

Detector de scurgeri de gaze si inundatii

- Nume: Veliscu Robert-Valentin
- Grupa: 341C4

Introducere

Acest proiect detecteaza scurgerile de gaze si inundatiile si trimite datele catre un server prin protocolul HTTP pentru a fi vizualizate de la distanta. Am ales sa fac acest proiect deoarece eu plec din Bucuresti acasa aproximativ o data la 2 saptamani si de fiecare data ma gandesc peste weekend daca am lasat aragazul sau o baterie pornite. Cred ca ar fi util si pentru alti oameni uituci. :)

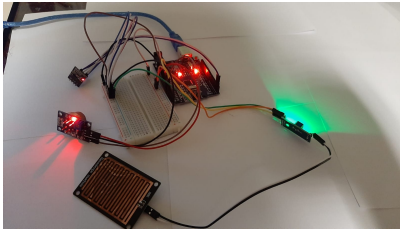
Descriere generală

Modulul Wifi ESP8266 se alimenteaza la iesirea de 3.3V a arduino-ului. Acesta are de asemenea si pin-urile Reset si CH_EN conectate la 3.3V. Pin-ul TX a modulului este conectat la pin-ul RX al arduino-ului, iar pin-ul RX a modulului este conectat la pin-ul TX al arduino-ului pentru a asigura comunicarea. Senzorul de gaze MQ-5 se alimenteaza cu 5V din arduino si are iesirea analog conectata la pin-ul A0 al arduino-ului. Senzorul de ploaie se alimenteaza si el cu 5V din arduino si are iesirea analog conectata la pin-ul A1 al arduino-ului. Atat modulul ESP8266 cat si senzorii au cate un pin GND si acestia sunt legati la GND-ul arduino-ului.

Hardware Design

Lista de piese:

- Arduino UNO
- Modul ESP8266 Serial Port
- Senzor detectie metan, gaze naturale MQ-5
- Modul detectare picaturi



Software Design

Cod Arduino

Setup

Algoritmul incepe prin configurarea initiala a microcontrolerului AVR si a modulului ESP8266 in functia `setup()`. In aceasta etapa, se realizeaza urmatoarele actiuni:

- Se seteaza tipul pinilor pentru conectarea senzorilor de gaz si apa (input).
- Se activeaza comunicarea seriala (USART) intre microcontroler si modulul ESP8266.
- Se stabileste conexiunea la reseaua Wi-Fi folosind functia `connectToWiFi()`.
- Se asteapta o scurta perioada de timp pentru ca modulul ESP8266 sa se conecteze la retea.

Loop

Dupa configurarea initiala, algoritmul intra in bucla principala `loop()`, unde se realizeaza urmatoarele actiuni:

- Se citesc valorile analogice de la senzorii de gaz si apa utilizand functia `readAnalogValue()`.
- Se analizeaza valorile citite pentru determinarea starii senzorului de gaz si a clasei de apa.
- Pentru senzorul de gaz, daca valoarea citita este mai mare de 250, se considera ca exista gaz si se seteaza variabila gaz la 1. Altfel, se seteaza gaz la 0.
- Pentru senzorul de apa, se realizeaza o serie de comparatii pentru a determina clasa apei in functie de valoarea citita (sunt 4 clase). Aceasta se atribuie variabilei `current_class`.
- Se verifica daca s-au produs modificari in starea senzorului de gaz sau in clasa apei fata de masuratorile anterioare.
- Daca s-au produs modificari, se seteaza variabila `ok` la 1 pentru a semnala ca trebuie trimise noi date catre server.
- Daca s-a detectat o schimbare si trebuie trimise noi date, se apeleaza functia `sendPostRequest()` pentru a realiza o cerere POST catre serverul extern, transmitand valorile de gaz si apa.
- Se asteapta o perioada de 5 secunde intre doua cereri consecutive.

Funcții auxiliare

În algoritmul principal am definit câteva funcții auxiliare care asigură funcționalitatea sistemului:

- `connectToWiFi()`: Această funcție realizează conectarea la rețeaua Wi-Fi utilizând SSID-ul și parola specificate. Comunicarea se face prin comenzi AT trimise modulului ESP8266.
- `sendPostRequest(int gaz, int apa)`: Funcția construiește o cerere POST formatată conform protocolului HTTP și o trimite către serverul extern. Cererea conține valorile de gaz și apă și este transmisă prin comunicarea serială cu modulul ESP8266.
- `readAnalogValue(int pin)`: Funcția citește valoarea analogică de la pinul specificat și o convertește într-o valoare digitală folosind modulul ADC al microcontrolerului AVR.

Cod Server

Server

- Se importă modulele necesare, cum ar fi Express, Body Parser și WebSocket. Se configurează adresa IP și portul serverului.
- Se configurează middleware-ul Express pentru a permite analiza cererilor și răspunsurilor HTTP. De asemenea, se setează motorul de vizualizare EJS pentru a genera pagini HTML.
- Odată ce se primește o cerere GET pentru ruta de bază "/", se randează template-ul EJS "index.ejs", care afișează valorile curente ale cheilor "gaz" și "apa".
- Atunci când se primește o cerere POST pentru ruta de bază "/", se actualizează valorile cheilor "gaz" și "apa" cu noile valori primite în corpul cererii POST. Apoi, se trimite o actualizare prin WebSocket către toți clienții conectați.
- Atunci când un client se conectează prin WebSocket, acesta este adăugat la lista de clienți conectați. Se trimite clientului valorile curente ale cheilor "gaz" și "apa". În momentul în care un client se deconectează, acesta este eliminat din lista de clienți.

Client

- Se stabilește o conexiune WebSocket către adresa IP și portul specificat.
- Când se primește un mesaj WebSocket, acesta este parsat într-un obiect JSON pentru a obține valorile gazului și apei.
- Utilizând un switch, se verifică valorile primite și se afișează mesaje corespunzătoare pentru fiecare valoare, însoțite de schimbări ale imaginii de fundal și a culorii textului.
- Elementele HTML cu id-urile "gaz" și "apa" sunt actualizate pentru a afișa valorile corespunzătoare.

Rezultate Obținute

În urma realizării proiectului am reușit să am un modul de detecție care trimite datele citite de la

senzori la un server remote si astfel ma pot bucura de weekenduri linistit.

Download

[Cod Sursa](#)

Bibliografie/Resurse

- [Senzor Gaz](#)
- [Senzor picaturi](#)
- [ESP8266](#)

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/drtranca/robert.veliscu>



Last update: **2023/05/30 08:33**