Lab 3. PWM and Speed Control of DC Brushed Motors

Dr. X. Li
Email: xhli@citytech.cuny.edu

Dept. of Computer Engineering Technology
New York City College of Technology
City University of New York

In this lab, we will learn Pulse Width Modulation (PWM) and practice using Arduino to generate PWM signals to control the speed of a DC brushed motor.

Electric motors (DC or AC) are broadly used in virtually everywhere to provide motion mechanisms, such as in automotives, household appliances, computers, printers, digital cameras and so on. They are constructed as either brushed or brushless motors. DC brushed permanent magnet motor is the most common one, which is so-called “regular DC motor”.

PWM is a very important technique that has been widely used in many applications, such as wireless communications, industrial automations, audio amplification, etc. By PWM, the signal is transmitted via a series of periodic pulses (i.e., ON/OFF sequences) at a certain frequency (usually quite high). This is equivalent to turn the switch between signal transmitter and the load on and off at a fast pace. The core principle of PWM (Pulse Width Modulation) is essentially to adjust the pulse’s width for controlling the average voltage delivered to the load. That is the very reason why it is named as “Pulse Width Modulation”. The longer the ON portion in the pulse is, equivalently the longer the switch is on compared to OFF, the higher the average voltage supplied to the load is.

According to the Speed-Torque-Voltage curve of DC motors which I explained in the lecture, at a fixed torque, the motor’s speed can be adjusted by changing its supply voltage. One may intend to use potentiometer to achieve this. But it is literally impossible to variably change supply voltage in a large range by using potentiometers. That is where PWM kicks in. By using PWM, we do not directly change the supply voltage applied to the motor, but change the duration of the voltage applied on the motor at a fast enough pace; consequentially the average voltage to the motor is changed. That is how PWM is used to control motor speed.

Note: Please read the above two paragraphs thoroughly, and make sure that you understand why and how PWM works.

Arduino has embedded feature and library function, "`analogWrite( )`", for the users to generate PWM signals via its PWM ports. The parameters of the PWM signal generated by Arduino can be adjustable for custom applications. This enables Arduino to be an ideal choice for the development of motion control applications in which speed, position, or just simple ON/OFF motion is required.

Devices and Materials Needed:

An Arduino I/O Board (Uno is recommended)
A DC brushed motor, a battery for the motor or power supply
A general purpose NPN transistor (e.g., 2N2222 for up to 500mA collector current)
A Diode 1N4001
Jumper cables and various passive components

Optional: Indicator LED

Arduino Functions Used in this lab:

analogWrite( )
Serial.begin( ), Serial.print( ) and Serial.println( )
Serial.available( ) and Serial.read( )
delay( )

Procedures:

1. Before you start the lab, please read the first page carefully and make sure that you clearly understand PWM.
   In your lab report, use your own words to explain what PWM is and how to use PWM signal to control DC motor speed.
2. Refer to the circuit schematics shown in the lecture, draw a motor drive circuit. Include the circuitry schematics in your lab report (no hand-drawn circuitry schematic allowed).
3. Choose an analog port on Arduino Uno board for outputting PWM signal. Wire up your motor driver circuit, and connect it to Uno.
   CAUTION: Check the datasheet of your transistor first! If you use a small power BJT transistor such as 2N2394, be noted that its maximum collector current (Icmax) is only 300mA. Check your motor datasheet to see if it is suitable to use the transistor.
4. If you use power transistor to drive the motor: Connect the PWM output port to the Base pin of power transistor through an 1kOhm resistor.
   If you use MOSFET to drive the motor: Connect the PWM output port you select in Step3 to the Gate pin of MOSFET.
5. Connect Uno to your computer, run Arduino IDE. Write a sketch to generate a PWM signal through the port you select. Note: You can refer to the sample code shown during the lecture.
6. Write a sketch, compile and correct it until the compilation succeeds. Upload the

sketch.

7. Test the sketch: Connect the PWM signal to an oscilloscope (do NOT disconnect PWM signal from the motor circuit). Modify the parameter of function `analogwrite( )` in your sketch, observe how the duty cycle of PWM signal changes on oscilloscope, and observe how the motor speed changes correspondingly.

   CAUTION: If you use a small power transistor, you need to pay close attention at the current drawn by the motor. You may not drive the motor speed too high.

   Select three different parameters for `analogWrite( )` and make the motor run at three different speeds (high/medial/low). At different motor speeds, record the PWM waveforms, find the duty cycles from the waveforms displayed on oscilloscope, and finish the following table. Include the table and PWM waveforms in your lab report.

   Note on how to find the duty cycle of a PWM waveform on oscilloscope: Zoom in the waveform by decreasing the horizontal scale of the oscilloscope display. Make sure you have at least a period of waveform shown on the display. Read the time durations of the “ON” pulse and “OFF” pulse in the waveform. Then you can simply use the definition of duty cycle to find the duty cycle. Reminder: duty cycle is a percentage number and has no any unit!

   Note on how to record the PWM waveforms: If the oscilloscope you are using has a USB interface, you can simply save the waveform data to your USB flash drive. Then Use the saved data to reconstruct the waveforms (for example by MS Excel). Otherwise, you can take a picture of the waveforms on the oscilloscope display, include the pictures in the report.

Motor Speed (High/Mid/Low)	Code of “analogWrite( )”	Calculated Duty Cycle (%) from the code	PWM Waveforms	Measured Duty Cycle (%) from the waveform
Low				
Medial				
High				

Note: include a sample calculation of duty cycle in your lab report!

8. Include your complete and correct sketch in the lab report.

Due Date of Lab Report: TBA on the Blackboard

Please submit your lab report via the Blackboard ONLY.

Reminder: Midterm Proposal to Due! See the Blackboard for deadline.

Do not make any figure too large in your lab report. Half page is the max size for any figure. Regarding to format of lab report, a "Lab Report Requirements" has been posted on the Blackboard. Please read it carefully!