

# Wordle LCD

Piele Bogdan-Gabriel, 333CCb

## Introducere

Proiectul este o adaptare a jocului Wordle cu scopul de a ghici un cuvânt scurt. Spre deosebire de Spânzurătoarea, în care literele sunt introduse una câte una, având o limită de greșeli, pentru Wordle jucătorul trebuie să introducă întreg cuvântul, având 6 încercări.

## Descriere generală

Jocul începe la apăsarea butonului START / RESET, după care urmează selectarea limbii (LEFT pentru română, RIGHT pentru engleză). Folosind butoanele UP, DOWN se poate cicla prin alfabet pentru selectarea literei corespunzătoare cursorului. Prin apăsarea LEFT și RIGHT se va trece prin cele 5 litere ale cuvântului introdus. La apăsarea SELECT se va trimite cuvântul spre verificare, dacă toate pozițiile sunt completate cu litere.

În jocul original, la introducerea unei încercări, literele sunt necolorate, dacă nu există în cuvânt, colorate cu galben dacă sunt prezente, dar pe altă poziție, sau cu verde dacă există pe poziția respectivă.

Cum voi folosi un display I2C LCD monocrom, acest tip de feedback va fi înlocuit cu păstrarea doar a literelor la poziția corectă în cuvântul introdus, urmând ca cele care se află de fapt pe altă poziție să fie indicate utilizatorului într-o listă, pe linia a doua a ecranului. În plus, la selectarea unui cuvânt, buzzer-ul și LED-ul vor avea un feedback scurt, intermitent, pentru fiecare literă așezată corect, și un feedback mai lung la ghicirea întregului cuvânt.

### Schemă bloc:



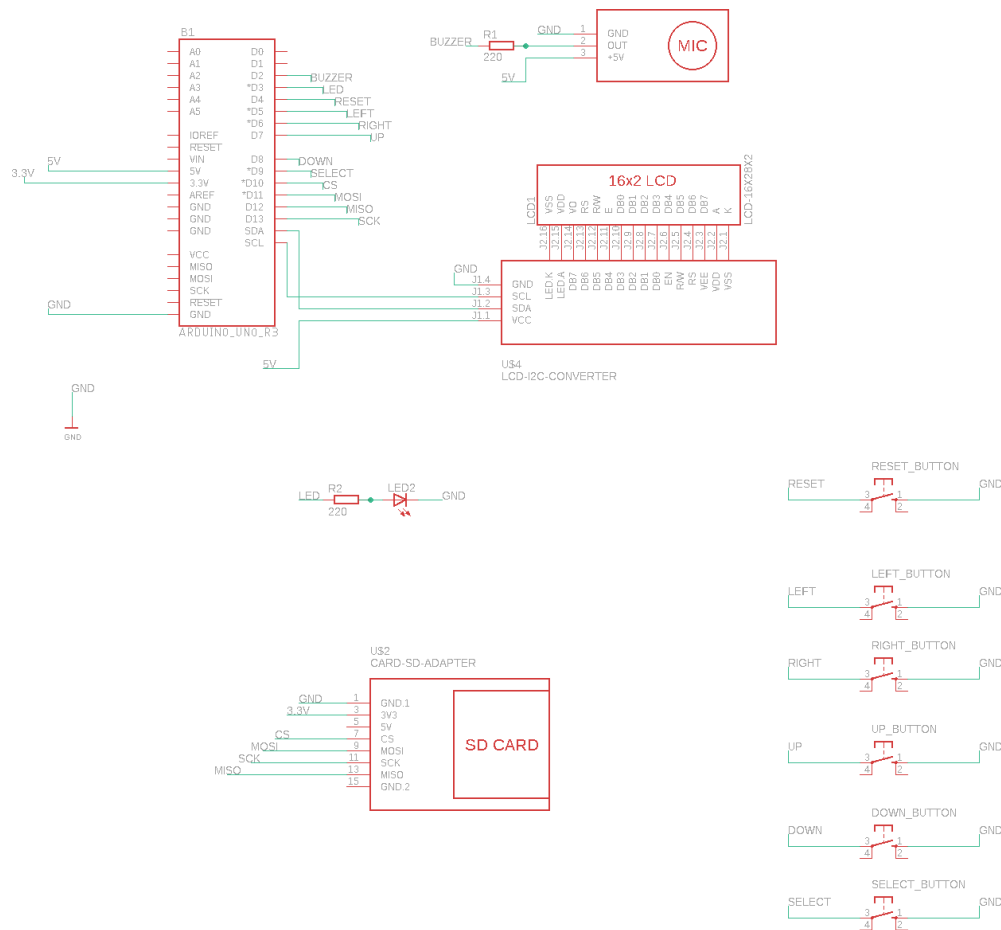
## Hardware Design

Listă de piese:

- placă arduino UNO
- breadboard
- fire

- rezistoare (limitarea curentului prin LED și Buzzer)
- card micro SD ("bază de date" din care se extrage aleator cuvântul) + adaptor microSD-SD + modulul corespunzător
- I2C LCD display
- 6 butoane (RESET, UP & DOWN, LEFT & RIGHT, SELECT)
- buzzer
- LED

Schema electrică:



## Software Design

Am dezvoltat proiectul în Arduino IDE, folosind ca biblioteci auxiliare:

- [SDFat](#) pentru extragerea unui cuvânt aleator din unul din cele două fișiere, en\_words.txt și ro\_words.txt aflate pe card
- [LiquidCrystal\\_I2C](#) pentru afișarea textului pe display

Structurile de date de menționat sunt

- **Button** - reținerea pinului, a stării curente (apăsăat - LOW, neapăsăat - HIGH) și a stării precedente. Motivul memorării stărilor este pentru evitarea executării unor instrucțiuni repetate la apăsarea continuă a unui buton
- Vectorul **triedMasks[26]** reține, sub forma unor măști de biți **pozițiile posibile** pentru fiecare

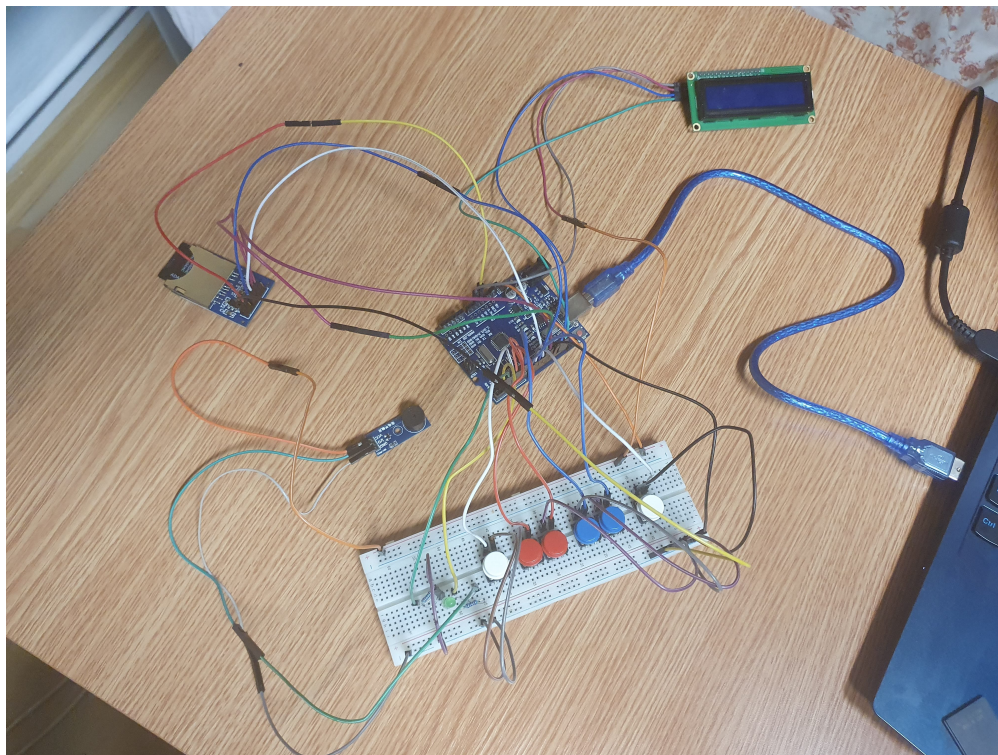
literă. La fiecare încercare se poate modifica masca unei litere. O idee pe care am avut-o inițial a fost ca prin acest array să sugerez jucătorului ce poziții mai sunt valabile pentru literele încercate, dar am renunțat pentru că am crezut că detaliile ar fi fost prea multe pentru un rând de display. Așadar, folosesc măștile doar pentru a deduce ce litere apar în cuvânt, dar pe altă poziție. În mod asemănător, `guessedMask` reprezintă o mască de 5 biți, în care fiecare bit corespunde o **poziție corect ghicită**. Am folosit variabila pentru păstrarea literelor ghicite la fiecare încercare, astfel încât utilizatorul să nu fie nevoit să treacă din nou prin literele deja aflate.

- Array-ul de cuvinte **words** - de-a lungul dezvoltării proiectului, am întâmpinat erori mai ales la prelucrarea datelor de pe card, așa că pun la dispoziție și un vector de cuvinte scurt (memoria este limitată, fiind un Arduino UNO, iar umplerea ei determină o funcționare necorespunzătoare). Astfel, se poate juca o varianta minimală a jocului și fără un card SD.

Funcțiile importante sunt:

- **extractWord()** + **readCard()** - citirea unui cuvânt aleator din unul din fișiere, în funcție de limba menționată de flag-ul `isEnglish`
- **getRandom** - extragerea unei variabile aleatoare, cu seed-ul pseudo-aleator furnizat de timpul de la rularea programului.
- Cum seed-ul nu este constant, iar numărul cuvintelor este la nivelul sutelor pentru română și miilor pentru engleză, termenii vor semăna foarte rar între două jocuri diferite. În cazul unei erori la utilizarea cardului, se va prelua un cuvânt din vectorul hard-codat `words`
- **processCard()** - testarea cardului pentru vizualizarea fișierelor
- **lcdPrintGuess()** - afișarea informațiilor relevante pe ecran, include și păstrarea literelor ghicite de user în cuvânt după selectarea acestuia
- **resetGame()** - reinițializarea parametrilor pentru începerea unui nou joc
- **playSong()** - redarea unei melodii pentru terminarea jocului, funcție modificată din git-ul Arduino Songs (relevant este și header-ul `song.h` din proiect)
- funcții de verificare ale apăsării unui buton (**checkReset()**, **checkButton(Button)**)

## Rezultate Obținute



[Video](#)

## Concluzii

Consider că proiectul este util pentru că

- Am reușit să îmi îmbunătățesc cunoștințele de Arduino UNO și să caut modulele potrivite în funcție de proiect
- Wordle este un joc plăcut și popular, iar adaptarea sa într-o formă “palpabilă” a fost o experiență interesantă
- Cu un astfel de proiect, un jucător poate descoperi și cuvinte noi și să își corecteze, eventual, ortografia pentru anumiți termeni.

## Download

[Github](#)

- [Download](#)

## Jurnal

Jurnal

- 2022/03/20 - Achiziționarea unui kit basic de piese (breadboard, butoane, fire, rezistoare)
- 2022/05/01 - Crearea paginii de OCW
- 2022/05/18 - Testarea pieselor și realizarea unui cod minimal, bazat mai mult pe Serial print
- 2022/05/24 - Următoarele piese - I2C LCD, buzzer, modul SD card
- 2022/05/25 - Implementarea unei variante de joc în care cuvintele sunt salvate în memoria plăcuței
- 2022/05/27 - Testarea bibliotecilor de SD Card și de redare de melodii prin buzzer.
- 2022/05/28 - Implementarea îmbunătățită a jocului, în care cuvintele nu mai sunt preluate din plăcuță ci din cardul de memorie
- 2022/05/29 - Completarea paginii de OCW

## Bibliografie/Resurse

Resurse software:

- [Jocul original](#)
- [Indicații pentru rezolvarea unei erori la utilizarea bibliotecii SDFat](#)
- [Programul recomandat pentru formatarea cardului](#)
- [Listă cuvinte română](#)
- [listă cuvinte engleză](#)
- [Melodia redată în cazul finalizării cu succes a jocului](#)

Resurse hardware (conectarea modulelor):

- [Buzzer](#)
- [Card](#)

[Export to PDF](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2022/cstan/1002>



Last update: **2023/03/15 14:49**