

Ministerul Educației și Cercetării
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Ingineria Software și Automatică

Raport

Curs: Internetul lucrurilor

Tema: Sensori - Achiziția de informații.

A elaborat:

st.gr.SI-211 Chirița Stanislav

A verificat:

Asist. Univ.Astafi Valentina

Chișinău 2024

Definirea problemei 2

Realizarea unei aplicații pentru MCU care va controla un motor cu comenzi receptionate de la interfața serială.

Aplicația va efectua

1. Citirea comenzilor serial;
2. Prelucrarea comenzilor;
3. Modificarea vitezei și direcției de rotație a motorului.

Obiective

- configurarea aplicației pentru citirea și prelucrarea comenzilor seriale;
- crearea schemei și codului conform sarcinii propuse;
- înțelegerea sistemului creat și procesele ce se întâmplă.

Introducere

Actuatorii reprezintă o componentă cheie în automatizarea și controlul sistemelor, având un rol vital în traducerea semnalelor sau datelor digitale în acțiuni fizice. Aceste dispozitive sunt absolut esențiale într-o gamă variată de domenii, de la industria manufacturieră și robotică până la domeniul medical, automotive și chiar în mediul casnic. Ei acționează ca intermediari între lumea digitală și cea fizică, transformând instrucțiuni și comenzi în mișcări, presiune, temperatură sau alte acțiuni specifice. Fără acționarea acuatorilor, multe dintre facilitățile și inovațiile tehnologice pe care le experimentăm în societatea modernă ar fi dificil de realizat sau chiar imposibil de obținut.

De exemplu, în industria manufacturieră, acționarea precisă a roboților industriali este esențială pentru asamblarea produselor. În domeniul medical, acționarea precisă a dispozitivelor medicale poate face diferența între viață și moarte. În domeniul automotive, controlul adecvat al frânelor, direcției și altor componente este crucial pentru siguranța șoferilor și pasagerilor. Acuratețea, fiabilitatea și reactivitatea acționării acuatorilor sunt, prin urmare, esențiale pentru funcționarea optimă a sistemelor automate și pentru satisfacerea cerințelor tehnologice din ce în ce mai complexe ale societății moderne.

Metode și materiale

Materiale Necesare:

- Arduino Board (de exemplu, Arduino Uno);
- breadboard și fire de conexiune;
- un motor în current continuu;
- editor Arduino IDE instalat pe un calculator;
- un LCD display.

Metoda de Implementare:

- conexiuni hardware:
 - conectarea motorului la pini Arduino.
- scrierea codului în Arduino IDE:
 - definirea variabilelor și pinilor corespunzători pentru motorul;
 - implementare funcții pentru citirea comenzilor serial, schimbarea vitezei, direcției a motorului.
- testarea și debugging:
 - încărcare cod pe Arduino folosind Arduino IDE;
 - monitorizare comportament motor;
 - identificare și rezolvare eventualelor erori în cod.

Rezultate

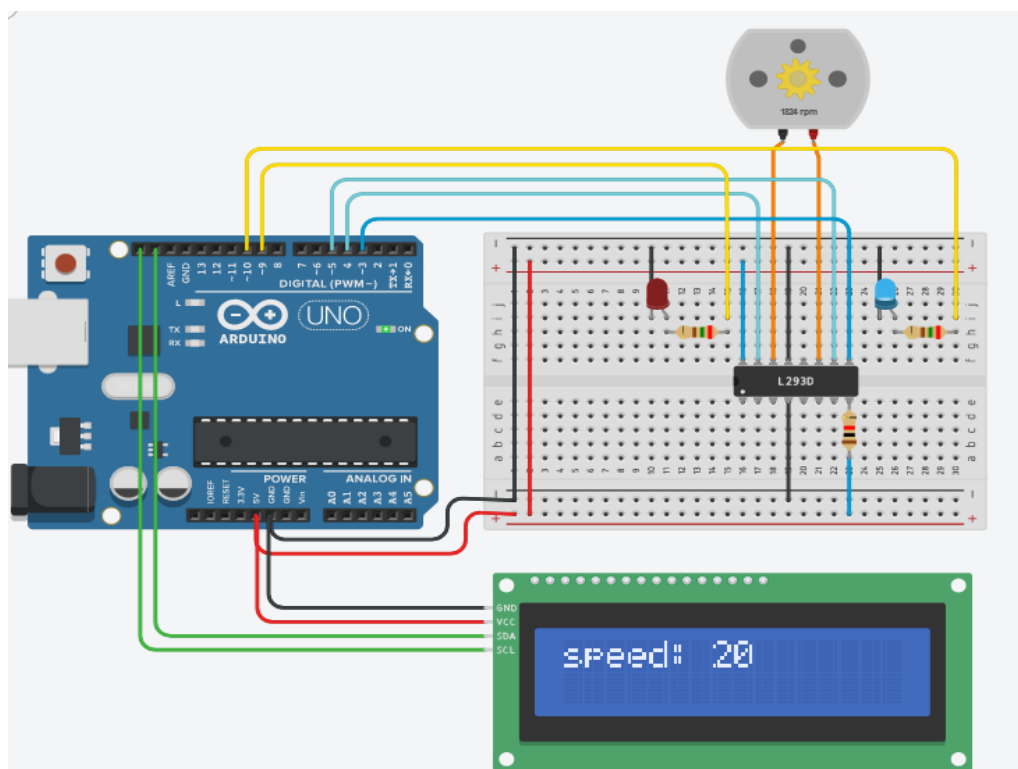


Figura 2 – Asamblarea circuitului pentru sarcina 2

Concluzii

În urma acestei lucrări de laborator, s-a dezvoltat o aplicație care permite controlul dispozitivelor de acționare, cum ar fi un bec electric și un motor de curent continuu, prin intermediul comenzilor primite de la o interfață serială. Aceste comenzi permit activarea și dezactivarea becului electric și controlarea motorului în ambele direcții cu o putere variabilă între -100% și 100%. Interfața serială facilitează comunicarea cu MCU, iar starea și setările dispozitivelor de acționare sunt raportate pe un afișaj LCD. De asemenea, pentru controlul acestor periferice s-au creat drivere la niveluri de abstractizare, asigurând astfel o gestionare eficientă și modulară a funcționalității dispozitivelor.

Anexa 2 – Codul sursă sarcina 2

```
#include <stdio.h>

#include <Adafruit_LiquidCrystal.h>

int mt = 3;

int dir1 = 5;

int dir2 = 4;

int L = 9;

int R = 10;

int speed = 0;

Adafruit_LiquidCrystal lcd(0);

void setup()

{

    Serial.begin(9600);

    pinMode(mt, OUTPUT);

    pinMode(dir1, OUTPUT);

    pinMode(dir2, OUTPUT);

    pinMode(L, OUTPUT);

    pinMode(R, OUTPUT);

    digitalWrite(dir1, LOW);

    digitalWrite(dir2, LOW);

    lcd.begin(16, 2);

}

void loop() {

    lcd.clear();

    speed = 0;

    if (Serial.available() > 0) {

        String command = Serial.readStringUntil('\n');

        speed = command.toInt();

        if (speed >= -100 && speed <= 100) {
```

```
if (speed >= 0) {

    digitalWrite(dir1, HIGH);

    digitalWrite(dir2, LOW);

    digitalWrite(R, HIGH);

    digitalWrite(L, LOW);

    analogWrite(mt, map(speed, 0, 100, 0, 255));

    lcd.print("speed: " + String(speed));

} else {

    digitalWrite(dir1, LOW);

    digitalWrite(dir2, HIGH);

    digitalWrite(R, LOW);

    digitalWrite(L, HIGH);

    analogWrite(mt, map(-speed, 0, 100, 0, 255));

    lcd.print("speed: " + String(speed));

}

} else {

    Serial.println("ERROR, between -100,100");}

delay(2000);}}
```