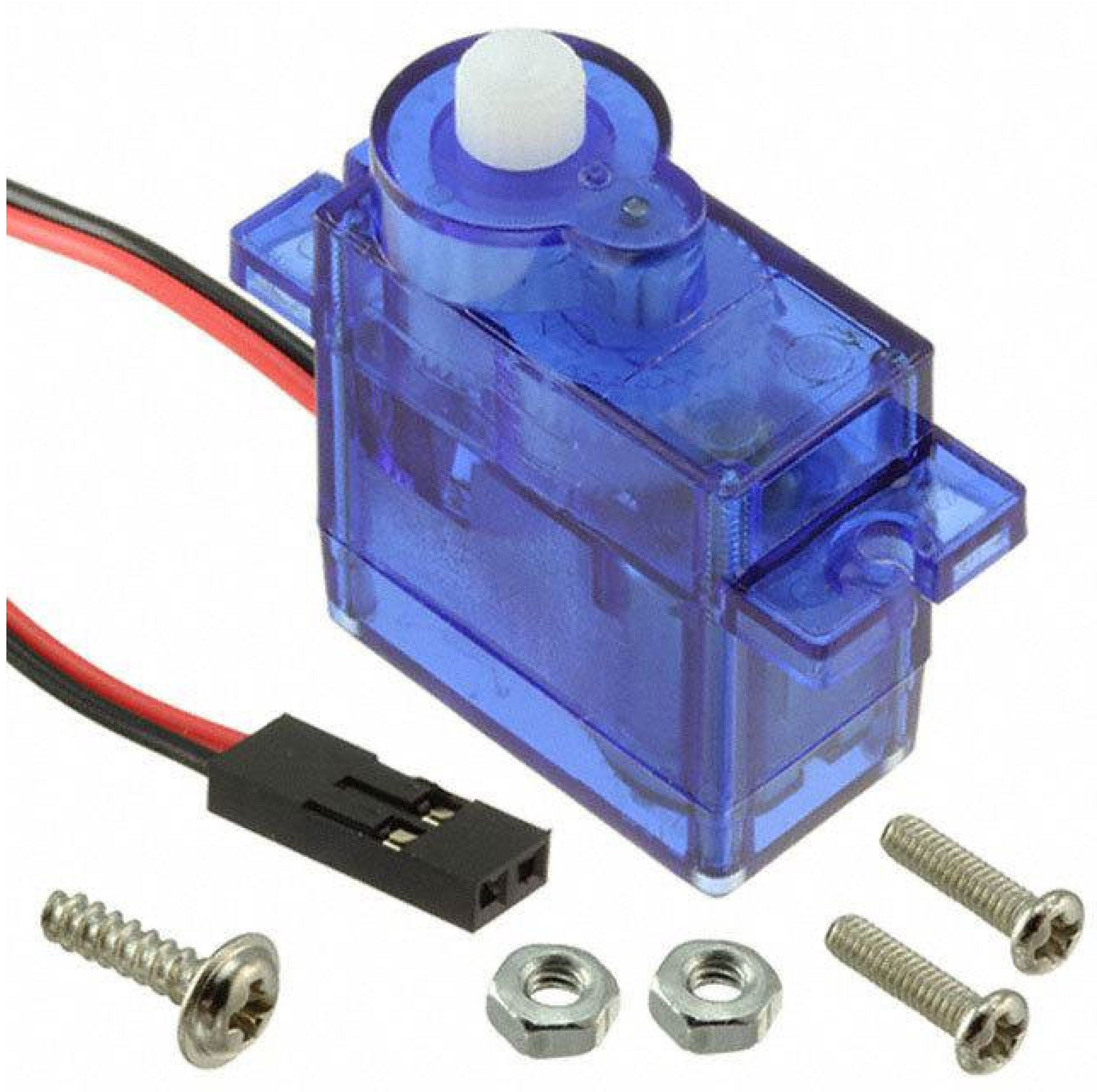
Servo Motor+ Ultrasonic Sensor

Huaishu Peng | UMD CS | Fall 2022



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?

Channel Creative
Youtube.com/ChannelCreativeVN
Kênh Sáng Tạo .COM
L
R
SELECT
MODE
START
Channel Creative
Youtube.com/ChannelCreativeVN
Facebook.com/ChannelCreative

What is a servo motor?

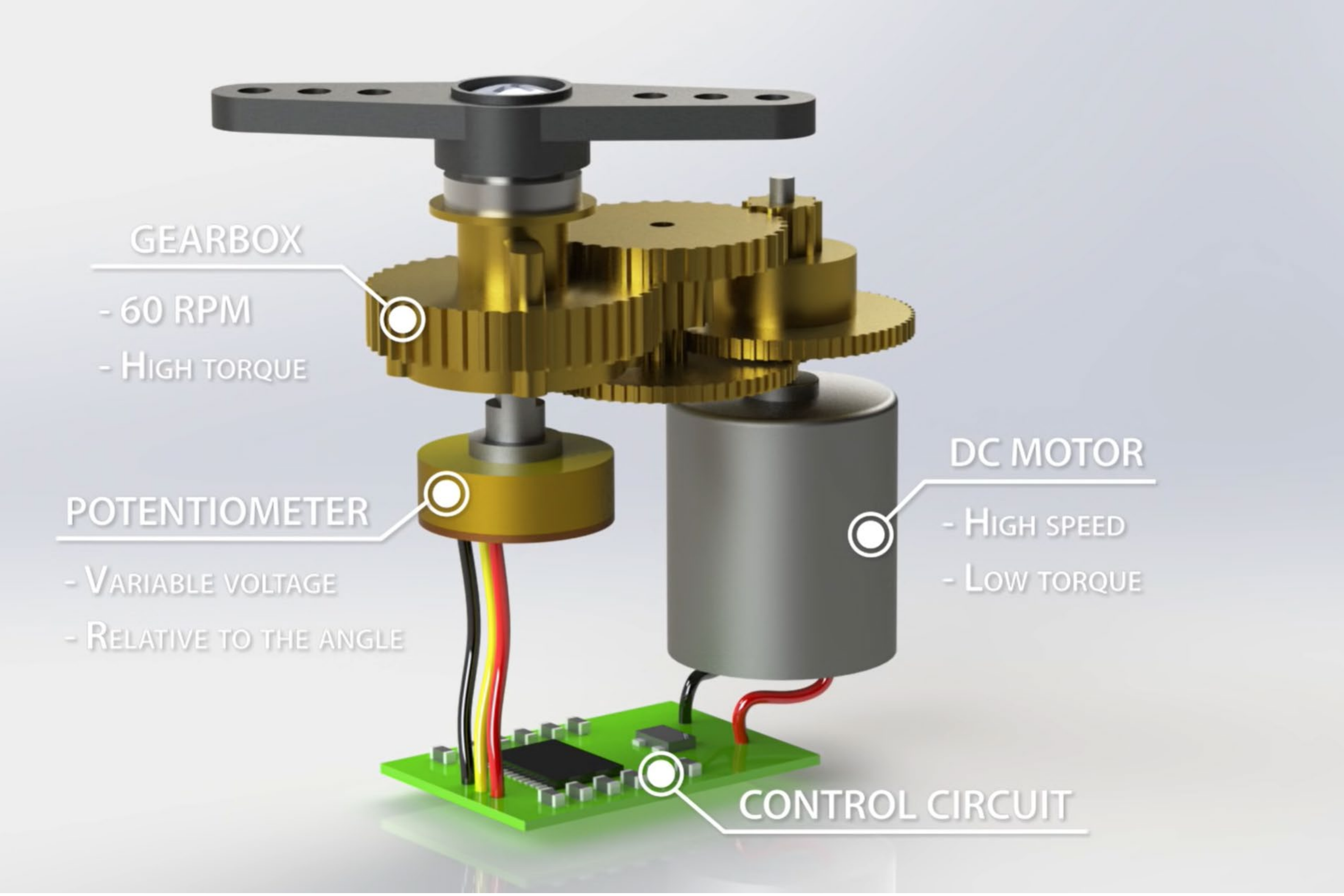
a motor with a built-in “servomechanism”.

Consist of:

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

A feedback device

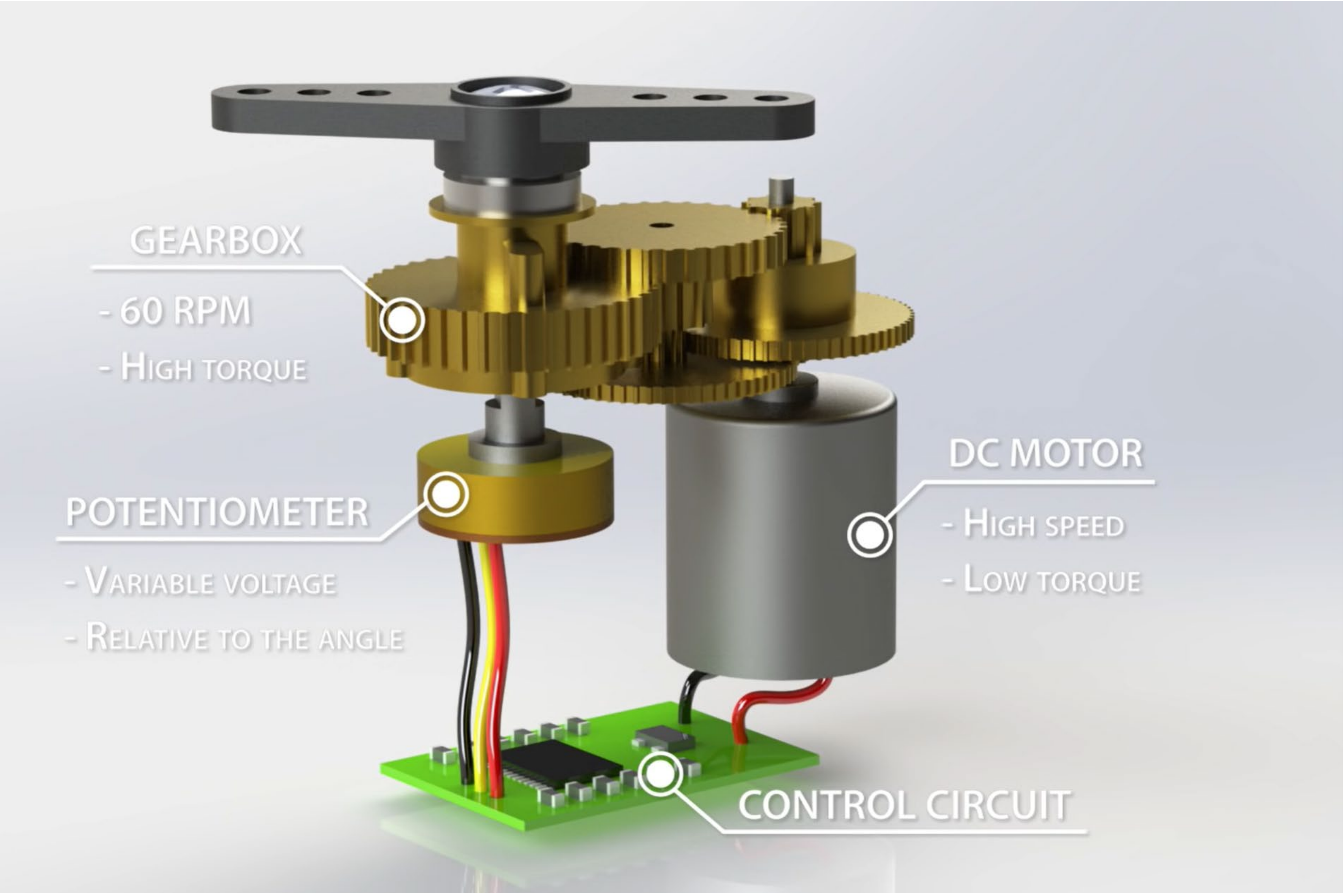
An electronic controller



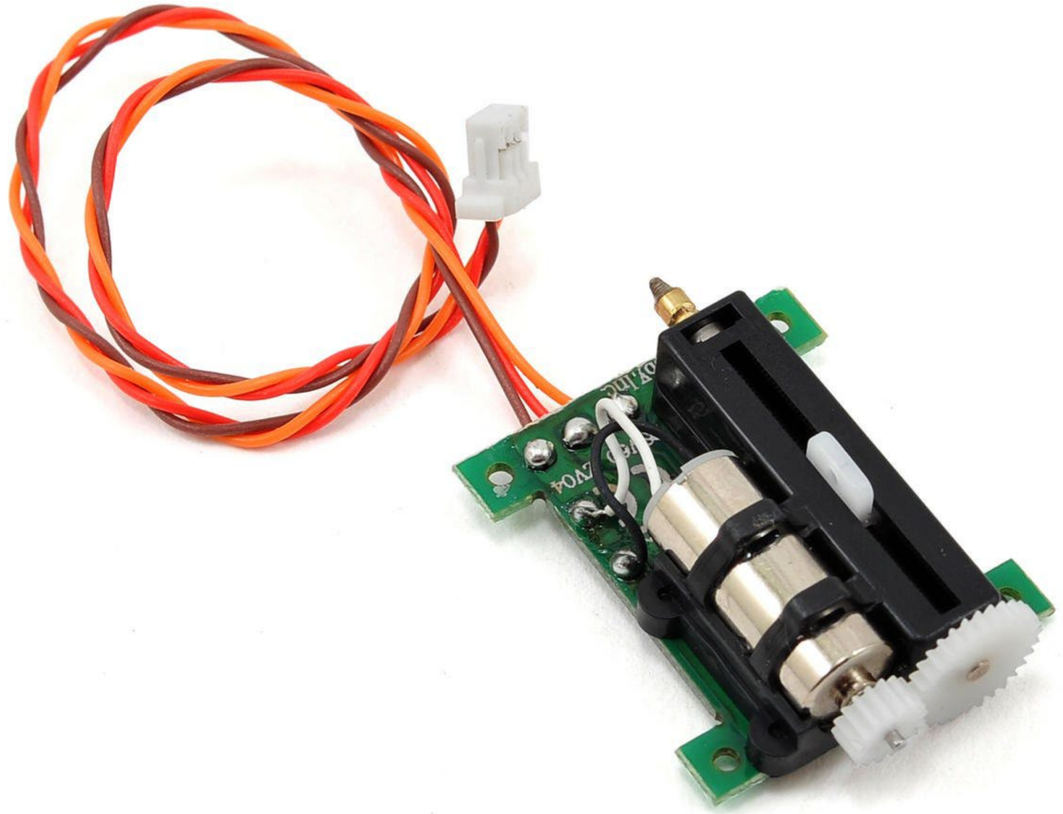
What is a servo motor?

Motors can be DC, or AC

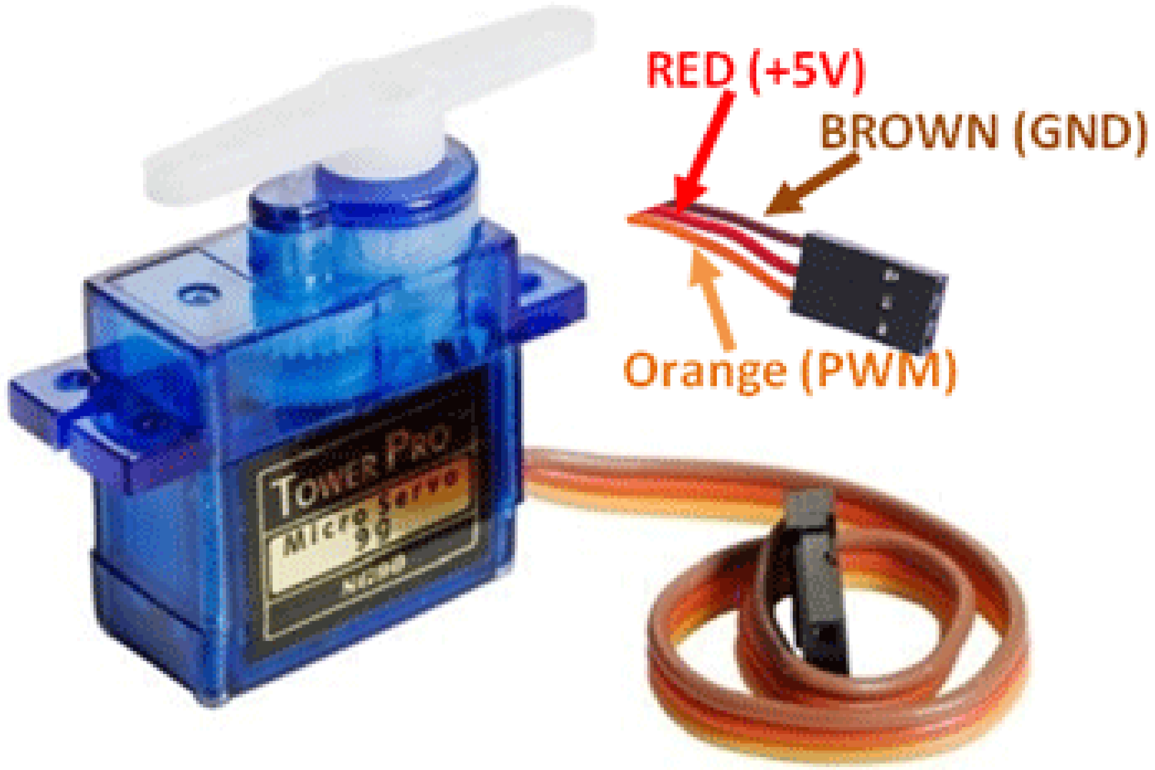
Feedback device can be encoder or other sensors for position sensing



Micro Servo 9g
SG90



Panasonic



Servo Motor

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

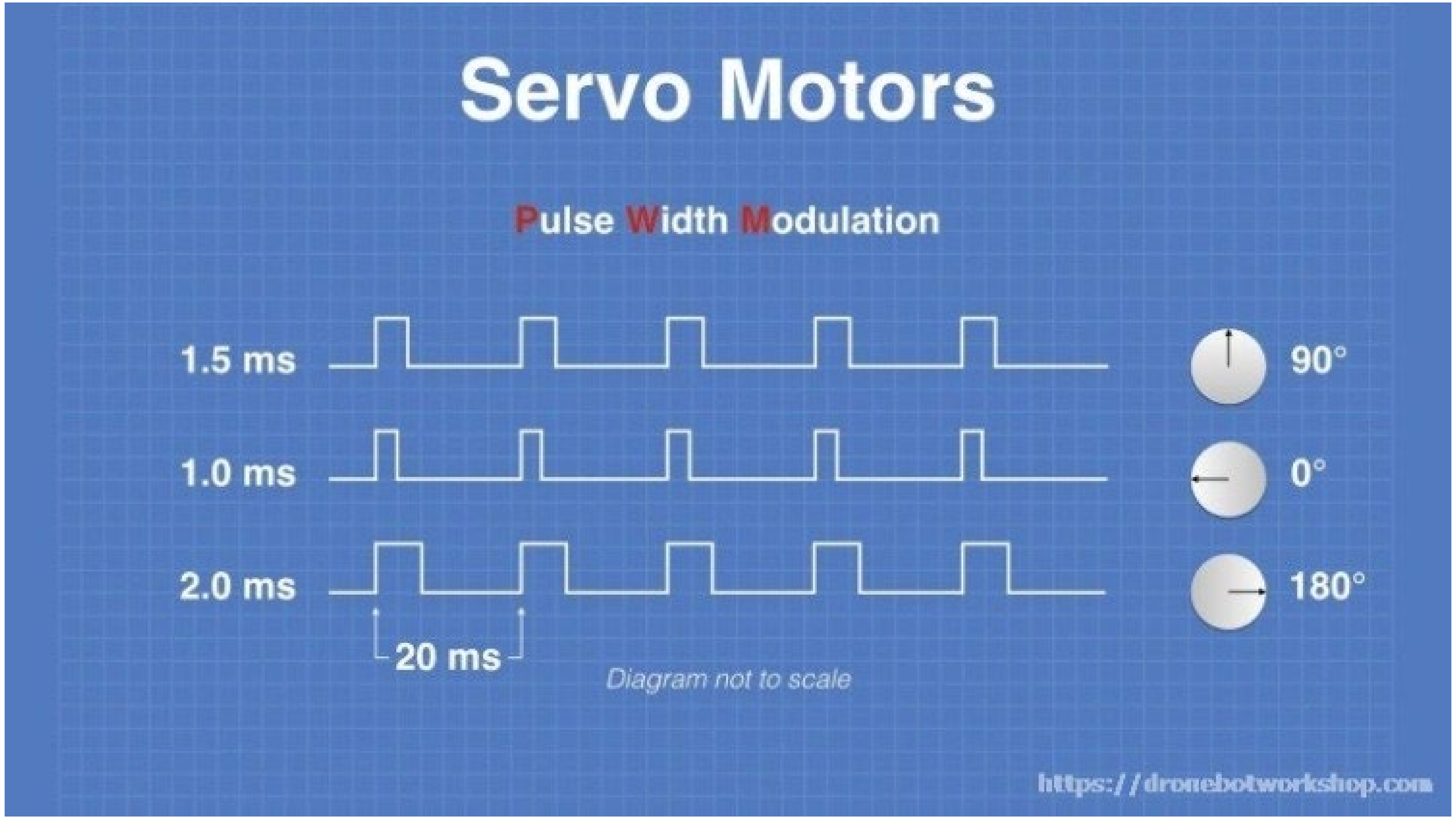
Rotational range: 0 ~ 180 degree

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

Operating Voltage: **5**V

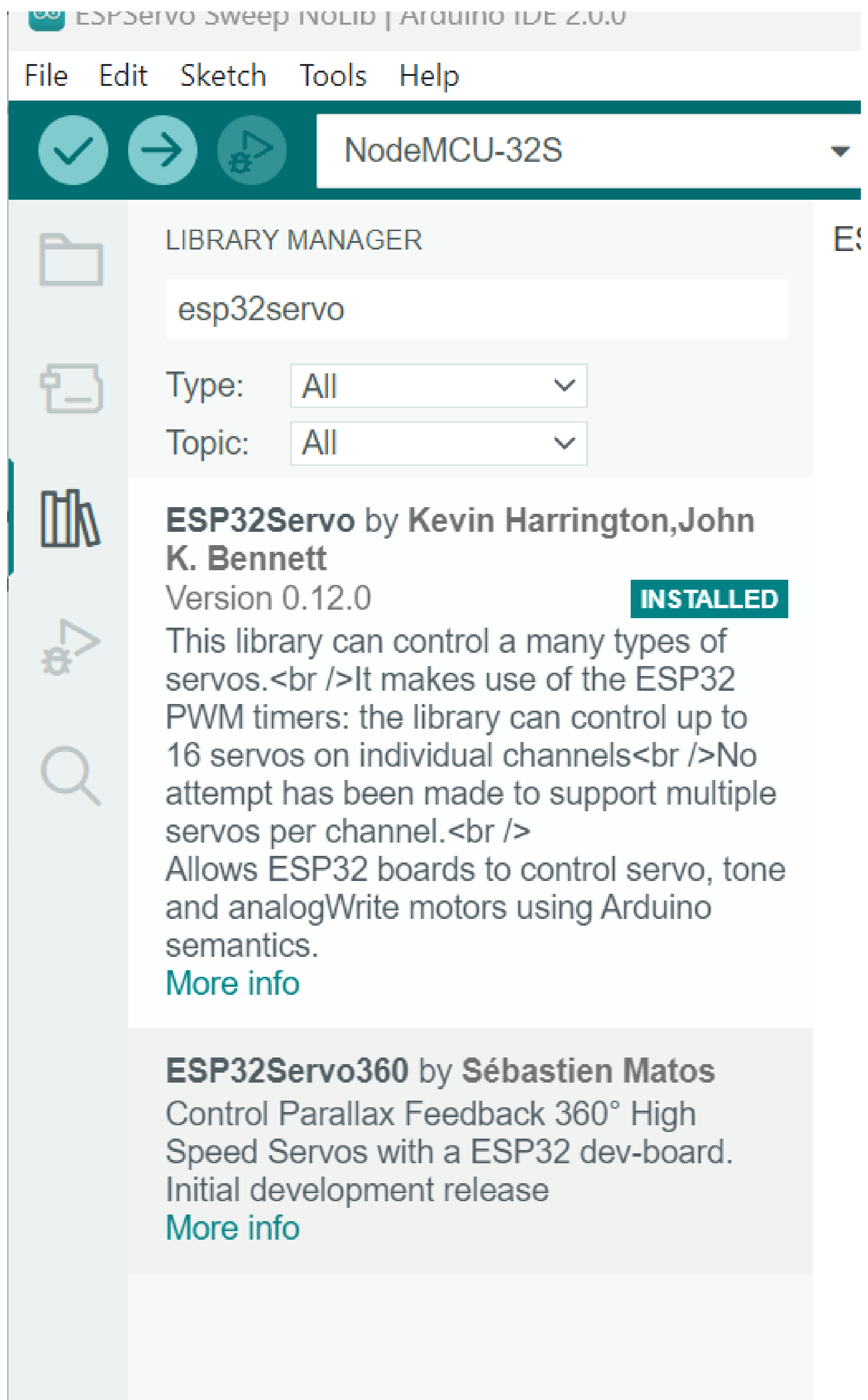
Pulse Width Modulation (PWM)

Controlling the position of a servo motor



Using a Servo lib

1. Install the ESP32servo lib from the library manager



NodeMCU-32S

esp32servo

Type: All

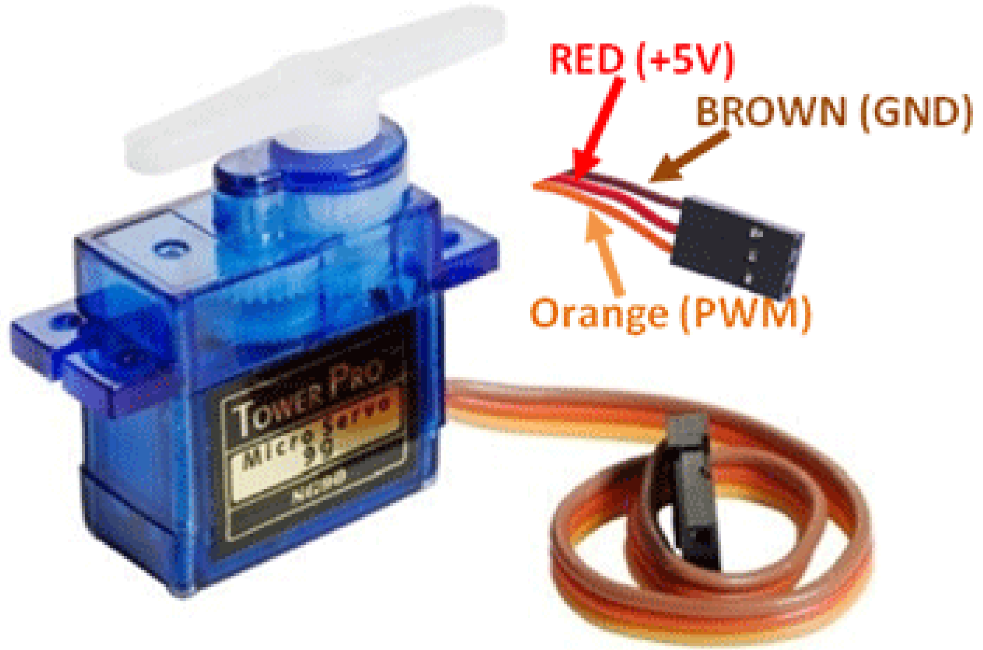
Topic: All

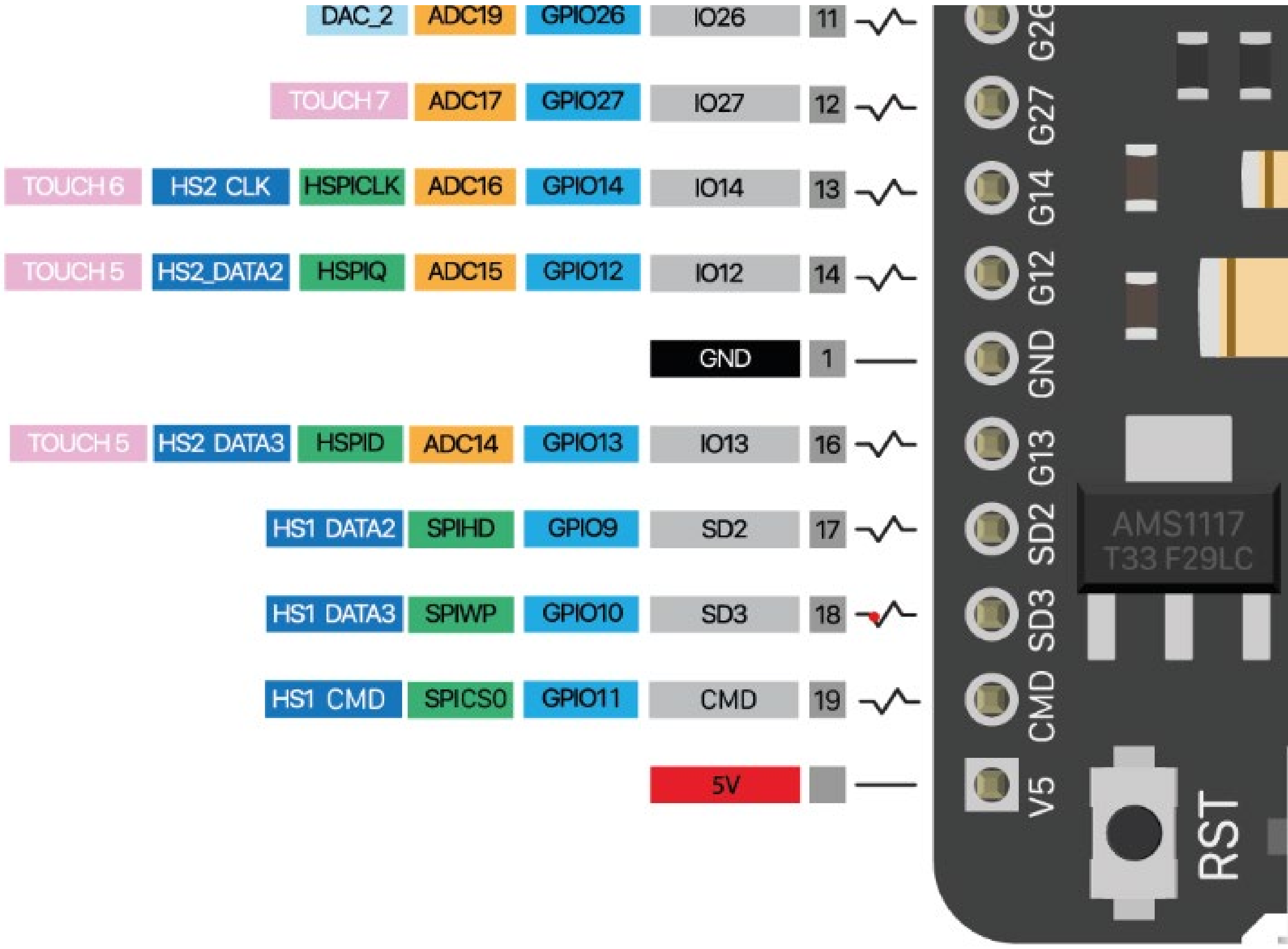
ESP32Servo by **Kevin Harrington,John K. Bennett**
Version 0.12.0
This library can control a many types of servos.<br />It makes use of the ESP32 PWM timers: the library can control up to 16 servos on individual channels<br />No attempt has been made to support multiple servos per channel.<br />
Allows ESP32 boards to control servo, tone and analogWrite motors using Arduino semantics.

ESP32Servo360 by **Sébastien Matos**
Control Parallax Feedback 360° High Speed Servos with a ESP32 dev-board.
Initial development release

Using a Servo lib

2. Wiring



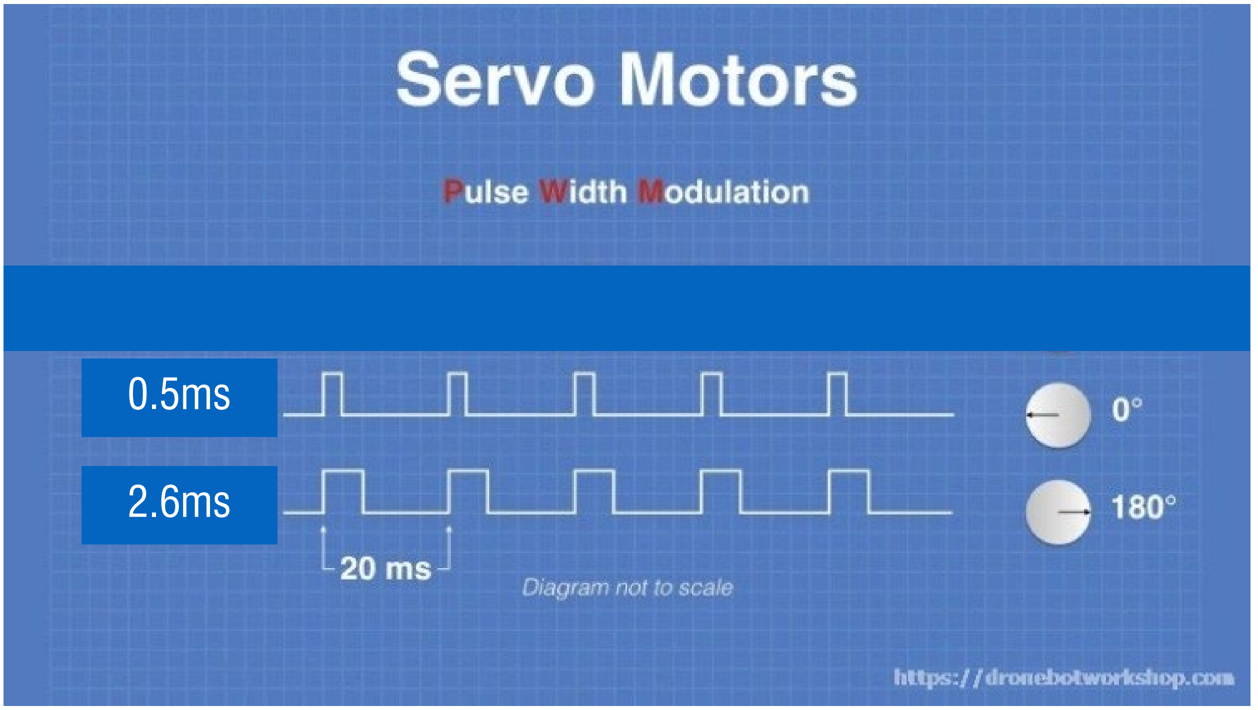

DAC_2 ADC19 GPIO26 IO26 11 G26
TOUCH7 ADC17 GPIO27 IO27 12 G27
TOUCH6 HS2 CLK HSPICLK ADC16 GPIO14 IO14 13 G14
TOUCH5 HS2_DATA2 HSPIQ ADC15 GPIO12 IO12 14 G12
GND 1 GND
TOUCH5 HS2 DATA3 HSPID ADC14 GPIO13 IO13 16 G13
HS1 DATA2 SPIHD GPIO9 SD2 17 SD2
HS1 DATA3 SPIWP GPIO10 SD3 18 SD3
HS1 CMD SPICS0 GPIO11 CMD 19 CMD
5V V5
AMS1117
T33 F29LC
RST

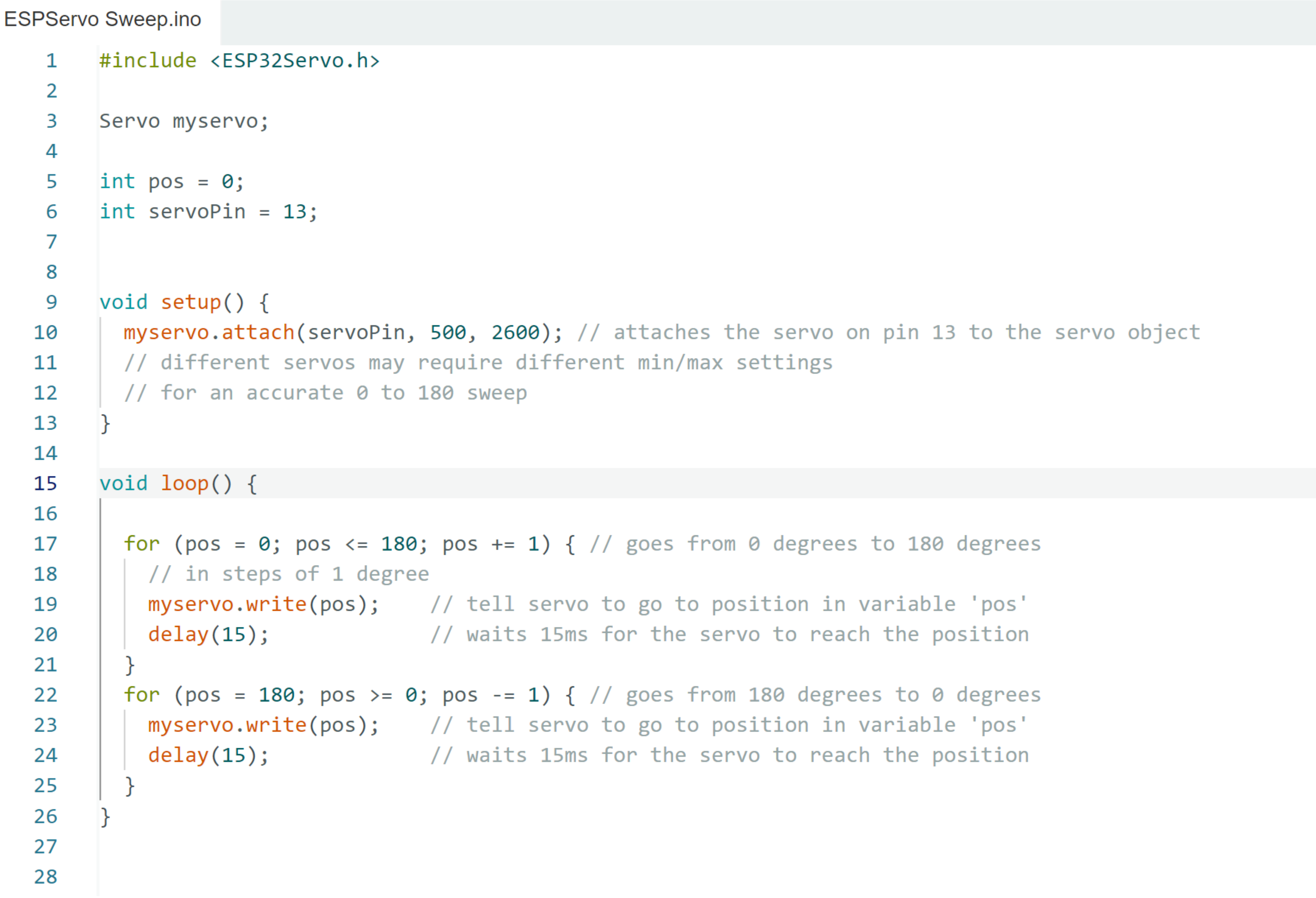
Wire connection: Power (Red) | GND (Brown) | Signal (Orange)

5V GND GPIO 13

Using a Servo lib

3. Coding



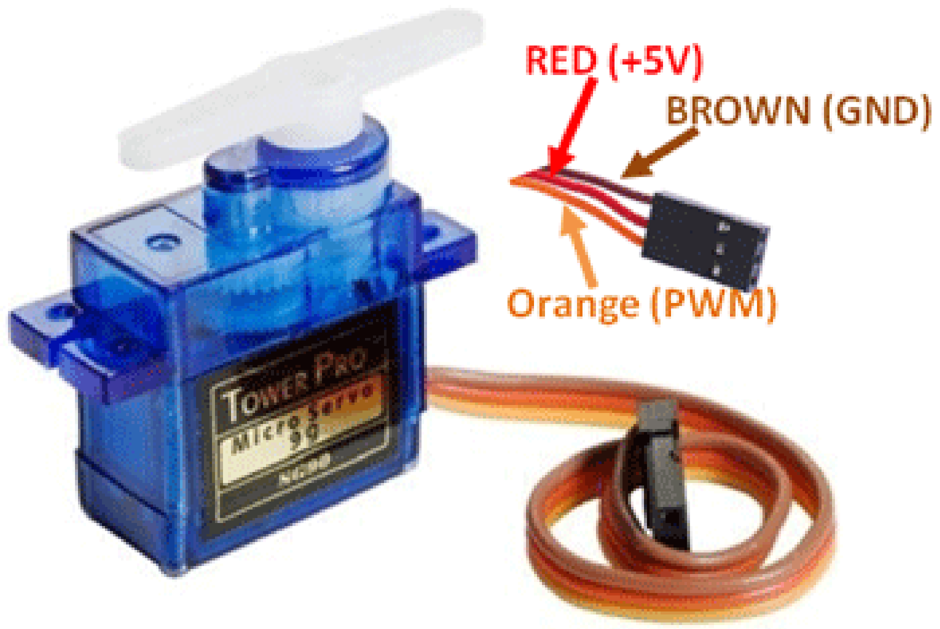

ESPServo Sweep.ino

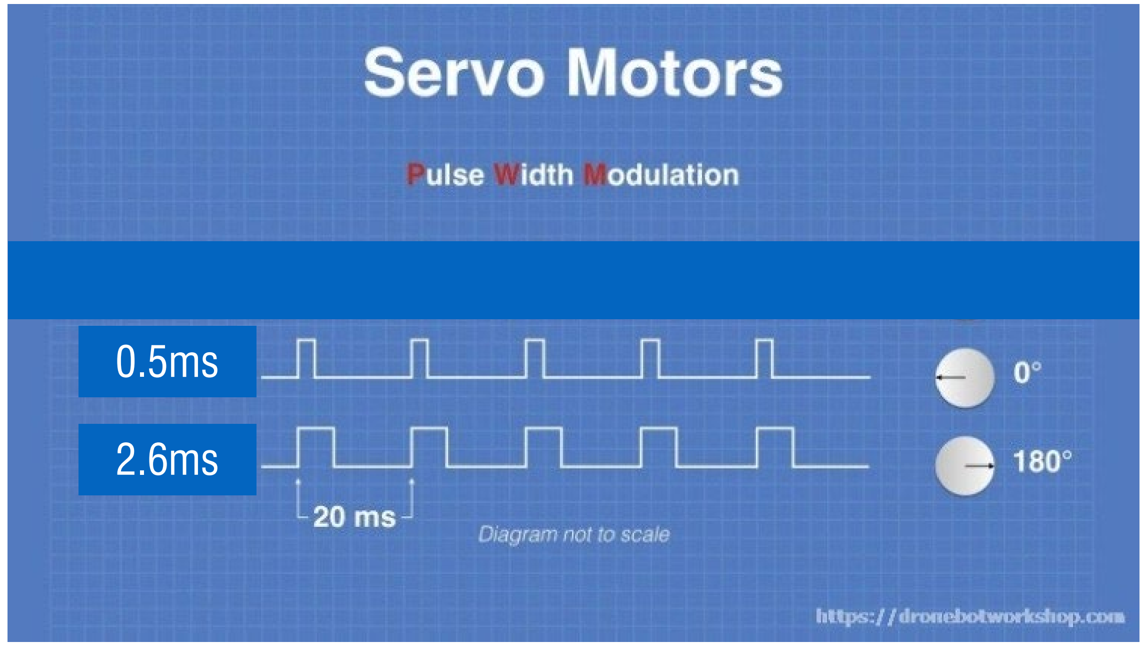
```
#include <ESP32Servo.h>

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

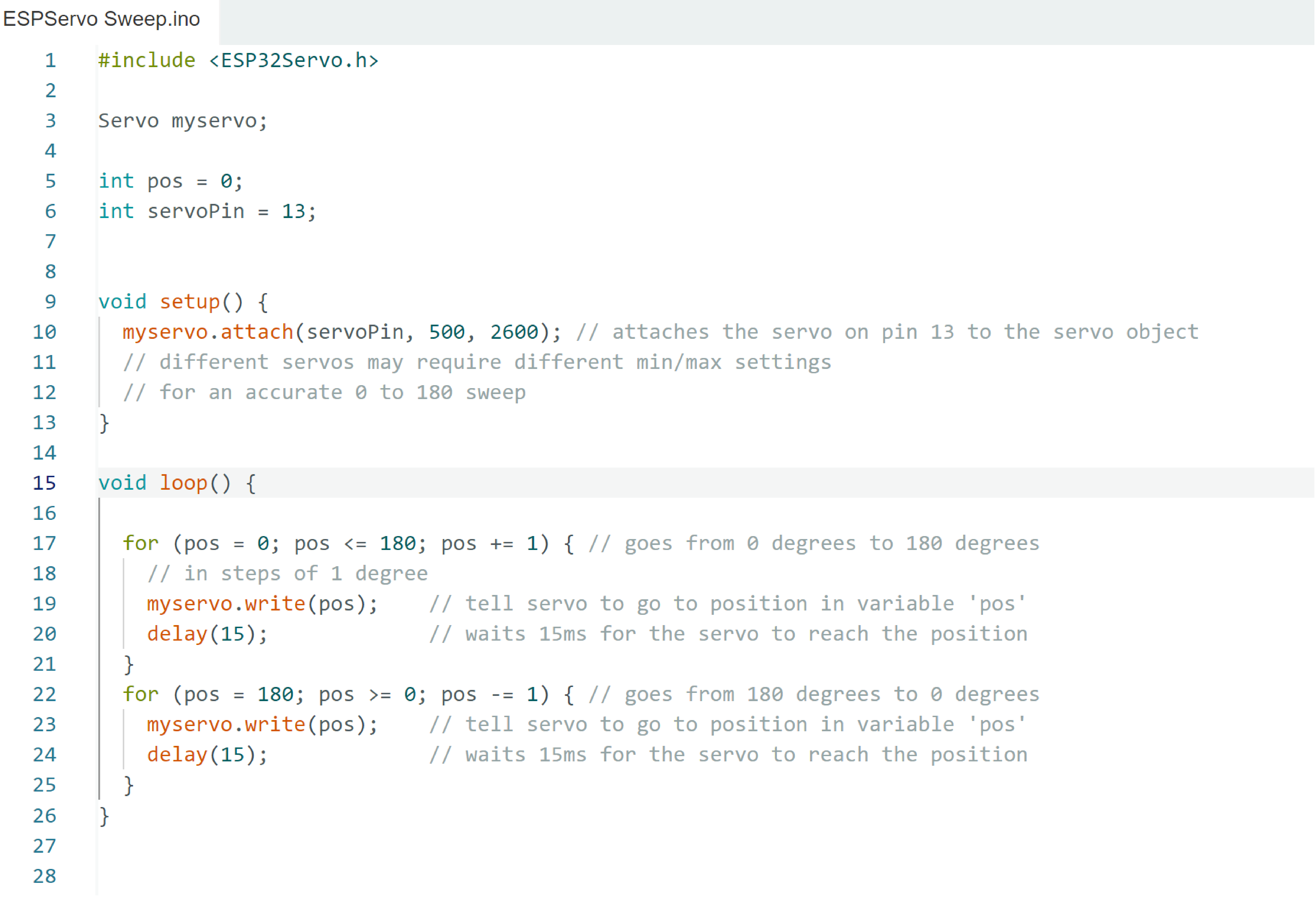
0.5ms -> 0 degree

2.6ms -> 180 degree

Supporting functions you might need:

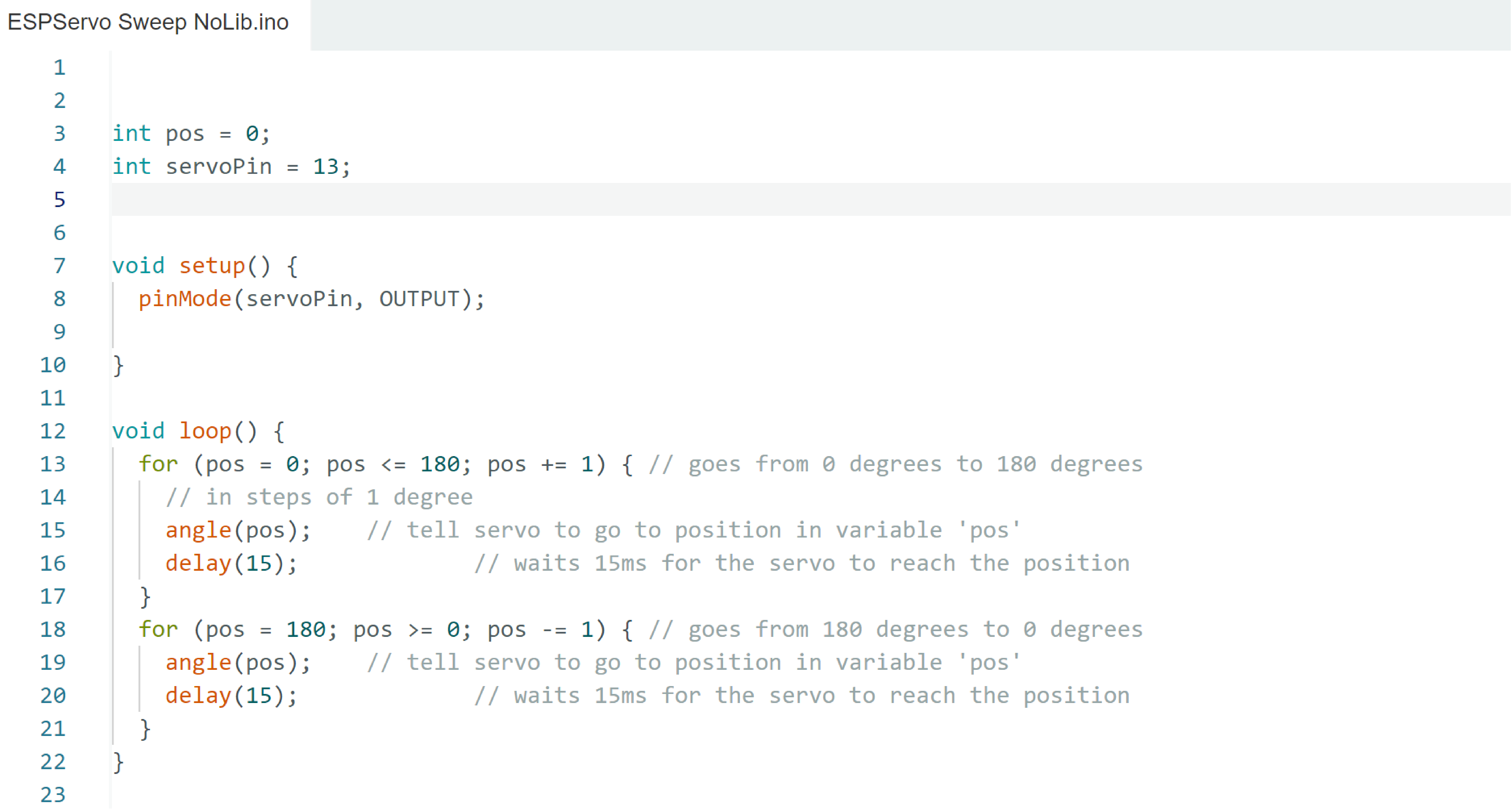
map(x, fromA, toA, fromB, toB);

delayMicroseconds(x);

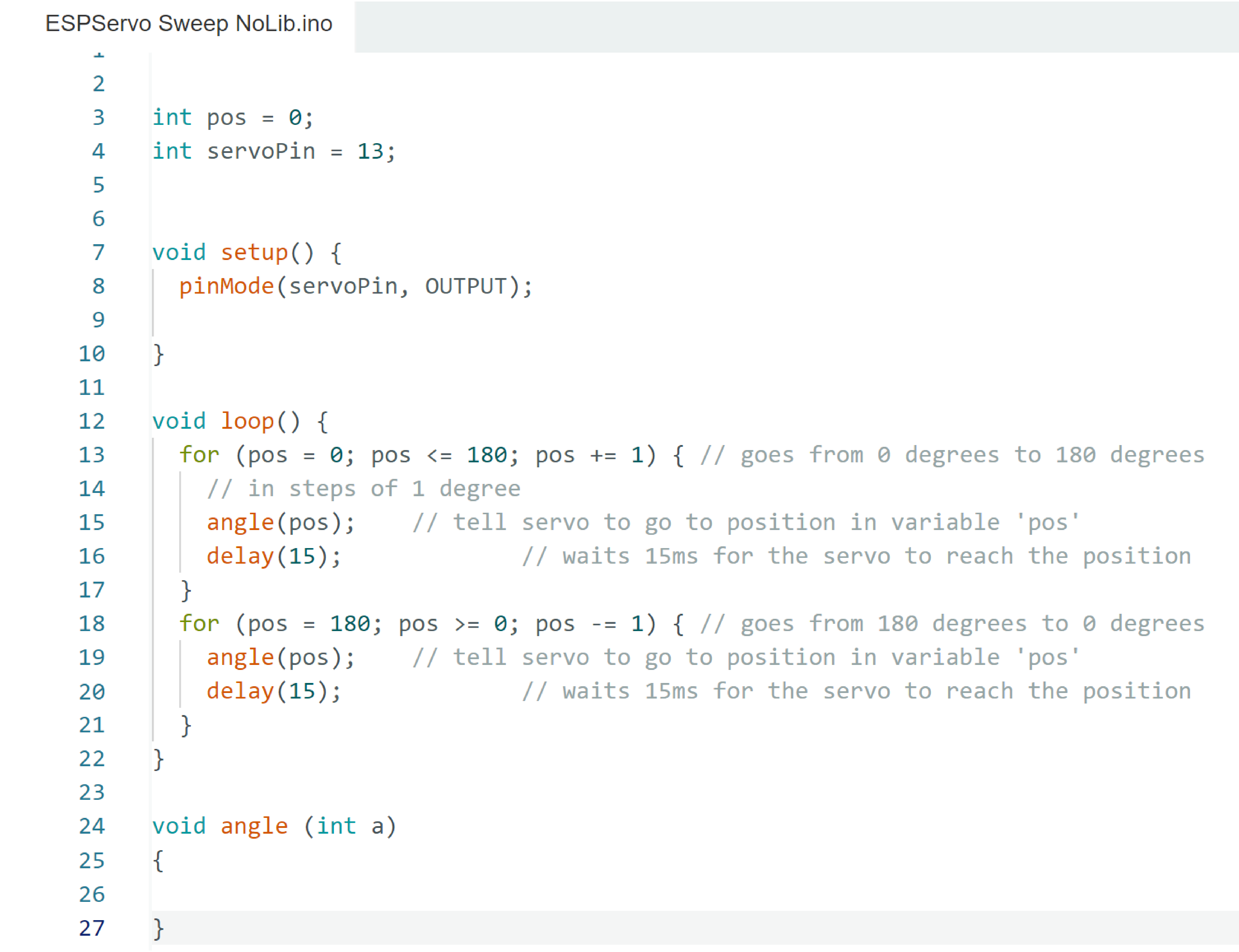


ESPServo Sweep.ino

```
#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
  }
}


```



```


int pos = 0;
int servoPin = 13;


void setup() {
  pinMode(servoPin, OUTPUT);

}

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

```



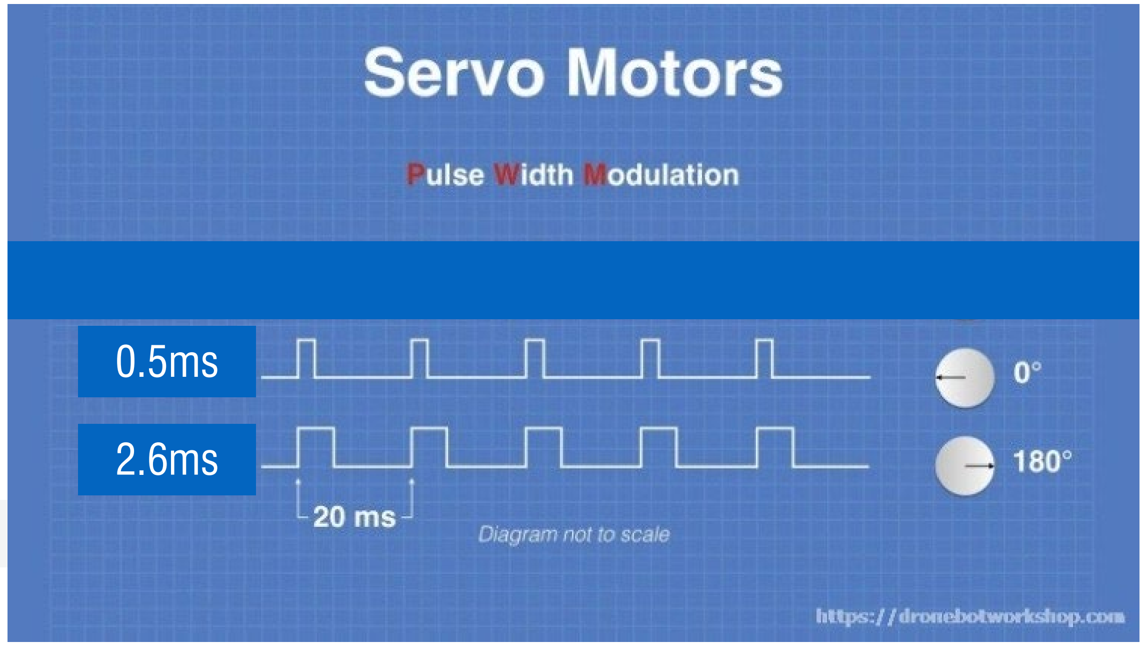
ESPServo Sweep NoLib.ino

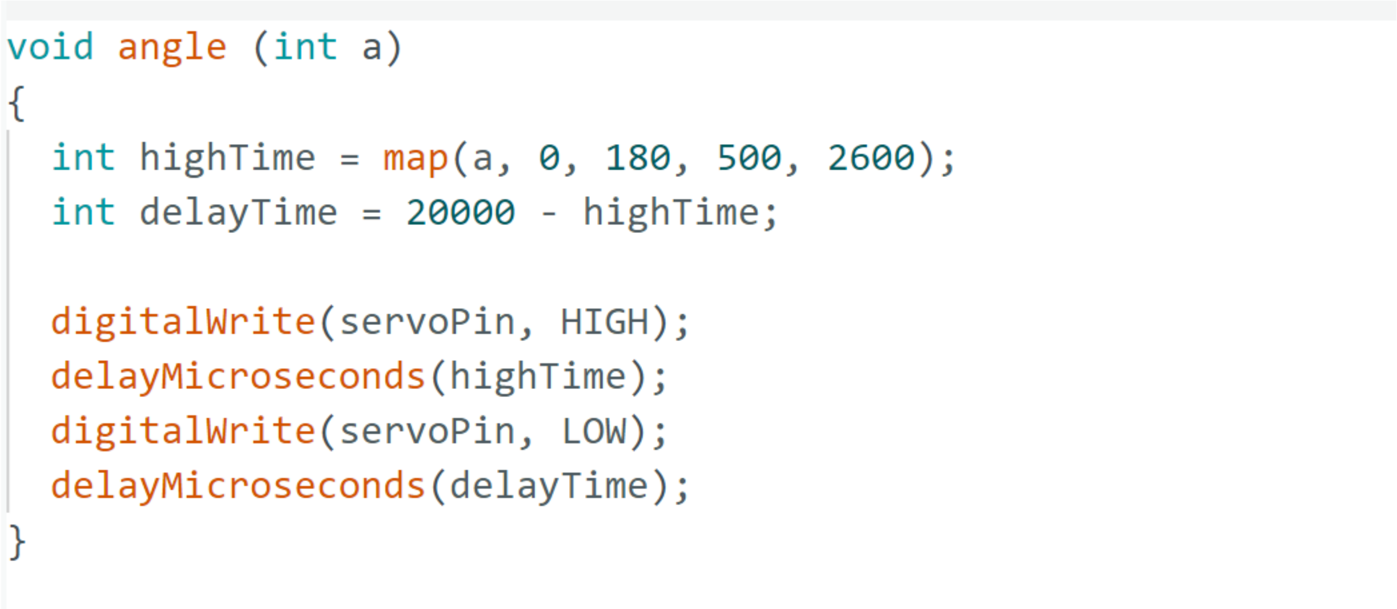
```
int pos = 0;
int servoPin = 13;


void setup() {
  pinMode(servoPin, OUTPUT);

}

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

void angle (int a)
{

}
```

Supporting functions you might need:

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

delayMicroseconds(x);



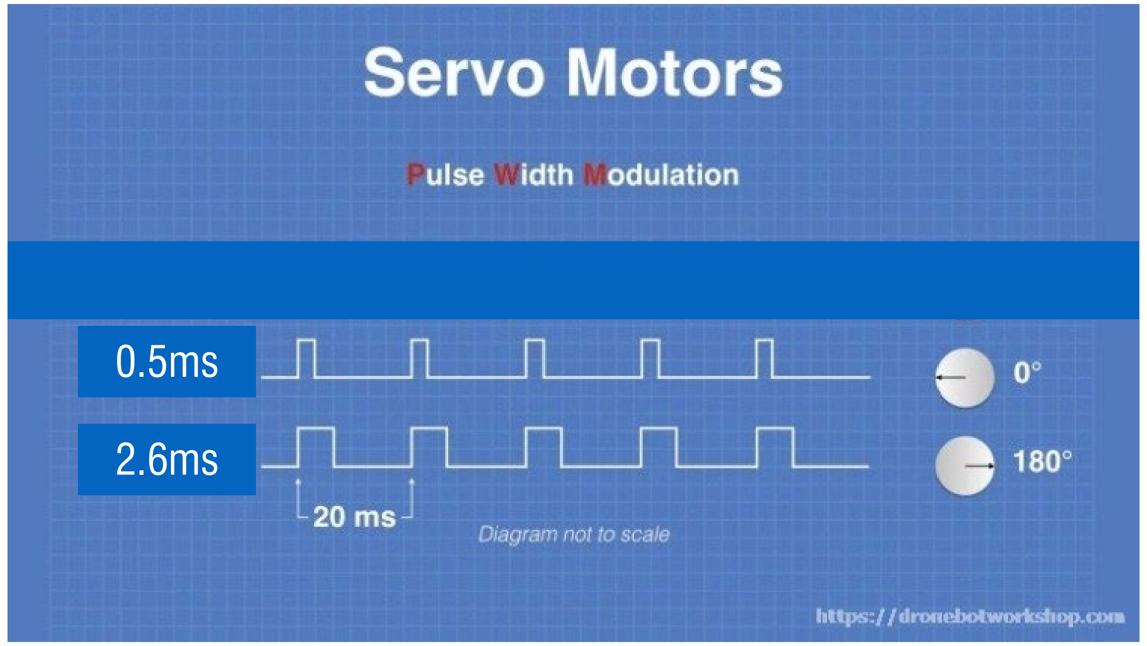


```
void angle (int a)
{
  int highTime = map(a, 0, 180, 500, 2600);
  int delayTime = 20000 - highTime;

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

Supporting functions you might need:

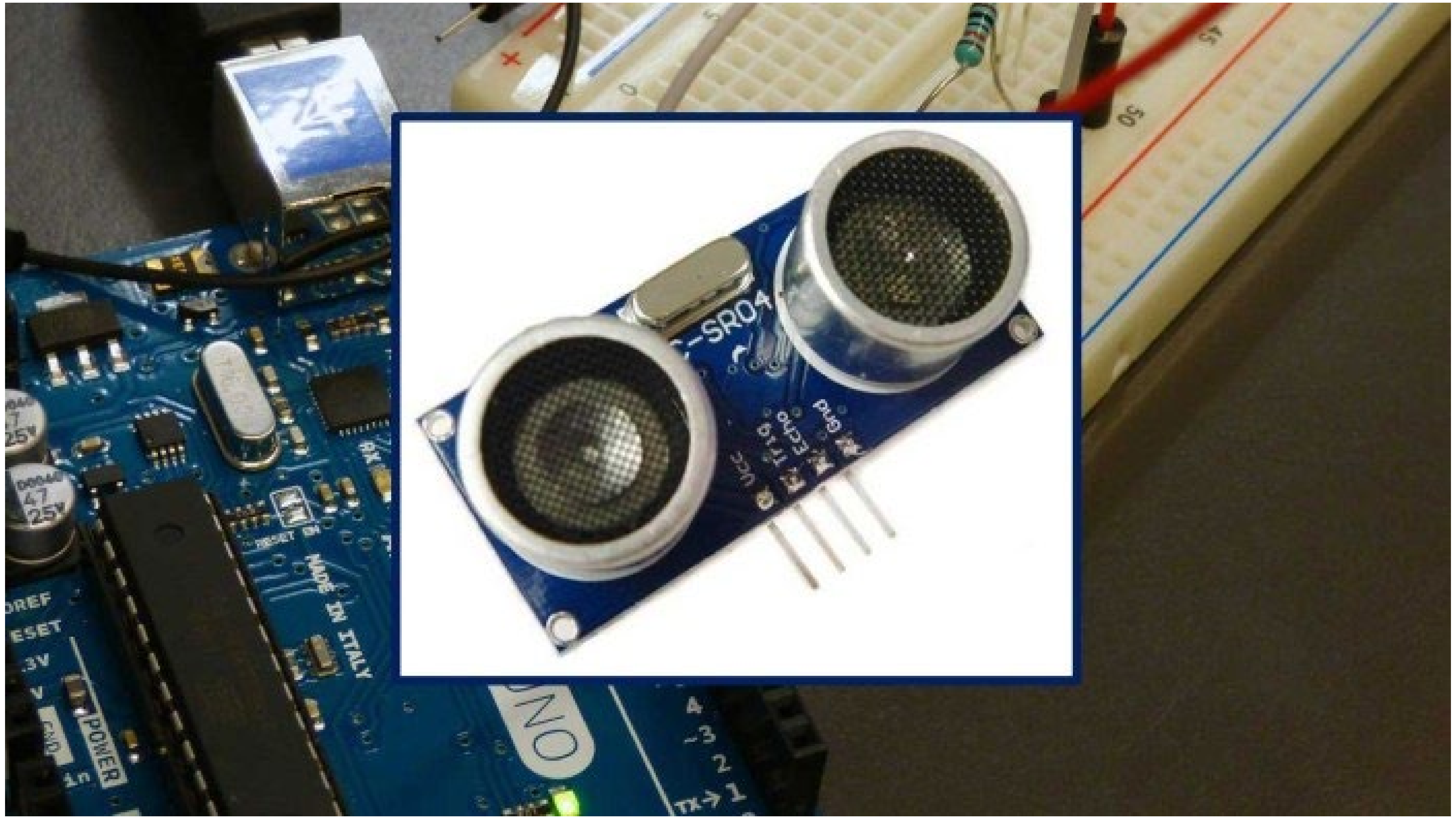
Map(x, fromA, toA, fromB, toB);

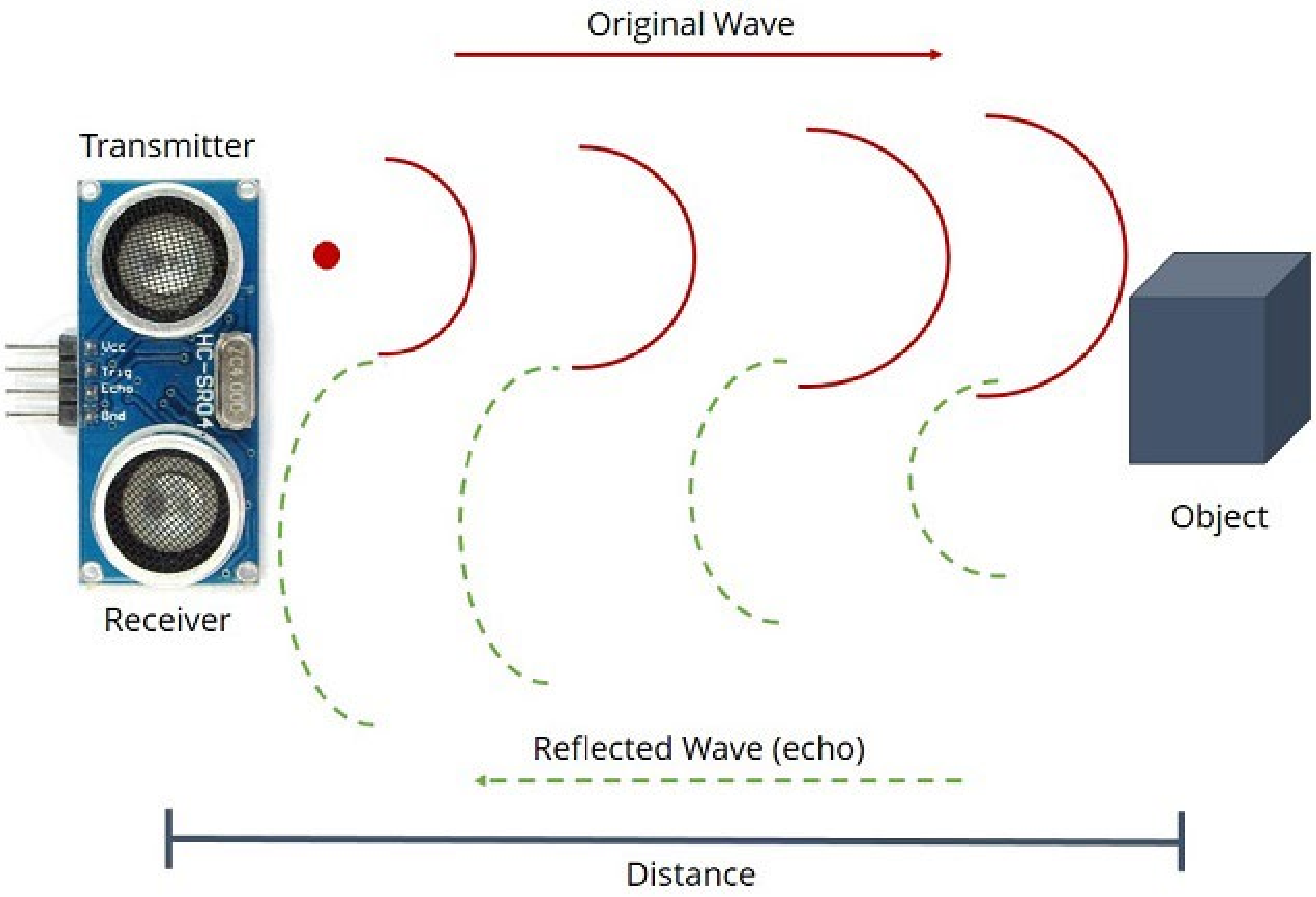
delayMicroseconds(x);



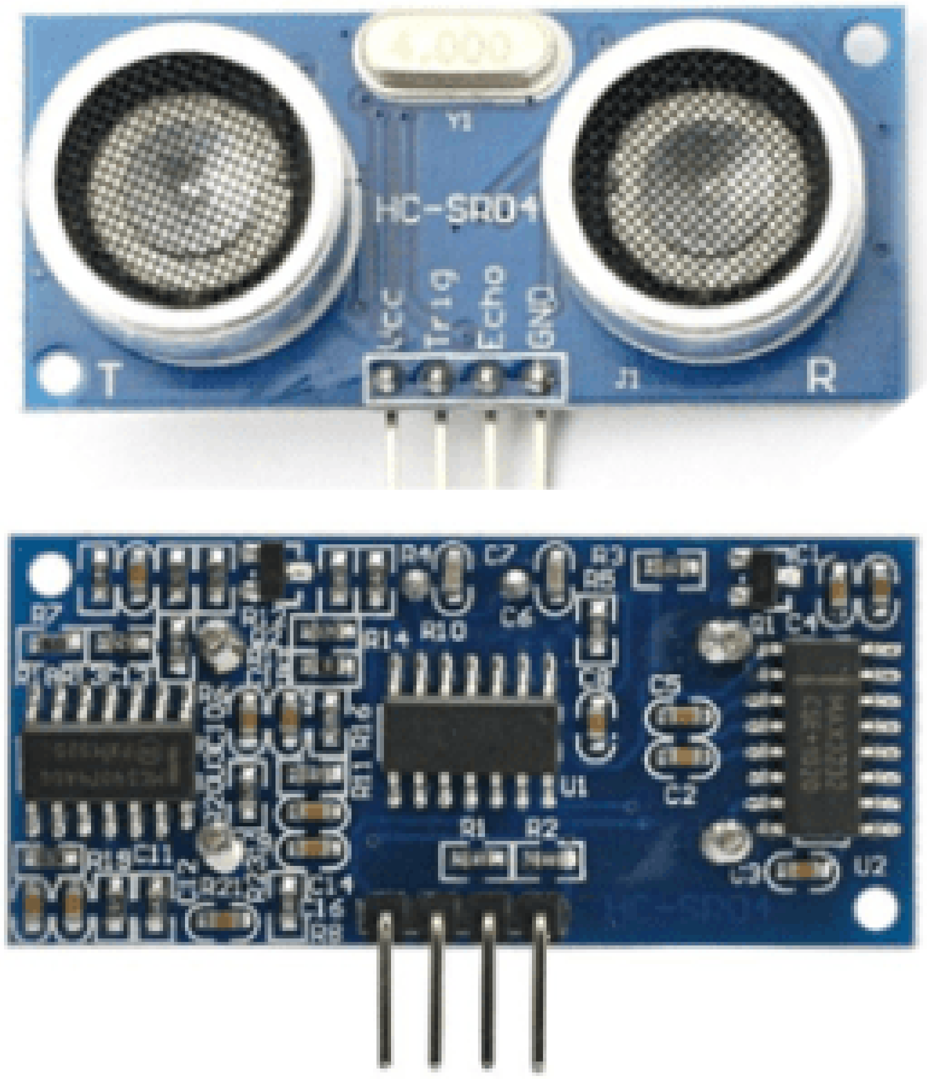
https://dronebotworkshop.com

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

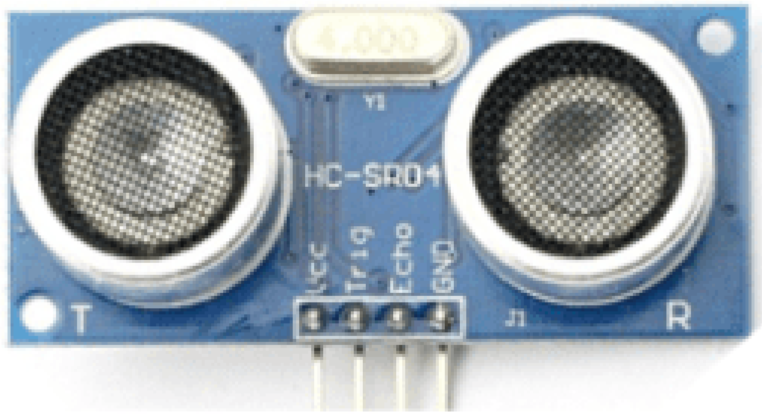
Arduino has library for it, but
Can you do it without library?

Hint:

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

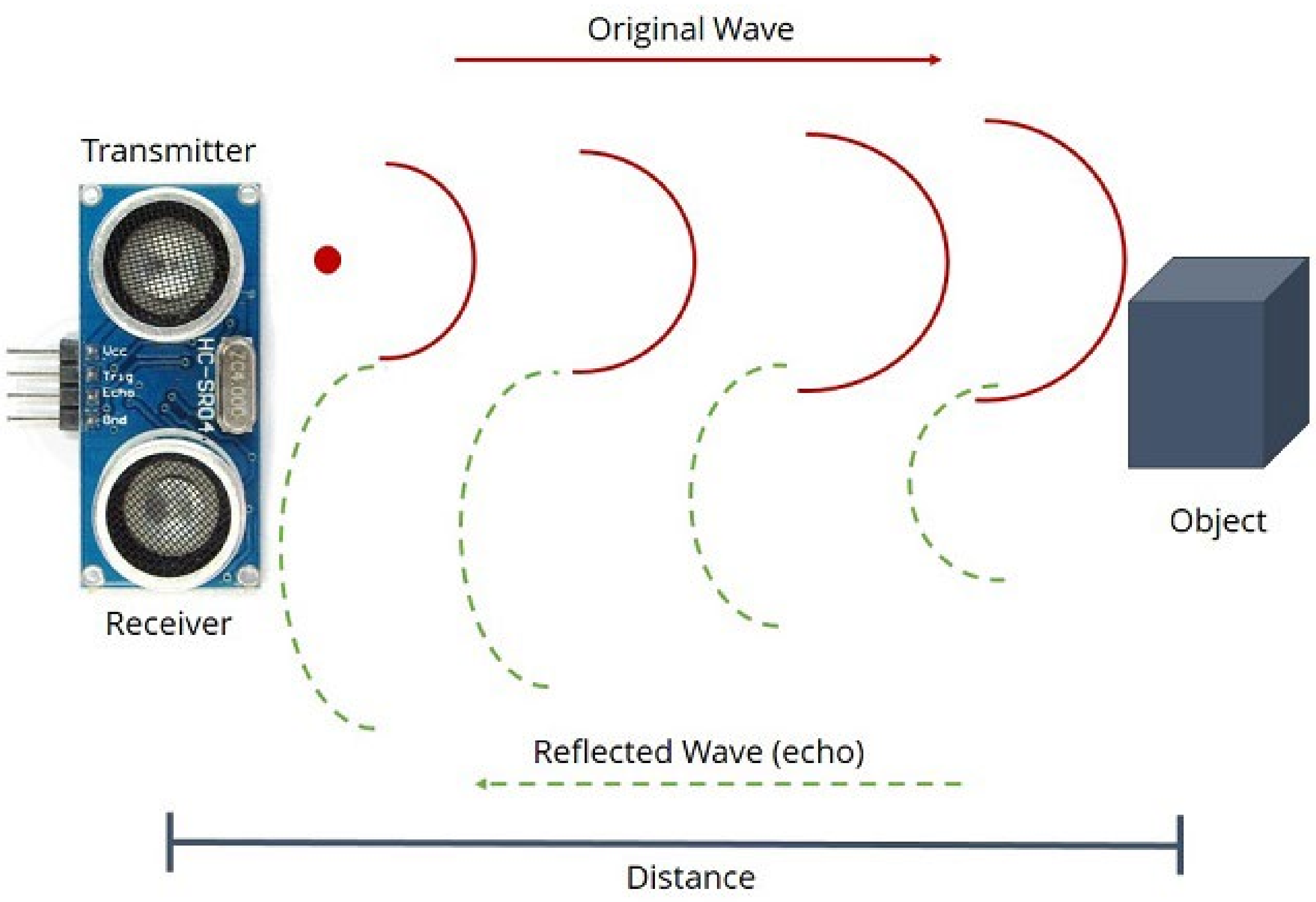
b. To read the sensing signal: a HIGH pulse whose duration is the time (in microseconds) from the sending of the ping to the reception of its echo off of an object.

Google: pulseIn()



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

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