



CROSSWARE[®]

Tools for Embedded Development

Development Suite for ARM[®]

Compiler Wizards Simulator Debugger Environment Variants

The Crossware Development Suite for ARM is a sophisticated environment that facilitates the rapid development of software for microcontrollers that use the ARM core. It includes:

- An advanced optimizing Embedded C++ and ANSI C compiler and libraries
- Code creation wizards to create configuration code, outline interrupt handlers for on-chip peripherals and peripheral specific application code
- A source level simulator to simulate the ARM core and the on-chip peripherals allowing debugging without hardware
- Interfaces that allows the simulator to be extended, facilitating complete target system simulation
- A source level debugger to allow debugging on the target hardware
- A USB JTAG interface to communicate between the debugger and the target hardware
- A multi-threaded, multi-target integrated development environment in which to write, edit and debug code, browse high level source code, access electronic copies of the manuals and much more



It inherits a set of user-friendly features that have evolved over a period of 10 years, or 20 years in the case of the compilation tool chain.

Everything has been developed entirely by Crossware engineers, is supported by Crossware engineers and continues to be enhanced by Crossware engineers in response to customer requirements.

C/C++ Compiler

- Highly optimizing Embedded C++ and ANSI C compiler with extensions for embedded development
- Pre-written library routines including 32-bit and 64-bit floating point arithmetic
- Support for 64-bit integer arithmetic
- Multiple function calling conventions
- Function inlining
- Advanced optimizations using program and data flow analysis
- Comprehensive source level debug output
- Data output for Embedded Development Studio source code browser
- In-line assembler
- Easy location of time-critical functions in high-speed RAM
- Generates code for the microprocessors and microcontrollers based on the ARM7, ARM9, Cortex-M3 and Cortex-M4 cores

Code Creation Wizards

The Code Creation Wizards simplify and accelerate the creation of C source code for the on-chip peripherals of an ARM variant.

The Wizards are able to create configuration code to configure the peripheral, application code to operate the peripheral, code to configure the interrupt controller for the peripheral and outline interrupt handlers for the peripheral.

The Wizards take two forms:

- GUI Wizards which present a point and click graphical interface to the user
- Context Wizards which present a point and click interface to a list of named registers and named bits

Context Wizards are provided for all on-chip peripherals.

ARM Development Suite Details

Compiler

Wizards

Simulator

Debugger

Environment

Variants

Simulator/Debugger

The simulator creates a virtual ARM chip that runs on your PC. It allows ARM programs to be run without any hardware.

The debugger downloads and runs your program on an ARM target board.

The simulator and debugger share a common user interface.

Both the simulator and debugger provide:

- Source and assembly level debugging
- Multiple memory views and watch windows
- Multiple register views with register tooltips
- Disassembly view
- Complex hierarchical source level drag-and-drop breakpoints
- Machine level breakpoints
- Call stack view
- Variable tooltips
- Multiple application debugging

The simulator provides:

- Simulation of the ARM instruction set
- Simulation of the ARM9 memory management unit (MMU) and fast context switch extension (FCSE)
- Simulation of on-chip peripherals
- Interfaces to allow the simulator to be extended
- State capture
- Source code profiling
- Code and data coverage
- Multiple cycle counters

The debugger provides:

- Downloading into RAM and on-chip flash memory
- Downloading into external flash memory
- Hardware breakpoints in flash and ROM
- Unlimited software execution breakpoints in RAM and flash memory
- Debugging with the ARM9 memory management unit (MMU) and fast context switch extension (FCSE) enabled

Debugger Interface

The debugger requires a connection to the target board and this is provided by the *Jaguar* USB JTAG interface.



Embedded Development Studio IDE

The Embedded Development Studio is the integrated development environment application at the heart of the ARM Development Suite.

It links together the simulator, debugger, code creation wizards and extension DLLs and also drives the compiler tool chain. (Note that the compiler tool chain can also be driven from the command line.)

The Embedded Development Studio also provides other facilities including:

- Creates new projects with an initial set of default files
- Allows the organisation of multiple projects into workspaces
- Context coloured editing of compiler and assembler source code
- Context coloured views of compiler/assembler listings
- Drag-and-drop linker scripting
- Scans the project source code to generate dependency information
- Uses the browse records generated by the compiler to accelerate source code navigation
- Text file searching
- Find and Find and Replace in source views
- Provides access to the electronic manuals including full search capability
- Automatic generation of a build script which can be used to build your project from the command-line or from within another application.
- Integrated terminal emulator facilities
- Customisable commands allowing the launch of other executables
- Generation of GUIDs
- Conversion of bitmaps to C structures
- Supports multiple Crossware suites and tool chains and third party tool chains

Variant Support

For details of support for specific ARM variants, please check the Crossware website.

Licensing

The Jaguar debugger interface also serves as a licence key. For the Professional edition, a separate licence key is provided.

System Requirements

Runs on Windows Windows XP SP3 and above.

For further information contact:

Crossware, Old Post House, Silver Street, Litlington, SG8 0QE, UK
Tel: +44 (0) 1763 853500, Fax: +44 (0) 1763 853330