V3.52 Commercial Release

- **Compiler**
  - Added support for the AVR64EA28, AVR64EA32, AVR64EA48 chips.
  - Improved the display speed for SSD1303, SSD1305, SSD1306, SSD1309, ST7565, ST7567, UC1608, UC1701, SH1106 controllers with 4 wire serial connection.
  - Added the #pragma message compiler directive.
  - Fixed compilation errors in boot.lib when using the Xmega E chips.
  - Fixed compilation errors when enabling the use of the CD and WP SD card socket switches for the AVR8X devices.
  - Fixed: #pragma fuses must expect 9, not 10, values for AVR8X chips.
- **CodeWizardAVR**
  - Added support for the AVR64EA28, AVR64EA32, AVR64EA48 chips.
  - Added the missing Timer/Counter TCB4 for the AVR128DA64 chip.
  - Added for AVR8X chips: check for analog comparator's outputs assigned to the same I/O port pin.
  - For AVR8X chips: limited the automatic scan of ADC inputs only to the external ones, because the internal ones don't have contigous values for MUXPOS.
  - Corrected for Xmega chips: the Xmega D3, D4 chips have 4 Event Channels, not 8.
  - Corrected the displayed XDIR USART signal remapping for AVR DA/DB/DD chips when RS485 mode was enabled.
  - Corrected the check for XDIR USART signal conflicts for AVR DA/DB/DD, AVR Tiny 2 chips when RS485 mode was enabled.
  - Corrected for AVR DD chips: USART0 pin remapping missed the option to use PORTA pins 2 & 3.
  - Corrected the Event Destinations for the AVR DD chips.
  - Corrected the code generation by I/O ports' event outputs for the AVR8X chips.
  - Corrected for the AVR DA, DB, DD chips: the initialization of the Voltage Level Monitor interrupt register BOD.INTCTRL was not correct.
- **Chip Programmer**
  - Added the CFD fuse for the ATmega324PB chip.
  - Added the possibility to specify the JTAG clock frequency.
  - Fixed: Loading/saving .BIN files was always performed from address 0 instead of the specified Start Address.
- **IDE**
  - There is no need to restart the CodeVisionAVR IDE anymore to apply the modified 'Show Edited File Path' and 'Show Functions List' options in the Settings|IDE menu.
  - Fixed: The Terminal was locked when the user scrolled the vertical scroll bar to the bottom during reception.

V3.51 Commercial Release

- **Compiler**
  - Added support for the AVR16DD14, AVR16DD20, AVR16DD28, AVR16DD32, AVR32DD14, AVR32DD20, AVR32DD28 and AVR32DD32 chips.
  - Improved warnings reporting for the linker.
- **CodeWizardAVR**
  - Added support for the AVR16DD14, AVR16DD20, AVR16DD28, AVR16DD32, AVR32DD14, AVR32DD20, AVR32DD28 and AVR32DD32 chips.

V3.50 Commercial Release

- **Compiler**
  - Added support for the AVR64DD14, AVR64DD20, AVR64DD28 and AVR64DD32 chips.
  - Added an additional check in *#pragma fuses* to prevent accidental UPDI disabling by incorrect programming of the *SYSCFG0.RSTPINCFG* fuse bits for AVR8X Tiny chips.
  - Improved the code size for the DS1307, DS1621, DS3231, PCF8563, PCF8583 libraries with TWI connection. Also for the DS1321 with bit-banged I2C.
  - Modified the FT5X06 touch screen controller library to be compatible with the AVR8X chips.
  - Fixed in the *Project|Configure|After Build* menu: The *Reset Pin* configuration setting was not saved for the AVR DA/DB chips.
- **CodeWizardAVR**
  - For ATtiny87/167 chips: Added the possibility to disable the digital input buffers for ADC inputs 8..11.
  - Updated the minimal/maximal ADC frequencies for AVR8X chips.
  - Fixed for AVR8X, DA, DB chips: Initializing the *CPUINT.LVL0PRI* and *CPUINT.LVL1VEC* registers was not correct for non-default interrupt priority.
  - Fixed for AVR8X chips: the *SET_SPIn_SS_LOW* and *SET_SPIn_SS_HIGH* macros defined in the generated *spi_init.h* header file were incorrect.
  - Fixed for AVR8X chips: In SPI initialization, writing to the *SPI.CTRLB* register reset the *SPI_ENABLE* bit in the *SPI.CTRLA* register.
  - Fixed for AVR DA/DB chips: The USART RS485 operating mode has only two settings: on or off. Also updated the AVR DA/DB header files to reflect that.
  - Fixed for Xmega and AVR8X chips: missing *#include <ft5x06.h>* in the generated *ports_init.c* file when the capacitive touch screen controller was used.
- **IDE**

- Improved the detection speed of an USB COM port used for connecting to an Arduino board.
- Improved compatibility with Windows 11 installed on a computer without serial COM ports.

V3.49 Commercial Release

- **Compiler**
  - Added support for the ST7735 TFT display controller.
  - Added support for TWI1 and SPI1 peripherals for the ATmega324PB chip.
  - Added support for the AVR32DB28, AVR32DB32 and AVR32DB48 chips.
  - Fixed missing '}' in the header file mega3209.h at line 2024.
- **CodeWizardAVR**
  - Fixed for AVR DB devices: changed OPAMP_OP0CTRLA_OUTMODE_OFF to OPAMP_OP0CTRLA_OUTMODE_OFF_gc for OPAMP Output Mode selection.
  - Fixed for AVR8X Tiny1 devices: when initializing the TWI0 pin remapping to the default I/O port pins changed

    PORTMUX.CTRLB&=PORTMUX_TWI0_DEFAULT_gc;

    to

    PORTMUX.CTRLB&= ~PORTMUX_TWI0_ALTERNATE_gc;
  - Fixed for AVR8X Tiny1 devices: when initializing the SPI0 pin remapping to the default I/O port pins changed

    PORTMUX.CTRLB&=PORTMUX_SPI0_DEFAULT_gc;

    to

    PORTMUX.CTRLB&= ~PORTMUX_SPI0_ALTERNATE_gc;
  - Fixed for AVR8X Tiny1 devices: when initializing the TWI0 pin remapping to the default I/O port pins changed

    PORTMUX.CTRLB&=PORTMUX_TWI0_DEFAULT_gc;

    to

    PORTMUX.CTRLB&= ~PORTMUX_TWI0_ALTERNATE_gc;
  - Fixed for AVR8X DB devices: "Cannot focus a disabled or invisible window." error message for conflicts between an OPAMP input or output and an I/O port.
  - Fixed for AVR8X DB devices: allow OPAMP inputs and output to be connected to ADC inputs.
  - Fixed for AVR8X devices: the EXTS status bit in the MCLKSTATUS register must be tested only if an external crystal is used as clock source.
  - Fixed for AVR8X devices: "List index out of bounds" error when generating ADC code without having previously displayed the ADC settings.

- Fixed for AVR8X devices: in order to comply with the device header files for the AVR DA/DB chips, when generating code for the ADC window mode interrupt, use the ADCx_WCMP_vect interrupt vector name, instead of ADCx_WCOMP_vect like for the rest of AVR8X chips.
- Fixed for ATtiny1626/1627 devices: wrong EVSYS register name for PORTC was used in the generated code. The correct one is EVSYS.USEREVSYSEVOUTC.
- Fixed for AVR8X devices: the Analog Comparator's output can be now inverted, even if its output is not directed to an I/O pin, but used internally.
- Fixed for AVR8X Tiny1, DA, DB devices: changed in the generated code

  PORTMUX_EVOUTn_bm // where n is the I/O port #: 0, 1, ...

  to

  PORTMUX_EVOUTp_bm // where p is the I/O port name: A, B, ...

- **Chip Programmer**
  - Added support for uploading code to the LGT8F328P MinEVB Arduino compatible boards.
  - Improved error reporting for Arduino uploading.
- **IDE**
  - Improved handling the disconnection of a COM port in the Terminal.

V3.48 Commercial Release

- **Compiler**
  - Added universal CodeVision bootloader library (boot.h) with code examples in the *Examples\Bootloaders* directory.
  - Added webserver with ENC28J60 and SD Card examples for the AVR128DA48 Curiosity Nano development board
  - Added support for the ATtiny424 chip
  - Added support for the SPI1 peripheral of AVR8X chips in the SD Card and ENC28J60 libraries
  - Added the possibility to place code in the boot section for XMega applications
  - Added the option in the Project|Configure|C Compiler|Advanced menu to disable hardware and data stack pointer initialization if an external startup.asm file is used. Added the _STACKS_INIT_DISABLED_ predefined macro for this case
  - Ensure that the startup.asm and vectors.asm files will be included from the directory where the .prj file is located
  - Fixed incorrect code generation for the __INIT64 and __EQU64 macros from math64.h when the constant argument was a negative integer in the range -1 to -(2^32-1)
  - Fixed the sign64 function from math64.h producing wrong results

- Fixed: The function lm75_temperature_10 from the TWI version of the LM75 library (lm75_twi.h) returned incorrect values for negative temperatures. The same function from the bit-banged I2C LM75 library (lm75.h) is OK
- Fixed: The fuse bytes values saved in the ELF file for the AVR DA/DB chips were not correct. Fuse programming after a successful build operation was however correct.

- **CodeWizardAVR**
  - Added for AVR8X chips the option to send/receive USART 9-bit data with low or high bit first
  - Fixed for the ATtiny426/427/826/827/1626/1627 chips: when ADC0 autoscan mode was used and AIN9 was selected as start or end input, then the generated code mentioned AIN17 instead of AIN9
  - Fixed for AVR8 chips with 4 USARTS: when generating code for USART3, the Communication Parameters for USART2 were used, if those were set for 9 bits/character
  - Fixed: for the AVR Tiny 2 (Atiny424/824/826/827/1624/1626/1627/324/326/327) chips the Timer/Counter A's Event Control Register has two inputs: Event Input A and Event Input B
  - Fixed: for AVR32 DA/DB chips PORTF pin 6 is input only
  - Fixed: for AVR8X chips a wrong/negative value was displayed for I/O port pins when trying to use a graphic display controller with TWI connection (ex. SSD1306)
  - Fixed: for AVR DA/DB chips, wrong TWI signals allocation in slave mode.
- **Chip Programmer**
  - Added support for the JTAG2UPDI programmer
  - Added support for the Arduino Nano Every board
  - Added support for uploading to Arduino boards with USB connection.
- **IDE**
  - Added an Uploader for the CodeVision bootloader library in the Tools menu
  - Added the possibility to automatically upload the code to the bootloader after a successful build operation
  - Added: When using Arduino boards connected to the PC, automatically try to select a COM port assigned to the board for program uploading
  - Fixed in Arduino Upload window: if a project is opened and there's no EEPROM hex file to program, don't show the non-existent file name in the 'Files > EEPROM' edit box
  - Fixed: when automatic programming of an AVR8X chip after build was enabled, the lock byte programming was forced, even if it was not activated.

V3.47 Commercial Release

- **Compiler**
  - Improved the code optimizer

- Added JPEG image display functions and code examples for them
- Added the missing *ASYNCUSER0* structure member in the tiny212.h, tiny214.h, tiny412.h and tiny414.h header files
- Added the missing *EVSYS_ASYNCUSER0_enum* enumeration in the tiny212.h, tiny214.h, tiny412.h and tiny414.h header files
- Modified the ILI9481, ILI9486 and ILI9488 TFT display libraries to be compatible with Arduino shields with 16-bit connection and missing /RD signal.

- **CodeWizardAVR**
  - Added code to reset the *WDRF* flag when configuring the watchdog for the AVR8L, ATtiny1634/441/828/841 chips
  - Fixed: Incorrect code was generated for the TCA timer of AVR DA/DB chips if the counter clock source was an event
  - Fixed: The /INT signal of the capacitive touch screen controller was always allocated to PORTA bit 0 for the following chips: AT90PWM2/216/3/316, AT90USB162/82, ATmega16/168/169/32/323/328/48/8535/88, ATmega32C1/64C1, ATmega16M1/32M1, ATmega16U2/32U2/8U2
  - Fixed: The Watchdog configuration settings were missing for the ATtiny441/841 chips
  - Fixed: For AVR8X chips the ADC sample length comment in the generated code displayed a value lower by 1 ADC clock cycle. The generated code was however correct.
  - For classic AVR8 chips corrected the text in the *Out. C* listbox from "Non-Inverted" to "Non-Inverted PWM" for Timer 3 output C in PWM modes
  - Fixed the generated code for the ATtiny202/212/214/402/412/414/814 AVR8X chips when configuring the WO0..5 outputs of TCA0 operating in split 8-bit PWM mode
  - Fixed: *ADC_SAMPNUM_NONE_gc*, instead of *ADC_SAMPNUM_ACC1_gc*, was used to initialize the *ADC.CTRLB* register for the AVR8X Tiny1 and Mega AVR0 chips, when "No Sample Accumulation" was specified for the 'Sample Accumulation' setting. The bug appeared in CodeVisionAVR V3.45 when support for AVR8X Tiny2 chips was added.
  - Fixed: For AVR DA/DB chips: *TCB_CLKSEL_CLKTCA_gc*, instead of *TCB_CLKSEL_TCA0_gc*, was used to initialize the TCB.CTRLA register when "Use the same clock source as TCA0" was specified for the *Clock Select* setting of TCB Timer/Counters.
- **Chip Programmer**
  - Fixed: When reading the AVR8X Fuse Bytes in the *Tools|Chip Programmer* menu and copying the read values to the *Fuse Settings*, the *Lock Device* check box was always set.
  - Removed the non-existing fuse byte *TCD0CFG* for the ATtiny202/402 chips.
- **LCD Vision**
  - Added the possibility to export JPEG compressed files as C source code, to be used by the JPEG display functions.

V3.46 Commercial Release

- **Compiler**
  - Added support for the ILI9481, ILI9486 and ILI9488 graphic display controllers
  - Added support for the SPI interface of the MegaAVR0, AVR DA/DB chips for the ILI9163 display controller.
  - Made the overflow warning messages when shifting left 8-bit and 16-bit operands more clearer
  - Changed the #pragma-s from *warn_8bit_shift_left_ovf* and *warn_16bit_shift_left_ovf* to *warn_8bit_op_shift_left_ovf* and *warn_16bit_op_shift_left_ovf*
  - Fixed assembly error 'Undefined symbol: __EEPROMRDB' for the following code:

```
eeprom int abc;
...
if (abc<0) ...
```

or

```
eeprom long abc;
...
if (abc<0) ...
```

  - Fixed PORTMUX.TWISPI error message when using SPI for Mega AVR0 chips (ATmega4808 and similar) in: spi.lib, enc28j60.lib, glcd_ili9340.lib, glcd_ili9341.lib, glcd_ra8875.lib, glcd_uc1610.lib, glcd_st7565.lib, glcd_st7789.lib, sdcard.lib
- **CodeWizardAVR**
  - Added code to clear the interrupt flags in the I/O port ISR generated for AVR8X chips
  - Fixed: for AVR8X chips the comment in the main file generated by the wizard specified an AVR core clock frequency without the main clock prescaler divider setting being applied. The same non-divided value was saved as AVR clock in the configuration of the generated project
  - Fixed: for AVR8X chips the SPI SCK setting in the wizard was not saved correctly.
- **Chip Programmer**
  - Added the CFD fuse for the ATmega328PB chip.

V3.45 Commercial Release

- **Compiler**
  - Added support for the new ATtiny3224, ATtiny3226, ATtiny3227, ATtiny824, ATtiny826 and ATtiny827 chips

- Improved the code optimizer.

- **CodeWizardAVR**
  - Added support for the new ATtiny3224, ATtiny3226, ATtiny3227, ATtiny824, ATtiny826 and ATtiny827 chips
  - Added temperature measurement code generation for the ATtiny1624, ATtiny1626, ATtiny1627, ATtiny3224, ATtiny3226, ATtiny3227, ATtiny824, ATtiny826 and ATtiny827 chips.

- **IDE**
  - Improved caret positioning in the Editor window when clicking with the mouse on a character.

V3.44 Commercial Release

- **Compiler**
  - Added support for the new AVR128DB chips
  - Added support for the new ATtiny1624, ATtiny1626, ATtiny1627 chips
  - Improved speed for 32-bit long/unsigned long calculations
  - Improved compilation speed
  - Fixed AVR8X EEPROM programming error within Atmel Studio 7 and modified EEPROM access functions for these chips
  - Fixed: EEPROM write not working for AVR DA chips
  - Fixed: the ATmega808/809 chips have 2 word interrupt vector spacing, but do not support the CALL instruction
  - Fixed in twi.lib: TWI1 for the ATmega328PB chip always sending slave address 0
  - Fixed: 'setjmp' function from setjmp.h not working correctly when having a pointer to the same 'env' variable as argument and optimization for SIZE set in the Project|Configure C Compiler|Code Generation menu

- **CodeWizardAVR**
  - Added support for the new AVR128DB chips
  - Added support for the new ATtiny1624, ATtiny1626, ATtiny1627, ATtiny3224, ATtiny3226, ATtiny3227 chips
  - Added a check for AVR8X chips: due to a bug in silicon, if the USART TxD pin is configured as open drain, then it must be set as an INPUT
  - Fixed: moved the alternate USART TXD/RXD signals for ATtiny202/402 from PORTB pins 1/2 to PORTA pins S
  - Fixed: Library error: stdio.lib(247): undefined symbol '_AVR8X_USART_' when the CodeWizard for AVR8X devices generated code only for USART receiver enabled
  - Fixed: some chips AVR DA chips don't have TCA1, but TCB clock source allowed to select the same clock source as TCA1

- Fixed: the datasheet for the ATtiny202/402/212/412 chips specifies wrong pins for the TWI SDA and SCL signals. The correct pins are: SCL - PORTA pin 1, SDA - PORTA pin 2
- Fixed: false warning about peripheral clock change for AVR8X chips
- Fixed: The peripheral clock divisor wasn't taken in account when calculating the ADC clock frequency for AVR8X chips

- **Chip Programmer**
  - The AVR DA, AVR DB, ATtiny1624/1626/1627/3224/3226/3227 chips can now be programmed using Atmel ICE, nEDBG, mEDBG programmers, even if Atmel Studio 7 is not installed
  - Fixed: Automatic chip programming after a successful build didn't work with the ERFOS PROG-S2 AVR programmer from www.diamex.de (not fully STK500 compatible).

V3.43 Commercial Release

- **Compiler**
  - Added support for the new AVR DA chips
  - Added #pragma nested_int-/+ to force disabling/enabling nested interrupts during interrupt processing in ISRs for Xmega, AVR8X, AVR DA chips
- **CodeWizardAVR**
  - Added support for the new AVR DA chips
  - Added a check for the AVR8X chips that the TCA timer/counter waveform output selection use the same I/O port
  - Added a check for the AVR8X chips that all the Analog Comparators use the same voltage reference
  - Added a check for the AVR8X chips that all the ADCs use the same voltage reference
  - Added a check for the AVR8X chips that all the DACs use the same voltage reference
  - Enabled generating code for graphic displays connected using the bit-banged I2C interface to the AVR8X chips
  - Fixed for AVR8X chips: graphic displays can use only SPI0
  - Fixed for AVR8X chips: incorrect TWI peripheral specified when using graphic displays
  - Fixed for AVR8X chips: TWI signal remaping setting wasn't applied
- **IDE**
  - Added the option to select a monochrome or color printer in the Page Setup window
  - Fixed: For AVR8X chips the Project|Configure|C Compiler|Libraries|Graphic Display menu allowed to use only TWI0 for graphic displays connected using the I^2C interface
- **Chip Programmer**
  - Added support for the nEDBG programmer

- Increased the allowable ISP SCK frequency to 4 MHz for the AVR Dragon, JTAGICE MkII and JTAGICE3 programmers.

V3.42 Commercial Release

- **Compiler**
  - Added support for the ATmega1608/1609/808/809 chips
  - Improved the code optimizer
  - Added gradient colored drawn line function *glcd_linegrad* in graphics.h
  - Added gradient filled rectangle function *glcd_bargrad* in graphics.h
  - Added gradient colored horizontal bar drawing function *glcd_bargradh* in graphics.h
  - Added gradient colored vertical bar drawing function *glcd_bargradv* in graphics.h
  - Added gradient colored circle arc drawing function *glcd_arcgrad* in graphics.h
  - Added the *C:\cvavr\Examples\Graphic Displays\Gradient ILI9163* example for gradient coloring functions
  - Updated in the mega3208.h, mega3209.h, mega4808.h and mega4809.h header files the names of the PORTMUX registers from: USARTA to USARTROUTEA; from TWISPI to TWISPIROUTEA; from TCA to TCAROUTEA; from TCB to TCBROUTEA in order to match the latest datasheets for these devices
  - Updated in the mega3208.h, mega3209.h, mega4808.h and mega4809.h header files the name of the TWI.BRIDGECTRL register to TWI.DUALCTRL in order to match the latest datasheets for these devices
  - Fixed: power management functions (sleep.h) not working for the ATtiny441/841 chips
  - Fixed: disabling the watchdog timer in the start-up code not working for the ATtiny441/841 chips
- **CodeWizardAVR**
  - Added for the AVR8X ATmega3208/3209/4808/4809 chips an additional reference value (AVDD) selection for the Analog Comparator's AC0 negative input
  - Added for the AVR8X ATmega1608/1609/808/8093208/3209/4808/4809 chips the possibility to correct the frequency of the RTC crystal clock
  - Modified for the ATmega3208/3209/4808/4809 chips the BOD level settings to 1.8V, 2.6V and 4.3V to match the latest datasheets for these devices
  - Fixed: for AT90CAN32/64/128 chips the External Interrupts 0, 1, 2 and 3 must also support Any Change as triggering mode.
- **IDE**
  - Speed-up displaying large number of errors/warnings in the Code Navigator
  - Added: A recently used project can be removed from the Start Page by right clicking on the corresponding link

- Modified: If a project is opened and *After Build|Action|Program Chip*, then set the project's default programmer settings to be the same as for *Tools|Chip Programmer*.

V3.41 Commercial Release

- **Compiler**
  - Further reduced Build time for large projects
  - Enabled bit-banged I^2C support for AVR8X chips
  - Fixed: the ELF production file didn't contain the lock bits settings for the AVR8X chips
  - Fixed: warning "local variable is used before its value is set" not working in *do ... while*.
- **Atmel Studio Extension**
  - Added information message about how to configure Visual AssistX when creating a new project.
- **IDE**
  - Fixed a bug that appeared in V3.40: any header files opened in the editor were closed after *Project|Configure* was executed
  - Fixed a bug that appeared in V3.40: sometimes the chip type specified for the currently opened project was not selected by default in the *Tools|Chip Programmer*.

V3.40 Commercial Release

- **Compiler**
  - Added support for the internal touch screen controller of the RA8875 chip
  - Added support for bit-banged I^2C for the XMega chips
  - Added a timeout in the *twi_master_trans* function of the TWI libraries for XMega and AVR8X chips
  - Reduced Build time for large projects
  - Increased the size of an internal buffer in order to remove the "Macro name table full" error message for projects with a very large number of #define-d preprocessor macros
  - Fixed: incorrect code was generated in inline functions for switch statements, when the switch parameter was an *(unsigned) int* or *(unsigned) char* with *ANSI char to int promotion* enabled
  - Fixed the *SP_LoadFlashWord* function in the *sp_driver.c* and *sp_driver.h* files for the AVR1316 and AVR1605 Atmel application note examples for Xmega chips. The bug prevented correct FLASH self-programming for chips with FLASH size larger than 256 kbytes.
- **CodeWizardAVR**
  - Fixed the error checking for output compare for Timers 0,1,2 for ATtiny441/841/828 chips
  - Fixed: when generating code for bootloaders, for AVR8X chips the BOOTEND fuse was set to a boot section size higher by 128 words than the one actually specified

- Fixed: the ATtiny1604/1606/1607/202/402/406/804/806/807 chips actually don't allow the pin reallocation of the TWI0 SCL and SDA signals using the PORTMUX register
- Fixed: the ATtiny202/204/214/402/404/414/814 chips actually don't have a CKOUT pin.

- **Chip Programmer**
  - Added a more intuitive fuse bit configuration in the *Tools|Chip Programmer* and *Project|Configure|After Build|Action:Program* Chip menus for the AVR8 chips
  - Added a warning when the Fuse and Lock Bits settings must be checked when a chip type has changed in the chip programmer or project configuration After Build programming
  - Modified the chip programmer settings to be specified in the *Project|Configure|After Build|Action: Program* Chip menu individually for each project
  - Moved the *Settings|Chip Programmer* menu to the *Tools|Chip Programmer|Settings*
  - Improved functionality when selecting boards in the Arduino upload menu.
- **IDE**
  - Improved compatibility with Windows 10
  - Improved compatibility with some poorly designed antivirus programs that produced false alarms because of the anti-hacking protection used in CodeVisionAVR
  - Significantly increased the speed of opening large projects
  - Added modified/saved line state display for editor's gutter
  - Added a comparison position marker in the *Compare Files* window
  - Added a 'Go to Line' item in the *Compare File* right click pop-up menu
  - Fixed errors in the Code Navigator when trying to save an opened .asm, .lst or .map file under a new name using *File|Save As*.

V3.39 Commercial Release

- **Compiler**
  - Added support for the ILI9163 TFT display controller and corresponding example programs
  - Added support for the SPI interface of the AVR8X chips in the libraries for the ILI9340, ILI9341, RA8875, ST7565, ST7567, ST7789 and UC1610 display controllers
  - Improved the code generator
  - Fixed in the *Project|Configure|C Compiler|Advanced* menu: the watchdog couldn't be disabled at start-up for the AVR8X chips
  - Fixed a bug that appeared in V3.38: Missing predefined macros for the ENC28J60 library: *_ENC28J60_SS_BIT_*, *_ENC28J60_SCK_BIT_*, *_ENC28J60_MOSI_BIT_* and *_ENC28J60_MISO_BIT_* when compiling for the ATxmega devices
  - Corrected the values of *EEPROM_PAGE_SIZE*, *MAPPED_PROGMEM_PAGE_SIZE*, *PROGMEM_PAGE_SIZE* macros in the *tiny3214.h*, *tiny3216.h* and *tiny3217.h* header files.

- **CodeWizardAVR**
  - Added code generation for initializing the BOD and VLM for the AVR8X chips.
- **Chip Programmer**
  - Added an option checkbox to enable/disable the memory programming verification during Arduino upload
  - Increased the maximal SCK clock frequency to 2.5 MHz when programming with AT Atmel-ICE in ISP mode.
- **IDE**
  - Improved error handling when opening projects
  - Added automatic notification when new CodeVisionAVR versions are released.

V3.38 Commercial Release

- **Compiler**
  - Added support for the AVR8X chips in the SPI, ENC28J60 and SD Card libraries
  - Added support for the ATmega328PB TWI1 controller in the BH1750, BME280, BMP085, BMP180, BMP280, DS1307, DS1621, DS3231, LM75, MS5611, PCF8563, PCF8583, PCF8574 and *alcd_twi* libraries
  - Added SD Card example for the ATmega4809 Xplained Pro board
  - Added ENC28J60 web server example for the ATmega4809 Xplained Pro board
  - Corrected the addresses of the *USICR, USISR, USIDR, USIBR* registers in the *tiny1634.h* header file
  - Corrected the bit definitions for the *DIDR0* and *SPMCSR* registers in the *mega328pb_bits.h* header file used for the ATmega328PB chip
  - Corrected the address of the *GPIOR1* register for AVR8X chips from 0x1B to 0x1D. The bug affected bit variables that were allocated to this register when bit variables size was between 9 and 15.
  - Fixed: the RAM mapped FLASH starts from 0x4000, not 0x8000, for the ATmega3208, ATmega3209, ATmega4808 and ATmega4809 chips. This resulted in reading wrong data from structures and arrays located in FLASH memory, but only when the array index was a constant.
- **CodeWizardAVR**
  - Added code generation for *Timer/Counter type D* in the CodeWizard AVR for the AVR8X ATtiny devices
  - Fixed: missing the possibility to use *PORTC* pin 6 as *RTCclk* output for the ATxmega128A1U/64A1U chips
  - Fixed: for the ATmega1284/1284P/324PB/328PB chips Timer3 Output B settings didn't set correctly the *COM3B0* and *COM3B1* bits in the *TCCR3A* register.
- **Chip Programmer**

- Added AVR8X support for programming with Atmel ICE, even when Atmel Studio 7 is not installed
- Fixed error when reading/writing the fuse bits of the Xmega A4, A4U, C3, C4, E5 chips when Atmel Studio was not installed.

V3.37 Commercial Release

- **Compiler**
  - Improved the code optimizer
  - Added library for the SH1106 graphic OLED display
  - Added library support for the SPI1 and TWI1 peripherals of the ATmega328PB chip
- **CodeWizardAVR**
  - Added code generation for the RTC and PIT in the CodeWizard for the AVR8X chips
  - Added code generation for the SPI1 and TWI1 in the CodeWizard for the ATmega328PB chip
  - Fixed: For the AVR8X chips the I/O port interrupt vectors must be named in the generated code: *PORTn_PORT_vect* instead of *PORTn_vect*
  - Fixed: The wizard didn't generate code to enable the DAC output 1 for the Xmega A3U chips
  - Fixed: There was no option to use the AVCC/2 ADC Voltage Reference selection for the Xmega D4 chips.
- **Chip Programmer**
  - Fixed: AVRDude not working correctly if the decimal separator was not set to '.' in Windows Control Panel number format in regional settings.

V3.36 Commercial Release

- **Compiler**
  - Modified the ENC28J60 Ethercard library so that the *ether_begin* function will correctly initialize the library's global variables every time it is called, even if no chip reset has occurred
  - Modified the ENC28J60 Ethercard library so that the *ether_clientIcmpRequest, ether_ntpRequest, ether_udpPrepare* and *ether_sendUdp* functions can be used even if there is no gateway present on the network (for example an ad-hoc connection between two computers)
  - Increased to 508 bytes the maximum size of the UDP packet that can be sent using the *ether_sendUdp* and *ether_makeUdpReply* functions from the ENC28J60 Ethercard library
  - Added an example for UDP communication for the ENC28J60 Ethercard library
  - Added support for ST7920 graphic displays with 160x32 resolution

- Fixed a bug that appeared in the SSD1306 library in V3.35: vertical display reversing didn't work

- **CodeWizardAVR**
  - Added code generation for the Analog Comparator and DAC for the AVR8X chips
  - Fixed a bug that appeared in the CodeWizard in V3.35: for some chips the check for conflicts between the bit-banged I^2C and other peripherals could result in a "Cannot focus a disabled or invisible window message"
  - Fixed: for AVR8X chips there was no code generated for the PORTMUX register to enable the EVOUT port outputs.

- **Chip Programmer**
  - Fixed: mEDBG and EDBG chip programmers not working for AVR8X chips in certain situations.

V3.35 Commercial Release

- **Compiler**
  - Added support for the ATmega3208/3209/4808 chips
  - Added the possibility for bit-banged I2C to specify separate I/O ports for the SDA and SCL signals
  - Added support for displays with 64x32 pixels resolution in the SSD1306 library
  - Improved the speed of the ILI9225 graphic library when using the SPI connection
  - Fixed in the DS3231 libraries: The *ds3231_get_temp* function returned wrong values for negative temperatures
  - Fixed in the SSD1963 library: Writing incorrect Green color bit 2 when using the 64k color mode, 8-bit bus interface and DB0..7 bits specified as not being in successive ascending order on the same I/O port

- **CodeWizardAVR**
  - Added Timer/Counter type A & B support for AVR8X chips
  - Fixed: Because of the limitation of the 8-pin package, the USART0 signals need always to be remapped for the ATtiny202/212/402/412 chips
  - Fixed: Because of the limitation of the 8-pin package, the TWI0 signals need always to be remapped for the ATtiny212/412 chips
  - Fixed: Wrong SPI0 /SS signal allocation for the ATtiny202/402 chips
  - Fixed the allocation of OC4D and /OC4D signals in for the ATxmega16U4/32U4 chips

- **Chip Programmer**
  - Added the possibility to program the ATxmega128A1U chip on the XMEGA A1U Xplained Pro Evaluation board using the built-in EDBG programmer
  - Added a warning that programming the DWEN fuse to 0 will disable further ISP programming

- Fixed: The Chip Programmer's *File > Save ELF File* menu didn't save the Fuse Bits settings for AVR8 and AVR8X chips; for Xmega chips the Fuse Bits were saved correctly.

V3.34 Commercial Release

- **Compiler**
  - Added a library for the ST1305 OLED display controller
  - Added support for the ATtiny202/204/402/404/406/412/804/806/807/1604/1606/1607 AVR8X chips
  - Modified the glcd_st7565.h header file to be compatible with the DOGL128-6 display from Electronic Assembly
  - Modified the lcd_init function in the SSD1803 LCD display library to clear the upper row of icons in the DIP203-6 display from Electronic Assembly
  - Modified the project configuration menu to allow sharing the same SPI peripheral between the graphic display and the SD card, using different /CS signals
  - Fixed the glcd_filltriangle function from graphics.h which produced a fill outside the triangle in certain cases
- **CodeWizardAVR**
  - Added support for the AVR8X ATtiny202/204/402/404/406/412/804/806/807/1604/1606/1607 AVR8X chips
  - Added code generation for the ADC of the AVR8X chips
  - Fixed: The CodeWizard didn't update the Timer0 OCR0B register with the value specified in the 'Compare B' edit box for the ATtiny4/5/9/10 chips
  - Fixed in the CodeWizardAVR for Xmega A3U chips: ADCB doesn't have the ADC8..15 inputs assigned to PORTA
  - Fixed the allocation of SPIC /SS, MOSI and SCK signals in for the ATxmega16C4/32C4 chips
- **IDE**
  - Fixed: Forced closing of the CodeVisionAVR IDE, while a large project was loading, could lead to the toolbars disappearing or access violations on subsequent IDE launches
  - Fixed: Incorrectly displayed fuse byte options for the Xmega chips when enabling automatic programming after build in the Project|Configure|After Build menu. The bug appeared in V3.33.

V3.33 Commercial Release

- **Compiler**
  - Added a library for the ST7793 TFT display controller
  - Added support for bit-banged I2C and TWI interface in the SSD1306 and SSD1309 graphic display libraries

- Added libraries and examples for the ENC28J60 Ethernet controller
- Added compiler support for the AVR8X ATmega3208/3209/4808/4809 chips
- Improved the code generated for the AVR8X chips
- Added the *rtc_set_week_day* and *rtc_get_week_day* functions in the DS1302 library
- Increased the maximum bit-rate for bit-banged I2C to 400kHz
- Fixed: @ operator address check for AVR8X EEPROM variables
- Fixed: I/O PORT F of ATmega64/128 not working for 1 Wire functions because PINF address is 0, not 0x60 as expected by the 1 Wire library

- **CodeWizardAVR**
  - Added a new Wizard for the AVR8X chips: ATtiny414/416/1614/1616/1617 and ATmega3208/3209/4808/4809
  - Improved the code generated by the CodeWizardAVR for Xmega chips when operating the USART in SPI mode
  - Added the 7N1, 7E2, 7O2, 8E2, 8O2, 9E2, 9O2 USART communication parameter settings in the CodeWizardAVR for classic AVR8 chips
  - Fixed: The ADCSRB register was incorrectly initialized by the CodeWizardAVR for the AT90CAN128/32/64 chips
  - Fixed: the CodeWizard for Xmega chips didn't #include "twi_init.h" when no TWI was used, but the option to generate code for disabled peripherals was enabled
  - Fixed: the CodeWizard for Xmega chips generated #include "dac_init.h" when the DAC wasn't used, but the option to generate code for disabled peripherals was enabled
- **Chip Programmer**
  - Added support for the Microchip EDBG programmer
- **IDE**
  - Improved compatibility with Visual Assist X from Atmel Studio
  - Fixed: If struct/union members autocomplete function was disabled in the *Settings > Editor > Autocomplete* menu, the text editor became unstable and *Code Information* didn't work correctly

V3.32a Commercial Release

- **Chip Programmer**
  - Fixed: If Atmel Studio was not installed, program uploading to Arduino boards produced errors.

V3.32 Commercial Release

- **Compiler**
  - Added compiler support for the ATtiny212/214/414/416/1614/1616/1617/3214/3216/3217 and ATA8510 chips

- Added libraries for the PCF8574(A) I^2C I/O expander in both bit-banged and TWI modes
- Added libraries and examples for alphanumeric LCDs connected to a PCF8574 I^2C I/O expander (Arduino LCD shields with I^2C connection)
- Updated the ILI9225 TFT display library to support the 4 wire SPI connection
- Improved the generated code for structure initialization during declaration
- Modified the EEPROM run-time library functions to properly initialize to zero the EEARH register for the ATtiny24/44/25/45/261/461 and ATmega48 chips. These chips have this register, altough they feature only 128 or 256 bytes of EEPROM.
- Fixed: After a member of a structure returned by a function was fetched, the interrupts remained globally disabled

- **CodeWizardAVR**
  - Fixed: The CodeWizardAVR didn't set the URSEL0 (URSEL1) bit to 1 when configuring the UCSR0C (UCSR1C) register for the ATmega162 chip's USART0 (USART1)
  - Fixed: The ATmega328PB DIDR0 register has 8 bits, not 5
  - Fixed: For obsolete chips with UART (AT90S4414/8515/8535, ATmega103/603), changed the generated code from: *#define DATA_OVERRUN (1<<OR)* to #define *DATA_OVERRUN (1<<DOR)*
  - Fixed: A 'List index out of bounds' error message was issued when selecting timer TC0 for the Xmega B1/B3 chips
  - Modified the wizard and new project creation template, so that it #includes the device specific header instead of the generic *io.h*, so that VAssist will work correctly in Atmel Studio 7
- **Chip Programmer**
  - Added support for the USBASP programmer
  - The chip programmer now supports AVRISP MkII, AVR Dragon, STK600, JTAGICE mKII, JTAGICE3 and AT ATMELICE without the need for Atmel Studio to be installed

V3.31 Commercial Release

- **Compiler**
  - Added support for the SSD1309 OLED display controller and corresponding example program
  - Improved error and warning reporting
  - Fixed: assembly error when the *cmp64u* function from *math64.h* was invoked
  - Fixed: *#pragma warn_data_stack_recursion+/-* didn't work
- **CodeWizardAVR**
  - Fixed: the CodeWizardAVR for Xmega chips didn't #include the appropriate touch screen support header file in the 'timers_init.c' source file
- **IDE**

- Added the possibility to upload compiled code to custom Arduino compatible bootloaders for all chips that have a boot area

V3.30 Commercial Release

- **Compiler**
  - Added compiler support for the ATtiny417 and ATtiny817 chips of the AVR8X family
  - Added support for SPI connection for the RA8875 graphic display controller
  - Added new examples: Atmel AVR1916 USB DFU Bootloader and ATtiny817 Xplained Mini
  - Fixed: incorrect ELF file was created if the options to include FLASH data and chip signature were not enabled
  - Fixed: argument values were incorrectly passed to an inline function, that was used to initialize a local variable during its declaration
  - Fixed: a warning *"unused function removed by the linker"* was issued when the result of an inline function was used to initialize a local variable during declaration
  - Fixed: when ANSI char to int promotion was enabled, an *unsigned* char struct bit field was considered as *signed* char when promoted to int
  - Fixed in the USB device library: incompatibility with the ATxmega64A1U/128A1U due to a bug in silicon for these chips
  - Fixed in the *glcd_types.h* header file: images with more than 65535 pixels from external memory were not displayed correctly
  - Fixed in the *tiny4313.h* header file: PCINT9, PCINT10 bit allocations for PCMSK1 register and PCINT13..PCINT17 bit allocations for PCMSK2 register
  - Fixed the *io.h* header file which failed to #include the header file for ATmega645P
- **CodeWizardAVR**
  - Fixed: The CodeWizard allowed to select any I/O port and bit, even if the hardware SPI interface was used for connection with a graphic display controller
- **IDE**
  - Improved the displaying of the function list of a C source file in the top of the editor pane
  - Improved the loading speed for large projects
- **Chip Programmer**
  - Added support for the Atmel mEDBG programmer (requires Atmel Studio 7 to be installed)
  - Added support for programming the fuse bits for the ATtiny4/5/9/10/20/40 chips
- **LCD Vision**
  - Added the possibility to import bit-mapped Windows fonts
  - Added the possibility to convert an image that is larger than the dimmensions supported by the graphic display controller. Useful for creating scrolled images.

V3.29 Commercial Release

- **Compiler**
  - Added support for the ATmega1284RFR2, ATmega2564RFR2 and ATmega644RFR2 chips
  - Improved error checking for cases when the '&' unary operator is applied to a Xmega register variable
  - Fixed: The program build counter was not incremented when using the Atmel Studio extension
- **IDE**
  - Fixed incompatibility of the commercial version's copy protection with the new update of MS Windows Defender.

V3.28 Commercial Release

- **Compiler**
  - Added support for the Atmega324PB chip
  - Improved the common sub-expression optimizer when accessing structures located in FLASH for the medium and large memory models
  - Added #pragmas to individually enable/disable different types of warnings from within the source code
  - Improved error and warnings checking
  - Improved the generated code when initializing local pointer variables during declaration
  - Added assembly definitions for the SPDR and SPSR registers of the ATmega 16M1/32M1/64M1/32C1/64C1 chips
  - Fixed: Executing a 'return' statement that calls a function which returns a structure/union, left the interrupts globally disabled
  - Fixed: In enhanced function parameter passing modes 1 and 2 only, when calling functions by an auto pointer, and the last function argument was a pointer to a local scalar variable, the called function address was not set correctly
  - Fixed: incorrect temperature measurements in the BMP085 and BMP180 libraries when the ambient temperature exceeded 60 degrees C
- **CodeWizardAVR**
  - Added support for the Atmega324PB chip
  - Fixed in the CodeWizard for Xmega devices: the Output Compare pin remapping was reset, when switching from one timer to another in the Settings panel
- **IDE**
  - Added: Clicking on a C identifier in the editor highlights all its occurrences in the file
  - Added support for the Artistic Style automatic source code formatter.

V3.27 Commercial Release

- **Compiler**
  - Added support for the ATtiny102/104 chips
  - Added libraries for the Ilitek ILI9331 and Sitronix ST7781 TFT controllers (advanced license only)
  - Improved controller chip detection for the SSD1289, ILI9325 and ILI9328 TFT controller libraries
  - Improved warnings reporting
  - Fixed a bug in the optimizer that appeared in V3.26: in certain very rare cases, only when optimization for Size was enabled, applying the ++ or – operators to a structure member or array element, affected the next RAM location instead of the correct one
  - Fixed the internal graphic buffer overflow when calling the *glcd_clear* and *glcd_init* functions in the UC1610 graphic display controller library
- **CodeWizardAVR**
  - Added support for the ATtiny102/104 chips
  - Fixed: the code generated for ATtiny441/841 in the Timer2 overflow interrupt service routine referenced TCNT2 instead of TCNT2H, TCNT2L registers
  - Fixed a bug that appeared in V3.26: a "List index out of bounds" error occurred when generating code for the ATmega1281/2561 chips
- **LCD Vision**
  - Added support for the Ilitek ILI9331 and Sitronix ST7781 TFT controllers.

V3.26 Commercial Release

- **Compiler**
  - Added library for the Bosch Sensortec BMP280 I^2C pressure and temperature sensor
  - Added library for the Bosch Sensortec BME280 I^2C humidity, pressure and temperature sensor
  - Added library for the ROHM Semiconductor BH1750FVI I^2C light intensity sensor
  - Added library for the Sitronix ST7789S TFT controller (advanced license only)
  - Improved the libraries for operating the ILI9340/ILI9341 TFT controllers in landscape mode
  - Added support for the ATmega328PB chip
  - Improved the speed of the code generated for de-referencing pointers to 16 and 32-bit variables
  - Using *#asm("cli"), #asm("sei"), #asm("nop"), #asm("wdr")* doesn't prevent function inlining anymore

- Fixed: a 32-bit overflow warning appeared in some cases in the preprocessor when the first operand of the '<<' operator was a negative constant
- Fixed: the ltoa64 function (math64.h) didn't output anything for input value 0

- **CodeWizardAVR**
  - Added support for the ATmega328PB chip
  - Added PRR register setting for ATmega48PB/88PB/168PB chips
- **IDE**
  - Improved the code autocomplete feature in the text editor
  - Fixed: Fuse and lock bits settings were not saved in the ELF file when using the Chip Programmer's *File|Save ELF File* menu
- **LCD Vision**
  - Added support for the Sitronix ST7789S TFT controller (advanced license only)
  - Made small improvements to the user interface.

V3.25 Commercial Release

- **Compiler**
  - Added the possibility, in the Project|Configure|C Compiler|Advanced menu, to reserve the usage of registers R2..R14 for user's inline assembly code or other needs, thus preventing their automatic allocation for global variables
  - Added the predefined chip type macro __ATxxxxxxxx__ for compatibility with AVR GCC
  - Modified the preprocessor to evaluate integer expressions on 64-bits instead of the 32-bits like in previous versions
  - Improved error reporting for compiler internal errors
  - Increased the size of the internal inline function code buffer
  - Added the definitions for the SPI signal bit masks *SPI_SS_bm, SPI_SCK_bm, SPI_MISO_bm and SPI_MOSI_bm* in the *xmbits_e5.h* header file
  - Added support for hardware SPI mode for the ST7565, ST7567 and UC1610 graphic display libraries
  - Added the *glcd_printf* and *glcd_printfxy* formatted output functions, *glcd_boxsize, glcd_boxsizef, glcd_boxsizee, glcd_outtextoffs, glcd_outtextoffsf* and *glcd_outtextoffse* functions in the graphic display library (*graphics.h*)
  - Added the *lcd_printf* and *lcd_printfxy* formatted output functions in the alphanumeric LCD libraries (*alcd.h, alcd_ks0073.h, alcd_ssd1803.h*)
  - Added the *fvprintf* function in the Standard I/O library (*stdio.h*)
  - Added the *strchrf* and *strrchrf* functions in the char string library (*string.h*)
  - Enabled the TWIE peripheral in the header file *xmega32d4.h*

- Added definitions for the DMA peripheral registers in the header file for the ATxmega384C3 chip
- Fixed the definition from *.equ NVM_CMD_READ_USER_SIG_ROW_gc=0x03* to *.equ NVM_CMD_READ_USER_SIG_ROW_gc=0x01* in the file *sp_driver.c* of the Xmega AVR1316 and AVR1605 Bootloader examples supplied with the compiler
- Fixed in the graphic display library (*graphics.h*): the *glcd_imagesize* function returning 0 for *width=_GLCD_MAXX_* or *height=_GLCD_MAXY_*
- Fixed bugs in the *glcd_block* and *glcd_lineh* functions (*graphics.h*) for 256x64 displays with the SSD1322 controller
- Fixed *reverse_x* and *reverse_y* not working in the library for the UC1610 display controller
- Fixed: the *glcd_rectrel* function from the graphic display library drawed the rectangle 1 pixel wider and taller than specified

- **CodeWizardAVR**
  - Improved the code generated for initializing the watchdog of non-Xmega chips
  - Fixed: code was generated using the symbol *ADC_CURRLIMIT_LARGE_gc* instead of *ADC_CURRLIMIT_HIGH_gc* for XMega C4 chips
  - Fixed the displayed ADC sample time specified for the Xmega B, C, D, E devices
  - Fixed: the I/O port used when checking the event system input source for Xmega E chips, as there is no PORTB for these chips
  - Updated for the ATxmega16/32C4 chips, the *USARTC0* signals on I/O *PORTC* pins according to the latest Atmel datasheet: *XCK0 - PC1, RXD0 - PC2, TXD0 - PC3*
  - Fixed: for ATxmega64/128A1U no code should be generated for the non-existent DAC *TIMCTRL* register
  - Fixed: for Xmega chips the *OCA* and *OCB Polarity* output settings were not retained when changing from one timer to another
- **IDE**
  - Added an enhanced code autocomplete feature in the text editor
  - Added in the *Project|Configure|C Compiler|Messages* menu the possibility to enable or disable the error message regarding the code size exceeding FLASH memory
  - Added the option in the *Project|Configure|C Compiler|After Build* menu to enable or disable automatic uploading to EEPROM during Arduino programming
  - Added in the Chip Programmer the *TOSCSEL* fuse bit for the Xmega B1, B3, C4 and D4 chips
  - Removed in the *Chip Programmer* the *DVSDON* fuse bit, which was also removed by Atmel from the datasheets for Xmega chips
- **Atmel Studio Extension**
  - Improved compatibility with Atmel Studio 7
- **LCD Vision**
  - Added the possibility to add or delete characters to the currently edited/imported font

- The last graphic controller selected during image editing/import is now remembered
- Fixed: Monochrome images were always saved as 24-bit RGB colour ones
- Fixed: font color preview showed only black/white after switching from editing an image for a monochrome graphic controller.

V3.24 Commercial Release

- **Compiler**
  - Added support for the Electronic Assembly DOGXL160-7 160x104 graphic displays with the UC1610 controller, with code examples
  - Added support for the Electronic Assembly DIP203 alphanumeric displays with the SSD1803 controller
  - Added the definitions for the ADC6D and ADC7D bits of the DIDR0 register in the *mega168pb_bits.h* header file for the ATmega48PB/88PB/168PB chips.
- **CodeWizardAVR**
  - Added the possibility to initialize the LM75, DS1621, PCF8563, PCF8583, DS1307, DS3231, BMP085, BMP180, MS5611-01BA TWI peripherals
  - Updated the ADC settings for ATmega48PB/88PB/168PB chips so that the digital buffers on inputs ADC6 and ADC7 can be disabled
  - Fixed: the PUEx I/O port pull-up enable registers were not correctly set for the ATtiny1634/20/40/441/828/841 chips.
- **IDE**
  - Improved the way the #include and Library paths are specified in the project configuration
  - Added in the *Settings|IDE* menu the possibility to enable/disable displaying the edited file path and function list in the editor pane.
- **Atmel Studio Extension**
  - Added support for Atmel Studio 7.0
  - Ensure that the CodeWizardAVR window gets focus when invoked from within Atmel Studio.

V3.23a Commercial Release

- **IDE**
  - Improved compatibility with Windows 10.

V3.23 Commercial Release

- **Compiler**
  - Added support for the Maxim DS3231 Real Time Clock, with code examples
  - Added support for ATxmega32D3 chip

- Updated the KS0108 graphic library so it can be used with a system clock up to 32 MHz
- Added support for graphic LCD displays with 160x80 resolution for the S1D13700 controller
- Improved the speed of the generated code for allocating/deallocating local register variables in interrupt service routines, when optimization for speed was selected
- Improved the speed/size of the generated code for allocating/deallocating local register variables
- Improved the speed/size of the generated code for initializing local variables during declaration
- Improved the C preprocessor
- Improved warning reporting
- Fixed: the LD R31,Z instruction sets the R31 register to 0 for AVR8L core chips: ATtiny4/5/9/10/20/40. This affected correctly reading 16-bit int values larger than 255 from FLASH for these chips.

- **CodeWizardAVR**
  - Modified the CodeWizardAVR for Xmega chips to create separate source files for initializing each group of peripherals
  - Added the possibility to jump to the definition of a function using a drop down list located in the top of the CodeWizard program preview window
  - Removed the non-existent AWEX for TCE0 for XMega A3U chips in the CodeWizardAVR and header file
  - Added the possibility in CodeWizardAVR to remap the Xmega E5 USART0 signals, when operating in the USART in SPI mode
  - Fixed: the Timer/Counter Interrupt settings for Xmega chips didn't fit in the screen of some laptops with low vertical display resolution
  - Fixed: the CodeWizardAVR for non-Xmega chips didn't generate code for USART(s) communicating with 9-bit data
  - Fixed: for the ATtiny441/841 chips the PRR register was not updated with the state of the enabled timers
  - Fixed: when assigning a value larger than 0xFF to the ATtiny261/461/861 10-bit Timer 1 OCR1A register, it wasn't truncated to 8-bits as required. However the TCH register was initialized correctly with the upper bits 8 and 9.

- **IDE**
  - Fixed: the “ character was not saved when specifying the *Command Line Parameters* in the *Tools*|*Configure*, *Project*|*Configure*|*Before Build* and *Project*|*Configure*|*After Build*|*Execute Program*|*Program Settings* menus.

- **Atmel Studio Extension**
  - Fixed: hint shadows remaining on the screen.

V3.22 Commercial Release

- **Compiler**
  - Added support for the ILI9225 and S6D0164 TFT display controllers with code examples
  - Added support for the SSD1331, SSD1332, SSD1351 color OLED display controllers with code examples
  - Modified the ILI9325 library to be compatible with the equivalent Raydium RM68090 TFT controller
  - Added the *glcd_ellipse* and *glcd_fillellipse* functions to the main graphic library, previously these functions being available only for the RA8875 TFT controller
  - Improved the generated code and reported warnings for the <<, >> operators when the second operand is a null or negative constant
  - Made small improvements to the code optimizer
  - Fixed a bug in the dead code optimizer, which appeared in V3.20, when testing the value of a *bit* type variable.

    Example:

    ```
    bit flag;
    unsigned char var;

    if (flag) var=92; else var=92;
    ```

    didn't remove the SBIS instruction which tested the *flag* variable, even if the same value was assigned to *var*.

    Now this code is optimized to:

    ```
    var=92;
    ```

- **IDE**
  - Added the possibility to jump to the definition of a function using a drop down list located in the top of the text editor's active panel
  - Made small improvements to the editor.

V3.21 Commercial Release

- **Compiler**
  - Added support for 64-bit integer mathematical operations (math64.h), with code example
  - Added support for the Bosch Sensortec BMP085, BMP180 and Measurement Specialties MS5611-01BA pressure sensors, with code examples

- Improved the speed of the <<, >> operators applied to 16 and 32 bit operands, when the second operand is a variable
- Improved the unsigned multiplication runtime library for ATtiny chips
- Improved the runtime library functions for the signed % operator
- Enabled the TWIE interface for new revisions of the Xmega D3 chips, which now have the bugs in silicon fixed
- Fixed: relative branch out of range assembly errors which occurred in some very rare situations

- **CodeWizardAVR**
  - Enabled the TWIE interface for new revisions of the Xmega D3 chips, which now have the bugs in silicon fixed
- **IDE**
  - Added the possibility to reset the Arduino boards using the corresponding button in the Terminal.

V3.20 Commercial Release

- **Compiler**
  - Added library support (with example) for the FocalTech Systems FT5206, FT5306 and FT5406 capacitive touchscreen controllers
  - Added support for 480x272 display resolution for the RA8875 controller library
  - Improved the code optimizer
  - Modified in the *ff.h* header the definitions for *_FS_READONLY, _FS_MINIMIZE, _FS_TINY, _FS_RPATH, _FS_REENTRANT, _USE_MKFS, _USE_FORWARD, _USE_LFN, _CODE_PAGE, _DRIVES, _MULTI_PARTITION* macros, so that they can be globally defined in the project configuration menu: *Project|Configure|C Compiler| Globally #define*
  - Fixed: When *#pragma data_allignment=2* or higher was used, the COFF debugging information sometimes didn't match the real RAM address of the global variables, which was correctly displayed in the .map file
  - Fixed in the *cvavr\Examples\Xmega\AVR1316 Bootloader\sp_driver.c* and *cvavr\ Examples\Xmega\AVR1605 Bootloader\sp_driver.c* files: in the *SP_ReadFlashPage* function the FLASH page counter in R24 should count WORDS, not bytes
- **CodeWizardAVR**
  - Fixed: ATmega1284 chips didn't have the option to set the CLKPR clock prescaler register
  - Fixed: The CodeWizard didn't allow to enable quadrature encoder support in the event system for the Xmega E5 chips
  - Improved quadrature encoder support in the CodeWizard for Xmega chips

- Fixed: Code generated for I/O Port ISRs by the CodeWizardAVR for Xmega chips other than E5, must not test and clear the PORTn.INTFLAGS register bits, as these are cleared automatically when the corresponding interrupt is serviced
- Improved the code generated by the CodeWizardAVR for processing Xmega E5 I/O port interrupts
- Added a check in the CodeWizardAVR for Xmega chips, that the ADC clock frequency does not exceed 2000kHz for A and AU chips and 1800kHz for the rest

- **IDE**
  - Improved the functionality of the Terminal sending a hex char: there is no need to press Enter to validate the entered hex number.

V3.19 Commercial Release

- **Compiler**
  - Added support for the RA8875 color TFT display controller (the library uses hardware acceleration for drawing lines, triangles, rectangles, circles, ellipses and fill operations)
  - Added the *glcd_triangle*, *glcd_filltriangle*, *glcd_fillrectround*, *glcd_ellipse* (RA8875 only) and *glcd_fillellipse* (RA8875 only) functions to the graphic display library
  - Made a small code optimization in the *glcd_line* function in the graphic display library
  - Modified the SSD1306 library to also support 128x32 OLED displays
  - Added in *glcd_ssd1306.h*, the *external_vcc* member to the *GLCDINIT_t* structure in order to enable/disable the generation of the Vcc voltage, using the internal DC/DC converter
  - Updated the XG7100 TFT controller library and header file
  - Added the definition for *_GLCD_FILL_PATTERN_HEIGHT_* in *glcd_types.h*
  - Fixed: an error occured after compiling a bootloader project and after that compiling an application project for a chip that doesn't have a boot area
- **CodeWizardAVR**
  - Improved the code generated by the wizard for Xmega I/O port interrupt service routines
- **IDE**
  - Added the option to wait for the user program, executed after project build, to finish before executing any other task
  - Improved the compatibility with Windows 8 and 8.1.

V3.18 Commercial Release

- **Compiler**

- Fixed: When the new Enhanced Function Parameter Passing Mode 2 (added in V3.16) was used and the size of the automatic local variables, initialized with constant values during declaration, exceeded 7 bytes, the value of the last 8 or 16 bit argument passed to a function was trashed. **Note:** this didn't affect old projects, that were created with Enhanced Function Parameter Passing Modes 0 or 1.
- Fixed: Atmel Studio extension's error/warning messages were truncated when CodeVisionAVR was installed under Windows 8.1.
- Corrected, in the Help and User Manual, the name of the USB CDC Virtual Serial Port Function from *usb_serial_rxavailable* to *usb_serial_available*, to match the header *usb_cdc.h*.
- Added a CAN example for the Atmega32M1 chip.

- **Chip Programmer**
  - Added support for the Atmel-ICE programmer (requires Atmel Studio 6.2.1502 to be installed).
  - Temporarily disconnect the Terminal during the upload to an Arduino board, if they both share the same COM port.
  - Added chip programming status display window for the Atmel Studio 6 extension.
  - Fixed: The Arduino uploader didn't recognize the ATmega8L/48PB/88PB/168PB chips for selecting a compatible board.

V3.17 Commercial Release

- **Compiler**
  - Added support for the ATmega48PB/88PB/168PB chips.
  - Added SPI support for the ILI9340 and ILI9341 TFT display controllers (Advanced version license). A new example is provided for the ILI9341 in SPI mode.
  - Improved error reporting for functions that were declared, but never defined. Issue the errors only if such functions were actually referenced in the program.
  - Added an option in the *Project|Configure|C Compiler|Advanced* menu (Advanced version license) to use the mangled name, when watching static global variables in the debugger. Useful when static variables with the same name were declared in different C program modules. If the option is enabled, the .map file will display the mangled name of the static variable.
  - Fixed: hardware stack unbalance in *setjmp.lib*
  - Fixed: USB library not working for Xmega chips in the Large memory model (bootloaders).
- **CodeWizardAVR**
  - Added support for the ATmega48PB/88PB/168PB chips.
  - Fixed: Incorrect CodeWizardAVR initialization of the *OSC.DFLLCTRL* register of the Xmega D3 and D4 chips, when the calibration of the internal 32 MHz RC oscillator was enabled.

- **Chip Programmer**
  - Added support for the ATmega48PB/88PB/168PB chips.
  - Enabled ATtiny441/841 programming support for STK600, AVR Dragon, JTAG ICE MkII and JTAG ICE3. Atmel Studio 6.2.1502 must be installed.
  - Added ELF production file support to the Arduino uploader.
- **IDE**
  - The Editor windows may now be arranged in a more flexible way in the workspace.
  - Added the possibility to compare the text from two files opened in the Editor.
  - Improved the appearance of the Terminal.

V3.16 Commercial Release

- **Compiler**
  - Added floating point support in (s)scanf. It can be enabled in the *Project|Configure|C Compiler|Code Generation|(s)scanf* options menu.
  - Added support for the SSD1306 OLED display controller, with code examples.
  - Added a new enhanced function parameter passing *Mode 2* that allocates the 1 or 2 byte function parameter(s) to register(s) inside the function, allowing further code size reduction and increasing program execution speed. It can be enabled in the *Project|Configure|C Compiler|Code Generation|Enhanced Parameter* menu. If this option is enabled for older projects, that use C functions written using inline assembly, these functions must be enclosed between *#pragma dstack_par+* and *#pragma dstack_par-* directives, in order to force using the old parameter storage *Mode 1* in the data stack.
  - Improved the code generation for inline functions.
  - Made various improvements in the code generator.
  - Added support for generating ELF production files for device programming in Atmel Studio.
  - Added the *#pragma fuses* and *#pragma lock_bits* directives to enable setting the fuses and lock bits from the source file and storing them in the ELF file.
  - Added support for programming the Moteino boards from www.lowpowerlab.com/Moteino.
  - Fixed: an exception occured in the Atmel Studio extension if programming errors occured during program uploading to an Arduino board.
- **CodeWizardAVR**
  - Fixed: in the code generated for ATtiny441/841 chips, the *REMAP* register must be initialized before enabling the USART0. Otherwise it will not have any effect.
  - Fixed: in the CodeWizardAVR generated code for ATtiny441/841 chips, the *REMAP* register must be initialized before enabling the SPI. Otherwise it will not have any effect.
  - Added PRR register initialization in the code generated for the ATtiny441/841 chips.

- Fixed: The CodeWizardAVR for Xmega chips dind't place the *_ALTERNATE_GETCHAR_* and *_ALTERNATE_PUTCHAR_* macro definitions before *#include <stdio.h>* when alternate versions of the *getchar* and *putchar* functions were defined.

- **Chip Programmer**
  - Added the possibility to load and save ELF production files.
  - Fixed: the chip programmer displayed an EEPROM buffer option for AVR8L chips, that do not have EEPROM.

V3.15 Commercial Release

- **Compiler**
  - Added support for the Ilitek ILI9328 (240x320 and 320x240 display modes) and Delcomp XG3A TFT controllers
  - Added support for 320x240 landscape display mode for the ILI9325, ILI9340, ILI9341, S6D1121 and SSD1289 TFT controllers
  - Added additional examples for these controllers, operating in landscape display mode
  - Added 8-bit interface support for the ILI9325 and SSD1289 TFT controllers
  - Updated the Delcomp XG7100 TFT controller library
  - Improved the CDC initialization in the USB device mode library for XMEGA chips
  - Improved the generated code for functions returning a structure or union
  - Added a warning when the unary '&' operator is applied to a register variable for the Xmega chips and its result is used for operations with pointers (for these chips the registers are not mapped to RAM addresses 0..31)
- **CodeWizardAVR**
  - Fixed for Xmega chips: no code was generated for analog comparator's interrupt service routines

V3.14 Commercial Release

- **Compiler**
  - Fixed: U0MAP and SPIMAP bit definitions of the REMAP register in the *tiny441.h* and *tiny841.h* header files
  - Fixed: the CCP register was not first set to 0xD8, when the MCUCR register of the ATtiny828 chip was initialized in the start-up code after reset
  - Added a SD Card example for the ATxmega128A4U chip
- **CodeWizardAVR**
  - Fixed for Xmega chips: when the 32 MHz internal RC oscillator was adjusted and calibrated to 48 MHz by de DFLL, using the USB Start of Frame, its frequency was considered to be 32 MHz instead of the correct 48 MHz

- Fixed: when the CLKPR register was initialized for the ATtiny441/828/841 chips, the wizard didn't first set the value of the CCP register to 0xD8

V3.13 Commercial Release

- **Compiler**
  - Added USB device mode library support and examples for Xmega chips
  - Added an USB composite device example for non-Xmega chips
  - Corrected the bootloader sizes for ATmega16M1/32M1 chips
  - Added the *usb_serial_select, usb_serial_getbuf, usb_serial_putbuf, usb_serial_putbuff and usb_serial_putbufe* functions in *usb_cdc.h*
  - Modified the *usb_serial_available* function from *usb_cdc.h* to return *unsigned char* data type only
  - Added the possibility to enable/disable the generation of the *WDR* instruction during EEPROM access and *delay_ms* function execution, in the *Project|Configure|C Compiler|Advanced* menu (Advanced license only)
  - Fixed: *#pragma data_alignment* didn't work correctly for values larger than 2
  - Fixed: *#pragma code_alignment* didn't work correctly for values larger than 4
- **CodeWizardAVR**
  - Added USB device mode support for Xmega chips
  - Improved the code generated by the CodeWizardAVR for 2 and 32 MHz RC oscillators using the DFLL for the Xmega chips
  - Fixed: Replaced 'ADC_REFSEL_VCCDIV2_gc' with 'ADC_REFSEL_INTVCC2_gc' in the code generated by the CodeWizardAVR for Xmega E5 chips when selecting the ADC reference as Vcc/2
  - Fixed: The PLL for Xmega chips can use the 32 MHz internal oscillator's output only divided by 4, not directly
  - Fixed: ATmega8/16/161/162/163/32/323 chips don't have an Input Capture register and interrupt for Timer/Counter 2
- **Chip Programmer**
  - Added to the Arduino uploader, a code size check to prevent overwriting the bootloader area
  - Improved error reporting for chip programming

V3.12 Commercial Release

- **Compiler**
  - Improved CodeVisionAVR project export to Atmel Studio when there is no exact chip type match

- When importing a CodeVisionAVR project in Atmel Studio 6.2, ensure that the active build configuration of the new created Atmel Studio solution matches the one from the CodeVisionAVR project .prj file
- Fixed errors which occurred when using the USB library for the ATmega U2, U4 and AT90USB82/162 chips

- **CodeWizardAVR**
  - Fixed: ADC reference type and clock frequency settings were not correctly read from the CodeWizardAVR .cwp project file
- **Chip Programmer**
  - Fixed: Reading chip signature, lock bits and calibration bytes didn't work correctly for STK600, AVRISP, AVR Dragon and JTAGICE when Atmel Studio 6.2 was installed

V3.11 Commercial Release

- **Compiler**
  - Added an USB device library for non-Xmega chips
  - Added new USB examples for CDC Virtual Serial Port, HID Keyboard, HID Mouse, HID Joystick and Generic HID
  - Improved the code optimizer
  - Updated the header files for the Xmega A1U, B1, B3, C3, C4, E5 devices, so that the access to virtual ports VPORT0..3 structures will be performed correctly
  - Improved inline functions support
  - Improved warnings reporting
  - Fixed: functions that contained a return statement in the last line, were not added to the .map file
  - Fixed: calling a function, using a pointer stored in EEPROM, trashed the last parameter passed in R26, R27 registers when the enhanced parameters passing option was enabled in the project configuration
  - Corrected boot area size for ATxmega8E5: 1024 words instead of 2048 words
  - Fixed a compilation error in ILI9340 and ILI9341 libraries when using a data bus with bits in non-consecutive ascending order
  - Fixed the non-local jump functions (setjmp.h) not working for non-Xmega chips with > 128k bytes of FLASH
  - Fixed the non-local jump functions (setjmp.h) not working for Xmega chips with > 128k bytes of FLASH, including the boot area
  - Fixed: assembly errors when the 1 Wire functions were used with ATmega64/128 I/O ports F and G.
- **CodeWizardAVR**
  - Redesigned the wizard to support the new USB library

- Improved the code generated by the wizard for ATmega16/32U4 Timer 4 and PLL
- Fixed: incorrect generated names WDT_WPER_xxxCLK_gc for the watchdog of all Xmega chips except A, for 125, 250, 500 ms timeout intervals
- Fixed: "Cannot focus a disabled or invisible window." message in CodeWizardAVR for Xmega after disabling a Timer/Counter
- Fixed: incorrect initialization, by the CodeWizardAVR, of the Timer 3 clock prescaller in the TCCR3B register for the ATmega162 chip

- **Chip Programmer**
  - Fixed: Arduino Mega 1280 boards use 115200 Baud rate

V3.10 Commercial Release

- **Compiler**
  - Added support for the UltraChip UC1608 graphic LCD controller
  - Improved the code optimizer
  - Added full support for inline functions
- **CodeWizardAVR**
  - For Xmega A1U, A3U, A4U and E5 chips corrected the checks, so that the DAC must be enabled when its output is connected to one of the analog comparator's inputs

V3.09 Commercial Release

- **Compiler**
  - Added support for the Samsung S6D1121 colour TFT LCD controller. Added code example for this controller
  - Added support for the ATtiny441 and ATtiny841 chips
  - Improved the code optimizer
  - For the Advanced version, added the possibility to customize the start-up initialization code (watchdog timer disable) in the Project|Configure|C Compiler|Advanced menu. For new projects, by default, the watchdog timer is not disabled after chip reset anymore, preventing the alteration of the MCUSR register
  - Added the pgmspace.h, eeprom.h and inttypes.h header files for compatibility with AVR GCC
  - Modified the start-up code for XMEGA chips that don't have RAMPX, RAMPY, RAMPD, RAMPZ registers
  - Fixed the __GETW1STACK, __GETD1STACK, __CALL2EN macros which trashed the R26, R27 registers, used for passing the last parameter when a function was called using a pointer to function, enhanced parameter passing was enabled and the function also referenced by pointer a register variable

- Corrected the MOSI and SCK SPI signals allocation for PORTC of XMEGA E5 chips in the SD Card library
- Updated the SD Card library to be compatible with XMEGA E5 chips
- Added SD Card example for the ATxmega32E5 chip

- **CodeWizardAVR**
  - Added support for the XMEGA E5 chips timer/counters TC4 and TC5
  - Improved the code generated for XMEGA chips that have only one ADC channel
  - Improved the code generated by the wizard for XMEGA E5 chips SPI initialization
  - Improved the code generated by the wizard for XMEGA ADC initialization
  - Improved the code generated by the wizard for XMEGA ADC offset compensation in unsigned conversion mode
  - Improved the code generated by the wizard for XMEGA RTC/RTC32 initialization
  - Added the possibility for XMEGA chips, to specify the phase for channels B, C, D outputs in timer frequency generation mode
  - Fixed: ADC configuration settings were not fully read from the CodeWizardAVR .cwp project file
  - Fixed the code generated for the ATtiny2313/4313 watchdog initialization
  - Fixed: Timer/Counter for PORTE doesn't have the Advanced Waveform Extension for the XMEGA D3 and D4 chips
  - Fixed: the XMEGA E5 chips don't have an USB controller, so there should not be any *'USB start of frame'* calibration option for the 32MHz internal oscillator
  - Fixed: MOSI and SCK SPI signals allocation for PORTC of XMEGA E5 chips
  - Corrected the code generated for XMEGA E5 chips ADC reference selection
  - Removed DAC reference selection for PORTD for XMEGA E5 chips
  - Fixed: the CodeWizard didn't allow to select the DAC output as source for the analog comparator's inputs for the XMEGA AU and E5 chips
  - Fixed: ADC noise reduction code generated for the ATtiny828 chip didn't use the *__sm_adc_noise_red* bit

- **CodeVisionAVR Chip Programmer**
  - Improved error reporting for chip signature mismatch when using the ATPROGRAM.EXE command line utility suplied with Atmel Studio
  - Enabled PDI programming using AVR Dragon for the XMEGA B1, B3, A1U, A3U, D3, D4 and E5 chips
  - Enabled ATmega8 programming for JTAGICE and AVR Dragon in Atmel Studio 6.1 SP2 or later
  - Fixed: STK600 supports only PDI programming for the XMEGA D3, D4 and E5 chips
  - Fixed: JTAGICE supports only PDI programming for the XMEGA D3 chips
  - Updated the BODLEVEL voltage values for the XMEGA B, C, D, E chips

- **Atmel Studio Extension**
  - Added compatibility with Atmel Studio 6.2
- **LCD Vision** (Advanced License only)
  - Added the possibility to import Chinese, Japanese, Korean fonts

V3.08 Commercial Release

- **Compiler**
  - Added support for the ILI9340, ILI9341 colour TFT LCD controllers
  - Added support for the ST7567 graphic LCD controller
  - Improved the generated code for functions returning structures or unions
  - Improved the TWI library for XMEGA chips for better handling recovery after an I^2C bus conflict
  - Fixed in the TWI library for non-XMEGA chips: matching any address for slave mode operation was not working
  - Added the missing *ADC_CH_MUXPOS_PIN8_gc...ADC_CH_MUXPOS_PIN11_gc* members of the *ADC_CH_MUXPOS_enum* in the *xmbits_a1.h, xmbits_a3.h, xmbits_a4.h and xmbits_a3b.h* header file
- **CodeWizardAVR**
  - Improved the code generated for the XMEGA E5 ADC and Watchdog timer
  - Fixed the code generated by the wizard for initializing the XMEGA E5 *OSC.DFLLCTRL* register when using the internal 32MHz oscillator, calibrated by the internal 32kHz oscillator
  - Fixed: the *VBAT.STATUS* status bit *BBBODF* was not cleared if the Battery Backup System had a failure, in the code generated by the wizard for the *rtc32_battery_backup_init* function, used to initialize the RTC32 peripheral for the ATxmega256A3B chip
- **CodeVisionAVR IDE**
  - Fixed: the SD card library configuration settings were reset when the active Build configuration was changed in the *Project*|*Configure*|*C Compiler*|*Code Generation* menu
- **Atmel Studio Extension**
  - The CodeVisionAVR installer applies to Atmel Studio 6.1 SP2 a patch (supplied by Atmel) in order to correct some issues in watching variables during debugging of COFF object files
  - Fixed: the SD card library configuration settings were reset when the active Build configuration was changed in the *Project*|*Configure*|*C Compiler*|*Code Generation* menu.

V3.07 Commercial Release

- **Compiler**

- Added the ATxmega192C3 chip
- Added support for the SSD1322 graphic OLED display controller
- Improved the code size of the SSD1963 library
- Fixed an error in the SSD1963 library which lead to incorrect *glcd_block* function display for *left* >= 256 and 256 colour mode
- Improved the resistive touchscreen library so that it can handle displays with 800x480 pixels
- Increased the maximum symbol name length from 64 to 80 characters
- Removed the warning "Expression with possibly no effect" for: (void) (expr)
- Improved error reporting
- Fixed: incorrect union initialization for the example below:

```
typedef union
{
long l;
char c;
} utest;

utest test={.c='A'};   // didn't initialize correctly
utest test={.l=12345}; // initialized correctly
utest test={12345};    // initialized correctly
```

- **CodeWizardAVR**
  - Fixed: for XMEGA D4 chips ports B and E have only 4 pins
  - Fixed: incorrect code was generated for initializing the Timer/Counter *CTRLE* register, when operating the timer in 8-bit mode for the XMEGA D chips
  - Fixed: the wizard failed to initialize the *PCMSK0* register for the ATtiny261/461/861 chips
- **CodeVisionAVR IDE**
  - Added the possibility to upload the executable program to an Arduino compatible development board after a successful *Build* or using the *Tools* menu
  - In the project configuration, selecting the input files didn't work correctly when using the keyboard to select the file in the tree. Mouse selection however worked properly.
- **Atmel Studio Extension**
  - Added the possibility to upload the executable program to an Arduino compatible development board after a successful *Build* or using the *Tools* menu

- Fixed: incorrect project file structure was displayed in the *Project|Configure* menu when the CodeVisionAVR project *.prj* file wasn't in the same directory as the Atmel Studio solution *.atsln* file
- Fixed: the extension didn't report the linker warnings and errors.

V3.06 Commercial Release

- **Compiler**
  - Enhanced the code optimizer
  - Added support for the ATmega64RFR2, ATmega128RFR2 and ATmega256RFR2 chips
  - Added the possibility to display the Function Call Tree in the Atmel Studio, for maximal data stack usage estimation
  - Made various small bug fixes and improvements in the Atmel Studio extension
  - Added the possibility to define additional global macros from the command line
  - Updated the definitions of the *DTPS1*, *DT1A* and *DT1B* registers in *tiny25.h*, *tiny45.h* and *tiny85.h* header files, in order to comply with the latest datasheet
  - Updated the header files for the XMEGA A1U, A3U, A3BU, A4U, B1, B3, D3 devices
  - Updated the XMEGA EBI SDRAM example program.
- **CodeWizardAVR**
  - Added support for the ATmega64RFR2, ATmega128RFR2 and ATmega256RFR2 chips
  - Updated the ADC code generated for the ATmega128RFA1 chip
  - Updated the code generated for the ATtiny1634 *CLKSR* register. Added the possibility to enable system clock output on the *CLKO* pin.
  - Improved the generated code for buffered interrupt driven serial communication, when the buffer size is set to 256 bytes
  - Fixed generated code for initializing the Timer/Counter *CTRLE* register, when operating the timer in 8-bit mode for the XMEGA AU, B and C chips
  - Fixed generated code for the *PORTCFG.CLKEVOUT* register of the XMEGA A chips
  - Fixed generated code for the *PORTCFG.EVOUT* register of XMEGA B1, B3 chips
  - Fixed: It was impossible to edit some values, using the backspace key, in the CodeWizardAVR text and spin edit controls when invoking the wizard from the Atmel Studio 6.1 Tools menu.
- **Chip Programmer**
  - For modern AVR chips, the signature checking is now performed before erasing the chip. For obsolete AT90S chips, this was not possible, if the lock bits were programmed to protect the FLASH.

V3.05a Commercial Release

- **Compiler**

- Added the *CLKCSR* and *CLKSELR* registers to the *tiny87.h* and *tiny167.h* header files.
- Fixed: the compiler stored redundant, not necessary, FLASH initialization data for char array structure members, that were initialized during declaration with literal strings. This bug appeared in V3.03.
- Fixed: assembly errors occured when building for the first time a newly created project (in the CodeVisionAVR IDE only) without using the wizard. This bug appeared in V3.05.
- Fixed: using the *#error* preprocessor directive produced a crash message in the Atmel Studio Extension. This bug appeared in V3.03.

V3.05 Commercial Release

- **Compiler**
  - The Extension for Atmel Studio is now compatible with Atmel Studio 6.1 SP1
  - The CodeVisionAVR installer applies a patch to Atmel Studio 6.1 in order to allow watching structures when debugging COFF files
  - Added support for the ATxmega64A1U/128A1U, ATxmega64B1/128B1, ATxmega64B3/128B3, ATxmega32C3/64C3/128C3/192C3/384C3, ATxmega32C4/64C4 and ATxmega8E5/16E5/32E5 chips
  - Added support for the ADS7843 and ADS7846 resistive touch screen controllers
  - Improved the graphic library for the SSD1963 controller
  - Added support for the Midas Components *MC122032B6W-BNMLW* graphic LCDs with the SED1520 controller, 122x32 resolution and active high chip select/enable signals: CS1, CS2, E
  - Added the *USIBR* register to the *tiny2313a.h* and *tiny4313.h* header files
  - Updated the name of the *PRR* register to *PRR0* in the *mega164a.h, mega324a.h, mega644a.h, mega1284.h* and *mega1284p.h* header files
  - Added the *PRR1* register in the *mega1284.h* and *mega1284p.h* header files
  - Enhanced the handling of the external startup initialization file: startup.asm
  - Fixed: the EEPROM location 0 is not used for XMEGA A chips due to a bug in silicon that prevents correctly writing to this location in EEPROM mapped to RAM mode
  - Fixed: the *glcd_getcharw* function (graphics.h) was not returning a correct value if the last character from the displayed font had ASCII code 0xFF. This affected also the *glcd_putchar* and *glcd_outtext...* functions which call *glcd_getcharw*.
  - Fixed: the definitions for *ADC_CH_MUXNEG_PIN4_gc..ADC_CH_MUXNEG_PIN7_gc* in the *xmbits_a1.h, xmbits_a3.h, xmbits_a3b.h, xmbits_a4.h* and *xmbits_d3.h* header files for the XMEGA A1, XMEGA A3, XMEGA A3B, XMEGA A4 and XMEGA D3 devices.
- **CodeWizardAVR**
  - Fixed: the CodeWizardAVR didn't generate code to initialize the TIMSK register for the ATmega128 chip

- Fixed: the CodeWizardAVR listed as one of the USI clock source options the Timer/Counter 0 overflow (valid only for ATtiny26) instead of Timer/Counter 0 compare match
- Fixed: the CodeWizardAVR generated code using *TWI_SDAHOLD_bm* instead of *TWI_SDAHOLD_gm* for disabled TWI for the XMEGA AU and D chips
- Fixed: the CodeWizatdAVR didn't allow to select the clock source for the ATtiny4/5/9/10 chips.

- **Chip Programmer**
  - Fixed: For AVRSIP MkII and Atmel Studio 6.x a fuse bits programming error message was issued for ATtiny13, even if the programming was correct.

V3.04 Commercial Release

- **Compiler**
  - Fixed: In certain cases the function with the __reset attribute wasn't executed.
  - Improved warning reporting in the Atmel Studio extension.
  - Updated the header files: *xmbits_a1.h, xmbits_a3.h, xmbits_a3b.h, xmbits_a4.h, xmbits_d3.h* for better compatibility with the code generated by the CodeWizard for the XMEGA ADC.
- **CodeWizardAVR**
  - Fixed: The CodeWizardAVR generated code using the *OSC_RC32MCREF_bm* symbol instead of *OSC_RC32MCREF_gm* for the XMEGA D4 chips.
  - The CodeWizardAVR for XMEGA chips generated an unsigned char *tx_counter_usartx* variable instead of unsigned int, for buffered interrupt driven USART transmission, when the TX buffer size was set to 256 bytes.

V3.03 Commercial Release

- **Compiler**
  - Integrated the compiler, as an extension, in Atmel Studio 6.1 or later.
  - Added the possibility to compile from the command line.
  - Added the compiler.h and interrupt.h header files needed for compiling Atmel ASF code.
  - Added support for the SSD1303/SH1101A graphic OLED display controllers.
  - Added support for graphic LCD displays with 132x32 resolution and ST7565 controller (Electronic Assembly DOGM132-5).
  - Added support for the ATtiny261A/461A/861A/828 chips.
  - Added in the .asm file the definition of the MSB register for the global variables allocated to registers. The string '_msb' is appended to the variable name in the definition.
  - Improved checking of the TWI and bit-banged I2C SCL bit rate in the CodeWizard and project configuration.

- Added the possibility to export the CVAVR projects to Atmel Studio 6.1.
- Added the possibility to have 2 build configurations for each project: Debug and Release.
- Added check for Project|Build: if no source file was modified and a COFF object file exists with valid timestamp, then no linking/assembling/COFF generation is performed.
- Updated all header files for XMEGA chips in order to improve compatibility with Atmel Studio 6.1.
- Added the possibility to initialize struct/unions using the .member=value format.
- Added new compiler directive: #pragma data_alignment=value and #pragma code_alignment=value.
- Added the _Pragma operator to comply with C99.
- Added the '_Bit' data type, equivalent to 'bit', in order to prevent naming incompatibilities with Atmel ASF code, when using the CodeVisionAVR specific 'bit'.
- Improved code optimizer.
- Improved compilation speed.
- Increased the number of allowed COFF symbols.
- Updated the default initialization values for 320x240 TFTs with SSD1963 controller in glcd_ssd1963.h.
- Added the predefined macros _MODEL_TINY_, _MODEL_SMALL_, _MODEL_MEDIUM_ and _MODEL_LARGE_ in the assembly generated code in order to allow generating memory model dependent assembly inline code.
- Added the missing CCSR register definition in the mega406.h header file.
- Corrected the DAC_REFSEL_AREFB_gc definition in the header files for the XMEGA A1, A3, A3U, A3B, A3BU, A4, A4U chips.
- Corrected the definition for the CRC_t structure in the header files for the XMEGA A3U, A3BU, D3 and D4 chips.
- Added USB CDC example for the ATxmega128A4U chip.
- Added USB Mass Storage example for the ATxmega256A3BU chip.
- Added an example for displaying images from SD Card on a TFT display with the SSD1289 controller.
- Enabled the Project|Configure|C Compiler|Advanced tab for the ATtiny4/5/9/10/20/40 chips (for Advanced license only).
- Fixed: data stack unbalance when enhanced parameter passing option is enabled in the project configuration and a function with variable number of arguments is called, with the last fixed parameter as a pointer to a char or int variable located in one or two registers.
- Fixed: reversing the Y display didn't work for the ST7565 graphic LCD controller.
- Fixed: the glcd_putimagex function was not accessing data from external memory.
- Fixed: sleep.h functions not working for the ATtiny4/5/9/10/20/40/48/87/88/167 chips.

- Fixed: XMEGA EBI initialization startup code altered the R30 register which needed to be 0 when clearing RAM after reset.
- Fixed: (s)printf "% .1f" format specifier didn't output the leading space for positive numbers.
- Fixed: (s)scanf when processing "%d" and similar formats didn't abort scanning if first char was non-digit.
- Fixed: (s)scanf didn't accept numbers preceded by '+' when processing "%d" and similar formats.

- **CodeWizardAVR**
  - Integrated the wizard, as an extension, in Atmel Studio 6.1 or later.
  - Improved the code generated by the CodeWizardAVR for XMEGA chips when the event system is used for quadrature decoding.
  - Renamed XMEGA AWEX FDEVMASK to FDEMASK in CodeWizardAVR generated code in order to be compatible with new XMEGA headers.
  - Updated the code generated by CodeWizardAVR for the XMEGA timer's high resolution extension.
  - Added the option to generate watchdog interrupts only, without chip reset.
  - Removed the TWI PORTE interface for the XMEGA D4 chips, as it is not functional.
  - Fixed: the "ADC Input Connected to GND for Offset Compensation" listbox listed only inputs 0 to 7 for the ATmega A3 chips.
  - Fixed: the CodeWizardAVR generated an empty assignment to the SFIOR register for the ATmega128/16/163/323/64/8 chips when the ADC was enabled.
  - Replaced SFIOR register with ADCSRB for ATmega16M1/32M1/64M1 chips.
  - Fixed: voltage reference on PORTB was missing from the ADC selection for XMEGA A4 chips.
  - Fixed: generated incorrect bit names for the TCCR1D register of the ATtiny87/167 chips.
  - Fixed: generated incorrect bit name for the ACIRS bit of the ACSR register of the ATtiny87/167 chips.
  - Fixed: The CodeWizardAVR didn't generate code for initializing the TIMSK register of the ATtiny26 chip.
- **Chip Programmer**
  - Fixed: The XMEGA A4 and A4U chips support only PDI programming mode.
  - Fixed: The XMEGA A4 and A4U chips don't have the JTAGEN fuse bit and JTAGUID.

- Updated the definitions for the ATA6286, ATmega328, ATmega1284, ATmega164A, ATmega165, ATmega165A, ATmega165V, ATmega168A, ATmega168PA, ATmega169, ATmega169A, ATmega169L, ATmega169V, ATmega324A, ATmega325A, ATmega325PA, ATmega3250A, ATmega3250PA, ATmega329A, ATmega329PA, ATmega3290A, ATmega3290PA, ATmega48A, ATmega48PA, ATmega644A, ATmega645A, ATmega645P, ATmega6450A, ATmega6450P, ATmega649A, ATmega6490A, ATmega6490P, ATmega88A, ATmega88PA chips in order to be compatible with ATPROGRAM.EXE from Atmel Studio 6.1.

- **CodeVisionAVR IDE**
  - Pressing Tab key when a block of text is selected, right indents it.
  - Added the %bc parameter, substituted with the active build configuration: DEBUG or RELEASE, in the command line passed to the user program in the Before Build and After Build project options.
- **LCD Vision** (Advanced License only)
  - Added support for the SSD1303/SH1101A controllers.
  - Added the possibility to export fonts/images as binary data, not only C code.