

# TWR-K60N512 Quick Start Demo

---

Lab Guide for MQX 3.7

Rev. 5

# Contents

<b>1 Purpose .....</b>	<b>3</b>
<b>2 Configure Hardware .....</b>	<b>3</b>
<b>3 Run the Demo.....</b>	<b>3</b>
<b>4 Development Software and Programming the Board .....</b>	<b>6</b>
4.1 CodeWarrior for Microcontrollers 10.1 .....	6
4.2 IAR Embedded Workbench for ARM 6.10 .....	13
<b>5 Task Aware Debugging.....</b>	<b>15</b>
<b>6 Demo Software Overview .....</b>	<b>15</b>
<b>7 Creating a new MQX project .....</b>	<b>16</b>
<b>8 OSJTAG.....</b>	<b>18</b>

## 1 Purpose

---

This lab document describes how to use the TWR-K60N512 demo that is pre-programmed onto your Tower module. It also covers how to configure your tower module to run the demo, and the steps to re-flash this project back onto your board.

Use Sections 2 and 3 to run the demo. Use Section 4 to learn how to re-flash the demo on the board.

## 2 Configure Hardware

---

If running the TWR-K60N512 board in stand-alone mode:

- 1) Plug-in the battery into the battery socket on the underside of the board
- 2) Change **J9** to pins 2-3 to connect the VBAT domain to run off the battery

If using the TWR-K60N512 board with the TWR-SER board (required for web server and USB demos):

- 1) Follow the steps in the section above plus:
- 2) On the TWR-K60N512 board:
  - a. Change **J6** to pins 2-3 to use the 50Mhz clock from the TWR-SER board
- 3) On the TWR-SER board:
  - a. Change **J2** to pins 3-4 to give a 50Mhz clock to the PHY
  - b. Add jumper to **J3** pins 2-3 to bring up the 50Mhz clock from the TWR-SER
  - c. Add jumper to **J12** pins 9-10 to put into RMII mode

If you do not have extra jumpers, you can use the ones on CAN\_SEL (**J5**) on the TWR-SER board

- 4) Connect the TWR-K60N512 and TWR-SER boards via the elevator boards. Make sure the primary side (often marked with a white stripe) goes into the elevator board with the white edges. Detailed instructions are in the Quick Start Guide inside the TWR-ELEV module.
- 5) Connect an Ethernet cable between the TWR-SER board and an Ethernet port on your computer
- 6) Connect a mini-B USB cable between the TWR-SER board and a USB port on your computer. This is only required for running the USB demo.
- 7) Note that the default jumper settings are in the Quick Start Guide for the TWR-K60N512 and TWR-SER if you want to change back to the default settings after you are done running the demo.

## 3 Run the Demo

---

- 1) Install the latest P&E Micro Kinetis Tower Toolkit under the Downloads tab on the [TWR-K60N512 website](#) if you have not done so already.
- 2) Connect a mini-B USB cable to the TWR-K60N512 tower module.

- 3) Allow the PC to automatically configure the OSJTAG drivers used for debugging and the serial-to-USB feature. If you have trouble connecting to the board or using the Serial Terminal, see [Section 8](#) for troubleshooting.

### Touch Demo

- 4) Press the E1-E4 touch pads to toggle on and off the LED's

### Memory Game

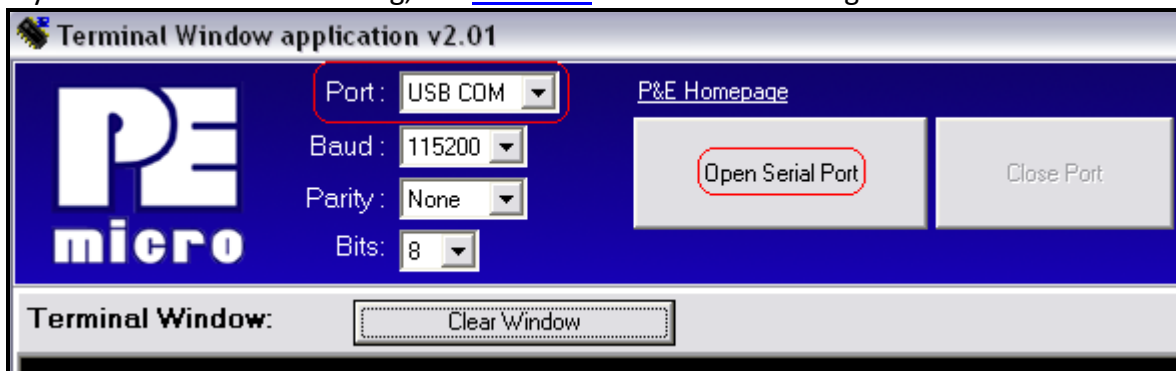
- 5) Press SW2 to play a memory game using the Touch pads E1-E4. A sequence will light up, and then press the touch sensors in the order flashed. If an incorrect sequence is input or you take too long, then all the lights will blink rapidly and the game will reset. See how far you can go!
- 6) Press SW1 to go back to the touch and accelerometer demo

### Accelerometer Demo

- 7) Tilt the board back and forth to see the LED's light up as it is tilted
- 8) Press SW1 and SW2 at the same time to calibrate the accelerometer. It will store the calibration into internal flash memory, so this will only need to be done one time while the board is flat.

### Terminal Demo

- 9) Open the P&E Terminal Utility by clicking on Start Menu->Programs->P&E Kinetis Tower Toolkit->Utilities->Terminal Utility
- 10) Make sure USB COM is selected with 115200 baud, and click on the "Open Serial Port" button. If you have trouble connecting, see [Section 8](#) for Troubleshooting.



- 11) In the terminal program, there will be a shell prompt after you hit the Enter key. Type **"help"** to see the full list of commands.
- 12) Type **"settime HH:MM"** to adjust the time to HH:MM.
- 13) Type **"accel start"** to print out the accelerometer data to the terminal. Type **"accel stop"** or press SW1 or SW2 to stop the printout.

### SD Card Demo

- 14) On **Rev D** TWR-K60N512 boards, there is an erratum that sends a reset to the Ethernet PHY on the TWR-SER when inserting an SD Card. This will cause a loss of communication with the web server. To get around this erratum, you need to switch on Write Protect on the SD card, and insert it before the board boots up. You can hit the Reset button after inserting the card to do this. Rev C boards do not have this issue.

Also note that in order to provide the same write functionality between RevC and RevD boards, the Write Protect position on the SD Card is ignored for this demo.

- 15) Insert a SD Card into the SD Socket on the underside of the board. You can then access the files and create new files and directories on the SD Card via the shell interface. Type **“help”** to see the full list of commands.

**USB Mouse Demo - The following steps require the TWR-SER peripheral module:**

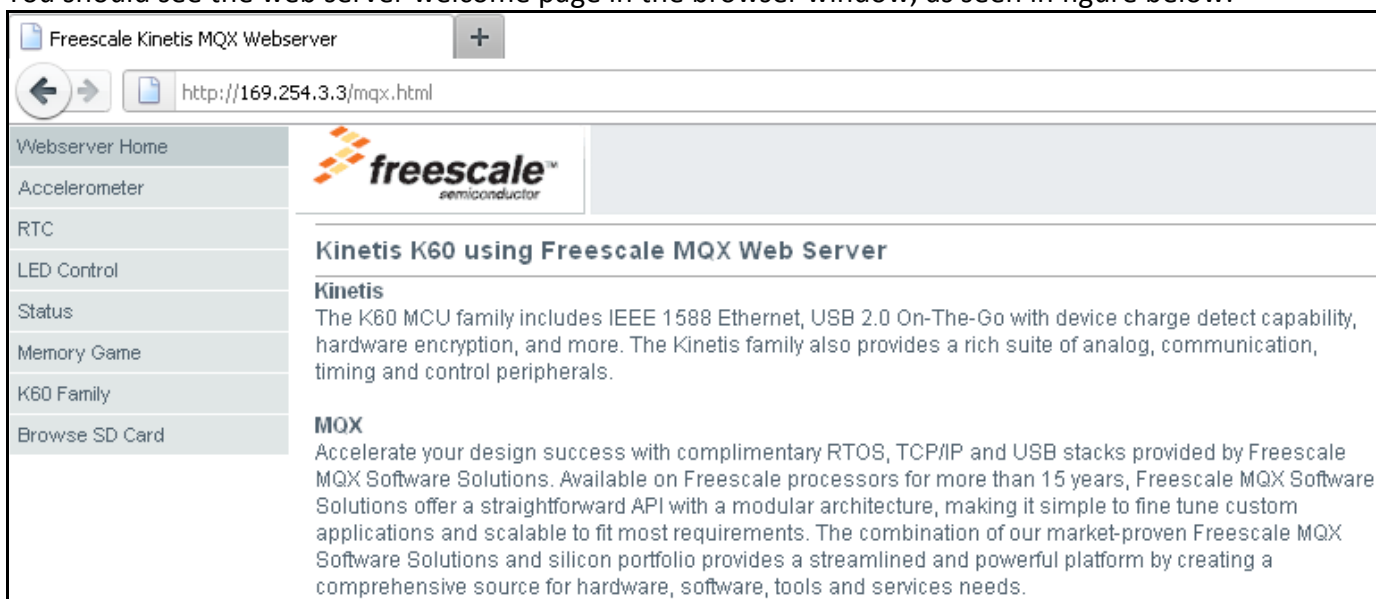
- 16) Plug in a USB mini-B cable into the USB mini-B connector on the TWR-SER.
- 17) The tower kit will then enumerate as a mouse on your computer. You can tilt the board around to move the mouse cursor. Press SW2 to left click. Press SW1 to right click.

**Web Server Demo - The following steps require the TWR-SER peripheral module:**

- 18) Finally we will run the web browser. The default IP address of the board is 169.254.3.3. Typically, when you connect your computer directly to the board, the computer will default to an auto IP address on the same subnet as the board (169.254.x.x), therefore requiring no setup.

Note: The PC may take a few minutes to default to the auto IP address and make the connection. Also ensure any browser proxy settings, VPN connections, and wireless connections are turned off since that can interfere with making a connection to the IP address.

- 19) However, if you have trouble connecting, you may configure the IP address of the computer manually. Select Start >Settings > Network Connections > Local Area Connection. Note your original TCP/IP settings, and then set your IP address to **169.254.3.4** and your subnet mask to **255.255.0.0**.
- 20) Open a web browser and go to the target device address. In this case, **169.254.3.3**
- 21) You should see the web server welcome page in the browser window, as seen in figure below:



- 22) Browse the links on the left hand side to see other pages. These pages include:
- Accelerometer data
  - Real Time Clock (displaying elapsed time since first boot-up, or the time set by the **settime** command)
  - Toggle LED's
  - Status of the switches, LED's, potentiometer, and ADC temperature sensor
  - The current score and high score in the memory game (not part of pre-flashed demo. This page is only in web-downloaded version)
  - More information on the Kinetis K60 family
- 23) Browse web pages on the SD card you inserted earlier by clicking on the Browse SD Card link on the left hand menu. It will link to an index.html file you can place on the SD Card. You can also access any other webpage on the SD card by going to **<board IP address>\sdcard\file\_name.html**
- (This feature is not part of the pre-flashed demo, and is only in web-downloaded version. Also note the errata described in the SD Card demo section above for **Rev D** TWR-K60N512 boards.)

## 4 Development Software and Programming the Board

---

The following instructions describe how to build and flash the Kinetis Quick Start Demo using MQX 3.7 with both CodeWarrior MCU 10.1 and IAR 6.10.

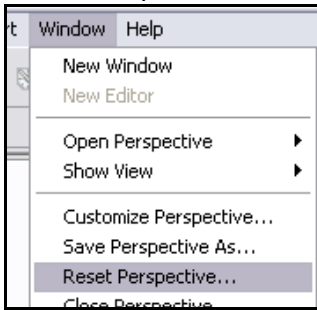
Note that for Windows Vista and Windows 7 users, it is recommended to install both the compiler IDE and MQX outside default **C:\Program Files** directory.

If using MQX 3.6.2, please see the QSD Lab for that release.

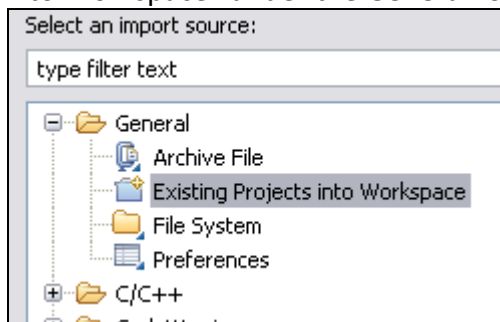
### 4.1 CodeWarrior for Microcontrollers 10.1

- 1) Install the latest P&E Micro Kinetis Tower Toolkit off the [TWR-K60N512 website](#) to install the newest OSJTAG and USB-to-Serial drivers. See [Section 8](#) for OSJTAG Troubleshooting.
- 2) Install CodeWarrior for Microcontrollers 10.1. The Evaluation edition or Standard version or higher will be required because of code size.
- 3) Then you will need to download the CW 10.1 patches. These patches can be found on the [CodeWarrior 10](#) website. You will need to download:
  - a. [MCU v10.1 Compiler Update](#)
  - b. [MCU v10.1 Kernel Awareness Update](#)
  - c. [MCU 10.1 Update](#)
- 4) Leave the patches zipped up, and follow the patch installation instructions which can be found [here](#). Follow the procedure laid out in Section C, "Offline Mode"
- 5) Install MQX 3.7. You can find this from the [MQX website](#)
- 6) You will need to enable the ESHDC and I2C0 drivers in the BSP library to use this demo.

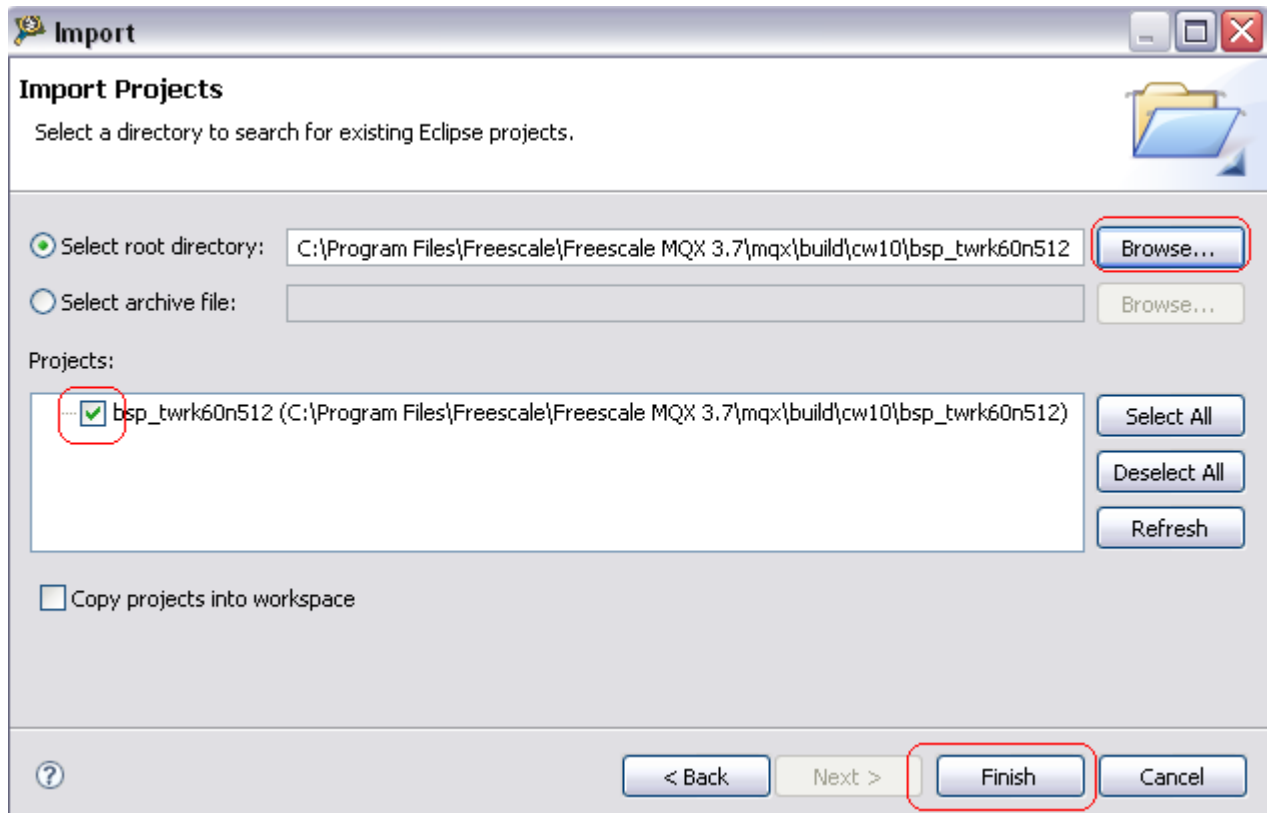
- 7) Open CW10. At the welcome screen, set the workspace to the MQX installation directory, which will typically be: **C:\Program Files\Freescale\Freescale MQX 3.7**
- 8) If you already have CW10 open, you can change the workspace by going to **File->Switch Workspace**
- 9) The first time you open CW10, you will be taken to the Welcome screen. Click on “Go To Workbench” in the lower left hand side.
- 10) The workbench view will open up. To ensure all the windows are properly set, go to Window->Reset Perspective



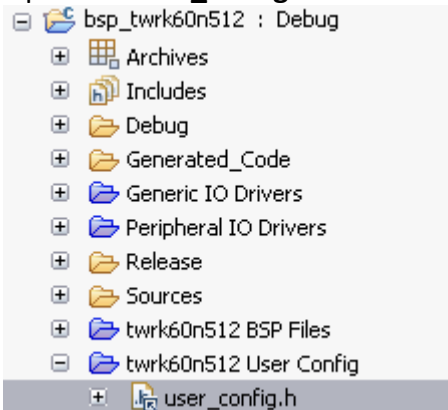
- 11) Click on **File->Import** in the menu bar. In the dialog box that comes up, select “Existing Projects into Workspace” under the General folder. Then click Next



- 12) On the next screen, select the “Select root directory:” option, and click on Browse
- 13) Navigate to the **C:\Program Files\Freescale\Freescale MQX 3.7\mqx\build\cw10\bsp\_twrk60n512** directory and hit OK.
- 14) Make sure **bsp\_twrk60n512** is checked, and hit Finish



15) Open the **user\_config.h** file in the **bsp\_twrk60n512** project



16) Change

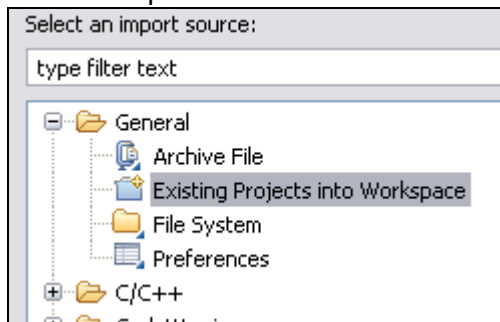
```
#define BSPCFG_ENABLE_I2C0    0
#define BSPCFG_ENABLE_ESDHC  0
to
#define BSPCFG_ENABLE_I2C0    1
#define BSPCFG_ENABLE_ESDHC  1
```

17) Save the **user\_config.h** file

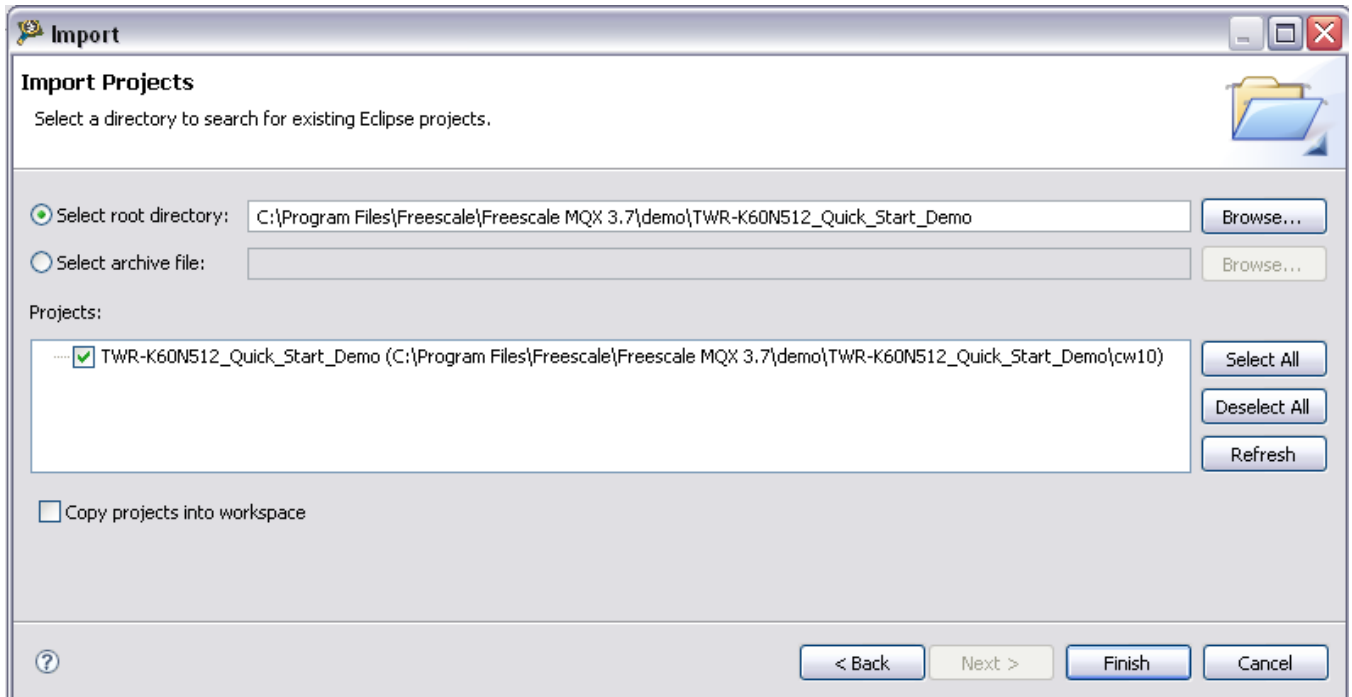
18) Click on the Make button  to re-compile the K60 BSP library.



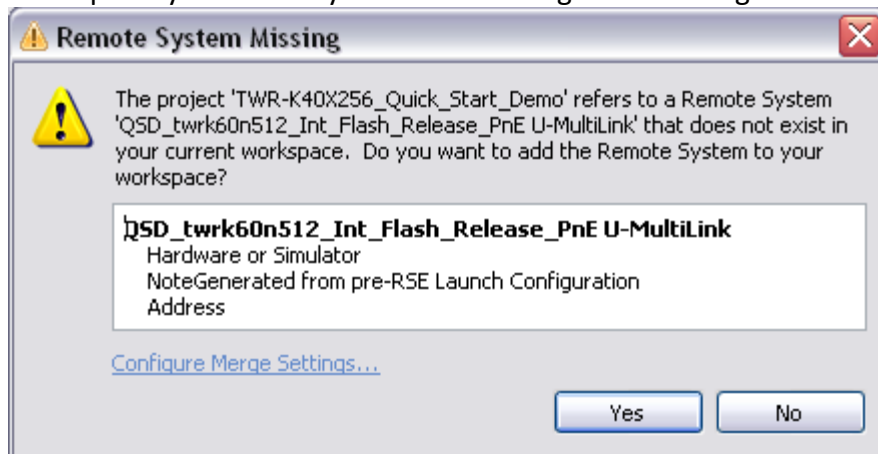
- 19) If you did not install the MQX project in the default **C:\Program Files\Freescale\Freescale MQX 3.7\** directory during installation, you must first recompile the rest of the MQX libraries. See the MQX release notes for more information before continuing on with the lab.
- 20) Download the Kinetis TWR-K60N512 Quick Start Demo Lab from the [TWR-K60N512 website](#). This document is part of that package, so you likely will have already done this.
- 21) Unzip the **TWRK60N512QSDLAB.zip** file you just download, and run the executable located inside. It will install the lab project and this lab guide into the **C:\Program Files\Freescale\Freescale MQX 3.7\demo** directory by default. If MQX is installed in another location, point the installer to the **\demo** directory at that location instead.
- 22) Next we need to import the Quick Start Demo into our CW10 workspace.
- 23) Click on **File->Import** in the menu bar. In the dialog box that comes up, select “Existing Projects into Workspace” under the General folder. Then click Next



- 24) On the next screen, select the “Select root directory:” option, and click on Browse
- 25) Navigate to the **C:\Program Files\Freescale\Freescale MQX 3.7\demo\ TWR-K60N512\_Quick\_Start\_Demo** directory and hit OK.
- 26) Make sure **TWR-K60N512\_Quick\_Start\_Demo** is checked, and hit **Finish**. **Please read the instructions below for the error dialog box that will likely come up after hitting Finish.**



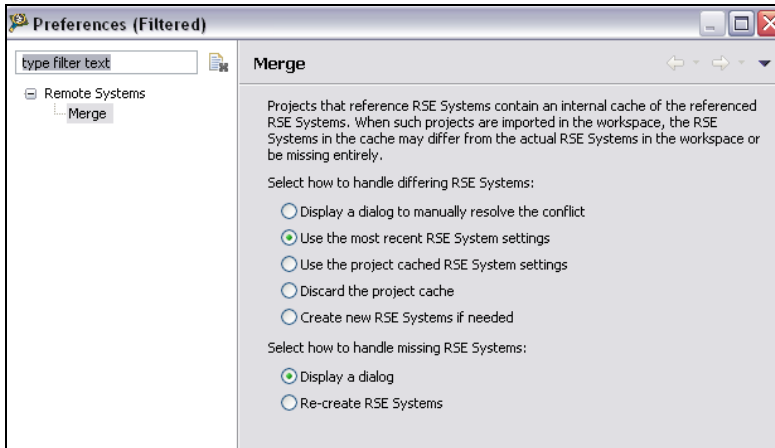
27) At this point you will likely see the following error message:



28) When importing a project, you will often see a “Remote Systems Changed” or “Remote System Missing” dialog box. This occurs when CodeWarrior detects a mismatch between the RSE (Remote System Explorer) settings in the project and the settings in the framework, or if the RSE system is missing in the project.

29) Click on “Configure Merge Settings” to avoid having the dialog box re-come up for this project.

30) On the screen that comes up, select the “Use the most recent RSE System settings” option, and click “Apply” and then “OK”

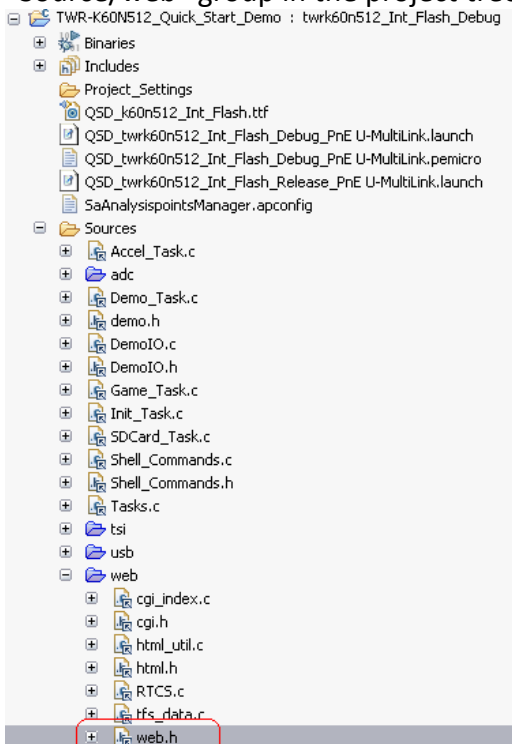


- 31) Then hit “Yes” to get rid of the error message. It is very important to not hit “No”
- 32) If you did hit “No”, you will need reset the RSE project settings. First close the project by right-clicking on it and selecting “Close Project”, and then open it up by right clicking on the project and selecting “Open Project”. The dialog box should come up again, and this time follow the steps listed above to solve this issue.
- 33) Repeat these steps for any other RSE message boxes that come up

- 34) Make sure the Debug configuration is set for the project



- 35) Open the web.h file in the project pane window. Double-click the file item located in the “Source/web” group in the project tree.




36) You can change the default IP address by modifying the ENET\_IPADDR define. For example, to set the target address to 169.254.3.3, and the line will be:

```
#define ENET_IPADDR IPADDR(169,254,3,3)
```

37) You could also use DHCP instead of a static IP address by setting

```
DEMOCFG_ENABLE_DHCP to 1 in web.h
```

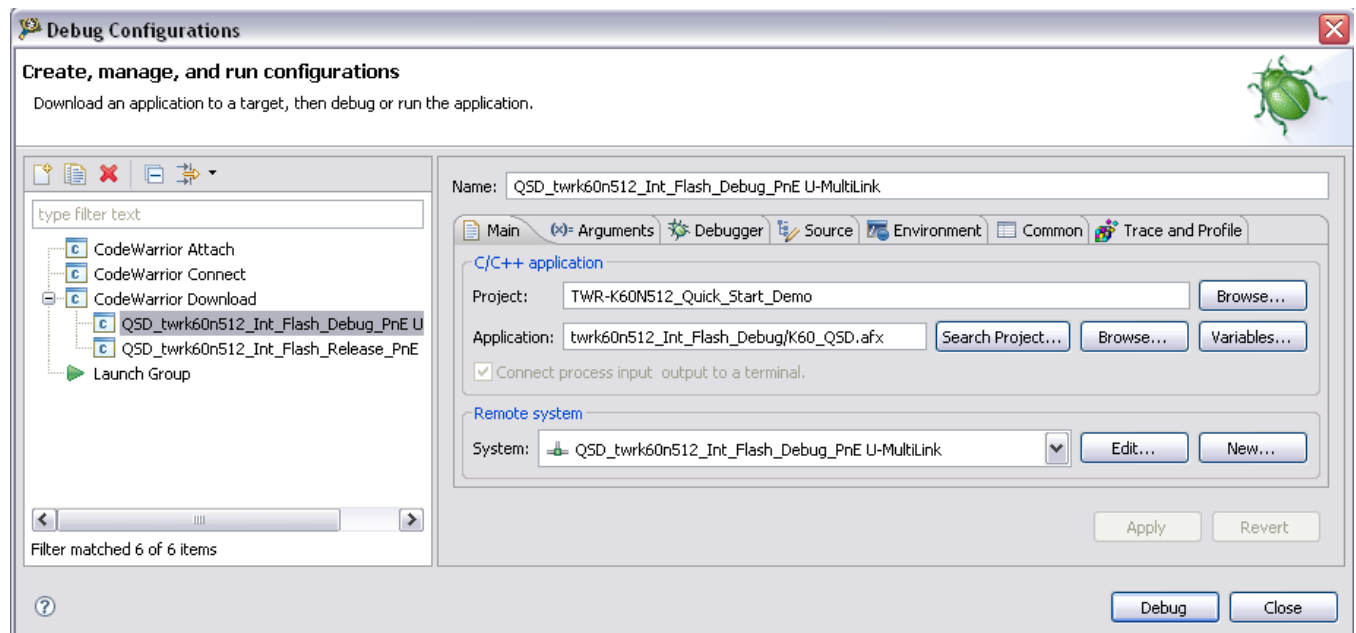
38) Save the file after you are done.

39) Build the project by clicking on the Hammer icon in the toolbar 

40) Double click on the **QSD\_k60n512\_Int\_Flash.ttf** file. In the dialog box that comes up, click "OK". The file will then open. You can then close it. You only need to do this step once after initially opening the project. The dialog box may also not appear, so do not worry if that is the case.

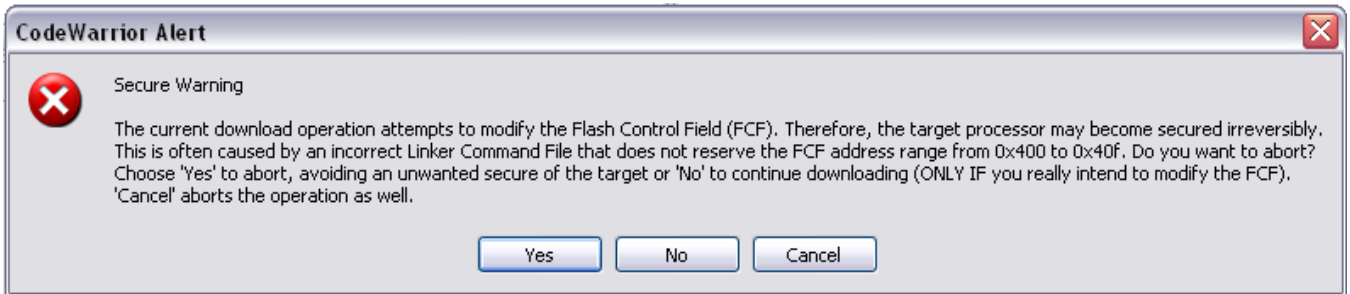
41) Click **Run->Debug Configurations...** in the menu bar, and select the

**QSD\_twrk60n512\_Int\_Flash\_Debug\_PnE U-MultiLink** configuration. Then hit the debug button in the bottom of the window




42) If this is the first time you've used CW10.1 with your board, you may get a dialog box asking to update the OSJTAG firmware. Unplug your board, put a jumper on **J10** which is labeled **JM60 Boot**, and plug the board back in. Hit "OK" on the dialog box, and the board OSJTAG firmware will update. When it is done, unplug the board, remove the jumper on **J10**, and hit "OK" again on the dialog box. If you do not have a spare jumper, you can temporarily use the one on J9.

43) The code will then be flashed to the board and the debugger started. This may take a several moments. You may also see the following Warning message:



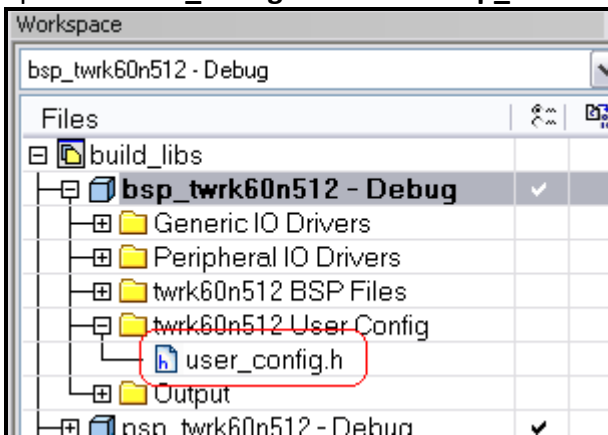
This is a check introduced with a CodeWarrior compiler patch. Hit **No** to continue flashing the part, as the BSP is configured to ensure the part remains unsecured.

44) Once the code is done flashing, the code will pause at the start of the MQX main() function.

45) Hit the run icon to continue the program execution. 

## 4.2 IAR Embedded Workbench for ARM 6.10

- 1) Install the latest P&E Micro Kinetis Tower Toolkit off the [TWR-K60N512 website](#) to install the newest OSJTAG and USB-to-Serial drivers. See [Section 8](#) for OSJTAG Troubleshooting.
- 2) Install IAR for ARM v6.10. The Evaluation edition or Full edition will be required because of code size.
- 3) Install MQX 3.7. You can find this on the [MQX website](#)
- 4) You will need to enable the ESHDC driver in the BSP library to use this demo.
- 5) Open the workspace at **C:\Program Files\Freescale\Freescale MQX 3.7\config\twrk60n512\iar\build\_libs.eww**
- 6) Open the **user\_config.h** file in the **bsp\_twrk60n512 – Debug** project




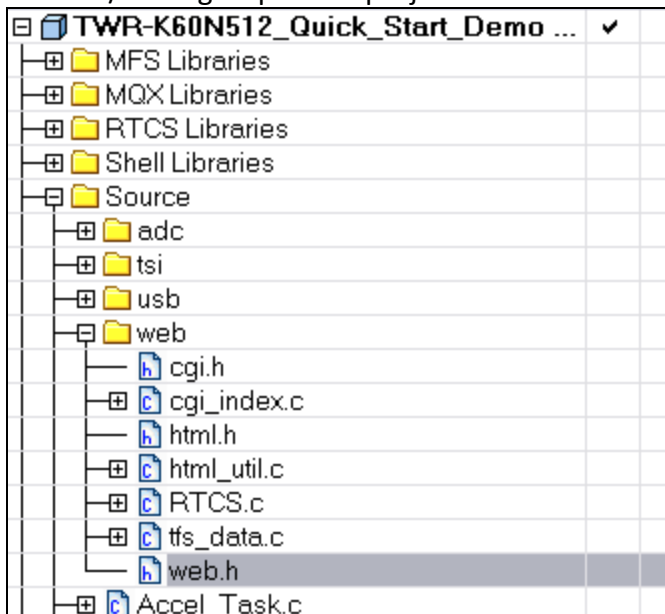
- 7) Change
 

```
#define BSPCFG_ENABLE_I2C0    0
#define BSPCFG_ENABLE_ESDHC  0
```

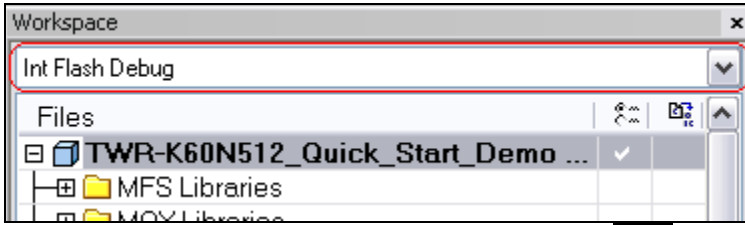
 to
 


```
#define BSPCFG_ENABLE_I2C0    1
#define BSPCFG_ENABLE_ESDHC  1
```
- 8) Save the **user\_config.h** file


- 9) Click on the Make button  to re-compile the K60 BSP library. Make sure **bsp\_twrk60n512 – Debug** project is bolded
- 10) If you did not install the MQX project in the default **C:\Program Files\Freescale\Freescale MQX 3.7\** directory during installation, you must first recompile the rest of the MQX libraries. See the MQX release notes for more information before continuing on with the lab.
- 11) Download the Kinetis TWR-K60N512 Quick Start Demo Lab from the [TWR-K60N512 website](#). This document is part of that package, so you likely will have already done this.
- 12) Unzip the **TWRK60N512QSDLAB.zip** file you just download, and run the executable located inside. It will install the lab project and this lab guide into the **C:\Program Files\Freescale\Freescale MQX 3.7\demo** directory by default. If MQX is installed in another location, point the installer to the **\demo** directory at that location instead.
- 13) Open the workspace file that will be located at **C:\Program Files\Freescale\Freescale MQX 3.7\demo\TWR-K60N512\_Quick\_Start\_Demo\iar\TWR-K60N512\_Quick\_Start\_Demo.eww**
- 14) Open the web.h file in the project pane window. Double-click the file item located in the “Source/web” group in the project tree.



- 15) You can change the default IP address by modifying the ENET\_IPADDR define. For example, to set the target address to 169.254.3.3, and the line will be:  
**#define ENET\_IPADDR IPADDR(169,254,3,3)**
- 16) You could also use DHCP instead of a static IP address by setting **DEMOCFG\_ENABLE\_DHCP** to **1** in **web.h**
- 17) Save the file after you are done.
- 18) Select the Int Flash Debug target



19) Compile the project by clicking the Make icon  (or right clicking on the project and select “Make”).



20) After compilation completes, download the code to the board and start the debugger by pressing the “Download and Debug” button .

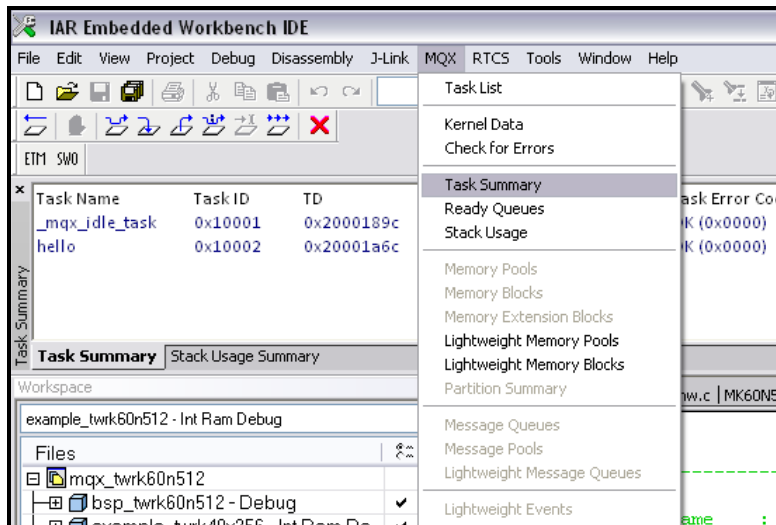
21) The code will download Flash, and the debugger screen will come up and pause at the first instruction. Hit the “Go” button to start running.



## 5 Task Aware Debugging

MQX comes with Task Aware Debugging (TAD) when using both CodeWarrior and IAR IDE’s. This feature is only available in the Evaluation or Full editions of those compilers.

To use TAD, start up an MQX project within the debugger. While the project is running, hit the **Break** button (  in IAR, or  in CodeWarrior) to pause the debugger. Then click on MQX->Task Summary to see all the tasks MQX is running and their current state. Explore the other options as well, and look at the Release Notes for more information on TAD.



## 6 Demo Software Overview

This demo is a combination of several tasks running in parallel, while using MQX drivers and stacks.

Looking at **Tasks.c** first, this contains the **MQX\_template\_list[]** data structure. This has the attributes for all the tasks that will be created by this demo, except for the webserver tasks. It includes the stack size for each tasks, their priority, and any other attributes. More details can be found in the MQX User Guide in the `<mqx_dir>\docs\mqx` directory.

The **Init\_Task** is the only task with the **MQX\_AUTO\_START\_TASK** attribute set, so it is the only task that will be running at bootup. This code for this task is in **Init\_Task.c**. This task will start all the other tasks for the demo.

Inside the initialization task, global variables are initialized and then the GPIO driver is initialized.

Next the **Accl\_Task** is started, which polls the MMA7660 sensor on the TWR-K60N512 over I2C. Then the **ADC\_Task** is started, which polls the ADC channel connected to the potentiometer, and the ADC temperature sensor. The values for these readings are stored in a global **SENSOR\_DATA** structure, so that their information can be easily re-used by other tasks.

Next the memory game task is started, which waits for the memory mode to start.

Next the **Demo\_Task** is started. This determines, based on the accelerometer readings and button presses, which mode the demo is in and how the LED's should be controlled. This is the main task for the Quick Start Demo.

The shell and SD Card tasks are started after that. And then the USB task is started which starts the USB stack and waits for a USB connection to enumerate as a HID class mouse.

The web server is then initialized, which creates two tasks for the RTCS stack and the http server. The web pages are stored in the `\web\web_pages` folder. The **mktfs.exe** executable converts them into ASCII arrays stored in the **tfs\_data.c** file, which is what is programmed into flash. The **cgi\_index.c** file contains the CGI functions for getting data to and from the web server.

Finally the TSI module is initialized and started. It performs some calibration and then uses the TSI ISR to detect touches. Depending on the demo mode, it will set different variables upon detecting a touch.

The initialization task then calls **\_task\_block()** to be permanently blocked, and the demo is now up and ready. See the comments in the source code for more information on how the demo works.

Also explore the documentation, app notes, and examples in the `<mqx_dir>\doc` folder and explore online at <http://freescale.com/mqx> for more information on using MQX.

---

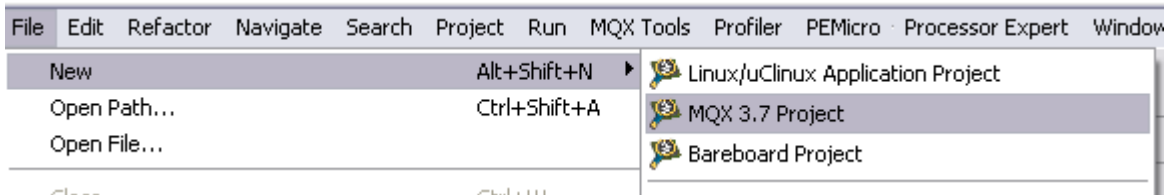
## 7 Creating a new MQX project

---

To create a new MQX project in CW10, use the following steps:

- 1) Go to File->New->MQX 3.7 Project





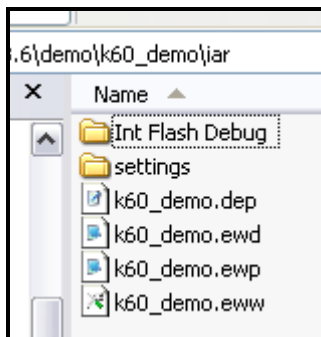
- 2) Follow the wizard to setup and create a new MQX project.

To create a bare-board MQX project in IAR, use the following steps:

- 1) Make sure you have not modified the files in the K60 Quick Start Demo
- 2) Double click on the **create\_new\_k60\_mqx\_project\_iar.exe** file inside the K60 Quick Start Demo.
- 3) Type in the new project's name, with no spaces, such as **k60\_hello**
- 4) Hit Enter and wait for it to finish creating the new project. The script will copy the project files from the K60 Quick Start Demo into a new directory in **<mqx\_dir>\demo**, modify them to use the new name, and copy the Hello World example from **<mqx\_dir>\mqx\examples\hello**
- 5) Hit Enter to close Window.

To manually copy the Quick Start Demo use the following steps:

1. Copy the **TWR-K60N512\_Quick\_Start\_Demo** folder and place the copy in the MQX demo folder at **C:\Program Files\Freescale\Freescale MQX 3.7\demo**
2. Rename the folder to the new project name, with no spaces. For example, **k60\_demo**
3. Inside the k60\_demo folder, go into the **iar** folder
4. Rename the four IAR files to k60\_demo



5. Open the k60\_demo.eww file and change the following line to point to the new project name:  
`<?xml version="1.0" encoding="iso-8859-1"??>`

```

<workspace>
  <project>
    <path>$WS_DIR$\k60_demo.ewp</path>
  </project>
  <batchBuild/>
</workspace>

```

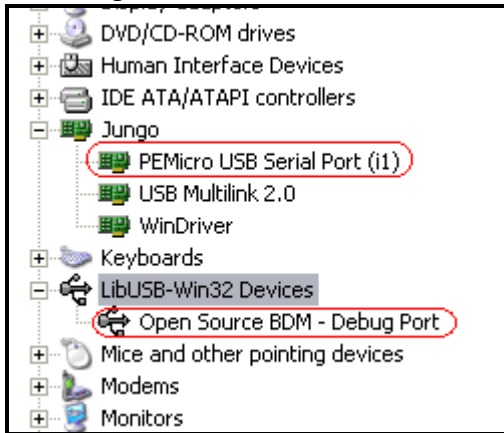
6. Save the file
7. Open the k60\_demo.eww workspace

## 8 OSJTAG

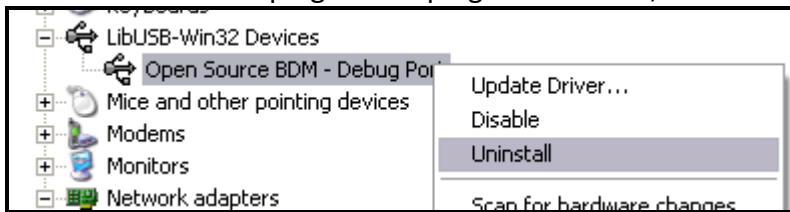
Open Source JTAG (also known as OSBDM on ColdFire tower boards) allows a user to program, debug, and get serial data from Kinetis devices via a USB cable. The firmware runs on a Freescale MCFS08JM60 on the underside of the Kinetis tower board.

The latest firmware and drivers can be found at <http://pemicro.com/osbdm>. See the **Installation and Operation** document for details on updating the firmware and drivers. If you are having trouble connecting, try updating to the latest drivers, virtual serial toolkit, and firmware located on that website.

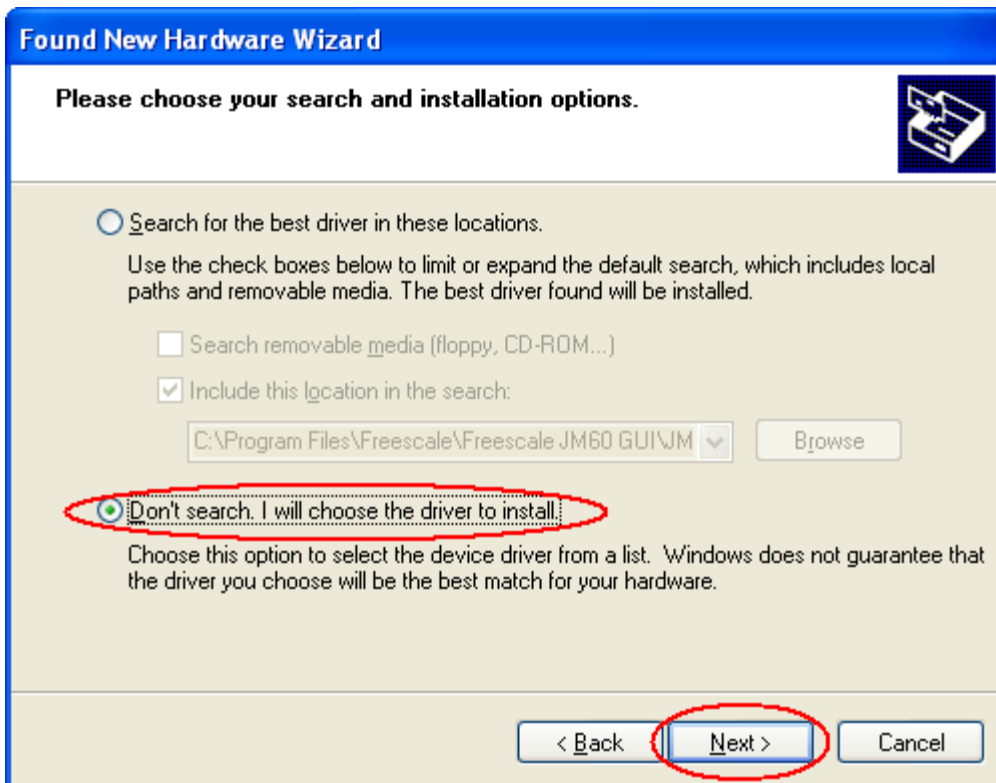
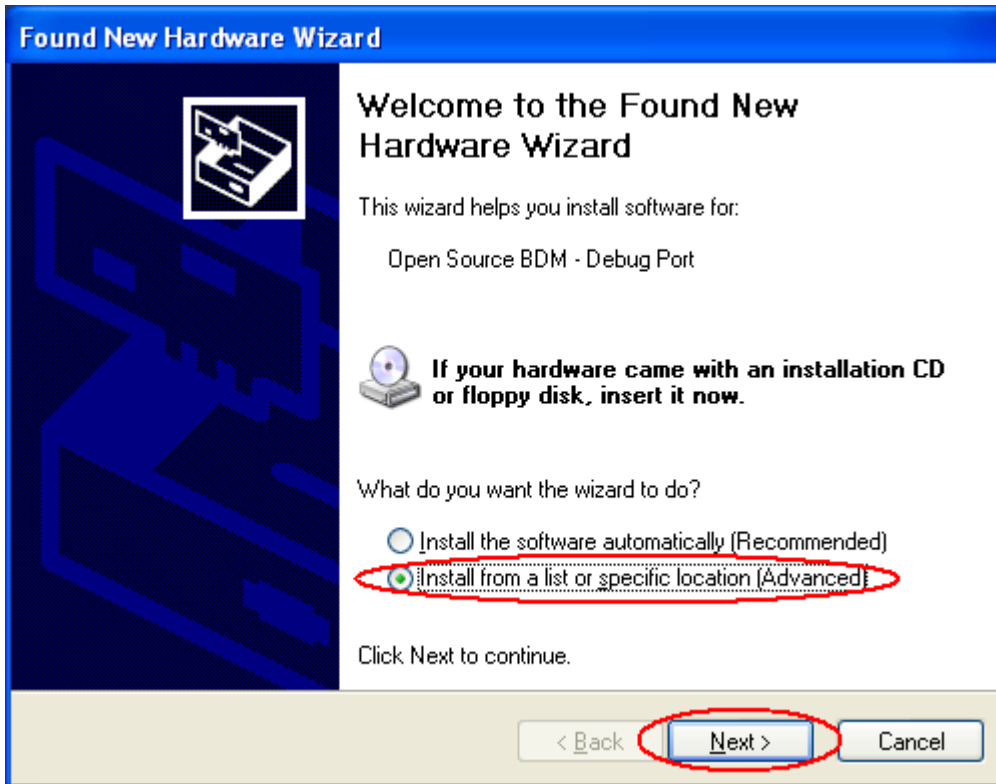
When the tower board is plugged in, it should enumerate as a composite device, with one driver for debugging, and the other as a serial port. If you go to the Device Manager you should see the following:

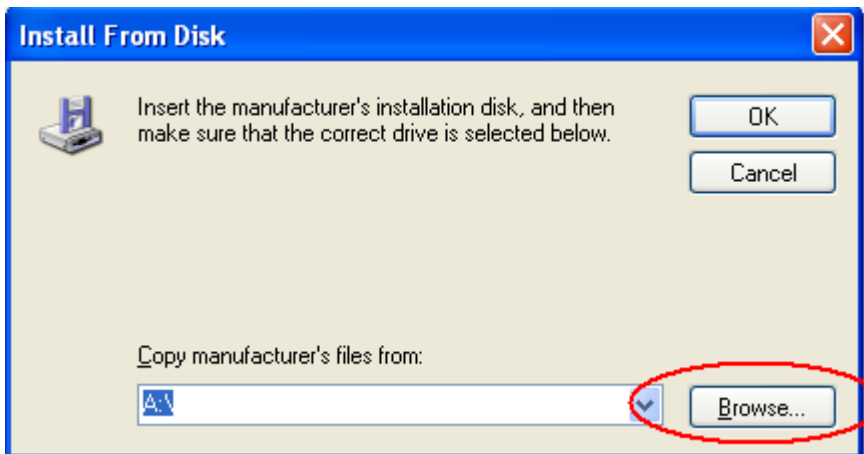
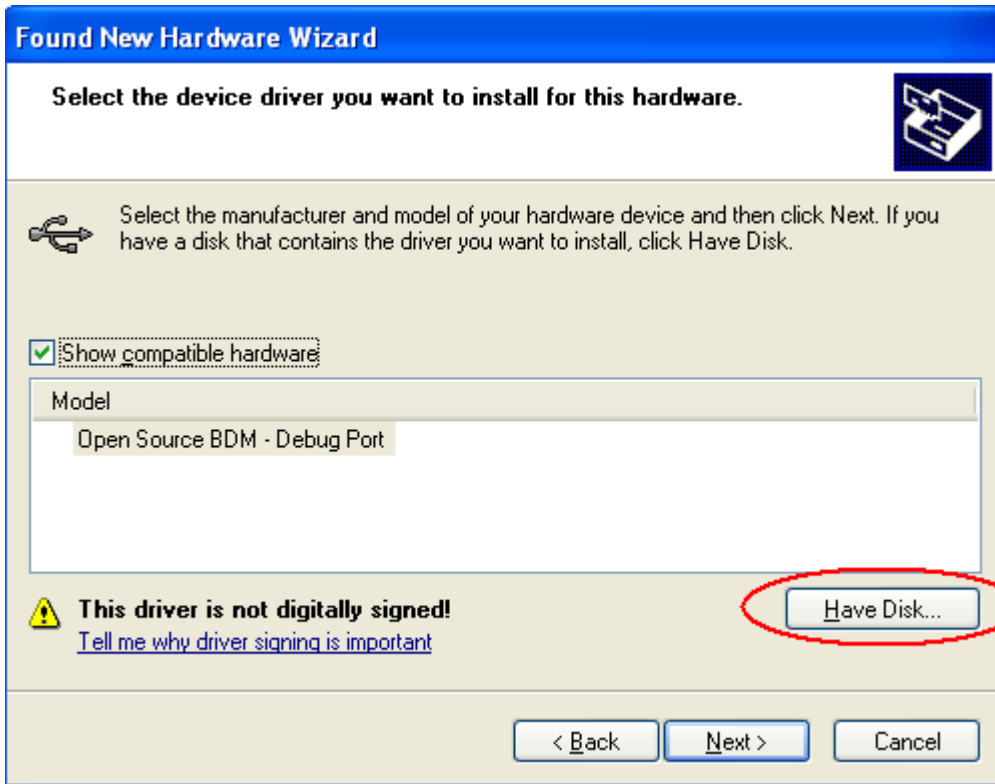


If you only see it enumerate as the Open Source BDM Debug Port, then your computer may automatically pick up an outdated driver. To fix this, right click on the OSBDM driver and select "Uninstall". Then unplug and re-plug in the board, and it should enumerate correctly.



If there are still problems with enumerating correctly, you can also manually select the drivers.





- For the Open Source BDM – Debug Port, use the driver at:  
**C:\pemicro\kinetis\_tower\_toolkit\Drivers\osbdm\OSJTAG\_Debug\_Interface\_libusb.inf**
- For the PEMicro USB Serial Port (i1), use the driver at:  
**C:\pemicro\kinetis\_tower\_toolkit\Drivers\osbdm\OSJTAG\_Serial\_Interface\_windriver\_version.inf**