V2.60 Commercial Release

- **Compiler**
  - Added support for the Ilitek ILI9325 graphic TFT LCD controller (only for Advanced license).
  - Improved the library for the Solomon Systech SSD1289 graphic TFT LCD controller.
  - Modified the alphanumeric LCD library (alcd.h) to be compatible with the Novatech NT3881 controller.
- **CodeWizardAVR**
  - Added the possibility to automatically set all the timer configuration registers for user specified operating mode, period and output(s) duty cycle.
  - Fixed: the Timer5 OC5A, OC5B, OC5C outputs were assigned to non-existent PORTM, instead of PORTL, for the ATmega640/1280/2560 chips.
  - Fixed: the OC4A, OC4B, OC4C OC5A, OC5B, OC5C outputs are not present for the ATmega1281/2561 chips.
  - Fixed: the ADTS0..3 bits in the SFIOR or ADCSRB registers were not set correctly, when using one of the ADC auto-trigger modes.
  - Fixed: the ADC auto-trigger modes were missing for the ATmega32 chip.
  - Fixed: USART code generated for the ATmega64 chip, used UMSEL00..UMSEL11 bit names instead of UMSEL0 and UMSEL1 in the UCSR0C and UCSR1C registers.
  - Fixed: ADC code generated for the ATmega64 chip, used the ADATE bit name, which is not present in the ADCSRA register of this chip.
- **Chip Programmer**
  - Added check for conflicts between the 'EEPROM|Program' and 'Preserve EEPROM' options.
  - Fixed error for restoring EEPROM contents after chip erase, when the 'Preserve EEPROM' option was enabled and Atmel Studio 6 was set as debugger.

V2.05.9 Commercial Release

- **Compiler**
  - Updated the SD Card library so that it will try to first #include the *ff.h* header file, if present, from the project directory. This allows to use custom settings for the file system for each project.
  - Modified the SD Card library to allow any polarity for the WP and CD SD card socket signals and also disabling their usage if needed. This is performed by using the *Project|Configure|C Compiler|Libraries|MMC/SD Card* menu.
  - Fixed a bug in the *sdcard_present* function from *sdcard.h* that prevented detecting the presence of some SD cards when the CD SD card socket signal was not used.
  - Updated the peripheral registers bit definitions for the ATtiny20/40/1634/261/461/861, AT90CAN32/64/128, AT90USB1286/1287, ATmega8U2/16U2/32U2/16U4/32U4/32U6/161/163/8535, AT86RF401 chips in the corresponding header files.
  - Added the \EXAMPLES ATxmega\SDCARD example for the ATxmega128A1 chip.
- **CodeWizardAVR**
  - Modified the CodeWizard for non-Xmega devices to generate code using symbolic names for peripheral configuration registers' bits.
  - Improved the CodeWizard's for non-Xmega devices user interface.
- **Chip Programmer**
  - Added the possibility to program the RSTDISBL fuse bit for the XMEGA chips.
  - Updated the chips signatures for the ATmega165A/325PA/3250A/645A/6450A and ATmega169A/329A/3290A/649A/6490A chips.
  - Added the possibility to enable or disable EEPROM programming when the 'Program All' function is used.

V2.05.8 Commercial Release

- **Compiler**
  - Added support for the ATtiny1634 chip
  - Added support for the Solomon Systech SSD2119 and Delcomp XG7100 color TFT graphic controllers (Advanced license required)
  - Added support for the Sitronix ST7920 graphic LCD controller
  - Improved the speed of the *glcd_clear* function for the SSD1289 and SSD1963 TFT controllers
  - Improved the speed of the horizontal and vertical line drawing for the SSD1289 and SSD1963 TFT controllers
  - Added the *cl_bits_order* member to the *GLCDINIT_t* structure for the SSD1289 and SSD1963 TFT controllers
  - Enhanced the code optimizer

- Fixed: the compiler generated hardware multiplication instructions for the ATmega8U2, ATmega16U2 and ATmega32U2 chips, which do not support them
- Fixed: assembly errors occured when using the *putchar* and *getchar* functions with the ATmega8U2, ATmega16U2 and ATmega32U2 chips
- Fixed: Due to a hardware bug in XMEGA D4 chips, writing to EEPROM trashed the first byte of the EEPROM page
- Corrected the definition of the frequency capture event from *TC_EVACT_FRW_gc* to *TC_EVACT_FRQ_gc* in all the *xmbits_nnn.h* header files for XMEGA devices
- Fixed: error message when using external RAM and specifying in the project configuration a heap size larger than device's internal RAM.

- **CodeWizardAVR**
  - Added support for the ATtiny1634 chip
  - Added in the CodeWizardAVR for XMEGA chips the possibility to specify, for the SPI operated in master mode, if the /SS signal will be used or not as chip select for peripherals connected to the SPI bus.
- **LCD Vision**
  - Added support for the SDD2119, ST7920 and XG7100 graphic controllers
  - Fixed: the Character Set grid was not cleared when creating/importing a font and characters from previous font remained there, if the character set range of the previous font was wider than the one of the new font.
- **Chip Programmer**
  - Corrected the signatures for the ATmega48A and ATmega168A chips.

V2.05.7a Commercial Release

- **IDE**
  - Fixed: on IDE startup the *Search|Find* menu and *Find* toolbar button were disabled until a compilation was performed.

V2.05.7 Commercial Release

- **Compiler**
  - Enhanced the code optimizer
  - Added support for the AVR Studio 5.1 and Atmel Studio 6 debuggers in the *Settings|Debugger* menu. Details about using CodeVisionAVR with these debuggers are available in the Help topics:
    - CodeVisionAVR IDE|Tools|The AVR Studio Debugger
    - CodeVisionAVR C Compiler Reference|Using the AVR Studio 4.19 Debugger
    - CodeVisionAVR C Compiler Reference|Using the AVR Studio 5.1 and Atmel Studio 6 Debuggers

    and in the chapters: 2.4.1, 3.21 and 3.22 of the CodeVisionAVR User Manual.
  - Added support for the SSD1963 color TFT graphic controller (Advanced license required)
  - Improved the speed of the SSD1289 graphic LCD library (Advanced license required)
  - Added function's memory address and size list to the .map file
  - Renamed the *adc* member of *GLCDINIT_t* structure to *reverse_x* in the header files for the SED1530, SPLC501C, ST7735 graphic LCD controllers.
  - Renamed the *seg_rev* member of *GLCDINIT_t* structure to *reverse_x* in the header file glcd_uc1701.h for the UC1701 graphic LCD controller
  - Renamed the *adc_rev132_x0* member of *GLCDINIT_t* structure to *rev132_x0* in the header files for the SED1530, SPLC501C, ST7735, UC1701 graphic LCD controllers
  - Added the *reverse_y* member to the *GLCDINIT_t* structure in the header files for the SED1530, SPLC501C, ST7735, UC1701 graphic LCD controllers
  - Added the *reverse_x* member to the *GLCDINIT_t* structure for the SSD1289 TFT controller. Replaced the *gate_scan* member with *reverse_y*.
  - Added transparency support for color images in graphics.h (documented in the Help and User Manual)
  - Improved the *delay_us* function (delay.h) so that short delays can be obtained even for low clock frequencies. Added a warning if the clock frequency is too low and the desired delay can't be obtained.
  - Added the *SCAN* (XMEGA ADC channel scan register) member to the *ADC_CH_t* structure in the xmstruct.h header file
  - Fixed: the *strlcpy* and *strlcpyf* functions (string.h) should copy maximum n-1 characters.
- **CodeWizardAVR**
  - Fixed: the CodeWizardAVR for XMEGA devices signaled that not enabled EBI /CS0../CS3 signals should be configured as outputs
- **LCD Vision**
  - Added support for the SDD1963 color TFT graphic controller

- Added scrollbars to LCD preview if the image doesn't fit in the docking panel
- Added the possibility to select opaque or transparent background when pasting during image editing
- Fixed: Image inverting was functional only for text inserting mode
- Fixed: Large font characters didn't fit in the preview window

- **Chip Programmer**
  - Added programming support for Atmel JTAGICE 3 (requires AVR Studio 5.1 or Atmel Studio 6 to be installed).
  - Fixed: incorrect FLASH page size for ATmega16U4 and ATmega32U4 chips, which lead to programming failure when using parallel port, STK500 and AVR910 programmers.

V2.05.6 Commercial Release

- **Compiler**
  - Enhanced the expression optimizer for cases when *ANSI char to int promotion* is enabled in the project configuration
  - Improved generated code when passing a struct/union with size 1, 2 or 4 as function argument
  - Added support for ATxmega64A3U, ATxmega128A3U, ATxmega192A3U, ATxmega256A3U, ATxmega256A3BU, ATxmega16A4U, ATxmega32A4U, ATxmega64A4U, ATxmega128A4U chips
  - Added support for the UltraChip UC1701 graphic LCD controller
  - Added in glcd_st7565.h, glcd_sed1530.h and glcd_spl501.h the adc_rev132_x0 member to the *GLCDINIT_t* structure, for displays that use reversed RAM column address driver (ADC=1) and the pixel with x=0 is connected to column driver #132
  - Added 102x64 display support for the ST7565 graphic LCD controller
  - Modified *ff.lib* so that LFN support can be enabled from *ff.h*
  - Modified *sdcard.lib* and the *Project|Configure|C Compiler|Libraries|MMC/SD/SD HC Card*
    menu to allow SD card sockets without a /CD signal to be used
  - Added the *sdcard_present* function in *sdcard.h* to check if a card is inserted, when the /CD card socket signal is not used
  - Changed the declaration of the *twi_init* funtion from *twix.h* to
    *void twi_init(TWI_t *module,bool ext_driver_intf,unsigned char sda_hold)* in order to be compatible with the ATxmega64A3U, ATxmega128A3U, ATxmega192A3U, ATxmega256A3U, ATxmega256A3BU, ATxmega16A4U, ATxmega32A4U, ATxmega64A4U, ATxmega128A4U chips
  - Improved the *twi_init* function (twi.h) for better handling recovery after a bus conflict
  - Modified the *snprintf, vsnprintf* functions (stdio.h) so that they will return the number of characters that would have been output, had the buffer been big enough (as required by C99). Previous versions returned the number of characters effectivelly written in the buffer (limited by the buffersize)
  - Fixed: RAM access code was generated when passing a struct/union located in FLASH or EEPROM as function argument
  - Fixed: bug in *glcd_sed1520.lib* that produced a compilation error for Xmega chips
  - Added missing TWI registers bit definitions in the *mega32u4_bit.h* header file
- **CodeWizardAVR**
  - Modified to generate the functions for external memory access, for graphic LCDs, only if the *Use Image Storage in External Memory* option is enabled
  - Removed the TWIE peripheral for the ATxmega256D3/192D3/128D3/64D3 chips, according to the errata from the latest Atmel datasheet
  - Fixed: for XMega chips the peripheral clock frequency (not the system clock frequency as is incorrectly specified in the current Atmel XMEGA A Manual Rev. 8077H-AVR-12/09) will be used for setting the value of the TWI baud rate register
  - Fixed: for XMega chips, in certain situations when using the differential input mode, the ADC positive and negative input selections were reset to 0, when switching the settings display between ADCA and ADCB
  - Fixed: for the ATtiny2313/4313 chips, when the Timer 0 OC0B output was used, no checks were performed if PORTD bit 5 was configured as output
- **LCD Vision**
  - Added support for creating, editing and converting graphic images
- **Chip Programmer**
  - Fixed: chip signature for ATmega328
  - Fixed: the WDP and WDWP Xmega fuse bits state was not correctly saved in the project file, when the option to program the chip after build was enabled

- Fixed: improper BODACT fuse programming for Xmega A chips, because of a mistake in Atmel XMEGA A Manual
- Fixed: the Xmega D chips don't have the JTAGEN fuse. The JTAGUID fuse bits were replaced with USERID fuse bits.

V2.05.5a Commercial Release

- **CodeWizardAVR**
  - improved the code generated for initializing the SPLC501C, SED1530, ST7565 and SSD1289 graphic LCD controllers

V2.05.5 Commercial Release

- **Compiler**
  - enhanced the code optimizer
  - added support for the SSD1289 graphic color TFT LCD controller (can be used only with an Advanced license)
  - added support for the ST7565 graphic LCD controller
  - made small optimizations in the SED1530 and SPLC501C libraries
  - fixed in the SPLC501C and SED1530 libraries: compile error when the D0..D7 signals were not set on the same I/O port in ascending order
  - fixed a bug in *glcd_line* function (graphics.h) which prevented drawing lines for y coordinates larger than _GLCD_MAXX_ value
  - rewrote the *glcd_floodfill* function (graphics.h) in order to significantly reduce Data Stack usage
  - added the *glcd_setlinethick*, *glcd_setfillcolor* and *glcd_getfillcolor* macros in *graphics.h*
  - added the *glcd_transparent* macro in *graphics.h* to enable/disable transparent text display mode for color displays
  - modified the declarations of *glcd_circle*, *glcd_arc*, *glcd_pieslice*, *glcd_fillcircle* functions (graphics.h) to support the new *GLCDRAD_t* data type (glcd_types.h) for radius parameter
  - renamed the *glcd_getlinewidth* function from *graphics.h* to *glcd_getlinethick*
- **CodeWizardAVR**
  - fixed: Division by 0 message when setting external clock value to 0 in CodeWizardAVR for Xmega chips
  - fixed: Timer 5 Clock Source: 'T5' pin name instead of 'T4'.

V2.05.4 Commercial Release

- **Compiler**
  - added support for the SPLC501C and SED1530 graphic LCD controllers
  - added the *f_mkfs* function (ff.h) for SD Card partitioning/formating
  - updated the '\EXAMPLES\SDCARD\SD Card Monitor' example
  - fixed: for the PCD8584, SED1335, S1D13700 and T6963C graphic LCD controllers the *glcd_putpixel* function was not clearing the pixel, if the foreground color was previously set to 0 using the *glcd_setcolor* function
  - changed .EQU __sm_mask=0xe0 to .EQU __sm_mask=0xf0 in the header files for the ATtiny4/5/9/10/20/40 chips
  - documented in the Help and User Manual the graphic library functions that are specific to the PCD8544, S1D13700, SED1335, SED1530, SPLC501C and T6963C controllers.
- **CodeWizardAVR**
  - fixed: in 16 bit operating mode, for certain required timer periods, the CodeWizardAVR generated incorrect values for the XMEGA Timer configuration registers when the Apply button was pressed.
- **IDE**
  - added the LCD Vision font editor for graphic LCDs.

V2.05.3a Commercial Release

- **Compiler**
  - improved code speed and size for the case: *signed_operand % 2^n*
  - added the PRR register in the *mega169p.h*, *tiny2313a.h*, *tiny4313.h* and *tiny4313_bits.h* header files
  - corrected several T6963C display's resolution from 160x60 to 160x80 pixels
  - fixed: due to the copy protection changes in V2.05.1, on some Windows installations, the linker sometimes issued an error about missing 'main' function
- **CodeWizardAVR**
  - fixed: the ATxmega64A3, ATxmega128A3, ATxmega192A3, ATxmega256A3 chips do not have DACA

- fixed: generated incomplete code for XMEGA ADC single ended input selection for inputs higher than 7

- **Programmer**
  - corrected the chip signature for ATmega325 and ATmega325V chips
- **IDE**
  - modified the licensing system to allow remote license activation/transfer/upgrade using the Internet

V2.05.3 Commercial Release

- **Compiler**
  - added support for graphic LCDs with 160x80, 160x128 resolution and Toshiba T6963C controller
  - added support for graphic LCDs with 180x32 resolution and SED1520 controller
  - modified the *glcd_putchar* function *(graphics.h)* so that it fills the spaces between the characters with the background color
  - added the *GLCD_CLEARBLOCK* and *GLCD_SETBLOCK* modes to the *glcd_block* function from the hardware abstraction layer for graphic LCDs
- **CodeWizardAVR**
  - fixed: when switching from the ATmega128 chip to ATmega103 the *Mode* options for *External Interrupts* 4 to 7 remained those of ATmega128
  - fixed: the comments for the code generated for disabling the *Digital Input Buffers* on AIN0 and AIN1 inputs were not correct for the ATmega48A/48PA/88A/88PA/168A/168PA/328/328P chips, altough the code itself was OK
  - fixed: entering an incorrect EBI base address resulted in an endless error message, without the possibility to correct it
- **Programmer**
  - corrected the chip signature for the ATmega88A chip

V2.05.2 Commercial Release

- **Compiler**
  - added support for graphic LCDs using the KS0108, SED1335, S1D13700, SED1520, T6963C, PCD8544 controllers, with code examples
  - added the _ENHANCED_FUNC_PAR_PASSING_ predefined macro to signal that the Enhanced function paramater passing option is enabled in the project configuration
  - added the AMISCR register in the *tiny87.h* and *tiny167.h* header files
  - fixed: the *memccpy* function from *string.h* now correctly returns a pointer to the next character in dest or a NULL pointer if the c character was not found in the first n bytes of src
  - fixed: incorrect .hex file was generated for bootloaders of XMEGA chips with 64k FLASH, if the source file of the bootloader contained an inline assembly.org directive with an address lower than 0x8000
  - fixed: "RAM location address is out of range" warning was issued in some cases when the @ operator was used
  - fixed: if the result of a *? :* operator, with operands 2 and 3 as floating point numerals, was assigned to an integer variable, it produced the value 0 instead of the truncated floating point
- **CodeWizardAVR**
  - renamed *TC_EVACT_FRW_gc* to *TC_EVACT_FRQ_gc* in the code generated for XMEGA chips and the *xmbits_a1.h, xmbits_a3.h, xmbits_a3b.h, xmbits_a4.h, xmbits_d3.h, xmbits_d4.h* header files
  - fixed: the generated code #included *mega48p.h* for the ATmega48A, respectivelly *mega48a.h* for the ATmega48PA chips
  - fixed: the ATxmega256A3B chip doesn't have USARTE1 and the PF5 pin
  - fixed: the ATxmega A4 chips don't have the DACA
  - fixed: the DIDR2 register was always set to 0 for the ATmega16U4 and ATmega32U4 chips, even if *Disable Digital Input Buffers* were set for ADC inputs 8 to 13
  - fixed: a code sequence was produced in the buffered USART receive ISR, that prevented the *Code Information|Functions* to correctly display functions that were defined after the ISR
  - fixed: the OVRTIM interrupt flag from the AT90CAN128's CANGIT register was cleared in the CAN_IT ISR, which prevented the CAN Timer Overrun ISR (CTIM_OVF) to be correctly serviced
- **IDE**
  - fixed: in some cases the Open/Save dialogs were displayed under the CodeWizardAVR window

- fixed: in some cases the I/O ports settings were not retained in the *Project|Configure|C Compiler|Libraries Bitbanged I2C* and *1 Wire* menus

V2.05.1b Commercial Release

- **Compiler**
  - fixed: if the *Enhanced Parameter Passing* option was enabled in the *Project|Configure|C Compiler|Code Generation* menu, the obsolete *lcd.lib* library functions didn't work correctly. The new *alcd.lib* library was not affected by this issue. This bug appeared in V2.05.1 compiler only.
  - fixed: the SCL bit rate for the bit-banged I2C functions (*i2c.h*) was significantly lower than the value specified in the *Project|Configure|C Compiler|Libaries|I2C* menu. This bug appeared in V2.05.1 compiler only.

V2.05.1a Commercial Release

- **Compiler**
  - fixed: if the *Enhanced Parameter Passing* option was enabled in the *Project|Configure|C Compiler|Code Generation* menu, a call to an **empty** function with the last parameter of 1..4 bytes in size, produced a data stack unbalance.
- **IDE**
  - added the option to exclude the *List Files (.asm, .lst* and *.map*) from the *Search in Files* and *Replace in Files* operations.

V2.05.1 Commercial Release

- **Compiler**
  - improved the expression optimizer for the situation when ANSI char to int promotion is enabled in the project configuration
  - enhanced the function parameter passing mechanism (the last function parameter with size 1..4 bytes is now passed using registers R24..R27). This can be enabled or disabled in the *Project|Configure|C Compiler|Code Generation* menu.
  - improved the error checking and reporting
  - added a TWI library (*twi.h)* that supports both master and slave modes
  - the PCF8563, PCF8583, DS1307, DS1621 and LM75 libraries can now be used with both bit-banged I2C and TWI. These libraries are now also compatible with XMEGA chips.
  - added the *week_day* parameter to the DS1307 library functions *rtc_set_date* and *rtc_get_date* (*ds1307.h*, *ds1307_twi.h*)
  - updated the DS1302 library to be compatible with XMEGA chips
  - added the possibility to configure the I/O port bits used by the I2C bus bit-banged library from the *Project|Configure|C Compiler|Libraries|I2C* menu, instead of using inline assembly like in previous versions of the compiler
  - added the possibility to enable/disable each linker warning in *Project|Configure|C Compiler|Messages*
  - increased the size of the compression table used when optimization for Size is enabled in the project configuration
  - updated the *tiny2313a.h* and *tiny4313.h* header files with the definitions for the PCMSK1, PCMSK2 registers and interrupt vectors PC_INT1, PC_INT2
  - updated the number of interrupt vectors for the ATmega644A/644PA chips in the compiler's internal tables
  - improved the alphanumeric LCD libraries (*alcd.h* and *alcd_ks0073.h*) to generate smaller code size, if the LCD data bits 4..7 are assigned to bits 0..3 or 4..7 on the same I/O port
  - fixed: memory access violation when inline assembly code was used to place a flash constant, accessible from C code, at a specific address and the *Project|Build* menu command was used instead of *Project|Build All*
- **CodeWizardAVR**
  - added DAC support for the XMEGA chips
  - added alphanumeric LCD controller type selection: HD44780 or KS0073
  - modified the code generated for TWI in order to support the new TWI library (*twi.h*)
  - modified the CodeWizardAVR for XMEGA chips to allow the event system to be configured to accept events from an I/O Port pin, even if it is set as an output
  - added support for the pin change interrupts of the ATtiny2313A and ATtiny4313 chips
  - fixed: the Timer5 OC5A, OC5B, OC5C outputs were assigned to PORTK, instead of PORTL, for the ATmega640/1280/2560 chips
  - fixed: the bit 0 (TOIE0) of the TIMSK0 register was not set when the overflow interrupt was enabled for Timer/Counter 0 of the AT90PWM2/3/216/316 chips

    - fixed: support for Timer/Counter 3 of the ATmega1284/1284P chips was missing because of an error in Atmel's datasheet 8272A-AVR-01/10
    - fixed: USART1 was missing for the ATmega644A/PA chips
    - fixed: the XMEGA A4 chips have only one ADC
    - fixed: the XMEGA A4 and D4 chips can use 12 ADC inputs in unsigned mode
    - fixed: the CodeWizardAVR didn't read correctly, from the previously saved project file, the event channel used to trigger a conversion for the XMEGA ADC
- **Chip Programmer**
    - corrected the FLASH page size for the ATmega8U2 chip (wrong value specified in Atmel's datasheet) which prevented correct FLASH programming using AVR910 and LPT port programmers
    - added the missing BODLVL2 fuse bit for the ATmega325A/325PA, ATmega3250A/3250PA, ATmega645A/645P and ATmega6450A/6450P devices
    - corrected the chip signatures for the ATmega325A/325PA/325V, ATmega 3250/3250V, ATmega645/645V and ATmega6450/6450V devices
- **IDE**
    - added: if an editor pane was previously maximized (in double pane modes), then pressing one of the *View|Dual Pane* buttons will restore the pane to it's previous size
    - added the *Project|Export* menu command
    - added in the Code Navigator a new node (*List Files*) for the *.asm, .lst* and *.map* files
    - added the option to create a new file and add it to the project's file list in the *Project|Configure|Files|Input* menu
    - added a status window, with the possibility to cancel the action, for the *Edit| Find in Files* and *Edit|Replace in Files* menu commands
    - added a function to scan the COM ports available in the system and list them in the Terminal and Programmer configuration windows
    - the *Project|Information* window now also displays the number of words reduced by code compression
    - added the option in the *Project|Configure|Before Build|Execute User's Program* menu to wait for the user's program to finish execution before starting the build process
    - fixed: if the Terminal was connected and *Settings|Terminal* menu command was executed, but the Cancel button was pressed, the Terminal was still disconnected, altough no changes were made to it's settings

V2.05.0 Commercial Release

- added ADC support for ATxmega chips in the CodeWizardAVR
- added support for the ATmega8U2, ATmega16U2, ATmega32U2, ATmega32A chips in the C Compiler, CodeWizardAVR and Programmer
- added support for the USB controller of the AT90USB82, AT90USB162, ATmega16U4 and ATmega32U4 chips in the CodeWizardAVR
- added a new alphanumeric LCD library (*alcd_ks0073.h*) for the Samsung KS0073 controller
- improved configuration settings for ATxmega EBI SDRAM mode
- disabled the possibility to specify the SCK clock rate for SPI slave mode in the CodeWizardAVR for non-ATxmega devices
- added code to clean up NVM controller command register after EEPROM write for ATxmega chips
- fixed: when the EBI was configured in SDRAM 3P mode, the upper 4 bits of PORTJ were not set as outputs for address lines A8..A11
- fixed: functions with *__reset* attribute were removed by the linker
- fixed: added code to ensure that RAMPX=0 after clearing/initializing more than 65535 bytes of RAM in the start-up code for the ATxmega chips
- added the missing definitions of EBI_SDCAS_bm, EBI_SDROW_bm and EBI_CS_SDSREN_bm used by the start-up code during EBI initialization for ATxmega chips
- corrected SP initialization for ATxmega chips when calling functions with *__reset* attribute and external RAM enabled
- fixed: the compiler could produce incorrect code (hardware stack unbalance) when adressing structure members in complex expressions using pointers, if the member's offset is higher than *64-sizeof(member)*
- fixed: because the ATxmega registers are not mapped to RAM, like in the rest of the AVR chips, the addresses of global variables allocated to registers R2..R14, obtained using the & operator, were not correctly passed as function arguments
- fixed: the CodeWizardAVR for ATxmega chips incorrectly set the upper 4 bits of the USART BAUDCTRLB register for negative values of BSCALE
- corrected the signature for the ATtiny4313 chip in the programmer

- added new ATxmega ADC examples in the *\EXAMPLES ATxmega\ADC* directory
- updated the Help topics: *CodeVisionAVR C Compiler Reference|Accesing the I/O Registers* and *CodeVisionAVR C Compiler Reference|Bit Level Access to the I/O Registers* with ATxmega specific examples.

V2.04.9a Commercial Release

- updated support for the ATmega165A/PA, ATmega325A/PA, ATmega3250A/PA, ATmega645A/P, ATmega6450A/P, ATmega48A/PA, ATmega88A/PA, ATmega168A/PA, ATmega328, ATmega169A/PA, ATmega329A/PA, ATmega3290A/PA, ATmega649A/P, ATmega6490A/P chips
- fixed: the ATxmega D3 and D4 chips have only 4 GPIORs

V2.04.9 Commercial Release

- added EBI support for ATxmega chips in the compiler project configuration (*Project|Configure|C Compiler|EBI Configuration*) and CodeWizardAVR
- added support for the ATxmega128D3, ATxmega192D3, ATxmega256D3, ATxmega64D4, ATxmega16D4, ATxmega32D4, ATxmega64D4 and ATxmega128D4 chips
- added a new alphanumeric LCD library (*alcd.h*) in order to support the ATxmega chips and allocate the LCD module signals to any pin of any I/O port. The signals allocation must be specified in the Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu. The new LCD library also contains the new function *putse*, used for displaying literal character strings located in EEPROM.The old LCD library (*lcd.h*) is maintained for code compatibility, but is not recommended for new projects.
- added the *strlcpy* and *strlcpyf* functions in *string.h* (safer versions of *strncpy* and *strncpyf)*
- added the *__reset* attribute to signal that a function must be executed imediatelly after chip reset
- made small improvements to the code optimizer
- added I/O port bit access macros: *iobits.h*
- fixed a bug in *sdcard.lib* configuration by the compiler, that prevented SD HC cards (4GB+) to be accessed by ATxmega chips that used other SPI than SPIC for communicating with the card
- fixed: when the /CS, /CD, /WP SD card library signals were set in the Project|Configure|C Compiler|Libraries|MMC/SD/SD HC Card menu to a bit which was used by the SPI, altough the I/O port was not the one allocated for the SPI, the bit was reset to 0 when the project configuration was opened again for modification
- fixed: not all the ATxmega I/O port pins were available for use as /CS, /CD, WP SDCARD signals in the Project|Configure|C Compiler|Libraries|MMC/SD/SD HC Card menu
- fixed: the CodeWizardAVR generated code for the ATmega640/1280/1281/2560/2561 chips' ADC accessed only the first 8 inputs
- fixed: the CodeWizardAVR generated code for the non-existant Timer/Counter 3 for the ATmega1284P chip
- fixed: when disabling the digital input buffers on analog comparator's input of ATmega1280/ 1281/ 2560/ 2561/ 128RFA1 chips, the CodeWizardAVR modified the DIDR0 register instead of DIDR1
- fixed: removed the TWID in the CodeWizardAVR for the ATxmega A3 chips
- fixed: clicking on a Find in Files node in the Code Navigator, moved the caret to the wrong line, if the line number was greater than 65536
- fixed: when an assembly error occured and the .asm file was already opened in another file pane than the active one, the file was reloaded in the active pane instead of the pane where it was already opened
- fixed: when a source file was modified and the project built, and after that all the changes in the source file were undone (using Edit|Undo) the file Modified status was cleared altough the file remained modified

V2.04.8a Commercial Release

- fixed in the C compiler: when one of the operands of the & binary operator was the numeral constant 1, the result was always assumed to be of 8 bit _Bool type, even if the other operand was 16 or 32bit in size. When the result was shifted left more than 8 bits, using the << operator, this lead to the result to become 0, a warning "shift result will be 0" beeing however generated.
- improved the SD Card library compatibility with ATxmega chips
- fixed in the CodeWizardAVR for ATxmega: when no TWI interface was enabled, the *twi_init* function was called, but *twix.h* was not #included
- fixed in the CodeWizardAVR for ATxmega: The ATxmega A4 chips don't have TWID
- fixed in the IDE: Search|Find in Files failed when the searched string was found in a *Read-Only* file that was not previously opened in the Editor.

V2.04.8 Commercial Release

- added library functions (*twix.h*) for ATxmega Two Wire Interface (TWI) support
- added TWI support in the CodeWizardAVR for ATxmega devices
- added TWI master and slave code examples for ATxmega devices
- increased the number of macros that can be defined in a program module to 10000
- fixed: in the MEDIUM memory model, arrays of pointers to functions stored in FLASH, were incorrectly indexed on 16 bits, instead of 24bit
- fixed: in certain situations, a linker warning/error "external ... declared, but never defined" was generated for an external global symbol that was defined in inline assembly
- fixed: if the editor pane was too small/inactive, clicking on an error/warning didn't make that pane visible/active.

V2.04.7a Commercial Release

- added an error message for the case when the *&* unary operator was applied to *sfrb* or *sfrw* I/O registers of ATxmega or reduced core chips (ATtiny4/5/9/10/20/40), because these registers are not mapped to RAM for these chips

V2.04.7 Commercial Release

- added ATxmega support for the 1 Wire functions
- the 1 Wire functions must be now configured using the Project|Configure|C Compiler|Libraries|1 Wire menu. There is no need to use inline assembly configuration anymore. Details are available in the updated Help topic and User Manual chapter 4.14.1: *1 Wire Protocol Functions*
- improved generated startup code size when global bit variables were declared in the program and the Project|Configure|C Compiler|Clear Global Variables at Program Startup option is disabled
- improved the code optimizer
- improved error checking and reporting
- the *&* unary operator can now be applied to I/O registers declared using the *sfrb* and *sfrw* keywords. It will return a pointer to the RAM address where these registers are mapped
- added support for the ATtiny4, ATtiny9, ATtiny20, ATtiny40, ATtiny2313A, ATtiny4313, ATmega128RFA1, ATmega164A/164PA/324A/324PA/644A/644PA/1284, ATmega169A/169PA chips
- renamed the PRR0 register to PRR in the *mega1284.h* and *mega1284p.h* header files
- added *.EQU __sm_adc_noise_red=0x02* in the header files for the ATmega16U4 and ATmega32U4 devices
- fixed: wrong initialization occured, if global bit variables initialized during declaration were located in both GPIOR and R2..R14 registers
- fixed a bug in *ftoa* (stdlib.lib) which lead to function locking when displaying values larger than 1e+38
- fixed: global pointers, stored in registers, were not correctly initialized during declaration for the ATxmega chips
- fixed: the compiler couldn't display correctly linker warnings for symbols used in library files
- fixed: in certain cases the MSB of ATxmega 16 bit I/O registers was written first (instead of the LSB). Example: RTC.CNT += 1
- fixed: no error was signaled when trying to assign a value to a struct/union member which was declared as constant, the struct/union itself not beeing declared as constant
- updated the *C Preprocessor* Help topic and User Manual chapter 3.1 with the newly added predefined macros: _EXTERNAL_STARTUP_ , _SRAM_START_ , _SRAM_END_ , _DSTACK_START_ , _DSTACK_END_ , _ATXMEGA_DEVICE_
- added new examples for ATxmega chips: Atmel application note AVR1300 “Using the ATxmega ADC”, using the DS18B20 temperature sensor and a procedure for writing the EEPROM for ATxmega256A3, ATxmega256A3B Rev.B which present a bug in silicon
- added 1 Wire support in the CodeWizardAVR for ATxmega devices
- added in the CodeWizardAVR for ATxmega digital filter code generation for Event Channel inputs
- modified the CodeWizardAVR initialization sequence for ATxmega I/O port pins used for USART TxD and XCK outputs in order to prevent generating a false start bit during USART initialization
- corrected the port and bits used by the CodeWizardAVR for the analog comparator and ADC of the ATmega1280/1281/2560/2561/640 chips
- fixed: the CodeWizardAVR didn't initialize correctly the ADMUX register with the voltage reference type for the ATtiny24/25/43U/44/45/48/88 chips

- added USARTF0 in the CodeWizardAVR for the ATxmega64A3/128A3/192A3/256A3/256A3B chips
- fixed: display correctly the note regarding the ClkPer output in the CodeWizardAVR for ATxmega devices
- fixed: the CodeWizardAVR for ATxmega chips generated incorrect comments for Event System Channels 1, 2 and 3 sources
- added AVR Dragon programming support for the ATxmega chips in JTAG mode
- added AVRISP MKII and STK600 programmer support for the ATtiny4/5/9/10/20 chips
- corrected the signature for the ATmega1284P chip
- updated the values for ATxmega BODLEVEL fuse bits in the chip programmer and project configuration so they match with AVR Studio 4.18 SP2
- implemented a split pane editor that allows two files to be edited at the same time
- improved the IDE compatibility with Windows 7
- changed the File|Close shortcut to Ctrl+W because it conflicted with Ctr+F4 for Project|Go to Previous Warning menu
- improved the Edit|Uncomment menu functionality

V2.04.6 Commercial Release

- improved the IDE compatibility with Windows 7
- enhanced the IDE windows docking manager and added various small improvements/fixes
- improved error and warning reporting in the compiler
- added CodeWizardAVR support for the RTC, RTC32, Battery Backup and SPI ATxmega peripherals
- added support for the ATxmega256A3B chip
- updated the definition for DMA_CH_struct in the *xmstruct.h* header file
- corrected the definition for NVM_PROD_SIGNATURES_struct in the *xmstruct.h* header file
- modified the Standard C I/O *printf* and *sprintf* functions from *stdio.h* to display the exponent floating point format as [-]d.dddddd e[±]dd or [-]d.dddddd E[±]dd
- added a 5us delay after executing the *i2c_write* function from *i2c.h* in order to improve I2C bus communication reliability
- added in the project configuration menu the possibility to use a two times slower *SCK* rate when communicating with SD Flash memory cards
- added in the project configuration menu the possibility to use an active low *Write Protect* signal when interfacing with SD Flash memory cards
- fixed in the CodeWizardAVR for the ATxmega chips: when the *External Clock* value was changed, the *ClkPer* and *ClkCPU* labels were not updated
- fixed in the CodeWizardAVR for the ATxmega chips: the TCE0 timer doesn't have AWEX for the A4 chips
- removed spike detection reset test in the CodeWizard for the ATxmega chips in order to comply with the ATxmega A manual rev. H 12/09
- added PDI ATxmega programming mode for the Atmel STK600 programmer
- added ATxmega programming support for the Ateml AVRISP MKII USB programmer
- fixed in the programmer the BOOTSZ0, BOOTSZ1, BODLEVEL0, BODlEVEL1 and BODLEVEL2 fuse bits positions for the ATmega328P and ATmega328PV chips
- fixed in the programmer the signature for the ATxmega32A4 chip

V2.04.5b Commercial Release

- added in *Settings*|*Editor*|*Text* the possibility to chose the background color of the selected code template
- added the possibility to remember the default working path when loading new projects
- fixed the registry write error when running for the first time under Windows 7

V2.04.5a Commercial Release

- small improvement in the license validity checks

V2.04.5 Commercial Release

- added support for the ATxmega devices in the CodeWizardAVR
- updated the ATxmega devices header files according to the latest datasheets
- addded support for the ATmega16HVA, ATmega8HVA, ATmega16HVB, ATmega32HVB and ATxmega324A1 devices
- updated the C compiler and CodeWizardAVR according to the latest ATtiny20 datasheet

- updated the *90usb82.h* and *90usb162.h* header files with the definitions of the WDTCKD and REGCR registers
- updated the *tiny87.h* and *tiny167.h* header files with the EEARH register definition
- added new compiler directive: *#pragma optsize_default*
- added the possibility to select, in the project configuration, the SPI controller used for FLASH memory card access by the ATxmega chips
- made some improvements in the compiler's code generator when producing code that uses the MUL and MULS instructions
- improved the code generated by the CodeWizardAVR for buffered interrupt driven serial communication for the case when the receive/transmit buffer size is 256 bytes
- improved compiler error checking and reporting
- improved the initialization of the CAN controller for the AT90CAN32/64/128 chips, in the CodeWizardAVR generated code
- added the TWI bus connect/disconnect interrupt service routine in the CodeWizardAVR generated code for the ATmega406 chip
- fixed: the \e escape sequence was not recognized
- fixed: replaced \0octal_number escape sequence with \octal_number in order to comply with C99 standard
- fixed: in the MEDIUM and LARGE memory models (for chips with >=128kbytes FLASH) functions expecting a 'flash' array parameter, performed the array indexing with a variable index on 16bits instead of 24bits
- fixed: for the ATxmega chips, writing the EEMAPEN bit from the NVM.CTRLB register doesn't require configuration change protection, so the startup code after reset was modified so that the CCP register will not be written when setting this bit
- fixed: updated *stdio.lib* to remove assembly errors that occured when the *putchar* and *getchar* functions were used for the ATmega16U4, ATmega32U4 and ATmega32U6 chips
- fixed the assembler errors which occured when variables located in EEPROM were used with the ATtiny87 and ATtiny167 chips. The EEARH register definition was missing from the generated .asm file.
- fixed: reversed the text in the CodeWizardAVR SPI tab Clock Phase selection box to: Cycle Start/Cycle Half
- fixed: set the correct model for ATxmega bootloaders depending on FLASH size:
  - SMALL for FLASH size <64k
  - MEDIUM for FLASH size =64k
  - LARGE for FLASH size >64k

V2.04.4a Commercial Release

- fixed: the following functions from *string.h* employed the LPM instruction, instead of the correct ELPM, when used in boot loaders for chips with 128kbytes or more FLASH and the SMALL memory model: *strcspnf, strlenf, strpbrkf, strrpbrkf, strstrf, strspnf*
- fixed a small typing error which could lead to assembly errors when initializing, during declaration, bit variables located in GPIOR registers.

V2.04.4 Commercial Release

- added new MMC/SD/SD HC FLASH Memory Card drivers (*sdcard.h*) and FAT12, FAT16, FAT32 access libraries (*ff.h, stdio.h*) with several examples, which were tested on the Mega128 MMC development board from Progressive Resources LLC
- added in the *\EXAMPLES ATxmega\AVR1605* directory, the port to CodeVisionAVR of the Atmel Application Note AVR1605 ATxmega Boot Loader
- the Standard C I/O Functions (*stdio.h*): *printf, sprintf, vprintf, vsprintf, snprintf, vsnprintf* were redefined and now return the number of outputed characters
- added specific FAT file access functions in the Standard C I/O Functions (*stdio.h*) library: *fgetc, fputc, fputs, fputsf, fprintf, fscanf, feof, ferror*
- improved error checking and reporting
- improved the generation of startup code for the ATxmega devices
- added the predefined preprocessor macro *_DSTACK_SIZE_* which specifies the size of Data Stack set in the project configuration
- fixed a bug in the code optimizer which, in some rare cases, could lead to a memory access violation error

V2.04.3a Commercial Release

- added support for the ATtiny5 chip (AVR Studio 4.17 must be installed)

- the IDE now saves/restores the editor tabs order, if these were re-aranged
- improved compatibility with version control systems that set the read-only attribute of the .prj project file

V2.04.3 Commercial Release

- improved the code generator and optimizer
- improved the floating point library for addition and substraction
- improved linker error and warning reporting
- added support for the ATtiny43U chip in the compiler, CodeWizardAVR and chip programmer
- added support for the ATtiny167, ATtiny87, ATmega16M1, ATmega32C1, ATmega32M1, ATmega64C1 and ATmega64M1 chips in the compiler and chip programmer
- updated the bit definitions in the header files for the AT90USB1286/1287/646/647 chips
- ATxmega virtual port registers are now accessed using the IN and OUT instructions, not LDS and STS like in previous versions
- improved interrupt service routine code generation for ATxmega chips for the MEDIUM and LARGE memory models
- fixed: the RAMPZ register wasn't set to 0, for ATxmega chips with >=128k FLASH and MEDIUM or LARGE memory models, when the Z register pair was used for RAM access.
- added ATxmega chip programming support for the Atmel JTAGICE MkII (requires AVR Studio 4.16.638 or later to be installed).
- added the *Edit|Go to Definition/Declaration* menu command in the IDE
- added various small improvements in the IDE
- updated the Help topic and User Manual chapter regarding *Structures*

V2.04.2c Commercial Release

- improved linker error checking for memory overlaping of variables placed at absolute addresses using the @ operator
- improved linker detecting of recursive function calls when estimating the Data Stack usage
- fixed: calling a function by casting an integer numerical literal constant with the value above 0xFFFF, in the LARGE memory model (chips with 256kbytes of FLASH), didn’t set correctly the value of the EIND register before the EICALL instruction
- small improvements in the IDE

V2.04.2b Commercial Release

- improved compiler error checking and reporting
- added support for the ATtiny13A chip (tiny13a.h header file)
- fixed a small bug in the syntax highlighting of comments
- increased to 256 the number of user defined keywords that can be syntax highlighted

V2.04.2 Commercial Release

- improved compiler error and warning checking
- added additional warnings for cases when the << operator may produce an overflow
- improved code optimizer
- small improvements in the C preprocessor
- enhanced linker checking for recursive function calls
- added support for the ATmega16U4 and ATmega32U4 chips
- fixed: in certain situations the assignment of the contents of a bit field, located in the MSB of an int, to the element of a char array, which was the member of a structure, produced incorrect results
- fixed: in some rare cases the R15 register was saved/restored in interrupt service routines even if it wasn’t used
- fixed: in order to eliminate naming conflicts with I/O registers bits definitions, the names of the **SPCR0, SPDR0, SPSR0** registers **were changed** to **SPCR, SPDR, SPSR** in the following header files: mega164.h, mega324.h, mega644.h, mega644p.h
- fixed the interrupt vector numbers for the Analog Comparators on PORTA and PORTB in the following header files: xmega64a1.h, xmega128a1.h, xmega192a1.h, xmega256a1.h, xmega64a3.h, xmega128a3.h, xmega192a3.h, xmega256a3.h
- fixed the interrupt vector numbers for the Analog Comparator on PORTA in the following header files: xmega16a4.h, xmega32a4.h xmega64a4.h, xmega128a4.h
- corrected the definitions for AC_MUXPOS_PIN3_gc ... AC_MUXPOS_PIN7_gc in the *xmbits_a1.h , xmbits_a3.h* and *xmbits_a4.h* header files for the ATxmega devices

- fixed: when the ATtiny10 Timer0 was configured in CTC or PWM modes and the OC0A/OC0B outputs were used, the CodeWizardAVR didn't perform any checks that the OC0A/OC0B pins were configured for output in the DDRB register
- added in the CodeWizardAVR the possibility to configure the Timer/Counter1 Dead Time register DT1 for the ATtiny261, ATtiny461 and ATtiny861 chips
- fixed: the SPIEN fuse bit can now be programmed in JTAG mode using the JTAGICE MkII, AVR Dragon and STK600 programmers
- improved Terminal error handling when using USB COM port addapters
- editor tabs can now be re-arranged using drag and drop
- the file close button is now placed on the active editor tab
- added a drop down button in the tab page control to allow selecting the active editor file
- made small bug fixes and improvements in the IDE

V2.04.1 Commercial Release

- fixed: in certain situations the linker could allow a variable, placed at an absolute address using the @ operator, to overlap another variable for which the @ operator wasn't used
- fixed: when using the LARGE memory model (for chips with 256kbytes of FLASH only), calling a function using a pointer to function, could sometimes result in a hardware stack unbalance, if the function would take as argument the address of a char or int local variable located in a register
- fixed: when a numeral with the value in the 0..31 range was casted to a pointer to a *volatile* char or int, dereferencing the pointer generated register R0..R31 access instructions, instead of the required LDS or STS. This created problems when accessing I/O registers, with addresses in the 0..31 range, for the ATxmega chips.
- removed the limitation: when a literal char string enclosed between quotes, is passed as an argument to a function parameter of pointer to char, the pointer now can point to any memory type: RAM, EEPROM or FLASH, not only FLASH like in previous versions. The *CodeVisionAVR C Compiler Reference*|*Constants* Help topic and User Manual chapter were updated to reflect this change.
- enhanced the code optimizer
- improved the linker: RAM is not allocated anymore for static variables defined in functions that are not used in the program and are removed by the linker
- improved the warning reporting for expressions with possibly no effect
- significantly improved speed for Project|Compile and Project|Build
- modified the Power Management functions (*sleep.h*) in order to ensure that interrupts are globally enabled before entering any power reduction mode. The global interrupt enable state is restored to it's initial value after exiting from any power reduction mode.
- updated the *mega162.h* header file so that the powerdown mode will be supported for the ATmega162 chip
- I/O registers bits definitions were added to the device header files. For projects created with prior versions, these definitions are not enabled by default. In order to enable them the *Project*|*Configure*|*C Compiler*|*Code Generation*|*Preprocessor*|*Include I/O Registers Bits Definitions* option must be activated.
- in order to eliminate naming conflicts with I/O registers bits definitions, the names of the **PCINT0, PCINT1, PCINT2, PCINT3** interrupt vectors **were changed** to **PC_INT0, PC_INT1, PC_INT2, PC_INT3** in the following header files: 90usb1286.h, 90usb1287.h, 90usb646.h, 90usb647.h, 90usb162.h, 90usb162.h, mega1280.h, mega1281.h, mega2560.h, mega2561.h, mega640.h, mega1284p.h, mega162.h, mega164.h, mega165.h, mega168.h, mega168p.h, mega169.h, mega324.h, mega325.h, mega325p.h, mega3250.h, mega3250p.h, mega328p.h, mega329.h, mega329p.h, mega3290.h, mega3290p.h, mega406.h, mega48.h, mega48p.h, mega644.h, mega644p.h, mega645.h, mega6450.h, mega649.h, mega6490.h, mega88.h, mega88p.h, tiny10.h, tiny13.h, tiny24.h, tiny25.h, tiny44.h, tiny45.h, tiny48.h, tiny84.h, tiny 85.h, tiny88.h, tiny261.h, tiny461.h, tiny861.h, tiny2313.h . **If these interrupt vectors are used in your programs, their names must be updated.**
- in order to eliminate naming conflicts with I/O registers bits definitions, the names of the **INT0...INT7** interrupt vectors **were changed** to **EXT_INT0...EXT_INT7** in the following header files: mega1280.h, mega1281.h, mega2560.h, mega2561.h, mega640.h, mega1284p.h, mega164.h, mega324.h, mega644.h, mega644p.h. **If these interrupt vectors are used in your programs, their names must be updated.**
- fixed: if the serial communication Terminal was active and changes were made to it's configuration, these changes were applied only after the Terminal was closed and reopened
- added the following shortcuts to the tabbed editor interface: *Ctrl+F4* - close current editor tabbed window, *Ctrl+Tab* - switch to the next editor tabbed window, *Ctrl+Shift+Tab* - switch to the previous editor tabbed window

- added the *File*|*Close Multiple* files menu command
- improved the *Project*|*Clean Up* function
- added the *Project*|*Check Syntax* menu, that performs a faster syntax check for the currently edited file. This function can be also executed from the popup menu that is invoked by right-clicking in the editor window.
- added the *Settings*|*IDE* menu that allows, among others, to disable the hints in the Code Navigator, Code Information and Function Call Tree windows
- improved the *Search*|*Find in Files* and *Search*|*Replace in Files* menu functions: all the project files are not automatically opened during multiple file search/replace anymore
- made some minor bug fixes and improvements in the IDE
- improved the CodeWizardAVR: it's window is not modal anymore
- changed: now the editor caret moves to the upper/lower row when it is located at the begining/end of the line and the left/right arrow keys are pressed
- documented in the Help and User Manual: right clicking on the down arrow located on the right of each toolbar, opens a pop-up menu that allows to individually show/hide the toolbars
- documented in the Help and User Manual topic: *CodeVisionAVR C Compiler Reference*|*Limitations* that the %s format specifier used by the Standard C I/O Functions : *printf, sprintf, snprintf, vprintf, vsprintf* and *vsnprintf* can't allow to output char strings longer than 255 characters

V2.04.0a Commercial Release

- minor tweaks in the copy protection in order to ensure better compatibility with some Windows Vista installations
- minor fixes in the menu toolbars

V2.04.0 Commercial Release

- redesigned and enhanced the IDE
- CodeVisionAVR is now compatible with Windows XP 64bit and Vista 64bit
- reduced the number of work files created by the compiler and IDE
- enhanced the editor's function parameters and structure/unions members autocomplete
- improved the compiler's store/load optimizer
- improved the linker when allocating global variables to registers
- added support for the ATxmega64A3, ATxmega128A3, ATxmega192A3 and ATxmega256A3 chips
- changed the name of the General Power Reduction register from PR to PRGEN in the *xmstruct.h* header file
- added the definition for the AES bit of the PRGEN register in the *xmbits_a1.h* and *xmbits_a4.h* header files
- corrected the address of the IRCOM register in the *xmega64a1.h, xmega128a1.h, xmega192a1.h* and *xmega256a1.h* header files
- corrected the bit positions of CLKOUT and EVOUT bits in the PORTCFG register in the *xmbits_a1.h* and *xmbits_a4.h* header files
- corrected the *__ds1820_scratch_pad* structure definition from the *ds1820.h* header file, in order to prevent the failure to detect a DS1820 chip by the *ds1820_init* function when ANSI char to int promotion was enabled in the project configuration
- corrected the *__ds18b20_scratch_pad* structure definition from the *ds18b20.h* header file, in order to prevent the failure to detect a DS18B20 chip by the *ds18b20_init* function when ANSI char to int promotion was enabled in the project configuration
- fixed: the result of *sizeof(array + scalar)* or *sizeof(array - scalar)* was the size of the array instead of the size of a pointer to that array
- fixed: the boot loader code start address was not correct for the ATxmega chips
- fixed: the compiler now generates correct instructions for reduced core chips (ATtiny10)
- changed: the compiler now produces a warning, that can be deactivated, for unknown #pragma directives, instead of an error like in previous versions
- added in Project|Configure|C Compiler|Code Generation the option *Enable auto Var. Watch in AVR Studio* in order to allow watching local automatic variables for reduced core chips (ATtiny10)
- added: locally declared enumerations can be allocated to registers, if possible
- fixed: the CodeWizardAVR didn't allow to enable the external interrupt INT1 for the ATtiny261, ATtiny461 and ATtiny861 chips
- added in the CodeWizardAVR the possibility to initialize the PORTA for the ATtiny48 and ATtiny88 chips in TQFP32 package

- increased the LPT port ISP FLASH programming speed under Windows 2000, XP and Vista
- added the Help topic: *CodeVisionAVR Integrated Development Environment|Using the Integrated Development Environment Workspace* in order to reflect the changes made in the new IDE
- updated the Help topics: *RAM Memory Organization and Register Allocation* and *Limitations* in order to provide additional information regarding the AVR8L core chips
- added the Help topic: *Transferring/Deactivating the License*
- moved the #pragma compiler commands descriptions from the *C Preprocessor* Help topic, to the new topic: *Compiler Directives*
- updated the MULTFILE example in order to better show the techniques used for projects consisting of multiple program modules

V2.03.9 Commercial Release

- added support for the ATA6285, ATA6286 and ATA6289 chips in the Compiler and Programmer
- added support for the ATtiny10 chip in the Compiler, CodeWizardAVR and Programmer
- improved the code optimizer
- improved error and warning checking
- added minor improvements to the IDE
- added the missing definition for the UCSR1D register in the 90usb82.h and 90usb162.h header files
- modified the linker so that global variables, placed at absolute addresses using the @ operator, will not overlap other global variables
- disabled *#pragma glbdef+* when compiling libraries
- fixed: when the SMALL memory model was used for bootloaders, the startup code incorrectly initialized the RAMPZ register of the ATxmega chips
- fixed: the *#pragma keep+* didn't have any effect if the variable, constant or function was previously declared as extern
- fixed: functions that return structures or unions could return incorrect results, if the returned structure/union was the last declared local auto variable
- fixed a bug which occured in the *lcd_putchar* function from *lcd.lib*, *lcd4x40.lib*, *lcdstk.lib*, when the text line was full and passing to the next line was necessary
- fixed: the new copy protection prevented the program to run under Windows 98SE

V2.03.8 Commercial Release

- added support for the ATmega1284P chip
- made small improvements to the syntax analyzer, linker and IDE
- fixed: initializing during declaration of a local bool variable with the result of an expression could sometimes not be performed correctly
- fixed: for the ATtiny261/461/861 chips, the CodeWizardAVR generated code for Timer0 overflow ISR referenced the TCNT0 register instead of TCNT0H and TCNT0L
- fixed: for the ATtiny24/44/84 chips, when the ADC was enabled, the CodeWizardAVR generated incorrect initialization of the ADLAR bit in the ADCSRB register
- fixed: in certain extremely rare cases the COFF object file generator could produce an internal error

V2.03.7 Commercial Release

- improved the code optimizer
- improved error checking
- added support for the ATxmega16A4, ATxmega32A4, ATxmega64A4 and ATxmega128A4 chips in the C Compiler and Programmer
- fixed: the malloc function (stdlib.h) didn't return a NULL pointer if the heap size allocated in the project configuration was set to 0
- fixed: an incorrect error message occurred when programming the lock bits for the ATmega48 chip using the JTAGICE MkII

V2.03.6 Commercial Release

- small improvements in the load/store code optimizer
- added support for anonymous structure/union members
- added support for the ATxmega192A1 and ATxmega256A1 chips
- fixed: in certain cases warnings appeared when initializing a global enumeration with it's own members

- fixed: no error was produced when the sizeof operator was applied to a structure or union that was never declared
- fixed: in certain cases bit variables were not listed in the .map file
- fixed: in certain cases pointer memory storage address didn't appear in the Code Information window
- fixed: the function ds1302read from ds1302.lib always returned 0
- fixed: after STK600 support has been added in V2.03.5, reading the fuse bytes for some chips produced errors with the AVRISP MkII and STK600 programmers

V2.03.5 Commercial Release

- fixed: in some very rare cases when two interrupt service routines shared the same code sequences at the end, the first interrupt service routine may not save all affected registers at startup
- fixed: COFF debugging information wasn't created if a global variable was not defined in a .C program module listed in Project|Configure|Files
- fixed: the line *#include <i2c.h>* was missing from the ds1621.lib file
- updated the User Manual and the Help code examples to use *#include <stdlib.h>* instead of *#include <math.h>* for the *abs* function definition

V2.03.4 Commercial Release

- improved compiler error and warning checking
- more detailed linker error reporting
- improved the code optimizer for logic expression evaluation
- increased the compilation speed
- added ATxmega128A1 and ATxmega64A1 chips support
  - support for the ATxmega chips is provided only for the C Compiler and STK600 Programmer in the JTAG mode. When creating a new project for these chips, do not use the CodeWizard and select the correct chip type in Project|Configure|C Compiler|Code Generation|Chip.
  - the Standard C I/O Functions use by default the USARTC0. This can be changed by defining the _ATXMEGA_USART_ macro as described in the corresponding Help topic.
  - the SPI functions use by default the SPIC controller on PORTC. This can be changed by defining the _ATXMEGA_SPI_ and _ATXMEGA_SPI_PORT_ preprocessor macros as described in the corresponding Help topic.
  - the I2C, 1 Wire, LCD, RTC functions do not yet support the ATxmega chips
- the @ operator now issues only a warning, not an error, for the attempt of placing a global variable outside the existing physical RAM area
- fixed: if a variable located in EEPROM was declared as extern in a program module and was later initialized during definition in another program module, the initialization wasn't performed, only memory space being allocated for it
- fixed: initialization during declaration of a local bit or float variable with the non-constant result of an expression wasn't performed
- fixed: global variables located in registers sometimes appeared listed several times in the .map file
- fixed some formating problems with the preprocessor output .i files
- fixed a typing error in ds1302.lib which lead to an assembly error
- modified the ftoa and ftoe functions from stdlib.h so that they will output the -NAN and NAN values
- modified the printf and sprintf functions from stdio.h so that they will output the -NAN and NAN values
- fixed: for chips that had the Crystal Oscillator Divider list box set to a value different than 1, the CodeWizardAVR didn't divide the clock frequency after loading a previously saved CodeWizardAVR project (.cwp) file
- improved the syntax highlighting for preprocessor directives
- multiple line comments can now be folded
- the opened header files are no more closed before Project|Compile, Project|Build or Project|Build All
- improved the items display speed for the Code Information window if alphabetical sort was enabled
- fixed: the bookmarks and folded code state was not preserved for the Header files after Project|Compile, Project|Build or Project|Build All
- added support for the Atmel STK600 programmer

- added in Settings|Terminal the option to enable or disable reseting, at Terminal startup, of the development board connected to the ISP

V2.03.3 Commercial Release

- fixed: in some cases when a syntax error was signaled in a header file, that was the last one in the Headers list displayed in the Code Navigator, this file remained locked and changes to it couldn't be saved in the Editor
- improved error reporting when initializing multidimensional arrays of char with literal strings
- fixed: a "possible loss of precision" warning was generated when assigning a PORT.bit to a bit variable
- fixed: a "possible loss of precision" warning was generated when a function returned a bool and the result of the return expression was obtained by using the ! operator applied to a char, int or long
- enabled the #pragma vector preprocessor directive
- fixed: 0b binary numbers were not syntax highlighted in the Editor
- fixed: in certain cases the Find in Files function took a very long time to be executed
- the text and background colors of the folded code block hint window now match the colors of the collapse mark hint

V2.03.2 Commercial Release

- new ANSI C compatible C front-end
- the bool data type was added
- the @ operator can now be used with variables located in EEPROM too
- there is no need to specify the 'flash' or 'eeprom' memory attribute during structure or union type definition. Therefore the same structure or union data type can be easily located in any memory area during variable declaration.
- the compiler now makes distinction between the 'const' type qualifier and the 'flash' memory attribute. In order to maintain compatibility with V1.25.x projects, the Project|Configure|C Compiler|Code Generation|Store Global Constants in FLASH Memory must be checked. If this option is not checked, identifiers declared with the 'const' type qualifier will be placed in RAM.
- the Code Information tab, present in the Navigator after the first Build, displays detailed information regarding defined data types, preprocessor macros, function and variables declarations, included header files, memory and register allocation
- the abs, cabs, labs and fabs functions were moved from the math.h header to the stdlib.h header
- the preprocessor does not use the old 'funcused' directive, when it is found it is always evaluated to 1
- added the #message preprocessor directive
- improved error and warning checking
- improved linker: only the modified C source files are compiled when the Project|Build command is executed
- improved COFF object file generator
- improved code optimizer
- completely redesigned text Editor with improved syntax highlighting for both C and AVR assembly
- added function parameters and global structure/union members auto complete
- improved code folding
- code folding state can be saved and restored
- bookmarks state can be saved and restored
- added automatic matching brace highlighting when the user places the cursor before the brace
- added automatic file saving at specified time interval
- the Code Navigator was redesigned and displays project information in a more logical way
- an additional Code Information tab is present after a project build. It displays detailed information about the included header files, preprocessor macro definitions, type definitions, global/static variable declarations, memory allocation and function definitions.
- added the Functions Call Tree tab in the Navigator
- the new Project|Configure|Files|Output Directories allows to specify in which directories the files generated by the compiler/linker will be placed.