IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Controlling The DC Motor Speed With PWM Using Arduino

Mrs. A.Gamyaveni[1], Bodapati Swathi[2], [3]Pallisetty Venkata SiriVennela[3], Chandra Mahima Chowdary[4], Amarthaluri Manikumar[5], Thota Saikumar[6]

Assistant Professor[1], [2]UG Students[2,3,4,5,6]
Department of Electrical and Electronics
Kkr & Ksr institute of technology and sciences, Guntur, India

Abstract: This research represents the use of PWM (pulse width modulation) of microcontroller for controlling on direct current (DC) motor. DC motor speed is controlled using PWM technique. The signal of PWM is produced by Arduino Uno board. The Arduino Uno reads thee input from the potentiometer, which serves as a speed control knob for the DC motor. It then generates PWM signals accordingly to control the MOSFET, adjusting the motor speed. The Hall effect sensor provides feedback on the motors rotational speed, enabling the Arduino to adjust the PWM duty cycle to maintain the desired speed. Micro C language is utilized for programming and controlling the Arduino uno board. The aim of this research is to control the speed and direction of a DC motor, so that it can be used in automation systems and industrial fields. The speed of the motor is controlled by adjusting the duty cycle of the PWM signal output by microcontroller.

Keywords: DC motor, Pulse width modulation, Arduino Uno, 16*2 I2c LCD, IRF 540N MOSFET, Hall effect sensor, Potentiometer, Power supply, Feedback control, Motor speed control, Magnet sensor

I. Introduction

In the realm of electronics and automation, precise control over the speed of direct current (DC) motors is indispensable for a multitude of applications, ranging from robotics to industrial machinery.

This project embarks on the journey of harnessing the power of Arduino microcontrollers and Pulse Width Modulation (PWM) techniques to achieve dynamic and accurate speed regulation of a DC motor. By leveraging an array of components including a 12V, 1A Switched-Mode Power Supply (SMPS) adapter, an Arduino Uno (often colloquially referred to as "Arduo limo"), a 16x2 I2C Liquid Crystal Display (LCD), an IRF 540N MOSFET, a Hall Effect sensor, a potentiometer for user input, and other essential elements, this project endeavors to construct a robust system. capable of seamlessly controlling the speed of a DC motor.

In the project of controlling the DC motor speed with PWM using Arduino, it is to regulate the motor speed by adjusting the Pulse Width Modulation signal generated by the Arduino. PWM is a technique where the average voltage supplied to the motor is controlled by varying the duty cycle of a square wave signal. The Arduino Uno board serves as the central controller, and it generates the PWM signals to regulate the motor speed. The speed of a DC motor is generally directly proportional to the supply voltage. The project's foundation lies in the 12V, 1A SMPS adapter, which serves as the primary power source to drive both the Arduino Uno and the DC motor. The Arduino Uno acts as the central processing unit, orchestrating the generation of PWM signals to modulate the speed of the motor. Visual feedback and user interaction are facilitated through the integration of a 16x2 I2C LCD, providing real-time display of motor speed and other relevant information.

The IRF 540N MOSFET serves as the intermediary between the Arduino's PWM to achieve the desired speed control. Additionally, a Hall Effect sensor is employed to provide feedback on the motor's speed or position, enhancing the system's accuracy and stability. A potentiometer allows users to manually adjust the motor speed, adding a layer of versatility and user-friendliness to the project. Supporting components such as 10k, 100% resistors, an IN4007 diode, a magnet sensor, and a magnet are strategically integrated to ensure the system's reliability, safety, and efficiency.

The primary objective of this project is to develop a sophisticated yet accessible system for controlling the speed of a DC motor using PWM signals generated by an Arduino microcontroller. By seamlessly integrating hardware components and software algorithms, the system aims to provide precise, responsive, and user-friendly speed regulation capabilities suitable for a wide range of applications. Whether it be for hobbyist projects, educational endeavors, or industrial automation setups, the versatility and effectiveness of Arduino-based motor control systems are showcased through this project. Through meticulous design, implementation, and testing, this project aims to demonstrate the seamless synergy between hardware and software in achieving optimal motor control performance.

II. METHODOLOGY

Design the circuit and connections based on the project requirement and component specifications. It includes connecting the Arduino uno to the various components such as the motor driver (IRF 540N), potentiometer, Hall effect sensor, and LCD display, ensuring proper wiring and interfacing. Assemble the hardware components according to the circuit, ensuring all the connections are secure and properly insulated to prevent short circuits. Write the Arduino code to implement the PWM control algorithm for regulating the speed of the DC motor. It involves initializing the PWM pins, reading input from the potentiometer, and adjusting the PWM output accordingly to control the motor speed.

Incorporate the Hall effect sensor and magnet sensor into the circuit and program to provide feedback on the motors speed. It involves reading sensor data and adjusting the PWM output to maintain the desired speed. Integrate the 16*2 I2C LCD display and potentiometer to provide a user-friendly interface adjusting the motor speed manually and displaying real-time feedback on the motors speed.

III. BLOCK DIAGRAM:

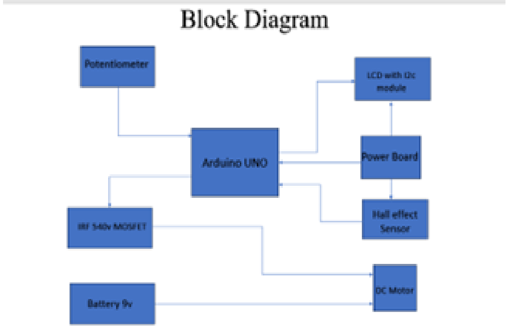


Fig (1) The Block Diagram of our proposed concept is given

1. Potentiometer: This is a variable resistor used to control the PWM signal. The user can adjust the potentiometer to change the motor speed.

2. Arduino UNO: This is the microcontroller that reads the potentiometer value, generates the PWM signal based on value.

reads the Hall effect sensor data and controls the LCD display.

3. IRF 540V MOSFET: This is a type of transistor that switches the motor on and off at a high frequency to control the motor speed. The switching frequency is determined by the PWM signal from the Arduino.

4. Battery 9v: This provides the power for the motor.

5. DC Motor: The speed of this motor is controlled by the PWM signal.

6. Hall Effect Sensor: This sensor detects the motor's speed and sends this data back to the Arduino.

7. LCD with i2c module: This displays the motor's speed and the duty cycle of the PWM signal.

8. Power Board: This provides power to the Arduino and other components.

IV. CIRCUIT DIAGRAM:

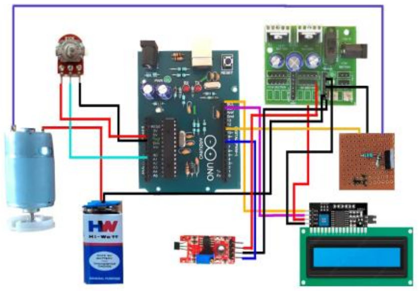


Fig (2)The Circuit Diagram of our proposed concept is given

V. ALGORITHM:

1. Start Program: The program begins.

2. Initialize LCD & Display Message: The LCD (Liquid Crystal Display) is initialized, and a message is displayed on it.

3. Read Potentiometer (Volume) Value: The program reads the value of a potentiometer, which is used to control the volume or intensity of something, in this case, the motor speed.

4. Map Potentiometer Value to Duty Cycle (0-255): The potentiometer value is then mapped to a duty cycle value between 0 and 255. The duty cycle is the percentage of one period in which a signal is active.

5. Set PWM Output Based on Duty Cycle: The program sets the Pulse Width Modulation (PWM) output based on the duty cycle. PWM is a method used for getting variable voltage over a digital signal.

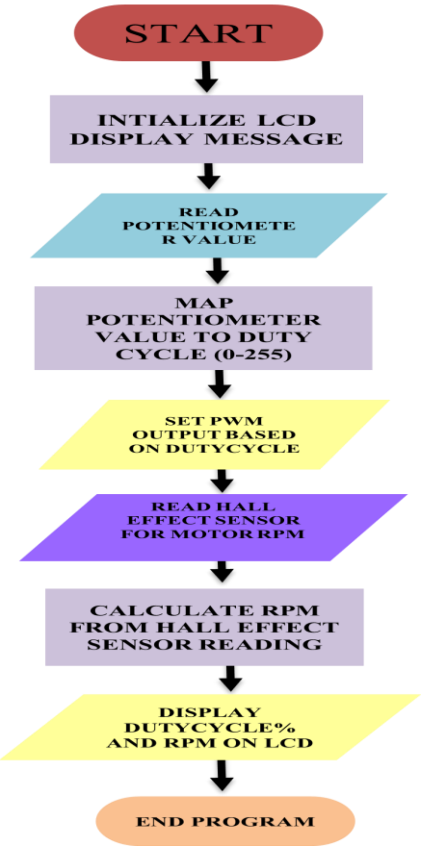
6. Read Hall Effect Sensor for Motor RPM: The program reads the value from a Hall Effect sensor, which is used to measure the motor's RPM.

7. Calculate RPM from Hall Effect Sensor Readings: The program calculates the RPM based on the readings from the Hall Effect sensor.

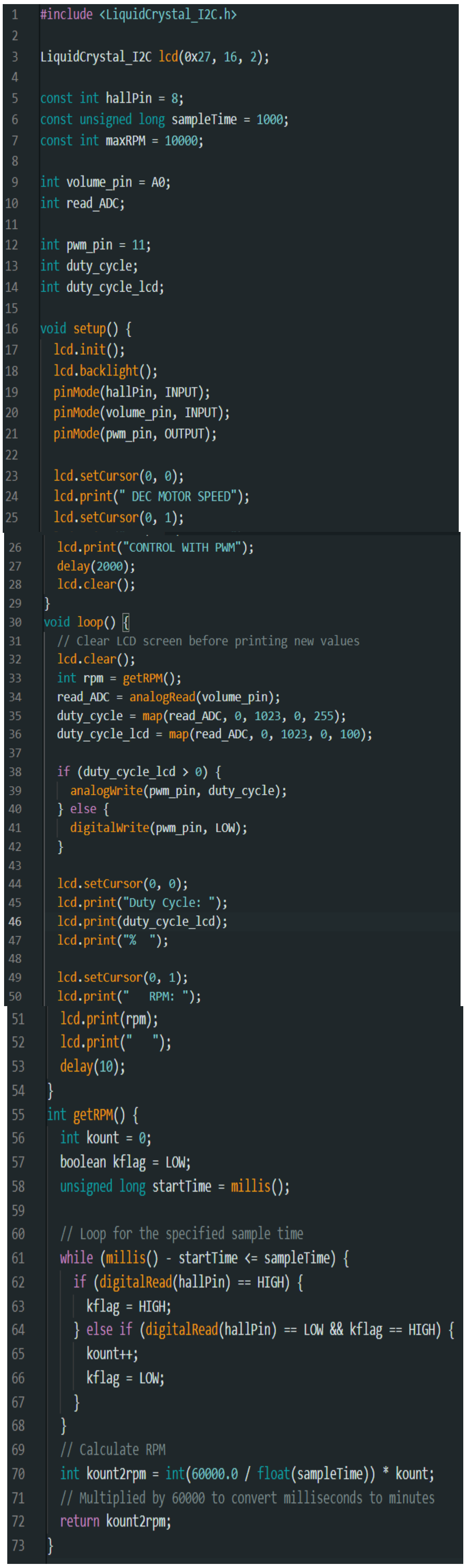
8. Display Duty Cycle and RPM on LCD: The calculated RPM and the duty cycle are displayed on the LCD.

9. End Program**:** The program ends.

VI. FLOWCHART:


START
INTIALIZE LCD DISPLAY MESSAGE
READ POTENTIOMETE R VALUE
MAP POTENTIOMETER VALUE TO DUTY CYCLE (0-255)
SET PWM OUTPUT BASED ON DUTYCYCLE
READ HALL EFFECT SENSOR FOR MOTOR RPM
CALCULATE RPM FROM HALL EFFECT SENSOR READING
DISPLAY DUTYCYCLE% AND RPM ON LCD
END PROGRAM

VII. ARDUINO CODE:

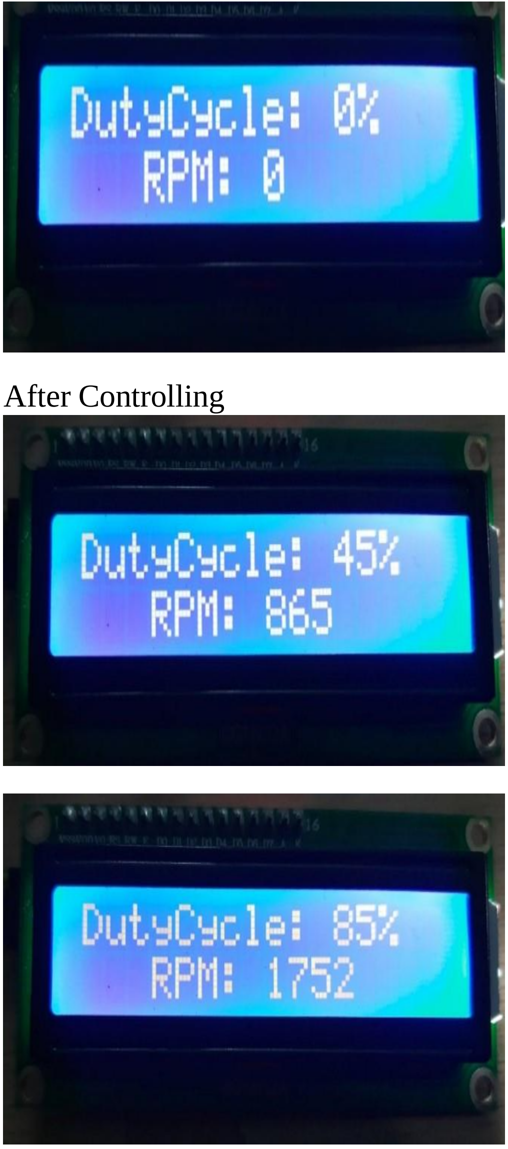


```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int hallPin = 8;
const unsigned long sampleTime = 1000;
const int maxRPM = 10000;

int volume_pin = A0;
int read_ADC;

int pwm_pin = 11;
int duty_cycle;
int duty_cycle_lcd;

void setup() {
  lcd.init();
  lcd.backlight();
  pinMode(hallPin, INPUT);
  pinMode(volume_pin, INPUT);
  pinMode(pwm_pin, OUTPUT);

  lcd.setCursor(0, 0);
  lcd.print(" DEC MOTOR SPEED");
  lcd.setCursor(0, 1);
  lcd.print("CONTROL WITH PWM");
  delay(2000);
  lcd.clear();
}
void loop() {
  // Clear LCD screen before printing new values
  lcd.clear();
  int rpm = getRPM();
  read_ADC = analogRead(volume_pin);
  duty_cycle = map(read_ADC, 0, 1023, 0, 255);
  duty_cycle_lcd = map(read_ADC, 0, 1023, 0, 100);

  if (duty_cycle_lcd > 0) {
    analogWrite(pwm_pin, duty_cycle);
  } else {
    digitalWrite(pwm_pin, LOW);
  }

  lcd.setCursor(0, 0);
  lcd.print("Duty Cycle: ");
  lcd.print(duty_cycle_lcd);
  lcd.print("%  ");

  lcd.setCursor(0, 1);
  lcd.print("   RPM: ");
  lcd.print(rpm);
  lcd.print("   ");
  delay(10);
}
int getRPM() {
  int kount = 0;
  boolean kflag = LOW;
  unsigned long startTime = millis();

  // Loop for the specified sample time
  while (millis() - startTime <= sampleTime) {
    if (digitalRead(hallPin) == HIGH) {
      kflag = HIGH;
    } else if (digitalRead(hallPin) == LOW && kflag == HIGH) {
      kount++;
      kflag = LOW;
    }
  }
  // Calculate RPM
  int kount2rpm = int(60000.0 / float(sampleTime)) * kount;
  // Multiplied by 60000 to convert milliseconds to minutes
  return kount2rpm;
}
```

Fig (4) The code images in the Arduino IDE software

VIII. RESULT:

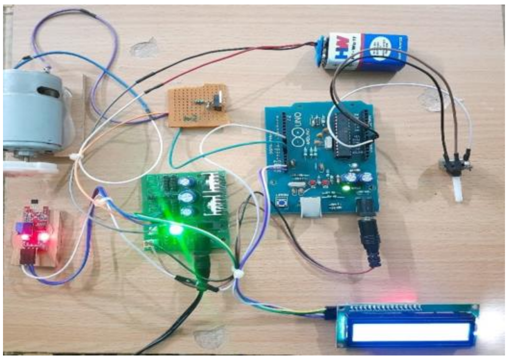
Before Controlling



After Controlling

Fig(5,6,7)Output of the project

IX. PROJECT KIT:



Fig(8)Project Kit image

In this project the DC power supply is supplied from the power board to all the components like Arduino uno, hall effect sensor, LCD display with I2C module. When we vary the potentiometer the Arduino reads its value according to the program then it generate the PWM signals, based on the pot value. The generated signals from specific pins(PWM pins), the generated PWM signals are given to the IRF 540N it supply's the power according to the PWM signals and the motor and the another terminal is connected to the battery then motor rotates. The motor shaft contains the magnet cap. In front of that cap, hall effect sensor is arranged. When the magnet is moves in front of the hall effect sensor it generates the signal. The generated signals are read by the Arduino uno, it calculate the RPM based on that signals. Then it print the data according to the program on the LCD display, it displays the PWM Duty cycle percentage and RPM.

OVERVIEW:

This project focuses on precise speed control of a DC motor using pulse width modulation (PWM) techniques. The system components include an Arduino Uno, a hall effect sensor, an LCD display with I2C module, and the robust IRF540N MOSFET as the motor driver. Detail explanation of the project

OPERATION FLOW:

1. Power Supply Distribution:
   - The DC power supply originates from the power board.
   - It is distributed to all components, including the Arduino Uno, hall effect sensor, and LCD display.
2. Potentiometer Variation:
   - As we adjust the potentiometer, the Arduino reads its value.
   - The Arduino program then generates PWM signals based on the potentiometer value.
3. PWM Signal Generation:
   - The generated PWM signals are directed to specific pins (PWM pins) on the Arduino.
   - These signals control the IRF540N MOSFET.
4. Motor Power Supply:
   - One terminal of the DC motor is connected to the IRF540N MOSFET.
   - The other terminal is connected to the battery.
   - As a result, the motor shaft begins to rotate.
5. Hall Effect Sensor Interaction:
   - Positioned in front of the motor shaft's magnet cap, the hall effect sensor detects the magnet's movement.
   - When the magnet moves in front of the sensor, it generates a signal.
6. RPM Calculation:
   - The Arduino reads the generated signals from the hall effect sensor.
   - Based on these signals, it calculates the motor's RPM (rotations per minute).
7. LCD Display:
   - The Arduino displays essential data on the LCD display:

- PWM Duty Cycle Percentage: Indicates the proportion of time the PWM signal is high.
- Motor RPM: Provides real-time feedback on the motor's rotational speed.

X. CONCLUSION:

This project showcases an integrated system that efficiently controls DC motor speed. Whether in industrial automation or hobbyist applications, our solution offers reliable and adaptable motor control. By utilizing an Arduino Uno and PWM techniques, the system can precisely adjust the motor's speed based on user input from a potentiometer. The system also provides real-time feedback on the motor's RPM and the PWM duty cycle via an LCD display. This makes the system highly suitable for both industrial automation and hobbyist applications, showcasing the versatility and reliability of such motor control solutions. This project is a great example of the practical application of electronics and programming in creating efficient and user-friendly systems.

XI. REFERENCES:

[1] Liu, J., Lin, F., & Shen, Y. (2019). Comparative study of PWM techniques for DC motor drive system. 2019 IEEE International Conference on Power, Electrical, and Electronics & Communication Engineering (PEECE), 1-5.

[2] Jain, A., Gajanan, A., & Bafna, M. (2020). Arduino based speed control of DC motor. 2020 IEEE International Conference for Innovation in Technology (INOCON), 1-5.

[3] Kumar, A., Chauhan, N., & Kumar, P. (2018). Arduino based speed control of DC motor. 2018 International Conference on Computing, Power, and Communication Technologies (GUCON), 46-49.

[4] Zhang, L., Zhu, S., & Li, Y. (2017). Research on DC motor speed control system based on PID algorithm. 2017 29th Chinese Control and Decision Conference (CCDC), 1518-1522.

[5] Wang, Y., Cai, Z., & Lu, Y. (2016). Design of microcontroller-based DC motor speed control system. 2016 IEEE International Conference on Applied System Invention (ICASI), 1237-1240.

[6] Li, J., Chen, X., & Yang, Y. (2019). Design and implementation of DC motor control system based on AVR microcontroller. 2019 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), 57-61.

[7] Sharma, A., & Kaushik, P. (2018). Design and development of temperature monitoring system using Arduino and liquid crystal display. 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 10-14.

[8] Singh, G., Rana, D., & Singh, A. (2020). Implementation of data logging system using Arduino and LCD display. 2020 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE),

[9]. JoSDC," Journal of Semiconductor Devices & Circuits." Volume 9, No.2, 2022 May- August

[10]. I-manager's, "Journal on IOT and Smart Automation". Volume 1, No. 1, 2023 January - June

[11]. Mayur Ramgiri, "INTERNET OF THINGS".

[12]. Raj Kamal, "Embedded Systems"

[13]. J.B. Gupta, "A course in Power Systems"

[14]. S Salivahana, "Electronic Devices and Circuits". Fifth Edition

[15]. B. Visvesvara Rao, K. Raja Rajeswari, P. Chalam Raju Panthulu, K. Bhaskara Rama Murty, "Electronic Circuit Analysis"

[16]. Philip T. Krein, "Elements of Power Electro