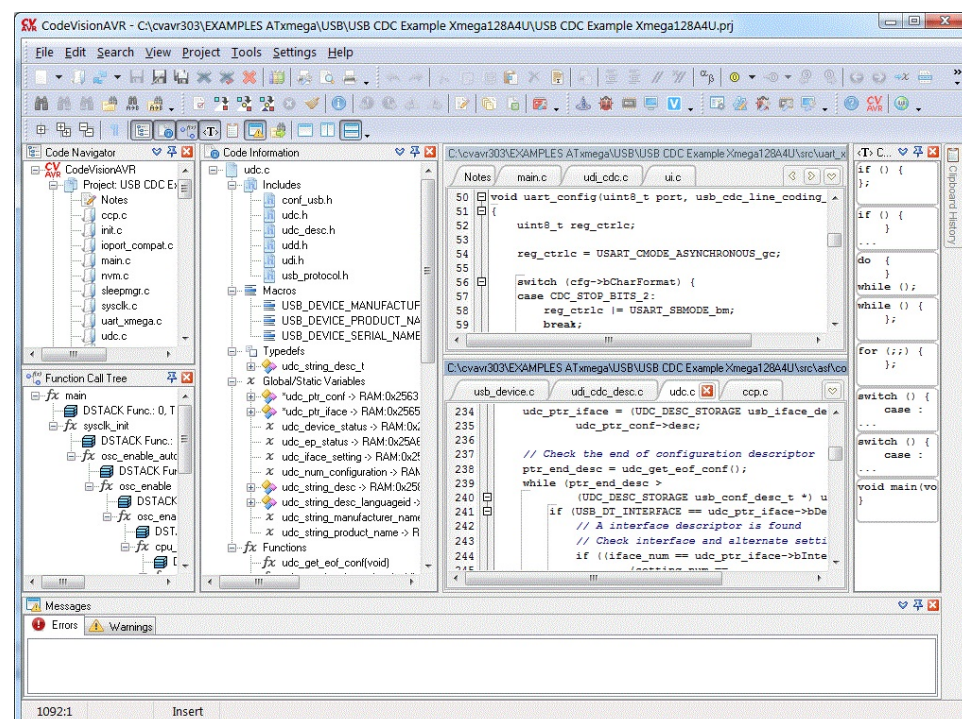


CodeVisionAVR

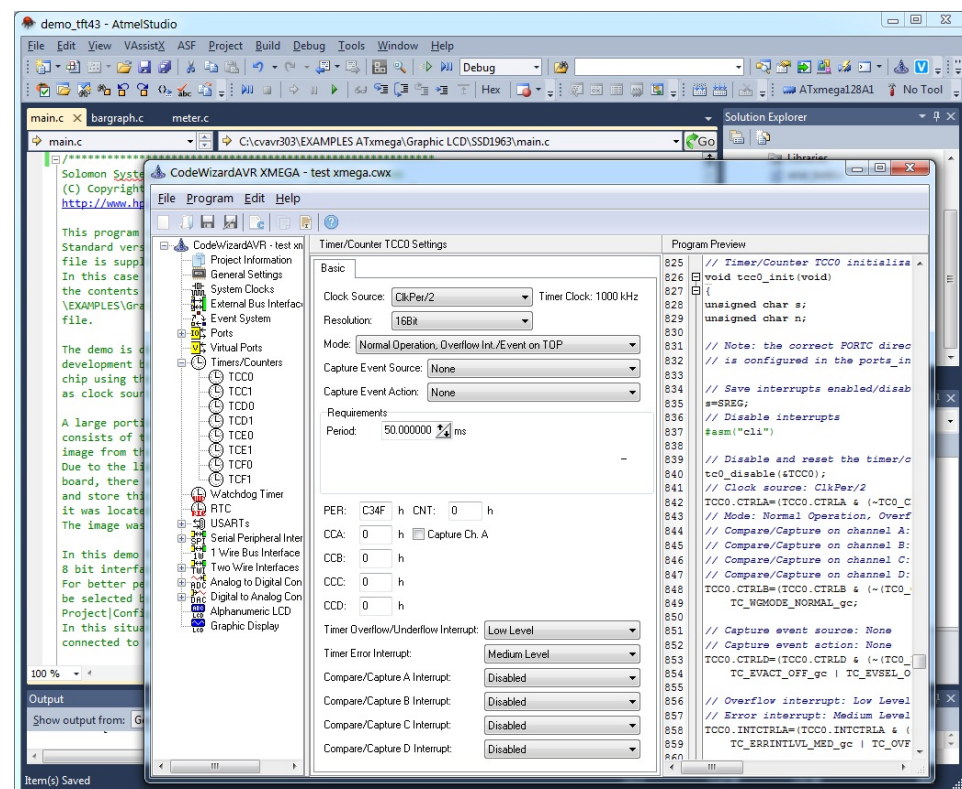
Integrated Development Environment for the 8-bit Atmel AVR, AVR8X and XMEGA Microcontrollers

Features Libraries CodeWizardAVR LCD Vision Chip Programmer Arduino Terminal Supported Chips Revision History Download

- Application that runs under Windows® XP, Vista, Windows 7, Windows 8 and Windows 10, 32-bit and 64-bit
- 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



- Besides its own IDE, CodeVisionAVR can be also used as an Extension integrated in [Atmel Studio 7](#).



- Supported data types: bit, bool, char, int, short, long, 64-bit long, float
- Fast floating point library with hardware multiplier and enhanced core instructions support for all the new ATmega chips
- AVR specific extensions for:
 - Transparent, easy accessing of the EEPROM & FLASH memory areas, **without the need of special functions like in other AVR compilers**
 - 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, ATmega165/169/325/3250/329/3290/645/6450/649/6490, 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 literal 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) in [Atmel Studio 7](#) and [AVR Studio 4.19](#) debuggers
- Fully compatible with Atmel's In-Circuit Emulators: AVR JTAG-ICE, AVR Dragon, etc.