Calculator de buzunar++

Nume: Negoiță Vlad-Andrei
Grupa: 331CA

Introducere

(acum mult, mult timp...) Eram la ora de fizică și, la o problemă destul de simplă, am ajuns la următoarea expresie de calculat:

$$ x = \frac{0.378 * 0.294 ^ 3 - \sqrt{0.764} * 0.245 ^ 2}{0.532 - 0.654}$$

După două încercări (evident nereușite) de a calcula, am tras concluzia că voi rămâne la calculul simbolic . Totuși, pentru când un rezultat numeric este important (de exemplu, poate fi interpretat mai departe), e nevoie de o metodă mai robustă decât a bate manual la calculator zeci de cifre. Pentru această situație am creat Calculatorul de buzunar++: un device simplu care, folosind PhotoMath, reduce semnificativ erorile umane în calculele numerice de rutină.

Descriere generală

Device-ul respectă următoarea secvență de pași:

1. așteaptă primirea unui semnal pentru capturarea unei poze (ex. apăsarea unui buton)
2. poza este împachetată și, folosing API-ul de la PhotoMath, este trimisă pentru a fi descifrată și, eventual, rezolvată
3. rezultatul este primit și procesat, urmând afișarea pe display-ul LCD al device-ului

Hardware Design

Pentru realizarea acestui proiect, am achiziționat următoarele componente:

- ESP32
- Camera OV7670
- Display LCD cu interfață I2C
- Breadboard, fire de legătură

Schema electrică:

Software Design

Pentru acest proiect, am folosit Arduino IDE. Codul poate fi accesat la adresa https://github.com/VladNegoita/Calculator.

În continuare, va fi descrisă succint modul de funcționare al componentei software.

După realizarea conexiunilor fizice, display-ul funcționează aproape imediat întrucât am folosit o bibliotecă care ușurează semnificativ utilizarea acestuia: **LiquidCrystal_I2C**.

Pentru driverele camerei, am identificat o soluție implementata pe github: https://github.com/bitluni/ESP32CameraI2S (există și un videoclip explicativ pe youtube asociat).

Camera realizează poze de rezoluție destul de mică, nefiind tocmai satisfăcătoare în practică. Pentru a ajunge la această concluzie, am folosit un WebServer unde am încărcat pozele realizate de cameră periodic și am interogat folosind Postman.

Întrucât pozele sunt primite într-un format **.bmp** și **.jpeg** este necesar pentru utilizarea API-ului este nevoie de o conversie între aceste formate. Întrucât nu există suport oferit pentru o astfel de operație pentru microcontroller-ul ales, am decis ca un script de python să se ocupe de conversie și de apelul API-ului, întorcând rezultatul pe serială. Această pivotare nu deviază de la scopul inițial al proiectului, întrucât doar furnizează un intermediar (un potențial server pentru o aplicație reală a calculatorului).

Rezultate Obţinute

Proiectul e capabil sa recunoască cifre, iar în condiții ideale poate rezolva și ecuații foarte simple. Singura limitare este camera, ce poate face poze de o rezoluție și calitate îndoielnică.

Concluzii

Proiectul este un PoC (proof of concept) și trebuie tratat ca atare. Am ajuns la concluzia că acest domeniu necesită multă răbdare și o planificare extrem de bună a resurselor. Pe lângă asta, documentația (incluzând datasheet-uri și resurse online) este extrem de limitată, asta fiind o problemă destul de mare în realizarea proiectului.

Bibliografie/Resurse

- https://github.com/bitluni/ESP32CameraI2S
- https://github.com/igrr/esp32-cam-demo
- https://www.optimusdigital.ro/ro/index.php?controller=attachment&id_attachment=204
- https://www.optimusdigital.ro/ro/index.php?controller=attachment&id_attachment=1479
- https://www.waveshare.com/wiki/LCD1602_I2C_Module

Export to PDF