

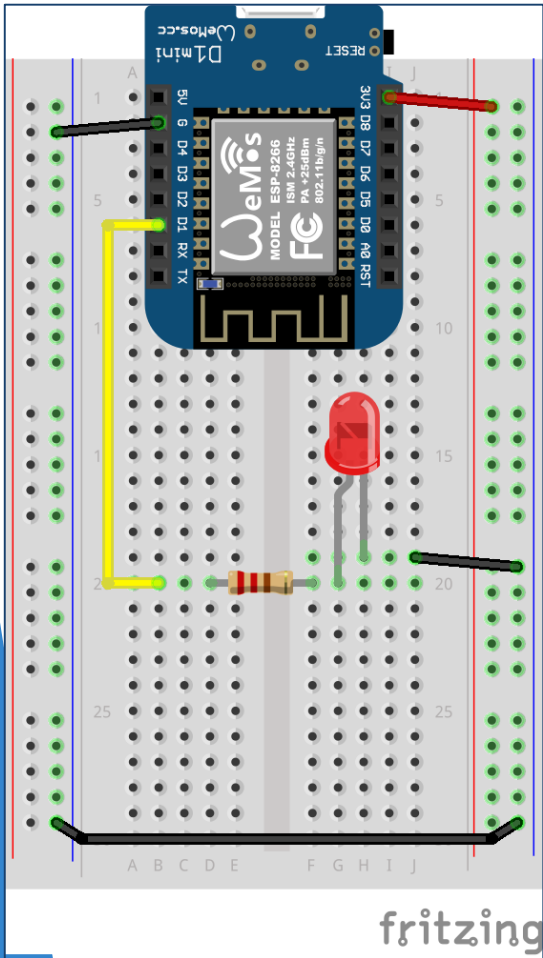
멜로디응용

박 정욱

배울 내용 (Contents)

- 조명 및 버튼과 LED 응용 내용 복습
 - LED 및 Switch를 사용한 신호등 회로 구성 가능
 - 본인의 목적에 맞는 프로그램 작성 가능
- LED & Switch를 사용한 모형 신호등 제작 - 복습
- 코드와 회로도를 응용한 조명 컨트롤 - 복습
- 멜로디 내용 복습과 프로젝트 진행

LED 기초 복습 - 단일 LED 테스트



- 단일 LED (=1개의 LED를 사용한 실험) 복습
 - 키트 내부의 테스트 LED를 사용.
- 회로 구성
 - VCC = 3.3V
 - D1
 - GND = Ground(접지)
- D1 핀에 데이터를 전달 할 때만 LED가 켜지도록 만들기.

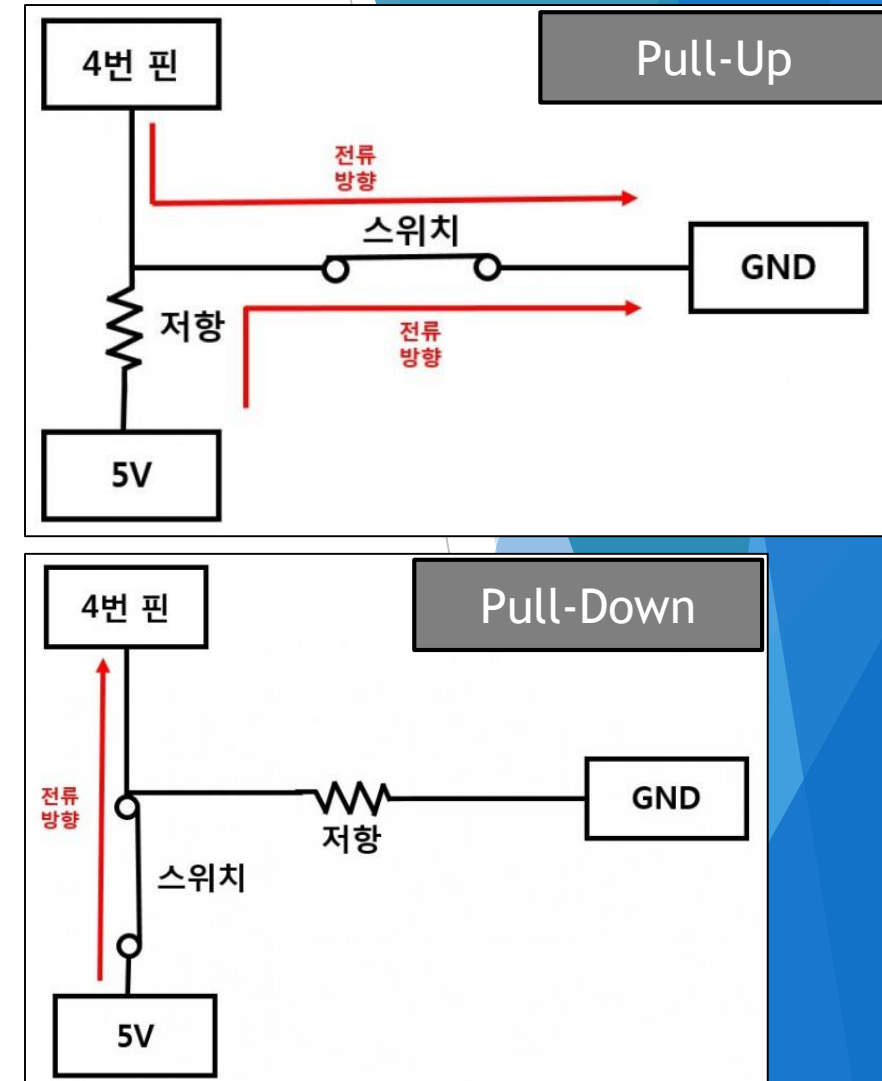
LED 기초 복습 - 단일 LED 테스트

■ Source

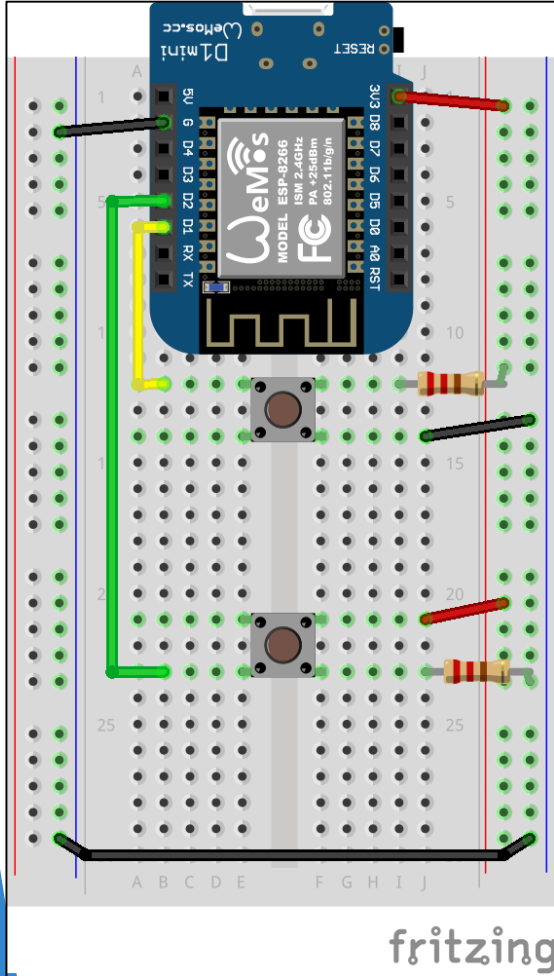
```
22_Lab-01
1  const int PIN_LED = D1;
2
3  // 프로그램 시작시 초기화 작업
4  void setup()
5  {
6      Serial.begin(115200);    // 시리얼 통신 초기화
7      pinMode(PIN_LED, OUTPUT);
8  }
9
10 void loop() {
11     digitalWrite(D1, HIGH);
12     delay(1000);
13     digitalWrite(D1, LOW);
14     delay(1000);
15 }
```

Switch(버튼 복습)

- 버튼은 기본적으로 Floating(플로팅) 상태에서 각각 회로 구성에 따라 Pull-Up과 Pull-Down 버튼으로 사용한다.
 - Pull-Up: 왼쪽 상단의 그림처럼 저항을 VCC에 연결한 상태
 - Pull-Down: 왼쪽 하단의 그림처럼 저항을 GND에 연결한 상태
- 기본적으로
Pull-Up은 눌렀을 때 0(Low), 가만히 있을 때 1(High)를 출력하며,
Pull-Down은 눌렀을 때 1(High), 가만히 있을 때 0(Low)를 출력 한다.



Switch 기초 복습



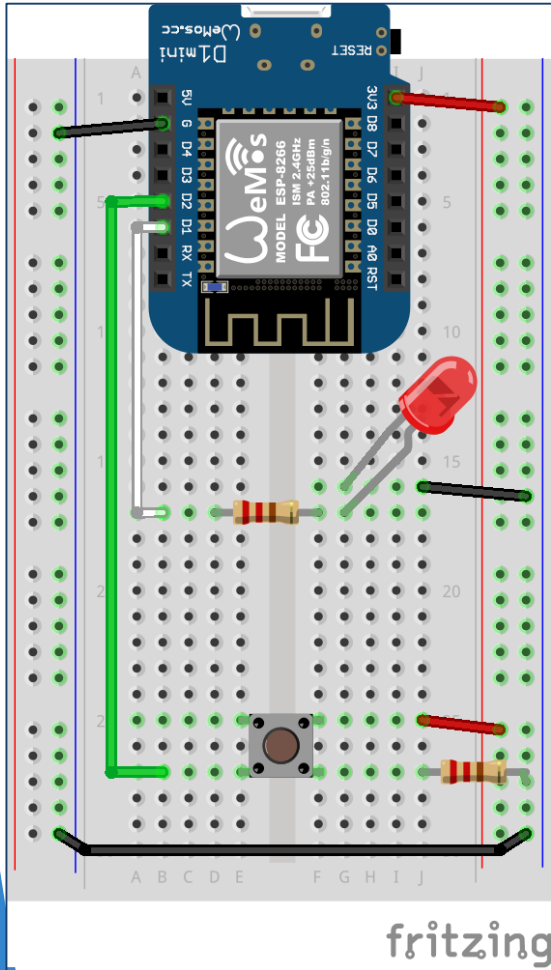
- 각각 Pull-Up(위), Pull-Down(아래) 회로를 둘다 구성하고 스위치 동작의 차이점을 구분한다.
- 회로 구성
 - VCC = 3.3v
 - 디지털핀 = D1, D2
 - GND = GND
- 소스를 코딩하고 동작 상태를 확인.

Switch 기초 복습

■ Source

```
22_Lab-02 $
1 const int btn1 = D1;
2 const int btn2 = D2;
3
4 void setup() {
5     Serial.begin(115200);
6
7     pinMode(btn1, INPUT);
8     pinMode(btn2, INPUT);
9 }
10
11 void loop() {
12     Serial.print("Button1 : ");
13     Serial.println(digitalRead(btn1));
14     Serial.print("Button2 : ");
15     Serial.println(digitalRead(btn2));
16     Serial.println("-----");
17     delay(500);
18 }
```

LED & Switch 연동 기초 - 복습



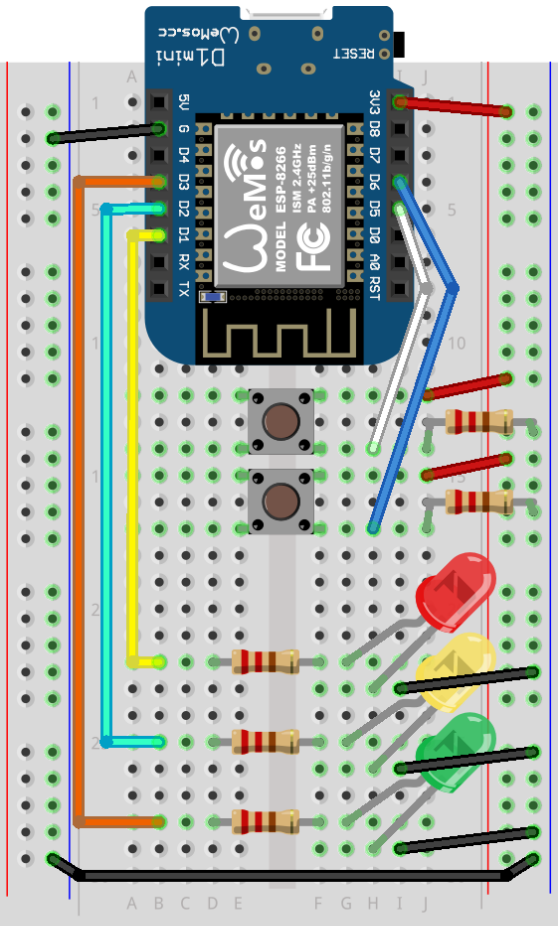
- Switch를 Pull-Down 회로로 구성 하고 LED 한 개를 사용하여 정상적인 동작을 확인 한다.
- 회로 구성
 - VCC = 3.3V
 - LED Pin = D1
 - Switch Pin = D2
 - GND = G

LED & Switch 연동 기초 - 복습

■ Source

```
22_Lab-03
1  const int button = D2;
2  const int LED = D1;
3
4  int buttonState = 0;
5
6  void setup() {
7      Serial.begin(115200);
8
9      pinMode(button, INPUT);
10     pinMode(LED, OUTPUT);
11 }
12
13 void loop() {
14     buttonState = digitalRead(button);
15
16     if (buttonState == HIGH) { // turn LED on:
17         digitalWrite(LED, HIGH);
18         Serial.print("HIGH");
19         Serial.println();
20     } else { // turn LED off:
21         digitalWrite(LED, LOW);
22         Serial.print("LOW");
23         Serial.println();
24     }
25
26     delay(500);
27 }
```

(통합) 신호등 복습



fritzing

■ 회로 구성(Schematic)

- 3개의 LED(각각 빨강, 초록, 노랑)에 D1, D2, D3와 VCC를 연결
- 2개의 버튼에 D5, D6과 VCC, GND를 연결
- 1개의 보행자 LED(추가)를 D0와 VCC에 연결

(통합)

■ Source

```
test | 아두이노 1.8.5
파일 편집 스케치 툴 도움말

test
const int green_led = D1;
const int yellow_led = D2;
const int red_led = D3;

const int pedestrian_led = D0;

const int button_reset = D5;
const int button_stop = D6;

int i;

void setup() {
  Serial.begin(115200);
  pinMode(green_led, OUTPUT);
  pinMode(yellow_led, OUTPUT);
  pinMode(red_led, OUTPUT);

  pinMode(pedestrian_led, OUTPUT);

  pinMode(button_stop, INPUT);
  pinMode(button_reset, INPUT);

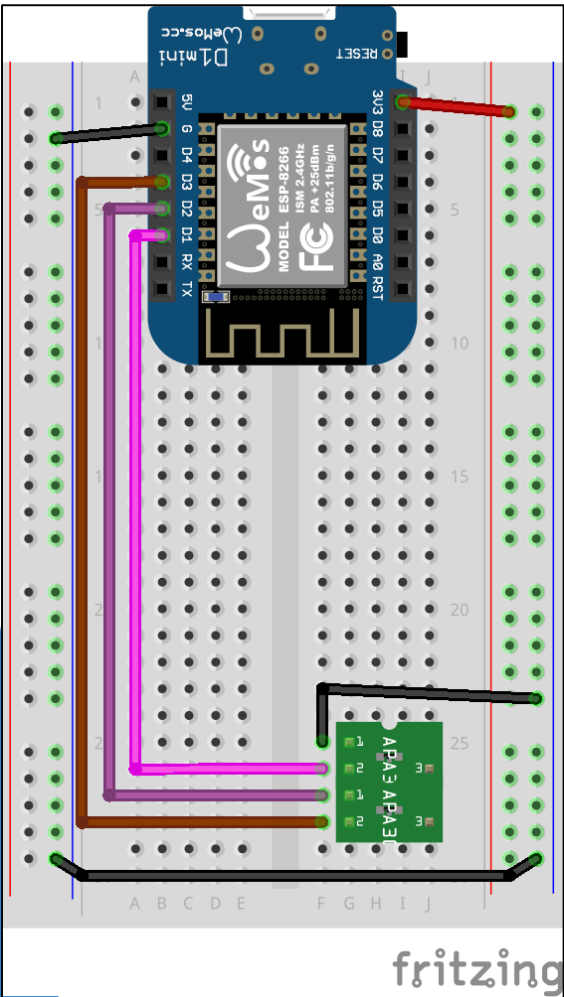
  i = 0;
}

void loop() {
  i++;
  if( digitalRead(button_reset) == 1 ){
    Serial.print("Button reset : ");
    Serial.println(digitalRead(button_reset));
    i = 0;
  } else{
    Serial.print("Button reset : ");
    Serial.println(digitalRead(button_reset));
  }
}
```

```
if( digitalRead(button_stop) == 0 ){
  Serial.print("Button stop : ");
  Serial.println(digitalRead(button_stop));
  |
  digitalWrite(red_led, HIGH);
  digitalWrite(green_led, LOW);
  digitalWrite(yellow_led, LOW);
  digitalWrite(pedestrian_led, HIGH);
  i = 1;
} else {
  Serial.print("Button stop : ");
  Serial.println(digitalRead(button_stop));
}

if ( i == 1 ) {
  digitalWrite(red_led, HIGH);
  digitalWrite(green_led, LOW);
  digitalWrite(yellow_led, LOW);
  digitalWrite(pedestrian_led, HIGH);
  delay(2000);
} else if ( i == 2 ) {
  digitalWrite(yellow_led, HIGH);
  digitalWrite(green_led, LOW);
  digitalWrite(red_led, LOW);
  digitalWrite(pedestrian_led, LOW);
  delay(2000);
} else if ( i == 3 ) {
  digitalWrite(red_led, LOW);
  digitalWrite(yellow_led, LOW);
  digitalWrite(green_led, HIGH);
  digitalWrite(pedestrian_led, LOW);
  delay(2000);
  i = 0;
}
delay(100);
}
```

시리얼 입출력을 사용한 조명 제어



- 회로 구성
- 키트에 있는 LED Array(3색 LED)를 사용하여 제작
- 첫번째 핀에 GND, 각각 나머지에 D1, D2, D3를 연결
- 시리얼 모니터에 1, 2, 3을 각각 입력하면 각기 다른 조명이 동작하도록 구성.

시리얼 입출

■ Source

```
07
1 const int PIN_LED_BLUE = D1;
2 const int PIN_LED_GREEN = D2;
3 const int PIN_LED_RED = D3;
4
5 int count;
6 int i;
7
8 // 프로그램 시작시 초기화 작업
9 void setup()
10 {
11     Serial.begin(115200);    // 시리얼 통신 초기화
12     pinMode(PIN_LED_BLUE, OUTPUT);
13     pinMode(PIN_LED_GREEN, OUTPUT);
14     pinMode(PIN_LED_RED, OUTPUT);
15
16     count = 0;
17     Serial.println("Start");
18 }
19
20 void loop() {
21     char getChar;
22     if (Serial.available())
23     {
24         getChar = Serial.read();
25         Serial.print("Readed data = ");
26         Serial.println(getChar);
27
28         if ( getChar == '1' ) {
29             for (i = 1; i <= 10; i++) {
30                 digitalWrite(PIN_LED_BLUE, HIGH);
31                 delay(100);
32                 digitalWrite(PIN_LED_BLUE, LOW);
33                 delay(100);
34             }
35         } else if ( getChar == '2' ) {
36             for (i = 1; i <= 10; i++) {
37                 digitalWrite(PIN_LED_GREEN, HIGH);
```

조명 제어

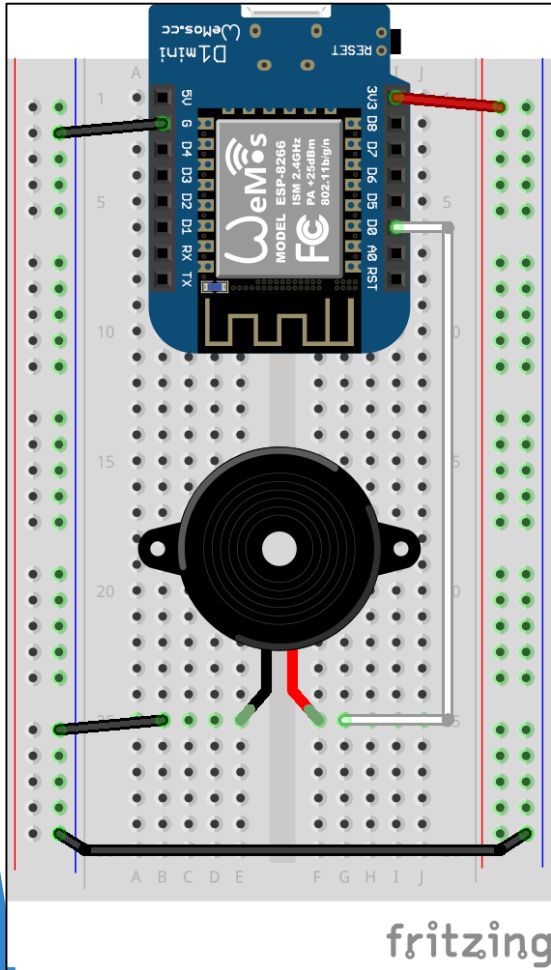
```
38         delay(100);
39         digitalWrite(PIN_LED_GREEN, LOW);
40         delay(100);
41     }
42 } else if ( getChar == '3' ) {
43     for (i = 1; i <= 10; i++) {
44         digitalWrite(PIN_LED_RED, HIGH);
45         delay(100);
46         digitalWrite(PIN_LED_RED, LOW);
47         delay(100);
48     }
49     count = 0;
50 }
51 }
52 }
```

멜로디 기본 내용



- 피에조 부저(스피커)로 간단한 스피커 제작을 진행 한다.
- -뽁 소리(비프음)만 사용 가능하며, 음높이를 사용하여 도~시 까지의 음을 표현 가능 하다.

멜로디 실습 기초



- 스피커 1개를 연결하여 멜로디 실습을 진행
- 회로 구성
 - 스피커의 -극에 GND
 - 스피커의 +극에 D0를 연결

멜로디 실습 기초

■ Source

```
22_Lab-06
1 #define C 262 // 도
2 #define D 294 // 레
3 #define E 330 // 미
4 #define F 349 // 파
5 #define G 392 // 솔
6 #define A 440 // 라
7 #define B 494 // 시
8
9 const int piezoPin = D0; // 부저의 핀
10
11 int tempo = 200; // duration 옵션 값 설정
12 int notes[25] = { G, G, A, A, G, G, E, G, G, E, E, D, G, G, A, A, G, G, E, G, E, D, E, C };
13
14 void setup() {
15     pinMode (piezoPin, OUTPUT);
16 }
17
18 void loop() {
19     for (int i = 0; i < 12; i++) {
20         tone (piezoPin, notes[ i ], tempo);
21         delay (300);
22     }
23     delay(100); // 멜로디 중간에 짧게 멈추는 용도
24
25     for (int i = 12; i < 25; i++) {
26         tone (piezoPin, notes[ i ], tempo);
27         delay(300);
28     }
29 }
```


- Q & A

배운 내용

- 신호등 만들기 복습(자세히)
- 조명 복습(다시)
- 멜로디 기초 실습