TWR-K60D100M Quick Start Demo

Lab Guide for MQX 4.0

Rev. 1

Freescale Semiconductor Inc.
TWRK60D100MQSDLAB

Contents

1 Purpose 3

2 Configure Hardware 3

3 Update OSJTAG 3

4 Run the Demo....5

5 Development Software and Programming the Board 7

5.1 CodeWarrior for Microcontrollers 10.4....8

5.2 IAR Embedded Workbench for ARM 6.50.6 11

6 Task Aware Debugging....14

7 Demo Software Overview 14

8 Creating a new MQX project 15

9 OSJTAG....16

1 Purpose

This lab document describes how to use the TWR-K60D100M 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, 3, and 4 to run the demo. Use Section 5 to learn how to re-flash the demo on the board.

2 Configure Hardware

If running the TWR-K60D100M board in stand-alone mode:

1) Insert the battery into the battery socket on the underside of the board
2) Change **J12** to pins 2-3 to connect the VBAT domain to run off the battery
3) Add a jumper on **J5** to use the potentiometer.

If using the TWR-K60D100M 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-K60D100M board:
    a. Change **J10** 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-K60D100M 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) Note that the default jumper settings are in the Quick Start Guide for the TWR-K60D100M and TWR-SER if you want to change back to the default settings after you are done running the demo.

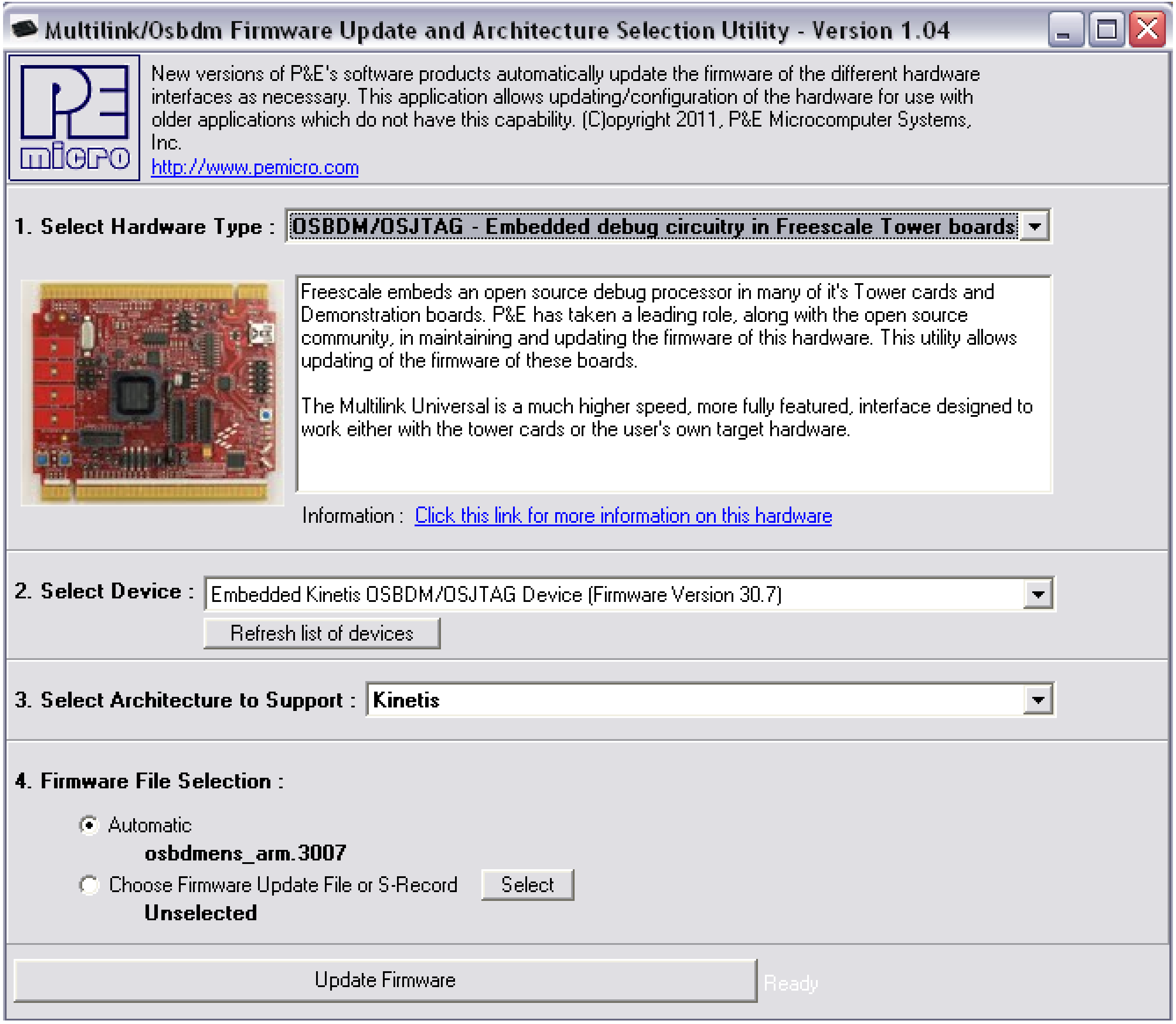
3 Update OSJTAG

Open Source JTAG 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.

First 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

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.

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.



Multilink/Osbdm Firmware Update and Architecture Selection Utility - Version 1.04

New versions of P&E's software products automatically update the firmware of the different hardware interfaces as necessary. This application allows updating/configuration of the hardware for use with older applications which do not have this capability. (C)opyright 2011, P&E Microcomputer Systems, Inc.
http://www.pemicro.com

1. Select Hardware Type : OSBDM/OSJTAG - Embedded debug circuitry in Freescale Tower boards

Freescale embeds an open source debug processor in many of it's Tower cards and Demonstration boards. P&E has taken a leading role, along with the open source community, in maintaining and updating the firmware of this hardware. This utility allows updating of the firmware of these boards.

The Multilink Universal is a much higher speed, more fully featured, interface designed to work either with the tower cards or the user's own target hardware.

Information : Click this link for more information on this hardware

2. Select Device : Embedded Kinetis OSBDM/OSJTAG Device (Firmware Version 30.7)

3. Select Architecture to Support : Kinetis

4. Firmware File Selection :

● Automatic
osbdmens_arm.3007

○ Choose Firmware Update File or S-Record
Unselected

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 **J19** and is labelled **JM60 Boot** on the silk screen. Then re-connect the board to your computer.

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.

4 Run the Demo

1) Connect a mini-B USB cable to the TWR-K60D100M tower module.
2) 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 9 for troubleshooting and make sure you completed the steps listed in Section 3.

Touch Demo

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

Memory Game

4) 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!
5) Press SW1 to go back to the touch and accelerometer demo

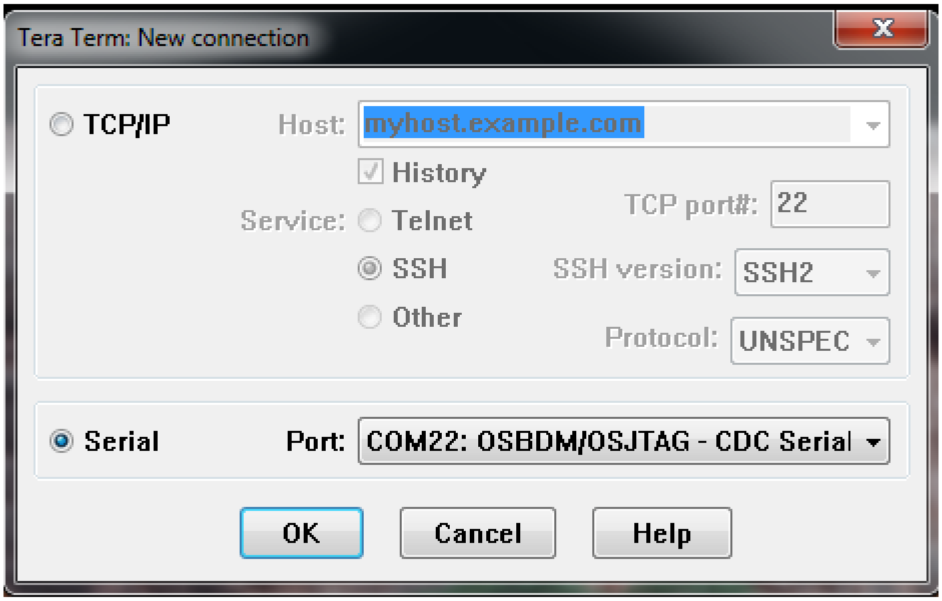
Accelerometer Demo

6) Tilt the board back and forth to see the LED's light up as it is tilted
7) 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

8) Open a terminal window (115200 baud, 8 data bits, no parity, 1 stop bit) and connect to the **OSBDM/OSJTAG – CDC Serial** port.

   Note: If you disconnect power, you will need to close the terminal window, connect power, and then re-open the terminal window



Tera Term: New connection

○ TCP/IP Host: myhost.example.com

☑ History

Service: ○ Telnet TCP port#: 22

● SSH SSH version: SSH2

○ Other Protocol: UNSPEC

● Serial Port: COM22: OSBDM/OSJTAG - CDC Serial

9) 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.
10) Type “**settime HH:MM**” to adjust the time to HH:MM.
11) 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

12) 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:

13) Using a second USB mini-B cable, plug it into the USB mini-B connector on the TWR-SER.
14) 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.

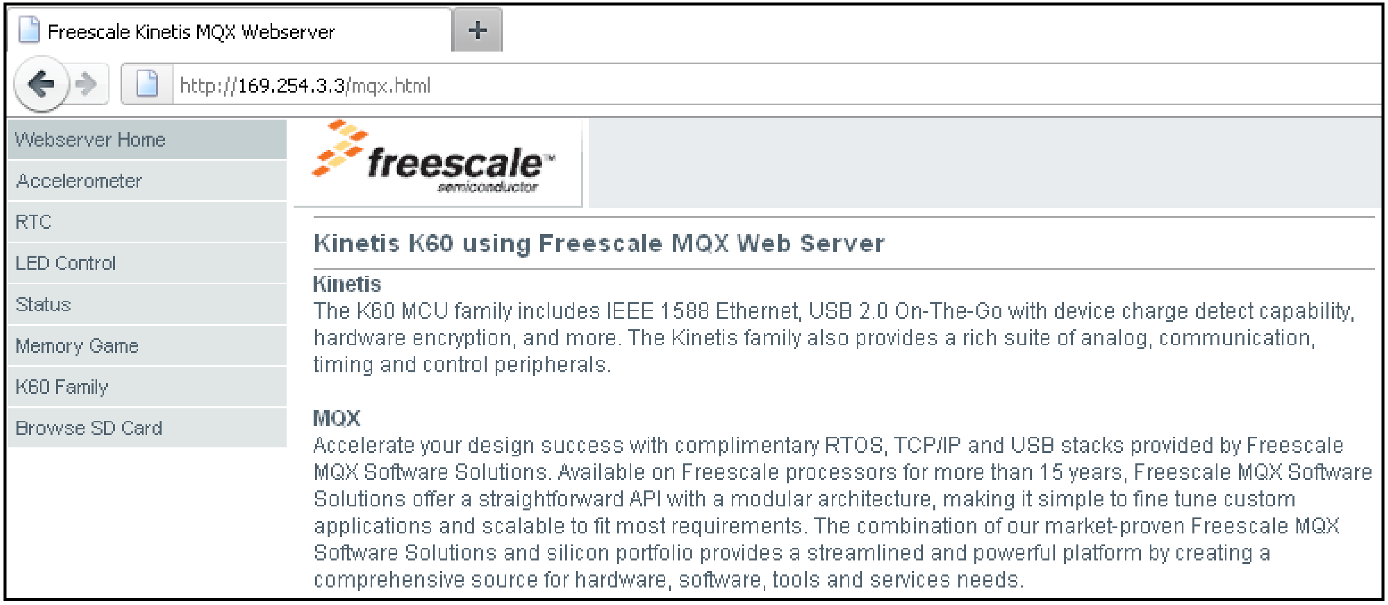
Web Server Demo - The following steps require the TWR-SER peripheral module:

15) 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.

16) 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 on your computer **169.254.3.4** and

your subnet mask to **255.255.0.0**. Make sure the computer IP address is different than the board address

17) Open a web browser and go to the target device address. In this case, **169.254.3.3**
18) You should see the web server welcome page in the browser window, as seen in figure below:



Kinetis K60 using Freescale MQX Web Server

Kinetis

The K60 MCU family includes IEEE 1588 Ethernet, USB 2.0 On-The-Go with device charge detect capability, hardware encryption, and more. The Kinetis family also provides a rich suite of analog, communication, timing and control peripherals.

MQX

Accelerate your design success with complimentary RTOS, TCP/IP and USB stacks provided by Freescale MQX Software Solutions. Available on Freescale processors for more than 15 years, Freescale MQX Software Solutions offer a straightforward API with a modular architecture, making it simple to fine tune custom applications and scalable to fit most requirements. The combination of our market-proven Freescale MQX Software Solutions and silicon portfolio provides a streamlined and powerful platform by creating a comprehensive source for hardware, software, tools and services needs.

19) 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
20) 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**

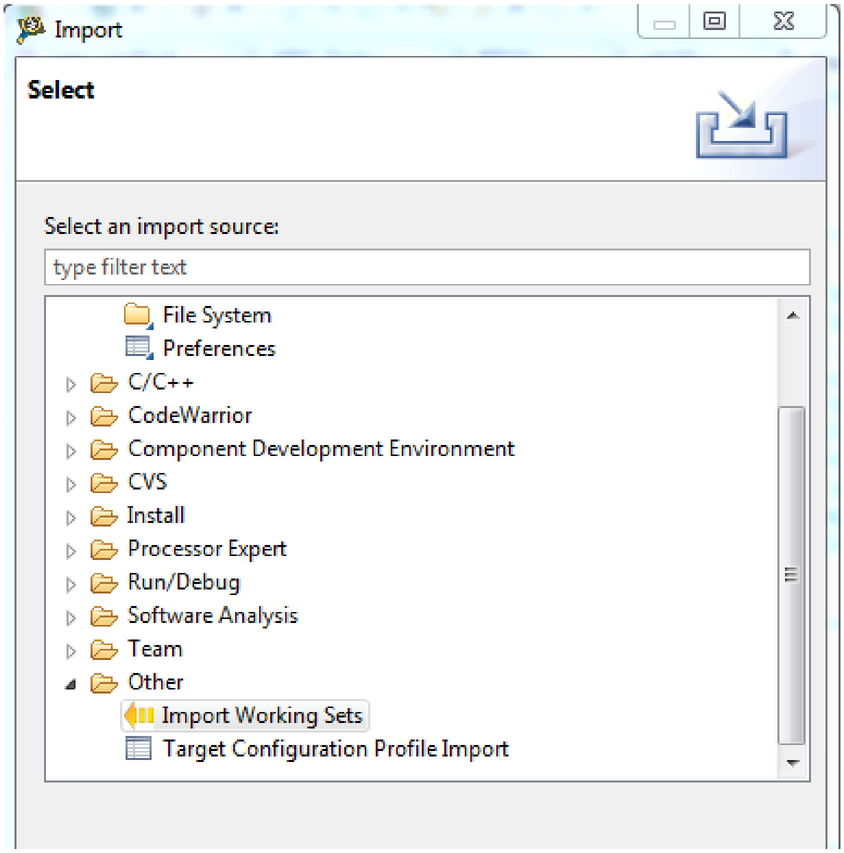
    (This feature is not part of the pre-flashed demo, and is only in web-downloaded version. Also note the errata described in the SD Card demo section above for **Rev D** TWR-K60D100M boards.)

5 Development Software and Programming the Board

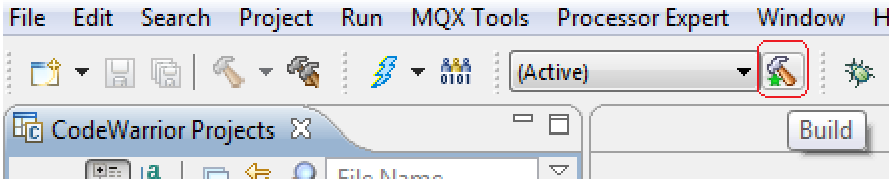
The following instructions describe how to build and flash the Kinetis Quick Start Demo using MQX 4.0 with both CodeWarrior MCU 10.4 and IAR 6.50.6

5.1 CodeWarrior for Microcontrollers 10.4

1) Follow the directions in Section 3 to update the OSJTAG firmware and drivers. See Section 9 for OSJTAG Troubleshooting.
2) Install CodeWarrior for Microcontrollers 10.4. The Evaluation edition or Standard version or higher will be required because of code size.
3) Install any CodeWarrior patches available.
4) Install MQX 4.0. You can find this from the MQX website
5) You must then compile the MQX libraries for the TWR-K60D100M board. The workspace at **<mqx_dir>\config\twrk60d100m\cw10\twrk60d100m.wsd** has all the libraries to build and can be imported going to the **File->Import** and then **Other->Import Working Sets**.



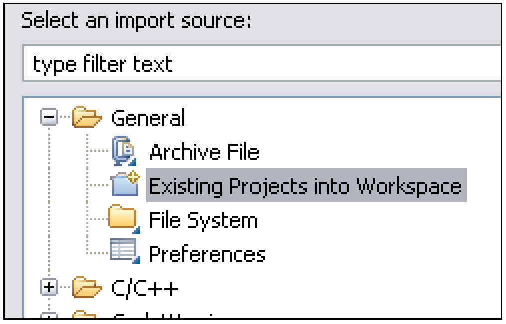
6) Click on the Build All button to re-compile all the K60 libraries.


File Edit Search Project Run MQX Tools Processor Expert Window H
(Active)
CodeWarrior Projects
Build
File Name

Use the default "Debug" targets. See the **Getting Started with Freescale MQX™ RTOS** document in the **<mqx_dir>\doc** folder for more information if needed on compiling the libraries before continuing on with the lab.

7) If you did not install the K60 Quick Start Demo to the default location inside of **C:\Freescale\Freescale_MQX_4_0\demo** folder, move the folder **TWR-K60D100M_Quick_Start_Demo** to that location. The path after the move should be **C:\Freescale\Freescale_MQX_4_0\demo\TWR-K60D100M_Quick_Start_Demo**
8) If MQX is installed in another location, move the folder to the **<mqx_dir>\demo** directory at that location instead.

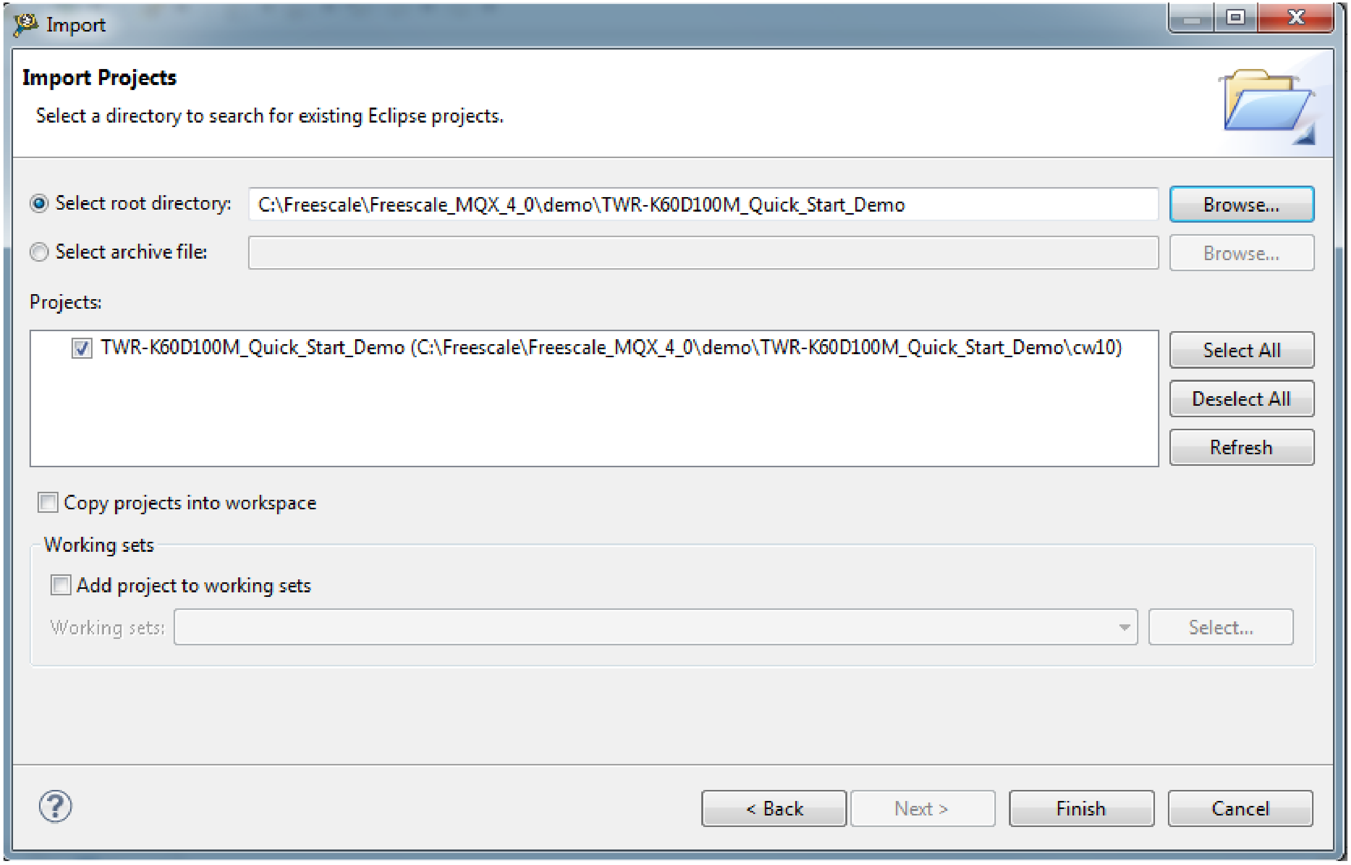
9) Next we need to import the Quick Start Demo into our CW10 workspace.
10) Click on **File->Import** in the menu bar. In the dialog box that comes up, select "Existing Projects into Workspace" under the General folder. Then click Next



Select an import source:

type filter text

11) On the next screen, select the "Select root directory:" option, and click on Browse
12) Navigate to the **C:\Freescale\Freescale_MQX_4_0\demo\TWR-K60D100M_Quick_Start_Demo** directory and hit OK.
13) Make sure **TWR-K60D100M_Quick_Start_Demo** is checked, and hit **Finish**.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\Freescale\Freescale_MQX_4_0\demo\TWR-K60D100M_Quick_Start_Demo

○ Select archive file:

Projects:

☑ TWR-K60D100M_Quick_Start_Demo (C:\Freescale\Freescale_MQX_4_0\demo\TWR-K60D100M_Quick_Start_Demo\cw10)

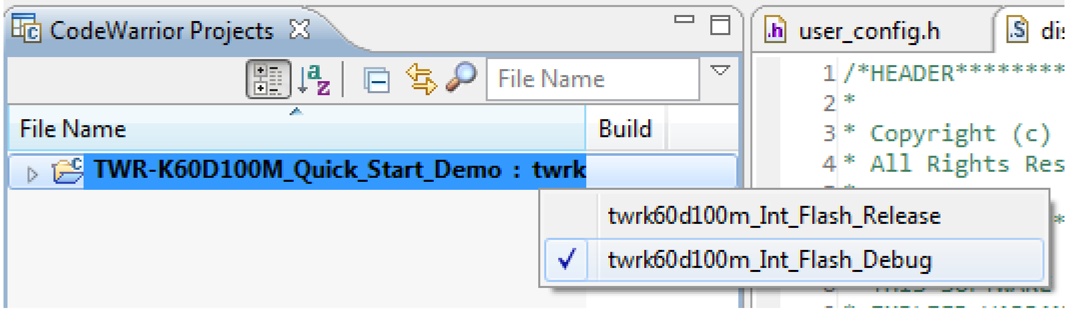
□ Copy projects into workspace

Working sets

□ Add project to working sets

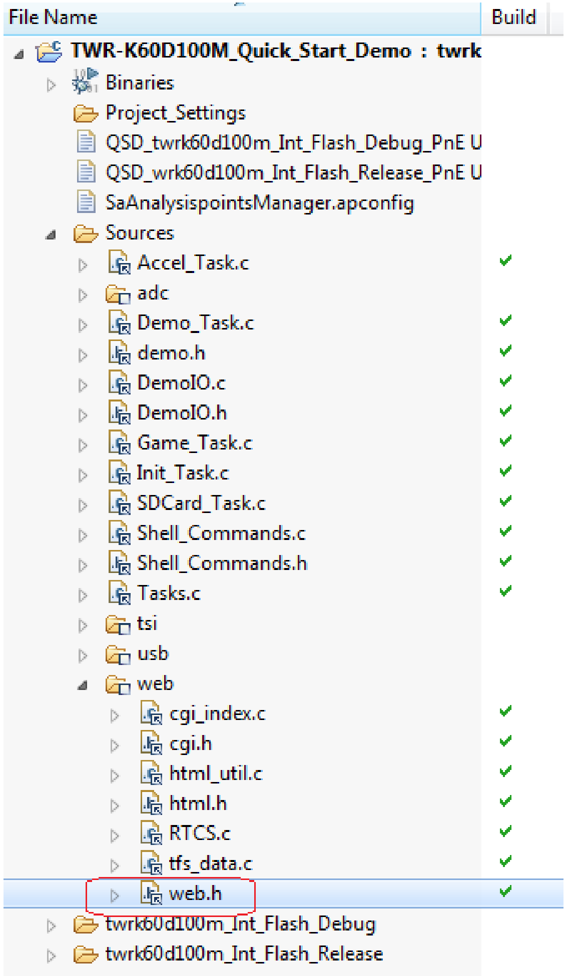
Working sets:

14) Make sure the Debug configuration is set for the project


CodeWarrior Projects
File Name
File Name
Build
TWR-K60D100M_Quick_Start_Demo : twrk
twrk60d100m_Int_Flash_Release
✓ twrk60d100m_Int_Flash_Debug
user_config.h
di

```
1 /*HEADER********
2 *
3 * Copyright (c)
4 * All Rights Res
```

15) 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.



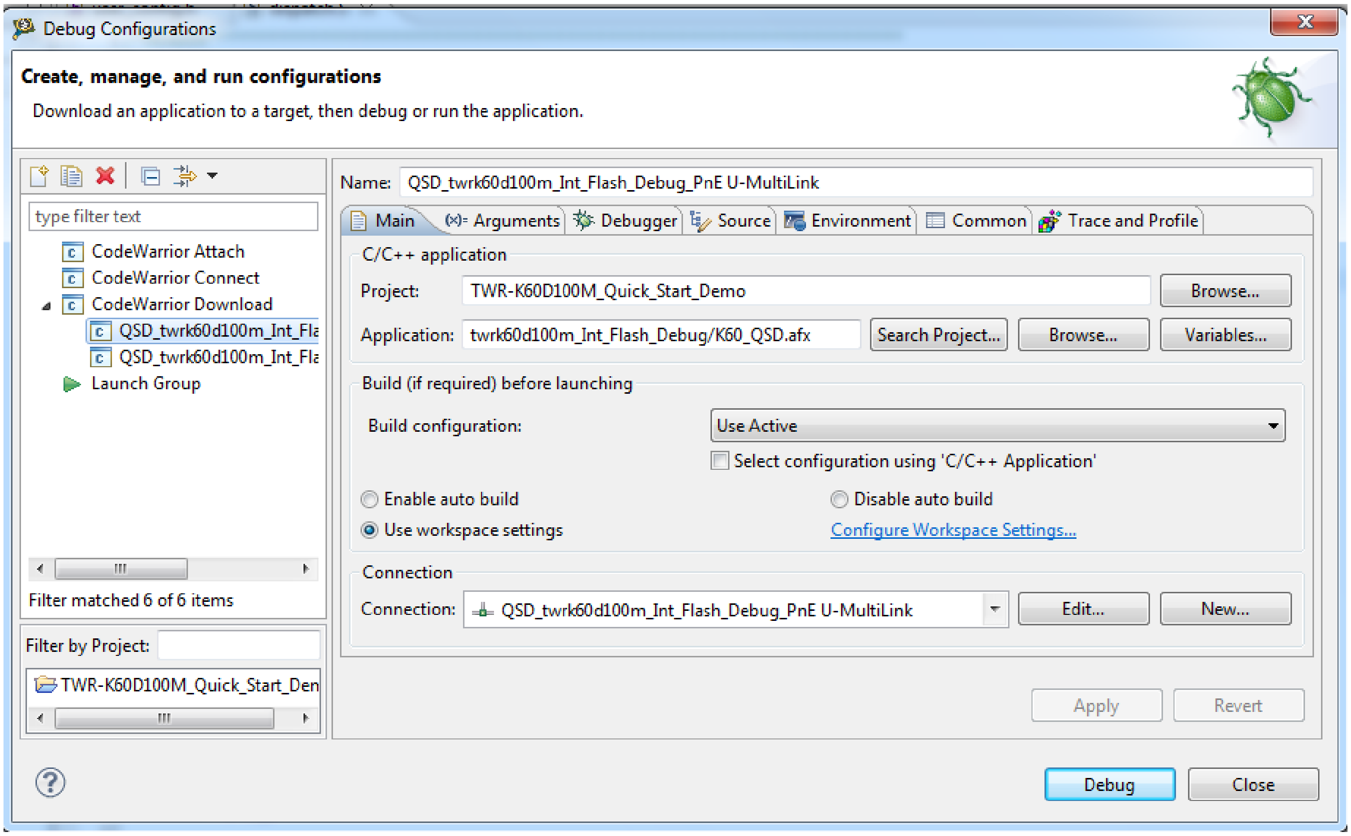
16) You can change the default IP address by modifying the ENET_IPADDR define. For example, to set the target address to 169.254.3.3, and the line will be:
#define ENET_IPADDR IPADDR(169,254,3,3)

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

18) Save the file after you are done.

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

20) Click **Run->Debug Configurations…** in the menu bar, and select the **QSD_twrK60D100M_Int_Flash_Debug_PnE U-MultiLink** configuration. Then hit the debug button in the bottom of the window



Debug Configurations

Create, manage, and run configurations

Download an application to a target, then debug or run the application.

type filter text

- CodeWarrior Attach
- CodeWarrior Connect
- CodeWarrior Download
  - QSD_twrk60d100m_Int_Fla
  - QSD_twrk60d100m_Int_Fla
- Launch Group

Filter matched 6 of 6 items

Filter by Project:

TWR-K60D100M_Quick_Start_Den

Name: QSD_twrk60d100m_Int_Flash_Debug_PnE U-MultiLink

Main | Arguments | Debugger | Source | Environment | Common | Trace and Profile

C/C++ application

Project: TWR-K60D100M_Quick_Start_Demo [Browse...]

Application: twrk60d100m_Int_Flash_Debug/K60_QSD.afx [Search Project...] [Browse...] [Variables...]

Build (if required) before launching

Build configuration: Use Active

□ Select configuration using 'C/C++ Application'

○ Enable auto build ○ Disable auto build

● Use workspace settings Configure Workspace Settings...

Connection

Connection: QSD_twrk60d100m_Int_Flash_Debug_PnE U-MultiLink [Edit...] [New...]

[Apply] [Revert]

[Debug] [Close]

21) If this is the first time you've used CW10.4 with your board, you may get a dialog box asking to update the OSJTAG firmware. Unplug your board, put a jumper on **J19** 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 **J19**, and hit "OK" again on the dialog box. If you do not have a spare jumper, you can temporarily use the one on J9.

22) The code will then be flashed to the board and the debugger started.

23) Once the code is done flashing, the code will pause at the starting instruction of the MQX.

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

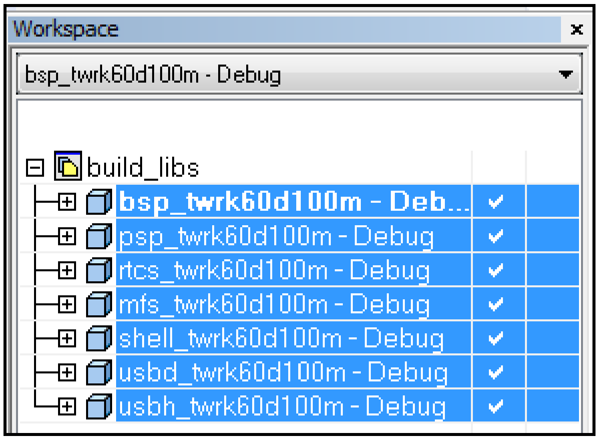
5.2 IAR Embedded Workbench for ARM 6.50.6

1) Follow the directions in Section 3 to update the OSJTAG firmware and drivers. See Section 9 for OSJTAG Troubleshooting.

2) Install IAR for ARM v6.60 or higher. The Evaluation edition or Full edition will be required because of code size.

3) Install MQX 4.0. You can find this on the MQX website

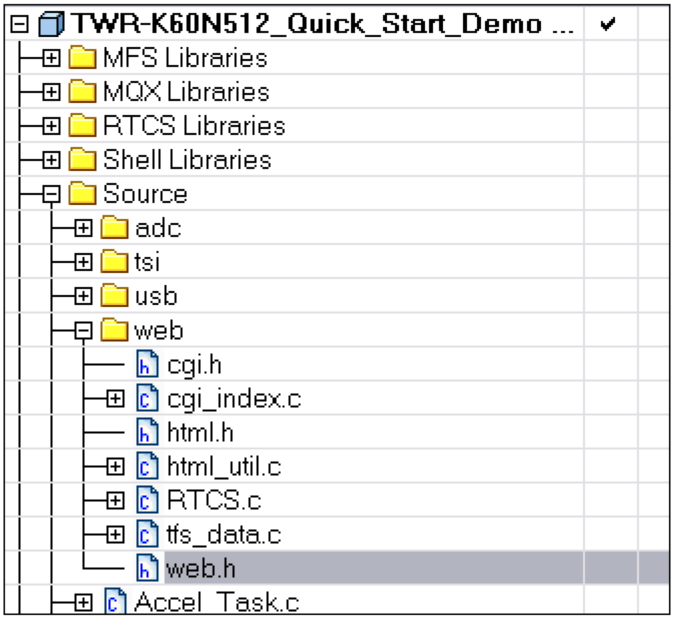
4) You must then compile the MQX libraries for the TWR-K60D100M board. The workspace at **<mqx_dir>\config\twrk60d100m\iar\build_libs.eww** has all the libraries to build. Use the default "Debug" targets and you can select all the libraries by holding the Ctrl key as you click on each project name, and after they are all highlighted, select "Rebuild All".


Workspace
bsp_twrk60d100m - Debug
build_libs
bsp_twrk60d100m - Deb... ✓
psp_twrk60d100m - Debug ✓
rtcs_twrk60d100m - Debug ✓
mfs_twrk60d100m - Debug ✓
shell_twrk60d100m - Debug ✓
usbd_twrk60d100m - Debug ✓
usbh_twrk60d100m - Debug ✓

See the **Getting Started with Freescale MQX™ RTOS** document in the **<mqx_dir>\doc** folder for more information if needed on compiling the libraries before continuing on with the lab.

5) If you did not install the K60 Quick Start Demo to the default location inside of **C:\Freescale\Freescale_MQX_4_0\demo** folder, move the folder **TWR-K60D100M_Quick_Start_Demo** to that location. The path after the move should be **C:\Freescale\Freescale_MQX_4_0\demo\TWR-K60D100M_Quick_Start_Demo**
6) If MQX is installed in another location, move the folder to the **<mqx_dir>\demo** directory at that location instead.

7) Open the workspace file that will be located at **C:\Freescale\Freescale_MQX_4_0\demo\TWR-K60D100M_Quick_Start_Demo\iar\TWR-K60D100M_Quick_Start_Demo.eww**
8) 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.



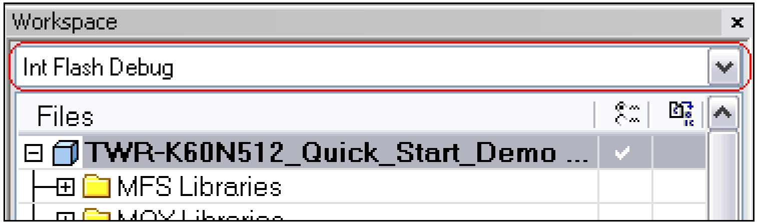
```
TWR-K60N512_Quick_Start_Demo ...  ✓
  MFS Libraries
  MQX Libraries
  RTCS Libraries
  Shell Libraries
  Source
    adc
    tsi
    usb
    web
      cgi.h
      cgi_index.c
      html.h
      html_util.c
      RTCS.c
      tfs_data.c
      web.h
    Accel_Task.c
```

9) 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)

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

11) Save the file after you are done.

12) Select the Int Flash Debug target

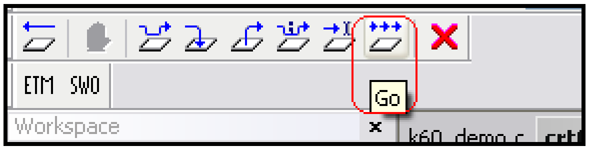


```
Workspace
Int Flash Debug
Files
TWR-K60N512_Quick_Start_Demo ...  ✓
  MFS Libraries
  MQX Libraries
```

13) Compile the project by clicking the Make icon (or right clicking on the project and select "Make").

14) After compilation completes, download the code to the board and start the debugger by pressing the "Download and Debug" button

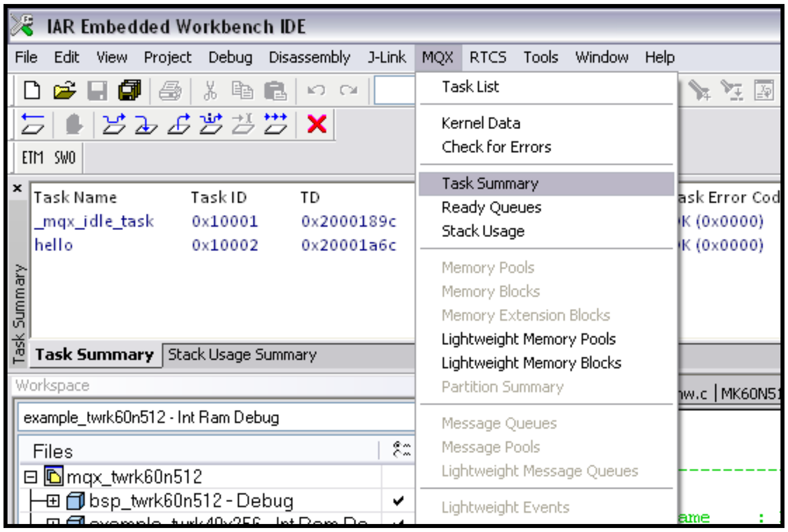
15) The code will download to Flash, and the debugger screen will come up and pause at the first instruction. Hit the "Go" button to start running.



6 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.



Task Name	Task ID	TD	ask Error Cod
_mqx_idle_task	0x10001	0x2000189c	K (0x0000)
hello	0x10002	0x20001a6c	K (0x0000)

7 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>\doc\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 LWGPIO driver is initialized.

Next the Accl_Task is started, which polls the MMA7660 sensor on the TWR-K60D100M 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. This is the same code as in the sdcard MFS example. 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.

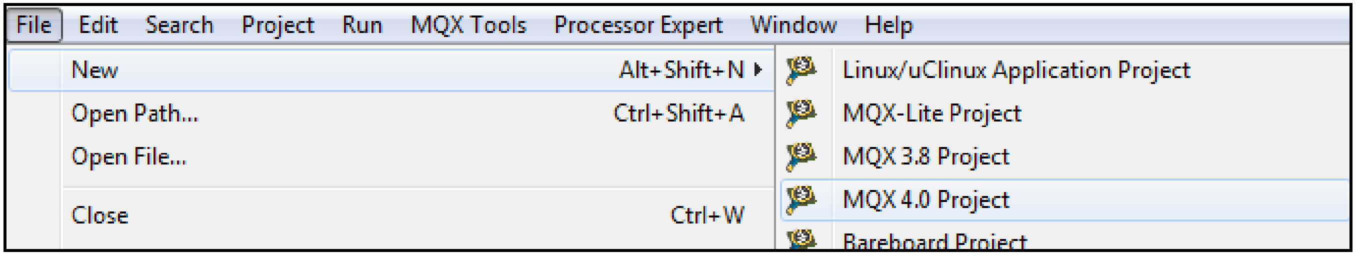
The initialization task then calls _**task_block()** to be permanently blocked, and the demo is now up and ready. See the comments in the source code for more information on how the demo works.

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.

8 Creating a new MQX project

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

1) Go to File->New->MQX 4.0 Project



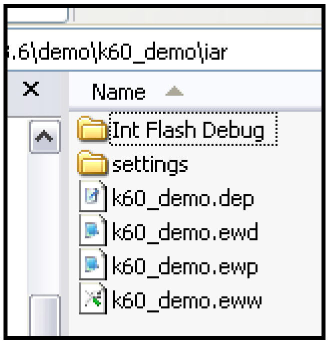
2) Follow the wizard to setup and create a new MQX project.

To create a basic MQX project in IAR via a script, use the following steps:

1) Make sure you have not modified the files in the K60 Quick Start Demo
2) Double click on the **k60d_make_new_iar_mqx_project.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 the Enter key. The script that is now running copies the project files from the K60 Quick Start Demo into a new directory in **<mqx_dir>\demo**, modifies them to use the new name, and copies the Hello World example from **<mqx_dir>\mqx\examples\hello**
5) After the script finishes, hit the Enter key to close the window.

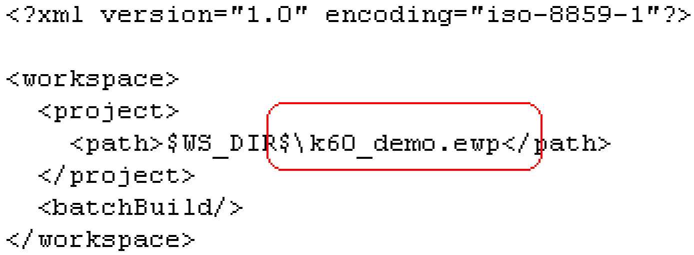
To copy the Quick Start Demo project in IAR manually, use the following steps:

1. Copy the **TWR-K60D100M_Quick_Start_Demo** folder and place the copy in the MQX demo folder at **C:\Freescale\Freescale_MQX_4_0\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



Int Flash Debug
settings
k60_demo.dep
k60_demo.ewd
k60_demo.ewp
k60_demo.eww

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>
```

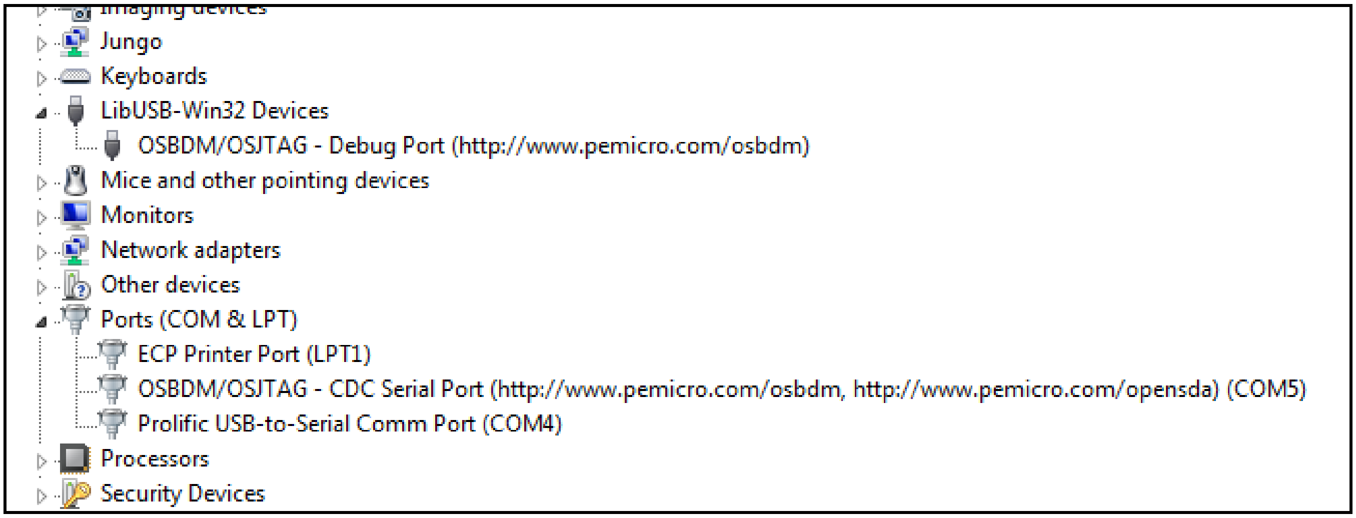
6. Save the file
7. Open the k60_demo.eww workspace

9 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 Section 3 or the **Installation and Operation** document on the P&E website 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.

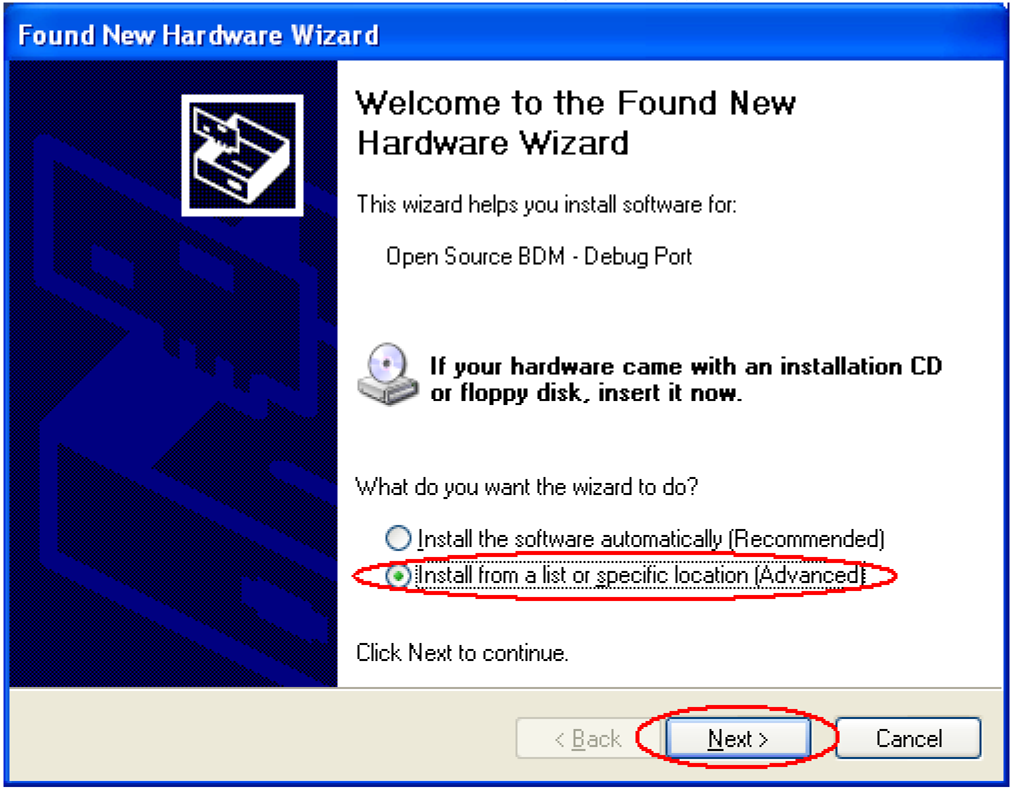
When the tower board is plugged in, it should enumerate as a composite device, with one driver for debugging, and the other as a serial port. If you go to the Device Manager you should see the following:



- Imaging devices
- Jungo
- Keyboards
- LibUSB-Win32 Devices
  - OSBDM/OSJTAG - Debug Port (http://www.pemicro.com/osbdm)
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
- Ports (COM & LPT)
  - ECP Printer Port (LPT1)
  - OSBDM/OSJTAG - CDC Serial Port (http://www.pemicro.com/osbdm, http://www.pemicro.com/opensda) (COM5)
  - Prolific USB-to-Serial Comm Port (COM4)
- Processors
- Security Devices

If you only see it enumerate as the OSBDM/OSJTAG 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.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

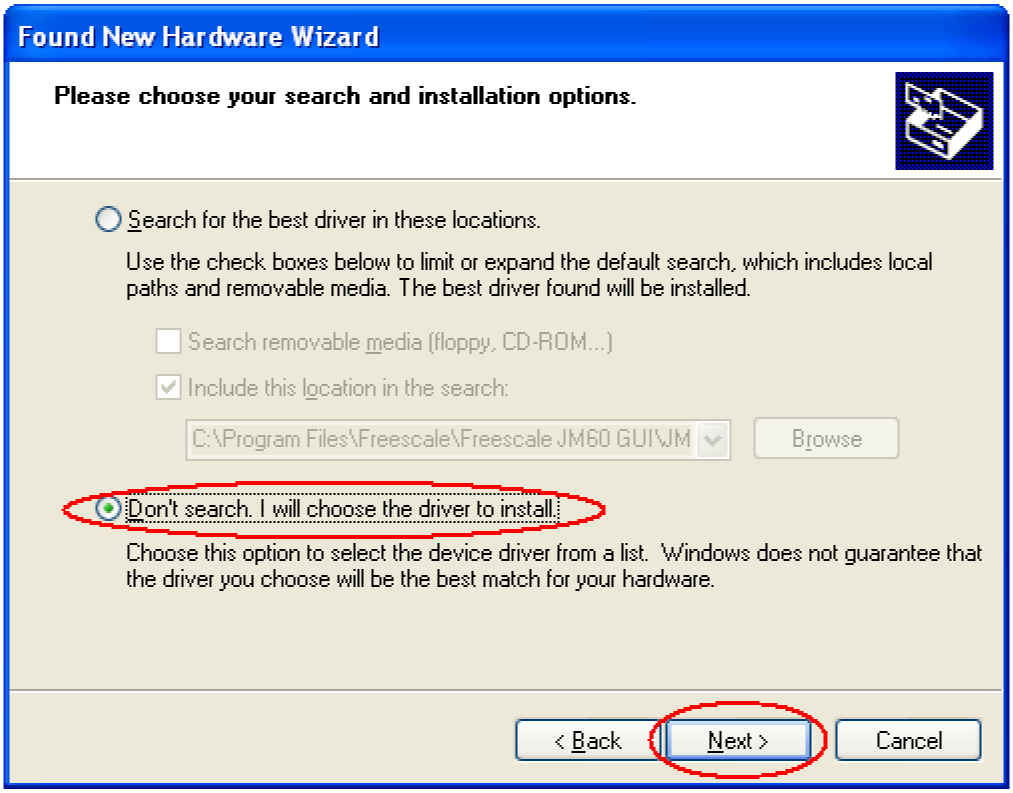
Open Source BDM - Debug Port

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)
● Install from a list or specific location (Advanced)

Click Next to continue.



Found New Hardware Wizard

Please choose your search and installation options.

○ Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

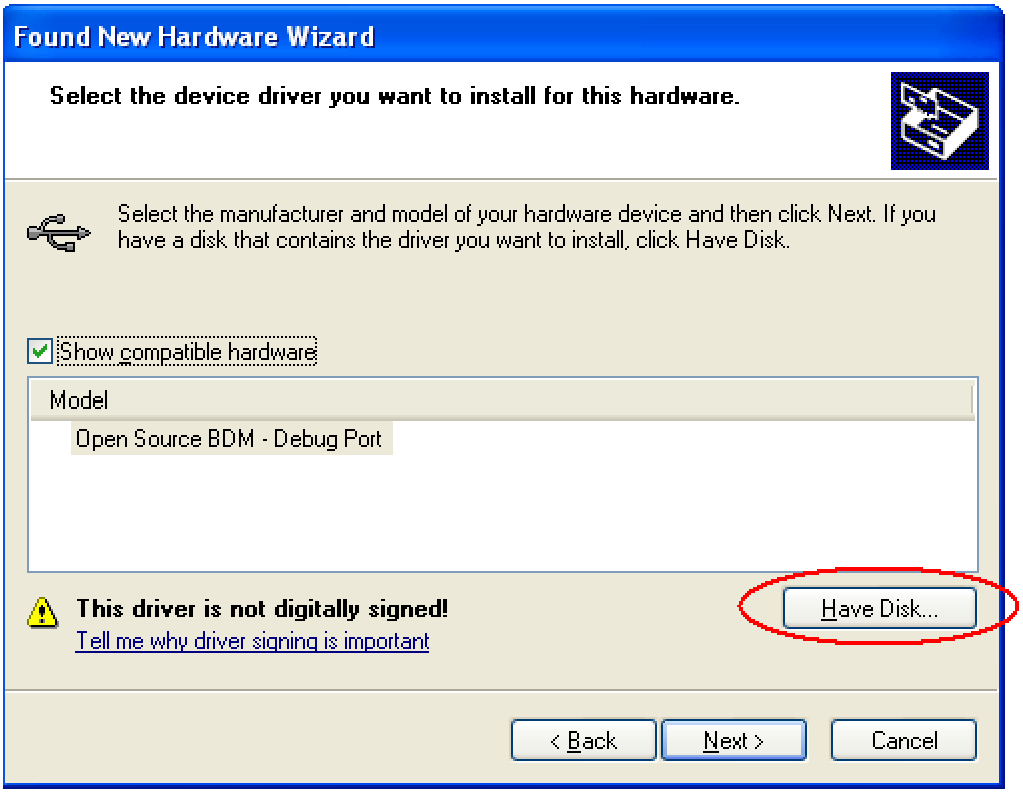
□ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

C:\Program Files\Freescale\Freescale JM60 GUI\JM

● Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.



Found New Hardware Wizard

Select the device driver you want to install for this hardware.

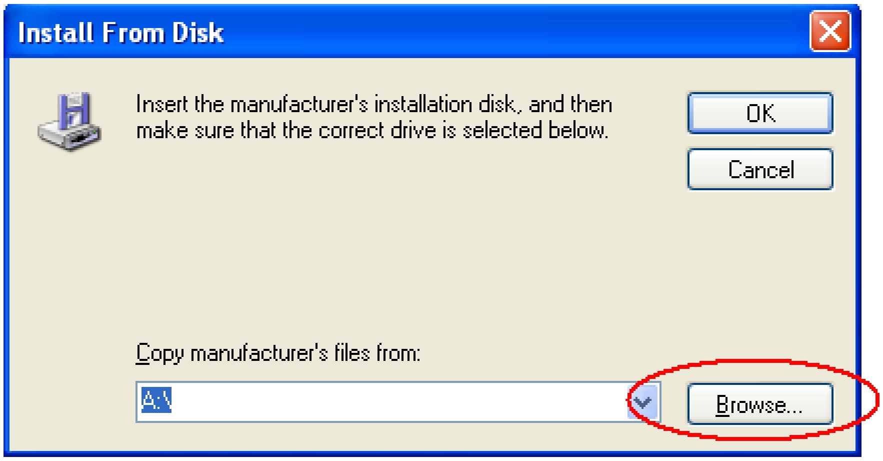
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

☑ Show compatible hardware

Model
Open Source BDM - Debug Port

This driver is not digitally signed!

Tell me why driver signing is important


Install From Disk

Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from:

A:\

- For the Open Source BDM – Debug Port, use the driver at:
  C:\PEMicro\osbdm_osjtag_virtual_serial_toolkit\Drivers\osbdm\OSJTAG_Debug_Interface_libusb.inf

- For the PEMicro USB Serial Port (i1), use the driver at:
  C:\PEMicro\osbdm_osjtag_virtual_serial_toolkit\Drivers\osbdmOSJTAG_Serial_Interface_windriver_version.inf