Fire alarm

Introducere

Prezentarea pe scurt a proiectului vostru:

- ce face

Proiectul va realiza diminuarea problemelor dintr-o încăpere atunci când este prezent un incendiu. Acesta va semnala pe un display faptul că un incendiu are loc și pe un card SD va salva toate log-urile relevante pentru evenimentele anterioare(timestamp, temperatură, etc.). De asemenea, printr-un difuzor va avea si un efect de alarmă.

- care este scopul lui

Scopul lui este de a preveni incendiile într-o încăpere.

- care a fost ideea de la care aţi pornit

Am pornit de la ideea de alarmă de incendiu simplu. Elementele de log-uri și de afișare pe un display au venit ulterior pentru a ține evidența evenimentelor.

Descriere generală

La acțiunea unui stimul extern pentru senzorul de temperatură se va transmite un semnal către microcontroller. Apoi, software-ul de pe microcontroller va putea transmite către celelalte componente semnale.

Hardware Design

Piesele hardware necesare:

- Microcontroller ATMega328PB
  - Mediază interacțiunea între celelalte piese și software
- Display LCD 1602 I2C
  - Afișează un mesaj în caz de incendiu
- Buzzer

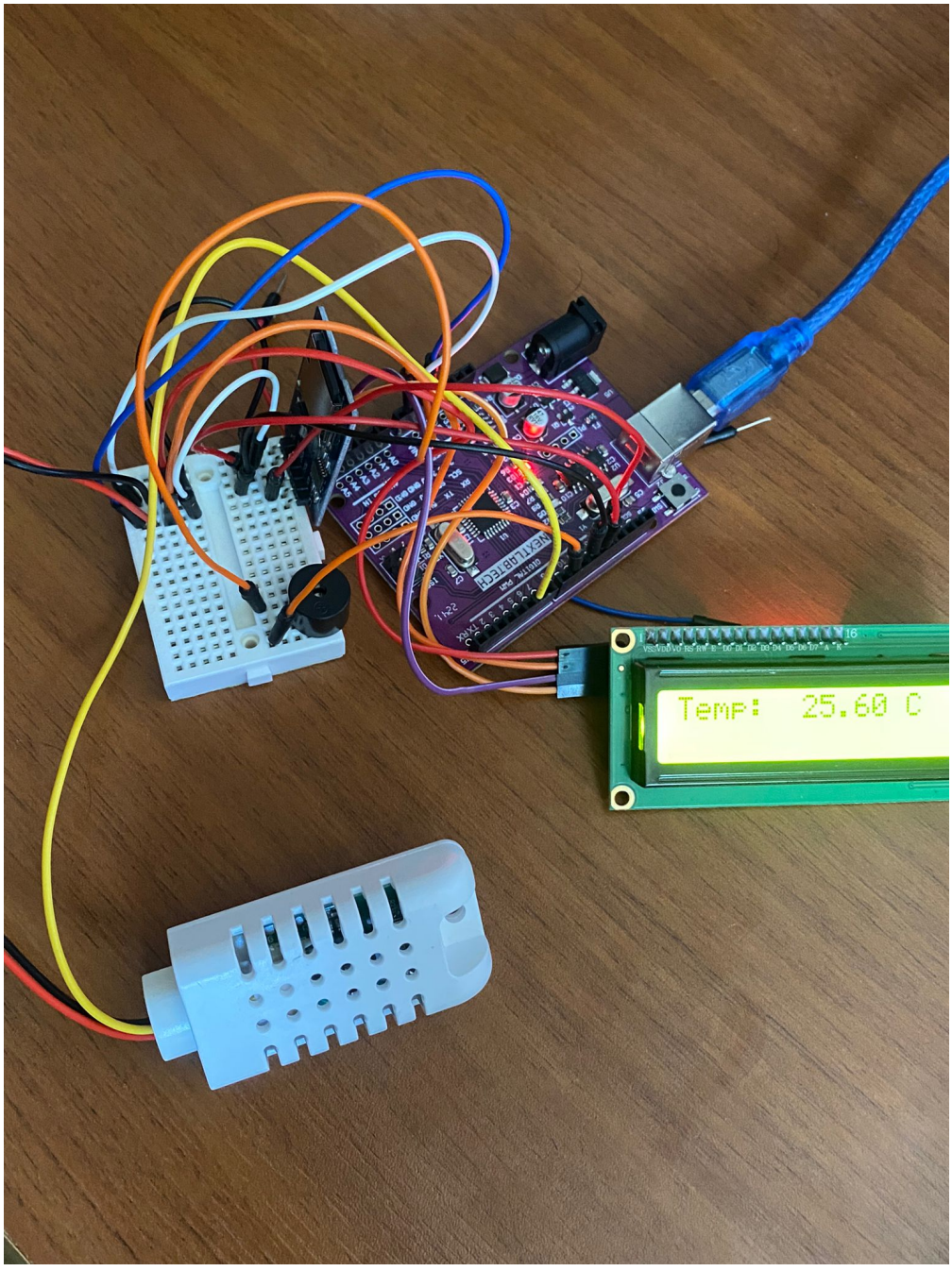
  - Pe post de alarmă
- Modul card microSD
  - Salvează log-urile pe un card SD
- Senzor temperatura DHT22

Software Design

Pentru programarea plăcii am folosit Arduino IDE, fără biblioteci specifice Arduino. Singurele biblioteci incluse sunt `<avr/io.h>` și `<util/delay.h>`, în rest s-au folosit funcții implementate manual (fiind un proiect orientat pe software). La baza design-ului software stau laboratoarele de I2C, SPI și datasheet-ul pentru senzorul DHT22, practic am încercat să integrez cât mai mult cod similar cu cel din laborator (plăcile fiind aproape identice). Implementarea ocupa de 2 ori mai putin spatiu decat cea cu biblioteci arduino. Particularități de implementare:

- **Implementarea comunicației I2C**:
  1. Am utilizat digitalWrite() pentru a comunica cu registrii SDA si SCL
  2. Inițializarea magistralei I2C se face setând pinii SDA și SCL ca ieșire
  3. Trimiterea unui bit pe magistrala I2C implică scrierea bitului pe SDA si setarea lui SCL pe high
  4. Citirea unui bit de pe magistrala I2C implica setarea lui SDA pe input si citirea lui
- **Implementarea comunicației SPI**:
  1. Comunicarea SPI are o implementare foarte similara cu cea din laborator: la inceput setez SCK, MISO, SS pe OUTPUT si MISO pe input
  2. Atunci cand are loc comunicatia efectiva pun data in registrul SPDR si il trimit
- **Citirea datelor de la senzorul DHT22**:
  1. Citirea datelor de la senzorul DHT22 se face folosind manipulări directe ale pinului de date(D6). Funcția `readDHT` inițializează comunicarea trimițând un semnal de start și apoi citește datele folosind cicluri de asteptare si delay-uri
  2. Pasii functiei sunt luati din datasheet-ul DHT22
- **Controlul buzzerului**:
  1. Controlul buzzerului se face folosind funcțiile `tone()` și `noTone()` pentru a emite sunete de avertizare atunci când temperatura depășește un anumit prag.
- **Afișarea datelor pe ecranul LCD**:
  1. Am implementat funcții pentru inițializare, curățare, setare cursor și scriere text pe LCD utilizând protocolul I2C. Inițializarea LCD-ului implică trimiterea unor comenzi de configurare prin magistrala I2C.
- **Utilizarea comunicației seriale**:
  1. Pentru debugging și monitorizarea datelor în timpul execuției programului, am folosit funcțiile `Serial.print()` și `Serial.println()`.

Rezultate Obţinute



Concluzii

Avand in vedere implementarea low-level si functionalitatea robusta, pot afirma ca proiectul isi atinge toate obiectivele propuse. Natura constructiei si codul compact poate reduce costurile semnificativ, intrucat placuta nu este folosita la capacitate maxima.

Download

proiect_ivv.zip

Bibliografie/Resurse

https://ocw.cs.pub.ro/courses/pm/lab/lab5-2023-2024
https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf
https://ww1.microchip.com/downloads/en/DeviceDoc/40001906A.pdf
https://www.nxp.com/docs/en/user-guide/UM10204.pdf
https://github.com/johnrickman/LiquidCrystal_I2C/blob/master/LiquidCrystal_I2C.cpp

Export to PDF