

CodeVisionAVR V2 Standard Features

- Application that runs under Windows® 2000, XP, Vista and Windows 7, 32bit and 64bit
- Easy to use Integrated Development Environment and **ANSI C compatible** Compiler
- Editor with auto indentation, syntax highlighting for both C and AVR assembler, function parameters and structure/union members autocomplete
- Supported data types: bit, bool, char, int, short, long, float
- Fast floating point library with **hardware multiplier** and enhanced core instructions support for all the new ATmega chips
- AVR specific extensions for:
 - Accessing the EEPROM & FLASH memory areas
 - Bit level access to I/O registers
 - Interrupt support
 - Support for placing bit variables in the General Purpose I/O Registers (GPOR) available in the new chips (ATtiny2313, ATmega48/88/168, ATmega165/169/325/329/3290/645/6450/649/6490, ATmega1280/1281/2560/2561/640, ATmega406 and others)
- Compiler optimizations:
 - Peephole optimizer
 - Advanced variables to register allocator, allows very efficient use of the AVR architecture
 - Common Block Subroutine Packing (what our competition calls “Code Compressor”), replaces repetitive code sequences with calls to subroutines. **This optimizer is available as Standard in CodeVisionAVR, at no additional costs, not like in our competitor’s products.**
 - Common sub-expression elimination
 - Loop optimization
 - Branch optimization
 - Subroutine call optimization
 - Cross-jumping optimization
 - Constant folding
 - Constant literal strings merging
 - Store-copy optimization
 - Dead code removing optimization
 - 4 memory models: TINY (8 bit data pointers for chips with up to 256 bytes of RAM), SMALL (16 bit data pointers for chips with more than 256 bytes of RAM), MEDIUM (for chips with 128k of FLASH) and LARGE (for chips with 256k or more FLASH). **The MEDIUM and LARGE memory models allow full FLASH addressing for chips like ATmega128, ATmega1280, ATmega2560, etc, the compiler handling the RAMPZ register totally transparently for the programmer. This feature is available as Standard in CodeVisionAVR, at no additional costs, not like in our competitor’s products.**
 - User selectable optimization for code Size or Speed
- Possibility to insert inline assembler code directly in the C source file
- VERY EFFICIENT USE OF RAM: Constant character strings are stored only in FLASH memory and aren't copied to RAM and accessed from there, like in other compilers for the AVR
- C Source level debugging, with COFF symbol file generation, allows variable watching (**including structures and unions**) and the use of the Terminal I/O in Atmel’s [AVR Studio Debugger](#)
- Fully compatible with Atmel’s In-Circuit Emulators: AVR JTAG-ICE, AVR Dragon, etc.
- Chips supported by CodeVisionAVR Standard:

- ATtiny5, ATtiny10 (reduced core)
- ATtiny13, ATtiny13A
- ATtiny167, ATtiny87
- ATtiny22
- ATtiny2313
- ATtiny24, ATtiny44, ATtiny84
- ATtiny25, ATtiny45, ATtiny85
- ATtiny26, ATtiny46, ATtiny86, ATtiny166
- ATtiny261, ATtiny461, ATtiny861
- ATtiny43U
- ATtiny48, ATtiny88
- AT90S2313
- AT90S2323, AT90S2343
- AT90S2333, AT90S4433
- AT90S4414, AT90S8515
- AT90S4434, AT90S8535
- AT90S8534
- AT90CAN32, AT90CAN64, AT90CAN128
- AT90PWM2, AT90PWM2B, AT90PWM3, AT90PWM3B, AT90PWM216, AT90PWM316
- AT90USB1286, AT90USB1287, AT90USB646, AT90USB647, AT90USB162, AT90USB82
- ATmega103
- ATmega128, ATmega1280, ATmega1281, ATmega1284P
- ATmega1284P
- ATmega16
- ATmega16HVA, ATmega 8HVA
- ATmega16HVB, ATmega32HVB
- ATmega16M1, ATmega32M1, ATmega64M1
- ATmega16U4, ATmega32U4
- ATmega161
- ATmega162
- ATmega163
- ATmega164, ATmega164P
- ATmega165
- ATmega169
- ATmega2560, ATmega2561
- ATmega32
- ATmega32C1, ATmega64C1
- ATmega32U6
- ATmega323
- ATmega324, ATmega324P
- ATmega325, ATmega325P, ATmega3250, ATmega3250P
- ATmega329, ATmega329P, ATmega3290, ATmega3290P
- ATmega406
- ATmega48, ATmega48P, ATmega88, ATmega88P, ATmega168, ATmega168P, ATmega328P
- ATmega603
- ATmega64
- ATmega640
- ATmega644, ATmega644P
- ATmega645, ATmega6450
- ATmega649, ATmega6490
- ATmega8
- ATmega8515
- ATmega8535
- **ATxmega128A1, ATxmega192A1, ATxmega256A1, ATxmega384A1, ATxmega64A1**
- **ATxmega128A3, ATxmega192A3, ATxmega256A3, ATxmega256A3B, ATxmega64A3**
- **ATxmega16A4, ATxmega32A4, ATxmega64A4, ATxmega128A4**
- FPSLIC AT94K05, AT94K10, AT94K20, AT94K40
- AT43USB355
- AT76C711
- AT86RF401
- ATA6285, ATA6286, ATA6289

- Supplementary libraries for:
 - Alphanumeric LCD modules for up to 4x40 characters
 - Philips I²C Bus
 - National Semiconductor LM75 Temperature Sensor
 - Maxim/Dallas Semiconductor DS1621 Thermometer/Thermostat
 - Philips PCF8563 and PCF8583 Real Time Clocks
 - Maxim/Dallas Semiconductor DS1302 and DS1307 Real Time Clocks
 - Maxim/Dallas Semiconductor 1 Wire protocol
 - Maxim/Dallas Semiconductor DS1820/DS18B20/DS1822 1 Wire Temperature Sensors
 - Maxim/Dallas Semiconductor DS2430/DS2433 1 Wire EEPROMs
 - SPI
 - **MMC/SD/SD HC FLASH Memory Card drivers and FAT12, FAT16, FAT32 access libraries**
 - Power management
 - Delays
 - BCD and Gray code conversion

- **Built-in CodeWizardAVR Automatic Program Generator, allows you to write in a matter of minutes all the code needed for implementing the following functions:**
 - External memory access setup
 - Chip reset source identification
 - Input/Output Port initialization
 - External Interrupts initialization
 - Timers/Counters initialization
 - Watchdog Timer initialization
 - UART initialization and interrupt driven buffered serial communication with the following parameters: 7N2, 7E1, 7O1, 8N1, 8N2, 8E1 and 8O1
 - Analog Comparator initialization
 - ADC initialization
 - SPI Interface initialization
 - CAN Interface initialization
 - Two Wire Interface initialization
 - I²C Bus, LM75 Temperature Sensor, DS1621 Thermometer/Thermostat, PCF8563, PCF8583, DS1302 and DS1307 Real Time Clocks initialization
 - 1 Wire Bus and DS1820/DS1822 Temperature Sensors initialization
 - LCD module initialization
 - USB controller for the AT90USB646, AT90USB647, AT90USB1286 and AT90USB1287 chips

- Built-in Serial Communication Terminal for debugging RS232, RS422, RS485

- Built-in In-System AVR Chip Programmer, compatible with the Atmel [STK500](#), STK501, STK502, [STK600](#), [AVRISP](#) (serial), [AVRISP MkII](#) (USB), [AVR Dragon](#), [JTAGICE MkII](#), Atmel [AVR910](#) application note, [Kanda Systems](#) STK200+ and STK300 development boards, [Vogel Elektronik](#) VTEC-ISP, [Dontronics](#) DT006, [Tietomyrsky](#) EXB2313, [4Ahead](#) AVR Board 1 and [Futurlec](#) JR-AVR AT90S2313 and AT90S8535 boards, features automatic programming after successful compilation. Supported chips:
 - ATtiny5, ATtiny10
 - ATtiny13/13A
 - ATtiny167/87
 - ATtiny22
 - ATtiny2313
 - ATtiny24/44/84
 - ATtiny25/45/85
 - ATtiny26/46/86/166
 - ATtiny261/461/861
 - ATtiny43U
 - ATtiny48/88
 - AT90S2313
 - AT90S2323/2343
 - AT90S2333/4433
 - AT90S4414/8515
 - AT90S4434/8535
 - AT90CAN32/64/128

- AT90PWM2/3 (B) /216/316
- AT90USB1286/1287/646/647/162
- ATmega603/103
- ATmega64/128
- ATmega1280/1281/1284P
- ATmega16
- ATmega16M1/32M1/64M1
- ATmega16U4/32U4
- ATmega161
- ATmega162
- ATmega163
- ATmega164, ATmega164P
- ATmega165
- ATmega169
- ATmega2560/2561
- ATmega32
- ATmega32C1/64C1
- ATmega32U6
- ATmega323
- ATmega324, ATmega324P
- ATmega325/325P/3250/3250P
- ATmega329/329P/3290/3290P
- ATmega48/88/168, ATmega48P/88P/168P/328P
- ATmega640
- ATmega644, ATmega644P
- ATmega645/6450
- ATmega8
- ATmega8515
- ATmega8535
- **ATxmega128A1, ATxmega192A1, ATxmega256A1, ATxmega64A1**
- **ATxmega128A3, ATxmega192A3, ATxmega256A3, ATxmega256A3B, ATxmega64A3**
- **ATxmega16A4, ATxmega32A4, ATxmega64A4, ATxmega128A4**
- AT86RF401
- ATA6286, ATA6285, ATA6289

For In-System programming of the other AVR chips (AT90S1200, ATtiny12, ATtiny15L) please use our Universal Programming software [ChipBlasterAVR](#)