

Figure A-1. Freescale eGUI/D4D block diagram

A.2.2 File structure

The following figure shows the file structure of eGUI/D4D, created from five types of files:

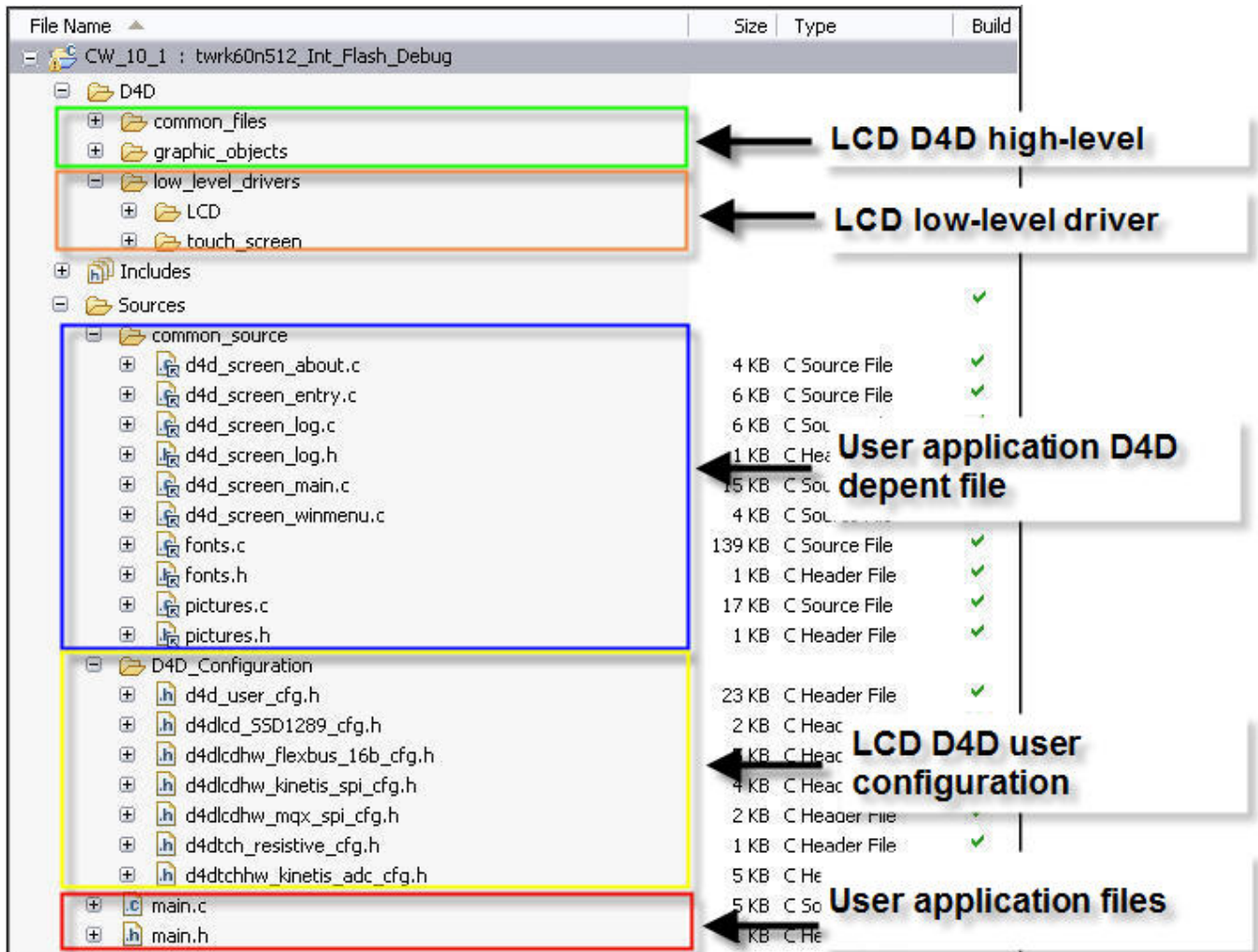


Figure A-2. eGUI/D4D file structure in CodeWarrior project.

The above project can be downloaded from Freescale Embedded GUI Software, available on freescale.com.

A.2.3 Adding eGUI library to a MQX RTCS application

Using the Freescale Embedded GUI Software, the user must add LCD D4D high-level driver, LCD low-level driver for Kinetis, and LCD D4D user configuration to the MQX RTCS application. The user can add D4D files either using “add files...” by drag-and-drop or right-clicking to the destination folder.

NOTE

These steps also apply to any MQX Project using CodeWarrior 10.2.

1. Add LCD D4D high-level driver to the RTCS MQX project. Once the files from Freescale_embedded_GUI_SW.zip file are extracted, the LCD D4D high-level driver files (common_files and graphic_objects folders) can be found at the D4D folder.

Freescale embedded graphical user interface (eGUI)

2. Since the Kinetis K60 device and TWR-LCD module are used for this project, the user needs to add only the low-level drivers for Kinetis K60 device and the screen module of the TWR-LCD. The following figure shows the drivers that must be added.

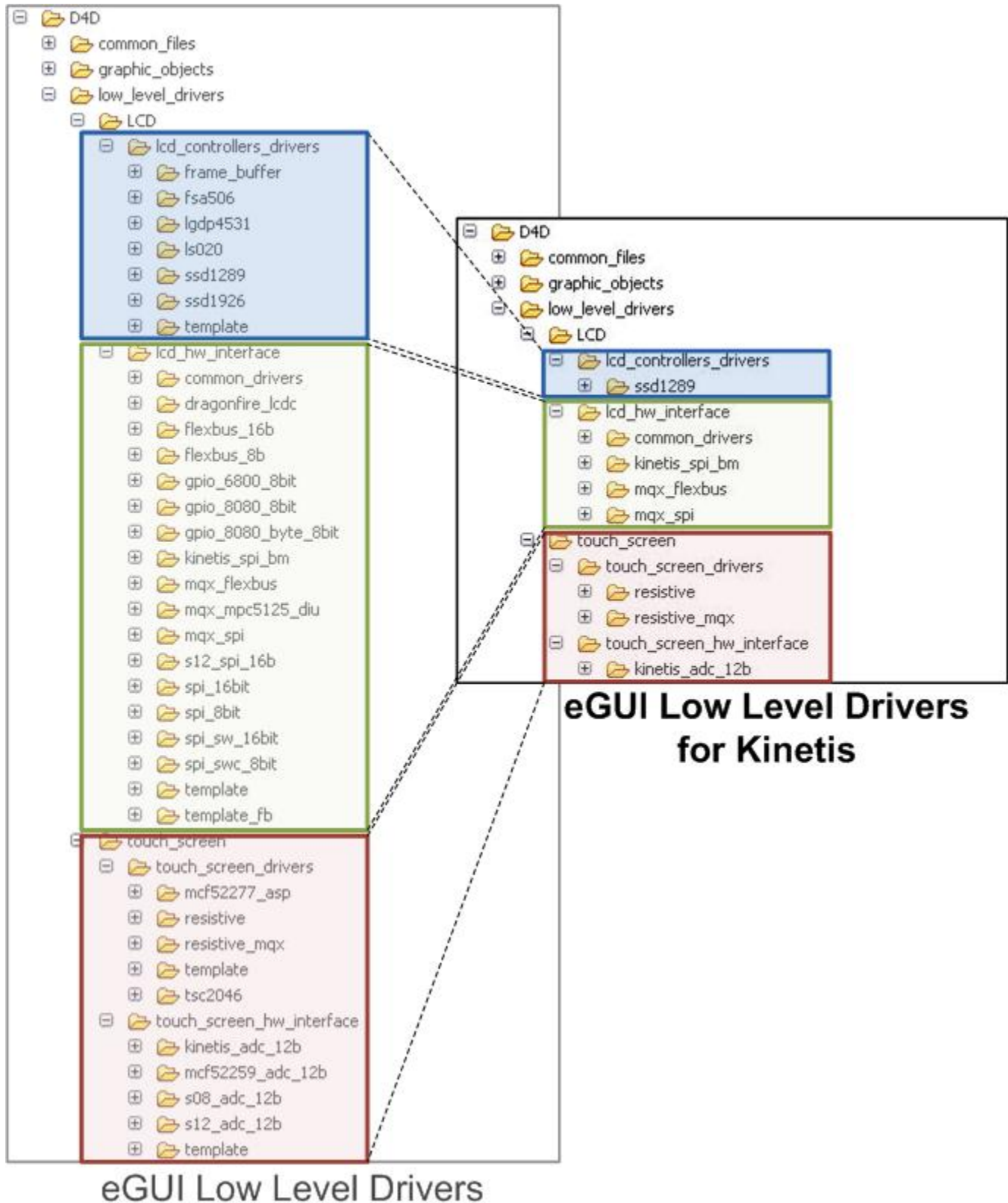


Figure A-3. eGUI low-level drivers for Kinetis.

3. Add LCD D4D user configuration files to the Sources folder of the MQX RTCS project. The LCD D4D user configuration files (D4D_configuration) can be found at this path ...*Official_Demos\EGUI_D4D_Demo\TWR_K60N512\MQX_3_7CW_10_1\Sources*.
4. Include the D4D library into the project by adding the below paths in the compiler settings (right-click and select Properties of the MQX RTCS Project). [Figure A-4](#) shows the windows settings where the path must be added.

```
"${PROJECT_LOC}/D4D"
"${ProjDirPath}/Sources"
"${PROJECT_LOC}/Sources/common_source"
"${PROJECT_LOC}/Sources/D4D_Configuration"
```

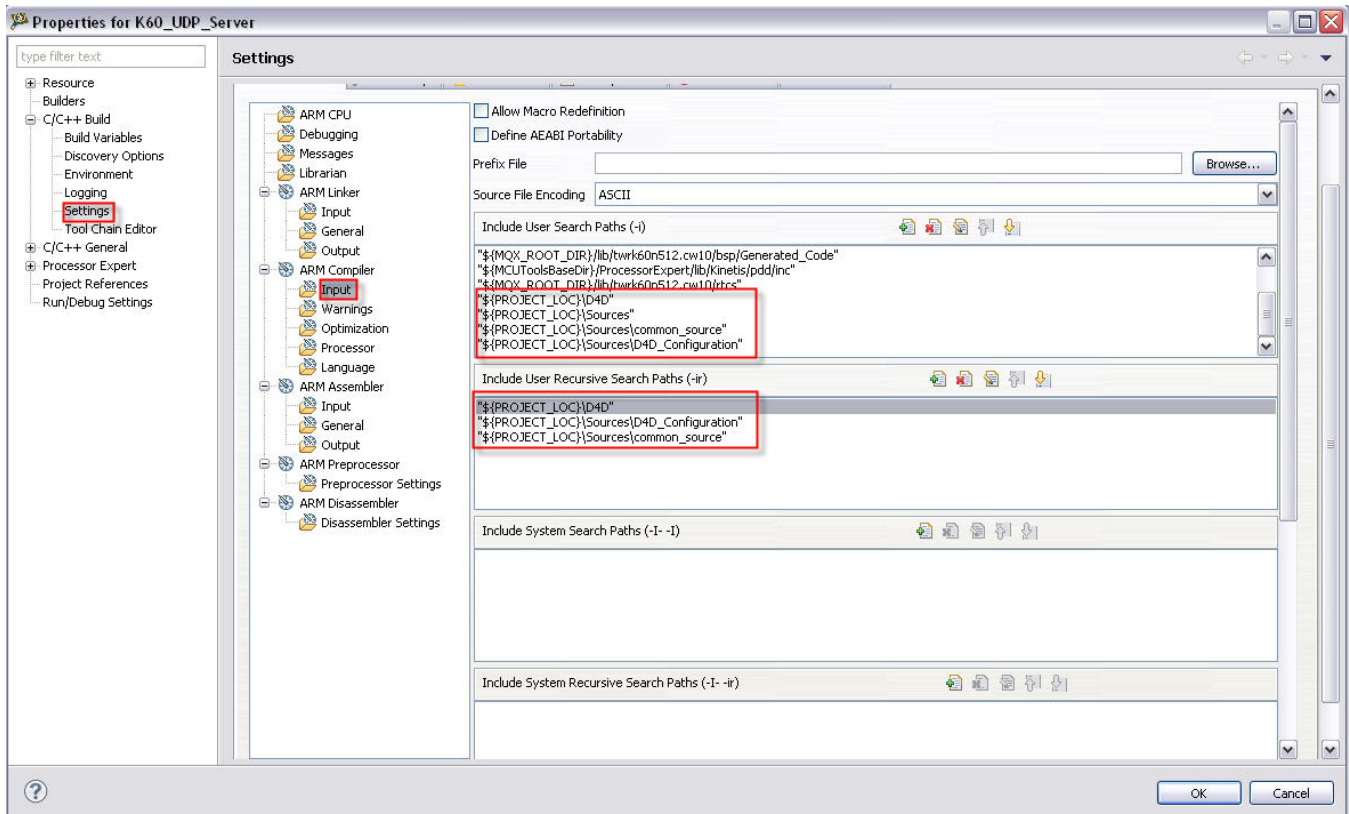


Figure A-4. User and recursive paths

5. Add the eGUI tasks to the MQX RTCS project. There are two tasks that must be added to keep eGUI alive:
 - Time task: This task periodically checks for events every 25 ms.
 - LCD task: This task handles the D4D display. The MQX RTCS project was made on MQX 3.8. The MQX developer team changed the touch screen driver (TCHRES) and added the feature to use two different ADC modules for reading X/Y axis. For this reason, the LCD task must be modified to work on MQX 3.8
6. Create the first Screen. The basic item of D4D structure is SCREEN. The screen represents the real one screen shown on LCD.

The screen contains:

 - List of all used objects on screen.
 - Functions bodies:
 - OnInit – One time called function with first use of screen
 - OnActivate – Function called on each activation of screen
 - OnDeactivate – Function called before deactivation of screen
 - OnMain – Function is called periodically when screen is active
 - OnObjectMsg - Function is called with all system messages of this screen

Create a new folder named as “common_source” in Source folder as shown in the figure below.

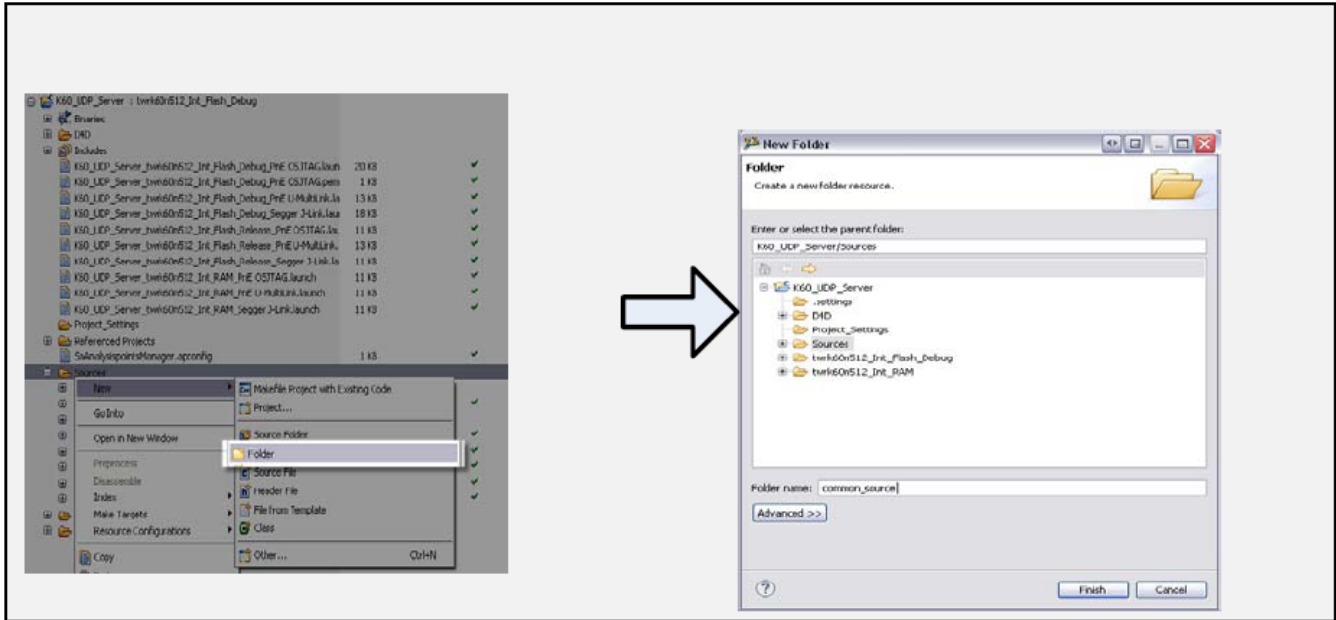


Figure A-5. Create a new folder.

1. Create a new c file named as “HOME.c” in common_source folder as shown in the following figure.

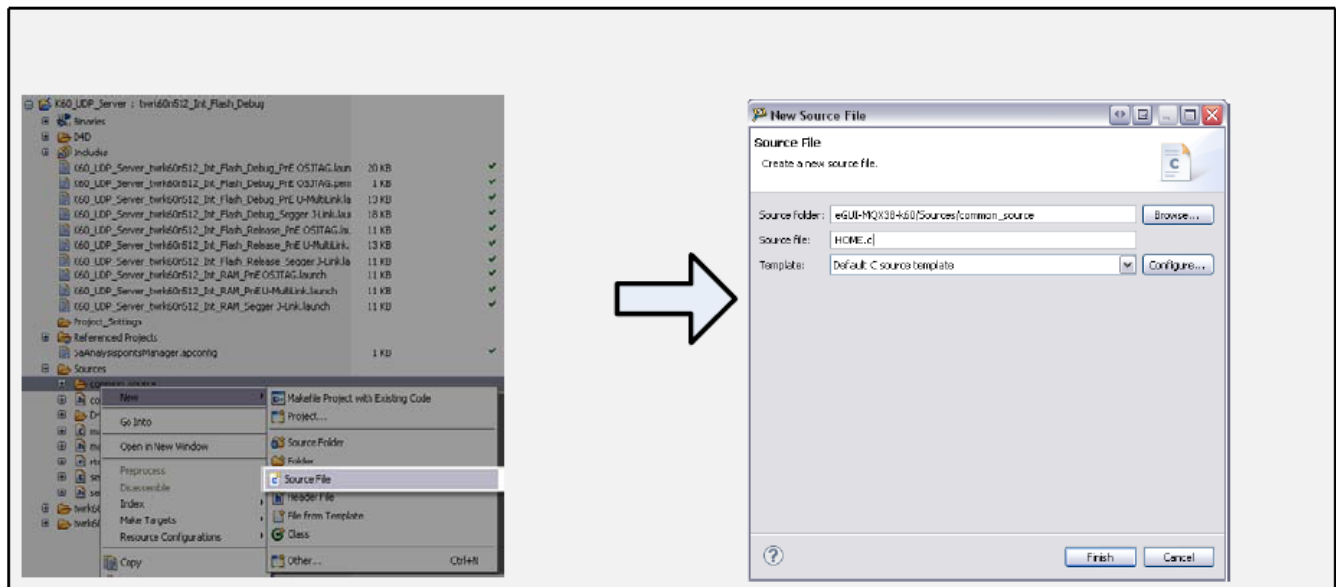


Figure A-6. Create a new C file.

2. Open the d4d_screen_template.c.template file at this path `...\D4D\configuration_example` from Freescale Embedded GUI software, downloaded from freescale.com.
3. Copy the code from d4d_screen_template.c.template file and paste it into HOME.c file.
4. Replace the name of the template screen from screen_template to screen_home.

```
D4D_DECLARE_STD_SCREEN_BEGIN(screen_home, ScreenHome_)
```

5. Find and replace of all the functions bodies from ScreenTemplate_ to ScreenHome_.
6. Add font.c and font.h from this path `_Official_Demos\EGUI_D4D_Demo\common_source` to common_source folder. font.c and font.h files contain two font types: Arial and Berlin Sans. For this project, only Arial and Berlin Sans fonts are used but using the Freescale Embedded GUI Image Converter Utility, the user can add installed Windows fonts. This utility can be downloaded from freescale.com searching for Freescale Embedded GUI Image Converter Utility.

7. At this point, the project structure must be like as shown in the following figure.

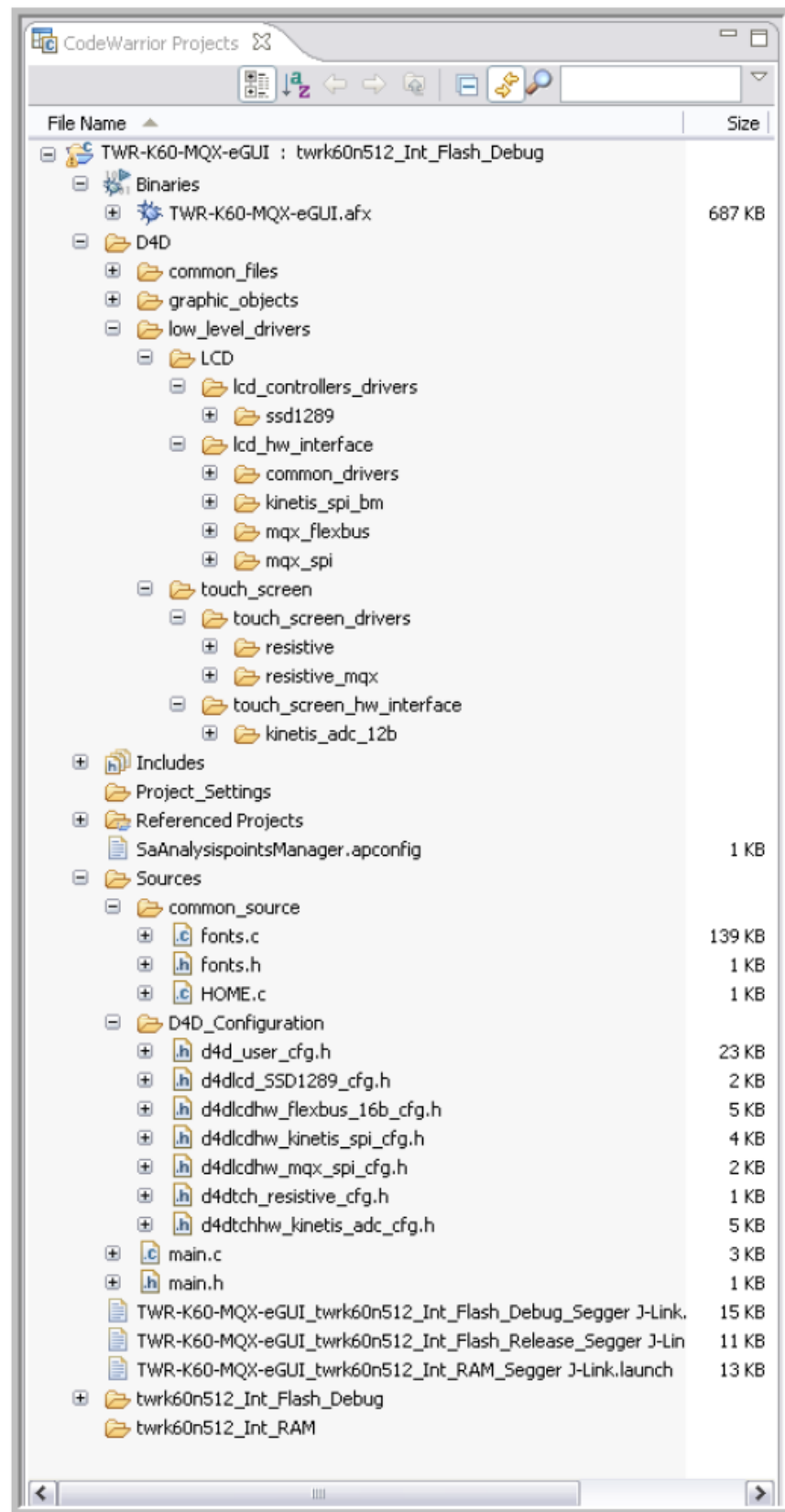


Figure A-7. MQX project with eGUI library.

Freescal embedded graphical user interface (eGUI)

8. Build and debug MQX project as shown in the figure below.

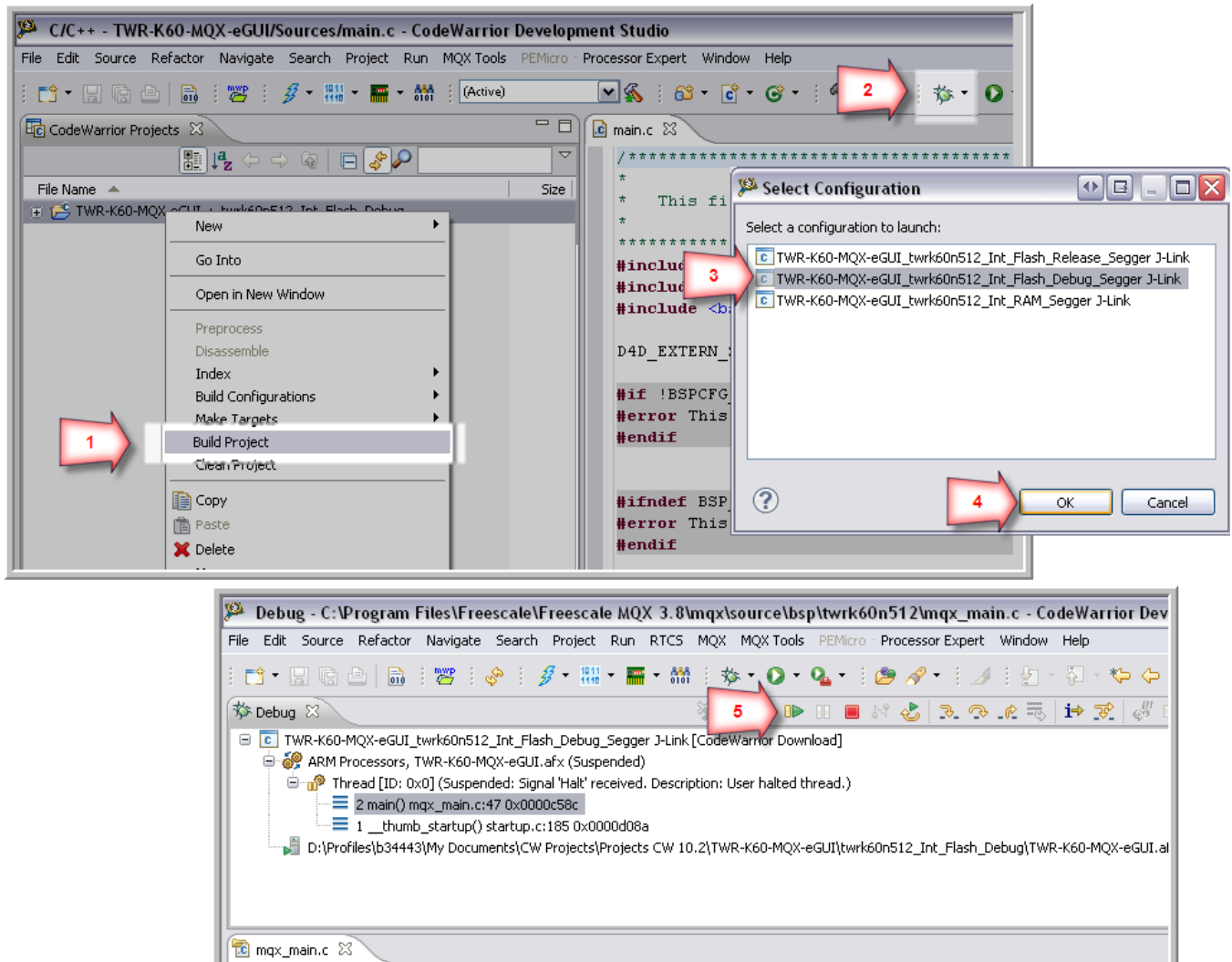


Figure A-8. Debugging a project.

9. When the user clicks the Resume icon, the TWR-LCD will display the calibration screen as shown in the following figure.

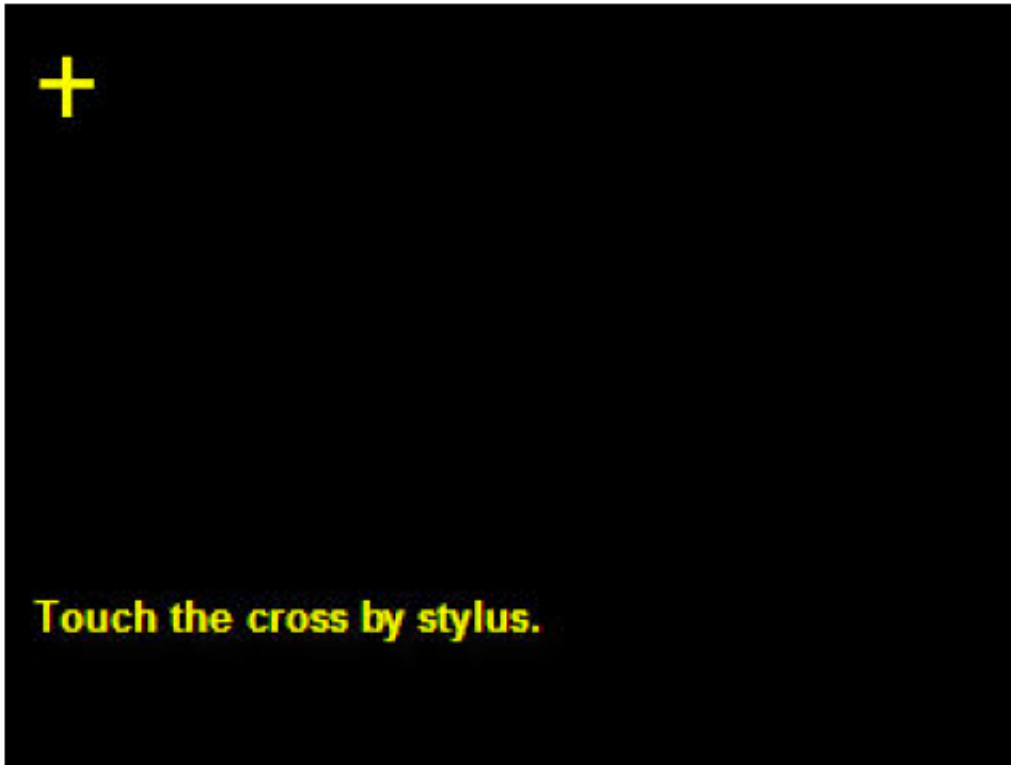


Figure A-9. eGUI screen calibration.

A.2.4 Developing a graphic user interface for the MQX RTCS application using eGUI library

At this point, it is possible to add as many screens and D4D objects as the user application requires.

The following section explains how to build five screens and how to use the objects needed in this particular application.

A.2.4.1 Creating the Home screen

This section explains how to build the home screen which is the screen that is shown on the TWR-LCD after the RTCS is initialized and the server binds a socket.

Using D4D_Button object, five buttons are added into the Home screen. These buttons are used to navigate among the five screens (1 button for Home screen and 4 for each of the rooms). D4D_PICTURE is used to add a picture as the home screen background. The following figure shows the Home screen.

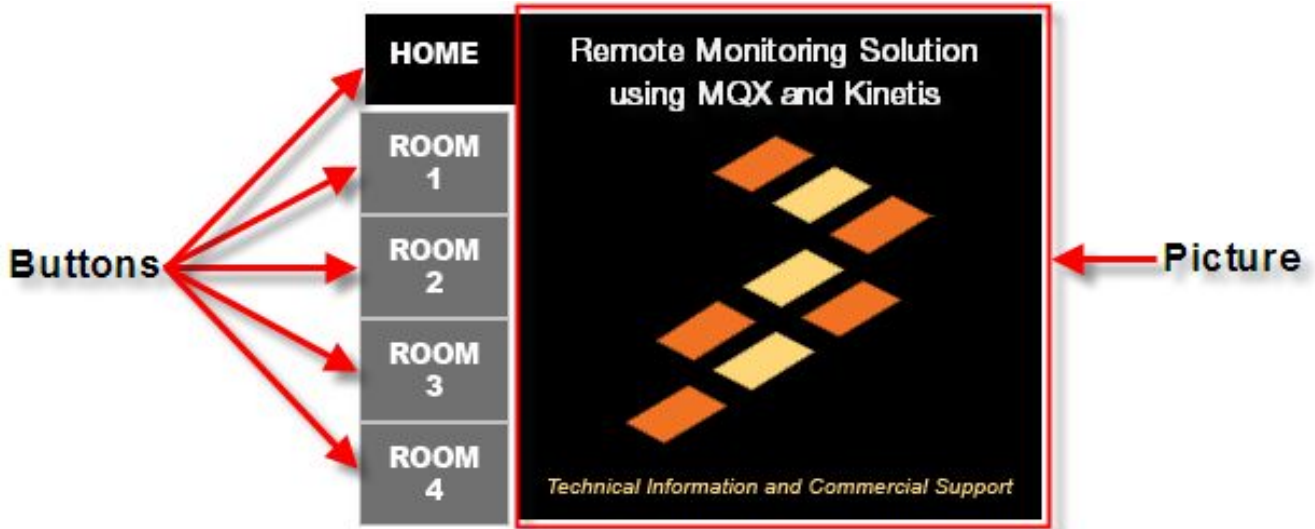


Figure A-10. Home screen.

1. D4D_PICTURE is an object used to show a picture on the screen. The Home screen background is shown in Figure A-11. This picture was created using a graphic editor software and was converted using Freescale Embedded GUI Image Converter Utility, that can be downloaded from freescale.com.



Figure A-11. Home screen background picture.

To add a picture, the macro `D4D_DECLARE_STD_PICTURE` is defined. As an example, to add the Home screen background picture, the macro is defined as follows.

```
D4D_DECLARE_STD_PICTURE (HOME, 73, 0, &bmp_Screen_Home)
```

- The name of the picture is HOME.
- The position is 73 in X-axis and 0 in Y-axis.
- The pointer to the bitmap array of the picture is bmp_Screen_Home.

NOTE

To convert an image into an array; Freescale provides a tool called Embedded GUI Image Converter Utility which can be downloaded from freescale.com. Also, see EGUICUG : Freescale Embedded GUI Converter Utility 2.0 Quick - User's Guide, available on freescale.com.

2. D4D_BUTTON object is intended to be used as a standard button and on this particular project is used to switch among different screens. Each of the five buttons of the Home screen has its own background picture, and as the Home Screen background, the buttons' background pictures were converted using "Freescale Embedded GUI Converter Utility". The button background pictures are shown in the following figure.



Figure A-12. Button background pictures in Home screen.

To add buttons, the macro `D4D_DECLARE_STD_BUTTON_AUTOSIZE` is used. The Room 1 button is described as an example below:

```
D4D_DECLARE_STD_BUTTON_AUTOSIZE(Room1_SH, NULL,
0, 48, &bmp_Room1_Button_Inactive, &bmp_Room1_Button_Inactive, NULL, Room1_Button)
```

- The name of the button is `Room1_SH`.
- There is no text used on this button.
- The position is 0 in X-axis; and 48 in Y-axis.
- The pointer to the bitmap array of the button is `bmp_Room1_Button_Inactive`.
- There is no font used on this button.
- The callback function when this button is pressed is `Room1_Button`.

The button object contains a few predefined constants that are used in a standard button declaration.

The screen behavior and visual aspect flags:

- `D4D_BTN_F_DEFAULT`: This is a help macro that is used for default configuration. The value of this macro is defined as follows:

```
D4D_OBJECT_F_VISIBLE |
D4D_OBJECT_F_ENABLE |
D4D_OBJECT_F_TABSTOP |
D4D_OBJECT_F_TOUCHENABLE |
D4D_OBJECT_F_FOCUSRECT
```

For this particular application, the `D4D_BTN_F_DEFAULT` flags were changed as below:

- `D4D_BTN_F_DEFAULT` value:

Freescal embedded graphical user interface (eGUI)

```
D4D_OBJECT_F_VISIBLE |  
D4D_OBJECT_F_ENABLED |  
D4D_OBJECT_F_TABSTOP |  
D4D_OBJECT_F_TOUCHENABLE |  
D4D_OBJECT_F_FASTTOUCH
```

D4D_BTN_F_DEFAULT flag can be modified in d4d_button.h file.

NOTE

For more information about the D4D flags, see eGUI/D4D Public Predefined Init General Object Flags section from the Freescal Embedded GUI User Manual that can be downloaded from [freescale.com](http://www.freescale.com) searching for “Freescal Embedded GUI (D4D)”.

A.2.4.2 Creating the Room screens

This section explains how to build the Room screen. Only Room 3 screen is explained as rest of the room screens are duplicated. The difference among the Room screens is that Room 1 shows data from client ID 1, Room 2 shows data from client ID 2, and so on. When Room 3 is active, it sends a signal to client ID 3 for starting the transmission and the rest of the clients are in stand-by.

As in Home screen, the Room screens have a five-button column to navigate the screens.

- D4D_Picture: This object is used to show the Room screen background.
- D4D_Label: This object shows the beats per minute and it is updated every 100 milliseconds.
- D4D_Graphic: This object prints the QRS complex sent by the client ID 3.



Figure A-13. Room screen

1. D4D_Picture: The Room screen background is shown in [Figure A-14](#) and the D4D_Picture configuration is the same as in home screen background in [Creating the Home screen](#).

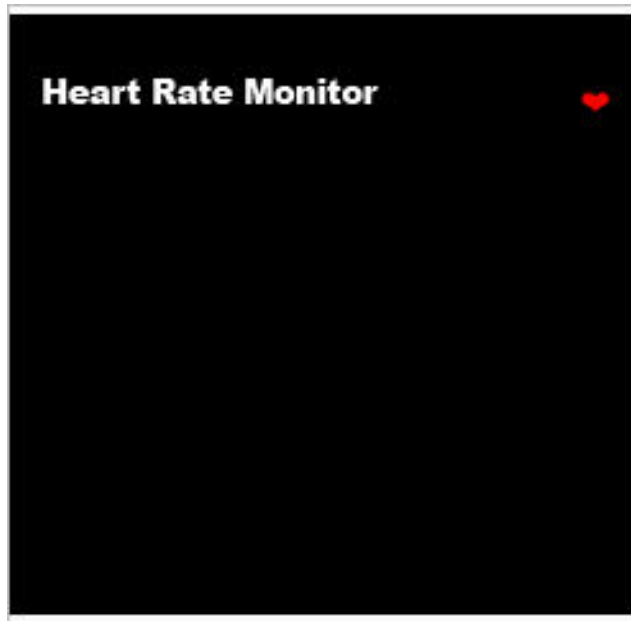


Figure A-14. Room screen background

2. D4D_Button: The button's background pictures are shown in [Figure A-14](#) and the D4D_Button configuration is explained in [Creating the Home screen](#).
3. D4D_Label: This object is prepared to be used as a visualization object to show simple text one-line information of an application in a graphical form. In this particular application, label shows the heart beats rate and is updated every 100 milliseconds.

To add label object, the macro D4D_DECLARE_LABEL is used. The label object is described below:

```
D4D_DECLARE_LABEL(BPM_R1, " ", 230, 5, 62, 40, D4D_LBL_F_DEFAULT, NULL,
FONT_BERLIN_SANS_FBDEMI12_BIG, NULL, NULL)
```

- The name of the label is BPM_R1.
- The string is " " on this label.
- The position is 230 in X-axis; and 5 in Y-axis.
- The size of the label is 62x40 pixels.
- The flags have the value of the macro D4D_LBL_F_DEFAULT.
- The default color scheme is used.
- The font type for the label is Berlin Sans 12.
- There is no callback function.

Every 10 milliseconds, the Room1_OnMain() function is called and updates the label value received from the client ID 1 with D4D_SetText as below:

```
D4D_SetText(&BPM_R1, bpm);
```

Where &BPM_R1 is the pointer to the label object and bpm is the string with the value received from the client ID 1.

4. D4D_Graph object is prepared to create a simple graph. The graph object definition macro is created from three individual parts that allow indicating a complete graph object with various counts of graph traces:
 - Begin part—This part specifies all the necessary parameters of the graph object itself.
 - Add trace to graph—This part allows to use multiple D4D_DECLARE_GRAPH_TRACE macros to add all traces into the graph.
 - End part—This part is used only to close the graph traces array definition.

In this project, graph object shows the QRS complex signal in one trace. This signal is sent by the client ID 3 every 10 milliseconds.

```
D4D_DECLARE_STD_GRAPH_BEGIN(room1_graph, "Room 1", 85, 50, 220, 185, 8, 4, 20,  
FONT_ARIAL7_WIDE, FONT_7)  
D4D_DECLARE_GRAPH_TRACE(dataTraceR1, D4D_COLOR_GREEN, D4D_LINE_THICK,  
D4D_GRAPH_TRACE_TYPE_LINE)  
D4D_DECLARE_GRAPH_END()
```

- The name of the graph is room1_graph.
- The string that appears as graph title is "Room 1".
- The position is 85 in x-axis; and 50 in Y-axis.
- Size of the graph is 220x185 pixels.
- The number of grid lines is 8 in X-axis and 4 in Y-axis.
- The Length of data buffers is 20.
- The font type for the graph title is FONT_ARIAL7_WIDE.
- The flags have the value of the macro D4D_GRAPH_F_DEFAULT
- The name of the trace is dataTraceR1.
- The color of the trace is green.
- The type of the trace D4D_LINE_THICK.

For this particular application, the D4D_GRAPH_F_DEFAULT flags were changed as below:

```
D4D_GRAPH_F_DEFAULT value:  
D4D_OBJECT_F_VISIBLE |  
D4D_OBJECT_F_ENABLED |  
D4D_OBJECT_F_FOCUSRECT |  
D4D_GRAPH_F_MODE_ROLLOVER
```

D4D_GRAPH_F_DEFAULT flag can be modified in d4d_user_cfg.h file.

The rest of the Room screens are duplicated with Room1 screen.

For the complete application project, see AN4644SW.zip file, available on [freescale.com](http://www.freescale.com).

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

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

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
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

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors 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 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 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 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 claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2013 Freescale Semiconductor, Inc.