

ESP32 온습도센서 & LED 제어 실습지시서

준비물

- ESP32 개발 보드 (예: ESP32 Dev Module)
- LED 1개
- 온습도센서 1개
- 330Ω 저항 1개 (LED 보호용)
- 브레드보드와 점퍼 와이어
- USB 케이블 (ESP32와 PC 연결용)

공통 회로 연결법

- 온습도센서 신호 → ESP32 GPIO 핀 (실습별 지정)
- 온습도센서 접지 → GND
- 온습도센서 전원 → VCC
- LED 긴쪽(+) → ESP32 GPIO 핀 (실습별 지정)
- LED 짧은쪽(-) → 330Ω 저항 → GND

소프트웨어 라이브러리 설정

- 온습도센서를 위한 라이브러리 설정
=> **platformio.ini** 파일에 다음과 같은 내용 추가
lib_deps = adafruit/DHT sensor library

프로젝트 A: 온습도값 시리얼 출력

1. 준비
 - 사용 부품: 온습도센서 1개, 점퍼선, ESP32
2. 배선
 - 온습도센서 : 18번
3. 업로드
 - 코드 스켈레톤

```
#include <Arduino.h>
#include <DHT.h>

#define DHTPIN 18          // DHT11 센서 데이터 핀을 ESP32의 18번 핀에 연결
#define DHTTYPE DHT11      // 사용하는 센서 종류 지정

DHT dht(DHTPIN, DHTTYPE); // DHT 센서 라이브러리 객체 생성 및 초기화

void setup() {
  Serial.begin(115200);    // 시리얼 통신 속도를 115200bps로 설정
  dht.begin();             // DHT 센서 초기화
}

void loop() {
  float temperature = dht.readTemperature(); // 온도값 읽기 (섭씨)
  float humidity = dht.readHumidity();       // 습도값 읽기 (%)

  Serial.print(temperature); // 온도를 시리얼 출력
  Serial.print("C ");
  Serial.print(humidity);    // 습도를 시리얼 출력
  Serial.println("%");

  delay(2000);               // 2초 대기 후 반복
}
```

4. 관찰
 - DHT11 센서에서 2초마다 온도와 습도를 읽어 시리얼 모니터에 출력한다. -
 - 센서 초기화와 시리얼 통신 설정을 포함한다.

프로젝트 **B**: 온도 임계값 초과 시 **LED** 경고

1. 준비
 - 사용 부품: 온습도센서 1개, LED 1개, 점퍼선, ESP32
2. 배선
 - 온습도센서 : 18번 , LED : 4번
3. 업로드
 - 코드 스켈레톤

```
#include <Arduino.h>
#include <DHT.h>

#define DHTPIN 18
#define DHTTYPE DHT11
#define LEDPIN 4           // LED가 연결된 핀번호 지정

DHT dht(DHTPIN, DHTTYPE); // 센서 객체 생성

void setup() {
  pinMode(LEDPIN, OUTPUT); // LED핀을 출력모드로 설정
  Serial.begin(115200);    // 시리얼 통신 시작
  dht.begin();             // 센서 초기화
}

void loop() {
  float t = dht.readTemperature(); // 온도값 읽기

  if (t > 28) {              // 온도가 28도 초과 시
    digitalWrite(LEDPIN, HIGH); // LED 켜
  } else {
    digitalWrite(LEDPIN, LOW);  // LED 끄
  }

  delay(2000);              // 2초 대기 후 반복
}
```

4. 관찰
 - 측정한 온도가 28도 이상일 경우 LED를 켜서 경고를 표시하고, 이하일 때는 LED를 끈다.
 - 온도 판독과 LED 출력 제어를 포함.

프로젝트 **C**: 습도에 따라 **LED** 점멸 속도 제어

1. 준비
 - 사용 부품: 온습도센서 1개, LED 1개, 점퍼선, ESP32
2. 배선
 - 온습도센서 : 18번 , LED : 4번
3. 업로드
 - 코드 스켈레톤

```
#include <Arduino.h>
#include <DHT.h>

#define DHTPIN 18
#define DHTTYPE DHT11
#define LEDPIN 4

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  pinMode(LEDPIN, OUTPUT); // LED핀 출력모드 설정
  Serial.begin(115200);
  dht.begin();
}

void loop() {
  float humidity = dht.readHumidity(); // 현재 습도 측정

  // 습도를 20~80% 범위로 받고 점멸 간격을 1000~200ms 범위로 매핑
  int interval = map(humidity, 20, 80, 1000, 200);

  digitalWrite(LEDPIN, HIGH); // LED ON
  delay(interval);           // 점멸 간격만큼 대기
  digitalWrite(LEDPIN, LOW); // LED OFF
  delay(interval);          // 점멸 간격만큼 대기
}
```

4. 관찰
 - 습도에 따라 LED 깜박임 속도를 조절한다. 습도가 높으면 LED가 더 빠르게 깜빡인다. 점멸 주기를 동적으로 조절하여 실시간 피드백 제공.

프로젝트 D: 온습도 5회 측정 로그 출력

1. 준비
 - 사용 부품: 온습도센서 1개, 점퍼선, ESP32
2. 배선
 - 온습도센서 : 18번
3. 업로드
 - 코드 스켈레톤

```
#include <Arduino.h>
#include <DHT.h>

#define DHTPIN 18
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
float tempLog[5];           // 온도 측정값 배열 저장소
float humidLog[5];          // 습도 측정값 배열 저장소
int index = 0;              // 배열 인덱스 변수

void setup() {
  Serial.begin(115200);      // 시리얼 통신 시작
  dht.begin();              // 센서 초기화
}

void loop() {
  // 현재 온습도값을 배열에 저장
  tempLog[index] = dht.readTemperature();
  humidLog[index] = dht.readHumidity();

  index++;                  // 다음 인덱스로 이동

  if (index >= 5) {         // 5개 측정값이 다 쌓이면
    for (int i = 0; i < 5; i++) {
      Serial.print(tempLog[i]); Serial.print("C ");
      Serial.print(humidLog[i]); Serial.print("% | ");
    }
    Serial.println();       // 줄바꿈으로 출력 종료
    index = 0;              // 인덱스 초기화 후 다시 측정 시작
  }

  delay(1000);              // 1초 간격으로 반복
}
```

4. 관찰
 - 실시간으로 5번 연속 온도와 습도를 측정해 배열에 저장한 뒤, 한꺼번에 시리얼 모니터에 출력해 변화 추이를 확인한다.

프로젝트 E: 시리얼 명령으로 **LED** 제어 및 센서값 요청

1. 준비- 사용 부품: 온습도센서 1개, LED 1개, 점퍼선, ESP32
2. 배선- 온습도센서 : 18번, LED : 4번
3. 업로드- 코드 스켈레톤

```
#include <Arduino.h>
#include <DHT.h>

#define DHTPIN 18
#define DHTTYPE DHT11
#define LEDPIN 4

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  pinMode(LEDPIN, OUTPUT); // LED 핀 출력모드 설정
  Serial.begin(115200);     // 시리얼 통신 시작
  dht.begin();             // 센서 초기화
}

void loop() {
  if (Serial.available()) { // 시리얼 입력이 있으면
    String cmd = Serial.readStringUntil('\n'); // 한 줄 명령 읽기
    cmd.trim(); // 공백 제거

    if (cmd == "on") // "on" 입력 시 LED 켜기
      digitalWrite(LEDPIN, HIGH);
    else if (cmd == "off") // "off" 입력 시 LED 끄기
      digitalWrite(LEDPIN, LOW);
    else if (cmd == "data") { // "data" 입력 시 온습도 출력
      float temperature = dht.readTemperature();
      float humidity = dht.readHumidity();
      Serial.print(temperature);
      Serial.print("C ");
      Serial.print(humidity);
      Serial.println("%");
    }
  }

  delay(100); // 입력을 기다리면서 CPU 점유율 절약
}
```

4. 관찰
- 시리얼 통신으로 "on", "off", "data" 명령을 받아 LED를 켜거나 끄고, "data" 입력 시 측정된 온습도 값을 시리얼 모니터에 출력한다.