

# 스위치에 따른 LED 밝기 변화

20170967 윤수름

# 목차

- 사용하는 기능
- 기대했던 점
- 만들려고 했던것
- 회로도
- 실행결과
- 코드설명
- 추가한 부분

# 사용하는기능

- PWM
- SW 1,2,3
- LED
- 7-segment

# 만들려고 했던 것

- pwm의 출력을 10가지로 나누고, LED에 연결하여
- 7segment또한 0~9까지의 숫자를 표현하게 한다음
- sw1을 누르면 7segment의 숫자가 증가
- sw2을 누르면 7segment의 숫자가 감소하고,
- 그 숫자에 따라 LED의 밝기를 조절하려고 함.



= 0일때 LED의  
밝기가 0



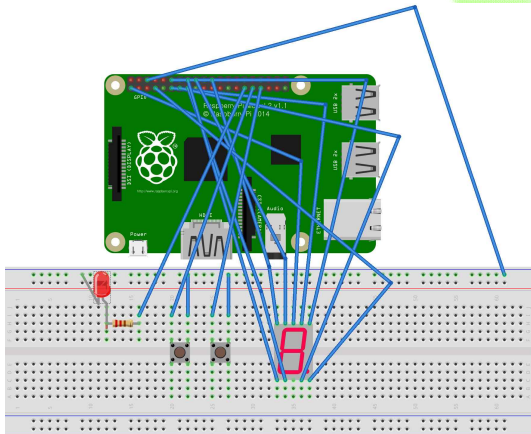
숫자가 증가함에 따라 LED의  
밝기가 순차적으로 증가

# 기대했던점

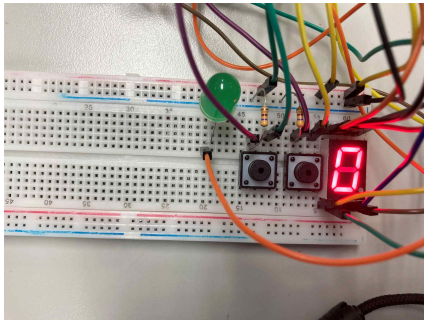
- 실생활에서 흔히 쓰는 실내등 같은 경우 밝기조절이 없음
- ->
- 밝기조절이 되는 전등을 만들어
- 무드를 원할 땐 밝기를 약하게
- 밝은 환경을 원할 땐 밝기를 강하게 만들려고 함

# 회로도

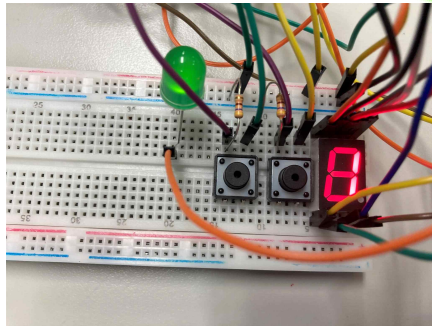
- 각 GPIO 0~7번은
- 7-segment의 A~G,DP에 연결함
- SW1,SW2는 각각 22,23에 연결함
- LED는 GPIO26번에 연결하여
- GPIO26번을 소프트웨어적으로
- PWM으로 사용함



## 실행결과

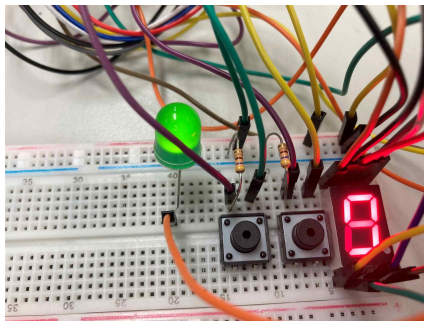
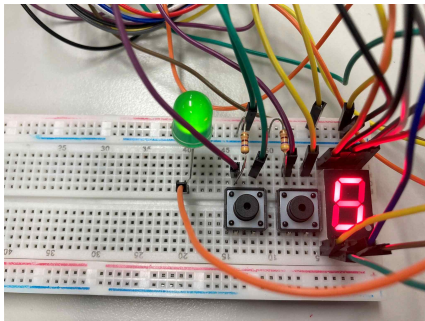


segment가 0일때 LED가 꺼짐



segment가 1일때 LED가 켜짐

# 실행결과



segment가 5일때 LED와  
9일때의 LED 밝기 비교



# 코드

```
37
38
39 pinMode(SW1, INPUT);
40 pinMode(SW2, INPUT);
41 pinMode(SEG_A, OUTPUT);
42 pinMode(SEG_B, OUTPUT);
43 pinMode(SEG_C, OUTPUT);
44 pinMode(SEG_D, OUTPUT);
45 pinMode(SEG_E, OUTPUT);
46 pinMode(SEG_F, OUTPUT);
47 pinMode(SEG_G, OUTPUT);
48 pinMode(SEG_DP, OUTPUT);
49 pinMode(LED, PWM_OUTPUT);
50 pwmSetMode(PWM_MODE_MS);
51 pwmSetRange(100);
52
```

C:\Users\Gyul\AppData\Roaming\MobaXterm\slas

pwmsetRange(100)

->pwm의 최대치로 100으로 정해둠  
pwmSetMode=>마크스페이스모드  
로 pwm출력

```
while(1)
{
    if(digitalRead(SW1) == HIGH)
    {
        num++;
        if(num > 9)
            num = 0;

        for(i=0; i<8; i++)
        {
            digitalWrite(segPins[i], sevenseg[num][i]);
        }

        if(num == 0)
        {
            pwmWrite(LED, 0);
        }
        else
        {
            pwmWrite(LED, (num + 1) * 10);
        }
        printf("NUM = %d\n", num);
        printf("LED = %d\n", (num + 1) * 10);

        delay(200);
    }
}
```

sw1이 입력될 때,  
num이 1씩 증가(num은 segment숫자)  
num이 9를 초과하면, 0으로 돌아감.  
만약 num=0이면 LED를 끄.  
LED밝기는 num+1 \* 10을 하여  
각 segment 마다 밝기를 10씩 차이나게 설계함.

# 추가한 부분

```
if(digitalRead(SW3) == HIGH)
{
    if(isOn)
    {
        isOn = 0;
        prevNum = num;
        prevIsOn = 1;
        pwmWrite(LED, 0);
        for(i=0; i<8; i++)
        {
            digitalWrite(segPins[i], HIGH);
        }
        printf("LED and segment lights turned off\n");
    }
    else
    {
        isOn = 1;
        num = prevNum;
        prevIsOn = 0;
        if(num == 0)
        {
            pwmWrite(LED, 0);
        }
        else
        {
            pwmWrite(LED, (num + 1) * 10);
        }
        for(i=0; i<8; i++)
        {
            digitalWrite(segPins[i], sevenseg[num][i]);
        }
        printf("LED and segment lights turned on\n");
    }
    delay(200);
}
```

isOn 변수를 사용하여 LED 및 segment의 상태를 추적함.

prevNum에는 이전 숫자값 즉, num을 저장하고

prevIsOn에는 LED값을 저장.

SW3 버튼을 누르면 isOn 상태를 변경하여, 이전 상태에 따라 LED 및 segment의 전원을 끄고 켤 수 있도록 수정함.



감사합니다.