

Universitatea Tehnică din Cluj-Napoca, Facultatea de Automatică și Calculatoare
Specializarea: Calculatoare și Tehnologia Informației

Documentație
Proiect Joc cu Mașinuțe

Proiect realizat de: Urcan Diana-Cristina

Grupa 30236, Seria B, Anul III

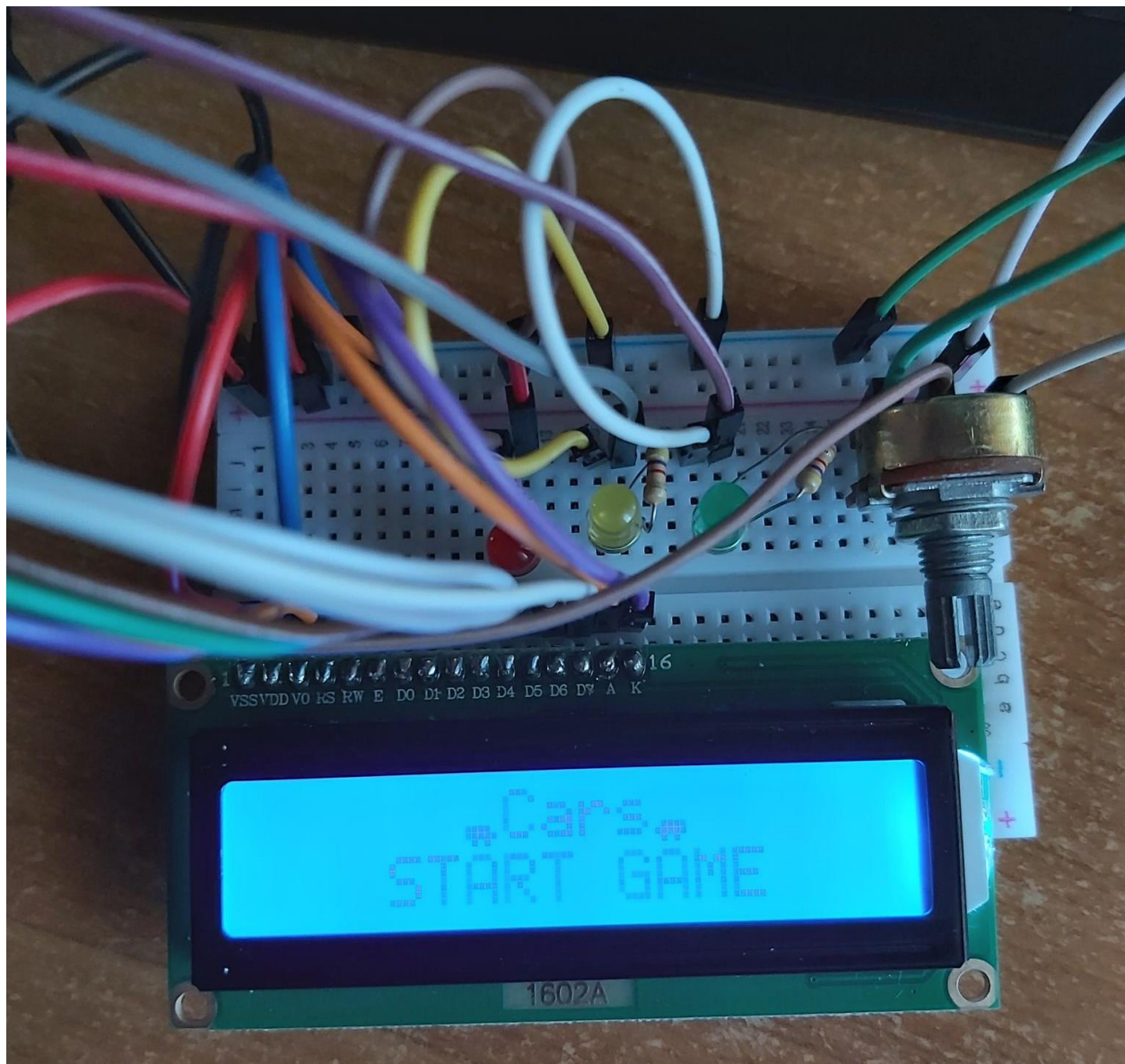
Data: 08.01.2021

Cuprins

1. Prezentare generală	3
2. Componente necesare.....	4
2.1. Arduino.....	4
2.2. LCD 16x2	4
2.3. Potențiomtru 10k.....	5
2.4. Butoane.....	5
2.5. Led-uri, rezistențe, fire, breadboard, alimentare	5
3. Conexiuni	6
4. Implementare	7
4.1 Afișare	7
4.2. Obstacole	7
4.3. Vieți.....	8
4.4. Bonus.....	8
4.5. Semafor.....	8
4.6. Opțiunile butoanelor	8
5. Utilizare	9
6. Dezvoltări ulterioare.....	10
7. Bibliografie.....	10

1. Prezentare generală

Proiectul realizat este unul interactiv – constă dintr-un joc realizat pe un LCD 16x2 cu mașinuțe. Utilizatorul are o mașinuță cu care trebuie să ocolească mașinile ce vin din direcția cealaltă -ele pot fi pe orice bandă. De altfel, există și bonusuri pentru scorul final, ce apar la momente aleatoare de timp - la fel, și vieți suplimentare, în condițiile în care utilizatorul pierde din vieți, în ciocnirea cu celelalte mașini sau la trecerea pe culoarea roșie a semaforului. Toate aceste mutări sunt controlate din potențiomtru și un bloc de butoane, cu patru opțiuni.



2. Componente necesare

Pentru realizarea proiectului s-au folosit următoarele componente

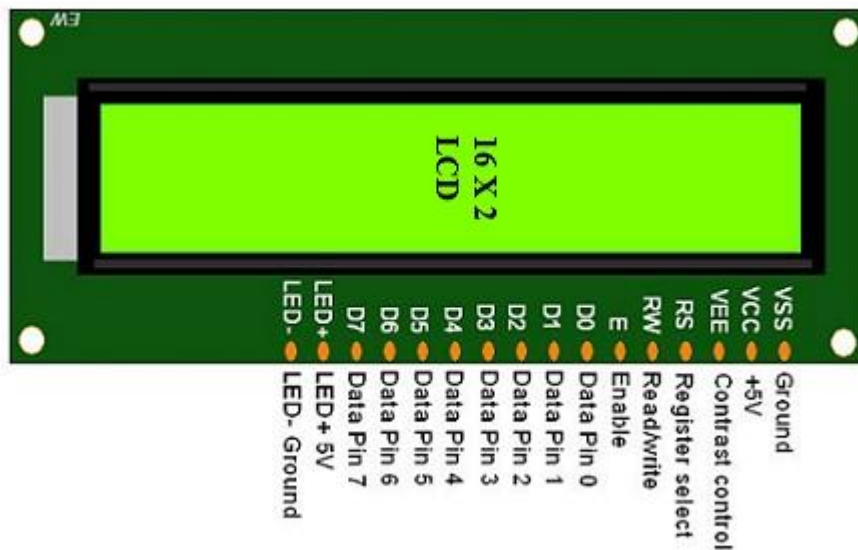
2.1. Arduino

Plăcuța utilizată este Arduino Mega 2560. Atât pentru o bună desfășurare a activităților din cadrul laboratorului, cât și pentru necesitatea numărului mai mare de pini, spre deosebire de Arduino Uno, sau numărului mai mare de întreruperi, s-a ales Arduino mega.



2.2. LCD 16x2

A fost utilizat un lcd 16x2 pentru afișarea și realizarea întregului program- acesta a necesitat cositorire, fiind unul detașat de un shield încorporat. Au fost setați pinii de intrare pentru date prin intermediul librăriei LiquidCrystal -tot prin intermediul acestei librării s-au create și caractere noi – memorate în CGRAM.



2.3. Potențiomtru 10k

Pentru controlarea vitezei mașinuței s-a folosit un potențiomtru liniar conectat în așa fel încât rotirea manetei în stânga să scadă viteza, iar rotirea în dreapta să îi crească viteza.



2.4. Butoane

S-a folosit un bloc de patru butoane pentru a putea realiza acțiunile de schimbare bandă, deplasare înainte/înapoi, start și stop joc.



2.5. Led-uri, rezistențe, fire, breadboard, alimentare

Trei led-uri pentru realizarea semaforului au fost folosite, iar pentru fiecare, câte o rezistență de 4.7kOhm. Toate componentele au fost conectate între ele prin fire mama-tată (pentru blocul de butoane), tata-tată pentru celelalte componente. Majoritatea au fost plasate pe breadboard.

Pentru conectarea plăcuței la laptop s-a folosit un cablu de alimentare normal



Toate aceste componente au fost conectate și programate în așa fel încât jocul să aibă o structură clară și permisivă pentru utilizator. În continuare, se va prezenta modul de conectare al componentelor, iar mai apoi modul de implementare al jocului.

3. Conexiuni

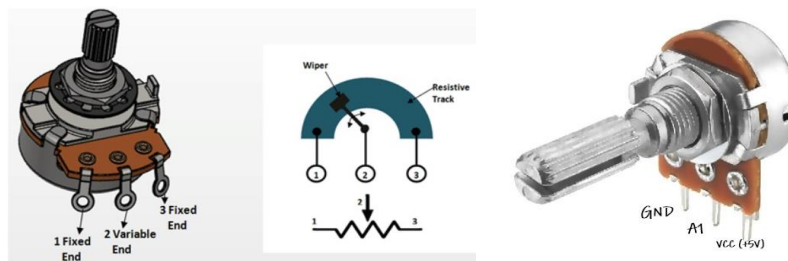
LCD 16x2



VSS -conectat la GND
 VDD - conectat la VCC (5V)
 V0 - conectat la pinul 8 pentru setarea contrastului
 RS - conectat la pinul digital 7
 RW - conectat la GND (nu se citesc date)
 E - conectat la pinul digital 6
 D0-D3 neconectati
 D4-D7 conectati la pinii digitali 5-2
 A - conectat la vcc (5v)
 K - conectat la GND

Pentru pinul 8, se realizează un `analogWrite(8, 100)`; pentru setarea contrastului pentru textul afișat pe lcd. LCD-ul a fost declarat prin intermediul librăriei `LiquidCrystal lcd(7, 6, 5, 4, 3, 2)`; fiecare pin fiind cel specific conexiunii (`lcd(RS, E, D4, D5, D6, D7)`).

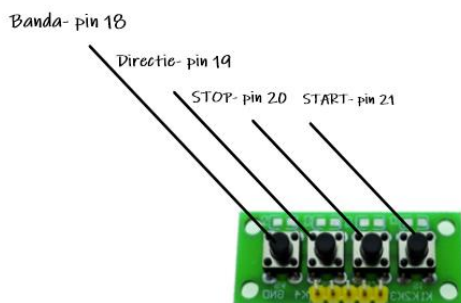
Potențiometrul 10k



Potențiometrul a fost conectat în așa fel încât prin învartirea manetei la stânga, scade rezistența, și scade viteza înregistrată pentru mașinuța utilizatorului, pinul de date este conectat la A1 de unde sunt preluate datele pentru gestionarea vitezei mașinuței. Această valoare a fost transformată prin anumite comparații în numărul de pătrățele cu care mașina avansează într-o anumită direcție. Potențiometrul a fost plasat pe breadboard.

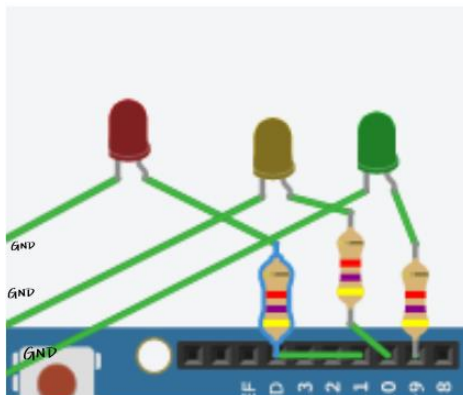
Butoane

Butoanele au fost conectate la pinii corespunzători tratării întreruperilor externe, pentru ca utilizatorul să aibă un control aproximativ total în ceea ce privește desfășurarea jocului -mai precis, să poată să schimbe banda oricând, să poată să schimbe direcția mașinii și să poată începe, respectiv termina, jocul.



LED-uri

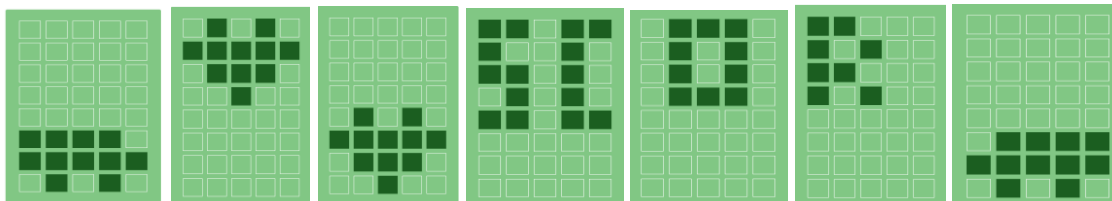
Cele trei led-uri pentru semafor au fost conectate direct pe breadboard, unde le-au fost puse rezistențe de 4.7kOhm pentru protecție. Pinii de semnal pentru ele sunt pinii digitali 9, 10 și 11.



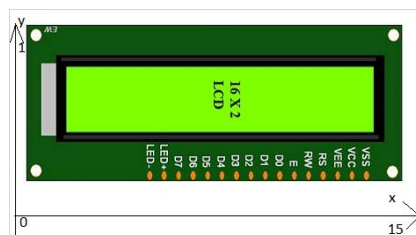
4. Implementare

4.1 Afișare

Mașinuțele și viețile au fost realizate folosind opțiunea de creare prin colorarea anumitor pixeli. La fel, mesajul de scor a fost scris folosind doar jumătate din pixeli, pentru o vizibilitate calitativă a jocului. Au fost folosite cele 8 poziții din memorie pentru a reține mașinuța utilizatorului, mașina ce vine din direcția opusă, inimioarele ce reprezintă viețile sale – cele poziționate superior în matricea pixelilor sunt cele actuale statusului său, iar cele inferioare sunt inimioarele bonus ce apar în momente aleatoare de timp, în condițiile în care utilizatorul a pierdut cel puțin o viață din cele 3. Mașinuțele ce trebuie evitate apar tot la momente aleatoare de timp și la coordonate ce se incrementează odată cu cele ale mașinuței utilizatorului. Aceste 7 caractere, împreună cu caracterul „nimic” formează datele din CGROM pentru joc.



4.2. Obstacole



Deplasarea mașinuțelor și jocul întreg au fost modelate după axele x și y. Pe axa y au fost considerate benzile (0 fiind banda de jos) și 1 fiind banda de sus. Pentru x au fost considerate valorile de la 0 până la lățimea lcd-ului.

Obstacolele sunt reprezentate de mașinuțe din direcția opusă, ce vin pe ambele benzi, la anumiți timpi. Poziția obstacolelor a fost aleasă aleatoriu – s-a selectat un x între 13 și 21 (pentru a veni din

celălalt capăt al benzii, acest x fiind decrementat la fiecare execuție a buclei principale) și un y (0 sau 1), însă, s-a asigurat posibilitatea depășirii prin menținerea a minim doi pixeli pentru a putea schimba banda, depăși și reveni pe banda anterioară – s-au evitat astfel cursele critice, prin verificarea și evitarea pozițiilor mașinilor.

Dacă utilizatorul se ciocnește de o altă mașină, pierde o viață.

4.3. Vieți

Inițial, utilizatorul are 3 vieți. Se verifică la fiecare execuție a buclei principale dacă s-a ciocnit de un obstacol, iar dacă da, îi sunt scăzute viețile. În momentul în care se detectează că nu este numărul maxim de vieți, se generează, în poziții aleatoare pe lcd, inimioare bonus, pe care le poate lua pentru a-și incrementa numărul de vieți. O singură inimioară este generată o dată și se aștepta atingerea acesteia.

4.4. Bonus

În anumite momente de timp, alese aleator, se afișează caracterul '\$', ce poate reprezenta un bonus pentru utilizator, dacă îl accesează în primele 20 de secunde ale jocului – el câștigând +10 la scorul final. Dacă nu, i se scade 50 din scorul pe care îl are. La fel, poate să apară doar un singur caracter bonus la un moment dat.

Atât bonusul, cât și inimioarele sunt generate prin alegerea aleatoare atât a anumitor timpuri, cât și a poziției lor pe lcd.

4.5. Semafor

Semaforul a fost realizat fără delay. La momente aleatoare de timp, s-au reținut timpurile actuali, ca mai apoi să se activeze ledul galben pentru 5 secunde, iar apoi semaforul roșu pentru încă 6 secunde. Dacă utilizatorul conduce în timp ce semaforul este roșu, pierde o viață.

4.6. Opțiunile butoanelor

Pentru fiecare buton a fost utilizată o întrerupere.

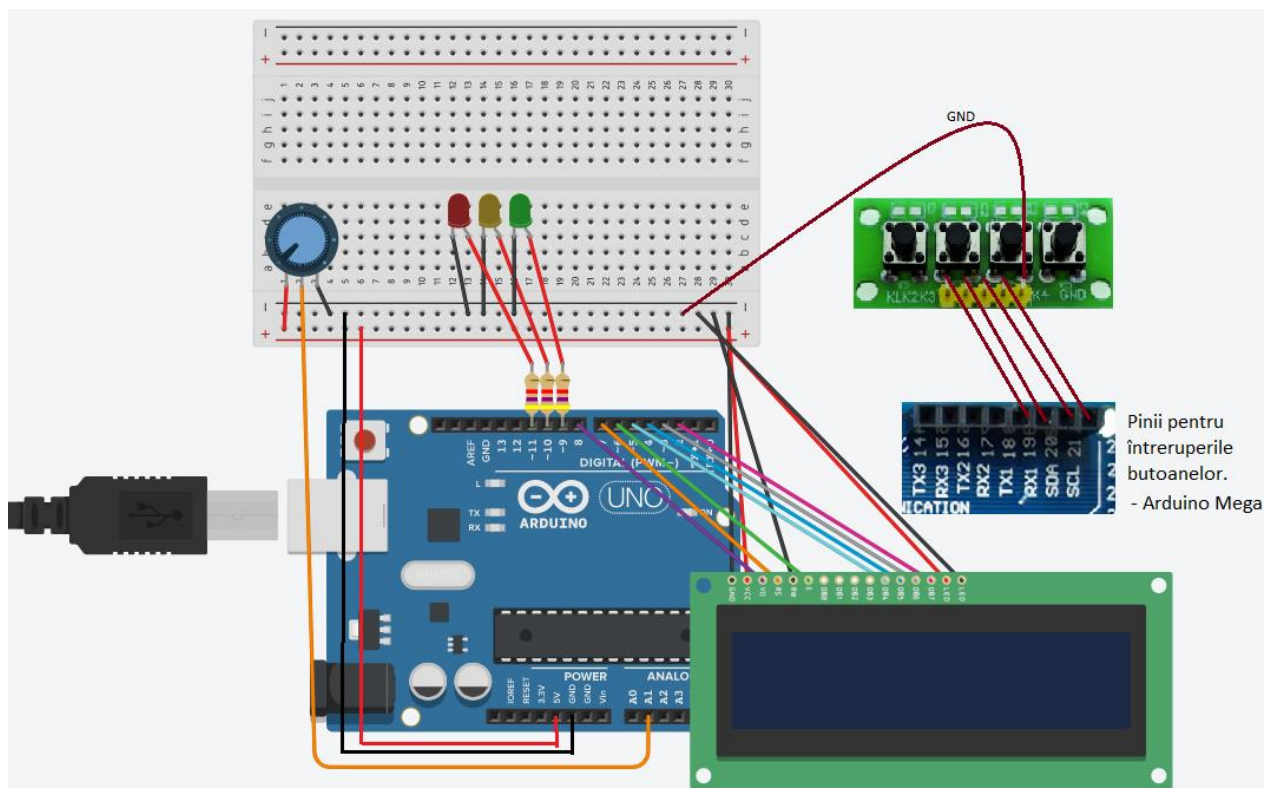
Pinul 21 – pentru buton ce activează întreruperea de începere a jocului. În bucla principală se așteaptă ca o anumită variabilă declarată volatile – pusă în memoria RAM, pentru a putea fi modificată în întrerupere – iar atunci când este activată (setată la true) în cadrul acestei întreruperi, prin apăsarea butonului, începe jocul.

Pinul 20 – pentru butonul ce oprește jocul – acesta dezactivează (setează pe false) variabila pentru începerea jocului și activează variabila pentru finalizarea jocului.

Pinul 19 – pentru butonul ce setează dacă mașina se deplasează în dreapta sau în stânga.

Pinul 18 – pentru butonul ce setează dacă mașina este pe banda 0 sau pe banda 1.

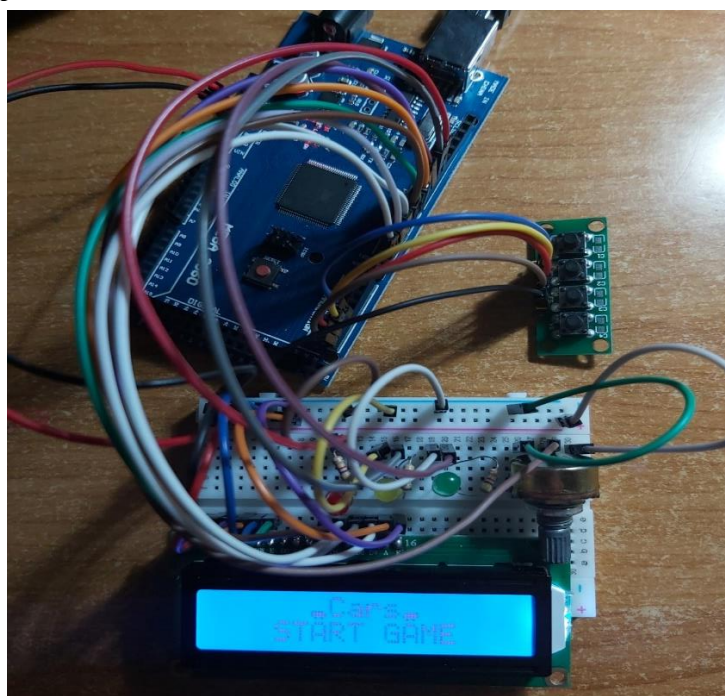
Toate aceste întreruperi sunt necesare deoarece întregul program trebuie să ruleze conform acțiunilor utilizatorului.



O parte din schema de montaj a fost realizată în tinkercad, însă partea ce necesită pinii de întrerupere de la Arduino Mega, a fost realizată în partea dreaptă, unde a fost adăugat blocul de butoane ce nu necesită rezistențe externe, căci sunt folosite cele interne, prin funcționalitatea de INPUT_PULLUP.

5. Utilizare

5.1. Pornire program



5.2. Start joc (primul buton)



5.3. Stop (al doilea buton)



6. Dezvoltări ulterioare

Aplicația poate fi dezvoltată prin folosirea unui LCD ce permite o gamă mai interesantă și mai extinsă de acțiuni, eventual culori și o variantă mai colorată și mai prietenoasă pentru joc. Mașina utilizatorului poate să aibă un claxon și se poate dezvolta un mod de joc interactiv cu doi jucători.

7. Bibliografie

<https://maxpromer.github.io/LCD-Character-Creator/>

<https://lastminuteengineers.com/arduino-1602-character-lcd-tutorial/>

<https://www.arduino.cc/en/Hacking/PinMapping2560>