

Air Drums

Autor: Tășcan Diana Ioana
Grupa: 332CB

Introducere

Proiectul are ca scop simularea cântatului la tobe prin mișcarea unor bețe de tobe în aer. Când bețele sunt mișcate, un difuzor va reproduce sunete de tobe diferite.

Descriere generală

Doi accelerometri ADXL345 sunt puși pe două linguri de lemn și aceștia vor detecta mișcarea lor în aer, vor trimite datele la placa Arduino, iar placa Arduino trimite sunetele la un difuzor 2W - 8ohm ce le va reproduce în funcție de bățul mișcat. Fiecare băț are implementat câte două sunete diferite care vor fi reproduse la întâmplare.

Schema Bloc



Hardware Design

- Cei doi accelerometri sunt conectați la un breadboard astfel: GND la GND, VCC și CS la VCC, SCL la A5 și SDA la A4. Datorită I2C, pentru a avea adrese diferite, un accelerometru va conecta SDO la VCC, iar celălalt va conecta SDO la GND.
- Pentru difuzor, pentru a controla curentul ce circulă la el, este folosit un tranzistor. Acesta are baza conectată la pinul 11, emitorul la GND, iar colectorul la difuzor. În cealaltă parte, difuzorul este conectat la VCC.

Schema electrica



Listă de componente

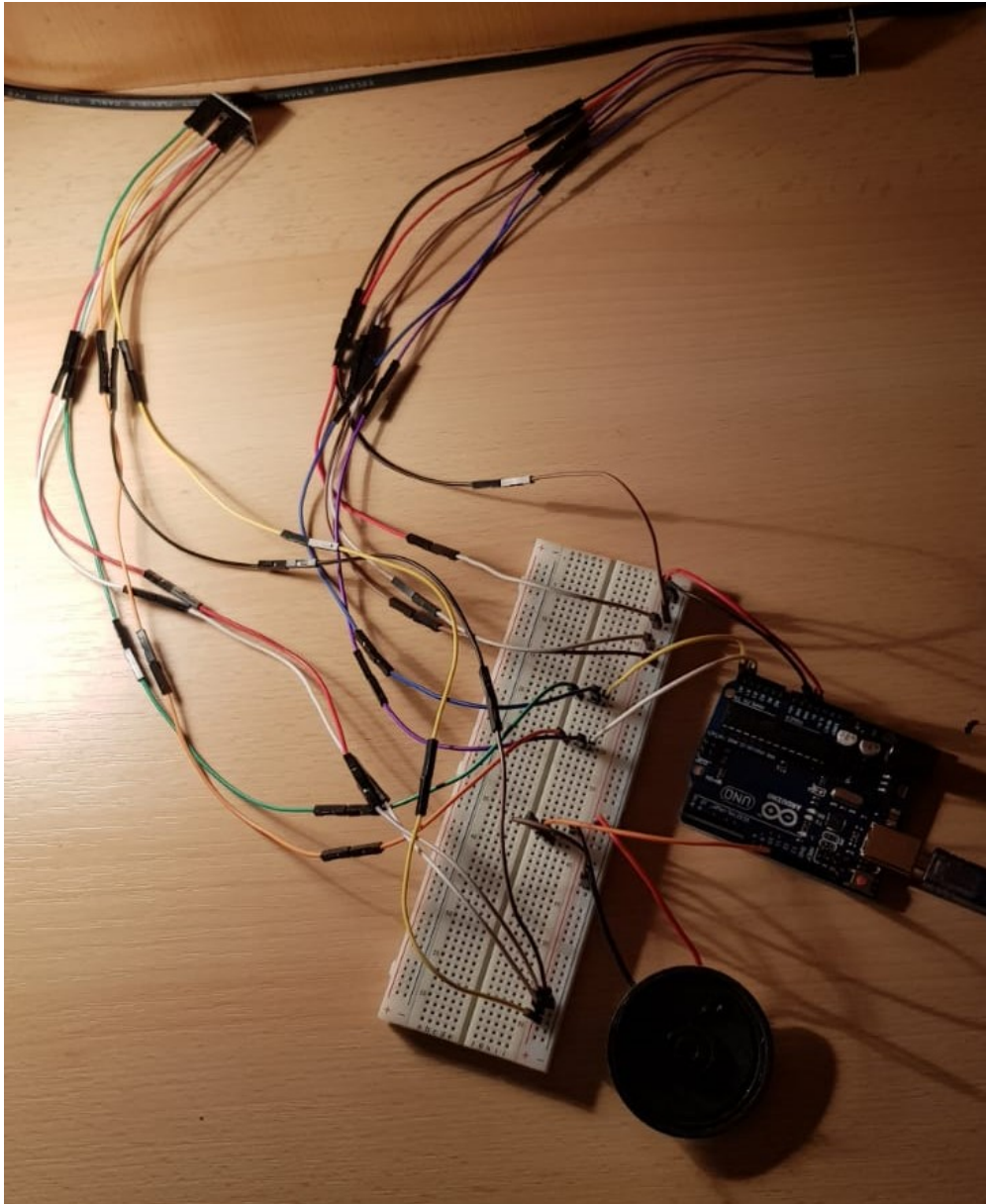
- [Arduino Uno](#)
- Breadboard
- [2x Accelerometri ADXL345](#)
- Bețe/Obiecte ce pot substitui bețe de tobe
- [Difuzor Arduino](#)
- [Tranzistor BC547BBK](#)
- Fire

Software Design

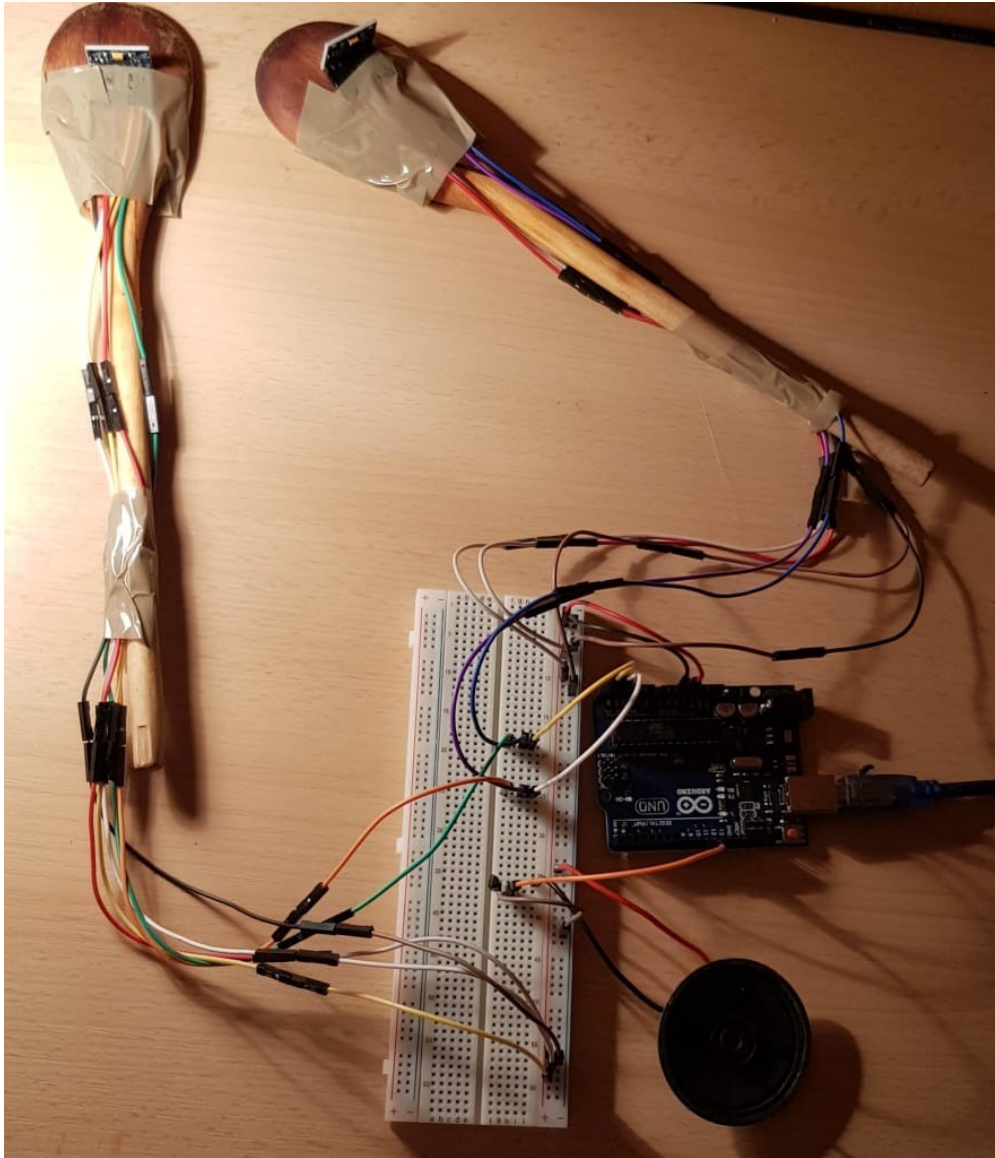
- Tot codul pentru placa Arduino se află în fișierul `airdrums.ino`, iar encodarea sunetelor se află în fișierul `sounds.h`.
- Pentru redarea sunetelor, am folosit biblioteca `PCM.h`.
- Pentru a ști pe ce adrese sunt accelerometrii, am folosit un program `I2C scanner`. După ce am găsit adresele, le-am salvat în 2 variabile globale. Pentru a detecta dacă un băț a fost mișcat, folosesc variabile globale pentru coordonatele lor în spațiu. O variabilă `bool` este folosită pentru a spune dacă bățul a fost mișcat destul de repede pentru a mima lovitura tobei.
- Pentru a obține coordonatele bățului, se citește de la accelerometrii în funcția `getCoords`. Pentru că se citesc doar câte 8 biți, dar valorile necesare sunt pe 16 biți, se face o shiftare a primilor 8 biți și se adaugă următorii 8. În funcție de care accelerometru este mișcat, se reactualizează coordonatele respective lui.
- Pentru a vedea dacă trebuie să se redea un sunet, în funcția `checkToPlay` se calculează distanța dintre coordonatele anterioare și noile coordonate, în funcție de care accelerometru a detectat mișcarea. Dacă mișcarea este detectată, se alege la întâmplare unul din două sunete encodate pentru acel accelerometru.
- În funcția `setup()`, se pornesc accelerometrele, iar pentru a schimba sensibilitatea lor, la `0x31` se scrie `3`, care face sensibilitatea mai mică. Se apelează `getCoords` pentru a lua coordonatele inițiate.
- În funcția `loop()` doar se verifică dacă trebuie să se redea un sunet pe difuzor.
- Pentru encodarea sunetelor, am găsit un pachet de sunete online care avea câteva sunete de tobe. Am luat pachetul și am folosit `Audacity` pentru a converti la un sunet `PCM`, apoi am folosit un program `EncodeAudio` pentru a obține un vector ce reprezintă encodarea frecvenței sunetului. Fiecare băț are doar câte două sunete, deoarece ocupă mult spațiu și nu am putut adăuga astfel alte sunete.

Rezultate Obținute

Modul fără bețe



Modul cu bețe



Video

[Short Demo Video](#)

Concluzii

- A fost destul de dificil să găsesc o metodă de encodare a sunetelor și dificil să aleg sunete care să fie scurte și reprezentative pentru tobe.
- A dat rateuri de câteva ori fără vreo explicație, a trebuit să închid și să redeschid aplicația ca să meargă...
- Un proiect plăcut, dar sunt dezamăgită că nu am putut adăuga mai multe sunete din cauza memoriei.

Bibliografie/Resurse

I2C Scanner

Tutorial convertire sunete

Pachetul de sunete

Documentație accelerometri

Air Drums

Document PDF

Codul și folder cu sunetele reproduse

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/air-drums>



Last update: **2021/05/27 18:09**