**Code uitleg:**

**Esp32:**

#define BLYNK\_PRINT Serial

#define BLYNK\_TEMPLATE\_ID "\*\*\*\*\*\*"

#define BLYNK\_TEMPLATE\_NAME "\*\*\*\*\*\*\*\*"

Dit stuk code configureert de instellingen voor de Blynk-app.

De Blynk print serial: Dit vertelt Blynk om uitvoer te sturen naar de seriële monitor voor debugdoeleinden.

Blynk\_template\_ID: Hier geef je het ID van het Blynk-template op dat je in de Blynk-app hebt gemaakt.

Blynk\_Template\_Name: Dit is de naam van het Blynk-template dat je in de Blynk-app hebt gemaakt.

#include <WiFi.h>

#include <PubSubClient.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include "DHT.h"

#include <EEPROM.h>

We gaan verschillende bibliotheken toevoegen:

* #include <WiFi.h>`: Deze bibliotheek maakt het mogelijk om met WiFi-netwerken te communiceren op een ESP32.
* #include <PubSubClient.h>`: Deze bibliotheek maakt het mogelijk om MQTT-berichten te versturen en ontvangen.
* #include <WiFiClient.h>`: Dit is een aanvullende bibliotheek die wordt gebruikt in combinatie met de WiFi-bibliotheek.
* #include <BlynkSimpleEsp32.h>`: Deze bibliotheek maakt verbinding met de Blynk-server en maakt communicatie met de Blynk-app mogelijk.
* ”DHT.h"`: Dit is de bibliotheek voor het aansturen van DHT-sensoren, die worden gebruikt voor het meten van temperatuur en vochtigheid.
* <EEPROM.h>`: Deze bibliotheek biedt toegang tot de EEPROM van de microcontroller, waardoor gegevens permanent kunnen worden opgeslagen.

const char\* ssid = "\*\*\*\*\*\*\*\*";

const char\* password = "\*\*\*\*\*\*\*\*";

const char\* mqttServer = "192.\*\*\*.\*.\*\*\*";

const int mqttPort = 1883;

const char\* mqttUser = "\*\*\*\*\*";

const char\* mqttPassword = "\*\*\*\*\*";

const char\* clientID = "\*\*\*\*\*";  // MQTT client ID

We gaan hier de nodige gegevens invullen. Eerst van de wifi de naam en wachtwoord. Vervolgens de data van de mqtt broker.

char auth[] = "\*\*\*\*\*\*\*\*\*";

Dit is de authentificatie code van je BLYNK code. Je kan deze in je BLYNK programma kopiëren en hier plakken.

#define DHTPIN 12

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

We geven aan op welke pin de dht11 is aangesloten en welk type dht we gebruiken.

WiFiClient espClient;

PubSubClient client(espClient);

We maken een wifi client aan genaamd espClient en een pubsubclient genaamd client voor mqtt communicatie.

BlynkTimer timer;

Dit voegen we toe om periodieke taken te plannen en uitvoeren in een Blynk-programma.

void sendSensor();

We maken deze void op voorhand aan.

void setup() {

  Serial.begin(115200);

  WiFi.begin(ssid, password);

  dht.begin();

We initaliseren de verbinding met een WIFI netwerk en de dht11 sensor

  pinMode(26, OUTPUT);  // ROOD = Verwarming

  pinMode(27, OUTPUT);  // BLAUW = Airco

  pinMode(25, OUTPUT);  // ORANJE = connectie

We stellen onze outputs vast.

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.println("Connecting to WiFi..");

  }

  Serial.println("Connected to the WiFi network");

  client.setServer(mqttServer, mqttPort);

  client.subscribe("raspi/#");  
We verbinden het apparaat met een WiFi-netwerk en configureren het om te luisteren naar MQTT-berichten van het "raspi/#" topic.

  Blynk.begin(auth, ssid, password, "server.wyns.it", 8081);

  timer.setInterval(1000L, sendSensor);

We starten BLYNK met de gegeven gegevens.

We stellen een interval in van 1sec bij de functie sendSensor.

}

void loop() {

  Blynk.run();

  timer.run();

We starten BLYNK en de timer.

  client.loop();

We maken een client loop aan voor de mqtt

  float t = dht.readTemperature();

We maken de variabele t die de data van dht.readTemperature opslaat.

  while (!client.connected()) {

    Serial.println("Connecting to MQTT...");

    digitalWrite(25, HIGH);

    delay(500);

    digitalWrite(25, LOW);

    if (client.connect("ESP32Client", mqttUser, mqttPassword)) {

      Serial.println("connected");

      digitalWrite(25, HIGH);

    }

  }

We gaan connectie maken met de MQTT. Wanneer hij zoekt zal de led op pin 25 flikkeren en wanneer hij verbonden is zal de led blijven branden.

  String payload = String(t);

  client.publish("esp32/kamer1/temp", payload.c\_str());

We publiceren met topic “esp32/kamer1/temp” de temperatuur van de dht11 we maken hiervoor eerst een string aan anders kunnen we de data niet publishen.

  if (digitalRead(27) == HIGH) {

    client.publish("esp32/kamer1/set", "airco1-AAN");

    Serial.println("A1");

  }

Als de led op pin 27 brandt zal er een bericht verstuurd worden met de topic “esp32/kamer/set” en met bericht airco1-AAN terwijl word er in de seriele monitor A1 geprint als controle.

  if (digitalRead(26) == HIGH) {

    client.publish("esp32/kamer1/set", "verwarming1-AAN");

    Serial.println("V1");

  }

Als de led op pin 27 brandt zal er een bericht verstuurd worden met de topic “esp32/kamer/set” en met bericht airco1-AAN terwijl word er in de seriele monitor A1 geprint als controle.

  delay(250);

  client.disconnect();  // disconnect from the MQTT broker

}

Na een vertraging van 250 miliseconden gaan we de verbinding terug verbreken met de mqtt

Door de loop blijft dit zich herhalen

void sendSensor() {

Deze void word aangemaakt voor de BLYNK app.

  float t = dht.readTemperature();

  Blynk.virtualWrite(V0, t);

}

We maken terug de variabele t aan met de data van de dht11 en sturen dit naar de virtuele pin0 van het BLYNK programma.

BLYNK\_WRITE(V2) {

We gaan de viruele pin2 vaststellen.

  float getal = param.asInt();

We maken de variabele getal aan om de waarde van de schuiver op de BLYNK app op te slaan.

  if (dht.readTemperature() < getal - 0.5) {

    digitalWrite(26, HIGH);

  } else if (dht.readTemperature() > getal + 0.5) {

    digitalWrite(26, LOW);

  }

Wanneer de waarde van getal groter is dan de gemeten temperatuur met een 0.5°C speling zal de led op pin26 branden als we er onder zitten zal de led uit gaan.

  if (dht.readTemperature() > getal - 0.5) {

    digitalWrite(27, HIGH);

  } else if (dht.readTemperature() < getal + 0.5) {

    digitalWrite(27, LOW);

  }

Wanneer de waarde van getal groter is dan de gemeten temperatuur met een 0.5°C speling zal de led op pin27 branden als we er onder zitten zal de led uit gaan.

}