Lab07 Servo Motor Interfacing

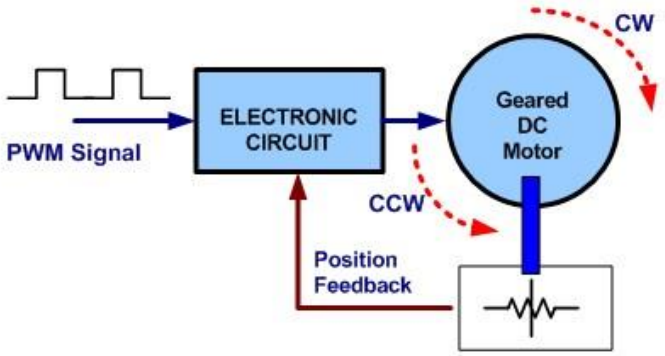
Engineering Mechanics (كلية فلسطين التقنية)

Interfacing Techniques Lab

Experiment No. 07

Servomotor Control and Interfacing

Typical Servo Motor



Servo Motor Block Diagram

The Servo Motor

☒ Introduction

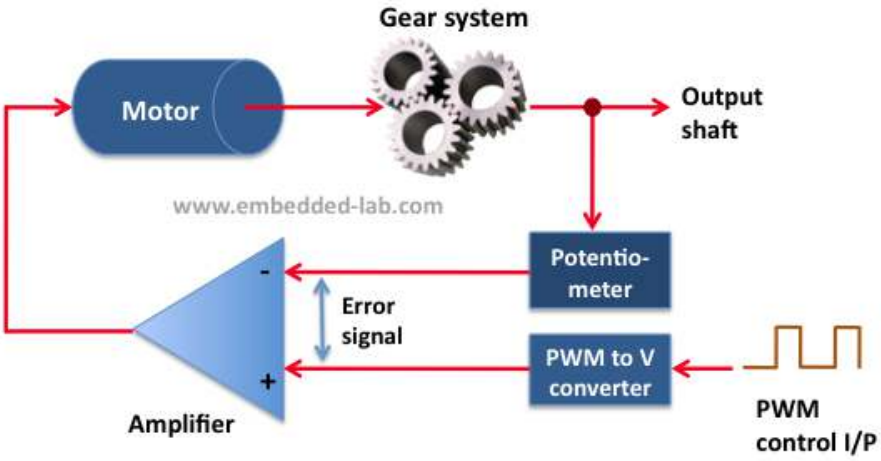
A servo motor is a special geared DC motor equipped with an electronic circuit for controlling the direction of rotation, as well as the position, of the motor shaft. Because servo motors allows precise angular positioning of their output shaft, they are used extensively in robotics and radio-controlled cars, airplanes, and boats to control the motion of their various parts.

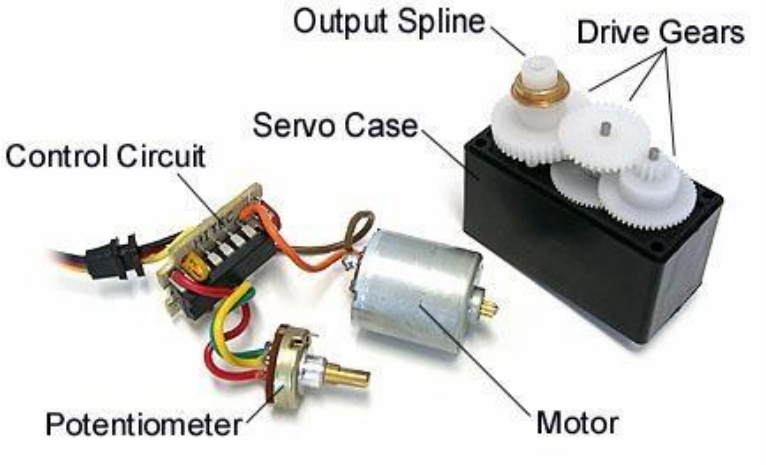
This lab introduces the student to the concept of controlling a servo-motor using Arduino Board

☒ Theory

A servo motor (or servo) is a little box that contains a DC motor, an output shaft (servo arm) which is connected to the motor through a series of gears, and an electronic circuit to control the position of the shaft. The objective of using a servo is to achieve precise angular positioning of an object.

In order to accomplish a servo function, an instantaneous positioning information of the output shaft is fed back to the control circuit using a transducer. A simplest way of doing this is by attaching a potentiometer to the output shaft or somewhere in the gear train. The control electronics compares the feedback signal (which contains the current position of the shaft) from the potentiometer to the control input signal (which contains information of the desired position of the shaft), and any difference between the actual and desired values (known as an error signal) is amplified and used to drive the DC motor in a direction necessary to reduce or eliminate the error. The error is zero when the output shaft gets to the desired position. The functioning block diagram of a typical servomotor is shown below.





☒ Objectives:

1. Study and Interface Servomotor.
2. Find the relation between the frequency , pulse duration and the angle of the servo motor

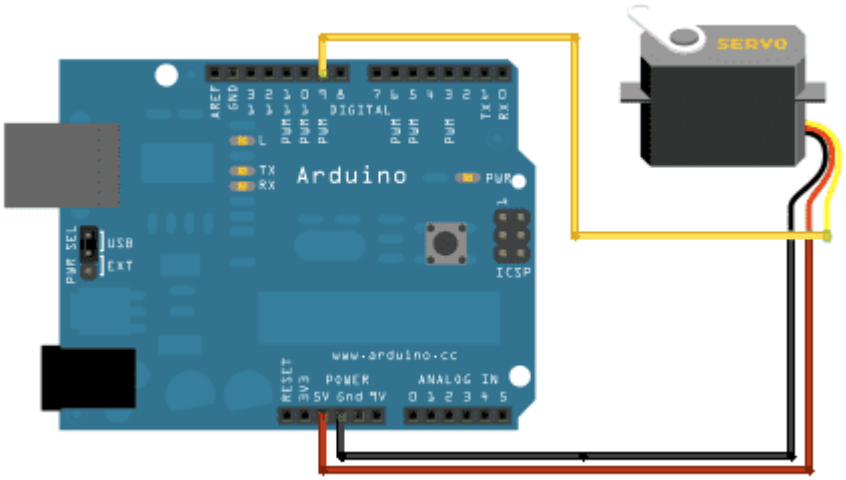
☒ Equipment's List:

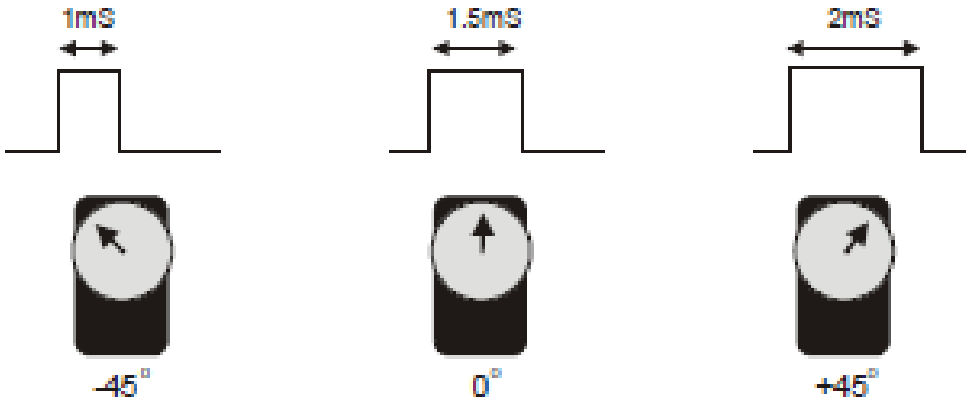
1. Power supply (5 ,12)Voltage.
2. Servo motor and data sheet.
3. Wires.
4. Oscilloscope.
5. microcontroller Study board 220.
6. Arduino Board

☒ Procedure:

1. Connect the RC servo motor to the microcontroller PWM output.
2. Supply the Servo motor from external 5v dc
3. Write a code to convert the required angle value to digital pulses to control the servo motor.
4. Write a code to control the angle of the servo motor using a potentiometer.
5. Follow The design in the following link for proteus servomotor project
   https://youtu.be/7MFGr1NhyIk?t=57

Note: When the servomotor is in operation, it will need about 500mA of current,

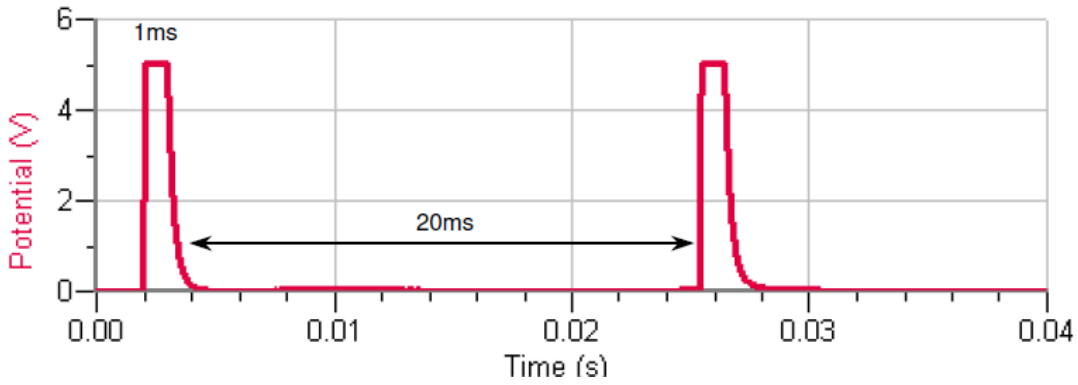


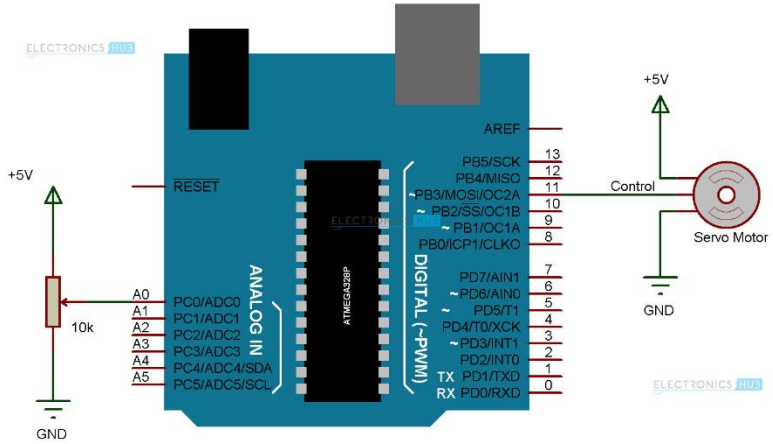


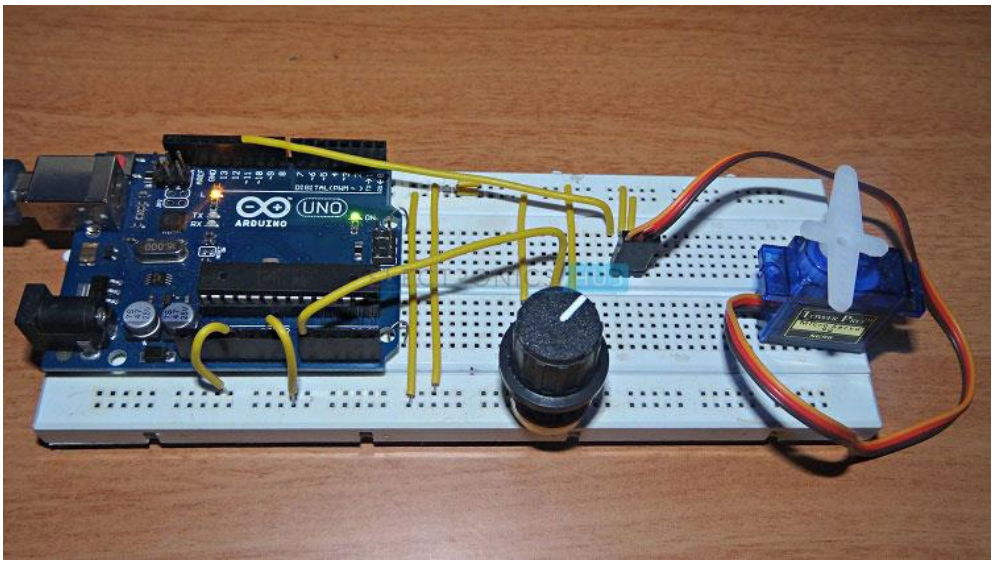
A pulse of 1ms will stop the RC servo at -45 Degrees.
A pulse of 1.5ms will stop the RC servo at 0 Degrees.
A pulse of 2ms will stop the RC servo at +45 Degrees.

Depending on the RC servo you use, these specifications will vary.

.







Code #2

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

Code #2

```
#include <Servo.h>
Servo servoknob;
int potpin = 0;
int val;
void setup()
{
  servoknob.attach(11);
}

void loop()
{
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 180);
  servoknob.write(val);
  delay(15);
}
```