Freescale Semiconductor

Document Number: USBTWRK24F120MUG Rev. 1.0, 08/2014

User's Guide

Freescale MQX USB Stack for TWR-K24F120M GA User's Guide

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 related to TWR-K24F120M.

Contents

1	Read Me First	1
2	Requirements for Building USB Examples	2
3	USB Code Structure	4
4	Compiling or Running the USB Stack and Example	s 5
5	USB Stack Configuration	19





© Freescale Semiconductor, Inc., 2014. All rights reserved.

2 Requirements for Building USB Examples

2.1 Hardware

- TWR-K24F120M board
- J-Link debugger
- USB cables

2.2 Software

- Freescale MQXTM RTOS for the TWR-K24F120M release package
- IAR Embedded Workbench for ARM Version 6.70.1, available for Kinetis devices
- Keil μVision5 Integrated Development Environment Version 5.0.5.15, available for Kinetis ARM[®] CortexM4 devices
- Keil.Kinetis_K20_DFP.1.0.2.pack for TWR-K24F120M which is available in <install_dir>/tools/keil_extensions/uVision4 in this release package
- GNU Tools for ARM Embedded Processors 4.7 2013Q3
- MinGW v3.82.90 with mingw32-base and msys-base packages installed
- Segger J-Link driver V4.88 or later

2.3 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 Board jumper settings

• J26 1-2: enables 5V VBUS

3 USB Code Structure

The USB stack is located in the usb_v2 subfolder of the MQX RTOS root folder. There are five subfolders in it:

🚞 adapter	File Folder	3/27/2014 1:33 PM
🚞 build	File Folder	3/27/2014 1:33 PM
🚞 example	File Folder	3/27/2014 1:33 PM
🚞 output	File Folder	3/27/2014 1:33 PM
🚞 usb_core	File Folder	3/27/2014 1:32 PM

Figure-2 usb_v2 folder structure

• adapter

Includes the adapter files that allow the USB stack to run on different RTOSs with the same USB core code.

• build

Includes the GCC make files.

• example

Includes all the source code and project files of the USB examples.

• output

The USB library binary file is generated into this folder and all the USB-related public header files are copied to this folder. The examples need to include one folder as the including path in the example project settings.

• usb_core

Includes the USB source files, such as HAL, controller driver, and class drivers. It also includes the USB library projects.

4 Compiling or Running the USB Stack and Examples

4.1 Step by step guide for IAR

This section takes IAR as an example to show how to build examples. The other tool chains have similar steps.

1. Open IAR as follows.



Figure-3 IAR

- 2. Add MQX RTOS bsp and psp projects by clicking **Project** → **Add Existing Project**. You can find the corresponding IAR project files in the following paths:
 - bsp <install_dir>/mqx/build/iar/bsp_twrk24f120m
 - o psp
 <install_dir>/mqx/build/iar/psp_twrk24f120m



Figure-4 bsp and psp projects

- 3. Add a USB stack library project in the following paths:
 - USB Device Stack

<install_dir>/usb_v2/usb_core/device/build/iar/usbd_mqx_twrk24f120m

• USB Host Stack

 $<\!\!install_dir\!>\!\!/usb_v2/usb_core/host/build/iar/usbh_mqx_twrk24f120m$

• USB OTG Stack

<install_dir>/usb_v2/usb_core/otg/build/iar/usbotg_mqx_twrk24f120m



Figure-5 Folder name of USB device stack

4. Add a USB example project.

All the USB examples are located in the example folder. The folder structure is as follows.



This guide adds the USB HID mouse device example.

Freescale MQX USB Stack for TWR-K24F120M GA User's Guide, Rev. 1.0, 08/2014

💥 IAR Embedded Workbench IDE	
<u>File Edit View Project CMSIS-DAP Tools Window Help</u>	
	* * 注 図 ● ● ● 個 時 時 時 8 2 上 上
Workspace	TAR Information Center for ARM
dev_hid_mouse_twrk24f120m - Int Flash Debug	
Files & Bi	
E Nutitled Workspace *	المارين في المحافظ ا
Heref 1 bsp twrk24f120m - Debug ✓	
📕 🕂 🗇 dev_hid_mouse_twrk24f120m - Int Flash Debug 🗸 🗸	
P → P psp_twrk24f120m - Debug	IAP Information Contor for APM
I I I usbd_mqx_twrk24f120m - Debug ✓	
	Here you will find all the information you need to get started: tutorials, examp
	and reference guides, support information, and release notes.
Overview bsp_twrk24f120m dev_hid_mouse_twrk24f120m psp_twrk24f120m	GETTING STARTED USER GUIDES EXAMPLE PROJECTS INTEGRA
Log	
Thu Jul 10, 2014 17:53:11: Loading the I-jet/JTAGjet driver	
8	
1 Brie	
Det	
Ready	NUM 🔜 🔬

Figure-7 Adding a USB example project

5. Build the bsp and psp libraries.

Save the workspace in the resulting dialog box and proceed with the builds.

Save Workspace As		Sec.	×
🕒 🖉 🖉 🖉 🖉	d ▶ iar ▶ bsp_twrk24f120m ▶	arch bsp_twrk24f120m	Q
Organize 🔻 New	folder		(?)
Downloads	 Name Debug 	Date modified 7/9/2014 10:42 AM	Type File folder
 Libraries Documents Git Music Pictures Videos 	Belease Bettings ■	7/9/2014 2:45 PM 7/9/2014 10:43 AM	File folder File folder
Computer	 ✓ 		4
File <u>n</u> ame: Save as <u>t</u> ype: V	Vorkspace Files (*.eww)		•
Hide Folders		<u>S</u> ave Can	cel

Figure-8	Save	Workspace	As	dialog	box
		pass		ananeg	~~~

6. Build a USB stack library.

IAR Embedded Workbench IDE	IS-DAP Tools Window Help	nati Wa	
D 🚅 🛛 🗗 😹 🖻 🕯			' '> 'x 'X 🔄 > # # # @ = @ 😲 % 🕭 🕭 🕭
Workspace		,	IAR Information Center for ARM
dev_hid_mouse_twrk24f120m · Int Fla	ash Debug	•	
Files		8:: 📭	le l
🛛 🖻 Untitled Workspace *			
H⊞ foll bsp_twrk24f120m - Del	bug rk24f120m - Int Flach Debug	ž	
B B B B B B B B B B B B B B B B B B B	bug	2	
usbd_mqx_twrk24f120	m - Debua	~	TAR Information Center for ARIVI
L L I L I L I L I L I L I L I L I L I L	Options	·	Here you will find all the information you need to get started: tutorials, examp
	Make		and reference guides, support information, and release notes.
	Compile		
	Rebuild All		
	Clean		
	Stop Build		
	Add +	<u> </u>	
	Remove		
	Rename		
Overview bsp_twrk24f120m	Version Control System	20m 🕢 🔸	CETTING STARTED USER GUIDES EXAMPLE PROJECTS INTEGRA V
* Log	Open Containing Folder		
Thu Jul 10, 2014 17:53	File Properties		
	Set as Active		
Log			
spug			
Clean and make the selected project	t		NHM

- 7. Check the USB library build result.
 - After the USB library is built, you can find the generated library binary file (usbd.a) under <install_dir>/usb_v2/output/twrk24f120m.iar/debug/usbd/mqx/
 - In addition, all the USB-related public header files are copied to this folder.

8. Build the USB device HID mouse example. The USB library must compile successfully. Otherwise, the build for the example project may fail.

V IAR Embedded Workbench IDE	the last of	
File Edit View Project CMSIS-DAP Tools	Window Help	
	- 4	> > > > = = = = # # # = # = = = = = = =
Workspace	×	TAR Information Center for ARM
dev_hid_mouse_twrk24f120m - Int Flash Debug	• ·	
Files	8:: B	
🗉 🖻 Untitled Workspace *		
—⊕ 🗇 bsp_twrk24f120m - Debug	~	
	Ontions	E
	Options	AR Information Center
📙 🖵 🗇 usbh_mqx_twrk24f120m - Debug	Make	lore you will find all the information you need to get a
	Compile	and reference guides, support information, and relea
	Rebuild All	
	Clean	
	Stop Build	
	Add	• 🐸 🖳 🗒
	Remove	
	Rename	
Overview bsp_twrk24f120m dev_hid_mouse_tv	Version Control System	
×	Open Containing Folder	
Thu Jul 10, 2014 17:53:11: Loading the	File Properties	
	Set as Active	
Log		
puda -		
		F
Clean and make the selected project		4

Figure-9 Building the USB device HID mouse example

- 9. Connect the J-Link to the JTAG port (J32) on TWR-K24F120M.
- 10. Connect the micro USB cable from a computer to J37 of TWR-K24F120M to power on the board.
- 11. Click **Download and Debug**. Wait for the downloading to finish.
- 12. Click **Go** to run the example.
- 13. Connect the micro USB cable from a computer to the J23 port of TRW-K24F120M to enable the USB mouse device to work on the computer.

After the mouse device is enumerated by the computer, the mouse will be active, and the mouse pointer draws a rectangle on the computer.

4.2 Additional actions for Keil

The compilation process for Keil is similar to that for IAR. This section focuses on the parts of the Keil downloading process that are different from IAR.

Before we can download the binary to the target board with Keil, we need to set a programming algorithm as follows.

1. Access the options for the target project by right-clicking the target project.

Options for Target 'hwtimer_twrk24f120m Int Flash Rel	ease'
Device Target Output Listing User C/C++ Asm	Linker Debug Utilities
Freescale MK24FN256xx12	
Xtal (MHz): 120.0	Code Generation
Operating system: None	Use Cross-Module Optimization
System-Viewer File (Sfr)	Use MicroLIB Big Endian
MK24F25612 svd	Floating Point Hardware: Use FPU
Use Custom SVD File	,
Pand /Only Marries Arres	Deed With Marray Area
Read/Only Memory Areas	Read/write Memory Areas
	I RAM1:
□ ROM2: O	□ RAM2: □
□ ROM3: ○	RAM3:
on-chip	on-chip
IROM1: 0x0 0x40000 (€	IRAM1: 0x20000000 0x30000 □
IROM2:	□ IRAM2: 0x1FFF0000 0x10000 □
	Incel Defaults Help

Figure-10 Options for the target project

2. Click the **Debug** tab.

Options for Target 'hwtimer_twrk24f120m Int Flash Rel	lease'
Device Target Output Listing User C/C++ Asm	Linker Debug Utilities
○ Use Simulator Settings □ Limit Speed to Real-Time	⊡se: I-LINK / J-TRACE Cortex Settings S
Load Application at Startup Run to main() Initialization File: Edit	Load Application at Startup Run to main() Initialization File:
Restore Debug Session Settings	Restore Debug Session Settings
CPU DLL: Parameter: SARMCM3.DLL -REMAP	Driver DLL: Parameter: SARMCM3.DLL -REMAP
Dialog DLL: Parameter: DCM.DLL -pCM4	Dialog DLL: Parameter: TCM.DLL -pCM4
ОК Саг	ncel Defaults Help

Figure-11 Debug tab

3. Click Setting next to the Use: J-Link/J-Trace Cortex option. The Cortex JLink/JTrace Target Driver Setup dialog box appears.

Cortex JLink/JTrace Target Driver Setup	
Debug Trace Flash Download	
J-Link / J-Trace Adapter	SW Device
SN: 1651344405	IDCODE Device Name Move
Device: J-Link Lite-FSL	SWD
HW : V1.00 dll : V4.84c	Down
PW: J-Link Lite-FSL VI compiled Ji Port: Max Clock: SW V 2MHz V Auto Clk	Automatic Detection ID CODE: Manual Configuration Device Name: Add Delete Update IR Ien: Cache Options Download Options
Connect: Normal Reset: Nor	mal
Interface TCP/IP USB © TCP/IP Scan State: ready	Port (Auto: 0) Autodetect JLink Info 0 0 1 ?
	OK Cancel Apply

Figure-12 Cortex JLink/JTrace Target Driver Setup dialog box

4. Click the **Flash Download** tab.

Cortex JLink/JTrace Target Driver S	etup		X
Debug Trace Flash Download			
Download Function C Erase Full Chip C Erase Sectors C Do not Erase	✓ Program ✓ Verify ✓ Reset and Run	AM for Algorithm	Size: 0x1000
Programming Algorithm			
Description	Device Size Device 1	Type Addres	ss Range
		Start:	Size:
	Add	emove	
		ОК	Cancel <u>Apply</u>

Figure-13 Cortex JLink/JTrace Target Driver Setup dialog box - Flash Download tab

Description	Flash Size	Device Type	Origin	
IKXX 256kB PFlash SEC(4	256k	On-chip Flash	Device Family Package	
DuCM360 128kB Flash	128k	On-chip Flash	MDK Core	
PC18xx/43xx S25FL032 SP	4M	Ext. Flash SPI	MDK Core	
C28F640J3x Dual Flash	16M	Ext. Flash 32-bit	MDK Core	
29GL064N Dual Flash	16M	Ext. Flash 32-bit	MDK Core	
\Keil\ARM\PACK\Keil\Kinetis_	K20_DFP\1.0	.3\Flash\MKP256_4	KB_SECTOR.FLM	

5. Click Add and select MKXX 256kB PFlash SEC(4KB).

Figure-14 Selecting MKxxN 256kB programming flash

4.3 Downloading GNU tools ARM embedded 4.7

The compilation process of GNU Tools ARM Embedded is similar to that of IAR and Keil. You need to access the corresponding folder and run mingw32-make to compile the project from the command line or just run the corresponding batch file build_gcc_arm.bat for each example.

Note: The makefile provided by the USB stack supposes that the GCC tool chain is installed in C:/PROGRA~1/GNUTOO~1/43F2B~1.720 (the default installation path). If the GCC tool chain is installed in another folder in your system, you need to change GCC_TOOLCHAIN_DIR in build\common\make\global.mak to the correct path manually. Otherwise, the compilation process will fail. In addition, the path should be in the short file name format. You can get the short file name by using the following command:

```
for %A in ("C:\Program Files\GNU Tools ARM Embedded\4.7 2013q3") do @echo %~sA
```

The string C:\Program Files\GNU Tools ARM Embedded\4.7 2013q3 in the command above should be replaced by the correct target long file name.

Some strange issues may occur when the default installation path of the GCC Tool Chain is changed, hence it is recommended not to change it. In addition, make sure that the **mingw32-base** and **msys-base** packages are installed in your system and the corresponding path has been added into the system path (MINGW\bin), and MINGW\msys\1.0\bin is not added to the system path.

The downloading steps are as follows:

1. Run J-Link GDB Server.

This application is installed along with the J-Link. Select MK24FN256xxx12 as the target device and click **OK**.

SEGGER J-Link GDB Server ¥4.88 - Config		
Connection to J-Link		
Target device		
MK24FN256xxx12		
Little endian		
Target interface		
SWD		
Speed		
C Auto selection		
C Adaptive glocking		
Command line option		
-select USB -device MK24FN256xxx12 -if SWD -speed 1000		
Cancel		

Figure-15 J-Link GDB Server Configuration dialog box

SEGGER J-Link GDB Server ¥4.88		_ _ ×	
<u>File</u> <u>H</u> elp			
GDB Waiting for connection J-Link Connected CPU MK24FN256xxx12	Initial SWD speed 1000 kHz Current SWD speed 1000 kHz 3.30 V	Localhost only Stay on top Show log window Generate logfile Verify download Init regs on start	
Log output: <u>C</u> lear log			
Target interface speed: 1000kHz Target endian: little Connecting to J-Link J-Link is connected. Firmware: J-Link OpenSDA compiled Apr 24 2014 14:43:37 Hardware: V1.00 S/N: 621000000 Checking target voltage Target voltage: 3.30 V Listening on TCP/IP port 2331 Connecting to targetConnected to target Waiting for GDB connection			
0 Bytes downloaded	L JTAG device		

Figure-16 J-Link GDB server configuration result

2. Run **arm-none-eabi-gdb** under the folder where the target binary is located.

In the example, this folder is under the following path:

<install_dir>/usb_v2/example/device/hid/hid_mouse/mqx/make/dev_hid_mouse_twrk24f120m/gcc_arm/intflash_release



Figure-17 arm-none-eabi-gdb folder

3. On the gdb client, run the following commands:

```
target remote localhost:2331
monitor reset
monitor flash device = MK24FN256xxx12
load dev_hid_mouse_twrk24f120m.elf
monitor reg pc = (0x00000004)
monitor reg sp = (0x00000000)
monitor go
```

The mouse becomes active on the computer.

5 USB Stack Configuration

5.1 Device configuration

All the device configurations are listed in the following file:

 $<\!\!install_dir\!>\!\!/usb_v2/usb_core/device/include/twrk24f120m/usb_device_config.h$

We can enable or disable the USB class driver through this file, and we can configure the object number to decrease the memory usage or increase the object number to meet some specific requirements.

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

Notes

1

The composite device examples can work only with:

USBCFG_DEV_COMPOSITE

All other non-composite device examples can work only with:

USBCFG_DEV_COMPOSITE 0

If incorrect settings are configured, a build error will occur and will need to be modified.

5.2 Host configuration

All the host configurations are listed in the following file:

<install_dir>/usb_v2/usb_core/host/include/twrk24f120m/usb_host_config.h

We can enable or disable the USB class driver through this file, and we can configure the object number to decrease the memory usage or increase the object number to meet some specific requirements.

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

5.3 OTG configuration

All the OTG configurations are listed in the following files:

- <install_dir>/usb_v2/usb_core/host/include/twrk24f120m/usb_device_config.h
- <install_dir>/usb_v2/usb_core/host/include/twrk24f120m/usb_host_config.h

You can enable or disable the USB class driver through these files, and configure the object number to decrease the memory usage or increase the object number to meet specific requirements.

If you change the configuration of the OTG stack, both the USB library project and the example project need to be re-built.

NOTE

The OTG example requests to use the mini receptacle on the TWR-SER board. The jumper settings should be:

- J4 1 2
- J27 1 2
- For the jumper settings on the TWR-SER board, see the TWR-SER user's guide.

Additional configuration is needed for the host mode:

USBCFG_HOST_PORT_NATIVE 0

Additional configuration is needed for the device mode:

USBCFG_DEV_COMPOSITE 0

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 solls products pursuant to standard terms and conditions of sale, which can be found at the following address: 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.





Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.



Как с нами связаться

Телефон: 8 (812) 309 58 32 (многоканальный) **Факс:** 8 (812) 320-02-42 **Электронная почта:** <u>org@eplast1.ru</u> **Адрес:** 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.