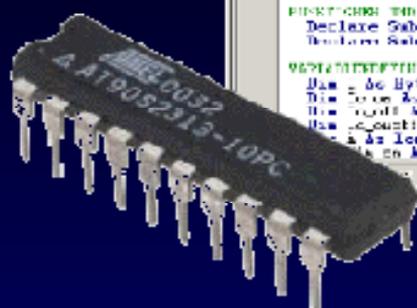
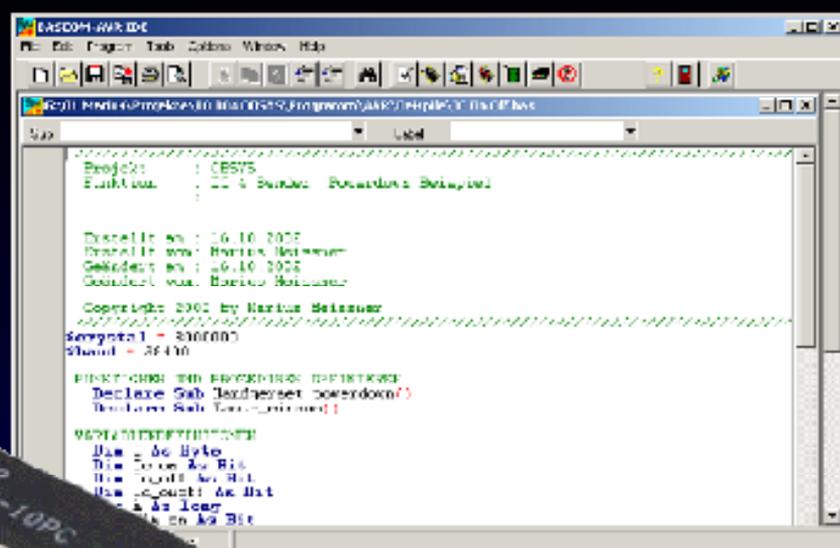


Marius Meissner



BASCOM-AVR

Version 1.11.7.4

Sprachbefehle

Mit CD-ROM



- Bascom-AVR Demo
- Beispielprojekte
- Datenblätter

Beschreibung der BASCOM-AVR Basic Befehle

Die nachfolgende Zusammenstellung der in BASCOM-AVR Basic verwendeten Befehle soll einen Überblick über die mittlerweile recht umfangreichen Entwicklungsmöglichkeiten der Sprachumgebung geben.

In der aktuellen BASCOM-AVR Basic Version (1.11.7.4) gibt es insgesamt 344 Sprachbefehle. Sie können in folgende Gruppen eingeteilt werden:

1. Prozessor Direktiven
2. Compiler Direktiven
3. Instruktionen
4. Funktionen
5. Hardwareeinstellungen

Die neben der Syntax in Klammern auftauchenden roten Zahlen sind die Versionen, ab denen die Befehle in der Version BASCOM-AVR Basic vorkommen. Diese wurden aus der history.txt, die mit der kommerziellen Version der BASCOM-AVR Basics ausgeliefert wird, ermittelt.

Alle Befehle, die mit ** gekennzeichnet sind, benötigen eine zusätzliche ASM Library (Bibliothek). Nicht jede Library ist Bestandteil des BASCOM-AVR Basics: Die fehlenden Libraries können zusätzlich bei Drittanbietern erworben werden.

Die Beschreibung eines jeden Sprachbefehls wurde nach einem festen Muster durchgeführt und wegen der Übersichtlichkeit jeweils auf einer neuen Seite platziert. Die Beschreibungen der Prozessor- und Compiler Direktiven, Instruktionen, Funktionen und Hardwareeinstellungen setzen sich aus folgenden Unterpunkten zusammen: Beschreibung, Syntax, Bemerkungen, siehe auch, ASM und Beispiel.

Beschreibung: Hier wird die Befehlfunktion beschrieben.

Syntax: Hier wird die Befehlssyntax beschrieben.

Bemerkungen: Hier werden die notwendigen Variablen bzw. Konstanten der Syntax beschrieben sowie verschiedene Informationen, die den Befehl erläutern bzw. deren Funktionalität vertiefen.

Siehe auch: Hier können ähnliche Befehle bzw. ganze Befehlsfamilien aufgeführt sein.

ASM: Hier werden die Assembler Befehle erläutert.

Beispiel: Hier folgt ein Quelltext Beispiel anhand dessen der Einsatz des Befehls dargestellt wird.

Die Unterpunkte „ASM“ und „Siehe auch“ sind optional und nicht bei jedem Befehl vorhanden.

Die vorhandenen Befehle sind alphabetisch aufgeführt. Auf eine Gliederung wurde bewußt verzichtet. Dies soll eine Suche durch Blättern im Buch verkürzen und vereinfachen.

Strukturierung der BASCOM-AVR Basic Befehle

Die nachfolgend beschriebenen Befehle können in die darzustellenden Funktionsgruppen eingeteilt werden:

Programmdokumentierung

Kommentierung

REM, '

Fehlerbehandlung

Debug

DBG, STCHECK

Variablen und Funktionen

Variablen und Konstanten

DIM, CONST, ALIAS, LOCAL, STRING, DEFBIT, DEFBYTE, DEFINT, DEFWORD, BYVAL

Funktionen und Prozeduren

SUB, END SUB, FUNCTION, CALL, DECLARE, END,

Verzweigungen, Schleifen und Sprunganweisungen

Schleifen

FOR, TO, STEP, EXIT FOR, NEXT, DO, UNTIL, EXIT DO, LOOP, WHILE, EXIT WHILE, WEND,

Verzweigungen

IF, THEN, ELSE, ELSEIF, END IF, SELECT, CASE.

Sprunganweisungen

ON .. /GOTO, GOSUB, VALUE

Datum Zeit Funktionen

Datumbearbeitung Standard

DATE\$, TIME\$

DATEIME.LIB

DAYOFWEEK, DAYOFYEAR, SECOFDAY, SECELAPSED, SYSDAY, SYSSEC, SYSSECELAPSED, DATE, TIME

Numerische Funktionen

Trigonometrie

SIN, COS, TAN, SINH, COSH, TANH, ASIN, ACOS, ATN, ATN2, RAD2DEG, DEG2RAD,

Potenzen, Wurzeln

SQR, POWER

Logarithmen

EXP, LOG, LOG10,

Wertumwandlung

ABS, BCD, FIX, ROUND, SGN

Kommazahlen

FRAC

Zeichenkettenverarbeitung**Zeichenketten-Informationen**

ASC, CHR, LEN, INSTR, HIGH, HIGHW, LOW, MAX, MIN

Zeichenketten-UmwandlungLCASE, UCASE, FORMAT, STR, STRING, VAL, HEXVAL, HEX, INCR,
DECR, SWAP, MAKEDEC, MAKEBCD, MAKEINT, BIN2GREY,
GREY2BIN, BIN, INT**Zeichenketten-Bearbeitung**

LTRIM, RTRIM, TRIM, LEFT, RIGHT, MID, FUSING, SPACE

Bit Manipulation

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

Prüfsummen

CRC8, CRC16, CHECKSUM.

Verzögerungen Unterbrechungen

WAIT, WAITMS, DELAY

Chip spezifische Funktionen**Ausgabe am LCD Standard**PRINT, DISPLAY ON/OFF, UPPERLINE, LOWERLINE, LCD, CURSOR
ON/OFF/BLINK/NOBLINK, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR
LEFT/RIGHT, CLS, LOCATE, HOME, DEFLCDCHAR, FOURTHLINE,
INITLCD, LCDCONTRAST, THIRDLINE**GLCD.LIB**

LOCATE, CLS, PSET, LINE, CIRCLE, SHOWPIC, SHOWPICE

GLCDSED.LIB

LCDAT, SETFONT, GLCDCMD, GLCDDATA

Tastatur/Maus, Tasten**Eingabe an der PC AT-Tastatur**

GETATKBD,

PS2MOUSE_EMULATOR.LIB

CONFIG PS2EMU

PS2MOUSEXY

SENDSCAN

AT_EMUALTOR.LIB

CONFIG ATEMU, SENDSCANKBD

Tasten Eingaben

DEBOUNCE, GETKBD

Kommunikation**UART**

PRINT, PRINTBIN, SHIFTOUT, SEROUT, SPC, OPEN, CLOSE, INKEY, SHIFTOIN, SERIN, INPUT, INPUTBIN, INPUTHEX, WAITKEY, BAUD, ECHO, ISCHARWAITING

I2C

I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND, I2CRECEIVE, IC2INIT

1WIRE

1WWRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST, 1WSEARCHNEXT, 1WVERIFY

SPI

SPIINIT, SPIIN, SPIOU, SPIMOVE.

IR-Kommunikation

RC5SEND, RC6SEND, SONYSEND, GETRC5, GETRC

TCPIP.LIB

CONFIG TCPIP, BASE64DEC, CLOSESOCKET, GETDSTIP, GETDSTPORT, GETSOCKET, IP2STR, SETTCP, SOCKETCONNECT, SOCKETLISTEN, SOCKETSTAT, TCPREAD, TCPWRITE, TCPWRITESTR, UDPREAD, UDPWRITE, UDPWRITESTR

X10

X10DETECT, X10SEND

Laufwerke**Laufwerke**

DRIVERESET, DRIVEINIT, DRIVEGETIDENTITY, DRIVEWRITESECTOR, DRIVEREADSECTOR, DRIVECHECK

AVR-DOS.LIB

BLOAD, BSAVE, CLOSE, DIR, DISKFREE, DISKSIZE, EOF, FILEATTR, FILEDATE, FILEDATETIME, FILELEN, FILETIME, FLUSH, FREEFILE, GET, INITFILESYSTEM, INPUT, KILL, LINE INPUT, LOC, LOF, OPEN, PRINT, PUT, SEEK, WRITE

BCCARD.LIB

CONFIG BCCARD, BCRESET, BCDEF, BCCALL

Magnetkarte

READMAGCARD

Interrupt Programmierung

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

IC Deaktivierung

POWERDOWN, IDLE, POWERSAVE

Speicherbearbeitung

INP, OUT, PEEK, POKE, CPEEK, CPEEKH,

Werte im Programm speichern

DATA, READ, RESTORE

Tabellen

LOOKUP, LOOKUPSTR, LOOKDOWN

EEPROM

READEEPROM, WRITEEEPROM

Sound

SOUND

Telefon

DTMFOUT

Puls generierung

PULSEIN, PULSEOUT

Takt

CLOCKDIVISION, CRYSTAL

Register

LOADADR, LOADLABEL, POPALL, PUSHALL, VARPTR

Zufall

RND

Prozessor Direktiven

#IF ELSE ENDIF

Compiler Direktiven

\$ASM, \$BAUD, \$BAUD1, \$BGF, \$BOOT, \$CRYSTAL, \$DATA, \$DBG,
\$DEFAULT, \$EEPROM, \$EEPROMHEX, \$EXTERNAL, \$INC,
\$INCLUDE, \$LCD, \$LCDPUTCTRL, \$LCDPUTDATA, \$LCDRS, \$LCDVFO,
\$LIB, \$MAP, \$NOINIT, \$NORAMCLEAR, \$PROG, \$REGFILE, \$ROMSTART,
\$SERIALINPUT, \$SERIALINPUT1, \$SERIALINPUT2LCD,
\$SERIALOUTPUT, \$SERIALOUTPUT1, \$SIM, \$TINY, \$WAITSTATE,
\$XRAMSIZE, \$XRAMSTART

Compiler Einstellungen

CONFIG 1WIRE, CONFIG ACI, CONFIG ADC, CONFIG ATEMU, CONFIG
BCCARD, CONFIG CONFIG, COM1CLOCK, CONFIG COM2, CONFIG
DATE, CONFIG DEBOUNCE, CONFIG GRAPHLCD, CONFIG I2CDELAY,
CONFIG I2CSLAVE, CONFIG INTx, CONFIG KBD, CONFIG KEYBOARD,
CONFIG LCD, CONFIG LCDBUS, CONFIG LCDMODE, CONFIG LCDPIN,
CONFIG RC5, CONFIG PORT, CONFIG SERIALIN, CONFIG SERIALIN1,
CONFIG SERIALOUT, CONFIG SERIALOUT1, CONFIG SERVOS, CONFIG
PS2EMU, CONFIG SDA, CONFIG SCL, CONFIG SPI, CONFIG TCPIP,
CONFIG TIMER0, CONFIG TIMER1, CONFIG TIMER2 und 3, CONFIG
WATCHDOG, CONFIG WAITSUART, CONFIG X10

#IF ELSE ENDIF (**>= 1.11.5.0**)

Beschreibung:

Ermöglicht die bedingte Aufnahme des Quellcodes beim kompilieren.

Syntax:

```
#IF Bedienung  
    Kommandos  
#ELSE  
    Kommandos  
#ENDIF
```

Bedienung: Gibt den logischen Ausdruck an, der ausgewertet wird.

Bemerkungen:

Die bedingte Aufnahme des Quellcodes beim kompilieren wird vom BASCOM-AVR Compiler unterstützt.

Was versteht man darunter?

Bei der bedingten Aufnahme des Quellcodes beim Kompilieren werden nur die Teile des Quellcodes kompiliert, die dem ausgewerteten logischen Ausdruck der Bedienung entsprechen.

Wird dieser Befehl nicht benutzt, wird der gesamte Quelltext kompiliert.

Die bedingte Kompilation erfordert eine Konstante, die bevor dieser Befehl eingesetzt werden kann, definiert werden muß.

```
CONST test = 1  
#IF TEST  
    Print "Das wird compaliert"  
#ELSE  
    Print "und das nicht"  
#ENDIF
```

Merke: Bei diesem Befehl darf nach IF kein THEN stehen. Darüber hinaus ist das #ENDIF nicht gleich #END IF. Dieses beinhaltet kein Leerzeichen.

Wird eine zusätzliche Bedienung gefordert kann #ELSE optional zum Einsatz kommen.

BASCOS-AVR BASIC besitzt die folgenden, durch den Compiler generierte, interne Konstanten:

```
_DATE_FORMAT      = 0  
_DATE_SEPARATOR  = 47  
_CHIP             = 0  
_RAMSIZE         = 128  
_ERAMSIZE        = 128  
_SIM             = 0  
_XTAL            = 4000000
```

_BUILD = 11174
_COMPILER = 90

_CHIP ist eine Zahl, die den Chip beschreibt; in diesem Fall 2313
_RAMSIZE ist die Größe des SRAM

_ERAMSIZE ist die Größe des EEPROM
_SIM ist auf 1 gesetzt, wenn \$SIM Directive genutzt wird
_XTAL beinhaltet den Wert des angegebenen Kristalls
_BUILD ist die Versionsnummer des Compilers.

Eine Versionsnummer kann zum Einsatz kommen, wenn ein Programm bestimmte Funktionen in unterschiedlichen Programmversionen zur Verfügung stellen soll.

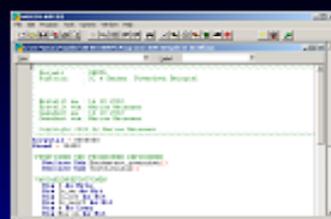
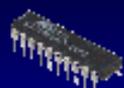
Beispiel:

```
#IF _BUILD >= 11162
    s = Log(1.1)
#ELSE
    Print "Funktion leider in Programmversion 1.11.6.2 implementiert"
#ENDIF
```

BASCOM-AVR

Version 1.11.7.4

IDE - Sprachbefehle



Sie wollen die Programmiersprache BASCOM-AVR nutzen? Schnell stellen Sie fest, dass es nur wenig deutschsprachige Literatur gibt. Dieses Buch soll Abhilfe schaffen. Es stellt die BASCOM-AVR Basic Sprachbefehle in deutscher Sprache vor und erläutert sie anhand von Quelltexten. Auf diese Weise erhalten Sie wertvolle Informationen zur professionellen Programmierung mit BASCOM-AVR Basic.

Ein Nachschlagewerk, das auf keinen Schreibtisch fehlen sollte.

Aus dem Inhalt

Fortgeschrittene wie Einsteiger lösen mit Hilfe dieses Buchs schnell die häufigen Problemstellungen. Dies spart Zeit und nebenbei werden durch das Stöbern im Fundus interessante Details der Programmiersprache vermittelt. Für den professionellen Einsatz finden Sie in diesem Buch alle Sprachbefehle einschließlich der Lib Bibliotheken bis zu Version BASCOM-AVR Basic 1.11.7.4.

Vorausgesetzt werden die üblichen Windows-Kenntnisse sowie Grundkenntnisse in den Bereichen Elektronik und Visual Basic.

Die beiliegende Demo-Version des BASCOM-AVR Programms ermöglicht es dem Leser, BASCOM-AVR in der lediglich 2 KB umfassenden Entwicklungsumgebung zu testen. Hierfür stehen zahlreiche Beispielanwendungen mit den dazu gehörenden Quelltexten zur Verfügung.

Auf der CD



- Bascom-AVR Demo
- Beispielprojekte
- Datenblätter