


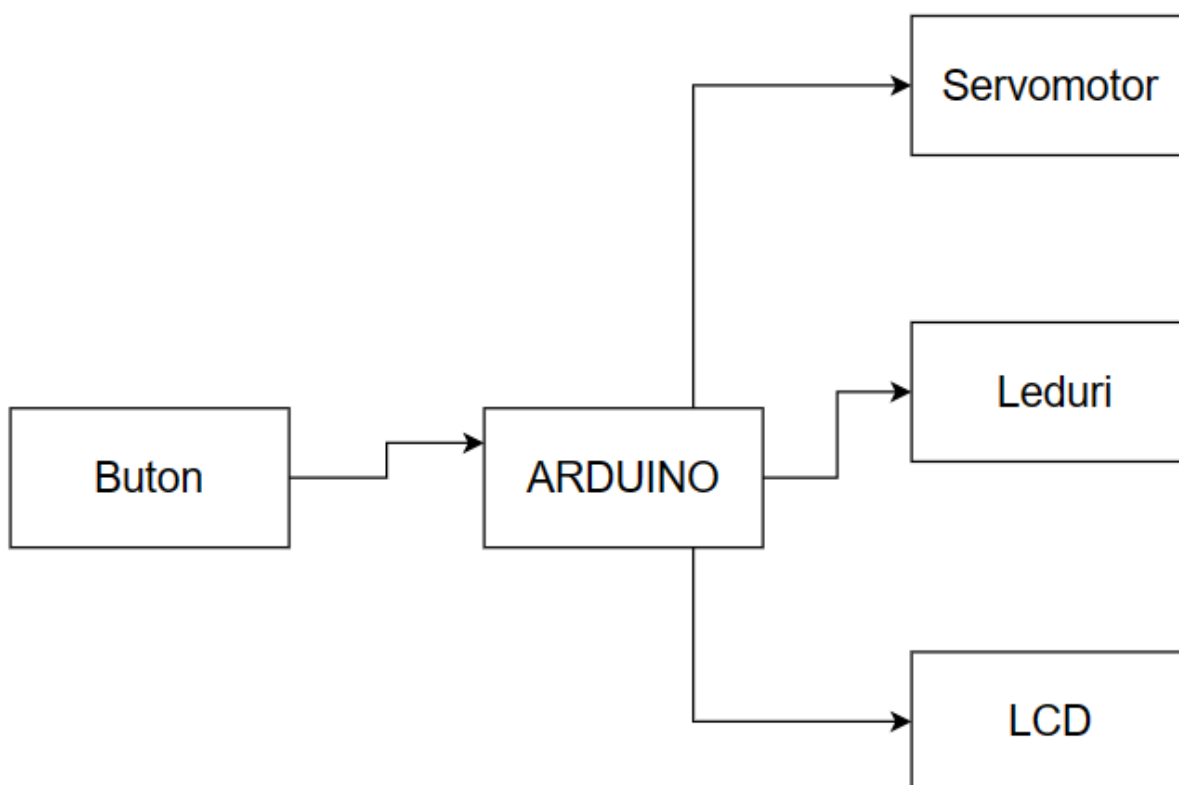
Dice Simulator - Voicila Alexandra

Introducere

Proiectul va simula aruncarea unui zar. In cazul in care ai noroc si nimeresti 6, se va deschide o cutie cu o mica surpriza. In total, vei avea 3 sanse de aruncare care vor fi contorizate si afisate prin intermediul unui ecran. Cred ca este un proiect amuzant si interactiv. In acelasi timp, consider ca proiectul imbina atat cunostinte de PM, cat si de probabilitate/matematica. Pentru a afla care este surpriza va astept la PM fair 

Descriere generală

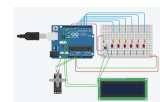
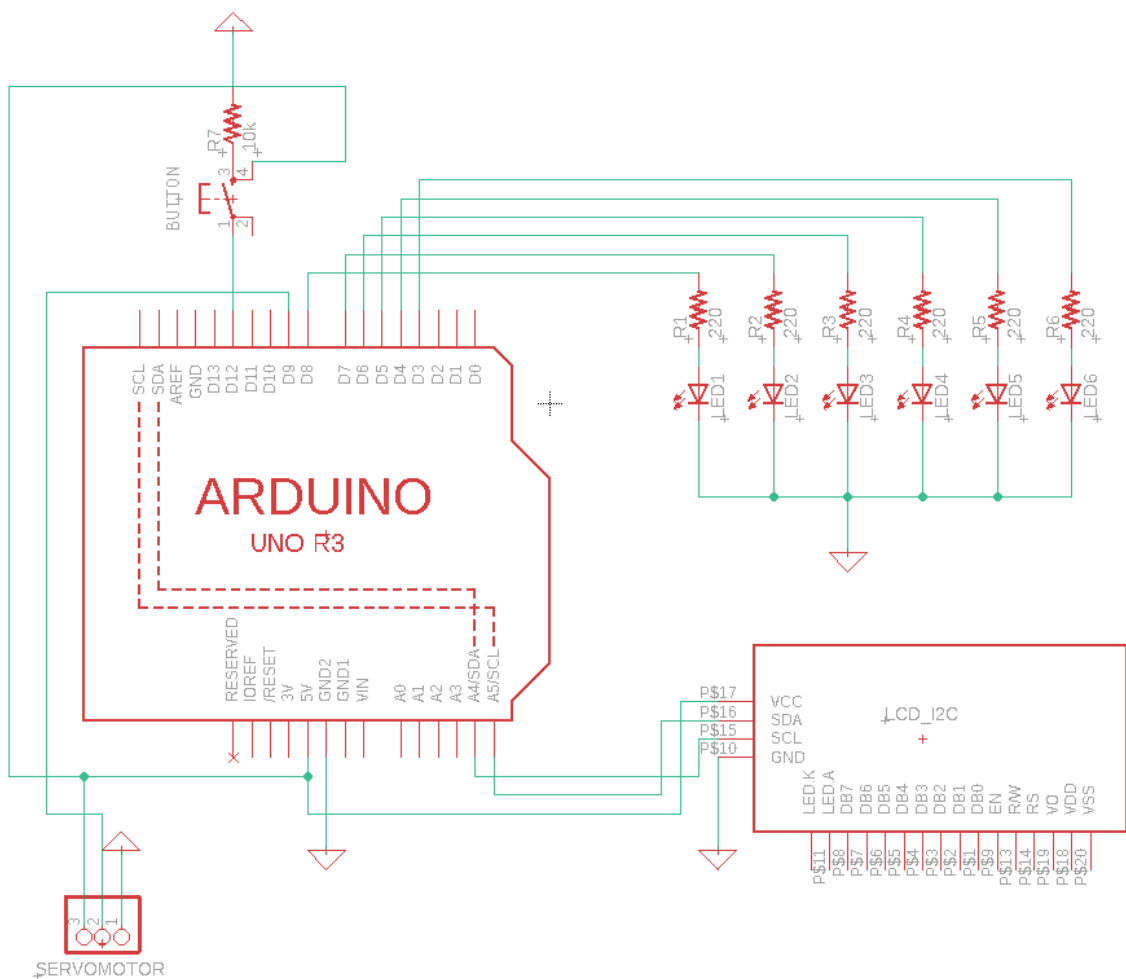
Prin apasarea butonului se va simula aruncarea. In functie de numarul random aruncat, se vor aprinde ledurile specifice. La fiecare aruncare, pe ecranul LCD-ului se va afisa un mesaj corespunzator: numarul de incercari ramase sau joc pierdut/castigat. In cazul in care se aprind toate cele 6 leduri, se va activa servomotorul care va duce la deschiderea cutiei. Dupa cele 3 incercari esuate de a arunca numarul castigator, jocul se reseteaza. Schema bloc:



Hardware Design

Lista componente:

- Arduino UNO
- LCD I2C 1604
- leduri
- fire
- rezistente variabile
- buton
- Servomotor SG90 cu limitator



Software Design

Dezvoltarea codului s-a realizat in Arduino IDE.

Laboratoare: I2C, Intreruperi, PWM

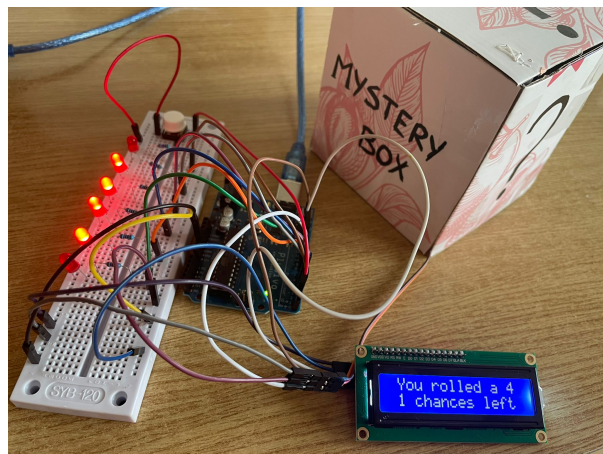
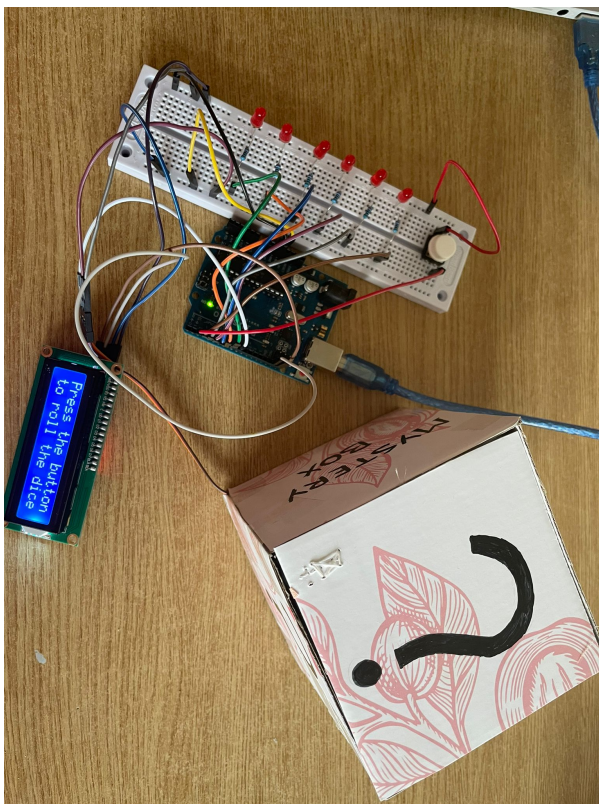
Biblioteci folosite au fost LiquidCrystal_I2C.h, Servo.h pentru controlarea servomotorului, respectiv a display-ul I2C de catre Arduino.

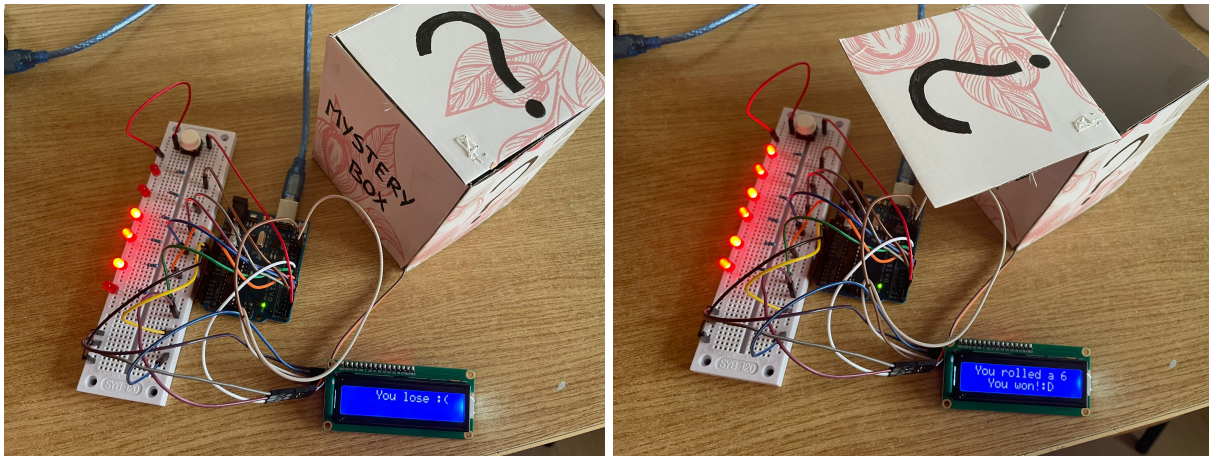
Pentru **initializare**, am folosit variabile globale precum button (pinul la care este conectat butonul), pozitia initiala a servomotorului (aleasa convenabil prin testare).

Functii:


- setup() - initializarea servomotorului, starii pinilor, seriei (pentru afisarea mesajelor de debug), a lcd-ului si a intreruperii folosite la apasarea butonului.
- displayNumOnDice() - se genereza numarul random in functie de care se face aprinderea ledurilor si afisarea unui mesaj specific pe lcd. In cazul in care se nimereste 6, jocul este "castigat", servomotorul se activeaza, deschizand cutia pe care o pastreaza deschisa 5s, un timp suficient pentru jucator sa isi ridice premiul.
- ISR() - rutina de tratare a intreruperii, facuta dupa debounceTime
- rollDice() - functia specifica "aruncarii zarului"
- loop() - apeleaza functia precedenta
- generateRandomSeed() - pentru a creste complexitatea, am folosit aceasta functie de generare a seed-ului in locul functiei obisnuite de random. Folosind functia de analogRead, se realizeaza o citire pe un pin neconectat, "in aer". Dar acest lucru nu este suficient deoarece tensiunea nu fluctueaza la un nivel considerabil. Folosindu-ma totusi de valorile citite, interpretate ca biti, se creeaza prin shiftari si adunari un rezultat final pentru seed pe 32 de biti.

Proiectul finalizat





Concluzii

Mi-a placut mult sa lucrez la acest proiect, fiind una dintre singurele ocazii de a interactiona cu partea hardware. Am avut unele dificultati la inceput (mi-era teama sa nu stric componentele) din cauza lipsei de experienta in lucrul cu circuite fizice. Pentru inceput, am realizat circuitul in thinkercad ca sa pot experimenta mai libera, fara sa am teama amintita anterior. Dar la final, am fost foarte multumita si mandra de proiectul realizat. 

Download

[src_dicesimulator.zip](#)

Jurnal

07.05 - etapa 2
21.05 - etapa hardware
27.05 - etapa software
28.05 - adaugarea ultimelor detalii
29.05 - finalizarea proiectului

Bibliografie/Resurse

<https://docs.arduino.cc/learn/electronics/servo-motors>
<https://arduinogetstarted.com/tutorials/arduino-lcd-i2c>

<https://rheingoldheavy.com/better-arduino-random-values>

<https://www.learnrobotics.org/blog/cool-arduino-projects-arduino-dice-simulation>

Export to PDF

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/fstancu/alexandra.voicila>



Last update: **2023/05/29 18:10**