Freescale Semiconductor, Inc
User's Guide

Document Number: MQXUSBSUG
Rev 1, 04/2015

MQX™ RTOS USB Stack User's Guide

1 Overview

This document describes how to compile the USB stack and examples, download a binary image, and run the examples. It takes TWRK22F120M Tower System module as an example, and provides the board specific information.

Contents

2 Requirements for Building USB Examples

TWR-K22F120M is taken as an example in the rest of the document. The operations of compiling, downloading, or running the examples on all the other boards are similar, except for the explicit description for specific boards.

2.1 Hardware

- TWR-K22F120M Tower System module
- (Optional) TWR-SER Tower System module and Elevator
- J-Link debugger(optional)
- USB cables

2.2 Software

- Freescale MQX™ RTOS 4.2 release package
- IAR Embedded Workbench for ARM® Version 7.30.4
- Keil µVision5 Integrated Development Environment Version 5.13, available for Kinetis ARM Cortex®-M4 devices
    - Keil.Kinetis_K20_DFP.1.2.0 pack
    - Keil.Kinetis_K60_DFP.1.3.0 pack
- Kinetis Design Studio IDE v2.5
- CodeWarrior 10.6 with all updates
- DS5 20.1 32bit
- GCC ARM 4.8-2014-q3

2.3 Board Jumper Settings

This document focuses on the USB-related jumper settings on the board. For other jumper settings, see the board-related user's guide. The board jumper settings are provided for all supported boards.

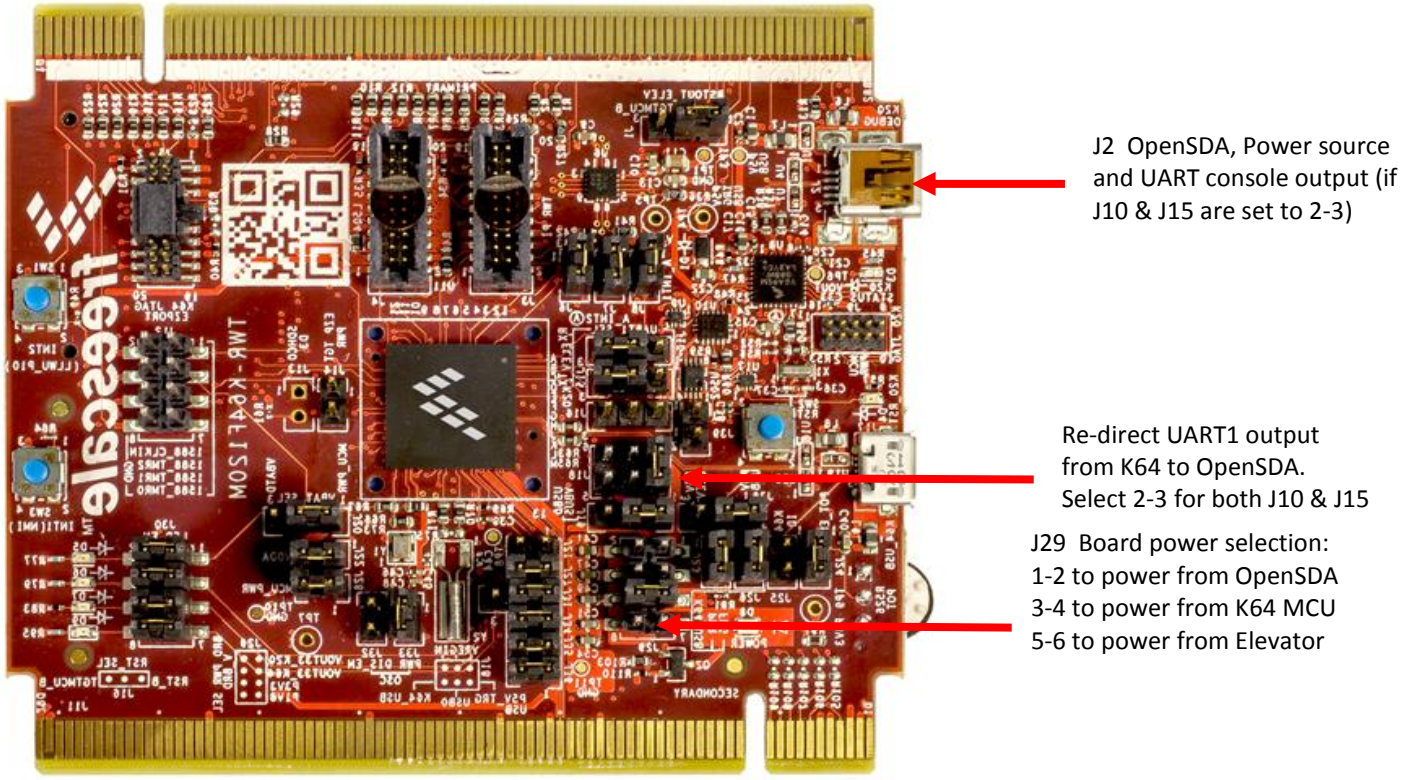

Figure-1 TWR-K64 Tower System module

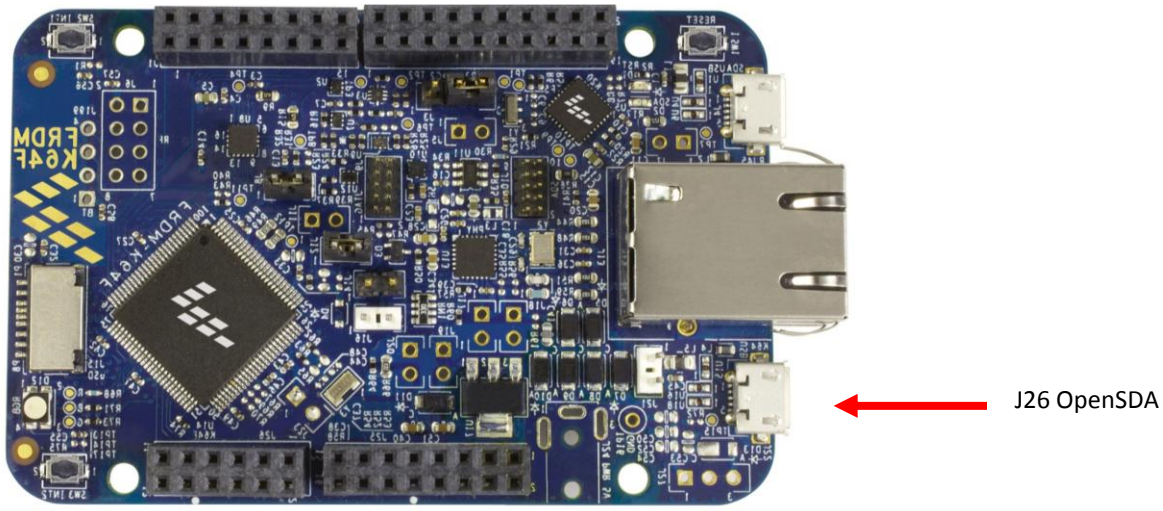

Figure-2 Freescale Freedom FRDM-K64F development platform

Connect J21

Install a 2.2uF capacitor in the position C33 as shown.

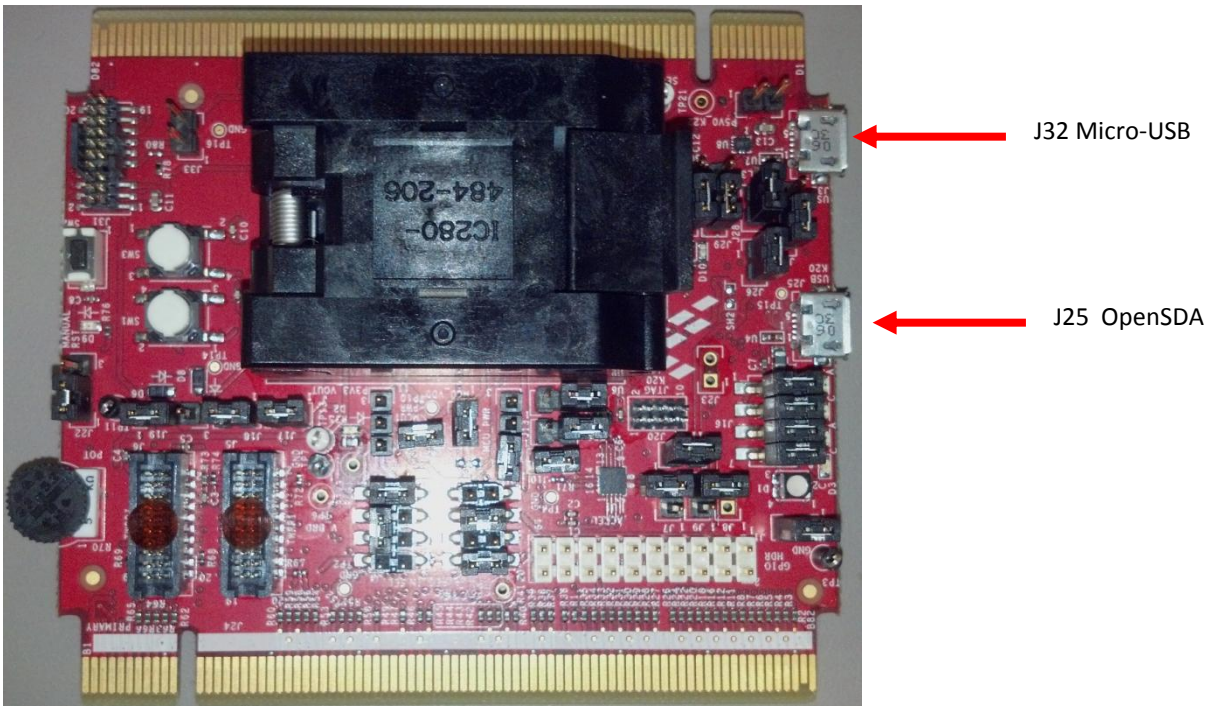

Figure-3 TWR-K22 Tower System module

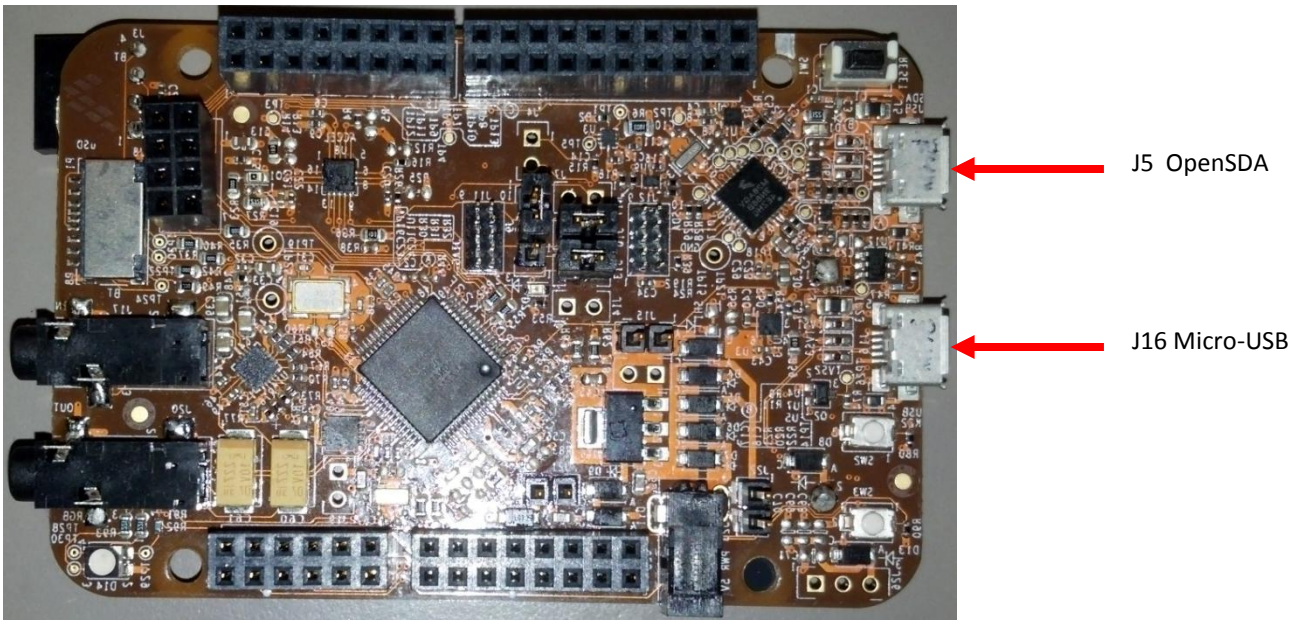

Figure-4 Freescale Freedom FRDM-K22 development platform

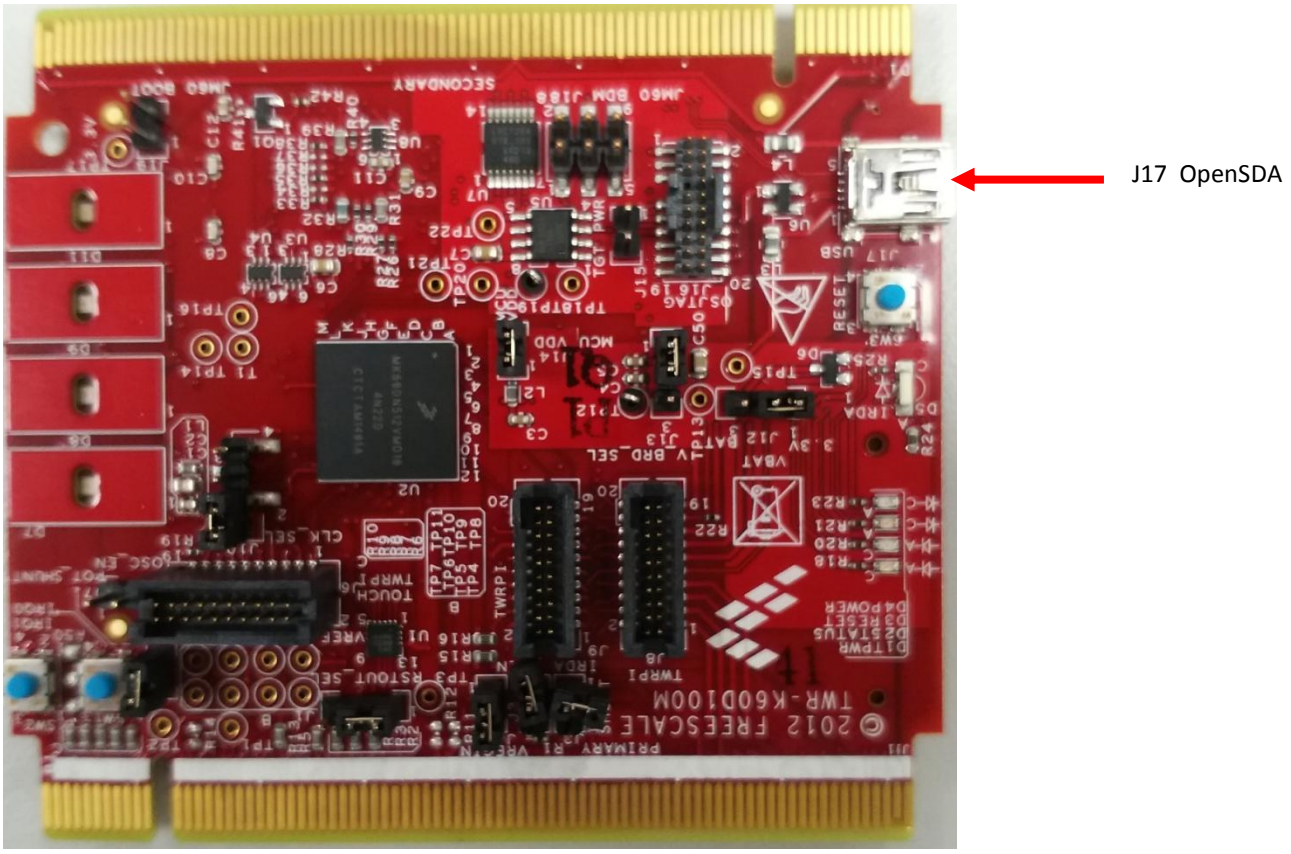

Figure-5 TWR-K60 Tower System module

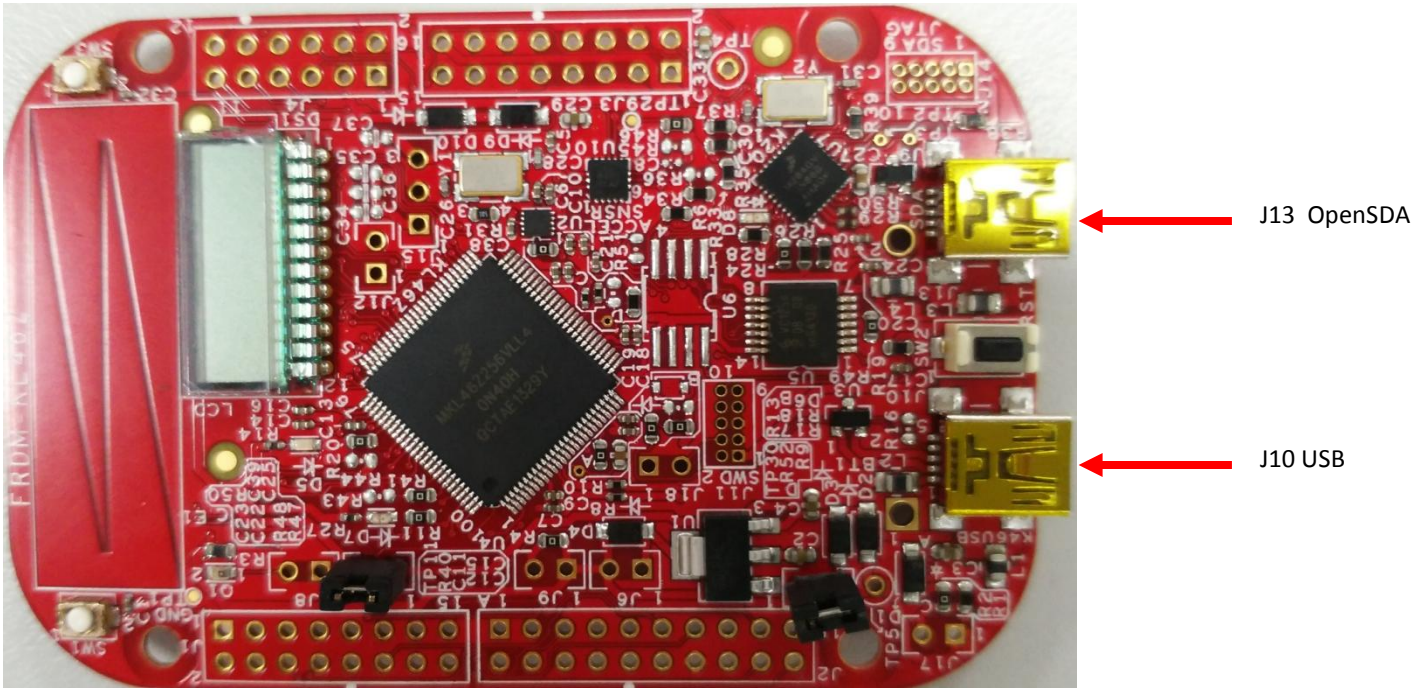

Figure-6 Freescale Freedom FRDM-KL46 development platform

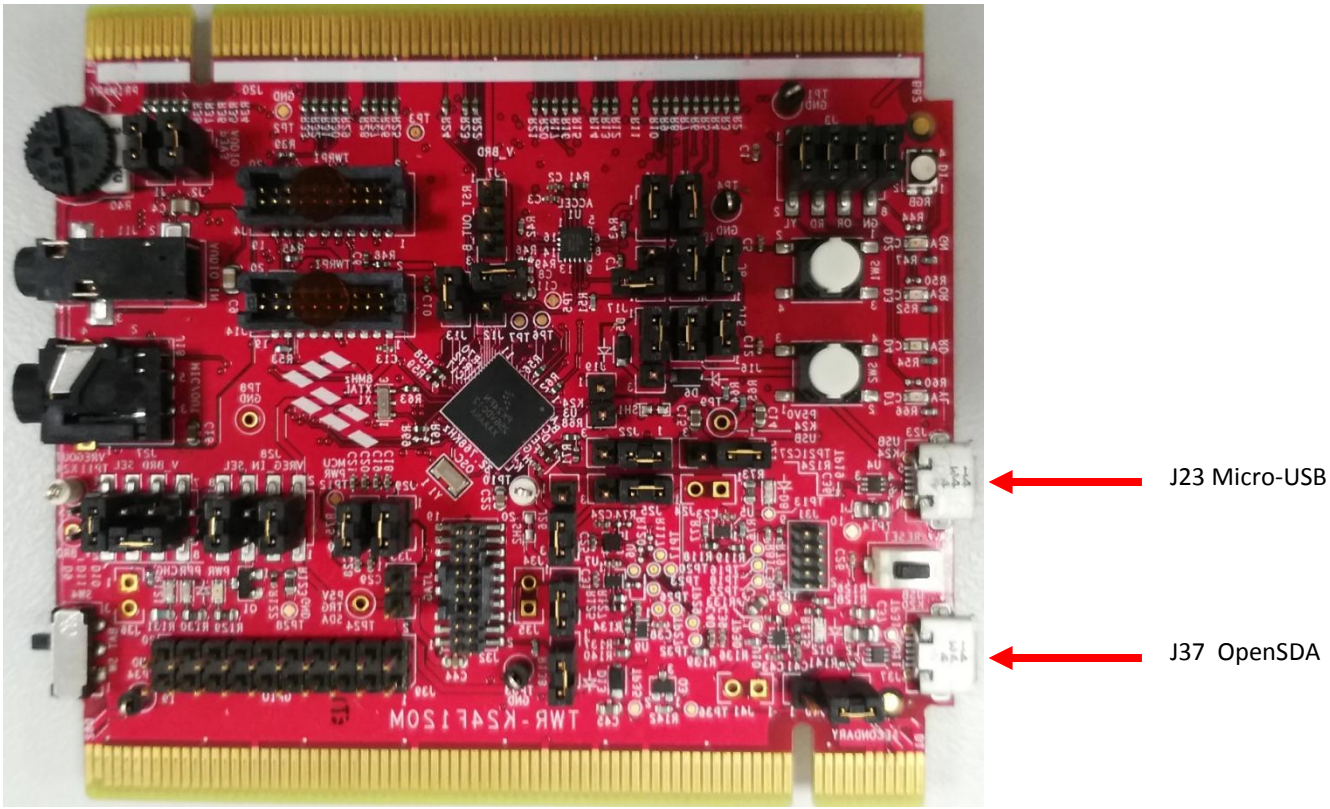

Figure-7 TWR-K24 Tower System module

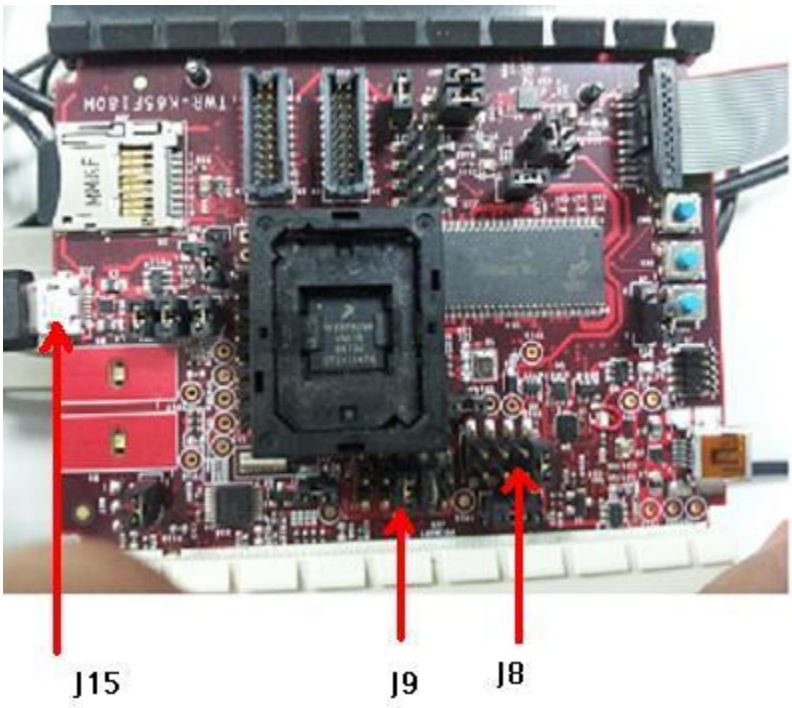

Figure-8 TWR-K65 Tower System Board jumper settings

- J8 1-2: ON
- J9 3-4: ON
- J9 5-6: ON

Note:

Only J15 on K65F180M can be used as the High Speed USB port, the USB port on TWR-SER2 board is not supported.

3 USB Code Structure

New USB stack is located in the "**usb_v2**" folder, in the root level of the MQX RTOS4.2 folder. There are some subfolders in it as shown in figure below:

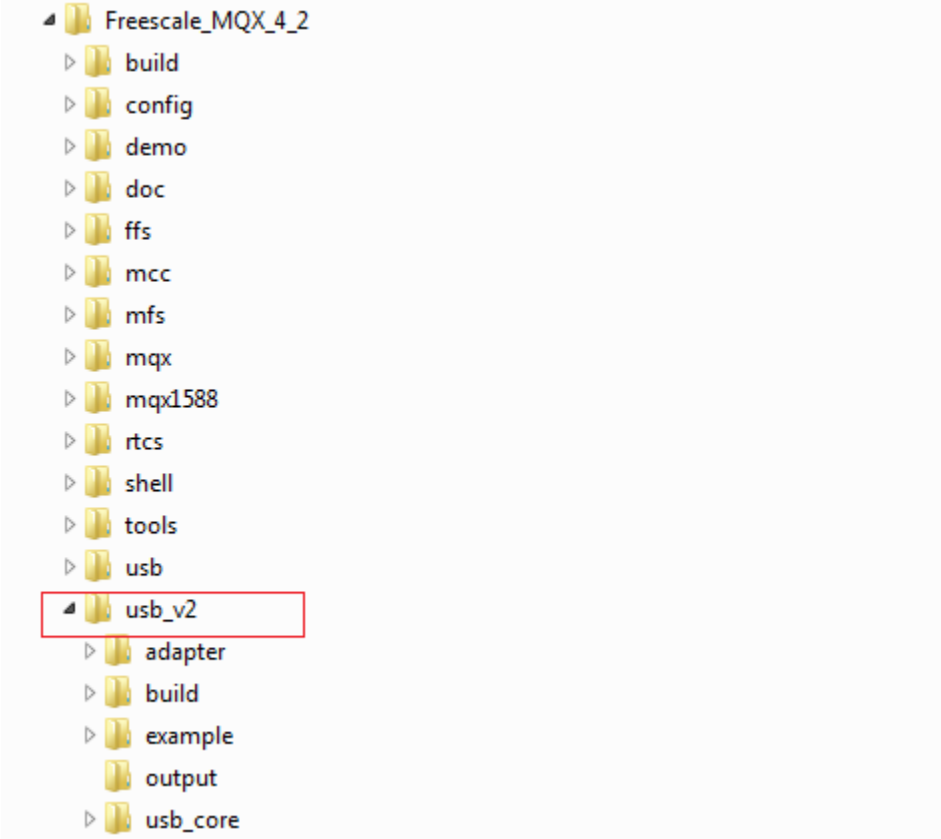

Figure-8 New USB stack folder structure

- adapter

  This subfolder includes the adapter files to support the USB stack running on a different RTOS with the same USB core code.

- build

  This subfolder includes arm gcc tool chains supported in USB stack.

- example

  This subfolder includes all source code and project files for the USB examples.

- output

  The USB library binary file will be generated into this folder and all the USB related public header files which may be used by the user will be copied to this folder so that the examples need to include one folder as the including path in the example project settings.

- usb_core

  This subfolder includes the USB source files, such as HAL, controller driver, class drivers, and the USB library projects.

4 Compiling or Running the USB Stack and Examples

4.1 Step-by-step guide for IAR

This section describes how to build and run USB example on IAR. The "**host_hid_mouse"** example will be used as an example.

1. Open IAR.
2. Add projects to IAR by clicking on **Project** tag and the select Add Existing Project. You can find corresponding IAR project files as below links.
    a. bsp library project
    `<install_dir>/mqx/build/iar/bsp_twrk22f120m`
    b. psp library project
    `<install_dir>/mqx/build/iar/psp_twrk22f120m`
    c. USB host library project
    `<install_dir>/usb_v2/usb_core/host/build/iar/usbh_mqx_twrk22f120m`
    d. USB host hid mouse example project
    `<install_dir>/usb_v2/example/host/hid/mouse/mqx/iar/host_hid_mouse_twrk22f120m`

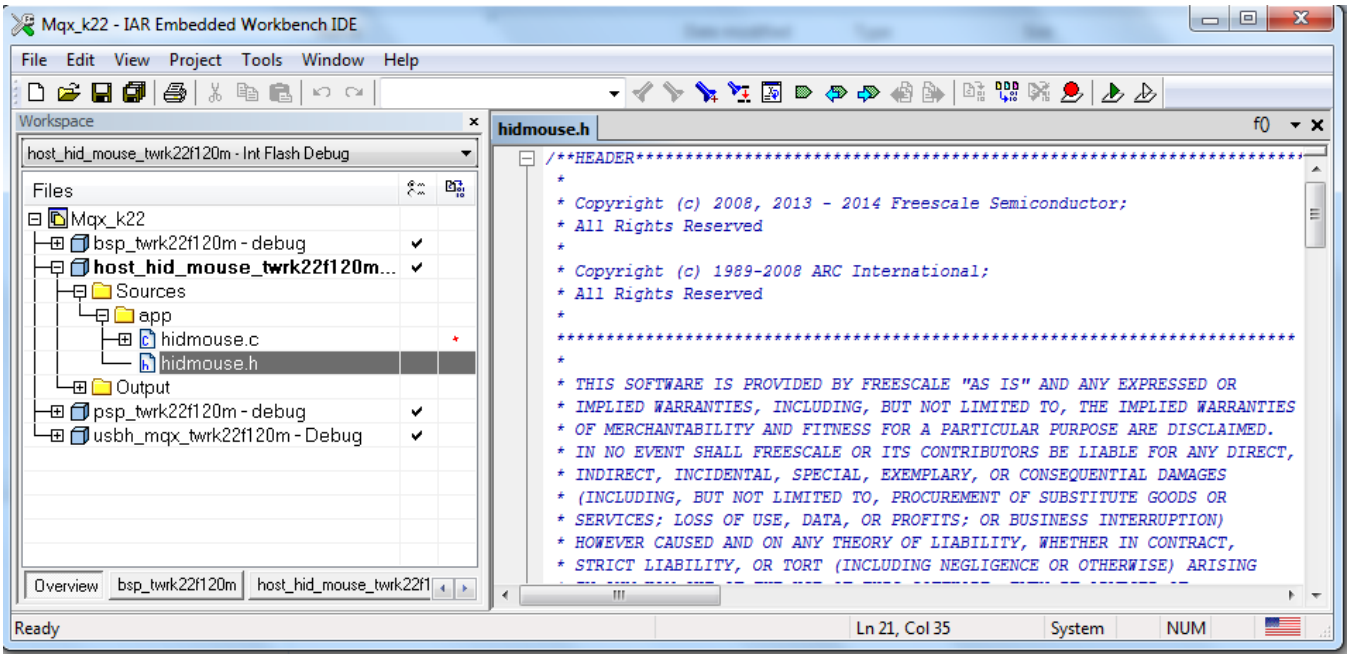

Figure-9 IAR workspace

3. Build the MQX RTOS **bsp** and **psp** library.
4. Build the **usbh_mqx_twrk22f120m** library.
5. Check the USB library build result.
6. After the USB library is built, the generated library binary file (libusbh_mqx.a) is located in `<install_dir>/usb_v2/output/twrk22f120m.iar/debug/usbh/mqx`.
7. All USB-related public header files are copied to this folder.
8. Build the **host_hid_mouse_twrk22f120m_mqx** example.

9. Connect the micro USB cable from a PC to the J25 of the TWR-K22F120M Tower System module to power on the board.
10. Click the "**Download and Debug**" button. Wait for the download to complete.
11. Click the "**Go**" button to run the example.

4.2 Step-by-step guide for KEIL

This section describes how to build USB stack and USB example on KEIL.

1. Open KEIL
2. Add project to KEIL by clicking on **Project** tag then select "**New Multi-Project Workspace…**" You can find corresponding KEIL project files as below links.
    a. bsp library project
    `<install_dir>/mqx/build/uv4/bsp_twrk22f120m`
    b. psp library project
    `<install_dir>/mqx/build/uv4/psp_twrk22f120m`
    c. USB host library project
    `<install_dir>/usb_v2/usb_core/host/build/uv4/usbh_mqx_twrk22f120m`
    d. USB host hid mouse example project
    `<install_dir>/usb_v2/example/host/hid/mouse/mqx/uv4/host_hid_mouse_twrk22f120m`

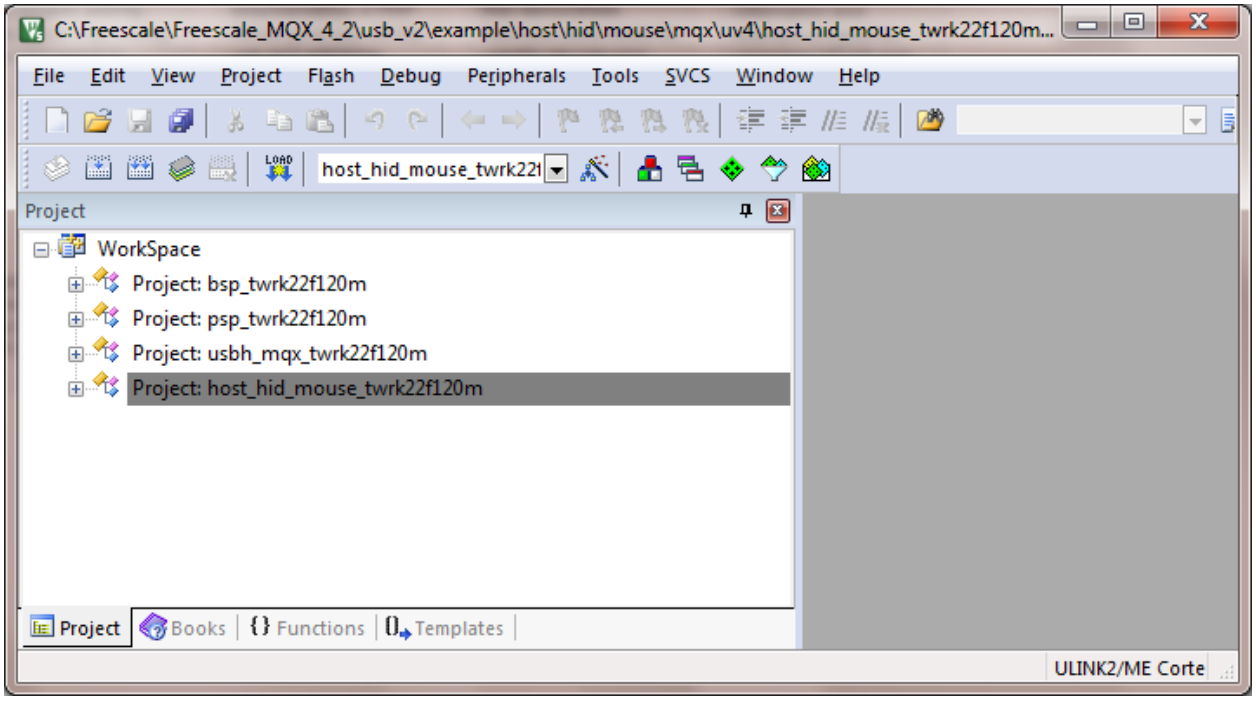

Figure-10 KEIL Workspace

3. Build the MQX RTOS **bsp** and **psp** library.
4. Build **the usbh_mqx_twrk22f120m** library.
5. Check the USB library build result.

6. After the USB library is built, the generated library binary file (libusbh_mqx.a) is located in `<install_dir>/usb_v2/output/twrk22f120m.uv4/debug/usbh/mqx`.
7. All USB-related public header files are copied to this folder.
8. Build the **host_hid_mouse_twrk22f120m_mqx** example.
9. Connect the micro USB cable from a PC to the J25 of the TWR-K22F120M Tower System module to power on the board
10. Click the "**Start/Stop**" debug session button. Wait for the download to complete.
11. Click the "**Go**" button to run the example.

4.3 Step-by-step guide for the Kinetis Design Studio IDE

1. Unlike IAR or KEIL, the Kinetis Design Studio IDE doesn't have a workspace. As a result, create a workspace and import Kinetis Design Studio IDE USB examples, platform libraries, and the USB stack library.
2. Select "**File**" then "**Import**" from the KDS IDE Eclipse menu.
3. Expand the General folder and select "**Existing Projects into Workspace**". Then, click the "**Next**" button.

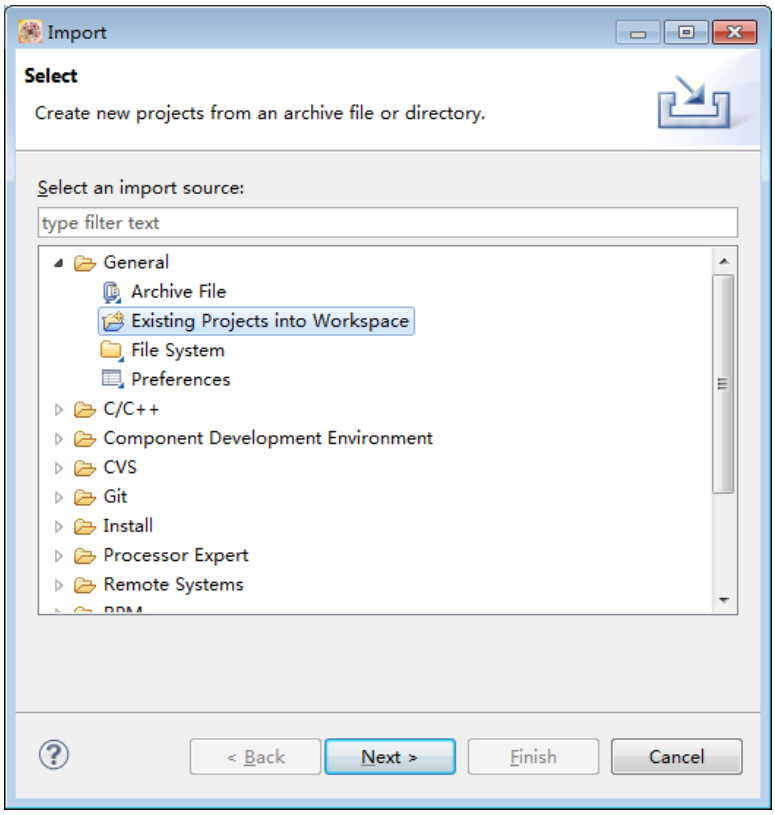

Figure-11 Selection of the correct import type in KDS IDE

4. Point the KDS IDE to the **bsp_twrk22f120m** in **<install_dir>/mqx/build/kds**. The import projects directory selection window should resemble this figure

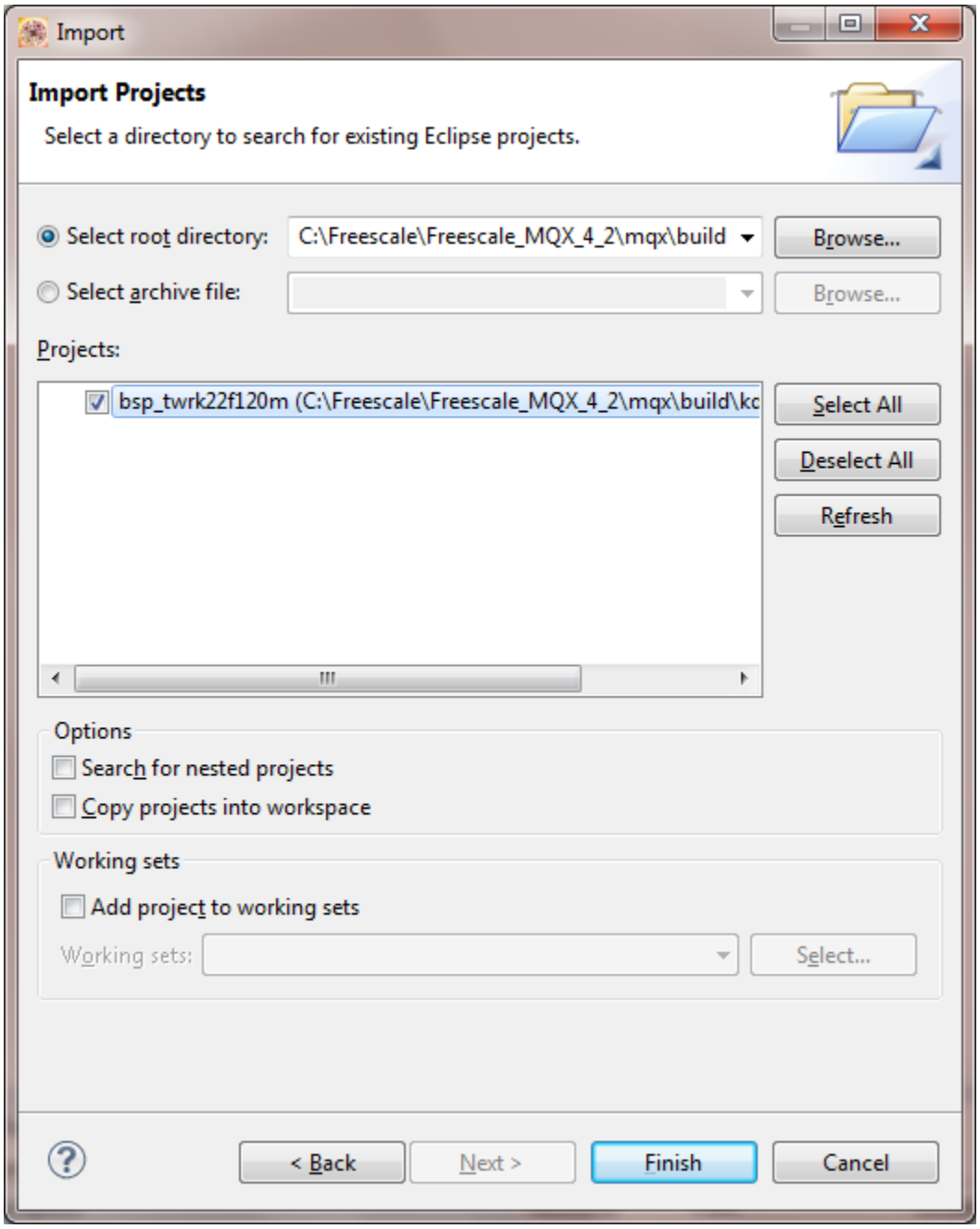

Figure-12 Selection of the K22 ksdk_platform_lib project

5. Following the same step to import the USB host library and the USB example, after importing them, the window should like this

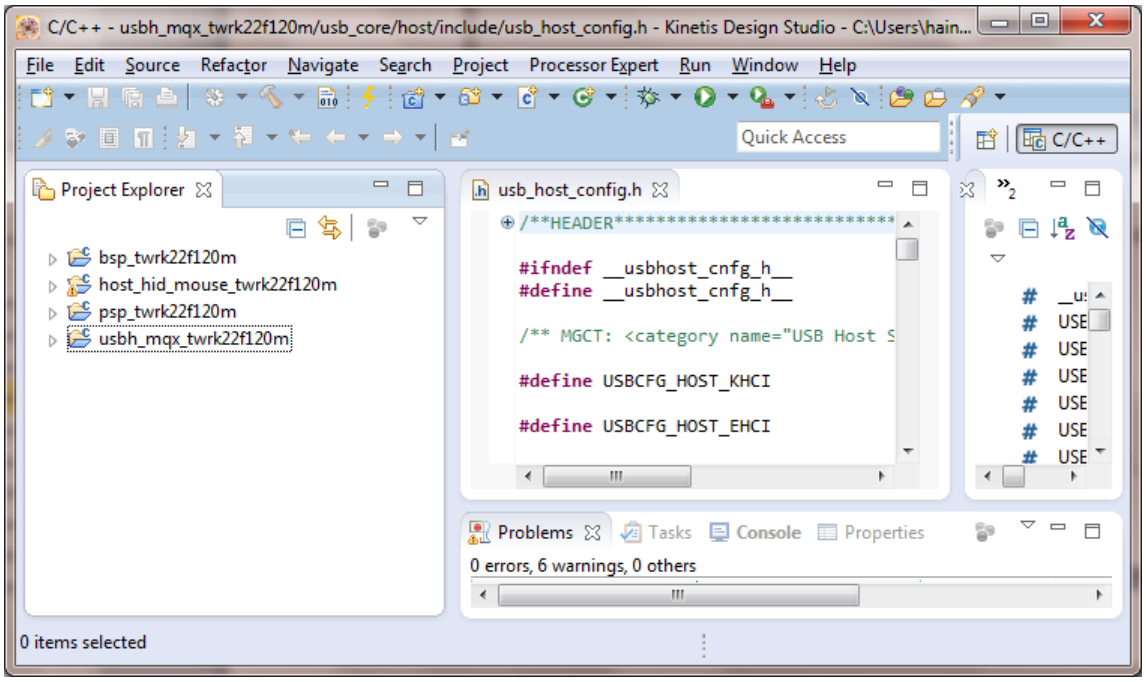

Figure-13 USB projects workspace

6. Choose the appropriate build target: "**Debug**" or "**Release**" by left-clicking the arrow next to the hammer icon as shown here.

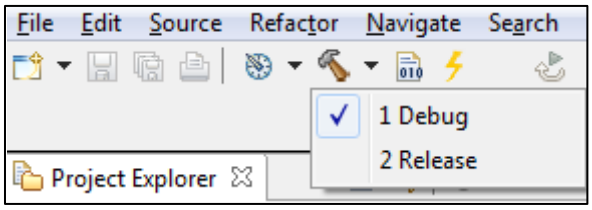

Figure-14 The hammer button

7. If the library build does not begin after selecting the desired target, left-click the hammer icon to start the build.
8. Following the same step to build the **bsp_twrk22f120m** library, **psp_twrk22f120m** library the **usbh_mqx_twrk22f120m** library and the **host_hid_mouse_twrk22f120m** example.
9. To check the debugger configurations, click the down arrow next to the green debug button and select "**Debug Configurations**".

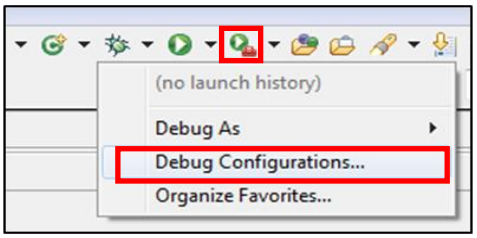

Figure-15 Debug configurations

10. After verifying that the debugger configurations are correct, click on the "**Debug**" button.

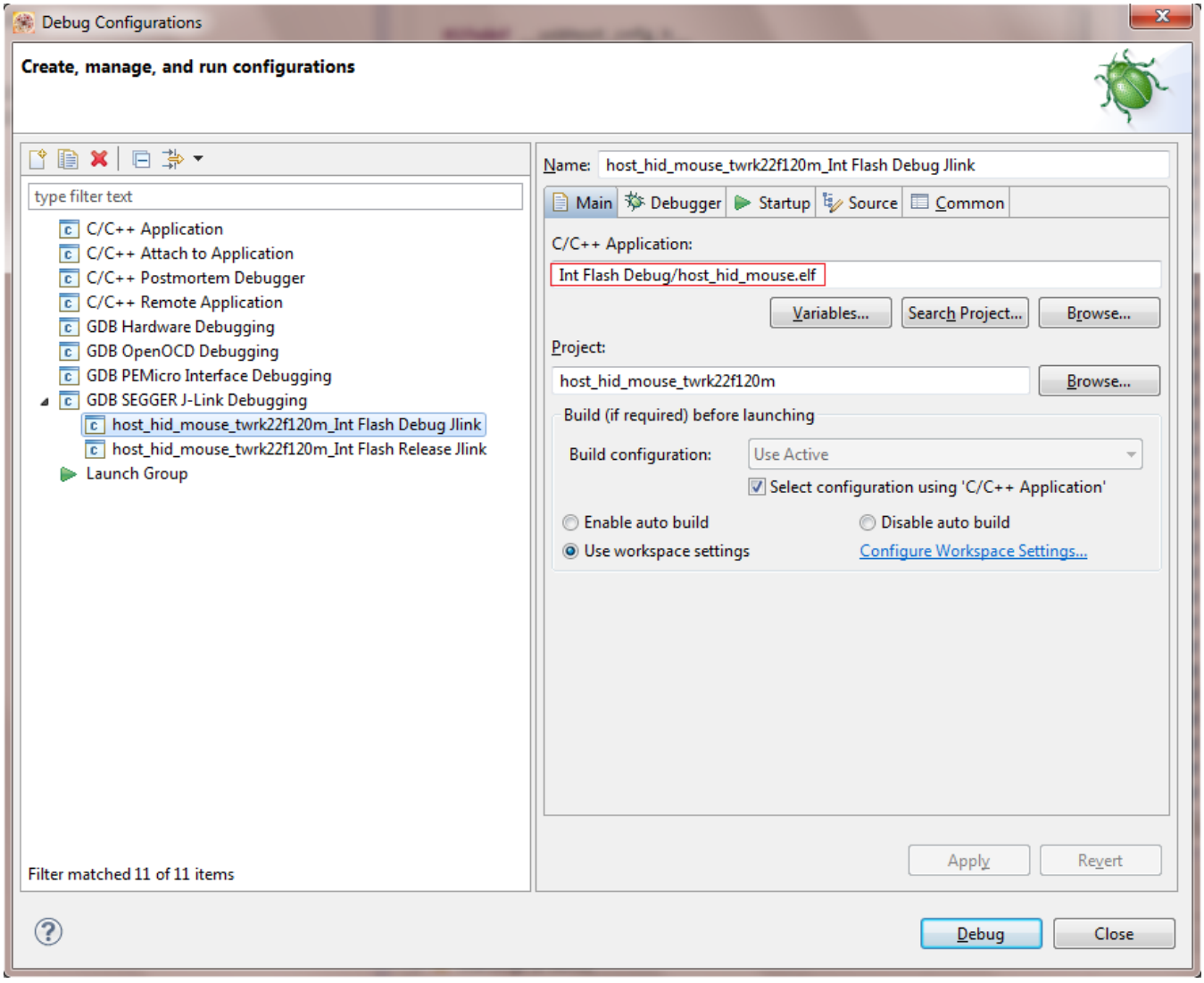

Figure-16 Kinetis Design Studio Debug configurations

11. The application is downloaded to the target and automatically run to main().

12. Run the code by clicking the "**Resume**" button to start the application

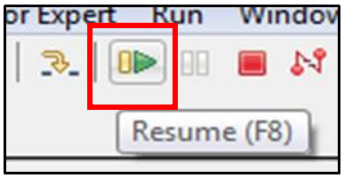

Figure-17 Resume button

4.4 Step-by-step guide for the DS5

1. Same with the Kinetis Design Studio, the DS5 also doesn't have a workspace. As a result, create a workspace and import Kinetis Design Studio USB examples, platform libraries, and the USB stack library. We use twrvf65gs10_a5 as example for DS5 build and download.
2. Select "**File**" then "**Import**" from the DS5 menu.
3. Expand the General folder and select "**Existing Projects into Workspace**". Then, click the "**Next**" button

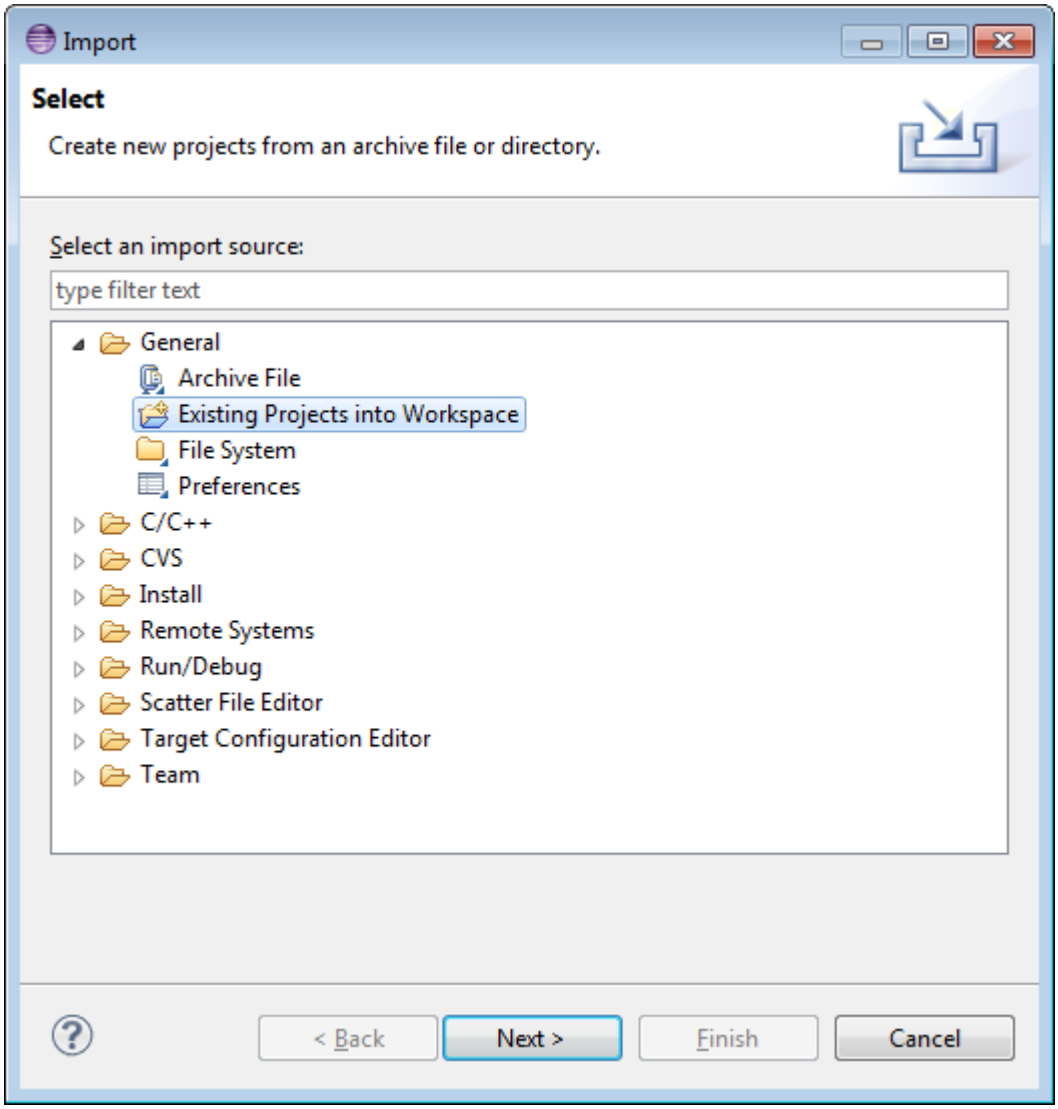

Figure 18 Import projects

4. Point the KDS IDE to the **bsp_twrvf65gs10_a5** in **<install_dir>/mqx/build/ds5**. The import projects directory selection window should resemble this figure

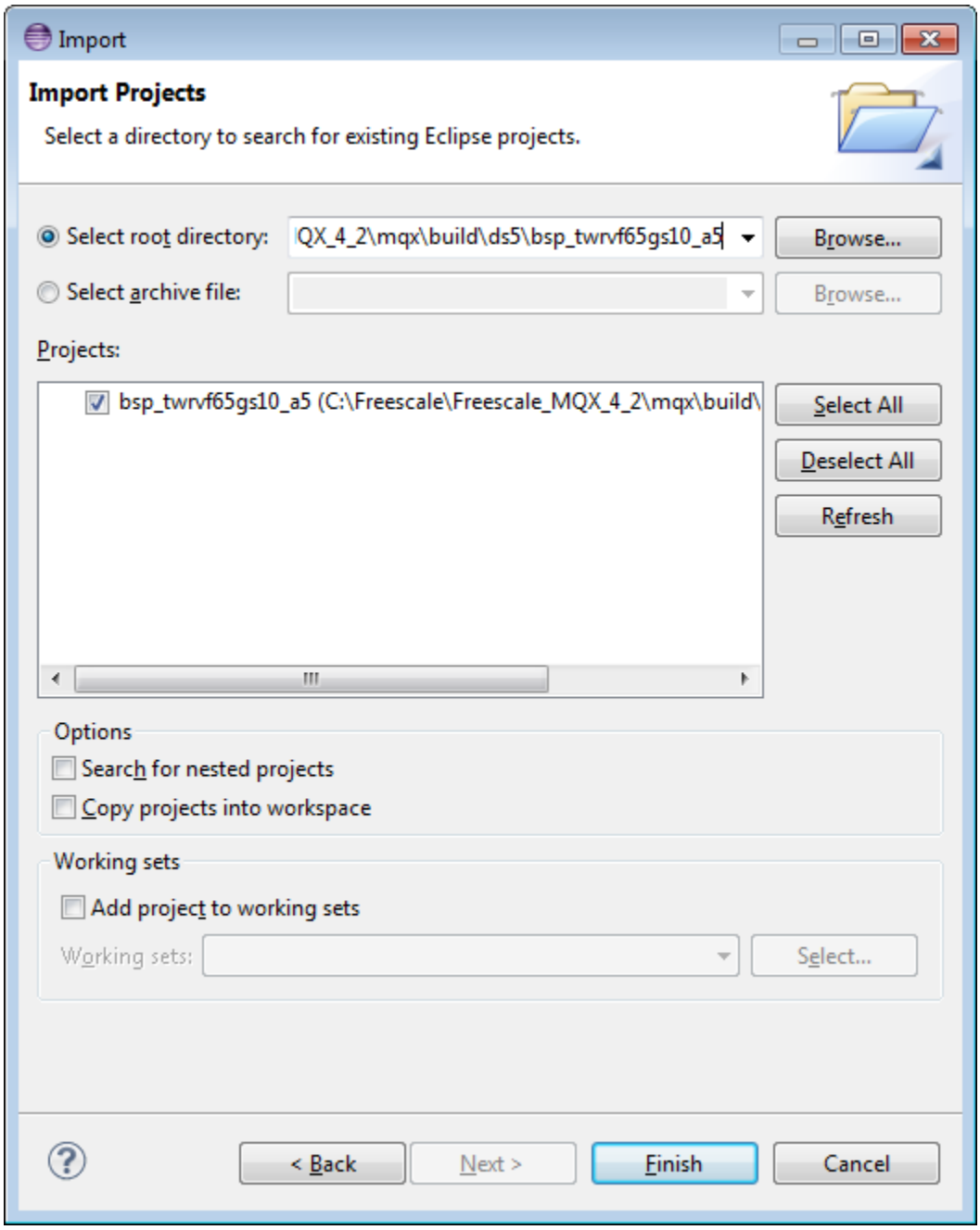

Figure 19 Import Eclipse projects

5. Following the same step to import the USB host library and the USB example, after importing them, the window should like this

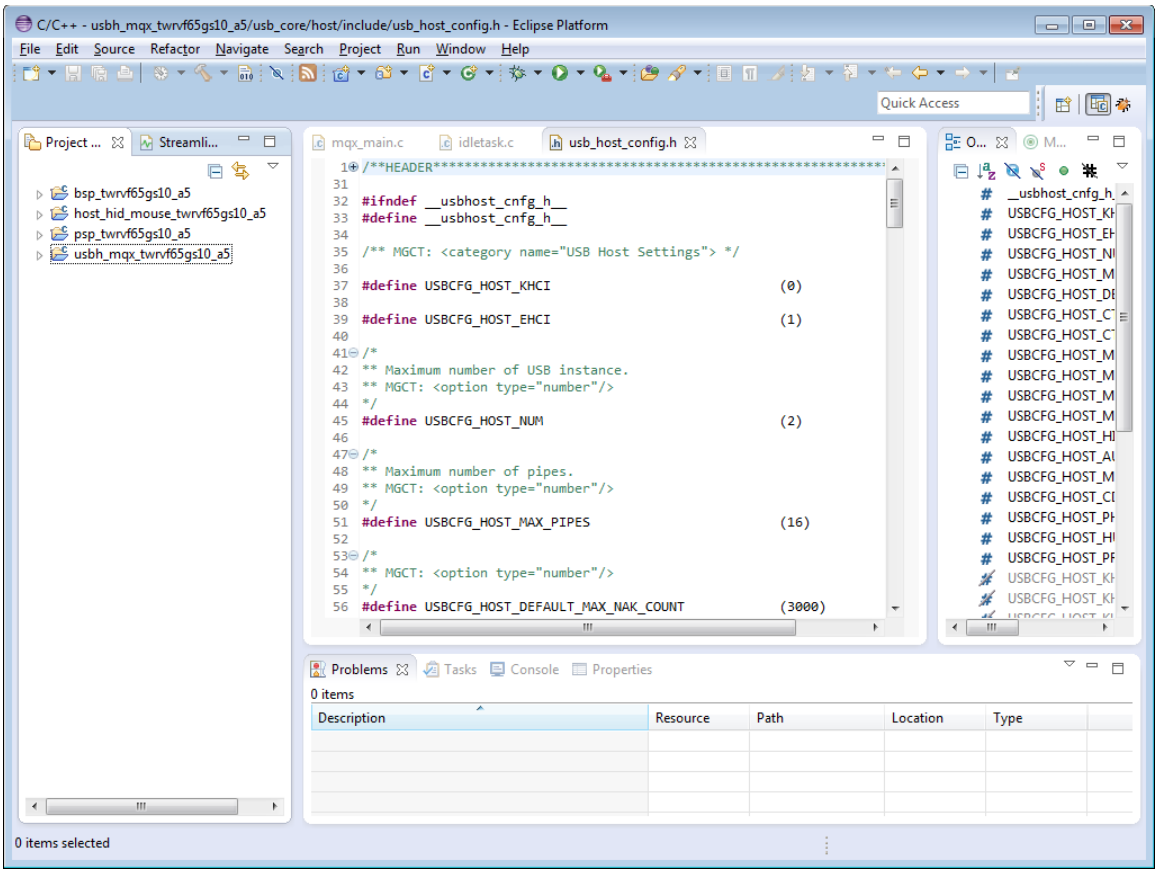

Figure 20 Importing projects

6. Choose the appropriate build target: "**Debug**" or "**Release**" by left-clicking the arrow next to the hammer icon as shown here.

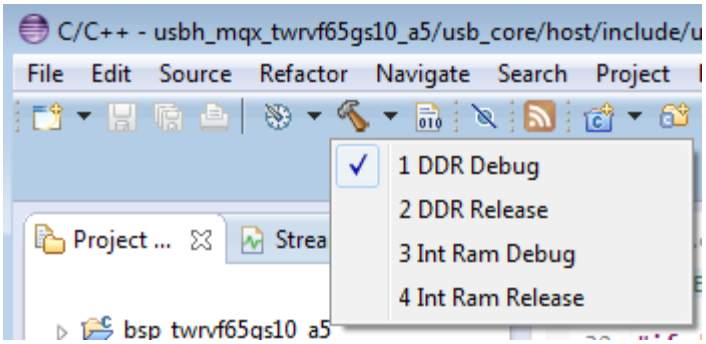

Figure 21 Debug

7. If the library build does not begin after selecting the desired target, left-click the hammer icon to start the build.
8. Following the same step to build the **bsp_ twrvf65gs10_a5** library, **psp_ twrvf65gs10_a5** library the **usbh_mqx_ twrvf65gs10_a5** library and the **host_hid_mouse_ twrvf65gs10_a5** example.

9. To check the debugger configurations, click the down arrow next to the green debug button and select "**Debug Configurations**".

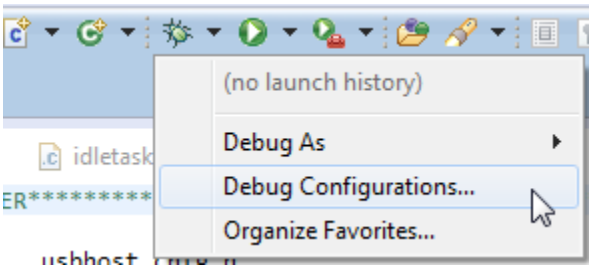

Figure 22 Debug configurations

10. After verifying that the debugger configurations are correct, click on the "**Debug**" button.

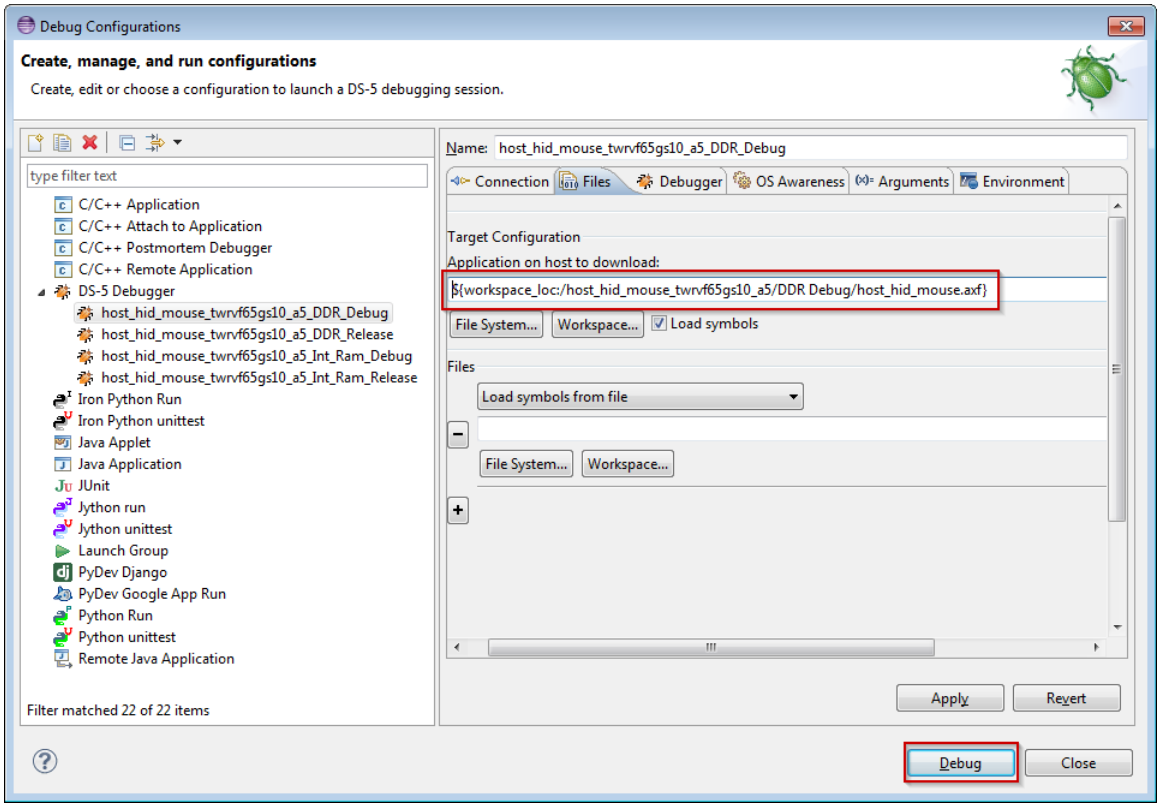

Figure 23 Debug button

11. The application is downloaded to the target and automatically run to main().
12. Run the code by clicking the "**Resume**" button to start the application

4.5 Step-by-step guide for the ARM GCC and KDS IDE GCC

4.5.1 Setup tool chains

4.5.1.1 Install GCC ARM Embedded tool chain

Download and install the installer from www.launchpad.net/gcc-arm-embedded.

4.5.1.2 Install MinGW

1. Download the latest mingw-get-setup.exe.
2. Install the GCC ARM Embedded toolchain. The recommended path is C:/MINGW, however, you may install to any location. Note that the installation path may not contain a space.
3. Ensure that the mingw32-base and msys-base are selected under Basic Setup.
4. Finally, click "**Installation**" and "**Apply changes**".

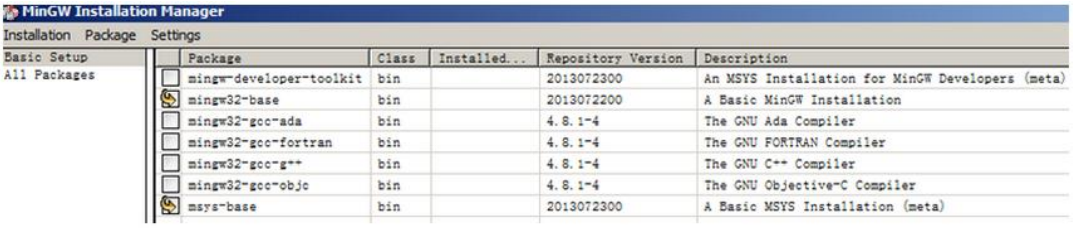

Figure 24 Setup MinGW and MSYS

5. Add paths C:/MINGW/msys/1.0/bin;C:/MINGW/bin to the system environment. Note that if the GCC aRM Embedded tool chain was installed somewhere other than the recommended location, the system paths added should reflect this change. An example using the recommended installation locations are shown below.

NOTE

There is a high chance that, if the paths are not set correctly, the tool chain will not work properly.

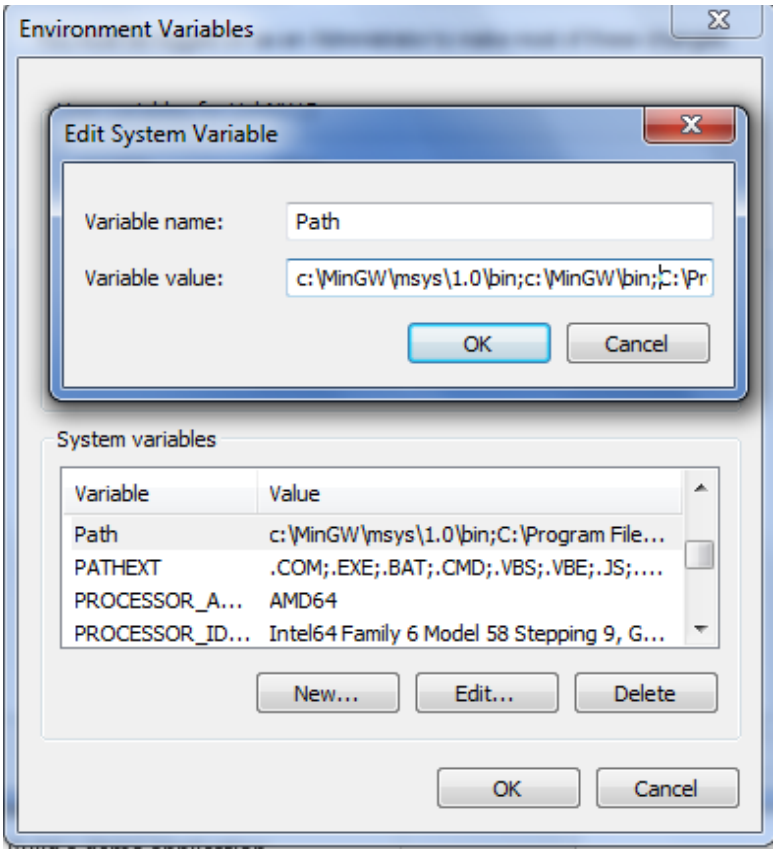

Figure 25 Add Path to systems environment

4.5.1.3 Add new system environment variable ARMGCC_DIR

Create a new system environment variable ARMGCC_DIR. The value of this variable should be the short name of the ARM GCC Embedded tool chain installation path.

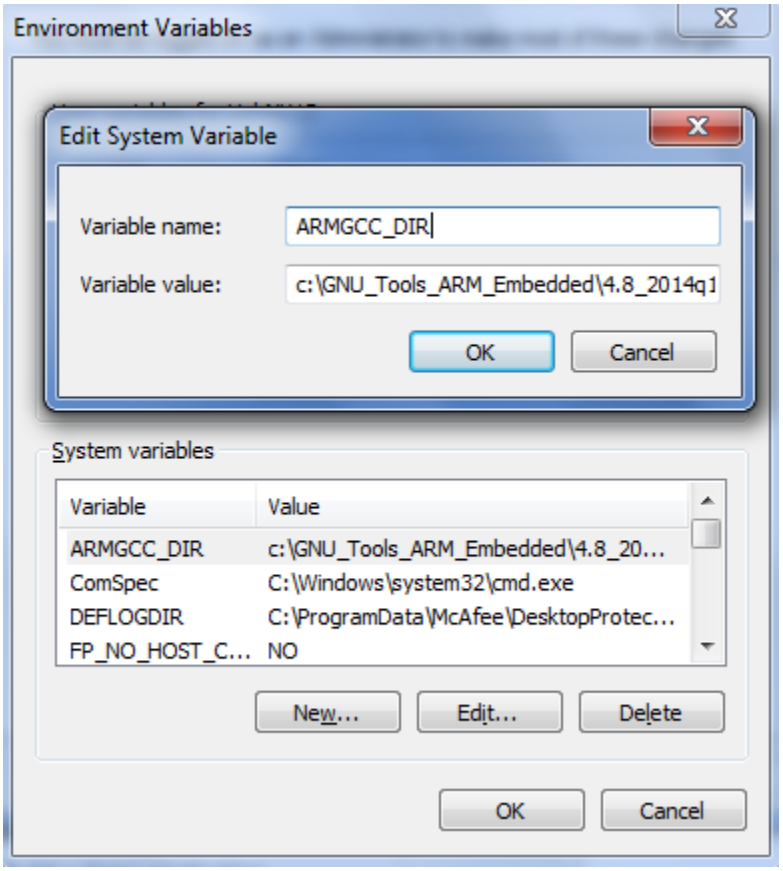

Figure 26 Add ARMGCC_DIR system variable

4.5.1.4 Add new system environment variable KDSGCC_DIR

Create a new system environment variable KDSGCC_DIR. The value for the variable is the name of the Kinetis Design Studio (KDS) IDE ARM GCC installation path. By default, KDS IDE is installed to the C:/Freescale/KDS_1.1.0/toolchain location.

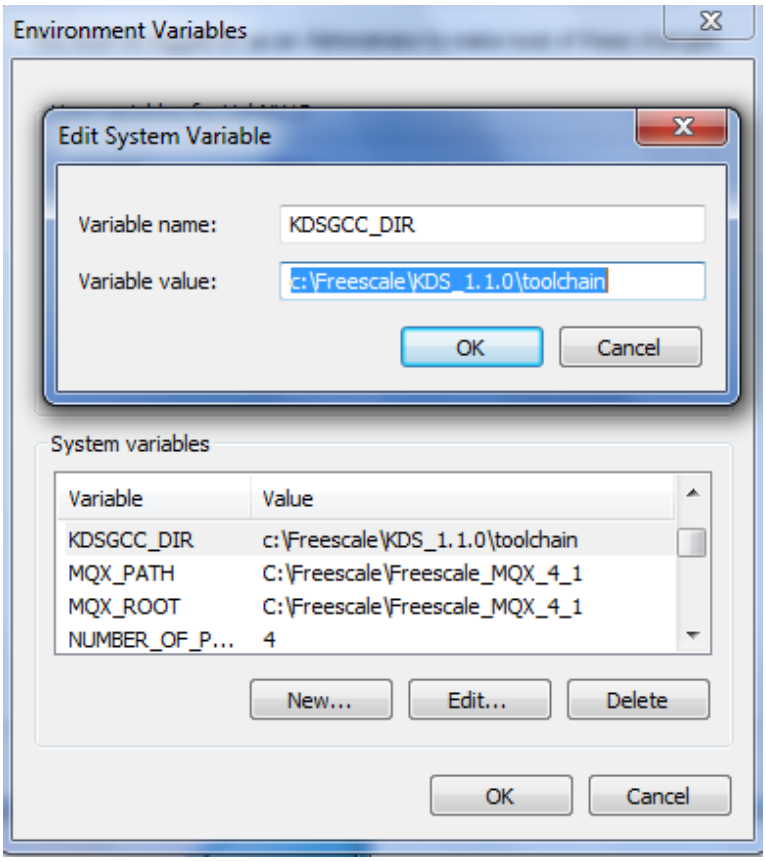

Figure 27 Add KDSGCC_DIR in system variable

4.5.2 Set the ARM Gcc Tool Chain Environment

The TOOLCHAIN_ROOTDIR and the GCC_VERSION should be unmasked and set to the correct value. These variables are in **<install_dir>/usb_v2/build/common/make/global.mak**. For example:

```
ifeq ($(TOOL),gcc_arm)

GCC_VERSION ?=4.8.4

TOOLCHAIN_ROOTDIR = C:/PROGRA~1/GNUTOO~1/43F2B~1.720

endif
```

4.5.3 Build the MQX RTOS library

To build the platform library, follow these instructions:

- Open a GCC ARM Embedded tool chain command window.
- Change the directory of the command window to the MQX RTOS **bsp** and **psp** lib directory in the MQX RTOS4.2 (one of these):

  `<install_dir>/mqx/build/make/bsp_twrk22f120m/`

  `<install_dir>/mqx/build/make/psp_twrk22f120m/`
- Run "**build_gcc_arm.bat**"
- The mqx library is generated in these directories according to the build target.

4.5.4 Build the USB host/device library

- Change the directory to the project directory

  `<install_dir>/usb_v2/usb_core/host/build/make/usbh_mqx_twrk22f120m`
- Run "**build_gcc_arm.bat**"

4.5.5 Build the USB demo

- Change the directory to the project directory:

  `<install_dir>/usb_v2/example/host/hid/mouse/mqx/make/host_hid_mouse_twrk22f120m`
- Run "**build_gcc_arm.bat**"

4.5.6 Run a demo application

This section describes steps to run a demo application using J-Link GDB Server application.

1. Connect the J-Link debug pod to the SWD/JTAG connector of the board.
2. Open the J-Link GDB Server application and modify your connection settings as shown in this figure.

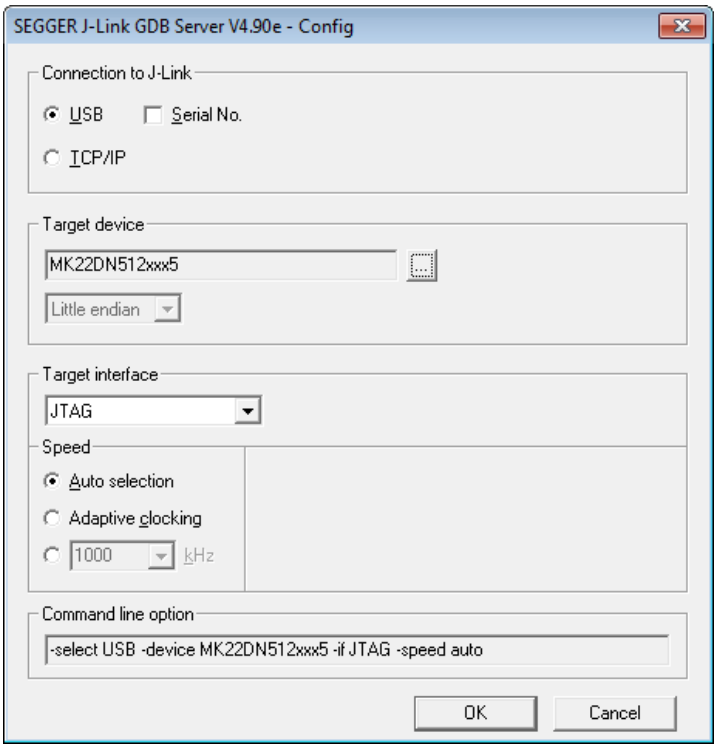

Figure 28 SEGGER J-Link GDB Server configuration

3. Once connected, the screen should resemble this figure:

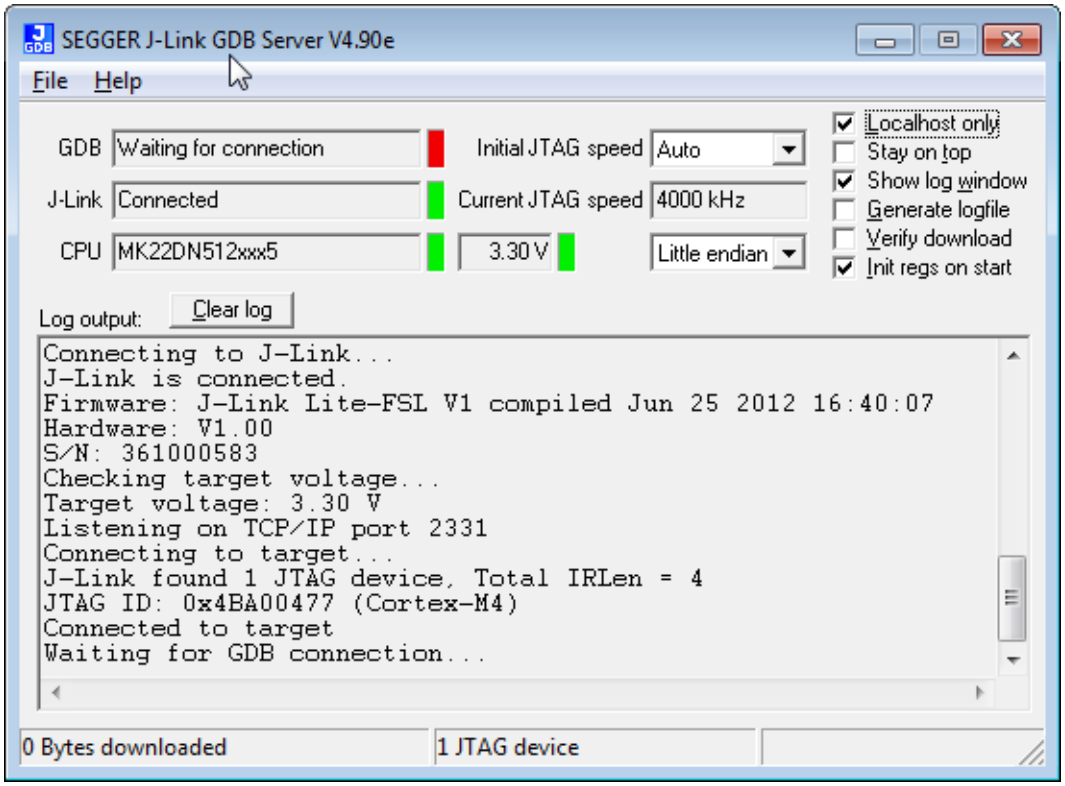

Figure 29 SEGGER J-Link GDB Server screen after successful connection

4. Open the ARM GCC command prompt and change the directory to the output directory of the desired demo. For this example, the directory is:

   `<install_dir>/usb_v2/example/host/hid/mouse/mqx/make/host_hid_mouse_twrk22f120m`

5. Run the command “arm-none-eabi-gdb.exe <DEMO_NAME>.elf”. For this example, it is “arm-none-eabi-gdb.exe host_hid_mouse_twrk22f120m.elf”.

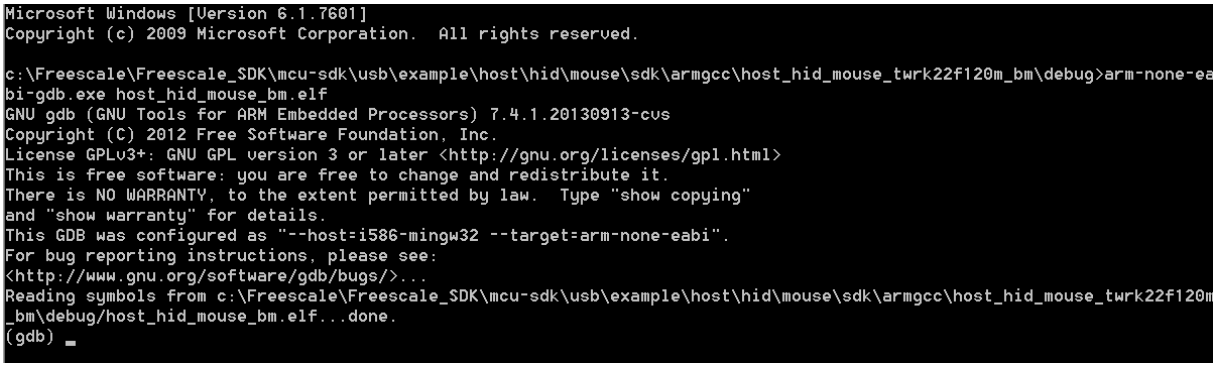

Figure 30 Run arm-none-eabi-gdb

6. Run these commands:
   a. “target remote localhost: 2331”
   b. “monitor reset”
   c. “monitor halt”
   d. “load”
   e. “monitor reset”
7. The application is downloaded and connected. Execute the “monitor go” command to start the demo application.

5 USB Stack Configuration

5.1 Device configuration

All device configurations are listed in this file:

```
<install_dir>/usb_core/device/include/BOARD_NAME/usb_device_config.h
```

Replace BOARD_NAME with the name of the board.

This file is used to either enable or disable the USB class driver. The object number is configurable either to decrease the memory usage or to meet specific requirements.

If the device stack configuration is changed, rebuild both the USB library and the example projects.

NOTE

The composite device examples works only with this setting:

```
USBCFG_DEV_COMPOSITE            1
```

All the other non-composite device examples work only with this setting:

```
USBCFG_DEV_COMPOSITE            0
```

If incorrect settings are configured, a build error occurs.

5.2 Host configuration

All the host configurations are listed in this file:

```
<install_dir>/usb_core/host/include/BOARD_NAME/usb_host_config.h
```

Replace BOARD_NAME with the name of the board.

This file is used to either enable or disable the USB class driver. The object number is configurable either to decrease the memory usage or to meet specific requirements.

If the device stack configuration is changed, rebuild both the USB library and the example projects.

NOTE

Micro and mini receptacles are available for the TWR-K22F120M Tower System module if the TWR-SER and elevator are used. Configure the software and hardware to switch between the two USB receptacles.

- To use the micro receptacle on the TWR-K22F120M Tower System module, the jumper settings should be (for both device and host):
    - J4 1-2
    - J27 2-3 (for rev. A)
    - J27 1-2 (for rev. B)

If the host stack is used, the additional configuration is needed:

USBCFG_HOST_PORT_NATIVE 1

- To use the mini receptacle on the TWR-SER Tower System module, the jumper settings should be (for both device and host):
    - J4 1-2
    - J27 1-2 (for rev. A)
    - J27 2-3 (for rev. B)
    - See the appropriate TWR-SER user's guide for the jumper settings on TWR-SER Tower System module.

If the host stack is used, the additional configuration is needed:

USBCFG_HOST_PORT_NATIVE 0

Additional configurations are not needed for the device because switching between the two USB receptacles doesn't require changing code in the device mode.

5.3 OTG configuration

All OTG configurations are listed in these files:

```
<install_dir>/usb_core/device/include/twrk22f120m/usb_device_config.h
<install_dir>/usb_core/host/include/twrk22f120m/usb_host_config.h
```

These files either enable or disable the USB class driver. The object number is configurable either to decrease the memory usage or to meet specific requirements.

If the OTG stack configuration is changed, rebuild both USB library and example projects.

NOTE

The OTG example for the TWR-K22F120M Tower System module requires the mini receptacle on the TWR-SER Tower System module. The jumper settings should be:

- J4 1-2
- J27 1-2 (for rev. A)
- J27 2-3 (for rev. B)
- See the appropriate TWR-SER user's guide for the jumper settings on the TWR-SER Tower System module.

The additional configuration is needed for the host mode:

```
USBCFG_HOST_PORT_NATIVE      0
```

The additional configuration is needed for the device mode:

```
USBCFG_DEV_COMPOSITE             0
```

NOTE

1. If the USB mini port (J14) on the serial board needs to be used as a device connector on the K64 Tower System module, J19 must be set to 2-3.

2. Because the K64_USB_DP and K64_USB_DN are not connected to the elevator micro USB port on the TWR-K64F120M Tower System module, the R522 and the R523 are not placed and only the micro USB port can be used.

3. If the TWRK-K64 Tower System module is a USB device when no OpenSDA power is supplied, the jumpers need to be set up like this:

   J29, 5-6
   J19, 2-3
   J18, 2-3.

4. If the Freescale Freedom FRDM-K64F is a USB device when no OpenSDA power is supplied, add a 0-ohm resistor on R61 to power on the P3V3_SDA.

Notice:

- Khci host can't work on HSRUN mode on TWRK65 platform. Because if K65 SYSCLOCK run in 180M, USB module can't get the 48M clock.

- Please change the MQX_ENABLE_HSRUN MACRO from 1 to 0 for KHCI host test .
-
- If you run the USB example on High speed mode, please make sure the HIGH_SPEED MACRO is set to 1, and example running in FULL speed mode, the MACRO should be set to 0.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
www.freescale.com/support

Document Number: USBKSDKUG
Rev. 1
04/2015