

ESP32 LED 실습 보고서

1. 구성: 준비 → 배선 → 업로드 → 관찰 → 기록 → 응용 체크리스트를 공통 구조로 제공
2. 부품 공통: ESP32-DevKit, LED 1~2개, 220Ω 저항(각 LED 직렬), 브레드보드, 점퍼선, USB 3. 3. 케이블
=> 권장 핀: LED+(애노드)=GPIO26/27, LED-(캐소드)=저항→GND

표지

- 단원명: ESP32 LED 기본 실습
- 반/번호/이름: _____
- 수행일: _____년 __ 월 __ 일

공통 안전/점검 체크

- USB 연결 전 배선 점검(쇼트 없음)
- LED 극성 확인(긴다리=+)
- 220Ω 직렬저항 사용
- GND 공통 연결
- 보드/포트 선택 확인

종합 평가 체크리스트(학생 자기평가)

- LED 극성/저항 개념을 설명할 수 있다.
- 디지털 출력(HIGH/LOW)로 LED를 제어할 수 있다.
- delay 값을 바꿔 깜박임 속도를 조절할 수 있다.
- 두 핀을 제어해 교대 점멸 패턴을 만들 수 있다.
- 버튼 입력(INPUT_PULLUP, 눌림=LOW)을 이해한다.
- 시리얼 명령을 파싱해 조건문으로 분기할 수 있다.
- 소프트 PWM 개념을 설명할 수 있다.

프로젝트 A: 기본 깜박이(Blink)

1. 준비
 - 사용 부품: LED 1개, 220Ω 1개, 점퍼선, ESP32
 - 사용 핀: GPIO26(출력), GND
2. 배선
 - GPIO26 → 220Ω → LED(+) → LED(-) → GND
3. 업로드
 - 코드 핵심: 디지털 출력 HIGH/LOW, delay()로 시간 지연
 - 코드 스켈레톤(빈칸 채우기)

```
#include <Arduino.h>
const int LED_PIN = 26;
void setup() {
    pinMode(LED_PIN, OUTPUT);
}
void loop() {
    digitalWrite(LED_PIN, _____); // LED 켜기(HIGH/LOW?)
    delay(1000);
    digitalWrite(LED_PIN, _____); // LED 끄기(HIGH/LOW?)
    delay(1000);
}
```

4. 관찰
 - 1초 간격으로 켜졌다 꺼진다.
5. 기록
 - 사용 핀 번호: _____
 - 빈칸 답: 켜기= _____, 끄기= _____
 - 동작 주기: 약 _____ 초
6. 응용
 - delay를 200ms/2000ms로 바꾸면 어떻게 보였는가? _____
 - 나만의 주기(예: 300/700ms)를 적용한 결과: _____

프로젝트 B: 두 LED 교대 점멸(Alternating)

1. 준비
 - 사용 부품: LED 2개, 220Ω 2개
 - 사용 핀: GPIO26, GPIO27
2. 배선
 - GPIO26 → 220Ω → LED1(+) → LED1(-) → GND
 - GPIO27 → 220Ω → LED2(+) → LED2(-) → GND
3. 업로드
 - 코드 스켈레톤(빈칸 채우기)

```
#include <Arduino.h>
const int LED1 = 26;
const int LED2 = 27;
void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
}
void loop() {
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, LOW);
  delay(____); // 지연 시간 1
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, HIGH);
  delay(____); // 지연 시간 2
}
```

4. 관찰
 - 두 LED가 번갈아 켜짐.
5. 기록
 - 사용 핀 번호: LED1= , LED2=
 - 지연 시간(ms): /
 - 관찰한 패턴: _____
6. 응용
 - 지연 시간을 다르게 주면 어떤 리듬이 되는가? _____
 - “두 번 빠르게-한 번 느리게” 패턴 구현 시 코드 변경 내용: _____

프로젝트 C: 버튼으로 LED 토글(Toggle)

1. 준비
 - 사용 부품: LED 1개, 220Ω 1개, 버튼 1개
 - 사용 핀: LED=GPIO26, 버튼=GPIO25(INPUT_PULLUP)
2. 배선
 - LED는 프로젝트 A와 동일
 - 버튼 한쪽 → GPIO25, 다른쪽 → GND
 - 내부 풀업 사용(INPUT_PULLUP)
3. 업로드
 - 코드 스켈레톤(빈칸 채우기)

```
#include <Arduino.h>
const int LED_PIN = 26;
const int BTN_PIN = 25;
bool ledState = false;
void setup() {
    pinMode(LED_PIN, OUTPUT);
    pinMode(BTN_PIN, _____); // INPUT_PULLUP / INPUT?
}
void loop() {
    if (digitalRead(BTN_PIN) == _____) { // 눌림 조건: LOW/HIGH?
        ledState = !ledState;
        digitalWrite(LED_PIN, ledState ? HIGH : LOW);
        delay(_____); // 바운싱 방지용 지연(예: 200)
    }
}
```

4. 관찰
 - 누를 때마다 LED ON/OFF 전환.
5. 기록
 - 버튼/LED 핀: _____ / _____
 - 빈칸 답:
 모드= _____,
 눌림조건= _____,
 지연= _____ms
 - 바운싱 현상 관찰(지연 제거 시): _____
6. 응용
 - 지연 대신 시간체크(millis)로 안티바운스 아이디어: _____
 - 길게 누르면 켜기, 짧게 누르면 끄기 같은 변형 아이디어: _____

프로젝트 D: 시리얼 명령으로 2LED 제어(Serial Command)

1. 준비
 - 사용 부품: LED 2개, 220Ω 2개
 - 사용 핀: GPIO26, GPIO27
2. 배선
 - 프로젝트 B와 동일
3. 업로드
 - 시리얼 모니터 Line ending을 "Newline(또는 NL)"로 설정
 - 코드 스켈레톤(빈칸 채우기)

```
#include <Arduino.h>
const int BLUE = 26;
const int GREEN = 27;
void setup() {
    Serial.begin(115200);
    pinMode(BLUE, OUTPUT);
    pinMode(GREEN, OUTPUT);
    Serial.println("Commands: blueON, blueOFF, greenON, greenOFF");
}
void loop() {
    if (Serial.available()) {
        String cmd = Serial.readStringUntil(____); // '\n'
        cmd.trim();
        if (cmd == "blueON")    digitalWrite(BLUE, HIGH);
        else if (cmd == "blueOFF") digitalWrite(BLUE, LOW);
        else if (cmd == "greenON") digitalWrite(GREEN, HIGH);
        else if (cmd == "greenOFF") digitalWrite(GREEN, LOW);
        else Serial.println("Unknown: " + cmd);
    }
}
```

4. 관찰
 - 명령어에 따라 LED가 동작.
5. 기록
 - 빈칸 답(줄바꿈 문자): _____
6. 응용
 - allON, allOFF, toggleBlue, speed=300 같은 커스텀 명령 추가

프로젝트 E: LED 디밍(소프트 PWM, 밝기 천천히 변화)

1. 준비
 - 사용 부품: LED 1개, 220Ω 1개
 - 사용 핀: GPIO26
2. 배선
 - 프로젝트 A와 동일
3. 업로드
 - 코드 스켈레톤(빈칸 채우기, 쉬운 소프트 PWM)

```
#include <Arduino.h>
const int LED_PIN = 26;
void setup() { pinMode(LED_PIN, OUTPUT); }
void loop() {
    for (int duty = 0; duty <= 10; duty++) { // (1) 최대 단계
        digitalWrite(LED_PIN, HIGH);
        delay(duty * ____); // (2) on 시간 배수(예: 10)
        digitalWrite(LED_PIN, LOW);
        delay((10 - duty) * ____); // (3) off 시간 배수(예: 10)
    }
    for (int duty = 10; duty >= 0; duty--) {
        digitalWrite(LED_PIN, HIGH);
        delay(duty * ____);
        digitalWrite(LED_PIN, LOW);
        delay((10 - duty) * ____);
    }
}
```

4. 관찰
 - LED가 서서히 밝아졌다 어두워짐.
5. 기록
 - 빈칸 답: (1)= _____, (2)= _____, (3)= _____
 - 느낀 밝기 변화 속도/부드러움: _____
 - 문제/해결: _____
6. 응용
 - 실제 PWM(ledcWrite) 사용 시 장점(더 부드럽고 CPU 점유 적음) 메모: _____
 - 단계 수(10 → 50) 변경 시 체감 변화: _____