
- fast alle Portleitungen werden über die Stiftleisten herausgeführt
- zwei Stiftleisten mit jeweils 2x25 Pin und 2 passende Buchsenleisten im Lieferumfang. Dadurch kann das Board bequem in eigene Schaltungen oder auf Experimentierplatinen/Lochraster gesteckt werden.
- Betriebsspannung stabilisierte 5 V (**nur ganz geringe Stromaufnahme von wenigen mA**)
- Sehr kompakt durch modernste winzige Bauteile (nur ca. 43 x 63mm und nur 9mm hoch)
- Voll kompatibel zu den RN-Definitionen für Controllerboards
- Programmierbar in zahlreichen Sprachen, z.B. Basic (BASCOS Compiler, eingeschränkt bis 4K wird mitgeliefert), C (C-Compiler GCC wird mitgeliefert), Assembler, Pascal
- Deutsche Dokumentation mit Basic Programmbeispiel
- Fertig aufgebaut (lediglich Stiftleiste muss eingelötet werden)
- Bereits mit Testprogramm vorprogrammiert / alle Fusebitinstellungen schon vorgenommen
- Für Eagle –Anhänger wird das Modul als Library für Eagle auf CD mitgeliefert.
- Starter- oder Applikationsboard nicht unbedingt notwendig

Fotos



Übliche Lieferumfang (kann sich ändern, gültig sind stets die Angaben im Shop bei der Artikelbeschreibung)



RN-Mega2560 Modul

Hier mit stapelbarer Steckverbindung abgebildet



www.robotikhardware.de

Im oberen Bild wurde das Modul individuell mit einer anderen Steckverbindung ausgestattet. Diese erlaubt auch das Einstecken des Modules wie mit herkömmlichen Stiftleisten, jedoch können auch Erweiterungen (z.B. Ram Erweiterung, SD-Card Speicher usw.) oben aufgesteckt werden.



Anwendung

Aufbau und Start

Ein Aufbau ist bei diesem Modul nicht notwendig, da es bereits fertig bestückt geliefert wird. Je nach Ausführung sind auch die beiden Stiftleisten bereits eingelötet oder liegen manuell bei. Siehe dazu Artikelbeschreibung im Shop (www.robotikhardware.de).

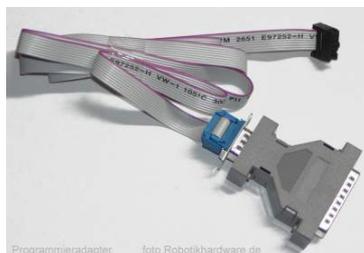
Liegen diese bei, müssen diese von ihnen eingelötet werden. Da es sich aber um Standard-Stiftleisten mit Kontaktabstand von 2,54mm handelt, dürfte dies auch von Einsteigern mit normalen Elektronik-LötKolben problemlos in wenigen Minuten erledigt sein.



Welche Art von Stiftleisten Sie einlöten, bleibt weitgehend Ihnen überlassen. Empfehlenswert ist es eine 25 Pin Stiftleiste, so wie auf dem Bild zu sehen, einzulöten. Auf einer Experimentierplatine kann dann als Gegenstück eine Buchsenleiste eingelötet werden.

Das Board kann danach über die Stiftleiste (oder ISP Wannenstecker) direkt mit 5V stabilisierter Gleichspannung versorgt werden. Achten Sie unbedingt darauf das die Spannung an den richtigen Pin gelegt wird (nachfolgende Pin-Beschreibung beachten). Ein Verpolen oder eine falsche Spannung zerstört das Modul (dies wäre kein Garantiefall und eine Reparatur wäre teuer).

Zum Programmieren kann der **Standard-ISP Programmer** für den Druckerport oder **ein BASCOM-USB-ISP Programmer** verwendet werden. Der neue **BASCOM-USB-ISP Programmer** hat den Vorteil das er schneller ist und zudem das Modul gleichzeitig mit 5V Spannung versorgen kann. Wenn Sie also nahezu immer mit Bascom Basic programmieren wollen, empfehlen wir ihnen den etwas teureren **BASCOM-USB-ISP Programmer** von www.robotikhardware.de zu erwerben.



Programmieradapter foto Robotikhardware.de

oder

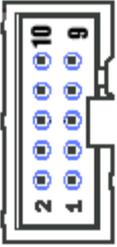


robotikhardware.de

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

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

Erläuterung der Anschlüsse, Regler und Kurzschlussbrücken

Anschluss-Bezeichnung	Erläuterung
ISP 	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 „ISP-Programmieradapter“ unter www.roboternetz.de) oder fertig bestellen (z.B. www.robotikhardware.de). 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
USB	USB-Anschluss für PC Über ein Standard USB-Kabel mit Mini-USB Stecker kann das Board mit dem PC verbunden werden. Bevor die Kabel eingesteckt werden, muss der mitgelieferte PC-Treiber installiert werden. Anschließend steht beim PC ein weiterer virtueller COM-Port zur Verfügung, bei dem ganz einfach Daten mit dem CONtrollerboard ausgetauscht werden können, ähnlich wie bei einer RS232 Verbindung. Jedes Terminalprogramm wäre dafür geeignet. Wird ein Bootloader installiert, so kann das Board auch über USB programmiert werden. Ein geeigneter Bootloader wird auf CD mitgeliefert!
RS232	PC kompatible RS232 Schnittstelle 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. Die Belegung ist kompatibel zum Roboternetz-Standard, als auch zum Conrad Roboter CCRP5: Pin 1 RX Pin 2 GND Pin 3 TX Ein geeignetes Anschlußkabel kann schnell selbst angefertigt werden oder gibt es über robotikhardware.de bereits fertig zu kaufen
JP3	RS232 aktivieren/deaktivieren Je nach Jumperstellung kann die RS232-Schnittstelle des Modules aktiviert oder deaktiviert werden. Perbindet man Pin 2 und 3 dieser Stiftleiste, ist er aktiviert. Pin 1 und 2 deaktiviert die Schnittstelle.
JP1	siehe nächsten Seiten
JP2	siehe nächsten Seiten

Diese Board hält sich an die Einheitlichen RN-Definitionen aus dem Roboternetz. Siehe dazu:

<http://www.roboternetz.de/wissen/index.php/RN-Definitionen>

Stiftleiste JP 1

Pin Nummer JP1	Funktion
Pin 1	PG5 / OC0B
Pin 2	PE0 / RXD0 / PCINT0 Wird für RS232 Treiber genutzt, falls dieser aktiviert ist.
Pin 3	PE1 / TXD0 Wird für RS232 Treiber genutzt, falls dieser aktiviert ist.
Pin 4	PE2 / XCK0 / AIN0
Pin 5	PE3 / OC3A / AIN1
Pin 6	PE4 / OC3B / INT4
Pin 7	PE5 / OC3C / INT5 Mit USB verbunden und darf nur gelesen werden. Ist High wenn USB-Bus Host angeschlossen. Kann per Software abgefragt werden.
Pin 8	PE6 / T3 / INT6
Pin 9	PE7 / CLK0 / ICP3 / INT7
Pin 10	VCC Spannungsversorgung +5V
Pin 11	GND
Pin 12	PH0 / RXD2 (freie RS232 /TTL Pegel)
Pin 13	PH1 / TXD2 (freie RS232 /TTL Pegel)
Pin 14	PH2 / XCK2
Pin 15	PH3 / OC4A
Pin 16	PH4 / OC4B
Pin 17	PH5 / OC4C
Pin 18	PH6 / OC2B
Pin 19	PB0 / SS / PCINT0 (SPI-Bus)
Pin 20	PB1 / SCK / PCINT1 (SPI-Bus und ISP)
Pin 21	PB2 / MOSI / PCINT2 (SPI-Bus und ISP)
Pin 22	PB3 / MISO /PCINT3 (SPI-Bus und ISP)
Pin 23	PB4 / OC2A / PCINT4
Pin 24	PB5 / OC1A / PCINT5
Pin 25	PB6 / OC1B / PCINT6
Pin 26	PB7 / OC0A / OC1C / PCINT7
Pin 27	PH7 / T4
Pin 28	PG3 / TOSC2 / Uhrenquarz bereits auf Modul
Pin 29	RESET (kurz mit GND verbinden um RESET auszulösen)
Pin 30	VCC Spannungsversorgung +5V
Pin 31	GND
Pin 32	PL0 / ICP4
Pin 33	PL1 / ICP5
Pin 34	PL2 / T5
Pin 35	PL3 / OC5A
Pin 36	PL4 / OC5B
Pin 37	PL5 / OC5C
Pin 38	PL6
Pin 39	PL7
Pin 40	PD0 / SCL /INT0 (I2C-Bus)
Pin 41	PD1 / SDA /INT1 (I2C-Bus)
Pin 42	PD2 / RXD1 /INT2 (freie RS232 /TTL Pegel)
Pin 43	PD3 / TXD1 /INT3 (freie RS232 /TTL Pegel)
Pin 44	PD4 / ICP1
Pin 45	PD5 / XCK1 (LED auf dem Modul, leuchtet bei Low)
Pin 46	PD6 / T1
Pin 47	PD7 / T0
Pin 48	GND
Pin 49	NC (nicht belegt)
Pin 50	NC (nicht belegt)

Stiftleiste JP 2

Pin Nummer JP2	Funktion
Pin 1 (51)	PG0 / WR
Pin 2 (52)	PG1 / RD
Pin 3 (53)	PC0 / A8
Pin 4 (54)	PC1 / A9
Pin 5 (55)	PC2 / A10
Pin 6 (56)	PC3 / A11
Pin 7 (57)	PC4 / A12
Pin 8 (58)	PC5 / A13
Pin 9 (59)	PC6 / A14
Pin 10 (60)	PC7 / A15
Pin 11 (61)	VCC Spannungsversorgung +5V
Pin 12 (62)	GND
Pin 13 (63)	PJ2 / XCK3 / PCINT11
Pin 14 (64)	PJ3 / PCINT12
Pin 15 (65)	PJ4 / PCINT13
Pin 16 (66)	PJ5 / PCINT14
Pin 17 (67)	PJ6 / PCINT15
Pin 18 (68)	PG2 / ALE
Pin 19 (69)	PA7 / AD7
Pin 20 (70)	PA6 / AD6
Pin 21 (71)	PA5 / AD5
Pin 22 (72)	PA4 / AD4
Pin 23 (73)	PA3 / AD3
Pin 24 (74)	PA2 / AD2
Pin 25 (75)	PA1 / AD1
Pin 26 (76)	PA0 / AD0
Pin 27 (77)	PJ7
Pin 28 (78)	VCC Spannungsversorgung +5V
Pin 29 (79)	GND
Pin 30 (80)	PK7 / ADC15 / PCINT23
Pin 31 (81)	PK6 / ADC14 / PCINT22
Pin 32 (82)	PK5 / ADC13 / PCINT21
Pin 33 (83)	PK4 / ADC12 / PCINT20
Pin 34 (84)	PK3 / ADC11 / PCINT19
Pin 35 (85)	PK2 / ADC10 / PCINT18
Pin 36 (86)	PK1 / ADC9 / PCINT17
Pin 37 (87)	PK0 / ADC8 / PCINT16
Pin 38 (88)	PF7 / ADC7 / TDI
Pin 39 (89)	PF6 / ADC6 / TDO
Pin 40 (90)	PF5 / ADC5 / TMS
Pin 41 (91)	PF4 / ADC4 / TCK
Pin 42 (92)	PF3 / ADC3
Pin 43 (93)	PF2 / ADC2
Pin 44 (94)	PF1 / ADC1
Pin 45 (95)	PF0 / ADC0
Pin 46 (96)	AREF (Referenzspannung für AD Wandlung)
Pin 47 (97)	GND
Pin 48 (98)	VCC Spannungsversorgung +5V
Pin 49 (99)	NC (nicht belegt)
Pin 50 (100)	NC (nicht belegt)

Die Pin-Nummern in Klammern sind die Pinbezeichnungen die verwendet werden wenn das Modul mit der Cad-Software Eagle in eigene Projekte eingebunden wird. Eine Bauteil Library liegt dem Modul auf CD bei.

USB-Schnittstelle nutzen

RN-Mega2560 verfügt insgesamt über vier RS232-Schnittstellen welche sogar gleichzeitig genutzt werden können. Zwei davon stehen mit TTL-Pegel (0 und 5V) und eine mit dem Standard PC Pegel (V24 +/-12V) bereit. Die vierte wird in der Regel für den USB-Treiber auf dem Board genutzt, kann jedoch ebenfalls freigeschaltet werden.

Über die USB-Schnittstelle kann somit das Board auch bequem an Rechner angeschlossen werden die keine frei RS232 Schnittstelle besitzen.

Die USB-Schnittstelle kann genutzt werden um Daten in beide Richtungen auszutauschen. Auch das Programmieren über die USB-Schnittstelle ohne ISP-Programmer ist möglich, wenn ein sogenannter "Bootloader" in den Controller geladen wird. Dazu benötigt man jedoch zumindest einmal ein herkömmliches ISP-Programmierkabel. Ein geeigneter Bootloader sollte auf der Robotikhardware CD zu finden sein.

Treiber installieren

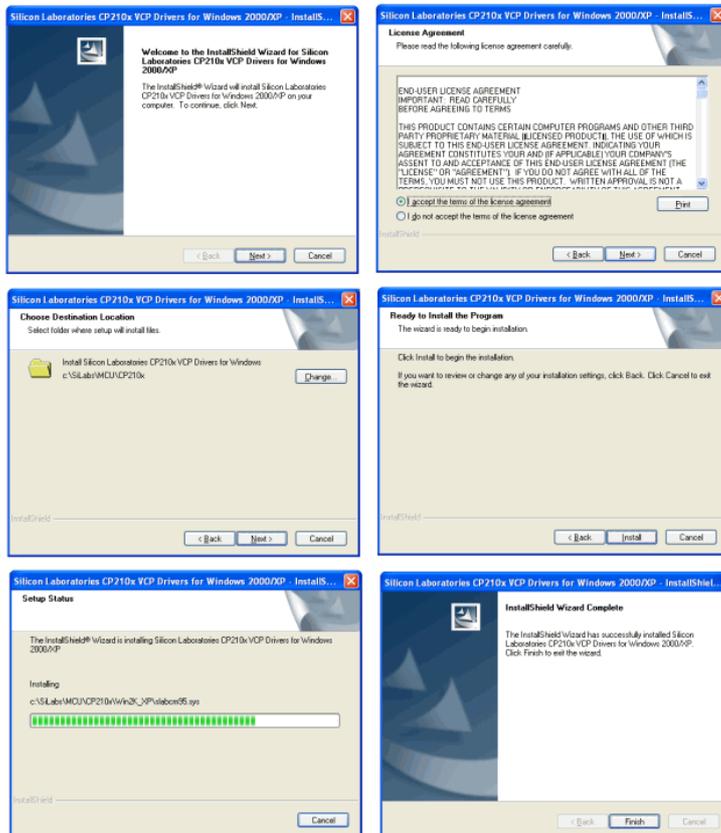
Um die USB-Schnittstelle überhaupt nutzen zu können, muss der PC das Board erkennen können. Um dies zu erreichen muss vor dem Anschluss ein Treiber unter Windows (oder anderes Betriebssystem) installiert werden. Der Ablauf ist kinderleicht, soll jedoch hier nochmals genau erläutert werden.

Befolgen Sie dazu folgende Schritte:

1. Das Board RN-Mega2560 mit Spannung versorgen aber **noch nicht** mit USB-Kabel verbinden!
2. Auf der Robotikhardware-CD finden Sie unter ihrem Betriebssystem (im Anleitungsverzeichnis) den geeigneten Treiber als ausführbare EXE-Datei. Für Windows XP und 2000 zum Beispiel das Programm

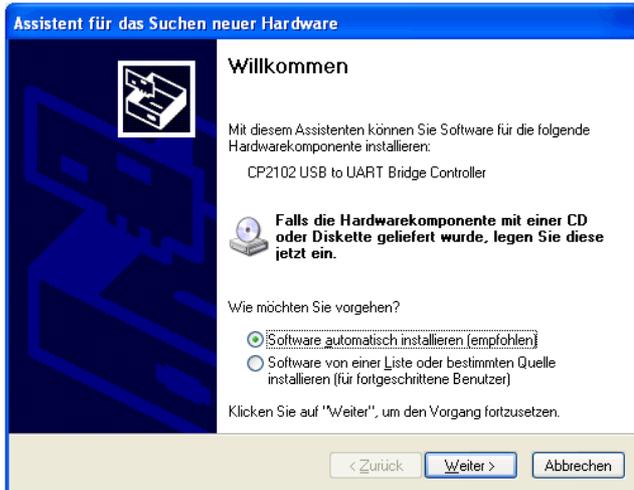
CP210x_VCP_Win2K_XP.exe

Dieses ist doppelt anzuklicken damit es ausgeführt wird. Anschließend erscheinen einige Dialoge die Sie alle unverändert mit Schaltfläche **NEXT** quittieren können. Das ganze dauert keine Minute!



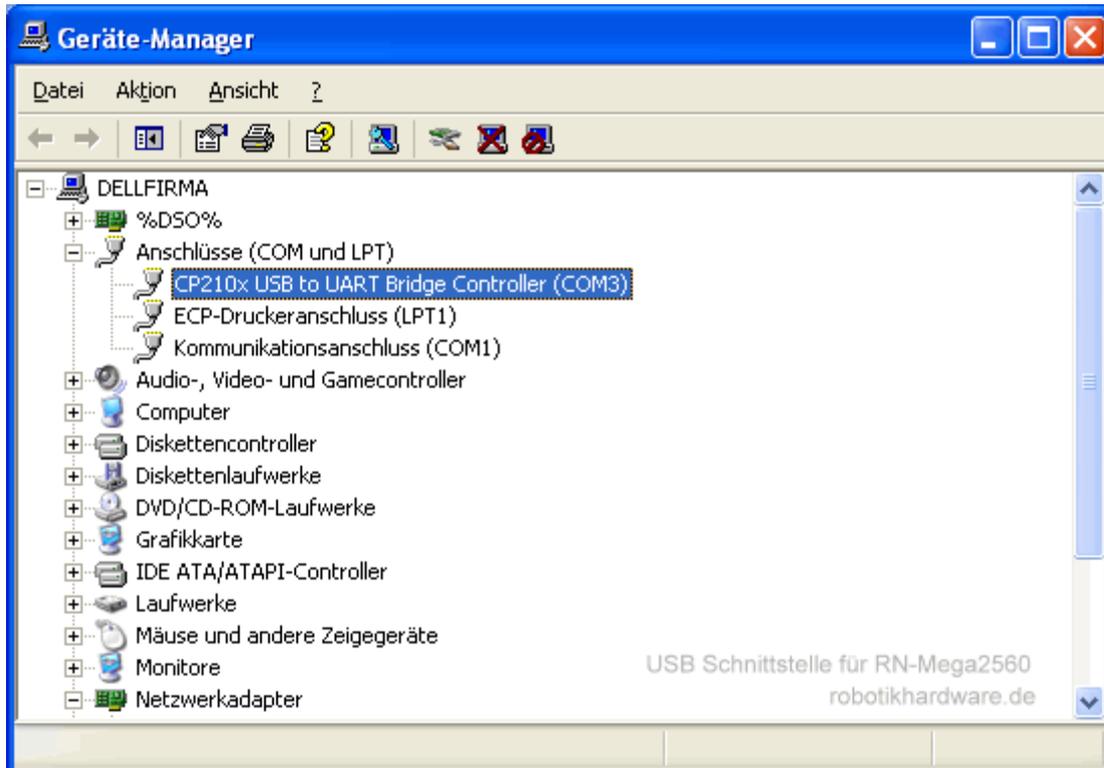
USB-Treiber für RN-Mega2560 installieren

- Nachdem der Treiber wie oben abgebildet installiert ist, verbinden Sie das Modul RN-Mega2560 über ein handelsübliches USB-Kabel (5pol Mini-USB-Stecker und USB A Standard Stecker) mit dem PC. Das USB Kabel ist in der Regel schon im Lieferumfang enthalten.
- Nun wird das Board erkannt und es erscheint ein Dialog. Lassen Sie die Einstellung auf "Treiber automatisch suchen" und klicken Sie auf weiter. Nun wird der Treiber automatisch korrekt eingerichtet, was Sie anhand mehrerer Meldungen erkennen.



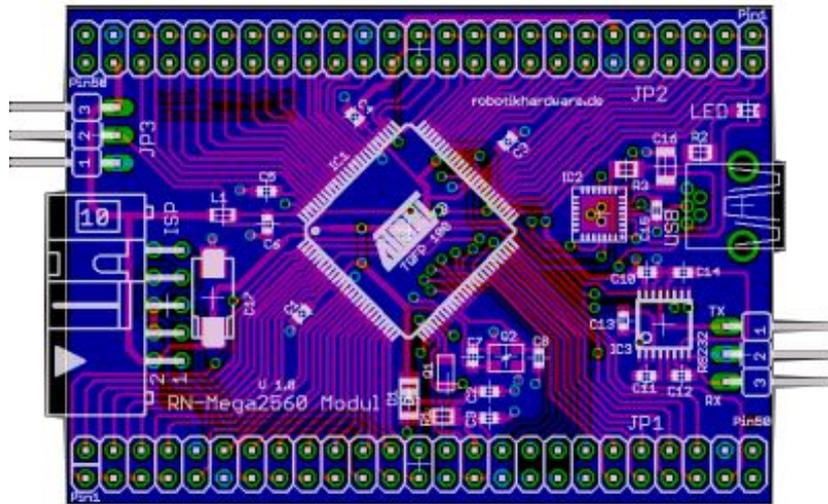
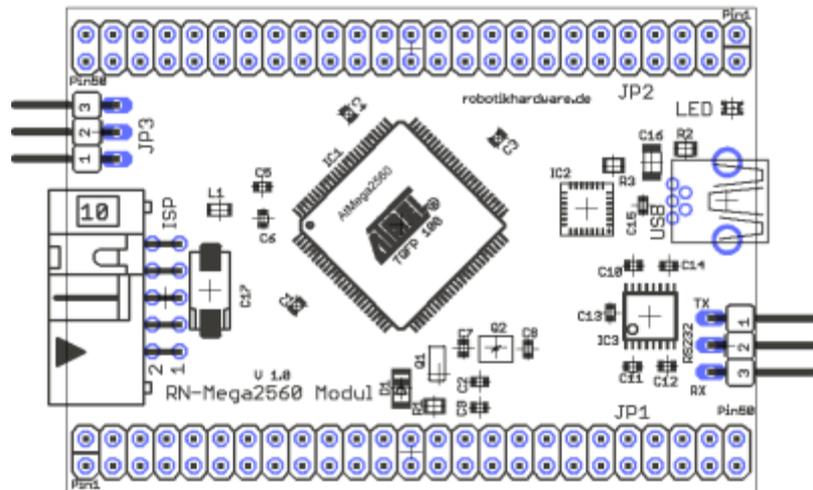
- Der USB-Treiber stellt ihnen automatisch einen weiteren COM-Port zur Verfügung. Je nach PC System kann die COM-Port Nummer etwas abweichen, in der Regel ist es jedoch COM-Port 3. Dieser COM-Port kann wie eine ganz normale RS232 Schnittstelle von jedem Programm (z.B. Terminalprogramm, Visual Basic usw.) angesprochen werden. Wenn Sie also ein Terminalprogramm starten und COM-Port 3 wählen (es geht auch das Bascom interne Terminalprogramm), dann werden Sie alle Daten lesen die über die USB-Schnittstelle eingehen. Alle Daten die Sie im Terminalprogramm eingeben werden per USB an das Modul RN-Mega2560 gesendet. Durch diese Verfahrensweise wird die USB-Schnittstelle besonders leicht verwendbar.

Wenn Sie nachschauen wollen welche COM-Port Belegung der Treiber wirklich angenommen hat, können Sie wahlweise auch den Gerätemanager aufrufen. Dort in der Liste sollten Sie bei angeschlossenem Board folgenden Eintrag finden:



Sobald Sie das USB-Kabel wieder herausziehen, verschwindet der Eintrag und die Schnittstelle wieder. Wird das Kabel wieder eingesteckt, erscheint der Treiber wieder neu. **Selbstverständlich braucht beim wiederholten Verbinden nichts mehr installiert werden!**

Bestückungsplan (Board ist bereits fertig bestückt)



RN-Mega2560 Modul

Modul mit leistungsstarkem 100 Pin AVR
und USB / RS232 / ISP passend
für alle 2,54 Raster / Platinen



Exklusiv bei www.robotikhardware.de

Board als Fertigbaustein über
www.robotikhardware.de

Der erste Test

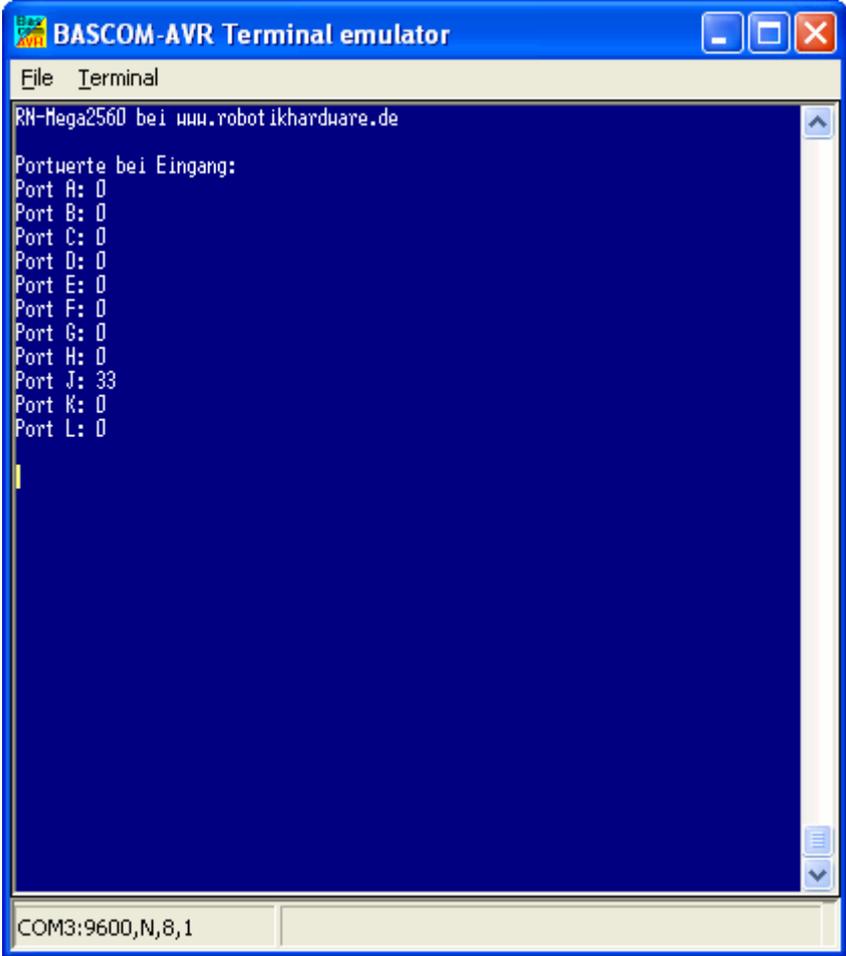
Das Testprogramm

Alle ausgelieferten Module RN-Mega2560 werden bereits vor der Auslieferung mit einem Testprogramm vorprogrammiert und ausgetestet. Auch die Fusebiteneinstellungen (Controller Konfiguration) die Anfängern oft Probleme macht, werden von uns bereits vorgenommen. Auf diese Weise ist für Sie der Einstieg und die Nutzung besonders leicht.

Wenn Sie das Modul erhalten, brauchen Sie das Board nur noch mit einer 5V Spannung zu versorgen und sofort wird das aufgespielte Testprogramm gestartet. 5V und GND können Sie an einen der Versorgungspins (zum Beispiel Pin 10 +5V und Pin 11 für GND) legen. Wenn Sie einen **BASOM USB ISP Programmer** haben, dann können Sie auch diesen für die Spannungsversorgung nutzen. Achten Sie bei diesen Programmer lediglich darauf das der Jumper am ISP (INT/EXT) gesteckt ist.

Sobald das Board mit Spannung versorgt wird, prüft das Testprogramm ob eine USB-Verbindung mit dem PC besteht. Ist diese korrekt vorhanden, dann leuchtet die LED auf RN-Mega2560 ca. 10 Sekunden dauerhaft auf. Wenn das USB-Kabel dagegen nicht korrekt verbunden ist oder der Treiber nicht korrekt installiert wurde, dann blinkt die LED ca. 10 Sekunden. Somit können Sie sehr gut die USB-Verbindung testen. Beachten Sie das die USB-Verbindung nur dann als korrekt angezeigt wird, wenn diese vor der Spannungsversorgung des RN-Mega2560 (RESET) vorhanden ist.

Nachdem die LED wieder ausgeht, wird eine Meldung mit 9600 Baud über USB vom Board RN-Mega2560 ausgegeben. Wenn Sie also ein Terminalprogramm mit richtiger Baudrate (9600 Baud / 8 Bit) und dem neuen COM Port (gewöhnlich COM 3) gestartet haben, dann werden Sie in etwa folgende Meldung nach dem Verlöschen der LED sehen:



```
BASCOM-AVR Terminal emulator
File Terminal
RN-Mega2560 bei www.robotikhardware.de
Portuerte bei Eingang:
Port A: 0
Port B: 0
Port C: 0
Port D: 0
Port E: 0
Port F: 0
Port G: 0
Port H: 0
Port J: 33
Port K: 0
Port L: 0
COM3:9600,N,8,1
```

Im oberen Bild wurde das in Bascom integrierte Terminalprogramm genutzt. Ausgegeben werden die Portzustände. In der Regel entsprechen die den oberen Werten. Sie können aber auch abweichen, da an die Ports ja noch keine Dinge (Sensoren/Aktoren) angeschlossen wurden. Da auch eine Pullup Widerstände per

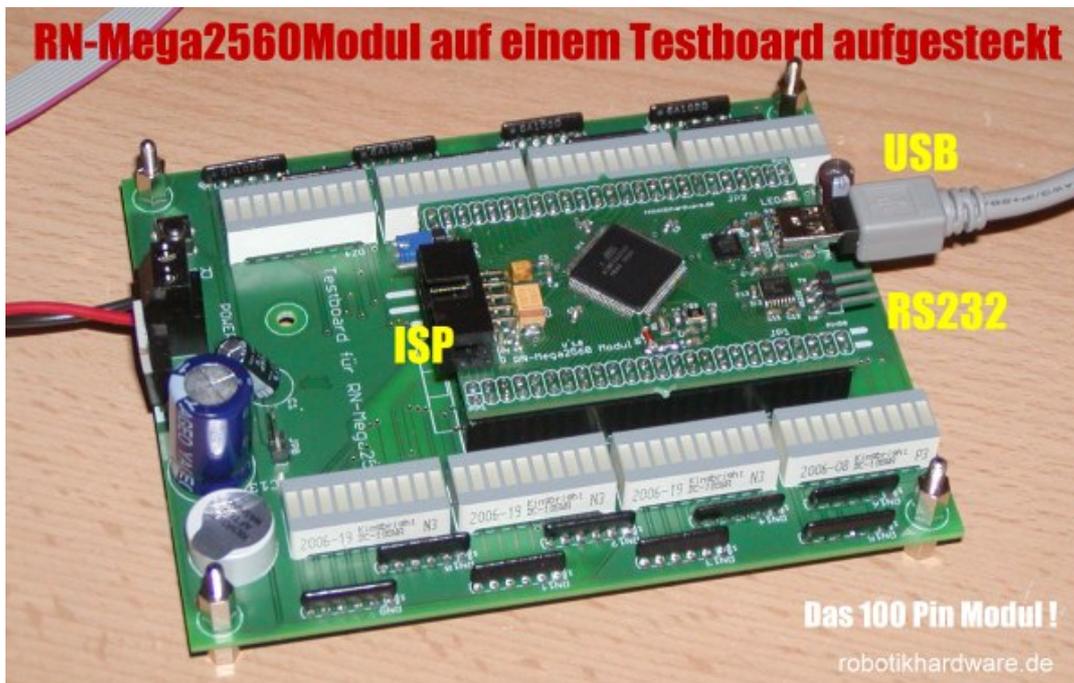
Software aktiviert wurden, ist der Zustand der Werte nicht 100% definiert. Insbesondere wenn man da sBoard kurz hintereinander mehrfach startet, können noch andere Werte in den Registern stehen. Die Ausgabe soll nur demonstrieren, da die USB-Verbindung perfekt funktioniert.

Nach dieser Ausgaben geht das Testprogramm in einen Porttest über. Dabei werden zunächst alle Ports auf Ausgang und High geschaltet. Danach werden 80 Ports geprüft indem 8 mal nacheinander immer 10 Ports auf LOW geschaltet werden.

Nach diesem Kurztest werden alle 80 Ports laufflichtartig kurz auf LOW geschaltet um die Funktion jedes einzelnen Ports zu testen.

Sie werden diesen Porttest kaum nachvollziehen können, da es recht aufwendig wäre so schnell hintereinander mit dem Multimeter alle Signale zu verfolgen. Dies können Sie sich auch ersparen, denn wir haben die Analyse des Tests schon für Sie vorgenommen.

Wir haben dafür ein Testboard, das über 80 Leuchtdioden verfügt. Ein aufgestecktes Board ergibt bei einem solchen Testboard ein schönes Laufflicht:



Ein solches Testboard werden Sie sicher nicht benötigen, dennoch haben wir für interessierte eine Eagle-Datei (CAD-Programm von CADSOFT) auf CD beigelegt. Sie können aus dem Beispiel ersehen wie leicht sich mit einer CAD-Software das Modul in eigene Anwendungen integrieren läßt.

Natürlich geht's auch ohne CAD-Software, denn außer 5V Spannungsreglung ist eigentlich alles wichtige bereits auf RN-Mega2560 vorhanden.

Für interessierte hier der Basic (Bascom Compiler: Version 1.11.8.3) Quellcode des Testprogrammes:

```
'#####
'mega2560modultest_2.bas Demo zu Modul RN-Mega2560
'für
'RoboterNetz Board RN-Mega2560 ab Version 1.0 und
'
'Aufgabe:
'Testet nach dem Start (RESET) zunächst ob USB
'bereit ist. Ist dies der Fall, dann muss LED
'ca. 10 Sekunden mit Dauerlicht leuchten.
'Ist USB nicht eingesteckt oder defekt, dann
'blinkt die LED nach dem RESET ca. 10 Sekunden
'Nach dem USB Test wird über USB noch eine Meldung
'mit Portwerten ausgegeben (9600 Baud).
'Anschließend werden alle Ports getestet indem
'Sie als Ausgang nacheinander in Blöcken zu 10
'Stück eingeschaltet werden. Also 8 Blöcke im Laufflicht
'hintereinander.
'Anschließend werden noch einmal alle LED's einzeln
'im Laufflicht durchgeschaltet.
```

```

'Autor: Frank
>Weitere Beispiele und Beschreibung der Hardware
>unter http://www.Roboternetz.de oder robotikhardware.de
>Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
#####

'Diese Anweisung setzt die FUsebits automatisch korrekt (Syntax $PROG LB, FB , FBH , FBX )
$prog , 255 , &B11011001 , 'Quarz an / Teiler aus / Jtag aus

$regfile = "m2560def.dat"
$hwstack = 82 '80
$framesize = 68 ' 64
$swstack = 68 '44

$crystal = 16000000 'Quarzfrequenz
'$baud = 19200

'Config Scl = Portd.0 'Ports fuer IIC-Bus
'Config Sda = Portd.1

Sound Portg.5 , 400 , 450 'BEEP
Sound Portg.5 , 400 , 250 'BEEP
Sound Portg.5 , 400 , 450 'BEEP

Config Pind.5 = Output
Led Alias Portd.5
Config Pine.5 = Input
Usb Alias Pine.5

Dim I As Integer
Dim U As Integer

Led = 1 : 'Aus
Waitms 200

If Usb = 1 Then 'Wenn USB erkannt dann
    Led = 0
    Wait 10
Else
    For I = 1 To 40
        Led = 1
        Sound Portg.5 , 400 , 250 'BEEP
        Waitms 100
        Led = 0
        Waitms 100
    Next I
End If

Config Com4 = 9600 , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol =
Open "com4:" For Binary As #4

Print #4 , "RN-Mega2560 bei www.robotikhardware.de"
Print #4 , ""
Print #4 , "Portwerte bei Eingang:"
Print #4 , "Port A: " ; Porta
Print #4 , "Port B: " ; Portb
Print #4 , "Port C: " ; Portc
Print #4 , "Port D: " ; Portd
Print #4 , "Port E: " ; Porte
Print #4 , "Port F: " ; Portf
Print #4 , "Port G: " ; Portg
Print #4 , "Port H: " ; Porth
Print #4 , "Port J: " ; Portj
Print #4 , "Port K: " ; Portk
Print #4 , "Port L: " ; Portl
Print #4 , ""

Config Porta = Output
Config Portb = Output
Config Portc = Output
Config Portd = Output
Config Porte = Output
Config Portf = Output
Config Portg = Output

```

```
Config Porth = Output
Config Portj = Output
Config Portk = Output
Config Portl = Output
```

```
Portg.0 = 1
Portg.1 = 1
Portc.0 = 1
Portc.1 = 1
Portc.2 = 1
Portc.3 = 1
Portc.4 = 1
Portc.5 = 1
Portc.6 = 1
Portc.7 = 1
```

```
Portj.2 = 1
Portj.3 = 1
Portj.4 = 1
Portj.5 = 1
Portj.6 = 1
Portg.2 = 1
Porta.7 = 1
Porta.6 = 1
Porta.5 = 1
Porta.4 = 1
```

```
Porta.3 = 1
Porta.2 = 1
Porta.1 = 1
Porta.0 = 1
Portj.7 = 1
Portk.7 = 1
Portk.6 = 1
Portk.5 = 1
Portk.4 = 1
Portk.3 = 1
```

```
Portk.2 = 1
Portk.1 = 1
Portk.0 = 1
Portf.7 = 1
Portf.6 = 1
Portf.5 = 1
Portf.3 = 1
Portf.4 = 1
Portf.2 = 1
Portf.1 = 1
Portf.0 = 1
```

```
Portd.7 = 1
Portd.6 = 1
Portd.5 = 1
Portd.4 = 1
Portd.3 = 1
Portd.2 = 1
Portd.1 = 1
Portd.0 = 1
Portl.7 = 1
Portl.6 = 1
```

```
Portl.5 = 1
Portl.4 = 1
Portl.3 = 1
Portl.2 = 1
Portl.1 = 1
Portl.0 = 1
Porth.7 = 1
Portb.7 = 1
Portb.6 = 1
Portb.5 = 1
```

```
Portb.4 = 1
Portb.3 = 1
Portb.2 = 1
Portb.1 = 1
Portb.0 = 1
Porth.6 = 1
Porth.5 = 1
Porth.4 = 1
Porth.3 = 1
Porth.2 = 1
```

```
Porth.1 = 1
Porth.0 = 1
Porte.7 = 1
Porte.6 = 1
Porte.4 = 1
Porte.3 = 1
Porte.2 = 1
Porte.1 = 1
Porte.0 = 1
```

Do

'Blöcke einzeln

```
Portg.0 = 0
Portg.1 = 0
Portc.0 = 0
Portc.1 = 0
Portc.2 = 0
Portc.3 = 0
Portc.4 = 0
Portc.5 = 0
Portc.6 = 0
Portc.7 = 0
Waitms 500
Portg.0 = 1
Portg.1 = 1
Portc.0 = 1
Portc.1 = 1
Portc.2 = 1
Portc.3 = 1
Portc.4 = 1
Portc.5 = 1
Portc.6 = 1
Portc.7 = 1
```

```
Portj.2 = 0
Portj.3 = 0
Portj.4 = 0
Portj.5 = 0
Portj.6 = 0
Portg.2 = 0
Porta.7 = 0
Porta.6 = 0
Porta.5 = 0
Porta.4 = 0
Waitms 500
Portj.2 = 1
Portj.3 = 1
Portj.4 = 1
Portj.5 = 1
Portj.6 = 1
Portg.2 = 1
Porta.7 = 1
Porta.6 = 1
Porta.5 = 1
Porta.4 = 1
```

```
Porta.3 = 0
Porta.2 = 0
Porta.1 = 0
Porta.0 = 0
Portj.7 = 0
Portk.7 = 0
```

```
Portk.6 = 0
Portk.5 = 0
Portk.4 = 0
Portk.3 = 0
Waitms 500
Porta.3 = 1
Porta.2 = 1
Porta.1 = 1
Porta.0 = 1
Portj.7 = 1
Portk.7 = 1
Portk.6 = 1
Portk.5 = 1
Portk.4 = 1
Portk.3 = 1
```

```
Portk.2 = 0
Portk.1 = 0
Portk.0 = 0
Portf.7 = 0
Portf.6 = 0
Portf.5 = 0
Portf.4 = 0
Portf.3 = 0
Portf.2 = 0
Portf.1 = 0
Waitms 500
Portk.2 = 1
Portk.1 = 1
Portk.0 = 1
Portf.7 = 1
Portf.6 = 1
Portf.5 = 1
Portf.4 = 1
Portf.3 = 1
Portf.2 = 1
Portf.1 = 1
```

```
Portd.7 = 0
Portd.6 = 0
Portd.5 = 0
Portd.4 = 0
Portd.3 = 0
Portd.2 = 0
Portd.1 = 0
Portd.0 = 0
Portl.7 = 0
Portl.6 = 0
Waitms 500
Portd.7 = 1
Portd.6 = 1
Portd.5 = 1
Portd.4 = 1
Portd.3 = 1
Portd.2 = 1
Portd.1 = 1
Portd.0 = 1
Portl.7 = 1
Portl.6 = 1
```

```
Portl.5 = 0
Portl.4 = 0
Portl.3 = 0
Portl.2 = 0
Portl.1 = 0
Portl.0 = 0
Porth.7 = 0
Portb.7 = 0
Portb.6 = 0
Portb.5 = 0
Waitms 500
Portl.5 = 1
Portl.4 = 1
Portl.3 = 1
Portl.2 = 1
```

```
Portl.1 = 1
Portl.0 = 1
Porth.7 = 1
Portb.7 = 1
Portb.6 = 1
Portb.5 = 1
```

```
Portb.4 = 0
Portb.3 = 0
Portb.2 = 0
Portb.1 = 0
Portb.0 = 0
Porth.6 = 0
Porth.5 = 0
Porth.4 = 0
Porth.3 = 0
Porth.2 = 0
Waitms 500
Portb.4 = 1
Portb.3 = 1
Portb.2 = 1
Portb.1 = 1
Portb.0 = 1
Porth.6 = 1
Porth.5 = 1
Porth.4 = 1
Porth.3 = 1
Porth.2 = 1
```

```
Porth.1 = 0
Porth.0 = 0
Porte.7 = 0
Porte.6 = 0
Porte.4 = 0
Porte.3 = 0
Porte.2 = 0
Porte.1 = 0
Porte.0 = 0
Portf.0 = 0
Waitms 500
Porth.1 = 1
Porth.0 = 1
Porte.7 = 1
Porte.6 = 1
Porte.4 = 1
Porte.3 = 1
Porte.2 = 1
Porte.1 = 1
Porte.0 = 1
Portf.0 = 1
```

```
Wait 1
```

```
'Alle Led's nacheinander einzeln
```

```
Portg.0 = 0 : Waitms 200 : Portg.0 = 1
Portg.1 = 0 : Waitms 200 : Portg.1 = 1
Portc.0 = 0 : Waitms 200 : Portc.0 = 1
Portc.1 = 0 : Waitms 200 : Portc.1 = 1
Portc.2 = 0 : Waitms 200 : Portc.2 = 1
Portc.3 = 0 : Waitms 200 : Portc.3 = 1
Portc.4 = 0 : Waitms 200 : Portc.4 = 1
Portc.5 = 0 : Waitms 200 : Portc.5 = 1
Portc.6 = 0 : Waitms 200 : Portc.6 = 1
Portc.7 = 0 : Waitms 200 : Portc.7 = 1
```

```
Portj.2 = 0 : Waitms 200 : Portj.2 = 1
Portj.3 = 0 : Waitms 200 : Portj.3 = 1
Portj.4 = 0 : Waitms 200 : Portj.4 = 1
Portj.5 = 0 : Waitms 200 : Portj.5 = 1
Portj.6 = 0 : Waitms 200 : Portj.6 = 1
Portg.2 = 0 : Waitms 200 : Portg.2 = 1
Porta.7 = 0 : Waitms 200 : Porta.7 = 1
Porta.6 = 0 : Waitms 200 : Porta.6 = 1
Porta.5 = 0 : Waitms 200 : Porta.5 = 1
```

```
Porta.4 = 0 : Waitms 200 : Porta.4 = 1
```

```
Porta.3 = 0 : Waitms 200 : Porta.3 = 1  
Porta.2 = 0 : Waitms 200 : Porta.2 = 1  
Porta.1 = 0 : Waitms 200 : Porta.1 = 1  
Porta.0 = 0 : Waitms 200 : Porta.0 = 1  
Portj.7 = 0 : Waitms 200 : Portj.7 = 1  
Portk.7 = 0 : Waitms 200 : Portk.7 = 1  
Portk.6 = 0 : Waitms 200 : Portk.6 = 1  
Portk.5 = 0 : Waitms 200 : Portk.5 = 1  
Portk.4 = 0 : Waitms 200 : Portk.4 = 1  
Portk.3 = 0 : Waitms 200 : Portk.3 = 1
```

```
Portk.2 = 0 : Waitms 200 : Portk.2 = 1  
Portk.1 = 0 : Waitms 200 : Portk.1 = 1  
Portk.0 = 0 : Waitms 200 : Portk.0 = 1  
Portf.7 = 0 : Waitms 200 : Portf.7 = 1  
Portf.6 = 0 : Waitms 200 : Portf.6 = 1  
Portf.5 = 0 : Waitms 200 : Portf.5 = 1  
Portf.4 = 0 : Waitms 200 : Portf.4 = 1  
Portf.3 = 0 : Waitms 200 : Portf.3 = 1  
Portf.2 = 0 : Waitms 200 : Portf.2 = 1  
Portf.1 = 0 : Waitms 200 : Portf.1 = 1
```

```
Portd.7 = 0 : Waitms 200 : Portd.7 = 1  
Portd.6 = 0 : Waitms 200 : Portd.6 = 1  
Portd.5 = 0 : Waitms 200 : Portd.5 = 1  
Portd.4 = 0 : Waitms 200 : Portd.4 = 1  
Portd.3 = 0 : Waitms 200 : Portd.3 = 1  
Portd.2 = 0 : Waitms 200 : Portd.2 = 1  
Portd.1 = 0 : Waitms 200 : Portd.1 = 1  
Portd.0 = 0 : Waitms 200 : Portd.0 = 1  
Portl.7 = 0 : Waitms 200 : Portl.7 = 1  
Portl.6 = 0 : Waitms 200 : Portl.6 = 1
```

```
Portl.5 = 0 : Waitms 200 : Portl.5 = 1  
Portl.4 = 0 : Waitms 200 : Portl.4 = 1  
Portl.3 = 0 : Waitms 200 : Portl.3 = 1  
Portl.2 = 0 : Waitms 200 : Portl.2 = 1  
Portl.1 = 0 : Waitms 200 : Portl.1 = 1  
Portl.0 = 0 : Waitms 200 : Portl.0 = 1  
Porth.7 = 0 : Waitms 200 : Porth.7 = 1  
Portb.7 = 0 : Waitms 200 : Portb.7 = 1  
Portb.6 = 0 : Waitms 200 : Portb.6 = 1  
Portb.5 = 0 : Waitms 200 : Portb.5 = 1
```

```
Portb.4 = 0 : Waitms 200 : Portb.4 = 1  
Portb.3 = 0 : Waitms 200 : Portb.3 = 1  
Portb.2 = 0 : Waitms 200 : Portb.2 = 1  
Portb.1 = 0 : Waitms 200 : Portb.1 = 1  
Portb.0 = 0 : Waitms 200 : Portb.0 = 1  
Porth.6 = 0 : Waitms 200 : Porth.6 = 1  
Porth.5 = 0 : Waitms 200 : Porth.5 = 1  
Porth.4 = 0 : Waitms 200 : Porth.4 = 1  
Porth.3 = 0 : Waitms 200 : Porth.3 = 1  
Porth.2 = 0 : Waitms 200 : Porth.2 = 1
```

```
Porth.1 = 0 : Waitms 200 : Porth.1 = 1  
Porth.0 = 0 : Waitms 200 : Porth.0 = 1  
Porte.7 = 0 : Waitms 200 : Porte.7 = 1  
Porte.6 = 0 : Waitms 200 : Porte.6 = 1  
Porte.4 = 0 : Waitms 200 : Porte.4 = 1  
Porte.3 = 0 : Waitms 200 : Porte.3 = 1  
Porte.2 = 0 : Waitms 200 : Porte.2 = 1  
Porte.1 = 0 : Waitms 200 : Porte.1 = 1  
Porte.0 = 0 : Waitms 200 : Porte.0 = 1  
Portf.0 = 0 : Waitms 200 : Portf.0 = 1
```

```
Loop  
End
```

Wie programmiert man RN-Mega2560

Wir haben es schon angesprochen, RN-Mega2560 kann im Prinzip in zahlreichen Sprachen von Assembler über C (GCC) bis hin zu Basic programmiert werden. Für C gibt es den kostenlosen GCC Compiler, für Assembler das AVR-Studio und für Basic den Bascom Compiler. Die Software ist auch auf CD enthalten, sie sollten jedoch stets schauen ob nicht schon ein Update auf den betreffenden Webseiten erhältlich ist und gegebenenfalls diese dann direkt von den Entwicklerseiten laden. Die Webseiten werden Webseiten auf unserer CD alle aufgelistet, siehe dazu auch unter www.rn-wissen.de

Empfehlenswert ist insbesondere auch der Bascom Basic Compiler in der neusten Version. Diese ist ebenfalls auf CD enthalten. Bascom basic ist eine unglaublich vielseitige Programmiersprache mit einem sehr großen Befehlsvorrat. Dadurch lassen sich auch komplexe Aufgaben und Vorhaben in sehr kurzer Zeit programmieren. Da der Code in echten Maschinencode kompiliert sind die Ausführungszeiten optimal für fast alle Aufgaben. Bascom ist jedoch mehr als nur ein Compiler, es ist eine komfortable Entwicklungsumgebung die alles enthält was man bei der Entwicklung benötigt, z.B. eingebauten Debugger, Terminalprogramm und Programmierer (Übertragungsprogramm).

Leider ist die kostenlose Version von Bascom nur für Programme bis 4000 Bytes verwendbar. Für größere Programme müssten Sie auf die Vollversion umsteigen, welche mit ca. 89 Euro bei robotikhardware.de aber noch recht preiswert ist. Bascom ist in der Lage nahezu alle AVR-Controller zu programmieren. Ein großer Vorteil von Bascom ist auch, dass es einige deutsche Bücher zu dem Compiler gibt, gerade Einsteiger werden die zu schätzen wissen.

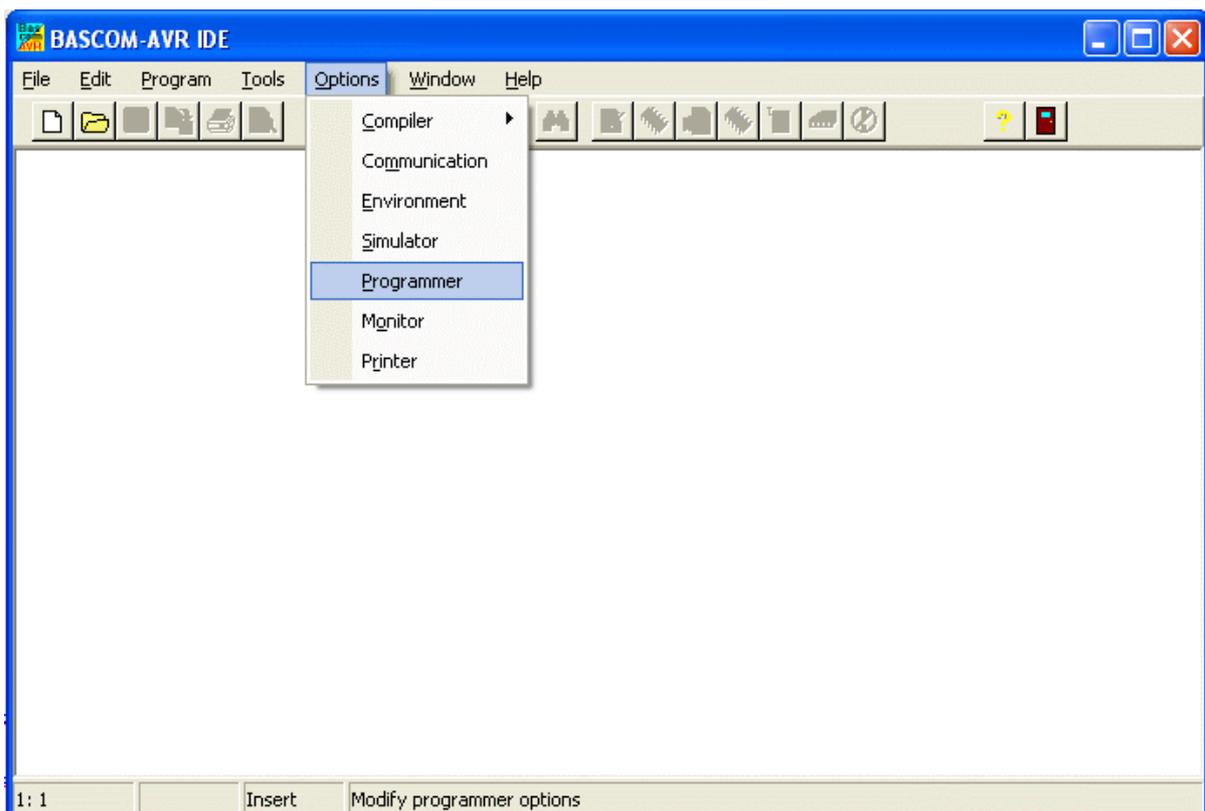
Nähere Infos / Einführungen zu Bascom auch hier: <http://www.roboternetz.de/wissen/index.php/Bascom>

Das deutsche Bascom Forum ist hier: <http://www.roboternetz.de/wissen/index.php/Bascom>

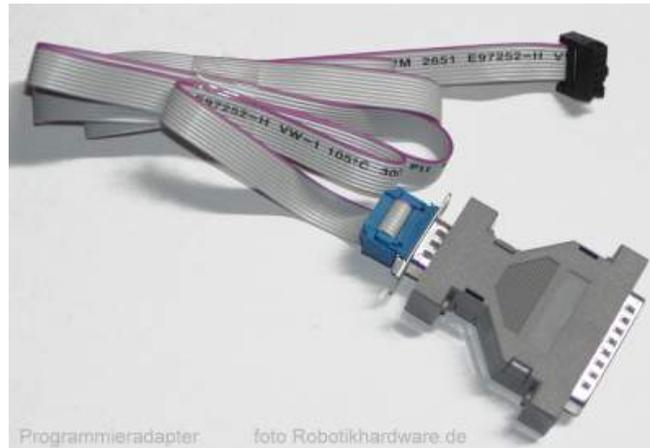
Programmieren mit Bascom

Für den ersten Einstieg empfehle ich die kostenlose mitgelieferte Testversion von Bascom zu nutzen. Mit 4K Codeerzeugung lässt sich schon einiges machen. Diese Testversion finden Sie auf der Robotikhardware CD im Verzeichnis Bascom oder laden Sie einfach direkt über eine Webseite <http://www.mcselec.com/> herunter.

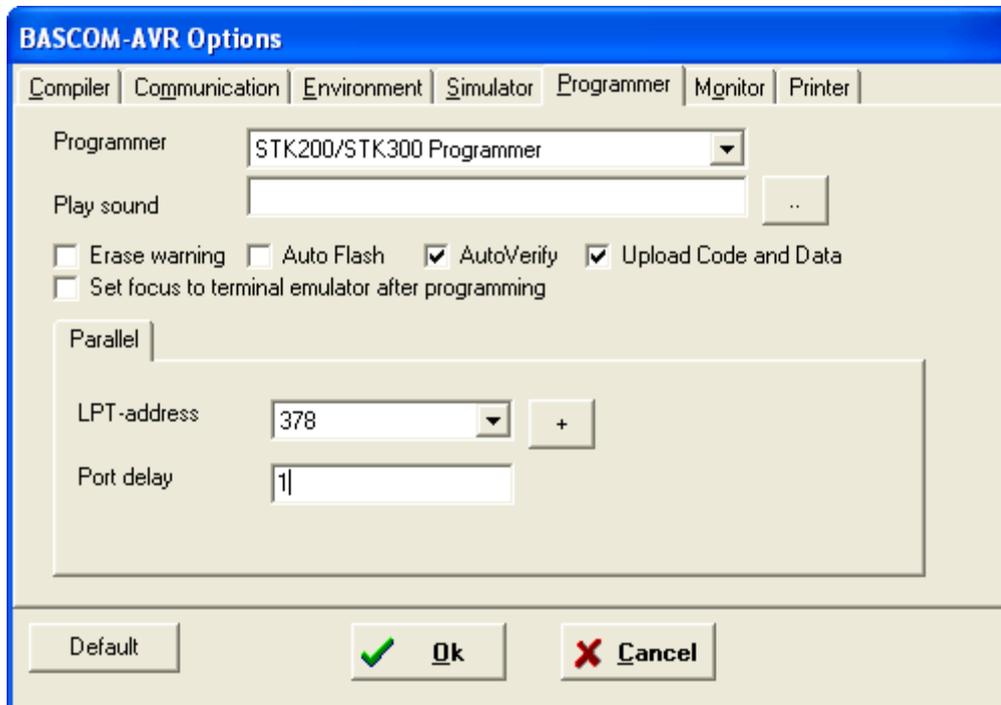
Bascom kann man ganz einfach mit SETUP installieren. Als erstes sollte man danach über Menüpunkt Optionen/Programmer festlegen, mit welchem Programmer (Programmierkabel) Programme vom PC auf das Modul RN-Mega2560 übertragen werden sollen.



Es gibt wie schon erwähnt eine ganze Reihe verschiedener "Programmer" auf dem Markt. Empfehlenswert ist wahlweise ein Standard-ISP-Programmer der an den Druckerport angeschlossen wird. Diesen gibt es bereits sehr günstig über robotikhardware.de.



Wird ein solches Kabel gewählt, dann sollten die Bascom Einstellungen wie im nachfolgenden Bild vorgenommen werden:



Dieser parallele Programmer ist der am häufigsten verwendete Programmer bei AVR Controllern. Vorallem wegen des günstigen Preises (unter 14 Euro) und der Unterstützung aller AVR Controller. Ein integrierter Treiberbaustein im Programmer schützt zudem ihren PC recht gut vor Störungen oder Fehlern beim basteln.

Anzumerken ist, das dieser Standard ISP-Programmer selbst keine 5V ausgeben kann. Er wird über das zu programmierende Board mit 5V versorgt. Das bedeutet er kann nur Boards programmieren, die bereits mit externer 5V Spannung versorgt werden.

Sollte es Übertragungsprobleme geben, empfehlen wir folgende Webseite:
http://www.roboternetz.de/wissen/index.php/RN-Board_FAQ-Seite

Alternative der Bascom USB Programmer

Da bei vielen Anwendern kein Druckerport mehr frei ist oder sogar gar keiner vorhanden ist, werden seit neustem die USB-Programmer immer beliebter. Leider hat sich hier kein solcher klarer Standard entwickelt wie bei dem parallelen Programmieren. Daher werden auf dem Markt unzählige verschiedene Varianten im Preisbereich zwischen 25 und 150 Euro angeboten.

Wir haben viele getestet und waren mit vielen nicht sonderlich zufrieden weil die meisten nur einen Teil der Controllertypen programmieren kann. Zudem mußten für neue Controller wieder Firmware-Update besorgt und eingespielt werden. Auch unterstützen nicht alle Bascom-Basic oder nicht alle Features von Bascom Basic.

Empfehlen können wir daher derzeit nur den neuen "BASCUM USB-ISP" Programmer. Er wird jetzt auch über robotikhardware.de angeboten. Er ist zwar nicht unbedingt der billigste, jedoch kann er alle Controller programmieren die auch Bascom programmieren kann. Kommt ein neuer Controller auf den Markt so muss man lediglich den Compiler updaten, ein umständliches Einspielen von neuer Programmer-Firmware ist bei diesem nicht notwendig. Da er speziell für Bascom entwickelt wurde, wird er direkt in der IDE unterstützt, alle Features wie z.B. die automatische Fusebitprogrammierung per Befehl, kann dieser Programmer ausführen.

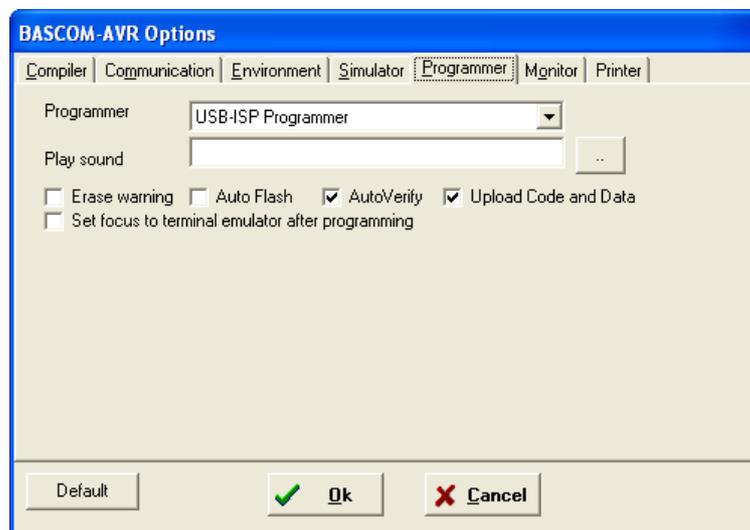
Weitere Vorteile sind:

- der Programmer ist sehr flott, große Programme werden wesentlich schneller übertragen als mit dem parallelen Standard Programmer.
- Der Programmer benötigt keine 5V. Im Gegenteil, durch einstecken eines Jumpers kann er AVR-Boards mit 5V (bis zu 40mA) versorgen. Dieser Umstand ist gerade bei RN-Mega2560 eine feine Sache.

Wer ausschließlich mit Bascom programmieren will, der ist also mit diesem neuen Programmer bestens versorgt:



Sollten Sie sich für diesen Programmer entschieden haben, so müssen die Bascom Optionen wie auf nachfolgendem Bild eingestellt werden:



Bitte vergessen Sie nicht, dass man für den USB-Programmer zunächst einen Treiber installieren muss. Bei neusten Bascom Versionen befindet sich dieser schon in einem Unterverzeichnis von Bascom. Er wird auf der Robotikhardware.de CD jedoch nochmals extra mitgeliefert.

Wir steigen ein

In dem Kapitel wollen wir noch einmal von vorne aufzeigen wie der Ablauf beim programmieren ist. Ich gehe davon aus das Sie Bascom installiert und den Programmer korrekt eingestellt und angeschlossen haben.

Zunächst ruft man die Bascom Entwicklungsumgebung auf. Anschließend legt man über File/New ein neue Datei an und gibt das nachfolgenden Beispielprogramme. ein. Alternativ kann man das ganze Beispielprogramm auch im Download-Bereich des Roboternetz herunterladen oder über die beiliegende CD laden.

Geben Sie also folgendes Beispielprogramm ein:

```
'#####
'erstebeispiel.bas Demo zu Modul RN-Mega2560
'für
'RoboterNetz Board RN-Mega2560 ab Version 1.0 und
'
'Aufgabe:
'Laesst LED blinken
'
'Autor: Frank
'Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de oder robotikhardware.de
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
'#####

'Diese Anweisung setzt die Fusebits automatisch korrekt
'(Syntax $PROG LB, FB , FBH , FBX )
$prog , 255 , &B11011001 , 'Quarz an / Teiler aus / Jtag aus

$regfile = "m2560def.dat"
$hwstack = 82
$framesize = 68
$swstack = 68

$crystal = 16000000 'Quarzfrequenz

Config Pind.5 = Output
Led Alias Portd.5

Do
  Led = 1
  Waitms 100
  Led = 0
  Waitms 100
Loop

End
```

Dies ist ein recht einfaches Programm. Die Aufgabe besteht nur darin die Leuchtdiode auf dem Board RN-Mega2560 ständig blinken zu lassen. Der eigentlich dafür zuständige Code ist nur dieser

```
Do
  Led = 1
  Waitms 100
  Led = 0
  Waitms 100
Loop
```

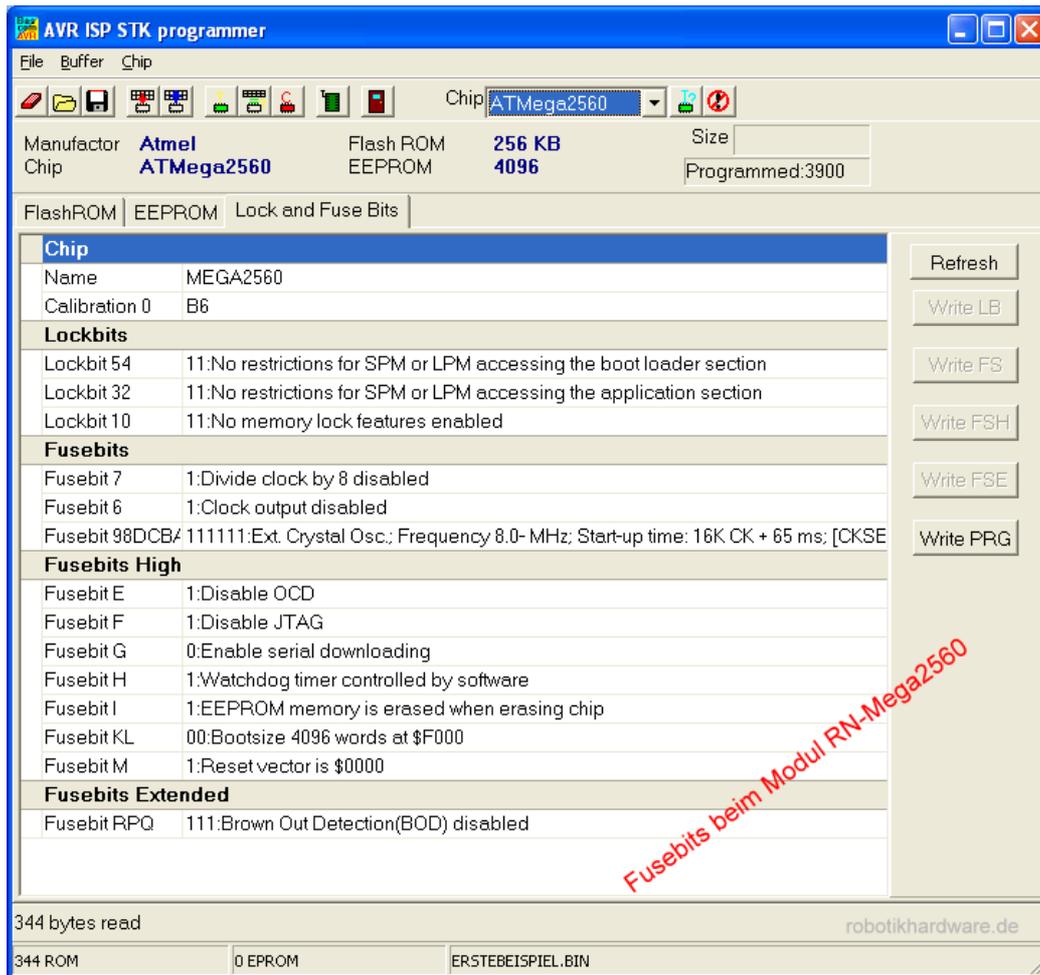
Die anderen Zeilen sind Kommentare oder grundlegende Anweisungen die man gewöhnlich bei allen Programmen für das Modul immer wieder gleich übernehmen kann. Die Anweisungen für den Stack sind natürlich hier im Beispiel übertrieben hoch angesetzt, da das Modul aber genügt Speicher besitzt wollten wir nicht geizen .

Sehr hilfreich ist die Anweisung ganz oben:

```
$prog , 255 , &B11011001 ,
```

Diese Anweisung sorgt dafür das die Fusebits automatisch beim übertragen des Programmes richtig gesetzt werden. "Fusebits" nennt man bestimmte Konfigurationseinstellungen beim Controller. So müssen über rdiese zum Beispiel festgelegt werden ob der Controller mit einem internen Takt oder über den Quarz-Takt betrieben werden soll. Die obere Einstellung sorgt dafür das der Quarz aktiviert und das Modul mit vollen 16 Mhz getaktet wird. Zudem wird die JTAG-Schnittstelle deaktiviert, da diese oft nicht gebraucht wird und sonst unnötig Ports belegen würde.

Man kann Fusebits in Bascom auch manuell über einen Dialog programmieren. Da müssten diese dann so aussehen:



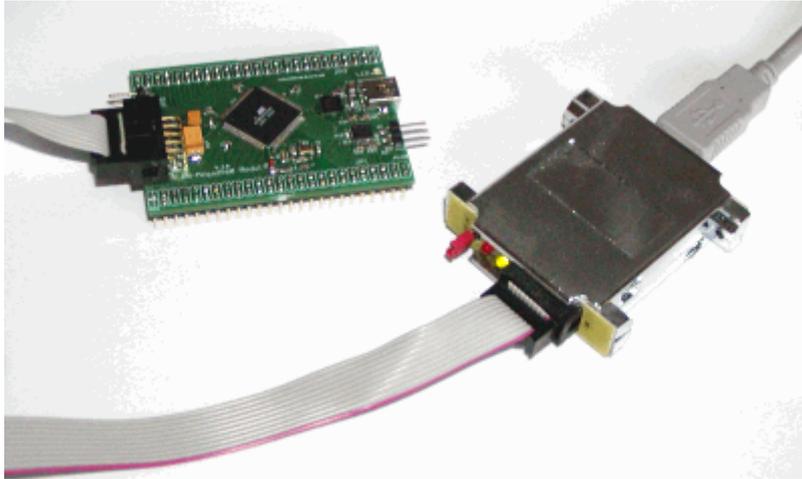
Beim manuellen einstellen der Fusebits muss man allerdings etwas vorsichtig vorgehen. Falsche Einstellungen können den Controller quasi unbrauchbar machen weil er dann nicht mehr über die üblichen ISP-Programmer programmierbar wäre. Insbesondere bei FUSEBIT 98DCB und FUSEBIT G muss man sehr vorsichtig sein.

Wie schon erwähnt, sie können sich jedoch die manuelle Programmierung der Fusebits sowieso ersparen., denn zum einen sind die bei unserer Auslieferung schon korrekt eingestellt und zum anderen würde die Anweisung

```
$prog , 255 , &B11011001 ,
```

im Programmcode völlig ausreichen damit Bascom alles alleine macht. Vorausgesetzt Sie haben entweder den schon beschriebenen parallelen oder den USB-Programmer.

jetzt aber zurück zu unserem Vorhaben den Blink-Beispielprogramm. haben Sie den Quellcode eingegeben oder geladen, so müssen Sie spätestens jetzt das Modul RN-Mega2560 mit dem PC verbinden. Auf dem unteren Bild wird dazu der USB-Programmer genutzt. Wie Sie erkennen ist links beim Programmer ein roter Jumper eingesteckt. Dadurch versorgt unser Programmer das Board gleichzeitig mit 5V, eine weitere Versorgungsspannung brauchen wir also nicht. Würden Sie statt dessen den parallelen Programmer nutzen, so müssten Sie das Board noch über die Stiftleisten mit 5V versorgen.



Zunächst compilieren wir das Programm indem wir das Symbol "schwarzes IC" anklicken. Sind im Quelltext noch Syntax Fehler enthalten, so werden diese nun unten aufgelistet, ansonsten wurde der Code korrekt erzeugt. Um das Programm nun zu übertragen, muss rechts oben in der Toolbar von Bascom ein grünes Symbol angeklickt werden. Auf dem unteren Bild ist es markiert:

```

#####
'erstebeispiel.bas Demo zu Modul RN-Mega2560
'für
'RoboterNetz Board RN-Mega2560 ab Version 1.0 und
'
'Aufgabe:
'Laesst LED blinken
'
'Autor: Frank
'Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de oder robotikhardware.de
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
#####

'Diese Anweisung setzt die Fusebits automatisch korrekt (Syntax $PROG LB, FB ,
$prog , 255 , &B11011001 , 'Quarz an / Teile:

$sregfile = "m2560def.dat"
$hwstack = 82 '80
$framesize = 68 '64
$swstack = 68 '44

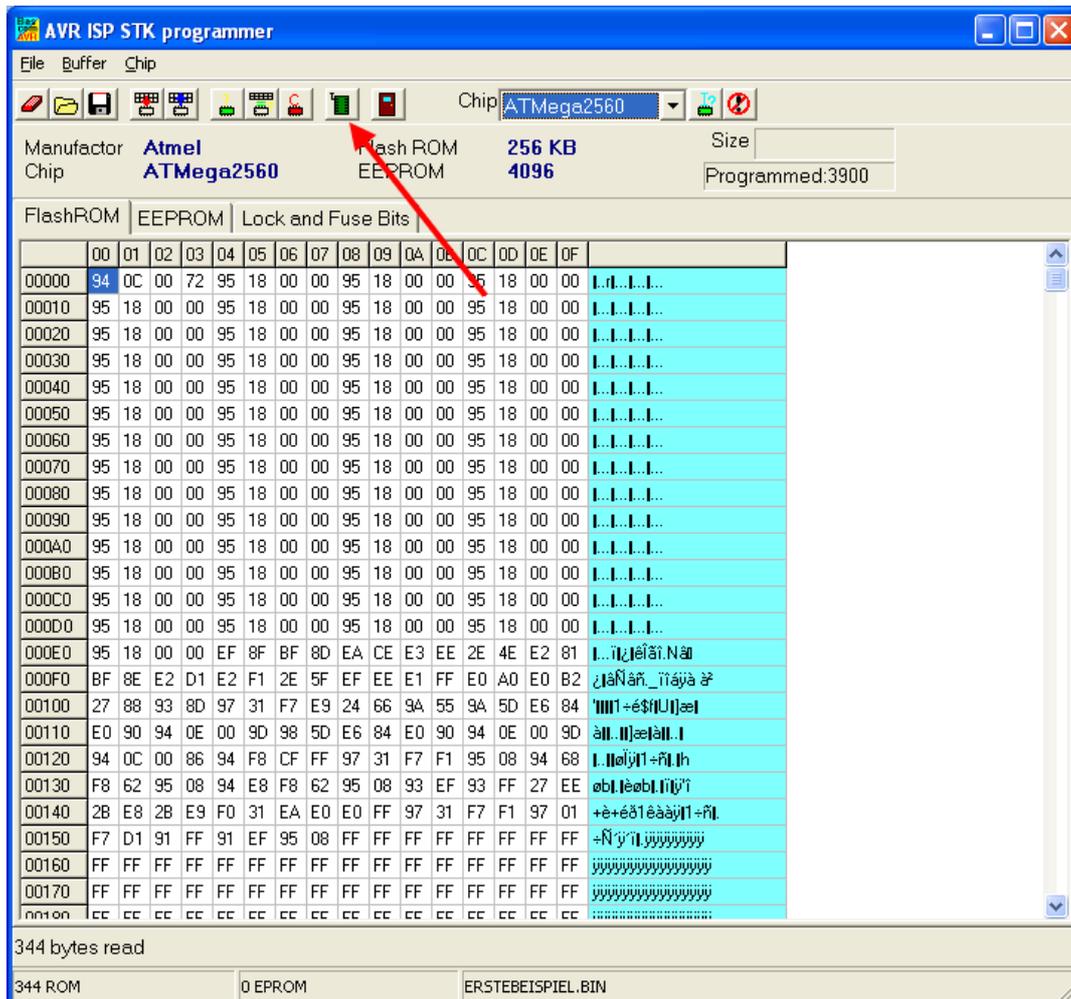
$crystal = 16000000 'Quarzfrequenz

Config Pind.5 = Output
Led Alias Portd.5

Do
  Led = 1
  Waitms 100
  Led = 0
  Waitms 100
Loop

End
  
```

Jetzt meldet sich der eingebaute Programmer von Bascom. Wenn alles richtig angeschlossen wurde, dann sollte dies in etwa so aussehen:



Wichtig ist das oben rechts bei Chip (im Bild blau markiert) ATMega2560 steht. Sollte hier etwas anderes stehen, dann haben Sie entweder noch einen veralteten Compiler oder aber irgend etwas stimmt mit der Programmer Verbindung oder Konfiguration nicht. Bevor Sie jedoch nach Fehlern suchen, sollte man das Programmerfenster nochmals schließen und es erneut versuchen. Auch das abziehen und neu anstecken des Boardes sollte man zunächst probieren, denn aufgrund der komplexen Firmware der USB-Programmer kommt es schon auch mal zu fehlerhaften Interpretation die beim zweiten Anstecken weg sind.

Sollte das alles nicht helfen, so ist folgende Webseite zu empfehlen:

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

Auf dieser Seite werden alle nur denkbaren Probleme aufgelistet.

Sollte der Chip korrekt angezeigt werden, so klickt man auch in diesem Dialog auf das grüne Symbol. Im oberen Bild ist dies ebenfalls durch einen Pfeil markiert. Nun wird das Programm in den Controller übertragen und anschließend automatisch verifiziert. Zudem werden automatisch, durch unsere schon genannte Anweisung, die Fusebits korrekt eingestellt. Das ganze dauert bei diesem kurzen Programm nur wenige Sekunden.

Bei erfolgreicher Übertragung sollte anschließend links unten in der Statusleiste dieses Dialogfeldes "VERIFY OK" zu lesen sein. Sofort nach der Übertragung führen die Programmer ein RESET beim Board aus, dadurch wird das Programm gestartet und die LED sollte dauerhaft blinken.

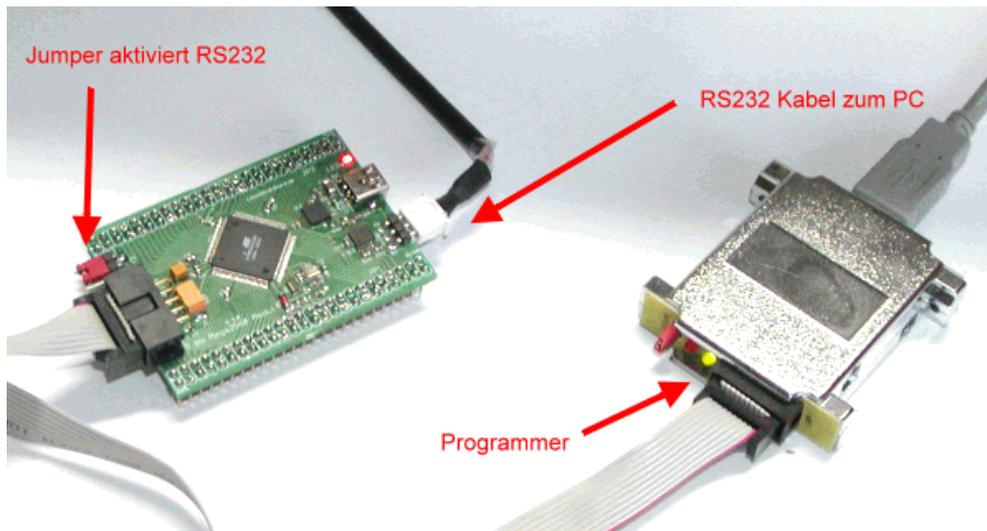
Das war´s!

Wir haben ein erstes Programm geschrieben und übertragen. Das Controllerboard arbeitet nun völlig unabhängig vom PC. Der Programmer kann also jederzeit entfernt werden, sobald das Board mit 5V versorgt wird, beginnt es zu blinken.

Jetzt prüfen wir die serielle Ausgabe über RS232

Jetzt demonstrieren wir wie einfach sich durch eine kleine Programmergänzung auch Daten mit dem PC über die serielle Schnittstelle austauschen lassen. Dazu müssen wir zunächst noch ein RS232-Kabel an die 3 polige Stiftleiste von RN-Mega2560 sowie an die 9 polige SUB-D Buchse vom PC anschließen. Passende RS232 kabel gibt's bei uns im Shop oder lassen sich einfach selbst löten.

Zudem müssen wir noch die serielle Schnittstelle am Board durch einen Jumper aktivieren, siehe Markierung im unteren Bild:



Unser Programm wir wie folgt:

```
'#####
'erstebeispiel2.bas Demo zu Modul RN-Mega2560
'für 'RoboterNetz Board RN-Mega2560 ab Version 1.0 und
'
'Aufgabe:
'Laesst LED blinken
'Gibt Zahlen ueber RS232 aus
'Autor: Frank
'Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de oder robotikhardware.de
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
'#####

'Diese Anweisung setzt die Fusebits automatisch korrekt (Syntax $PROG LB, FB , FBH , FBX )
$prog , 255 , &B11011001 , 'Quarz an / Teiler aus / Jtag aus

$regfile = "m2560def.dat"
$hwstack = 82 '80
$framesize = 68 '64
$swstack = 68 '44

$crystal = 16000000 'Quarzfrequenz

Config Pind.5 = Output
Led Alias Portd.5

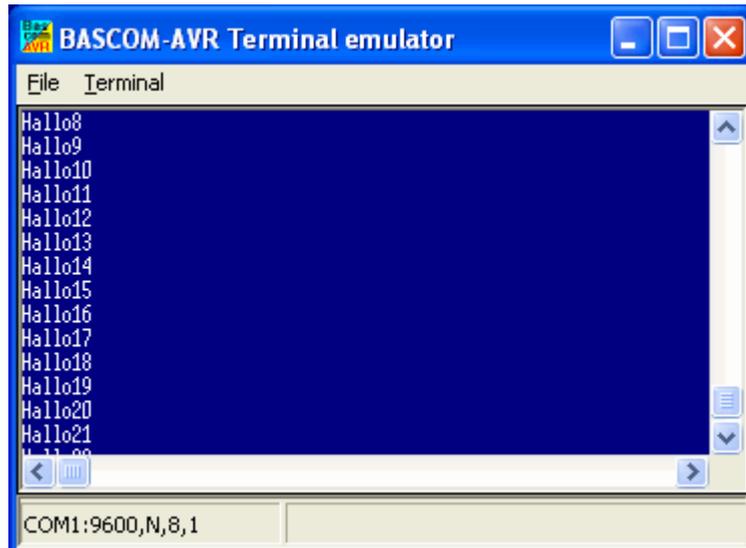
$baud = 9600
Dim I As Integer

Do
  Led = 1
  Waitms 100
  Led = 0
  Waitms 100
  Print "Hallo" ; I
  I = I + 1
Loop

End
```

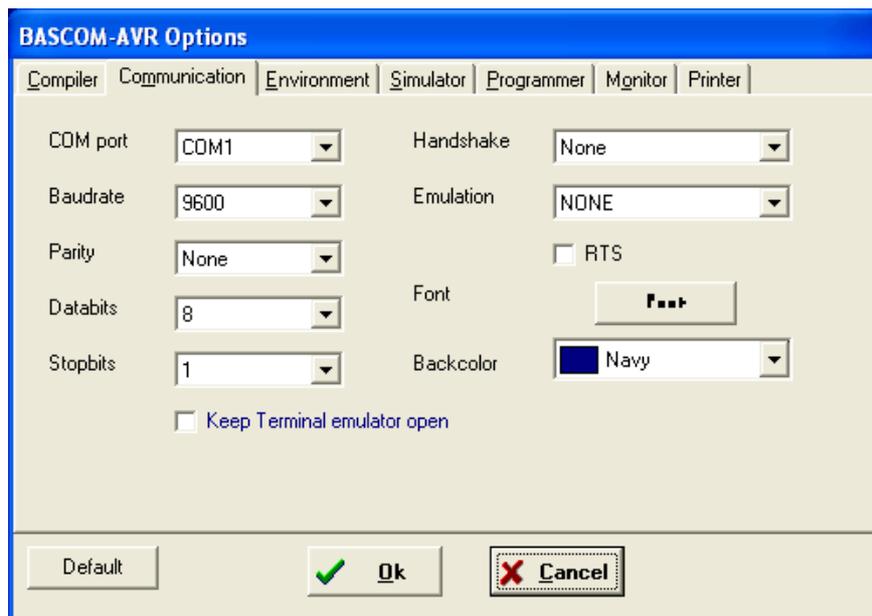
Das Programm wird auf die gleiche Weise compiliert und übertragen wie unser erstes Programm, wir ersparen uns dazu deshalb die Wiederholungen.

Zunächst werden wir gar keinen Unterschied bemerken, die LED blinkt wie zuvor. Wir müssen also noch ein Terminalprogramm starten, welches uns die eingehenden Daten auch anzeigt. Am einfachsten starten wir dazu das in Bascom integrierte Terminalprogramm indem wir rechts oben in der Haupttoolbar das Modem-Symbol anklicken (sieht auch ein wenig wie eine liegende Ampel aus). Jetzt sollte in etwa folgendes erscheinen:



Sie müssten eine ständig aufsteigende Zahlenfolge sehen. Diese Daten werden ständig von RN-mega2560 übertragen.

Erscheint nix oder wirre zeichen, so kann dies zwei Ursachen haben. Prüfen Sie zunächst ob auch die korrekte Baudrate im Terminalfenster eingestellt ist. Da wir im Quellcode 9600 Baud programmiert haben, sollte dies auch in ihrem Fenster so eingestellt werden. Über den Menüpunkt Terminal/Settings sollten folgende Einstellungen gewählt werden:



Sollte trotz dieser Einstellungen immer noch nix erscheinen, dann wurde vermutlich der dreipolige RS232 Stecker falschrum aufgesteckt. Drehen Sie diesen einfach um, dadurch wird RX/TX vertauscht und es sollte spätestens dann klappen.

Jetzt tauschen wir Daten über USB aus

Wir haben eben gesehen wie leicht man Daten über RS232 austauschen kann. Jetzt werden wir das gleiche Programm geringfügig ändern um zu zeigen das es genauso leicht auch über USB möglich ist. Wir entfernen zunächst das RS232 Kabel und verbinden statt dessen das Board über das beiliegende USB-Kabel mit dem PC.

Das Programm ändern wir soweit ab, das es so aussieht:

```
#####
'erstebeispiel3.bas Demo zu Modul RN-Mega2560
'für
'RoboterNetz Board RN-Mega2560 ab Version 1.0 und
'
'Aufgabe:
'Laesst LED blinken
'Gibt Zahlen ueber USB aus

'Autor: Frank
'Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de oder robotikhardware.de
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
#####

'Diese Anweisung setzt die Fusebits automatisch korrekt (Syntax $PROG LB, FB , FBH , FBX )
$prog , 255 , &B11011001 , 'Quarz an / Teiler aus / Jtag aus

$regfile = "m2560def.dat"
$hwstack = 82 '80
$framesize = 68 ' 64
$swstack = 68 '44

$crystal = 16000000 'Quarzfrequenz

Config Pind.5 = Output
Led Alias Portd.5

Config Com4 =9600 , Synchron = 0 , Parity = None , Stopbits =1 , Databits =8 , Clockpol =0
Open "com4:" For Binary As #4 'USB Buchse

Dim I As Integer

Do
  Led = 1
  Waitms 100
  Led = 0
  Waitms 100
  Print #4 , "Hallo" ; I
  I = I + 1
Loop

End
```

Wenn Sie das Programm genau vergleicht, dann wird man erkennen das lediglich die zwei Zeilen

```
Config Com4 =9600 , Synchron = 0 , Parity = None , Stopbits =1 , Databits =8 , Clockpol =0
Open "com4:" For Binary As #4 'USB Buchse
```

hinzukamen und die Print-Anweisung etwas verändert wurde:

```
Print #4 , "Hallo" ; I
```

Da RN-Mega2560 vier serielle Schnittstellen besitzt, wird für USB einfach die vierte Schnittstelle ausgewählt und die Baudrate festgelegt. Beim Print Befehl muss man lediglich noch angeben das über die vierte serielle Schnittstelle ausgegeben wird. **Wer sagt jetzt noch das USB kompliziert ist , einfacher geht's ja wohl kaum?**

Empfangen können die Daten wie zuvor mit dem eingebauten Bascom Terminalprogramm. Nicht vergessen das Sie dort aber für USB einen anderen COM-Port anwählen müssen, gewöhnlich 3 weil der USB-reiber COM 3 dort eingerichtet hat.

Natürlich geht auch beides

Selbstverständlich kann man auch in einem Programm gleichzeitig Daten über mehrer Schnittstellen ausgeben. Dies zeigt das untere Demoprogramm. Das Demo demonstriert auch wie man im Quellcode abfragen kann ob überhaupt eine USB-Verbindung besteht. Zudem wird bei diesem Demo noch ein Signalton über Port G5 ausgegeben. Dieser ist natürlich nur hörbar, wenn man ein Piezo-Lautsprecher an diesen Pin anschließt.

```
#####
'Mega2560modultest_rs232.bas Demo Zu Modul Rn -mega2560
'für RoboterNetz Board RN-Mega2560 ab Version 1.0 und
'Nicht vergessen vorher USB Driver zu installieren
'Ist auf CD von Robotikhardware.de vorhanden
'
'Aufgabe:
'Gibt eine Zeile über die 3 polige RS232
'als auch eine Zeile über die USB Schnittstelle aus
'Autor: Frank
>Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de oder robotikhardware.de
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
#####

'Diese Anweisung setzt die Fusebits automatisch korrekt (Syntax $PROG LB, FB , FBH , FBX )
$prog , 255 , &B11011001 , 'Quarz an / Teiler aus / Jtag aus

$regfile = "m2560def.dat"
$hwstack = 82 '80
$framesize = 68 ' 64
$swwstack = 68 '44

$crystal = 16000000 'Quarzfrequenz
$baud = 9600

Sound Portg.5 , 400 , 450 'BEEP
Sound Portg.5 , 400 , 250 'BEEP
Sound Portg.5 , 400 , 450 'BEEP

Config Pind.5 = Output
Led Alias Portd.5

Config Pine.5 = Input
Usb Alias Pine.5 'Ist 1 wenn USB angeschlossen

Config Com4= 9600 , Synchron= 0 , Parity =None , Stopbits =1 , Databits = 8 , Clockpol = 0
Open "com4:" For Binary As #4 'USB Buchse

Dim I As Integer
Dim U As Integer

For I = 1 To 10
  Led = 1
  Waitms 100
  Led = 0
  Waitms 100
Next I

I = 0
Do
  I = I + 1
  'Dies wird über herkömmliche RS232 ausgegeben
  Print "Hallo Seriell 1 (mit MAX) : " ; I
  If Usb = 1 Then Print "USB ist angeschlossen"
  Print

  'Dies wird über USB Schnittstelle ausgegeben
  Print #4 , "Hallo Seriell 4 (USB) : " ; I
  Wait 1
Loop

End
```

Daten per USB empfangen

Natürlich lassen sich Daten über RS232 oder USB nicht nur ausgeben sondern auch empfangen. Auch dazu ein kleines Beispiel das Eingaben per USB empfängt.

```
#####
'mega2560modultest_usb_input.bas Demo Zu Modul Rn -mega2560
'für 'RoboterNetz Board RN-Mega2560 ab Version 1.0 und
'Nicht vergessen vorher USB Driver zu installieren
'Ist auf CD von Robotikhardware.de vorhanden
'
'Aufgabe:
'Empfängt Zeichen über RS232 und gibt dieses als
'Zeichen und als ASCII-Zahlencode zurück
'Autor: Frank
'Weitere Beispiele und Beschreibung der Hardware
'unter http://www.Roboternetz.de oder robotikhardware.de
'Eigene Programmbeispiele sind im Roboternetz gerne willkommen!
#####

'Diese Anweisung setzt die Fusebits automatisch korrekt (Syntax $PROG LB, FB , FBH , FBX )
$prog , 255 , &B11011001 , 'Quarz an / Teiler aus / Jtag aus

$regfile = "m2560def.dat"
$hwstack = 82 '80
$framesize = 68 ' 64
$swstack = 68 '44

$crystal = 16000000 'Quarzfrequenz
$baud = 9600

Config Pind.5 = Output
Led Alias Portd.5

Config Pine.5 = Input
Usb Alias Pine.5 'Ist 1 wenn USB angeschlossen

Config Com4 =9600 , Synchron =0 , Parity =None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Open "com4:" For Binary As #4 'USB Buchse

Dim I As Integer
Dim U As Integer
Dim B As Byte

For I = 1 To 10
  Led = 1
  Waitms 100
  Led = 0
  Waitms 100
Next I

  Wait 1
  Print #4 , "Hallo USB"

I = 0
Do
  B = Waitkey(#4)
  Waitms 500
  Print #4 , Chr(b )
  Print #4 , Chr(b) ; "(" ; B ; ")" ;
Loop

End
```

Dieses testprogramm wartet stets auf ein Zeichen und gibt dieses dann als Zeichencode und als Zahlencode wieder zurück. Es lässt sich einfach testen indem man bei einem aktiviertem Bascom Terminalfenster auf eine Buchstabentaste drückt. Dieser Buchstabe wird dann vom Board RN-Mega2560 empfangen und wieder zurück wie geschildert zurückgesendet und im Terminalfenster angezeigt.

Achtung, in der derzeitigen neuen Bascom Version 1.11.8.3 gibt es bei einigen Bascom Versionen noch einen BUG bei dem CHR-Befehl, wenn diese in Zusammenhang mit COM4 benutzt wird. Sollten die Daten also nicht ganz so wie erwartet zurück kommen, liegt es an diesem Bascom Bug. Der Hersteller MCSELEC wird diesen jedoch im nächsten Update korrigieren (eventuell ist es inzwischen auch schon korrigiert).

Damit haben Sie den ersten Einstieg erfolgreich abgeschlossen.

Wenn Sie die Demoprogramme gründlich studieren werden Sie viele Sachen davon ableiten und in eigenen Programmen verwenden können. Der große Mikrocontroller auf RN-Mega2560 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 vor allem auch das Datenblatt zum ATmega2560 empfehlen. Dieses befindet sich natürlich mit auf der CD.

Auch finden Sie unter www.rn-wissen.de viele Quellcodebeispiele zu kleineren Boards wie RN-Mega128Funk, RN-Control, RN-MiniControl oder RN-Mega8. Alle diese Bascom-Beispiele lassen sich mit geringen Änderungen auch an RN-Mega2560 anpassen. So werden Sie schnell sehen wie Sie Ultraschallsensoren, Infrarotsensoren, temperatursensoren, einen Kompass, Drehgeber und viele mehr ganz einfach anschließen und programmieren können. Es ist zu erwarten das in Kürze auch weitere Beispiele zu RN-Mega2560 unter www.rn-wissen.de veröffentlicht werden. RN-Wissen ist eine offene WIKI-Seite, jeder kann dort Beiträge erweitern oder eigene Programme veröffentlichen, es wäre schön wenn auch Sie davon Gebrauch machen würden, es hilft allen.

Weiterhin sind zu empfehlen:

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". Bezugsquelle: z.B. robotikhardware.de oder Autor
Buch: Programmieren mit Bascom AVR Autor Kühnel	Buchhandel oder robotikhardware beziehbar
http://www.roboternetz.de	Das inzwischen größte deutsche Forum was sich mit Robotik- und Mikrocontrollern sowie auch der Bascom-Programmierung beschäftigt. Hier kann man einfach nachfragen.
http://www.rn-wissen.de	Umfangreiches Wissensportal zum Thema Roboter und AVR-Controller. Hier findet man viele Erklärungen und Beispielsprogramme
http://www.atmel.com/	Der Hersteller des Controllers bietet zahlreiche Datenblätter und Assembler Entwicklungsumgebung
http://www.roboternetz.de/wissen/index.php/RN-Board_FAQ-Seite	Frage und Antwortseite zu RN-Boards
http://www.roboternetz.de/wissen/index.php/Kategorie:Quellcode_Bascom	Beispielprogramme zu AVR Boards
http://www.roboternetz.de/wissen/index.php/Buchvorstellungen	Passende Buchvorstellungen

und eine ganz wichtige Seite ist:

<http://www.rn-wissen.de>

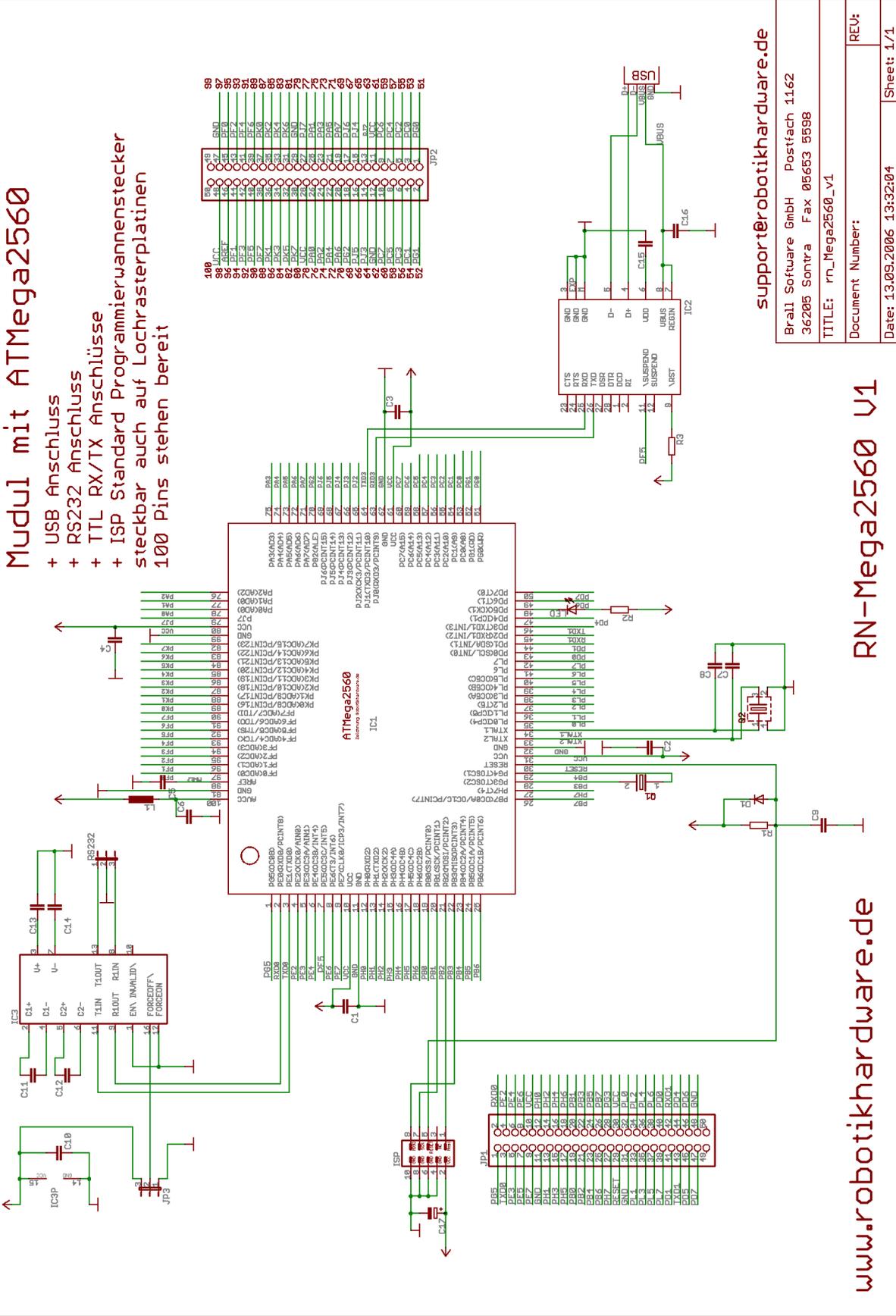
und

<http://www.roboternetz.de>

Schaltplan des Boards

Modul mit ATmega2560

- + USB Anschluss
- + RS232 Anschluss
- + TTL RX/TX Anschlüsse
- + ISP Standard Programmierwannenstecker steckbar auch auf Lochrasterplatten
- 100 Pins stehen bereit



support@robotikhardware.de

Brall Software GmbH Postfach 1162
36208 Sontra Fax 05653 9598

TITLE: rn_Mega2560_v1

Document Number:

Date: 13.09.2006 13:32:04

www.robotikhardware.de

RN-Mega2560 v1

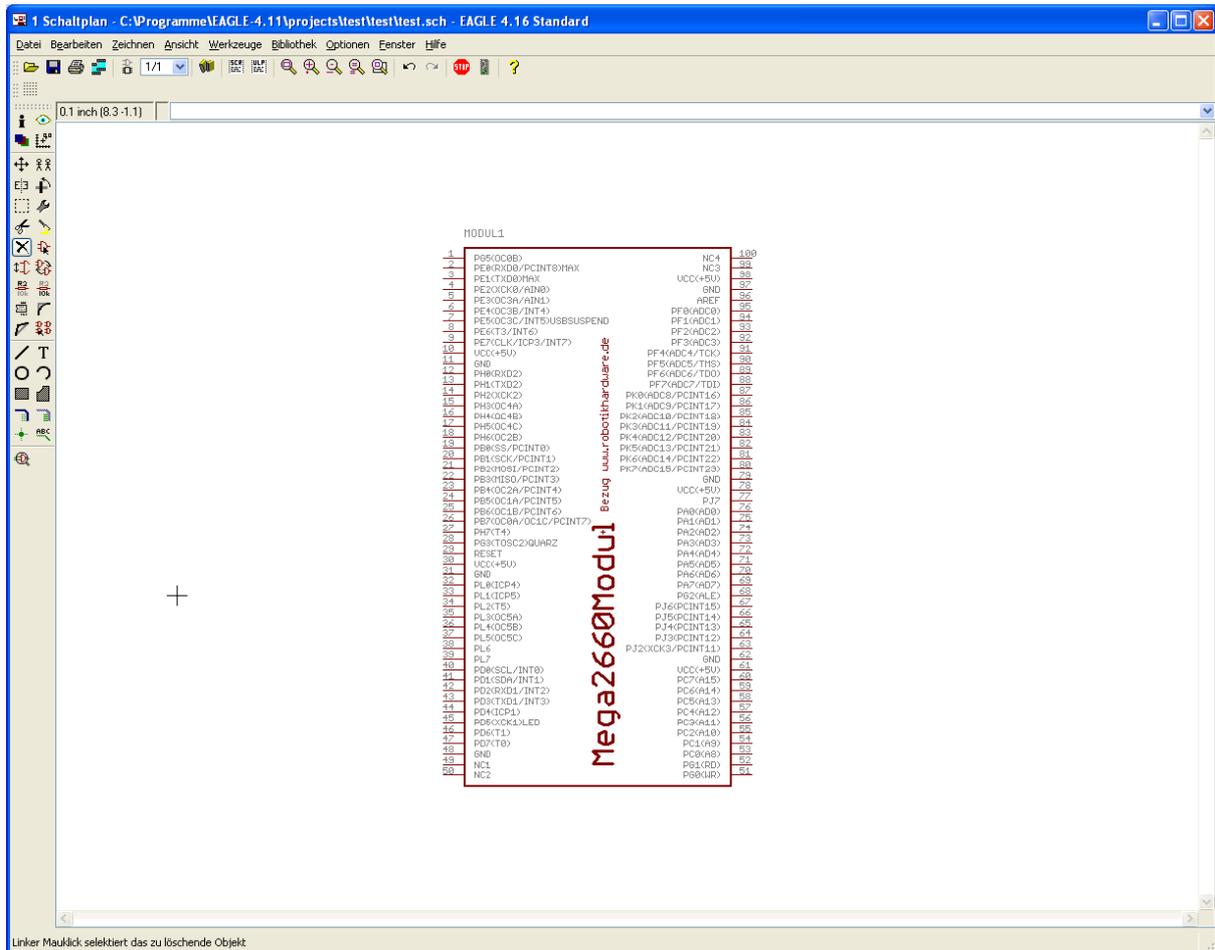
REV:

Sheet: 1/1

RN-Mega2560 in eigene Schaltungen einbinden

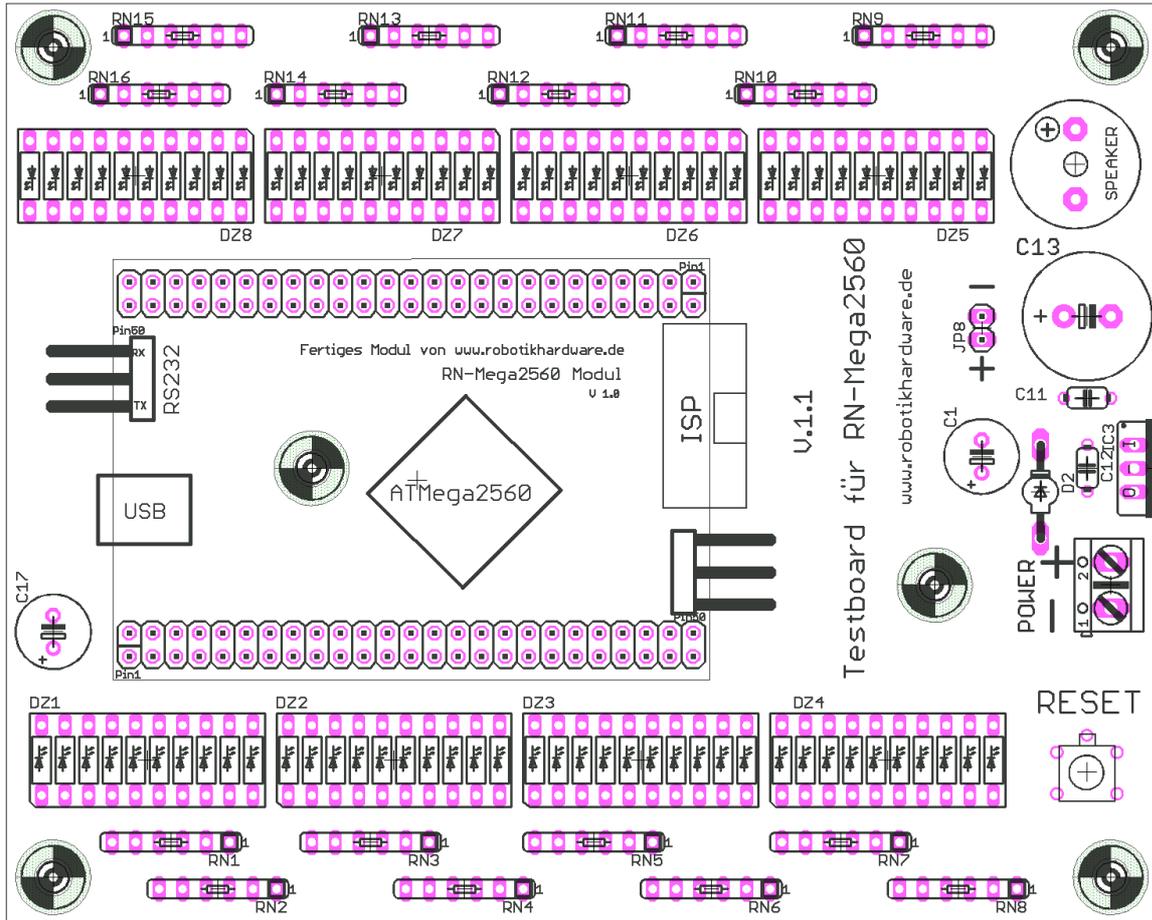
Wenn mit der Layoutsoftware EAGLE von CadSoft Schaltungen entwickelt werden, dann kann das Modul RN-Mega2560 auch als Modul eingefügt werden. Dadurch lassen sich Steckplätze sehr einfach auf eigenen Platinen vorsehen. Eine entsprechende Eagle-Library mit dem Modul liegt auf CD als **robotikhardware.lbr** in dem Anleitungsverzeichnis

Das Modul sieht in Eagle so aus:



Beispiel eines Testboardes (Bestückungsplan)

Eagle-Dateien sind auf CD enthalten



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

RN-Mega2560 Modul

Modul mit leistungsstarkem 100 Pin AVR
und USB / RS232 / ISP passend
für alle 2,54 Raster / Platinen



Exklusiv bei www.robotikhardware.de

Online-Bestellung <http://www.robotikhardware.de>

Anwendungsprogramme und Programmschnipsel findet man auch im Roboternetz-Download-Bereich.
Es wäre schön wenn auch Sie dort etwas beitragen könnten

Haftung, EMV-Konformität

Alle Teile der Schaltung wurden sorgfältigst geprüft und getestet. Trotzdem kann ich natürlich keine Garantie dafür übernehmen, daß alles einwandfrei funktioniert. Insbesondere übernehme ich keine Haftung für Schäden, die durch Nachbau, Inbetriebnahme etc. der hier vorgestellten Schaltungen entstehen. Derjenige, der den Bausatz zusammenbaut oder das Modul fertigt und in einem Gehäuse montiert, gilt als Hersteller und ist damit selbst für die Einhaltung der geltenden Sicherheits- und EMV-Vorschriften verantwortlich.

Für Schäden die durch fehlerhaften Aufbau entstanden sind, direkt oder indirekt, ist die Haftung generell ausgeschlossen. Schadensersatzansprüche, gleich aus welchem Rechtsgrund, sind ausgeschlossen, soweit nicht vorsätzliches oder grob fahrlässiges Handeln vorliegt. Sofern wir haften, umfaßt unsere Haftung nicht solche Schäden, die nicht typischerweise erwartet werden konnten. Haftung und Schadensersatzansprüche sind auf den Auftragswert / Bauteilwert beschränkt. Bei der Lieferung von Fremdprodukten als auch Software gelten über diese Bedingungen hinaus die besonderen Lizenz- oder sonstigen Bedingungen des Herstellers. Die Anleitung wurde sorgfältig erarbeitet und geprüft. Dennoch können wir nicht ausschließen das sich noch Fehler eingeschlichen haben. Auch dafür kann keine Haftung übernommen.

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. Vor Öffnen eines Gerätes stets den Netzstecker ziehen oder sicherstellen, daß das Gerät stromlos ist. Bauteile, Baugruppen oder Geräte dürfen nur in Betrieb genommen werden, wenn sie vorher berührungssicher in ein Gehäuse eingebaut wurden. Während des Einbaus müssen sie stromlos sein.

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 nichtgewerblichen 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 Einflusses liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen. Bausätze sollten bei Nichtfunktion mit einer genauen Fehlerbeschreibung (Angabe dessen, was nicht funktioniert...denn nur eine exakte Fehlerbeschreibung ermöglicht eine einwandfreie Reparatur!) und der zugehörigen Bauanleitung sowie ohne Gehäuse zurückgesandt werden. Zeitaufwendige Montagen oder Demontagen von Gehäusen müssen wir aus verständlichen Gründen zusätzlich berechnen. Bereits aufgebaute Bausätze sind vom Umtausch ausgeschlossen. Bei Installationen und beim Umgang mit Netzspannung sind unbedingt die VDE-Vorschriften zu beachten. Geräte, die an einer Spannung ≤ 35 V betrieben werden, dürfen nur vom Fachmann angeschlossen werden. In jedem Fall ist zu prüfen, ob der Bausatz für den jeweiligen Anwendungsfall und Einsatzort geeignet ist bzw. eingesetzt werden kann.

Die Inbetriebnahme darf grundsätzlich nur erfolgen, wenn die Schaltung absolut berührungssicher in ein Gehäuse eingebaut ist. Sind Messungen bei geöffnetem Gehäuse unumgänglich, so muß aus Sicherheitsgründen ein Trenntrafo zwischengeschaltet werden, oder, wie bereits erwähnt, die Spannung über ein geeignetes Netzteil, (das den Sicherheitsbestimmungen entspricht) zugeführt werden. Alle Verdrahtungsarbeiten dürfen nur im spannungslosen Zustand ausgeführt werden.

Derjenige, der einen Bausatz fertigt 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.

Betriebsbedingungen

Der Betrieb der Baugruppe darf nur an der dafür vorgeschriebenen Spannung erfolgen.

Bei Geräten mit einer Betriebsspannung 35 Volt darf die Endmontage nur vom Fachmann unter Einhaltung der VDE Bestimmungen vorgenommen werden.

Die Betriebslage des Gerätes ist beliebig.

Bei der Installation des Gerätes ist auf ausreichenden Kabelquerschnitt der Anschlußleitungen zu achten!

Die angeschlossenen Verbraucher sind entsprechend den VDE Vorschriften mit dem Schutzleiter zu verbinden bzw. zu erden.

Die zulässige Umgebungstemperatur (Raumtemperatur) darf während des Betriebes 0°C und 40°C nicht unter-, bzw. überschreiten.

Das Gerät ist für den Gebrauch in trockenen und sauberen Räumen bestimmt.

Bei Bildung von Kondenswasser muß eine Akklimatisierungszeit von bis zu 2 Stunden abgewartet werden.

In gewerblichen Einrichtungen sind die Unfallverhütungsvorschriften des Verbandes der gewerblichen Berufsgenossenschaften für elektrische Anlagen und Betriebsmittel zu beachten.

In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfewerkstätten ist das Betreiben von Baugruppen durch geschultes Personal verantwortlich zu überwachen.

Betreiben Sie die Baugruppe nicht in einer Umgebung in welcher brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können.

Falls das Gerät einmal repariert werden muß, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen!

Das Gerät wurde für den Modellbau- und Robotik-Bastler konzipiert, welche das Produkt unter ständiger Aufsicht für Experimente nutzen.

Für jegliche andere Verwendung wird keinerlei Garantie oder Haftung übernommen.

Dringt irgendeine Flüssigkeit in das Gerät ein, so könnte es dadurch beschädigt werden.

Für medizinische Anwendungen, Geräte oder andere Anwendungen bei der ein Ausfall oder eine Fehlfunktion einen Personen- oder Sachschaden zur Folge ist die Schaltung nicht geeignet und darf dort nicht eingesetzt werden. Es wird keinerlei Haftung übernommen..