

# Proximity Sensing Software Quick Reference User's Guide

**Devices Supported: HCs08, RS08 and V1 MCU**

## Proximity Sensing Software

### Overview

Any MCU can be enabled for touch sensing capabilities with a properly designed layout and a simple software module. This document explains how to use the Proximity Sensing Software in different MCU's boards. This application can be used in conjunction with the Proximity daughter board PROXIMITYBOARD0 for a complete evaluation set.

### NOTE

The Proximity Software is compatible with all S08, RS08 and V1 microcontrollers but hardware is necessary to test it. This document explains how to setup up the Proximity Software with a DEMO board and the PROXIMITYBOARD0 board, which are included in the KITPROXIMITYEVM. The kit can be purchased at [www.freescale.com/proximity](http://www.freescale.com/proximity).

### System Requirements

The requirements needed for the system are listed below.

#### Proximity Software Requirements:

- Freescale MCU (A demo board such as DEMOQE128 is recommended)
- CodeWarrior 5.1 or later
- Properly designed hardware for electrodes (PROXIMITYBOARD0 recommended)

#### Proximity GUI Requirements:

- Freemaster 1.3 software for BDM connection.
- .Net Framework 2.0
- MCU loaded with the Proximity Software
- BDM connection to MCU (included in most Demo boards, currently only P&E modules are supported)
- Certified USB cable to avoid noise
- 15Mb of hard drive space

# Application Setup

## Overview

This section explains how to program your MCU and how to connect the MCU evaluation board with the PROXIMITYBOARD0 board to test the Software.

### NOTE

This application is configured by default for the DEMOQE128 board. Keep in mind the electrode pins can be configured by the user. The intention of this software is for the users to create their own applications.

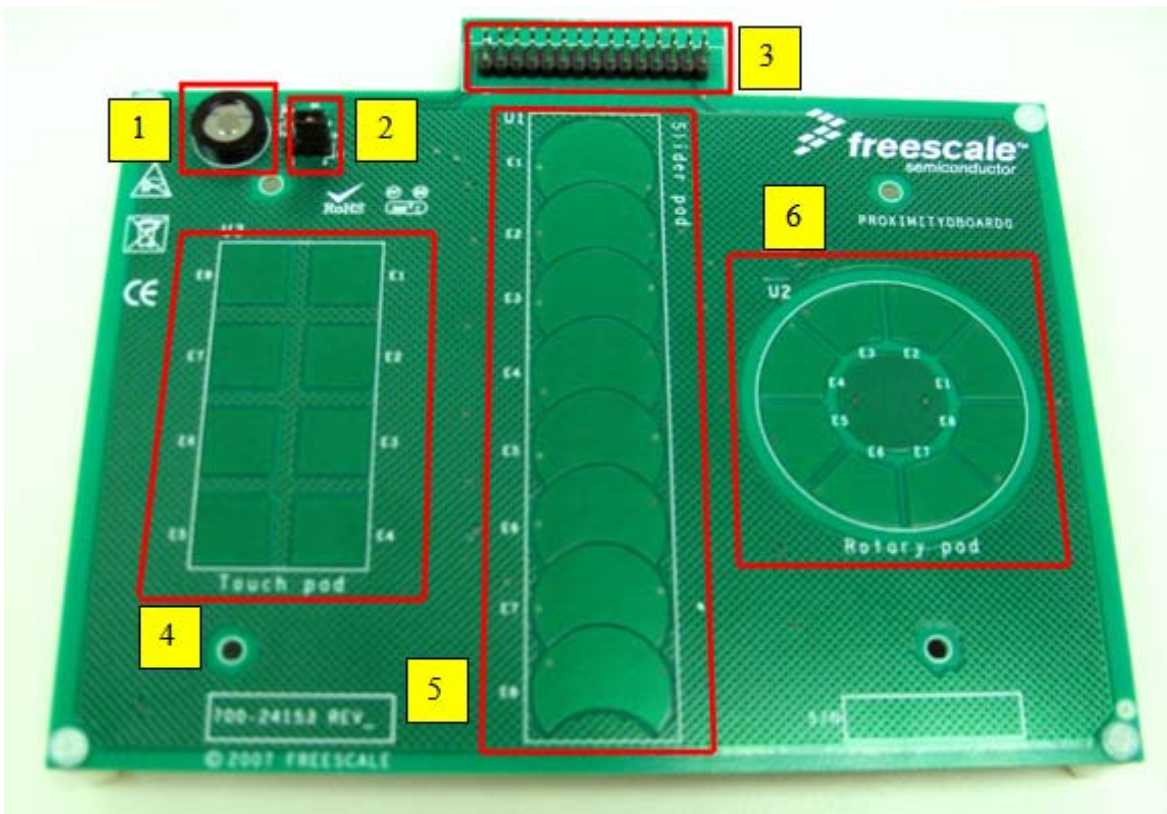
## Hardware Setup

### Board Description

The next figure shows the board PROXIMITYBOARD0. This board has three different electrode arrangements, a jumper, a pin header and a buzzer.

### NOTE

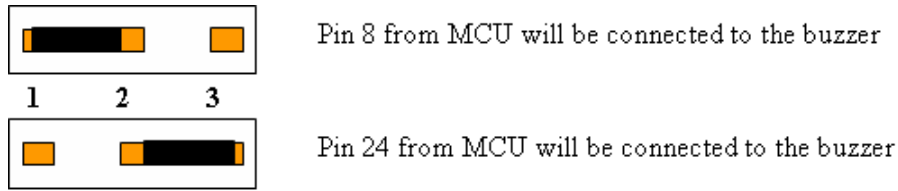
The three different electrode arrangements of the PROXIMITYBOARD0 are connected in parallel. Electrode1 (E1) from the Key pad is connected to E1 from the Slider and E1 from the Rotary.



**Figure 1. PROXIMITYBOARD0 Board**

- |                  |            |
|------------------|------------|
| 1. Buzzer        | 4. Key pad |
| 2. Jumper        | 5. Slider  |
| 3. 30 pin header | 6. Rotary  |

Figure 2 shows the possible connections depending of the jumper position.



**Figure 2. Jumper Connections**

Table 1 shows the header pins that are connected to the electrodes in the PROXIMITYBOARD0 board. For detailed information about connections, please refer to [Appendix A](#) of this document.

**Table 1. PROXIMITYBOARD0 30 Pin Header**

Signal	Pin	Pin	Signal
Vcc	1	2	NC
GND	3	4	NC
NC	5	6	NC
NC	7	8	Buzzer
Electrode 1	9	10	NC
Electrode 2	11	12	NC
Electrode 3	13	14	NC
Electrode 4	15	16	NC
Electrode 5	17	18	NC
Electrode 6	19	20	NC
Electrode 7	21	22	NC
Electrode 8	23	24	Buzzer
NC	25	26	NC
NC	27	28	NC
NC	29	30	NC

## Compatibility

The PROXIMITYBOARD0 board is only compatible with the following MCU boards.

- DEMO9S08QG8
- DEMO9S08QD4
- DEMOQE128
- DEMO9S08AW60
- DEMO9S08DZ60
- DEMO9S08LC60
- DEMOJMSKT
- DEMO9S08RG60

## How to Connect the PROXIMITYBOARD0 with an MCU Demo Board

1. Remove the jumpers that share functions in the Demo board. For example, in DEMOQE128 remove the jumpers associated to PTC0, PTC1 and PTB5 pins. For detailed information regarding jumper removal, see [Appendix B](#).
2. Find pin 1 in the male header of the PROXIMITYBOARD0.
3. Locate the female header slot number 1 in the supported Demo board as shown in [Figure 3](#). The back side of the DEMOQE128 board is shown in [Figure 4](#).



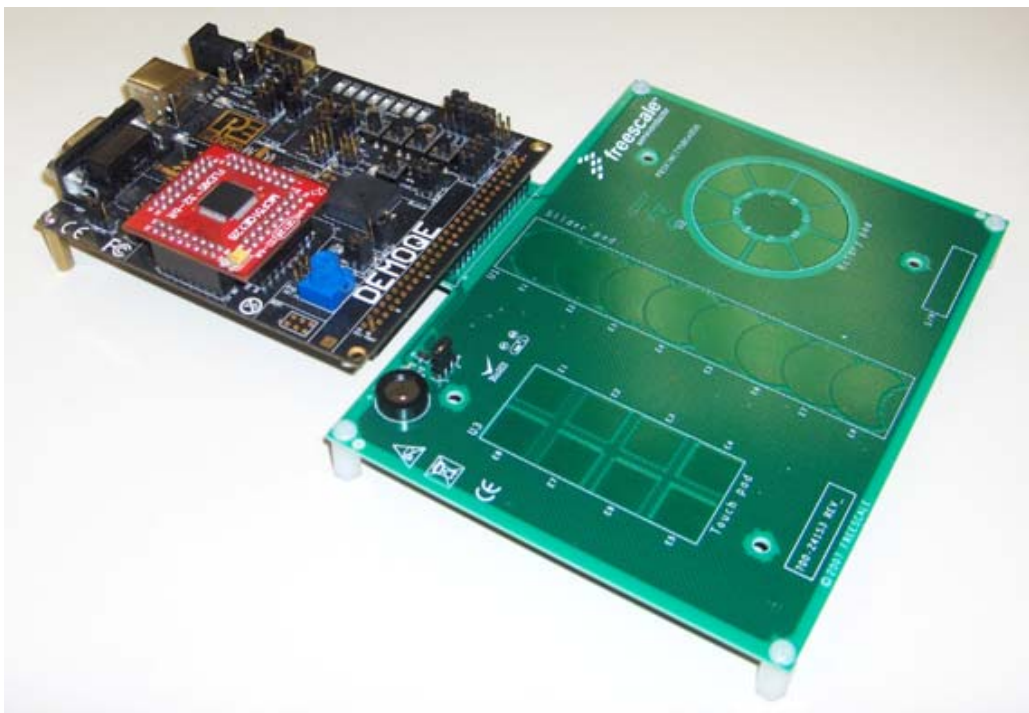
Figure 3. MCU Port Socket Header

4. Plug PROXIMITYBOARD0 socket into the MCU port header. Remember that pins 1 and 3 from both headers must be connected. The two boards must look like the example below.



**Figure 4. The Two Boards Connected**

5. Connect the USB cable to your PC, and the other side of the cable to the DEMO board.
6. The boards are connected as shown in [Figure 5](#).



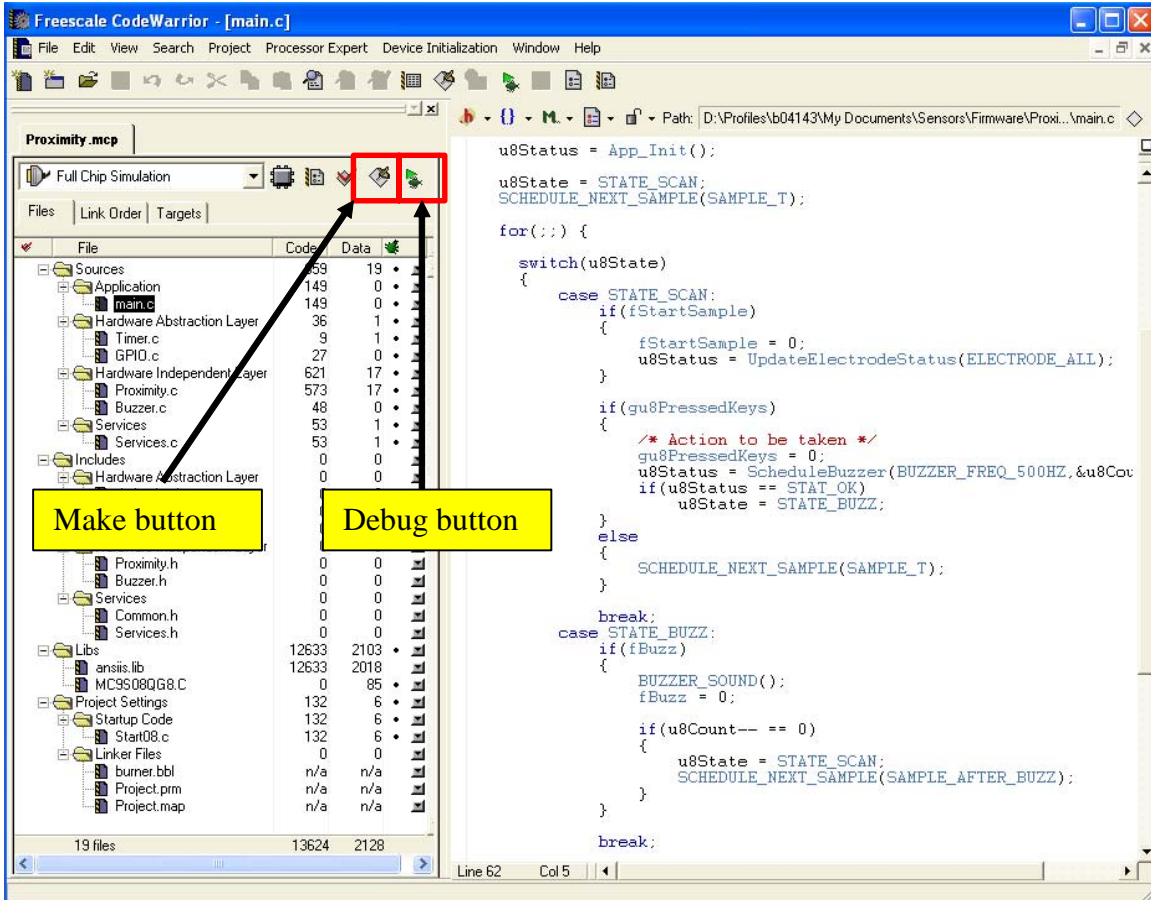
**Figure 5. System Setup**

# Firmware Setup

This section explains how to program any supported Freescale MCU with the Proximity Software. Also described is how to change from one demo board to another one and how to configure the software for a custom application.

## How to Program the MCU

1. It is necessary to have installed CodeWarrior 5.1 or later to program the required MCU. If it is not installed, please install it.
2. You can find the Proximity project in: C:/Program Files/Freescale/Freescale Proximity Sensing Software/proximity Source Code. There are two folders, just select the desired one.
3. Open CodeWarrior and open the Proximity.mcp project file.
4. Click on the Make button as shown in [Figure 6](#)



**Figure 6. CodeWarrior Buttons**

5. Click on the Debug button.
6. A new window appears, then click Connect and the debugger window will pop up. If this action fails, verify the hardware connections.

7. In the debugger window click on the Run button

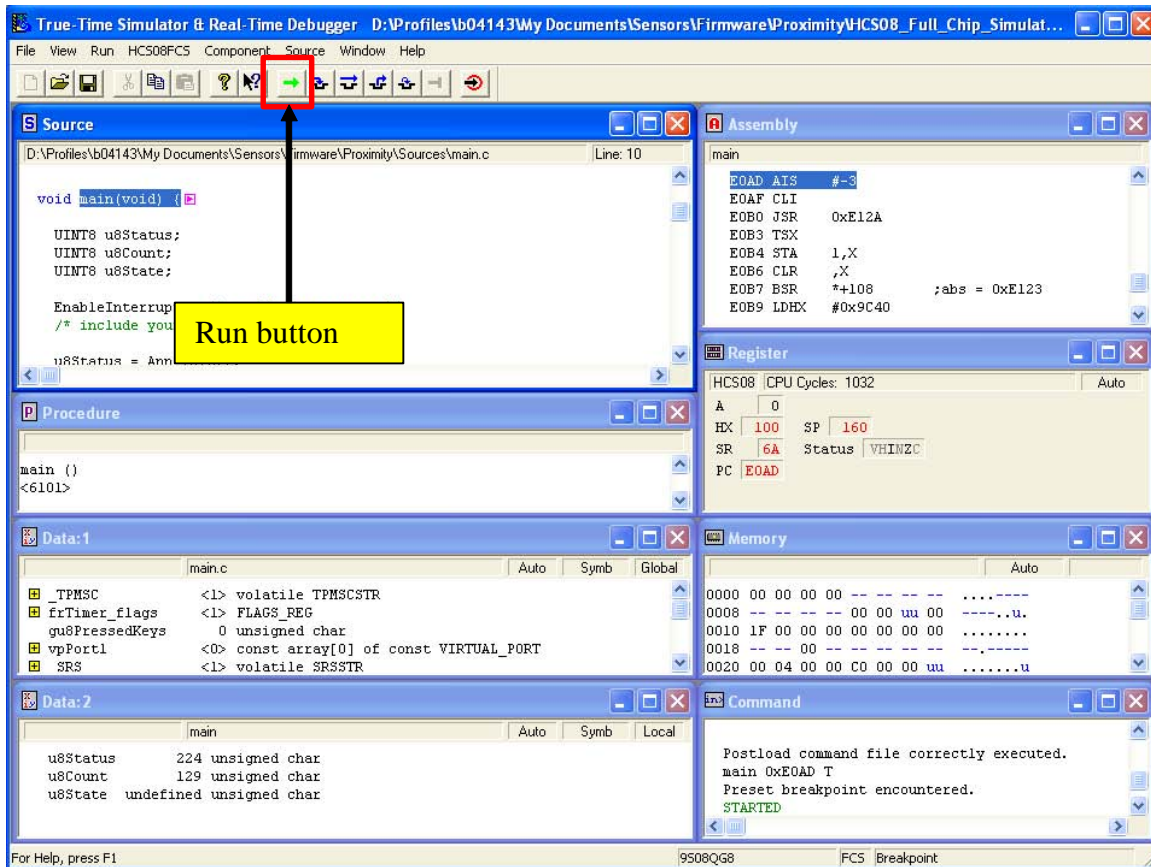


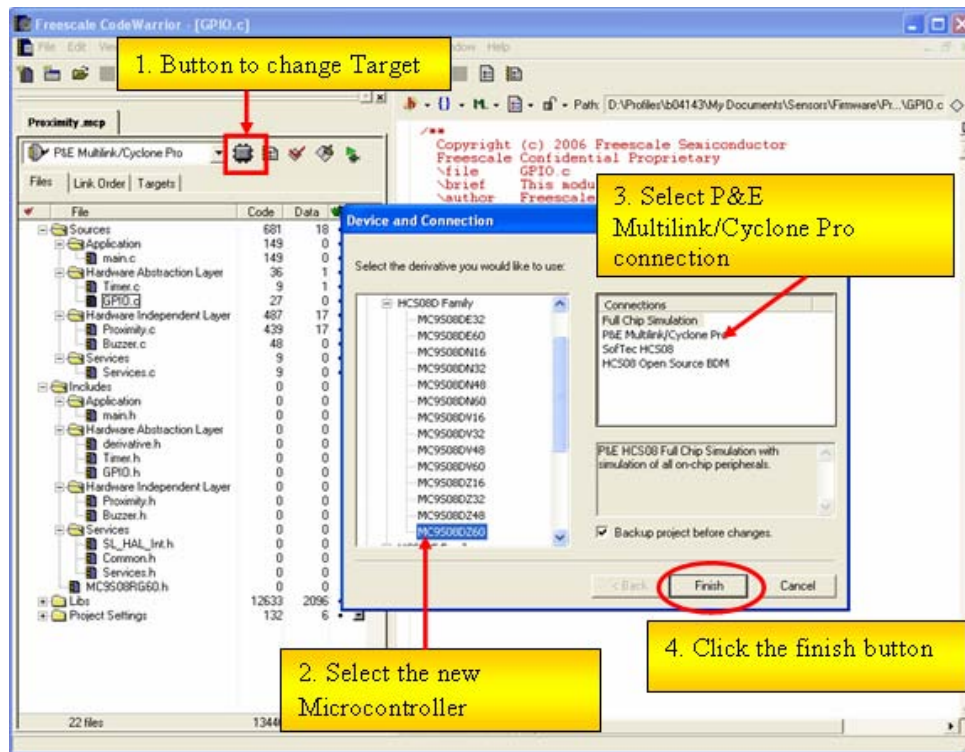
Figure 7. Run Button

8. Now the microcontroller is running the Proximity Sensing Software.

## Power Demo Board

The Proximity Software uses the MC9S08QE128 microcontroller by default. If the user wants to change the board, for example to the DEMO9S08DZ60, follow the next steps:

1. Change the target in CodeWarrior as shown in the steps in the figure below.



**Figure 8. Changing the Target**

2. Compile the Proximity Project.
3. Load the Proximity Software in the DZ60 microcontroller
4. Plug the DEMO9S08DZ60 board with the PROXIMITYBOARD0 board.
5. The Demo board has been changed.



## Building an Application

If a user builds their own application and wants to use the Proximity Software as a guide, some considerations must be taken:

- Change the required GPIOs (this will be explained further in the document). The pins used to connect the electrodes must be I/O.
- Change the Timer interrupt vector number in the Timer.h file, as the timer overflow vector number is different in the S08 families.
- Change the DD\_OFFSET, within the GPIO.h file, as some MCUs have the data direction register in a different memory address.

The next steps explain how to correctly use the pins that will be connected to the electrodes.

1. Open the GPIO.c file that contains all the MCU definitions.
2. Search at the end of the file the line “User defined” (Figure 9). There are two definitions that might be used to write the pin number to be connected to the electrodes. vpPort1 is used to define the port pins that will be connected to the electrodes. vpPort2 is used to define the pin that will drive the buzzer. Remember that Proximity Software uses only eight electrodes, and if it is desired to change the number of electrodes, other steps will be necessary to do. Please refer to AN3579 for detailed information.

```
#else          /** User defined */
const VIRTUAL_PORT vpPort1[] = { {GPIO_PIN(A,0)},
                                  {GPIO_PIN(A,0)},
                                  {GPIO_PIN(A,0)},
                                  {GPIO_PIN(A,0)},
                                  {GPIO_PIN(A,0)},
                                  {GPIO_PIN(A,0)},
                                  {GPIO_PIN(A,0)},
                                  {GPIO_PIN(A,0)} };

const VIRTUAL_PORT vpPort2[] = { {GPIO_PIN(A,1)} };

#endif
```

**Figure 9. GPIO.c File**

3. The next figure shows the GPIO.c file configured to work with the DEMOQE128 board and the PROXIMITYDBOARD0.

```
/** DEMOQE128 */
const VIRTUAL_PORT vpPort1[] = { {GPIO_PIN(A,2)},
                                  {GPIO_PIN(A,3)},
                                  {GPIO_PIN(C,0)},
                                  {GPIO_PIN(C,1)},
                                  {GPIO_PIN(B,3)},
                                  {GPIO_PIN(B,4)},
                                  {GPIO_PIN(B,2)},
                                  {GPIO_PIN(B,5)} };

const VIRTUAL_PORT vpPort2[] = { {GPIO_PIN(E,7)} };
```

**Figure 10. GPIO.c File for DEMOQE128**

Table 2 shows the pins from the DEMOQE128 MCU port that will be connected to PROXIMITYBOARD0 board and were previously configured in Figure 10.

**Table 2. DEMOQE128 MCU Port**

Signal	Pin	Pin	Signal
VDD	1	2	PTA5/RESET
VSS	3	4	PTA5/RESET
PTB1	5	6	PTA4
PTB0	7	8	PTE7
PTA2	9	10	VREFH
PTA3	11	12	VREFL
PTC0	13	14	PTA0
PTC1	15	16	PTA1
PTB3	17	18	PTF0
PTB4	19	20	PTF1
PTB2	21	22	PTA6
PTB5	23	24	PTA7
PTD1	25	26	PTH6
PTD2	27	28	PTH7
PTD0	29	30	PTD4
PTD3	31	32	PTD5
PTC2	33	34	PTD6
PTC3	35	36	PTD7
PTC4	37	38	PTC7
PTC5	39	40	PTC6
PTF2	37	38	PTB7
PTF3	39	40	PTB6
PTF4	39	40	PTG0
PTF5	37	38	PTG1
PTF6	39	40	PTH0
PTF7	39	40	PTH1
PTG2	37	38	NC
PTG3	39	40	NC

## Conclusion

This users guide explains the Proximity Sensing Software and the porting to different Freescale microcontroller platforms.

The software is provided to speed up the design of new applications and to test the Proximity Sensing Technology.

This document describes the required changes for the system to work properly. The example application demonstrates how to use the PROXIMITYBOARD0 and the supported MCU tools. Any new design for a particular application with the required electrodes arrangement should consider this document and reference material below:

- AN3579 - Enabling an MCU for Touch Sensing with Proximity Sensor Software

## Appendix A: Electrode Configuration for Commonly Used Demo Boards

Table 3. Configuration Table for the Demo Boards

Signals	Demo QE128	Demo 9S08QG8	Demo 9S08QD4	Demo 9S08AW60	Demo JM	Demo 9S08RG60	Demo 9S08LC60	Demo 9S08DZ60
Electrode 1	PTA2	PTA2	PTA2	PTG0	PTG0	PTC0	PTC4	PTG2
Electrode 2	PTA3	PTA3	PTA3	PTG1	PTG1	PTC1	PTC5	PTG3
Electrode 3	PTC0	PTA5 <sup>(1)</sup>	PTA5 <sup>(2)</sup>	PTE2	PTE2	PTD6	PTC2	PTD2
Electrode 4	PTC1	PTA0	PTA0	PTE3	PTE3	PTB7	PTC3	PTD3
Electrode 5	PTB3	PTB3	NC	PTE6	PTE5	PTC4	PTB5	PTE4
Electrode 6	PTB4	PTB4	NC	PTE5	PTE4	PTC5	PTB4	PTE5
Electrode 7	PTB2	PTB2	NC	PTE7	PTE6	PTC6	PTB6	PTE3
Electrode 8	PTB5	PTB5	NC	PTE4	PTE7	PTC7	PTB7	PTE2
Time Vector Number	7	7	7	11	15	6	7	11
Buzzer	PTE7/PTA7	PTB7/NC	PTB7/NC	PTG4/PTB7	NC/PTB7	PTB2/PTA1	NC/PTA7	PTF7/PTB0

**NOTES:**

1. This pin is the RESET pin. This pin must not be connected to an electrode, because it operates only as input.
2. This pin is the RESET pin. This pin must not be connected to an electrode, because it operates only as input.

## Appendix B: Jumper Configurations for Demo Boards

The following figures show the jumpers that need to be removed from the supported demo boards.

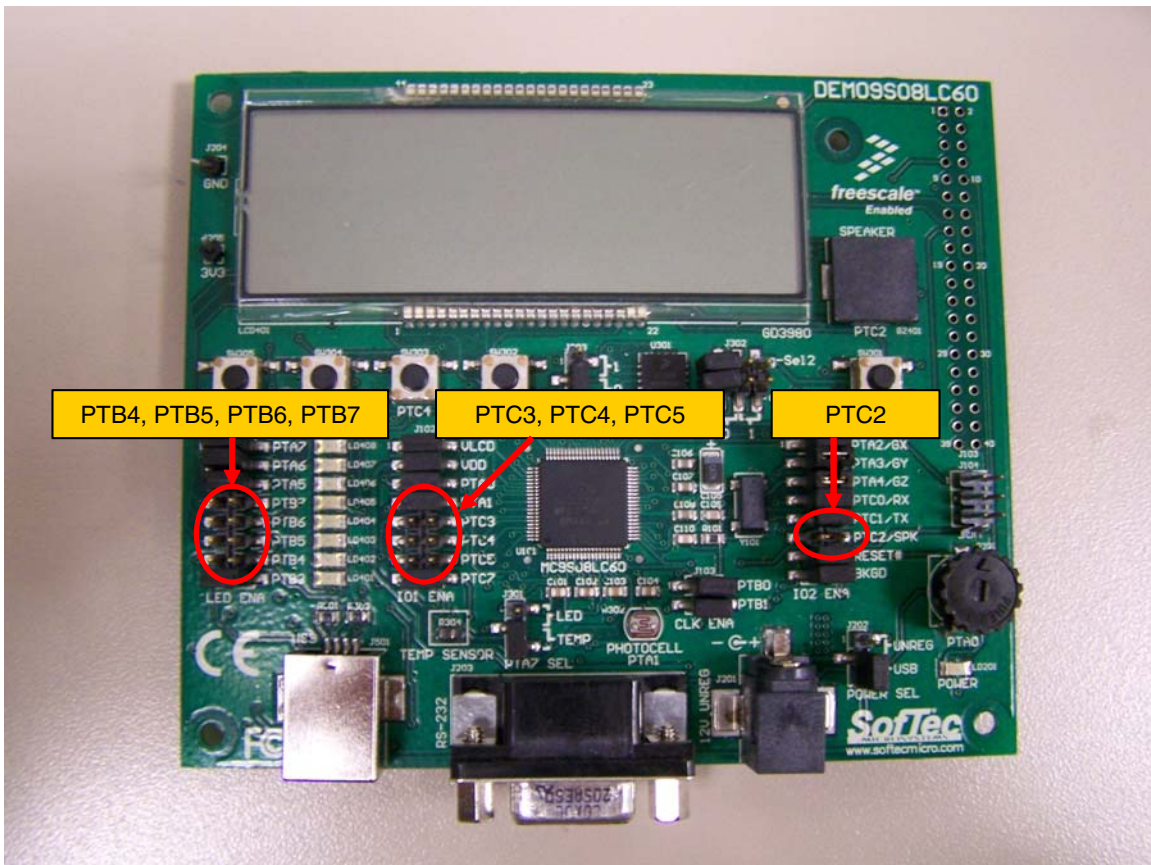


Figure 11. DEMO9S08LC60 Board

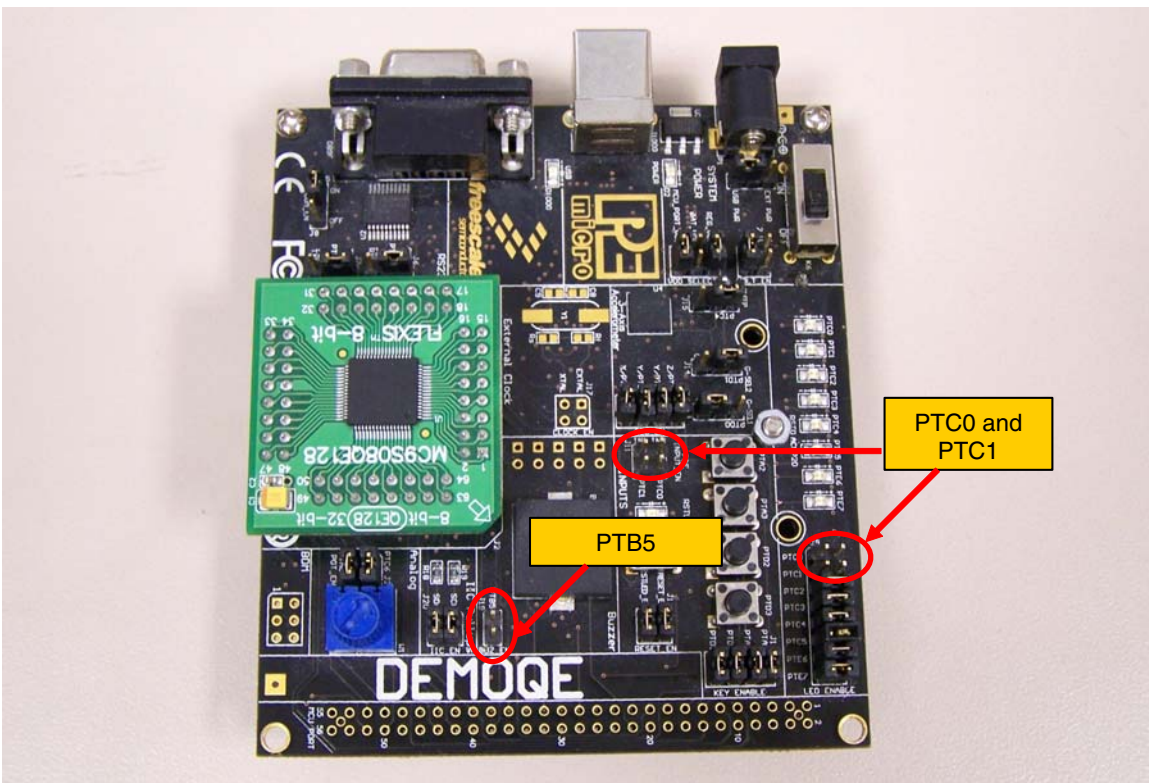


Figure 12. DEMOQE Board

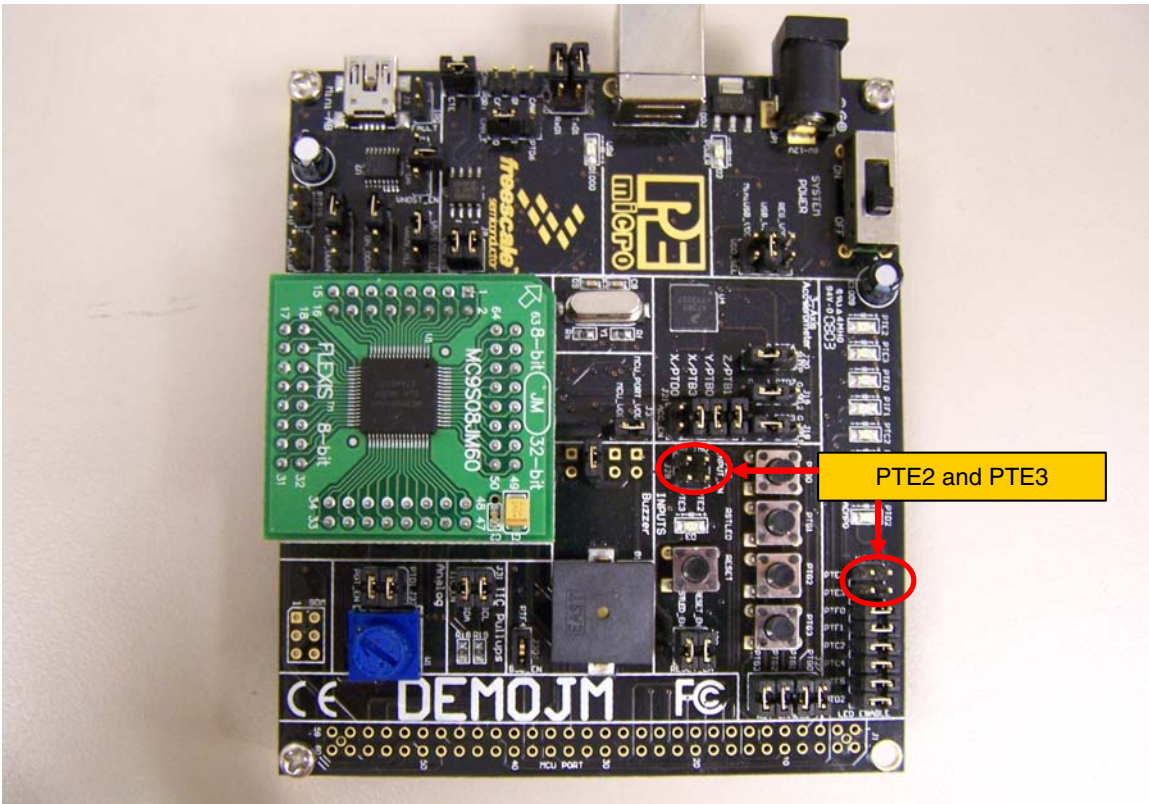


Figure 13. DEMOJM Board

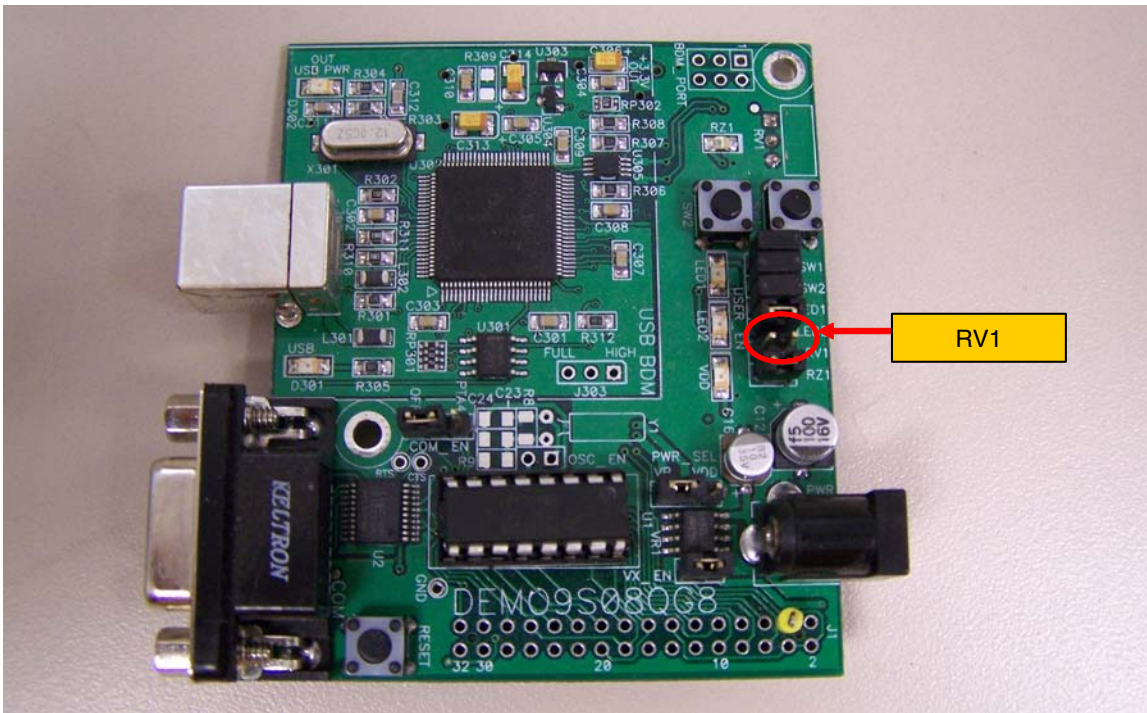


Figure 14. DEMO9S08QG8 Board

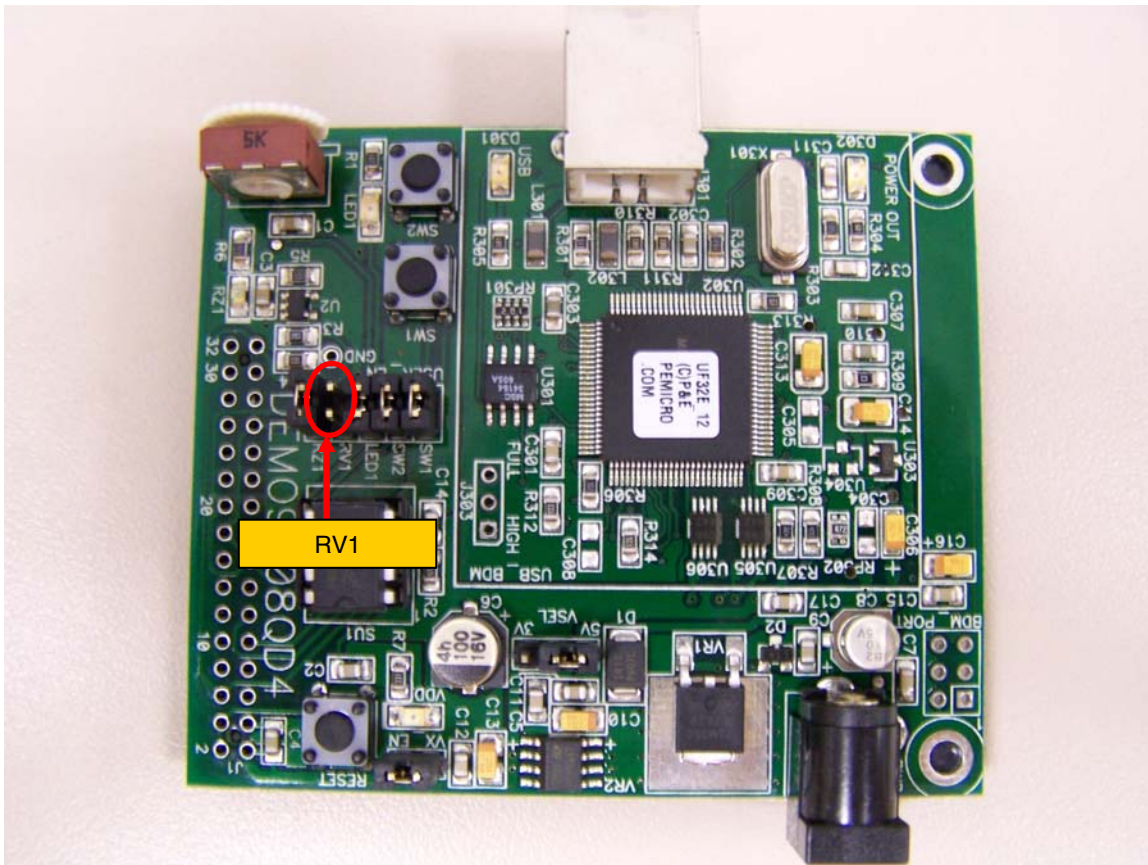


Figure 15. DEMO9S08QD4 Board

In the following boards, the jumpers do not need to be removed.

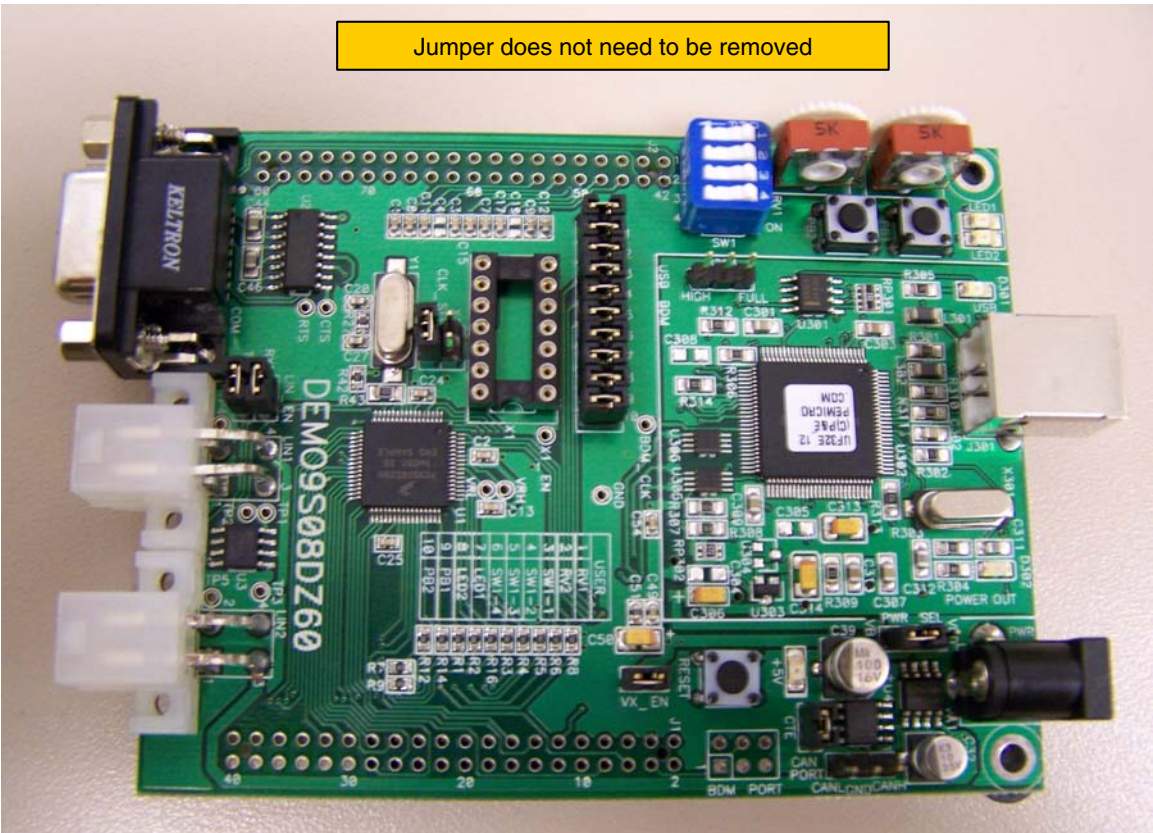


Figure 16. DEMO9S08DZ60 Board

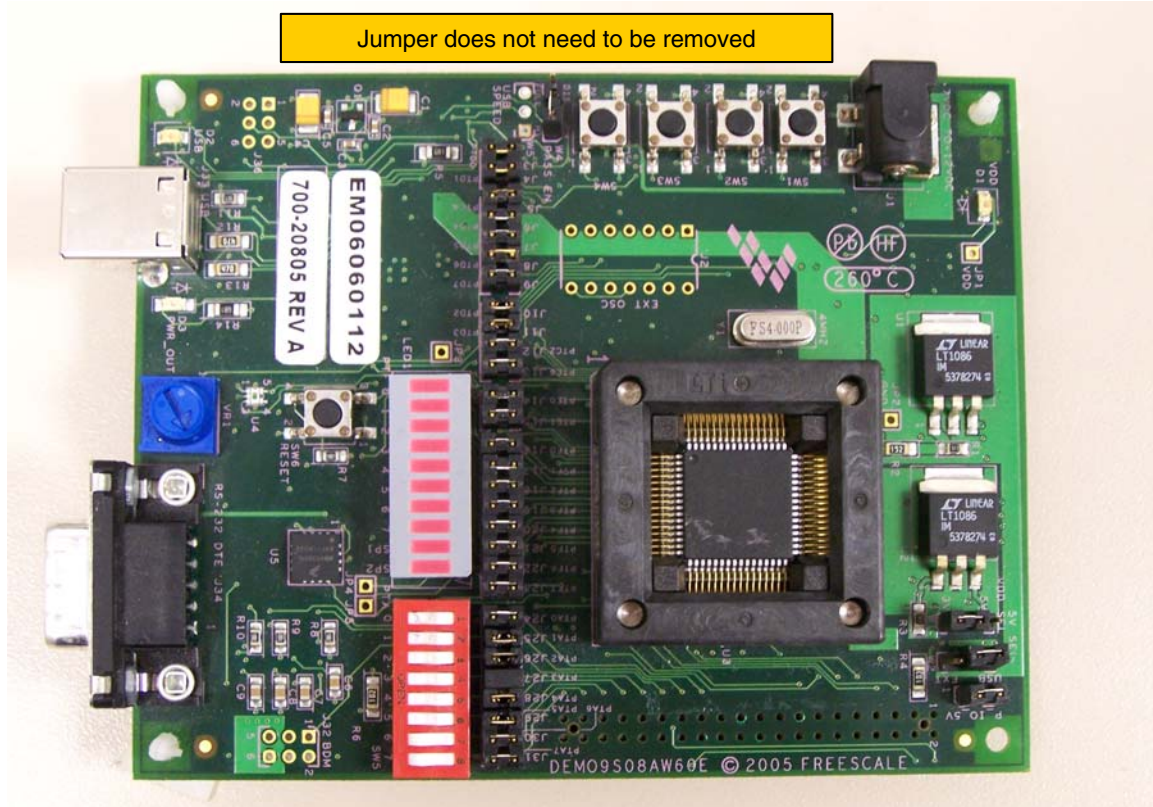
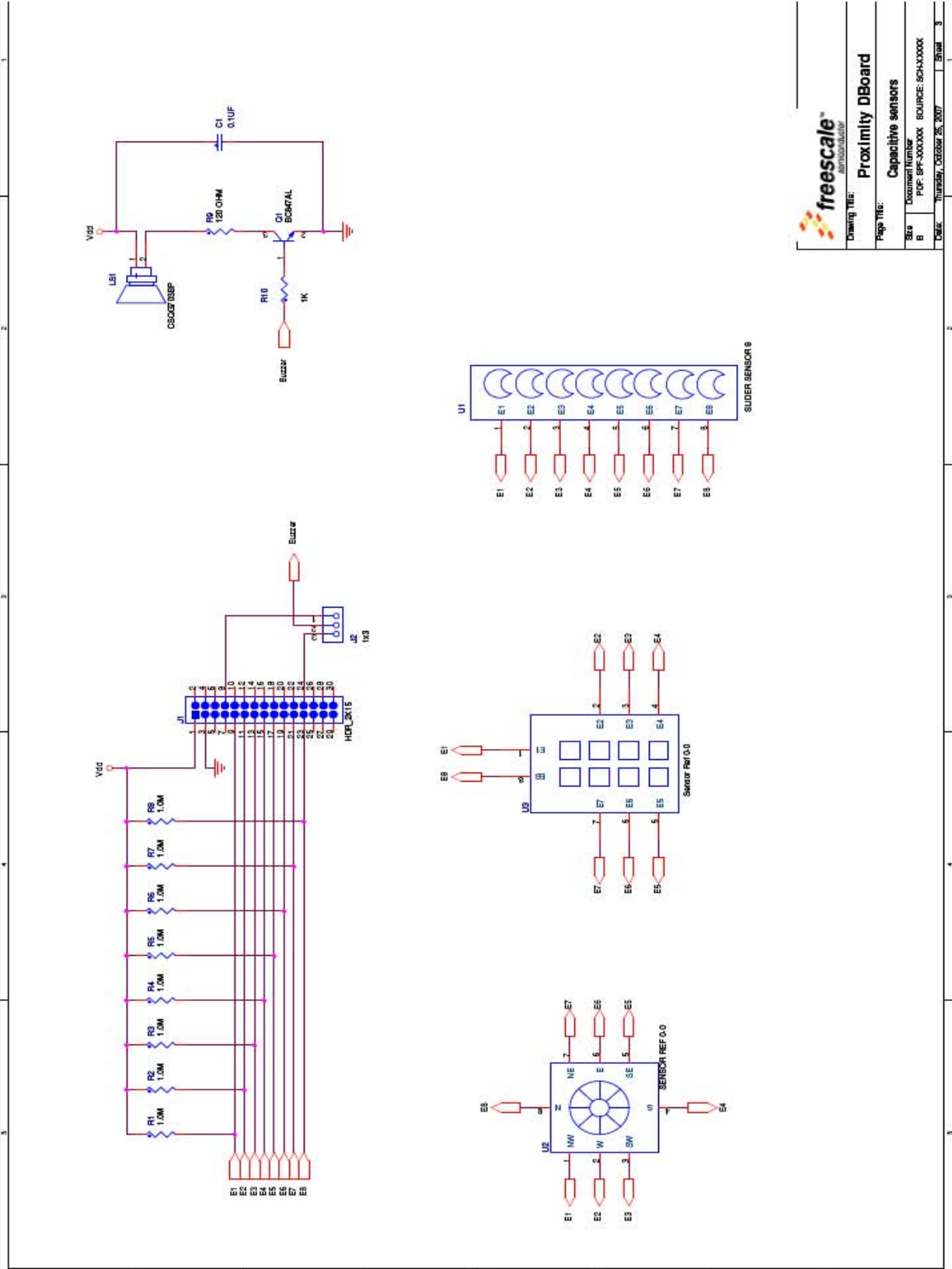


Figure 17. DEMO9S08AW60 Board

# Appendix C: Proximity PROXIMITYBOARD0 Board Schematics



<b>Drawing Title:</b> Proximity DBoard	<b>Page Title:</b> Capacitive sensors
<b>Doc. Number:</b> PAF-BRF-XXX-XXX	<b>Source:</b> SCH-XXXXX
<b>Date:</b> Thursday, October 25, 2007	<b>Sheet:</b> 3

Figure 18. PROXIMITYBOARD0 Board Schematics



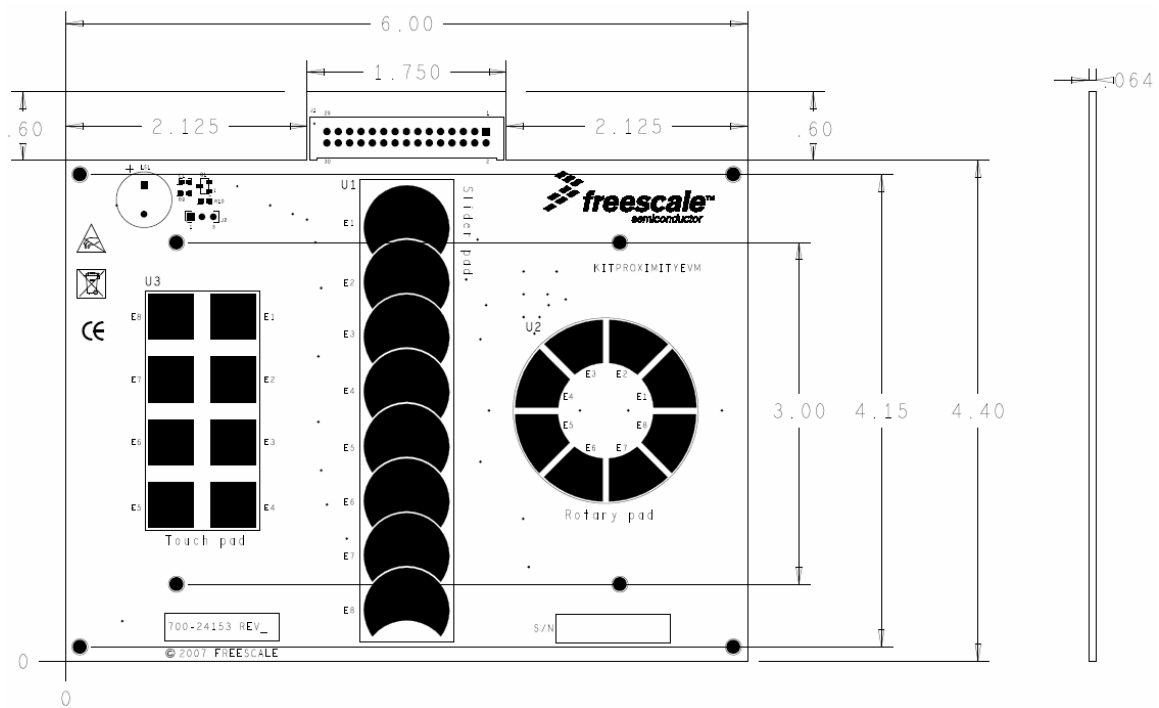


Figure 19. PROXIMITYBOARD0 Board Top View

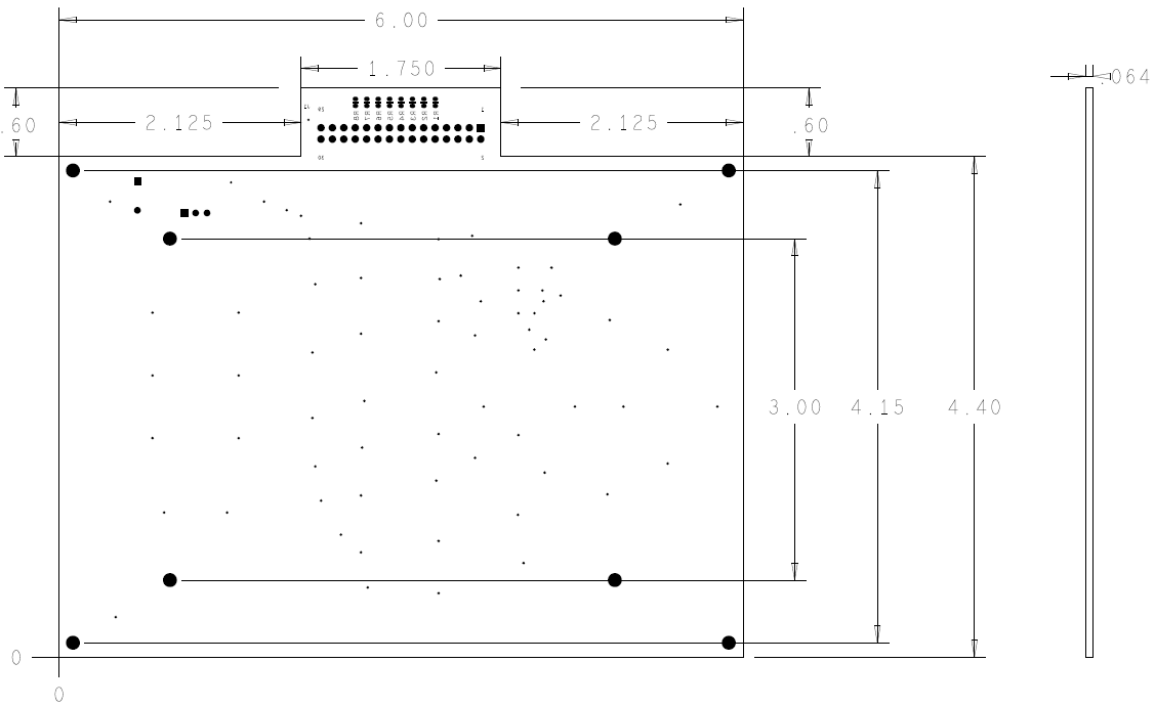


Figure 20. PROXIMITYBOARD0 Board Bottom View

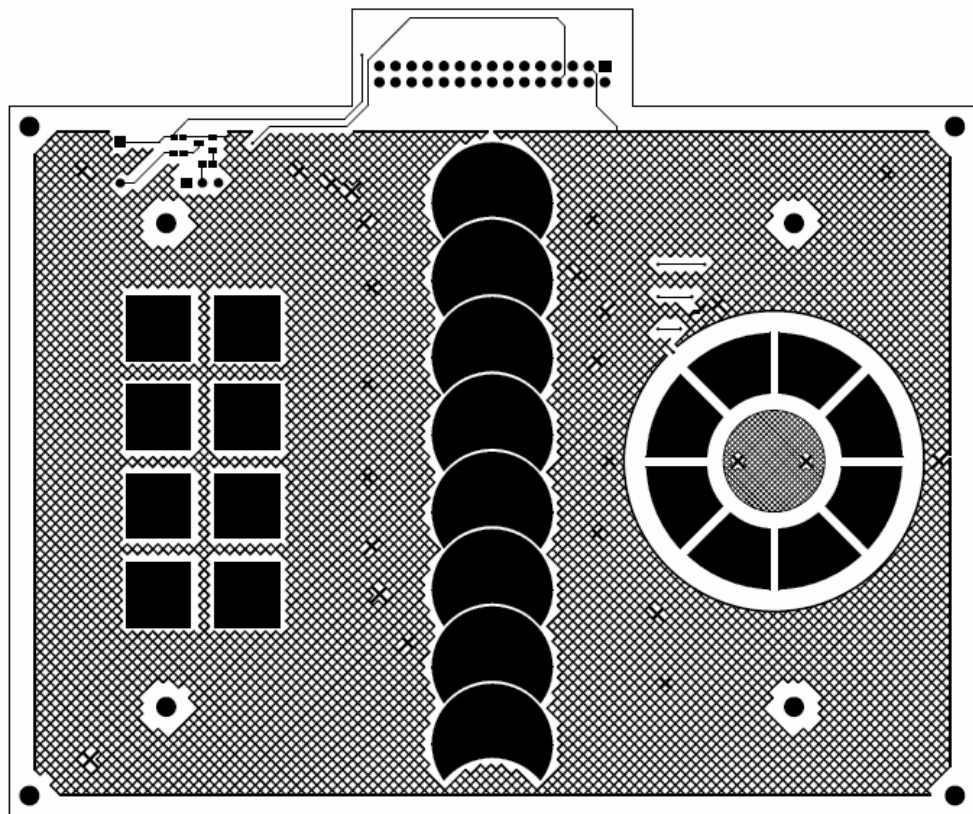


Figure 21. PROXIMITYBOARD0 Board Top Etch View

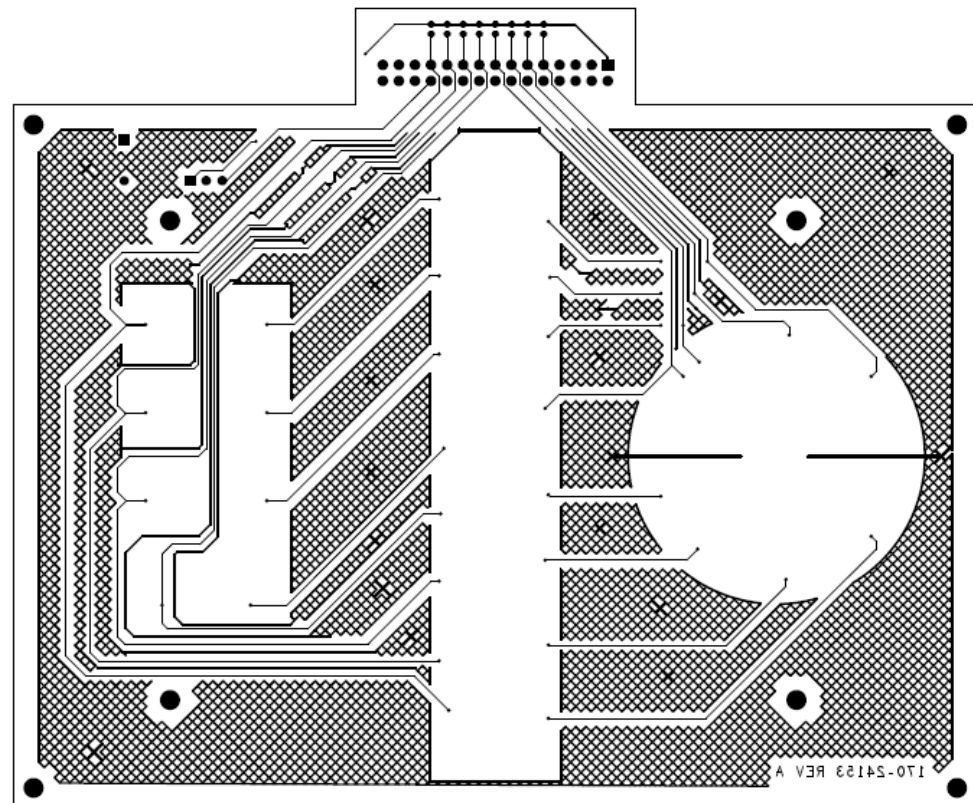


Figure 22. PROXIMITYBOARD0 Board Bottom Etch View

## **How to Reach Us:**

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **Web Support:**

<http://www.freescale.com/support>

### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor, Inc.  
Technical Information Center, EL516  
2100 East Elliot Road  
Tempe, Arizona 85284  
1-800-521-6274 or +1-480-768-2130  
[www.freescale.com/support](http://www.freescale.com/support)

### **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)  
[www.freescale.com/support](http://www.freescale.com/support)

### **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](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor China Ltd.  
Exchange Building 23F  
No. 118 Jianguo Road  
Chaoyang District  
Beijing 100022  
China  
+86 010 5879 8000  
[support.asia@freescale.com](mailto: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 +1-303-675-2140  
Fax: +1-303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.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.

© Freescale Semiconductor, Inc., 2008. All rights reserved.