

Compresor Auto

Petru Zincenco
333AC

La început, nu mi-a fost clar cum să abordez acest proiect, deoarece ideile mele nu aveau aplicabilitate practică în utilizarea mea de zi cu zi. Totul s-a schimbat într-o zi, când mașină mea a făcut pană și am constatat că compresorul meu nu funcționa. Astfel, obiectivul acestui proiect este de a automatiza funcționarea compresorului, astfel încât să îmi fie mult mai ușor să îl operez.

Componente hardware

Step-down joacă un rol crucial în acest proiect, deoarece Arduino Nano necesită o tensiune de alimentare de 5V, în timp ce tensiunea disponibilă în bateria mașinii variază între 12-14V. Prin intermediul acestei componente, tensiunea este convertită la nivelul necesar pentru a asigura funcționarea corectă a Arduino Nano.

Senzorul de presiune are capacitatea de a măsura presiunea până la 4 bar. Scopul acestuia constă în a evalua presiunea din roată și a transmite datele analogice la Arduino Nano.

Releul are funcția de a activa și dezactiva compresorul, fiind controlat de către Arduino Nano.

Pe afișajul de 16x2 caractere vor fi afișate următoarele informații: presiunea în roată și presiunea dorită. Comunicarea între Arduino Nano și afișajul LCD este realizată prin intermediul protocolului I2C.

Componentele hardware utilizate în proiect sunt:

- Microcontrollerul Arduino Nano
- Senzor de presiune
- Display LCD 16x2 I2C
- Buton rotativ (OKY3431-4)
- Releu (SSR-40DD)
- Step down (LM2596)



Puteti obține o copie în format PDF a schemei electrice prin descărcare de aici.

[schematic_carcompressor.pdf](#)

Software

Software-ul include următoarele funcționalități: ADC (convertor analog-digital), Timer și I2C. Acestea sunt utilizate pentru prelucrarea datelor și controlul compresorului.

Mai jos puteți vedea schema bloc a proiectului:



Descrierea funcțiilor:

În funcția `setup()`, se realizează inițializarea pinilor și a afișajului LCD. De asemenea, se configurează întreruperile pentru encoder și un timer pentru citirea valorii presiunii la fiecare 100 de milisecunde.

Funcția `shaftMoved()` este apelată atunci când encoderul este rotit și ajustează presiunea țintă în funcție de direcția de rotație a encoderului.

Funcția `setPressure()` afișează pe LCD valoarea presiunii țintă setată de utilizator. Dacă butonul de pe encoder este apăsat, se trece la următoarea etapă a procesului.

Funcția `pumpAir()` afișează pe LCD valoarea presiunii țintă și presiunea reală citită. Se activează compresorul și se monitorizează presiunea. Dacă presiunea reală depășește valoarea țintă, compresorul se oprește și procesul se încheie.

Funcția `displayForceStop()` afișează un mesaj de oprire forțată pe LCD și oprește compresorul. Apoi, se revine la meniul principal.

Funcția `forceStop()` verifică dacă butonul a fost apăsat timp de cel puțin 1 secundă pentru a opri procesul de umflare în mod forțat.

În funcția `loop()`, se verifică dacă procesul trebuie oprit în mod forțat sau dacă se continuă cu procesul de umflare. În funcție de valoarea variabilei `panel`, se apelează funcțiile corespunzătoare.

Funcția `getPressure()` citește valoarea presiunii de la un senzor analog și o convertește într-un interval de la 0 la 10.

În întreruperea `TIMER1_COMPA_vect`, se actualizează valoarea presiunii și se gestionează timpul de așteptare pentru citirea precisă a presiunii.

Această întrerupere controlează atât citirea presiunii din roată, cât și activarea compresorului. Pentru a asigura o citire corectă a presiunii, compresorul nu poate fi pornit în același timp, așadar compresorul va fi activ timp de 35 de secunde, urmat de o secundă în care se va efectua citirea presiunii.

Mediu de dezvoltare

- Ca mediu de dezvoltare am folosit Arduino IDE.
- Pentru controlul display-ului cu I2C am folosit biblioteca LiquidCrystal_I2C

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

Download

Codul sursă pentru compresor poate fi descărcat de aici. [automatedcompressor.zip](#)

Puteți descărca o copie a paginii în format PDF de aici. [Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/alucaci/automatedcompressor>



Last update: **2023/05/30 09:05**