Smart Home System

Introducere

Proiectul consta in realizarea unui dispozitiv care raspunde nevoii de automatizare a unor procese in ceea ce priveste locuinta proprie. Una dintre functionalitati va fi monitorizarea temperaturii si a umiditatii dintr-o camera, cu ajutorul unui senzor specific, iar cand se depaseste o limita setata de utilizator, va declansa un ventilator ce va mentine sau scadea temperatura/umiditatea. De asemnea, toate datele vor fi trimise si stocate catre un server. O alta functionalitate va fi monitorizarea accesului intr-o anumita zona de actiune, unde va fi montat un senzor de miscare care cand va detecta miscare, va comunica cu un senzor magnetic de usa, care monitorizeaza daca o usa este deschisa sau inchisa, iar in cazul in care se detecteaza miscare si usa este deschisa, se va trimite un semnal de avertizare pe server.

Scopul dispozitivului este monitorizarea conditiilor aerului din casa, care este foarte util in timpul verii si de asemenea prevenirea tentativelor de spargere pe perioada in care nu este nimeni acasa.

Am ales acest proiect, deoarece mi se pare un start bun in ceea ce priveste automatizarea propriei case si deoarece se pliaza nevoilor mele.

Descriere generală

block_diagram.png

Hardware Design

Lista de piese:

- ESP WROOM 32
- Modul PIR - senzor de prezenta, miscare
- Senzor Magnetic MC-38 usa
- Modul ventilator 5V L9110
- Senzor de temperatură și umiditate - DHT11
- Rezistente
- Fire de legatura
- Buton

Schema electrica:

hardware.png

Software Design

Mediu de dezvoltare : Arduino IDE

* Script-ul care citeste datele de la senzori, proceseaza aceste date si le comunica server-ului, dar si comanda ventilatorul.

Biblioteci externe:

- DHT.h → pentru procesarea datelor oferite de senzorul DHT11
- WiFi.h → pentru a conecta ESP32-ul la internet
- ThingSpeak.h → pentru a putea conecta Arduino la platforma ThingSpeak, unde vom incarca si vizualiza datele

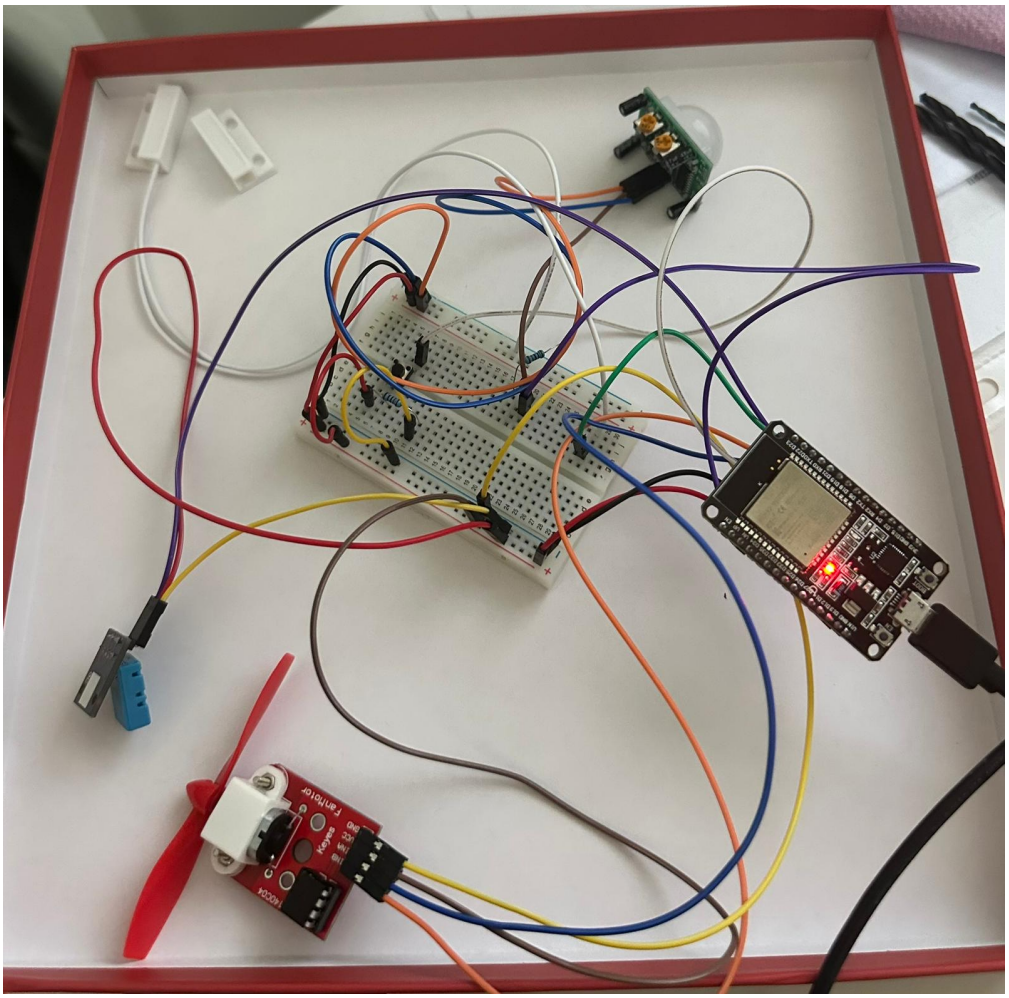
Codul sursă se află în secțiunea ***Download***.

In functia **Setup()**, am realizat conexiunea la internet prin functia **begin** a bibliotecii “Wifi”, am pornit serverul **ThingSpeak**, catre care va urma sa trimitem datele de la senzori si pe langa acestea, am configurat toti pinii conectati la placuta, fie ei fiind de **OUTPUT**, fie de **INPUT**, pentru cei din urma folosind si rezistente de PULLUP.

In functia **Loop()**, citesc datele de la fiecare senzor, respectiv de temperatura, de umiditate, de detectare a miscarii si de detectare a usii daca este deschisa sau inchisa. Pentru senzorii de temperatura si umiditate, verific daca valorile primite sunt valabile (!= nan), deoarece aceasta bucata de cod ma poate ajuta la debug atunci cand unul din senzori se arde. Pe baza valorilor de temperatura si umiditate, calculez o noua valoare numita **indice de caldura**, ce reprezinta temperatura resimtita si pe baza caruia voi controla ventilatorul. Acesta va porni cand se depaseste o valoare critica setata de utilizator. (avand in vedere conditiile actuale si precizia senzorului DHT11, am setat valoarea critica ca fiind 30 grade Celsius). Pe langa acestea, ventilatorul poate fi comandat si dintr-un buton, conectat la placuta, dar numai in momentul cand nu este depasita temperatura critica. Tot in functia de loop, am realizat transmiterea datelor catre **ThingSpeak**. In ceea ce priveste temperatura si umiditatea, se face cate o cerere de tip POST, o data la 20 de secunde, iar referitor la senzorii ce asigura securitate usii de intrare, se trimit date mai des, o data la 10 secunde. In cazul in care se detecteaza o miscare, se trimit date despre starea usii, daca este deschisa sau inchisa, iar daca aceasta este deschisa, se aprinde un buton rosu de avertizare. De asemenea, am creat si un buton de avertizare pentru temperaturi ridicate, de data aceasta temepratura critica fiind 28 grade Celsius. (**ATENTIE** - butonul se activeaza pe baza temperaturii de pe server, iar ventilatorul pe baza indicelui de caldura sau temperatura resimtita din camera). Trimiterea catre server a fost destul de usoara, deoarece m-am folosit de functiile **setField** si **writeFields** din biblioteca **ThingSpeak**.

Rezultate Obţinute

hardware.jpg



temperature_chart.png

humidity_chart.png

temperature_light.png

security_charts.png

security_alert.png

Concluzii

Totul e bine cand se termina cu bine. O expereinta foarte interesanta, fiind prima interactiune pe cont propriu cu domeniul hardware. Am intampinat dificultati in proictarea hardware, nereusind sa realizez si o macheta pe care sa aplic proiectul, acest fapt din cauza firelor multe, dar si scurte. De asemenea,

din punct de vedere software, am avut probleme si cu conectarea placutei la WiFi din cauza diferentelor de bandwidth, facandu-le incompatibile, ceea ce mi-a luat aproape o zi din implementarea efectiva. In rest, totul a mers bine si acest proiect m-a facut sa descopar o parte frumoasa a hardware-ului si cu siguranta voi mai realiza astfel de proiecte, mai performante, mai eficiente pentru uzul propriu.

Download

smart_home_system_filip_fabian.zip

Jurnal

- 24.04 → alegere tema proiect
- 08.05 → milestone documentatie
- 10.05 → comanda piese
- 15.05 → milestone hardware
- 22.05 → milstone software
- 26.05 → finalizare proiect
- 29.05 → finalizare pagina wiki

Bibliografie/Resurse

https://randomnerdtutorials.com/

- cel mai utilizat

https://www.instructables.com/How-to-Use-a-Magnetic-Door-Switch-Sensor-With-Ardu/

https://lastminuteengineers.com/pir-sensor-arduino-tutorial/

https://www.onetransistor.eu/2018/01/compute-heat-index-arduino-dht.html

https://www.arduinolibraries.info/libraries/thing-speak

https://github.com/arduino-libraries/WiFi

Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2023/razvans/smart_home_system

Last update: **2023/05/30 11:54**