Configure the board pin multiplexer FRDM K64F

March 6, 2018

- 1. Add a new pin
 - Open a K64F project in MCUXpresso. (We are using the example frdm64f_blink_led from the skeleton code here).
 - Right click the project name in the project explorer pane. Select "MCUXpresso Config Tools" \rightarrow "Open Pins". This will open the Pin

0	Сору	њU	
	🛅 Paste	жV	
🍟 🔓 Project 🛿 🛃 P	🔀 Delete	\mathbf{X}	nain.c 🚺 main.c 🔀
(x)=	Source	•	2⊕ * The Clear BSD License
k k k k k k k k k k k k k k k k k k k	Move		4 (*Suchem includes */
► Frdmk64f_blink	Rename	F2	5 /*System includes.*/
Image: Second secon	≧ Import ☑ Export		7 8 /* Kernel includes. */ 9 #include "FreeRTOS.h"
	Build Project Clean Project Refresh Close Project Close Unrelated Projects	F5	<pre>0 #include "timers.h" 1 2 /* Freescale includes. */ 3 #include "fsl_device_registers.! 4 #include "fsl_debug_console.h" 5 #include "board.h" 6 #include "fsl_pit.h" /* period 7</pre>
	Build Configurations Build Targets Index	* * *	8 #include "pin_mux.h" 9 #include "clock_config.h" 0 1
	Validate Run As Debug As Profile As Restore from Local History Launch Configurations Smart update Utilities Tools Hanage SDK Components	* * * * * *	<pre>2 * Periodic Interrupt Timer (PI * define PIT_IRQ_ID PIT0_IRQn 5 #define PIT_SOURCE_CLOCK for PIT dri # define PIT_SOURCE_CLOCK CLOCK_ volatile bool pitIsrFlag = fals volatile uint32_t systime = 0; 2 /************************************</pre>
	MCUXpresso Config Tools		Open Pins
	⅔ Run C/C++ Code Analysis Team Compare With Configure Source	* * * *	<pre>Un Open Clocks</pre>
frdmk64f_blink_led	Properties	жI	

editing screen.

• Select "Peripheral Signals". There should be subfolders for GPIOA, GPIOB, GPIOC, GPIOD, GPIOE each with a set of pins. Select "GPIOB $\rightarrow 23$. [we chose GPIOB \rightarrow PTB23 you can choose any pin

you like].						
📰 Pins 🐼 Peripheral Signals 🕱 🗖 🗖	Package X			QQ(🕈 🛱 🗖 🗖		
Image: Second state state Image: Second state Image: Second state Image: Second s	VUU 3-3 VERCIM ADC0_DPH ADC0_DPH ADC0_DPH ADC1_SELSFFEI1/ ADC1_SELSFFEI1/					
GPIO, 9 » [57] PTB9/SPIT_PCS1/C GPIO, 10 » [58] ADC1_SE14/PTB1	B Routed Pins					
GPIO, 11 » [59] ADC1_SE15/PTB1	Q type filter text					
GPIO, 16 » [62] PTB16/SPI1_SOU GPIO, 17 » [63] PTB17/SPI1_SIN/L	Routed Pins for BOARI	D InitPins 5				
GPIO, 18 * [64] PTB18/CAN0_TX/f GPIO, 19 * (65] PTB19/CAN0_RX/I GPIO, 20 * [66] PTB20/SPI2_PCS(GPIO, 21 * [67] PTB21/SPI2_SCK/ GPIO, 22 * [68] PTB22/SPI2_SOU' GPIO, 23 * [66] PTB22/SPI2_SOU' GPIO, 23 * [66] PTB22/SPI2_SIN/5 GF Ceneral purpose I/O pin 23	# Peripheral Sig 62 UARTO RX 63 UARTO TX 33 GPIOE GI 67 GPIOB GI 68 GPIOB GI	gnal Route to X UART0_RX VART0_RX UART0_TX PIO, 26 PTE26 PIO, 21 PTB21 PIO, 22 PTB22	Label U7[4]/UART0_RX U10[1]/UART0_TX J2[1]/D12[4]/LEDRGB_GREEN D12[3]/LEDRGB_BLUE D12[3]/LEDRGB_RED	Identifier Direction DEBUG_UART_RX Input DEBUG_UART_TX Not Specif LED_GREEN Not Specif LED_RED Not Specif		
Proble Non pin routed						

- GPIO, 23 (should show up in the "Routed Pins" Window after you click the green check box.
- Click "Export" in the sources pane. Save and overwrite pin_mux.c and pin_mux.h files to project_location/project_name/board

👪 Registers 🗟 Sour	ces 🛛 🗖 🗖			
i Generated code preview				
pin_mux.c pin_mux.h				
<pre>/************************************</pre>				
#ifndef _ PIN_MUX_H_ #define _ PIN_MUX_H_				
/*************************************	*******************			
<pre>/*! @brief Direction type */ typedef enum _pin_mux_direction {</pre>				
/*! * @addtogroup pin_ * @{ */	mux			
/*************************************	***************			
<pre>#if defined(cplusplus) extern "C" {</pre>				

- Refresh pin_mux.c and pin_mux.h by right clicking them in the project explorer and choosing "refresh".
- You should be able to access the pin in your software. See our skeleton

code for toggling a GPIO pin here for an example.