

سناریوی HR202 و

IoT Jump

برای پایش رطوبت محیط



www.sasadra.com



1. هدف

هدف از اجرای این سناریو، این است که بتوان مقدار رطوبت محیط را در اپلیکیشن IoT Jump مشاهده و پایش کرد.



2. معرفی HR202

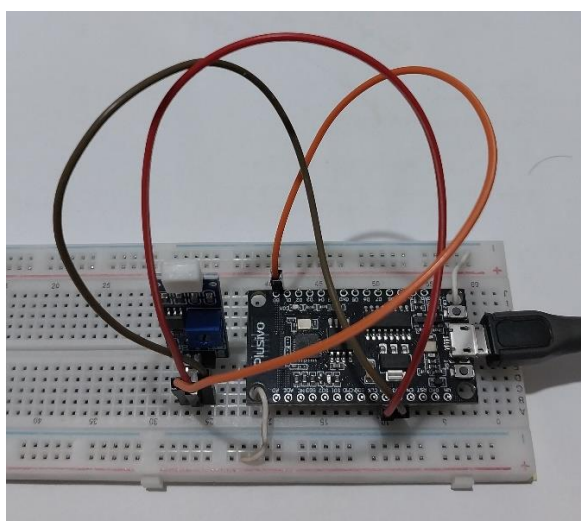
ماژول حسگر یا سنسور رطوبت HR202 برای اندازه‌گیری میزان رطوبت محیط استفاده می‌شود. این ماژول در مکان‌هایی مانند بیمارستان، انبار، کارگاه، و صنایع نساجی، تنباکو، دارویی، هواشناسی و غیره کاربرد دارد. از ویژگی‌های این ماژول می‌توان به مواردی مانند مصرف برق کم، محدوده اندازه‌گیری گسترده، پاسخ سریع، ضد آلودگی، ثبات زیاد و نسبت کارایی به قیمت بالا اشاره کرد.



3. دستورالعمل

ابتدا می‌باید ماژول HR202 به ESP8266 متصل گردد برای این کار:

- پایه VCC ماژول HR202 را به پایه 3 یا 5 ولت ESP8266
- پایه GND ماژول HR202 را به پایه GND یا زمین ESP8266
- و پایه D ماژول HR202 را به یکی از پایه‌های D اتصال دهید.



سپس می‌باید ESP8266 را با کابل به رایانه وصل کرده و در نرم‌افزار Arduino برنامه زیر را اجرا کنید. دقت کنید که ssid ، password ، mqttServer و mqttPort باید با مقادیر درست جایگزین شوند. این برنامه، مقدار رطوبت خوانده شده توسط HR202 را دریافت کرده، به کارگزار MQTT متصل می‌شود و mqttTopic تعریف شده را با مقدار عددی رطوبت منتشر می‌کند.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// WiFi credentials
const char* ssid = "123";
const char* password = "123";
```



```
// MQTT Broker
const char* mqttServer = "abc.def.com";
const int mqttPort = 1883;
const char* humidity_topic = "IoTJump_humidity"; // MQTT topic to publish
humidity data

const int HR202_PIN = A0; // GPIO pin connected to HR202 sensor OUT
const float HR202_VCC = 3.3; // Voltage supplied to the HR202 sensor
const float HR202_RESOLUTION = 1023.0; // ADC resolution (for analogRead)

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  delay(10);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }

  Serial.println("Connected to WiFi");

  // Set MQTT server and callback function
  client.setServer(mqttServer, mqttPort);
  client.setCallback(callback);
}

void callback(char* topic, byte* payload, unsigned int length) {
  // Handle incoming MQTT messages (if needed)
}

void reconnect() {
  // Loop until we're reconnected to the MQTT server
  while (!client.connected()) {
    Serial.println("Attempting MQTT connection...");

    // Attempt to connect
    if (client.connect("ESP8266Client")) {
```



```
        Serial.println("Connected to MQTT server");
    } else {
        Serial.print("Failed, rc=");
        Serial.print(client.state());
        Serial.println(" Retrying in 5 seconds...");
        delay(5000);
    }
}
}

float readHR202Humidity() {
    // Read humidity data from HR202
    int rawReading = analogRead(HR202_PIN);
    float voltage = (float)rawReading * (HR202_VCC / HR202_RESOLUTION);

    // Add your calibration formula here to convert voltage to humidity
    // Example:
    float humidity = x + y;

    return humidity;
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }

    float humidity = readHR202Humidity();

    char payload[16]; // Adjust buffer size as needed
    sprintf(payload, sizeof(payload), "%.2f", humidity);
    client.publish(humidity_topic, payload);

    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.println("");

    delay(10000); // Wait for 10 seconds before the next reading
}
```



← ویرایش پنل

نام پنل *
میزان رطوبت

غیر فعال کردن پیشوند داشبورد

موضوع *
IoTJump_humidity

حداکثر Payload	حداقل Payload
۱۰۰	۰
فاکتور	واحد
۱	%

نوع نمایش پیشرفت
گرد ▼

رنگ
▼

رنگ پویا

فعال سازی اعلان (notification)

payload ، داده‌ای از نوع JSON است

نمایش زمان دریافت

QoS
۰ ▼

ذخیره لغو

سپس در اپ IoT Jump باید یک اتصال به کارگزار MQTT ساخته شده و در آن اتصال، یک پنل پیشرفت (Progress) با مشخصات زیر برای دریافت و مشاهده مقدار رطوبت ساخته شود.

نحوه ساخت اتصال و داشبورد در سناریوی قبلی (ESP8266) توضیح داده شده است.

دقت کنید که می‌توان از پنل‌های دیگری مانند انواع نمودار، اندازه‌گیر (Gauge)، لغزنده و ... نیز برای مشاهده مقدار رطوبت استفاده کرد. نکته مهم دیگر اینکه مشخصات کارگزار و topic یا موضوع MQTT می‌باید هم در اپ و هم در برنامه آردوینو یکسان باشند.



4. تماس با ما

فیلم آموزش این سناریو در صفحات اجتماعی اپ موجود است. اگر سوال خاصی دارید و یا قابلیت جدیدی می‌خواهید، با ما تماس بگیرید.



<https://instagram.com/IoTJump>



<https://twitter.com/IoTJump>



<https://facebook.com/IoTJump1>

ایمیل ما sa.sadra@yahoo.com 