**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică şi Microelectronică**

**Departamentul Inginerie Software și Automatică**

**Programul de studii: Tehnologia Informației**

**Lucrare de laborator Nr.4**

Disciplina: Internet of Things

Tema: Actuatori

**A efectuat:** st.gr.TI-194,Zavorot Daniel

**A verificat:** asist. univ., Litra Dinu

**Chişinău  2022**

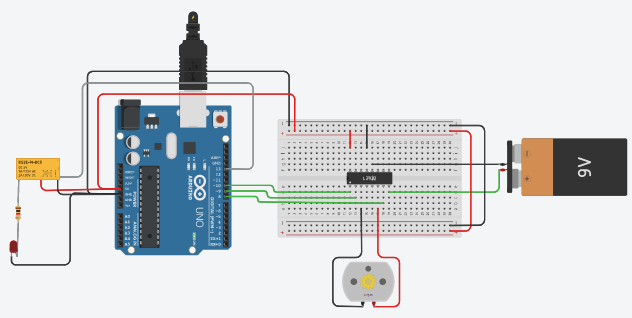
**Sarcina lucrării:**

  Sa se realizeze o aplicație in baza de MCU care va controla dispozitivele de acționare cu comenzi recepționate de la interfața seriala si raportare către LCD.   
Dispozitivele de acționare vor fi următoarele:

* un bec electric prin intermediul releului cu comenzi de ON si OFF
* un motor in curent continuii cu comenzi de setare a puterii motorului intre (-100% .. 100%) adică înainte si înapoi, si viteza prin intermediul driverului L298

Driverele de control a periferiilor se vor realiza pe nivele de abstracție

**Rezultate**



**Concluzia:**

In urma elaborării laboratorului dat, ne-am familiarizat cu actuatorii si am făcut diferența intre un senzor si un actuator, fiind device-uri opuse (unul prea informația din mediu, altul o preia din interior). De asemenea cu ajutorul motorului am putut vedea cum informația interna este transmisa spre mediul extern.

**Anexe**

**Anexa A** – Listing cod sursă

#include <LiquidCrystal\_I2C.h>  
  
#define RELAY\_PIN 11  
  
static FILE uartout = {0} ;  
  
LiquidCrystal\_I2C lcd(0x27, 20, 4);  
  
static int my\_putChar( char ch, FILE \* stream)  
{  
 lcd.print( ch );  
 return 0;  
}  
  
char my\_GetChar(FILE \* f)  
{  
 while (! Serial.available());  
 return Serial.read();  
}  
  
void setup()  
{  
 Serial.begin(9600);  
 fdev\_setup\_stream (&uartout, my\_putChar, my\_GetChar, \_FDEV\_SETUP\_RW);  
 stdin = stdout = &uartout ;  
  
 lcd.init();  
 lcd.backlight();  
  
 pinMode(RELAY\_PIN, OUTPUT);  
  
 Serial.print("Start\n");  
 printf("Start");  
}  
  
  
void loop()  
{  
 char command[20] = {};  
 scanf("%s", command);  
  
 if (strcmp(command, "on") == 0)  
 {  
 lcd.clear();  
 digitalWrite(RELAY\_PIN, HIGH);  
 Serial.print("BULB is turned ON\n");  
 printf("BULB is turned ON");  
 }  
 else if (strcmp(command, "off") == 0)  
 {  
 lcd.clear();  
 digitalWrite(RELAY\_PIN, LOW);  
 Serial.print("BULB is turned OFF\n");  
 printf("BULB is turned OFF");  
 }  
 delay(2000);  
}

**Anexa B** – Listing cod sursă

#include <LiquidCrystal\_I2C.h>

#define IN\_1 2

#define IN\_2 3  
#define PWM 10

#define MIN\_VAL -100

#define MAX\_VAL 100

int percentSpeedValue = 0;

int newValue = 0;

int direction = 0;

static FILE uartout = {0};

LiquidCrystal\_I2C lcd(0x27, 20, 4);

static int my\_putChar(char ch, FILE \*stream) {

lcd.print(ch);

return 0;

}

char my\_GetChar(FILE \*f) {  
 while (!Serial.available());

return Serial.read();

}

int checkLimit(int speed) {

if (speed < MIN\_VAL || speed > MAX\_VAL) {

return 0;

}

return speed;

}

// Layer I

void setPwm(int pwmValue) {

analogWrite(PWM, pwmValue);

}

void pinControl(bool a, bool b) {

digitalWrite(IN\_1, a);

digitalWrite(IN\_2, b);

}

// Layer II

void setSpeed(int percentSpeed) {

int pwmSpeed = 0;

pwmSpeed = (int) percentSpeed \* 2.55;

setPwm(pwmSpeed);

}

int setDirection(int direction) {

if (direction < 0) {

pinControl(1, 0);

return 1;

}

if (direction > 0) {

pinControl(0, 1);

return 2;

}

if (direction == 0) {

pinControl(0, 0);

return 0;

}

return -1;

}

// Layer III

void motorControl(int direction, int speed) {

setSpeed(speed);

setDirection(direction);

}

// Layer IV

void displayToLcd(int direction, int speed\_value) {

lcd.clear();

switch (direction) {

case 1:

printf("Inainte: %d", speed\_value);

break;

case -1:

printf("Inapoi: %d", speed\_value);

break;

default:

printf("Stop");

break;

}

}

void setup() {

Serial.begin(9600);

fdev\_setup\_stream(&uartout, my\_putChar, my\_GetChar, \_FDEV\_SETUP\_RW);

stdin = stdout = &uartout;

lcd.init();

lcd.backlight();

pinMode(IN\_1, OUTPUT);

pinMode(IN\_2, OUTPUT);

pinMode(PWM, OUTPUT);

Serial.print("Start\n");

printf("Start");

percentSpeedValue = Serial.parseInt();

}

void loop() {

newValue = Serial.parseInt();

if (newValue != 0) {

percentSpeedValue = checkLimit(newValue);

if (percentSpeedValue < 0) {

direction = -1;

percentSpeedValue \*= -1;

} else if (percentSpeedValue > 0) {

direction = 1;

} else {

direction = 0;  
 }

}

motorControl(direction, percentSpeedValue);

displayToLcd(direction, percentSpeedValue);

Serial.print("Speed: ");

Serial.println(percentSpeedValue);

}