

# ESP32 가변저항 & LED 제어 실습지시서

## 준비물

- ESP32 개발 보드 (예: ESP32 Dev Module)
- LED 1개
- 가변저항 1개
- 330Ω 저항 1개 (LED 보호용)
- 브레드보드와 점퍼 와이어
- USB 케이블 (ESP32와 PC 연결용)

## 공통 회로 연결법

- 가변저항 가운데 다리 → ESP32 GPIO 핀 (실습별 지정)
- 버튼 오른쪽 다리 → GND
- 버튼 왼쪽 다리 → VCC
- LED 긴쪽(+) → ESP32 GPIO 핀 (실습별 지정)
- LED 짧은쪽(-) → 330Ω 저항 → GND

## 프로젝트 **A**: 가변저항 값 읽고 시리얼 출력하기

1. 준비
  - 사용 부품: LED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - 가변저항 → GPIO 34번 핀 / LED → GPIO 4번 핀
3. 업로드
  - 코드 스켈레톤(빈칸 채우기)

cpp

```
// 가변저항이 연결된 핀 번호 정의
const int potPin = 34;

void setup() {
    // 시리얼 통신 시작, 모니터와 통신 속도 설정
    Serial.begin(115200);
    Serial.println("가변저항 값 읽기 시작");
}

void loop() {
    // 가변저항 값 아날로그 읽기 (0~4095)
    int potValue = analogRead(potPin);

    // 읽은 값 시리얼로 출력
    Serial.print("가변저항 값: ");
    Serial.println(potValue);

    delay(300); // 0.3초 지연
}
```

4. 관찰
  - 가변저항의 아날로그 값을 읽고, 그 값을 시리얼 모니터에 출력한다.

## 프로젝트 **B**: 가변저항 값 기준으로 **LED ON/OFF** 제어

1. 준비
  - 사용 부품: LED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - 가변저항 → GPIO 34번 핀 / LED → GPIO 4번 핀
3. 업로드
  - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int potPin = 34;
const int ledPin = 4;

void setup() {
    pinMode(ledPin, OUTPUT); // LED 핀을 출력 모드로 설정
    Serial.begin(115200);
    Serial.println("LED ON/OFF 제어 시작");
}

void loop() {
    int potValue = analogRead(potPin);

    Serial.print("가변저항 값: ");
    Serial.println(potValue);

    // 값이 기준 초과 시 LED 켜기, 아니면 끄기
    if (potValue > 2048) {
        digitalWrite(ledPin, HIGH);
        Serial.println("LED ON");
    } else {
        digitalWrite(ledPin, LOW);
        Serial.println("LED OFF");
    }
    delay(200);
}
```

4. 관찰
  - 가변저항 값이 2048을 초과하면 LED를 켜고, 그렇지 않으면 끈다.

## 프로젝트 C: 가변저항으로 LED 밝기 조절 (PWM 사용)

1. 준비
  - 사용 부품: LED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - 가변저항 → GPIO 34번 핀 / LED → GPIO 4번 핀
3. 업로드
  - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int potPin = 34;
const int ledPin = 4;

void setup() {
  Serial.begin(115200);
  // PWM 채널 설정: 채널0, 주파수 5000Hz, 해상도 8비트(0~255)
  ledcSetup(0, 5000, 8);
  ledcAttachPin(ledPin, 0); // 4번 핀과 채널0 연결
  Serial.println("LED 밝기 조절 시작");
}

void loop() {
  int potValue = analogRead(potPin);

  // 0~4095 범위 값을 0~255로 매핑
  int brightness = map(potValue, 0, 4095, 0, 255);

  Serial.print("가변저항 값: ");
  Serial.print(potValue);
  Serial.print(", 밝기 값(PWM): ");
  Serial.println(brightness);

  // PWM으로 LED 밝기 조절
  ledcWrite(0, brightness);

  delay(100);
}
```

4. 관찰
  - 가변저항 값을 읽어 LED 밝기를 부드럽게 조절한다.

## 프로젝트 D: 가변저항 값으로 LED 점멸 속도 조절하기

1. 준비
  - 사용 부품: LED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - 가변저항 → GPIO 34번 핀 / LED → GPIO 4번 핀
3. 업로드
  - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int potPin = 34;
const int ledPin = 4;

void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(115200);
    Serial.println("LED 점멸 속도 조절 시작");
}

void loop() {
    int potValue = analogRead(potPin);

    // 점멸 주기 매핑: 1000ms에서 100ms 사이로 변환
    int interval = map(potValue, 0, 4095, 1000, 100);

    Serial.print("가변저항 값: ");
    Serial.print(potValue);
    Serial.print(", 점멸 간격: ");
    Serial.print(interval);
    Serial.println("ms");

    digitalWrite(ledPin, HIGH);
    delay(interval);
    digitalWrite(ledPin, LOW);
    delay(interval);
}
```

4. 관찰
  - 가변저항 값을 이용해 LED가 점멸하는 속도를 조절한다.

## 프로젝트 E: 가변저항 값 출력과 LED ON/OFF 표시

1. 준비
  - 사용 부품: LED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - 가변저항 → GPIO 34번 핀 / LED → GPIO 4번 핀
3. 업로드
  - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int potPin = 34;
const int ledPin = 4;

void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(115200);
    Serial.println("가변저항 값 출력 및 LED 상태 표시");
}

void loop() {
    int potValue = analogRead(potPin);

    Serial.print("가변저항 값: ");
    Serial.println(potValue);

    if (potValue > 2048) {
        digitalWrite(ledPin, HIGH);
        Serial.println("LED ON");
    } else {
        digitalWrite(ledPin, LOW);
        Serial.println("LED OFF");
    }
    delay(300);
}
```

4. 관찰
  - 시리얼 출력과 LED 제어를 동시에 하며, LED는 가변저항 값 기준으로 켜다  
닫다.