

Music With Arduino Accelerometer

Autor

Chiroiu-Jitianu Doina

Introducere

Proiectul constă în realizarea unui instrument muzical, realizat cu Arduino UNO, prin intermediul unui accelerometru cu giroscop pentru interacționare și interpretarea mișcării.

Rolul proiectului este familiarizarea atât cu Arduino, cât și cu interfațarea senzorilor de tip MPU6050.

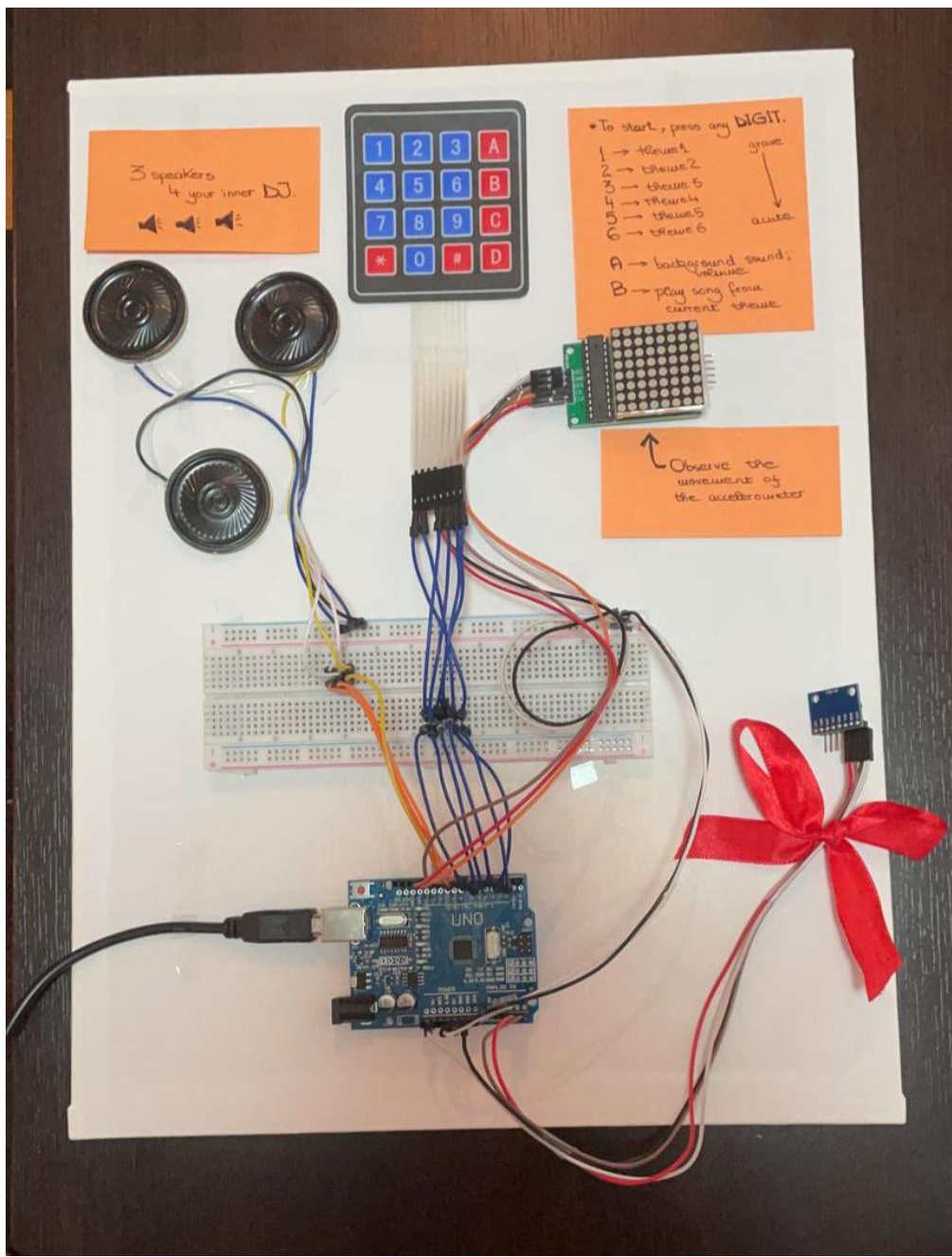
Descriere generala

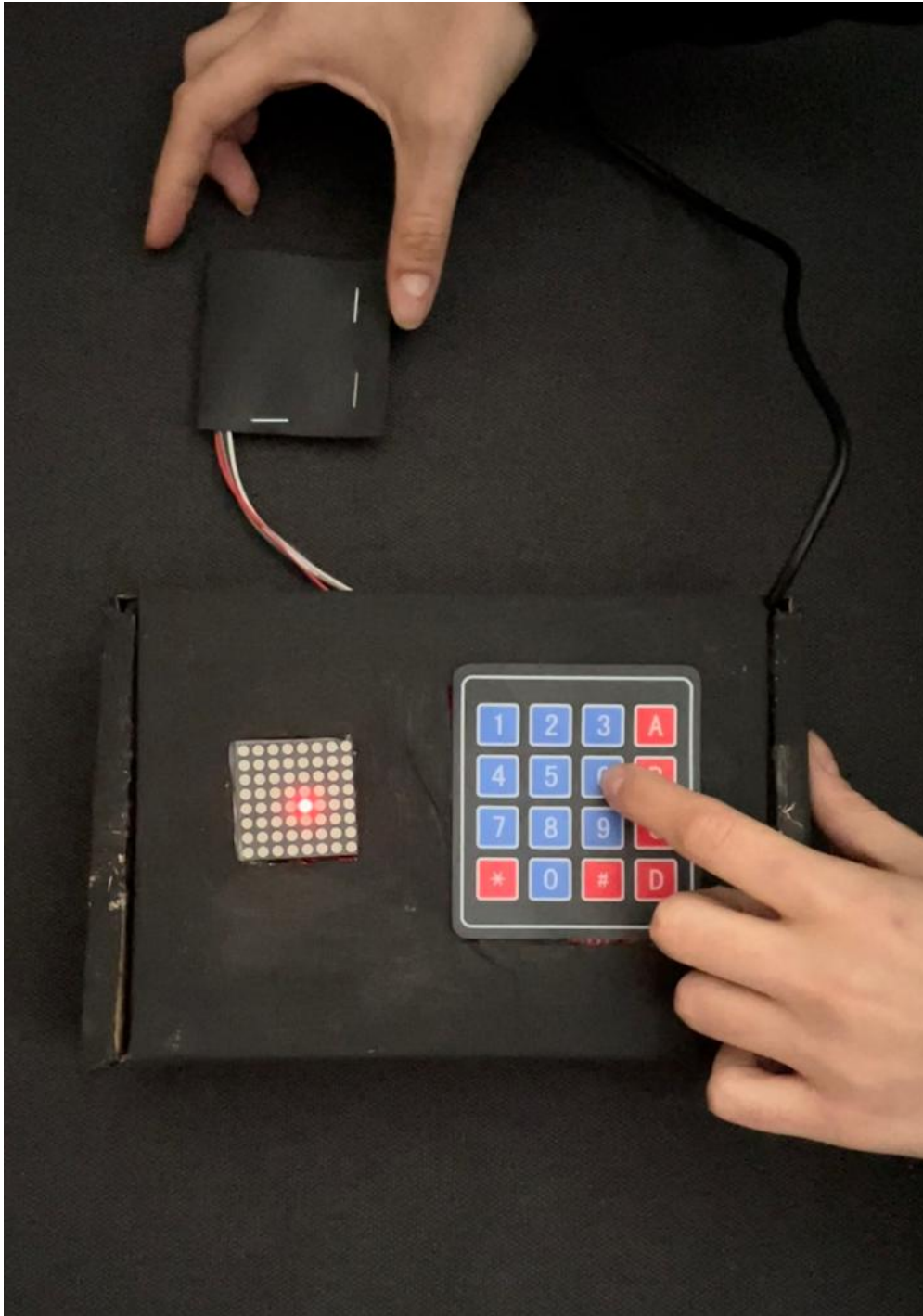
Utilizatorul va mișca accelerometrul, iar prin intermediul senzorilor săi, datele caracteristice mișcării (acelerația și rotația) vor fi transmise pe seriale și interpretate de către program. Totodată, utilizatorul poate interacționa cu aplicația prin intermediul unei tastaturi, pe baza căreia poate alege o temă muzicală (gamă de note) sau poate modifica volumul.

Odată ce sunt interpretate datele, în funcție de gama aleasă, se vor reprezenta sunetele corespunzătoare mișcării efectuate prin difuzoarele atașate.

Schema bloc



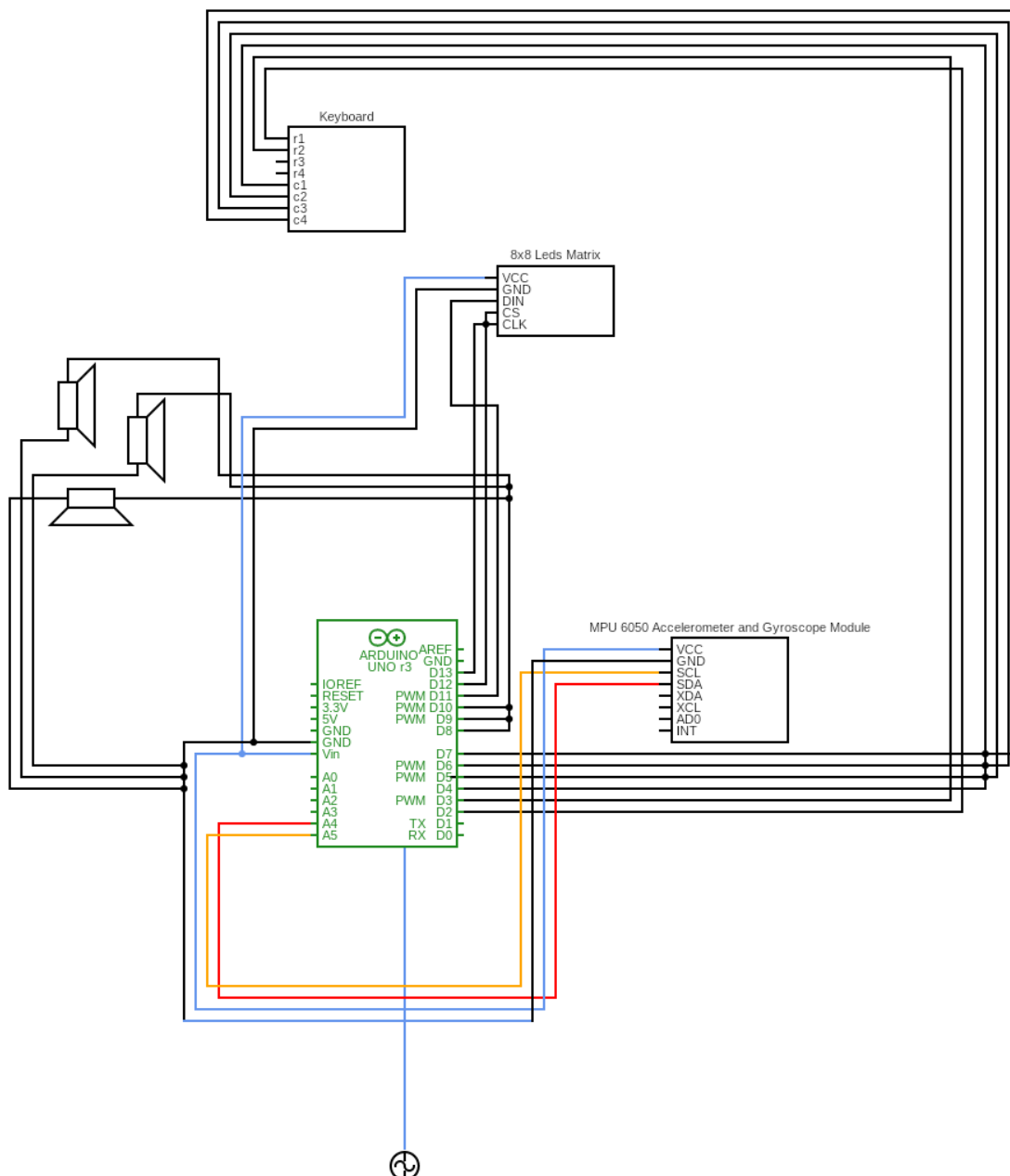




Hardware Design

- Arduino UNO
- MPU6500 Accelerometer & Gyroscope Module
- 8x8 Leds Matrix
- Keyboard
- Speakers
- Breadboard
- Fire de legatura

Schema electrica



Detalii hardware:

- modul MPU-6050

Pentru conectarea accelerometrului, am conectat VCC-ul la 5V, GND la GND, și cele două seriale de date astfel: SCL la A5, SDA la A4. Astfel, se realizează comunicația I2C necesară.

- 8x8 Leds Matrix

În vederea folosirii unui obiect ledController, a trebuit sa conectez pinii DIN, CLK și DS. Am ales pentru acest element pinii D11, D13 si D12.

- Keypad

Se conecteaza pinii corespunzatori liniilor 1 și 2 (pinii D2 și D3) și pinii pentru cele 4 coloane (pinii D4,

D5, D6 și D7). Astfel, se va vor mapa acești pini pe vectorul de keys ($\{\{1, 2, 3, A\}, \{4, 5, 6, B\}\}$), pe care îl utilizăm în crearea obiectului intermediar keypad.

- Speakers

Fiecare speaker este conectat la GND cu un fir, și cu celălalt la un pin aferent. Am folosit pinii 8, 9 și 10 pentru cele 3 speakere.

Software Design

Proiectul meu constă, la nivel de cod, în implementarea programului ce va fi încărcat în placa (folosind Arduino IDE) și în implementarea programului ce procesează datele share-uite de către Arduino prin terminalul serial (folosind Processing IDE).

În ceea ce privește bibliotecile necesare, am folosit:

- ItemI2Cdev.h
- ItemMPU6050.h
- LedControl.h
- Keypad.h
- Wire.h

Structurile auxiliare sunt, în mare parte, structuri standard C. Pe lângă acestea, am utilizat totuși o structură Grid, necesară compunerii obiectului LedController ce servește afișării pe matricea de leduri, și un obiect Keypad pentru abstractizarea logicii tastaturii folosite.

Detalii de implementare:

note & game muzicale

Am folosit o varietate de note muzicale, din game diferite, pe care le-am grupat câte 10 în 6 grupuri stocate în vectorul "melodies". Astfel, avem 6 teme muzicale disponibile.

setup()

Pentru a realiza conexiunea cu senzorul MPU-6050, solicităm adresa corespunzătoare (0x68). După ce am preluăm adresa conexiunii de tip I2C, vom trezi senzorul setând 0 pe acea adresă. Pentru calibrare, am setat flaguri pe cele 3 adrese unde preluăm datele pentru cele 3 axe (X, Y, Z).

loop()

După cererea de receptionare de la adresa MPU, începe transmisia de la acea adresa pentru a prelua măsurile de accelerație, rotație și timp. Fiindu-mi necesare aceste mărimi, folosesc în total 14 registre (2 pentru măsurarea timpului, câte 2 pentru datele de pe fiecare axă - deci în total 6 pentru accelerație și două pentru rotație).

Stocăm datele în variabile ținute global în program, și începem procesarea acestora pentru a le putea reprezenta pe ieșirile noastre.

- Redarea sunetelor

Am procesat valorile măsurate ale accelerației: AcX, AcY, AcZ astfel: Calculez offset-ul obținut pe fiecare axa, și mapăm acele valori în range-ul [0, 8] pentru a folosi rezultatul pentru afișarea pe matricea de leduri.

Am procesat valorile măsurate ale rotației: GyX, GyY, GyZ astfel: Calculăm unghiurile de rotație: roll - axa X, pitch - axa Y, rbe - axa Z. Formulele utilizate sunt:

```
roll = atan(GyY / sqrt(pow(GyX, 2) + pow(GyZ, 2))) * 180 / PI; // rotation X-axis
pitch = atan(-1 * GyX / sqrt(pow(GyY, 2) + pow(GyZ, 2))) * 180 / PI; // rotation Y-axis
rbe = atan(GyZ / sqrt(pow(GyX, 2) + pow(GyY, 2))) * 180 / PI; // rotation Z-axis
```

Datele roll, pitch și rbe sunt puse terminalul serial.

Pentru a găsi sunetul corespondent din gama curentă, vom mapa aceste date la range-ul [0, 10], rezultatul reprezentând în final indexul din vectorul de sunete → melodies[theme][index]. Acest sunet va fi invocat pe difuzoare.

- Leds Matrix

După maparea datelor în range-ul [0, 8] (dimensiunea matricii), se va seta ledul corespunzător acestei mapari, prin obiectul ledController. Astfel, valorile recepționate de la accelerometrul cu giroscop se pot pune în corespondență cu aprinderea unui led în spațiul matricii. Obținem deci, o reprezentare vizuală a mișcării accelerometrului.

- Keypad

Pe lângă interacțiunea prin accelerometru și interpretarea mișcării acestuia, instrumentul dispune de interacțiune printr-o tastatură. Obiectul keypad este intermediarul în cod pentru citirea unei taste. Lucrăm cu primele două linii de taste, incarcand semnăturile lor în vectorul keys[.].

La apăsarea unei taste cu semnătura unei cifre de la 1 la 6, se va seta o tema muzicala din cele 6 disponibile și va porni instrumentul. La apăsarea tastei 'A', voi trezi încă un difuzor dintre cele 3 disponibile pentru un plus de volum. Când se apasă 'A' a treia oară, volumul revine la stadiul inițial. La apăsarea tastei 'B', voi invoca notele din gama curentă la rand, pentru a reprezenta la nivel de sunet o melodie simplă.

Legat de procesarea datelor în cadrul PC-ului, am implementat un program ce realizează design-ul accelerometrului (în mod simplist), și îi stimulează rotația cu datele preluate de pe portul de comunicație.

Rezultatul Final

- [cod github](#)
- [video](#)

Concluzie

Proiectul m-a învățat foarte multe despre lumea Arduino și posibilitățile nenumărate pe care le avem în implementarea prin plăcute. Una dintre provocările majore a fost realizarea conexiunii I2C, între plăcuta Arduino UNO și modulul de accelerometru cu giroscop MPU-6050. Pe lângă studiul în ceea ce privește legarea serialelor de date, am implementat în totalitate sistemul de recepționare a datelor.

Ulterior, o alta etapa importanta a fost procesarea datelor cu formule matematice, pentru obținerea unor rezultate relevante: sunete și afișare în concordanță cu mișcarea efectuată.

Acest proiect m-a familiarizat cu programarea plăcuțelor Arduino, dar și cu utilizarea perifericelor (tastatura, afișarea prin matrice de leduri) și cu modul de funcționare a unui senzor MPU.

Mi-a plăcut să lucrez la proiect și am învățat câte probleme poate pune hardware-ul într-o astfel de aplicație, învățând să mă adaptez.

Download

[music-with-arduino-accelerometer-master.zip](#)

Bibliografie / Resurse

<https://www.instructables.com/Accelerometer-MPU-6050-Communication-With-AVR-MCU/>

<https://www.electronicwings.com/arduino/mpu6050-interfacing-with-arduino-uno>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/music-with-arduino-accelerometer> 

Last update: **2021/06/01 18:34**