NXP Semiconductors User's Guide

MCUXpresso USB PD Migration Guide

1. About this document

This document describes how to migrate from the FRDM-KL27Z USB Power Delivery software middleware to SDK v2.2-based platforms.

The migration between the two devices requires software changes and may require some hardware modifications, such as jumper configurations.

Contents

1.	About this document	1
2.	System overview	2
3.	Hardware considerations	2
4.	Software migration	3



2. System overview

The USBPD-C-SHIELD uses Arduino-compatible headers to communicate between the PTN5110 Type-C Port Controller (TCPC) USB Power Delivery PHY and the FRDM-KL27Z working as a Type-C Port Manager (TCPM).



Figure 1. System overview

3. Hardware considerations

The communication between the TPCM MCU and the USB-PD shield requires six Arduino signals. To drive the PTN5110, two pins (D14, D15) are used for the I²C and a third one (D8) as an interrupt-capable input nALERT. There is a second I²C at A4 and A5 (multiplexed with D14 and D15) to provide flexibility on platforms using the I²C instance at D14 and D15.

The TCPM output pin EXTRA_EN_SRC (D4) enables or disables the NX20P5090 unidirectional power switch that provides 9 V to the VBUS. Table 1 shows the pinout for different Kinetis FRDM boards.

Arduino	Name	Shield	FRDM-KL27	FRDM-K64 revE
D4	EXTRA_EN_SRC	J403-05	PTA13	PTB23
D8	nALERT	J404-01	PTE31	PTC12
D14	PTN5110_SDA	J404-09	PTD6 I2C1	PTE25 I2C0
D15	PTN5110_SCL	J404-10	PTD7 I2C1	PTE24 I2C0
A4	SDA	J405-05	PTB1 I2C0	PTC11 I2C1
A5	SCL	J405-04	PTB0 I2C0	PTC10 I2C1

Table 1. US	3 PD-C-SHIELD	pinout
-------------	----------------------	--------

The usb_pd_freertos demo application uses two switches. One controls the power requests and the other switch calls for the power change. Table 2 shows the switch assignment (depending on the development board).

Switch	FRDM-KL27		FRDM-K64	
Power request	SW1	PTA4	SW2	PTC6
Power change	SW3	PTC1	SW3	PTA4

Table 2. Switch assignment

4. Software migration

4.1. FRDM-K64F rev. E

The FRDM-KL27Z USB Power Delivery MCUXpressso SDK middleware software enables you to migrate to other Kinetis platforms. The following subsections show how to enable the USBPD-C-SHIELD on the FRDM-K64F revE platform.

4.1.1. IAR Embedded Workbench® IDE

- 1. Follow the instructions in Appendix A to build an SDK package.
- 2. Follow the instructions in Appendix B to build an SDK package for a selected platform (for example; FRDM-K64F) using the IAR IDE.
- 3. Use the Windows[®] OS Explorer to copy the content from the ...\boards\frdmk64f\project_template\cproject_generator_templates\iar folder.
- 4. Open a second Explorer window and navigate to ...*boards\frdmk64f\usb_examples*. Add a new folder with the name of the USB Power Delivery example project being ported (for example; *usb_pd*).
- 5. Inside the recently-created folder (*usb_pd*), add a new folder named *freertos*.
- 6. Add a new directory inside the *freertos* folder and name it *iar*.
- 7. Paste the files copied in step 2 into the ...*boards\frdmk64f\usb_examples\usb_pd\freertos\iar* folder.
- 8. Replace all *\$[project_name]* file names with *usb_pd_freertos*.



Figure 2. Replacing files

9. Use a text editor to open *usb_pd_freertos.eww*.

MCUXpresso USB PD Migration Guide, Rev. 0, 10/2017

- 10. Inside the text editor, find all "\$[project_name]" strings and replace them with "usb_pd_freertos".
- 11. Copy the board-support SDK template files *board.c*, *board.h*, *clock_config.c*, *clock_config.h*, *pin_mux.c*, and *pin_mux.h*, which can be found in the ...*boards\frdmk64f\project_template* folder.
- 12. In the second Explorer window, navigate to the ...\boards\frdmk64f\usb_examples\usb_pd\freertos folder and paste the files copied in the previous step.
- 13. Copy the FreeRTOS configuration template file *FreeRTOSConfig.h*, which can be found in the*rtos\freertos_9.0.0\template_application\ARM_CM4F* folder.
- 14. Navigate to the ...\boards\frdmk64f\usb_examples\usb_pd\freertos folder and paste the *FreeRTOSConfig.h* file copied in the previous step.
- 15. In the first Explorer window, navigate to the ...\boards\frdmk64f\cmsis_driver_examples\i2c\interrupt_transfer folder and copy the RTE_Device.h file.
- 16. Paste *RTE_Device.h* into the second Explorer window (...\boards\frdmk64f\usb_examples\usb_pd\freertos).
- 17. Obtain the USB Power Delivery source code from the software downloaded in step 1 (MCUXpresso SDK for FRDM-KL27Z with the USB Power Delivery).
 - a. Unzip the FRDM-KL27Z SDK folder.
 - b. Use the Explorer to copy the main.c, pd_app.c, pd_app.h, pd_app_demo.c, pd_command_app.c, pd_command_interface.c, pd_command_interface.h, pd_power_app.c, pd_power_interface.c, pd_power_interface.h, usb_io.h, usb_kinetis_io_drv.h, usb_pd_config.h, and usb_pit_drv.h files located in the ...\boards\frdmkl27z\usb_examples\usb_pd\freertos folder.

🚱 🔍 🖷 🕼 « boards > lpcxpresso54608 > usb_examples > usb_pd > freertos > 🔹 🔹	49 Search freertos	٩
Organize 🕶 📓 Open Burn New folder	33	• 🛯 0
w w besid. besid.		
15 items selected	🚑 Comp	uter

Figure 3. Copying files

- c. Paste the files into the ...\boards\frdmk64f\usb_examples\usb_pd\freertos folder.
- d. To add the Power Delivery middleware support, copy the FRDM-KL27Z *pd* folder located in the ...*middleware\usb_1.7.0* folder.

- e. Insert the copied *pd* folder into the FRDM-K64F middleware software located in the ...*middleware\usb_1.6.3* folder.
- f. Replace the FRDM-K64F *usb_misc.h* file located in the ...*middleware\usb_1.6.3\include* folder with the version found in the FRDM-KL27Z SDK package.
- 18. Follow the steps in Appendix C to add the Arduino signals to the *pin_mux.c* and *pin_mux.h* files using the MCUXpresso Config Tools.
- 19. Using the IAR IDE, open the FRDM-KL27Z *usb_pd_freertos* workspace located in the *...\boards\frdmkl27z\usb_examples\usb_pd\freertos\iar* folder and the recently created FRDM-K64F workspace in the *...\boards\frdmk64f\usb_examples\usb_pd\freertos\iar* folder.
- 20. In the FRDM-K64F workspace, look for the *doc* folder in the "Workspace" window and remove it from the project by right-clicking it and selecting the "Remove" option.



Figure 4. Removing the doc folder

21. Copy the folder structure from the FRDM-KL27Z workspace. To add a new folder, right-click the project name, expand the "Add" menu, and select the "Add Group…" option. In the "Add Group" wizard, write the folder name into the text box and click the "OK" button.



Figure 5. Copying folder structure

22. After adding all the groups, the folder structure should look like this:

🗆 🌒 usb_pd_freertos - Debug *
— 🖬 board
- 🕀 📕 drivers
🕂 📮 💼 freertos
🖵 📠 portable
🖬 sources
- 🕀 💼 startup
🖬 cmsis_wrapper
— 🖬 include
📠 osa
—— 🔳 pd
🖬 ptn5110
🗕 🖿 🖬 utilities
- utility
🖵 🖬 Output

Figure 6. Resulting folder structure

23. Add the preprocessor paths to the workspace. To use the FRDM-KL27Z project options as a reference, navigate to "Project Options..." → "C/C++ Compiler" → "Preprocessor" and copy all the text in the "Additional include directories" field.

Options for node "usb_	pd_fre	ertos"
Options for node "usb_ Calegory: General Options Static Analysis Runtime Checking C/C+++ Compiler Assembler Output Converter Custom Build Build Actions	pd_fre	ertos" Factory Settings Discard Unused Publics Language 2 Code Optimizations Output List Preprocessor Ignore standard include directories Additional include directories: (one per line)
Build Actions Linker Debugger Simulator Angel E CADI		SPROJ_DIR\$/././//inidleware/usb_16.3/pd/pthS110 SPROJ_DIR\$///////devices/MKL27Z644/drivers SPROJ_DIR\$//////devices/MKL27Z644/drivers SPROJ_DIR\$//////devices/MKL27Z644/drivers SPROJ_DIR\$//////devices/MKL27Z644 Preinclude file:
GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro BDT		Defined symbols: (one per line) DEBUG=1 CPU_MKL27Z54VLH4 USB_STACK_FREERTOS_HE USB_STACK_FREERTOS_
ST-LINK Third-Party Driver	~	OK Cancel

Figure 7. Adding preprocessor paths

24. Paste the paths into a text editor and replace all "MKL27Z644" occurrences with "MK64F12". Replace the "ARM_CM0" occurrences with "ARM_CM4F" and copy the resulting text.

<pre>\$PROJ_DIR\$/////rtos/freertos_9.0.0/Source/portable/IAR/ARM_CM4F</pre>
<pre>\$PROJ_DIR\$//////rtos/freertos_9.0.0/Source/include</pre>
<pre>\$PROJ_DIR\$/////CMSIS/Include</pre>
\$PROJ_DIR\$/
<pre>\$PROJ_DIR\$/////CMSIS/Driver/Include</pre>
<pre>\$PROJ_DIR\$//////rtos/freertos_9.0.0/Source</pre>
<pre>\$PROJ_DIR\$//////middleware/usb_1.6.3/pd</pre>
<pre>\$PROJ_DIR\$//////middleware/usb_1.6.3/osa</pre>
<pre>\$PROJ_DIR\$//////middleware/usb_1.6.3/include</pre>
<pre>\$PROJ_DIR\$//////middleware/usb_1.6.3/pd/cmsis_wrapper</pre>
\$PROJ_DIR\$///
<pre>\$PROJ_DIR\$/////devices/MK64F12/drivers</pre>
<pre>\$PROJ_DIR\$//////devices/MK64F12/cmsis_drivers</pre>
\$PR0J_DIR\$//////middleware/usb_1.6.3/pd/ptn5110
<pre>\$PROJ_DIR\$//////devices/MK64F12/utilities</pre>
<pre>\$PROJ_DIR\$/////devices/MK64F12</pre>

Figure 8. Replacing text

25. Replace the FRDM-K64F workspace preprocessor paths with the previously-copied values. Before clicking the "OK" button, change the values inside the "Define symbols" text box: add "CPU_MK64FN1M0VLL12", "USB_STACK_FREERTOS_HEAP_SIZE=32768", "USB_STACK_FREERTOS", and "FSL_RTOS_FREE_RTOS".

Figure 9. Changing values in text boxes

26. Navigate to "Project Options..." → "Assembler" → "Preprocessor", add the following paths into the "Additional include directories" section, and click the "OK" button:

```
$PROJ_DIR$/../../../..
$PROJ_DIR$/..
```

Options for node "usb_pd	l_fre	ertos"	X
Calana			
Lategory:		Factory Settings	
General Options			
Static Analysis			
Runtime Checking			
C/C++ Compiler		Language Output List Preprocessor Diagnostics Extra Options	
Assembler			
Output Converter		Ignore standard include directories	
Custom Build			
Build Actions			
Linker		Additional include directories: (one per line)	
Debugger			
Simulator		sPROJ_DIR\$/	
Angel	-		
CADI		· · · · · · · · · · · · · · · · · · ·	
CMSIS DAP		Defined autholes (one per line)	
GDB Server			
IAR ROM-monitor		A	
I-jet/JTAGjet			
J-Link/J-Trace			
TI Stellaris			
Macraigor			
PE micro			
RDI			
ST-LINK			
Third-Party Driver	-		
		OK Cancel	

Figure 10. Adding paths

27. After including the directories, compiler paths, and defined symbols, add the source code to the workspace. To do that, right-click a group name (for example; board), expand the "Add" menu, and click the "Add Files..." option. In the "Add Files" wizard, use Table 3 as a guideline for which groups, paths, and files to add.

Group	Path	Files
board	\boards\frdmk64f\usb_examples\usb_pd\freertos	board.c, board.h, clock_config.c, clock_config.h, pin_mux.c, pin_mux.h
drivers	\devices\MK64F12\drivers	fsl_clock.c, fsl_clock.h, fsl_dmamux.c, fsl_dmamux.h, fsl_edma.c, fsl_edma.h, fsl_gpio.c, fsl_gpio.h, fsl_i2c.c, fsl_i2c.h, fsl_i2c_edma.c, fsl_i2c_edma.h, fsl_lptmr.c, fsl_lptmr.h, fsl_pit.c, fsl_pit,h, fsl_port.h, fsl_uart.c, fsl_uart.h, fsl_sim.c, fsl_sim.h
drivers	\devices\ MK64F12\drivers\cmsis_drivers	fsl_i2c_cmsis.c, fsl_i2c_cmsis.h
freertos\portable	\rtos\freertos_9.0.0\Source\portable\IAR\ARM_CM4F	fsl_tickless_generic.c, fsl_tickless_systick.c, port.c, portasm.s, portmacro.h
freertos	\rtos\freertos_9.0.0\Source	croutine.c, event_groups.c, list.c, queue.c, task.c, timers.c
freertos	\rtos\freertos_9.0.0\Source\portable\MemMang	heap_4.c
freertos	\rtos\freertos_9.0.0\Source\include	croutine.h, deprecated_definitions.h, event_groups.h, FreeRTOS.h, freertos_tasks_c_additions.h, list.h, mpu_prototypes.h, mpu_wrappers.h, portable.h, projdefs.h, queue.h, semphr.h, StackMacros.h, task.h, timers.h
sources	\boards\frdmk64f\usb_examples\usb_pd\freertos	FreeRTOSConfig.h, main.c, pd_app.c, pd_app.h, pd_app_demo.c, pd_command_app.c, pd_command_interface.c, pd_command_interface.h, pd_power_app.c, pd_power_interface.c, pd_power_interface.h, RTE_Device.h, usb_pd_config.h, usb_timer.h
usb\cmsis_wrapper	\middleware\usb_1.6.3\pd\cmsis_wrapper	usb_cmsis_iic_wrapper.c, usb_cmsis_wrapper.c, usb_cmsis_wrapper.h
usb\include	\middleware\usb_1.6.3\include	usb.h, usb_misc.h
usb\osa	\middleware\usb_1.6.3\osa	usb_osa.h, usb_osa_freertos.c, usb_osa_freertos.h
usb\pd	\middleware\usb_1.6.3\pd	usb_pd.h, usb_pd_connect.c, usb_pd_interface.c, usb_pd_interface.h, usb_pd_msg.c, usb_pd_phy.h, usb_pd_policy.c, usb_pd_spec.h, usb_pd_timer.c, usb_pd_timer.h
usb\ptn5110	\middleware\usb_1.6.3\pd\ptn5110	usb_pd_ptn5110.h, usb_pd_ptn5110_connect.c, usb_pd_ptn5110_hal.c, usb_pd_ptn5110_interface.c, usb_pd_ptn5110_msg.c, usb_pd_ptn5110_register.h
Utility	\boards\frdmk64f\usb_examples\usb_pd\freertos	usp_ro.ri, usp_kinetis_ro_arv.c, usp_pit_arv.c, usb_timers.h

Table 3. FRDM-K64F source files

4.1.2. MCUXpresso IDE

- 1. Follow the instructions in Appendix A to build the SDK package for FRDM-KL27Z using the MCUXpresso IDE.
- 2. Follow the instructions in Appendix B to build the SDK for a selected platform (for example; FRDM-K64F) using the MCUXpresso IDE.
- 3. In Explorer, navigate to ...*boards\frdmk64f\usb_examples*. Add a new folder with the name of the USB Power Delivery example project to port (for example; *usb_pd*). Inside this folder, add a new folder named *freertos*.
- 4. Copy the SDK board template files *board.c*, *board.h*, *clock_config.c*, and *clock_config.h*, which can be found in the ...*boards\frdmk64f\project_template* folder.
- 5. In the second Explorer window, navigate to the ...*boards\frdmk64f\usb_examples\usb_pd\freertos* folder and paste the files copied in the previous step.
- 6. Copy the FreeRTOS configuration template file (*FreeRTOSConfig.h*) which can be found in the ...*rtos\freertos_9.0.0\template_application\ARM_CM4F* folder.
- 7. Navigate to the ...*boards\frdmk64f\usb_examples\usb_pd\freertos* folder and paste the *FreeRTOSConfig.h* file copied in the previous step.
- 8. In the first Explorer window, navigate to the ...\boards\frdmk64f\cmsis_driver_examples\i2c\interrupt_transfer folder and copy the *RTE_Device.h* file.
- 9. Paste the *RTE_Device.h* file into the ...*boards\frdmk64f\usb_examples\usb_pd\freertos* folder.
- 10. Obtain the USB Power Delivery source code of the downloaded software (Appendix A).
 - a. Unzip the FRDM-KL27Z SDK package folder.
 - b. Use the Explorer to copy the example.xml, main.c, pd_app.c, pd_app.h, pd_app_demo.c, pd_command_app.c, pd_command_interface.c, pd_command_interface.h, pd_power_app.c, pd_power_interface.c, pd_power_interface.h, usb_io.h, usb_kinetis_io_drv.c, usb_pd_config.h, usb_pd_freertos.xml, usb_pit_drv.c, and usb_timer.h files located in the ...\boards\frdmkl27z\usb_examples\usb_pd\freertos folder.



Figure 11. Copying files

- c. Paste the files into the ...\boards\frdmk64f\usb_examples\usb_pd\freertos folder.
- d. To add the Power Delivery middleware support, copy the FRDM-KL27Z *pd* folder located in the ...*middleware\usb_1.7.0* folder.
- e. Insert the copied *pd* folder into the FRDM-K64F middleware software located in the ...*middleware*\usb_1.6.3 folder.
- f. Replace the FRDM-K64F *usb_misc.h* file located in the ...*middleware**usb_1.6.3**include* folder with the version found in the FRDM-KL27Z SDK.
- 11. Follow Appendix C to add the Arduino signals to the *pin_mux.c* and *pin_mux.h* files using the MCUXpresso Config Tools.
- 12. Using a text editor, open the *example.xml* file located in the ...*boards\frdmk64f\usb_examples\usb_pd\freertos* folder. Replace all FRDM-KL27Z, frdmkl27z, fsl_dma., fsl_i2c_dma, MKL27Z644, 1.7.0, and ARM_CM0 occurrences with FRDM-K64F, frdmk64f, fsl_edma., fsl_i2c_edma, MK64F12, 1.6.3, and ARM_CM4F.
- Using a text editor, open the usb_pd.xml file located in the ...\boards\frdmk64f\usb_examples\usb_pd\freertos folder. Replace all frdmkl27z, drivers.dma, MKL27Z644, and fpu.none occurrences with frdmk64f, drivers.edma, MK64F12, and fpu.fpv4.hard.
- 14. Repeat steps 3-13 for the *usb_pd_battery* and *usb_pd_source_charger* example projects.
- 15. Use a text editor (such as Notepad++) to open the *FRDM-KL2Z_manifest.xml* file located in the ...*SDK_2.2.1_FRDM-KL27Z* folder and the *FRDM-K64F_manifest.xml* file located in the ...*SDK_2.2_FRDM-K64F* folder.
- 16. Copy the lines 570-584 from the FRDM-KL2Z_manifest.xml file.

🔚 FRD	M-KL27	ζ_manfestzmi [Σ]
568 569		<pre></pre> <pre><</pre>
570	-	<pre><example category="usb_examples/usb_pd" id="frdmkl27z_usb_examples_usb_pd_freertos" name="freertos" toolchain="mcuxpresso"></example></pre>
571	•	<pre><external path="boards/frdmkl27z/usb_examples/usb_pd/freertos" type="xml"></external></pre>
572		<files mask="usb_pd_freertos.xml"></files>
573		
574		
575	-	<pre><example category="usb_examples/usb_pd_battery" id="frdmkl27z_usb_examples_usb_pd_battery_freertos" name="freertos" toolchain="mcuxpresso"></example></pre>
576	•	<pre><external path="boards/frdmkl27z/usb_examples/usb_pd_battery/freertos" type="xml"></external></pre>
577		<files mask="usb_pd_battery_freertos.xml"></files>
578		
579		
580	-	<pre><example category="usb_examples/usb_pd_source_charger" id="frdmkl27z_usb_examples_usb_pd_source_charger_freertos" name="freertos" toolchain="mcuxpresso"></example></pre>
581	•	<pre><external path="boards/frdmkl27z/usb_examples/usb_pd_source_charger/freertos" type="xml"></external></pre>
582		<files mask="usb_pd_source_charger_freertos.xml"></files>
583		
584		
585		<pre></pre>
586		

Figure 12. Copying lines

17. In the FRDM-K64F_manifest.xml file, paste the previously copied text after line 955. After that, replace all frdmkl27z occurrences with frdmk64f.

FRIDM	HU272_mandestant 🔝 🖶 FRDM-K6F_manifestant 🖾
951 Ø	<pre></pre>
953	
954	
955 956	<pre>< <example category="usb_examples/usb_pd" io="trdmkb4_usb_examples_usb_pd_treertos" name="treertos" toolchain="mcuxpresso"></example></pre>
957	<pre><files mask="usb_pd_freertos.xml"></files></pre>
958	<pre></pre>
959	<pre></pre>
960 961 962	<pre>cexample id="frdmk64f_usb_examples_usb_pd_battery_freertos" name="freertos" category="usb_examples/usb_pd_battery" toolchain="mcuxpresso"></pre>
963	
964	L
965	<pre><cexample category="usb_examples/usb_pd_source_charger" id="frdmk64f usb_examples usb_pd_source_charger_freertos" name="freertos" toolchain="mcuxpresso"></cexample></pre>
966	<pre>cexternal path="boards/frdmk64f/usb_examples/usb_pd_source_charger/freertos" type="xml"}</pre>
967	<pre><files mask="usb_pd_source_charger_freertos.xml"></files></pre>
968	
969	L
970	/ avamples

Figure 13. Copying lines

- 18. Copy lines 3086-3136 from the *FRDM-KL2Z_manifest.xml* file.
- 19. In the *FRDM-K64F* manifest.xml file, paste the previously copied text after line 3866. After that, replace all MKL27Z644, MKL27Z64xxx4, and 1.7.0 occurrences with MK64F12, MK64FN1M0xxx12, and 1.6.3.

4.1.3. Migrating source files in both IDEs

At this stage, only the *pin_mux.c* and *pin_mux.h* files align with the FRDM-K64F development board. Follow these steps to migrate the software to a new platform:

*utility**usb_io.h*—add the following lines of code:

```
typedef enum _k64_ports
{
        kPTA = 0,
        kPTB,
        kPTC,
        kPTD,
        kPTE
}k64 ports;
#define configMAX PRIORITIES
                                            8
```

*sources**FreeRTOSConfig.h*—replace these definitions with the values highlighted in **orange**:

```
#define configUSE_TIME_SLICING
                                            1
#define configTOTAL_HEAP_SIZE
                                  ((size_t)(<mark>5*1024</mark>))
#define configTIMER TASK PRIORITY (configMAX PRIORITIES - 1)
```

- sources\main.c—inside HW_TimerInit, replace PIT_IRQn with PIT0_IRQn: <u>NVIC_SetPriority(PIT0_IRQn</u>, PD_TIMER_INTERRUPT_PRIORITY);
- Use Table 1 to find the GPIO assigned to the PTN5150 I²C signals A4 and A5. For example, the FRDM-K64F rev E IOs are PTC11-I2C1_SDA and PTC10-I2C1_SCL.

```
#define I2C1_SCL (10U)
#define I2C1 SDA (11U)
```

 Look for the BOARD_I2C1_ReleaseBus function and change the number highlighted in green to the I²C instance connected to D14 – D15. For example, this function can be named BOARD_I2C0_ReleaseBus for the FRDM-K64F.

```
void BOARD_I2C0_ReleaseBus(void)
```

 Look for the BOARD_I2C0_ReleaseBus function and change the number highlighted in green to the I²C instance connected to A4 – A5. For example; this function can be named BOARD_I2C1_ReleaseBus for the FRDM-K64F.

```
void BOARD_I2C1_ReleaseBus(void)
```

 Modify the BOARD_I2C1_ReleaseBus function IO numbers so that they are compatible with the GPIOs used for the FRDM-K64F A4 – A5 signals.

```
void BOARD I2C1 ReleaseBus (void)
{
    uint8 t i = 0;
    gpio pin config t pin config;
   port pin config t i2c pin config = {0};
    /* Config pin mux as gpio */
    i2c pin config.pullSelect = kPORT PullUp;
    i2c pin config.mux = kPORT MuxAsGpio;
    pin config.pinDirection = kGPIO DigitalOutput;
    pin config.outputLogic = 1U;
    CLOCK EnableClock (kCLOCK PortC);
    PORT_SetPinConfig(PORTC, I2C1_SCL, &i2c_pin_config);
    PORT SetPinConfig(PORTC, I2C1 SDA, &i2c pin config);
    GPIO PinInit(GPIOC, I2C1 SCL, &pin config);
    GPIO PinInit(GPIOC, I2C1 SDA, &pin config);
    /* Drive SDA low first to simulate a start */
    GPIO WritePinOutput (GPIOC, I2C1 SDA, OU);
    i2c release bus delay();
    /* Send 9 pulses on SCL and keep SDA high */
    for (i = 0; i < 9; i++)
    {
        GPIO WritePinOutput (GPIOC, I2C1 SCL, OU);
        i2c release bus delay();
        GPIO WritePinOutput (GPIOC, I2C1 SDA, 1U);
        i2c release bus delay();
        GPIO WritePinOutput (GPIOC, I2C1 SCL, 1U);
        i2c release bus delay();
        i2c release bus delay();
```

```
}
/* Send stop */
GPIO_WritePinOutput(GPIOC, I2C1_SCL, 0U);
i2c_release_bus_delay();
GPIO_WritePinOutput(GPIOC, I2C1_SDA, 0U);
i2c_release_bus_delay();
GPIO_WritePinOutput(GPIOC, I2C1_SCL, 1U);
i2c_release_bus_delay();
GPIO_WritePinOutput(GPIOC, I2C1_SDA, 1U);
i2c_release_bus_delay();
```

 Modify the BOARD_I2C0_ReleaseBus function IO numbers so that they are compatible with the GPIOs used for the FRDM-K64F D14 – D15 signals.

```
void BOARD I2C0 ReleaseBus(void)
{
    uint8 t i = 0;
    gpio_pin_config_t pin_config;
    port_pin_config_t i2c_pin_config = {0};
    /* Config pin mux as gpio */
    i2c pin config.pullSelect = kPORT PullUp;
    i2c_pin_config.mux = kPORT_MuxAsGpio;
    pin config.pinDirection = kGPIO DigitalOutput;
    pin config.outputLogic = 1U;
    CLOCK EnableClock (kCLOCK PortE);
    PORT_SetPinConfig(PORTE, 24u, &i2c_pin_config);
    PORT_SetPinConfig(PORTE, 25u, &i2c_pin_config);
    GPIO_PinInit(GPIOE, 24u, &pin_config);
    GPIO PinInit (GPIOE, 25u, &pin config);
    /* Drive SDA low first to simulate a start */
    GPIO WritePinOutput(GPIOE, 25u, 0U);
    i2c release bus delay();
    /* Send 9 pulses on SCL and keep SDA high */
    for (i = 0; i < 9; i++)
    {
        GPIO WritePinOutput(GPIOE, 24u, 0U);
        i2c release bus delay();
        GPIO WritePinOutput(GPIOE, 25u, 1U);
        i2c release bus delay();
        GPIO WritePinOutput(GPIOE, 24u, 1U);
        i2c release bus delay();
        i2c_release_bus_delay();
    }
    /* Send stop */
    GPIO WritePinOutput(GPIOE, 24u, 0U);
    i2c_release_bus_delay();
```

```
GPIO_WritePinOutput(GPIOE, 25u, 0U);
i2c_release_bus_delay();
GPIO_WritePinOutput(GPIOE, 24u, 1U);
i2c_release_bus_delay();
GPIO_WritePinOutput(GPIOE, 25u, 1U);
i2c_release_bus_delay();
```

}

- Go to the main function and rename BOARD_I2C0_ReleaseBus with the I²Cn instance used for A4 A5 (BOARD_I2C1_ReleaseBus).
- Use Table 1 to find the GPIO assigned to the EXTRA_EN_SRC output signal (D4). On the FRDM-K64F rev.E, the PTB23 output pin is used. Update the HW_GpioExternalSourceEnable IO software to be compatible with PTB23, as shown below.

```
void HW_GpioExternalSourceEnable(uint8_t enable)
{
    if (enable)
    {
        USB_GpioOutputWritePin(kPTB, kPTB, 23, 1);
    }
    else
    {
        USB_GpioOutputWritePin(kPTB, kPTB, 23, 0);
    }
```

• Modify the port and pin number values inside the HW_GpioReadPowerRequestSW and HW_GpioReadPRSwapSW functions using the information in Table 2.

```
uint8_t HW_GpioReadPowerRequestSW(void)
{
   return USB_GpioInputReadPin(kPTC, kPTC, 6);
}
uint8_t HW_GpioReadPRSwapSW(void)
{
   return USB_GpioInputReadPin(kPTA, kPTA, 4);
```

- }
- Look for the HW_GpioInit function and modify the GPIO initialization with the FRDM-K64F IOs. Use Table 1 and Table 2 as a reference (external power control = EXTRA_EN_SRC, SW1 = SW2, SW3= SW3, PD PHY interrupt = nALERT).

```
/* the external power control pin */
USB_GpioOutputInit(kPTB, kPTB, 23);
    /* SW1 */
    USB_GpioInputInit(kPTC, kPTC, 6);
    /* SW3 */
    USB_GpioInputInit(kPTA, kPTA, 4);
    /* the PD PHY interrupt gpio */
    USB_GpioInterruptInit(kPTC, kPTC, 12u, kUSB_GpioInterruptLogicZero,
    HW_GpioPDPHYIntCallback);
• Look for the PORTB_PORTC_PORTD_PORTE_IRQHandler function and change the
```

IRQHandler portion to match the port used on the nALERT pin.

```
void PORTC_IRQHandler(void)
{
```

MCUXpresso USB PD Migration Guide, Rev. 0, 10/2017

```
if (PORTC->ISFR & (1U << 12u))
{
    PORTC->ISFR = 1U << 12u;
    if (!(GPIOC->PDIR & (1U << 12u)))
    {
        HW_GpioPDPHYIntCallback();
    }
}</pre>
```

• *board**board*.*h*—using a text editor, open the

 $...\boards\frdmk64f\demo_apps\hello_world\board.h$ file and copy all the BOARD_SWn definitions. Navigate to the IAR IDE workspace, double-click the *board**board.h* file, and paste the content after line 55.

```
/* Define the port interrupt number for the board switches */
#define BOARD_SW2_GPIO GPIOC
#define BOARD_SW2_PORT PORTC
#define BOARD_SW2_GPIO_PIN_6U
#define BOARD_SW2_IRQ_PORTC_IRQN
#define BOARD_SW2_IRQ_HANDLER PORTC_IRQHandler
#define BOARD_SW2_NAME "SW2"
#define BOARD_SW3_GPIO_GPIOA
#define BOARD_SW3_GPIO_PIN_4U
#define BOARD_SW3_GPIO_PIN_4U
#define BOARD_SW3_IRQ_PORTA_IRQN
#define BOARD_SW3_IRQ_HANDLER PORTA_IRQHandler
#define BOARD_SW3_IRQ_HANDLER PORTA_IRQHANDLER
#define BOARD_SW3_IRQ_HANDLER PORTA_IRQHANDLER
```

BOARD_ARDUINO_I2C_IRQ, and BOARD_ARDUINO_I2C_INDEX.

#define	BOARD	ARDUINO	INT	IRQ	PORTC_IRQn
#define	BOARD	ARDUINO	I2C	IRQ	I2C0_IRQn
#define	BOARD	ARDUINO	_I2C_	INDEX	0

- *sources\pd_app.h*—replace all sw1State and sw1Time occurrences with sw2State and sw2Time. Use CTRL + H to speed up the replacing process.
- *sources\pd_app_demo.c*—replace all sw1State and sw1Time occurrences with sw2State and sw2Time. Use CTRL + H to speed up the replacing process.

To import and build the MCUXpresso IDE projects, follow these steps:

- 1. Import the K64 SDK into the MCUXpresso IDE by dragging and dropping the unzipped *SDK_2.2_FRDM-K64F* folder into the "Installed SDK" view.
- 2. Import the *usb_pd* example application by clicking "Import SDK example(s)..." located in the "Quick Start Panel" in the lower left-hand corner.
- 3. Click the "frdmk64f" board image to select the project that can run on that board and click the "Next" button.

Please select a board			
Board and/or Devi	ce selection page		
SDK MCUs	Available boards		12 12 2
1CUs from installed SDKs	Please select an available board for your proj	ect.	
 KG KL2x 	formk2Zz 3	sor	
	0xxx12 using board: FRDM-K64F	SDKs for selected MCU	
elected Device: MK64FN1M		Name Version	Location Control Control Cont

Figure 14. Selecting a project

4. Use the arrow button to expand the "usb_examples" category, and then click the checkbox next to "usb_pd" to select that project. Deselect the "Enable semihost" option in the project options and click the "Next" button.

You have selected '1' projects to import.	1	NPr
ocation: C\mcuxpresso\pd\frdmk64f_		Browse_
Project Type	Project Options	
O Project	Enable semihost Copy sources Import other files	
xamples		₩ 2 2 % 9 8
type to filter		
Image: Service, printer virtual, plain, text_Itte Image: Service, virtual, camera Image: Service, virtual, camera, ite Image: Service, virtual, vir		
Bub host_printer_plan_text wb_usp.uspend_resume_device_hid_mouse ub_usp.uspend_resume_device_hid_mouse_lite ub_uspend_resume_host_hid_mouse voit voit		E

Figure 15. Importing a project

5. In the "Advanced Settings" area, untick the "Redirect SDK "PRINTF" to C library "printf" checkbox to use the MCUXpresso SDK console functions for printing instead of the generic C library functions. Click the "Finish" button.

y Settings + (and hosting variant) Pacific point version of printf Pacify point version of point version of printf Pacify	Cf - L Darry Settings Library type (and hosting variant) Redilb (nehost-mf) • Medilb Use fosting point version of print! NewlibNanc: Use floating point version of print! • Redilb: Use fosting point version of print! NewlibNanc: Use floating point version of print! • Redilb: Use fosting point version of print! NewlibNanc: Use floating point version of print! • Redil: Use fosting point version of print! NewlibNanc: Use floating point version of scant! • Redirect point/fixent to UMAT NewlibNanc: Use floating point version of scant! • mory Ordinguration • • • mory Status • • • M SRAM_UPPER RAM 0.400000 0.400000 M SRAM_UPPER RAM 0.400000 0.40000 M FLEX_RAM RAM3 0.400000 0.40000	
(and horing variant) Rettle (chordstaft) Rettle (chordstaft) floating point version of printf NewlibNanc: Use floating point version of printf dharacer rather than string based printf NewlibNanc: Use floating point version of printf dharacer rather than string based printf NewlibNanc: Use floating point version of scanf wittyScant to Ukart Itema floating figuration S s PROGRAM FLASH RAM 0x00000 sRAM_UPER RAM SAM_LOWER RAM RAM3 0x1400000 dh100000 0x10000	BitBark Ype (and hotting variant) Reditib (achost-stn) • Reditib Use floating point version of printf NewliBNano: Use floating point version of printf Reditib Use floating transmit mark that sing based printf NewliBNano: Use floating point version of scant Reditis Use character after than sing based printf NewliBNano: Use floating point version of scant Reditise character after than sing based printf NewliBNano: Use floating point version of scant Reditise character after than sing based printf NewliBNano: Use floating point version of scant Reditise character to TIM NewliBNano: Use floating point version of scant Reditect printf/scant to UART Ready Redit Procenting and the sing based printf NewliBNano: Use floating point version of ortication Stark LOVER RaAM M SRAM_UPPER RAM Out100000 M FLEX_RAM RAM Out100000 Notice Out100000 Nare textings	Ibinary type (and hosting variant) Rediib (chohosten) • Rediib: Use Anactive Hand tring based printf NewlibNanc: Use floating point version of printf Rediib: Use Anactive Hand tring based printf NewlibNanc: Use floating point version of printf Rediid: Use Anactive Hand tring based printf NewlibNanc: Use floating point version of printf Rediid: Use Anactive Hand tring based printf NewlibNanc: Use floating point version of scanf Redirect print/scand to UMAT Newnoy Configuration wenoy details > ype Name Alias Location AM SRANLUPPER RAM 0x2000000 0x20000 AM SRANLUPPER RAM3 0x100000 0x10000 AM SRANLUPPER RAM3 0x100000 0x10000 AM REX_RAM RAM3 0x100000 0x10000 Hooting Yorth Yee Float (HardABE)
floating point version of printf ☐ NewliBNAnci: Use floating point version of printf obaracter trather than string based printf ☐ NewliBNanci: Use floating point version of scanf KPRIVET to Clibany "printf" Intficant to IDM Intficant to IDM PROGRAM,FLASH Alias Location Size Driver PROGRAM,FLASH Raih 0x00000 Driver SRAM_UDVER RAM2 ALIAS 0x00000 RLEX,RAM RAM3 0x100000 0x10000	NextRibuting point version of printf NextRibuting point version of printf NextRibuting point version of printf NextRibuting point version of printf NextRibuting point version of printf NextRibuting point version of printf NextRibuting point version of scant NextRibuting point version of scant NextRibuting point version of scant NextRibuting point version of scant NextRibuting point version of scant NextRibuting point version of scant Indirect point version to UMAT NextRibuting point version of scant mory details NextRibuting point version of scant Non Name Alias Location Size Driver nh PROGRAM_FLASH Flash 0x0 M SRAM_UPPER RAM 0x00000 TTFE_4K-cfx M SRAM_UPPER RAM3 0x1400000 0x10000 M FLEX_RAM RAM3 0x1400000 0x10000	Reditis: Use floating point version of printf New/IBNAnc: Use floating point version of printf Redit: Use character rather than string based printf New/IBNAnc: Use floating point version of printf Redite: SUP (RMR)F to C library 'printf' Redit: SUP (RMR)F to C library 'printf' Re
character trafter than string based printf NewlibNano: Use floating point version of scanf K*PRINT* to C library *printf* mt/scanf to IIM floating point version of scanf floating point vers	Name Alias Location Driver mory details Name Alias Location Driver selines DKI (PARA) Rash Location Driver M SRAM_UPPER RAM 0x00000 FIFE_4K-cfx Edit M SRAM_UPPER RAM 0x100000 FIFE_4K-cfx Edit M SRAM_UPPER RAM 0x10000 0x10000 FIFE_4K-cfx M SRAM_UPPER RAM 0x10000 0x10000 Edit	Redite: Use character rather than string based printf □ NewlibNanc: Use floating point version of scanf Reditect SDK* VPRNTF to C library 'printf" Reditect printfycant to ITM Reditect printfycant to UNAT Wemory Configuration emory details ype Name Alias Location SRAM_LDVFR RAM 0x000000 bci00000 Alias Location SRAM_LOVER RAM2 0x100000 bci00000 Alias Location0 SRAM_LOVER RAM2 AuA RELKRAM AuA Did00000 bci00000 bci00000 bci00000 bci00000 AuA RELK_RAM Rediter printfycart to train Rediter printfycart to train AuA REAL bci00000 bci00000 AuA RELK_RAM bci4000000 bci0000 Autority Printfycart to train Rediter printfycart to train Rediter printfycart to train Rediter printfycart to train Rediter printfycart to train Rediter printfycart to train
K 'PRINTF' to C libray 'printf' Intficant to DIAT figuration S Name Alias Location Size Driver PROGRAM_FLASH Flash 0x0 0x10000 FFFE_4K.cfx SRAM_UDPER RAM 0x200000 0x3000 FFFE_4K.cfx REX.RAM 0x400000 0x1000 FFFE_4K.cfx FLEX.RAM RAM3 0x1400000 0x1000 FFFE_4K.cfx FLEX.RAM RAM3 0x1400000 0x1000 FFFE_4K.cfx	Name Alias Location Size Driver stafunct Sprint/Scant to UABT PROGRAM.FLASH Flash 0x0 0x100000 FTFE_4K.cfx Edited. stafunct Sprint/Scant to UABT Size Driver Driver 0x100000 FTFE_4K.cfx Edited.	Statuse SDK "PRINT" to C library "print" Redirect print/scant to IIM Status Status Status Am SRAM_UPPER RAM SAAL_UPPER RAM3 Out10000 Out10000 Value RAM4 SRAM_LOVER RAM3 Out10000 Value Redirect prints
ntf/scant to UJART s Name Alias Location Size Driver PROGRAM_FLASH Flash 0x00000 Driver SRAM_UPPER RAM 0x1400000 Driver FLEX_FRAM_CONTHRODO Drivoon FLEX_FRAM RAM3 0x1400000 Drivoon	Bedirect print/cond to UMAT remory Configuration mory details De Name Alias Location Size Driver with PROGRAM_FLASH Flash 0x0 0x100000 FTFE_4Kcfx Eath M SRAM_UPPER RAM 0x000000 0x30000 Mission Eath M FREALNOVER RAM3 0x1400000 0x10000 Mission Eath M FREALNOVER RAM3 0x1400000 0x10000 Mission Satisfies Eath	Redirect printf/scant to UART Redirect printf/scant to UART Wemory Configuration emory details ype Name Alias Location Status Driver Iain PROGRAM_LLASH Alias Location SRAM_LIPPER RAM Outprotocol 0x00000 AM SRAM_LOWER RAM2 0x100000 AM FLEX_RAM AM 0x4000000 Iardware settings tfloating Point type
Influenzation s S Name Alias Location Size Driver PROGRAM_FLASH Flash 0x0 0x100000 FFFE_4K.cfx Edit SRAM_UDPER RAM2 0x100000 0x10000 Edit FLEX_RAM RAM3 0x1400000 0x10000 Edit	Bedirect print/scart to UART emory Configuration mory details Dev Name Alias Location Size Driver shh PROGRAM_FLASH Flash 0x0 0x100000 FFFE_4K.ctx M SRAM_LIPPER RAM 0x2000000 0x30000 M GRAM_FLASH Flash 0x0 0x40000 M GRAM_FLASH Flash 0x0 0x40000 M GRAM_FLASH Flash 0x0 0x40000 M GRAM_FLASH Flash 0x0 0x10000 M GRAM_FLASH Flash 0x0 M GRAM_FLASH Flash 0x0 0x0 M GRAM_FLASH Flash 0x0 0x0 M GRAM_FLASH <td< td=""><td>IRedirect print/scan to UART Vermory Configuration syne Name Alias Location Size Driver Isah PROGRAM_FLASH Flash 0x0 0x100000 FIFE_4K.cfx Edit. Axid SRAM_LOPPER RAM 0x200000 0x100000 0x10000 Edit. Axid SRAM_LOVER RAM2 0x100000 0x10000 0x10000 Edit. Axid FLEX_RM RAM3 0x100000 0x10000 Edit. Edit. Floating Point Yeer FMA41420 State State State State</td></td<>	IRedirect print/scan to UART Vermory Configuration syne Name Alias Location Size Driver Isah PROGRAM_FLASH Flash 0x0 0x100000 FIFE_4K.cfx Edit. Axid SRAM_LOPPER RAM 0x200000 0x100000 0x10000 Edit. Axid SRAM_LOVER RAM2 0x100000 0x10000 0x10000 Edit. Axid FLEX_RM RAM3 0x100000 0x10000 Edit. Edit. Floating Point Yeer FMA41420 State State State State
Figuration Size Driver Procession Size Driver Procession Size Driver Procession Size Driver Procession	Permory Configuration mory details pe Name Alias Location Size Driver sin PROGRAM_FLASH Flash 0x0 0x100000 FTFE_4K.cfx M SRAM_UPPER RAM 0x000000 0x30000 0x30000 FTFE_4K.cfx M SRAM_UPVER RAM 0x100000 0x10000 0x10000 M FLEX_RAM RAM3 0x14000000 0x10000 0x10000 xetware settings	Memory Configuration emory details pipe Name Alias Location Size Driver Iaih PROGRAM_FLASH Flash 0x0 0x100000 FFE_4K.cfx XAM SRAM_UOPER RAAM 0x2000000 0x20000 Driver XAM SRAM_UOPER RAAM 0x10000 Driv000 EBRL XAM FLAX_RAM RAM3 0x1400000 Dri0000 Dri0000 EBRL Ladware settings
Infiguration s S Name Alias Location Size Driver PROGRAM FLASH Flash 0x0 0x100000 FIFE_4K.cfx SRAM_UPPER RAM 0x2000000 0x100000 FIFE_4K.cfx RAM_UVER RAM2 0x1400000 0x10000 FIEL FLEX_RAM RAM3 0x1400000 0x1000 FIEL	Bennoy Gentguration mory details mory details Size Driver sh PROGRAM_FLASH Flash 0x0 0x100000 FTFE_4K.cfx M SRAM_LOWER RAM2 0x100000 0x100000 Bennov Bennov M SRAM_LOWER RAM2 0x100000 0x10000 Bennov Bennov Bennov wdware settings Bennov Bennov <t< th=""><th>Memory details Size Driver ype Name Alias Location Size Driver tash PROGRAM_FLASH Flash Bu0 0x100000 FFFE_dKcfz EBIL AM SRAAL_UPPER RAM 0x20000 0x20000 0x20000 Dx20000 Dx20000 Location EBIL AM SRAAL_COVER RAM2 0x1m0000 0x10000 Dx10000 Location EBIL Auth FLEX_RAM RAM3 0x14000000 0x1000 Location EBIL Ardware settings - - - - - -</th></t<>	Memory details Size Driver ype Name Alias Location Size Driver tash PROGRAM_FLASH Flash Bu0 0x100000 FFFE_dKcfz EBIL AM SRAAL_UPPER RAM 0x20000 0x20000 0x20000 Dx20000 Dx20000 Location EBIL AM SRAAL_COVER RAM2 0x1m0000 0x10000 Dx10000 Location EBIL Auth FLEX_RAM RAM3 0x14000000 0x1000 Location EBIL Ardware settings - - - - - -
s Alias Location Size Driver NexoGRAM_FLASH Flash 0x0 0x100000 FIFE_4K.cfx SRAM_UPPER RAM 0x2000000 0x20000 Response SRAM_UPPER RAM2 0x110000 0x10000 Response FLEX_RAM RAM3 0x1400000 0x1000 Response	Bit Constraint Size Driver ph PROGRAM_FLASH Flash 0x0 0x100000 FTFE_4K.cfx M SRAM_UPPER RAM 0x200000 0x30000 0x30000 FTFE_4K.cfx M SRAM_UPPER RAM 0x200000 0x30000 0x30000 0x30000 0x30000 M SRAM_UOVER RAM3 0x1400000 0x10000	emory details Size Driver tain PROGRAM_FLASH Flash 0x0 0x100000 FIFE_4K.cfx ALM SRAM_UPPER RAM 0x2000000 0x20000 FIFE_4K.cfx ALM SRAM_UPPER RAM2 0x100000 0x20000 FIFE_4K.cfx ALM SRAM_UPPER RAM3 0x1400000 0x10000 FIFE_4K.cfx ALM SRAM_UPPER RAM3 0x14000000 0x10000 FIFE_4K.cfx ALM SRAM_UPPER RAM3 0x14000000 0x1000 FIFE_4K.cfx ALM SRAM_UPPER RAM3 0x14000000 0x1000 FIFE_4K.cfx ALM SRAM_UPPER RAM3 0x14000000 0x1000 FIFE_4K.cfx
Name Alias Location Size Driver PRCGRAM_FLASH Filah 0x0 Ox100000 FIFE_4K.cfx SRAM_UPER RAM 0x2000000 0x0000 FIFE_4K.cfx SRAM_UOWER RAM2 0x11ff0000 0x10000 FIFE_4K.cfx FLEX_RAM RAM3 0x14f00000 0x10000 FIFE_4K.cfx	pe Name Alias Location Size Driver uh PROGRAM_FLASH Flash 0x0 0x10000 FTFE_4K.cfx wh SRAM_UPPER RAM 0x2000000 0x30000 Mid000 Mid0000 Mid00000 Mid0000 Mid00000 Mid0000 Mid0000 Mid0000 Mid0000 Mid0000 Mid00000 Mid0000 Mid0000 Mid0000 Mid0000 Mid00000 Mid0000 Mid00000 Mid00000 Mid00000 Mid000000 Mid000000 Mid000000 Mid0000000 Mid0000000 Mid0000000 Mid0000	ype Name Alias Location Size Driver Ian PROCRAM_LLASH Flash 0x0 0x00000 FITE_4K.cfx AM SRAM_LDYER RAM 0x000000 0x00000 0x00000 FITE_4K.cfx EBIL AM SRAM_LOWER RAM2 0x1ff00000 0x10000 0x10000 FITE_4K.cfx EBIL AM FLEX_RAM RAM3 0x1000000 0x10000 0x10000 FITE_4K.cfx EBIL Floating Point Yet FM44(HardABE) FITE_4K.cfx
PROGRAM_FLASH Flash 0x0 0x100000 FTFE_4K.ctx SRAM_UPPER RAM 0x2000000 0x30000 FEE_4K.ctx SRAM_UVPER RAM2 0x101000 0x10000 FEE_4K.ctx FLEX_RAM RAM3 0x1400000 0x10000 FEE_4K.ctx	sh PROGRAM_FLASH Flash 0x0 0x100000 FFFE_4K.cfx M SRAM_LOVER RAM2 0x100000 0x30000 0x30000 0x100000 0x1000000 0x1000000 0x1000000	Iash PROGRAM_FLASH Flash 0x0 0x100000 FIFE_4Kcfx IAM SRAM_UPPER RAM 0x2000000 0x30000 0x30000 EBIL AM SRAM_LOVER RAM2 0x10000 0x10000 0x10000 Image: Comparison of the c
SRAM_UPPER RAM 0x2000000 0x20000 East SRAM_UPPER RAM2 0x110000 0x10000 East FLEX_RAM RAM3 0x14000000 0x10000 East	MM SRAM_UPPER RAM 0.2000000 0-630000 Edit M SRAM_LOWER RAM2 0x11ff0000 0x10000 0x10000 Edit M FLEX_RAM RAM3 0x14000000 0x10000 0x10000 0x10000 0x10000 Edit wdware settings RAM2	AM SRAM_UPPER RAM 0-2000000 0-20000 AM SRAM_UPPER RAM2 0-100000 0-20000 AM FLAULOWER RAM3 0-10000 0-20000 0-20000 AM FLAUCRAM RAM3 0-1400000 0-20000 0-20000 0-20000 AM FLAUCRAM RAM3 0-14000000 0-20000 0-20000 0-20000 Amore settings FRonting Point type [FH-4] (HardABD - - -
SRAM_LOWER RAM2 0x1/m000 0x1/000 0x1/000 FLEX_RAM RAM3 0x1/4000000 0x1/000 0x1/000 tings tings tings tings tings	M SRAM_LOWER RAM2 0.110000 0.10000 M FLEX,RAM RAM3 0.1400000 0.1000	LAM SRAM_LOWER RAM2 0.1110000 0.0000 MM FLEX_RAM RAM3 0.1400000 0.1000 Hardware settings tfloating Point type [FP4 (HardAB)) •
REX_RAM RAM3 0x14000000 0x1000	M FLEX,RAM RAM3 0x14000000 0x1000 ardware settings	IAM FLEX_RAM RAM3 0x14000000 0x1000 Hardware settings t Flow Flo
Itings	ardware settings	terdware settings t Floating Point type [FP-4 (HardAB) •
		t Floating Point type [FPv4 (HardABI)
	Flasting Deletiting and a second	• Hoating Point type [FPv4 (HardABI)
FPv4 (HardABI)	FPv4 (HardABI)	
		MCU C Compiler
int type [FPy4 (HardABI)	Floating Point type FPv4 (HardABI)	
	611 6 6 H	MCU C Compiler

Figure 16. Importing a project

Compile and flash the example.

For more information, see the *MCUXPresso SDK USB Type-C PD Stack User's Guide* located in the ...*SDK_2.2.1_FRDM-KL27Z\docs\usb* folder.

Appendix A. SDK Builder FRDM-KL27Z

Use the online MCUXpresso Config Tools to create a custom SDK package for the FRDM-K64F board.

1. Open a web browser and navigate to the MCUXpresso homepage mcuxpresso.nxp.com.



Figure 17. MCUXpresso Config Tools

2. Click the "Login to view configurations" button to create a new configuration.

NXP MCUXpresso OVERVIEW TO	OOLS - MANAGE -			English -	👤 Guest 🗸	ļ .
	MCUXpresso Config Tools pro with a Kinetis or LPC-based M development.	SO CONFIG TOOLS vides a set of system configuration tools that ICU solution. Let it be your guide from first ev	thelp users of all levels raluation to production	G		
		Select or create a configuration Cogin to view configurations				
	Config Settings Specify optional middleware and environment settings for your configuration	SDK Builder Generate a downloadable SDK archive for use with desktop MCUXpresso Tools	Project Cioner Download an existing standalone SDK example project			
Feedback	Pins Tool Assign signals to pins, set electrical properties, and generate initialization code	Clocks Tool Setup the system clocks and generate initialization code	Project Generator Create a new SDK project using source code from pin and clocks tools	5		
	What's new	Additional Link	cs			
	Figure 18.	MCUXpresso Config	Tools			

3. You are redirected to the <u>www.nxp.com</u> login page. Enter your account information or create a

MCUXpresso USB PD Migration Guide, Rev. 0, 10/2017

new account.

4. Back on the MCUXpresso homepage, click the drop-down box and select the "New Configuration" option.



Figure 19. Creating a configuration

5. Search for the board name (for example; FRDM-KL27).

Create a New Configuration

Search by device, board, kit name and filter by supported middleware.

Sear	ch by Name		
	FRDM-KL27Z		\otimes
Sele	et a Device, Board, or Kit		
	 Boards 		
	FRDM-KL27Z		
	 Processors 		
	▼ Kits		
Nam	e your configuration		
	Select Configuration	Specify Additional	Jump start your

Figure 20. Creating a configuration

6. Select a board from the list and provide a name for the configuration. Click the "Specific Additional Configuration Settings" button to select from FreeRTOS, IAR toolchain, and USB middleware.

FRDM-KL27Z Board FRDM-KL27Z Select a Device, Board, or Kit Device MKL27Z644 Soards Core Type / Max Freq Cortex-MDP / FRDM-KL27Z Memory Size 64 KB Flash 16 KB RAM 16 KB RAM	ecui	rch by Name	Hardware Details	
Select a Device, Board, or Kit Select a Device, Board, or Kit Boards FRDM-KL27Z Processors Kits Kits Device MKL27Z644 Core Type / Max Freq Cortex-MOP / Memory Size 64 KB Flash 16 KB RAM 16 KB RAM 16 KB RAM 16 KB RAM 17 Kits		FRDM-KL27Z	Board	FRDM-KL27Z
Select a Device, Board, of Kit Core Type / Max Freq Cortex-M0P / Boards FRDM-KL27Z Processors Kits General Select a Device, Board, of Kit Core Type / Max Freq Cortex-M0P / Memory Size 64 KB Flash 16 KB RAM Kits Core Type / Max Freq Cortex-M0P / Memory Size 64 KB Flash 16 KB RAM Kits Kits Core Type / Max Freq Cortex-M0P / Memory Size 64 KB Flash 16 KB RAM 	Solo	et a Davice, Board, or Kit	Device	MKL27Z644
 ▼ Boards FRDM-KL27Z ○ Processors ○ Kits 	Sele	a beaue, board, or Kit	Core Type / Max Freq	Cortex-MOP / 48M
FRDM-KL27Z 16 KB RAM Processors Kits		 Boards 	Memory Size	64 KB Flash
 ▼ Processors ▼ Kits 		FRDM-KL27Z		16 KB RAM
▼ Kits		▼ Processors		
		▼ Kits		
Name your configuration				

Figure 21. Creating a configuration

- 7. In the "Configuration Settings" section, set the following:
- Host OS \rightarrow Windows.
- Toolchain/IDE \rightarrow IAR Embedded Workbench for Arm.
- Middleware \rightarrow USB Stack, FreeRTOS.

Host OS	Toolchain / IDE		
Windows -	IAR Embe	dded Workbench for ARM 🔹	Set as default
lect Optional Middlewa	re in your SDK downlo	and generated projects, and will impact Der	nheral Tool settings
lect Optional Middlewa ections here will be included	re in your SDK downlo	oad, generated projects, and will impact Per	pheral Tool settings
lect Optional Middlewa ections here will be included 2 items selected	re in your SDK downlo	oad, generated projects, and will impact Pen Selected Middleware	pheral Tool settings

Figure 22. IAR IDE

Or:

- Host OS \rightarrow Windows.
- Toolchain/IDE \rightarrow MCUXpresso IDE.
- Middleware \rightarrow USB Stack, FreeRTOS.

Configuration Settings

Specify included middleware, RTOS selections, and development preferences.

Set as default rojects, and will impact Peripheral Tool settings	MCUXpresso IDE	s -	Windows
rojects, and will impact Peripheral Tool settings	9		
Middleware	Selected Mic	ted	2 items selected

Figure 23. MCUXpresso IDE

8. After selecting the above settings, click the "Go to SDK Builder" button.



es projects included in the SDK Download and Generate ain / IDE	Device Core Type / Max Freq Memory Size	KL27Z644 Cortex-M0P / 48MHz 64 KB Flash
es projects included in the SDK Download and Generate ain / IDE	d Projects	Cortex-M0P / 48MHz 64 KB Flash
Embedded Workbench for ARM	Set as default	16 KB RAM
Cownload, generated projects, and will impact Periphera	al Tool settings	
USB Type-C PD stack, FreeRTOS		
DF	0K download, generated projects, and will impact Periphera Selected Middleware USB Type-C PD stack, FreeRTOS	DK download, generated projects, and will impact Peripheral Tool settings Selected Middleware USB Type-C PD stack, FreeRTOS Jump start your



9. Click the "Download Now" button to download the SDK package.

NOTE

You may see the "Request to Build" button instead of the "Download Now" button. Click the "Request to Build" button and when the package is built, a notification to download it appears in the "SDK Archive" section. To access the "SDK Archive" section, click the "MANAGE" tab and select the "SDK Archive" option.

SDK Builder Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.	I	
Current Configuration	Hardware Details	
FRDM-KL27Z 🍽	Board Device Core Type / Max Freg	FRDM-KL27Z MKL27Z644 Cortex-MDP / 48MHz
Review SDK Details Items listed on the side panel will be included in your SDK download. These selections can be edited using the Tools -> Configurations Settings page	Memory Size	64 KB Flash 16 KB RAM
This MCUXpresso SDK configuration is available for direct download	SDK Version: Host OS: Toolchain:	KSDK 2.2.1 Windows IAR Embedded Workbench for ARM
Download Now SDK_2.2.1_FRDM-KL27Z	Middleware:	USB Type-C PD stack, FreeRTOS
	Base SDK:	MCUXpresso SDK API Reference Manual

Figure 25. SDK Builder

10. Agree to the software terms and conditions.

P	MCUXpresso	OVERVIEW	TOOLS - MA	NAGE -						🕼 English 🗸	💄 Bryan 🗸	ψ ι
-eedback		Soff Please re EULA LA_OPT IMPOR This is a and NX code for docume License downlou Softwar continuu 1.	A the following agreement Software Content BASE_LICENSE v17 Do TANT. Read the following unidicate that you acce y, to bind your company Legal agreement betwee D J. ("NXP"). It concer m and any accompanyin natation relating to the Lice D J. ("NXP"). It concer m and any accompanyin natation relating to the Lice a to be governed by this J. DEFINITIONS 1.1. For NXP ti more than fifty percent (5	ent and click "I AG (Registry ecomber 2016 g NXP Software L by the terms of the to these terms. Yu en you, as an auth my your rights to u ng written materialis rensed Software p to the bound I Software. If you c control. Any copies Agreement. Your p he term "Affiliate" I	Condition REE" at the bottom be Agreement and you u ou may then download NXP SOFTWARE Orized representative so the software identifi a (the "Licensed Softw rovided to you by NXP up the terms of this Ag hange your mind later of the Licensed Softw virior use will also contil ncludes entities Contr stock, or decision-mail	ns afore downloading y acknowledge that y d or install the file. <u>LICENSE AGREEI</u> of your omployer, o. under this Licensec under this License under blue Lice ware that you but nue to be governed rolled by NXP where king authority in the	our software. ely. By selecting uu have the author <u>HENT</u> I flyou have no e Content Register Content Register Content Register Content Register Software may in in consideration tready distribute. by this Agreeme "Control" means event that there i	g the "I Accept" button at the e rifty, for yourself or on behalf of mployer, as an individual (tog and provided to you in binary clude any updates or error cor for NXP allowing you to acce the terms of this Agreement, d delete all copies of the Lice 3, where permitted, and do not nt. direct or indirect beneficial ov s no voting stock, in another e	end of this of your'). or source rrections or as the do not on to seed t destroy will whership of entity.			
	Privacy Policy Terms of	f Use Contact							© 2016	NXP Semiconducto	rs. All rights reserve	ed.
					•							

Figure 26. Software terms and conditions

11. Unzip the SDK to a folder (for example; *SDK_2.2_FRDM-KL27Z*).

Image: SDK_2.2.1_FRDM-KL27Z
SDK_2.2.1_FRDM-KL27Z.zip

Figure 27. SDK folder

Appendix B. SDK Builder FRDM-K64F

Use the online MCUXpresso Config Tools to create a custom SDK package for the FRDM-K64F board.

1. Open a web browser and navigate to the MCUXpresso homepage mcuxpresso.nxp.com.



Figure 28. MCUXpresso homepage

2. Click the "Login to view configurations" button to create a new configuration.



Figure 29. MCUXpresso Config Tools

MCUXpresso USB PD Migration Guide, Rev. 0, 10/2017

- 3. You are redirected to the <u>www.nxp.com</u> login page. Enter your account information or create a new account.
- 4. Back on the MCUXpresso homepage, click the drop-down box and select the "New Configuration" option.



Figure 30. Creating a configuration

5. Search for the board name (for example; FRDM-K64F).

Create a New Configuration

Search by device, board, kit name and filter by supported middleware.

Sear	ch by Name		
	FRDM-K64F	×	
Sele	at a Device, Board, or Kit		
	▼ Boards		
	FRDM-K64F		
	▼ Processors		(No configuration selected)
	▶ Kits		
Nam	e your configuration		
	Select Configuration Specify Additional Jump start your Configuration Settings configuration		

Figure 31. Creating a configuration

6. Choose a board from the list and provide a name for the configuration. Click the "Specify Additional Configuration Settings" button to select from FreeRTOS, IAR toolchain, and USB middleware.

Create a New Configuration

Search by device, board, kit name and filter by supported middleware.

FRDM-K64F	8
ect a Device, Board, or Kit	
▼ Boards	
FRDM-K64F	
▼ Processors	
Kits	
▶ Kits	

Hardware Details

Board Device Core Type / Max Freq Memory Size FRDM-K64F MK64F12 Cortex-M4F / 120MHz 1024 KB Flash 256 KB RAM

Figure 32. Creating a configuration

- 7. In the "Configuration Settings" section, set the following:
- Host OS \rightarrow Windows.
- Toolchain/IDE \rightarrow IAR Embedded Workbench for Arm.
- Middleware \rightarrow USB Stack, FreeRTOS.

	₹DM-K64F ¥	
Developer Environme	ent Settings	
elections here will impact	Toolchain / IDE	included in the SDK Download and Generated Projects
Windows	IAR Embedde	ed Workbench for ARM -
Select Optional Middl	eware	
Gelect Optional Middl Belections here will be incl	eware uded in your SDK download,	generated projects, and will impact Peripheral Tool settings
Select Optional Middl	eware uded in your SDK download,	generated projects, and will impact Peripheral Tool settings
Select Optional Middl Selections here will be incl 2 items selected	leware uded in your SDK download,	generated projects, and will impact Peripheral Tool settings Selected Middleware

Figure 33. Configuration settings

Or:

- Host OS \rightarrow Windows.
- Toolchain/IDE \rightarrow MCUXpresso IDE.
- Middleware \rightarrow USB Stack, FreeRTOS.

FRI	DM-K64F 💙	
eveloper Environmer	t Settings	
elections here will impact f	les and examples	projects included in the SDK Download and Generated Projects
Windowe	MCUN	
elect Ontional Middle	ware	
elect Optional Middle	ware ded in your SDK do	ownload, generated projects, and will impact Peripheral Tool settings
elect Optional Middle elections here will be includ 2 items selected	ware ded in your SDK do	ownload, generated projects, and will impact Peripheral Tool settings Selected Middleware

Figure 34. Configuration settings

8. After setting the configurations, click the "Go to SDK Builder" button.

Configuration S Specify included middleware	RTOS selections, and development preferences.	1	
Current Configuration		Hardware Details	
FRDM-K64F	*	Board	FRDM-K64F
		Device	MK64F12
Developer Environment Setting Selections here will impact files and ex	IS ramples projects included in the SDK Download and Generated Projects	Core Type / Max Freq Memory Size	Cortex-M4F / 120MHz 1024 KB Flash 256 KB RAM
Host OS	Toolchain / IDE		
Windows -	IAR Embedded Workbench for ARM 🔹 Set as defaul	t	
Select Optional Middleware Selections here will be included in your 2 items selected	SDK download, generated projects, and will impact Peripheral Tool settings		
	USB stack, FreeRTOS		
Return to Overview	Go to SDK Builder configuration		

Figure 35. Configuration settings

9. Click the "Download Now" button to download the SDK package.

NOTE

If you see the "Request to Build" button instead of the "Go to SDK Builder" button, click the "Request to Build" button. When the package is built, a notification to download it appears in the "SDK Archive" section.

SDK Builder Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.		
Current Configuration	Hardware Details	
FRDM-K64F ♥ Review SDK Details Items listed on the side panel will be included in your SDK download. These selections can be edited using the Tools -> Configurations Settings page	Board Device Core Type / Max Freq Memory Size SDK Details 💉 Edit	FRDM-K64F MK64F12 Cortex-M4F / 120MHz 1024 KB Flash 256 KB RAM
Click the link below to request this specific MCUXpresso SDK Build In general, SDK builds should complete within a few minutes. You will be notified via email and notifications in the upper right corner of this webpage. Package Name	SDK Version: Host OS: Toolchain: Middleware: Documentation	KSDK 2.2.0 Windows IAR Embedded Workbench for ARM USB stack, FreeRTOS
Request Build SDK_2.2_FRDM-K64F	Base SDK:	MCUXpresso SDK API Reference Manual

Figure 36. SDK Builder

10. To access the "SDK Archive" section, click the "MANAGE" tab and select the "SDK Archive" option.



11. Agree to the software terms and conditions.



12. Unzip the SDK package to a folder (for example; SDK_2.2_FRDM-K64F)





Appendix C. Using MCUXpresso Config Tools for FRDM-K64F pins' initialization

1. Open the MCUXpresso Config Tools. If you don't have this program installed, visit www.nxp.com/mcuxpresso/config and click the "DOWNLOADS" tab to download and install it.



Figure 40. MCUXpresso icon

2. The wizard asks whether you want to start the development with or without an SDK package. Choose to start the development with the SDK package and select the "Create new configuration" option.

Create a new configuration		
Select SDK Package		
Specified SDK path does not exist: ""		
Start development with the selected MG	UXpresso SDKv2 Package (SDK can be obtained at <u>http://mcu</u>	(presso.nxp.com)
Select SDK folder:	- Browse	
Oreate new configuration		
Use this option to create empty o	onfiguration for selected processor/board/kit/template or create	configuration from existing SDK example project.
Clone an example project		
Select this option if you want exa	nple project with all sources for selected toolchain. The project	will not be editable using MCUXpresso Config Tools.
 Start development without an MCUXpr 	sso SDK Package	C
Use this option if you have not downl	aded an SDK package yet. Tool will be limited to only Pins and	Clocks Tools. It is possible to specify SDK path later.
		Next Circle
	< Back	ivext > Cancel

Figure 41. Creating a configuration

3. Use the "Browse…" button to navigate to the location where you unzipped the FRDM-K64F SDK installation. Select the SDK top-level folder from your file system and click the "OK" button.

SDK_2.2.1_FRDM-KL27Z		
>], boards		
Della CMSIS		
b levices		
b 👢 docs		
b 👢 middleware		
> 👢 rtos		
It tools		=
b L SDK_2.2_FRDM-K22F		
Experimentary		
Iar-cmsismanager-workspace		
) Links		
b 👢 mcuxpresso		
Ny Documents		
Ny Music		-
Folder: SDK_2.2_FRDM-K64F		
	_	

Figure 42. Selecting the SDK folder

4. The wizard asks you whether to create a new configuration or clone an example project. Select the "Create new configuration" option and click the "Next" button to continue.

K Create a new configurati	n		
Select SDK Package			
Start development with	the selected MCUXpresso SDKv2 Package (SDK can be	obtained at <u>http://mcuxpresso.nxp.co</u>	om)
Select SDK folder: C	\nxp\MCUXpressoSDK\SDK-FRDM-KL27Z +	Browse	
Create new configu	ration		
Use this option to	create empty configuration for selected processor/boa project	rd/kit/template or create configuratio	n from existing SDK example project.
Select this option	if you want example project with all sources for selecte	d toolchain. The project will not be ed	litable using MCUXpresso Config Tools.
Start development with	out an MCUXpresso SDK Package		
Use this option if you	nave not downloaded an SDK package yet. Tool will be	limited to only Pins and Clocks Tools	. It is possible to specify SDK path later.
		< Back Next >	Finish Cancel

Figure 43. Selecting the SDK package

5. Select the "hello_world" option inside the "Name your configuration" section, rename it to "usbpd_k64", and click the "Finish" button.



Create a new configuration		
Create a new configuration		
Select Processor/Board		
type filter text		
Examples cmsis_driver_examples adcl5[ow_power bubble dac_adc ecompass fmobt_adcl6 ffmquad_decoder hello_wordid hello_wordid mbedtls/mbedtls_benchmark mbedtls/mbedtls_beftest		
Select version		
SDK v2 +		
Name your configuration		
usbpd_k64f		
	- Deale Marcha	Canal Canal

Figure 44. Selecting the "hello_world" option

6. If necessary (it may already be open), open the pins tool by selecting Tools → Pins from the toolbar.

Tools Pins Views Help
Clocks
Pins
8 Project Generator

Figure 45. Opening the Pins tool

7. Use Table 2 to locate the FRDM-K64F pin assignment for the power request and power change switches. Route SW2 (PTC6) and SW3 (PTA4) as the GPIO inputs with the pull-up enabled. Search for PTC6 in the "Pins" view. Click the box under the GPIO column.

🖲 *Pins - u	sbpd_k64.mex (MK64FN1M	Dxxx12)	
File Edit	Tools Pins Views Hel	р	
🗄 Pins 🖾	🜵 Peripherals		- 0
- 50	PTC6		
Pin	Pin name	Label	Identifie
78	CMP0_IN0/PTC6/LLW	U8[11]/SW2	SW2;AC

Figure 46. Pins tool

8. A new window pops up, showing all signals available on this pin. Select the GPIO functionality and click the "Done" button.



Figure 47. Available signals

9. Inside the "Routed Pins" tab, modify the PTC6 settings. Change the pin identifier to SW2 by right-clicking the identifier box and select the "Label" and "Identifier" tabs. Write "SW2" into the "Identifier" text box and click the "OK" button. Configure the input direction and select the "Pullup" in the "Pull select" tab and the "Enabled" option in the "Pull enable" tab.

oute	d Pins													
filte	rted													
ute	d Pins 🔘	3 0	•											
	Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate	Open drain	Drive strength	Pull select	Pull enable	Passive filter	Digital filter	
	UARTO	RX	UARTO RX	U7[4]/UARTO RX	DEBUG UART RX	Input	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a	
	UARTO	TX	UARTO_TX	U10[1]/UART0_TX	DEBUG_UART_TX	Not Specified	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a	
3	GPIOC	GP10, 6	PTC6	U8[11]/SW2	SW2	Input	Slow	Disabled	Low	Pullup	Enabled	Disabled	n/a	
					SW2									
					ACCEL_INT1 Pin	identifier used t	or Edefine co	de generation. U	se Pins view table	to define it.				
					Not Specified	J		ar generation of						

Figure 48. Modifying PTC6 settings

- 10. Repeat steps 7 to 9 to configure SW3 (PTA4).
- 11. Use Table 1 to locate the FRDM-K64F pin assignment for EXTRA_EN_SRC (PTB23). Route the PTB23 as a GPIO output. Search for PTB23 in the pins view. Click the box under the "GPIO" column. A new window pops up and displays all signals available on this pin. Select the "GPIO" functionality and click the "Done" button.
- 12. In the "Routed Pins" tab, change the pin identifier to EXTRA_EN_SRC by right-clicking the "Identifier" box and selecting the "Label" and "Identifier" tabs. Write "EXTRA_EN_SRC" into the "Identifier" text box and click the "OK" button. Change the direction to "Output".

THE	er text													
ute	d Pins 🔘	5 🔘 🦉												
	Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate	Open drain	Drive strength	Pull select	Pull enable	Passive filter	Digital filter	
2	UARTO	RX	UART0_RX	U7[4]/UARTO_RX	DEBUG_UART_RX	Input	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a	
3	UARTO	TX	UART0_TX	U10[1]/UART0_TX	DEBUG_UART_TX	Not Specified	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a	
78	GPIOC	GP10, 6	PTC6	U8[11]/SW2	SW2	Input	Slow	Disabled	Low	Pullup	Enabled	Disabled	n/a	
38	GPIOA	GP10, 4	PTA4	SW3	SW3	Input	Slow	Disabled	Low	Pullup	Enabled	Disabled	n/a	
59	GP10B	GP10, 23	PTB23	J1[10]	EXTRA_EN_SRC	Output	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a	

Figure 49. Changing the pin identifier

13. Use Table 1 to locate the FRDM-K64F pin assignment for nALERT (PTC12). Route PTC12 as a pin input. Search for PTC12 in the pins view. Click the box under the "GPIO" column. A new window pops up and displays all signals available on this pin. Select the GPIOC functionality and click the "Done" button.



Figure 50. Available signals

14. Inside the "Routed Pins" tab, change the pin identifier to nALERT by right-clicking the "Identifier" box and select the "Label" and "Identifier" tabs. Write "nALERT" into the Identifier text box and click the "OK" button. Edit the "Mode" setting to "PullUp".

Rout	ed Pins												- 6
ype filt	er text												
Route	d Pins 🕒	6	N N										
#	Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate	Open drain	Drive strength	Pull select	Pull enable	Passive filter	Digiti
62	UART0	RX	UART0_RX	U7[4]/UART0_RX	DEBUG_UART_RX	Input	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a
63	UART0	TX	UART0_TX	U10[1]/UART0_TX	DEBUG_UART_TX	Not Specified	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a
78	GPIOC	GPIO, 6	PTC6	U8[11]/SW2	SW2	Input	Fast	Disabled	Low	Pullup	Enabled	Disabled	n/a
38	GPIOA	GPIO, 4	PTA4	SW3	SW3	Input	Fast	Disabled	High	Pullup	Enabled	Disabled	n/a
69	GPIOB	GPIO, 23	PTB23	J1[10]	EXTRA_EN_SRC	Output	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a
84	GPIOC	GPIO 12	PTC12	12[2]	DALERT	Input	Fact	Disabled	Low	Pullup	Enabled	Disabled	nla

Figure 51. "Routed Pins" tab

MCUXpresso USB PD Migration Guide, Rev. 0, 10/2017

15. Inside the "Routed Pins" tab, add a new function by clicking the "+" icon at the bottom of this tab.

					Mar out - and - an	AND ALL AND AL	PIRACUTE CLUMA ADDR VSBA CMPT_ANDPTAN2 PTANETER_PCCK	anos Anos Anos Anos Anos Anos Anos Anos A	J				
🔁 Rou	ted Pins												
type f	ilter text												
Rout	ed Pins 🔘	6 🔘											
	Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate	Open drain	Drive strength	Pull select	Pull enable	Passive filter	Digital f
62	UARTO	RX	UARTO_RX	U7[4]/UART0_RX	DEBUG_UART_RX	Input	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a
	UARTO	TX	UART0_TX	U10[1]/UART0_TX	DEBUG_UART_TX	Not Specified	Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a
63		GPIO, 6	PTC6	U8[11]/SW2	SW2	Input	Fast	Disabled	Low	Pullup	Enabled	Disabled	n/a
63 78	GPIOC							a local de la	Histo	Duttern	Easthlad	Disabled	210
63 78 38	GPIOC	GPIO, 4	PTA4	SW3	SW3	Input	Fast	Disablea	High	Pullop	Endbled	Disablea	nyu
63 78 38 69	GPIOC GPIOA GPIOB	GPIO, 4 GPIO, 23	PTA4 PTB23	SW3 J1[10]	SW3 EXTRA_EN_SRC	Output	Fast Fast	Disabled	Low	Pulldown	Disabled	Disabled	n/a
63 78 38 69 84	GPIOC GPIOA GPIOB GPIOC	GPIO, 4 GPIO, 23 GPIO, 12	PTA4 PTB23 PTC12	SW3 J1[10] J2[2]	SW3 EXTRA_EN_SRC nALERT	Input Output Input	Fast Fast Fast	Disabled Disabled Disabled	Low Low	Pulldown Pullup	Disabled Disabled	Disabled Disabled Disabled	n/a n/a

Figure 52. Adding a new function

- 16. Change the name of the recently created function to I2Cn_InitPins, where **n** corresponds to the (D14, D15) I2Cn number provided in Table 1. For the FRDM-K64F platform, the function should be named I2C0_InitPins. To rename the function, right-click its current name, then click the "Properties" option, write "I2C0_InitPins" into the "Function name" text box, and click the "OK" button.
- 17. Repeat step 16 for I2Cn_DeinitPins.
- 18. Use Table 1 to locate the FRDM-K64F pin assignment for PTN5110_SDA (PTE25) and PTN5110_SCL (PTE24).
- 19. Select the I2C0_InitPins function and route the PTN5110_SDA and PTN5110_SCL as the I²C pins. Search for "PTE25" in the "Pins" view and click the box under the GPIO column. A new window pops up and displays all signals available on this pin. Select the I2C0_SDA functionality and click the "Done" button.

Pins -	usbpd_k64.mex (MK64	FN1M0xx12)																
Pins 🖾	Peripherals	пер	- 0	Package 1													Q	0.00
141	pte25									وي المار	li ul	11 12						
Pin ^	Pin name	Label	Identify							PICE CONTRACT	200	NA_B NA_B	PTC22					
32	12C0_SDA	J2[18]/U8	(6]/12C ACCEL							10,000 10,00000 10,0000 10,00000000	TEN. TEN. TEN. TEN. TEN. TEN.	1, 187 1, 187 1, 187 1, 187 1, 187 1, 188 1, 188 1, 188 1, 188 1, 188	ALINA P					
	🕅 pin [32]	83								ENNELEN	EFFF	DEF DEF	AND DEL					
	All signals on pin [32]:												E .				1
	ADC0_SE18 (4	4DC0,SE,18)						ADC1 SE4#/PT	w E	ADCO	ADCI	CAND		VOD124				
	EWM_IN (EW	M,IN)						ADC1_SESa/PT ADC0_D	E1/	CMPU	DACO	CMP2 DASA		VSS123 CMP1_IN1/	TCM			
	PTE25 (GPIOE	.GP10,25)						AD CO_DI PTEALLINU	P2/	ENET	BW	EzPer		ADDD_SEM	VFTC1/			
	UART4_RX (U	ART4,RXQ						PTES/SPI1_PC	\$2/	FB FTM2	FRE	FTMI		ADC0_SE14	VPTCO/			
								V	0016	GPIOB	GFIOC	6/101		PT822/5P/2 PT821/5P/2	SOUT/			
	- C	Done						USB	D DP	GPIOE	1200	1201		PTB20/SPI2 PTB20/SPI2	PCSIV			
		Done						VO	UT33	LDWU	LPTMPD	010		PTBILICAN	0_TXF			
								ADCB	OP1	P080	RCM	RTC		UARTD_RX				
								ADC1	OP1	SDHC	SM	SPID	LY	VSS100	APTRO 14			
				Routed Pin	s			- Alter						- Hear Mark				··· 🗆
				type filter test														
				Routed Pins	0 1	0~~												
					Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate	Open drain	Drive stren	Pull select	Pull enable	Passive filter	Digital filter	
				32	12C8	SDA	I2C0_SDA	J2[18]/U8[_	Not Specif	input/Out	Fast	Disabled	Low	Pulidown	Disabled	Disabled	n/a	
*	(M))			BOARD_InitP	ins [2C1]	nitPins 33 I2C	1_DeinitPins	0										

Figure 53. Available signals

- 20. In the "Routed Pins" tab, change the "Drive strength" to "High".
- 21. Repeat steps 24 and 25 for the I2C4_SCL signal.

MCUXpresso USB PD Migration Guide, Rev. 0, 10/2017

🗗 Rou	ted Pins													- 5
type fi	lter text													
Rout	ed Pins 🕒	2	▲ ∨							_				
#	Peripheral	Signal	Route to	Label	Identifier	Direction	Slew rate	Open drain	Drive strength	Pull select	Pull enable	Passive filter	Digital filter	
32	12C0	SDA	I2C0_SDA	J2[18]/U8[6]/I2C0_SDA	PTN5110_SDA	Input/Output	Fast	Disabled	High	Pulldown	Disabled	Disabled	n/a	
31	12C0	SCL	I2C0_SCL	J2[20]/U8[4]/I2C0_SCL	PTN5110_SCL	Input/Output	Fast	Disabled	High	Pulldown	Disabled	Disabled	n/a	
										-				
			M 1200 D : 14											
BOAR	D_InitPins	12CU_InitPins	20 Demit	Pins U										

Figure 54. Changing drive strength

22. Select the I2C0_DeinitPins function and route PTN5110_SDA and PTN5110_SCL as GPIOs. Search for PTE25 in the pins view and click the box under the GPIO column. A new window pops up and displays all signals available on this pin. Tick the PTE25 GPIO checkbox and click the "Done" button.

E Prins 24	Peripheral	s [- 0	O Package 33														QQ0 *	-
100	PTE25								ADID_DMD/	ENET	DIM	B Pol		4000_164	M				
Rin *	Bin erme	Label	Mandidian					PTE	BrSPIT_PCS2/	FB	FM	FTM		ADC0_SET	APTOD/				
FIN I	Pin name	Label (1996)	apentiner					20	VODIN	GPIDE	GRIDC	GPIO	0	PT822/1PG	SOUTH				
2 52	PIEZJ	witeh optoherer.	PTINDILIU_SUA						USRC.DP	- GPIDE	000	1201		PT820/1PG	PCSER				
									U100_DH#	1202	1250	JTAD		PTB16/CAP PTB16/CAP	0_F0/ 0_T0/				
									VREDIN-	LDMU RDM	LPIMRD	OSC BTC		UARTE_TX					
	NO	a [27]	152						AD CO_DMI	SDHC	SM	SFID		V00112					
	200 10	an gang	[445]						ADC1_DM1	SPIT	590	SUPP	4.9	ADC1_SET	SPTB11/				
	Aver 50	dues ou biu (253							ADCD_DPA/	System Control	THU	C UAR	9	ADC1_1E1 PTROFILE1	PCSV				
		ADC0_SEL8 (ADC0,SE,	8)						ADC1_DRM	GARTI	CINARITS	UAR	3	ADCD_SET	VPTBO/				
		EWM_IN (EWM_IN)							VDDA-	VEF	0100	C USIN	~	ADCD_SES	in the second				
		UARTE RX (UARTE RX		Rauted Pins					VILA	MK64EN1N NK64EN1N NK64EN1N NK64EN1N NK64EN1N NK64EN1N	10VLL12-		ackage 2005 2005 2005 2005 2005 2005 2005 200		9V				÷.,
				Routed Pins			Pouteto	[shel	Mantifias	Direction	Sieurate	Ones dais	Drive steen	Dull relact	Dull apphie	Dansies Ober	Dialtal Otar		
				22	respirerat	Signal	ATE 15	Laber	ETABLIC CDA	Aird Facel	SIEWTREE	Open chain	Unive stren	Puil select	Pull enabled	Passive ritter	Dignaliniter		
				34	UNUE	0990, 25	PILO	Mispos.	PINGINGSUM	TVOR Specific	Fost	Display	LOW	Publicown	Daabura	Disabled	eve .		

Figure 55. Available signals

- 23. Repeat the previous step for the PTN5110_SCL GPIO signal.
- 24. Repeat steps 19-23 for the A4-A5 routed pins.
- 25. Export the *pin_mux.c* and *pin_mux.h* files by clicking the "Source" tab on the right side and clicking the "export" icon.

🔐 Registers 📄 Log 🖸 Sources 🛛 🗖 🗖	
pin_mux.c pin_mux.h	
/* * TEXT BELOW IS USED AS SETTING FOR TOOLS ***********************************	
<pre>pin_iddels: {pin_undels: {pin_undels: {pin_num: '60', pin_signal: PT62/LLWU_P8/SPI0_PCS5/F8_AD28, label {pin_num: '76', pin_signal: PT64/LLWU_P8/SPI0_PCS0/UART1_TX/FTM0_Ct {pin_num: '32', pin_signal: ADC0_SE18/PTE25/UART4_TX/I2C0_SCA/EWM1_ {pin_num: '31', pin_signal: ADC0_SE18/PTE24/UART4_TX/I2C0_SCL/EWM1_ * BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOLS */</pre>	
#include "fsl_common.h" #include "fsl_port.h" #include "pin_mux.h"	

Figure 56. Sources tab

26. Select the folder to export the *pin_mux.c* and *pin_mux.h* files to. For the usb_pd_freertos project, select the ...*boards\frdmk64f\usb_examples\usb_pd\freertos* folder and click the "Finish" button.

Export	
Export Pins Source Files	
To directory: 2.2_FRDM-K64F\boards\frdmk64f\usb_examples\usb_pd\ Cortex-M4F	freertos Browse
Finish	Cancel

Figure 57. Export window

How to Reach Us:

Home Page: nxp.com Web Support: nxp.com/support

arm

Information in this document is provided solely to enable system and software implementers to use NXP 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. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, and Kinetis are trademarks of NXP B.V. IAR Embedded Workbench is a registered trademark owned by IAR Systems AB. All other product or service names are the property of their respective owners.

Arm, the Arm logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2017 NXP B.V.

Document Number: USBPDMUG Rev. 0 10/2017

