**MINISTERUL EDUCAŢIEI, CULTURII ȘI CERCETĂRII**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică şi Microelectronică**

 **Departamentul Ingineria Software și Automatică**

Raport

**la disciplina Internetul lucrurilor IOT**

**Tema: Actuatori**

A efectuat: st.gr. TI-204 Agatiev Dumitru

A verificat: asist.univ. Litra Dinu

Chișinău 2023

**Definirea problemei 1**

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

Aplicația va efectua

1. Citirea comenzilor serial;
2. Prelucrarea comenzilor;
3. Schimbarea stării releului;

**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 bec electric;
* editor Arduino IDE instalat pe un calculator;
* un LCD display.

Metoda de Implementare:

* conexiuni hardware:
* conectarea becului electric la un pin analogic Arduino;
* scrierea codului în Arduino IDE:
* definirea variabilelor și pinilor corespunzători pentru becul și motorul;
* implementare funcții pentru citirea comenzilor serial, schimbarea stării a becului.
* testarea și debugging:
* încărcare cod pe Arduino folosind Arduino IDE;
* monitorizare comportament bec;
* identificare și rezolvare eventualelor erori în cod.

**Rezultate**

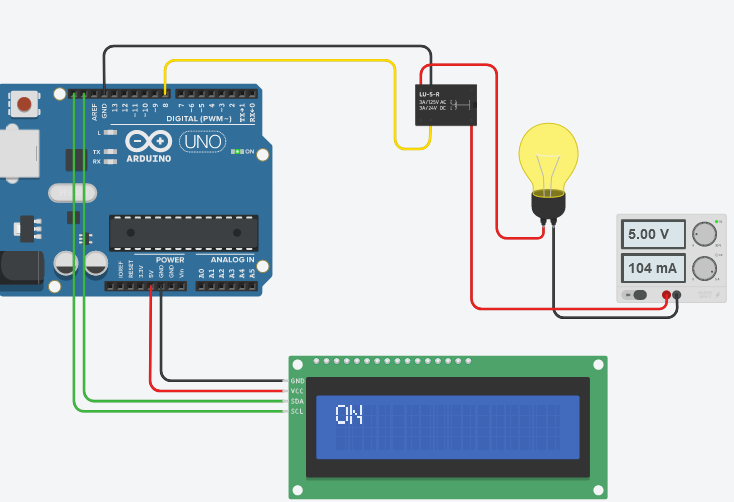
**

Figura 1 – Asamblarea circuitului pentru sarcina 1

**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 motrolului 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**

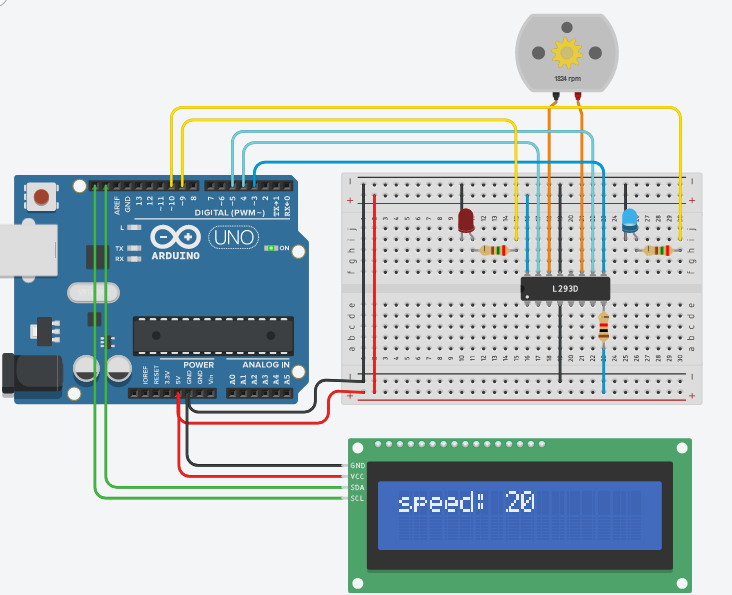
**

Figura 2 – Asamblarea circuitului pentru sarcina 2

**Concluzii**

În urma acestui laborator, am confirmat rolul vital pe care actuatorii îl au în domeniul automatizării și controlului sistemelor, deoarece ei transformă semnalele de comandă în acțiuni fizice tangibile, precum activarea unui releu sau controlul unui motor. Implementarea cu succes a codului de control pentru acești actuatori este esențială pentru asigurarea preciziei și controlului fin asupra dispozitivelor hardware.

Unul dintre rezultatele notabile ale acestei experiențe de laborator a fost reușita noastră în controlul unui releu, care a demonstrat abilitatea noastră de a activa sau dezactiva o sursă de alimentare, cum ar fi un bec, folosind comenzile primite prin intermediul portului Serial. Această funcționalitate simplă ne-a deschis uși pentru controlul dispozitivelor de bază, iar dezvoltarea și testarea controlului motorului ne-au permis să ajustăm cu succes viteza și direcția acestuia. Prin intermediul interfeței Serial, am gestionat intrările de la utilizator și am asigurat funcționarea corectă a motorului într-un interval specific de viteză și direcție.

Această experiență de laborator nu reprezintă doar o reușită tehnică, ci și o etapă semnificativă în procesul nostru de învățare. Am acumulat cunoștințe esențiale privind configurarea și controlul actuatorilor într-un mediu controlat de un microcontroler, deschizând calea către o mai bună înțelegere a automatizării și controlului sistemelor.

Codul pe care l-am dezvoltat în cadrul acestui laborator poate servi ca temelie pentru proiecte ulterioare și dezvoltarea sistemelor mai complexe. Această experiență ne pregătește pentru a explora domenii avansate precum robotică, automatizare industrială și control de proces, în care cunoștințele dobândite aici vor juca un rol esențial.

**Anexa 1 – Codul sursă sarcina 1**

#include <stdio.h>

#include <Adafruit\_LiquidCrystal.h>

int led = 8;

Adafruit\_LiquidCrystal lcd(0);

void setup()

{

Serial.begin(9600);

lcd.begin(16, 2);

pinMode(led, OUTPUT);

}

void loop() {

lcd.clear();

if (Serial.available() > 0) {

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

if (command == "ON") {

digitalWrite(led, HIGH);

Serial.println("on");

lcd.print("ON");

} else if (command == "OFF") {

digitalWrite(led, LOW);

Serial.println("off");

lcd.print("OFF");

}

delay(1000);

}}

**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);}}