

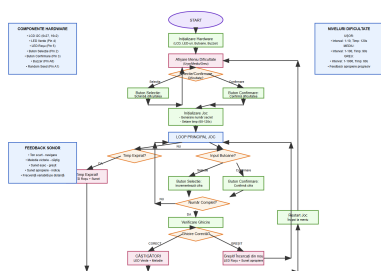
Guess the number

Introducere

Am creat un joc cu Arduino care îți permite să ghicești numere. Ai un ecran care îți arată informațiile, două butoane pentru a juca, LED-uri care se aprind în diferite culori și un buzzer care face sunete. Jocul funcționează în felul următor: alegi cât de greu vrei să fie - poți ghici numere de la 1 la 10 dacă vrei să fie ușor, de la 1 la 100 pentru mediu, sau de la 1 la 1000 dacă vrei să fie greu. Fiecare nivel îți dă timp diferit să ghicești - de la 2 minute (cel ușor) până la 1 minut (cel greu). Buzzer-ul face sunete diferite să îți arate cât de aproape ești. Dacă ești foarte aproape, sunetele sunt rapide și înalte. Dacă ești departe, sunetele sunt lente și joase. LED-ul verde se aprinde când câștigi, cel roșu când greșești, și sunt melodii speciale pentru când câștigi sau pierzi. Introduci numărul cifră cu cifră apăsând butoanele, și poți să încerci din nou de câte ori vrei fără să oprești jocul.

Descriere generală

O schema bloc:



Componente Hardware - Arduino (microcontroller principal) - LCD I2C 16x2 (afișare informații) - 2 LED-uri (feedback vizual - verde/roșu) - 2 Butoane (input utilizator - selecție/confirmare) - Buzzer (feedback audio)

Module Software - Modul Inițializare - configurare hardware și meniu principal - Modul Dificultate - selecție nivel și setare parametri - Modul Joc Principal - loop principal cu timer și input - Modul Verificare - logica de ghicire și feedback - Modul Audio - sisteme de sunete contextuale

Ordinea de execuție: Inițializare → Selecție Dificultate → Joc Activ → Feedback → Reset/Restart

Hardware Design

Componente Hardware:

1. Microcontroller Arduino

- Model: Arduino UNO (ATmega328P)
- Tensiune operare: 5V
- Memorie: 32KB Flash, 2KB SRAM, 1KB EEPROM

2. Display LCD I2C (16×2)

- Specificații: 16 coloane × 2 rânduri
- Protocol: I2C (Inter-Integrated Circuit)
- Tensiune: 5V
- Pini conectați:

1. SDA → A4
2. SCL → A5
3. VCC → 5V
4. GND → GND

3. LED-uri Indicator

LED Verde (Succes)

- Pin Arduino: 4

LED Roșu (Eroare/Timp Expirat)

- Pin Arduino: 5

4. Butoane de Control

Buton Selecție

- Pin Arduino: 2

Buton Confirmare

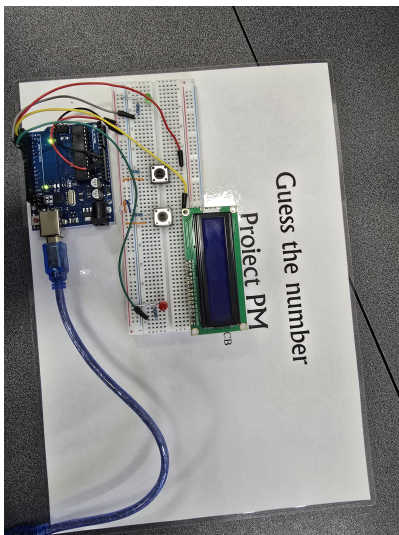
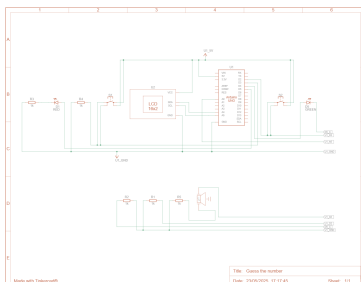
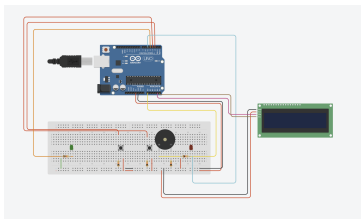
- Pin Arduino: 3

5. Buzzer Audio

- Pin Arduino: A0
- Tip: Buzzer pasiv

- Tensiune: 5V

- Control: functia `tone()`



Software Design

Mediu de dezvoltare: Arduino IDE

Librarii: LiquidCrystal_I2C.h

Structuri si algoritmi:

State machine cu stari: diffSelect, gameOver, selectMode pentru controlul fluxului de executie

Timer non-blocking bazat pe millis() pentru countdown-ul in timp real

Debouncing software cu interval de 300ms pentru stabilitatea butoanelor si prevenirea apasarilor multiple

Algoritm de construire progresiva a numarului (cifra cu cifra) adaptat pe dificultate:

Easy (1-10): doar unitati

Medium (1-100): zeci + unitati

Hard (1-1000): sute + zeci + unitati

Generator aleator pentru randomizare cu randomSeed

Sistem de feedback audio adaptat cu frecvente variabile bazate pe proximitatea ghicirii

PWM si tone() pentru controlul buzzer-ului si generarea tonurilor

Algoritm de verificare cu feedback directional (prea mic/prea mare)

Sistem de proximitate inteligent cu praguri adaptive in functie de dificultate

Functii implementate:

setup() - initializare hardware si configurare pin-uri

loop() - bucla principala cu gestionarea starilor

handleDiffSelect() - gestioneaza selectia si confirmarea nivelului de dificultate

initGame() - seteaza parametrii specifici dificultatii (limite, timp, pozitii cifre)

checkTime() - monitorizeaza timer-ul si actualizeaza display-ul in timp real

checkGuess() - compara numarul introdus cu cel ales random si ofera feedback

resetGuess() - reseteaza starea pentru o noua incercare fara restart complet

timeUp() - gestioneaza sfarsitul jocului cand timpul expira

showDiffMenu() - afiseaza meniul principal de selectare dificultate

showDiff() - afiseaza dificultatea curenta selectata

proximitySound() - calculeaza si reproduce feedback audio bazat pe distanta pana la raspuns

winSound() - melodie de victorie cu tonuri crescatoare

failSound() - sunet de esec cu tonuri descrescatoare

quickBeep(), shortBeep(), confirmBeep() - feedback audio pentru actiuni utilizator

Rezultate Obținute

Proiectul și-a atins toate obiectivele, creând un joc fara probleme.

Limitări identificate:

- Lipsa persistenței datelor la restart (scoruri, statistici)
- Feedback audio limitat la un singur buzzer (monoton)
- Interface fixă fără personalizare pentru utilizatori
- Dependența de alimentarea externă (fără baterie)

Ce mai poate fi adaugat:

- LED RGB pentru efecte vizuale mai complexe în loc de LED-urile simple
- Card SD pentru salvarea scorurilor și statisticilor de joc
- Mod Multiplayer - doi jucători alternează
- Mod Sprint - ghicește cât mai multe numere în timp limitat

Concluzii

Proiectul a fost finalizat cu succes, demonstrând funcționarea stabilă a tuturor componentelor și modulelor implementate. Experiența de dezvoltare a oferit cunoștințe practice în programarea Arduino și integrarea sistemelor embedded. Rezultatele obținute confirmă eficiența soluțiilor adoptate pentru realizarea jocului.

Download

Arhiva [guessthenumber.zip](#)

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/rnedelcu/catalin.aldobritei>



Last update: **2025/05/28 10:38**