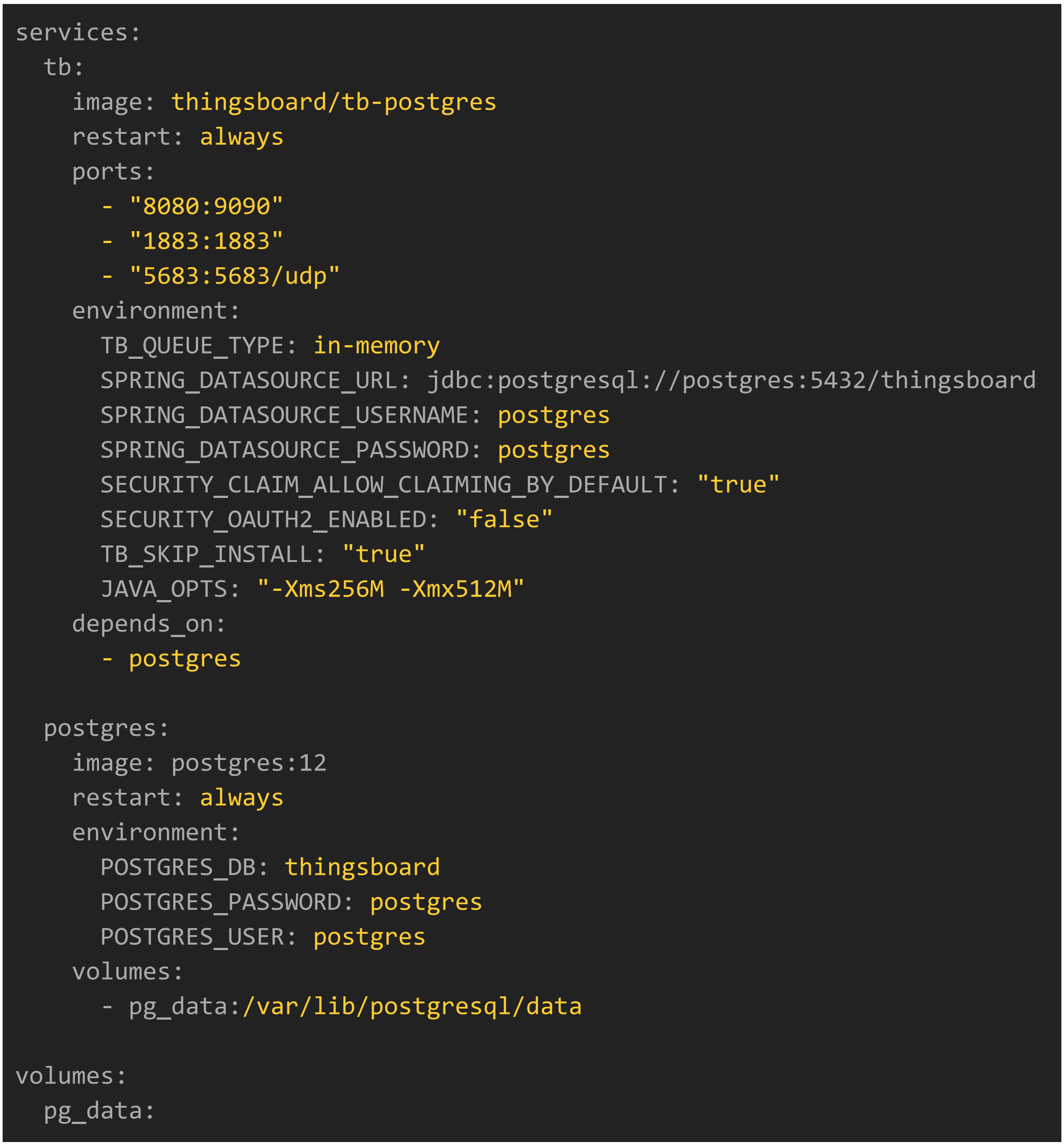
سوال دوم)

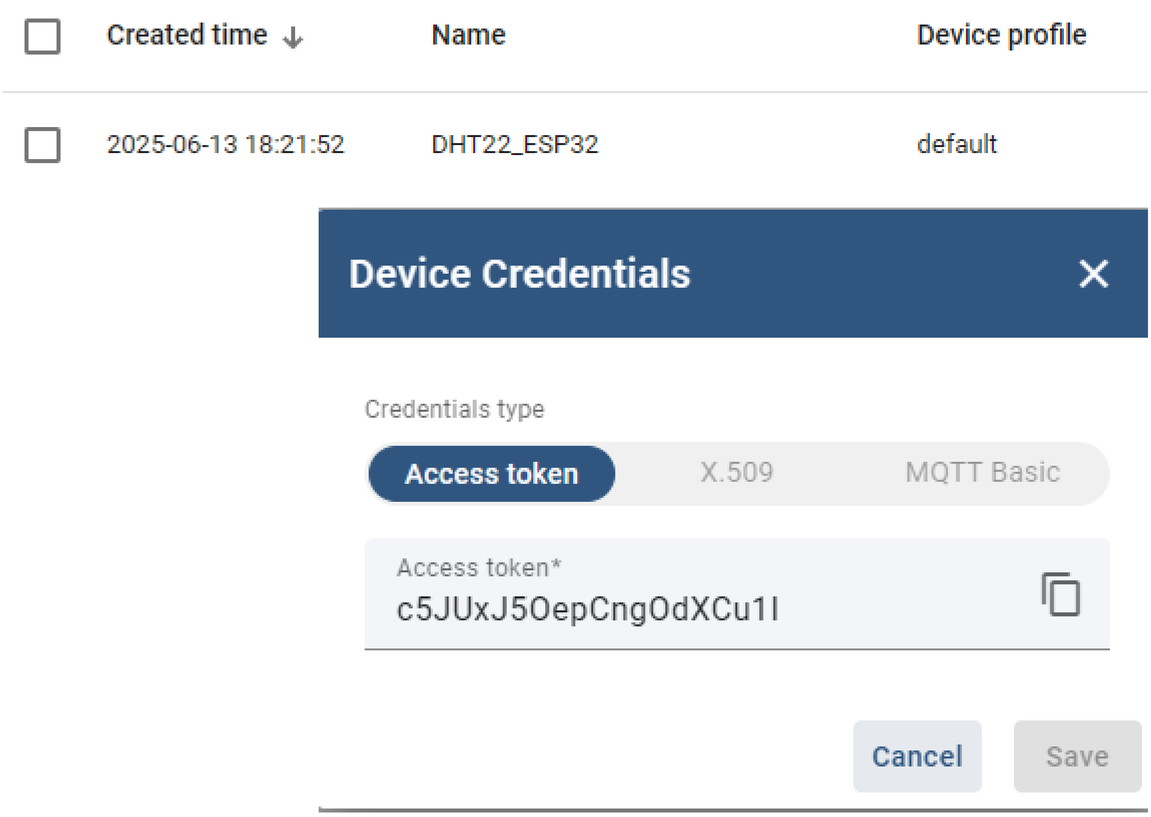
الف)

با استفاده از این configuration توسط docker پلتفرم thingsboard را بالا آوردیم:



```yaml
services:
  tb:
    image: thingsboard/tb-postgres
    restart: always
    ports:
      - "8080:9090"
      - "1883:1883"
      - "5683:5683/udp"
    environment:
      TB_QUEUE_TYPE: in-memory
      SPRING_DATASOURCE_URL: jdbc:postgresql://postgres:5432/thingsboard
      SPRING_DATASOURCE_USERNAME: postgres
      SPRING_DATASOURCE_PASSWORD: postgres
      SECURITY_CLAIM_ALLOW_CLAIMING_BY_DEFAULT: "true"
      SECURITY_OAUTH2_ENABLED: "false"
      TB_SKIP_INSTALL: "true"
      JAVA_OPTS: "-Xms256M -Xmx512M"
    depends_on:
      - postgres

  postgres:
    image: postgres:12
    restart: always
    environment:
      POSTGRES_DB: thingsboard
      POSTGRES_PASSWORD: postgres
      POSTGRES_USER: postgres
    volumes:
      - pg_data:/var/lib/postgresql/data

volumes:
  pg_data:
```

ب)



□	Created time ↓	Name	Device profile
□	2025-06-13 18:21:52	DHT22_ESP32	default

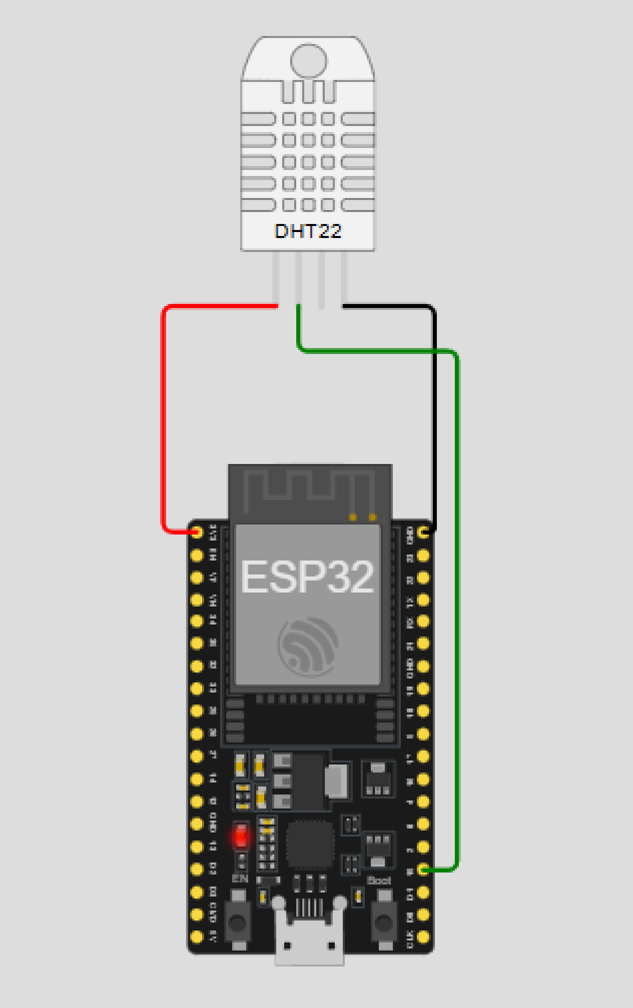
Device Credentials

Credentials type

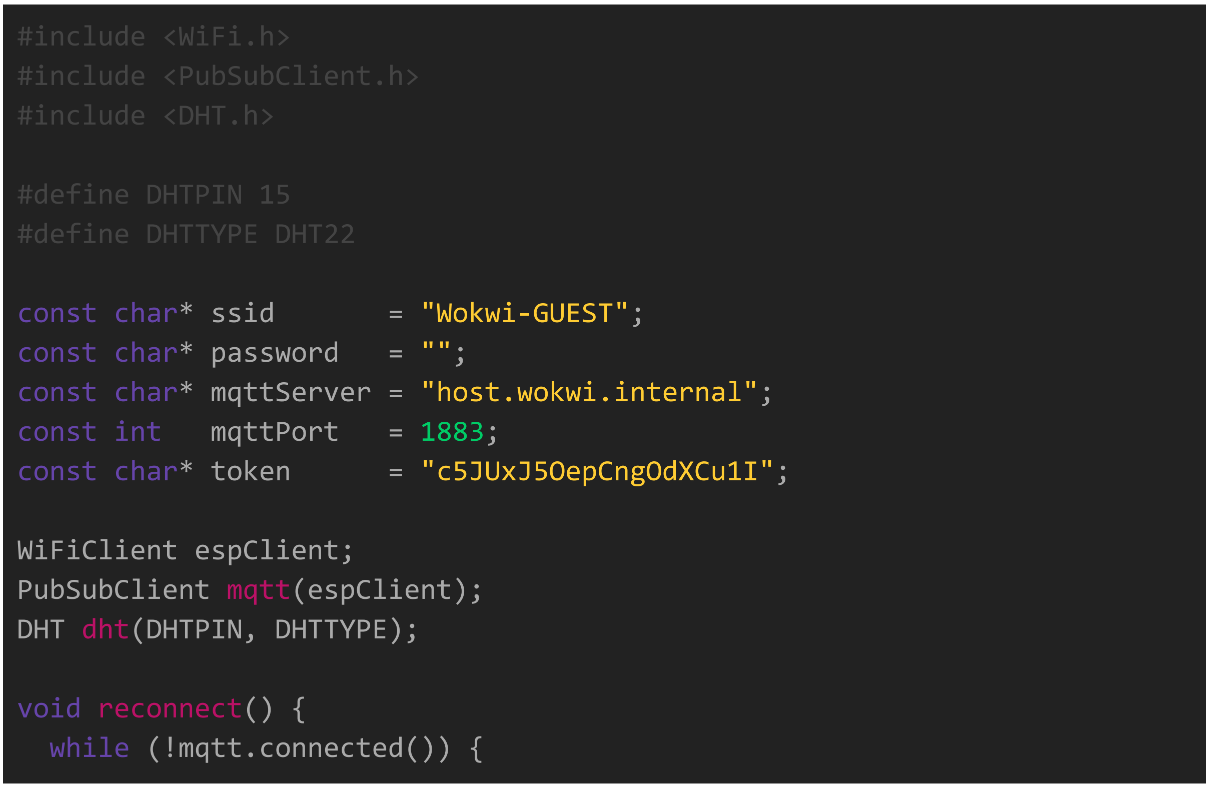
Access token*
c5JUxJ5OepCngOdXCu1I

ج)

مدار با استفاده از ESP32 و DHT22 در wokwi ساخته شد:

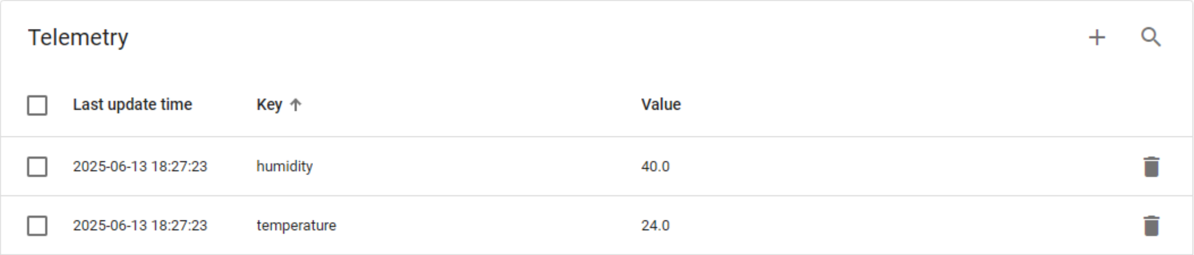


کد مربوط به مدار و فرستادن داده به thingsboard:



```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>

#define DHTPIN 15
#define DHTTYPE DHT22

const char* ssid       = "Wokwi-GUEST";
const char* password   = "";
const char* mqttServer = "host.wokwi.internal";
const int   mqttPort   = 1883;
const char* token      = "c5JUxJ5OepCngOdXCu1I";

WiFiClient espClient;
PubSubClient mqtt(espClient);
DHT dht(DHTPIN, DHTTYPE);

void reconnect() {
  while (!mqtt.connected()) {
    Serial.print("Connecting to MQTT...");
    if (mqtt.connect("ESP32Client", token, nullptr)) {
      Serial.println("connected!");
    } else {
      Serial.print("failed, rc=");
      Serial.print(mqtt.state());
      Serial.println(" try again in 3 seconds");
      delay(3000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  dht.begin();

  WiFi.begin(ssid, password, 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected");

  mqtt.setServer(mqttServer, mqttPort);
}

void loop() {
  if (!mqtt.connected()) reconnect();
  mqtt.loop();

  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (!isnan(temperature) && !isnan(humidity)) {
    String payload = "{\"temperature\":" + String(temperature) +
                     ",\"humidity\":" + String(humidity) + "}";
    mqtt.publish("v1/devices/me/telemetry", payload.c_str());
    Serial.println("Published: " + payload);
  } else {
    Serial.println("Failed to read from DHT");
  }

  delay(5000);
}
```

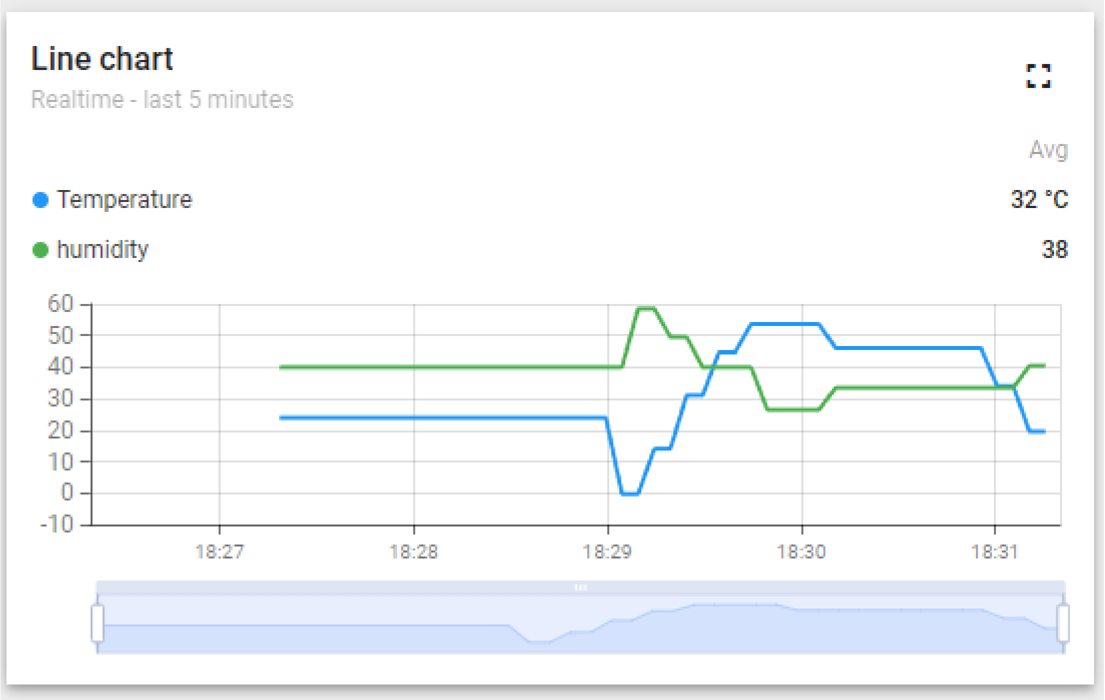
د)



Telemetry

□	Last update time	Key ↑	Value
□	2025-06-13 18:27:23	humidity	40.0
□	2025-06-13 18:27:23	temperature	24.0

ه)



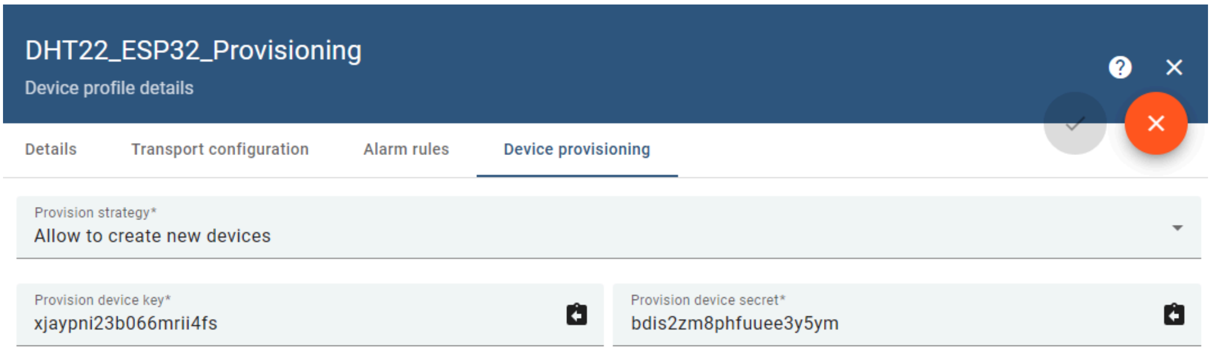
Line chart

Realtime - last 5 minutes

	Avg
● Temperature	32 °C
● humidity	38

سوال سوم)

الف)



DHT22_ESP32_Provisioning

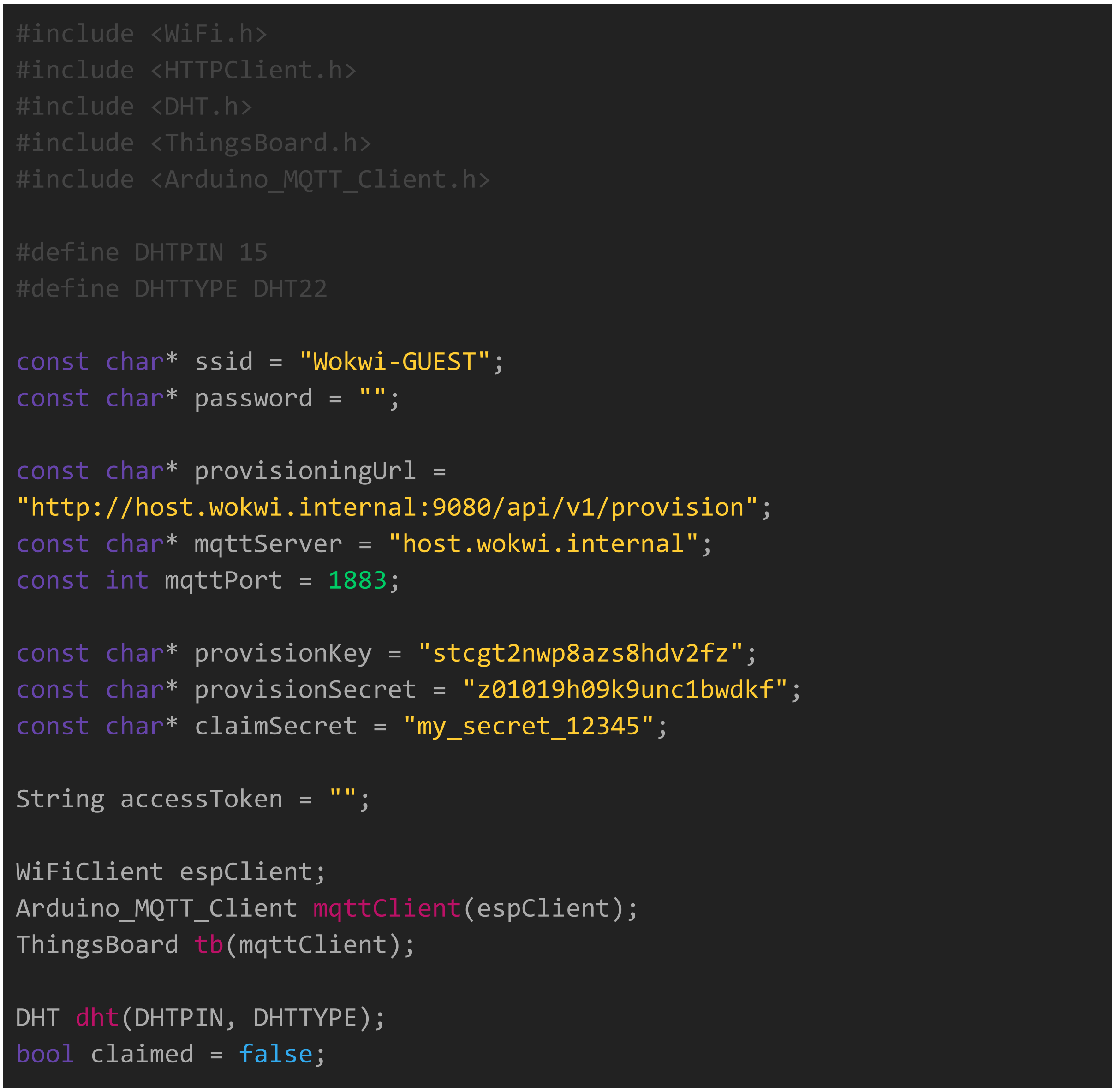
Device profile details

Provision strategy*
Allow to create new devices

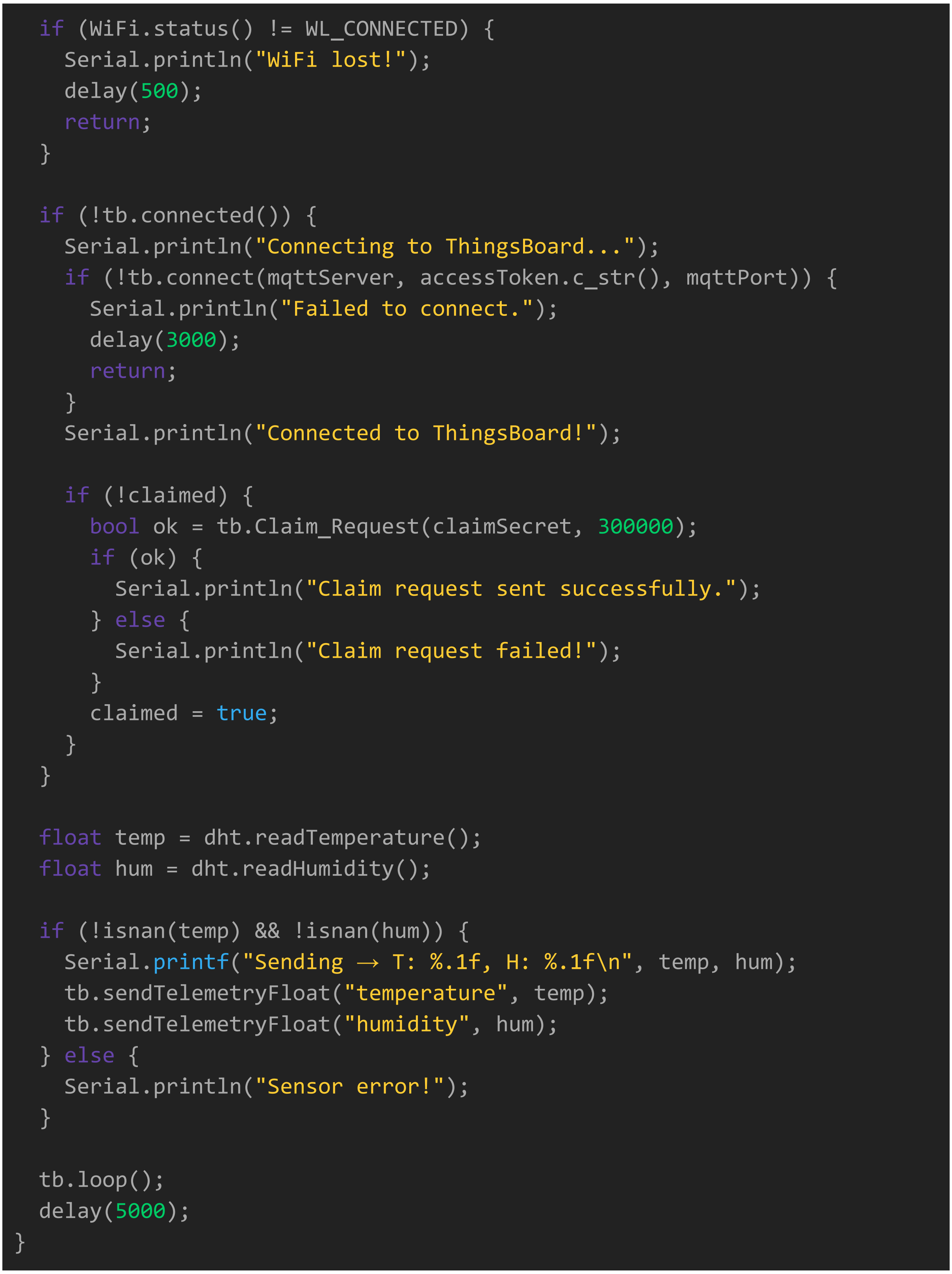
Provision device key*
xjaypni23b066mrii4fs

Provision device secret*
bdis2zm8phfuuee3y5ym

ب)

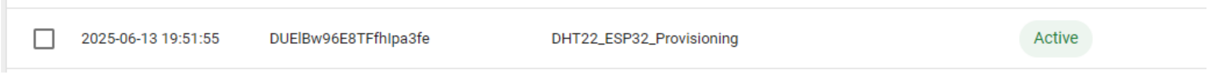


```
#include <WiFi.h>
#include <HTTPClient.h>
#include <DHT.h>
#include <ThingsBoard.h>
#include <Arduino_MQTT_Client.h>

#define DHTPIN 15
#define DHTTYPE DHT22

const char* ssid = "Wokwi-GUEST";
const char* password = "";

const char* provisioningUrl =
"http://host.wokwi.internal:9080/api/v1/provision";
const char* mqttServer = "host.wokwi.internal";
const int mqttPort = 1883;

const char* provisionKey = "stcgt2nwp8azs8hdv2fz";
const char* provisionSecret = "z01019h09k9unc1bwdkf";
const char* claimSecret = "my_secret_12345";

String accessToken = "";

WiFiClient espClient;
Arduino_MQTT_Client mqttClient(espClient);
ThingsBoard tb(mqttClient);

DHT dht(DHTPIN, DHTTYPE);
bool claimed = false;
bool provisionDevice() {
  HTTPClient http;
  http.begin(provisioningUrl);
  http.addHeader("Content-Type", "application/json");

  String body = "{\"provisionDeviceKey\":\"" + String(provisionKey) +
                "\",\"provisionDeviceSecret\":\"" +
String(provisionSecret) + "\"}";

  int httpCode = http.POST(body);
  if (httpCode == 200) {
    String response = http.getString();
    int i1 = response.indexOf("\"credentialsValue\":\"") + 20;
    int i2 = response.indexOf("\"", i1);
    accessToken = response.substring(i1, i2);
    Serial.println("Access token: " + accessToken);
    http.end();
    return true;
  } else {
    Serial.println("Provision failed: " + http.getString());
    http.end();
    return false;
  }
}

void setup() {
  Serial.begin(115200);
  dht.begin();

  WiFi.begin(ssid, password, 6);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(200);
  }
  Serial.println("\nWiFi connected!");

  if (!provisionDevice()) {
    Serial.println("Provisioning failed!");
    while (true);
  }

  mqttClient.setServer(mqttServer, mqttPort);
}

void loop() {
```



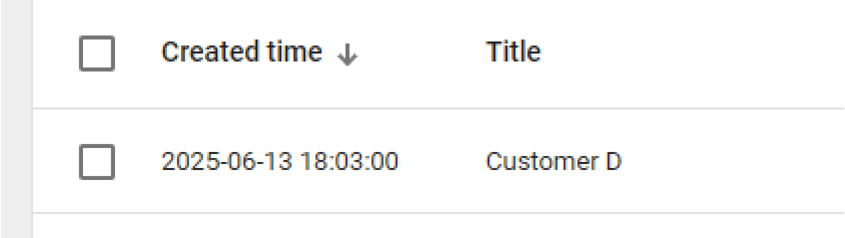
```
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi lost!");
    delay(500);
    return;
  }

  if (!tb.connected()) {
    Serial.println("Connecting to ThingsBoard...");
    if (!tb.connect(mqttServer, accessToken.c_str(), mqttPort)) {
      Serial.println("Failed to connect.");
      delay(3000);
      return;
    }
    Serial.println("Connected to ThingsBoard!");

    if (!claimed) {
      bool ok = tb.Claim_Request(claimSecret, 300000);
      if (ok) {
        Serial.println("Claim request sent successfully.");
      } else {
        Serial.println("Claim request failed!");
      }
      claimed = true;
    }
  }

  float temp = dht.readTemperature();
  float hum = dht.readHumidity();

  if (!isnan(temp) && !isnan(hum)) {
    Serial.printf("Sending → T: %.1f, H: %.1f\n", temp, hum);
    tb.sendTelemetryFloat("temperature", temp);
    tb.sendTelemetryFloat("humidity", hum);
  } else {
    Serial.println("Sensor error!");
  }

  tb.loop();
  delay(5000);
}
```

ج)



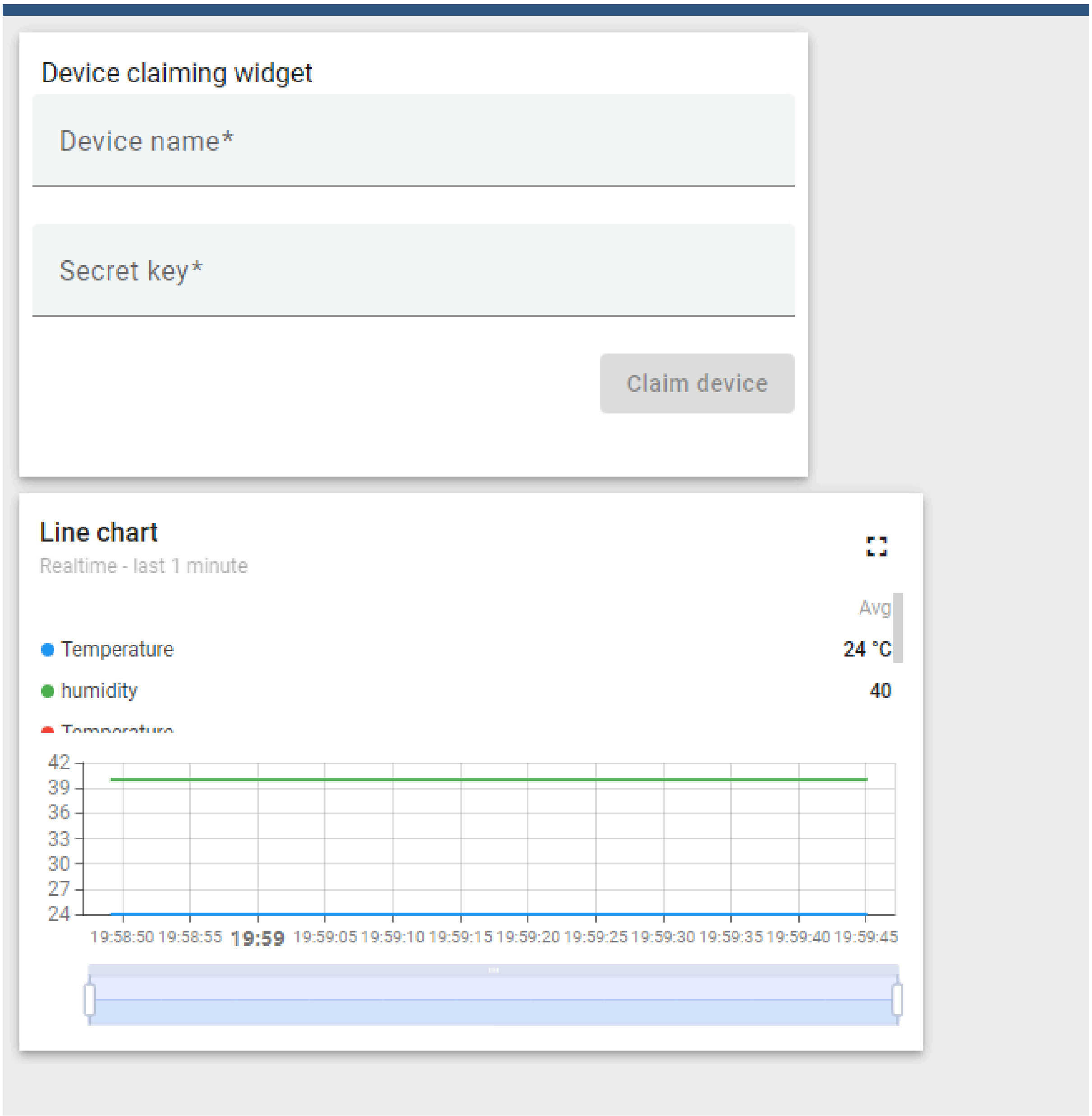
□	2025-06-13 19:51:55	DUElBw96E8TFfhIpa3fe	DHT22_ESP32_Provisioning	Active

د)



□	Created time ↓	Title
□	2025-06-13 18:03:00	Customer D

ه)


Device claiming widget
Device name*
Secret key*
Claim device
Line chart
Realtime - last 1 minute
Avg
Temperature
24 °C
humidity
40
Temperature
42
39
36
33
30
27
24
19:58:50
19:58:55
19:59
19:59:05
19:59:10
19:59:15
19:59:20
19:59:25
19:59:30
19:59:35
19:59:40
19:59:45