Boxa Smart "Spectrum"

Introducere

Spectrum este o boxa inteligenta ce poate stoca si reda piesele tale favorite. Este un dispozitiv portabil cu o interfata prietenoasa ce poate fi folosit in timp ce iti faci temele, in timpul unui antrenament sportiv sau pur si simplu cand ai chef sa asculti muzica. Proiectul a fost gandit initial pentru a refolosi anumite piese precum o baterie de la un telefon vechi sau un difuzor de la o boxa stricata pentru a realiza un produs util.

Descriere generală

Boxa prezinta o lista de cantece preincarcate pe cardul SD. Cantecele sunt scrise bare-metal, direct pe cardul SD, folosind un script in Python de pe Laptop si interfata seriala. Aceasta este afisata pe ecran de unde utilizatorul poate, folosind encoderul rotativ, selecta piesa dorita. Odata ce este selectata o piesa, se poate regla volumul, pune pauza, sau se poate reveni la meniul anterior.

Hardware Design

Lista pieselor folosite:

- Placa Atmega328p Xplained
- Difuzor 2W 8 Ohm Reciclat
- Amplificator PAM8403
- Acumulator 2200mAh Galaxy Core 2 Reciclat
- Modul TP4056 incarcare Acumulator Li-Ion
- Modul KY-040 Encoder
- Modul MT3608 ridicator de tensiune 3v3 → 5v
- Modul Adaptor Card SD
- Display OLED I2C 128×64
- Rezistenta 4.7k - 2 buc
- Rezistenta 1.0k - 2 buc
- Condensator 22nF
- Condensator 10nF
- Condensator electrolitic 330uF (low-ESR) - 2 buc

- Condensator electrolitic 47uF

Bateria trebuie incarcata folosind un IC special TP4056 pentru a putea fi utilizata. Tensiunea este ridicata la 5V folosind un regulator in comutatie de tip boost, folosind chip-ul MT3608. Un comutator este folosit pentru a porni/opri alimentarea la boxa.

Deoarece microcontroller-ul Atmega328p nu prezinta iesiri analogice, sunetul este generat folosind un semnal modulat PWM. Frecventa este de 65kHz pentru a putea include intreaga banda sonora (20Hz - 22kHz). Semnalul este trecut printr-un filtru trece jos pentru a pastra doar componenta continua. Alegem frecventa de taiere ~ 7 kHz si punem un filtru de ordin 2 pentru a obtine o rejectie mai buna a armonicelor. In serie cu acesta se adauga un condensator pentru a recentra semnalul in jurul 0V.

Simulare in Falstad unde se alterneaza duty-cycle-ul PWM pentru a genera un semnal analogic la iesire. Se poate observa ca filtrarea nu elimina zgomotul in totalitate insa difuzorul in sine functioneaza ca un filtru trece jos, deci rezultatul este acceptabil.

Software Design

Pentru proiect am propus realizarea intregului cod bare-metal, fara sa utilizez librarii externe. Codul este realizat in Microchip Studio si Python pentru incarcarea pieselor pe cardul SD.

Incepand cu interfata hardware, am programat comunicarea prin I2C (i2c.h) cu display-ul cu controller SSD1306 (ssd1306.h), si am implementat o functii pentru afisarea liniilor text. Am creat propriul font, cu caractere de 6×8 pixeli, pentru a putea afisa text si imagini pe ecran. Fontul este realizat in excel si tradus ulterior in octeti.

Pentru a stoca piesele, am programat o interfata SPI (spi.h), precum si instructiunile necesare comunicarii cu cardul SD (sd.h), conform SD Card Physical Specification. Fiecare fisier stocat are un header ce contine nr. total de esantioane si titlul piesei, urmat de esantioanele cantecului. Acestea sunt codificate la o rezolutie de 8 biti, 20k sample-uri pe secunda, pe un singur canal. Fiecare fisier este localizat la o locatie specifica in memorie, pentru a putea fi gasit usor.

Utilizatorul are nevoie de o metoda de a interactiona cu dispozitivul, iar aceasta este facilitata folosind encoderul KY040. Acesta are un buton ce se poate apasa si roti pentru a realiza 3 comenzi diferite. Directia de rotatie este determinata folosind intreruperi si un automat pe stari finite (encoder.h).

Pentru a genera sunetul, folosim 2 timere. Primul genereaza semnalul PWM si este setat la 65kHz. Al doilea provoaca executarea unei intreruperi ce modifica duty-cycle-ul pentru semnalul PWM (0 - 255).

Acesta este configurat sa ruleze la 20kHz. Cu aceste 2 timere putem genera un semnal sonor plecand de la o lista de esantioane.

O alta limitare a microcontroller-ului este ca desi acesta este dotat cu 2KB de memorie RAM, fisierele contin milioane de esantioane. Pentru a putea reda sunetul trebuie sa citim sunetul in timp ce il difuzam. Folosim 2 buffere de 512B. La fiecare moment de timp, intr-unul citim esantioanele de pe cardul SD si din celalalt extragem valoarea necesara setarii duty-cycle-ului pentru timer-ul PWM. Odata ce buffer-ul de citire devine gol, acesta este interschimbat cu buffer-ul de scrie si programul continua.

- (etapa 3) surse şi funcţii implementate

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Concluzii

Download

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

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele 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

- 10 Aprilie - Am comandat piesele necesare
- 16 Aprilie - Am realizat toate conexiunile electrice conform schemei
- 25 Aprilie - Am finalizat testarea pentru fiecare component (Display, Sunet, Card SD, Encoder)
- 3 Mai - Inceput etapa de documentare a proiectului

Bibliografie/Resurse

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

- Ghid programare card SD
- Datasheet controller display SSD1306
- Datasheet encoder KY-040

Export to PDF