TWR-MCF51JG Getting Started Guide

MCF51JG256 Software Development

Rev. 1

Table of Contents

Step-By-Step Instructions

Supported: Windows XP/ Windows Vista/ Windows 7

1   Purpose

The purpose of this guide is to help you get started with the TWR-MCF51JG Tower module so you can quickly start developing your application. The guide will walk you through the installation and usage of software and tools for the ColdFire+ MCF51JG and MCF51QW microcontrollers. This guide will cover Freescale CodeWarrior IDE and Freescale MQX RTOS.

2   Download and Install Software and Tools

Download software and documentation under "Jump Start Your Design" at www.freescale.com/TWR-MCF51JG.

2.1 Install the P&E OSBDM OSJTAG Virtual Serial Toolkit.

Install the P&E Micro OSBDM OSJTAG Tower Toolkit to install the OSBDM and USB-to-Serial drivers. You may need to restart your computer for proper driver installation.

See Section 9 OSBDM Overview for details on OSBDM.

2.2 Install CodeWarrior for Microcontrollers

Evaluation editions are available. Search www.freescale.com:
https://www.freescale.com/webapp/Download?colCode=CW_MCU_10_2_EVAL.

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CW-MCU10&fpsp=1&tab=Design_Tools_Tab

Note: The MQX software package was tested with CodeWarrior for MCUs v10.2, v10.3 and v10.4

2.3 Install Freescale MQX 3.8.1 for TWR-MCF51JG

At the time of this document release, MQX RTOS for TWR-MCF51JG is provided in a standalone package download. Search www.freescale.com: FSLMQXOS_3_8_1_TWRMCF51JG. This is a separate download than the main MQX installer.

Steps

1. Download FSLMQXOS_3_8_1_TWRMCF51JG (FSLMQX 3.8.1 TWRMCF51JG.exe) to any directory on your computer. Prior installation of MQX 3.8.1 is not required.

2. Click FSLMQX 3.8.1 TWRMCF51JG.exe.

3. Review the click-through license agreement.

4. Select *complete* installation. This will install all MQX software for TWR-MCF51JG and the MQX Task Aware Debugging plug-in for CodeWarrior 10.2.

5. The default <MQX installation directory> for Windows XP is:
   C:\Program Files\Freescale\Freescale MQX 3.8 TWRMCF51JG

6. The default <MQX installation directory> for Windows Vista / Windows 7 is:
   C:\Freescale\ Freescale MQX 3.8 TWRMCF51JG

Note that the pre-compiled MQX libraries have references to the Windows XP default directory. Therefore, if you are using Windows Vista, Windows 7, or you did not install to the default directory for Windows XP, then we highly recommend that you recompile the MQX libraries.

3   Configure the Hardware

If running the TWR- MCF51JG board in stand-alone mode (i.e. without other Tower modules):

1. Connect one end of the USB cable to your computer and the other end to the OSBDM mini-USB connector J14. Allow the computer to automatically configure the USB drivers if needed. Click "*Next"* through the driver installation windows.

If using the TWR- MCF51JG board within the Freescale Tower System with other Tower modules:

1. Follow the steps in the section above.

2. Connect the TWR- MCF51JG and other Tower modules 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   Open the Terminal Utility (See section 9 OSBDM)

Go to Start…Programs…P&E OSBDM OSJTAG Virtual Serial Toolkit...Utilities…Terminal Utility

Set the Baud rate to 19200 (important – this is not the default).
Other settings – Port: USB COM, Parity: None, Bits: 8.
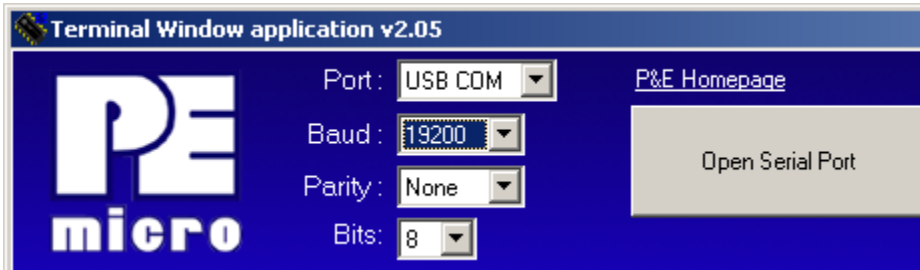
Click Open Serial Port.

If you cannot open serial port:

Note that depending on the drivers that you use, you might have to change the "Port:" connection.
Refer to OSBDM/OSJTAG Com Port section.

5   Explore the Out-of-Box Demo

The Out-of-Box demo application pre-programmed on the MCU flash utilizes the MQX Real Time
Operating System. This simple application demonstrates the conservation of power by going into a low
power state after a certain timeout period. Additionally, it demonstrates waking up from low power
with a Real Time Clock (RTC) or button press wake-up events.

Initially, the orange LED blinks at a slow rate. The microcontroller prints a short message to the
terminal once a second.

Press the SW2 button. The demo application goes into timeout mode. It will count down from 20
seconds and then go to very low power stop (VLPS) mode.

After 10 seconds, the RTC will wake it back up and start the demo over. Pressing SW2 again will also wake it up.

6   Re-Programming the Out-of-Box Demo

If you erase the demo application on the TWR-MCF51JG, it can be re-flashed by either programming the pre-compiled binary data file (s-record) or by opening the project in CodeWarrior, re-compiling, and programming the application in the MCU flash.

The demo application files are available within the Quick Start Package in the file: *twr-mcf51jg_demo_rev1.1.zip*

6.1 Method 1 – Programming the s-record

The pre-compiled demo application s-record is *twr-mcf51jg_demo_precompiled.S19* which can be found in the in the *twr-mcf51jg_demo* folder of the Quick Start Package file *twr-mcf51jg_demo_rev1.1.zip*
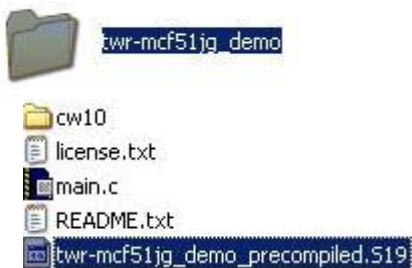
Steps
1. The simplest method to establish the correct connection settings for the TWR-MCF51JG is to create a new bareboard project in CodeWarrior. Follow the steps in section Creating a new Bareboard Project with CodeWarrior

2. Next, on the top toolbar, right click on the arrow next to the lightning bolt (Flash Programmer), select "Flash File to Target".
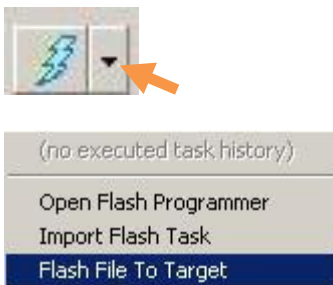
3. At File Options, click Browse... chose File System.

4. Browse for the file: *twr-mcf51jg_demo_precompiled.S19*

5. Click Erase and Program

6. Wait until the erase and program operations are complete.

   ```
   Program Command Succeeded
   Flash Operation. done
   ```

7. Press the Reset button on the board. The demo is now programmed and running!

6.2 Method 2 – Opening the project, compiling, programming.

The following instructions describe placing the project folder within the MQX directory tree, opening the project in CodeWarrior, re-compiling, and programming the application in the MCU flash.

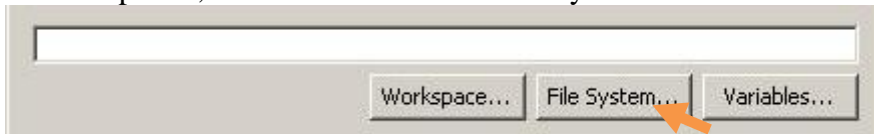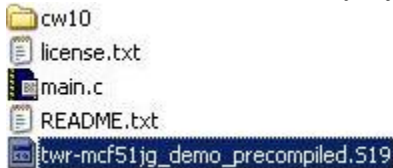The demo application project can be found in the Quick Start Package file *twr-mcf51jg_demo_rev1.1.zip*

Steps
1. Follow steps under Set values and recompile BSP

2. Install MQX for TWR-MCF51JG. See section Install Freescale MQX 3.8.1 for TWR-MCF51JG

3. Copy and paste the folder twr-mcf51jg_demo into the directory:
   <MQX installation directory> Freescale MQX 3.8 TWRMCF51JG**demo**\

4. Open the project in CodeWarrior. See section on opening a MQX example application project.

5. Select the twr-mcf51jg_demo_twrmcf51jg project.

6. Click the Build icon to compile the project.

7. Right-Click on the project name in the project panel and select **Debug As -> Debug Configurations**. Under **CodeWarrior Download** select *twr-mcf51jg_demo_twrmcf51jg_Int_Flash_Debug_OSBDM*

8. This will download the project to the board and start the debugger.

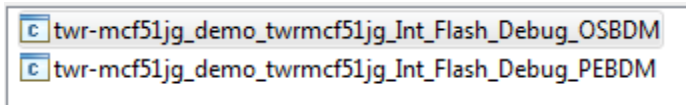| Debug As | ▶ | c 1 CodeWarrior Attach |
Profile As	▶	c 2 CodeWarrior Connect
Convert To...		c 3 CodeWarrior Download
Edit Linked Resources Locations...		Debug Configurations...
Team	▶	
Compare With	▶	202
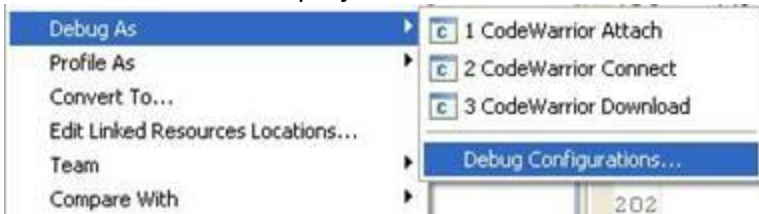
Right click project name…choose Debug As… Debug Configurations…

c twr-mcf51jg_demo_twrmcf51jg_Int_Flash_Debug_OSBDM
c twr-mcf51jg_demo_twrmcf51jg_Int_Flash_Debug_PEBDM

In Debug configurations select, twr-mcf51jg_demo_twrmcf51jg_Int_Flash_Debug_OSBDM Then click the debug button.

Alternatively, you if you have previously launched a debug session, you can launch it again by

clicking on the Debug button .

9. The code will then be automatically flashed to the internal flash memory of the MCU. Wait some time until it downloads completely. Once it download completely the debugger starts.

10. The Debug perspective will open and the code will pause at the start of the MQX main function.

11. Click the run icon to continue the program execution.

7   Using MQX RTOS

Refer to the Install Freescale MQX 3.8.1 for TWR-MCF51JG section for MQX installation instructions

7.1 Review MQX documentation

Documentation can be found in the directory:
<MQX Installation Directory>\Freescale MQX 3.8 TWRMCF51JG\doc\

Release notes for this MQX standalone package for TWR-MCF51JG:
FSL_MQX_TWRMCF51JG_release_notes.pdf

MQX getting started guide: FSL_MQX_getting_started.pdf
and MQX 3.8.1 release notes: FSL_MQX_release_notes.pdf

7.2 Recompile MQX libraries

The MQX package for TWR-MCF51JG comes with pre-compiled libraries (found in <MQX Installation Directory>\ Freescale MQX 3.8 TWRMCF51JG\lib\twrmcf51jg.cw10\). The pre-compiled MQX libraries have references to the Windows XP default directory. The debugger uses these references to locate library source files when debugging.

If you are using Windows Vista, Windows 7, or you did not install to the default directory on Windows XP, then we highly recommend that you recompile the MQX libraries, so the references are updated to the correct directory path.
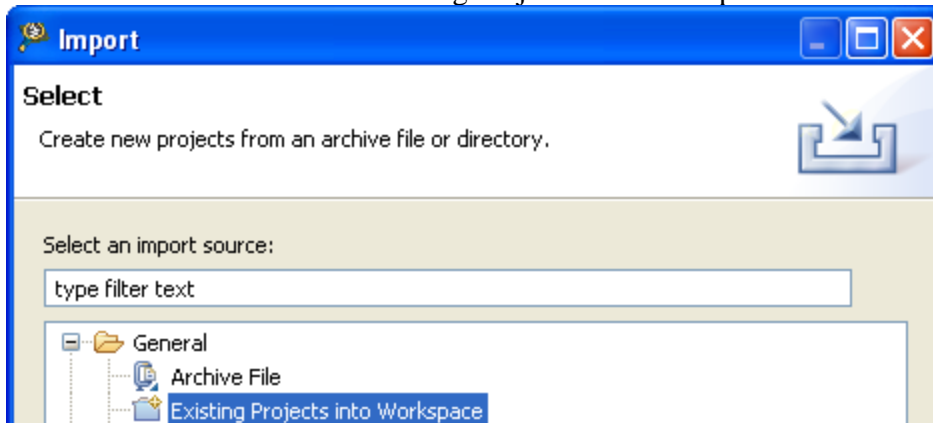
Steps
1. Open CodeWarrior.

2. Select a workspace directory. Any workspace directory is acceptable, however we recommend using the directory which you installed Freescale MQX 3.8 TWRMCF51JG.

   Example workspace:

   Workspace: | C:\Freescale\Freescale MQX 3.8 TWRMCF51JG | ▼ | Browse...

3. Import the library projects. Go to File…Import. When the Import dialogue box appears, expand the General folder and select Existing Projects into Workspace. Click Next.

   Import
   Select
   Create new projects from an archive file or directory.

   Select an import source:
   type filter text

   General
     Archive File
     Existing Projects into Workspace

4. Next, select the directory of the projects you wish to import.

5. Browse to the directory **<MQX Install Directory>\mqx\build\cw10** to import the MQX processor support package (psp) and board support package (bsp) projects.
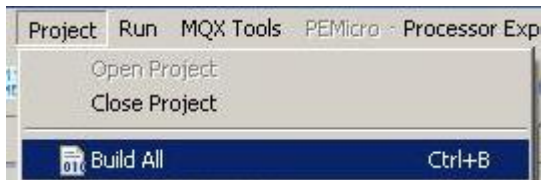
   (e.g. C:\Freescale\Freescale MQX 3.8 TWRMCF51JG**mqx\build\cw10**)

   Click OK.

   Important Note: Do not select the checkbox to copy projects into workspace.

6. Click Finish. The projects will now be imported into the workspace for modifying and rebuilding.

7. Select Project…Build All.

8. Click on each project in the Codewarrior project browser and select Edit…Delete.   This will remove the project from the workspace. It will not delete the file from the directory that it is located in.

   Note: if you plan on using the other mqx libraries (i.e. mfs, rtcs, usb, or shell), import and build those projects from their respective folders as well.
   (e.g. for RTCS, import <Install Directory>\Freescale MQX 3.8 TWRMCF51JG\rtcs\build\cw10)

7.3 Opening a MQX application project

1. Open CodeWarrior.

2. Select a workspace directory. Any workspace directory is acceptable, however we recommend using the directory which you installed Freescale MQX 3.8 TWRMCF51JG.

Example workspace:

Workspace: | C:\Freescale\Freescale MQX 3.8 TWRMCF51JG | ▼ | Browse...

3. Import the project. Go to File…Import. When the Import dialogue box appears, expand the General folder and select Existing Projects into Workspace. Click Next.
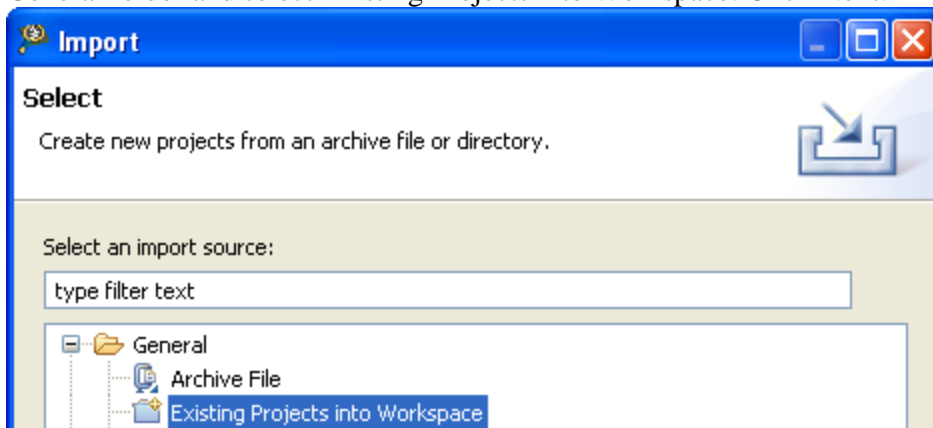
4. Next, select the directory of the project you wish to import.
(e.g. <MQX Installation Directory>\demo\twr-mcf51jg_demo)

5. Click OK.

Important Note: Do not select the checkbox to copy projects into workspace.

6. Click Finish. The projects will now be imported into the workspace for modifying and rebuilding.

7.4 Set values and Recompile BSP

The BSP package for TWR-MCF51JG comes with choices to enable MQX_ENABLE_LOW_POWER and BSPCFG_ENABLE_RTCDEV. These are required in order to run the out of box demo. The BSP needs to be imported and recompiled in order to make sure the values are set.
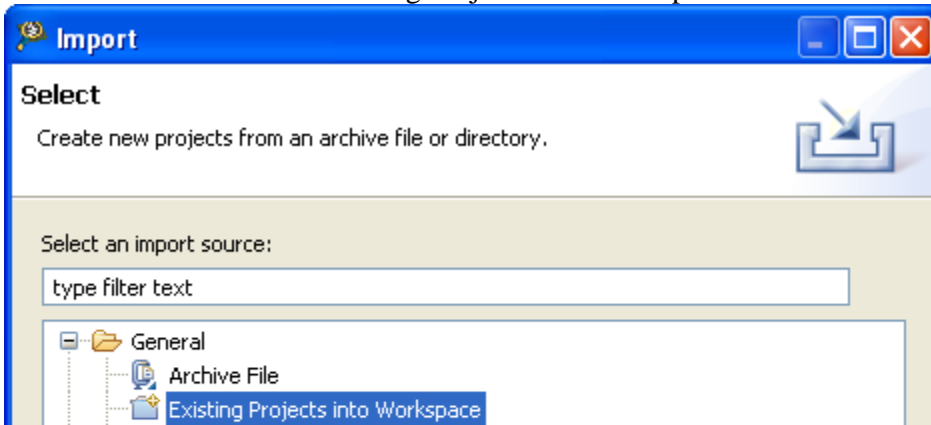
Steps
1. Open CodeWarrior.

2. Select a workspace directory. Any workspace directory is acceptable, however we recommend using the directory which you installed Freescale MQX 3.8 TWRMCF51JG.
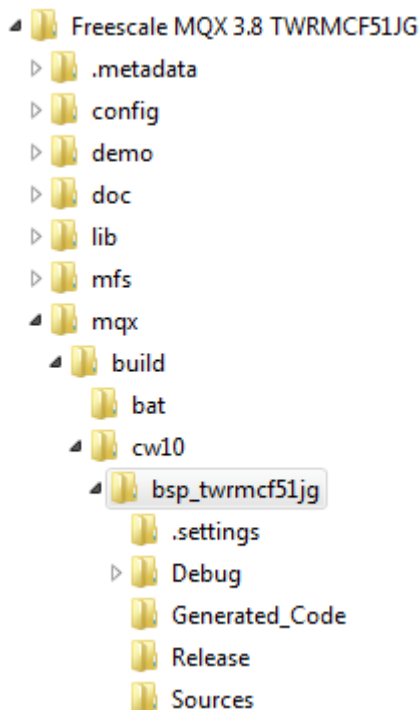
Example workspace:

Workspace: C:\Freescale\Freescale MQX 3.8 TWRMCF51JG   ▼   Browse...
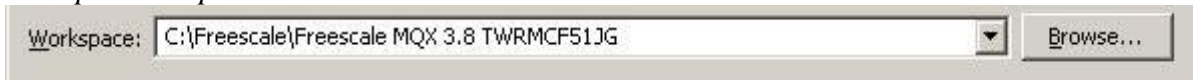
3. Import the bsp project. Go to File…Import. When the Import dialogue box appears, expand the General folder and select Existing Projects into Workspace. Click Next.

Import

Select

Create new projects from an archive file or directory.

Select an import source:

type filter text

- General
  - Archive File
  - **Existing Projects into Workspace**

4. Next, select the directory of the projects you wish to import.

5. Browse to the directory <MQX Installation Directory>\Freescale MQX 3.8 TWRMCF51JG\mqx\build\cw10\bsp_twrmcf51jg to import the bsp project.

- Freescale MQX 3.8 TWRMCF51JG
  - .metadata
  - config
  - demo
  - doc
  - lib
  - mfs
  - mqx
    - build
      - bat
      - cw10
        - bsp_twrmcf51jg
          - .settings
          - Debug
          - Generated_Code
          - Release
          - Sources

Click OK.

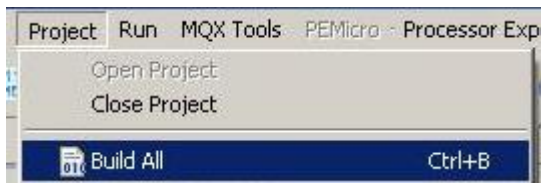**Important Note: Do not select the checkbox to copy projects into workspace.**

6. Click Finish. The project will now be imported into the workspace for modifying and rebuilding.

7. Select the bsp_twrmcf51jg project in the Codewarrior workspace window

8. Open user_config.h

9. Browse the BSP parameters and make sure that BSP_ENABLE_RTCDEV is set to true.

BSPCFG_ENABLE_GPIODEV	false	☑	user_config.h
BSPCFG_ENABLE_RTCDEV	true	☑	user_config.h
BSPCFG_ENABLE_FLASHX	false	☑	user_config.h

10. Browse the MQX parameters and make sure that MQX_ENABLE_LOWPOWER is set to true.

MQX_DEFAULT_USER_ACCESS	true	☐	mqx_cnfg.h
MQX_ENABLE_USER_STDAPI	true	☐	mqx_cnfg.h
MQX_ENABLE_LOW_POWER	true	☑	user_config...

11. Select Project…Build All.

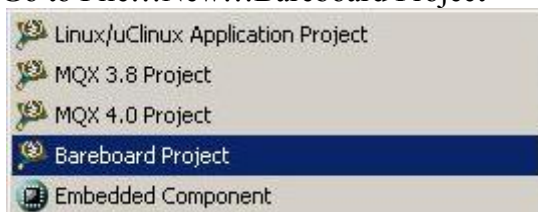|				
Project	Run	MQX Tools	PEMicro	Processor Exp
Open Project				
Close Project				
Build All			Ctrl+B	

8   Creating a new Bareboard Project with CodeWarrior

Steps
1. Open Freescale CodeWarrior for MCUs. Choose any workspace you like.

2. Go to File…New…Bareboard Project

- Linux/uClinux Application Project
- MQX 3.8 Project
- MQX 4.0 Project
- Bareboard Project
- Embedded Component

3. Give the project a name. Click Next.

Project name: test123

4. Choose the device. Select ColdFire+ V1…MCF51JG Family…MCF51JG256. Click Next

ColdFire+ V1
  MCF51JF Family
  MCF51JG Family
    MCF51JG64
    MCF51JG128
    MCF51JG256

5. Choose the connection. Select Open Source BDM. Unselect all others. Note: If you have another debug interface pod, you can select it here. Just plug it into J17 (51JG BDM) header.
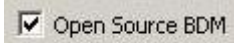
☑ Open Source BDM

6. Click Finish. Note: no changes are necessary to the remaining settings, so instead of clicking Next, you can just click Finish.

7. Now you have a bareboard project complete with main function and interrupt handling.

9   OSBDM Overview

Open Source BDM (OSBDM) allows a user to program, debug, and get serial data from ColdFire+ devices via a USB cable. The firmware is located on a Freescale MCFS08JM60 on the underside of the TWR-MCF51JG ColdFire+ Tower module.

When the Tower module is plugged in, it should enumerate as a composite device, with one driver for debugging, and the other as a serial port.

Get all the latest drivers and utilities for OSBDM from P&E Micro at
http://www.pemicro.com/osbdm/index.cfm

9.1 IMPORTANT NOTE ON OSBDM FIRMWARE REVISIONS

9.1.1 Revision 30.23 – Serial data passes to P&E Terminal Utility

At the time of this document release, the OSBDM firmware revision pre-programmed on TWR-MCF51JG is revision 30.23. This revision is fully compatible with CodeWarrior 10.2 and the P&E OSBDM OSJTAG Virtual Serial Toolkit.

### 9.1.2 Revision 31.25 – Serial data passes to COM port on computer

If you use CodeWarrior 10.3, then CodeWarrior will upgrade the OSBDM firmware revision to revision 31.25. Follow directions on the pop up windows in Codewarrior 10.3

## 9.2 OSBDM Troubleshooting:

If OSBDM is not working, do the following.

Open Control Panel and search Device Manager. Refer to steps below

9.2.1.1 PEMicro USB Serial Port
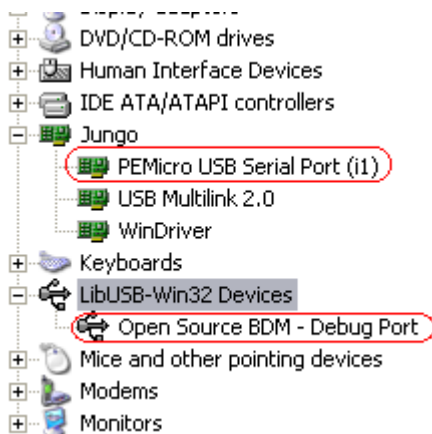
If you only see it enumerate as the Open Source BDM Debug Port, then your computer may be automatically picking up an outdated driver. To fix this, right click on the OSBDM driver and select "Uninstall". Then unplug and re-plug in the board, and it should enumerate correctly.
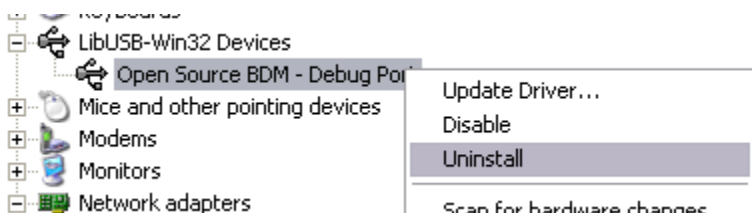
Depending on the OBDSM driver, it might look different on your device manager. In this case:

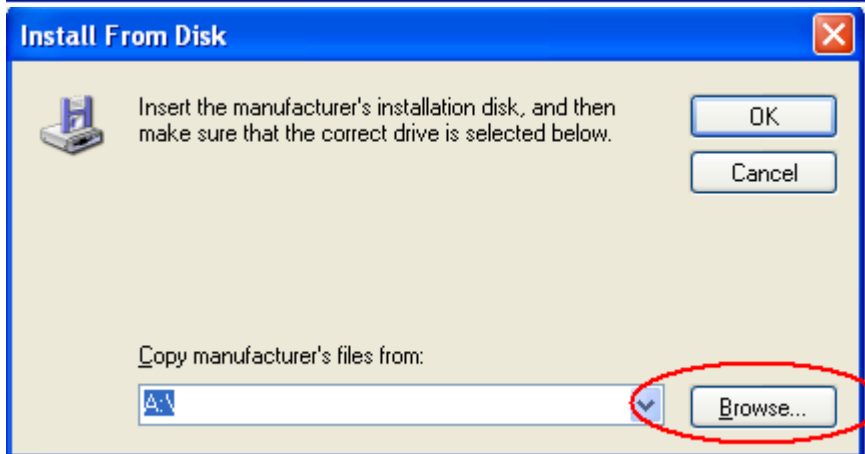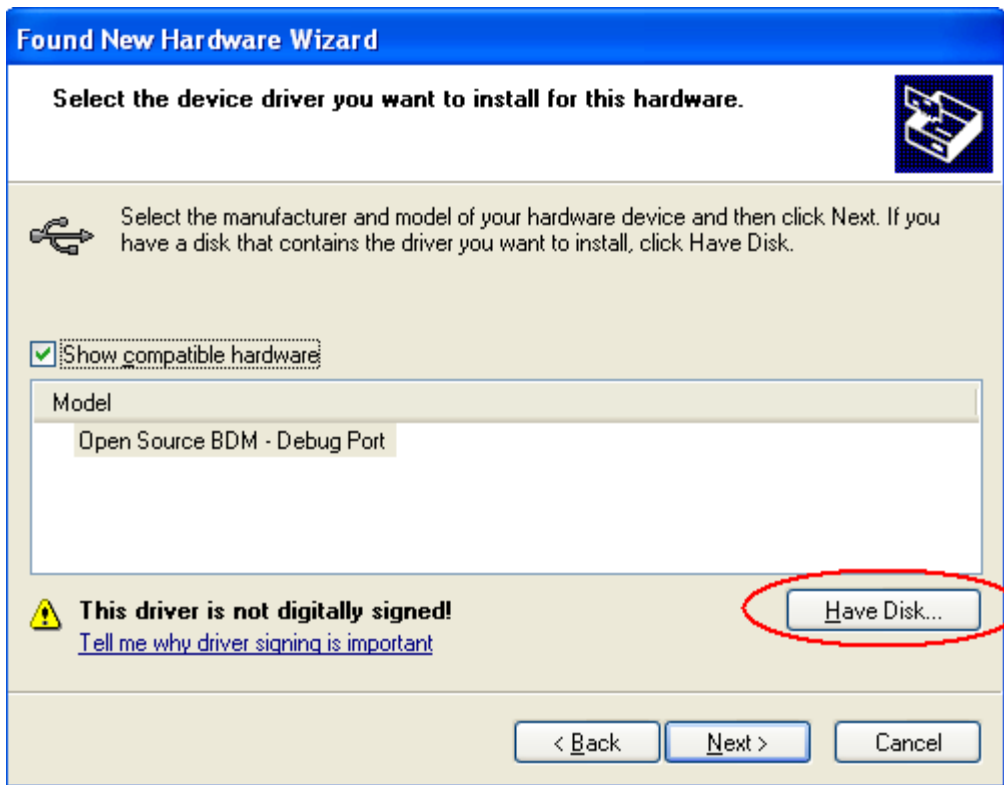### 9.2.1.2 OSBDM/OSJTAG COM Port

In this case, make sure that the PE micro terminal window is using port COM5. The COM port will vary depending on your system. So please check for the appropriate COM Port being used on your Device Manager.

If there are still problems with enumerating correctly, you can also manually select the drivers.
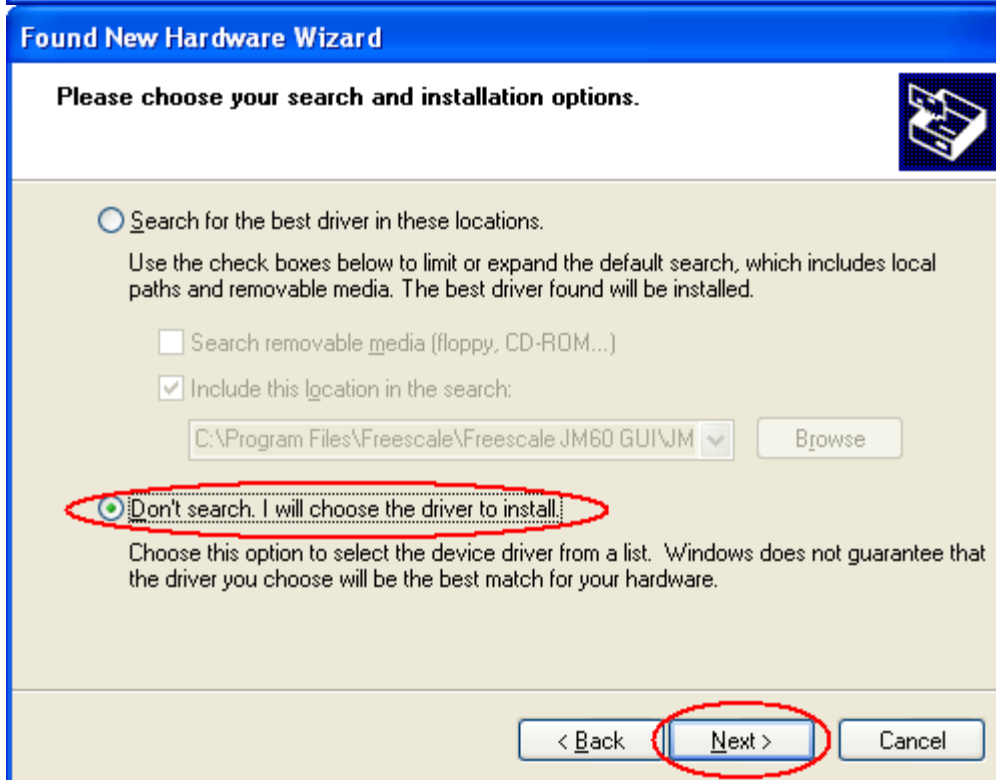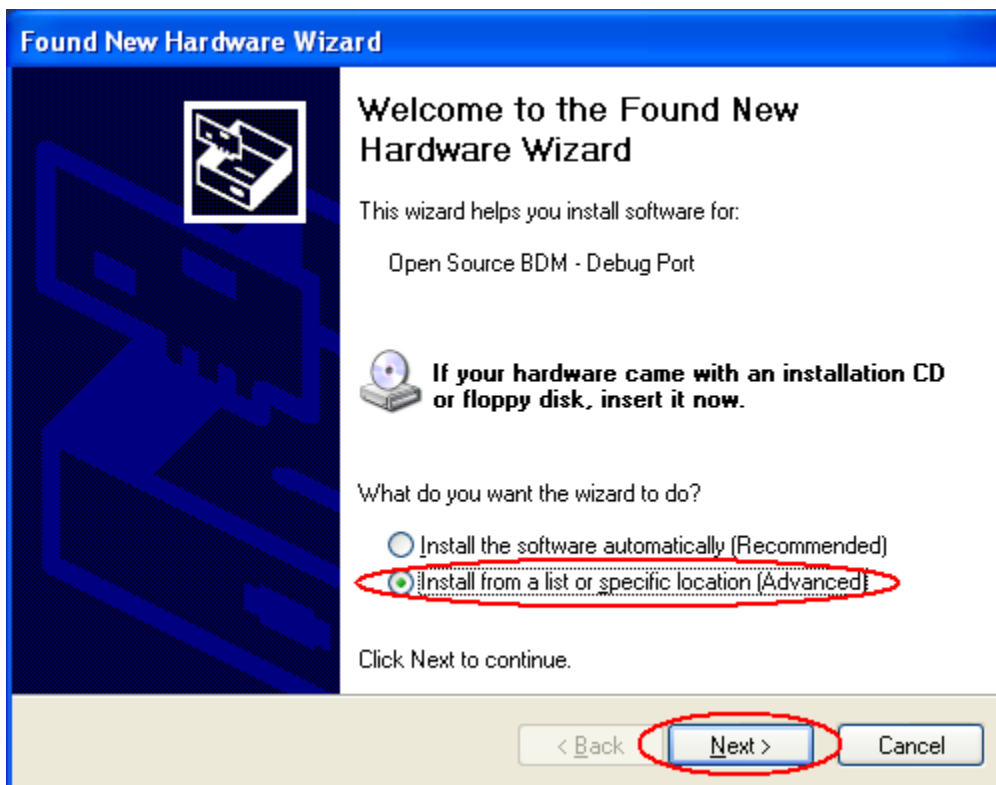
For the **Open Source BDM – Debug Port,** use the driver at:
C:\..\PEMicro\..\Drivers\osbdm\OSJTAG_Debug_Interface_libusb.inf

For the **PEMicro USB Serial Port (i1)**, use the driver at: **C:**
C:\..\PEMicro\..\Drivers\osbdm\OSJTAG_Serial_Interface_windriver_version.inf