

Radio Fm

Introducere

Acest proiect consta in realizarea unui aparat radio FM care ofera posibilitatea de a asculta orice statie radio din intervalul 87.5 - 108.0 MHz cu posibilitatea de a salva postul ascultat pe unul dintre cele patru canale implementate.

Ideea a pornit de la pasiunea aparatelor radio si a muzicii. Aparatul radio are o utilitate zilnica pentru oamenii care isi doresc sa asculte muzica, stiri, emisiuni sau pentru cei ce isi doresc un fundal sonor placut in timp ce se ocupa de treburi casnice sau teme.

Schema bloc



Hardware Design



Lista piese

Arduino UNO R3	Optimus digital
Rotary Encoder Module	Optimus digital
KCD10-101 Switch	Optimus digital
400p HQ Breadboard	Optimus digital
1602 LCD	Optimus digital
2 x 18650 Battery Case	Optimus digital
4x White Button with Round Cover	Optimus digital
Buzzer	Optimus digital
TDA2822	Optimus digital
TEA5767	ArduShop
Difuzor	tehnoelectric
4xRezistentă 10k	
5xLed	
Rezistentă 220	

Fire	
------	--

Software Design

Descrierea codului aplicatiei (firmware)

Firmware-ul este dezvoltat pentru un radio digital construit in jurul platformei Arduino Uno R3. Proiectul permite utilizatorului sa asculte posturi FM, sa comute intre ele, sa caute automat frecvente, sa seteze un temporizator de oprire (sleep timer), sa activeze functia mute si sa salveze posturi preferate in memoria EEPROM.

Interfata cu utilizatorul se face prin intermediul unui ecran LCD 1602 (cu interfata I2C), un rotary encoder (cu switch), patru butoane, si un switch pentru comutarea modului de functionare al butoanelor.

Mediu de dezvoltare

- **Placa:** Arduino Uno R3
- **Mediu de dezvoltare:** Arduino IDE
- **Limbaj de programare:** C++

Librarii si surse third-party

- **TEA5767N.h:** librarie folosita pentru controlul modului radio FM TEA5767.
- **LiquidCrystal_I2C.h:** pentru afisarea informatiilor pe ecranul LCD 1602 cu interfata I2C.
- **EEPROM.h:** librarie nativa Arduino pentru citirea si scrierea datelor persistente (posturi salvate).

Structura codului

- **Initializare Sistem**
 - Se initializeaza componentele: LCD, radio TEA5767, buzzer, butoane, encoder, LED de stare.
 - Se incarca frecventa radio salvata anterior din EEPROM.
 - Se afiseaza informatiile initiale pe LCD (frecventa, semnal, canal memorat).
- **Control Encoder**
 - Encoderul incrementa sau decrementeaza frecventa curenta.
 - Se detecteaza directia de rotatie (stanga/dreapta) si se actualizeaza frecventa.
 - Dupa un interval (default 1 secunda), se face actualizarea efectiva a semnalului radio si LCD-ului.
- **Butonare - Functionalitati Multiple**
 - Fiecare dintre cele 4 butoane are 2 functii, in functie de starea comutatorului fnSw (Function

Switch):

- Salvare/Accesare posturi memorate (FM1-FM4): cu fnSw activat, fiecare buton incarca un canal memorat si actualizeaza frecventa.
- Functii speciale (cu fnSw dezactivat):
 - Buton 1/2: cautare automata a postului anterior/urmator (search()).
 - Buton 3: Mute/Unmute.
 - Buton 4: Timer pentru oprire automata sau reactivare din standby.
- **Timer de Oprire Automata**
 - La apasarea butonului 4 in mod normal, se poate activa un cronometru de oprire (sleepTimer).
 - Utilizatorul poate selecta durata in minute (de la 1 la 10 minute).
 - Dupa expirare, sistemul intra in standby: radio se opreste, LED-ul si afisajul sunt dezactivate.
- **EEPROM - Salvare si Incarcare Date**
 - Se salveaza frecventa curenta intr-o adresa EEPROM corespunzatoare (FM1-FM4).
 - La repornirea sistemului, frecventa si butonul utilizat anterior sunt incarcate automat din EEPROM (LAST).

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

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

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/apredescu/luca.spataru>



Last update: **2025/05/28 12:05**