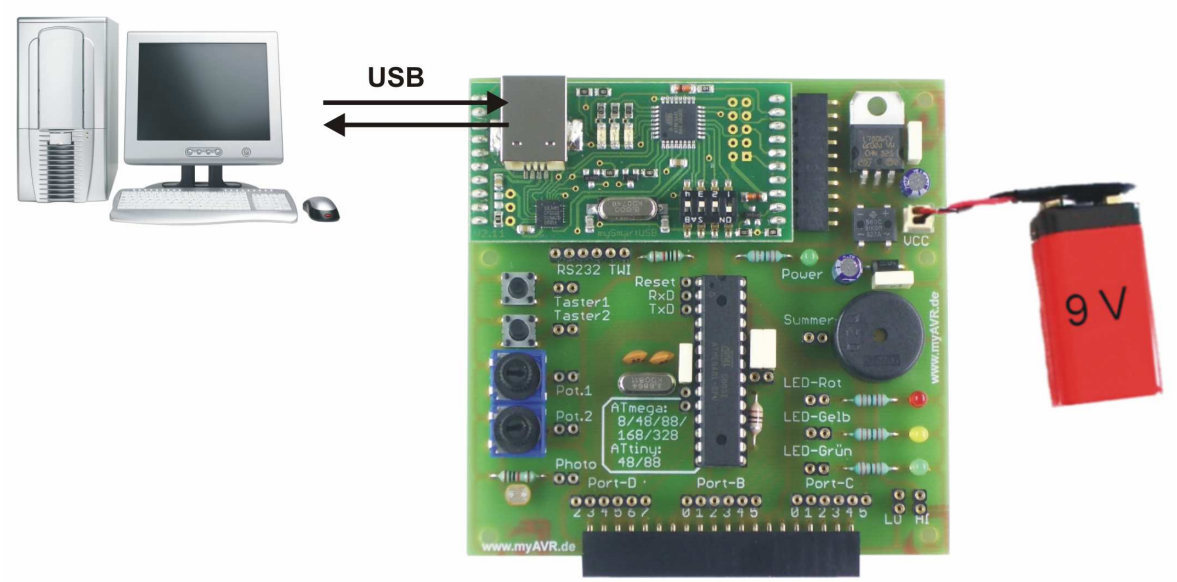
Stefan Hoffmann

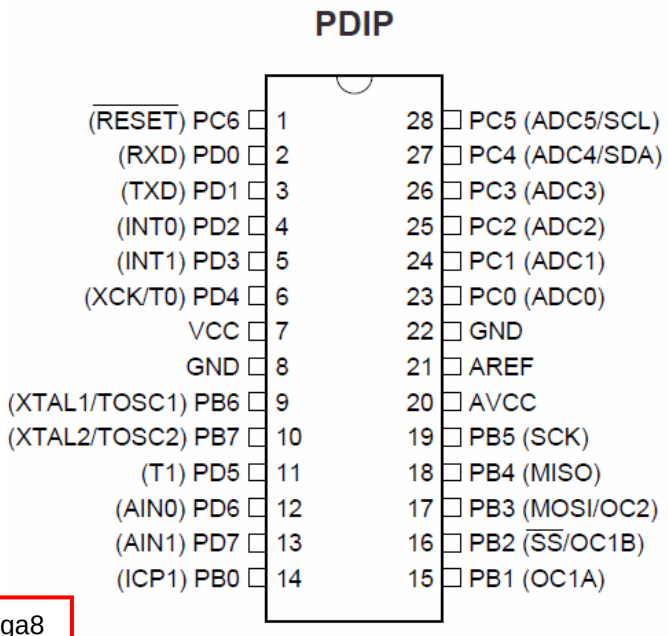
Leichter Start mit BASCOM und myAVR

1 Entwicklungsumgebung für AVR Mikrocontroller

Für die Arbeit mit BASCOM und myAVR-Produkten benötigt man einen PC mit Windows-Betriebssystem und einer BASCOM-Installation (siehe unten), ein myAVR Board MK2 mit einem USB-Programmer sowie ein USB-Kabel und eine externe Spannungsversorgung (9V Batterie oder geregeltes 9V Netztteil). Alles außer dem PC wird von myAVR als Komplettset günstig und in hochwertiger Ausführung angeboten.



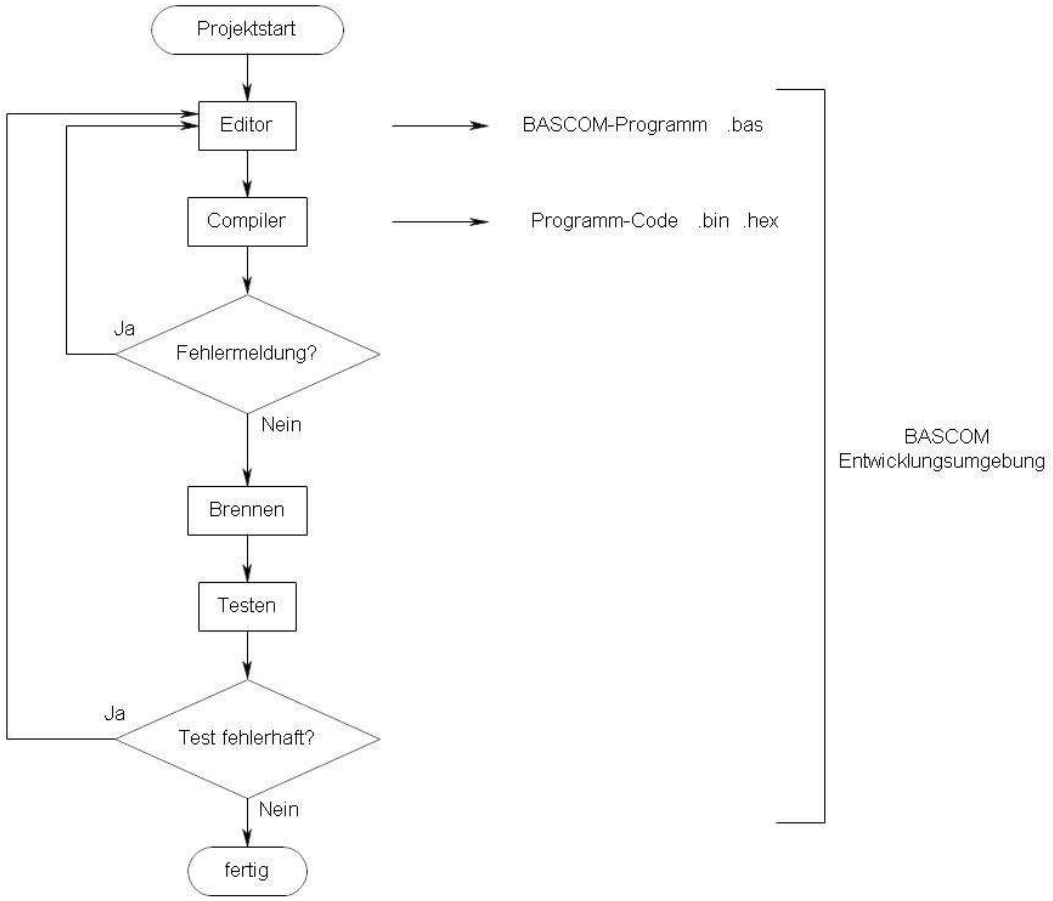
Auf dem Board befinden sich außer dem Programmer und einer Fassung für den Ziel-Mikrocontroller vom Typ ATmega8 auch diverse Ein-/Ausgabe-Komponenten:
- 3 verschiedenfarbige Leuchtdioden (LEDs) (Kathode gegen Masse geschaltet.)
- 2 Taster (gegen Masse tastend)
- 1 Piezo-Schallgeber
- 2 Potentiomenter (als Spannungsteiler geschaltet)
- 1 Lichtsensor (als Spannungsteiler geschaltet)



Pinbelegung ATmega8

2 Die Vorgehensweise

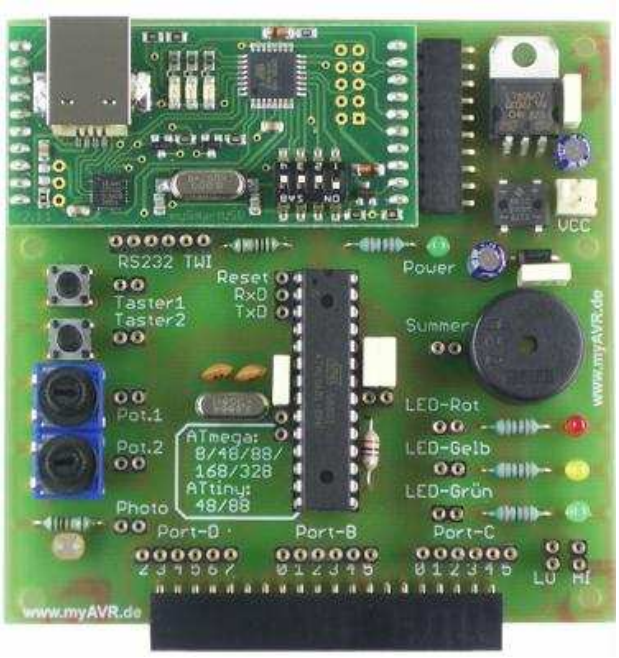
Der Ablauf des Programmierens bestehend aus Editieren – Kompilieren – Brennen – Testen sieht folgendermaßen aus:

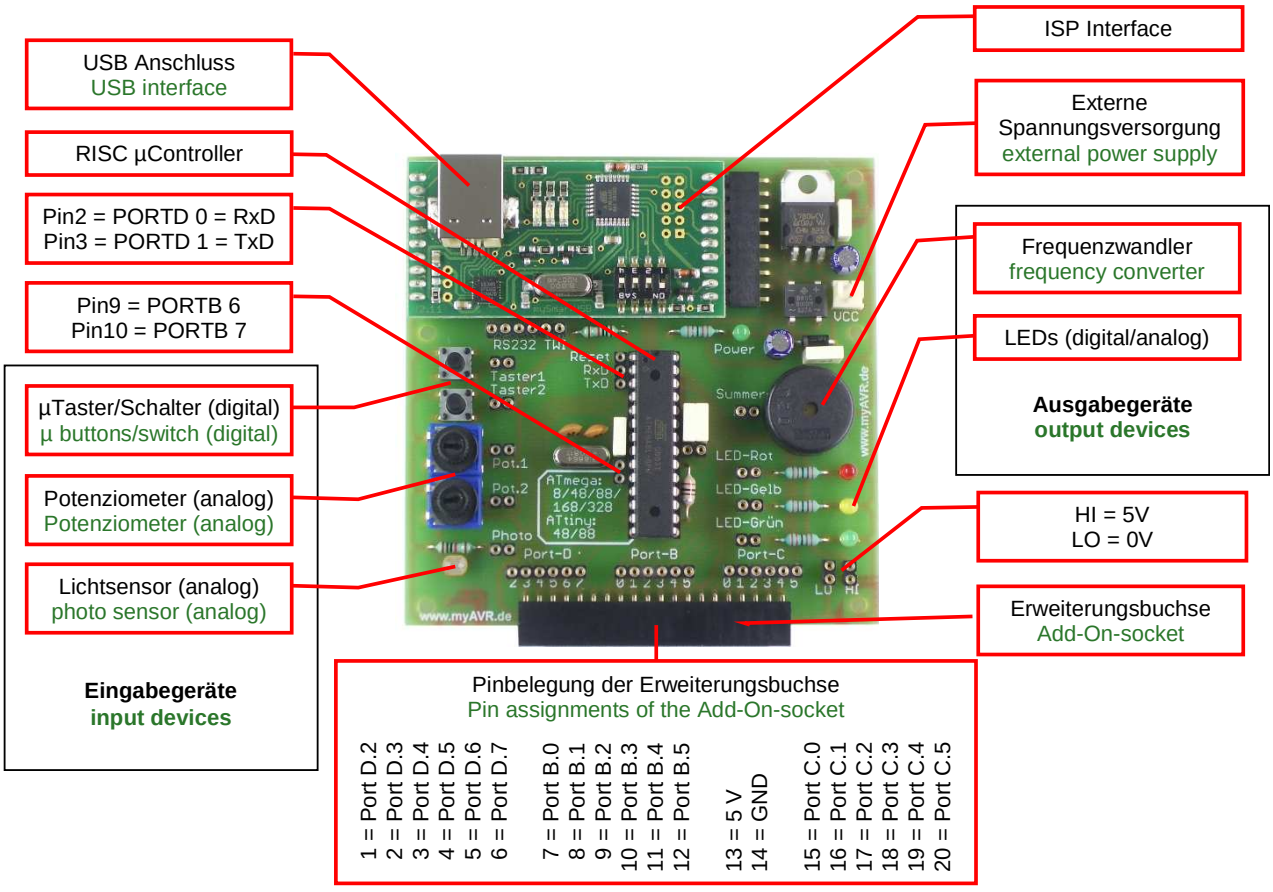


Zunächst wird ein BASCOM-Programm als Text im BASCOM-Editor eingegeben oder ein vorhandenes BASCOM-Programm in den Editor geladen. Der BASCOM-Editor stellt die verschiedenen BASCOM-Sprachkonstrukte in unterschiedlichen Farben dar. Bei Eingaben im BASCOM-Editor gibt es sehr viele kontextsensitive Vorschläge und durch Drücken der F1-Taste erhält man kontextsensitive ausführliche BASCOM-Hilfe.
Als nächstes wird das im Editor befindliche Programm kompiliert, d.h. geprüft und in Maschinensprache übersetzt.
Wurde dieser Schritt fehlerfrei abgeschlossen, dann wird das Programm auf den Mikrocontroller übertragen – "gebrannt".

3 Die Hardware

Experimentierboard myAVR Board MK2 für USB-Anschluss:





Bevor das Board an den PC angeschlossen wird, muss ein USB-Treiber auf dem PC installiert werden. Die Treiber stehen zum Download bereit von
www.myAVR.de → online-Shop → Download →
Suchbegriff „Treiber“ oder „dl46“.
Eine technische Beschreibung zum mySmartUSB MK2 findet man als Download mit dem Suchbegriff „MK2“ oder „dl48“, in dieser ist ab Seite 9 die Treiberinstallation auch noch einmal genau beschrieben.
Die Treiber und Beschreibungen sind auch auf der Ressource CD enthalten.
Beschreibungen findet man unter „Dokumente“ und die Treiber unter „Tools“.

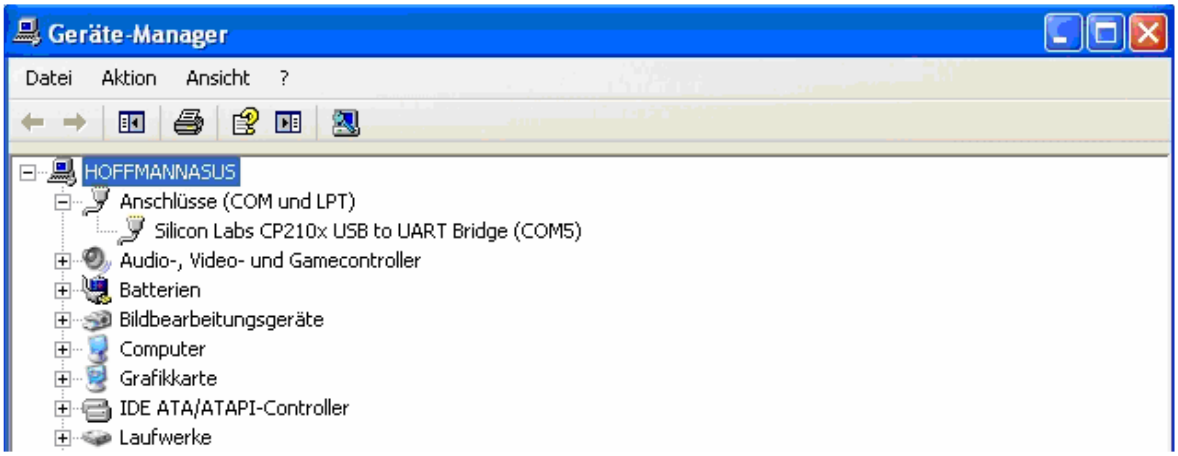
- Treiber downloaden (Windows 7 Version bzw. Version für sonstige Windows-Versionen)
- Dateien extrahieren
- Installations-exe ausführen
- Jetzt kann das Board mit dem USB-Kabel an den PC angeschlossen werden und wird erkannt.

Eine sehr schöne und sinnvolle Ergänzung ist ein
- LCD Add-On von myAVR

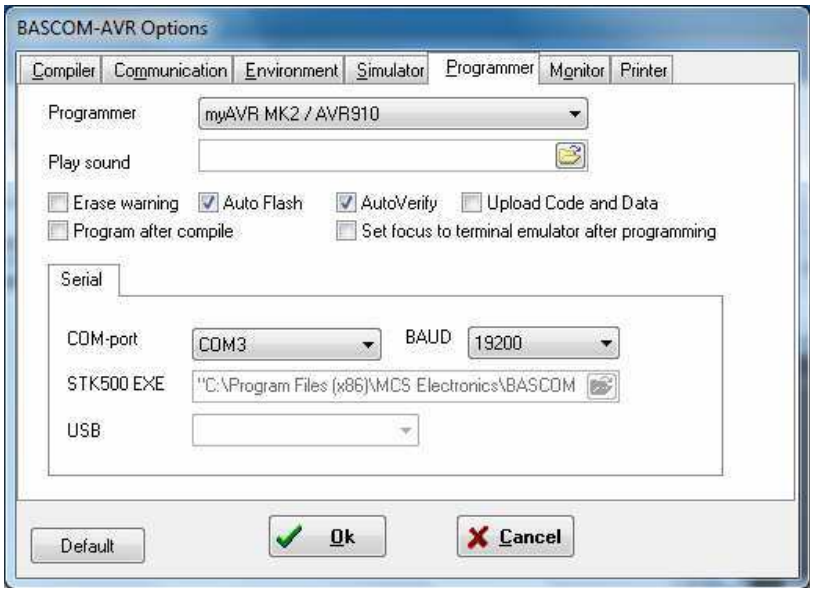
4 Die Software

An Software wird außer einem PC mit Windows nur die Entwicklungsumgebung BASCOM-AVR benötigt, die es als uneingeschränkte Vollversion im myAVR-Shop gibt. Eine Demoversion von BASCOM finden Sie auf: http://www.mcselec.com unter *Downloads → BASCOM → BASCOM-AVR → BASCOM-AVR Demo*.
Dabei sind die Lizenzbedingungen zu beachten und die Programmgröße ist bei der Demoversion auf 4K Bytes begrenzt.

Nach der Installation von BASCOM muss der Programmer eingestellt werden. Dafür wird der Name des virtuellen COM-Ports im Windows-Gerätemanager nachgesehen: (hier als Beispiel COM5)



In den Optionen wird nun der COM-Port und der Programmer eingestellt. Die Sprache der Benutzeroberfläche kann hier auch geändert werden, z.B. Deutsch.



5 BASCOM-Beispiele

5.1 Das erste BASCOM-Programm: LED blinkt

Als erstes Programm wird meistens ein Programm erstellt, welches eine Leuchtdiode blinken lässt.
Das Programm ist einfach, verständlich und man kann durch dieses kleine Programm den Prozess der Entwicklung komplett durchlaufen.

Das BASCOM-Programm beginnt mit einem Block aus Kommentaren, in denen das Programm beschrieben wird.
Es folgt ein Deklarationsteil (Prozessor, Taktgeschwindigkeit, Speicheraufteilung).

Bei Mikrocontroller-Programmen befindet sich der Haupt-Teil meistens in einer Endlos-Schleife (DO..LOOP).
In der Schleife wird die LED ein- und ausgeschaltet.

```
'-----------------------------------------------------------------------------
'Programmname:          Programm1_LED_blinkt.bas
'Funktion:              eine LED blinkt
'Mikrocontroller:       Mega8
'Input:                 -
'Output:                LED an Port B.0
'-----------------------------------------------------------------------------

$regfile = "m8def.dat"                  ' eingesetzter Mikrocontroller
$crystal = 1000000                      ' eingestellte Taktfrequenz
$hwstack = 40
$swstack = 32
$framesize = 60


Config Portb = Output                   'gesamten Port B als Ausgabeport

Do                                      'Schleifenanfang
  Portb.0 = 1                           'LED an
  Wait 1                                '1 Sekunde warten
  Portb.0 = 0                           'LED aus
  Wait 1                                '1 Sekunde warten
Loop                                    'Schleifenende

End                                     'Programmende
```

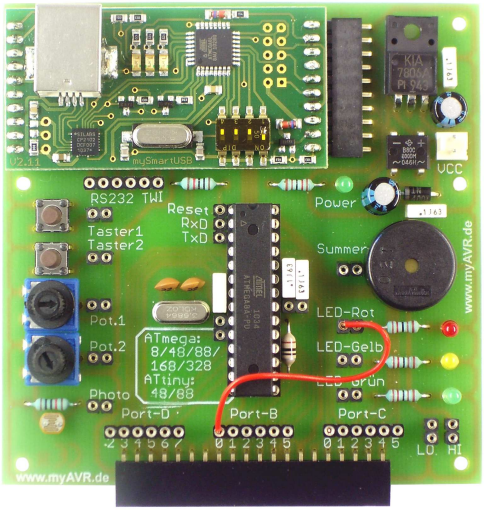
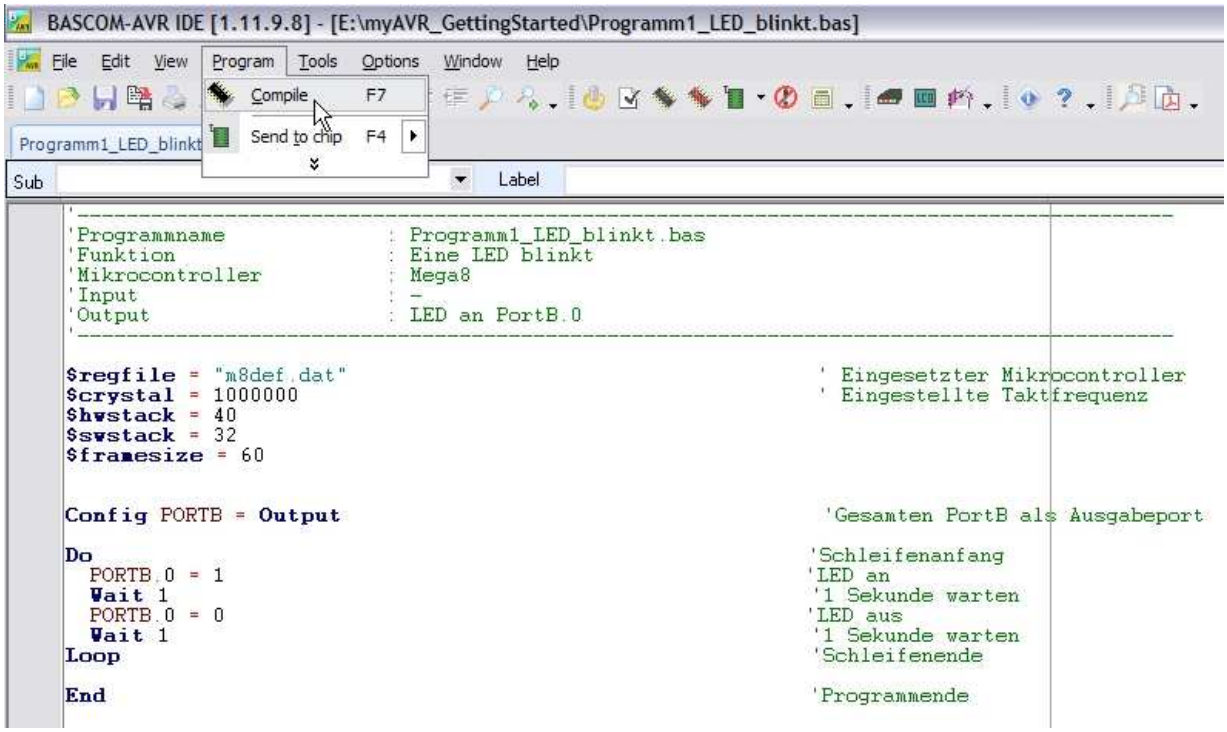
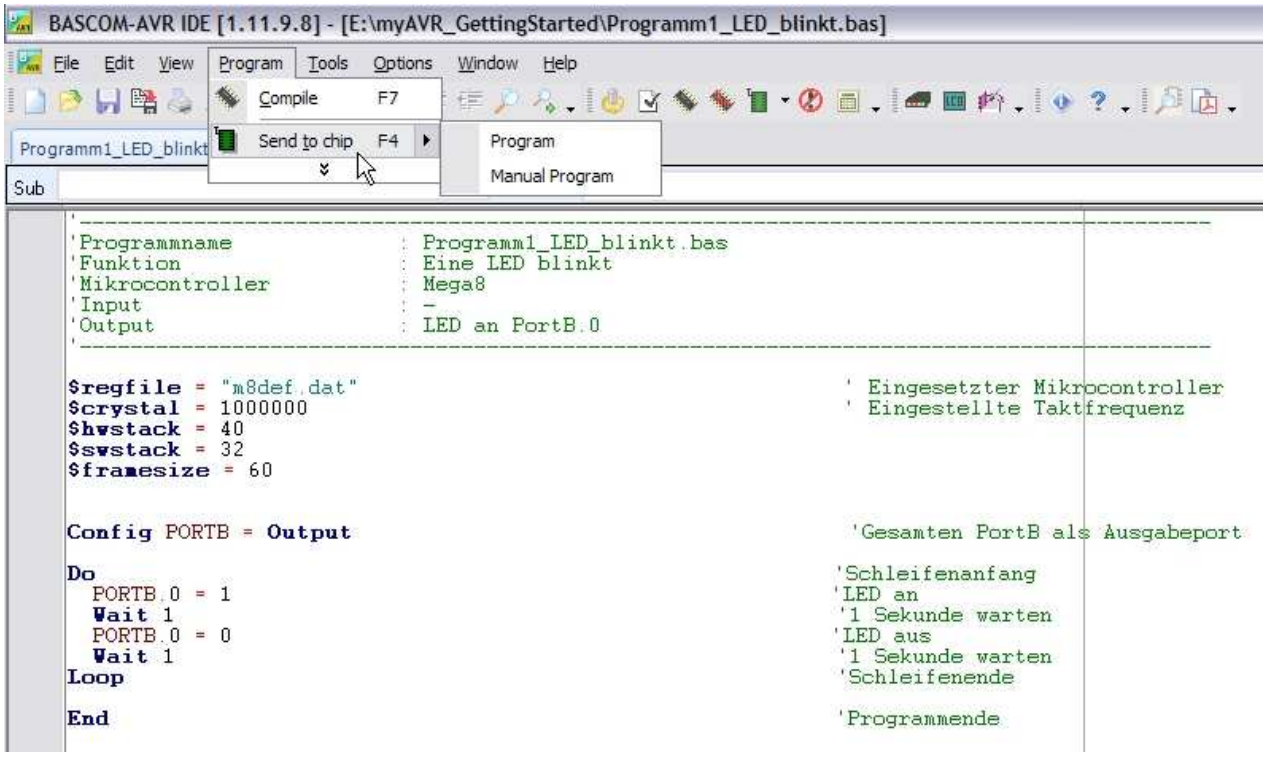


Abbildung:
Verdrahtung auf dem „myAVR Board MK2 “ für „LED blinkt“

Wenn das Programm im Editor komplett ist, dann wird das Programm kompiliert:



Nach fehlerfreiem Kompilieren wird das Programm auf den Mikrocontroller übertragen:



Nach erfolgreichem Brennen startet sofort das Programm auf dem Mikrocontroller. Es kann getestet werden, ob das BASCOM-Programm die gewünschte Funktionalität hat oder ob Modifikationen nötig sind.

5.2 Ampel

Wenn durch das Blinkprogramm einmal die Umgebung erfolgreich getestet wurde, kann man das Programm leicht modifizieren – zum Beispiel ein kleines Ampelprogramm mit 3 LEDs schreiben:

```
'-----------------------------------------------------------------------------
'Programmname:          Programm2_Ampel.bas
'Funktion:              einfache Ampel
'Mikrocontroller:       Mega8
'Input                  :-
'Output                 : LEDs an Port B.0 (rot), Port B.1 (gelb), Port B.2 (grün)
'-----------------------------------------------------------------------------

$regfile = "m8def.dat"                      ' eingesetzter Mikrocontroller
$crystal = 1000000                          ' eingestellte Taktfrequenz
$hwstack = 40
$swstack = 32
$framesize = 60

Config Portb = Output                       'gesamten Port B als Ausgabeport

Rote_led Alias Portb.0                      'Aliasnamen vergeben
Gelbe_led Alias Portb.1
Gruene_led Alias Portb.2


Do
  Rote_led = 1                              'rote LED an
  Wait 3                                    '3 Sekunde warten
  Gelbe_led = 1                             'gelbe LED an
  Wait 1                                    '
  Rote_led = 0 : Gelbe_led = 0              'rot und gelb aus
  Gruene_led = 1                            'grün an
  Wait 3
  Gruene_led = 0                            'grün aus
  Gelbe_led = 1                             'gelb an
  Wait 1
  Gelbe_led = 0                             'gelb aus
Loop

End
```

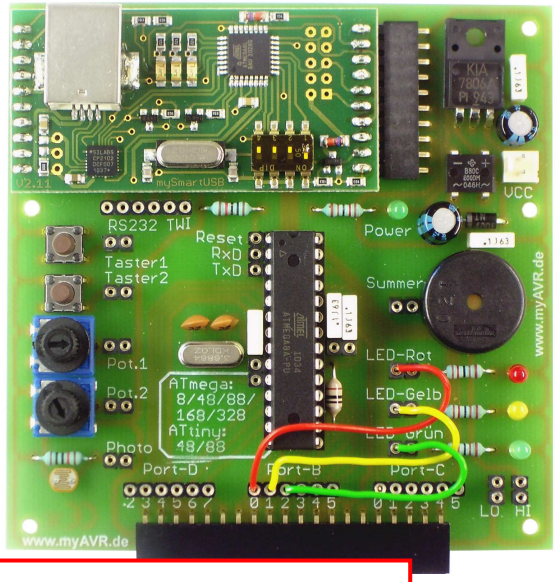


Abbildung:
Verdrahtung auf dem „myAVR Board MK2 “ für „Ampel“

5.3 Tastertest

In diesem Beispiel wird eine Eingabe durch Taster abgefragt:

```
'-----------------------------------------------------------------------------
'Programmname:          Programm3_Tastentest.bas
'Funktion:              Reaktion auf Tasteneingaben
'Mikrocontroller:       Mega8
'Input:                 Tasten an Port D.2 und Port D.3
'Output:                LEDs an Port B.0 (rot), Port B.1 (gelb), Port B.2 (grün)
                        Piezo_Speaker an Port B.3
'-----------------------------------------------------------------------------

$regfile = "m8def.dat"                  ' eingesetzter Mikrocontroller
$crystal = 1000000                      ' eingestellte Taktfrequenz
$hwstack = 40
$swstack = 32
$framesize = 60

Config Portd.2 = Input                  'D.0 als Eingabepin
Config Portd.3 = Input                  'D.1 als Eingabepin
Portd.2 = 1                             'Pullup-Widerstände
Portd.3 = 1

Config Portb = Output                   'gesamten Port B als Ausgabeport

Taster1 Alias Pind.2                    'PIN für Eingabe   PORT für Ausgabe !!!
Taster2 Alias Pind.3
Rote_led Alias Portb.0
Gelbe_led Alias Portb.1
Gruene_led Alias Portb.2
Piezo_speaker Alias Portb.3

Dim Zaehler As Byte                     'Variable Zaehler deklarieren

For Zaehler = 1 To 3
  Gruene_led = 0 : Rote_led = 1         'rot an
  Wait 1
  Rote_led = 0 : Gelbe_led = 1          'gelb an
  Wait 1                                '
  Gelbe_led = 0 : Gruene_led = 1        'grün an
  Wait 1
Next Zaehler


Do

If Taster1 = 0 Then Sound Piezo_speaker , 220 , 189   'wenn Taster1 gedrückt, Ton erzeugen

If Taster2 = 0 Then
  Gruene_led = 1                                      'wenn Taster2 gedrückt, dann grün an
Else                                                  'sonst..
  Gruene_led = 0                                      'grün aus
End If

Loop

End
```

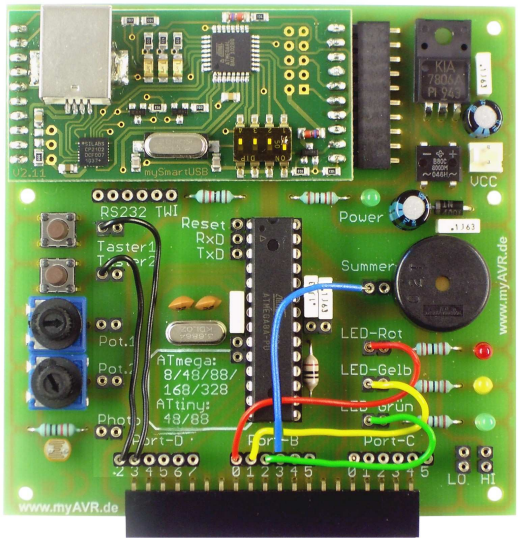


Abbildung:
Verdrahtung auf dem „myAVR Board MK2“
für „Tastertest“

5.4 Analoge Eingabe

Im Gegensatz zu den digitalen Eingaben durch Taster im vorigen Programm gibt es oft die Notwendigkeit, analoge Größen einzulesen.
Hier werden Potentiometer bzw. Lichtsensor-Eingaben ausgewertet:

```
'-----------------------------------------------------------------------------
'Programmname:          Programm4_AnalogeEingabe.bas
'Funktion:              Reaktion auf Analoge Eingabe
'Mikrocontroller:       Mega8
'Input:                 Potentiometer oder LDR an PortC.0
'Output:                LEDs an PortB.0 (rot), PortB.1 (gelb), PortB.2 (grün)
'-----------------------------------------------------------------------------

$regfile = "m8def.dat"                      ' eingesetzter Mikrocontroller
$crystal = 1000000                          ' eingestellte Taktfrequenz
$hwstack = 40
$swstack = 32
$framesize = 60

Config Portb = Output                       'gesamten Port B als Ausgabeport

Rote_led Alias Portb.0
Gelbe_led Alias Portb.1
Gruene_led Alias Portb.2

Dim Analogeingang As Word                   'Variable für Analogeingang deklarieren

Config Adc = Single , Prescaler = Auto , Reference = Avcc  'Analogwandler definieren

Do

Analogeingang = Getadc(0)                   'Analogeingang einlesen und Wert in Variable
                                            Analogeingang
Select Case Analogeingang                   'Je nach Inhalt der Variablen Analogeingang..
  Case 0 To 333:
    Rote_led = 0
    Gelbe_led = 0
    Gruene_led = 1                          'grüne LED an

  Case 334 To 666:
      Rote_led = 0
    Gelbe_led = 1                           'gelbe LED an
    Gruene_led = 0

  Case 667 To 1023:
      Rote_led = 1                          'rote LED an
    Gelbe_led = 0
    Gruene_led = 0

End Select

Loop

End
```

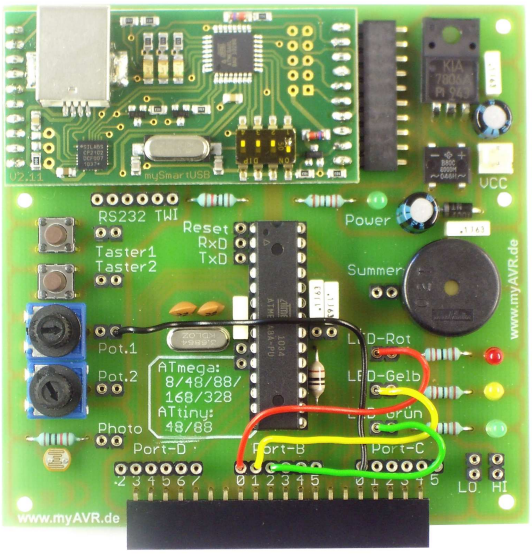


Abbildung:
Verdrahtung auf dem „myAVR Board MK2“
für „Analoge Eingabe“

5.5 LCD-Erweiterung

Sehr elegant lassen sich Textausgaben durch das myAVR LCD Add-On darstellen:

```
'-----------------------------------------------------------------------------
'Programmname:          Programm5_LCDisplay.bas
'Funktion:              Ausgabe auf LC-Display
'Mikrocontroller:       Mega8
'Input:                 Taster an Port D.0  (auch RxD-Pin)
'Output:                LC-Display an Port D.2 bis Port D.7
'-----------------------------------------------------------------------------

$regfile = "m8def.dat"                      ' eingesetzter Mikrocontroller
$crystal = 1000000                           ' eingestellte Taktfrequenz
$hwstack = 40
$swstack = 32
$framesize = 60

Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7 ,
 E = Portd.3 , Rs = Portd.2                 (Wichtig !!! Zeilenumbruch nicht im Editor,
                                            nur hier im Dokument)

Config Lcd = 16 * 2
Cls                                         'Displayinhalt löschen
Cursor Off                                  'Schreibmarke nicht anzeigen

Config Portd.0 = Input                      'D.0 als Eingabepin
Portd.0 = 1                                 'Pullup-Widerstand
Taster1 Alias Pind.0                        'PIN für Eingabe   PORT für Ausgabe !!!

Dim Variable As Byte                        'Laufvariable fuer Schleife

Do

Locate 1 , 1                                'Screibmarke positionieren'
If Taster1 = 0 Then                         'wenn Taster gedrückt..
  Lcd "Taster: -1      "                    'Text anzeigen
Else                                        'sonst..
  Lcd "Normal: +1      "                    'anderen Text
End If

Locate 2 , 1
If Taster1 = 1 Then                         'Wenn Taster nicht gedrückt..
  Incr Variable                             'hochzählen
Else                                        'sonst..
  Decr Variable                             'herunterzählen
End If

If Variable < 100 Then Lcd " "              'Führende Leerzeichen bei kleineren Zahlen
If Variable < 10 Then Lcd " "
Lcd Variable                                'Inhalt der Variablen anzeigen

Waitms 500                                  '500 Millisekunden = 1/2 Sekunde warten'
Loop

End
```

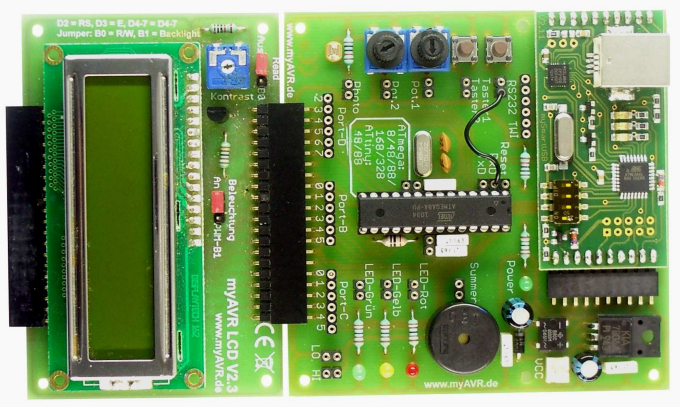


Abbildung:
Verdrahtung auf dem „myAVR Board MK2“ für „LCD-Erweiterung“

6 Literatur

Hoffmann, Stefan
Einfacher Einstieg in die Elektronik
mit AVR-Mikrocontroller und BASCOM
ISBN 978-3-8391-8430-1
Erhältlich im myAVR-Shop.