



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Freescale In-Circuit Emulator Base

User Manual

Rev. 1.1
1/2005



How to Reach Us:

USA/Europe/Locations Not Listed:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

Japan:

Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

Home Page:

www.freescale.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 licenses 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 that 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.

Learn More: For more information about Freescale products, please visit www.freescale.com.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005. All rights reserved.

Table of Contents

1	Overview	11
	System Requirements	11
	System Features	12
	System Components	13
	Basic Components	13
	Additional Components	14
2	Getting Started	15
	Setting Up The Development System	15
	Setting Up The Hardware	15
	Installing CodeWarrior Software	17
	Creating a Project	20
	Adding Files	26
	Building (Compiling) a Project	27
	Establishing Communication	28
	Establishing Communication Through Ethernet Port	28
	Establishing Communication Through USB Port	29
	Starting the Debugger	30
3	Configuring The System	35
	Setting Up The System	35
	Specifying A Target	35
	Specifying Communication Information	36
	Assigning An IP Address To The FSICEBASE	38
	Specifying a Memory Map	40
	Specifying the Clock Speed	42
	Loading Program In Debugger	44
	Setting Up Logic Cables And Connectors	44
	Modifying Values In Memory	46
	Using Log Files	47

Table of Contents

4	Using The MON08 Debug Port	49
	Setting Up Hardware For the MON08 debug port	49
	Establishing Communication Through The MON08 Debug Port.	50
5	Using the Debugger	55
	Starting The Debugger	55
	Changing CPU Register Values.	56
	Using the Register Window to Change Register Values	56
	Using the Command Line to Change Register Values.	57
	Viewing Memory.	57
	Resetting The Emulation System	59
	Using Breakpoints And Watchpoints	60
	Setting Breakpoints on Lines of Code.	61
	Setting Breakpoints on Addresses	63
	Controlling Program Execution	65
6	Using the Bus State Analyzer	67
	Overview	67
	Using the Bus State Analyzer	68
	Defining Events	68
	Recording Modes.	70
	Continuous: all cycles	70
	Continuous: events only.	70
	Counted: all cycles.	70
	Counted: events only	70
	Sequential: A+B+C+D.	71
	Sequential: A+B -> C+D	71
	Sequential: A -> B -> C !D	71
	Sequential: A -> B -> C -> D.	71
	Sequential: Nth event: A+B+C+D.	71
	Time Tag Clock Frequency	72
	Collecting Bus Data.	72
	Viewing Data	73

A	S-Record Format	75
	S-Record Content	75
	S-Record Types	76
	Creating S-records	77
	Example	77
B	FSICEBASE Debugger Scripting Commands	79
	Command Reference	79
	Index	81



Table of Contents

About This Book

This manual is for developers, testers, application engineers, and anyone interested in using the Freescale In-Circuit Emulator Base (FSICEBASE) development system.

Organization

- **Introduction**

An overview of the Freescale In-Circuit Emulator Base (FSICEBASE) and development system.

- **Getting Started**

Explains basic procedures to get started using the FSICEBASE and development system.

- **Configuring The System**

Explains how to configure the software and hardware.

- **Using The MON08 Debug Port**

Explains how to use the MON08 debug port of the FSICEBASE.

- **Using the Debugger**

Explains how to use the debugger to execute and analyze code. Explains how to perform debugging tasks such as stepping through code, changing values in registers and memory, and viewing data in memory.

- **Using The Bus State Analyzer**

Explains how to use the Bus State Analyzer (BSA). Explains how to define events, capture data, and view captured data.

- **Appendix A: S-Record Format**

Describes the format of the S-Record file.

- **Appendix B: FSICEBASE Debugger Scripting Commands**

Lists commands only used with FSICEBASE.

Revision History

The following table summarizes revisions to this document.

Revision History

Location	Revision
Metrowerks, Austin, TX	Rev. 1.0 (Initial publication)
Metrowerks, Austin, TX	Rev. 1.1 (Content update)

Suggested Reading

Application Note: Getting Started with HCS08 and CodeWarrior Using C, AN2616, 11/2003.

Motorola HC08 CPU Awareness and True-Time Simulation, Metrowerks, 2004.

CodeWarrior™ Development Studio IDE 5.5 User's Guide, Metrowerks, 2004.

Conventions

This document uses the following notational conventions:

- Text in `Courier` monospaced type indicates commands, command parameters, code examples, expressions, datatypes, and directives.
- Text in *italic* type indicates replaceable command parameters.
- Text in **bold** type indicates names of labels and menu items on-screen.

Definitions, Acronyms, and Abbreviations

The following list defines the abbreviations used in this document.

BSA	Bus State Analyzer
EM	Emulator Module
FSICEBASE	Freescale In-Circuit Emulator Base
LAN	Local Area Network

MCU

Microcontroller Unit

MON08

Monitor for HC08 MCUs



Overview

This manual contains information about how to use the Freescale In-Circuit Emulator Base (FSICEBASE). The FSICEBASE is a tool that helps you develop applications for embedded systems based on MC68HC08 microcontroller units (MCU).

NOTE The Freescale In-Circuit Emulator Base (FSICEBASE) is similar to a system formerly produced by Motorola called the MMDS05/08 Motorola Modular Development System (MMDS). If you worked with the MMDS05/08, many of the features of the FSICEBASE will be familiar to you.

This chapter contains the following sections:

- System Requirements
- System Features
- System Components

The FSICEBASE works with CodeWarrior software, which is produced by Metrowerks, Inc. The CodeWarrior software provides an integrated development environment that includes an editor, assembler, and a user interface to the FSICEBASE system.

The environment allows you to perform source-level debugging. The CodeWarrior software also simplifies the process of managing and building a software project, and debugging code for an embedded MCU system. The benefit to you is reduced development time.

System Requirements

The FSICEBASE system requires a host computer with the following minimum specifications:

- Processor: 200 MHz Pentium® II processor or AMD-K6® class processor
- Operating System: Microsoft® Windows® 2000/XP
- RAM: 128 MB
- Hard drive space: Compact software installation: 232 MB
Full software installation: 344 MB
- USB port or Ethernet port to connect host computer to the FSICEBASE

System Features

The Freescale In-Circuit Emulator Base (FSICEBASE) is a full-featured development system that provides in-circuit emulation. Features include:

- Real-time, non-intrusive, in-circuit emulation
- Allows you to set four complex data or instruction breakpoints; a breakpoint can be qualified by an address, an address range, data, or externally connected logic clips.
- Up to 128k real-time variables (any ROM or RAM memory area)
- Up to 128 Kbytes of emulation memory to accommodate the largest available ROM size of current HC08 MCUs
- Unlimited hardware instruction breakpoints and data watchpoints over up to 128-K memory map (conditional or unconditional)
- Built-in real-time bus state analyzer:
 - Large 1.33M x 92-bit real-time trace buffer - capture up to 24-bit address, 8-bit data, 32-bit time tag, instruction bit, R/W bit, extended access bit, trigger, and 24-bits of external clip information
 - Four hardware triggers for controlling real-time bus analysis or to provide additional complex breakpoints
 - Nine triggering modes
 - Display or dump real-time trace data as raw data, disassembled instructions, raw data and disassembled instructions, or assembly-language source code
 - As many as 1.33M pre- or post-trigger points
 - Trace buffer can be filled while single-stepping through user software
 - Large 32-bit time tag
 - Custom time tag clock from 4100Hz to 40MHz in 5kHz steps or from an external source, permitting wide time variance between analyzer events
 - Up to 24 general-purpose logic inputs, 5 of which can be used to trigger the bus state analyzer
- Custom MCU clock from 4100Hz to 40MHz in 5kHz steps or from external source
- Command and response logging to disk files
- C/C++ and/or assembly-language source-level debugging
- On-screen, context-sensitive help via pop-up menus and windows
- Emulation that allows multiple types of reset
- Meets ECC92 European electromagnetic compatibility standards

System Components

The FSICEBASE system includes the basic components that you need to use with an emulation module (EM) and target cable/adaptor assemblies. You can separately purchase these and other additional components that will complete or enhance your debugging and emulation debug system.

Basic Components

The Freescale In-Circuit Emulator Base product includes the following components

- Base station
 - The connectors on the top of the box let you connect an emulation module (EM).
- Cables, connectors, and adapters:
 - Crossover Ethernet cable (connects directly to an Ethernet Network Interface Card (NIC) on a PC)
 - Straight-through Ethernet cable (connects to a hub or switch)
 - Universal Serial Bus (USB) cable
 - MON08 port cable (16-pin ribbon cable)
 - External power supply (+5VDC 3.7A regulated) and universal plug adapters
 - Bus state analyzer logic clip cable assemblies: twisted-pair cables and hooks that connect the station module to your target system. You can also use the cable assembly to connect a test fixture, a clock, or any other circuitry that you might use to perform analysis. One end of each cable assembly has a molded connector, which fits into the FSICEBASE. Leads at the other end of each cable terminate in female probe tips. Hooks come with the cables.
- System software
 - CodeWarrior software, featuring an editor, C/C++ compiler, assembler, and C/C++ and/or assembly source level debugger
- Documentation:
 - Freescale In-Circuit Emulator Base User Manual (this manual)
 - CodeWarrior User's Manual (online)
 - Freescale In-Circuit Emulator Base Quick Start
 - Online Help and PDFs

Additional Components

You can purchase other components to complete and/or enhance your development efforts:

- An emulation module (EM) - required for emulation, but not for MON08 debug

An emulation module (EM) is a printed circuit board that emulates the features of a specific set of microcontroller units (MCUs). An EM completes the functionality of the FSICEBASE for a particular MCU or MCU family and is required to use the FSICEBASE emulation features. The FSICEBASE works with a variety of EMs. You can purchase EMs separately from the FSICEBASE.

The two connectors on the bottom of the EM fit into one of the sets of connectors on the top of the FSICEBASE box. The connections provides power and signal connections.

Connection to your target system is then made through a separately-purchased target cable, target head adapter, and footprint adapter that attaches to a target connector located on the top of the EM board.

- Optional target cable

You can separately purchase a target cable that is part of a cable assembly which is used to connect a target system to the FSICEBASE.

- Optional target head adapter

You can separately purchase a target head adapter that is part of a cable assembly which is used to connect a from a footprint adapter to the target cable.

- Optional footprint adapter

You can separately purchase a footprint adapter that connects to the target system and plugs into the target head adapter.

- Optional Bus State Analyzer (BSA) cables

The base station contains ports for three BSA pods. You can purchase BSA cables in addition to those supplied with the FSICEBASE system.

- Optional programming adapter

A programming adapter is a board that provides a MON08 connector and several sockets you can use to program the flash of MCUs in particular packages.

Getting Started

This chapter explains how to set up the development system. This chapter contains the following sections:

- Setting Up The Development System
- Creating a Project
- Adding Files
- Building (Compiling) a Project
- Establishing Communication
- Starting the Debugger

Setting Up The Development System

The Freescale In-Circuit Emulator Base (FSICEBASE) development system includes basic connection cables and software. You need to connect the cables and install the software in order to use the FSICEBASE. This section contains the following topics to help you set up the system:

- Setting Up The Hardware
- Installing CodeWarrior Software

Setting Up The Hardware

This section explains how to connect a host computer to the Freescale In-Circuit Emulator Base (FSICEBASE). There are three ways to connect a host computer to the FSICEBASE:

- directly from the USB port of a host computer to the FSICEBASE USB port
- directly from the Ethernet port of a host computer to the FSICEBASE Ethernet port (crossover connection)
- from the host computer, through a Local Area Network (LAN), to the FSICEBASE Ethernet port (straight-through connection)

Getting Started

Setting Up The Development System

To connect the FSICEBASE to a host computer:

1. If you are using an Ethernet connection to connect your host computer to the FSICEBASE through a Local Area Network (LAN):
 - a. Connect host computer to LAN
 - b. Connect FSICEBASE to LAN
 - Make sure power supply is not connected to FSICEBASE
 - Connect one end of Ethernet cable to Ethernet port of FSICEBASE (make sure to use the straight-through Ethernet cable when connecting to LAN)
 - Connect other end of Ethernet cable to Local Area Network (LAN)

NOTE To complete the connection through a LAN, you will need to obtain the IP address, subnet mask, and default gateway information from your network administrator. You will use this information in a later step.

2. If you are using an Ethernet connection to connect your host computer directly to the FSICEBASE (not through a LAN):
 - Make sure power supply is not connected to FSICEBASE
 - Connect one end of Ethernet cable to Ethernet port of FSICEBASE (make sure to use the cross-over Ethernet cable when connecting directly to a Network Interface Card (NIC))
 - Connect other end of USB cable to host computer

NOTE The host computer (PC) must have an assigned IP address and a subnet mask that matches the FSICEBASE.

3. If you are using a USB connection to connect your host computer directly to the FSICEBASE:
 - Make sure power supply is not connected to FSICEBASE
 - Connect square-shaped end of USB cable to FSICEBASE
 - Connect other end of USB cable to host computer

NOTE You must first install the CodeWarrior software for your PC to properly recognize the FSICEBASE USB device.

4. Connect Power supply to FSICEBASE
 - a. Connect round end of 5-volt power cord to barrel connector on FSICEBASE
 - b. Plug power supply into surge-protected strip
 - c. Connect surge-protected strip to AC outlet

5. Switch Power switch to on

Power and ready LED lights after the base station finished start-up sequence.

There are three status LEDs on the box: busy, ready, and error. The FSICEBASE base station takes about 5-10 seconds to start up. After powering the unit, you must wait for the ready LED to light before attempting to communicate.

The FSICEBASE is now ready to accept communication with a host computer. You will need to install the CodeWarrior software, create a project, and start the debugger to establish communication between your host computer and the FSICEBASE.

Installing CodeWarrior Software

The software portion of the development system consists of two parts. One part is the CodeWarrior Development Studio for HC(S)08 v3.1, which does not contain support for the FSICEBASE. The other part is a software patch that adds support to the CodeWarrior software Development Studio for HC(S)08 v3.1 for the FSICEBASE. This software patch is only required for CodeWarrior fDevelopment Studio for HC(S)08 v3.1. The following procedure explains how to install both pieces of software.

To install the CodeWarrior software on your host computer:

1. Install CodeWarrior IDE
 - a. Insert **CodeWarrior Development Studio** CD into CD-ROM drive — CW Auto Install begins

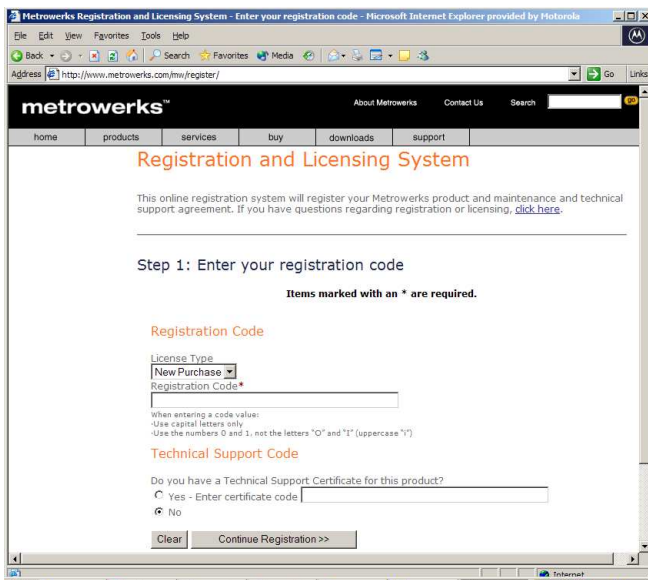
NOTE If Auto Install does not start, run launch.exe, which is located in the root directory of the CD.

- b. Follow setup program's on-screen instructions
2. Restart your computer — operating system reboots which ensures that CodeWarrior IDE finds newly installed drivers
3. Launch CodeWarrior IDE
 - a. Select **Start > Programs > Metrowerks CodeWarrior > CW08 V3.1** — menu appears
 - b. Select **CodeWarrior IDE** — IDE starts; main window appears
4. Register CodeWarrior software

Getting Started

Setting Up The Development System

Figure 2.1 Step 1 of the Register Form



- a. Select **Help > Register Product** from IDE main menu — CodeWarrior IDE starts browser, takes you to Step 1 of Metrowerks' on-line registration form (Figure 2.1)
- b. Enter your registration code in appropriate field

NOTE If you downloaded the software from the Metrowerks web site, you might not have a registration code. You can request a registration code from one of these addresses:

Americas, Asia: license@metrowerks.com
 Europe, Africa: license_europe@metrowerks.com
 Japan: j-license@metrowerks.com

- c. Click **Continue Registration** button — Step 2 appears
- d. Follow on-screen instructions to complete remaining pages of form (Thank You page is last) — within a few minutes Metrowerks emails your license authorization code

NOTE You do not need to wait for your license authorization code to begin using your CodeWarrior development studio. Your temporary license lets you create the sample project in demo mode.

5. Obtain license key

- a. From email message you receive from Metrowerks, copy license authorization code
 - b. Start CodeWarrior IDE
 - c. From CodeWarrior main menu bar, select **Help > License Authorization** — **License Authorization** dialog box appears
 - d. Paste license authorization code into **License Authorization** dialog box
 - e. Click **OK** button — **License Authorization** dialog box updates; Metrowerks emails you copy of new license key
 - f. Copy license key from **License Authorization** dialog box to clipboard, or wait for email from Metrowerks
 - g. Click **OK** button — **License Authorization** dialog box closes
 - h. From IDE main menu bar, select **File > Exit** – IDE closes
6. Install license key
- a. Open `license.dat` — you can use a plain text editor such as Notepad to open `license.dat` file

NOTE You can find the `license.dat` file in the directory where you installed the CodeWarrior software. The directory default is:
`C:\Program Files\Metrowerks\CW08 V3.1`

- b. Copy license key you received from Metrowerks
- c. Paste license key on new line at bottom of `license.dat` file
- d. Save `license.dat` file
- e. Close `license.dat` file – license is installed; IDE uses new license next time you start the CodeWarrior IDE

NOTE Do not move or delete the `license.dat` file. If you receive additional keys in the future for other CodeWarrior components, you can add the additional keys to the `license.dat` file.

7. Install software patch
- a. Insert the software service pack CD into your CD-ROM drive or download the service pack from <http://www.metrowerks.com>.
 - b. Run the installation executable and follow the on-screen instructions to install the service pack to the directory where you installed the CodeWarrior software. The default installation directory for CodeWarrior Development Studio for HC(S)08 v3.1 is:
`C:\Program Files\Metrowerks\CW08 V3.1`

Getting Started

Creating a Project

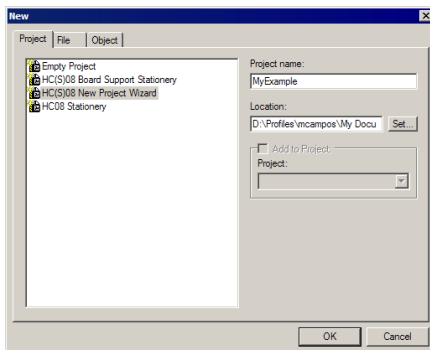
After you have installed the CodeWarrior software and license, you need to establish communication between your host computer and the FSICEBASE. You can perform application development with the CodeWarrior software without the connection, but you will not be able to use the features of the FSICEBASE until you establish communication.

Creating a Project

To create a project in the CodeWarrior IDE:

1. Launch CodeWarrior IDE
 - a. Select **Start > Programs > Metrowerks CodeWarrior > CW08 V3.1** — menu appears
 - b. Select **CodeWarrior IDE** — IDE starts, and CodeWarrior window appears
2. From IDE main menu bar, select **File > New** – New window appears (Figure 2.2)

Figure 2.2 New Window in the CodeWarrior IDE

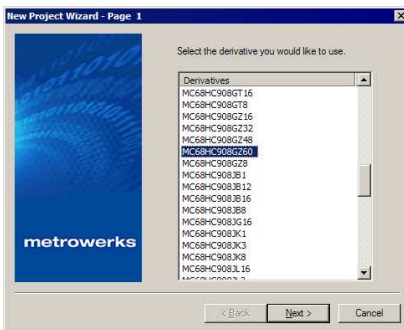


3. Create new project

NOTE You can use the New Project Wizard, the Board Support Stationery, or the HC08 Stationery to create a project. This procedure shows you how to use the New Project Wizard. We use an MC68HC908GZ60 target as an example.

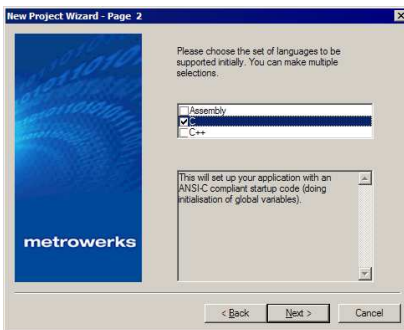
- a. Select **HC(S)08 New Project Wizard**
- b. In **Project name** text box, type name you want to give project – IDE automatically adds .mcp extension when it creates project
- c. Click **OK** button — first page of New Project Wizard appears

Figure 2.3 New Project Wizard - Page 1



- d. Select **MC68HC908GZ60**
- e. Click **Next** button — Page 2 of New Project Wizard appears

Figure 2.4 New Project Wizard - Page 2

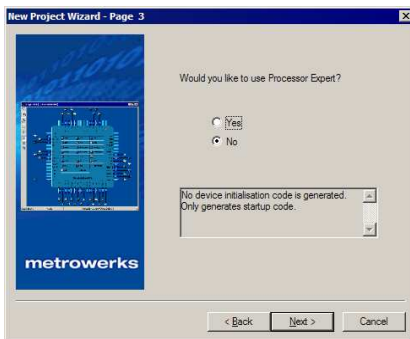


- f. Make sure **C** checkbox is marked
- g. Click **Next** button — Page 3 of New Project Wizard appears; allows you to specify whether you want project configured to use Processor Expert

Getting Started

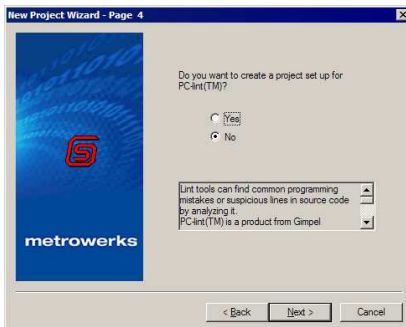
Creating a Project

Figure 2.5 New Project Wizard - Page 3



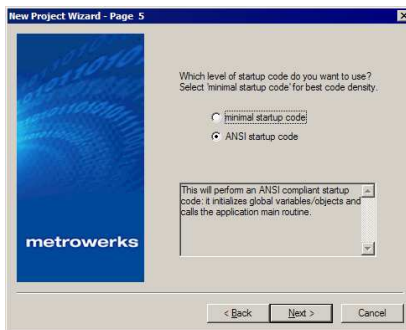
- h. Select **No**
- i. Click **Next** button — Page 4 of New Project Wizard appears; allows you to specify whether you want project configured to work with PC-lint

Figure 2.6 New Project Wizard - Page 4



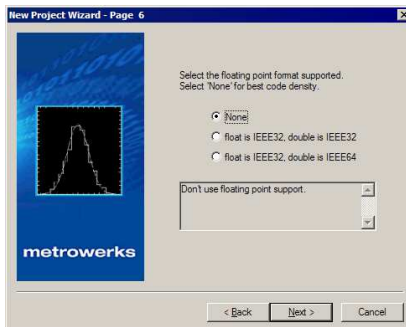
- j. Select **No**
- k. Click **Next** button — Page 5 of New Project Wizard appears; allows you to specify the code that the Wizard will place in the project as the startup code

Figure 2.7 New Project Wizard - Page 5



- l. Select **ANSI startup code**
- m. Click **Next** button — Page 6 of New Project Wizard appears; allows you to specify floating point format that project should be configured to support

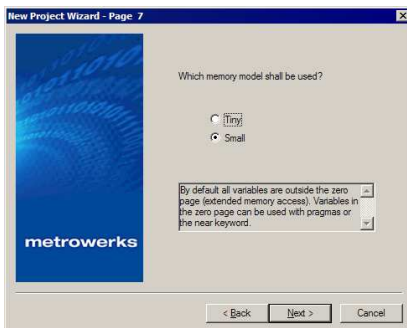
Figure 2.8 New Project Wizard - Page 6



- n. Select **None**
- o. Click **Next** button — Page 7 of New Project Wizard appears; allows you to specify memory model that project should be configured to support
- p. New Project Wizard - Page 7

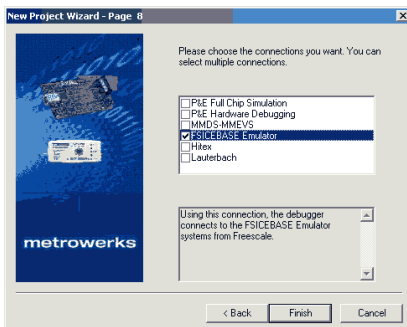
Getting Started

Creating a Project



- q. Select **Small**
- r. Click **Next** button — Page 8 of New Project Wizard appears; allow you to specify connections that project should be configured to support; Wizard creates a separate build target for each connection that you select

Figure 2.9 New Project Wizard - Page 8

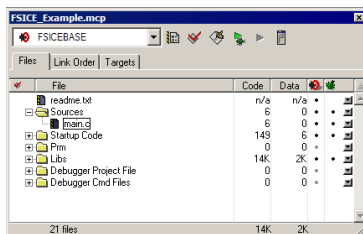


- s. Check **FSICEBASE Emulator** checkbox

NOTE If you want your project to include a build target that connects through the MON08 target interface, select **P&E Hardware Debugging**. You can select more than one connection.

- t. Click **Finish** button — system creates new project based on information you specified in New Project Wizard; Project window appears, docked at left side of main window

Figure 2.10 Project Window



NOTE To undock Project window, double-click the docking handle (double gray lines at top of the Project window). To re-dock window, right click in title bar of Project Window, and select **Docked**.

The project is ready for your use. You can add build targets, edit source files, modify build options, add files, and perform other development tasks. The following steps explain how to perform some of these tasks.

For more complete information on how to perform all development tasks with the IDE, see *The CodeWarrior IDE User Guide*.

4. Select build target

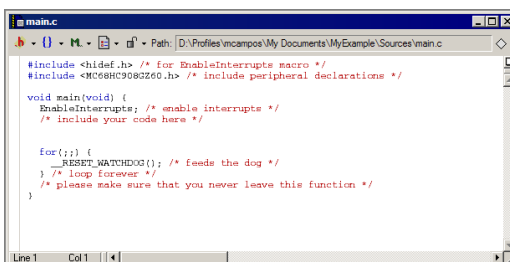
You can define multiple build targets in a single project. To use the build target that connects to the FSICEBASE, select FSICEBASE.

- a. In Project window click drop-down menu
- b. Select **FSICEBASE**

5. Edit source code

- a. Click + sign next to Sources folder — tree expands
- b. Double click **main.c** – Editor window opens displaying contents of main.c file

Figure 2.11 main.c in Editor Window



- c. Make changes to contents of main.c file if desired