

Smart Fire Alarm - Dragos-Sorin Barbu

Introducere

Proiectul vizeaza implementarea unui sistem integrat de alertare și prevenire a incendiilor, care imbina functionalitati de detectare a gazelor, fumului si flacarilor. Scopul principal al acestui sistem este de a asigura un nivel crescut de siguranta si protectie pentru persoane si proprietati in cazul unui incendiu, precum si de a facilita prevenirea acestora.

Ideea de la care am pornit a fost sa combin mai multe tipuri de senzori intr-un singur sistem, astfel incat sa pot monitoriza si evalua riscul de incendiu intr-un mod cat mai eficient. Am considerat ca aceasta abordare poate contribui la reducerea consecintelor negative ale incendiilor si la salvarea de vieti.

Pe langa afisarea unui mesaj corespunzator pe ecran si activarea unui difuzor, sistemul trimite notificari si pe telefonul utilizatorului prin intermediul unui modul Bluetooth, ceea ce ii permite acestuia sa raspunda rapid si eficient in situatii critice.

Descriere generală

Proiectul va include trei senzori principali:

Senzorul de temperatura: Acest senzor se va ocupa de monitorizarea temperaturii si va emite un semnal luminos atunci cand valoarea stabilita este depasita, indicand astfel o posibila problema.

Senzorul de gaz: Acest senzor are rolul de a verifica nivelul de gaz din spatiul respectiv si de a alerta utilizatorul printr-un semnal luminos daca nivelul depaseste limita prestabilita, semnalizand astfel posibilitatea unei scurgeri de gaz.

Senzorul de flacara: Acest senzor este responsabil pentru declansarea unei alarme sonore de tip buzzer in functie de valoarea detectata. Astfel, daca senzorul identifica prezenta flacarilor, alarma se va activa pentru a atrage atentia asupra unui posibil incendiu.

La declansarea oricarui senzor, printr-un modul Bluetooth se vor transmite catre telefon mesaje corespunzatoare de avertizare.

Schema bloc



Hardware Design

Lista de piese:

- placa Arduino Uno
- modul bluetooth
- modul senzor gaz
- modul senzor temperatura
- modul senzor flacara
- ecran LCD
- buzzer
- led-uri
- potentiometru
- baterii
- rezistente
- fire



Software Design

Pentru început, am inclus bibliotecile necesare pentru a controla componentele hardware. Biblioteca `dht11.h` permite comunicarea cu senzorul DHT11, care măsoară temperatura și umiditatea, în timp ce `LiquidCrystal_I2C.h` gestionează afișajul LCD. De asemenea, utilizez `SoftwareSerial.h` pentru a permite comunicarea serială prin alți pini decât cei standard, necesară pentru modulul Bluetooth.

În primele linii, am definit pinii cu care urmează să lucrez și variabile de stare în funcție de senzorii declanșați.

Apoi, mi-am definit o serie de funcții ajutătoare: `lcd_clean`, `temp_hum_sensor`, `air_quality_sensor`, `fire_sensor`, `print_sensor_readings`.

- `lcd_clean`: Această funcție este folosită pentru a șterge textul afișat pe ecranul LCD. Ea scrie spații goale pe fiecare caracter de pe ecran și apoi așteaptă 50 de milisecunde pentru a se asigura că operația de ștergere este finalizată.
- `temp_hum_sensor`: Această funcție citește valorile de temperatură și umiditate de la senzorul DHT11. Dacă valorile se află în afara intervalului sigur (umiditate sub 20% sau peste 80%, temperatura sub 15°C sau peste 35°C), funcția activează ledul și buzzerul, afișează un avertisment pe ecranul LCD și trimite avertismentul și prin Bluetooth. Dacă valorile se află în intervalul sigur, funcția oprește ledul și buzzerul.
- `air_quality_sensor`: Această funcție citește valoarea de la senzorul de calitate a aerului MQ135. În funcție de această valoare, dacă este peste 500, se consideră un nivel periculos de înalt, dacă este peste 280, se consideră un nivel ușor periculos, iar dacă este sub 280, se consideră un nivel sigur. În cazul unui nivel periculos sau ușor periculos, funcția activează ledul și buzzerul, afișează un avertisment pe ecranul LCD și trimite avertismentul și prin Bluetooth.
- `fire_sensor`: Această funcție citește valoarea de la senzorul de flacără. Dacă valoarea este peste 500,

se consideră că nu există incendiu. Dacă valoarea este sub 500, se consideră prezența unui incendiu și funcția activează ledul și buzzerul, afișează un avertisment pe ecranul LCD și trimite avertismentul și prin Bluetooth.

- `print_sensor_readings`: Această funcție este responsabilă pentru afișarea valorilor senzorilor pe ecranul LCD, pe consola serială și prin Bluetooth. Afișarea pe LCD este organizată astfel: pe prima linie se afișează temperatura și umiditatea, iar pe a doua linie se afișează calitatea aerului. Ea se apelează doar dacă nicio alertă nu este activă.

În setup, am configurat pinii pentru intrare sau ieșire, am inițializat ecranul LCD și seriala. În loop, senzorii sunt citiți în mod constant prin intermediul funcțiilor și dacă citirile lor sunt în afara valorilor sigure, se activează avertismentele. Dacă toate citirile sunt în valorile sigure, se afișează citirile pe LCD, pe consola serială și prin Bluetooth. Aici folosesc valorile întoarse de funcțiile senzorilor, pentru a putea afișa starea mediului atunci când nicio valoare nu se află în zona critică:

```
if (airQualityAlert == 0 && fireAlert == 0 && tempHumAlert == 0) {  
    print_sensor_readings();  
}
```

În cod se află mai multe delay-uri. Acestea sunt necesare pentru a asigura timp necesar scrierii atât pe LCD, cât și pe telefon prin Bluetooth. Am experimentat cu mai multe valori pentru a găsi punctul în care informațiile sunt afișate corespunzător, iar delay-urile au valori cât mai mici.

Rezultate Obținute

[Link demo](#)

Concluzii

Proiectul a fost un exercițiu plăcut în dezvoltarea unui sistem utilizabil, implicând integrarea și coordonarea mai multor componente hardware și software. A demonstrat eficiența și flexibilitatea platformei Arduino în domeniul monitorizării mediului, unde capacitatea de a citi date de la o varietate de senzori și de a reacționa corespunzător a fost esențială. În plus, proiectul a evidențiat importanța feedbackului continuu, atât prin intermediul afișajului LCD, cât și prin Bluetooth, în menținerea unei monitorizări eficiente și a conștientizării situațiilor de risc.

Dificultăți întâmpinate:

- În primul rând, calibrarea senzorilor pentru a obține citiri precise a reprezentat o provocare, în special pentru senzorul de calitate a aerului MQ135, a cărui citire poate fi influențată de o varietate de factori. De asemenea, pentru a calibra acest senzor, a fost necesară rularea continuă, timp de 24 de ore pentru a asigura atingerea parametrilor.
- Modulul Bluetooth a prezentat mai multe probleme de conectivitate, fiind necesar să utilizez modul AT pentru a-l reseta și pentru a alege alt baud rate. De asemenea, a trebuit să găsesc un telefon cu Android pentru a-l putea utiliza.
- gestionarea simultană a mai multor sarcini, cum ar fi citirea de la senzori, afișarea datelor și

gestionarea alertelor, a necesitat un design atent și optimizat al codului.

Sugestii de îmbunătățire:

- se poate adăuga o funcționalitate de înregistrare a datelor, astfel încât să se poată monitoriza evoluția alertelor pe o perioadă mai lungă de timp, fiind păstrate și data și ora producerii acestora (un fel de black box).
- se poate adăuga o pompă de apă pentru a se acoperii și partea de stingere a incendiului.

Download

[barbu_dragos_sorin_335cb.zip](#)

Jurnal

07/05/2023 → realizare pagina ocw, introducere, schema bloc

19/05/2023 → Completat partea hardware

25/05/2023 → Completat partea software

27/05/2023 → Finalizat Documentație

Bibliografie/Resurse

<https://reference.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>

<https://ocw.cs.pub.ro/courses/pm>

<https://projecthub.arduino.cc/arcaegecengiz/using-dht11-12f621>

https://www.youtube.com/watch?v=ykgt4v7_4y4

[Export to PDF](#)

From:

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

Permanent link:

<https://ocw.cs.pub.ro/courses/pm/prj2023/avaduva/smartfirealarm>



Last update: **2023/05/30 00:29**