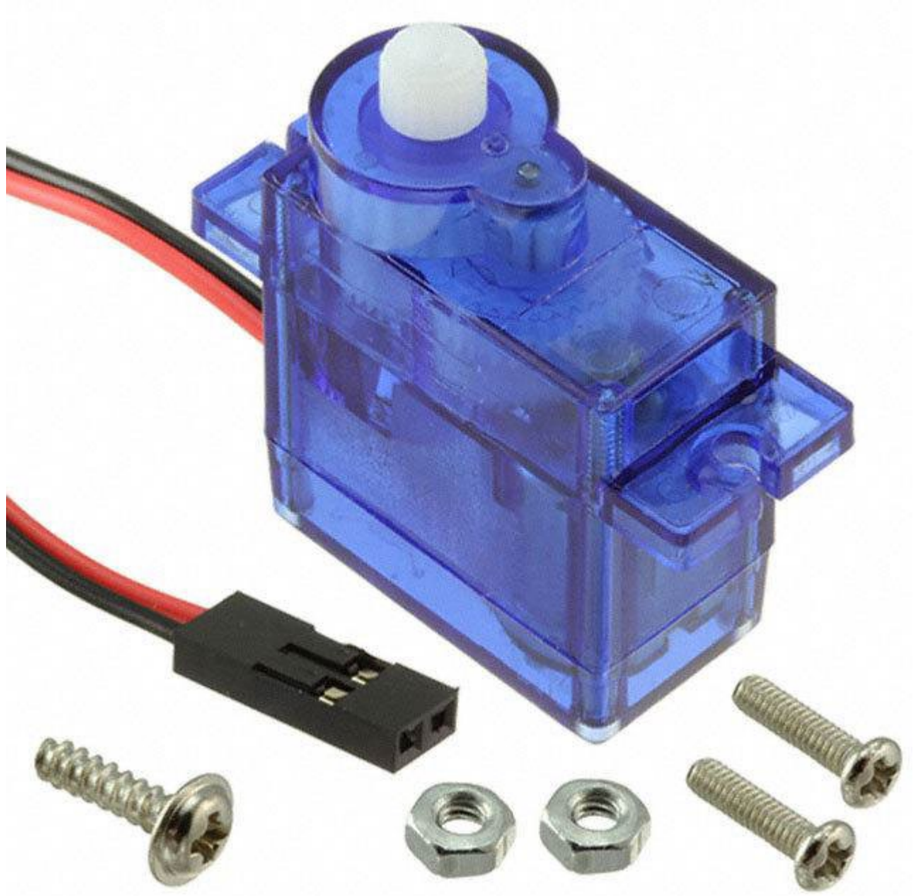
Servo Motor+ Ultrasonic Sensor

Huaishu Peng | UMD CS | Fall 2023



Servo Motor

Low-speed, high-torque motor with precise positional control

A control signal is sent to the servo
to position the shaft to desired angle.

When to use it?

Youtube.com/ChannelCreativeVN
Kênh Sáng Tạo .COM
SELECT
MODE
START
Youtube.com/ChannelCreativeVN
Facebook.com/ChannelCreative

What is a servo motor?

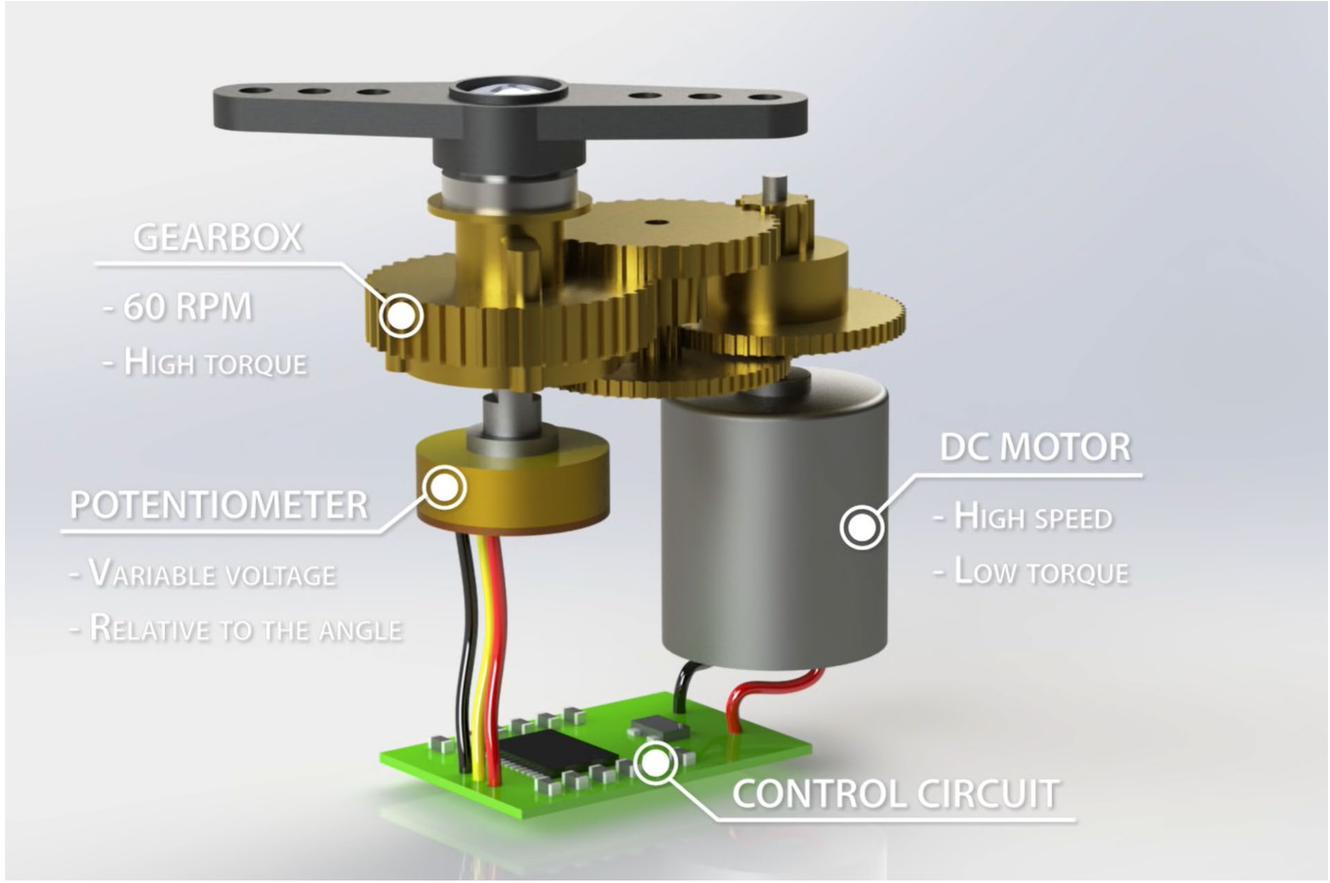
a motor with a built-in “servomechanism”.

Consist of:

An electric motor (e.g., DC motor)

A feedback device

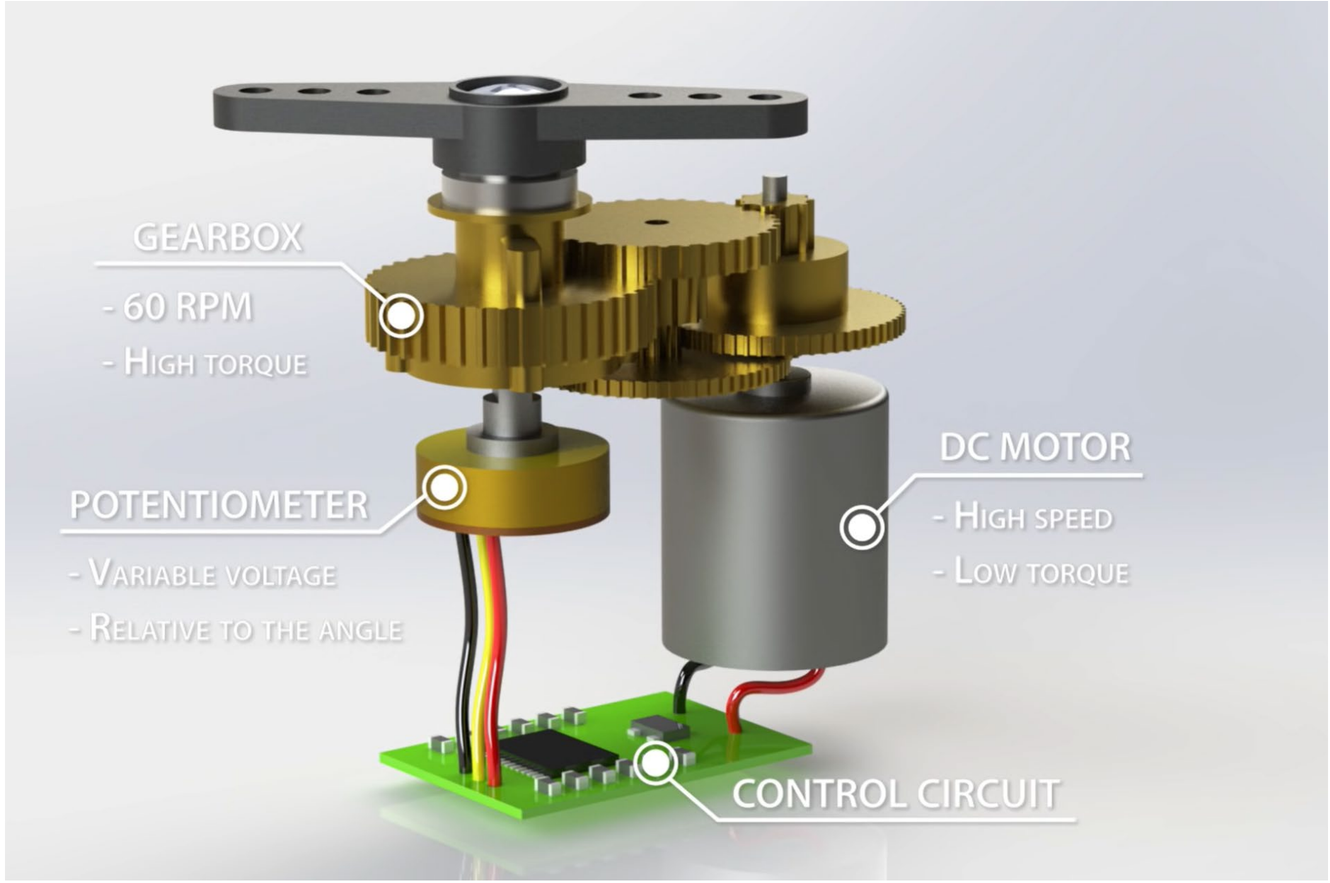
An electronic controller



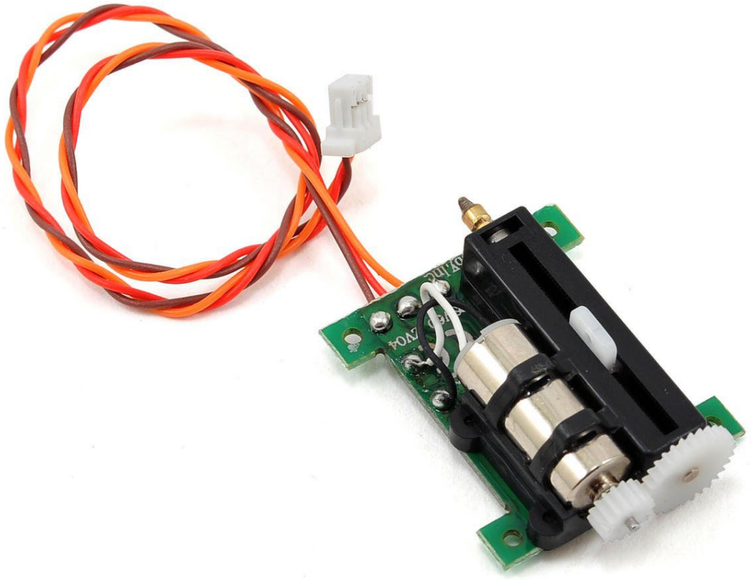
What is a servo motor?

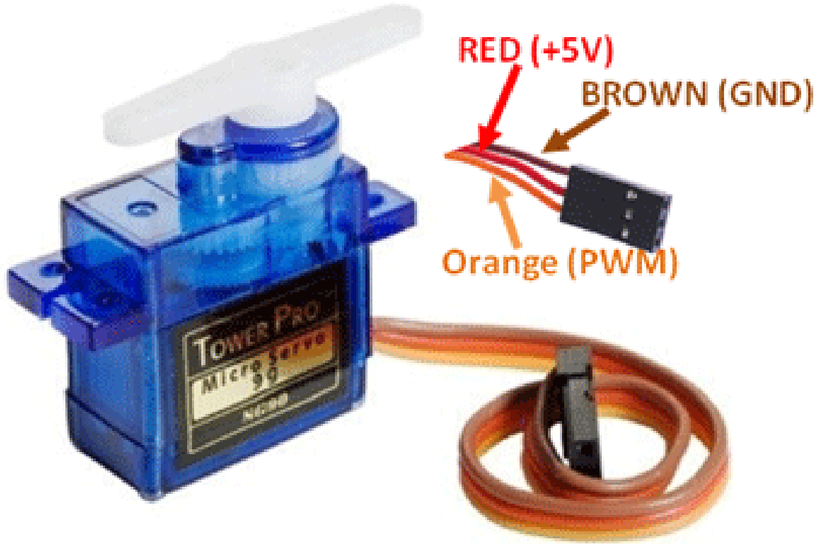
Motors can be DC, or AC

Feedback device can be encoder or other sensors for position sensing



Micro Servo
9g
SG90





Servo Motor

What we have today is a RC servo motor (because it is designed for RC car at the beginning)

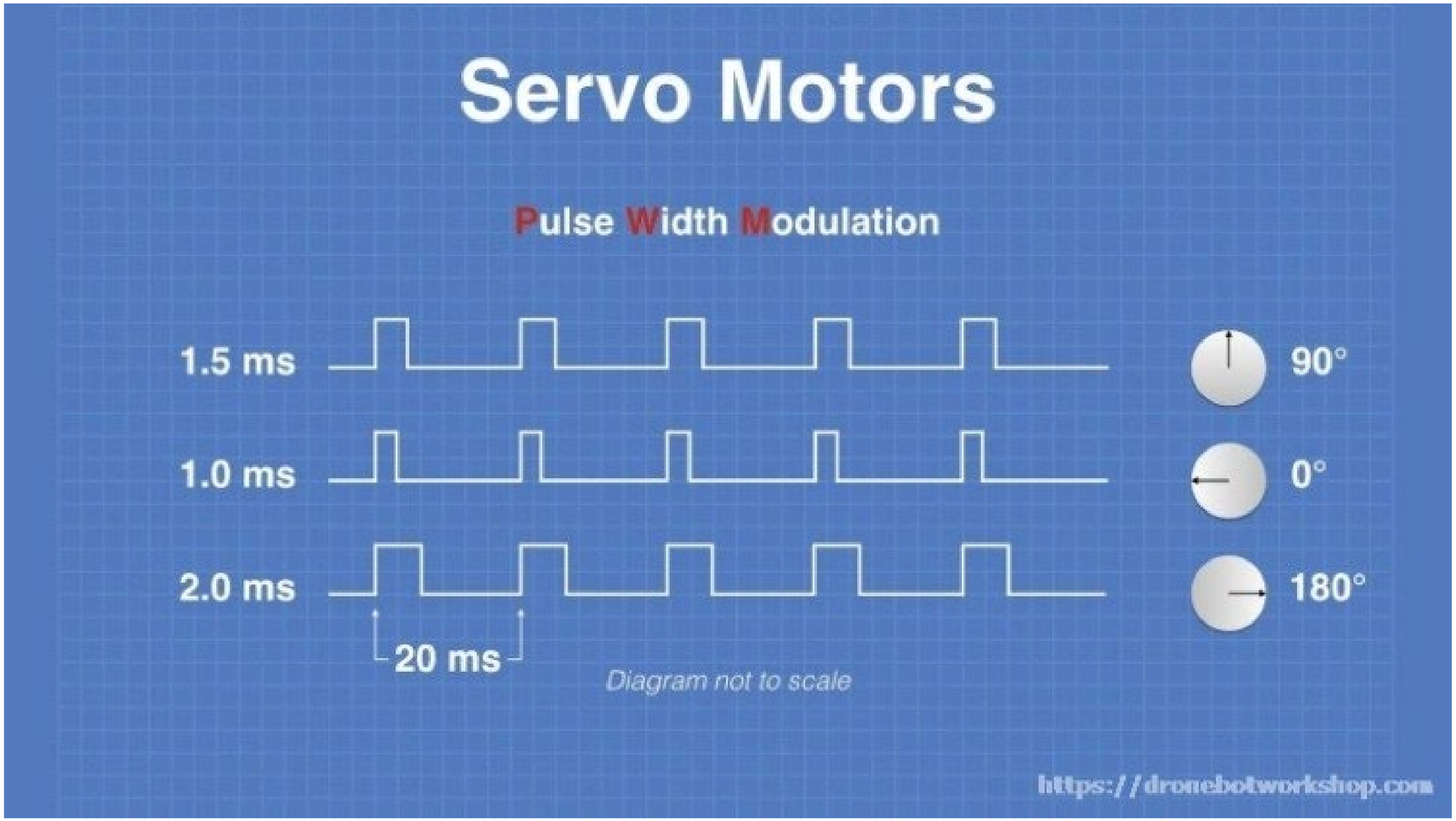
Rotational range: 0 ~ 180 degree

Wire connection: Power (Red) | GND (Brown) | Signal (Orange)

Operating Voltage: **5**V

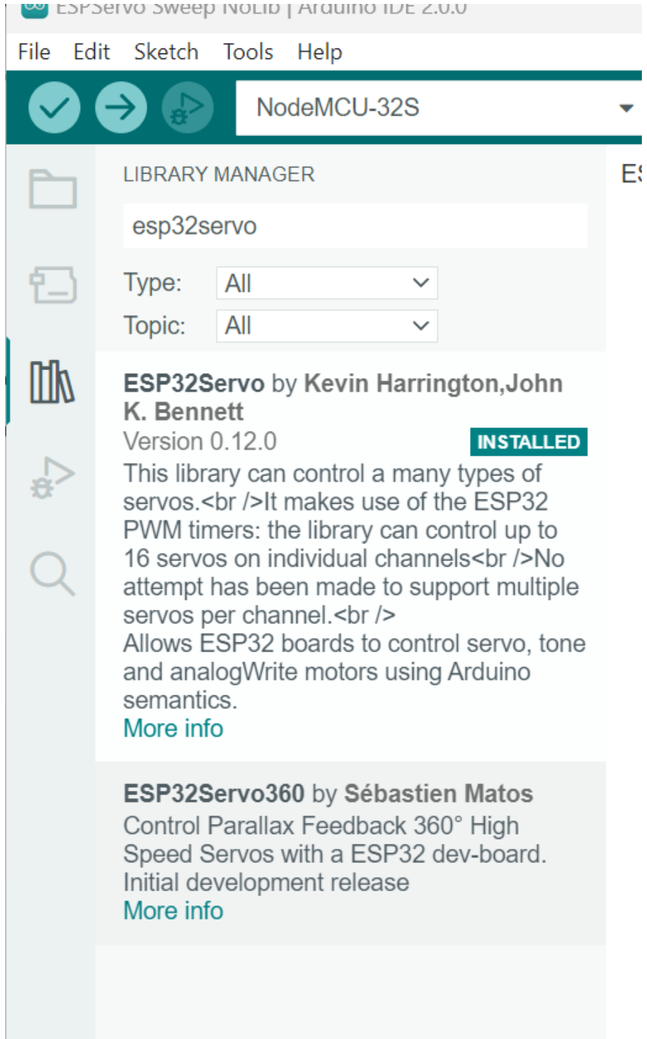
Pulse Width Modulation (PWM)

Controlling the position of a servo motor



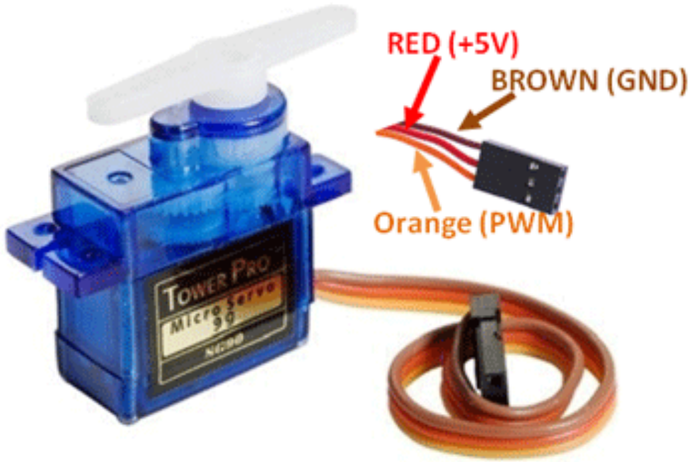
Using a Servo lib

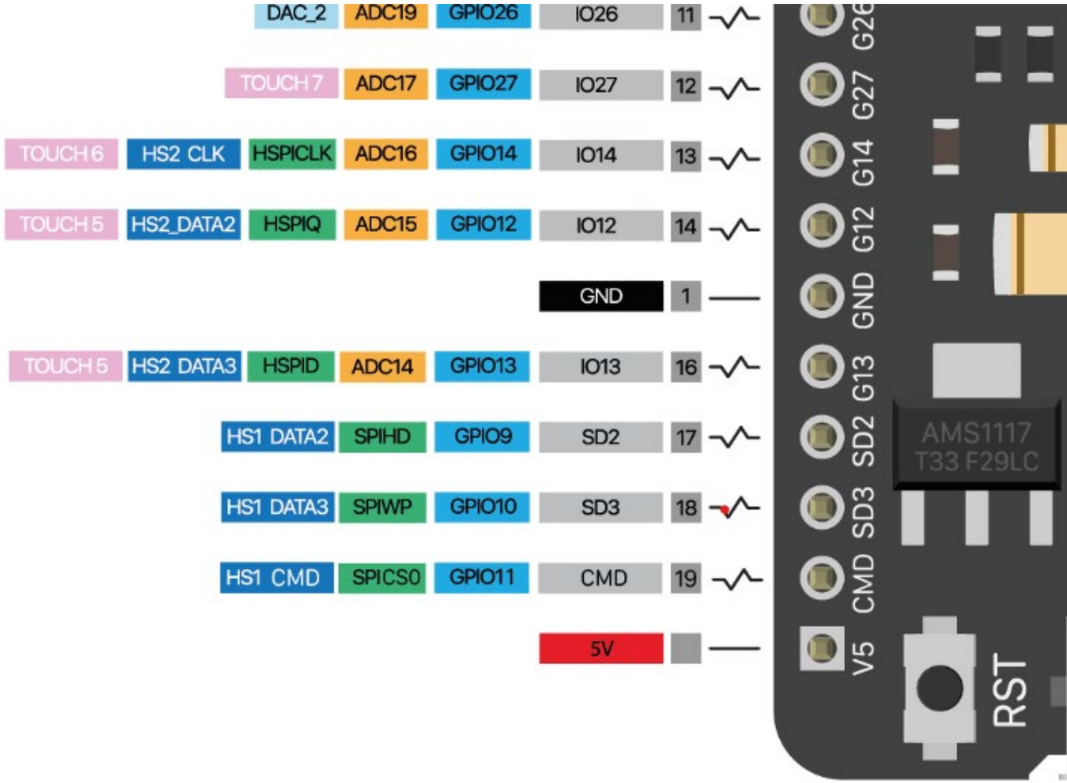
1. Install the ESP32servo lib from the library manager



Using a Servo lib

2. Wiring



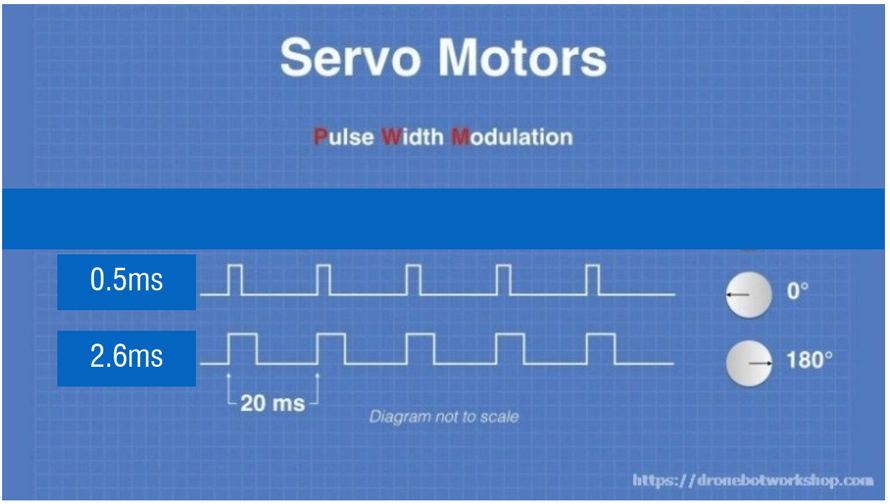


Wire connection: Power (Red) | GND (Brown) | Signal (Orange)

5V GND GPIO 13

Using a Servo lib

3. Coding



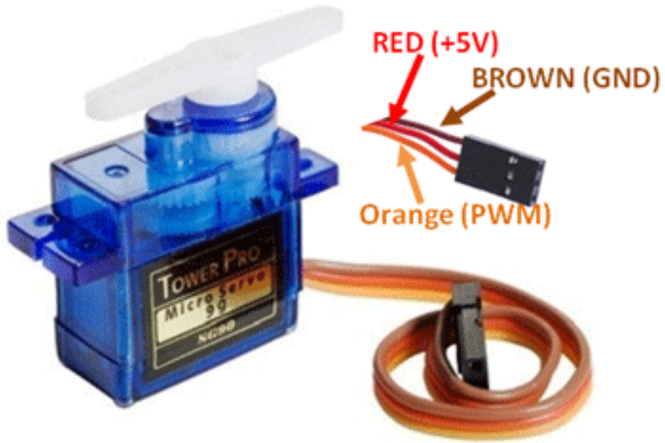
ESPServo Sweep.ino

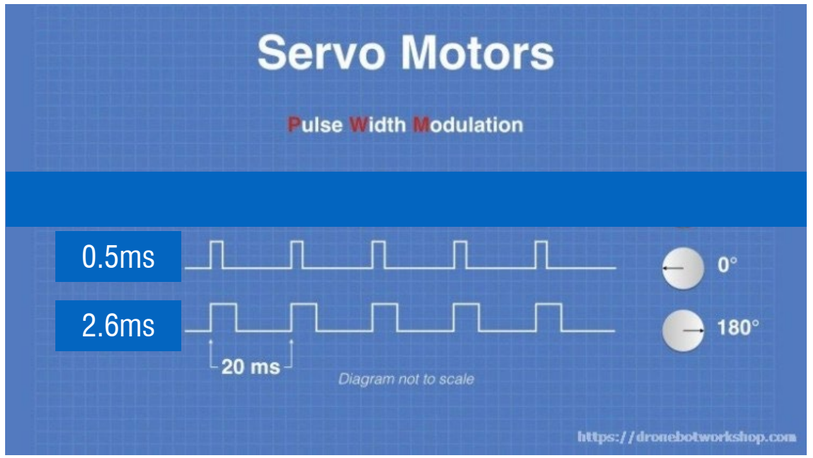
```
#include <ESP32Servo.h>

Servo myservo;

int pos = 0;
int servoPin = 13;


void setup() {
  myservo.attach(servoPin, 500, 2600); // attaches the servo on pin 13 to the servo object
  // different servos may require different min/max settings
  // for an accurate 0 to 180 sweep
}

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
  }
}


```

Pulse Width Modulation (PWM)

Can you control the servo motor without using Lib but just PWM?

Hint:





For our servo motor:

Pulse width: 20ms

0.5ms -> 0 degree

2.6ms -> 180 degree

Supporting functions you might need:

map(x, fromA, toA, fromB, toB);

delayMicroseconds(x);

ESPServo Sweep.ino

```
#include <ESP32Servo.h>

Servo myservo;

int pos = 0;
int servoPin = 13;


void setup() {
  myservo.attach(servoPin, 500, 2600); // attaches the servo on pin 13 to the servo object
  // different servos may require different min/max settings
  // for an accurate 0 to 180 sweep
}

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
  }
}


```

ESPServo Sweep NoLib.ino

```
1
2
3   int pos = 0;
4   int servoPin = 13;
5
6
7   void setup() {
8     pinMode(servoPin, OUTPUT);
9
10  }
11
12  void loop() {
13    for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
14      // in steps of 1 degree
15      angle(pos);    // tell servo to go to position in variable 'pos'
16      delay(15);             // waits 15ms for the servo to reach the position
17    }
18    for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
19      angle(pos);    // tell servo to go to position in variable 'pos'
20      delay(15);             // waits 15ms for the servo to reach the position
21    }
22  }
23
```

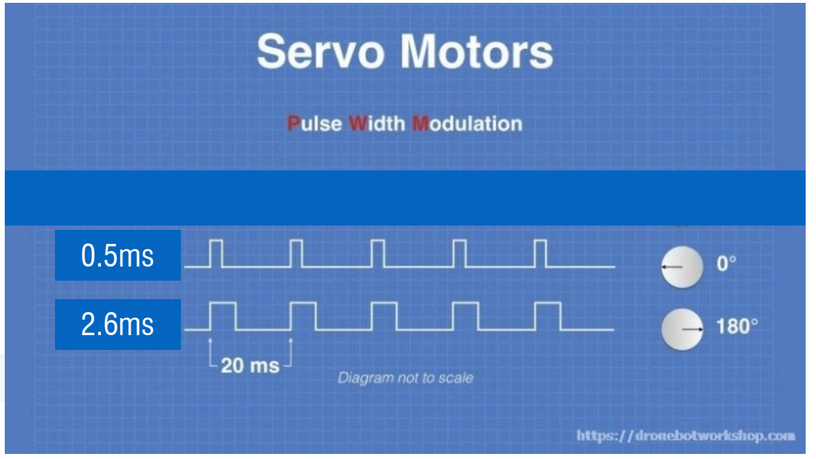
ESPServo Sweep NoLib.ino

```
1
2
3   int pos = 0;
4   int servoPin = 13;
5
6
7   void setup() {
8     pinMode(servoPin, OUTPUT);
9
10  }
11
12  void loop() {
13    for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
14      // in steps of 1 degree
15      angle(pos);    // tell servo to go to position in variable 'pos'
16      delay(15);             // waits 15ms for the servo to reach the position
17    }
18    for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
19      angle(pos);    // tell servo to go to position in variable 'pos'
20      delay(15);             // waits 15ms for the servo to reach the position
21    }
22  }
23
24  void angle (int a)
25  {
26
27  }
```

Supporting functions you might need:

Map(x, fromA, toA, fromB, toB);

delayMicroseconds(x);

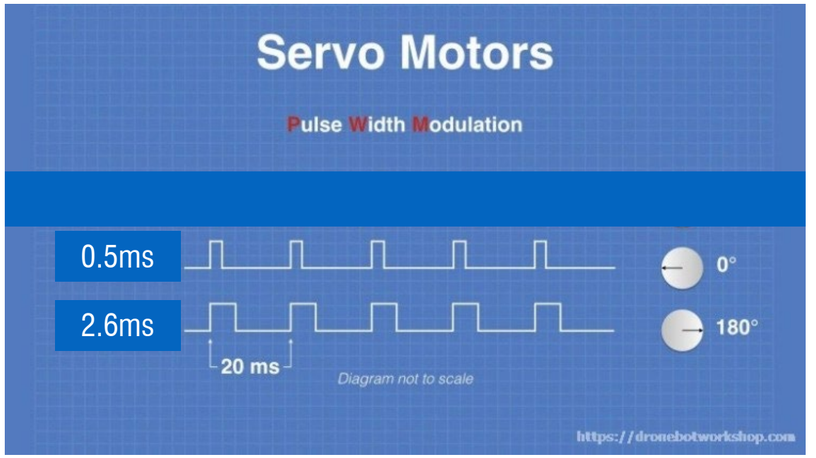


```
void angle (int a)
{
  int highTime = map(a, 0, 180, 500, 2600);
  int delayTime = 20000 - highTime;

  digitalWrite(servoPin, HIGH);
  delayMicroseconds(highTime);
  digitalWrite(servoPin, LOW);
  delayMicroseconds(delayTime);
}
```

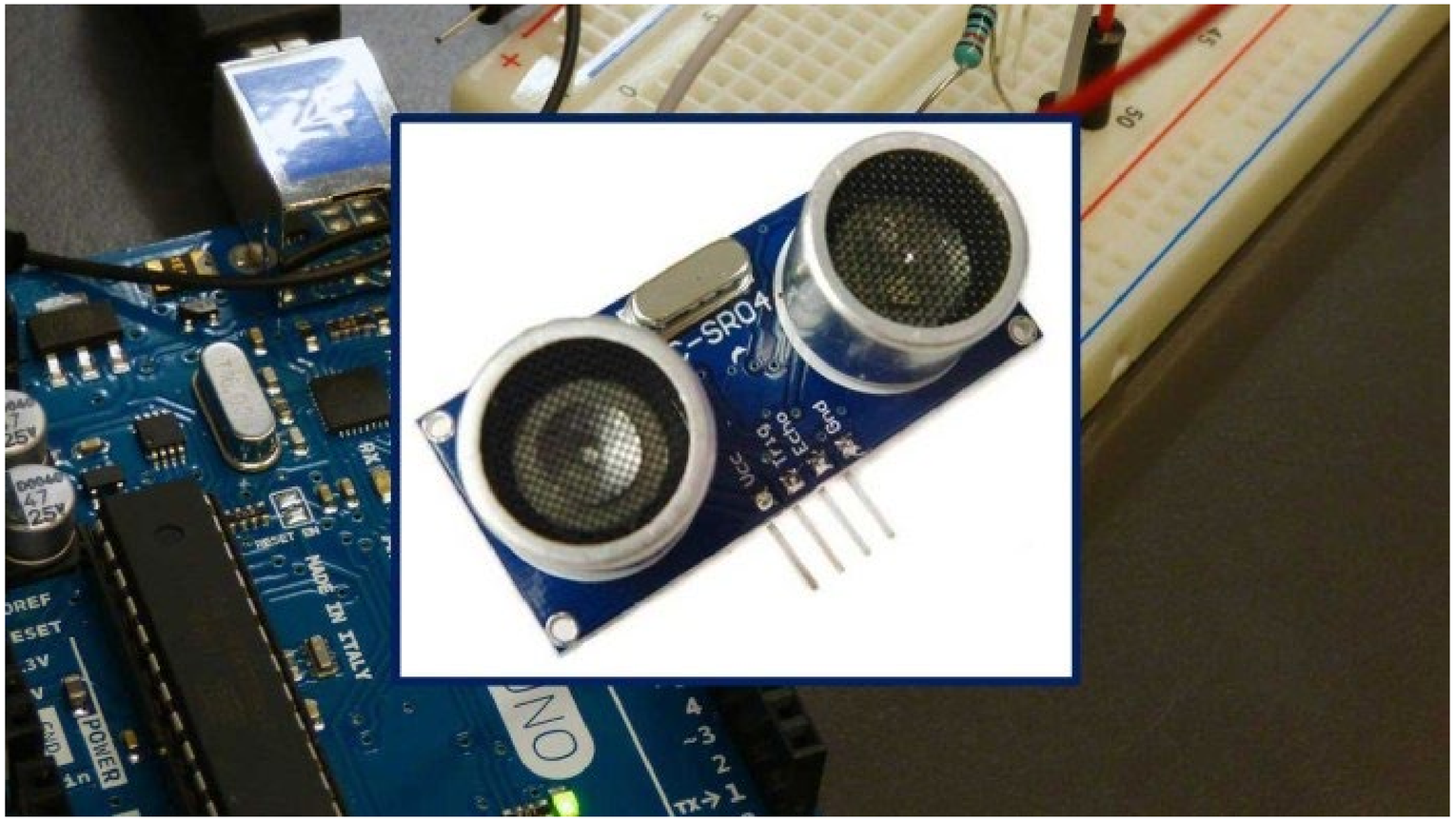
Supporting functions you might need:

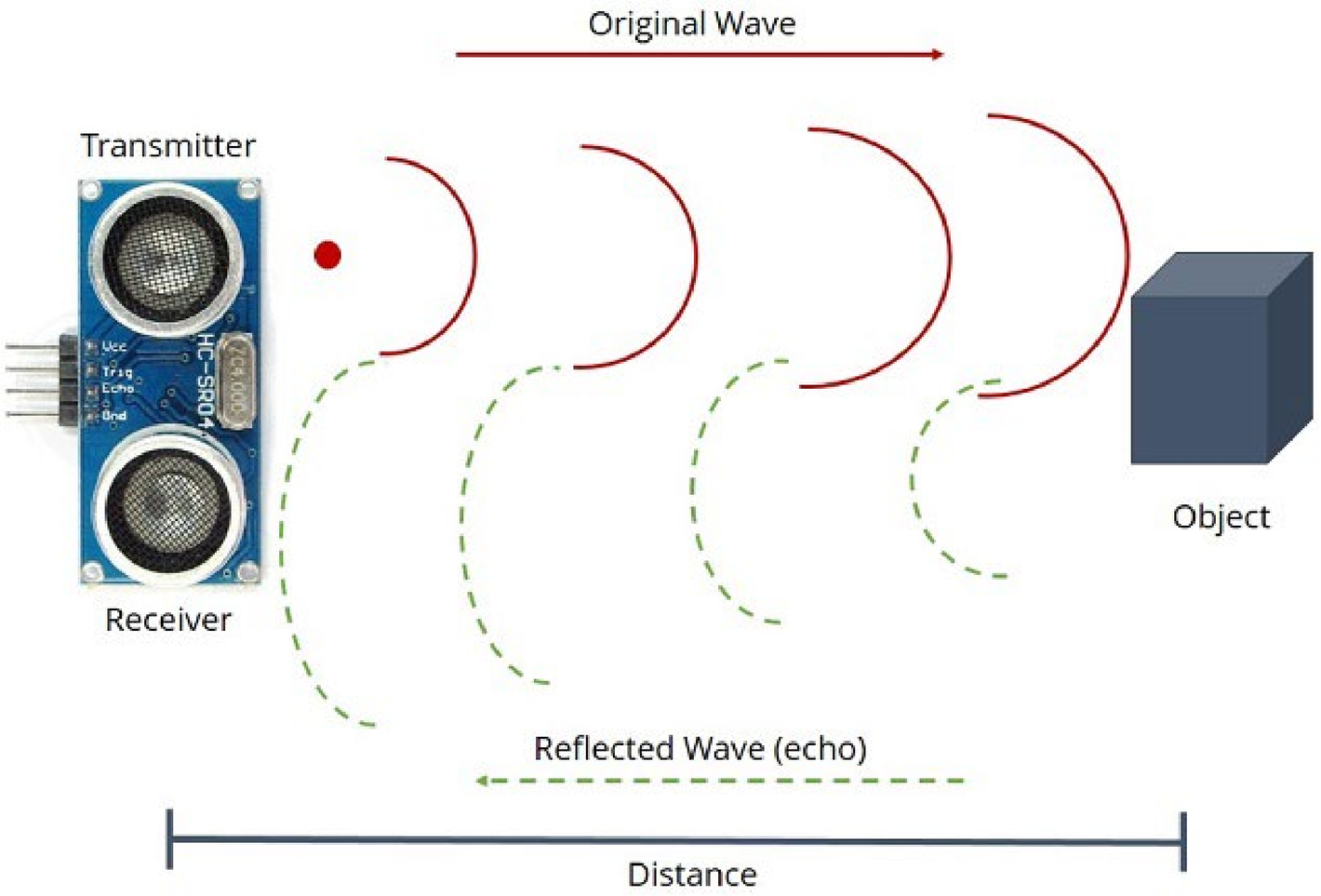
Map(x, fromA, toA, fromB, toB);

delayMicroseconds(x);

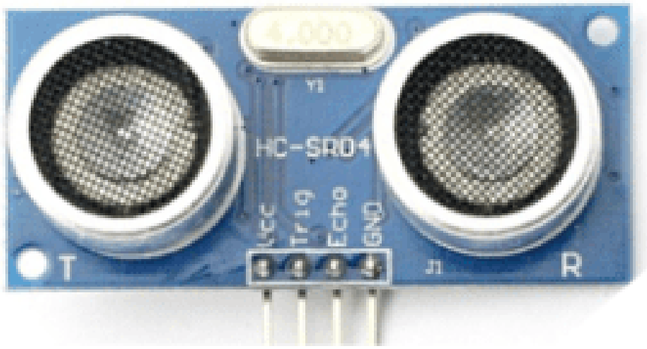


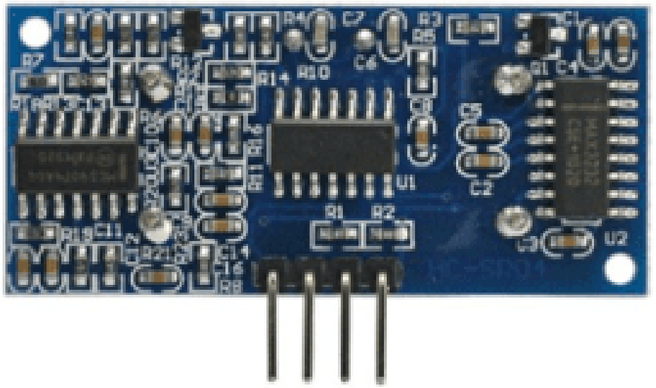
Ultrasonic Sensor HC – SR04





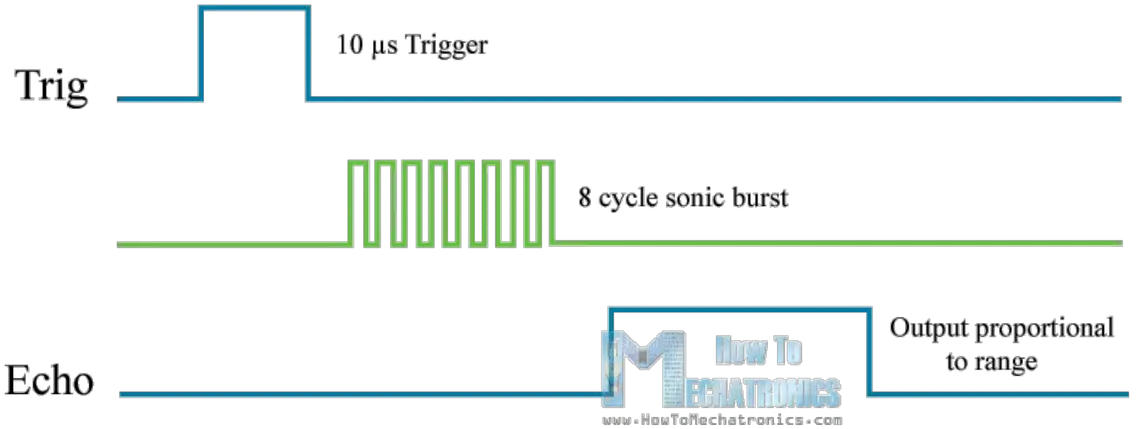
The time between the transmission and reception of the signal allows us to calculate the distance to an object. This is possible because we know the sound's velocity in the air





VCC: +5VDC
Trig : Trigger (**OUTPUT**)
Echo: Echo (**INPUT**)
GND: GND

Arduino has library for it, but Can you do it without library?

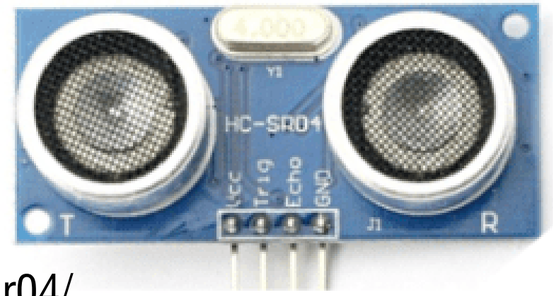


Hint:

a. The sensor is triggered by a HIGH pulse of 10 or more microseconds.

b. To read the sensing signal: a HIGH pulse whose duration is the time (in microseconds) from the sending of the ping to the reception of its echo off of an object.

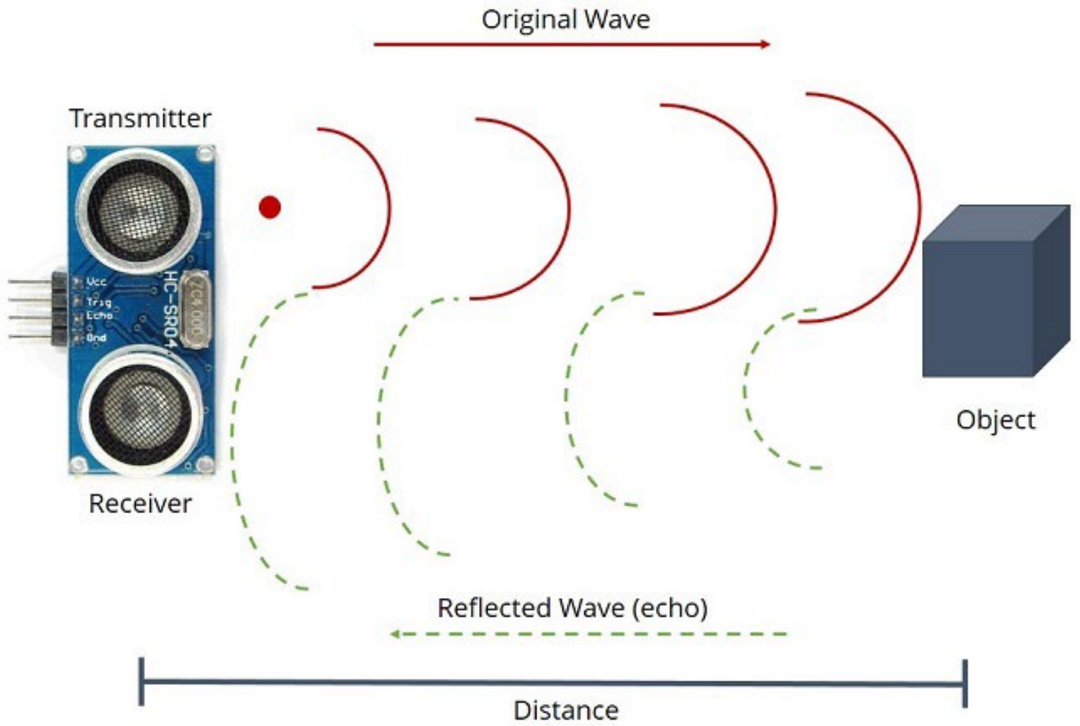
Google: pulseIn()



VCC: +5VDC
Trig : Trigger (**OUTPUT**)
Echo: Echo (**INPUT**)
GND: GND

https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

VCC: +5VDC
Trig : Trigger (**OUTPUT**)
Echo: Echo (**INPUT**)
GND: GND




```
int dist()
{
  long duration, distance = -1;

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) * 0.0343;

  if(distance >= 20)
  {
    Serial.println("Out of range");
  }
  else
  {
    Serial.print(distance);
    Serial.println(" cm");
  }
  return distance;
}
```

Assignment:

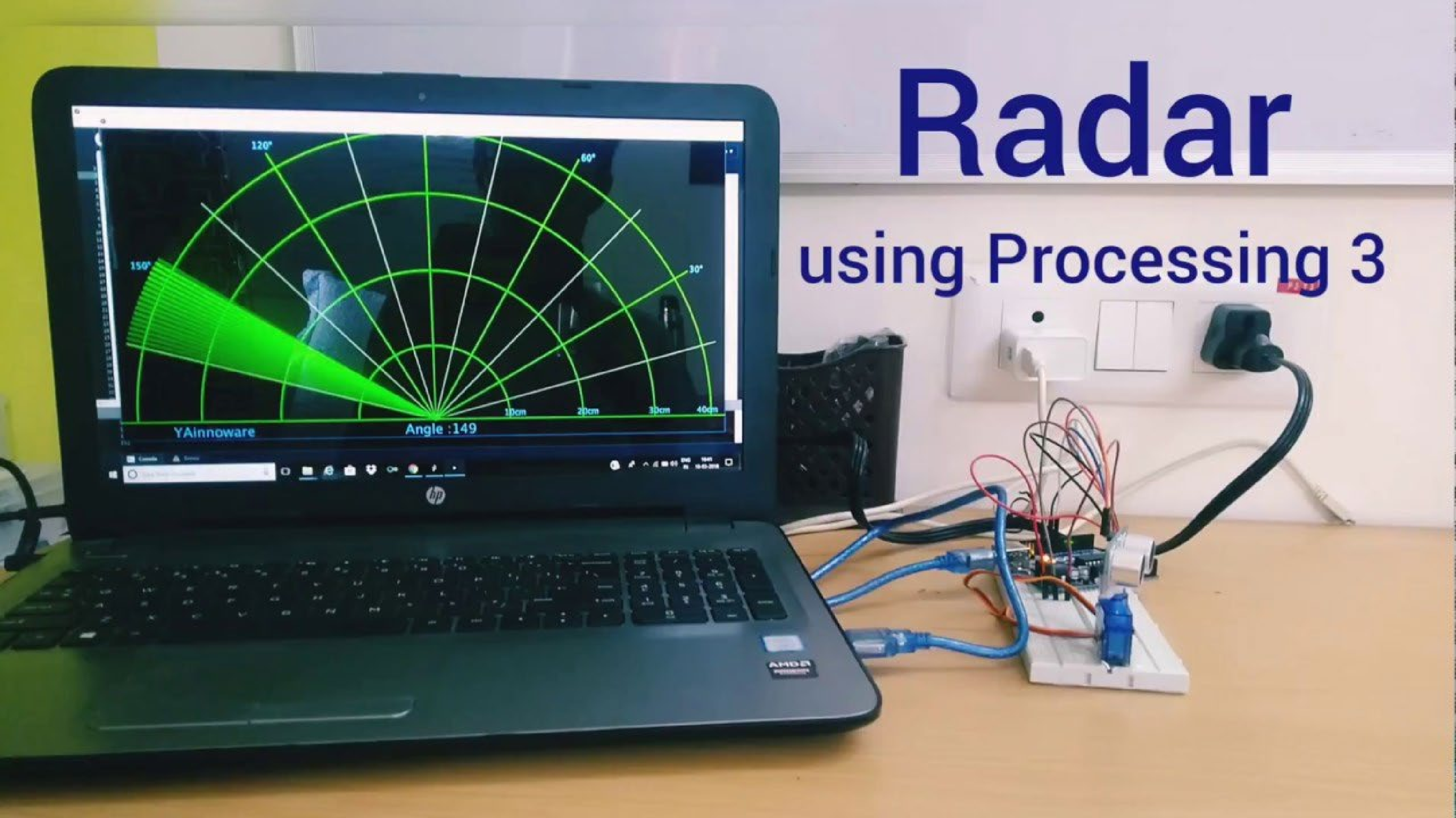
Make a simple radar with the servo motor and the ultrasonic sensor
Code should not use the servo library.

You can use duct tape/hot glue/rubber band/screws from the sandbox to ensure the ultrasonic sensor is firmly attached to the servo motor.

Submission:
Code + Video

In the video, please put some obstacles in front of your radar and show the distance reading changes from the serial port.

Optional: write a GUI with processing. You will get up to 5 bonus points in Assignment 4 for the GUI processing program.


Radar
using Processing 3
120°
60°
150°
30°
10cm
20cm
30cm
40cm
YAinnoware
Angle :149