

# LAB: EXTI & SysTick

---

**Date : 2023.10.13**

**Author/Partner: Duwon Yang**

**Github : [LINK](#)**

**Demo Video : [P1 LINK](#) / [P2 LINK](#)**

## Introduction

---

In this lab, We create a simple program using interrupt that display the number counting from 0 to 9 with Button at a rate of 1 second.

We used Nucleo-F411RE to implement this program, and the library by creating a HAL driver for GPIO digital input and output control.

## Requirement

### Hardware

- MCU
  - NUCLEO-F411RE
- Actuator/Sensor/Others:
  - 4 LEDs and load resistance
  - 7-segment display(5101ASR)
  - Array resistor (330ohm)
  - decoder chip(74LS47)
  - breadboard

### Software

- Keli uVision, CMSIS, EC\_HAL library

## Problem 1: Counting numbers on 7-Segment using EXTI Button

### 1-1. Create HAL library

save **ecEXTI.h**, **ecEXTI.c** in directory **EC \lib\**

Declare the following functions in library : **ecEXTI.h**

#### **ecEXTI.h**

```
void EXTI_init(GPIO_TypeDef *port, int pin, int trig_type, int priority);  
void EXTI_enable(uint32_t pin); // mask in IMR
```

```
void EXTI_disable(uint32_t pin); // unmask in IMR
uint32_t is_pending_EXTI(uint32_t pin);
void clear_pending_EXTI(uint32_t pin);
```

1-2. Procedure

1. Create a new project under the directory `\repos\EC\LAB\LAB_EXTI`
  - The project name is **"LAB\_EXTI"**
  - Create a new source file named as **"LAB\_EXTI.c"**
2. Include library in `\repos\EC\lib\`
  - **"ecGPIO.h, ecGPIO.c"**
  - **"ecRCC.h, ecRCC.c"**
  - **"ecEXTI.h, ecEXTI.c"**
3. Use the decoder chip(**74LS47**). Connect it to the bread board and 7-segment display.
4. Check if every number, 0 to 9 can be displayed properly on the 7-segment.
5. Create a code to display the number counting from 0 to 9 and repeats using library function of EXTI.

Configuration

Digital In for Button (B1)	Digital Out for 7-Segment decoder
Digital In	Digital Out
PC13	PA7, PB6, PC7, PA9
PULL-UP	Push-Pull, No Pull-up-Pull-down, Medium Speed

Circuit Diagram



-Effective in real-time systems and event-driven systems

-CPU resources are not wasted when it is idle

- **Disadvantages**

-frequent interrupt occurrences can increase CPU load

-Implementation and management of interrupt handling code can add complexity to the system

## 2. What would happen if the EXTI interrupt handler does not clear the interrupt pending flag?

check with your code

It can lead to continuous interrupt handling, delayed responses, and resource utilization issues in the microcontroller.

### Code

as button(B1) is pressed, 7-segment displays '0'~'9' according to External Input Interrupt.

```
void EXTI15_10_IRQHandler(void) {
    if (is_pending_EXTI(BUTTON_PIN)){
        cnt++;
        sevensegment_display(cnt % 10);
        if(cnt > 9) cnt = 0;
        clear_pending_EXTI(BUTTON_PIN);
    }
}
```

### Results

demo video : [LINK](#)

In this experiment, We implemented a program on the Nucleo board that can turn on 7-segment display and displays '0'~'9' as the button is pressed. In order to implement the above function, We use External Input Interrupt method(EXTI).

## Problem 2: Counting numbers on 7-Segment using SysTick

### 2-1. Create HAL library

save **ecSysTick.h**, **ecSysTick.c** in directory **EC \lib\**

Declare the following functions in library : **ecSysTick.h**

#### **ecEXTI.h**

```
void SysTick_init(uint32_t msec);
void delay_ms(uint32_t msec);
```

```
uint32_t SysTick_val(void);
void SysTick_reset (void);
void SysTick_enable(void);
void SysTick_disable (void)
```

2-2. Procedure

- 1. Create a new project under the directory \repos\EC\LAB\LAB\_SysTick
  - The project name is "LAB\_EXTI\_SysTick"
  - Create a new source file named as "LAB\_EXTI\_SysTick.c"
- 2. Include library in \repos\EC\lib\
  - "ecGPIO.h, ecGPIO.c"
  - "ecRCC.h, ecRCC.c"
  - "ecEXTI.h, ecEXTI.c"
  - "ecSysTick.h, ecSysTick.c"
- 3. Use the decoder chip(74LS47). Connect it to the bread board and 7-segment display.
- 4. Check if every number, 0 to 9 can be displayed properly on the 7-segment.
- 5. Create a code to display the number counting from 0 to 9 and repeats at the rate of 1 second.
- 6. When the button is pressed, it should start from '0' again.

Configuration

Digital In for Button (B1)	Digital Out for 7-Segment decoder
Digital In	Digital Out
PC13	PA7, PB6, PC7, PA9
PULL-UP	Push-Pull, No Pull-up-Pull-down, Medium Speed

Circuit Diagram



number counting from 0 to 9 displays at 7-segment at the rate of 1 second, and it repeats. When the button is pressed, its number reset as '0' using EXTI

6 / 7

```
}  
}
```

## Results

In this experiment, We implemented a program on the Nucleo board that can turn on 7-segment display and displays '0'~'9' at the rate of 1 second. as the button is pressed, start from '0' again. In order to implement the above function, use SysTick for making Timer. and use EXTI for button recognition.

demo video : [LINK](#)

## Reference

---

<https://ykkim.gitbook.io/ec/ec-course/lab/lab-report-template>

<https://ykkim.gitbook.io/ec/stm32-m4-programming/example-code#button-interrupt>

<https://ykkim.gitbook.io/ec/ec-course/lab/lab-exti-and-systick>

## Troubleshooting

---

delay function makes sevensegment stop counting

```
void EXTI15_10_IRQHandler(void) {  
    if (is_pending_EXTI(BUTTON_PIN)){  
        cnt++;  
        sevensegment_display(cnt % 10);  
        if(cnt > 9) cnt = 0;  
        delay_ms(500);  
        clear_pending_EXTI(BUTTON_PIN);  
    }  
}
```

At Problem 1, tried to delay\_ms function to make debouncing. However, as the button is pressed, sevensegment counted 0 to 1 and stopped. So, We check out the Interrupt EXTI and SysTick priority. after revising it, it works well.