TUTORIAL: External Interrupt

LED Toggle with Push-Button

Name: 김은찬 ID: 21801017

# I. Introduction

In this tutorial, we will learn how to use External Interrupt. We will create functions that capture the falling edge trigger by pushing a button using an external interrupt.

The objectives of this lab are to learn how to

* Configure External input (EXTI) interrupt with NVIC
* Create your own functions for configuration of interrupts

### Hardware

NUCLEO -F411RE

### Software

Keil uVision IDE, CMSIS, EC\_HAL

### Documentation

[STM32 Reference Manual](https://ykkim.gitbook.io/ec/stm32-m4-programming/hardware/nucleo-f411re#manual-documentation)

# II. Basics of External Interrupt (EXTI)

## A. Register List

List of external interrupt (EXTI) registers used in this tutorial [Reference Manual ch7, ch10.2]

|  |  |  |
| --- | --- | --- |
| Type | Register Name | Description |
| SYSCFG | SYSCFG\_EXTICRx | External Interrupt Configuration, x=1 to 4  EXTICR1: for pin0~pin3 , EXTICR2: for pin4~pin7, etc |
| EXTI\_ | EXTI\_IMR | Interrupt Mask |
|  | EXTI\_FTSR  EXTI\_RTSR | Falling/Rising Trigger Selection |

Schematic

텍스트, 도표, 평면도, 개략도이(가) 표시된 사진

자동 생성된 설명

## B. Register Setting

**(Digital Input Setting)**

* Enable GPIO peripheral clock **RCC->AHB1ENR**
* Configure DigitalIn pin

**(EXTI Setting)**

* Enable SYSCFG peripheral clock. **RCC->APB2ENR**
* Connect the corresponding external line to GPIO **SYSCFG->EXTICR**
* Configure the trigger edge. **EXTI->FTSR/RTSR**
* Configure Interrupt mask **EXTI->IMR**
* Enable EXTI. **EXTI->IMR**

**(NVIC Setting)**

* Configure the priority of EXTI interrupt request. **NVIC\_SetPriority(**)
* Enable EXTI interrupt request. **NVIC\_EnableIRQ(**)

**(EXTI Use)**

* Create user codes in handler **EXTIx\_IRQHandler()**
* Clear pending bit after interrupt call

# III. Tutorial

## A. Register Configuration

**1. Pin Initialization & Set LED and Push-button**

LED:Port A Pin 5 / Output / Push-Pull / No Pull-Up & No Pull-Down

Push-button: Port C Pin 13 / Input / No Pull-Up & No Pull-Down

|  |
| --- |
| *// code using your library functions*  *GPIO\_init(GPIOA, 5,OUTPUT);*  *GPIO\_pupd(GPIOA, 5, EC\_NONE);*  *GPIO\_init(GPIOC, 13,INPUT);*  *GPIO\_pupd(GPIOC, 13, EC\_NONE);* |

**2. Enable Peripheral Clock:** SYSCFGEN

* **RCC\_APB2ENR:** Enable SYSCFG

RCC->APB2ENR |= 1<<14

|  |
| --- |
| *// Paste RCC\_APB2ENR register map* |

**3. EXTI Initialization & Connect Push-button to EXTI line**

* **SYSCFG\_EXTICR4:** Connect PC\_13(push-button) to EXTI13 line

SYSCFG->EXTICR[3] &=~ 15≪4 // clear bits [3:0]

SYSCFG->EXTICR[3] |= 2<<4 // set to 0010 for PC[13]

|  |
| --- |
| // past register map here |

* **EXTI\_FTSR:** Enable Falling Trigger

EXTI->FTSR |= 1<<13 // TR13=1

|  |
| --- |
| // past register map here |

* **EXTI\_IMR:** Interrupt NOT masked (Enable)

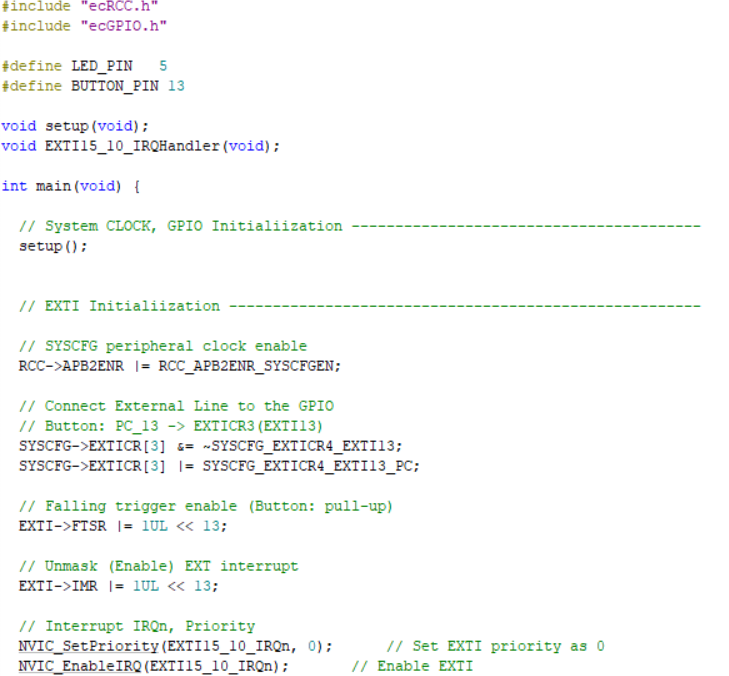
EXTI->IMR |= 1<<13 // MR13 = 1

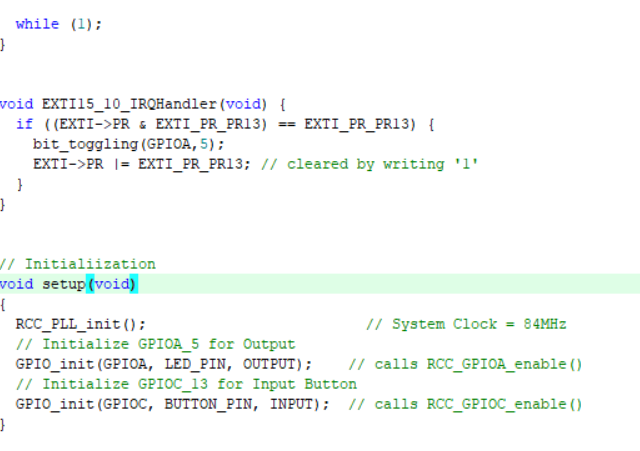
|  |
| --- |
| // past register map here |

## B. Programming

**Procedure**

* Create a new folder ‘**EC/Tutorial/TU\_EXTI/**’
* Open the program ‘Keil uVision5’ and create a new project.
* Name the project as ‘**TU\_EXTI**’.
* Create a new item called ‘TU\_EXTI.c’ and use the given source code [Click here to download](https://github.com/ykkimhgu/EC-student/tree/main/tutorial/tutorial-student)
* This is an example code for turning LED on/off with the button input trigger.
* Fill in the empty spaces in the code.
* Run the program and check your result.
* Your tutorial report must be submitted to LMS





## Appendix

[See here for MCU resources](https://ykkim.gitbook.io/ec/resource/nucleo-f411re)

1. Pin Configuration of NUCLE-F401RE

**텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 번호, 평행, 폰트이(가) 표시된 사진

자동 생성된 설명**

1. LED/Button Circuit Diagram

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 도표, 지도, 평면도이(가) 표시된 사진

자동 생성된 설명