Dieser Abschnitt ist als Schnelleinstieg in die Programmierung von AVR-Mikrocontrollern mit BASCOM gedacht. Dabei werden Kenntnisse einer Programmiersprache wie BASIC, C/C++ oder PASCAL vorausgesetzt.
BASCOM ist kein BASIC-Interpreter wie der Name vielleicht vermuten lässt, sondern ein echter Compiler. Genau wie bei Assembler und C ist das Endprodukt der Programmierung mit BASCOM-AVR auch ein HEX-File, die per SPI auf den Controller gebrannt wird.
Die Beliebtheit von BASCOM resultiert daraus, dass die Sprache stark an BASIC angelehnt ist und der Einstieg durch mächtige und verständliche Befehle recht leicht gemacht wird. Leider verbirgt diese Hochsprache etwas die interne Struktur und Funktionsweise des Mikrocontrollers und so manche spezielle Aufgabenstellung lässt sich mit BASCOM dann doch nur schwer lösen.

Installation und Vorbereitung

Wo bekommt man BASCOM?
BASCOM-AVR© ist ein Produkt der Firma MCS Electronics in Holland.

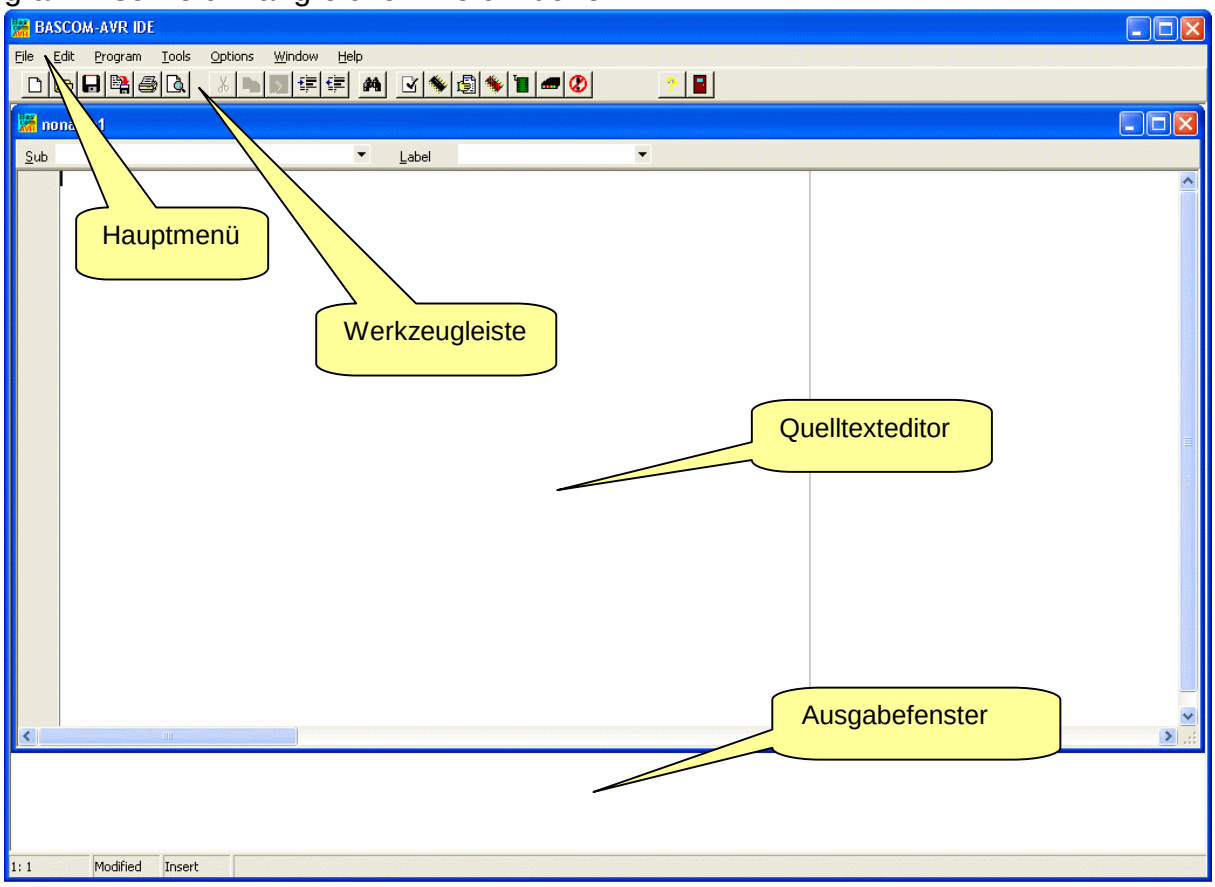
MCS Electronics
De Paal 112, 1351JJ
Almere-Haven
HOLLAND
http://www.mcselec.com/

Die Software ist lauffähig unter Windows 95, 98, NT, 2000 und XP. Der Preis für die Basisversion liegt bei 79,00 €.

Was ist bei der Installation zu beachten?
Die Installation wird mit *setup.exe* gestartet und ist unproblematisch. Bei der Demo ist zu beachten, dass das heruntergeladene ZIP-Archiv in einem temporären Verzeichnis gespeichert und entpackt werden muss.
Mit *setup.exe* wird BASCOM dann aus diesem Verzeichnis heraus installiert. Dieses Verzeichnis kann nach der Installation gelöscht werden.
Unter Windows NT, 2000 und XP ist darauf zu achten, dass der Installationsvorgang mit Administratorrechten durchgeführt werden muss. Die Deinstallation erfolgt über Systemsteuerung / Software entfernen.

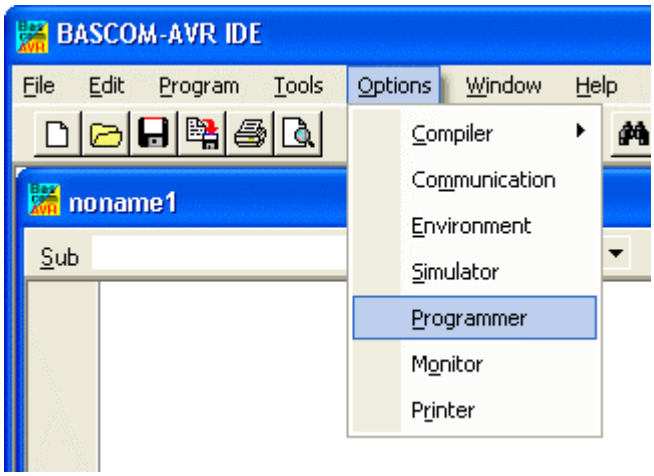
Die BASCOM Benutzeroberfläche.

Die Entwicklungsumgebung (IDE, Integrated Development Environment) besitzt einen Quelltexteditor mit Syntaxhervorhebung, integriertem Compiler und Brennprogramm sowie umfangreiche Hilfefunktionen.



Wie konfiguriert man BASCOM für das myAVR Board 1 LPT?

Bevor die Arbeit beginnen kann, sind noch ein paar Konfigurationsschritte nötig. Das myAVR Board 1 LPT verfügt über einen SP12-kompatiblen Parallelport-Programmer und eine serielle Schnittstelle. Damit die erstellten HEX-Dateien in den FLASH-Programmspeicher des Controllers gebrannt werden können, muss die Programmerkonfiguration eingestellt und das myAVR Board angeschlossen sein.



Die Option „Auto Flash“ erleichtert den Brennvorgang. Dabei werden die nötigen Schritte

- Programmer und Controller checken,
- FLASH löschen und
- FLASH schreiben

in einem Arbeitsgang durchgeführt. Für Fortgeschrittene stellt BASCOM noch den „Manuel Programmer“ bereit. Hier können zum Beispiel auch die FUSE-Bits manipuliert werden (äußerste Vorsicht!).

Die folgende Abbildung zeigt die korrekte Einstellung für das myAVR Board 1 LPT

Programmer : Universal MCS Interface, WinAVR and SP12

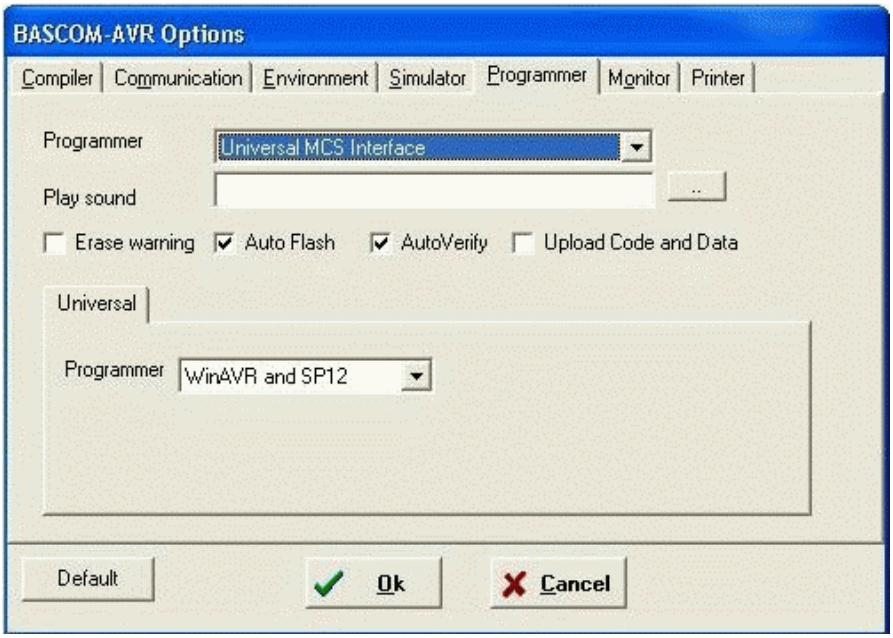


Abbildung: Programmer-Einstellungen für das myAVR Board 1 LPT

Über die serielle Schnittstelle kann mit dem myAVR Board kommuniziert werden. BASCOM bietet dafür ein Terminal und ein DEBUGGER/SIMULATOR an. Dazu sind ein Null-Modemkabel und eine freie COM-Schnittstelle am PC notwendig. Die zu tätigenden Einstellungen in BASCOM sind die Folgenden:

PORT:	COM x
BAUD-Rate:	9600
Bits:	8,N,1
Handshake:	keine
Emulation:	keine

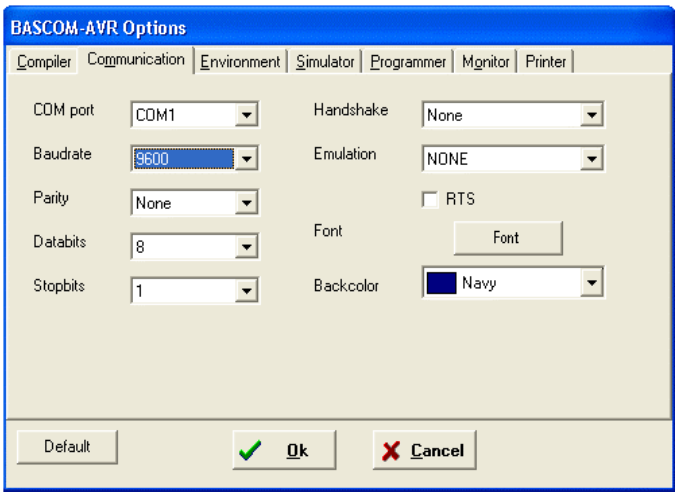
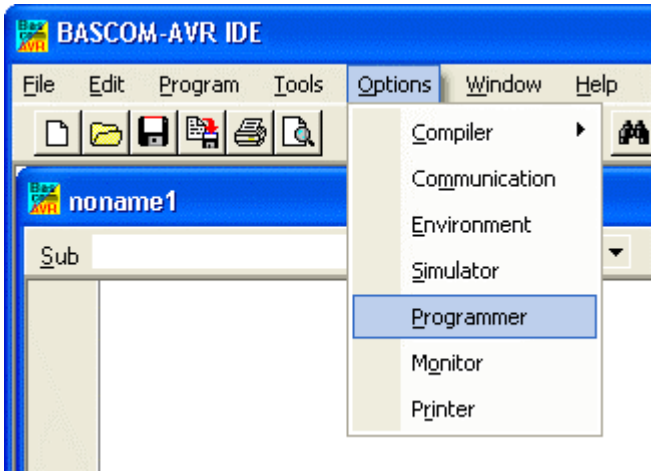


Abbildung: Einstellung für serielle Kommunikation

Wie konfiguriert man BASCOM für das myAVR Board 2 USB?

Auch bei der USB-Version sind zuvor noch einige Einstellungen nötig, bevor man beginnen kann. Das myAVR Board 2 USB verfügt über einen AVR910 kompatiblen USB-Programmer, sowie eine USB-Schnittstelle. Damit die erstellten HEX-Dateien in den FLASH-Programmspeicher des Controllers gebrannt werden können, muss auch hier die Programmerkonfiguration eingestellt und das myAVR Board 2 USB angeschlossen sein.



Die folgende Abbildung zeigt die korrekte Einstellung für das myAVR Board 2 USB

Programmer : AVR ISP Programmer
COM-Port : x (siehe Gerätemanager)
BAUD : 19200

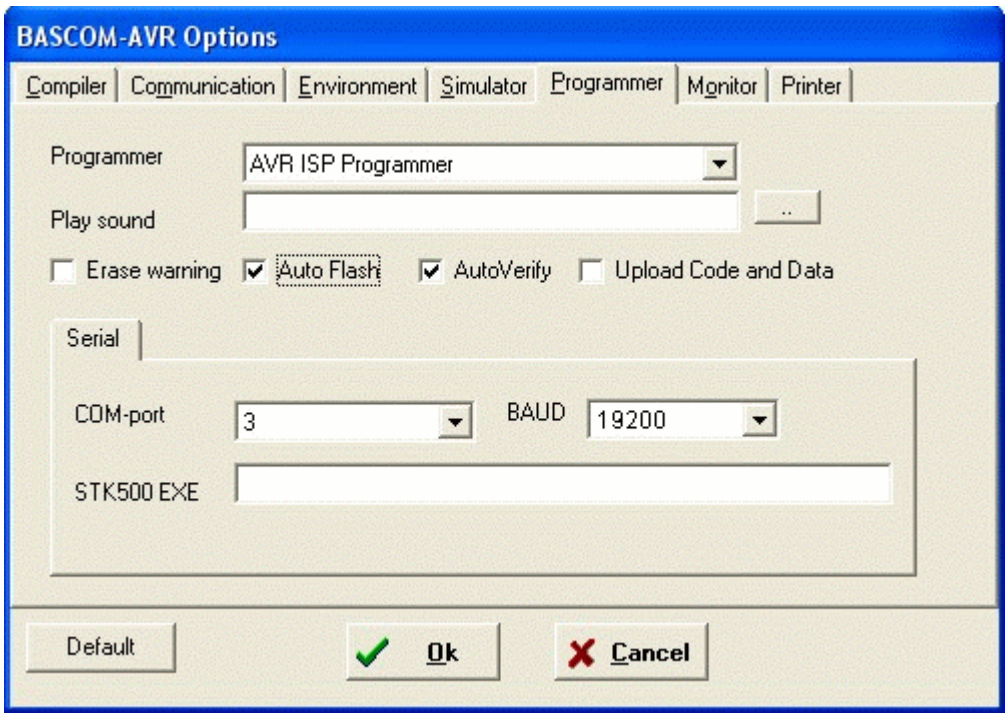


Abbildung: Programmer-Einstellungen für das myAVR Board 2 USB

Grundstruktur eines AVR BASCOM-Programms

Die Grundstruktur eines BASCOM-Programms ist der Struktur eines C-Programms sehr ähnlich. Die Definitionsdatei für AVR-Controller ist die Datei "*.def". In C ist dies die Datei "io.h" und beim Assembler ist das die Datei "avr.h".
Genau wie in C und in Assembler gliedert sich das Hauptprogramm (main) in die Initialisierungssequenz und eine Unendlichschleife, in der die eigentliche Funktion der Mikrocontrolleranwendung ausgeführt wird.

```
'-----------------------------------------------------
' myAVR.bas Vorlage für myAVR Board
' Grundstruktur eines µC-Programms
'-----------------------------------------------------
$regfile = "m8def.dat"
$crystal = 3686400
' hier Initialisierung durchführen
Do
   ' hier Verarbeitungsaufgabe durchführen
Loop
End
```

BASCOM-Befehlssatz

Im Folgenden werden die verfügbaren Sprachkonstrukte aufgelistet. Für eine detaillierte Beschreibung kann die Hilfe in der Entwicklungsumgebung genutzt werden. Dazu stellt man einfach den Cursor auf das entsprechende Statement und betätigt die Taste F1.

Programmsteuerung:

Bedingungen:	IF, THEN, ELSE, ELSEIF, END IF, SELECT, CASE.
Schleifen:	DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, STEP, EXIT FOR.
Unterprogramme:	ON .. GOTO/GOSUB.

Eingaben und Ausgaben:

UART:	PRINT, INPUT, INKEY, PRINT, INPUTHEX, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE.
LCD:	LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, SPC.
PORT/PIN:	DDRx, PORTx.n, DEBOUNCE, SHIFTIN, SHIFTOUT, GETATKBD, SERIN, SEROUT.
TWI/I²C:	I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND and I2CRECEIVE.
1WIRE:	1WWRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST, 1WSEARCHNEXT.
SPI:	SPIINIT, SPIIN, SPIOUT, SPIMOVE.

Mathematische Funktionen:
AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN, COS, TAN, ATN, ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN, POWER, RAD2DEG, DEG2RAD, LOG10, TANH, SINH, COSH.

Bitmanipulation
SET, RESET, ROTATE, SHIFT, BITWAIT, TOGGLE.

Stringmanipulation
STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LTRIM, RTRIM, TRIM, LCASE, UCASE, FORMAT, FUSING, INSTR.

Variablen
DIM, BIT, BYTE, INTEGER, WORD, LONG, SINGLE, STRING, DEFBIT, DEFBYTE, DEFINT, DEFWORD.

Interruptprogrammierung
ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE, COUNTERx, CAPTUREx, INTERRUPTS, CONFIG, START, LOAD.

Kommentare:
REM, ' .

Verschiedenes:
SWAP, END, STOP, CONST, DELAY, WAIT, WAITMS, GOTO, GOSUB, POWERDOWN, IDLE, DECLARE, CALL, SUB, END SUB, MAKEDEC, MAKEBCD, INP,OUT, ALIAS, DIM , ERASE, DATA, READ, RESTORE, INCR, DECR, PEEK, POKE, CPEEK, FUNCTION, READMAGCARD, BIN2GREY, GREY2BIN, CRC8, CRC16, CHECKSUM.

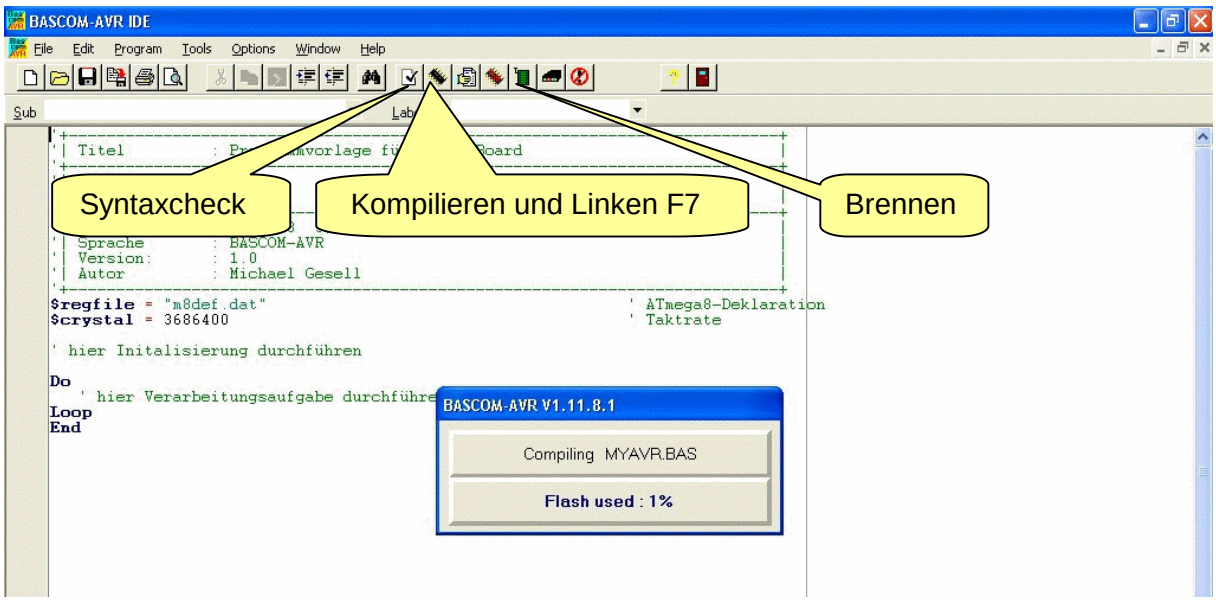
Compileranweisungen, Direktiven
$INCLUDE, $BAUD and $CRYSTAL, $SERIALINPUT, $SERIALOUTPUT, $RAMSIZE, $RAMSTART, $DEFAULT XRAM, $ASM-$END ASM, $LCD, $EXTERNAL, $LIB.

Compilieren und Brennen mit BASCOM

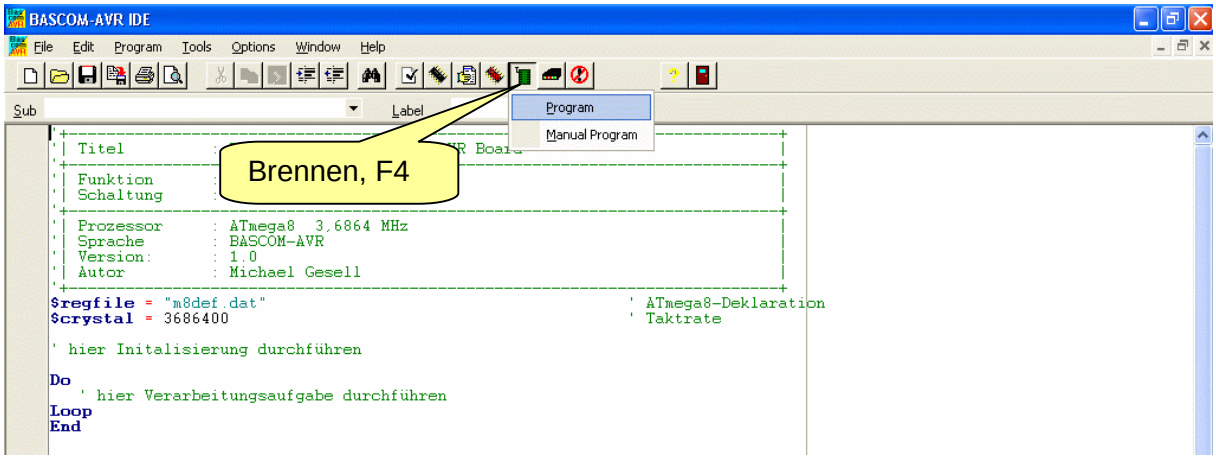
Geben Sie das folgende BASCOM-Programm ein und speichern Sie dieses als „myAVR.bas“.

```
'-----------------------------------------------------------
' myAVR.bas BASCOM Vorlage für myAVR Board
'-----------------------------------------------------------
$regfile = "m8def.dat"
$crystal = 3686400
' hier Initalisierung durchführen
Do
   ' hier Verarbeitungsaufgabe durchführen
Loop
End
```

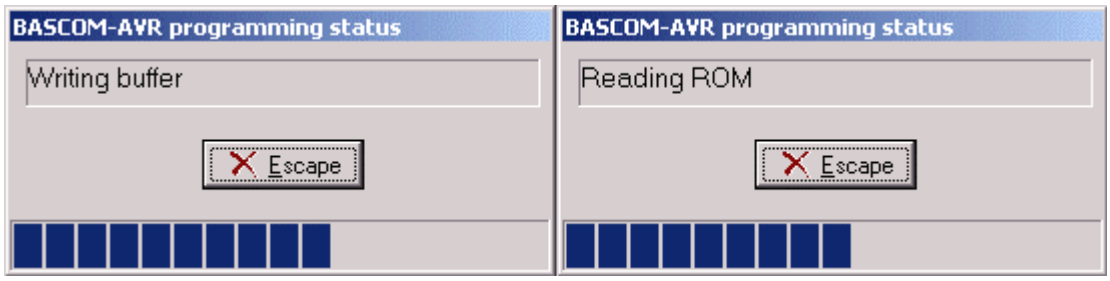
Nach dem Eingeben und Speichern übersetzen (compile current file, F7) Sie das Programm. Wenn keine Fehler angezeigt werden und das myAVR Board angeschlossen ist, kann der Controller programmiert werden (run programmer, F4).



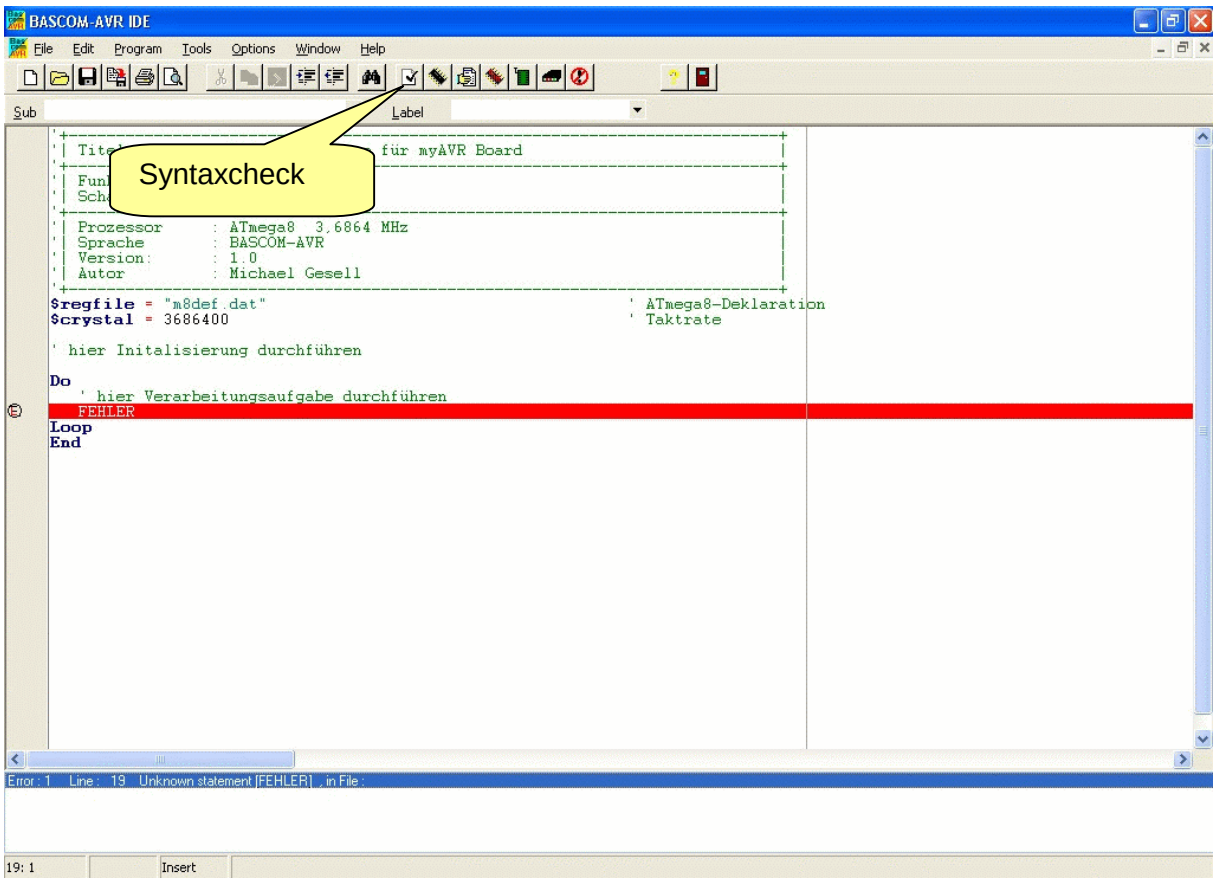
BASCOM versorgt das myAVR Board 1 LPT nicht mit der nötigen Versorgungsspannung über den LPT-PORT. Es ist also immer die Spannungsquelle (z.B.: Batterie) an das myAVR Board 1 LPT anzuschließen. Ebenso ist es zu empfehlen bei der Nutzung des USB-Ports am myAVR Board 2 USB eine externe Spannungsquelle anzuschließen.



Der Brennvorgang (Controller identifizieren, FLASH und EEPROM löschen, schreiben und überprüfen) wird angezeigt. Bei Problemen erscheint eine Fehlerausschrift.



Die Korrektheit des eingegebenen Quellcodes kann mit einem Syntaxcheck überprüft werden.



Ausgaben in BASCOM

Digitale Ausgaben können in BASCOM genau wie in C als Wertzuweisung an den Port (`Port x = wert`) oder das gewünschte Bit (`Port x.n = 0|1`) erfolgen. Der Compiler setzt die Befehle entsprechend um. Zu beachten ist, dass trotzdem über die Steuerregister `DDR x` die Datenrichtung vorher festgelegt werden muss.

Beispiele:

```
Ddrb   = &B11111111           ' gesamter Port B auf Ausgang
Portb  = &B00000001           ' Port B.0 auf 1

Config Portb = Output         ' gesamter Port B auf Ausgang
Portb = &B00000001            ' Port B.0 auf 1

Ddrb.0   = 1                  ' Port B.0 auf Ausgang
Portb.0  = 1                  ' Port B.0 auf 1

'-----------------------------------------------------------
' Titel          : Beispiel "Hallo Welt" für myAVR Board
' Datei          : hallowelt.bas
'-----------------------------------------------------------
' Funktion       : LED einschalten
' Schaltung      : PB0 an LED
'-----------------------------------------------------------
' Prozessor      : ATmega8  3,6864 MHz
' Sprache        : BASCOM-AVR
' Version:       : 1.0  , 02.08.2004
' Autor          : Michael Gesell
'-----------------------------------------------------------
$regfile = "m8def.dat"            ' Prozessortyp ATmega8
$crystal = 3686400                ' Taktrate

Ddrb = &B00000001                 ' Port B.0 auf Ausgang

Do                                ' Begin Mainloop
   Portb.0 = 1                    ' PB 0 LED an
Loop                              ' Ende Mainloop
End                               ' Programmende
'-----------------------------------------------------------
```

Variablen und deren Verarbeitung in BASCOM

Variablen werden mit dem Schlüsselwort `DIM` deklariert. Initialwerte sind danach als Wertzuweisung anzugeben. Diese werden im FLASH gespeichert und beim Programmstart automatisch in den SRAM übertragen. Bei Feldern (Arrays), wie zum Beispiel Zeichenketten, ist die Anzahl der Feldelemente als Faktor anzugeben.

Beispiele:

```
'ein Byte
Dim Mybyte As Byte
'ein 5 Byte
Dim myarray As Byte * 5
Dim Myword As Word
Dim Mylong As Long
Dim Myint As Integer
Dim Mystring As String * 10
```

Die Verarbeitung der Daten erfolgt dann in der Hauptschleife (`mainloop`).

Wertzuweisung	: **=**
Arithmetische Operationen	: **+, -, *, /, ^**
Logische Operationen	: **=, <, >, <=, >=, <>, NOT, AND, OR, XOR**
Bit-Operationen	: **NOT, AND, OR, XOR, ROTATE, SHIFT**

Als Übung soll das Lauflichtbeispiel in BASCOM realisiert werden. Dabei wurde ein Bit in einem Register links verschoben (Rotation), kurz gewartet und das Register an einem PORT ausgegeben.

Benötigte BASCOM Befehle: **`DIM, ROTATE, WAITMS`**.

Nutzen Sie die Hilfe (F1), um die Befehle kennen zu lernen.

```
'-----------------------------------------------------------
' Titel         : Lauflicht für  myAVR Board
'-----------------------------------------------------------
' Funktion      : Lauflicht
' Schaltung     : PD5-PD7 an LED`s
'-----------------------------------------------------------
' Prozessor     : ATmega8  3,6864 MHz
' Sprache       : BASCOM-AVR
' Version       : 1.2  , 23.08.2004
' Autor         : Michael Gesell
'-----------------------------------------------------------
$regfile = "m8def.dat"          ' Prozessortyp ATmega8
$crystal = 3686400              ' Taktrate

Dim Mybyte as Byte              ' ein Byte als Variable
Ddrd   = &B11100000             ' PD 5-PD 7 auf Ausgang
Portd  = &B00000000             ' alle LEDs off

Mybyte = 1                      ' Startwert &B00000001

Do                              ' Beginn Mainloop
   Portd = Mybyte               ' Ausgabe
   Waitms 100                   ' Warte kurz
   Rotate Mybyte, Left          ' Bit laufen lassen (Rotation)
Loop                            ' Ende Mainloop
End                             ' Programmende
'-----------------------------------------------------------
```


Eingaben in BASCOM

Eingabeoperationen werden in BASCOM ebenfalls durch Wertzuweisungen oder auch durch Vergleichsoperationen in Bedingungen „direkt“ ausgeführt. Sie können die Ports und I/O-Register wie jede andere Variable handhaben.

Beispiele:

```
'eine Ausgabe an PORT B
PortB = Mybyte
'eine Eingabe von PIN B
Mybyte = PinB
'eine Vergleichsoperation, bei der von PIN B.0 gelesen wird
If Pinb.0 = 0 Then
'lesen eines I/O-Registers (Timer2)
Mybyte = Tcnt2
```

Zu beachten sind wiederum die Steuerregister `DDR x`, in denen die Datenrichtung festgelegt wird. Eine besondere Form der Ausgabe ist der Befehl `config`. Dieser realisiert mehr oder weniger komplexe Initialisierungen, welche letztlich Ausgaben in den entsprechenden I/O-Registern sind.

```
'entspricht Ddrb = &B00000000
config Portb = Input
'entspricht Ddrd = &B11111111
config Portd = Output
```

```
' VARIANTE 1 ohne Variable
'-----------------------------------------------------------------
' Titel         : intelligenter Lichtschalter für myAVR Board
'-----------------------------------------------------------------
' Funktion      : LED bei Tastendruck ON
' Schaltung     : Taste an PD.5, rote LED an PD.6
'-----------------------------------------------------------------
' Prozessor     : ATmega8  3,6864 MHz
' Sprache       : BASCOM-AVR
' Version:      : 1.1  , 20.08.2004
' Autor         : Michael Gesell
'-----------------------------------------------------------------
$regfile = "m8def.dat"       ' Prozessortyp ATmega8
$crystal = 3686400           ' Taktrate

Ddrd   = &B01000000          ' PD 6 auf Ausgang
Portd  = &B00100000
Do
    If Pind.5 = 0 Then       ' Eingabe PORT D.5!
       Portd.6 = 1           ' dann PD 6 einschalten
    Else                     ' sonst
       Portd.6 = 0           ' PD 6 anschalten
    End If
Loop
End
'-----------------------------------------------------------------
```

```
' VARIANTE 2 mit Variable
'-----------------------------------------------------------
' Titel         : intelligenter Lichtschalter für myAVR Board
'-----------------------------------------------------------
' Funktion      : LED bei Tastendruck ON
' Schaltung     : Taste an PD.5, rote LED an PD.6
'-----------------------------------------------------------
' Prozessor     : ATmega8  3,6864 MHz
' Sprache       : BASCOM-AVR
' Version:      : 1.2  , 20.08.2004
' Autor         : Michael Gesell
'-----------------------------------------------------------
$regfile = "m8def.dat"       ' Prozessortyp ATmega8
$crystal = 3686400           ' Taktrate

Ddrd   = &B01000000          ' PD 6 auf Ausgang
Portd  = &B00100000
Dim Mybyte as Byte           ' ein Byte als Variable
Do
    Mybyte = Pind            ' Eingabe von PORT D !
    If Mybyte.5 = 0 Then     ' Wenn PD 5 angeschalten
       Portd.6 = 1           ' dann PD 6 einschalten
    Else                     ' sonst
       Portd.6 = 0           ' PD 7 anschalten
    End If
Loop
End
'-----------------------------------------------------------
```

Nachwort

Diese Kurzeinleitung ist ein Auszug aus dem myAVR Lehrbuch Mikrocontrollerprogrammierung welches sich unter anderem den myAVR Einsteigersets beiliegt. Mehr Informationen erhalten Sie unter **www.myAVR.de** .

