

Impostor Detector

Mogodeanu Claudiu 331CC

Introducere

O reinterpretare a minigame-ului "Start Reactor" din popularul joc AmongUs ([video](#)).

- Jucatorul va avea de replicat cateva secvente succesive in care se aprind niste led-uri dintr-o matrice, cu dificultate reglabila si dinamica.
- Jocul se va activa daca o persoana este detectata in proximitate.
- In functie de rezultatul actiunilor, se porneste o alarma sau jucatorul primeste un premiu, aflat intr-o cutie initial inchisa.

Descriere generală

Ordinea etapelor in care functioneaza jocul este urmatoarea:

1. Prin intermediul senzorului de distanta se detecteaza daca un User este in apropiere, caz in care se incepe jocul propriu-zis. Cat timp jocul este jucat, senzorul este dezactivat.
 2. Se genereaza o secventa aleatoare in care se aprind cateva LED-uri de pe matrice. (se poate repeta acest pas de maximum 5 ori)
 3. User-ul interactioneaza cu matricea de butoane si incearca sa reproduca secventa aparuta anterior:
 1. Daca greseste: se incheie jocul si se trimite un semnal la buzzer pentru a anunta ca s-a gasit Impostorul.
 2. Daca secventa este corecta: se reia pasul **2**, doar ca secventele vor fi mai lungi.
 4. Secventele au fost completate cu succes, deci Arduino actioneaza servomotorul, care deschide cutia cu premiu.
- In timp ce este randul user-ului, acesta poate decide sa scada sau sa creasca dificultatea pentru rundele urmatoare, apasand pe cele 2 butoane aditionale, care trimit semnal microcontroller-ului.
 - Pe timpul jocului, pe un ecran LCD apare constant statusul jocului.

SCHEMA BLOC:



Hardware Design

LISTA DE PIESE SI SCOPUL LOR:

- **Arduino Uno MEGA328P** → microcontroller-ul care este centrul proiectului
- **Matrice de butoane 4×4** → user-ul apasa pe butoane pentru a replica secventa
- **Matrice de LED-uri WS2812** → se aprind secvente de led-uri, urmarite de user
- **Modul LCD 1602** → afiseaza statusul jocului
- **Buzzer Pasiv** → emite o alarma atunci cand se greseste secventa
- **Butoane aditionale** → se poate schimba frecventa cu care se aprind led-urile
- **Senzor Ultrasonic de Distanță HC-SR04** → daca cineva este aproape, se porneste jocul
- **Micro Servomotor SG90** → cand se castiga jocul, se deschide o cutie

SCHEMA HARDWARE

Mediu de dezvoltare: Fusion 360



ALEGEREA PINILOR

- **Modulul LCD** este conectat prin **I2C** ⇒ am folosit pinii de date si clk ai arduino Uno, acestia fiind A4 si A5.
- **Butoanele de dificultate** folosesc **intreruperi externe** (pentru a putea avea efect in timpul altor actiuni), iar pe Uno acestea exista pe pinii D2 si D3.
- **Servomotorul** se foloseste de **PWM** incorporat, dar trebuie sa fie conectat pe un pin care accapta PWM. Am ales D10.
- Restul componentelor au nevoie de **semnale digitale**, am incercat sa le grupez pe cat posibil.
- Pinii pentru traseele de putere (**GND si 5V**) sunt toti uniti intre ei si folosesc doar cate un pin de la microcontroller.

FORMA INITIALA SI DOVADA FUNCTIONARII PARTIALE



- Am testat individual fiecare componenta (dupa ce am dus toate firele), mai am de realizat software-ul complet
- Componentele sunt lipite/insurubate pe o placa de plexiglas pentru a le da o forma compacta

FORMA FINALA



Software Design

Mediu de dezvoltare: VSCode cu PlatformIO, limbajul C/C++

Biblioteci externe folosite:

- [FastLED](#) → Matricea de LED-uri
- [Keypad](#) → Matricea de butoane
- [LiquidCrystal_I2C](#) → Ecran LCD pe I2C

- [Servo](#) → Servomotor, pentru incorporare PWM
- [Arduino Songs](#) → Sunetul din song.cpp, folosit la final

Proiectul propriu-zis: [github link](#)

Flow-ul principal poate fi reprezentat printr-un state machine destul de basic:
(notatiile pentru stari sunt aceleasi ca cele din fisierul States.h)



Descriere sumara a functiilor si a scheletului:

- Am incercat sa modularizez pe cat posibil codul, dar nici sa nu am fisiere aproape goale.
- **My_pins.h** contine alocarea de pini pentru cele 2 matrice si pentru butoanele separate, iar LCD-ul si buzzer-ul care sunt folosite in paralel cu celelalte functionalitati, au functiile definite in **Buzzer_lcd.cpp**
- Se porneste din starea CHECK_DISTANCE, care este practic o stare idle, in care actioneaza doar senzorul de distanta pentru a detecta daca exista cineva in apropiere (ca in jocurile din AmongUs unde trebuia sa apropii mana pentru a da start). Functiile principale pentru aceasta sunt **read_distance()** si **get_average_distance()**, aflate in **Distance_sensor.cpp**. Pentru ca citirile mai dadeau si fals pozitive de genul: distanta 1 sau 0 chiar daca nu era nimic prin apropiere, a trebuit sa ma ocup de calibrarea acestuia. Am decis sa calculez media a 10 masuratori consecutive (facute la cateva milisecunde una de alta) inainte de a considera o valoare "pe bune". Abia apoi, daca distanta este si in cazul acesta de cativa cm, pornim jocul propriu-zis.
- Se va considera starea BLINK_LEDS de maximum **max_level** ori, dupa fiecare nivel (apel de **random_led_sequence()**) utilizatorul trebuind sa repete secventa, fiind in starea REPLICATE. In momentul in care greseste o data, suna buzzer-ul si se reseteaza tot jocul. De exemplu, daca trebuie replicata o secventa de 5 led-uri si se greseste la al 3lea, nici nu se mai incearca ultimele 2.
- Cand se completeaza toate nivelele, se ajunge in starea GG, in care se activeaza servomotorul si se apeleaza functia **sing()**.
- In fisierul **song.cpp** am pus o secventa din Arduino Songs (link mai sus), fiind apelata prin sing().
- Buzzer-ul, prin **small_buzz()** si **big_buzz()** din **Buzzer_lcd.cpp**, este folosit si pentru a anunta jucatorul cand poate incepe sa faca o actiune.
- Oricand, pe parcursul jocului, se poate apasa pe unul din butoanele aditionale, care prin intreruperile **INTO** sau **INT1** modifica timpul in care un led sta aprins, marind sau micșorand dificultatea. Am folosit si un debounce diy bazat pe millis() trecute de la ultima apasare de astfel de buton pentru a evita detectiile eronate. Observasem initial ca mi se lua de 2-3 ori un buton, dar acum merg as expected.
- Aditional, inainte de a incepe jocul (adica de a detecta senzorul ceva), cand se apasa pe butoanele separate, se poate seta numarul nivelelor pe care o sa le aiba utilizatorul de replicat pe parcursul jocului urmator.
- Am incercat, pe cat posibil, sa fac lucrurile exact cum aveau loc in minigame-ul din AmongUs.

Laboratoare folosite:

- **intreruperi** la butoanele aditionale
- **I2C** la ecran LCD
- **PWM** la servomotor
- **GPIO** la butoanele aditionale

Rezultate Obținute

Video demonstrativ: [Impostor Detector](#)

Concluzii

Având în vedere că a fost prima mea interacțiune (mai mult decât niste laboratoare cu schelet și plăci deja existente) cu un Arduino, am învățat destul de multe lucruri interesante.

Ținând cont de lipsa de experiență, am decis din start să nu mă îndrept spre un proiect extrem de complex, deși nici acesta nu este naturalnic, astfel încât să dețin totuși control asupra lui și să învăț/folosesc cât mai multe elemente de bază.

Îmi plac bibliotecile externe, dar am folosit și câteva registre

Nu sunt extrem de mulțumit de aspectul proiectului, mai ales de cable management, dar este totuși destul de compact și complet funcțional.

Download

Remember git for source code: [github project](#)

Cod, schematic și imagini: [Arhiva](#)

Sincer relevant este doar repo-ul de github. Restul sunt imagini care există pe pagina de ocw deja.

Jurnal

- 25.04 - Comandat piese
- 28.04 - Creat pagina ocw inițială
- 02.05 - Done Milestone 1 (Documentație)
- 05.05 - Inițiat jurnal
- 12.05 - Unit fire și lipire inițială (hardware inițial)
- 15.05 - Done Milestone 2 (Documentație Hardware)
- 19.05 - Gata 90%+ din software (de aici pot adăuga chestii extra)
- 20.05 - Done Milestone 3 (Documentație Software)
- 25.05 - Rezultate finale (video YouTube)
- 26.05 - Finalizat pagina OCW

Bibliografie/Resurse

1. [Draw.io](#) - pentru diagrame și scheme bloc
2. [ATmega328P_Datasheet](#)

3. [Optimus Digital](#) - doar un exemplu de piesa. Si restul au fost cumparate tot de aici
4. [Laboratoare TSC](#) - cum am invatat sa creez schematic si piese in Fusion 360
5. [Modul-4x4-keypad-matrix-buttons](#) - cod matrice butoane
6. [Tutorial lcd](#) - samples de cod LCD

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/fgul/claudiu.mogodeanu>



Last update: **2024/05/26 19:38**