


Dokumentation Smarthome

Ziele des Projekts

- Der Server baut unabhängig vom lokalen WLAN-Netz ein eigenes Netz (Access-Point) auf
- die beiden ESPs messen jeweils die Temperatur
- ein Spannungsmesser misst die an einem Solarpanel anliegende Spannung
- ein Mikroschalter teilt mit, ob das Fenster offen oder geschlossen ist
- ein Regensensor beurteilt die Wetterlage (trocken oder Regen)
- ein Fotowiderstand schaltet bei Dämmerung/Dunkelheit eine LED ein
- der als Server agierende ESP sammelt die Daten (Spannung am Solarpanel, Zustand des Fensters, Temperaturen und Wetterlage) und stellt sie auf einer Webseite dar

 192.168.4.1

Messdaten

Wohnzimmer: 18,70°C
Esszimmer: 19,60°C
Energie am Solarpanel: 16,50 V
Fenster Wohnzimmer: offen
Wetterlage: Regen

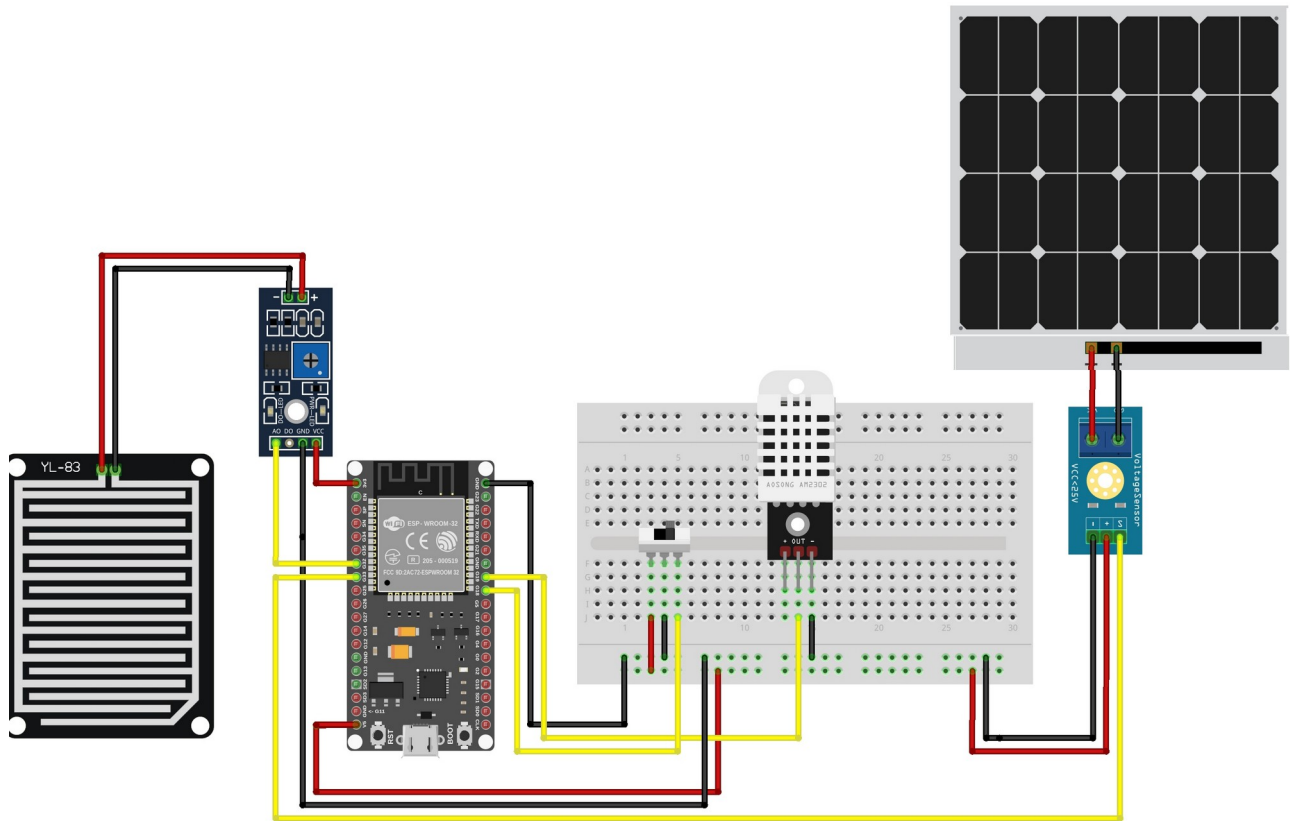
[aktualisieren](#)

Benötigte Bauteile

2 ESP32
2 DHT22
Solarpanel
Spannungsmesser
Mikroschalter
Fotowiderstand
LED
Widerstand 10 kΩ
Widerstand 220 Ω

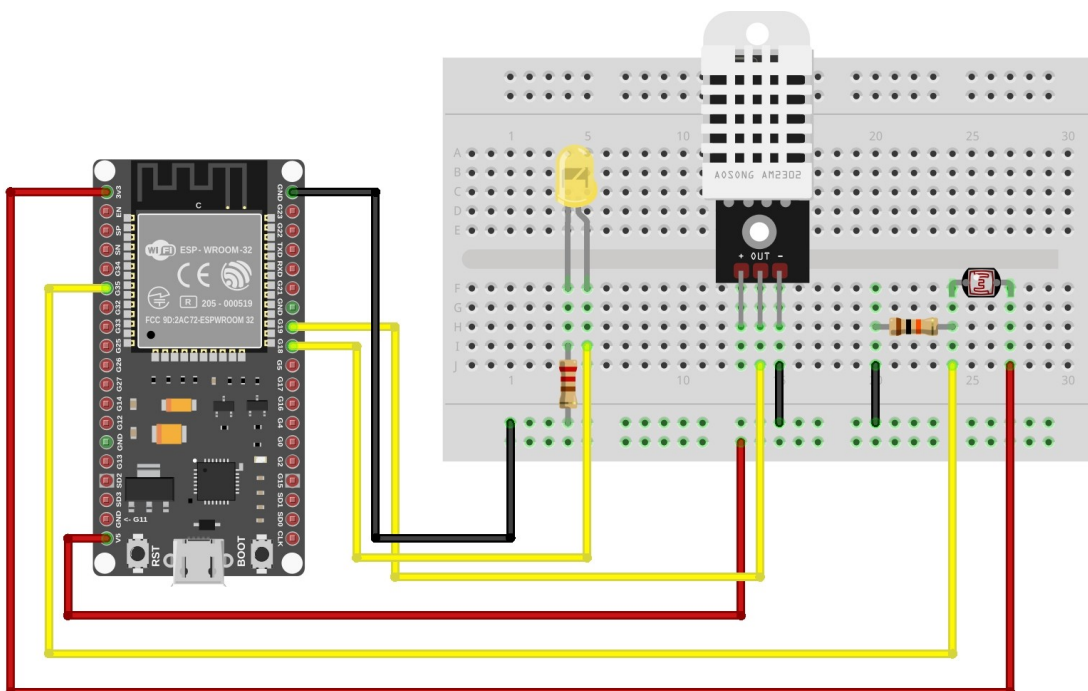
Schaltpläne

Server



fritzing

Klient



fritzing

Die Programme

Server

```
// benötigte Bibliotheken
#include "WebServer.h"
#include "WiFi.h"
#include "DHT.h"

// digitaler Pin des DHT-Sensors
#define SENSOR_DHT 19

// digitaler Pin Schalter
#define Schalter 18

// analoger Pin Spannungssensor
#define SpannungSensor 32

// analoger Pin Regensensor
#define Tropfensensor 33

// Sensortyp DHT festlegen
// DHT22 oder DHT11
#define SensorTyp DHT22

// Sensor DHT einen Namen zuweisen (dht)
DHT dht(SENSOR_DHT, SensorTyp);

// IP-Adressen des Klienten
const char* Klient = "192.168.4.2";

// String für die Messung
String Messung;

// Netzwerkname und Passwort des ESP
char Router[] = "ESPServer";
char Passwort[] = "espserver";

// der ESP agiert gleichzeitig als Server (Gateway) und Klient
IPAddress ip(192, 168, 4, 1);
IPAddress gateway(192, 168, 4, 1);
IPAddress subnet(255, 255, 255, 0);

// Webserver dem Port 80 zuordnen
WebServer Server(80);

// HTML-Grundstruktur
String Seitenkopf =
    "<head><style>"

    // farbiger Button mit CSS
    ".farbigeBox {"
    "background-color: ivory;"
    "color: black;"
    "width: 650px;"
    "padding: 20px;"
    "text-align: left;"
    "font-size: 40px;"
    "font-family: arial;"
    "}"
    "</style>"

    // refresh -> Seite automatisch aktualisieren
    "<meta http-equiv=\"refresh\" content=\"30\"></head>";
void setup()
```

```
{
  Serial.begin(9600);

  // ESP als Access-Point (AP) konfigurieren
  WiFi.softAPConfig(ip, gateway, subnet);

  // Access-Point starten
  WiFi.softAP(Router, Passwort);

  // / -> Aufruf der URL, SeiteBauen -> Aufruf der Funktion
  Server.on("/", SeiteBauen);
  Server.begin();

  // DHT starten
  dht.begin();

  // Schalter als Eingang
  pinMode(Schalter, INPUT);
}

void loop()
{
  // auf Anfragen warten
  Server.handleClient();

  // Messdaten vom Klienten holen
  DatenHolen();
}

void SeiteBauen()
{
  // Messwert DHT ermitteln
  // . durch , ersetzen
  String Temperatur = String(dht.readTemperature());
  Temperatur.replace(".", ",");

  // Spannung Solarpanel messen
  float geleseneSpannung = analogRead(SpannungSensor) * (3.3 / 4096.0);

  // in String umwandeln, . durch , ersetzen
  String EingangsSpannung = String(geleseneSpannung * 5);
  EingangsSpannung.replace(".", ",");

  // Zustand Schalter
  String ZustandSchalter;
  if (digitalRead(Schalter)) ZustandSchalter = "offen";
  else ZustandSchalter = "geschlossen";

  // Tropfsensor
  String Regenmenge;
  float MesswertTropfsensor = analogRead(Tropfsensor);

  // Maximalwert 4095
  if (MesswertTropfsensor > 4000) Regenmenge = "trocken";
  else Regenmenge = "Regen";
  Serial.println(MesswertTropfsensor);

  // Seite zusammen bauen
  String Seite = "";
  Seite += Seitenkopf;
  Seite += "<h1 align=\"left\">Messdaten</h1>";
  Seite += "<div align=\"left\">";

  Seite += "<div class=\"farbigeBox\">";
  Seite += "Wohnzimmer: ";
  Seite += Temperatur + "&deg;C";
  Seite += "<br>";
  Seite += "Esszimmer:";
```

```
Seite += String(Messung);

Seite += "<br>";
Seite += "Energie am Solarpanel: " + EingangsSpannung + " V";

// Fenster
Seite += "<br>";
Seite += "Fenster Wohnzimmer: " + ZustandSchalter;

// Tropfensensor
Seite += "<br>";
Seite += "Wetterlage: " + Regenmenge;
Seite += "</div>";

// Button aktualisieren
Seite += "<hr><input style=\"font-size:16pt; font-weight:bold;\"";
Seite += "background-color:#55A96B;\"";
Seite += "display:block; cursor:pointer;\"type=\"button\"\"";

// IP für den Button aktualisieren (location.href)
// muss mit dem Wert für IPAdress übereinstimmen (. statt ,)
Seite += " onClick=\"location.href='http://192.168.4.1'\" value=\"aktualisieren\">\"";
Seite += "<hr>";

// Seite übermitteln
Server.send(200, "text/html", Seite);
}

void DatenHolen()
{
    WiFiClient client;

    // wenn der Klient nicht verbunden ist -> zurück und neuer Versuch
    if (!client.connect(Klient, 80))
    {
        return;
    }

    /*
    GET-Anfrage senden
    /Messung -> Adresse (URL) für die zu übermittelnden Werte
    wird vom Klienten festgelegt
    Klient -> der zweite ESP
    HTTP/1.1 -> Abfrageprotokoll
    \r\n -> return mit anschließender neuer Zeile
    */
    client.print(String("GET ") + "/Messung" + " HTTP/1.1\r\n" + Klient + "\r\n");
    unsigned long LetzteZeit = millis();

    // Wartezeit bis zur Übermittlung der Daten
    while (!client.available() && ((millis() - LetzteZeit) < 3000))
    {
        delay(1);
    }

    // der Klient ist verfügbar
    while (client.available())
    {
        // den mit GET erhaltenen String bis zum return (\r) lesen
        Messung = client.readStringUntil('\r');
    }
}
```

Klient

```
// benötigte Bibliotheken
#include "WebServer.h"
#include "WiFi.h"
#include "DHT.h"

// digitaler Pin DHT-Sensor
int SENSOR_DHT = 19;

// Sensortyp festlegen
// DHT22 oder DHT11
#define SensorTyp DHT22

// Sensor DHT einen Namen zuweisen
DHT dht(SENSOR_DHT, SensorTyp);

// analoger Pin Fotowiderstand 35
#define Fotowiderstand 35

// digitaler Pin LED
#define LED 18

// ESP als AP
char Router[] = "ESPServer";
char Passwort[] = "espserver";

// IP des Klienten
IPAddress ip(192, 168, 4, 2);

// Gateway ist der der andere ESP
IPAddress gateway(192, 168, 4, 1);
IPAddress subnet(255, 255, 255, 0);

// Webserver dem Port 80 zuordnen
WebServer Server(80);

void setup()
{
    Serial.begin(9600);
    pinMode(LED, OUTPUT);

    // WiFi starten
    WiFi.config(ip, gateway, subnet);
    WiFi.begin(Router, Passwort);

    // Adresse (URL) festlegen (/Messung)
    // SeiteBauen -> zu übermittelnde Daten
    Server.on("/Messung", SeiteBauen);
    Server.begin();

    // DHT starten
    dht.begin();
}

void loop()
{
    // auf Anfragen warten
    Server.handleClient();
}

void SeiteBauen()
{
    // Temperatur messen
    // . durch , ersetzen
    String Temperatur = String(dht.readTemperature());
    Temperatur.replace(".", ",");
}
```

```
// Wert Fotowiderstand lesen
int SensorWert = analogRead(Fotowiderstand);

// Messwert des Fotowiderstands anzeigen
// Wert 1000 an Gegebenheiten anpassen
// Serial.println(SensorWert);
// delay(500);
if (SensorWert < 1000) digitalWrite(LED, HIGH);
else digitalWrite(LED, LOW);

// Seite zusammenbauen
String Seite = Temperatur + "&degC";

// Seite übermitteln
Server.send(200, "text/html", Seite);
}
```