

## Rareș FOLEA (87478) - Sah electronic

Adresa de e-mail: rares.folea@stud.acs.upb.ro

### Introducere

Proiectul propune o implementare a unui joc de sah electronic, in mod player vs computer, de tipul [Mephisto Chess Challenger](#). Proiectul presupune implementarea unei table de sah controlata direct de microcontroller, prin intermediul careia jucatorul si calculatorul pot interactiona usor, a elementelor de control ale tablei si a implementarii protocolului de comunicare UCI (Universal Chess Interface). De asemenea, este necesara configurarea unei legaturi seriale de comunicare intre microcontroller si calculatorul pe care ruleaza engine-ul de sah.



Mephisto Chess Challenger

### Descriere generala

#### Placa de baza

Placa de baza contine un microcontroller Atmega324PA ce este responsabil de controlul logicii placii de sah si de conexiunea USART catre un dispozitiv pe care ruleaza un engine de sah, comunicand mutarea aleasa de jucator si raspunsul primit de la engine catre placa de sah.

#### Placa de sah

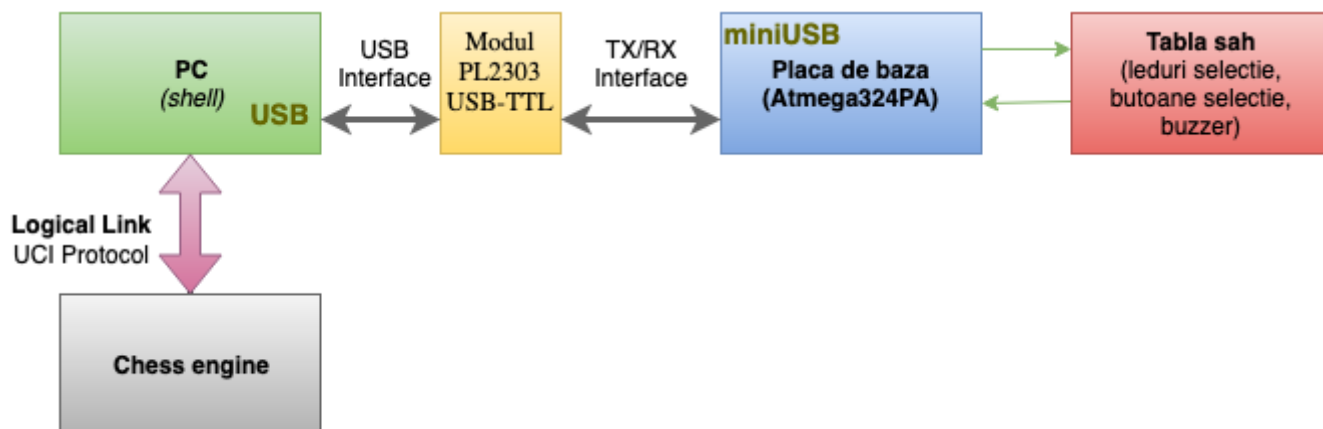
Placa propriu-zisa de sah permite jucatorului sa mute piesele, prin actionarea butoanelor corespunzatoare, iar calculatorului o modalitate de a interactiona cu jucatorul, ilustrand mutarea aleasa prin utilizarea unor leduri de selectie pe linii/coloane. Placa are si un suport LCD text, pentru a preciza mutarea in format UCI si pentru alte informatii referitoare la starea jocului. De asemenea, placa contine si un buzzer, pentru a semnala mutari invalide/ selectie incorecta sau alte erori aparute.

#### Protocolul de comunicare

Universal Chess Interface (UCI) este un protocol de comunicare care permite enginului de sah sa comunice cu interfetele utilizatorilor. Acesta va fi folosit pentru a interschimba mutari intre jucator si calculator, si va rula intern in calculator, care primeste/trimite date de la/catre placa de sah, utilizand un modul de conversie USB-TTL, pe microcontroler fiind configurata o conexiune USART (Universal Synchronous-Asynchronous Receiver/Transmitter), cu parametrii

```
--baudrate 9600 --databits 8 --flow none --stopbits 1 --parity even
```

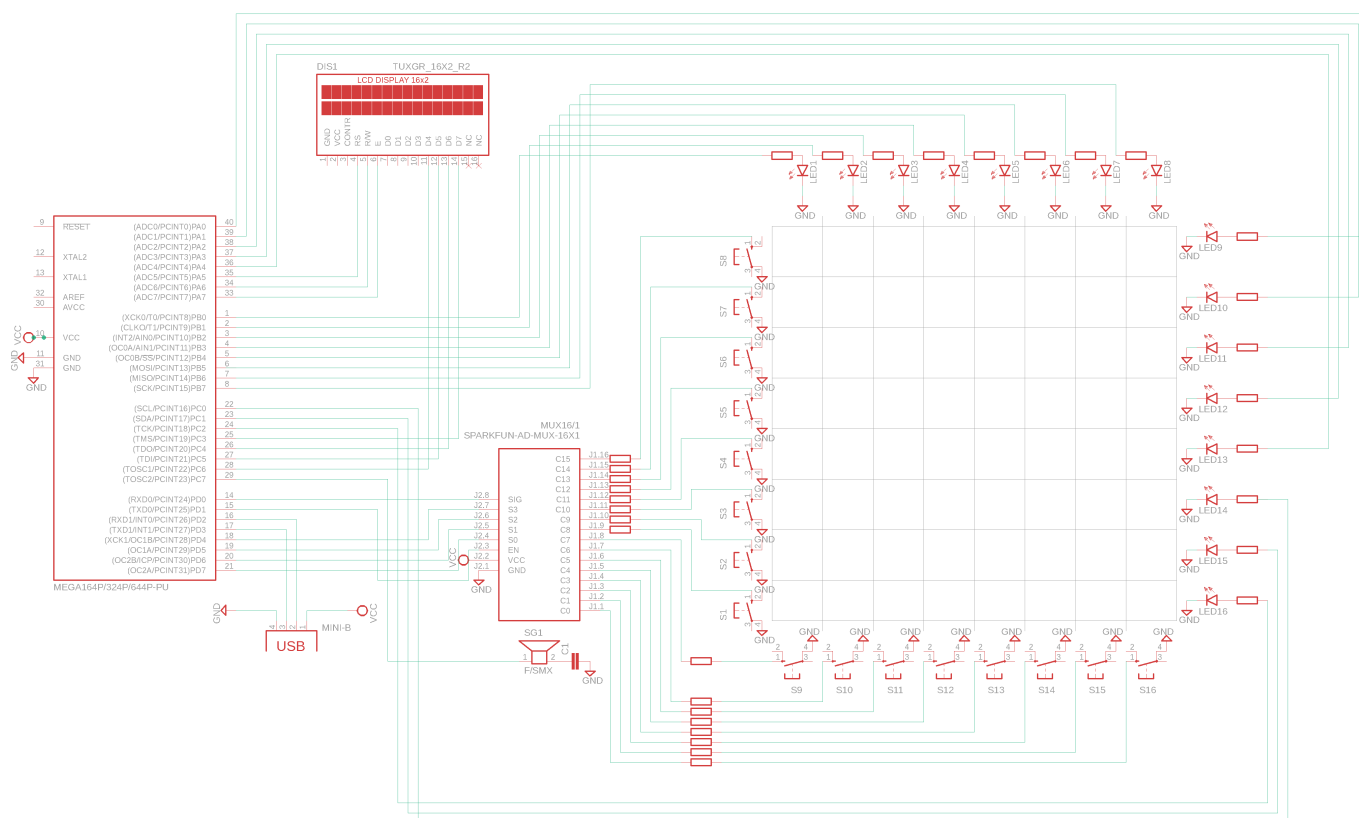
#### Schema bloc



### Lista Piese

- Modul Atmega324PA
- Cablaj de test, simplu placat - 300x400mm
- Rezistor, 0,25 W - 100 - 100 buc
- Cablu, 40 fire, colorat, cu conectori mama-mama - 20 cm
- Cablu, 40 fire, colorat, cu conectori mama-tata - 30 cm
- Intrerupator, fara retinere - 6x6x7 mm - 20 buc
- LED 5 mm, rotund, mat, intermitent, 100 lucsi, rosu, 20 mA - 20 buc
- Modul LCD 1602 cu Backlight Albastru de 3.3 V
- Header de Pini Mama 8p 2.54 mm - 4 buc
- Modul Multiplexor Analogic cu 16 Canale
- Demultiplexor 3/8 - 2 buc
- Buzzer activ
- Modul PL2303 USB-TTL

### Hardware Design



### Software design

Software-ul necesar se imparte in trei mari categorii: gestionarea tablei de sah de catre microcontroller, engine-ul de sah si asigurarea modalitatii de comunicare intre engine-ul de sah si software-ul de gestiune al tablei.

1. Implementarea software pentru Atmega 324PA urmareste simularea unui automat finit cu stari. Am definit 2 clase de stari, starea finala fiind determinata prin produs cartezian intre starile individuale.

- Clasa 1 de stari: Fiecare mutare este descrisa prin 4 apasari de buton, doua pentru selectia liniei si a coloanei casutei sursa, respectiv doua pentru selectia liniei si a coloanei casutei destinatie. Astfel, o mutare, va fi codificata, ca un tuplu [linie\_sursa, coloana\_sursa, linie\_destinatie, coloana\_destinatie]. spre exemplu, mutarea unui pion in deschidere Catalana va fi codificata: d2d4
- Clasa 2 de stari: Jucatorul care trebuie sa mute. Aceasta stare va fi codificata prin 0/1, iar tranzitiile vor fi catre jucatorul opus: 1/0

Ca urmare, vom avea un automat finit cu 8 stari principale, la care adaugam o stare de reset. Din software, se activeaza rezistentele de pull-up pentru porturile care asteapta semnal de la butoane (PA0-7), iar detectia apasarii butonului se face prin verificari secventiale si se verifica valoarea periodic. In cazul in care butonul apasat este cel apasat, se avanseaza cu o tranzitie catre starea urmatoare a mutarii, iar in cazul in care s-a finalizat o mutare, se face trecerea la starea 0 a urmatorul jucator. Implementarea este realizata in C.

2. Engine-ul de sah utilizat este [MisterQueen](#), un engine open-source. Au fost putine modificarile pe care le-am facut, pentru o uniformizare a interfetei de comunicare si dezactivarea suportului de multi-threading, pentru a asigura asteptarea finalizarii operatiilor si doar dupa acest punct se comunica alegerea mutarii catre software-ul intermediar de comunicare cu microcontrollerul. Principala functionalitate este de oferirea de bestmove, pentru un scenariu de joc dat. Interfata de comunicare este asigurata dupa pornirea engine-ului, prin comanda de initializare UCI a pozitiei de joc:

```
position startpos moves d2d4 g8f6 c2c3 c7c5
go
```

Implementarea este realizata in C.

3. Implementearea software-ului ce asigura modalitatii de comunicare intre engine-ul de sah si microcontroller. Implementarea este realizata in bash (Linux), utilizand putty. Pentru ascularea pe /dev/ttyUSB0, se utilizeaza comanda `sudo putty /dev/ttyUSB0 -log test -serial -sercfg 9600,8,e,1,N`. Bucla principala respecta formatul:

```
repeta:
    verifica actualizare
    daca exista actualizare:
        daca este mutarea calculatorului:
            transmite cerere engineului
            asteapta raspuns
            transmite raspuns
```

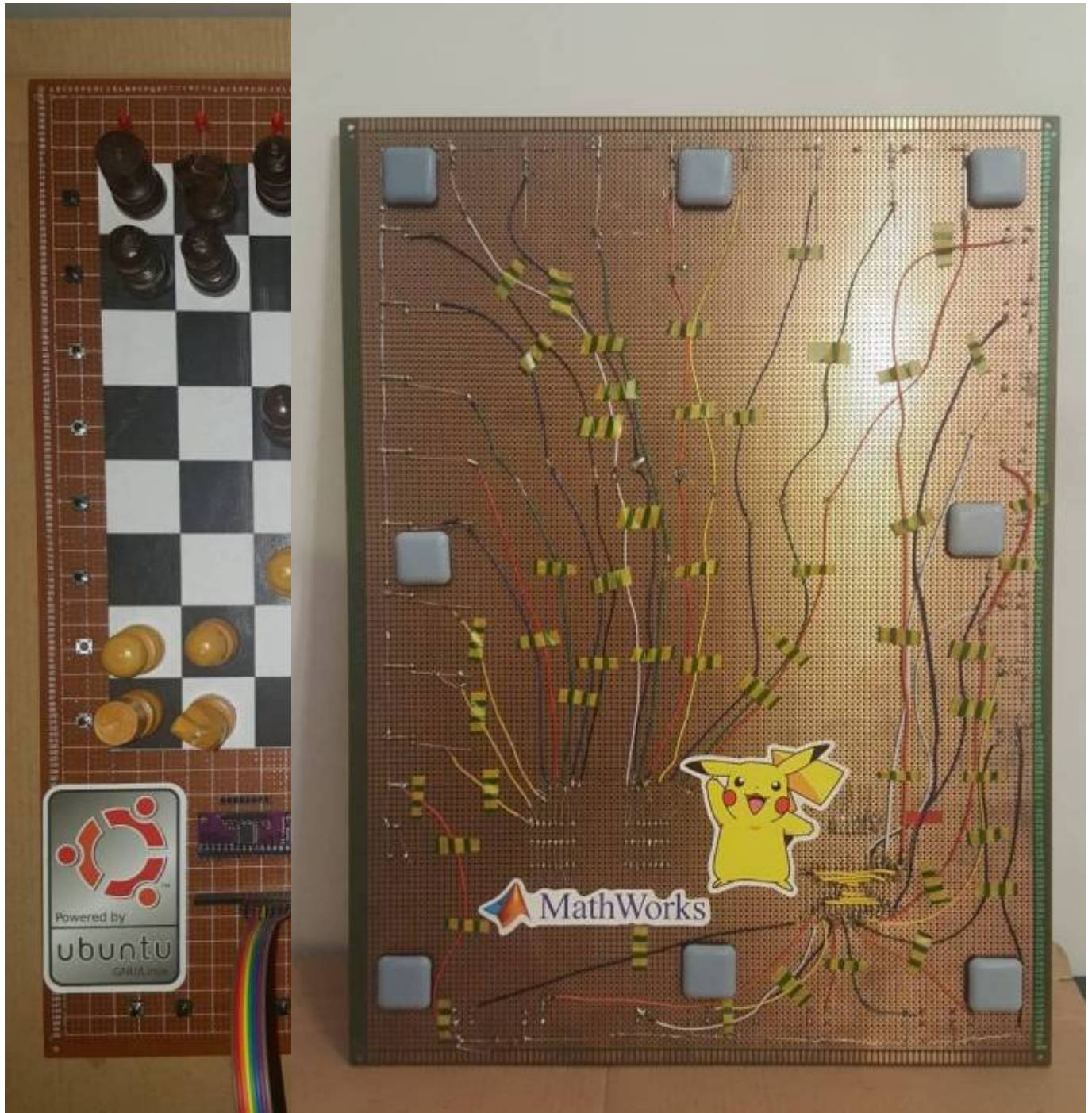
Setup proiect

1. make
2. Terminal 1: `sudo ./bootloadHID -r sah.hex` (comanda se va executa cu pinii PD6 si PD7 liberi)

3. Terminal 2: `rm test; sudo chmod 666 /dev/ttyUSB0; sudo putty /dev/ttyUSB0 -log test -serial -sercfg 9600,8,e,1,N`
4. Terminal 3: `sudo ./wrapper.sh`

#### Rezultate obținute

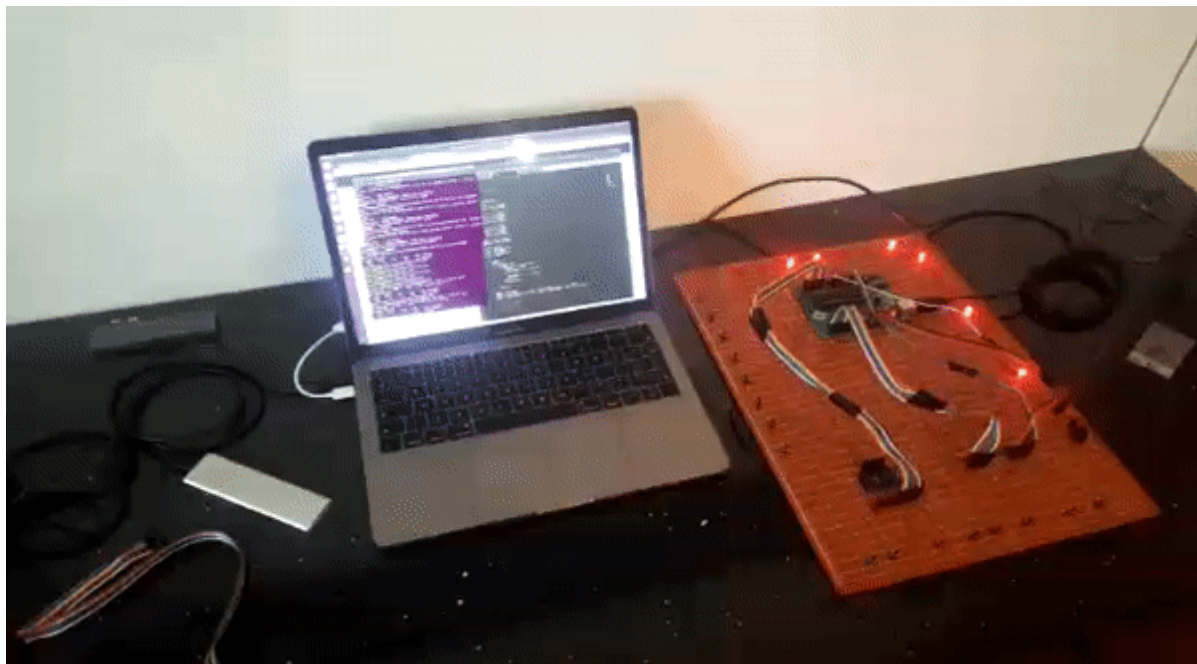
Rezultatele au fost conforme cu specificatiile. S-a obtinut o tabla electronica de sah, pe care se poate juca impotriva calculatorului, bot-ul de sah fiind cotate in jurul valorii de 2000 pe scara de rating ELO.



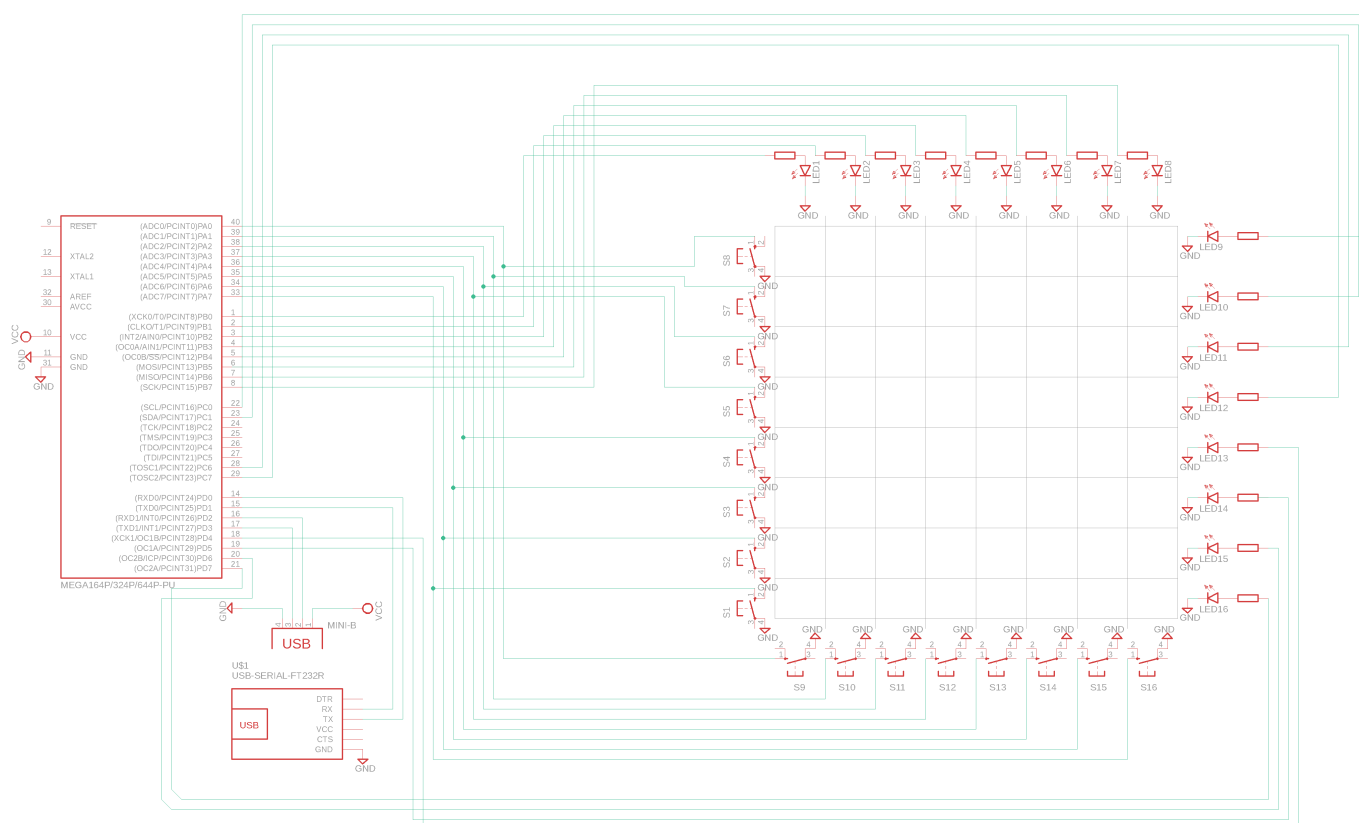
Jurnal

- 22 Aprilie 2019: Comanda de piese necesare.

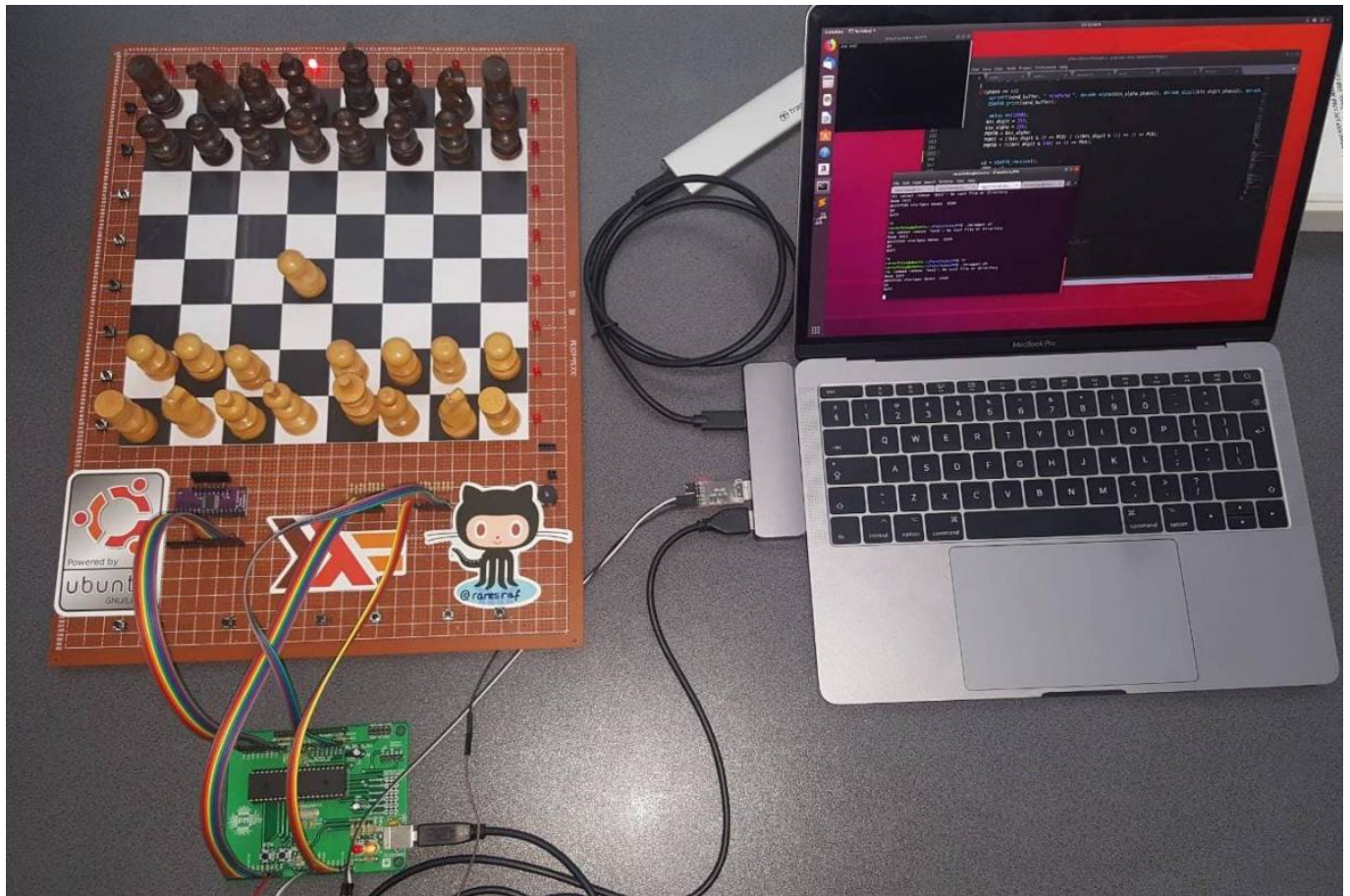




- 8 Mai 2019: Comanda de piese necesare.
- 9 Mai - 11 Mai: Lipirea placii de sah cu componentele necesare si testarea hardware a acestora
- 11 Mai: Test de hardware pentru placa de sah
- 12 Mai: Nereusirea stabilirii comunicatiei intre placa de baza si calculator. Am decis achizitionarea unui modul PL2303 USB-TTL. Am configurat USB0 pe Atmega 324PA pe pinii PD0 si PD1, iar in urma conectarii TX/RX si GND, am reusit stabilirea comunicatiei cu placa de baza pe Linux (Ubuntu 18.04), prin /dev/ttyUSB0
- 17 Mai: Defectiune la modulul multiplexor cu 16 canale. Ca urmare a avariei, am decis inlocuirea acestuia cu o multiplexare "manuala", ce aduce constrangerea selectia pozitiei impune intai selectia liniei, dupa care se impune selectia coloanei.
- 18 Mai: Noua schema hardware. A fost necesara eliminarea LCD-ului text din lipsa de porturi ramase pe microcontroller.



- 20 Mai: Adaugare feature-uri engine: suport pentru mutari speciale, precum rocade si En passant.
- 24 Mai: PM Fair - demo



Descărcare

Arhiva [proiect](#)



Link catre repository-ul de [github](#)



Documentatia in format PDF: [PDF](#)

Bibliografie și resurse

1. [UCI Protocol](#)
2. [MisterQueen Chess Engine](#)
3. [Command-line putty](#)
4. [HIDBootloader](#)
5. [Datasheet AVR](#)



From:

<http://cs.curs.pub.ro/wiki/pm/> - **PM Wiki**

Permanent link:

<http://cs.curs.pub.ro/wiki/pm/prj2019/adragan/raesraf>

Last update: **2019/05/24 15:38**

