

PROIECT INFORMATICA APLICATA

NOKIA LCD CONTROL USING BLUETOOTH LOW ENERGY

NANESCU EDUARD-DANIEL

TAIGA RAUL-ALEXANDRU

GRUPA 411A

CUPRINS

1. Introducere
2. Considerente teoretice
3. Implementare
4. Concluzii
5. Bibliografie

I. INTRODUCERE

Prezenta lucrare practică are în vedere realizarea unui program capabil să redea pe un ecran de tip [Nokia 5510/3310 LCD](#) text, imagini și derularea conținutului afișat în ambele direcții (stânga sau dreapta) prin utilizarea unor comenzi specifice.

Pentru a urmări scopul lucrării a fost nevoie de următoarele componente:



Fig. 1 - ESP32



Fig. 2 - Ecran Nokia 5510



Fig. 3 - Potentiometru



Fig. 4 - Rezistente



Fig. 5 - Fire Jumper



Fig. 6 - LED



Fig. 7 - Display LED 3 cifre



Fig. 8 - Breadboard

II. CONSIDERENTE TEORETICE

1. Modulul ESP32:

În vederea realizării proiectului, am utilizat placa de dezvoltare ESP32 oferită de la facultate și componente adiționale, achiziționate separat.

Modulul ESP32 este un modul SoC (System on Chip) fabricat de compania Espressif Systems, bazat pe microprocesorul Tensilica Xtensa LX6 cu unul sau două nuclee și o frecvență de lucru de între 160 și 240MHz precum și un coprocesor ULP (Ultra Low Power). Suplimentar, acesta dispune de comunicație WiFi și Bluetooth (clasic și low-energy) integrate, precum și de o gamă largă de interfețe periferice.

ESP32 este facut pentru dispozitive mobile, electronice wearable și aplicații Internet-of-Things (IoT). Prezintă toate caracteristicile de ultimă generație ale cipurilor de putere scăzută, inclusiv scalarea dinamică a puterii.

Spre exemplu, într-o aplicație de IoT de putere scăzută cu un hub de senzori, plăcuța ESP32 pornește periodic, atunci când întâlnește o condiție anume. Ciclul de putere redusă este folosit pentru a reduce cantitatea de energie de care se folosește cipul. Output-ul amplificatorului de putere este de asemenea ajustabil, contribuind astfel la un compromis între raza de comunicare, rata de transfer a datelor și consumul de putere.

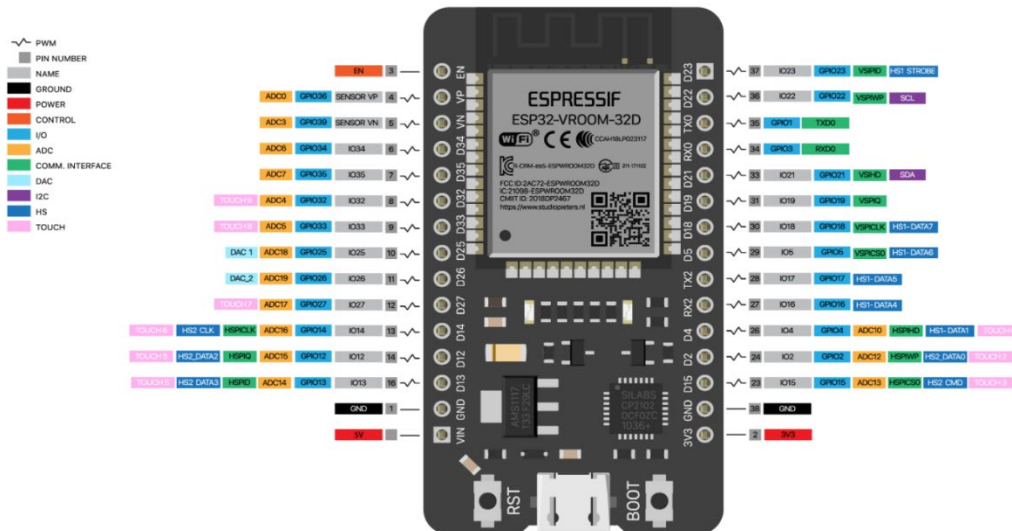


Fig. 8 – ESP32

2. Protocoalele si metodele de comunicatie utilizate:

Componenta software este cea care asigură funcționalitatea fiecărei componente hardware - este un ansamblu de programe, rutine. Placa de dezvoltare ESP32 se conectează la computer prin intermediul unui cablu USB, urmând a face legătura dintre aceasta și mediul de programare. VSCode este un mediu de programare open-source care facilitează scrierea de cod și încărcarea acestuia pe microcontroler, alături de extensia PlatformIO. Tipurile de metode de comunicatie folosite sunt enumerate dupa cum urmeaza:

Consola seriala – este un terminal ce faciliteaza comunicarea intre mediul de dezvoltare si placuta ESP, similar Command Prompt-ului din Windows sau terminalului din Linux.

JSON (JavaScript Object Notation) – o modalitate de stocare a datelor ce le reprezinta sub forma de text, fiind usor de prelucrat in majoritatea limbajelor de programare. Acesta poate stoca de la valori simple, la structuri de date complexe.

WiFi – o familie de protocoale de retea wireless folosite pentru conectarea la retele locale si internet. In cazul de fata, ESP-ul nostru este setat in modul STA (station) pentru a se conecta la un access point disponibil. Placuta cauta prima retea disponibila (fara parola) si se conecteaza la aceasta. In cazul in care nu gaseste nicio retea de acest tip, afiseaza un mesaj in consola seriala.

- In proiect, este utilizat pentru a prelua informatii serializate JSON de pe un server extern prin intermediul protocolului HTTP, metoda GET. Astfel programul nostru face rost de un sir de caractere serializat pe care urmeaza sa il proceseze. In cazul in care metoda GET esueaza (primim un response code diferit de 200), programul va afisa in consola seriala un mesaj corespunzator.

HTTP (HyperText Transfer Protocol) – protocolul de transfer al informatiei specific aplicatiilor web. Acesta a fost creat pentru a facilita comunicarea între clienți și servere. Funcționează foarte ușor, urmând modelul client-server. Clientul inițiază o conexiune pentru a transmite o cerere, și așteaptă să primească un răspuns de la server.

- **Metoda GET** – o reprezentare a unei anumite resurse de orice tip (dacă fișierul este o pagină WEB, atunci acesta va returna codul HTML, dacă este un fișier text, va returna textul din acel fișier, etc).

Bluetooth Low Energy (BLE) – varianta cu un consum mai mic de energie a bine-cunoscutului protocol Bluetooth, cu o rază mai mică de acțiune, dar aceeași bandă de 2.4GHz. Aceasta funcționează fundamental diferit de predecesorul său, operând pe baza de unul sau mai multe servicii ce pot conține la rândul lor, mai multe caracteristici, care mai departe conțin valorile ce reprezintă informația propriu-zisă.

- Caracteristicile pot fi configurate pentru 3 tipuri de operații: read, write și notify, astfel încât dacă o caracteristică este modificată de operația write, clientul este notificat (dacă caracteristica respectivă permite notificarea).

3. Componente utilizate:

A. ESP32:



Mai bine spus “creierul” din spatele proiectului nostru, ce se ocupa cu rularea programului scris de noi despre care am discutat [anterior](#).

B. Breadboard



Un [breadboard](#) este o placă de prototipare care permite realizarea rapidă și cu ușurință a circuitelor. În interior se află unele plăcuțe metalice (contacte) dispuse în sistem grilă. Pe astfel de plăci se testează funcționalitatea circuitelor, înainte de realizarea circuitului imprimat propriu-zis.

C. Ecran Nokia 5510/3310 LCD



Display de tip LCD, Nokia 5110, cu o rezoluție de 84x48 pixeli, compatibil Arduino și alte plăci de dezvoltare.

Funcționează pe o tensiune logică de 3.3V și este capabil să afișeze atât caractere alfanumerice cât și imagini. Operează prin conectarea celor 8 pini la ESP-32.

D. Potentiometru



Un [potențiomtru](#) este un instrument pentru variația potențialului electric (tensiune) într-un circuit. Potențiometrul este un simplu dispozitiv folosit pentru a măsura EMF, TPD, și rezistența internă a unei celule. Acesta constă dintr-o placă cu o sârmă de wolfram sau mangan montată pe ea. Acesta funcționează pe principiul faptului că căderea de potențial între două puncte ale unei sârme cu secțiune transversală uniformă, este direct proporțională cu distanța dintre puncte.

E. Rezistori



O altă componentă importantă în realizarea proiectului o constituie [rezistorii](#). Rezistorii sunt componente electrice care limitează sau regulează fluxul de curent electric printr-un circuit. Valoarea rezistenței unui rezistor este definită ca raportul dintre tensiunea pe porțiunea respectivă de circuit U și curentul electric I ce parcurge acea porțiune de circuit.

F. Fire jumper si cablu USB



Pentru conectivitatea dintre componentele de pe placa de dezvoltare, am utilizat [fire jumper](#), pentru a conecta pinii de date, ground sau pentru a alimenta cu electricitate diversii consumatori. Pentru a realiza conectarea dintre placa de dezvoltare și computer am folosit un cablu [USB tip A în microUSB](#). USB (Universal Serial Bus) este o interfață de tip plug-and-play care permite comunicarea între dispozitive și periferice.

G. LED



Un [LED](#) este o diodă semiconductoare ce emite lumină la polarizarea directă a joncțiunii p-n. Efectul este o formă de electroluminescență.

În proiect este folosit pentru a ne indica primirea datelor prin intermediul BLE de la client.

H. Display LED 3 cifre



Are aceleași caracteristici și funcționalitate precum LED-ul. În proiect [Display-ul](#) este folosit pentru a ne indica starea dispozitivului. După ce acesta se porneste va afișa "OK" pe ecran, iar la conectarea unui client prin BLE, va afișa "Con".

III. IMPLEMENTARE SOFTWARE

Cum functioneaza?

Aplicatia incarcata pe placuta ESP-32 foloseste toate modurile de I/O disponibile, precum WiFi, Bluetooth Low Energy, intrarea micro-usb pentru incarcarea programului si desigur, pinii laterali prin care comunicam cu celelalte componente instalate pe placa de prototipare.

Aceasta are o structura simpla, dar robusta, care permite comunicarea cu un client conectat prin Bluetooth Low Energy, ce transmite date serializate JSON, pe care programul le deserializeaza si le interpreteaza, urmand ca apoi sa realizeze actiunile cerute (daca primeste o actiune valida), iar in final sa trimita un raspuns corespunzator catre client, serializat de asemenea JSON.

Dupa pornirea placutei aceasta seteaza modul pinului 21 ca OUTPUT pentru LED-ul ce se aprinde la primirea datelor prin BLE. Apoi, afiseaza un mesaj de intampinare pe ecranul Nokia 5510 pentru a ne arata ca totul functioneaza corespunzator cu acesta. Mai departe, porneste serverul BLE si incearca sa se conecteze la o retea WiFi. In final, pe display-ul LED este afisat "OK", iar LED-ul verde se va aprinde de 3 ori pentru a indica finalizarea procesului de initializare.

La conectarea unui dispozitiv prin BLE, pe display-ul LED este afisat "Con", si asteapta un mesaj serializat JSON valid pentru a realiza actiunea ceruta. In cazul in care placuta nu este conectata la WiFi, actiunea setImage nu va functiona, iar un mesaj corespunzator va fi afisat in consola seriala.

Organigrama codului poate fi vizualizata pe urmatoarea pagina.

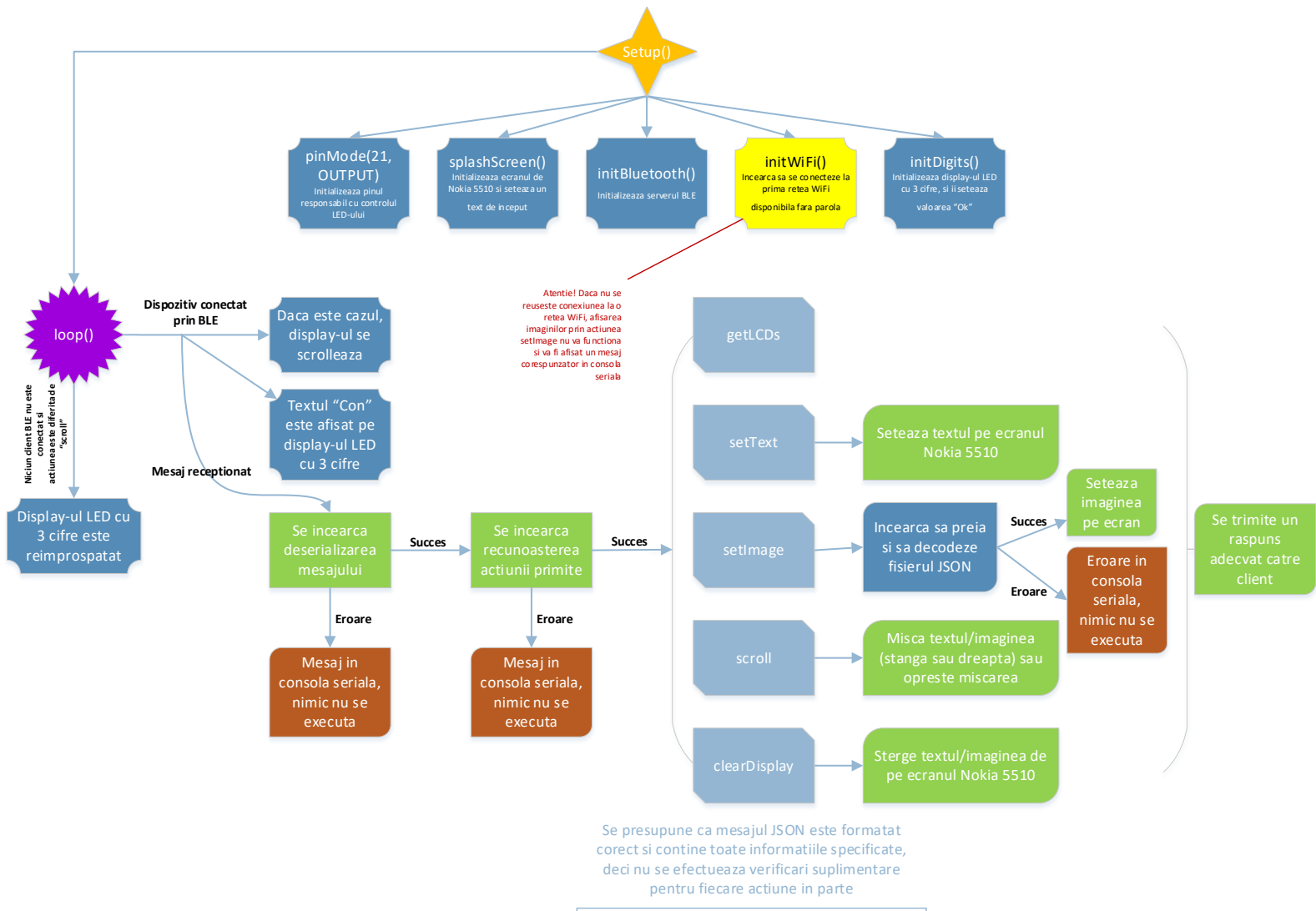


Fig. 9 – Organigrama codului

IV. CONCLUZII

În concluzie, suntem de părere că implementarea proiectului a fost un succes, având în vedere respectarea cerințelor impuse. Proiectul poate avea o gamă variată de întrebări, putând fi cu ușurință modificat și adaptat nevoilor ulterioare având în vedere afișajul ofertant de tip Nokia 5510 și varietatea de moduri de conexiune folosite.

Pe parcursul dezvoltării am întâmpinat dificultăți în serializarea JSON a mesajelor de răspuns către client, dar după căutări succesive, am reușit să soluționăm problema, înlocuind biblioteca pe care o foloseam pentru interacțiunea cu JSON (de la rapidjson la ArduinoJSON).

O îmbunătățire pe care am putea să o aducem proiectului ar fi adăugarea mai multor opțiuni de formatare a textului pe ecranul de Nokia, având în vedere că acesta suportă marimi diferite ale fontului, culori de fundal pentru acesta și rotirea pe verticală.

V. BIBLIOGRAFIE

Informațiile referitoare la modulul ESP32, metodele de comunicație (WiFi, JSON, BLE, HTTP) sunt extrase din materialele suport oferite pe Moodle.

[Ecran Nokia](#)

[Breadboard](#)

[Potentiometru](#)

[Rezistori](#)

[LED](#)

VI. ANEXA (COD SURSA)

Codul poate fi vizualizat online la adresa:

<https://github.com/Moshulika/PIA-ESP32/blob/master/src/main.cpp>