**COMSATS University**

**Islamabad**



**Lab Report # 05**

**Real Time Embedded Systems**

**(EEE-446)**

|  |
| --- |
| **Interfacing and Controlling DC Motor and Stepper Motor with Arduino** |

**Submitted By:**

**Arwa Aamir (FA16-EEE-002)**

**Submitted To:**

**Dr. Ahsen Malik**

**Lab # 05**

**Interfacing and Controlling DC Motor and Stepper Motor with Arduino**

## Objectives

* Understand the basics about interfacing of above motors
* Implementation and circuitry to control above motors

## Tools

* Arduino
* Proteus ISIS

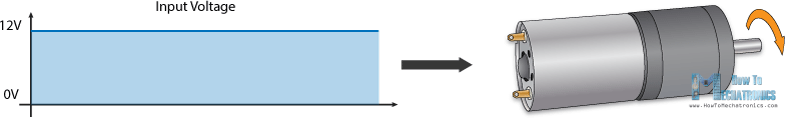
## Pre Lab

Please read the theoretical background of DC Motor and Stepper Motor.

#### DC Motor

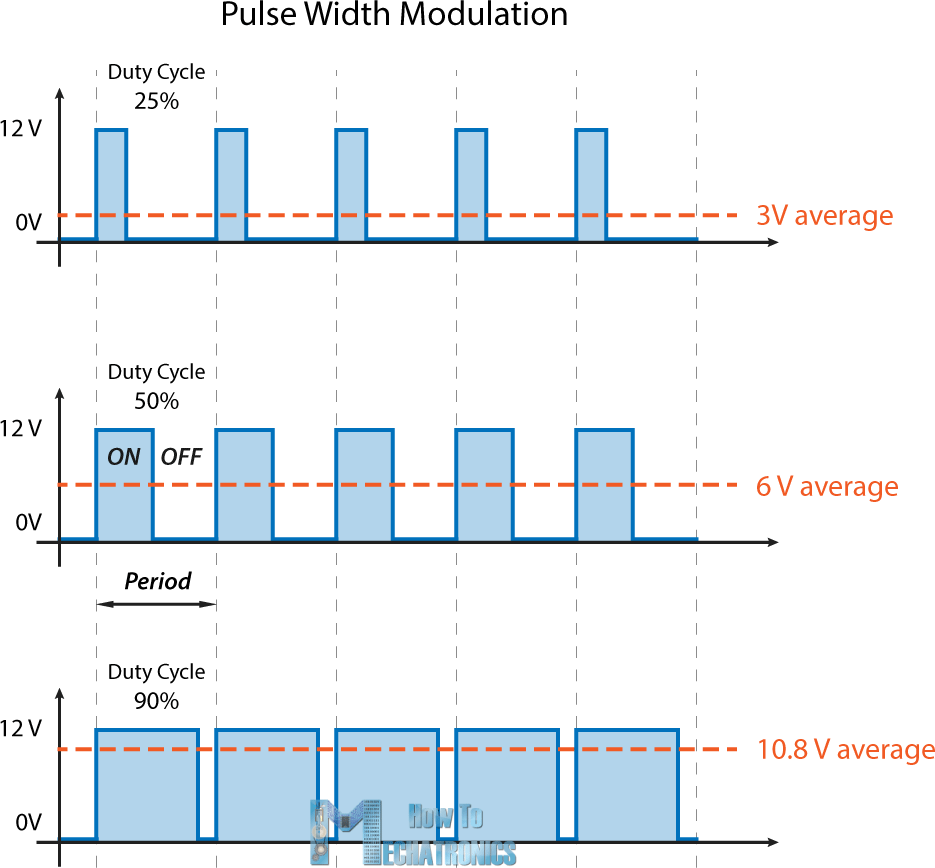
We well take a look at some basic techniques for controlling DC motors through which we will learn how to control DC motors using the L298N driver and the Arduino board.

We can control the speed of the DC motor by simply controlling the input voltage to the motor and the most common method of doing that is by using PWM signal.

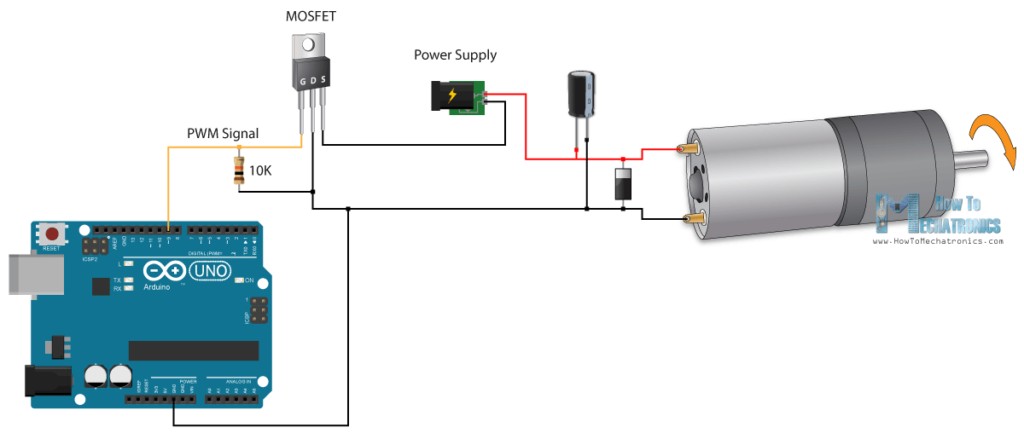


**PWM DC Motor Control**

PWM, or pulse width modulation is a technique which allows us to adjust the average value of the voltage that’s going to the electronic device by turning on and off the power at a fast rate. The average voltage depends on the duty cycle, or the amount of time the signal is ON versus the amount of time the signal is OFF in a single period of time.



So depending on the size of the motor, we can simply connect an Arduino PWM output to the base of transistor or the gate of a MOSFET and control the speed of the motor by controlling the PWM output. The low power Arduino PWM signal switches on and off the gate at the MOSFET through which the high power motor is driven.



### H-Bridge DC Motor Control

On the other hand, for controlling the rotation direction, we just need to inverse the direction of the current flow through the motor, and the most common method of doing that is by using an H- Bridge. An H-Bridge circuit contains four switching elements, transistors or MOSFETs, with the motor at the center forming an H-like configuration. By activating two particular switches at the same time we can change the direction of the current flow, thus change the rotation direction of the motor.

**In-Lab Task 1**

**Connect a button and variable resistor to Arduino, when button is pressed, motor moves forward and when button is not pressed, motor moves reverse. And when variable resistor is at max value, motor has maximum speed and when min value motor has minimum speed.**

|  |
| --- |
| **ARDUINO IDE CODE** |
| void setup() {  // put your setup code here, to run once:  pinMode(A0,INPUT);  pinMode(3,OUTPUT);  pinMode(4,OUTPUT);  pinMode(5,OUTPUT);  pinMode(2,INPUT\_PULLUP);  digitalWrite(2,HIGH);  Serial.begin(9600);  }  void loop() {  int a=analogRead(A0);  unsigned int per=map(a,0,1023,0,255);  Serial.println(per);  analogWrite(3,per);  int flag=digitalRead(2);  Serial.println(flag);  if(flag==LOW)  {  digitalWrite(5,HIGH);  digitalWrite(4,LOW);  }  else  {  digitalWrite(5,LOW);  digitalWrite(4,HIGH);  }  delay(500);  } |

**PROTEUS IMPLEMENTAION:**

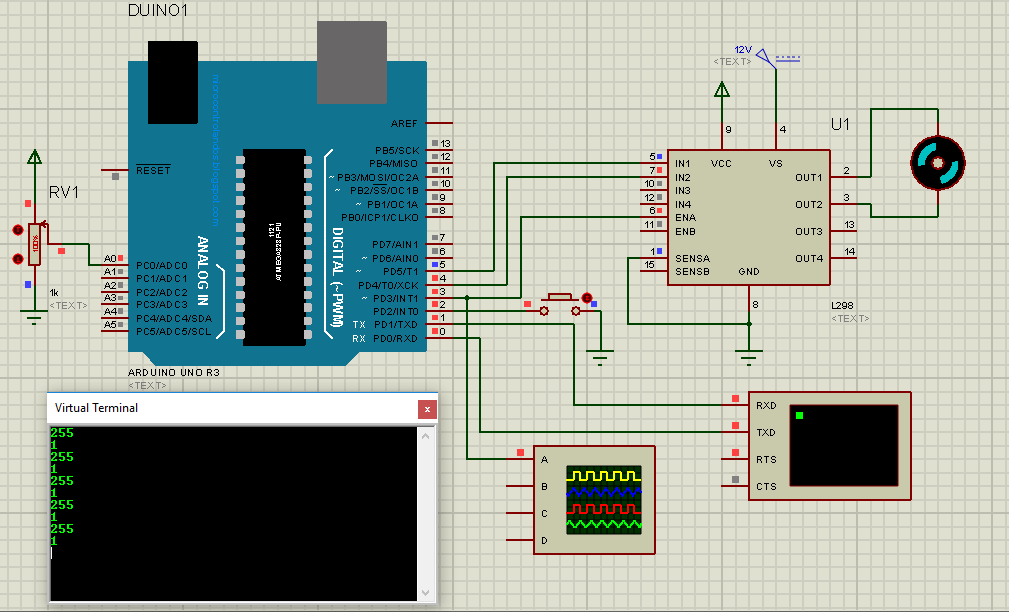


Figure : Button Not Pressed, Anticlockwise Motion, Maximum Speed

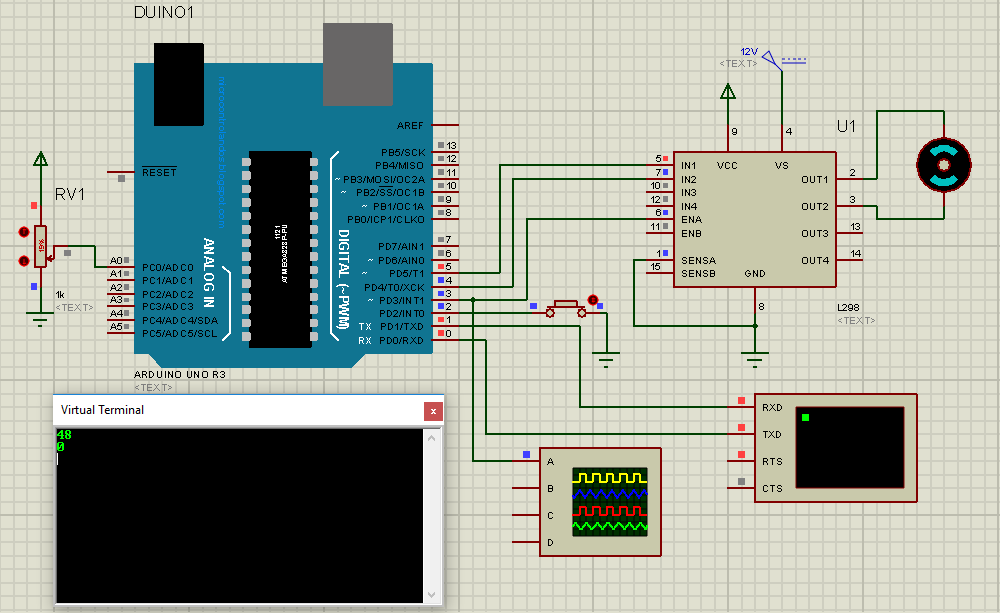
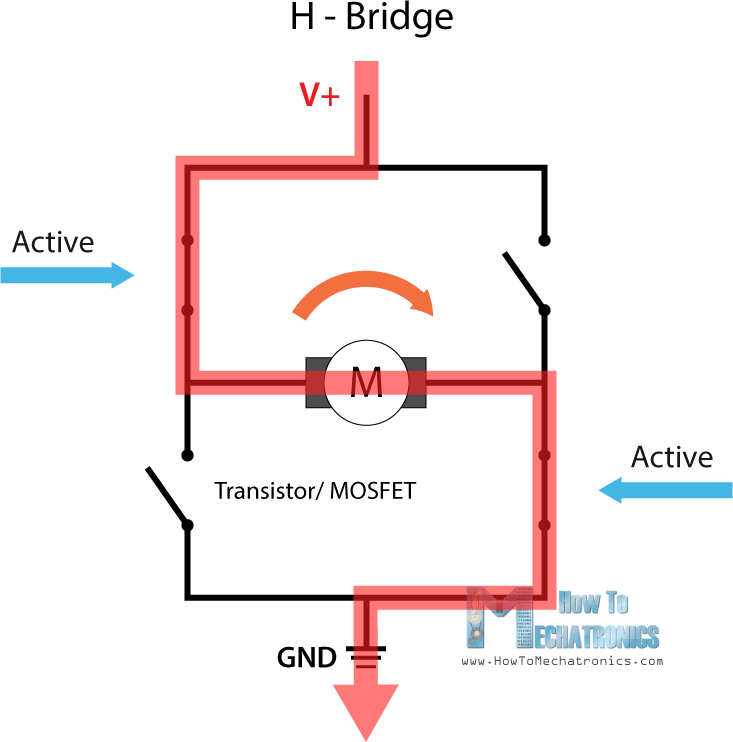


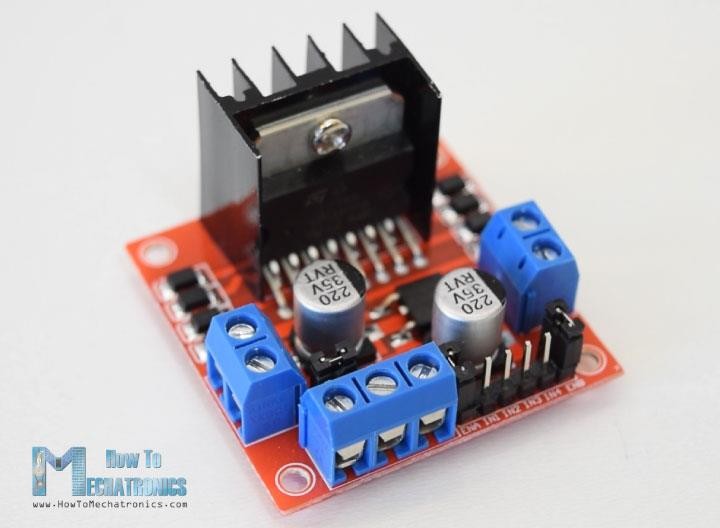
Figure : Button Pressed, Clockwise Motion, Minimum Speed



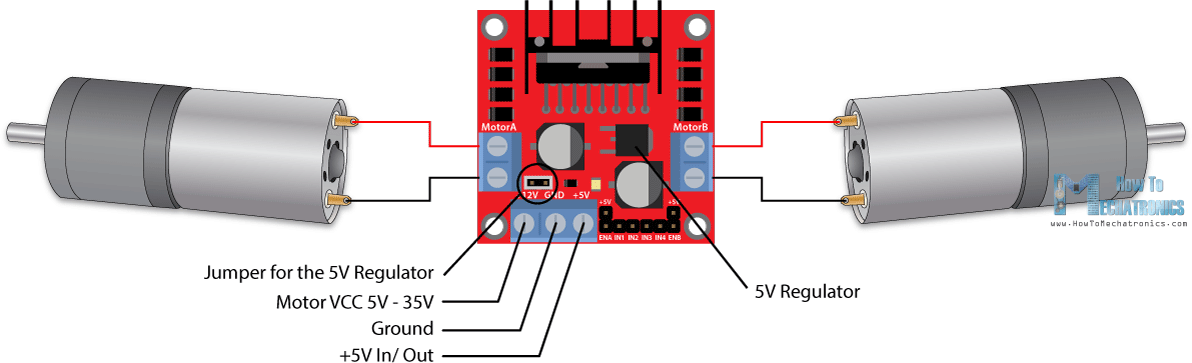
So if we combine these two methods, the **PWM** and the H-Bridge, we can have a complete control over the DC motor. There are many DC motor drivers that have these features and the L298N is one of them.

### L298N Driver

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.



Let’s take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.



This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because

those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly.

#### Stepper Motor

A stepper motor is an electromechanical device it converts electrical power into mechanical power. Also it is a brushless, synchronous electric motor that can divide a full rotation into an expansive number of steps. The motor’s position can be controlled accurately without any feedback mechanism, as long as the motor is carefully sized to the application. Stepper motors are similar to switched reluctance motors.

The stepper motor uses the theory of operation for magnets to make the motor shaft turn a precise distance when a pulse of electricity is provided. The stator has eight poles, and the rotor has six poles. The rotor will require 24 pulses of electricity to move the 24 steps to make one complete revolution. Another way to say this is that the rotor will move precisely 15° for each pulse of electricity that the motor receives.



# Types of Stepper Motor:

There are three main types of stepper motors, they are:

1. Permanent magnet stepper
2. Hybrid synchronous stepper
3. Variable reluctance stepper

**Permanent Magnet Stepper Motor:** Permanent magnet motors use a permanent magnet (PM) in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets.

**Variable Reluctance Stepper Motor:** Variable reluctance (VR) motors have a plain iron rotor and operate based on the principle that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles.

**Hybrid Synchronous Stepper Motor:** Hybrid stepper motors are named because they use a combination of permanent magnet (PM) and variable reluctance (VR) techniques to achieve maximum power in a small package size.

# Advantages of Stepper Motor:

1. The rotation angle of the motor is proportional to the input pulse.
2. The motor has full torque at standstill.
3. Precise positioning and repeatability of movement since good stepper motors have an accuracy of 3 – 5% of a step and this error is non cumulative from one step to the next.
4. Excellent response to starting, stopping and reversing.
5. Very reliable since there are no contact brushes in the motor. Therefore the life of the motor is simply dependent on the life of the bearing.
6. The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.
7. It is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft.
8. A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses.

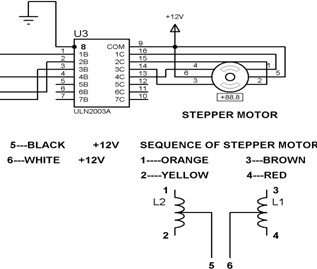
# Applications:

1. **Industrial Machines** – Stepper motors are used in automotive gauges and machine tooling automated production equipments.
2. **Security** – new surveillance products for the security industry.
3. **Medical** – Stepper motors are used inside medical scanners, samplers, and also found inside digital dental photography, fluid pumps, respirators and blood analysis machinery.
4. **Consumer Electronics** – Stepper motors in cameras for automatic digital camera focus and zoom functions.

And also have business machines applications, computer peripherals applications.

# Operation of Stepper Motor:

Stepper motors operate differently from [DC brush motors](https://www.edgefxkits.com/four-quadrant-dc-motor-control-without-microcontroller), which rotate when voltage is applied to their terminals. Stepper motors, on the other hand, effectively have multiple toothed electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external control circuit, for example a microcontroller.



**In-Lab Task 2**

**Control the steps of a stepper motor in full stepping mode using ULN2003 IC. Use push button to decide the direction of motion.**

|  |
| --- |
| **ARDUINO IDE CODE** |
| bool a[4][4]={{1,0,0,0},  {0,1,0,0},  {0,0,1,0},  {0,0,0,1}};  bool a1[4][4]={{0,0,0,1},  {0,0,1,0},  {0,1,0,0},  {1,0,0,0}};  void setup() {  // put your setup code here, to run once:  pinMode(2,OUTPUT);  pinMode(3,OUTPUT);  pinMode(4,OUTPUT);  pinMode(5,OUTPUT);  pinMode(6,INPUT\_PULLUP);  }  void loop() {  // put your main code here, to run repeatedly:  int k =digitalRead(6);  int delay1=700;  if (k==LOW)  {  for(int j=0;j<4;j++)  {  digitalWrite(2,a[j][0]);  digitalWrite(3,a[j][1]);  digitalWrite(4,a[j][2]);  digitalWrite(5,a[j][3]);  delay(delay1);  }  }  else  {  for(int j=0;j<4;j++)  {  digitalWrite(2,a1[j][0]);  digitalWrite(3,a1[j][1]);  digitalWrite(4,a1[j][2]);  digitalWrite(5,a1[j][3]);  delay(delay1);  }  }  } |

**PROTEUS IMPLEMENTATION:**

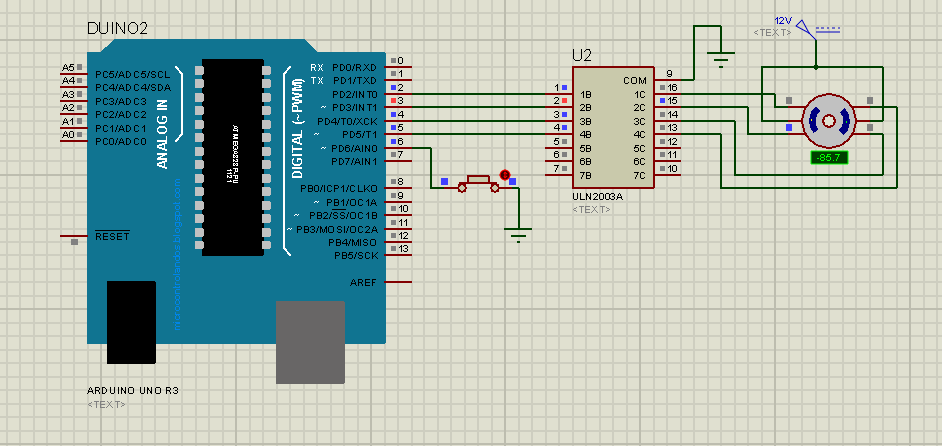


Figure : Button Pressed: Anticlockwise Motion

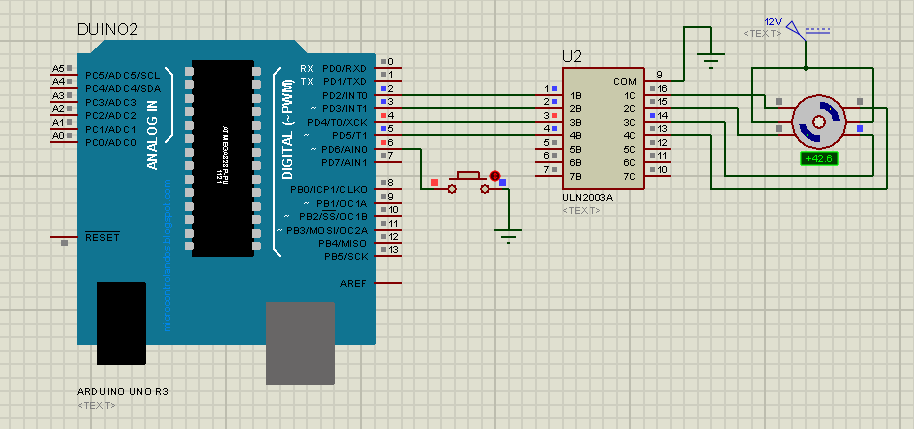


Figure : Button Not Pressed: Clockwise Motion

**Post-Lab Task 1**

**Design H-Bridge for 12V/10A rating using MOSFETs and control speed of DC-Motor using this circuit is Software.**

|  |
| --- |
| **ARDUINO IDE CODE** |
| void setup() {  // put your setup code here, to run once:  pinMode(A0,INPUT);  pinMode(10,OUTPUT);  pinMode(9,OUTPUT);  pinMode(2,INPUT\_PULLUP);  digitalWrite(2,HIGH);  Serial.begin(9600);  }  void loop() {  int a=analogRead(A0);  unsigned int speed1=map(a,0,1023,0,255);  Serial.print("Speed=");  Serial.println(speed1);    int flag=digitalRead(2);  Serial.println(flag);  if(flag==LOW)  {  analogWrite(10,speed1);  digitalWrite(9,LOW);  }  else  {  digitalWrite(10,LOW);  analogWrite(9,speed1);  }  delay(500);  } |

**PROTEUS IMPLEMENTATION:**

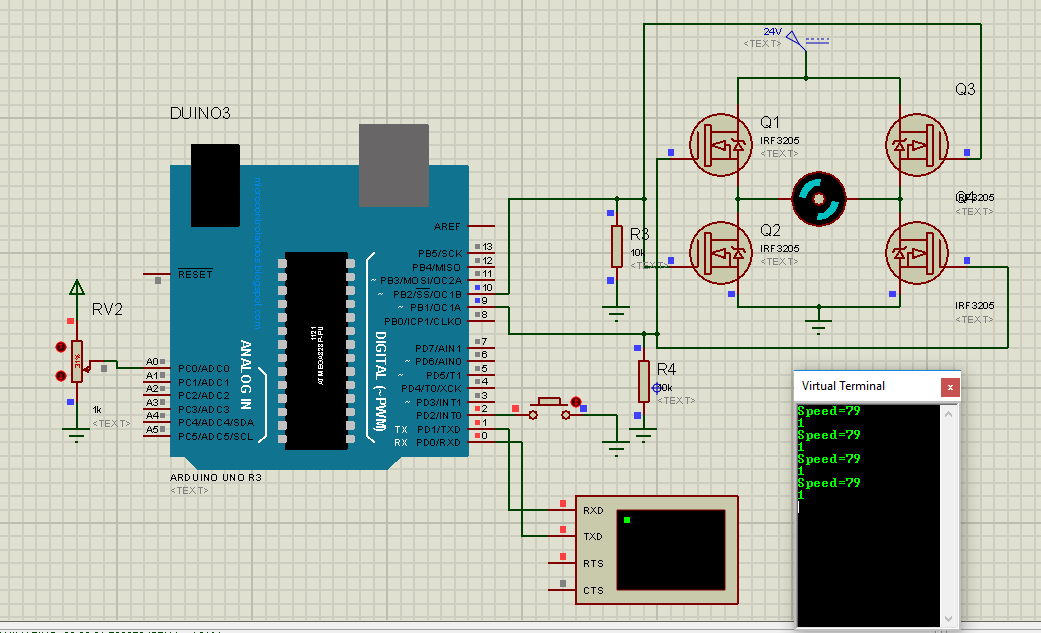


Figure : Button Not Pressed: Clockwise Motion; Speed controlled with potentiometer displayed on virtual terminal

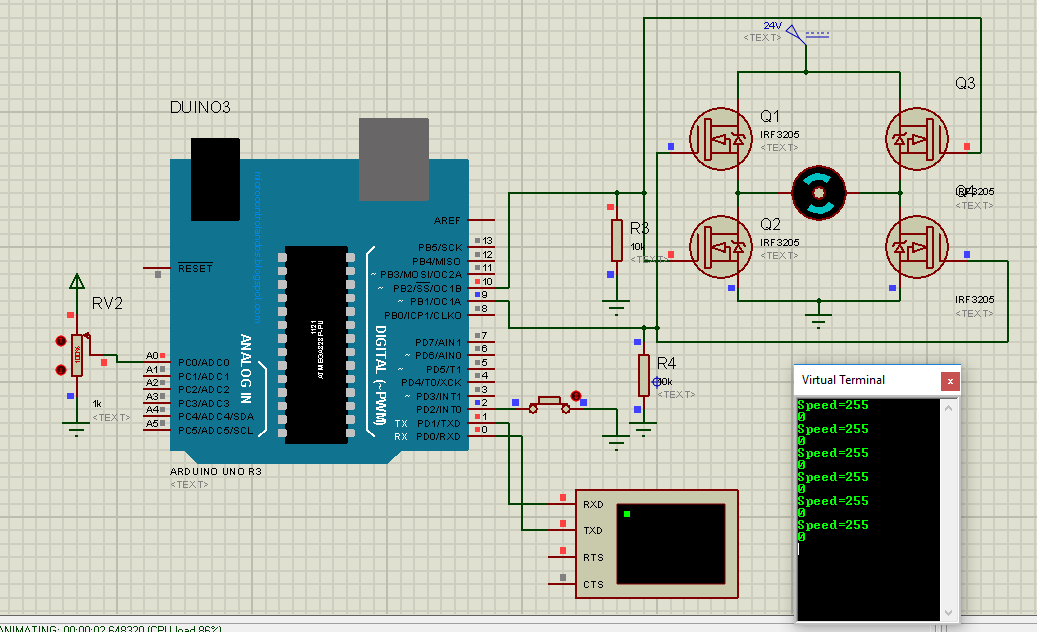


Figure : Button Pressed: Anticlockwise Motion; Speed controlled through potentiometer and displayed on virtual terminal

**Post-Lab Task 2**

**Control the stepper motor motion and direction using A4988 IC using Hardware.**

|  |  |
| --- | --- |
| **ARDUINO IDE CODE** | **PROTEUS SCHEMATIC** |
|  |  |

**Critical Analysis/Conclusion:**

|  |
| --- |
| In this lab we learnt about different methods of interfacing different motors with our Microcontroller, the uses and applications of these motors and what methods of interfacing best suit our requirements.  Normal DC Gear Motors interfaced with L298N motor driver were used in the first task  In the second task however we had an in-depth study of Stepper motors, their types and their drivers.  We also learnt that at high voltages and currents it is better to use power MOSFETs based H Bridge. If it is required to drive a DC motor direncly from Microcontroller output pins we can use Darlington pair of capacitors to provode adequate current and voltage. We can also use solid state relays and optocouplers/optoisolators for switching purposes on an H-Bridge but they don’t provode very fast switching due to mechanical delays. We should also use a flywheel diode in parallel with the motor as it is an inductive load and when it stops it produced a huge back current which can damage the Microcontroller. We can also use an NPN transistor for this purpose which doesn’t allow any current to flow towards the base that is connected to the microcontroller pin. |

