

Animatronic Teddy Bear

Introducere

Proiectul reprezintă o jucărie interactivă ursuleț de pluș controlat printr-o serie de butoane, având drept funcționalități realizarea unui dans prin mișcarea alternativă a brațelor, salutul, atât prin mișcarea brațului (hand-wave) cât și sonor, opțiunea de a înregistra voce și a o reda ulterior, cu posibilitatea de a alege una din mai multe voci disponibile

Scopul proiectului este de a crea o jucărie entertaining ce poate fi folosită și ca audio recorder și de a da viață unui simplu ursuleț de plus. În acest fel, vreau să aduc la viață visul oricărui copil de a vorbi cu ursulețul său și de a-l vedea interacționând cu el.

Ideea principală de la care am pornit a fost să transform o jucărie simplă lipsită de interacțiune și mișcare în ceva care poate reacționa la dorințele utilizatorului punându-i la dispoziție o serie de comenzi pe care le poate da pentru a produce diferite mișcări.

Fiind un proiect cu importanță artistică, este un mod de a satisface copilul interior al oricărui adult și a-i aduce aminte de momentele în care își imagina că jucăriile lui prind viață.

Descriere generală



Elementul central este placa ce conține microcontrollerul (ATMega328P) și are același pinout design ca un Arduino UNO R3, care controlează servomotoarele și difuzorul, primește input de la microfon și de la butoane și comunică cu modulul SD pentru a stoca și citi sunetele înregistrate.

Servo S/D: Sunt două servomotoare conectate mecanic la un schelet ce va fi introdus în fiecare braț al ursulețului (S - brațul stâng, D - brațul drept)

Modul SD: cititor de card SD pe care vor fi stocate fișierele audio

Buton A - în modul playback, oprește playback-ul și trece în modul motion; în modul motion oprește mișcarea și trece în modul playback; în oricare mod, hold 1s oprește acțiunea și trece în modul record; în modul record oprește înregistrarea și trece înapoi în modul motion


Buton B - în modul playback are funcție de pause/resume și hold 1s trece la track-ul anterior; în modul motion are funcție de salut; în modul record nu face nimic

Buton C - în modul playback are funcție de switch voice și hold 1s trece la track-ul următor; în modul motion are funcție de dans; în modul record nu face nimic

Hardware Design

Listă de piese:

- Placă Plusivo compatibilă Arduino UNO R3
- 2 x servomotor
- modul microfon cu preamplificator MAX4466
- buzzer pasiv
- 3 x push button
- modul card microSD + card microSD
- 1 x rezistență 100 Ω

Schema electrică: 

[Schema electrică a modului de amplificare MAX4466:](#)



Descriere pini:

servo: D9, D10 - pentru ca au OC1A si OC1B si i-am folosit pentru a genera semnal PWM de la Timer/Counter1

microfon: A1 - pentru ca este analogic cu functie de ADC

modul card sd: SCK pe D6 pentru ca are Timer/Counter1; CS, MOSI, MISO pe D7, D5, D4 pentru ca impreuna cu D6 sunt in linie

difuzor: D2 - pin digital liber

butoane: D11, D12, D13 - pini digitali ramasi liberi

Software Design

Am folosit PlatformIO pentru a scrie codul aplicatiei.

Am incercat multiple solutii de biblioteci pentru comunicarea cu cardul SD si lucrul cu fisiere wav, dar pana la urma urmatoarea varianta este cea care a mers:

- biblioteca SdFat pentru comunicare prin SPI cu SD-ul. PetitFAT32 folosita la laborator are doar un subset din functionalitati si aveam nevoie de functionalitate de a creare si append intr-un fisier.

- in biblioteca TRMpcm, exista functii pentru inregistrarea fisierelor wav si redarea lor. Din pacate, aceasta biblioteca nu-mi lasa posibilitatea sa inregistrez folosind Timer1 si sa redau folosind Timer0 sau Timer2.

In cele din urma, am implementat eu functia de recording urmand functia din aceasta biblioteca, si functia de play am adaptat-o dupa cea de la laborator. Astfel, pentru inregistrare am folosit ADC si pentru redare am folosit PWM

- pentru managementul butoanelor am folosit PCINT2

- am o masina de stari care controleaza toate efectele pe care le au butoanele si activeaza/dezactiveaza timerele potrivite in situatie.

Rezultate Obținute

Rezultate se pot vedea [aici](#)

Concluzii

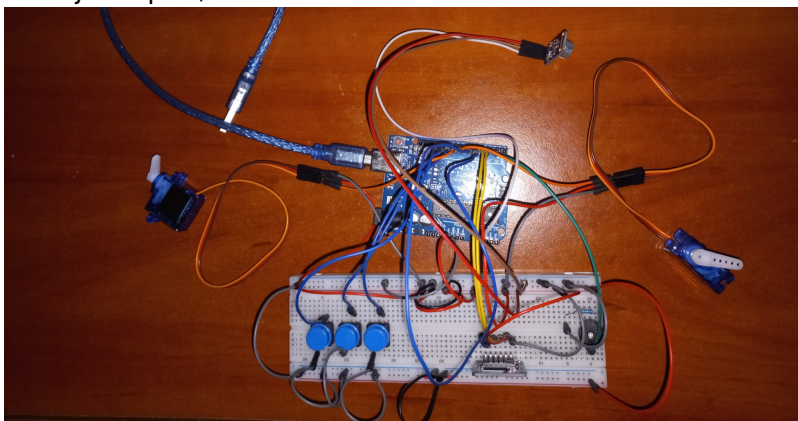
Download

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

Milestone hard:
cablaj complet, verificat functionare servomotoare



<https://youtube.com/shorts/7pH7bME-TYw?feature=share>

Bibliografie/Resurse

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

Resurse hardware: datasheet modul MAX4466: [datasheet](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/azamfir/stefan.sterea>



Last update: **2024/05/27 17:51**