

Projekt Arduino-Weather Dokumentation

AUTOREN

Diego Fantino Im20b
Filip Slavkovic Im20b

DATUM

25. Januar 2023

LEHRPERSON

Kevin Maurizi



Inhaltsverzeichnis

API	3
Vorgehensweise	4
Weitere Ideen.....	6

API

Diese API ermöglicht dem Nutzer in Echtzeit Wetterdaten von einem Server zu holen. Die API ist gratis, hat aber dafür gewisse Beschränkungen. Man braucht sich lediglich zu registrieren und hat dann schon seinen API-Key, den man in seinem Code verwenden kann. Die Daten der API werden immer als JSON mitgegeben, somit ist es die einzige Herausforderung, die Daten des JSON zu parsen. Dies hängt von der Programmiersprache und IDE ab. Den API-Key kann man für seine Wünsche anpassen. Man braucht lediglich die Stadt, das Land und die gewünschte Einheit und schon kann man sich die Daten herunterziehen. Man kann sich unter anderem die aktuelle Temperatur, die Luftfeuchtigkeit, den Luftdruck, die Wetterbeschreibung und das Bild dazu, sowie die Luftgeschwindigkeit herunterziehen.

Vorgehensweise

In diesem Projekt mussten wir uns zuerst mit der OpenWeatherMap-API befassen. Wir haben herausgefunden, dass hier alles schon vorgegeben ist. Die Daten der API werden als JSON gegeben, somit mussten wir eine ArduinoJSON-Library in unserem Code importieren, damit wir die Daten vom JSON-Format konvertieren und in unserem Code parsen können.

Wir haben zunächst zwei Methoden aufgesetzt: "Setup" und "Loop". Vor den Methoden haben wir Konstanten festgelegt, für die Stadt und das Land. Diese Konstanten brauchten wir für unseren API-Key, den wir dort eingefügt haben. Des Weiteren haben wir einen Timer für die Aktualisierung der Daten gebraucht. Damit die API auf dem M5Stick laufen kann, brauchen wir eine Hotspot-Verbindung, die wir durch Filips Handy sichergestellt haben. In der Setup- und Loop-Methode überprüfen wir die Internetverbindung zuallererst und dann nehmen wir die Daten der API und parsen diese als Objekt in unseren Code. Diese Daten können wir dann aufrufen und in unserer Konsole wie auch auf dem LCD-Display des M5Sticks anzeigen lassen. Hier ein Bild des Codeausschnittes:

```
//Checking if the wifi is connected with the m5stickc
if(WiFi.status()== WL_CONNECTED){

    jsonBuffer = httpGETRequest(serverPath.c_str());
    Serial.println(jsonBuffer);
    JSONVar myObject = JSON.parse(jsonBuffer);

    if (JSON.typeof(myObject) == "undefined") {
        Serial.println("Parsing input failed!");
        return;
    }

    //displaying the informations on the lcd-display of the m5stickc
    M5.Lcd.println("Zurich ");
    M5.Lcd.println("Temp °C: ");
    M5.Lcd.println(myObject["main"]["temp"]);
    M5.Lcd.println("Description: ");
    M5.Lcd.println(myObject["weather"][0]["description"]);
    M5.Lcd.println("Pressure: ");
    M5.Lcd.println(myObject["main"]["pressure"]);
    M5.Lcd.println("Humidity: ");
    M5.Lcd.println(myObject["main"]["humidity"]);
    M5.Lcd.println("Wind Speed: ");
    M5.Lcd.println(myObject["wind"]["speed"]);
```

Die Loop-Methode macht genau dasselbe, nur dort wird im Abstand von 10 Sekunden der Code immer wieder aktualisiert und dann auf dem M5Stick angezeigt.

Das Resultat sieht wie folgt aus:



Hier sieht man den Standort, die Temperatur, die aktuelle Wetterbeschreibung, den Luftdruck, die Luftfeuchtigkeit und die Windgeschwindigkeit. Alle Einheiten sind in unseren bekannten Einheiten angezeigt.

Weitere Ideen

Der Code funktioniert so, wie wir es uns vorgestellt haben. Wir hätten zusätzlich noch ein Bild der aktuellen Wettersituation einbetten können, nur dafür hatten wir leider keine Zeit.

Wir sind aber trotzdem zufrieden mit dem Resultat!