

MUNI
FI



PV198 - GPIO

One-chip Controllers

Daniel Dlhopolček, Marek Vrbka

Faculty of Informatics, Masaryk University

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!

What is GPIO

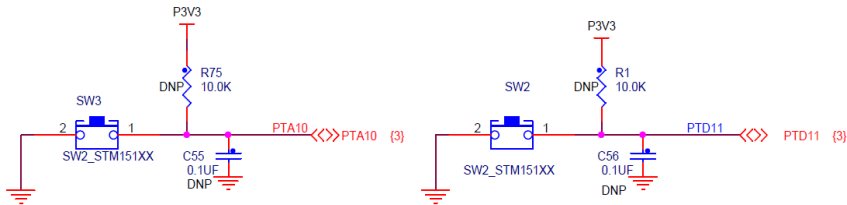
- **GPIO** – **G**eneral **P**urpose **I**nput **O**utput
- Direct control of pins of the MCU
- Basic interaction with external world
- Can be programmed as Input or Output
- Has only 2 states (logic 0, logic 1)

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
 - HW debounce(add capacitor)
 - SW debounce(wait few miliseconds)

Steps required to create an application

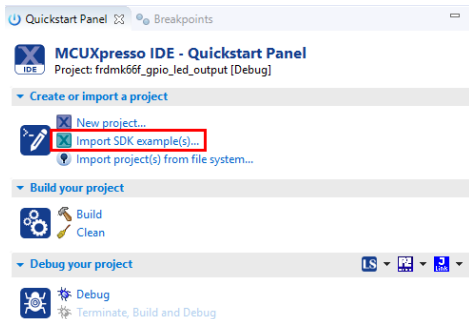
- Initialize(MCUXpresso Configuration Tools help here)
 - Pin
 - Clocks
 - Peripherals
- Write application code

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)...



- Open the K6x
- Select the MK66FN2M0xxx18
- Click the board image

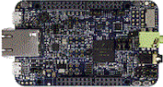
X Board and/or Device selection page

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.

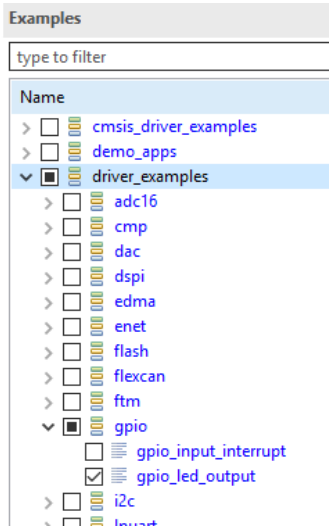
Supported boards for device: MK66FN2M0xxx18



SDK

[frdmk66f](#)

- Open driver_examples → gpio
- Select the “gpio_led_output” example
- Click Finish



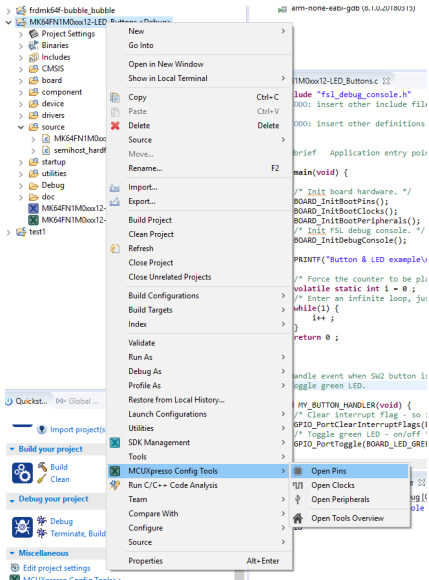
Opened example project

- Pins, 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

Application Button using config tools



The screenshot shows an IDE interface with a project tree on the left and a code editor on the right. The project tree is expanded to show the 'source' folder. A context menu is open over the 'source' folder, with 'MCUxpresso Config Tools' selected. A sub-menu is also open, showing options like 'Open Pins', 'Open Clocks', 'Open Peripherals', and 'Open Tools Overview'.

```
arm-none-eabi-gdb (6.11.0.20180513)
MK64FN1M0xx12-LED
  Project Settings
  Binaries
  Includes
  CMSIS
  board
  component
  device
  drivers
  source
    MK64FN1M0xx12-LED
    semihost_hardf
  startup
  utilities
  Debug
  doc
  MK64FN1M0xx12-LED
  MK64FN1M0xx12-LED
  test1

MK64FN1M0xx12-LED
  Project Settings
  Binaries
  Includes
  CMSIS
  board
  component
  device
  drivers
  source
    MK64FN1M0xx12-LED
    semihost_hardf
  startup
  utilities
  Debug
  doc
  MK64FN1M0xx12-LED
  MK64FN1M0xx12-LED
  test1

MK64FN1M0xx12-LED_Buttons.c
#include "fsl_debug_console.h"
#define INSERT_OTHER_INCLUDE_FILES_HERE
#define INSERT_OTHER_DEFINITIONS_HERE

brief Application entry point
main(void) {
    /* Init board hardware. */
    BOARD_InitBootPins();
    BOARD_InitBootClocks();
    BOARD_InitBootPeripherals();
    /* Init FSL debug console. */
    BOARD_InitDebugConsole();

    PRINTF("Button & LED example\n");

    /* Force the counter to be global. */
    volatile static int i = 0;
    /* Enter an infinite loop, just to keep the processor busy. */
    while(1) {
        i++;
    }
    return 0;
}

handle event when SW2 button is pressed
toggle green LED.

MY_BUTTON_HANDLER(void) {
    /* Clear interrupt flag - so that we can receive another interrupt. */
    GPIO_PortClearInterruptFlags(GPIO_PORTC_DR0_REG, GPIO_PIN_1);
    /* Toggle green LED - on/off */
    GPIO_PortToggle(BOARD_LED_GREEN_GPIO_PORT, BOARD_LED_GREEN_GPIO_PIN);
}

MCUxpresso Config Tools
  Open Pins
  Open Clocks
  Open Peripherals
  Open Tools Overview
```

You should see the Pin tool now

The screenshot shows the MikroC IDE interface for the MK64FN1M0VLL12 microcontroller. The Pin tool window is open, displaying a list of pins and their configurations. The 'Routed Pins for BOARD_v4Pins' table is visible at the bottom. The right sidebar shows the project configuration and a list of generated code files.

#	Peripheral	Signal	Route to	Label	Identifier	Direction	GPI0 instal..	GPI0 inter	Stow rate	Open drain	Drive stren	Pull select	Pull enable	Passive filter	Digital filter
1	AD_CONVERTER	AD_CONVERTER													
2	AD_CONVERTER	AD_CONVERTER													
3	AD_CONVERTER	AD_CONVERTER													
4	AD_CONVERTER	AD_CONVERTER													
5	AD_CONVERTER	AD_CONVERTER													
6	AD_CONVERTER	AD_CONVERTER													
7	AD_CONVERTER	AD_CONVERTER													
8	AD_CONVERTER	AD_CONVERTER													
9	AD_CONVERTER	AD_CONVERTER													
10	AD_CONVERTER	AD_CONVERTER													
11	AD_CONVERTER	AD_CONVERTER													
12	AD_CONVERTER	AD_CONVERTER													
13	AD_CONVERTER	AD_CONVERTER													
14	AD_CONVERTER	AD_CONVERTER													
15	AD_CONVERTER	AD_CONVERTER													
16	AD_CONVERTER	AD_CONVERTER													
17	AD_CONVERTER	AD_CONVERTER													
18	AD_CONVERTER	AD_CONVERTER													
19	AD_CONVERTER	AD_CONVERTER													
20	AD_CONVERTER	AD_CONVERTER													
21	AD_CONVERTER	AD_CONVERTER													
22	AD_CONVERTER	AD_CONVERTER													
23	AD_CONVERTER	AD_CONVERTER													
24	AD_CONVERTER	AD_CONVERTER													
25	AD_CONVERTER	AD_CONVERTER													
26	AD_CONVERTER	AD_CONVERTER													
27	AD_CONVERTER	AD_CONVERTER													
28	AD_CONVERTER	AD_CONVERTER													
29	AD_CONVERTER	AD_CONVERTER													
30	AD_CONVERTER	AD_CONVERTER													
31	AD_CONVERTER	AD_CONVERTER													
32	AD_CONVERTER	AD_CONVERTER													
33	AD_CONVERTER	AD_CONVERTER													
34	AD_CONVERTER	AD_CONVERTER													
35	AD_CONVERTER	AD_CONVERTER													
36	AD_CONVERTER	AD_CONVERTER													
37	AD_CONVERTER	AD_CONVERTER													
38	AD_CONVERTER	AD_CONVERTER													
39	AD_CONVERTER	AD_CONVERTER													
40	AD_CONVERTER	AD_CONVERTER													
41	AD_CONVERTER	AD_CONVERTER													
42	AD_CONVERTER	AD_CONVERTER													
43	AD_CONVERTER	AD_CONVERTER													
44	AD_CONVERTER	AD_CONVERTER													
45	AD_CONVERTER	AD_CONVERTER													
46	AD_CONVERTER	AD_CONVERTER													
47	AD_CONVERTER	AD_CONVERTER													
48	AD_CONVERTER	AD_CONVERTER													
49	AD_CONVERTER	AD_CONVERTER													
50	AD_CONVERTER	AD_CONVERTER													
51	AD_CONVERTER	AD_CONVERTER													
52	AD_CONVERTER	AD_CONVERTER													
53	AD_CONVERTER	AD_CONVERTER													
54	AD_CONVERTER	AD_CONVERTER													
55	AD_CONVERTER	AD_CONVERTER													
56	AD_CONVERTER	AD_CONVERTER													
57	AD_CONVERTER	AD_CONVERTER													
58	AD_CONVERTER	AD_CONVERTER													
59	AD_CONVERTER	AD_CONVERTER													
60	AD_CONVERTER	AD_CONVERTER													
61	AD_CONVERTER	AD_CONVERTER													
62	AD_CONVERTER	AD_CONVERTER													
63	AD_CONVERTER	AD_CONVERTER													
64	AD_CONVERTER	AD_CONVERTER													
65	AD_CONVERTER	AD_CONVERTER													
66	AD_CONVERTER	AD_CONVERTER													
67	AD_CONVERTER	AD_CONVERTER													
68	AD_CONVERTER	AD_CONVERTER													
69	AD_CONVERTER	AD_CONVERTER													
70	AD_CONVERTER	AD_CONVERTER													
71	AD_CONVERTER	AD_CONVERTER													
72	AD_CONVERTER	AD_CONVERTER													
73	AD_CONVERTER	AD_CONVERTER													
74	AD_CONVERTER	AD_CONVERTER													
75	AD_CONVERTER	AD_CONVERTER													
76	AD_CONVERTER	AD_CONVERTER													
77	AD_CONVERTER	AD_CONVERTER													
78	AD_CONVERTER	AD_CONVERTER													
79	AD_CONVERTER	AD_CONVERTER													
80	AD_CONVERTER	AD_CONVERTER													
81	AD_CONVERTER	AD_CONVERTER													
82	AD_CONVERTER	AD_CONVERTER													
83	AD_CONVERTER	AD_CONVERTER													
84	AD_CONVERTER	AD_CONVERTER													
85	AD_CONVERTER	AD_CONVERTER													
86	AD_CONVERTER	AD_CONVERTER													
87	AD_CONVERTER	AD_CONVERTER													
88	AD_CONVERTER	AD_CONVERTER													
89	AD_CONVERTER	AD_CONVERTER													
90	AD_CONVERTER	AD_CONVERTER													
91	AD_CONVERTER	AD_CONVERTER													
92	AD_CONVERTER	AD_CONVERTER													
93	AD_CONVERTER	AD_CONVERTER													
94	AD_CONVERTER	AD_CONVERTER													
95	AD_CONVERTER	AD_CONVERTER													
96	AD_CONVERTER	AD_CONVERTER													
97	AD_CONVERTER	AD_CONVERTER													
98	AD_CONVERTER	AD_CONVERTER													
99	AD_CONVERTER	AD_CONVERTER													
100	AD_CONVERTER	AD_CONVERTER													

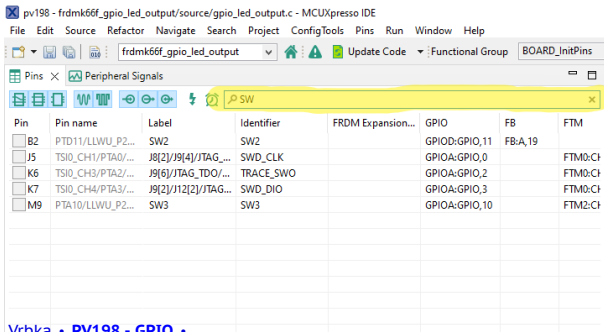
Initialization

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



The screenshot shows the MCUXpresso IDE interface. The title bar indicates the project is 'pv198 - frdmk66f_gpio_led_output/source/gpio_led_output.c'. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, ConfigTools, Pins, Run, Window, and Help. The toolbar shows various icons, including a search icon, and the text 'frdmk66f_gpio_led_output' is visible. Below the toolbar, the 'Pins' tool is active, displaying a table of pins. The table has columns for Pin, Pin name, Label, Identifier, FRDM Expansion..., GPIO, FB, and FTM. The rows for SW2 and SW3 are highlighted in yellow. SW2 is associated with GPIO:GPIO,11 and FB:A,19. SW3 is associated with GPIOA:GPIO,10 and FTM2:C1.

Pin	Pin name	Label	Identifier	FRDM Expansion...	GPIO	FB	FTM
<input type="checkbox"/>	B2	PTD11/LLWU_P2...	SW2		GPIO:GPIO,11	FB:A,19	
<input type="checkbox"/>	J5	TSIO_CH1/PTA0/...	J8[2]/J9[4]/JTAG...	SW2	GPIOA:GPIO,0		FTM0:C1
<input type="checkbox"/>	K6	TSIO_CH3/PTA2/...	J9[6]/JTAG_TDO/...	TRACE_SWO	GPIOA:GPIO,2		FTM0:C1
<input type="checkbox"/>	K7	TSIO_CH4/PTA3/...	J9[2]/J12[2]/JTAG...	SWD_DIO	GPIOA:GPIO,3		FTM0:C1
<input type="checkbox"/>	M9	PTA10/LLWU_P2...	SW3		GPIOA:GPIO,10		FTM2:C1

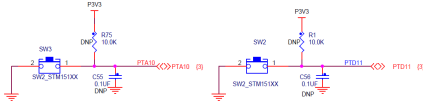
- Code preview was updated
- If you check the Code Preview tab, you should see that the `pin_mux.c` file now has extra SW2 and SW3 configuration
- You should see in the Routed Pins tab (lower-left corner) that button pins are routed to PTD11 (SW2) and PTA10 (SW3)

Routed Pins

type filter text

Routed Pins for BOARD_InitBUTTO... 2

#	Peripheral	Signal	Route to	Label	Identifier	Direction
B2	GPIOB	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?

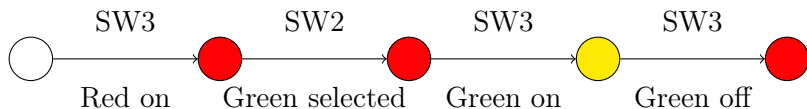
Work Progress

- 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 Branch - “Week_02” (Case sensitive!)
- Git tag - “Submission_02_x”
- One project per branch!

MUNI

FACULTY

OF INFORMATICS