

Pong Multiplayer Game

Introducere

Ce face?

Proiectul implică dezvoltarea unui dispozitiv interactiv de divertisment, conceput pentru a oferi utilizatorilor o experiență de joc captivantă. Acest dispozitiv permite utilizatorilor să selecteze modul de joc (single player sau multiplayer) prin intermediul butoanelor disponibile, pentru un joc de Ping Pong. Apoi, ei interacționează cu joystick-urile dedicate pentru a participa la joc, care se desfășoară într-o serie de „best of 3”, afișându-se pe ecranul conectat. În funcție de rezultat, buzzer-ul va crea un sunet de învingător sau pierzător.

Scopul lui?

Scopul principal al dispozitivului este de a oferi o modalitate distractivă și interactivă de petrecere a timpului liber, aducând împreună elemente clasice de joc și tehnologie modernă.

Alte considerente

Consider că acest joc nu are o utilitate mare în zona educației sau medicinei, dar conferă o notă de nostalgie jucătorilor care au avut primul contact cu un astfel de joc. Prin acest proiect încerc să dezvolt un joc vechi într-un mod cât mai plăcut pentru toată lumea.

Ideea de la care am pornit pentru acest proiect a fost una simplă: îmi doream să fac ceva „catchy”, dar suficient de complicat și interactiv.

Descriere generală

Funcționarea sistemului de joc:

1. Inițializarea jocului:

- Utilizatorul pornește sistemul și alege, prin interacțiunea cu butonul, modul de joc: singleplayer,

multiplayer.

2. Începerea jocului

- Odată inițializat jocul, acesta va începe, afișând pe ecran scorul actual (0 - 0) și creând jucătorii (cele 2 palete)

3. Procesarea mișcărilor

- În funcție de modul de joc ales, acțiunea joystick-urilor va afecta poziția paletelor (sau a unei singure paletă).
- Monitorizarea acțiunilor asupra joystick-urilor se va face constant de către placa de acțiune (Arduino).
- Prin atingerea bilei de joc cu o paletă, bila se va mișca către terenul advers.
- Afișarea mișcărilor corecte va fi procesată de placa Arduino și transmisă către utilizator prin display-ul LED I2C.

4. Încheierea jocului

- Jocul se va încheia odată cu atingerea scorului de 2/3 puncte de oricare dintre jucători.
- Procesarea mișcărilor joystick-urilor va fi oprită.
- Va fi acționat buzzer-ul pentru a transmite sunetul corespunzător finalului de joc.

5. Rematch

- Odată cu finalizarea jocului anterior va fi pornită opțiunea de rematch care va reinițializa un joc după formatul anterior.

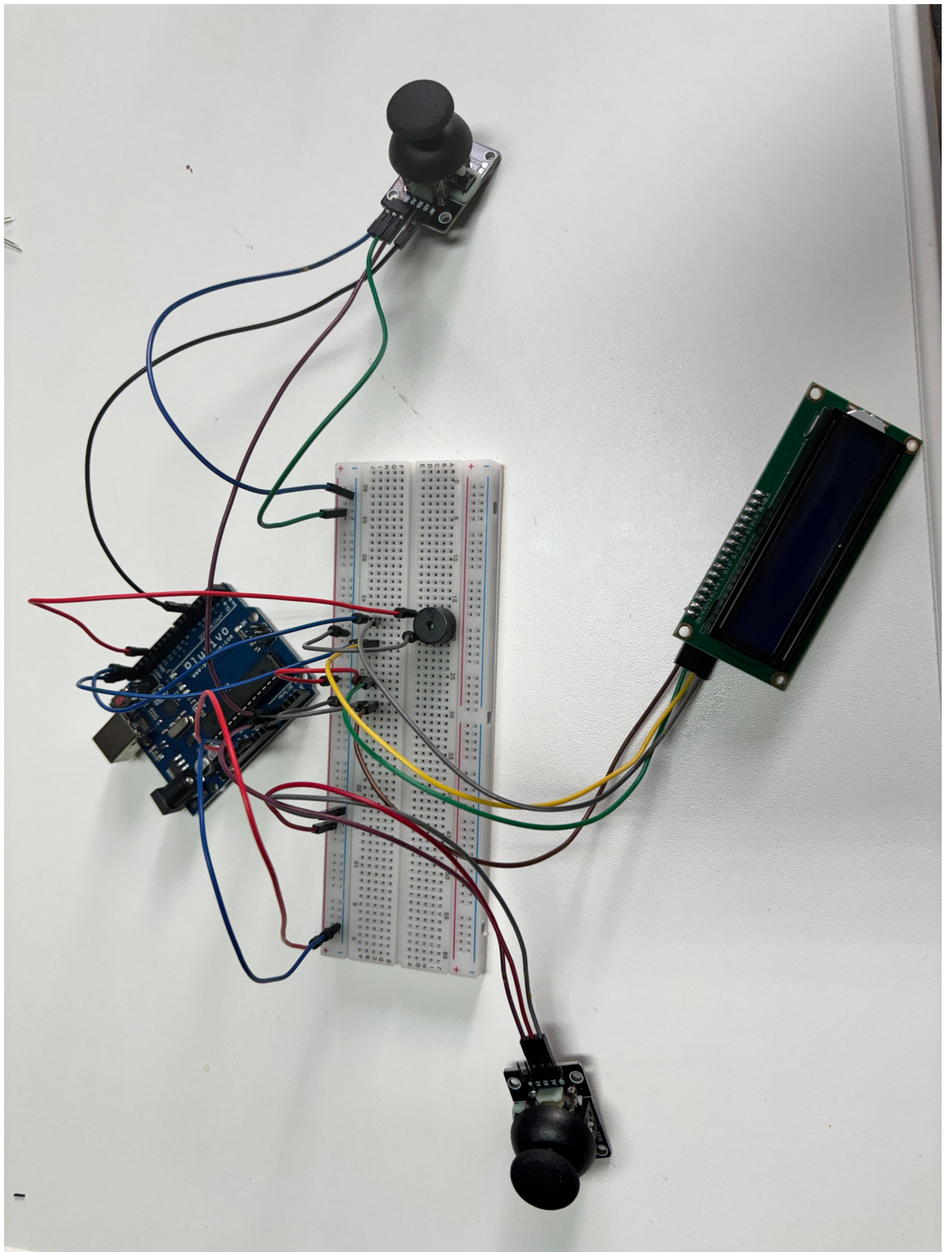


Hardware Design

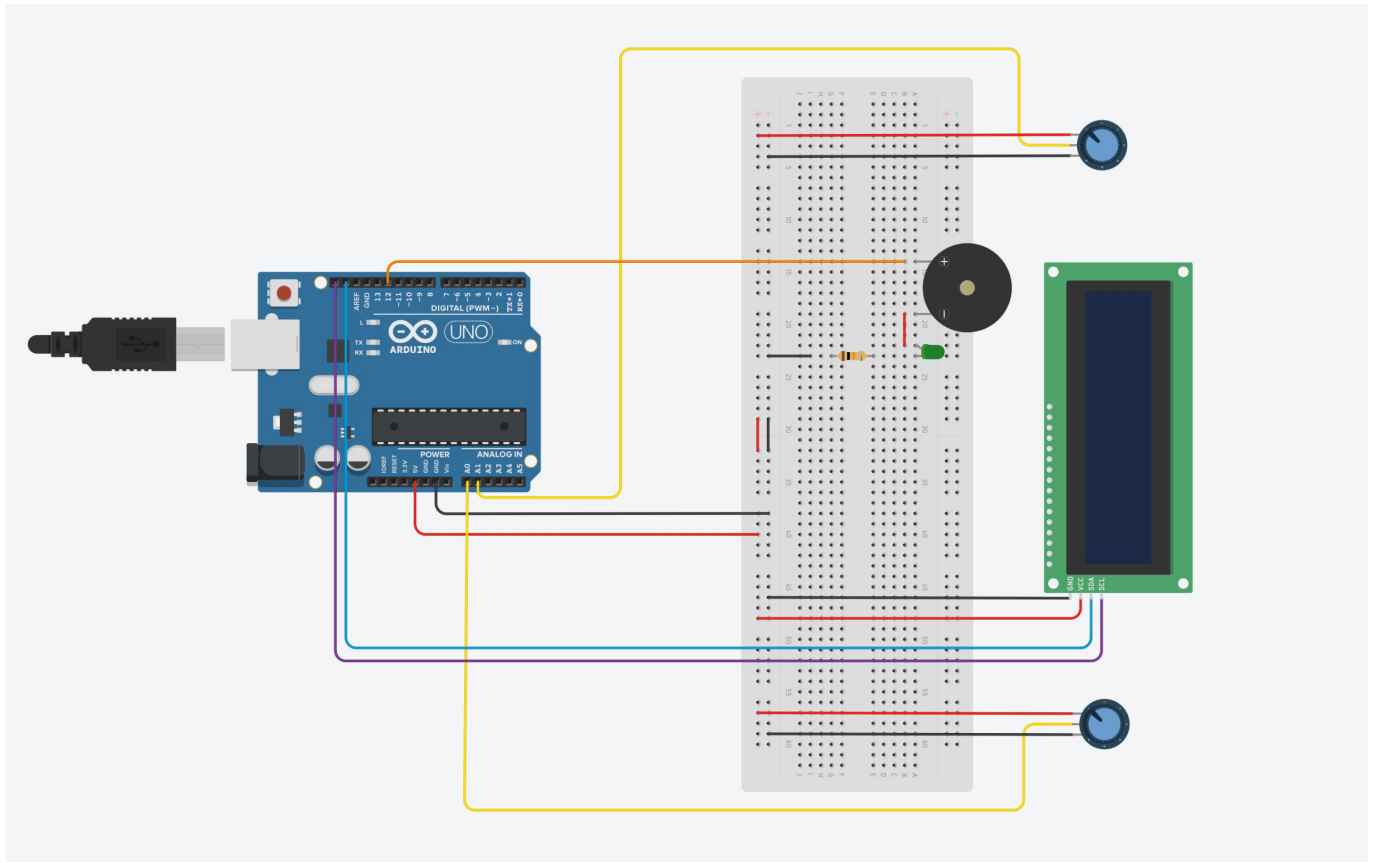
Lista componente:

Nume componentă	Cantitate
Arduino UNO (echivalent ATmega328P)	1
Buzzer	1
Joystick Analog	2
Display LCD (I2C)	1
Breadboard	1
LED (diverse culori)	1/2
Rezistor (diverse valori)	1/2
Cabluri	nelimitat

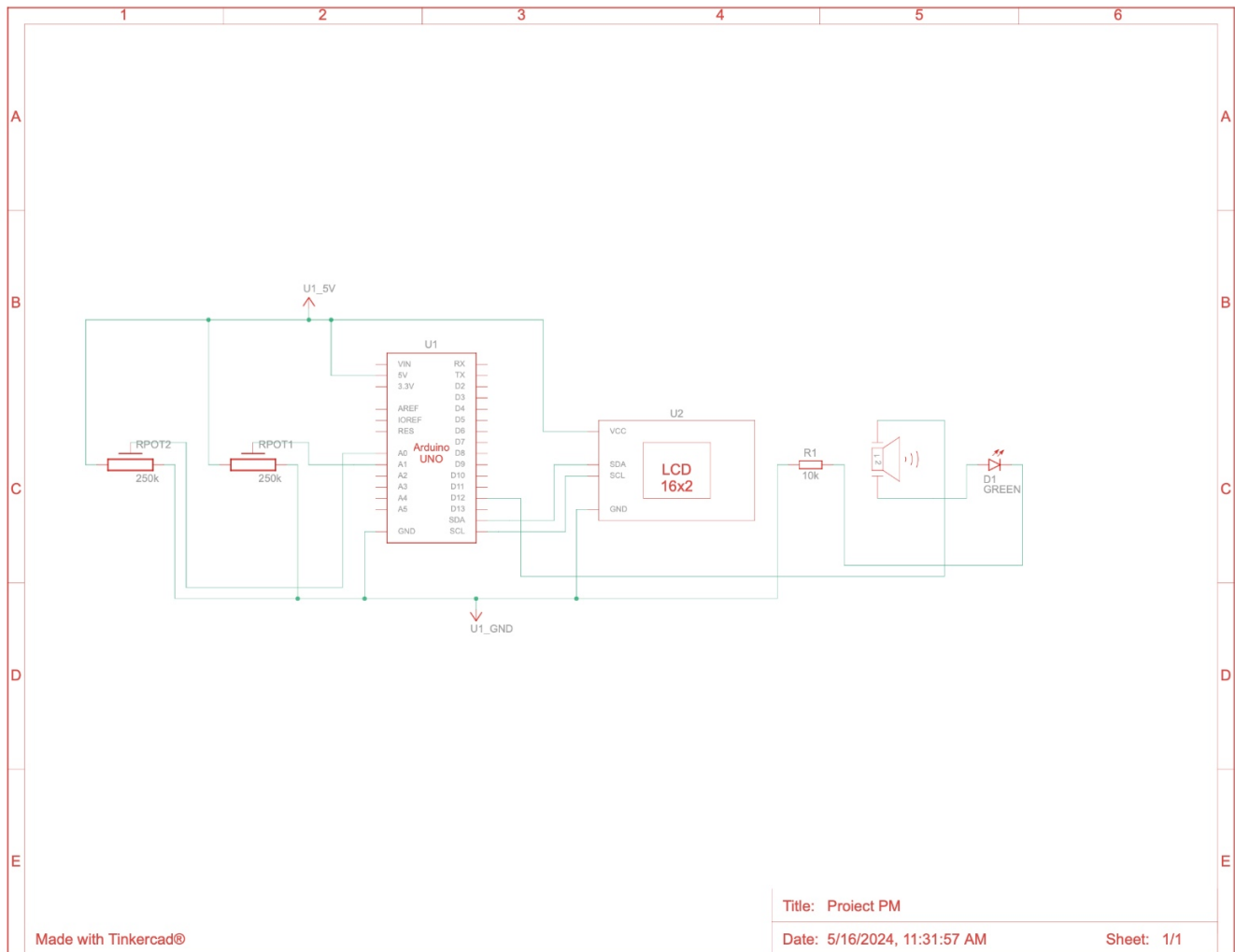
Cum arată până acum:



Modelul sistemului (cu componente vizibile):



Schematicul sistemului:



- Am utilizat potențiometre în locul joystick-urilor analogice din cadrul ambelor scheme, deoarece nu am putut găsi online un model care să se potrivească.
- Urmează să construiesc din carton (sau orice material mai accesibil) o carcasă pentru a putea juca fără posibilitatea de a deconecta cablurile.

Software Design

Descrierea codului aplicației

• Mediu de dezvoltare:

Pentru dezvoltarea acestui proiect, a fost utilizat mediu de dezvoltare Arduino IDE. Acesta oferă un mediu prietenos și ușor de utilizat pentru programarea microcontrolerelor din familia Atmega.

• Librării și surse 3rd-party:

Au fost utilizate următoarele librării:

1. **Wire.h**: pentru comunicarea I2C între Arduino și display-ul LCD.
2. **LiquidCrystal_I2C.h**: pentru gestionarea și controlul display-ului LCD I2C.

• **Algoritmi și structuri pe care plănuiesc să le implementez:**

Algoritmii principali includ:

1. Controlul mingii în jocul Pong, incluzând detectarea coliziunilor cu pereții și paletetele.
2. Actualizarea scorurilor și afișarea acestora pe display.
3. Detectarea intrărilor analogice de la joystick-uri pentru controlul paletelor.

1. **Surse și funcții implementate:**

1. Inițializarea și configurarea display-ului LCD I2C.
2. Implementarea funcției de desenare a terenului de joc și a scorului.
3. Controlul mișcării mingii și detectarea coliziunilor.
4. Implementarea sunetelor pentru diferite evenimente în joc (coliziuni și puncte).

Stadiul actual al implementării software

Codul sursă: [GitHub Link](#)

Funcții create și utilizate

1. **setup** - Inițializează afișajele LCD, setează modurile pinurilor pentru butoane, LED-uri și buzzer, creează caracterul personalizat al mingii și pregătește jocul pentru selecția inițială a modului și pornirea jocului.
2. **loop** - Bucloa principală a jocului care gestionează mișcarea mingii, mișcarea paletelor, actualizarea scorului și apelarea funcției `drawScore` când este necesară o actualizare.
3. **drawScore** - Actualizează scorurile, cronometrul și rundele câștigate pe cel de-al doilea afișaj LCD, asigurându-se că doar părțile necesare sunt actualizate pentru a evita pâlpâirea.
4. **showStartMessage** - Afișează mesajul de pornire pe primul afișaj LCD pentru a informa jucătorii că jocul este gata de jucat.
5. **showBallPossession** - Afișează un mesaj care indică ce jucător deține mingea la începutul rundei.
6. **ledShowAndPlayMelody** - Aprinde LED-urile și redă o melodie folosind buzzer-ul când un jucător câștigă o rundă sau un joc.
7. **waitForRestart** - Afișează un mesaj care solicită jucătorilor să apese un buton pentru a începe o nouă rundă. Așteaptă până când butonul este apăsat.
8. **resetRound** - Resetează scorurile pentru runda curentă și pregătește jocul pentru runda următoare, afișând din nou mesajul de pornire.

9. **resetGame** - Resetează toate scorurile și rundele câștigate, pregătind jocul pentru un nou meci de la început și afișează mesajul de pornire.
10. **gameOver** - Afișează un mesaj care indică ce jucător a câștigat jocul.
11. **selectMode** - Permite jucătorilor să selecteze între modul single-player și multiplayer folosind joystick-ul și butonul de selecție.
12. **displayModeMenu** - Afișează meniul de selecție a modului pe cel de-al doilea afișaj LCD.
13. **waitForModeSelection** - Instrucționează jucătorul să verifice cel de-al doilea afișaj pentru selecția modului și așteaptă selectarea modului.
14. **movePlayer2Automatically** - Mișcă automat paleta jucătorului 2 pe baza poziției mingii, utilizată în modul single-player.
15. **setup** - Inițializează jocul prin configurarea afișajelor LCD, a pinurilor de intrare/ieșire și a condițiilor de pornire pentru joc.
16. **loop** - Bucla principală a jocului care rulează continuu, gestionând intrările jucătorilor, mișcarea mingii, actualizările scorurilor și schimbările de stare ale jocului.
17. **waitForRestart** - Afișează un mesaj care cere jucătorilor să apese un buton pentru a începe o nouă rundă și așteaptă apăsarea butonului.
18. **resetRound** - Resetează scorurile pentru runda curentă și pregătește jocul pentru o nouă rundă, afișând mesajul de pornire.
19. **resetGame** - Resetează toate variabilele jocului la stările inițiale, pregătind pentru un nou joc.
20. **gameOver** - Afișează un mesaj care indică ce jucător a câștigat jocul.
21. **selectMode** - Gestionează selecția modului single-player sau multiplayer de către jucător.
22. **displayModeMenu** - Afișează meniul de selecție a modului pe cel de-al doilea afișaj LCD.
23. **waitForModeSelection** - Afișează instrucțiuni pentru a verifica al doilea afișaj pentru selecția modului și așteaptă alegerea jucătorului.
24. **movePlayer2Automatically** - Mișcă automat paleta jucătorului 2 pentru a urmări mingea în modul single-player.

Elementul de noutate al proiectului este dat de posibilitatea de a interacționa direct cu sistemul prin intermediul joystickurilor.

Laboratoare utilizate

1. [Laboratorul 3: Timere, Pulse Width Modulation \(PWM\)](#)
2. [Laborator 4: Analog Digital Convertor \(ADC\)](#)
3. [Laboratorul 6: I2C \(Inter-Integrated Circuit\)](#)

Rezultate Obținute

[Youtube Link](#)

Concluzii

Download

Arhiva care conține toate fișierele create pentru acest proiect se poate descărca de pe link-ul de [GitHub](#).

Bibliografie/Resurse

Hardware Utilizat

1. Placă Arduino

- [Arduino Uno Fake](#)

2. Afișaj LCD I2C

- [LCD 1602 I2C](#)

3. Joystick-uri Analogice

- [Joystick Module](#)

4. Butoane Digitale

- [Push Buttons](#)

5. Buzzer

- [Buzzer Module](#)

Software Utilizat

1. Arduino IDE

- [Arduino IDE Download](#)

2. Configurare Display

- [I2C Display](#)

2. Biblioteci Arduino

- **Wire.h**
 - [Arduino Wire Library](#)
- **LiquidCrystal_I2C.h**
 - [LiquidCrystal I2C Library](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/andrei.napruiu>



Last update: **2024/05/28 18:29**