BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

INTERFACING TECHNIQUES

ENCS4380

Task *#3*

DC Motors with Encoders & Serial Communication

Prepared by:

Mohammad Fareed **ID: 1212387**

Ahmad Hamdan **ID: 1210241**

Qossay Rida **ID: 1211553**

Mohammed Abed Alkareem **ID: 1210708**

Instructor: **Dr. Wasel Ghanem**

Teaching assistant: **Mr. Loai Shehadeh**

Date: **Friday, June 14, 2024**

Abstract

This project demonstrates using an encoder with a DC motor controlled by an Arduino, utilizing a joystick for speed and direction. Real-time feedback on motor position, speed, and direction is displayed on an LCD. This setup highlights the importance of feedback systems in motor control and automation.

Table of Content

Table of Figures

1. Theory

1.1 DC Motors

A DC motor is an electrical machine that converts electrical energy into mechanical energy. It is based on the idea that when a current carrying conductor is placed in a magnetic field, it produces mechanical force. The direction of the force is determined by the left hand rule. Since DC motors and DC generators have the same construction, they can be used interchangeably.

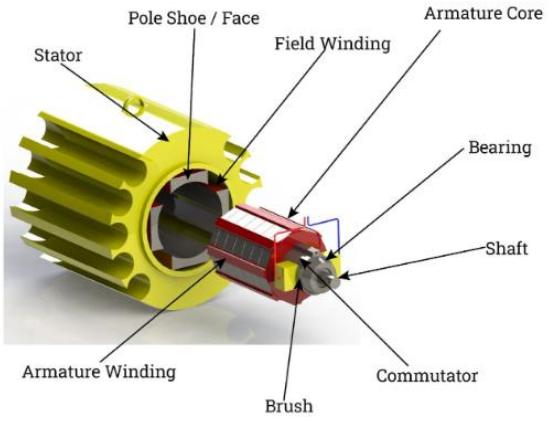


Figure 1: DC Motors

1.2 Encoders

DC motor encoders are used for speed control feedback in DC motors where an armature or rotor with wound wires rotates inside a magnetic field created by a stator. The DC motor encoder provides a mechanism to measure the speed of the rotor and provide closed-loop feedback to the drive for precise speed control.

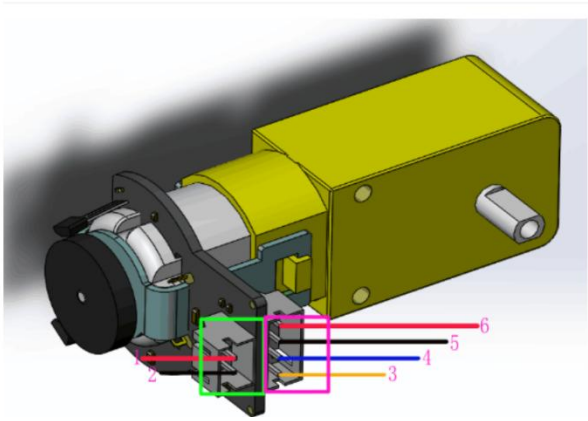


Figure 2: A DC Motor Encoder

2. Procedure

2.1 Tinkercad Circuit

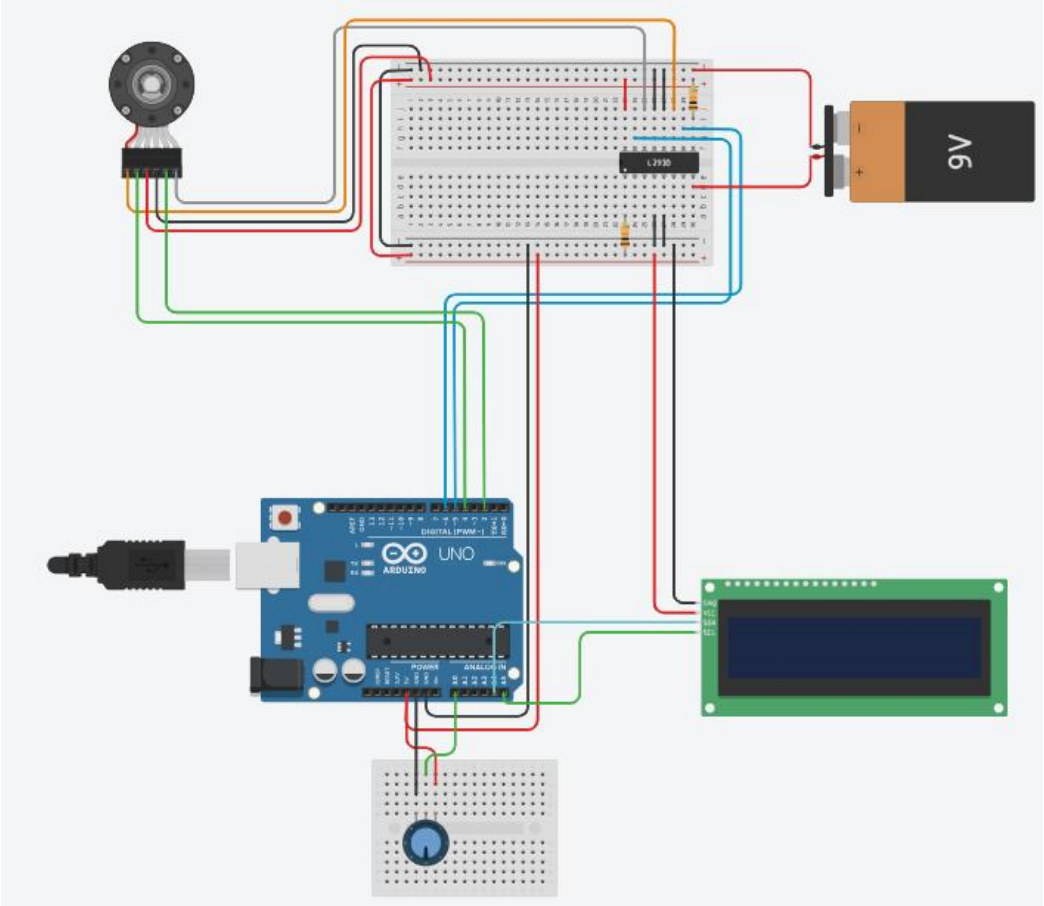


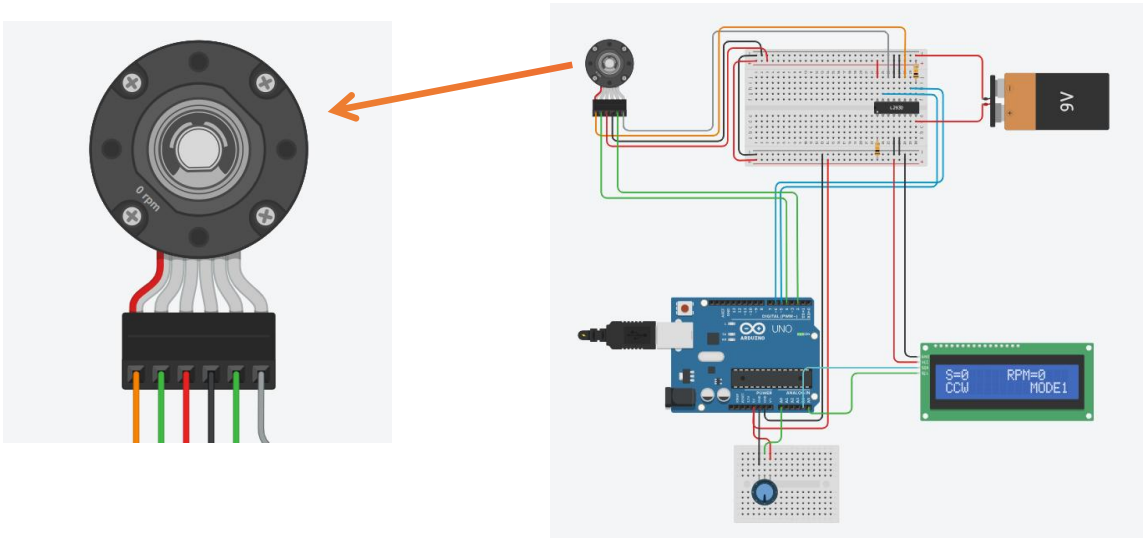
Figure 3: Tinkercad Circuit

We connected the Arduino Uno board with the potentiometer which will work as the Joystick in this task, and the DC motor with the encoder along with the LCD which will show the direction and speed of the motor.

After discussing with the teaching assistance Mr. Loai, we noticed that using the LCD with the I2C protocol causes an issue with the DC motor stopping it from running in tinkercad. We were unable to read the encoder despite the circuit working fine with no issues. We will be showing some of the pictures from tinkercad in this report.

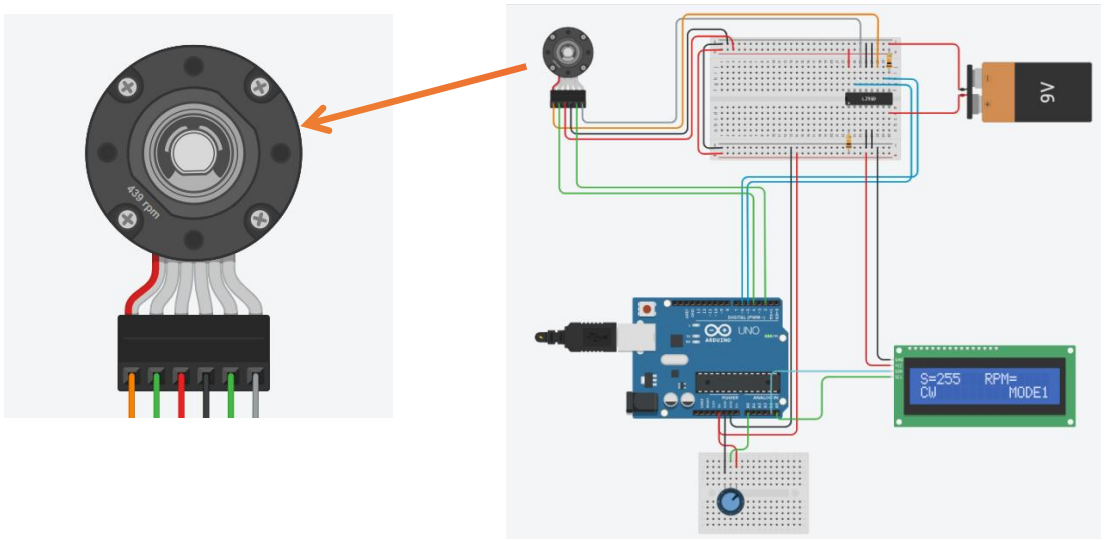
Morever, we will be using the results from the lab to explain and show our work, we attached a link to the drive where we created a full video displaying everything required along with the code and the link to the ti.

When the potentiometer is pointing down it means that the motor's speed is zero (direction is not important)



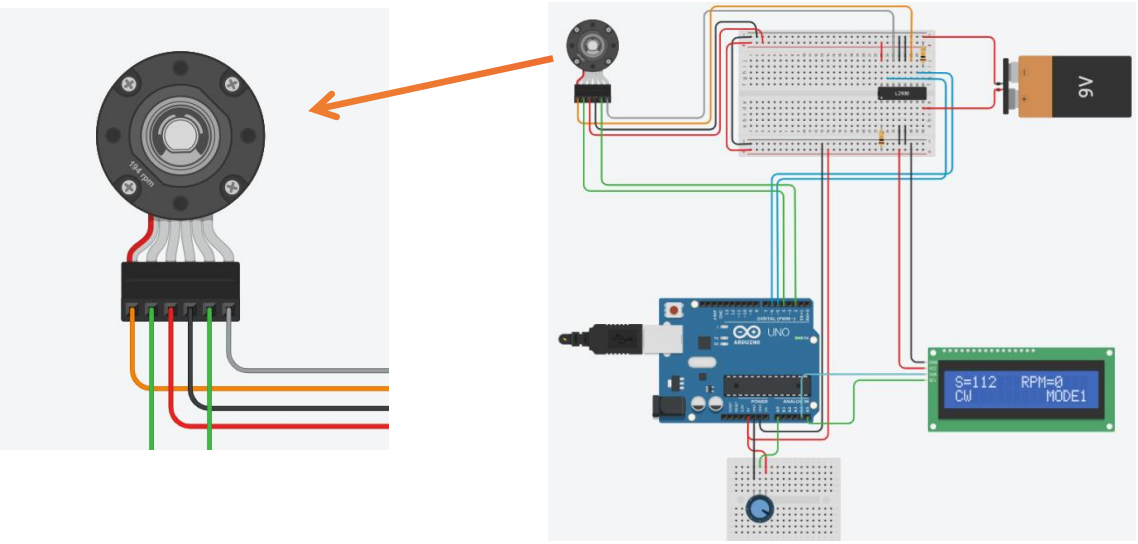
We can notice that the LCD shows that the Speed is zero and the direction which is not relevant in this case is CCW. (RPM sadly does not give a result in tinkercad, we will show it in the powerpoint slides for the on-hands circuit). But we can see the RPM on the motor.

When the potentiometer is increased to the right side, the speed will increase depending on the increase of the potentiometer and the motor will spin in the CW direction.



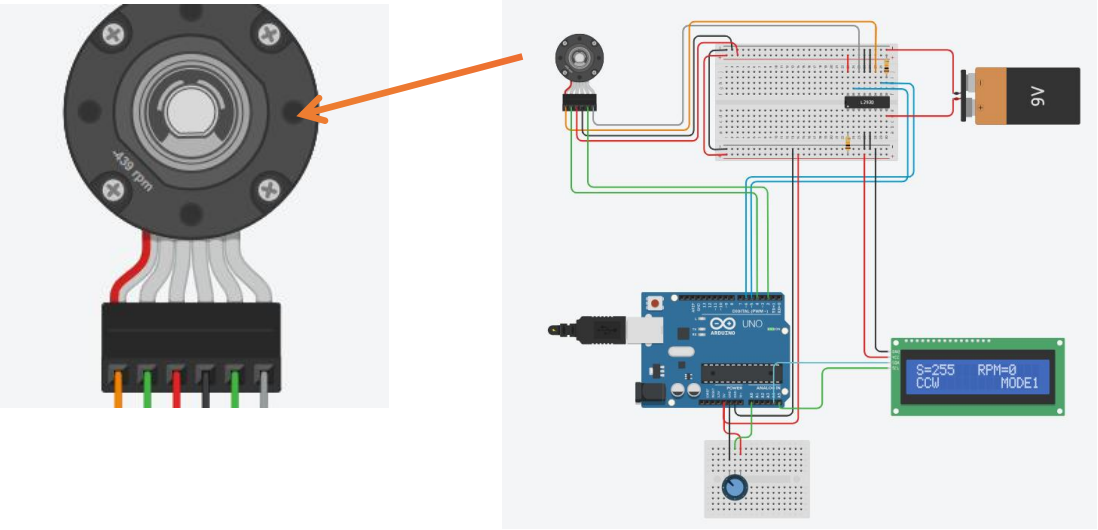
We can notice that the LCD shows that the Speed is 255 which is the maximum speed the motor can reach, and the direction which the motor is rotating in is CW. We can see that the motor is spinning in 439 rpm.

Now we will increase the potentiometer to the right side with a lower speed (meaning that we won't increase the potentiometer fully to the right). Let's not forget that the right side indicates a CW rotation.



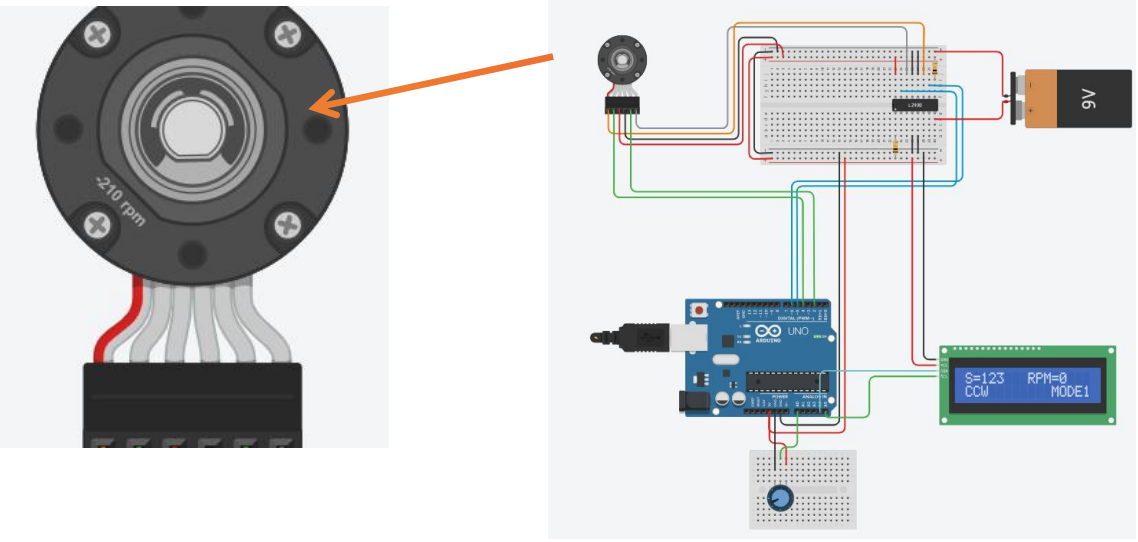
We can notice that the LCD shows that the Speed is 112 which is almost half of the maximum speed the motor can reach, and the direction which the motor is rotating in is CW. We can see that the motor is spinning in 193 rpm.

When the potentiometer is increased to the left side, the speed will increase depending on the increase of the potentiometer and the motor will spin in the CCW direction.



We can notice that the LCD shows that the Speed is 255 which is the maximum speed the motor can reach, and the direction which the motor is rotating in is CCW. We can see that the motor is spinning in -439 rpm.

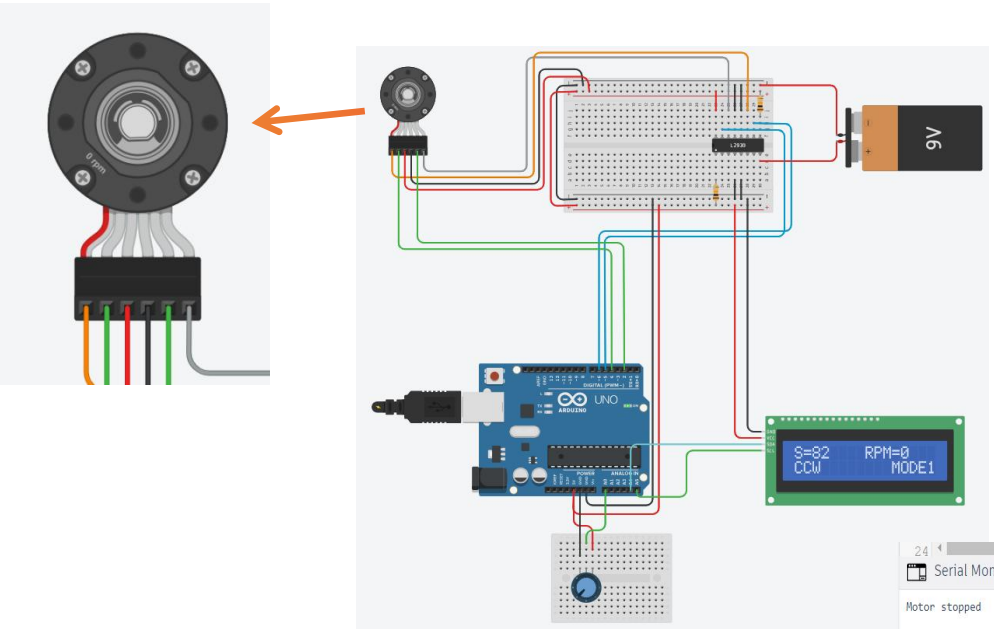
Now we will increase the potentiometer to the left side with a lower speed (meaning that we won't increase the potentiometer fully to the left). Let's not forget that the left side indicates a CCW rotation.



We can notice that the LCD shows that the Speed is 123 which is almost half of the maximum speed the motor can reach, and the direction which the motor is rotating in is CCW. We can see that the motor is spinning in -210 rpm.

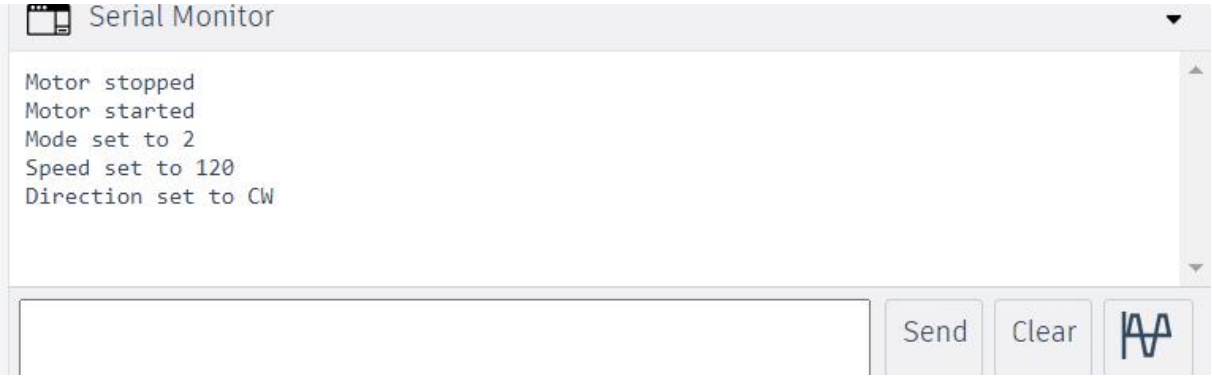
2.2 Serial Communication in Tinkercad

Eager to get the bonus, we implemented serial communication in this circuit. We will be controlling the motor using entered commands. We will be able to start and stop the motor, switch the motor to mode 2 which will disable any signals coming from the potentiometer and only allow us to control the motor with the commands.

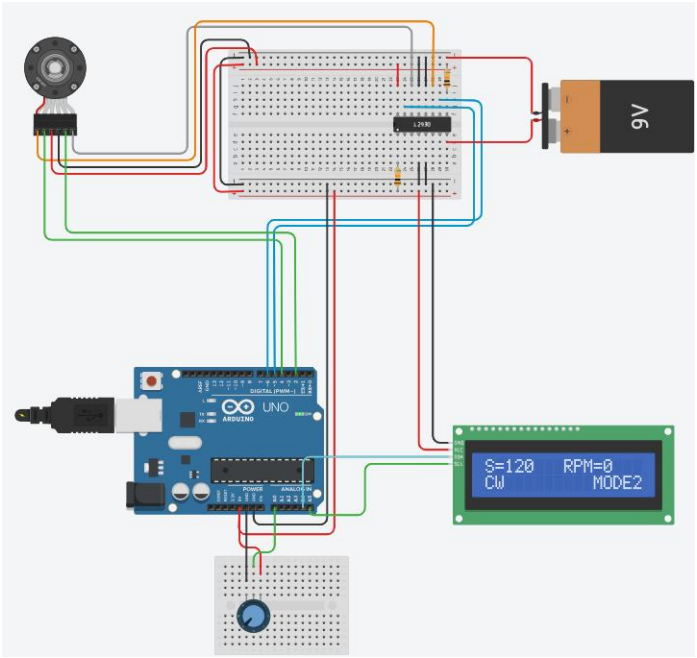


We notice that despite the potentiometer giving a signal, the motor is not spinning since we stopped it using serial comm. Of course, we added the “START” command to start it again.

In addition to the start and stop commands. We can switch the motor to mode 2 which will take the speed and direction from the commands as well.



First, we started the motor since it was already stopped in the previous part. Then we set the mode to mode 2 using the command “SET_MODE 2” , set the speed using the command “SET_SPEED 120” and set the direction using the command “SET_DIR CW”.



We can notice the result of our commands. The LCD shows that the mode is mode 2, the direction is CW and the speed is 120.

2.3 On-hands Circuit

The drive link where the video of the on-hands circuit with a test run is

https://drive.google.com/file/d/1hCkWGnPIKBkvPuGMqLGqNcHgwN52jl3l/view?usp=drive_link

References

[1] https://wiki.dfrobot.com/Micro_DC_Motor_with_Encoder-SJ01_SKU__FIT0450

[2] https://www.dynapar.com/technology/encoder_basics/motor_encoders/#:~:text=DC%20motor%20encoders%20are%20used,drive%20for%20precise%20speed%20control.

[3] https://www.iqsdirectory.com/articles/electric-motor/dc-motors.html

Appendix

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

//**********************************
//  Qosay Rida - 1211553
//  Mohammed Abed Alkareem - 1210708
//  Ahmad Hamdan - 1210241
//  Mohammad Fareed - 1212387
//**********************************


// Pin definitions
const byte motorPin1 = 6;
const byte motorPin2 = 5;
const byte sensorPin1 = 2;
const byte sensorPin2 = 3;
const byte potPin1 = A0;

const float encoderValue = 6.2;
const int threshold = 512;

// Variables
byte motorDirection1 = 0;
bool motorStatus1 = false;
float targetSpeed1 = 0;
volatile long sensorCount1 = 0;
float rpm1 = 0;
float speedError;
long previousTime1 = 0;
bool motorRunning = true; // Variable to keep track of motor state
byte mode = 1; // 1: MODE1, 2: MODE2

// Function declarations
void sensorInterrupt1();
void calculateRPM();
void readPotentiometers();
void controlMotors();
void updateDisplay();
void handleSerialCommands();
void startMotor();
void stopMotor();
void setDirection(String dir);
void setSpeed(int speed);
```

```
void setMode(byte newMode);

void setup() {


            lcd.init();
            lcd.backlight();

   // Initialize the LCD
   lcd.begin(16, 2);
   lcd.setCursor(0, 0);
   lcd.print("S=");

   lcd.setCursor(8, 0);
lcd.print("RPM=");

   lcd.setCursor(0, 1);
   lcd.print("CW ");
           lcd.setCursor(11, 1);
   lcd.print("MODE1");

   Serial.begin(9600);

   // Configure sensor pins as input
   pinMode(sensorPin1, INPUT_PULLUP);
   pinMode(sensorPin2, INPUT_PULLUP);
  pinMode(potPin1, INPUT);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
   // Attach interrupts for sensor pins
   attachInterrupt(digitalPinToInterrupt(sensorPin1), sensorInterrupt1, CHANGE);
}

void loop() {
   handleSerialCommands();
   if (motorRunning) {
      calculateRPM();
      if (mode == 1) {
         readPotentiometers();
      }
      controlMotors();
   }
}

void sensorInterrupt1() {
```

```
  sensorCount1++;
}

void calculateRPM() {
  unsigned long currentTime1 = millis();
  if ((currentTime1 - previousTime1) >= 100) {
    previousTime1 = currentTime1;

    if (motorDirection1 == 0) {
      rpm1 = sensorCount1 * 10 / encoderValue;
    } else {
      rpm1 = -sensorCount1 * 10 / encoderValue;
    }

  lcd.setCursor(12, 0);
  lcd.print("    ");
  lcd.setCursor(12, 0);
  lcd.print((int)abs(rpm1));

    sensorCount1 = 0;
  }
}

void readPotentiometers() {
  int potValue = analogRead(potPin1);

  if (potValue < threshold) {
    motorDirection1 = 1;
    lcd.setCursor(0, 1);
    lcd.print("CCW");
  } else {
    motorDirection1 = 0;
    lcd.setCursor(0, 1);
    lcd.print("CW ");
  }

        if (abs(potValue - 511) < 11) {
  targetSpeed1 = 0;
        } else if (potValue <= 511) {
  targetSpeed1 = map(potValue, 0, 511, 255, 0);
 } else if (potValue > 511) {
  targetSpeed1 = map(potValue, 512, 1023, 0, 255);
 }
```

```
    lcd.setCursor(2, 0);
    lcd.print("    ");
    lcd.setCursor(2, 0);
    lcd.print((int)abs(targetSpeed1));


}

void controlMotors() {
    if (motorDirection1 == 0) {
        analogWrite(motorPin1, targetSpeed1);
        analogWrite(motorPin2, 0);
    } else {
        analogWrite(motorPin1, 0);
        analogWrite(motorPin2, targetSpeed1);
    }
}

void handleSerialCommands() {
    if (Serial.available() > 0) {
        String command = Serial.readStringUntil('\n');
        command.trim(); // Remove any whitespace characters

        if (command == "START") {
            startMotor();
        } else if (command == "STOP") {
            stopMotor();
        } else if (command.startsWith("SET_DIR")) {
            String dir = command.substring(8);
            setDirection(dir);
        } else if (command.startsWith("SET_SPEED")) {
            int speed = command.substring(10).toInt();
            setSpeed(speed);
        } else if (command.startsWith("SET_MODE")) {
            byte newMode = command.substring(9).toInt();
            setMode(newMode);
        } else {
            Serial.println("Unknown command");
        }
    }
}

void startMotor() {
    motorRunning = true;
    Serial.println("Motor started");
```

```
}

void stopMotor() {
  motorRunning = false;
  analogWrite(motorPin1, 0);
  analogWrite(motorPin2, 0);
  Serial.println("Motor stopped");
}

void setDirection(String dir) {
  if (dir == "CW") {
    motorDirection1 = 0;
    lcd.setCursor(0, 1);
    lcd.print("CW ");
    Serial.println("Direction set to CW");
  } else if (dir == "CCW") {
    motorDirection1 = 1;
    lcd.setCursor(0, 1);
    lcd.print("CCW");
    Serial.println("Direction set to CCW");
  } else {
    Serial.println("Invalid direction");
  }
}

void setSpeed(int speed) {
  if (speed >= 0 && speed <= 255) {
    targetSpeed1 = speed;
     lcd.setCursor(2, 0);
    lcd.print("    ");
    lcd.setCursor(2, 0);
   lcd.print((int)abs(targetSpeed1));
    Serial.print("Speed set to ");
    Serial.println(speed);
  } else {
    Serial.println("Invalid speed");
  }
}

void setMode(byte newMode) {
  if (newMode == 1 || newMode == 2) {
    mode = newMode;
    Serial.print("Mode set to ");
    Serial.println(newMode);
    if (mode == 2) {
```

```
            lcd.setCursor(11, 1);
            lcd.print("MODE2");
        } else {
            lcd.setCursor(11, 1);
            lcd.print("MODE1");
        }
    } else {
        Serial.println("Invalid mode");
    }
}
```