

Ministerul Educației și Cercetării
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Ingineria Software și Automatică

Raport

Curs: Internetul lucrurilor

Tema: Comunicare cu periferii - MQTT

A elaborat:

st.gr.SI-211 Chirița Stanislav

A verificat:

Asist. Univ.Astafi Valentina

Chișinău 2024

1 Definirea problemei

Sa se realizeze o aplicatie ce va implementa comunicatiile intre echipamente dupa cum urmeaza:

1. Realizarea unei aplicatii de comunicare Internet prin protocolul MQTT pentru interactiunea cu o resursa Cloud
2. Colectarea datelor de la sensori si trimitere catre un broker MQTT
3. urmarirea mesajelor de la un broker MQTT si setarea starii unui actuator la alegere
4. Datele sunt vizualizate si controlate de la un dashboard Internet (*ThingsBoard* sau *HiveMQ*)

2. Descrierea funcțiilor

setup()

Această funcție este apelată o singură dată la repornirea sau alimentarea microcontrollerului ESP32 și are rolul de a inițializa conexiunea WiFi și clientul MQTT:

- `Serial.begin(9600);` – Inițializează comunicația serială cu o viteză de transfer de 9600 bps pentru afișarea datelor și a mesajelor de diagnostic pe consola serială.
- `dht.begin();` – Inițializează senzorul DHT22 pentru a permite citirea datelor de temperatură și umiditate.
- `WiFi.begin(ssid, password);` – Conectează ESP32 la rețeaua WiFi specificată.
- **Bucă de așteptare pentru conectarea la WiFi** – Verifică starea conexiunii WiFi și continuă execuția doar după stabilirea acesteia.
- `client.setServer(mqttServer, mqttPort);` – Configurează adresa serverului MQTT și portul de conectare.
- **Bucă de așteptare pentru conectarea la MQTT** – Încearcă să conecteze clientul la brokerul MQTT și continuă execuția după realizarea conexiunii cu succes.

loop()

Funcția `loop()` este apelată în mod repetat și conține logica principală a programului:

- `float humidity = dht.readHumidity();` – Citește valoarea curentă a umidității de la senzorul DHT22.
- `float temperature = dht.readTemperature();` – Citește valoarea curentă a temperaturii de la senzorul DHT22.
- **Verificare conexiune MQTT** – `if (client.connected())` – Asigură că clientul este conectat la brokerul MQTT înainte de a publica datele.
- `client.publish("v1/devices/me/telemetry", ...)` – Trimite datele de temperatură și umiditate în format JSON către topicul specificat pe brokerul MQTT.
- `client.loop();` – Menține activă conexiunea cu brokerul MQTT și gestionează eventualele mesaje primite sau trimise.
- `delay(2000);` – Introduce o pauză de 2 secunde pentru a limita frecvența de citire și trimitere a datelor.

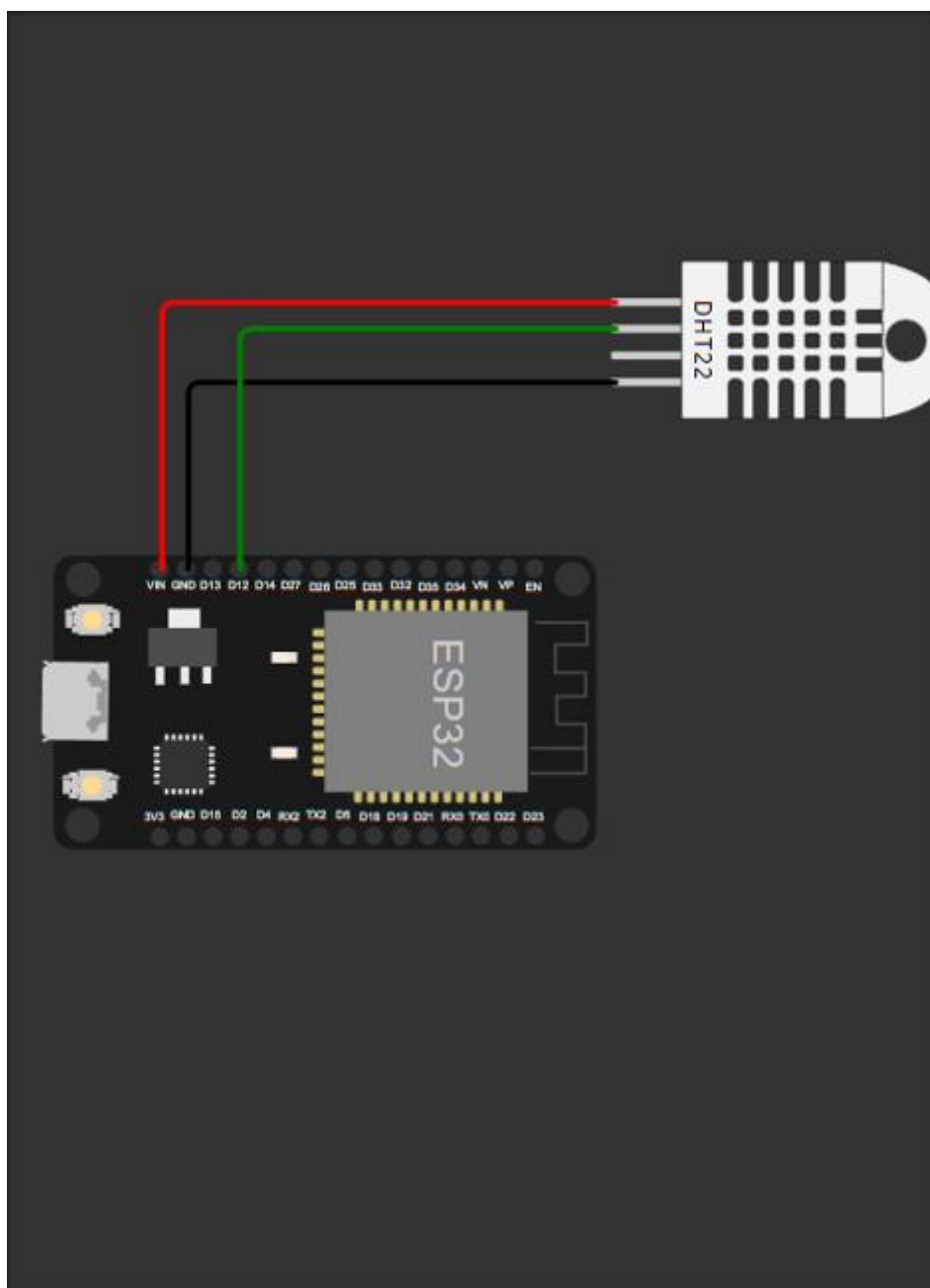


Figura 1 – Circuitul

Concluzie

Proiectul demonstrează implementarea cu succes a unui sistem IoT bazat pe protocolul MQTT pentru colectarea și transmiterea datelor de la senzori către un broker MQTT, precum și pentru gestionarea unui actuator prin mesaje de la broker. Utilizarea ESP32 ca dispozitiv central permite conectivitatea WiFi eficientă