ESP32 버튼 **& 3**색 **RGB-LED** 제어 실습지시서

준비물

- ESP32 개발 보드 (예: ESP32 Dev Module)
- 버튼 1개
- 3색 RGB-LED 1개
- 330Ω 저항 3개 (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:** 자동 색상 순환기

1. 준비
   - 사용 부품: 3색 RGB-LED 1개, 점퍼선, ESP32

2. 배선
   - 3색 RGB-LED의 R:25번, G:26번, B:27번

3. 업로드
   - 코드 스켈레톤

```
#include <Arduino.h>   // Arduino 기본 라이브러리 포함

// RGB LED 핀 번호 정의
const int R_PIN = 25;  // 빨강(R) LED → GPIO 25번 핀
const int G_PIN = 26;  // 초록(G) LED → GPIO 26번 핀
const int B_PIN = 27;  // 파랑(B) LED → GPIO 27번 핀

void setup() {
  // 각 핀을 PWM 채널에 연결
  ledcAttachPin(R_PIN, 0);  // R_PIN을 PWM 채널 0번에 연결
  ledcAttachPin(G_PIN, 1);  // G_PIN을 PWM 채널 1번에 연결
  ledcAttachPin(B_PIN, 2);  // B_PIN을 PWM 채널 2번에 연결

  // 각 PWM 채널 설정 (채널번호, 주파수 5000Hz, 해상도 8비트)
  // 해상도 8비트 → 0 ~ 255 값으로 밝기 제어 가능
  ledcSetup(0, 5000, 8);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
}
```

```
void loop() {
  // ------------------------------
  // 1단계: 빨강 LED 점점 켜짐
  // ------------------------------
  for (int i = 0; i <= 255; i++) {
    ledcWrite(0, 255 - i);  // 빨강: 255(꺼짐) → 0(최대 밝기)
    ledcWrite(1, 255);      // 초록: 꺼짐
    ledcWrite(2, 255);      // 파랑: 꺼짐
    delay(10);              // 10ms 대기 (부드럽게 변화)
  }

  // ------------------------------
  // 2단계: 초록 LED 점점 켜짐
  // ------------------------------
  for (int i = 0; i <= 255; i++) {
    ledcWrite(0, 255);      // 빨강: 꺼짐
    ledcWrite(1, 255 - i);  // 초록: 255(꺼짐) → 0(최대 밝기)
    ledcWrite(2, 255);      // 파랑: 꺼짐
    delay(10);              // 10ms 대기
  }
```

```
  // ------------------------------
  // 3단계: 파랑 LED 점점 켜짐
  // ------------------------------
  for (int i = 0; i <= 255; i++) {
    ledcWrite(0, 255);      // 빨강: 꺼짐
    ledcWrite(1, 255);      // 초록: 꺼짐
    ledcWrite(2, 255 - i);  // 파랑: 255(꺼짐) → 0(최대 밝기)
    delay(10);              // 10ms 대기
  }
}
```

4. 관찰

- RGB LED가 자동으로 빨강 → 초록 → 파랑 순서로 천천히 변화

- `ledcWrite()`는 ESP32의 PWM 출력 함수이며, 0~255 값으로 LED 밝기를 제어
- 공통 Vcc(아노드) LED이므로 값이 `255 - i` 형태로 반전
- 결과적으로, 계속 순환하는 무드등 효과가 남

프로젝트 **B:** 버튼 기반 색상 전환기

1. 준비
   - 사용 부품: 3색 RGB-LED 1개, 버튼 1개, 점퍼선, ESP32

2. 배선
   - 3색 RGB-LED의 R:25번, G:26번, B:27번, 버튼: 4번

3. 업로드
   - 코드 스켈레톤

```
#include <Arduino.h>   // 아두이노 기본 라이브러리

// RGB LED 핀 정의
const int R_PIN = 25;
const int G_PIN = 26;
const int B_PIN = 27;

// 버튼 입력 핀
const int BUTTON = 4;  // 버튼을 GPIO 4번 핀에 연결

// 현재 LED 색상 모드를 저장하는 변수
int mode = 0;

// ------------------------------------
// RGB LED 색상 설정 함수
// 매개변수 r,g,b 값(0~255)을 받아 LED 제어
// 공통양극 RGB LED를 가정하여 255-r 형태로 출력
// ------------------------------------
void setColor(int r, int g, int b) {
  ledcWrite(0, 255 - r);  // R 채널
  ledcWrite(1, 255 - g);  // G 채널
  ledcWrite(2, 255 - b);  // B 채널
}
```

```
void setup() {
  pinMode(BUTTON, INPUT_PULLUP);  // 버튼을 풀업 입력으로 설정 (눌리면 LOW)

  // RGB LED 핀을 PWM 채널에 연결
  ledcAttachPin(R_PIN, 0);  // R_PIN → 채널 0
  ledcAttachPin(G_PIN, 1);  // G_PIN → 채널 1
  ledcAttachPin(B_PIN, 2);  // B_PIN → 채널 2

  // PWM 채널 설정 (주파수 5000Hz, 해상도 8비트)
  ledcSetup(0, 5000, 8);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
}
```

```
void loop() {
  // ----------------------------
  // 버튼 눌림 감지
  // ----------------------------
  if (digitalRead(BUTTON) == LOW) {   // 버튼이 눌려서 LOW 신호가 들어오면
    mode = (mode + 1) % 4;            // mode 값을 0~3 사이에서 순환
    delay(200);                       // 디바운스(버튼 튀는 현상 방지)
  }

  // ----------------------------
  // mode 값에 따라 색상 변경
  // ----------------------------
  if (mode == 0) setColor(255, 0, 0);          // 빨강
  else if (mode == 1) setColor(0, 255, 0);     // 초록
  else if (mode == 2) setColor(0, 0, 255);     // 파랑
  else if (mode == 3) setColor(255, 255, 255); // 흰색
}
```

4. 관찰

- 버튼을 누를 때마다 LED 색상이 빨강 → 초록 → 파랑 → 흰색 순서
- `mode` 변수를 이용해 버튼 누를 때마다 값이 증가하며, 나머지 연산 `%4`로 순환
- `INPUT_PULLUP` 모드이므로 버튼은 GPIO4 ↔ GND로 연결
- 버튼 입력 처리, 디바운스, 상태 변수를 배우기에 적합

프로젝트 C:온보드 버튼(BOOT 버튼) 활용

1. 준비
   - 사용 부품: 3색 RGB-LED 1개, 점퍼선, ESP32

2. 배선
   - 3색 RGB-LED의 R:25번, G:26번, B:27번

3. 업로드
   - 코드 스켈레톤

```
#include <Arduino.h>   // 아두이노 기본 라이브러리 포함

// RGB LED 핀 번호 정의
const int R_PIN = 25;
const int G_PIN = 26;
const int B_PIN = 27;

// ESP32 보드의 기본 BOOT 버튼 (GPIO 0)
const int BOOT_BUTTON = 0;

// 현재 색상 모드 저장 변수
int colorMode = 0;

// ------------------------------------
// RGB 색상 설정 함수
// r, g, b 값(0~255)을 받아 LED PWM 출력
// 공통양극 RGB LED 기준: 0=최대 밝기, 255=꺼짐
// ------------------------------------
void setColor(int r, int g, int b) {
  ledcWrite(0, 255 - r);  // R 채널
  ledcWrite(1, 255 - g);  // G 채널
  ledcWrite(2, 255 - b);  // B 채널
}
```

```ino
void setup() {
  // 버튼을 풀업 입력으로 설정
  // (누르지 않으면 HIGH, 누르면 GND와 연결되어 LOW)
  pinMode(BOOT_BUTTON, INPUT_PULLUP);

  // RGB LED를 PWM 채널에 연결
  ledcAttachPin(R_PIN, 0);  // 빨강 → 채널 0
  ledcAttachPin(G_PIN, 1);  // 초록 → 채널 1
  ledcAttachPin(B_PIN, 2);  // 파랑 → 채널 2

  // PWM 채널 설정 (주파수: 5000Hz, 해상도: 8비트)
  ledcSetup(0, 5000, 8);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
}
```

```ino
void loop() {
  // ----------------------------
  // 버튼 눌림 감지
  // ----------------------------
  if (digitalRead(BOOT_BUTTON) == LOW) {   // 버튼 눌리면 LOW 신호
    colorMode = (colorMode + 1) % 3;       // 모드 0~2 순환
    delay(300);                            // 디바운스 & 누른 상태 중복 입력 방지
  }

  // ----------------------------
  // colorMode 값에 따른 색상 변경
  // ----------------------------
  if (colorMode == 0) setColor(255, 0, 0);    // 빨강
  else if (colorMode == 1) setColor(0, 255, 0); // 초록
  else if (colorMode == 2) setColor(0, 0, 255); // 파랑
}
```

4. 관찰
   - ESP32 보드에 기본으로 달린 **BOOT** 버튼을 사용
   - BOOT 버튼을 누를 때마다 LED 색이 빨강 → 초록 → 파랑 순서
   - 별도의 외부 버튼을 연결하지 않아도 된다는 장점

프로젝트 **D:** 랜덤 색상 생성기

1. 준비
   - 사용 부품: 3색 RGB-LED 1개, 점퍼선, ESP32

2. 배선
   - 3색 RGB-LED의 R:25번, G:26번, B:27번

3. 업로드
   - 코드 스켈레톤

```
#include <Arduino.h>


// RGB LED 핀 정의
const int R_PIN = 25;
const int G_PIN = 26;
const int B_PIN = 27;


// ------------------------------------
// RGB 색상 출력 함수
// 매개변수 r, g, b (0~255) 값 입력
// 공통양극 RGB LED 기준 → (0=최대 밝기, 255=꺼짐)
// ------------------------------------
void setColor(int r, int g, int b) {
  ledcWrite(0, 255 - r);  // R 채널
  ledcWrite(1, 255 - g);  // G 채널
  ledcWrite(2, 255 - b);  // B 채널
}
```

```
void setup() {
  // 랜덤 시드 초기화
  // ESP32에서는 보통 analogRead(0) 값이 잡음(noise)을 포함해 난수 초기화에 사용됨
  randomSeed(analogRead(0));

  // RGB LED를 PWM 채널에 연결
  ledcAttachPin(R_PIN, 0);
  ledcAttachPin(G_PIN, 1);
  ledcAttachPin(B_PIN, 2);

  // PWM 채널 설정 (주파수: 5000Hz, 해상도: 8비트)
  ledcSetup(0, 5000, 8);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
}
```

```
void loop() {
  // ------------------------------------
  // 무작위 색상 생성
  // random(0, 256) → 0~255 값 생성
  // ------------------------------------
  int r = random(0, 256);  // 빨강 값
  int g = random(0, 256);  // 초록 값
  int b = random(0, 256);  // 파랑 값

  // RGB LED 색상 적용
  setColor(r, g, b);

  // 2초 유지 후 다음 색상으로 변경
  delay(2000);
}
```

4. 관찰
   - 전원이 켜지고 실행될 때마다 무작위 RGB 값이 생성되어 LED 색상이 변화
   - 2초마다 새로운 색상으로 변경
   - `randomSeed(analogRead(0))`은 ESP32의 아날로그 노이즈를 이용해 매번 다른 패턴을 생성

프로젝트 **E:** 타이머 기반 알람 **LED**

1. 준비
   - 사용 부품: 3색 RGB-LED 1개, 점퍼선, ESP32

2. 배선
   - 3색 RGB-LED의 R:25번, G:26번, B:27번

3. 업로드
   - 코드 스켈레톤

```
#include <Arduino.h>

// RGB LED 핀 번호 정의
const int R_PIN = 25;
const int G_PIN = 26;
const int B_PIN = 27;

// 프로그램 시작 시각 저장 변수
unsigned long startTime;

// ------------------------------------
// RGB 색상 출력 함수
// r, g, b (0~255) 입력
// 공통양극 RGB LED 기준: 0=최대 밝기, 255=꺼짐
// ------------------------------------
void setColor(int r, int g, int b) {
  ledcWrite(0, 255 - r);  // R 채널
  ledcWrite(1, 255 - g);  // G 채널
  ledcWrite(2, 255 - b);  // B 채널
}
```

```ino
void setup() {
  // 시작 시각 저장
  startTime = millis();

  // RGB LED 핀을 PWM 채널에 연결
  ledcAttachPin(R_PIN, 0);
  ledcAttachPin(G_PIN, 1);
  ledcAttachPin(B_PIN, 2);

  // PWM 채널 설정 (주파수: 5000Hz, 해상도: 8비트)
  ledcSetup(0, 5000, 8);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
}
```

```ino
void loop() {
  // 경과 시간 계산 (초 단위)
  unsigned long elapsed = (millis() - startTime) / 1000;

  // ----------------------------
  // 시간에 따른 LED 제어
  // ----------------------------
  if (elapsed < 5) {
    setColor(0, 0, 255);        // 0~5초: 파랑
  }
  else if (elapsed < 10) {
    setColor(0, 255, 0);        // 5~10초: 초록
  }
  else {
    // 10초 이후: 빨강 LED 깜빡임
    if ((elapsed % 2) == 0)
      setColor(255, 0, 0);      // 짝수 초 → 빨강 켜기
    else
      setColor(0, 0, 0);        // 홀수 초 → LED 끄기
  }
}
```

4. 관찰

- 실행 후 시간에 따라 LED 색상이 변화
  0~5초: 파랑/ 5~10초: 초록/ 10초 이후: 빨강이 1초마다 깜빡임
- `millis()`를 사용하여 비차단 타이머 방식으로 구현
- 간단한 시각적 알람/타이머 프로젝트로 응용 가능