

ESP32 버튼 & LED 제어 실습지시서

준비물

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

공통 회로 연결법

- 버튼 한쪽 다리 → ESP32 GPIO 핀 (실습별 지정)
- 버튼 다른쪽 다리 → GND
- LED 긴쪽(+) → ESP32 GPIO 핀 (실습별 지정)
- LED 짧은쪽(-) → 330Ω 저항 → GND

프로젝트 **A**: 버튼 눌렀을 때 **LED** 켜기/끄기

1. 준비
 - 사용 부품: LED 1개, 푸쉬버튼 1개, 점퍼선, ESP32
2. 배선
 - 버튼 → GPIO ___ 핀 / LED → GPIO ___ 핀
3. 업로드
 - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int buttonPin = ___; // 버튼 핀 번호 입력
const int ledPin = ___;    // LED 핀 번호 입력

void setup() {
  pinMode(buttonPin, INPUT); // 버튼을 입력으로 설정
  pinMode(ledPin, OUTPUT);   // LED를 출력으로 설정
}

void loop() {
  // 버튼 상태 읽기
  if (digitalRead(buttonPin) == HIGH) {
    digitalWrite(ledPin, HIGH); // 버튼 눌림 → LED ON
  } else {
    digitalWrite(ledPin, LOW);  // 버튼 땀 → LED OFF
  }
}
```

4. 관찰
 - 버튼이 눌릴 때 LED가 켜지고 땀 때 LED가 꺼지는 기본 동작 이해하기
5. 기록
 - 버튼 → GPIO ___ 핀
 - LED → GPIO ___ 핀

상세 동작

- `pinMode()` 로 버튼은 입력, LED는 출력으로 설정한다.
- `digitalRead(buttonPin)` 을 통해 버튼이 눌렸는지(높은 상태) 확인한다.
- 버튼이 눌렸으면 LED를 켜다. 아니면 끈다.
- 이 과정을 계속 반복하여 실시간 감지한다.

프로젝트 **B**: 버튼 누를 때마다 **LED** 상태 바꾸기 (토글)

1. 준비
 - 사용 부품: LED 1개, 푸쉬버튼 1개, 점퍼선, ESP32
2. 배선
 - 버튼 → GPIO ____ 핀 / LED → GPIO ____ 핀
3. 업로드
 - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int buttonPin = ____;
const int ledPin = ____;

bool ledState = false;           // LED가 켜져있는지 상태 저장
int lastButtonState = LOW;       // 이전 버튼 상태 저장

void setup() {
    pinMode(buttonPin, INPUT);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    int buttonState = digitalRead(buttonPin);

    // 버튼이 눌린 순간(LOW→HIGH) 감지
    if (buttonState == HIGH && lastButtonState == LOW) {
        ledState = !ledState;      // LED 상태 반전
        digitalWrite(ledPin, ledState ? HIGH : LOW); // LED 켜기/끄기
        delay(300);                // 중복 입력 방지 딜레이
    }

    lastButtonState = buttonState; // 현재 상태를 저장
}
```

4. 관찰
 - 한 번 버튼을 누를 때마다 LED 상태(켜짐/꺼짐)를 반전시키는 토글 동작 구현
5. 기록
 - 버튼 → GPIO ____ 핀
 - LED → GPIO ____ 핀

프로젝트 C: 버튼 누르는 동안 LED 깜빡이기

1. 준비
 - 사용 부품: LED 1개, 푸쉬버튼 1개, 점퍼선, ESP32
2. 배선
 - 버튼 → GPIO ____ 핀 / LED → GPIO ____ 핀
3. 업로드
 - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int buttonPin = ____;
const int ledPin = ____;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // 버튼 눌림 여부 확인
  if (digitalRead(buttonPin) == HIGH) {
    digitalWrite(ledPin, HIGH); // LED ON
    delay(200);                // 0.2초 대기
    digitalWrite(ledPin, LOW);  // LED OFF
    delay(200);                // 0.2초 대기
  } else {
    digitalWrite(ledPin, LOW);  // 버튼 안눌리면 LED OFF 유지
  }
}
```

4. 관찰
 - 버튼을 누르고 있는 동안 LED가 0.2초 간격으로 켜졌다 꺼졌다 반복하도록 하기
5. 기록
 - 버튼 → GPIO ____ 핀
 - LED → GPIO ____ 핀

프로젝트 **D**: 버튼 누르면 **LED** 켜지고 **2초** 후 자동 꺼짐

1. 준비
 - 사용 부품: LED 1개, 푸쉬버튼 1개, 점퍼선, ESP32
2. 배선
 - 버튼 → GPIO ___ 핀 / LED → GPIO ___ 핀
3. 업로드
 - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int buttonPin = ____;
const int ledPin = ____;

bool ledOn = false;           // LED 켜진 상태 저장
unsigned long ledOnTime = 0;   // LED 켜진 시간 저장

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // 버튼 눌릴 때 LED 켜기
  if (digitalRead(buttonPin) == HIGH && !ledOn) {
    digitalWrite(ledPin, HIGH);
    ledOn = true;           // LED 켜진 상태 갱신
    ledOnTime = millis();   // 현재 시간 저장
  }

  // 2초가 지나면 LED 끄기
  if (ledOn && (millis() - ledOnTime > 2000)) {
    digitalWrite(ledPin, LOW);
    ledOn = false;         // LED 꺼진 상태 갱신
  }
}
```

4. 관찰
 - 버튼을 누르면 LED가 켜지고, 2초 지난 후 자동으로 LED 꺼짐 구현
5. 기록
 - 버튼 → GPIO ___ 핀
 - LED → GPIO ___ 핀

프로젝트 E: 버튼 누른 횟수만큼 LED 깜빡이기

1. 준비
 - 사용 부품: LED 1개, 푸쉬버튼 1개, 점퍼선, ESP32
2. 배선
 - 버튼 → GPIO ___ 핀 / LED → GPIO ___ 핀
3. 업로드
 - 코드 스켈레톤(빈칸 채우기)

```
cpp

const int buttonPin = ___;
const int ledPin = ___;

int pressCount = 0;
int lastButtonState = LOW;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH && lastButtonState == LOW) {
    pressCount++;          // 버튼 누른 횟수 증가

    // 누른 횟수만큼 LED 깜빡임
    for (int i = 0; i < pressCount; i++) {
      digitalWrite(ledPin, HIGH);
      delay(200);
      digitalWrite(ledPin, LOW);
      delay(200);
    }
  }

  lastButtonState = buttonState; // 상태 저장
}
```

4. 관찰
 - 버튼 누를 때마다 누른 횟수를 세고, 그 횟수만큼 LED 깜빡임 구현
5. 기록
 - 버튼 → GPIO ___ 핀
 - LED → GPIO ___ 핀