



IoT Project

(IOT)

“Microcontroller-based Wearable
Blood Pressure Monitoring Device
with GPS and SMS Feature through
Mobile App.”

Prepared by

Hojjat Yazdan - 99242155

Amirhossein Dibaj - 99242059

Artin Elhamirad - 99242017

Group 1

IoT Project

❖ چگونگی کار با thingspeak یکی از پلتفرم های معروف IOT

Thingspeak چیست؟

ThingSpeak یک پلتفرم تحلیل داده های IoT است که امکان جمعیت، تجسم و تحلیل جریان های داده زنده در فضای ابری را فراهم می کند. این پلتفرم به دستگاه ها اجازه می دهد که داده های خود را به ThingSpeak ارسال کنند و با استفاده از ابزارهای تجسم فوری، داده ها را به صورت آنلاین تحلیل و پردازش کنند.

ThingSpeak اغلب برای نمونه سازی و تست مفاهیم سیستم های IoT که نیاز به تحلیل دارند، مورد استفاده قرار می گیرد در واقع در یک سرور مرکزی منبع باز کار می کند. این سرور می تواند به صورت محلی نصب شود یا در فضای ابری اجرا شود. سرور ابزارهای مختلفی برای تجسم، پردازش و تحلیل داده ها ارائه می دهد و دارای کتابخانه های نرم افزاری است که امکان تعامل با دستگاه های IoT را فراهم می کند.

ویژگی اصلی ThingSpeak پشتیبانی از محیط توسعه MATLAB در فضای ابری است که شامل خروجی گرافیکی و چندین جعبه ابزار MATLAB می شود و دسترسی به دستورات MATLAB برای پردازش و تجسم داده ها را فراهم می کند. این پشتیبانی رایگان MATLAB تنها در پلتفرم ابری عمومی موجود است و در نصب های محلی موجود نیست و دسترسی به جعبه ابزارها نیاز به مجوز دارد.

۱) چگونگی و چارچوب ارسال داده توسط ThingSpeak

ThingSpeak از پروتکل های ارتباطی HTTP و MQTT برای ارتباط با تمامی پلتفرم های اصلی سخت افزار IoT مانند Arduino, Raspberry Pi, Electric Imp, Particle Photon و ESP8266 و همچنین سیستم عامل های موبایل و دسکتاپ پشتیبانی می کند. ThingSpeak برای تبادل و ذخیره سازی داده ها از دو مفهوم اصلی کانال ها و پیام ها استفاده می کند. دستگاه ها و برنامه ها داده ها را از طریق پیام هایی که با سرور مرکزی تبادل می شود به کانال ها می خوانند و می نویسند.

۲) چگونگی کار با ThingSpeak و تحلیل داده های ثبت شده و کشیدن نمودار

در نسخه ابری ThingSpeak، قابلیت های غنی پردازش و تجسم داده ها از طریق برنامه های ThingSpeak ارائه می شود. این برنامه ها می توانند یک بار، به صورت دوره ای یا به عنوان واکنش به پیام جدید اجرا شوند و امکان پردازش داده ها، تجسم آن ها با تولید نمودارها، یا توزیع آن ها در سایر کانال ها یا در اینترنت با انتشار آن ها در صفحات وب و شبکه های اجتماعی را فراهم می کنند.

IoT Project

پشتیبانی داخلی از اسکریپت‌ها و توابع MATLAB در برنامه‌ها در فضای ابری به دلیل استفاده گسترده از MATLAB در محیط‌های دانشگاهی بسیار جذاب است، که منجر به انتقال آسان تر الگوریتم‌های جدید پردازش و تحلیل داده‌ها از محیط‌های شبیه‌سازی به بسترهای آزمایشی می‌شود.

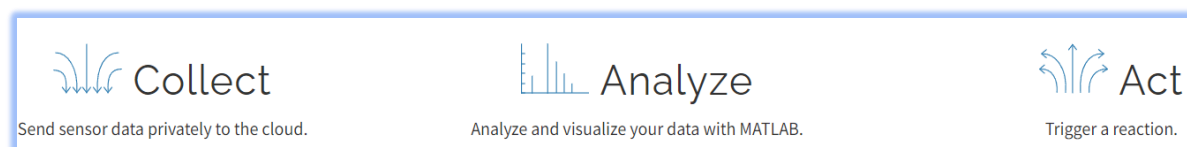
با این حال، این ویژگی‌ها در نصب‌های سرور محلی موجود نیستند و تنها قابلیت‌های پایه تجسم داده‌ها از طریق نمودارهای سری زمانی و گوگل گیج‌ها در رابط وب پلتفرم و پردازش داده‌ها با استفاده از افزونه‌های مبتنی بر جاوا اسکریپت ارائه می‌شوند. ادغام با خدمات خارجی با صادر کردن داده‌ها در قالب‌های JSON، CSV و XML یا با استفاده از پروتکل ThingHTTP امکان‌پذیر است.

امنیت در ThingSpeak با پشتیبانی از رمزگذاری انتها به انتها با TLS تضمین می‌شود. ThingSpeak می‌تواند به صورت محلی با کلون کردن مخزن GitHub موجود نصب شود و مستندات غنی در وبسایت پلتفرم ارائه شده است، اگرچه بیشتر بر نسخه ابری پلتفرم تمرکز دارد.

در جدول زیر به شمای کلی از این پلتفرم نمایش داده شده است:

Platform	Comm. Protocols	Data Processing	Data Visualization	Integration	Security	Installation Procedure	Documentation
ThingSpeak	HTTP, MQTT	HTML JavaScript plugins in a local installation, Matlab Analysis Apps in the cloud version	time series visualization and Google Gauges in a local installation, Matlab Visualization Apps in the cloud version	JSON, XML, CSV, REST API, ThingHTTP	TLS	installation from sources	Github repository and platform website, mostly focused on cloud version

حال به چگونگی جمع‌آوری، تجزیه و تحلیل و عمل بر روی داده‌های IoT با استفاده از ThingSpeak می‌پردازیم که شامل ۳ بخش زیر است :



سنسورها معمولاً داده‌ها را به صورت محلی حس کرده و پردازش می‌کنند. ThingSpeak به سنسورها، ابزارها و وبسایت‌ها این امکان را می‌دهد که داده‌ها را به فضای ابری ارسال کنند، جایی که این داده‌ها در کانال‌های خصوصی یا عمومی ذخیره می‌شوند. به طور پیش‌فرض، ThingSpeak داده‌ها را در کانال‌های خصوصی ذخیره می‌کند، اما کانال‌های عمومی نیز برای اشتراک داده‌ها با دیگران قابل استفاده هستند. وقتی داده‌ها در یک کانال ThingSpeak ذخیره شدند، می‌توان آن‌ها را تحلیل و تجسم کرد، داده‌های جدید محاسبه کرد، یا با رسانه‌های اجتماعی، خدمات وب و دستگاه‌های دیگر تعامل داشت.

IoT Project

حال به چگونگی ساخت یک کانال جدید میپردازیم :

۱. ورود به حساب کاربری

۲. ایجاد کانال

که در نهایت تب زیر نمایش داده میشود :

My Channels

New Channel

Search by tag

Q

Name	Created	Updated
<div><div><div><div><div><div></div></div><div>Dew Point Measurement</div></div></div><div><div>Private</div><div>Public</div><div>Settings</div><div>Sharing</div><div>API Keys</div><div>Data Import / Export</div></div></div></div>	2024-06-24	2024-06-24 14:04

تحلیل و تجسم داده‌ها با MATLAB

ذخیره‌سازی داده‌ها در فضای ابری دسترسی آسان به داده‌ها را فراهم می‌کند. با استفاده از ابزارهای تحلیلی آنلاین، می‌توانید داده‌ها را کاوش و تجسم کنید، روابط، الگوها و روندها را در داده‌ها کشف کنید، داده‌های جدید را محاسبه کنید و آن‌ها را در نمودارها، چارت‌ها و گیج‌ها تجسم کنید.

حال به بخش آنالیز میپردازیم:

- **تبدیل، ترکیب و محاسبه داده‌های جدید:** با استفاده از MATLAB، داده‌ها را به اشکال مختلف تبدیل کنید، داده‌های مختلف را ترکیب کنید و داده‌های جدیدی محاسبه کنید.
- **زمان‌بندی محاسبات برای اجرا در زمان‌های مشخص:** محاسبات را به گونه‌ای زمان‌بندی کنید که در زمان‌های خاصی اجرا شوند.
- **درک بصری روابط در داده‌ها با استفاده از توابع تجسم داخلی:** با استفاده از توابع تجسم داخلی MATLAB، روابط موجود در داده‌ها را به صورت بصری درک کنید.
- **ترکیب داده‌ها از چندین کانال برای ساخت تحلیل‌های پیچیده‌تر:** داده‌ها را از چندین کانال ترکیب کنید تا تحلیل‌های پیچیده‌تری بسازید.

مزایای استفاده از MATLAB در ThingSpeak

- **دسترسی به ابزارهای تحلیلی قوی MATLAB:** یکی از ابزارهای قوی تحلیلی است که قابلیت‌های گسترده‌ای برای تجزیه و تحلیل داده‌ها فراهم می‌کند.

IoT Project

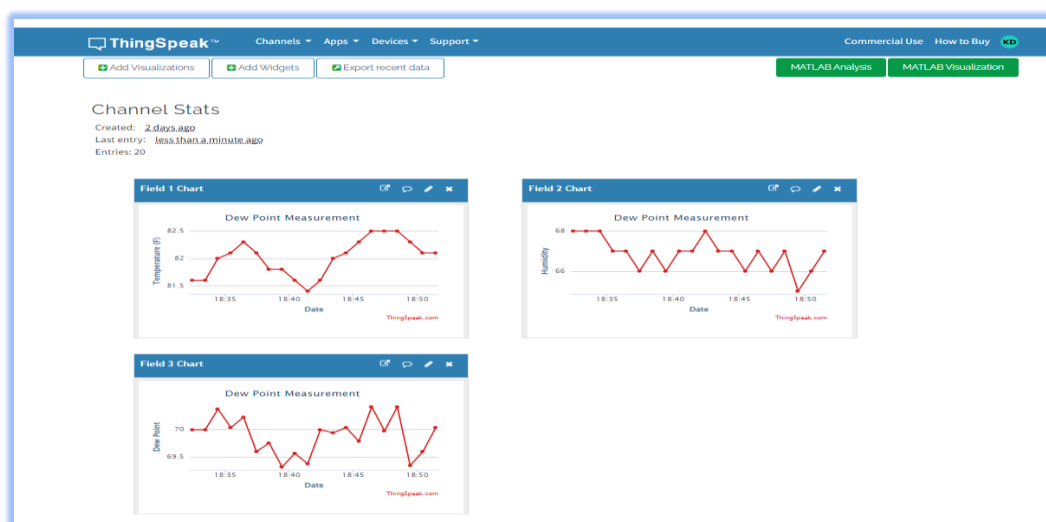
- تجسم داده‌ها به صورت حرفه‌ای: با استفاده از توابع تجسم MATLAB، می‌توانید نمودارها و گراف‌های حرفه‌ای ایجاد کنید که به درک بهتر داده‌ها کمک می‌کند.

- ادغام آسان با ThingSpeak: ThingSpeak امکان ادغام مستقیم با MATLAB را فراهم می‌کند که فرآیند تحلیل و تجسم داده‌ها را ساده‌تر می‌کند.

با این ابزارها، می‌توانید داده‌های IoT خود را به صورت کارآمد تحلیل و تجسم کنید و از بینش‌های حاصل از آن‌ها برای بهبود عملکرد سیستم‌های خود استفاده کنید.

در این مثال به چگونه می‌توانید داده‌ها را از یک کانال ThingSpeak بخوانید، آن‌ها را تحلیل کنید و نتایج را به کانال دیگری بنویسید. در این مثال خاص، داده‌های رطوبت و دما از کانال عمومی WeatherStation خوانده می‌شوند و به کانال Dew Point Measurement نوشته می‌شوند.

نمودار برای آنالیز داده‌ها:



و در مرحله آخر به واکنش دادن می‌پردازیم:

واکنش به داده‌ها در ThingSpeak

عمل کردن بر روی داده‌ها می‌تواند به سادگی دریافت یک توییت باشد وقتی که دمایی که اندازه‌گیری می‌کنید بالاتر از ۷۰ درجه فارنهایت می‌رود. یا می‌توانید یک عمل پیچیده‌تر مانند روشن کردن یک موتور وقتی سطح آب در مخزن شما زیر یک حد مشخص می‌افتد تنظیم کنید. حتی می‌توانید دستگاه‌هایی مانند قفل‌های درب با باتری را با استفاده از برنامه TalkBack از راه دور کنترل کنید. یا می‌توانید حساب تویتر خود را به ThingSpeak متصل کنید و یک واکنش ایجاد کنید تا هر بار که سطح نقطه شبنم بالای ۶۰ درجه فارنهایت می‌رود، یک توییت ارسال شود.

IoT Project

خلاصه پروژه :

در این پروژه سیستم پردازشی اصلی یک server web متصل به سنسور SRF04 و نیز actuator هایی که هم چند عدد LED و هم Servo موتور می باشند، در این پروژه client انتخاب شده موبایل می باشد که به واسطه ی یک مودم Wi Fi در همان شبکه ی محلی به آن وصل می شود؛ با اتصال client موبایل به این سیستم پردازشی، با داده هایی که از SRF04 میگیریم LED ها روشن و خاموش می کنیم و دیتا های سنسور را نمایش می دهیم همچنین با انتخاب گزینه های مناسب فرمان لازم را برای تغییر زاویه Servo موتور به سیستم پردازشی ارسال می کنیم.

۱. سیستم پردازشی انتخاب شده مبتنی بر ESP32 می باشد.
۲. سنسور ای که اطلاعات محیطی را به ما می دهد سنسور فاصله سنج SRF04 می باشد.
۳. actuator های که بر روی محیط عملی را اجرا می کنند، سه عدد LED و نیز Servo موتور می باشند.
۴. پروتکل لایه اپلیکیشن استفاده شده در طراحی ما HTTP می باشد و IDE مورد استفاده ما نیز Arduino است.

کد سیستم پردازشی اصلی :

```
#include <WiFi.h>
#include <WebServer.h>
#include <ESP32Servo.h> // Include the ESP32Servo library

// Network settings
const char *ssid = "IoT-Group1-AP";
const char *password = "12345678";

// IP settings
IPAddress local_IP(192, 168, 1, 1);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

// Define pins for SRF04 sensor, LEDs, and servo motor
const int trigPin = 5;
const int echoPin = 18;
const int led1 = 19;
const int led2 = 21;
const int led3 = 22;
const int servoPin = 23;

// Create a WebServer object on port 80
WebServer server(80);
```

IoT Project

```
// Create a Servo object
Servo myServo;
int servoAngle = 0; // Initial servo angle

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

    // Set sensor pins as input and output
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    // Set LED pins as output
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);

    // Attach the servo motor to the pin
    myServo.attach(servoPin);
    myServo.write(servoAngle);

    // Start AP mode with IP settings
    if (!WiFi.softAPConfig(local_IP, gateway, subnet)) {
        Serial.println("AP Config Failed");
    }
    if (!WiFi.softAP(ssid, password)) {
        Serial.println("AP Failed");
        return;
    }
    Serial.println("Access Point started");
    Serial.print("IP address: ");
    Serial.println(WiFi.softAPIP());

    // Set up the routes and handlers for the web server
    server.on("/", handleRoot);
    server.on("/setServo", handleSetServo);
    server.begin();
    Serial.println("Web server started");
}

void loop() {
    server.handleClient();
    long distance = readSRF04();

    // Control LEDs based on distance
    digitalWrite(led1, distance < 10 ? HIGH : LOW);
    digitalWrite(led2, distance < 20 ? HIGH : LOW);
    digitalWrite(led3, distance < 30 ? HIGH : LOW);
}
```

IoT Project

```
}

// Function to read data from SRF04 sensor
long readSRF04() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    long distance = duration * 0.034 / 2;

    return distance;
}

// Handler for the root route
void handleRoot() {
    long distance = readSRF04();

    // Control LEDs based on distance
    digitalWrite(led1, distance < 10 ? HIGH : LOW);
    digitalWrite(led2, distance < 20 ? HIGH : LOW);
    digitalWrite(led3, distance < 30 ? HIGH : LOW);

    String message = "<html><head><title>SRF04 Sensor Data</title></head>";
    message += "<style>";
    message += ".circle {width: 50px; height: 50px; border-radius: 50%; display: inline-block; margin: 10px;}";
    message += ".on {background-color: green;}";
    message += ".off {background-color: black;}";
    message += "</style></head><body>";
    message += "<h1>SRF04 Sensor Data</h1>";
    message += "<h2>Hojjat Yazdan 99242155 , Amirhossein Dibaj 99242059 , Artin Elhamirad 99242017</h2>";
    message += "<p>Distance: " + String(distance) + " cm</p>";
    message += "<div class='circle ' + String(digitalRead(led1) == HIGH ? \"on\" : \"off\") + \"'></div>";
    message += "<div class='circle ' + String(digitalRead(led2) == HIGH ? \"on\" : \"off\") + \"'></div>";
    message += "<div class='circle ' + String(digitalRead(led3) == HIGH ? \"on\" : \"off\") + \"'></div>";
    message += "<h2>Servo Control</h2>";
    message += "<input type='range' min='0' max='180' value='" + String(servoAngle) + "' id='servoSlider' onchange='updateServo(this.value)'>";
    message += "<p>Angle: <span id='servoAngle'>" + String(servoAngle) + "</span> degrees</p>";
    message += "<script>";
```


IoT Project

```
message += "function updateServo(val) {"";
message += "  var xhr = new XMLHttpRequest();"";
message += "  xhr.open('GET', '/setServo?angle=' + val, true);"";
message += "  xhr.send();"";
message += "  document.getElementById('servoAngle').innerText = val;"";
message += "}";
message += "</script>";
message += "</body></html>";
server.send(200, "text/html", message);
}

// Handler to set the servo angle
void handleSetServo() {
  if (server.hasArg("angle")) {
    servoAngle = server.arg("angle").toInt();
    myServo.write(servoAngle);
  }
  server.send(200, "text/plain", "OK");
}
```

توضیحات کد مربوطه :

کتابخانه های مربوط به وای فای و وب سرور را برای ایجاد وب سرور به صورت AP را اضافه می کنیم، از طرفی از کتابخانه مرتبط به ESP32 برای سروو موتور اضافه می کنیم، سپس تنظیمات شبکه و نیز IP را انجام می دهیم ، پس از آن پین هایی که برای LED ها و نیز SRF04 و نیز Servo Motor می باشند را اضافه می کنیم و سپس پورت ۸۰ برای HTTP وب سرور انتخاب می کنیم.

در بخش ستاپ مقدار باد ریت ارسال داده های سریال را انتخاب می کنیم، تنظیمات پین ها برای LED ها و نیز SRF04 و نیز Servo Motor را انجام داده و دستور شرطی برای وضعیت AP قرار می دهیم و در غیر اینصورت اگر با موفقیت وب سرور تولید گردد، چند پیام برای اتصال موفق نشان می دهیم و سپس مسیرها و کنترل کننده ها (routes and handlers) را برای وب سرور تنظیم می کنیم.

در تابع حلقه کنترل سمت کلاینت را هر بار بروز رسانی میکنیم تا مشتری با رفرش کردن صفحه مقادیر جدید را بدست آورد. و پس از آن برای روشن کردن LED ها بدون رفرش دستورات آن را در حلقه نوشته ایم.

یک تابع برای خواندن دیتای SRF04 نیز داریم که با استفاده از ارسال پالس توسط trigpin مقادیر را از echo می گیرد و میخواند .

IoT Project

یک تابع دیگر برای هندلر است که مقادیر را در صفحه وب نشان دهد و به وب پیغام ارسال کند و از آن دریافت کند. با دستورات Html و CSS و Java Script صفحه وب را به صورت دلخواه ساخته و مقادیر را نشان می دهیم و در بخش هایی حالت های LED ها و مقدار فاصله را نشان می دهد و در بخش دیگر مقدار زاویه را برای تنظیم سروو می گیرد و در تابع آخر که برای سوو موتور است ارسال میکند و آن تابع طبق آن درجه می چرخد.

نتیجه پیاده سازی

SRF04 Sensor Data

Distance: 6 cm



Servo Control



Angle: 72 degrees