MASARYK UNIVERSITY
Czech Republic

PV198 – One-chip Controllers

GPIO – LED & Button

Content

1. **What is GPIO**
2. What is it used for
3. How does it work
    1. Switch debouncing
4. Application
    1. Output – LED using SDK example
    2. Input – Button using Config Tools

What is GPIO

- **GPIO** – **G**eneral **P**urpose **I**nput **O**utput
- Direct control of pins of the MCU
- Basic interaction with external world
- Can be programmed as Input or Output
- Has only 2 states (logic 0, logic 1)

Content

1. What is GPIO
2. **What is it used for**
3. How does it work
    1. Switch debouncing
4. Application
    1. Output – LED using SDK example
    2. Input – Button using Config Tools

What is it used for

- Anything that works with 2 states – on/off

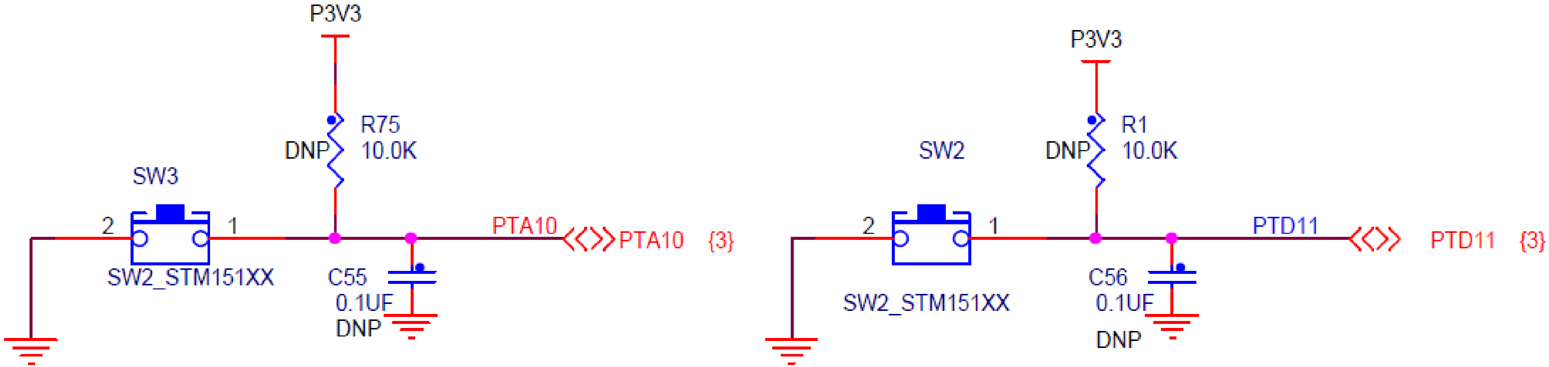
- LED
- Buttons
- Sensors
- And used by more sophisticated peripherals

Content

1. What is GPIO
2. What is it used for
3. **How does it work**
    1. Switch debouncing
4. Application
    1. Output – LED using SDK example
    2. Input – Button using Config Tools

How buttons on the board work

- Connects pit to ground (logic 0) or to voltage (logic 1)



How does it work – Switch debouncing

- Bouncing
  - Looks like button is pressed multiple times
  - Cause by mechanical contact of the switch

How doeas it work – Switch debouncing

- Bouncing
  - Looks like button is pressed multiple times
  - Cause by mechanical contact of the switch
- Solution
  - HW debounce(add capacitor)
  - SW debounce(wait few miliseconds)

Content

1. What is GPIO
2. What is it used for
3. How does it work
    1. Switch debouncing
4. **Application**
    1. Output – LED using SDK example
    2. Input – Button using Config Tools

Application

Steps required to create an application:

- Initialize(MCUXpresso Configuration Tools help here)
    - Pin
    - Clocks
    - Peripherals
- Write application code

Application

1. **Write everything from scratch**

   □Error-prone , time demanding, tedious

2. **USE SDK example**

   ✓□Works out-of-box

   □More difficult to modify

3. **USE Config Tools**

   ✓□Easy to use and modify

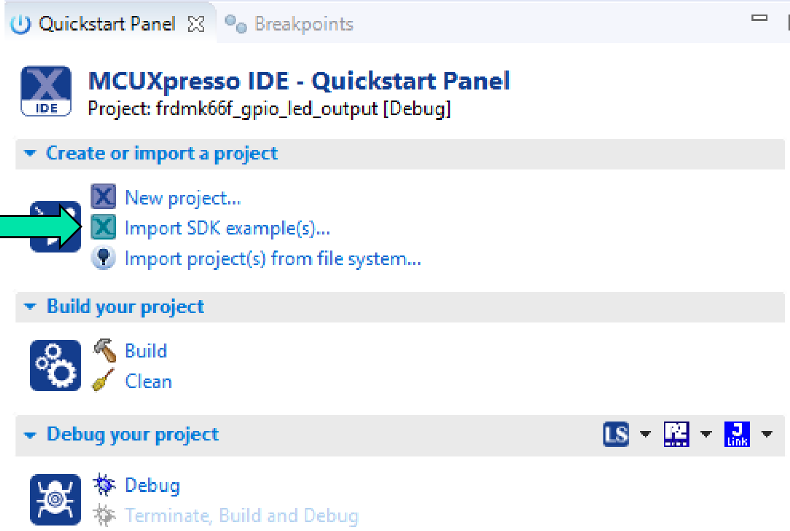
Content

1. What is GPIO
2. What is it used for
3. How does it work
    1. Switch debouncing
4. **Application**
    1. **Output – LED using SDK example**
    2. Input – Button using Config Tools

Application – LED using SDK example

Import SDK example “gpio_led_output”

- Select import SDK examples(s)…



Quickstart Panel | Breakpoints

MCUXpresso IDE - Quickstart Panel
Project: frdmk66f_gpio_led_output [Debug]

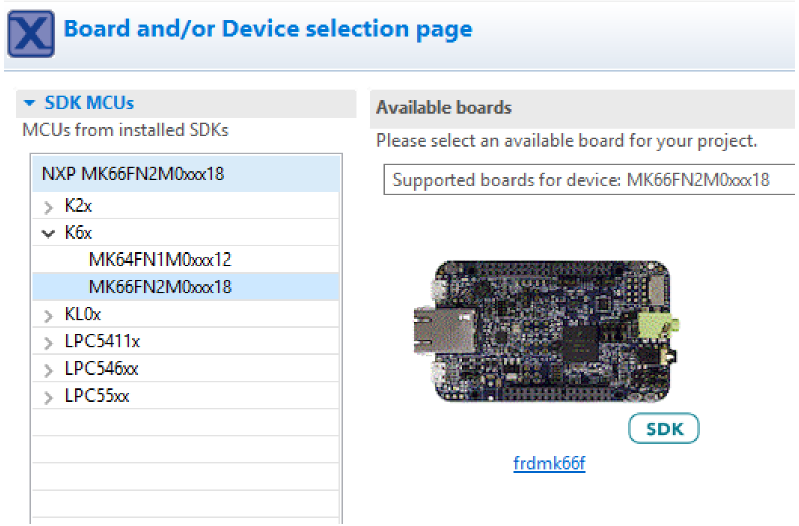
Create or import a project
- New project...
- Import SDK example(s)...
- Import project(s) from file system...

Build your project
- Build
- Clean

Debug your project
- Debug
- Terminate, Build and Debug

Application – LED using SDK example

- In the SDK wizard unfold the K6x, select the MK66FN2… and click on the board image



Board and/or Device selection page

SDK MCUs

MCUs from installed SDKs

NXP MK66FN2M0xxx18

- K2x
- K6x
  - MK64FN1M0xxx12
  - MK66FN2M0xxx18
- KL0x
- LPC5411x
- LPC546xx
- LPC55xx

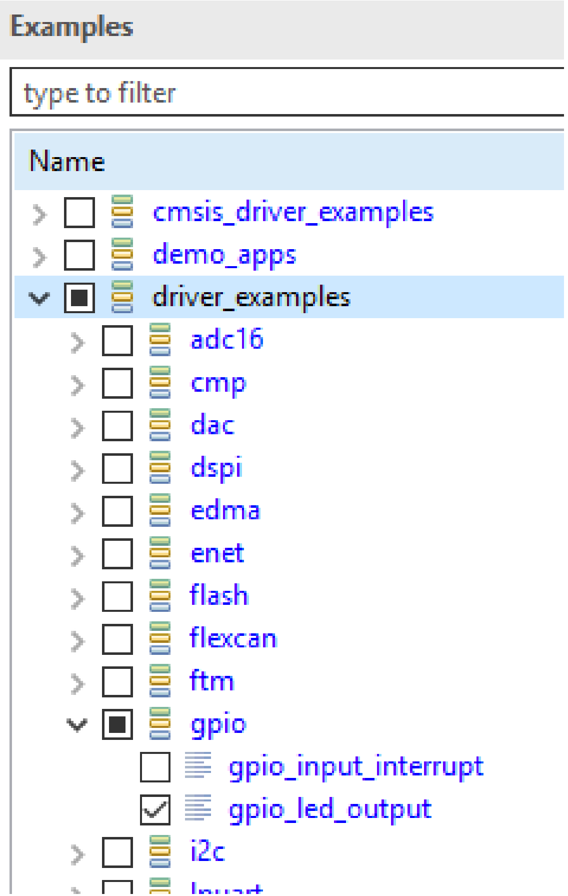
Available boards

Please select an available board for your project.

Supported boards for device: MK66FN2M0xxx18

Application – LED using SDK example

- Unfold *driver_examples -> gpio*
- Select *“gpio_led_output”* example
- Click Finish



Examples

type to filter

Name

- › □ cmsis_driver_examples
- › □ demo_apps
- ⌄ ■ driver_examples
  - › □ adc16
  - › □ cmp
  - › □ dac
  - › □ dspi
  - › □ edma
  - › □ enet
  - › □ flash
  - › □ flexcan
  - › □ ftm
  - ⌄ ■ gpio
    - □ gpio_input_interrupt
    - ☑ gpio_led_output
  - › □ i2c
  - › □ lpuart

Application – LED using SDK example

- Go through code together in detail
  - Pins, clocks are already configured
  - GPIO_PinInit
  - GPIO_PortToggle

Content

1. What is GPIO
2. What is it used for
3. How does it work
    1. Switch debouncing
4. **Application**
    1. Output – LED using SDK example
    2. **Input – Button using Config Tools**

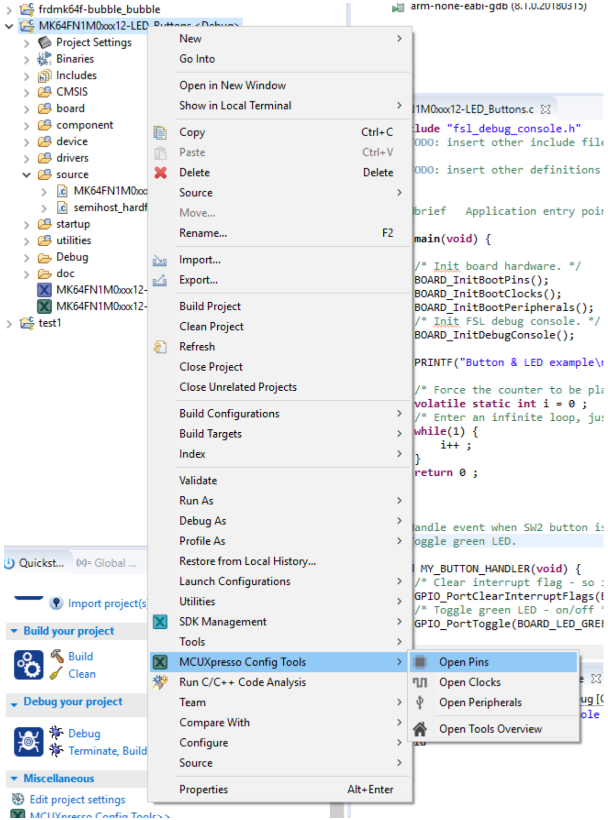
Application – Button using Config Tools

Goal:

- Press SW2 button to print text into console

Application – Button using Config Tools

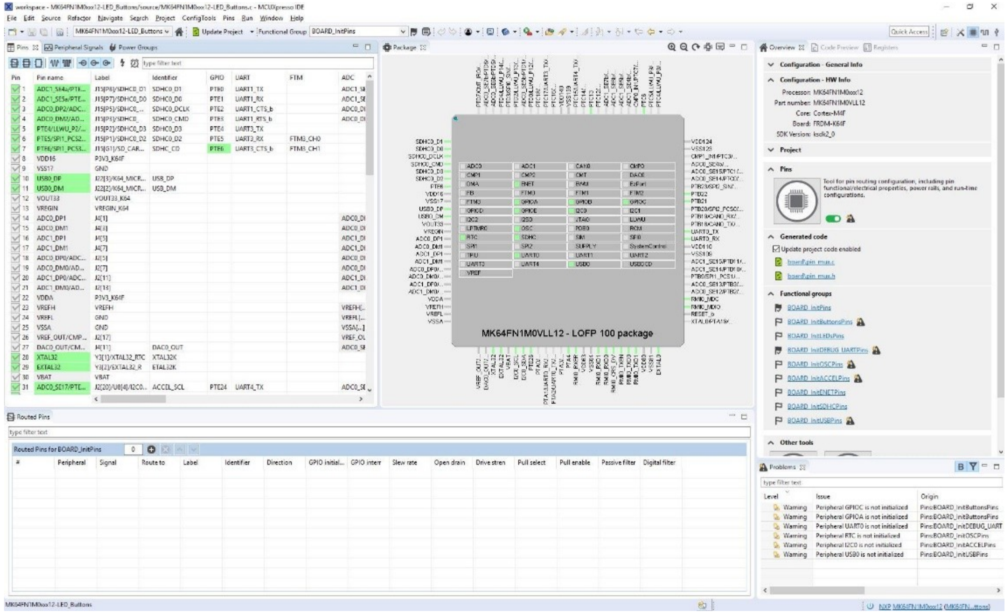
- Create new project
- Open Config Tools



```
lude "fsl_debug_console.h"
ODO: insert other include file

ODO: insert other definitions


brief   Application entry poi

main(void) {

/* Init board hardware. */
BOARD_InitBootPins();
BOARD_InitBootClocks();
BOARD_InitBootPeripherals();
/* Init FSL debug console. */
BOARD_InitDebugConsole();

PRINTF("Button & LED example\

/* Force the counter to be pla
volatile static int i = 0 ;
/* Enter an infinite loop, ju
while(1) {
    i++ ;
}
return 0 ;



andle event when SW2 button i
oggle green LED.

 MY_BUTTON_HANDLER(void) {
/* Clear interrupt flag - so 
GPIO_PortClearInterruptFlags(
/* Toggle green LED - on/off 
GPIO_PortToggle(BOARD_LED_GRE
```

Application – Button using Config Tools

- You should see Pin tool now:



workspace - MK64FN1M0xxx12-LED_Buttons/source/MK64FN1M0xxx12-LED_Buttons.c - MCUXpresso IDE

Pin	Pin name	Label	Identifier	GPIO	UART	FTM	ADC
☑ 1	ADC1_SE4a/PTE...	J15[P8]/SDHC0_D1	SDHC0_D1	PTE0	UART1_TX		ADC1_SE
☑ 2	ADC1_SE5a/PTE...	J15[P7]/SDHC0_D0	SDHC0_D0	PTE1	UART1_RX		ADC1_SE
☑ 3	ADC0_DP2/ADC...	J15[P5]/SDHC0_...	SDHC0_DCLK	PTE2	UART1_CTS_b		ADC0_DI
☑ 4	ADC0_DM2/AD...	J15[P3]/SDHC0_	SDHC0_CMD	PTE3	UART1_RTS_b		ADC0_DI
☑ 5	PTE4/LLWU_P2/...	J15[P2]/SDHC0_D3	SDHC0_D3	PTE4	UART3_TX		
☑ 6	PTE5/SPI1_PCS2...	J15[P1]/SDHC0_D2	SDHC0_D2	PTE5	UART3_RX	FTM3_CH0	
☑ 7	PTE6/SPI1_PCS3...	J15[G1]/SD_CAR...	SDHC_CD	PTE6	UART3_CTS_b	FTM3_CH1	
☑ 8	VDD16	P3V3_K64F					
☑ 9	VSS17	GND					
☑ 10	USB0_DP	J22[3]/K64_MICR...	USB_DP				
☑ 11	USB0_DM	J22[2]/K64_MICR...	USB_DM				
☑ 12	VOUT33	VOUT33_K64					
☑ 13	VREGIN	VREGIN_K64					
☑ 14	ADC0_DP1	J4[1]					ADC0_DI
☑ 15	ADC0_DM1	J4[3]					ADC0_DI
☑ 16	ADC1_DP1	J4[5]					ADC1_DI
☑ 17	ADC1_DM1	J4[7]					ADC1_DI
☑ 18	ADC0_DP0/ADC...	J2[5]					ADC0_DI
☑ 19	ADC0_DM0/AD...	J2[7]					ADC0_DI
☑ 20	ADC1_DP0/ADC...	J2[11]					ADC1_DI
☑ 21	ADC1_DM0/AD...	J2[13]					ADC1_DI
☑ 22	VDDA	P3V3_K64F					
☑ 23	VREFH	VREFH					VREFH(...
☑ 24	VREFL	GND					VREFL[...
☑ 25	VSSA	GND					VSSA[...]
☑ 26	VREF_OUT/CMP...	J2[17]					VREF_OL
☑ 27	DAC0_OUT/CM...	J4[11]	DAC0_OUT				ADC0_SE
☑ 28	XTAL32	Y3[1]/XTAL32_RTC	XTAL32K				
☑ 29	EXTAL32	Y3[2]/EXTAL32_R	ETAL32K				
☑ 30	VBAT	VBAT					
☑ 31	ADC0_SE17/PTE...	J2[20]/U8[4]/I2C0...	ACCEL_SCL	PTE24	UART4_TX		ADC0_SE

Routed Pins for BOARD_InitPins 0

#	Peripheral	Signal	Route to	Label	Identifier	Direction	GPIO initial...	GPIO interr	Slew rate	Open drain	Drive stren	Pull select	Pull enable	Passive filter	Digital filter

Configuration - General Info

Configuration - HW Info

Processor: MK64FN1M0xxx12
Part number: MK64FN1M0VLL12
Core: Cortex-M4F
Board: FRDM-K64F
SDK Version: ksdk2_0

Project

Pins

Tool for pin routing configuration, including pin functional/electrical properties, power rails, and run-time configurations.

Generated code

☑ Update project code enabled

board/pin_mux.c
board/pin_mux.h

Functional groups

BOARD_InitPins
BOARD_InitButtonsPins
BOARD_InitLEDsPins
BOARD_InitDEBUG_UARTPins
BOARD_InitOSCPins
BOARD_InitACCELPins
BOARD_InitENETPins
BOARD_InitSDHCPins
BOARD_InitUSBPins

Other tools

Problems

Level	Issue	Origin
Warning	Peripheral GPIOC is not initialized	Pins:BOARD_InitButtonsPins
Warning	Peripheral GPIOA is not initialized	Pins:BOARD_InitButtonsPins
Warning	Peripheral UART0 is not initialized	Pins:BOARD_InitDEBUG_UART
Warning	Peripheral RTC is not initialized	Pins:BOARD_InitOSCPins
Warning	Peripheral I2C0 is not initialized	Pins:BOARD_InitACCELPins
Warning	Peripheral USB0 is not initialized	Pins:BOARD_InitUSBPins

MK64FN1M0xxx12-LED_Buttons

NXP MK64FN1M0xxx12 (MK64FN...ttons)

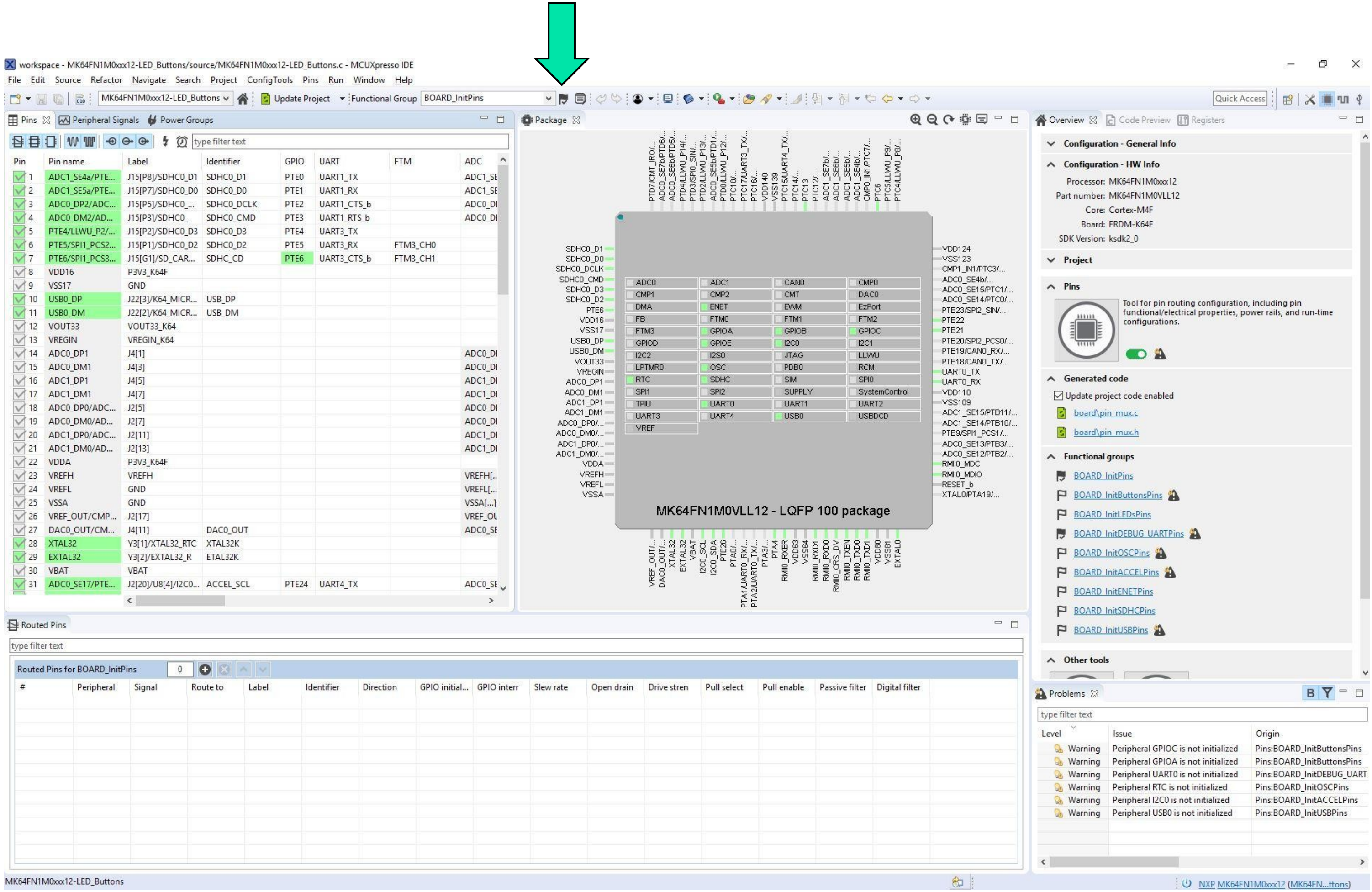
Application – Initialization

How configuration tools can help us:

- Modify settings easily
- Visual representation of a configuration
- Great for custom boards (our board already has a lot of useful DEFINEs, which is not a case when new board is created)

Application – Button using Config Tools

- Pins tool contains predefined configurations
- Open dialog with functional groups by clicking on the icon


workspace - MK64FN1M0xxx12-LED_Buttons/source/MK64FN1M0xxx12-LED_Buttons.c - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project ConfigTools Pins Run Window Help
MK64FN1M0xxx12-LED_Buttons
Update Project
Functional Group
BOARD_InitPins
Quick Access
Pins
Peripheral Signals
Power Groups
type filter text
Package
MK64FN1M0VLL12 - LQFP 100 package
ADC0 ADC1 CAN0 CMP0 CMP1 CMP2 CMT DAC0 DMA ENET EWM EzPort FB FTM0 FTM1 FTM2 FTM3 GPIOA GPIOB GPIOC GPIOD GPIOE I2C0 I2C1 I2C2 I2S0 JTAG LLWU LPTMR0 OSC PDB0 RCM RTC SDHC SIM SPI0 SPI1 SPI2 SUPPLY SystemControl TPIU UART0 UART1 UART2 UART3 UART4 USB0 USBDCD VREF
Routed Pins
Overview
Code Preview
Registers
Problems
MK64FN1M0xxx12-LED_Buttons
NXP MK64FN1M0xxx12 (MK64FN...ttons)

Pin	Pin name	Label	Identifier	GPIO	UART	FTM	ADC
☑ 1	ADC1_SE4a/PTE...	J15[P8]/SDHC0_D1	SDHC0_D1	PTE0	UART1_TX		ADC1_SE
☑ 2	ADC1_SE5a/PTE...	J15[P7]/SDHC0_D0	SDHC0_D0	PTE1	UART1_RX		ADC1_SE
☑ 3	ADC0_DP2/ADC...	J15[P5]/SDHC0_...	SDHC0_DCLK	PTE2	UART1_CTS_b		ADC0_DI
☑ 4	ADC0_DM2/AD...	J15[P3]/SDHC0_	SDHC0_CMD	PTE3	UART1_RTS_b		ADC0_DI
☑ 5	PTE4/LLWU_P2/...	J15[P2]/SDHC0_D3	SDHC0_D3	PTE4	UART3_TX		
☑ 6	PTE5/SPI1_PCS2...	J15[P1]/SDHC0_D2	SDHC0_D2	PTE5	UART3_RX	FTM3_CH0	
☑ 7	PTE6/SPI1_PCS3...	J15[G1]/SD_CAR...	SDHC_CD	PTE6	UART3_CTS_b	FTM3_CH1	
☑ 8	VDD16	P3V3_K64F					
☑ 9	VSS17	GND					
☑ 10	USB0_DP	J22[3]/K64_MICR...	USB_DP				
☑ 11	USB0_DM	J22[2]/K64_MICR...	USB_DM				
☑ 12	VOUT33	VOUT33_K64					
☑ 13	VREGIN	VREGIN_K64					
☑ 14	ADC0_DP1	J4[1]					ADC0_DI
☑ 15	ADC0_DM1	J4[3]					ADC0_DI
☑ 16	ADC1_DP1	J4[5]					ADC1_DI
☑ 17	ADC1_DM1	J4[7]					ADC1_DI
☑ 18	ADC0_DP0/ADC...	J2[5]					ADC0_DI
☑ 19	ADC0_DM0/AD...	J2[7]					ADC0_DI
☑ 20	ADC1_DP0/ADC...	J2[11]					ADC1_DI
☑ 21	ADC1_DM0/AD...	J2[13]					ADC1_DI
☑ 22	VDDA	P3V3_K64F					
☑ 23	VREFH	VREFH					VREFH[...
☑ 24	VREFL	GND					VREFL[...
☑ 25	VSSA	GND					VSSA[...]
☑ 26	VREF_OUT/CMP...	J2[17]					VREF_OU
☑ 27	DAC0_OUT/CM...	J4[11]	DAC0_OUT				ADC0_SE
☑ 28	XTAL32	Y3[1]/XTAL32_RTC	XTAL32K				
☑ 29	EXTAL32	Y3[2]/EXTAL32_R	ETAL32K				
☑ 30	VBAT	VBAT					
☑ 31	ADC0_SE17/PTE...	J2[20]/U8[4]/I2C0...	ACCEL_SCL	PTE24	UART4_TX		ADC0_SE

Routed Pins for BOARD_InitPins 0

#	Peripheral	Signal	Route to	Label	Identifier	Direction	GPIO initial...	GPIO interr	Slew rate	Open drain	Drive stren	Pull select	Pull enable	Passive filter	Digital filter

Configuration - General Info

Configuration - HW Info

Processor: MK64FN1M0xxx12
Part number: MK64FN1M0VLL12
Core: Cortex-M4F
Board: FRDM-K64F
SDK Version: ksdk2_0

Project

Pins

Tool for pin routing configuration, including pin functional/electrical properties, power rails, and run-time configurations.

Generated code

☑ Update project code enabled

board\pin_mux.c

board\pin_mux.h

Functional groups

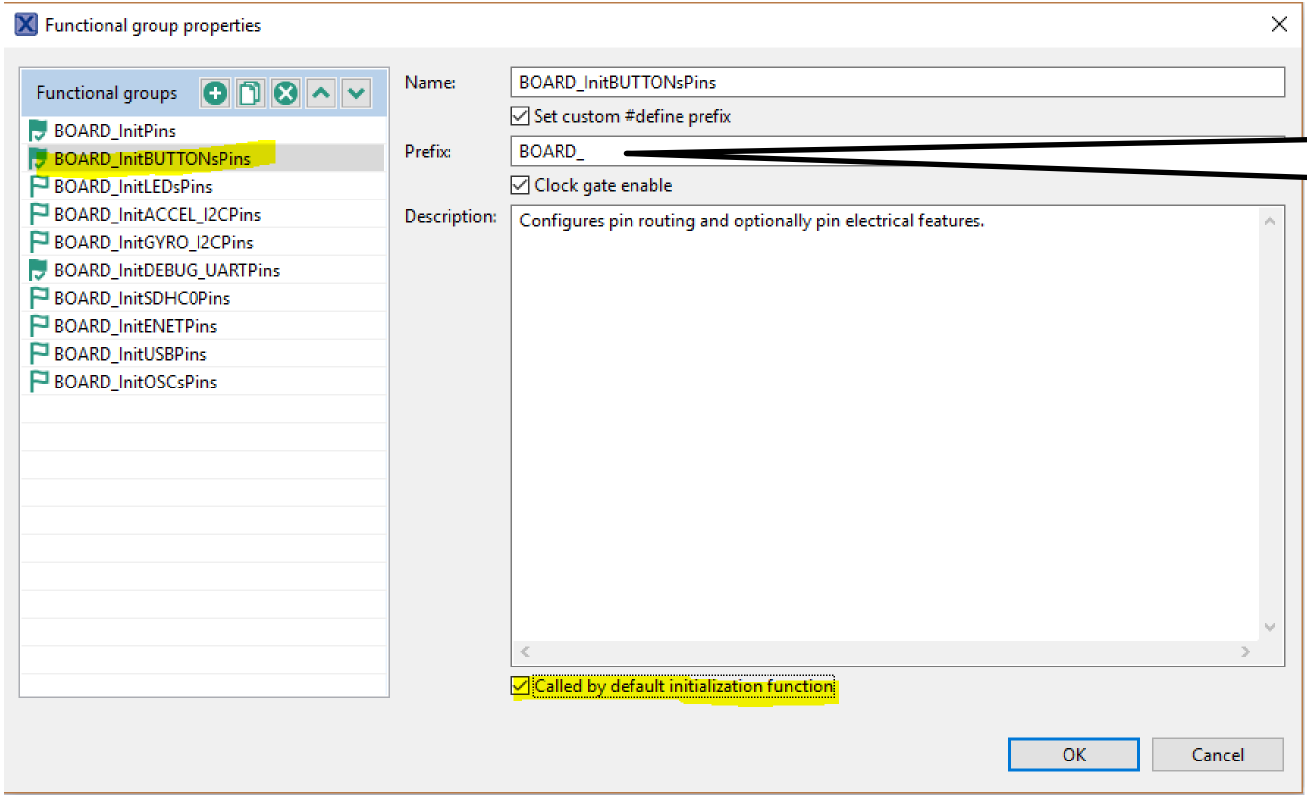
- BOARD_InitPins
- BOARD_InitButtonsPins
- BOARD_InitLEDsPins
- BOARD_InitDEBUG_UARTPins
- BOARD_InitOSCPins
- BOARD_InitACCELPins
- BOARD_InitENETPins
- BOARD_InitSDHCPins
- BOARD_InitUSBPins

Other tools

Level	Issue	Origin
Warning	Peripheral GPIOC is not initialized	Pins:BOARD_InitButtonsPins
Warning	Peripheral GPIOA is not initialized	Pins:BOARD_InitButtonsPins
Warning	Peripheral UART0 is not initialized	Pins:BOARD_InitDEBUG_UART
Warning	Peripheral RTC is not initialized	Pins:BOARD_InitOSCPins
Warning	Peripheral I2C0 is not initialized	Pins:BOARD_InitACCELPins
Warning	Peripheral USB0 is not initialized	Pins:BOARD_InitUSBPins

Application – Button using Config Tools

- Dialog with predefined functional group appears
- Set checkbox at the bottom of the dialog for functional group: “BOARD_InitButtonsPins” to call initialization code for the group in default initialization function



Functional group properties

Functional groups

- BOARD_InitPins
- BOARD_InitBUTTONsPins
- BOARD_InitLEDsPins
- BOARD_InitACCEL_I2CPins
- BOARD_InitGYRO_I2CPins
- BOARD_InitDEBUG_UARTPins
- BOARD_InitSDHC0Pins
- BOARD_InitENETPins
- BOARD_InitUSBPins
- BOARD_InitOSCsPins

Name: BOARD_InitBUTTONsPins

☑ Set custom #define prefix

Prefix: BOARD_

☑ Clock gate enable

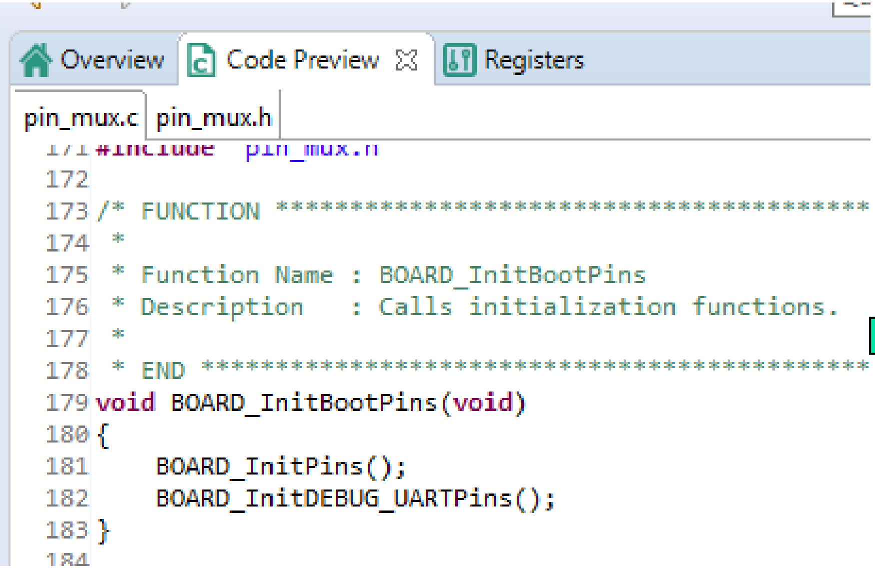
Description: Configures pin routing and optionally pin electrical features.

☑ Called by default initialization function

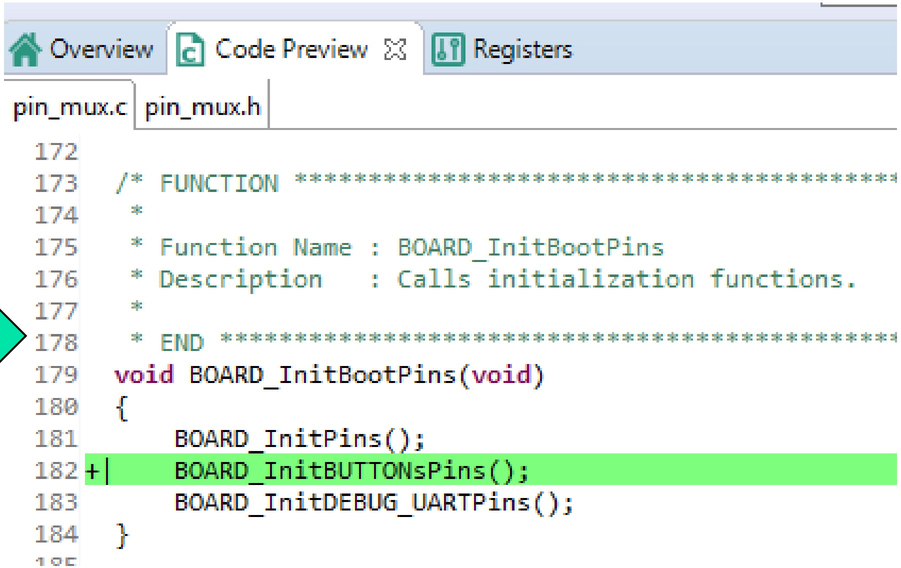
Prefix used in generated code

Application – Button using Config Tools

- Code preview is updated
- Initialization function now calls 1 more function



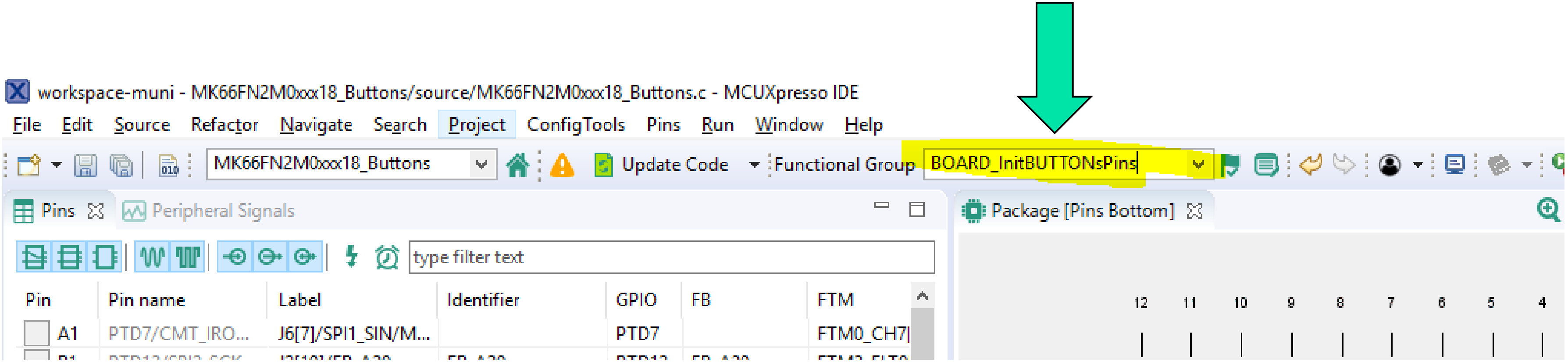
```
#include "pin_mux.h"

/* FUNCTION ****************************************
 *
 * Function Name : BOARD_InitBootPins
 * Description   : Calls initialization functions.
 *
 * END *********************************************
void BOARD_InitBootPins(void)
{
    BOARD_InitPins();
    BOARD_InitDEBUG_UARTPins();
}
```



```

/* FUNCTION ****************************************
 *
 * Function Name : BOARD_InitBootPins
 * Description   : Calls initialization functions.
 *
 * END *********************************************
void BOARD_InitBootPins(void)
{
    BOARD_InitPins();
    BOARD_InitBUTTONsPins();
    BOARD_InitDEBUG_UARTPins();
}
```

Application – Button using Config Tools

- Select “BOARD_InitBUTTONsPins” functional group from a combo-box



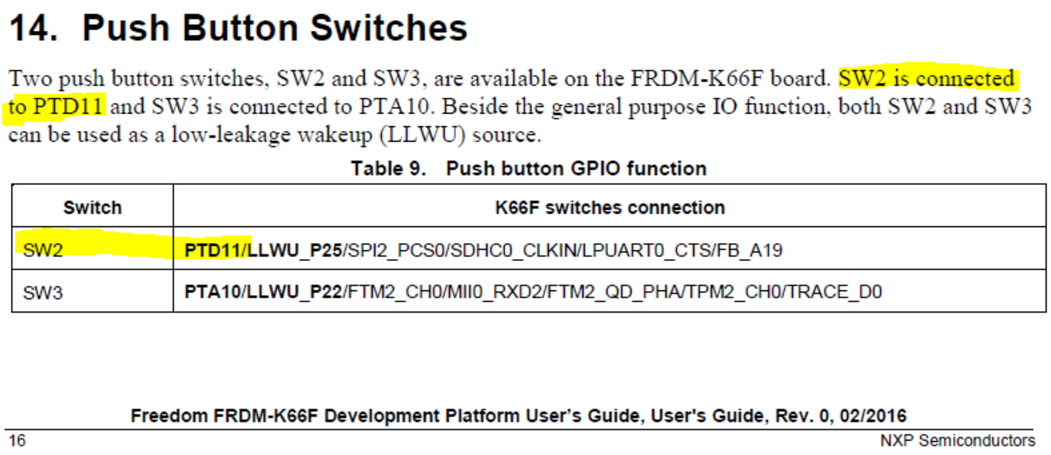
workspace-muni - MK66FN2M0xxx18_Buttons/source/MK66FN2M0xxx18_Buttons.c - MCUXpresso IDE

MK66FN2M0xxx18_Buttons

Functional Group BOARD_InitBUTTONsPins

type filter text

Pin	Pin name	Label	Identifier	GPIO	FB	FTM
□ A1	PTD7/CMT_IRO...	J6[7]/SPI1_SIN/M...		PTD7		FTM0_CH7|



14. Push Button Switches

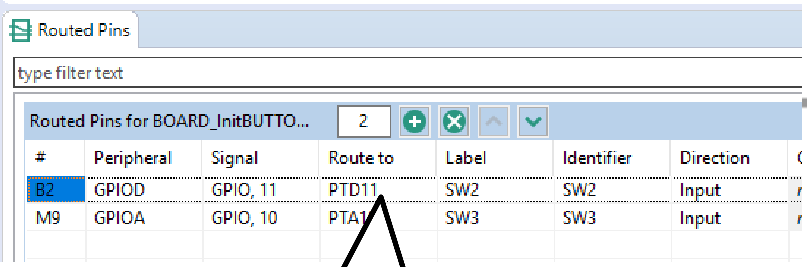
Two push button switches, SW2 and SW3, are available on the FRDM-K66F board. SW2 is connected to PTD11 and SW3 is connected to PTA10. Beside the general purpose IO function, both SW2 and SW3 can be used as a low-leakage wakeup (LLWU) source.

Table 9. Push button GPIO function

Switch	K66F switches connection
SW2	**PTD11/LLWU_P25**/SPI2_PCS0/SDHC0_CLKIN/LPUART0_CTS/FB_A19
SW3	**PTA10/LLWU_P22**/FTM2_CH0/MII0_RXD2/FTM2_QD_PHA/TPM2_CH0/TRACE_D0

Freedom FRDM-K66F Development Platform User's Guide, User's Guide, Rev. 0, 02/2016

16 NXP Semiconductors



#	Peripheral	Signal	Route to	Label	Identifier	Direction
B2	GPIOD	GPIO, 11	PTD11	SW2	SW2	Input
M9	GPIOA	GPIO, 10	PTA1	SW3	SW3	Input

GPIO signal is routed to "PTD11"

Audio

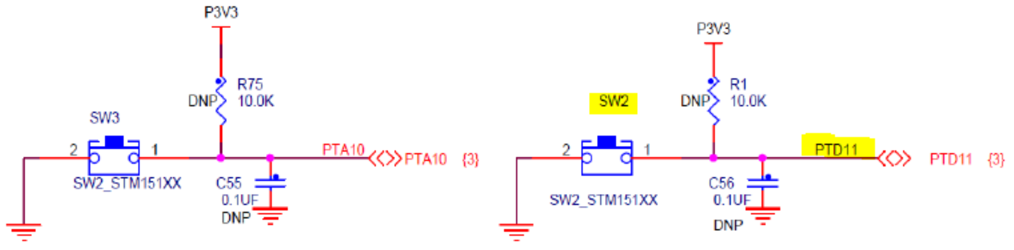
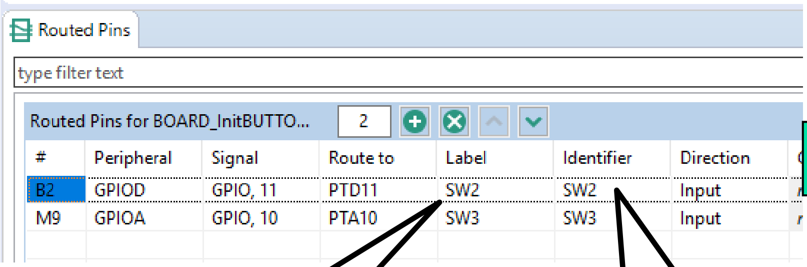


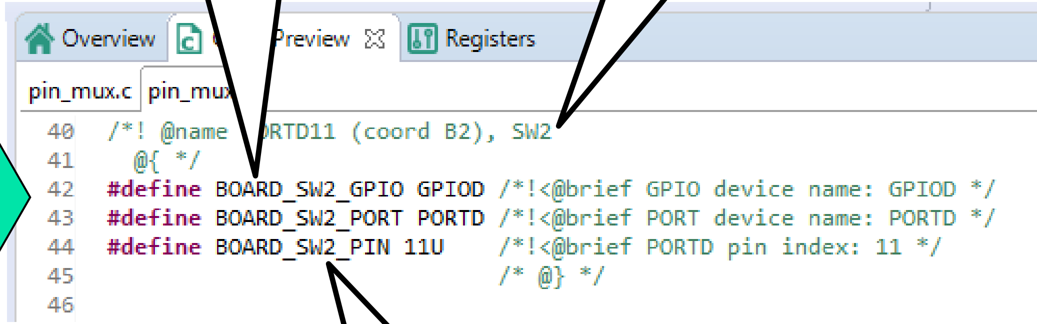
Figure 17. Push button switches



type filter text

Routed Pins for BOARD_InitBUTTO... 2

#	Peripheral	Signal	Route to	Label	Identifier	Direction
B2	GPIOD	GPIO, 11	PTD11	SW2	SW2	Input
M9	GPIOA	GPIO, 10	PTA10	SW3	SW3	Input



```
/*! @name  ORTD11 (coord B2), SW2
  @{ */
#define BOARD_SW2_GPIO GPIOD /*!<@brief GPIO device name: GPIOD */
#define BOARD_SW2_PORT PORTD /*!<@brief PORT device name: PORTD */
#define BOARD_SW2_PIN 11U    /*!<@brief PORTD pin index: 11 */
                             /* @} */
```

Prefix used in generated code

User friendly Label – specific for our board

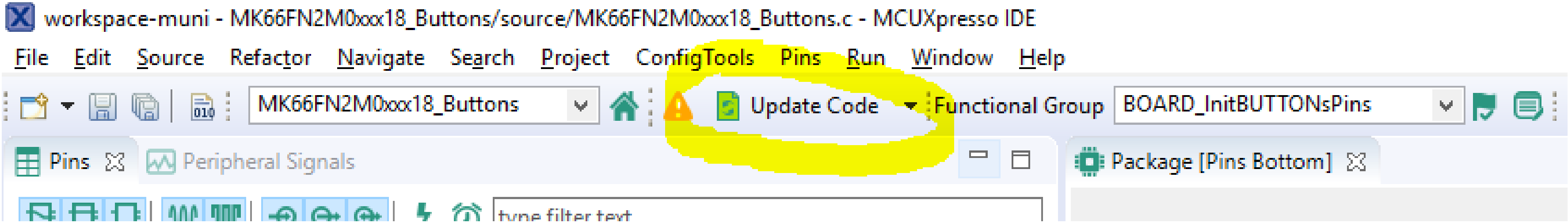
User-friendly Label – specific for our board

Identifier used in generated code

Identifier used in generated code

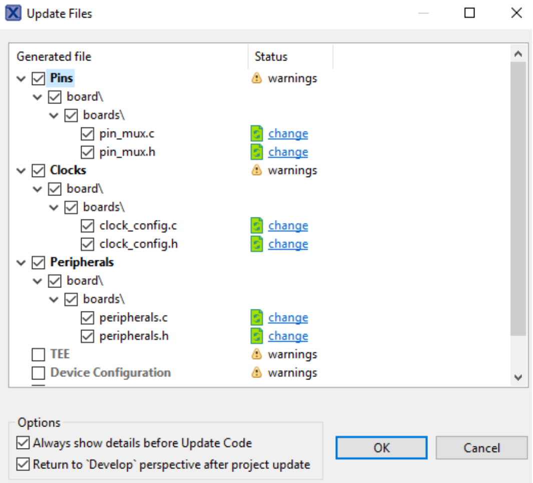
Application – Button using Config Tools

- Your project needs to be updated by newly generated code now – this action is done automatically
- Whenever generated code does not match the code is your project, it is displayed by changed color of the icon in the toolbar



Application – Button using Config Tools

- Update Project Files dialog should appear
- You can view changes to each file by clicking on the "change" text in the "Status" column
- Press "OK" to update your project



Update Files

Generated file	Status
☑ Pins	warnings
☑ board\	
☑ boards\	
☑ pin_mux.c	change
☑ pin_mux.h	change
☑ Clocks	warnings
☑ board\	
☑ boards\	
☑ clock_config.c	change
☑ clock_config.h	change
☑ Peripherals	
☑ board\	
☑ boards\	
☑ peripherals.c	change
☑ peripherals.h	change
☐ TEE	warnings
☐ Device Configuration	warnings

Options

☑ Always show details before Update Code

☑ Return to `Develop` perspective after project update

Application – Button using Config Tools

- Clocks tool – preconfigured, you can view clock settings
- Peripherals tool – no need for now

Application – Button using Config Tools

- Write application code
  - Read current state of the GPIO(button)
  - Detect button press

      Print text into console

Application – Button using Config Tools

- When you press the button, text is printed more than once

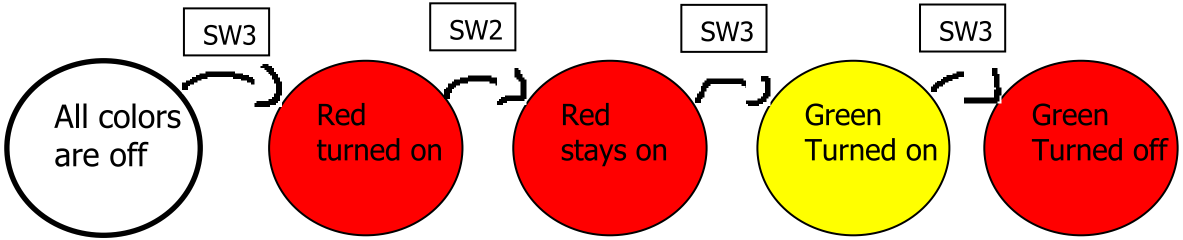
- Why?
- How to resolve this issue?

Work Progress

- Write an applications that toggles **GREEN** LED when **SW3** button is pressed
- Fix the issue with button press being registered more than once
- Make LED change color every time it is turned on (there are 3 LEDs on the board: Red, Green, Blue)

Homework

- Write an applications that reacts to both buttons
- SW3 turns on/off selected color
- SW2 changes selected color (R -> G -> B -> R …)
- At start all colors are turned off, and RED is selected



Homework - git

- Git Branch- “Week_02”
- Git tag – “Submission_02_x”