

Statie Meteo - Dumitru Vlad Alexandru

Introducere

Proiectul ales consta intr-o statie de monitorizare meteo. Sistemul capteaza date atat online (OpenWeather), cat si fizic (senzor de temperatura). Astfel se vor afisa (pe un LCD) detalii precum temperatura si umiditatea sprecifice localitatii si pozitiei curente (interior / exterior). Acest sitem are scopul de a usura accesul utilizatorului la datele meteo de interes (locale).

Descriere generală

Prin intermediul unui desktop/laptop conectat la Arduino se poate schimba/seta localitatea dorita si reseaua Wi-fi. Odata setate, este nevoie de o sursa separata de alimentare. Modulul Wi-fi CH340 cere (si primeste) date de la openweathermap.org la intervale de cateva minute si trimite informatiile primite catre Arduino. Senzorul DHT11 capteaza date similare la intervale de cateva secunde si trimite informatiile catre Arduino. Toate datele se vor afisa pe LCD (16x2) separat (localitate/local).

Schema bloc



Hardware Design

Lista de piese

- Arduino UNO
- Modul WiFi CH340
- Senzor de temperatura si umiditate DHT11
- LCD + modul I2C
- Breadboard
- Fire conexiune
- Rezistente

Scheme hardware



Software Design

Mediu de dezvoltare

[Arduino IDE](#)

Librarii si surse 3rd-party

- [Arduino_JSON](#)
- [DHTlib](#)
- [LiquidCrystal_I2C](#)
- [ESP8266 Package](#)

Surse si functii implementate

ESP8266.ino

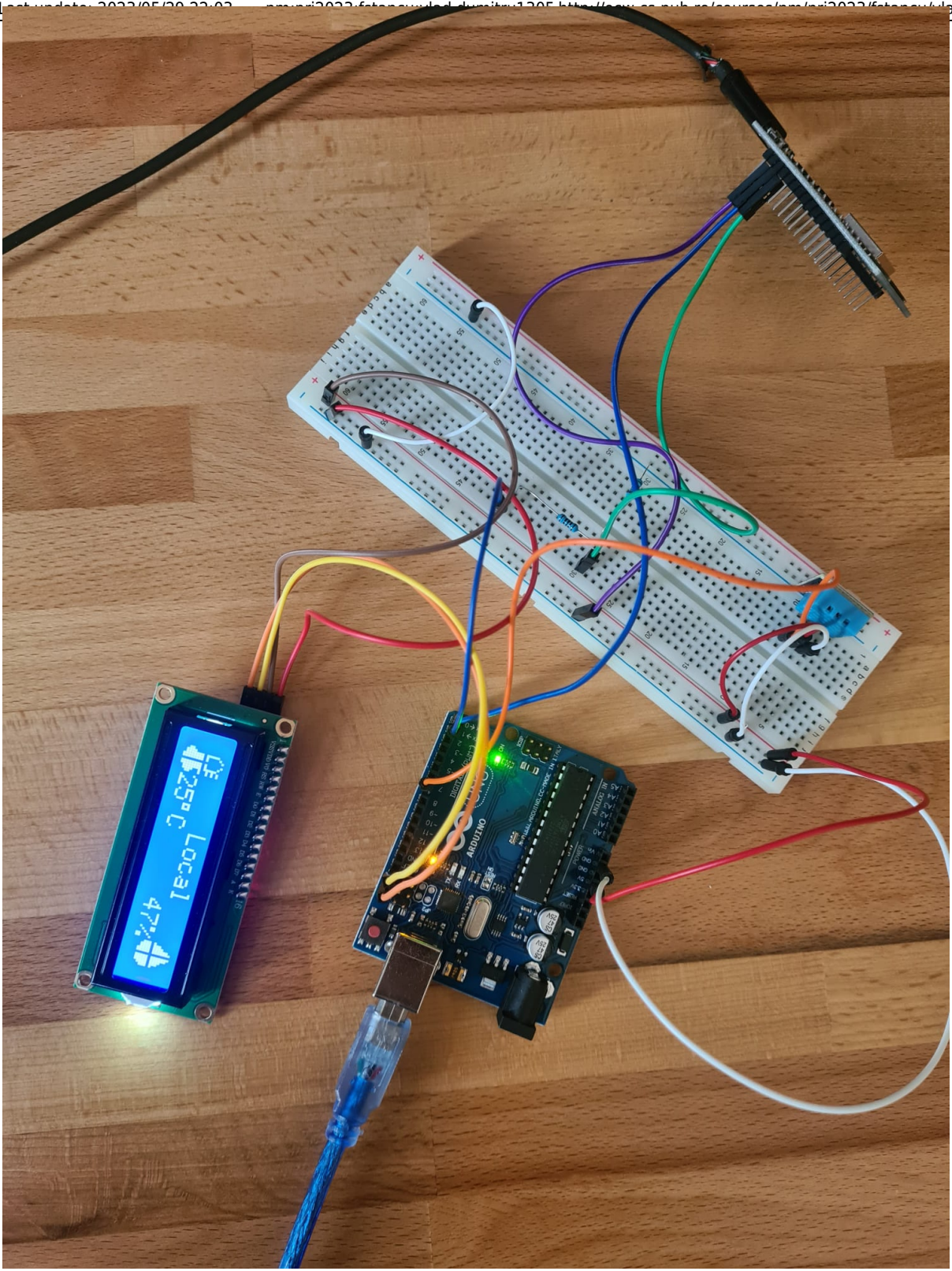
- `connectToWiFi()`
- Realizeaza conectarea la reseaua WiFi si asteapta o conexiune stabila.
- `httpGETRequest(const char* path)`
- Realizeaza conexiunea HTTP catre *path* dat ca argument.
- Realizeaza cerere HTTP de tip GET si returneaza rezultatul.
- `sendData()`
- Construieste pachetul de date - delimitat de caracterele < si >.
- Trimite pachetul catre dispozitivul conectat, prin interfata seriala USART.
- `setup(void)`
- Seteaza BAUD rate-ul.
- Apeleaza functia `connectToWiFi()`.
- `loop()`
- La intervale de 10 minute, este apelata functia `httpGETRequest(const char* path)`.
- Se prelucreaza rezultatul (intors sub forma de JSON) si se inregistreaza temperatura si umiditatea curente.
- In cazul in care conexiunea WiFi a fost intrerupta, se va incerca reconectarea.

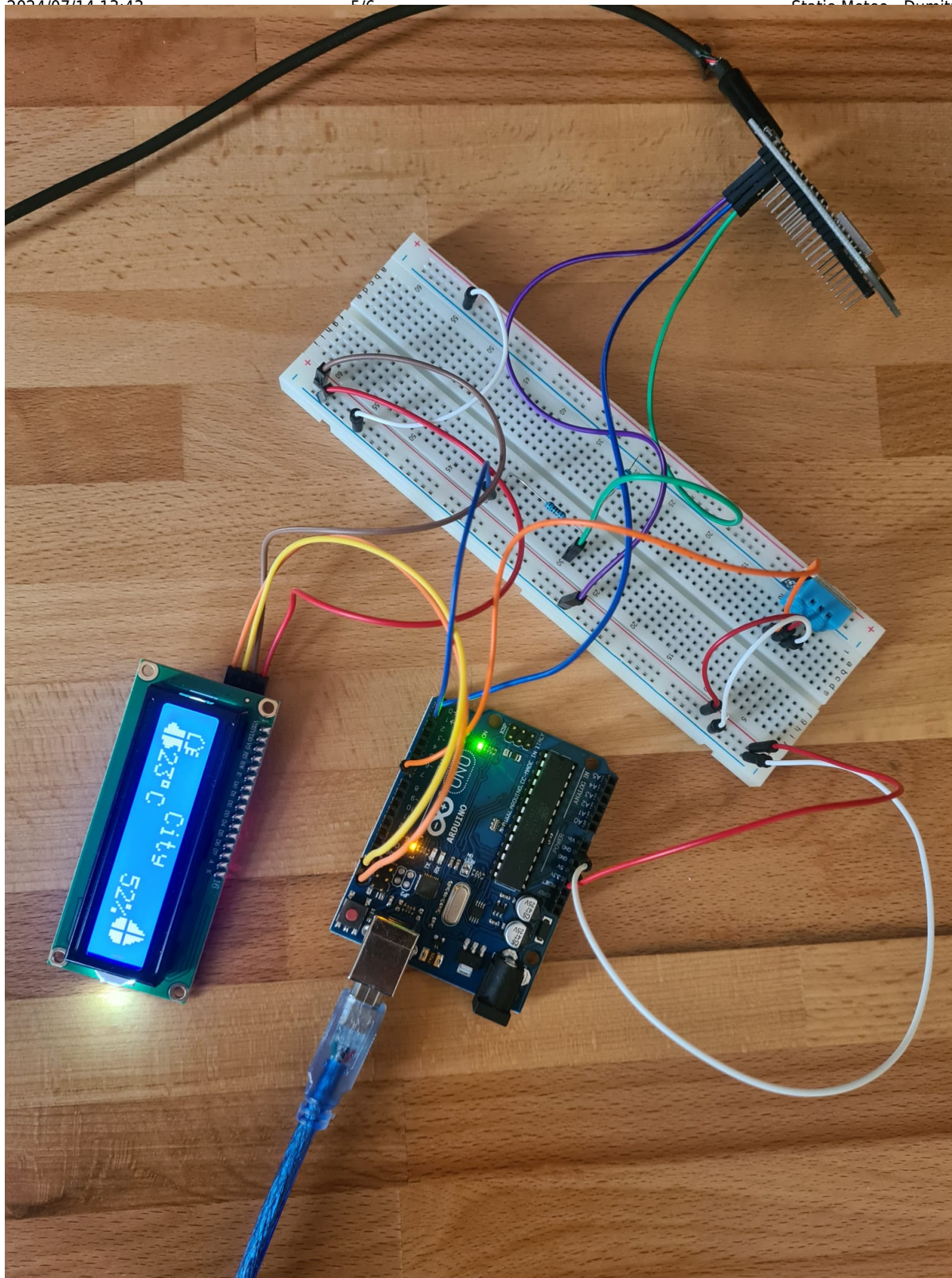
- ~~Datele inregistrate se trimit constant, actualizandu-se doar dupa intervalul de 10 minute.~~

Arduino.ino

- `printLCD(int temp, int humidity, int location)`
- Scrie pe LCD datele primite ca argument (impreuna cu simboluri expresive).
- `printDHT()`
- Inregistreaza datele citite de catre senzorul DHT.
- Apeleaza functia `printLCD(int temp, int humidity, int location)` cu datele citite.
- `printWiFi()`
- Daca exista date disponibile pentru a fi citite, se vor inregistra, prelucra si afisa (asemenea DHT).
- `setup(void)`
- Seteaza BAUD rate-ul.
- Initializeaza LCD-ul.
- Construiește simbolurile folosite in afisare.
- `loop()`
- Apeleaza functiile `printDHT()` si `printWiFi()` cu o pauza de 5 secunde intre ele.

Rezultate Obținute





Concluzii

- Proiectul poate deveni chiar folositor dacă și se îmbunătățește portabilitatea + folosirea unui cadru pentru piese.
- A fost interesant procesul de realizare a proiectului - de la alegerea pieselor la scrierea codului.

- Nu cred ca voi refolosi piesele utilizate.

Download

[surse_meteo.7z](#)

Bibliografie/Resurse

Resurse Software / Hardware

- <https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>
- <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>
- <https://www.hackster.io/jacoslabbert99/arduino-lcd-icons-custom-characters-548f38>
- <https://www.instructables.com/loT-ESP8266-Series-1-Connect-to-WIFI-Router/>
- <https://randomnerdtutorials.com/esp8266-nodemcu-http-get-post-arduino/>
- <https://robotic-controls.com/learn/arduino/arduino-arduino-serial-communication>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/fstancu/vlad.dumitru1305>



Last update: **2023/05/29 22:03**