

PV198 - GPIO

One-chip Controllers

Daniel Dlhopolček, Marek Vrbka, Jan Koniarik, Oldřich Pecák, Tomáš Rohlínek, Ján Labuda, Jan Horáček, Matúš Škvarla

Faculty of Informatics, Masaryk University

2/2023

Content

GPIO Overview

Buttons

 Debouncing

Application

 LED using SDK example

 Button using config tools

Homework

- Have you checked the preliminaries in study materials?
- Do not forget to setup a new branch for this week (*Week_02*)!

What is GPIO

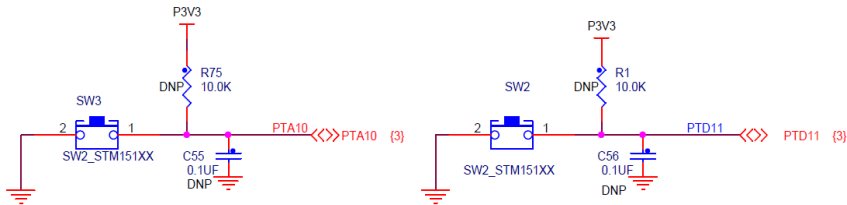
- **GPIO** – **G**eneral **P**urpose **I**nput **O**utput
- Direct control of pins of the MCU
- Basic interaction with external world
- Can be programmed as *input* or *output*
- Has only 2 states (logic 0, logic 1)

What is it used for

- Anything that works with 2 states – on/off
- LED
- Buttons
- Sensors
- And used by more sophisticated peripherals

How buttons on board work

Connects pin to ground (logic 0) or to voltage (logic 1)



Button debouncing

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

Steps required to create an application

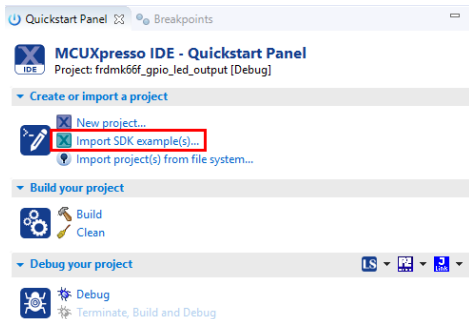
- Initialize (*MCUXpresso Configuration Tools* help here)
 1. Pin
 2. Clocks
 3. Peripherals
- Write application code

Peripheral configuration options

1. Write everything from scratch
 - Error prone, time demanding, tedious
2. Use SDK example
 - Works out of box
 - Difficult to modify
3. Use config tools
 - Easy to use and modify

LED using the SDK example

- Select Import SDK example(s)...



1. Open the K6x
2. Select the MK66FN2M0xxx18
3. Click 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

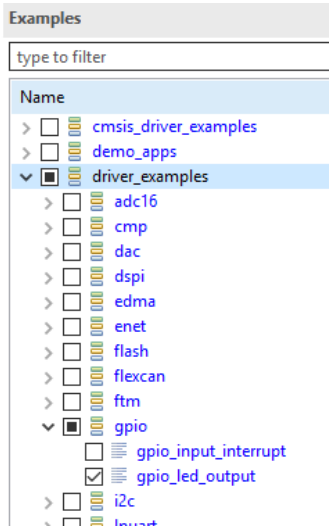
Please select an available board for your project.

Supported boards for device: MK66FN2M0xxx18



[frdmk66f](#) SDK

1. Open `driver_examples` → `gpio`
2. Select the `gpio_led_output` example
3. Click *Finish*



Opened example project

- Pins and clocks are already configured
- `GPIO_PinInit`
- `GPIO_PortToggle`

Button control program

- We will show you how to check for button presses
- The end goal is to write program which will print text to console when SW2 is pressed

You should see the Pin tool now

The screenshot shows the MikroC IDE interface with the Pin tool configuration window open. The Pin tool window displays a list of pins and their configurations, including pin name, label, identifier, direction, and other properties. The 'Routed Pins' window is also open, showing the routing status for various pins. The 'Configuration - HW Info' panel on the right shows the device name, processor, and other hardware details.

Pin Tool Configuration:

| Pin | Pin name | Label | Identifier | GPIO | UART | FTM | ADC |
|-----|----------------|-------------------|------------|-----------|-------------|----------|----------|
| 1 | AD0C1_S04V_PTE | J19P1/S04C0_01 | SDH0C_D1 | PT08 | UART1_TX | | AD0C1_S0 |
| 2 | AD0C1_S04V_PTE | J19P17/S04C0_02 | SDH0C_D0 | PT01 | UART1_RX | | AD0C1_S0 |
| 3 | AD0C1_S04V_PTE | J19P15/S04C0_... | SDH0C_D0K | PT02 | UART1_CTS_b | | AD0C1_S0 |
| 4 | AD0C1_S04V_PTE | J19P13/S04C0_... | SDH0C_CMD | PT03 | UART1_PFS_b | | AD0C1_S0 |
| 5 | FTM4_U0H0_PTE | J19P2/S04C0_03 | SDH0C_D3 | PT04 | UART1_TX | | |
| 6 | FTM4_U0H0_PTE | J19P19/S04C0_02 | SDH0C_D2 | PT05 | UART1_RX | | |
| 7 | FTM4_U0H0_PTE | J19P10/S04_CAR... | SDH0C_CD | PT06 | UART1_CTS_b | FTM4_CH0 | |
| 8 | VDD18 | FT03_KM6F | | | | FTM4_CH1 | |
| 9 | VSS17 | GN0 | | | | | |
| 10 | USBD_DP | J20P1/K54_M0C... | USB_DP | | | | |
| 11 | USBD_DM | J20P2/K54_M0C... | USB_DM | | | | |
| 12 | VOUT33_K54 | VOUT33_K54 | | | | | |
| 13 | VREG0N | VREG0N_K54 | | | | | |
| 14 | AD0C1_S01 | J4T1 | | AD0C1_D1 | | | |
| 15 | AD0C1_S01 | J4T2 | | AD0C1_D1 | | | |
| 16 | AD0C1_S01 | J4T3 | | AD0C1_D1 | | | |
| 17 | AD0C1_S01 | J4T7 | | AD0C1_D1 | | | |
| 18 | AD0C1_S01V_PTE | J2T1 | | AD0C1_D1 | | | |
| 19 | AD0C1_S01V_PTE | J2T7 | | AD0C1_D1 | | | |
| 20 | AD0C1_S01V_PTE | J2T11 | | AD0C1_D1 | | | |
| 21 | AD0C1_S01V_PTE | J2T13 | | AD0C1_D1 | | | |
| 22 | VDDA | FT03_KM6F | | AD0C1_D1 | | | |
| 23 | VREFH | VREFH | | | | | |
| 24 | VREFL | GN0 | | | | | |
| 25 | VSSA | GN0 | | | | | |
| 26 | VREF_OUT1_COMP | J2T7 | | VREF_OUT1 | | | |
| 27 | DACL_OUT1_COMP | J4T11 | DACL_OUT1 | | | | |
| 28 | XTAL32 | X3P1/XTAL32_RTC | XTAL32K | | | | |
| 29 | EXTAL32 | X3P2/XTAL32_R | EXTAL32K | | | | |
| 30 | VBAT | | | | | | |
| 31 | AD0C1_S017_PTE | J20P13/UR14/QC... | ACC0C_SCL | PT04 | UART4_TX | | AD0C1_S0 |

Routed Pins for BOARD_v4Pins:

| # | Peripheral | Signal | Route to | Label | Identifier | Direction | GPIO instal.. | GPIO inter | Stw rate | Open drain | Drive stre | Pull select | Pull enable | Passive filter | Digital filter |
|----|------------|--------|----------|-------|------------|-----------|---------------|------------|----------|------------|------------|-------------|-------------|----------------|----------------|
| 1 | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | | | | |

Configuration - HW Info:

- Configuration - HW Info
- Processor: MK64FN1M0VLL12
- Part number: MK64FN1M0VLL12
- Core: Cortex-M4
- Board: FTD4-484F
- SZE Version: kmd2_0

Generated code:

- Update project code enabled
- board_v4Pins.c
- board_v4Pins.h

Functional groups:

- BOARD_v4Pins
- BOARD_v4PinsPins
- BOARD_v4PinsLEDsPins
- BOARD_v4PinsUSBDPins
- BOARD_v4PinsUSBPins
- BOARD_v4PinsUSBCPins
- BOARD_v4PinsUSBRPins

Other tasks:

- Problems
- Issues

Problems:

| Level | Issue | Origin |
|---------|-------------------------------------|--------------------------|
| Warning | Peripheral GPIOC is not initialized | FuncBOARD_v4PinsPins |
| Warning | Peripheral GPIOA is not initialized | FuncBOARD_v4PinsPins |
| Warning | Peripheral UART0 is not initialized | FuncBOARD_v4PinsUSBDPins |
| Warning | Peripheral RTC is not initialized | FuncBOARD_v4PinsUSBCPins |
| Warning | Peripheral I2C is not initialized | FuncBOARD_v4PinsUSBRPins |
| Warning | Peripheral USB0 is not initialized | FuncBOARD_v4PinsUSBRPins |

Initialization

How configuration tools can help us:

- Modify settings easily
- Visual representation of configuration
- Great for custom boards (generates defines for custom boards that simplify management)

Configuration

- Pins tool contains predefined configurations
- We should already see the red LED configured
- Add the configuration for SW2 and SW3 buttons
 - Search for SW2 and SW3 on the “pins” window
 - Click on the checkboxes for SW2 and SW3 and add the GPIO option
 - This will call initialization code for the button pins on program startup

pv198 - frdmk66f_gpio_led_output/source/gpio_led_output.c - MCUXpresso IDE

File Edit Source Refactor Navigate Search Project ConfigTools Pins Run Window Help

frdmk66f_gpio_led_output Update Code Functional Group BOARD_InitPins

Pins × Peripheral Signals

SW

| Pin | Pin name | Label | Identifier | FRDM Expansion... | GPIO | FB | FTM |
|-----------------------------|-------------------|----------------------|------------|-------------------|---------------|---------|---------|
| <input type="checkbox"/> B2 | PTD11/LLWU_P2... | SW2 | SW2 | | GPIOD:GPIO,11 | FB:A,19 | |
| <input type="checkbox"/> J5 | TSIO_CH1/PTA0/... | J8[2]/J9[4]/JTAG... | SWD_CLK | | GPIOA:GPIO,0 | | FTM0:Cl |
| <input type="checkbox"/> K6 | TSIO_CH3/PTA2/... | J9[6]/JTAG_TDO/... | TRACE_SWO | | GPIOA:GPIO,2 | | FTM0:Cl |
| <input type="checkbox"/> K7 | TSIO_CH4/PTA3/... | J9[2]/J12[2]/JTAG... | SWD_DIO | | GPIOA:GPIO,3 | | FTM0:Cl |
| <input type="checkbox"/> M9 | PTA10/LLWU_P2... | SW3 | SW3 | | GPIOA:GPIO,10 | | FTM2:Cl |

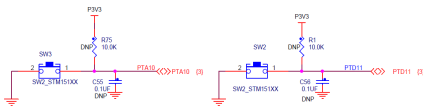
- *Code preview* was updated
- If you check the *Code Preview* tab, you should see that the `pin_mux.c` file now has extra SW2 and SW3 configuration
- You should see in the *Routed Pins* tab (lower-left corner) that button pins are routed to PTD11 (SW2) and PTA10 (SW3)

Routed Pins

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 |



Updating code

- Click the *Update Code* button
 - It opens the *Update Files* dialog
 - You can check which changes will be made
 - For now, just click OK

Writing actual code

Task - Reading a button and printing to console

- Read the current state of the GPIO Button (SW2 and/or SW3)
- If button is pressed, print text to console
- Otherwise, do nothing

Issues

- When you press the button, text is printed several times
 - Why?
 - What are the ways to resolve it?

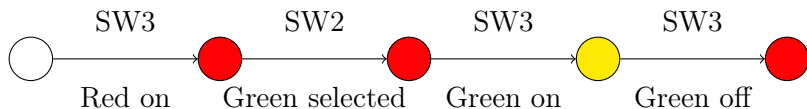
Work Progress

- Write an application that toggles green LED when SW3 is pressed
- Fix the issue with the button press being registered more than once
- Make the LED change color every time it is turned on
 - There are three controllable LEDs on the board

Homework

Write an application which reacts to both buttons

- SW2 selects color
- SW3 toggles the color on and off
- All colors start turned off
- Selected color starts on red
- Colors switch in the following order: Red \rightarrow Green \rightarrow Blue \rightarrow Red...
- Application must be immune to the effects of bouncing



Submission

- Git tag - “Submission_02_x”
- One project per branch!

MUNI

FACULTY

OF INFORMATICS