Polytechnic University of Puerto Rico
San Juan Campus

Electrical and Computer Engineering Faculty

Microprocessor Interfacing Laboratory

Laboratory 5: DC Motor Control and Direction

Kelvin Figueroa Figueroa - 108946
Rafael Gonzalez Cartagena - 90352

Professor: Dr. Roman Lopez

April 27, 2022

Contents

List of Tables

List of Figures

1 Objective

This laboratory is focused on powering a DC motor using Arduino Mega to move the motor. It is important to remember that the motor speed is controlled by controlling the driving voltage, where the higher the voltage the higher the speed. During this lab we will also be working with increasing and decreasing the speed of the motor according to its speed percentage and we will also be working to make the motor rotate in both directions when it reaches its minimum speed. Additionally, it will also be using the serial monitor to display the amount of voltage/speed used to drive the motor.

In order to make this application possible, make sure to have the following materials,

1. Arduino Mega ADK
2. LCD shield
3. DC motor module

2 Procedures

This laboratory consists of four parts, where the first part (refer to section 2.1) consists of making the DC motor to turn to the right, the second (refer to section 2.2) to make the DC motor turn to the left, in the third part (refer to section 2.3) we make the motor turn to both sides changing direction when the speed reaches zero and at the end (refer to section 3) we print the value of the speed.

See table 2 to see the Arduino's Pinout

Table 1: Arduino and Dc motor PinOut

DC motor Pins	**Arduino Pinout**
IA	Pin 12
IB	Pin 13
VCC	VCC
GND	GND

2.1 Rotate Right DC motor

In this first part, the DC motor is made to rotate to the right and increase its speed as the seconds pass. It is important to mention that in order to make a right turn, the IA pin of the DC motor module must be HIGH and the IB pin must be LOW, as well as the lowest speed would be 50V, in order to be able to use the Serial Monitor later, and the highest would be 250V.

1. First pins IA and IB are initialized as OUTPUT in the Void Setup() function, see figure below,

```
void setup() {
  pinMode(motorA, OUTPUT);
  pinMode(motorB, OUTPUT);
  Serial.begin(9600);
}
```

Figure 1: Setup Function

2. Second the Void Loop() function is created. Inside the function the following variables are created to rotate the DC motor,
   (a) speeding, which is a boolean variable set to TRUE, in order to specify if the rotation is increasing.
   (b) rotateRight, which is a boolean variable set to TRUE, in order to specify if the motor rotates Right or Left.
   (c) motorSpeed, which is an int variable with the minimum value 50.
3. Within an if statement the function analogWrite() is used to rotate the motor Right by letting Pin IA with the initial speed (50V) and Pin IB as LOW and increasing the motorSpeed by one.

2.2 Rotate Left DC motor

Similar to the first part, we are looking for the motor to turn to the left, increasing the speed over the seconds. It is important to mention that the variable "rotateRight" being FALSE makes the motor turn to the left.

1. Combining the process of the first part, we added an else where the analogWrite() function is used to rotate the motor to the left, leaving the IA pin as LOW and the IB pin with the initial speed (50V) and increasing the speed by one.

2.3 Rotate Right and Left DC motor

This section is a little more complex, since it seeks to rotate the motor to the right to reach the maximum speed, at the time of reaching this speed the motor must lower the speed to 50V and then increase the speed by rotate to the left.

1. Using an if statement with conditions where the motorSpeed is less than 250 but higher than 50, and is speeding to the maximum velocity, the if/else from the previous sections are implemented. See figure 2

```
if((motorSpeed >= 50 && motorSpeed < 250) && speeding){
  if(rotateRight){
    analogWrite(motorA, motorSpeed);
    analogWrite(motorB, LOW);
  }
  else{
    analogWrite(motorA, LOW);
    analogWrite(motorB, motorSpeed);
  }
  motorSpeed = motorSpeed + 1;
  i++;
}
```

Figure 2: Increase motorSpeed to 250V

2. Then a second if (else if) with conditions where the motorSpeed is less or equal to 250 but higher or equal to 50, and the variable speeding is set to FALSE. After the else, the motorSpeed decreases by one. See figure 3

```
else if((motorSpeed >= 50 && motorSpeed <=250) && speeding == false){
  if(rotateRight){
    analogWrite(motorA, motorSpeed);
    analogWrite(motorB, LOW);
  }
  else{
    analogWrite(motorA, LOW);
    analogWrite(motorB, motorSpeed);
  }
  motorSpeed = motorSpeed - 1;
  i--;
}
```

Figure 3: Decrease motorSpeed to 50V

At the end of the process an else is included to change the rotation of the DC motor, by changing the variable rotateRight from TRUE to FALSE. See figure 4.

```
else{
  motorSpeed = 50;
  rotateRight = !rotateRight;
  i = 0;
}
```

Figure 4: Change rotateRight variable

2.4 Rotating and displaying velocity of the DC motor

In this section the processes of the previous sections are implemented and the aim is to print on the screen the speed obtained by the motor when rotating. When printing on the screen we also want to mention which way the motor is rotating.

1. Before starting the process, create the following variables,

(a) printed, which is a boolean variable set to TRUE, in order to print a message when there is a rotation to the right.

(b) i, which is a short variable used to store the value of the increased or decreased speed of the motor. This variable is used later to calculate the voltage usage of the DC motor.

2. Using an if statement with conditions where the DC motor rotates to right and the motorSpeed is equal to 50V, and the variable printed is set to TRUE, the serial monitor will print the direction of the fan. At the end, the variable "printed" is set to false in order to not print the sentence again. Refer to figure 5.

```
if(rotateRight && motorSpeed == 50 && printed){
  Serial.println("\nThe current direction of the fan is: RIGHT ROTATION.\n");
  printed = false;              // Variable is set to false so to not out
}
```

Figure 5: Print: Rotate Right

3. Then an "else if" is used with the conditions where "rotateRight" and "printed" are FALSE and the motor speed is 50V, the serial monitor will print the direction of the fan. At the end, the variable "printed" is set to TRUE in order to not print the sentence again. Refer to figure 6.

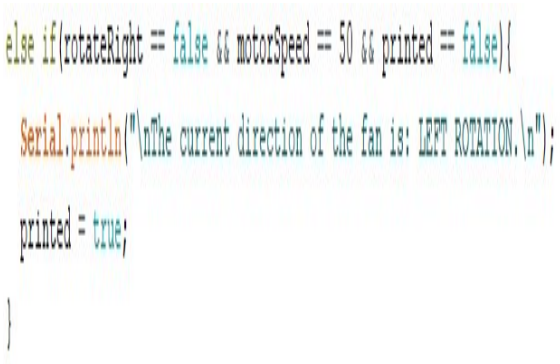

Figure 6: Print: Rotate Left

4. To calculate the percentage of use, an if condition is met when (i/20) is equal to zero. When evaluating the module, if it gives zero, the serial monitor prints a message with the percentage of use divided by half (i/2) and then prints the speed of the DC motor. See figure 7.

```
if(i % k == 0){
  float tmp = ((float)(motorSpeed-50)/(205))*100;
  Serial.print("The current speed voltage value (V) of the Fan Motor Usage ");
  Serial.print(i/2);
  Serial.print("% is: ");
  Serial.println(motorSpeed);
}
```

Figure 7: Print percentage of use and velocity of the DC motor

It should be noted that the descriptions of all the steps provided in this section were obtained with the laboratory's documentation [1].

3 Results

In this section, it is documented the various results of the procedures related with **Laboratory 5: DC Motor Control and Direction** experiment. As such, they are found in the following *subsections*: **The C Embedded Programs of the Results** and **The Videos of the Results**.

3.1 The C Embedded Programs of the Results

The following lines of *C Embedded* code illustrate, in essence, the program that was developed for the realization of this experiment. It is imperative to note that the following program was constructed per the guidelines presented in this website [2].

DC Motor Control and Direction Program

```
1 const int motorA = 12;              // Pin for motor A
2 const int motorB = 13;              // Pin for motor B
```

```
3  const int k = 20;                    // A constant variable
       which will determine which values of the speed voltage will
       be "printed" on the screen
4
5  void setup() {
6    pinMode(motorA, OUTPUT);
7    pinMode(motorB, OUTPUT);
8    Serial.begin(9600);
9  }
10
11 void loop() {
12   static bool rotateRight = true; // Boolean variable to
         determine if the motor is rotating to the right. If not,
         it rotates to the left.
13   static bool speeding = true;    // Boolean varibale to
         determine if the motor is speeding up. If not, it is
         slowing down.
14   static bool printed = true;     // Boolean variable to
         determine if specific values have already been printed on
         the screen.
15   static int motorSpeed = 50;     // Counter variable for the
         current voltage values at one specific time of the motor.
16   static short i = 0;
17
18   /*
19    * If condition to output on the screen whenever the motor
        will rotate to the right. It should be noted that such a
        message must only be outputed once.
20    * Hence:
21    *  - rotateRight must be true
22    *  - motorSpeed must be equal to the lowest value for which
        the motor will start to rotate. (It was found to be 50,
        as in Volts).
23    *  - printed must be true since its boolean (Its true for
        when the motor start to rotate to the right)
24    */
25   if(rotateRight && motorSpeed == 50 && printed){
26     Serial.println("\nThe current direction of the fan is:
         RIGHT ROTATION.\n");
27     printed = false;              // Variable is set to false
```

```
      so to not output again the message for the time being.
28  }
29  /*
30   * If condition to output on the screen whenever the motor
       will rotate to the left. It should be noted that such a
       message must only be outputed once.
31   * Hence:
32   *  - rotateRight must be false
33   *  - motorSpeed must be equal to the lowest value for which
       the motor will start to rotate. (It was found to be 50,
       as in Volts).
34   *  - printed must be true since its boolean (Its false for
       the case in which the motor starts rotating to the left)
35   */
36  else if(rotateRight == false && motorSpeed == 50 && printed
      == false){
37    Serial.println("\nThe current direction of the fan is: LEFT
        ROTATION.\n");
38    printed = true;
39  }
40  /*
41   * If condition to output both the percentage and its
       associated voltage usage value of the motor.
42   */
43  if(i % k == 0){
44    Serial.print("The current speed voltage value (V) of the
        Fan Motor Usage ");
45    Serial.print(i/2);
46    Serial.print("% is: ");
47    Serial.println(motorSpeed);
48  }
49
50  /*
51   * If condition to enable the motor to accelerate in either
       right or left direction. It should be noted that such a
       motor will accelerate in the range of 50 <= motorSpeed <
       250.
52   * Furthermore:
53   * - The value of 50 comes from the fact that the DC motor
       needs 50 Volts so to start rotating.
```

```
54     * - The value of 250 comes from the fact that the DC motor's
           maximum speed is near the 250 Volts.
55     */
56    if((motorSpeed >= 50 && motorSpeed < 250) && speeding){
57      if(rotateRight){
58        analogWrite(motorA, motorSpeed);
59        analogWrite(motorB, LOW);
60      }
61      else{
62        analogWrite(motorA, LOW);
63        analogWrite(motorB, motorSpeed);
64      }
65      motorSpeed = motorSpeed + 1;
66      i++;
67    }
68    /*
69     * This else if condition follows the same steps of the
           previous if; however, insetad of accelerating the motor,
           it just desacelerates it.
70     */
71    else if((motorSpeed >= 50 && motorSpeed <=250) && speeding
          == false){
72      if(rotateRight){
73        analogWrite(motorA, motorSpeed);
74        analogWrite(motorB, LOW);
75      }
76      else{
77        analogWrite(motorA, LOW);
78        analogWrite(motorB, motorSpeed);
79      }
80      motorSpeed = motorSpeed - 1;
81      i--;
82    }
83    /*
84     * Whenever both of the previous if conditions fail, the
           motorSpeed and i will be reset to 50 and 0,
           respectively, and the direction of the motor will become
           that of its oposite.
85     */
86    else{
```

```
87      motorSpeed = 50;
88      rotateRight = !rotateRight;
89      i = 0;
90    }
91    /*
92     * If condition change the speeding varibale (if the motor
           will speed up or slow down) whenever the motorSpeed of
           the motor is either 250 or less than 50.
93     */
94    if(motorSpeed == 250 || motorSpeed < 50){
95      speeding = !speeding;
96    }
97    delay(50); // Delay to make motor rotate for more time.
98  }
```

3.2 The Videos of the Results

In relation with the previous subsection, the following table, i.e., **Table 2**, illustrates the names of the video files related with the already illustrated *C Embedded* program:

Table 2: Video Names of Laboratory Results

Video Names of Laboratory Results
Video_1_Motor
Video_2_Motor

4 Analysis of Results

Since the experiment only tasked with both the construction of a DC Motor circuit and its implementative *C Embedded* program, it was found to be, at least, short. With that being said, it should be noted that completing it, however, was arduous given the following two (2) complications:

- After many deliberations, it was found that the constructed DC Motor circuit needed an external power supply, of which it was simply implemented unto itself. However, albeit this drawback, a new way was

found for the creation of a simpler version of the same circuit without such an addition.

- The *C Embedded* Program that was used for this laboratory, which included lines with *pulse-related* mathematical equations, was completely modified since not once it was possible to implement it properly with the already described circuit.

5 Conclusion

In conclusion, it was possible to construct and control a DC Motor Circuit with the aid of both a *C Embedded* program and an *Arduino Mega2560 ADK Board* for the changing of direction and speed of itself.

References

The following *References* related with this document are in the IEEE Format, as shown below:
[1] Roman Lopez, Phd., "Laboratory 5: DC Motor Control and Direction". Mega. 2022. [PDF]
[2] I. Lohs, “First Test: Super StarterKit from Elegoo - Motor 3-6V DC,” Arduino Project Hub. [Online]. Available: https://create.arduino.cc/projecthub/ingo-lohs/first-test-super-starterkit-from-elegoo-motor-3-6v-dc-5b199d. [Accessed: 27-Apr-2022].