

Tutorial: GPIO Digital Out

- LED On-

I. Overview

In this lab, we will learn how to control digital Output of GPIOs of the MCU board to turn on/off an LED.

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. Pre-Lab

A. Bit Operation

- Fill in the blanks. You should write the answer in both hexa-decimal and binary number.

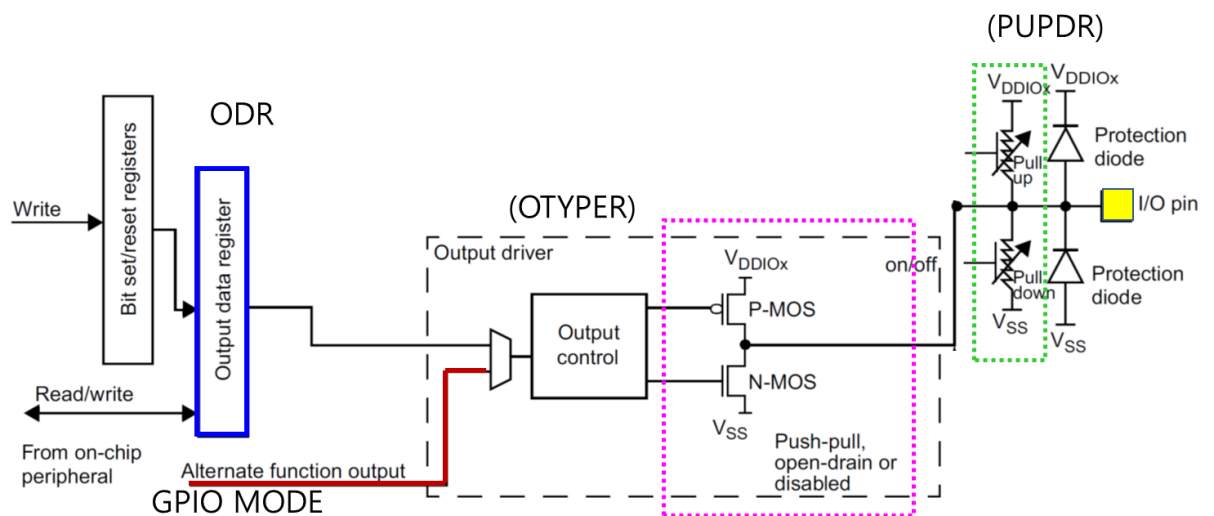
Bit operation	Description	Answer
$0xA \& 0x2$	AND	
$0x3 \mid 0xC$	OR	
$\sim 0x7$	NOT	
$0xF \wedge 0x5$	XOR(Toggle)	
$0x1 \ll 3$	Shift left	
$0xC \gg 2$	Shift right	
$0x11 \mid= 1 \ll 3$	Set bit	
$0xFF \&= \sim 1 \ll 4$	Clear bit	
$val = 0x0F \& 1 \ll 3$	Read bit	

B. GPIO Digital Out Register

- List GPIO registers for this LAB

Type	Register Name	Description
GPIO	GPIOx MODER	Mode: Output/Input/Analog
	GPIOx OTYPER	Output Type: Opendrain/Push-Pull
	GPIOx OSPEEDR	Output Speed:
	GPIOx PUPDR	Pull-Up Pull-Down:
	GPIOx ODR	Output Data Register

- Schematic



- Process of GPIOx register initiation

0. Enable Peripheral Clock (**AHB1ENR**)
1. Configure as Digital Output (**MODER**)
2. Configure pull-up/down resistors (**PUPDR**)
3. For Output: Configure Output Type (**OTYPE**)
4. For Output: Configure Output Speed (**OSPEEDR**)
5. Output Data (**ODR**)

C. Register Setting

1. GPIO: Digital Out - Pin Initialization & Set LED

Port A Pin 5 / Output / Push-Pull / No Pull-Up & No Pull-Down / Default Speed

define LED_PIN 5

- MODER:** Output

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]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
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]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

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	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

GPIOA->MODER &= ~(3<<(LED_PIN *2)); // clear bit 10, 11

GPIOA->MODER |= 1<<(LED_PIN *2); // set bit 10

- OTYPER:** Push-Pull

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:16 Reserved, must be kept at reset value.
 Bits 15:0 **OTy**: Port x configuration bits (y = 0..15)
 These bits are written by software to configure the output type of the I/O port.
 0: Output push-pull (reset state)
 1: Output open-drain

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																	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	
Value																	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	

GPIOA->OTYPER &= ~(1<< LED_PIN)

- **PUPDR:** no pull-up no pull-down

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																																

GPIOA->PUPDR &= ~(3<(< LED_PIN *2));

GPIOA->PUPDR __= _____

- **ODR:** Set LED

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																																

GPIOA->ODR __= _____;

D. GPIO Register tutorial

- Open the program 'Keil uVision5' and create a new project.
"repos/EC/Tutorial/TU_GPIO_Digital_Out_LED/"
- Name the project as 'TU_GPIO_Digital_Out_LED'.
- Create a new item called 'TU_GPIO_Digital_Out_LED.c'
- Copy and paste the given source code on '[TU_GPIO_Digital_Out_LED_student.c](#)'.
- This is an example code of turning ON LED *without* button input. You will modify this code to include the push-button input. Do not worry if you do not understand what this code means yet. You will learn one by one in the following few weeks.
- Download [ecRCC_student.h](#) and [ecRCC_student.c](#), you can rename them as **ecRCC.h** and **ecRCC.c**.
- Then save them in lib folder. "repos/EC/lib"
- Include provided ecRCC.h and ecRCC.c library files in your project.

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

#define LED_PIN    5    //LD2

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

    /* Part 2. GPIO Register Setting */
    // 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

    // Dead loop & program hangs here
    while(1){
        // GPIOA->ODR = 1UL << LED_PIN; // Set LED_PIN = H, others=L
        GPIOA->ODR |= (1UL << LED_PIN); // Change only LED_PIN = H
    }
}
```

- Compile(F7) and flash(F8) the source code on to the MCU board.
- Verify the program by checking if the LED is turned ON.

Appendix

1. Pin Configuration of NUCLE-F401RE

Figure 18. NUCLEO-F401RE

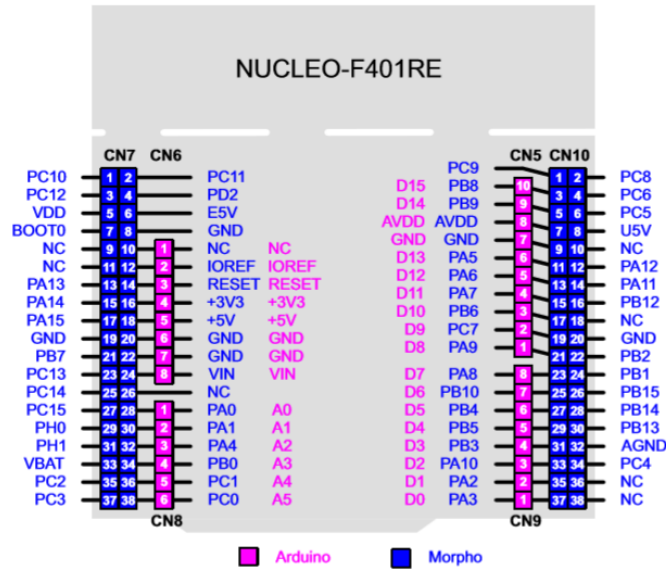


Table 29. ST morpho connector on NUCLEO-F401RE, NUCLEO-F411RE, NUCLEO-F446RE

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

