**7. RESULTS AND DISCUSSIONS**

**7.1 RESULTS**

The designed model in our project works for both open loop and closed loop type. Thus practice of speed control of DC motor can be done in two ways. They are

1. Duty ratio as input in Open-loop control
2. Speed in RPM (set speed) as input in Closed loop control

The model is designed and executed by using Simulink MATLAB 2016a, and implemented with the help of Arduino Mega 2560 developer board. Here Simulink plays a major role of communication interface between user and the Arduino. Simulink takes the inputs from user and converts them to the code which is understandable by Arduino, and then Arduino Produces required outputs.

In feedback path, Arduino sends the Hall Effect sensor outputs to the Simulink through DIGIAL INPUT block, and then Simulink calculates the speed and plots it in the SCOPE. It helps in analyzing the system response for different inputs.

Fig.7.1 Developed Kit For Speed Control Of DC Motor Using H-Bridge Controller through Simulink

**7.1.1 OPEN LOOP CONTROL MODEL**

For a given input of duty ratio from Simulink and Potentiometer, the resultant outputs are shown below.

**7.1.1.1 OPEN LOOP CONTROL WITH SIMULINK:**

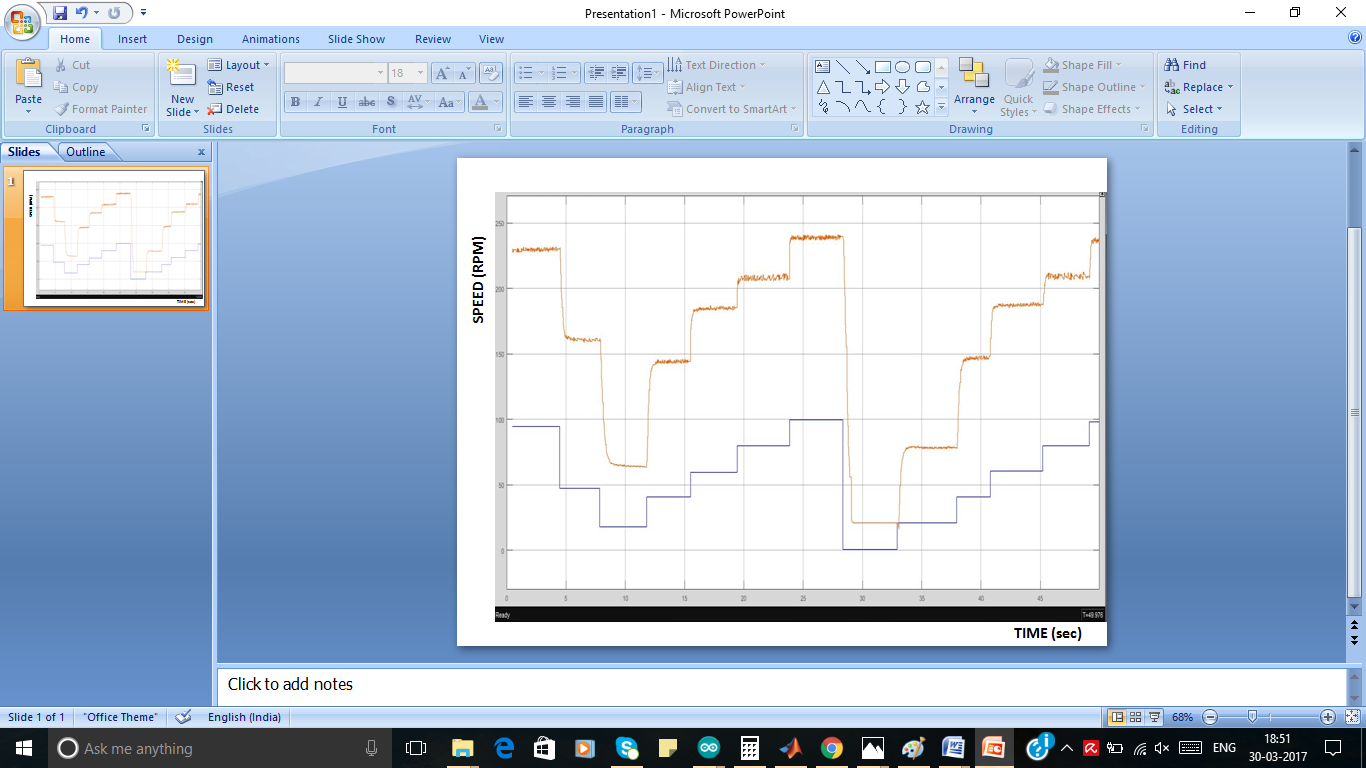
****

Figure 7.2 Result analysis for an input from Simulink constant source block

**BLUE Duty Ratio** **ORANGE Actual Speed of Motor**

The result shows that the motor is responding to the Simulink input linearly, here in the scope a graph for variation of Duty ratio and Speed of DC Motor are plotted with respect to time.

**7.1.1.2 OPEN LOOP CONTROL WITH POTENTIOMETER:**

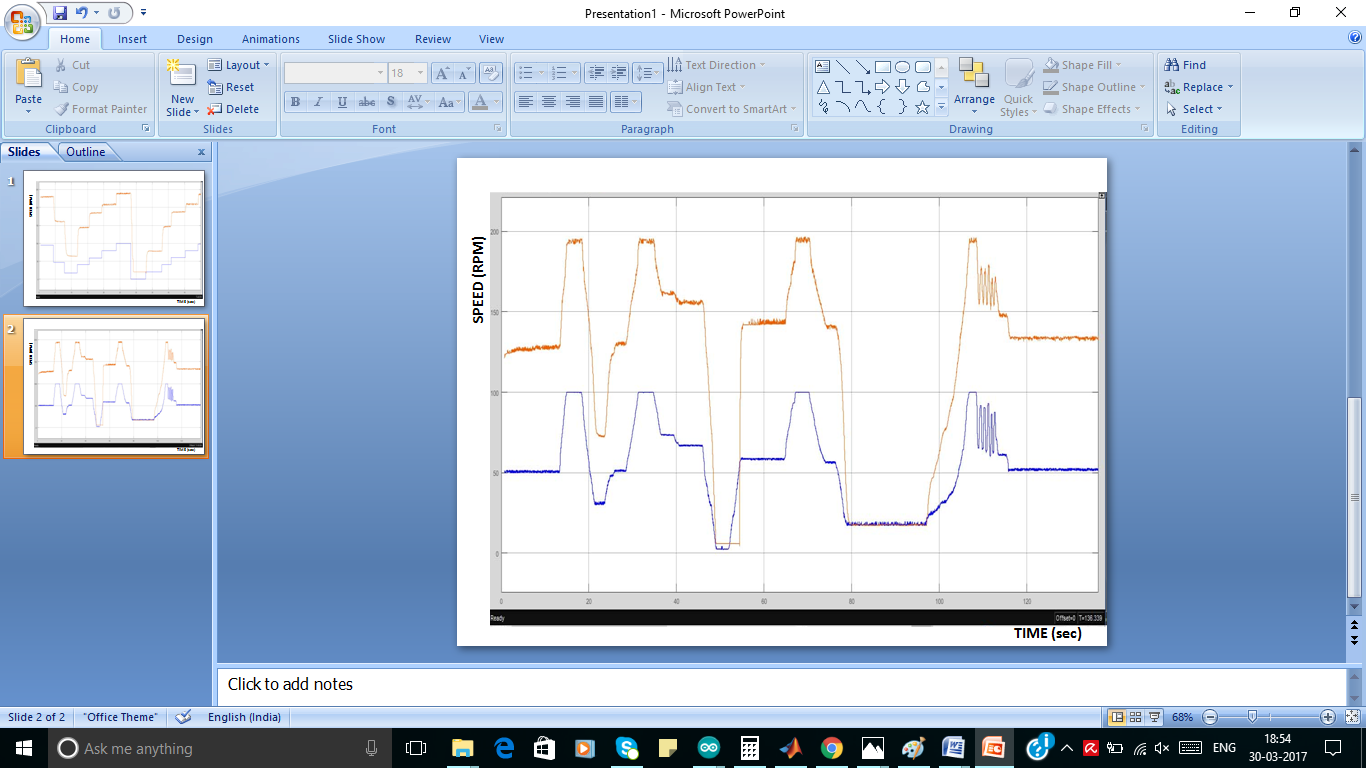
****

Figure 7.3 Result analysis for an input from Potentiometer

**BLUE Duty Ratio** **ORANGE Actual Speed of Motor**

The result shows that the motor is responding to the Potentiometer input linearly, here in the scope a graph for variation of Duty ratio and Speed of DC Motor are plotted with respect to time.

**7.1.2 CLOSED LOOP CONTROL MODEL**

In closed loop control directly speed of the motor is given as input which is a step signal of step time 5sec and value 185 RPM. Here controller compares the actual speed of motor and set speed and calculates the duty ratio, required to reach the set speed. Then PWM signal will be generated by Arduino this controlled firing pulses are fed to the optocoupler in order to control the speed of DC motor. The resultant output will be as follows:

**7.1.2.1 CLOSED LOOP CONTROL WITH P CONTROLLER**

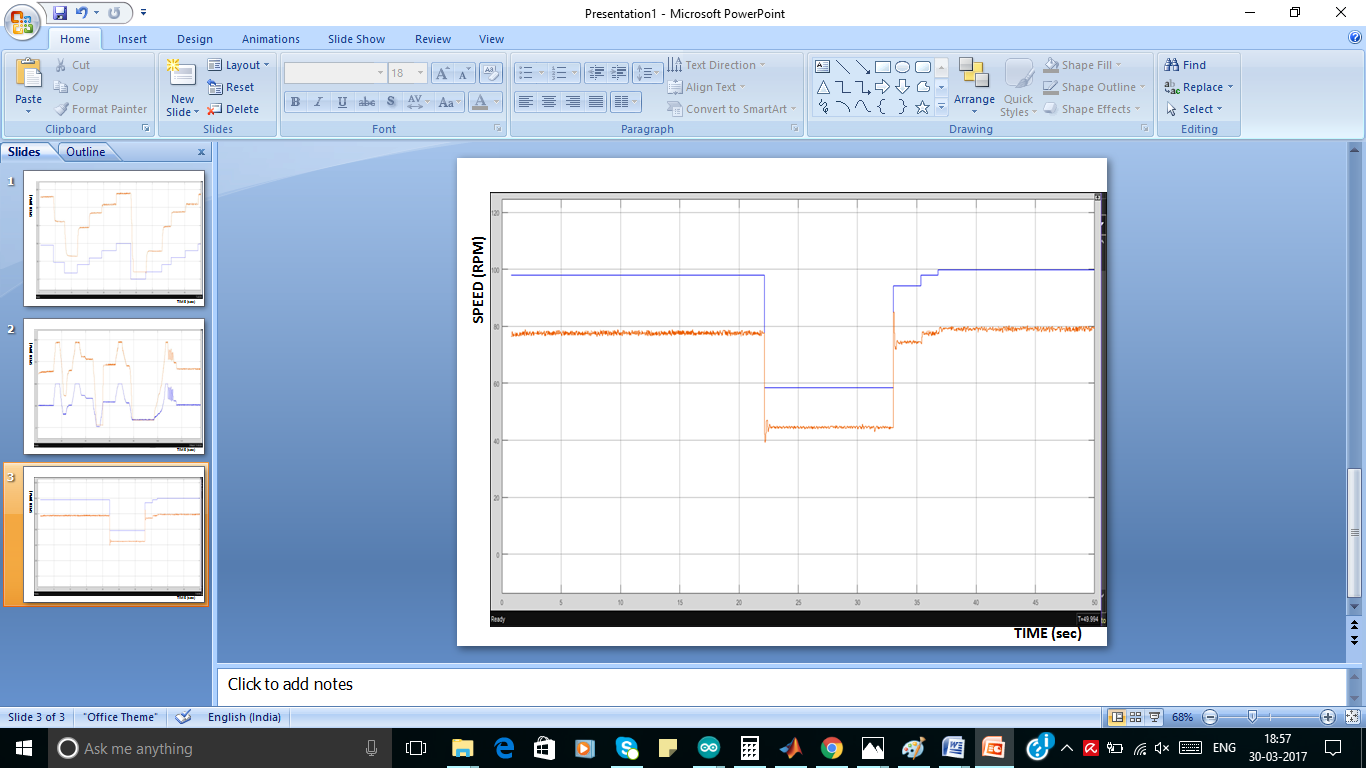
****

Figure 7.4 Closed Loop Control With P Controller

**BLUE Duty Ratio** **ORANGE Actual Speed of Motor**

The above shown figure is the response of the closed loop control of DC motor using only P controller, here we can analyze that steady state error is occurred, to reduce the steady state error we are going for PI controller

**7.1.2.2 CLOSED LOOP CONTROL WITH PI CONTROLLER**

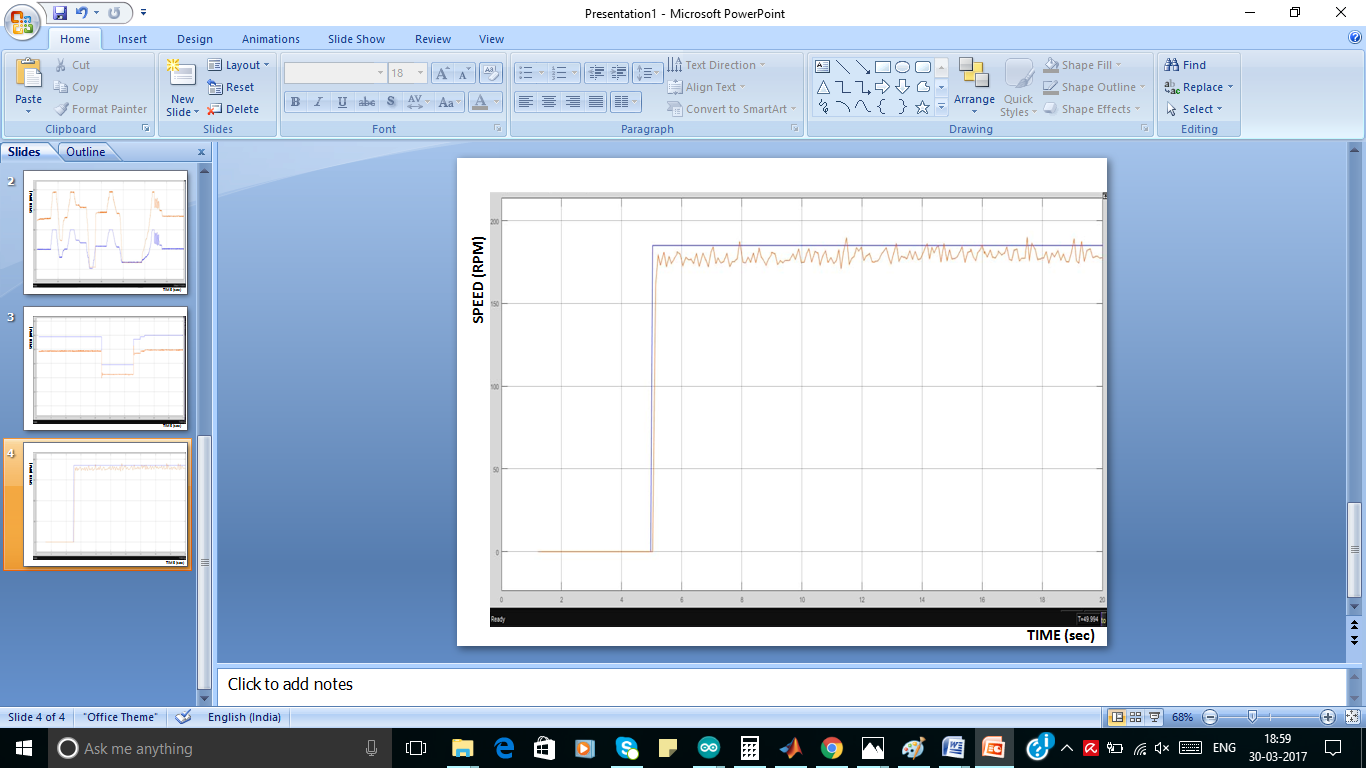


Figure 7.5 Closed Loop Control With PI Controller

**BLUE Duty Ratio** **ORANGE Actual Speed of Motor**

The above shown figure is the response of the closed loop control of DC motor using PI controller for a fixed input of step signal, here we can analyze that steady state error is reduced but oscillations are occurred, to reduce the steady state error we are going for PID controller

**7.1.2.3 CLOSED LOOP CONTROL WITH PID CONTROLLER**

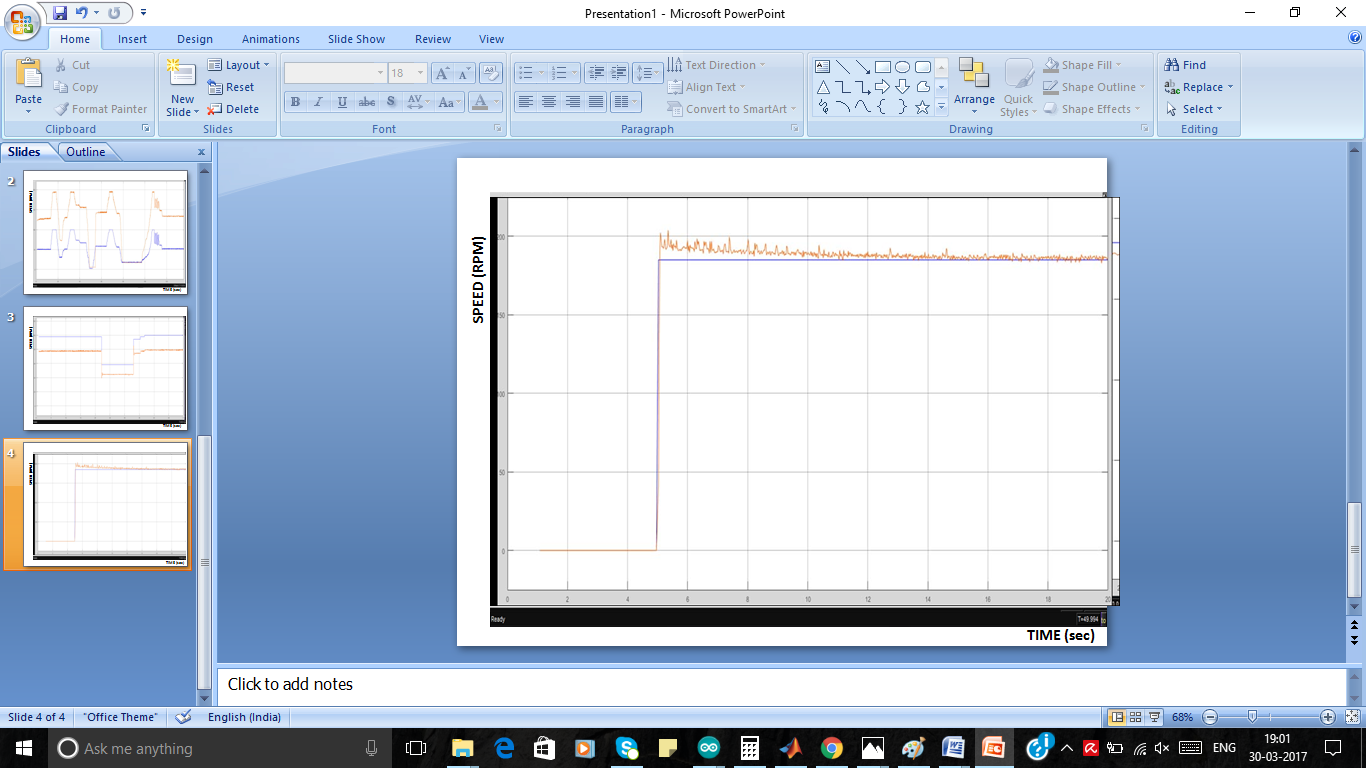
****

Figure 7.6 Closed Loop Control With PID Controller

**BLUE Duty Ratio** **ORANGE Actual Speed of Motor**

The above shown figure is the response of the closed loop control of DC motor using PID controller for a fixed input of step signal, here we can analyze that steady state error is reduced and oscillations are also reduced finally system stabilized in 4 seconds.

**7.1.3 DIRECTION CONTROL OF DC MOTOR**

H-Bridge controller is capable of controlling speed and direction of DC motor, direction of DC motor is controlled by changing the polarity a supply at armature terminal.

In our project direction control result is shown by using Digital Multi Meter, it is capable of showing the positive and negative voltages across its probes.

**7.1.3.1 FORWARD DIRECTION OF DC MOTOR**

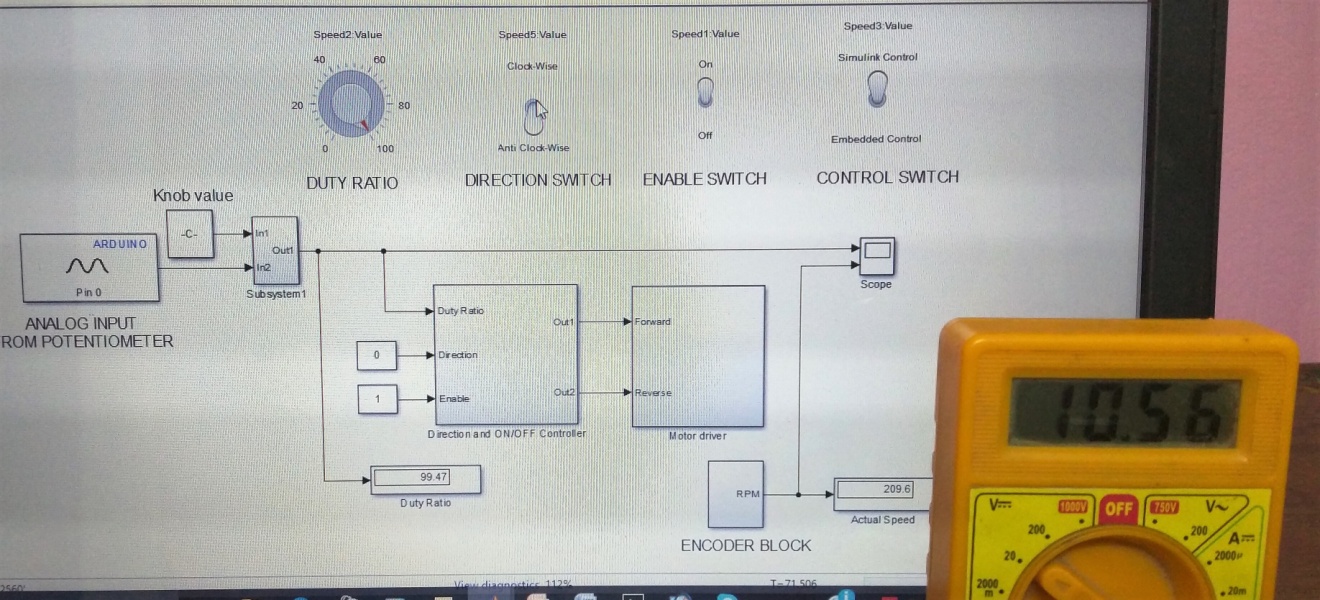
****

Figure 7.7 Forward direction of DC motor

In the above figure we can observe that the Direction Switch is in down position and giving a value 0 to the direction input to the DIRECTION AND ON/OFF Subsystem, and then a positive voltage across the motor terminals is shown in DMM

**7.1.3.2 REVERSE DIRECTION OF DC MOTOR**

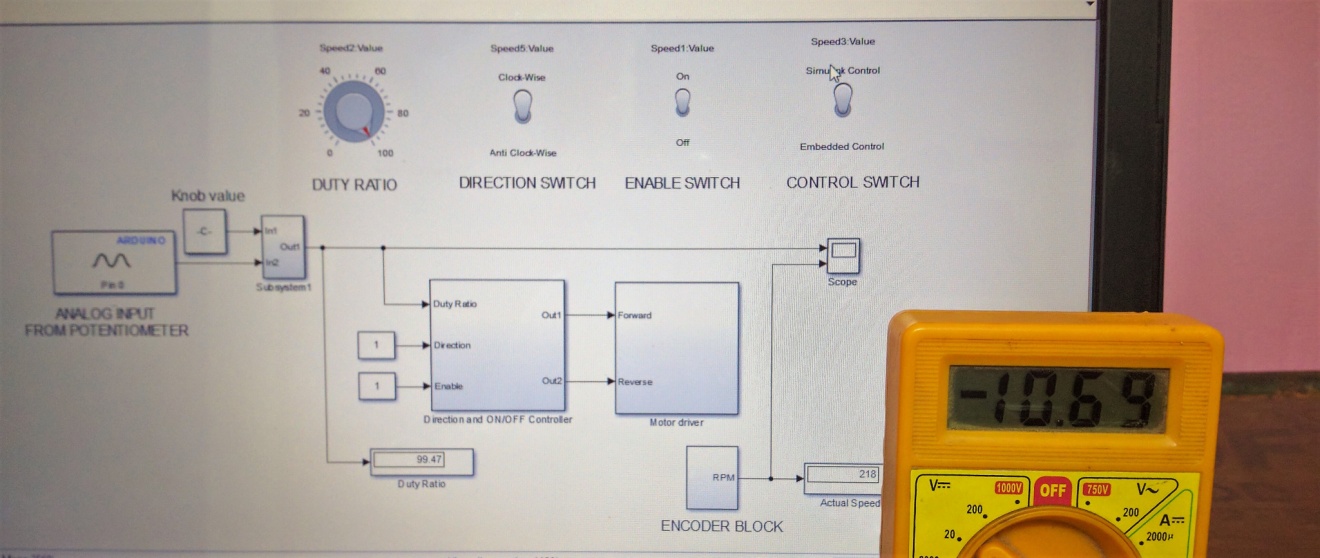
****

Figure 7.8 Reverse direction of DC motor

In the above figure we can observe that the Direction Switch is in up position and giving a value 1 to the direction input to the DIRECTION AND ON/OFF Subsystem, and then a negative voltage across the motor terminals is shown in DMM, it represents the change of direction of motor.

**7.2 DISCUSSIONS**

* In this project exact required speed is achieved by microcontroller which reduces the complexity of the speed control.
* In this project the need for using embedded C language for programming ATMEGA2560 is eliminated by using Simulink.
* Simulink eliminated the use of large programs and complex instruction set by providing Simulink support packages for Arduino.
* External mode in Simulink helped a lot in plotting the response of motor in the scope and also provided us a feasibility to change parameters while running the model.
* Accurate output is obtained with in fraction of seconds by using other electronic devices like 4N35, ARDUINO MEGA 2560, TIP41C BJT which added extra support for controlling speed .
* For proper controlling action, voltage conversions are carried out by power supply unit RPS (Regulated Power Supply), it provides safe power output and also produces a pure DC supply.