Faculty of Engineering and Technology
Electrical and Computer Engineering Department

Computer Design Laboratory
ENCS4110

Report of Experiment No. 10

DC Motors and PWM

Student's Name: Maha Mali

Student's ID: 1200746

Sec: 1

Instructor: Dr. Abualseoud Hanani

Teacher assistant: Eng. Raha Zabadi

Date: 26/8/2022

Abstract

The main goal of this experiment is to learn how to generate a Pulse Modulation Width (PWM) signal using Arduino. Also, control the speed of a DC motor using Pulse Width Modulation (PWM). Because DC motors need high current, we can't drive the motor from a digital pin (even if the motor runs on 5 volt). Therefore, we need a driving circuit. Also, the ability to change the rotation of a DC motor using an H bridge. The circuits will be generated using the Tinker Cad website

Table of Contents

List Of Figures

Table

1.Theory

1.1 Arduino UNO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing [1].

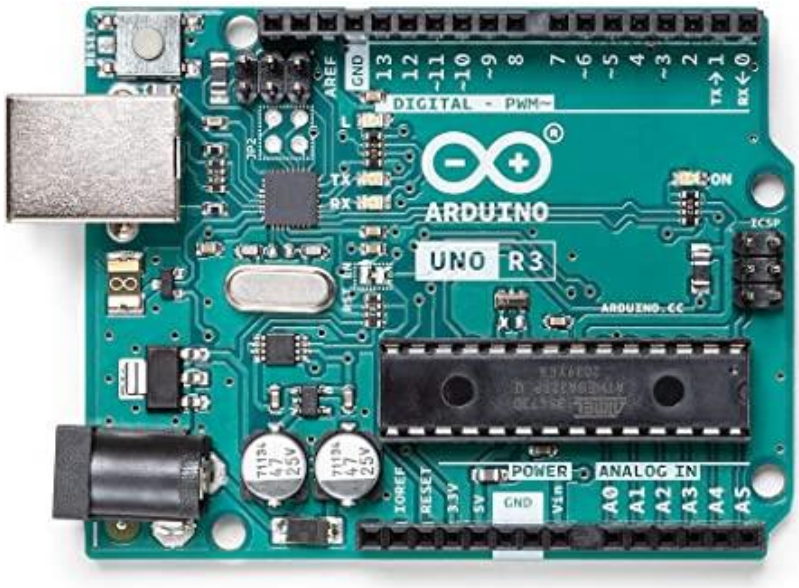


Figure1: Arduino UNO

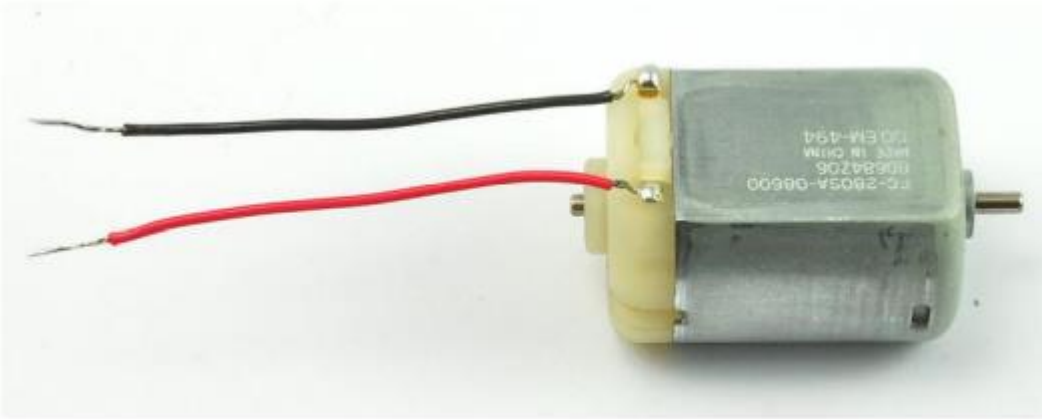
1.2 Driver Circuits

In electronics, a driver is a circuit or component used to control another circuit or component, such as a high-power transistor, liquid crystal display (LCD), stepper motors, and numerous others [2].

They are usually used to regulate current flowing through a circuit or to control other factors such as other components and some other devices in the circuit. The term is often used, for example, for a specialized integrated circuit that controls high-power switches in switched-mode power converters. An amplifier can also be considered a driver for loudspeakers, or a voltage regulator that keeps an attached component operating within a broad range of input voltages [2].

1.2.1 DC motors

A DC motor is any of a class of rotary electrical motors that converts direct current (DC) electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor [3].



Therefore, we need a driving circuit to run the motor. We can use a BJT transistor as a driver for the DC motor. We need a resistor on the base to limit the current from the digital pin. The diode is needed to protect the transistor. Because motors usually draw large current, MOSFET transistors are usually used to drive DC Motors [4].

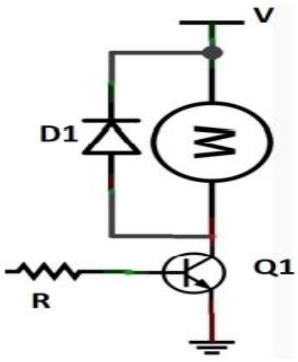


Figure2: Drive a Motor using BJT

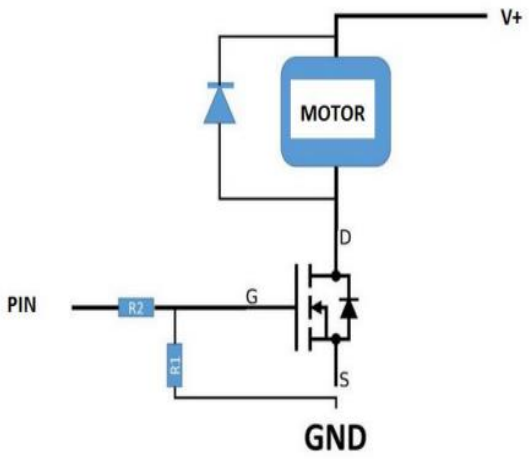


Figure3: Drive a Motor using BJT MOSFET

1.2.3 Pulse Width Modulation (PWM)

Pulse width modulation reduces the average power delivered by an electrical signal by converting the signal into discrete parts. In the PWM technique, the signal's energy is distributed through a series of pulses rather than a continuously varying (analogue) signal [5].

In addition to that, we can pulse output on and off fast such that the connected device sees the result as a lower voltage. This way, we can control the speed of a DC-motor. We define:

Duty cycle= (ON period)/ (ON+OFF periods)

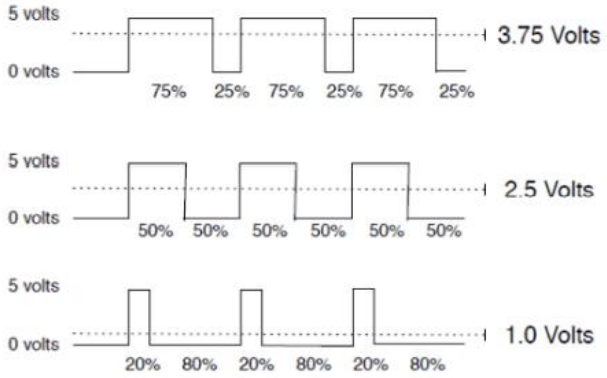


Figure4:PWM Signal

So, for a duty cycle of 50% and source voltage =12 volt, the load will see 6 volts.

In Arduino we can use **analog Write (pin, value)** to generate a PWM signal with duty cycle value/255 where 0≤value≤ 255.

1.2.4 H-Bridge

A H-bridge is an electronic circuit that switches the polarity of a voltage applied to a load. These circuits are often used in robotics and other applications to allow DC motors to run forwards or backwards, also to change the rotation direction of a DC-motor we need to reverse the polarity [6].

This can be done using H-Bridge as shown in Figure 5. If Q1 and Q4 are ON while Q2 and Q3 are OFF, the motor can run clockwise, now of Q2 and Q3 are ON, while Q1 and Q4 are OFF, the motor is going to reverse the direction [4].

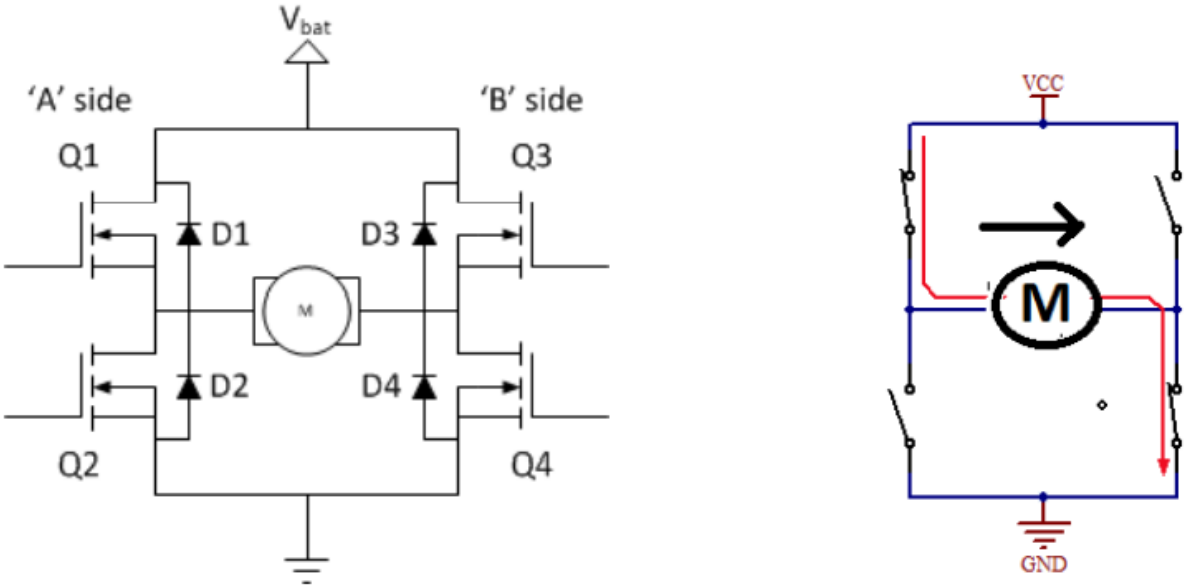


Figure5: H-bridge

2. Procedure and Discussion

2.1 Use a transistor the control the speed of a DC-Motor

2.1.1 Circuit Connection

In this section, we will build a circuit that allows us to control the speed of the DC Motor. The circuit consists of: transistor to drive DC Motors, diode to safeguard the transistor, and the DC Motors that we'll be working with to regulate their speed.

At first, we connected the circuit as shown in figure 6, the positive line is connected to +5V on the breadboard, whereas the ground pin is connected to the ground line. The diode and motor are then wired in parallel, with the positive pin linked to the positive of the voltage, the positive side of the motor connected to the collector leg, the emitter connected to the ground, and the base connected to a resistor connected to Arduino's digital pin 9.

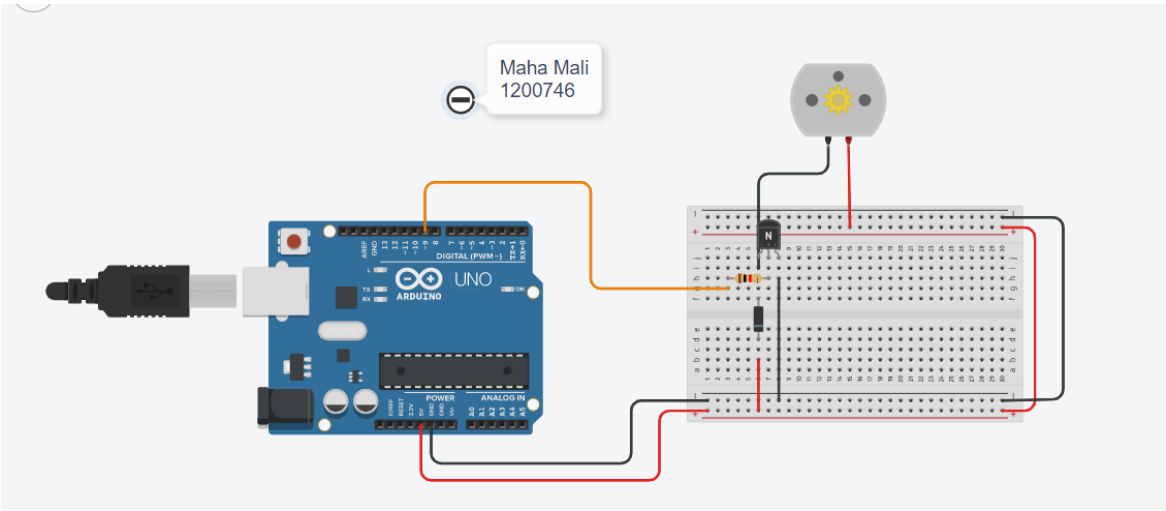


Figure 6: Circuit Connection of Drive a Motor using BJT

2.1.2 Code [1]

First, we make pin 9 as an output pin, and this is done by using **pinMode(9, OUTPUT)** , then we turn on the motor at maximum speed and this is done by using the value **HIGH with digitalWrite(9, HIGH)** to start the DC Motor. And on the motor, we can see that the motor went to the greatest value or the maximum speed 255, which is 9922 rpm.

[1] Written code can be found in the appendix.

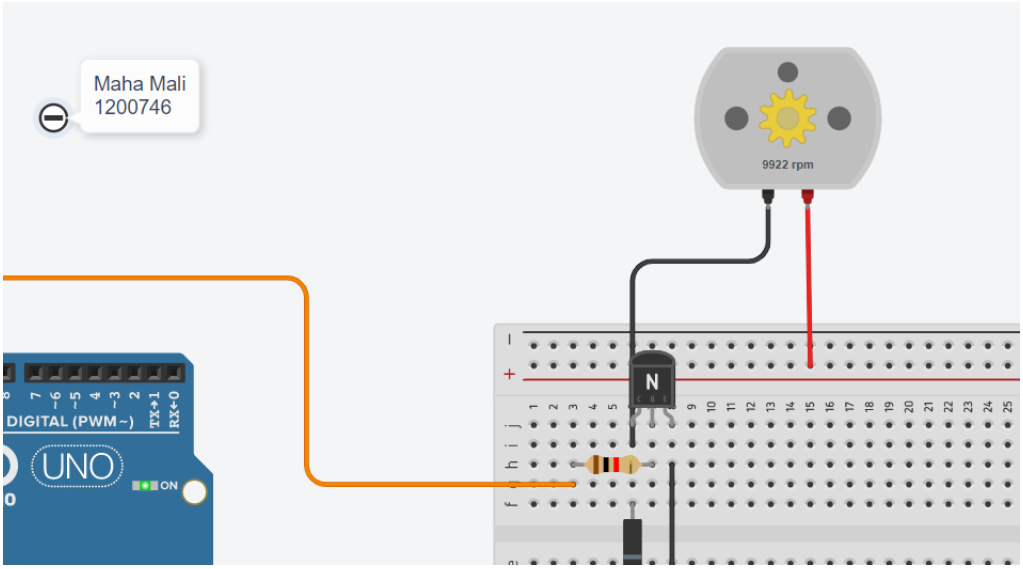


Figure 7: Motor on Maximum Speed

When changing the value from HIGH to LOW, the motor speed will be 0 rpm. But to change the speed and set other values to it we have to use another function which is **analogWrite(pin, Value)**. The value must be between 0 and 255.

Function	Speed(rpm)	Duty Cycle
digitalWrite(9, HIGH)	9922	100%
digitalWrite(9, HIGH)	0	0%
analogWrite(pin, Value)	≈7800	78%
analogWrite(pin, Value)	≈2500	25%

Table 1:Duty Cycle for Different Value

The link of the circuit:
https://www.tinkercad.com/things/bzw77AmfUro-tremendous-snicket/editel?sharecode=HiSBmTZ9LTKe5gk-ik9ht01pzNr5_E1Ov7EyK-X_RRc

2.2. Task1

The task: using Serial monitor, write a code to read a number from keyboard and change the speed of the motor. You have to check if the number between 0 and 255. Print an error message if the number outside this range. Show the result to the instructor.

2.2.1 Circuit Connection

The circuit connection is the same as the previous example as shown in figure 6.

2.2.1 code for task 1[2]

First, we can read the Motor speed from the user by using **int speed = Serial.parseInt()**. In addition to that , pin 9 is defined as an output by using **pinMode(9, OUTPUT)**, and the speed value must be between 0 and 255 or an error message will appear to the screen, because the output is 8 bits, so the value must be between 0 and 255. The speed is set using the analogWrite function.

The link of the circuit:
https://www.tinkercad.com/things/bzw77AmfUro-use-a-transistor-the-control-the-speed-of-a-dc-motor/editel?sharecode=HiSBmTZ9LTKe5gk-ik9ht01pzNr5_E1Ov7EyK-X_RRc

[2] Written code can be found in the appendix

2.3 Use H-Bridge to control the speed and direction of a DC-Motor

In this part we will use L298 Motor Driver Module to control the speed and direction of a DC-Motor. The H-Bridge, which allows more than one motor to be connected or controlled in different directions.

2.3.1 Circuit Connection

The Microcontroller ground pin was connected to the breadboard's negative line, and the 5-volt port is connected to the H-Bridge's positive line. The power pins of both sides are connected to the 5-volt H-Bridge's pin, and each side has two ground pins. The ground pin on the Arduino is connected to the two pins on the selected side, the order input pin on the H-less Bridge is connected to the Arduino's less order digital pin, and the opposite for the highest order. The less ordered pin on the DC-Motor is connected to the negative side, the highest order output pin is connected to the positive side, and the enable pin on the H-Bridge is connected to the positive side. As shown on figure

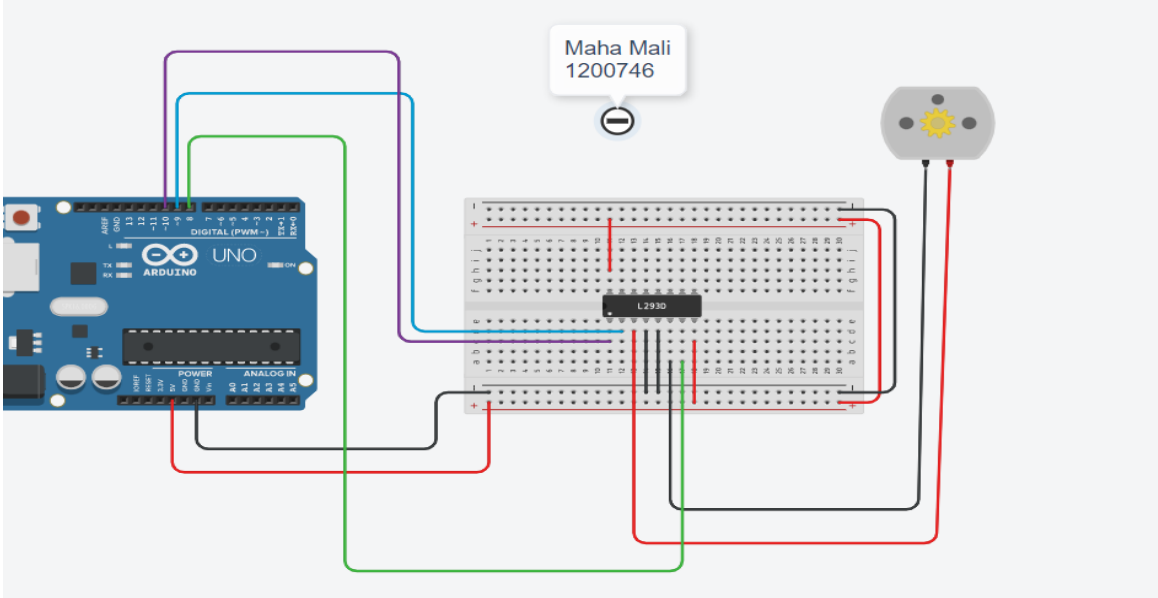


Figure 8: Circuit Connection of Drive a Motor using H-Bridge's

2.3.2 The code [3]

The three ports of Arduino (pin:8,0,10) are set as outputs **pinMode(pin, OUTPUT)**, and the motor will rotate clockwise and give a positive speed value if pin 8 is LOW and pin 9 is HIGH, and counterclockwise if the pins are set up the other way. The **analogWrite** function is used to set the speed,

[3] Written code can be found in the appendix

and it receives a digital pin number linked to the H-Bridge enables' pin and a speed value will provide the instruction to rotate the motor at the desired speed.

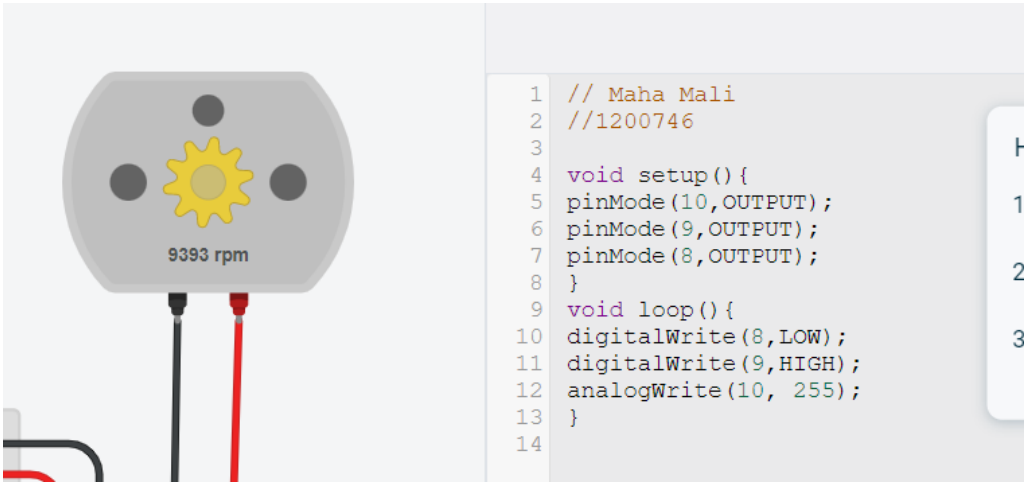


Figure9: DC Motor Rotates clockwise

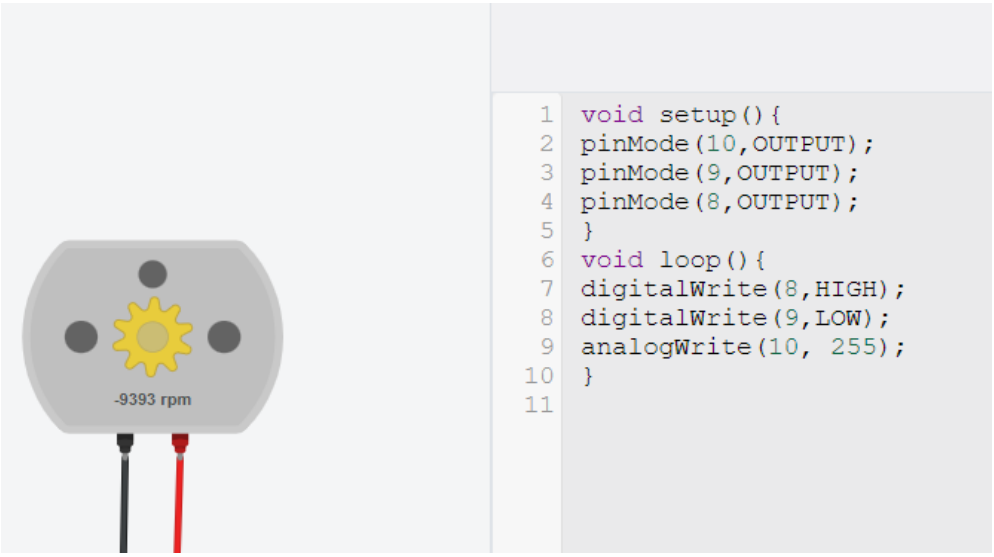


Figure10: DC Motor Rotates Counterclockwise

The link of the circuit:
https://www.tinkercad.com/things/2DKoBtwZ32n-funky-stantia/editel?sharecode=wQd4foGcoR3tmbxMbWw-3hFrCAX2nHIL0TRVEpHlSKk

2.4. Task2

The task: write code such that you can control the direction and speed of a DC-Motor from the keyboard. Show the result to the instructor

2.4.1 Circuit Connection

The circuit connection is the same as the previous example as shown in figure 8.

2.4.2 the code for task 2[4]

The pinMode(pin, OUTPUT) was used to make the three ports output. Then, to read a character, Serial.read() was used, followed by Serial.parseInt() to read the speed. The motor's speed and direction are regulated based on the input according to the caricature and the speed.

The link of the circuit:
https://www.tinkercad.com/things/2DKoBtwZ32n-funky-stantia/editel?sharecode=wQd4foGcoR3tmbxMbWw-3hFrCAX2nHIL0TRVEpHlSKk

[4] Written code can be found in the appendix

3. Conclusion

In this experiment we got to know the Arduino, DC-Motor, and H-Bridge, and how they work, also we become able to write the code for it, and connection the circuit. We discovered that Arduino coding is a simple way to control the speed and path of a DC-Motor, that the motor's rotation direction can be varied depending on the ports of an H Bridge, that such a transistor or even an H-Bridge can control the DC-motor's speed, and that a DC-Motor can generate a lot of current, necessitating a running circuit.

4.Feedback

The experiment was good and interesting, also the time was enough, so we finished the experiment early.

5. References

[1] https://www.arduino.cc/en/Guide/Introduction . Accessed on 24-08-2022 at 6:22PM.

[2] https://en.wikipedia.org/wiki/Driver_circuit . Accessed on 24-08-2022 at 6:30PM.

[3] https://en.wikipedia.org/wiki/DC_motor . Accessed on 24-08-2022 at 6:38PM.

[4] *Exp2_DC-Motor+and+PWM.pdf . Accessed on 24-08-2022 at 6:44PM.

[5] https://byjus.com/physics/pulse-width-modulation/ . Accessed on 24-08-2022 at 7:04PM.

[6] https://en.wikipedia.org/wiki/H-bridge . Accessed on 24-08-2022 at 7:11PM.

6. Appendix

6.1 code of Using a transistor the control the speed of a DC-Motor

```
// Maha Mali

//1200746

void setup(){

pinMode(9,OUTPUT);

}

void loop(){

digitalWrite(9,HIGH);

}
```

6.2 Code for Task 1

```
// Maha Mali

//1200746


void setup(){

pinMode(9,OUTPUT);

Serial.begin(9600);

}

void loop(){

Serial.print("enter value for speed between 0to 255\n");

int speed = Serial.parseInt();

Serial.println(speed);

if (speed < 0 || speed > 255){

Serial.print("erorr\n");

}

else{
```

```
analogWrite(9,speed);
}
delay(1000);
}
```

6.3 Code of Using an H-bridge to control the speed and direction of a DC-Motor

```
// Maha Mali
//1200746
//clock wise
void setup(){
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
pinMode(8,OUTPUT);
}
void loop(){
digitalWrite(8,LOW);
digitalWrite(9,HIGH);
analogWrite(10, 255);
}
```

```
// Maha Mali
//1200746
```

```
//counter clockwise

void setup(){

pinMode(10,OUTPUT);

pinMode(9,OUTPUT);

pinMode(8,OUTPUT);

}

void loop(){

digitalWrite(8,LOW);

digitalWrite(9,HIGH);

analogWrite(10, 255);

}
```

6.4 Code for Task 2

```
void setup(){
 pinMode(10,OUTPUT);
 pinMode(9,OUTPUT);
 pinMode(8,OUTPUT);
 Serial.begin(9600);
}
void loop(){
 Serial.println("enter a to direction for Clockwise");
 Serial.println("enter B to direction Counterclockwise:");
 delay(5000);
 if(Serial.available()>0){
 char direction = Serial.read();
 Serial.println(direction);
 if(direction == 'F'){
 digitalWrite(8,LOW);
 digitalWrite(9,HIGH);
}else if(direction == 'B'){
 digitalWrite(8,HIGH);
 digitalWrite(9,LOW);
 }
 Serial.println("Enter  speed between 0 to 255:");
```

```
  delay(5000);
  int speed = Serial.parseInt();
  Serial.println(speed);
  if(speed >= 0 && speed <= 255){
  analogWrite(10, speed);
  }
  delay(5000);
  speed = 0;
  }
}
```