|  |  |
| --- | --- |

Final Project

Elyse Addison

Praprara Owodeha- Ashaka

Phillip Teye Aborsi

Engineering Department, Ashesi University

[24-25\_SEM1\_EE320\_A] – Power Electronics

Mr. Bright Tetteh

18th April, 2025.

**1. Introduction**

A DC motor is an electrical machine which, when provided with direct electrical energy, converts it into mechanical energy. It is based on electromagnetic induction, where a conductor carrying current placed in a magnetic field experiences force to rotate. DC motors are used in robotics, electric vehicles, and some industrial machinery as well as household devices. A DC motor can be used at places where control is required. DC motors are often used in trolleys, electric trains, production systems, elevators because they are used at places where speed control is required.

**2. Project Objective**

The objective of this project is to design, build and test a full bridge DC-DC converter motor driver to control the direction and speed of a DC motor. The objective is also to control the speed of the motor using microcontrollers for Pulse Width Modulation (PWM).

**2.1 Speed Control**

DC motors run at different speeds depending on the average voltage applied to them. However, instead of directly varying the voltage (which is inefficient and can overheat the motor), we use Pulse Width Modulation(PWM), a high technique where the motor is turned ON and OFF rapidly at a higher frequency. Microcontrollers are used for pulse width modulation (PWM) to control the speed. The PWM signal is sent to the gate of the MOSFET in the motor driver circuit. The MOSFERT acts as a switch, turning the motor power ON and OFF according to the PWM pulses.

**2.2 Directional Control**

Bidirectional control means being able to run the motor both forward and in reverse. For DC motors, the direction of rotation depends on the polarity of the voltage applied to its terminals. We used MOSFETS and L239D IC for bidirectional control of the motor.

**3.0 Design requirements**

The project's goal is to control a motor's speed and direction. The motor is the only load in this circuit so component selection should depend on the voltage rating and the current it will draw.

The voltage rating is 3-6 V. The current rating is ≤ 150 mA at 3 V and ≤ 200 mA at 6 V. Assuming the worst-case scenario, we need components that can handle **200mA.**

**CRITERIA FOR SELECTING TRANSISTOR**

Transistors are needed to act as switches to enable directional motor control. If a pair of transistors is switched on, the motor turns in one direction, and if the other pair of transistors is switched on, the motor turns in the other direction. The transistor should be able to handle the maximum current that the load can draw (ID for MOSFET and IC for BJT)

| MOSFET | VDS | ID | RDS | VGS |
| --- | --- | --- | --- | --- |
| IRF540N (N channel) | 100 v | 23 A | 0.04 Ω | 20 V |
| IRF9540 (P channel) | -100 V | -13 A | 0.2 OHMS | 20 |

**The N channel one is better than the P channel one. Higher ID, lower RDS .**

From the table, it is observed that the IRF9540(P channel) values are negative and the IRF540N values are positive. This is because for the N channel, the current flows from the drain to the source. To turn on the N channel MOSFET, the gate must be more positive than the source. For the P channel MOSFET, the current flows from the source to the gate. To turn it on, the gate must be more negative than the source. From the table, it is observed that the N channel MOSFET (IRF540N) has a lower resistance (0.04 ohm) compared to the P channel MOSFET which has a relatively higher resistance (0.2 ohms). The smaller resistance means that it would allow an appreciable amount of current to flow through it. Considering this and many more, we decided to pick the N channel MOSFET (IRF540N) since this would allow enough current to flow to run the motor. Also, a smaller resistance value means small power losses and higher efficiency and hence our choice of the N channel MOSFET.

We also considered using BJTs because of the initial concern that we would need a gate driver and so we considered the BJTs in the below circuit. However, with our motor being powerable by the 5V, we saw that we could go with the MOSFETs, and since MOSFETs switch faster and are typically conventionally used for switching purposes.

| BJT | VCE(max) (or VCEO) | Ic | Max VCE(SAT) | Minimum β |
| --- | --- | --- | --- | --- |
| 2N3904 | 40 | 200 mA | 0.2 | 30 |
| TIP31C | 100 | 3A | 1.2 | 10 |
| BC547B | 45 | 100mA | 0.60 | 200 |

## 4.0 **Simulations**

In simulating the circuit for both objectives we used Proteus and LTSpice. We used the Simulations to see how efficiently our motor will move. From the start we considered using the TIP31c but from our simulations it couldn’t move the motor at all with the other components we were using. We did power calculations and some deep analysis to find out why it was not working.

**Power Losses Calculations**

IRF540N MOSFET.

Power loss = current through load x RDs

Power losses =

Power losses for TIP 31c =

From the Simulations the motor didn’t move when we used the TIP31C transistor and this was mainly due to some reasons such as :

* The voltage drop across the BJT was too high to drive the motor. The voltage of 1.2 V was too much to drive the 3-6 v Arduino motor that was used. This can lead to a reduction in the voltage that will be needed to run the motor.
* The MOSFET, however, has a low drain source resistance of 0.04 ohms which indicates that the voltage drop is very small and that the motor will get the necessary voltage to run.
* The IRF540n can handle higher current with its maximum being 23 A unlike the TIP 31c
* From analysis and research MOSFETS have high switching speed which makes them efficient when going in forward and reverse directions because of their lower on resistance. BJT’s are however slow and cannot handle the quick switching.
* BJT’s also draw more current than the MOSFETS because they require a bse current to control the collector current whilst MOSFETS are solely a voltage controlled device because they require minimal gate current.

Diode Selection

| DIODE | VRMM | IF or just I | IFSM | Trr | VF |
| --- | --- | --- | --- | --- | --- |
| 1N4007 | 1000 | 1.0 | 30 A |  | 1.1 V |
| FR207 | 1000 | 2.0 | 60 | 500 ns | 1.2 V |

* The FR207 diode is a suitable choice for motor control due to its fast recovery time and low forward voltage drop, making it ideal for high-speed switching and freewheeling applications. In motor control, these features are important for protecting switching transistors from voltage spikes and improving efficiency
* The FR207 diode is designed for fast recovery, unlike the 1N4007, which is too slow for switching applications.
* It can also handle more current than the 1N4007

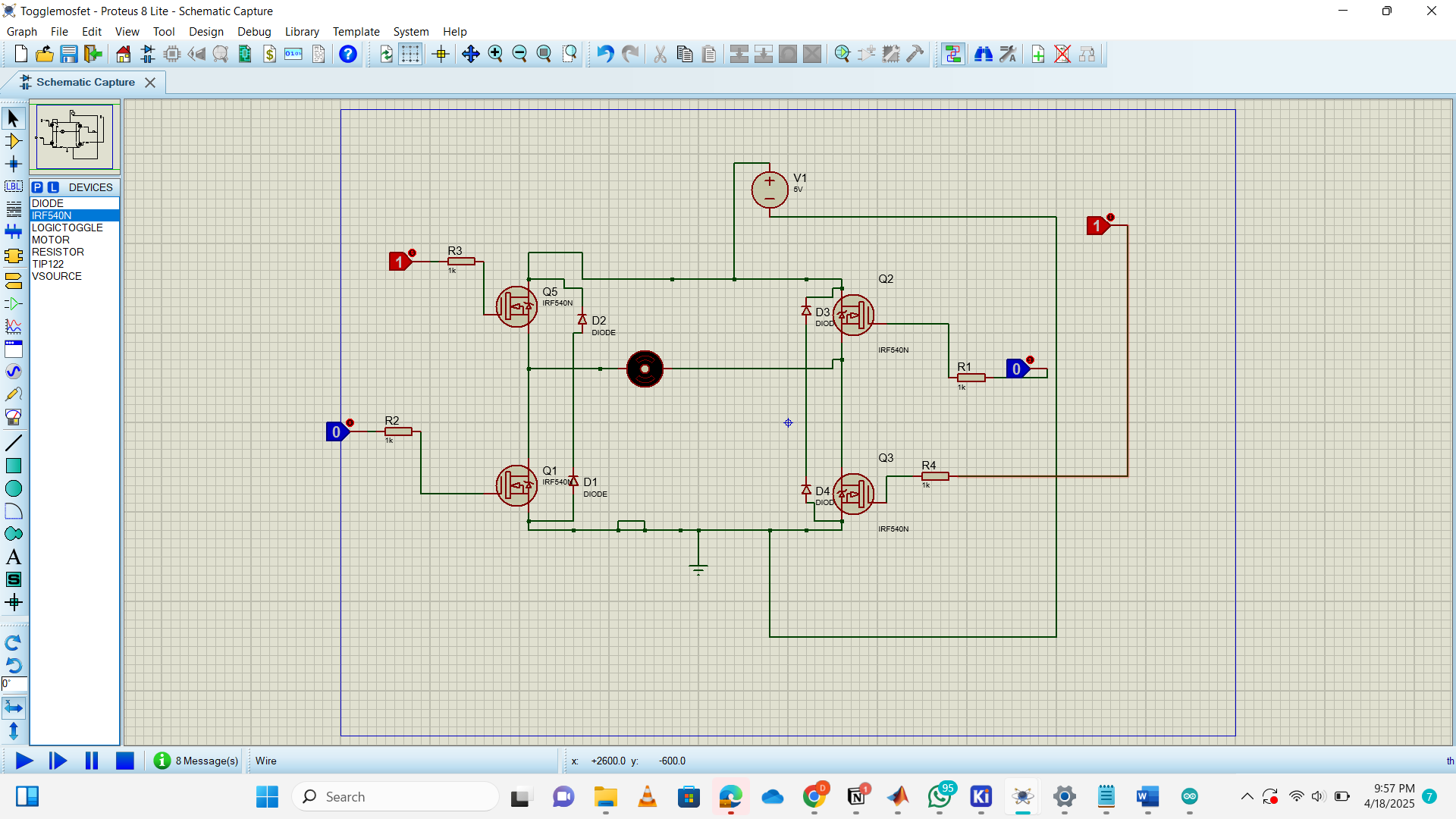


Fig 1 : Simulation with 1RF540n MOSFET

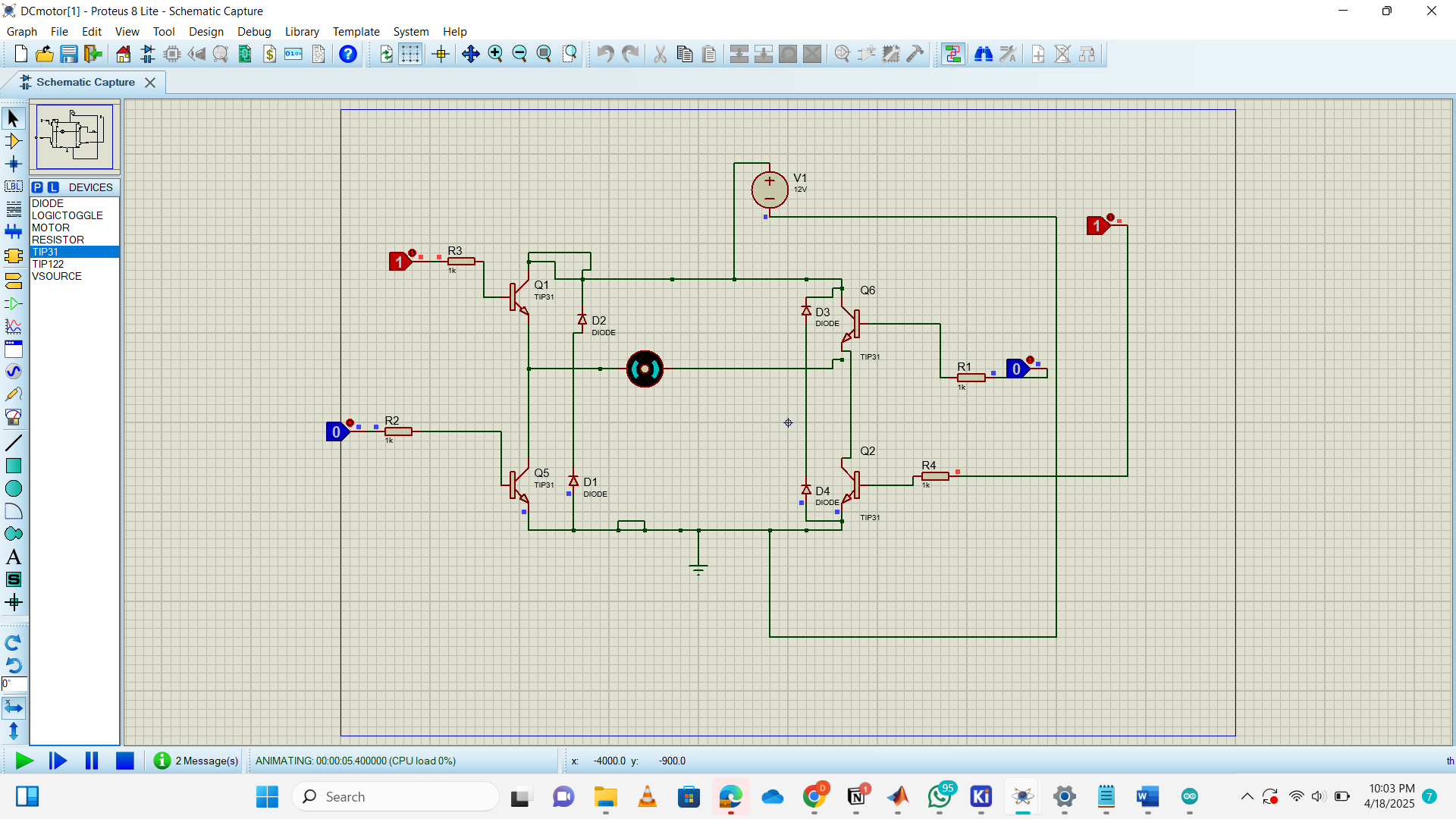
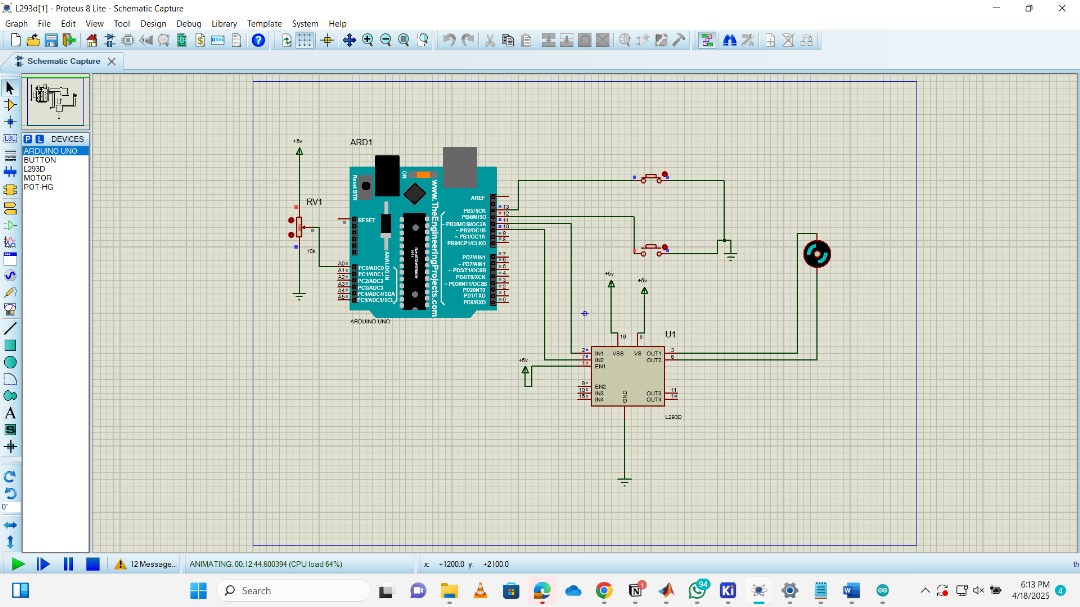


Fig 2: Simulation with TIP 31c Transistor

**Simulations with L293D**



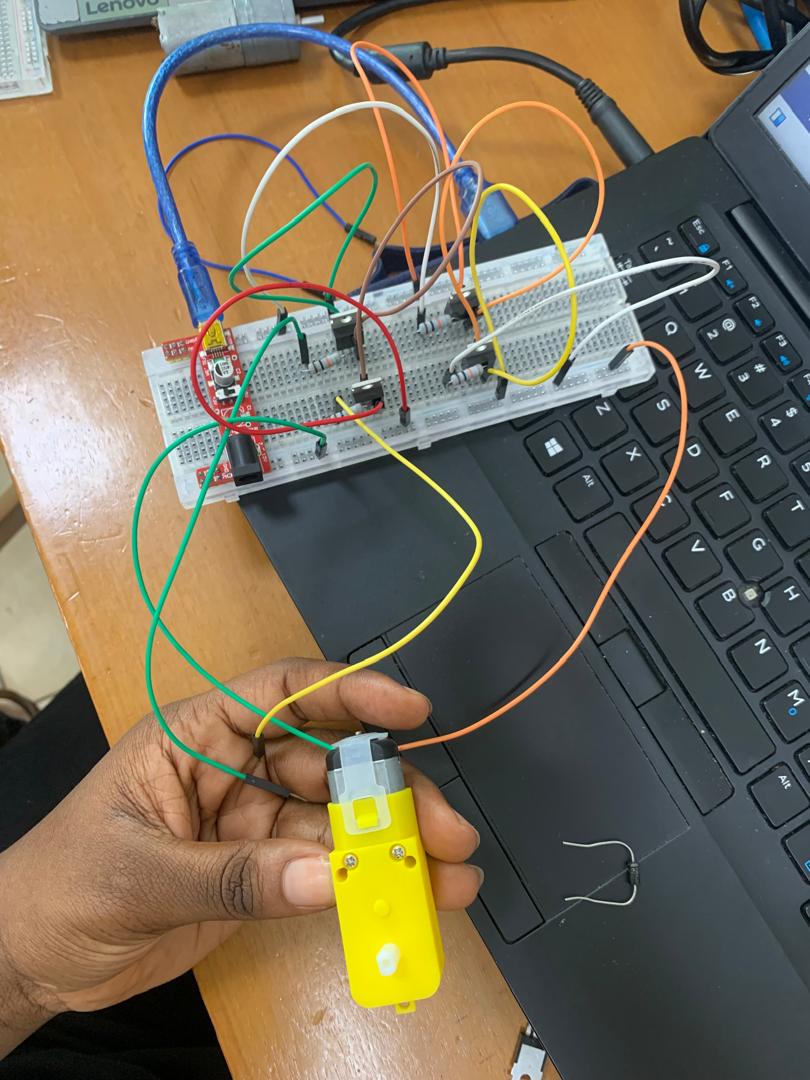
**Fig 3**: Simulations with L293d

We used the L2493D IC to drive the motor clockwise and anticlockwise using switches.

**5.0 Prototype Build**

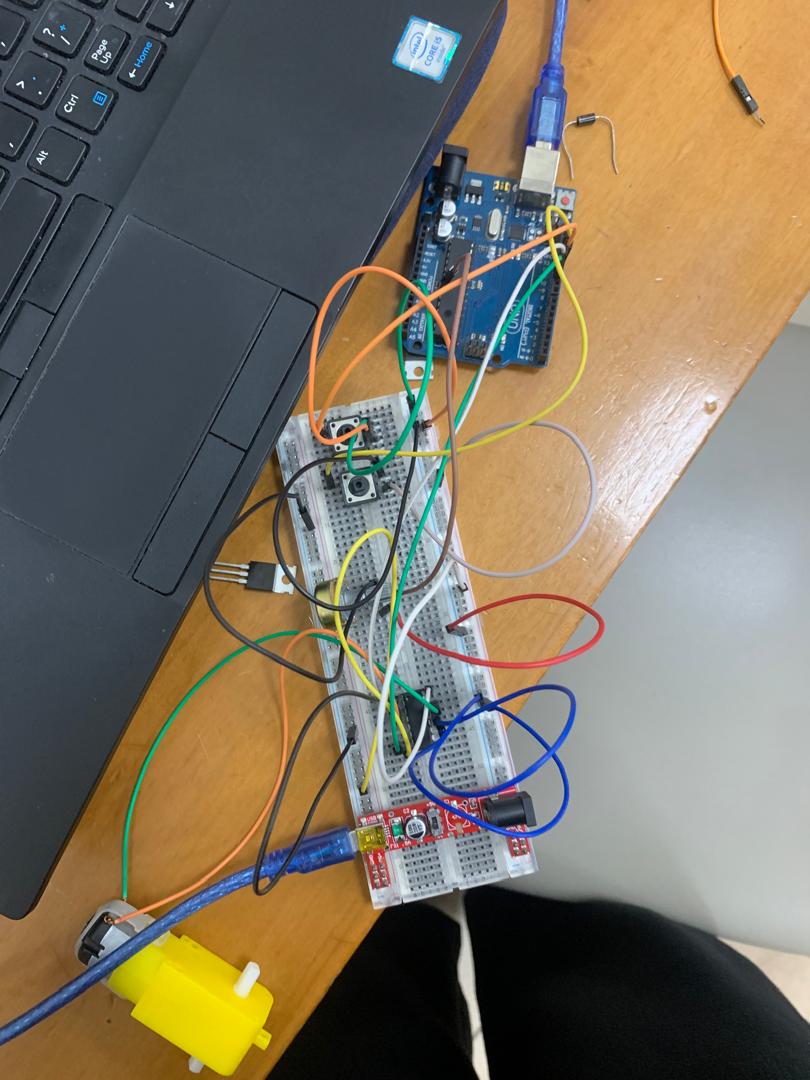
**H bridge with MOSFET**

In building this circuit we decided to use high and low signals to control the direction of the motor. There were 4 MOSFETs with the 2 high sides connected to positive voltage and the 2 low sides connected to negative voltage. To the input of Q1 and Q4 we applied a high and to Q3 and Q2 we applied a low to the input of the transistor for clockwise and vice versa for anticlockwise motion. We used the flyback diode (FR207) to protect the MOSFETs from voltage spikes and the effects of back emf. With the help of the Arduino speed control with PWM was achieved.



**Motor driver circuit with L293D**

We connected the motor to two outputs on the motor driver for speed and bidirectional control. Digital pins of the arduino were connected to the input of the L239D motor driver for PWM signals. This helped to control the motor in the clockwise and anticlockwise direction. We connected two push buttons to the circuit (one for clockwise direction and the other for anticlockwise direction). After supplying our 5V to the circuit, the motor starts running in the clockwise direction and we press one push button to turn the motor in the other direction. To change direction back to clockwise, we press the other push button to turn the motor. This is done consecutively to switch between clockwise and anticlockwise directions.



**Testing and Results**

**Bidirectional Control**

In testing for the bidirectional control of the circuit for the MOSFET, we connected the input signals to vcc and ground to see if there will be a change in the direction of the motor. We tested when the input signals to Q1 and Q4 were connected to vcc and the others to ground and the input signals to Q2 and Q3 were connected to V+cc and the others to ground. We faced some issues where the directions were not right because of incorrect components used.

Changing the transistor from a BJT to a MOSFET helped tremendously as we saw the motor moving. We used toggles, that is,, using vcc and ground as input signals and PWM signals. In testing for the L293d circuit, we used the Proteus Simulation to test the circuit and code, which was to control the speed in the clockwise and counterclockwise directions. We used switches as a way to control the clockwise and anticlockwise direction as a check to see if everything was happening as it should.

**Speed Control**

Duty Cycle = %

Duty Cycle = % = 100%

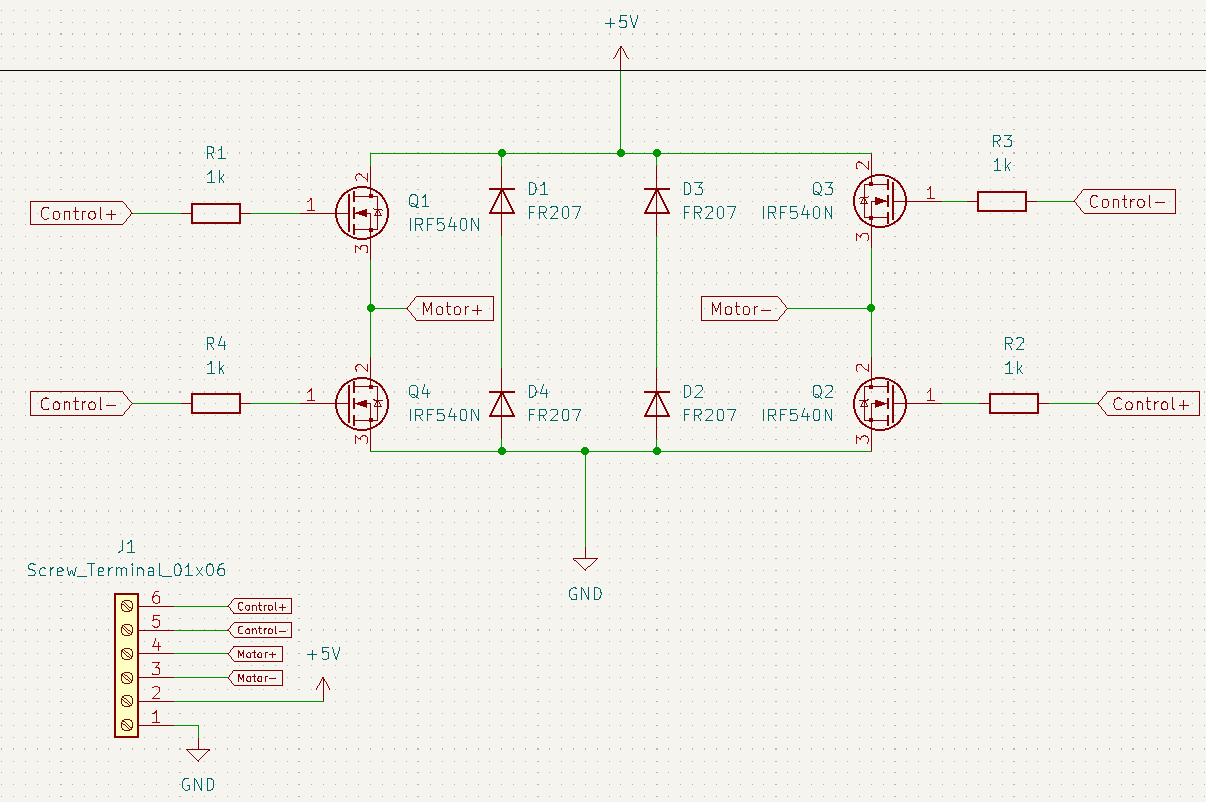
Duty Cycle = % = 0%

Duty Cycle = % = 21.57%

In the speed control of the motor, we changed the duty cycles until we got the maximum speed which was supposed to be 255. We used this speed control for both the H bridge and the L293D circuit.

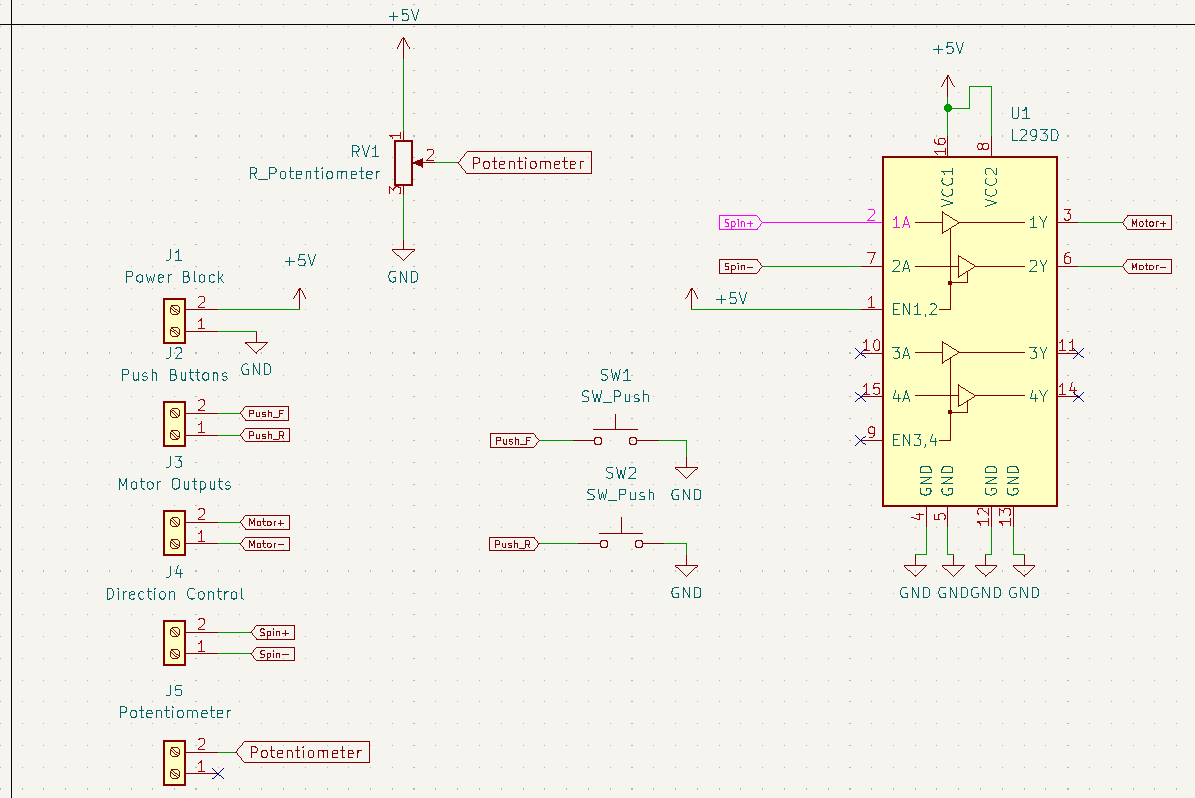
**PCB DESIGN**

Using KICAD, we designed the schematic for the two PCBs based on datasheet suggestion and control requirements, selecting adequate footprints based on the components parts we have physically available. The 3D model of the potentiometer in the motor driver controlled PCB was not available in the KICAD software, thus only the footprint is visible.



**FIG: SCHEMATIC OF THE TRANSISTOR CONTROL CIRCUIT**

The 1k resistor connected to the gate limits the current through it.

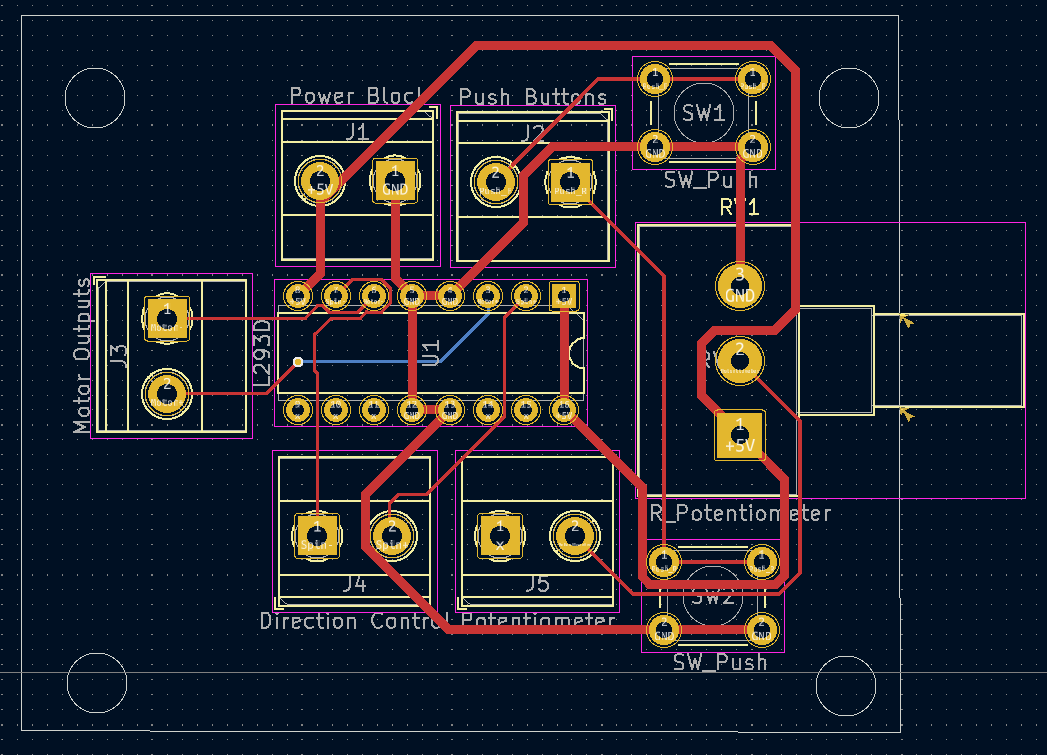


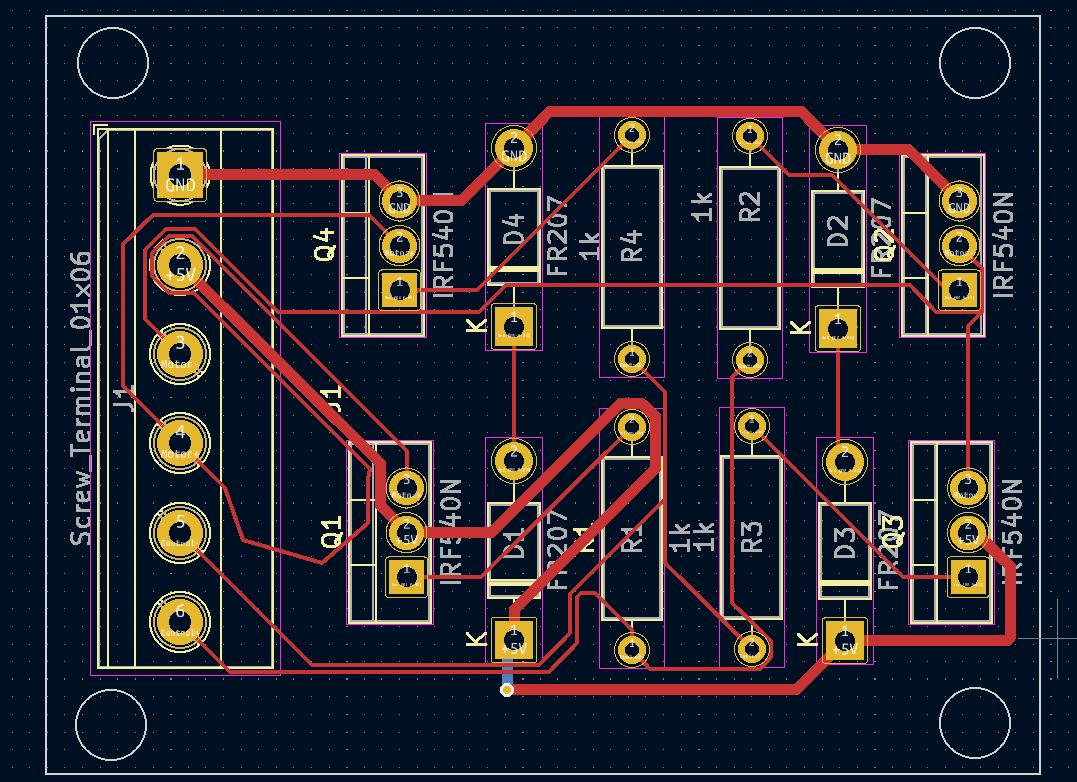
**Figure: SCHEMATIC OF the motor driver controlled circuit**

The potentiometer is

The push buttons receive signals from the microcontroller to control the direction of rotation. PUSH\_F when pressed makes the motor rotate clockwise and PUSH\_R makes it rotate counterclockwise

SPIN + and spin - take signals from the microcontroller (arduino Uno) to dictate the output to the motor. From the datasheet, VCC1 is to be connected to a voltage up to 36 V. However, since our motor is rated 3-6V, we could use the 5V used to power the driver, also as our motor input voltage.

**FIG: Motor driver PCB layout**

****

**FIG: Transistor switching controlled circuit**

**ROUTING**

For routing, the default track width was 0.25 mm. For the power lines (VCC and GND), we had to use a wider track width (0.6 mm) to safely handle higher currents by reducing resistance, minimizing voltage drop, and preventing excessive heat buildup. Using net labels we connected the. Since there was no sizing limitation on the PCB, space was efficiently utilised, posing mo problems to complete manual routing. The 3D viewer was a guide to ensure that essential components like the terminal blocks were correctly positioned.

**5.0 Appendix**

const int clockwise = 13;

const int anticlockwise = 12;

const int pin1 = 11;

const int pin2 = 10;

int clockwisestate = 0;

int anticlockwisestate = 0;

int speed = 0;

void setup() {

pinMode(clockwise, INPUT\_PULLUP);

pinMode(anticlockwise, INPUT\_PULLUP);

pinMode(pin1, OUTPUT);

pinMode(pin2, OUTPUT);

Serial.begin(9600);

}

void loop() {

clockwisestate = digitalRead(clockwise);

anticlockwisestate = digitalRead(anticlockwise);

if (clockwisestate == LOW) {

digitalWrite(pin2, LOW);

Serial.println("Direction: FORWARD");

for (speed = 0; speed <= 255; speed += 5) {

analogWrite(pin1, speed);

Serial.print("Speed: ");

Serial.println(speed);

delay(30);

if (digitalRead(clockwise) == HIGH) break;

}

while (digitalRead(clockwise) == LOW) {

analogWrite(pin1, 255);

Serial.print("Speed: ");

Serial.println(255);

delay(100);

}

digitalWrite(pin1, LOW);

Serial.println("Motor Stopped");

}

else if (anticlockwisestate == LOW) {

digitalWrite(pin1, LOW);

Serial.println("Direction: REVERSE");

for (speed = 0; speed <= 255; speed += 5) {

analogWrite(pin2, speed);

Serial.print("Speed: ");

Serial.println(speed);

delay(30);

if (digitalRead(anticlockwise) == HIGH) break;

}

while (digitalRead(anticlockwise) == LOW) {

analogWrite(pin2, 255);

Serial.print("Speed: ");

Serial.println(255);

delay(100);

}

digitalWrite(pin2, LOW);

Serial.println("Motor Stopped");

}

else {

digitalWrite(pin1, LOW);

digitalWrite(pin2, LOW);

}

}

