Intelligent cooling system

* Nume si Prenume: Stamatin Teodor

* Grupa: 335CA

Introducere

Proiectul este bazat pe platforma ESP8266 si transforma un simplu ventilator intr-un sistem inteligent, capabil sa raspunda automat la schimbarile de temperatura din interiorul unui spatiu. Este foarte asemanator cu un cooler de laptop sau cu un aer conditionat modern, si are doua moduri de functionare:

1. AUTO: porneste ventilatorul in mod automat in functie de temperatura de interior. Functioneaza in trepte, depinzand de temperatura, pana cand aceasta ajunge inapoi sub pragul definit.
2. MANUAL: permite manipularea vitezei ventilatorului de catre utilizator cu ajutorul unui potentiometru.

Aceste moduri de functionate comuta intre ele prin apasarea unui buton. Pe un ecran OLED se va putea vedea temperatura si viteza de rulare a ventilatorului, iar pe ecranul pc-ului se vor afisa mesaje si informatii precum modul de functionare si temperatura.

Scopul proiectului este cresterea confortului termic dintr-un spatiu (mare - camera sau mic - carcasa unui laptop) si reducerea consumului de energie, prin modul inteligent de actionare al ventilatorului. Eu voi implementa in proiect ideea de cooler de laptop.

Descriere generală

Pentru a porni dispozitivul, utilizatorul trebuie sa conecteze cooler-ul la PC. Cooler-ul porneste in modul AUTO, deci se va activa atunci cand diferenta de temperatura este depasita, iar cand temperatura scade sub limita inferioara, ventilatorul se opreste. Prin actionarea asupra potentiometrului de catre utilizator, acesta poate ajusta viteza ventilatorului in modul MANUAL. Pentru a fi mai interactiv am ales sa folosesc un ecran LCD care sa afiseze in timp real viteza pe care o are ventilatorul sub forma unui grafic. Schema bloc:

Hardware Design

Mai jos este lista cu principalele componente folosite in acest proiect, impreuna cu link-urile lor:

Componenta	Descriere	Link
ESP8266	Microcontroller	Link
DHT11	Senzor temperatura	Link
ST7735	Ecran OLED	Link
PWM Fan	Ventilator	Link
Buton	Mod selectare	Link
Potentiometru 10kΩ	Control manual viteza	Link
9V Battery	Baterie	

Schema electrica

Schema este facuta in AutoDesk Fusion 360. Intrucat nu am gasit decat schema electrica a potentiometrului si a butonului, am ales sa imi fac propria librarie in cam am facut design-ul celorlalte piese:

Conexiune intre piese

- Afisajul OLED (ST7735): este conectat la pinii D3 (RS), D4 (RST), D5 (SCL), D7 (SDA) si D8 (CS) ai placutei ESP8266. Alimentarea este facuta prin VCC (5V) si GND. Afisajul are rolul de a vizualiza temperatura masurata, starea curenta a modului (automat sau manual), precum si viteza ventilatorului sau temperatura.

- Potentiometrul: este conectat la pinul analogic A0 al ESP8266, iar celelalte doua capete la GND si 3.3V. Acesta este utilizat pentru a seta manual viteza ventilatorului atunci cand sistemul se afla in modul manual.

- Butonul: are rolul de a comuta intre cele doua moduri de functionare (automat/manual). Este conectat la pinul digital D2 si nu am conectat o rezistenta de pull-up deoarece am activat rezistenta interna de pull-up a ESP-ului prin software:

```
const int buttonPin = D2;
pinMode(buttonPin, INPUT_PULLUP);
```

- Senzorul de temperatura DHT11: este conectat la pinul D1 al placutei ESP8266 pentru semnalul de date, iar alimentarea este realizata prin VCC (3.3V) si GND. Acesta masoara temperatura din spatiul in care se afla si o transmite catre ESP pentru procesare in modul automat.

- Controller-ul PWM: este alimentat separat de la o baterie de 9V deoarece puterea data de ESP era insuficienta; ventilatorul nu pornea la un input mic. Acest controller permite ajustarea vitezei ventilatorului in functie de semnalul primit de la potentiometru.

- Ventilatorul de 12V: este conectat la controllerul PWM si alimentat direct de la acesta. Controlul vitezei se face prin semnalul PWM venit de la ESP.

- PC-ul: este conectat prin USB pentru comunicatie seriala. Acesta permite monitorizarea comportamentului sistemului in timp real.

Rezultat final

Software Design

Mediul de dezvoltare folosit este Arduino IDE versiunea 1.8.19 cu baud rate 115200.

Biblioteci utilizate

Adafruit_GFX.h si Adafruit_ST7735.h – pentru controlul grafic al afisajului TFT SPI.

DHT.h – biblioteca pentru senzorul DHT (Digital Humidity and Temperature), care este folosit pentru a citi temperatura si umiditatea din mediul inconjurator. Biblioteca contine functii specializate pentru initializarea senzorului si pentru citirea datelor de temperatura si umiditate.

SPI.h – pentru comunicatia cu afisajul pe ecran, biblioteca simplifica utilizarea ecranului, oferind functii pentru initializare, scriere de text, controlul iluminarii de fundal si pozitionarea cursorului.

Variabile globale pentru pinii ecranului + initalizare:

```
#define TFT_CS     D8
#define TFT_RST    D4
#define TFT_DC     D3
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
```

Macro folosit pentru folosirea potentiometrului de la jumatate spre maxim deoarece in zona inferioara puterea semnalului PWM era prea mica

```
#define ADJ_ADC(val) ( ((val) / 2) + 512 )
```

Initializare senzor temperatura

```
#define DHTPIN D1
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

Structura generala

In functia setup() se initializeaza serialul pentru debugging, senzorul DHT11, afisajul TFT, pin-ul pentru controlul PWM si butonul, setat cu INPUT_PULLUP. Se afiseaza un mesaj de start atat pe serial, cat si pe ecran.

```
pinMode(buttonPin, INPUT_PULLUP);
dht.begin();
tft.initR(INITR_BLACKTAB);
pinMode(fanPwmPin, OUTPUT);
```

Programul foloseste o variabila globala “stare” care determina modul de functionare:

stare = 1 → Mod automat (control pe baza temperaturii)

stare = 2 → Mod manual (control prin potentiometru)

Comutarea intre stari se face la apasarea butonului, folosind o verificare de tip “edge detection” (LOW urmat de HIGH).

Mod automat (stare 1)

In acest mod, senzorul DHT11 citeste temperatura ambientala. In functie de temperatura, se seteaza automat o valoare PWM corespunzatoare si o treapta de viteza pentru ventilator:

```
float temperatura = dht.readTemperature();
int pwmValue = 0;
    int treapta = 0;
    if (temperatura < 24) {
      pwmValue = 0;
      treapta = 0;
    } else if (temperatura < 26) {
      pwmValue = 90;
      treapta = 1;
    } else if (temperatura < 28) {
      pwmValue = 145;
      treapta = 2;
    } else if (temperatura < 30) {
      pwmValue = 200;
      treapta = 3;
    } else {
      pwmValue = 255;
```

```
      treapta = 4;
    }
```

Temperatura (°C)	PWM	Treapta
< 24	0	0
24–26	90	1
26–28	145	2
28–30	200	3
≥ 30	255	4

Valorile sunt transmise prin analogWrite() catre pinul PWM al ventilatorului. Pe afisaj se arata temperatura si treapta de viteza.

```
analogWrite(fanPwmPin, pwmValue);
```

Mod manual (stare 2)

In acest mod, controlam manual viteza ventilatorului printr-un potentiometru conectat la pinul analog A0. Valoarea citita este scalata intre 0 si 255 folosind:

```
int ajustat = ((analogRead(potPin) / 2) + 512);
int pwm = map(ajustat, 512, 1023, 0, 255);
```

Se afiseaza o bara de progres si valoarea bruta citita, pentru a oferi un feedback vizual clar.

```
tft.fillRect(10, 40, 100, 10, ST77XX_BLACK);
tft.fillRect(10, 60, 100, 10, ST77XX_BLACK);
int latimeBar = map(ajustat, 512, 1023, 0, 100);
tft.fillRect(10, 40, latimeBar, 10, ST77XX_GREEN);
```

Afisaj TFT

Interfata grafica pe ecranul TFT reda:

- Mesaje informative despre stare si temperatura
- Treapta de viteza (in mod automat)
- Bara de progres (in mod manual)
- Ecranul este sters local (doar pe zonele de interes) inainte de afisare, pentru a evita “ghostingul”.

Debugging + Informatii

Prin Serial Monitor se ofera informatii utile pentru depanare:

- Temperatura curenta
- Valoarea ADC de la potentiometru
- Nivelul PWM trimis ventilatorului
- Schimbarea modului de operare

Design Decisions

Alegerea modului de control dual (automat/manual)

Implementarea a celor doua moduri de functionare arata flexibilitatea proiectului: consumatorul poate lasa sistemul sa regleze automat viteza ventilatorului in functie de temperatura sau poate regla manual prin potentiometru. Comutarea modurilor cu un buton simplifica interactiunea si este gestionata printr-un mecanism de detectie a frontului crescator (de la LOW la HIGH), evitand fluctuatiile bruste.

Controlul PWM, fara feedback de la ventilator

Desi initial era prevazut si un fir pentru citirea turatiei ventilatorului (RPM), l-am scos deoarece arata fluctuatii prea mari. PWM-ul este controlat direct prin analogWrite.

Scalarea valorilor analogice

Potentiometrul are o iesire intre 16 si 1024, dar in mod manual se doreste control fin doar pentru jumatatea superioara (512–1024), care este mapata la intervalul PWM (0–255). Acest lucru asigura ca consumatorul are un control mai precis in intervalul util, evitand ca viteza sa fie prea mica si ventilatorul sa nu porneasca.

Afisajul grafic

Afisajul TFT este utilizat pentru a reda vizual starea sistemului. Am optat pentru un display grafic, nu alfanumeric, pentru a permite o interfata mai dinamica – inclusiv progress bar in mod manual, mesaje de stare si trepte numerice. Folosirea bibliotecilor Adafruit_GFX si Adafruit_ST7735 a permis o integrare rapida cu functii de desen eficiente si control precis al pozitionarii textului si elementelor grafice.

Senzorul DHT11

Senzorul DHT11 a fost ales pentru simplitatea sa si integrarea usoara cu biblioteci existente. Desi nu este cel mai precis sau rapid senzor, este suficient pentru a detecta praguri de temperatura pentru controlul ventilatorului.

Rezultate Obţinute

https://www.youtube.com/watch?v=xnyV8hBOCRM

Concluzii

Proiectul functioneaza asa cum a fost planificat. Comutarea intre modurile automat si manual se face corect, ventilatorul raspunde bine la comenzi, iar informatiile de pe ecran si din Serial Monitor confirma ca totul merge cum trebuie.

Download

Arhiva cu implementarea si schema electrica se poate descarca de aici.

Jurnal

- Milestone 1: 06.05.2025 - Alegerea temei proiectului + documentatie
- Milestone 2: 13.05.2025 - Terminarea documentatiei + implementarea hardware
- Milestone 3: 20.05.2025 - Terminarea partii harware si a software-ului

Bibliografie/Resurse

- https://www.optimusdigital.ro/ro/
- https://www.emag.ro/kit-wireless-super-starter-cu-esp8266-programabil-cu-arduino-ide-x0012yt7g9/pd/DN82W2MBM/?ref=history-shopping_348224276_77141_1
- https://www.instructables.com/Programming-ESP8266-ESP-12E-NodeMCU-Using-Arduino-/
- https://www.snapeda.com/parts/DHT11/UNIVERSAL-SOLDER%20Electronics%20Ltd/view-part/?comp

any=UPB+Bucharest&ref=search&t=DHT11
- https://drawio-app.com/
- https://www.youtube.com/watch?v=9-AZF6udg-Q

Export to PDF