

---

# DEMO9S08QG8\_APP Example Application User Guide

---

## Introduction

This document describes how to set up and use the DEMO9S08QG8\_APP example application that comes with the DEMO9S08QG8 demo board. This example application allows the user to try out several features on Freescale's MC9S08QG8 microcontroller.

---

## Requirements

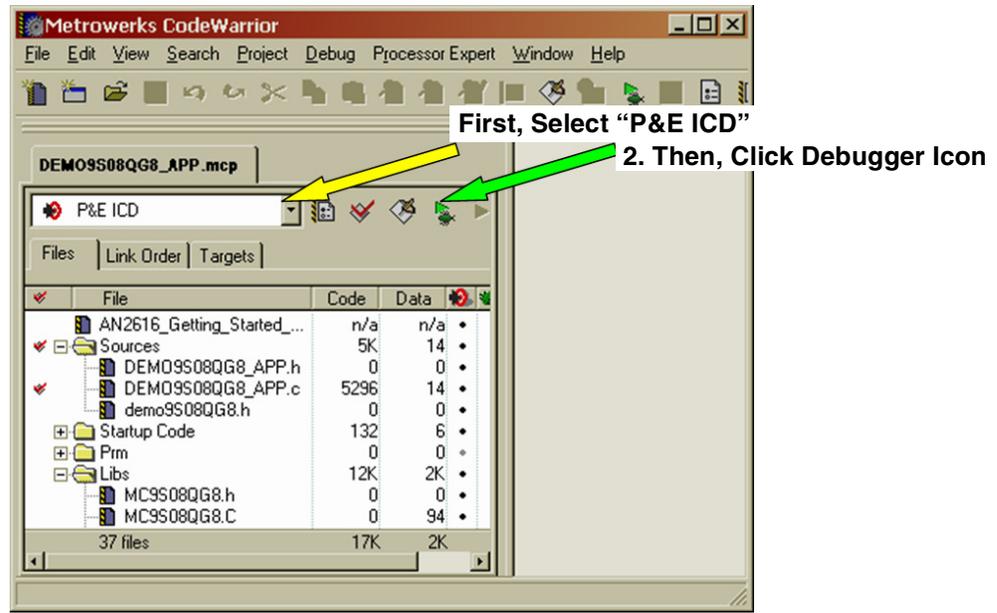
To run this software, you will need the following:

- DEMO9S08QG8 demo board with an MC9S08QG8 in the 16 DIP socket
  - DB9 serial cable
  - PC with:
    - CodeWarrior Development Studio for HC(S)08
    - CodeWarrior Service Pack for MC9S08QG8 (download from [www.metrowerks.com](http://www.metrowerks.com))
    - 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 open the DEMO9S08QG8\_APP project.
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. Then click the debugger icon, as shown in [Figure 1](#)). This will launch the debugger application and ask, "Erase and program FLASH". Click on "Yes". When the programmer pop-up window goes away, the MCU is programmed.

## Running the Demo



After the MCU is programmed, launch the terminal emulator. The emulator must be configured to use the COM port that the DB-9 cable is plugged into, usually COM1. The COM port settings must be:

- 4800 baud
- 8 bits
- No parity
- 1 stop bit

The demo is now ready to run.

---

## Running the Demo

If necessary, open the True-Time Simulator. Select Run -> Start/Continue(F5) to launch the application.

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:
```

This demo relies on the internal clock reference to generate bus frequency and the communications baud rate. Therefore, the MCU's internal clock reference must be trimmed to 31.25 kHz and the trim value stored in

FLASH at location 0xFFAF. CodeWarrior with the MC9S08QG8 Service Pack will trim and save the trim value automatically.

The 31.25 kHz reference will generate an 8.0 MHz bus clock in FEI mode. The demo will light LED1, and the MCU's TPMCH1 pin will be outputting a PWM signal running at 1/10,000 of the bus frequency. If the internal reference is trimmed, this PWM signal should be 800 Hz  $\pm 2\%$ .

### RESET Button and SRS Register

After every chip reset, the display will show the value of the System Reset Status (SRS) register and state the cause or causes of the last reset.

For this demo application, the  $\overline{\text{RESET}}$  pin on the MC9S08QG8 is configured as a general purpose PTA5 input, not as  $\overline{\text{RESET}}$ . However, the program uses PTA5 to determine whether the COP should be serviced. If the RESET button (connected to the PTA5/ $\overline{\text{IRQ}}$ / $\overline{\text{TCLK}}$ / $\overline{\text{RESET}}$  pin) is not pressed, the COP will be serviced and no COP reset will be generated. If the RESET button is pressed, the COP will not be serviced and will therefore timeout, forcing a chip reset.

Because the cause of the reset is actually the COP and not the  $\overline{\text{RESET}}$  pin function, the SRS will indicate a COP reset occurred instead of a pin reset.

---

## Demo Menu Options

The user can choose from eight options to explore a handful of the MC9S08QG8's features. The options allow the user to choose from several ICS setups, change the TPM clock source, reconfigure the MTIM settings, measure the RTI's internal reference, and try out the different stop modes, the ACMP, and the ADC modules.

To make a selection, simply type the number next to the menu item. Some menu choices will require a two-digit value. For these items, type the two-digit number followed by <Enter>. When entering these two-digit values, the delete and backspace keys will not work. To re-type a value, enter a non-hexadecimal value and the menu will ask for a new entry.

### ICS setup

This option allows the user to select from five different settings of the Internal Clock Source (ICS) module. Four of the settings are using FEI mode (FLL-enabled, internal clock reference) with different bus divider values and one option selects FBI mode (FLL-bypassed, internal clock reference).

### TPM PWM setup

This option simply lets the user change the clock source to the TPM (Timer/PWM Module). One of the choices, "TCLK source", requires a clock signal to be input on the PTA5/ $\overline{\text{IRQ}}$ / $\overline{\text{TCLK}}$ / $\overline{\text{RESET}}$  pin. If no clock source is present, the timer will not count and the PWM signal will freeze. The demo will continue to run.

### MTIM setup

This menu option allows the user to do several things with the MTIM (Modulo Timer Module): initiate a software PWM signal running at 1/100 of the bus frequency, choose a different clock source, set the clock source prescaler, set the timer modulo value, start, stop and reset the MTIM.

As with the "TPM PWM setup", if the TCLK is selected as the MTIM clock source, a clock signal must be input on the PTA5/ $\overline{\text{IRQ}}$ / $\overline{\text{TCLK}}$ / $\overline{\text{RESET}}$  pin. If no clock source is present, the MTIM will not count and the software PWM signal will freeze. The demo will continue to run.

### RTI setup

This menu item has several choices for using the RTI (Real-Time Interrupt) feature. All three choices configure the RTI to use its internal 1-kHz clock source. This clock source is independent of the ICS's internal clock reference and cannot be trimmed like the internal clock reference can.

The three choices are: measure the RTI timeout period, start the RTI, and stop the RTI.

Measuring the RTI configures the RTI for a 1 second timeout period and runs the TPM and MTIM while waiting for the RTI flag to timeout. The TPM and MTIM counter values are displayed along with the expected value if the RTI's clock source was exactly 1 kHz. Values less than expected mean the RTI clock is fast; values greater than expected mean the RTI clock is slow.

The "RTI on" option will not appear to do anything on the screen. However, if the RTI is turned on and then the "STOP tests" menu is used to enter stop 2 or stop3, the RTI will automatically wake the MCU out of the stop mode. The "RTI off" can be used if the user wishes to remain in the stop modes.

### STOP test

The "STOP test" options allow the user to configure the MCU for either stop1, stop2, or stop3 mode and then enter stop mode. As mentioned in the "RTI setup", if the RTI is on and the MCU enters stop2 or stop3 (the RTI does not run in stop1 mode), then the RTI will wake the MCU from stop when it times out.

#### NOTE

*After stop1 or stop2 has been selected, the selected stop mode cannot be changed until the MCU has been powered off and back on.*

#### NOTE

*If stop1 or stop2 is selected, the internal pullup on the PTA5/ $\overline{IRQ}$ / $\overline{TCLK}$ / $\overline{RESET}$  pin is disabled. Therefore, an external pullup is required on this pin or the MCU may not stay in stop mode.*

### ACMP test

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.

### ADC test

The "ADC test" has two options: measure the voltages on the photosensor (RZ1) and the potentiometer (RV1) or measure the bandgap voltage. The reading of the bandgap voltage will not change (except for very small changes due to board noise) unless the  $V_{DD}$  of the MCU is changed.