

Stonks

Autor: Eren Murat

Grupa: 334CA

Introducere

- Display LED pe care sunt afișate datele stock-urilor din portofoliu, în timp real.
- Stonks prezintă datele, cum ar fi prețul actual unui share, schimbarea zilnică în procente sau prețul de deschidere în mod minimalist.

Descriere Generala

Se cicleaza prin companiile aflate in portofoliu, iar la un interval de timp este inițiat un http request către API pentru noul simbol, care răspunde cu informațiile cerute într-un JSON. Acest JSON este parsat pe NodeMCU, iar datele sunt pasate prin driverele MAX7219 conectate in serie si ajung la matricea LED.

Hardware Design

Componente:

- NodeMCU
- Breadboard
- Sursa de alimentare pentru breadboard
- 5 drivere MAX7219
- 5 matrice LED 8×8
- Fire

Schema electrica:



Software Design

Mediul de dezvoltare folosit este Arduino IDE, cu configurările necesare pentru NodeMCU. Pentru afisare am inclus biblioteca MAX7219_Dot_Matrix.h care contine un alfabet pentru traducerea caracterelor intr-o stare a led-urilor. Biblioteca bitBangedSPI.h implementeaza protocolul SPI si este folosita in cadrul MAX7219_DotMatrix.h pentru a permite conectarea in serie (topologie daisy chain) a mai multor drivere MAX7219.

Initializez un display format din chips drivere care au CS conectat pe GPIO2.

```
MAX7219_Dot_Matrix display (chips, 2);
```

Stabilirea conexiunii reprezinta un proces simplu, acesta fiind unul dintre motivele pentru care am ales sa folosesc NodeMCU, in detrimentul unui combo Arduino Uno si ESP8266.

```
WiFi.mode(WIFI_STA);  
WiFiMulti.addAP("SSID", "password");
```

Este timpul sa trecem la loop, unde se intampla magia! La fiecare iteratie, verific intai starea conexiunii.

```
void loop(){  
    if ((WiFiMulti.run() == WL_CONNECTED)) {  
        // creez un client pentru care trimit cererile  
        std::unique_ptr<BearSSL::WiFiClientSecure>client(new  
BearSSL::WiFiClientSecure);  
        // do more stuff  
    }  
}
```

Pentru a ma asigura ca utilizatorul primeste informatii relevante, iar in acelasi timp numarul de cereri ramane rezonabil, am decis sa reimprospatez datele o data la un minut. De aceea, la fiecare loop verific daca au trecut 60 secunde de la ultima cerere:

```
    if (currentMillis - previousMillis >= interval) {  
        // execut o cerere pentru fiecare ticker din portofoliu  
        for (int i = 0; i < NUM_TICKERS; ++i) {  
            String request = "https://finnhub.io/api/v1/quote?symbol=" +  
tickers[i] + "&token=" + apiToken;  
            if (https.begin(*client, request)) {  
                // primesc raspunsul, verifica daca este valid si trec la  
parsare  
                // ...  
                String payload = https.getString();  
                DynamicJsonDocument doc(1024);  
                DeserializationError error = deserializeJson(doc, payload);  
                // ...  
            }  
            https.end();
```

```
    }  
}
```

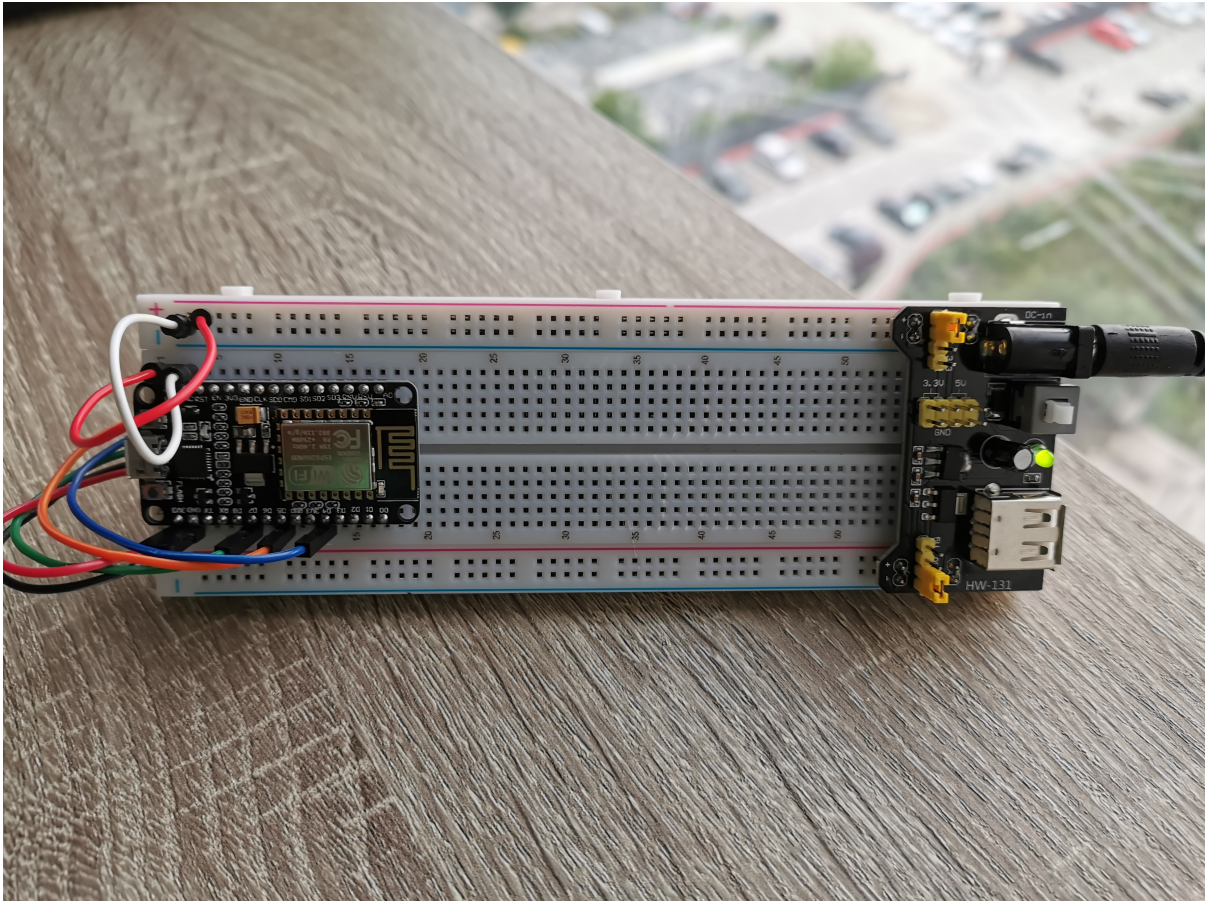
DynamicJsonDocument este un buffer folosit in cadrul versiunii 6 de ArduinoJSON pe care il initializez cu capacitatea fixa de 1024 octeti. Functia [deserializeJson](#) parseaza inputul JSON si scrie rezultatul in buffer, intorcand eventual o eroare.

La finalul fiecarui loop, afisez textul si il shiftez spre stanga. Acest proces se intampla la un interval de 40 milisecunde prin functia updateDisplay().

```
void updateDisplay() {  
    display.sendSmooth (message, messageOffset);  
  
    if (messageOffset++ >= (int) (strlen (message) * 8)) {  
        messageOffset = - chips * 8;  
    }  
}
```

Rezultate Obtinute

Stonks se comporta exact cum ma asteptam si imi ofera doar datele esentiale, cu precizie si usurinta. Mi-am propus sa creez un dispozitiv rapid, compact si independent de computer. Am reusit toate acestea cu un NodeMCU, un breadboard care ii ofera integritate structurala si o sursa de alimentare pe breadboard cu doua intrari (DC si USB) pentru a putea aseza device-ul oriunde. La inceputul dezvoltarii, l-am proiectat sa functioneze cu 5 drivere MAX7219, acest lucru traducandu-se intr-o rezolutie de 8×40. Din pacate, nu am prevazut incercarile esuate de a lega driverele si am ramas in prezent cu un display de 8×24. Astfel, citirea informatiile devine putin mai dificila, dar conceptul este acelasi si i se poate observa utilitatea in continuare.



Concluzii

In concluzie, doresc sa prezint planul de imbunatatire si extindere a *Stonks*. Voi construi un website prin care sa pot controla ticker-ele din portofoliu, trimitand comenzi de add si remove. In plus, voi oferi un nivel nou de utilitate prin posibilitatea afisarii orei si a datelor meteorologice. Toate acestea vor veni intr-o carcasa custom si *Stonks* va fi gata pentru a-si lua locul pe birou!

Download

Versiunea pdf a paginii se poate descărca de aici: [Stonks](#)

Bibliografie/Resurse

<https://www.gammon.com.au/forum/?id=11516>

<https://arduinojson.org/v6/doc/>

<https://finnhub.io/>

Demo

https://www.youtube.com/watch?v=-q5PAhELp_Y

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/cghenea/stonks>



Last update: **2021/06/04 17:41**