

# IAR Embedded Workbench® for 32-bit AVR

*IAR Embedded Workbench® is a set of highly sophisticated and easy-to-use development tools for embedded applications. It integrates the IAR C/C++ Compiler™, assembler, linker, librarian, text editor, project manager, and C-SPY® Debugger in an integrated development environment (IDE). With its built-in chip-specific code optimizer, IAR Embedded Workbench generates very efficient and reliable FLASH/PROMable code for the 32-bit AVR microprocessors. In addition to this solid technology, IAR Systems also provides professional worldwide technical support.*

## MODULAR AND EXTENSIBLE IDE

- Powerful project management allowing multiple projects in one workspace
- Hierarchical project representation
- Smart source browser
- Feature-rich editor with code templates and multi-byte support
- Tool options configurable on global, group of source files, or individual source files level
- Flexible project building via batch build, pre/post-build or custom build with access to external tools in the build process.
- Build integration with IAR visualSTATE
- Integration with Subversion and other source code control systems
- Extensive device support with ready-made header files, device description files and linker command files
- Ready-made code and project examples for the Atmel evaluation boards

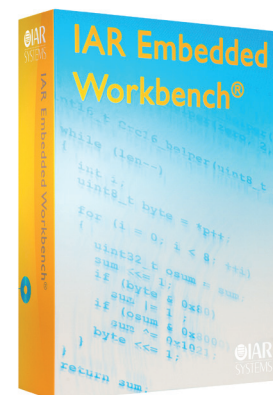
## HIGHLY OPTIMIZING C/C++ COMPILER

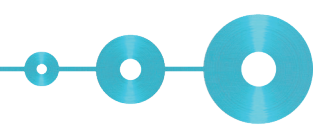
- Support for C, Embedded C++ and Extended Embedded C++
- Compliance with the ISO C99 standard
- Automatic checking of MISRA C rules
- Support for all devices in the UC3A, UC3B, UC3C, UC3D and UC3L families (up-to-date device support list available at <http://www.iar.com/ewavr32>)
- Language extensions for embedded applications with target-specific support
  - Extended keywords for data/functions defining and declaring with memory/type attributers

- Pragma directives for controlling compiler's behavior, such as how it allocates memory
- Intrinsic functions for direct access in C source to low-level processor operations
- 32- and 64-bit floating-point types in standard IEEE format
- Multiple levels of optimizations on code size and execution speed
- Advanced global and target-specific optimizer
- Parallel datapath recombination to utilize simd instructions
- Support for ETSI standard for fixed point arithmetic
- Support for interrupt and exception handling in C/C++
- Atmel ABI compliant
- Multi-file compilation support
- Support for Atmel FlashVault technology

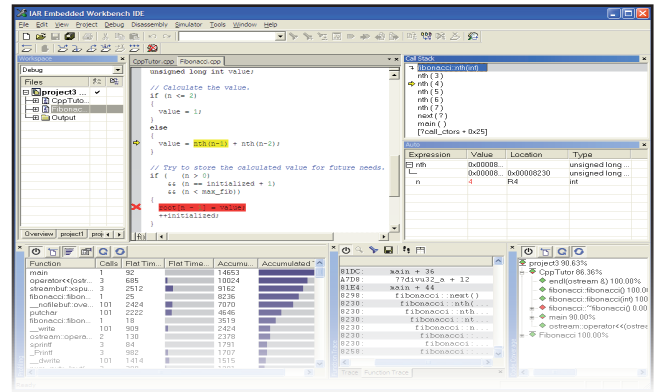
## STATE-OF-THE-ART C-SPY® DEBUGGER

- Support for debugging multiple images
- Complex code and data breakpoints
- Very fine granularity execution control (function call-level stepping)
- Stack window to monitor the memory consumption and integrity of the stack
- Timeline window allows correlated visualization of call stack and interrupt log plotted against time
- Complete support for stack unwinding even at high optimization levels
- Profiling and code coverage performance analysis tools
- Trace simulation utility with expressions to examine execution history
- Versatile monitoring of registers, structures, call chain, locals, global variables and peripheral registers
- Smart STL container display in Watch window
- Symbolic memory window and static watch window





- Extensive instruction set simulation
  - MMU & segment handling
  - Exceptions
  - Interrupt simulation
  - Trace support
  - Macro system to simulate external actions, file I/O, peripherals etc.
- True editing-while-debugging
- RTOS-aware debugging with built in plugins for
  - OSEK Run Time Interface (ORTI)
  - Express Logic ThreadX



## HARDWARE DEBUGGER SUPPORT

- JTAGICE mkII
  - NanoTrace support
  - aWire debug support
- JTAGICE3
  - NanoTrace support
  - JTAG support
  - aWire debug support
- AVR ONE!
  - NanoTrace support
  - aWire debug support
  - Buffered Auxillary Trace
  - Streaming Auxillary Trace with Flash loader, Hardware and software breakpoints, and USB and serial connection

## IAR ASSEMBLER

- A powerful relocating macro assembler with a versatile set of directives and operators
- Built-in C language preprocessor, accepting all C macro definitions

## IAR XLINK LINKER

- Complete linking, relocation and format generation to produce FLASH/PROMable code
- Flexible segment commands allowing detailed control of code and data placement
- Optimized linking removing unused code and data
- Direct linking of raw binary images, for instance multimedia files
- Optional code checksum generation for runtime checking

## IAR visualSTATE®

- IAR visualSTATE is a suite of graphical design automation tools for embedded systems.
  - Design an embedded application by drawing objects, events, actions etc in a flowchart-like manner
  - Perform extensive tests before committing to hardware: validation of the application behavior, regression testing, verification of the run-time model and simulation on-chip

- Comprehensive cross-reference and dependency memory maps
- Support for over 30 industry-standard output formats including ELF/DWARF where applicable, compatible with most popular debuggers and emulators
- Automatic selection of smallest printf/scanf formatter

## IAR LIBRARY AND LIBRARY TOOLS

- All required ISO/ANSI C and C++ libraries and source included
- All low-level routines such as writechar and readchar provided in full source code
- Lightweight runtime library, user-configurable to match the needs of the application; full source included
- Library tools for creating and maintaining library projects, libraries and library modules
- Listings of entry points and symbolic information

## COMPREHENSIVE DOCUMENTATION

- Efficient coding hints for embedded application
- Extensive step-by-step tutorials
- Context sensitive help and hypertext versions of the user documentation available online
- User guides in PDF format

## INFORMATION CENTER

Web-based navigation system that gives easy access to tutorials, product documentation, and example projects.

## FREE EVALUATION SOFTWARE

Free evaluation softwares—32KB KickStart and 30-day evaluation versions are available at <http://www.iar.com/ewavr32>

- Automatically generate micro-tight C/C++ code that is 100% consistent with your design as well as complete design documentation

Together with IAR Embedded Workbench, IAR visualSTATE forms a complete set of development tools for the AVR32 microprocessors, supporting you through the entire development process.

[www.iar.com](http://www.iar.com)

IAR Systems, IAR Embedded Workbench, C-SPY, visualSTATE, The Code to Success, IAR KickStart Kit, IAR and the logotype of IAR Systems are trademarks or registered trademarks owned by IAR Systems AB. J-Link and J-Trace are trademarks licensed to IAR Systems AB. All other trademarks or registered trademarks mentioned in this document are the property of their respective owners and no rights are claimed for these. ©Copyright 2006-2011 owned by IAR Systems.