

# Freescale BM USB Stack Porting New Platform User's Guide

## 1 Read Me First

This document provides the detailed steps to port the USB Unify Stack to a new platform. There are two main steps for porting:

- Porting the sources code
- Porting examples

### Contents

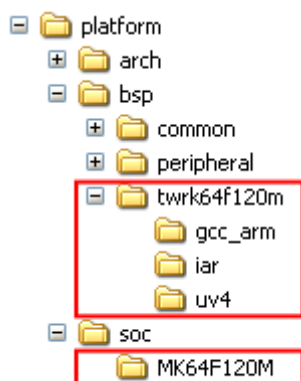
<b>1</b>	<b>Read Me First</b>	<b>1</b>
<b>2</b>	<b>Porting the Source Code</b>	<b>2</b>
<b>3</b>	<b>Porting Examples</b>	<b>6</b>



## 2 Porting the Source Code

### 2.1 Porting platform files

The following figure shows the folder structure and all the source files that need to be ported after this step. The folders marked with red borders contain source files that need to be ported. The platform directory is `/Src/platform`.



The following are the files that you need to port in each folder. The other files are common and not need to be ported.

- BSP code

Port the following files in `.../Src/platform/bsp/{BOARD_NAME}`:

- Linker file for each mode: device, host, OTG and for IDEs  
These files are in `.../Src/platform/bsp/{BOARD_NAME}/{IDE_NAME}`
- Vector table and startup code for each IDE: IAR, Keil, gcc\_arm.  
These files are in `.../Src/platform/bsp/{BOARD_NAME}/{IDE_NAME}`
- `bsp.h`: defines the system clock (SYS\_CLK) and SCI channel (SCI\_CHANNEL) used for this platform.
- `derivative.h`: includes the derivative-specific header.
- `init_gpio.c`: GPIO functions to initialize IO of necessary peripherals such as SCI.

- SOC code:

Port the following files in `/Src/platform/soc/{PLATFORM_NAME}`:

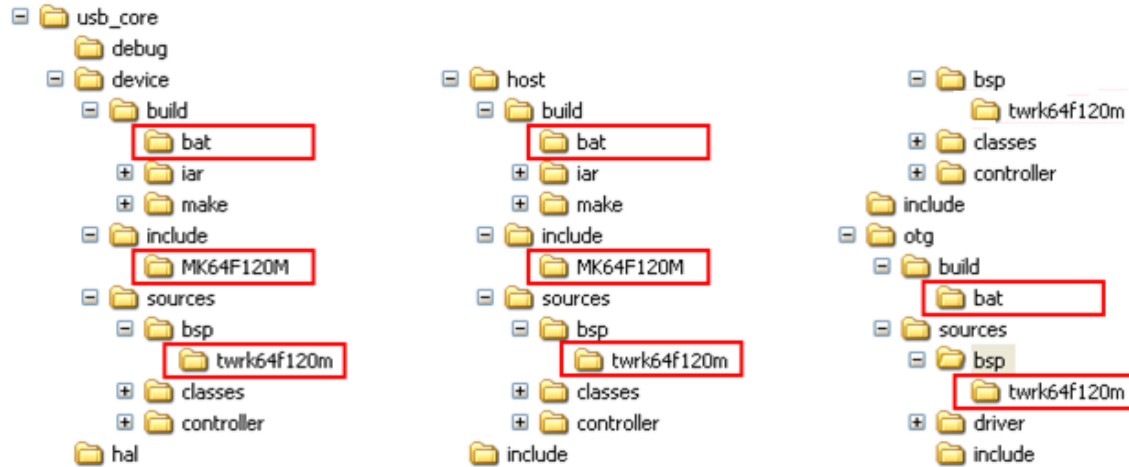
- SOC header files
  - Copies the header file for platform to `.../Src/platform/soc/{PLATFORM_NAME}` and modify it (if needed).

- Defines the platform core in the begin file, such as:
  - CPU\_IS\_ARM\_CORTEX\_M4 is 1 in MK64F12.h
- soc\_config.h: defines endianness of platform.
- soc\_init.c: contains the SYS\_Init() function to switch the chip to generate the system and bus desire clock. This function is called from the main function.
- soc\_isr.c and soc\_isr.h:
  - Port the function soc\_install\_isr(): This function is used to install the interrupt functions and parameter for each interrupt vector used in all the examples. You need to modify this function to install the interrupt vector of a new platform and write the interrupt function for each interrupt vector based on the format of the old platform.
- soc\_usb.c: This file contains the functions:
  - soc\_get\_usb\_base\_address(): gets the USB base address of the platform
  - soc\_get\_usb\_vector\_number(): gets the USB vector number
- Peripheral files: USB and other peripheral drivers such as SCI, I2C, and PIT. You have to decide which peripheral you need to port first. The following table shows the peripherals used in the existing examples of the USB Unify Stack.

Peripherals	Examples	Examples use peripheral interrupt
I2C	Audio Speaker Device HID Mouse OTG	-
I2S	Audio Speaker Device	Audio Speaker Device
PIT	All Host and OTG examples Audio Speaker Generator	All Host and OTG examples
SCI	All Host and OTG examples	CDC Host

## 2.2 Porting USB driver

The following figure shows the folder structure and all the files that need to be ported after this step. The folders marked with red borders contain the files that need to be ported. The directory of usb\_core is `.../USB_Unify_Stack/Src/usb_core`.



### 2.2.1 Device

- BSP code
  - Port `/Src/usb_core/device/sources/bsp/{BOARD_NAME}/usb_dev_bsp.c`. This file contains the functions:
    - `bsp_usb_dev_io_init()`: initializes GPIO and generates 48 MHz clock for the USB module
    - `bsp_usb_dev_init()`: initializes the register in the USB module
- Configuration file
  - Defines the configuration for this platform in `/Src/usb_core/device/include/{BOARD_NAME}/usb_device_config.h`.
- .bat file
  - Creates a bat file for this platform in `/Src/usb_core/device/build/bat/`, which is used to copy the header files to the output folder

### 2.2.2 Host

- BSP code
  - Port `/Src/usb_core/host/sources/bsp/{BOARD_NAME}/usb_host_bsp.c`. This file contains the functions:
    - `bsp_usb_host_io_init()`: initializes GPIO and generate the 48 MHz clock for the USB module

- `bsp_usb_host_init()`: initializes the register in the USB module
- Configuration file
  - Define the configuration for this platform in  
`/Src/usb_core/host/include/{BOARD_NAME}/usb_host_config.h`
- .bat file
  - Create a bat file for this platform in `/Src/usb_core/host/build/bat/`, which is used to copy the header files to the output folder

### 2.2.3 OTG

- BSP code
  - Port `/Src/usb_core/otg/sources/bsp/{BOARD_NAME}/usb_otg_bsp.c`. This file contains the functions:
    - `bsp_usb_host_io_init()`: initializes GPIO and generate the 48 MHz clock for the USB module
    - `bsp_usb_host_init()`: initializes the register in the USB module
- .bat file
  - Create a bat file for this platform in `/Src/usb_core/otg/build/bat/`, which is used to copy the header files to the output folder

## 2.3 Creating a library project

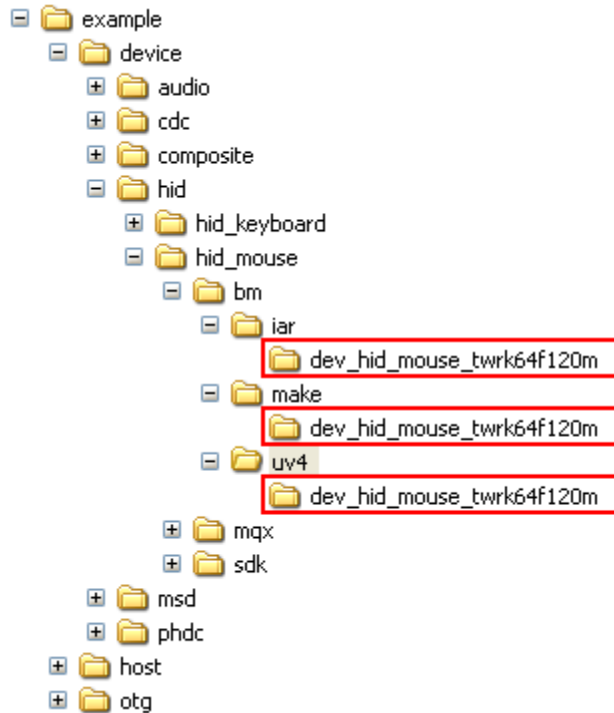
Create a new library project for the new platform. For example, the library file for twrk64f120m is kept in:

- `/Src/usb_core/device/build/iar/usbd_bm_twrk64f120m`
- `/Src/usb_core/host/build/iar/usbh_bm_twrk64f120m`
- `/Src/usb_core/otg/build/iar/usbotg_bm_twrk64f120m`

## 3 Porting Examples

### 3.1 Creating an example project

List all the examples for each IDE that you want to port for new platform. The project file will be kept in the following folder structure:



The steps for creating a new project are as follows:

1. Create a new empty project.
2. Choose a device.
3. Add all the source code into project based on the existing platform.
4. Change the linker file.
5. Change the include path.
6. Change the debug configuration.

### 3.2 Porting user\_config.h

After creating all the examples, you need to create and define the specific configuration in the `user_config.h` file for the new platform in each project directory. The macros that need to be defined are:

- `PRINTF_ENABLE`: use/not use SCI module to print to terminal.
- `DELAY_ENABLE`: use/not use PIT module to delay.

- TASK\_ENABLE: use/not use Periodic Task
- DEVICE\_APP: define new application is device application or not. This define can change the heap and stack size of the new application.
- I2C\_ENABLE: use/not use the I2C module.
- I2S\_ENABLE: use/not use the I2S module.
- PORTx\_ENABLE: use/not use the interrupt for port x.

#### **How to Reach Us:**

##### **Home Page:**

[www.freescale.com](http://www.freescale.com)

##### **Web Support:**

[www.freescale.com/support](http://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](http://freescale.com/SalesTermsandConditions).

Freescale, Kinetis, and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. 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.