**Proiectare cu microprocesoare**

**Proiect**

**Nume :** Gabriel Antonesi

**Profesor îndrumător:** Mircea Paul Mureșan

**Grupa:** 3023**5**

Cuprins

[1. Schema 3](#_Toc29773933)

[2. Cerința 5](#_Toc29773934)

[3. Specificații 5](#_Toc29773935)

[4. Manual de utilizare 6](#_Toc29773936)

[5. Descrierea principalelor funcționalități 6](#_Toc29773937)

[6. Bibliografie 8](#_Toc29773938)

# Schema

Shield-urile sunt PCB-uri (plăci de circuit imprimat – Printed Circuit Board) care pot fi

plasate deasupra plăcilor Arduino, extinzându-le astfel capabilitățile. Există o varietate foarte mare de astfel de shield-uri, precum: XBee shield, SD-Shield, H-Bridge shield, etc. Pe PCBurile Shield-urilor, anumite fire sunt trase către baretele de pini care urmează sa fie introduse în Arduino. Pinii folosiți sunt în general specificați în foaia tehnică a produsului.

LCD-ul utilizează pinii digitali de la 2 la 7, astfel: pinul digital 7 - RS; pinul digital 6 - EN; pinul digital 5 - DB4; pinul digital 4 - DB5; pinul digital 3 - DB6; pinul digital 2 - DB7.

Shield-ul este bazat pe controllerul clasic, care se folosește la LCD-uri, Hitachi HD44780. LCD-urile care folosesc cel mult 80 de caractere distincte și cel mult 4 linii de afișare au nevoie de un singur controller. LCD-urile care au mai mult de 80 de caractere au nevoie de două controlere.

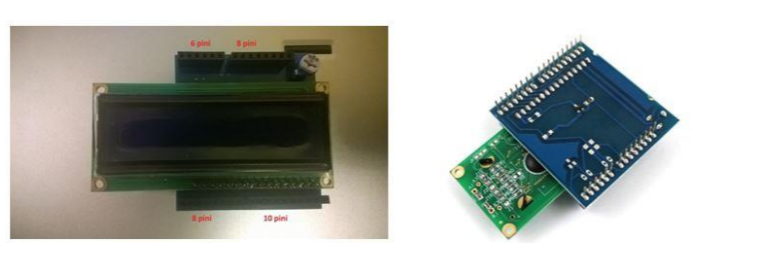


Figura 1. Shield-ul LCD Arduino

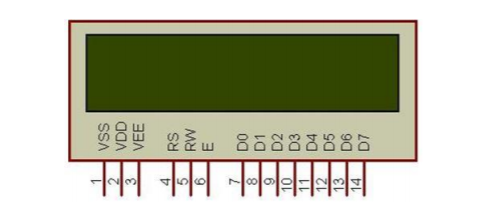
Controllerul HD44780 conține două registre pe 8 biți: registrul de date și registrul de

instrucțiuni. Registrul de instrucțiuni e un registru prin care LCD primește comenzi (shift, clear etc). Registrul de date este folosit pentru a acumula datele care vor fi afișate pe display. Când semnalul Enable al LCD-ului este activat, datele de pe pinii de date sunt puse în registrul de date, apoi mutate în DDRAM (memoria de afișaj, Display Data RAM) și afișate pe LCD. Registrul de date nu este folosit doar pentru trimiterea datelor către DDRAM ci și

către CGRAM, memoria care stochează caracterele create de către utilizator (Character Generator RAM).

Display Data Ram (DDRAM) stochează datele de afișare, reprezentate ca și caractere de

8 biți. Capacitatea extinsă a memoriei este de 80 X 8 biţi, sau 80 de caractere. Memoria rămasă liberă poate fi folosită ca un RAM generic. Pe LCD-ul nostru se afișează doar 2x16 caractere, deoarece aceasta este dimensiunea afișorului, dar controllerul poate stoca 80.



Semnificația pinilor shield-ului LCD:

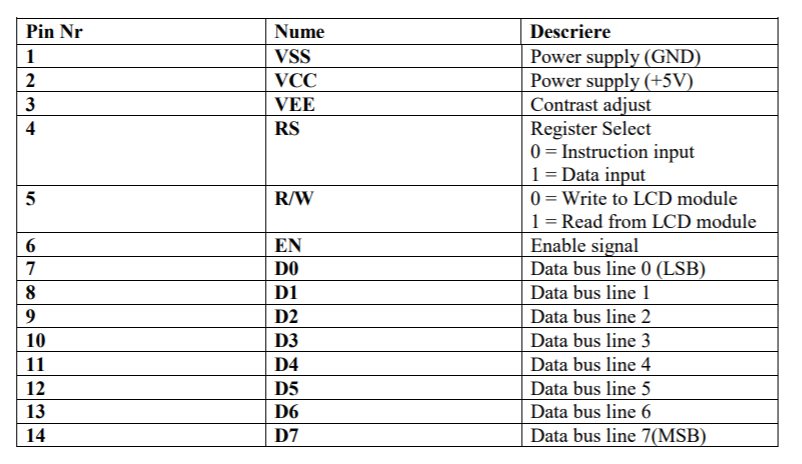


Figura 2. Specificațiile pinilor

Alte specificații ale circuitului:

* Tensiunea de funcționare are valori între 4.7V și 5.3V
* Curentul consumat este de 1 mA
* Dispune de afișaj alfanumeric
* Conține doua rânduri orizontale paralele care pot fiecare afișa până la 16 caractere
* Fiecare caracter e constituit de o caseta de dimensiune 5x8 pixeli

# Cerința

Proiectul propune implementarea pe shield-ul LCD a unui joc distractiv care va imita celebrul T-Rex Run! conceput de gigantul Google. Utilizatorul va putea controla o animație a unui om în încercarea sa de a ocoli obstacolele de pe traseu pentru a obține un scor cât mai bun.

# Specificații

* 1. **Componente fizice utilizate**

Simularea jocului presupune utilizarea următoarelor componente:

* O placă de dezvoltare compatibila (Arduino Uno)
* Un shield LCD 16x2
* Un buton
* O rezistență
* O placă de protitipizare (*breadboard)*
* Un cablu USB

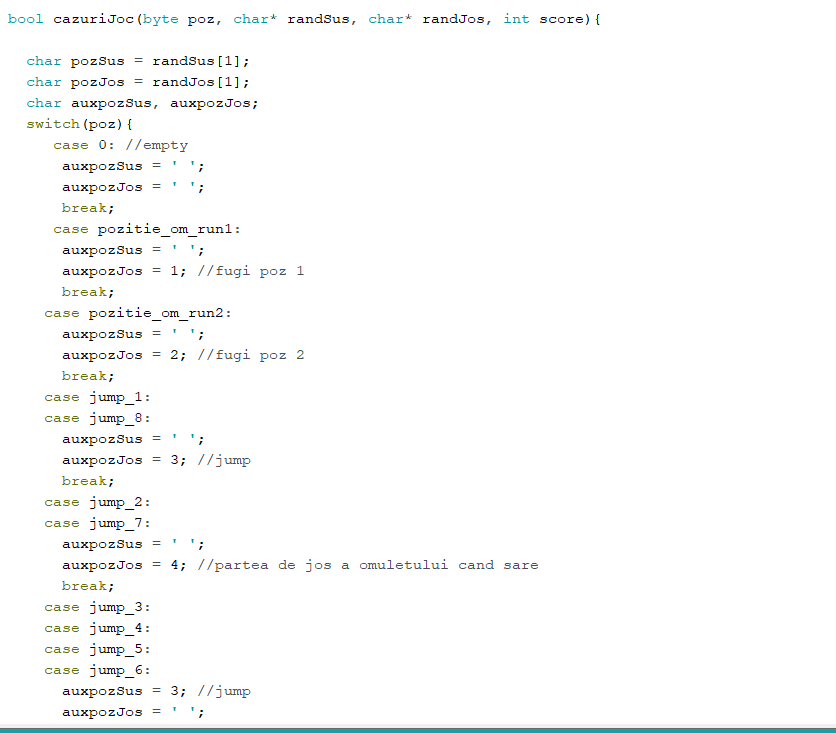
# Manual de utilizare

Pentru utilizarea sistemului, utilizatorul va trebui sa conecteze componentele precum o arată descrierea de mai sus, urmând la final să cupleze sistemul la o sursă de alimentare prin intermediul cablului USB. Butonul de inițiere al sistemului este reprezentat de cel conectat la placă, apăsarea lui, atașată-i fiind o întrerupere, ducând la pornirea propriu-zisă a jocului.

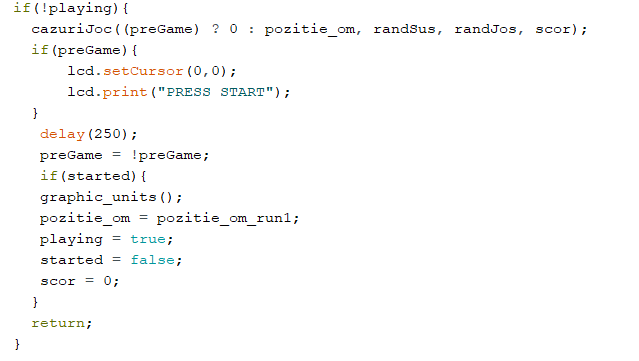
După acționarea butonului de start, pe ecranul LCD va începe animația jocului. Obstacolele vor fi generate în mod aleatoriu pe cele două rânduri, jucătorul trebuind să le evite. La o acționare a butonului de joc, protagonistul va executa un salt în încercarea de a evita obstacolul. În cazul în care apare o coliziune între jucător și obstacol sesiunea va fi încheiată. Scorul este egal cu numărul de porțiuni parcurse până la întâlnirea unei coliziuni.

# Descrierea principalelor funcționalități

În implementarea proiectului am creat mai multe funcții pentru a ajunge la scopul final dorit. Una dintre acestea este funcția *cazuriJoc* care are ca parametrii poziția cursorului, vectorii de pixeli reprezentând cele două rânduri de pe LCD și scorul. Această funcție selectează modul în care protagonistul va fi desenat. Totodată, funcția tratează cazul în care jocul se termina și cazul de afișare al scorului pe mai multe poziții, fără a intra în conflict cu obstacolele.



Înainte ca butonul de start să fie apăsat pentru prima oară, există un ecran de preGame:



# Bibliografie

<https://components101.com/sites/default/files/component_datasheet/16x2%20LCD%20Datasheet.pdf>

<https://www.arduino.cc/en/Tutorial/HelloWorld>

<https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/336-3.pdf>