

Auto Fan

Nume: Mitran Andrei-Gabriel

Grupă: 333CA

Introducere

Proiectul se axează pe crearea unui sistem de ventilare inteligent, care se activează automat atunci când temperatura din încăpere atinge un anumit prag. Scopul său principal este de a asigura răcirea utilizatorului în timpul activităților desfășurate la birou. În plus, sistemul se oprește automat în funcție de luminozitatea ambientală, contribuind astfel la reducerea consumului de energie, mai ales în condiții de întuneric.

Ideea proiectului a pornit de la nevoia de a oferi un confort termic optim și o eficiență energetică crescută în medii de lucru statice, cum este biroul. Scopul a fost dezvoltarea unui dispozitiv care să ofere o răcire automată și eficientă, fără a necesita intervenția manuală constantă a utilizatorului.

Proiectul se remarcă prin utilitatea sa pentru alții, deoarece abordează o necesitate frecventă în medii de lucru interioare, oferind o soluție convenabilă și eficientă pentru gestionarea temperaturii ambientale.

Descriere generală

Se începe cu detectarea temperaturii ambientale folosind un senzor dedicat. Când temperatura atinge un anumit prag, microcontrolerul Arduino primește datele de la senzor și activează ventilatorul pentru a începe răcirea. În paralel, un alt senzor măsoară luminozitatea camerei, iar în funcție de aceasta, microcontrolerul se poate opri automat pentru a economisi energie, mai ales în condiții de întuneric.

Pe lângă gestionarea ventilatorului, microcontrolerul afișează temperatura curentă pe un ecran LCD și indică starea camerei (întuneric sau lumină) folosind LED-uri colorate. De asemenea, sistemul poate fi controlat și de la distanță, utilizând o telecomandă.

Ventilatorul continuă să funcționeze până când temperatura scade sub prag sau după trecerea unui anumit interval de timp pentru a economisi energie. Astfel, acest sistem oferă o soluție automată și eficientă pentru gestionarea temperaturii și a consumului de energie în mediile de lucru interioare.

Concepte folosite:

- **GPIO** - Majoritatea componentelor funcționează prin GPIO.
- **Timere și Întreruperi** - Sunt folosite de telecomandă IR la comunicarea cu senzorul IR și implicit cu Arduino Uno.
- **PWM** - Pini digitali pot genera semnale PWM. Astfel, se controlează motorul.



Hardware Design

Listă de componente:

- 1 x Arduino Uno (Plusivo Uno R3) - Placa principală, ce alimentează toate componentele (cu excepția motorului)
- 1 x placă de cablaj
- 1 x termorezistor - Senzor de temperatură
- 1 x modul LCD1602 - Afișează diferite informații utile (temperatura curentă, temperatura de threshold, modul de funcționare etc.)
- 1 x potențiomtru (10K) - Reglează contrastul LCD-ului
- 1 x L293D IC - Controlează motorul
- 1 x lamă de ventilator
- 1 x motor 3-6V - Folosit ca ventilator împreună cu lama
- 1 x modul de alimentare
- 1 x baterie 9V - Alimentează ventilatorul împreună cu modulul și adaptorul
- 1 x adaptor 9V 1A
- 1 x fotodiodă - Senzor de lumină
- 2 x LEDs (1 x roșu, 1 x albastru) - Indică modul de funcționare
- 1 x modul receptor pentru telecomandă infraroșu
- 1 x telecomandă infraroșu - Facilitează controlul temperaturii de threshold și a modului de funcționare
- rezistori (220 Ω , 1K etc.) și fire

Descriere pini:

- LCD
 - VSS - GND
 - VDD - 5V
 - VO - potențiomtru (contrast)
 - RS - D7 (controlul memoriei)
 - R/W - GND
 - E - D8 (enable)
 - D0-D3 (LCD) - neconectați
 - D4-D7 (LCD) - D9-D12 (pe Arduino)

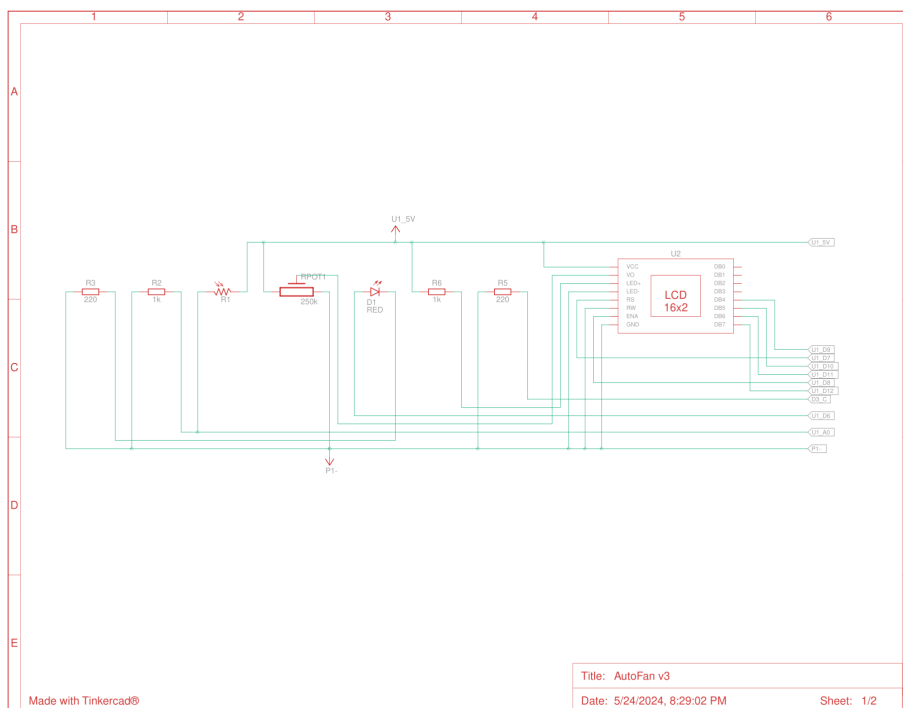
- A - 5V
- K - GND
- LED-uri (la D2 și D6)
- L293D IC
 - VS - 5V
 - INPUT2 - D3
 - OUTPUT2 - - motor
 - GND(x2) - GND
 - OUTPUT1 - + motor
 - INPUT1 - D4
 - ENABLE - D5 (PWM)
- senzor IR
 - VCC - 5V
 - GND - GND
 - DATA - D13
- fotorezistor (la A0)
- termorezistor (la A1)

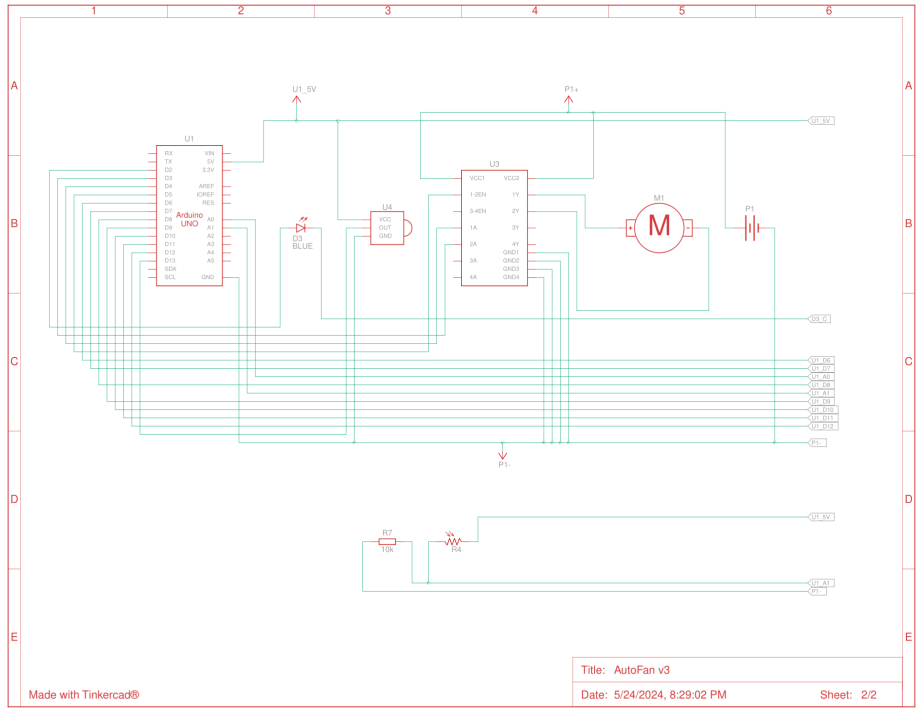
Schemă Tinkercad:



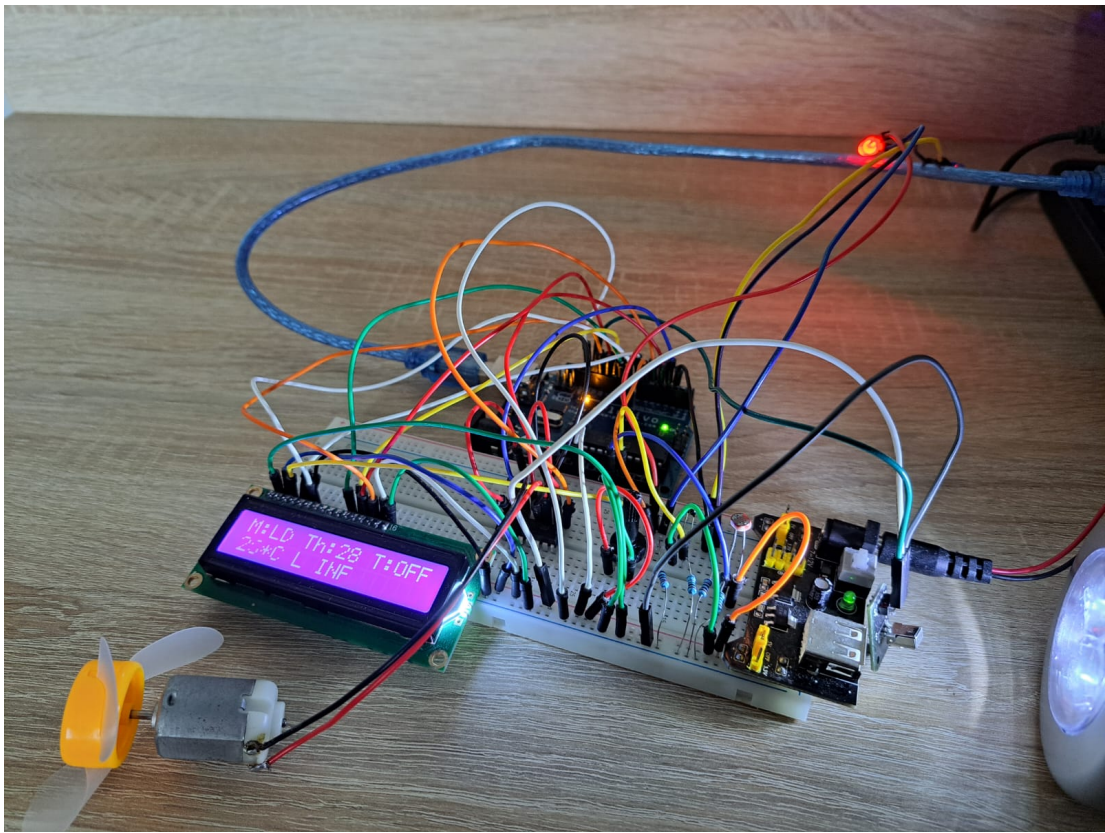
* Nu am găsit termorezistor în TinkerCad, de aceea sunt două fotorezistoare prezente în schemă (precum și în cea electrică).

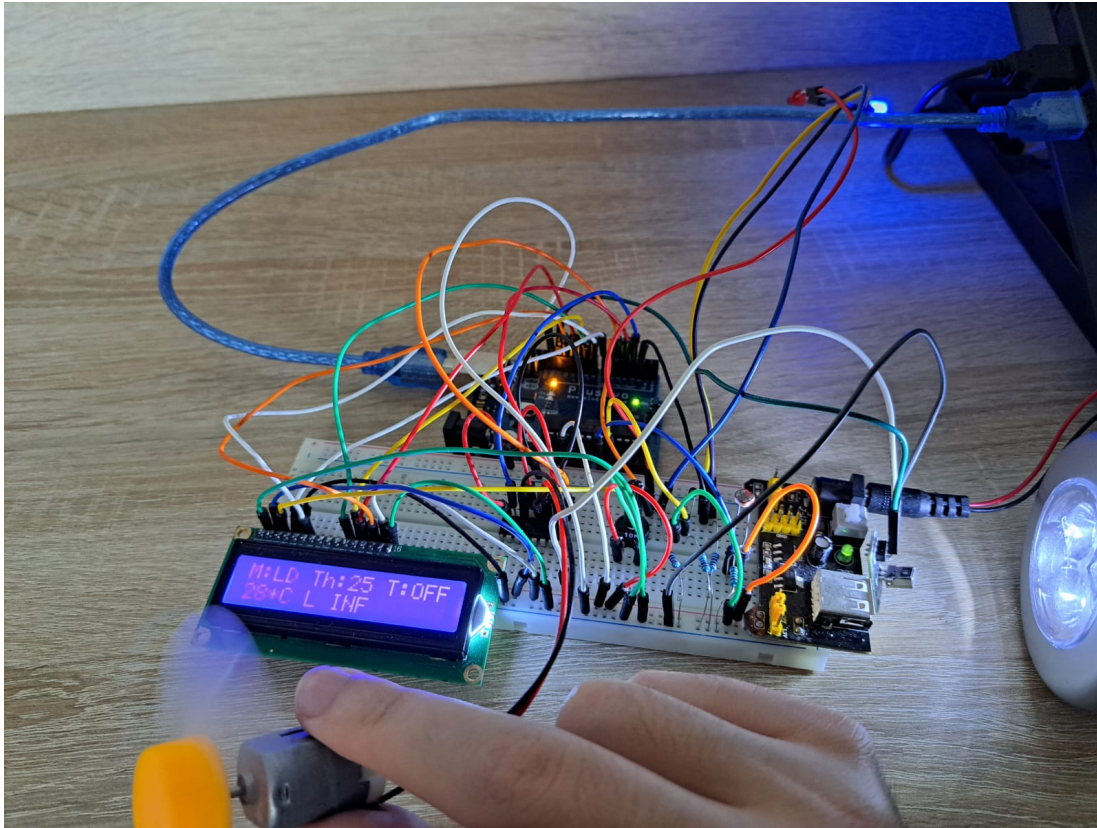
Schemă electrică:





Real life:





Demo: [AutoFan v3](#)

Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare: ArduinoIDE
- librării: LiquidCrystal (pentru LCD), IRremote (pentru telecomanda IR)

GitHub: [AutoFan](#)

Variabile Globale:

1. tempPin, receiver, ENABLE, DIRA, DIRB, RED, BLUE: Definirea pinilor pentru senzorul de temperatură, receptorul IR și componente hardware.
2. tempThreshold: Threshold-ul de temperatură pentru pornirea sau oprirea ventilatorului.
3. timeThreshold: Threshold-ul de timp pentru funcția de oprire automată a sistemului.
4. mode: Indicator pentru modul de funcționare al sistemului (dependent sau independent de lumină).
5. systemRunning: Indicator pentru starea de funcționare a sistemului.
6. autoOffEnabled: Indicator pentru funcția de oprire automată a sistemului.
7. startTime: Variabilă pentru stocarea timpului de început pentru countdown.

Funcții:

translateIR(): - Se execută în funcție de codul IR primit și realizează următoarele acțiuni:

1. Schimbă threshold-ul de temperatură cu butoanele CH- și CH+.
2. Schimbă modul de funcționare a sistemului cu butonul CH.
3. Activează/dezactivează funcția de oprire automată a sistemului cu butonul PAUSE.
4. Schimbă threshold-ul de timp cu butoanele VOL- și VOL+.

colour(int colour):

- Schimbă culoarea LED-urilor (roșu sau albastru) în funcție de parametrul "colour".

stopSystem():

- Oprește sistemul, dezactivând ventilatorul și afișând "OFF" pe LCD.

loop():

1. Verifică dacă sistemul rulează și dacă a trecut threshold-ul de timp pentru oprire automată, în caz afirmativ, oprește sistemul.
2. Afișează "OFF" și oprește sistemul în cazul în care "systemRunning" este "false".
3. Primește și interpretează comenzile de la telecomandă în funcție de codul IR recepționat.
4. Calculează și afișează temperatura, statusul luminii (L/D), countdown-ul până la oprire (sau "INF" pentru oprire automată dezactivată) și alte informații pe LCD.
5. Pornește sau oprește ventilatorul în funcție de temperatură și modul de funcționare ales.

Calibrarea senzorilor

În codul dat, este folosit un senzor de temperatură și unul de lumină care returnează o valoare analogică în funcție de temperatura detectată, respectiv de intensitatea luminii ambientale. Această valoare analogică este citită folosind funcția analogRead() a Arduino.

Calibrarea senzorului de temperatură constă în transformarea valorii analogice citite într-o valoare de temperatură corespunzătoare. Pentru acest lucru, sunt folosite unele formule și constante matematice, bazate pe caracteristicile senzorului și pe legile fizicii. În acest caz, este folosit un senzor de temperatură NTC (Negative Temperature Coefficient), care își schimbă rezistența electrică în funcție de temperatură. De asemenea, deoarece fotorezistoarele variază rezistența în funcție de intensitatea luminii, o valoare mai mică a citirii analogice indică o lumină mai puternică, în timp ce o valoare mai mare indică o lumină mai slabă.

Formulele (specifice senzorilor) sunt utilizate pentru a asigura o corectitudine cât mai mare în măsurare folosind senzorii conectați la Arduino.

Rezultate Obținute

1. **Eficiența Energetică Îmbunătățită:** Implementarea funcției de oprire automată a sistemului a dus la o reducere a consumului de energie.
2. **Flexibilitate și Adaptabilitate:** Ajustarea pragurilor de temperatură și timp în timp real a crescut gradul de personalizare al sistemului.

Concluzii

Proiectul a reprezentat o experiență extrem de valoroasă pentru mine, oferindu-mi oportunitatea de a aplica cunoștințele tehnice într-un mod practic și creativ. Integrarea diverselor tehnologii și soluții m-a ajutat să dobândesc o înțelegere profundă și expertiză în domeniul automatizării și controlului dispozitivelor. Lucrul de unul singur m-a încurajat să îmi dezvolt abilitățile de rezolvare a problemelor și de gestionare a resurselor. Această experiență m-a motivat să continui să explorez și să învăț în domeniul ingineriei și al tehnologiei, contribuind la dezvoltarea mea profesională și personală pe termen lung.

Download

Codul/Firmware, include bibliotecile utilizate și README (.zip):

[autofan-main.zip](#)

Jurnal

- 04/05/2024 - Creare pagină wiki.
- 17/05/2024 - Versiune funcțională fără telecomandă adăugată.
- 24/05/2024 - Versiune funcțională cu telecomandă adăugată. Senzorul de temperatură DHT11 a fost înlocuit de un termorezistor.
- 25/05/2024 - Update pagină OCW cu documentație și demo.



Bibliografie/Resurse

Ghid de la [Plusivo](#).

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/andrei.mitran3011>



Last update: **2024/05/25 15:17**