

-----  
BASCOS-AVR History

(c) 1995-2012 MCS Electronics

the history items are very short descriptions of changes. it is only intended  
to allow you to find the info in the help.

-----  
2.0.7.5.002

- \$hwstack,\$swstack and \$framesize used with comment like \$hwstack=40 '500 resulted in wrong values.
- rc5 background option only worked for 8 Mhz

2.0.7.5

- config COM , IRDA baud rate calculation fixed.
- version() with parameter 3 will return file name : Print Version(3)
- xmega autoprog will set lock bit last now.
- RS485 (config print) support added to printbin also changed
- rc5.lib added, sponsored by Lumicon, <http://www.lumico.de> , see config rc5, mode=background
- improved usb read handling for quicker flash programming.
- eeprom programming xmega/PDI mode fixed.
- \$stackdump added. See also stackdump.bas. Based on code from MWS.
- history.txt is shown once for each new version of the IDE
- ERR is set to 1 when there is not enough frame space. you use frame space when using locals and passing parameter byval.  
also a test like : if left(string,3)="value" will use temp space.
- added link to the bascom-avr.xml file location in the Help, About window.
- boot loader samples improved : bblocklocal incremented after a valid block.
- huge memory on Xmega bug fixed : rampX and r12 inside an ISR were not preserved.
- config xram for xmega, fixed bug for sram mode. also, baseadr changed, see help.
- \$timeout help updated (it works for SW UART too), and INPUT with \$timeout added to SW UART.
- spiinit, will set only miso to output when slave mode is selected.
- w5200 support added.
- W5100 tcp/ip support added.
- w5300 tcp/ip support added.
- xmega : saving watchdog cause in r0 was not implemented. also, the flags were not cleared at start up.
- an updated version of bascomp.exe was not included in the release.
- sample files are now located in users\USER\Documents\MCS Electronics\BASCOM-AVR\Samples
- split screen editor fixed.
- added CAN support. See CONFIG CANBUS, CANMOB, CANBAUD,CANRESET, CANSEND, CANRECEIVE,etc.
- removed backspace from INPUT.
- added M8U2 and M16U2 support
- fixed RTC32 and added new sample from MAK3. See XM256A3BU-RTC32.bas
- STCHECK checking framesize (error 3) did not work. So you could get error 3 while there was no error.
- DTMFOUT did not saved r23 which could lead to erratic behavior for chips with TIMSK in high IO regs.
- documented LCDCMD and LCDDATA
- When adding files in Project Mode, you can select multiple files using CTRL/SHIFT.
- Adding a CONSTANT \_CHECKVAL with a value <>0 will test for illegal chars in VAL()  
Const \_CHECKVAL =1 : Dim S as string \* 10 : s = "1234a" : w = val(s) : print ERR
- config clock, gosub=sectic option was sensitive to position.
- user sub/functions calls are protected against interrupts. See \$noframeprotect
- xmega config event\_system : portA and portB were reversed. ADCB\_CH0/ADCB\_CH3 would result in a wrong setting too.
- config servos has a new mode to use servos with less system load. this mode works on xmega as well.
- using an overlay array with config base=0, would result in the wrong overlay address.
- full version can run updatewiz from the IDE

#### 2.0.7.4.002

- fixed varptr() error
- updated dat files to reflect new atmel web location
- added \$xteakey to encrypt the binary file with a 128 bit key. (see help)

#### 2.0.7.4 BETA

- SSD1325 lib updated. fonts were not shown properly (there was a pixel between each line)
- AVR-DOS SDHC driver updated for Xmega and included.
- oled sample included
- modbus slave included
- right margin line can be adjusted.
- DCF77 support added for Xmega
- internal numeric conversion routine will report non-numerical trailing data that was previously ignored (spaces, etc.)
- added support for xmega big memory(8MB), this is still considered a BETA feature.
- added Find in Files option to the IDE.
- changed HWMUL=1 into HWMUL=0 for usb162,usb82,m32u2def
- STOP TIMER1 did not stop timer for tiny25/45/85
- attiny25 adc noise canceler.bas sample added from MAK3
- attiny13A.dat file powermode section corrected.
- .align worked with word address instead of byte address as intended.
- A = &H6000 Or &H8000 resulted in error 51.
- using select case with a string function would only work for the first case.
- RTC32 support added for Xmega
- atmega128RFAL dat file added.
- attiny85 as SPI SLAVE using USI sample added from MAK3
- data line would not convert a const correct for single/double : const x=12.34 : data x! would fail
- the following code is no longer required for the xmega:  
\$lib "xmega.lib" : \$external \_xmegafix\_clear : \$external \_xmegafix\_rol\_r1014
- CTRL+J will show templates.
- XTEAENCODE and XTEADECODER statements added to encrypt/decrypt using XTEA algo
- added hexval.lib from user MWS. This is an alternative lib for HEXVAL that removes spaces and sets ERR if illegal characters are found
- string() and space() when used with a string array with a variable index : var(varx)=space(x) will result in a memory overwrite, depending on the index and address of the array.
- attiny261/461/861 have one shared interrupt flag for PCINT but 2 individual enable flags. Added to the dat files.
- added BANDGAP=ENABLED|DISABLED to CONFIG ACI. It will only work for chips that support this option(most new chips)
- for external programmer, {CHIP} can be used to specify the device/chip.
- m32M1 missed PCICR register in dat file. The prescale value for timer0 was not correct loaded for config timer0
- [FLAG] section updated. Some chips have combined flag/normal in extended IO space. For example USISR in the M169  
These regs need special handling to reset a bit.
- waitus improved. normally an internal sub routine is called but for some xtal values and delays, this is not possible.  
in such a case, NOPS are generated. But this can be as much as 14 nops. In such cases a small single byte loop delay is generated.
- INP/OUT support Xmega huge memory
- added Arduino V2 programmer (uses stk500v2 protocol) for M2560 R3
- CONFIG XPIN added for xmega pin setup.
- print bin() used with a long/dword would overflow the internal string buffer. the frame is used now.
- array with negative offset like : ar(constant - var) pointed to the wrong memory. var-constant was no problem.
- added project mode option with explorer. you might need to reset the docking to make it visible.
- ON interrupts accepts SAVE and SAVEALL too. SAVE is the default and used when no parameter is provided.
- 2074 requires an update of the bascomp.exe command line utility

### 2.0.7.3

- If writing assembler, please use \$ASM..\$NO ASM or the !  
While asm is currently supported without preceding !, it is not allowed in the future.  
So this is good:  
\$asm  
  nop  
\$end asm  
! nop  
But this is not good:  
nop
- When using CONFIG LCD, use X instead of \* : config LCD = 16x2 instead of 16\*2  
Both are valid, but the \* will become invalid in the future.
- When using parameter options with spaces like PWM A, use underscores instead : PWM\_A  
Both are valid, but spaces are not allowed in the future.
- use CONFIG POWERMODE instead of the POWER statement. See Help.
- getRc\_m128\_PINF.lib added for getrc() on M128, portf.
- shiftout option parameter did not allow constants (numeric parameter had to be specified)
- added .align mnemonic to the internal assembler
- m48def.dat, WDVALUE must be : WDVALUE=16,32,64,128,256,512,1024,2048,4096,8192
- CONFIG TCx , PWM and WAVE modes will automatically set the port direction register for COMPAREx options.  
But it also set the other port direction bits to output which was wrong.
- xmega LCD simulation added
- shiftout option parameter did not allow constants (numeric parameter had to be specified)
- xmega uart simulation corrected.
- Editor did not repaint properly in some cases when adding a new line.
- avrisp mkII programmer can write all xmega fusebytes individually.
- config print1|print2|print3 settings were not cleared at the begin of the compilation.  
This would give the effect that when used once, the config would remain in effect till the IDE was restarted.
- \$FILE would append to the error file instead of overwriting it.

### 2.0.7.2

- fonteditor plugin accepts 64x64 fonts now.
- config TCxx changed : now the wgmode parameter is expected :  
config TCxx=NORMAL|FREQ|PWM|PWM\_TOP|PWM\_BOT|PWM\_TOPBOT, other params....
- config timer has a new optional parameter : configuration  
Config Timer1 = Timer , Prescale = 1 , Configuration = Abc  
Config Timer1 = Timer , Prescale = 1024 , Configuration = Test  
The config is stored in a name you provide. With this configuration you can start the timer using this configuration:  
Start Timer1 , Abc  
Start Timer1 , Test  
See Help.
- the IDE sometimes painted code as comment.
- oled ssd1325 support added for 96x64 display. also included, an RLE compressor for 16 grey scale images.  
based on bascom code from Robert Wolgajew
- dmxslave support for xmega added. dmx samples from MAK3 added.
- new xmega samples from MAK3. XMEGA\_easy\_Frequency\_Generation.bas and XMEGA\_single\_slope\_PWM.bas
- setting bit with option base 0, could give range error.  
this fix, also fixes other bit range error.

- mixing byte with string constants in DATA lines created wrong data image.
- #if someStrConst = "constantvalue" was not correct evaluated in the preprocessor.
- secelapsed had no reference to used routines : mul32 and mul16
- print/lcd on arrays like : var(idx+1) , gave an error
- ##### Please read the important info about timer pwm mode of 2071.

#### 2.0.7.1

- corrected help for shiftin, external clock mode. (edges were reversed for external clock)
- attiny20 added.
- ##### IMPORTANT #####
- compare CLEAR UP and compare CLEAR DOWN, were reversed. This mean they would select the reverse mode.  
if you use timers in PWM mode with CLEAR UP/DOWN mode, you might need to check it (depending on your app)
- #####
- updated config spi help for xmega.
- F1 will now show Index tab with the help word selected.
- m328def.dat added (but you should use new m328P chip and dat, only added for users with old chips)
- Hardware Simulator did not used PIN registers with extended address correct. (for example PINK for m2560)
- xmega rampz problem fixed for right() function and for readeeprom/writeeprom. these functions used a different way to load the Z pointer.
- string() function used with local variables trashed the registers.
- added waitms reference to config graphlcd.
- when using config submode=new, and not using any serial command in the main module, the uart was not initialized.
- inkey/ischarwaiting code added for xmega, serial buffered mode
- using {xxx} in string compare did not work.
- added Setreg regname, value
- added Getreg(regname)  
Now you can set/get register values from basic
- lwire for xmega bug fixed for lwrite using a constant.
- lwire sending 1 byte code reduced

#### 2.0.7.0 few sub versions have skipped because of this new feature

- config submode=new will use a new mode with the following changed rules:
  - declare function/sub is not required
  - sub/function code is included only when used
  - sub/function code is included after your code, not at the place where it is in your code
  - while tested on a number of projects and all samples, this mode is off by default to maintain compatibility
- the IDE has an option to use only com ports that exist. the com ports (terminal emu, sim, programmers) are changed under their name not by a relative index. this means you must select the comports once.  
when you do not select the option to use com ports that exist, you get a list of COM1-COM255.
- loadlabel() will work on EPROM too now, but it must be used after the eeprom image. Thus the labels must be placed before the loadlabel() function
- readeeprom help , bylabel help
- double\_trig did not use \_lpbyte causing rampz problem
- m1284pdef.dat had a wrong timer alias for ovf3

#### 2.0.6.3

- date\$/time\$ when used in compare (if then, select case etc.) will load the date/time before the comparison now.  
you should consider using internal variables \_day, \_hour etc. however!
- delchar/delchars loaded pointers in wrong order.
- using input on com3 for m2560 failed.(duplicate label used)
- added tnl67def.dat file.
- sim did not show hex/bin values properly for dword variables
- binval/hexval dword support added.

- added m324PAdef.dat
- bit arrays calculation changed, required by a previous address calculation fix.

#### 2.0.6.2

- added sample for MyAVR MK3 (display, 7-seg)
- extended fuse byte added to 2313,2313A and 4313.
- if you use INITLCD in your code, the compiler will not call `_init_lcd` any longer as part of the initialization. this way you can control when the lcd is initialized. for an xmega this should be done after the oscillator is chosen and the clock is stable.
- lcd4busy\_anypin lib changed to support xmega
- set/reset will use register r23 instead of r24

#### 2.0.6.1

- power modes did not update power mode registers, this would make the power modes fail.
- rs485 sample added from MAK3 for xmega and arduino
- dsl307\_GER.lib added for European date/time convention. Mods by MWS.
- pulsein adapted for xmega, also sample added from MAK3.
- report module calculated baud, but this gave an error if the uart was not used and bascomp.exe was used.
- dma channel repeat counter is a byte (and probably a word in the future) but it is moved back from the WIO section since it is a byte.
- {xxx} dit not work for data statements due a change in 2060.

#### 2.0.6.0

- \$aeskey 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 will create an AES 128 bit encrypted file. you can NOT simulate this file since it is encrypted. it is intended to be used with the AES bootloader from Reinhars. this last minute addition is not documented in the help yet!
- sample included to calculate the median for a big data set. see ReMedian 1V0.bas, written by Per Svensson
- xml28A1-ReadFusesWriteLockBits.bas based on user reinhars code and various DMA samples form user MAK3
- insertchar, delchar and delcharS statements added, seel also del\_insert\_chars.bas
- version(2) will show \$version info, see also new \$version
- Lookup() with an index crossing a 64KB boundary would fail for chips with >64KB.
- search dialog history can be cleared. Also removed the autocomplete.
- DMA support added to Xmega,also DMA demo added, int added to dat files.
- ST7565R lcd driver added for graphical displays, also xmega sample added for this lib (display uses 3v3)
- using {034} and quotes gave a problem when they were unbalanced.
- stk500 did not load the address for each 64 K page.
- fixed bug in the new lcd4busy\_anypin.lib, pin def was missing an underscore
- the flash2ram asm code checked page boundaries but this was also done by `_lpmbyte`. This resulted in a wrong rampz page when using READ on a page boundary.
- shiftin/shiftout accept a variable for nr of bits
- version() accepts user version string for example version("my data")
- chips with movw instruction will use movw if defined in the dat file
- config com2 stopbits=2 did not set the usb bit
- PORTB = Not DDRB (using not on a port with both the source and target a low port registers) would load the wrong register.
- assigning a string constant with a dot caused a wrong conversion of the string table.
- toggle word/long.bit toggled wrong bit
- When using \$SIM, i2c clock stretch is ignored in the simulator.
- \$file = "somefile.bin" will create different file output. if you main file is main.bas all output files will be somefile.xxx. The main.bas will be unaltered.
- config printbin=extended, fixed for xmega in xmega.lib
- wdvalue=wd timer values, added to the dat files. selecting a wrong value will give an error.
- enable interrupts will auto enable MED interrupts on the XMEGA. This will only happen the first time you use enable interrupts. It will use these settings : CONFIG PRIORITY=static,vector=application,MED=enabled
- varsize("varname") will return the length of the variable in bytes. example:

```

dim s as string * 10 : const c1=varsize("s") : c1 will become 11 (strings have additional byte)
dim b(4) as byte : const c2=varsize("b") : c2 will become 4 because byte uses 1 byte, and array is 4
varsize is an interal function, intended to be used with CONST.
- left() bug fixed when source and target where arrays with variable index. this would trash the registers that holds the lenght

2.0.5.0
- simulator tooltip will show alias and constants too. (if compiled with 2050)
- xmega buffered input and output were not using the buffers.
- tcpip chip reset time extended to 200 ms.
- ml64P,324P and 644P did not have uart2=5 in the ini file, so UBRRHI was not set.
- changed help about LIBUSB. Most important : new libusb supports all windows versions including Win7-64 !!!
- str2digits added, converts a string into an array of numeric digits
  s="12345" : str2digits s, ar(1) : ar(1) contains number of digits, ar(2) = 5, ar(3)=4, etc.
- XV-GDM128X64 Graphic Display Library for 128 * 64 from avr-praxis added.
- See http://mikrocontroller-praxis.de/de/Display/Monochrom-Grafik-FSTN/XV-GDM128X64-RGB.html
- Arduino bootloader support added. Select 57600 baud and the com port. flash/EEPROM is supported, but arduino does not support fuse/lock bytes.
  Tested with arduino mega and arduino nano v3.
- STK600 now also works in ISP mode for non xmega/pdi chips.
  also, the clock frequency was not calculated properly for the stk600.
- there was no error when assigning too big string constants to strings in subs/function.
  if you do not specify the size of a parameter, you still will not get an error.
  declare sub test(s as string * 10) : here the length is defined. this is the maximum allowed length.
  since you can call a sub from multiple places/subs, you need to define the maximum.
- TIMER4, TIMER5 supported added to the LOAD statement.
- ml280/2560/644p,m2561,m1281 did not use the proper register when using START TIMER.
- xmega 256xxx chips bootloader space was not addressed with the proper page loaded in rampz.
- dword type added. a dword is a double word and uses 4 bytes just as a long.
  just as like a word, it can not hold/show negative numbers. Notice that this is a first release, some function/statements might not support it.
  Inform support if there is a problem.
- at90pwm216 dat file added.
- when a string is passed with byval, a string with the required length is passed.
  So when passing "abc" it will pass a string with length of 3.
  When you alter the string, like adding spaces, you will overwrite the length.
  There is no way to know how long it can be/become.
  To override this, you may specify the length of the string in the declaration.
  normally you declare only the string type : dim test(byval s as string)
  But an optional length will set the maximum size.
  dim test(byval s as string * 20) will set the string to a length of 20.
- local variables may now have the same name as global variables.
- simulator changed so it requires less cpu time when paused
- charpos() used with an offset of 0 did not work. also new option/parameter added.
- quote() did append to the end of string.
- programming xmega 256 chips, programmed the full memory, not the project size.
- checksumXor() added
- simulator support added for config base=0
- xmega printbin did not support printing constants.
- xmega config print/2/3/4 for RS485 support added.
- if using WR pin with LCD in pin mode, the lib lcd4busy_anypin will be used. This lib is based on Luciano's LUC_lcd4busy lib.
  the lib will be automatically used when using the WR pin, so there is no need to include it.
  Example : Config Lcdpin = Pin , Db4 = Portb.2 , Db5 = Portb.3 , Db6 = Portb.4 , Db7 = Portb.5 , E = Portb.1 , Rs = Portb.0 , Wr = Portb.6
- sort added for byte, integer and word arrays
- modded rnd() so it returns a larger number(a word)

```

- \$crypt option added to use crypted code, this is an add on, not enabled by default.
- m8515 prescale value for 64 and up was incorrect.
- tiny48/88, timer0 configured non existing register. also removed all options that are not present in timer0.
- tiny24/45/85 Config timer0 = timer , Prescale = 1 , Compare A = Toggle , Clear Timer = 1 would set up portb.3 instead of portb.0
- simulator supports xmega interrupts.
- \$notransform option added. while the compiler by default changes code depending on the chip, you might want to get an error.  
When you set \$notransform ON, the compiler will not transform IN into an LDS when needed. You will get an error. Intended for asm programmers.  
\$notransform On  
cbi portL,1  
\$notransform Off
- avr-dos changed to xmega.
- fm24c64\_256.lib added for replacing eeprom with ramtron eeprom.
- glcdEADOGM128x6 , rotate option added by MWS to rotate screen by 180 degrees. use CONFIG parameter ROTATE=180
- glcdRA6963.lib added for RA6963 chip which replaces the T6963C
- AVR910/MYAVR MK2 programmer, blockmode support added for quicker programming. Also timeout added to options.
- \$bigstring gave a problem with big dimensioned strings because of the stack frame correction
- improved stack clean up code, depending on the size different code will be called eliminating the loading of R25 in most cases
- config BASE will create \_BASE constant. Also changed compiler and a number of libs to use the proper base. Code using elemtn (1) was replaced by (\_BASE)
- fixed a bug in glcdEADOGM128x6.lib where big non-TT fonts were not displayed correct.

#### 2.0.4.0

- \$include did not work for network/different drives.
- increased flash size for xmega with extended bootloader address
- config event\_system added for Xmega
- new xmega samples from MAK3 added
- getrc5 supported by xmega
- xmega CONFIG POWER\_REDUCTION added
- changed config port0 to vport : config vport0=A, vport1=B,vport2=., vport3=..
- config error=ignore,nr=ignore [,nr2=ignore] will ignore errors. this will allow you to compile your code despite a possible error.  
depending on the error, the resulting code will (not) work.
- buffered serial output (com1-com4) supported for xmega.
- freeing a frame with more then 255 bytes resulted in a frame space leak. (for example when using a big string as a local in a sub/function)
- simulator improved for xmega
- xmega ERAM can be assigned directly with a constant too
- xmega SPI print improved : using a comma you can specify the number of bytes to send  
print #1, ar(10), 10 ; 123 ; L,2  
would print 10 bytes of the array ar(), then a byte with value 123, and then 2 bytes of the variable L  
values are separated by semi colon (;) and the comma (,) serves to specify the number of bytes to send. the number of bytes may be a constant or variable.  
when no value is specified the data length is used. so for a long , 4 bytes would be sent.
- xmega SPI INPUT improved :  
input #1, var ;1 , ar(1);3  
would get 1 byte into variable var. and then 3 bytes into array ar(). Notice that INPUT uses the , as separator and INPUT the ;  
Thus for specifying the number of bytes, the ; is used. This is reversed in PRINT and INPUT.  
Normally the PRINT uses ; to separate and INPUT uses , to separate.
- Arabic translation for IDE completed. Also added BDI right to left support for Arabic. Thanks Walid !
- sample for xmega reset
- set,reset and toggle use special xmega registers for less code
- Russian translation for IDE completed. Thanks Alexander !
- array index calculation generates unneeded code for some cases. It adds a zero index which does harm but creates more code.
- defword and defbyte generated an error after an update of the parser.

### 2.0.3.0

- mmc-xmega.lib added for using hardware SPI with xmega.
- xmega int() function did not work since it used indirect addressing mode for registers.
- xmega SPI print/input did not use the right counting register, resulting in random read/writes.
- pinout files added
- label and variable names must be unique. this because a function like lookup for example can work with variables as well previously the variable would be used if both a label and variable existed. now you will get an error and you need to change the name of the label or variable
- updated libusb info in the help
- xmega virtual portmap config
- updated xmega pwm samples and config tcx will set output pin to output in freq/pwm mode when the CCx is enabled
- STK600 PDI programmer did not identified the chip.
- JTAG ICE programming fixed for both USB and COM port. Programmer works only in JTAG mode.
- bit compare bug from 2.0.0.0 fixed
- xml28a3def.dat , tcl added
- Open File has a sub menu so you can open a file from the samples folder directly.
- bookmark images show larger number for better readability

### 2.0.2.0

- start/stop timer accepts xmega timer names
- Full IDE is translated into Russian.
- config tcXX added for xmega timers
- config acxx added for xmega comparator
- simulator can simulate projects with up to 35 include files.
- dat file for ml6U4 added.
- all missing dat files added for Xmega
- loadadr did not took config base into account leading to index error warning for config base 0.
- atmel programmers can load/save in HEX format too.
- ADR2 bug fixed. In 2.0.0.0 the ADR2 could point to the wrong label.
- stk600-PDI support added. When selecting stk600, PDI will be used (for xmega programming)
- \$prog accepts lock and fuse bytes for xmega
- options, simulator added to save eeprom state.
- checkfloat() function added by Josef which can check a single or double and returns status about nan,zero,sign and infinity.
- pulseout changed to work with xmega.
- mcs bootloader new reset option. you can send a string for reset. To send special ascii use {}. {027} will send ascii 27. To send { you have to use the ascii value as well since { is used to send special characters.
- mcs bootloader can use DTR or RTS for the reset.
- xmega gosub=sectic did not set \_SECTIC constant.
- xml28 bootloader rewritten for XM32A4, by MAK3
- config DACB , configured DACA.
- split() when array was dimmed the same size as the number to split, the returned count was 1 too high.
- alias bug from 2.0.0.0 fixed.

### 2.0.1.0

- lwire changed so it works with Xmega as well.
- various samples in the samples\xmega folder
- bug in \$swcheck fixed. the soft stack is now checked from within the other stack check routines which saves code.
- xm32A4 added, thanks MAK3 !
- getadc() for Xmega has an additional parameter so it can pass both channel and mux.
- \$NORAMPZ will result in rampz treating as 1.12.0.0. In 2.0.0.0 the rampz is cleared after it has been used for pointing to flash memory. If your app worked fine and you can not allow the extra code you can use the directive.

- xmega EBI(xram) support added. See config XRAM. Don't be intimidated by the number of options, just use the sample.
- config clock uses the RTC now of the Xmega. See the xml28-RTC.bas example
- watchdog support added for Xmega. See the sample xml28-WD.bas
- added DISABLE JTAG to disable the jtag interface via software. this works for normal and xmega chips.
- \$forcesofti2c will force the xmega to use software i2c, it was used internally. it will not be added to the help.
- adding array index support feature resulted in a bug when using locals
- lcdvdf.lib added for a special VFD display module from "Electronic Design Bitzer". LCDAUTODIM added.  
this is a 20x4 vfd display immune for ESD.
- config lcdpin creates constants for the pins which are used by the lib. also MODE added for 20x4VFD.  
you can also define busy for a busy pin. this is a fragment from a lib that shows the generated names. they are also visible in the report.

```
sbi _lcddb4dir, _lcddb4_pin      ; db4
sbi _lcddb5dir, _lcddb5_pin      ; db5
sbi _lcddb6dir, _lcddb6_pin      ; db6
sbi _lcddb7dir, _lcddb7_pin      ; db7
sbi _lcddbedir, _lcddbe_pin      ; enable/strobe HIGH
sbi _lcdresetdir, _lcdreset_pin  ; reset line to Output
sbi _lcddbe, _lcddbe_pin         ; strobe high
cbi _lcdreset, _lcdreset_pin     ; reset low
sbi _lcdreset, _lcdreset_pin     ; reset was LOW so make it HIGH
```

- when not ubrhi setting updated the wrong register for chips with shared registers.
- config DP="," | "." to select the decimal point for the fusing() and single str() functions. Default is "."
- i2crepstart did not work for Xmega
- i2cstop for xmega did not free bus after i2creceive/i2csend, so the i2cstop command has been extended to send multiple commands.

#### 2.0.0.0 , yes this is the new version number

- Josef rewrote the single power() to fix a bug with the sign. This new code does use more code.
- RAMPZ : when Z is loaded, the RAMPZ is set when it was used before to point to data in flash memory.  
This will generate more code but is needed because when RAMPZ is changed and data is exchanged between X and Z, it is important that Z points to the proper data in RAM.
- readsig() added for xmega, use it to load calibration data
- beta : added stackcheck directive \$hwcheck,\$framecheck and \$softcheck
- xmega buffered serial input supported
- xmega config adc did not support CH1\_GAIN - CH3\_GAIN parameters.
- split() supports variable for the delimiter too now. (previously it only supported a constant)
- declare function/sub REQUIRE (). So declare sub test is NOT ok, but declare sub test() is.
- Xmega AES encryption/decryption added. See xml28-aes.bas
- xmega TWI support added. By default TWIC will be used unless you use OPEN "TWIx"  
config TWI will enable the TWI so it is important to add it : CONFIG TWI|TWIC|TWID|TWIE|TWIF=baud  
open "TWIC" for binary as #1  
i2cinit #1  
i2cstart #1  
i2cwbyte value ,#1  
i2crbyte value,ack|nack,#1  
i2cstop #1  
without # the default TWIC will be used.
- various xmega uart bugs resolved (some could lead to a stack error)
- \$bgf had a problem with big images. this was actually a problem of the assembler. this could give problems for large font files too.
- Xmega baud rate calculation improved, now the lowest possible error will be calculated for BSEL/BSCALE.
- config servo has a new parameter and alternative code for working with servos. when using INTERVAL=t , then the alternative code will be used.  
It is best to use a reload value of 100 and an interval of 100. This is optimized for normal RC servos.
- instead of ar(a+1) you can also use a variable : ar(A+J)

- bylabel added to sub/function which allows to pass the address of a label.
- uarts of type 3 (shared UBRI register) did not set UMSEL bit resulting in bad data bits. using CONFIG COM did/does work correct. This problem was only visible when
  - using the old style where the com port is automatic configured when using \$BAUD.
- soft spi now supports mode 0,1,2 and 3. if not specifying mode=, then the default mode 1 will be used.
  - when using the MODE, you can also specify setup and speed. setup and speed expect a numeric constant.
  - setup is the delay in uS before sampling the MISO pin. speed is a delay in us. when you specify 1, it will result in a 2 uS or 500 KHz clock.(estimated)
  - you can use either SS or NOSS, NOSS and SS are inverted. to disable SS use : SS=NONE, or NOSS=1
- toggle for arrays with constant bit number worked on element 1.
- DIDR reg added to tiny2313
- mcs bootloader can send EEP file too if it exists.
- accessing an array in a sub/function with a local var and offset (var(a+3)) did not added the offset to the element.
- m48Pdef.dat file added
- getrc5 , use 2000 for the 131 ms wait as it was in 1.11.9.8
- config print2 end print3 did not set rs485 level for the PRINT statement.
- the idle, powersave and powerdown are deprecated. use power [option].
  - the old statements will convert automatic to the new power statement.
- ml284p added to config clock.
- tiny13A dat file added. Notice that this A version has the same signature as the normal Tiny13.
- in options, enviroment, IDE you can select to check the version when you start the IDE
- tiny25,45 and 85 did not had a timer1 reference in the dat file (alias for tcnt1)
- crc32 uses a word for the number of bytes instead of a byte
- the simulator saves the state of the EEPROM. This means that when writing the EEPROM, you
  - Use the popup menu to clear the EEPROM (set all values to FF)
- uspasp and other atmel programmers did not erase the chip when using 'autoprogram'. Also auto refresh for lock and fuse bits was not enabled.
- stcheck used brge(signed compare) instead of brsh(unsigned compare)
- tiny48/tiny88 do not support HW MUL, dat files adjusted.
- mmc.lib saves R23 in case an extened port is used for the interface.
- charpos() accepts additional parameter with starting value. (>0)
- ##### IMPORTANT FEATURE #####
  - config base=0 | 1 added. By default arrays are 1 based in bascom. with config base=0, arrays start a 0.
  - dim ar(5) will occupy elements 0-4 ! In some old basic(QB) and VB it would take 0-5 thus 1 additional element.
- config lcdpin did not override options from the CFG file when bus mode was selected.
- waitos accepts a variable too. it will only work reliable when the oscillator speed is high enough.
- config i2cdelay did not update \_\_\_I2CDELAY constant used by soft i2c.
- mcs bootloader will ask if EEP file need to be programmed when this file exists.
- max() procedure (not the function) returned wrong index value for bytes which had the ms bit set(signed compare was used instead of unsigned).

#### 1.12.0.0

- Xmega caused a bug in non-xmega timer handling : non xmega have a normal abd absolute address for IO regs which differ by 32. (normal registers space).
  - This offset was determinted too late, resulting in a wrong address for internal variables like timer0.
  - These special byte variables are now placed in the dat file.
  - This error was not visible when you compiled your code more then once! Only a fresh start and first compilation would reveal this bug.
- swap supports swap of double too. Also, swap with a single parameter can swap a nibble or 2 bytes:
  - swap b where b = &B11110000 will result in 00001111
  - for word/integer, it will swap the bytes : w=&B11110000\_1100\_0000 -> 1100\_0000\_1111\_0000 or &HABCD becomes CDAB.
- bccard port parameter changed into full port name. See help.
- config twislave has a new option : userack=ON|OFF. When on, you can set ack/nack from your code. Also fixed a bug where the default setting did not save all registers.
- m32U4 : getadc() did not used ADC\_MUX value from the dat file.

- myAVR programmer, changed error messages to info msg.
- settings : 19200 baud.
- internal assembler allowed a backwards jump of 1 too many. in glcdeadogml28x6 this caused a wrong jump for chips>64KB.
- using \$noramclear in combination with \$initmicro could give an memory access error(invalid pointer).
- m32U4.dat file had missing mux5 bit, also corrected interrupt section.
- Windows 7-64 bit support added for LIBUSB.
- crc16 change for word sized arrays did not work for words but only bytes.
- internal loadbit routine which was rewritten in 11199 to check for boundary errors, had a bug for arrays with a constant index. The offset was 1 off.
- this means that sombit=ar(3).8 would load the wrong bit location.
- RTF export will place the content in the clipboard too.
- power mode bug fixed: SE was not set when none of the SM bits were 1.
- usbasp (and other atmel programmers) gave a division by zero error for AT90XXXX chips (chips that do not use page programming)
- replaceChars string, oldchar,newchar will replace all characters in string.
- added tiny4313 support(programming untested)
- added button to insert \$regfile, and \$stack to the editor

#### 1.11.9.9

- fixed bug for SEROUT where using a variable for the number of bytes would overwrite the X register
- right() function used with a variable, and without \$bigstring could lead to a pointer error, resulting in wrong result.
- Xmega SPI support added. SPI for xmega also supports dynamic channels. The variable has to start with bSPI in that case:
 

```
Open "SPI,Master,mode0,clk2,msb" For Binary As #bspivar      ' use a dynamic channel
Print #bspivar , W
Input #bspivar , W
Instead of a CONFIG, OPEN is used. Notice that you have to open
```
- Xmega dynamic UART channel added. When using a variable that begins with BUART, you can use a dynamic channel:
 

```
For Buart_channel = 0 To 7      'when using a variable, notice that the index is 0 based !
Print #buart_channel , "UART :" ; Buart_channel
waitms 1000
Next
```

The channel does not to be opened in this case.
- \$eepromsize added : \$eepromsize=value. This directive can be used in cases where you use an alternative eeprom library. for example to use a quicker and more reliable eeprom such as an external ramtron chip. Using an alternative library will use an external eeprom instead of the internal eeprom.
- But in cases where your external eeprom is larger then the internal, you need to override the size. You can do this with the \$eepromsize directive.
- You have to take in mind that programming the eeprom will program still the internal eeprom ! So use these options with care.
- when using ISIS simulator, the IDE will set focus to ISIS when pressing F2.
- config rc5 , wait feature documented. default is 100 (6.4 ms)
- crc16() will pass a word instead of a byte. this means you can get crc16 of a bigger array.
- config kbd has an additional parameter named COLS. By default this is 4. Some chips do not have port pin 7 and for these cases you can use COLS=3, or COLS=2
- This does assume that columns are connected to the high port nibble.
- writeeeprom/readeeprom when used with a constant address did not store the address when used with auto increase:
 

```
writeeeprom var, 1 : writeeeprom var
```
- split() function did not return empty array elements. Those were ignored:
 

```
consider this string : "1,2,,4,5" . it would return 1,2,4 and 5. Since there is a delimiter, it is better to return an empty element.
```
- megal284P, M644P dat file , stk500 eeprom section corrected.
- tinym61 have a 16 bit timer0 but the LSB and MSB registers are not linear in memory. If used in 16 bit mode, you need to handle this yourself.
 

```
reading : read LSB first, then MSB. Writing : write MSB first, then LSB. (does NOT apply to Xmega)
```

- DMXSLAVE needs aliases for UCSRx registers. These were added to the DAT files.
- PORTA, DDR and PIN registers added to tiny48 and tiny88 since TQFP version has this port.
- M32M1 support added. Notice that it does not have TWI or a UART. The LIN can be used in UART mode but is not supported yet.
- [FLAG] section added to dat files. Some micro's have the flag register located in non-bit addressable space in such a case, SBI will not work to reset the bit and OUT or STS is used. For FLAG registers, the current content may not be read. While for normal registers, when setting a bit, the content must be read before the bit is set.
- mega3250P support added.
- CONFIG INPUT defaulted to the default settings. this was caused by modification for the xmega. Also notice that CONFIG INPUT0 need to be specified as INPUT or INPUT1.
- USBASP programmer problem fixed with internal memory pointer(resulting in wrong flashing)
- dat files include info for new IDE, same dat file set will be used.
- CLOSE for SW UART explained in more detail in the help.
- print preview and print use the same font settings now.
- stk500 external programmer(stk500) was called with wrong COM parameter after higher COM ports were added to options. In this change the numbers where changed into COMx.
- tiny2313 dat file had a type error for IDLE mode.
- tiny48/88 TQFP32 pinout xml files had a few errors
- instead of config lcd=16\*2 you can use X instead of \*. This is needed for beta IDE. Example : config lcd=16x2
- CONFIG DATE needs "" for the separator : config date=dmy,""

#### 1.11.9.8

- #AUTOCODE, #ENDAUTOCODE and #DEFINE added
- xmega inkey() for COM4-7 did not work.
- LCD simulator did not clear CGRAM.
- an update in a component caused the atmel programmers, autoprogram option to fail
- RND() function improved so it wont give the same sequence
- xmega uart support created bug in chr() function for channel 1.

#### 1.11.9.7

- This is a new Xmega update but also included some fixes and features for non Xmega.
- CONFIG ADCA of CONFIG ADCB added for ADC converter
- config INPUT|INPUT1|INPUT2... INPUT8, is supported by Xmega
- all uarts are supported now. use COM1-COM8.
- While all UARTS are supported, some options are not implemented such as buffered serial COM.
- new power mode support. the IDLE, POWERSAVE and POWERDOWN are replaced by POWER MODE
- the old modes still work but will be removed in a future version. It is advised to use the new POWER command which supports all modes.
- The supported modes depend on the micro hardware.
- For example in the M88DEF.DAT for the Mega88 you will find :  
Powermodes=5  
PM1=Idle,SMCR.SM0-0,SMCR.SM1-0,SMCR.SM2-0  
PM2=Powerdown,SMCR.SM0-0,SMCR.SM1-1,SMCR.SM2-0  
PM3=Standby,SMCR.SM0-0,SMCR.SM1-1,SMCR.SM2-1  
PM4=ADCnoise,SMCR.SM0-1,SMCR.SM1-0,SMCR.SM2-0  
PM5=Powersave,SMCR.SM0-1,SMCR.SM1-1,SMCR.SM2-0  
In this case the modes are IDLE, POWERDOWN,STANDBY,ADCNOISE,POWERSAVE  
Use as : POWER IDLE  
Other chips might have more modes.
- m328P has some fusebits on a different fuse compared to m88, m168p. this is now corrected.
- number of include files increased to 500. It was actual 500, but a type error resulted in 50 instead of 500.
- a bug in an update of a third party component caused an error on windows 2000.
- portQ, portR added to CONFIG for Xmega

- added COM5,6 and 7 for Xmega. Notice that for Xmega you can dynamic change the COM at run time.
- CONFIG COM for xmega changed : instead of using COM0 it is better for compatibility with other commands to start with COM1  
So the range for xmega is now the same as for normal AVR : COM1-COM8  
Internally, the uart channel is working with 0-7.  
The TX pin is set to output in this release.

#### 1.11.9.6

- Elektor ATM18 LCD driver added. Use \$lib "lcd4-elektor.lbx". See also sample elektor-lcd-final.bas

### XMEGA NOTES ###

! In XMEGA the memory is not linear, you can not access the registers with a pointer! registers are in a separate dataspace

! In XMEGA the access order of 16/24/32 bit registers is changed. It is now from LSB-MSB for both reading and writing !

Following comment is for XMEGA only. Please look in the help for a full description.

- CONFIG COM starts at 0 for xmega , other parameters differ slightly too  
Config Com0 = 38400 , Mode = Asynchronous , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
- When using EEPROM, you need to add : Config Eeprom = Mapped
- CONFIG DACA|DACB : Config Dacca = Enabled , Io0 = Enabled , Channel = Single , Reference = Intlv , Interval = 64 , Refresh = 64
- Config Osc = Enabled , 32mhzosc = Enabled
- Config Sysclock = 32mhz , Prescalea = 1 , Prescalebc = 1\_1
- Config Priority = Static , Vector = Application , Lo = Enabled

Notice that Xmega support is not fully implemented. But it is well enough to use it and get familiar with it.

In the coming period there will be a number of xmega updates that will implement all 8 UARTS, ADC, multiple SPI , TWI and timers.

- Notice that while JTAG programming is supported now, it does NOT support Xmega JTAG. Atmel has changed the protocol and they could not give an update yet that describes the changes related to XMEGA. So JTAG will work for non-xmega but not for Xmega.

We advise to use stk600 programmer and load the/a bootloader.

- close (for UART) is now ignored. close was never needed but when you used it, it had to be done after code that was using the UART (last lines of your program).

now it does not matter anymore where you put the close, since you can not reuse a SW UART, it is ignored

- Box in glcd.lib had an error for filling the complete box
- glcd showpic command has been changed so it will not use R23 for retrieving the data. This was done for cases where the control port is an extended port (portF on M128)

and these chips can not handle all asm commands. R23 is used in such cases.

- attiny88 support added. notice that this is not a good replacement for Mega88 since it has less memory and NO UART
- attiny48 support added, see sample file.

- usb add on updated with mouse sample and some mods for the new M32U.

- M32U (m32 with usb) added.

- using NOT on a byte, indexed as a bit, and assigned to a bit, was processed as byte.

- M64def file did not had xram setting info.

- Print with a large number of parameters could lead to a crash.

- bccard lib adjusted for ZC312 cards. Notice that these cards need some delay between 2 BCCALLS.

- tiny261,461 and 861 had a type error for the LKJ-110=Brown Out 1.8 V entry (the - was an = by mistake)

- USBASP programmer support added.

- added relative to other asm mnemonics too (brcc, brcs,etc)

- Charpos() function added which is similar to INSTR() but searches for a single digit and is quicker and uses less code.

- you need to select COM port again for STK500 native programmer. (the way it is stored has been changed)

Also added option to skip the setting of the ISP frequency, this is done for incompatible programmers.

You can enter any clock freq now. when it is below 4000, 4000 will be used!

- Added myAVR MK2 programmer. This programmer uses AVR910 AN so will work for all AVR910 based programmers

- CONFIG EXTENDED\_PORT= WARNING | ERROR

by default chips that have extended ports can not handle CBI instructions. the internal assembler will convert the code automatic.

but for some instructions a register is needed.

This option is intended for asm programmers : you can get either a warning or an error.

- m88Pdef.dat : the megajmp value was wrong resulting in a wrong interrupt jump  
Also added reference for CONFIG CLOCK
- M124P,324P and 644P had TCCR2 mapped to the wrong register in the def file. it must map to TCCR2B
- \$BIGSTRINGS directive added together with bigstrings.lib. With this directive, you can use big strings. Big strings are strings that are more then 255 bytes and can be up to 65535 bytes long(if you have enough memory)
- This is a first release. The library should work on all string operations but was not tested extensively for all string operations. Please report if you find a bug.

#### 1.11.9.5

- added support for KS108 with 3 chips : 192 \* 64
- The timeout for the USB programmers that need LIBUSB can be adjusted in the options. Some USB1.1 hubs in combination with a lot of USB devices connected, need a bigger timeout. Default is 100 (msec). But you can increase in steps till it works properly.
- M128can and M32CAN used wrong timer2 definition when configuring the timer2.
- m165 dat file needed the TCCR2 alias for TCCR2A.
- attiny13 did not start when using START TIMER0
- generated files are in same case as the source file
- tiny25,tiny45,tiny85 used the wrong register when updating the prescaler
- simulator with real RS-232 data, did not set data receive interrupt. So when using buffered serial IO, it would not be read.
- cls LINE for KS108 did not worked for the entire line(different chips)
- tiny2313 compare config for OCOA did not send portb.2 but portd.6
- \_\_LTMPC was created twice. Does not have impact but looks odd in the report.
- .DEF will now report error 358 if a def is used twice
- M88PDEF.DAT had a wrong interrupt table(wrong address), WRAP value was wrong too.
- Instead of a label or number, you can also use relative jumps + will jump forward, - backward  
rjmp +1 ; address +1 rcall -2 ; call address at address -2 rjmp 0 ; goto RESET address !
- display accepts 3 parameters : DISPLAY ON|OFF , CURSOR|NOCURSOR , BLINK|NOBLINK  
while you can also use cursor for this, the cursor commands uses an internal variable, which in some cases can get confused.  
The display statements will write the value that reflects the parameters and does not use an internal variable to maintain the state.
- M124P,324P and 644P had TCCR2 mapped to the wrong register in the def file. it should map to TCCR2A and not to TCCR2B
- M48,88,168 did not used AS2 bit to set the Async mode

#### 1.11.9.4

- m32CAN.DAT added (AT90CAN32)
- KS108 lib added from Nard Awater that supports 128\*128 KS108 (4 chips)
- \$projecttime added
- fixed some problems to
- toggle had a bug for chips with extneded port on a high address where STS needed to be used. When SBI/SBI on the PIN register could be used there was no problem, but the m2560 has a few port on an extended address. in such cases, the register was read and altered, resulting in some other pins to toggle!
- Config Printbin = Extended : fixed a bug,because of a wrong register def, wrong register was used.
- at90128CAN, config clock set the wrong timer
- config dcf77 : when using update, following error was reported : updatetime changed. This was due missing remark (;)
- split() function bug fixed. when array element was full and next character was delimiter, a new index was added.
- atmega1284p support added. sample m1284p.bas added.
- DMX receiving added. See dmx-receive.bas and CONFIG DMXSLAVE
- changed SETFONT. For chips with more then 128KB, there is no room for the full address.  
beside the internal variable \_\_FONTTABLE which contains the address of the font, the variable \_\_FONTTABLE\_H (BYTE) is used to store the 64KB page number.  
When you made your own LCD driver you need to change the \_gwrite\_lcdchar routine :
- \*#IF \_ROMSIZE>65536

```

; ----- REMOVE THESE LINES -----
;* Clr r0                ; temp reg
;* lsl R30               ; shift to left
;* Ror r31               ; rotate left with carry
;* Ror R0                ; MSB of address is in R0 LS bit
;* Out RAMPZ, R0        ; set the page
;----- ADD FOLLOWING LINES -----
* Lds r0,{__fonttable +2} ; load page number
* Out RampZ,r0          ; set rampz register to load proper page
*#ENDIF
This code is only used for micros with multiple 64KB pages such as the Megal28, Megas2561, etc.

```

- bug : function warning also warned for subs
- megal68P support added. sample ml68p.bas added
- Mega88P support added. sample m88p.bas added
- m325def.dat : some aliases added for changed register names. (timer0,timer2)
- usb-isp programmer : when there is no EEP file you can not change/upload the EEPROM image that is manual modified. You need an EEP file. With this modification, you do not need an EEP file in order to change the EEPROM of the chip.
- fixed i2c\_extended.lib. This lib is intended to be used on extended ports(port with an address>&H3F) , but it failed for these extended ports. This lib is NOT available in the demo.
- when not using the config SPI, SPIIN option, SPIIN will behave as in version 1.11.9.0 where SPIIN will send data from the variable. See the help.
- new Elektor programmer support had some issues. Also renamed to USBprog since it is well known by this name.
- Vista 64 driver for USB-ISP added. Thanks to Martin Verschuren for the docs and INF update.
- box/boxfill had a bug in glcd.lib that could lead to a SW stack overflow.
- Kamami, 'KamProg for AVR' support added
- PCF8833 color LCD lib modified to support chips > 64K

### 1.11.9.3

- ```
# VISTA 64 NOTE : either run as admin, or put the sample files in the document folder.
```
- The Program Files folder does not allow changes.
  - CONFIG SPI has a new option. Some SPI chips like to have sent a value of 0 when reading from the SPI. Others seem to require a 0. The CONFIG SPI is enhanced with the option SPIN. SPIN = value. By default, the value is zero.
  - updatewiz 1.0.0.17 will backup ALL files in case you have selected to backup before you update. the unzip process is also revised to be quicker.
  - Max() function for words did not returned the right result.
  - mkII USB programmer support added. It has the same interface as the new stk500 interface.
  - stk500 support added. this is a native programmer which does not need stk500.exe It replaces the stk500 V2 which used stk500.exe see for more info the Help (stk500 programmer)
  - function warning : STATEMENT NOT ALLOWED AFTER FUNCTION ASSIGNMENT, see FUNCTION in Help.
  - at90pwm2\_3.dat file, WDTCR added.
  - usb1287 timer mapping corrected. (start/stop timer0/timer2)
  - Quote() function added. var= quote(string) results in "string". Usable with HTML pages.
  - \$LCD and \$LCDRS will have an automatic constant named \_\_LCDRS\_ADR and \_\_LCD\_ADR, which can be used in your own lib
  - tiny84 had wrong page size, resulting in programming problems
  - loading a bit from a constant variable data type bigger then byte, did not used the bit offset to load the bit :  
portd.6 = someWord(1).15 ' failed to load the bit from the proper byte of the word
  - parser of the IDE failed for &H with \$hwstack,\$swstack,\$framesize.
  - all word registers are placed in a [WIO] section. These are all written with MSB first, LSB last order.  
A number of aliases are added to. For example TCNT1 will have an entry named TIMER1 too.
  - optimized number of instructions in byte handling.
  - OUT instruction accepts port variables too. But take care, read the HELP.
  - 2343def.dat timer0 alias added

#### 1.11.9.2

- M328P supported added
- 192\*64 graphic lcd added
- added split screen to editor.
- m644 had an error for INT2 EICRA bits.
- FLIP support added. See also flipDLLcopy.cmd
- power() function for doubles, returns the same result as Excel now. Previous it returned the same result as QB/VB which is wrong for negative numbers.
- resource add on added. this is an add on option, not available by default. See \$resource
- flip USB programmer support added for supported USB chips
- maximize all windows option added
- added support for config clock for M48
- config USB in help contained wrong link in index.
- pdf update has a cancel option now
- reference added to rampz for > 64KB chips
- new shiftin variant added. see config shiftin and shiftin
- SW UART supports PORTA-PORTL now (for m2560 and similar)
- wrong tab was painted red when code was modified
- attiny25/45/85 had the wrong A/D reference ADC\_REFMODEL. Changed from 7 to 9
- attiny/24/44/84 did not had WDTCR register reference (WDTCR)
- serin/serout did not work with high extended port registers found in m1280/m2560 like PORTL.
- glcdks108.lib had a problem with crossing 64K boundaries in M128 and other >64KB chips. rampz could have the wrong value when font data was shown. resulting in displaying odd data on the screen.
- added support for m644P. (picopower)
- CLS [option] failed while introducing new option for CLS for EADOGM128
- \* note added to help for getadc() value.
- using CONFIG COM2/3/4 multiple times, incombination with buffered serial input/output, cleared an int. flag first occurrence of CONFIG COMx does not preserve the flag, all others will do.
- analog comparator has 2 inputs now in sim so anin1 has no fixed level but you can adjust it
- help button added in sim
- makemodbus() function did not set R25(address of variable) causing it to fail the second time it is called
- getkbd() was not simulated correctly for all chips

#### 1.11.9.1 - m640 support added.

- TAB interface added to IDE.
- Modbus master function added : makemodbus(). This function need to be used with PRINT. See help for more info.
- double and single libs \_Adjust\_RAMPZ\_Down fixed. Could only give a problem on biggest micro with all flash used.
- Added support for 'TrueType' fonts. The fonteditor is not changed yet! It will be enhanced and published at a later stage. It will only work for EADOGM128. This display uses the ST7565. Other similar displays could handle it too. Check out the code in the lib. The font needs a change for this option. When you open the font with notepad you will see something like :  
Font8x8:  
\$asm  
.db 1,8,8,0  
^^^ the last zero need to be changed into the char spacing. For example 2.  
Since the value is not used with other fonts it does not matter when you add it.  
The code that will display the character will check the left space and the right space of the character.  
For example the i will take less space then the W. So the character i and W bytes are :  
.db 0,0,0,125,64,0,0,0 ; i  
.db 0,60,96,48,96,60,0,0 ; w  
The code in the lib might be hard to understand but it need to check additional bytes for big fonts to.  
Essential the zero's are counted at the left and at the right. So an 'i' will take only 3 columns.

When you display 'iiiiiiii' it will take less space than 'wwwwwww'. It looks better.

One problem is when you update a location in a loop : 111 will take less space than 888.

So the previous data is not overwritten then. For this purpose the CLS statement was enhanced. See below.

- CLS accepts additional params for EADOGM128 : cls Y,X1,X2 [char]

It will clear a part of the specified line. The optional char is the reverse value to use.

If not specified, a 0 will be used which will blank the line.

- support added for EADOG128M graphical LCD. See EADOGM128.BAS sample.

Note that this is a serial display and it can NOT read data. Meaning that a number of graphical commands are not supported!

- bascomp.exe command line compile utility supports all DAT files now. Will automatic supports all future dat files.

- enhanced wd timer need to be cleared soon as possible. you can use \$initmicro to do. when the micro has an enhanced WD timer, it is always reset.

but when you need to determine the cause of the reset you need to check the MCUSR register. Since the WD overflow flag is cleared, you have to read this register with the \$initmicro option.

- the sim contained a bug which could lead till internal recursion, resulting in a stack overflow.

- ks108 libray supports INITLCD. Also timing changed to support high crystals like 16 MHz.

- AD converter code has been rewritten. While the first AD converters were simple, the latest ones split the MUX bits among different registers and add

other options as well. While the CONFIG ADC used to store the setting during compilation and this was used the load the ADCSR register, this has been changed.

Now it will set the ADCSR register and other registers(ADCSRB, ADMUX) once.

getad() will now pass the channel and set the proper mux bits in admux and adcsrB. The optional parameter is added to the ADMUX.

This means that when you used the additional parameter, you have to change your code. In most cases it is sufficient to remove the additional parameter and provide the proper channel value.

The dat file has a new setting : ADC\_MUX=6,ADMUX0-4,ADCSRB.3 ; 6 bits, reg-bits, optional other bits

This sample from the tiny261 informs that the muxbits 0-4 are in admux and that the upper bit is in adcsrb.

While we checked these new settings twice, we could not test it on real chips, it was a tested with real chips on a number of chips.

In case of a problem contact support and/or check the setting. Notice that the AD converter is started automatic with CONFIG ADC.

An example with temperature reading has been added (CHIPS\tiny261.bas)

- config lwire has optional param : [,extended=0|1]

when using multiple port pins on mixed ports, that have normal IO address and extended IO address, the compiler will choose one.

I would not advise using multiple ports for multiple lwire busses, but when you do that and the ports meet the above condition,

like PORTD and PORTL , you need the additional EXTENDED=1 parameter. This will instruct the compiler to generate the whole pin address (r30+r31)

and not the lower address which would be sufficient for normal ports, or for extended ports only.

- AN\_READHITAG\_EM4095.BAS sample added. this is the sample from the reference KIT.

- received tiny261 chip and added support. a sample program was added too.

- setting/resetting a bit of an array manipulated the wrong bit (when using SET/RESET)

- gps-distance.bas example added for calculating distance and course

- AT KEYBOARD EMU : some motherboard seem to require a reversed ID. Usually AB83 must be sent.

But these require 83AB. CONFIG PS2 has an additional option : ,init=&H83AB , which is the default.

To send in reverse order, use : ,init=&H83AB

You need the updated library which is available via the register.

- when using TOGGLE on a PORT pin, the PIN register will be set, resulting in a toggle of the PORT pin.

This will only work when the DAT file has the NEWPORT=1 option. Old chips do not support this option. In case of a problem you should use NEWPORT=0. The setting in the DAT files reflect the DATA Sheet. It was not tested with all chips, a random test was performed.

Because you can write to a PIN register now, the error check has been removed.

- SHIFT RIGHT has extra option to preserve sign bit. SHIFT var, RIGHT, noshifts, SIGNED

This option will preserve the sign bit and add it back later after the shifts have been performed

- SPIIN sends a 0 now when getting the value from the SPI-bus. This keeps MOSI 0.

- changed all dat files with ADFR (free running mode) since the start bit was absent

- tiny461/861 had wrong timer1 reference.

- single.lbx , \$external conditional compilation fixed

- \$external when used with conditional compilation was not working the right way
- tiny85 had wrong romize in the def file,also address wrap was not set.
- tiny84 had address wrap not set.
- crc16UNI() function added.
- crc8-16-32.bas changed. demonstartes crc16UNI() function too.
- usbl62 support added. See also CONFIG USB.
- simulator did not stopped correctly when interrupted by compilation
- the lcd-pcf8833 lib did not had a refernce to the needed \_BOX code.
- timer4 and timer5 were not added as 16 bit variables.

1.11.9.0 - added mega406 support( beta , limited tested).

- you might get errors (ERROR 353) compiling existing code. This for example when ADR or ADR2 which are now reserved were used as variables.

The basmon.bas sample for example has been modified. Some other sample programs are changed too. The problem is simple to fix by choosing a different name which is not reserved.

The same problem you get when you try to create a constant which is a reserved word.

- added support for EM4095 chip to read RFID. See ReadHITAG
- cpeekH() changed. You can specify an additional parameter with the page : var=cpeekh(address,page)
- twi master time out added. chip state machine could come in wrong state where it would return only F8. this would freeze the application.
- also help updated for twi slave, new images added, GENCALL option added to CONFIG TWISLAVE.
- i2cscan.bas sample added to scan the bus for devices and test the general call address.
- added simple macro support. this for upcoming USB support.
- macro do NOT support return param and params yet.
- added library for HTRC110 RFID chip. See ReadHITAG
- fixed HWMUL setting in tiny24,tiny44 and tiny84. These chips do NOT have HWMUL
- added BOXFILL command and BOX to the generic graphic lcd
- added support for SED180\*32 from EA. (3 SED1520 chips)
- added total PDF download. you need to add the button under the Tools menu.
- Added USB-ISP option to set the speed of the programmer
- you can update all PDF's from atmels site at once
- capture3 was not recognized as internal variable.
- pdf files can be downloaded now when they are not present in the PDF subdir.
- this specially for customers who download and do not use the CD-ROM.
- The files are downloaded from Atmels site.
- When the link changes, please send an email to support@mcselec.com
- added support for mega645
- added support for mega649

1.11.8.9

- getrc5 did not worked with optional timer2 on M48/m88/ml68
- the internal dat file checker was enhanced to check for some other potential problems(like constant with same name as register)
- simulator did not work for extended ports(port that have a higher location that fit into a byte).
- especial simualtion LCD connected to a high port (portH or PORTL) did not work.
- enable/disable will show all interrupts, depending on the used chip
- support for AT90PWM2 and AT90PWM3 added.(some advanced features are not yet supported with a high level basic command)
- Note that PWM2 and PWM3 chips have the same ID. So we have 1 dat file that you can use for all chips.
- The PWM2 is the same as PWM3 but with only 24 pins. Look at the at90pwm2\_3.bas sample.
- Getadc() will set adcsrb.mux5 bit for channel>7 on m2560, m1280.
- PDF can be used on dual/virtual monitors. The window is repainted correctly.
- ADR Label will create label data
- ADR2 Label will create label data multiplied by 2, needed to get the address in memory since word addresses are used
- These are ASM mnemonics. DATAPTR can be used to read the address:

```

Restore Label
Read DATAPTR ' read 2 bytes and assign to internal datapointer R8/R9
Read B ' read a byte
End
Label:
! ADR2 Testlabel
nop
testlabel:
data 10,20
This sample will first set the datapointer to address of LABEL
Then the ADR2 data will be read which points to testlabel
The next READ will thus read from label testlabel
See adr2.lib

```

#### 1.11.8.8

- lwire did not work on extended ports with address > &HFF. When the lib was written there were no high ports. The mcs.lib has been changed and when a port is used with high address, the internal constant \_LWEXT is set to 1. The mcs.lib checks this constant whether to set R31 to 0 or not. It does mean however that when you use a port with extended address, all lwire routines should work with extended address. For example PORTL is extended register. You can not mix lwire on PORTL and PORTB. While the compiler could also pass the R31 value each time it saves code to set the register in the lib. But it has the described limitation that you can now work with both normal and extended regs.
- Progy programmer support added.
- Started localization of the IDE. See options IDE to change the language.
- some of the dat file had a double entry for WGM01. This value had the wrong value.
- sht11.bas sample from Helmut Hoerschgl added that shows how to use the SHT11 with crc checksum.
- Simulator, COM port for simulation can be different than terminal emulator com port. Specify the COM port in Options, Simulator. Previous it would use the same COM port as the terminal emulator.
- IDE search is changed search. Hotkeys added to indent, unindent, remark Dropdown label and sub search will put the found item on top in the editor And the jump color was changed into red again. The green was an error from the conversion from D5 to D7.
- added tiny861.dat file with sample in chips dir.
- tcpchecksum supports 2 additional/optional word params. This way it can be used in all UDP/TCP related checksums. w=tcpchecksum(array(start), bytes [,w1] [,w2])
- ip\_checksum.bas added to Samples\TCP dir. It demonstrates tcpchecksum.
- added option to simulator to invert leds.
- the new color grid option in the simulator had a bug : you could not manual edit the register and memory window.
- ALIAS : added a check for alias with arrays. An alias can not be an array. ar(10) alias portd.0 is NOT ALLOWED. The alias must be a name without ().
- dat files M48 and M88 had a wrong INT vector offset.
- Pinout XML files for Mega161, Mega162, 4414, 4434, M165, M169, 90SCAN added
- single.lbx for scientific notation was not recompiled.
- vartest("test") added for conditional compilation. This will test if a constant exist and then perform the test.
 

```

exampele : #IF vartest("myconstz>10")
This will test if myconstZ > 10. The test will return false in case the constant does not exist.
It will NOT give an error. So the advantage is that you can test for a constant and get no error when it exists.
But the disadvantage is that when you test for a constant and you make a typing error, it will result in 0, and no error will be reported.

```

#### 1.11.8.7 Small additions to 1.11.8.6

- config dcf77 has additional sectic parameter.

the lib source is added to the commercial version too.

- Megal281 dat file added
- m168 dat file did not had tccr2 defined. config clock needs it.
- tcpchecksum() function did not work for big data since a word was used to add, and not a long. the function was written for the PING which did not need a lot of data. but now it will work for other data as well.
- the tcp lib had a problem for the slave address of the iim7010.it only worked with &H80 since a constant was not replaced by the internal slave constant.
- the m2561 dat file missed an interrupt and it caused a number of interrupts not to work.
- the config serialin1/2/3 was not well described in the help. further, the samples were adusted since the following label names were changed :  
Serial3ByteReceived into Serial2ByteReceived  
Serial4ByteReceived into Serial3ByteReceived  
Serial3CharMatch into Serial2CharMatch  
Serial4CharMatch into Serial3CharMatch  
This makes the label names more consistent with the uart names.  
When your code uses the BYTEMATCH option, it is important that you change it !!!

#### 1.11.8.6

- M164P added , attiny461 added.
- programmer: auto program will test the lock and fusebits when \$prog is used when a fuse is wrong/bad programmed, you get an error message.
- programmer, hex values of lock and fusebits are shown in the category
- DCF77TIMEZONE function added.
- simulator can use serial input from serial port too. Previous it only could send data to the serial port. Now you can use real serial input in the simulator.
- terminal emu, cursor pos is reset when it is cleared.
- pdf has a problem with dual monitor. this issue is NOT addressed yet.
- PDF : index can be hidden to get more screen space
- CloseSocket has a new optional param. : CloseSocket socket, prm  
Prm can have bit 0 and/or bit 1 set.  
Value : 1 - wait until all data is sent by the chip  
2 - wait until we get status \$18 or \$8 from the chip  
3 - this combines the above 2 options.  
\*\*\* Note that by default, both conditions are checked.\*\*\*  
so when you specify a value of 3, it would be the same as not using the optional parameter.
- Pinout viewer had a bug in loading wrong xml file, in case the path had an embedded space.
- the recent addition of [intlist] caused a lot of unforeseen problems. When all sample would have been tested, we would have found this problem.  
Since it is a major job to test them all, and the estimated impact was very low, random samples were tested. This is not considered good practice and for this reason we implemented a great idea from Luciano : the batch compiler. Under tools(you might need to add a button yourself) you find this new option.  
It allows to compile a number of files automatic. Some files are include files which should not be compiled as a main program and for this reason,  
have a look at the next new option.
- \$nocompile will not compile a source file. for example an include file. this prevent compiling include files which are not project main files.
- added option to play sound/wav after compilation
- sim IO-registers are sorted.
- simulator A/D uses steps of 1 instead of 10.
- WATCHDOG was not recognized any longer. Watchdog is a virtual name for WDT.  
WATCHDOG has been added to the dat files.  
A typical entry looks like : WDT. You can add multiple names : WDT@WATCHDOG
- enable serial/disable serial did not work anymore with the new intlist section.

this because enable serial in fact will set 2 bits. it is added again.

- tiny2313 dat file had an error in the intlist section with a double timer name. all dat files were checked again for errors.
- extended stk500 can also use USB. (just like the normal stk500)
- some dialog windows have an option 'do not show again'. For example when you compile while you are simulating, this will stop the simulation.

you might want to get this warning. but others might not want this warning and click the 'do not show again'. The Options, IDE, has a button to reset it.

- Pinout viewer can zoom in and out now. when you double click a pin, the color is changed into green. this way you can find the pin back later more easy.
- A double click on the chip will show a window with the specific chip properties
- Pinout viewer enhanced with 'cheap search' so you can search for a chip with certain properties. such as 2 uarts.
- good news for most users from Europe. The DCF-77 atom clock synchronized signal can be decoded. the result of research and hard work from Josef Vögel. The CONFIG DCF77 in the help will show which versatile options you have. The DCF signal is not available worldwide.
- tcp/ip UDP write wait on CSEND status. This is according to Wiznet API. But when the network cable is disconnected, it will wait for ever. You can provide optional timeout. See the PING\_TWI Sample.
- simulator, register, memory and IO window rewritten. It will color now as following: stop : black, red : memory changed, blue, memory was changed at some point.
- ISP programmer window did not resize correctly
- the new handling of CONFIG INT and CONFIG XRAM introduced a bug: when using the second time, the bits were not cleared.
- dat files check. and fusebit sections updated
- Simulator status will be shown in main window, the register, IO and memory windows are dockable. When you compile and the sim is running, you will get a warning.
- The M128 dat file had an error in the int-section. This resulted in wrong RX-enable bit set. Other chips with same problem : m32,m128can,m48,m88,m168,m16,m165
- USB-ISP programmer had a problem with programming the last page.
- wrong version of glcd.lib was in the distribution. causing T6963C displays not to work
- when using INPUT with UART4, the wrong code was referenced, resulting in a label not found error.

#### 1.11.8.5

second - New PDF viewer had a problem when no PDF was opened it kept on processing messages.

#### 1.11.8.5

- UART0 has a new option for the INPUT behaviour. Config Input0 = CR|LF|CRLF|LFCR , Echo = CR|LF|CRLF|LFCR  
The first option determines which character is used to end the INPUT. Default is CR  
The second option determines what is echoed back to the terminal. Default is CRLF.  
Of course when NOECHO option is used, there is no echo.  
A sample you find in the M324P.bas sample.  
NOTE : THIS works only for UART0. When we get positive feedback about this option it will be added to other UARTS as well
- CONFIG XRAM = mode [ , WaitstateLS=wls , WaitStateHS=whs ]  
mode : enabled/disabled , wls depends on chip. 0-3 for most new chips.  
This replaces \$XA and \$WAITSTATE directive to support new chips.  
While \$XA and \$WAITSTATE still work, they will be removed in the future since CONFIG XRAM does the same and more.
- buffered serial input will set ERR variable if an overflow occurs.  
while CTS/RTS checked for the buffersize, normal buffered input did not !  
This could lead to sync. problems since \_rs\_bufcountRx was updated.  
When a char does not fit into the buffer, it will be read from the UDR register but it is gone.  
When it is important not to lose data, you should use CTS/RTS.

- mega1280 added.
- bootloader updated with new chips.
- serial buffered bug for m25xx fixed.
- M624P support added.
- CONFIG ADC also supports :  
INTERNAL\_1.1 and INTERNAL\_2.56 for the reference.
- constants defined by atmel, are colored different in the IDE now.
- stk500 extended interface supports newer AVR Studio
- Added internal PDF viewer.
- Help About has link to settings file
- Report shows EEPROM Code map when using ERAM variable with DATA lines.
- DAT files renamed :  
at12def.dat into attiny12.dat  
at15def.dat into attiny15.dat  
at22def.dat into attiny22.dat  
at26def.dat into attiny26.dat  
When you source uses \$regfile, you have to change the datfilename.  
Since your old DAT file will not be removed, your source will compile ok but you can not use the new features then.
- DIM AT can have an index for arrays too :  
dim ar(10) as string \* 10  
dim var as byte at x(2) overlay
- The simulator had a problem with determining the V-flag (signed overflow flag)
- registers and ASM are shown in capital
- comment is no longer reformatted
- tiny85 added.
- In order to support new chips, it was always needed to make some modifications to the compiler.  
The goal for this release was to put all dependencies from the source code to the dat files.  
This makes it simpler to support new chips. And you can do it even yourself.  
Of course some chips require special support (like USB, CAN)  
Impact for statements using new DAT files : ENABLE, DISABLE, ON .. INT  
With impact we mean that even while we tested the new features, a bug might be introduced.
- [INTLIST] section added to dat files. When this list exist in the dat file, it is used.  
Otherwise internal code is used.
- Mega325 support added. M325.bas sample file added.
- SWAP with 2 longs had a bug since it used the wrong registers for the swap
- dat file of tiny2313 revised.
- clear serialout0 code had a bug : a type error in the register name resulted in a wrong returned value.  
this has no effect on the serial buffered code but when you check the value of the buffer in your code, it would not function correct.
- extended registers with address >FF were not handled correct in comparions:  
if PINK.0 = 1 then would fail since the upper register was cleared and not loaded with the MSB of the extended address.
- \$include is not aligned left anymore when a line is reformatted
- Mega329 support added
- base64encoder, the last byte was not properly encoded. fixed.
- bootloader.bas sample enhanced to support new chips like m2560, m329
- M644def.dat : Old compatible SPI register names added. CONFIG SPI=Hard would give an error otherwise  
pcint2 changed into pcint3 (type error)
- mcsbyteint.lib had a potential problem in the num->string conversion. The wrong register was cleared.
  
- New Serial buffers had a problem when used with other interrupts. The int. would cause the buffer routine  
to use a wrong counter value in some cases. A hard to find problem since it was hard to reproduce.
- adc\_int.bas sample changed. the FREE mode must be used for the ints.
- format() function had a bug when assigning the same string as the source.

- For COM2,Com3,COM4 it is important that you use : Config COMx  
This will setup the UART register(s).
- 90sCAN def file updated.
- added support for inputbin/printbin, and other missing code for uart3 and uart4  
such as rs485 support for uart3 and uart4, bufspace().
- second uart inkey/waitkey code had a bug.
- load timer did not wrote the MSB first for 16 bit timers. This can cause a wrong load of the timer since 16 bit regs must be written with MSB first.
- reworked the uarts. each uart is defined by a model number in the dat file
  - model 0: basic uart, with UDR,USR and UCR registers ,ubrr=0-255
  - model 1 : basic uart with udr and MP mode, ucsra, ucsrb, ubrr=0-4095
  - model 2 : extended uart with ucsra, ucsrb,ucsrc, ubrr=0-4095, own registers (non shared)
  - model 3 : extended uart with ucsra,ucsrb,ucsrc, ubrr=0-4095,shared register for setting UBRH / UCSRC
  - model 4 : extended uart with ucsra,ucsrb and both uarts share the same ubrH register for the upper 4 bits
  - model 5 : extended register with own registers and optional SPI mode.
 When you define your own dat files, you need to add the model. Most new chips will be model 5.
- CLEAR xxx had a bug. the pointer in buf\_clear was not increased resulting in not clearin tail and counter variables.
- CONFIG COM syntax support setting of BAUD.
- toggle for bits improved.
- the single.lbx alternative LIB had a problem with large negative numbers. And CONST -3e-15 was not proper evaluated.
- tiny44 and tiny84 support added
- simulator did not simulate reception of char right for chips with extended usr/udr registers
- CONFIG TIMER0 created wrong init code for some chips. (timer 0 comes in different flavors). in the near future new support for timers will be added which should make it easier.
- EDB sample files added to distribution
- USB programmer had a problem sometimes programming an EEPROM. fixed.
- INTS for simulation added to m2560,m2561,tiny24
- mega2560 support added. All 4 UARTS are supported. See sample under CHIPS dir.
- added tiny24 support

1.11.8.3 - when you open 'read only' files you get a warning and it is not possible to save the file under the same name.

- support for EA-DOG LCD's. See EADOG-M163.bas sample.
- tcp/ip readtcp, help changed. readtcp() returns a status byte, not the number of received bytes.
- tcpchecksum() function added
- bootloader sample added for using the second uart
- config print|print0 fixed. the flag was not reset so after first char, the level was changed too soon.
- \$timeout is also implemented for UART1 now. SO \$timeout work onboth UART0 and UART1.
- USB programmer switch for stk500 and similar added. You need to use stk500 programmer to use this option.  
It will add -cUSBserial instead of using the COM port.
- DEBUG statement added
- color LCD display support added
- Mega2561 support added
- Mega165 support added
- the help file(HTML) is no longer a 'stay on top' window.
- M16def.dat file modified(i2c pins added)
- serial buffers modified. CLEAR command added to clear buffers.
- CTS/RTS support added
- mega16 prescale 64 was not in the old datasheet for timer 0. new datasheet show 64 instead of 32. so the compiler has been changed too.
- get/put added for HW UART
- base64encoder routine had a bug which resulted in wrong conversion in some cases(depends on the used input)
- new single-> asc conversion lib

this allows output in scientific notation : 12.2E15 for example  
To override the default routine and use scientific notation use :  
CONFIG SINGLE=SCIENTIFIC , DIGITS= 0|7  
The value of digits param must be 0 (no trailing zeros), 1-7 for number of digits behind comma  
See single\_scientific.bas sample. LIB written by Josef Vögel.

- problem with for..next with words fixed.
- baud rate check did also checked non-UART chips such as tiny26. Fixed.
- CONFIG TWISLAVE has new optional paramter : SAVE=NOSAVE | SAVE=SAVE=NOSAVE can be used when you use asm in your slave interrupts. SAVE= can be used when you use basic inside the slave interrupt.
- 1.11.8.2 - the bufspace() parameter did not match the docs.  
the compiler has been changed to match the docs.  
(previous , 0 was bufout0, 1, was bufout 1, 2 wsa bufin0, and 3 was bufin1
- the revised code for .. next code did not work correct with local byte/word and in combination with 'optimize'.
- Help, IDE, Forum did not go to the forum.
- Help, Knowledge Base : selecting cancel also executed the browser.
- new MEMCOPY() function to copy blocks of bytes with optional increase if source/target
- \* BASCOM will fail on XP 64 bit edition. This is identified.  
Support for XP 64 will be made available in 1.11.8.3
- automatic conversion SBIS,SBIC, SBI and CBI did not work correct for address >\$1F  
this resulted for example in SPI not working for M168.
- fonteditor plugin updated to support bigger fonts
- 1.11.8.1 (special build : 1.11.8.1.002)
  - mcs.lib was not compiled to lbx.
  - IDE has hard to see support icon
  - most important and reason for update of compiler: RESTORE for Mega128 (or other chips with more then 64KB and using more then 64KB still had a flaw, or better a new bug caused by the fix).
- 1.11.8.1
  - by default HTML Help is selected now for a new install.
  - you can send a question to the KB from the help menu. it will use your default browser.
  - when a fusebit is changed, the WRITE FS/LB etc. can be clicked.  
when no changes are made they will remain disabled.
  - the TAB key did not result in an editor TAB but normal TAB.
  - the Lookup() function is used in the datatype lib but did not reflected the changes for big table lookup
  - SNTP() function added to TCP/IP. Based on Ben Zijlsta's and Josef Vögel code.
  - CONFIG TCP/IP extended to support TWI/I2C mode.
  - Added bufspace(3) for second UART
  - Adding a lib multiple times results in a warning and the lib will be ignored.
  - Added Varexist( "varname") to compiler directive  
This way you can execute code depending on the existence(DIMMED) of a variable  
#IF varexist("S")  
  'something  
#ENDIF
  - BETA : Added SETTCPREGS and GETTCPREGS for setting/getting registers from the w3100  
You can set time out or retry register this way for example  
settcpregs \$93,w,2 ' \$93 is MSB of timeout, W is a word variable, 2 is 2 bytes to write  
w=gettcpregs(&H93,2) ' get 2 registers from W3100 and store in variable W  
Note that you need to specify the MS byte of the address without the baseaddress offset.  
For example IRTS register is located in \$92 and \$93. You specify the highest address then.
  - M644 : ADC reference has INTERNAL\_1.1 and INTERNAL\_2.56 for the REFERENCE paramter

Also fixed wrong timer register references  
- internal optimise function checkLDS() did not checked for arrays. So when you left away the index in an IF condition you did not got an error message.

- loadlabel : rampz will be 1 for M128 when the address > 64 KB
- bootloader.bas for m64/m128 and other chips with 128 page size had a problem. The last page was not written. The loader is changed and the upload loader has been revised too.
- tiny13.dat file tccr0 register had wrong address.
- read/write had a problem for M128 when between reads, the rampz register was altered by the user or the program. As RESTORE/READ will store a pointer in R8/R9, and M128 needs RAMPZ too for big programs where data can cross the 64K boundary. a byte from SRAM was taken to add it to the R8/R9 pointer. the code was placed in a subroutine saving code space for succesive calls.
- getattr() for AVR-DOS fixed for empty param
- for..next with bytes with step > 1 failed

1.11.8.0

- \$PROGRAMMER option added to specify programmer
- \$XA added (xram access)
- base64ENC function added to code a string in BASE64
- Added MakeTCP() function to convert data into an IP number  
Simple form : Zz = Maketcp(192.168.0.6)  
With var and numeric data : Zz = Maketcp(192 , 168 , 0 , B4)  
And you can reverse the order with an additional param : Zz = Maketcp(b4 , 9 , 168 , 192 , 1)
- tiny25 and tiny45 dat files fixed : the chip does not support HW MUL
- M169 also support soft CLOCK mode now
- simulator input test file .SI documented  
Showin Hex option also added to help.
- \$loadersize = 1024  
new option for \$loader so you can specify in your program how much space the loader uses otherwise loader space could be overwritten. this is for usage with external programmers  
See also new setting in bootloader(programmer options)
- config clockdiv=1|2|4|8|16|32|64|128|256  
this is a new option only valid for newer chips with CLKPR register  
It will divide the clock frequency.
- Min() for bytes did not work as it started with 0 and never found a smaller value
- all files that are open when closed, will be loaded upon the next start.
- simulator did not simulate adc for M48/M88/M168
- SENDSCANKBD accepts a variable or constant also. Previous it only accepted a label.
- improved MULL16 routine for HW mul. This is placed into mcslib also.  
Also new is Hardware Mul for Long multiplication. Credits to Josef.  
Not only the code will be reduced by using HW MUL(when the chip supports it) but the code will execute quicker too.
- RS485 did not wait until transmission was completed. Now the code will wait for TXC complete.
- pcint sample added (PCINT\_CHANGE.BAS)
- extended I2C support added. This is an add on lib so you can use i2c on portF of a m64/m128  
the lib is only provided with the full version
- tiny13 timer0 fixed. the wrong timer model was used.
- PRINTBIN does not print arrays() > 256.  
A new option allows to print larger arrays.  
Config Printbin = Extended  
As it will generate more code, (word index instead of byte index)  
use it only when you need to print big arrays with PRINTBIN
- \$regfile lists all dat files in the editor.
- M48,M88,M168,M32 def files fusebits tables updated
- DEFDBL added

- Hex(),READ,RESTORE supports double too.
- LOADWordADR added. This loads Z and RAMPZ.  
Usage : LoadWordAdr LABEL
- bytematch added to buffered serial input
- Lookupstr() usesd \_lpmbyte now for 64K boundary check  
All other code using ELPM/LPM was changed to use \_LPMBYTE too.
- Lookdown() for integer/word resulted in offset error for LDD.
- \$LOADER support
- SPLIT() function added  
Bcount = Split( "this is a test" , Ar(1) , " ")  
It will split a string into elements
- varptr() can be used in CONST to get the address of a variable :  
CONST xx = varptr("varname") ' get address of the variable
- BIN2GREY renamed to BIN2GRAY, and GREY2BIN renamed to GRAY2BIN.
- all register names are shown in proper color now when you use \$REGFILE
- AVR-DOS, Creating file and Directory also fill the creation date/time field in Directory  
Double read and write (GET, PUT) in ASCII and Binary were added by Josef.  
INPUT and PRINT can work with the DOUBLE in AVR-DOS
- When you DIM a DOUBLE, the double.lbx is added automatic.
- GetADC() routine in lib fixed for M48, M88, M168. The config was not preserved.
- All i2c code (soft and twi) , master and slave , were re-tested. twi master and slave code is improved.
- Getadc() in free running mode, needs the following command :  
SET ADCSR.6 in order to work. It depends on the chip.

#### 1.11.7.9

- support for AtTINY25 and atTINY45 added.
- TWI lib changed. Interrupt bit is not set as it does not use an ISR to service the int.
- SHIFT+F4 resets the chip for most programmers.
- PageDelay added for custom page delay. Edit the bascavr.ini file like this :  
--look for section [PROGRAMMER]  
-- add PAGEDELAY=20  
The default is 30. When you do not add anything, 30 will be used.  
This is an advanced option that allow you to experiment with the page delay.
- RC5SENDEXT added. this will send Extended RC5 code.  
The TOGGLE byte must contain the the extended code AND the MS bit must be set.  
TOGBIT = &B1110\_0000 'toggle bit set and leading bit is 1, bank 1 selected(normal RC5).  
TOGBIT = &B1010\_0000 'toggle bit set and leading bit is 1, bank 0 selected(exstended RC5) .  
  
Rc5sendEXT Togbit , Address , Command
- M168 dat file fixed. pagesize was wrong leading to error in programming.
- simulator fixed. a double word instruction after sbrc was not in all cases executed.  
this resulted in displaying wrong double values.
- config int =change did set wrong bits for M8,M48,M88,M168.
- SQR() for double added.

#### 1.11.7.8 NOTE

: 30 - 6 - 2005  
MCS is celebrating 10 year of BASCOM. We are expanding business and changing a number of things you will notice in the future.  
One thing is that there will be more frequent updates.  
There is never a good moment to release as we want to put in as much as possible.  
This release is only there to fix some M48 and M88 issues and to release the M168 def file.

There will be a release each end of the week with at least 1 new atmel chip.  
This release also has no HELP update. This is due to the fact that we are converting the help to a new Help system  
The new help and manuals will be released within 1-2 weeks.

- Support for a new USB programmer has been added. This USB programmer is currently still under test, and available soon from our webshop.
- M168def.dat file added. Not tested with a real chip.
- M48-88-168-PWM.BAS sample added for showing using 6 PWM channels
- conditional compilation with #IF did not work correct with nesting
- Watchdog simulation improved.
- CONFIG INT, also supports CHANGE now : CONFIG INT1=CHANGE ' to trigger on a change of the input level
- Added support for AVR-DOS 5 : GETATTR, MKDIR, CHDIR, RMDIR, MKDIR, NAME <old> AS <new>  
AVR-DOS is an add on product, the LBX is free for personal use. The LIB source can be purchased from Josef.
- Added double TRIG functions SIN, COS, TAN, ASIN, ACOS, ATN, ATN2, COSH, SINH, TANH, POWER, EXP, LOG, LOG10, RAD2DEG, DEG2RAD  
The Double TRIG LIB is written by Josef Franz Vögel
- \$initmicro was called at the wrong moment. On Mega chips, this would give problems as the interrupt address 2 was used too.
- Serial output buffer for the second UART(COM2) also uses improved buffered code.  
Bufspace(1) returns the amount of free space in the output buffer for COM2.  
Bufspace(0) returns the amount of free space in the output buffer for COM1.
- CONFIG PRINT and CONFIG PRINT1 added for RS485  
CONFIG PRINT0 = PORTB.1 , MODE=SET|RESET ' will make PORTB.1 high while printing and if SET is specified  
^^^ UART0. Use CONFIG PRINT1 for the second UART
- \$notypecheck and \$typecheck added to pass custom data in subroutines
- min/max function for WORDS fixed. The wrong pointer value was loaded.
- CONFIG LCD, enhanced with param :  
CONFIG LCD = 16\*2 , CHIPSET=KS077  
^^ new option
- CONFIG SERVO enhanced with selectable TIMER0 or TIMER1.
- demo version fixed , it could not compile up to 4KB
- simulator supports sbrs/sbrc correct after double word instructions  
lookup.bas sample added: lookupStr() can use >255 entries now  
m103def.dat file changed : this micro has NO HW mul unit/instruction

#### 1.11.7.7 RELEASE

Same release but with EXE unshrunked. Shrinking the exe caused problems at some OS/PC.  
Also right i2c.lib version 1.11.7.7b as the wrong version was distributed.

#### 1.11.7.7 RELEASE

- Bufspace() function added. this will return the space left in the serial buffer.  
implemented only for serial output buffer 0.  
var = bufpace(0) ' 0 for output buffer 0  
Use in combination with buffered serial output.
- Added directives \$HWSTACK, \$SWSTACK and \$FRAMESIZE.
- increased string constant space("text")
- new introduced Encoder function had bug, fixed.
- buffered serial output had a problem. it did not sent all the data from the buffer when the data was full.  
the fix, needs some more code so your program will get a bit longer. It does work very reliable now. When the buffer is full, the data is not written.

#### 1.11.7.6 BETA

- interface has XP look now as the menu looked strange on XP.
- double1.bas and double2.bas samples added.  
When you did not noticed yet, the DOUBLE is a great extension written by Josef Franz Vögel.  
Ok, you need 8 bytes to store each double, but you get the ability to crunch huge numbers with the range from ~ E-308 to ~ E308

- m88.bas files added with m88 def file.
- assigning a byte from an integer array with a constant : byte=integer(2) would assign not the LSB of the integer to the byte
- \$external \_spiin will bind in spiIN routine. \$external \_SPIOUT will bind in spiout code
- added Link in IDE to MCS Forum and MCS Shop.
- osccal will return multiple osccal settings in the programmer interface when the chip supports it
- mega48.bas sample added
- added getadc() second optional parameter to specify offset for GAIN, differential input etc.
- tiny26 timer1 configuration fixed(timer1 in tiny26 differs from the normal timer1)
- added M128CAN dat file. tested with programmer.
- Added replace button to the toolbar
- Added Editor option to insert/remove code remark '
- Added number of free bytes to the report
- Version() supports new param : version(1) will create dd-mm-yyyy format while version(0) or version() will create mm-dd-yyyy format
- problem with extended ports fixed
- auto update removed from IDE. Use update wizard.
- selecting dat file from Options, Compiler did not work anymore
- using OPEN with SW UART on portG was not supported.

#### 1.11.7.5 BETA

- added TWI master replacement library, see TWI-master.bas sample
- Megal69 ButterFly LCD driver added.(lcd\_butterfly.lib)
- ASCII tool allows to lookup and insert ASCII chars.  
Press the Right mouse button in the editor to show the ASCII insert window.
- CONFIG PORTx, bug fixed when providing multiple values.
- Min() and Max() added for byte arrays
- Watchdog simulation added to the simulator.
- SBIC is transformed into LDS + SBRC automatic for IO registers with address > &H31 (R0 is used)
- SBIS is transformed into LDS + SBRS automatic for IO registers with address > &H31 (r0 is used)
- SBI is transformed into LDS + SBR + STS for IO registers with address > &H31 (r23 is used)
- CBI is transformed into LDS + CBR + STS for IO registers with address > &H31 (r23 is used)
- These transformations will ensure that old code will work on the new (extended) registers.
- CRC32 function added.
- New Encoder function added for using rotary encoders
- \$INITMICRO added to add user code at startup  
\_init\_micro: is called
- CPSE instruction was not simulated correct for the entire range of object codes.
- VERSION() function added that shows date and time of compilation.  
Intended usage is to get a unique version number
- BITS() and NBITS() added
- SONYSEND has new additional param to sepcify number of bits  
SONYSEND LONG, 12 | 15 | 20  
The data to send must be of type LONG in that case.  
Number of bits must be 12, 15 or 20.
- DIM AT option can use offset too:  
DIM var AS BYTE AT varX + 1 OVERLAY  
                                  ^--- offset
- ADCD word variable is ADCL + ADCH so you can read/write simple to the data registers of the A/D converter
- copy , paste and logging added to the simulator RS-232 area.
- PUSHALL/POPALL and the interrupt routines do not save/restore R8/R9 registers anymore. R8/R9 are used as datapointer by READ/RESTORE
- val() for string -> single will eat all leading space : s=" 123" . will be converted ok.
- A warning is added to the report when STR() is used on a single and the target string is too small.  
For example you can dim astr as string \* 4. And use : astr=str(x) . Since the result might not fit into astr, you overwrite memory.
- WRITEEEPROM will disable interrupts. And enable them when complete.

- Bookmarks are saved in .BM file and loaded automatic.
- STK500 programmer will optional verify
- baud #1, 9600 did not work for the software uart when the chip had no HW UART.
- LCD 20\*4A which uses special KS chip, is initialised now in extended function mode.
- Added \$TIMEOUT support for serial channel 0.  
\$TIMEOUT = 100000 ' will wait relatively time. Basicly the timeout is a long that is counting down.
- Fixed SERIN statement. Multiple bytes were not handled.
- KS0108 driver added for 128\*64 graph display. See ks108.bas sample
- Programmer can write \$PROG to the editor.
- New GETATKBDRAW() function. This will return a byte from the keyboard and this is the SCAN code.

#### 1.11.7.4 RELEASE (Special build 1.11.7.4.002)

- Megal62 support added
- mega8515 and mega8535 jump error solved.
- lcd 20\*4A added. Address starts at \$00,\$20,\$40,\$60
- support for 86rf401 added
- IP2STR(ipnum) function added to convert IP number into a string.
- \$INC label , size|nosize , "file.txt"  
This directive will bind in any file. It will insert the label, optional the size, and the file content

#### 1.11.7.4 BETA

- X10SEND is extended with optional bytes/data to send.
- \$PROG lb,fuse,fusehig,fuseExtended  
This will automatic program the fusebits with auto program.  
See help for more info.
- SETTCP statement added.
- Problem with UDP fixed.
- DHCP sample added. Written by Atilio Mosca.
- new plugin added for showing bmp files.
- color printing is changed to B&W due to some problems. Color support will be added again later.
- The demo version can use the code optimisation too now.
- New Loader(programmer) for Lawicel STAVER.
- New communication control saxcomm7.ocx , replaces mhcomm32.ocx. VT220 terminal emulation removed.
- MULS,MULSU, FMUL,FMULS and FMULSU menmonics added
- 1200def.dat file added for the 90s1200. Only programming is supported. And of course asm programming.  
Programming did not work on a STK200.
- A complete FILE System : AVR-DOS is now supported.  
Josef Franz Vögel wrote the complete library that is now integrated in the compiler.  
A great piece of work. Since you can rewrite the driver, you can use it not only with CF-cards,  
but also with XRAM(for simulation) or write a driver for a serial eeprom.
- the AVR ISP programmer settings can be changed.  
Edit bascavr.ini and add the section [AVRISP]  
then add : COM=19200,n,8,1  
This is the default. When you made your own dongle, you can increase the baud rate
- programmer from www.elm-chan.org/works/avrX added to Universal Interface
- LCDCONTRAST value  
value must be a value from 0-3. This will set the contrast on some text LCD's.  
Will not work with all lcd display. Noritake display should work.
- Added support for the MEGA8515 and MEGA8535.
- OVERLAY has a new option to specify the name of a variable instead of an address:  
dim w as word at &H64  
Dim B1 as BYTE AT W OVERLAY ' this will place the byte at the same address as variable W

- CONFIG ACI=ON|OFF, COMPARE=ON|OFF , TRIGGER = TOGGLE|RISING|FALLING  
The TRIGGER param was not right documented. The param is TOGGLE, RISING or FALLING.
- Bootm8.bas bootloader added. Sample for the M8, written by David Moran.
- MOVW added to simulator and internal assembler. Usage :  
movw r0,r12 ; copy r12 to r0 and r13 to r1  
The W is for WORD since it copies to registers

### 1.11.7.3

- X10SEND command added to send X10 commands with TW-523
- Mouse and AT keyboard emulator libraries available.
- W3100A chip supported with TCP/IP and UDP

### 1.11.7.2

will take

- Getadc() performs 1 conversion instead of 2. Use a dummy read in your program when the first result of getadc() gives a bad output. The first conversion might give a wrong result. For this reason the previous versions performed 2 conversions. But this additional time. For this reason, only 1 conversion is now performed.
- T6963C 160\*48 display added.
- You can add \_ at the end of a line to indicate that the line will be continued on the next line.  
For example : print "this is a test" \_  
                  print " this is part of the previous line"  
Note that no characters may be plaed after \_ , only spaces.
- DateTime.lib added. This library is written by Josef Franz Vögel. It extends the build in soft clock routines. Look in help under ASM libraries, DATTIME. Also dateTime\_test1.bas sample added.
- CONFIG I2CSLAVE has new options:  
config i2cslave = address, INT = INT0, TIMER= TIMER0  
'By default timer0 and INT0 are used. When you change the timer and/or int, you need to change the library code too.
- Supported programmer can be run manual by selecting Program from the menu.
- M16 support tested with sample chips. M16.bas file added.
- added optional check for ON value GOTO/GOSUB  
When you give the last parameter the name CHECK, the compiler will check the value against the number of supplied labels.  
When you use ON value GOSUB 11,12 and value is 10 there are not enough labels and there will be a stack overflow.  
When you use ON value GOSUB 11,12,CHECK the compiler will chack and find there are only 2 labels.  
Values less then 2(in this sample) will not be executed so the line after ON value .. is executed.
- CONFIG KBD has an additional parameter : DELAY = value  
This additional delay causes a delay in mS after the routine has finished.  
The added standard 100 mS delay is removed because it was incompatible with previous released.  
To get the delay again use the additional parameter : delay = 100
- CONFIG ACI=ON|OFF, COMPARE=ON|OFF : added :  
TRIGGER = TOGGLE|RISING|FALLING  
config aci=on|off, compare=on|off, trigger=TOGGLE|rising|falling
- support added for M32
- lockbit programming added for serial programmer
- ISP programming can program faster now. Use a small number for port delay like 0.1 or 0.01 for faster programming.
- LIB compiler gives a meaningful line number in the ERR file now.
- New IsCharWaiting() function did not work in combination with buffered serial input.
- Support for M64 added, not tested(no sample chips available yet)
- \$boot option info: when you use \$boot = FE00 it points to a word address.  
So the binary address will be multiplied by 2 : 1FC00
- New HW mul routine destroyed r0 which was used for the sign : signed integer muls resulted in an error

- 1.11.7.0
  - showpic gave error 5 for T6963C displays caused by new SED support.
  - support for (asm) 0b011 resulted in problem for &H0B, &HB worked.
  - simulator keypad was invisible
  - support for {xxx} characters failed.
  - added CONFIG DATE = DMY|MDY|YMD, SEPARATOR = /.-  
Will be used for date/time library functions
- 1.11.6.9
  - LCD can be used in 8 bit PINMODE too : config LCD = PIN, PORT=PORTB, E = PORTD.2, RS=PORTD.3
  - config INTx=CHANGE added. This only works for newer MEGA chips that can trigger on a change of the input level.
  - simple STK500 programmer programs EEP file too. Use \$eepromhex to generate Intel Hex EEP files.
  - LCD in pin mode supports 8 bits : CONFIG LCDPIN = PIN, PORT=PORTB, RS=pin, E = pin
  - IsCharWaiting() function added to detect if a serial char is waiting
  - !!!! ERR bit is not set anymore with INKEY. Use Ischarwaiting now to detect the presense of a char
  - Added RC6SEND. Works the same as RC5SEND but uses RC6 mode 0 protocol.  
The address and command are both 8 bits.
  - Multiply for bytes and words use the MUL instruction automatic for the (mega) chips that support this instruction
  - When using HTML Help the hhActiveX.dll must be installed
  - CONFIG ACI=ON|OFF, COMPARE=ON|OFF  
Will switch ACI On or Off and will allow ACI to trigger input capture of timer1 by ACI.
  - SPC() for graphical LCD's fixed. R24 was destroyed after the first space.
  - tiny26 support added.
  - Bin2Grey had a wrong reference to the asm code for integers/words, resulting in an error message.
  - Removed the credits form. It is in the helpfile now.
  - Added plugin font editor and plugin support.
  - SERIN and SEROUT added. These are dynamic software UART routines.  
All parameters can be set at run time!  
It works for any pin. And also for the same pin.  
Serin S , 0 , D , pin , 19200 , 0 , 8 , 1  
Serout S , 0 , D , 1 , 19200 , 0 , 8 , 1  
S - a variable  
0 - number of bytes to send/receive , for a string, always 0  
D - the port to use, in this case D  
pin - the pin to is(0-7)  
then the baudrate, parity(0=NONE,1 = EVEN, 2 = ODD)  
number of databits(7 or 8) and the number of stopbits(1 or 2)  
You can use variables for all parameters. This makes it simple to change them at run time.  
Note that these routines are stand alone: they can not be mixed with OPEN/CLOSE
  - CONFIG CLOCK has a new option : CONFIG CLOCK=USER  
This will dimension the internal variables so you can use DATE\$ and TIME\$  
You need to fill the variables yourself with your own clock code.
  - Added boot128X.bas This is a variant of the normal bootloader.  
It will wait for some magic bytes from the serial port at start up.  
Also the pages are erased before they are written.  
Note that the bootloader programs only serve as samples. There is no support intended for every AVR micro.
  - Improved uS delay timing used for ISP programming.
  - Register R6 is not saved anymore in interrupts. This to allow to set the ERR bit from inside aninterrupt without the need to use NOSAVE option for this.
  - \$LCDVFO added for VFO displays. They need a very short enable pulse to avoid flicker.
  - getkbd changed: a delay of 100 mS is inserted. This delay solves the problem of some keyboards that return values higher then 16.
  - glcdSED.lib added for SED1565 graphical display support.  
Should work with all SED based chips. See sed.bas

GLCDDATA and GLCDCMD added.  
LCDAT added for SED LCD  
display on/off works  
SETFONT added for using different fonts  
See SED.bAS demo file

- circle statement fixed : fourth quadrant had x and Y reversed
- extended RC5 receive added thanks to Gert Boer who did the research and modified the code to support extended RC5 reception. Since there is an additional bit, the toggle bit has been moved to bit position 7. The extended bit is in position 6!
- loadlabel() function added. See readdata.bas sample
- boot128.bas sample added for M128 bootloader.
- tiny15.bas sample added that shows how to use tiny chips without sram

1.11.6.8

- I2CINIT added to help. Can be used to re-init the I2C pins if altered when writing to PORT or DDR by the user.
- Showpic and ShowpicE statements did not worked for all pictures. The BGF RLE compressed files are now smaller but need to be recreated because of the new RLE encoding/decoding technique.
- writeeprom with strings, writes only the length of the string and not the dimensioned size of the string now.
- print(preview) can be done in B&W or color. See options, printer
- print(preview) line numbers are also printed (by 1.1.6.7 update they were omitted)
- baud rate error calculation in report was wrong.
- LINE draw routine had a problem for some angles. fixed.

1.11.6.7

- LINE routine rewritten. Draws perfect lines now for any angle.
- Graphic LCD supports mode 6 too now for showing pictures. You need to recreate the pictures for mode 6 !
- Graphic lcd 128\*128 supported
- print preview and printing works in color now for color printers.

1.11.6.6

- Universal programmer support for the HotChip programmer.
- Josef Franz Vögel worked hard and here are his latest routines:  
ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN and rewritten SQR.  
SHIFTSL and SHIFTSR are shift routines for single variables and are used by the SHIFT statement that now can shift a single too.  
INT was modified and POWER takes the sign into account now so that ^ will use the power routine now.  
Also SQR() for byte,int,word and long added. Also an alternative SQR routine lib is provided. It works faster but uses more code.
- buffer serialinput rewritten. Also added for second UART:  
CONFIG SERIALIN1 = BUFFERED, size = 30  
CONFIG SERIALOUT1=BUFFERED,size=20
- added spi-softslave.bas that show how to implement a SPI slave using software.  
Great for chips like the 2313 that do not have SPI
- added M161.bas to show how to use 2 hardware UARTS  
also added: baud1,\$baud1,\$serialinput1,\$serialoutput1
- added ds2405.bas sample for 1wire switch
- servo code optimised. Uses a word now instead of 2 separate bytes.
- Circle statement added. Circle(x,y),radius,color . Thanks to Josef Franz Vögel who wrote the circle code!
- picture data can be stored in EEPROM too now. Use showpicE instead of showpic to show from EEPROM.
- Graphic converter saves data RLE encoded. Showpic is using RLE decoding now. This saves a lot of space.  
Thanks to Brian Millier
- STK200 circuit changed in helpfile since it showed a wrong connection.
- ATN function from new fl\_trig.lib did not work for negative values
- Mid with local strings failed
- INPUT num and VAL() can detect a leading + too now. For example +12.
- M323def.dat contained some wrong bit value number for powersaving.
- added custom lcd lib with busy check lcd4busy.lib.

1.11.6.5 BETA

Do NOT forget to compile your project before simulating it.  
The new DBG file is not compatible with the old one and simulating a program  
compiled with a previous version will give problems!

- 240 \* 128 lcd now supported
- added LINE statement for graphical lcd
- waitui routine optimized. It is more accurate now. See also help.
- Added autobackup see environment IDE options
- ABS() works for singles too now
- Frac() function added, returns fraction of a single  
sin,cos,tan,atan, int, rad2deg,deg2rad, log10,tanh,sinh,cosh  
power. log and exp replaced with new routines from Josef Franz Vögel  
Josef provided a demo too. When there is a problem don't blame Josef since the adaption into the compiler  
can be the problem. All was tested however :-)
- the simulator can now print to the comm port.
- min() and max can return an index too now:  
max(w(1), m,idx) ' m will return the max value, idx the index  
min(w(1),m,idx) ' m will return the min value, idx the index  
When used with an index, it must be used as a statement, not as a function !
- M128 support added : use M128103.dat for the M103 compatible mode.  
Use M128def.dat for the M128 mode
- support for WinAvr and SP12 added to the universal programmer
- err bit added to help of i2c
- M8 support finished.  
config COM1=dummy,synchrone=0|1,parity=none|disabled|even|odd,stopbits=1|2,databits=4|6|7|8|9,clockpol=0|1
- log window added to the auto update function.
- calculation of the user sram available space was wrong. See also \$dbg and DBG
- \$DBG and DBG added for determining stack overflows  
use tool stack analyser. Requires that softstack is set > 0
- pulse generator added to the simulator interrupts TAB.  
Select the counter pin and the number of pulses to generate.  
Timer simulation must be checked to use this option.
- variable color can be selected in the environment options too now.
- using SET/RESET in combination with alias reported an error when the alias was a register.
- error 70 will be reported when baud rate error > 10%  
The report already shows the error by the way but this give you a clear indication that the selected baudrate will not work
- crc16 added(already in 1.11.6.4.)
- eurtimedate.lib added for soft clock : gives DD-MM-YY format instead of mm/DD/YY
- added fuse bit I and H to the programmer for JTAG bit programming.
- assigning a string in a sub with left or right could give error 260(string too short).
- external programmer accepts {FILE} to insert the program file and {EEPROM} to insert the EEP file on the command line
- .equ accepts expressions too now.  
.equ myval = (10+2)/4  
ldi r24,myval+2 '5  
ldi r24,asc("A")+1 ; load with 66  
works for all asm mnemonics that accept a constant value
- added check for compiling a LIB file. An <libname>.ERR file will be created when an error occurs
- DEMO can compile lib files too now.
- New option to dim variables as pointers:  
Dim A As Byte At \$60 'dim at hex 60  
Dim Y As Byte At \$60 Overlay 'use overlay to point to same location  
A = 1 : Print Y ' will print 1  
Do not make mistakes with different datatypes allthoug this could be handy too:

```

dim L as long at $60
dim b1 as byte at $60 overlay
dim b2 as byte at $61 overlay
dim b3 as byte at $62 overlay
dim b4 as byte at $63 overlay
l=&H12345678 : print hex(b4) ' should print 12

```

space So this can be an alternative to alter byte positions within a word/long since the duplicate/overlayed var does not use any additional

OVERLAY does not DIM a real variable. It may only be used after other variables occupy the space.

OVERLAY is in fact a pointer to a space in memory with the specified size!

So even when you overlay a LONG, it will not use the 4 bytes that are normally used by a LONG.

- Extended stk500 support is provided now. This is a new option in the programmers list.
- Also : use new \$eepromhex directive to create Intel HEX EEPROM data files for the stk500.
- Bin2grey and grey2bin function work for integer/word and long too now.
- Thanks to Josef Franz Vögel for an improved algo and the word/long versions.
- The I2c routines are in the library i2c.lib. Config i2cdelay has been changed.
- Timing is in uS now.
- config spi accepts a new parameter for SS. Intended to be used when you want to handle multiple slaves
- i2c wbyte can be simulated now in the simulator.
- AT option gives an error now when used at an occupied address
- idle and powerdown example added
- lcd emulator fixed: deflcdchar 0 could not be displayed.
- also fixed: lines 3 and 4 were not updated
- getatkbd() routine has a bug in 1.1164:fixed
- The new getatkbd routine works better with shift key.

#### 1.11.6.4

- printbin also accepts an additional numeric param to indicate the number of bytes to send :  
printbin ar(1) ; 2 ' send 2 bytes from ar(1)
- rc5send and sonysend IR remote control transmission statements.
- inputbin accepts an additional param when used with arrays now:  
inputbin ar(1),2 ' will read 2 BYTES. The additional param must be a number
- waitms problem fixed. Bug introduced in 1.11.6.3 : MSB >0 resulted in no delay.
- getkbd changed to support 6 rows  
config getkbd=portb, rows=6, row5=pinD.6, row6=pinD.7 ,debounce=20  
not tested with hardware
- BINVAL() option added. Will convert a string in binary format into a number.  
s= "01011" : v = binval(s) 'v will be 11
- The IDE shows new chips but they are not yet supported(M8,M16,M64,M128)  
Once it is verified with sample chips it will be released.
- Renamed M32 into M323 in case Atmel will come up with a M32 chip.
- Fixed : INPUTBIN with strings deleted the last entered character.
- Enlarged resolution of servo routines.  
It will use 10 uS steps now at the cost of an extra byte from sram.  
Only change in your code is the reload value. Must be 10 now instead of 100.
- LIGHT ALIAS PINB.1  
CONFIG LIGHT = INPUT
- Min() and Max() return the minimum and maximum value of a WORD array.
- Grey2Bin and Bin2Grey function added for Grey-code conversion.
- Security byte added to EEPROM WRITE routines. R23 is set to 123 before the write routine is called.
- Added Universal interface for programmers. This means that you can customize the pins used  
by your programmer!
- M163 fuse bits programming problem fixed. Datasheet had a wrong description. prog.ini changed.

- inkey() will set ERR when no data is returned  
When data is returned (0 for example) ERR will be cleared.  
This way you can use ERR to detect if a byte was received. This works also for 0 bytes.
- config RC5 has an optional param to select timer2:  
config rc5= pind.2, timer=2  
Works only on chips with timer2
- Added line numbers to the IDE
- Added snapshot to simulator
- Added \$MAP directive that generates line,address info in the report
- Added string variable to DTMFOUT : DTMFOUT s , time  
s can be a string assigned with "12345" for example. s may not be a constant!  
Also fixed a problem with the duration. A register was destroyed so the duration was not correct.

#### 1.11.6.3. beta

- Added warning when \$SIM is used while trying to program a chip
- indexing bits with an array did not work like ar(i).var = 1
- \$BOOT option added. This to allow inclusion of a bootloader program  
\$BOOT = \$1E00 ' MEGA 161 se help
- SPM asm mnemonic added for boot loader support
- Bin() function added. Bin() converts a numeric variable in the binary string representation.  
dim b as byte : b = &B11001010 : Print Bin(b) ' prints 11001010
- Lookdown() function added for bytes and integers.  
Dim Idx as integer, search as byte,entries as byte  
search = 3 : entries = 5  
Idx = lookdown(search,label,entries)  
nop  
Label:  
data 1,2,3,4,5  
  
Search is a byte variable that holds the number to search  
Entries is the number of entries in the table  
The result will be -1 when not found or will be 3 in this example since the third value will match  
Lookdown also works for integers. Search must be dimmed as integer in that case.
- Optimized string<>num conversion routines. FRAME size must have a minimal length of 16 now.  
This space is used for the num<>str conversion like when you use INPUT integer. THE data you enter is stored in this buffer and then converted to an integer.
- Added range check to floating point mul routine. Otherwise when mul with a very small number, there result might be wrong
- Servo routine changed so the limit will be 20 mS when reload = 100 will be used.  
Servos need a 20mS pulse. Before 25 mS was used. Also the servos are now synchronized turned on.
- Optimized hex 2 num conversion routine. HEXVAL() may have a string of odd bytes too now.  
Before hexval() needed for example 0A. Now you may use A as input.
- When you use Save As, you will be asked to leave the CFG file.  
When you choose Yes, the CFG file is copied so the old one still exists.  
When you choose No, the CFG file is renamed so the old one is deleted.
- the path is saved and restored now between sessions. So opening a file from c:\test will set the path to c:\test the next time you run BASCOM.
- Added GETADC() simulation. In the hw simulator you can provide the value of the adc
- INT0/INT1 is now simulated too when setting/resetting the port bits in the HW simulator window.
- Also added a port delay to the parallel printer port based programmers.  
You can use this option when the programmer runs to fast on a very fast PC.  
The generated clock takes the PC speed in consideration so it should not be needed. Added for testing purposes.  
Default value should be set to 0.
- Added a test mode to the Sample Electronics programmer.

In case of problems go to the File, Test, Sample Electronics Programmer and choose the menu items to set the various used pins. After setting a pin to +5V you can read the value with a DVM to see if it works.

- \_ADD32 library routine was double referenced.
- SPC() added : Print "test" ; Spc(5) ; "this"  
SPC() prints spaces. The difference with SPACE() is that space() will create a temp buffer when used with print. SPC() will just send the number of specified spaces. Will work with LCD too : LCD "Hello" ; spc(3) ; "world"
- Returning the MID() of a string where the starting position was > the length of the string gave a wrong result:  
s="abc" : var = Mid(s,4) returned the string starting at pos4 without checking the length of the string.  
The result should be an empty string. Also changed the mid statement.
- INSTR() function had a bug in the library code. When a part was found but not totally matched, the returned position was wrong.
- echo on ,echo off added. This replaces the INPUT var NOECHO option.  
When you had code like : INPUT var NOECHO , you must rewrite it to :  
ECHO OFF  
INPUT var  
'to turn it on for the next INPUT :  
ECHO ON  
Existing code will still work but it saves code to turn echo on and off only when needed.

#### 1.11.6.2

- Added HTML help (bascavr.chm) that can be used on W98 and W2000.  
To use the HTML help instead of the old windows help, select the option from Options Environment IDE.  
Also added helplinks to the menu. So pressing F1 while on a menu item will bring up the help.
- New : \$TINY is intended for tiny support. At the moment it will not setup any stack pointers since there is no sram in the tiny parts.  
You can write in asm or may use part of the statements since only a 3 level hw stack is inside the tiny parts.  
There will be a separate lib for the tiny with a subset of BASCOM statement.  
To use a variable dim it like : dim var as IIRAM byte  
Future support for tinys will be for byte and int/word only
- help file was updated for the language reference.
- string -> single conversion had a problem where there were zero's after the dp : 0.01 was converted to 0.1
- the right() function returned wrong number of chars in the 1.11.6. Also optimized and help file corrected.
- The compare mode simulation was working only for pin PORTD.5 since the 8515 was used to develop BASCOM.  
Changed to look at the chip.
- The mcsbyteint.lib NUM2STR conversion routine was bigger than the original MCS routine.  
So I put the original code in the lib.  
When using the generic byte,int,word,long lib from mcs.lib printing a few ints/words : 836 bytes  
When using the previous mcsbyteint.lib : 798 bytes  
With the new routine : 748 bytes  
(bytes including the total program btw)
- Added DTMFOUT number, time  
number is a numeric variable or constant and time is time in mS the dtmf is generated.  
Uses TIMER1 in interrupt mode. See dtmfout.bas example  
Not tested with telephone line.
- \_CheckEepromReady had a bug. The Z-pointer was increased before the whole address was transferred to the address registers. Only occurred in big chip with >255 byte EEPROM
- Print Fusing(s,"##.##") had a problem
- Mid(t\$(1),1,1) = "a" ' had a problem

#### 1.11.6.2

17-apr-2001

- IMPORTANT NOTE : since the HW stack, soft stack and framse are all selectable now, it means you need to make the settings for your existing programs. Look at STCHECK.
- Added CRC8 check :  
result = crc8(array(1),7) ' to look at first 7 bytes of array
- Fixed memory leak in the IDE.

- Changed bascavr.krf (help notes file)
- Added LOG() and EXP() function. Also corrected power bug with singles :  $2^{.5}$  for example did not work  
Using ^ with singles will use log() and exp() functions.
- Select case optimisation in 1.11.6.1 had a bug for select case with strings.
- Internal variables can be optional showed in the report.
- Fixed a problem with ^ for single variables.  $s^x$  did set X to 0.
- Report stays on top and is refreshed after compilation.
- Added STCHECK for checking stack sizes.
- rewrote the floating point > string routine so now it is compatible with the routine from BASCOM-8051. It gives more digits after the DP.  
It will use 24 bytes of framespace max.
- string concatenation with arrays and reusing the same string had an error.  
for example : ar(i)= "abc" + left\$(ar(i),x). Here ar(i) is used and overwritten.

#### 1.11.6.1

- Added waitkey() support for the software uart : var = waitkey(#1)
- Added SERVO driver implementation. See the servos.bas example.  
Control up to 16 servos!
- Added interrupt driven AT Keyboard read sample.
- Fixed a bug in Left\$() introduced in 1.11.6 beta
- Moved string compare to the mcs.lib and optimized the code. No local storage is used anymore for string-constant compares.
- The Lib had an error for registers. The incorrect address was used.  
When you use a register in a lib, you must include the register address like:  
.equ EEDR=\$1d  
This error caused the EPROM read/write routines in the lib to fail.
- Optimized SELECT CASE.  
When you need multiple test like:  
if a =1 then  
elseif a=2 then  
elseif a=3 then  
end if  
It is better to write :  
select case a  
case 1 :  
case 2 :  
case else  
end select  
It generates less code.
- The moved EPROM write and read routines had an error in the mcs.lib. The test for the EPROM size should be >256 instead of >255  
Also a label name was mistyped resulting in an error.
- The new waitms routine caused an error with debounce.
- The softclock routine works until 2099 now.
- Added copy to clipboard function to the Report window.

#### 1.11.6.0

- BETA
- Added MUL to simulator and it is used for the M161 and M163 that support the MUL instruction.
- Added some basic programming support for the STK500(serial programming).  
Last minute add so it only erases, programs and verifies.
- Added custom LCD libraries so you can change them when needed.  
Not all LCD related routines are in these libs yet(home etc)  
There are 2 versions :  
LCD4.LIB with the example LDCUSTOM4BIT.BAS This is an optimized version of the normal LCD routines but it uses fixed pins.

- LCD4E2.LIB with the example LCDCUSTOM4BITE2.BAS This is a version that supports 2 E lines for 40\*4 displays. It also uses fixed pins.
- Added active filename to the programmer.
  - Added time in mS to the cycle counter in the simulator
  - \$ROMSTART = &H4000 can be used to let the HEX file start at \$4000
    - Use with \$NOINIT to exclude int. vector setup. Can be used for bootloader
  - Added memory map variable display to the simulator Memory window.
  - ROTATE LEFT had a bug for words. Also optimized the rotate asm code.
    - Also optimized the SHIFT asm code.
  - Added a byte optimized version of the num<>str conversion routines.
    - This because the conversion routines are optimized for byte,int,word and longs. This means that when you use only byte conversion, there is overhead for the int and longs.
    - \$LIB "MCSBYTE.LIB" will reduce code size when only byte conversions are needed.
    - \$LIB "MCSINT.LIB" will reduce code size when no LONG conversions are needed.
  - WAITMS now accepts a WORD instead of a BYTE.
    - \$OLDWAITMS will use BYTE version of WAITMS since the wider range of the new WAITMS needs 2 instructions more to pass the value of the delay.
    - By using WAITMS 1000 you dont need WAIT 1 for example.
  - START AC and STOP AC did the reverse thing. START AC, stopped and STOP AC , started the AC comparator.
  - New entered crystal values are stored now.
  - sbci instruction was not simulated correct.
  - the sim did not cleared the I bit in an ISR.
    - MEGA chips return addresses were also not simulated correct.
  - using multiple {065} in a string constant did not worked. Only the first was changed to A in this example
  - optimised ON GOSUB, and also fixed problem that occured when used more than once.
  - \_CHIP , \_RAMSIZE , \_ERAMSIZE and \_XTAL constants added
  - Optimised BYTE varable code generation like : byte = num,byte = byte, byte=var(num)
  - Rewrote Str(), val() and Hex() and Hexval routines. They are in the lib now.
    - When using Hexval(), the input must be a multiple of 2
  - Added support for the M32 chip. I dont have info on this chip so please dont ask about it!
    - All I know it is M163 with 32K ROM,2K RAM and 1 K ERAM
  - deflng,defsnrg ??
  - Added LPM rx,Z and LPM rx,Z+ , ELPM rx,Z and ELPM rx,Z+
    - Can not be used on all AVR chips. In fact I dont know which one supports it :-)
    - But feedback from Atmel learned that it is almost certain that new batches will have support for these instructions
  - MEGA103 : Constants above 64KB were addressed 1 byte to high. Resulting in not showing first chars.
  - converting "12456" to a single gave wrong results : \_check0 routine changed in mcs.lib
  - Using STOP generated duplicate labels.
  - HIGHW() added.
  - CONFIG KBD = PORTB,DEBOUNCE = value (default 20)
  - SECTIC option added to the software clock. It can be used to jump into a user routine every time the second is updated:
    - CONFIG CLOCK=SOFT , GOSUB = SECTIC
    - You must must write a routine like this :
    - sectic:
      - Print "\*" '1 second past away
      - return
    - Note that since all registers are saved.
- 1.11.5.0 HELP FILE NOT UPDATED YET
- Added expressions to constants.
    - CONST a = 2
    - CONST b = a \* 2 + (10/2) ' for example
  - Added conditional compilation that can be used nested:

```

#if a = 2
    Print "this will be compiled"
#else
    Print "this not since a =2"
#endif

```

See const.bas example

- Programmer(s) (parallel based) did not work for low XTAL's like 1.8 MHz on the 2313.
- Now when you fill in the XTAL speed in the Compiler Communication options, a slower clock is used for xtals < 2 MHz.
- Simulator did not simmed ELPM instruction incorrect, resulting in strange behaviour when object code >64KB.
- Added HighW() function that returns the 2 MS bytes of a long.
- Adjusted lwire timings since DS2405 gave problems.
- Added lwVERIFY routine to verify if a lwire device exists on the bus
  - lwverify array(1) [,port,pin]
  - ERR will be set to 1 if the 8 byte number provided in array() does not exists
  - Otherwise ERR will be 0 indicating that is has been found
  - The routine calls lwsearchNext()
- The M163 had a problem with some rjmp's. THIS because memory wrap was enabled and the M163 does not need memory wrap for RJMP/RCALL.
- \$NORAMCLEAR will not clear the RAM content after a reset.
- The new lwire search routines contained some serious bugs.
  - I misinterepreted the algo so it seemed to work but not for many sensors.
  - The array that gets assigned must stay the same between a first and the next reads.
  - To solve this I use another 8 bytes of SRAM (\_\_\_lwire\_data)
- Forget to mention it in 1.11.4.0 but the CONFIG TIMER has a new option. It is in the help.
  - With the option you can set the OC pin for output or not.

1.11.4.0

20 dec 2000

- Added lwsearchfirst() function to scan for the first ID on the lwire bus.
  - Added lwsearchNext() function to scan for the next ID. Use lwsearchNext() in a loop to get all the devices until the first byte of the result is 0.
- Added lwirecount() to count the number of lwire devices on the bus.
  - See the lwireSearch.bas example.
- During startup, the basc-avr.dll version is checked. When it does not match you will get an error message. You must delete other versions of basc-avr.dll in that case.
  - Common an older version could be found in the windows\system dir.
  - Only a copy should be in the BASCOM application dir!
- \$WAITSTATE compiler directive will set the wait state and allow xram access.
  - It is the same as enabling these features from the Compiler Chip Options
- Added string=Fusing(single,mask)
  - Mask may be "#.##" or "#.&&" where # means round and & means no rounding
  - See help.
- Added check for wrong asm code sequence like sbic followed by lds ( a 2 word instrction).
- Added chip reset to the programmer.
- Labels and subroutines are now sorted in the IDE lookup lists.
- Finished M163 lock and fuse bit programming.
  - Since I have only 1 sample I did not test all the options since some can only be unprogrammed in parallel mode.
- Using BUS MODE for the LCD without XRAM access enabled will force xram mode now.
  - A warning is added to make the user aware of this fact.
- The new M163 has options for the reference voltage
  - For this chip you can use the additional param :
  - Config Adc = Single , Prescaler = Auto, Reference = Internal

The reference param may be :

OFF : AREF, internal reference turned off  
AVCC : AVCC, with external capacitor at AREF pin  
INTERNAL : Internal 2.56 voltage reference with external capacitor at AREF pin

Using the additional param on chip that do not have the internal reference will have no effect.

- The `_CHIP` constant in the report is an integer that represents the chip number.  
It is linked to the chip selection list where you select the chip. It starts with 0 for the 2313.  
It is used for conditional compilation in the library
- New M163 shown wrong rom image size in report.
- Added buffered serial reception.  
CONFIG SERIALIN = BUFFERED, SIZE = 20  
This will use an input buffer of 20 bytes.  
You must enable interrupts yourself because it is interrupt based.  
See the sample RS232buffer.bas
- Also added for output  
CONFIG SERIALOUT = BUFFERED, SIZE = 20  
This will create buffered output where the data is sent on the background when the UART transmitter is ready.  
See rs232bufferout.bas
- The AVR ISP programmer had a strange error : some code would pop up a Zmodem dialog window.  
Probably the object code triggered the zmodem condition.
- Added label option to READEEPROM and WRITEEEPROM. See help or EEPROM2.BAS example
- The parity was not handled correctly for O and E with the software UART. The bit counting/parity check routine was omitted in this version by mistake.
- using \$regfile for the new M163def.dat loaded the attiny.dat file because of the copy & paste, forget to edit problem
- Using CASE with a constant or variable beginning with IS failed because of the CASE IS test.
- Using E as variable failed because it was seen as a numeric constant.

1.11c

- 13 nov 2000
- \$serialinput2lcd did not work. Only the first char was displayed.
  - Added support for the M163. The M161 shows up too but since I do not have this chip yet, I can not support it yet. So the M161 WILL NOT WORK.  
The M163 is tested and works. The programmer is partially finished. Writing boot and fuse bits does not work.  
More over, I am not satisfied with the Options TAB yet so it will be changed.
  - Added TIME\$ and DATE\$ variables.  
Together with the CONFIG CLOCK = SOFT, you get a clock  
This works for the 8535, MEGA103 and M163 only. They have a special timer that can run async. with 32768 KHz.  
Date\$ format is MM/DD/YY , TIME\$ is hh:mm:ss  
See the megaclock.bas example
  - TIMER0 and TIMER2 for the Mega103/603 are now correctly handled.
  - TIMER2 for 8535 is now supported :  
CONFIG PRESCALE = 1/8/32/64/128/256/1024, CLEAR TIMER = 1/0, COMPARE= DISCONNECT/SET/TOGGLE/CLEAR, PWM=ON/OFF , ASYNC=ON/OFF
  - The M103 can be used for 128K now. The LPM instruction only works for the lower 64KB.  
ELPM is now used with the RAMPZ register for the MEGA103.  
The ELPM instruction is also recognized now by the compiler/assembler.
  - Added cpeekH() function to get a byte from code memory of a MEGA103.  
This chip has 64K Word memory. Cpeek() will peek into the lower 64KB and cpeekH() will look into the upper 64KB.
  - ASCII characters above 127 are converted wrong.  
è for example gets the wrong ascii value of 235.  
VB will do the same thing btw. So I am not sure if it is a bug. But it behaves like one.  
Possible for some fonts/language settings this bug does not occur.

Since I can not change it in the tools I am using to create the compiler I created a work around: s = "abc{137}" {ascii} will be converted to the proper value. The ð can be created with {239} for example.

See language fundamentals in the help.

- The cycles can be cleared now in the SIM by right clicking on the statusbar.
- Passing a value BYREF when the previous param was an array failed when a sub was called.
- All SRAM is cleared now when the micro is initiated.  
Variables don't have to be set to 0 or "" now before you use them.
- .equ can be used too now within BASCOM.
- Added bcd.lib and bin2bcd16.bas example.  
Received from a user and shows how to use functions from libs.

1.11b

- Added full support of 7, 8 bits and N, E or O parity to the software UART for both input and output.
- Programming >64K failed for the M103. You need to use the BIN file btw since the HEX files work up to 64KB.
- Using a loop with operator < resulted in an error because of duplicate labels were generated.
- Reported rom size was wrong. So check on code size also failed when it was 2 bytes oversized.
- Optimized some more code. This especially for chips with SRAM of up to 255 bytes.  
This means you can put more code in your 2313 now.
- Added auto Sim option. So you can simulate automatic after successful compilation.  
If auto program is enable too, the chip will be programmed first!
- Passing a numeric array to a sub/function passed the wrong base address.  
Passing ar(1) , actually passed address of ar(2). So there was an offset of 1.
- !!!! - DOWNT0 is rewritten. You can use TO no also for negative steps:  
For a = 10 TO 1 STEP -1 'for example  
X = -2  
For a = 10 TO 1 STEP X 'using a variable gives more code since the sign of the variable must be tested.  
Using a negative step with a variable only works for integer, long and single
- Using more than 10 software uarts gave an error because of a duplicate generated label.
- The terminal emulator stays on top now and the settings are accesable from the terminal window menu.
- When you empty the path in the IDE options, the location of the last opened file will be remembered now.
- When you use File, Save As .. and the name exists, you will get a warning if you want to overwrite it.
- referencing variables in ASM with {} did not report them as used in the report.
- ' did not work as comment in asm code.
- HEX files in the programmer may have split addresses too now.  
This way you can program chips with hex files generated by other tools.
- enable SPI must follow after the SPIINIT statement, otherwise SPI interrupt will not be enabled.  
Or you may add , INTERRUPT = ON after ths CONFIG SPI statement. When you use this additional parameter it does not matter in which order you use SPIINIT.
- Added PULSIN statement.  
Pulsein wVar, PINx, pin , state  
Where wvar is a word variable. PINx is PINB, PIND etc.  
pin is a numeric constant or variable for the pin to sample(0-7)  
State can be 0 or 1. 0 means sample on 0-1 transition. 1 means sample from 1-0 trans.  
Err will be set to 1 in case of a time out. wvar will be 0 in that case too.  
wvar returns time of pulse in units of 10 uS (approx.). Will work more accurate on higher xtals.  
For lower XTAL see mcs.lib source for suggested modifications.
- remark ' did reformat to position 1. It should not reformat when 'is the first position.
- forgot to include the bascavr.kf in the distribution
- Added error message when you assign a PINx register. These registers are read only.
- Probably of a hang up some old fixed things were not saved.  
The Optimise button was still not selectable.

1.11a - 1.11 could not be uploaded because of problems of my ISP.  
31 aug 2000 I added new things and checked some problems so here is 1.11a

- BAUD = 2400 will work on the hardware UART as normal.  
But added BAUD #x, yyy where x is the channel number and yyy is the new baud rate to use.  
use it with the software UART channels to change the baud rate at run time.  
It will use 2 bytes of SRAM to hold/load the bit delay time.  
CRYSTAL has been changed see this topic in the help file.
- PG302 programmer did not function because for some reason the baudrate has changed in my delphi form?
- Fixed LCD 16 \* 1A mode. It did not wrap to line 2.
- The CONFIG SPI = Hard ... did not set the SPE bit so hardware SPI did not work.  
You had to set it manual.
- Added ReadMagcard routine to read magnetic cards data.  
Both 5 bit and 7 bit coding are supported.
- The IDE now has an option to jump to a label or sub or function.  
When you type there are also pop up boxes with paramers. This can be switches off.  
It does not work 100% yet but it gives you a nice idea of how it can help you while you type.
- Added FORMAT() function for formatting numeric strings:  
s = Format(st, " +00000.00")  
When there are leading spaces, spaces will be added.  
When there is a + , the + will be added when there is no - sign.  
The number of zero's determen the number of leading zero's.  
Optional is the decimal point which you can insert into a number.
- Using DEBOUNCE on a Mega with port F failed because it has no DDR register.(F is always input)
- Compare pin A /B are set now for the 8535 and the MEGA and other parts. PrVIOUSLY only the pin on the 8515 was set.
- Changed lwire routines. Now they are in the mcs.lib. when multiple pins  
are used no extra code is needed anymore.(almost duplicate code was generated for each pin)  
But the syntax has changed:  
When only the default lwire pin is used the syntax remains the same.  
For the explicit pin specification for use with multiple pins, the syntax is :  
lwreset port,pin Where port is PINB or PIND or any other PIN register. pin is a pin number  
in the range from 0-7 and may be a constant or variable.  
For lwwrite with port option:  
lwwrite bytes,PINx,pin where PINX is PINB for example and pin is a variable or constant for the pin number  
For lwread:  
var = lwread(bytes,PINx,pin). See above
- Open can be used with 7 databits Odd and Even parity too now.
- Added Checksum() function that returns a checksum of a string.  
Dim S As string \* 10 , b as byte  
s= "hello" : b=checksum(s)
- The AT option (DIM ... XRAM AT) did not worked for the upper 32 KB.
- Added LET for compatibility with BASIC stamps : LET a = 1
- Mid() function did not addressed correct character with LOCAL strings.

1.11 @##@\$@ could not upload it because my ISP had troubles.  
And I had to leave for vacation. So I upload 1.11a

10 aug 2000

- Fixed problem in ISP programmer. Writing a buffer without Auto Program failed.  
Also writing a Mega chip failed when too many pages were written.
- Added delay to LCD init routine so the LCD has the time to reset.  
The AVR is running faster after a reset and this caused a problem in some cases with a soft reset.
- Inkey() for the software UART triggered on the wrong bit level.  
In effect making it not work for low crystal values. For higher XTAL's it worked.

- Added variable index to ERAM variable.  
So now it is almost a normal variable! You can do only assignments on them however.  
Dim Ar(10) As Eram Byte , Idx As Byte , B As Byte  
For Idx = 1 To 10  
    Ar(idx) = Idx  
Next  
For Idx = 1 To 10  
    B = Ar(idx)  
    Print B  
Next  
End
- added sincos.bas example.
- Optimize button was accidentally disabled. Must be only disabled for DEMO.
- END will disable interrupts and generate an endless loop.  
This has not changed. STOP will not disable interrupts now and will only generate an endless loop.
- Added warning when there is an END statement after the DATA statements.  
DATA lines should be the last in your program.
- Finished the SIM for the LCD and LED part.  
Some exciting things will follow too for the SIM.  
ERAM is simulated too now in the simulator.  
What remains is simulation of the timers.
- Added Real hardware simulation. Together with the BASMON.BAS file that you need for it. (in DEMO only available as compiled file for the AVR90S2313)
- Added mylib.lib and mylib.lbx to the LIB  
And libdemo.bas to the samples.  
It shows how to create your own ASM SUBS and FUNCTIONS
- Fixed FP compare routine.

#### 1.10

This release is made public to fix the I/O 998 error on some machines.  
The simulator part is not finished! The LCD simulation will work however.  
The led's dont reflect the right state! But I had to stop somewhere...

- In an interrupt routine you may use multiple Return statements now.  
But the first RETURN that is outside an IF .. THEN or other loop will generate a RETI (and will restore all used registers).
- File Save As, now also renames the CFG file so you don't loose your settings.
- Added TO to the CASE statement:  
Select case Var  
    Case 1 TO 10 : Print "A is 1-10"  
    case Else  
End Select
- Added multiple test :  
If a >10 AND a <15 OR A = 100 Then  
    'a is 11-14 or 100  
End If  
You can not use () and the tests are evaluated from left to right.
- PRINTBIN uses ; as a data seperator now just like B8051.  
It used , but that was wrong. , is for INPUTBIN.
- .equ is now also recognized.
- Mid(s,z,[num]) = "const" can now be used too.
- Floating point compare routine rewritten.

- Also fixed bug in FP addition with negative numbers.
- Simulator did not handle the stack pointer right.
- Also not all LOCAL vars and passed variables were shown.
- The CONFIG SPI= HARD mode contained a bug in the evaluation. Caused errors 12.
- The \$include directive did not put the right file number in the object file.
- And as a result the Simulator showed an error with an invalid filename.
- 1.0.09c - Aj, the b version contained a serious bug.
- 8 may 2000 The lib routines were not bind in LIB order.
- This caused out of range errors. So it was good to make a check for it!
- I also added checking for a , when needed but also put it on ASR.
- And ASR does not need a second param :-( So the LCD example for example gave errors.
- I also added line length of 1024 instead of 255.
- Lines longer than 255 caused a gpf in the reformatter routine.
  
- 1.0.09b - Added PULSOUT statement.
- PULSEOUT PORT, PIN, PERIOD
- Where PORT is PORTB, PORTD or any other port.
- PIN = the pinnumber (or a variable between 0-7)
- PERIOD = 1-65535. Each period will toggle the state for 1 uS with 4 MHZ clock
- You can insert NOP mnemonics for higher xtals too in the ASM source.(\_PULSE\_OUT)
- Optimized the ISP programmers : they should work much faster now.
- Finished the variable bit support. Now you can use : If Var.IDX = 1 then
- The SPI in hardware mode is full configurable now from the CONFIG SPI directive.
- Added var = INKEY(#1) support. So you can poll a software UART to see if there is data available.
- You could use an interrupt handler that triggers on the positive edge and that will get the data with INKEY(#x)
- Added feature to compile a LIB into object code.
- The compiled LIB has the extension LBX.
- A LIB can be compiled from the Tool LIB manager options.
- Works only in commercial edition.
- With the ability to create a LBX you can make add on libraries for BASCOM and sell the ASM source. The LBX has no comment and only the labels and jumps are in the source code format.
- Added range checking on assembler jumps like breq lbl.
- Also added check for using the right registers with instructions that use constants (like LDI reg,255)
- The software UART can use lower baudrates too now. I changed the timing to allow this.
- It also works with higher xtals and low baudrates now.
- Added interrupt simulation. You must press the button to generate an interrupt.
- Only the global int flag is checked and the individual enable/disable flags are not taken into account.
- Added CONFIG INTx = prm , support where x is 0 or 1.
- The parameter can be : LOW LEVEL , FALLING , RISING
- INT4-INT7 were already supported (on the MEGA 103 and 603).
- INT2-3 are always low level triggered! So there is no option available.
- Added warning to the report when a PORT or PORT PIN is assigned to a variable.
- You may read the PORT of course so that is why it is a warning.
- In 90% PIN or PINX.Y is ment.
- Rewrote \_Float2Long and \_Long2Float conversion routines. Now they work within the full range.
- Added Watch windows. In this window you add a conditon. when the condition is true the simulation halts. For example : var >= 2
- Added Local variable watch window.

- Variables will be shown automatic. So you can't add them.
- Added \$SIM option for simulator. This option will remove delays like WAIT and WAITMS for faster simulation.  
It may not be used for the final compilation otherwise there are no delays!
  - 123e4 (scientific format) is also recognized now
  - Added support for the KITSRUS programmer K122
- 1.0.09a
- Added OutW address,value to store a word/integer
  - Added InpW(address) to retrieve a word/integer
  - Inp() and Out work on bytes. InpW() and OutW work with integer/words
  - Added variable for bit operations on variables:  
Supported was a numeric constant like : SET var.1  
Now you can use a (byte) variable for the position:  
SET var.x  
Works with RESET and PRINT too. Compare on this type like  
IF var.x = 1 is not supported yet.  
The Simulator is getting shape now : you can change variables during simulation.  
The sessions variables will be saved too now.  
But it is not finished yet so some buttons will do nothing yet.
  - Conversion from an int/long to a float did not work for negative numbers. Fixed.
- 1.0.0.9
- The POWERSAVE statement was not recognized.
  - The ISP programmer had a bug with files > 64K.  
It also seems a bit slow compared to the ISP software from Atmel.  
I am researching this one.
  - New is the code optimizer. It only does a few optimisations yet.  
But it is a good start. Other code was optimized too.
  - I almost finished the simulator. It is not ready but since I am a bit late with this update I decided to show the current state.  
You dont need to send bug reports about it because it is not yet finshed!  
To display the BASCOM simulator right click the Simulator button.  
This will bring up the Sim window. Writing the object code simulator was a hard job so finishing the user interface will be easy :-)  
So within 1 or 2 weeks it will be 99% finished.  
Be sure to set the AVR Object code generation on otherwise it will not work.
  - The int/long conversion to a single has a bug that it does not work for negative numbers.  
Will be fixed soon.
  - Fixed bug in SPACE() function.
  - Added type conversion to SINGLE calculations.  
So you can use : single = word \* single 'for example!
  - Adci instruction was used but does not exist! Changed to Adc
  - Usage of . inside a string constant led to problems. It was recognized wrong as a floating point number.
  - Added GetATKBD() function to use an AT keyboard as an input device.
  - Fixed bug in displaying SINGLES. Number with a fraction were not displayed correctly.
  - Added Optimize option to the File generation options.  
But at this moment only a couple of optimisations are made.  
So the code optimisation is not completed!
  - Changed \$INCLUDE so it will use the same path as the main program when no path is specified.
  - Added External RAM Access option to the compiler settings. This to enable the

external RAM access even when you dont have RAM selected.  
You can must use this for INP() and OUT statements and for LCD in bus mode.  
Before this options was available you had to select XRAM even when it was not available.

#### 1.0.0.8c

- Optimized ON ... GOTO. It now uses a jump table which is more efficient.
- When the first single was assigned the value of 0.0 the compiler complained about \_\_\_NM\_XXXX value not found. Assigning it after it was 0.1 for example was not a problem.
- Added RND() function. Returns a random number.  
var = Rnd(limit)
- INPUTHEX and Hexval() had a bug that a value of 100 was not completely evaluated.
- Optimized DATA handling. DATA is stored sequential now in code.  
This saves bytes when bytes are stored(they use 2 code bytes before).  
The LookUp() And LookUpStr() functions are also optimized because data is stored better now.  
This also fixes a bug when string data was used in combination with bytes with the value 0.(it was seen as a string terminator)  
The largest value for the index can be 65535 now instead of 255. This for large DATA tables.
- MakeInt(LSB,MSB) had parameters reversed so it worked like MSB,LSB : fixed to LSB,MSB as stated in Help.
- CONFIG SPI showed CLK instead of CLOCK for the CLOCK parameter in the helpfile.
- By implementing the \_SubFromFrame routine I forgot to remove the previous code so calling a SUB from within a SUB with BYVAL, resulted in a bad frame with unexpected values.
- \$XRAMSTRING directive did not work properly because of a bug in the compiler I used.
- Added an option to IDE options to program automatic after succesfull compilation.
- Added SQR() routine to the FP routines.

#### 1.0.0.8b

4 feb 2000

- Fixed a bug in the INPUT routine. No LF was sent.
- Added VAL() and STR() for floating point. This took more time than planned :-(  
INPUT also works with SINGLE's now. So only trig function must be implemented now.
- Added routines so assigning and reading with STRINGS to/from EEPROM will take the length of the string into account.
- Added type conversion for singles between bytes,int,words and longs.  
See single.bas file.
- The reload value for the RC5 command was not correctly calculated for all XTAL's.
- When \$EEPROM was used to generate an EEP file, garbage was put in the file for some data types.
- Not a bug but a tip : When SHIFTIN is used, you must use PIN.x and not PORT.x for the input pin.  
The PORT returns the value that was last written to it, not the actual value on the PIN.  
The samples showed the wrong PORTB instead of PINB. This was a (paste) bug.
- A For..Next in combination with DOWNT0 and bytes or words for the loop counter, did not finished the loop completely. For a = 10 downto 1 'did go to 2!
- Optimized LOCATE statement. And added STK200 samples for pin and bus mode.

#### 1.0.0.8a

- DATA with strings and bytes caused a byte after an uneven number of characters not to be read.
- Assigning an EEPROM byte with a byte failed because I mistyped the register name.  
Caused error : Error : 200 Line : xxx .DEF not found [\_TEMP2[addk255]] , in File
- I published the changes but before sending a notification a bug was reported:  
I updated my Delphi4 to Delphi5 and it required me to change some variable types.  
But it had the side effect that the passed port pins for the LCD for example were null terminated.  
So the report did not show the info after the LCD pins.
- To compensate for the new download I added direct assignment to ERAM variables:  
Dim X As Eram Byte  
X = 12 'now works  
Previously you could only assign a byte to the ERAM var.
- I also added the Instr() function:  
pos = Instr([start , ] source, search)  
Where start is an optional starting position to search, source is the string to search

and search is the string or string constant to search for in string.  
When found pos will hold the first position the string was found, when not found will return 0.

- 1.0.0.8
- Added a simple LIB manager to the Tool Menu.
  - Passing an array to a SUB/Function did not worked right.  
See the declare.bas example for the right syntax.
  - Added SPIMOVE() function:  
result = SpiMove(value)  
value is a byte send to the SPI and result is the result received from the SPI.  
Sends/receives one byte only.
  - Spiout did not send out all the bytes. The pointer was not increased.
  - Printing a single <0 did not print the right value and trashed internal memory.
  - Reading a double port register such as CAPTURE1 read from wrong address.
  - Assigning one of the double hardware registers such as COMPARE1A resulted in a wrong address assignment. Also PWM1A was not recognized internal as a double HW register.  
These registers must be written with the MSB first and that is why they are defined in BASCOM as Word variables.
- 1.0.0.7c  
7 dec 1999
- Ai, I introduced another bug with the LIB support.  
I moved a subroutine and it depended on the wait subroutine below it.  
This caused the LCD routines to behave incorrect when printing a value because the delay did not walk to the wait routine but into another routine :-(  
Well in order not to have you download a new version each day I have tested the whole package again.
- 1.0.0.7b  
6 dec 1999
- There was an error in the \$LIB support. It included the routines multiple times when they were used more than once. And this caused an error message about duplicate labels.
  - I got some questions about passing arrays to subs. The declare.bas file has an example.
- 1.0.0.7  
5 dec 1999
- Added CONFIG INTx = prm , support where x is 4 to 7. The INT4 and INT7 are on the MEGA 103 and 603.  
The parameter can be : LOW LEVEL , FALLING , RISING
  - Made a begin with LIB support. This way you have the source ASM which you can change or by writing your own LIB you can override the MCS routines.  
In order for this new version to work, you must create a SUB directory named LIB in the BASCOM-AVR directory. The MCS.LIB file must be copied to this subdirectory  
With the \$LIB "mylib.lib" [, "other.lib"] you tell the compiler about your own library  
The MCS.LIB is always added by default and will be searched last.  
With the \$EXTERNAL myroutine1 [, myroutine2] you inform the compiler which routines must be included from the LIBRARIES.  
See HELPFILE for more details.
  - Added selectable colors for the editor. Comment can also displayed in Italic and statements can be displayed in bold.
  - Added support for Eddie McMullens SPI programmer.(Beta : untested)
  - Added waitus statement. It wait for the number of micro seconds. But it needs a constant!
  - The Auto Program option will now display a seperate window.  
When it is finished with programming a sound can be played. The sound must be specified from the Options Programmer menu.
  - The programmer option 'Auto Verify' will only verify when autoprogram is chosen.  
During programming a window is shown now and you can abort programming by pressing Escape.  
When verification fails the program icon will be shown in RED.

- Added PORTE and PORTF to the IDE(MEGA).
- SET/RESET in a function or SUB failed to assign.
- code for 4433 and 4434 chip recognition was changed by Atmel in the PDF. So I changed them too.
- Added part of the Floating point routines.  
FP unit is supplied by Jack Tidwell and only slightly modified for BASCOM-AVR.  
Jacks FP unit is the best FP unit around IMO.
- Added test for SHIFT and ROTATE statement. When the number of shifts was 0 a bad result was returned.
- Sample Electronics simple programmer tested and verified ok.  
This is just a simple cable programmer. But it works great!  
Look in the helpfile for more details.
- array(a+1) = array(b-1) is now possible
- Replaced 4433def.dat file since the bit definitions were incorrect.
- Added MEGA support:  
CLOCKDIVISION = value , where value ranges from 2-129  
A value of 2 will divide the XTAL frequency by 2 for example  
All INT's are supported now.(ENABLE INT4 and DISABLE INT5 for example)
- Added Getrc5() statement. See help or samples rc5.bas
- Accessing the COUNTER and other double hardware registers failed because the wrong order was used. For writing the MSB must be written first!
- Fixed LookupStr(). Did not function with value of 0.
- Exit Sub caused problem with LOCAL's.
- Added check for LOCAL's not used inside of IF THEN..
- Added parameter number check for SUBS/FUNCTIONS
- Flash Programmer had bad values for AT22 and Megal03.
- Flash programmer shows size of EEPROM too now.
- Optimized Hex() code.
- Added simple programmer cable support (Sample Electronics design). In Beta.
- Floating point is in Beta. You can use all math instructions but there is no input/output yet. So Print/input/val and str will not work.  
At the moment only:  
Dim S as single : s = 1.1 : s = s + 2.2 : Dim z as single : z = 5.12 : s = s \* Z  
So just basic math +-\*/  
You can verify output with AVR Studio and the hex content of the memory locations.  
So not really usable yet. Rest will follow soon.
- Fixed MEGA label calculation.
- Added support for the STK300 programmer.  
So the STK300 will work with the STK300 dongle.  
For some strange reason the STK200 dongle did not work.  
And they should be the same. When somebody knows the answer...
- Added support for ATtiny22(great tiny chip!).
- bcd() and Makebcd() did not work correct for all values.
- 2323 and 2343 chips were not recognized by programmer.(wrong signatures)
- I changed the byte order in the BIN files since they were reversed.  
I leave the swapping option since some programmers have an option for this.  
If no feedback about it I will remove the reverse byte option it in next release!
- I2CRECEIVE did not send ACK for receiving multiple bytes.
- Added \$LCDPUTDATA and \$LCDPUTCTRL compiler directives for LCD redirection.  
\$LCDPUTDATA = MyLCDDataOut 'will call MyLCDDataOut whenever data must be send to the LCD display.  
\$LCDPUTCTRL = MyLCDCtrlOut 'will call MyLCDCtrlOut whenever a control byte must be send to the LCD display controller.

1.0.0.6

19 oct 1999

- The data/control byte is available in register R24.
- Added delay for Software uart : CONFIG WAITSUART = mS  
This compiler directive will delay for the specified time in mS after the last byte is sent with the software UART.
- For the tiny 8 pin chips with internal osc. on 1 MHz there can be a problem with the software uart delay routine. This because the 1 MHz is not very stable. I added a reserved variable CRYSTAL which you can dimension as a BYTE and which can be changed at run time to change the baudrate:  
Dim Crystal As Byte  
Crystal = 66 'for 2400 baud at 1 MHz , 31 for 4800 baud, 14 for 9600 baud  
'use the above values as a starting point. And at run time just change the Crystal variable.
- Debounce did not work correct.  
The data direction is set in code but I can imagine that you prefer to set it yourself once. So feedback appreciated about this.
- Added COM5-COM8 to the environment options for the terminal emulator.
- Added CONFIG WAITLCD compiler directive.  
Specifying a value > 0 will result in waiting the number of specified mS after the last character has been sent.  
This is special user requested feature and is only usefull in some cases.
- Added CONFIG I2CDELAY compiler directive.  
CONFIG I2CDELAY = 1 'will load the value of 1 into the timing register  
Normally the XTAL is used to calculate the value automticly. But for faster I2C you can use your own parameter. 1 is the minimum value otherwise you get a loop of 255.
- Comparing bits resulted in Error 60
- MakeDec() function did not work for values > &H89
- LDS was not recognized by internal assmblar when used without !
- Added option to refer to the bitnumber of a bit.  
Bits are stored in bytes and when you must refer to a bit you must use:  
Dim B1 As bit  
Lds R16,{B1} 'load R16 with value of address of B1  
Sbrs R16, BIT.B1 ' BIT. followed by the variablename of the BIT will replace the BIT.B1 with the memory position of the bit in the byte.  
In this case since B1 was the first dimensioned BIT, it is 7  
So SBRS R16,7 will be generated.
- Byte array address was not calulated correct.
- Added sample for \$SERIALINPUT
- Added PushAll and PopAll statements to save/restore all used registers
- Added sample simm100.bas for new AVR SimmStick
- After Waitkey() or Inkey() the RX was disabled and not re-enabled.  
So INPUT did not work after this anymore.
- Added <file>.DBG output file for simulator testing and for next new feature.
- Added check to programmer for chip=chip to program  
This will prevent you from programming a chip with wrong target code
- Added LOG mode to the terminal emulator for logging output.
- added support for PRINTBIN with channel(any port pin)
- added support for INPUTBIN with channel(any port pin)
- serial port was initialized when using LCD statement. This caused serial portpin not to function for other pusposes.
- use of \$REGFILE generated wrong OBJ files for AVR Studio.
- Bit variables can be tested with BITWAIT too now.  
Dim B As Bit  
BitWait B, Set

- 'BUT take care that it will be set by your software, in an ISR for example otherwise you will have a loop that will not terminate.
- timer1.bas sample did not mention that timer needed a clock for the pwm mode CLEAR DOWN parameter generated wrong setting.
- 2333def.dat,4433def.dat, did not contain EEARL .equ needed by the compiler
- 4433def.dat file contained wrong flashend value. Should be FF instead of 7F
- Low() and High() functions optimized and fixed.
- WAITMS was not accurate enough. Changed. Because WAIT uses WAITMS it is also more accurate.

1.0.0.5

08/29/1999

- Some users need MEGA chip code so these chips can be selected now. This is NOT the official release for the MEGA support however since the generated code is not tested in all aspects.
- Added support for external programmers.
- Added option to generate BIN and HEX files with the LSB and MSB reversed. This is needed by some programmers.
- The DEFLCDCHAR statement displayed the first row on the last row.
- Added support for multi 1 wire devices. The following statements/functions have additional parameters:  
LWRESET , LWWRITE , LWREAD()
- Fixed integer mul with 2 integers. Returned wrong result for values > 255.
- Added check for 1wire being busy while giving a lwreset. Err will return 1 if it was busy.
- Added DAT files for the 90S2333 and the 90S4433
- Added code templates. Pressing CTRL+J will pop up the code template list. This way you can write standard code quicker.

08/15/1999

- Added PowerSave statement for 8535 chips.
- Added support for TIMER2 in 8535.
- Fixed Lookup() and optimized the code for lookup.
- Fixed SELECT statement. Parser generated an error.
- Fixed LONG division routine
- Fixed SHIFT and ROTATE RIGHT with LONGS.
- Printing a BIT variable printed reversed value.
- GetADC() returned value of previous conversion. Not documented in the pdf from Atmel but 2 conversions are needed for the right result.
- STK200 programmer did not verified last byte.

08/03/1999

- Replaced NT driver for parallel port access for a faster one.
- Added support for PG302 programmer.
- Added support for STK200 programmer.
- Fixed I2CREAD function.
- Fixed software UART when used with 2 stop bits
- Added GetKBD() function to scan a matrix keyboard. Config KBD= PORTB ' will define port B for the matrix key board ' The upper nibble of the port goes to the rows, the lower nibble of the port goes to the columns. var = getKBD() ' will return the key pressed. 16 if no key is pressed.
- Fixed ABS() function. It did not work when assigning to another variable.
- Fixed array problem with byte as an index.