8-bit Low Pin Count MCU Hands On Seminar

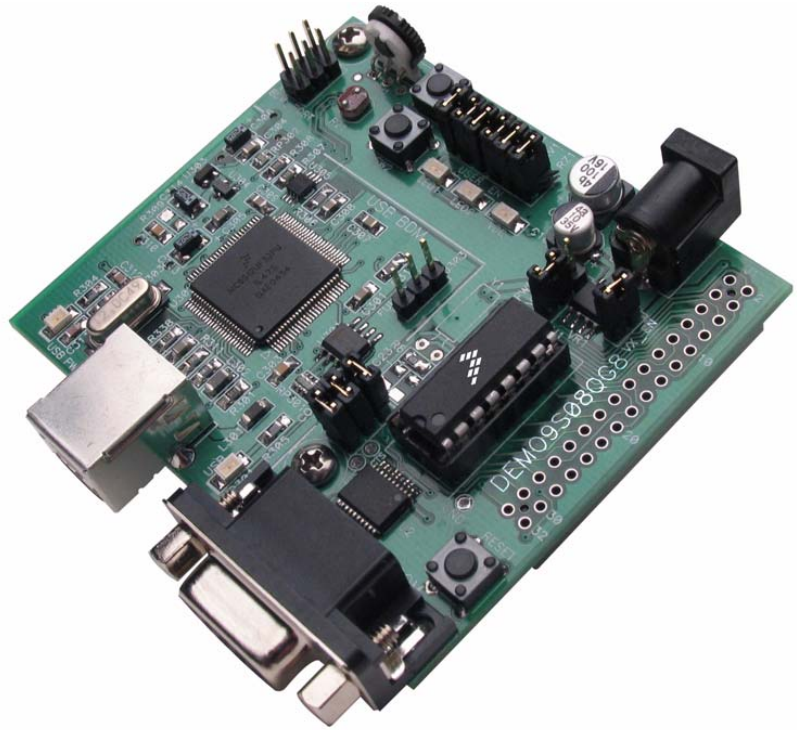
CodeWarrior Hands On Lab

October 25, 2005

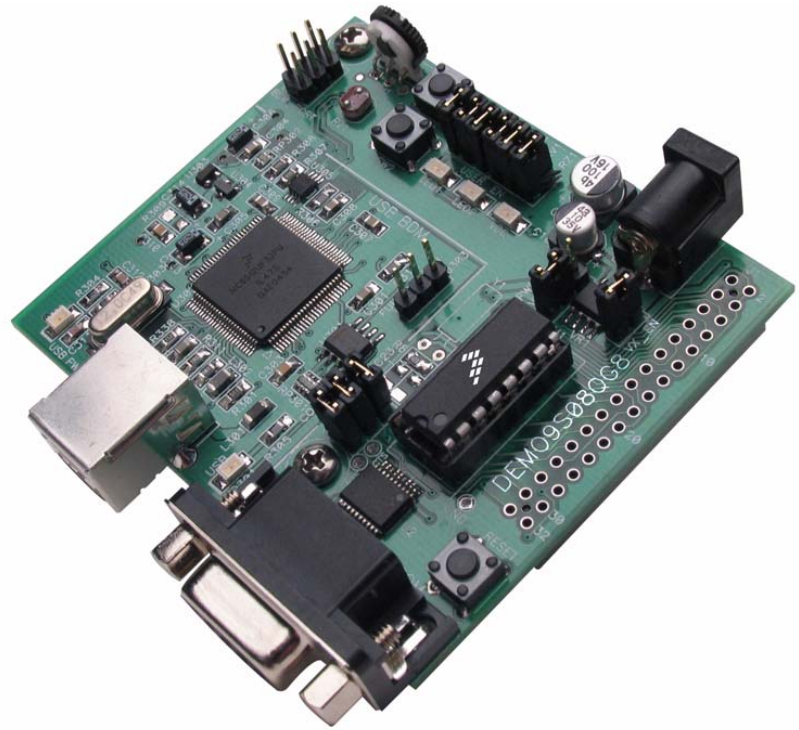
freescale™ semiconductor

Agenda

- MC9S08QG8 Demo Kit
- CodeWarrior Installation
- Lab 1 CodeWarrior Project
- Lab 2 Application Software
- Lab 3 ICS Lab
- Lab 4 MTIM Lab
- Lab 5 Analog Comparator Lab
- Lab 6 ADC Lab
- Questions



- **DEMO9S08QG8**
  - DEMO9S08QG8 Demo Board
  - DEMO9S08QG8 User Guide
  - DEMO9S08QG8 Quick Start Guide
  - 68HC(S)08 Development Board CD
  - CodeWarrior CD’s
  - USB Cable
  - RS-232 Serial Cable (not included)

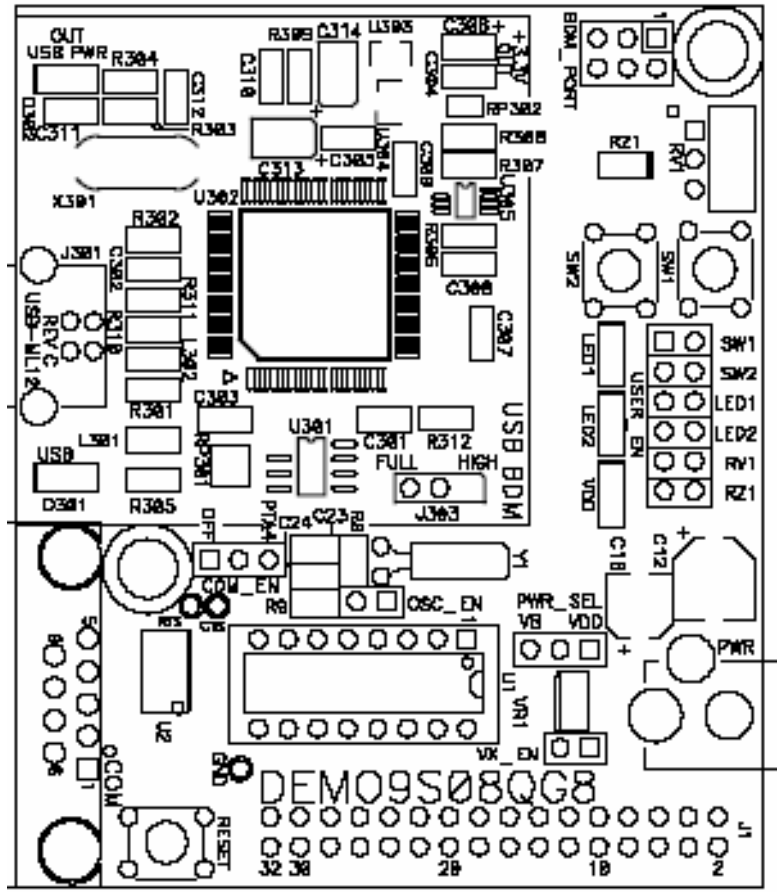


Complete List of HCS08 Tools is Available in the Development Tool Sector Guide.

MCS08QG8 Demo Kit

Features

- Integrated USB-BDM
- RS-232 Serial Port w/ DB9 Connector
- SPI, IIC ports available on Connector J1
- External 32.768 kHz Clock Oscillator (not installed)
- Power Input Selection Jumper
- Power input from USB-BDM
- Power input from on-board regulator
- Power input from Connector J1
- Power output through Connector J1
- User Components Provided
- 3 Push Switches; 2 User, 1 Reset
- 3 LED Indicators; 2 User, 1 VDD

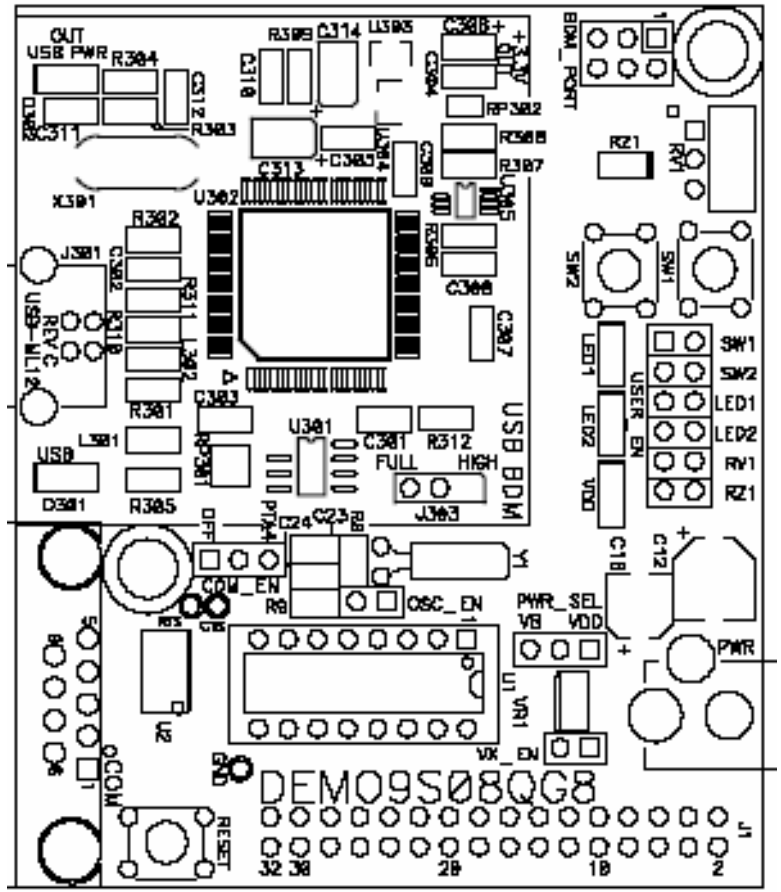


MCS08QG8 Demo Kit

Jumpers
- USER_EN
- PWR_SEL
- COM_SEL
- VX_EN
- OSC_EN (not installed)

Connectors
- 32-pin MCU I/O Connector
- 2.0mm Barrel Connector
- BDM_PORT Pin Header (not installed)
- DB9 Serial Connector

Specifications:
- Board Size 2.9” x 2.5”
- Power Input:
- USB Cable – 500mA max
- PWR Connector - 9VDC typical, +7VDC to +18VDC

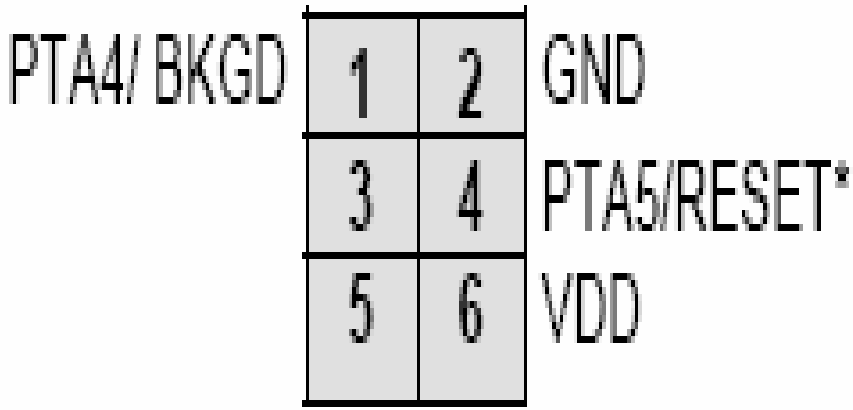


- Run Mode
  - Run mode allows the user application to execute when power is applied to the board or the RESET button is pressed. Use the following settings to configure the DEMO908QG8 board for RUN Mode using the USB bus to power the board.

- Debug Mode
  - Debug Mode supports application development and debug using the HCS08/HC(S)12 background debug mode (BDM). Background mode is accessible using either the integrated USBBDM or an external HCS08/HC(S)12 BDM cable. Use of the integrated BDM requires only a host PC with an available USB port and an A/B USB cable.

0x0000 – 0x005F	Direct Page Registers	96 bytes
0x0060 – 0x025F	RAM	512 bytes
0x0260 – 0x17FF	Unimplemented	5,536 bytes
0x 1800 – 0x184F	High Page Registers	80 bytes
0x1850 – 0xDFFF	Unimplemented	51,120 bytes
0xE000 – 0xFFAF	FLASH	8192 bytes
0xFFB0 – 0xFBFF	Non-Volatile Registers	
0xFC00 – 0xFFFF	Vectors	

BDM Port Header



Power Select

PWR_SEL

VB VDD

CONFIGURATION:

Select power input from USB-BDM

PWR_SEL

VB VDD

Select power input from VR1

Voltage External Enable

ON Enable power connection to connector J1

VX_EN

OFF Disable power connection to connector J1

VX_EN

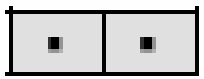
CAUTION: Do not apply power to connector J1 while also sourcing power from either the PWR connector or the USB-BDM circuit. Damage to the board may result.

NOTE: Do not exceed available current supply from USB-BDM cable or on-board regulator when sourcing power through connector J1 to external circuitry.

Oscillator Enable

ON Enables Crystal Oscillator Input to MCU

OSC_EN

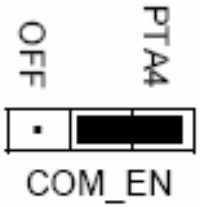


OFF

OSC_EN

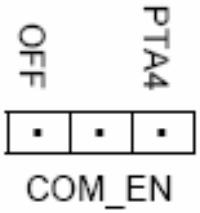
NOTE: This option header is not installed in default configuration.

Communication Enable



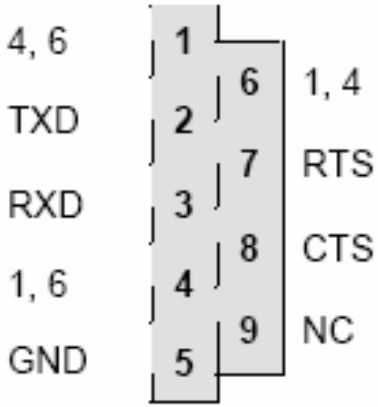
MCU controls RS-232 Transceiver

RS-232 Transceiver disabled



RS-232 Transceiver always ON. AutoShutoff disabled

COM Connector



Female DB9 connector that interfaces to the HC(S)08 internal SCI1 serial port via the U2 RS232 transceiver. It provides simple 2 wire asynchronous serial communications without flow control. Flow control is provided at test points on the board.

Pins 1, 4, and 6 are connected together.

- User Options
  - Pushbutton Switches
    - > SW1 and SW2 connect to input ports PTA2 and PTA3 respectively.
  - LED Indicators
    - > MCU ports PTB6 and PTB7 drive LED1 and LED2 respectively.
  - Potentiometer
    - > The potentiometer is connected between VDD and GND with the center tap providing the divider output. This center tap is connected to the MCU on signal PTA0.
  - Photocell
    - > Current flow within the phototransistor is inversely proportional to light intensity incident on the surface of the device. A rail-to-rail OP amp at U2 boosts the photocell output to useable levels. This signal is available to the MCU on signal PTA1.

User Options

Jumper	On	Off	MCU PORT	MCU PIN
SW1	Enable SW1	Disable SW1	PTA2	14
SW2	Enable SW2	Disable SW2	PTA3	13
LED1	Enable LED1	Disable LED1	PTB6	6
LED2	Enable LED2	Disable LED2	PTB7	5
RV1	Enable RV1	Disable RV1	PTA0	16
RZ1	Enable RZ1	Disable RZ1	PTA1	15

I/O Port Connector

VDD	1	2	PTA5/RESET/IRQ/TCLK
VSS	3	4	PTA5/RESET/IRQ/TCLK
PTB1/KBI1P5/ADC1P5/TXD1	5	6	PTA4/BKGD/MS/ACMP10
PTB0/KBI1P4/ADC1P4/RXD1	7	8	PTB7/SCL1/EXTAL
PTA2/KBI1P2/ADC1P2/SDA1	9	10	PTB6/SDA1/XTAL
PTA3/KBI1P3/ADC1P3/SCL1	11	12	
PTA5/RESET/IRQ/TCLK	13	14	
PTA0/KBI1P0/ADC1P0/TPM1CH0/AMCP+	15	16	
PTB3/KBI1P7/ADC1P7/MOSI1	17	18	PTA1/KBI1P1/ADC1P1/ACMP1-
PTB4/MISO1	19	20	PTA0/KBI1P0/ADC1P0/TPM1CH0/AMCP+
PTB2/KBI1P6/ADC1P6/SPSCK1	21	22	
PTB5/TPM1CH1/SS1	23	24	
PTA1/KBI1P1/ADC1P1/ACMP1-	25	26	
PTB6/SDA1/XTAL	27	28	
PTB7/SCL1/EXTAL	29	30	
PTA4/BKGD/MS/ACMP10	31	32	

Default Jumper Settings

Black blocks indicate the "on" or "installed" position of jumpers. Please check these settings before continuing.

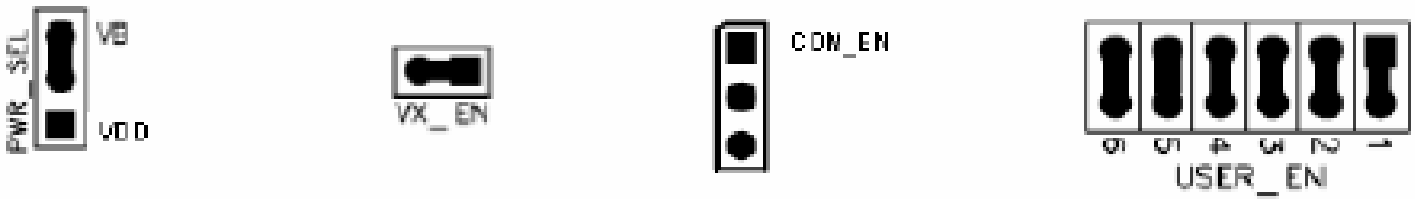


Figure 1. DEMO9S08QG8 Default Settings

Run Test Program

1. Check the jumper settings and make sure they are in the default position. Use Figure 1 as a guide.

2. Connect the USB cable to the PC and then to the board. If you are using this board for the first time, please follow the instructions on the screen to install the USB device properly. (If you have not installed CW 3.1 and the service pack, you will not have the proper files used to recognize the Demo Board.)

3. After the USB drivers are installed correctly, the USB, USB PWR OUT, and VDD LEDs will be on.

4. Press SW1 and LED1 will toggle.

5. LED2 will blink at a fast rate.

CodeWarrior Installation

1. Install CodeWarrior HC(S)08 CD
   Reference “CodeWarrior Quick Start” Section A

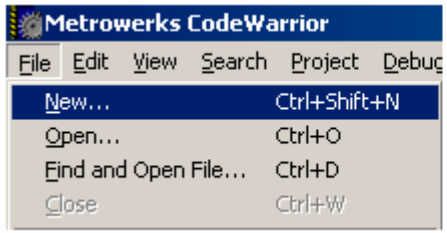
2. Install CodeWarrior Service Packs
   Install CW08 V3.1 USB Service Pack
   Install CW08 V3.1 QG8 Service Pack

3. Register Software
   Reference “CodeWarrior Quick Start” Section B

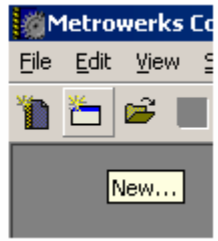
4. Install License Key
   Reference “CodeWarrior Quick Start” Section C

Lab 1 – CodeWarrior Project

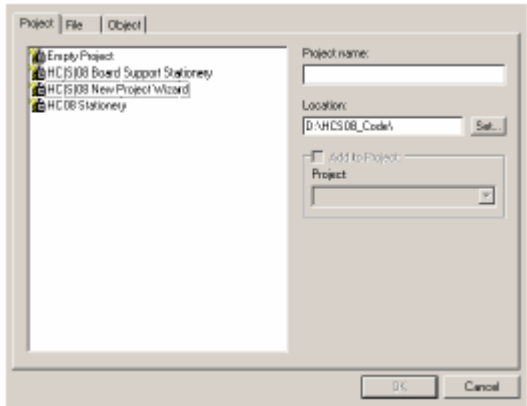
1. Create a new project by selecting **New…** from the **File** menu:



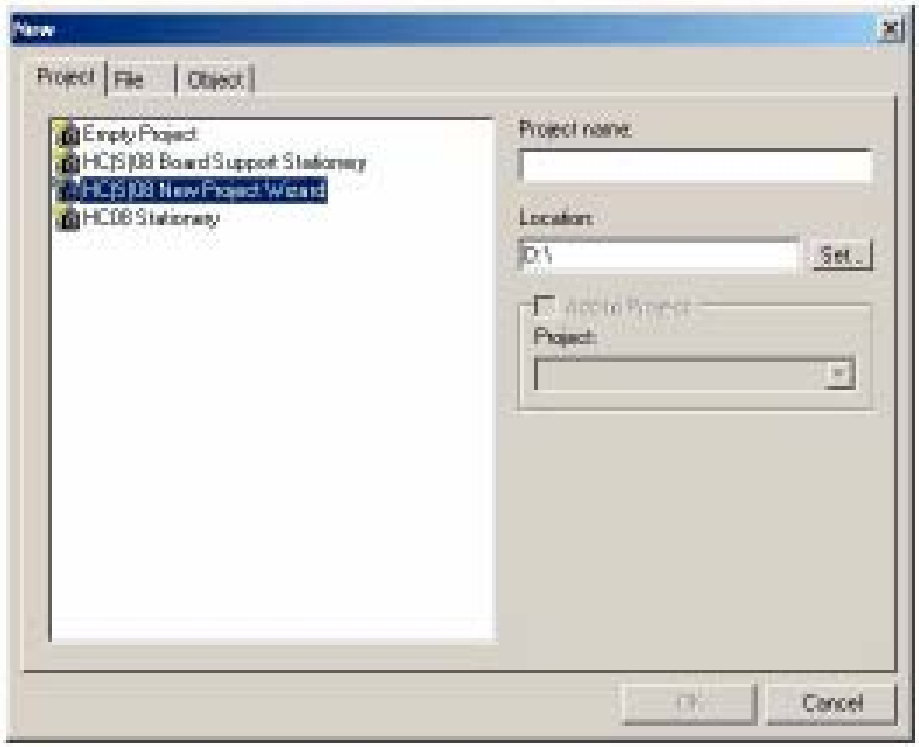
Or by pressing the New button:



Either of which will invoke the project creation dialog box shown below.

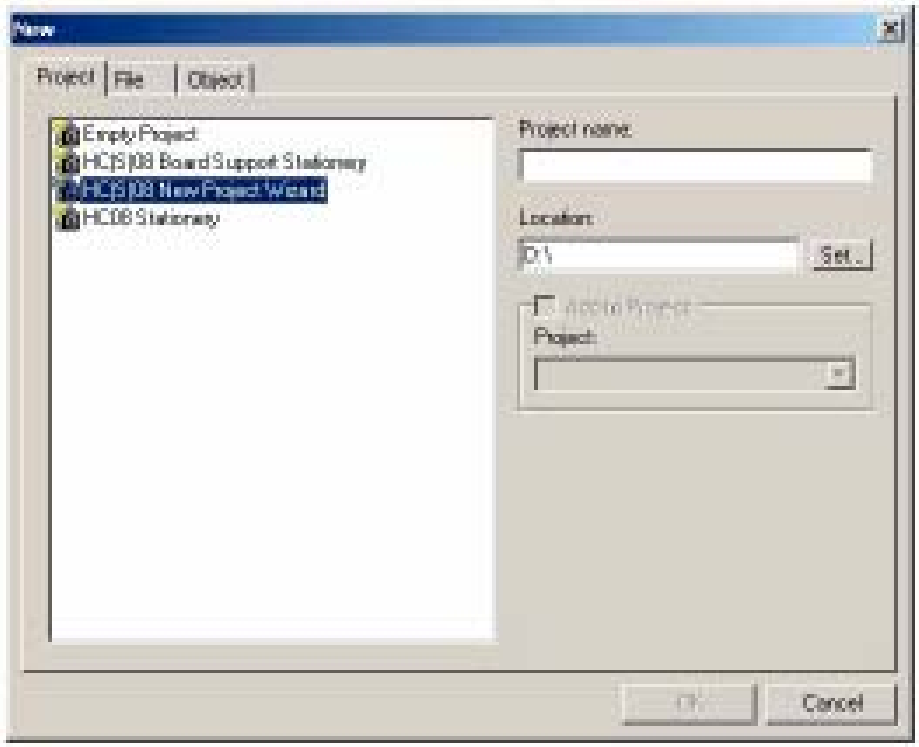


2. Select **HC(S)08 New Project Wizard**:

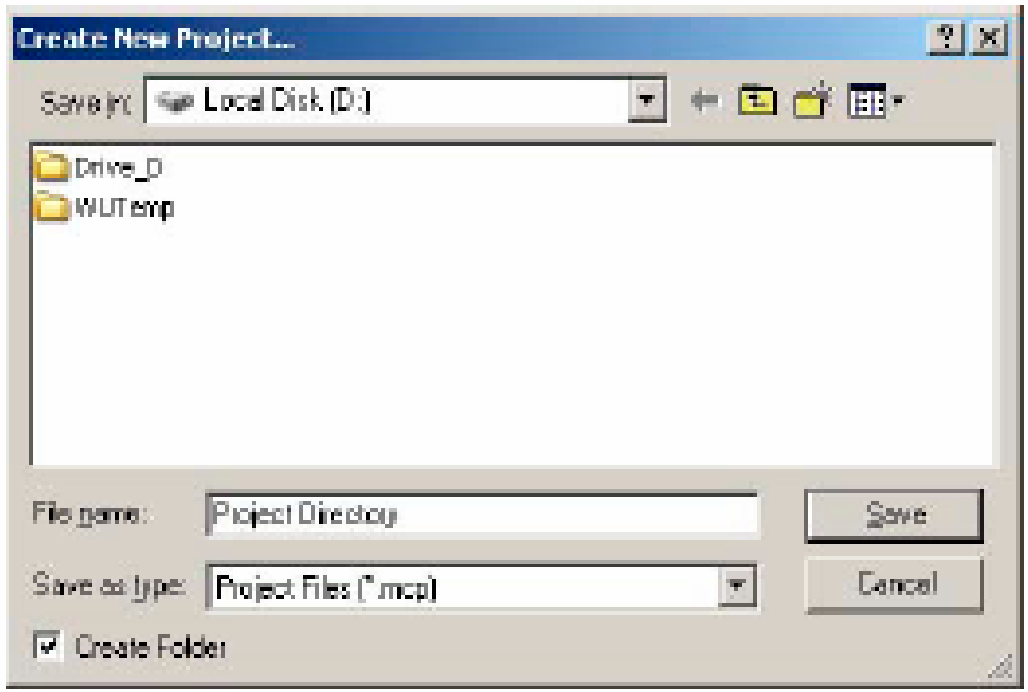


Lab 1 – CodeWarrior Project

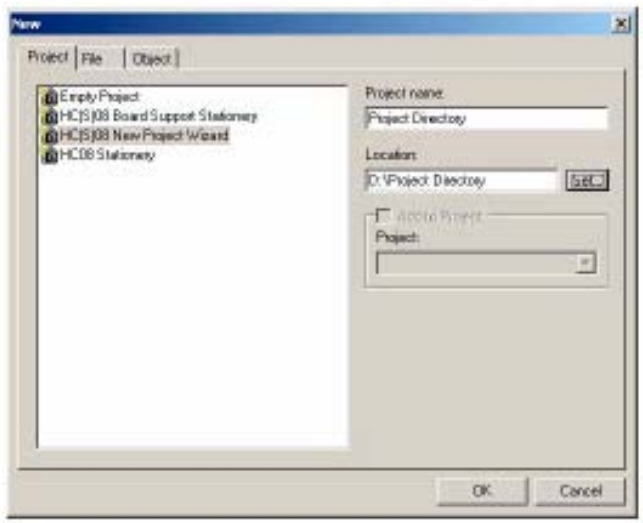
3. Select **SET** button to set the directory **Location.** This allows you to select the parent directory for the project:



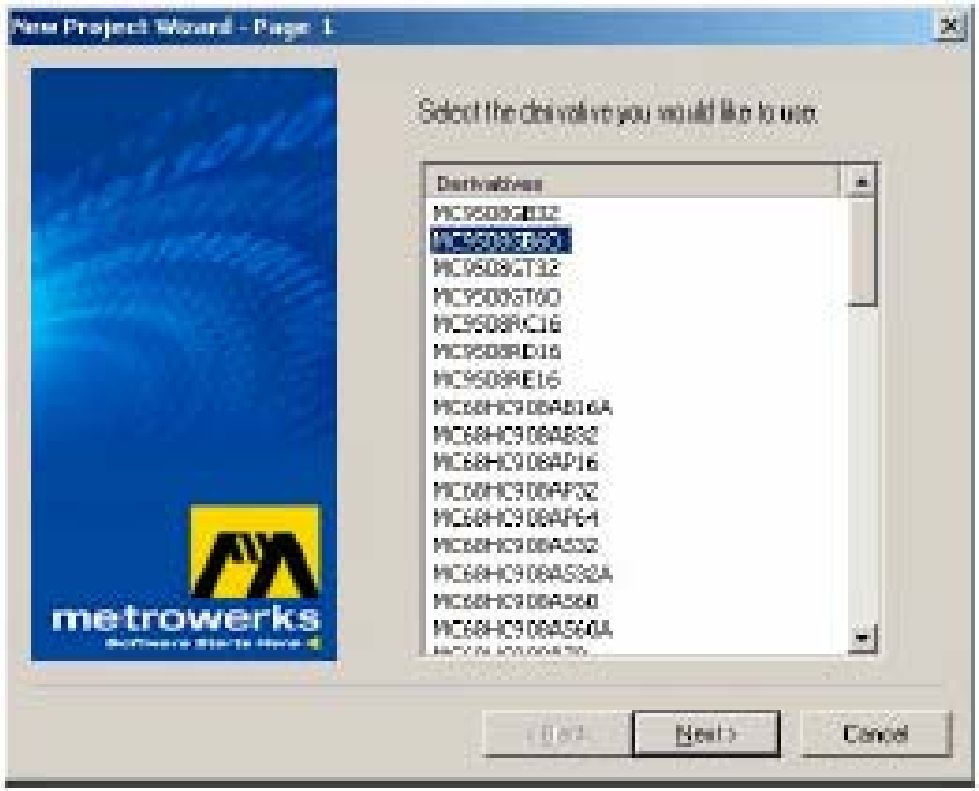
4. Once the root directory for the project has been located, enter the directory to be created (for example, **Project Directory**, as shown in the diagram above).
5. Select **Save**, and the new project directory / file name will be displayed.



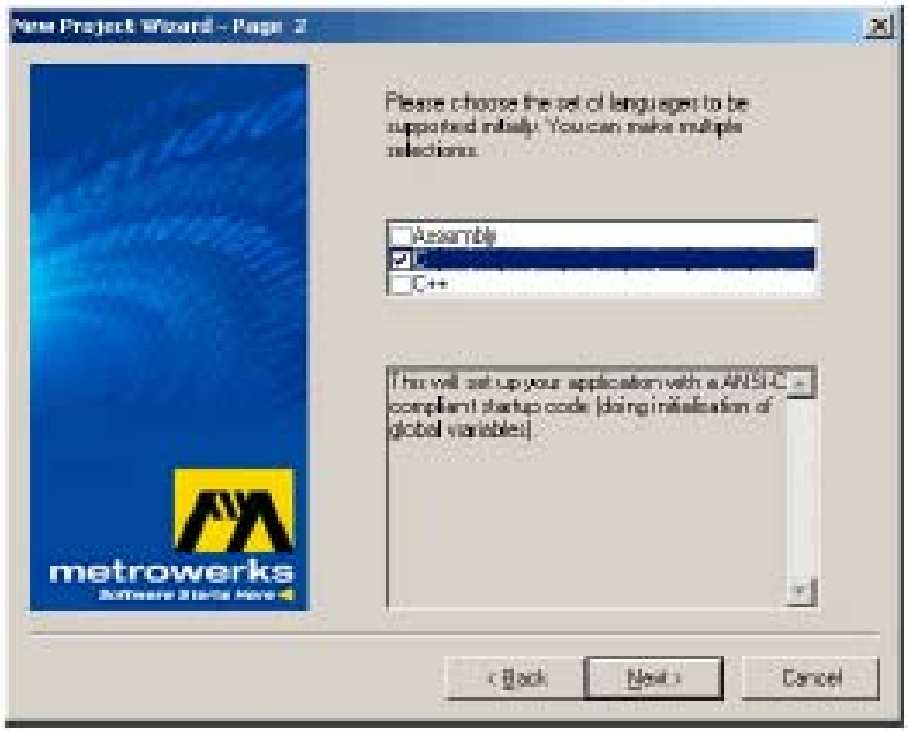
6. In the **Project name** field, you may enter the name of the project file to be created such as **Demo** (by default it will be the same as the directory name):
7. Select **OK**.



8. Device list will appear. Select **MC9S08QG8** and press **Next**:

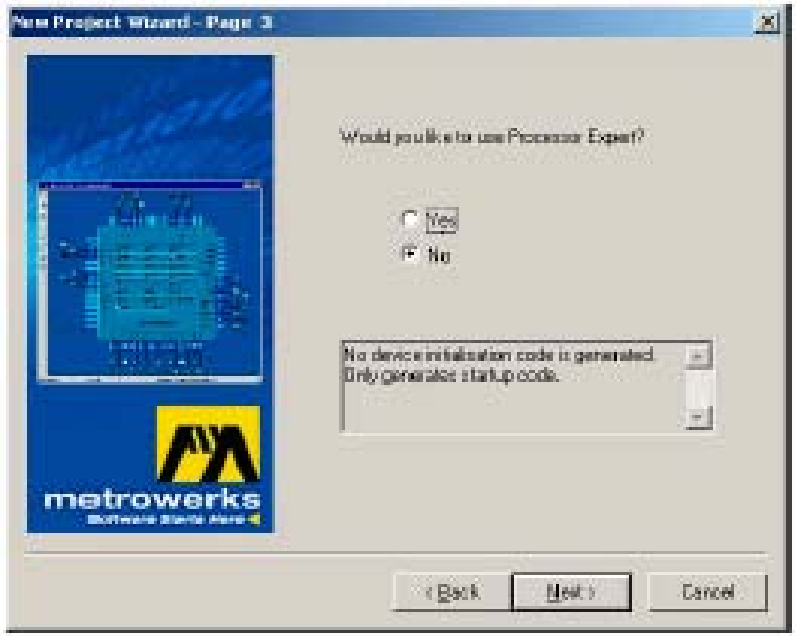


9. Language dialogue box will appear. Select **C** and press **Next**:

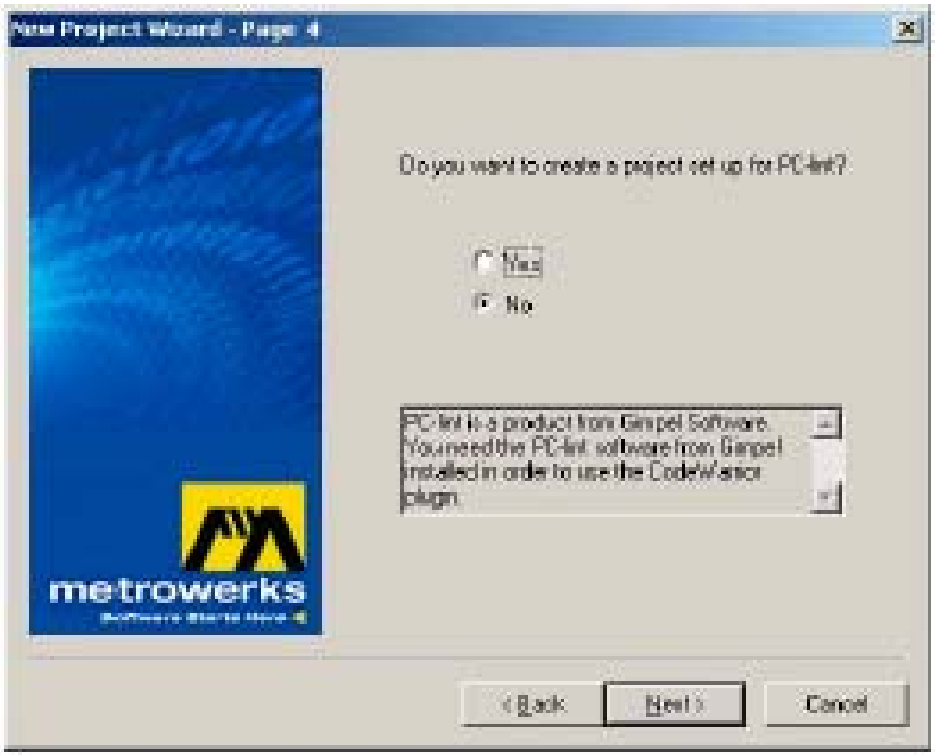


Lab 1 – CodeWarrior Project

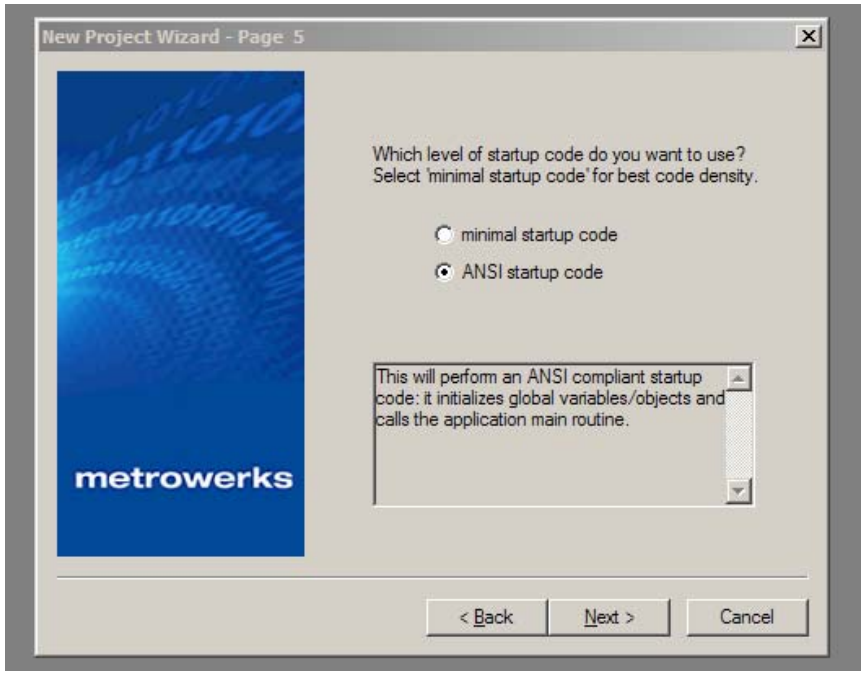
10. Processor expert dialog box will appear. Select **No** and press **Next.**
Note Processor Expert support for the QG8 has not been released.



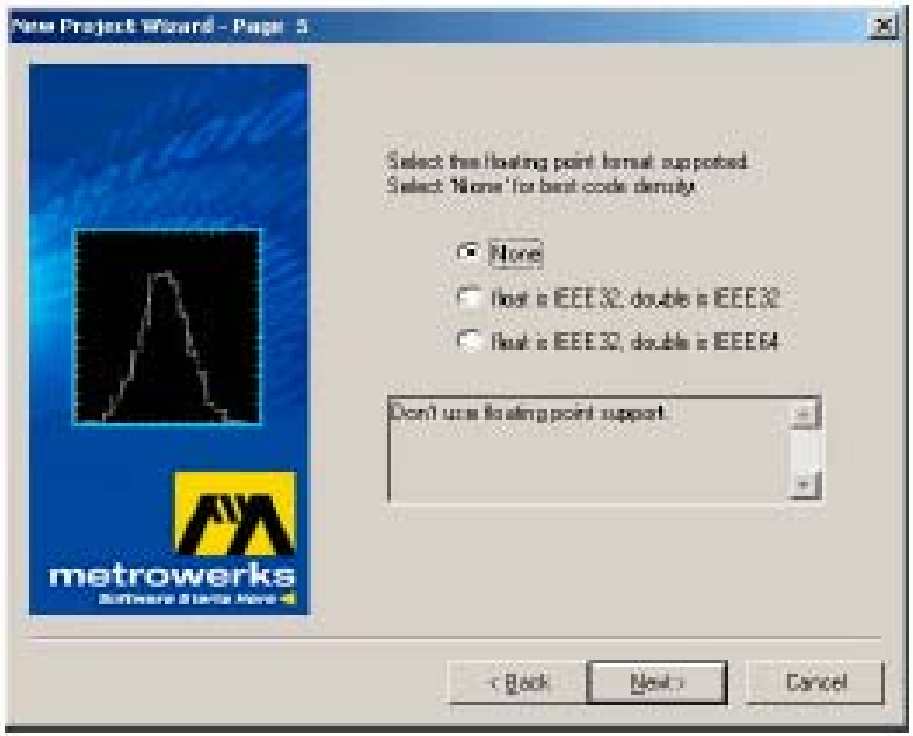
11. PC-lint option will appear. Select **No** and press **Next**:



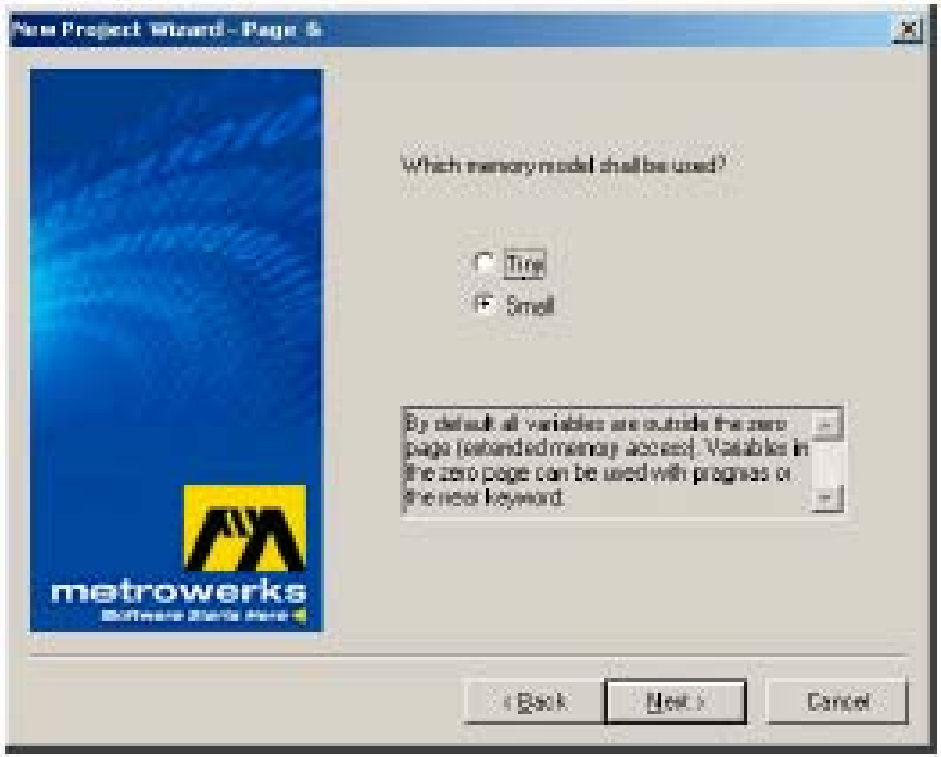
12. Startup code option will appear. Select **ANSI startup code** and press **Next**:



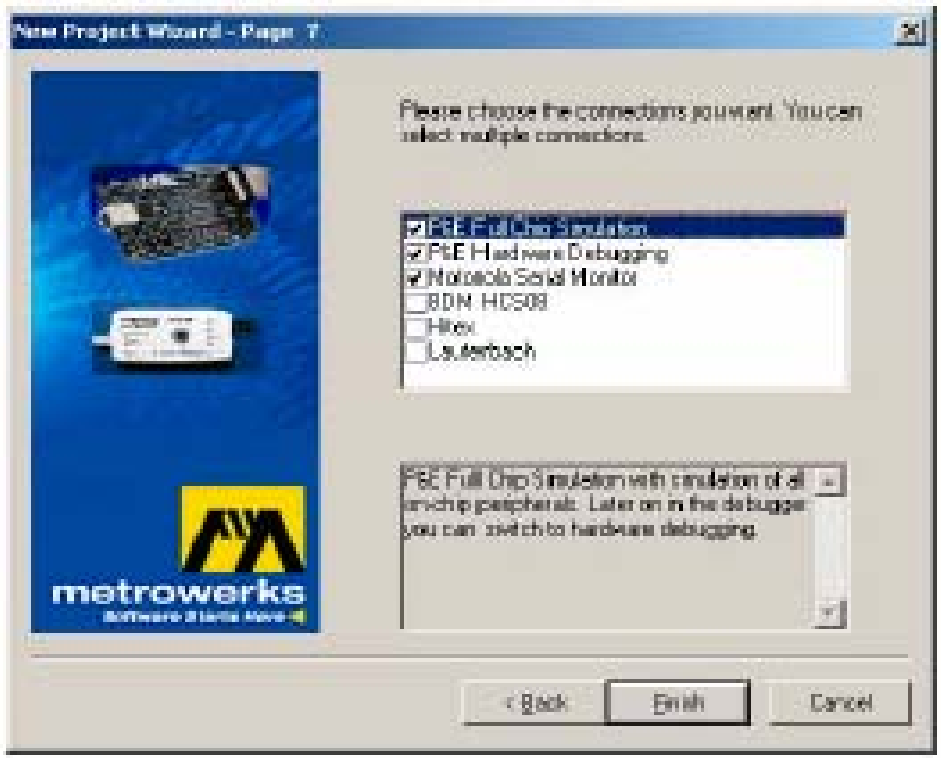
13. Floating point support dialog box will appear. Select **None** and press **Next**:



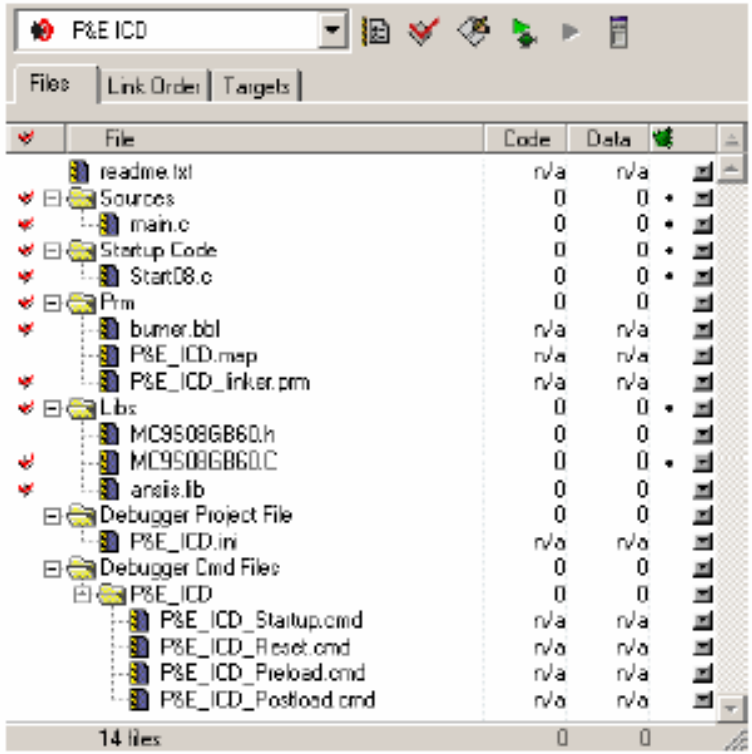
14. Memory model selection will appear. Select **Small** and press **Next**:



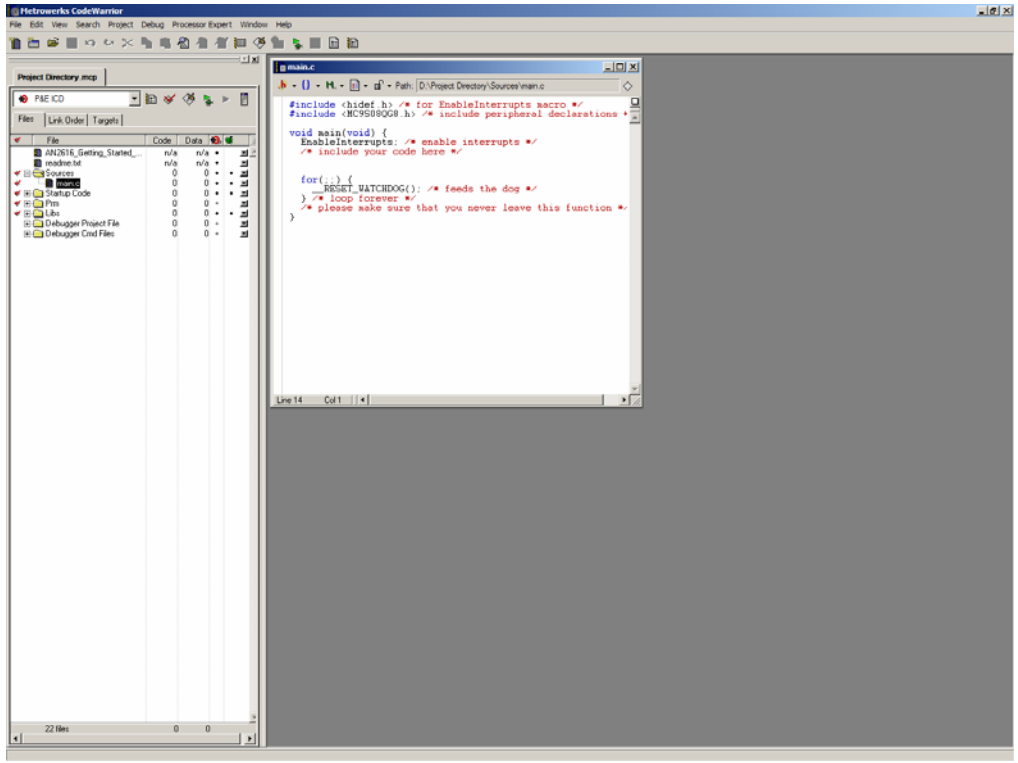
15. Connection dialog will appear. Select **P&E Full Chip Simulation** and **P&E Hardware Debugging**. Press **Finish** to create project:



16. Once the project wizard has finished, a project, including a skeleton application, has been created. The project window will look something like:

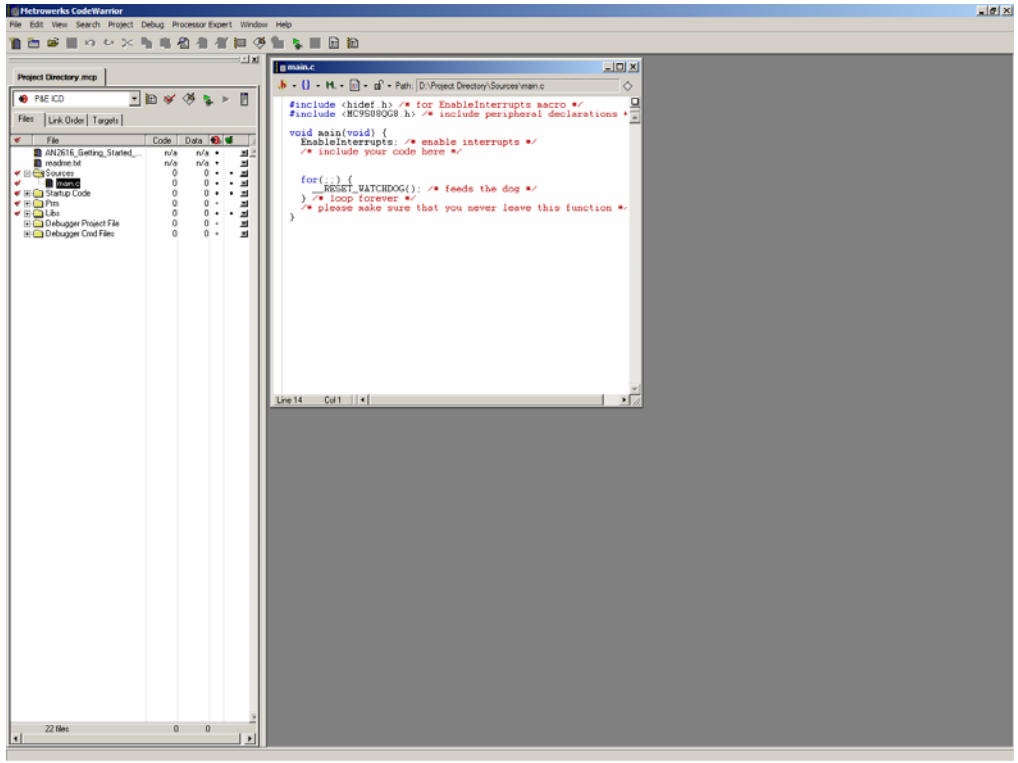


17. Click on **Sources**, **main.c**. At this point we are ready to insert application code:



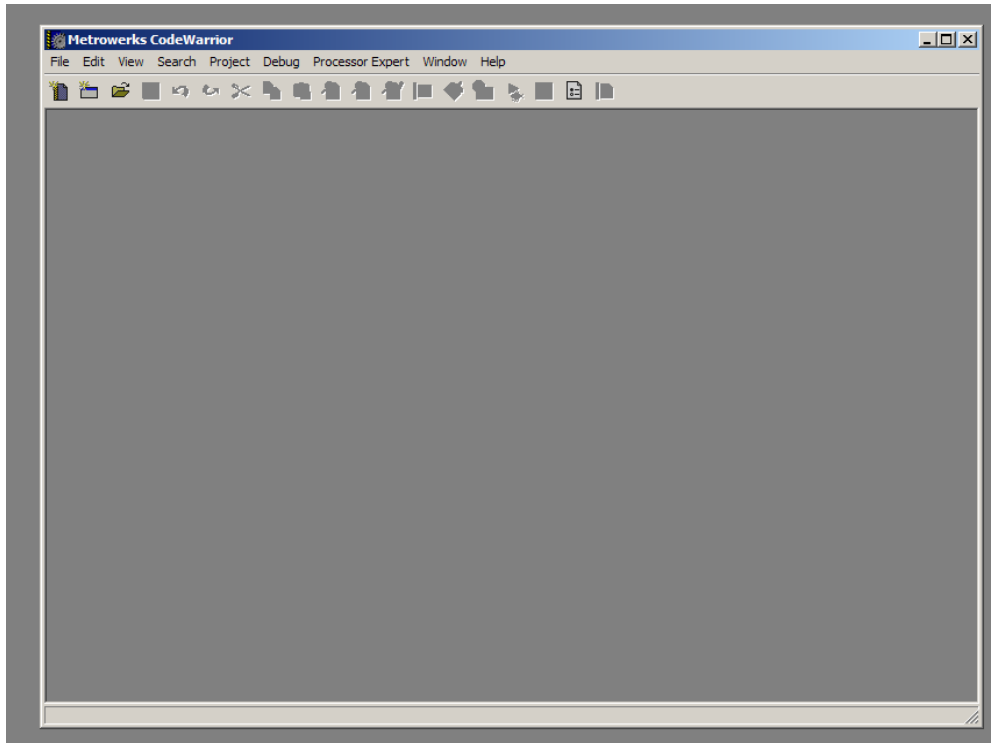
Lab 1 – CodeWarrior Project

18. This project could be compiled and run but it would not do much. Let's use the Demo_S08QG8_Test project to demonstrate debugging:

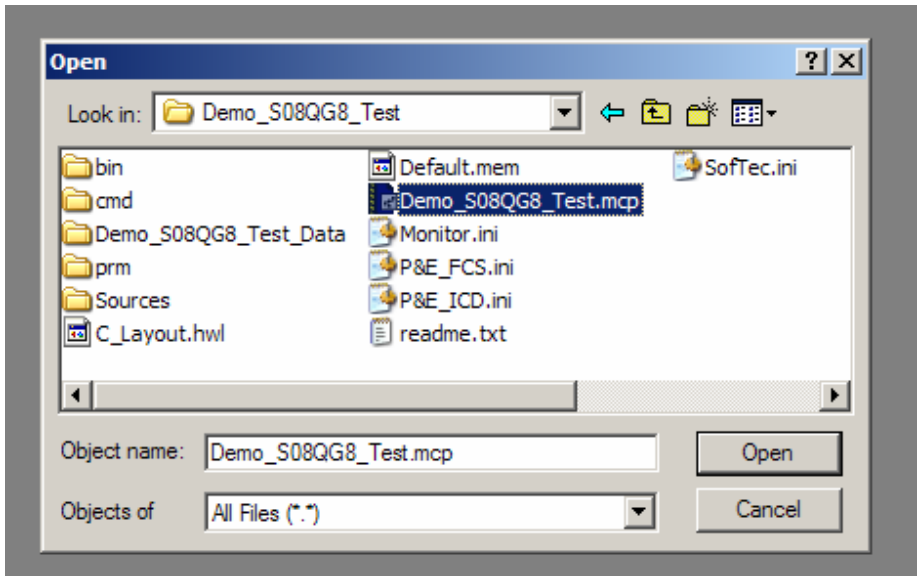


Lab 1 – CodeWarrior Project

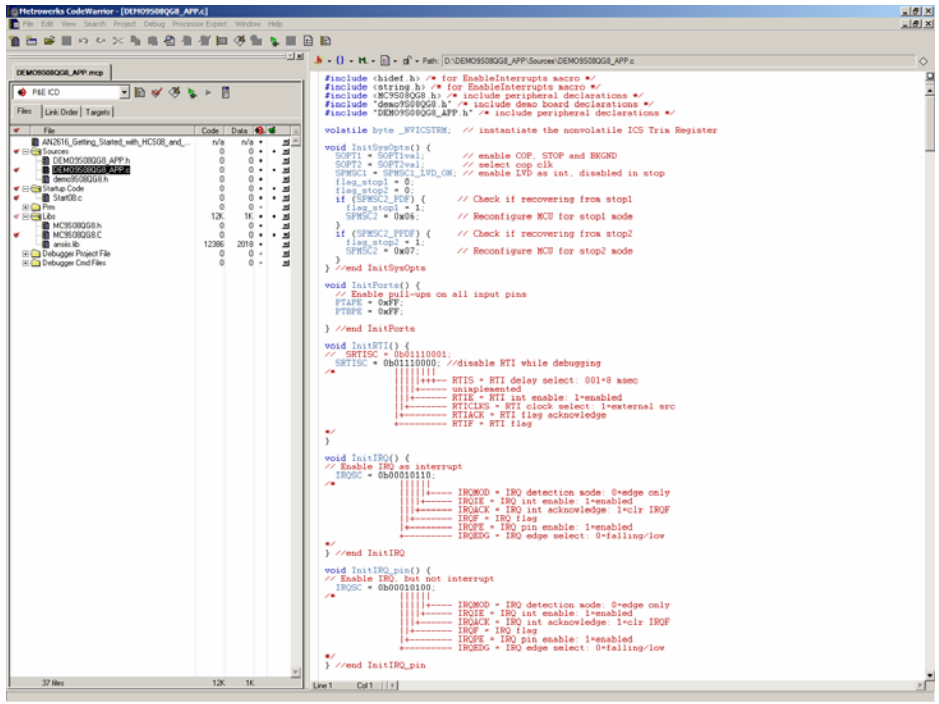
1. Close project files to start a new project:



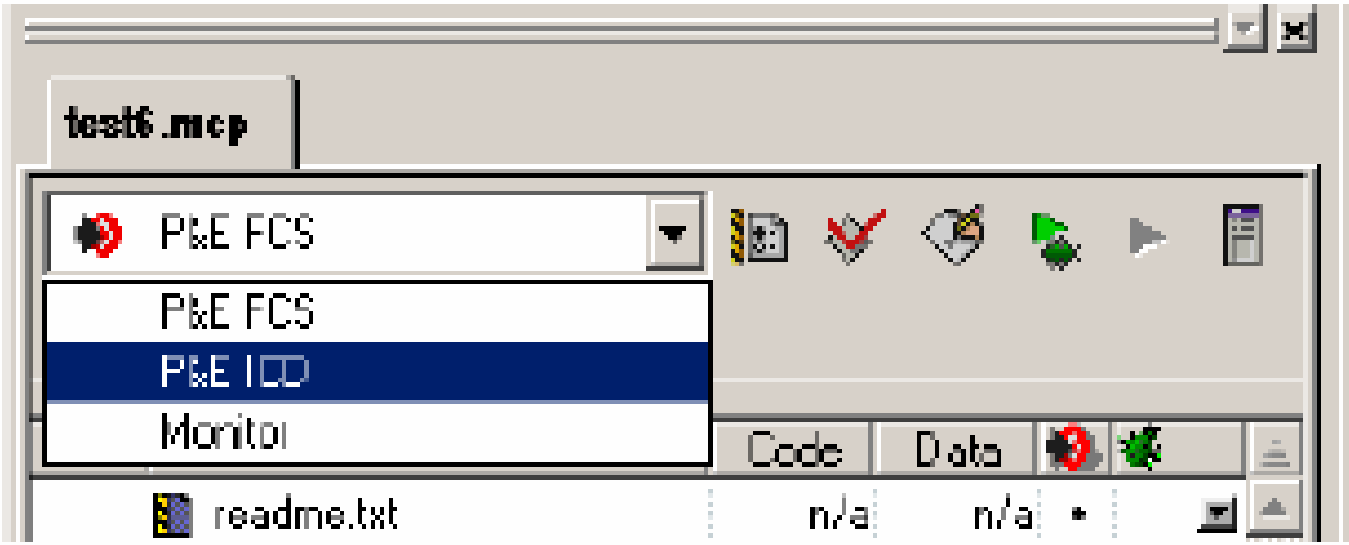
2. We will use the **Demo_S08_Test** application code to demonstrate debugging. Software is included on the Demo CD. Project files should be pre-installed on the Desktop. CodeWarrior project file convention is .mcp. Select **File**, **Open**, **Desktop\Demo_S08_Test\Demo_S08_Test.mcp**:



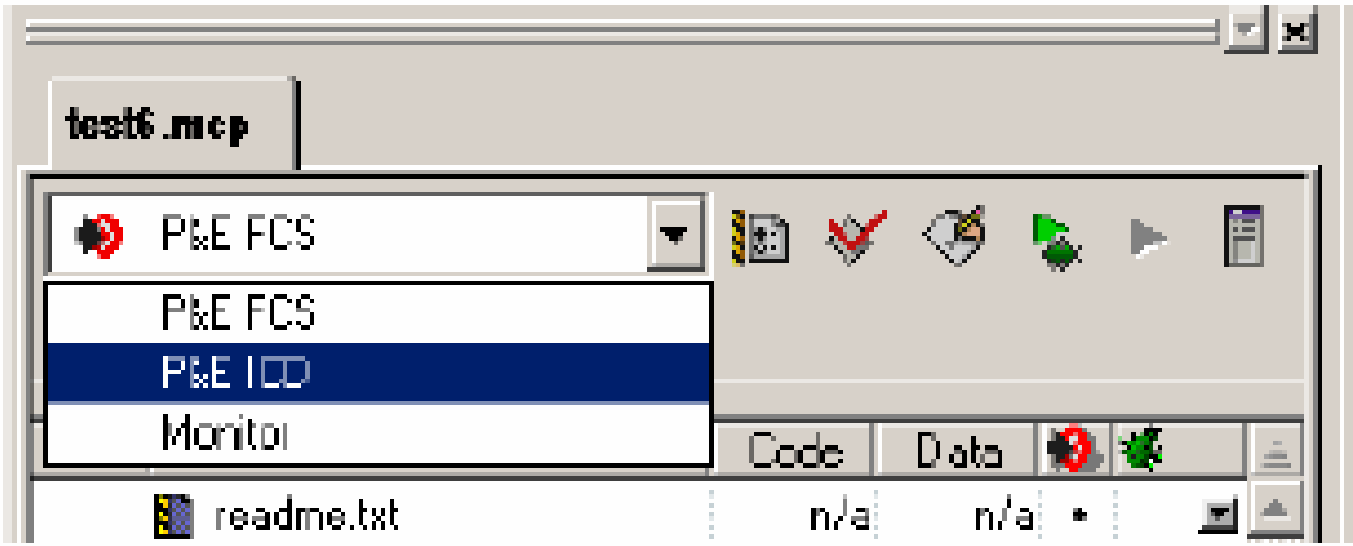
3. Project window will open. Source files can be opened up for editing:



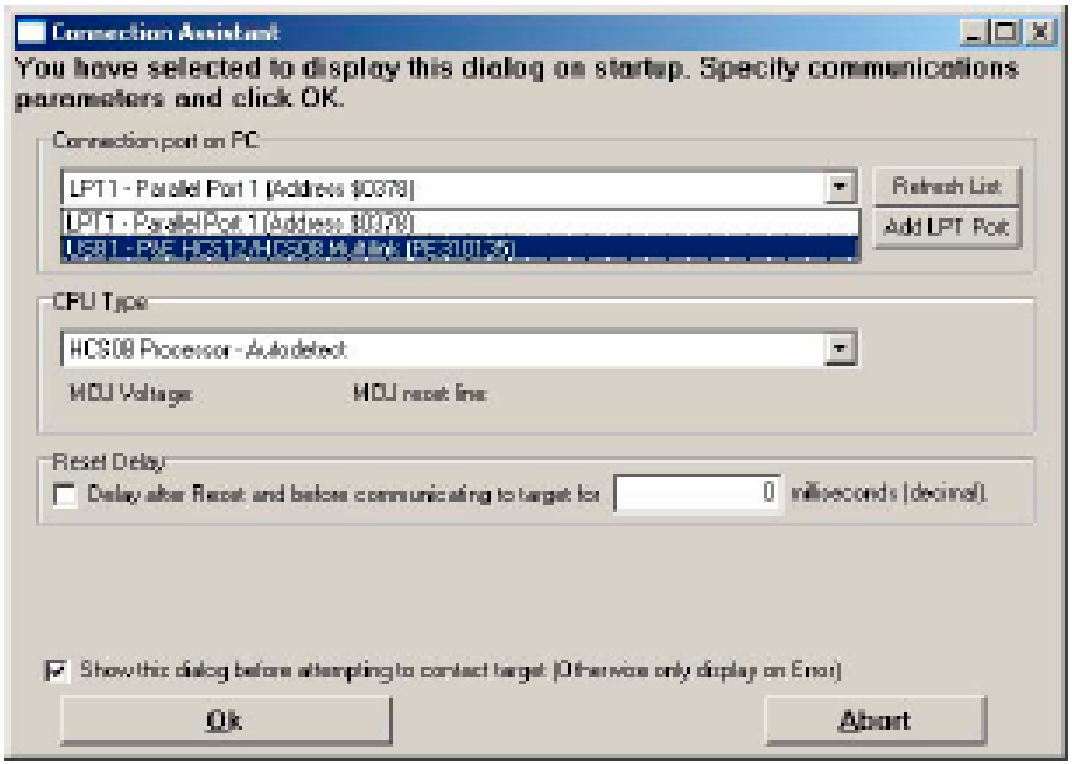
4. From the target connection pull-down, ensure that **P&E ICD** is selected:



5. Select Debug **GREEN ARROW**. This compiles and links the code, and invokes the True-Time simulator and Real-Time debugger (the HCS08 debugger). As this is the first time the code has been run, the debugger does not know which BDM device is to be used to connect to the target, so it pops up a dialog box for configuration:

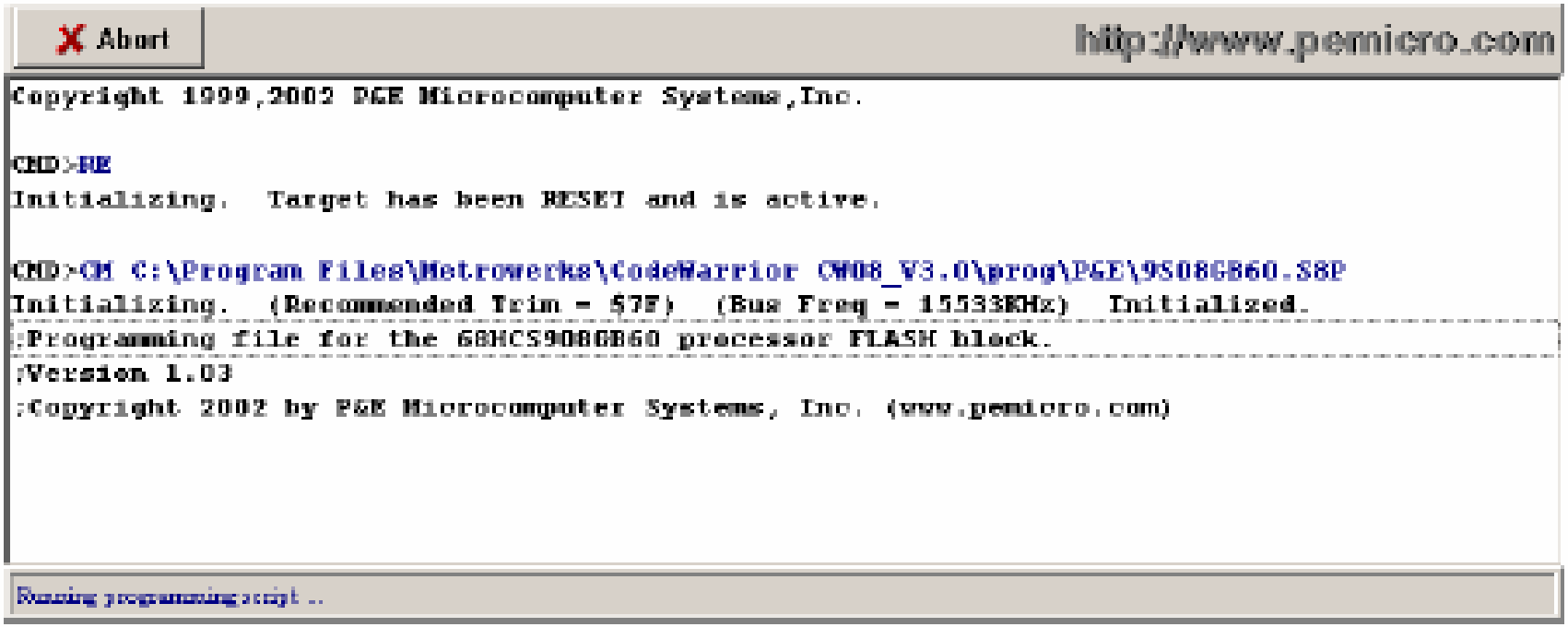


6. Ensure **USB HCS08/HCS12 Multilink - USB Port** is being used:



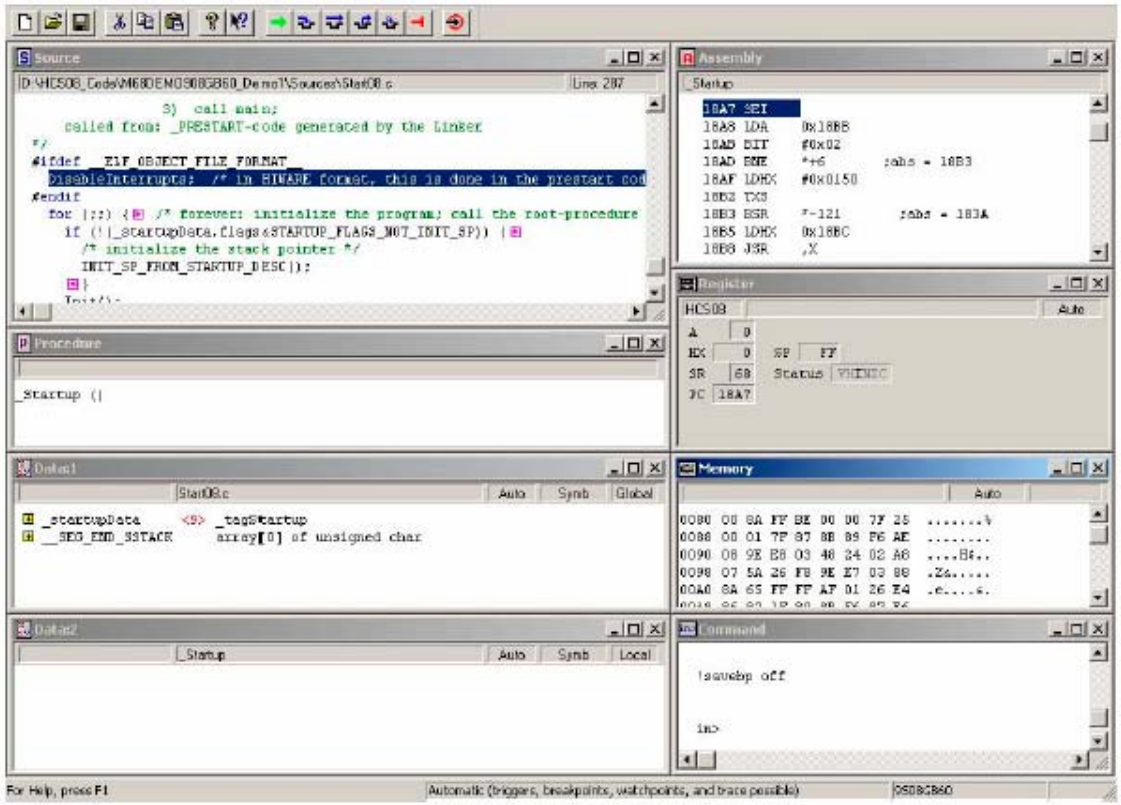
7. Next, a dialog pops up asking to erase and program FLASH. Press **Yes**.

8. Another window pops up showing the erasure and programming of the FLASH:

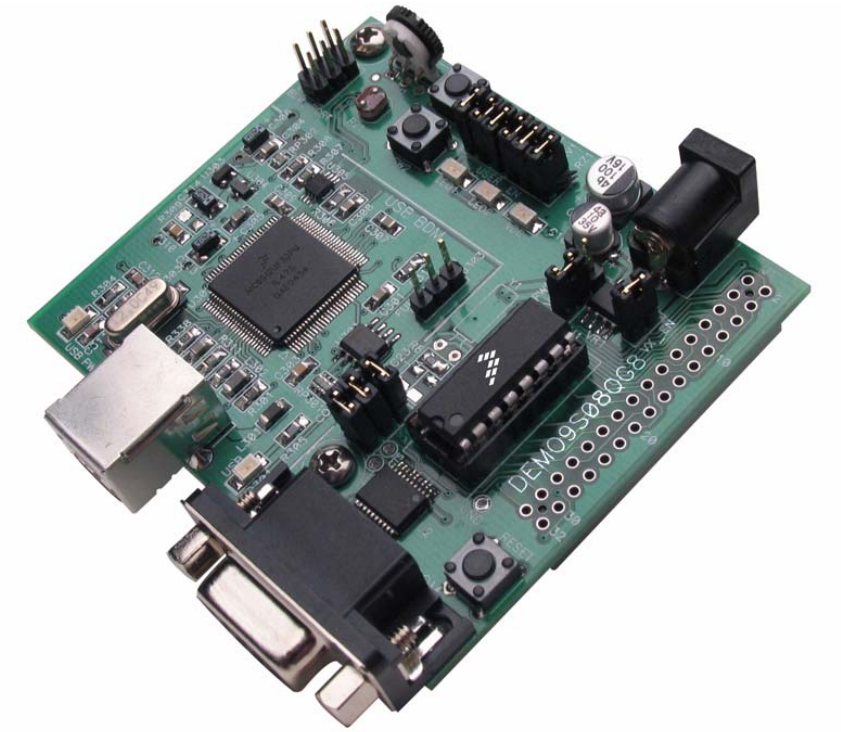


Lab 1 – CodeWarrior Debugging

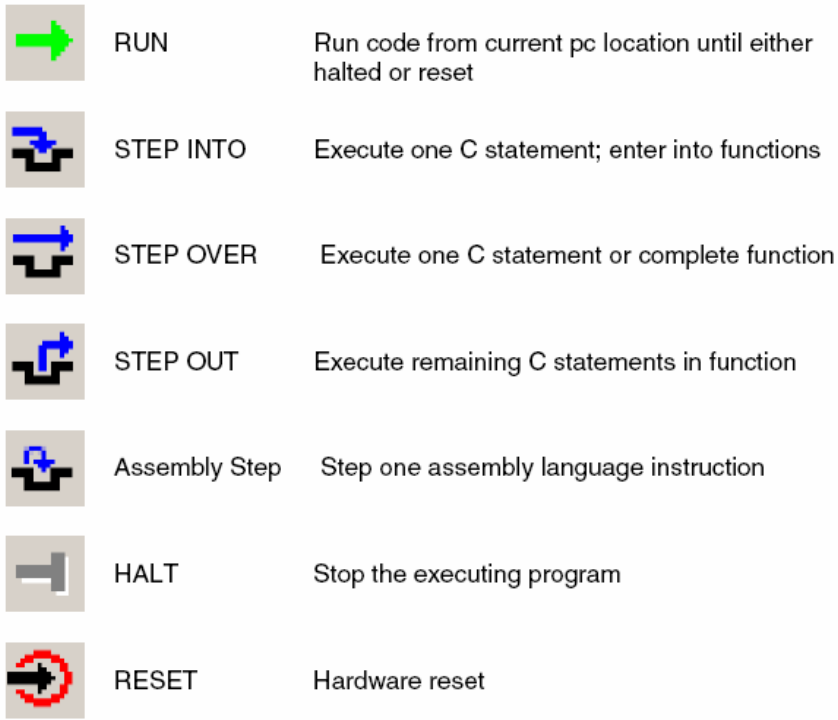
9. Once BDM communication has been established with the device, the debugger will show its debug window. Press Start/Continue **GREEN ARROW** and code should execute:



10. Application should be running. LED2 will flash continuously. Pressing SW1 toggles LED1.



11. Use Run Time controls to demonstrate best-in-class BDM debugging capacities:



RUN	Run code from current pc location until either halted or reset
STEP INTO	Execute one C statement; enter into functions
STEP OVER	Execute one C statement or complete function
STEP OUT	Execute remaining C statements in function
Assembly Step	Step one assembly language instruction
HALT	Stop the executing program
RESET	Hardware reset

Lab 2 – Example Application

This lab code is intended to introduce the DEMO9S08QG8 demo board. Connecting a serial cable from the DEMO9S08QG8 to a PC with a Terminal emulation program running allows the user to interact with the MCU and try out several of the features.

- **Demo Code Functions**
  1: ICS setup
  2: TPM PWM setup
  3: MTIM setup
  4: RTI setup
  5: STOP test
  6: ACMP test
  7: ADC test

Lab 2 – Example Application

Requirements

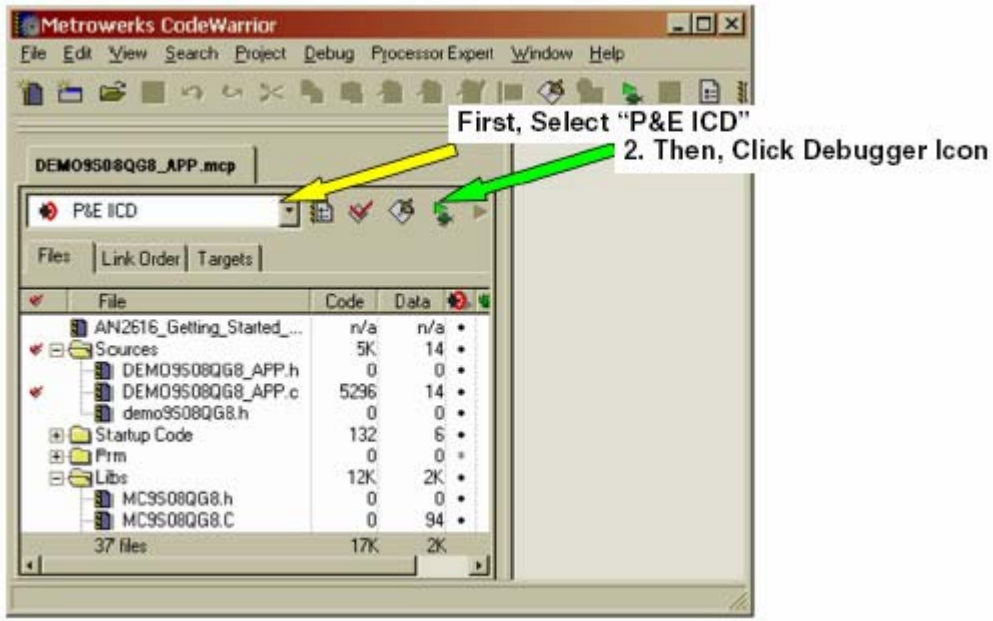
- DEMO9S08QG8 demo board with an MC9S08QG8 in the 16 DIP socket
- DB9 serial cable
- CodeWarrior Development Studio for HC(S)08– CodeWarrior Service Packs for MC9S08QG8 and USB (download from www.metrowerks.com)
- DEMO9S08_APP.mcp Demo Project on CD.
- Terminal emulation program, e.g., HyperTerminal

Setting up the Demo

1. Connect the DEMO9S08QG8 to the PC as described in the Quick Start Guide. The jumper configuration for this application is:
   - > COM_EN jumper removed
   - > SW1, SW2, LED1, RV1, RZ1 jumpers installed
   - > LED2 jumper optional
2. Connect the DB9 serial cable and the USB cable between the DEMO9S08QG8 and the PC.
3. Open CodeWarrior IDE and close any open projects.
4. Select **File**, **Open**:
   Desktop\Demo_9S08_App\Demo_9S08_App.mcp

Setting up the Demo

4. Compile for the P&E ICD target and program the MCU by selecting “P&E ICD” from the pull-down menu beneath the project name.



HyperTerminal Terminal Settings

- 4800 Baud
- 8 Data Bits
- No parity
- 1 Stop Bit
- No Flow Control

Running the Demo

If everything is configured correctly, the terminal window will display the following as soon as the MCU is running:

```
Welcome to the MC9S08QG8 Demo Application
V1.0

Main Test Menu:
  1: ICS setup
  2: TPM PWM setup
  3: MTIM setup
  4: RTI setup
  5: STOP test
  6: ACMP test
  7: ADC test

Select number to execute:
```

Lab 3 – Internal Clock Source

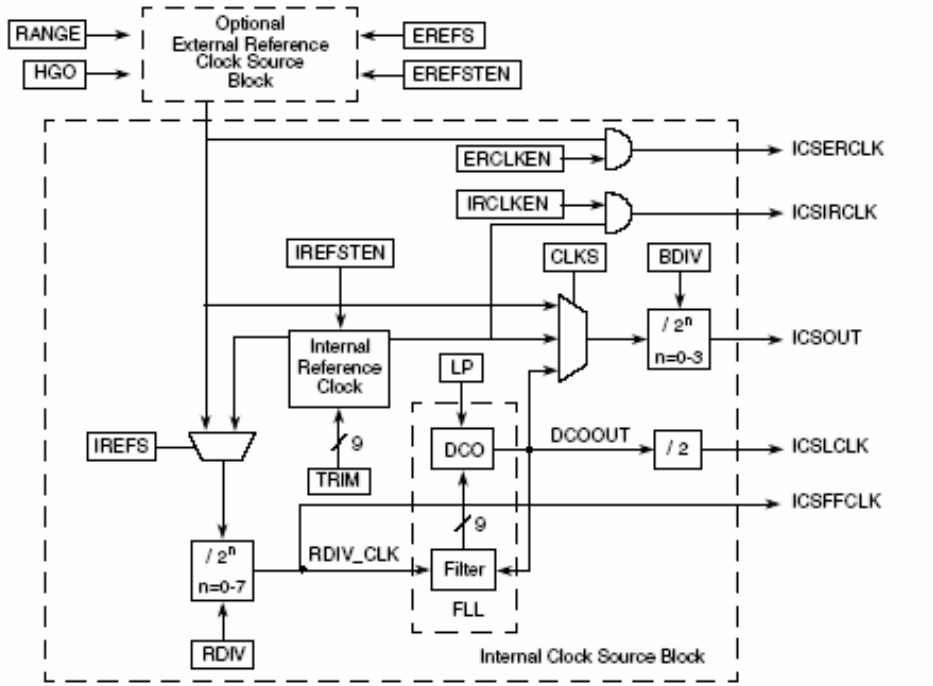


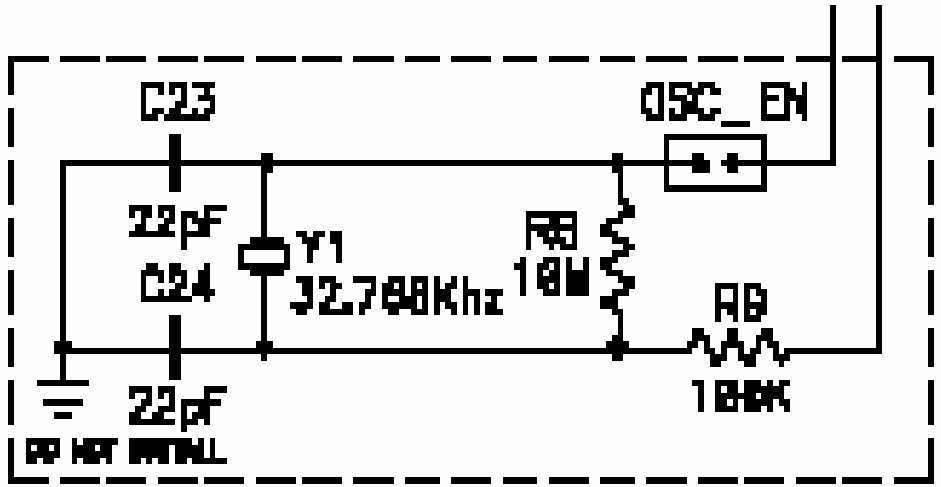
Figure 10-2. Internal Clock Source (ICS) Block Diagram

16-MHz **Internal Clock Source** (ICS), Low Power Oscillator 31.25 kHz to 38.4 kHz or 1 MHz to 16 MHz (XOSC)

Lab 3 – Internal Clock Source

- Internal Clock Source (ICS) Features
  - Frequency-locked loop (FLL) is trimmable for accuracy
    - > 0.2% resolution using internal 32-kHz reference
    - > 2% deviation over voltage and temperature using internal 32-kHz reference
  - Internal or external reference clocks up to 5-MHz can be used to control the FLL
    - > 3 bit select for reference divider is provided
  - Internal reference clock has 9 trim bits available
  - Internal or external reference clocks can be selected as the clock source for the MCU
  - Whichever clock is selected as the source can be divided down
    - > 2 bit select for clock divider is provided
    - > Allowable dividers are: 1, 2, 4, 8
    - > BDC clock is provided as a constant divide by 2 of the DCO output
  - Control signals for a low power oscillator as the external reference clock are provided
    - > HGO, RANGE, EREFS, ERCLKEN, EREFSTEN
  - FLL engaged internal mode is automatically selected out of reset

External Crystal Circuit



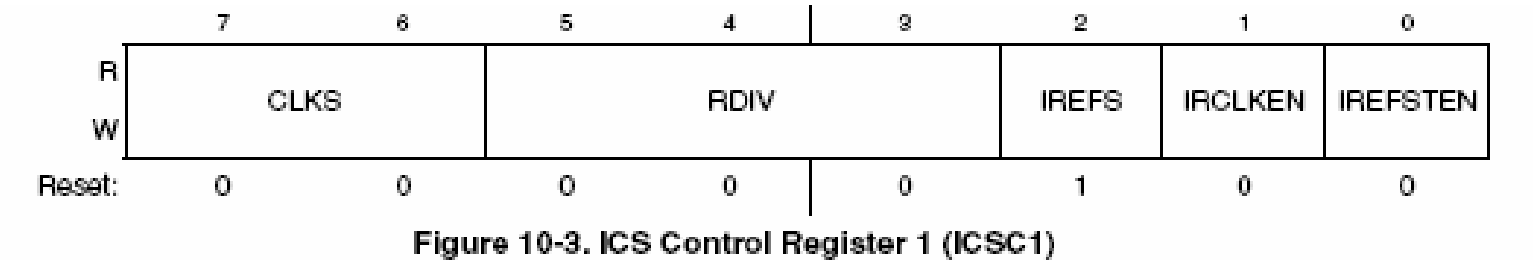
- Note external crystal components are not stuffed on Demo Board.

Lab 3 – Internal Clock Source

- ICS's seven modes of operation
  - FLL Engaged Internal (FEI): This is the default mode of operation. The ICS supplies a clock derived from the FLL which is controlled by the internal reference clock.
  - FLL Bypassed Internal (FBI): In this mode the FLL is enabled and controlled by the internal reference clock, but it is bypassed. The ICS supplies a clock derived from the internal reference clock.
  - FLL Bypassed Internal Low Power (FBILP): In this mode the FLL is disabled and bypassed, and the ICS supplies a clock derived from the internal reference clock. The BDC clock is not available.
  - FLL Engaged External (FEE): In this mode the ICS supplies a clock derived from the FLL which is controlled an external reference clock.

- ICS's seven modes of operation (continued)
  - FLL Bypassed External (FBE): In this mode the FLL is enabled and controlled by an external reference clock, but it is bypassed. The ICS supplies a clock derived from the external reference clock.
  - FLL Bypassed External Low Power (FBELP): In this mode the FLL is disabled and bypassed, and the ICS supplies a clock derived from the external reference clock. The BDC clock is not available.
  - Stop: In stop mode the FLL is disabled and the internal or external reference clocks can be selected to be enabled or disabled. The BDC clock is not available and the ICS does not provide an MCU clock source.

Lab 3 – Internal Clock Source



	7	6	5	4	3	2	1	0
R W	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
Reset:	0	0	0	0	0	1	0	0

Figure 10-3. ICS Control Register 1 (ICSC1)

ICS Control Register 1

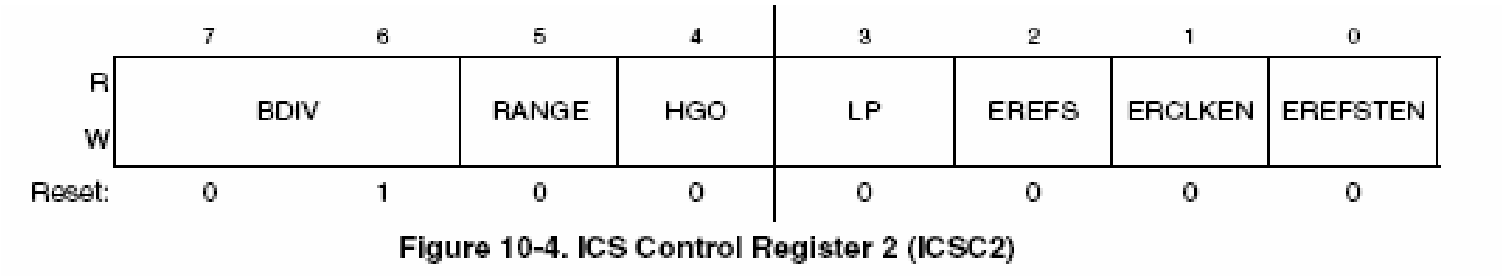
CLKS bits select the clock source that controls the bus frequency.

RDIV bits select the amount to divide down the FLL reference clock.

IREFS selects between using the internal reference clock (1) or an external reference clock (0)

IRCLKEN enables (1) or disables (0) the internal reference clock for use as ICSIRCLK

IREFSTEN controls whether the internal reference clock remains enabled (1) or not (0) when the ICS enter stop mode

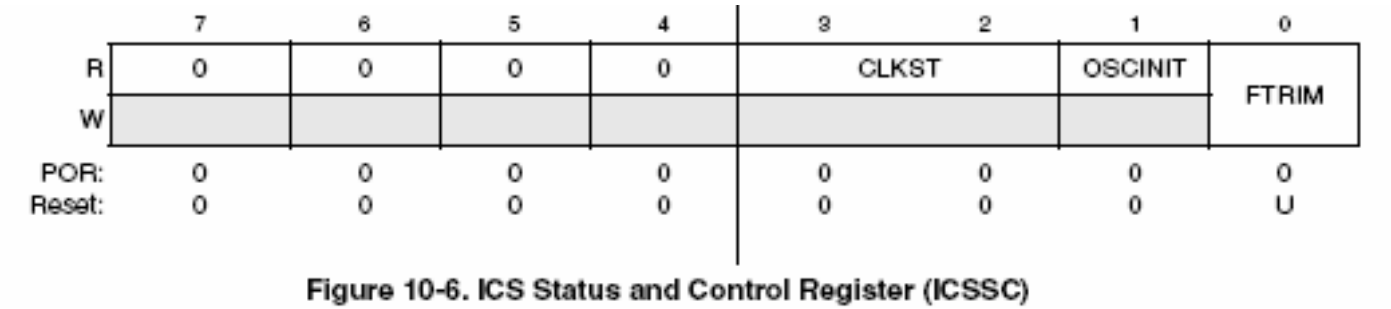


	7	6	5	4	3	2	1	0
R W	BDIV		RANGE	HGO	LP	EREFS	ERCLKEN	EREFSTEN
Reset:	0	1	0	0	0	0	0	0

Figure 10-4. ICS Control Register 2 (ICSC2)

ICS Control Register 2

BDIV bits select the amount to divide down the clock source selected by the CLKS bits
RANGE selects high frequency (1) or low frequency (0) range for the external oscillator
HGO configures the external oscillator for high gain (1) or low power (0) operation
LP controls whether the FLL is disabled (1) or not (0) in FLL bypassed modes
EREFS selects whether an oscillator (1) or an external clock source (0) is used for the external reference clock
ERCLKEN enables (1) or disables (0) the external reference clock for use as ICSERCLK
EREFSTEN controls whether the external reference clock remains enabled (1) or not (0) when the ICS enter stop mode



	7	6	5	4	3	2	1	0
R	0	0	0	0	CLKST		OSCINIT	FTRIM
W								
POR:	0	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	U

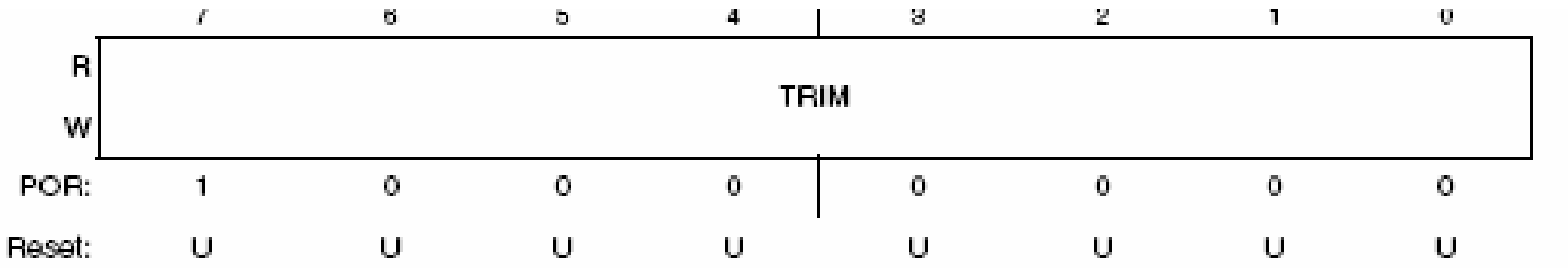
Figure 10-6. ICS Status and Control Register (ICSSC)

ICS Status and Control Register

CLKST bits indicate the current clock mode

OSCINIT this bit indicates if the initialization cycles of the external oscillator clock have completed

FTRIM this bit controls the smallest adjustment of the internal reference clock frequency



	7	6	5	4	3	2	1	0
R W	TRIM							
POR:	1	0	0	0	0	0	0	0
Reset:	U	U	U	U	U	U	U	U

Figure 10-5. ICS Trim Register (ICSTRM)

ICS Trim Register

TRIM bits control the internal reference clock frequency by controlling the internal reference clock period (the bits' effect is binary weighted). Increasing the binary value in TRIM will increase the period, and decreasing the value will decrease the period. An additional trim bit is available in the ICSSC register.

Lab 3 – Internal Clock Source

Select Option 1: ICS setup

Main Test Menu:
1: ICS setup
2: TPM PWM setup
3: MTIM setup
4: RTI setup
5: STOP test
6: ACMP test
7: ADC test

Select number to execute: 1

Lab 3 – Internal Clock Source

Select Option 2: ICS in FEI, BDIV 2

ICS Setup Menu:
1: ICS in FEI, BDIV=1 (set BAUD to 4800 bps) (8 MHz bus)
2: ICS in FEI, BDIV=2 (set BAUD to 4800 bps) (4 MHz bus)
3: ICS in FEI, BDIV=4 (set BAUD to 4800 bps) (2 MHz bus)
4: ICS in FEI, BDIV=8 (set BAUD to 4800 bps) (1 MHz bus)
5: ICS in FBI (set BAUD to 110bps) (16 kHz bus)
6: ICS in FBILP (set BAUD to 110bps) (16 kHz bus)

Select test number to execute: 2

FEI – FLL Engaged Internal Clock
FBI – FLL Bypass Internal Clock LP – Low Power

Lab 3 – Internal Clock Source

ICS Application Code

```
// load trim value if location not blank
if (NVICSTRM != 0xFF) {
   ICSTRM = NVICSTRM;
   }

// output of FLL is selected, reference divider 1, internal reference clock selected
ICSC1 = 0x04;

// bus divider 2
ICSC2 = 0x40
```

Internal reference clock, FLL output, BDIV = 2,

fbus = (firc * 512 / BDIV) / 2

fbus = (32 kHz * 512 / 2) / 2 = 4.1 MHz bus

Note above settings are same as register default values. Optionally use scope to monitor PWM output on J1-23.

- **ICS Design Tips**
  - When ICS is configured for FEE or FBE mode, input clock source must be divisible using RDIV to within the range of 31.25 kHz to 39.0625 kHz.
  - Check the External and Internal Oscillator Characteristics Appendix for electrical and timing specifications.
  - The external oscillator can be configured to provide a higher amplitude output for noise immunity. This mode of operation is selected by HGO=1.
  - When switching modes of operation, if the newly selected clock is not available, the previous clock will remain selected.
  - The TRIM and FTRIM value will not be affected by a reset.

Lab 4 – Modulo Timer Module

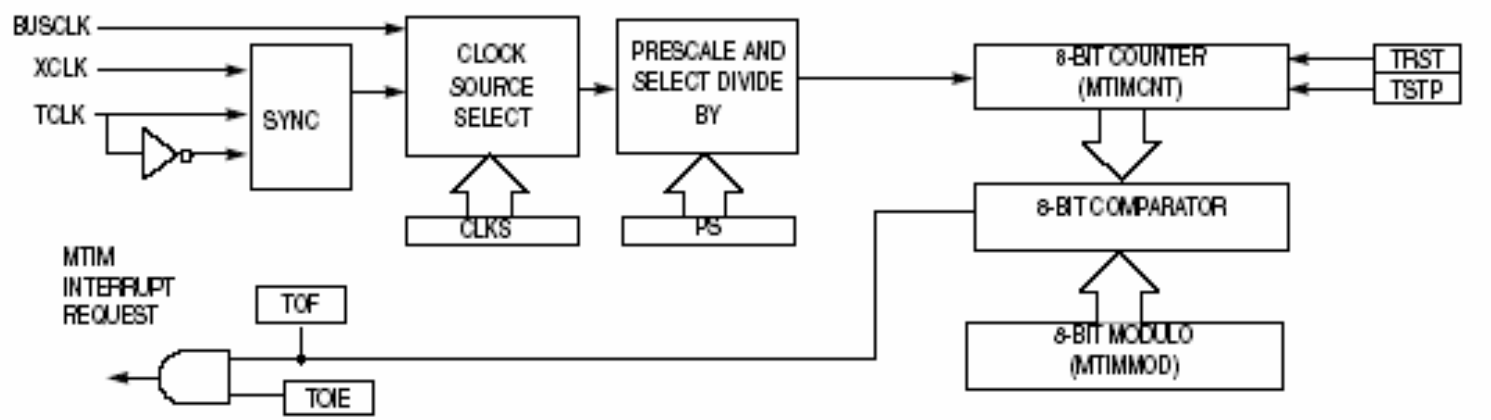
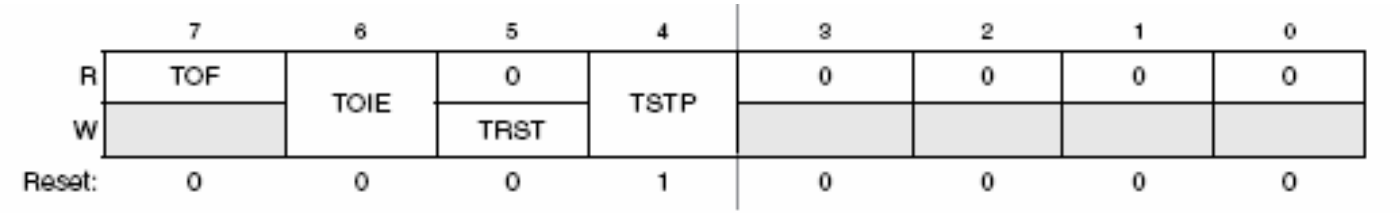


Figure 13-2. Modulo Timer (MTIM) Block Diagram

Lab 4 – Modulo Timer Module

- Modulo Timer features include:
  - 8-bit up-counter
  - Free-running or 8-bit modulo limit
  - Software controllable interrupt on overflow
  - Counter reset bit (TRST)
  - Counter stop bit (TSTP)
- Four software selectable clock sources for input to prescaler:
  - System bus clock — rising edge
  - Fixed frequency clock (XCLK) — rising edge
  - External clock source on the TCLK pin — rising edge
  - External clock source on the TCLK pin — falling edge
- Nine selectable clock prescale values:
  - Clock source divide by 1, 2, 4, 8, 16, 32, 64, 128, or 256



	7	6	5	4	3	2	1	0
R	TOF	TOIE	0	TSTP	0	0	0	0
W			TRST					
Reset:	0	0	0	1	0	0	0	0

Figure 13-3. MTIM Status and Control Register

MTIM Status and Control Register

MTIMSC contains the overflow status flag and control bits, which are used to configure the interrupt enable, reset the counter, and stop the counter.

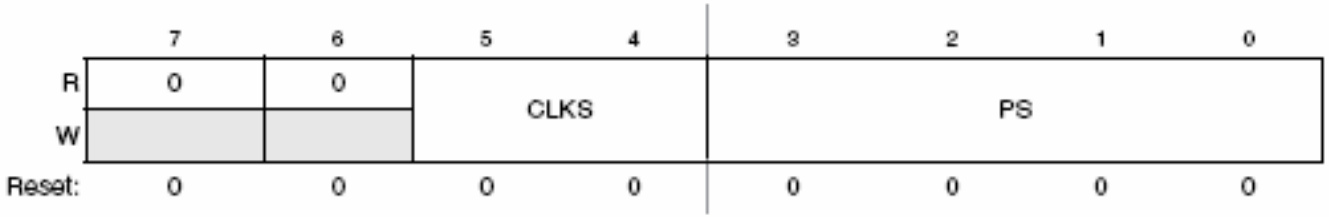


Figure 13-4. MTIM Clock Configuration Register

MTIM Clock Configuration Register

MTIMCLK contains the clock select bits (CLKS) and the prescaler select bits (PS).

Select Option 3: MTIM setup

Main Test Menu:
1: ICS setup
2: TPM PWM setup
3: MTIM setup
4: RTI setup
5: STOP test
6: ACMP test
7: ADC test

Select number to execute: 3

Lab 4 – Modulo Timer Module

MTIM Setup

Set Prescaler, Modulo and Clock Source as shown in “ “:

MTIM Setup:

1: 100X PWM from busclk
2: Set prescalar “0”
3: Set modulo “FF”
4: Set clock source “XLCK”
5: Start MTIM
6: Stop MTIM
7: Reset MTIM
e: Exit MTIM menu

Select test number to execute:

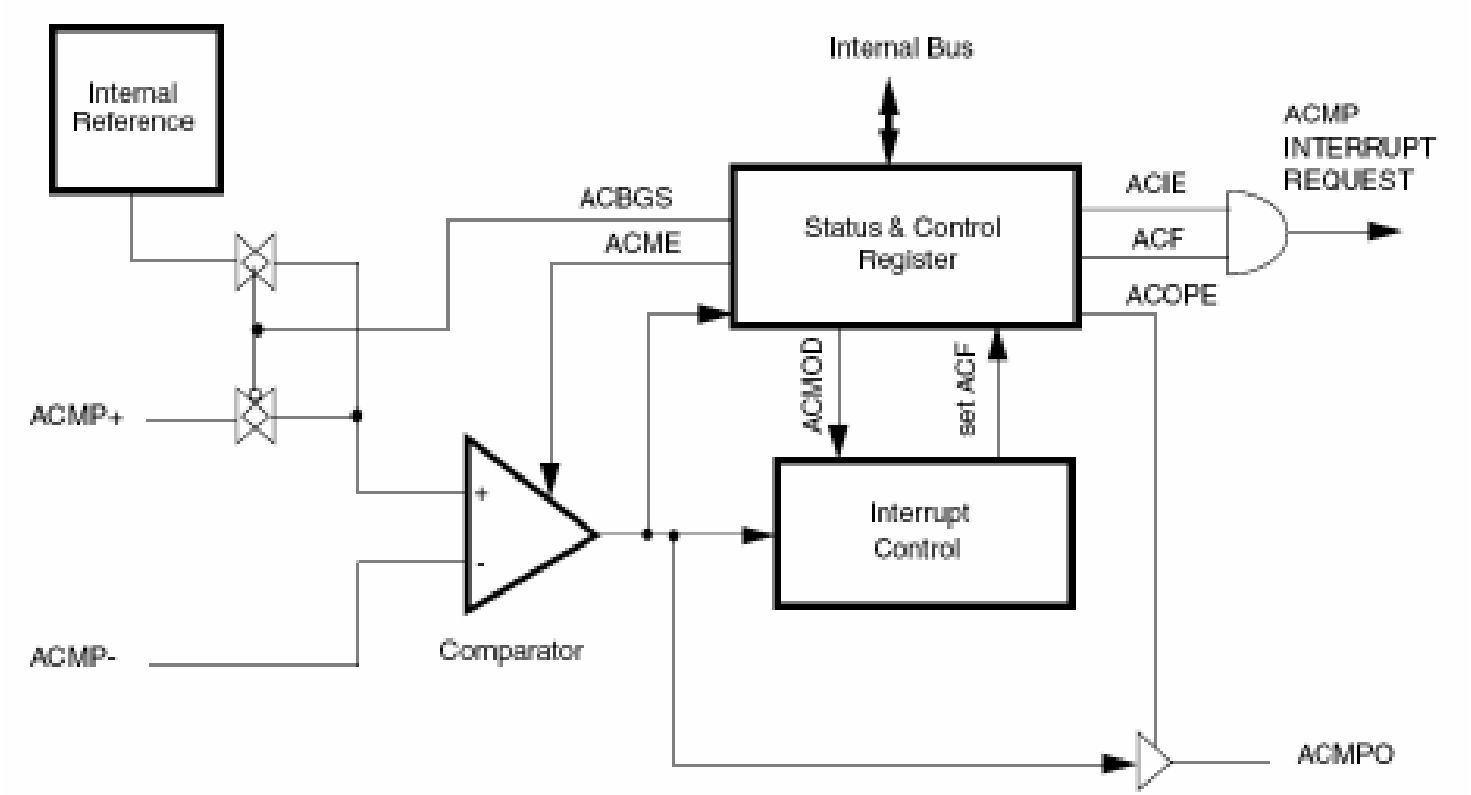
Lab 4 – Modulo Timer Module

MTIM will toggle LED1 at ~ 32 Hz

LED Toggle = Modulo / (XCLK / Prescaler)
LED Toggle = 255/(16,000/1) = 16 msec

Change LED blink rate with different Prescaler and Modulo values.

Lab 5 – Analog Comparator



Lab 5 – Analog Comparator

- ACMP has the following features:
  - Full rail-to-rail supply operation.
  - Less than 40 mV of input offset.
  - Less than 15 mV of hysteresis.
  - Selectable interrupt on rising edge, falling edge, or either rising or falling edges of comparator output.
  - Option to compare to fixed internal bandgap reference voltage.
  - Option to allow comparator output to be visible on a pin, ACMPO.

1. Configure the Analog Comparator Register (ACMPSC)

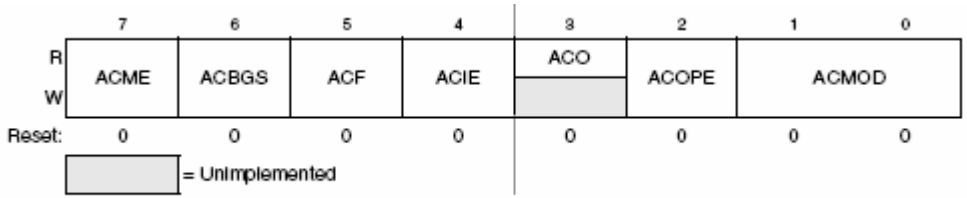


Figure 8-3. ACMP Status and Control Register

2. Declare ACMP interrupt Service Routine
3. From this point on, the code execution is performed inside the ACMP interrupt service routine. The code inside needs to Clear ACMP interrupt flag.

Select Option 6: ACMP test

Main Test Menu:
1: ICS setup
2: TPM PWM setup
3: MTIM setup
4: RTI setup
5: STOP test
6: ACMP test
7: ADC test

Select number to execute: 6

Lab 5 – Analog Comparator

Select Option 1: Enable ACMP

Analog Comparator Menu:
1: Enable ACMP
2: Disable ACMP

Select test number to execute: 1
Adjust RV1 or RZ1 to trigger comparator
Press SW1 or SW2 on DEMO9S08QG8 to exit

Lab 5 – Analog Comparator

- When enabled, the ACMP (Analog Comparator) test configures the ACMP to compare the voltage on the photosensor (RZ1) to the potentiometer (RV1).
- The output of the ACMP is configured by software to be the trigger for the TPMCH0 input capture. The demo measures the time between each toggle of the ACMP output.
- Adjusting RV1 will change the amount of darkness necessary to toggle the ACMP.

Lab 5 – Analog Comparator

- Use a business card to block light to RZ1 (photo sensor). Application code will measure number of seconds between toggles. The display will show:

```
Press SW1 or SW2 on DEMO9S08QG8 to exit
Time since last ACMP toggle = $0001 seconds
Press SW1 or SW2 on DEMO9S08QG8 to exit
Time since last ACMP toggle = $0001 seconds
Press SW1 or SW2 on DEMO9S08QG8 to exit
```

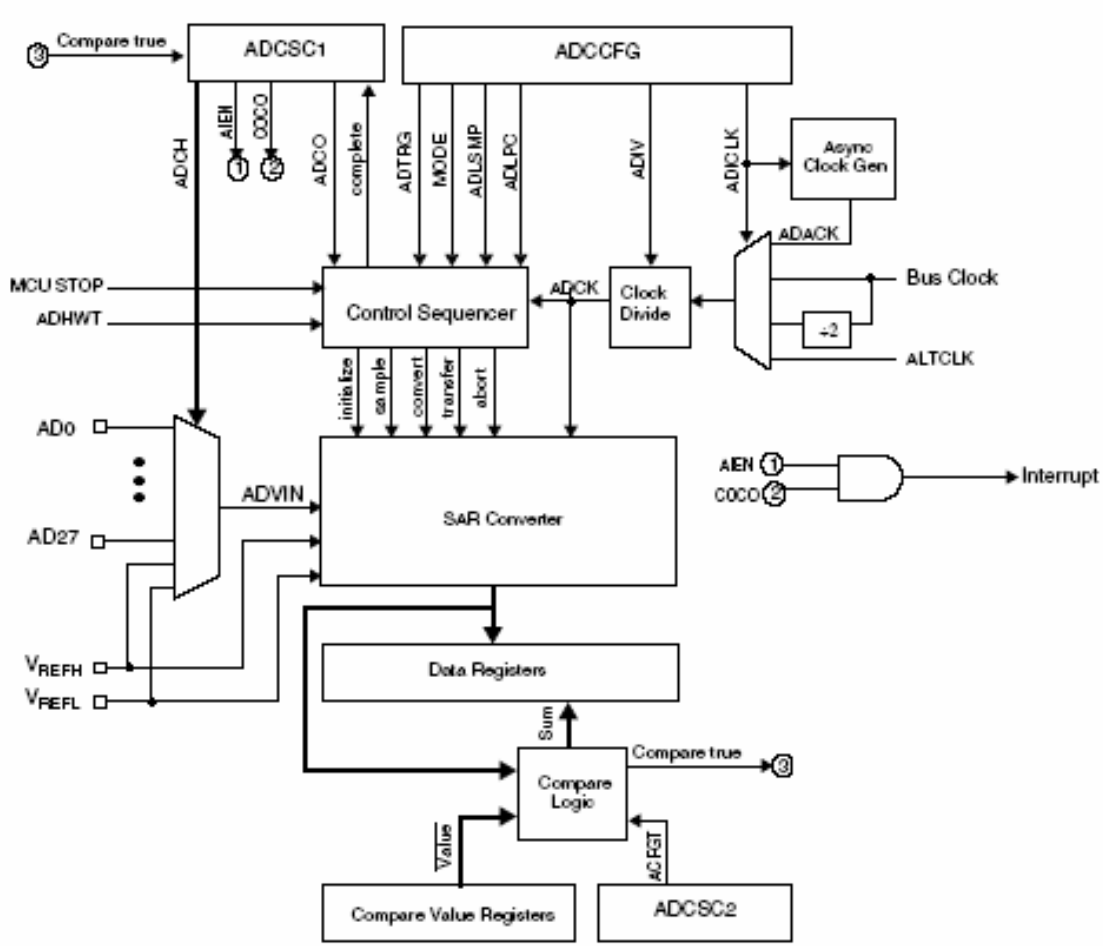


Figure 9-2. ADC Block Diagram

- ADC Features
  - Eight input channels for the QG8
  - Two options for resolution: 8-bit or 10-bit, configurable by software
  - Conversion type adaptable to each application: allows single or continuous conversion
  - Includes a conversion complete flag and a conversion complete interrupt, thus allowing the user to choose either polling or an interrupt-based approach
  - Selectable ADC clock frequency: includes a bus clock prescaler and asynchronous clock source (ADACK).

	7	6	5	4	3	2	1	0
R W	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
Reset:	0	0	0	0	0	0	0	0

Figure 9-11. Pin Control 1 Register (APCTL1)

	7	6	5	4	3	2	1	0
R W	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
Reset:	0	0	0	0	0	0	0	0

Figure 9-13. Pin Control 3 Register (APCTL3)

	7	6	5	4	3	2	1	0
R W	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
Reset:	0	0	0	0	0	0	0	0

Figure 9-12. Pin Control 2 Register (APCTL2)

ADC Pin Control Registers

They are used to disable the I/O port control of MCU pins used as analog inputs. APCTL1 is used to control the pins associated with channels 0–7 of the ADC module. APCTL2 is used to control channels 8–15 of the ADC module. APCTL3 is used to control channels 16–23 of the ADC module.

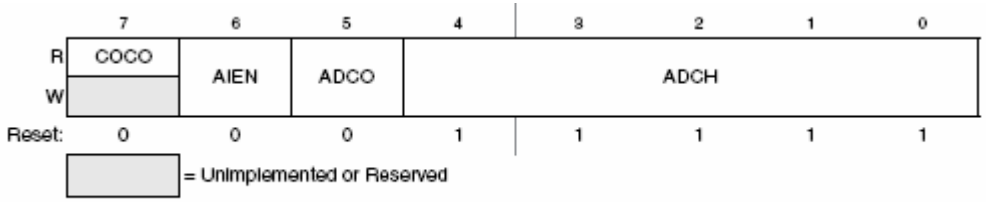
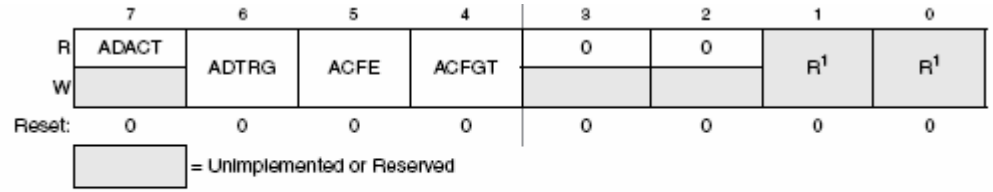


Figure 9-3. Status and Control Register (ADCSC1)



1 Bits 1 and 0 are reserved bits that must always be written to 0.

Figure 9-5. Status and Control Register 2 (ADCSC2)

ADC Status and Control Registers

Writing ADCSC1 aborts the current conversion and initiates a new conversion (if the ADCH bits are equal to a value other than all 1s). The ADCSC2 register is used to control the compare function, conversion trigger and conversion active of the ADC module.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	ADR9	ADR8
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-6. Data Result High Register (ADCRH)

	7	6	5	4	3	2	1	0
R	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-7. Data Result Low Register (ADCRL)

ADC Data Result Registers

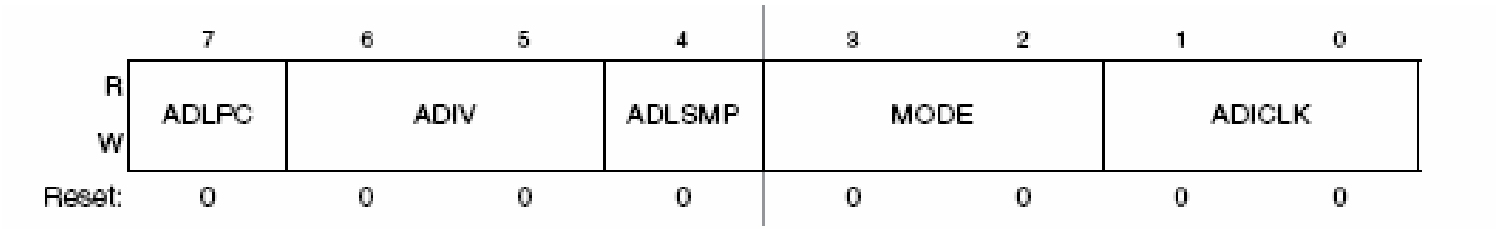
ADCRH contains the upper two bits of the result of a 10-bit conversion. When configured for 8-bit conversions both ADR8 and ADR9 are equal to zero. ADCRL contains the lower eight bits of the result of a 10-bit conversion, and all eight bits of an 8-bit conversion.

Figure 9-8. Compare Value High Register (ADCCVH)

Figure 9-9. Compare Value Low Register(ADCCVL)

ADC Compare Value Register

ADCCVH holds the upper two bits of the 10-bit compare value. These bits are compared to the upper two bits of the result following a conversion in 10-bit mode when the compare function is enabled. In 8-bit operation, ADCCVH is not used during compare. ADCCVL holds the lower 8 bits of the 10-bit compare value, or all 8 bits of the 8-bit compare value.



	7	6	5	4	3	2	1	0
R W	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
Reset:	0	0	0	0	0	0	0	0

Figure 9-10. Configuration Register (ADCCFG)

ADC Configuration Register

ADCCFG is used to select the mode of operation, clock source, clock divide, and configure for low power or long sample time.

Select Option 7: ADC test

Main Test Menu:

1: ICS setup
2: TPM PWM setup
3: MTIM setup
4: RTI setup
5: STOP test
6: ACMP test
7: ADC test

Select number to execute: 7

Select Option 1: Meas RV1 & RZ1

ADC Menu:

1: Meas RV1 & RZ1

2: Meas Internal Bandgap

e: Exit ADC menu

Select test number to execute: 1

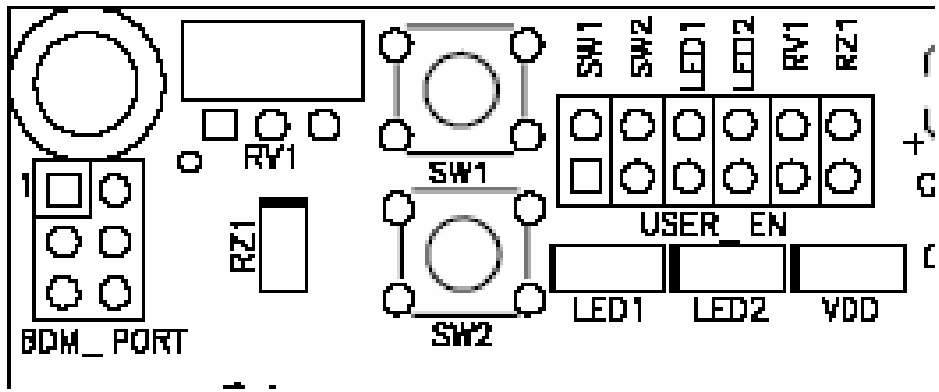
Value of RV1 (potentiometer) = 03FF

Value of RZ1 (photo sensor) = 0104

We will add code to read on-chip temperature sensor

ADC Measurements

Adjust RV1 (potentiometer) and cover RZ1 with paper (photo sensor). Note output value change. Current flow within the phototransistor is inversely proportional to light intensity incident on the surface of the device.



Select Option 2: Meas Internal Bandgap

ADC Menu:
1: Meas RV1 & RZ1
2: Meas Internal Bandgap
e: Exit ADC menu

Select test number to execute: 2
Internal bandgap = 016D (1.76 VDC)

The reading of the bandgap voltage will not change (except for very small changes due to board noise) unless the VDD of the MCU is changed.

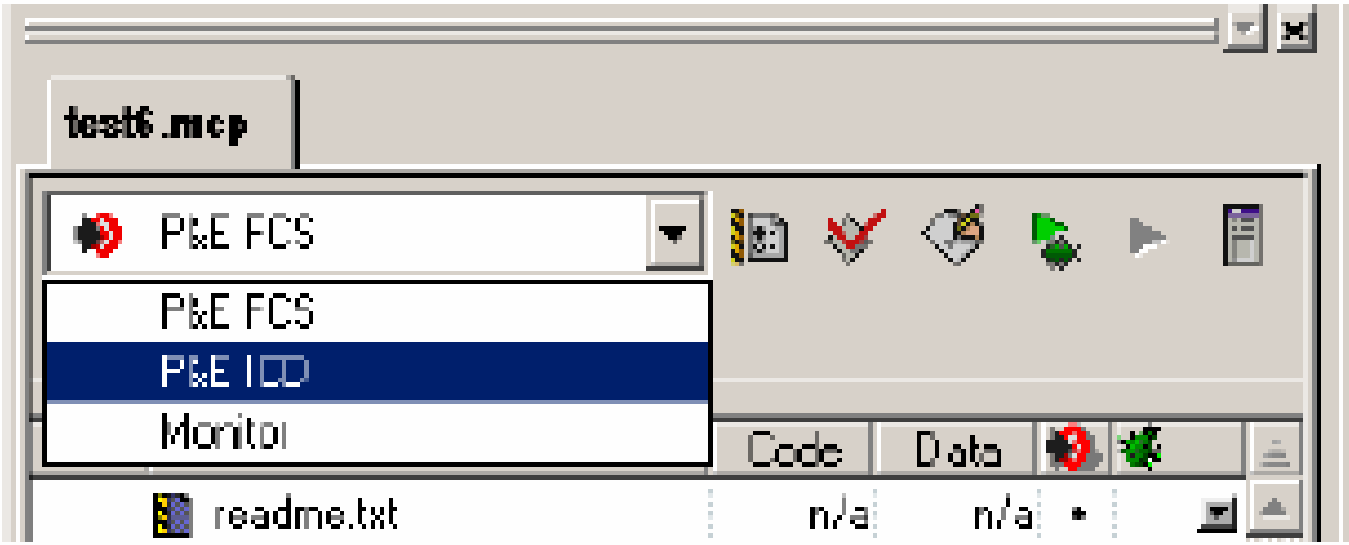
Lab 6 – ADC Temperature Sensor

Edit Application Code to include Temperature Sensor measurement code. Internal temperature sensor is multiplexed on ADC channel 26 (0x1A). Append code to adc_meas(void) function as follows:

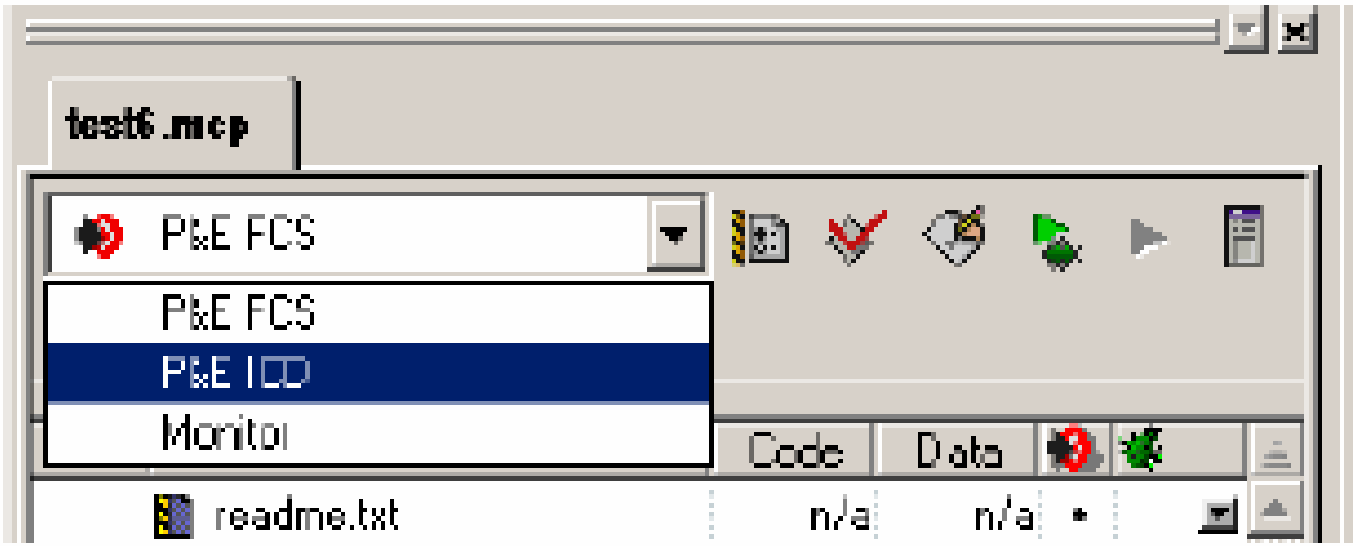
```
ADC_Go(1);
SendMsg("\r\nValue of RZ1 (photo sensor) = ");
SendMsg(num2asc(ADC_val_H));
SendMsg(num2asc(ADC_val_L));
ADC_Go(0x1A);
SendMsg("\r\nValue of Temperature Sensor = ");
SendMsg(num2asc(ADC_val_H));
SendMsg(num2asc(ADC_val_L));
SendMsg("\r\n");
```

Lab 6 – ADC Temperature Sensor

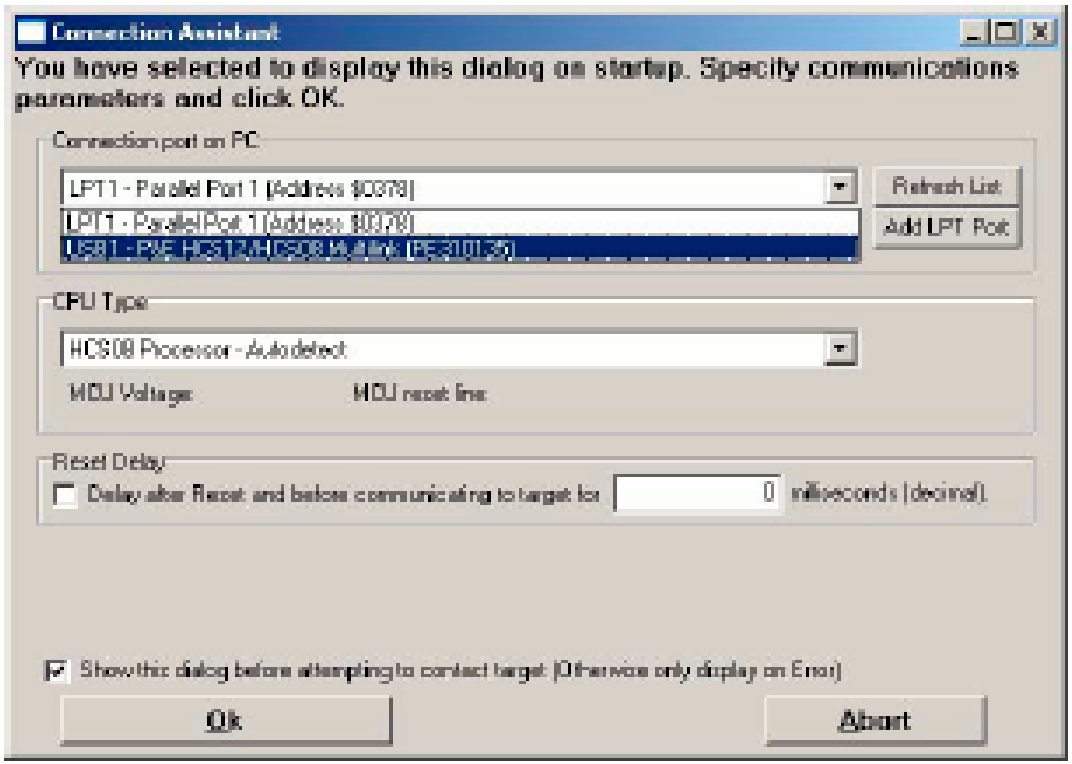
1. From the target connection pull-down, ensure that **P&E ICD** is selected:



2. Select Debug (**GREEN ARROW**). This compiles and links the code, and invokes the True-Time simulator and Real-Time debugger (the HCS08 debugger). As this is the first time the code has been run, the debugger does not know which BDM device is to be used to connect to the target, so it pops up a dialog box for configuration:

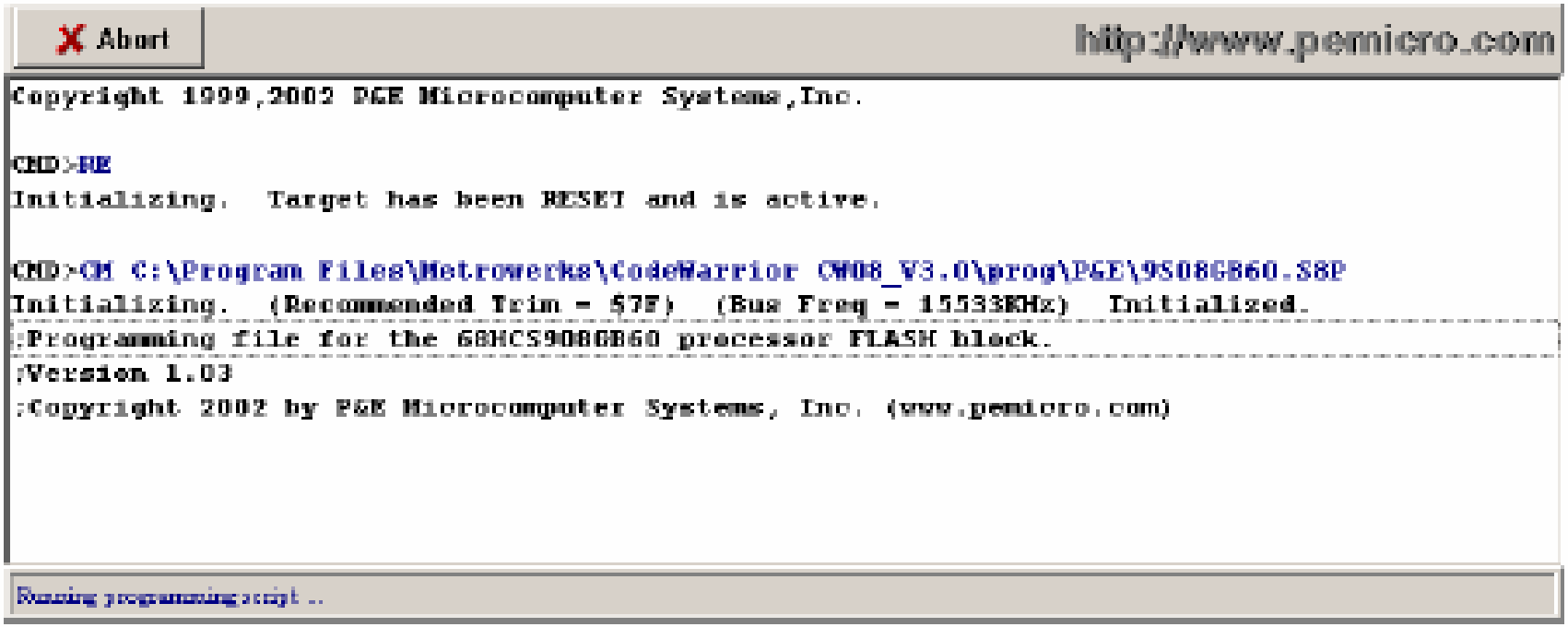


2. Ensure **USB HCS08/HCS12 Multilink - USB Port** is being used:



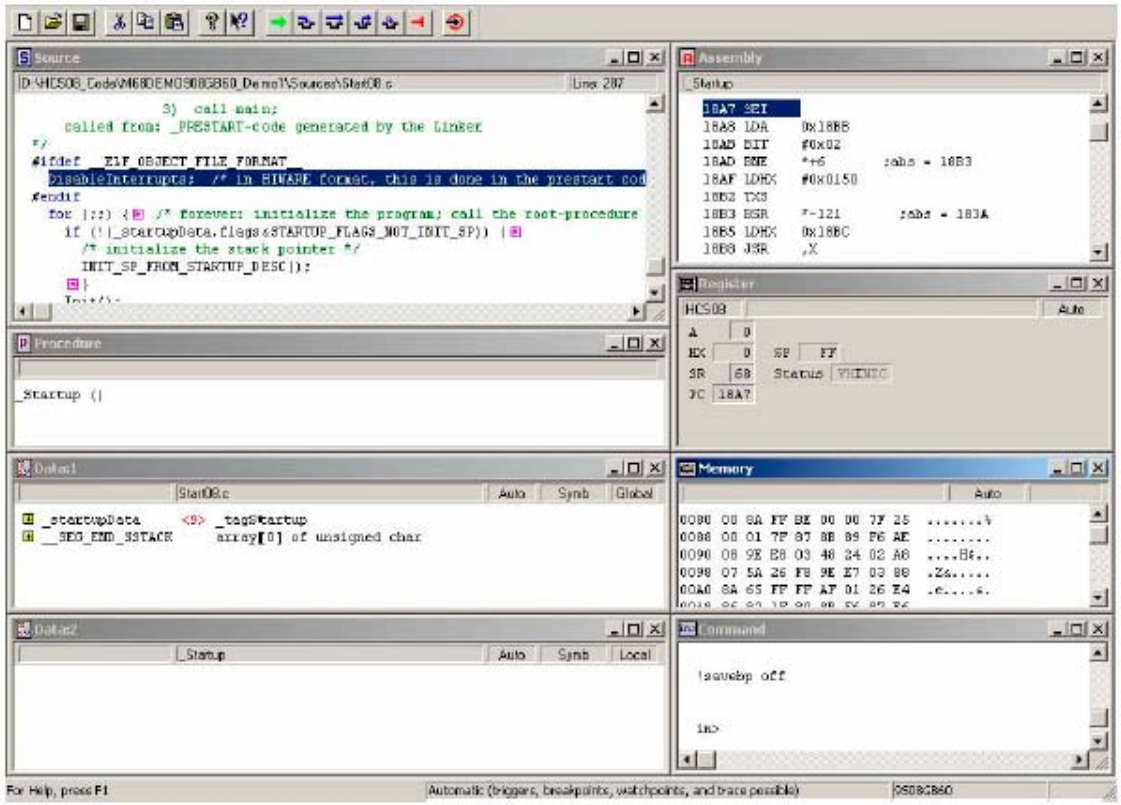
3. Next, a dialog pops up asking to erase and program FLASH. Press **Yes**:

4. Select to proceed. Another window pops up showing the erasure and programming of the FLASH:



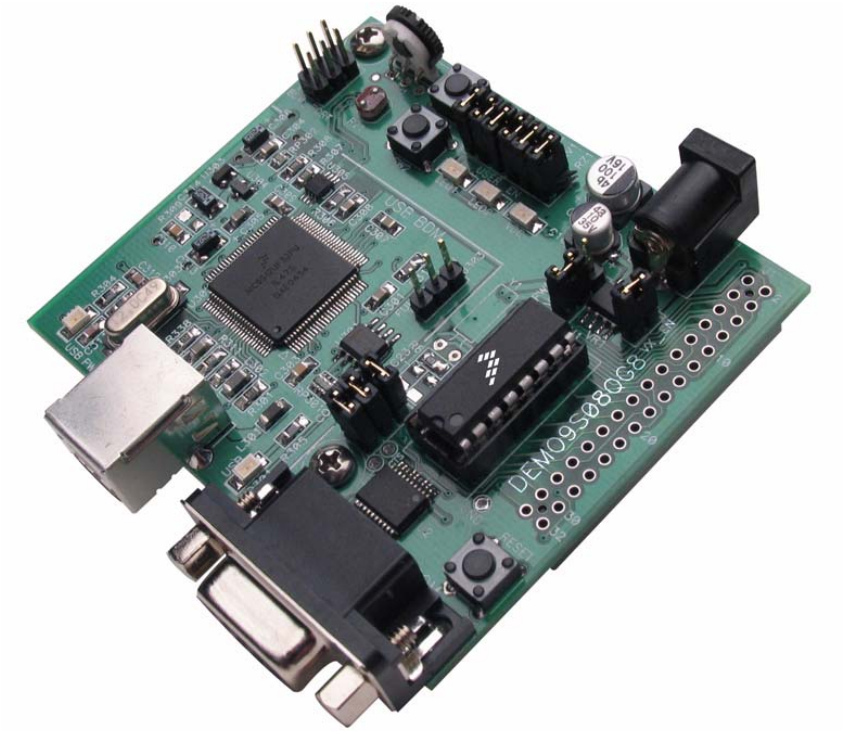
Lab 6 – ADC Temperature Sensor

5. Once communication (BDM or Monitor) has been established with the device, the debugger will show its debug window:



Lab 6 – ADC Temperature Sensor

6. Press **RUN** button and code should execute. LED1 should be lit.



Lab 6 – ADC Temperature Sensor

Select Option 7: ADC test

Main Test Menu:
1: ICS setup
2: TPM PWM setup
3: MTIM setup
4: RTI setup
5: STOP test
6: ACMP test
7: ADC test

Select number to execute: 7

ADC Options

ADC Menu:
1: Meas RV1 & RZ1
2: Meas Internal Bandgap
e: Exit ADC menu

Select test number to execute: 1
Value of RV1 (potentiometer) = 03FF
Value of RZ1 (photo sensor) = 0104
Value of Temperature Sensor = 00D3

Temperature Conversion

Temp = 25 - ((VTEMP -VTEMP25) ÷ m) where:

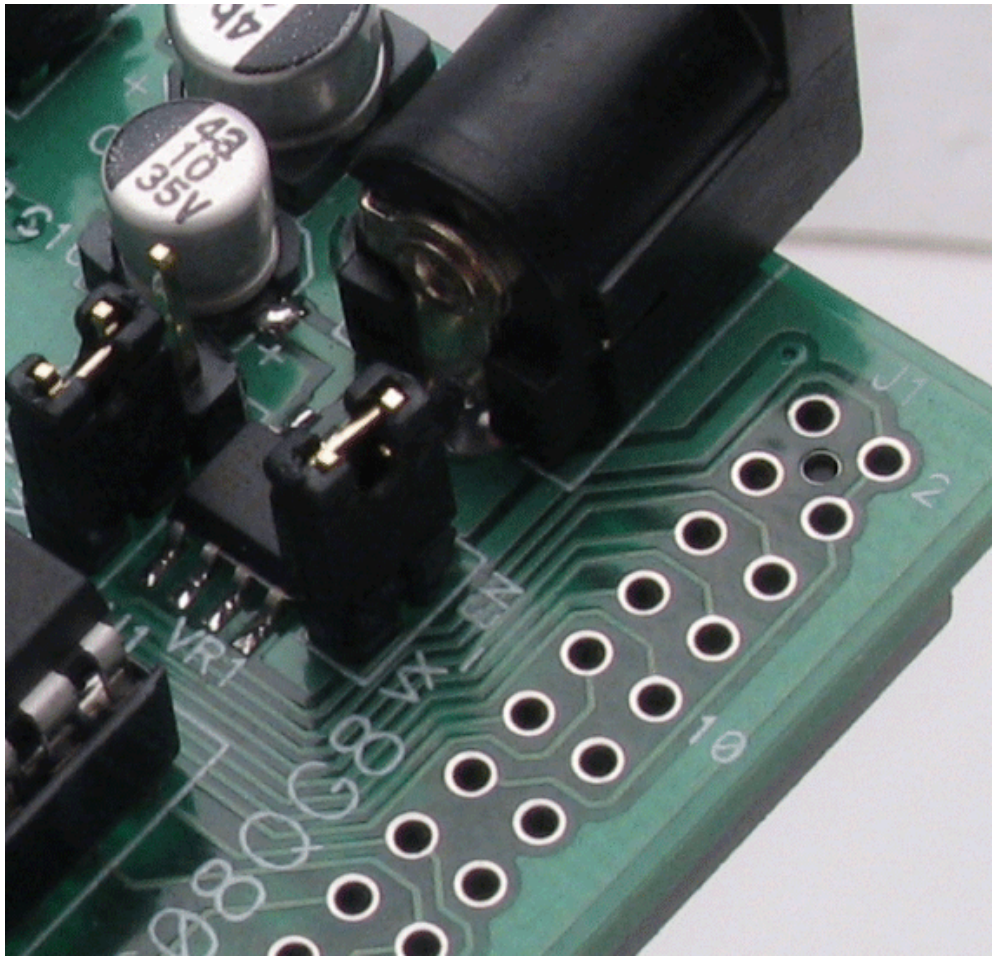
— VTEMP is the voltage of the temperature sensor channel at the ambient temperature.

— VTEMP25 is the voltage of the temperature sensor channel at 25°C.

— m is the hot or cold voltage versus temperature slope in V/°C. Specified at 1.646 from -40 to 25C, 1.769 from 25 to 85C.

Temperature sensor accuracy is being characterized at the factory. Application Note AN3031 will be published.

Questions



Questions, Summary and Wrap-Up

Thank You!!!