City, University of London

Department of Electrical and Electronic Engineering

LAB - MANUAL

EE3600 / EE3700 Design - III

Arduino Servo Motor

T1 LAB - 5

Version: 2020/2021

1.0 Introduction

As mentioned in our Arduino based DC Motor control project, a DC Motor is one of the most used electric motor in electronics, robotics, toys etc. It generally consists of only two wires for electrical connection. When these wires are properly connected and given electrical supply (like a battery), the motor starts rotating. A technique called Pulse Width Modulation (PWM) allows us to control the speed of rotation of the motor.

The Servo Motor is entirely different from that of a DC Motor. A Servo Motor is a type of actuator that provides high precision control of linear or angular position. A typical servo motor consists of four components (or parts): a DC Motor (or AC Motor), a gear unit, a position and speed sensing device and a control unit. Servo Motors are used in applications where remarkably high precision motion is required like assembly robots, computer numerical controls etc.

Servo motors are extremely useful in so many different applications; it would be good to learn how to control them. In this design project we will learn how servo motor works and how to control servo motors using the Arduino platform.

2.0 Required Components

- Arduino UNO
- SG90 Servo Motor
- Breadboard
- Ultrasonic Sensor- (HC-SR04)
- Connecting wires

3.0 Components Description

3.1 Servo Motor

Servo motor is an electrical device which can be used to rotate objects (like robotic arm) precisely. Servo motor consists of DC motor with error sensing negative feedback mechanism. This allows precise control over angular velocity and position of motor. In some cases, AC motors are used. It is a closed loop system where it uses negative feedback to control motion and final position of the shaft. It is not used for continuous rotation like conventional AC/DC motors. It has rotation angle that varies from 0° to 180°.

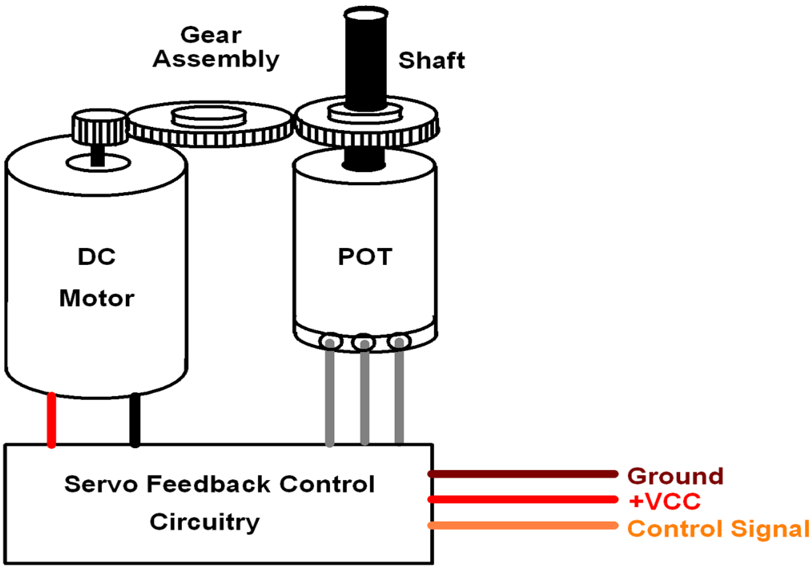

Figure 1: Servo motor mechanism

It consists of DC motor, gear assembly and feedback control circuitry. PWM signal is used to control the servo motor. It is applied on control signal pin. Servo feedback control circuitry contains comparator which compares the control signal (PWM) and potentiometer reference signal to generate error signal which is later amplified and given to the DC motor.

DC motor shaft is connected to potentiometer shaft (knob) through a gear assembly. So rotating DC motor rotates potentiometer, which in turn changes the potentiometer reference signal given to the comparator. At some position of shaft, both potentiometer signal and control signal strength match, which produces zero error signal output. Hence rotation continues until the comparator output error signal becomes zero and DC motor stops.

Servo motor position is controlled by pulse width modulation (PWM) signal. The width of the control signal pulse is used to vary shaft position of servo motor.

Figure 2: Micro Servo SG90

We can vary SG90 Micro servo motor angular rotation in between 0° to 180° angle with PWM signal as shown in below Figure 3.

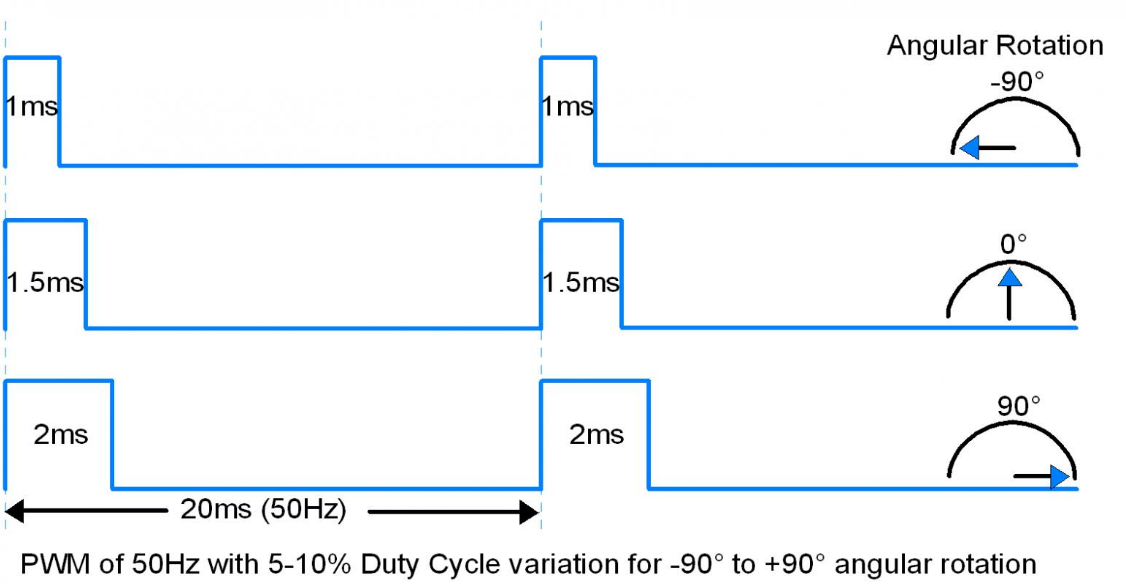

Figure 3: Pulse with modulation signal

Figure 3 shows angular rotation of servo shaft. It uses PWM of 50Hz frequency with TON variation from 1ms to 2ms. The servo motor rotates 90° in either direction from its middle position i.e. it gives control over 180° of its rotation.

3.1.1 Specifications of Micro Servo Motor - SG90

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/ 60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 µs
- Temperature range: 0 ºC – 55 ºC

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left

4.0 Working of the Project

4.1 Servo motor controlled by Arduino

The main components of this project is Arduino UNO board and servo motor. The design of the circuit is straight forward. As mentioned earlier, a servo motor has three wires: two for electrical connection and one for control signal.

So, connect the red wire to +5V supply and brown wire to ground. As the control signal is a PWM, the control wire wire must be connected to any of the PWM output pins of the Arduino UNO board. In this project, we are connecting the control wire of the servo motor to **Pin 11** of the Arduino.

No	Servo Motor	Connection to Arduino Pins
1	**GND (Brown)**	**GND**
2	**VCC (Red)**	**5V**
3	**DATA (Orange)**	**9**

Table 1: Servo-Arduino pin configuration

4.2 Servo motor controlled by Ultrasonic sensor and Arduino

In the second part of this design project the **ultrasonic sensor** and Arduino controls the servo motor. The ultrasonic sensor will detect the object in front of the sensor if an object came closer than 20 cm (you can define any distance) to the ultrasonic sensor it will send a command to the Arduino microcontroller to operate the servo motor.

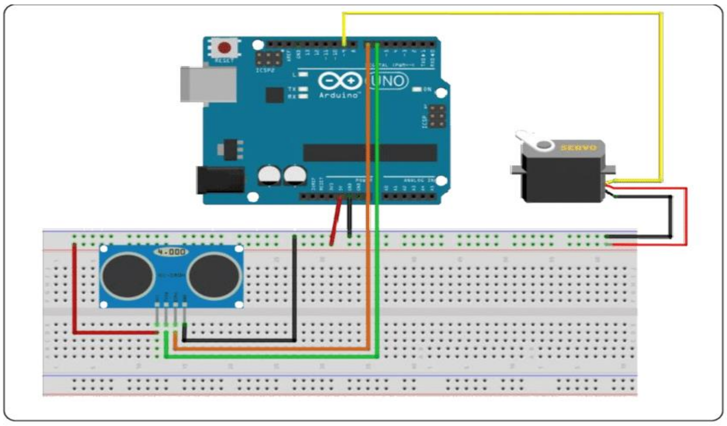

Figure 3: Arduino – Ultrasonic circuit diagram

Pin configuration of the Ultrasonic Sensor:

No	Ultrasonic Sensor	Connection to Arduino Pins
1	**GND**	**GND**
2	**VCC**	**5V**
3	**Trig**	**2**
4	**Echo**	**2**

Table 2: Ultrasonic sensor - Arduino pin configuration

5.0 Code

Now we must write a program to control the servo motor. The sketch is quite simple. We just need to define the pin where the servo motor is connected, define that pin as an output, and in the loop, section generate pulses with the specific duration and frequency as we needed for the design.

6.0 Tasks

1. Understand the overall concept of this design,
2. Familiarise with hardware setup,
3. Understand the given sketch for this design in **Section 4.1,**
4. Investigate how you can control the servo motor movement (angle range),
5. Modify the given sketch to adopt the hardware setup in **Section 4.2** using this lab and the previous labs experiences,
6. Suggestions for future improvements.

7.0 Sample MCQs for Quiz

7.1) Main difference between servo motor and standard motor is that

1) Servomotor has low inertia and higher starting torque
2) Servomotor has inertia low starting torque
3) Servomotor has high inertia and high starting torque
4) None of these

7.2) What does this **delay(2500);** syntax do to your servo motor?

1) Delay the movement in 2500 seconds
2) Delay the movement in 2.5 second
3) Delay the movement in .25 seconds
4) Delay the movement in 250 milliseconds

7.3) Is this **myServo.attachh(90);** a correct syntax?

1) Yes
2) No

7.4) What is the syntax error on this code?

```
void setup ()
{
myServo.writes(180);
}
```

1) Servo
2) writes
3) ()
4) {}

7.5) Will this **myServo.write(190)** syntax move the 180-degree servo motor properly?

1) Yes
2) No

End of Lab - 5