

Getting Started with the Microcontroller Student Learning Kit Project Board

Using the MC9S12C32 Microcontroller Application Module

1 Introduction

This guide provides the steps necessary to begin development with a HCS12 family microcontroller (MCU) using your application module in conjunction with the project board.

This guide is separated into four sections. First, the software setup section will assist in installing the CodeWarrior™ development tools. Second, the hardware setup section will step through configuring the project board and application module. Third, the development section illustrates how to create, build, and debug an application. Fourth, the demonstration application takes the knowledge learned in step 1-3 and provides a simple example that exercises the many user-accessible features on the project board.

Refer to the project board user guide, application module user guide, and CodeWarrior documentation for more details.

Contents

1	Introduction.....	1
2	Software Setup	2
3	Hardware Setup	2
	3.1 Unpack	2
	3.2 Configure the Application Module.....	3
	3.3 Configure the Project Board	4
4	Development	9
	4.1 Creating and Building a Project in CodeWarrior Environment	9
	4.2 Debugging your Application.....	14
5	Running the Demonstration Application	16
6	Revision History	17

2 Software Setup

The following software installation is a one-time, required procedure per host PC that you intend to use with the project board and/or application module. After completing the software setup on your host PC, you can skip the software section on subsequent uses of your project board and/or application module and begin the hardware setup or development sections.

To install and register CodeWarrior development tools for HC(S)12 MCUs onto a host PC, complete the instructions outlined in sections A of the CodeWarrior Development Studio for Freescale 68HC12/HCS12/HCS12X/XGATE Microcontrollers quick start pamphlet found inside the supplied case labeled CodeWarrior Development Studio for HCS12(X) Microcontrollers Academic Edition.

NOTE

Academic Edition development tools are full-featured and not for commercial development. They may only be licensed to students, faculty, and staff of accredited institutions.

Periodically check <http://www.freescale.com/codewarrior> for updates and patches to your development tools. For questions on CodeWarrior development tools, go to <http://www.freescale.com/support>

3 Hardware Setup

3.1 Unpack

- 1) Open the shipping carton and remove the contents. Locate the packaging list document and verify items have been received from the packaging list.

Inspect both the project board and the application module for damage that may have occurred during shipping. If damage is found, contact the manufacturer at support@axman.com for assistance.

3.2 Configure the Application Module

- 1) Verify that all the jumper settings are configured properly on the application module as shown by Figure 1. Application Module Jumper Settings. Highlighted blocks indicate the on or installed position of jumpers. Table 1 summarizes the default state for the jumper settings. Please refer to application module user guide for more details and alternative configurations.

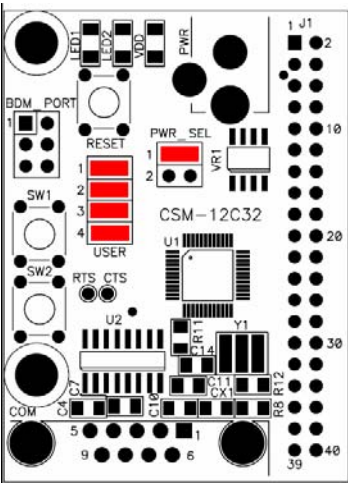


Figure 1. Application Module Jumper Settings

Table 1. Application Module Jumper Settings

Jumper Position	Jumper ID	Description
<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 10px; margin-right: 5px;"></div> <div style="margin-right: 10px;">1</div> </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 10px; margin-right: 5px;"></div> <div style="margin-right: 10px;">2</div> </div>	PWR_SEL	Source power input from connector J1..
<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 10px; margin-right: 5px;"></div> <div style="margin-right: 10px;">SW1</div> </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 10px; margin-right: 5px;"></div> <div style="margin-right: 10px;">SW2</div> </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 10px; margin-right: 5px;"></div> <div style="margin-right: 10px;">LED1</div> </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 10px; margin-right: 5px;"></div> <div style="margin-right: 10px;">LED2</div> </div>	USER	Enables access to application modules features for switch 1 (SW1), switch 2 (SW2), LED1, and LED2 to simplify circuit connections and emphasize software development.

3.3 Configure the Project Board

- Place the project board on a flat and sturdy surface. Ensure sufficient space is available around the project board to safely construct and test prototyped circuits.

Verify that all the jumper settings are configured properly on the project board as shown by Figure 2. Highlighted blocks indicate the on or installed position of jumpers. Circles indicate the location on the project board where the jumpers are located.

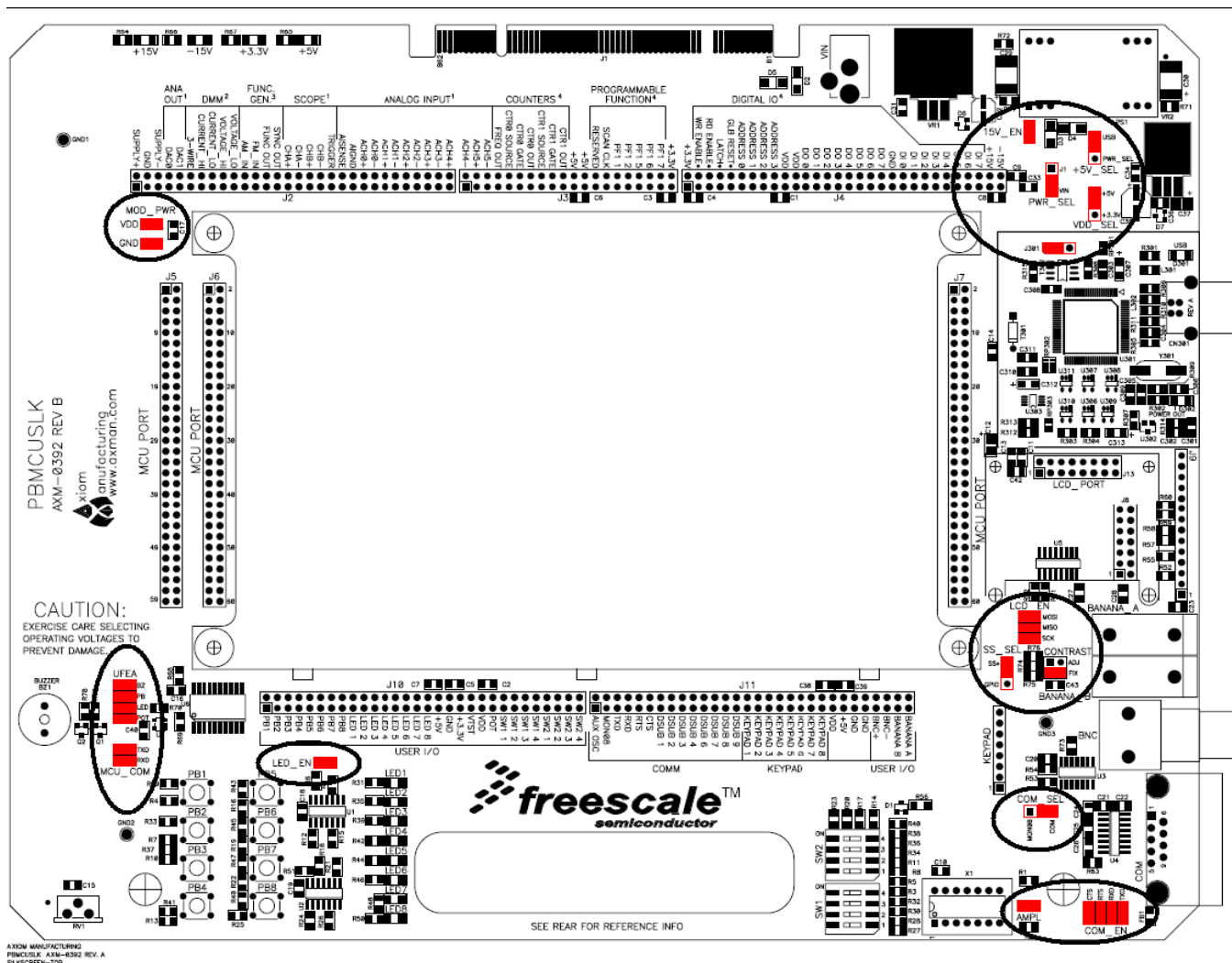





















Figure 2. Project Board Jumper Settings

Table 2 summarizes the default state for the jumper settings. Please refer to project board user guide for more details and alternative configurations.

Table 2. Project Board Jumper Settings

Jumper Position	Jumper ID	Description
 J1 VIN	PWR_SEL	Selects V_{in} (wall power transformer) as the default power source for the project board.
 USB PWR_SEL	+5V_SEL	Selects USB (power via USB cable from host PC) as the default power source for the project board. This jumper overrides PWR_SEL in USB position.
 +5V +3.3V	VDD_SEL	Sets VDD to +5V for project board.
 15V_EN	15V_EN	Enables PS1 output (± 15 V supply from dc-dc converter) to the project board. Selects if BDM Multilink pod on the project board has a generic or unique ID associated with the host PC.
 J301	J301	Selects if BDM Multilink pod on the project board has a generic or unique ID associated with the host PC.
 VDD GND	MOD_PWR	Selects dedicated SS input source to MCU_23 (SS) on MCU PORT connector to transfer data to LCD panel.
 SS GPIO	SS_SEL	Selects dedicated SS input source to MCU_23 (SS) on MCU PORT connector to transfer data to LCD panel.
 MOSI MISO SCK	LCD_EN	Connects dedicated SPI signals to LCD port.
 ADJ FIX	CONTRAST	Selects fixed LCD contrast voltage
 MONO8 COM	COM_SEL	Selects RS-232 communications for project board COM port.
 CTS RTS RXD TXD	COM_EN	Enables all RS-232 communication signals individually for project board COM port.

 AMPL	AMPL	Oscillator output at reduced amplitude – 3.3 Vpp
 LED_EN	LED_EN	Enables all LED outputs on project board.
 BZ  PB  LED  POT	UFEA	Enables access to direct connect features for buzzer, four push buttons, four LEDs, and potentiometer to MCU PORT connector to simplify circuit connections and emphasize software development.
 RXD  TXD	MCU_COM	Connects dedicated SCI signals to COM port.

Install the application module in the MCU PORT connector (J5) on the project board. Align pin 1 on the application module with pin 1 of the MCU PORT connector on the project board as marked by Figure 3.

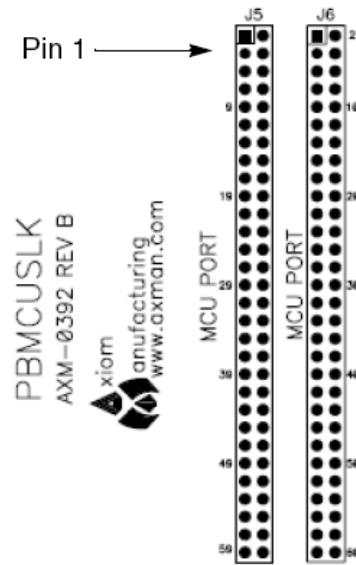


Figure 3. Installation of Application Module

Power up your host PC and connect the supplied USB cable to an available USB port on your host PC.

NOTE

Prior to executing the next steps, all CodeWarrior development tools must be installed per the instructions in the software setup section.

Connect the other end of the USB cable to the USB connector on the project board. Figure 4 illustrates the USB connector on the project board for the BDM Multilink pod interface to power, program, erase, and debug the application module board.

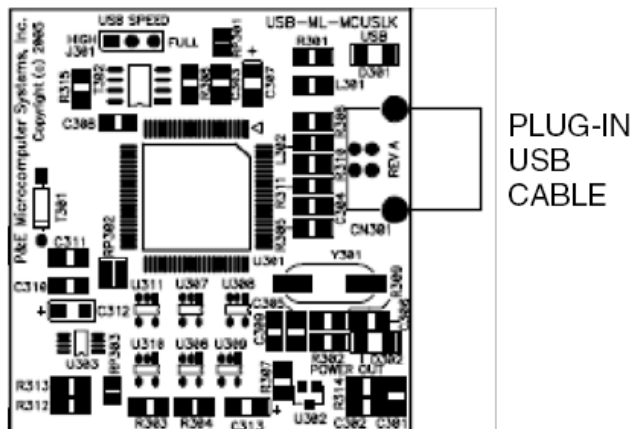


Figure 4. Installation of USB Cable

NOTE

Steps 8 through 11 are a one-time, required procedure per host PC that you intend to use with the project board and can be skipped on subsequent uses of your project board.

Allow your host PC to detect your new hardware. You will see a pop-up message in the task bar stating, “Found New Hardware.” Without any user action, the Found New Hardware Wizard will appear as illustrated by Figure 5 and prompt you to install software for USB Multilink 2.0.



Figure 5. Found New Hardware Wizard (Screen 1)

Insert into host PC the CD labeled MCU Project Development Board.

Select Install the software automatically (recommended) from the Found New Hardware Wizard and click Next to continue with installation.

Installation will begin with the Found New Hardware Wizard (Screen 2) illustrated by Figure 6. No user action necessary until software is completed installing.

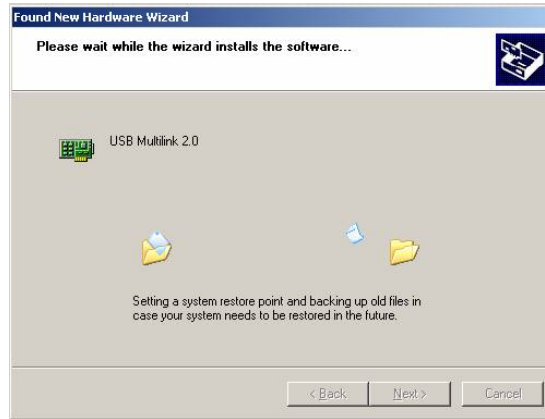


Figure 6. Found New Hardware Wizard (Screen 2)

Successful installation is verified by Found New Hardware (Screen 3) illustrated by Figure 7. Click Finish to complete installation and close the wizard.



Figure 7. Found New Hardware Wizard (Screen 3)

Verify that the +3.3 V LED, +5 V LED, and both green and yellow USB LEDs on the project board are lit.

Verify that the VDD LED on the application module board is lit.

If the LEDs in steps 13 and 14 are not lit, verify the jumper options are set as described in steps 2 through 6. Also, see the troubleshooting tips section in the project board user guide.

The project board and the application module are now ready for use.

4 Development

4.1 Creating and Building a Project in CodeWarrior Environment

After installing and registering the CodeWarrior development tools as described in Section 2, “Software Setup,” and completing the project board and application module configuration as described in Section 3, “Hardware Setup,” you can now begin to develop your application for a target microcontroller. This section will create and build a project under the CodeWarrior for HCS12(X) development tool platform.

NOTE

The instructions below are slightly different from the steps described in sections B and D of the quick start pamphlet included in the CodeWarrior for HCS12(X) development tools CD case. The instructions below are adjusted for the MCUSLK and MC9S12C32 application module.

- 1) Launch the CodeWarrior IDE
 - a) Select: Start > Programs > Freescale CodeWarrior > CW for HC12 V4.5; a menu will appear
 - b) Select: CodeWarrior IDE. IDE will start and a CodeWarrior window will appear

Create a new project

- 2) From the IDE main menu bar, select: File > New; a new window will appear as seen in Figure 8.

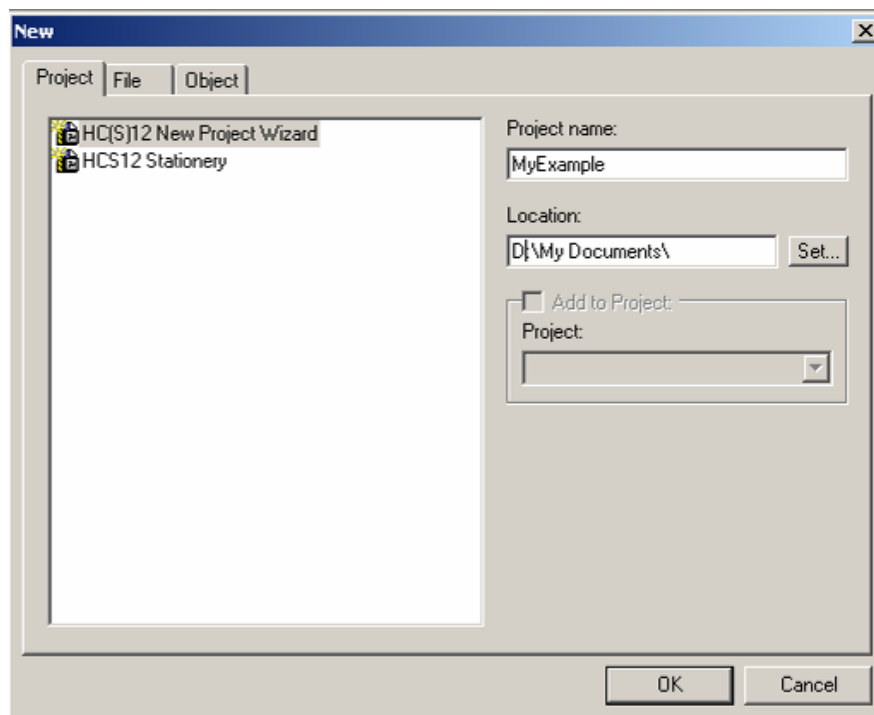


Figure 8. CodeWarrior New Project Window

- a) Select: HC(S)12 New Project Wizard.

- b) In the Project name text box, type the name you want to give the project. IDE automatically adds .mcp extension when it creates project.
- c) In the Location text box, set location where you want the project to be created.
- d) Click OK — the first page of the new project wizard will appear explaining to you how the project wizard works, proceed to the second page by clicking OK again (see Figure 9a)

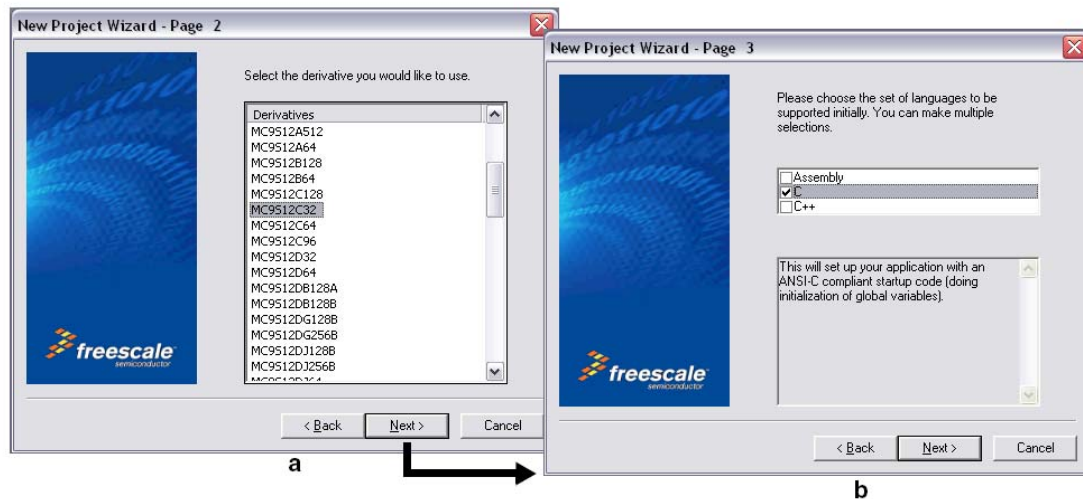


Figure 9. CodeWarrior New Project Wizard - Page 2 and Page 3

- e) Select MC9S12C32.
- f) Click Next — page 3 of the new project wizard will appear (see Figure 9b).
- g) Make sure C checkbox is marked. If you are using assembly or C++, mark the respective selection.
- h) Click Next — page 4 of the new project wizard will appear (see Figure 10a).

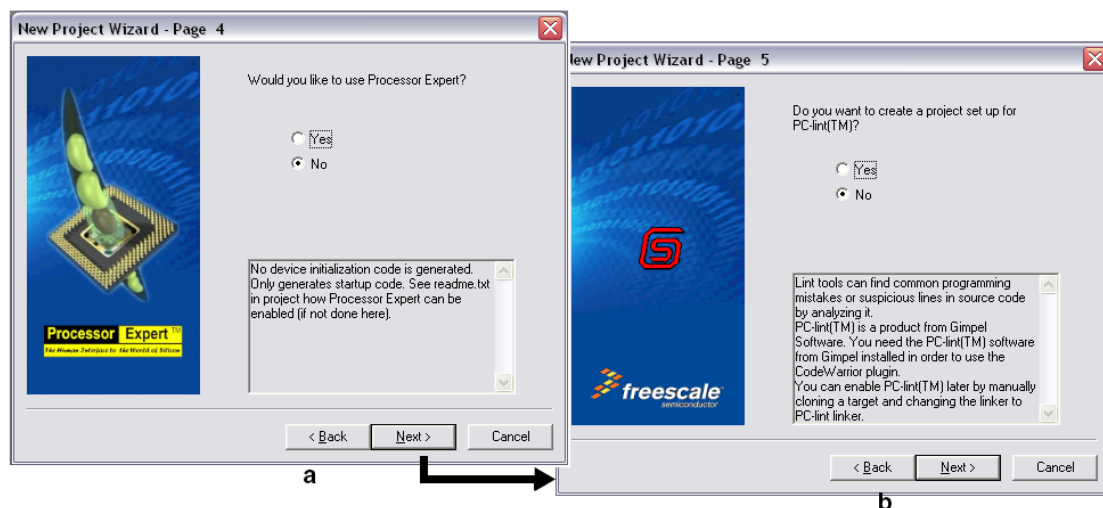


Figure 10. CodeWarrior New Project Wizard Page 4 and Page 5

- i) Select No, you do not want your project configured with Processor Expert.

- j) Click Next — page 5 of the new project wizard will appear (see Figure 10b).
- k) Select No, you do not want your project configured to work with PC-lint.
- l) Click Next — page 6 of the new project wizard will appear (see Figure 11a).

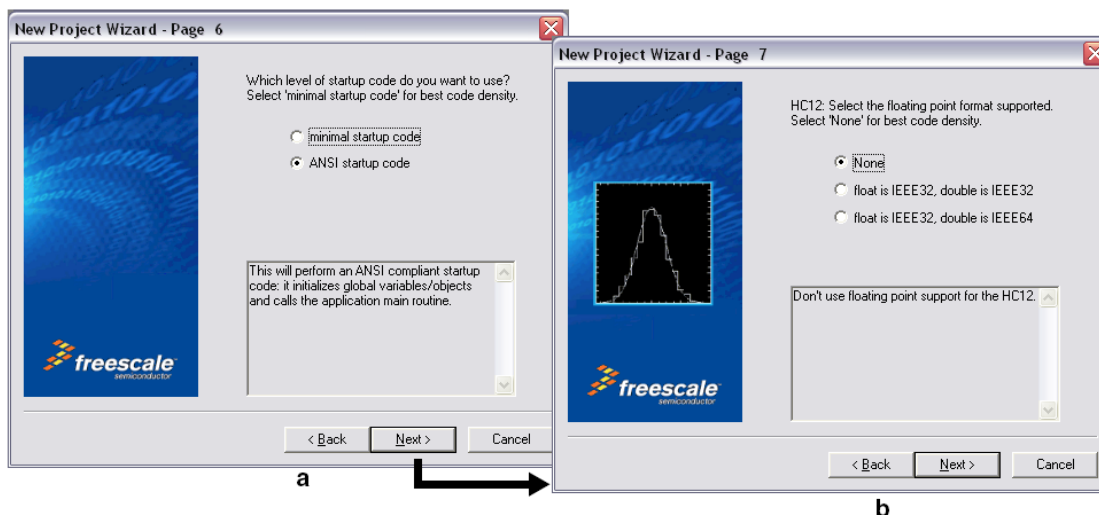


Figure 11. CodeWarrior New Project Wizard Page 6 and Page 7

- m) Select ANSI startup code.
- n) Click Next — page 7 of the new project wizard will appear (see Figure 11b).
- o) Select None, you do not want to specify a floating point format.
- p) Click Next — page 8 of the new project wizard will appear (see Figure 12a).
- q) Select Small memory model
- r) Click Next — page 9 of the new project wizard will appear (see Figure 12b)

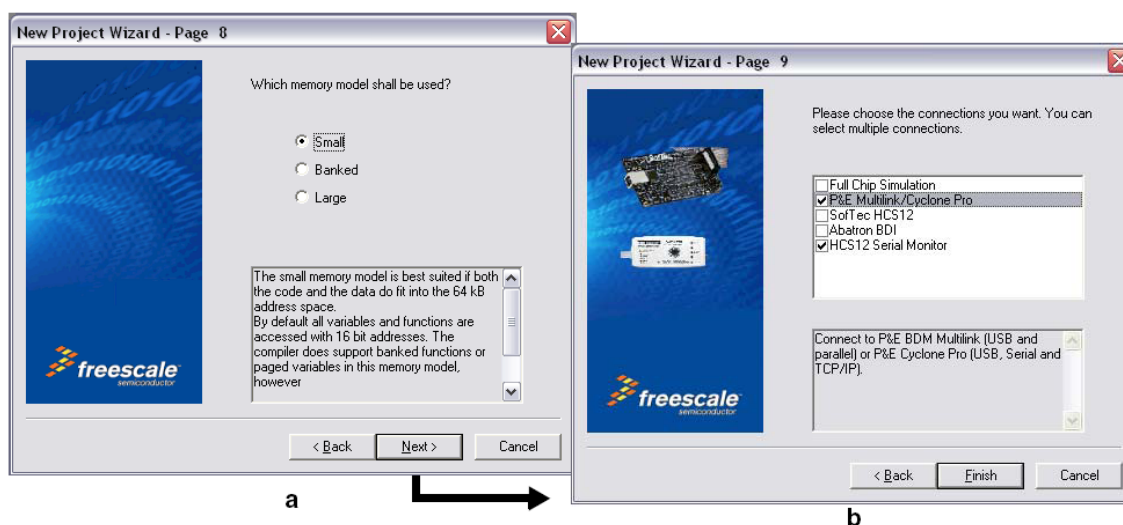


Figure 12. CodeWarrior New Project Wizard Page 8 and Page 9

- s) Page 9 allows you to specify connections that the project should be configured to support. Select P&E Multilink/Cyclone Pro and HCS12 Serial Monitor.
- t) Click Finish — the wizard creates a new project based on information you specified. A project window will appear, docked at left side of main window (see Figure 13).

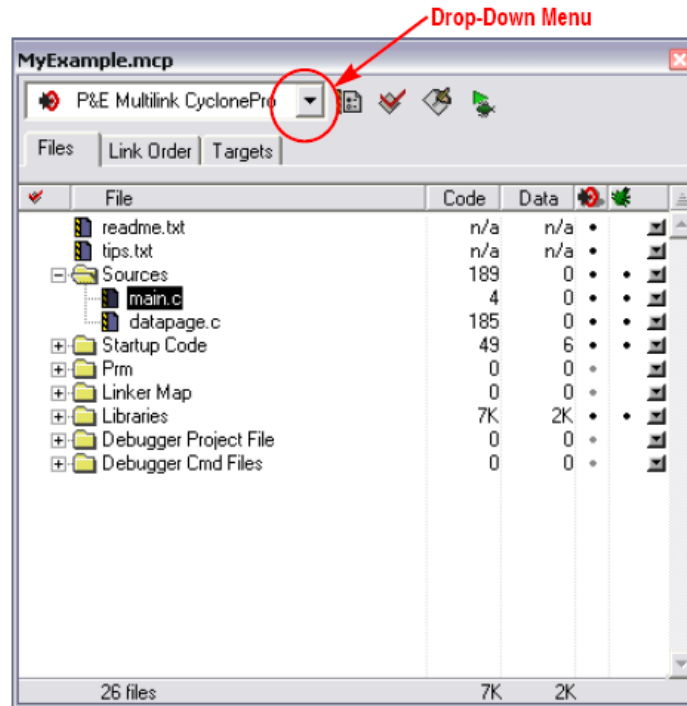


Figure 13. CodeWarrior Project Window

NOTE

To undock the 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.

3) Select Build Target.

Your project can contain multiple build targets. For this example, we use the P&E Multilink CyclonePro target, which uses the USB BDM pod on the project board. This BDM pod provides bidirectional communication between the microcontroller and the debugger.

- a) Click the drop-down menu of the project window (see Figure 13).
- b) Select P&E Multilink CyclonePro.

- 4) Edit source code.
 - a) Click the + sign next to the sources folder to reveal files (see Figure 13).
 - b) Double-click the main.c file. An editor window will open and display the contents of main.c (see Figure 14)

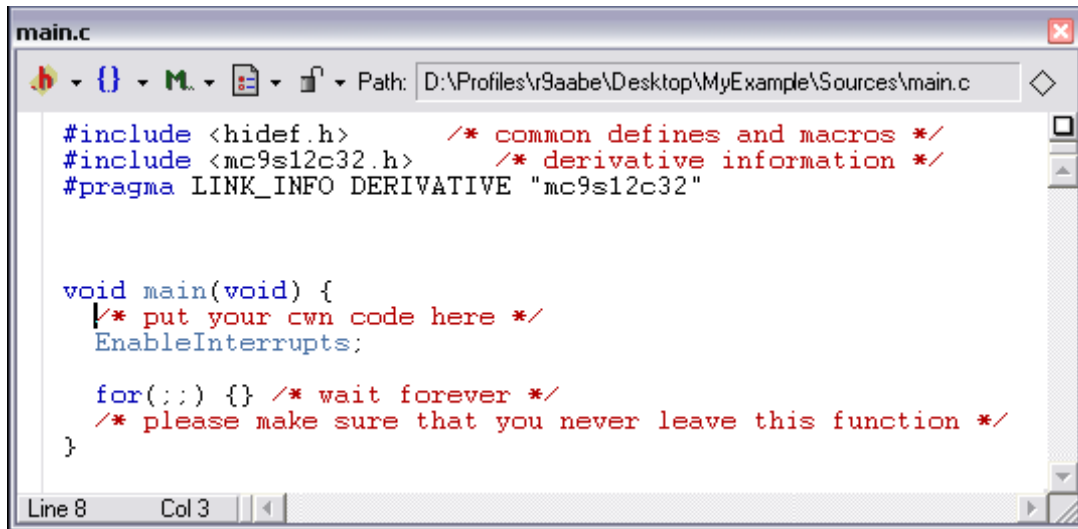


Figure 14. Editor Window - main.c

- c) Make changes to the contents of main.c file, if desired.
 - d) If you make changes to main.c, from IDE main menu bar, select File > Save.
- 5) Add files (if appropriate).
 - a) Highlight the Sources folder.
 - b) From the IDE main menu bar, select Project; a menu will appear.
 - c) Select Add Files. A dialog box will appear.
 - d) Navigate to the directory that contains the file you want to add.
 - e) Highlight the filename of the file you want to add to your project.
 - f) Click Open — the Add Files dialog box will appear.
 - g) Check the checkbox for each build target to which the file applies.
 - h) Click OK — the Add Files dialog box closes. In the Project window, the filename of added file will appear under the Sources folder.
- 6) Build project.
 - a) From the IDE main menu bar, select Project.
 - b) Select Make. IDE builds (assembles, compiles, and links) project. Error and warnings window will open and show error messages and warning messages, if appropriate.

4.2 Debugging your Application

The following steps explain how to establish communication and upload your application software to the application module. This will allow you to debug your application through CodeWarrior True-Time Simulator & Real-Time Debugger using the BDM pod interface on the project board. Section 2, “Software Setup” and Section 3, “Hardware Setup” must be completed before executing the steps in this section.

- 1) Make sure power is applied to the project board. If not, go back and complete hardware setup section instructions.
- 2) Start debugger.
 - a) From the main menu bar in CodeWarrior IDE, select Project.
 - b) Select Debug — the Debugger window will open in MultilinkCyclonePro target.
 - c) You will be prompted by a Loader Warning window that the debugger will mass-erase and program the microcontroller with your application. Click OK to continue.

NOTE

If prompted by debugger with a communication setup window, select the appropriate host PC USB port being used by the BDM pod and click OK.

- d) Once the debugger is launched a window similar to Figure 15 will appear. The operation of the various subwindows is listed in Table 3 on the subsequent page. The configuration of sub-windows can be user configured and saved using File->Save Workspace

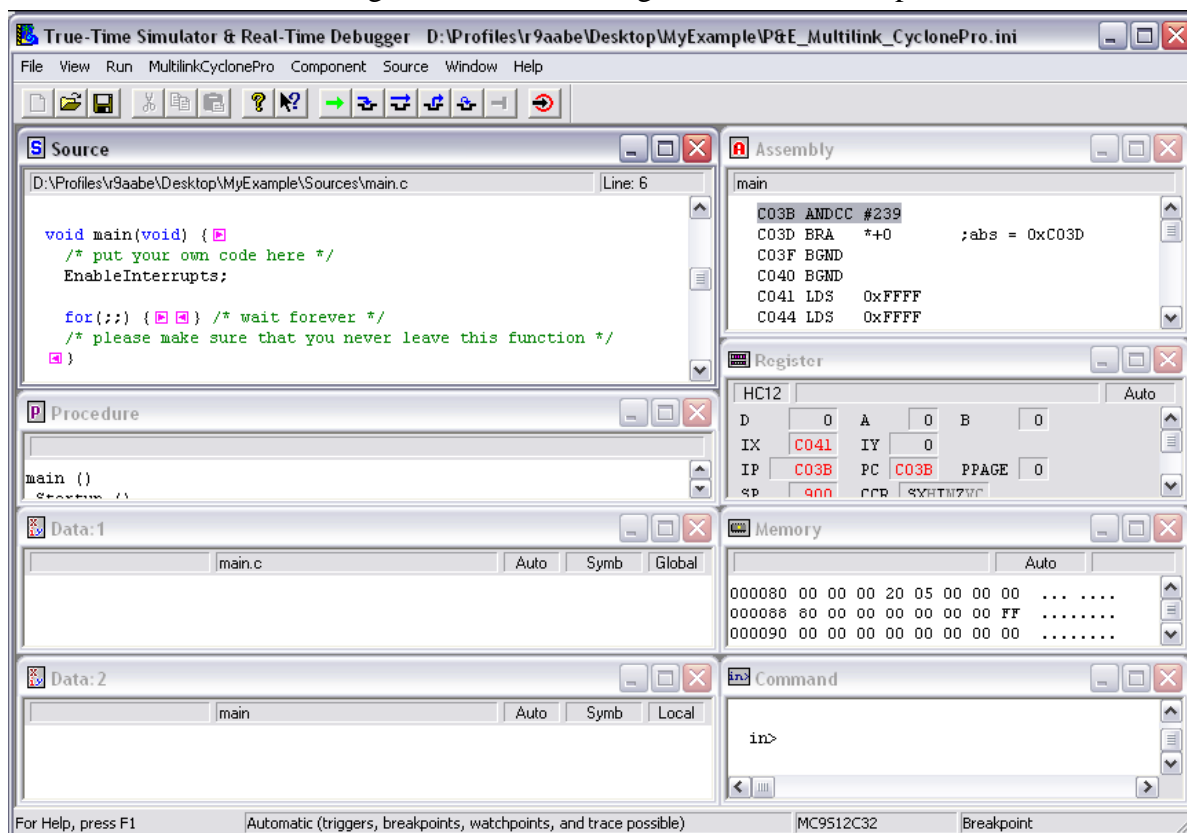



Figure 15. CodeWarrior Debugger Window



Table 3. Debugger Window Descriptions

Window Name	Description
Source and Assembly	Display programmed application software in C and assembly programming languages, respectively
Register	Displays the MCU internal registers
Memory	Displays the active value at each memory address in the MCU internal memory map
Procedure	Displays the active procedure being evaluated
Data 1 and Data 2	Display and global or local variables in your application software and their active value.
Command	Allows user control and log command execution






- 3) Reset the target by clicking on the reset target icon. 
- 4) Right-click anywhere in the Source window and select Open Source File. The Source Files menu will appear. To view between source files in your application software, select the appropriate file in the menu, then click OK.
- 5) Right-click on executable line of source code in Source window to set breakpoints or triggers in program code.
- 6) Run application.
 - a) From debugger main menu, select Run — the Run menu will appear.
 - b) Select Start/continue; program will execute until encountering first breakpoint, if breakpoint is present. Command pane displays program status.

NOTE

Alternatively, you can click on start/continue icon  in the debugger taskbar

- 7) Click start/continue icon, , to resume program code execution (if breakpoint occurred) or click halt icon, , to stop program execution.

NOTE

The debugger taskbar includes other icons for debugging which allow single step , step over , step out , assembly step , and reset target  functions.

- 8) In the debugger window tool bar, select: File > Exit (to exit debugger).
- 9) In the IDE main window tool bar, select: File > Exit (to exit CodeWarrior IDE).

Congratulations — You have successfully developed a project using CodeWarrior development tools with your project board and application module.



5 Running the Demonstration Application

The following example demonstrates the PBS12C32SLK hardware and provides an example code using the CodeWarrior software tools.

- 1) Configure the project board to the default jumper positions. Refer to Section 3.2 steps 1-15 for more information.
- 2) Unzip and extract PBS12C32SLKSW.zip to a directory on your PC.

NOTE

You can download the above file at www.freescale.com and searching for PBS12C32SLK. This software is intended to be used with the Project Board. Also, there is software example for use with the module independently, and only with on-board peripherals (APS12C32SLKSW).

- 3)
- 4) From the location that you extracted the files to, open the PBS12C32SLK_Demo folder. Double-Click on PBS12C32SLK_Demo.mcp
- 5) Click the Run Debug  button to launch the real-time debugger.
- 6) Click the Run button  to run the application in the debugger.
- 7) At this time you should also notice that LED's 1-4 on the project board are beginning to cycle and the LCD displays .PBS12C32SLKSW.
- 8) Press PB1 on the project board and adjust the potentiometer. The buzzer is enabled with a 440Hz frequency and the potentiometer value is sampled to control the tone of the buzzer while the button is pressed. Also, note that while pressed the LCD display is now showing PBMCUSLK BUZZER1.
- 9) Press PB2 on the project board and adjust the potentiometer. The buzzer is enabled with an 880Hz frequency and the potentiometer value is sampled to control the tone of the buzzer while the button is pressed. Also, note that while pressed the LCD display is now showing PBMCUSLK BUZZER2.
- 10) PB3 is reserved for the motor control demonstration, this requires external hardware interfacing. Please see Motor Control White Paper for information on connecting this feature. Also, note that while pressed the LCD display is now showing .MOTOR CONTROL.
- 11) Press PB4 on the project board and adjust the potentiometer. LED's [1..4] cycling speed is increased or decreased based on the position of the potentiometer while the button is pressed. Also, note that while pressed the LCD display is now showing LEDSPD CONTROL.

6 Revision History

Version	Date	Revised By	Description of Changes
0	8/2006	r2aakl	Initial Revision
1	7/2007	R2aakl	Corrected Software File Names

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 or +1-480-768-2130
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 or +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 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Design and/or Manufacturing services for this product provided by:

Axiom Manufacturing
2813 Industrial Lane
Garland, Tx. 75041
Phone: 972-926-9303
Web: www.axman.com
Email: sales@axman.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.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.

All other product or service names are the property of their respective owners.