

Gesture controlled music player

Introducere

- Proiectul oferă posibilitatea utilizatorului de a **controla volumul și melodia curentă** prin intermediul unui set de gesturi predefinite.
- Scopul proiectului este de a oferi o alternativă mai interactivă a acestor acțiuni.
- Ideea a venit ca urmare a faptului că ascult muzică constant și folosirea unor comenzi de acest tip ar fi facilă.
- Proiectul ar fi util în situații asemănătoare sau în cazul calculatoarelor care nu dispun de dispozitive periferice.

Descriere generală



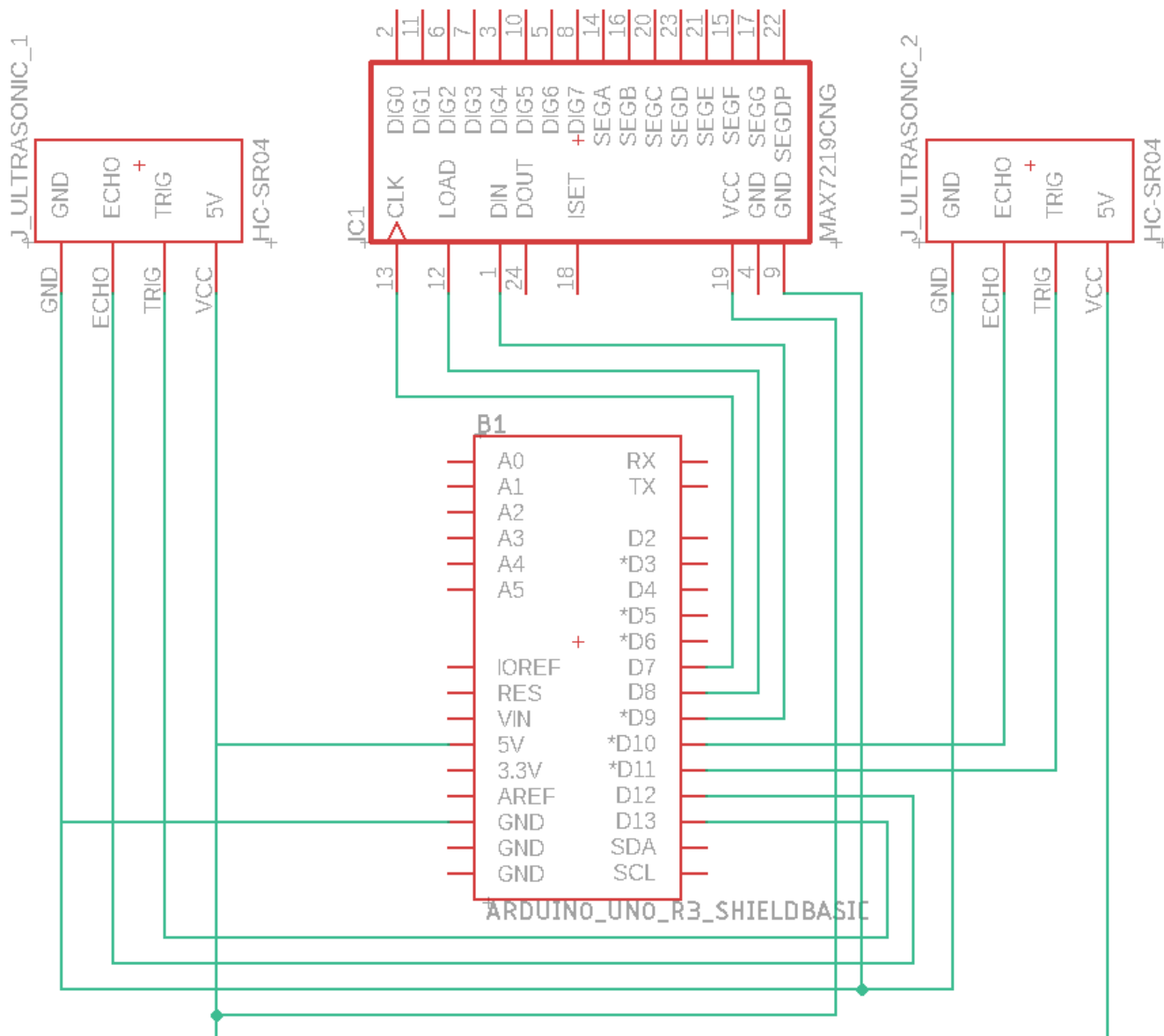
1. Se inițiază comunicarea cu Python prin intermediul portului ales.
2. 2 **notificări** de tip toast vor apărea pe ecranul calculatorului odată ce aceasta s-a realizat.
3. Pentru a porni sistemul de recepționare a gesturilor, utilizatorul trebuie să țină ambele mâini în fața celor doi senzori.
4. Odată pornit, unul dintre următoarele gesturi poate fi ales:
 1. ambele mâini în fața senzorilor ⇒ **play/pause** în funcție de stadiul melodiei curente
 2. numai o mână în fața sensorului din stânga:
 1. pentru mai puțin de 0,5 secunde ⇒ **swipe left** (se trece la melodia anterioară din playlist)
 2. pentru mai mult de 0,5 secunde ⇒ **volume down**
 3. numai o mână în fața sensorului din dreapta:
 1. pentru mai puțin de 0,5 secunde ⇒ **swipe right** (se trece la melodia următoare din playlist)
 2. pentru mai mult de 0,5 secunde ⇒ **volume up**
5. Gesturile vor fi recepționate numai după apariția imaginii de clepsidră pe matricea de LED-uri.
6. Pentru fiecare gest se va afișa un **icon sugestiv pe matricea de LED-uri**.
7. Fiecare gest va fi transmis mai departe programului Python care va realiza acțiunea prin intermediul bibliotecii PyAutoGUI.

Hardware Design

Listă de componente

1. Arduino UNO
2. Mini breadbord
3. 2 Senzori ultrasonici HC-SR04
4. Modul cu matrice de LED-uri MAX7219
5. Multe fire
6. Suport

Schemă electrică



Software Design

1. Mediu de dezvoltare:

1. PyCharm
2. Arduino IDE

2. Librării și surse 3rd-party:

1. Python:

- **time** pentru funcția sleep
- **serial** pentru comunicația serială
- **pyautogui** pentru realizarea acțiunilor menționate prin automatizare GUI
- **win10toast** pentru notificările de tip toast

2. Arduino:

- **LedControl.h** pentru lucrul cu matricea de LED-uri
- **LibPrintf.h** pentru funcția printf

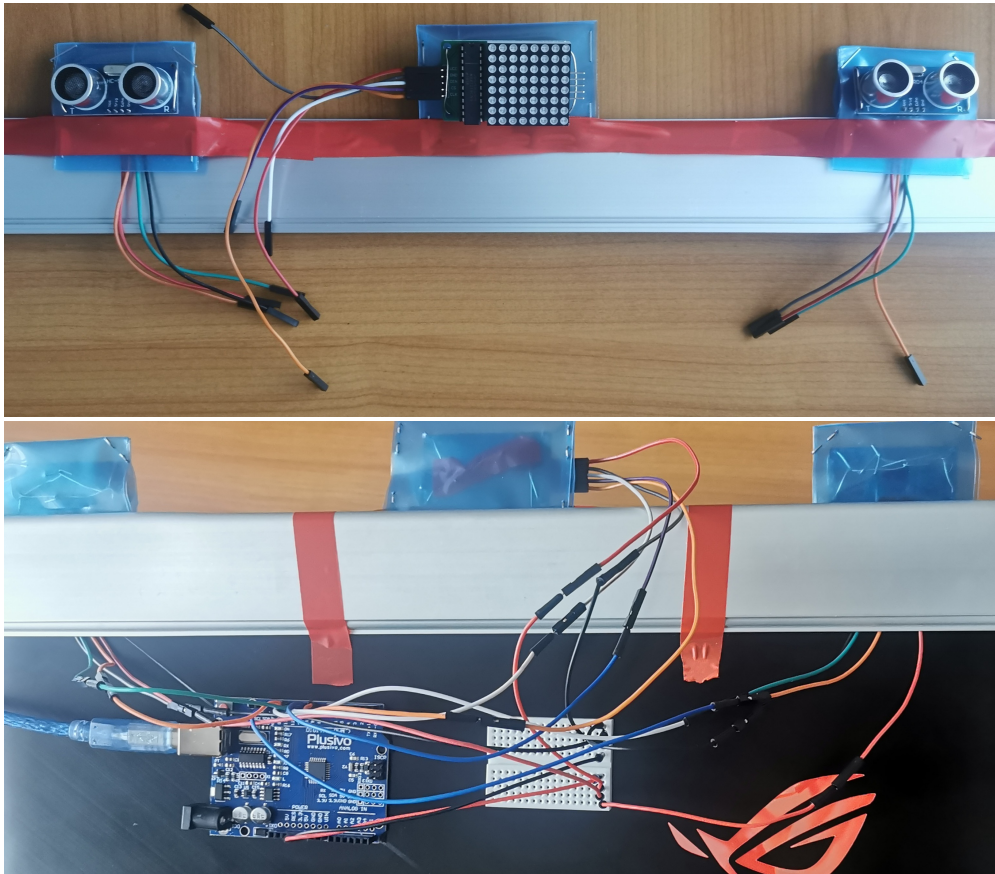
3. Fișiere sursă:

1. Python: main.py
2. Arduino: main.ino

Rezultate obținute

- S-a obținut o **acuratețe destul de crescută** a recunoașterii gesturilor.
- Sensorii de distanță funcționează foarte bine în majoritatea cazurilor, cu excepția unor cazuri când la distanță foarte mică generează valori foarte mari eronate.
- Pentru a include mai multe acțiuni/gesturi, ar trebui mărit numărul de senzori folosiți.

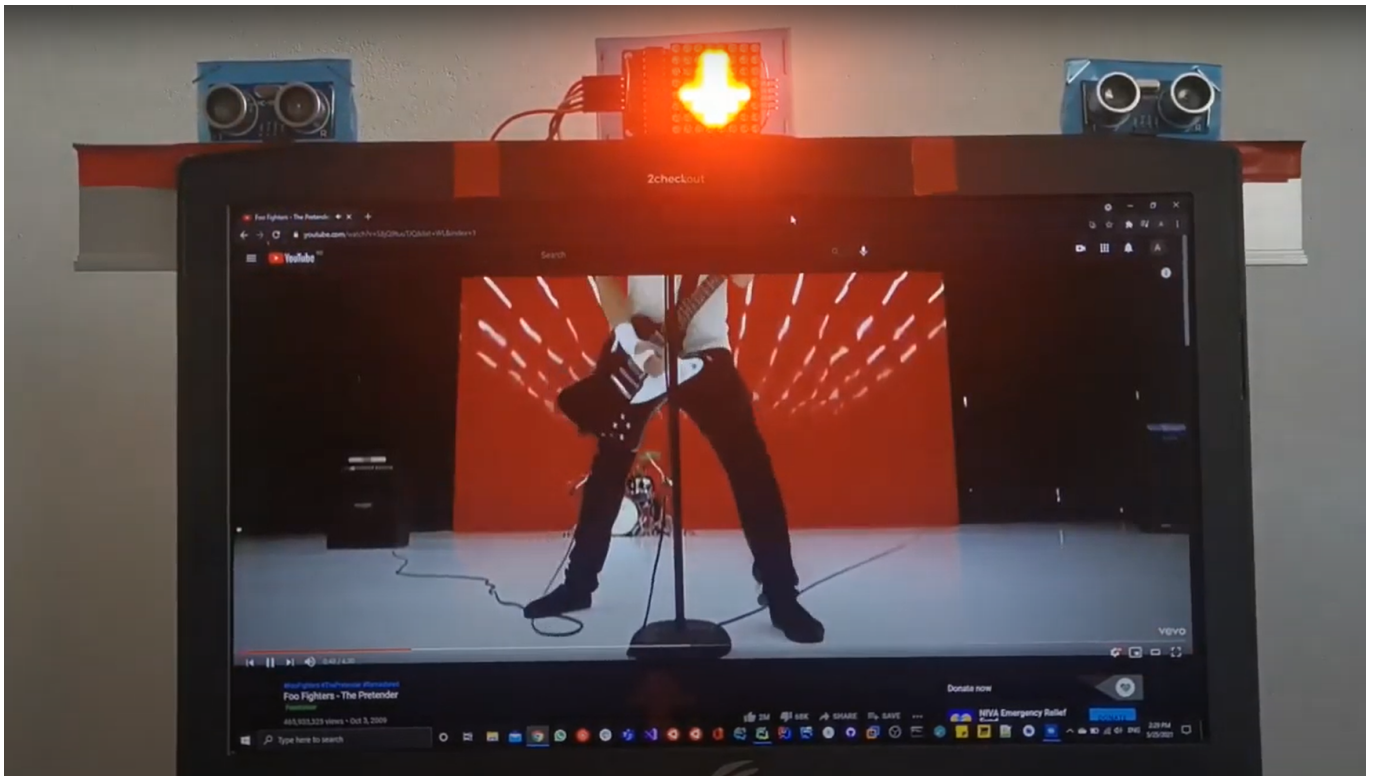
Poze proiect



Demo

Last update:

2021/05/26 pm:prj2021:apredescu:gesture_controlled_music http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/gesture_controlled_music
11:34



Concluzii

- A fost foarte interesant să văd cum Arduino poate fi conectat cu Python.
- Prin Python putem folosi algoritmi, structuri de date și librării mai complexe cu care putem prelucra datele capturate de senzori.
- Per total, un proiect foarte util prin care am putut să facem ceva și fizic.

Download

[Arhivă cod sursă](#)

[Export to PDF](#)

Bibliografie/Resurse

- [Link proiect](#)
- [Arduino based gesture control of your computer](#)
- [Controlling an LED matrix](#)
- [Windows 10 toast notifications](#)
- [Documentation for PyAutoGUI](#)
- **Google**

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/gesture_controlled_music



Last update: **2021/05/26 11:34**

