LAB: GPIO Digital InOut 7-segment

Date: 2022-10-09

Author/Partner: 21800447 Jeahyun Oh / 21800805 SeungEung Hwang

Github: https://github.com/Ohjeahyun1/EC-jeahyun-447.git

Demo Video: https://youtu.be/C21kMEDQyEM

Introduction

In this lab, you are required to create a simple program to control a 7-segment display to show a decimal number (0~9) that increases by pressing a push-button.

You must submit

- LAB Report (*.md & *.pdf)
- Zip source files(main*.c, ecRCC.h, ecGPIO.h etc...).
    - Only the source files. Do not submit project files

Requirement

Hardware

- MCU
    - NUCLEO-F401RE
- Actuator/Sensor/Others:
    - 7-segment display(5101ASR)
    - Array resistor (330 ohm)
    - breadboard

Software

- Keil uVision, CMSIS, EC_HAL library

Problem 1: Connecting 7-Segment

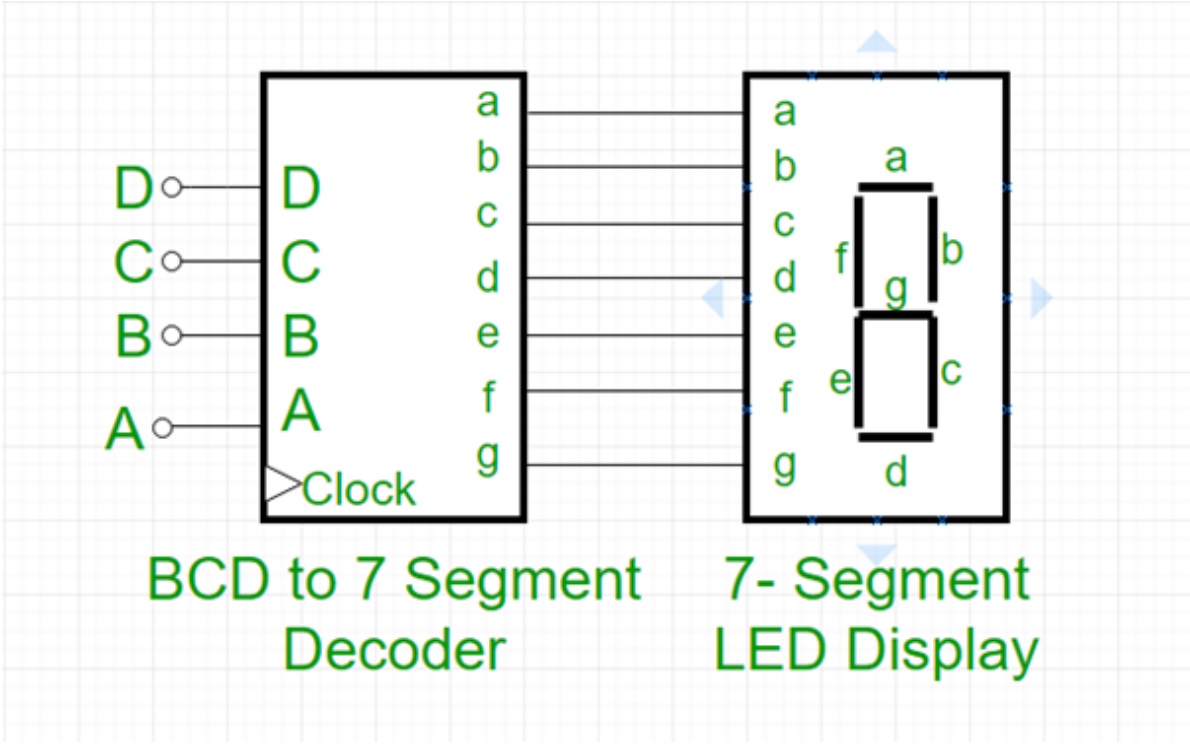
Procedure

Review 7-segment Decoder and Display from Digital Logic lecture.

- Read here: 7-segment BCD tutorial

The popular BCD 7-segment decoder chips are **74LS47 and CD4511**.

Instead of using the decoder chip, we are going to make the 7-segment decoder with the MCU programming.



Connect the common anode 7-segment with the given array resistors.

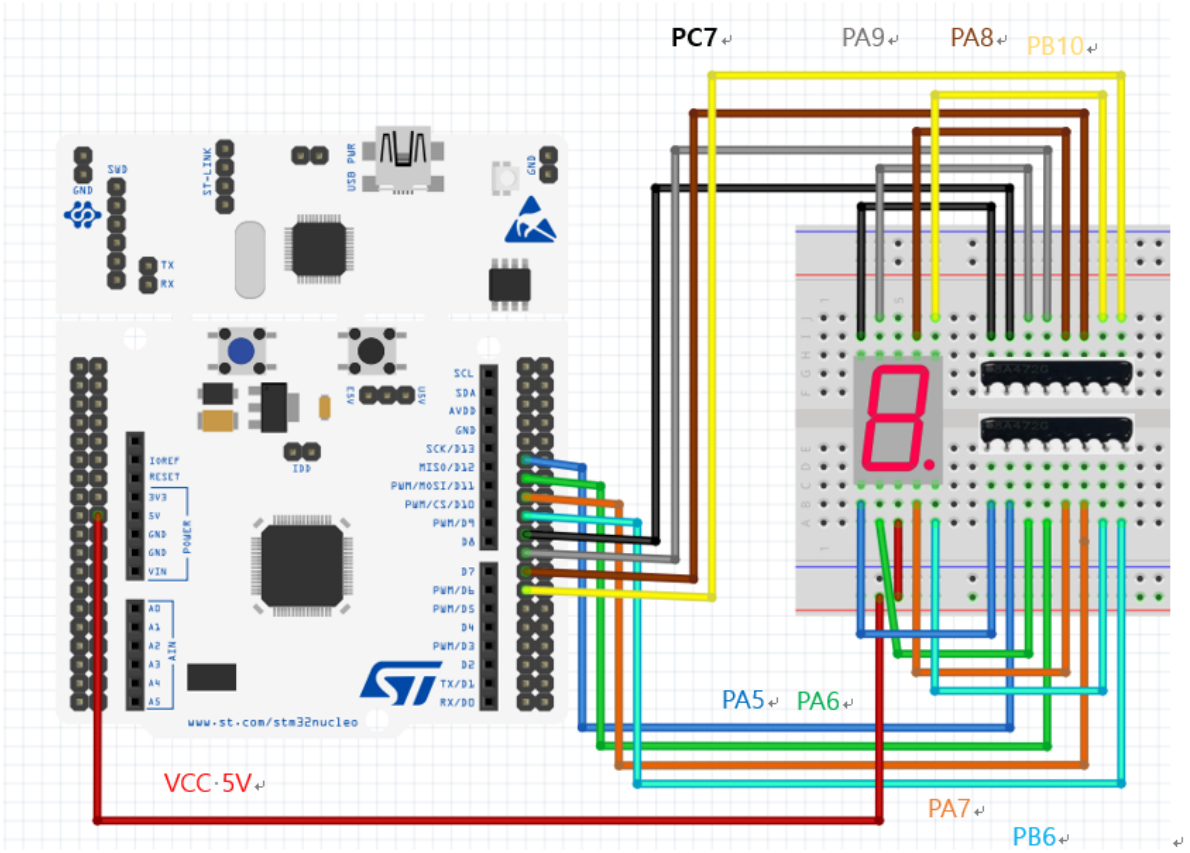
Apply VCC and GND to the 7-segment display.

Apply 'H' to any 7-segment pin 'a'~'g' and observe if that LED is turned on or off

- example: Set 'H' on PA5 of MCU and connect to 'a' of the 7-segment.

Connection Diagram

Circuit diagram



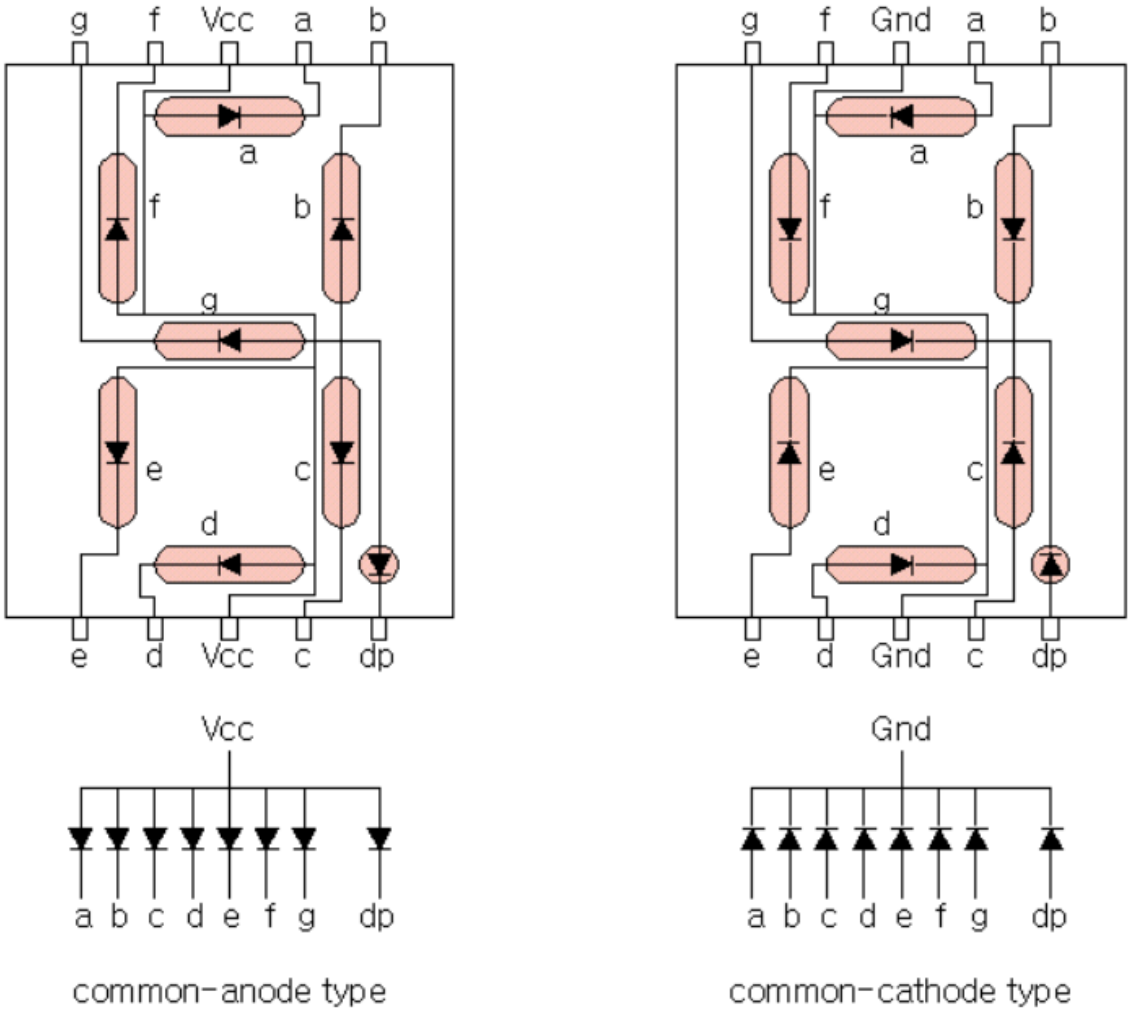
Discussion

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

TRUTH TABLE(common anode)

DECIMAL	Inputs(Binary)				Outputs							
	D	C	B	A	a	b	c	d	e	f	g	dp
0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	1	0	0	1	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0	1
3	0	0	1	1	0	0	0	0	1	1	0	1
4	0	1	0	0	1	0	0	1	1	0	0	1
5	0	1	0	1	0	1	0	0	1	0	0	1
6	0	1	1	0	0	1	0	0	0	0	0	1
7	0	1	1	1	0	0	0	1	1	1	1	1
8	1	0	0	0	0	0	0	1	1	0	0	1
9	1	0	0	1	0	0	0	1	1	0	0	1
10	1	0	1	0	X	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X	X

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



The Common-Anode Type is a seven-segment segment in which the Anode of the internal LED is connected to the Common Pin and each of the eight pins of the Cathode is connected.The Common-Cathode Type is a seven-segment segment in which the cathode of the internal LED is connected to the Common Pin and each of the eight pins of the Anode.

In the Common-Anode type, when the VCC is connected to the Common Pin and the GND is connected to each pin(LOW), the LED is turned on. In the Common-Cathode type, the LED is turned on when the GND is connected to the Common Pin and the VCC(HIGH) is connected to each pin.

1. 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, the LED pin turns on when "LOW" is given to the LED pin from the MCU.

Problem 2: Display 0~9 with button press

Procedure

1. Create a new project under the directory `\repos\EC\LAB\LAB_GPIO_7segment`

- The project name is "**LAB_GPIO_7segment".**
- Create a new source file named as "**LAB_GPIO_7segment.c"**
- Refer to the sample code

> You MUST write your name on the source file inside the comment section.

2. Include your updated library in `\repos\EC\lib\` to your project.

- **ecGPIO.h, ecGPIO.c**
- **ecRCC.h, ecRCC.c**

1. Declare and Define the following functions in your library
    - You can refer to an example code of 7-segment control

ecGPIO.h

```
void sevensegment_init(void);

void sevensegment_decoder(uint8_t  num);
```

1. First, check if every number, 0 to 9, can be displayed properly
2. Then, create a code to display the number from 0 to 9 with each button press. After the number '9', it should start from '0' again.

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

Exercise

Fill in the table

Port/Pin	Description	Register setting
Port A Pin 5	Clear Pin5 mode	GPIOA->MODER &=~(3<<(5*2))
Port A Pin 5	Set Pin5 mode = Output	GPIOA->MODER |=(1<<(5*2))
Port A Pin 6	Clear Pin6 mode	GPIOA->MODER &=~(3<<(6*2))
Port A Pin 6	Set Pin6 mode = Output	GPIOA->MODER |=(1<<(6*2))
Port A Pin Y	Clear PinY mode	GPIOA->MODER &=~(3<<(Y*2))
Port A Pin Y	Set PinY mode = Output	GPIOA->MODER |=(1<<(Y*2))
Port A Pin 5~9	Clear Pin5~9 mode	GPIOA->MODER &=~(1023<<(5*2))
	Set Pin5~9 mode = Output	GPIOA->MODER |=(341<<(5*2))
Port X Pin Y	Clear Pin Y mode	GPIOX->MODER &=~(3<<(Y*2))
	Set Pin Y mode = Output	GPIOX->MODER &=~(3<<(Y*2))
Port A Pin 5	Set Pin5 otype=push-pull	GPIOA->OTYPER &=~(1<<5)
Port A Pin Y	Set PinY otype=push-pull	GPIOA->OTYPER &=~(1<<Y)
Port A Pin 5	Set Pin5 ospeed=Fast	GPIOA->OSPEEDR &=~(3<<(5*2)) GPIOA->OSPEEDR |=(2<<(5*2))
Port A Pin Y	Set PinY ospeed=Fast	GPIOA->OSPEEDR &=~(3<<(Y*2)) GPIOA->OSPEEDR |=(2<<(Y*2))
Port A Pin 5	SetPin5 PUPDR=no pullup/down	GPIOA->PUPDR &=~(3<<(5*2))
Port A Pin Y	SetPinY PUPDR=no pullup/down	GPIOA->PUPDR &=~(3<<(Y*2))

Code

Your code goes here: https://github.com/Ohjeahyun1/EC-jeahyun-447/blob/faf7c19083db5d3518ca44300b2ce6d1a63803f6/lab/LAB_GPIO_7segment.c
https://github.com/Ohjeahyun1/EC-jeahyun-447/blob/f6c41c20631975c92a8c2e39104d73bf4f34d6df/include/ecGPIO.c

Explain your source code with necessary comments.

Description with Code

- Description 1

```
// When button pressed Output 7segment(0~9) in order
void sevenseg_decode(int number){
    int seven[11][8] ={
        //row- number , col - a,b,c,d....DP
                   //a,b,c,d,e,f,g,DP
                    {0,0,0,0,0,0,1,1},          //zero
                    {1,0,0,1,1,1,1,1},          //one
                    {0,0,1,0,0,1,0,1},          //two
                    {0,0,0,0,1,1,0,1},          //three
                    {1,0,0,1,1,0,0,1},          //four
                    {0,1,0,0,1,0,0,1},          //five
                    {0,1,0,0,0,0,0,1},          //six
                    {0,0,0,1,1,1,1,1},          //seven
                    {0,0,0,0,0,0,0,1},          //eight
                    {0,0,0,1,1,0,0,1},          //nine
                    {1,1,1,1,1,1,1,0}           //dot
  };
    //outputs 7segment LEDs
    GPIO_write(GPIOA,8,seven[number][0]);        // a
```

```
    GPIO_write(GPIOB,10,seven[number][1]);       // b
    GPIO_write(GPIOA,7,seven[number][2]);        // c
    GPIO_write(GPIOA,6,seven[number][3]);        // d
    GPIO_write(GPIOA,5,seven[number][4]);        // e
    GPIO_write(GPIOA,9,seven[number][5]);        // f
    GPIO_write(GPIOC,7,seven[number][6]);        // g
    GPIO_write(GPIOB,6,seven[number][7]);        // dp

}
```

- Description 2

```
// 7segment init,otype,pupdr,ospeed
void sevenseg_init(void){
    GPIO_init(GPIOA, 8, OUTPUT);      //a
    GPIO_init(GPIOB, 10,OUTPUT);      //b
    GPIO_init(GPIOA, 7, OUTPUT);      //c
    GPIO_init(GPIOA, 6, OUTPUT);      //d
    GPIO_init(GPIOA, 5, OUTPUT);      //e
    GPIO_init(GPIOA, 9, OUTPUT);      //f
    GPIO_init(GPIOC, 7, OUTPUT);      //g
    GPIO_init(GPIOB, 6, OUTPUT);      //DP

    GPIO_pupdr(GPIOA, 8, EC_NOPUPD);  // GPIOA 5 pupdr -> NO pull up Pull down
    GPIO_pupdr(GPIOB, 10,EC_NOPUPD);  // GPIOB 10 pupdr -> NO pull up Pull down
    GPIO_pupdr(GPIOA, 7, EC_NOPUPD);  // GPIOA 7 pupdr -> NO pull up Pull down
    GPIO_pupdr(GPIOA, 6, EC_NOPUPD);  // GPIOA 6 pupdr -> NO pull up Pull down
    GPIO_pupdr(GPIOA, 5, EC_NOPUPD);  // GPIOA 5 pupdr -> NO pull up Pull down
    GPIO_pupdr(GPIOA, 9, EC_NOPUPD);  // GPIOA 9 pupdr -> NO pull up Pull down
    GPIO_pupdr(GPIOC, 7, EC_NOPUPD);  // GPIOC 7 pupdr -> NO pull up Pull down
    GPIO_pupdr(GPIOB, 6, EC_NOPUPD);  // GPIOB 6 pupdr -> NO pull up Pull down

    GPIO_otype(GPIOA, 8, PP);         // GPIOA 5 otype -> push pull
    GPIO_otype(GPIOB, 10,PP);         // GPIOB 10 otype -> push pull
    GPIO_otype(GPIOA, 7, PP);         // GPIOA 7 otype -> push pull
    GPIO_otype(GPIOA, 6, PP);         // GPIOA 6 otype -> push pull
    GPIO_otype(GPIOA, 5, PP);         // GPIOA 5 otype -> push pull
    GPIO_otype(GPIOA, 9, PP);         // GPIOA 9 otype -> push pull
    GPIO_otype(GPIOC, 7, PP);         // GPIOC 7 otype -> push pull
    GPIO_otype(GPIOB, 6, PP);         // GPIOB 6 otype -> push pull

    GPIO_ospeed(GPIOA,8,SMED);        // GPIOA 5 ospeed -> Medium speed
    GPIO_ospeed(GPIOB,10,SMED);       // GPIOB 10 ospeed -> Medium speed
    GPIO_ospeed(GPIOA,7,SMED);        // GPIOA 7 ospeed -> Medium speed
    GPIO_ospeed(GPIOA,6,SMED);        // GPIOA 6 ospeed -> Medium speed
    GPIO_ospeed(GPIOA,5,SMED);        // GPIOA 5 ospeed -> Medium speed
    GPIO_ospeed(GPIOA,9,SMED);        // GPIOA 9 ospeed -> Medium speed
    GPIO_ospeed(GPIOC,7,SMED);        // GPIOC 7 ospeed -> Medium speed
    GPIO_ospeed(GPIOB,6,SMED);        // GPIOB 6 ospeed -> Medium speed

}
```

- Description 3

```
int main(void) {
    // Initialiization --------------------------------------------------------
```

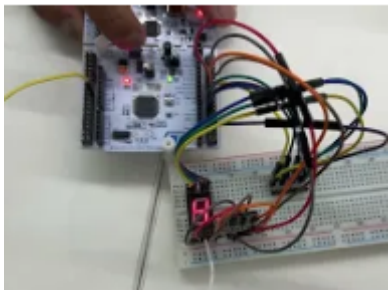
```
    setup();
    unsigned int cnt = 0;
    // Inifinite Loop ----------------------------------------------------------
    while(1){
        //sevensegment output
        sevenseg_decode(cnt % 10);                      //not to make over 10
        if(GPIO_read(GPIOC, BUTTON_PIN) == 0) cnt++;    //if button pressed
7segment output up(0~9)
        if (cnt > 9) cnt = 0;                           //over 10 -> 0
        delay_ms(50);                                   //delay for debouncing
    }
}


// Initialiization
void setup(void)
{
    RCC_HSI_init();
    SysTick_init();
    GPIO_init(GPIOC, BUTTON_PIN, INPUT);  // calls RCC_GPIOC_enable() and button
pin mode -> input
    GPIO_pupdr(GPIOC, BUTTON_PIN, EC_PU); // GPIOC button pin pupdr -> pull up
    sevenseg_init();                        // 7segment init,otype,ospeed,pupdr

}
```

Results

Experiment images and results



When the button is pressed, the output of the 7segment LED changes from 0 to 9.

Demo Video

Add demo video link:https://youtu.be/C21kMEDQyEM

Reference

Complete list of all references used (github, blog, paper, etc)

https://dokkodai.tistory.com/89

Troubleshooting

(Option) You can write Troubleshooting section