

Integration of the USB Stack and Kinetis SDK

1 Read Me First

This document describes how to compile the USB stack and examples, download a binary image and run the examples. This document also provides the board-specific information for the TWR-K64F120M Tower System module and the FRDM-K64F board.

Contents

1	Read Me First	1
2	Release Scope	2
3	Requirements for Building USB Examples	3
4	USB Code Structure	5
5	Compiling or Running the USB Stack and Examples	6
6	Integrate USB stack with other RTOS	12
7	USB Stack Configuration	12
8	Revision History	13

2 Release Scope

2.1 Hardware

- Supports TWR-K64F120M and FRDM-K64F.

2.2 Software

- TWR-K64F120M/ FRDM-K64F have HID mouse host/device and CDC device examples.
- Supports Bare-Metal.
- Other RTOS are supported on the TWR-K64F.

3 Requirements for Building USB Examples

3.1 Hardware

- TWR-K64F120M/FRDM-K64F
- (Optional) TWR-SER and Elevator
- J-Link debugger
- USB cables

3.2 Software

- Freescale Kinetis SDK release package
- IAR Embedded Workbench for ARM Version 6.70.3, available for Kinetis devices

3.3 Board Jumper Settings

This document focuses on the USB-related jumper settings on the board. For the other jumper settings, refer to the board-related user guide.

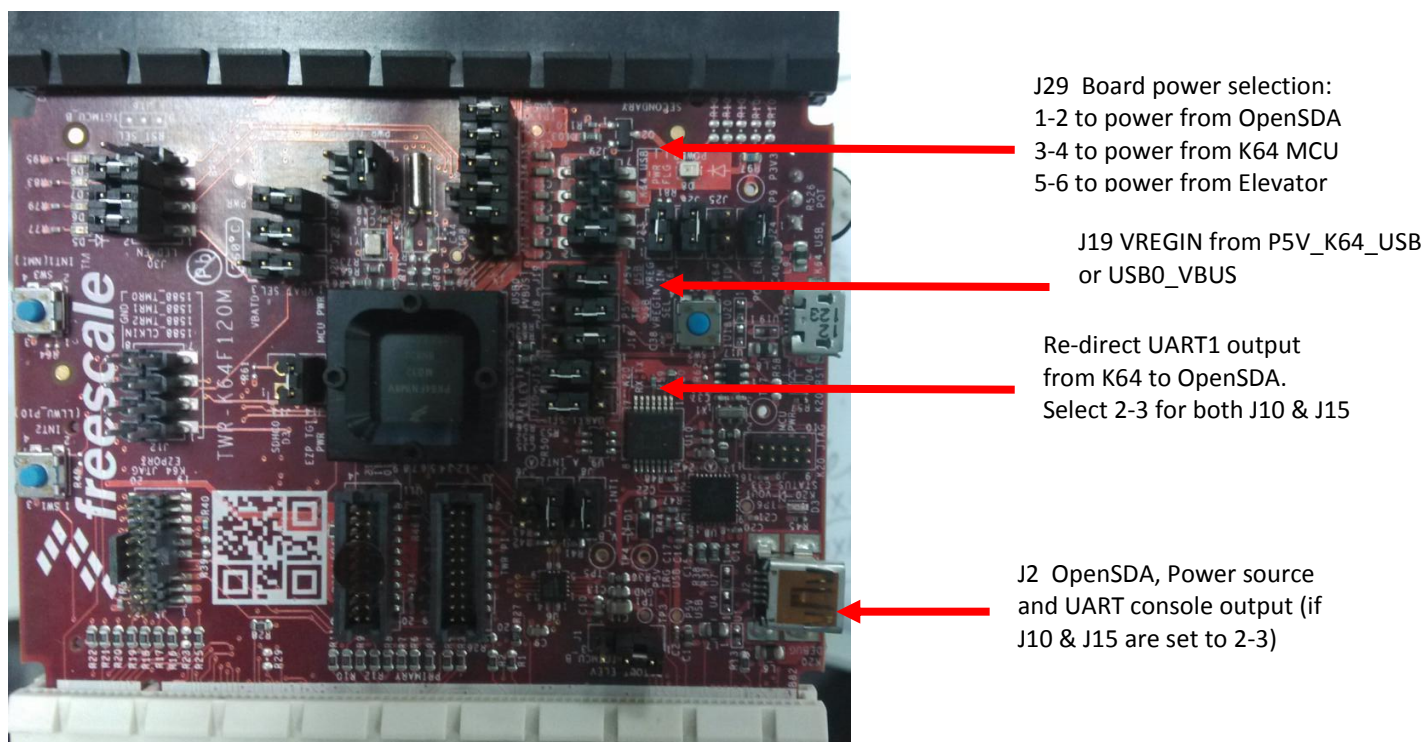
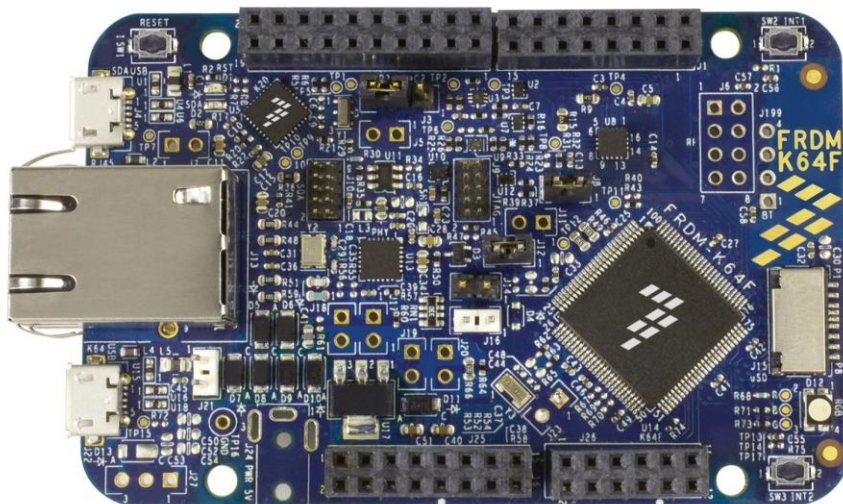


Figure-1 TWR-K64



- Connect J21
- Install a 2.2uF capacitor in the position C33 as shown.

Figure-2 FRDM-K64F

4 USB Code Structure

The USB code is located in usb folder in the root level of the Kinetis_SDK folder.

Name	Date modified	Type	Size
apps	3/8/2014 7:24 AM	File folder	
boards	3/8/2014 7:24 AM	File folder	
doc	3/8/2014 7:24 AM	File folder	
lib	3/8/2014 7:24 AM	File folder	
mk	3/8/2014 7:24 AM	File folder	
platform	3/8/2014 7:24 AM	File folder	
rtos	3/8/2014 7:24 AM	File folder	
usb	3/11/2014 12:48 PM	File folder	

Figure-3 Kinetis SDK Folder Structure

The usb folder includes the source code, projects, and tools. There are three subfolders:

Name	Date modified	Type	Size
adapter	3/8/2014 7:24 AM	File folder	
example	3/8/2014 7:24 AM	File folder	
output	3/11/2014 12:48 PM	File folder	
usb_core	3/8/2014 7:24 AM	File folder	

Figure-4 USB Folder Structure

- adapter
It includes the adapter files which can make the USB stack run on a different RTOS with the same USB core code.
- example
It includes all the source code and project files for the USB examples.
- usb_core
It includes the USB source files, such as HAL, controller driver, and class drivers. It also includes the USB library projects.

5 Compiling or Running the USB Stack and Examples

5.1 Step-by-Step Guide for IAR

This section shows how to use IAR. Open IAR is shown here:

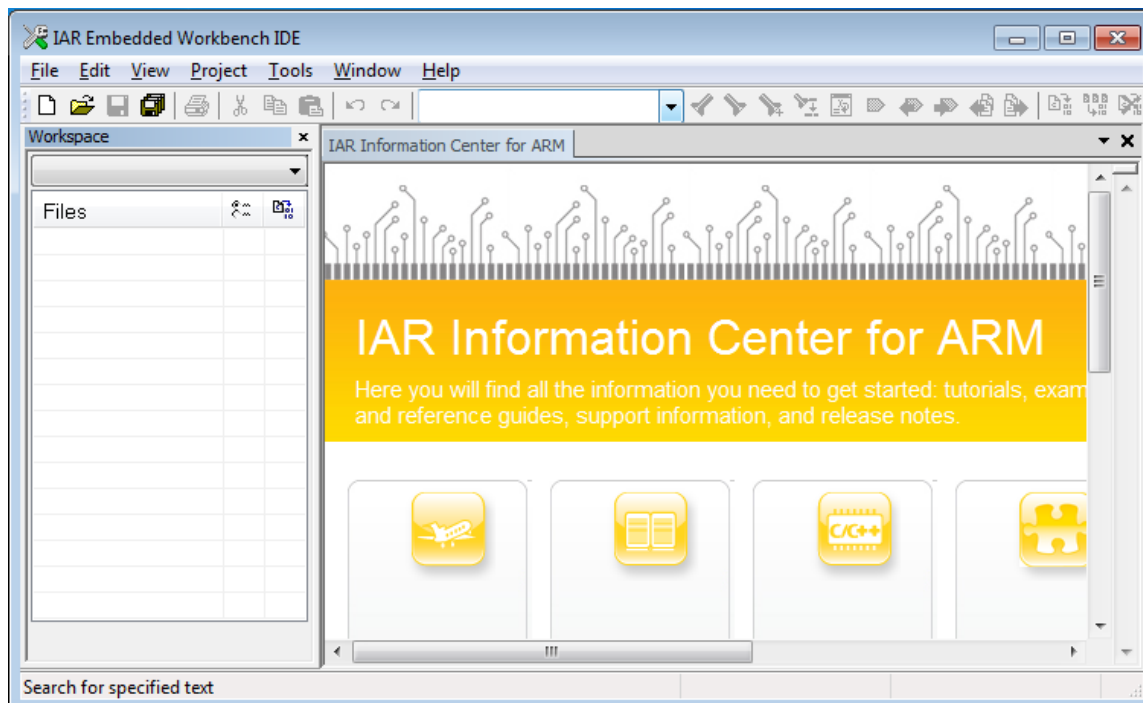


Figure-5 IAR

1. Open the workspace corresponding to different chips.

You can find the corresponding IAR workspace files supported in this release in these paths:

usb\example\host\hid\mouse\sdk\iar\ host_hid_mouse_twrk64f120m_bm\

hid_mouse_host_twrk64f120m.eww

usb\example\host\hid\mouse\sdk\iar\ host_hid_mouse_frdmk64f120m_bm \

host_hid_mouse_frdmk64f120m_bm.eww

usb\example\device\hid\hid_mouse\sdk\iar\dev_hid_mouse_twrk64f120m_bm\

dev_hid_mouse_twrk64f120m.eww

usb\example\device\hid\hid_mouse\sdk\iar\dev_hid_mouse_frdm64f120m_bm\

dev_hid_mouse_frdm64f120m.eww

usb\example\device\cdc\virtual_com\sdk\iar\dev_cdc_virtual_com_frdmk64f120m_bm\

dev_cdc_virtual_com_frdmk64f120m.eww

usb\example\device\cdc\virtual_com\sdk\iar\dev_cdc_virtual_com_twrk64f120m_bm\

dev_cdc_virtual_com_twrk64f120m_bm.eww

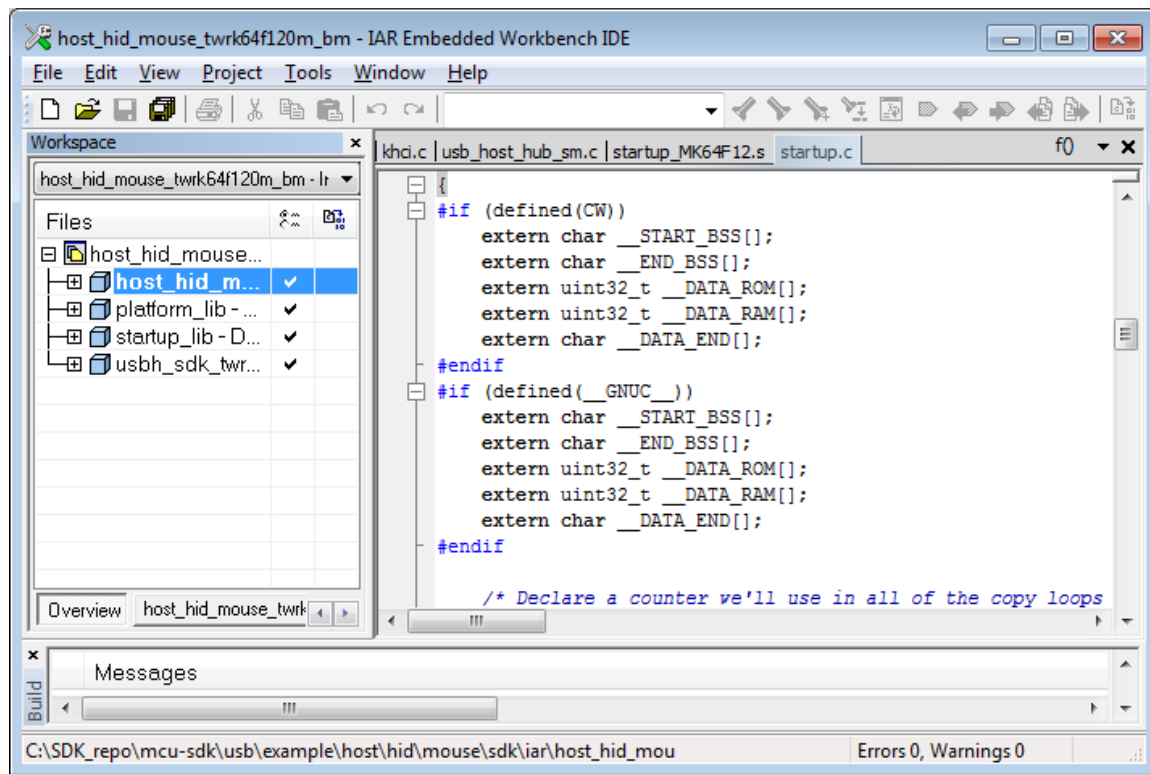


Figure-6 Workspace

This is a step that must be completed to integrate the USB stack into the SDK. Otherwise, the USB example does not run.

2. Change the HEAP size in the corresponding linker files:
K64: platform\linker\iar\K64F12\K64FN1Mxxx12_flash.icf
From
define symbol __ICFEDIT_size_heap__ = 0x2000;
To
define symbol __ICFEDIT_size_heap__ = 0x4000;
3. Build the platform_lib library.

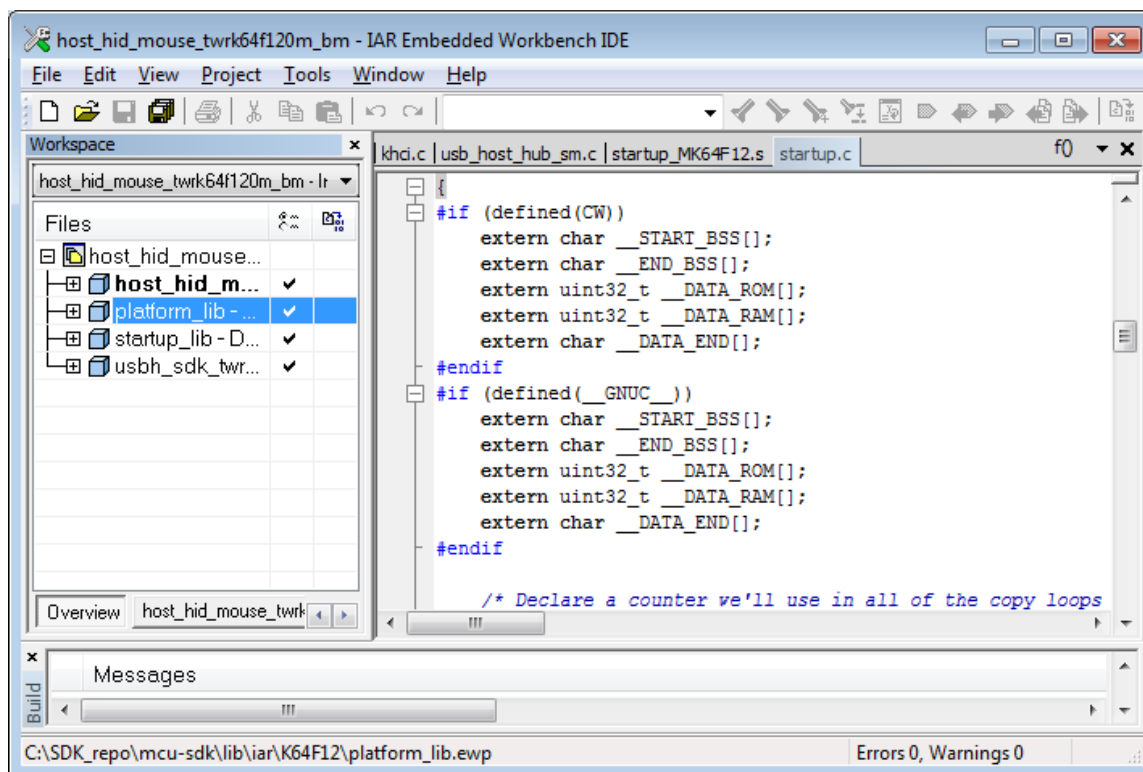


Figure-7 Build the platform library project

4. Build the startup_lib library.

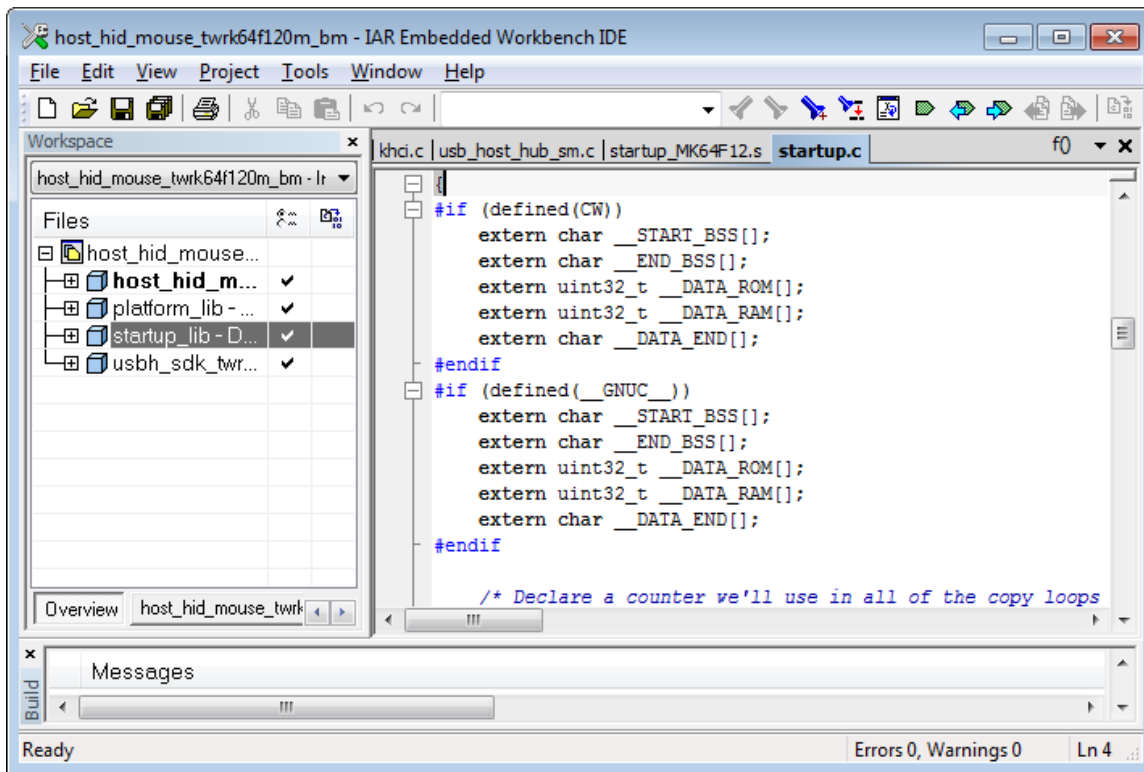


Figure-8 Build the startup library project

5. Build the USB stack.

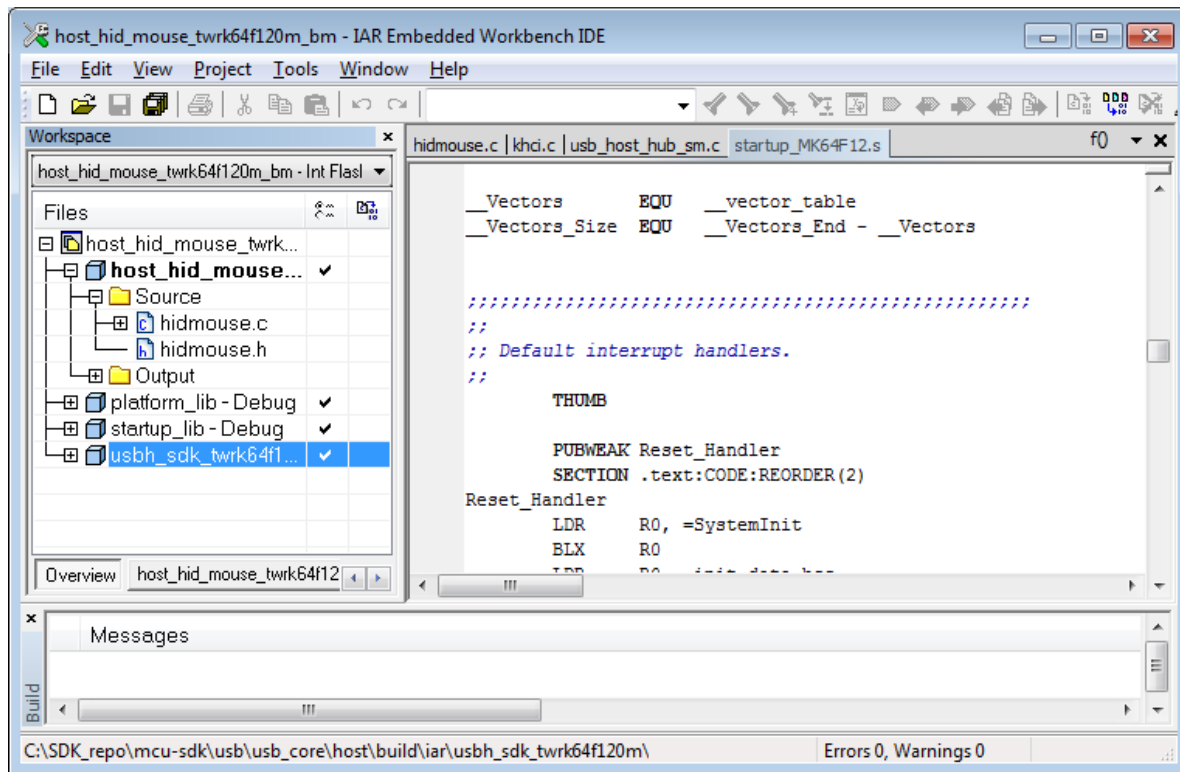


Figure-9 Build the USB stack project

6. Check the results of building the USB library.

After the USB library is built, you can find the generated library binary file (usbh.a) under output\twrk64f120m.iar\debug\usbh\sdk\.

In addition, all the USB-related public header files that may be used by the user are copied to this folder.

7. Build the HID mouse example.

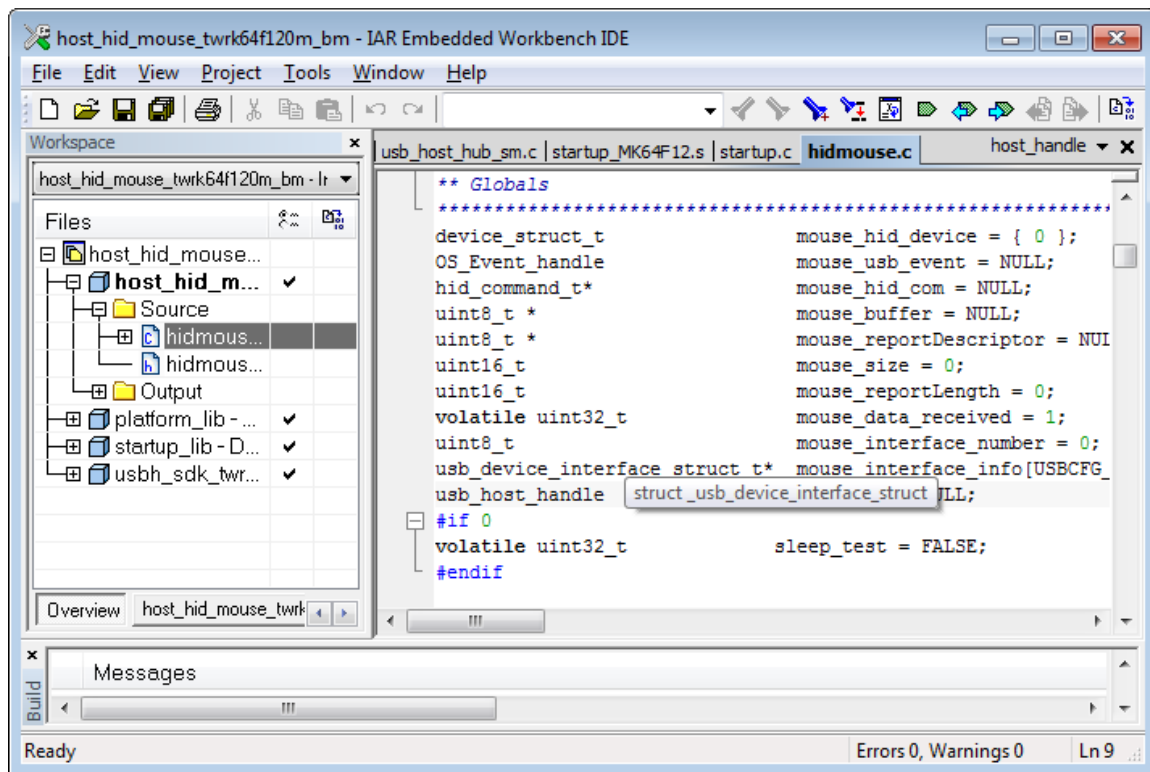


Figure-10 Build the HID mouse project

8. After all building steps are finished, run the examples.
 - Connect the J-Link to the JTAG port J5 on the TWR-K64F120M.
 - Connect the micro USB cable from the PC to J2 on the TWR-K64F120M to power the board.
 - Click Download and Debug. Wait for the downloading to be finished.
 - Click Go to run the example.
 - Connect the micro USB cable from the mouse to J17 on the TWR-K64F120M to verify that the mouse is working on the platform.
 - If it is a CDC virtual_com project, please see [usb\example\device\cdc\virtual_com\readme.pdf](#) for more details.

Note

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

9. After the mouse is attached, the movement of the mouse is displayed on the COM port.
For the TWR-K64F120M, the default UART output is from J2 on the TWR-K64F120M board.

6 Integrate USB stack with other RTOS

The USB workspaces for FreeRTOS, Freescale MQX™ RTOS, μ C/OS II, and μ C/OS III are in these locations:

- `usb\example\host\hid\mouse\sd\iar\ host_hid_mouse_twrk64f120m_freertos\host_hid_mouse_twrk64f120m_freertos.eww`
- `usb\example\host\hid\mouse\sd\iar\ host_hid_mouse_twrk64f120m_mqx\host_hid_mouse_twrk64f120m_mqx.eww`
- `usb\example\host\hid\mouse\sd\iar\ host_hid_mouse_twrk64f120m_ucosii\host_hid_mouse_twrk64f120m_ucosii.eww`
- `usb\example\host\hid\mouse\sd\iar\ host_hid_mouse_twrk64f120m_ucosiii\host_hid_mouse_twrk64f120m_ucosiii.eww`

7 USB Stack Configuration

7.1 USB Device Configuration

All USB device configurations are listed in this file:

```
usb_core\device\include\MK64F120M\usb_device_config.h
```

The USB class driver can be enabled or disabled by using this file. Additionally, the object number can be configured either to decrease the memory usage or to increase the object number to meet specific requirements.

If you change the configuration of the USB device stack, both the USB library project and the example project need to be rebuilt.

7.2 USB Host Configuration

All USB host configurations are listed in this file:

```
usb_core\host\include\MK64F120M\usb_host_config.h
```

The USB class driver can be enabled or disabled by using this file. Additionally, the object number can be configured either to decrease the memory usage or to increase the object number to meet specific requirements.

If you change the configuration of the USB host stack, both the USB library project and the example project need to be rebuilt.

Note

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, R522 and R523 are not placed and only the micro USB port can be used.

If the TWRK-K64 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.

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

8 Revision History

Revision/Date	Description
1.0.0-beta 2014	Initial release

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

www.freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Tower is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The ARM Powered Logo is a trademark of ARM Limited.

©2014 Freescale Semiconductor, Inc.

