

# ESP32 OLED & 가변저항 제어 실습지시서

## 준비물

- ESP32 개발 보드 (예: ESP32 Dev Module)
- OLED 1개
- 가변저항 1개
- 브레드보드와 점퍼 와이어
- USB 케이블 (ESP32와 PC 연결용)

## 공통 회로 연결법

- OLED SDA 핀 → D21 핀
- OLED SCL 핀 → D22 핀
- OLED 접지 → GND
- OLED 전원 → VCC
- 가변저항 → D34 핀

## 소프트웨어 라이브러리 설정

- OLED를 위한 라이브러리 설정

=> platformio.ini 파일에 다음과 같은 내용 추가

lib\_deps =

adafruit/Adafruit SSD1306@^2.5.9

adafruit/Adafruit GFX Library@^1.11.5

## 프로젝트 A: OLED 기본 텍스트 출력 시스템

1. 준비
  - 사용 부품: OLED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - OLED(SDA): 21번, OLED(SCL): 22번, 가변저항: 34번
3. 업로드
  - 코드 스켈레톤

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED 가로 픽셀 수
#define SCREEN_HEIGHT 64 // OLED 세로 픽셀 수
#define OLED_RESET -1 // 리셋 핀 미사용
#define SCREEN_ADDRESS 0x3C // I2C 주소 (0x3C)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS); // 디스플레이 초기화
  display.clearDisplay(); // 화면 지우기

  display.setTextSize(2); // 글자 크기 설정 (2배)
  display.setTextColor(SSD1306_WHITE); // 글자 색상: 흰색
  display.setCursor(10, 20); // 출력 위치 설정 (x=10, y=20)
  display.println("Hello!"); // 문자 출력
  display.display(); // 실제 화면에 표시
}

void loop() {
  // 반복 동작 없음
}
```

4. 관찰
  - ESP32와 OLED를 연결하여 화면에 "Hello!"라는 문구를 표시하는 예제
  - OLED에 고정된 문자열("Hello!")을 한 번 표시하고 끝나는 프로그램

## 프로젝트 B: OLED 글자 크기 비교 출력 시스템

1. 준비
  - 사용 부품: OLED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - OLED(SDA): 21번, OLED(SCL): 22번, 가변저항: 34번
3. 업로드
  - 코드 스켈레톤

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS); // OLED 초기화
  display.clearDisplay();
}

void loop() {
  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE);

  display.setTextSize(1); // 글자 크기 1
  display.setCursor(0, 0);
  display.println("Size 1");

  display.setTextSize(2); // 글자 크기 2
  display.setCursor(0, 16);
  display.println("Size 2");

  display.setTextSize(3); // 글자 크기 3
  display.setCursor(0, 40);
  display.println("Size 3");

  display.display(); // 출력 반영

  delay(5000); // 5초 대기 후 반복
}
```

4. 관찰

- 텍스트 크기를 1, 2, 3단계로 바꿔가며 한 화면에 출력.
- OLED의 글자 크기 변화 효과를 학습 예제.
- 한 화면에 서로 다른 글자 크기를 보여주는 예제.

## 프로젝트 C: OLED와 가변저항 값 읽기 표시 시스템

1. 준비
  - 사용 부품: OLED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - OLED(SDA): 21번, OLED(SCL): 22번, 가변저항: 34번
3. 업로드
  - 코드 스켈레톤

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C
#define POTENTIOMETER_PIN 34 // 가변저항 연결 핀 (D34)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS); // OLED 초기화
  display.clearDisplay();
  pinMode(POTENTIOMETER_PIN, INPUT); // 아날로그 입력 설정
}

void loop() {
  int potValue = analogRead(POTENTIOMETER_PIN); // 0~4095 범위 값 읽기

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 10);
  display.println("Potentiometer Value:");

  display.setTextSize(2);
  display.setCursor(0, 30);
  display.println(potValue); // 측정값 출력

  display.display();
  delay(200); // 0.2초마다 업데이트
}
```

4. 관찰

- 가변저항(포텐시오미터)의 아날로그 값을 읽어서 OLED에 수치로 표시.
- 아날로그 입력의 변화를 눈으로 확인.
- 가변저항을 돌리면 OLED 화면의 숫자가 변화함.

## 프로젝트 D: OLED와 가변저항으로 시각화 시스템

1. 준비
  - 사용 부품: OLED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - OLED(SDA): 21번, OLED(SCL): 22번, 가변저항: 34번
3. 업로드
  - 코드 스켈레톤

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define POT_PIN 34 // 가변저항 연결 핀

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
}

void loop() {
  int val = analogRead(POT_PIN); // 0~4095 읽기
  int percent = map(val, 0, 4095, 0, 100); // %로 환산
  int barLength = map(val, 0, 4095, 0, 128); // 막대 길이 계산

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.print("Brightness: ");
  display.print(percent);
  display.println("%");

  display.drawRect(0, 20, 128, 10, SSD1306_WHITE); // 그래프 틀
  display.fillRect(0, 20, barLength, 10, SSD1306_WHITE); // 내부 막대 채우기

  display.display();
  delay(200);
}
```

4. 관찰

- 가변저항 값을 0~100%로 변환하여 숫자와 함께 막대 그래프로 표현.
- 시각적으로 직관적인 값을 표시하는 예제.
- 가변저항을 돌리면 OLED에 퍼센트와 막대 그래프가 실시간으로 변함.



## 프로젝트 E: OLED와 가변저항을 이용한 사인파 출력

1. 준비
  - 사용 부품: OLED 1개, 가변저항 1개, 점퍼선, ESP32
2. 배선
  - OLED(SDA): 21번, OLED(SCL): 22번, 가변저항: 34번
3. 업로드
  - 코드 스켈레톤

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <math.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
int potPin = 34;

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
}

void loop() {
  int potVal = analogRead(potPin);          // 가변저항 값 읽기 (0~4095)
  int amplitude = map(potVal, 0, 4095, 10, 30); // 진폭 범위 10~30으로 변환

  display.clearDisplay();

  for (int x = 0; x < SCREEN_WIDTH; x++) {
    float angle = (float)x / SCREEN_WIDTH * 2.0 * 3.14159 * 4; // 주기 4번 생성
    int y = SCREEN_HEIGHT / 2 + amplitude * sin(angle);          // 중앙 기준 사인파
    display.drawPixel(x, y, SSD1306_WHITE);                      // 픽셀 그리기
  }

  display.display();
  delay(50);
}
```

4. 관찰
  - 가변저항 값으로 사인파의 높이(amplitude)를 조절하여 OLED 화면에 움직이는 파형처럼 출력.
  - 수학 함수와 센서값을 결합한 시각화 예제.
  - OLED에서 사인파 형태를 표시하며, 가변저항값으로 파형의 높이 변경.