Ministerul Educației și Cercetării al Republicii Moldova Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică și Microelectronică Departamentul Ingineria Software și Automatică

RAPORT

**Lucrarea de laborator nr. 4**

**Disciplina: Internetul lucrurilor**

**Tema: Actuatori cu interfata binara** - **Releu**.

A efectuat:

st. gr.TI-215 Josan Artiom

A verificat:

Asistent Universitar Litra Dinu

Chişinău 2024

**Tema: Actuatori cu interfata binara** - **Releu**.

**Obiectiv:** Crearea unei aplicații pentru generarea de acțiuni în mediul extern, conversia unui semnal în putere, lucrul cu o unitate.

**Definirea problemei:**

Creați o aplicație bazată pe MCU care va controla unitățile de acționare folosind comenzile primite de la interfața serială și raportate la LCD.

Mecanismele executive vor fi următoarele:

* bec printr-un releu cu comenzi de pornire si oprire;
* Motor DC cu comenzi pentru setarea puterii motorului în interval (-100% .. 100%), adică înainte și înapoi și viteză prin șoferul L298.
* Driverele de control periferice vor fi executate la niveluri de abstractizare.

**Scurtă teorie:**

Arduino poate genera un semnal pentru a controla un actuator liniar, cu toate acestea, acest semnal are tensiune și curent scăzut, așa că nu puteți utiliza doar actuatorul. Pentru a face acest lucru, trebuie să utilizați un driver hardware între Arduino și unitate. Există multe tipuri de circuite integrate și module (cum ar fi L293D, L298N) care pot fi utilizate ca drivere de acţionare liniare.

Driverul L298N are două canale, numite canal A și canal B. Astfel, driverul L298N poate controla două actuatoare liniare independent în același timp.

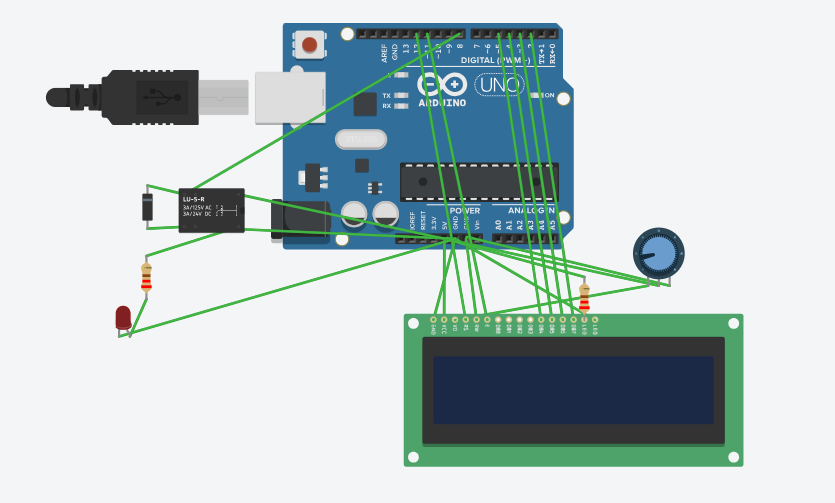
Materiale

**Instrumente**

* LCD - afișaj pentru afișarea informațiilor.
* Arduino UNO - un microcontroler.
* Rezistor - folosit pentru reglarea tensiunii.
* Bec - se va aprinde și se va stinge prin intermediul unui releu.
* Releu - un dispozitiv care vă permite să comutați echipamentul în modul automat.
* Acționare electrică cu punte în H - pentru acționarea unui motor electric.
* Un motorreductor este un dispozitiv care combină funcțiile unui motor electric și ale unei cutii de viteze.

**Rezultate**

Comenzile introduse prin portul serial sunt procesate corect, aprinzând lumina și afișând rezultatele pe ecran, controlând funcționarea motorreductorului, ceea ce indică scrierea corectă a codului și asamblarea circuitului electric.



***Figura 1 - Schema de circuit pentru led și releu***

**Codul:**

#include <Wire.h>

#include <stdio.h>

#include <stdlib.h>

#include <LiquidCrystal.h>

#define RELAY\_PIN 13

LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

int lcdCol = 0;

int lcdRow = 0;

int rawBufferCount = 0;

void setup() {

lampInit();

Serial.begin(9600);

LCD.begin(16, 2);

FILE\* stream = fdevopen(lcdPutChar, NULL);

stdin = stdout = stream;

}

void loop() {

if (Serial.available()) {

String command = Serial.readString();

if (command == "on") {

lampOn();

printf("\n\nThe lamp is on");

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

lampOff();

printf("\n\nThe lamp is off");

}

}

}

void clear() {

if (lcdRow > 1) {

lcdRow = 0;

lcdCol = 0;

LCD.clear();

}

}

void newLine() {

lcdRow++;

clear();

lcdCol = 0;

}

int lcdPutChar(char c, FILE\* f) {

if (lcdCol > 15) {

newLine();

}

if (c == '\n') {

newLine();

} else {

LCD.setCursor(lcdCol, lcdRow);

LCD.print(c);

lcdCol++;

}

return 0;

}

void lampInit() {

relayInit();

}

void lampOn() {

relayOn();

}

void lampOff() {

relayOff();

}

void relayInit() {

pinMode(RELAY\_PIN, OUTPUT);

}

void relayOn() {

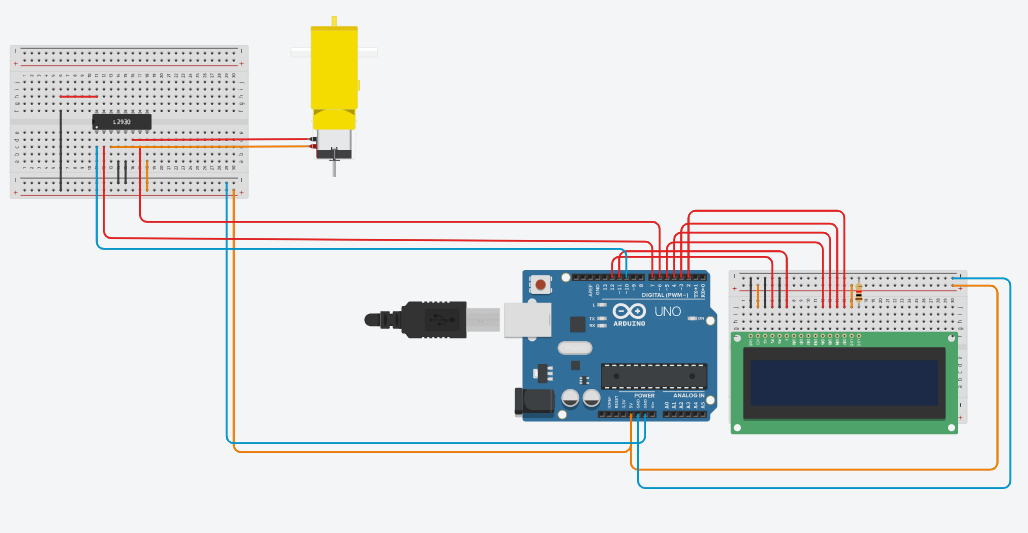
digitalWrite(RELAY\_PIN, HIGH);

}

void relayOff() {

digitalWrite(RELAY\_PIN, LOW);

}

**4.2. Actuatori cu interfata analogica** - **Motor in curent continuu**

***Figura 2 - Schema de circuit pentru un motor electric și un motorreductor***

**Cod program:**

#include <stdio.h>

#include <LiquidCrystal.h>

#define RIGHT\_DIR (1)

#define LEFT\_DIR (-1)

#define L293D\_IN1\_PIN 7

#define L293D\_IN2\_PIN 6

#define L293D\_ENA\_PIN 10

#define MOTOR\_MAX\_POWER 143

LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

int lcdCol = 0;

int lcdRow = 0;

int motorPower = 0;

void setup() {

motorInit();

Serial.begin(9600);

LCD.begin(16, 2);

FILE\* stream = fdevopen(lcdPutChar, NULL);

stdin = stdout = stream;

}

void loop() {

if (Serial.available()) {

String command = Serial.readString();

if (command == "up") {

motorPowerUp(20);

printf("\n\nPower up +20");

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

motorPowerDown(20);

printf("\n\nPower down -20");

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

motorStop();

printf("\n\nStop");

}

}

}

void clear() {

if (lcdRow > 1) {

lcdRow = 0;

lcdCol = 0;

LCD.clear();

}

}

void newLine() {

lcdRow++;

clear();

lcdCol = 0;

}

int lcdPutChar(char c, FILE\* f) {

if (lcdCol > 15) {

newLine();

}

if (c == '\n') {

newLine();

} else {

LCD.setCursor(lcdCol, lcdRow);

LCD.print(c);

lcdCol++;

}

return 0;

}

void motorInit() {

l293dInit();

}

void motorPowerUp(int powerOffset) {

motorPower = motorSetPower(motorPower + powerOffset);

}

void motorPowerDown(int powerOffset) {

motorPower = motorSetPower(motorPower - powerOffset);

}

void motorStop() {

l293dIn1Low();

l293dIn2Low();

motorPower = 0;

}

int motorSetPower(int motorPower) {

if (motorPower > MOTOR\_MAX\_POWER) {

motorPower = MOTOR\_MAX\_POWER;

}

if (motorPower < -MOTOR\_MAX\_POWER) {

motorPower = -MOTOR\_MAX\_POWER;

}

int powerOut = abs(motorPower);

l293dSetEnA(powerOut);

if (motorPower > 0) {

motorSetDirection(RIGHT\_DIR);

} else {

motorSetDirection(LEFT\_DIR);

}

return motorPower;

}

void motorSetDirection(int motorDirection) {

if (motorDirection == RIGHT\_DIR) {

l293dIn1High();

l293dIn2Low();

} else if (motorDirection == LEFT\_DIR) {

l293dIn1Low();

l293dIn2High();

}

}

void l293dInit() {

pinMode(L293D\_IN1\_PIN, OUTPUT);

pinMode(L293D\_IN2\_PIN, OUTPUT);

pinMode(L293D\_ENA\_PIN, OUTPUT);

}

void l293dIn1High() {

digitalWrite(L293D\_IN1\_PIN, HIGH);

}

void l293dIn1Low() {

digitalWrite(L293D\_IN1\_PIN, LOW);

}

void l293dIn2High() {

digitalWrite(L293D\_IN2\_PIN, HIGH);

}

void l293dIn2Low() {

digitalWrite(L293D\_IN2\_PIN, LOW);

}

void l293dSetEnA(int powerOut) {

int outPwm = map(powerOut, 0, 143, 0, 255);

analogWrite(L293D\_ENA\_PIN, outPwm);

}