



Quickly Turning Ideas into Microcontroller-Based Applications

with CodeWarrior™ Development Tools
and UNIS Processor Expert™



Freescale Semiconductor
Author, Chris Joffrain, Product Manager

Document Number: PROCESSPERTWP
Rev. 0
11/2005





Year after year, time to market is getting shorter and shorter, thus putting pressure on project managers, software and hardware designers and marketing managers to complete projects more quickly. At the same time, the integration of complex features in microcontrollers demands more time for engineers to understand and master the capabilities of these devices. The integration of state-of-the-art tools such as CodeWarrior™ development tools and UNIS Processor Expert™ technology alleviates this problem by dramatically reducing the total design time from the verification of the idea to the validation of a final prototype.

CONTENTS

1. UNIS Processor Expert™	3	8. Fully Integrated Source Code Editor	6
2. CodeWarrior™ IDE	4	9. File/Folder Compare, Graphical View of Differences	6
3. CodeWarrior Development Tools Meet the Needs of MCU-Based Applications Development	4	10. Drag-and-Drop Linker	6
4. Open Plug-In Architecture for Greater Flexibility	4	11. Fully Integrated Debugger	7
5. Project Manager	5	12. Building Projects.....	7
6. Navigate Source Code (Browser)	5	13. Instruction-Set Simulator	8
7. Compiler Optimizes Software Performance and Time to Market	5	14. Stationery Templates.....	8
		15. Conclusion.....	8

Over the last 30 years, microcontrollers, often referred to as microcontroller units (MCUs) have become a pervasive part of our daily lives. Every day, MCUs find their way into new applications, ranging from remote car door controllers in a key ring to smart credit cards, engine control units and automatic braking systems (ABS) control units.

Whether small or large, MCU-based applications development requires the same software engineering process as other architectures. After product management or marketing has defined the requirements for a new product, engineers develop a plan to create—in the given time to market window—the product that will meet the given specifications. The plan, often called a statement of work (SOW), describes all the features the product will have, the resources assigned to the project, the time lines and deliverables, as well as what the team may need and the potential risks in the project. This usually comprehensive document does not always accurately take into account the amount of time that the engineering team needs to fully understand the MCU, as well as the time required to get the best performance out of the MCU chosen for the project. This piece of information, which is often difficult to appreciate and quantify, may end up being the critical determining factor in the success of the project and its timely release to a very competitive market.

Freescale Semiconductor's MCUs are at the heart of countless applications, and to help designers maintain a competitive advantage, CodeWarrior development tools supporting these MCUs are now integrated with Processor Expert, providing engineers with an integrated development environment (IDE) for design, implementation, verification and optimization of embedded microcontroller-based applications. This highly productive combination allows for efficient use of the MCU and its peripherals, enabling building of portable solutions and saving development time and cost. Initially introduced for the 68HC08 family, this combined package will soon also be released for the other microcontroller families such as 68HC12 and others.

1. UNIS Processor Expert™

Processor Expert is the first component-based application builder and system integration tool that provides a unified approach to embedded systems design. This high-productivity development environment offers efficient management of the MCU and its peripherals, allowing the building of portable solutions and reducing the time and costs of development. Processor Expert contains, among its many features, an expert knowledge database of MCU models, which is used during the design phase to check the user inputs against the technical specifications of the selected microcontroller.

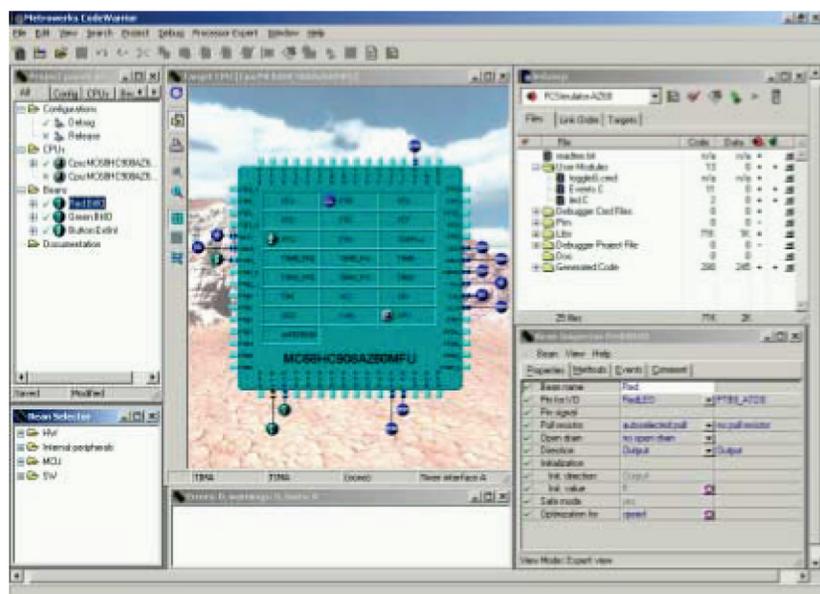
The world of embedded application development often requires a deep knowledge of both hardware and software to determine what portions should be implemented in hardware and what portions should be implemented in software.

With Processor Expert all the work covering the core of the application, the MCU and its behavior becomes a configuration process which can be changed easily at any time. It is a process where changes in the hardware are reflected in the application software configuration, and where changes in peripheral settings are immediately transformed into application source code. The built-in knowledge about the possible hardware settings decreases the learning curve, as well as the number of incorrect decisions.

Processor Expert provides detailed information about MCU resources, from package to peripheral and registers allocation. It also assists in setting up timing details and parametric searches of the right derivative for the application.

The Target MCU view shows current status of all microprocessor pin connections and reflects the configuration process. It contains an image of the microcontroller package and gives information about each pin, connected devices and MCU status. The package can be rotated, so that users can see on screen what they have on the board for easy orientation. The MCU block view provides graphical design of the MCU functionality by direct approach to the peripherals and the definition of the mode in which they are used. Some of the main features offered by Processor Expert are:

- > Choice of the MCU from multiple derivatives list
- > Detailed description of MCU pins
- > Configuration of functions and settings for selected MCU and its peripherals



- > Description of the system during initialization and at runtime
- > Design of blocks in the application from pre-built functional components
- > Design of application using callable functions and events (e.g., interrupts)
- > Hand-packed drivers
- > Library of components for typical functions
- > Code generation for the components used in the project
- > Implementation of user-written code
- > Integration with CodeWarrior editor, compiler, simulator/debugger

2. CodeWarrior IDE

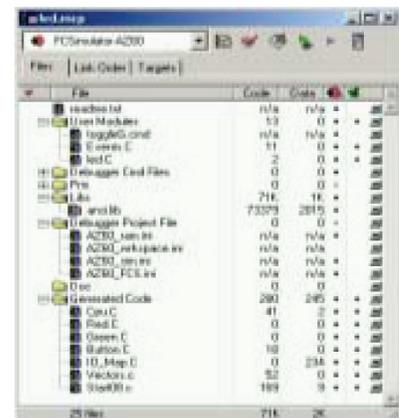
The CodeWarrior integrated development environment (IDE) offers unique features and seamless access to a variety of software development tools, allowing customization of the environment by the user. It combines a powerful source code editor and browser, proven compilers and linkers, a sophisticated project manager, a user-friendly debugger-simulator and a rich set of libraries, all under one roof. The IDE makes it possible for developers to edit, manage, compile, link and debug applications with less effort and better results.

The user can navigate, browse, examine, compile, link, edit and run the code without exiting the IDE, which makes for an easier interaction between tasks, such as tracing into the source code, comparing files and debugging the application using a simulator or a hardware platform. The user can also configure options for code generation, project navigation and other operations.

3. CodeWarrior Development Tools Meet the Needs of MCU-Based Applications Development

The CodeWarrior award-winning development environment integrates all of the following components within a flexible, extensible architecture:

- > Intuitive, state-of-the-art project manager and build system
- > Highly optimized C and EC++/C++ compiler designed to reduce the need for time-consuming assembly language programming
- > EABI-compliant macro assembler
- > Build tools generating/using industry-standard ELF/DWARF object files format
- > Graphical, RTOS-aware debugger to simplify debugging of complex algorithms
- > Cycle-accurate simulation enabling software development ahead of hardware
- > Simple I/O stimulation
- > Visualization tools
- > Integrated burner utility to have deep control over generated S-record, Intel® hex or binary output files
- > Integrated support for hardware evaluation boards



The CodeWarrior IDE goes well beyond code generation and basic debugging. An engineer can launch the CodeWarrior IDE once and stay there throughout the entire development cycle. By providing all of the functionality in one application, CodeWarrior development tools help shield developers from the relative complexity of a new processor, allowing them to focus on innovation and time to market.

4. Open Plug-In Architecture for Greater Flexibility

Developers often use tools in unexpected ways and the various choices that drive development cycles result in unique requirements. In order to accommodate a wide variety of needs and individual preferences the CodeWarrior IDE was designed to be flexible and highly extensible. The environment can be modified in two ways: by replacing an existing component or by adding a new component. Components that can be replaced include the editor, search engine, source code browser, compiler, linker and debugger. Similarly, specialized tools such as version-control software can be

integrated quickly and easily through the use of plug-ins. A plug-in is essentially a software module that provides specific services to the user via the IDE. Plug-ins are typically implemented as dynamically linkable or shared libraries. A CodeWarrior plug-in behaves very much like the plug-in used in today's Web browsers; that is, if the plug-in is available when the application starts it automatically updates the development environment's capabilities. A good example of this plug-in capability is the new addition of Processor Expert.

A plug-in communicates with the IDE through a platform-neutral applications programming interface (API). This enables third-party developers to create fully integrated additions to the CodeWarrior tool set. The CodeWarrior IDE currently recognizes several different types of plug-ins for compilers, linkers, preference panels and version-control software. In addition, there are APIs defined for debugger plug-ins.

5. Project Manager

The CodeWarrior project manager provides a fast, convenient, automatic interface for orchestrating the operation of all tools throughout the development cycle, such as creating projects, opening projects, adding files and saving projects. In addition, a developer can move files in the Project window, mark files for debugging, create nested projects and build targets, and divide the Project window into file groups.

The project manager keeps track of all files used throughout a project. Files can be added, removed or arranged in user-defined groups. Files are organized into functional categories, such as source files, libraries, include files and more. The project manager maintains detailed knowledge of where files are stored and how they relate to each other. For every given build the project manager tracks:

- > All files and libraries
- > Link order
- > Dependencies
- > Compiler and linker settings

The sum of this information is called a build target. A project can contain multiple build targets, and each build target can include different files or specify different compiler/linker settings. For example, a project might contain a simulate target, a debug target and a release target. The project manager also provides time-saving information, such as which files need to be recompiled and the size of text and data files in the project. Finally, the project manager incorporates drag-and-drop functionality to change the linking order, so that compile, make, link or debug can be initiated with a single-click.

6. Navigate Source Code (Browser)

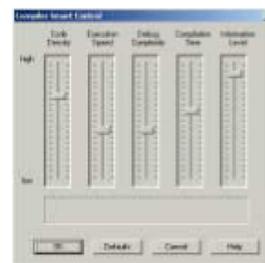
A browser allows the user to sort through a code database generated by a compiler. Historically, programmers have used browsers primarily with object-oriented code, but the CodeWarrior IDE browser works with both procedural and object-oriented code. The browser supports most compilers, including C, C ++, Pascal and Java™.

7. Compiler Optimizes Software Performance and Time to Market

The new MCU market presents a peculiar challenge to developers. Time to market constraints and requirements give developers the challenge to function as high-level system designers, programming system components at a high level while making the best use of limited resources such as memory space.

Programming in a high-level language, such as C and EC++/C++, helps reduce development time and ensure portability. This is especially true when developers are working with a new or unfamiliar architecture. Yet, ultimately, the final application may be optimized to meet real-time requirements. If the C compiler is not optimized for the architecture, it can mean long hours of hand tuning assembly language code.

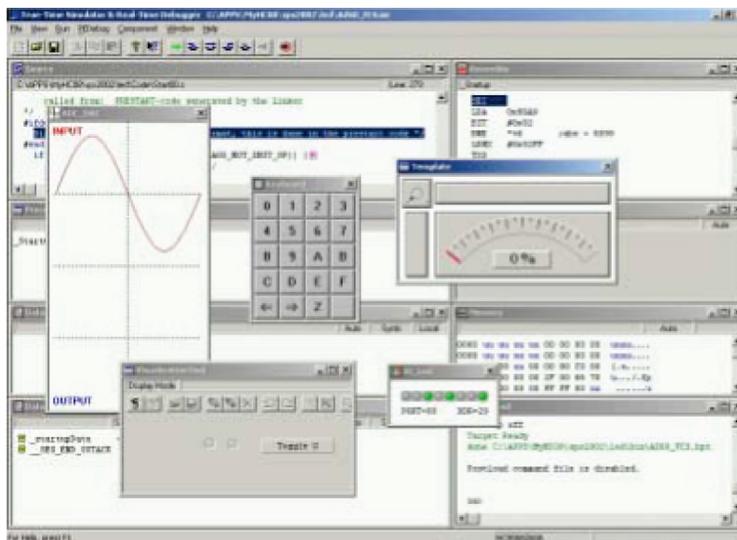
Robust CodeWarrior compiler technology allows developers to work confidently in a high-level language environment and, in many cases, time-critical applications can be coded entirely in standard C. Interactions with the compiler, linker and assembler are simplified through easy-to-use visual dialogs, eliminating the need to remember cryptic flags and commands. The build options are saved within the project for immediate recall. Developers can quickly correct compiler errors by simply double-clicking on the error.



11. Fully Integrated Debugger

The CodeWarrior debugger is an ideal tool for the user to identify errors, remove them and validate the code. It provides commands to execute one or several lines of code at a time, suspend execution when control reaches a specified point or interrupt a program that changes the value of a designated memory location. It displays detailed information to examine and change the values of variables, to inspect the contents of the processor's registers and to open memory windows to display and edit bytes or word locations. When the debugger stops a program, the user can view the chain of function calls. The debugger interfaces either with a simulator or with target boards. The debugger includes:

- > SourceC/C++ and/or Assembler code window
- > Breakpoint setting capability
- > Register windows
- > Processor I/O modules window
- > Software breakpoint window
- > Conditional breakpoints
- > Initialization files
- > Reading and writing memory
- > Visualization tools
- > Command files



12. Building Projects

The CodeWarrior IDE provides many different ways to build projects. Developers can instruct the IDE to compile a single file, multiple/selected files or all files in the project. When an error occurs during compilation, the Errors and Warnings window shows the error message in the Message List pane. The location in the source code that the message refers to is displayed in the Source Code pane. This allows developers to navigate to a spot in the source code pane, then inspect and correct the offending code.

The multitarget capability of the CodeWarrior IDE gives users the ability to develop code on simulator and hardware at the same time (phases of projects) and also the flexibility to migrate projects from one processor to another within a family to support new generations of products. Simple commands will allow developers to build multiple targets with no more effort than it takes to build a simple target.

The CodeWarrior IDE has commands to pre-process, pre-compile, compile, update, link, run, disassemble and check syntax. It uses the settings in the build targets within a project to determine how to process files and data and automatically selects the compilers and linkers to operate on the appropriate files when a command to compile or link is issued. No complex make files are required. However, for those who need specific control of how a project is built, the CodeWarrior IDE accepts files of command lines, even imported make files from other builds.

The CodeWarrior IDE has options to suit any programming need, such as customizing menus, key bindings and toolbars, or controlling the appearance of specific menu commands; creating new command groups in which to place menu commands; and associating a command line with a new menu command. It supports scripting via Perl scripts, Microsoft® Visual Basic® scripts, batch files (Windows®) and AppleScript® scripts (Macintosh®). When scripting the IDE, repetitive, time-consuming or complex tasks can be automated.

13. Instruction-Set Simulator

Time to market pressures do not allow software developers the option of waiting for final working hardware before evaluating application code. Serious developers are asking for an efficient simulation tool as a means to create production-quality code ahead of hardware availability, especially with the combined use of the project manager and the multitarget capability of the IDE. The CodeWarrior IDE offers a fully functional, simulated 68HC08 target environment for loading, executing and debugging applications. This includes accessing memory, setting breakpoints and single-stepping through code. The simulator provides a variety of high-performance features, including:

- > Complete instruction-set simulation of the 68HC08 core
- > Tracking of cycles and instructions simulation
- > Access to registers and memory with display in various formats
- > I/O stimulation/visualization to provide data input and output interface to the application

The simulator can be used to quickly and accurately model a variety of design parameters that often drive cost-versus-performance decisions. With CodeWarrior development tools, the software team can start developing production-level code well ahead of hardware availability. Critical algorithms can be modeled, analyzed and optimized to ensure that performance requirements are met, thereby reducing project risk. Furthermore, the simulator, when used as one of the targets in the project manager, allows developers to reduce the debugging time of the hardware platform by using the working simulated code as a reference.

14. Stationery Templates

Traditional application programming requires tedious manual setup work. When a new application is created, developers are responsible for coding every aspect of the program. They must specify all initialization routines, the interactions between the various pieces of the program and the user interface. Stationery templates in CodeWarrior development tools offer relief from these repetitive processes so that developers can concentrate their efforts on the application. The templates contain code that handles generic initialization routines for the various memory and peripheral modules of specific processors. In addition, the code can be modified and new templates can be created to suit any application's specific needs.

15. Conclusion

The costs of development using microcontrollers are increasingly affected by the software content of the development projects and the complexity of modern microcontrollers. Part of these costs is often reflected in the learning curve associated with new development tools. This additional cost is important enough for developers to reject the most suitable microcontroller for an application because it would take too much effort and time to learn a new set of tools. Software engineers need a high-performance development environment designed to use all of the capabilities of the microcontroller architecture while simultaneously minimizing the underlying complexity. It is critical that microcontroller software development tools provide the same level of ease-of-use and extensibility as mainstream microprocessor tools in order to minimize the new product introduction cycle. CodeWarrior development tools, with Processor Expert, provide an integrated suite of development tools designed to work together to dramatically simplify applications software development.

CodeWarrior development tools and Processor Expert take users to a new level of microcontroller-based development. Additionally, CodeWarriorU classes are available at no cost on the Web for developer teams. These classes act as supplemental tools, offered by Freescale to help developers learn, from their desks and at their own pace, how to make the best use of CodeWarrior tools.

How to Reach Us:**Home Page:**

www.freescale.com

e-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor

Technical Information Center, CH370

1300 N. Alma School Road

Chandler, Arizona 85224

1-800-521-6274

480-768-2130

support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH

Technical Information Center

Schatzbogen 7

81829 Muenchen, Germany

+44 1296 380 456 (English)

+46 8 52200080 (English)

+49 89 92103 559 (German)

+33 1 69 35 48 48 (French)

support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.

Headquarters

ARCO Tower 15F

1-8-1, Shimo-Meguro, Meguro-ku,

Tokyo 153-0064, Japan

0120 191014

+81 3 5437 9125

support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.

Technical Information Center

2 Dai King Street

Tai Po Industrial Estate,

Tai Po, N.T., Hong Kong

+800 2666 8080

support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor

Literature Distribution Center

P.O. Box 5405

Denver, Colorado 80217

1-800-441-2447

303-675-2140

Fax: 303-675-2150

LDCForFreescaleSemiconductor

@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright license granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.