Software Development Tools

CodeWarrior® Development Studio

for Power Architecture™ Processors V8.8, Linux Platform Edition (Linux-hosted)

and Linux Application Edition (Linux-hosted)

Overview

CodeWarrior® Development Studio for Power Architecture™ Processors, Linux Platform Edition, version 8.8 is a comprehensive solution designed to accelerate embedded Linux development on a variety of Freescale Power Architecture processors. The Platform Edition tools provide the features you need, from board bringup capabilities through development of kernel, driver, stacks and applications, when building complex embedded devices. The award-winning CodeWarrior® integrated development environment (IDE) provides a high-quality, full-featured development environment; it comes as a complete IDE designed for each stage of the development process – from board bring-up through embedded application development. State-of-the-art debugging technology, the simplicity of an intuitive development environment and robust run-control raise hardware board bring-up and C/C++ embedded application development to a new level. In addition, developers remain productive by using a consistent development environment across all supported workstations and Freescale silicon platforms. Just as importantly, as an integral part of Freescale, our unique knowledge and early access to Freescale silicon and designers, gives customers comfort knowing that they can rely on CodeWarrior tools for early support of new Freescale processors and customer support required for the rigors of today's shorter project life cycles.

Specifically, the CodeWarrior Development Studio, Linux Platform Edition gives you the ability to deploy and debug a Linux kernel, in addition to creating Linux applications. You can simultaneously debug an application process while using a JTAG connection to debug the kernel — all with source code views and visibility into the memory translations associated with the Linux OS. The Linux Platform Edition tools offer you the ability to debug existing ELF symbolics compiled prior to using CodeWarrior tools and to automatically configure all the source browsing information for immediate source code debugging of an existing package. Furthermore, for those migrating from other RTOS solutions, a familiar project manager is provided; this avoids the need to understand gmake — further accelerating developers into embedded Linux application development.

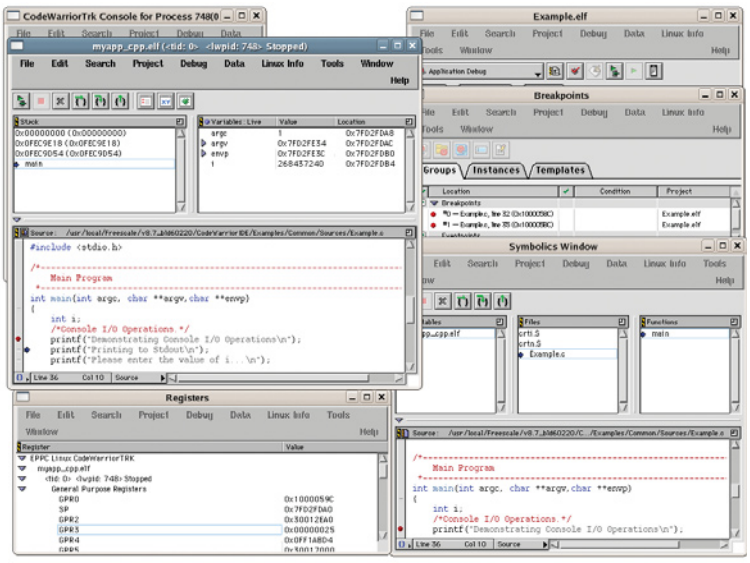
New Processors and Platforms Supported by Version 8.8, Linux Platform Edition

- MPC8313-RDB
- MPC8315-RDB
- MPC8323E-RDB
- MPC8349E-mITXE
- MPC8360-MDS-PB 2.1 Si
- MPC8360-RDK 2.1 Si
- MPC837x-MDS/RDB
- MPC8544DS 1.1 Si
- MPC8555CDS
- MPC8548CDS 2.1 Si
- MPC8568E-MDS 1.0/1.1 Si
- PPCEVAL-DS-8572 1.0 Si
- PCEVALHCPD-8610E
- MCEVALHPCN-8641D

Highlights

Control

- Concurrently debugging multiple applications, process and threads
- Create, load and debug Linux kernel modules
- Support for Uboot (and other boot loaders) debugging
- Source files can be modified while debugging (original is cached)
- Browser support in the debugger
- Integrated hardware diagnostics for board level testing



- Integrated source-code navigation provides quick and easy access to functions and files

freescale semiconductor

- Highly efficient C/C++ compiler with cutting-edge optimization technology for fast, compact code
  - Complete control of code and data memory allocation
  - Options to pack or byte-swap structures to match existing data types
  - Supports position independent code (PIC) and position independent data (PID)

Flexibility

- Show the state of individual kernel threads
- Source files can be modified while debugging
- Both 2.6.x and 2.4.x Linux kernels are supported
- Examples available showing debugging applications and kernel modules

Visibility — Graphical Views of:

- "Always on" stack crawl and local variables
- Breakpoints—hardware and software
- Advanced Process Management and Display — View Linux target processes including those at the kernel level, and individual process parameters such as status, command line executable, environment variables and memory maps.

Product Features

Board Bring-Up

The CodeWarrior debugger provides complete control over all board settings, including initializing and saving register values and memory configuration. The debugger also includes a comprehensive set of hardware diagnostics and robust flash programming algorithms supporting an extensive list of flash devices.

Hardware Diagnostics

The CodeWarrior Development Studio comes with diagnostics tests that determine if the basic hardware is functional. These tests include:

- Memory read/write: Perform diagnostic tests by writing and reading memory through a connecting probe
- Scope loop: Repeat memory reads and writes through a connecting probe
- Memory tests: Perform different tests on the hardware including:
  - Walking ones test
  - Address test
  - Bus noise test

Specify any combination of the tests and the number of passes. Results are displayed in a log window after all passes are complete.

Flash Programming

Program on-chip and off-chip flash devices from within the same graphical user interface used to troubleshoot the application. No boot code is required to run on the target system in order to use the programming features of the CodeWarrior flash programmer. Support is provided for hundreds of popular flash memory devices.

Search Engine and Text Editor

Fast, semantic code navigation makes it possible to find specific code structures among hundreds of directories and files quickly and easily. Seamless integration between the CodeWarrior search engine and the text editor allows code changes to reflect immediately in the browser without recompiling.

CodeWarrior Debugger

The CodeWarrior debugger packs a wide array of high-powered features designed to help the developer find and repair software defects quickly, while providing maximum control to debug complex hardware and software issues including:

- User-configurable workspaces: Focus on complex debugging tasks, each workspace contains just the set of views needed for the task at hand
- Hardware and unlimited software breakpoints
- Eventpoints: Perform a specified task when the program executes a specific line of source code or when an associated conditional expression evaluates to true. Set an eventpoint to perform a task (i.e., run a script, play a sound or collect trace data) for superior control over your code execution. Eventpoints can be:
  - Log Point–Logs a string or expression to a file and records messages to the Log window
  - Pause Point–Pauses execution to refresh debugger data–great for watching a variable change over time
  - Script Point–Runs a script or application
  - Skip Point–Skips execution of a line of source code
- Watchpoints: Set a watchpoint at a location in memory to halt program execution when that point in memory changes value or, for some devices, when the memory location is accessed. Then examine the call chain, check register and variable values, and step through your code
- Single-stepping: Step Into, Step Over and Step Out of functions
- Variable view on mouse over: When in the debugger, you can mouse over a variable in the source display and get the current value of that variable
- Display stack trace: The "Call Stack" view provides an easy display of all procedures (functions) active in the calling chain and enables the developer to follow the progress of a program through its hierarchical call structure
- Local variables display: View the variables local to the current function
- Memory view: Display and modify the contents of the target's memory. Quickly find a value in memory, compare memory regions or upload and download memory to a file using this view
- Register view: View extensive information on CPU core and peripheral registers, as well as user-defined registers. The registers displayed can also include bit-level details for an English language equivalent of register contents
- Object file format: The CodeWarrior debugger supports STABS and ELF/DWARF formats
- Mixed language debugging: The CodeWarrior debugger supports mixed language debugging in C, C++ and Assembly language by automatically analyzing the file in view and adjusting the expression evaluation and data display accordingly
- Profile window: Improve the performance of your code by using the built-in profiler to analyze function performance when using the CodeWarrior Compiler
- Command-line window: Use the command-line interface together with various scripting engines, such as the Microsoft® Visual Basic®, the Java, TCL, Python and Perl to automate testing, standardize data-logging or uncover that hard-to-find problem

- Seamless integration with CodeWarrior USB and Ethernet TAPs. The CodeWarrior debugger is fully integrated with these hardware probes, resulting in optimized run-control with fast downloads. A target connection wizard simplifies and automates the task of defining new connection definitions based on hardware and communication parameters

CodeWarrior Build Tools (Compiler, Assembler, Linker)

The CodeWarrior compiler produces exceptionally fast, compact, EABI-compliant object code. Our proprietary optimization techniques enable the programmer to balance execution speed with code size while intelligent defaults can generate optimal code.

Key Features Include

- inclusion of the GNU Compiler Collection (GCC), associated utilities, and core C libraries to provide a full development solution.
- CodeWarrior tools can integrate to GCC compilers and libraries that you have from third parties.
- Standards conformance (ANSI and EABI) for maximum tool interoperability
- Complete control of code and data memory allocation
- Options to pack or byte-swap structures to match existing data types
- Supports position independent code (PIC) and data (PID)
- Board support routines for bare board applications (no OS)

Assembler: full-featured macro assembler that is invoked automatically by the project manager or as a complete standalone assembler for generating object modules

Linker: offers precise control over the allocation, placement and alignment of code and data in memory.

System Requirements

- Hardware PC with 1.4 GHz
- Intel® Pentium®-compatible processor (or better)
- 512 MB RAM (1 GB recommended)
- CD-ROM drive

This CodeWarrior product is tested and supported by Freescale on the following Linux operating system distributions: Red Hat® Enterprise Linux 3.0 and Red Hat Enterprise Linux 4.0

- Free Disk Space 1.6 GB, plus space for projects and source code

Required Host-Target Interfaces

- CodeWarrior USB TAP
  - CWH-UTP-PPCD-HE
  - CWH-UTP-PPCC-HE
- CodeWarrior Ethernet TAP
  - CWH-ETP-JTAG-HE
  - CWH-ETP-DPI-HE

Product Support

- Program provides online help and documentation
- New product purchase includes one-year technical support which can be renewed

Part Numbers

- CWS-PPC-LLPLT-CX – Platform Edition, single seat license
- CWS-PPC-LLAPP-CX – Application Edition, single seat license

Supported Freescale Processors

- 5200, 7448, 8xx, 82xx, 83xx, 85xx and 86xx

Linux Application Edition Features

The CodeWarrior Development Studio, Linux Application Edition offers the following subset of features from the Platform Edition:

- Compiler and linker:
  - Supports GNU compiler collection
  - Ability to integrate alternative GNU compilers
- Application debug
  - Serial connectivity
  - Networking connectivity
- Full multi-threading/multi-tasking application debugging
  - Individually control threads
  - Follow fork/child debugging
  - Full multi-process debug from one IDE
  - Step one thread while all others are stopped
  - Process and thread list

NOTE: For Kernel-level debugging, CodeWarrior Development Studio Linux Platform Edition is required.

Learn More: For current information about Freescale products and documentation, please visit **www.freescale.com**.

950-00080
REV E

freescale semiconductor