

# Arcade Pushing Machine

## Introducere

Proiectul constă într-un joc pe Arduino care măsoară forța cu care utilizatorul poate să apese pe un senzor de forță rezistiv și afișează scorul pe un ecran LCD, acompaniat de animații și sunete pe un difuzor. Principalul obiectiv al acestui proiect este să ofere o experiență distractivă și plăcută utilizatorului, în orice fel de context social.

Ideea de la care am pornit este chiar jocul foarte popular de la orice parc de distracții în care poți obține un scor cât mai mare, cu cât lovești mai tare cu pumnul.

## Descriere generală

Proiectul are următorul flow:

- La pornirea aparatului, se afișează pe display mesajul *Ești gata?*, iar led-ul roșu stă aprins
- Atunci când senzorul de mișcare detectează apariția unei persoane în fața aparatului, display-ul și difuzorul vor transmite mesajul *Start!*, se aprinde led-ul verde, iar senzorul de forță e pregătit și așteaptă o lovitură
- La momentul atingerii senzorului de forță, scorul obținut va fi afișat incremental pe display, alături de un mesaj audio corespunzător
- Timp de câteva secunde, led-ul roșu indică că nu se poate reîncerca lovitura. La momentul aprinderii celui verde, se detectează din nou mișcarea și jocul se poate relua.



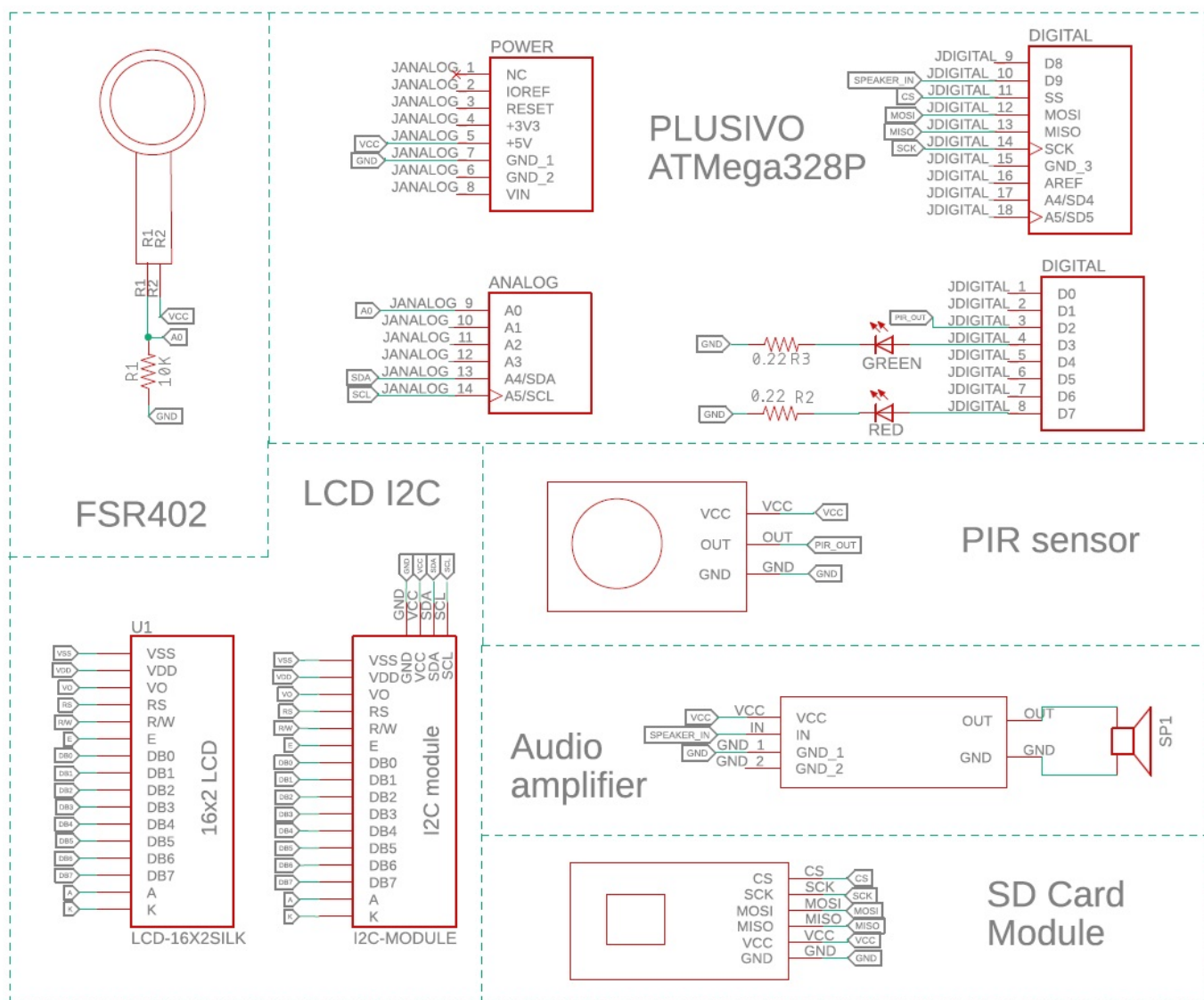
## Hardware Design

### Componente hardware utilizate:

- Plusivo ATmega328P
- modul microSD cu SPI
- card microSD 16GB
- difuzor
- modul amplificator audio LM386
- ecran LCD cu I2C
- senzor de forță rezistiv FSR402

- modul senzor de mișcare PIR
- un led verde și unul roșu
- rezistențe (220 și 10k)
- breadboard
- fire mamă-tată și tată-tată

## Schema electrică



**Ecranul LCD cu interfață I2C** are rolul de a afișa mesaje interactive pe parcursul jocului, dar și valorile măsurate de senzorul de forță rezistiv. Pinul SDA e conectat la A5 de pe placa Arduino, pentru a transmite datele, iar pentru semnalul de ceas am conectat pinul SCL la A4.

**Senzorul de forță rezistiv (FSR402)** măsoară forța aplicată prin creșterea rezistenței în momentul apăsării pe acesta, trimițând rezultatele pe ecranul LCD. Este înseriat cu o rezistență de 10kΩ și apoi conectat la pinul A0 al plăcuței Arduino pentru citirea valorii analogice.

**Senzorul PIR de mișcare** detectează mișcarea în fața aparatului și trimite un semnal către Arduino,

pentru a putea aprinde LED-urile corespunzătoare. Pinul OUT e conectat la pinul D2, pentru a se putea folosi de întreruperi.

**Modulul de card SD** este conectat la plăcuță prin pinii interfeței SPI: CS (D10 - SS), MOSI (D11), MISO (D12) și SCK (D13), utilizând un card micro-SD de 16GB de la Kingston, pe care am stocat fișiere audio .wav cu următoarele specificații: 16kHz, 8-bit resolution (128kbps bit rate), canal MONO, U8 PCM, pentru a putea fi redade.

**Modulul LM386** ajută la amplificarea semnalului audio, până la un nivel care poate fi auzit clar prin difuzorul de 8Ω. Amplificatorul este conectat la pinul D9.

## Software Design

### Mediu de dezvoltare

Codul sursă a fost implementat în **Arduino IDE 2.3.2**.

### Biblioteci utilizate

- **Wire.h**: este necesară pentru interacțiunea cu LCD-ul I2C
- **LiquidCrystal\_I2C.h**: utilizată pentru facilitarea afișajelor mesajelor pe display
- **TMRpcm.h**: este folosită pentru redarea pe difuzor a fișierelor audio WAV
- **SD.h**: biblioteca e necesară pentru a citi datele stocate pe cardul SD
- **SPI.h**: utilizată pentru a comunica cu cardul SD

### Funcționalități din laborator prezente în implementare

1. **I2C (Inter-Integrated Circuit)**: Protocol utilizat pentru comunicarea cu afișajul LCD. Permite conectarea ușoară a dispozitivului, folosind un număr minim de pini
2. **SPI (Serial Peripheral Interface)**: Necesari pentru comunicarea cu cardul SD, utilizând o viteză mare de transfer de date
3. **Întreruperi**: Utilizate pentru a detecta mișcarea folosind senzorul PIR, permițând o reacție rapidă la evenimentele externe, fără a bloca execuția principală a programului


### Funcțiile implementate

- **setup()**: Inițializează pinii, bibliotecile și componentele hardware (LCD, card SD, difuzor). Așteaptă calibrarea senzorului PIR pentru 15 secunde, apoi redă un sunet ce semnalizează ca jocul poate începe

- **restart():** Resetează variabilele și pregătește sistemul pentru o nouă sesiune de joc
- **loop():** Menține led-urile aprinse și mesajele de pe display afișate pe tot parcursul jocului. Monitorizează continuu starea senzorului PIR, iar atunci când parametrul *motion\_status* are valoarea *HIGH* pentru prima oară de la începerea sesiunii de joc, semnalizează faptul că jucătorul poate să înceapă apăsarea pe senzorul de forță. După ce s-a indicat acest lucru, programul monitorizează starea senzorului FSR prin citirea analogică de pe pinul A0, iar atunci când detectează o apăsare validă, se afișează un mesaj și se redă un sunet corespunzător. Sunt permise apoi până la încă 4 sesiuni de joc
- **play\_sound(char \*message):** Redă un fișier audio de pe cardul SD, în funcție de parametrul *message*
- **display\_message(char \*message):** Afișează un mesaj pe LCD, în funcție de parametrul *message*
- **classify\_score(int score):** Clasifică scorul obținut de la senzorul FSR, setând mesajul corespunzător. Un scor slab este considerat unul sub 550, scorul mediocru se încadrează între 550 și 820 de puncte, scorul bun este cel între 820 și 930, iar tot ce este peste 930 este considerat foarte bun
- **show\_score(int score):** Afișează scorul incremental pe LCD
- **pir\_interrupt():** Funcția de întrerupere care setează starea de mișcare detectată de senzorul PIR, abia după ce senzorul și-a terminat calibrarea

## Rezultate Obținute



## Concluzii

Acest proiect a fost o oportunitate foarte bună de a pune în practică cunoștințele acumulate la PM, dar și de la ED. Am folosit și Fusion încă o dată în viață, pentru realizarea schemei electrice. A fost o provocare inginerească, care m-a ajutat să înțeleg mai bine interacțiunea dintre hardware și software. Am experimentat utilizarea întreruperilor, a senzorilor, afișarea pe LCD, redarea fișierelor audio și, mai ales, am înțeles cu adevărat cum se folosește un Arduino. Proiectul a ieșit exact așa cum mi-am dorit, captivant, distractiv și fără bug-uri. 

## Download

Arhiva cu implementarea și schema electrică se poate descărca de [aici](#).

# Jurnal

- 7 aprilie 2024 - Mi-am pus la punct ideea proiectului
- 9 aprilie 2024 - Am împrumutat un Plusivo și niste fire 
- 27 aprilie 2024 - Am realizat Wiki-ul pe ocw
- 28 aprilie 2024 - Am comandat piesele hardware necesare rămase
- 10 mai 2024 - Au venit piesele și m-am apucat de montat piesele pe rând, pentru a le testa funcționalitatea
- 11 mai 2024 - Am creat un proiect funcțional!
- 14 mai 2024 - Am început să fac schema în Fusion
- 17 mai 2024 - Am lipit difuzorul și senzorul de forță
- 17 mai 2024 - Am cumpărat un modul de amplificare audio, deoarece nu funcționa difuzorul fără
- 17 mai 2024 - Am făcut montajul final
- 22 mai 2024 - Am rupt din greșeală firele difuzorului și acum trebuie să lipesc din nou 
- 22 mai 2024 - Am făcut ultimele modificări la schematic
- 22 mai 2024 - Am proiectat carcasa 3D
- 23 mai 2024 - Am finalizat montajul
- 24 mai 2024 - Am finalizat partea Software
- 25 & 26 mai 2024 - Update-uri pe wiki

## Bibliografie/Resurse

### Resurse Hardware

- [Arduino Uno Pins - A Complete Practical Guide](#)
- [How to Use I2C LCD with Arduino | Very Easy Arduino LCD I2C Tutorial | Arduino 16x2 LCD I2C Tutorial](#)
- [3 options for Playing Audio on Arduino](#)
- [How to Use a PIR Sensor with Arduino: Wiring & Code for Motion-activated Projects](#)
- [How to use Force Sensitive Resistors \(FSR\) with Arduino - Simple Force Sensing Arduino Tutorial](#)
- [Proiectare și asamblare carcasă](#)
- [Optimus Digital](#)
- [ArduShop](#)

### Resurse Software

- [Voice Generator - Text to Speech](#)
- [Convert audio to WAV](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/diana.cismaru>



Last update: **2024/05/27 13:12**