

Busan Software Meister High School

# *MICROPROCESSOR*

2309 양유빈

20230413

마이크로프로세서

손정웅선생님

# *OVERVIEW*

- 풀업 및 풀다운, LED와 저항
- 스위치
- 레지스터 직접 제어
- 내장 버튼으로 내장 led 제어
- LED 4개 연속으로 점멸

# 풀업 및 풀다운, LED와 저항

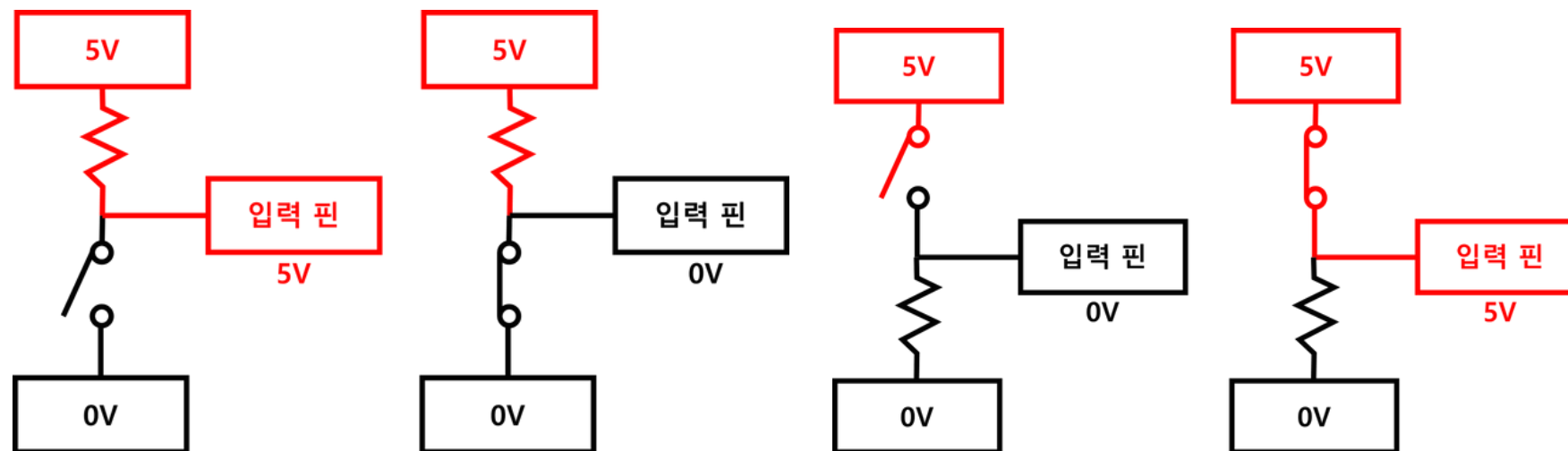
Pull-up and pull-down, LEDs and resistors

- 플로팅: 입력값이 어떤 상태인지(0, 1 등) 알 수 없는 상태.  
가장 위험한 상태.
- 풀업 저항은 10k옴, 풀다운 저항은 4.7k옴 사용.
- 풀업 사용 多.

- 저항: 전기의 흐름(전류) 제어

- LED 저항값 계산 (표) (+LED 연결 저항 값 계산):

색상	구 분	최소전압	최대전압	전류(일반)	전류(최대)
적●	Red	1.8V	2.3V	20 mA	50 mA
등●	Orange	2.0V	2.3V	30 mA	50 mA
황●	Real Yellow	2.0V	2.8V	20 mA	50 mA
초●	emerald Green	1.8V	2.3V	20 mA	50 mA
초●	Real Green	3.0V	3.6V	20 mA	50 mA
청●	sky Blue	3.4V	3.8V	20 mA	50 mA
청●	Real Blue	3.4V	3.8V	20 mA	50 mA
자●	Pink	3.4V	3.8V	20 mA	50 mA
백○	White	3.4V	4.0V	20 mA	50 mA



# 스위치

switch

초기 접점 상태에 따른 분류:

- NO : Normally Open 단자, 평소에는 연결되어 있지 않는(open) 상태이다가 스위치가 동작하면 연결(Close) 됨.
- NC : Normally Close 단자, 평소에는 연결되어있다가(Close) 스위치가 동작하면 끊어짐(open).

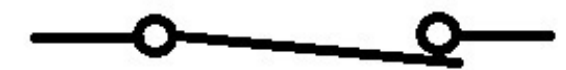
**a** 접점

NO = Normally Open



**b** 접점

NC = Normally Close

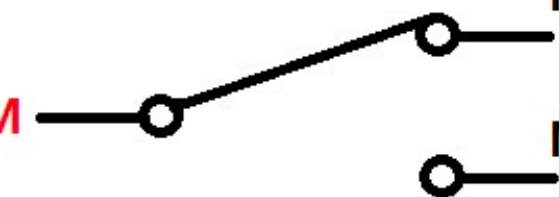


**c** 접점

COM

NC

NO



# 내장 버튼으로 내장 led 제어

Built-in LED control with built-in buttons

// main.c에서 코드 수정

```
1 // 내장 버튼으로 LED를 제어할 수 있도록 프로그래밍 하시오.
2 #include "main.h"
3
4 int main(void) {
5     RCC->APB2ENR |= (0x1<<2) | (0x1<<4);
6     GPIOA->CRL &= ~(1<<22); // ~(0x1<<22);
7     GPIOA->CRL |= (1<<20); // (0x1<<20); output select
8     GPIOC->CRH &= ~(1<<22); // ~(0x1<<22);
9     GPIOC->CRH |= (1<<23); // (0x1<<23); input select
10    GPIOC->ODR |= (1<<13); // (0x1<<13);
11
12    while(1) {
13        if(GPIOC->IDR & 0x00002000) {
14            GPIOA->BSRR |= (0x1<<21); //0x00200000; //LED reset
15        } else {
16            GPIOA->BSRR |= (0x1<<5); //0x00000020; //LED set
17        }
18    }
19 }
```

# LED 4개 연속으로 점멸

4 LEDs blink in succession

```
1 // LED 4개를 연속하여 점멸하도록 프로그래밍 하시오.
2 #include "main.h"
3
4 void Delay_Timer(uint32_t time){
5     for(; time > 0; time--){
6     }
7 }
8 void set_led(uint8_t n){
9     switch(n){
10     case 1:
11
12         GPIOA->BSRR |= 0x01<<12;
13         Delay_Timer(500000);
14         GPIOA->BSRR |= 0x01<<28;
15         Delay_Timer(500000);
16         break;
17     case 2:
18         GPIOA->BSRR |= 0x01<<11;
19         Delay_Timer(500000);
20         GPIOA->BSRR |= 0x01<<27;
21         Delay_Timer(500000);
22         break;
23     case 3:
24         GPIOB->BSRR |= 0x01<<12;
25         Delay_Timer(500000);
26         GPIOB->BSRR |= 0x01<<28;
27         Delay_Timer(500000);
28         break;
29     case 4:
30         GPIOB->BSRR |= 0x01<<11;
31         Delay_Timer(500000);
32         GPIOB->BSRR |= 0x01<<27;
33         Delay_Timer(500000);
34         break;
35     }
36 }
37
```

```
38 int main(void) {
39     RCC->APB2ENR |= (0x1<<2)|(0x1<<3);
40     GPIOA->CRH &= ~0x44000;
41     GPIOA->CRH |= 0x11000;
42     GPIOC->CRH &= ~0x44000;
43     GPIOC->CRH |= 0x11000;
44
45     while(1) {
46         for(uint8_t i=1;i<5;i++){
47             set_led(i);
48         }
49     }
50 }
```