

# Raport PRIoT

## Sistem pentru Monitorizarea Temperaturii și Umidității cu Notificări și Control la Distanță

Dumitrescu Rareș-Matei

December 10, 2024

### Contents

<b>1</b>	<b>Introducere</b>	<b>1</b>
1.1	Descrierea proiectului . . . . .	1
1.2	Obiective . . . . .	1
<b>2</b>	<b>Arhitectură</b>	<b>2</b>
2.1	Schema topologiei rețelei . . . . .	2
2.2	Dashboard pentru User . . . . .	2
2.3	Protocolul de comunicare . . . . .	3
<b>3</b>	<b>Cod sursă</b>	<b>3</b>
3.1	ESP32 - Configurare senzori și transmitere date . . . . .	3
<b>4</b>	<b>Vizualizare date</b>	<b>3</b>
4.1	Node-RED . . . . .	3
<b>5</b>	<b>Concluzie</b>	<b>4</b>

## 1 Introducere

### 1.1 Descrierea proiectului

Acest proiect IoT permite monitorizarea temperaturii și umidității în timp real folosind senzori DHT11/DHT22 conectați la un microcontroller ESP32. Scopul proiectului este să ofere date despre mediu printr-un dashboard web, alerte în timp real pentru depășirea limitelor și posibilitatea de a controla un actuator de la distanță.

### 1.2 Obiective

- Monitorizarea temperaturii și umidității în timp real.
- Vizualizarea datelor pe un dashboard web.
- Notificarea utilizatorului în caz de depășire a limitelor.

- Controlul unui ventilator/LED prin aplicație web.

## 2 Arhitectură

### 2.1 Schema topologiei rețelei

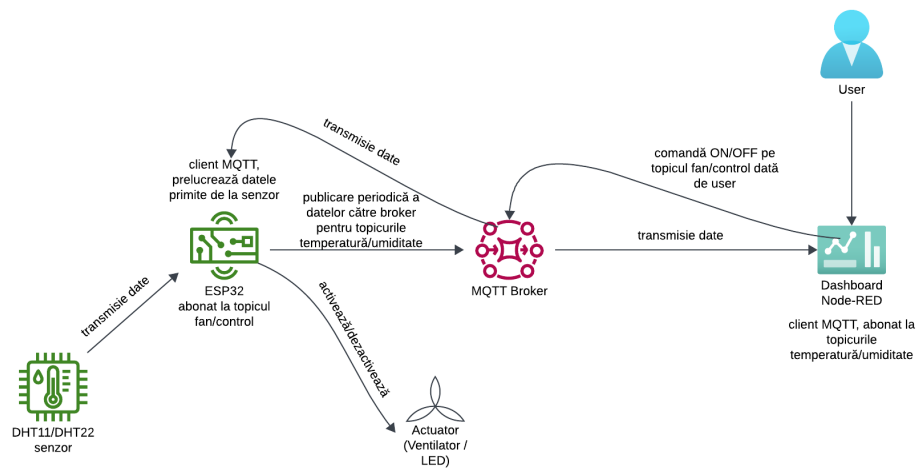


Figure 1: Schema topologiei rețelei

### 2.2 Dashboard pentru User

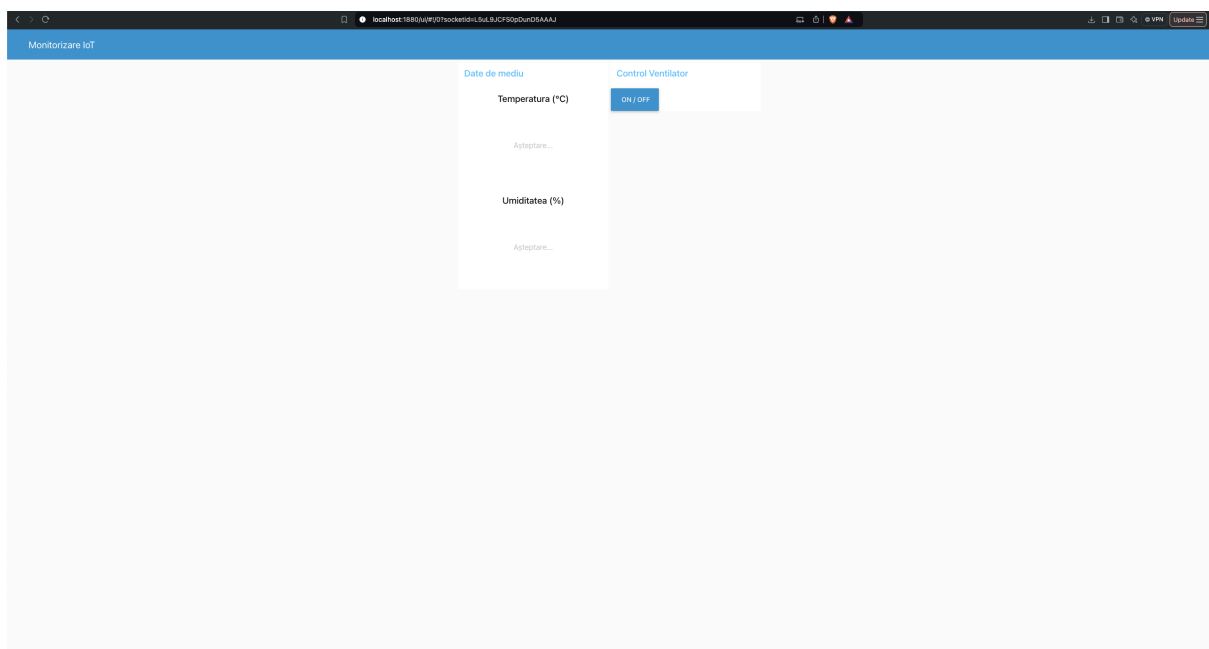


Figure 2: Dashboard pentru User

Sistemul constă dintr-un senzor DHT11/DHT22 conectat la un ESP32 care transmite date către un broker MQTT. Datele sunt preluate și afișate pe un dashboard web realizat cu Node-RED.

## 2.3 Protocolul de comunicare

Protocolul MQTT a fost ales pentru transmiterea eficientă a datelor, datorită consumului redus de resurse și scalabilității.

## 3 Cod sursă

### 3.1 ESP32 - Configurare senzori și transmitere date

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <DHT.h>
4
5 // Configura ii senzor
6 #define DHTPIN 5
7 #define DHTTYPE DHT11
8 DHT dht(DHTPIN, DHTTYPE);
9
10 // Configura ii WiFi i MQTT
11 const char* ssid = "Numele_WiFi";
12 const char* password = "Parola_WiFi";
13 const char* mqtt_server = "broker.hivemq.com";
14 WiFiClient espClient;
15 PubSubClient client(espClient);
16
17 void setup() {
18   Serial.begin(115200);
19   dht.begin();
20   WiFi.begin(ssid, password);
21   client.setServer(mqtt_server, 1883);
22 }
23
24 void loop() {
25   if (!client.connected()) reconnect();
26   client.loop();
27   float temp = dht.readTemperature();
28   float hum = dht.readHumidity();
29   if (!isnan(temp) && !isnan(hum)) {
30     client.publish("temperature", String(temp).c_str());
31     client.publish("humidity", String(hum).c_str());
32   }
33   delay(2000);
34 }
```

Listing 1: Cod ESP32 pentru monitorizare temperatură și umiditate

## 4 Vizualizare date

### 4.1 Node-RED

Dashboard-ul este configurat pentru a afișa temperatura și umiditatea în timp real și pentru a trimite notificări atunci când valorile depășesc limitele predefinite.

- **\*\*Noduri folosite:\*\***

- MQTT In - pentru datele de la senzori.
  - Chart - pentru graficele temperaturii și umidității.
  - Notification - pentru alerte.
- 

## 5 Concluzie

Această etapă a proiectului a stabilit baza arhitecturală și a realizat prototipul pentru transmiterea și vizualizarea datelor.