

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 vieti(3) initiale.Daca numarul de vieti 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 cuvant si returnez numarul de aparitii,iar in vectorul pus tin pozitiile la care se afla litera gasita daca exista.

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

Functie in care verific prin functia search de cate ori apare litera in cuvant si scad din lungimea totala a cuvantului nr de aparitii(daca apare) sau scad numarul de vietii.La final returnez numarul de vietii ramase.

3. **Win()**

Functie in care se ajunge daca tot cuvantul a fost ghicit.Se notifica acest fapt prin mesaj pe ecranul LCD si un sunet corespunzator.Se reseteaza variabilele folosite mai apoi.

4. **endGame()**

Functie in care se ajunge daca numarul de vietii a ajuns la 0(jocul a fost pierdut).Se notifica acest fapt prin mesaj pe ecranul LCD si un sunet corespunzator.Se reseteaza variabilele folosite mai apoi.

5. **setup()**

Functie in care am initializat lcd-ul,am setat pinul PWM pentru buzzer ca OUTPUT si am pornit interfata seriala.

6. **loop()**

Aici a fost implementata logica jocului.

Prima oara se citeste cuvantul de pe interfata pe baza caruia sunt setate variabilele len,flag,count si v.Pentru afisarea unei variabile de tip String pe LCD am facut conversia la un vector de char prin [toCharArray\(\)](#) si dupa am afisat.

Apoi se preia cheia apasata si se incepe verificarea literelor.getIdx este o variabila folosita pentru a salva cheia apasata.

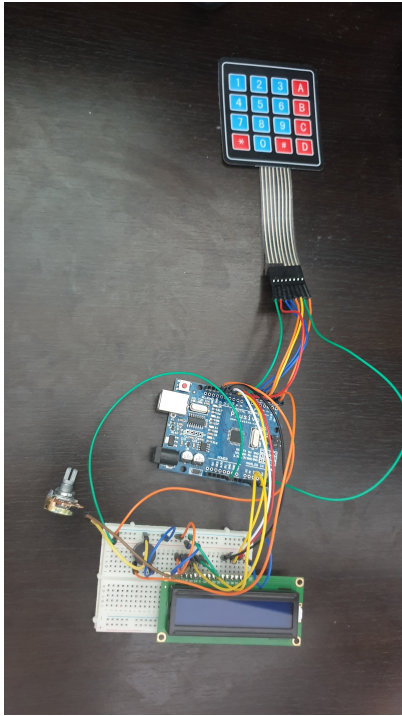
Pentru cheia 7 verific litera obtinuta si pentru celelalte de la 0 la 9 pun in vector numarul de apasari care se reseteaza daca a fost apasat de mai mult de 3 ori..Primele 4 litere(a,b,c,d) au fost initializate cu cele de pe tastatura.

Pentru verificare prima oara selectez litera pe care o caut pe baza numarului de apasari si a tastei,apoi verific daca ea a mai fost apasata deja si dupa apelez functia CheckKeypad.

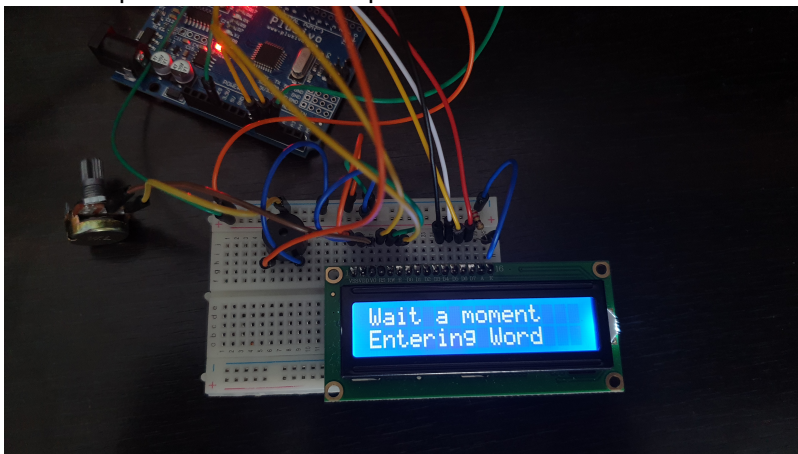
Pentru sunete am folosit functia [tone\(\)](#) ce genereaza un semnal de PWM cu 50% duty cycle (pin-ul pe care este setat trebuie sa aiba suport pentru PWM).

Rezultate Obținute

Montaj:



Se asteapta cuvantul venit pe interfata USART



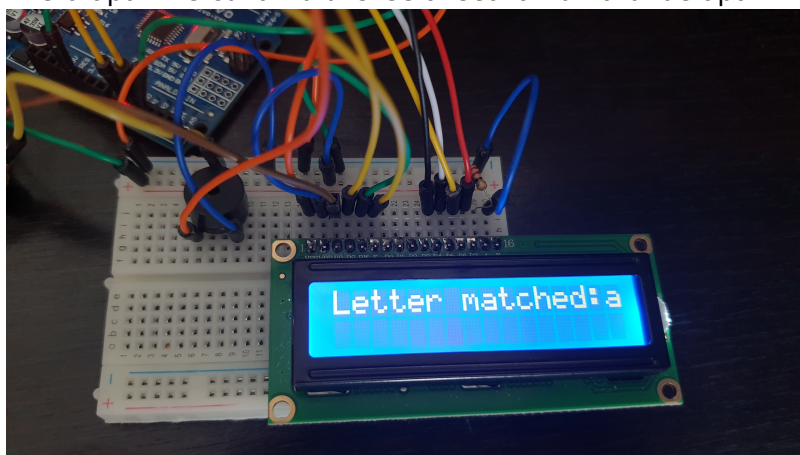
Se afiseaza prima si ultima litera



Se incepe verificarea literelor apasate



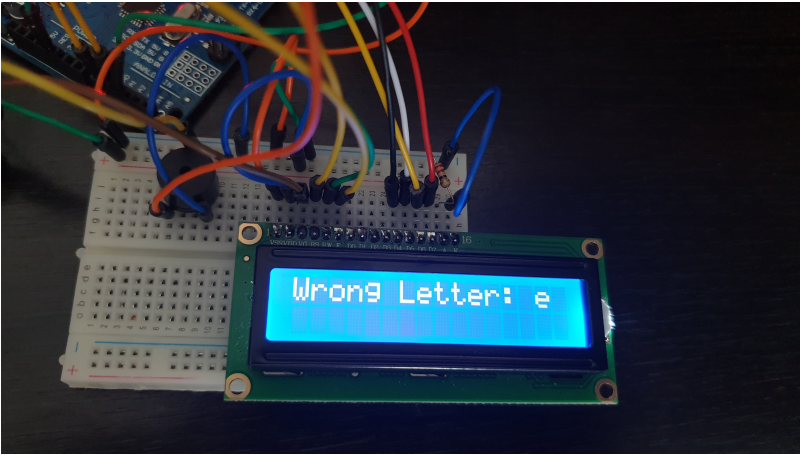
Litera apartine cuvântului si se afiseaza numarul de aparitii si pozitiile acesteia :



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**