

TUTORIAL: Digital Out

Name: Yechan Kim

ID: 22100153

I. Introduction

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

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

Hardware

NUCLEO -F411RE

Software

Keil uVision IDE, CMSIS, EC_HAL

Documentation

[STM32 Reference Manual](#)

II. Basics of GPIO Out

A. Bit Operation

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

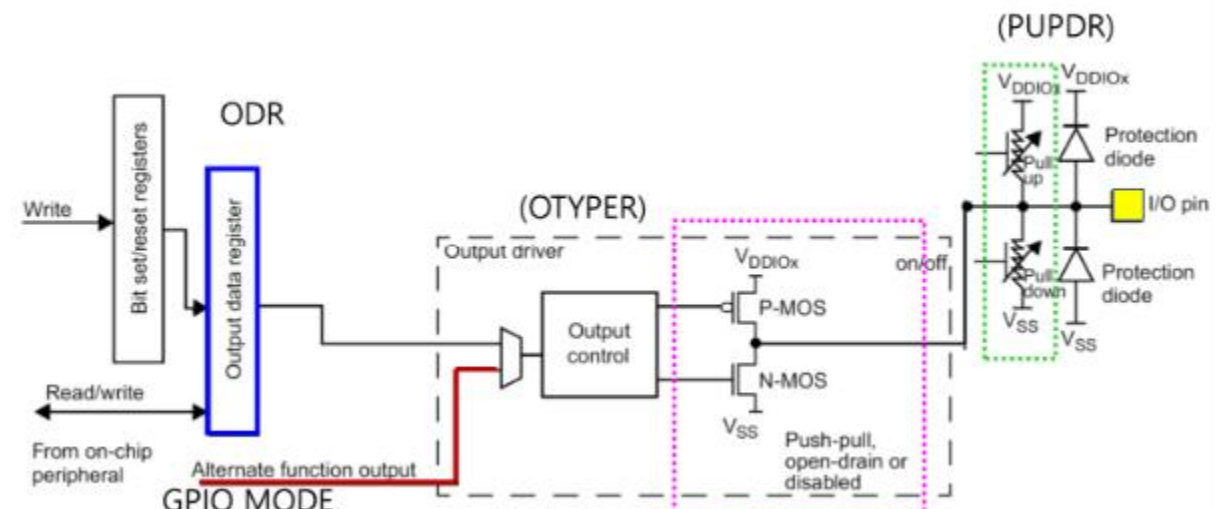
Bit operation	Description	Answer
0xA & 0x2	AND	0x2
0x3 0xC	OR	0xF
~0x7	NOT	0x8
0xF ^ 0x5	XOR(Toggle)	0xA
0x1 << 3	Shift left	0x8
0xC >> 2	Shift right	0x3
0x11 = 1<<3	Set bit	0x19
0xFF &= ~1<<4	Clear bit	0xEF
val=0x0F & 1<<3	Read bit	0x0F

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**)

- **ODR:** Set LED (ODR5=1)

GPIOA->ODR |= 1<<LED_PIN;

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
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

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
Initial	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
Logic	Bitwise OR																															
mask	0	0	0	0	0	0	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
Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x

B. Programming

Preparation

- 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(file) and name it as 'TU_GPIO_Digital_Out_LED.c'
- Copy and paste from the source code '[TU_GPIO_Digital_Out_LED_student.c](#)'.
- Download [ecRCC_student.h](#) and [ecRCC_student.c](#). Then, change the file names as **ecRCC2.h** and **ecRCC2.c**.
- Save them in \lib folder: “**repos/EC/include**”

Exercise

This is an example code of turning ON LED *without* 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.

Fill in the empty spaces in the code. Then, compile(F7) and flash(F8) the source code onto the MCU board. Verify the program by checking if the LED is turned ON.

Solution

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

#define LED_PIN 5 //LED

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<<(3*LED_PIN)); // 00: none

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

    // Dead loop & program hangs here
    while(1){
        // GPIOA->ODR = 1UL<< LED_PIN; // Set LED_PIN = 0, otherwise=1.
        GPIOA->ODR |= (1UL<< LED_PIN); // Change only LED_PIN = 1
    }
}
```

Appendix

[See here for MCU resources](#)

1. Pin Configuration of NUCLEO-F401RE

Figure 18. NUCLEO-F401RE

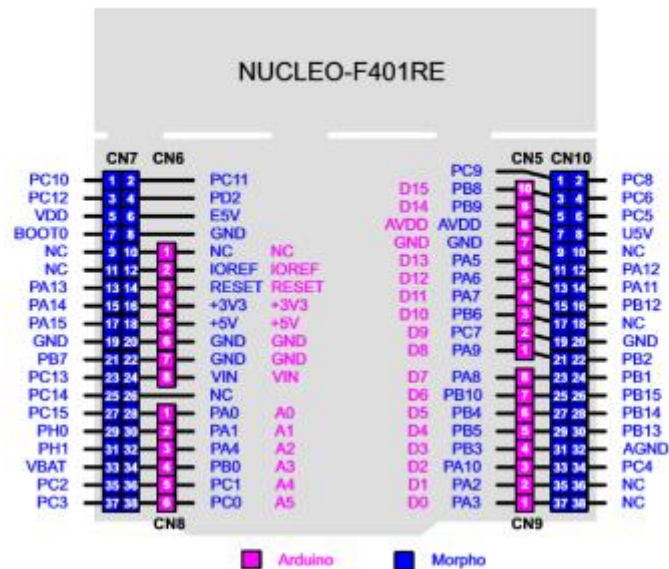


Table 29. ST morpho connector on NUCLEO-F401RE, NUCLEO-F411RE, NUCLEO-F446RE

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	ESV	6	5	PB9	PC5	6
7	BOOT0 ⁽¹⁾	GND	8	7	AVDD	USV ⁽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

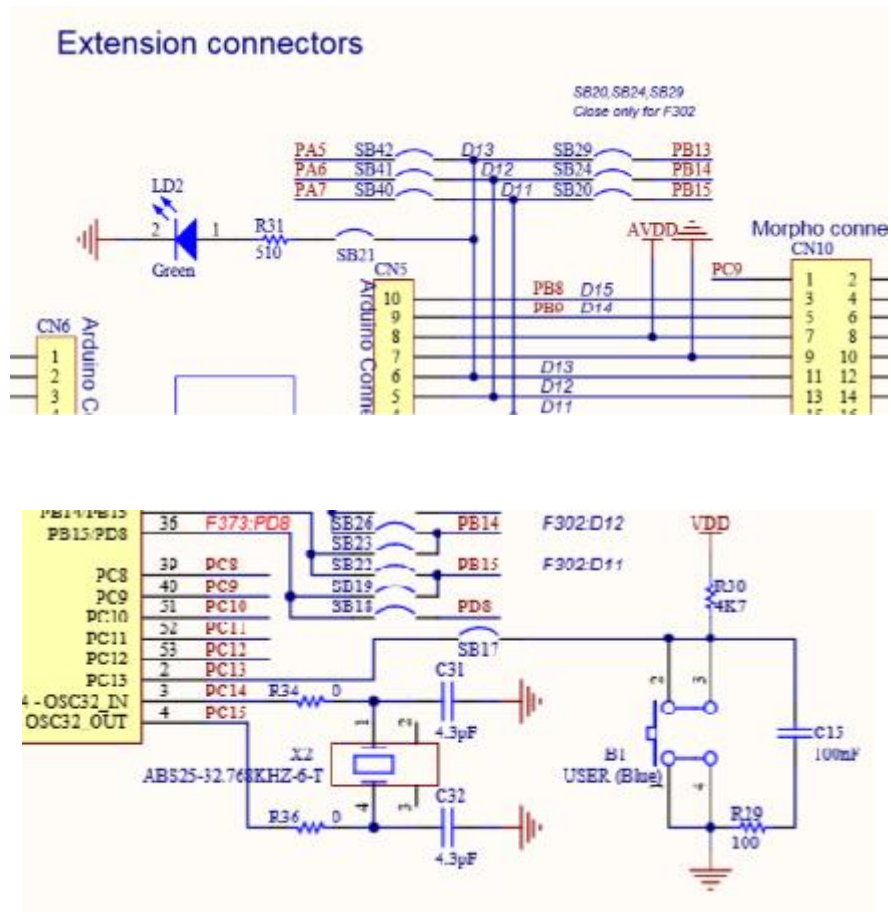
1. Default state of BOOT0 is 0. It can be set to 1 when a jumper is on pins-7 of CN7. Two unused jumpers are available on CN11 and CN12 (bottom side of the board).

2. USV 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 I/O pins if ST-LINK part is not cut.

4. Refer to [Table 10. Solder bridges](#) for details.

2. LED/Button Circuit Diagram



Results

Source Code

```
/**
 * *****
 * @author SSSLAB
 * @Mod 2024-8-23 by YKKIM
 * @brief Embedded Controller: Tutorial Digital Out
 * - Turn on LED LD2
 *
 * *****
 */

// GPIO Mode : Input(00), Output(01), AlterFunc(10), Analog(11, reset)
// GPIO Speed : Low speed (00), Medium speed (01), Fast speed (10), High speed (11)
// GPIO Output Type: Output push-pull (0, reset), Output open drain (1)
// GPIO Push-Pull : No pull-up, pull-down (00), Pull-up (01), Pull-down (10), Reserved (11)

#include "stm32f4xx.h"
#include "ecRCC2.h"

#define LED_PIN PA_5 //LD2
#define LED_PIN 5

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

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

    // GPIO Output Type Register
    GPIOA->OTYPER &= ~(1<< LED_PIN) ;
    // Clear '00'

    // 0:Push-Pull

    // GPIO Pull-Up/Pull-Down Register
```

Embedded Controller

```
GPIOA->PUPDR  &= ~(3<<( LED_PIN *2));
// 00: none

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

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

return 0;
}
```

Screen Shot

```
Executing task: C:\Users\User\.platformio\penv\Scripts\platformio.exe run --target upload --environment TU_GPIO_Digital_Out_LED_student

Processing TU_GPIO_Digital_Out_LED_student (platform: ststm32; board: nucleo_f411re; framework: cmsis)
-----
Verbose mode can be enabled via '-v, --verbose' option
CONFIGURATION: https://docs.platformio.org/page/boards/ststm32/nucleo_f411re.html
PLATFORM: ST STM32 (19.3.0) > ST Nucleo F411RE
HARDWARE: STM32F411RET6 100MHz, 128KB RAM, 512KB Flash
DEBUG: Current (stlink) On-board (stlink) External (blackmagic, cmsis-dap, jlink)
PACKAGES:
- framework-cmsis @ 2.50501.200527 (5.5.1)
- framework-cmsis-stm32f4 @ 2.6.11
- tool-dfuutil @ 1.11.0
- tool-dfuutil-arduino @ 1.11.0
- tool-ldscripts-ststm32 @ 0.2.0
- tool-openocd @ 3.1200.0 (12.0)
- tool-stm32duino @ 1.0.2
- tool-stm32flash @ 0.7.0
- toolchain-gccarmnoneabi @ 1.70201.0 (7.2.1)
LDF: Library Dependency Finder -> https://bit.ly/configure-pio-ldf
LDF Modes: Finder ~ chain, Compatibility ~ soft
Found 0 compatible libraries
Scanning dependencies...
No dependencies
Building in release mode
Compiling .pio\build\TU_GPIO_Digital_Out_LED_student\FrameworkCMSIS\gcc\startup_stm32f411xe.o
Compiling .pio\build\TU_GPIO_Digital_Out_LED_student\FrameworkCMSIS\system_stm32f4xx.o
Compiling .pio\build\TU_GPIO_Digital_Out_LED_student\src\include\ecPinNames.o
Compiling .pio\build\TU_GPIO_Digital_Out_LED_student\src\include\ecRCC2.o
Compiling .pio\build\TU_GPIO_Digital_Out_LED_student\src\include\ecSTM32_simple.o
Compiling .pio\build\TU_GPIO_Digital_Out_LED_student\src\tutorial\TU_GPIO_Digital_Out_LED\TU_GPIO_Digital_Out_LED_student.o
include\ecSTM32_simple.c: In function 'GPIO_init':
include\ecSTM32_simple.c:25:3: warning: implicit declaration of function 'RCC_GPIOA_enable'; did you mean 'ARM_MPU_Enable'? [-Wimplicit-function-declaration]
    RCC_GPIOA_enable();
    ^~~~~~
    ARM_MPU_Enable
include\ecSTM32_simple.c:27:3: warning: implicit declaration of function 'RCC_GPIOB_enable'; did you mean 'ARM_MPU_Enable'? [-Wimplicit-function-declaration]
    RCC_GPIOB_enable();
    ^~~~~~
```

Embedded Controller

```
ARM_MPU_Enable
include\ecSTM32_simple.c:29:3: warning: implicit declaration of function 'RCC_GPIOC_enable'; did you mean 'ARM_MPU_Enable'? [-Wimplicit-function-declaration]
RCC_GPIOC_enable();
~~~~~
ARM_MPU_Enable
Linking .pio\build\TU_GPIO_Digital_Out_LED_student\firmware.elf
Checking size .pio\build\TU_GPIO_Digital_Out_LED_student\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [          ] 0.4% (used 512 bytes from 131072 bytes)
Flash: [          ] 2.8% (used 14608 bytes from 524288 bytes)
Configuring upload protocol...
AVAILABLE: blackmagic, cmsis-dap, jlink, nbed, stlink
CURRENT: upload_protocol = stlink
Uploading .pio\build\TU_GPIO_Digital_Out_LED_student\firmware.elf
xPack Open On-Chip Debugger 0.12.0-01004-g9ea7f3d04-dirty (2021-01-30-15:04)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
debug_level: 1

srst_only separate srst_nogate srst_open_drain connect_deassert_srst

[stm32f4x.cpu] halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x88000c98 msp: 0x20020000
** Programming Started **
** Programming Finished **
** Verify Started **
** Verified OK **
** Resetting Target **
shutdown command invoked

===== [SUCCESS] Took 4.16 seconds =====

Environment          Status    Duration
-----
TU_GPIO_Digital_Out_LED_student  SUCCESS  00:00:04.156
===== 1 succeeded in 00:00:04.156 =====

Terminal will be reused by tasks, press any key to close it.
```