

Smart chessboard

Introducere

- Proiectul meu își propune implementarea unei table de șah inteligentă, capabile să detecteze mutările pieselor și să evedențieze mutările valide pentru piesa selectată.
- În plus, sistemul va integra un display LCD dedicat afișării timpului rămas pentru fiecare jucător, precum și semnalării erorilor în cazul efectuării unor mutări invalide.
- Ideea proiectului provine din experiența personală. Jucând șah cu fratele meu mai mic atât pe platforma chess.com, cât și pe o tablă fizică, am observat cât de folositoare este funcționalitatea de a vedea toate mutările posibile ale unei piese. Așadar, îmi propun să fac și o implementare fizică, cu scopul de a îmbunătăți experiența de joc și de a face șahul mai accesibil și interactiv..

Descriere generală

- Circuitul propus are la bază un microcontroler ESP32, care acționează ca unitate centrală de control și coordonează interacțiunea dintre toate componentele sistemului.
- Tabla de șah este acoperită de o matrice de switch-uri reed, câte unul pentru fiecare căsuță, pentru a detecta poziția pieselor. Pentru a economisi pini de pe ESP32, această matrice este controlată cu ajutorul unui registru de deplasare 74HC595.
- ESP32 interacționează cu o bandă adresabilă de led-uri, utilizate pentru a evedenția mutările valide ale pieselor selectate.
- Se utilizează două butoane pentru a schimba jucătorul activ.
- În plus, este folosit un display pentru a afișa mesaje relevante(timpul rămas, atenționare în cazul unei erori).

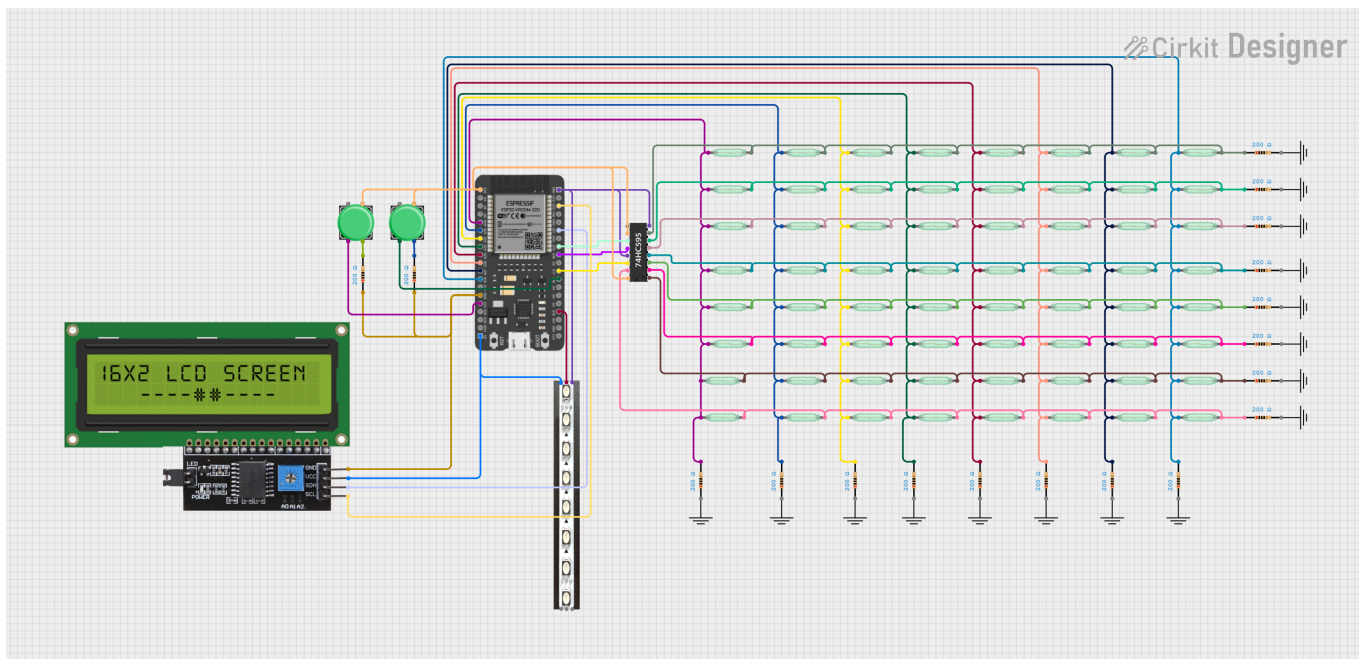


Hardware Design

Componente necesare:

- [microcontroler ESP32](#)
- 64 x [switch-uri reed magnetic](#)
- [LCD 1602 cu Interfata I2C si Backlight Albastru](#)
- 11 x [Banda programabila led-uri RGB](#)
- 1 x 74HC595N
- 2 x [Buton](#)
- Rezistențe, condensatoare, fire
- Sursă de alimentare - TBA

Diagramă circuit:



Explicare cablaj

- 74HC595N

Pin	Legătură ESP32	Notă
DS	GPIO17	Date (serial input)
SHCP	GPIO18	Ceas (shift clock)
STCP	GPIO19	Latch (stochează ieșirea)

Ieșirile 00-07 sunt conectate la matricea de switch-uri, fiecare ieșire corespunzând unui rând.

- Bandă led

Pin	Legătură ESP32	Notă
DIN	GPIO15	Semnal date LED (neopixel)

Banda led are nevoie de alimentare VCC de 5V.

- LCD I2C

Pin	Legătură ESP32	Notă
SDA	GPIO21	I2C - date
SCL	GPIO22	I2C - ceas

- Matrice switch-uri reed

Pin	Legătură ESP32	Notă
COL0	GPIO34	doar input
COL1	GPIO35	doar input
COL2	GPIO32	
COL3	GPIO33	

COL4	GPIO25	
COL5	GPIO26	
COL6	GPIO27	
COL7	GPIO14	

- Butoane jucători

Pin	Legătură ESP32	Notă
WHITE_BTN	GPIO13	Înterupere + rezistență de pull-down
BLACK_BTN	GPIO16	Înterupere + rezistență de pull-down

Realizare tablă de șah

- [Bandă led pe suportul inferior al tablei](#)
- [Testare funcționalitate 74HC595N](#)
- [Suport reed-uri](#)

Software Design

- mediu de dezvoltare: Arduino IDE
- librării : Adafruit_NeoPixel.h pentru programarea bandei led adresabile, LiquidCrystal_I2C.h și Wire.h pentru LCD
- noțiuni de laborator folosite: I2C, timer, întreruperi
- github repo: <https://github.com/alex2004-l/smart-chessboard>

Implementare

- Aplicația este structurată ca o mașină de stări: BEGINNING, READY, PICK, MOVE, CAPTURE, END;
- Pentru a urmări pozițiile pieselor pe tablă se folosesc doi vectori: board și previous_board.
- press_button_white() și press_button_black() sunt ISR-uri (Interrupt Service Routines) care schimbă starea jocului când jucătorul apasă butonul.
- Detectarea mutărilor pieselor prin comparația board vs previous_board
- Validarea mutărilor pe baza tipului de piesă (folosind funcțiile din moves.h)
- Implementarea unui sistem de cronometru (folosind hardware timer)
- Detectarea capturilor și actualizarea stării jocului (CAPTURE, END)
- Afișarea mutărilor și timpului pe LCD
- Controlul LED-urilor în funcție de mutările făcute sau starea curentă

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuieți să le implementați


- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

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

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

[Export to PDF](#)

- <https://forum.arduino.cc/t/74hc595-shift-register-adjustments-for-esp32/1164766>
- <https://docs.arduino.cc/tutorials/communication/guide-to-shift-out/>
- https://www.youtube.com/watch?v=D_QBFIQk-o&ab_channel=GreatScott%21

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/iotelea/alexandra.lache0802>



Last update: **2025/05/26 19:55**