TUTORIAL: Digital I/O

LED Toggle with Push-Button

Name: 김은찬 ID:21801017

# I. Introduction

We will learn how to control digital I/O of GPIOs of the MCU board to turn on/off an LED with a push-button input. The LED should be turned on when the button is pressed.

The objectives of this lab are to learn how to

* Read and configure registers of digital GPIO of MCU
* Program firmware to control digital input/output pins
* Create your own functions for GPIOs

### 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 GPIO IN

## A. GPIO Digital Out Register

List GPIO registers for this LAB

|  |  |  |
| --- | --- | --- |
| Type | Register Name | Description |
| GPIO | MODER | Mode: Input |
|  | PUPDR | Pull-Up Pull-Down: |
|  | IDR | Input Data Register |

Schematic

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

Process of GPIOx register initiation

|  |
| --- |
| 0. Enable Peripheral Clock (**AHB1ENR**)  1. Configure as Digital Input (**MODER**)  2. Configure pull-up/down resistors (**PUPDR**)  3. Read Data **(IDR)** |

# III. Tutorial

## A. Register Configuration

**1. GPIO: Digital In - Pin Initialization & Read PushButton**

Port C Pin 13 / Input // Pull-Up

use **#define BUTTON\_PIN 13**

* **MODER:** Input (MODER5[1:0]=01)

|  |
| --- |
| 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진  자동 생성된 설명 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Register** | **31** | **30** | **29** | **28** | **27** | **26** | **25** | **24** | **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Mask** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Value** | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | 0 | 1 | x | x | x | x | x | x | x | x | x | x |

GPIOC ->MODER &= ~(3<<( BUTTON\_PIN \*2));

GPIOC->MODER |= 0<<( BUTTON\_PIN \*2);

* **PUPDR:** pull-up (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

|  |
| --- |
| *Register map from reference manual* |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Register** | **31** | **30** | **29** | **28** | **27** | **26** | **25** | **24** | **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Mask** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Value** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

GPIOC->PUPDR &= ~(3<<( BUTTON\_PIN \*2));

GPIOC->PUPDR \_\_= ;

* **IDR:** Read Push-Button Value (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Register map from reference manual* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **Register** | **31** | **30** | **29** | **28** | **27** | **26** | **25** | **24** | **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | | **0** |
| **Mask** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| **Value** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |

GPIOC->IDR = ;

## B. Programming

**Preparation**

* + Open the program ‘Keil uVision5’ and create a new project.

“repos/EC/Tutorial/TU\_GPIO\_Digital\_InOut\_LED\_Button/”

* + Name the project as ‘TU\_GPIO\_Digital\_InOut\_LED\_Button’.
  + Create a new item called ‘TU\_GPIO\_Digital\_InOut\_LED\_Button.c’
  + Copy and paste from the source ‘[TU\_GPIO\_Digital\_InOut\_LED\_Button\_student.c](https://github.com/ykkimhgu/EC-student/tree/main/tutorial/tutorial-student)’.
  + Include provided ecRCC.h and ecRCC.c library files in your project.

**Exercise**

Fill in the empty spaces in the code. Then, compile(F7) and flash(F8) the source code

on the MCU board.

텍스트이(가) 표시된 사진

자동 생성된 설명

## 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