

K20 50MHz Bare Metal Examples

Lab Guide for Example Code Projects

Rev. 0.2

Contents

1 Purpose.....	3
2 Directory Structure	3
3 Toolchain Support	3
3.1 IAR Embedded Workbench	3
3.2 CodeWarrior 10.x	4
4 Examples	4
5 Update OSJTAG	4
6 Configure Hardware.....	5
7 Launch the Terminal.....	6
8 Development Software and Programming the Board.....	6
8.1 CodeWarrior for Microcontrollers 10.x	6
8.2 IAR Embedded Workbench for ARM 6.30	10
9 Creating New Projects	10
10 OSJTAG	11
11 Codewarrior Updates.....	13

1 Purpose

This lab document describes how to use the K20 50 MHz bare metal example projects. Included demos are pre-configured to run on TWR-K20D50M board. These are general purpose demos that could be ported to run on any board that uses a Kinetis K20 50 MHz processor as long as the particular device supports the required features for the demo.

Some demos might require additional tower module components. Each project includes a readme.txt file in the project directory that includes information on the demo project including configuration directions that are specific to that particular demo.

2 Directory Structure

The directory structure for the bare metal examples has three main folders—build, src and util. The build folder contains all development toolchain specific files. All of the source code files are in the src directory. The util folder includes compiler patches referenced in this document.

The source tree is broken up as follows:

- src\common\ - Common utilities such as compiler startup are provided in this directory
- src\cpu\ - CPU specific initialization and header files here
- src\drivers\ - Drivers for some of the various peripherals are provided here.
- src\platforms\ - Each supported platform has a header file that defines board specific information, such as the input clock frequency used for that board.
- src\projects\ - This directory holds all the individual example project source code
- src\semlib\ - Utilities to handle native compiler support for printf and scanf

3 Toolchain Support

Currently the IAR 6.30 and CW10.1 tool chains are supported.

3.1 IAR Embedded Workbench

IAR workspace, project, linker, and support files are provided in kinetis-sc\build\iar. Each example has its own directory and within that directory is a workspace file (.eww) that will load the supported projects (usually one project for each supported hardware platform). Each project contains several configuration options that can be selected using a drop menu. The configuration support different link targets for different Kinetis memory configurations (e.g. RAM_16KB or FLASH_128KB_PFLASH).

3.2 CodeWarrior 10.x

CodeWarrior 10.x project files can be found in the kinetis-sc\build\cw folder. Again, each example has its own directory.

4 Examples

There are many example projects that highlight the operation of different modules. The "hello_world" demo is the simplest example, and this project is the baseline used for developing other examples. The hello_world project will perform basic initialization for the board and then display the device configuration information on the terminal (default baud rate is 115200).

Each example includes a readme.txt file in the workspace or project directory that gives a description of what the project does and describes any configuration needed to use it.

5 Update OSJTAG

The TWR-K20D50M includes the OSJTAG circuit. By default all of the demo projects are setup to use OSJTAG to download and debug code. The OSJTAG circuit is also used as the primary serial communication port.

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.

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

P&E micro 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 its 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)**
Refresh list of devices

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 **J34**. 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.

6 Configure Hardware

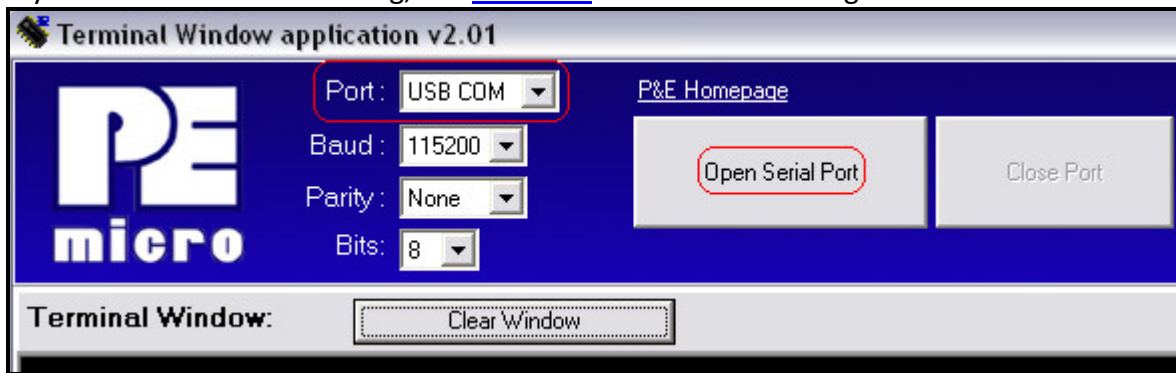
The hello world example can be run using a TWR-K20D50M stand-alone. No other tower boards/components are required for this example.

- 1) Note that the default jumper settings are in the Quick Start Guide for the TWR-K20D50M. If you want to change back to the default settings after you are done running the demo.

- 2) Connect a mini-B USB cable between the TWR board and the USB port on your computer.
- 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 9](#) for troubleshooting and make sure you completed the steps listed in [Section 3](#).

7 Launch the Terminal

- 1) Open the P&E Terminal Utility by clicking on Start Menu->Programs->P&E Kinetis Tower Toolkit->Utilities->Terminal Utility
- 2) 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.

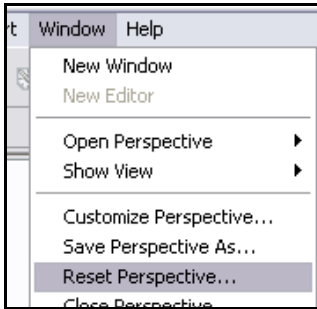


8 Development Software and Programming the Board

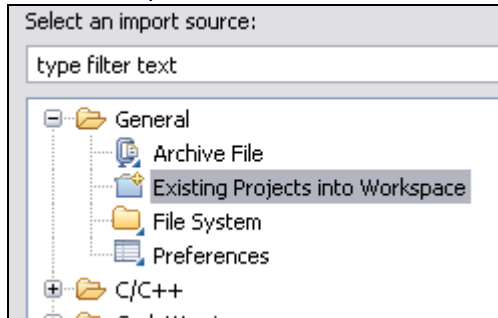
The following instructions describe how to build and flash the hello world demo using CodeWarrior MCU 10.x or IAR 6.30.

8.1 CodeWarrior for Microcontrollers 10.x

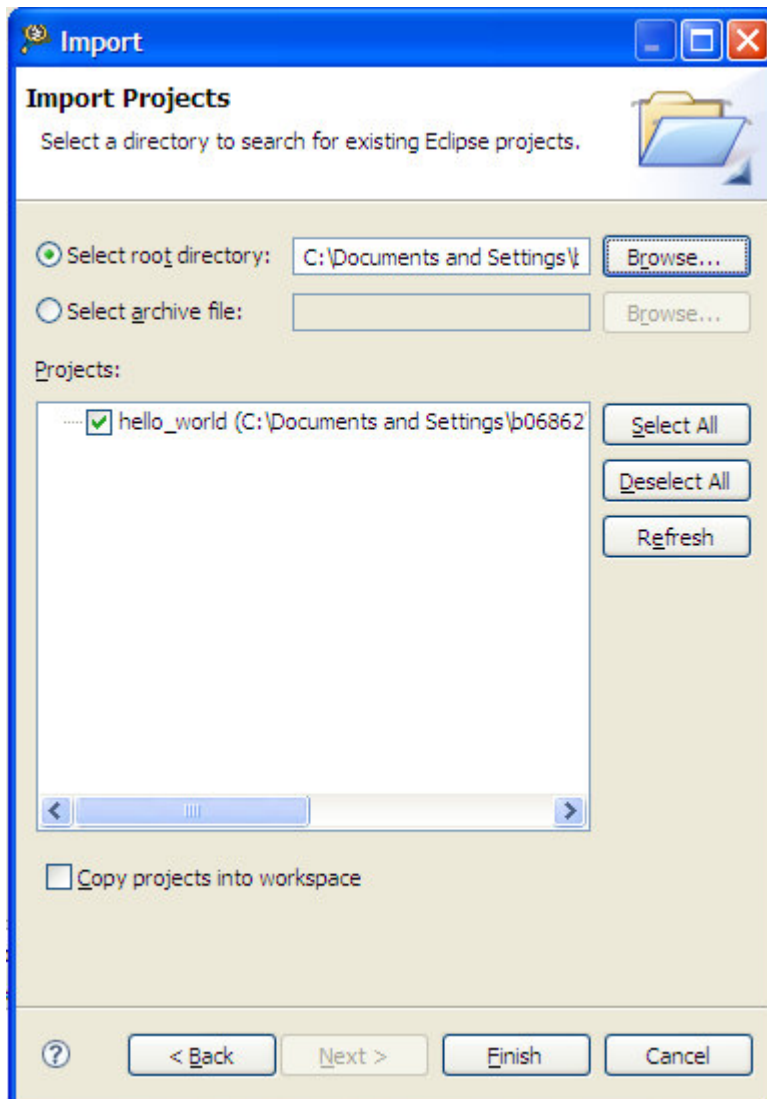
- 1) Follow the directions in [Section 3](#) to update the OSJTAG firmware and drivers. See [Section 9](#) for OSJTAG Troubleshooting.
- 2) Install CodeWarrior for Microcontrollers 10.x.
- 3) Make sure you have the latest Codewarrior updates and patches. Go to [Section 11](#).
- 4) Open CW10. At the welcome screen, set the workspace to the "kinetis-sc" folder location. These projects use workspace relative paths, so in order for the project to find all files correctly the workspace must be set to the kinetis-sc folder (the one that contains the build and src directories).
- 5) If you already have CW10 open, you can change the workspace by going to **File->Switch Workspace**
- 6) 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.
- 7) The workbench view will open up. To ensure all the windows are properly set, go to Window->Reset Perspective



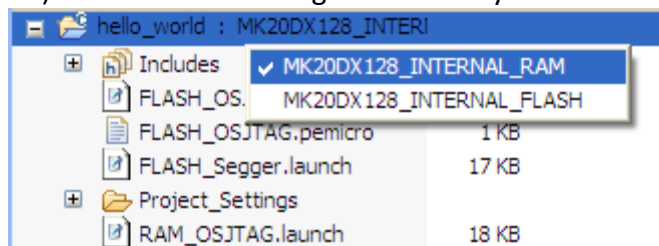
- 8) Click on **File->Import** in the menu bar. In the dialog box that comes up, select “Existing Projects into Workspace” under the General folder. Then click Next



- 9) On the next screen, select the “Select root directory:” option, and click on Browse
10) Navigate to the `..\kinetis-sc\build\cw\hello_world` directory and hit OK.
11) Select the projects to import. Then select Finish.



12) Select the link configuration that you would like to build.

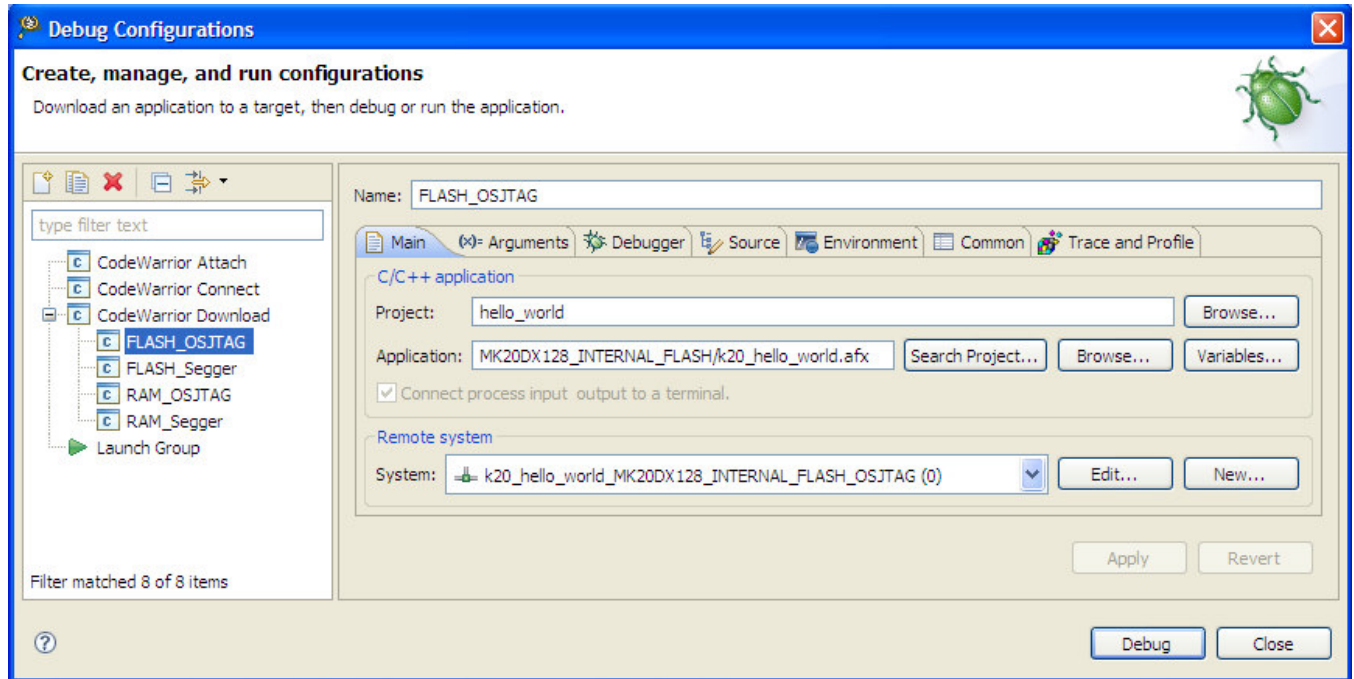


13) Build the project by clicking on the Hammer icon in the toolbar

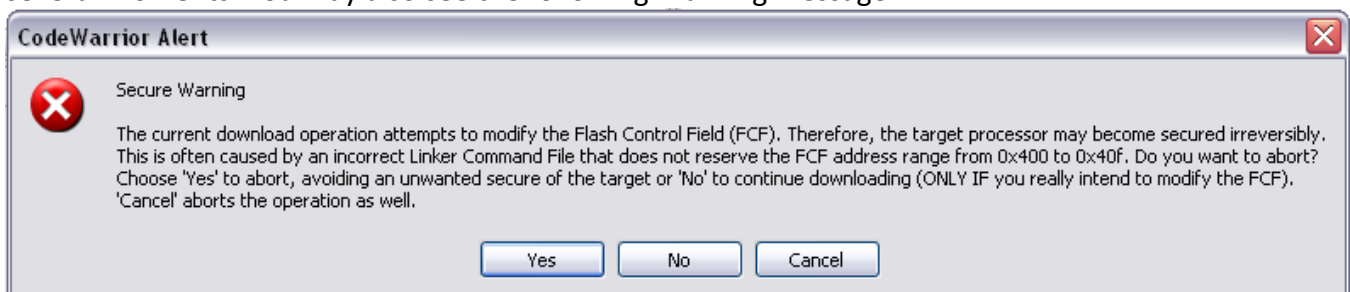


14) Click **Run->Debug Configurations...** in the menu bar, and select the debug configuration appropriate for the project, build configuration, and hardware connection you are using. For example, if you are building the hello_world project to run from the internal RAM and using


OSJTAG as your debugger, then you would select the **FLASH_OSJTAG** configuration. Then hit the debug button in the bottom of the window



- 15) If this is the first time you've used CW10.x with your board, you may get a dialog box asking to update the OSJTAG firmware. Unplug your board, put a jumper on **J34 (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 **J34**, and hit "OK" again on the dialog box. If you do not have a spare jumper, you can temporarily use the one on J7.
- 16) The code will then be downloaded to the board and the debugger started. This may take a several moments. You may also see the following Warning message:

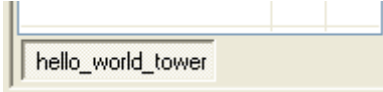


This is a check introduced with a CodeWarrior compiler patch. Hit **No** to continue flashing the part, as the sample code is configured to ensure the part remains unsecured.

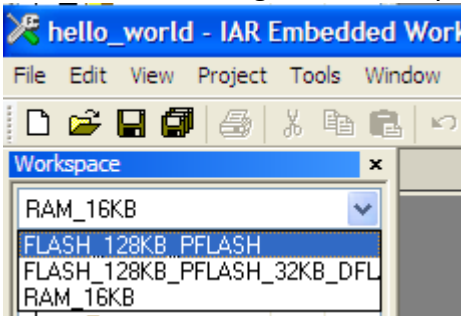
- 17) Once the code is done flashing, the code will pause at the start of the main() function.
- 18) Hit the run icon to continue the program execution. 
- 19) On the terminal you should see the hello world message.



8.2 IAR Embedded Workbench for ARM 6.30

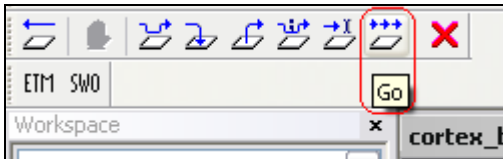
- 1) Follow the directions in [Section 3](#) to update the OSJTAG firmware and drivers. See [Section 9](#) for OSJTAG Troubleshooting.
- 2) Install IAR for ARM v6.30 or higher.
- 3) Open the workspace at `..\kinetis-sc\build\iar\hello_world\hello_world.eww`
- 4) Unzip `utils\iar_flash_p0_8kb_ram.zip` and follow instructions stated in `readme.txt`.
- 5) The project for the board is selected



- 6) Select the link configuration that you want to build.



- 7) Compile the project by clicking the Make icon  (or right clicking on the project and select "Make").
- 8) After compilation completes, download the code to the board and start the debugger by pressing the "Download and Debug" button 
- 9) The code will download, and the debugger screen will come up and pause at the first instruction. Hit the "Go" button to start running.



- 20) On the terminal you should see the hello world message.

9 Creating New Projects

The `kinetis-sc\build\iar\make_new_project.exe` or `kinetis-sc\build\cw\make_new_cw_project.exe` files can be used to clone the `hello_world` project for the appropriate toolchain. The script will prompt you for a name to use for the new project, and then create copies of all needed files and folders.

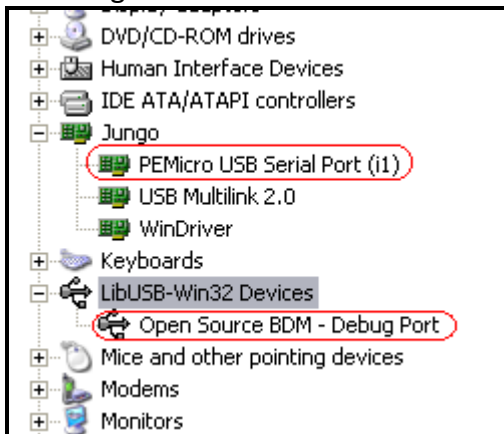
Note `kinetis-sc\build\cw\make_new_cw_project.exe` doesn't clone the `src\hello_world` folder. This must be manually created. The `kinetis-sc\build\iar\make_new_project.exe` doesn't need to create any additional folder.

10 OSJTAG

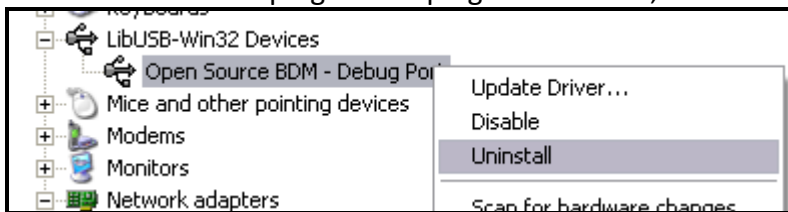
Open Source JTAG (also known as OSBDM on ColdFire tower boards) allows a user to program, debug, and get serial data from Kinetis devices via a USB cable. The firmware runs on a Freescale MCFS08JM60 on the underside of the Kinetis tower board.

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.

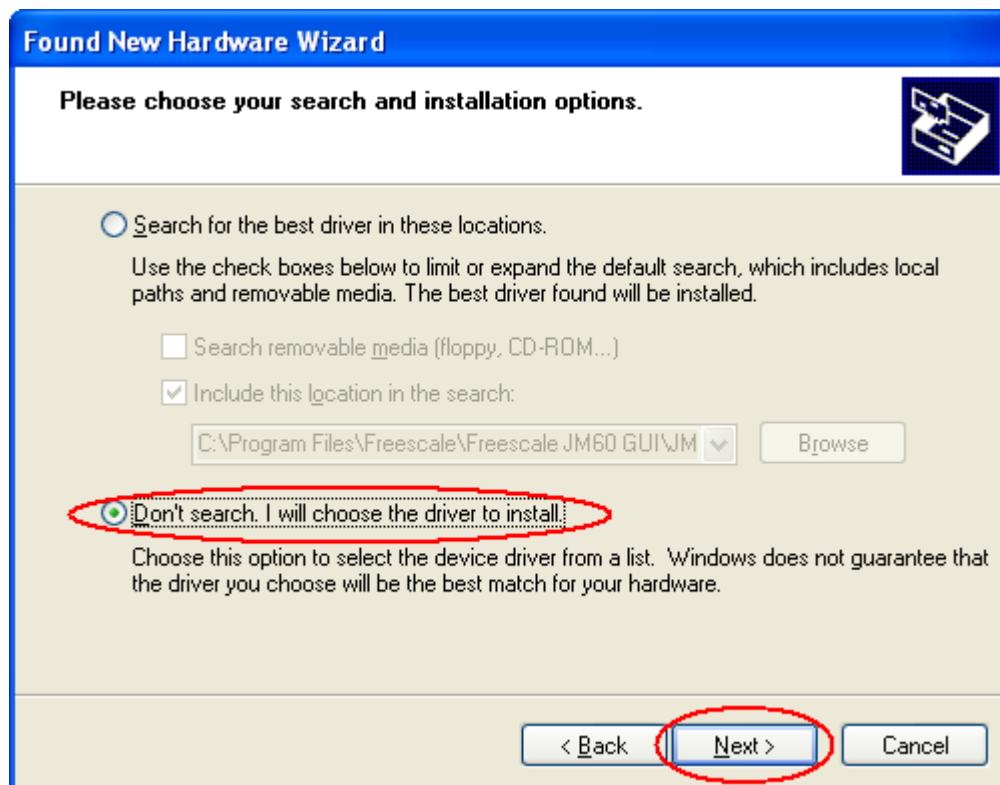
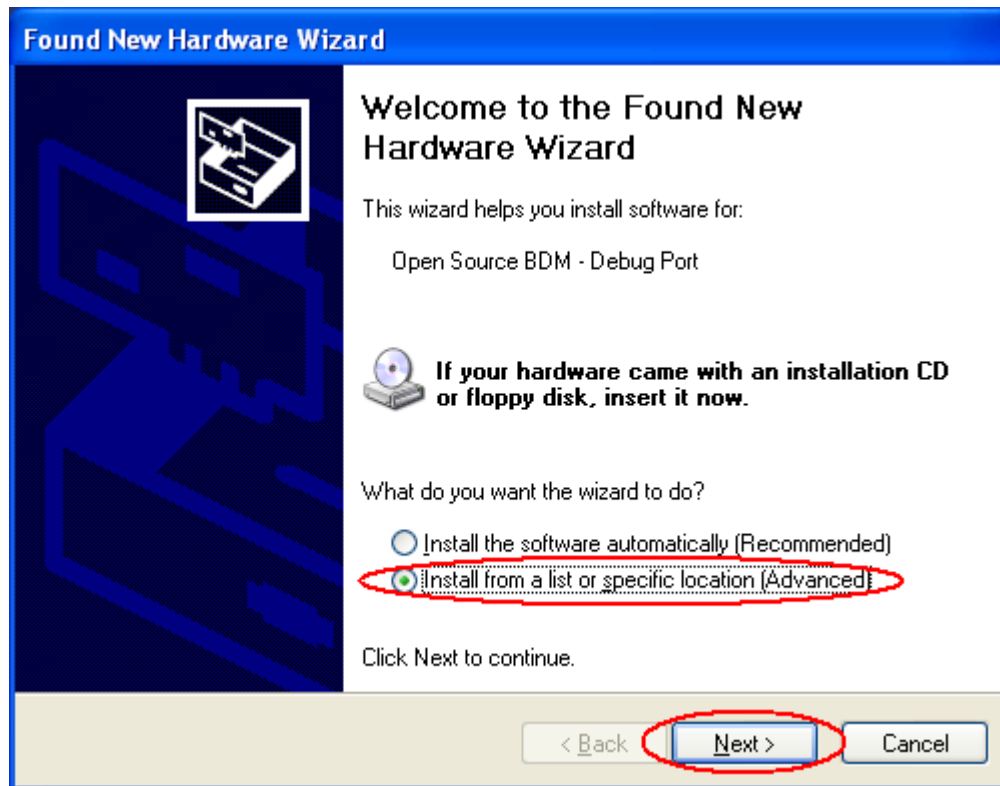
When the tower board is plugged in, it should enumerate as a composite device, with one driver for debugging, and the other as a serial port. If you go to the Device Manager you should see the following:

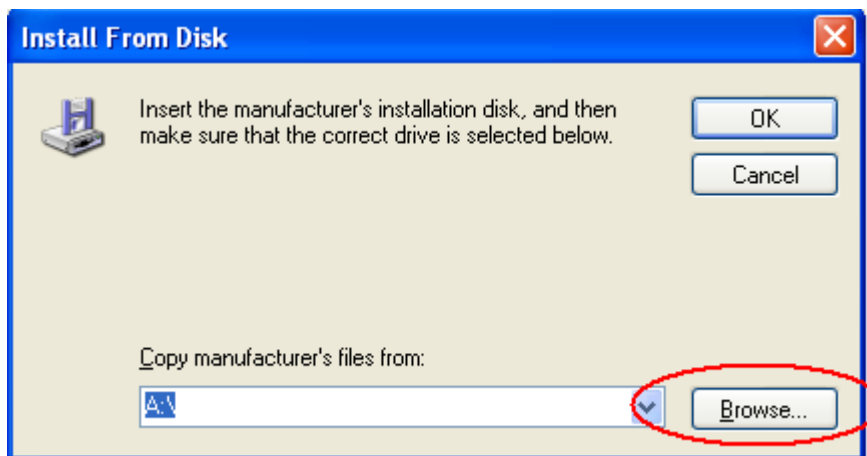
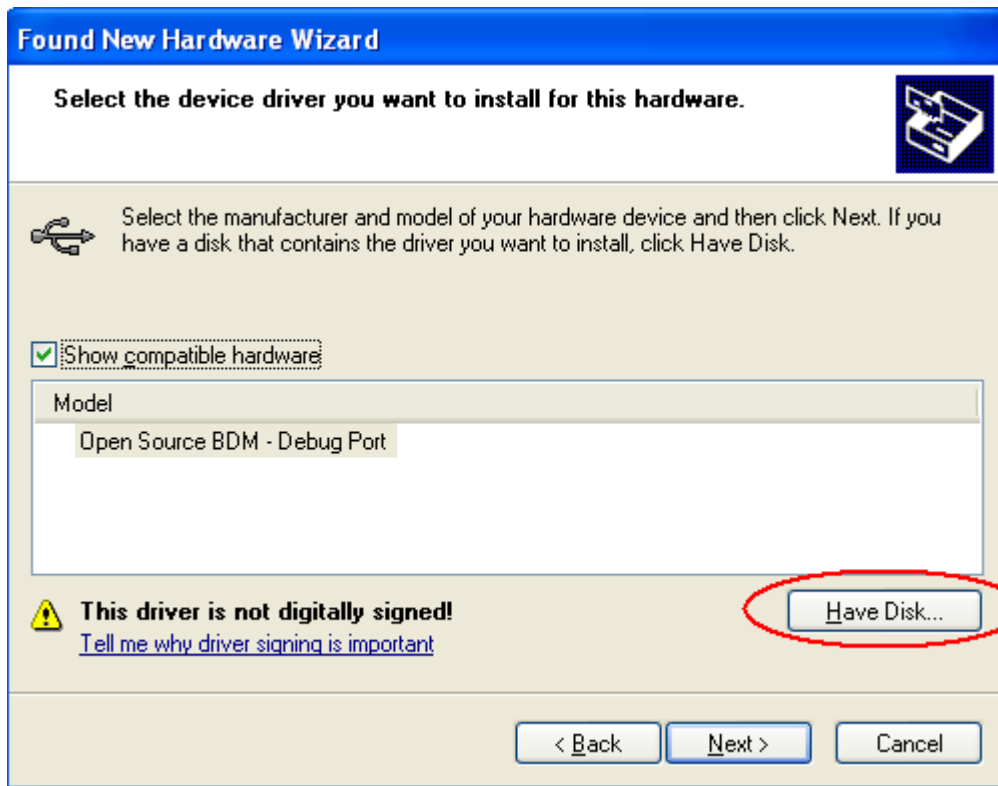


If you only see it enumerate as the Open Source BDM Debug Port, then your computer may automatically picking up an outdated driver. To fix this, right click on the OSBDM driver and select “Uninstall”. Then unplug and re-plug in the board, and it should enumerate correctly.



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





- 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`

11 Codewarrior Updates

- 1) For Codewarrior v10.1, the TWR-K20D50M example codes require the latest updates. Go to “Help” and “Install New Software”. Select the drop off menu and select “FSL MCU Eclipse Update Site”. Install the following two updates as shown in the next figure.

Work with: FSL MCU Eclipse Update Site - http://freescale.com/igfiles/updates/Eclipse/MCU10_1/com.freescale.mcu.update.site

type filter text

Name	Version
<input type="checkbox"/> CodeWarrior ARM Debugger	10.1.1.FSL_b0982
<input type="checkbox"/> CodeWarrior ARM Debugger	10.1.1.FSL_b0982
<input type="checkbox"/> CodeWarrior ColdFire Debugger	10.1.1.FSL_b5481
<input type="checkbox"/> CodeWarrior Core Debugger	7.15.1.FSL_b8726
<input type="checkbox"/> CodeWarrior Core Debugger NLS	7.15.1.FSL_b8726
<input checked="" type="checkbox"/> CodeWarrior Core IDE	1.0.1
<input type="checkbox"/> CodeWarrior HC(S)08 Debugger	10.1.1.FSL_b5491
<input type="checkbox"/> CodeWarrior Kernel Awareness SDK	7.15.0.FSL_b8910
<input type="checkbox"/> CodeWarrior Power Architecture Debugger	10.0.1.FSL_b2241
<input type="checkbox"/> CodeWarrior RS08 Debugger	10.1.1.FSL_b5491
<input type="checkbox"/> CodeWarrior RS08 Debugger	10.1.1.FSL_b5491
<input type="checkbox"/> ColdFire Database	1.0.2
<input checked="" type="checkbox"/> CW MCU v10.1 Update	1.0.1
<input type="checkbox"/> Kinetis Runtime	1.0.4
<input type="checkbox"/> MCU	10.1.13
<input type="checkbox"/> MCU Debug Connections	10.1.2.FSL_b5492
<input type="checkbox"/> MCU v10.1 Compiler Update (ColdFire, Kinetis, Qorivva)	1.0.2

Details

☒ Show only the latest versions of available software ☐ Hide items that are already installed
☐ Group items by category [What is already installed?](#)
☒ Contact all update sites during install to find required software

- 2) Install the com.freescale.mcu10_1.Kinetis_50MHz_K10_K20.win.v1.0.2.zip patch by following the steps as stated in the document **How_to_install_ServicePackUpdaterArchiveforEclipseCodeWarrior.pdf**. Both files are stored in \kinetis-sc\utils

Note:

Make sure all dependencies are already installed before service pack. See section 2 “Install the Service Pack dependencies”.