

ESP32 OLED & 가변저항 제어-2 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 볼륨 표시 바 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 스케레톤

```
#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 입력 핀

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

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // OLED 초기화
  display.clearDisplay(); // 화면 초기화
}

void loop() {
  int val = analogRead(POT_PIN); // 가변저항 값 읽기 (0~4095)
  int volume = map(val, 0, 4095, 0, 100); // 0~100%로 변환
  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("Volume: ");
  display.print(volume);
  display.println("%");

  // 볼륨 바 그리기 (테두리와 채우기)
  display.drawRect(0, 20, 128, 15, SSD1306_WHITE);
  display.fillRect(0, 20, barLength, 15, SSD1306_WHITE);

  display.setCursor(0, 40);
  if(volume == 0) display.println("Mute");
  else if(volume < 70) display.println("Normal");
  else display.println("Max");
```

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

4. 관찰

- 가변저항을 돌리면 OLED에 **볼륨 수치(%)**와 **막대그래프**가 표시됨.
- 볼륨이 **0%면 "Mute"**, **70% 이상이면 "Max"**, 그 외엔 "Normal"로 출력됨.
- 시각적인 인터페이스로 볼륨 컨트롤 구현 가능.

프로젝트 B: OLED 신호 세기 표시 시스템

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

```
#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 핀

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

// 신호 세기 막대 그리기 함수
void drawSignalBars(int level) {
  int barHeights[5] = {4, 8, 12, 16, 20}; // 각 막대 높이
  int x_start = 10;
  int y_base = 40;

  for(int i = 0; i < 5; i++) {
    int x = x_start + i * 20;
    int y = y_base - barHeights[i];
    if(i < level) {
      display.fillRect(x, y, 10, barHeights[i], SSD1306_WHITE);
    } else {
      display.drawRect(x, y, 10, barHeights[i], SSD1306_WHITE);
    }
  }
}

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

void loop() {
  int val = analogRead(POT_PIN);
  int level = map(val, 0, 4095, 0, 5); // 신호 단계 0~5
  int percent = map(val, 0, 4095, 0, 100); // 퍼센트 변환

  display.clearDisplay();
```

```
display.setTextSize(1);  
display.setTextColor(SSD1306_WHITE);  
display.setCursor(0,0);  
display.print("Signal Strength: ");  
display.print(percent);  
display.println("%");  
  
drawSignalBars(level); // 신호 막대 출력  
  
display.display();  
delay(200);  
}
```

4. 관찰

- 가변저항의 위치에 따라 **신호 세기(0~100%)** 표시.
- 막대 5개 중 몇 개를 채워서 현재 신호 강도를 시각적으로 표현.
- **Wi-Fi 신호 표시 스타일**을 연습하기 좋은 예제.

프로젝트 C: OLED와 비트맵 이미지 활용 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>      // 그래픽 디스플레이용 기본 라이브러리
#include <Adafruit_SSD1306.h>  // SSD1306 OLED 디스플레이용 라이브러리

#define SCREEN_WIDTH 128      // OLED 가로 해상도 (픽셀)
#define SCREEN_HEIGHT 64     // OLED 세로 해상도 (픽셀)
#define OLED_RESET -1        // 리셋 핀 사용 안 함
#define POT_PIN 34           // 가변저항(Potentiometer) 연결 핀 (ADC 입력)

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

// 16x16 외계인 비트맵 이미지 (PROGMEM: 플래시 메모리에 저장)
const unsigned char alienBitmap[] PROGMEM = {
  0b00011000, 0b00011000,
  0b00111100, 0b00111100,
  0b01111110, 0b01111110,
  0b11111111, 0b11111111,
  0b11100111, 0b11100111,
  0b11111111, 0b11111111,
  0b10111101, 0b10111101,
  0b00100100, 0b00100100,
  0b01000010, 0b01000010,
  0b10000001, 0b10000001,
  0b01000010, 0b01000010,
  0b00100100, 0b00100100,
  0b00011000, 0b00011000,
  0b00100100, 0b00100100,
  0b01000010, 0b01000010,
  0b10000001, 0b10000001
};

void setup() {
  // OLED 디스플레이 초기화 (I2C 주소: 0x3C)
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
}

void loop() {
  int val = analogRead(POT_PIN); // 가변저항 값 읽기 (0~4095)
```

```

// 가변저항 값을 x 좌표로 매핑 (0~(128-16)=112)
int x = map(val, 0, 4095, 0, SCREEN_WIDTH - 16);

int y = 24; // y 좌표 고정 (화면 수직 중앙 근처)

display.clearDisplay(); // 화면 지우기
// 외계인 출력
display.drawBitmap(x, y, alienBitmap, 16, 16, SSD1306_WHITE);
display.display(); // 화면에 실제로 그림 표시

delay(50); // 너무 빠른 갱신 방지 (약간의 딜레이)
}

```

4. 관찰

- 가변저항을 돌리면 **외계인 이미지가 좌우로 이동**.
- 비트맵 처리, 이미지 출력 및 애니메이션 개념 학습에 적합.

프로젝트 D: OLED를 활용한 슬라이드 잠금 해제 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 핀

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

// 슬라이더 그리는 함수
void drawSlider(int pos) {
  display.drawRect(10, 30, 108, 20, SSD1306_WHITE); // 슬라이더 영역
  display.fillRect(pos, 32, 10, 16, SSD1306_WHITE); // 이동하는 바
}

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

void loop() {
  int val = analogRead(POT_PIN);
  int pos = map(val, 0, 4095, 10, 108); // 바의 위치

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,0);
  display.print("Slide to Unlock:");

  drawSlider(pos);

  if(pos >= 100) { // 거의 끝까지 이동 시
    display.setCursor(0, 55);
    display.println("Unlocked!");
  }
```



```
display.display();  
delay(150);  
}
```

4. 관찰

- 가변저항을 오른쪽 끝까지 돌리면 **"Unlocked!"** 표시.
- 실제 잠금 해제 UI와 유사한 기능을 구현.
- 인터랙티브 UI 제작에 활용 가능.

프로젝트 E: OLED를 이용한 캐릭터 선택기

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

```
#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 핀

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

// 캐릭터 배열
const char* characters[] = {"Cat", "Dog", "Panda", "Alien", "Robot", "Ghost", "Frog", "Lion"};
const int charCount = sizeof(characters) / sizeof(characters[0]);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  randomSeed(analogRead(0)); // 무작위용 초기값
}

void loop() {
  int val = analogRead(POT_PIN);
  int index = map(val, 0, 4095, 0, charCount - 1); // 인덱스 선택

  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE);

  display.setTextSize(1);
  display.setCursor(0, 0);
  display.print("Your Character:");

  display.setTextSize(3);
  display.setCursor(0, 20);
  display.print(characters[index]); // 캐릭터 이름 표시

  display.display();
  delay(300);
}
```

4. 관찰

- 가변저항을 돌리면 8개의 캐릭터 중 하나가 선택됨.
- 게임 캐릭터 선택 화면 UI처럼 활용 가능.
- 선택한 문자열을 화면에 큼지막하게 출력.