

# Crack the code

## Introducere

Proiectul este un joc de tip Crack the Code. Acesta consta din ghicirea unui cod numeric de 4 cifre generat aleatoriu de calculator. Jocul are doua dificultati: easy si hard, diferenta dintre ele constand in cantitatea de informatii pe care o primeste jucatorul ca raspuns. Astfel, jucatorul va incerca diferite grupuri de 4 cifre si va primi ca raspuns numarul de cifre care sunt corecte pe pozitia corecta.

De exemplu:

Daca codul generat este 4921 si jucatorul introduce 3912, o singura cifra este corecta pe pozitia corecta (adica cifra 9)

Scopul este ca jucatorul sa ajunga la valoarea generata initial intr-un timp cat mai scurt.

## Descriere generala

Jucatorul se va folosi de un numpad pentru a tasta cifrele dorite, buzzer-ul emitand un semnal sonor pentru fiecare cifra tastata. Acestea vor aparea pe un ecran de tip LCD pentru a vedea posibilele erori de tastare si a le putea remedia. Pentru modificarea valorii tastate, jucatorul se va folosi de butonele stanga (\*) dreapta (#) pentru a ajunge la cifra gresita si va apasa o alta cifra pentru a o suprascrie.

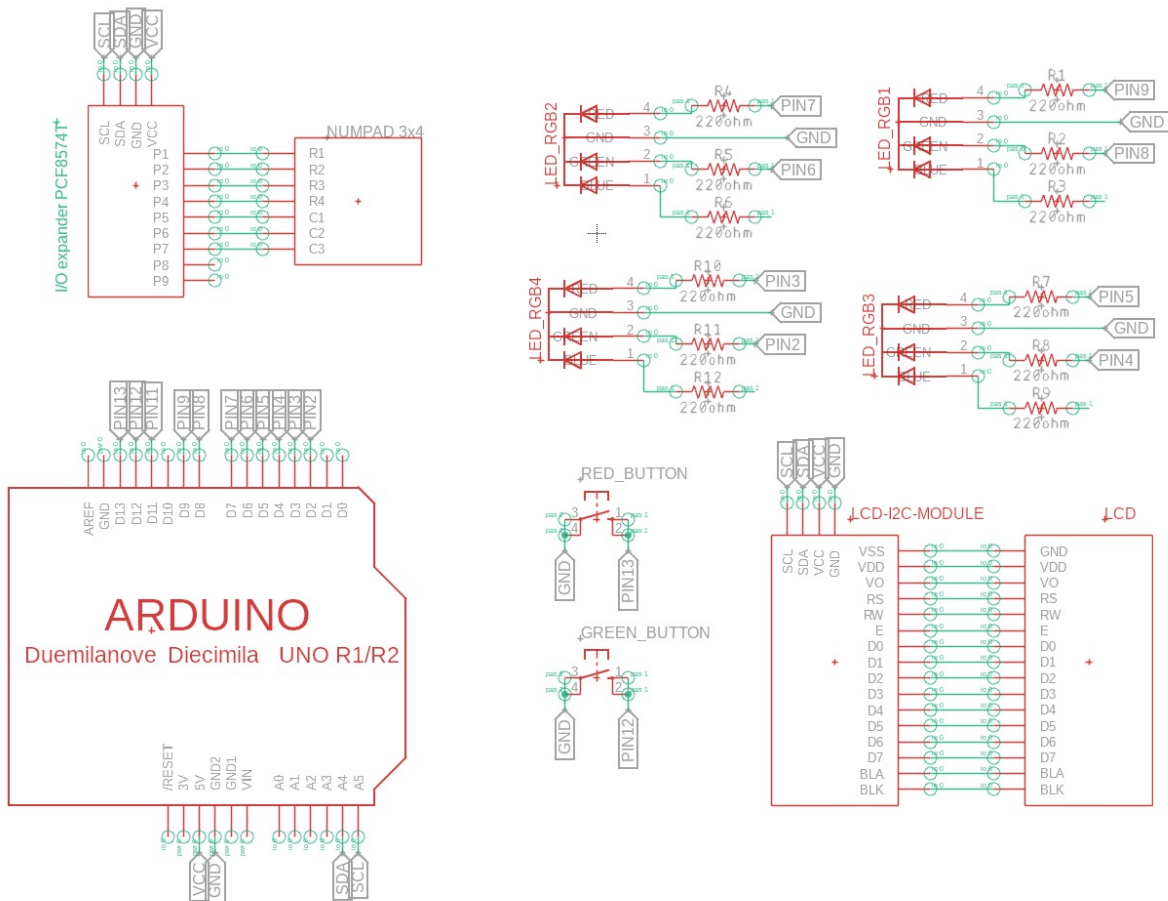
Dupa ce jucatorul va apasa pe butonul verde cu rol de "submit", jocul va transmite raspunsul, aprinzand ledurile RGB cu verde pentru fiecare cifra corecta pe pozitia corecta si cu rosu in caz contrar. De aici jucatorul poate apasa pe butonul verde pentru a mai introduce un cod sau pe butonul rosu pentru a renunta, caz in care codul va fi afisat.



## Hardware Design

Lista de piese principale:

- placuta arduino uno R3
- 4 leduri RGB
- numpad
- ecran LCD
- un buton de submit
- un buton de reset joc
- buzzer



## Software Design

Descrierea codului aplicației (firmware):

- Mediu de dezvoltare este Arduino IDE.
- Librariile include sunt Keypad\_I2C.h si LiquidCrystal\_I2C.h pentru simplificarea utilizarii LCD-ului si al Numpad-ului cu ajutorul protocolului I2C.

## Organizarea Codului

Structuri folosite:

- structura pentru led-uri contine doua campuri: redPin - numarul pinului pentru culoarea rosie si greenPin - pinul pentru culoareaverde. Pinul albastru va ramane in aer pentru ca nu am nevoie de aceasta culoare.
- enum pentru starile prin care va trece jocul: MENU, CHOOSE\_DIFF, GAME, EASY\_CHECK\_CODE, HARD\_CHECK\_CODE, FINAL, CHOOSE\_GIVE\_UP, FINAL\_GIVE\_UP
- array pentru codul generat de calculator (code[4])
- array pentru codul introdus de jucator (code\_input[4])

Funcții implementate standard:

- `setup()`: initializarea pinilor ca INPUT sau OUTPUT
- `loop()`: apelează funcțiile specifice pentru state-ul în care se află jocul. Va începe cu MENU.

Funcții implementate pentru state-uri:

- `menu()`: va printa numele jocului și trecerea spre următorul state, va aștepta o apăsare de buton verde pentru a trece la state-ul CHOOSE\_DIFF.
- `choose_diff()`: va printa opțiunile de dificultate "easy" și "hard". Jucătorul poate alege navigând stanga prin apăsarea (\*) și dreapta prin apăsarea (#). Pentru a confirma alegerea, se așteaptă apăsarea butonului verde. Pentru a se întoarce în state-ul de pornire se va apăsa butonul roșu.
- `game()`: va procesa tastele apăstate pe numpad de către jucător și va verifica corectitudinea acestora (`correct_code[i] = 1`, dacă este corectă și `correct_code[i] = -1`, dacă este greșită). Fiecare apăsare va genera un sunet prin buzzer. Jucătorul poate adăuga cifrele în ce ordine dorește, pentru a face acest lucru va muta cursorul stanga(\*) dreapta(#). Când o cifră este selectată, va palpăi și dacă jucătorul apăsa o altă tastă, cifră va fi suprascrisă. Pentru a trimite codul, jucătorul va apăsa butonul verde. În situația în care jucătorul nu a tastat toate cele 4 cifre, buzzer-ul va scoate un sunet și jocul nu va progresa. Dacă jucătorul vrea să renunțe la jocul curent, poate apăsa pe butonul roșu.
- `easy_check_game()`: va calcula câte cifre sunt corecte și va aprinde atâtea leduri verzi, restul roșii. Va scrie pe lcd câte cifre au fost corecte și le va arăta poziția în cod. Dacă jucătorul vrea să renunțe la jocul curent, poate apăsa pe butonul roșu.
- `hard_check_game()`: va calcula câte cifre sunt corecte și va aprinde atâtea leduri verzi, restul roșii. Va scrie pe lcd câte cifre au fost corecte. Dacă jucătorul vrea să renunțe la jocul curent, poate apăsa pe butonul roșu.
- `final()`: va fi apelat dacă toate cifrele sunt corecte. Va printa un mesaj de încurajare și va întreba jucătorul dacă vrea să joace din nou. La apăsarea butonului verde, se va întoarce în state-ul de menu.
- `choose_give_up()`: va fi apelată dacă jucătorul vrea să renunțe prin apăsarea butonului roșu. Jucătorul poate alege între "yes" și "no" prin mutarea cursorului stanga(\*) dreapta(#). Pentru a confirma decizia, jucătorul trebuie să apese butonul verde.
- `final_give_up()`: va scrie pe lcd care este codul corect și va întreba jucătorul dacă vrea să joace din nou. Pentru a face acest lucru, jucătorul trebuie să apese butonul verde, iar jocul se va întoarce în state-ul CHOOSE\_DIFF.

Funcții ajutoare:

- `setColor(int redValue, int greenValue, int blueValue, RGBLed led)`: Valorile pentru roșu, verde și albastru și ledul pe care să îl aprindă.
- `beep(int ms, uint8_t freq)`: Numărul de milisecunde și frecvența pentru buzzer
- `get_button_input(int buttonPin)`: debounce button
- `pressed_one_time(int buttonPin)`: va verifica dacă butonul a fost apăsat și va returna true când butonul nu mai este eliberat.

## Rezultate Obținute

Rezultatele obținute:

GitHub: <https://github.com/AndreeaDrehuta/CrackTheCode>

Youtube: <https://youtu.be/m9wbOW-NLac>

## Concluzii

O experienta interesanta

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).

**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

- Update schema fusion
- Adaugare Software Design
- Adaugare github
- Adaugare youtube

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/mdinica/andreea.drehuta>



Last update: **2024/05/26 21:28**