

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
CATEDRA CALCULATOARE

Arduino Slot Machine

Student : Hulea Andrei-Florin

Grupa : 30235

An universitar : 2020-2021

1. Introducere

Scopul acestui proiect este de a implementa cu folosind o placă Arduino Mega un joc de noroc cat mai realist si usor de folosit.

2. Solutie

Pentru acest proiect am ales sa folosesc urmatoarele componente:

- Arduino Mega 2560
- Breadboard
- Buzzer pasiv
- 2 LED-uri (galben si rosu)
- Display LCD 16x2
- Potentiometru
- 3 butoane (numerotate dreapta la stanga)
- 3 rezistente

In primul rand, pe parcursul jocului vor aparea urmatoarele 4 ecrane pe LCD: ecranul de inceput, ecranul de inceput, ecranul de scor, ecranul de dublare si ecranul de game over.

Cand jocul este pornit, ne vom afla pe ecranul de inceput cu 100 de puncte. Pentru a putea incepe jocul trebuie sa setam valoarea cu care vom paria pe parcursul acestuia (variabila bet). Aceasta poate lua valori de la 10 la 100 si o vom seta cu ajutorul butoanelor 2 si 3, care sunt folosite pentru incrementarea si decrementarea acesteia.

Dupa ce am setat variabila bet in ecranul de inceput, trebuie apasat butonul 1 pentru a incepe jocul, astfel afandu-ne pe ecranul scor unde sunt extrase 14 simboluri random, fiecare cu valori diferite. Simbolul cu cele mai multe aparitii din valoarea curenta va fi cel care se va adauga la scor dupa aplicarea formulei, doar daca numarul de aparitii este mai mare sau egal cu 5. Formula dupa care se calculeaza scorul castigat este urmatoarea:

```
inttoDouble = valoare_simbol * nr_aparitii_simbol * (bet / 10);
```

Valoriile celor 6 simboluri existente, cu exceptia semnului intrebarii, sunt de la 1 la 6, crabul avand cea mai mica valoare si regele cea mai mare. Numarului de aparitii al celui mai frecvent simbol din rularea curenta ii este adaugat si numarul semnelor de intrebare din rularea curenta, counter-ele pentru acestea fiind resetate la fiecare rulare. De asemenea se poate observa ca cat valoarea bet-ului este mai mare cu atat castigul potential este mai mare. Dezavantajul unui bet mare este ca la fiecare rulare, valoarea acestuia este substrata din scorul curent. Asadar

daca avem bet-ul setat pe 100 si la prima rulare nu avem nici un simbol cu numarul de aparitii mai mare decat 5, jocul va fi pierdut.

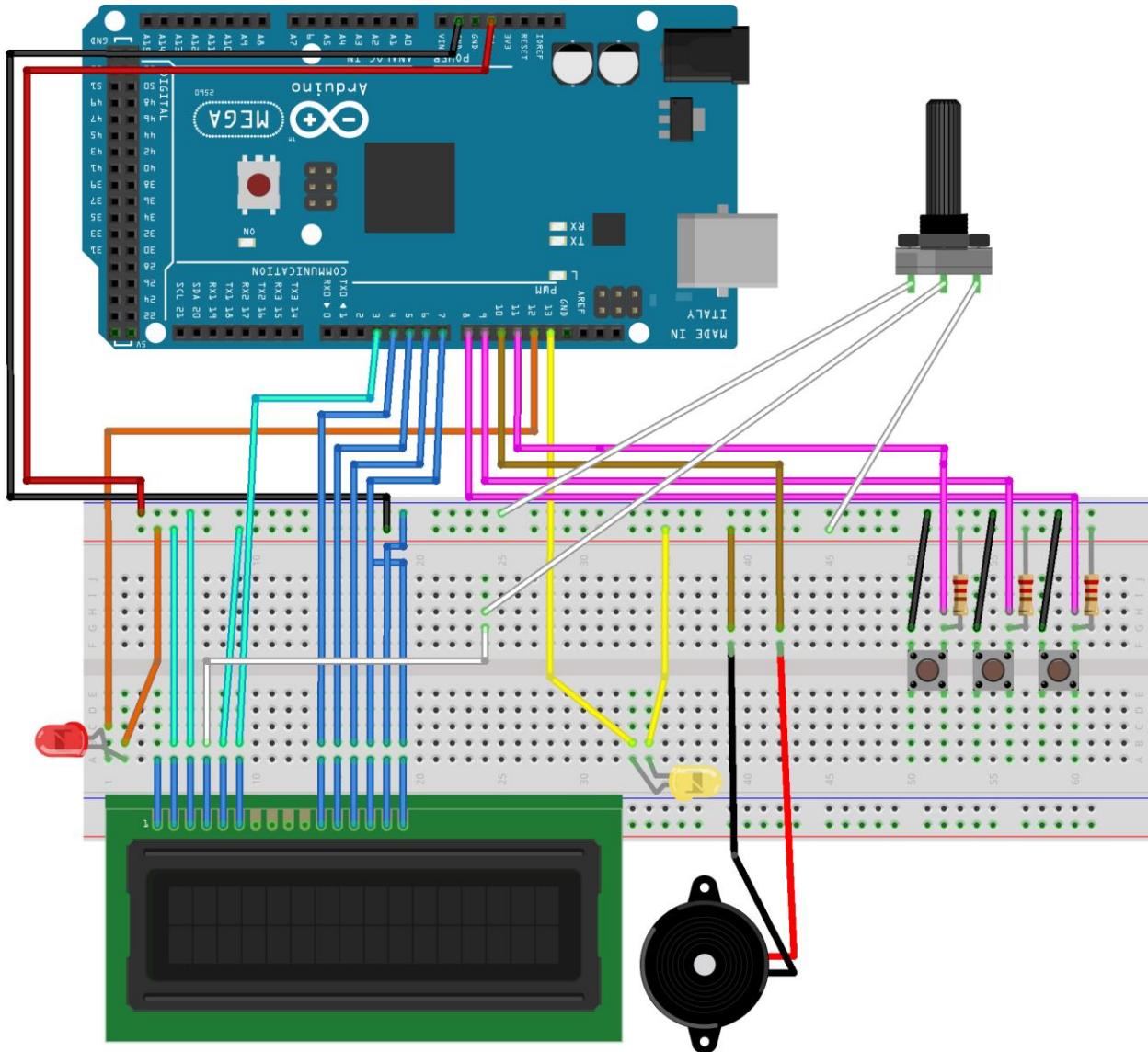
Daca in rularea curenta avem un simbol cu numarul de aparitii mai mare decat 5, vom fi trimisi la ecranul de dublat, de unde putem iesi cu ajutorul butonului 1 pentru a ne pastra castigul si a continua rularile. In cazul in care dorim sa ramanem pe ecranul de dublat si sa dublam scorul castigat in rularea curenta (variabila `toDouble`), vom putea face acest lucru cu ajutorul butoanelor 2 si 3, de maxim 5 ori. Pentru ca dublarea sa fie eligibila, jucatorul trebuie sa ghiceasca valoarea unei variabile random care poate lua valorile 0 sau 1. In cazul in care variabila este 0, jucatorul va trebui sa apese butonul 2, iar daca este 1 va trebui sa apese butonul 3. In cazul in care jucatorul greseste valoarea acesteia, variabila `toDouble` va lua valoarea 0, deci scorul castigat in rularea curenta va fi pierdut. In caz contrar daca jucatorul inca nu a dublat valoarea de 5 ori si doreste sa revina la ecranul de scor ca sa continue jocul, va putea face asta prin apasarea butonului 1.

Daca pe parcursul jocului, valoarea scorului ajunge sa fie mai mica sau egala cu 0, vom fi trimisi la ecranul de game over. Daca dorim sa incepem alt joc, putem prin simpla apasare a butonului 1.

Pe parcursul jocului, buzzer-ul si LED-urile au rolul de a adauga un nivel de realism jocului. Buzzer-ul va emite sunete cand este selectata valoarea bet-ului, cand sunt rulate valorile random, la orice activitate din ecranul de dublari si la game over. De asemenea LED-urile complementeaza buzzer-ul, acestea aprinzandu-se simultan cu majoritatea sunetelor emise de acesta, in principal rolul LED-ului rosu fiind de a indica o valoare gresita si a celui galben o valoare corecta.

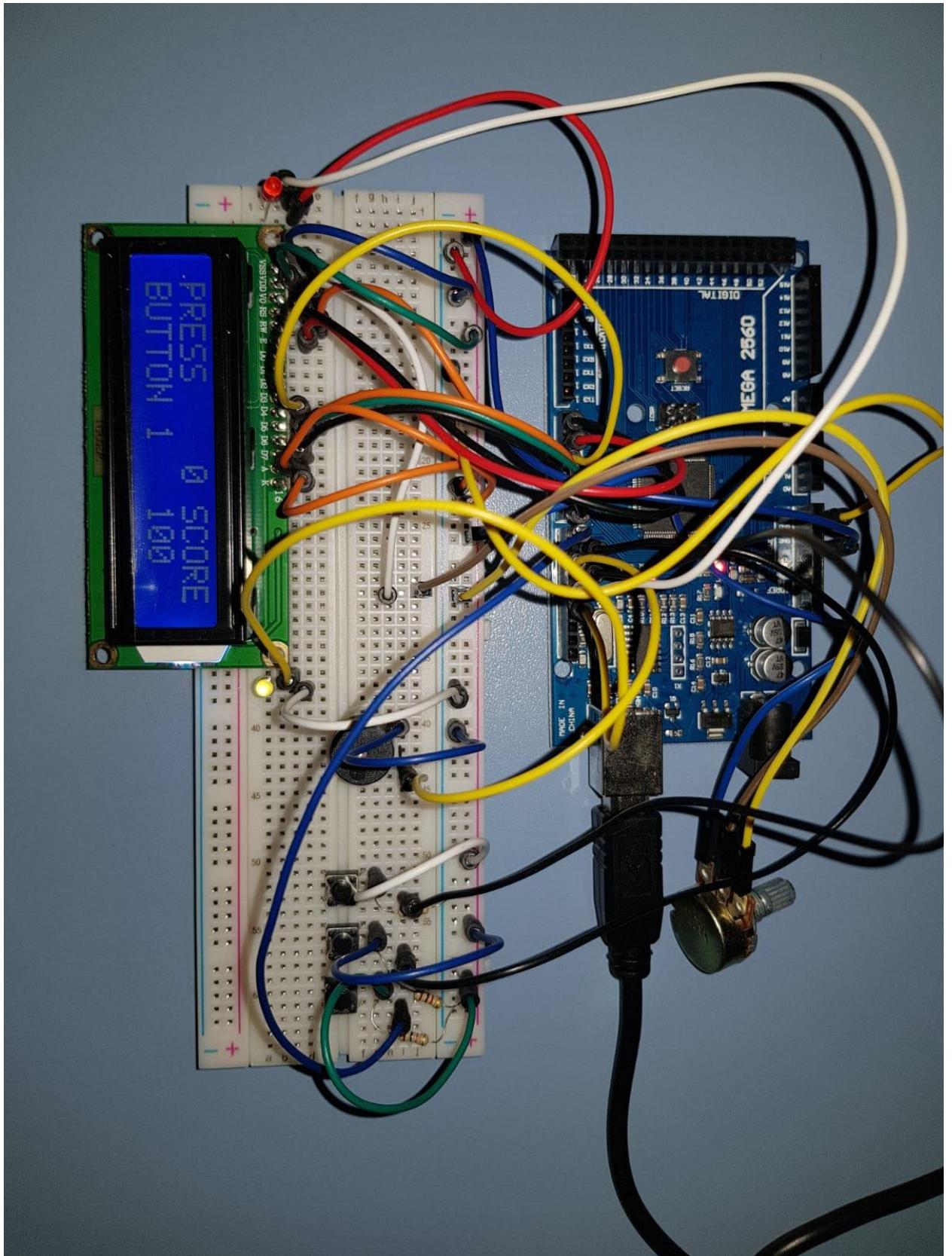
Nu in ultimul rand, deoarece functia `random()` are rolul unui pseudo-generator de numere random, pentru a obtine numere "true" random am folosit functia `randomSeed()` pe un pin neconectat. Aceasta are rolul de a incepe de la o pozitie diferita de fiecare data din secventa aleatoare generata de functia `random()`, astfel obtinand valori diferite de fiecare data.

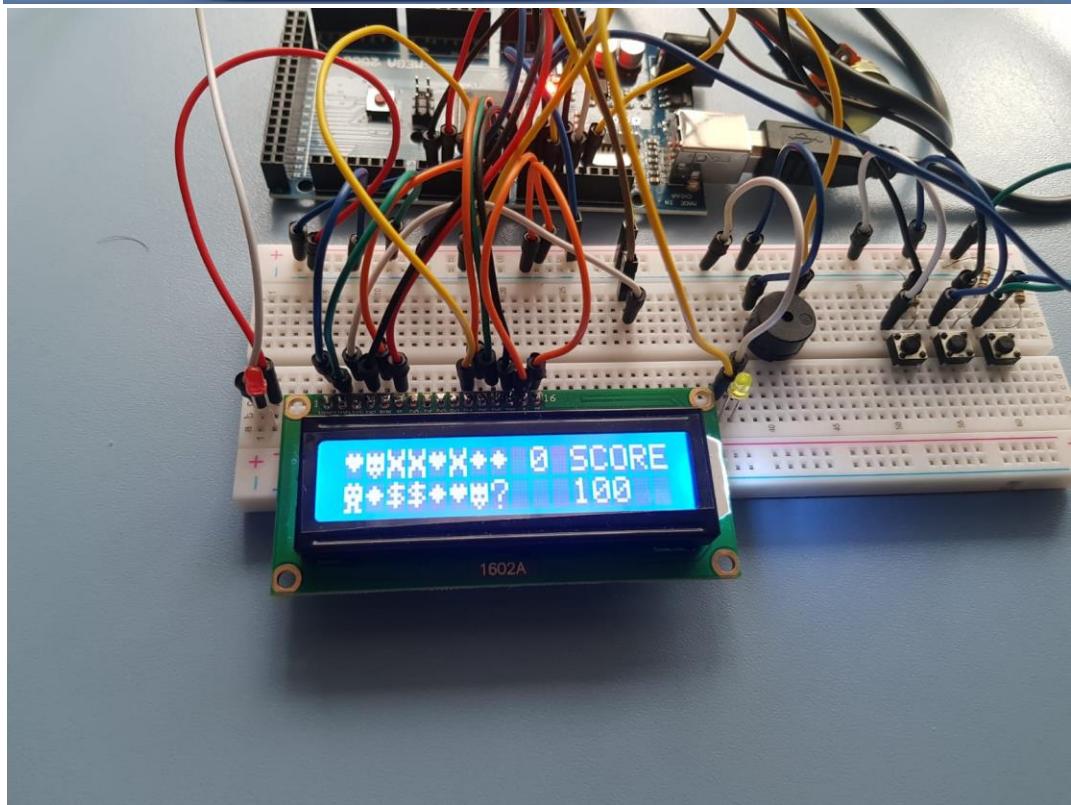
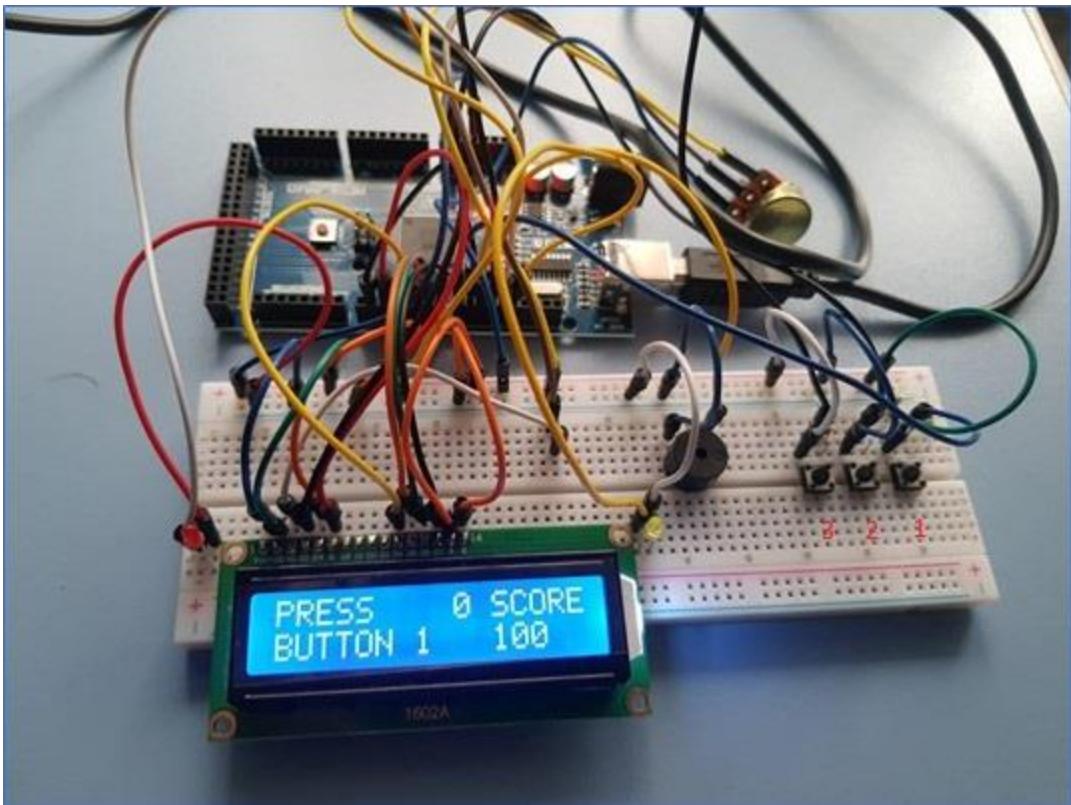
3. Diagramme

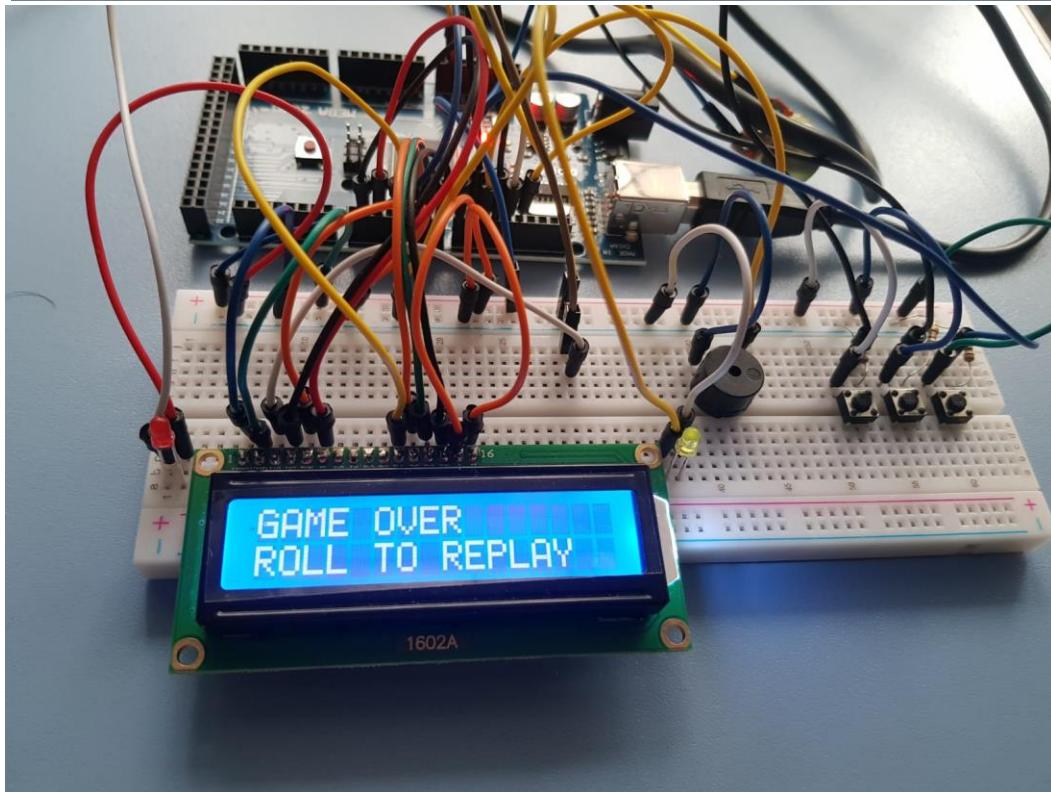
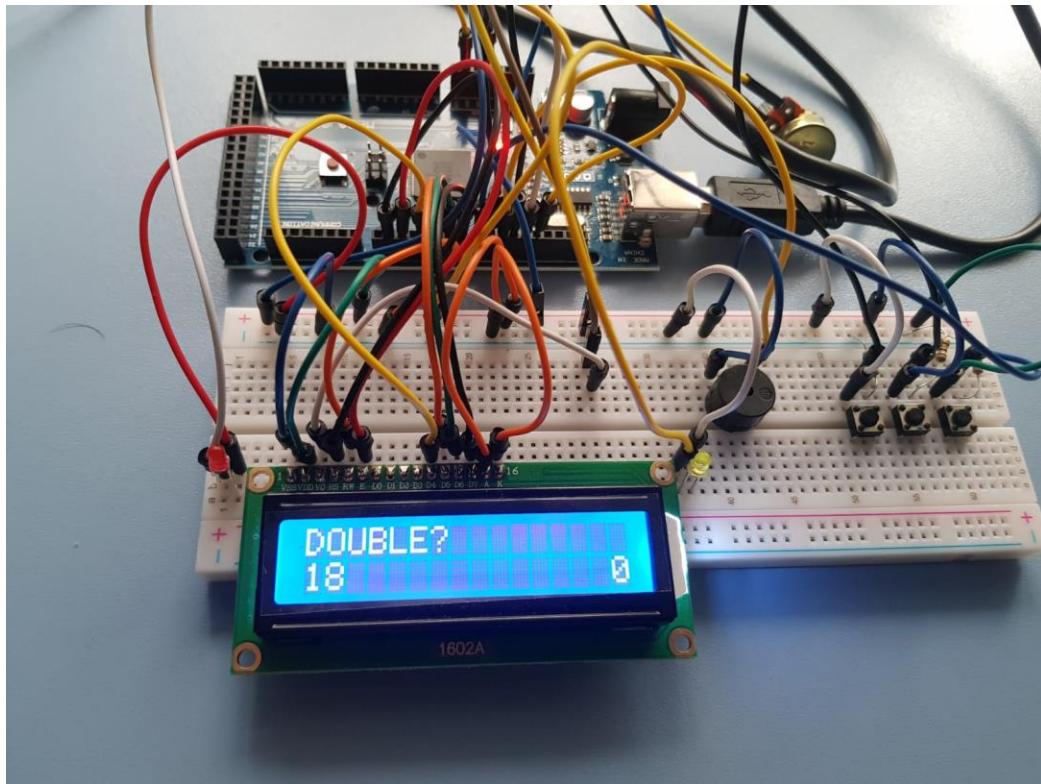


fritzing

4. Poze project







5. Bibliografie

- [random\(\) - Arduino Reference](#)
- [randomSeed\(\) - Arduino Reference](#)
- Indrumator de laborator