



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 5, May 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Arduino Based DC Motor Speed Control Using PWM Technique

Sindhu H K , Suresh K, Varun Kumar L S, Thippeswamy K G

U.G. Students, Department of Electrical and Electronics Engineering, Ballari Institute of Technology and Management, Ballari, Karnataka, India

ABSTRACT: The suggested system involves integrating an Arduino board with a DC motor driver circuit in order to regulate the speed of the motor. By adjusting the duty cycle of the PWM signal generated by Arduino, which in turn controls its average voltage supplied to the motor, it can govern its speed. This mode is easy to apply since it assures smooth and efficient speed control making it suitable for industrial automation and other related fields. Arduino Uno board, a DC motor, along with power supply components required make up this project's hardware setup. On the other hand; software part entails programming Arduino so as to generate PWM signals using its libraries built-in thus enabling precise user input based or predefined parameter oriented changes in velocity. Arduino Uno, receives the related readings from. This is used to regulate the velocity of the dc motor, can produce pulse-width-modulated signals to control the motor drive that regulates motor speed. The Hall Effect Sensor gives feedback on the rate at which the motor is spinning, allowing Arduino to tune its PWM's duty cycle so as to keep pace with it and maintain the desired speed. Micro C programming language is used in writing programs that run on an Arduino uno board. The purpose of this study was to develop an industrial automation system for controlling a DC motor at specific speeds and directions. By varying the duty cycle, we can change the speed of a direct current (dc) motor through generating different PWM signals via microcontroller.

KEYWORDS: Arduino Uno, board, DC motor, Duty cycle

I. INTRODUCTION

The demand for precise control in numerous applications even crosses over to DC motors. The aim of this project is to create a user-friendly and cost-effective system for controlling the speed of a DC motor using Pulse Width Modulation (PWM) and an Arduino microcontroller. DC motors are found in almost every electronic device today, such as robots, drones, fans, turntables etc. But often they fail in their work due to inability to control their speed accurately. This project seeks to provide a ready solution for that. Arduino is a very versatile platform that can be used by anyone who wants to interact with electronics through programming. In this project, we use its processor capabilities to generate PWM signals. By means of PWM technique, we alter the average voltage delivered to the motor by switching on and off the input power supply within one continuous cycle. In other words we can change the motor's rotation speed by changing the duty cycle (%) which represents how long or how short or time during each cycle when power is on.

In the fast-paced world of technology, precision is highly valued. In controlling robotic arms that move delicately or rotating a player's disc smoothly, speed control is an important aspect. This specific project explores DC motor speed control, making you the controller of this critical part. We will journey using Arduino platform and Pulse Width Modulation (PWM), an influential technique. DC motors are very common and they power a variety of things from the moving fans that keep us cool to the busy robots that are revolutionizing industries. However, their untapped potential lies in not being able to control their speeds with accuracy. Imagine if a record player was revolving at light speed or a robotic arm went berserk! This project aims at solving this problem by presenting a user-friendly and cheap answer to truly utilizing all capabilities of direct current motors. In this project I will take you step-by-step through all that you need from hardware components up till circuit assembly, but then we shall focus on one code written for Arduino which actually solves all your problems concerning it(PWM).

II. RELATED WORK

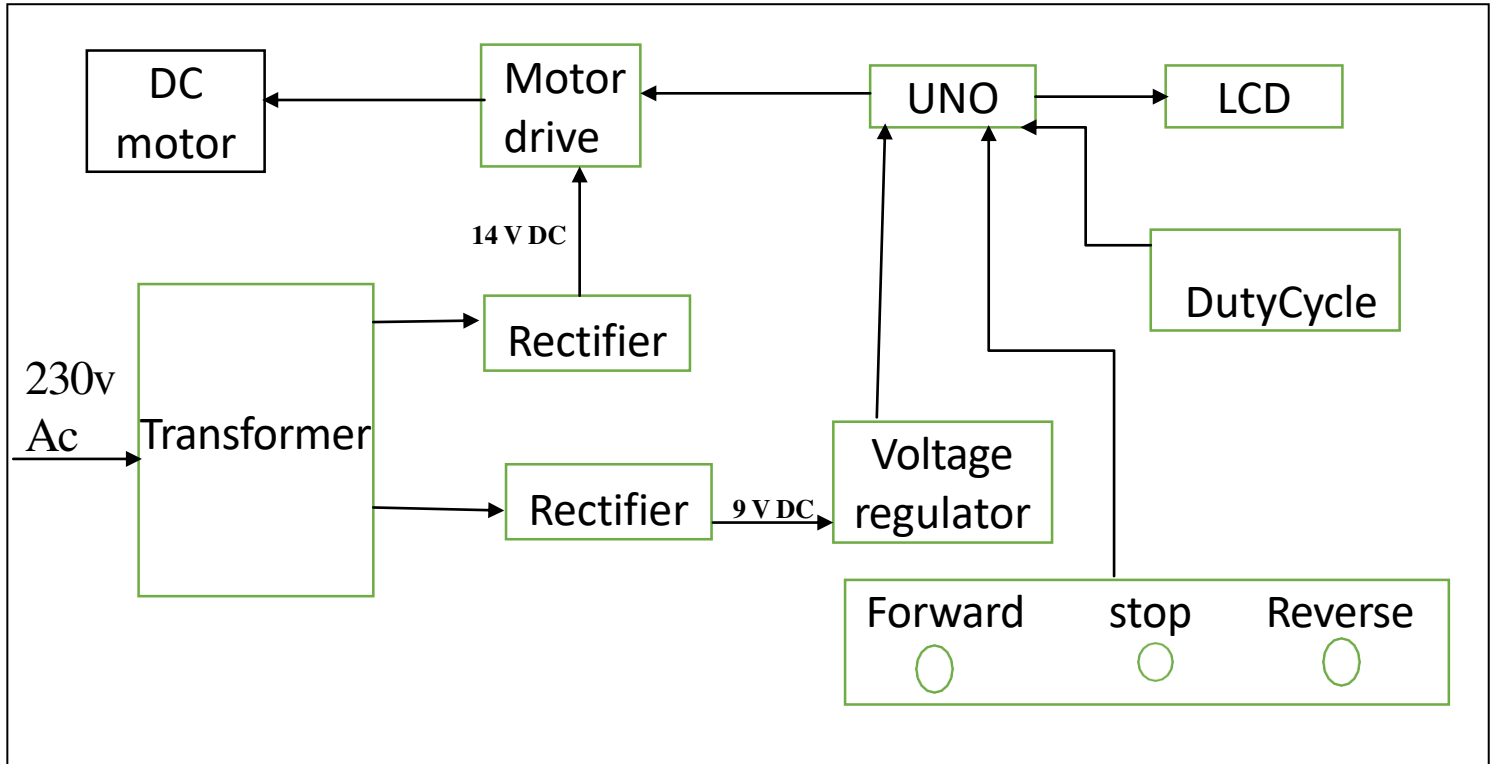
To make matters worse, all the most important current achievements and forthcoming trends will be further discussed in this article which are found on online platforms lay down the groundwork for this field. You are going to find different guides through websites or even YouTube that can provide you with a comprehensive understanding of what it looks like at its most elementary level. These materials give a precise explanation of the hardware wiring such as Arduino pins up to motor drivers or anything else that may be required, and further it has given in detail how code should be written including PWM generation functions and user input handling modules. An accurate following of these resources enables novices to build their own systems from scratch. One good example is “DC Motor Speed Control using PWM”.

The field of Arduino-based DC motor speed control is constantly in flux. Researches also involve more complicated algorithms for improving motor speed control such as PID (Proportional-Integral-Derivative) Control. Thus, it makes use of past mistakes and adjusts the PWM duty cycles accordingly thus achieving greater precision and smoothness in speed regulation. In addition, incorporating wireless communication modules like Bluetooth allows remote control of motor speeds. This makes it possible to have applications where remote operation will be needed for example robot control or adjusting fan speeds from a distance. By exploring this extensive background work, you will get insights into existing approaches, identify potential extensions for your own project and stay at the forefront of developments in this vibrant area among others. It does not matter whether you are putting up your first system as a beginner or an expert who wants to go beyond the normal expectations; there is a wealth of knowledge and projects out there that can act as resources while one learns how to master DC motor speed control using Arduino with PWM.

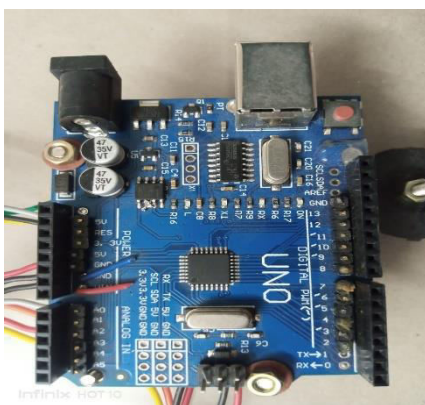
III. METHODOLOGY

In this methodology, an Arduino board with L298N motor driver module will be used for controlling the speed of a DC motor by using Pulse Width Modulation (PWM) technique. To deliver average power to a load, such as DC motors, Pulse Width Modulation (PWM) is a technique that involves rapidly switching on and off the power. The PWM signal's duty cycle determines how much average power is supplied to the load. Higher duty cycles imply that power is ON for a longer duration and therefore higher speed while lower duty cycles mean that power is ON for briefer periods leading to slower motor speed. To achieve this goal, the Arduino board uses its internal Analog Write() function to generate the PWM signal which in turn produces a PWM output on specific digital pins. Then it forwards this signal through some pins of L298N motor driver module responsible for managing speed of connected dc motor which are named “ENA”. L298N module operates between Arduino and dc motors. It takes in PWM signal from Arduino and utilizes it in regulating power supplied to the motor. In addition, it can give more current and voltage than what Arduino can directly provide making it good for driving motors. The two control pins “IN1” and “IN2” in L298N module can determine the direction of rotation of the motor hence their connections must be correct at all times except when they are not needed.

The Arduino has a total of eight digital pins, which can be used for connecting to L298N. By changing the status of these pins from HIGH to LOW or vice versa, its course gets changed. In software implementation, the Arduino reads the input from a potentiometer (if used) or sets the duty cycle of a PWM signal using predetermined value. This value of duty cycle is then passed as an argument to Analog Write() function where it's converted into equivalent PWM signal and sent to the L298N module. If present, the potentiometer functions as an Analog input to Arduino in order that user could adjust motor speed manually. The potentiometer reading is done through a call on Analog Read() command and then scaled appropriately between 0 and 255 for use in setting up PWM duty cycle. Testing and calibration are also part of the procedure to confirm whether there is any discrepancy in motor response with regards to different values of duty cycles. To achieve precise and adjustable speed control for a DC motor using an L298N motor driver module this methodology combines them with Arduino's PWM output capabilities.



Block Diagram: Arduino Based DC Motor Speed Control Using PWM Technique



Fig(a): Arduino Uno

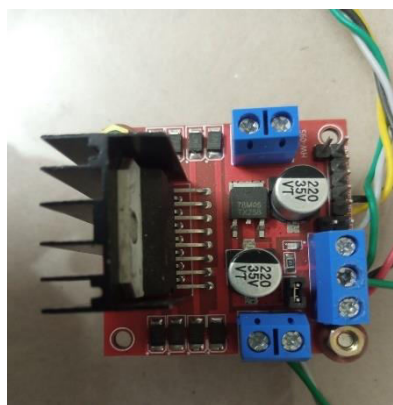
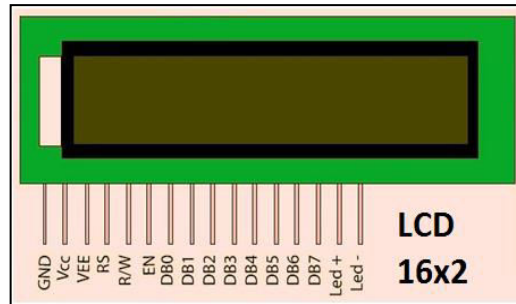


Fig (b): L298N (motor drive)



Fig(C): DC motor



Fig(d): 16X2 LCD Display

IV. EXPERIMENTAL RESULTS

1. The DC engine must rotate at various rates of speed based on the frequency of PWM duty cycles. When PWM is at its minimum duty cycle (e.g., 25%), the motor will rotate very slowly. The motor should increase its rotating speed in proportion to the rise in PWM duty cycle (e.g., 50%, 75%).
2. At a zero percent PWM duty cycle, as there is no current supplied to the motor, it will remain still. There may be some smallest threshold level of PWM duty cycle (e.g. 10-15%) below which, because torque is not enough, the motor might not spin or move inconsistently.



Fig(e): Duty cycle (50 % clock wise)



Fig(f): Duty cycle (50 % anti clock wise)

3. Motor speed due to parameters including friction, load and other factors related to the type of motor may not be linearly proportional to applied PWM duty cycles.
4. There are ranges of some specific pulse width modulation times when you can hear noise or vibrations especially while shifting between different speeds by motors operating on them. Due to inertia and time taken for the motor reach optimum speed, some slight delay or lag might be experienced by the motor when changing PWM duties.
5. Anything from voltage ratings, currents as well as various specifications relating with electric motors could limit maximum achievable engine velocity even with having 100% pulse width modulation
6. At higher speeds and for longer periods during work this machine can generate heat that may interfere with its operation thereby necessitating cooling measures
7. The motor can be made to rotate in the opposite direction by means of the signal polarity change or simply changing it using some logic on Arduino.
8. To ensure closed loop control algorithm is accurate and stable, incorporating feedback mechanisms like encoders and tachometers improve the accuracy and stability in motor speed control.
9. To suit different motor types (e.g. brushless DC motors, stepper motors) or meet specific application demands such as load handling, positioning control etc., more components might need to be included into the experimental setup or changes may be necessary



V. CONCLUSION

The Arduino based DC motor speed control using PWM technique experiment successfully demonstrated the principles of pulse width modulation and its application in controlling the speed of a DC motor. By varying the duty cycle of the PWM signal generated by the Arduino board, the speed of the connected DC motor was effectively modulated. The experimental results confirmed that as the PWM duty cycle increased, the motor speed increased proportionally, with the motor operating at its maximum speed when the duty cycle was set to 100%. Conversely, at lower duty cycles, the motor rotated at slower speeds, with the motor remaining stationary at a 0% duty cycle. It was observed that there was a minimum threshold duty cycle below which the motor did not rotate or rotated erratically due to insufficient torque. Additionally, the motor's response to changes in the PWM duty cycle exhibited a slight delay or lag due to its inertia and the time required to reach the desired speed.

REFERENCES

- [1] Qasim Al Azze, Mohammed Hasan Ali "Design and implement of pulse width modulation with low-cost hardware in the loop" June 2020
- [2] Bashar Abbas Fadheel1, Ali Jafer Mahdi1 , Hussain Fadhel Jaafar2, Muhammad Shahzad Nazir3, Muntaha Sattar Obaid1 and Sarah Hussein Musa1 , 2020
- [3] Ellen Margirahayu*, Junaidi, Gurum Ahmad Pauzi, and Sri Wahyu Suciayati" DC Motor Speed Control System with PWM (Pulse Width Modulation)" 2022
- [4] Raza, W., Nayyar, A., Babu, B. R., & Siddique, N. (2018). Controlling the speed of a DC motor with pulse width modulation. International Journal of Engineering Research and Technology,
- [5] Karthik, S., Malathi, V., & Praveen Kumar, R. (2016). Arduino based speed control of DC motor using PWM technique. International Journal of Innovative Research in Science, Engineering and Technology
- [6] Mohd Zaki, M. F., Mohd Nasir, N. F., Buyamin, S., Ahmad, N. N., & Hashim, N. M. (2018). Arduino-based DC motor speed control using PWM technique. Journal of Telecommunication, Electronic and Computer Engineering (JTEC)
- [7] Sharma, P., & Bhatti, T. S. (2017). Speed control of a DC motor using PWM technique. International Journal of Engineering and Computer Science,
- [8] Rani, A., & Sharma, S. (2020). Design and implementation of Arduino based PWM technique for DC motor speed control. International Journal of Innovative Technology and Exploring Engineering (IJITEE)
- [9] Reddy, K. V., & Reddy, K. V. (2016). Speed control of DC motor using Arduino microcontroller with pulse width modulation technique. International Journal of Research in Engineering and Technology



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details