LAB: GPIO Digital InOut 7-segment

Date: 2023-10-02

Author/Partner: Noh YunKi/Kim EunChan

Demo Video: [link][https://youtu.be/0d-F9z-DcdY]

Introduction

In this lab, we will count each numbers(0~9) and display each numbers on the 7-segment display device(5101ASR). To do this work, we have to understand how to work this 7-segment display device. In addition to this, because there are hardware method and software method to operate 7-segment display device, we have to understand about 'decoder chip(74LS47)'(in case of hardware) and about how to compose matrix to make decoder in programming.

Requirement

Hardware

- MCU
  - NUCLEO-F411RE
- Actuator/Sensor/Others:
  - 7-segment display(5101ASR)
  - Array resistor (330 ohm)
  - decoder chip(74LS47)
  - breadboard

Software

- Keil uVision, CMSIS, EC_HAL library

Problem 1: Connecting 7-segment Display

Procedure

To display number on the 7-segment displays, decoder is necessary whether by using hardware or software.

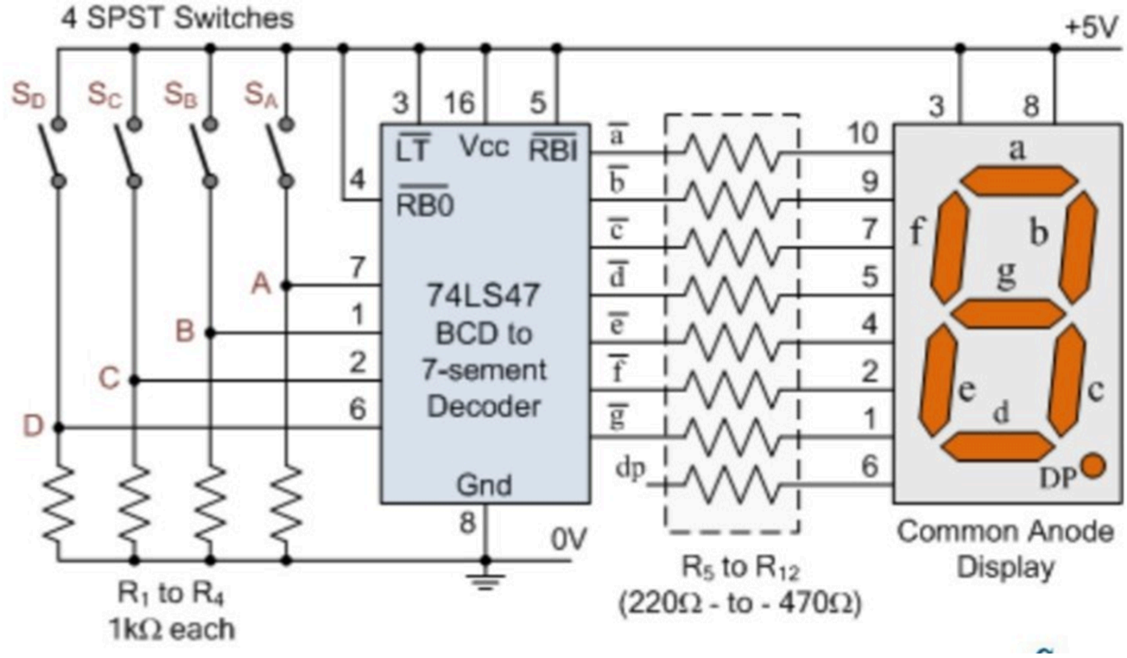


Figure 1. 7-Segment Decoder Hardware

In this lab, I will make 7-segment decoder with the MCU programming.

Connecting Diagram

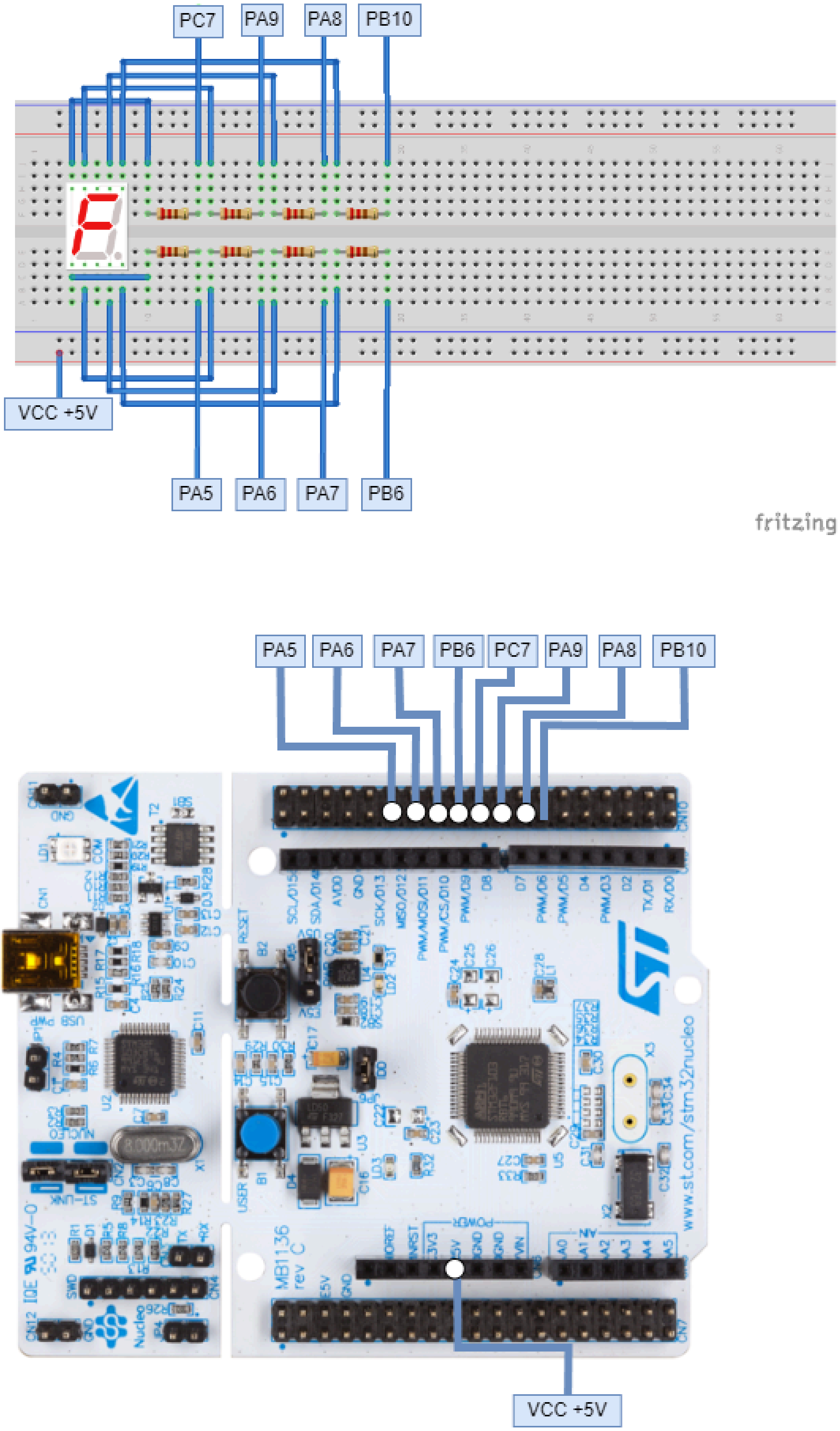


Figure 2. 7-Segment Diagram

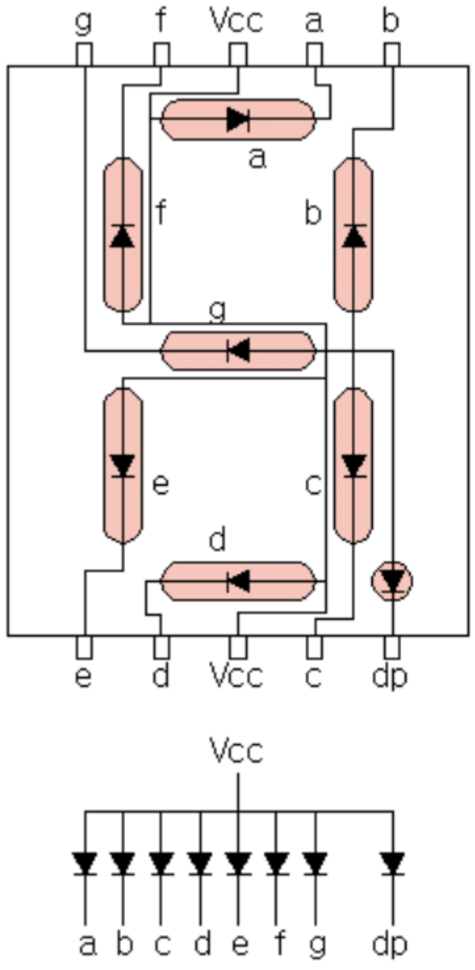
Discussion

1. Draw the truth table for the BCD 7-segment decoder with the 4-bit input.

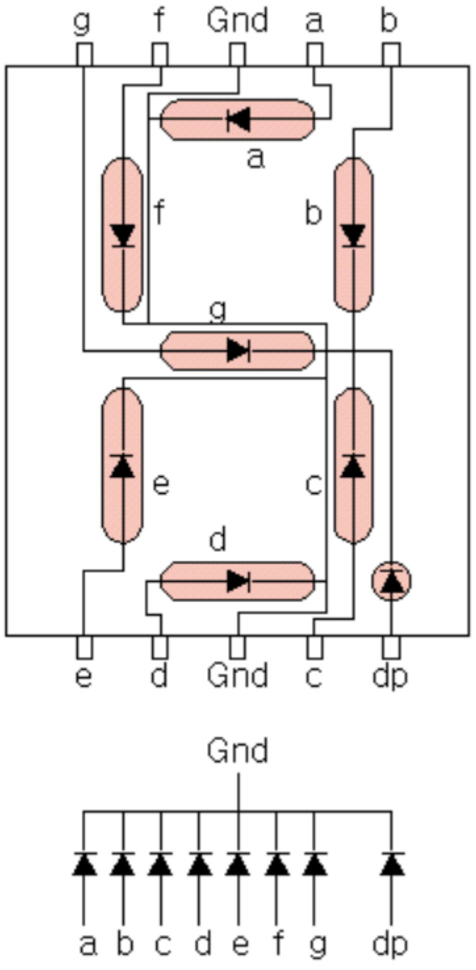
A	B	C	D	a	b	c	d	e	f	g
0	**0**	**0**	**0**	1	1	1	1	1	1	0
0	**0**	**0**	**1**	0	1	1	0	0	0	0
0	**0**	**1**	**0**	1	1	0	1	1	0	1
0	**0**	**1**	**1**	1	1	1	1	0	0	1
0	**1**	**0**	**1**	1	0	1	1	0	1	1
0	**1**	**1**	**0**	1	0	1	1	1	1	1
0	**1**	**1**	**1**	1	1	1	0	0	0	0
1	**0**	**0**	**0**	1	1	1	1	1	1	1
1	**0**	**0**	**1**	1	1	1	1	0	1	1

Table 1. BCD 7-segment decoder with the 4-bit input

2. What are the common cathode and common anode of 7-segment display?



common-anode type



common-cathode type

Figure 3. Common-Anode & Cathode

Common-Anode is the type of that all of the LED's positive poles are connected at the same one node. These positive poles' one node is connected with source(5V). In contrast of Anode, Common-Cathode is the type of that all of the LED's negative poles are connected at the same one node. These negative pole's node is connected with ground(GND). Because of circuit's structural difference between Common-Anode and Cathode, there are difference about how to work LED pins. When turning on LEDs of Common-Anode type, each LED's pin must to be 0V. However in the case of Common-Cathode type, we have to supply 5V on the LED's pin to turn on each LED. In this way, each types of 7-segment display shows each numbers by using LEDs.

3. Does the LED of a 7-segment display (common anode) pin turn ON when 'HIGH' is given to the LED pin from the MCU?

No. Because this Common-Anode type turn on the led's pin when the led's pin is connected with ground, to turn on the led pin, 'LOW' must to be given to the LED pin from the MCU.

Problem 2: Display 0~9 with button press

Configuration

Digital In for Button (B1)	Digital Out for 7-Segment
Digital In	Digital Out
PC13	PA5, PA6, PA7, PB6, PC7, PA9, PA8, PB10 ('a'~'h', respectively)
PULL-UP	Push-Pull, No Pull-up-Pull-down, Medium Speed

Table 2. Configuration

Code

- main code

```
/**
*******************************************************************************
* @author:  NohYunKi
* @date: 23-10-03
* @content:   23_2_Embedded Controller[LAB Digital In/Out 7-segment Display]
*******************************************************************************
*/

#include "stm32f4xx.h"
#include "ecRCC.h"
#include "ecGPIO.h"

#define PA5 5
#define PA6 6
#define PA7 7
#define PA8 8
#define PA9 9
#define PB6 6
#define PB10 10
#define PC7 7
```

```
#define PC13 13

void setup(void);
void sevensegment_init(void);
void sevensegment_decoder(uint8_t  num);

int main(void) {
    // Initialiization --------------------------------------------------------
    setup();
    unsigned int cnt = 0;

    // Inifinite Loop ----------------------------------------------------------
    while(1){
        // Input: 0 ~ 9
        sevensegment_decoder(cnt % 10);
        // Button pressed --> count
        if(GPIO_read(GPIOC, BUTTON_PIN) == 0) cnt++;
        // count > 9 --> reset
        if (cnt > 9) cnt = 0;
        // set volatile variable to maintain 'i' value in infivite loop.
        for(volatile int i = 0; i < 250000;i++){}
    }
}

void setup(void){


    RCC_HSI_init();
    sevensegment_init();

}

void sevensegment_init(void)
{
    // Array for Port's pin number
    int PA[5]={PA5,PA6,PA7,PA8,PA9};
    int PB[2]={PB6,PB10};

    //--------- 'GPIO_setting' is in 'ecGPIO.c' ---------//

    /*-----------------------------------------------------
   GPIOA setting:
   Digital Out, Push-Pull, No Pull-up-Pull-down, Medium Speed
   ------------------------------------------------------*/
    for(int i = 0; i < 5; i++){
        GPIO_setting(GPIOA, PA[i], OUTPUT, EC_PUSH_PULL, EC_NONE, EC_MEDIUM);
    }

    /*-----------------------------------------------------
   GPIOB setting:
   Digital Out, Push-Pull, No Pull-up-Pull-down, Medium Speed
   ------------------------------------------------------*/
    for(int i = 0; i < 2; i ++){
        GPIO_setting(GPIOB, PB[i], OUTPUT, EC_PUSH_PULL, EC_NONE, EC_MEDIUM);
    }

    /*-----------------------------------------------------
   GPIOC7 setting:
   Digital Out, Push-Pull, No Pull-up-Pull-down, Medium Speed
   ------------------------------------------------------*/
    GPIO_setting(GPIOC, PC7, OUTPUT, EC_PUSH_PULL, EC_NONE, EC_MEDIUM);

    /*-----------------------------------------------------
   GPIOC13 setting:
   Digital In, Pull-up
   ------------------------------------------------------*/
    GPIO_init(GPIOC, 13, INPUT);
    GPIO_pupd(GPIOA, PA7, EC_PU);                 // 3.Pull-up

    // Initial setting '0'
    GPIO_write(GPIOA, PA5, LOW);
    GPIO_write(GPIOA, PA6, LOW);
    GPIO_write(GPIOA, PA7, LOW);
    GPIO_write(GPIOA, PA8, LOW);
    GPIO_write(GPIOA, PA9, LOW);
    GPIO_write(GPIOB, PB6, LOW);
    GPIO_write(GPIOB, PB10, LOW);
    GPIO_write(GPIOC, PC7, HIGH);
}

void sevensegment_decoder(uint8_t  num)
{
    /*
    Matrix for each number
    : {PA5,PA6,PA7,PA8,PA9,PB6,PB10,PC10}x10
    */
    int number[10][8]={
        {0,0,0,0,0,0,0,1},  //  zero
        {1,1,0,1,1,0,0,1},  //  one
        {0,0,1,0,1,0,0,0},  //  two
        {1,0,0,0,1,0,0,0},  //  three
        {1,1,0,1,0,0,0,0},  //  four
        {1,0,0,0,0,0,1,0},  //  five
        {0,0,0,0,0,0,1,0},  //  six
        {1,1,0,0,1,0,0,1},  //  seven
        {0,0,0,0,0,0,0,0},  //  eight
        {1,1,0,0,0,0,0,0}   //  nine
    };

    // Array for Port pin's number
    int PA[5]={PA5,PA6,PA7,PA8,PA9};
    int PB[2]={PB6,PB10};

    // Turn on or off each LED pins
    for(int i=0; i < 8; i++){
        if(i < 5) GPIO_write(GPIOA, PA[i], number[num][i]); // PA5 ~ PA9
        else if(i == 5 || i == 6) GPIO_write(GPIOB, PB[i-5], number[num][i]); //
PB6, PB10
        else GPIO_write(GPIOC, PC7, number[num][i]); // PC10
    }
}
```

- GPIO_setting

```
void GPIO_setting(GPIO_TypeDef *Port, int pin, int mode, int otype, int pupd, int
ospeed){
    GPIO_init(Port, pin, mode);
    GPIO_otype(Port, pin, otype);
    GPIO_pupd(Port, pin, pupd);
    GPIO_ospeed(Port, pin, ospeed);
}
```

Results

Demo video: [link][https://youtu.be/0d-F9z-DcdY]

Problem 3: Using both 7-Segment Decoder and 7-segment display

Connection Diagram

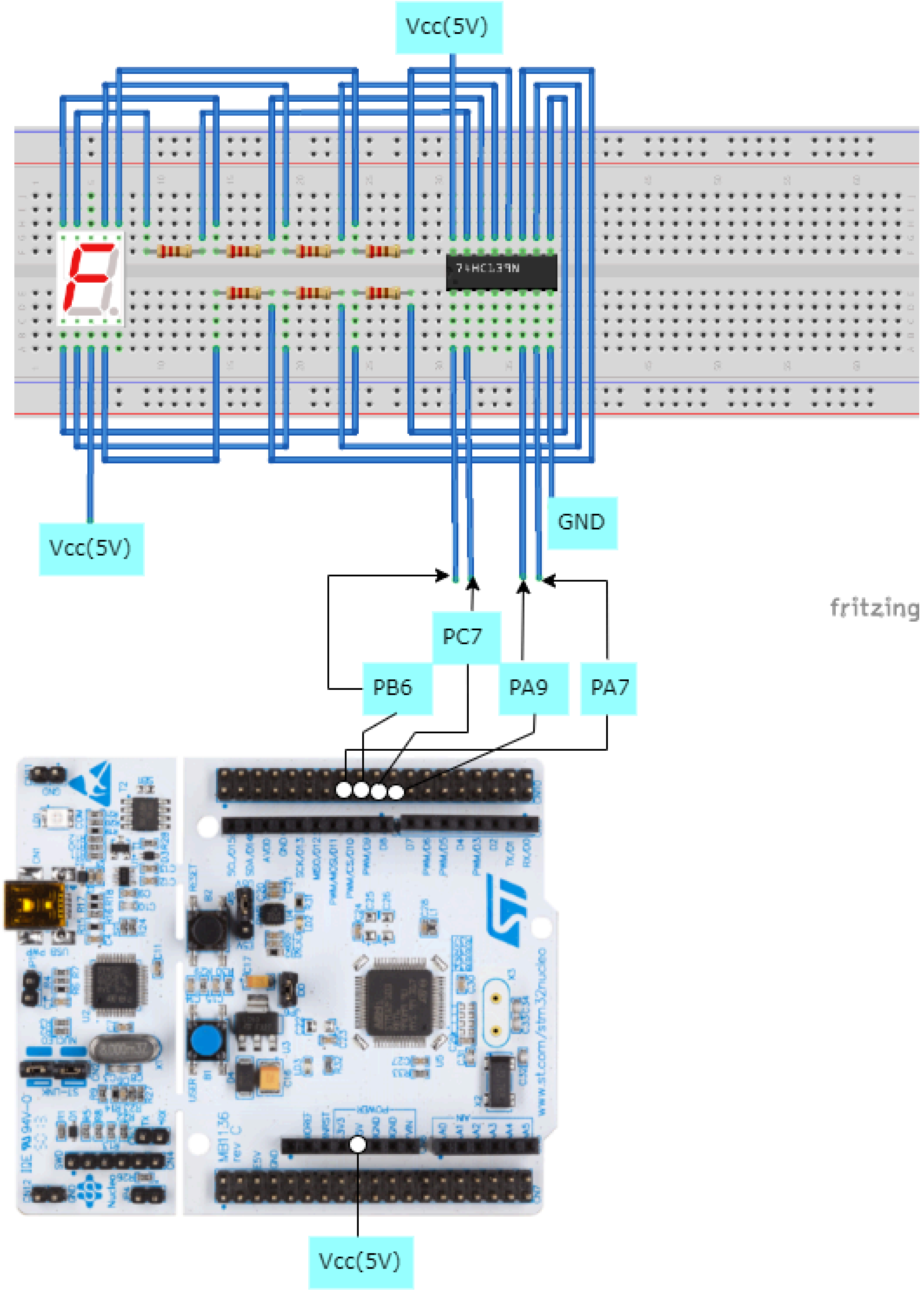


Figure 4. 7-segment decoder(74LS47)

Configuration

Digital In for Button (B1)	Digital Out for 7-Segment
Digital In	Digital Out
PC13	PA7, PB6, PC7, PA9
PULL-UP	Push-Pull, No Pull-up-Pull-down, Medium Speed