LABORATORY 4

DC AND SERVO MOTOR CONTROL CIRCUITS

OBJECTIVES

1. To study the Pulse Width Modulation (PWM) DC motor control.
2. To build and test the Function Generator PWM control circuit.
3. To build and test Arduino –controlled DC motor.
4. To study the PWM control principles of a Servo Motor
5. To build and test Arduino –controlled Servo Motor circuit.

INFORMATION

1. DC Motor

A DC motor consists basically of two parts: the stationary body of the motor called the "Stator" and the inner part which rotates producing the movement called the "Rotor". For D.C. machines the rotor is commonly termed the "Armature". A simplified diagram of the DC Motor is shown in Figure 4.1.

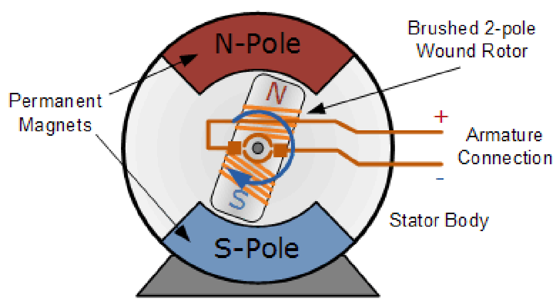


Figure 4.1. A simplified diagram of the DC Motor

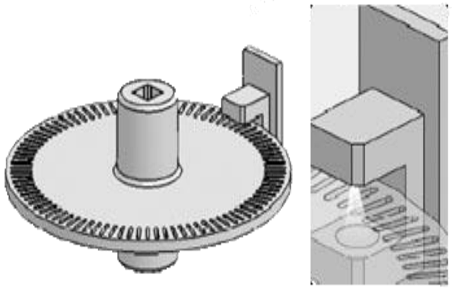
- The current flowing within the rotor coils induces a magnetic field with north and south poles. This magnetic field is repelled or attracted by the stator's permanent magnets producing a rotational movement of the armature around the motors central axis.
- The armature is connected to a voltage source through two semicircle contacts on the shaft. As the armature rotates and aligns itself with the permanent field, the semicircle contacts exchange positions, reversing the voltage and flipping the magnetic field. This continues the rotational motion.
- The rotational speed of a DC motor depends upon the interaction between two magnetic fields, one set up by the stator's stationary permanent magnets and the other by the armatures rotating electromagnets and by controlling this interaction we can control the speed of rotation.

2. Optical Shaft Encoder

A shaft encoder is used to determine the speed and direction of rotation of a shaft, motor, etc.

2.1. Single output encoder

A single output encoder works by shining light onto the edge of a disk outfitted with evenly spaced slits around the circumference. As the disk spins, light passes through the slits and is blocked by the opaque spaces between the slits. The encoder then detects how many slits have had light shine through, and in which direction the disk is spinning. The encoder we use is shown in Figure 4.2. and it generates 90 pulses per revolution. The angular resolution is therefore 4 degrees per pulse.



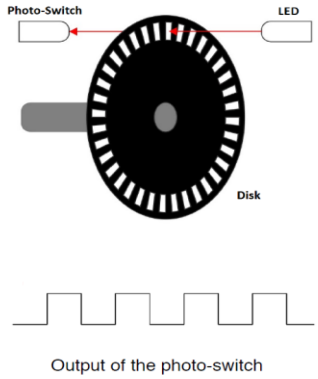


Figure 4.2. Optical Shaft Encoder

- The most common configuration uses a single track.
- When the disk rotates, the photo-switch detects transitions in the level of light and generates electrical pulses.
- Counting the pulses enables the user to track and determine the speed of rotation (RPM).

2.2. The Incremental Quadrature Encoder

- It has two photo-switches (A and B) that are offset by quarter of a cycle – Figure 4.3.
- The phase difference of the resulting signals indicates the direction of rotation – Figure 4.4.

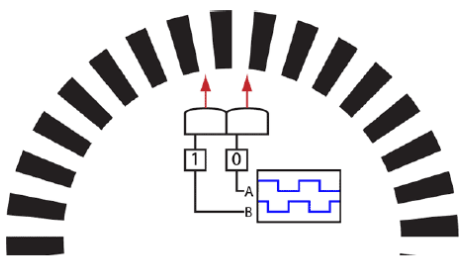
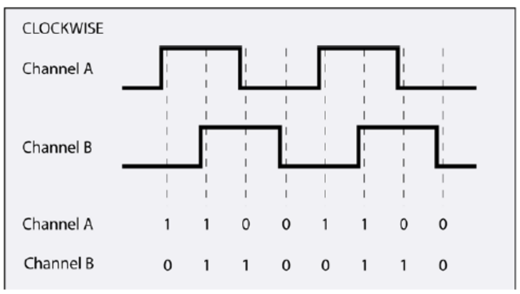
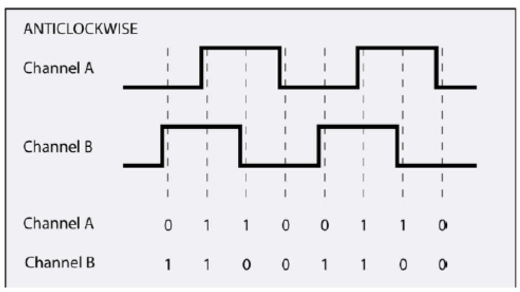


Figure 4.3.Quadrature Encoder diagram





Clockwise rotation: B switches from 0 to 1 when A=1

Counterclockwise rotation: B switches from 0 to 1 when A=0

Figure 4.4. Determining the direction of the rotation using quadrature encoder

- By counting the rising and falling edges of Channel A and Channel B the quadrature encoder resolution can be four times as much as a single output encoder, as it is shown in Figure 4.5.

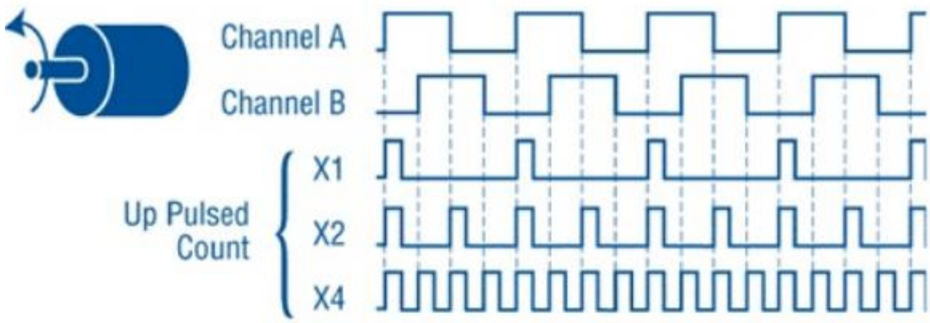


Figure 4.5. Enhancing the resolution

- Resolution X1 – Counts the rising edges of Ch. A, i.e. number of pulses per revolution

  X1= N
- Resolution X2 – Counts the falling edges of Ch. A;

  X2 = 2*X1
- Resolution X4 - Counts the rising and falling edges of both Ch. A and Ch. B

  X4 = 4*X1

3. Pulse Width Modulation speed control for DC Motor

3.1.PWM (Pulse Width Modulation) speed control works by driving the DC motor with a series of "ON-OFF" pulses and varying the duty cycle of the pulses while keeping the frequency constant, thus changing the Average Voltage applied to the Armature, as it is shown in Figure 4.6. The duty cycle is the fraction of time that the output voltage is "ON" compared to when it is "OFF".

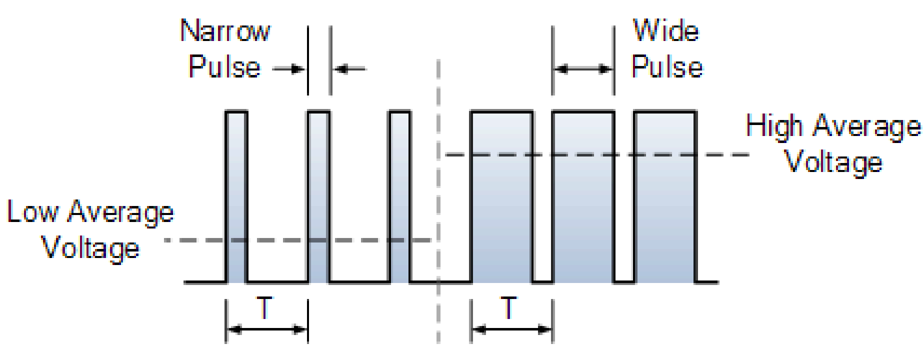


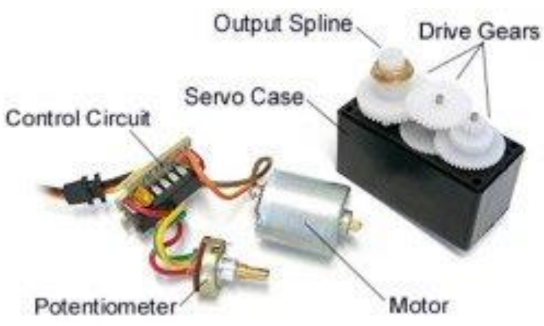
Figure 4.6. PWM (Pulse Width Modulation) speed control

4. Servo Motor

A **servomotor** is a rotary or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable DC motor coupled to a sensor for position feedback.

The very simplest servomotors use position-only sensing via a potentiometer. Using the voltage of the potentiometer tap, the position of the motor is compared to the set position, the external input of the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction to bring the output shaft to the appropriate position. As the positions approach the same value, the error signal reduces to zero and the motor stops.

The simple servo motor package is shown in Figure 4.7.



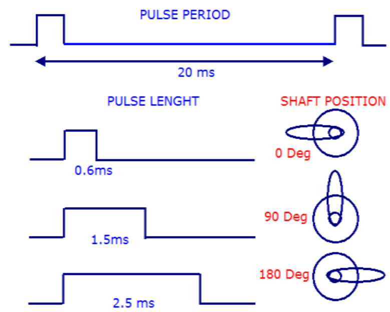


Figure 4.7. Servo Motor

Figure 4.8. Servo Motor Control signals

The chip inside the Servo contains a timer that produces pulse signals from the potentiometer. These signals are similar to the input control pulses, shown in Figure 4.8. These two pulse signals (the ones you are sending and the ones generated by the potentiometer) are fed into a pulse width comparator. This comparator produces the signals indicating which direction the motor should turn in. These are fed into an H-bridge circuit to drive the motor.

EQUIPMENT

1. Digital multimeter BK PRECISION 2831D
2. Digital oscilloscope Tektronix TDS 210
3. Function Generator Wavetek FG3B.
4. PROTO-BOARD PB-503 (breadboard)
5. MOSFET IRF540
6. Arduino Uno board
7. VEX DC motor
8. VEX Optical Encoder
9. Servo Motor

PRE-LABORATORY PREPARATION

The lab preparation must be completed before coming to the lab. Show it to your TA for checking at the beginning of the lab and get his/her signature.

1. DC motor

1.1. For the encoder with 90 pulses per revolution compute the factor to convert the frequency of the pulses to the speed in rpm.

1.2. Arduino DC motor PWM controlled with potentiometer

- Study the example sketch provided in lecture notes for controlling the speed of a DC motor by the Arduino module, using a potentiometer.
- Make a copy of the sketch ready to be uploaded in the lab. The source code could be found at this tutorial: http://www.instructables.com/id/Arduino-DC-motor-speed-control-potentiometer/

1.3. Arduino DC motor PWM controlled by the Serial Monitor

- Study the example sketch provided in lecture notes for controlling the speed of a DC motor by the Arduino module, using the Serial Monitor.
- Calculate the necessary Serial Monitor settings for duty cycle values listed in Table 4.3.
- Make a copy of the sketch ready to be uploaded in the lab. The source code can be found in the lecture notes.

2. Servo motor

2.1. Servo Motor in Sweep Mode

- Study the example “Sweep Mode” sketch provided in lecture notes for controlling a Servo motor by the Arduino module.
- Make a copy of the sketch ready to be uploaded in the lab. The source code can be found in the lecture notes.

2.2. Servo Motor controlled with potentiometer

- Study the example “Potentiometer Control” sketch provided in lecture notes for controlling a Servo motor by the Arduino module.
- Make a copy of the sketch ready to be uploaded in the lab. The source code can be found in the lecture notes.

PROCEDURE

1. DC Motor manual PWM.

1.1. Construct the assembly shown in Figure 4.10 using the following components from the provided kit:

- 2 standoffs #8-32 x0.5” LG
- 2 SHCS #8-32 x0.25” LG (standoffs to plate)
- 2 SHCS #8-32 x0.375” LG (encoder to standoffs)
- 2 SHCS #6-32 x0.25” LG (motor to plate)
- Vex structure 5x3 holes
- Vex encoder
- Vex square shaft 2” LG
- Vex 269 DC motor

Note: Do not fully tighten screws until after shaft has been inserted. Align the motor to the encoder then hand-tighten screws.

1.2. Connect the encoder and motor in a circuit as shown in Figure 4.11.

Note: Use the 5V source on the breadboard for the encoder and 7V from the Power Supply for the DC motor.

When the layout has been completed, have your TA check your breadboard for errors and get his/her signature in the Signature section of the LMS.

1.3. Ensure the function generator is connected through its TTL OUTPUT, which is used to simulate a PWM signal coming from a control circuit. Connect Ch1 of the oscilloscope to the Function Generator.

1.4. Set the frequency on the function generator to about 5 kHz (use the 10 kHz scale).

1.5. Use the CURSOR menu of the oscilloscope to measure the Period ON and Total Period of the input signal, as it is shown on Figure 4.9. Set the duty cycle to about 15% and record the time periods in Table 4.1 and calculate the Measured Duty Cycle. Accuracy of ±2% is acceptable.

1.6. Without changing the FG settings move the CH1 of the Oscilloscope to the output of the Encoder, measure the Encoder frequency [Hz] and record it in Table 4.1.

1.7. Calculate the RPM of the DC motor.

1.8. Repeat the steps for all listed Target duty cycles.

1.9. Demonstrate the circuit to the TA and get your Marking Sheet signed.

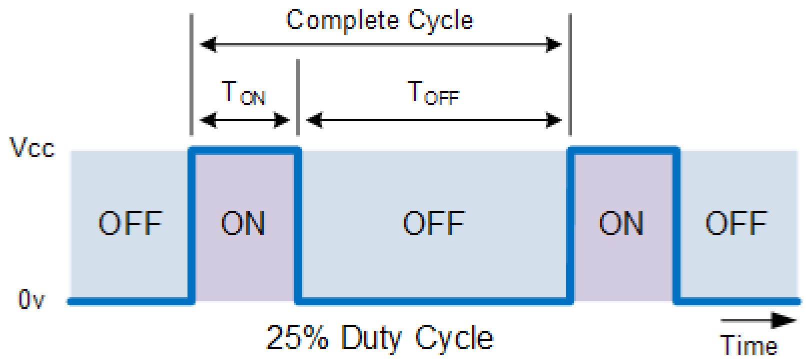


Figure 4.9. Duty cycle measurements

Notes:

- You do not need to re-adjust the frequency after changing the duty cycle.
- To get a duty cycle <50% the duty knob must be PUSHED IN.
- To get a duty cycle >50% the duty knob must be PULLED OUT.

Figure 4.10. DC Motor and Encoder assembly

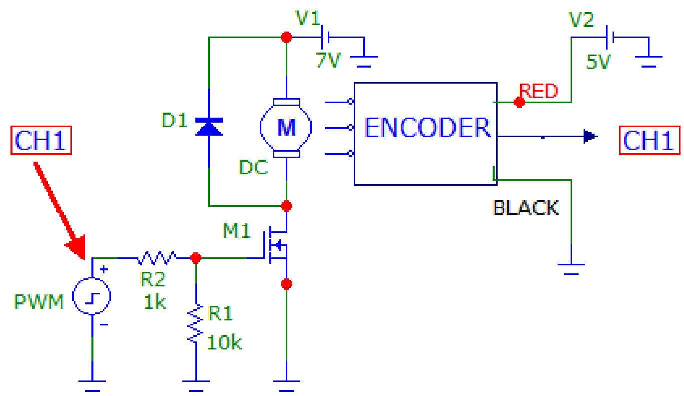


Figure 4.11. DC Motor and Encoder electrical diagram

2. Arduino DC motor PWM controlled with potentiometer

2.1. Connect the input of the Motor Control circuit to Digital Output pin 3 of the Arduino board.
2.2. Connect the 10k Potentiometer to the power supply of the Arduino and connect the slider to the Analog Input pin A0, as it is shown in Figure 4.12.
2.3. Connect Ch1 of the oscilloscope to Encoder CH1 in order to measure the frequency of the signal. Connect Digital Multimeter to pin A0 to measure the input DC voltage.

NOTE:
Provide COMMON GROUND connection between Arduino and other power supply voltages (+5V for the Encoder and +7V for the DC motor)!

When the layout has been completed, have your TA check your breadboard for errors and get his/her signature on the Marking Sheet.

2.4. Upload the program code for this experiment
2.5. At each listed input voltage measure the frequency of the encoder, using the CH1 of the oscilloscope. Calculate and record the RPM of the DC motor in Table 4.2.
2.6. Demonstrate the circuit to the TA and get your Marking Sheet signed.

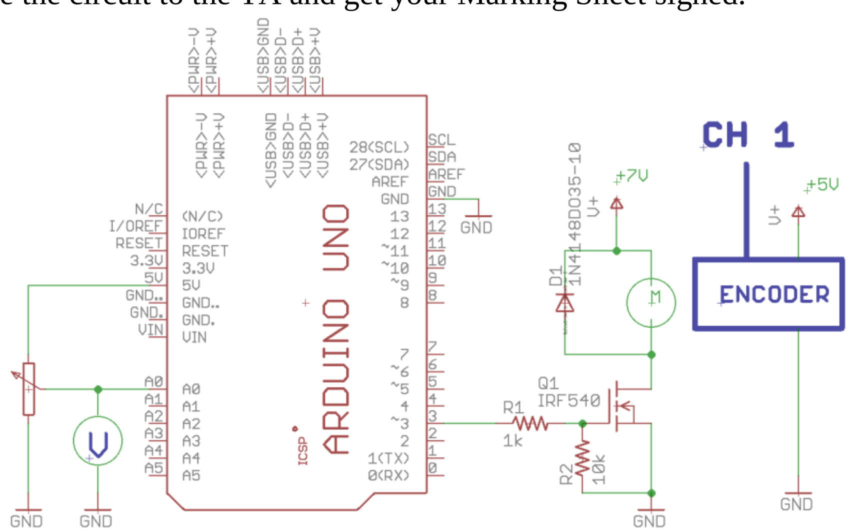


Figure 4.12. DC motor control circuit diagram

3. Arduino DC motor PWM controlled by Serial Monitor

3.1. Without changing the circuit of Figure 4.12 upload the Serial Monitor Control code from the tutorials to the Arduino board.
3.2. Open the serial monitor.
3.3. Set the Serial Monitor to “no line ending”.
3.4. Enter your calculated HEX value from the pre-lab (between 0 to 255) and hit enter.
3.5. At each listed duty cycle measure the frequency of the encoder, using the CH1 of the oscilloscope. Calculate and record the RPM of the DC motor in Table 4.3.
3.6. Demonstrate the circuit to the TA and get your Marking Sheet signed.

4. Servo Motor in Sweep Mode

4.1. Use the provided Servo Motor in your Arduino kit, shown in Figure 4.12 and connect it to Arduino board.

NOTE: The servo motor pin connector diagram is shown in Figure 4.13. Provide proper +5V and GND power supply. Connect the Control pin (orange wire) to Digital pin D9, as it is shown in Figure 4.14.

Figure 4.12. Micro Servo Motor

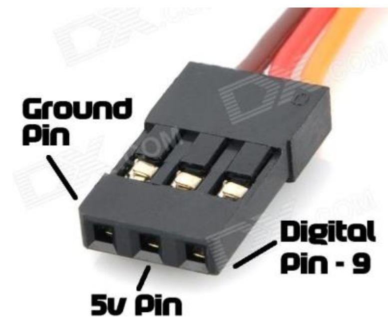


Figure 4.13. Servo Motor pin connector

4.2 Upload the program code for this experiment.
4.3 Demonstrate the circuit to the TA and get your Marking Sheet signed.

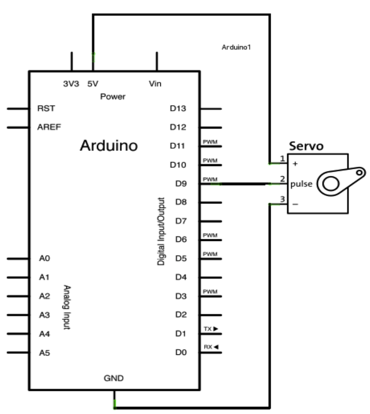


Figure 4.14. Servo Motor Connection

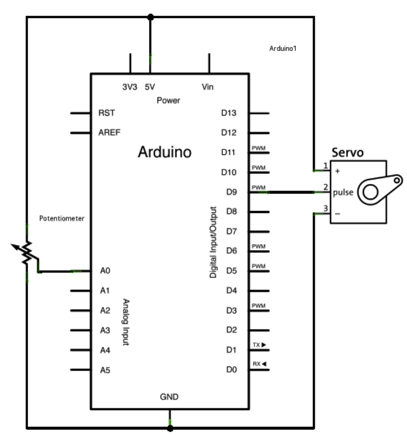


Figure 4.15. Servo Motor – Potentiometer control

5. Servo Motor controlled by potentiometer

5.1. Build the circuit in Figure 4.15 by connecting a 10k potentiometer to pin A0.
5.2. Upload the program code for this experiment.

5.3. Connect CH1 of the Oscilloscope to the control pin of the servo motor (D9). Using the Cursors menu measure the generated by Arduino pulses width at angles between 0^{o} and 180^{o}, as it is shown in Table 4.4.

5.4. Demonstrate the circuit to the TA and get your Marking Sheet signed.

REPORT

Record in the LMS results for all experiments and answer questions.

LAB MEASUREMENTS SHEET – LAB #4

Name ________________________

Student No____________________

Workbench No_____

NOTE: Questions are related to observations, and must be answered as a part of the procedure of this experiment.

*Sections marked * are pre-lab preparation and must be completed BEFORE coming to the lab.*

1. **DC motor.**

*1.1. ** If the encoder has 90 pulses per revolution compute the factor to convert from the frequency of the pulses to the speed in rpm. [5 marks]

2. DC Motor PWM Control

2.1. DC motor Speed Measurements using manual PWM control [25 Marks]

Table 4.1.

Target Duty Cycle (%)	Period On (us)	Total Period (us)	Measured Duty Cycle (%)	Encoder Frequency (Hz)	Calculated RPM
15					
25					
50					
75					
90					

2.2. DC motor Speed Measurements - Arduino PWM controlled by potentiometer [25 Marks]

Table 4.2.

Input Voltage (V)	Encoder Frequency (Hz)	Calculated RPM
1		
2		
3		
4		
5		

2.3. DC Motor Speed Measurements- Arduino PWM controlled with Serial Monitor [20 Marks]

Table 4.3

Duty Cycle (%)	*Serial Monitor Value *	Encoder Frequency (Hz)	Calculated RPM
15			
25			
50			
75			
90			

2.4. Compare your lab measurements in Table 4.1 and Table 4.3 or the same Duty Cycle PWM settings. What method of PWM provides more precise DC motor speed control? [5 Marks]

____
____
____
____

3. Servo Motor PWM measurements

Table 4.4.

Shaft Position [deg]	Pulse width [ms]
0^o	
45^o	
90^o	
135^o	
180^o	

SIGNATURE AND MARKING TABLE – LAB #4

TA Name:___________________

Check boxes			Task	Max. Marks	Granted Marks	TA Signature
			Pre-lab completed	15		
			Manual PWM control	20		
			Arduino PWM controlled by pot.	20		
			Arduino PWM controlled by Serial Mon.	20		
			Servo Motor Sweep Mode	10		
			Servo Motor PWM controlled by pot.	15		
			TOTAL MARKS	**100**		