**Documentație Proiect: Detector de Fum cu Senzor MQ-7**

Indrumator: Paula Elena Stef

Student: Jarda Adina-Ionela

**Cuprins**

1. **Introducere: scop si motivatie**
2. **Cercetare Bibliografica**
3. **Solutia propusa si implementarea**
4. **Testare si Validare**
5. **Concluzii**

**1. Introducere: scop și motivație**

În cadrul acestui proiect, am ales să dezvolt un detector de fum bazat pe senzorul **MQ-7**, care este capabil să detecteze monoxidul de carbon (CO) în atmosferă. Scopul principal al acestui proiect este să creez un sistem simplu, eficient și fiabil care poate avertiza utilizatorul despre prezența unui nivel periculos de monoxid de carbon în aer.

Am ales acest proiect deoarece monoxidul de carbon este un gaz incolor, inodor și foarte periculos, care poate fi greu de detectat fără echipamente adecvate. De multe ori, oamenii nu sunt conștienți de nivelurile periculoase ale acestui gaz din cauza lipsei unui miros sau a unui gust distinct. Astfel, o soluție accesibilă și eficientă pentru monitorizarea acestuia poate contribui la siguranța locuinței sau a locurilor de muncă.

Prin implementarea acestui sistem, am dorit să explorez și să apreciez capacitatea senzorilor de gaz de a detecta substanțe periculoase, precum și cum aceste date pot fi utilizate pentru a activa un mecanism de avertizare.

**2. Cercetare bibliografică**

Există mai multe soluții deja implementate care folosesc senzori pentru detecția monoxidului de carbon, dar cele mai multe sunt destinate sistemelor industriale și pot fi costisitoare și complexe. În plus, multe dintre ele nu includ un afișaj clar al nivelurilor de CO, având doar semnale de alarmă sonore sau vizuale. Comparativ cu soluțiile comerciale, proiectul meu are ca avantaj prețul scăzut și implementarea ușoară, având totodată o componentă vizuală (LCD) care ajută utilizatorul să înțeleagă rapid nivelul gazului detectat.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Soluție** | **Cost** | **Dificultate implementare** | **Consum de energie** | **Adaptabilitate** |
| Sistem comercial CO | Ridicat | Mare | Ridicat | Limmitata |
| Detector de CO cu Arduino | Mic | Mic | Mic | Mare |

**3. Soluția propusă și implementarea**

**Descrierea generală a soluției**

Soluția propusă constă într-un detector de monoxid de carbon folosind senzorul MQ-7, un LED care semnalizează pericolul, un buzzer care emite un sunet alertant și un afișaj LCD care arată mesajul "EXIT" atunci când nivelul de CO depășește pragul stabilit.

**Descrierea teoretică a algoritmului**

Algoritmul principal urmărește să citească valorile de pe senzorul MQ-7 și să le compare cu un prag prestabilit (în acest caz, valoarea de 100). Dacă valoarea citită depășește acest prag, sistemul activează LED-ul care va pâlpâi, emite un semnal acustic prin buzzer și afișează mesajul "EXIT" pe LCD. Dacă nivelul de CO este în limite normale, sistemul va afișa un mesaj corespunzător pe LCD și nu va activa LED-ul sau buzzer-ul.

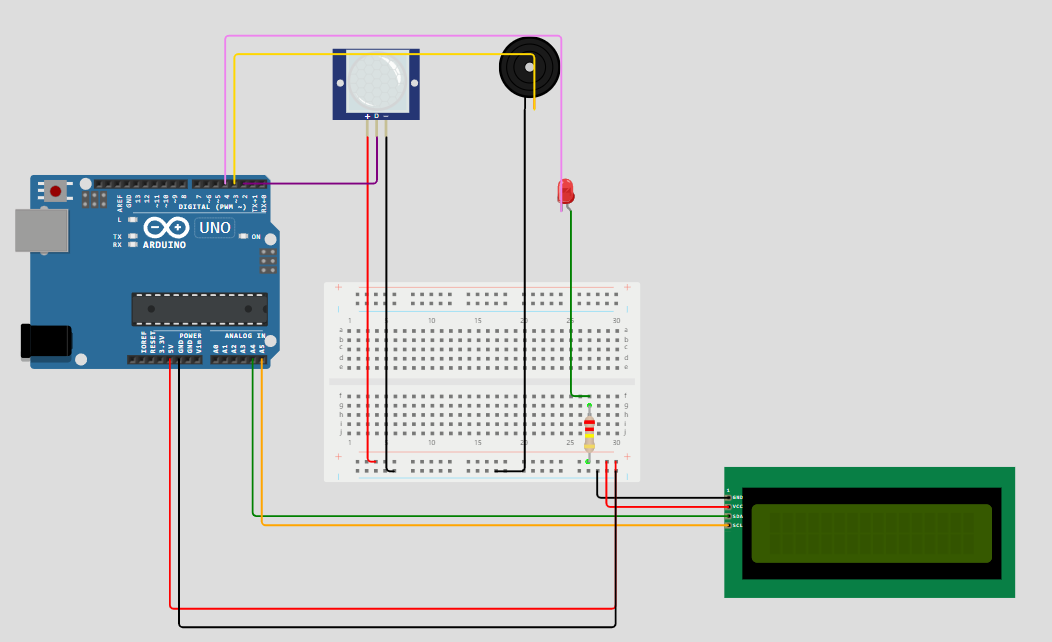
**Hardware**

Componentele hardware utilizate în proiect sunt:

* **Senzor MQ-7**: Senzorul de gaz MQ-7 este folosit pentru a detecta monoxidul de carbon. Semnalul său analog este citit de Arduino.
* **LED**: Folosit pentru a semnaliza vizual pericolul de monoxid de carbon.
* **Buzzer**: Avertizează sonor prezența monoxidului de carbon.
* **LCD 16x2**: Afișează mesaje pentru utilizator.
* **Arduino Uno**: Placa de dezvoltare care controlează logica și interacțiunea dintre componentele hardware.

**Schema circuitului**

* **MQ-7** se conectează la un pin analogic (D2) al Arduino.
* **LED** se conectează la un pin digital (D4).
* **Buzzer** se conectează la un pin digital (D3).
* **LCD** se conectează prin I2C la pinii A4 (SDA) și A5 (SCL).



**Software**

Programul citește periodic nivelul de CO de la senzor, compară cu pragul și acționează corespunzător.

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

const int MQ7\_PIN = 2; // Senzorul MQ-7

const int LED\_PIN = 4; // LED-ul

const int BUZZER\_PIN = 3; // Buzzer-ul

LiquidCrystal\_I2C lcd(0x27, 16, 2);

const int CO\_THRESHOLD = 100;

unsigned long previousMillis = 0;

const long blinkInterval = 500;

int ledState = LOW;

void setup() {

pinMode(LED\_PIN, OUTPUT);

pinMode(BUZZER\_PIN, OUTPUT);

Serial.begin(9600);

}

void loop() {

int coLevel = analogRead(MQ7\_PIN);

Serial.print("Nivel CO: ");

Serial.println(coLevel);

if (coLevel > CO\_THRESHOLD) {

lcd.clear();

lcd.setCursor(6, 0);

lcd.print("EXIT");

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= blinkInterval) {

previousMillis = currentMillis;

ledState = !ledState;

digitalWrite(LED\_PIN, ledState);

if (ledState == HIGH) {

tone(BUZZER\_PIN, 1000);

} else {

noTone(BUZZER\_PIN);

}

}

} else {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("nivel normal");

lcd.setCursor(0, 1);

lcd.print("de oxigen");

digitalWrite(LED\_PIN, LOW);

noTone(BUZZER\_PIN);

}

delay(100);

}

1. **Definirea pinii pentru componente**:  
const int MQ7\_PIN = 2; const int LED\_PIN = 4; const int BUZZER\_PIN = 3;  
Acești pini sunt folosiți pentru conectarea senzorului MQ-7 (pentru citirea nivelului CO), LED-ului (pentru semnalizare vizuală) și buzzer-ului (pentru semnalizare sonoră).

2. **Inițializarea ecranului LCD**:  
LiquidCrystal\_I2C lcd(0x27, 16, 2);  
Se configurează un ecran LCD compatibil cu protocolul I2C, având adresa **0x27** și dimensiunea de 16x2 caractere.

3. **Setarea pragului CO și controlul temporizării LED-ului**:  
const int CO\_THRESHOLD = 100; unsigned long previousMillis = 0; const long blinkInterval = 500; int ledState = LOW;  
Se definește un prag de 100 pentru nivelul de CO. De asemenea, variabilele pentru temporizare (previousMillis și blinkInterval) controlează clipirea LED-ului.

4. **Configurarea componentelor în setup**:  
void setup() { pinMode(LED\_PIN, OUTPUT); pinMode(BUZZER\_PIN, OUTPUT); Serial.begin(9600); }  
Setăm pinii LED și buzzer ca ieșiri și pornim comunicația serială pentru afișarea valorilor senzorului în consolă.

5. **Citirea senzorului și afișarea valorilor**:  
int coLevel = analogRead(MQ7\_PIN); Serial.print("Nivel CO: "); Serial.println(coLevel);  
Se citește valoarea de la senzorul MQ-7 (semnal analogic) și se afișează în consolă pentru diagnosticare.

6. **Verificarea depășirii pragului CO**:  
if (coLevel > CO\_THRESHOLD) { lcd.clear(); lcd.setCursor(6, 0); lcd.print("EXIT"); }  
Dacă nivelul de CO depășește pragul setat, ecranul LCD afișează mesajul „EXIT”.

7. **Alertă cu LED și buzzer**:  
unsigned long currentMillis = millis(); if (currentMillis - previousMillis >= blinkInterval) { previousMillis = currentMillis; ledState = !ledState; digitalWrite(LED\_PIN, ledState); if (ledState == HIGH) { tone(BUZZER\_PIN, 1000); } else { noTone(BUZZER\_PIN); } }  
LED-ul începe să clipească (în funcție de intervalul definit) și buzzer-ul emite un sunet la fiecare aprindere a LED-ului.

8. **Comportament normal pentru nivel redus de CO**:  
else { lcd.clear(); lcd.setCursor(0, 0); lcd.print("nivel normal"); lcd.setCursor(0, 1); lcd.print("de oxigen"); digitalWrite(LED\_PIN, LOW); noTone(BUZZER\_PIN); }  
Dacă nivelul de CO este sub prag, pe LCD este afișat „nivel normal de oxigen”, iar LED-ul și buzzer-ul sunt dezactivate.

9. **Întârziere între citiri**:  
delay(100);  
Se introduce o întârziere de 100 ms pentru a preveni citirea continuă și suprasolicitarea componentelor.

**4. Testare și validare**

**Probleme întâmpinate**

* **Calibrarea senzorului**: Senzorul MQ-7 poate necesita un timp de încălzire pentru a ajunge la o valoare stabilă, ceea ce a dus la erori temporare în citirea nivelurilor de CO la început.
* **Senzorul sensibilitate**: Senzorul poate fi influențat de alți factori de mediu, ceea ce poate determina o detectare falsă a CO.

**Teste de calibrare**

Am efectuat teste la diferite valori de monoxid de carbon și am comparat răspunsul senzorului la diferite nivele. De asemenea, am ajustat poziționarea senzorului pentru a maximiza sensibilitatea la monoxidul de carbon.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**5. Concluzie**

Scopul inițial a fost îndeplinit: am realizat un detector de monoxid de carbon eficient și ușor de implementat. Soluția oferă atât semnalizări vizuale, cât și sonore pentru a alerta utilizatorul asupra unui pericol iminent. În plus, soluția este ușor de adaptat pentru a fi folosită în diferite medii sau pentru a adăuga noi caracteristici (de exemplu, conectivitate la internet pentru alertă de la distanță).

Implementarea poate fi îmbunătățită prin utilizarea unor senzori mai sensibili sau prin adăugarea unui modul de comunicare wireless pentru a trimite alerte pe telefon sau email în caz de urgență.