# LAB: Digital In/Out

**Date:** 2022-09-20

**Author/Partner:**

**Github:** repository link

**Demo Video:** Youtube link

# Introduction

In this lab, you are required to create a simple program that toggle multiple LEDs with a push-button input. Create HAL drivers for GPIO digital in and out control and use your library.

You must submit

- LAB Report (*.md & *.pdf)
- Zip source files(main*.c, ecRCC.h, ecGPIO.h etc...).
    - Only the source files. Do not submit project files

## Requirement

### Hardware

- MCU
    - NUCLEO-F401RE
- Actuator/Sensor/Others:
    - LEDs x 3
    - Resistor 330 ohm x 3, breadboard

### Software

- Keil uVision, CMSIS, EC_HAL library

# Problem 1: Create EC_HAL library

## Procedure

Create the library directory `\repos\EC\EC_HAL\lib\`.

Save your header library files in this directory. See here for detail.

> DO NOT make duplicates of library files under each project folders

List of functions for Digital_In and Out .

Library file LINK such as github

**ecRCC.h** (provided)

```
void RCC_HSI_init(void);
void RCC_GPIOA_enable(void);
void RCC_GPIOB_enable(void);
void RCC_GPIOC_enable(void);
```

**ecGPIO.h**

```
void GPIO_init(GPIO_TypeDef *Port, int pin,  int mode);
void GPIO_write(GPIO_TypeDef *Port, int pin,  int output);
int  GPIO_read(GPIO_TypeDef *Port, int pin);
void GPIO_mode(GPIO_TypeDef* Port, int pin, int mode);
void GPIO_ospeed(GPIO_TypeDef* Port, int pin,  int speed);
void GPIO_otype(GPIO_TypeDef* Port, int pin,  int type);
void GPIO_pudr(GPIO_TypeDef* Port, int pin,  int pudr);
```

- Example code

```
/* ecGPIO.c  */

// Input(00), Output(01), AlterFunc(10), Analog(11)
void GPIO_mode(GPIO_TypeDef *Port, int pin, int mode){
   Port->MODER &= ~(3UL<<(2*pin));
   Port->MODER |= mode<<(2*pin);
}
```

# Problem 2: Toggle LED with Button

## Procedure

1) Create a new project under the directory `\repos\EC\LAB\`

- The project name is "**LAB_GPIO_DIO_LED".**
- Name the source file as "**LAB_GPIO_DIO_LED.c"**

2) Include your library **ecGPIO.h, ecGPIO.c** in `\repos\EC\EC_HAL\lib\`.

> You MUST write your name in the top of the source file, inside the comment section.

3) Toggle the LED by pushing button.

- Pushing  button (LED ON), Pushing Button (LED OFF) and repeat

## Configuration

| Button (B1) | LED |
|---|---|
| Digital In | Digital Out |
| GPIOC, Pin 13 | GPIOA, Pin 5 |
| PULL-UP | Open-Drain, Pull-up, Medium Speed |

## Code

Your code goes here:  ADD Code LINK such as github

Explain your source code with necessary comments.

#### Sample Code

```
#define LED_PIN     5
#define BUTTON_PIN 13

void setup(void);

int main(void) {
    // Initialiization
    setup();

    // Inifinite Loop
    while(1){
        if(GPIO_read(GPIOC, BUTTON_PIN) == 0)       GPIO_write(GPIOA, LED_PIN,
HIGH);
        else                                        GPIO_write(GPIOA, LED_PIN,
LOW);
    }
}


// Initialiization
void setup(void)
{
    RCC_HSI_init();
    GPIO_init(GPIOC, BUTTON_PIN, INPUT);  // calls RCC_GPIOC_enable()
    GPIO_init(GPIOA, LED_PIN, OUTPUT);    // calls RCC_GPIOA_enable()
}
```

## Discussion

1) What the differences between open-drain and Push-pull for output pin?

2) Find out a typical solution for software debouncing and hardware debouncing. What method of debouncing did this NUCLEO board used for the push-button(B1)?

3) Check the output pin with oscilloscope and observe how the signals change with input button

# Problem 3: Toggle LED with Button

## Procedure

1) Create a new project under the directory `\repos\EC\LAB\`

- The project name is "**LAB_GPIO_DIO_multiLED".**
- Name the source file as "**LAB_GPIO_DIO_multiLED.c"**

> You MUST write your name in the top of the source file, inside the comment section.

2) Include your library **ecGPIO.h, ecGPIO.c** in `\repos\EC\EC_HAL\lib\`.

3) Connect 4 LEDs externally with necessary load resistors.

- As Button B1 is Pressed, light one LED at a time, in sequence.
- Example: LED0--> LED1--> ...LED3--> ...LED0....

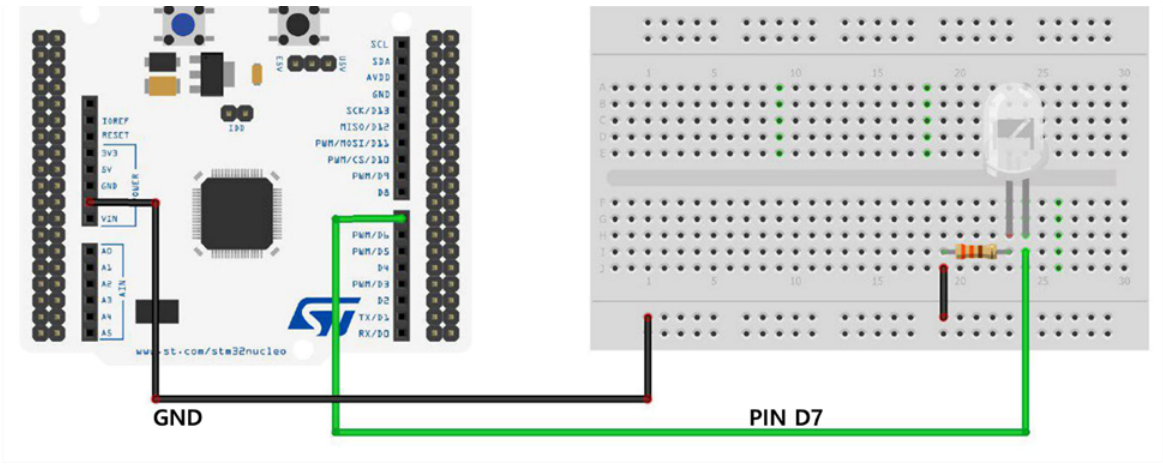
## Configuration

| **Button** | **LED** |
|---|---|
| Digital In | Digital Out |
| GPIOC, Pin 13 | PA5, PA6, PA7, PB6 |
| PULL-UP | Push-Pull, Pull-up, Medium Speed |

## Circuit Diagram

Circuit diagram

> You need to modify the circuit diagram




## Code

Your code goes here: [ADD Code LINK such as github]

Explain your source code with necessary comments.

```
// YOUR MAIN CODE ONLY
```

## Results

Experiment images and results

> Show experiment images /results

Add [demo video link]

## Discussion

1) Find out a typical solution for software debouncing and hardware debouncing. What method of debouncing did this NUCLEO board used for the push-button(B1)?

> Answer discussion questions

# Reference

Complete list of all references used (github, blog, paper, etc)

# Troubleshooting

(Option) You can write Troubleshooting section