

Flame Fighter

Introducere

Flame Fighter detectează și stinge automat incendiile. Atunci când sistemul identifică semnale de foc, acesta emite un sunet de alarmă și declanșează un jet de apă pentru a îl stinge.

Scopul proiectului este acela de a oferi o soluție eficientă pentru a stinge focul, reducând astfel riscurile de rănire al celor din jur. Acesta poate fi utilizat atât pentru case, pentru autobuze, pentru grădinițe , scoli și la locurile de munca.

Ideea a pornit de la dorința de a crea un sistem de stingere a incendiilor care să fie independent și automat, oferind astfel mai multa siguranță în cazul izbucnirii unui incendiu, mai ales în momentele când nu este nimeni prezent sa îl stingă.

Asigură o protecție suplimentară, intervenind rapid pentru a stinge flăcările. Acest lucru poate salva vieți și reduce semnificativ pierderile materiale.

Descriere generală



Senzorul de foc detectează prezența flăcărilor și transmite un semnal către Arduino UNO. Arduino UNO procesează semnalul de la senzor și activează Relay Module. Relay Module controlează atât Buzzer-ul care creează semnalul de alarma, cât și Water Pump pentru a stinge focul. Bateriile reîncărcabile furnizează energia necesară pentru întregul sistem, alimentând Relay Module, Buzzer, și Water Pump, respectiv Arduino UNO.

Hardware Design

Lista de piese

1. Flame Sensor Module LM393
2. Arduino Uno
3. Buzzer
4. Pipe 1m
5. Water Pump
6. Relay Module 5V - 1 Channel

7. 2 x Rechargeable Battery

Schema Electrica



Componente și rolul lor

Arduino Uno:

- Coordonează funcționarea întregului sistem pe baza programului încărcat pe acesta. De aici pleacă semnalele pentru controlul celorlalte dispozitive. Furnizează tensiune pentru Relay Module și Flame Sensor.

Flame Sensor:

- Acesta detectează radiația infraroșie și transmite semnale la Arduino, pe baza detecției flăcării.

Rechargeable Battery:

- Surse de alimentare a circuitului:

```
BT1:  
Tensiune pentru motorul DC și Buzzer.  
BT2:  
Alimentare principală pentru placa Arduino.
```

Water Pump:

- Porneste pompa de apă.

Buzzer:

- Emite un sunet pentru a semnaliza detectarea flăcărilor.

Relay Module:

- Permite controlul pompei de apă și al Buzzer-ului prin semnalele primite de la Arduino.

Pini

Arduino:

```
D13: Utilizat pentru a controla modulul de releu. Arduino trimite  
semnalele pentru activarea/dezactivarea releului  
Vin: Pentru a alimenta placa Arduino cu tensiunea de la baterie (7.4V)  
A0: Pinul A0 este un pin analogic pe Arduino care poate citi semnale  
digitale de la senzorul de flacăra.
```

BT1:

```
+7.4V: Conectat la pinul NO al relay-ului  
GND: Conectat la pinul GND
```

BT2:

```
+7.4V: Conectat la pinul VIN al Arduino-ului  
GND: Conectat la pinul GND
```

Flame Sensor:

```
VCC: Conectat la pinul de 3.3V al Arduino-ului pentru alimentare  
GND: Conectat la pinul GND  
DO (Digital Output): Conectat la pinul A0 al Arduino pentru a trimite  
semnale digitale către Arduino când detectează flăcăra
```

Motorul DC (M1):

```
+: Conectat la pinul COM al relay-ului  
-: Conectat la pinul GND
```

Buzzer (BZ1):

```
+: Conectat la pinul COM al relay-ului  
-: Conectat la pinul GND
```

Relay Module:

```
VCC: Conectat la pinul 5V al Arduino pentru alimentare  
GND: Conectat la pinul GND  
IN: Conectat la pinul digital 13 al Arduino-ului pentru a primi semnale  
de control (activarea și dezactivare relay)  
COM (Common): Conectat la motorul DC și Buzzer.  
NO (Normally Open): Conectat la pinul + al bateriei BT1.
```

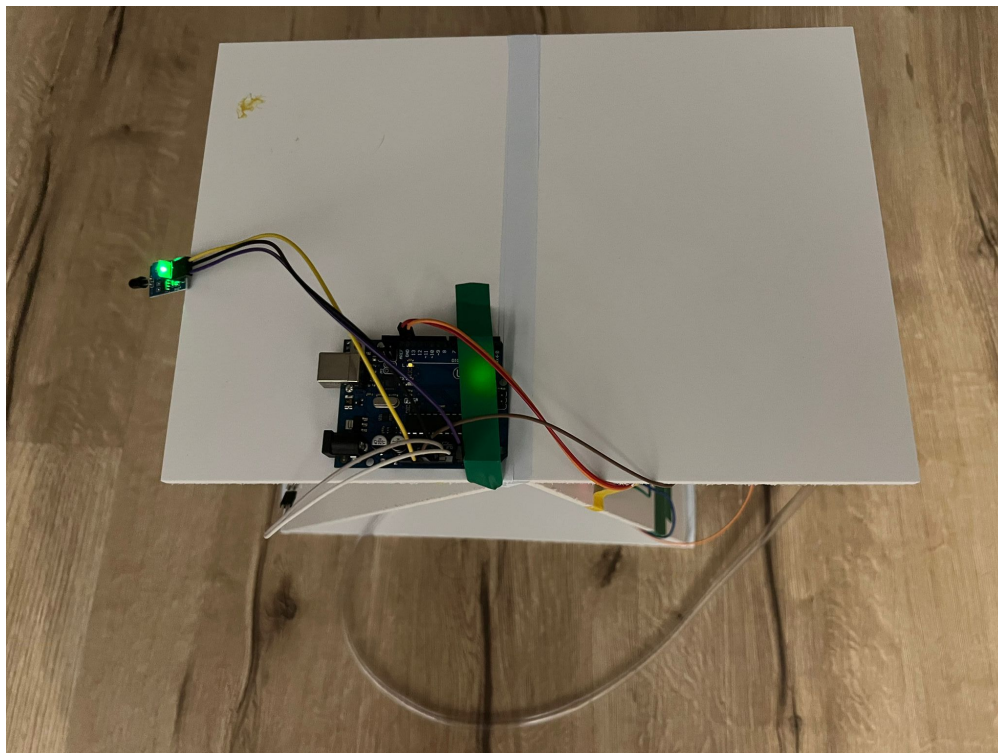
Functionare

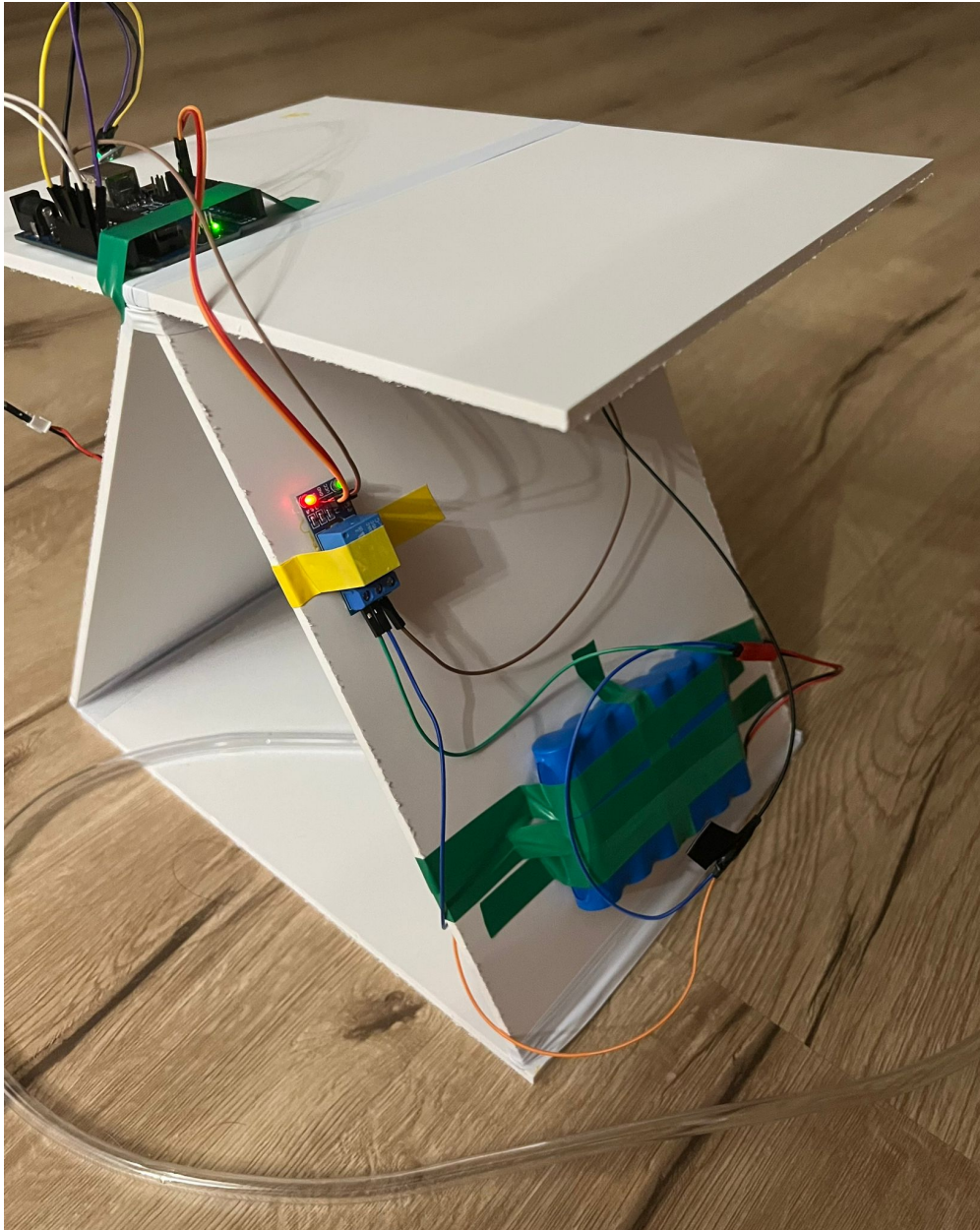
Bateria BT2 de 7.4V furnizează tensiune pentru placa Arduino. Modulul releu și senzorul de flăcără sunt alimentate prin intermediul pinilor de alimentare ai plăcii Arduino. Dacă senzorul detectează o flăcără, el va trimite un semnal digital la pinul A0 al Arduino-ului. Arduino trimite un semnal la pinul 13, activând releul. Când releul este activat, pinul NO conectează motorul și buzzer-ul la la BT1. Astfel se pornește pompa de apă pentru a stinge flăcăra și buzzer-ul pentru a declanșa alarma.

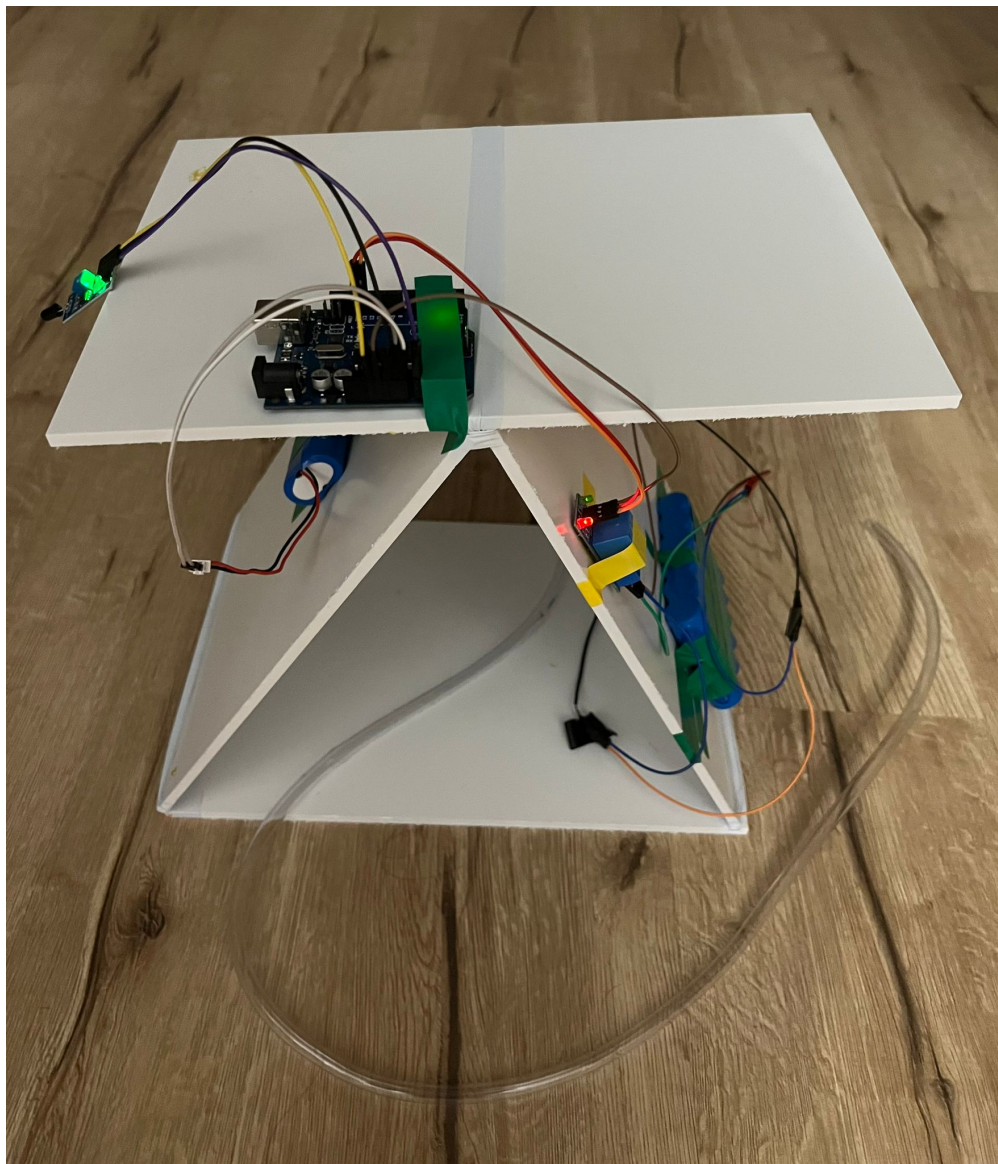
Folosirea a doua baterii

BT2 alimentează Arduino. Este nevoie și de bateria BT1 deoarece motorul poate consuma mult curent. Bateria BT1 este conectată la pinul NO și furnizează tensiune doar când releul este activ. Astfel Arduino poate controla activarea și dezactivarea motorului fără a afecta sursa de alimentare a propriului circuit.

Stadiul actual







Software Design

Mediul de dezvoltare: Arduino 1.8.19

Cod:

```
int sensor=0;
int relay= 13;
void setup()
{
  pinMode(relay,OUTPUT);
  pinMode(sensor,INPUT);
  Serial.begin(9600);
}
void loop()
{
  int val = analogRead(sensor);
```

```
Serial.println(val);
if (val <= 100) {
  digitalWrite(relay, LOW);
  delay(4500);
} else {
  digitalWrite(relay, HIGH);
}
delay(500);
}
```

Variabilele sensor și relay

- Sunt utilizate pentru a stoca pinii la care sunt conectate senzorul (conectat la pinul analogic A0) și relay-ul (conectat la pinul digital D13)

Funcția setup()

pinMode(relay, OUTPUT): Configurează pinul relay (pinul 13) ca ieșire. Acest pin va fi utilizat pentru a controla releul

pinMode(sensor, INPUT): Configurează pinul sensor (pinul 0) ca intrare. Acest pin va fi utilizat pentru a citi valorile de la senzor

Serial.begin(9600): Inițializează comunicarea serială la o rată de transfer de 9600 baud. Aceasta permite trimiterea datelor către Serial Monitor pentru a vizualiza valorile senzorului

Astfel cu cel de input voi putea citi date, iar cu cel de output pot trimite semnale de tensiune

Funcția loop()

Este utilizată pentru a rula în mod continuu codul principal al programului.

int val = analogRead(sensor): Citește valoarea analogică de la senzorul conectat la pinul sensor (pinul A0) și o stochează în variabila val

Serial.println(val): Trimite valoarea citită de la senzor către Serial Monitor pentru a fi afișată

if (val <= 100): Verifică dacă valoarea citită de la senzor este mai mică sau egală cu 100

digitalWrite(relay, LOW): Dacă valoarea este mai mică sau egală cu 100, setez pinul relay (pinul 13) la LOW, ceea ce va porni buzzer-ul și motorul

delay(4500): sa funcționeze mai mult timp

digitalWrite(relay, HIGH): Dacă valoarea este mai mare de 100, setez pinul relay la HIGH, ceea ce va opri buzzer-ul și motorul

delay(500): Pune în așteptare execuția codului pentru 500 de milisecunde înainte de a repeta bucla

Utilizarea functionalitatilor din laborator in cadrul proiectului:

Laborator 0 - GPIO: <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2023>

- Am utilizat funcționalități GPIO pentru a releul care gestionează atât un buzzer, cât și un motor.

Utilizarea GPIO-urilor în proiectul meu este esențială pentru protecția plăcii Arduino.

Setarea Pinilor Controlul Releului

Am folosit releul pentru a nu se folosi aceeași sursă de alimentare pentru motor, cu Arduino.
Pentru că dacă foloseau aceeași sursă de alimentare exista posibilitatea de a arde placa Arduino.
Arduino poate furniza doar un curent limitat.
Motoarele pot necesita curenți mult mai mari decât atât.
Dacă ar folosi aceeași sursă de alimentare ca Arduino și ar trage prea mult curent, ar putea arde placa Arduino.
Utilizarea unui releu asigură să nu treacă curentul mare necesar pentru motor prin placa Arduino.

Laborant 1 - UART: <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2023>

```
Serial.begin(9600): Inițializează comunicația serială la 9600 baud.  
Serial.println(): Folosita pentru a trimite date de la Arduino către laptop.
```

Am folosit funcțiile de comunicație serială pentru a monitoriza valorile citite de la senzor.

Laborator 4 - ADC: <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2023-2024>

```
analogRead() pentru a citi valorile analogice de la senzor.
```

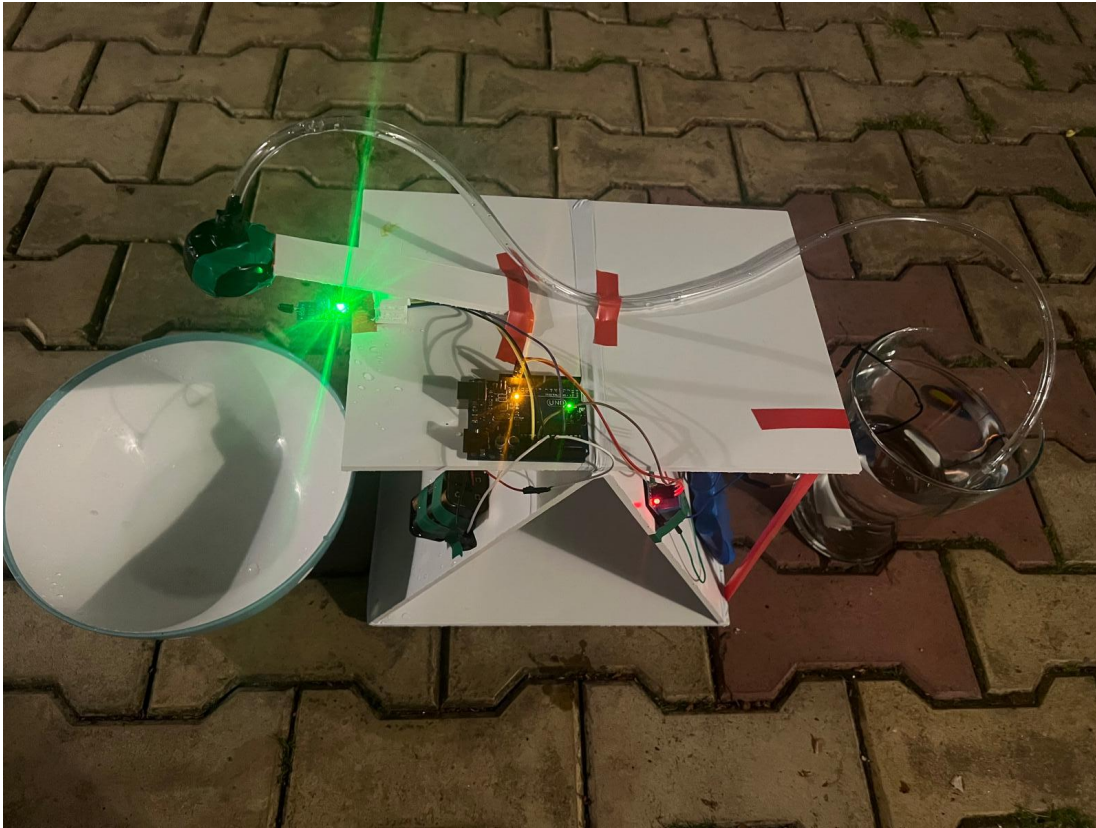
Astfel valorile analogice sunt convertite în valori digitale.

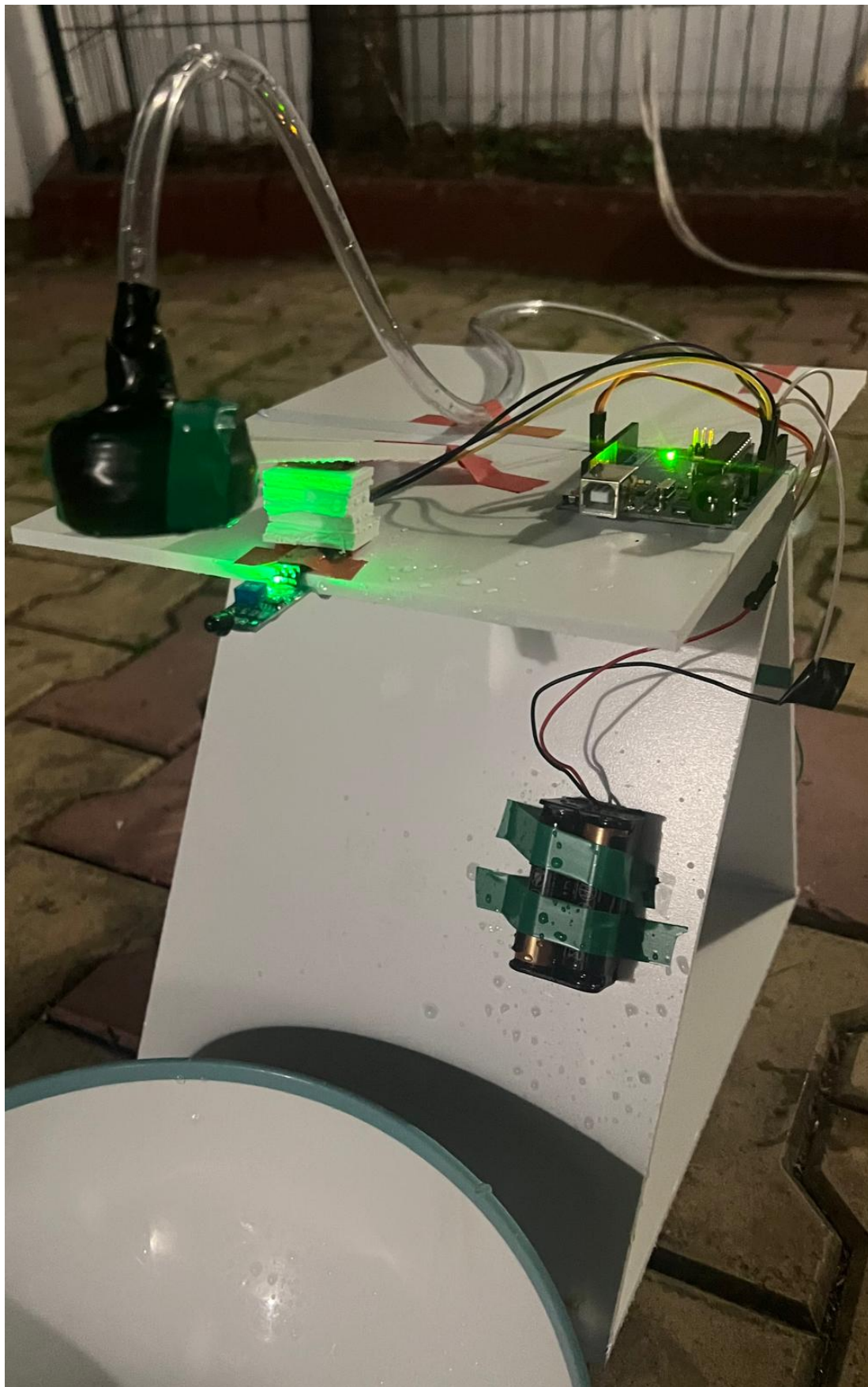
Video: <https://youtube.com/shorts/-dSsuAELeEU?si=m3XrS77JXSJUmaag>

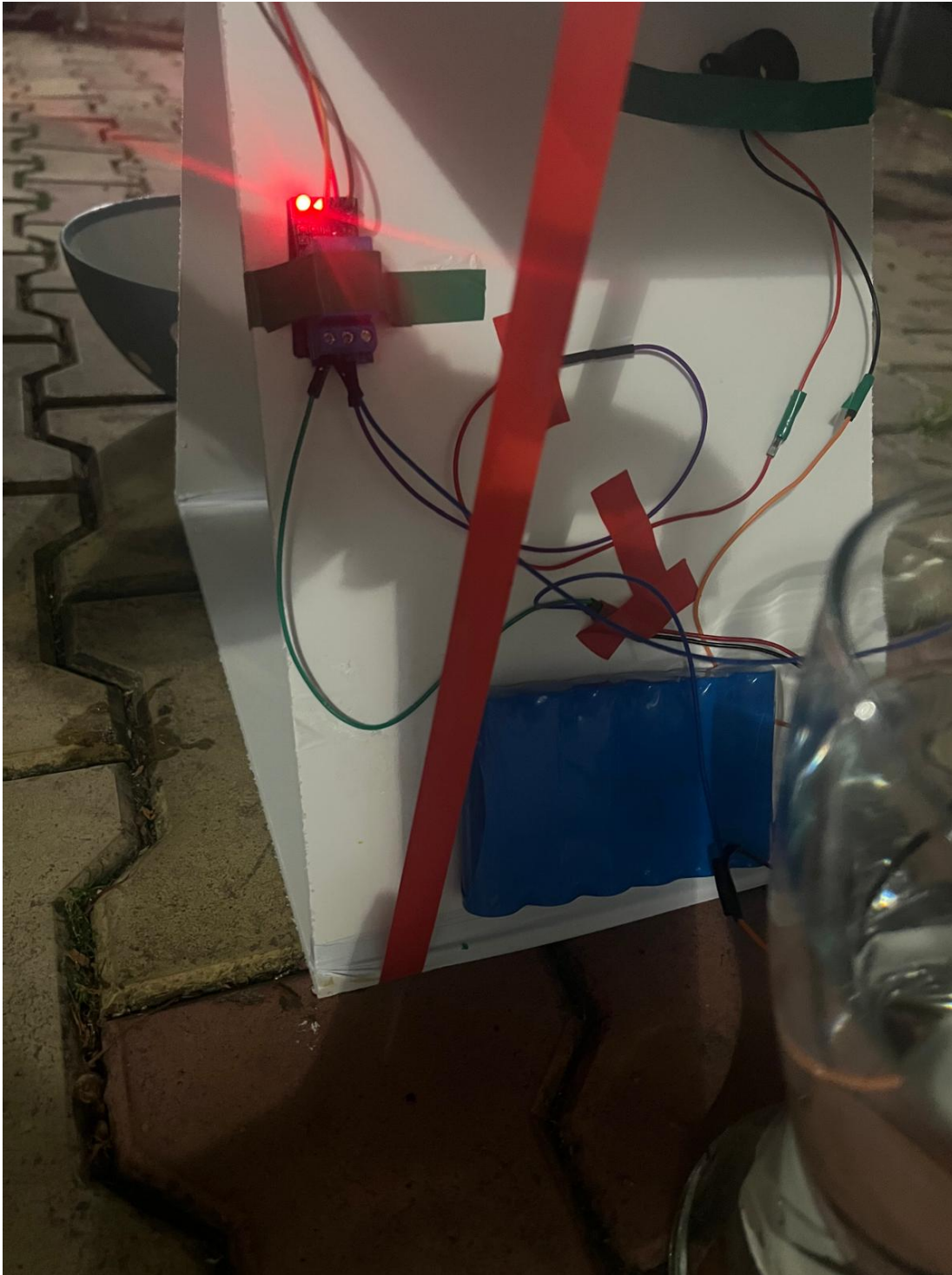
Rezultate Obținute

Proiectul reprezintă un extingtor automat de foc. Atunci când senzorul detectează prezența unei flăcări, motorul pornește, activând un jet de apă care stinge focul. Simultan, se activează și un buzzer pentru a emite un semnal sonor de alarmă, semnalând astfel prezența focului.

Stare finală:







Concluzii

Proiectul funcționează corect, îndeplinește ideea principală de la care am plecat atunci când am vrut să îl creez. Am învățat cum să folosesc Kicad. De asemenea, am învățat și cum să lipesc firele electrice. Utilizarea unui relee a fost necesară pentru a conecta motorul la o sursă de alimentare diferită de cea a plăcii Arduino, asigurând astfel protecția acesteia împotriva unei posibile daune.

Download

[flame_fighter_niculae_karla_333cc.zip](#)

Bibliografie/Resurse

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

<https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-serial-monitor/>

<https://www.hnhcart.com/blogs/sensors-modules/relay-module>

<https://www.circuitbasics.com/what-is-a-buzzer/>

<https://besomi.com/product/lm393-flame-sensor/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/aungureanu/karla.niculae>



Last update: **2024/05/27 00:30**