

Electronic Chess

Introducere

Proiectul Electronic Chess își propune realizarea unei table de șah electronice capabile să detecteze și să înregistreze automat mutările efectuate de jucători în timpul unei partide. Scopul principal este digitalizarea jocului fizic de șah pentru a facilita analiza ulterioară a jocurilor, oferind astfel o unealtă utilă pentru pasionații de șah care vor să își îmbunătățească strategia și stilul de joc.

Această platformă vine în sprijinul jucătorilor amatori și profesioniști care doresc să revadă și să analizeze partidele jucate și să identifice greșelile comise.

Descriere generală

Laboratoare folosite: GPIO, Întreruperi, Timere, I2C.

Schema bloc:



Module hardware:

- Tabla de șah fizică: 64 de senzori Reed Switch (câte unul pentru fiecare pătrățel), care detectează prezența pieselor.
- Multiplexoare: utilizate pentru a reduce numărul de pini necesari pe microcontroller pentru citirea celor 64 de senzori.
- ESP32: microcontroller central ce gestionează toate operațiunile (citirea mutărilor, afișarea pe ecran, comunicarea cu aplicația android).
- LCD și butoane: interfață de utilizator pentru afișarea informațiilor și controlul meniului.

Module Software:

- ESP32: citirea senzorilor, detectarea mutărilor, comunicarea cu aplicația Android și afișarea informațiilor pe LCD.

- Android: afișarea tablei în format digital, transmiterea controlului jocului către ESP32, recepționarea stării tablei de la ESP32.

Interacțiune între module:

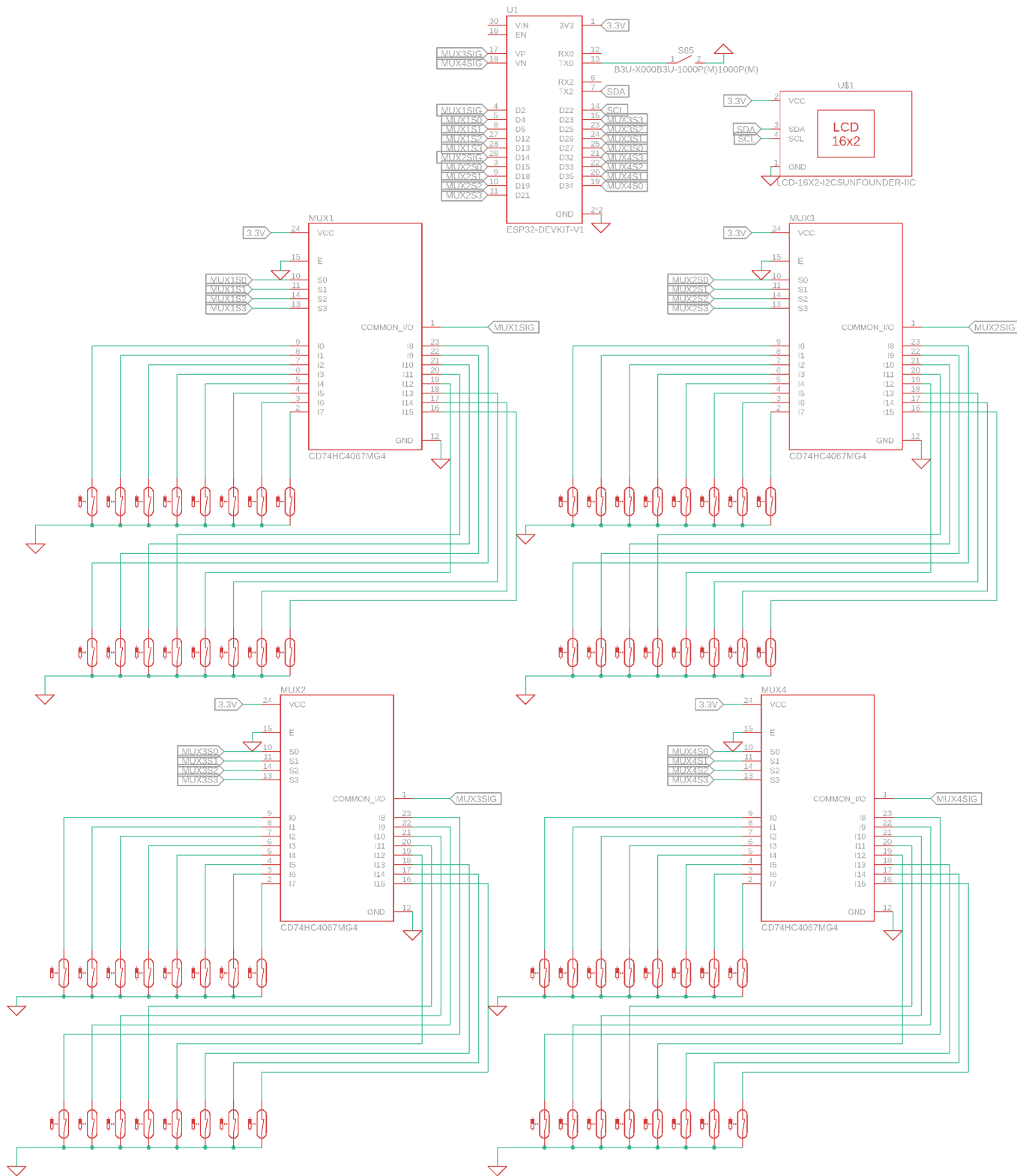
- Când jucătorul mută o piesă, senzorii Reed detectează poziția inițială și finală a mutării.
- Jucătorul apasă pe buton pentru a confirma mutarea. Starea tablei este salvată de ESP32.
- ESP32 transmite starea tablei către aplicația Android.
- LCD-ul afișează timpul rămas pentru fiecare jucător.

Hardware Design

Listă de piese:

- ESP32
- 64x Reed Switch
- 4x Multiplexor 16:1 cd74hc4067
- LCD cu modul I2C
- Buton
- Sursă 3.3V
- Breadboard
- Fire de legatură

Schema electrică:



Multiplexor 16:1 cd74hc4067

- Pentru a reduce drastic numărul de pini necesari la ESP32 pentru scanarea celor 64 de senzori.
- Multiplexoarele sunt conectate similar. Mai jos este un exemplu pentru unul dintre ele.
- Pinii I0 - I15 (canalele multiplexorului) sunt conectati la switch-urile Reed.

Pin	Pin ESP32	Descriere
VCC	3V3	Alimentare
GND	GND	Ground

EN	GND	Enambel mux
SIG	GPIO4	Outputul MUX-ului
S0	GPIO5	Selectie
S1	GPIO8	Selectie
S2	GPIO27	Selectie
S3	GPIO28	Selectie

LCD Cu modul I2C

- Ecranul va fi folosit pentru afișarea timpului rămas pentru fiecare jucător.

Pin	Pin ESP32	Descriere
VCC	3V3	Alimentare ecran
GND	GND	GND pentru ecran
SDA	GPIO7	Linia de date
SCL	GPIO14	Linia ceasului

Software Design

Mediu de dezvoltare: PlatformIO, Android Studio

Biblioteci:

- Arduino.h: Biblioteca de bază pentru programarea microcontroller-ului.
- LiquidCrystal_I2C: pentru controlul LCD-ului
- BluetoothSerial.h: pentru comunicarea cu aplicația Android.
- Wire.h: pentru interfațarea I2C cu LCD-ul.
- Ticker.h: pentru gestionarea temporizărilor.

Funcționalități din laborator:

- GPIO: Conectarea MUX-urilor.
- Întreruperi: Pentru înregistrarea unei mutări odată cu apăsarea butonului.
- Timere: pentru ceasul folosit în jocul de șah.
- I2C: Pentru conectarea ecranului LCD.

Funcționalități implementate pe ESP32:

Inițializarea componentelor:

- LCD, multiplexoare, butoane, Bluetooth.
- Configurarea întreruperii pentru butonul de confirmare mutare.

Scanarea tablei:

- Prin multiplexoare 16:1, se scanează în mod secvențial toți cei 64 de senzori Reed.
- Se detectează pozițiile în care s-au făcut modificări (piese ridicate sau plasate).

Detectarea mutărilor:

- Se compară starea anterioară și cea curentă a tablei.
- Se extrage perechea (poziție inițială, poziție finală) care corespunde mutării efectuate.

Confirmarea mutării:

- Mutarea este validată și înregistrată doar după apăsarea butonului fizic.
- Se activează o întrerupere pe pinul butonului, care declanșează salvarea mutării.

Comunicarea cu aplicația Android:

- Starea actualizată a tablei este trimisă prin Bluetooth către aplicație.
- ESP32 ascultă și comenzile primite (start, reset joc, schimbare mod, etc.).

Afișarea informațiilor pe LCD:

- Timpul rămas pentru fiecare jucător este afișat și actualizat la fiecare mutare.
- Se pot adăuga mesaje de status sau indicații pentru jucători.

Gestionarea timpului (timer digital):

- Se folosește o funcție periodică pe bază de timer (Ticker) care scade timpul rămas la fiecare secundă.
- Timerul se oprește și se pornește în funcție de mutarea jucătorului.

Funcționalități aplicație Android:

Interfață grafică a tablei:

- Afișează în timp real poziția pieselor pe tabla virtuală.
- Simulează tabla fizică pe ecranul telefonului.

Comunicare Bluetooth:

- Aplicația primește datele de la ESP32 (poziții piesă, mutări, ceas).
- Poate trimite comenzi precum: start, reset joc.

Istoric mutări:

- Se salvează într-o listă vizibilă în aplicație toate mutările efectuate.

Notificări și feedback:

- Notificare final joc (mat, remiză, abandon).

Citire canal multiplexor:

```
int readMuxChannel(int channel) {  
    // Set the address lines (S0 to S3)  
    for (int i = 0; i < 4; i++) {  
        digitalWrite(selectPins[i], (channel >> i) & 0x01);  
    }  
  
    delayMicroseconds(10); // Small delay to allow settling  
  
    // Read the value from the selected MUX channel  
    return digitalRead(sigPin);  
}
```

Timer:

```
volatile int timeRemaining = 0; // Init when game starts  
volatile bool updateLCD = false;  
  
void tick() {  
    if (timeRemaining > 0) {  
        timeRemaining--;  
        updateLCD = true;  
    }  
}  
  
void startTimer(int secunde) {  
    timeRemaining = secunde;  
    timerTicker.attach(1.0, tick);  
}  
  
void stopTimer() {  
    timerTicker.detach();  
    timeRemaining = 0;  
    updateLCD = true;  
}
```

Comunicare prin bluetooth:

```
if (SerialBT.available()) {  
    String mesaj = SerialBT.readStringUntil('\n');  
    mesaj.trim();  
    mesaj.toLowerCase();  
  
    if (mesaj == "start") {  
        startTimer(5 * 60); // 5 minute  
        SerialBT.println("Timer pornit");  
    } else if (mesaj == "reset") {  
        stopTimer();  
        SerialBT.println("Timer oprit/resetat");  
    }  
}
```

```
    } else {  
        SerialBT.println("Comanda necunoscuta");  
    }  
}
```

Rezultate Obținute

Funcționalități implementate cu succes:


- Detectarea mutărilor pe tabla de șah
- Interfață LCD funcțională pentru afișarea timpului
- Mod de joc 1v1 complet operațional
- Salvarea mutărilor în format digital

Limitări:

- Nu se detectează automat promovarea piesei - necesită intervenție de la utilizator

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

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/prj2025/mdinica/alin_ioan.alexandru



Last update: **2025/05/28 11:27**