

Guessing Colors Game

Nume: Rădulescu Matei

Grupă: 333CA

Introducere

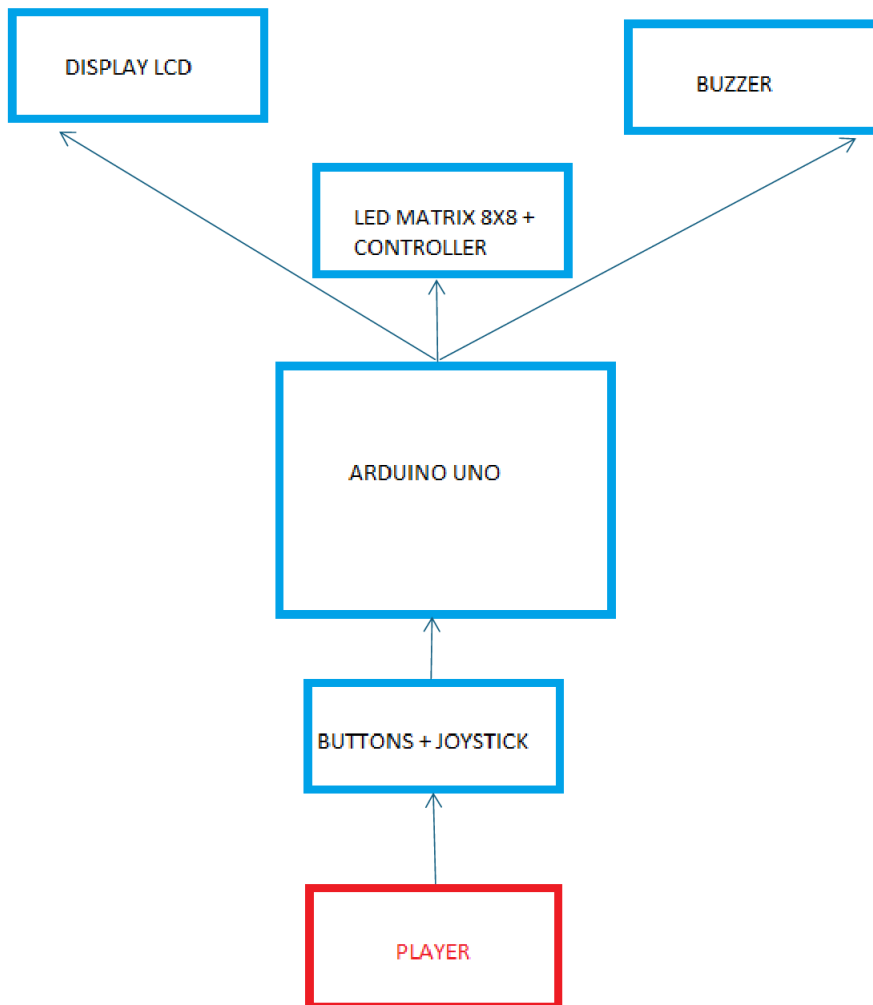
Joc cu LED matrix: pe matrice este afișată o selecție de puncte. Jucătorul trebuie apoi să ghicească poziția acestora. Există butoane pentru start, selectarea dificultății, un joystick pentru poziționare și un buton back pentru mutările greșite. Jocul include și sonorizare cu buzzere pentru cazurile de ghicire corectă și greșită.

Descriere generală

Un proiect Arduino de joc cu LED matrix implică utilizarea unei matrice de LED-uri pentru a afișa o varietate de cuburi. Scopul acestui joc este ca jucătorul să ghicească poziția acestora.

La pornirea jocului, matricea de LED-uri afișează o selecție aleatorie de cuburi timp de 5 secunde, permițând jucătorului să memoreze pozițiile acestora. După ce cuburile dispar de pe matrice, jucătorul preia controlul folosind joystick-ul pentru a muta un cursor pe matrice. Prin intermediul butonului de pe joystick, jucătorul poate alege să plaseze un cub într-o anumită poziție. În cazul în care jucătorul face o greșeală, poate folosi butonul roșu pentru a reveni la o matrice goală și a ghici din nou.

Jocul continuă până când toate pozițiile sunt ghicite corect. La final, pentru a semnaliza succesul sau eșecul jucătorului, buzzerul emite o melodie de câștig sau pierdere.



Hardware Design

Listă piese:

Arduino UNO R4 MINIMA × 1 - Acționează ca unitatea centrală de control pentru toate componentele.

8 X 8 LED Matrix × 1 : Conenctată la GND si 5V și 3 pini digitali pentru control. Reprezintă mediul pentru joc, pe care apar luminile ce trebuie ghicite.

LCD display cu I2C x 1: Controlerul I2C este conectat la GND, 5V. Iar SDA și SCL la pinul respectiv al plăcuței Arduino pentru afișare.Display pentru meniul jocului si scor.

Joystick x 1 : Conectat la GND, 5V, doi pini analogici pentru a citi axele OX și OY și un pin digital pentru controlul său ca buton. Controlează selecția celulei ce va fi ghicită.

Button × 3: Conectate la un pin analogic cu o rezistență de 10k pentru a citi corect starea și la 5V în cealaltă stare. Au roluri de start, de selecție dificultate și de resetare a ghicirii curente.

Rezistenta 10K x 3: Folosite pentru butoane.

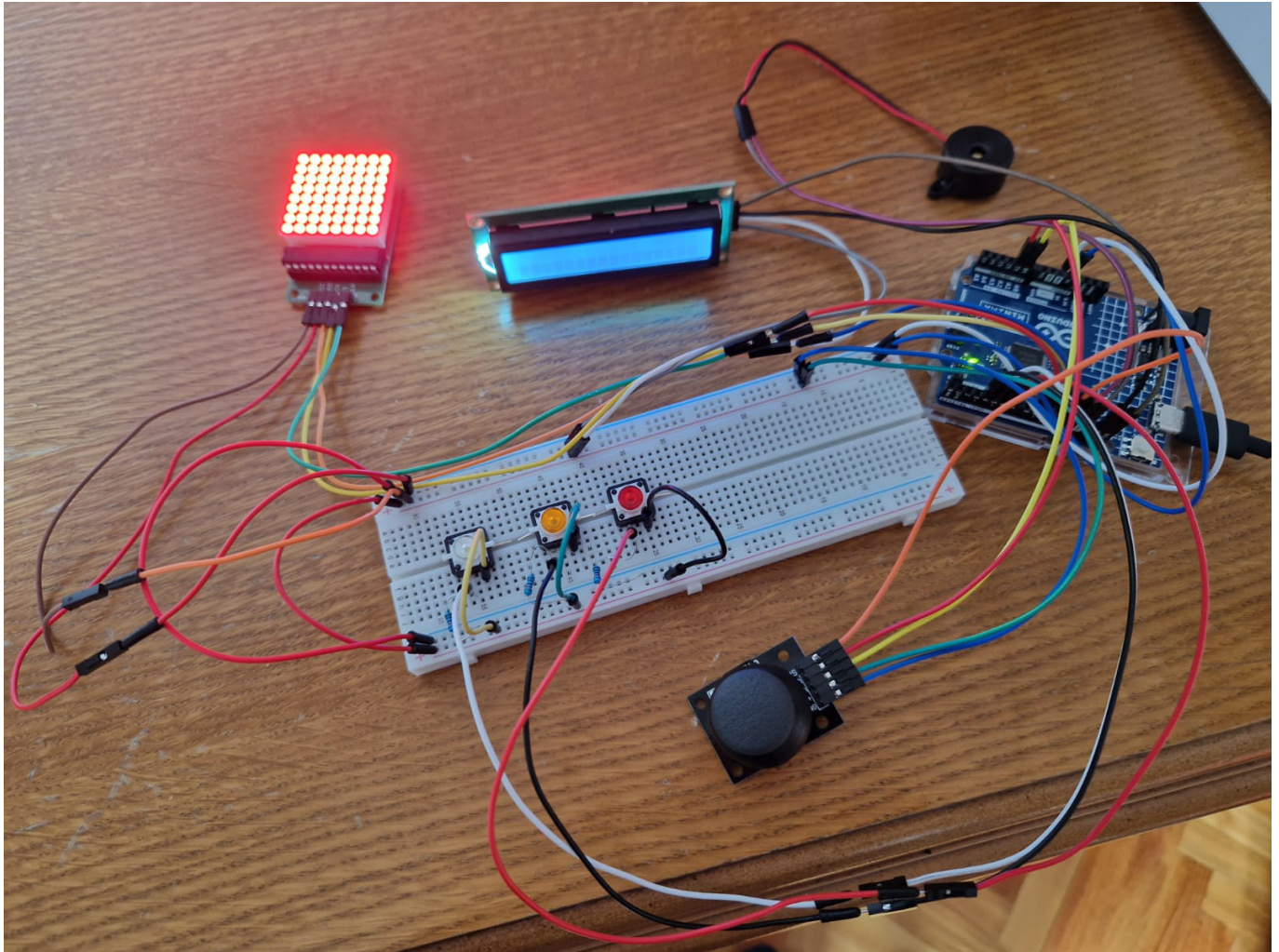
Buzzer x 1: Conectat la GND și un pin digital pentru control, face sunete pentru ghicire corectă și incorectă.

Pentru a implementa hardware-ul jocului cu LED matrix folosind Arduino UNO, matricea 8×8 de LED-uri a fost conectată la pinii digitali ai Arduino-ului pentru a controla rândurile și coloanele. LCD-ul cu I2C a fost conectat la pinii de comunicație SDA SI SCL pentru a afișa mesaje. Joystick-ul a fost conectat la pinii analogici pentru axele X și Y, iar butonul joystick-ului la un pin digital. Cele trei butoane au fost conectate la pini digitali, fiecare cu o rezistență de 10K pentru pull-down. Buzzerul a fost conectat la un pin digital PWM pentru a emite sunete. Aceste conexiuni permit controlul matricei LED, afișarea informațiilor pe LCD, primirea inputului de la joystick și butoane, și emiterea de sunete, completând astfel funcționalitatea jocului.

Imaginea urmatoare contine schema tinkercad a circuitului, joystick-ul si led-ul de matrici 8×8 au fost inlocuite cu NeoPixel Jewel deoarece nu există în aplicație. Pot arată cablajul ,deoarece au același număr de 5 intrări, cel de jos reprezintă joystick-ul, iar celălalt matricea. Plăcuța folosită este înlocuită cu un R3 clasic, identice în pini.



Imaginea următoare conține implementarea hardware, ecranul matrix-ul și buzzer-ul sunt active, joystick-ul și butoanele necesită implementare software pentru funcționalitate:



Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare: Arduino IDE
- librării: LedControl.h și LiquidCrystal_I2C.h
- algoritmi: mișcarea cu joystick-ul pe un matrix LED, blink pentru recunoașterea selecției.

Acest cod este destinat unui proiect Arduino care combină o matrice LED, un afișaj LCD, un joystick și mai multe butoane pentru a crea un joc interactiv. În esență, jucătorul trebuie să recreeze un model afișat pe matricea LED folosind joystick-ul și butoanele. Codul include două biblioteci: "LedControl.h" pentru controlul matricei LED și "LiquidCrystal_I2C.h" pentru gestionarea afișajului LCD.

Pinii pentru controlul matricei LED sunt definiți astfel: pinul de date (dinPin) este conectat la pinul 11, pinul de ceas (clockPin) la pinul 10 și pinul de încărcare (loadPin) la pinul 9. Joystick-ul este conectat la pinii analogici A0 și A1 pentru axele X și Y, iar butonul joystick-ului este conectat la pinul 12. Mai sunt trei butoane suplimentare, fiecare conectat la pinii 5 (pinBw), 6 (pinBo) și 7 (pinBr). De asemenea, un buzzer este conectat la pinul 13 pentru a reda sunete la sfârșitul jocului.

Variabilele de stare ale jocului includ booleeni pentru a urmări stările butoanelor, începutul jocului, selecția dificultății, sfârșitul jocului și victoria. Variabila count urmărește numărul de mutări efectuate, iar selectedDifficulty urmărește nivelul de dificultate selectat, care poate varia între 1 și 5. Obiectele lcd și lc sunt folosite pentru a controla afișajul LCD și matricea LED.

Sunt definite trei matrice de 8×8 pentru a reprezenta diferite stări ale jocului: EmptyMatrix pentru a goli afișajul, CurrentMatrix pentru a ține modelul care trebuie reprodus și BuiltMatrix pentru a urmări modelul curent al jucătorului.

Funcția de configurare (setup()) inițializează afișajul LCD, matricea LED și pinii necesari, precum și un random seed pentru a genera matrice random. În bucla principală (loop()), jocul trece prin mai multe etape.

În prima etapă, afișajul LCD arată mesajul "Press Start", iar jocul așteaptă apăsarea butonului alb pentru a începe. După începerea jocului, jucătorul selectează dificultatea prin apăsarea butonului portocaliu, incrementând nivelul de dificultate de la 1 la 5 și confirmând selecția cu butonul alb. La confirmare, se generează un model aleatoriu în CurrentMatrix, care este afișat pe matricea LED pentru o scurtă perioadă.

În etapa de joc, dacă jocul s-a încheiat, afișajul LCD arată rezultatul (victorie sau înfrângere) și solicită repornirea jocului. Dacă jocul nu s-a terminat, afișajul indică jucătorului să apese butonul roșu pentru a resete matricea de construită dacă dorește.

Jucătorul poate plasa un cub prin apăsarea joystick-ului, cu care se și mișcă pe matrice. Intervalul de mișcare al joystick-ului este între 200 și 500.

Rezultate Obținute

Am realizat un joc funcțional, montat și scris în întregime de mine, reușind să fac ceva ce inițial părea inabordabil.

Download

<https://github.com/Radulescu-Matei/Guessing-Game>

Bibliografie/Resurse

LedControl.h : <https://wayoda.github.io/LedControl/>

LiquidCrystal_I2C.h : https://github.com/johnrickman/LiquidCrystal_I2C

Arduino R4 minima: <https://docs.arduino.cc/hardware/uno-r4-minima/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/radulescu.matei>



Last update: **2024/05/27 10:52**