The jumper settings on the MCU Tower module are set to enable operation of the Quick Start application programmed into the flash memory of the MC9S08LL64 MCU.
Introduction

The MC9S08LL64 device is Freescale’s low-power microcontroller with an integrated liquid crystal display (LCD) driver. TWR-S08LL64 contains an on-module display that allows developers to explore software development using the integrated LCD driver. This Lab Tutorial guide is designed to get you ready to develop your next LCD application using the MC9S08LL64 within minutes.

Note:

This Lab Tutorial can be followed once the steps in the Quick Start Guide, installing all of the software and documentation, have been finalized.

Lab Using Quick Start Code, Open CodeWarrior and the Project

This lab will highlight the capabilities of the MC9S08LL64 microcontroller with the application provided by the Quick Start Lab. Pushing switch SW2 will step you through these four states.

1. Low power operation with time display
2. Potentiometer vs. light sensor comparison
3. Accelerometer demo, X, Y and Z output to SCI
4. ADC demo: when in the ADC demo, pushing SW4 will increment the channel converted and displayed

1. Install software and tools as directed in the application provided by the Quick Start Lab. This lab will highlight the capabilities of CodeWarrior and the Project CodeWarrior and Lab Using Quick in the Quick Start Guide, installing all of the software development using the integrated display that allows developers to explore low-power microcontroller with an integrated liquid crystal display (LCD) driver. TWR-S08LL64 contains an on-module display that allows developers to explore software development using the integrated LCD driver. This Lab Tutorial guide is designed to get you ready to develop your next LCD application using the MC9S08LL64 within minutes.

Note:

This Lab Tutorial can be followed once the steps in the Quick Start Guide, installing all of the software and documentation, have been finalized.

Lab Using Quick Start Code, Open CodeWarrior and the Project

This lab will highlight the capabilities of the MC9S08LL64 microcontroller with the application provided by the Quick Start Lab. Pushing switch SW2 will step you through these four states.

1. Low power operation with time display
2. Potentiometer vs. light sensor comparison
3. Accelerometer demo, X, Y and Z output to SCI
4. ADC demo: when in the ADC demo, pushing SW4 will increment the channel converted and displayed

1. Install software and tools as directed in the Quick Start Guide.

TWR-S08LL64—Lab Tutorials 1 and 2 (sheet 1 of 2)

Set Up

TWR-S08LL64 Module

1. Connect the 10-pin connector to the 10-pin header on the module labeled COM PORT and J3 (see Figure 4). During states 2, 3 and 4 data is sent out the SCI port on the MCU to the terminal console on your computer.
2. Connect a RS232 cable from the DB9 pinjail to your computer serial port.

Start P&E Toolkit Application and Enter the Programmer/Debugger

1. Open P&E Multilink Toolkit Launch Pad and Terminal Window. From Windows start menu, select “Programs > P&E Embedded Multilink Toolkit > Toolkit Launchpad”.
2. Start a terminal console and configure your computer’s com port for 19200 baud, 8 bit, 1 stop, no parity.
3. Using CodeWarrior, compile and program the MC9S08LL64 microcontroller with the application by clicking on “Debug” button or hitting F5 on your keyboard, launching the programmer and the open source BDM debugger.
4. If the message “Load Executable File” appears, click “Yes.”
5. If the messages “Set Connection” and “Set Derivative” appear, set HCS08 – FSL Open Source BDM and derivative to “MC9S08LL64” and click “OK.”
6. If the message “Setting HCS08 and FSL Open Source BDM and derivative to mass erase the non-volatile memory of the current device, then reprogram the application will stop execution of the current one” appears, click “OK.”
7. When the message, “Loading a new application will stop execution of the current one” appears, click “OK.”
8. When the message, “The debugger is going to mass erase the non-volatile memory of the current device, then reprogram the application” appears, click “OK.”

Explorem Project Window of Quick Start Code

The figures below illustrate the “Processor Expert” and “Files” tabs.

Processor Expert Files

There are four tabs in the project window:
- Processor Expert
- Files

2. Choose the “Start Using CodeWarrior” button.
3. Using CodeWarrior, click File > Open and open the PE_LL64 Quick_Start.mcp file from the directory on your c:\ where you unzipped the compressed projects. This is the Quick Start project which uses CodeWarrior’s Processor Expert for device initialization.

Move the cursor to the line of code that begins with “//” (and remove the comment “//” from the existing line that begins with “#”)

1. Open P&E Multilink Toolkit Launch Pad and Terminal Window. From Windows start menu, select “Programs > P&E Embedded Multilink Toolkit > Toolkit Launchpad”.
2. Start a terminal console and configure your computer’s com port for 19200 baud, 8 bit, 1 stop, no parity.
3. Using CodeWarrior, compile and program the MC9S08LL64 microcontroller with the application by clicking on “Debug” button or hitting F5 on your keyboard, launching the programmer and the open source BDM debugger.
4. If the message “Load Executable File” appears, click “Yes.”
5. If the messages “Set Connection” and “Set Derivative” appear, set HCS08 – FSL Open Source BDM and derivative to “MC9S08LL64” and click “OK.”
6. If the message “Setting HCS08 and FSL Open Source BDM and derivative to mass erase the non-volatile memory of the current device, then reprogram the application will stop execution of the current one” appears, click “OK.”
7. When the message, “Loading a new application will stop execution of the current one” appears, click “OK.”
8. When the message, “The debugger is going to mass erase the non-volatile memory of the current device, then reprogram the application” appears, click “OK.”

Explore Project Window

The figures below illustrate the “Processor Expert” and “Files” tabs.

Processor Expert Files

There are four tabs in the project window:
- Processor Expert
- Files

2. Choose the “Start Using CodeWarrior” button.
3. Using CodeWarrior, click File > Open and open the PE_LL64 Quick_Start.mcp file from the directory on your c:\ where you unzipped the compressed projects. This is the Quick Start project which uses CodeWarrior’s Processor Expert for device initialization.
4. If the message “Load Executable File” appears, click “Yes.”
5. If the messages “Set Connection” and “Set Derivative” appear, set HCS08 – FSL Open Source BDM and derivative to “MC9S08LL64” and click “OK.”
6. If the message “Setting HCS08 and FSL Open Source BDM and derivative to mass erase the non-volatile memory of the current device, then reprogram the application will stop execution of the current one” appears, click “OK.”
7. When the message, “Loading a new application will stop execution of the current one” appears, click “OK.”
8. When the message, “The debugger is going to mass erase the non-volatile memory of the current device, then reprogram the application” appears, click “OK.”

Explore Project Window of Quick Start Code

The figures below illustrate the “Processor Expert” and “Files” tabs.

Processor Expert Files

There are four tabs in the project window:
- Processor Expert
- Files

2. Choose the “Start Using CodeWarrior” button.
3. Using CodeWarrior, click File > Open and open the PE_LL64 Quick_Start.mcp file from the directory on your c:\ where you unzipped the compressed projects. This is the Quick Start project which uses CodeWarrior’s Processor Expert for device initialization.
4. If the message “Load Executable File” appears, click “Yes.”
5. If the messages “Set Connection” and “Set Derivative” appear, set HCS08 – FSL Open Source BDM and derivative to “MC9S08LL64” and click “OK.”
6. If the message “Setting HCS08 and FSL Open Source BDM and derivative to mass erase the non-volatile memory of the current device, then reprogram the application will stop execution of the current one” appears, click “OK.”
7. When the message, “Loading a new application will stop execution of the current one” appears, click “OK.”
8. When the message, “The debugger is going to mass erase the non-volatile memory of the current device, then reprogram the application” appears, click “OK.”

Clock Display State

Upon power-on or reset, the LCD will flash all segments on, then off, and will display “6L64” and “CL.” Next, it enters a low-power time/day display mode. The display is initialized with a clock value and the day of the week. The MCU will keep track of time with the TCD module running off the 32.768 watch crystal.

Highlighted features include the TOD and LCD module operations in low-power stop mode. While in stop mode, the LCD blinking function is operational.

Measure current: To measure the average current of the MCU, remove the jumper “MCU IDD” J2 and place a current meter between the two pins of J2. Reset the TWR_S08LL64 module with the Reset button. You should measure less than 3 micro-amps.

Wake up time: The MCU is currently waking up every second and updating the seconds counter in the software. To reduce power consumption, the MCU wakeup time could be changed to 60 seconds. This reduces the times the MCU has to wake up, enter run mode and update the time on the LCD display.

In the file LL64_Demo.c the code to enable 60 second wake up is commented out. In the function “void StopClock(void)”, comment out the existing line that begins with “vnTOD_Init” and remove the comment “//” from the next line. When programmed with this new code, the MCU will wake up every 60 seconds. This changes from a second to the match interrupt. Measure the current again and compare.
Potentiometer vs. Light Sensor State

This state uses two ADC channels to read the position of the potentiometer and the light sensed by R21, the on-module light sensor. Press switch SW2 and the program enters Potentiometer vs. Light Sensor state. The display shows “PL” and the potentiometer value on the large characters on the bottom and the light sensor value on the smaller top right two characters. The third character displays the < > or = character comparing the POT to the light sensor. The two values are also sent through the SCI port at 19.2 K baud.

1. Launch the “Serial Grapher” utility from the PEMICRO UTILITY LAUNCH PAD program.
2. Select your computer’s COM port and set the baud rate to 19200, 8 bits, 1 stop bit, no parity. (Typically COM1)
3. Click “Open Serial Port and Start Demo.” The following should appear in the terminal window:

   “POT = FD Light Sensor Z1 = C4 ”

4. Press SW1 switch returns with a more filtered average of the raw accelerometer data. Pressing SW2 switch outputs a filtered version. Pressing the SW3 switch will return to streaming raw data from the accelerometer.

ADC Demo State

Press switch SW2 again and enter the ADC demo state. The LCD displays “ADC” and the 12-bit value in the first three characters as well as the channel number in the two upper right characters. Pushing SW4 will increment the channel converted and displayed.

This mode measures and displays the selected ADC channel while sending the data to the SCI output. Using the same terminal setup as State 2 you will be able to see the following data being displayed in the terminal window.

<table>
<thead>
<tr>
<th>Channel</th>
<th>Input</th>
<th>Pin Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AD0</td>
<td>ADP0</td>
</tr>
<tr>
<td>1</td>
<td>AD1</td>
<td>ADP1</td>
</tr>
<tr>
<td>10</td>
<td>AD10</td>
<td>ADP10</td>
</tr>
<tr>
<td>11</td>
<td>AD11</td>
<td>ADP11</td>
</tr>
<tr>
<td>12</td>
<td>AD12</td>
<td>ADP12</td>
</tr>
<tr>
<td>13</td>
<td>AD13</td>
<td>ADP13</td>
</tr>
<tr>
<td>14</td>
<td>AD14</td>
<td>ADP14</td>
</tr>
<tr>
<td>15</td>
<td>AD15</td>
<td>ADP15</td>
</tr>
</tbody>
</table>

Press SW1 switch returns with a more filtered average of the raw accelerometer data. Pressing SW2 switch outputs a filtered version. Pressing the SW3 switch will return to streaming raw data from the accelerometer.

Accelerometer Graphing State

Press switch SW2 and the program enters the Accelerometer Graphing state. This mode uses the 3-axis accelerometer, the ADC and the SCI to measure and output the 3-axis data. Using the same setup as State 2, you will be able to graph the movement of the accelerometer as you move the module. Lab 2 provides a more in-depth accelerometer demo. To get to the display below you must zoom in and then reengage by clicking the Play button.

Below is a table of the ADC channel assignments.

<table>
<thead>
<tr>
<th>Channel (Hex)</th>
<th>Function 1</th>
<th>Function 2</th>
<th>Function 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Connection</td>
<td>JP10</td>
<td>J1-A20</td>
</tr>
<tr>
<td>4</td>
<td>Potentiometer</td>
<td>Zero-G Accelerometer</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>X – Axs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Y – Axs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Z – Axs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SW3</td>
<td>JJ-B27</td>
<td>JJ-B28</td>
</tr>
<tr>
<td>9</td>
<td>No Connection</td>
<td>JJ-B29</td>
<td>JJ-B30</td>
</tr>
<tr>
<td>0A</td>
<td>RZ1 Light sensor</td>
<td>SW1</td>
<td>J1-A27</td>
</tr>
<tr>
<td>0b</td>
<td>SW2</td>
<td>JP10</td>
<td>J1-A28</td>
</tr>
<tr>
<td>13</td>
<td>VREFD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>VLCD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>VLL1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A</td>
<td>Temp Sensor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1B</td>
<td>Bandgap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1D</td>
<td>VREFO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1E</td>
<td>VREFL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Run Demo and Observe Graph

1. Remove the RZ1 jumper from JP6 so that the light sensor does not interfere with the operation of SW1. The rest of the jumpers should be in their default position.
2. Using CodeWarrior, click File→Open and open the TWR9S08LL64_Accelerometer project file from the directory on your c:\ where you unzipped the compressed projects.
3. Compile the code and program the MCU by clicking on “Debug” button, launching the debugger.
4. When the message, “Loading a new application will stop execution of the current one” appears, click “OK.”
5. When the message, “The debugger is going to mass erase the non-volatile memory of the current device, then reprogram the application” appears, click “OK.”
6. A debugger environment will open. From the main menu, choose “Run > Start/Continue.” The program will be executed in real-time.
7. Launch accelerometer utility from the PEMICRO TOOLKIT LAUNCH PAD and select “Accelerometer.”
8. Set your computer’s Com port, set the baud rate to 19200 and click “Open Serial Port and Start Demo.”