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.

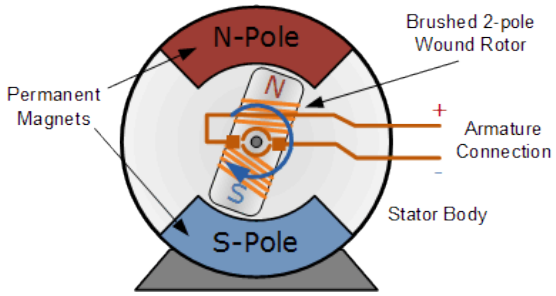


Figure 4.1. A simplified diagram of the DC Motor

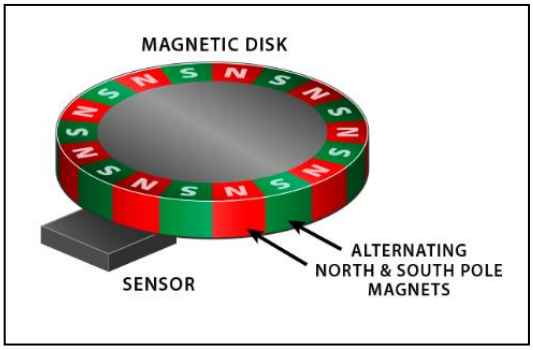
- The current flowing within the rotor coils induces a magnetic field with north and south poles. This magnetic field is repelled or attracted by the stator's permanent magnets producing a rotational movement of the armature around the motors central axis.
- The armature is connected to a voltage source 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. Magnetic Hall sensor 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

The magnetic encoder detects rotational position information as changes of the magnetic field, converts them into electrical signals, and outputs them. The simplest magnetic encoder consists of a permanent magnet and a magnetic sensor.



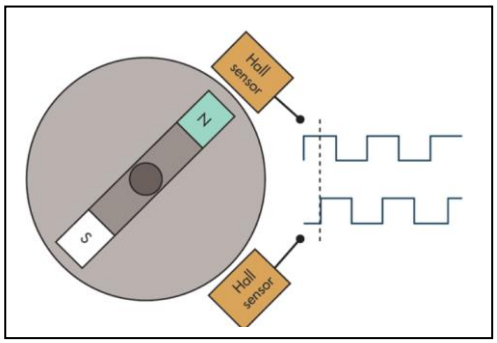
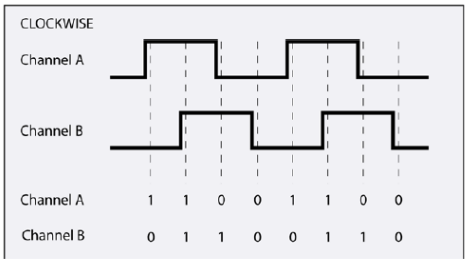
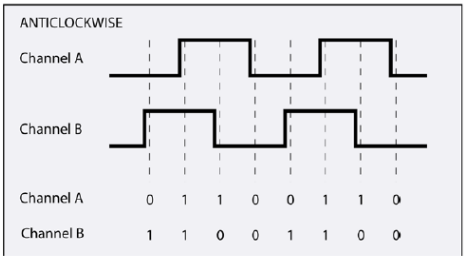


Figure 4.2. Single output and Quadrature Magnetic Hall Sensor Shaft Encoders

- The Hall-Effect sensor detects the magnetic field change and then the electronics circuit converts it to a square-wave voltage output.
- The magnetized rotor is attached to the DC motor rotor therefore both turn in synchrony.
- The Magnetic Shaft Encoder of the N20 DC motor generates 7 PPR (pulses per rotation) Counting the pulses enables the user to track and determine the speed of rotation (RPM) of the DC motor – Figure 4.4.
- The N20 - 3 DC motor reduction gear box has a ratio of 150:1.
- The N20 - 3 DC motor has two hall sensors (A and B) that are offset by quarter of a cycle and they form a Quadrature Encoder
- The phase difference of the resulting signals indicates the direction of rotation – Figure 4.3.

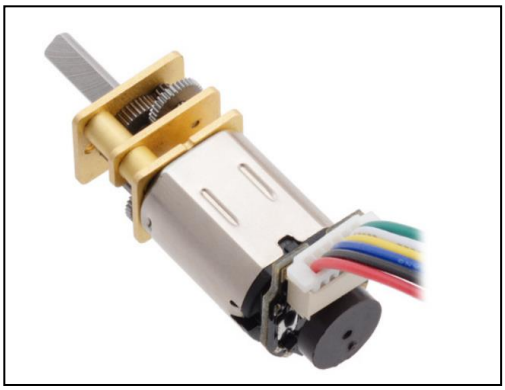


Clockwise rotation: B switches from 0 to 1 when A=1



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

Figure 4.3. Determining the direction of the rotation using quadrature encoder



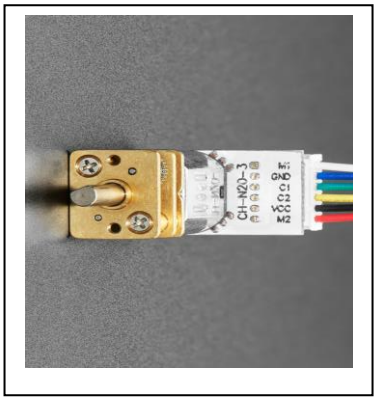


Figure 4.4. N20 -3 DC motor with reduction gear box

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 4.5. The duty cycle is the fraction of time that the output voltage is "ON" compared to when it is "OFF".

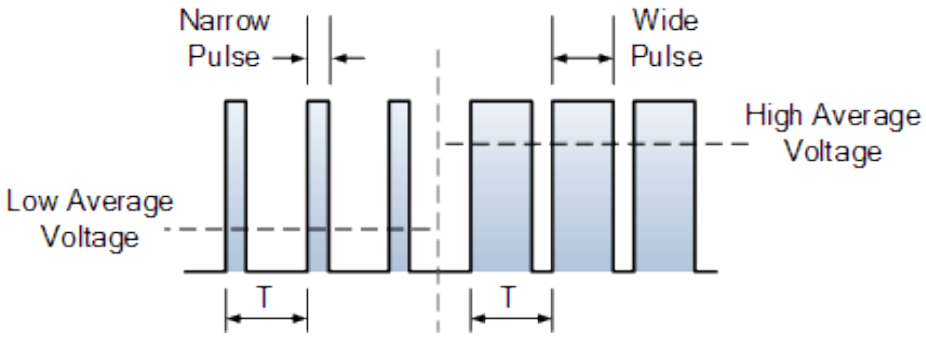


Figure 4.5. PWM (Pulse Width Modulation) speed control

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, anhttps://en.wikipedia.org/wiki/Error_signal 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 4.6.

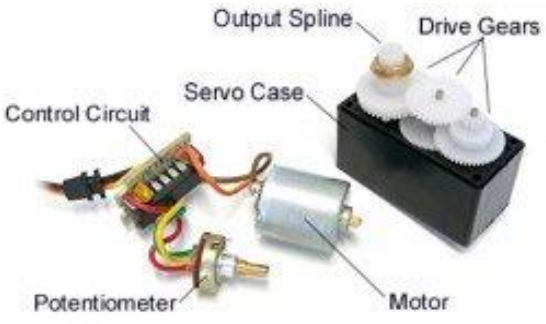


Figure 4.6. Servo Motor

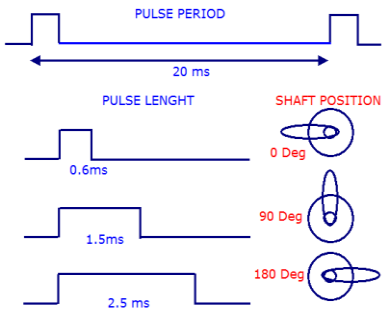


Figure 4.7. Servo Motor Control signals

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 4.7. 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 Rigol DM 3058
2. Digital oscilloscope Rigol DS1000Z
3. Function Generator Mulicomp PRO MP75066
4. PROTO-BOARD PB-503 (breadboard)
5. MOSFET 2N7000
6. Arduino Uno board
7. Geared DC motor with magnetic Encoder CH-N20-3
8. 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 N20-3 DC motor with a magnetic encoder with 7 pulses per revolution, attached directly to the shaft, and build-in 150:1 reduction gear box, compute the factor to convert the frequency of the encoder pulses (pulse/sec) to the output shaft speed in RPM (rotation per minute). You have to use the conversion factor to calculate the output shaft RPM during the experiments.

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.

Use the provided N20 -3 DC motor with built-in encoder and 150:1 gear box.

1.1. Connect the encoder and motor in a circuit as shown in Figure 4.10 using the 2N7000 MOSFET transistor package diagram on Fig. 4.8. Use the 5V DC source from the Power Supply for the DC motor and the Encoder. Follow the correct polarity of the connections.

DC motor and Encoder connection chart

Pin Name	Wire Color	Signal
M1	Brown	GND for the DC Motor
GND	Red	GND for the Encoder
C1	White	Encoder Ch.1
C2	Purple	Encoder Ch.2
VCC	Green	Encoder Power +5V
M2	Black	DC Motor Power +5V

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.

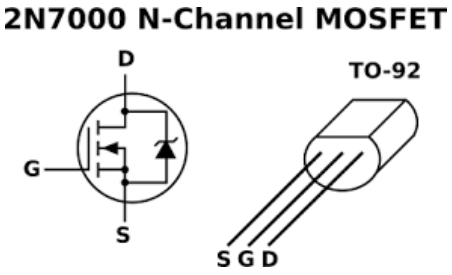


Fig. 4.8. 2N7000 pin layout

1.2. Use the function generator to provide a PWM signal. Connect Ch1 of the oscilloscope to the encoder output and measure the frequency of the generated signal at different duty cycle settings.

1.3. Set the Function Generator parameters to:

- Frequency: 100Hz
- Amplitude: 5 Vp-p
- DC offset: 2.5V
- Duty cycle: 10 - 90%

1.4. Set the duty cycle to 10% and record the time periods in Table 4.1

1.5. Measure the time parameters of the input signal using the Oscilloscope and record the time periods in Table 4.1.

1.6. Calculate the RPM of the DC motor output shaft after the gear box.

1.7. Repeat the steps for all listed Target duty cycles.
1.8.Demonstrate the circuit to the TA and get your Marking Sheet signed.

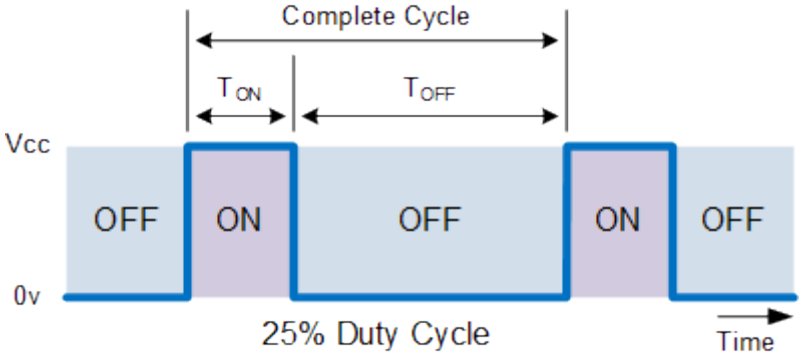


Figure 4.9. Duty cycle measurements

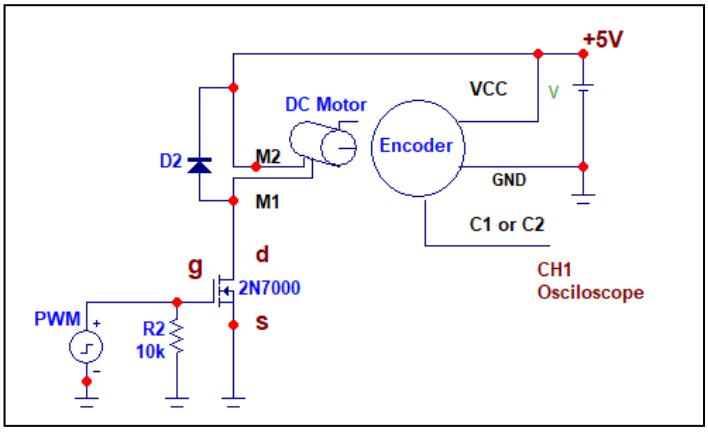


Figure 4.10. DC Motor and Encoder electrical diagram

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 4.11.
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:
Use the +5V and GROUND connection from Arduino as a power supply voltage +5V !

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.

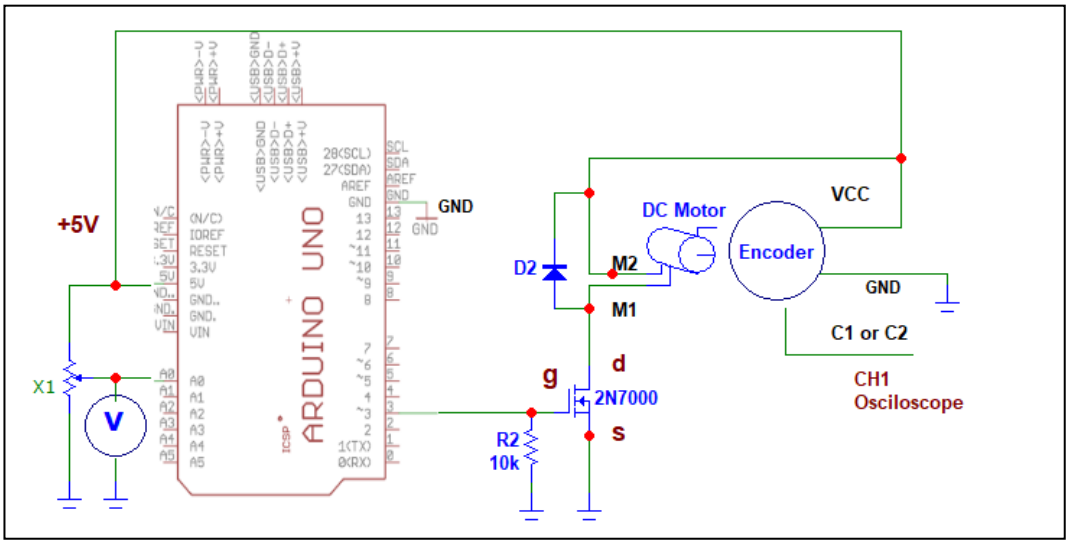


Figure 4.11. DC motor control circuit diagram

3. Arduino DC motor PWM controlled by Serial Monitor

3.1. Without changing the circuit of Figure 4.11 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

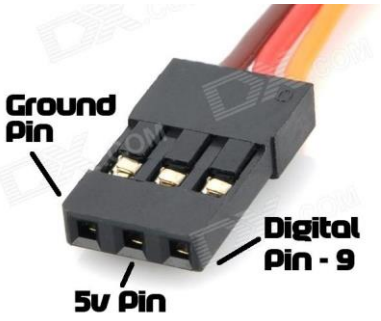


Figure 4.13. Servo Motor pin connector

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

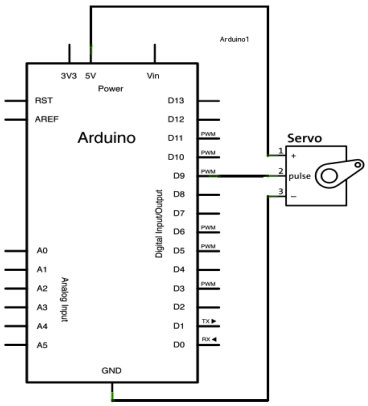


Figure 4.14. Servo Motor Connection

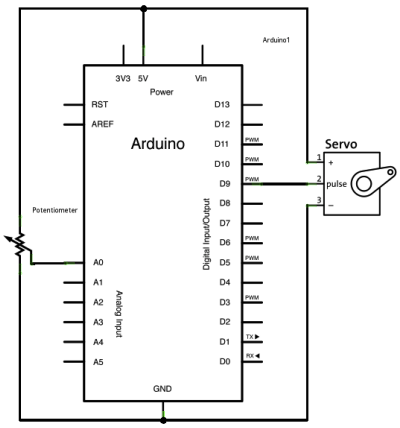


Figure 4.15. Servo Motor – Potentiometer control

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 Arnav Goyal

Student No 251244778

Workbench No #35

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. * For the encoder with 7 pulses per revolution and a reductor gear box of 75:1, compute the factor to convert from the frequency of the pulses to the output shaft speed in rpm. [5 marks]

f_pulse * 60 / (7 * 150) = rpm

conversion factor = 60/(7*150)

2. DC Motor PWM Control

2.1. DC motor Speed Measurements using Function Generator in manual PWM control [25 Marks]
Table 4.1.

Target Duty Cycle (%)	Period On (us)	Total Period (us)	Encoder Frequency (Hz)	Calculated RPM
10	1000	10,000	348	19.885
20	2000	10,000	900	51.43
30	3000	10,000	1100	62.85
40	4000	10,000	1200	68.57
50	5000	10,000	1220	69.71
60	6000	10,000	1300	74.28
70	7000	10,000	1330	76
80	8000	10,000	1370	78.28
90	9000	10,000	1390	79.42

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	641	36.62
2	1090	62.285
3	1290	73.71
4	1350	77.14
5	1420	81.14

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
10	0.1 * 255 = 26	0	0
20	0.2 * 255 = 51	620	35.42
30	0.3 * 255 = 77	961	54.91
40	0.4 * 255 = 102	1120	64
50	0.5 * 255 = 128	1230	70.28
60	0.6 * 255 = 153	1300	74.285
70	0.7 * 255 = 179	1330	76
80	0.8 * 255 = 204	1370	78.285
90	0.9 * 255 = 230	1410	80.57

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]

I think that the PWM Control from the function generator (Table 4.1) results in more accurate speed control. I think this is the case because the arduino PWM doesnt turn on at 10% duty cycle and the fact that the function generator probably produces a 'cleaner' PWM signal than the arduino does, which can result in a more accurate speed control.

3. Servo Motor PWM measurements

Table 4.4.

Shaft Position [deg]	Pulse width [ms]
0°	0.540
45°	1.04
90°	1.48
135°	1.94
180°	2.380

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**		