

TWR-K70F120M Quick Start Demo

Lab Guide for MQX 3.8

Rev. 0.2

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 IAR Embedded Workbench for ARM 6.30	7
6 Task Aware Debugging.....	10
7 Demo Software Overview	10
8 Creating a new MQX project	11
9 OSJTAG.....	12

1 Purpose

This lab document describes how to use the TWR-K70F120M 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-K70F120M board in stand-alone mode:

- 1) Plug-in the battery into the battery socket on the underside of the board
- 2) Change **J17** to pins 2-3 to connect the VBAT domain to run off the battery

If using the TWR-K70F120M 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-K70F120M board:
 - a. Remove **J19** 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-K70F120M 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-K70F120M 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 (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.

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.



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 **J10** 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-K70F120M 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

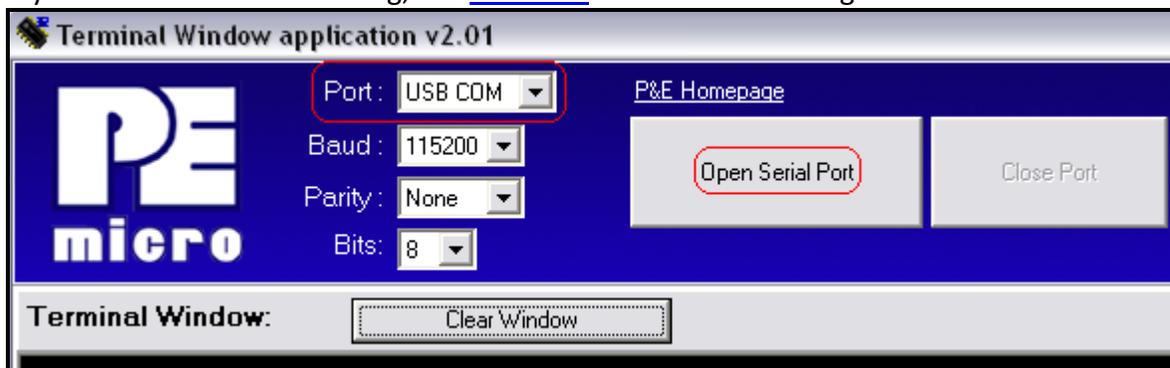
- 4) Press SW2 to play a memory game using the Touch pads E1-E4. A sequence will light up, and then press the touch sensors in the order flashed. If an incorrect sequence is input or you take too long, then all the lights will blink rapidly and the game will reset. See how far you can go!
- 5) Press 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 the P&E Terminal Utility by clicking on Start Menu->Programs->P&E Kinetis Tower Toolkit->Utilities->Terminal Utility
- 9) Make sure USB COM is selected with 115200 baud, and click on the "Open Serial Port" button. If you have trouble connecting, see [Section 9](#) for Troubleshooting.



- 10) 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.
- 11) Type "settime HH:MM" to adjust the time to HH:MM.

12) 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

13) 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:

14) Plug in a USB mini-B cable into the USB mini-B connector on the TWR-SER.
 15) 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:

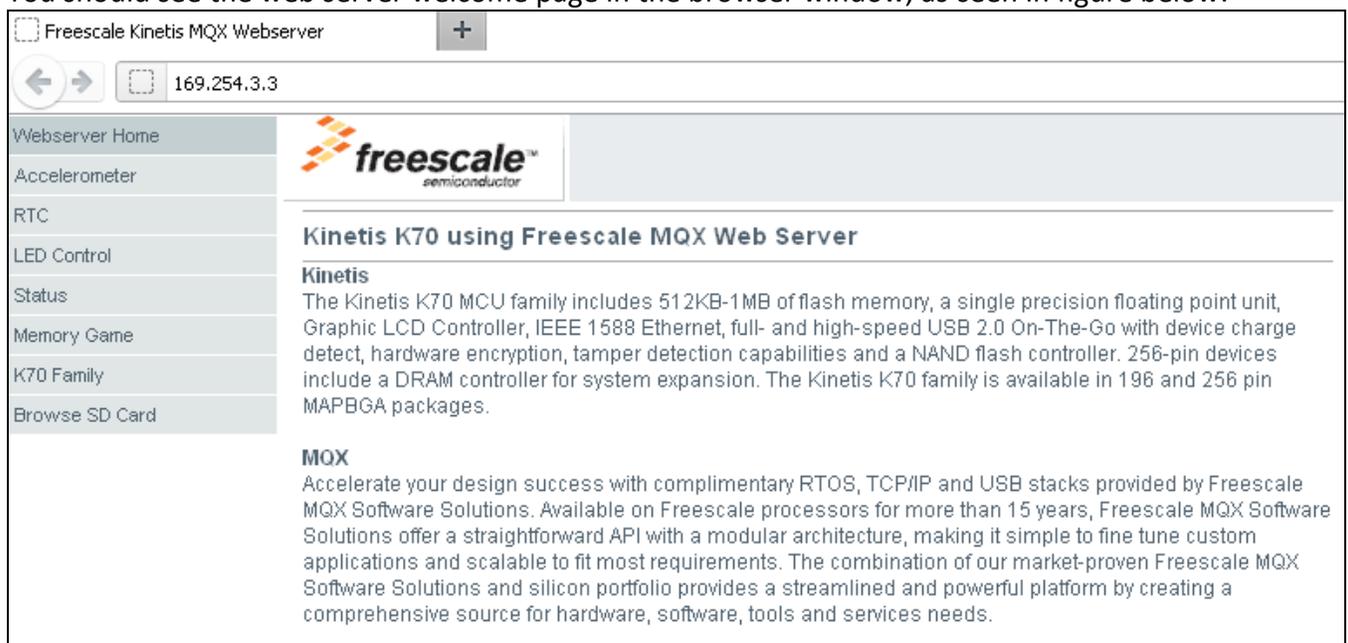
16) Finally we will run the web browser. The default IP address of the board is 169.254.3.3.
 Typically, when you connect your computer directly to the board, the computer will default to an auto IP address on the same subnet as the board (169.254.x.x), therefore requiring no setup.

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.

17) 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**.

18) Open a web browser and go to the target device address. In this case, **169.254.3.3**

19) You should see the web server welcome page in the browser window, as seen in figure below:



20) 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 K70 family

21) 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**

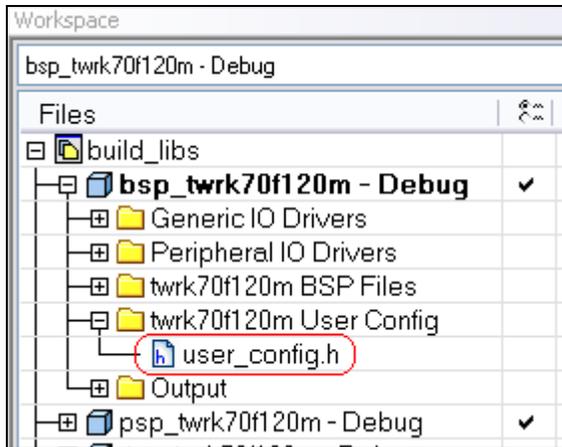
5 Development Software and Programming the Board

The following instructions describe how to build and flash the Kinetis Quick Start Demo using MQX 3.8 with IAR 6.30 (instructions for CW10.2 will be released after official MQX support for CW10.2 becomes available).

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.

5.1 IAR Embedded Workbench for ARM 6.30

- 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 v6.30 or higher. The Evaluation edition or Full edition will be required because of code size.
- 3) Install MQX 3.8. You can find this on the [MQX website](#)
- 4) You will need to enable the ESHDC and I2C0 drivers in the BSP library to use this demo.
- 5) Open the workspace at **C:\Program Files\Freescale\Freescale MQX 3.8\config\twrK70F120M\iar\build_libs.eww**
- 6) Open the **user_config.h** file in the **bsp_twrK70F120M – Debug** project

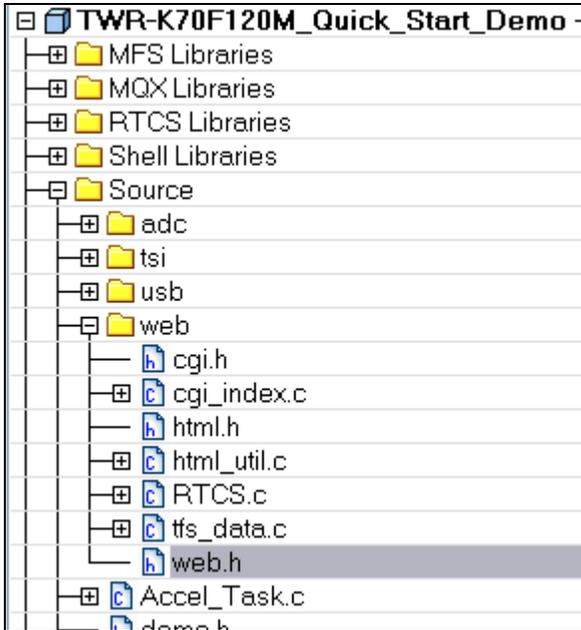


7) Change

```
#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 K70 BSP library. This makes the edits in the `user_config.h` file take effect. Make sure **bsp_twrk70f120m – Debug** project is bolded
- 10) If you did not install the MQX project in the default **C:\Program Files\Freescale\Freescale MQX 3.8** 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-K70F120M Quick Start Demo Lab from the [TWR-K70F120M website](#). This document is part of that package, so you likely will have already done this.
- 12) Unzip the **TWRK70F120MQSDLAB.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 3.8\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 3.8\demo\TWR-K70F120M_Quick_Start_Demo\iar\TWR-K70F120M_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.



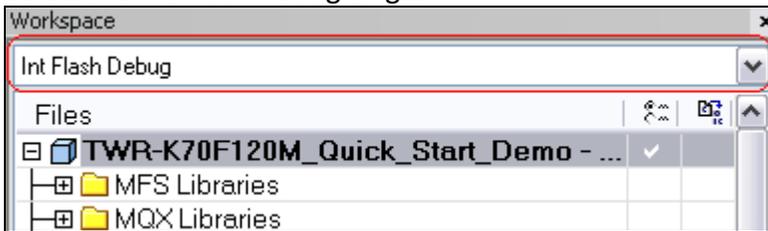
15) You can change the default IP address by modifying the ENET_IPADDR define. For example, to set the target address to 169.254.3.3, and the line will be:

```
#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 

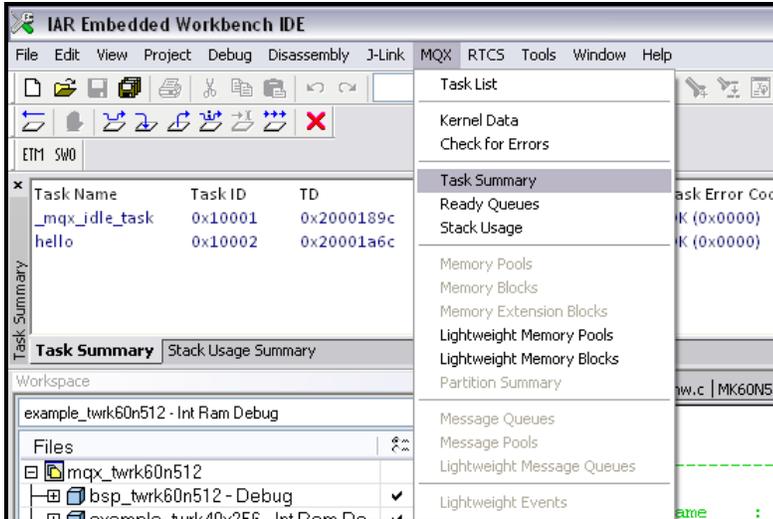
21) The code will download Flash, and the debugger screen will come up and pause at the first instruction. Hit the "Go" button to start running.



6 Task Aware Debugging

MQX comes with Task Aware Debugging (TAD) when using both CodeWarrrior 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 CodeWarrrior) 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.



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>\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-K70F120M 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.

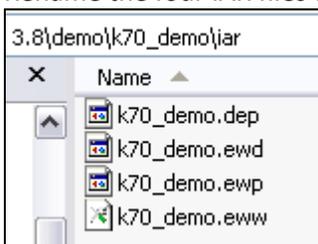
8 Creating a new MQX project

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

- 1) Make sure you have not modified the files in the K70 Quick Start Demo
- 2) Double click on the `create_new_k70_mqx_project_iar.exe` file inside the K70 Quick Start Demo.
- 3) Type in the new project's name, with no spaces, such as `K70_hello`
- 4) Hit Enter and wait for it to finish creating the new project. The script will copy the project files from the K70 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-K70F120M_Quick_Start_Demo` folder and place the copy in the MQX demo folder at `C:\Program Files\Freescale\Freescale MQX 3.8\demo`
2. Rename the folder to the new project name, with no spaces. For example, `k70_demo`
3. Inside the `k70_demo` folder, go into the `iar` folder
4. Rename the four IAR files to `k70_demo`



5. Open the k70_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\k70_demo.eww</path>
  </project>
  <batchbuild/>
</workspace>
```

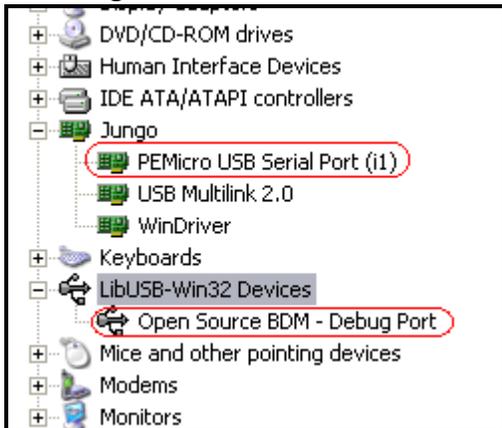
6. Save the file
7. Open the k70_demo.eww workspace

9 OSJTAG

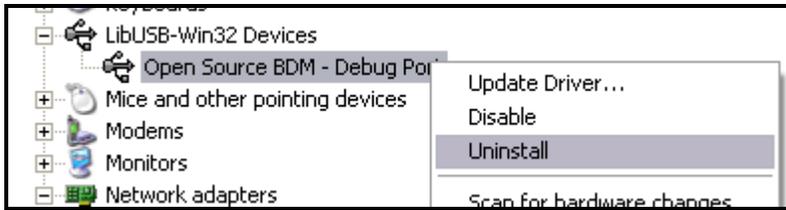
Open Source JTAG (also known as OSBDM on ColdFire tower boards) allows a user to program, debug, and get serial data from Kinetis devices via a USB cable. The firmware runs on a Freescale MCFS08JM60 on the underside of the Kinetis tower board.

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.

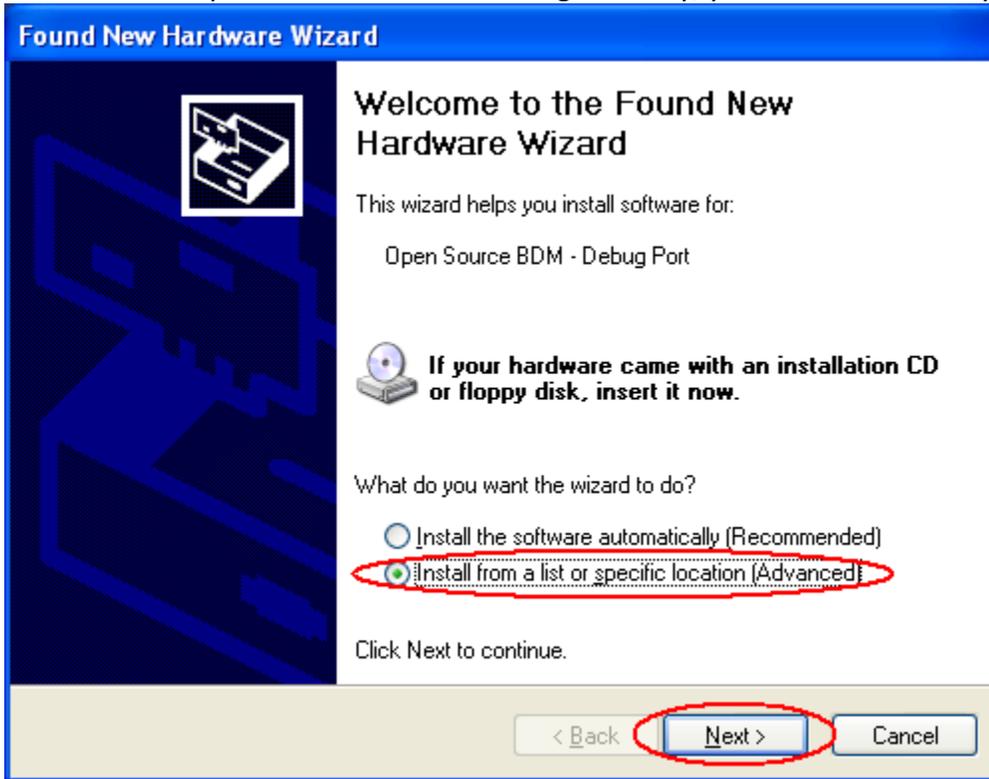
When the tower board is plugged in, it should enumerate as a composite device, with one driver for debugging, and the other as a serial port. If you go to the Device Manager you should see the following:



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.



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 Browse

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.

< Back Next > Cancel

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.

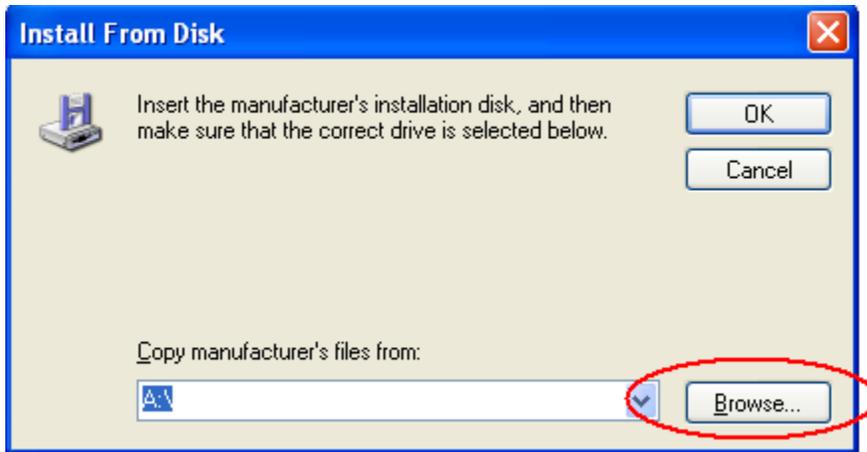
Show compatible hardware

Model
Open Source BDM - Debug Port

⚠ This driver is not digitally signed!
[Tell me why driver signing is important](#)

Have Disk...

< Back Next > Cancel



- 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