

3. Rezumatul lucrării (maxim 12 pagini) va conține obligatoriu următoarele informații minime:

a) Titlul proiectului

## **Fire Fighter**

b) Secțiune

### **Tehnologia informației**

c) Categorie

### **Juniori**

d) Scopul;

Am ales această temă deoarece în România se întâmplă foarte multe incidente din pricina focului, astfel ca noi am decis să creăm acest robot pentru a pune priceperea în salvarea unor vieți.

e) Obiective;

Proiectul nostru constă într-un robot care poate detecta focul prin intermediul unui senzor de foc/căldură și pe care îl poate stinge cu ajutorul unei pompe și a unui servomotor care să direcționeze apa către foc. De asemenea, Fire Fighter se poate deplasa către locul incendiului pe roți acționate de motoare. Scurgerile de gaze sunt detectate cu ajutorul unui senzor de gaz. Astfel, atunci când Fire Fighter detectează un incendiu sau o scurgere de gaze, acesta va avertiza proprietarul printr-un sms.

f) Problema identificată spre rezolvare;

Noi am observat că sute de oameni mor în incendii și de obicei motivul este faptul că nu au timp să apeleze 112 și pompierii nu ajung destul de devreme, iar noi am găsit o soluție.

g) Echipa de proiect (nume, prenume elev, clasă și unitatea școlară), cu precizarea rolurilor fiecărui membru în echipă. Implicarea altor persoane în proiect/colaboratori externi și motivul implicării;

Membrii echipei sunt xxxxx și xxxxxxxx. Xxxxxx s-a ocupat de codarea senzorilor de flacără, senzorului de gaz, a buzzerului, motor driverului, a releului și pompei, servo motorului și a motoarelor. Xxxxxx s-a ocupat de implementarea tehnicii de trimitere a sms-urilor.

h) Coordonatorul echipei (nume, prenume, instituția reprezentată) și eventualele persoane colaboratoare/sușinătoare, implicate în proiect/colaboratori externi și motivul implicării;

xxxxxxx a fost coordonatorul acestei echipe iar xxxxxxx este instituția reprezentată. Aceasta a fost singura persoană implicată în acest proiect, ajutându-ne cu rezolvarea erorilor.

i) Etape parcurse;

Noi mai întâi am lucrat la baza proiectului, începând astfel să îi adăugăm componente și să ajungem la rezultatul final.

Pentru a realiza proiectul nostru și a codifica placa Arduino UNO, am folosit platforma **Arduino**. Pe această platformă codarea se face prin limbajul de programare C++.

**Arduino** este o companie open-source care produce atât plăcuțe de dezvoltare bazate pe microcontrolere, cât și partea de software destinată funcționării și programării acestora. Pe lângă acestea include și o comunitate uriașă care se ocupă cu creația și distribuirea de proiecte care au ca scop crearea de dispozitive care pot sesiza și controla diverse activități sau procese în lumea reală.

**SIM800L** este un modul celular miniatural care permite transmisia GPRS, trimiterea și primirea SMS-urilor și efectuarea și primirea apelurilor vocale. Costul redus și amprenta redusă și suportul de frecvență quad band fac din acest modul soluția perfectă pentru orice proiect care necesită conectivitate pe distanță lungă. După conectarea modului de alimentare pornește, caută rețeaua celulară și se autentifică automat. LED-ul de la bord afișează starea conexiunii.

Servomotoarele nu sunt o clasă specifică de motoare, deși termenul servomotor este adesea folosit pentru a se referi la un motor adecvat pentru utilizare într-un sistem de control în buclă închisă. Servomotoarele sunt utilizate în aplicații precum robotică, mașini CNC și producție automată.

Un motor este un dispozitiv electronic care ajută la transformarea energiei electrice în energie mecanică. Prin urmare, un driver de motor permite efectuarea unor lucrări automate folosind energie electrică. Există mai multe tipuri diferite de motoare electrice. Aceste tipuri includ motoarele de curent continuu, motoarele pas cu pas și motoarele servo. Principiile și caracteristicile lor de funcționare sunt factorii de diferențiere între aceste motoare

O **pompă** este o mașină sau un aparat care transformă energia, dintr-una din formele sale mecanice, în formă de energie hidraulică sau pneumatică, în scopul transportării fluidului care primește energia utilă. Energia mecanică poate proveni din forța musculară sau de la un motor de antrenare.

O **placă de breadboard** este o placă de plastic dreptunghiulară cu o grămadă de găuri mici în ea. Aceste găuri vă permit să introduceți cu ușurință componente electronice în prototipul (adică să construiți și să testați o versiune timpurie a) unui circuit electronic, ca acesta cu o baterie, comutator, rezistor și un LED.

**L298N motor driver este un modul pentru controlul motoarelor.** Cu ele puteți utiliza coduri simple pentru programează placa noastră Arduino și să poată controla motoarele de curent continuu într-un mod simplu și controlat. În general, acest tip de modul este utilizat mai mult în robotică sau în actuatori care utilizează motoare, deși poate fi folosit pentru o multitudine de aplicații.

Relev este o componentă electronică, un dispozitiv, care produce anumite modificări (cum ar fi închiderea și deschiderea unui circuit) pe baza unui parametru care variază (precum tensiunea electrică aplicată), permițând controlarea unui curent de intensitate mare cu ajutorul unui curent de intensitate mică.

Modulul LM2596 are un contor de tensiune și un voltmetru ce afișează autocalibrarea. Utilizează o eroare de voltmetru de tensiune microprocesor mai avansată  $\pm 0.05V$ , domeniu  $0 \sim 40V$ . (Pentru a asigura voltmetrul de precizie, asigurați-vă că tensiunea de intrare este de 4.5V sau mai mare).

Un **detector de gaze** este un aparat dotat cu senzori care permit detectarea prezenței unor gaze în mediul ambiant.

Acest tip de detector este folosit frecvent în echipamente de siguranță, pentru a detecta apariția (scăpările) de gaze toxice, inflamabile etc..., astfel încât acestea să nu atingă niveluri periculoase pentru sănătate și/sau incinta respectivă și sursa de gaz să fie oprită automat (cum este cazul în care focul alimentat de sursa respectivă s-a întrerupt).

Modulul constă într-un buzzer piezoelectric pasiv, care poate genera tonuri între 1,5 și 2,5 kHz prin comutarea și oprirea la frecvențe diferite.

Codul folosit de noi este urmatorul

```
#include <Servo.h>
#include <SoftwareSerial.h>
Servo servo;
int pompa = 12;
int Buzzer = 11;
#define sensor A0
int motor1pin1 = 2;
int motor1pin2 = 3;
int motor2pin1 = 4;
int motor2pin2 = 5;
int isFlamePin1 = 7;
int isFlamePin2 = 9;
int isFlamePin3 = 10;

int isFlame1 = HIGH;
int isFlame2 = HIGH;
int isFlame3 = HIGH;
int angle = 10;

int gsmPin1 = 6;
int gsmPin2 = 13;
SoftwareSerial myGSMSerial(gsmPin1, gsmPin2);
//Mai sus am definit pini pe care i-am folosit
void updateSerial()
{
    delay(500);
    while (Serial.available())
    {
        myGSMSerial.write(Serial.read());
    }
    while(myGSMSerial.available())
    {
        Serial.write(myGSMSerial.read());
    }
}

void sendSMSflame()
{
    myGSMSerial.println("AT"); //Once the handshake test is successful, it will
    back to OK
    updateSerial();
    myGSMSerial.println("AT+CMGF=1"); // Configuring TEXT mode
    updateSerial();
    myGSMSerial.println("AT+CMGS=\"+40741790460\""); // enter your phone number
    here (prefix country code)
    updateSerial();
    myGSMSerial.print("Alert,flame detected"); // enter your message here
```

```

    updateSerial();
    myGSMSerial.write(26);
    updateSerial();

}

void sendSMSgas()
{
    myGSMSerial.println("AT"); //Once the handshake test is successful, it will
back to OK
    updateSerial();
    myGSMSerial.println("AT+CMGF=1"); // Configuring TEXT mode
    updateSerial();
    myGSMSerial.println("AT+CMGS=\"+40741790460\""); // enter your phone number
here (prefix country code)
    updateSerial();
    myGSMSerial.print("Alert,gas detected"); // enter your message here
    updateSerial();
    myGSMSerial.write(26);
    updateSerial();

}

//Mai sus am definit sms-urile timise de SIM 800 L

void setup() {
    pinMode(motor1pin1, OUTPUT);
    pinMode(motor1pin2, OUTPUT);
    pinMode(motor2pin1, OUTPUT);
    pinMode(motor2pin2, OUTPUT);
    pinMode(pompa,OUTPUT);
    pinMode(Buzzer, OUTPUT);
    pinMode(isFlamePin1, INPUT);
    pinMode(isFlamePin2, INPUT);
    pinMode(isFlamePin3, INPUT);
    Serial.begin(9600);
    myGSMSerial.begin(9600);
    servo.attach(8);
    servo.write(angle);

}

void loop(){
    int value = analogRead(sensor);
    updateSerial();
    if (value >= 350){
        sendSMSgas();
        digitalWrite(Buzzer, HIGH);
        delay(500);
        digitalWrite(Buzzer,HIGH);
    }
}

```

```

    delay(500);
}
else {
    digitalWrite(Buzzer, LOW);
}
isFlame1 = digitalRead(isFlamePin1);
isFlame2 = digitalRead(isFlamePin2);
isFlame3 = digitalRead(isFlamePin3);
// Aici am definit actiunea senzorului de gaz.
if (isFlame1 == LOW)
{
    sendSMSflame();
    digitalWrite(motor1pin1,LOW);
    digitalWrite(motor1pin2, HIGH);
    digitalWrite(motor2pin1, LOW);
    digitalWrite(motor2pin2, HIGH);
    delay(2000);
    digitalWrite(motor1pin1, LOW);
    digitalWrite(motor1pin2, LOW);
    digitalWrite(motor2pin1, LOW);
    digitalWrite(motor2pin2, LOW);
    Serial.println("FLAME, FLAME, FLAME 1 ");
    digitalWrite(Buzzer, HIGH);
    digitalWrite(pompa,LOW);
    delay(200);
    digitalWrite(pompa,HIGH);
    for(angle = 10; angle < 180; angle++)
    {
        servo.write(angle);
        digitalWrite(pompa,LOW);
        delay(20);
        digitalWrite(pompa,HIGH);
    }
    // now scan back from 180 to 0 degrees
    for(angle = 180; angle > 10; angle--)
    {
        servo.write(angle);
        digitalWrite(pompa,LOW);
        delay(20);
        digitalWrite(pompa,HIGH);
    }
}
// now scan back from 180 to 0 degrees

else
{
    Serial.println("no flame");
    digitalWrite(Buzzer, LOW);
    digitalWrite(8,LOW);
}

```

```

    digitalWrite(motor1pin1, LOW);
    digitalWrite(motor1pin2, LOW);

    digitalWrite(motor2pin1, LOW);
    digitalWrite(motor2pin2, LOW);
}
// Mai sus am definit ce se va intampla in cazul in care senzorul din mijloc
va simti flacara.

```

```

    if (isFlame2 == LOW)
    {
        sendSMSflame();
        digitalWrite(motor1pin1, LOW);
        digitalWrite(motor1pin2, LOW);
        digitalWrite(motor2pin1, LOW);
        digitalWrite(motor2pin2, HIGH);
        delay(1000);
        digitalWrite(motor1pin1, LOW);
        digitalWrite(motor1pin2, LOW);
        digitalWrite(motor2pin1, LOW);
        digitalWrite(motor2pin2, LOW);
        Serial.println("FLAME, FLAME, FLAME 2 ");
        digitalWrite(Buzzer, HIGH);
        digitalWrite(pompa, LOW);
        delay(200);
        digitalWrite(pompa, HIGH);
        for(angle = 10; angle < 180; angle++)
        {
            servo.write(angle);
            digitalWrite(pompa, LOW);
            delay(20);
            digitalWrite(pompa, HIGH);
        }
        // now scan back from 180 to 0 degrees
        for(angle = 180; angle > 10; angle--)
        {
            servo.write(angle);
            digitalWrite(pompa, LOW);
            delay(20);
            digitalWrite(pompa, HIGH);
        }
    }
    // now scan back from 180 to 0 degrees

else
{
    Serial.println("no flame");
    digitalWrite(Buzzer, LOW);
    digitalWrite(8, LOW);
}

```

```

digitalWrite(motor1pin1, LOW);
digitalWrite(motor1pin2, LOW);
digitalWrite(motor2pin1, LOW);
digitalWrite(motor2pin2, LOW);
}
// Mai sus am definit ce se va intampla in cazul in care un senzor din lateral
va simti flacara.

    if (isFlame3 == LOW)
    {
        sendSMSflame();
        digitalWrite(motor1pin1, LOW);
        digitalWrite(motor1pin2, HIGH);
        digitalWrite(motor2pin1, LOW);
        digitalWrite(motor2pin2, LOW);
        delay(1000);
        digitalWrite(motor1pin1, LOW);
        digitalWrite(motor1pin2, LOW);
        digitalWrite(motor2pin1, LOW);
        digitalWrite(motor2pin2, LOW);
        Serial.println("FLAME, FLAME, FLAME 3");
        digitalWrite(Buzzer, HIGH);
        digitalWrite(pompa, LOW);
        delay(200);
        digitalWrite(pompa, HIGH);
        for(angle = 10; angle < 180; angle++)
        {
            servo.write(angle);
            digitalWrite(pompa, LOW);
            delay(20);
            digitalWrite(pompa, HIGH);
        }
        // now scan back from 180 to 0 degrees
        for(angle = 180; angle > 10; angle--)
        {
            servo.write(angle);
            digitalWrite(pompa, LOW);
            delay(20);
            digitalWrite(pompa, HIGH);
        }
    }
    // now scan back from 180 to 0 degrees
else
{
    Serial.println("no flame");
    digitalWrite(Buzzer, LOW);
    digitalWrite(8, LOW);
    digitalWrite(motor1pin1, LOW);
    digitalWrite(motor1pin2, LOW);
}

```

```
    digitalWrite(motor2pin1, LOW);  
    digitalWrite(motor2pin2, LOW);  
  }  
}  
//Aici am definit ce se va intampla daca un senzor din lateral va simti  
flacara.
```

k) Date experimentale și detalii esențiale ale experimentelor;

l) Concluzii;

In concluzie acest mic roboțel va fii de ajutor in toate casele in care va fii amplasat si poate salva vieți.

m) Anexe (imagini, grafice, scheme etc).