Desktop robot

Introducere

Proiectul constă într-un robot pentru desktop, dispozitiv ce are ca scop ușurarea activităților de acasă sau de la birou. Acest robot oferă diferite funcționalități utilizatorului: poate afișa temperatura din încăpere, umiditatea, presiunea aerului, ora, sau chiar și notificări pe care utilizatorul le poate seta dintr-o aplicație ce controlează robotul. Cu această aplicație utilizatorul poate alege exact ce va fi afișat pe ecran. Utilitatea acestui robot vine de la faptul că oferă o interfață interactivă utilizatorului și aduce un plus de confort prin oferirea mai multor informații la un loc.

Descriere generală

Robotul este controlat printr-o aplicație. Astfel, este la latitudinea utilizatorului ce va afișa pe ecranul robotului: poate afișa ora curentă, temperatura camerei, umiditatea sau presiunea. De asemenea, dacă utilizatorul dorește să își pună reminders sau notificări pentru viitor, acesta și le poate seta din aplicație, și vor apărea pe ecranul robotului și se va aprinde un LED pentru a semnala afișarea unei notificări.

Odată ce utilizatorul s-a decis ce să fie afișat pe ecran, acesta poate să schimbe prin funcționalitatea touchscreen a ecranului diferitele moduri de afișare ale orei, temperaturii etc, sau chiar și să aleagă diferitele “fețe” ale robotului.

Hardware Design

Listă componente:

- Modul ESP32 Dev Kit
- Plăcuță PCB prototipare 10x10cm
- 2.8'' SPI TFT display touchscreen
- Modul ceas RTC DS3231
- Senzor umiditate si temperatura DHT11
- Modul senzor de presiune atmosferică BMP280

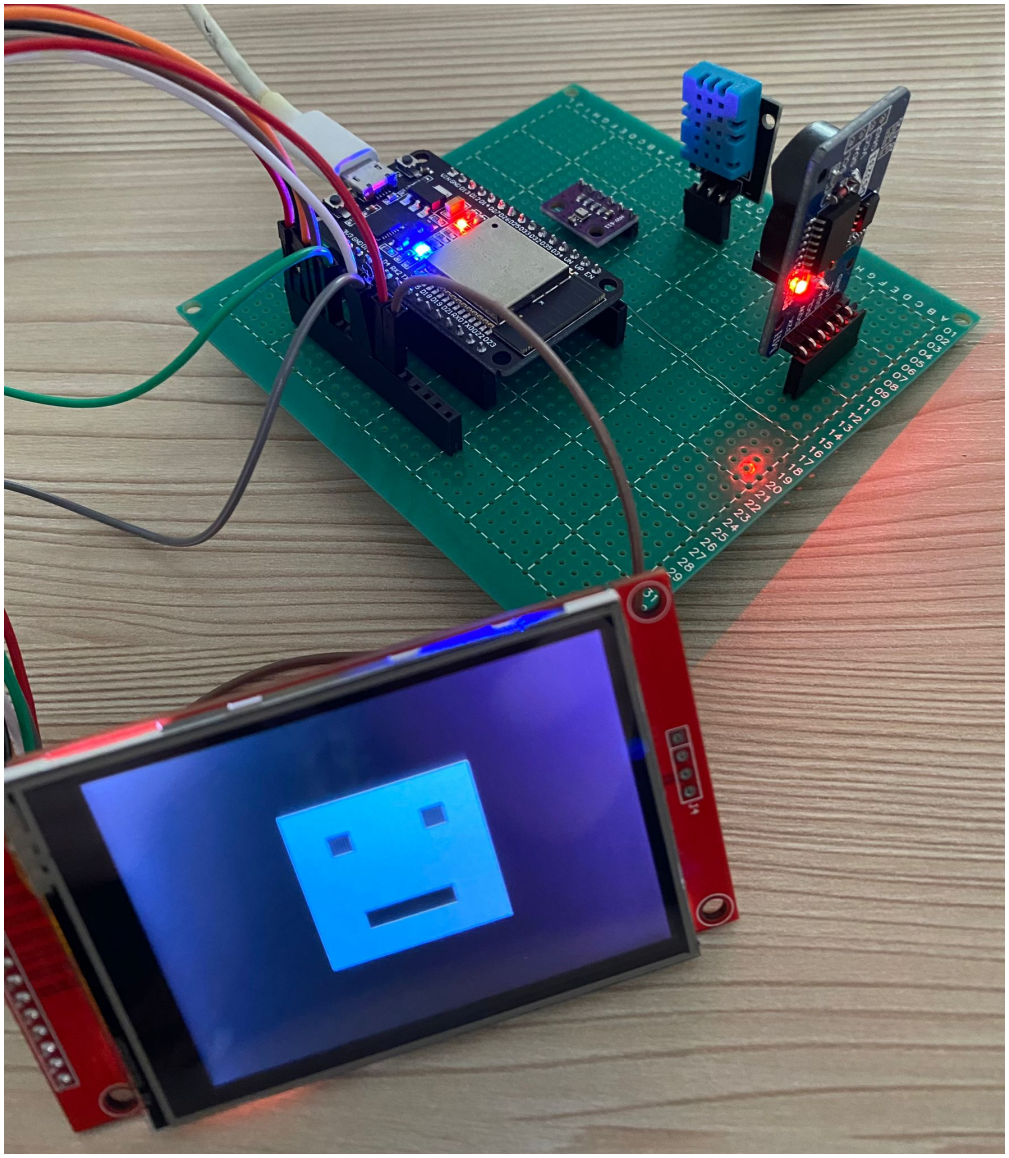
Schematic:

Circuitul a fost realizat pe o placă de prototipare PCB de 10x10cm. Toți senzorii sunt legați la modulul ESP32:

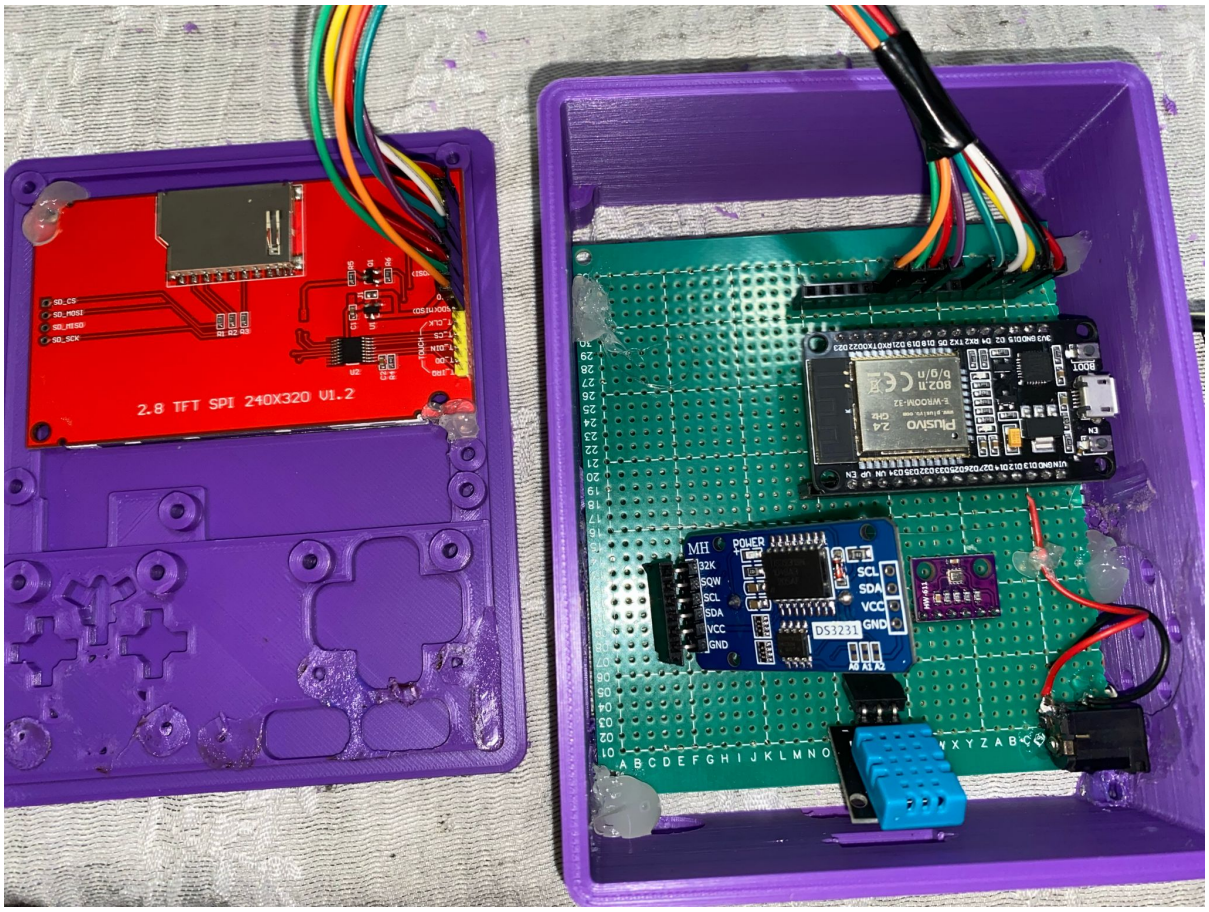
- Temperatura și umiditatea sunt citite cu ajutorul senzorului DHT11
- Ora curentă și data sunt citite cu senzorul RTC D23231 prin I2C
- Presiunea aerului este citită cu senzorul BMP280 prin SPI

În funcție de preferința utilizatorului, datele vor fi afișate pe un display LCD ce comunică prin SPI.

Etapa 1:



Etapa finală



Aici asamblat plăcuța în carcasa imprimată 3D. Pentru a lipi display-ul și plăcuța am folosit silicon, și pentru alimentare am folosit un battery holder cu mufă jack și 4 baterii de 1.5V.

Software Design

Mediul de dezvoltare folosit a fost Arduino IDE. Biblioteci folosite:

- Pentru display: **TFT_eSPI.h**
- Pentru senzorul de temperatură: **DHT.h**
- Pentru modulul de ceas: **RTClib.h** si **Wire.h**
- Pentru senzorul de presiune atmosferică: **Adafruit_BMP280.h**
- Pentru crearea unui server web asincron: **WiFi.h** si **ESPAsyncWebServer.h**

Pentru modulul de ceas, care functioneaza prin I2C, a trebuit sa creez separat o instanta de TwoWire pentru a seta 2 pini de pe placuta ESP32 pentru comunicare I2C, deoarece nu am folosit pinii I2C default de pe aceasta.

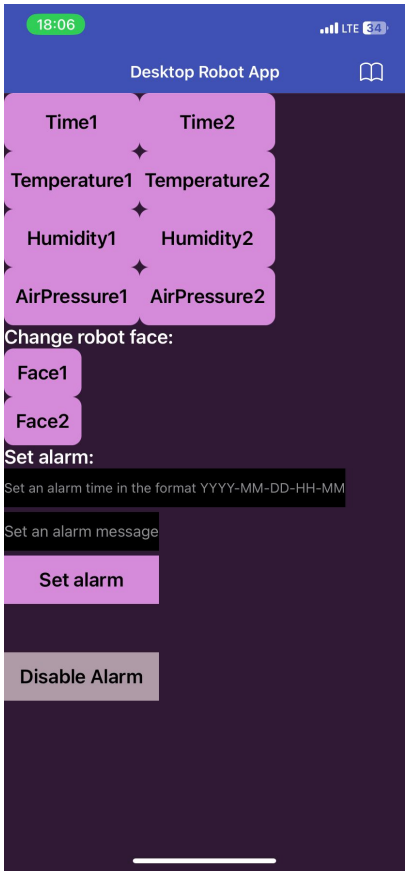
In functia de **setup()**, m-am ocupat de initializarea tuturor senzorilor si componentelor necesare:

- **Serverul web asincron** si un handler pentru request-urile facute din aplicatie. Mai intai, robotul se conecteaza la Wi-fi si din aplicatie se pot alege diverse moduri de afisare a ecranului si ce exact sa se afiseze pe ecran. Fiecare buton apasat din aplicatie va fi un request de tip GET pe ruta "http://<IP_ESP32>/display?mode=valoare". Valoarea va determina ce se va afisa pe ecran la momentul curent de timp (temperatura, data si ora, umiditatea sau presiunea aerului). De

asemenea, exista si ruta “http://<IP_ESP32>/display?time=valoare_timp&message=mesaj” pentru ca utilizatorul sa isi poata seta o notificare/reminder ce va fi afisat pe ecran atunci cand isi doreste. Odata ce va fi afisata notificarea respectiva, utilizatorul poate dezactiva notificarea apasand un buton de “Disable” ce va trimite un request de “http://<IP_ESP32>/display?alarm=stop”.
- Am initializat aici **modulul de ceas** si instanta de TwoWire pentru aceasta
- Am initializat **senzorul de presiune atmosferica** BMP280. Senzorul poate fi folosit sa comunice ori prin I2C ori prin SPI, insa am facut comunicarea prin SPI, si am setat sampling mode-ul senzorului la cel normal.
- Am initializat si un **timer** pe care il folosesc la modul “default” de afisare al ecranului: cand inca nu a apasat utilizatorul niciun buton, robotul isi schimba fetele odata la 10 secunde.

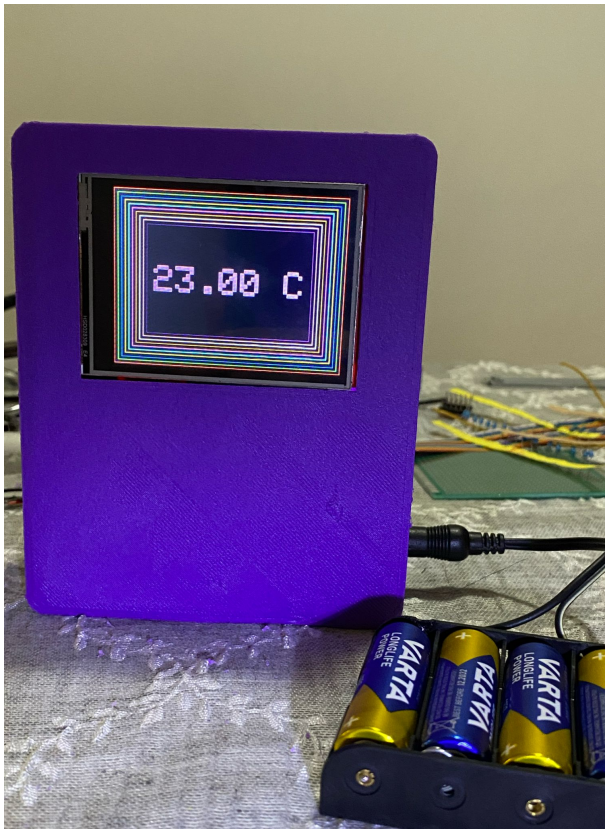
In functia **loop()** am implementat flow-ul display-ului: utilizatorul poate afisa pe ecran urmatoarele lucruri: temperatura, umiditatea, data si ora, si presiunea atmosferica si altitudinea. Fiecare dintre aceste componente vin in doua moduri diferite de afisare pe care le poate schimba utilizatorul oricand din aplicatie.

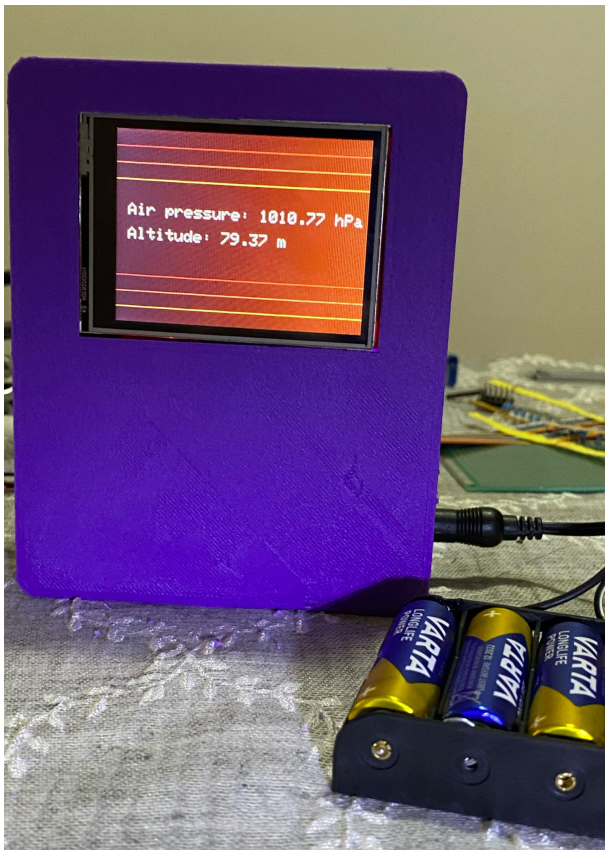
Aplicația a fost făcută cu MIT App Inventor.



Rezultate Obţinute

Acesta este unul dintre modurile de afișare ale temperaturii și unul pentru presiunea atmosferică și altitudine.





Concluzii

Din păcate, nu am reușit să fac să funcționeze touchscreen-ul la display, însă per total a fost un

proiect interesant și de la care am învățat multe lucuri (și răbdare).

Download

proiect_pm_ada.zip

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

- https://www.luisllamas.es/en/esp32-timers/
- https://www.instructables.com/How-to-Connect-BMP-280-to-ESP32-Get-Pressure-Tempe/
- https://electropeak.com/learn/interfacing-2-8-inch-tft-lcd-touch-screen-with-esp32/
- https://randomnerdtutorials.com/esp32-tft-touchscreen-display-2-8-ili9341-arduino/
- https://learn.adafruit.com/adafruit-gfx-graphics-library/minimizing-redraw-flicker
- https://microcontrollerslab.com/esp32-ds3231-real-time-clock-rtc-oled/

Export to PDF