



TWR-K53N512 Quick Start Demo Lab Guide

Board configuration, software, and development tools

Rev. 2

2011-08-26

Most up-to-dated lab guide and source code are available at

www.freescale.com/twr-k53n512

Freescale, the Freescale logo, CodeWarrior, ColdFire, C-Ware, the Energy Efficient Solutions logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. ColdFire+, CoreNet, Flexis, Kinetis, MXC, Platform in a Package, Processor Expert, and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2011 Freescale Semiconductor, Inc.

Contents

| | |
|---|-----------|
| 1 Purpose | 3 |
| 2 Configure Hardware | 3 |
| 3 Run the Demos..... | 3 |
| 4 Development Software and Programming the Board | 6 |
| 4.1 CodeWarrior Development Studio for Microcontrollers 10.1..... | 6 |
| 4.2 IAR Embedded Workbench for ARM 6.10.5 | 13 |
| 5 Task Aware Debugging..... | 15 |
| 6 Demo Software Overview | 15 |
| 7 Creating a new MQX project | 17 |
| 8 OSJTAG..... | 18 |
| 9 Update OSJTAG | 21 |
| 10Resouces | 22 |

1 Purpose

This lab document will familiarize you with the TWR-K53N512 kit and two development tools using Freescale RTOS MQX.

You will learn the full features of the **TWR-K53N512** demos that are pre-programmed onto your Tower module. You will also learn how to access the source code using development tool IAR 6.10.5 or the CodeWarrior 10.1 development tool.

2 Configure Hardware

If using the **TWR-K53N512 board in stand-alone mode**:

- 1) Plug-in the battery into the battery socket on the underside of the board
- 2) Change **J16** to pins **2-3** to connect the VBAT domain to run off the battery (Optional)
- 3) Change **J11** to pins **1-2** to use the 50Mhz clock from the **TWR-K53N512** board (Default)

If using the **TWR-K53N512 board with the TWR-SER board** (required for web server and USB demos):

- 1) On the **TWR-K53N512** board:
 - a. Change **J11** to pins **2-3** to use the 50Mhz clock from the **TWR-SER** board
- 2) 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
 - d. If you want to power from **TWR-SER** USB, add jumper to **J16 3-4**
- 3) Connect the **TWR-K53N512** 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 box.
- 4) Connect an Ethernet cable between the **TWR-SER** board and an Ethernet port on your computer
- 5) 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.
- 6) Note that the default jumper settings are in the Quick Start Guide for the **TWR-K53N512** and **TWR-SER** if you want to change back to the default settings after you are done running the demo.

NOTE: If you do not have extra jumpers, you can use the ones on CAN_SEL (**J5**) on the **TWR-SER** board

3 Run the Demos

- 1) Install **P&E OSBDM Virtual Serial Toolkit** from DVD or <http://www.pemicro.com/osbdm>
- 2) Connect the **TWR-K53N512** tower module to your PC using a mini-B USB cable.

- 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.

Segment LCD Demo

- 4) When **TWR-K53N512** first powers up, the **TWRPI-SLCD** display will toggle all LCD segments on and off.
- 5) Press **SW2** to cycle between displaying the time (seconds, hour and minutes), potentiometer position, and temperature.

Accelerometer Demo

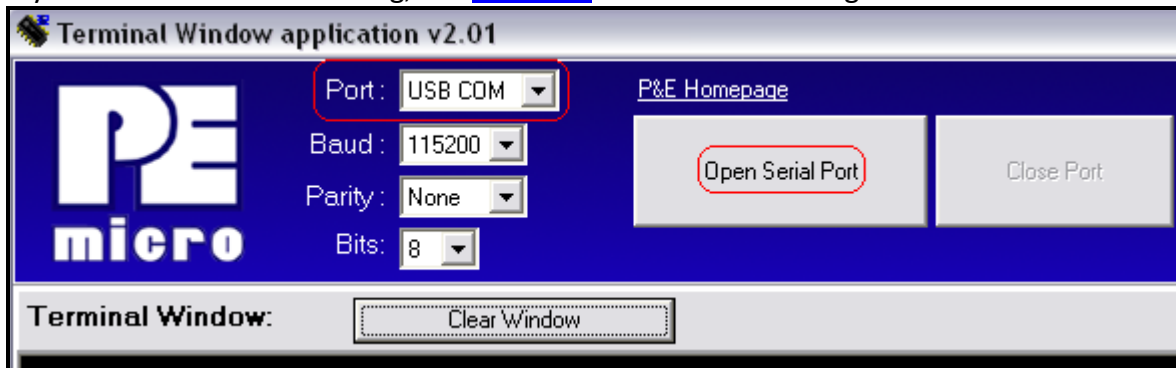
- 6) Make sure Jumper **J21** is installed (default) to power the accelerometer.
- 7) Tilt the board back and forth to see the LED's light up as it is tilted.

Touch Demo

- 8) Press SW1 twice to enter the Touch Demo. Press the touch pad for the corresponding LED, D6 or D7, to toggle on and off. Remove the **TWRPI-SLCD** module if it blocks you from the touch pads.

Terminal Demo

- 9) Open the P&E Terminal Utility by clicking on Start Menu->Programs->P&E OSBDM OSJTAG Virtual Serial 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 -TWR-K53N512 kit does not include a SD card

- 14) Insert an 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. For example, after you insert your SD card, press Enter for shell prompt. Then you can type "**dir**" to access what is in the SD card.

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

- 15) On **TWR-SER** board, make sure jumper **J16** is installed for **2-3**.
- 16) On **TWR-K53N512** board, make sure jumper **J18** is installed (default) so that the USB transceiver is powered on.
- 17) Connect the **TWR-SER** card to your PC using a mini-B to A type USB cable.
- 18) 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.

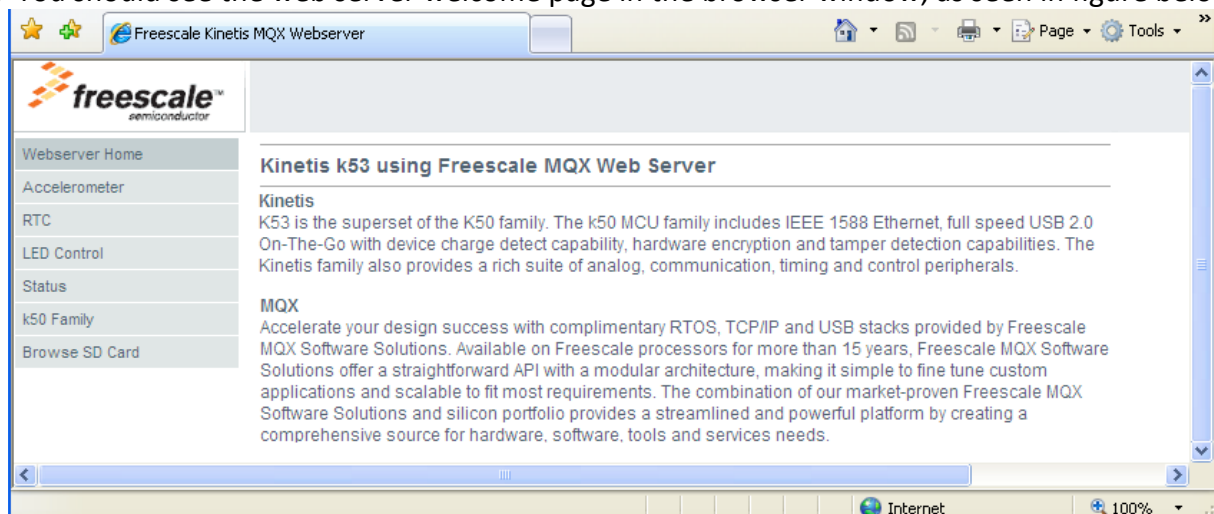
Web Server Demo - The following steps require the TWR-SER peripheral module:

- 19) Ensure to follow jumpers' settings, boards assembly, and cable connections accordingly to the steps in [Section 2](#) under the section **TWR-K53N512 board with the TWR-SER board**.
- 20) Open up a 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.
- 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.
- This Demo has been tested with IE explorer browser.

- 21) 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**, set your subnet mask to **255.255.0.0**, and set your gateway to **192.168.1.1**
- 22) Open a web browser and go to the target device address. In this case, **169.254.3.3**
- 23) You should see the web server welcome page in the browser window, as seen in figure below:



- 24) 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
 - More information on the Kinetis K50 family
- 25) Create and save an index.html file onto an SD card (not included in the TWR-K53N512 kit).
 - 26) Insert the SD card into the TWR-K53N512 module.
 - 27) Click on "Browse SD card" to browse the index.html (created by you) from the SD Card. You can also access any other webpage on the SD card by going to **<board IP address>\sdcard\file_name.html**

4 Development Software and Programming the Board

Instructions below show you how to build and flash the Kinetis Quick Start Demo using MQX 3.7 with both CodeWarrior MCU 10.1 and IAR 6.10.5. You may apply the same steps mentioned in this section to program other projects as well. Make sure you follow the steps in order.

Note : For Windows® Vista and Windows® 7 users, it is recommended to install both the compiler IDE and MQX at a directory other than the default **C:\Program Files** directory.

Note: If you open a project that was developed in IAR 6.10.5 with a newer version of IAR 6.20.x, re-opening the same project in IAR 6.10.5 again will cause compiling issues.

4.1 CodeWarrior Development Studio for Microcontrollers 10.1

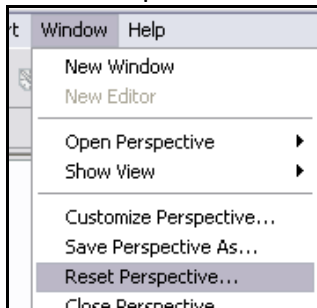
- 1) Optional - See [Section 8](#) for OSJTAG Troubleshooting or follow the directions in [Section 9](#) to update the OSJTAG firmware and drivers. This step is only required if you have trouble with programming via OSJTAG or printing to the utilities of the **P&E OSBDM OSJTAG Virtual Serial Toolkit**
- 2) Install **CodeWarrior Development Studio for Microcontrollers 10.1** from DVD. The Evaluation edition will be required because of the large code size needed for the MQX Quick Start Demo software. Other CodeWarrior licensing versions are also available at www.freescale.com/codewarrior
- 3) You will need to update the latest CW 10.1 Patches in order for CW 10.1 to support K50 family devices. Follow the **Codewarrior MCU V10.1 Update Guide** from DVD or from [this URL](#) to get all of the CW10.1 updates via the online method. You can also use the offline instructions to install the MCU v10.1 Compiler Update, MCU v10.1 Kernel Awareness Update, and MCU 10.1 Update with the files downloaded from the [CodeWarrior update repository \(URL\)](#)
- 4) Install Freescale RTOS **MQX 3.7** from DVD or from www.freescale.com/MQX
- 5) Install Freescale RTOS **MQX 3.7 Patch for TWR-K53N512** from DVD or from www.freescale.com/MQX

Note: The source code for the Quick Start Demos is bundled with **MQX 3.7 Patch for TWR-K53N512**. Once installed, you will be able to find the project at the default installation directory at:

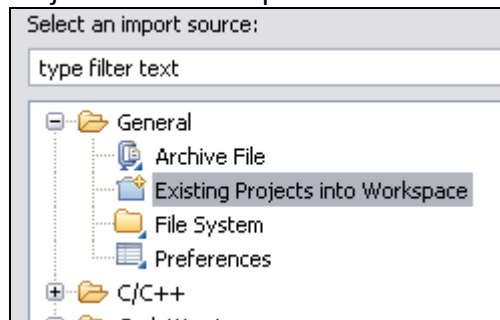
C:\Program Files\Freescale\Freescale MQX 3.7\demo\TWR-K53N512_Quick_Start_Demo

If there are future updates, it will be a file call **TWRK53N512QSDLAB.zip** from the <http://www.freescale.com/twr-k53n512>

- 6) You will need to enable the ESHDC and I2C1 drivers in the BSP library to use this demo. By default, the MQX 3.7 user_config.h had already been configured to run the demos indicated in this document. If you have previously modified that file, please follow **Step 7** to **Step 18**. Otherwise jump to **Step 19**.
- 7) Open CW10.1 At the welcome screen, set the workspace to the MQX installation directory, which, by default will 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

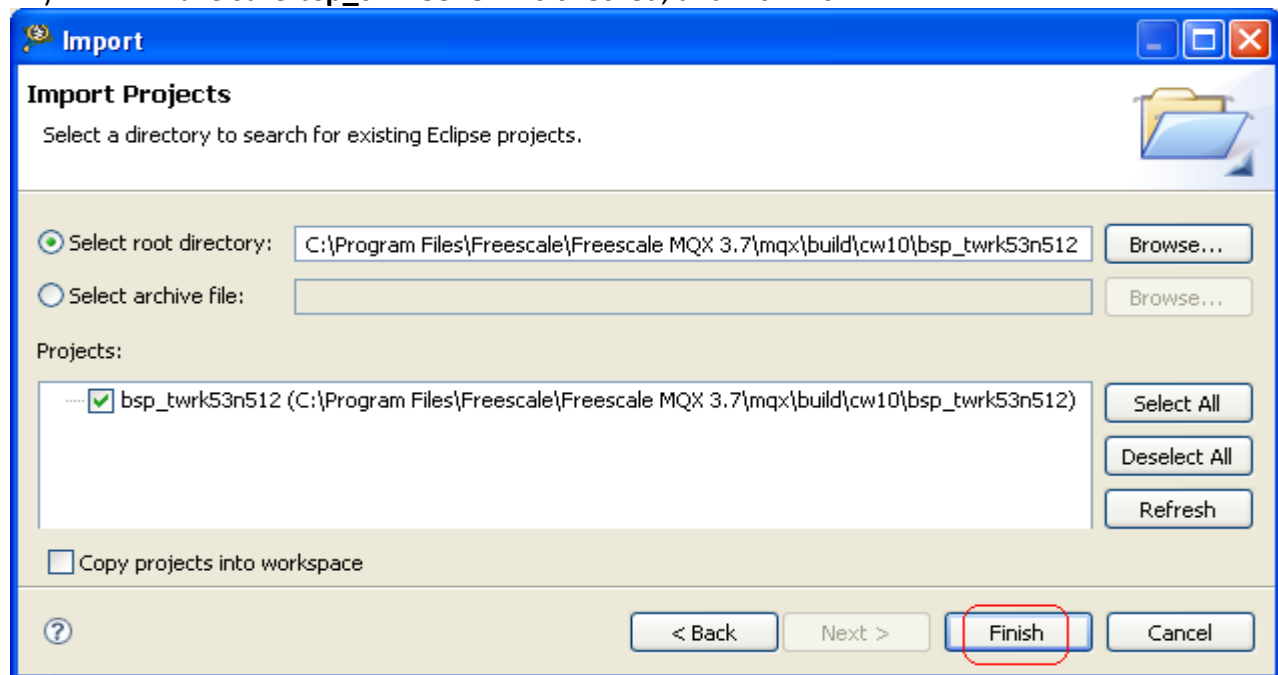


- 11) Click on **File->Import** in the menu bar. In the dialog box that comes up, select “Existing Projects into Workspace” under the General folder. Then click Next

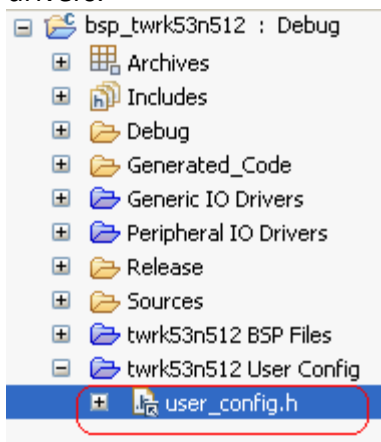


- 12) On the next screen, select the “Select root directory:” option, and click on Browse
- 13) Navigate to the **C:\Program Files\Freescale\Freescale MQX 3.7\mqx\build\cw10\bsp_twrk53n512** directory and hit OK.

- 14) Make sure **bsp_twrk53n512** is checked, and hit Finish




- 15) Open the **user_config.h** file in the **bsp_twrk53n512** project to enable the ESDHC and I2C0 drivers.



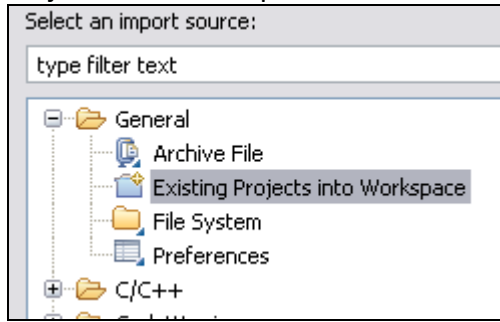
- 16) Change

```
#define BSPCFG_ENABLE_I2C1    0
#define BSPCFG_ENABLE_ESDHC  0
to
#define BSPCFG_ENABLE_I2C1    1
#define BSPCFG_ENABLE_ESDHC  1
```

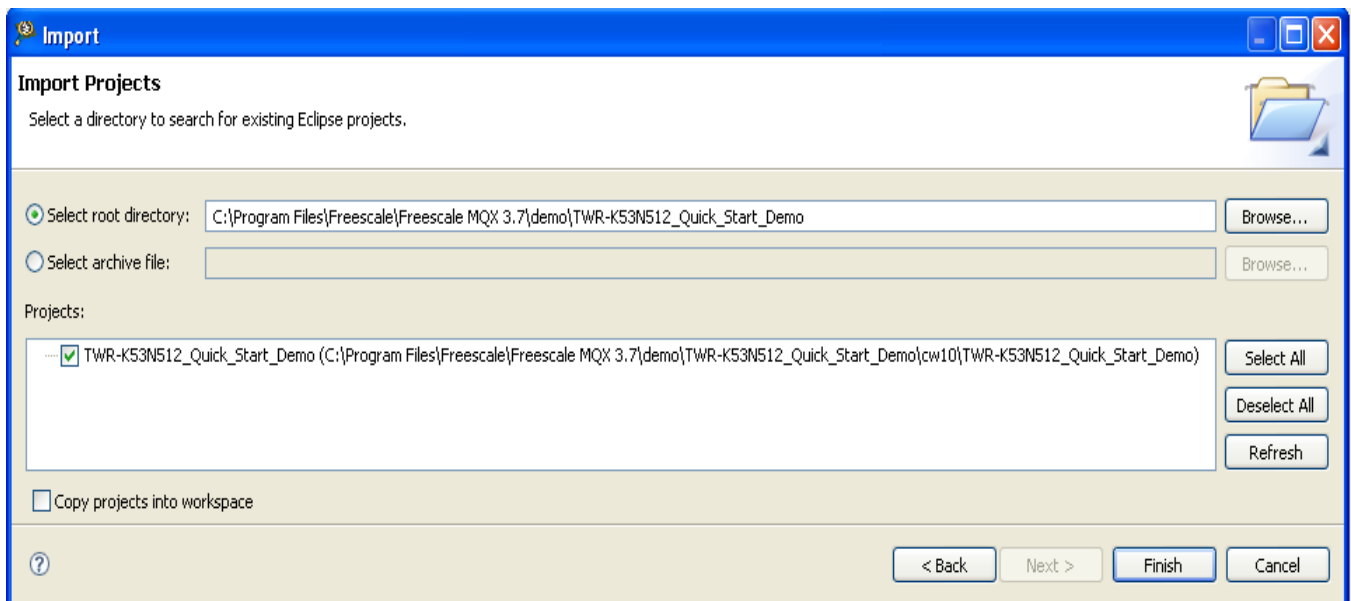
- 17) Save the **user_config.h** file.

- 18) Click on the Make button  to re-compile the K53 BSP library. This makes the edits in the **user_config.h** file take effect.

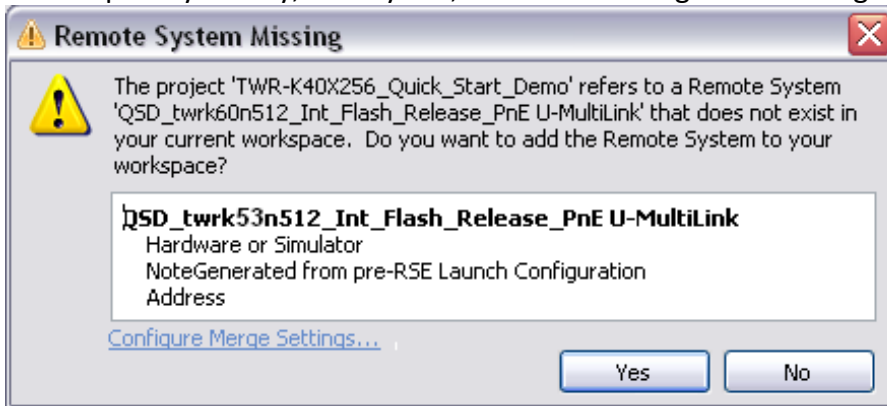
- 19) If you did not install the MQX project in the default **C:\Program Files\Freescale\Freescale MQX 3.7** directory during installation, you must first recompile the rest of the MQX libraries. See the MQX release notes for more information before continuing on with the lab.
- 20) Next we need to import the Quick Start Demo into our CW10 workspace.
- 21) 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



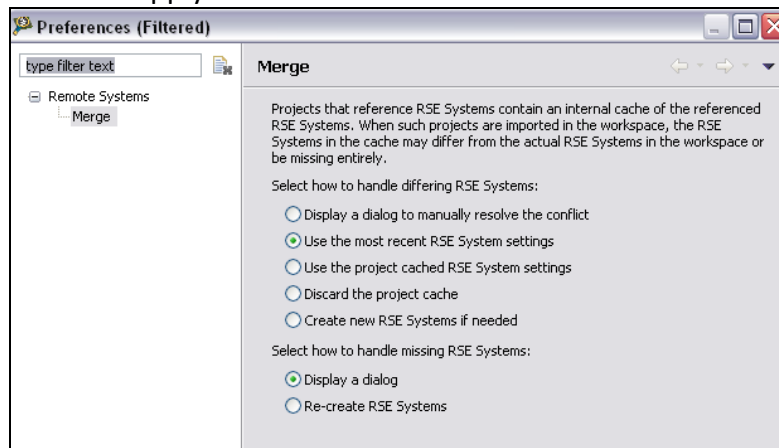
- 22) On the next screen, select the “Select root directory:” option, and click on Browse
- 23) Navigate to the **C:\Program Files\Freescale\Freescale MQX 3.7\demo\ TWR-K53N512_Quick_Start_Demo** directory and hit OK.
- 24) Make sure **TWR-K53N512_Quick_Start_Demo** is checked, and hit **Finish**. **Please read the instructions below for the error dialog box that will likely come up after hitting Finish.**



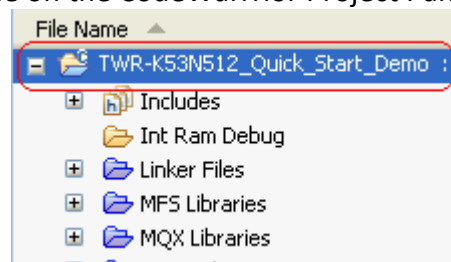
25) At this point you may, or may not, see the following error message:



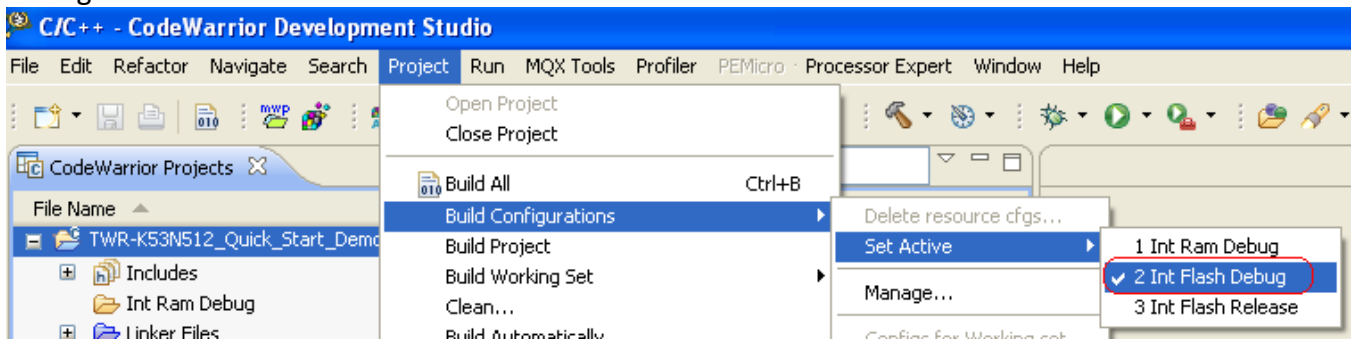
- 26) When importing a project, you will often see a “Remote Systems Changed” or “Remote System Missing” dialog box. This occurs when CodeWarrior detects a mismatch between the RSE (Remote System Explorer) settings in the project and the settings in the framework, or if the RSE system is missing in the project.
- 27) Click on “Configure Merge Settings” to avoid having the dialog box re-come up for this project.
- 28) On the screen that comes up, select the “Use the most recent RSE System settings” option, and click “Apply” and then “OK”



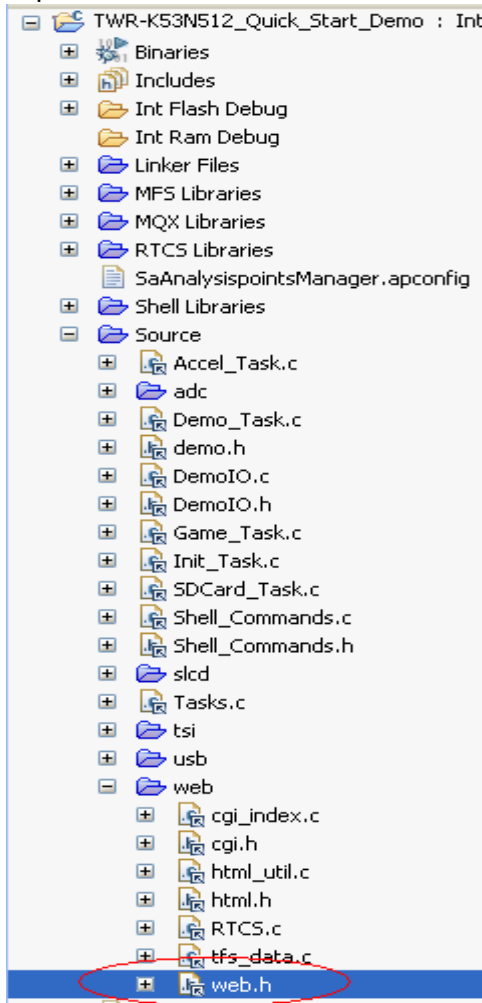
- 29) Then hit “Yes” to get rid of the error message. It is very important to not hit “No”
- 30) If you did hit “No”, you will need to reset the RSE project settings. First close the project by right-clicking on it and selecting “Close Project”, and then open it up by right clicking on the project and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



- 33) On the CodeWarrior 10.1 Menu choose “Project -> Build Configuration -> Set Active -> Int Flash Debug”.



- 34) Open the web.h file located in the “Source/web” group in the project tree.




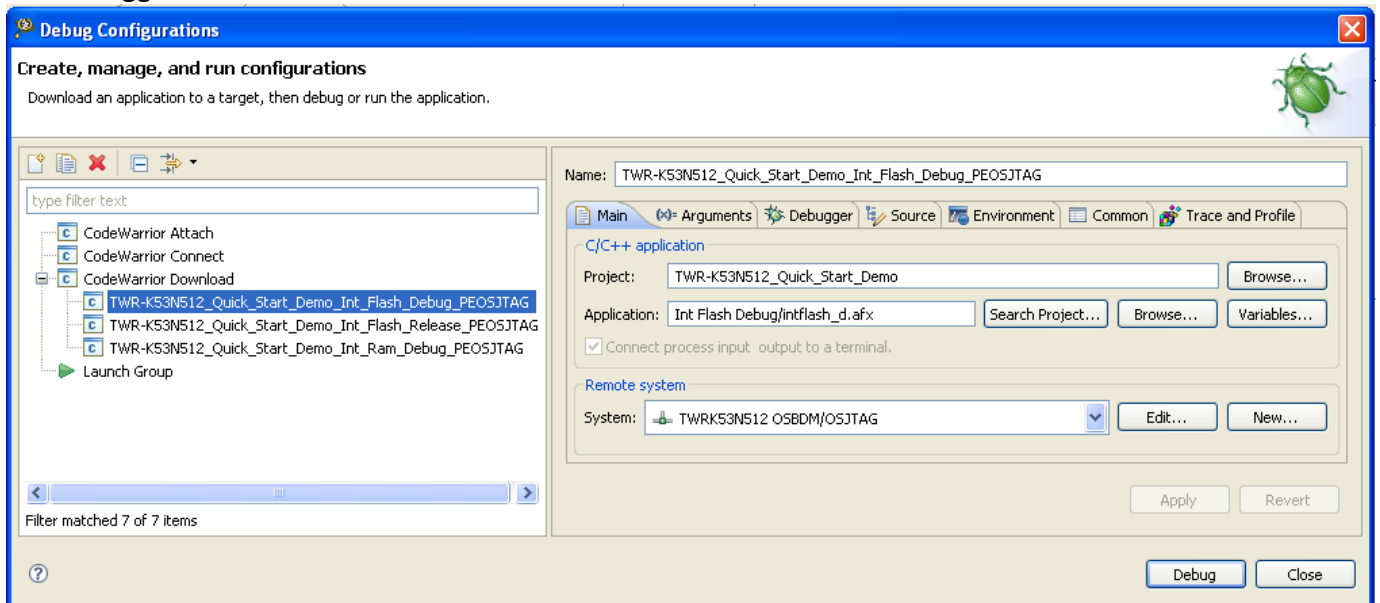
- 35) You can change the default IP address by modifying the ENET_IPADDR definition. For example, to set the target address to 169.254.3.3, and the line will be:

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

- 36) You could also use DHCP instead of a static IP address by setting **DEMOCFG_ENABLE_DHCP** to **1** in **web.h**

- 37) Save the file after you are done.

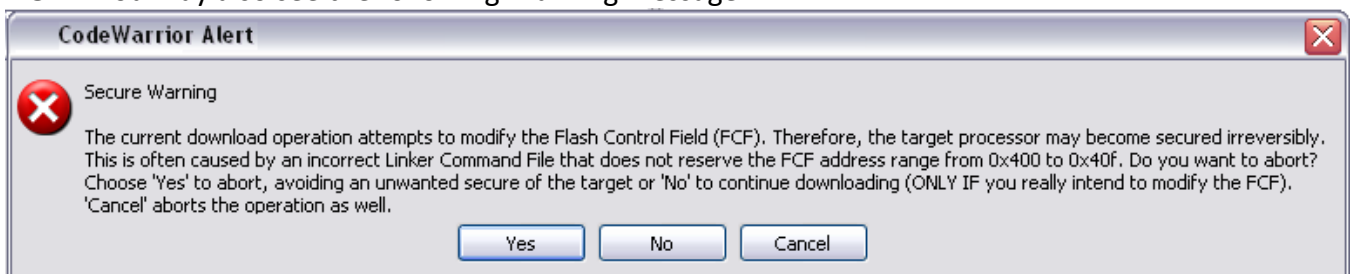
- 38) Build the project by clicking on the Hammer icon in the toolbar 
- 39) Double click on the **TWR-k53n512_Int_Flash.ttf** file. In the dialog box that comes up, click “OK”. The file will then open. You can then close it. You only need to do this step once after initially opening the project. The dialog box may also not appear, so do not worry if that is the case.
- 40) Click **Run->Debug Configurations...** in the menu bar, and select the **TWR-K53N512_Quick_Start_Demo_Int_Flash_Debug_PESJTAG** configuration. Then hit the debug button in the bottom of the window. The code will then be flashed to the board and the debugger will start.



Note: Note for future reference, make sure that the target memory selection here is the same as **step 33**. For example if the build setting is flash base, the debug should be configured as flash base as well.


Note: 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. In that case, unplug your board. Put a jumper on **J28** 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 **J28**, and hit “OK” again on the dialog box.

NOTE: 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.

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

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

4.2 IAR Embedded Workbench for ARM 6.10.5

43) Optional - See [Section 8](#) for OSJTAG Troubleshooting or follow the directions in [Section 9](#) to update the OSJTAG firmware and drivers. This step is only required if you have trouble with programming via OSJTAG or printing to the utilities of the **P&E OSBDM OSJTAG Virtual Serial Toolkit**

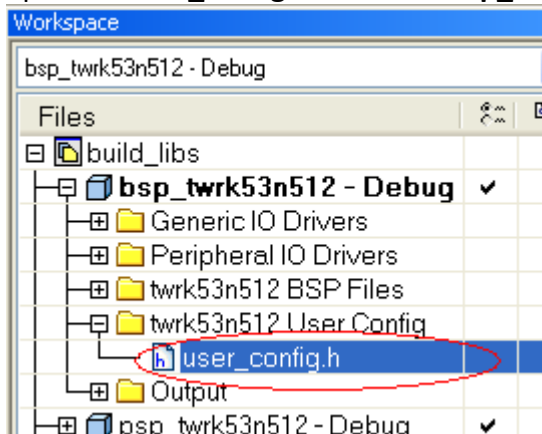
- 1) Install **IAR Embedded Workbench for ARM 6.10.5** from DVD. The Evaluation edition or Full edition will be required because of the large code size needed for the MQX Quick Start Demo software. Newer versions are also available from www.IAR.com.
- 2) Install Freescale RTOS **MQX 3.7** from DVD or from www.freescale.com/MQX
- 3) Install Freescale RTOS **MQX 3.7 Patch for TWR-K53N512** from DVD or from www.freescale.com/MQX

Note: The source code for the Quick Start Demos is bundled with **MQX 3.7 Patch for TWR-K53N512**. Once installed, you will be able to find the project at the default installation directory at:

C:\Program Files\Freescale\Freescale MQX 3.7\demo\TWR-K53N512_Quick_Start_Demo

If there are future updates, it will be a file call **TWRK53N512QSDLAB.zip** from the <http://www.freescale.com/twr-k53n512>

- 4) You will need to enable the ESHDC and I2C1 drivers in the BSP library to use this demo. By default, the MQX 3.7 user_config.h had already been configured to run the demos indicated in this document. In the case that you have previously modified that file, please follow **Step 6** to **Step 10**. Otherwise jump to **Step 11**
- 5) Open the workspace at **C:\Program Files\Freescale\Freescale MQX 3.7\config\twrk53n512\iar\build_libs.eww**
- 6) Open the **user_config.h** file in the **bsp_twrk53n512 – Debug** project




- 7) Change


```
#define BSPCFG_ENABLE_I2C1    0
#define BSPCFG_ENABLE_ESDHC  0
to
#define BSPCFG_ENABLE_I2C1    1
```

```
#define BSPCFG_ENABLE_ESDHC 1
```

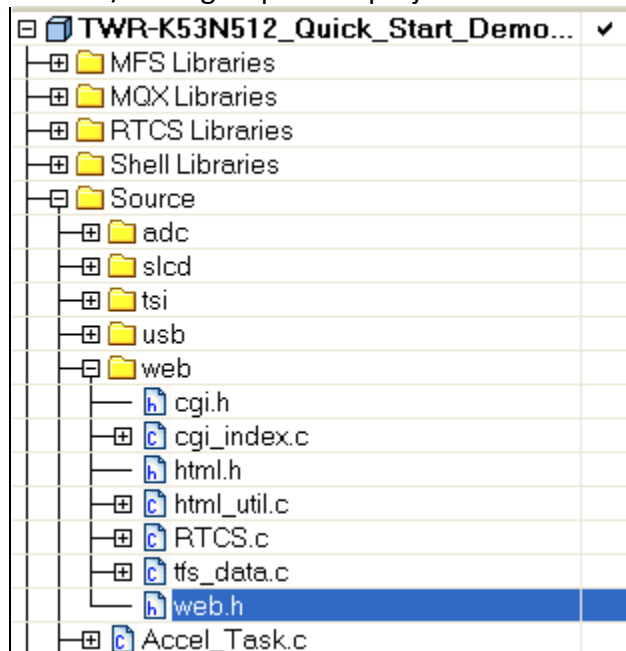
8) Save the **user_config.h** file

9) Click on the Make button  to re-compile the K53 BSP library. This makes the edits in the user_config.h file take effect. Make sure **bsp_twrk53n512 – 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) Open the workspace file that will be located at **C:\Program Files\Freescale\Freescale MQX 3.7\demo\TWR-K53N512_Quick_Start_Demo\iar\TWR-K53N512_Quick_Start_Demo.eww**

12) 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.



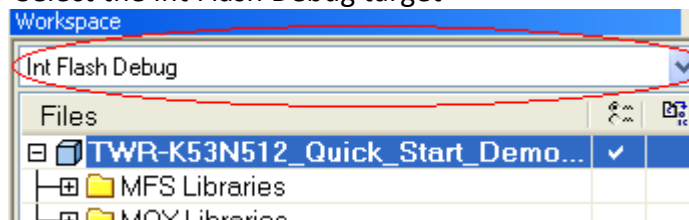
13) You can change the default IP address by modifying the ENET_IPADDR definition. For example, to set the target address to 169.254.3.3, and the line will be:



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

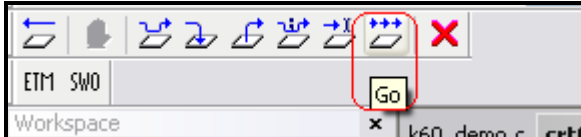
14) You could also use DHCP instead of a static IP address by setting **DEMOCFG_ENABLE_DHCP** to **1** in **web.h**

15) Save the file after you are done.

16) Select the Int Flash Debug target





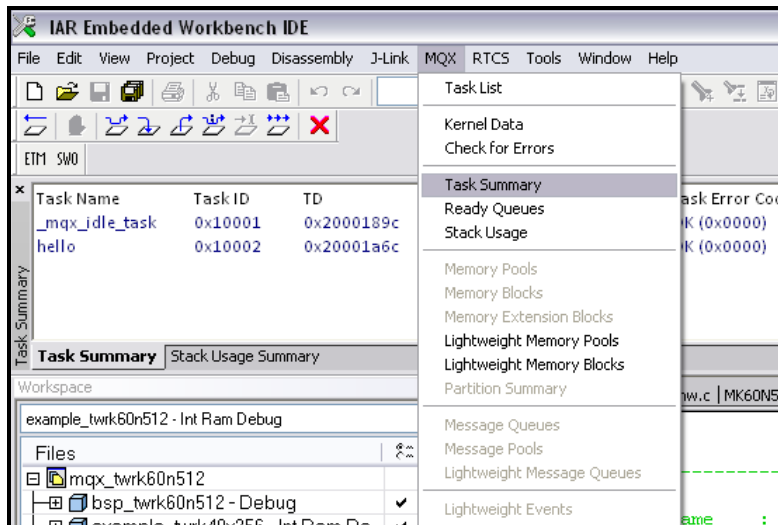
- 17) Compile the project by clicking the Make icon  (or right clicking on the project and select “Make”).
- 18) After compilation completes, download the code to the board and start the debugger by pressing the “Download and Debug” button .
- 19) The code will download Flash, and the debugger screen 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 these 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. More information on the TAD is available in the [Release Notes](#).



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.

Then the Segment LCD task is started, which initializes the TWRPI-SLCD and determines what to display on the SLCD depending on which mode it is currently in.

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 **Accel_Task** is started, which polls the MMA7660 sensor on the **TWR-K53N512** over I2C.

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 **SLCD_Task** and **Shell_Task** are started after that. Then lastly, 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 **mkdfs.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.

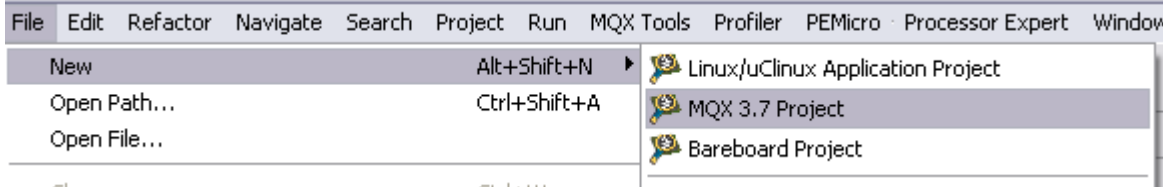
The initialization task then calls **_task_block()** to be permanently blocked, and the demo is now up and ready. See the comments in the source code for more information on how the demo works.

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 CW10.1, use the following steps:

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



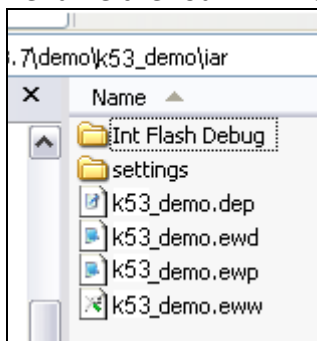
- 2) Follow the wizard to setup and create a new MQX project.

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

- 1) Make sure you have not modified the files in the K53 Quick Start Demo
- 2) Double click on the **create_new_k53_mqx_project_iar.exe** file inside the K53 Quick Start Demo.
- 3) Type in the new project's name, with no spaces, such as **k53_hello**
- 4) Hit Enter and wait for it to finish creating the new project. The script will copy the project files from the K53 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-K53N512_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, **k53_demo**
- 3) Inside the k53_demo folder, go into the **iar** folder
- 4) Rename the four IAR files to k53_demo



- 5) Open the k53_demo.eww file with a text editor instead of IAR and change the following line to point to the new project name:

```
<?xml version="1.0" encoding="iso-8859-1"?>

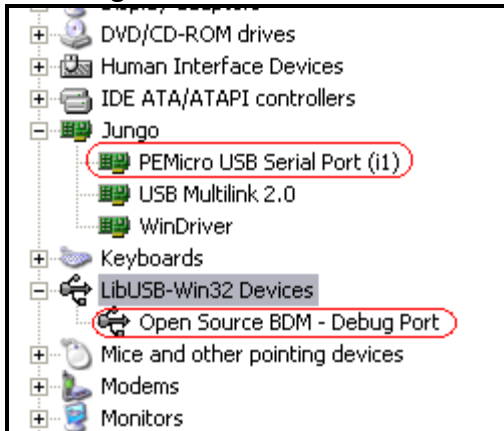
<workspace>
  <project>
    <path>$WS_DIR$\k53_demo.ewp</path>
  </project>
  <batchBuild/>
</workspace>
```

- 6) Save the file
- 7) Open the k53_demo.eww workspace

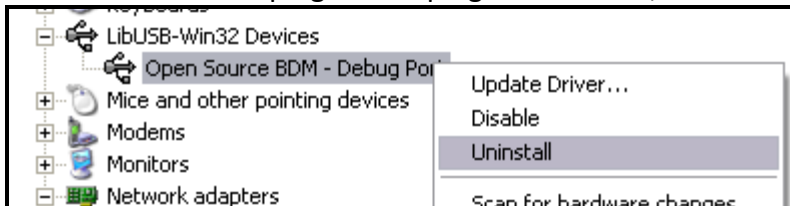
8 OSJTAG

Open Source JTAG (also known as OSBDM on Freescale tower boards) allows a user to program, debug, and get serial data from Kinetis devices via a single USB connection. The firmware is programmed on a Freescale MCFS08JM60 MCU on the underside of the TWR-K53N512 board.

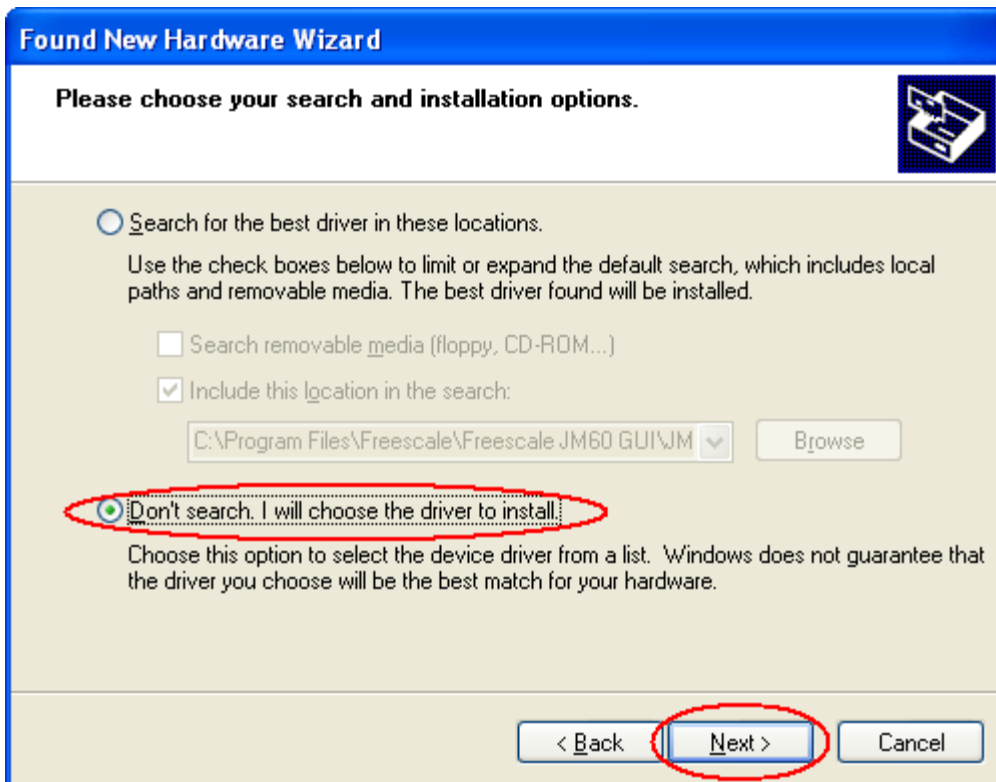
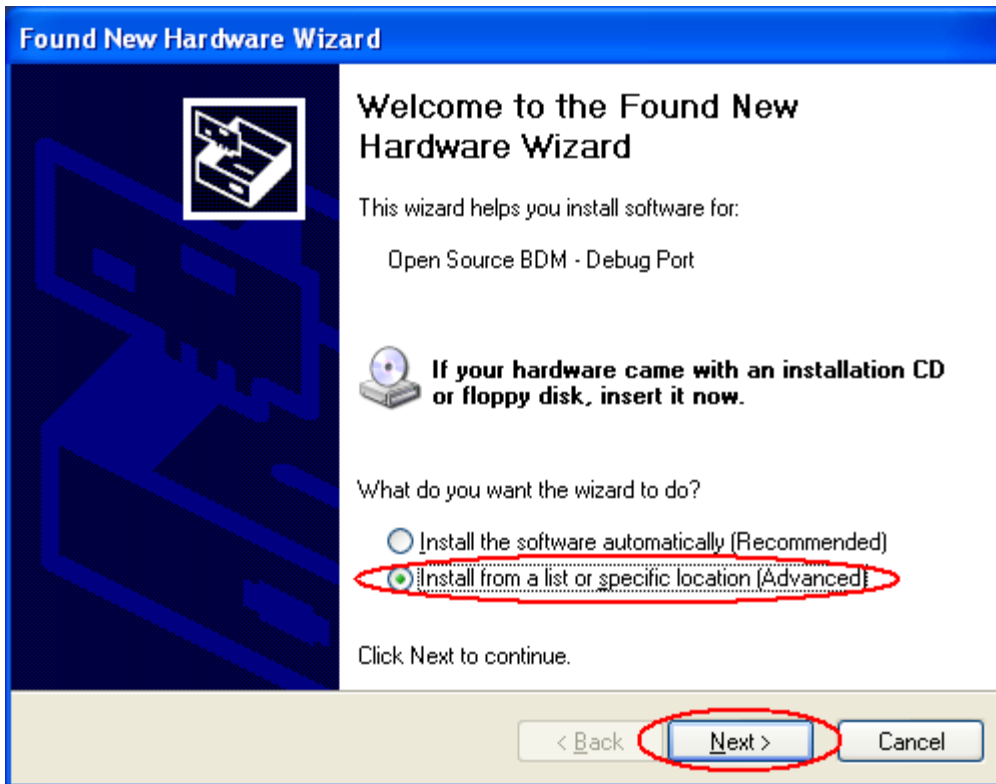
When the tower board is plugged in, it should enumerate as a composite device, with one driver for debugging, and the other as a serial port. If you go to the Device Manager you should see the following:

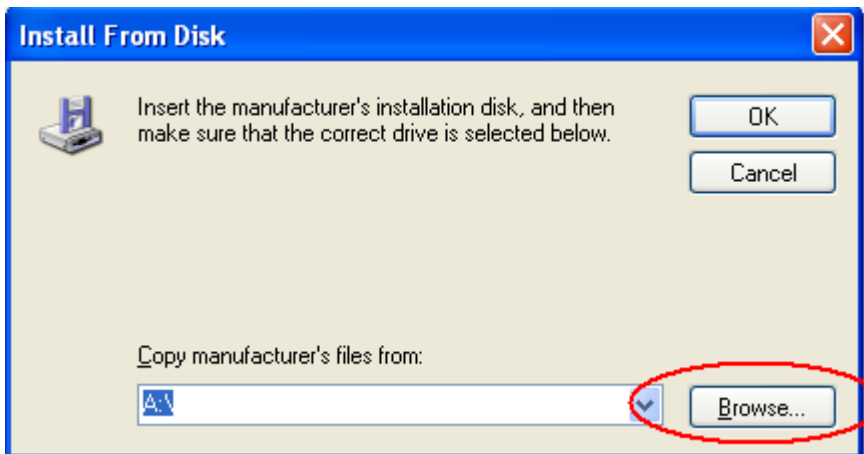
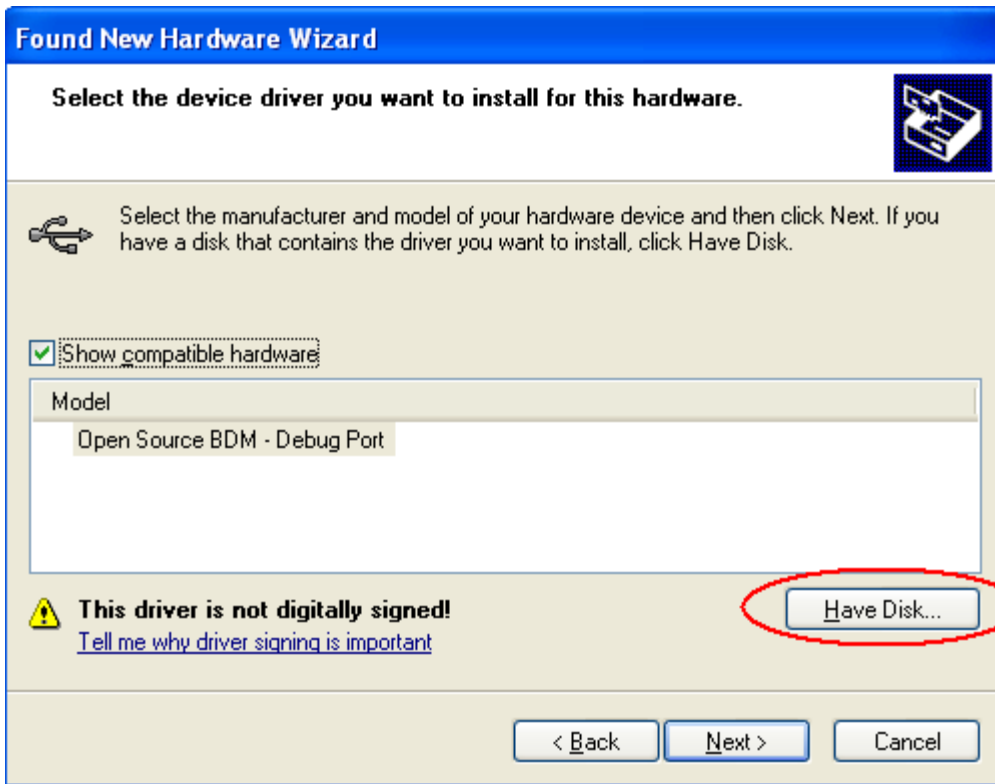


If you only see it enumerate as the Open Source BDM Debug Port, then your computer may automatically picking up an outdated driver. To fix this, right click on the OSBDM driver and select "Uninstall". Then unplug and re-plug in the board, and it should enumerate correctly.



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





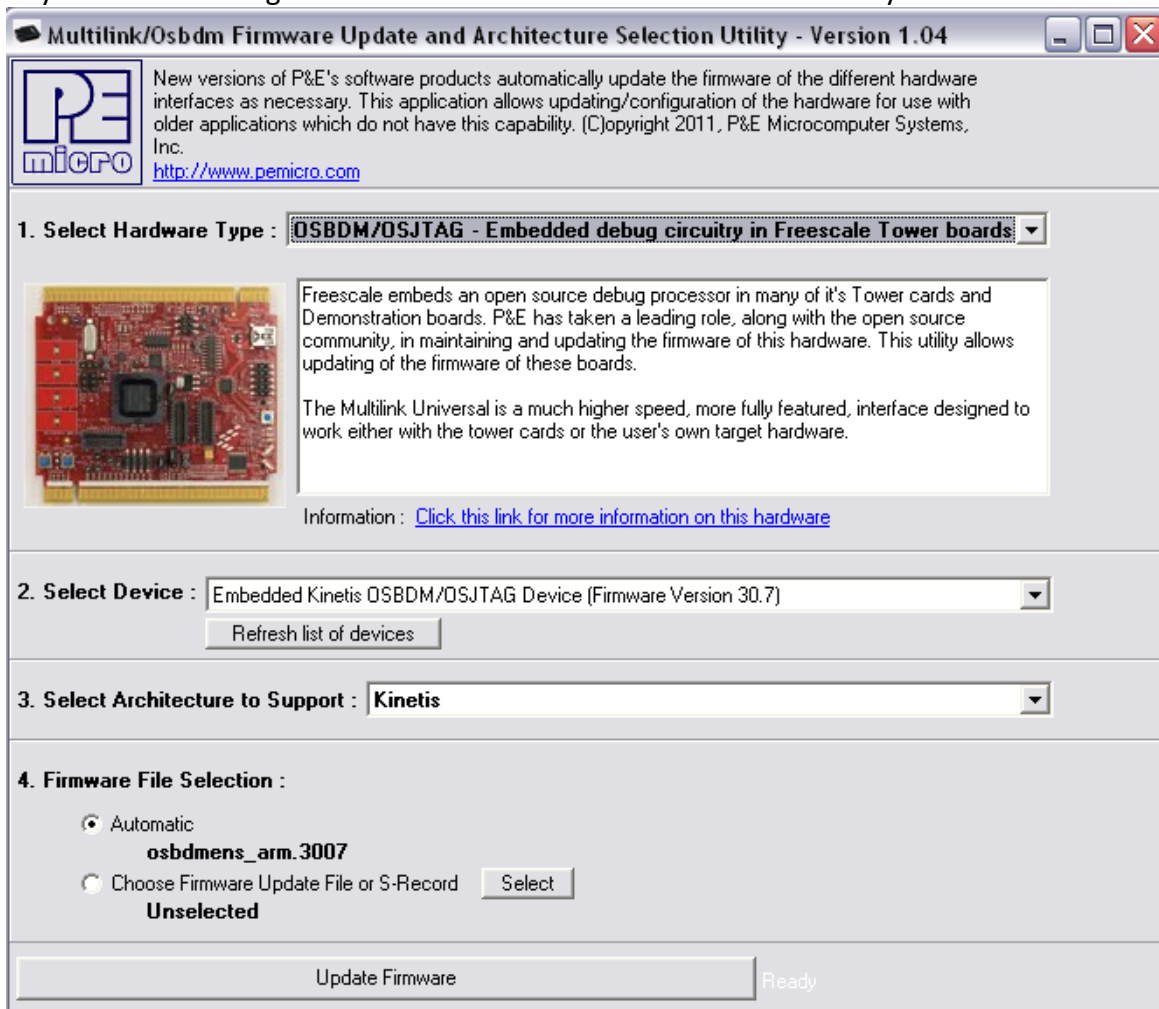
- For the Open Source BDM – Debug Port, use the driver at:
C:\pemicro\kinetis_tower_toolkit\Drivers\osbdm\OSJTAG_Debug_Interface_libusb.inf
- For the PEMicro USB Serial Port (i1), use the driver at:
C:\pemicro\kinetis_tower_toolkit\Drivers\osbdm\OSJTAG_Serial_Interface_windriver_version.inf

9 Update 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. To ensure compatibility between the drivers, firmware, and terminal window, the latest versions of each need to be installed.

NOTE: Out of the box TWR-K53N512 is programmed with OSBDM firmware version 29.07. This version will support IAR ARM version 6.10.5. OSBDM firmware version 30.x and up will not support IAR 6.10.5. It will only support IAR 6.20 and up.

- 1) Download and install both of the latest **P&E Firmware Updates and Recovery** and **OSBDM Virtual Serial Toolkit** programs which can be found at <http://www.pemicro.com/osbdm>
- 2) Make sure your tower board is plugged in, and run the **P&E Firmware Updater Utility** to use the OSJTAG boot loader to upgrade to the latest OSJTAG version.
- 3) Under “Select Hardware Type” make sure OSBDM/OSJTAG is selected. It should automatically detect your board settings and fill out the rest of the fields automatically.



- 4) Click on “Update Firmware” to update the firmware. It will prompt you to disconnect the USB cord from your computer, and then short the JM60 boot loader jumper header. It is **J28** from the **TWR-K53N512** board and is labelled **JM60 BOOT** on the silk screen. Then re-connect the board to your computer.
- 5) The firmware will then be updated on your board. When it is finished, it will prompt you again to disconnect the USB cable, remove the jumper, and then re-connect the board again. OSJTAG is now updated.

10 Resources

For most up-to-dated resources, please visit <http://www.freescale.com/TWR-K53N512> and look for the following titles:

- **Freescale MQX RTOS 3.7 Patch for TWR-K53N512** includes source code of the quick start demos and MQX K50 family devices support for IAR 6.10.5 and CW 10.1. Any future updates to Quick Start Demo Software and Lab Guide will be posted online as a zip package called TWRK53N512QSDLAB.
- **TWR-K53N512-UM**: TWR-K53N512 board User's Manual.
- **KQRUG**: Most up-to-date Kinetis Peripheral Module Quick Reference.
- **KINETIS512_SC**: Most software mentioned in the in the KQRUG document. Software has been updated to support TWR-K53N512.
- **TWRK53QSG**: TWR-K53N512-KIT Quick Start Guide. Also included in this DVD.
- Under the Documentation tab of the site, a series of software and documents are also available for applications such as **Spirometer, Pulse Oximeter, Activity Monitor, Blood Pressure Monitor, and Electrocardiograph & Heart Rate Monitor**.