앱인벤터, ESP32로 사물인터넷 디자인하기

코딩거점학교

차례

1. 디지털 출력

1 ESP32 소개

MCU(마이크로 컨트롤러 유닛)

- 회사에 따라 NodeMCU, DEVKIT, LOLIN, TIGO 등 제품 생산

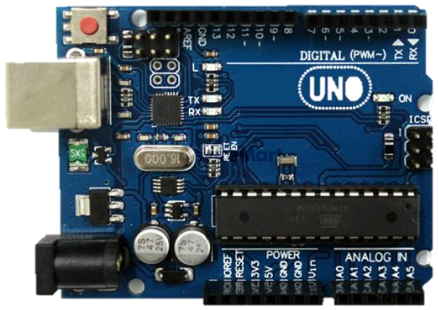
3.3V 전압 사용

개발 환경: IDF, 스케치, 마이크로파이선

CH340, CH9102, CP2102 등의 칩 사용

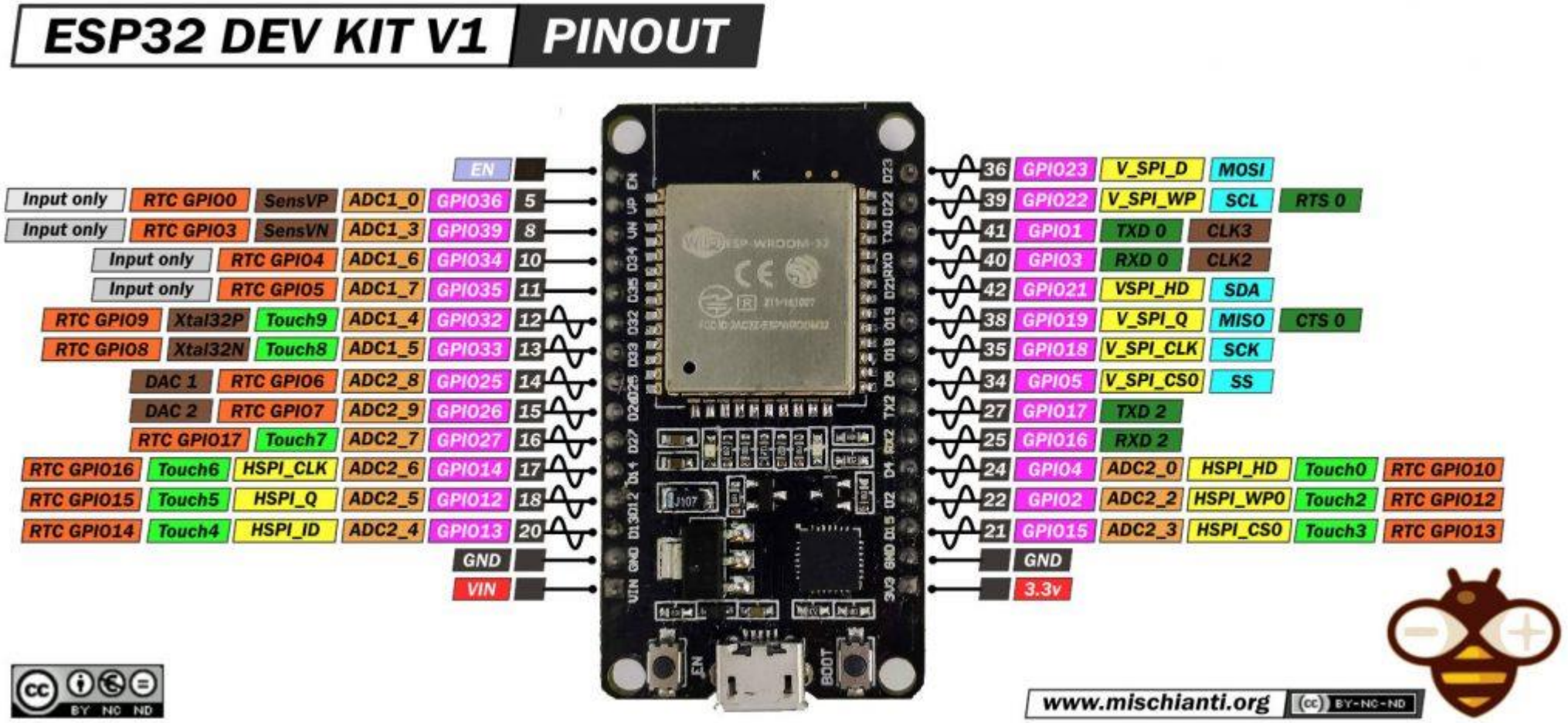
우리가 사용할 ESP-WROOM-32버전은 ESP32와 결합하여 외부장치를 편리하게 연결 가능

1 ESP32 소개



아두이노 우노	성능	ESP32
8비트 싱글코어 16MHz	**처리 속도**	**32비트** 듀얼코어 80~240MHz
2Kbyte	**메모리**	512Kbyte
X	**와이파이**	**O**
X	**블루투스**	**O**
14개	**입출력핀**	36개(analogWrite 대신 채널)
6개(0~1023으로 표현)	**아날로그 입력핀**	16개(0~4095로 표현)
0개	**DAC 핀**	2개

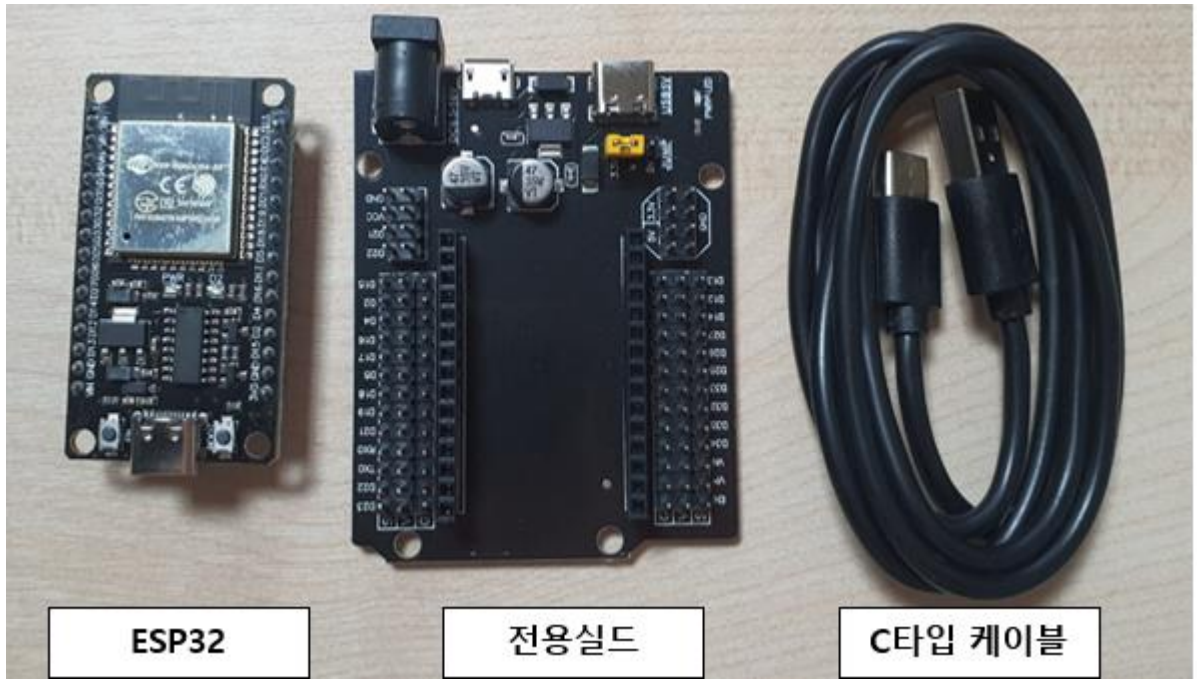
1 ESP32 소개



ESP32 보드는 회사마다 핀 배열이 다름

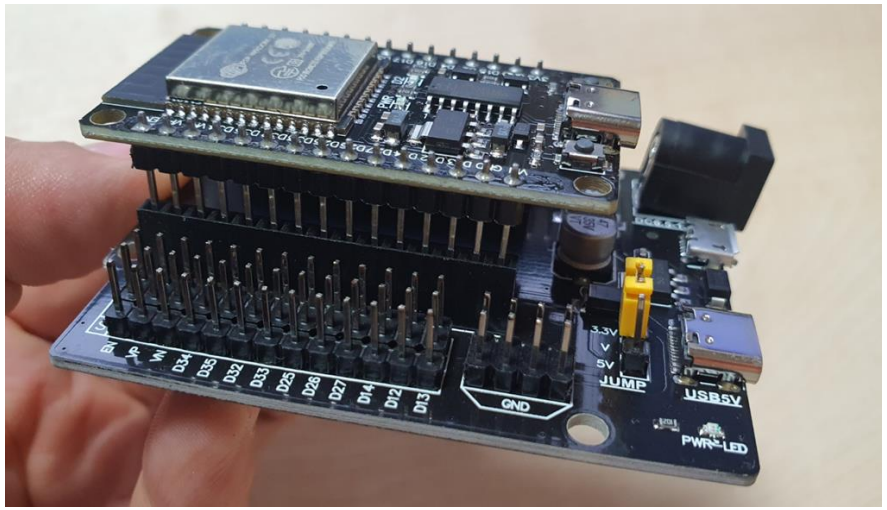
ESP32에 사용한 번호는 분홍색 GPIO 번호임

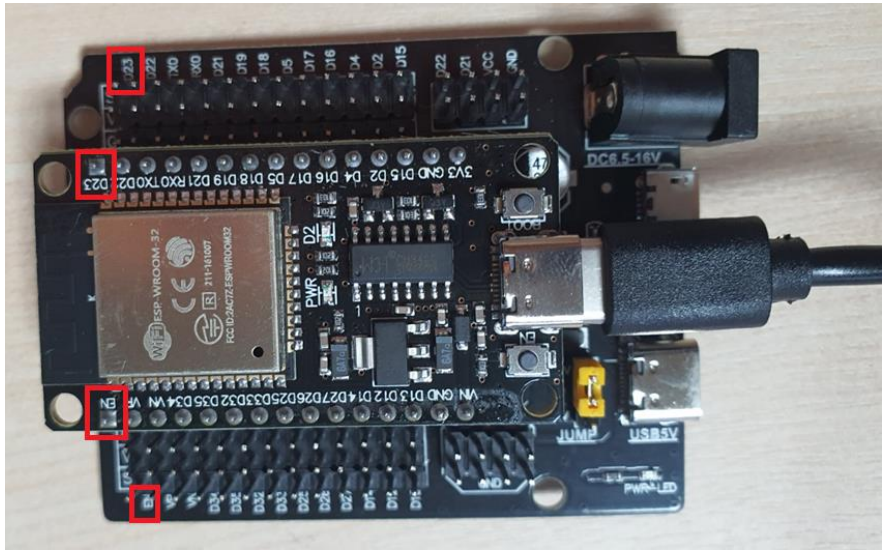
1 ESP32 소개



ESP32, ESP32 실드, C타입 케이블 준비

1 ESP32 소개





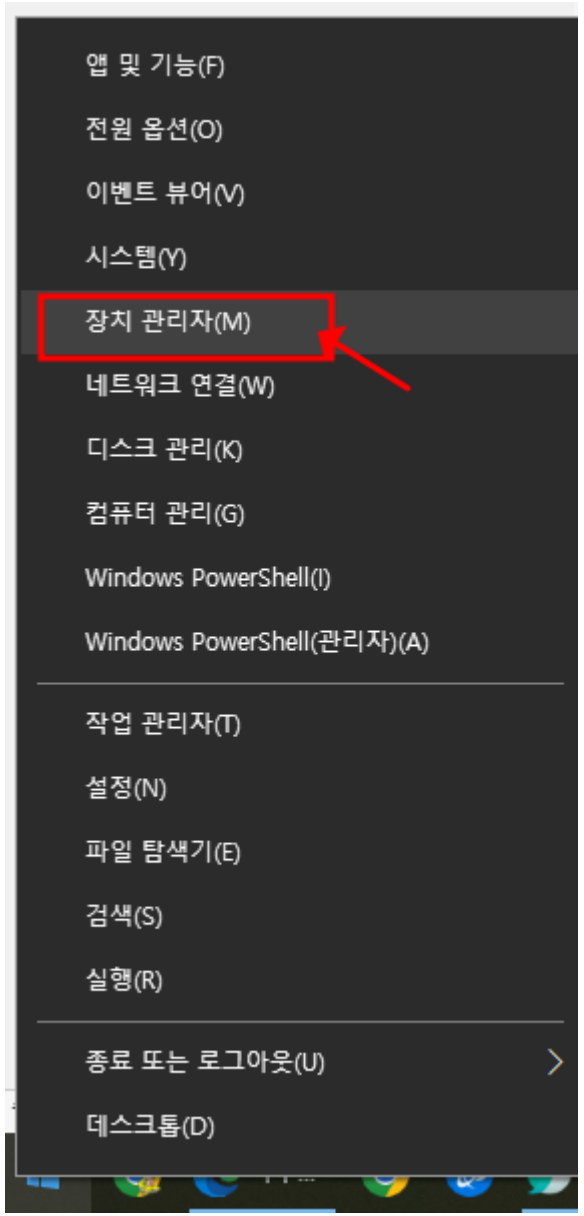
ESP32, ESP32 실드 결합 후 PC와 연결

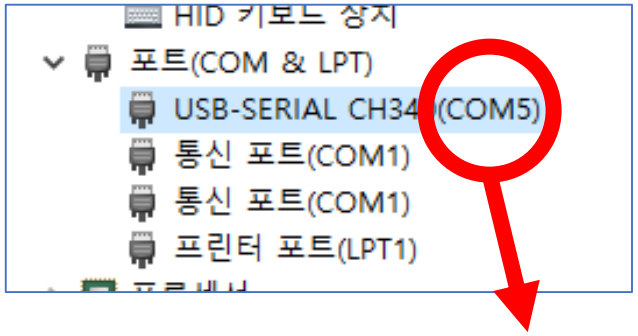
※ ESP32에 꽂아야 함! 실드의 전원에 꼽지 않도록 주의!

1 ESP32 소개

ESP32는 ch340 드라이버가 설치되어 있어야 사용 가능

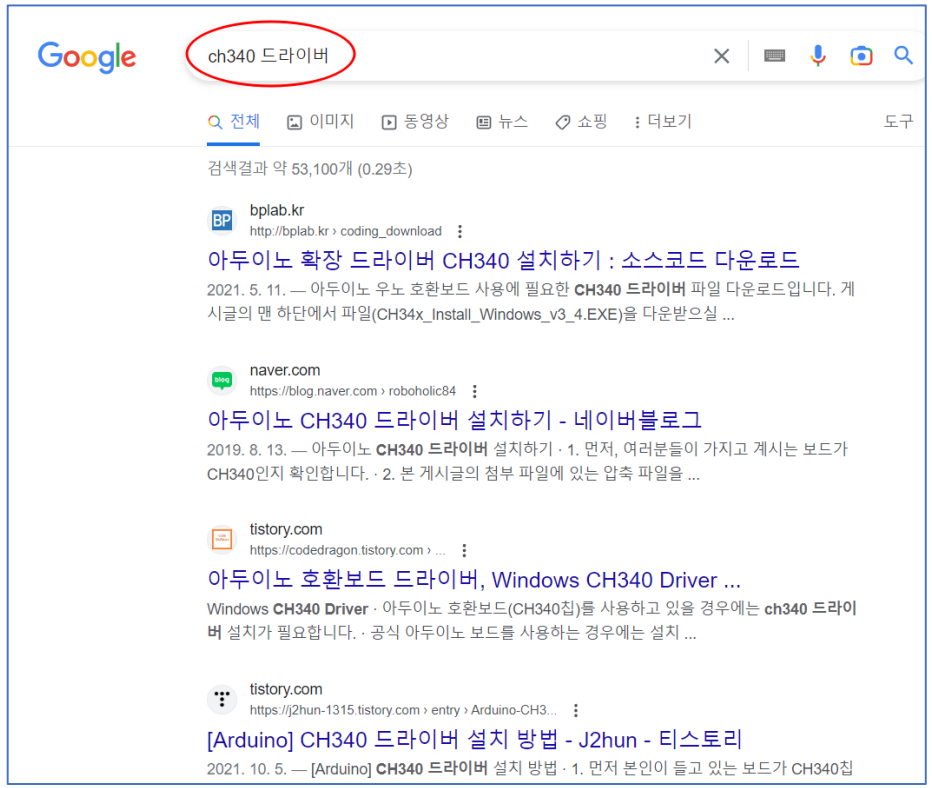
장치관리자-port에서 설치 여부 확인

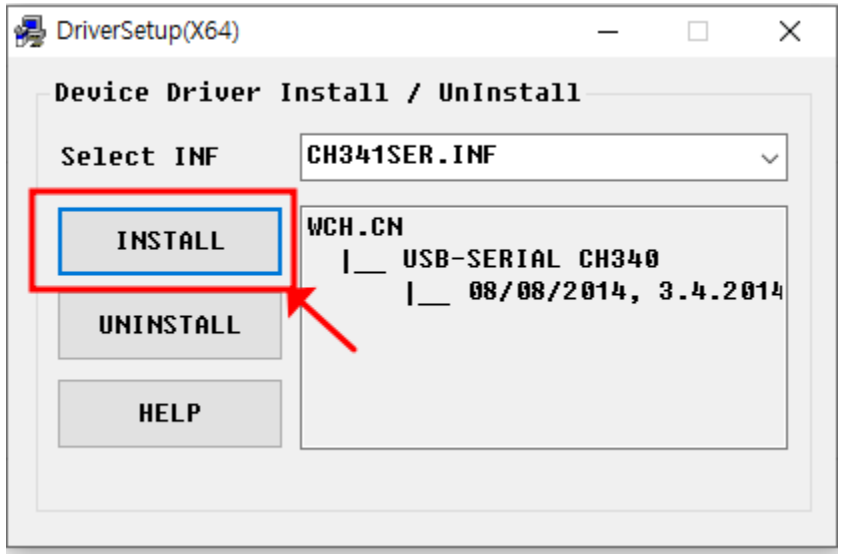




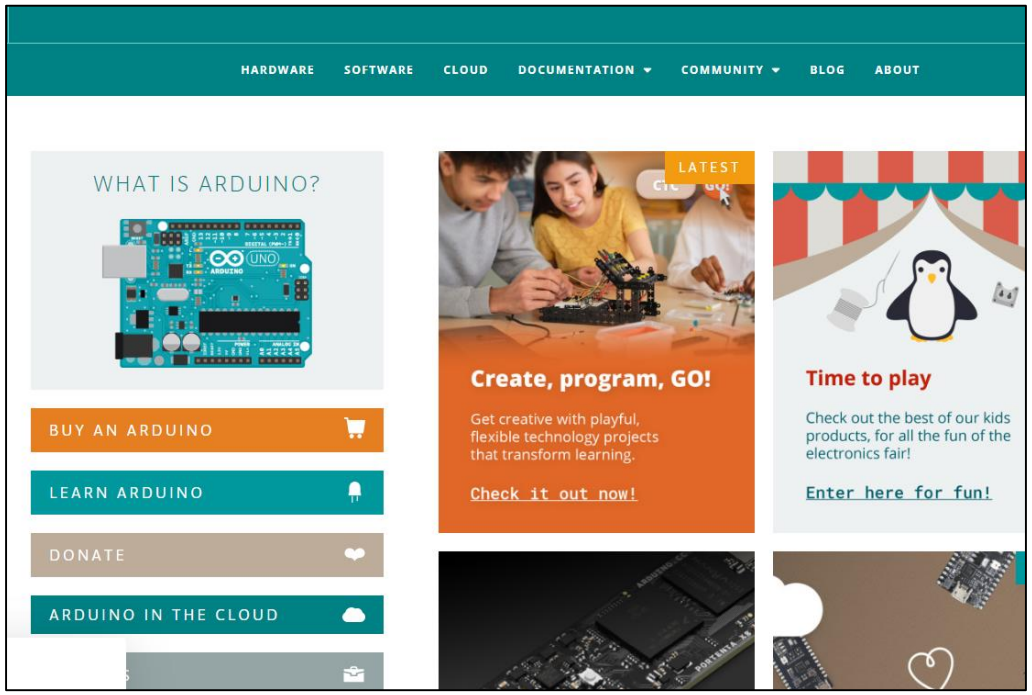
포트 숫자 기억

1 ESP32 소개

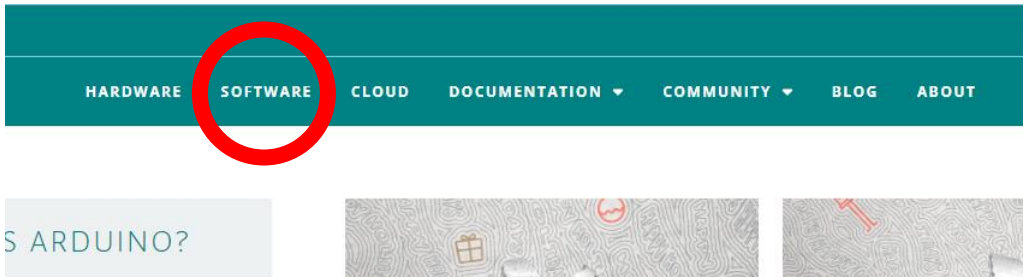

Google
ch340 드라이버
전체
이미지
동영상
뉴스
쇼핑
더보기
도구
검색결과 약 53,100개 (0.29초)
bplab.kr
http://bplab.kr › coding_download
아두이노 확장 드라이버 CH340 설치하기 : 소스코드 다운로드
2021. 5. 11. — 아두이노 우노 호환보드 사용에 필요한 CH340 드라이버 파일 다운로드입니다. 게시글의 맨 하단에서 파일(CH34x_Install_Windows_v3_4.EXE)을 다운받으실 ...
naver.com
https://blog.naver.com › roboholic84
아두이노 CH340 드라이버 설치하기 - 네이버블로그
2019. 8. 13. — 아두이노 CH340 드라이버 설치하기 · 1. 먼저, 여러분들이 가지고 계시는 보드가 CH340인지 확인합니다. · 2. 본 게시글의 첨부 파일에 있는 압축 파일을 ...
tistory.com
https://codedragon.tistory.com › ...
아두이노 호환보드 드라이버, Windows CH340 Driver ...
Windows CH340 Driver · 아두이노 호환보드(CH340칩)를 사용하고 있을 경우에는 ch340 드라이버 설치가 필요합니다. · 공식 아두이노 보드를 사용하는 경우에는 설치 ...
tistory.com
https://j2hun-1315.tistory.com › entry › Arduino-CH3...
[Arduino] CH340 드라이버 설치 방법 - J2hun - 티스토리
2021. 10. 5. — [Arduino] CH340 드라이버 설치 방법 · 1. 먼저 본인이 들고 있는 보드가 CH340칩


DriverSetup(X64)
Device Driver Install / UnInstall
Select INF
CH341SER.INF
INSTALL
UNINSTALL
HELP
WCH.CN
|__ USB-SERIAL CH340
|__ 08/08/2014, 3.4.2014

2 ESP32 개발 환경



https://arduino.cc 접속



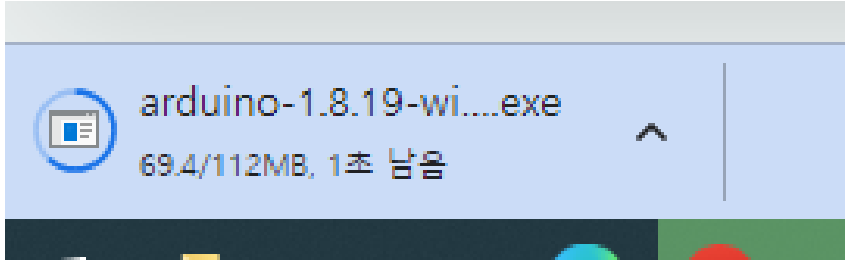
SOFTWARE 클릭

2 ESP32 개발 환경

스크롤을 내려서
Arduino IDE 1.8.19 다운

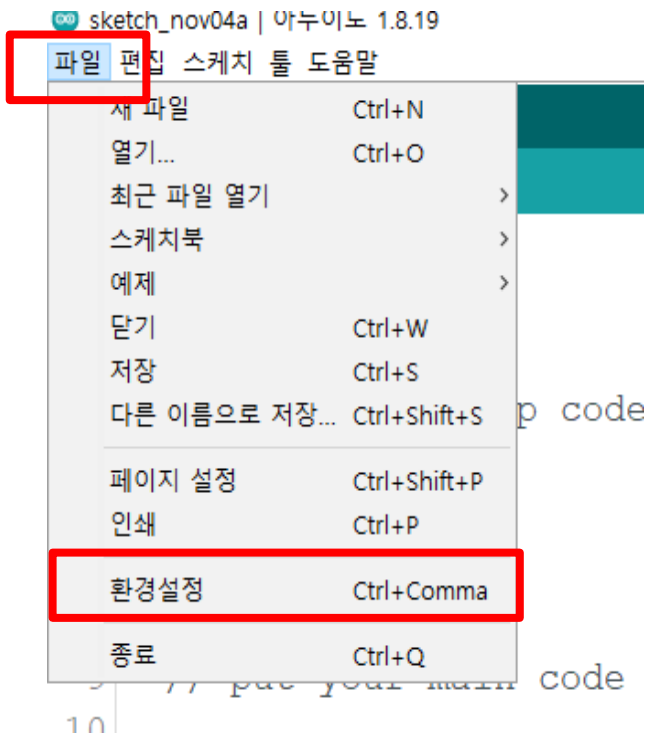
2 ESP32 개발 환경

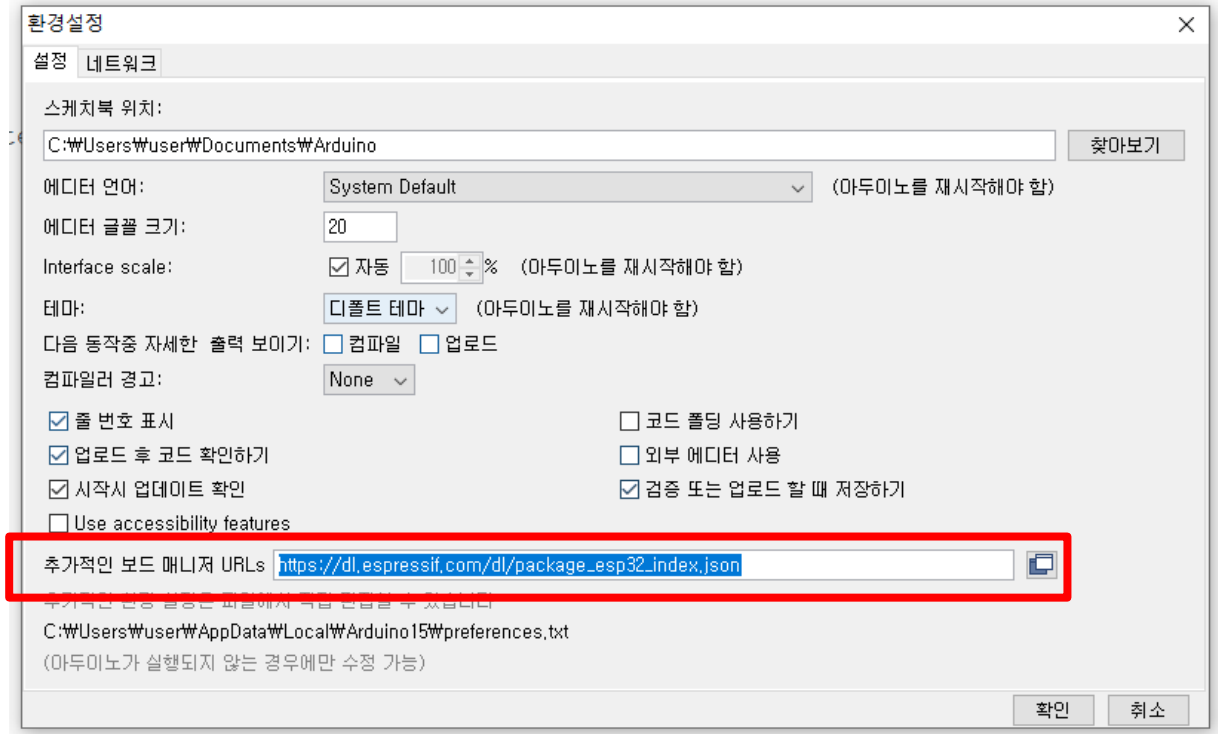
JUST DOWNLOAD 클릭



다운 받은 파일 실행

2 ESP32 개발 환경



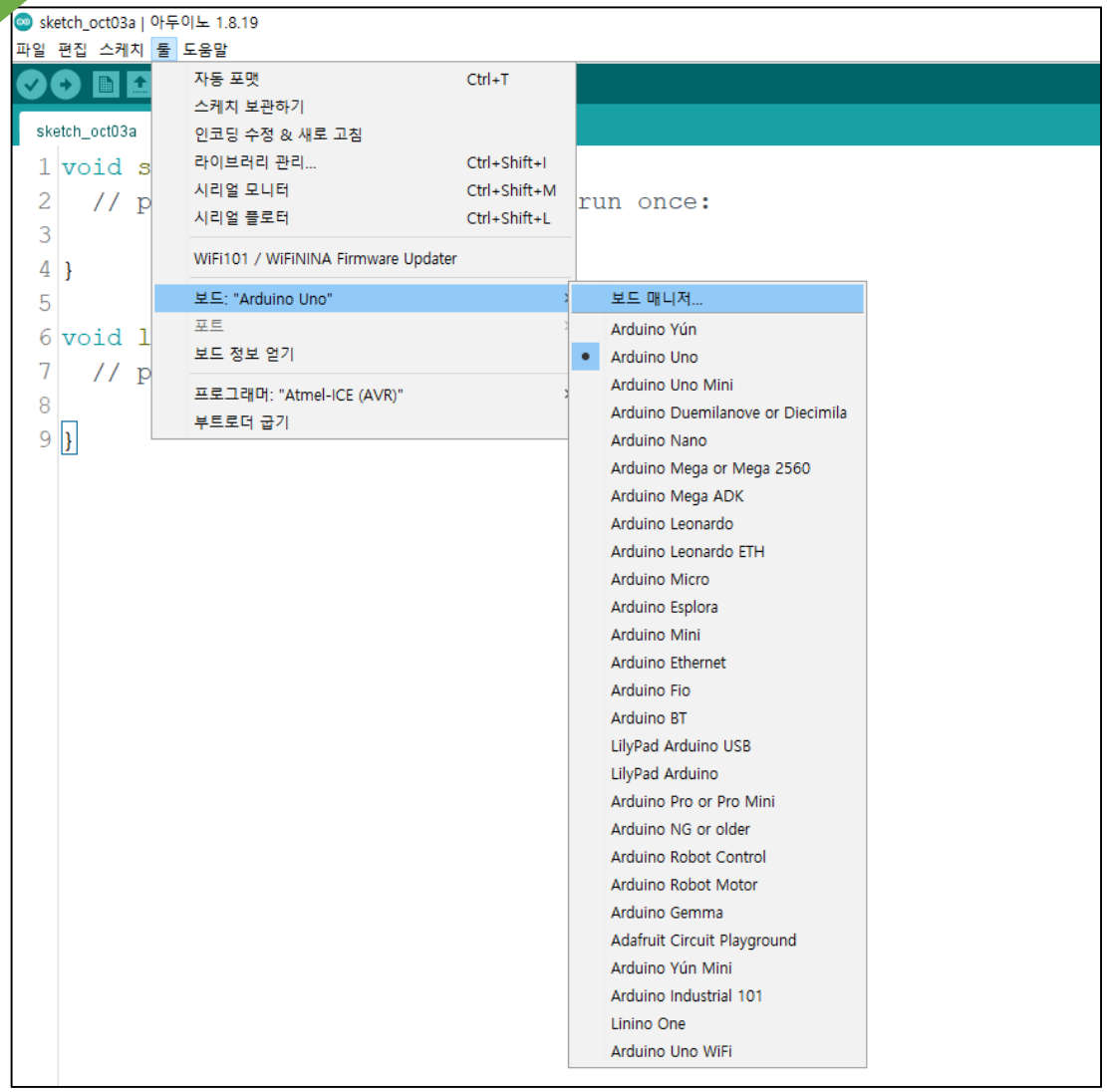


파일 >> 환경 설정

추가적인 보드 매니저 URLs에

https://dl.espressif.com/dl/package_esp32_index.json 주소를 추가

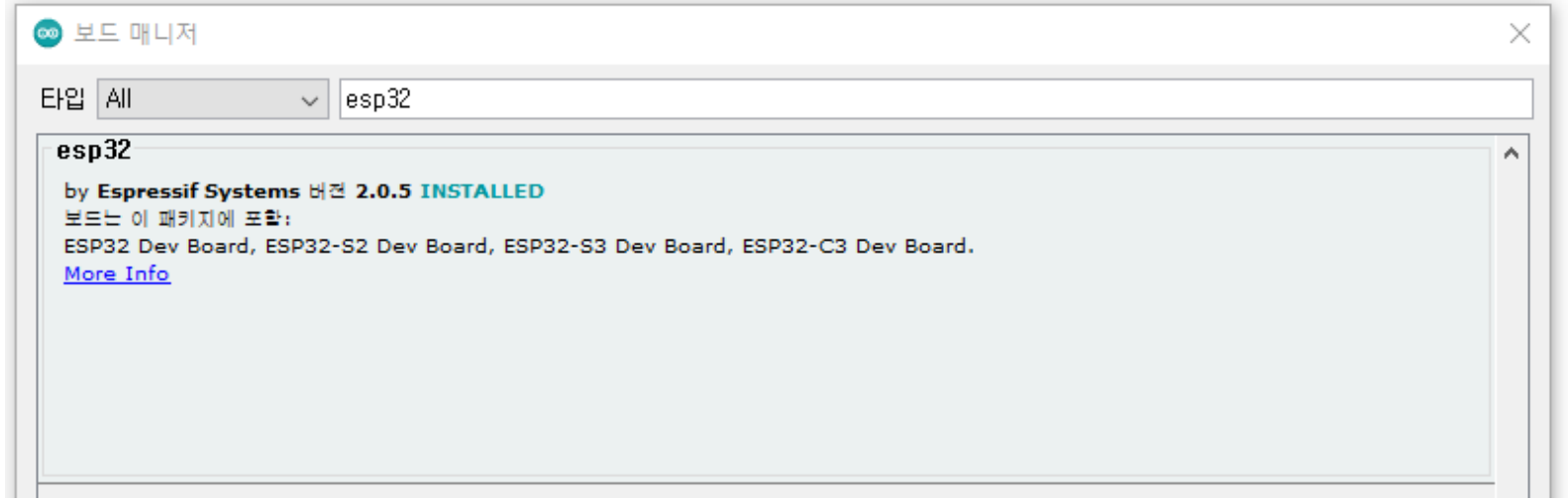
2 ESP32 개발 환경



툴 메뉴에서

보드>>보드 매니저 선택

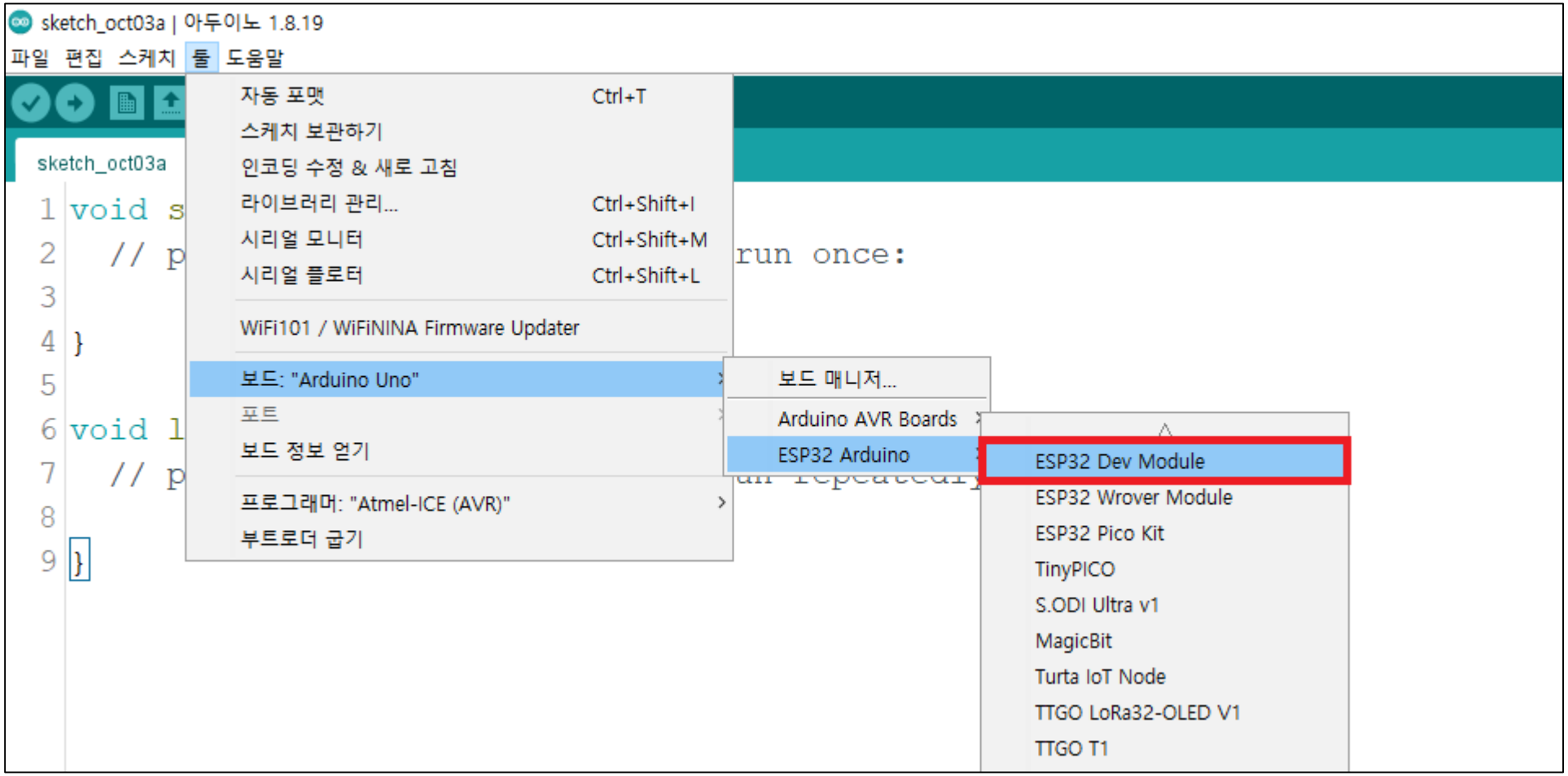
2 ESP32 개발 환경



ESP32 보드 검색하여

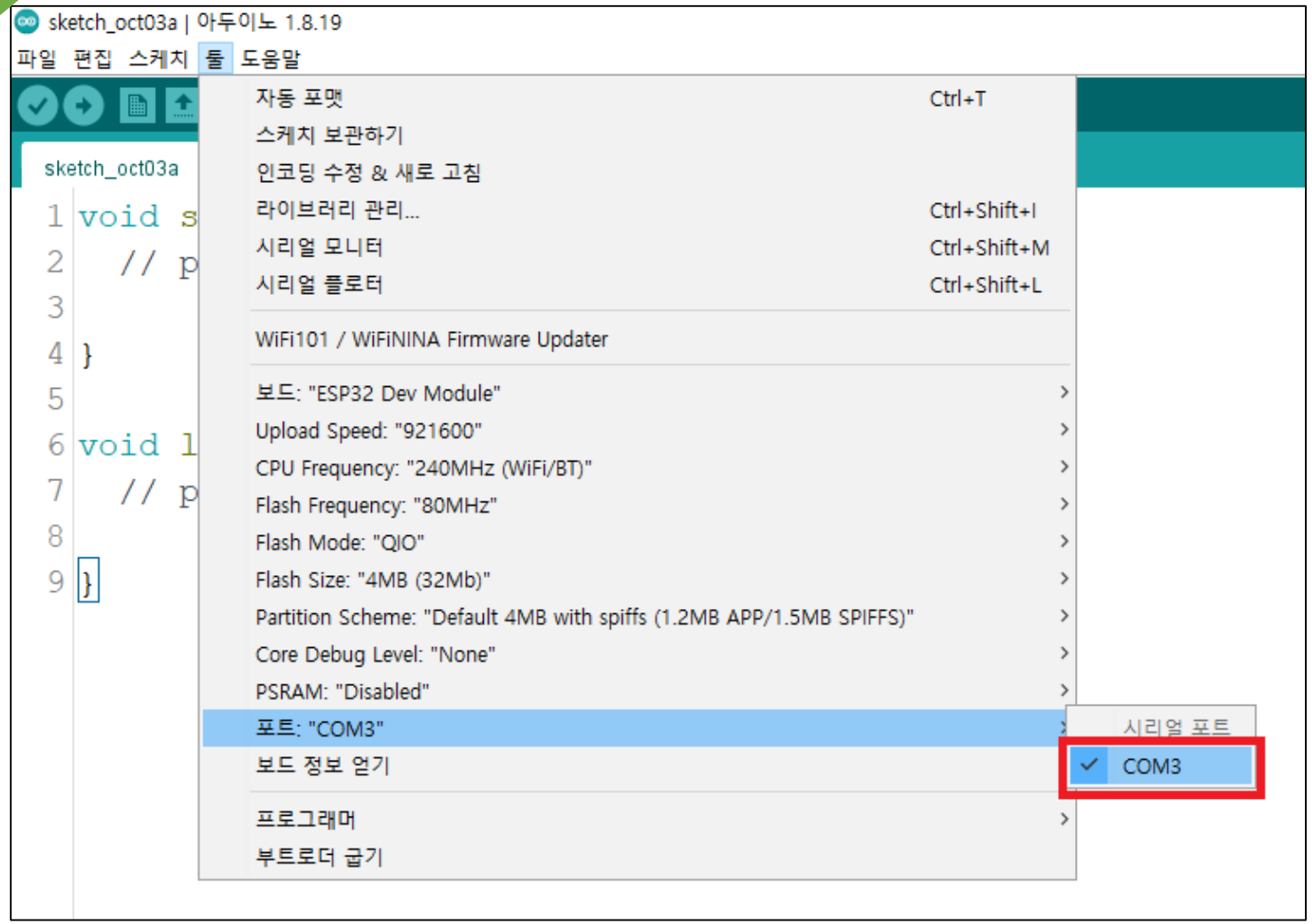
설치

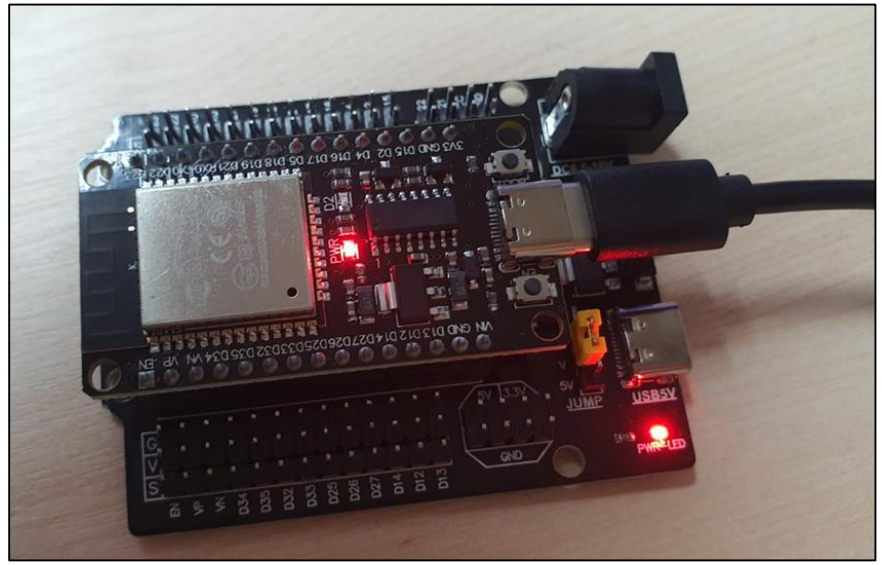
2 ESP32 개발 환경



툴 >> 보드 >> ESP32에서 ESP32 Dev Module 선택

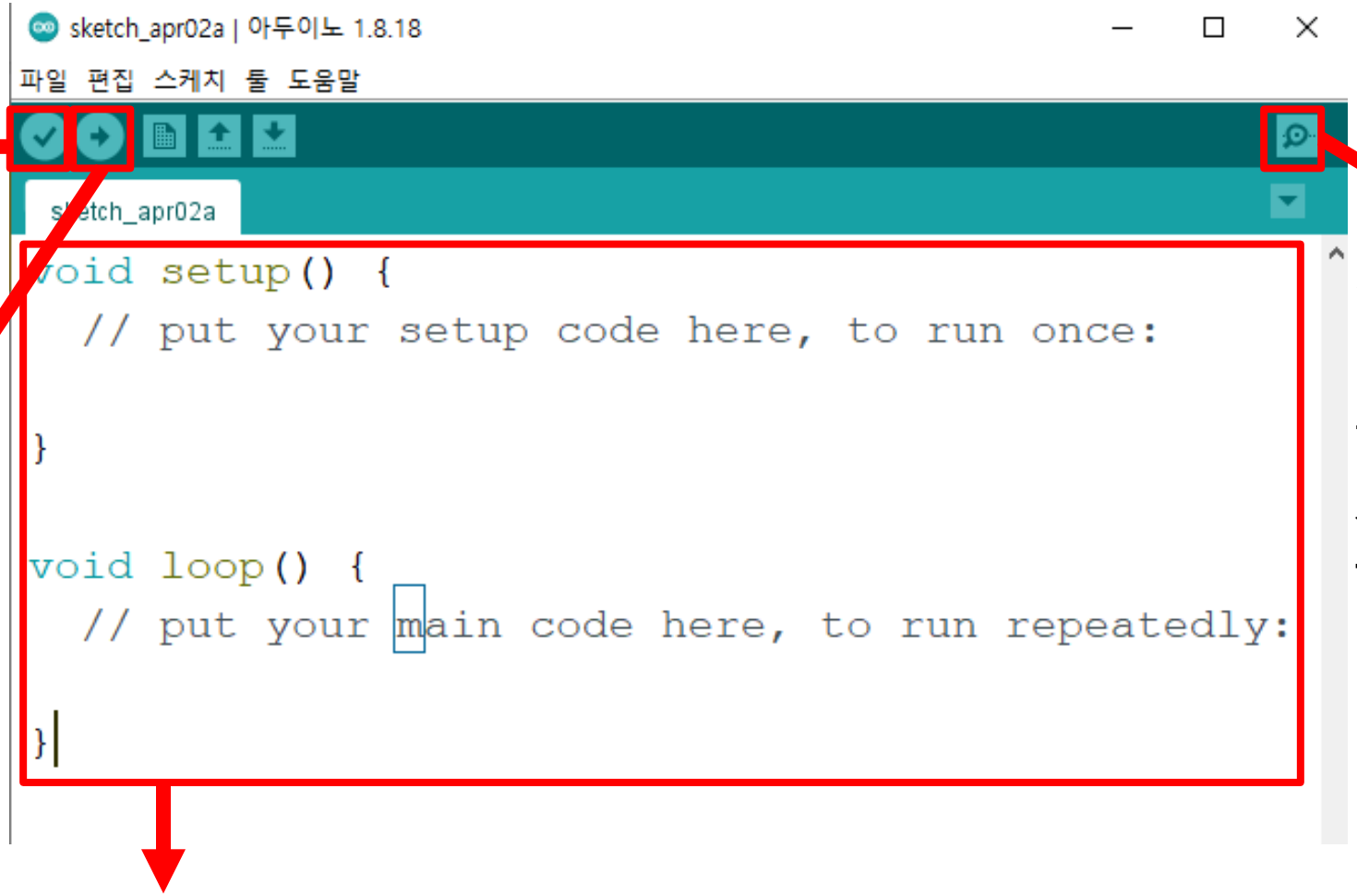
2 ESP32 개발 환경





포트 선택 (장치관리자에서 본 포트 번호)

2 ESP32 개발 환경



확인 버튼

-컴파일을 함
-코드 이상 유무 확인 가능

업로드 버튼

-컴파일 후 보드에 코드를 업로드함

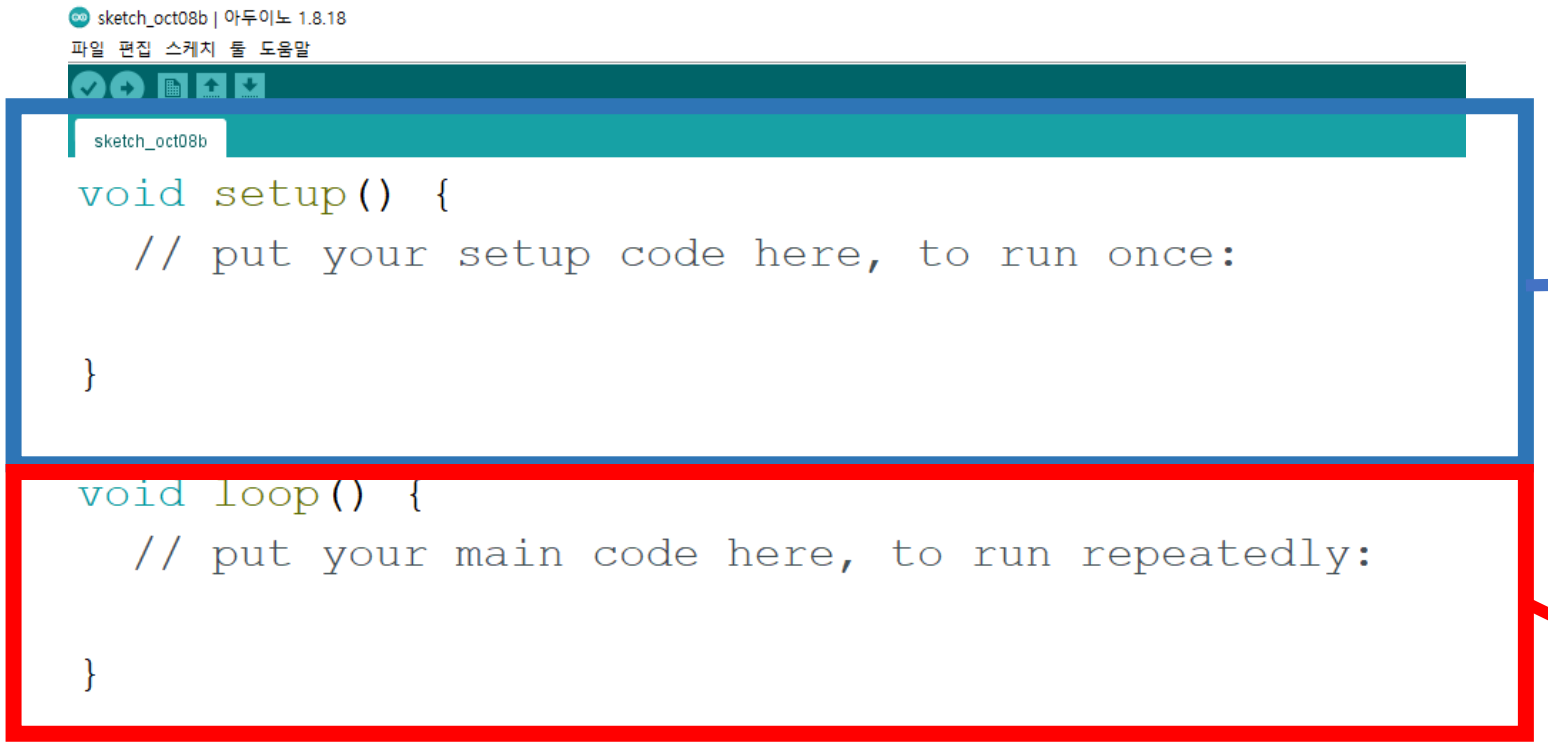
시리얼 모니터 버튼

-컴퓨터와 ESP32 간에 시리얼 통신으로 ESP32의 상태를 모니터링할 수 있음

코드 입력 창

-명령어를 입력하여 프로그램을 만들 수 있음

2 ESP32 개발 환경


sketch_oct08b | 아두이노 1.8.18
파일 편집 스케치 툴 도움말
sketch_oct08b
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}
한 번 만 실행
반복 실행

3 ESP32 작동 테스트

setup 함수에 pinMode 입력 2번 핀을 output 설정

```
void setup() {

pinMode(2, OUTPUT);

}
```

3 ESP32 작동 테스트

- digitalWrite 명령어로 2번 핀에 전기 신호를 줌
- delay 명령어로 0.3초 유지(1000이 1초임)

```
void loop() {

digitalWrite(2, HIGH);

delay(300);
```

3 ESP32 작동 테스트

이번에는 전기 출력 신호를 주지 않고 0.3초간 유지함

```
digitalWrite(2, LOW);

delay(300);

}
```

3 ESP32 작동 테스트

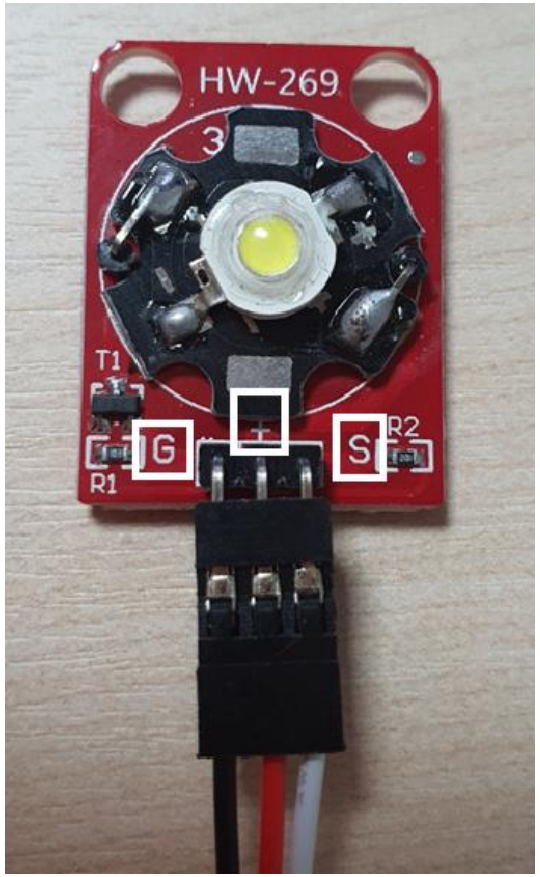
테스트 예제

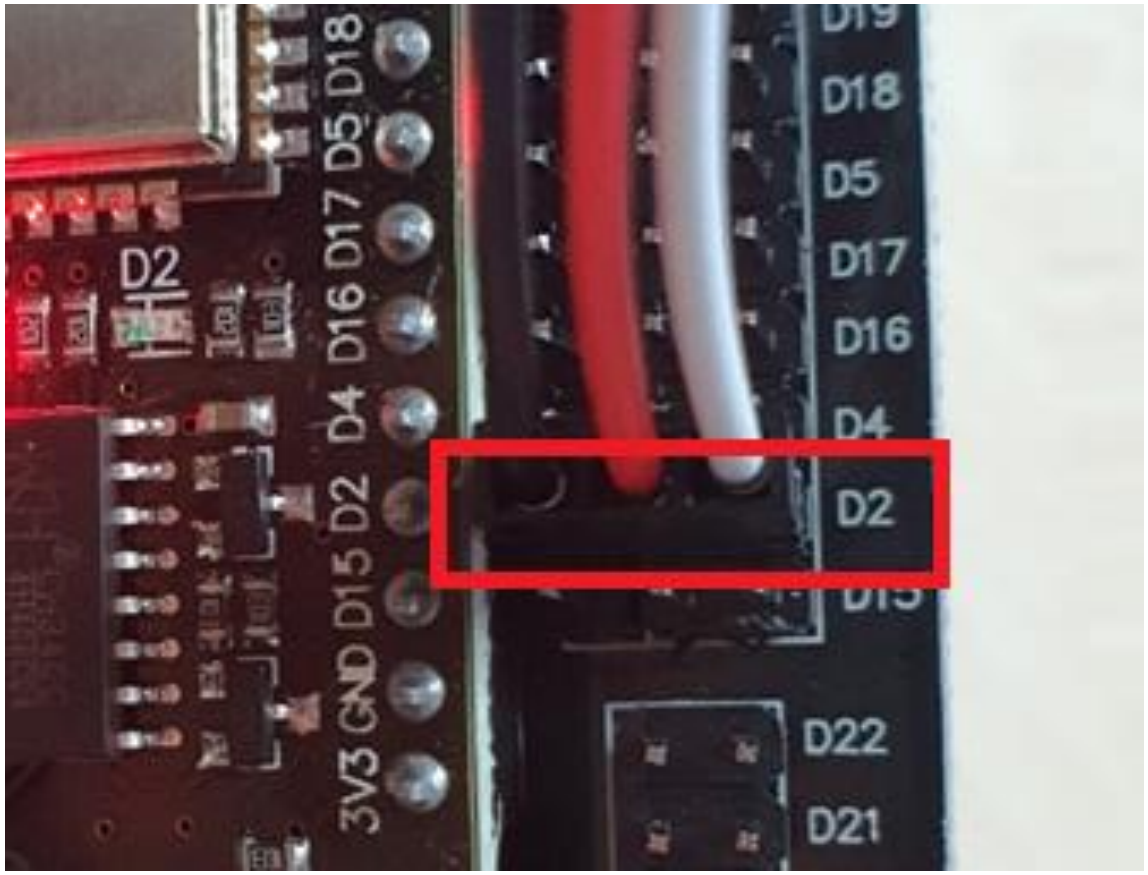
```
void setup() {
pinMode(2, OUTPUT);
}
void loop() {
digitalWrite(2, HIGH);
delay(300);
digitalWrite(2, LOW);
delay(300);
}
```

ESP32 2번 핀은 내부 led와 연결되어 있음

0.3초 간격으로 3.3V 전압이 입력됨

3 ESP32 작동 테스트



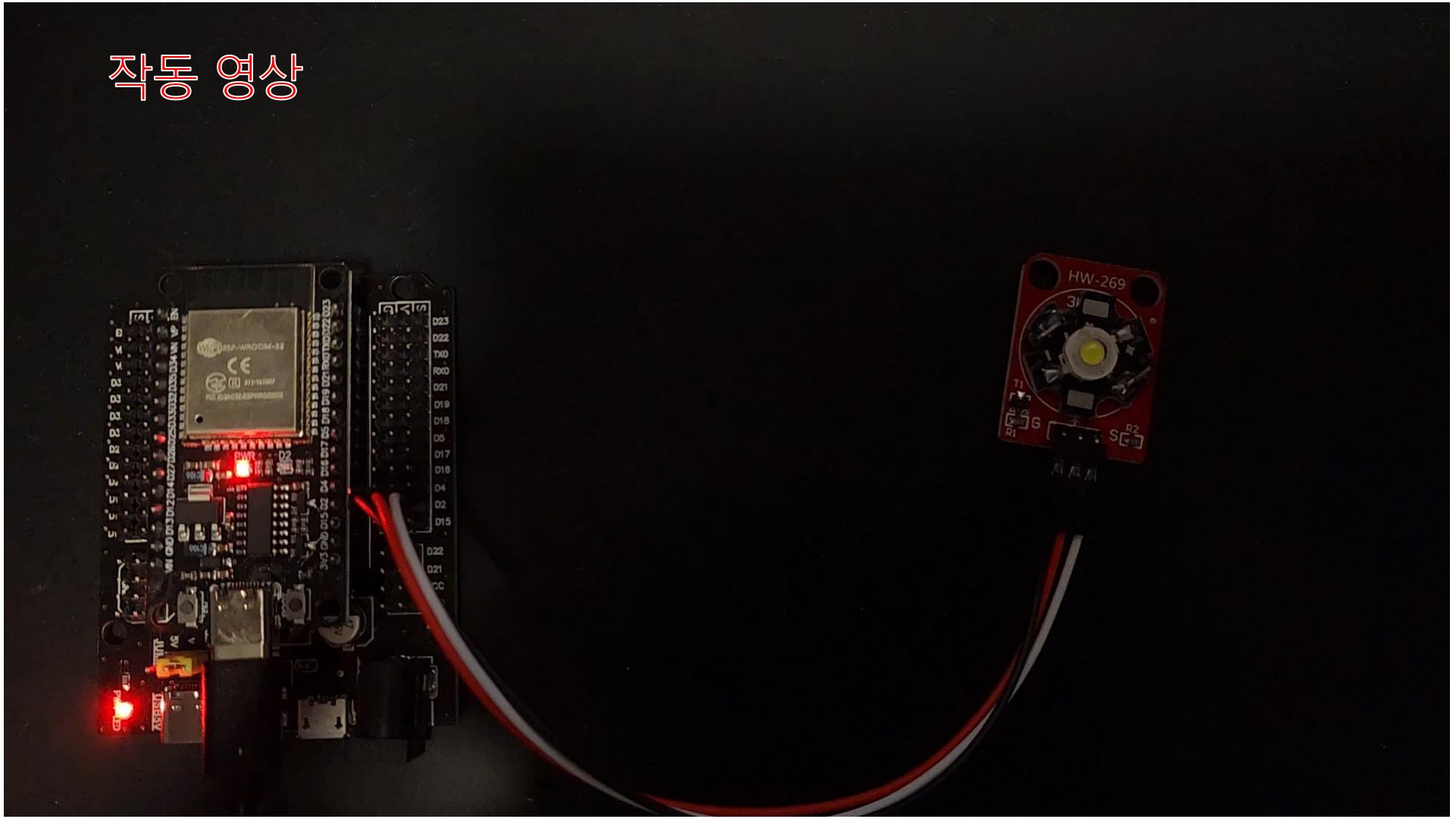


디지털 2번 핀을 외부 기기와도 연결 가능

ESP32에서 3W LED를 꺼내 연결하면 온보드 LED와 똑같이 점등됨

(눈부심 주의!　아, 내 눈~ㅠㅠ)

3 ESP32 작동 테스트



4 LED 제어 – SOS 신호

도전해 볼까요?

아래 내용을 참고로 해서 13번 LED로 모스부호 sos를 표시해 주세요

1. 모스부호 SOS는 ... ___ ...(짧은 신호 3번, 긴 신호 3번, 짧은 신호 3번)입니다.(돈돈돈쯔쯔쯔돈돈돈)
2. 짧은 신호 간격은 0.2초(0.2초 켜졌다가 0.2초 꺼짐), 긴 신호 간격은 0.5초(0.5초 켜졌다가 0.2초 꺼짐)로 합니다.
3. SOS를 다 친 후에는 1초 후에 다시 반복합니다.

4 LED 제어 – SOS 신호

pinMode로 2번 핀을 출력 설정

```
void setup() {

  pinMode(2, OUTPUT);

}
```

4 LED 제어 – SOS 신호

변수 i가 0, 1, 2 로 값이 바뀔 때마다 아래 명령어 실행(세 번 반복함)

```
void loop() {
  for(int i=0 ; i<3 ; i++) {
```

4 LED 제어 – SOS 신호

0.2초간 켜고 0.2초간 꺼짐

```
digitalWrite(2, HIGH);
delay(200);
digitalWrite(2, LOW);
delay(200); }
```

4 LED 제어 – SOS 신호

0.5초간 켰다가 0.2초간 꺼짐

```
for(int i=0 ; i<3 ; i++) {
        digitalWrite(2, HIGH);
        delay(500);
        digitalWrite(2, LOW);
        delay(200); }
```

4 LED 제어 – SOS 신호

0.2초간 켜고 0.2초간 꺼짐

```
for(int i=0 ; i<3 ; i++) {
        digitalWrite(2, HIGH);
        delay(200);
        digitalWrite(2, LOW);
        delay(200); }
```

4 LED 제어 – SOS 신호

반복 후에 1초간 기다림

```
 delay(1000);

}
```

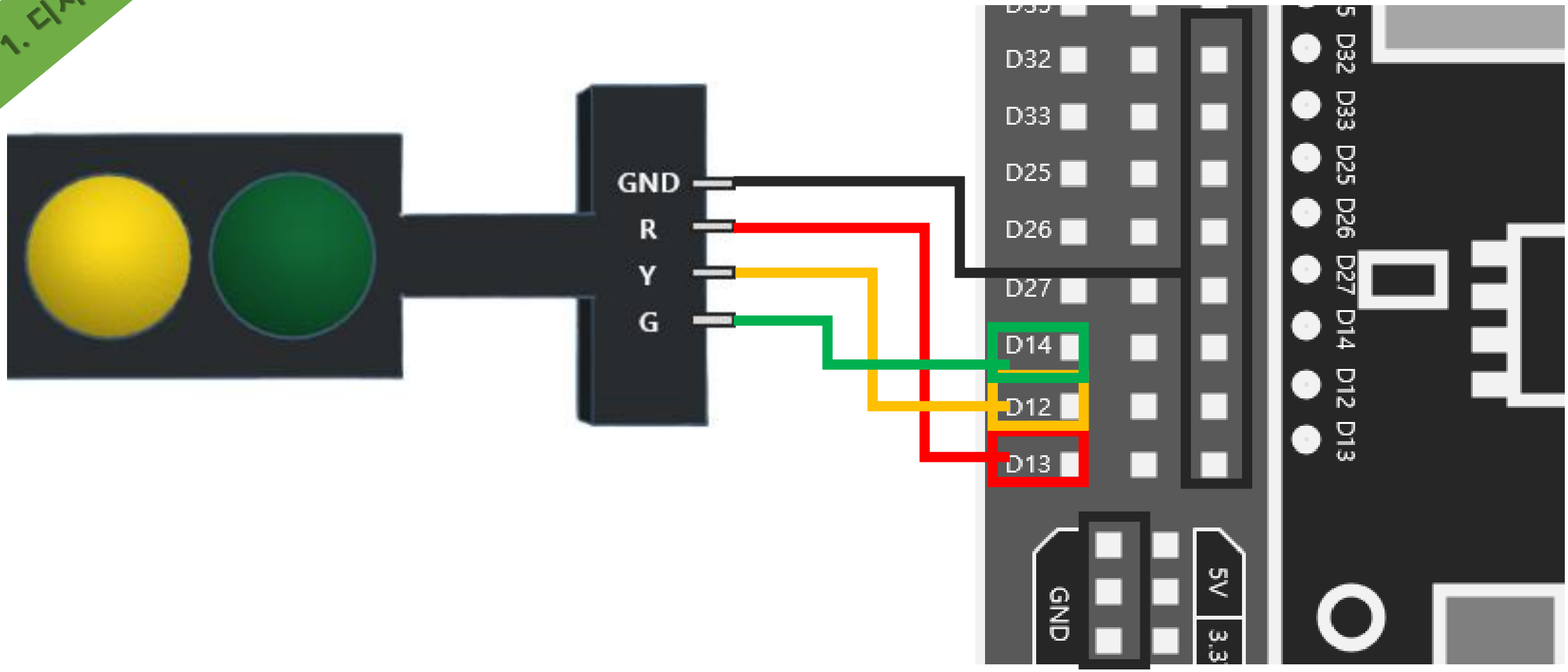
4 LED 제어 – SOS 신호

전체 코드

```
void setup() {
  pinMode(2, OUTPUT);
}
void loop() {
  for(int i=0 ; i<3 ; i++) {
    digitalWrite(2, HIGH);
    delay(200);
   digitalWrite(2, LOW);
    delay(200); }
for(int i=0 ; i<3 ; i++) {
    digitalWrite(2, HIGH);
    delay(500);
   digitalWrite(2, LOW);
    delay(200); }
```

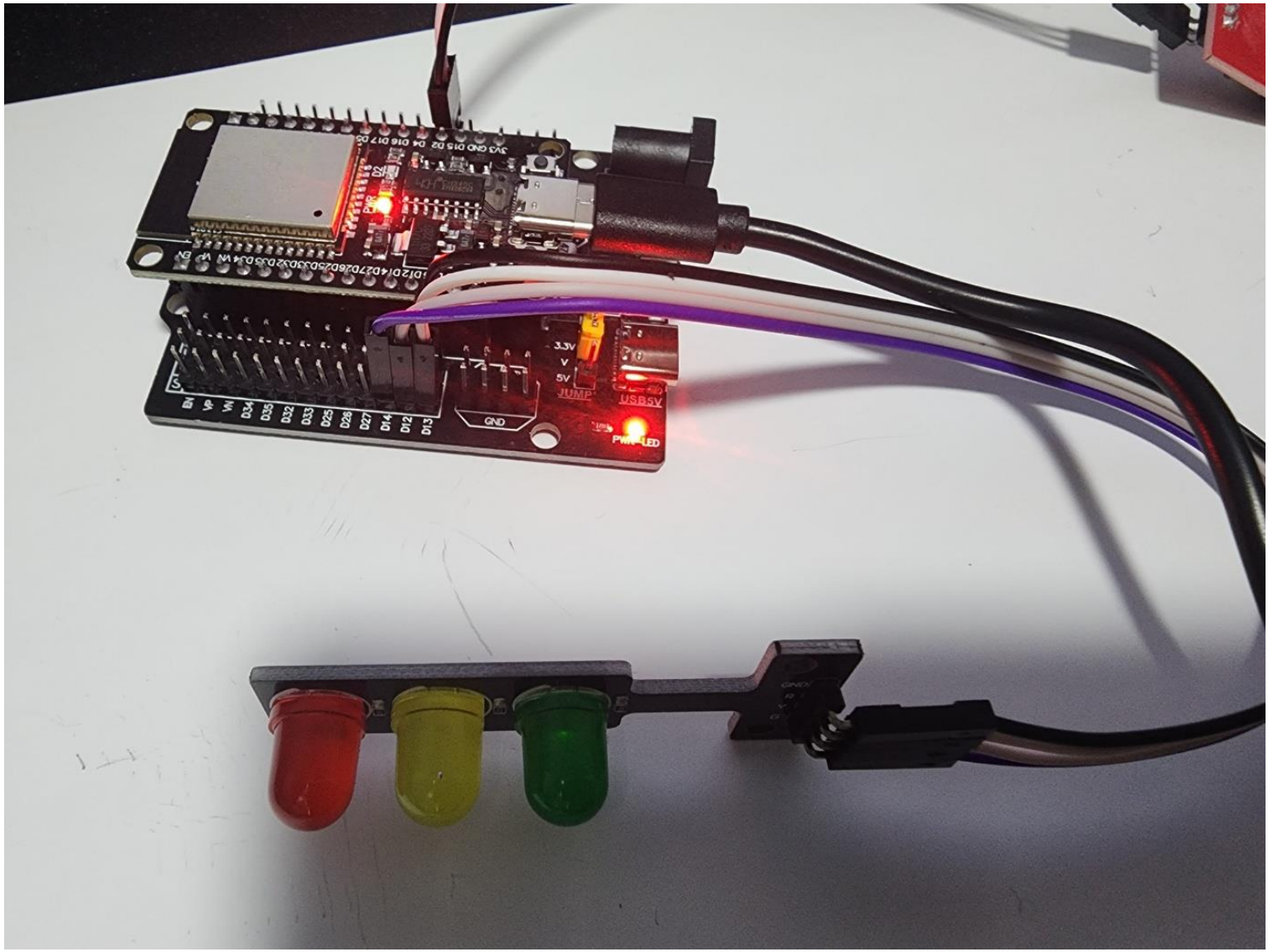

```
 for(int i=0 ; i<3 ; i++) {
    digitalWrite(2, HIGH);
    delay(200);
   digitalWrite(2, LOW);
    delay(200); }
    delay(1000);
}
```

5 신호등 LED 제어



R은 GPIO13, Y는 GPIO12, G는 GPIO14, GND는 GND핀 아무 데나 연결

5 신호등 LED 제어



5 신호등 LED 제어

신호등의 빨강, 노랑, 초록 LED에 ESP32에 실제 꼽은 것과 같이 핀을 할당해 줍니다.

```
int red    = 13;

int yellow = 12;

int green  = 14;
```

5 신호등 LED 제어

빨강, 노랑, 초록 핀을 출력으로 설정합니다.

```
void setup() {
    pinMode(red, OUTPUT);
    pinMode(yellow, OUTPUT);
    pinMode(green, OUTPUT);
}
```

5 신호등 LED 제어

빨강 LED 만 켜고 다른 LED는 끈 상태로 0.5초간 유지합니다.

```
void loop() {
  digitalWrite(red, HIGH);
  digitalWrite(yellow, LOW);
  digitalWrite(green, LOW);
  delay(500);
```

5 신호등 LED 제어

노랑 LED 만 켜고 다른 LED는 끈 상태로 0.5초간 유지합니다.

```
digitalWrite(red, LOW);
digitalWrite(yellow, HIGH);
digitalWrite(green, LOW);
delay(500);
```

5 신호등 LED 제어

초록 LED 만 켜고 다른 LED는 끈 상태로 0.5초간 유지합니다.

```
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  digitalWrite(green, HIGH);
  delay(500);
}
```

5 신호등 LED 제어

코딩 예제

```
int red    = 13;
int yellow = 12;
int green  = 14;

void setup() {
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}
```

```
void loop() {
  digitalWrite(red, HIGH);
  digitalWrite(yellow, LOW);
  digitalWrite(green, LOW);
  delay(500);
  digitalWrite(red, LOW);
  digitalWrite(yellow, HIGH);
  digitalWrite(green, LOW);
  delay(500);
```

```
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
  digitalWrite(green, HIGH);
  delay(500);
}
```

6 신호등 LED – 신호등 구현

도전해 볼까요?

신호등을 만들어봅시다

1. 빨간 LED 4초 켜짐
2. 빨간 LED 꺼지고 노란 LED 1초 켜짐
3. 노란 LED 꺼지고 초록 LED 1초 켜짐
4. 초록 LED 0.2초 간격으로 5번 깜빡임(for문 이용)

6 신호등 LED – 신호등 구현

```
int red    = 13;
int yellow = 12;
int green  = 14;

void setup() {
    pinMode(red, OUTPUT);
    pinMode(yellow, OUTPUT);
    pinMode(green, OUTPUT);
}
```

- 핀 번호 지정
- 핀 모드 설정

6 신호등 LED – 신호등 구현

빨강 LED를 켜고 초록 LED를 끈 상태로 4초간 유지합니다.

```
void loop() {
  digitalWrite(red, HIGH);
  digitalWrite(green, LOW);
  delay(4000);
```

6 신호등 LED – 신호등 구현

- 노랑 LED 만 켜고 빨강 LED는 끈 상태로 1초간 유지합니다.
- 노랑 LED를 끄고 초록 LED를 끈 상태로 1초간 유지합니다.

```
digitalWrite(red, LOW);
digitalWrite(yellow, HIGH);
delay(1000);
digitalWrite(yellow, LOW);
digitalWrite(green, HIGH);
delay(1000);
```

6 신호등 LED – 신호등 구현

초록 LED를 끈 상태로 0.2초, 켠 상태로 0.2초 유지하기를 5회 반복합니다.

```
for(int i=0 ; i<5 ; i++) {
    digitalWrite(green, LOW);
    delay(200);
    digitalWrite(green, HIGH);
    delay(200);
  }
}
```

6 신호등 LED – 신호등 구현

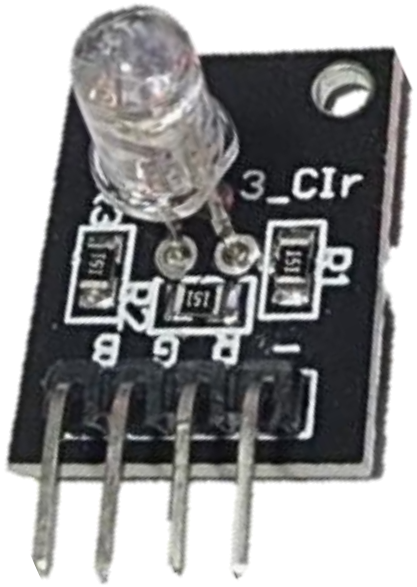
정답 예시 코드

```
int red    = 13;
int yellow = 12;
int green  = 14;

void setup() {
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}
```

```
void loop() {
  digitalWrite(red, HIGH);
  digitalWrite(green, LOW);
  delay(4000);
  digitalWrite(red, LOW);
  digitalWrite(yellow, HIGH);
  delay(1000);
  digitalWrite(yellow, LOW);
  digitalWrite(green, HIGH);
  delay(1000);
```

```
  for(int i=0 ; i<5 ; i++) {
    digitalWrite(green, LOW);
    delay(200);
    digitalWrite(green, HIGH);
    delay(200);
  }
}
```

3색 LED 제어



D1	3색 LED	D2	3색 LED(SMD)	D3	레이저송신	D4	2색 LED	D5	2색 LED(소)	AD6	터치센서
D7	수동 버저	D8	능동 버저	D9	7색 점멸 LED	D10	적외선송신	D11	릴레이	AD12	리드센서
D13	버튼(스위치)	D14	미니 리드센서	D15	틸트센서(수은)	D16	틸트센서(볼)	D17	충격센서	AD18	불꽃감지센서
D19	노크센서	D20	포토인터럽트	D21	적외선수신	D22	온도(DS18b20)	D23	온습도(DHT11)	AD24	리니어홀센서
D25	마그네틱홀센서	D26	IR트래킹센서	D27	적외선거리센서	D28	로터리 엔코더	D29	매직라이트컵	AD30	사운드센서
A31	빛센서(CdS)	A32	온도센서(NTC)	A33	자기장센서	A34	심장박동센서	A35	조이스틱모듈	AD36	고감도사운드
										AD37	온도센서

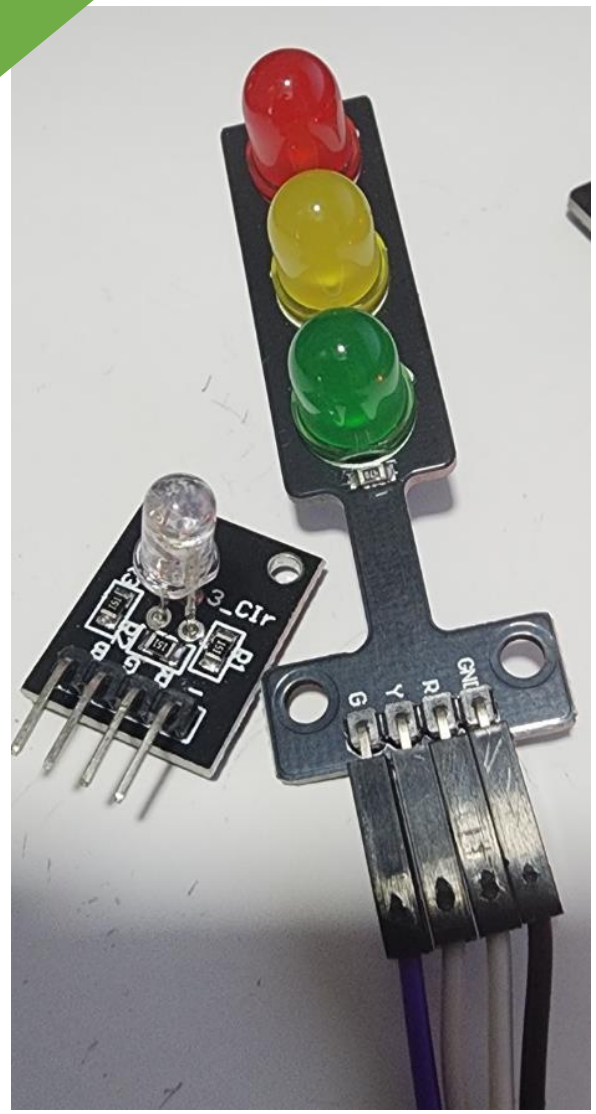
하쌤보드 HAN INNOVATION

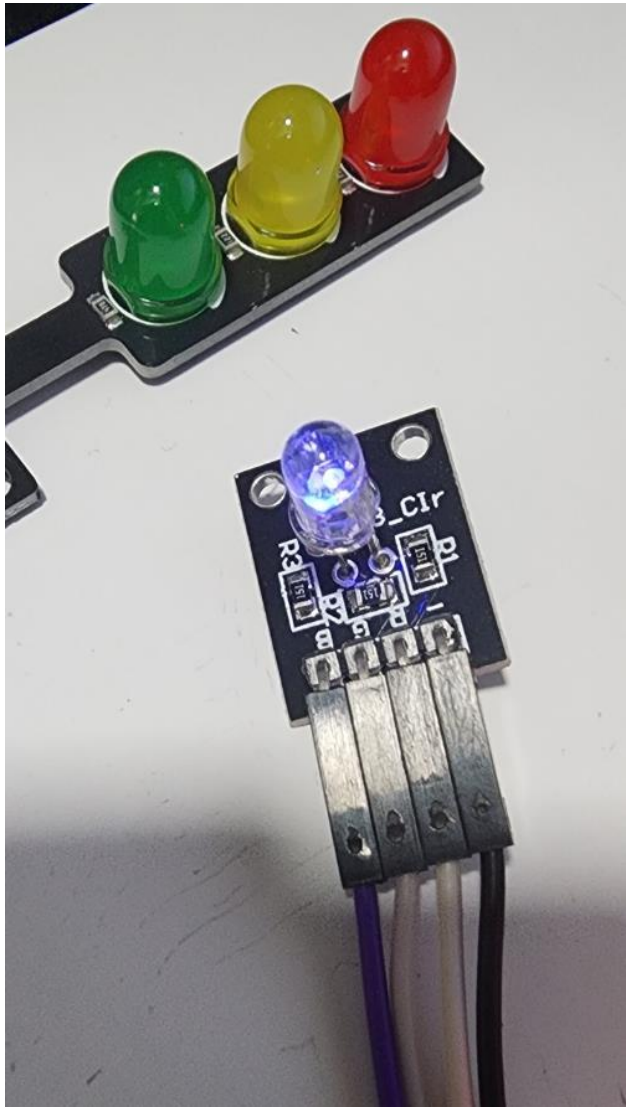
※ 초록색(D1~D11) : 디지털 출력장치
※ 노란색(D13~D29) : 디지털 센서(입력장치)
※ 분홍색(A31~A35) : 아날로그 센서(입력장치)
※ 붉은색(AD6~AD37) : 아날로그 or 디지털 출력

아두이노 **내부풀업** (INPUT_PULLUP)을 활성화 시켜야 사용 가능한 센서

센서 상자에서 3색 LED를 꺼냅니다.

3색 LED 제어





신호등 LED에 꼽았던 선을 뽑아 그대로 3색 LED에 꼽는다.

R은 위치가 같고 yellow와 green이 green과 blue로 바뀌었으므로 변수명을 다음과 같이 변경한다.

yellow	green
green	blue

3색 LED 제어

앞서 코딩했던 신호등 led 제어 코드를 불러옵니다.

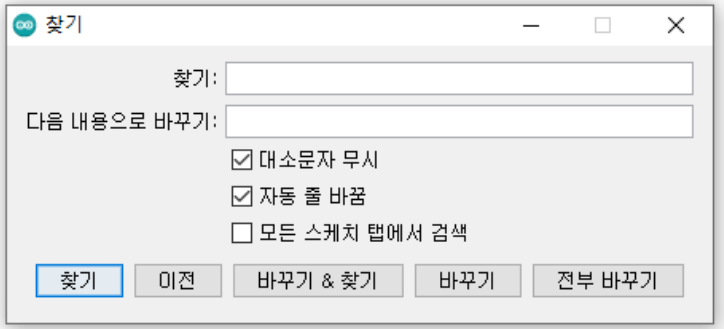
ctrl + f를 눌러 글자를 전부 바꾸기로 변환합니다.

green → blue

yellow → green

05_traffic_light

```
int red    = 13;
int yellow = 12;
int green  = 14;

void setup() {
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}

void loop() {
  digitalWrite(red, HIGH);
  digitalWrite(yellow, LOW);
  digitalWrite(green, LOW);
  delay(500);
  digitalWrite(red, LOW);
  digitalWrite(yellow, HIGH);
  digitalWrite(green, LOW);
  delay(500);
  digitalWrite(red, LOW);
  digitalWrite(yellow, LOW);
```



3색 LED 제어

코딩 예제

```
int red   = 13;
int green = 12;
int blue  = 14;

void setup() {
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(blue, OUTPUT);
}
```

```
void loop() {
  digitalWrite(red, HIGH);
  digitalWrite(green, LOW);
  digitalWrite(blue, LOW);
  delay(500);
  digitalWrite(red, LOW);
  digitalWrite(green, HIGH);
  digitalWrite(blue, LOW);
  delay(500);
```

```
  digitalWrite(red, LOW);
  digitalWrite(green, LOW);
  digitalWrite(blue, HIGH);
  delay(500);
}
```

8 3색 LED – 7가지 색

아래의 패턴이 0.5초 간격으로 일어날 수 있도록 해 보자

빨강 → 빨강+초록 → 초록 → 초록+파랑 → 파랑 → 파랑 +빨강 → 빨강+초록+파랑

※ 햇빛의 간섭으로 정확한 색 표현은 되지 않음

8 3색 LED – 7가지 색

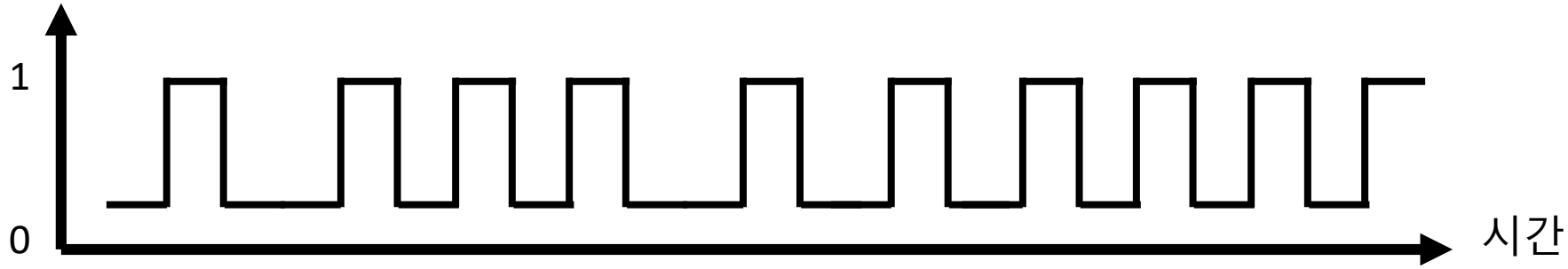
전체 코드

```
int red   = 13;
int green = 12;
int blue  = 14;
void setup() {
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(blue, OUTPUT);
}
void loop() {
  digitalWrite(red, HIGH);
  digitalWrite(green, LOW);
  digitalWrite(blue, LOW);
  delay(500);
  digitalWrite(red, LOW);
  digitalWrite(green, HIGH);
  digitalWrite(blue, LOW);
  delay(500);
  digitalWrite(red, LOW);
  digitalWrite(green, LOW);
  digitalWrite(blue, HIGH);
  delay(500);
  digitalWrite(red, HIGH);
  digitalWrite(green, LOW);
  digitalWrite(blue, HIGH);
  delay(500);
  digitalWrite(red, LOW);
  digitalWrite(green, HIGH);
  digitalWrite(blue, HIGH);
  delay(500);
}
```

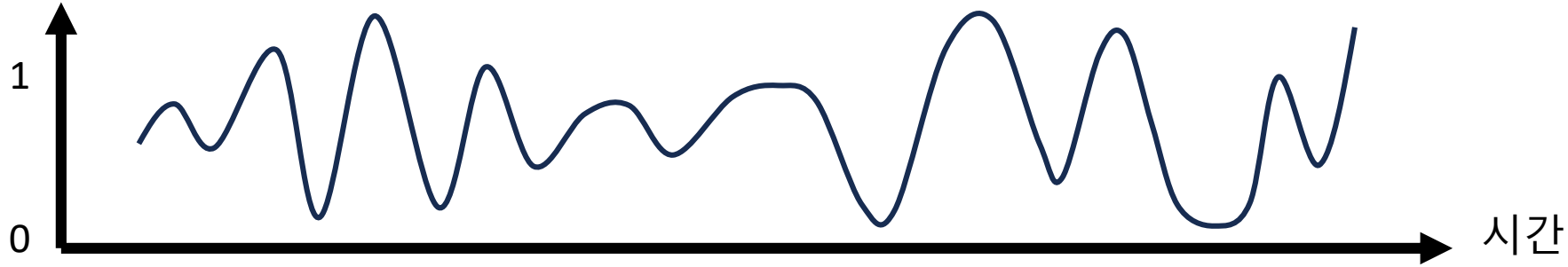
2. PWM 출력

1 PWM 출력이란

디지털 : TRUE(1), FALSE(0) 두 가지 값만 가진 비연속적인 수



아날로그 : 시간에 따라 다양한 값으로 바뀌는 연속적인 수



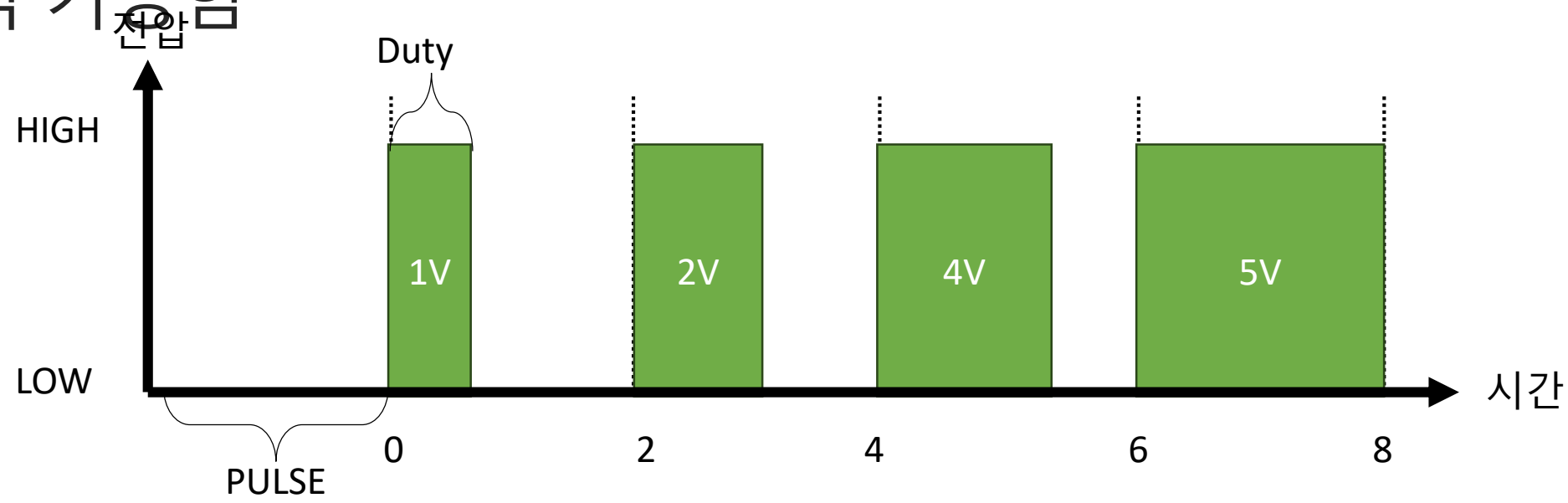
1 PWM 출력이란

ESP32 → PWM을 이용하여 아날로그 출력처럼 사용함

※ 34, 35, 36, 39 를 제외한 모든 핀에서 PWM을 이용한 아날로그 형태의 출력이 가능함

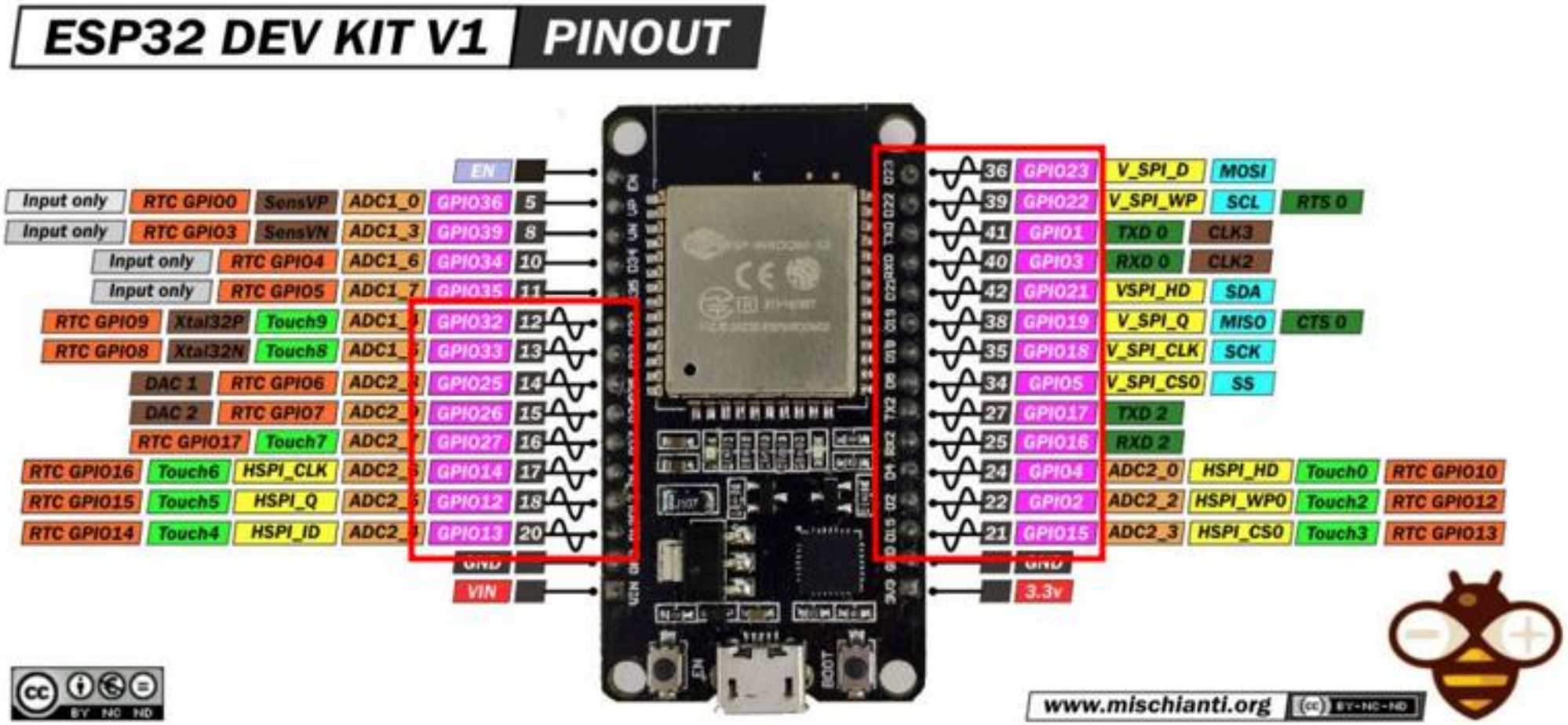
PWM은 펄스 폭 변조(Pulse Width Modulation)의 약자로 전압 지속의 폭을 조절한다는 의미

시간에 따라 HIGH(3.3V)와 LOW(0V)의 비율(Duty Cycle)을 조절하여 아날로그처럼 출력 가능함



1 PWM 출력이란

ESP32에서는 물결 표시가 된 GPIO 핀에서 PWM 출력이 가능하다



2 3색 LED의 PMW 제어

LED 채널 설정

```
void setup() {

  ledcSetup(0, 5000, 8);

  ledcSetup(1, 5000, 8);

  ledcSetup(2, 5000, 8);
```

2 3색 LED의 PMW 제어

LED 핀 번호와 채널 연결

```
  ledcAttachPin(13, 0);

  ledcAttachPin(12, 1);

  ledcAttachPin(14, 2);

}
```

2 3색 LED의 PMW 제어

채널 만들기(ESP32에서는 analogWrite 대신 채널을 사용함)

※ 채널 세팅 후 핀 번호를 채널에 붙이는 방식

코딩 예제

```
void setup() {
  ledcSetup(0, 5000, 8);     // 0번 채널에 5000Hz 8비트 세팅
  ledcSetup(1, 5000, 8);     // 1번 채널에 5000Hz 8비트 세팅
  ledcSetup(2, 5000, 8);     // 2번 채널에 5000Hz 8비트 세팅
  ledcAttachPin(13, 0);    // 13번 핀을 0번 채널로 설정
  ledcAttachPin(12, 1);    // 12번 핀을 1번 채널로 설정
  ledcAttachPin(14, 2);    // 14번 핀을 2번 채널로 설정
}
```

2 3색 LED의 PMW 제어

ledcWrite(채널 번호, 0~255)

코딩 예제

```
void loop() {
  ledcWrite(0, 107);     // 빨강 밝기 107
  ledcWrite(1, 219);     // 초록 밝기 219
  ledcWrite(2, 197);     // 파랑 밝기 197
}
```

2 3색 LED의 PMW 제어

setColor 함수에 107, 219, 197 을 매개 변수로 입력

```
void loop() {
  setColor(107, 219, 197);
}
```

2 3색 LED의 PMW 제어

- setColor 함수의 매개변수로 redValue, greenValue, blueValue 사용
- redValue 값으로 0번 채널, greenValue 값으로 1번 채널, blueValue 값으로 2번 채널에 밝기 출력

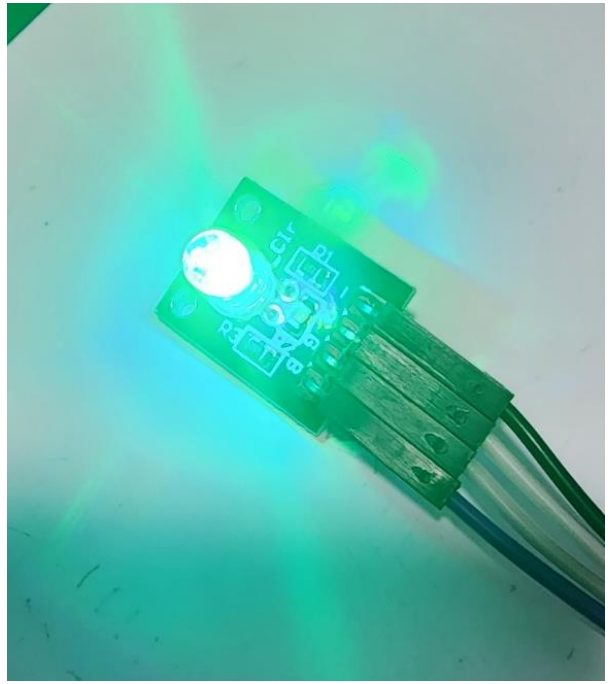
```
void setColor(int redValue, int
 greenValue, int blueValue) {
    ledcWrite(0, redValue);
    ledcWrite(1, greenValue);
    ledcWrite(2, blueValue);
}
```

2 3색 LED의 PMW 제어

함수 이용 가능

코딩 예제

```
void loop() {
  setColor(107, 219, 197);
}


void setColor(int redValue, int greenValue, int blueValue) {
    ledcWrite(0, redValue);
    ledcWrite(1, greenValue);
    ledcWrite(2, blueValue);
}
```

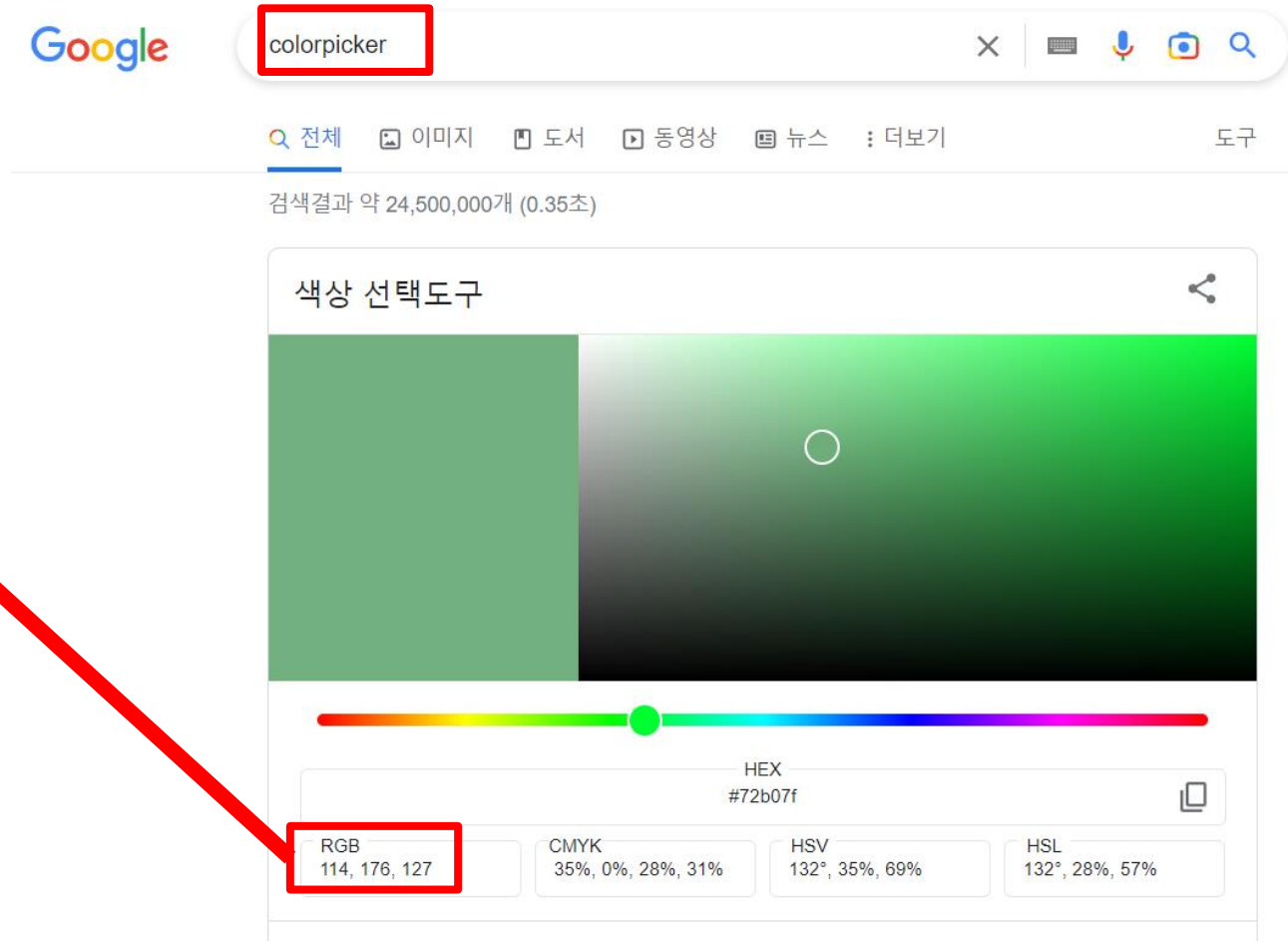


3 3색 LED의 7가지 색 표현

도전해 볼까요?

마음에 드는 7가지 색깔이 0.5초 간격으로 반복되어 나오도록 해봅시다.

colorpicker 검색



3 3색 LED의 7가지 색 표현

```
void setup() {
  ledcSetup(0, 500, 8);
  ledcSetup(1, 500, 8);
  ledcSetup(2, 500, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

- 채널 설정
- 핀과 채널 연결

3 3색 LED의 7가지 색 표현

setColor함수 호출로 7가지 색깔 표현

```
void loop() {
  setColor(255, 89, 43);
  delay(500);
  setColor(250, 164, 52);
  delay(500);
                :
                :
                :
```

첫 번째 색

두 번째 색

3 3색 LED의 7가지 색 표현

setColor 함수 정의

```
void setColor(int redValue, int
 greenValue, int blueValue) {
    ledcWrite(0, redValue);
    ledcWrite(1, greenValue);
    ledcWrite(2, blueValue);  }
```

3 3색 LED의 7가지 색 표현

전체 코드 예시

```
void setup() {
  ledcSetup(0, 5000, 8);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

```
void loop() {
  setColor(255, 89, 43);
 delay(500);
 setColor(250, 164, 52);
 delay(500);
 setColor(236, 255, 150);
 delay(500);
 setColor(12, 240, 58);
 delay(500);
 setColor(150, 163, 250);
 delay(500);
 setColor(213, 177, 240);
 delay(500);
 setColor(232, 12, 151);
 delay(500);
}

void setColor(int redValue,
int greenValue, int blueValue)
{   ledcWrite(0, redValue);
    ledcWrite(1, greenValue);
    ledcWrite(2, blueValue);  }
```

4 3색 LED의 무작위 색 표현

도전해 볼까요?

색깔이 0.5초 간격으로 무작위로 섞여 나올 수 있도록 하기

4 3색 LED의 무작위 색 표현

난수란 = 무작위 수

난수를 생성하는 내장 함수 rand()

rand() 함수에 의해 생성되는 난수 : 0 ~ 32767

4 3색 LED의 무작위 색 표현

```
void setup() {
  ledcSetup(0, 500, 8);
  ledcSetup(1, 500, 8);
  ledcSetup(2, 500, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

- 채널 설정
- 핀과 채널 연결

4 3색 LED의 무작위 색 표현

```
void loop() {
  int r = rand()%256;
  int g = rand()%256;
  int b = rand()%256;
  setColor(r, g, b);
  delay(500);
}
```

- 0~32767 사이의 난수를 256으로 나눈 나머지 값인 0~255중 하나의 숫자를 r, g, b 변수에 입력
- setColor 함수로 빛 출력
- 0.5초간 유지 후 계속 반복

4 3색 LED의 무작위 색 표현

setColor 함수 정의

```
void setColor(int redValue, int
 greenValue, int blueValue) {
    ledcWrite(0, redValue);
    ledcWrite(1, greenValue);
    ledcWrite(2, blueValue);  }
```

4 3색 LED의 무작위 색 표현

도전해 볼까요?

rand() 함수에 의해 생성되는 난수 : 0 ~ 32767

난수 활용 코딩 예제

```
void setup() {
  ledcSetup(0, 500, 8);
  ledcSetup(1, 500, 8);
  ledcSetup(2, 500, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

```
void loop() {
int r = rand()%256;
int g = rand()%256;
int b = rand()%256;
  setColor(r, g, b);
 delay(500);
}
```

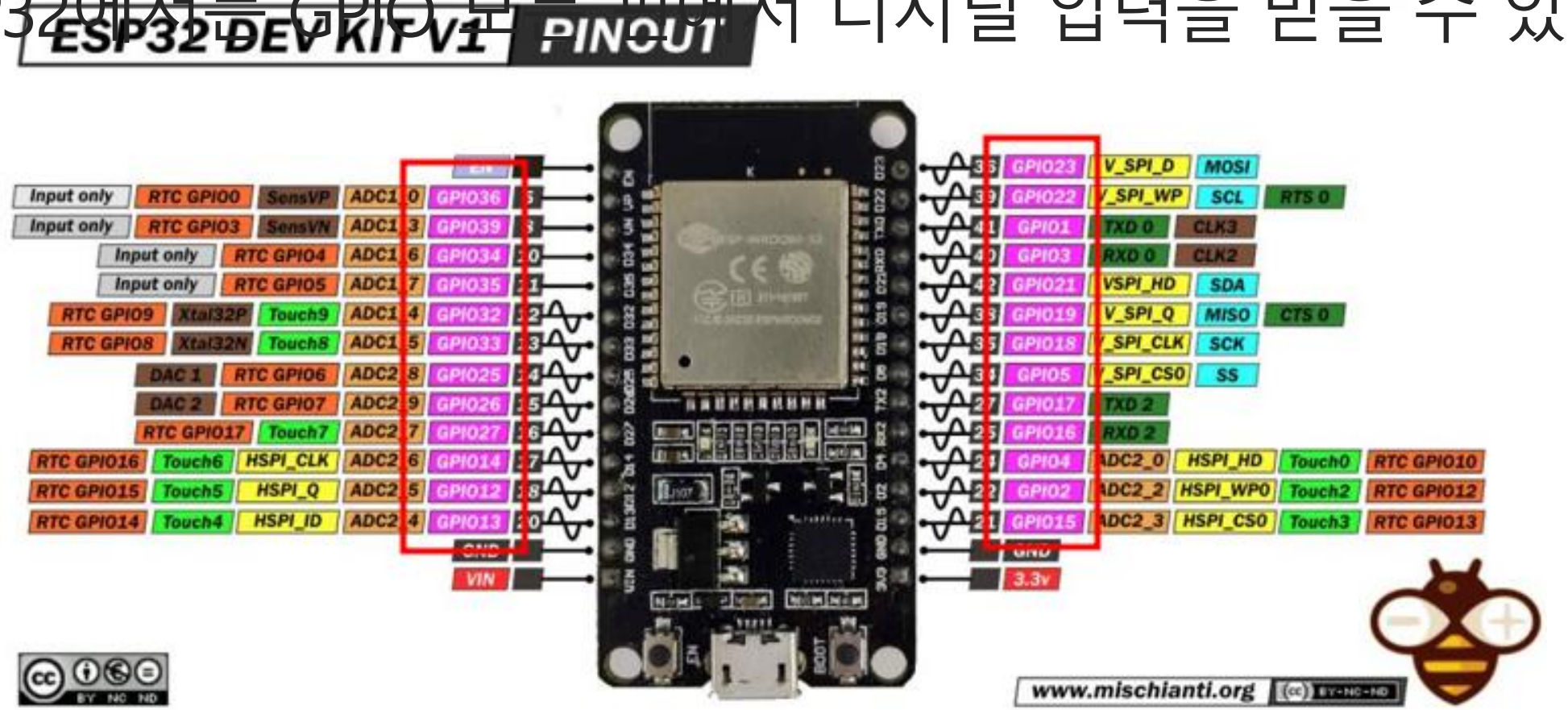
```
void  setColor(int  redValue,
int greenValue, int blueValue)
{
   ledcWrite(0, redValue);
    ledcWrite(1, greenValue);
    ledcWrite(2, blueValue);
 }
```

3. 디지털 입력

1 디지털 입력이란

디지털 입력이란 0과 1로 된 비연속적인 값을 받는 것을 말합니다.

ESP32에서는 GPIO 모든 핀에서 디지털 입력을 받을 수 있습니다.



2 시리얼 통신이란

시리얼 = 직렬

시리얼 통신은 데이터를 하나씩 순차적으로 전송하는 통신 방법

32비트의 데이터를
순차적으로 발송

1
0
1
0
1
1
1
0
0
1
0
1
0
0
1
0
0
0
1
1
0
1
0
1
0
1
0
1

10101110010100100100011010101010101

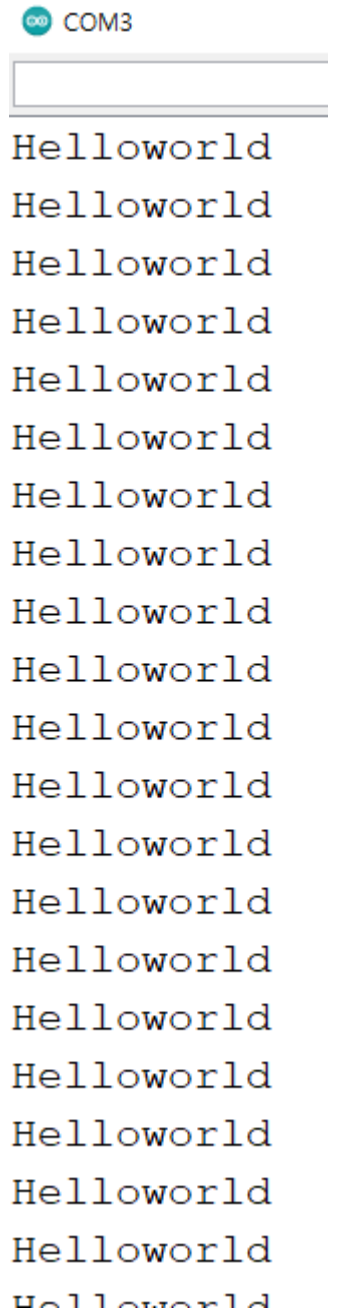
최소 1개의 통신선이 필요
함

2 시리얼 통신이란

코딩 예제

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print("Hello");
  Serial.println("world");
}
```

9600 baud(보드레이트) 속도로 시리얼 통신 시작



3 버튼 연결

D1	3색 LED	D2	3색 LED(SMD)	D3	레이저송신	D4	2색 LED	D5	2색 LED(소)	AD6	터치센서
D7	수동 버저	D8	능동 버저	D9	7색 점멸 LED	D10	적외선송신	D11	릴레이	AD12	리드센서
D13	버튼(스위치)	D14	미니 리드센서	D15	틸트센서(수은)	D16	틸트센서(볼)	D17	충격센서	AD18	불꽃감지센서
D19	노크센서	D20	포토인터럽트	D21	적외선수신	D22	온도(DS18b20)	D23	온습도(DHT11)	AD24	리니어홀센서
D25	마그네틱홀센서	D26	IR트래킹센서	D27	적외선거리센서	D28	로터리 엔코더	D29	매직라이트컵	AD30	사운드센서
A31	빛센서(CdS)	A32	온도센서(NTC)	A33	자기장센서	A34	심장박동센서	A35	조이스틱모듈	AD36	고감도사운드
										AD37	온도센서

허쌤보드 HAN INNOVATION

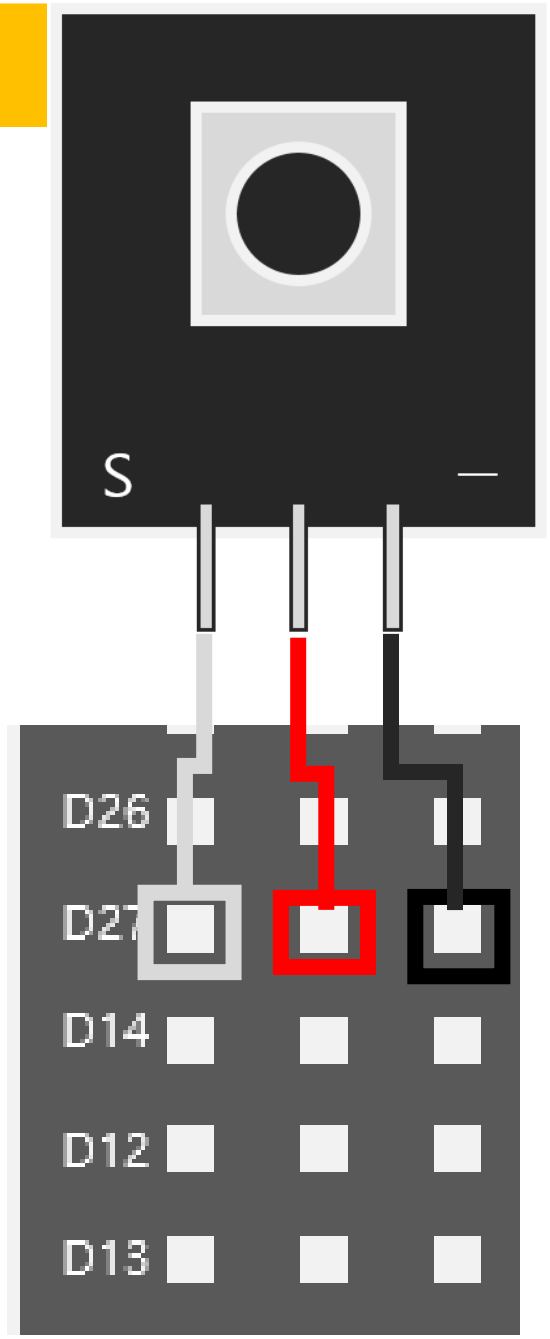
※ 초록색(D1~D11) : 디지털 출력장치
※ 노란색(D13~D29) : 디지털 센서(입력장치)
※ 분홍색(A31~A35) : 아날로그 센서(입력장치)
※ 붉은색(AD6~AD37) : 아날로그 or 디지털 출력

아두이노 내부풀업(INPUT_PULLUP)을 활성화 시켜야 사용 가능한 센서

센서 상자에서 버튼 모듈을 꺼냅니다.

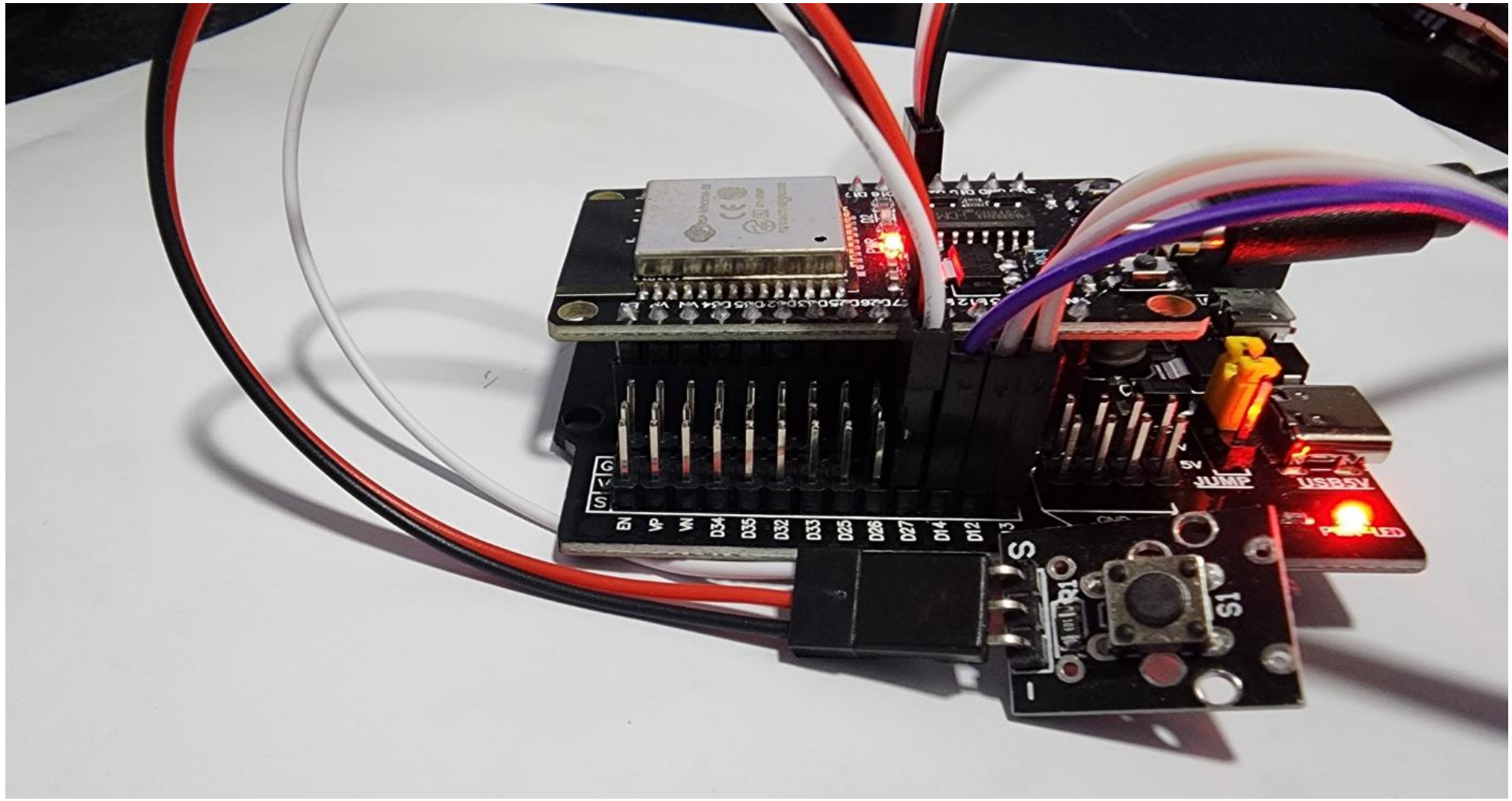
3 버튼 연결

버튼 모듈을 D27번에 연결

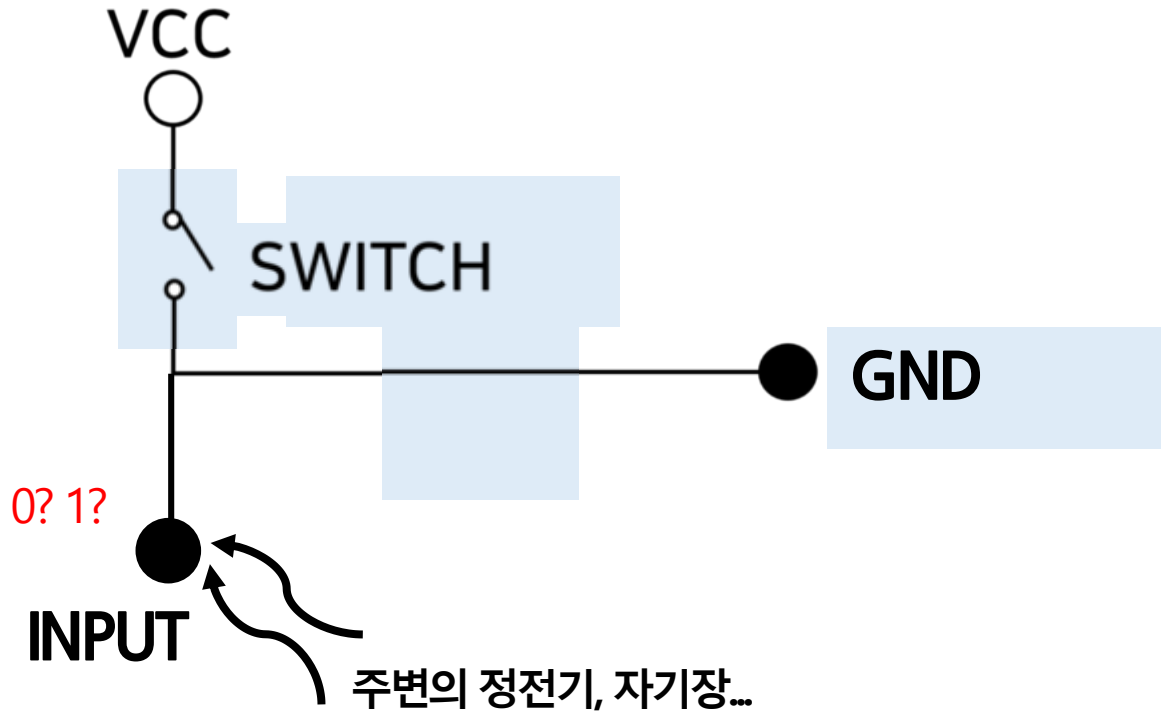


3 버튼 연결

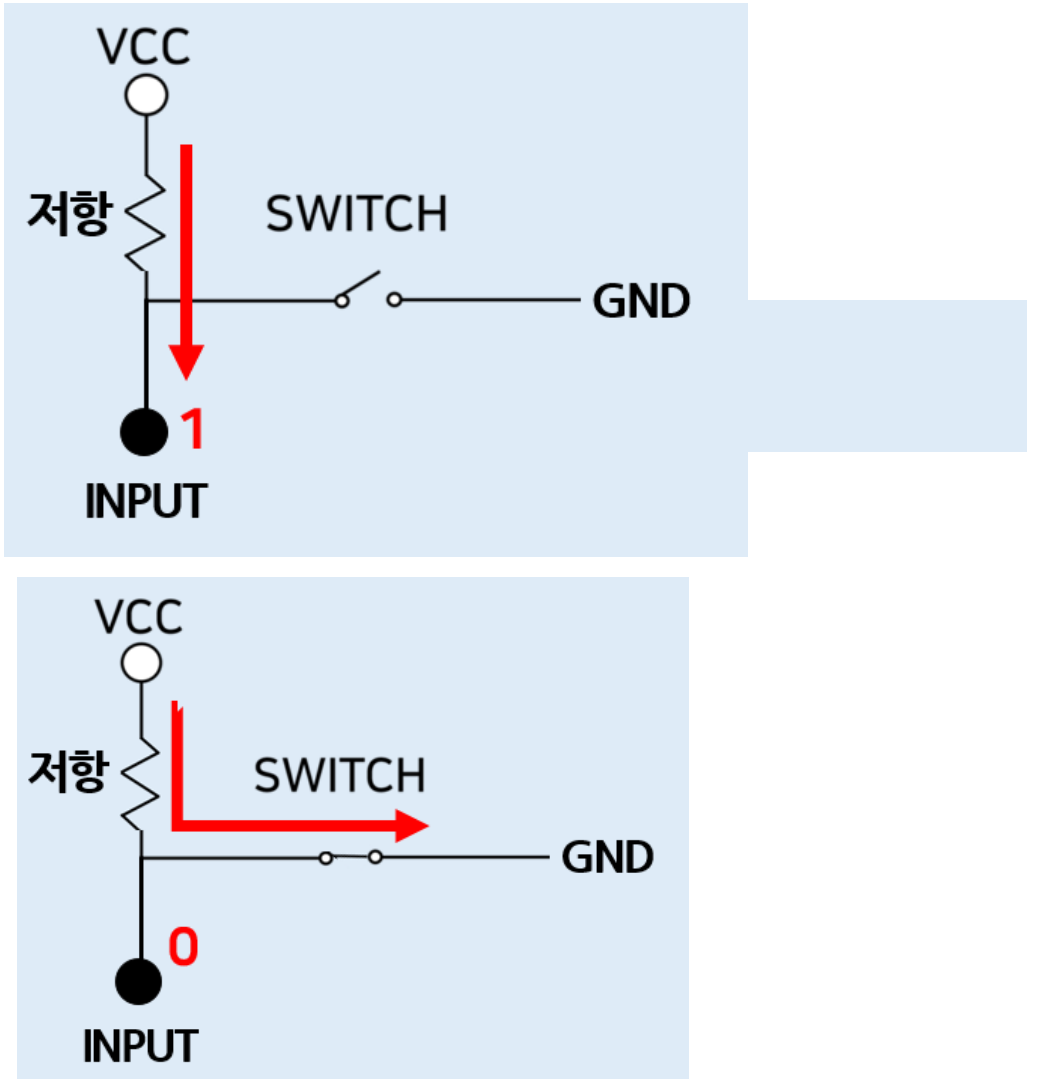
연결 모습



3 버튼



플로팅 현상 발생



풀업 저항

4 버튼 – 시리얼 통신 출력

시리얼 통신 시작
27번 핀을 풀업 디지털 입력으로 설정

```
void setup() {

Serial.begin(9600);

pinMode(27, INPUT_PULLUP);

}
```

4 버튼 – 시리얼 통신 출력

시리얼 통신 시작
27번 핀을 풀업 디지털 입력으로 설정

```
void loop() {

int button = digitalRead(27);

Serial.println(button);

delay(100);}
```

4 버튼 – 시리얼 통신 출력

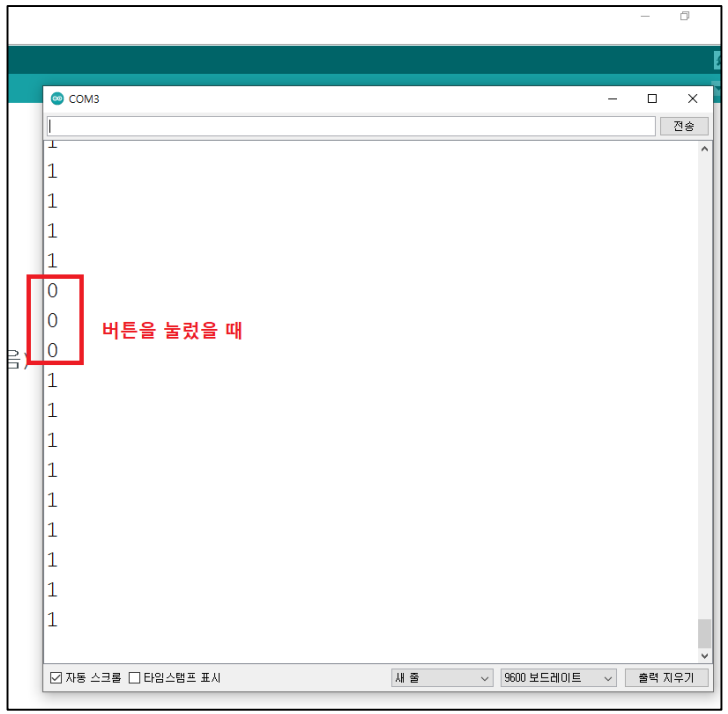
코딩 예제

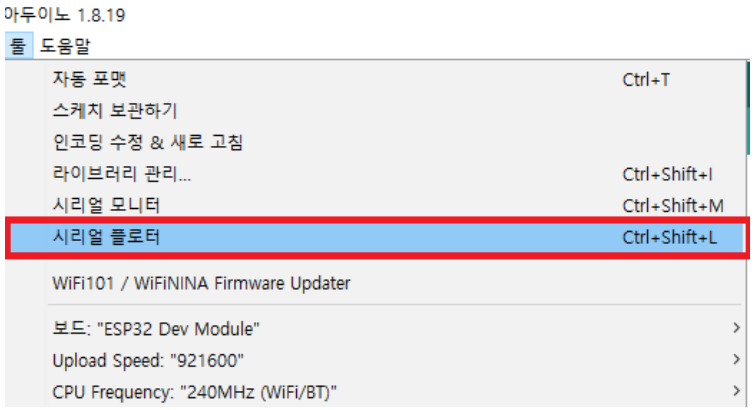
```
void setup() {
Serial.begin(9600);                    //아두이노-PC 통신시작(9600 속도)
pinMode(27, INPUT_PULLUP);             //버튼연결 27핀을 풀업 입력으로 사용
}
void loop() {
int button = digitalRead(27);          //27핀의 값을 읽어서 변수에 넣음
Serial.println(button);                //센서의 값(변수)을 PC로 보냄
delay(100);                            //0.1초 대기(0.1초에 1회 센서 읽음)
}
```

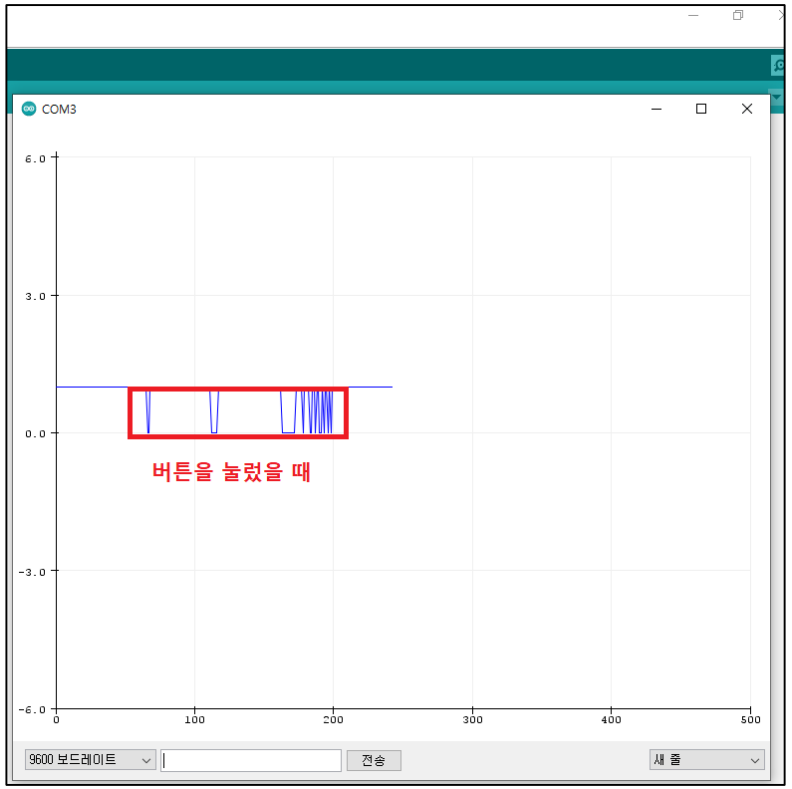
4 버튼 – 시리얼 통신 출력

시리얼 모니터와 시리얼 플로터로 버튼의 풀업 저항 결과 확인

버튼 누르면 → 0, 버튼 안 누르면 → 1







5 버튼 – LED 제어

27번 핀을 풀업 디지털 입력으로 설정
2번 핀을 출력으로 설정

```
void setup() {
    pinMode(27, INPUT_PULLUP);
    pinMode(2,OUTPUT);
}
```

5 버튼 – LED 제어

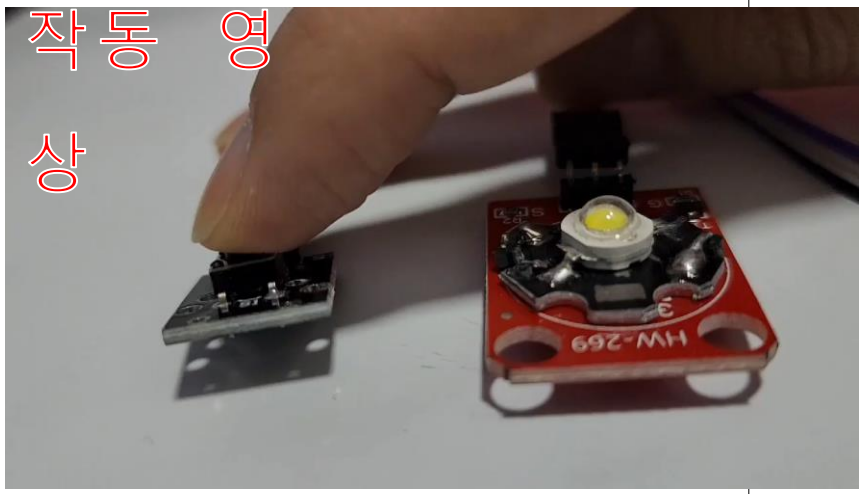
버튼으로 LED 제어. 누르지 않으면 꺼지고 누르면 켜짐

```
void loop() {
    int button = digitalRead(27);
    if(button==1){
        digitalWrite(2,0);
    }else{
      digitalWrite(2,1);
       }
```

5 버튼 – LED 제어

코딩 예제

```
void setup() {
   pinMode(27, INPUT_PULLUP);              //27핀 버튼
   pinMode(2,OUTPUT);                      //2번 핀 3W LED
}
void loop() {
   int button = digitalRead(27);
   if(button==1){
       digitalWrite(2,0);                  // 버튼을 안 누르면 꺼진다.
   }else{
     digitalWrite(2,1);                    // 버튼을 누르면 켜진다.
      }
}
```



6 버튼 – 3색 LED 무작위 표현

도전해 볼까요?

3색 LED 버튼 제어

1. 버튼을 누를 때마다 3색 LED 색깔이 랜덤으로 바뀌게 하기

6 버튼 – 3색 LED 무작위 표현

```
void setup() {
   pinMode(27, INPUT_PULLUP);
   ledcSetup(0, 500, 8);
   ledcSetup(1, 500, 8);
   ledcSetup(2, 500, 8);
   ledcAttachPin(13, 0);
   ledcAttachPin(12, 1);
   ledcAttachPin(14, 2);
}
```

- 27번 핀 입력 설정
- 채널 설정
- 핀과 채널 연결

6 버튼 – 3색 LED 무작위 표현

```
void loop() {
 int button = digitalRead(27);
   if(button==0){
    int r = rand()%256;
    int g = rand()%256;
    int b = rand()%256;
    setColor(r, g, b);
    delay(500);
   } else { setColor(0, 0, 0); }
```

버튼을 누르면 r, g, b 값을 랜덤으로 섞어서 setColor함수로 표현하기

6 버튼 – 3색 LED 무작위 표현

setColor 함수 정의

```
void setColor(int redValue, int
 greenValue, int blueValue) {
    ledcWrite(0, redValue);
    ledcWrite(1, greenValue);
    ledcWrite(2, blueValue);  }
```

6 버튼 – 3색 LED 무작위 표현

코딩 예제

```
void setup() {
  pinMode(27, INPUT_PULLUP);
  ledcSetup(0, 5000, 8);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

```
void loop() {
  int button = digitalRead(27);
  if(button==0){
   int r = rand()%256;
   int g = rand()%256;
   int b = rand()%256;
   setColor(r, g, b);
   delay(500);
  }else{
    setColor(0, 0, 0);
    }
}
```

```
void setColor(int redValue, int greenValue, int blueValue)
{
  ledcWrite(0, redValue);
   ledcWrite(1, greenValue);
   ledcWrite(2, blueValue);
 }
```

틸트센서(수은)

D1	3색 LED	D2	3색 LED(SMD)	D3	레이저송신	D4	2색 LED	D5	2색 LED(소)	AD6	터치센서
D7	수동 버저	D8	능동 버저	D9	7색 점멸 LED	10	적외선송신	D11	릴레이	AD12	리드센서
D13	버튼(스위치)	D14	미니 리드센서	D15	틸트센서(수은)	16	틸트센서(볼)	D17	충격센서	AD18	불꽃감지센서
D19	노크센서	D20	포토인터럽트	D21	적외선수신	D22	온도(DS18b20)	D23	온습도(DHT11)	AD24	리니어홀센서
D25	마그네틱홀센서	D26	IR트래킹센서	D27	적외선거리센서	D28	로터리 엔코더	D29	매직라이트컵	AD30	사운드센서
A31	빛센서(CdS)	A32	온도센서(NTC)	A33	자기장센서	A34	심장박동센서	A35	조이스틱모듈	AD36	고감도사운드
										AD37	온도센서

허쌤보드 HAN INNOVATION

※ 초록색(D1~D11) : 디지털 출력장치
※ 노란색(D13~D29) : 디지털 센서(입력장치)
※ 분홍색(A31~A35) : 아날로그 센서(입력장치)
※ 붉은색(AD6~AD37) : 아날로그 or 디지털 출력

아두이노 내부풀업(INPUT_PULLUP)을 활성화 시켜야 사용 가능한 센서

센서 상자에서 틸트센서(수은)을 꺼냅니다.

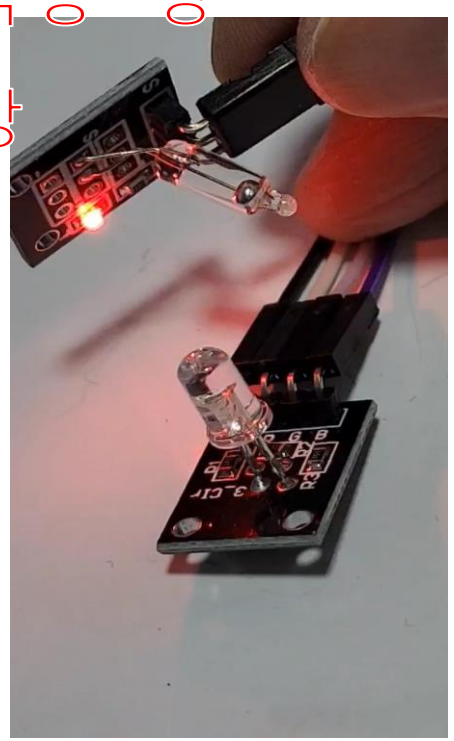
3. 디지털 입력

틸트센서(수은)

각도 스위치 모듈

- 37종 중 **D15**, 틸트 센서(수은)이라고도 함.
- 기울어짐을 판단하는 용도로 사용
- 예를 들어, 화분 또는 의자가 쓰러졌음을 판단하는 용도로 메이킹 가능
- 각도 스위치 모듈 2개를 서로 다른 방향으로 연결하면 상하좌우 4방향으로 기울어졌음을 판별 가능

작동 영상



버튼에서 선을 뽑고 그대로 각도 스위치 모듈에 꼽습니다.

8 틸트센서(볼)

D1 3색 LED	D2 3색 LED(SMD)	D3 레이저송신	D4 2색 LED	D5 2색 LED(소)	AD6 터치센서
D7 수동 버저	D8 능동 버저	D9 7색 점멸 LE	D10 적외선송신	11 릴레이	AD12 리드센서
D13 버튼(스위치)	D14 미니 리드센서	D15 틸트센서(수은	D16 틸트센서(볼)	17 충격센서	AD18 불꽃감지센서
D19 노크센서	D20 포토인터럽트	D21 적외선수신	D22 온도(DS18b20)	D23 온습도(DHT11)	AD24 리니어홀센서
D25 마그네틱홀센서	D26 IR트래킹센서	D27 적외선거리센서	D28 로터리 엔코더	D29 매직라이트컵	AD30 사운드센서
A31 빛센서(CdS)	A32 온도센서(NTC)	A33 자기장센서	A34 심장박동센서	A35 조이스틱모듈	AD36 고감도사운드
					AD37 온도센서

※ 초록색(D1~D11) : 디지털 출력장치
※ 노란색(D13~D29) : 디지털 센서(입력장치)
※ 분홍색(A31~A35) : 아날로그 센서(입력장치)
※ 붉은색(AD6~AD37) : 아날로그 or 디지털 출력

아두이노 내부풀업(INPUT_PULLUP)을 활성화 시켜야 사용 가능한 센서

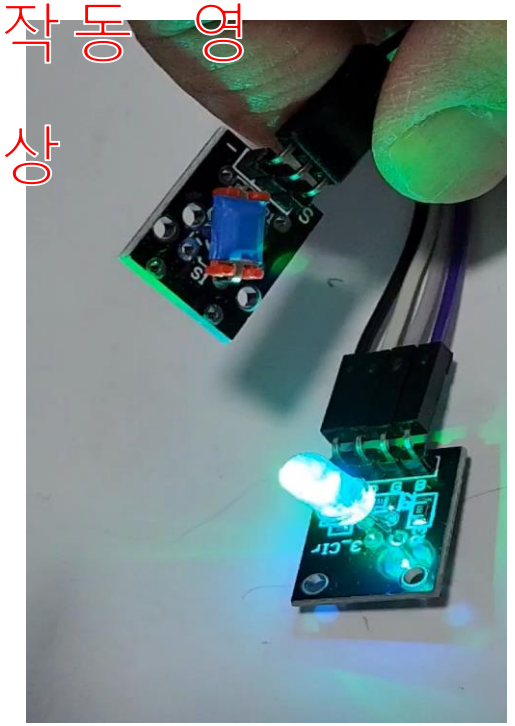
센서 상자에서 틸트센서(볼)를 꺼냅니다.

8 틸트센서(볼)

각도 스위치 모듈

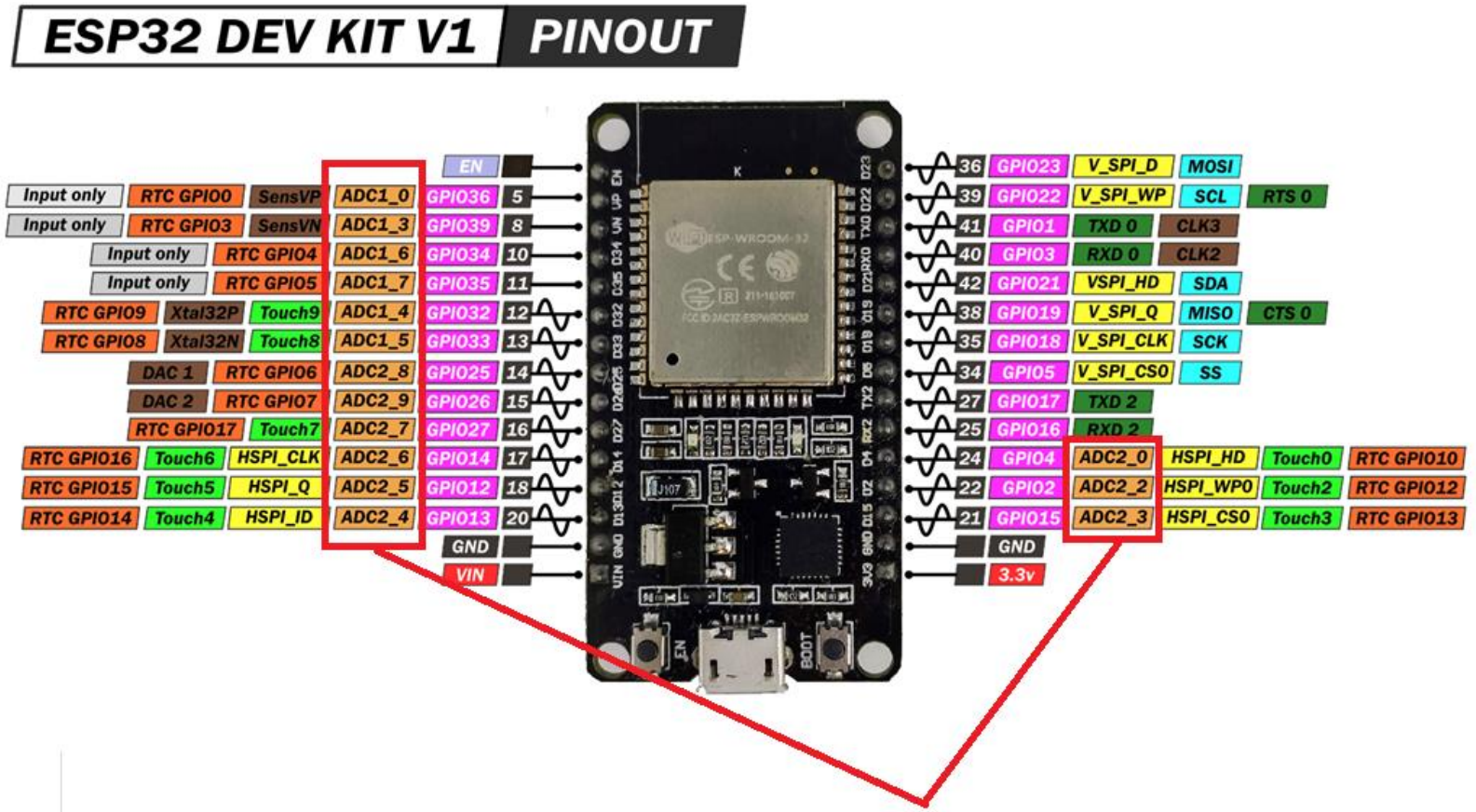
- 37종 중 **D16**, 앞의 각도 스위치 모듈과 차이는 없음.
- 각도 스위치 모듈은 수은방울이 움직이면서 동작하는데, 볼 스위치 모듈은 작은 금속 구슬이 구르면서 동작한다는 차이만 있음.

작동 영상



4. 아날로그 입력

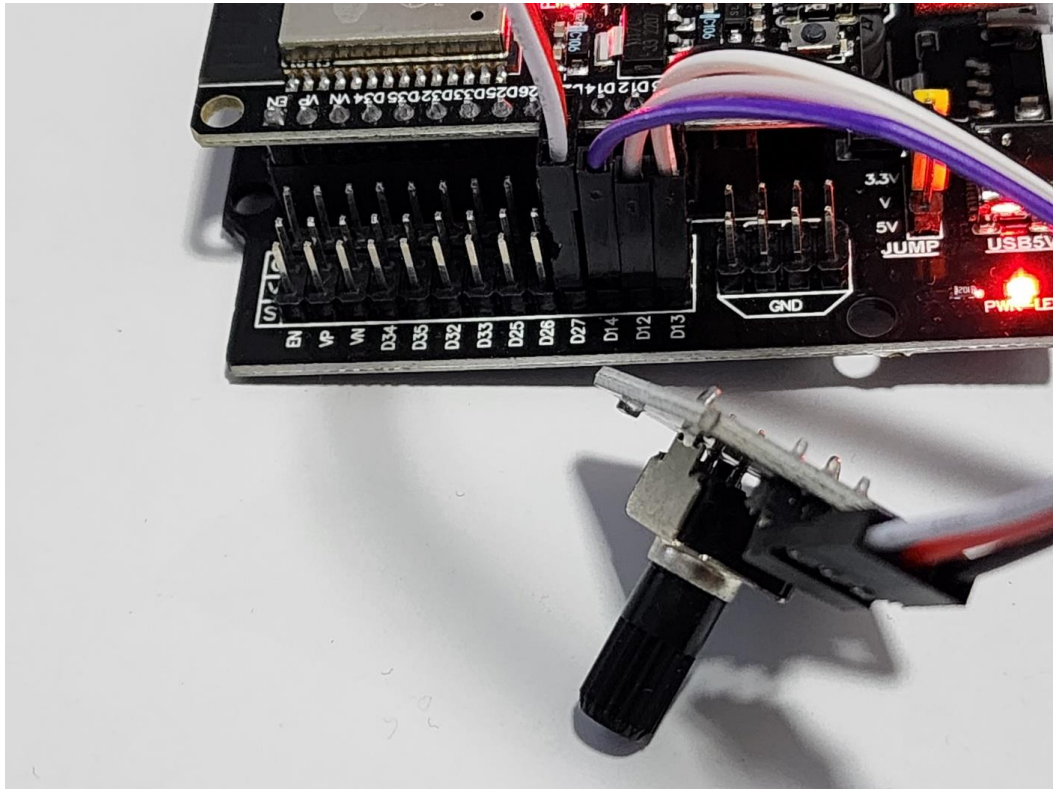
1 아날로그 입력



- ESP32에서 아날로그 입력은 0V~3.3V의 값을 구분
- 3.3V를 12비트 단계로 구분하여 값을 인식
- 핀맵 기준으로 ADC핀에서 가능

2 가변저항 테스트

회전부를 통해 저항값을 조절할 수 있음



틸트 센서를 뽑고 그대로 가변저항과 D27을 연결

2 가변저항 테스트

시리얼 통신 시작
27번 핀을 입력으로 설정(풀업 저항 아님)

```
void setup() {

Serial.begin(9600);

pinMode(27, INPUT);

}
```

2 가변저항 테스트

시리얼 통신 시작
27번 핀을 풀업 디지털 입력으로 설정

```
void loop() {

int sensor = digitalRead(27);

Serial.println(sensor);

delay(50);}
```

2 가변저항 테스트

코딩 예제

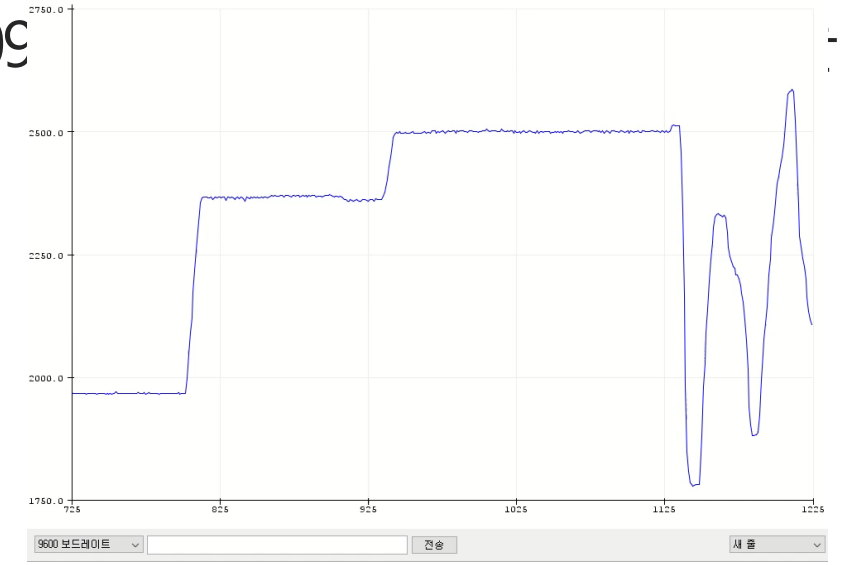
```
void setup() {
  Serial.begin(9600);
  pinMode(27, INPUT);
}

void loop() {
  int sensor = analogRead(27);
  Serial.println(sensor);
  delay(50);
}
```

* 가변저항은 포텐셔미터 또는 볼륨이라는 명칭으로도 불림

* 좌우로 다이얼을 돌리면 내부저항이 변하여 0V~3.3V의 값이 아두이노로 흘러 들어가게 됨

* 이 ~40 임

```
1104
1055
1008
999
1001
999
997
999
910
774
668
609
```



3 가변저항 – LED 제어

도전에 볼까요?

3W LED 제어

1. 가변저항을 사용하여 3W LED 밝기를 제어하자

3 가변저항 – LED 제어

0번 채널에 2번 핀 연결
27번 입력(풀업 저항 아님)

```
void setup() {
  ledcSetup(0, 5000, 8);
  ledcAttachPin(2, 0);
  pinMode(27, INPUT);
}
```

3 가변저항 – LED 제어

27번핀에서 아날로그 입력으로 읽은 값을 sensor 변수에 저장합니다.
센서 값으로 0번 채널의 밝기를 설정합니다.

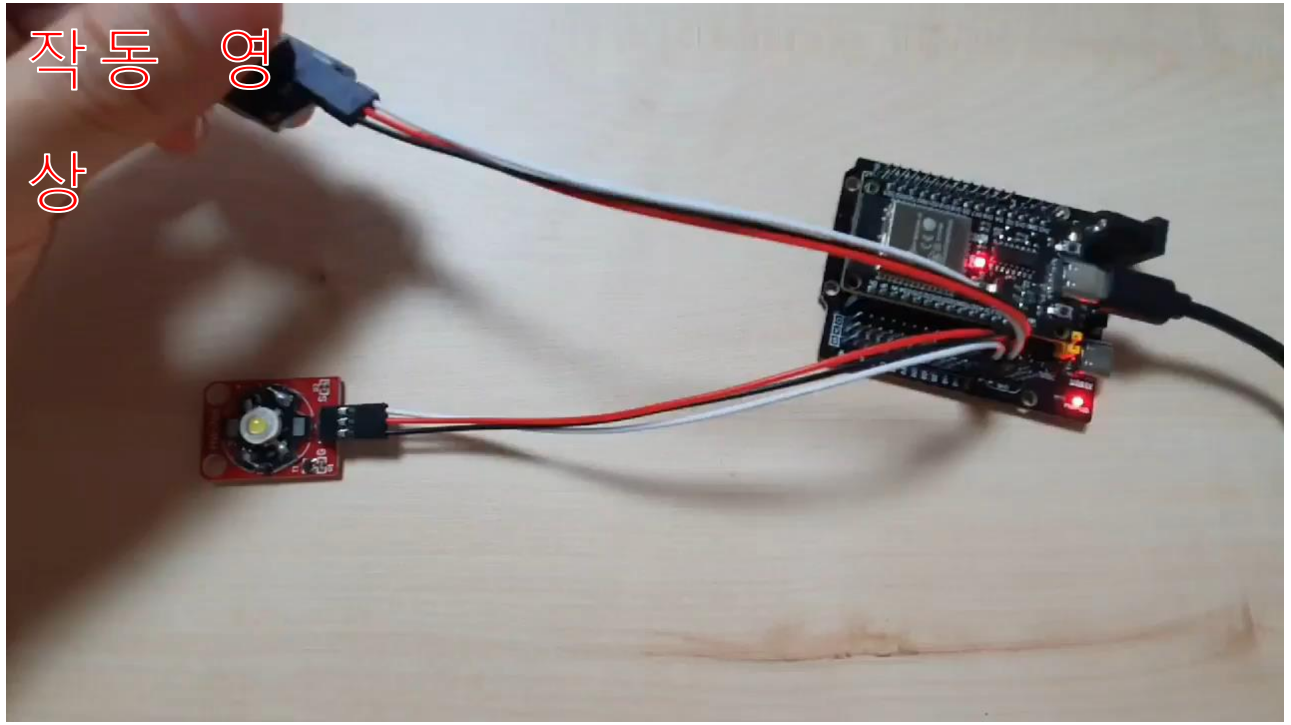
```
void loop() {
  int sensor = analogRead(27);
  ledcWrite(0, sensor);
  delay(50);
}
```

무엇이 틀렸을까요?

3 가변저항 – LED 제어

오답 코딩 예제

```
void setup() {
  ledcSetup(0, 5000, 8);
  ledcAttachPin(2, 0);
  pinMode(27, INPUT);
}

void loop() {
  int sensor = analogRead(27);
  ledcWrite(0, sensor);
  delay(50);
}
```



- LED에서 PWM은 0~255까지만 출력 가능
- 가변저항은 0~4095까지 입력
- 따라서 가변저항 값이 255를 넘어가면 PWM은 0이 됨

3 가변저항 – LED 제어

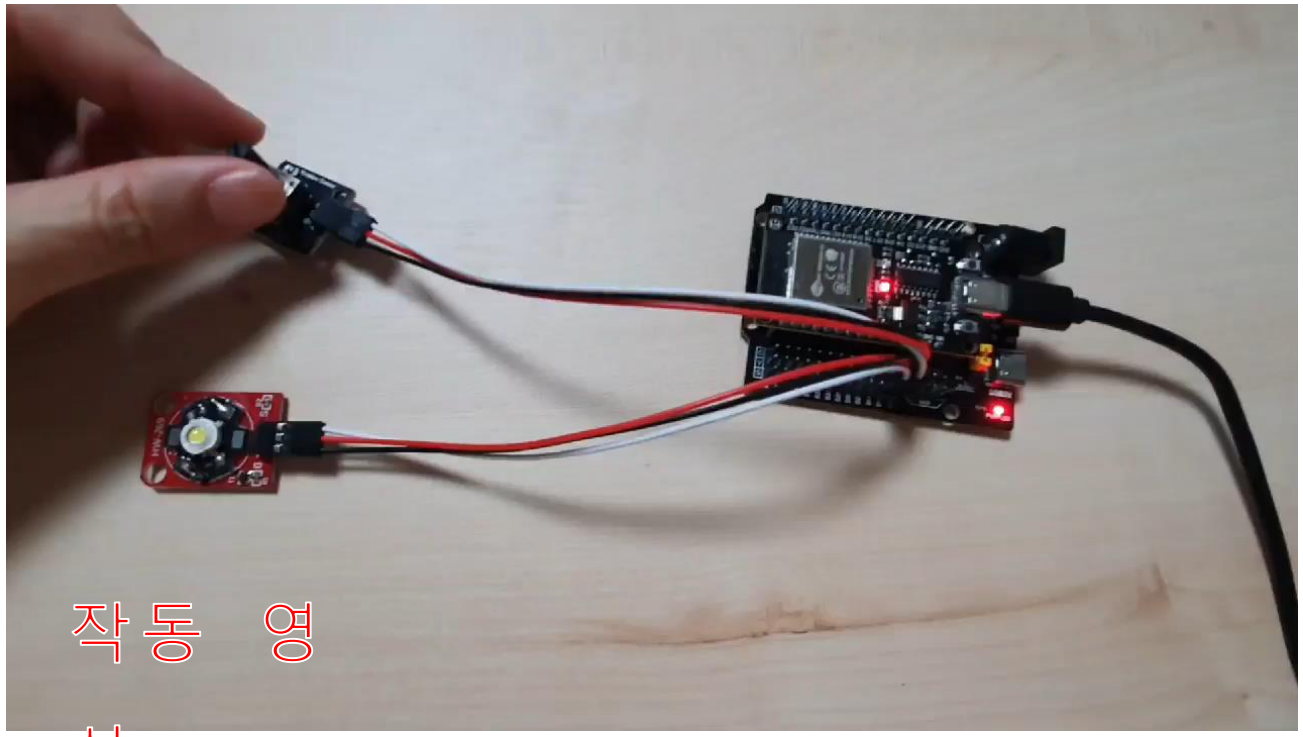
sensor값을 매핑함
map(매핑할 대상, 시작 값, 끝 값, 매핑한 시작 값, 매핑한 끝 값)

```
void loop() {
  int sensor = analogRead(27);
  sensor = map(sensor, 0, 4095, 0, 255);
  ledcWrite(0, sensor);
  delay(50);
}
```

3 가변저항 – LED 제어

정답 코딩 예제

```
void setup() {
  ledcSetup(0, 5000, 8);
  ledcAttachPin(2, 0);
  pinMode(27, INPUT);
}

void loop() {
  int sensor = analogRead(27);
  sensor = map(sensor, 0, 4095, 0, 255);
  ledcWrite(0, sensor);
  delay(50);
}
```



4 가변저항 – 3색 LED 제어

도전해 봅시다

가변 저항으로 3색 LED 제어

1. 가변 저항 값을 0~511으로 조정

2. 가변저항이 0이면 빨간빛, 0보다 크고 255보다 작거나 같다면
가변 저항 값이 커지는 동안 빨간빛은 점점 줄어들고, 초록빛이 점점 밝아진다.

3. 가변저항이 255보다 크고 511보다 작거나 같다면
가변 저항 값이 커지는 동안 초록빛은 점점 줄어들고, 파란빛이 점점 밝아진다.

4 가변저항 – 3색 LED 제어

```
void setup() {
   pinMode(27, INPUT);
   ledcSetup(0, 500, 8);
   ledcSetup(1, 500, 8);
   ledcSetup(2, 500, 8);
   ledcAttachPin(13, 0);
   ledcAttachPin(12, 1);
   ledcAttachPin(14, 2);
}
```

- 27번 핀 입력 설정
- 채널 설정
- 핀과 채널 연결

4 가변저항 – 3색 LED 제어

- 27번 핀의 아날로그 값을 받아 sensor에 입력
- 0~4095의 값을 0~511로 매핑하여 다시 sensor 값에 입력

```
void loop() {

  int sensor = analogRead(27);

  sensor = map(sensor, 0, 4095, 0, 511);
```

4 가변저항 – 3색 LED 제어

255보다 매핑한 센서값이 작거나 같으면
가변저항에서 전압 값을 증가시켰을 때 r값을 줄이고 g값을 늘림

```
if(sensor <= 255) {
    setColor(255-sensor, sensor, 0);
  }
```

4 가변저항 – 3색 LED 제어

그렇지 않으면 가변저항에서 전압 값을 증가시켰을 때
g값을 줄이고 b 값을 늘림

```
    else {
      setColor(0, 511-sensor, sensor-256);
    }
}
```

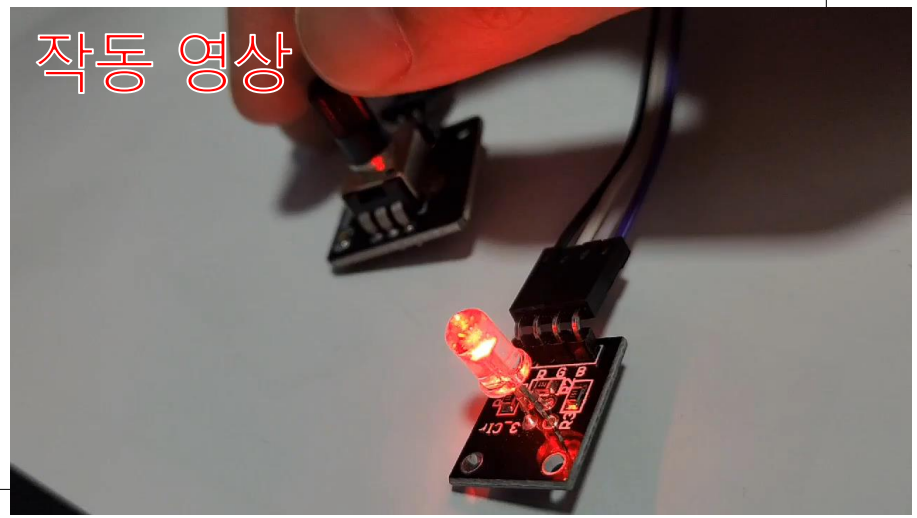
4 가변저항 – 3색 LED 제어

정답 코딩 예제

```
void setup() {
  pinMode(27, INPUT);
  ledcSetup(0, 500, 8);
  ledcSetup(1, 500, 8);
  ledcSetup(2, 500, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

```
void loop() {
  int sensor = analogRead(27);
  sensor = map(sensor, 0, 4095, 0, 511);
  if(sensor <= 255) {
   setColor(255-sensor, sensor, 0);
  } else {
   setColor(0, 511-sensor, sensor-256);
  }
}
```

```
void setColor(int redValue, int greenValue, int blueValue) {
  ledcWrite(0, redValue);
  ledcWrite(1, greenValue);
  ledcWrite(2, blueValue);
 }
```



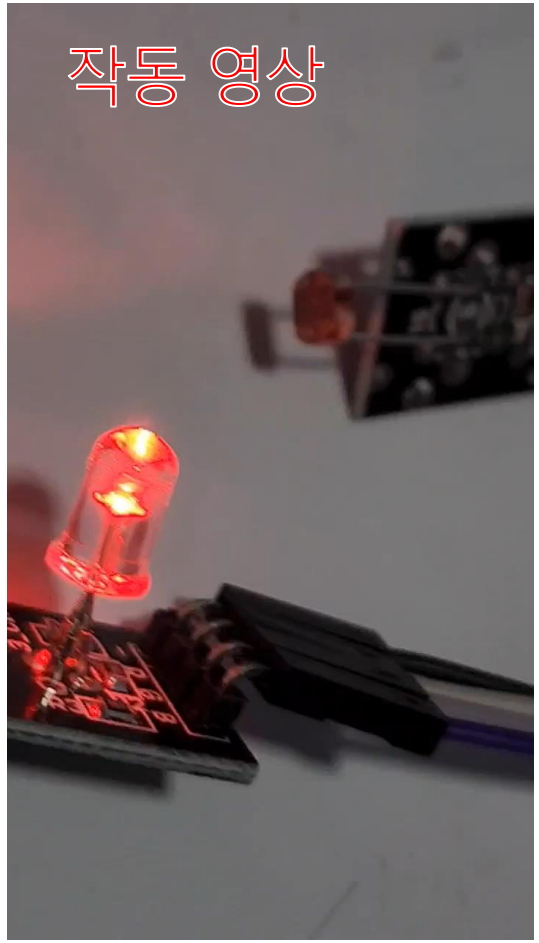
5 빛 센서(Cds)

센서 상자에서 빛센서(cds)를 꺼냅니다. 가변 저항과 교체합니다.

5 빛 센서(Cds)

```
void setup() {
   pinMode(27, INPUT_PULLUP); 내부 풀업 저항
   ledcSetup(0, 500, 8);
   ledcSetup(1, 500, 8);
   ledcSetup(2, 500, 8);
   ledcAttachPin(13, 0);
   ledcAttachPin(12, 1);
   ledcAttachPin(14, 2);
}
```

5 빛 센서(Cds)



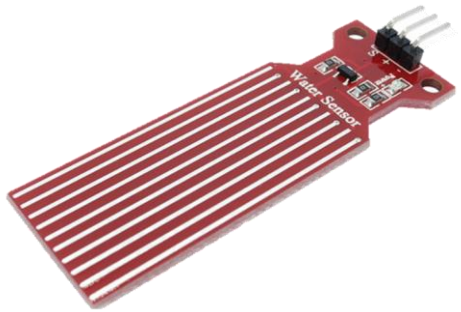
전체 코드

```
void setup() {
  pinMode(27, INPUT_PULLUP);
  ledcSetup(0, 500, 8);
  ledcSetup(1, 500, 8);
  ledcSetup(2, 500, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

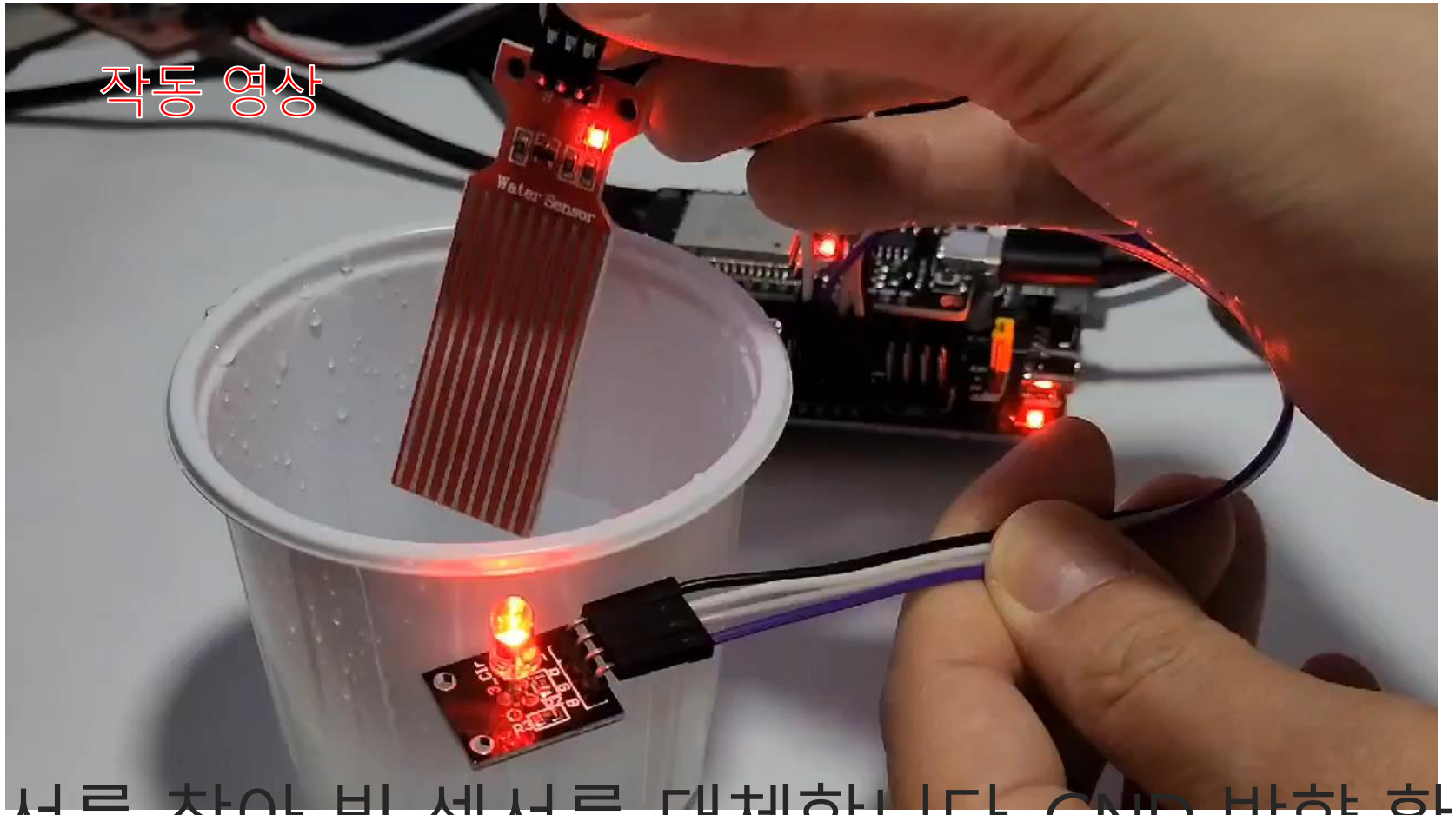
```
void loop() {
  int sensor = analogRead(27);
  sensor = map(sensor, 0, 4095, 0, 511);
  if(sensor <= 255) {
   setColor(255-sensor, sensor, 0);
  } else {
   setColor(0, 511-sensor, sensor-255);
  }
}
```

```
void setColor(int redValue, int greenValue, int blueValue) {
  ledcWrite(0, redValue);
  ledcWrite(1, greenValue);
  ledcWrite(2, blueValue);
 }
```

6 워터센서



워터센서



외부에서 워터센서를 찾아 빛 센서를 대체합니다. GND 방향 확인

5. 앱인벤터로 ESP32 제어하기

1 블루투스 통신

- 블루투스 통신: 근거리 디지털 장치 간에 데이터 전송에 사용하는 무선 통신
- ESP32에 블루투스 모듈이 내장되어
- 블루투스 통신으로 노트북, 스마트폰 등 근거리 무선통신 활용 가능

1 블루투스 통신

- 블루투스 헤더 파일 가져오기
- 블루투스 오브젝트 생성하기

```
#include "BluetoothSerial.h"

BluetoothSerial BT;
```

1 블루투스 통신

- 시리얼 통신 시작하기
- 블루투스 통신 시작(블루투스 기기 이름 지정)

```
void setup() {
  Serial.begin(9600);
```

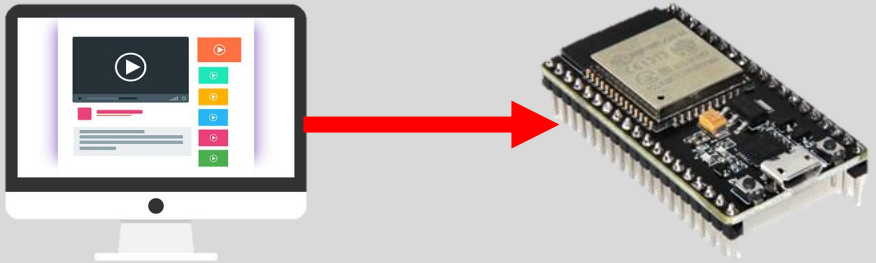
1 블루투스 통신

블루투스 통신 시작(블루투스 기기 이름 지정)

```
    BT.begin("ESP32_my");
}
```

1 블루투스 통신

시리얼 통신으로 데이터 값이 입력되면 그 입력된 값을 블루투스 통신으로 출력

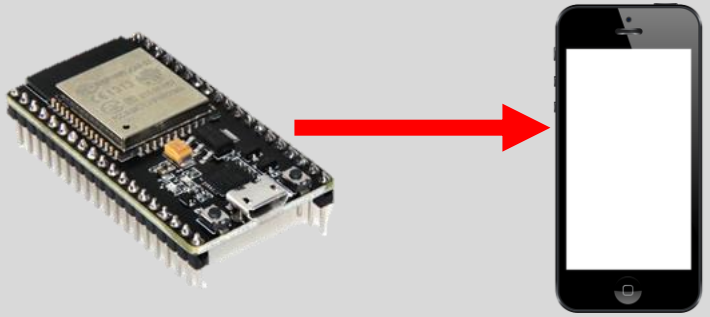


```
void loop() {
  if (Serial.available()) {
    BT.write(Serial.read());
  }
```

1 블루투스 통신

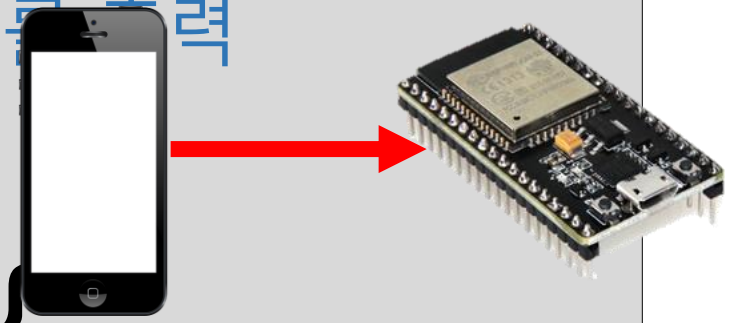
시리얼 통신으로 입력 받은 데이터를 블루투스로 출력

```
void loop() {
  if (Serial.available()) {
    BT.write(Serial.read());
  }
```

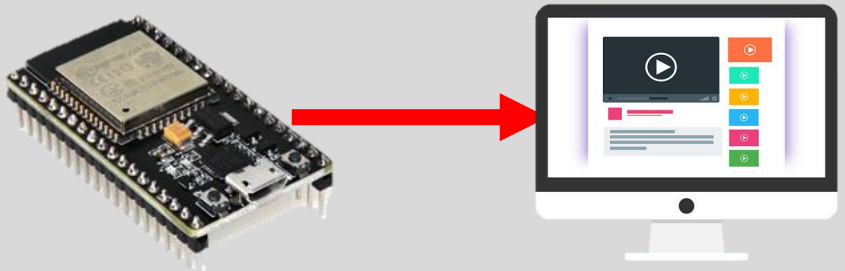


1 블루투스 통신

시리얼 통신으로 블루투스 통신으로 입력 받은 데이터를 출력



```
if (BT.available()) {
    Serial.write(BT.read());
 }
```



1 블루투스 통신

0.02초 간격을 둠

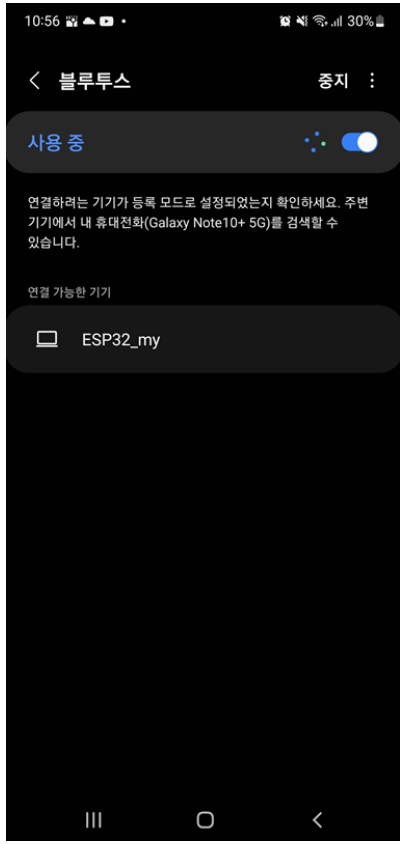
```
    delay(20);
}
```

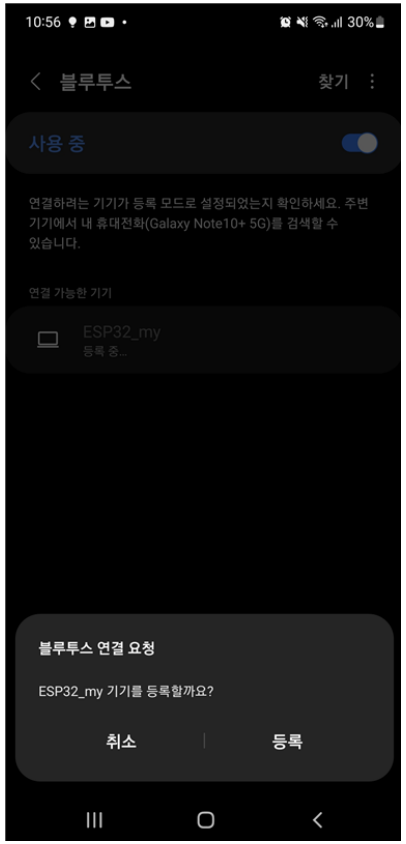
1 블루투스 통신

블루투스 시리얼 통신 예제

```
#include "BluetoothSerial.h"
BluetoothSerial BT;

void setup() {
  Serial.begin(9600);
  BT.begin("ESP32_my");
}

void loop() {
  if (Serial.available()) {
    BT.write(Serial.read());
  }
  if (BT.available()) {
    Serial.write(BT.read());
  }
  delay(20);
}
```

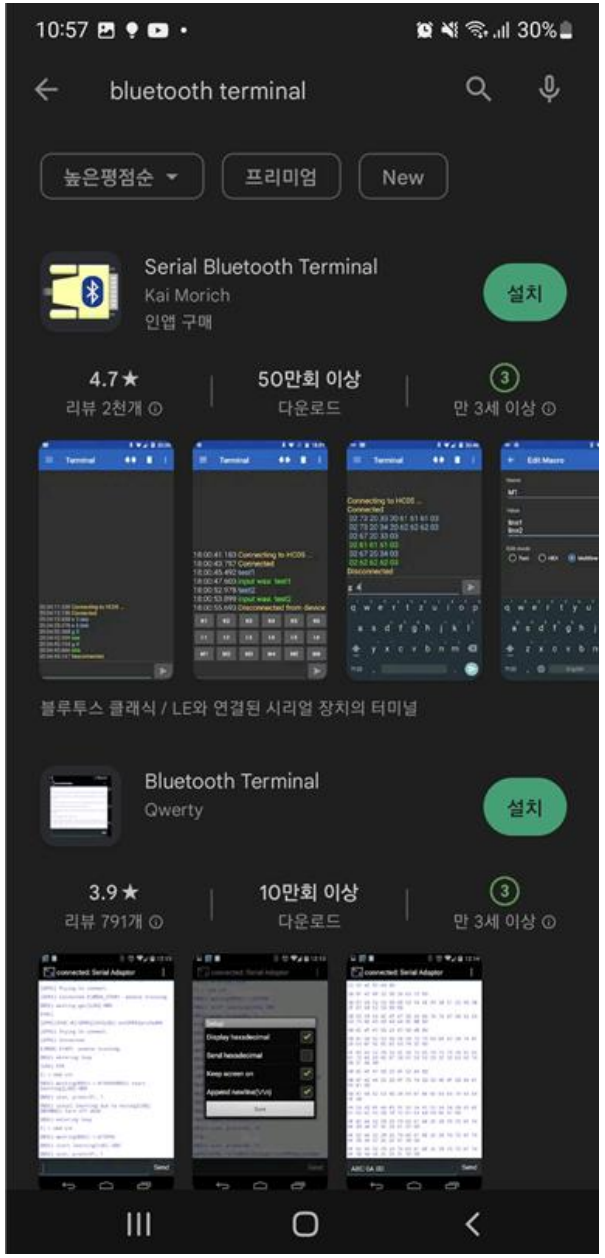
기기 등록



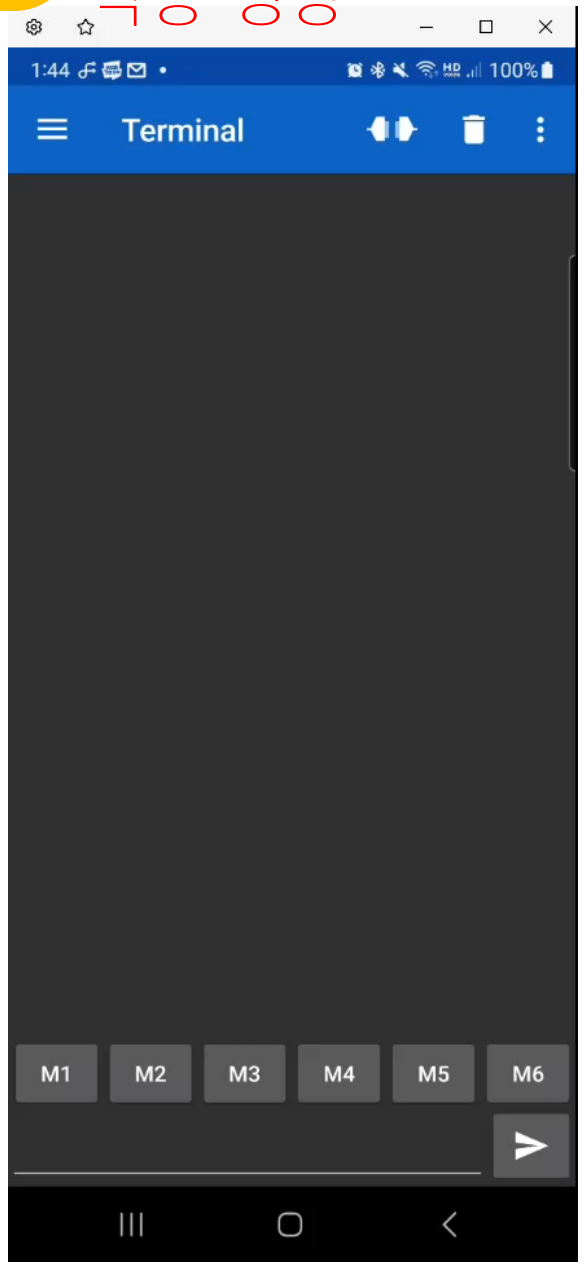


1 블루투스 통신

테스트하기 위해 Serial Bluetooth Terminal 앱을 다운 받아 esp32와 연결

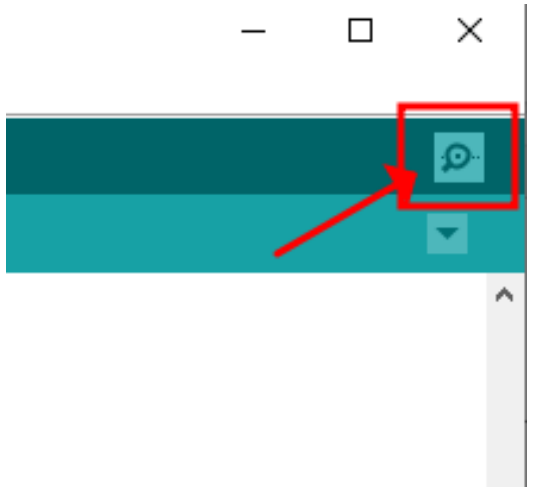


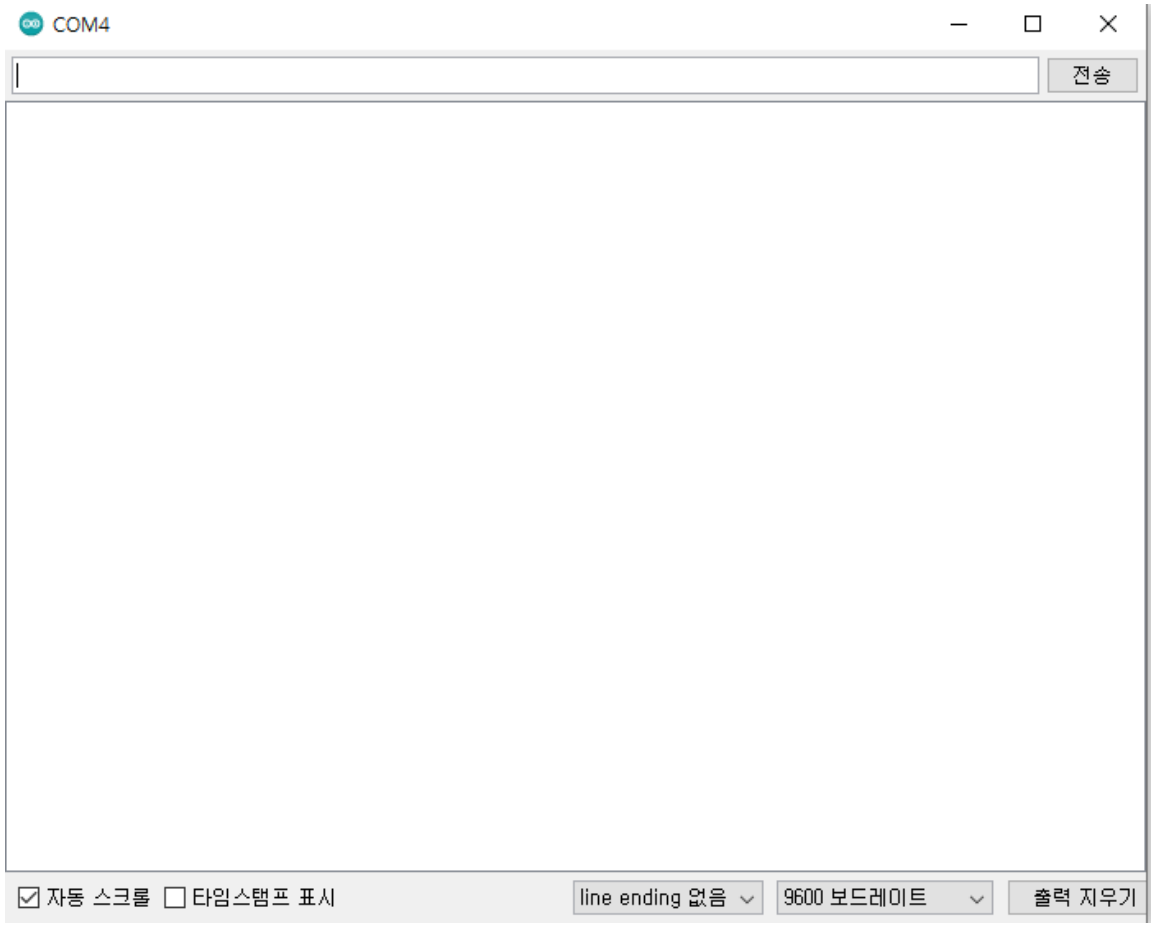
작동 영상



1 블루투스 통신

시리얼 모니터 버튼을 눌러 시리얼 모니터 창을 띄웁니다.

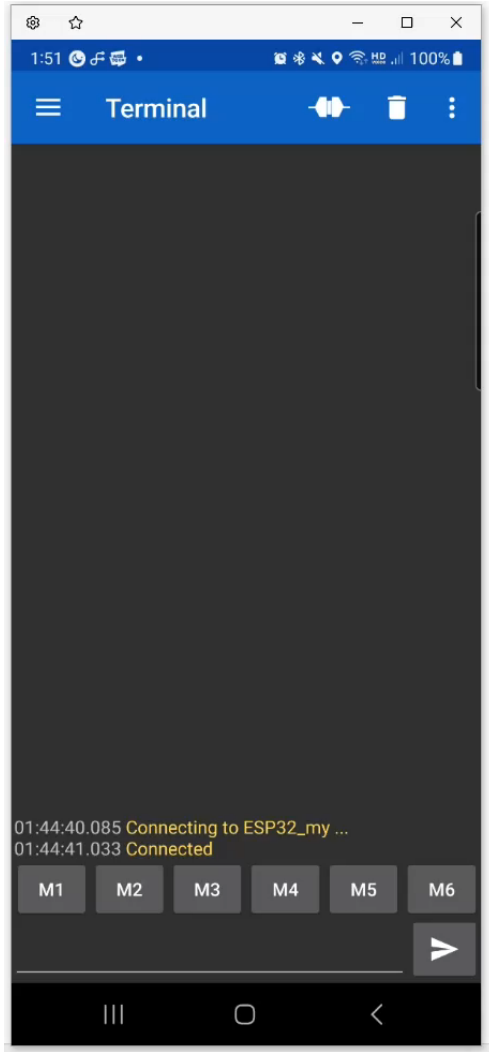




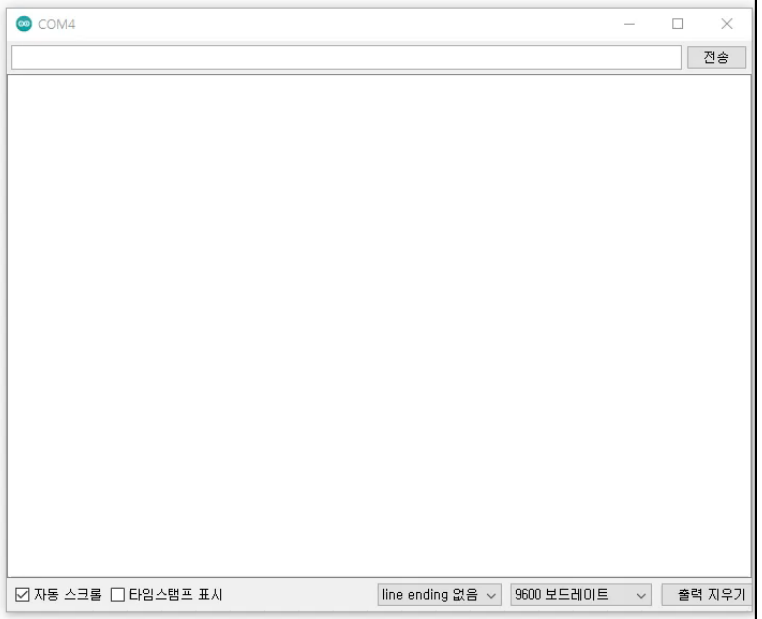
1 블루투스 통신

시리얼 통신으로 데이터를 입력하면 블루투스로 출력이 됩니다.

```
void loop() {
  if (Serial.available()) {
    BT.write(Serial.read());
  }
```



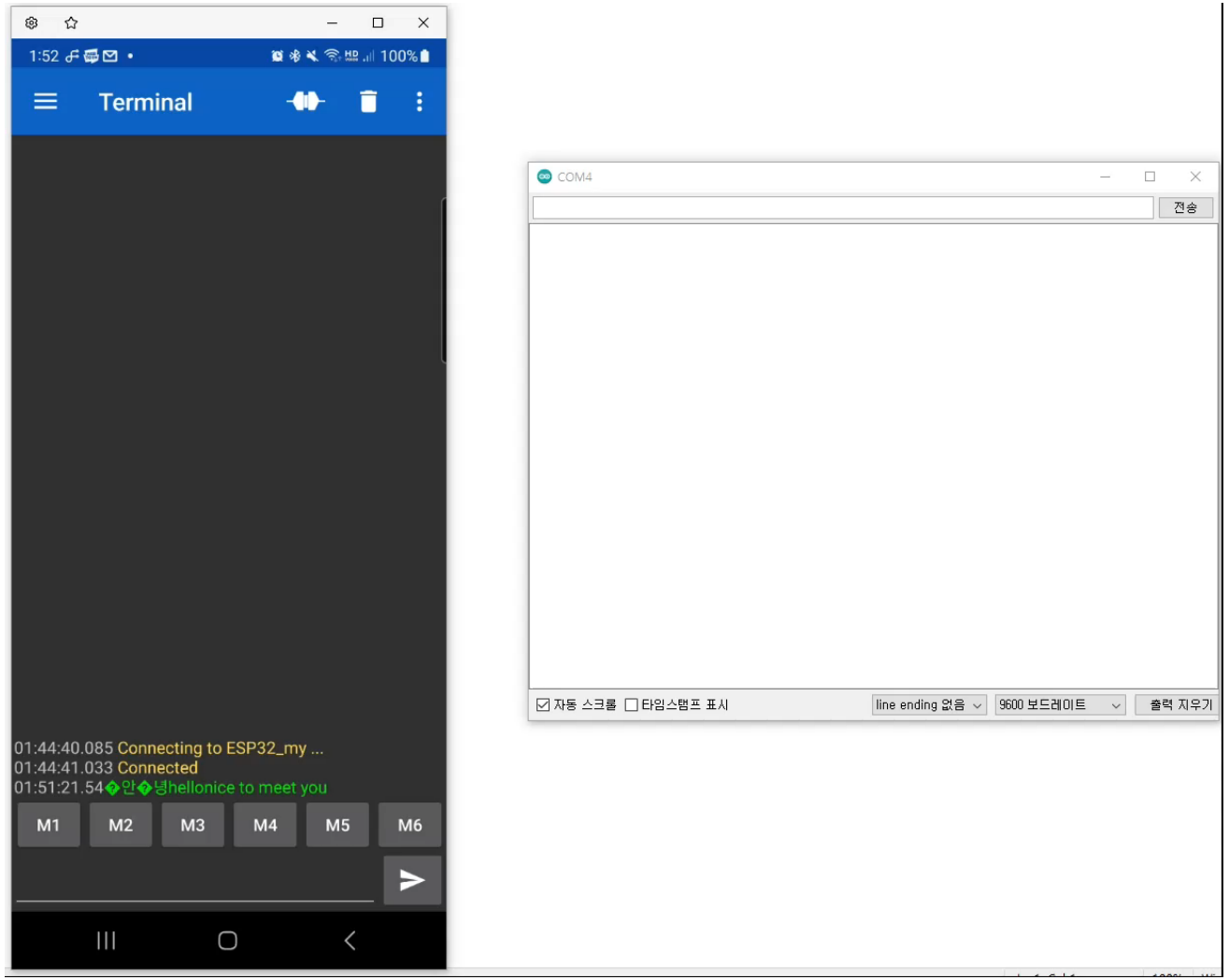
작동 영상



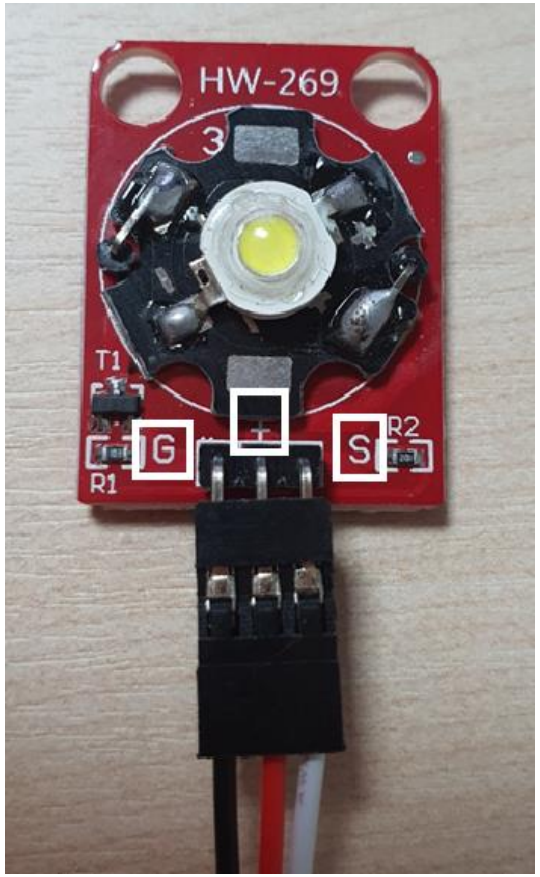
1 블루투스 통신

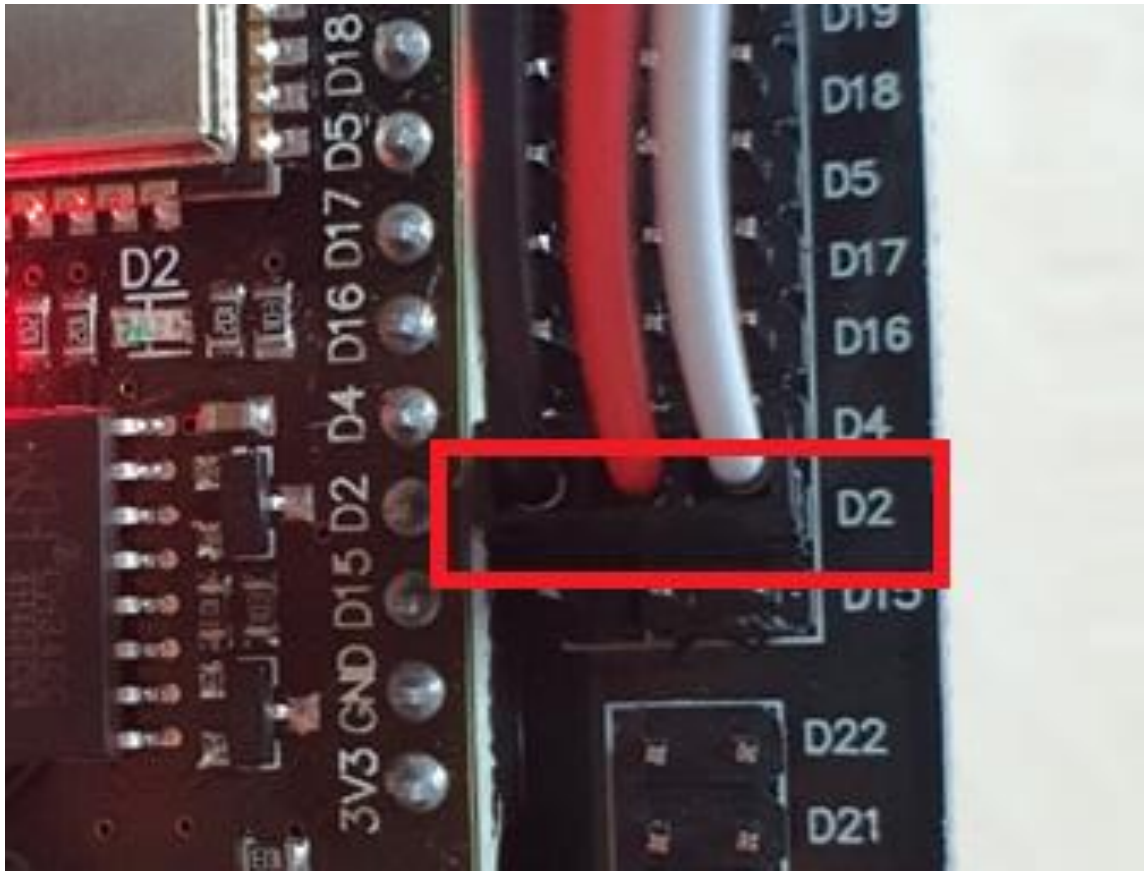
블루투스 통신으로 데이터를 입력하면 시리얼 통신으로 출력이 됩니다.

```
if (BT.available()) {
  Serial.write(BT.read());
}
```



2 무선으로 LED 제어





3W LED를 2번 핀에 연결

2 무선으로 LED 제어

불 변수인 onOff를 만들고 초기값을 false로 줍니다.

```
#include "BluetoothSerial.h"
BluetoothSerial BT;

bool onOff = false;
```

2 무선으로 LED 제어

2번 핀을 출력으로 하고 ESP32_my라는 이름으로 블루투스 통신을 시작합니다.

```
void setup() {
  pinMode(2, OUTPUT);
  BT.begin("ESP32_my");
}
```

2 무선으로 LED 제어

블루투스 통신으로 들어온 데이터를 문자형 변수 c에 입력합니다.

```
void loop() {
  if (BT.available()) {
    char c = BT.read();
    if(c == 'A'){
      digitalWrite(2, onOff = !onOff);
    }
  }
  delay(20);}
```

2 무선으로 LED 제어

c에 저장된 데이터가 문자 A라면 2번 핀의 출력 값을 지금과 반대로 바꿉니다.

```
void loop() {
  if (BT.available()) {
    char c = BT.read();
    if(c == 'A'){
      digitalWrite(2, onOff = !onOff);
    }
  }
  delay(20);}
```

2 무선으로 LED 제어

코드의 안정성을 위해 딜레이를 조금 줍니다.

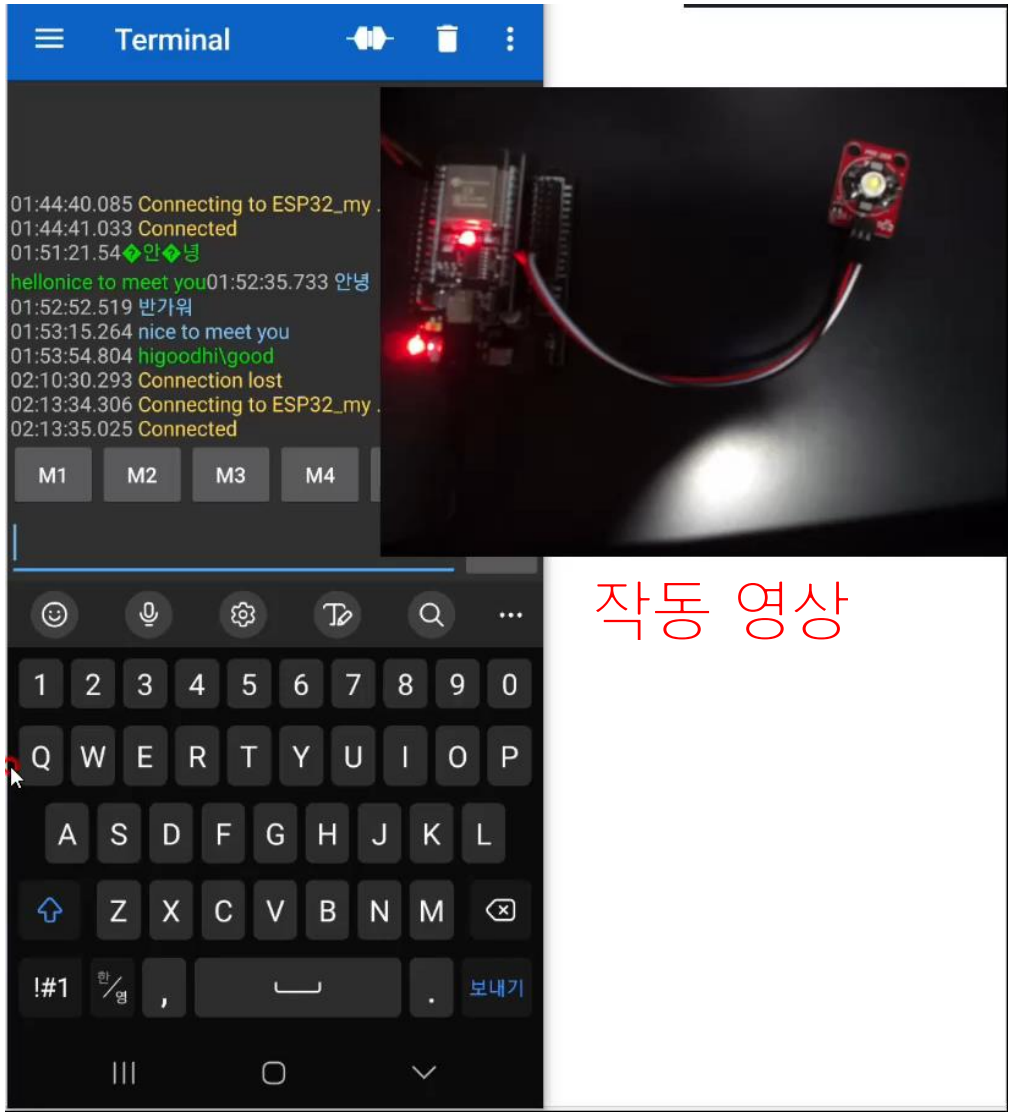
```
void loop() {
  if (BT.available()) {
    char c = BT.read();
    if(c == 'A'){
      digitalWrite(2, onOff = !onOff);
     }
  }
  delay(20);}
```

2 무선으로 LED 제어

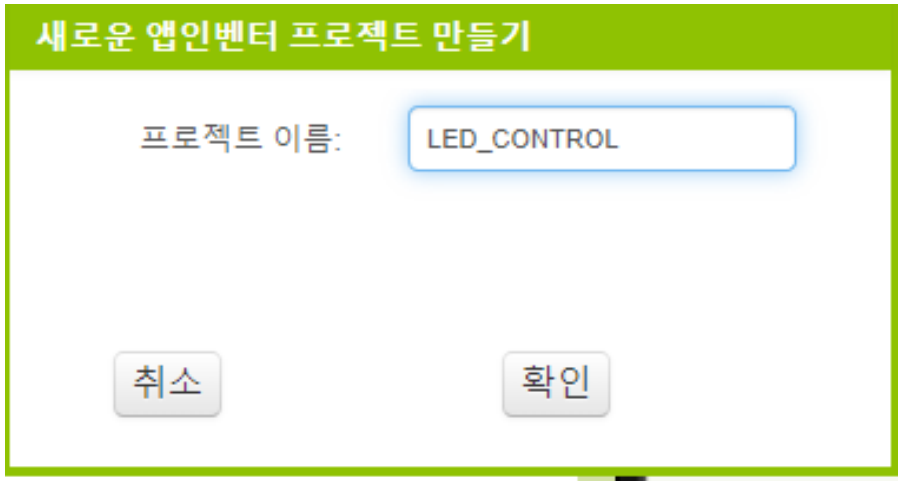
전체 코드

```
#include "BluetoothSerial.h"
BluetoothSerial BT;

bool onOff = false;

void setup() {
  pinMode(2, OUTPUT);
  BT.begin("ESP32_my");
}

void loop() {
  if (BT.available()) {
    char c = BT.read();
    if(c == 'A'){
      digitalWrite(2, onOff = !onOff);
     }
   }
   delay(20);
}
```

터미널앱에서 A를 입력하면 LED가 제어 됨



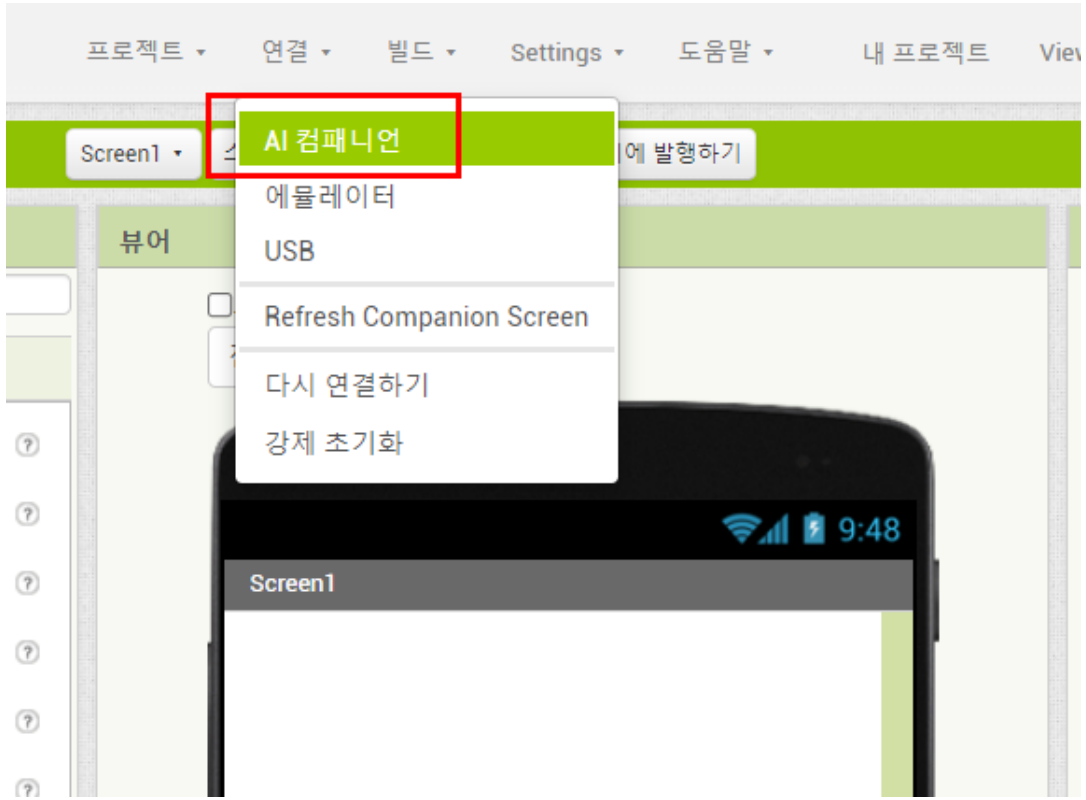
앱 제작의 실제1 - LED를 무선으로 켜는 앱



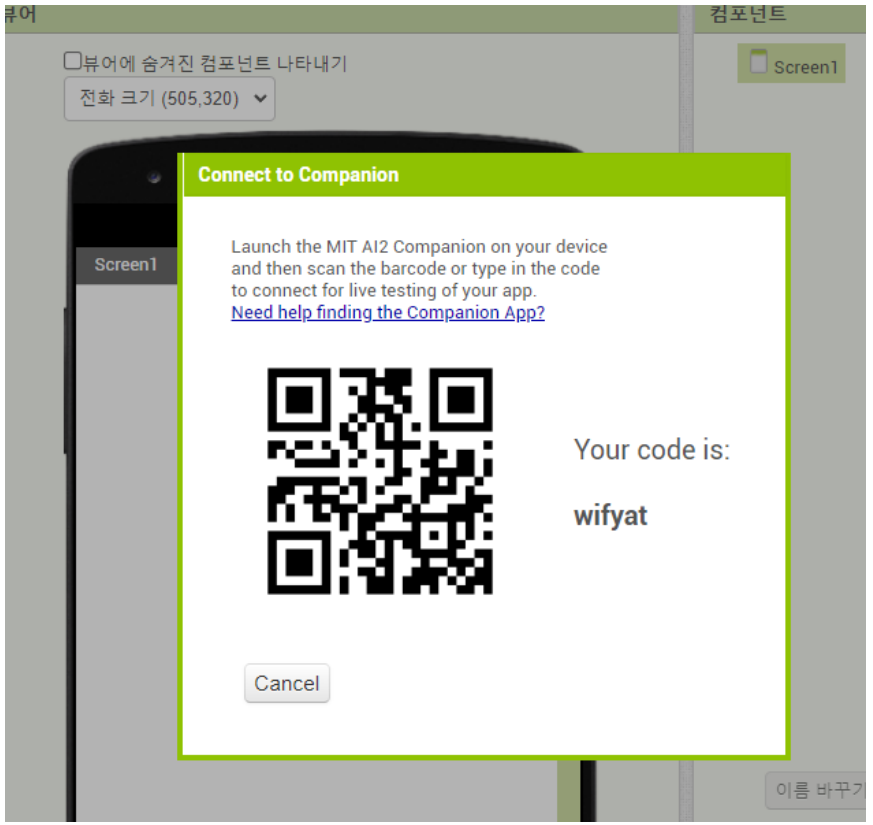
앱 이름 LED_CONTROL

앱 제작의 실제1 - LED를 무선으로 켜는 앱

연결 – AI 컴패니언 클릭



QR코드를 mit ai2 companion 앱으로 찍음



앱 제작의 실제1 - LED를 무선으로 켜는 앱

Screen1 수평 정렬 가운데

앱 제작의 실제1 - LED를 무선으로 켜는 앱

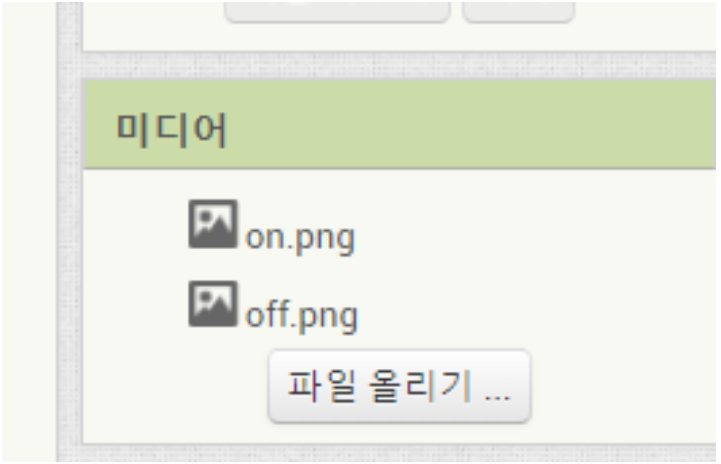
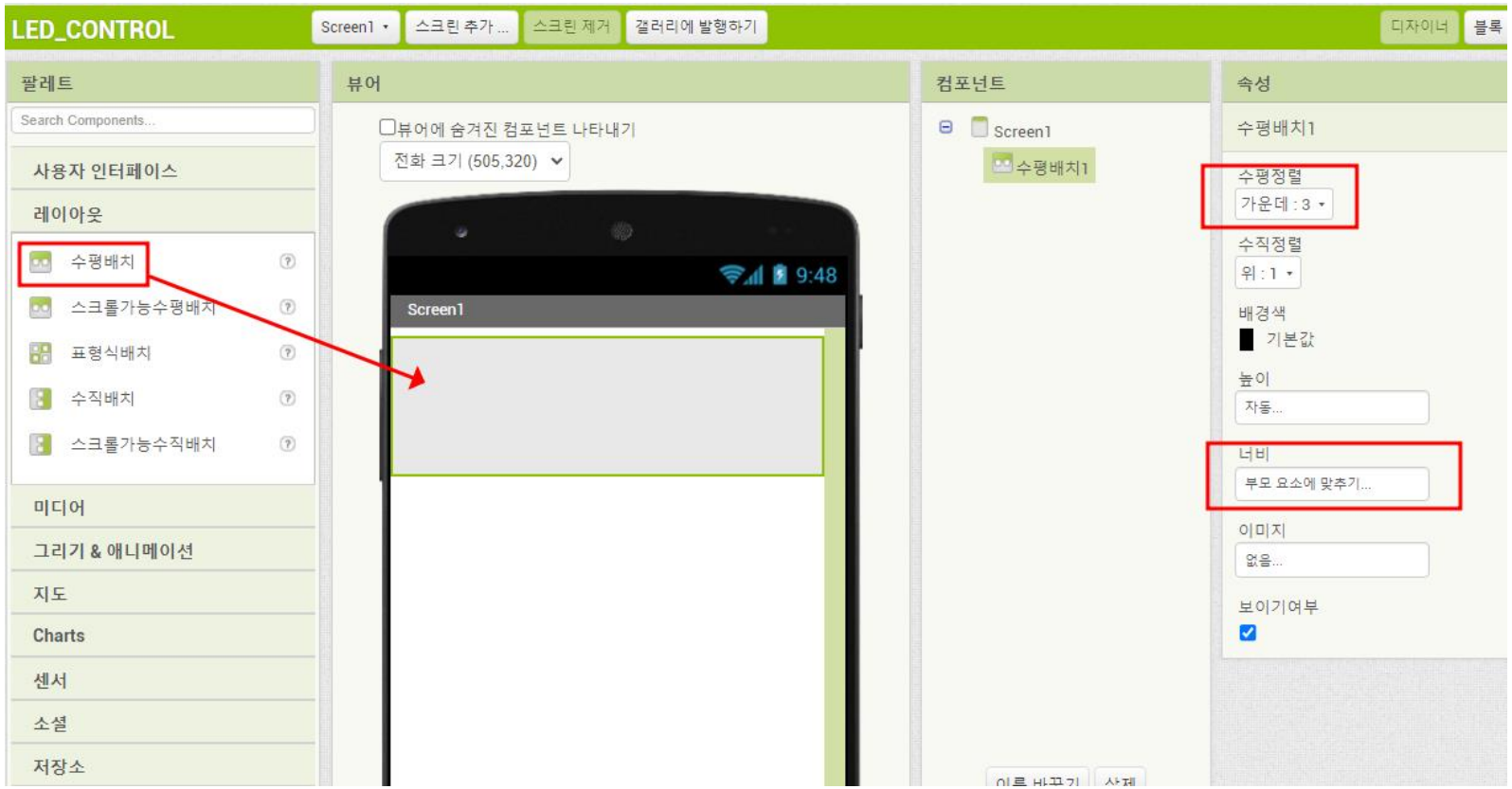


그림 가져오기

앱 제작의 실제1 - LED를 무선으로 켜는 앱



수평 배치를 가져온다.

수평정렬: 가운데

너비: 부모요소

앱 제작의 실제1 - LED를 무선으로 켜는 앱

수평 배치 안으로

목록선택버튼

버튼 추가

앱 제작의 실제1 - LED를 무선으로 켜는 앱

버튼 사이에 레이블 추가

앱 제작의 실제1 - LED를 무선으로 켜는 앱

레이블 텍스트: 레이블1

텍스트색상: 없음

앱 제작의 실제1 - LED를 무선으로 켜는 앱

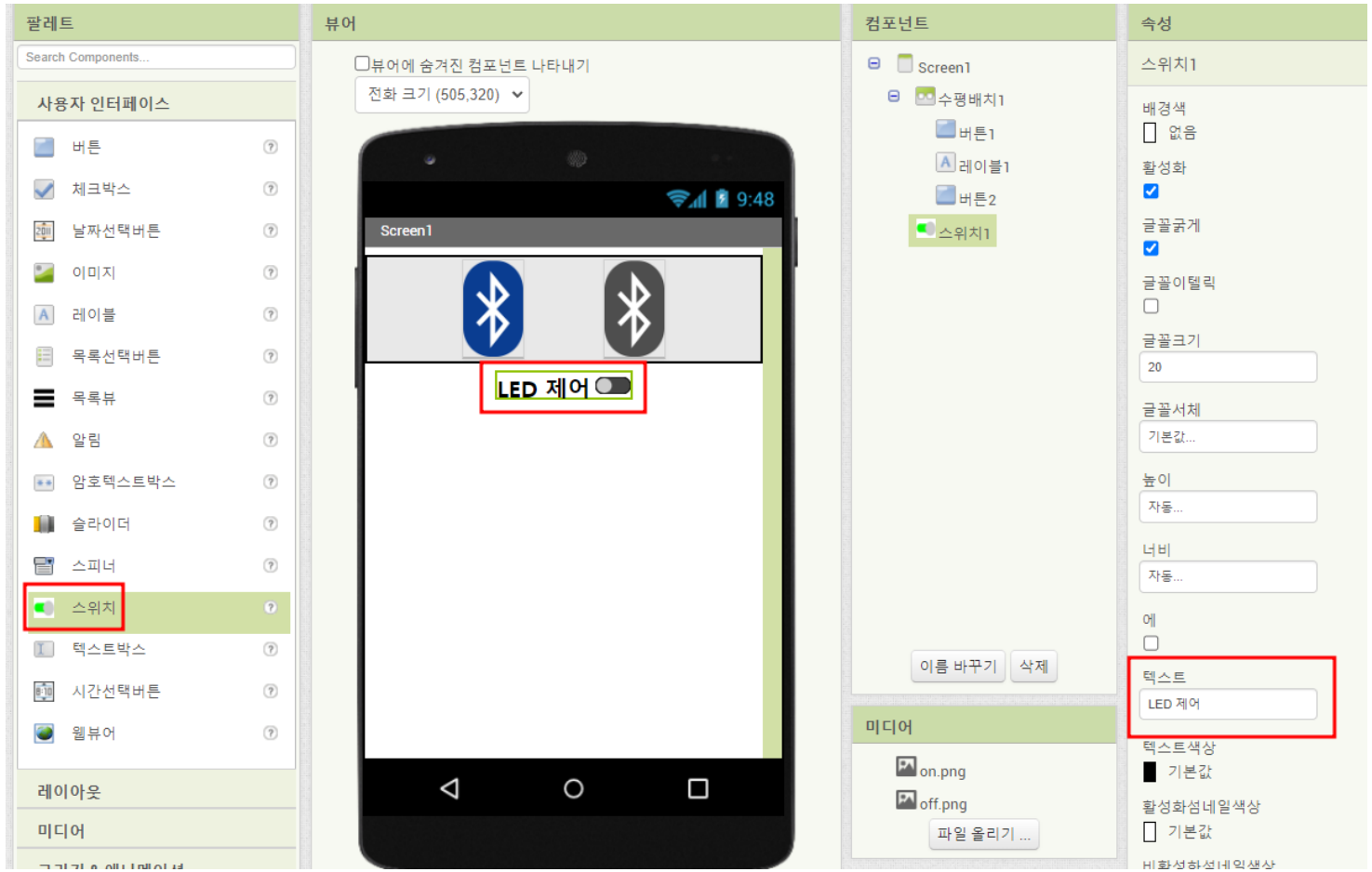
높이80 너비50 조정

텍스트 삭제

앱 제작의 실제1 - LED를 무선으로 켜는 앱

같은 방법으로 off 버튼 추가

앱 제작의 실제1 - LED를 무선으로 켜는 앱



LED 제어 버튼 만들기

앱 제작의 실제1 - LED를 무선으로 켜는 앱

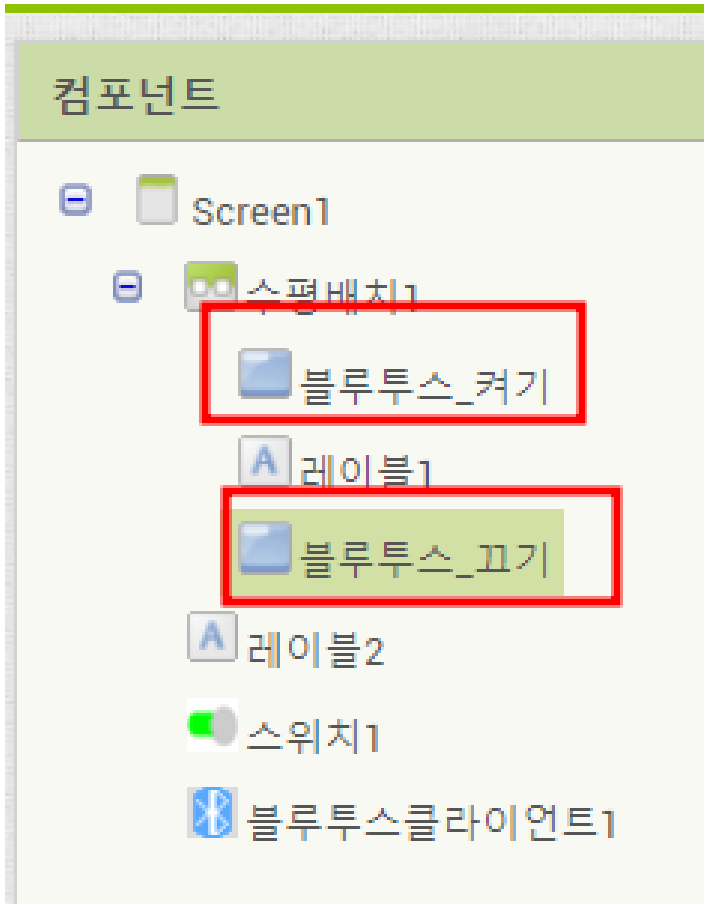
레이블을 사이에 넣어

스위치 간격 조절

앱 제작의 실제1 - LED를 무선으로 켜는 앱

블루투스 클라이언트 추가

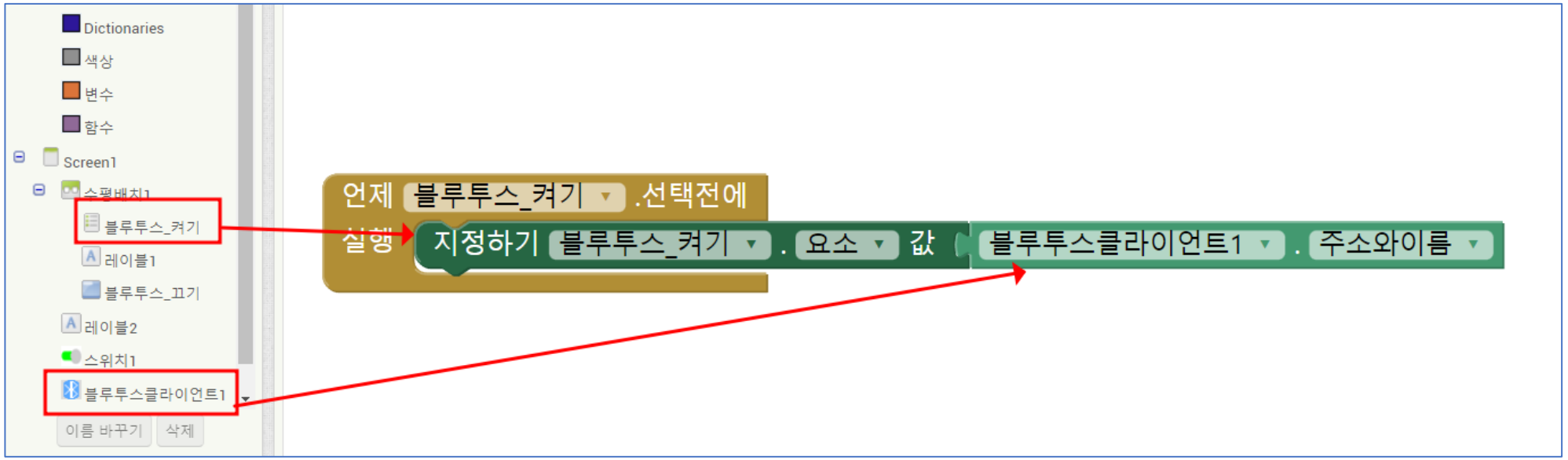
앱 제작의 실제1 - LED를 무선으로 켜는 앱



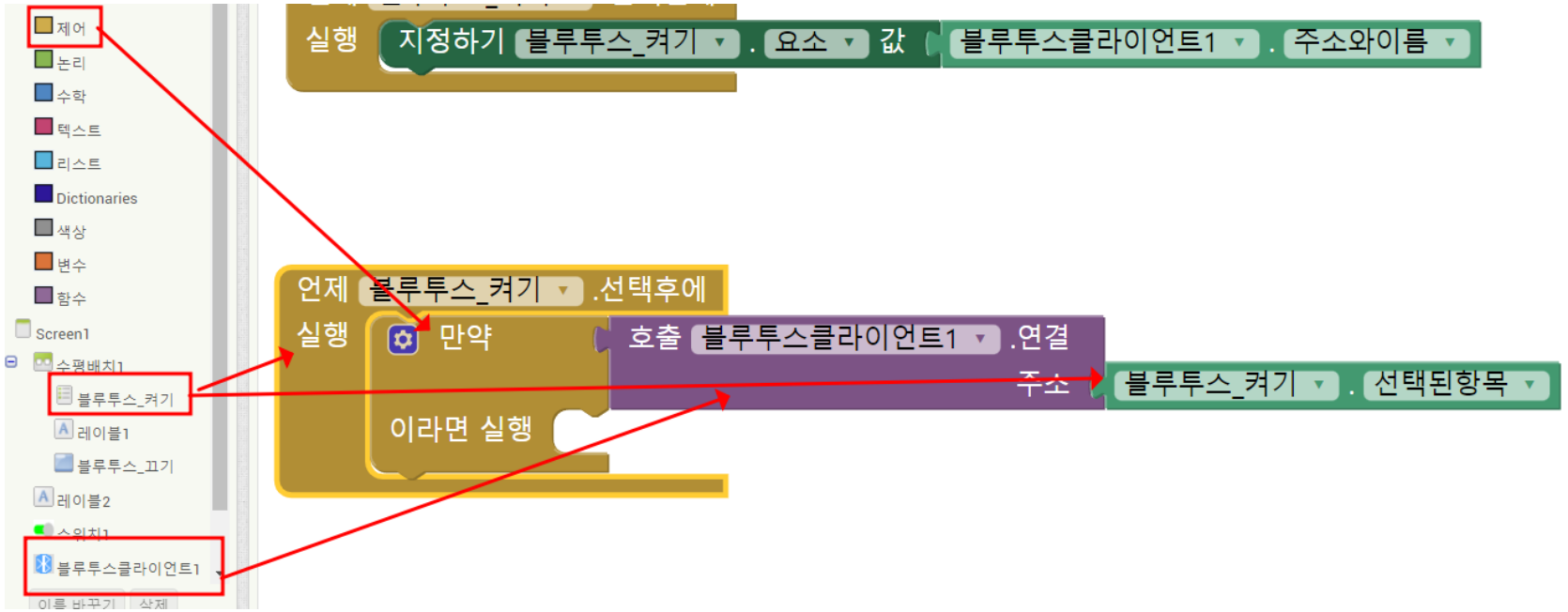
컴포넌트 이름 바꾸기

앱 제작의 실제1 - LED를 무선으로 켜는 앱

목록선택버튼을 누르면
주변의 블루투스 목록이 모두 뜨도록
한다.

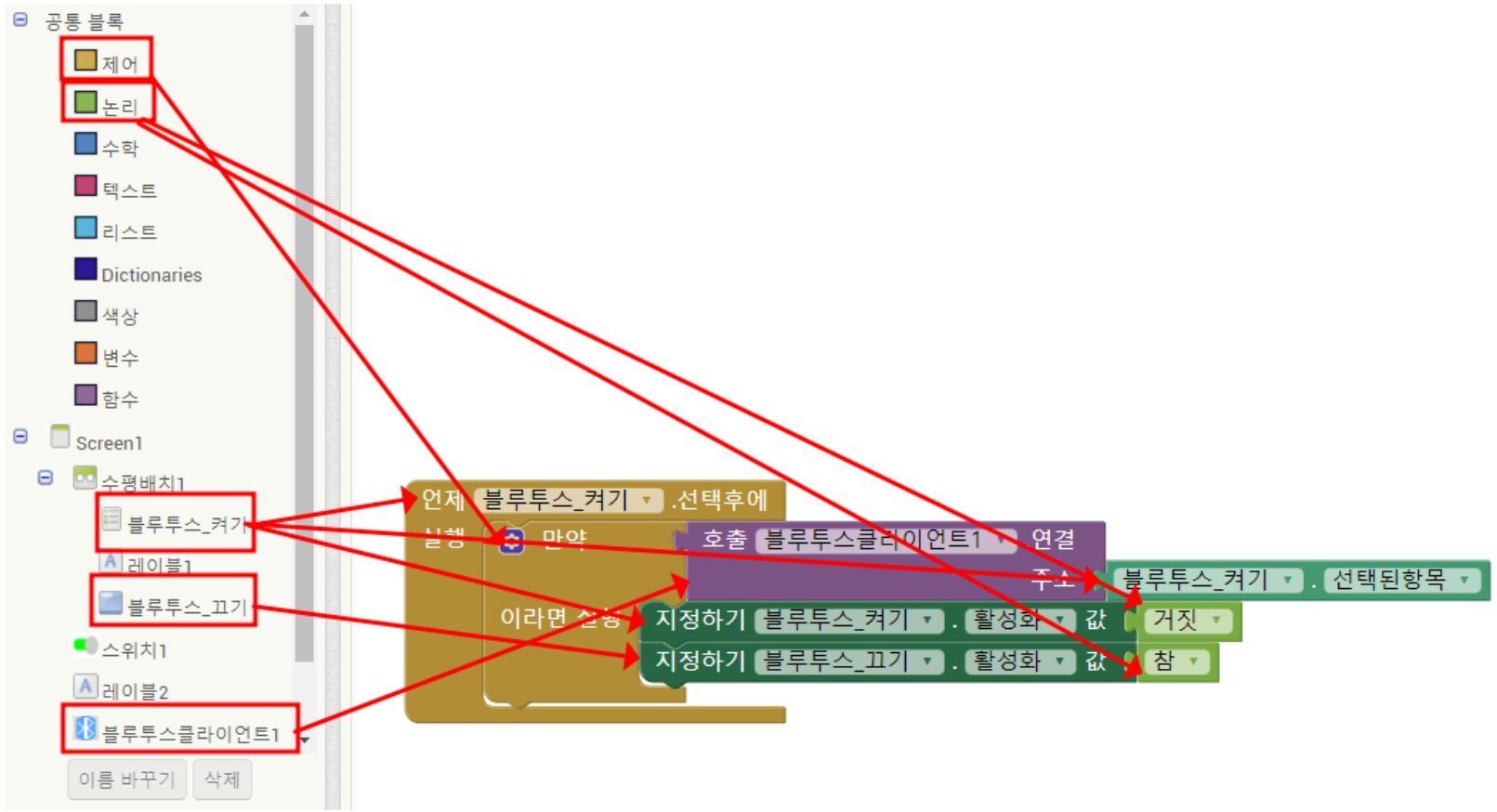


앱 제작의 실제1 - LED를 무선으로 켜는 앱



블루투스가 선택된 항목으로 연결되면

앱 제작의 실제1 - LED를 무선으로 켜는 앱



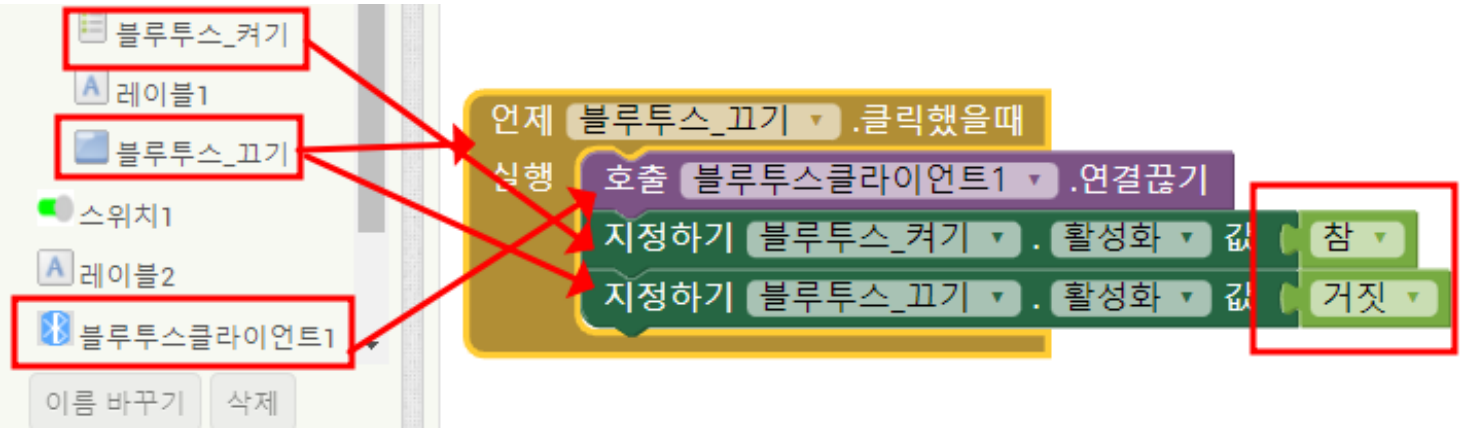
블루투스 켜기 버튼을

비활성화하고

블루투스 끄기 버튼을

활성화한다.

앱 제작의 실제1 - LED를 무선으로 켜는 앱

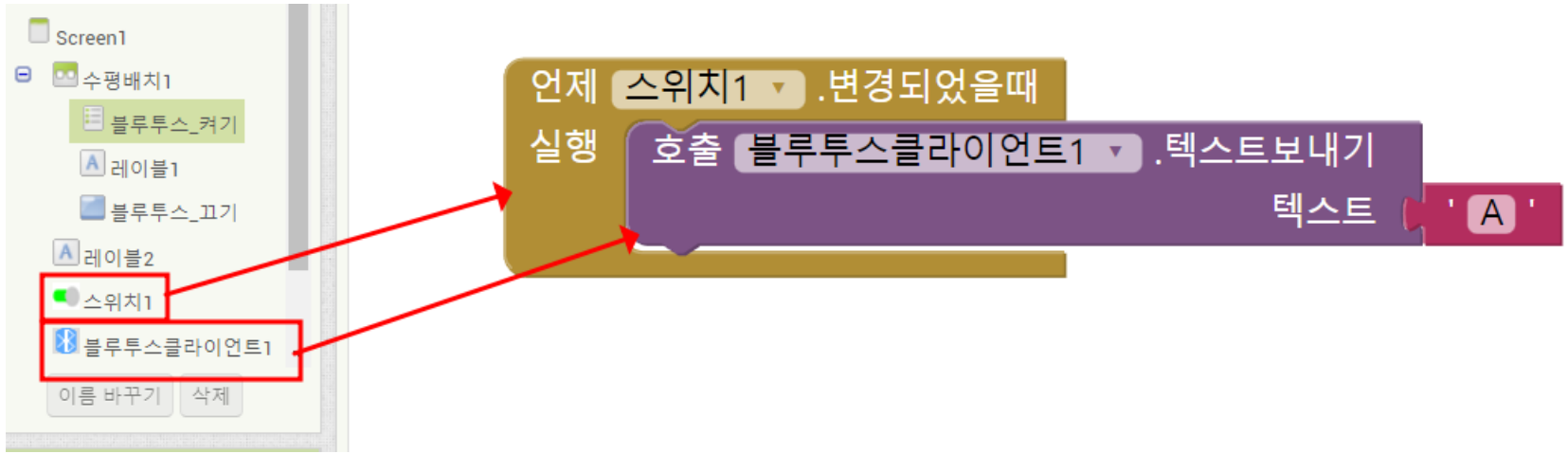


블루투스 끄기 버튼을 클릭하면

블루투스 켜기 버튼을 활성화하고

시블루투스 끄기 버튼을 비활성화한다.

앱 제작의 실제1 - LED를 무선으로 켜는 앱

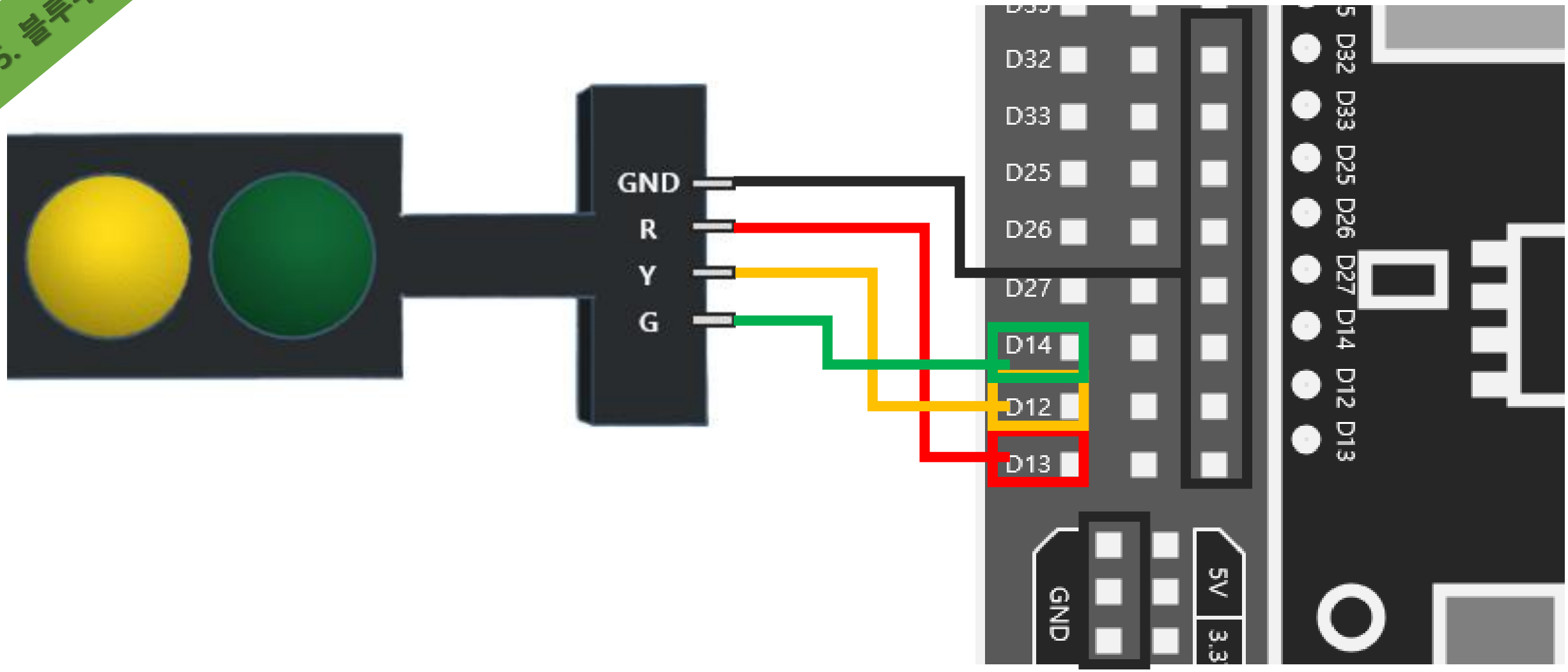


스위치가 변경되었을 때 A를 블루투스로 전송한다.

앱 제작의 실제1 - LED를 무선으로 켜는 앱

작동 영상

2 무선으로 신호등 LED 제어



R은 GPIO13, Y는 GPIO12, G는 GPIO14, GND는 GND핀 아무 데나 연결

2 무선으로 신호등 LED 제어

도전해 볼까요?

블루투스로 신호등 제어하기

A로 빨간불 제어, B로 노란불 제어, C로 초록불 제어

2 무선으로 신호등 LED 제어

- 블루투스 통신 시작(블루투스 기기 이름 지정)
- 오브젝트 생성 및 불 변수 생성

```
#include "BluetoothSerial.h"
BluetoothSerial BT;

bool onOff = false;
```

2 무선으로 신호등 LED 제어

```
int r = 13;
int y = 12;
int g = 14;


void setup(){
  pinMode(r, OUTPUT);
  pinMode(y, OUTPUT);
  pinMode(g, OUTPUT);
  BT.begin("ESP_my");
}
```

- 핀 번호 지정 및 핀 모드 설정
- 블루투스 통신 시작

2 무선으로 신호등 LED 제어

A를 입력 받아 LED 제어

```
void loop(){
  if(BT.available()) {
    char c = BT.read();
    if(c == 'A') {
      digitalWrite(r, onOff = !onOff);
    }
```

2 무선으로 신호등 LED 제어

B, C 입력 시 각각 y, g 변수 반대로 변경

```
  if(c == 'B') {
        digitalWrite(y, onOff = !onOff);
     }
  if(c == 'C') {
        digitalWrite(g, onOff = !onOff);
     }
     delay(20);
   }
}
```

2 무선으로 신호등 LED 제어

코드 전체

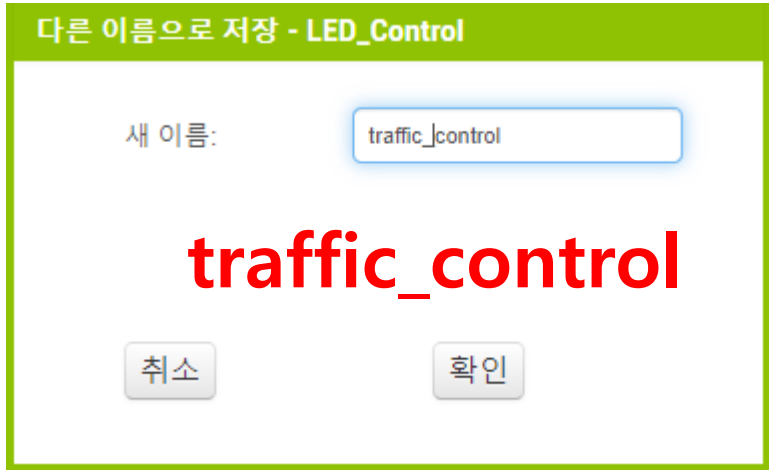
```
#include "BluetoothSerial.h"
BluetoothSerial BT;

bool onOff = false;
int r = 13;
int y = 12;
int g = 14;

void setup(){
  pinMode(r, OUTPUT);
  pinMode(y, OUTPUT);
  pinMode(g, OUTPUT);
  BT.begin("ESP_my");
}
```

```
void loop(){
  if(BT.available()) {
    char c = BT.read();
    if(c == 'A') {
      digitalWrite(r, onOff = !onOff);
    }
    if(c == 'B') {
      digitalWrite(y, onOff = !onOff);
    }
    if(c == 'C') {
      digitalWrite(g, onOff = !onOff);
    }
    delay(20);
  }
}
```

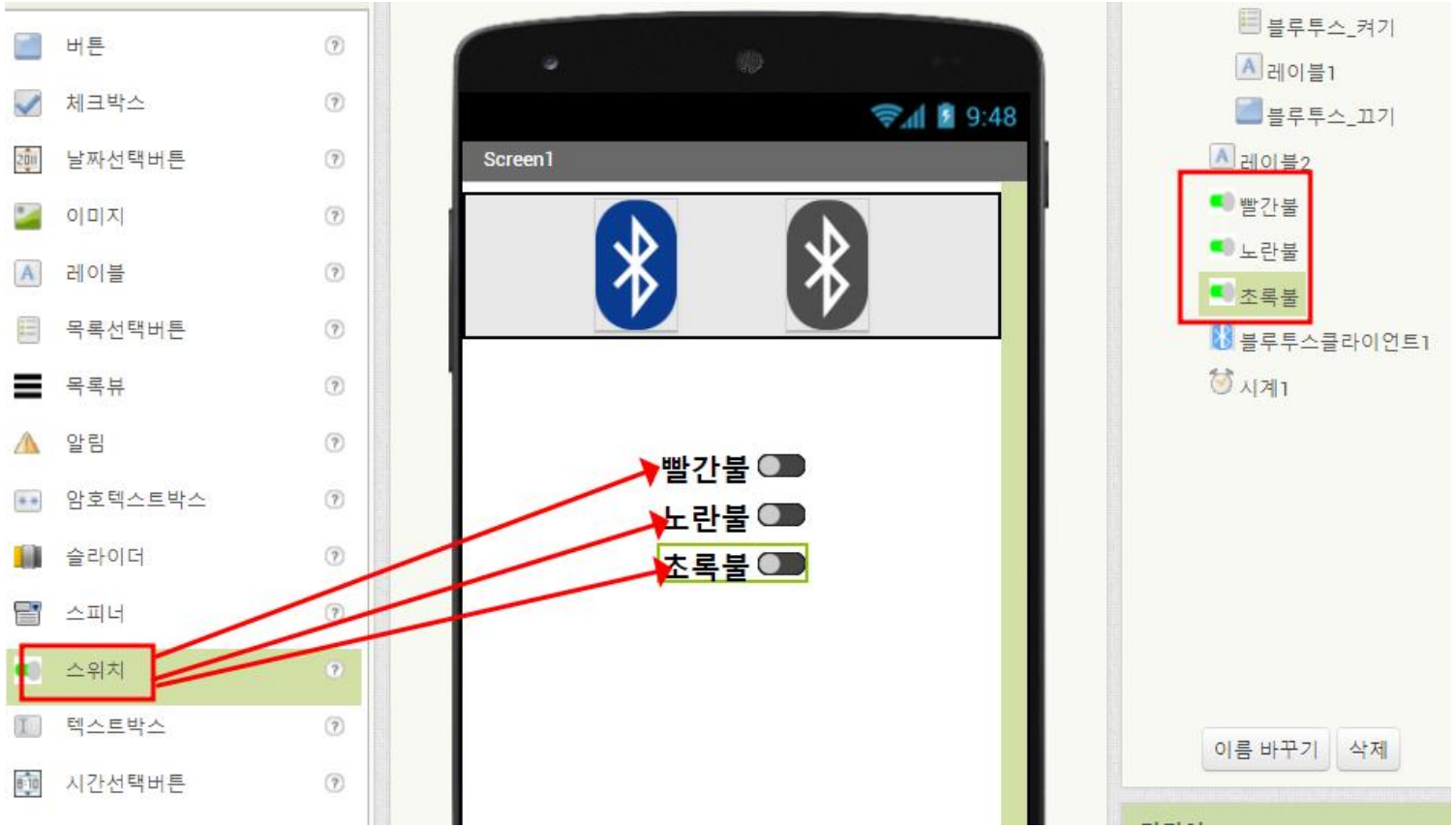
A, B, C로 제어하는 신호등을 만드는 코딩입니다.

앱 제작의 실제2 - 신호등LED 제어하는 앱

led제어 앱을 불러와 다른 이름으로 저장합니다.



앱 제작의 실제2 - 신호등LED 제어하는 앱



스위치를 추가하고

컴포넌트 이름을

변경합니다.

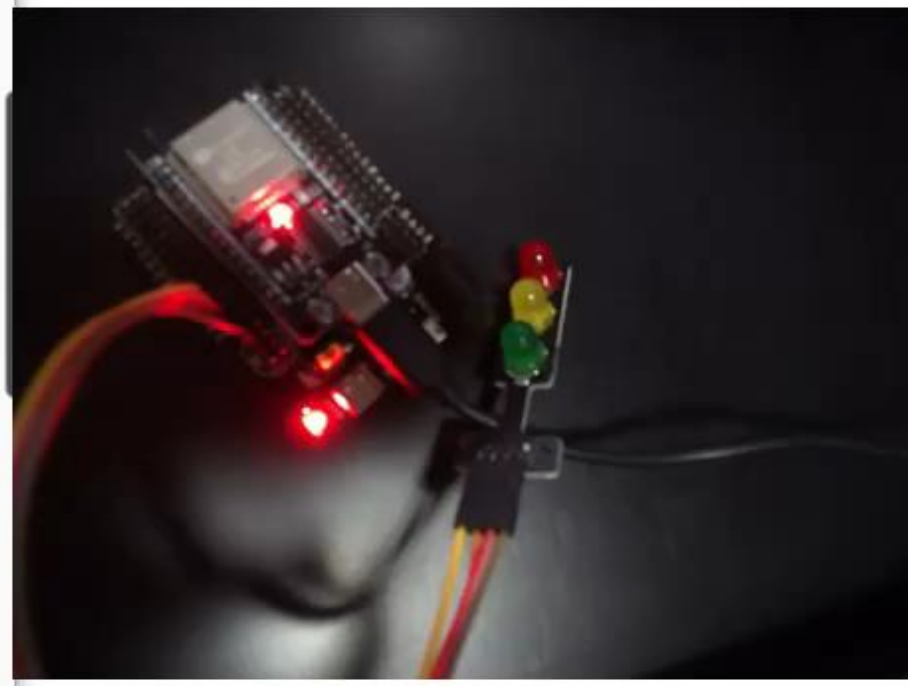
앱 제작의 실제2 - 신호등LED 제어하는 앱

스위치 블록을 추가하여

A, B, C를 블루투스로 송출할 수 있도록 합니다.

앱 제작의 실제2 - 신호등LED 제어하는 앱

작동 영상



3 무선으로 3색 LED 제어

블루투스 통신으로 3색 LED의 색깔을 섞어 보자

3 무선으로 3색 LED 제어

```
void setup() {
  ledcSetup(0, 500, 8);
  ledcSetup(1, 500, 8);
  ledcSetup(2, 500, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
```

채널 및 핀 설정

3 무선으로 3색 LED 제어

블루투스 통신 시작
시리얼 통신 시작

```
    BT.begin("ESP_my");
    Serial.begin(9600);
}
```

3 무선으로 3색 LED 제어

블루투스 통신으로 데이터가 들어오면 문자형 변수 c에 입력

```
void loop() {
  if(BT.available()) {
    char c = BT.read();
```

3 무선으로 3색 LED 제어

변수 c에 r로 시작하는 값이 들어오면 r뒤의 숫자로 0번 채널 밝기 설정
변수 c에 g로 시작하는 값이 들어오면 g뒤의 숫자로 1번 채널 밝기 설정

```
switch (c) {
      case 'r':
        ledcWrite(0, BT.parseInt());
        break;
      case 'g':
        ledcWrite(1, BT.parseInt());
        break;
```

3 무선으로 3색 LED 제어

변수 c에 b로 시작하는 값이 들어오면 b뒤의 숫자로 2번 채널 밝기 설정

```
        case 'b':
          ledcWrite(2, BT.parseInt());
          break;
      }
    }
}
```

3 무선으로 3색 LED 제어

parseInt 명령어의 이해

-parseInt(문자형 자료)를 넣으면 문자형 자료 중 정수 부분만을 숫자형으로 바꿔줌

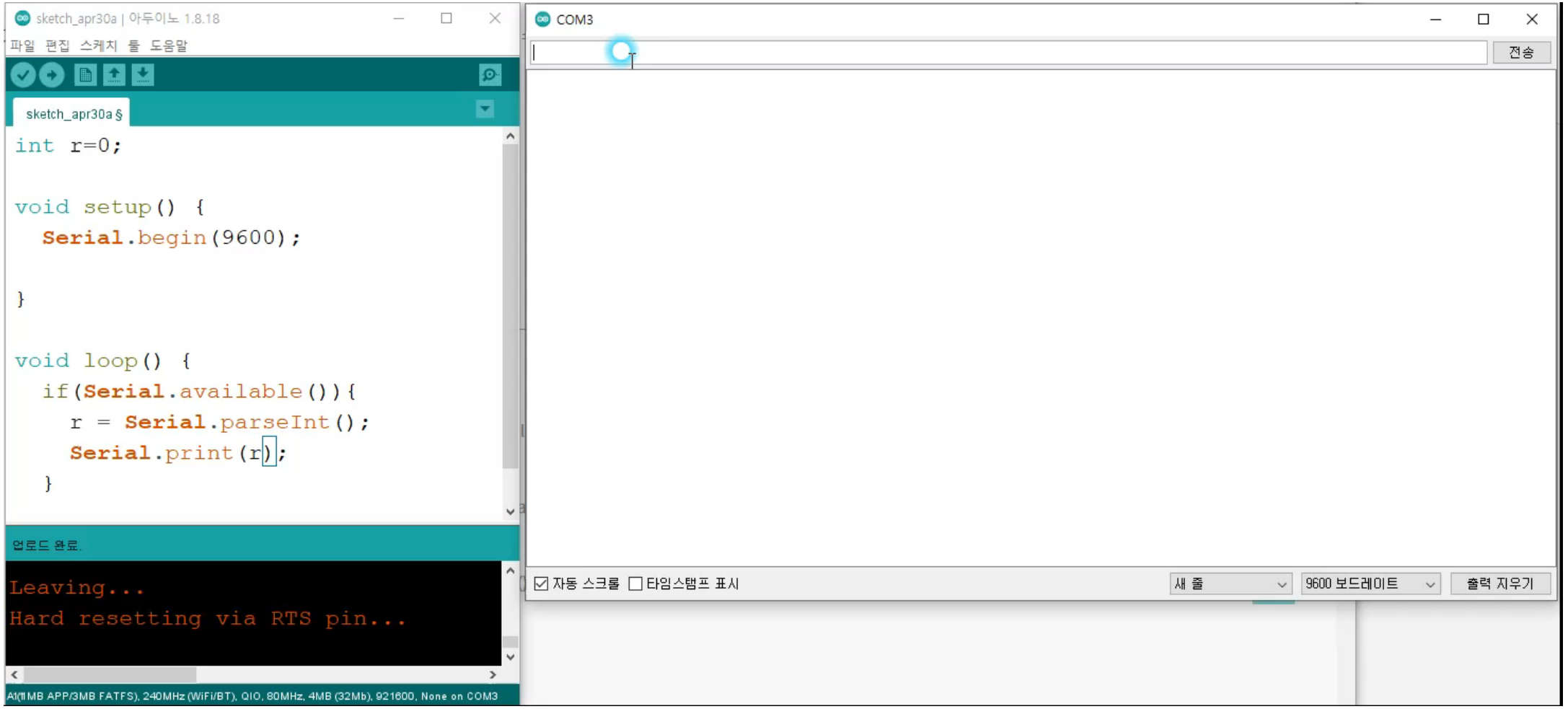
-long 형으로 인식하기 때문에 숫자 하나가 아니라 여러 개를 동시에 받아옴.

-정수만 숫자형으로 바꿔주기 때문에 정수의 앞이나 끝에 오는 문자는 무시됨.

-정수 중간에 문자나 공백이 올 경우 그것을 기준으로 앞의 정수와 뒤의 정수

3 무선으로 3색 LED 제어

시리얼 모니터로 parseInt 결과 확인



3 무선으로 3색 LED 제어

앱의 동작 과정

```
void loop() {
  if(BT.available()) {
    char c = BT.read();
    long bt = BT.parseInt();
    Serial.println(c); // 블루투스로 들어온 값
    Serial.println(bt); // BT.parseInt 값
    switch (c) {
      case 'r':
        ledcWrite(0, BT.parseInt());
        break;
      case 'g':
        ledcWrite(1, BT.parseInt());
        break;
      case 'b':
        ledcWrite(2, BT.parseInt());
        break;
```

빨강 슬라이더를 움직였을 때

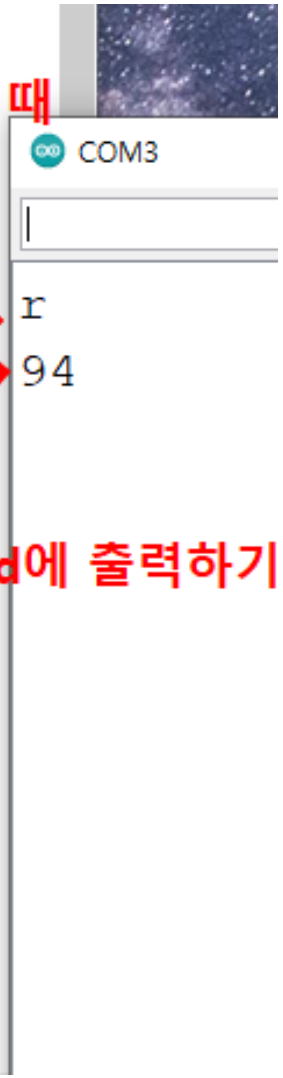
r

94

c == 'r' 일때

94

94를 red led에 출력하기



3 무선으로 3색 LED 제어

코드 전체

```
#include
"BluetoothSerial.h"
BluetoothSerial BT;

void setup() {
  ledcSetup(0, 500, 8);
  ledcSetup(1, 500, 8);
  ledcSetup(2, 500, 8);
  ledcAttachPin(13, 0);
  ledcAttachPin(12, 1);
  ledcAttachPin(14, 2);
}
```

```
void loop() {
  if(BT.available()) {
    char c = BT.read();
  switch (c) {
      case 'r':
        ledcWrite(0, BT.parseInt());
        break;
      case 'g':
        ledcWrite(1, BT.parseInt());
        break;
}
```

앱 제작의 실제3 - 3색 LED 제어하는 앱

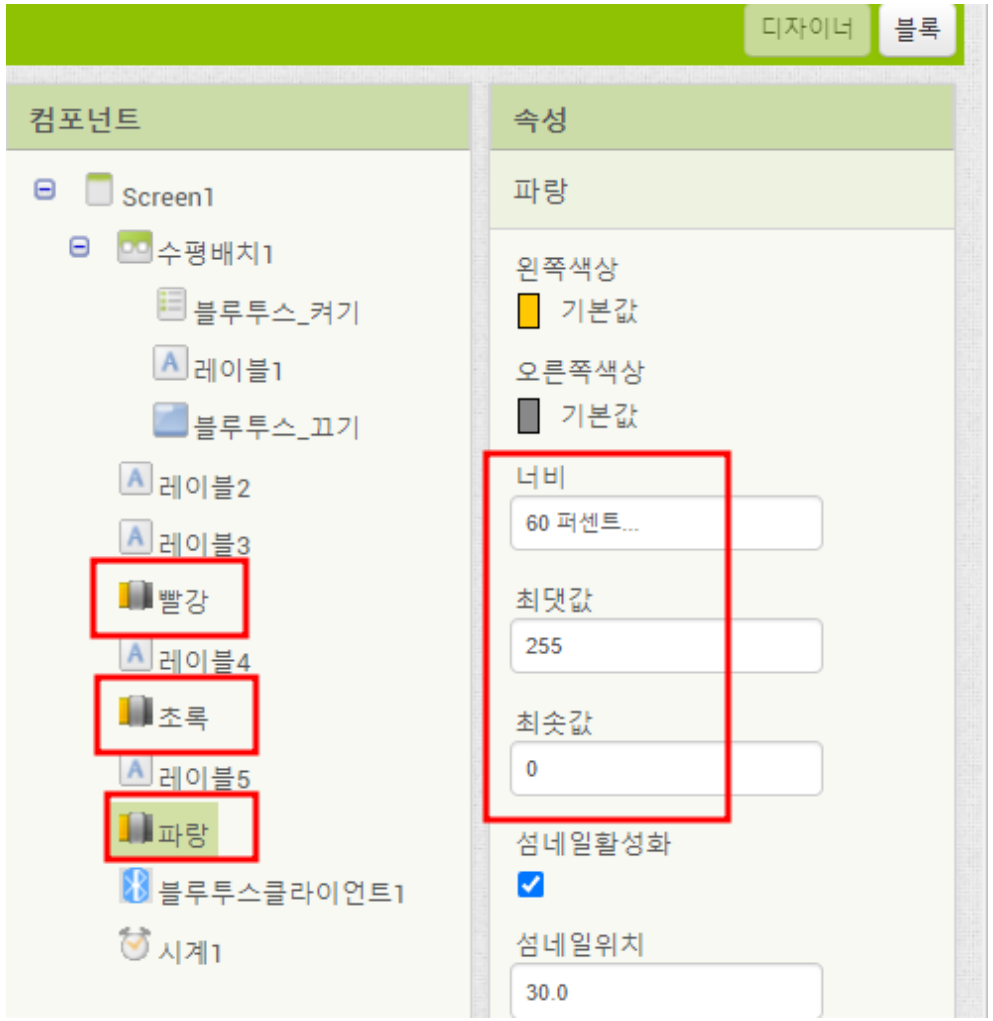
슬라이더를 추가하여 섬네일 값을 블루투스로 송출할 수 있도록 합니다.

앱 제작의 실제3 – 3색 LED 제어하는 앱

레이블 3개, 슬라이더 3개를 교차해서 가져온다.

레이블 이름을 각각 빨강, 초록, 파랑으로 한다.

앱 제작의 실제3 - 3색 LED 제어하는 앱



컴포넌트에서 레이블의 이름을 빨강, 초록, 파랑으로 수정한다.

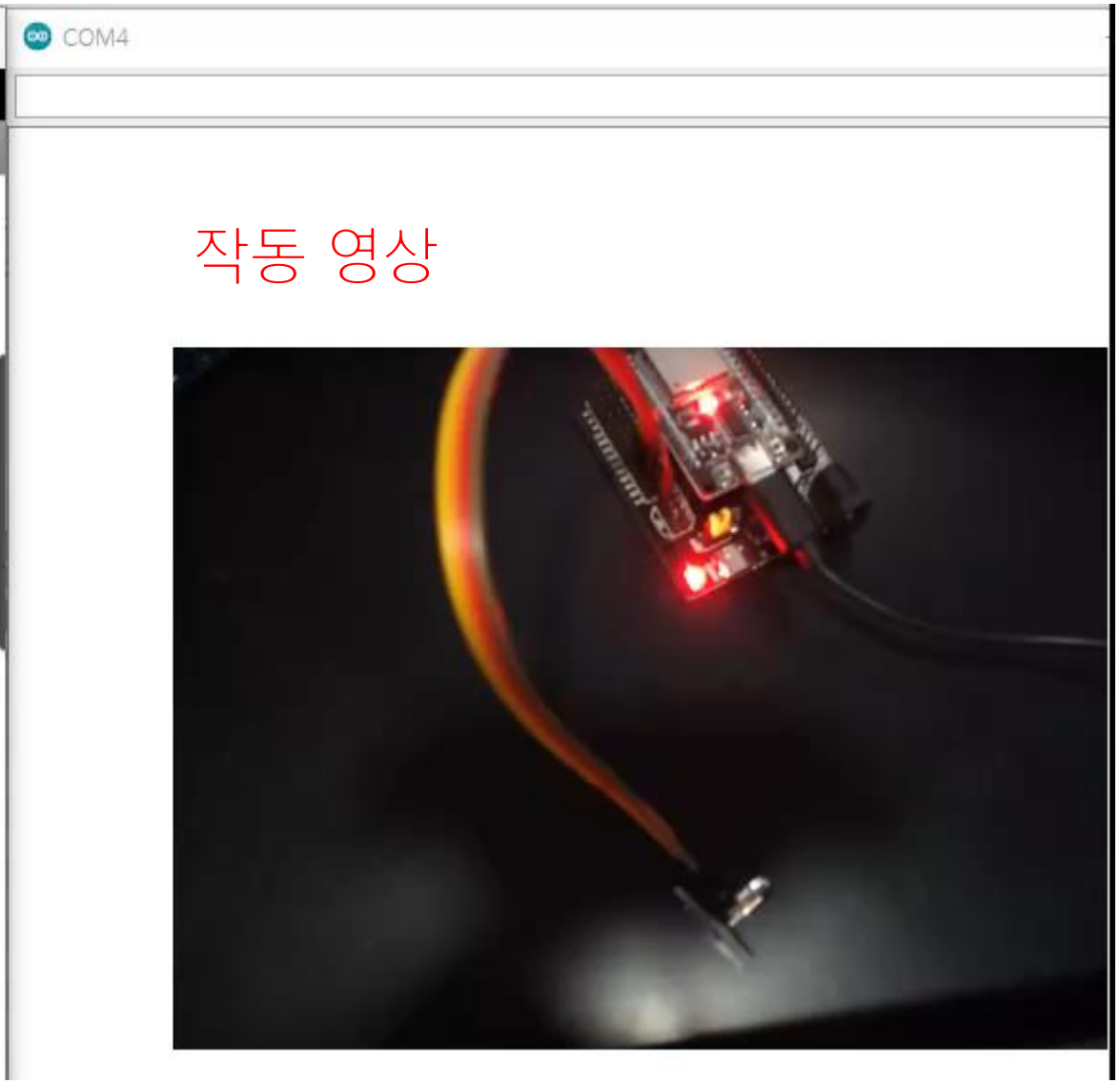
속성은 다음과 같이 입력한다.

- 너비 60퍼센트
- 최댓값 255
- 최솟값 0

앱 제작의 실제3 - 3색 LED 제어하는 앱

슬라이더 위치가 변경되었을 때 섬네일 위치를 전송한다.

앱 제작의 실제3 - 3색 LED 제어하는 앱



작동 영상

5. 나만의 IoT 만들기

어떤 상황에 쓰는 코딩인가요?

예시) 밤에 누워서 폰 보다가 일어나 불 끄기 귀찮은 상황

어떤 순서로 작동하나요?

예시) 1. 스마트폰 앱을 연다 2. 불 끄기 버튼을 누른다. 3. 불이 꺼진다.

1.

아두이노 코드를 복사해서 넣어주세요

아두이노 코드	

앱인벤터 코드를 캡처해서 넣어주세요(윈도우키+shift키+s)

기대 효과는 무엇인가요?

예시) 졸린 상황에서 일어나 불을 끄면서 잠이 깨지 않고 숙면을 취할 수 있다.

유의점

- 업데이트 시 에러가 나는 경우 **BOOT 버튼을 누른 상태에서 업로드**를 한다. 그래도 안 되면 다음과 같이 해결한다.

  1. EN 버튼을 한 번 누른다.
  2. BOOT 버튼을 누른 채로 EN 버튼을 한 번 누른다.
  3. 업로드가 시작되면 BOOT 버튼을 누른 채로 업로드를 한다.
  4. 업로드 후에 정상작동을 안 할 시에는 다시 업로드 버튼을 누르고 에러 메시지가 나올 때까지 기다린 후 작동해 본다.

수고하셨습니다

감사합니다!