

# Joculet Chrome

## Introducere

Prezentarea pe scurt a proiectului vostru:

- Implementarea jocului care apare pe Chrome o data ce se pierde conexiunea la internet
- Scopul player-ului e de a controla personajul prin apasarea butonelor pentru a trece peste obstacole.
- Acesta se poate juca pana la impactul cu un obstacol si atunci e Game Over
- este un joculet dragut si nostalgic

## Descriere generală

Jocul va fi realizat folosind o placuta Arduino care va avea modul OLED, card SD, speaker, LED-uri si butoane. Player-ul va misca personajul folosind butonul pentru a evita obstacolele. In timpul jocului, speaker-ul va face play la theme music, iar LED-ul va indica starea personajului: verde - in viata, rosu - mort. Toate acestea vor fi afisate pe display.

Schema bloc:



## Hardware Design

Lista de piese:

- arduino
- modul OLED
- modul Speaker
- modul LED-uri
- butoane
- modul cititor carduri SD
- fire

Schema electrica:

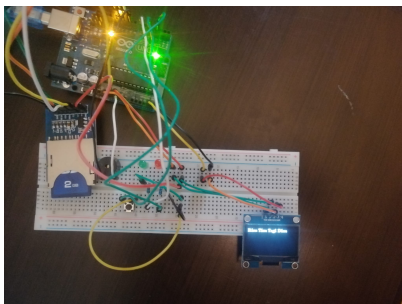


Legarea cititor Card SD: Pentru legarea cardului SD, am legat la pinii de la cititor corespunzatori placii arduino MISO - D12, MOSI - D11, SCK - D13, CS - D10 (am ales eu, nu are un pin corespunzator pe placa), GND-ul e legat la GND si ca tensiunea am ales-o pe cea de 5V.

Legare display: Asemănător ca la card SD, am legat la pinii corespunzatori placii: SDA - A4, SCL - A5, GND-ul e legat la GND, iar VCC-ul e legat la tensiunea de 3.3V.

La celelalte componente, a fost o legare destul de simplă, fiecare fiind legată la GND și un pin pe care l-am ales.

Set-up:



Testare componenta: Am ales testarea display-ului prin afișarea unui string.



## Software Design

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

Mediul de dezvoltare al proiectului este Arduino IDE. Librării folosite: U8g2lib, SD, SPI, TMRpcm. Am folosit U82glib pentru a desena pe ecran jocul în sine, iar SD, SPI și TMRpcm sunt folosite pentru a pune melodii de pe cardul SD în Main Menu.

**Concept general:** Jocul functioneaza prin aratarea main menu-ului si incepe propriu-zis o data cu apasarea butonului. In continuare, starea player-ului (adica de viata si moarte) este aratarea prin luminarea LED-ului (verde - in viata, rosu - mort). Player-ul se va deplasa si va trebui prin apasarea butonului sa sara pentru a depasi obstacolele. Daca acesta va interfera cu un obiect, display-ul va arata meniul de Game Over si se va lumina LED-ul rosu.

### Algoritmi si functii:

setup()

In functia, setup se face initializarea piniilor si se va arata mesajul de intro pana la apasarea butonului.

introMessage()

Folosind, u8g2lib se deseneaza meniul principal.

play()

Functia pentru jocul respectiv. Se initializeaza obstacolele, player-ul si scorul acestuia. La baza, are un loop din care se iese numai daca se face o coliziune dintre player si obstacol. In loop, mai intai citesc sa vad daca am un input si butonul a fost apasat. Daca da, si momentan player nu este in actiunea de a sari, il fac sa sara.

Daca sar, updatez coordonatele player-ului sa sara. Dupa acest pas, verific daca exista o coliziune dintre player si obstacol. Daca exista, ies din acest loop. Altfel, verific sa vad daca obstacolele peste care am sarit au iesit din display si daca da, le voi initializa din nou sa apara pe display.

Odata cu pierderea jocului si iesirea din loop, desenez meniul de GameOver si afisez scorul.

gameOver()

Folosind din nou u8g2, voi desena meniul de game over. Pot relua jocul prin apasarea butonului.

### Concepte folosite de la laborator

- GPIO - folosit pentru pinii de input si output: buton, led, buzzer.
- UART - folosit pentru Serial.print(), utilizat mai mult pentru debugging.
- I2C - folosit pentru display-ul OLED.
- SPI - folosit pentru card-ul SD.

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Link video joculet:

<https://www.youtube.com/watch?v=CPqGosRE6fo>


Link video buzzer:

[https://www.youtube.com/watch?v=Q\\_gELkM0WD4](https://www.youtube.com/watch?v=Q_gELkM0WD4)

## Concluzii

Implementarea joculeului a fost una simplistica care a avut cateva dificultati, dar a fost un journey dragut. Printe dificultati s-a numarat lipsa memoriei dinamice de pe placa o data cu adaugarea de sunet si faptul ca testul audio care se aude destul de prost, dar acesta functioneaza. Inca o dificultate am putea zice a fost vanatoare de componente. In rest, totul a fost interesant si a fost smooth sailing. Recomand.

## 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**.

Link Github: <https://github.com/pauna0311/ProiectPM>

## Jurnal

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



Grafic Gant pentru planificare.

Scopuri de atins:

1. Sa fie functional

## Bibliografie/Resurse

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

Surse: <https://kits.plusivo.com/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/irina.pauna0311>



Last update: **2024/05/25 13:33**