

Hangman LCD

Autor : Udristioiu Alexandru

Grupa : 331CB

Email : udristioiualexandru@gmail.com

Introducere

Proiectul meu se bazeaza pe jocul Hangman(spanzuratoarea).Mesajele specifice verificarii fiecarei litere se vor afisa pe un LCD , precum si castigarea/pierderea jocului.

Descriere generală

Voi folosi un keypad pentru introducerea literelor si voi verifica daca se afla in cuvant,moment in care voi semnala daca aceasta exista printr-un sunet provenit de la un buzzer.Altfel va scadea numarul de vietii(3) initiale.Daca numarul de vietii este 0 atunci jocul s-a terminat(cu mesaj pe ecran + sunet). Output-ul se va afisa pe un LCD si atunci cand s-au ghicit toate literele(+ sunet). Cuvantul ce se doreste ghicit va fi citit de pe interfata seriala.Din lungimea totala a cuvantului se scade numarul de aparitii al literei ghicite.Atunci cand s-a ajuns la 0 inseamna ca acesta a fost complet gasit.Pe ecran se vor afisa si mesaje in cazul in care litera a fost deja verificata,s-a regasit in cuvantul dat(unde se afiseaza si numarul de aparitii + pozitia literei) sau ea nu apare .

Schema bloc:



Hardware Design

Lista de piese:

- * Arduino
- * Buzzer
- * Rezistente
- * Keypad 4×4
- * Breadboard
- * Fire
- * Ecran LCD 1602
- * Potentiometru

Schema electrica



Software Design

Am lucrat in mediul de dezvoltare [Arduino IDE](#). Ca si biblioteci am folosit urmatoarele :

- LiquidCrystal.h - partea de LCD si interfatarea cu aceasta
- Keypad.h - lucrul cu tastatura matriceala 4×4 (link download : [Keypad](#))

Variabile folosite :

1. lcd - de tip LiquidCrystal setat cu pinii respectivi
2. alphabet[26] - toate literele din alfabet
3. Pentru tastatura :
 - rows - numar de linii
 - columns - numar de coloane
 - pinRows - pinii asociati pe linii
 - pinColumns - pinii asociati pe coloane
 - keys[rows][columns] - matricea de taste
 - keyboard - de tip Keypad setata cu parametrii de mai sus (pentru keys se foloseste makeKeyMap() pentru conversia de la 2d la un key map inteles de librarie)
4. st - cuvantul dat
5. len - lungimea cuvantului
6. nrLives - numarul de vietii ale jucatorului setat la 3
7. key - char pentru primirea tastei apasate
8. count[14] - vector in care tin pentru fiecare tasta numarul de apasari
9. v - sir de caractere unde tin literele apasate deja
10. flag - variabila folosita pentru a citi o singura data cuvantul

Functii :

1. **search(char letter, String st, int pos[])**

Caut litera in cuvânt și returnez numărul de apariții, iar în vectorul pus în pozițiile la care se află litera găsită dacă există.

2. **CheckKeypad(char c, String word)**

Funcție în care verific prin funcția search de câte ori apare litera în cuvânt și scad din lungimea totală a cuvântului nr de apariții (dacă apare) sau scad numărul de vieți. La final returnez numărul de vieți rămase.

3. **Win()**

Funcție în care se ajunge dacă tot cuvântul a fost ghicit. Se notifică acest fapt prin mesaj pe ecranul LCD și un sunet corespunzător. Se resetează variabilele folosite mai apoi.

4. **endGame()**

Funcție în care se ajunge dacă numărul de vieți a ajuns la 0 (jocul a fost pierdut). Se notifică acest fapt prin mesaj pe ecranul LCD și un sunet corespunzător. Se resetează variabilele folosite mai apoi.

5. **setup()**

Funcție în care am initializat lcd-ul, am setat pinul PWM pentru buzzer ca OUTPUT și am pornit interfața serială.

6. **loop()**

Aici a fost implementată logica jocului.

Prima oară se citește cuvântul de pe interfața pe baza căreia sunt setate variabilele len, flag, count și v. Pentru afișarea unei variabile de tip String pe LCD am făcut conversia la un vector de char prin [toCharArray\(\)](#) și după am afișat.

Apoi se preia cheia apăsată și se începe verificarea literelor. `getIdx` este o variabilă folosită pentru a salva cheia apăsată.

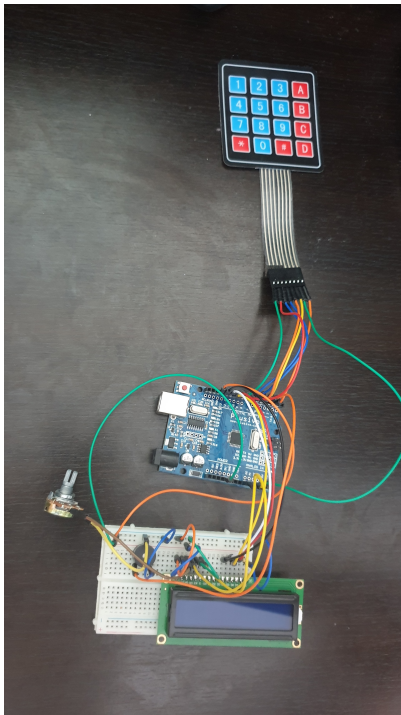
Pentru cheia 7 verific litera obținută și pentru celelalte de la 0 la 9 pun în vector numărul de apăsări care se resetează dacă a fost apăsată de mai mult de 3 ori. Primele 4 litere (a, b, c, d) au fost initializate cu cele de pe tastatură.

Pentru verificare prima oară selectez litera pe care o caut pe baza numărului de apăsări și a tastei, apoi verific dacă ea a mai fost apăsată deja și după apelez funcția `CheckKeypad`.

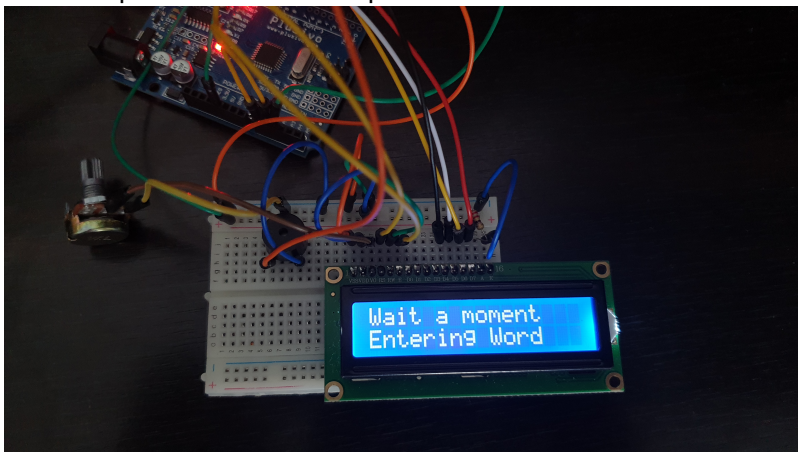
Pentru sunete am folosit funcția [tone\(\)](#) ce generează un semnal de PWM cu 50% duty cycle (pin-ul pe care este setat trebuie să aibă suport pentru PWM).

Rezultate Obținute

Montaj:



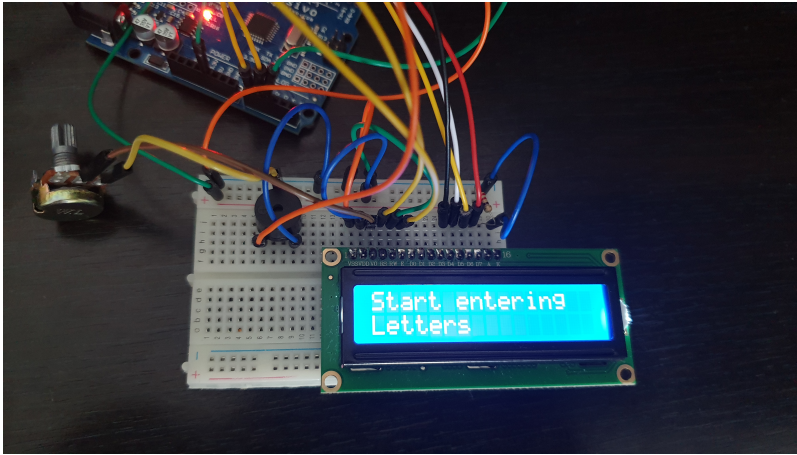
Se asteapta cuvantul venit pe interfata USART



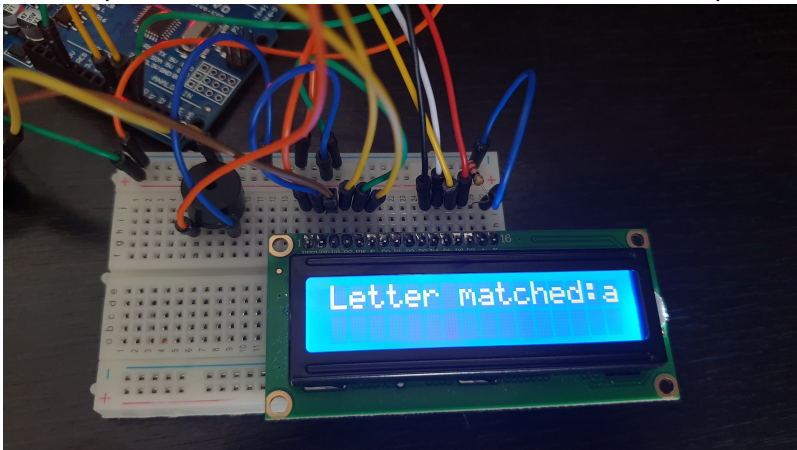
Se afiseaza prima si ultima litera



Se incepe verificarea literelor apasate



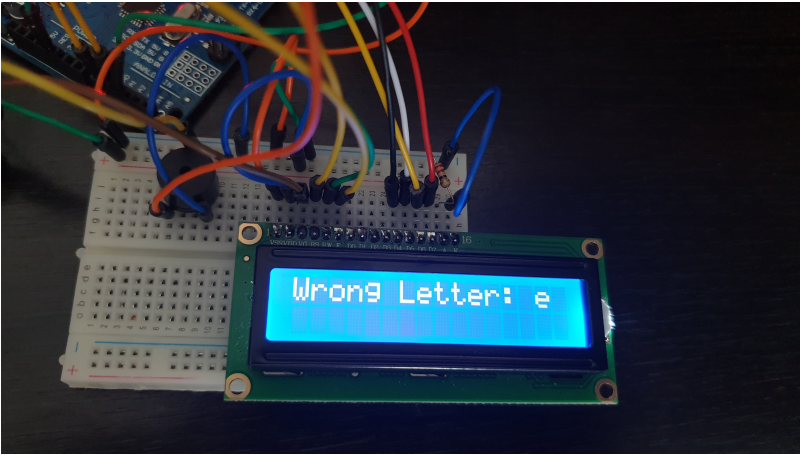
Litera apartine cuvântului si se afiseaza numarul de aparitii si pozitiile acesteia :



Toate literele au fost ghicite :



Litera nu este in cuvânt :



Numarul de vieti este 0 si s-a incheiat jocul :



Litera a fost deja verificata :



Demo-ul proiectului se gaseste pe drive aici : [Hangman_demo](#)

Concluzii

A fost un proiect interesant. Consider ca este o experienta din care am foarte multe de invatat.

Download

In arhiva se afla fisierul Arduino si un readme pentru asocierea litera-tasta:
[331cb_udristioiualexandru_hangman.zip](#)

Jurnal

- * 24 aprilie - Alegere tema proiect
- * 25 aprilie - Creare wiki
- * 29 aprilie - Adaugare schema electrica
- * 30 aprilie - Implementare cod(tinkercad)
- * 11 mai - Achizitionare piese
- * 14 mai - Realizare montaj
- * 16 mai - Imbunatatire cod si verificare pe montaj
- * 25 mai - Finalizare pagina wiki

Bibliografie/Resurse

PDF

- * <https://playground.arduino.cc/Code/Keypad/>
- * <https://www.arduino.cc/en/Reference/LiquidCrystal>
- * <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2021>
- * <https://ocw.cs.pub.ro/courses/pm/lab/lab2-2021> - interfata USART
- * <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2021> - PWM
- * <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2021> - LCD
- * Schema bloc a fost facuta in <https://app.diagrams.net>
- * Schema electrica a fost facuta in <https://www.tinkercad.com/dashboard>

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/hangman_lcd



Last update: **2021/05/27 17:20**