Electronic Speed Control (ESC) PWM Fullbridge Motor DC Based on Arduino

***Budi Pramono Jati**
Program Studi Teknik Elektro,
Fakultas Teknologi Industri,
Universitas Islam Sultan Agung
Jl. Raya Kaligawe Km. 4 Semarang, Indonesia,
budipramono@unissula.ac.id

Jenny Putri Hapsari
Program Studi Teknik Elektro,
Fakultas Teknologi Industri,
Universitas Islam Sultan Agung
Jl. Raya Kaligawe Km. 4 Semarang, Indonesia,
jenny@unissula.ac.id

Article history: Received Nov 15, 2022 | Revised Feb 06, 2024 | Accepted Feb 09, 2024

Abstract **– The problem with DC motors is speed regulation, adjusted to the needs in the field. The speed of a DC motor can be changed by adjusting the motor voltage. To overcome this, the motor voltage is regulated using Pulse Wide Modulation (PWM). PWM works by changing the duty cycle of the voltage in the form of pulses so that the average voltage can be changed. The solution is to use an Electronic Speed Control (ESC) controller where the pulse voltage duty cycle is realized using a variable PWM generator using an Arduino which is fed to a fullbridge MOSFET inverter. This paper discusses a DC motor ESC using PWM with a full bridge which is used for the inverter process, the aim is to prove the effect of voltage changes on the speed of a DC motor. The model is specified as a DC motor controlled with a fullbridge PWM inverter. PWM ESC consists of three main parts, namely AC to DC converter, fullbridge mosfet inverter, PWM variable signal source. The parameters determined are: change in duty cycle of the square wave signal. The method used is fullbridge PWM in the form of manufacturing design (hardware and software) and testing of DC motor speed controllers with PWM. The ESC parts consist of: fullbridge switching inverter, PWM signal generator, Arduino, Arduino sketch software, SMPS power supply. The ESC is built using 8 130A 200V MOSFETs, with a duty cycle varying from 0 -100%.**

Keywords: ESC, arduino nano, PWM, mosfet full bridge, motor DC.

I. INTRODUCTION

The development of high-performance motor drives is crucial for industrial applications. Motor drives must have good dynamic speed command tracking and load regulation response [1]. One useful device for converting electrical energy into mechanical energy, for example, is the direct current (DC) motor. DC motors are electric motors that require DC voltage to operate. DC motors inherently have high speed and low torque, variable characteristics, and are widely used in variable speed drives. DC motors are commonly used by industries, especially those requiring constant rotational speeds. The advantages of DC motors include ease of speed and load variation control. Forward/backward current or positive/negative voltage determines the direction of rotation of DC motors. Meanwhile, changes in coil voltage determine the speed of DC motors [2][3][4][5][6].

Pulse Width Modulation (PWM) is a modulation technique that manipulates the pulse width without changing the amplitude value. Speed control of DC motors can be achieved by adjusting the PWM signal. The speed of a DC motor can be controlled by varying the density of PWM pulses in an Arduino microcontroller. Pulse width modulation works by rapidly switching power to the motor on and off. DC voltage is converted into a square wave signal [3][5][7][8].

An H-bridge, commonly known as an H-Bridge, is a circuit that can be utilized to control DC motors. Components for controlling the direction of rotation of a DC motor in two directions can use an H-Bridge motor driver transistor or MOSFET. MOSFETs can control the direction of motor rotation using either PWM or TTL logic signal methods (High and Low) [8][9].

Changing the polarity of the voltage supplied to the coil winding is done to reverse the direction of rotation of a DC motor. The H-bridge configuration is a common method used to change the direction of a DC motor. The name H-bridge is given because the circuit resembles the letter "H." ESC (Electronic Speed Control) DC motor drivers can be designed to run DC motors in both clockwise and counterclockwise directions. MOSFETs are easier to operate with voltage compared to bipolar transistor switches driven by current. Therefore, with MOSFET control, its PWM is more efficient, tighter, and faster in response [8][10][11].

MOSFET drivers are the most popular and economical MOSFET driving circuits. The IR2110 IC is a MOSFET driver that has high voltage and power speed IGBTs and independent reference outputs on both the high and low sides [12][13].

The problem of ESC PWM controller can be realized using a variable duty cycle PWM generator with Arduino, whose signal is fed to a switching MOSFET full bridge so that the speed and direction of rotation of the DC motor can be changed. ESC PWM consists of three main parts, namely AC to DC

Converter, switching MOSFET full bridge, and variable duty cycle PWM signal source. The average output voltage of ESC can be adjusted by the switching full bridge, which will control the duty cycle of the MOSFET on/off. PWM generator is a square wave generator, which can adjust the duty cycle.

This research will develop an ESC PWM DC motor device based on Arduino switching full bridge using 8 MOSFETs rated at 200Vdc 130A to generate high power. The device created in this study can also be implemented for electric vehicles. Furthermore, this device can be used to control the speed of DC motors.

Several studies on DC motor control with PWM have been conducted by previous researchers, including: "Design and Implementation of Cheap MOSFET-Based Chopper Driver for DC Motor Speed Control" [1], "DC Motor Speed Control Using Pulse Width Modulation" [3], "Analysis of DC Motor Speed Control: An Overview" [4], "Selection and Implementation of H-Bridge in DC Motor Control" [8], and "Brush DC Motor Control with H-Bridge Driver IC" [11].

II. Basic Theory

A simple DC motor consists of a set of stationary magnets in the stator and an armature with one or more insulated wire windings wound around a soft iron core that concentrates the magnetic field. The ends of the wire windings are connected to the commutator, allowing each armature coil to be energized alternately and connecting the rotating coil to an external power supply through carbon brushes [2], as shown in Figure 1.

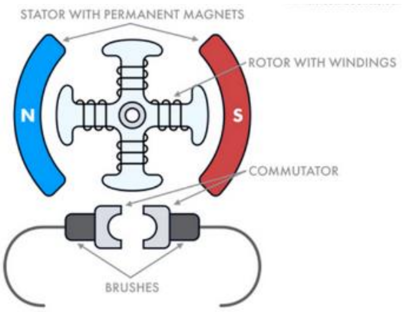

Figure 1. Principle of a DC motor with permanent magnets [14]

The DC motor can be modeled with an equivalent circuit as shown in Figure 2.

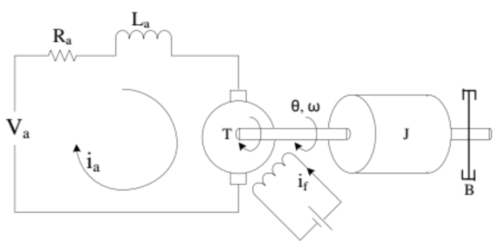

Figure 2. DC motor model [2]

Equation 1 and the block diagram of the DC motor in Figure 3 are derived from the equivalent circuit in Figure 2.

$$V_a(t) - V_{ggl}(t) = L\frac{di_a(t)}{dt} + R_a i_a(t) \quad (1)$$

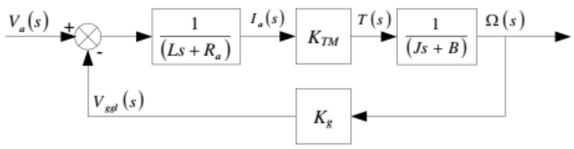

Figure 3. Block diagram of the DC motor model [2]

In MOSFET switching techniques, there are two types: the switching half-bridge, which utilizes only one MOSFET to control PWM switching voltage, and the second type is the switching full bridge, which employs 4 MOSFETs to control the direction of current, allowing the direction of motor rotation to be adjusted forward or backward, as shown in Figure 4.

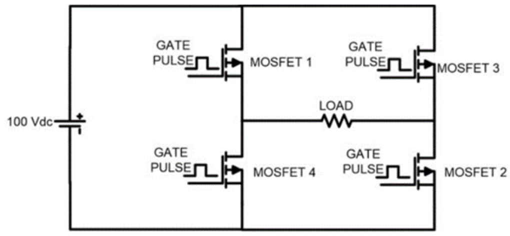

Figure 4. Full Bridge MOSFET Switching [15]

PWM is a modulation technique that can control the average voltage value in electronic devices by rapidly turning power on and off. The ratio of ON time to the signal within one period is the duty cycle, and the average voltage is shown in Figure 5 and Equation 2.

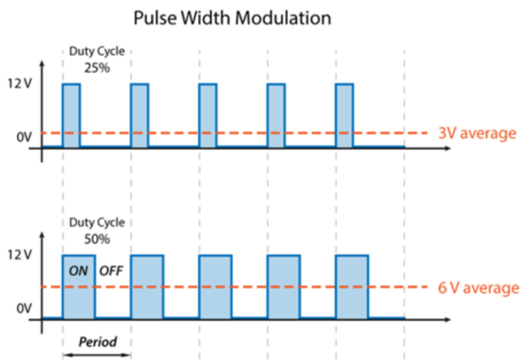

Figure 5. Relationship between duty cycle and voltage [16]

$$V_{out} = duty\ cycle \times V_{in} \quad (2)$$

III. Methods and Design

The DC motor ESC in this study utilizes four switches controlled by Arduino, allowing the direction of motor rotation to be adjusted forward or backward, as shown in Figure 6.

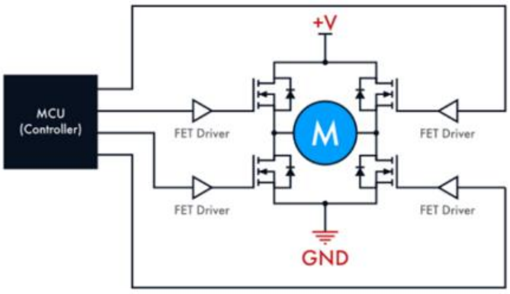

Figure 6. Basic full bridge H-ESC circuit [14]

If you want to provide MOSFET input from the microcontroller, you need something to increase the gate current up to 1A. There are many "logic-level" MOSFETs that can be driven from a 3.3V or 2.5V logic level. A MOSFET driver is used to boost the drive voltage and gate current, in this study using the IRS2110 MOSFET driver IC.

To implement PWM ESC, a block diagram is created as shown in Figure 7.

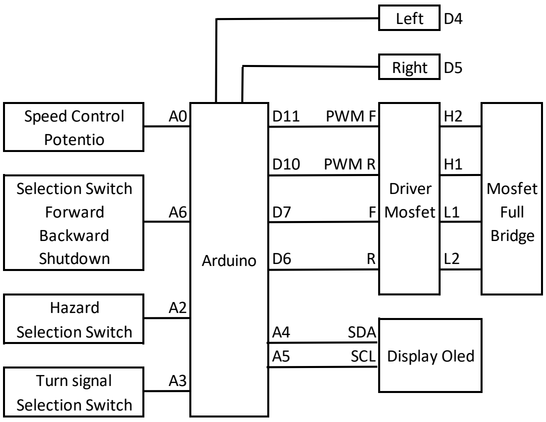

Figure 7. Block diagram of DC motor ESC with Arduino Nano

Figure 7 illustrates the block diagram of the PWM ESC, which consists of: speed control potentiometer, forward and reverse switches, turn signal switches, PWM generator with Arduino, MOSFET driver, and MOSFET full bridge. The PWM signal is synthesized using control functions from the Arduino microcontroller. The number of pins used for control and sensors is one of the crucial factors in selecting a suitable controller. The pin requirements for control in this study are: 4 analog inputs, 2 display controls (SCL and SDA), 4 data outputs, and 2 PWM outputs. The Arduino Nano is the suitable controller in terms of pin requirements, sensors used, and physical compatibility. The Arduino Nano has 8 analog inputs, 14 input-output pins, 6 of which can be used to control PWM signals [17].

The software is written in the Arduino programming language and compiled using the Arduino IDE software. The controller and PWM generator circuit with Arduino are shown in Figure 8.

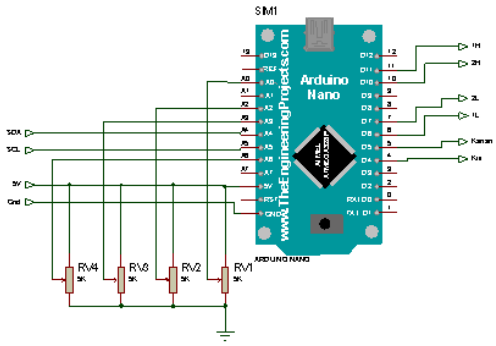

Figure 8. Arduino controller circuit

The flowchart of the Arduino-based DC motor ESC software system as shown in Figure 9.

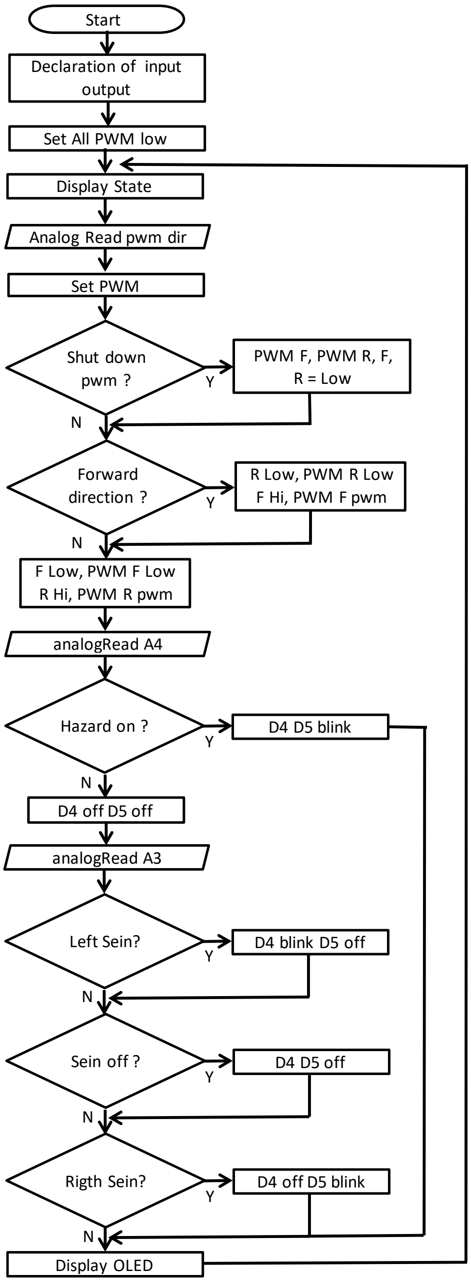

Figure 9. Flowchart of Arduino-based DC motor ESC software

The inputs and outputs to be controlled by the Arduino microcontroller in the DC motor ESC system are: A0 (speed control), A2 (hazard), A3 (turn signal), A4 (SDA), A5 (SCL), A6 (forward/backward control), D10 (PWMF), D11 (PWMR), D6 (reverse), D7 (forward), D4 (left turn signal LED), D5 (right turn signal LED). The total number of required inputs and outputs is 12 pins. This pin count is the reason for using the Arduino Nano.

The driver circuit is shown in Figure 10, and the full bridge MOSFET circuit is shown in Figure 11.

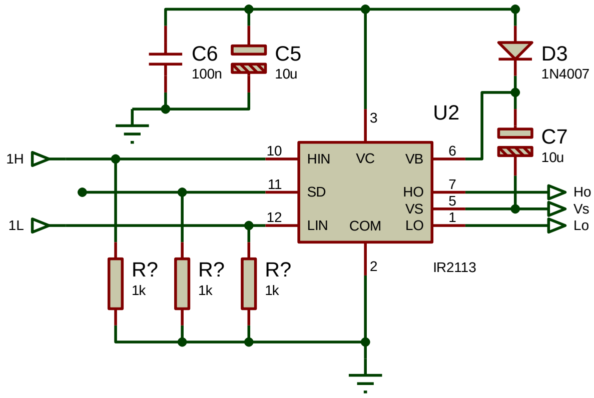

Figure 10. MOSFET driver circuit

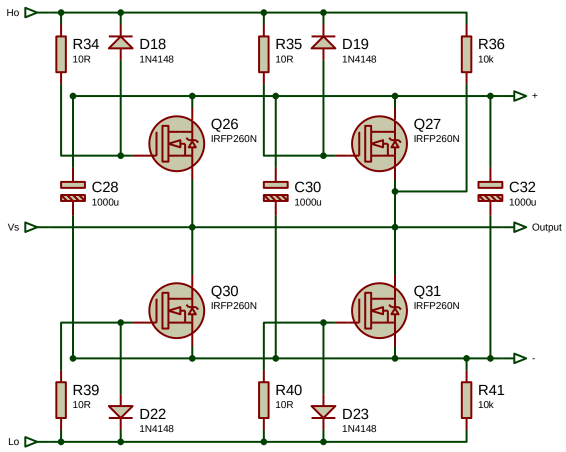

Figure 11. Full bridge MOSFET circuit

The results of testing the device after following several stages are:

1. Testing of the high-side MOSFETs (near Vcc): When gate pins 1H and 2H are supplied with 12V (>5Vdc), the MOSFETs will saturate. When the driver MOSFET inputs 1H and 2H are supplied with 5V, the MOSFETs will saturate. From this test, it is concluded that the MOSFET driver circuit and high-side MOSFETs (near Vcc) can work properly.
2. Testing of the low-side MOSFETs (near ground): When gate pins 1L and 2L are supplied with 12V (>5Vdc), the MOSFETs will saturate. When the driver MOSFET inputs 1L and 2L are supplied with 5V, the MOSFETs will saturate. From this test, it is concluded that the MOSFET driver circuit and low-side MOSFETs (near ground) can work properly.
3. After being connected to the Arduino pins, when the direction of rotation is changed from forward to reverse, there is no overlap switching that could cause a short circuit in the MOSFET circuit.
4. When the PWM duty cycle potentiometer is adjusted, the speed of the DC motor can be controlled.
5. When the control switch is positioned to shut down, the DC motor stops. When the switch is changed to forward or reverse positions, the polarity and direction of the motor can move forward or backward. It is concluded that the forward-reverse and shutdown switches can function normally.
6. When the turn signal switch is positioned left or right, the left or right LED will blink. When the hazard switch is activated, both the left and right LEDs will blink simultaneously. It is concluded that the turn signal and hazard switches can function normally.
7. The results of testing the DC motor ESC show that it can move smoothly and normally, and it can rotate at high RPM. The minimum duty cycle for the PWM of the DC motor is 0%, and the maximum duty cycle is 100%. The DC motor can rotate forward up to 3191 RPM and rotate backward up to 3323 RPM maximum. The cables leading to the motor, cables leading to the power supply unit (PSU), MOSFETs, and DC motor do not become hot.

The research flowchart includes: determining the operating voltage, selecting the controller type, designing the MOSFET driver, designing the full bridge MOSFET switching, creating the software flowchart, designing the ESC software, testing the system to ensure proper operation, as shown in Figure 12.

The testing of PWM ESC is carried out by observing the ESC output voltage and DC motor RPM values for duty cycle values from 0% to 100%. The research module that has been created is shown in Figure 13. The Arduino-based DC motor ESC device consists of: Full bridge switching MOSFETs, MOSFET driver with IC IR2110, Arduino microcontroller, Arduino sketch software, switches for controlling left and right turn signal lights, switch for controlling hazard lights, switch for controlling forward/reverse/shutdown PWM motor rotation direction, and a potentiometer for adjusting motor rotation speed.

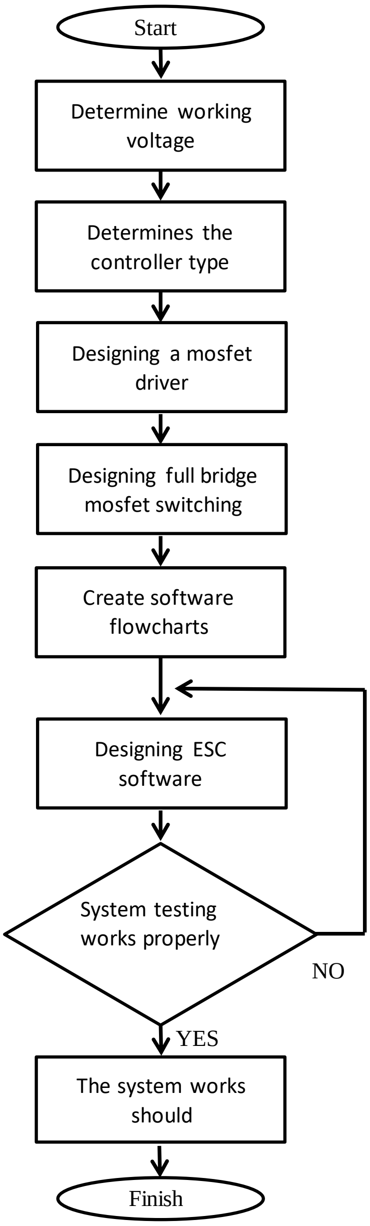

Figure 12. Research flowchart for the design and testing of PWM full bridge DC motor ESC

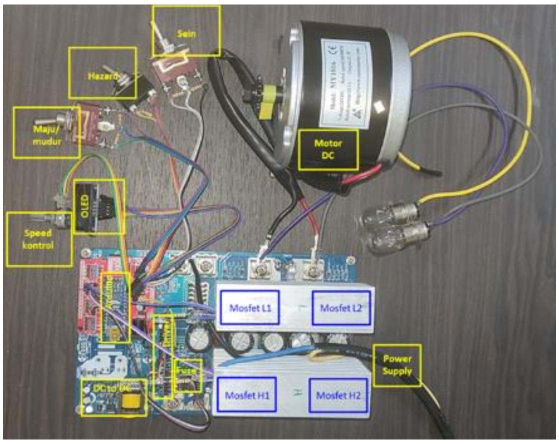

Figure 13. Testing of PWM ESC

IV. RESULTS AND DISCUSSION

Performance testing of the full bridge PWM ESC device was conducted in the laboratory using a main supply of +24 VDC because the motor used is a 24 VDC motor. Testing of the high-side MOSFETs (near Vcc) revealed that when gate pins 1H and 2H were supplied with 12V (>5Vdc), the MOSFETs would saturate, causing the source to be connected to the drain. When the driver MOSFET inputs 1H and 2H were supplied with 5V, the MOSFETs would saturate, indicating that the MOSFET driver circuit and high-side MOSFETs (near Vcc) could work properly.

Testing of the low-side MOSFETs (near ground) showed that when gate pins 1L and 2L were supplied with 12V (>5Vdc), the MOSFETs would saturate, causing the source to be connected to the drain. When the driver MOSFET inputs 1L and 2L were supplied with 5V, the MOSFETs would saturate, indicating that the MOSFET driver circuit and low-side MOSFETs (near ground) could work properly.

The PWM signal generator circuit was tested using an oscilloscope, and the output waveform was a square wave signal whose duty cycle could be adjusted when the potentiometer was adjusted. The average output voltage of the ESC could be adjusted from 0V to Vcc, and the speed of the DC motor could be controlled accordingly.

When the direction control switch was changed, the motor rotation could be adjusted accordingly. When the turn signal switch was changed, the turn signal LED could blink left or right as required. When the shutdown switch was activated, the motor would stop rotating.

The Arduino-based DC motor ESC device has the following specifications: operating voltage 12-80 VDC (maximum MOSFET specification 200VDC), maximum MOSFET current 260A, ESC output voltage 0-Vcc (according to the position of the potentiometer or duty cycle), PWM square wave output waveform, Arduino Nano microcontroller, Full bridge switching MOSFET type, IRFP4668 MOSFET type parallel 2 pieces in 4 sets, forward-reverse rotation direction, navigation LED for left-right turn signals and hazard lights. The DC motor used during testing is a 24 Volt 250 watts motor.

The results of testing the PWM ESC for duty cycle changes from 10-100% yield the duty cycle vs Vout as shown in Table 1, and the duty cycle vs RPM as shown in Table 2.

Table 1. Duty cycle vs V_{out}

Duty cycle (%)	V_{out} Forward (V)	V_{out} Reverse (V)
10	2.3	2.3
20	4.6	4.6
30	7	7
40	9.5	9.5
50	11.8	11.8
60	14.2	14.2
70	16.4	16.5
80	18.9	18.9
90	21.3	21.2
100	23.8	23.8

Table 2. Duty cycle vs rpm

Duty cycle (%)	Speed Forward (rpm)	Speed Reverse (rpm)
10	782	775.7
20	1588	1598
30	2121	2181
40	2477	2531
50	2695	2758
60	2842	2930
70	2946	3046
80	3026	3127
90	3090	3200
100	3191	3323

The graph of duty cycle vs Vout as shown in Figure 14, and the graph of duty cycle vs RPM as shown in Figure 15.

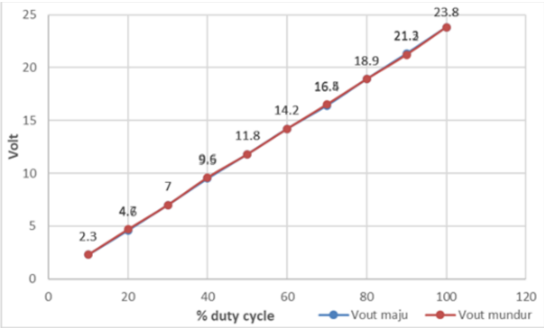

Figure 14. Graph of duty cycle vs V_{out}

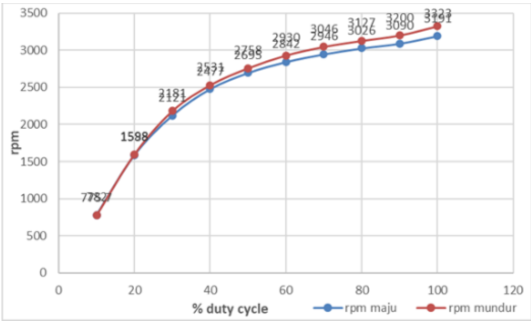

Figure 15. Graph of duty cycle vs rpm

Referring to Figure 7, the data in Table 1, the data in Table 2, Figure 14, and Figure 15, the output results of the PWM signal from the PWM ESC connected to the DC motor were obtained. The PWM is adjusted using a potentiometer. The testing results show that the DC motor ESC can move the motor forward and backward, and the motor speed can be adjusted. The operational duty cycle range of the DC motor ESC is set from 0% to 100%. At a duty cycle of 10%, the DC motor RPM is low, while at a duty cycle of 100%, the DC motor RPM is high. Specifically, at a duty cycle of 10%, the DC motor rotates forward at 782 RPM, rotates backward at 775 RPM, while at a duty cycle of 100%, the DC motor rotates forward at 3191 RPM and rotates backward at 3323 RPM. The cables leading to the motor, cables leading to the PSU, and the MOSFETs do not become hot.

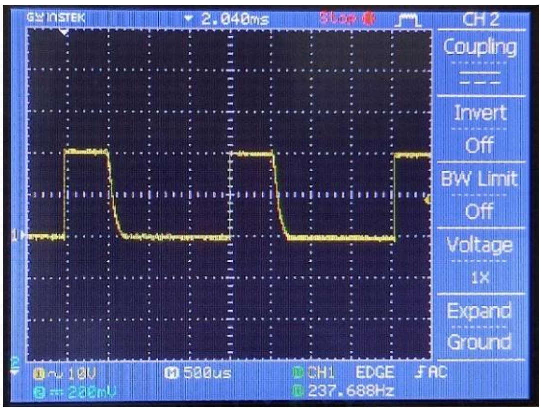

Figure 16. Test results of PWM signal at 25% duty cycle

Figure 17. Measurement results of speed at 25% duty cycle

Figure 18. Measurement results of speed at 100% duty cycle

V. Conclusion

The full bridge DC motor ESC is capable of controlling the speed of the DC motor. The duty cycle of the full bridge ESC is directly proportional to the output voltage of the ESC, approaching linearity, but not linear with respect to RPM. At a duty cycle of 10%, the DC motor rotates forward at 782 RPM, rotates backward at 775 RPM, while at a duty cycle of 100%, the DC motor rotates forward at 3191 RPM and rotates backward at 3323 RPM.

References

[1] A. Islam, A. K. M. Shamim, H. Ullah, M. Arif, and S. Bhuiyan, "Design and Implementation of a Low-cost MOSFET Based Chopper Drive DC Motor Speed Control," *J. Sci. Technol.*, vol. 9, no. January, pp. 4–8, 2017.

[2] K. Fathoni and F. Suni, "Perancangan Kendali Kecepatan Motor Arus Searah Menggunakan Metode Root Locus," *J. Tek. Elektro*, vol. 8, no. 2, pp. 39–42, 2016.
[3] R. Kumar, M. Ray, S. Basak, and T. Ray, "Speed Control of Dc Motor Using pulse Width Modulation," 2019.
[4] S. . Labaran and H. Bashir, "DC Motor Speed Control Analysis : An Overview," *3Rd Natl. Eng. Conf. Bridg. Gap between Acad. Ind.*, no. December 2018, 2018.
[5] D. Setiawan, "Sistem Kontrol Motor DC Menggunakan PWM Arduino Berbasis Android System," *J. Sains, Teknol. dan Ind.*, vol. 15, no. 1, pp. 7–14, 2017.
[6] A. R. J. Wiriawan and A. Irawan, "Pengaturan Kecepatan Motor DC dengan Kontrol Proporsional Integral Derifatif (PID) Berbasis Lab View," *Telekontran J. Ilm. Telekomun. Kendali dan Elektron. Terap.*, vol. 4, no. 2, pp. 13–24, 2016, doi: 10.34010/telekontran.v4i2.1886.
[7] A. S. S. Azhar and E. Ariyanto, "Pengendalian Kecepatan Putaran Motor DC Pada Penyangrai Kopi Menggunakan Pwm Berbasis Atmega 16," *Gema Teknol.*, vol. 19, no. 1, p. 12, 2016, doi: 10.14710/gt.v19i1.21957.
[8] D. Phan, "Selecting and Implementing H-Bridges in DC Motor Control," 2011.
[9] E. C. Sunarto and B. Yulianti, "Rancang Bangun Prototipe Alat Angkut Helikompter Berbasis Arduino, " *TESLA*, vol. 20, no. 2, p. 157, 2018, doi: 10.24912/tesla.v20i2.2992.
[10] T. Özer, S. Kivrak, and Y. Oguz, "H Brıdge DC Motor Drıver Desıgn and Implementatıon with Usıng dsPIC30f4011," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 6, no. May, 2017.
[11] Rohm Semiconductor, "Controlling DC Brush Motors with H-bridge Driver IC," 2018.
[12] L. Balogh, "Fundamentals of MOSFET and IGBT Gate Driver Circuits," 2018. [Online]. Available: https://www.ti.com/lit/ml/slua618a/slua618a.pdf?ts=1611669659975&ref_url=https%253A%252F%252Fwww.ti.com%252Fww%252Fen%252Fpower-training%252Flogin.shtml.
[13] International Rectifier, "Datasheet Gate Driver Ir2110," 2019.
[14] A. Solovev, "What It Is, Design Principles & Circuit Examples," *Integra Sources*, 2021. https://www.integrasources.com/blog/dc-motor-controller-design-principles/ (accessed Nov. 06, 2022).
[15] R. Kansagara, "Single Phase Half Bridge and Full Bridge Inverter using MATLAB," *Circuit Digest*, 2019. https://circuitdigest.com/electronic-circuits/single-phase-half-bridge-and-full-bridge-inverter-circuit-using-matlab (accessed Nov. 06, 2022).
[16] Dejan, "Arduino DC Motor Control Tutorial – L298N | PWM | H-Bridge," 2020. https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/ (accessed Nov. 06, 2022).
[17] Arduino, "Arduino Nano," vol. 9210, no. 662, pp. 3–8, 2019, [Online]. Available: https://www.arduino.cc/.