

ESP32 OLED 종합 제어 실습지시서

준비물

- ESP32 개발 보드 (예: ESP32 Dev Module)
- OLED 1개
- CDS 1개
- LED 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
- CDS → D34 핀
- LED → D2 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: OLED를 이용한 CDS 값 출력 시스템

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

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

#define CDS_PIN 34 // CDS 조도센서 연결 핀
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

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

void setup() {
  Serial.begin(115200);
  pinMode(CDS_PIN, INPUT); // CDS 핀을 입력으로 설정

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // OLED 초기화
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  int cdsValue = analogRead(CDS_PIN); // 빛의 세기 측정 (0~4095)

  display.clearDisplay();
  display.setCursor(0, 20);
  display.print("CDS: ");
  display.println(cdsValue); // OLED에 CDS 값 출력
  display.display();

  delay(500);
}
```

4. 관찰
 - CDS 센서의 조도값을 실시간으로 OLED에 숫자로 출력.
 - **밝을수록 값이 높아지고**, 어두우면 값이 낮아짐.
 - Serial 출력도 병행 가능.

프로젝트 B: 조도에 따른 LED 및 OLED 상태 표시 시스템

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

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

#define CDS_PIN 34
#define LED_PIN 2 // 내장 LED 또는 외부 LED 핀

#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() {
  pinMode(CDS_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);

  display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  int cdsValue = analogRead(CDS_PIN); // 조도 값 읽기

  display.clearDisplay();
  display.setCursor(0, 0);
  display.print("CDS Value: ");
  display.println(cdsValue);

  if (cdsValue < 1000) { // 어두운 조건
    digitalWrite(LED_PIN, HIGH); // LED 켜기
    display.setCursor(0, 20);
    display.println("LED: ON (Dark)");
  } else { // 밝은 조건
    digitalWrite(LED_PIN, LOW); // LED 끄기
    display.setCursor(0, 20);
```

```
    display.println("LED: OFF (Bright)");  
  }  
  
  display.display(); // OLED 출력  
  delay(500);  
}
```

4. 관찰

- 조도값이 1000 이하(어두움)이면 LED를 켜고, OLED에 "LED: ON (Dark)" 출력.
- 1000 이상(밝음)이면 LED를 끄고 "LED: OFF (Bright)" 출력.
- 자동 조명 제어 시나리오를 구현한 예제.

프로젝트 C: OLED를 활용한 메시지 출력 시스템

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

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

#define CDS_PIN 34
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

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

void setup() {
  pinMode(CDS_PIN, INPUT);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  int cdsValue = analogRead(CDS_PIN); // 조도 읽기
  display.clearDisplay();

  if (cdsValue < 1000) { // 어두운 경우
    display.setCursor(0, 20);
    display.setTextSize(2);
    display.println("Good Night");
  } else { // 밝은 경우
    display.setCursor(0, 20);
    display.setTextSize(1);
    display.println("Good Morning");
  }

  display.display();
  delay(500);
}
```

4. 관찰

- 어두울 때는 **"Good Night"**, 밝을 때는 **"Good Morning"** 이 OLED에 표시.
- 텍스트 크기 변화를 통해 시각적 강조도 함께 표현.

프로젝트 D: 조도센서 이용한 LED 및 메시지 출력 시스템

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

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

#define CDS_PIN 34
#define LED_PIN 2
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

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

void setup() {
  pinMode(CDS_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  int cdsValue = analogRead(CDS_PIN);
  display.clearDisplay();

  if (cdsValue < 1000) {
    digitalWrite(LED_PIN, HIGH);
    display.setCursor(0, 20);
    display.println("Good Night :)");
  } else {
    digitalWrite(LED_PIN, LOW);
    display.setCursor(0, 20);
    display.println("LED OFF");
  }
  display.display();
  delay(500);
}
```

4. 관찰

- 어두울 때 LED를 켜고 "Good Night :)" 메시지를 출력.
- 밝을 때는 LED를 끄고 "LED OFF" 메시지 출력.
- A, B, C의 기능을 조합한 응용형.

프로젝트 E: 사인파, 조도기반 PWM LED 출력 시스템

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

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

#define CDS_PIN 34
#define LED_PIN 2
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define OLED_ADDR 0x3C

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

const int graphHeight = 40;
const int graphWidth = SCREEN_WIDTH;

float angle = 0;
float baseFreq = 0.05;
float freq = baseFreq;

void setup() {
  pinMode(CDS_PIN, INPUT);

  // LED PWM 설정 (채널 0, 주파수 5kHz, 해상도 8bit)
  ledcSetup(0, 5000, 8);
  ledcAttachPin(LED_PIN, 0);

  display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  int cdsValue = analogRead(CDS_PIN);

  // CDS 값에 따라 사인파 주파수 조절 (0.01 ~ 0.2)
  freq = map(cdsValue, 0, 4095, 1, 20) / 100.0;
```

```

// 사인파 계산
float sineValue = sin(angle);
int pwmVal = (int)(abs(sineValue) * 255.0);
ledcWrite(0, pwmVal); // LED 밝기 제어

display.clearDisplay();
display.setCursor(0, 0);
display.print("CDS: "); display.println(cdsValue);
display.print("Freq: "); display.println(freq, 3);
display.print("PWM: "); display.println(pwmVal);

// OLED에 사인파 그래프 그리기
for (int x = 0; x < graphWidth; x++) {
    float val = sin(angle + x * freq);
    int yPos = (int)((val + 1.0) * (graphHeight / 2));
    display.drawPixel(x, graphHeight - yPos + 20, SSD1306_WHITE);
}

display.display();

angle += freq;
if (angle > 2 * PI) angle -= 2 * PI;

delay(30);
}

```

4. 관찰

- 조도 센서(CDS) 값 읽기: 주변이 밝으면 값이 커지고, 어두우면 작아짐.
- 사인파 주파수 조절: CDS 값에 따라 사인파의 주파수를 바꿉니다.
→ 밝을수록 사인파가 더 빠르게 변화함.
- LED 밝기 조절: 사인파의 절댓값을 이용해 LED를 부드럽게 깜빡이도록(PWM) 제어.
- OLED 화면 표시: CDS 센서값, 주파수, 현재 LED 밝기(PWM 값)를 표시하고, 아래쪽에 실시간 사인파 그래프를 그림.