Tutorial: GPIO Digital I/O
- LED Toggle with Push-Button -

I. Overview

In this lab, 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, and vice versa.

The objectives of this lab are learning 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

Preparation:

- You need to read about the following registers: GPIO, 'STM Reference Manual pg. 145-163'

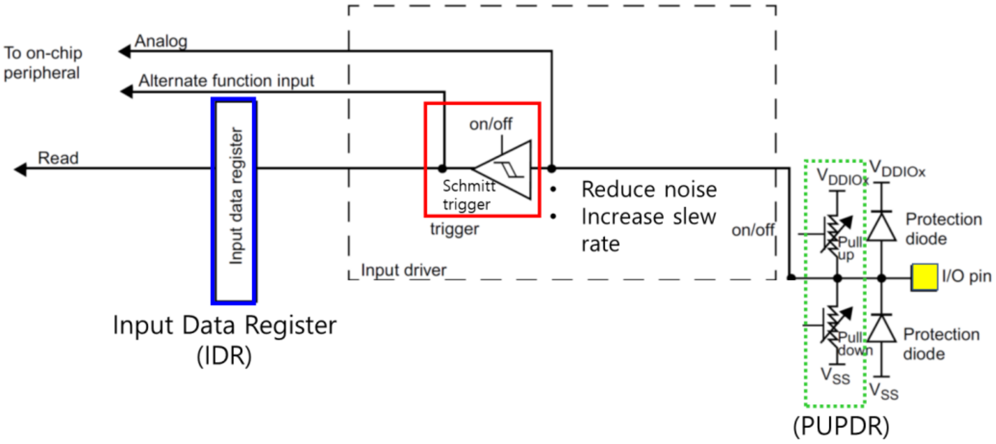
II. Tutorial

A. GPIO Digital Input 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



- Process of GPIO register initiation

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

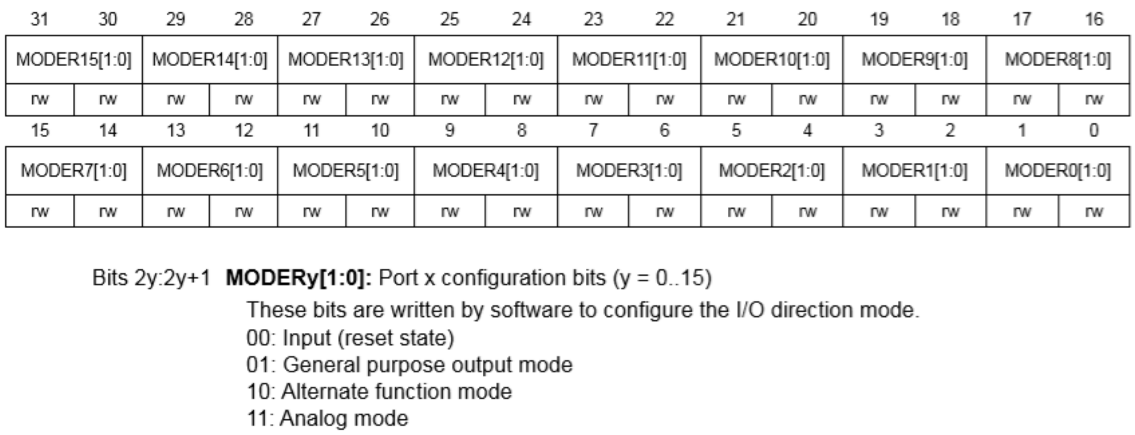
B. Register Setting

2. GPIO: Digital In - Pin Initialization & Read PushButton

Port C Pin 13 / Input // Pull-Up

# define BUTTON_PIN 13

- **MODER:** Input



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y:2y+1 **MODERy[1:0]:** Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O direction mode.
00: Input (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

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	0	0	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																																

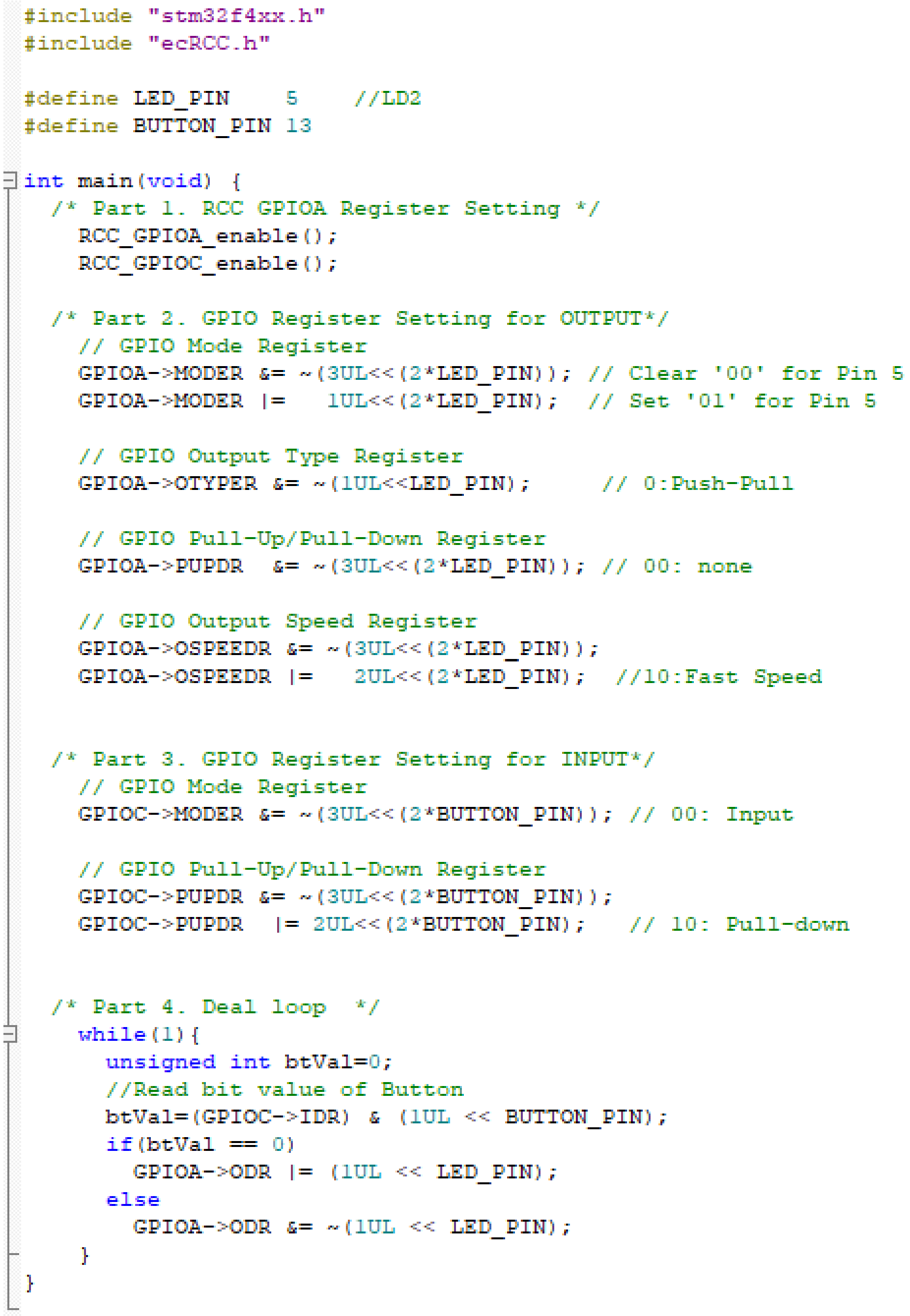
GPIOC->IDR = ____;

C. GPIO Register tutorial

- 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 the given source code on '**TU_GPIO_Digital_InOut_LED_Button_student.c**'.
- This is an example code of setting GPIO digital input.
- Include provided ecRCC.h and ecRCC.c library files in your project.



```c
#include "stm32f4xx.h"
#include "ecRCC.h"

#define LED_PIN    5    //LD2
#define BUTTON_PIN 13

int main(void) {
  /* Part 1. RCC GPIOA Register Setting */
    RCC_GPIOA_enable();
    RCC_GPIOC_enable();

  /* Part 2. GPIO Register Setting for OUTPUT*/
    // GPIO Mode Register
    GPIOA->MODER &= ~(3UL<<(2*LED_PIN)); // Clear '00' for Pin 5
    GPIOA->MODER |=   1UL<<(2*LED_PIN);  // Set '01' for Pin 5

    // GPIO Output Type Register
    GPIOA->OTYPER &= ~(1UL<<LED_PIN);      // 0:Push-Pull

    // GPIO Pull-Up/Pull-Down Register
    GPIOA->PUPDR  &= ~(3UL<<(2*LED_PIN)); // 00: none

    // GPIO Output Speed Register
    GPIOA->OSPEEDR &= ~(3UL<<(2*LED_PIN));
    GPIOA->OSPEEDR |=   2UL<<(2*LED_PIN);  //10:Fast Speed


  /* Part 3. GPIO Register Setting for INPUT*/
    // GPIO Mode Register
    GPIOC->MODER &= ~(3UL<<(2*BUTTON_PIN)); // 00: Input

    // GPIO Pull-Up/Pull-Down Register
    GPIOC->PUPDR &= ~(3UL<<(2*BUTTON_PIN));
    GPIOC->PUPDR  |= 2UL<<(2*BUTTON_PIN);    // 10: Pull-down


  /* Part 4. Deal loop  */
    while(1){
      unsigned int btVal=0;
      //Read bit value of Button
      btVal=(GPIOC->IDR) & (1UL << BUTTON_PIN);
      if(btVal == 0)
        GPIOA->ODR |= (1UL << LED_PIN);
      else
        GPIOA->ODR &= ~(1UL << LED_PIN);
    }
}
```

- Compile(F7) and flash(F8) the source code on to the MCU board.

Appendix

See here for MCU resources

1. Pin Configuration of NUCLE-F401RE

Figure 18. NUCLEO-F401RE

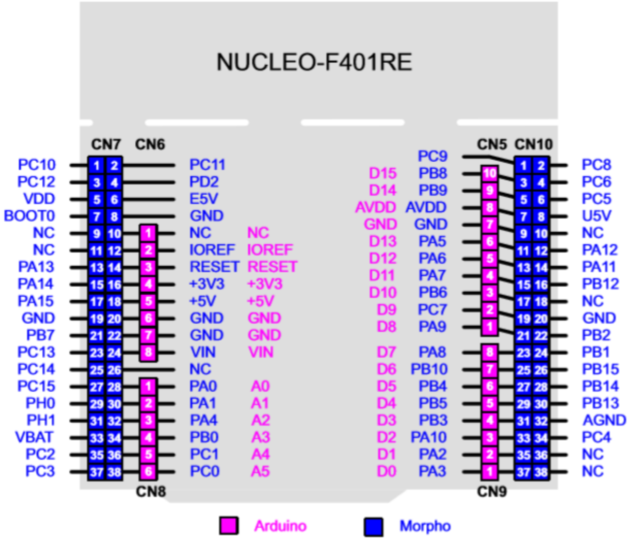
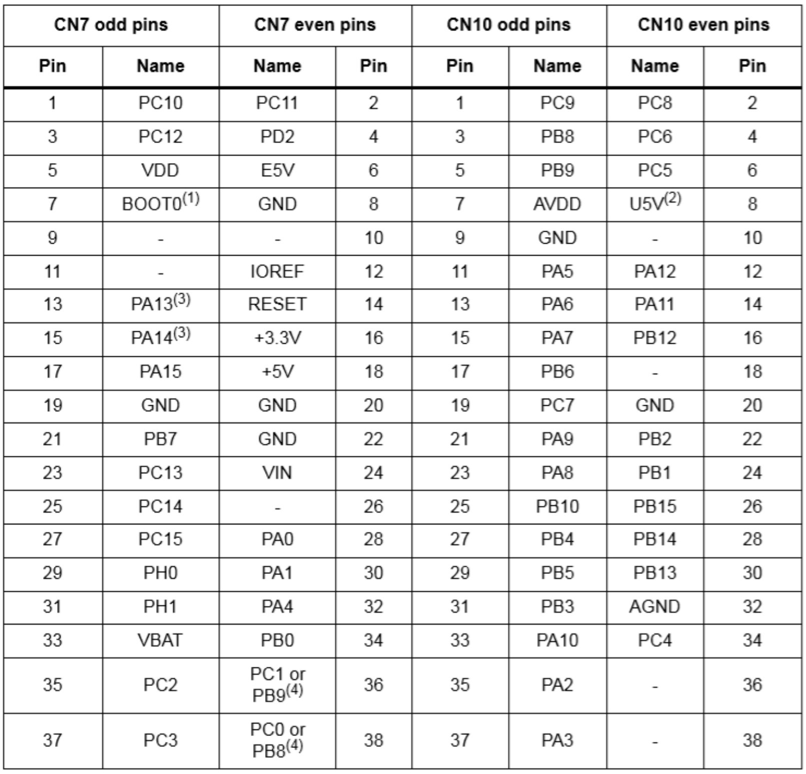


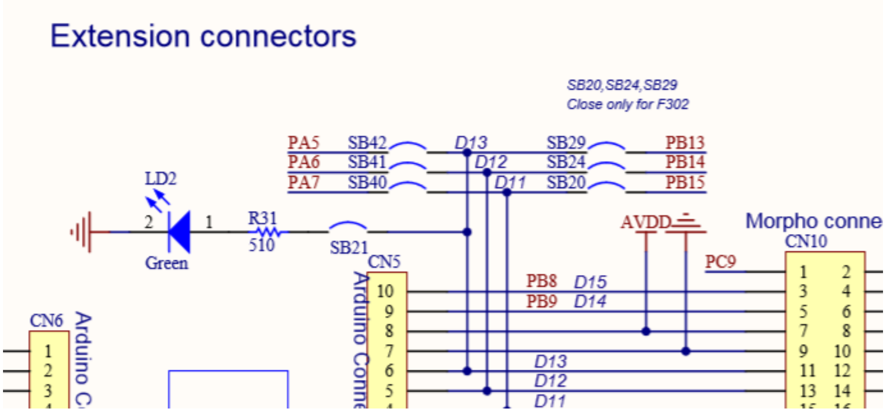
Table 29. ST morpho connector on NUCLEO-F401RE, NUCLEO-F411RE, NUCLEO-F446RE

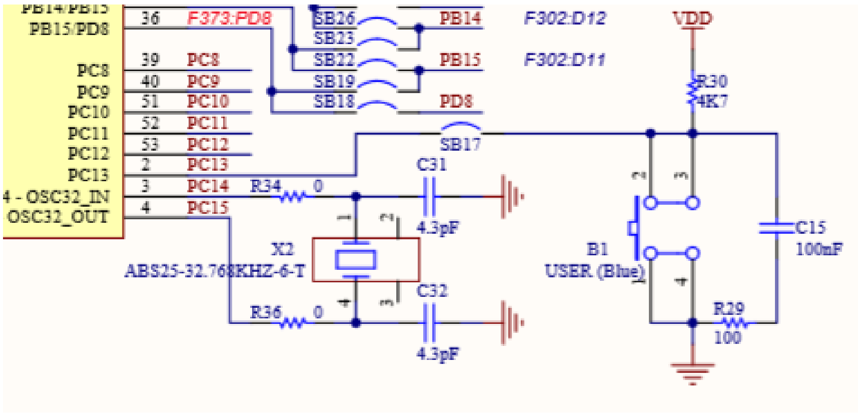


CN7 odd pins		CN7 even pins		CN10 odd pins		CN10 even pins	
Pin	Name	Name	Pin	Pin	Name	Name	Pin
1	PC10	PC11	2	1	PC9	PC8	2
3	PC12	PD2	4	3	PB8	PC6	4
5	VDD	E5V	6	5	PB9	PC5	6
7	BOOT0(1)	GND	8	7	AVDD	U5V(2)	8
9	-	-	10	9	GND	-	10
11	-	IOREF	12	11	PA5	PA12	12
13	PA13(3)	RESET	14	13	PA6	PA11	14
15	PA14(3)	+3.3V	16	15	PA7	PB12	16
17	PA15	+5V	18	17	PB6	-	18
19	GND	GND	20	19	PC7	GND	20
21	PB7	GND	22	21	PA9	PB2	22
23	PC13	VIN	24	23	PA8	PB1	24
25	PC14	-	26	25	PB10	PB15	26
27	PC15	PA0	28	27	PB4	PB14	28
29	PH0	PA1	30	29	PB5	PB13	30
31	PH1	PA4	32	31	PB3	AGND	32
33	VBAT	PB0	34	33	PA10	PC4	34
35	PC2	PC1 or PB9(4)	36	35	PA2	-	36
37	PC3	PC0 or PB8(4)	38	37	PA3	-	38

1. Default state of BOOT0 is 0. It can be set to 1 when a jumper is on pin5-7 of CN7.Two unused jumpers are available on CN11 and CN12 (bottom side of the board).
2. U5V is 5 V power from ST-LINK/V2-1 USB connector and it rises before +5V.
3. PA13 and PA14 share with SWD signals connected to ST-LINK/V2-1, it is not recommend to use them as IO pins if ST-LINK part is not cut.
4. Refer to *Table 10: Solder bridges* for details.

2. LED/Button Circuit Diagram


Extension connectors
SB20,SB24,SB29
Close only for F302
PA5
PA6
PA7
SB42
SB41
SB40
D13
D12
D11
SB29
SB24
SB20
PB13
PB14
PB15
LD2
2
1
Green
R31
510
SB21
AVDD
Morpho conne
CN10
PC9
CN5
Arduino Conne
10
9
8
7
6
5
PB8
D15
PB9
D14
D13
D12
D11
CN6
Arduino C
1
2
3


PB14/PB15
PB15/PD8
PC8
PC9
PC10
PC11
PC12
PC13
4 - OSC32_IN
OSC32_OUT
36
39
40
51
52
53
2
3
4
F373:PD8
PC8
PC9
PC10
PC11
PC12
PC13
PC14
PC15
SB26
SB23
SB22
SB19
SB18
PB14
PB15
PD8
F302:D12
F302:D11
SB17
VDD
R30
4K7
C31
4.3pF
R34
0
X2
ABS25-32.768KHZ-6-T
R36
0
C32
4.3pF
B1
USER (Blue)
C15
100nF
R29
100