

Temperature-controlled Fan

Autor: Vîță Cristian, 332CC

Introducere

Proiectul va ajuta la determinarea temperaturii si umiditatii ambientale afisand pe un ecran valorile respective. Scopul proiectului este de a vedea cum se pot manipula miscarile unui ventilator in functie de temperatura ambientală; de asemenea si umiditatea ambientală este preluata de catre un senzor DHT22, iar cele doua valori cat si frecventa de rotatie a ventilatorului (exprimata in procente) vor fi afisate pe un ecran LCD ce dispune de 4 linii de afisare cu 20 de caractere pe fiecare linie.

Descriere generala

Functionalitatea ventilatorului se va desfasura intr-un interval de temperaturi, valoarea minima fiind de aproximativ 27°C, care daca nu este atinsa, ventilatorul nu va porni. In functie de valoarea temperaturii ambientale, ventilatorul va avea o frecventa de rotatie ce oscileaza, iar atunci cand valoarea temperaturii va atinge valoarea maxima (aprox 40°C), ventilatorul va functiona cu viteza maxima (Fan Speed 100%). Ulterior, temperatura, umiditatea si frecventa vor fi afisate pe un ecran LCD ce comunica cu placa Arduino prin protocolul I2C.

Schema bloc:



Hardware Design

Schema electrica:

Pentru comunicarea intre placa Arduino UNO si senzorul de umiditate si temperatura am folosit pinul A0, ecranul este conectat la Arduino prin pinii SCL, SDA, VCC si GND (m-am folosit de protocolul I2C pentru a face comunicarea mai usoara), iar ventilatorul comunica cu placa prin intermediul bazei tranzistorului 2N2222 care este legata la placa prin pinul D9, am conectat GND-ul de la baterie la emitorul tranzistorului cat si GND-ul ventilatorului la colectorul tranzistorului, iar emitorul

tranzistorului a fost legat la GND-ul placii Arduino.



Lista componente:

- [Arduino UNO](#)
- [Breadboard](#)
- [Senzor DHT22](#)
- [Ventilator DC 12V](#)
- [Tranzistor 2n2222](#)
- [Baterie de 9V](#)
- [Ecran LCD 20*4 cu I2C](#)
- [Fire tata-tata](#)
- [Fire mama-tata](#)

Software Design

Descriere firmware

- Pentru realizarea schemei electrice am folosit [Proteus](#)
- Mediul de dezvoltare: [Arduino IDE](#)
- Biblioteci folosite:
 1. [LiquidCrystal](#)
 2. [DHT Library](#)

Rezultate Obținute

Pentru prezentarea rezultatelor obtinute am realizat un video ce poate fi accesat la urmatorul link.

[Video Prezentare Rezultate Obtinute](#)

Concluzii

Acest proiect a fost o buna ocazie de a vedea cum se poate programa o componenta electrica dupa un anumit scop, sau dupa o anumita dorinta. Implementarea acestui proiect a fost interesanta si intriganta, dupa parerea mea, si ma bucur ca pe parcursul ei nu am intampinat niciun fel de problema

sau defectiune a unei componente, si ca totul a decurs exact cum m-am asteptat. Pe langa partea de conectare efectiva a componentelor am avut ocazia sa folosesc si un soft specializat de design hardware, precum Proteus, care mi-a fost de foarte mare ajutor in gasirea componentelor electrice de care am avut nevoie pentru a crea atat proiectul, cat si schema electrica. A fost o experienta unica (primul proiect hardware de pana acum) si sunt multumit ca l-am dus la bun sfarsit.

Download

[Arhiva cu cod sursa + README](#)

Jurnal

- 26 Aprilie -> alegere tema proiect, creare pagina proiect pe OCW
- 29 Aprilie -> adaugare introducere, schema bloc, descriere generala proiect pe OCW
- 11 Mai -> achizitionarea componentelor necesare proiectului si testarea lor pentru asigurarea functionalitatii corecte a acestora
- 18 Mai -> adaugare schema electrica pe OCW
- 21 Mai -> inceperea realizarii partii software a proiectului
- 27 Mai -> finalizarea functionalitatii proiectului si testarea acestuia, adaugare cod sursa, README

Bibliografie/Resurse

- [Export to PDF](#)
- [DHT22 Sensor datasheet](#)
- [DHT11 vs DHT22](#)
- [Arduino UNO for beginners](#)
- [How to use a breadboard](#)
- [Liquid Crystal I2C Library](#)
- [DHT11 & DHT22 Library](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/temperature-controlled_fan



Last update: **2021/06/01 13:39**

