

Defuse the bomb puzzle

Lăbău-Cristea Andrei-Liviu 334CB

Introducere

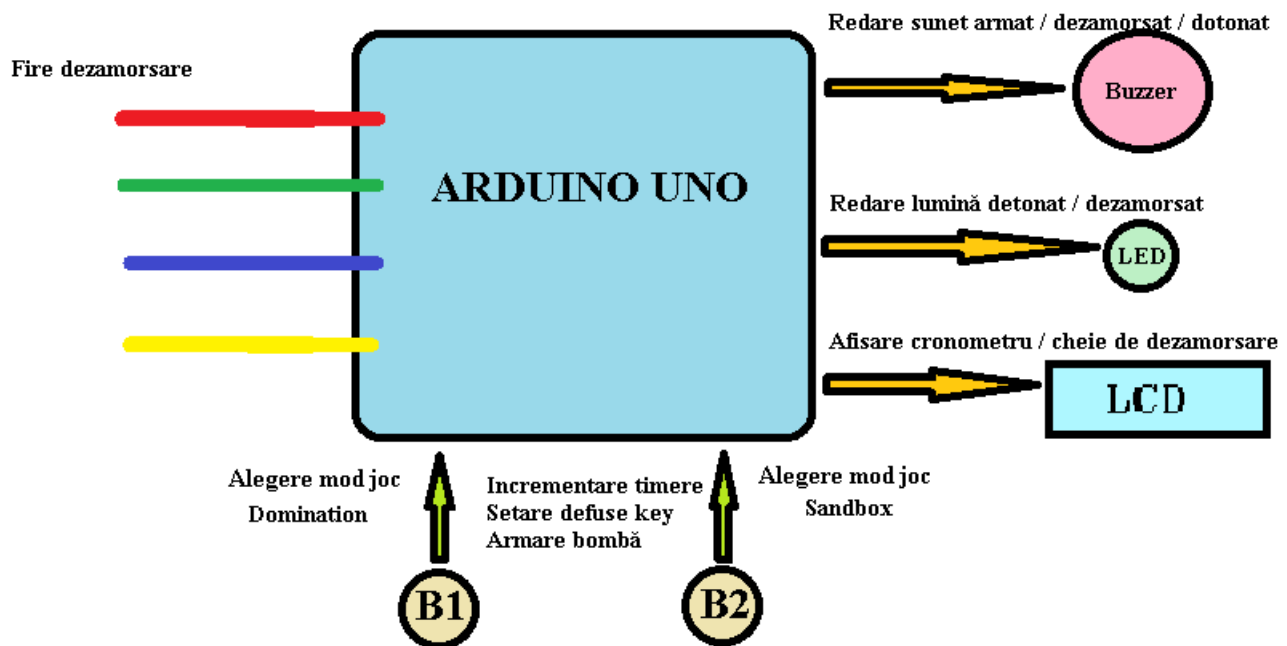
Ideea proiectului este de a reface in format fizic, folosind o placuta Arduino, un puzzle asemanator cu bomba din [Counter Strike](#) la care se mai adauga cateva features extra pentru o complexitate mai mare. Scopul este de a avea un puzzle care sa ofere o experienta imersiva in pielea unui inginer genist care trebuie sa isi foloseasca intuitia si indiciile primite pentru a dezamorsa o bomba contra timp. Pentru a da mai multa libertate de alegere utilizatorului vor exista 2 moduri de joc:

- Sandbox: Se va cere sa se seteze un cronometru si o cheie de dezamorsare.
- Domination: Cronometrul si cheia de dezamorsare vor fi generate random.

DISCLAIMER: Proiectul este realizat in scop didactic, nu include materiale inflamabile sau explozibile si nu are nicio legatura cu proiectarea unei bombe reale!

Descriere generală

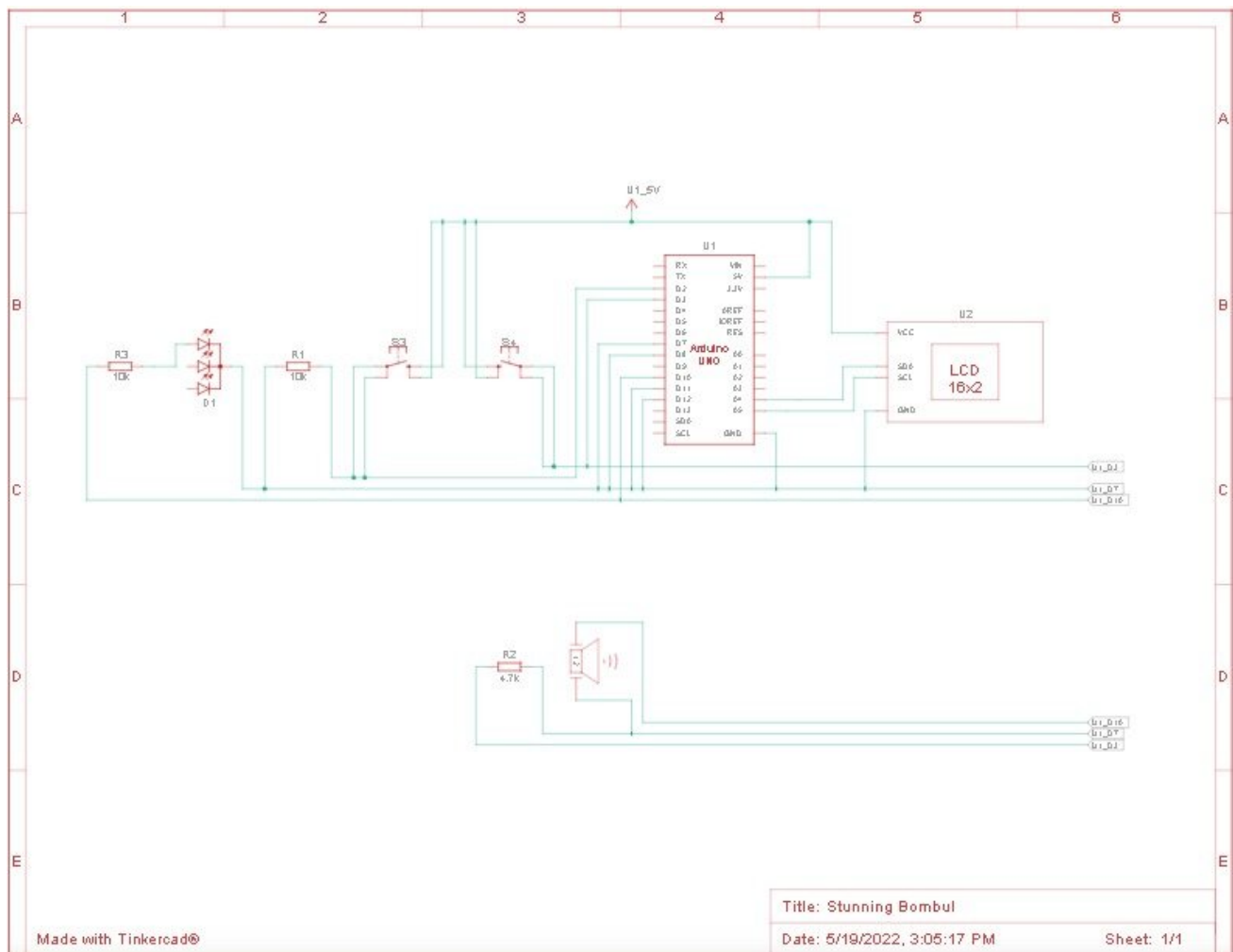
Schema bloc



Utilizatorul are posibilitatea initial de a selecta un mod de joc cu ajutorul a 2 butoane: sandbox sau domination. Se va cere sa se seteze un cronometru si o cheie de dezamorsare (in modul sandbox) sau vor fi generate random (in modul domination). Apoi i se va cere sa introduca codul de armare a bombei cu ajutorul unui numpad si un buzzer va emite un sunet care sa indice faptul ca bomba a fost armata si timpul va incepe sa se scurga. Pentru a dezamorsa bomba utilizatorul trebuie sa decupleze 3 fire in ordinea indicata de cheia de dezamorsare. Daca firele sunt scoase in alta ordine sau nu sunt scoase in timp util se va emite un sunet care sa indice faptul ca bomba a fost detonata. De asemenea va exista posibilitatea de a dezamorsa bomba si prin introducerea codului secret.

Hardware Design

Schema hardware



Lista componente

- 1 Arduino Uno [1]
- 1 LCD 16x2 I2C
- 1 Buzzer
- 1 LED
- 2 butoane cu 4 picioare + 2 rezistori
- Fire dezamorsare

Software Design

Mediu de dezvoltare folosit: Arduino IDE

Biblioteci folosite:

- [LiquidCrystal I2C](#)

Surse și funcții implementate: Codul sursă al proiectului este disponibil [aici](#).

Am ales să implementez acest proiect sub forma unui automat finit de stări a cărei diagramă se poate găsi mai jos. În total am adăugat 22 de stări principale pentru a izola cât mai bine diferitele

funcționalități ale jocului și pentru a oferi o experiență cât mai plăcută și intuitivă utilizatorului. În continuare voi face o scurtă prezentare a flowului prin aceste stări:

1. Selectarea modului de joc: utilizatorul alege dacă vrea să joace modul sandbox sau domination. Se verifică dacă care buton este apăsat, se generează beepul de apăsare și se înaintează în starea corespunzătoare. Dacă este ales domination automatul intră în starea 1, iar dacă este ales sandbox automatul intră în starea 7. Dacă nu este apăsat niciun buton automatul rămâne în starea 0.
2. S-a intrat în modul domination. Se generează random un timer între 20 și 40 secunde. După generare se înaintează în starea 2.
3. Se așteaptă input de la utilizator pentru a înainta în starea 3 prin apăsarea butonului 1.
4. Se generează un număr random între 1 și 25, fiecare corespunzător unei secvențe de culori de fire ce trebuie decuplate în ordinea indicată pentru a dezamorsa bomba. După generarea secvenței de dezamorsare se înaintează în starea 4.
5. Se schimbă lumina ledului și se calculează a 6a respectiv a 3a parte din timpul total (în secunde) avut la dispoziție pentru dezamorsare. Acești timpi sunt necesari pentru a crea un efect de accelerare a beepaitului și ledului atunci când timpul se apropie de final. Se înaintează în starea 5.
6. Se intră în starea de dezamorsare. În această stare cronometrul este afișat pe lcd împreună cu o secvență de 3, 2 respectiv 1 „*” care să indice stadiul deamorsării (câte fire au fost decuplate și câte mai sunt de decuplat). Verificările pentru stadiul deamorsării se fac cu alte 3 stări separate pe care le voi detalia mai jos. În funcție de acțiunile utilizatorului (decuplează firele în ordinea corectă sau greșită, se încadrează sau nu în timp) automatul trece în una din stările finale 6 (bomba a fost detonată) sau 18 (bomba a fost dezamorsată).
7. Dacă se ajunge în starea 6 este game over. Se generează un mesaj de final și un sunet apoi se trece în starea 18 pentru a pregăti jocul de reset.
8. Starea 7 reprezintă de fapt prima stare în care se intră după alegerea modului de joc dacă a fost ales modul sandbox. În această stare se afișează un mesaj care să indice utilizatorului că trebuie să seteze manual timerul. Mesajul se păstrează pe ecranul lcd și se trece în starea 8 (am ales să delimitez aceste 2 stări pentru a elimina efectul de flickering al lcd-ului).
9. Se așteaptă input de la utilizator: dacă se apasă butonul 1 atunci numărul de minute din timer este incrementat, iar dacă se apasă butonul 2 se trece în starea 20 răspunzătoare ce incrementarea numărului de secunde. Timpul minim care poate fi setat în modul sandbox este 1 minut.
10. Starea 9 este prima stare din modulul de configurare a secvenței de dezamorsare.
11. Utilizatorul poate alege în ce ordine trebuie decuplate firele pentru a dezamorsa bomba. În starea 10 se poate itera prin fire cu butonul 1 și se trece mai departe cu butonul 2.
12. În starea 11 se setează primul fir din secvența de dezamorsare (ultimul la care s-a rămas în starea anterioară) și se trece departe. Funcționalitățile butoanelor sunt la fel ca în starea anterioară.
13. În starea 12 se iterează prin cele 3 fire rămase. Funcționalitățile butoanelor sunt similare cu cele din starea 10.
14. În starea 13 se alege firul 2 și se trece mai departe.
15. În starea 14 se iterează prin cele 2 fire rămase. Funcționalitățile butoanelor sunt similare cu cele din starea 10.
16. În starea 15 se setează și ultimul fir din secvență și se afișează secvența finală pentru a aminti utilizatorului ce a ales.
17. În starea 16 se așteaptă input de la utilizator pentru a trece arma bomba și a porni modulul de dezamorsare (se trece în starea 4).
18. Starea 17 este starea de reset în care automatul intră după terminarea unui ciclu complet al

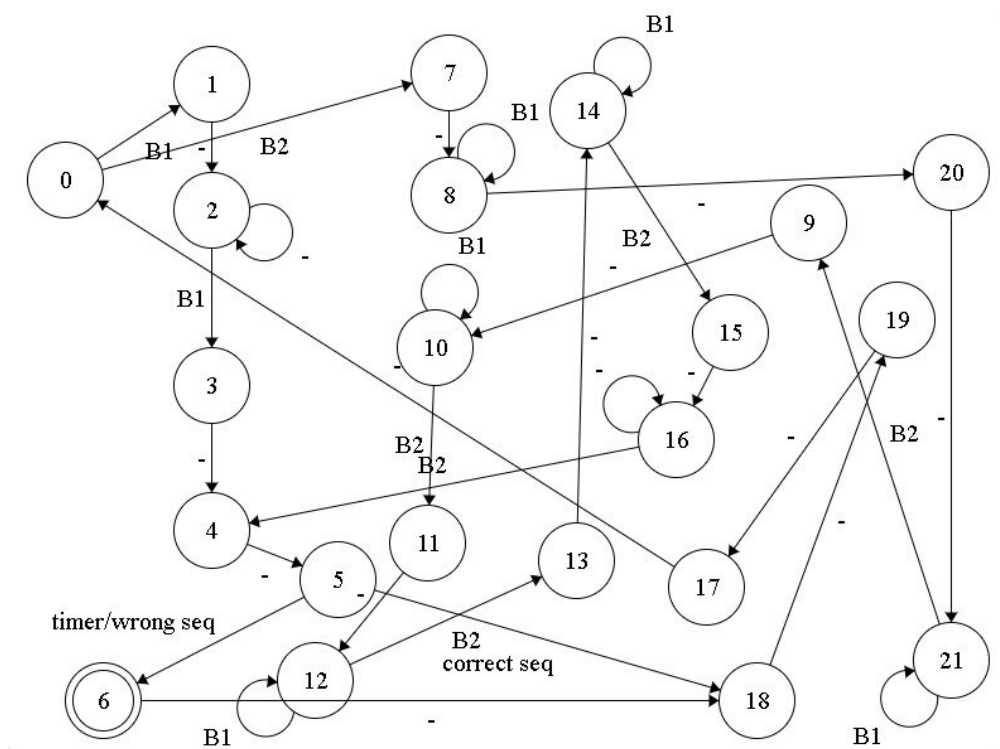
jocului și se pregătește să o ia de la capăt, intervenția fizică a utilizatorului fiind minimă (recuplarea la ground a firelor de dezamorsare decuplate în runda anterioară).

19. Starea 18 este o stare de reset în care se afișează mesajul corespunzător.
20. Starea 19 este o stare de reset în care se afișează timerul de reset (automatul trece în starea 0 după 10 secunde de la terminarea runde).
21. Stările 20 și 21 sunt stări în care se poate seta numărul de secunde din timerul custom din modul sandbox. Acestea au fost adăugate abia la final deoarece acest feature nu era în planul inițial.

Pe lângă cele 22 stări principale am mai folosit și 4 stări în care automatul poate intra odată ce s-a declanșat procesul de dezamorsare. Utilitatea acestora poate fi înțeleasă mai bine în raport cu întreg mecanismul de dezamorsare: Pentru a tine minte care este secvența corectă de fire ce trebuie decuplate m-am folosit de 5 arrayuri de valori boolene. Fiecărui index îi corespunde o culoare stabilită arbitrar: 0 - roșu, 1 - albastru, 2 - verde, 3 - galben. 0 - fir cuplat, 1 - fir decuplat. Unul dintre arrayuri reprezintă secvența curentă de fire. Celelalte 4 reprezintă șabloane de referință pentru defuse stateurile 1, 2 și 3. Sunt populate în funcție de modul de joc ales: în dominaion în mod random, iar în sandbox pe baza firelor alese de utilizator. În funcție de valorile din acest array în starea 4 se determina un defuse state:

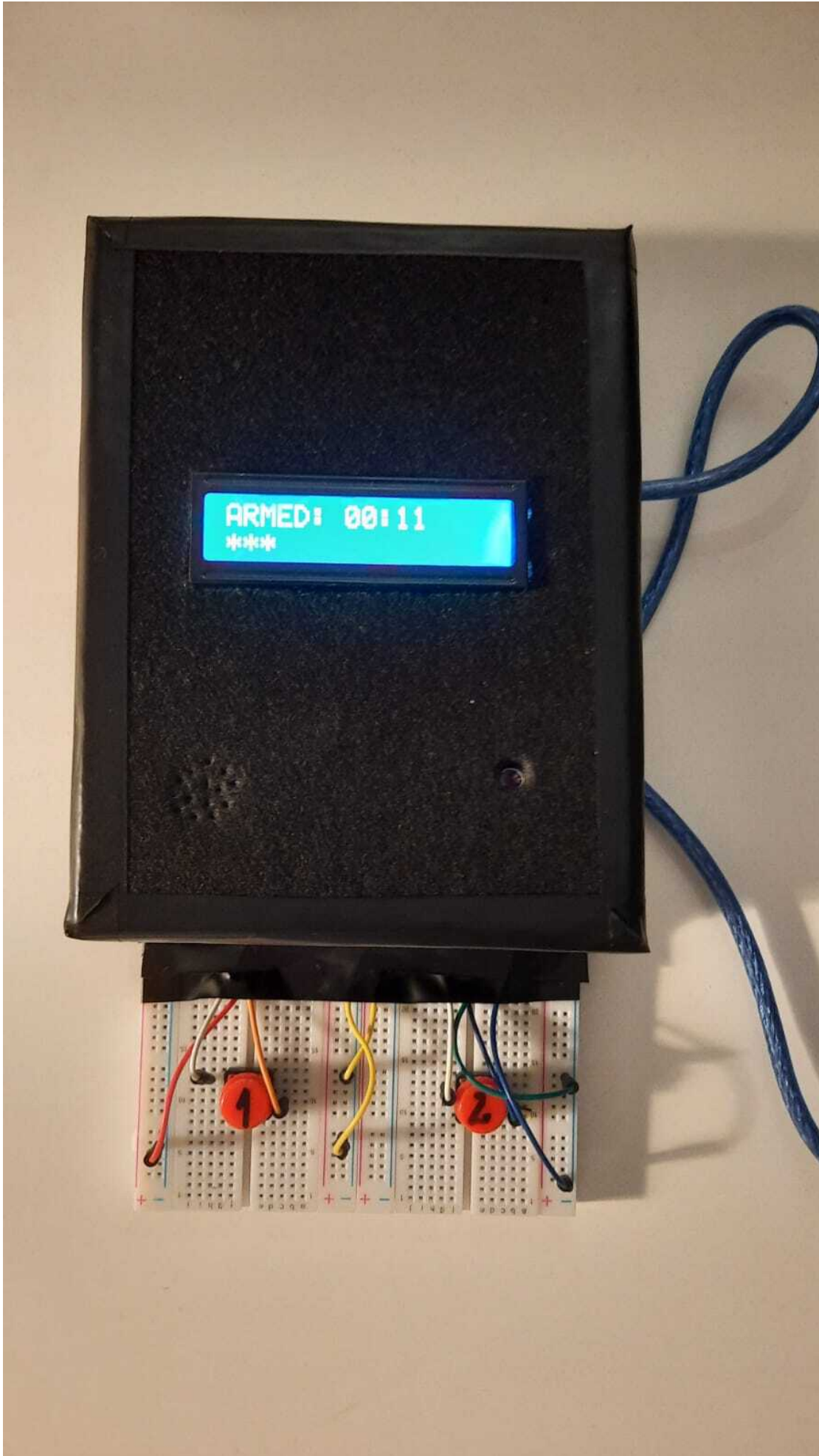
1. Toate firele sunt cuplate - bomba este full armată
2. Un fir a fost decuplat în mod corect
3. Doua fire au fost decuplate în mod corect
4. Toate firele au fost decuplate în mod corect - bomba este dezamorsată

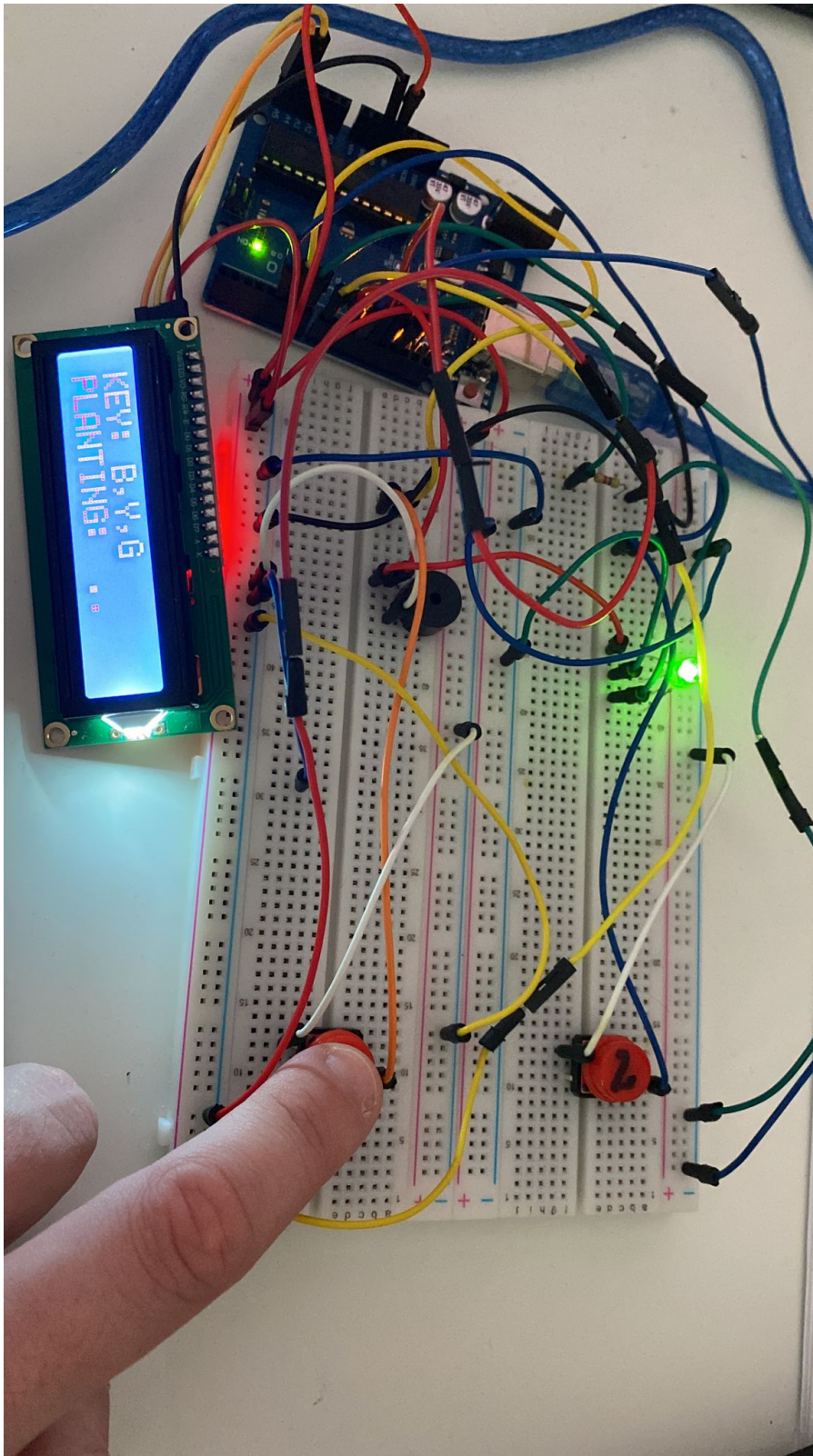
Trecerea dintr-un defuse state în altul se face secvențial. La fiecare pas, în funcție de defuse state se verifică dacă valorile din arrayul defuse stateului curent corespund cu valorile setate în șablonul de referință. Dacă arrayul defuse state curent nu corespunde cu șablonul defuse state atunci se generează detonarea bombei. Defuse stateul curent este determinat prin funcția `void check_D_current()` care face `digitalRead` pe pinii corespunzători firelor de dezamorsare. Acestea sunt declarate cu `pinmode input_pullup`, astfel încât atunci când sunt cuplate `digitalRead` returnează 0, iar când sunt decuplate returnează 1. Valorile sunt afișate de asemenea și la Serial Monitor.



Rezultate obtinute

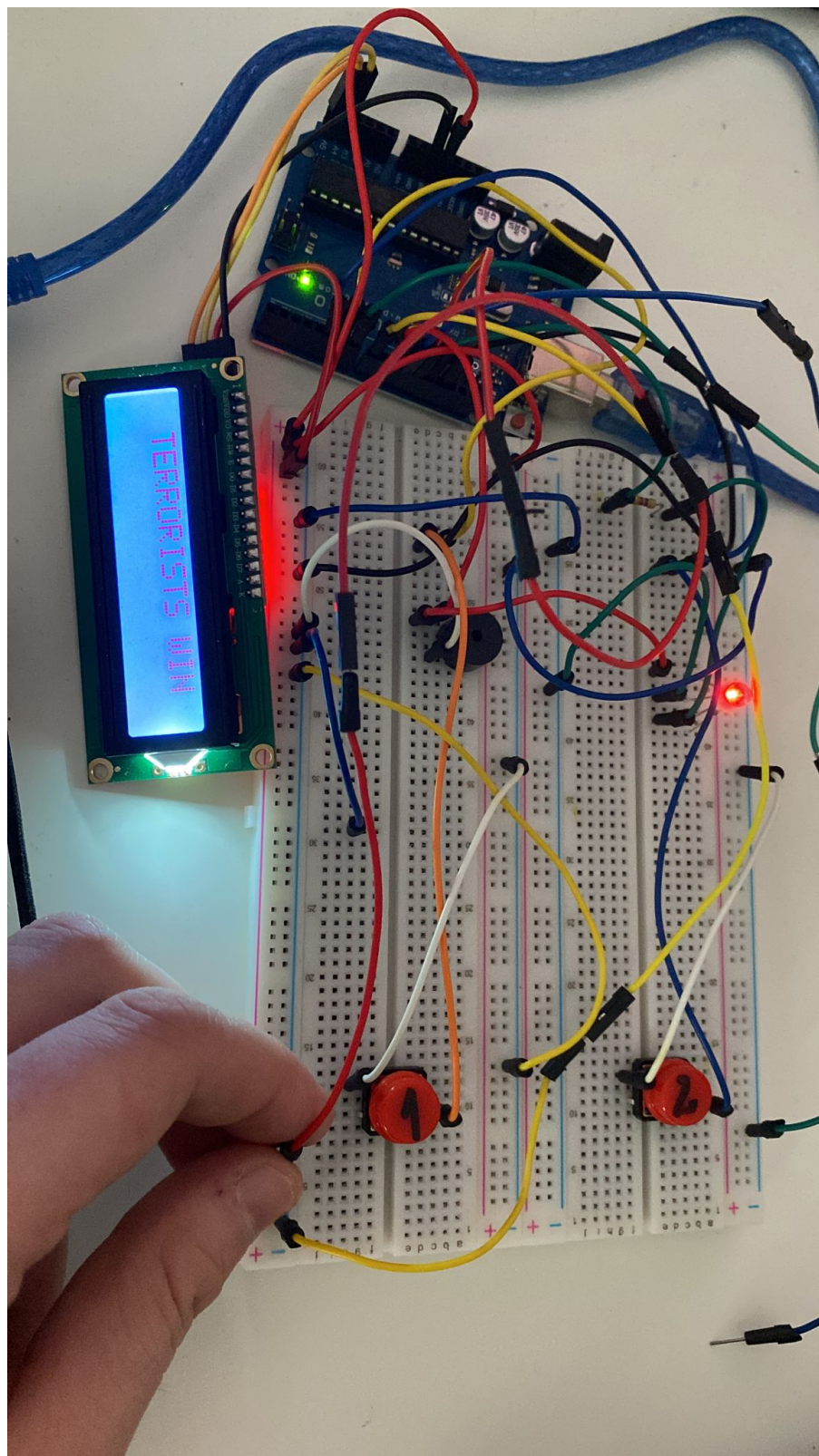
În final am reușit să implementez funcționalitățile de bază pe care mi le-am propus la începutul proiectului, însă cu mici modificări aduse pe parcurs datorită calității îndoielnice a componentelor avute la dispoziție. Atașez mai jos imagini cu proiectul final.





Concluzii

Sunt mulțumit de rezultatul obținut întrucat am reușit să refac un joc drag mie. Am mai lucrat înainte cu Arduino, dar aceasta a fost o provocare. Din păcate nu am reușit să adaug un feature extra, acela



de la timp folosit. Nu am reușit să conectez Modulul GSM (sim800l) pe care am încercat să-l folosesc pentru a realiza legătura cu apelurile telefonice nu a reușit să se conecteze la rețea, deși cablajul și alimentarea le-am realizat după indicațiile din datasheet.

Download

[defuse_the_bomb.zip](#)

Jurnal

01.05.2022 Creare pagina wiki.

12.05.2022 Creare repo [GitHub](#).

14.05.2022 Implementare functionalitati de baza:

1. afisaj
2. generare cheie dezamorsare
3. setare timer
4. joc lumini + sunet
5. dezamorsare cu decuplare fire
6. detonare

Probleme intampinate: efect de bouncing foarte mare pe butoane.

15.05.2022 Adaugare moduri de joc Domination si Sandbox.

19.05.2022 Update schema bloc, schema hardware și documentație.

26.05.2022 Finalizare documentație.

Bibliografie/Resurse

[Atmega328p datasheet](#)[1]

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2022/sgherman/defuse_bomb_puzzle



Last update: **2022/05/26 11:44**