IAR Embedded Workbench Version 4.41 for ARM®

C/C++ Compiler and Debugger Tools for ARM

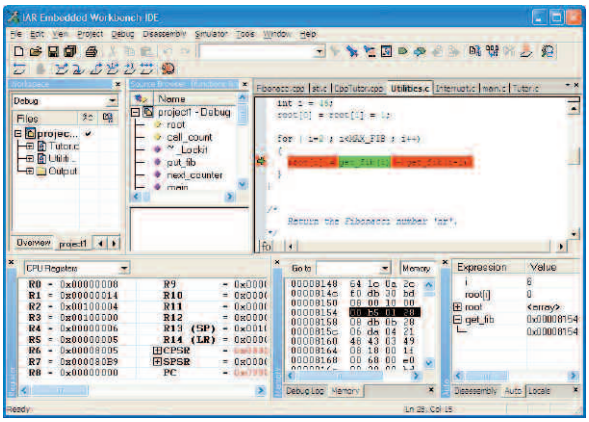
IAR Embedded Workbench® is a set of highly sophisticated and easy-to-use development tools for programming ARM® embedded applications. It integrates the IAR C/C++ compiler, assembler, linker, librarian, text editor, project manager, and C-SPY® debugger in one Integrated Development Environment (IDE). With its built-in chip-specific code optimizer, IAR Embedded Workbench generates very efficient and reliable FLASH/PROMable code for ARM devices. In addition to this solid technology, IAR Systems provides professional, worldwide technical support.

Supported Devices

- ARM7 (ARM7TDMI, ARM7TDMI-S, and ARM720T)
- ARM9 (ARM9TDMI, ARM922T, and ARM940T)
- ARM9E (ARM926EJ-S, ARM946E-S, and ARM966E-S)
- ARM11
- Cortex-M3 (the ARM Cortex-M3 processor offers significant benefits to system and software developers)

Key Components

- IDE with project management tools and editor
- Highly-optimizing ARM compiler supporting C and C++
- Configuration files for ARM chips from Analog Devices, Atmel, Freescale, Intel, Luminary Micro, NXP, STMicroelectronics, and Texas Instruments
- Extensive JTAG and RDI debugger support
- Optional IAR J-Link and IAR J-Trace hardware debug probes
- Run-time libraries including source code
- Relocating ARM assembler
- Linker and librarian tools
- C-SPY debugger with ARM simulator, JTAG support, and support for RTOS-aware debugging on hardware
- Evaluation edition of IAR PowerPac RTOS and file system bundle
- RTOS plug-ins available from IAR Systems and RTOS vendors
- Code templates for commonly used code constructs
- Sample projects for evaluation boards from many different manufacturers
- User and reference guides, both printed and in PDF format
- Context-sensitive online help



Highlights in the Current Version

- IAR PowerPac bundled evaluation edition of RTOS and file system for ARM
- Live watch on target hardware
- Code coverage using IAR J-Trace
- Comprehensive Flash loader support
- I/O register definition files
- More than 400 sample projects for different evaluation boards

Supported ARM Cores and Devices

IAR Embedded Workbench supports ARM7, ARM9, ARM9E, ARM11, and Cortex-M3 devices from these manufacturers:

- Analog Devices
- Atmel
- Freescale Semiconductor
- Intel
- Luminary Micro
- NXP
- STMicroelectronics
- Texas Instruments

For more information on Arrow's Development Tools, pricing and availability, **visit www.arrowdevtools.com** or call **1-866-910-3650**.

Chip-Specific Support ▸

- Ready-made C/C++ and assembler peripheral register definition files
- Multiple code and data models (where applicable)
- Extensive set of language features for PROMable embedded code, including memory keywords, intrinsic functions, interrupt functions, memory-mapped I/O ports, etc.
- Sample projects for evaluation boards from IAR Systems, Analog Devices, ARM, Atmel, Freescale, Keil, LogicPD, Luminary Micro, Nohau, NXP, Phytec, STMicroelectronics, and Texas Instruments
- Support for 4 GB applications in ARM and Thumb® mode
- Each function can be compiled in ARM or Thumb mode
- Vector Floating Point (VFP) coprocessor code generation
- Flash loaders included for devices from Analog Devices, Atmel, Freescale, Luminary Micro, NXP, STMicroelectronics, and Texas Instruments
- ARM Angel debug monitor support

Embedded Focus ▸

- Advanced generic and processor-specific optimizations for speed and memory footprint
- Lightweight runtime library: user-configurable to match the needs of the application; full source included
- Flexible memory handling allows detailed control of code and data placement
- Unnecessary functions and variables are removed
- Application-wide type checking of C/C++ variables and functions at link time
- Optional flexible checksum generation for image runtime verification
- Automatic placement of code and data in non-contiguous memory regions
- Powerful relocating macro assembler with a versatile set of directives and operators

Embedded Debugging ▸

- Fully integrated debugger for source and disassembly level debugging
- Very fine granularity execution control (function call-level stepping)
- Complex code and data breakpoints
- Versatile monitoring of data: locals, watch, auto, live watch, and quick watch windows; register and memory windows
- STL container awareness
- C/C++ call stack window that also shows the function to be entered; double-click on any function in call chain updates the editor, locals, register, watch, and disassembly windows to display the state of that particular function at the time of call
- Trace utility to examine execution history: moving around in the trace window updates the editor and disassembly windows to show the appropriate location
- Terminal I/O emulation
- Interrupt and I/O simulation
- C-like macro system to extend debugger functionality
- Application program system calls emulated by the host
- Code coverage and profiling performance analysis tools
- Support for the ARM Debug Communication Channel (DCC)
- Generic Flash loader with API guide
- Multiple Flash loaders supported
- Debugger software development kit for third-party extensions such as real-time operating systems and emulator drivers
- Command line debugger utility

Graphical IDE ▸

- Hierarchical project presentation
- Multiple projects within the same workspace
- Dockable windows and multiple views
- Source browser
- Library tools included for creating and maintaining libraries
- Integration with source code control systems
- Text editor with multi-byte character support: context-sensitive help system; syntax coloring; unlimited undo/redo; find; search; replace; incremental search; bookmarks; error tags; previous/next navigation; matching brackets; smart indentation; code breakpoint set/clear/enable/disable; and multiple panes
- Command line build utility

Language and Standards ▸

- The C programming language, as standardized by ISO/ANSI C94, with selected features from C99
- Embedded C++ extended with templates, multiple and virtual inheritance, namespaces, and other C++ features that do not cause an overhead in size or speed; full Embedded C++ library containing string, streams, etc., as well as the Standard Template Library (STL)
- IEEE-754 floating-point arithmetic
- MISRA C checker
- Supports a wide range of industry-standard debug and image formats: compatible with most popular debuggers and emulators, including ELF/DWARF where applicable

User Assistance ▸

- Ready-made sample projects and project templates
- Context-sensitive online help with library function lookup
- Printed user guides with extensive step-by-step tutorials
- User friendly, detailed, and precise error messages and warnings