

Sistem de inregistrare a datelor - Cohea Ioan-Tudor 333CA

Introducere

Ce face proiectul?

Proiectul este destinat inregistrarii anumitor date din atmosfera inconjuratoare prin intermediul mai multor senzori specializati. Acesti senzori pot inregistra date precum umiditatea, temperatura si gradul de fum care este prezent in atmosfera, acestea urmand sa fie afisate pe un LCD impreuna cu un grafic al anumitor inregistrari. Mai mult, datele inregistrate in ultima ora se vor memora prin intermediul unui modul MicroSD si al unui card de 32GB care va fi conectat la sistem.

Care este scopul proiectului?

Acesta are scopul de a:

- oferi inregistrari la o acuratete ridicata a anumitor factori inconjuratori;
- analiza acestora prin intermediul graficelor care urmeaza sa fie generate;
- usura detectarea anumitor factori periculosi pentru sanatatea umana precum fumul sau gazul;

Care a fost ideea de pornire a proiectului?

Ideea proiectului a pornit din dorinta de a crea un dispozitiv care poate inregistra simultan mai multe date despre atmosfera inconjuratoare. Astfel, acest sistem cu mai multi senzori mi s-a parut o idee interesanta, pentru a putea monitoriza oricand, cu usurinta, numerosi indici legati de atmosfera din casa. Mai mult, scurgerile de gaze si incendiile reprezinta o cauza importanta a numeroase incidente, mai ales in locuintele private, asa ca acest sistem se va dovedi foarte util.

De ce credeți că este util pentru alții și pentru voi?

Avand in vedere ca in aceasta era tehnologica strangerea de date este cruciala pentru buna functionare a oricarui sistem, necesitatea unui inregistrator de date este facila si poate fi folosit de

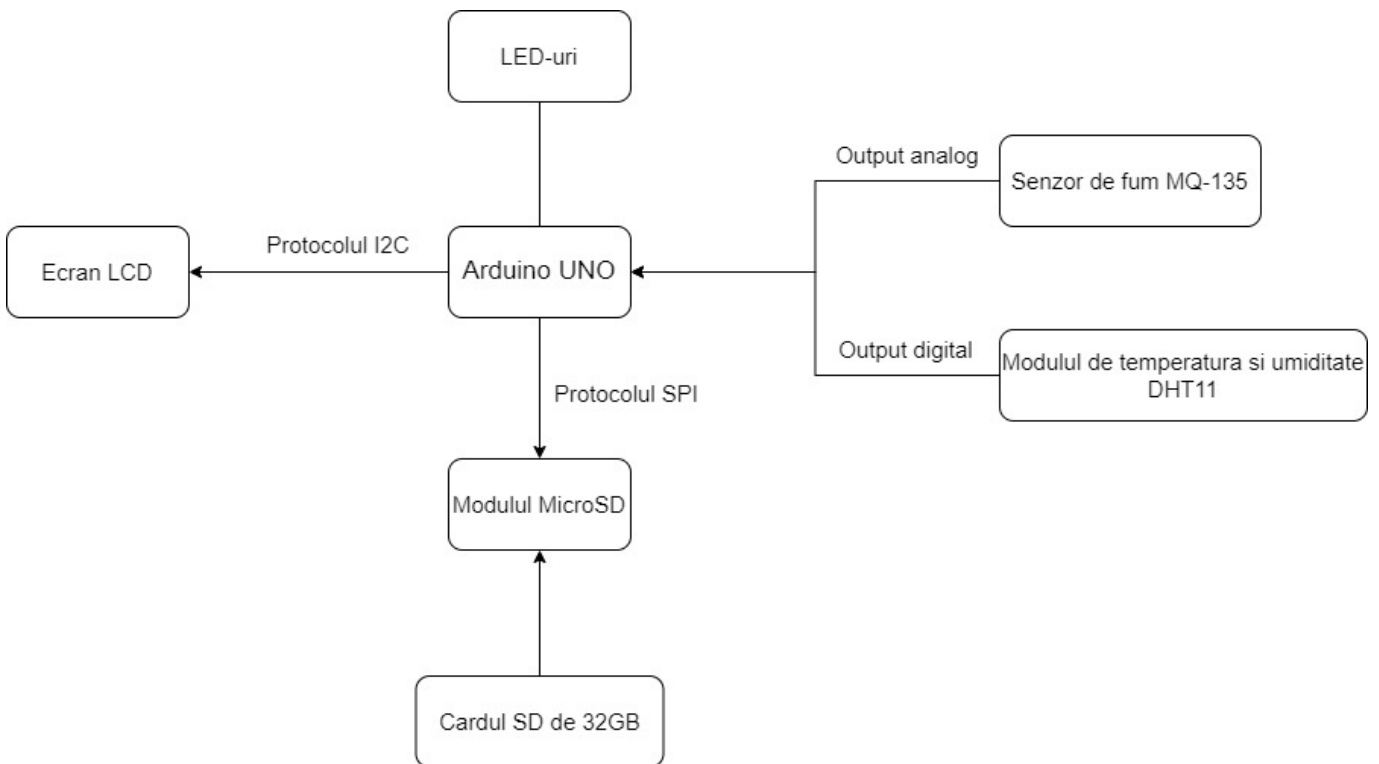
catre oricine in orice conditii. Mai mult, consider ca acest sistem este util pentru prevenirea incendiilor si pentru atentionarea la aparitia temperaturilor severe sau aparitia unui nivel de gaze periculos.

Descriere generală

Descriere proiect

Proiectul consta in atasarea si conectarea mai multor senzori care inregistreaza date din atmosfera inconjuratoare. Senzorul de temperatura va trimite date direct, prin intermediul protocolului de I2C, catre ecran, urmand sa se genereze si un grafic al inregistrarilor. Aceste inregistrari sunt retinute cu ajutorul unui modul MicroSD si a unui card SD de 32GB. Senzorii de umiditate si de fum vor fi de asemenea conectati la LCD pentru a trimite date catre acesta. De asemenea, sistemul va contine si un buzzer care se va activa in momentul in care senzorul de fum va detecta o valoare peste pragul maxim admis. Pentru acuratete, datele livrate de catre detectorul de fum prin intermediul unui pin analogic vor fi citite cu ajutorul unui ADC care este controlat prin setarea anumitor registri.

Descriere block:



Hardware Design

Produsele folosite in implementarea hardware a proiectului sunt:

- Arduino UNO
- Modul MicroSD cu card SD de 32GB
- Ecran LCD
- Buzzer
- Senzor de temperatura si umiditate DHT11
- Senzor de fum MQ-135
- Led-uri si fire care conecteaza intre ele modulele
- Rezistente pentru protectia let-urilor

Software Design

Pentru implementarea proiectului se folosesc tool-uri si protocoale

- legatura cu ecranul LCD este realizata prin intermediul protocolului I2C;
- legatura cu modulul MicroSD se realizeaza prin intermediul protocolului SPI;
- senzorii de temperatura si de umiditate si fum vor scoate la output semnale diferite, lucru care va trebuie rezolvat prin conctarea la porturile corecte;
- LCD-ul va fi initializat si se pot citi de pe porturi valorile oferite de catre senzori;

Biblioteci pentru gestiunea senzorilor

- LiquidCrystal_I2C.h pentru display-ul LCD
- dht.h si Adafruit_Sensor.h pentru senzorul de temperatura/umiditate DHT11
- SD.h si SPI.h pentru modulul de card microSD

Setup function

In aceasta functie se seteaza pinii corespunzatori pe in sau pe out, se initializeaza LCD-ul si baud rate-ul. De asemenea, aici se verifica si daca operatiile facute pe cardul SD sunt corecte.

```
// initializarea pinilor, output pentru led-uri si buzzer si input pentru senzori  
pinMode(greenLed, OUTPUT);  
pinMode(redLed, OUTPUT);  
pinMode(buzzer, OUTPUT);
```

```
pinMode(smokeA0, INPUT);
pinMode(data_pin, OUTPUT);

// setare lcd
lcd.init();
lcd.clear();
lcd.backlight();

if (SD.begin(sd_cs)) {
    Serial.println("SD card initialization successful.");
} else {
    Serial.println("SD card initialization failed.");
    return;
}

File data_file = SD.open("data.txt", FILE_WRITE);
```

De asemenea, aici se seteaza si registrele folosite pentru a initializa ADC-ul cu care se citeste de pe pinul analogic valoarea data de catre senzorul de fum

```
// setup-ul pentru citirea pinului smokeA0 folosind registri si ADC
ADMUX = 0;

// setez canalul 1 pentru multiplexor
ADMUX |= (0 << MUX0);
ADMUX |= (1 << REFS0);
ADCSRA = 0;

// prescaler la 128
ADCSRA |= (7 << ADPS0);

// enable ADC
ADCSRA |= (1 << ADEN);
```

Initializari

```
// initializez lcd-ul
LiquidCrystal_I2C lcd(0x27, 16, 2);

// initializez modulul de DHT11
dht DHT;
```

Variabile relevante

```
// Valoarea de threshold care poate fi modificata
int sensorThres = 94;
```

```
// fisierul deschis pe cardul SD de unde iau valorile
File data_file;

// pinul digital dupa care citesc valoarea senzorului de temperatura
int data_pin = 8;
```

Implementare

LCD-ul este initializat cu ajutorul laboratorului 6 de I2C care este un protocol de comunicatie. Cele mai importante linii ale magistralei I2C sunt SDA(linia de date) si SCL(linia pentru clock). Ambii pini de SDA si SCL sunt conectati la pinii speciali de pe placuta, intrucat aceasta este deja configurata pentru acest protocol.

```
// setare initiala LCD
lcd.init();
lcd.clear();
lcd.backlight();
lcd.begin(16, 2);
lcd.setCursor(0, 0);
```

Legatura cu cardul SD este realizata prin intermediul protocolului de SPI care are 4 semnale:

- MOSI - transmite date de la Master la Slave
- MISO - transmite date de la Slave la Master
- SCLK - sincronizarea dispozitivelor
- CS/SS - selectarea dispozitivului Slave de catre master in cazul in care sunt mai multe

```
data_file = SD.open("data.txt", FILE_WRITE);

// scrierea se face la fiecare iteratie si se opreste in momentul in care se
// termina tot programul
data_file.print("AS:");
data_file.print(analogSensor);
data_file.println(",");

// inchid fisierul
data_file.close();
```

Senzorul de temperatura si umiditate DHT11 trebuie sa fie initializat, iar dupa aceea, se citesc datele de pe pinul digital asociat acestuia folosindu-se 2 functii specifice bibliotecii dht.h. Ulterior, acestea sunt printate pe lcd si pe baza inregistrarilor se vor genera graficele.

```
// initializez senzorul
dht DHT;

// citesc datele date de catre senzor de pe pinul setat
int read_data = DHT.read11(data_pin);
```

```
// inregistrez temperatura si umiditate
float temp = DHT.temperature;
float umid = DHT.humidity;

lcd.print(static_cast<float>(temp), 0);
lcd.print(static_cast<float>(umid), 1);
```

Toate aceste calcule sunt realizate in interiorul unui if care numara intervale de doua secunde fara a folosi o functie de delay generala pe toata functia de loop. Daca am folosi un delay, atunci si sunetul buzzer-ului ar trebuie sa fie intrerupt o data la doua secunde, ceea ce nu este de dorit. Aceasta solutie este aleasa deoarece conform datasheet-ului, senzorul DHT11 are nevoie de 2 secunde intre inregistrari pentru a se calibra intern.

```
unsigned long currMill = millis();

if (currMill - prevMill >= interval) {
    prevMill = currMill;
}
```

Senzorul de fum MQ-135 este folosit pentru a inregistra nivelul de fum din atmosfera. Acesta foloseste Adc pentru a citi de pe pinul analogic. De asemenea, la acest senzor este conectat un buzzer care este implementat pentru a se activa in momentul in care se depaseste o valoare de threshold care poate fi modificata din cod

```
int sensorThres;

while (!(ADCSRA & (1 << ADIF)));
uint16_t analogSensor = ADC;

// daca media calculata cu inregistrarile senzorului depaseste
// pargul admis, atunci se activeaza buzzerul, altfel nu
if (static_cast<float>(sum) / count > sensorThres) {
    // aprind becul
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    // buzzer
    tone(buzzer, 1000, 200);
} else {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    noTone(buzzer);
}
```

Laboratoarele folosite sunt: I2C, ADC, SPI, UART.

Rezultate Obținute

Semnalul generat de catre senzorul de fum in momentul in care este supus unei surse de gaz:



Semnalul generat de catre senzorul de temperatura si umiditate in cazul in care acesta inregistreaza:

- temperatura in cazul in care se inregistreaza o crestere de un grad



- umiditatea in cazul in care se inregistreaza o crestere aproape exponentiala a umiditatii



De asemenea, este interesant de mentionat si cum arata output-ul generat in fisierul de pe cardul SD:

```
AS: 78 ,  
AS: 78 ,  
AS: 78 ,  
AS: 78 ,  
AS: 78 ,  
AS: 78 ,
```

Concluzii

Acest proiect a fost foarte interesant pentru ca nu ne lovim in fiecare zi de problema implementarii unui sistem cu tot cu partea de hardware. Lucrul cu care am ramas din acest proiect este ca este foarte important sa iti cunosti piesele si tool-urile cu care lucrezi si sa nu te arunci in a implementa pana nu citesti cu atentie documnetatia despre piesa respectiva. De asemenea, proiectul a adus si o satisfactie in momentul in care puteai sa zici ca inregistrarile pe care le primeai erau facut aproape integral de catre tine.

Un alt lucru cu care am ramas dupa acest proiect este ca nu este obligatoriu ca soft-ul sa fie bun pentru ca si hardware-ul sa fie bun. Am invatat ca sunt doua entitati separate care trebuie tratate si separat si impreuna pentru buna functionare a proiectului.

Download

[pm.zip](#)

Jurnal

12.04.2023 - Alegere tema proiect

24.04.2023 - Achizitie piese

16.05.2023 - Finalizare etapa hardware

25.05.2023- Finalizare etapa software

28.05.2023 - finalizare pagină documentație

Bibliografie/Resurse

Resurse Hardware

- <https://cleste.ro/promo/17-electric-sales>
- <https://projecthub.arduino.cc/arcaegecengiz/using-dht11-12f621>

Resurse software

- <https://microcontrollerslab.com/interfacing-mq-135-gas-sensor-arduino/>
- <https://projecthub.arduino.cc/arcaegecengiz/using-dht11-12f621>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/adarmaz/sistem-inregistrare-date>



Last update: **2023/05/28 20:03**