DEMO9S08QD4

Demonstration Board for Freescale MC9S08QD4

Axiom Manufacturing • 2813 Industrial Lane • Garland, TX 75041
Email: Sales@axman.com Web: http://www.axman.com

CONTENTS

FIGURES

TABLES

REVISION

Date	Rev	Comments
August 18, 2006	A	Initial Release.

CAUTIONARY NOTES

Electrostatic Discharge (ESD) prevention measures should be used when handling this product. ESD damage is not a warranty repair item.

Axiom Manufacturing does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under patent rights or the rights of others.

EMC Information on the DEMO9S08QD4 board:

This product, as shipped from the factory with associated power supplies and cables, has been verified to meet with **FCC** requirements as a **CLASS A** product.

This product is designed and intended for use as a development platform for hardware or software in an educational or professional laboratory.

In a domestic environment, this product may cause radio interference in which case the user may be required to take adequate prevention measures.

Attaching additional wiring to this product or modifying the product operation from the factory configuration may affect its performance and cause interference with other apparatus in the immediate vicinity. If such interference is detected, suitable mitigating measures should be taken.

TERMINOLOGY

This development board uses option selection jumpers. A jumper is a plastic shunt that connects 2 terminals electrically. Terminology for application of the option jumpers is as follows:

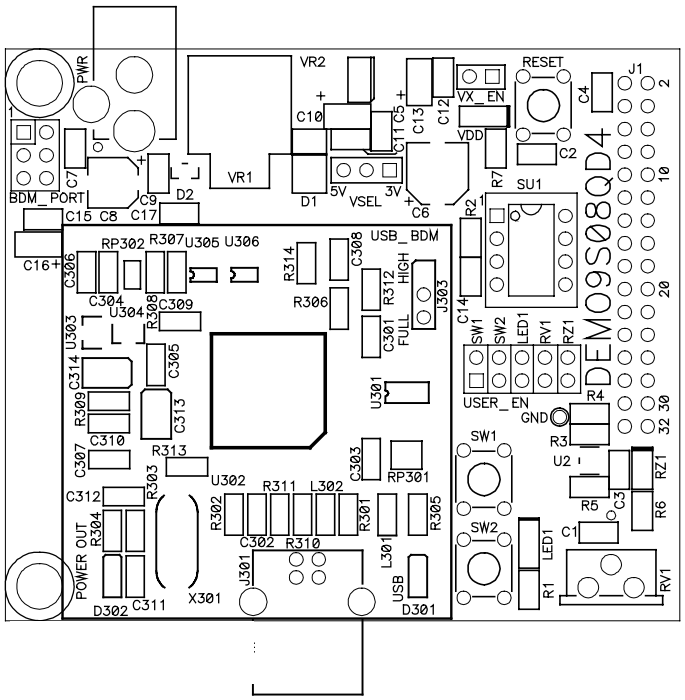
Jumper on, in, or installed - jumper is installed such that 2 pins are connected together.

Jumper off, out, or idle - jumper is installed on 1 pin only. It is recommended that jumpers be idled by installing on 1 pin so they will not be lost.

FEATURES

The DEMO9S08QD4 is an evaluation or demonstration board for the MC9S08QD4 MCU. Application development is quick and easy with the integrated USB-BDM, sample software tools, and examples. A standard 6-pin, BDM port is also provided to support use of a HCS08/HCS12 BDM cable; however, this connector is not installed in default configurations. A 32-pin connector allows connecting the DEMO9S08QD4 board to an expanded evaluation environment.

Features:



- MC68HC9S08QD4 MCU, 8 PDIP, Socketed
  - 4K Bytes Flash
  - 256 Bytes RAM
  - 4 KBI Inputs
  - 4 GPIO, 1 input only, 1 output only
  - 2ch, TIM1, 16-bit Timer/PWM
  - 1ch, TIM2, 16-bit Timer/PWM
  - 4ch, 10-bit ADC
  - 32 kHz, Internal Clock Source
  - Low-Voltage Detect w/ reset
- Integrated USB-BDM
- BDM_PORT header for BMD cable support (not installed)
- Internal Oscillator Trimmable to 0.2%
- 3ch, 16-bit, Timer / Pulse Width Modulator
- 4ch, 10-bit Analog to Digital Converter
- +5V or +3.3V Operation
- Power Input Selection Jumpers
  - Select operating voltage
  - Power input from USB-BDM
  - Power input from on-board regulator
  - Power input from Connector J1
  - Optional Power output through Connector J1
- User Components Provided
  - 3 Push Switches; 2 User, 1 Reset
  - 4 LED Indicators; 1 User, +5V, USB, POWER_OUT
  - 5K ohm POT
  - Light Sensor
- Jumpers
  - Power Select
  - VSEL
  - VX_EN
  - USER_EN
  - J303 (not installed)
- Connectors
  - 32-pin MCU I/O Connector
  - 2.0mm Barrel Connector
  - BDM_PORT
- Supplied with USB Cable, Documentation (CD), and User Guide

Specifications:
Board Size 2.71” x 2.5”
Power Input: 9VDC typical, +6VDC to +20VDC

REFERENCES

Reference documents are provided on the support CD in Acrobat Reader format.

DEMO9S08QD4_UG.pdf	DEMO9S08QD4 User Guide (this document)
DEMO9S08QD4_SCH_C.pdf	DEMO9S08QD4 Board Schematic, Rev. C
DEMO9S08QD4QSG.pdf	DEMO9S08QD4 Quick Start Guide
AN2627.pdf	Cycle-by-Cycle Instruction Details for HC(S)08 MCU's
DEMO_9S08QD4_Test.zip	CodeWarrior demonstration program code
DEMO_9S08QD4_App.zip	CodeWarrior application code demo

GETTING STARTED

To get started quickly, please refer to the DEMO9S08QD4 Quick Start Guide included with the development kit. This quick start shows how to connect the board to the PC, run an LED test program, install the correct version of CodeWarrior Development Studio, and load a simple CodeWarrior analog-to-digital (ATD) program. Refer to Appendix A for details on using the ATD application.

OPERATING MODES

The DEMO9S08QD4 board operates in two basic modes Run Mode, or Debug Mode. Run Mode supports user application operation from Power-On or Reset. Debug Mode supports the development and debug of applications via the BDM_PORT. See the related sections below for quickly starting the board in the desired mode of operation.

The board has been preloaded with a demonstration program that operates in the Run Mode. The VDD LED is lit when power is applied to the board and the PWR_SEL option header is set correctly.

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 DEMO908QD4 board for RUN Mode using the USB bus to power the board. The necessary P&E drivers must be installed on the test PC first. See the POWER section below for details on configuring the board for alternate power input.

1. Connect auxiliary equipment to board if needed.

2. Configure the board option jumpers as shown.

Table 1: Run Mode Setup

VSEL	5V or 3V as needed
PWR_SEL	Set to VB
VX_EN	ON if required
USER_EN	All ON

3. Connect the USB cable to an open USB port on the host PC and attach to the USB port on the target board. The USB, USB_PWR, and VDD LEDs will light and the loaded application will begin to execute.

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 USB-BDM or an external HCS08/HC(S)12 BDM cable. Use of the integrated BDM requires a host PC with an available USB port, an A/B USB cable, the necessary P&E drivers. The USB cable used must be USB 2.0 compliant. A 6-pin BDM_PORT header supports the use of an external BDM cable. This header is not installed in default configurations. The steps below describe using the integrated USB-BDM. See the POWER section below for details on configuring the board for alternate power input.

1. Connect auxiliary equipment to board if needed.

2. Install and launch CodeWarrior Development Studio for HC(S)08, P&E PKGHCS08, or other software capable of communicating with the HCS08 MCU.

3. Configure the board option jumpers as shown.

Table 2: Debug Mode Setup

VSEL	5V or 3V as needed
PWR_SEL	Set to VB
VX_EN	ON if required
USER_EN	All ON

4. Connect the supplied USB cable between an available USB port on the host PC and the USB connector on the board.

Hosting development software will establish background communication.

SOFTWARE DEVELOPMENT

Software development requires the use of an HCS08 assembler or compiler and a host PC running a BDM interface. CodeWarrior Development Studio for HC(S)08 is supplied with this board for debugging and flash programming.

MEMORY MAP

The table below shows the MC9S08QD4 memory map. Attempting to access unimplemented memory locations will cause an illegal-address reset. The memory map is grouped into 3 broad categories; Registers, RAM, and Flash. In the memory map below, the non-volatile registers and vector tables are located at the top of the Flash block.

Table 3: Memory Map

0x0000 – 0x005F	Direct Page Registers	96 bytes
0x0060 – 0x015F	RAM	256 bytes
0x0160 – 0x17FF	Unimplemented	5,792 bytes
0x 1800 – 0x184F	High Page Registers	80 bytes
0x1850 – 0xDFFF	Unimplemented	55,216 bytes
0xF000 – 0xFFAF	FLASH	4,096 bytes
0xFFB0 – 0xFBFF	Non-Volatile Registers	
0xFC00 – 0xFFFF	Vectors	

NOTE: Accessing unimplemented memory locations causes an illegal-address reset.

DEVELOPMENT SUPPORT

Application development and debug for the target MC9S08QD4 is supported through the BDM interface. The debug interface consists of an integrated USB-BDM debugger and an optional 6-pin header (BDM_PORT). The BDM_PORT header is not installed in default configuration and may be installed by the user if needed.

Integrated BDM_PORT

The DEMO9S08QD4 board features an integrated USB-BDM debugger from P&E Microcomputer Systems. The integrated debugger supports application development and debugging via the background debug mode. A USB, type B, connector provides connectivity between the target board to the host PC. The necessary software drivers are available in CodeWarrior or from P&E Microcomputers directly.

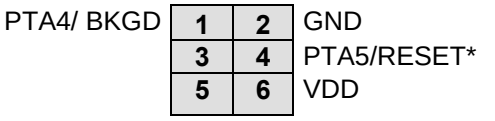
The integrated debugger provides power and ground to the target, thereby eliminating the need to power the board externally. When used, power from the USB-BDM circuit is derived from the USB bus; therefore, total current consumption for the target board, and all connected circuitry, must not exceed **500mA**. Excessive current drain will violate the USB specification causing the USB bus to disconnect; power is removed from the target forcing a POR.

CAUTION: Violating the USB specification causes the USB bus to disconnect forcing the target to reset.

BDM_PORT Header

An HCS08/HC(S)12 BDM cable may be attached to a 6-pin BDM_PORT port header. This header is not installed in default configuration. Use of this port requires the user to install a 2x3, 0.1" center, pin header. Refer to the BDM cable documentation for details on the use of the BDM cable. The BDM_PORT header pin-out is shown below.

Figure 1: BDM Port



See the HCS08 Device User Guide for complete BDM_PORT documentation

NOTE: This header is not installed in default configuration.

POWER

The DEMO9S08QD4 is designed to be powered from the integrated USB-BDM during application development. However, a 2.0mm barrel connector has been applied to support stand-alone operation. Additionally, the board may be powered through connector J1. The board may also be configured to supply power through connector J1 to external circuitry.

The USB-BDM uses the USB bus to provide the target board with +5V for the upper rail. Total current consumption of the board and connected circuitry must be limited to less than **500mA**. Excessive current drain will violate the USB specification causing the USB bus to disconnect. This will force a power-on-reset (POR) at the target MCU.

CAUTION: Violating the USB specification causes the USB bus to disconnect forcing the target to reset.

A 2.0mm barrel connector input has been provided to allow stand-alone operation. Voltage input at this connector should be limited to between +6V and +20V. Two LDO regulators provide both +5V and +3.3V to the target MCU. Either voltage is selectable using the VSEL option header. Both regulators provide over-current and over-temperature protection. Stand-alone operation is also supported through connector J1.

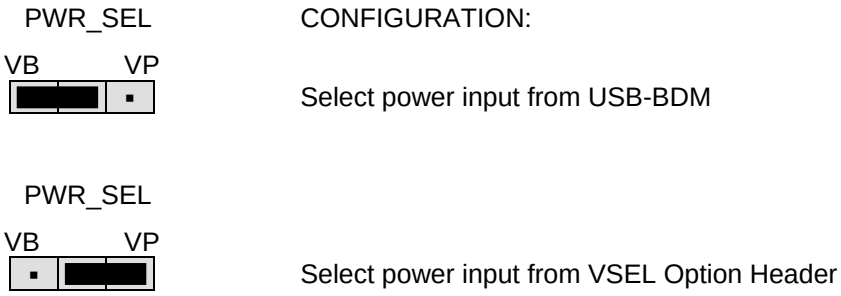
POWER SELECT

Power may be applied to the board through the integrated USB-BDM circuitry, a 2.0mm barrel connector, or through connector J1. Power selection is achieved by using 2 option headers: PWR_SEL option header and the VX_EN option header.

The PWR_SEL option header selects power input either from the integrated USB-BDM circuitry or from the on-board voltage regulator. Power input selection, from the USB-BDM or the on-board power supply, is mutually exclusive. This prevents power-input contention from damaging the board. The figure below details the PWR_SEL header connections.

PWR_SEL

Figure 2: PWR_SEL Option Header



Power from the integrated BDM is drawn from the USB bus and is limited to **500mA**. Excessive current drain will violate the USB specification will cause the USB bus to disconnect.

CAUTION: Violating the USB specification causes the USB bus to disconnect forcing the target to reset.

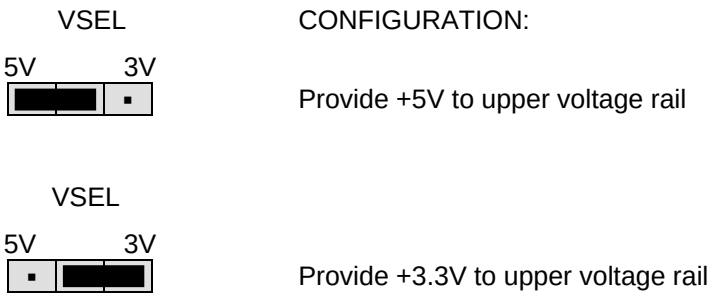
The on-board voltage regulator (VR1) accepts power input through a 2.0mm barrel connector (PWR). Input voltage may range from +5V to +18V. The voltage regulator (VR1) provides a +3.3V fixed output limited to 250mA. Over-temperature and over-current limits built into the voltage regulator protects the device from excessive stresses.

The user should consider the maximum output current limit of the selected power source when attempting to power off-board circuitry through connector J1.

VOLTAGE SELECT

The DEMO9S08QD4 is designed to allow the user to drive the target MCU at either +5V or +3.3V. However, this option is only available when the board is powered from the external power supply connected to the barrel connector. The VSEL option header selects the voltage applied to the upper voltage rail. An LDO voltage regulator at VR1 creates the +5V voltage rail. An LDO regulator at VR2 creates the +3.3V rail from the +5V output of VR1.

Figure 3: VSEL Option Header

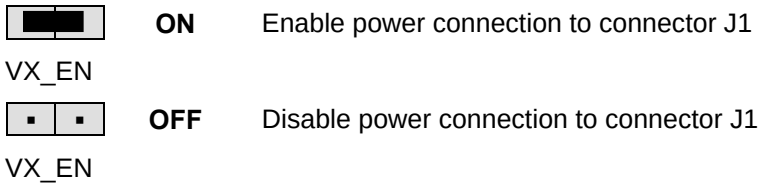


VX_EN

The VX_EN option header is a 2-pin jumper that connects the target board voltage rail to J1-1. J1-3 is connected directly to the ground plane. Use of this feature requires a regulated +3.3V input power source. This power input is decoupled to minimize noise input but is not regulated. Care should be exercised when using this feature; no protection is applied on this input and damage to the target board may result if over-driven. Also, do not attempt to power the target board through this connector while also applying power through the USB-BDM or the PWR connector; damage to the board may result.

Power may be sourced to off-board circuitry through the J1 connector. The current limitation of the USB bus or the on-board regulator must be considered when attempting to source power to external circuitry. Excessive current drain may damage the target board, the host PC USB hub, or the on-board regulator. The figure below details the VX_EN option header connections.

Figure 4. VX_EN Option Header



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 the USB-BDM cable or on-board regulator when sourcing power through connector J1 to external circuitry.

RESET SWITCH

The RESET switch provides a method to apply an asynchronous reset to the MCU and is connected directly to the PTA5/RESET* input on the MCU. Pressing the RESET switch forces the MCU RESET* input low. The MC9S08QD4 MCU applies an internal pull-up on the RESET* line to prevent spurious resets and allow normal operation.

LOW VOLTAGE DETECT

The MC9S08QD4 includes an internal Low Voltage Detect (LVD) to protect against under-voltage conditions. The LVD is enabled out of RESET. Consult the MC9S08QD4 Device User Guide for details on configuring LVD operation.

TIMING

Timing for the DEMO9S08QD4 is provided by an internal 32 kHz frequency reference and an internal frequency-locked loop (FLL). The FLL output is trimmable to ± 0.2% of nominal. Refer to the MC9S08QD4 Device User Guide for further details on clock operation.

No external timing option is available on the DEMO9S08QD4

USER OPTIONS

The DEMO9S08QD4 includes various input and output devices to aid application development. User I/O devices include 2 momentary pushbutton switches, 1 green LED, 1 potentiometer, and 1 phototransistor. Each device may be enabled or disabled individually by the USER_EN option header. Each user enable is clearly marked as to functionality.

Pushbutton Switches

Two push button switches provide momentary, active-low input, for user applications. Pull-ups internal to the MCU must be enabled to provide error free switch operation. Pushbutton switches SW1 and SW2 are enabled to the MCU I/O ports by the USER option bank. SW1 and SW2 connect to input ports PTA2 and PTA3 respectively. Table 4 below details the user jumper settings.

LED Indicators

Indicator LED1 is enabled from the HC(S)08 I/O port by the USER option bank. Each LED is active-low and illuminates when a logic low signal is driven from the respective MCU I/O port. MCU ports PTA4 drives LED1. Table 4 below details the user jumper settings.

Potentiometer

A 5k ohm, thumb-wheel type, potentiometer at RV1 provides variable resistance input for user applications. The output is the result of a voltage divider that changes as the thumb-wheel is turned. 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. Table 4 below details the user jumper settings.

Photocell

A surface-mount phototransistor provides light sensitive, variable input for user applications. 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. Table 4 below details the user jumper settings.

Table 4: User Option Jumper Settings

Jumper	On	Off	MCU PORT	MCU PIN
SW1	Enable SW1	Disable SW1	PTA2	6
SW2	Enable SW2	Disable SW2	PTA3	5
LED1	Enable LED1	Disable LED1	PTA4	2
RV1	Enable RV1	Disable RV1	PTA0	8
RZ1	Enable RZ1	Disable RZ1	PTA1	7

I/O PORT CONNECTOR

This port connector provides access to DEMO9S08QD4 I/O signals. Signal positions not shown are not connected on the board.

Figure 5: MCU I/O Port Connector

VDD	1	2	PTA5/RESET*/IRQ*/TPM2CHO1
VSS	3	4	PTA5/RESET*/IRQ*/TPM2CHO2
	5	6	PTA4/BKGD/MS/TPM2CHO0
	7	8	
PTA2/KBI1P2/ADC1P2/TCLK1	9	10	
PTA3/KBI1P3/ADC1P3/TCLK2	11	12	
PTA5/RESET/ITQ/TPM2CHO1	13	14	
PTA0/KBI1P0/ADC1P0/TPM1CH0	15	16	
	17	18	PTA1/KBI1P1/ADC1P1/TPM1CH1
	19	20	PTA0/KBI1P0/ADC1P0/TPM1CH0
	21	22	
	23	24	
PTA1/KBI1P1/ADC1P1/TPM1CH1	25	26	
	27	28	
	29	30	
PTA4/BKGD/MS/TPM2CHO0	31	32	

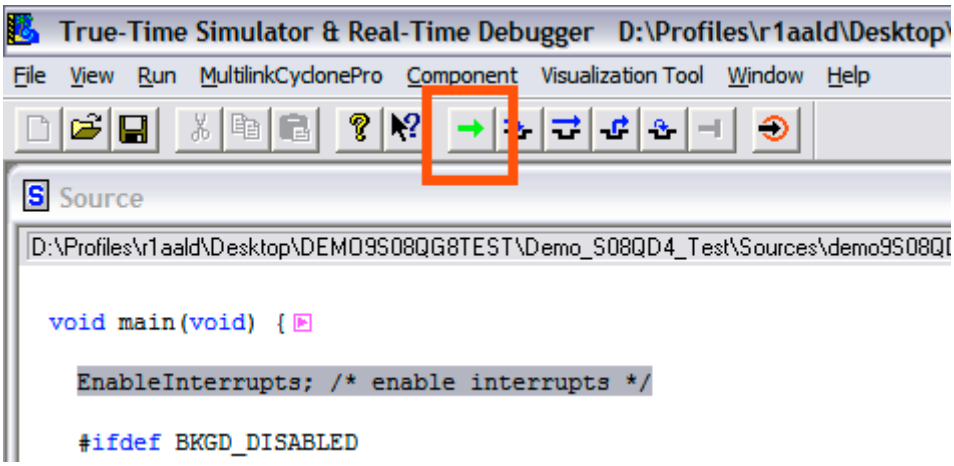
APPENDIX A

DEMO9S08QD4_App Program

Debug Mode

Follow the instructions provided in the Quick Start guide document section “Use the USB BDM to install DEMO9S08QD4_APP program”. To start program execution, press the Start/Continue (F5) button, in the True-Time Simulator, as shown in Figure 6 below. The application code executes.

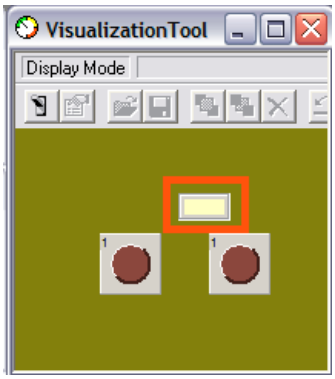
Figure 6: Run ATD Program



The application code configures the ADC to read the potentiometer (RV1) and sensor (RZ1) inputs. The resulting ADC value is then used to turn the Visualization Tool LED on and off. Refer to Figure 7 below. SW1 and SW2 are used to select between ADC inputs. The Visualization Tool LED turns on while the ADC result is greater and 0x1FF. When the ATD result falls below 0x1FF, the LED turns off indicating an alarm state.

Press SW1 on the demo board to configure ADC reading of the potentiometer (RV1). Rotate RV1 back and forth to make the Visualization Tool LED change states. Press SW2 on the demo board to configure ADC readings of the photo-sensor (RZ1). Cover or uncover RZ1 to make the Visualization Tool LED change states.

Figure 7: Visualization Tool



Stand-Alone Mode

On the DEMO9S08QD4, the BKGD pin is shared with the LED1. During programming and debugging, the LED shows BDM activity rather than the expected LED behavior. In the example above visualization tool was used to show the expected LED1 behavior.

For stand-alone operation, the BKGD pin must be disabled. This allows PTA1 to act as a general-purpose input/output pin. PTA1 drives LED1. To disable the BKGD pin, modify the demo9s08qd4_App.c file as shown in Figure 8 and Figure 9 below.

Figure 8: DEMO9S08QD4_APP.C Before Change

```
************************************************************************************


#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */
#include "demo9S08QD4.h" /*include demo board declarations */

#define Enter_Wait asm(wait)
//#define BKGD_DISABLED

static char ADC_val_H;
static char ADC_val_L;
static int  ADC_val = 0;
static char channel = 0;
```

Figure 9: DEMO9S08QD4_APP.C After Change

```
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */
#include "demo9S08QD4.h" /*include demo board declarations */

#define Enter_Wait asm(wait)
#define BKGD_DISABLED

static char ADC_val_H;
static char ADC_val_L;
static int  ADC_val = 0;
static char channel = 0;
```

Note: This modification disables BKGD pin. Application code may not be troubleshot or debugged while the BKGD pin is disabled.

The test code will execute out of POR. The modified code can not be debugged while the BKGD pin is disabled. Return the code to its original form to allow debugging using the True-Time Simulator & Real-Time Debugger.