

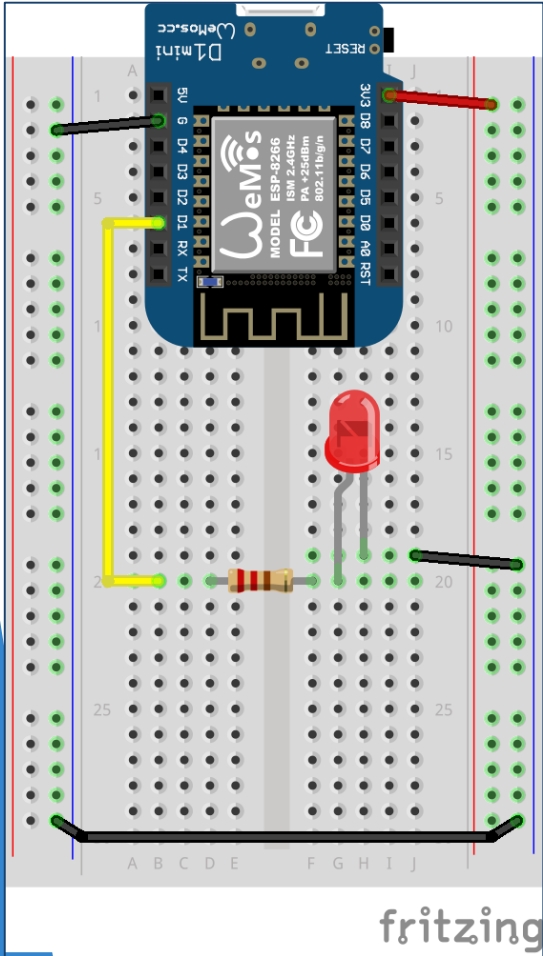
# 조명

박 정욱

# 배울 내용 (Contents)

- 다중 LED를 사용한 회로 작성 및 코딩을 설계.
  - 스스로 LED 회로 구성이 가능하다.
  - 본인의 목적에 맞는 프로그램 소스 작성이 가능하다.
- Switch의 개요와 원리.
- LED와 Switch 기타 부품들을 사용한 모형 신호등 제작 - 복습
- 소스 코드를 사용한 LED(조명) 제어
- 이론 내용 및 프로젝트 내용 이해

# LED 기초 복습 - 단일 LED 테스트



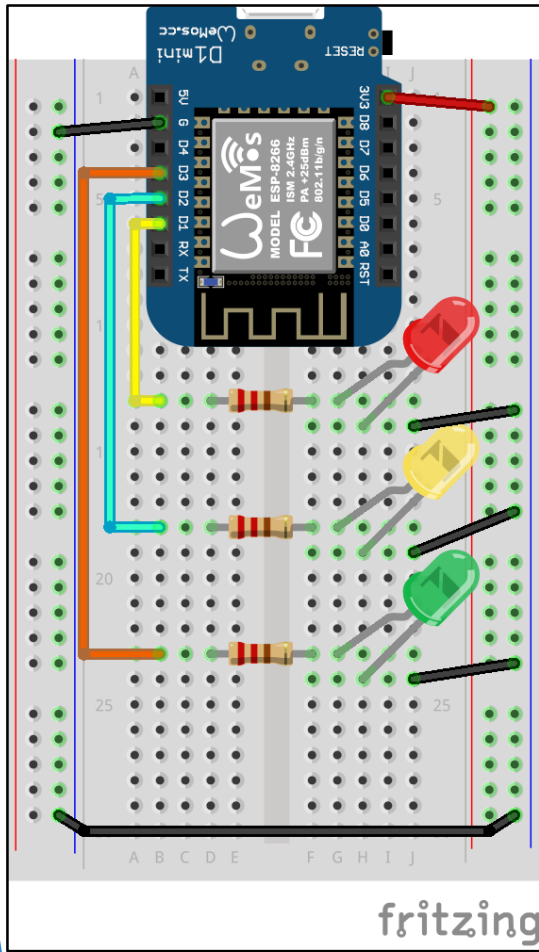
- 단일 LED (=1개의 LED를 사용한 실험) 복습
  - 키트 내부의 테스트 LED를 사용.
- 긴 다리(+)에 저항과 VCC를 연결 하고, 짧은 다리(-)에 GND를 연결.
  - VCC = 3.3V
  - Digital Pin = D1
  - GND = Ground(접지)
- 이전 테스트와 다르게 프로그래밍 코드를 작성하여 D1 핀에 데이터를 전달 할 때만 LED가 켜지도록 만들기.

# LED 기초 복습 - 단일 LED 테스트

## Source

```
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 }
```

# LED 기초 복습 - 다중 LED 테스트



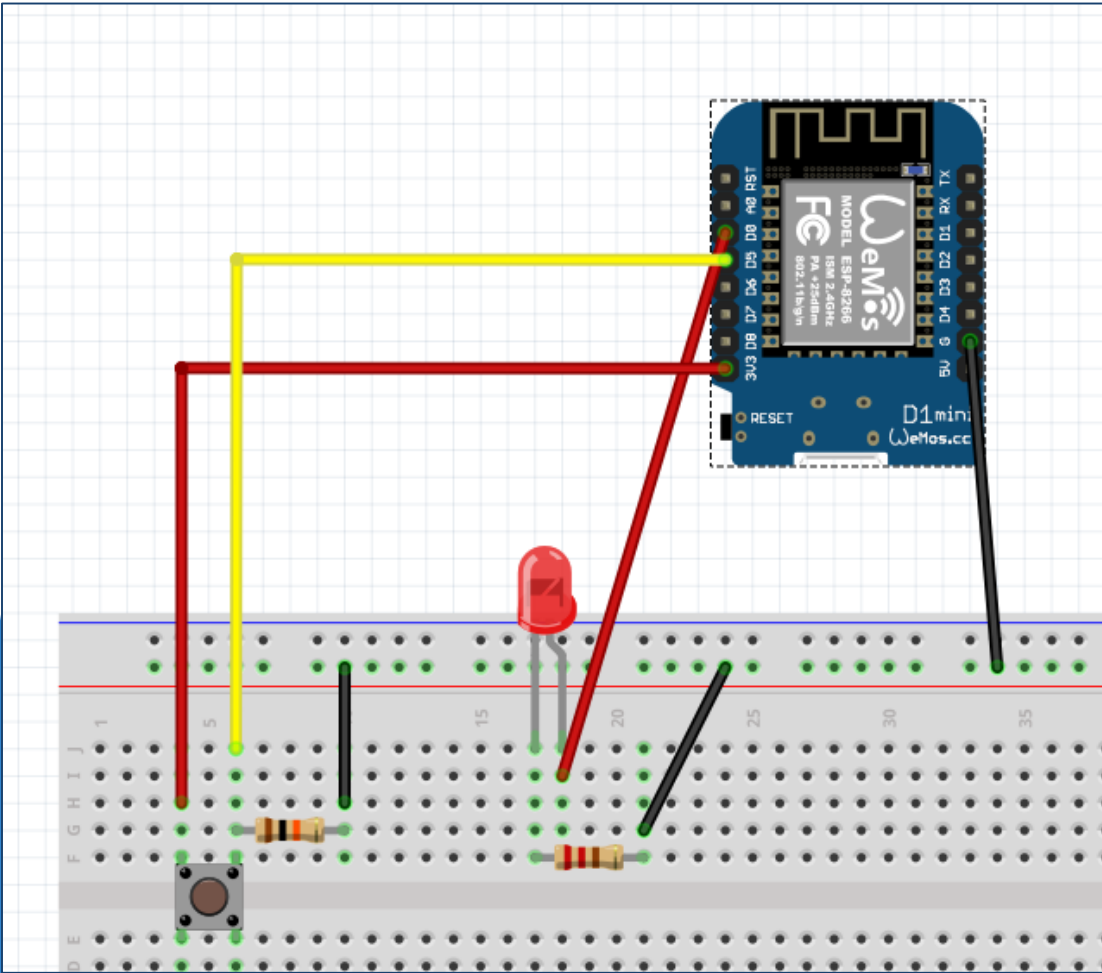
- 회로 구성
  - GND에 각각 LED의 짧은 다리(-)를 연결
  - 각각의 Digital Pin에 각각의 LED의 긴 다리(+)를 연결 (이 때 빨간색 LED에는 220옴의 저항을 달아준다.)
- 단일 LED 테스트에서 한 과정과 마찬가지로 이번에는 프로그래밍을 통한 코딩을 활용하여 LED를 각각 제어 할 수 있도록 한다.

# LED 기초 복습 - 다중

## ■ Source

```
1 const int PIN_LED_GREEN = D1;
2 const int PIN_LED_YELLOW = D2;
3 const int PIN_LED_RED = D3;
4 int i;
5
6 // 프로그램 시작시 초기화 작업
7 void setup()
8 {
9     Serial.begin(115200);    // 시리얼 통신 초기화
10    pinMode(PIN_LED_GREEN, OUTPUT);
11    pinMode(PIN_LED_YELLOW, OUTPUT);
12    pinMode(PIN_LED_RED, OUTPUT);
13
14    i = 0;
15 }
16
17 void loop() {
18     i++;
19     if ( i == 1 ) {
20         digitalWrite(PIN_LED_GREEN, HIGH);
21         digitalWrite(PIN_LED_YELLOW, LOW);
22         digitalWrite(PIN_LED_RED, LOW);
23         delay(2000);
24     } else if ( i == 2 ) {
25         digitalWrite(PIN_LED_GREEN, LOW);
26         digitalWrite(PIN_LED_YELLOW, HIGH);
27         digitalWrite(PIN_LED_RED, LOW);
28         delay(2000);
29     } else if ( i == 3 ) {
30         digitalWrite(PIN_LED_GREEN, LOW);
31         digitalWrite(PIN_LED_YELLOW, LOW);
32         digitalWrite(PIN_LED_RED, HIGH);
33         delay(2000);
34         i = 0;
35     }
36 }
```

# Switch 기초 복습 - 단일 스위치 테스트



- 회로 구성(잘 보이도록)
- 우선 Button을 1개 연결하여 LED의 제어가 가능하도록 구성해 본다.

# Switch 기초 복습 - 다익 스위치 테스트

## ■ Source

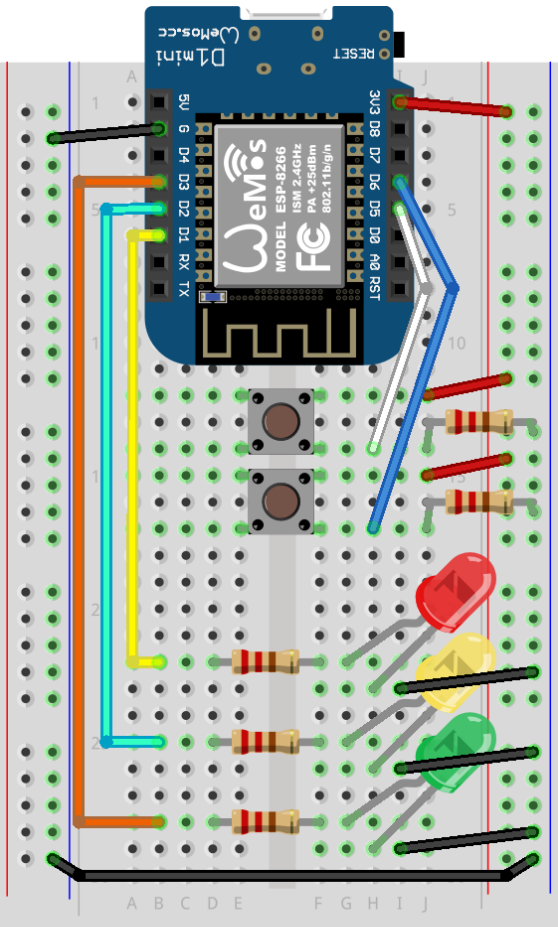
```
03
1 const int D5buttonPin = 14;
2 const int D0ledPin = 16;
3
4 int buttonState = 0;
5
6 void setup() {
7   pinMode(D0ledPin, OUTPUT);
8   pinMode(D5buttonPin, INPUT);
9 }
10
11 void loop() {
12   buttonState = digitalRead(D5buttonPin);
13
14   if (buttonState == HIGH) {
15     // turn LED on:
16     digitalWrite(D0ledPin, HIGH);
17   } else {
18     // turn LED off:
19     digitalWrite(D0ledPin, LOW);
20   }
21 }
22
```



# 모형 신호등 제작 + 심화 (복습)

- 기본에 더해서 긴급 구조 버튼 및 일시 정지 버튼 만들기 - 심화
  1. 신호등 제어에 반드시 필요한 긴급 구조 버튼 : 버튼 클릭 시, 신호등이 적색으로 켜지며, 버튼을 떼기 전까지 적색 신호를 유지
  2. 일시 정지 버튼 : 신호등의 현재 순서에 상관 없이 신호등을 초기화 하고, 현재 신호에서 변경되지 않도록 멈추었다가 다시 시작하는 기능
  3. LED를 한 개 더 추가하여, 보행자 신호 제작 : 보행자 신호는 기존 신호등이 적색 상태 일 때 녹색(OK)를 유지하며, 이후 적색으로 유지된다.

# 모형 신호등 제작 + 심화



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

# 모형 신호등 제작 + 심화

- Code(소스 구성)

```
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));
  }
}
```

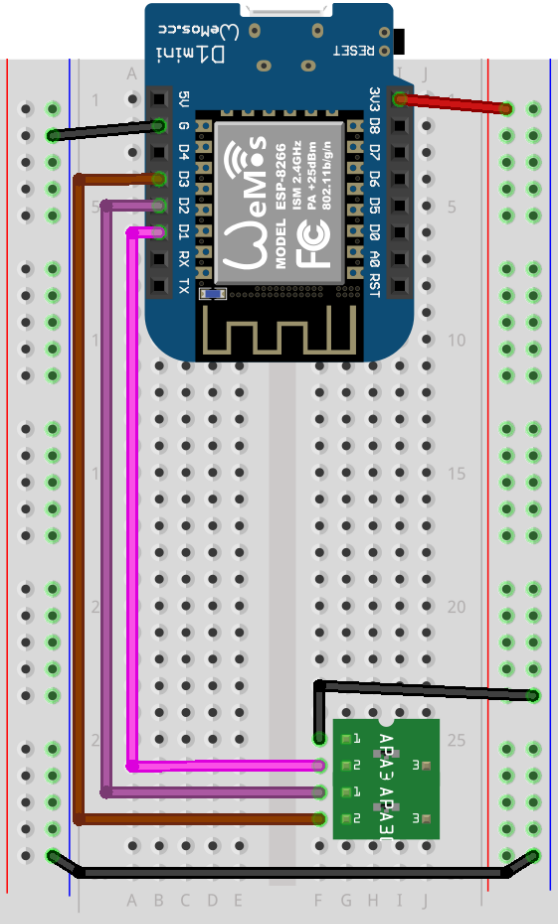
# 모형 신호등 제작 + 심화

- Code(소스 구성) (이어서)

```
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);
}
```

# LED Array를 사용한 기초 실습



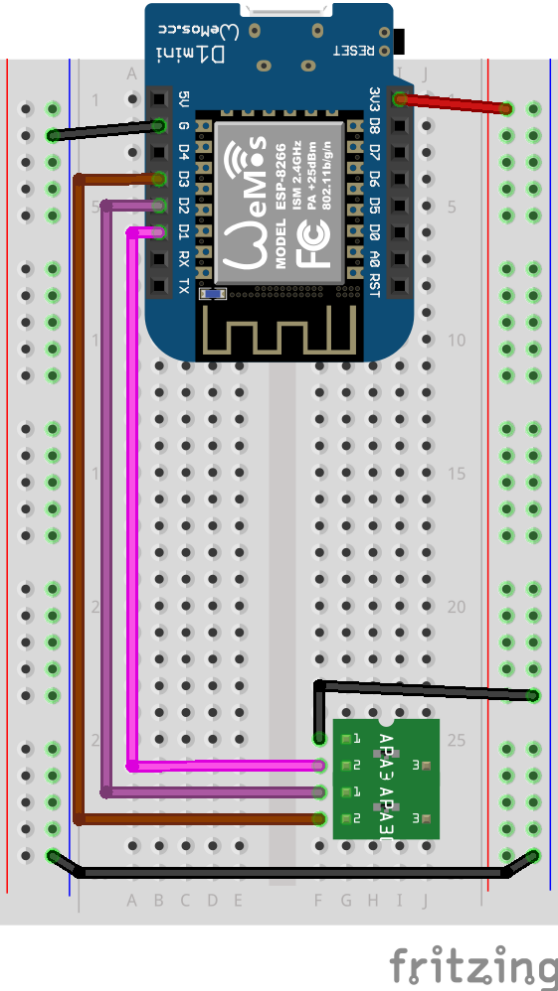
- 회로도
- 키트에 있는 LED Array(3색 LED)를 사용하여 제작
- 첫번째 핀에 GND, 각각 나머지에 D1, D2, D3를 연결

# LED Array를 사용한 :

## ■ Source

```
1 const int PIN_LED_GREEN = D1;
2 const int PIN_LED_YELLOW = D2;
3 const int PIN_LED_RED = D3;
4 int i;
5
6 // 프로그램 시작시 초기화 작업
7 void setup()
8 {
9     Serial.begin(115200);    // 시리얼 통신 초기화
10    pinMode(PIN_LED_GREEN, OUTPUT);
11    pinMode(PIN_LED_YELLOW, OUTPUT);
12    pinMode(PIN_LED_RED, OUTPUT);
13
14    i = 0;
15 }
16
17 void loop() {
18     i++;
19     if ( i == 1 )
20     {
21         digitalWrite(PIN_LED_GREEN, HIGH);
22         digitalWrite(PIN_LED_YELLOW, LOW);
23         digitalWrite(PIN_LED_RED, LOW);
24         delay(2000);
25     } else if ( i == 2 ) {
26         digitalWrite(PIN_LED_GREEN, LOW);
27         digitalWrite(PIN_LED_YELLOW, HIGH);
28         digitalWrite(PIN_LED_RED, LOW);
29         delay(2000);
30     } else if ( i == 3 ) {
31         digitalWrite(PIN_LED_GREEN, LOW);
32         digitalWrite(PIN_LED_YELLOW, LOW);
33         digitalWrite(PIN_LED_RED, HIGH);
34         delay(2000);
35         i = 0;
36     }
37 }
```

# 반복문을 사용한 조명 제어



- 회로도
- 키트에 있는 LED Array(3색 LED)를 사용하여 제작
- 첫번째 핀에 GND, 각각 나머지에 D1, D2, D3를 연결
- 프로그램을 작성하여 LED가 각각 3번, 3번, 3번씩 반복하여 깜빡이도록 제작
  - for문 사용

# 반복문을 사용한 조명

## ■ Source

```
06
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 }
18
19 void loop() {
20     count++;
21     if ( count == 1 )
22     {
23         for ( i = 1; i <= 3; i++ ) {
24             digitalWrite(PIN_LED_BLUE, HIGH);
25             delay(200);
26             digitalWrite(PIN_LED_BLUE, LOW);
27             delay(200);
28         }
29     } else if ( count == 2 ) {
30         for ( i = 1; i <= 3; i++ ) {
31             digitalWrite(PIN_LED_GREEN, HIGH);
32             delay(200);
33             digitalWrite(PIN_LED_GREEN, LOW);
34             delay(200);
35         }
36     }
```

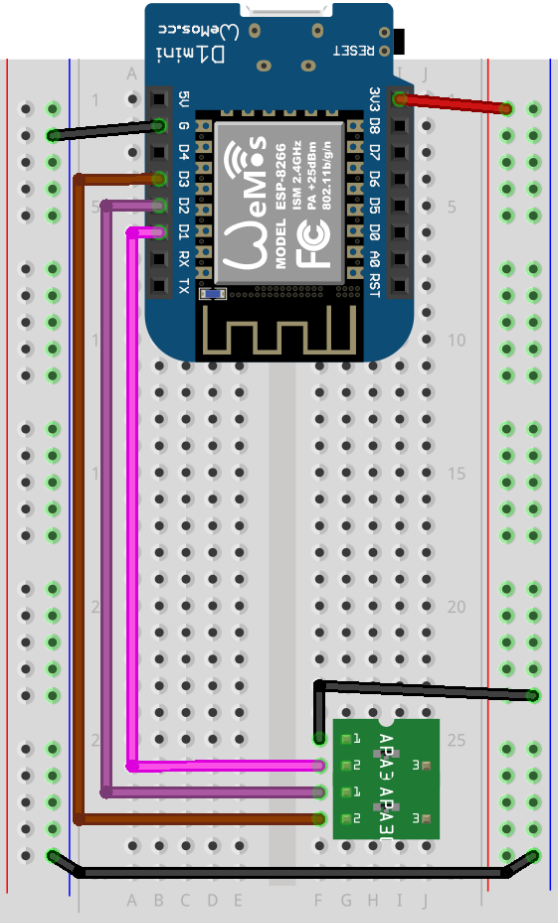


# 반복문을 사용한 조명 제어

- Source

```
36 } else if ( count == 3 ) {  
37     for ( i = 1; i <= 3; i++ ) {  
38         digitalWrite(PIN_LED_RED, HIGH);  
39         delay(200);  
40         digitalWrite(PIN_LED_RED, LOW);  
41         delay(200);  
42     }  
43     count = 0;  
44 }  
45 }
```

# 시리얼 입출력을 사용한 조명 제어



fritzing

- 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 입출력을 사용

## ■ 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);
```

# 시리얼 입출력을 사용한 조명 제어

## ■ Source

```
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 }
```

# 반복문을 사용한 조명 제어 - 응용

- 물리적 버튼이나 프로그래밍 수정을 통하여 조명(LED)의 행동을 제어 해 본다.
  - Ex) 버튼을 클릭하면 다음 LED만 동작하도록 제어
- 반복문의 횟수를 제어하여 조명의 깜빡임 횟수를 조정 해 본다.
- 시리얼 모니터를 사용하여 특정 값이 나왔을 때 조명이 동작하도록 제어

■ QnA

# 오늘 배운 내용

- LED가 무엇인지 이해한다.
- GND와 VCC의 차이를 알고 설명 할 수 있다.
- 한 개의 LED 및 다중 LED의 처리를 할 수 있다.
- LED들을 사용하여 가상의 신호등을 제작 할 수 있다.
- 조건과 반복문을 사용한 코드 응용
- 이론 내용 복습 및 프로젝트 내용 이해.