SERVO MOTOR INTERFACING

Introduction: Servo Motor

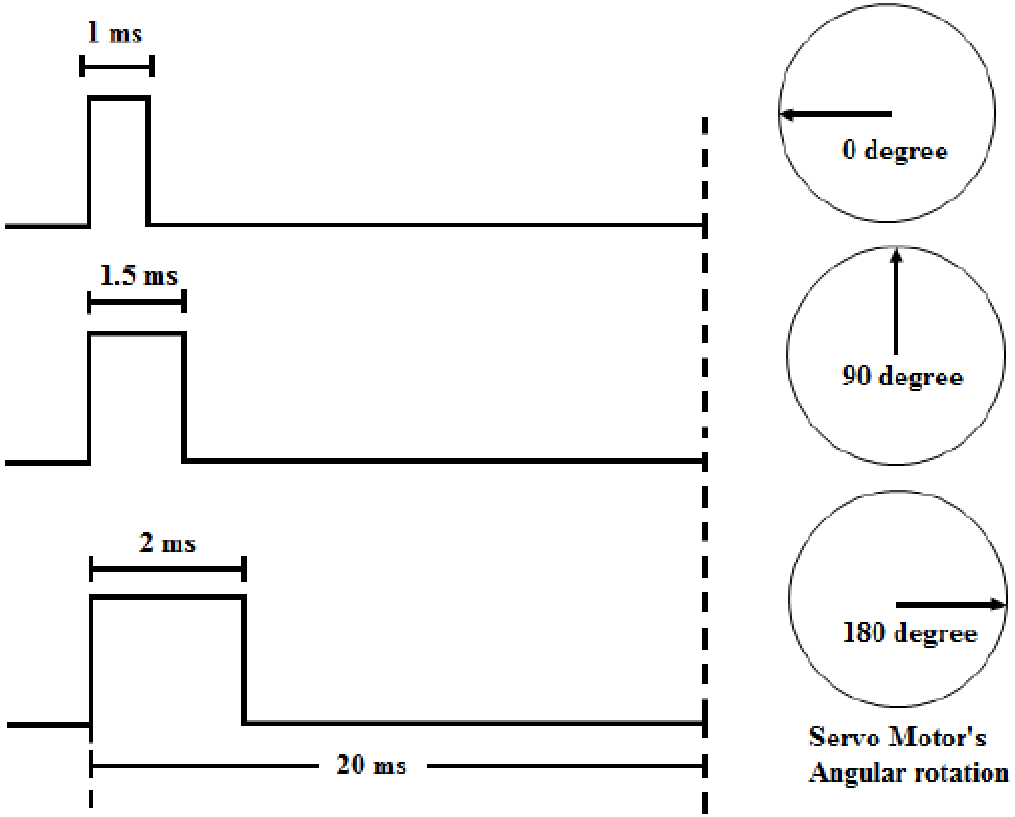
Servomotor is widely used in RC planes, cars, toys and application areas where precise control of motor shaft rotation angle is needed. This motor allows to position the motor shaft at a particular angle ranging from 0 to 180 degrees.

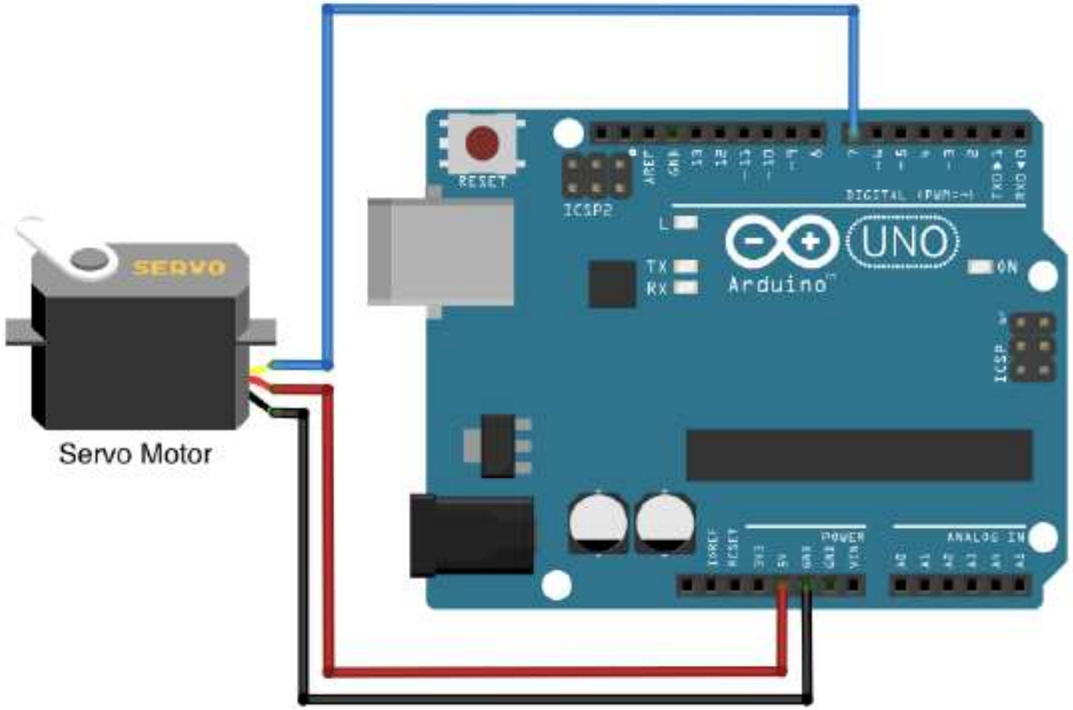
These motors are available in different torque ranges, like: 3 Kg/cm, 6 Kg/cm, 12K/cm or even more. The torque of the motor is specified in Kg/cm and it means the amount of load that can be lifted by the servo motor at a particular distance.

If the motor has a torque of 12 Kg/cm then it means that the shaft of the servo motor can lift 12 Kg load provided the load is suspended 1 cm away from the shaft of the motor.

The signal pin of servo motor is given a Pulse Width Modulated (PWM) Wave to position the motor shaft at a particular angle.

Control Logic





The rotation angle strongly depends on the width of the PWM wave that is applied at the signal pin of the servo motor. Servo motor keeps a track on the signal pulse every 20 msec.

Working of a Servo Motor

Internally the servo motor consists of the following parts: Potentiometer that is coupled to the motor shaft and is used to get the current shaft position, gear mechanism which reduces the speed of the motor but increases the torque.

It also has a motor which rotates till the error between the current position and desired position becomes zero.

The PWM signal applied to the servo motor for positioning the motor shaft at a particular angle is compared with the internal signal that is generated by potentiometer.

The error signal generated is amplified by the control circuit and is applied to the motor so that it can rotate by desired angle. The motor will keep on rotating till the error signal between the applied signal and that of the potentiometer output becomes zero.

Important Functions

servo.attach ( )

This function is used to attach Servo variable to a particular pin of Arduino Uno board. The pin assigned using servo.attach ( ) function is the signal pin that will provide PWM signal to servo motor so that it can be positioned at a desired angle.

Syntax:

servo.attach (pin)
servo.attach (pin, min, max)

Here, servo is a variable of type Servo, pin is the Arduino pin number that will control the servo motor, min and max specifies the minimum and maximum pulse width in microseconds. Default value of min is 544 micro seconds (for 0 degree rotation) and that of max is 2400 micro seconds (for 180 degree rotation).

Example of using servo.attach ( )

```
servo.attach (9);
```

In case you declare the variable name as myservo and want to attach pin 9 to control servo motor, then it can be done using the following statement:

```
myservo.attach (9);
```

The variable name can be defined just after you include the header file in your program.

```
#include <Servo.h> // declare header file for using servo motor functions

Servo myservo; // define variable myservo of type Servo
```

servo.write ( )

Syntax:

servo.write (angle)

This function writes the value on the servo motor. Here, servo is the variable declared in the code of type Servo. You need to specify the angle at which you want to position your servo motor and it can range from 0 to 180 degree.

```
servo.write (90); // this will position the servo motor shaft at an angle of 90 degree
```

servo.writeMicroseconds ( )

Syntax:

servo.writeMicroseconds (uS);

Here, servo is a variable of type Servo and uS is the time value that is specified in micro seconds.

This function is used to position the shaft of the servo motor at a particular angle but here instead of passing the angle value in the function we pass the delay in microseconds and accordingly servo motor will rotate by a particular angle.

For a standard servo you can pass values in between 1000 micro seconds to 2000 micro seconds. Passing a value of 1000 micro seconds will place the motor in fully counter clock wise direction and passing 2000 micro seconds will place the motor in fully clock wise direction. Value of 1500 microseconds will place the motor in between. You can play around these values and see what values work for the servo motor that you are using. Typically, you can vary these values in the range of 700 to 2300.

Servo.writeMicroseconds (1500); // this will place the motor shaft in the middle point.

servo.read ( )

This function is used to read the current angular position of the servo motor. This will return the angle value in between 0 to 180 degree and it is going to be the same as the last angle value that is passed to the servo motor using servo.write ( ) function.

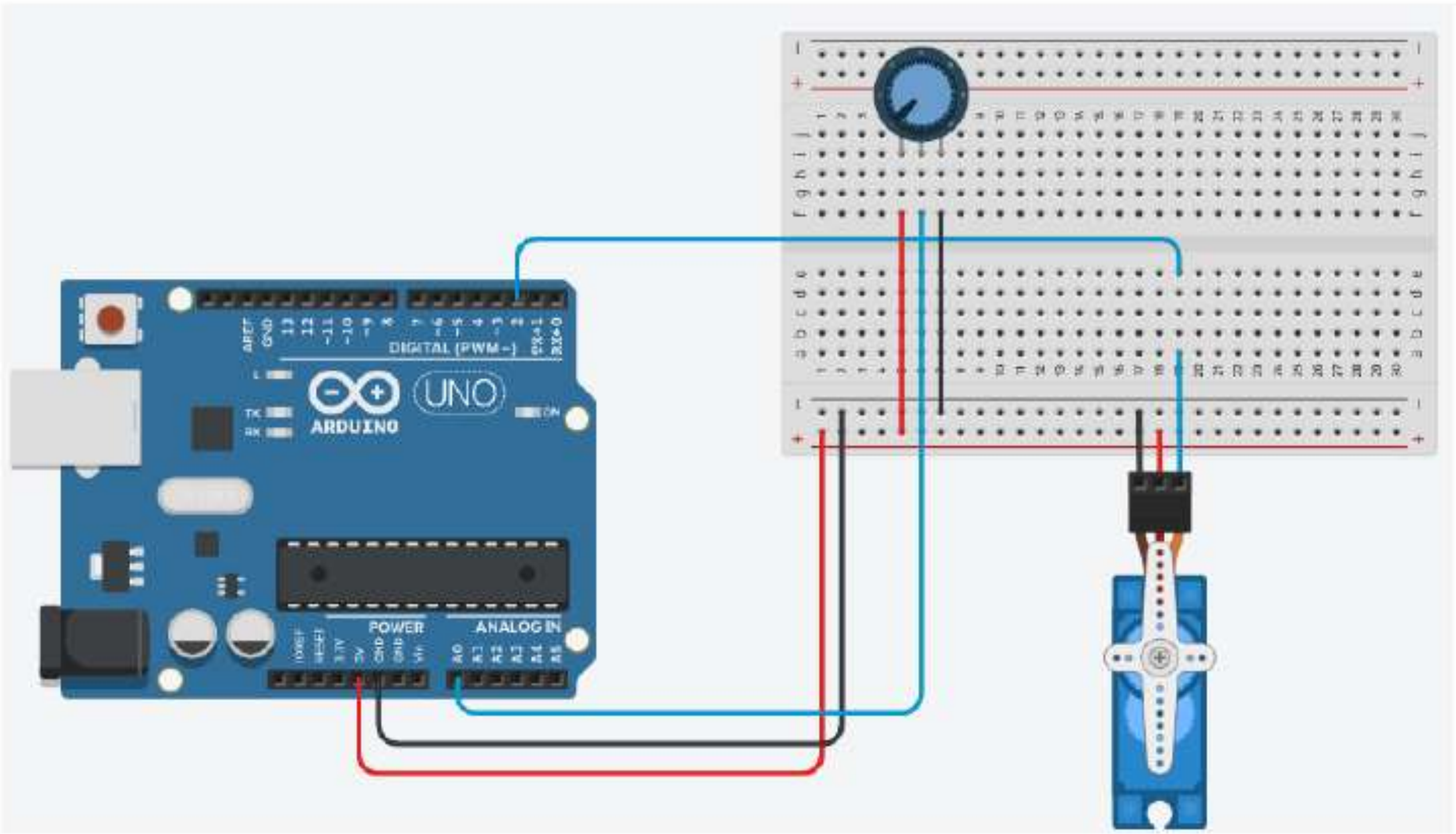
servo.attached ( )

This function is used to check whether Servo variable is attached to a particular pin of Arduino Uno board or not. This function returns true in case servo variable is attached to the Arduino pin and it will return false in case servo variable is not attached to the port pin of Arduino board.

servo.detach ( )

This function is used to detach the Servo variable from the Arduino pin. It is a point to note that in Arduino uno board if you detach all the Servo variables used in Arduino code then pin 9 and 10 can be used for PWM output generation.

Programming Example



Code: Servo Motor Control Through Potentiometer

```
#include <Servo.h>
// importing the Servo library
Servo servo;
// creating servo object to control the motor
int temp, m;
// creating a variable named temp to store adc o/p value
void setup() {
// put your setup code here, to run once:
servo.attach(2);
// connecting the servo motor to pin 7 on the Arduino board
}
void loop() {
// put your main code here, to run repeatedly:
temp = analogRead(A0); // read the A0 channel voltage
m = map(temp,0,1023,0,90); // map the ADC o/p from 0 to 90 degree
servo.write(m); // move servo to desired position
delay(100); // give delay of 100 msec
}
```