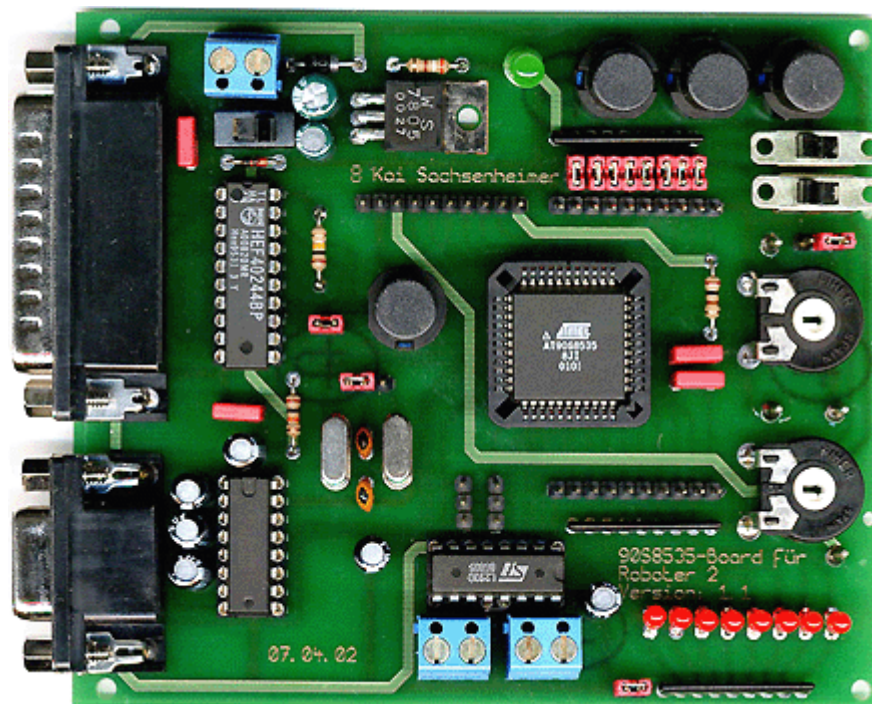


Bau- und Bedienungsanleitung des AVR Controllerboards mit dem 90S8535 Mikrocontroller von Atmel

Stand: 27.04.2002



Inhalt

INHALT	2
TECHNISCHE DATEN	3
FEATURES.....	3
EIGENSCHAFTEN DER AVR CONTROLLER.....	3
BESTÜCKUNG DER PLATINE	4
BESTÜCKUNGSPLAN.....	4
REIHENFOLGE DER BESTÜCKUNG ENTSPRECHEND DER TABELLE VON OBEN NACH UNTEN.....	5
STÜCKLISTE DES AVR-BOARDS.....	6
<i>Sonstiges:</i>	6
ANSICHT DER PLATINE VON OBEN	7
KURZE BESCHREIBUNG DER HARDWARE	8
<i>Die digitalen I/O-Ports</i>	8
<i>Die analogen Eingänge</i>	8
<i>Die Referenzspannung</i>	8
<i>Einstellung des LCD-Kontrastes</i>	8
<i>Die serielle Schnittstelle</i>	8
<i>Lastausgänge</i>	8
<i>Reset-Jumper K4</i>	8
<i>Jumper Stellungen</i>	9
INBETRIEBNAHME	9
SOFTWARE ZUM PROGRAMMIEREN DES CONTROLLERS	9
PROGRAMMIEREN DES CONTROLLERS MIT BASCOM	9
FEATURES.....	9
BÜCHER ZUM THEMA BASCOM.....	10
EINSTELLUNGEN IN BASCOM.....	10
ENDLICH DEN CONTROLLER PROGRAMMIEREN.....	12
PROGRAMMBEISPIELE FÜR BASCOM	13
KOMMUNIKATION MIT DER SERIELLEN SCHNITTSTELLE.....	13
ANSTEUERUNG DES BRÜCKENTREIBERS.....	14
<i>Aufgabe des Brückentreibers</i>	14
<i>Prinzipieller Aufbau des Brückentreibers</i>	14
<i>Ansteuerung eines Motors</i>	14
A. <i>Rechtslauf</i>	15
B. <i>Linkslauf</i>	15
C. <i>Freilauf</i>	15
D. <i>Bremsen</i>	15
<i>Funktionstabelle</i>	15
<i>Programmbeispiel</i>	16
ANSCHLUSS EINES LCD-DISPLAYS.....	16
<i>Pinnbelegung und Anschluss an die Platine</i>	17
<i>Programmbeispiel</i>	17
SCHALTPLAN	18

Technische Daten

- Betriebsspannung 8 bis 16 Volt DC
(je nach dem welche Elkos verwendet wurden auch bis 30V)
- Stromaufnahme in Ruhe: 90 mA
- Platinengröße: 120mmx100mm

Features

- AT90S8535 Controller von Atmel
- 8-Bit-RISC-Architektur
- 8 K Programmspeicher (Flash)
- 512 Bytes EEPROM
- Taktfrequenz 4 bzw. 8 MHz (mit Jumper K5 wählbar)
- 8 Analoge Eingänge bzw. 8 Digitale Ein/Ausgänge (Port A)
- 8 Digitale Ein/Ausgänge (Port C)
- 4 Lastausgänge für Motoren (M1 und M2) max. Strom 600 mA
- 8 LEDs (LD2 bis LD9)
- 2 Ausgänge für die Ansteuerung von Servos (K6, K7)
- 3 Taster und 2 Schalter zur Eingabe (S3 bis S7)
- Resettaster (S2)
- Seriell Interface zur Kommunikation mit einem Terminal-Programm (K2)
- Anschluss des Parallelport des PCs zum programmieren des Controllers (K1)
- Einstellung der Referenzspannung des A/D-Wandlers (P1)
- Kontrasteinstellung für ein LCD-Display (P2)
- Einfacher Anschluss eines LCD-Displays (Stiftleiste K17)

Eigenschaften der AVR Controller

Die AVR-Mikrocontroller-Familie von Atmel erblickte 1997 das Licht der Welt. Die neue 8-Bit-RISC-Architektur wurde von dem norwegischen Designhaus Nordic VLSI in Trondheim in enger Zusammenarbeit mit dem schwedischen Compilerhersteller IAR entwickelt. Später kaufte Atmel dieses europäische Design um eine eigene Mikrocontrollerfamilie auf den Markt zu bringen. Bei der RISC-Architektur wird die Taktfrequenz intern nicht geteilt, d.h. ein 4 MHz Quarz erzeugt einen Befehlsdurchsatz von bis zu 4 Millionen Befehlen pro Sekunde (Das sind 4 MIPS, Million Instructions per Second).

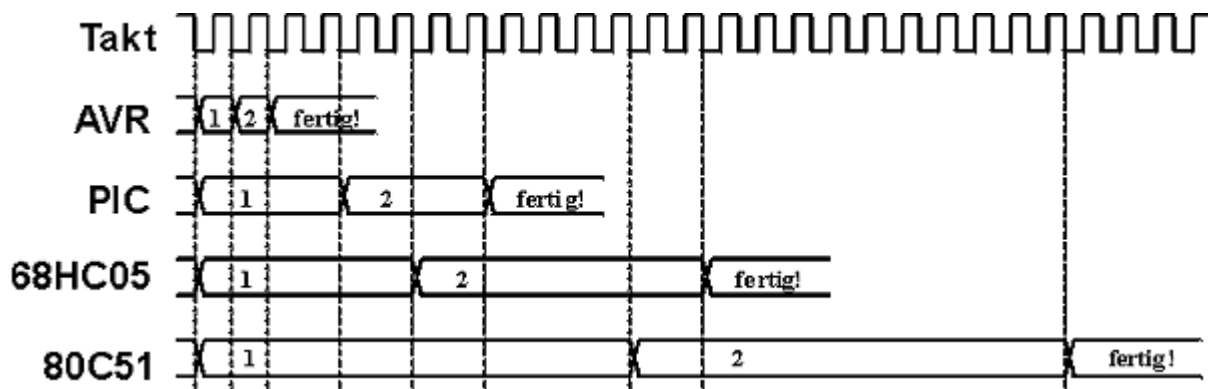


Abb.: Die Ausführungszeiten für zwei Befehle bei verschiedenen Prozessoren im Vergleich

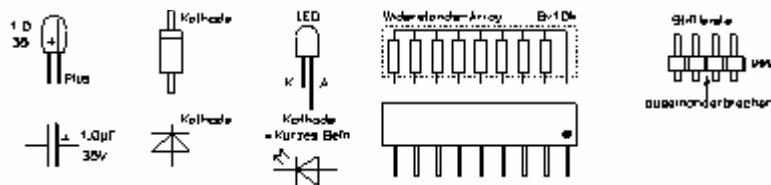
Deshalb können die AVR-Controller mit ihrer relativ niedrigen Taktfrequenz auch ohne Probleme mit anderen Controller-Familien, die wesentlich höher getaktet sind, mithalten. Durch die niedrige Taktfrequenz ist die Stromaufnahme auch sehr klein. Die AVR-Controller-Familie besitzt je nach Controllertyp folgende **Eigenschaften**:

- bis zu 48 programmierbare Ein-/Ausgabeleitungen

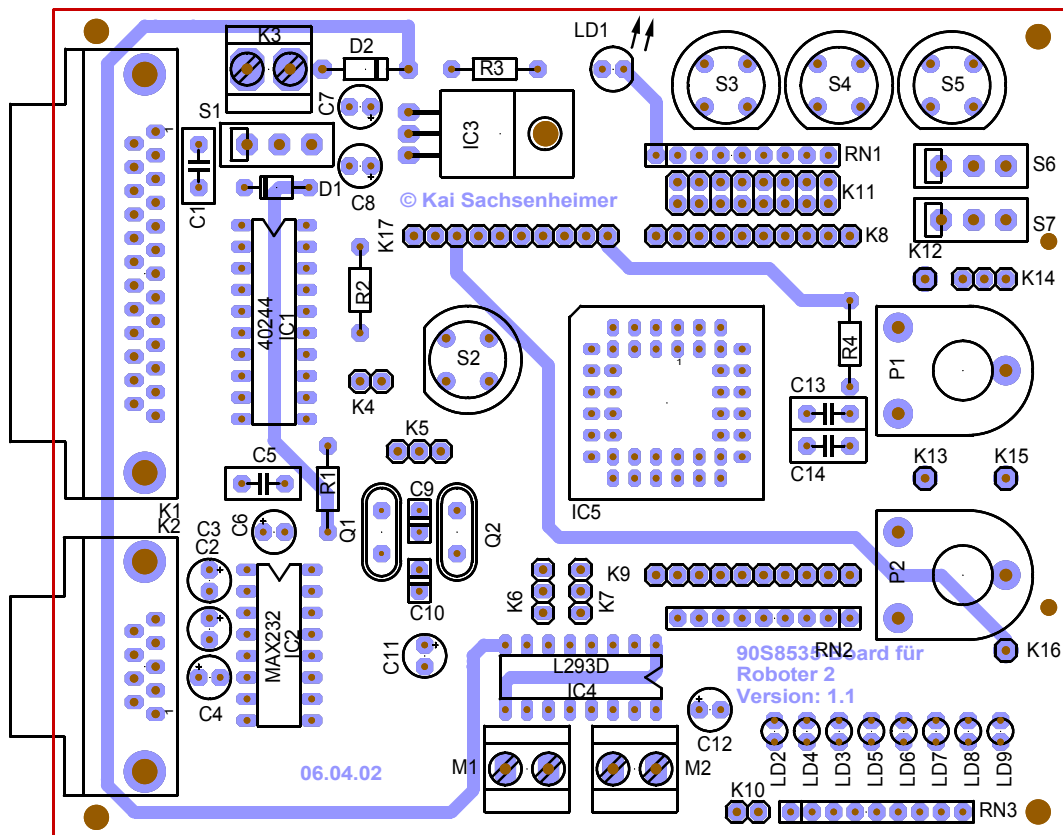
- ein programmierbarer UART
- eine synchrone SPI-Schnittstelle
- bis zu drei 8 Bit Timer/Counter
- ein 16 Bit Timer/Counter mit Compare- und Capture-Funktionen
- bis zu vier PWM-Ausgänge
- eine Echtzeituhr
- ein 10 Bit - A/D-Wandler mit bis zu 8 Kanälen
- ein Brown Out – Detector
- ein Analogkomparator
- ein integrierter Watchdog-Timer sowie
- die Möglichkeit der Programmierung im System

Bestückung der Platine

Polung der Kondensatoren, Dioden, LEDs, Widerstands-Arrays beachten:



Bestückungsplan



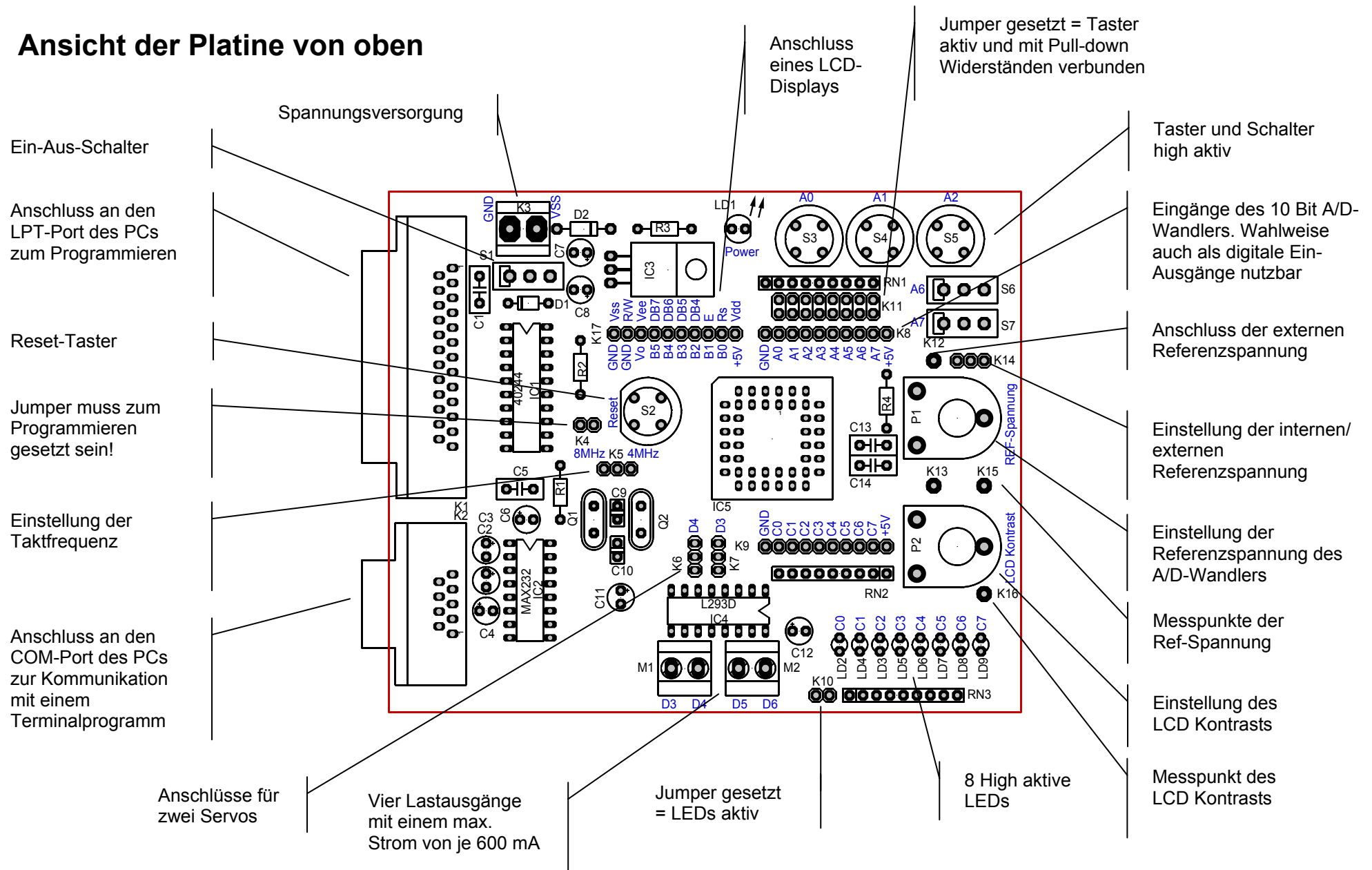
Reihenfolge der Bestückung entsprechend der Tabelle von oben nach unten

Bauteile:	Werte:	Zu beachten:
Widerstände und Potis		Farbcode
Widerstands-Arrays	8x10k und 8x1k	Polung
Dioden	1N4148 1N4001	Polung
Kondensatoren	10nF, 100nF, 22p	
Elkos	1 μ F, 10 μ F, 100 μ F	Polung
LEDs		Polung Die kurzen Beine der 8 LEDs (LD2-LD3) müssen zum Widerstandnetzwerk (RN3) zeigen
Jumper		Stiftleiste auseinander brechen, kurze Stiftseite von oben durch die Platine stecken Jumper noch nicht setzen
Stiftleisten		Auf erforderliche Pinzahl kürzen
IC-Sockel und Spannungsregler		Polung beachten ICs noch nicht einsetzen
Taster und Schalter		
D-Sub-Buchsen und Stecker		Festhalteklappen Zusammenbiegen, sodass sie durch die Bohrungen passen und von unter verlöten
Stromversorgungsklemme und Klemmen für Motoren		

Stückliste des AVR-Boards

Anzahl:	Beschreibung:	Name:	Bestell Nr.:	Firma:
3	Kondensator, 100nF, RM5	C1, C5, C14	MKS-2 100N	www.reichelt.de/
6	Elko 10µF, 35V	C2, C3, C4, C6, C11, C12	rad 10/35	www.reichelt.de/
1	Elko 100µF, 16V	C7	rad 100/16	www.reichelt.de/
1	Elko 1µF, 63V	C8	rad 1/63	www.reichelt.de/
1	Kondensator, 10nF, RM5	C13	MKS-2 10N	www.reichelt.de/
2	Kondensator, 22pF, RM2,5	C9, C10	Kerko 22p	www.reichelt.de/
1	Dioden 1N4148	D1	1N 4148	www.reichelt.de/
1	Dioden 1N4001	D2	1N 4001	www.reichelt.de/
1	74HC(T)244 oder 40244	IC1	74HC 244	www.reichelt.de/
1	MAX232	IC2	MAX 232 CPE	www.reichelt.de/
1	Spannungsregler 7805	IC3	µA 7805	www.reichelt.de/
1	Brückentreiber 293D	IC4	L 293 D	www.reichelt.de/
1	Controller AT90S8535 PLCC	IC5	AT 90S8535 PLCC	www.reichelt.de/
1	IC Fassung DIL 20	IC1	GS 20	www.reichelt.de/
2	IC Fassung DIL 16	IC2, IC4	GS 16	www.reichelt.de/
1	PLCC-Fassung	IC5	PLCC 44	www.reichelt.de/
1	D-SUB25 Stecker gewinkelt	K1	D-SUB ST 25EU	www.reichelt.de/
1	D-SUB9 Buchse gewinkelt	K2	D-SUB BU 09EU	www.reichelt.de/
3	Anschlussklemme, RM5	K3, M1, M2	AKL 055-02	www.reichelt.de/
2	Stiftleiste 1x36pol, gerade	K4, K5, K6, K7, K8, K9, K10, K14, K17	Stiftl. 36G	www.reichelt.de/
1	Stiftleiste 2x10pol, gerade	K11	Stiftl. 2x10G	www.reichelt.de/
4	Lötstifte 1mm (1 Pack = 100 Stück)	K12, K13, K15, K16	RTM 1-100	www.reichelt.de/
1	LED, 5mm, grün	LD1	LED 5mm ST-gn	www.reichelt.de/
8	LED, 3mm, rot, Low-Current	LD2 – LD9	LED 3mm 2MA-rt	www.reichelt.de/
2	Poti, liegend, für Steckachse, 10k	P1, P2	PT 15-L 10K	www.reichelt.de/
1	Quarz 8 MHz HC18	Q1	8-HC18	www.reichelt.de/
1	Quarz 4 MHz HC18	Q2	4-HC18	www.reichelt.de/
1	Widerstand 10k	R1	1/4W 10K	www.reichelt.de/
1	Widerstand 100k	R2	1/4W 100K	www.reichelt.de/
1	Widerstand 270	R3	1/4W 270	www.reichelt.de/
1	Widerstand 100	R4	1/4W 100	www.reichelt.de/
2	Widerstandsnetzwerk 8x10k, 9pol	RN1, RN2	SIL 9-8 10K	www.reichelt.de/
1	Widerstandsnetzwerk 8x1k, 9pol	RN3	SIL 9-8 1K	www.reichelt.de/
3	Miniaturschiebeschalter 1xUM, Print	S1, S6, S7	70 80 54-22	www.conrad.de
4	Taster rund	S2 – S5	DT 6sw	www.reichelt.de/
Sonstiges:				
1	D-SUB9 Kabel (Stecker/Buchse)		AK 230	www.reichelt.de/
1	D-SUB25 Kabel (Stecker/Buchse)		AK 404	www.reichelt.de/
1	Netgerät mit ca. 9V und 200 mA			
12	Jumper		Jumper rt	www.reichelt.de/

Ansicht der Platine von oben



Kurze Beschreibung der Hardware

Die digitalen I/O-Ports

Der Port A des Controllers ist direkt auf die Stiftleiste K8 und der Port C auf die Stiftleiste K9 herausgeführt. Beide Ports besitzen Pull-Down-Widerstände, welche die Ports im Ruhezustand auf ca. 0V legen.

Am Port A sind zusätzlich drei Taster und zwei Schalter angeschlossen. Der Ruhezustand der Taster und Schalter ist ein Lowsignal (ca. 0V). Erst beim Betätigen der Taster beziehungsweise der Schalter ändert sich das Signal von dem Lowzustand (ca. 0V) in einen Highzustand (ca. 5V).

Erst wenn die entsprechenden Jumper auf der Stiftleiste K11 gesetzt sind, sind die Taster und Schalter mit dem jeweiligen Port des Controllers verbunden.

Am Port A kann man wahlweise digitale Signale ein-ausgebend oder mit Hilfe des 10 Bit A/D-Wandlers des Controllers analoge Signale (Spannungen von 0 bis 5V) einlesen.

Will man den A/D-Wandler benutzen, sollte man die entsprechenden Jumper der Stiftleiste K11 entfernen, weil sich sonst ein Spannungsteiler mit dem Widerstandsnetzwerk RN1 bildet und die Messwerte verfälscht.

Der Port C kann allerdings nur reine digitale Signale verarbeiten.

Ist der Jumper K10 gesetzt, können mit diesem Port direkt die acht LEDs auf der Platine angesteuert werden.

Die analogen Eingänge

Der Controller besitzt einen 10 Bit A/D-Wandler mit 8 Kanälen, der an der Stiftleiste K8 zusammengefasst ist. Wahlweise können diese Eingänge auch als normale I/O-Ports benutzt werden (siehe oben).

Die Referenzspannung

Die Referenzspannung für den A/D-Wandler lässt sich im Bereich von 0 bis 5 Volt mit dem Poti P1 einstellen. Man kann mit Hilfe eines Jumpers K 14 zwischen der internen und externen Referenzspannung wählen.

Einstellung des LCD-Kontrastes

Wie bei der Referenzspannung kann man die Spannung am Poti P2 zwischen 0 und 5V einstellen. Dadurch lässt sich der Kontrast des LCD-Displays am Pin Vo bzw. Vee der Stiftleiste K17 (siehe Ansicht der Platine von oben) variieren.

Die serielle Schnittstelle

Das Controllerboard verfügt über eine serielle Schnittstelle, die auf die D-SUB-Buchse K2 geführt ist. Diese ist mit dem Pegelwandler MAX232 realisiert. Mit ihr können Daten vom PC mit Hilfe eines Terminalprogramms gesendet und empfangen werden.

Lastausgänge

Mit Hilfe des Brückentreibers können 2 Motoren unabhängig voneinander angesteuert werden. Der maximale Strom der Motoren darf 600 mA nicht überschreiten. Die Spannung an den Brückentreiberausgängen entspricht der Versorgungsspannung, die an der Schraubklemme K3 anliegt. Bevor der Brückentreiber eine Spannung ausgibt, muss er zuerst aktiviert werden.

Dazu programmiert man den Controller so, dass er am Port D7 (siehe Schaltplan) ein Highsignal ausgibt. Dadurch werden beide Freigabeeingänge des Brückentreibers aktiviert, sodass er nun betriebsbereit ist.

Reset-Jumper K4

Der Jumper K4 muss zum programmieren gesetzt sein, damit der Controller beim Programmieren zurückgesetzt werden kann. Wird das Board von der parallelen Schnittstelle des PCs entfernt, kann der Controller durch einen entsprechenden Zustand an dem D-SUB-Stecker der parallelen Schnittstelle, auch während des Normalbetriebes zurückgesetzt werden. Um dies zu vermeiden kann man den Jumper K4 nach dem Programmieren entfernen.

Jumper Stellungen

Jumper	Normal	Funktion
K4	gesetzt	Reset Jumper (siehe oben)
K5	auf 8 MHz gesetzt	Wahl der Taktfrequenz 4MHz oder 8MHz
K10	gesetzt	Wenn Jumper gesetzt = LEDs aktiv
K11	alle gesetzt	Schalter und Taster mit Port A und mit Pull Down Widerständen verbunden (siehe oben)
K14	gesetzt (zeigt von der Platine weg)	Wahl zwischen interner und externer Referenzspannung

Inbetriebnahme

- Setzen der Jumper wie oben beschreiben.
- Verbinden des Controllerboards mit der Parallelen Schnittstelle des PCs, mit Hilfe eines SUB-D-Verlängerungskabels (25 Pol; jeweils ein Stecker und eine Buchse).
- ICs in die Fassungen setzen.
- Anschluss der Versorgungsspannung an die Schraubklemme
Polung beachten!
- Schalter S1 einschalten
Die Grüne Kontroll-LED sollte jetzt leuchten. Sie zeigt an, dass die Spannungsquelle richtig angeschlossen wurde und dass die Platine jetzt mit Spannung versorgt ist.

Die Platine ist jetzt **betriebsbereit** und muss nun nur noch programmiert werden.

Software zum Programmieren des Controllers

Für die ersten Programmierversuche der AVR Platine eignet sich besonders gut die Entwicklungsumgebung **BASCOM**.

Wie der Name schon verrät wird hier in Basic programmiert. Natürlich kann man den Controller auch in anderen gängigen Programmiersprachen programmieren. Zum Beispiel findet man auf der Webseite des Controllerherstellers (www.atmel.com) einen Assembler und einen ISP Programmer, die von der AVR Platine unterstützt werden.

Einen C Compiler kann man sich ebenfalls mit Hilfe des Internets beschaffen.

Die AVR Platine ist mit dem bekannten **Starterkit STK200 kompatibel**, d.h. alle Programmer, die das STK200 Starterkit unterstützen können auch die AVR Platine programmieren.

Programmieren des Controllers mit BASCOM

BASCOM ist eine komplette Entwicklungsumgebung zum Programmieren und zum Debuggen der AVR Controller von Atmel.

Es ist eine freie Demoversion und das Handbuch auf der Seite:

http://www.mcselec.com/download_avr.htm verfügbar, lediglich der Programmcode wurde auf 2 KB beschränkt. Dies ist allerdings für kleinere Projekte vollkommen ausreichend.

Mit BASCOM-AVR lassen sich alle AVR-Typen ab AT90S2313 bearbeiten. Allein die kleinen tinyAVRs und der AT90S1200 sind nicht enthalten, da diese Bausteine keinen bzw. zu wenig RAM aufweisen.

In der Software findet man viele nützliche Tools und Ideen, so z.B. einen LCD-Designer, mit dem man Zeichensätze für (die üblichen HD44780-basierten) Alfanumerischen LC-Displays auf einfache, grafische Weise erstellen kann.

Der Befehlsumfang des Compilers beschränkt sich nicht nur auf die zahlreichen BASIC-Standardkommandos, sondern bietet darüber hinaus auch spezielle Anweisungen, z.B. für LCD, I2C-Bus und 1-Wire-Bus (Dallas).

Features

- Strukturiertes Basic mit Labels
- Editor mit Syntax-Highlighting
- IF/THEN/ELSE, Schleifen- und CASE-Konstrukte

- Bezeichner mit bis zu 32 signifikanten Zeichen
- Bit, Byte, Integer, Word, Long, and String Variablen (demnächst auch Float)
- Integrierter Terminal Emulator mit Downloadoption
- Unterstützt SPI-Programmierung (z.B. über AVR-Starterkits)
- Kontextsensitives Hilfesystem

Bücher zum Thema BASCOM

Mittlerweile sind zwei deutschsprachige Bücher von Claus Kühnel erschienen.

- Programmieren der AT90S23x3 mit BASCOM-AVR
Claus Kühnel
228 Seiten
ISBN 3-907857-00-3
Preis DM 29,-

Dieses Buch ist eine Bearbeitung des [Buches zu BASCOM-AVR](#) (ISBN: 3-89811-937-8) speziell zugeschnitten auf die AT90S23x3 mit einem Codebereich von 2 KB. Diese Mikrocontroller lassen sich mit der gratis downloadbaren BASCOM-AVR Demo ohne Einschränkungen programmieren

- [Programmieren der AVR RISC Mikrocontroller mit BASCOM-AVR](#)
Claus Kühnel
250 Seiten
ISBN: 3-89811-937-8
Preis: DM 49,00 EUR 25,05

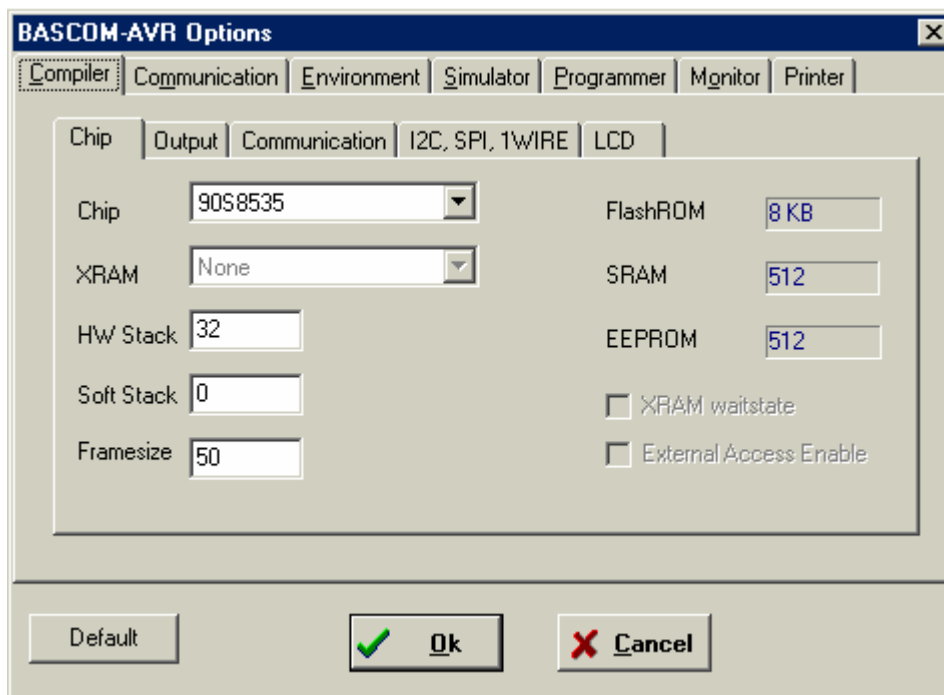
Dieses Buch behandelt die Programmierung von Mikrocontrollern. Beschrieben wird die Arbeit mit der komfortablen Entwicklungsumgebung BASCOM-AVR.

Weiter Information gibt es auf der Webseite von [Claus Kühnel: http://www.ckuehnel.ch/](http://www.ckuehnel.ch/).

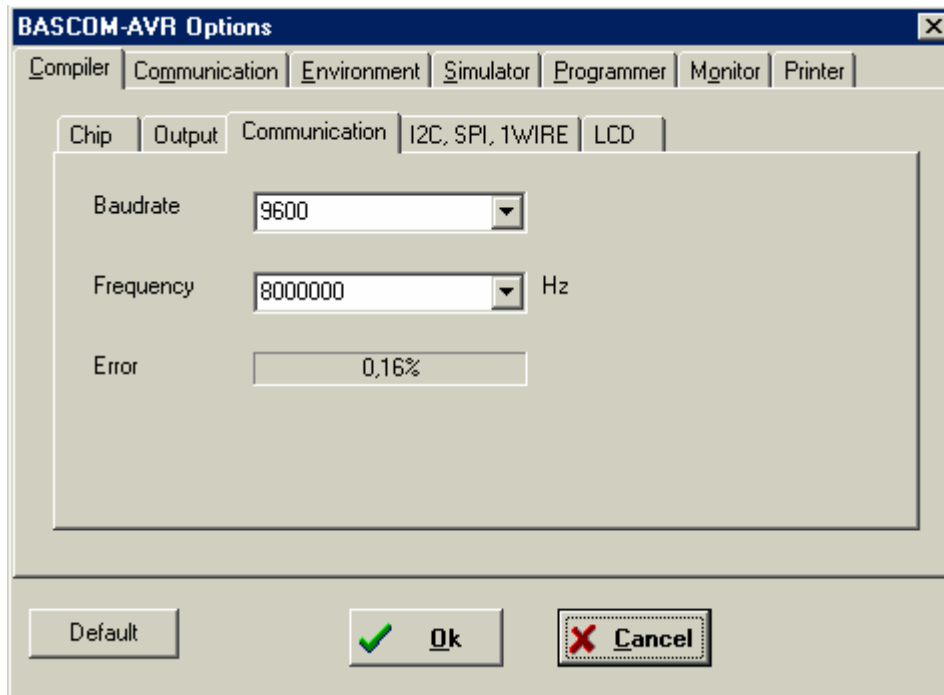
Einstellungen in BASCOM

Vor dem Programmieren des Controllers muss man in BASCOM folgendes einstellen:

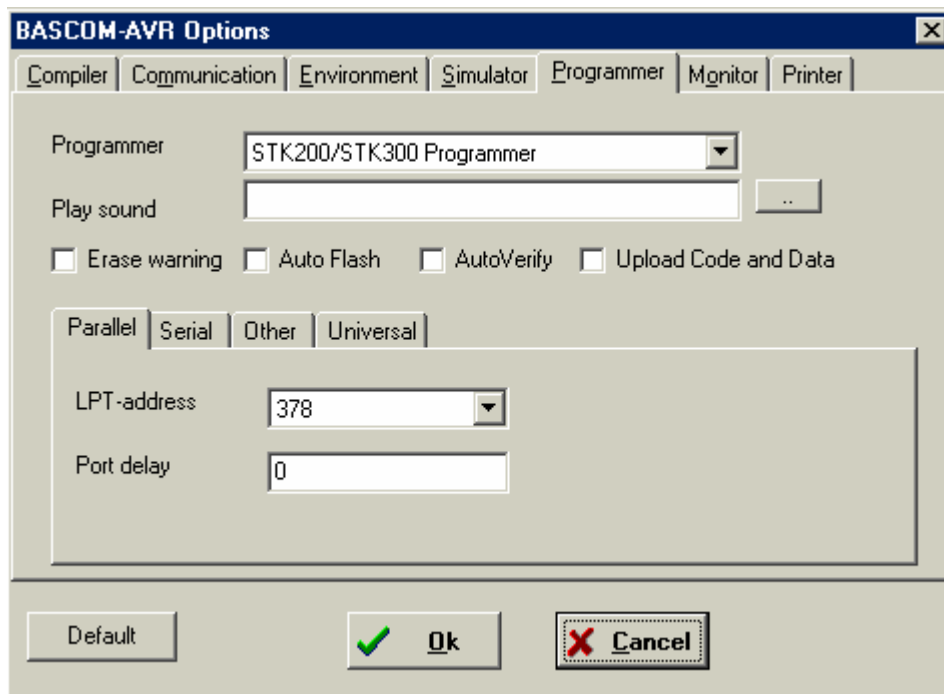
- Auswahl des Controllers (90S8535) im Menü **Options/Compiler/Chip**:



- Taktfrequenz (8 MHz, je nach Jumpereinstellung) und die Baudrate für die serielle Schnittstelle im Menü **Options/Compiler/Communication** einstellen:



- Programmer (Das Controllerboard ist mit dem STK 200 kompatibel) und Wahl des Druckerports (LPT-Adresse) an dem die AVR-Platine mit dem PC verbunden ist im Menü **Options/Compiler/Programmer**:



- Fertig mit den Einstellungen in BASCOM!

Endlich den Controller Programmieren

Wurden alle genannten Einstellungen an der AVR Platine und in BASCOM vorgenommen kann man endlich den AVR Controller Programmieren.

Dazu erstellt man im Menü **File/New** eine neue Datei und fügt den folgenden oder seinen eigenen Quellcode ein:

```
'LED Lauflicht im Dualcode:
'*****

Config Portc = Output           'Portc wird für die Ausgabe benutzt

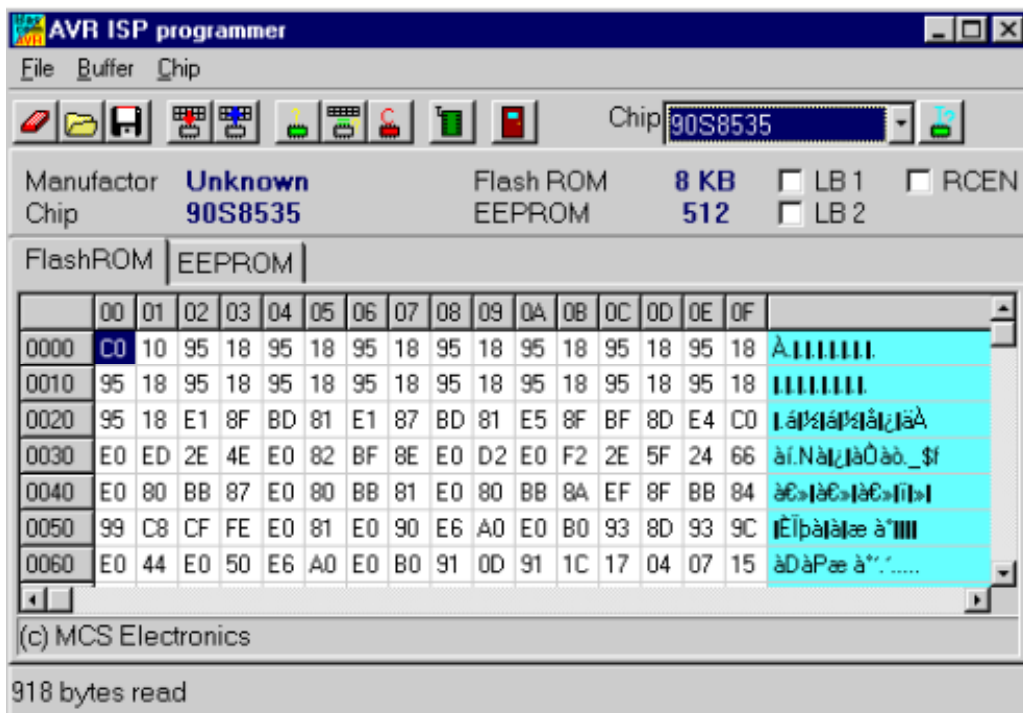
Dim X As Byte                  'Variable wird Deklariert

Losgehts:
  For X = 0 To 255              'Zählt von 0 bis 255
    Portc = X                  'Ausgabe an den LEDs
    Waitms 200                 'Wartet ca. 200 ms
  Next                          'Schleife beendet wenn x = 255
  Goto Losgehts                'Erneut For-Schleife ausführen

End
```

Danach kompiliert man den Basic-Quelltext in dem man im Menü **Programm** auf **Compile** klickt oder die Taste **F7** drückt. Dadurch wird ein Intel-Hex-File erzeugt mit dem man schließlich den Controller programmieren kann.

Nun drückt man die Taste **F4** und der AVR-Programmer erscheint.



Wurde die AVR-Platine korrekt aufgebaut und alle genannten Einstellungen an der AVR Platine und in BASCOM vorgenommen, müsste der Programmer den AVR-Contoller automatisch erkennen. Wie in der obigen Abbildung muss dann als Chip der 90S8535 erkannt werden.

Dann im Menü **Chip** den Button **Autoprogram** betätigen.

Danach ist der Controller programmiert.

Nun müsste das übertragene Programm **automatisch starten** und die LEDs im Dualcode aufleuchten, sofern der LED Jumper K10 gesetzt wurde.

Programmbeispiele für BASCOM

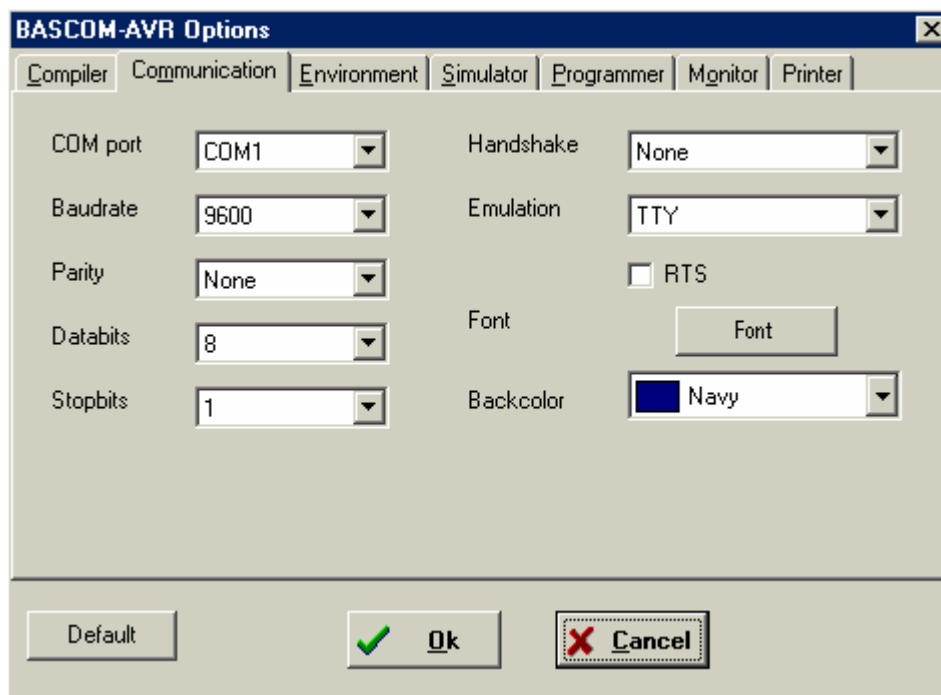
Kommunikation mit der seriellen Schnittstelle

Hierzu verbindet man die AVR-Platine (D-SUB9-Buchse K2) mit einem D-SUB9-Kabel (mit einem Stecker und mit einer Buchse) und mit der seriellen Schnittstelle des PCs.

Zuvor muss man noch ein paar Dinge im Menü **Options\Communication** von BASCOM einstellen:

- Die Baudrate auf den Standardwert von 9600 einstellen, mit welcher der Mikrokontroller mit dem Terminalprogramm kommunizieren soll.
- Den COM-Port des PCs wählen, an dem die AVR-Platine angeschlossen ist.

Der Menüpunkt **Communication** von BASCOM sollte so aussehen:



Danach erstellt man wieder ein neues Dokument in BASCOM und fügt den folgenden Quelltext ein:

```
'Ausgabe über die seriellen Schnittstelle:
'*****

Dim X As Byte           'Variable deklarieren

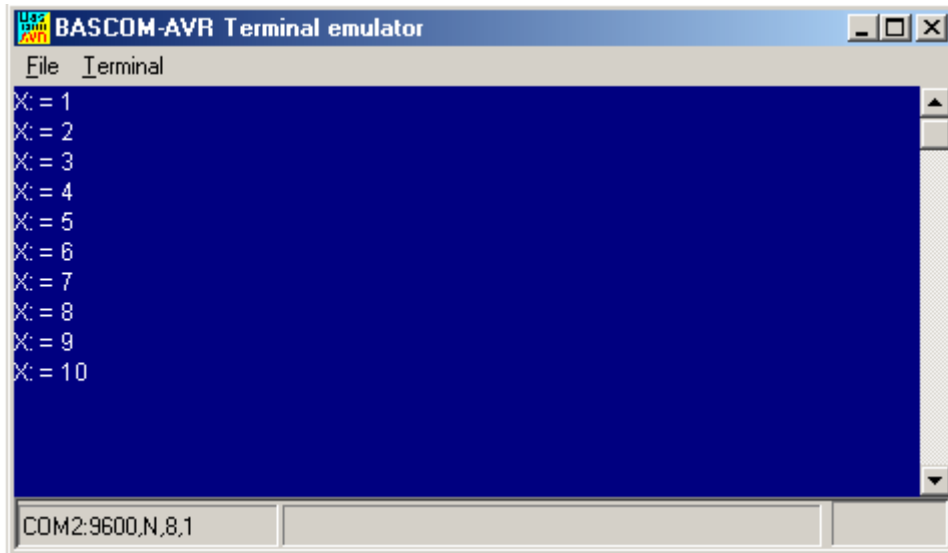
For X = 1 To 10         'Zählt von 1 bis 10
  Print "X: = " ; X     'Ausgabe von x über die serielle Schnittstelle
  Wait 1               'Wartet eine Sekunde
Next                    'Schleife beendet wenn x = 10

End
```

Natürlich muss man den Quelltext wieder kompilieren und den Controller (wie oben beschrieben) programmieren.

Um das Programm zu testen öffnet man den Terminal-Emulator im Menü **Tools** von BASCOM.

Das Programm sollte nach dem Programmieren des Controllers und dem Drücken des Reset-Tasters automatisch starten und wie folgt aussehen:

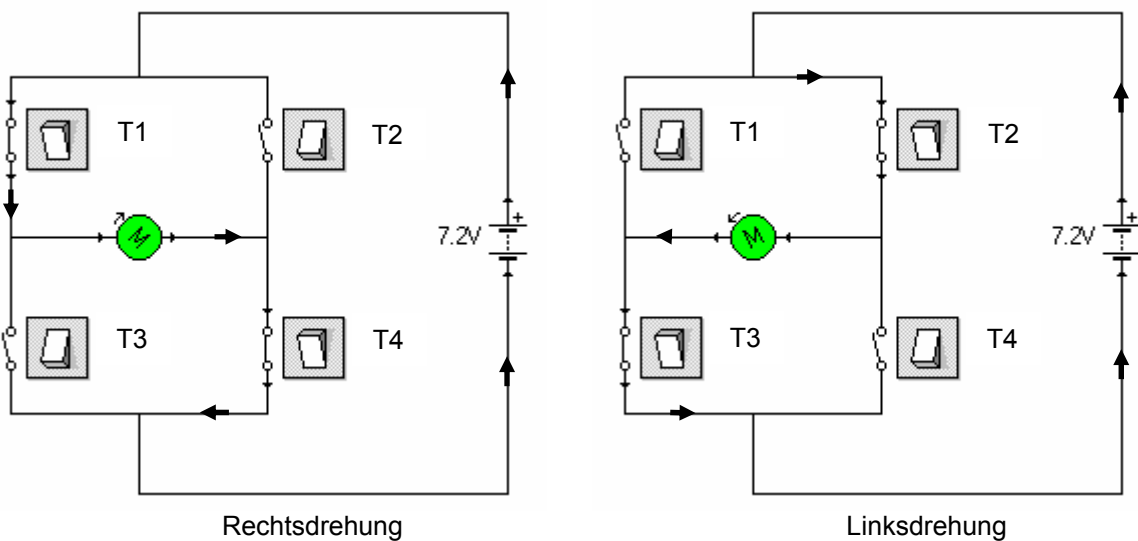


Ansteuerung des Brückentreibers

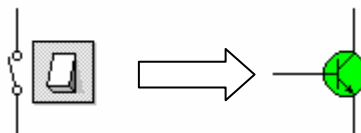
Aufgabe des Brückentreibers

Der Brückentreiber hat die Aufgabe die Drehrichtung der Motoren zu steuern. Jeder Motor muss sich in beide Richtungen drehen können. Beide Motoren benötigen deshalb eine Brückenschaltung, die über die Logiksignale des Mikrocontrollers gesteuert wird.

Prinzipieller Aufbau des Brückentreibers



Die vier Steuertransistoren wurden durch Schalter ersetzt, um die Funktion zu verdeutlichen.



Ansteuerung eines Motors

Je nach dem welche Transistoren angesteuert werden, wird die Drehrichtung des Motors bestimmt. Bei 0 sperrt der Transistor und bei 1 leitet er. Daraus ergibt sich die folgende Funktionstabelle:

T4	T3	T2	T1	Motor
0	X	0	0	Leerlauf
1	0	0	1	Rechtsdrehung
0	1	1	0	Linksdrehung
1	1	0	0	Bremsen
0	0	1	1	

A. Rechtslauf

T1 und T4 geschlossen. Dadurch fließt ein Strom von + 7,2 V über T1 von links nach rechts durch den Motor nach 0 V.

B. Linkslauf

T2 und T3 geschlossen. Dadurch fließt der Strom von + 7,2 V über T2 von rechts nach links durch den Motor nach 0 V, wodurch sich der Motor entgegengesetzt dreht.

C. Freilauf

Wird kein Transistor angesteuert läuft der Motor langsam aus.

D. Bremsen

Werden die beiden oberen oder die beiden unteren Schalter geschlossen, so wird die vom Motor erzeugte Spannung kurzgeschlossen. Dadurch stoppt der Motor ohne auszulaufen.

Auf keinen Fall dürfen T1 und T3 oder T2 und T4 gleichzeitig geschlossen werden, sonst entsteht ein Kurzschluss. Dies wird aber durch die interne Logik des LD293D verhindert.

Funktionstabelle

Die Funktionstabelle gibt Übersicht über die Beschaltung des Brückentreibers mit dem Controller der AVR-Platine (siehe auch Seite 7). Die Zeichen in Klammern entsprechen den einzelnen Ports des Controllers:

IN1 (D3)	IN2 (D4)	Freigabe (D7)	Motor 1
X	X	0	aus
0	0	1	bremsen
0	1	1	links
1	0	1	rechts
1	1	1	bremsen

IN1 (D5)	IN2 (D6)	Freigabe (D7)	Motor 2
X	X	0	aus
0	0	1	bremsen
0	1	1	links
1	0	1	rechts
1	1	1	bremsen

Die Brückentreiberausgänge können einen maximalen Dauerstrom von ca. 600 mA liefern. Die Spannung an den Brückentreiberausgängen entspricht der Versorgungsspannung, die an der Schraubklemme K3 anliegt.

Programmbeispiel

```
'Funktionstest des Brückentreibers (Motor 1):
'*****

Config Portd = Output           'Portd wird für die Ausgabe benutzt
Portd.7 = 1                     'Brückentreiber aktivieren

Links:
Portd.3 = 1
Portd.4 = 0
Wait 5                          '5 Sekunden warten

Rechts:
Portd.3 = 0
Portd.4 = 1
Wait 5                          '5 Sekunden warten

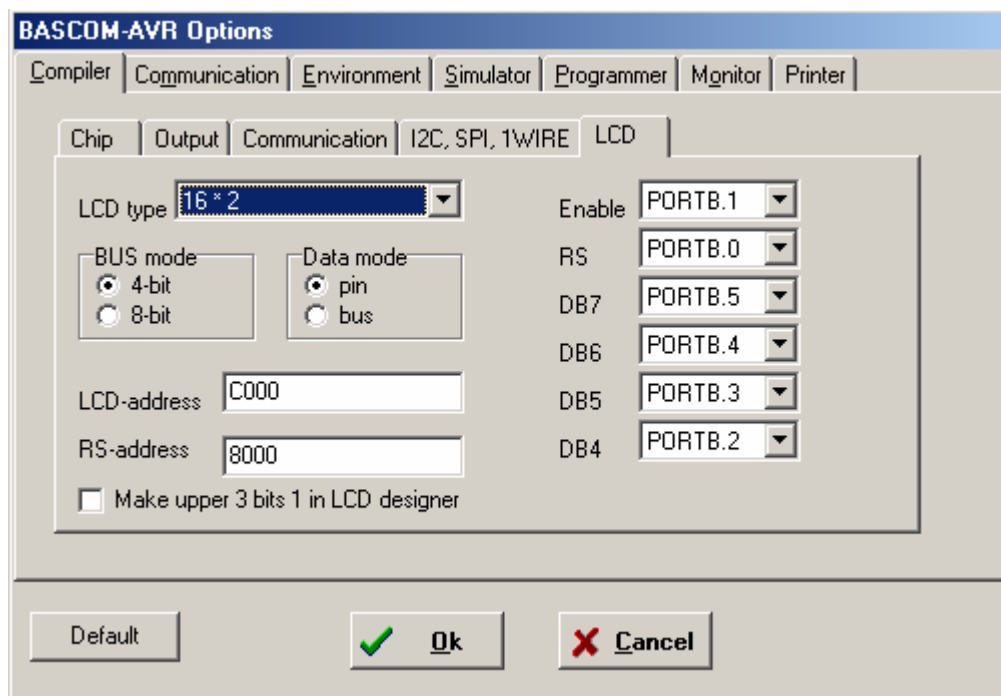
End
```

Anschluss eines LCD-Displays

Das LCD-Display wird im 4-Bit Pin-Modus betrieben. Dadurch werden die Datenleitungen DB0 bis DB3 eingespart.

Bevor man ein LCD-Display mit BASCOM ansteuern kann, müssen erst wieder ein paar Einstellungen im Menü **Options/Compiler/LCD** vorgenommen werden.

Als erstes wählt man das LCD-Display aus. Hier wurde ein Display mit 16 Zeichen pro Zeile mit zwei Zeilen ausgewählt. Nun nimmt man die Einstellung für die AVR Platine wie folgt vor:



Die meisten LCD-Displays verwenden den Controller **HD44780** oder einen kompatiblen (z.B. KS0070) und haben 14 oder 16 Pins. Die Pinbelegung ist meist immer identisch:

Pinbelegung und Anschluss an die Platine

LCD-Display			AVR Platine Stecker K17
Pin:	Name:	Funktion:	Anschluss (siehe Seite 7):
1	V _{SS}	Power Supply (GND)	GND
2	V _{DD}	Power Supply für Logik (+5V)	+5V
3	V _O	Kontrasteinstellung (Poti P2)	V _O
4	RS	Register select Signal	Port B0
5	R/W	Lese/Schreibe Signal	GND
6	E	Freigabe Signal	Port B1
7	DB0	Datenleitung	Nicht angeschlossen
8	DB1	Datenleitung	Nicht angeschlossen
9	DB2	Datenleitung	Nicht angeschlossen
10	DB3	Datenleitung	Nicht angeschlossen
11	DB4	Datenleitung	Port B2
12	DB5	Datenleitung	Port B3
13	DB6	Datenleitung	Port B4
14	DB7	Datenleitung	Port B5
15	A _{LED}	LED-Beleuchtung Anode (+)	Kein Anschluss vorgesehen. Wird nur gebraucht für die Hindergrundbeleuchtung
16	A _{LED}	LED-Beleuchtung Kathode (-)	

Programmbeispiel

```
'Ansteuerung eines LCD-Displays
```

```
!*****
```

```
Dim X As Byte
```

```
Cls                                'Löscht das Display
Lcd "Hallo"                        'Ausgabe
Wait 1
Lowerline                          'Untere Zeile
Lcd "Welt"                          'Ausgabe
Wait 1

Cls                                'Löscht das Display
Lcd "Hallo Welt"
Wait 1
For X = 1 To 16
    Shiftlcd Right                  'Schiebt den Text nach rechts
    Wait 1
Next

Cls
Home                                'Kursor auf Anfang setzen
LCD "Ende"

End
```

Nun verdreht man das Poti P2 (Kontrast) solange, bis man auf dem LCD Display etwas erkennen kann. Wenn auf dem LED-Display kein Text erscheint, sollte man die Einstellungen in BASCOM noch mal überprüfen.

Schaltplan

