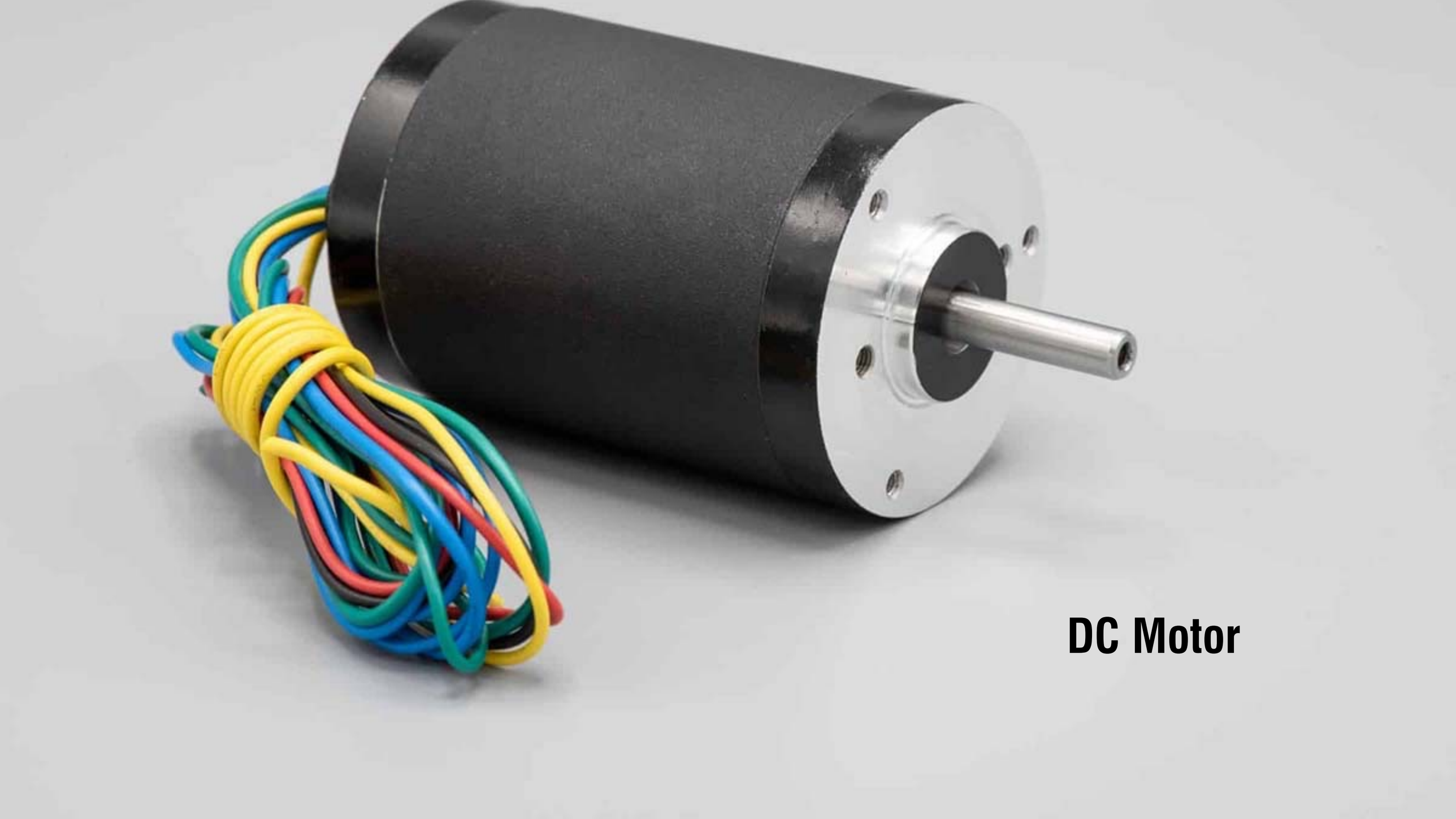
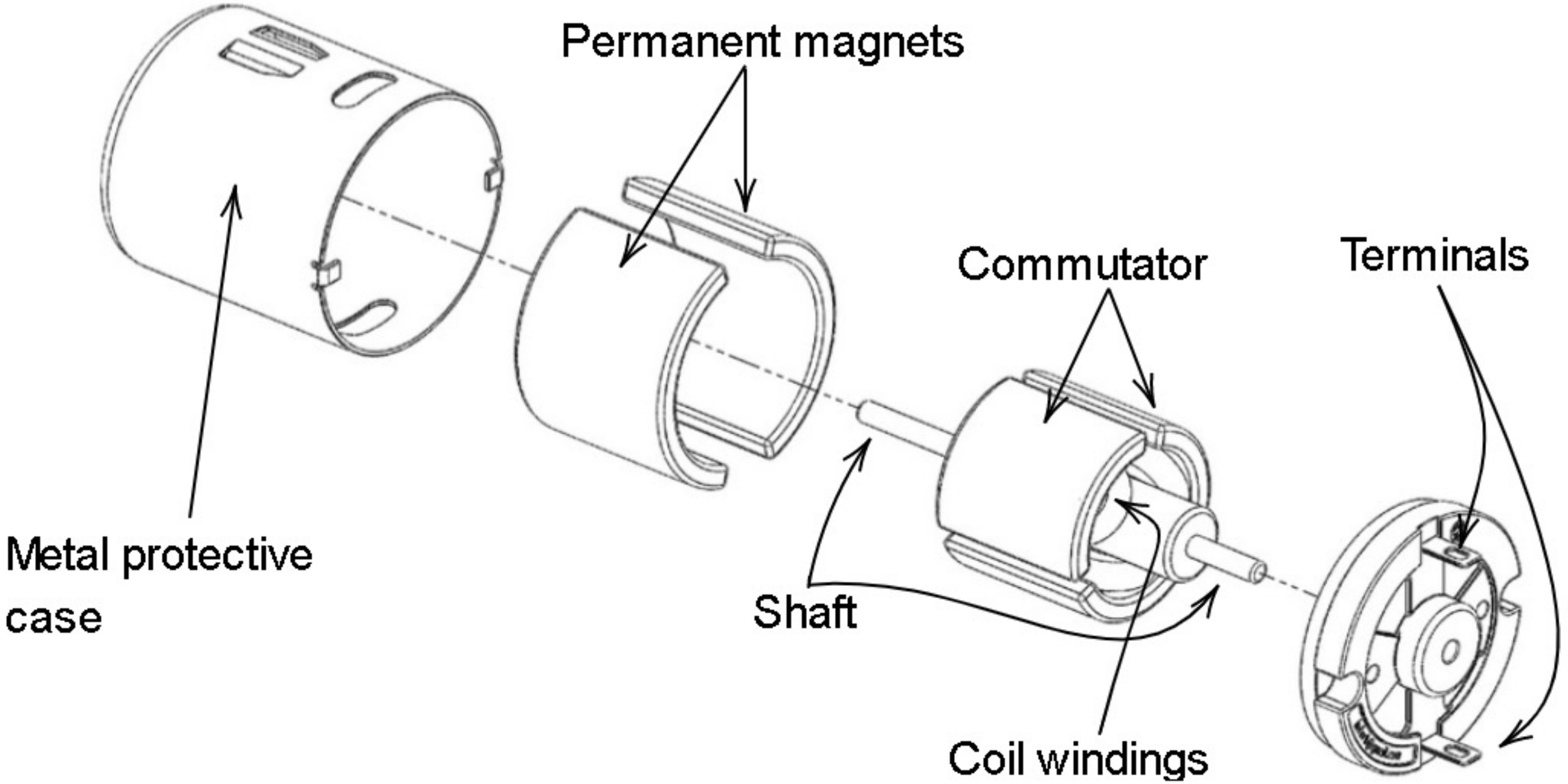
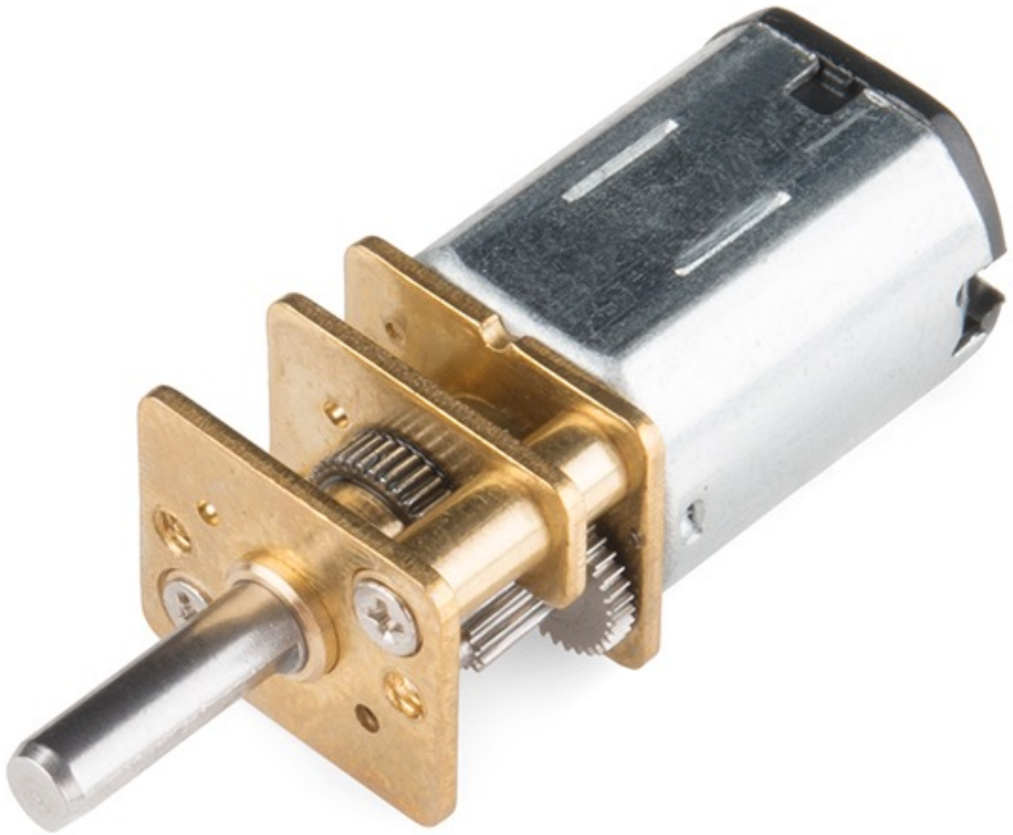
Servo + Stepper

Huaishu Peng | UMD CS | Spring 2026

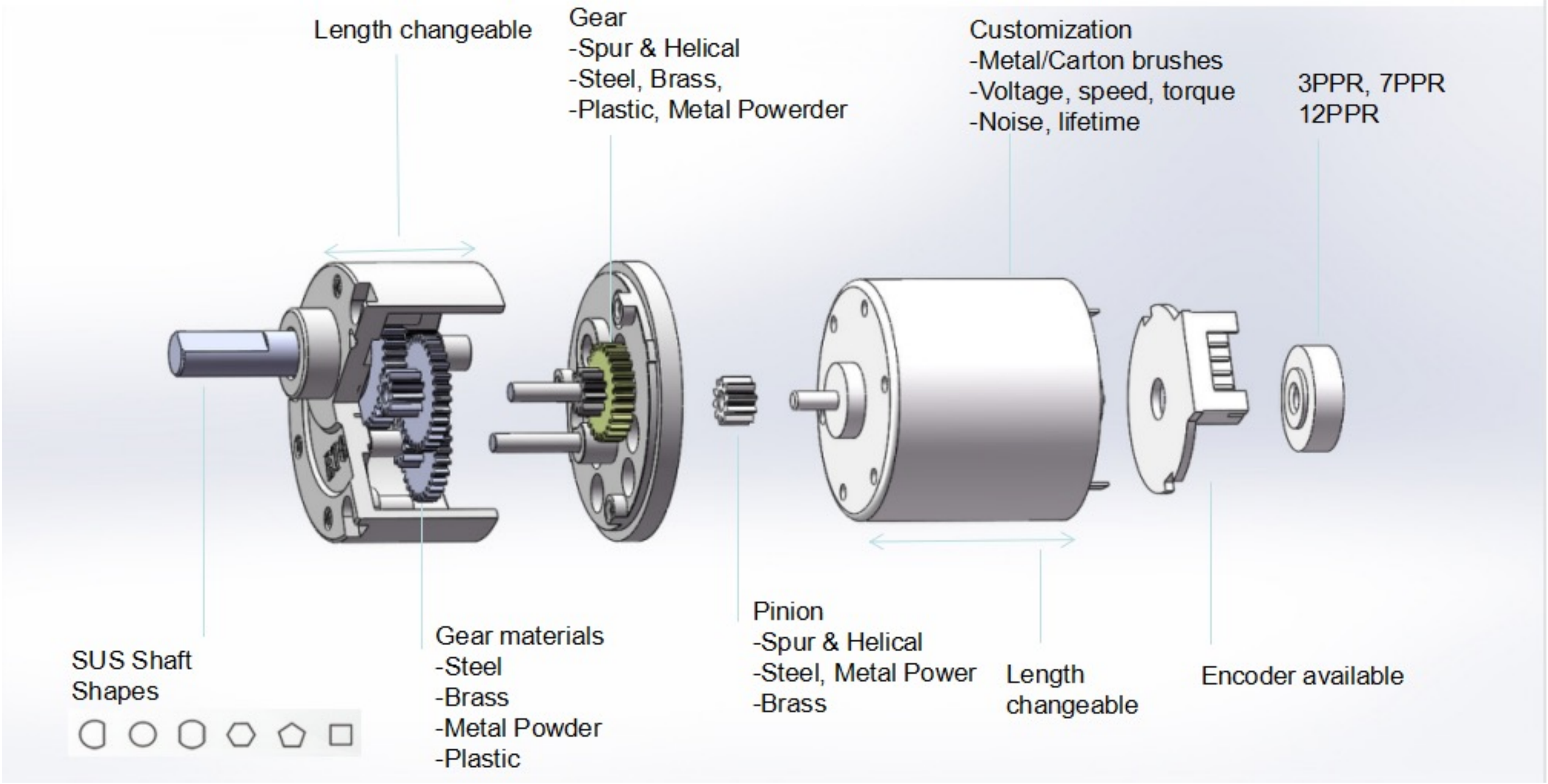


DC Motor


Permanent magnets
Commutator
Terminals
Metal protective case
Shaft
Coil windings



Gearmotor



https://www.etonm.com/About/5408770.html

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
Channel Creative
SELECT
MODE
START
Channel Creative
Youtube.com/ChannelCreativeVN
Facebook.com/ChannelCreative

What is a servo motor?

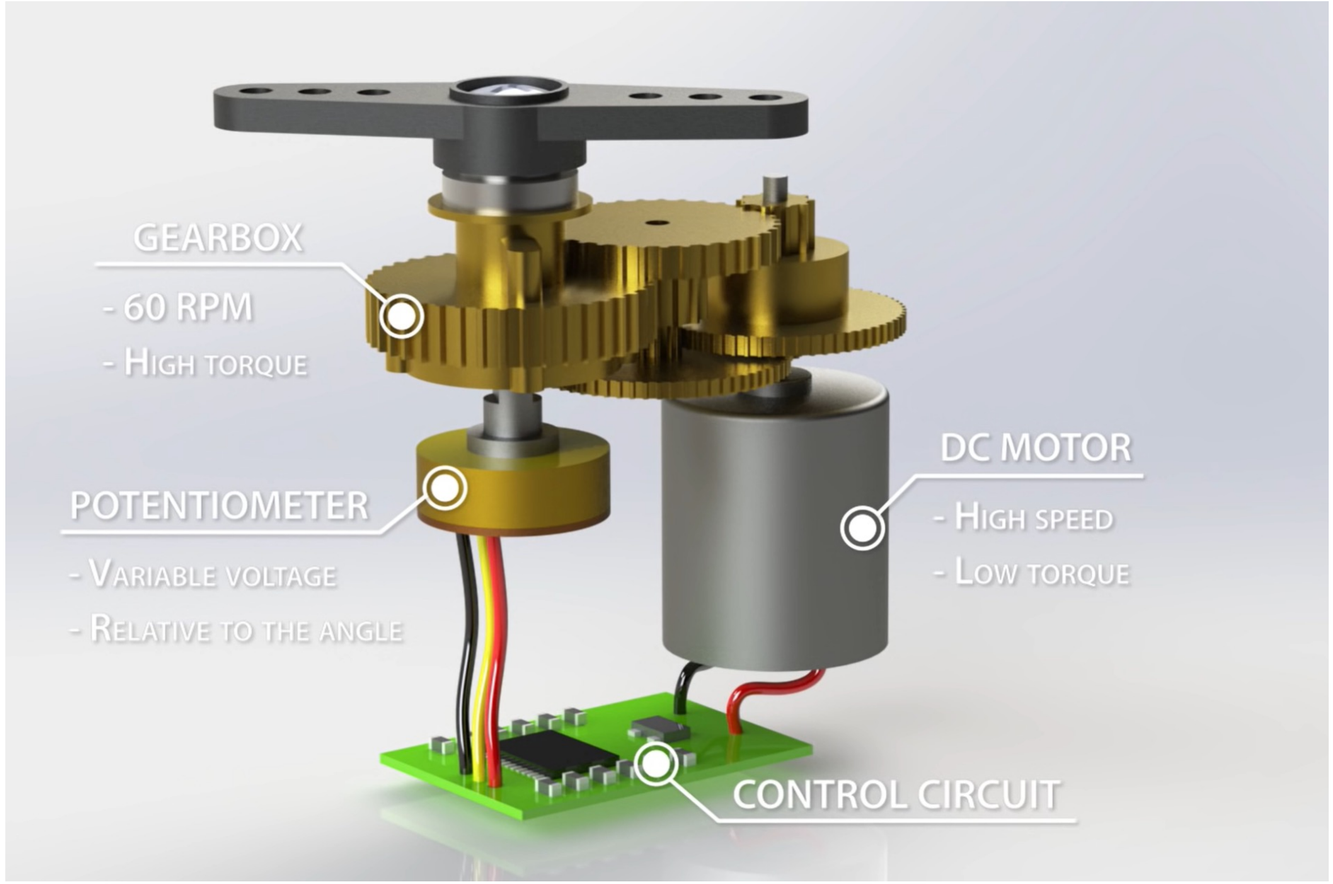
a motor with a built-in “servomechanism”.

Consist of:

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

A feedback device

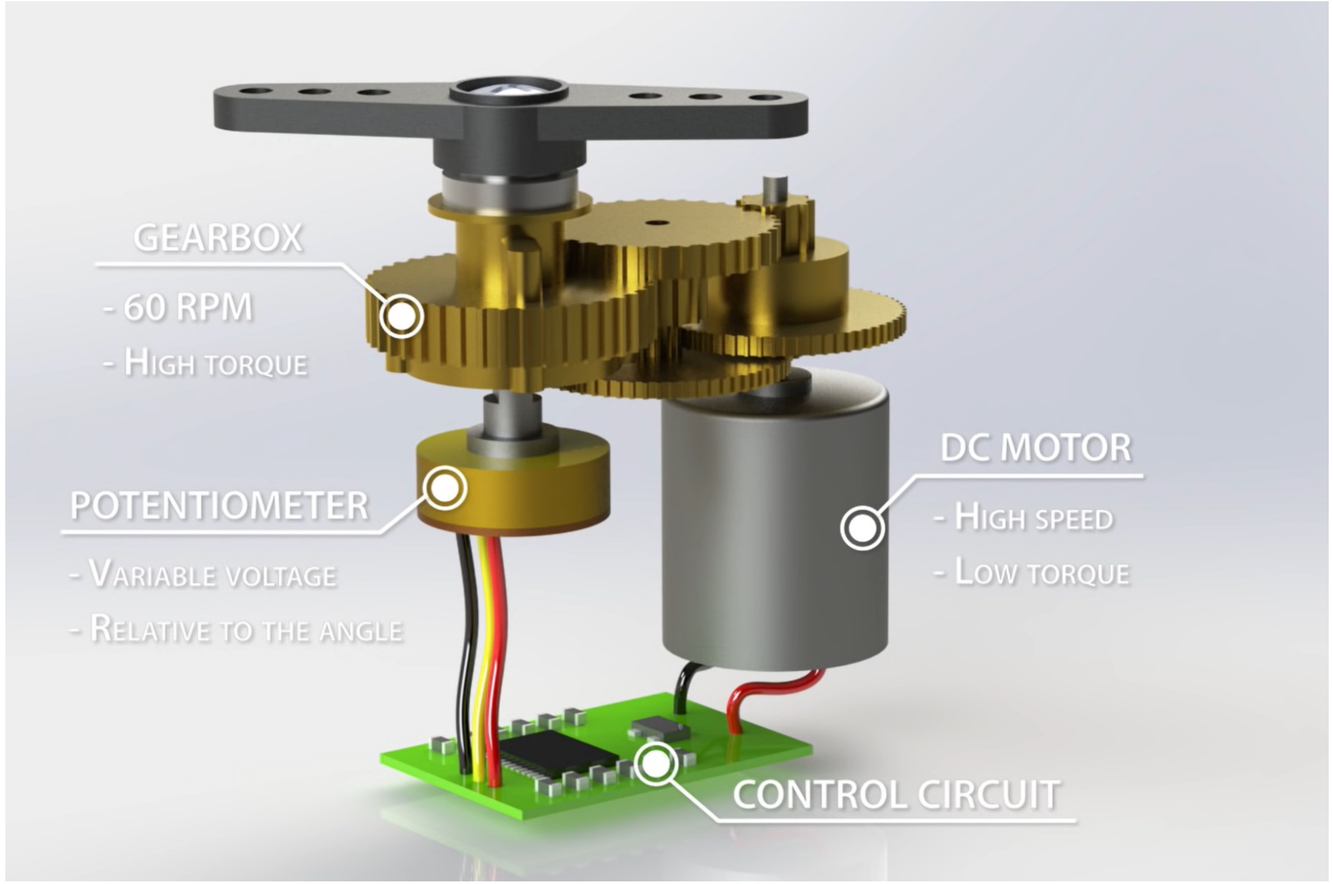
An electronic controller



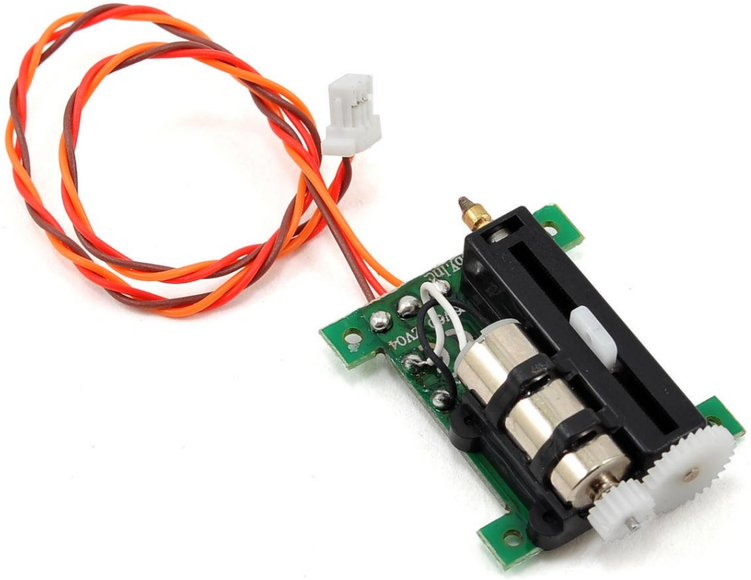
What is a servo motor?

Motors can be DC, or AC

Feedback device can be encoder or other sensors for position sensing




Micro Servo
9g
SG90



Servo Motor

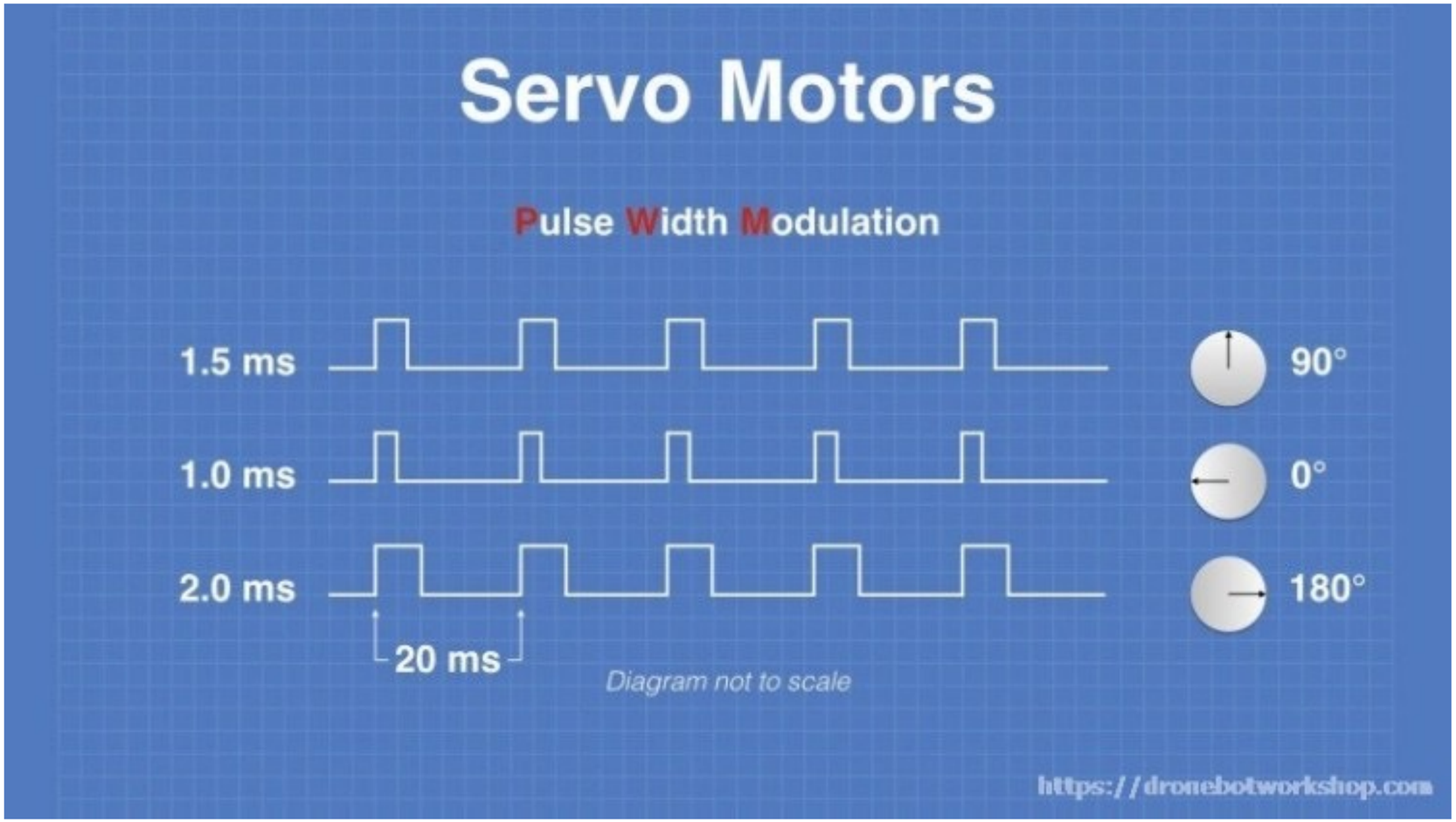
What we have today is a RC servo motor (because it is designed for RC car at the beginning)

Rotational range: 0 ~ 180 degree

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

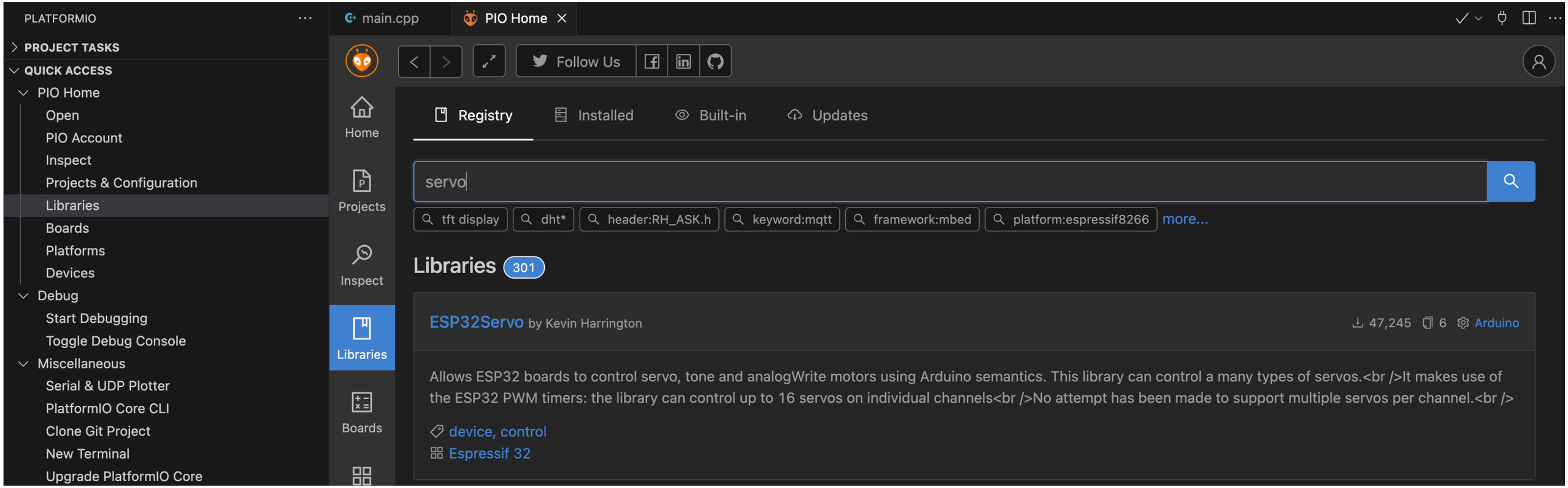
Operating Voltage: **5**V

Controlling the position of a servo motor



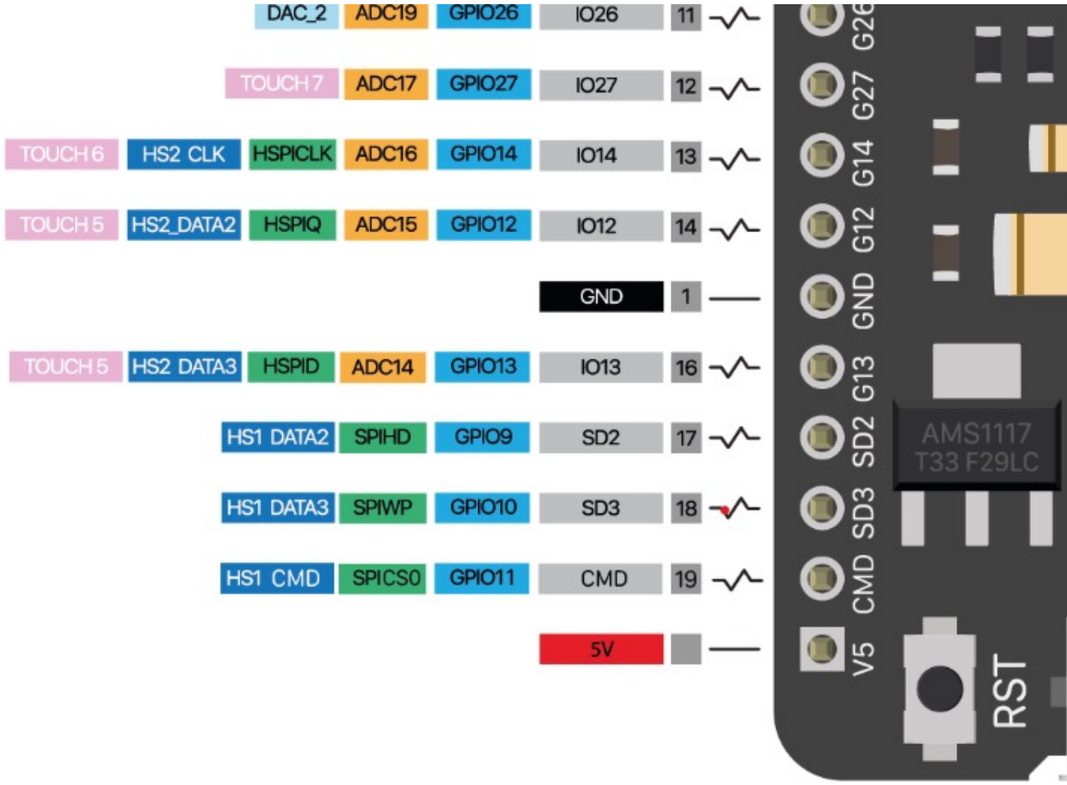
Using a Servo lib

1. Install the ESP32servo lib from the library manager



Using a Servo lib

2. Wiring

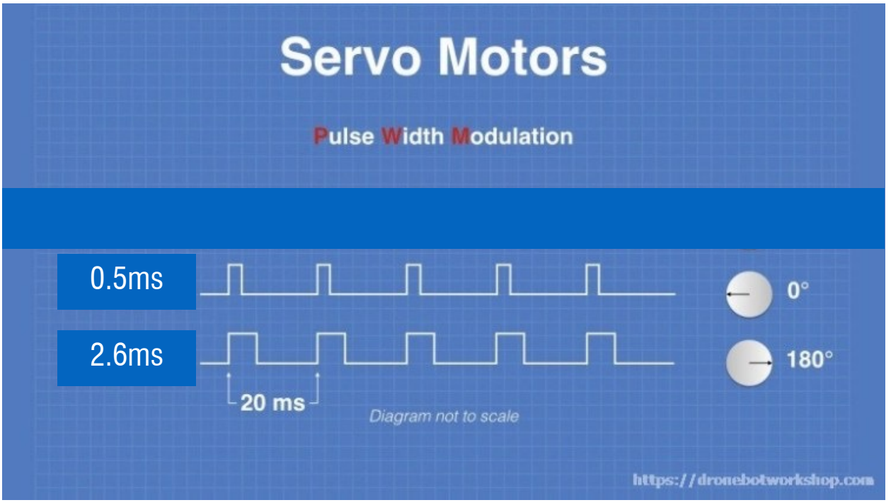


Wire connection: Power (Red) | GND (Brown) | Signal (Orange)

5V GND GPIO 13

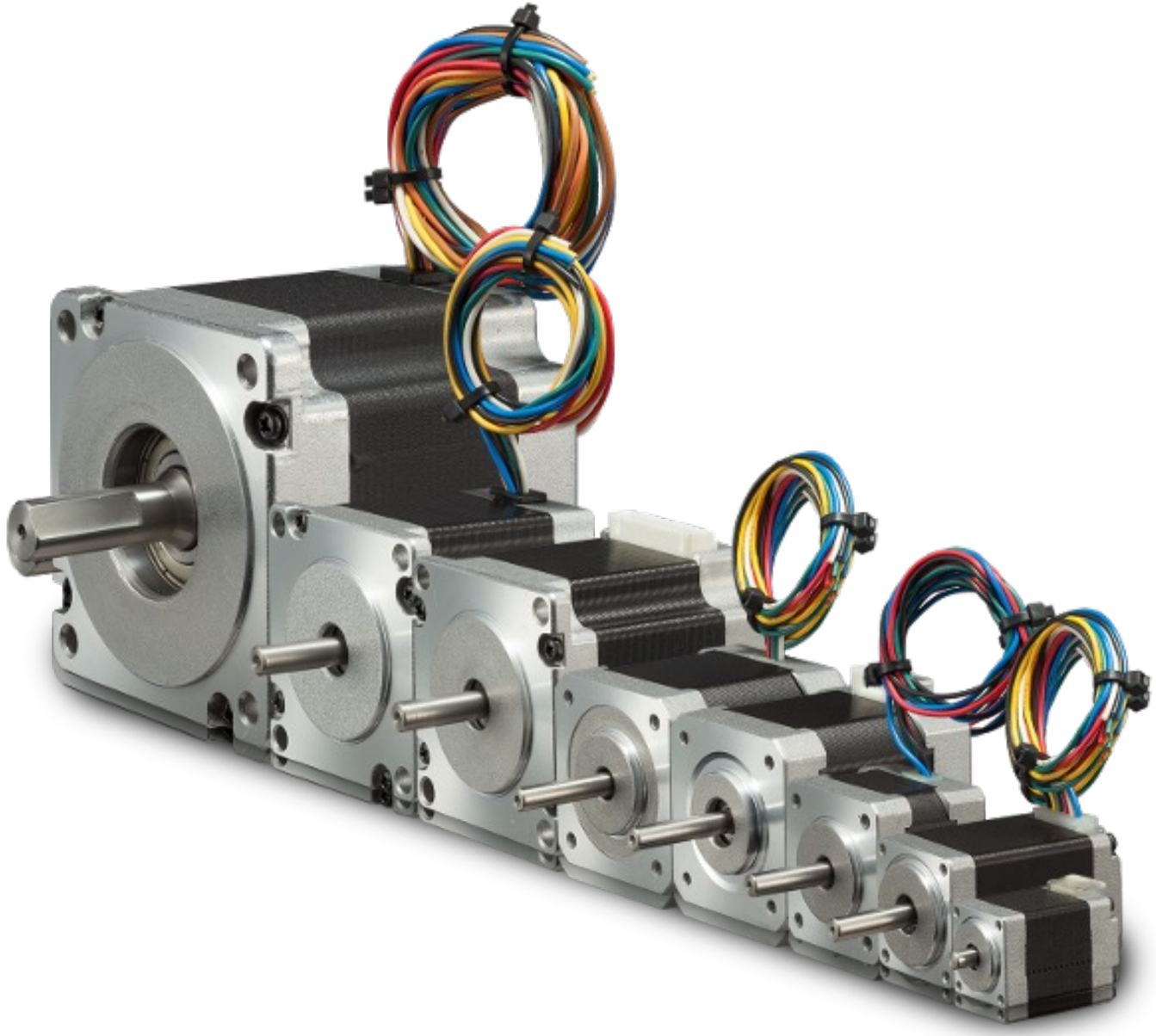
Using a Servo lib

3. Coding




```ino
#include <Arduino.h>
#include <ESP32Servo.h>

int pos = 0;
int servoPin = 13;
Servo myservo;

void setup() {
  myservo.attach(servoPin, 500, 2600);
}

void loop() {
  for (pos = 0; pos<180; pos+=1) {
    myservo.write(pos);
    delay(15);
  }

  delay(500);

    for (pos = 180; pos>0; pos-=1) {
    myservo.write(pos);
    delay(15);
  }

  delay(500);
}
```

Stepper Motor

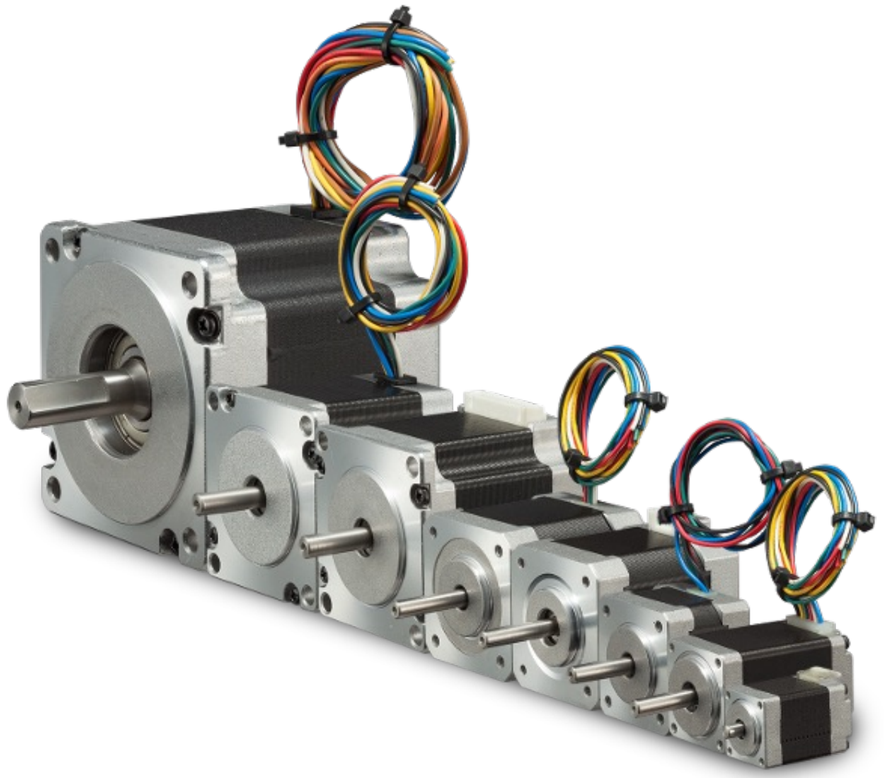


Special type of motor moving **by steps**

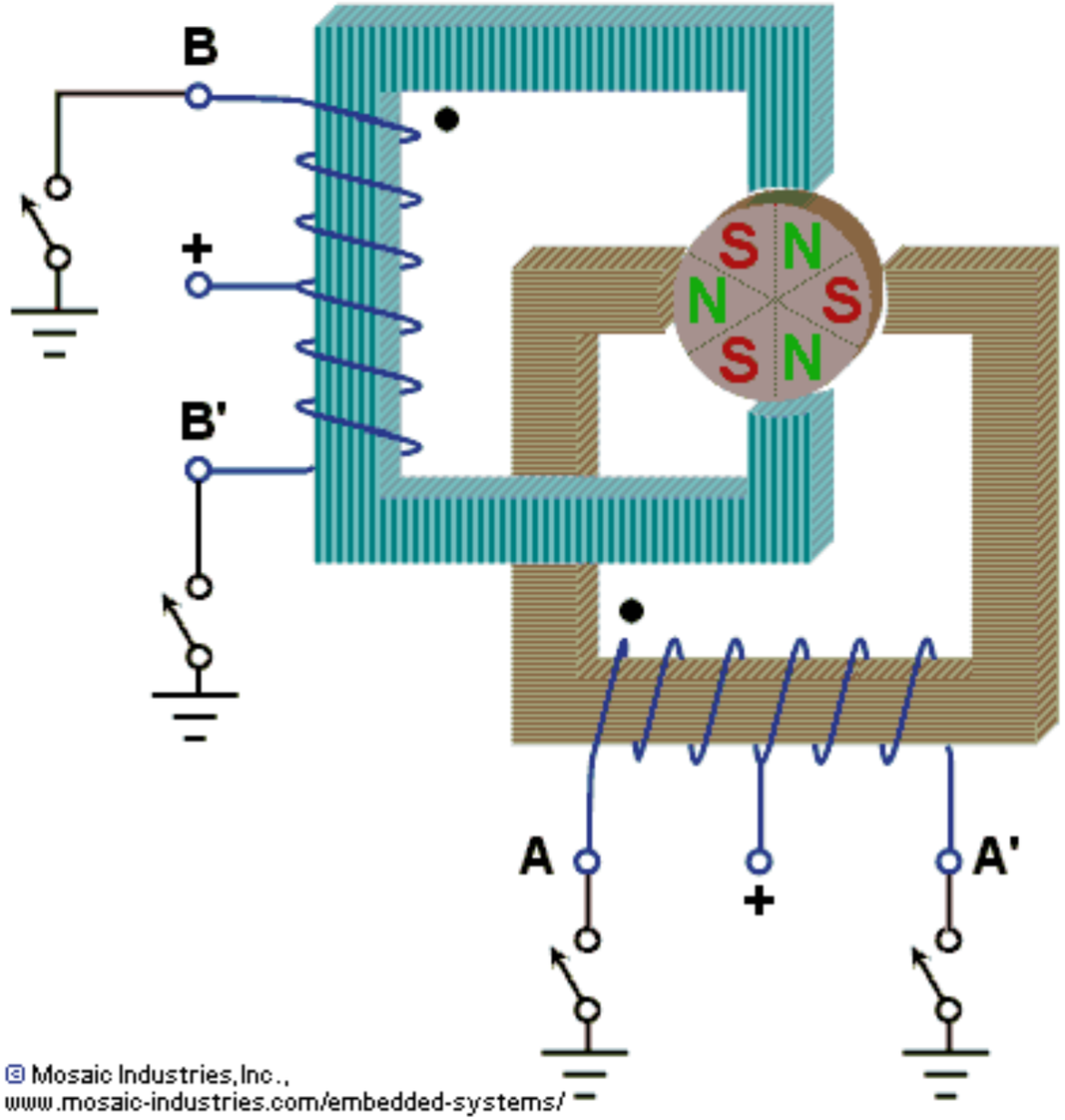
– Useful for ***precise*** motion
    Printers, scanners, robots…

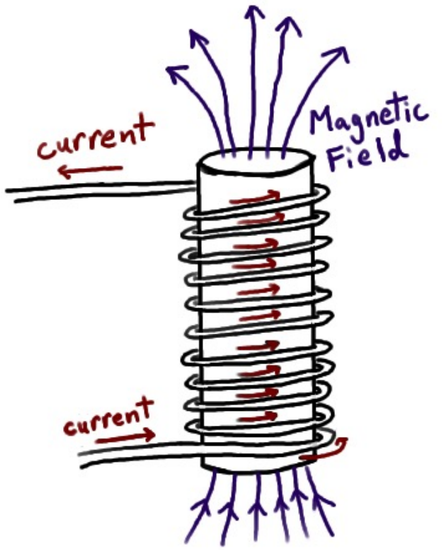
– High holding torque
    But can slip

– Two kinds
    Bipolar & Unipolar



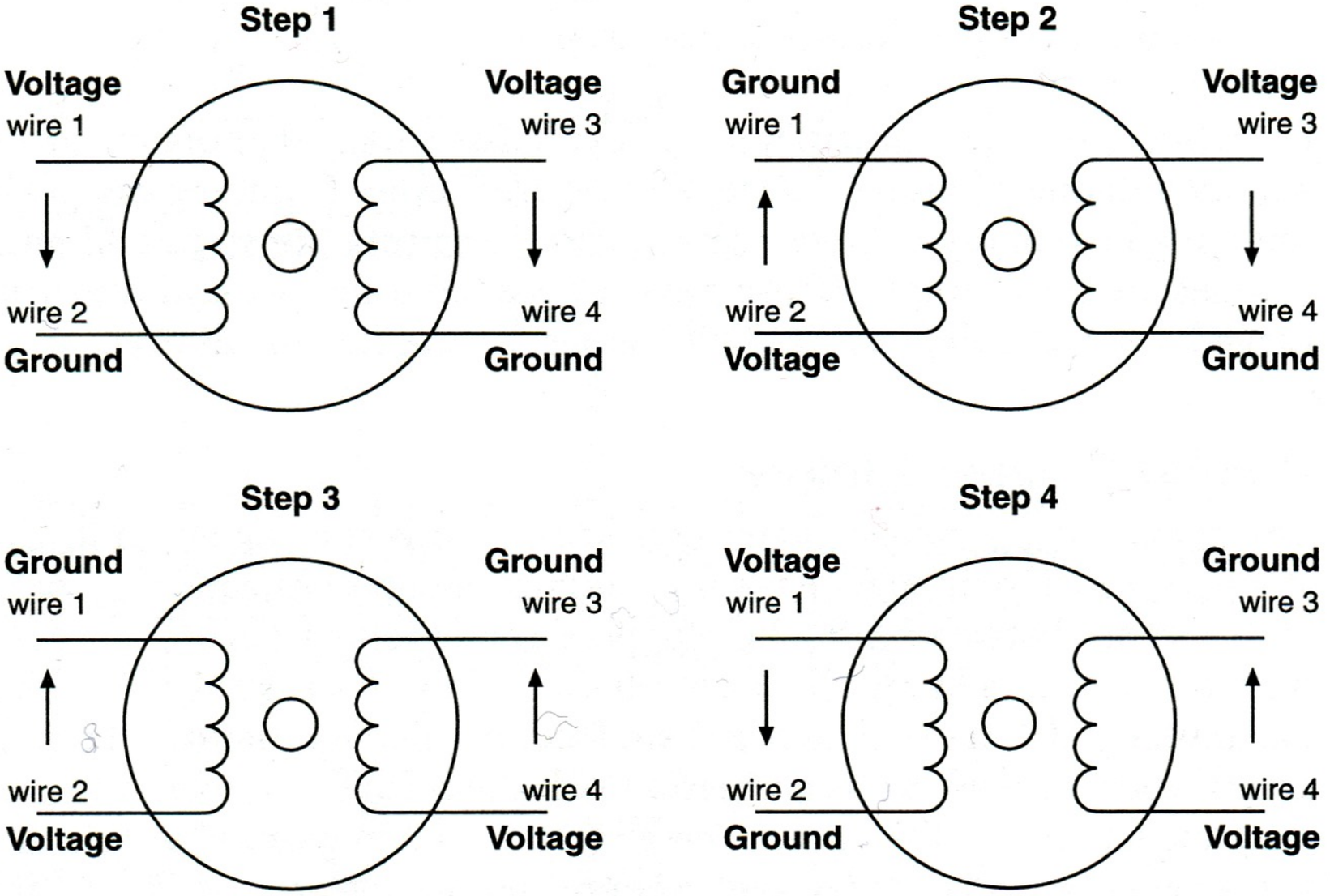
Unipolar



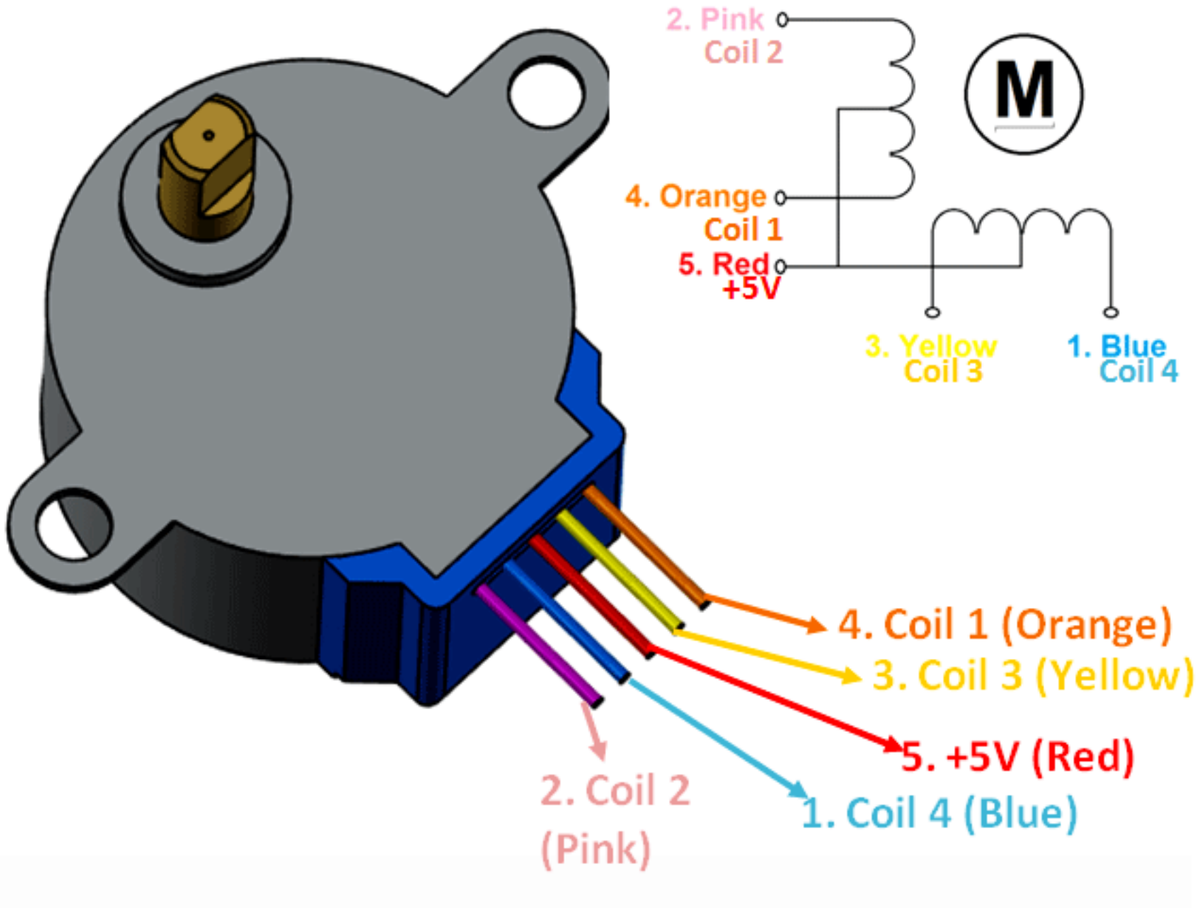


Bipolar

2 coils
4 wires



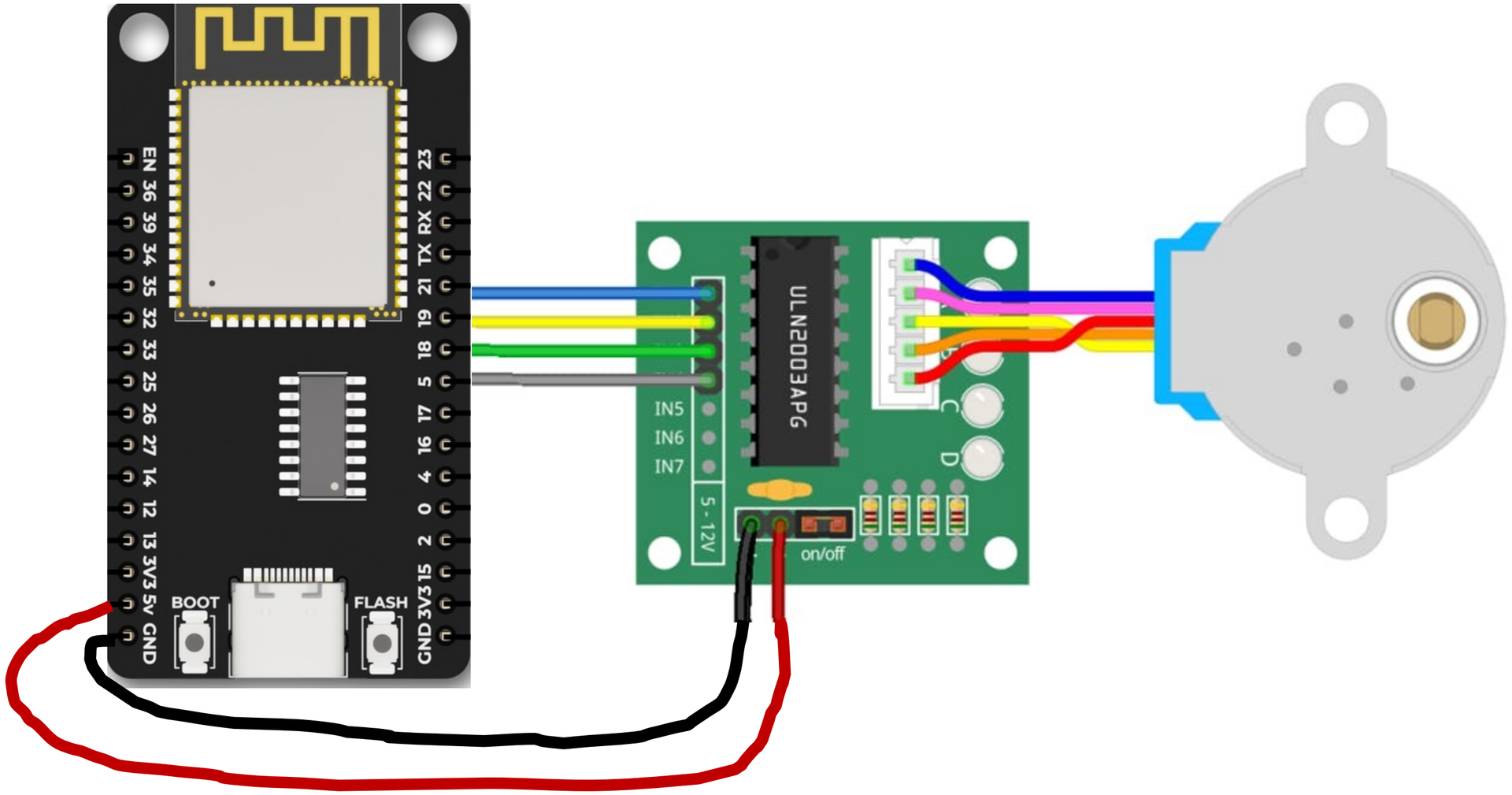
That’s what we use for today’s class



Do not modify circuit while ESP32 is on!

Why we need a separate driver?

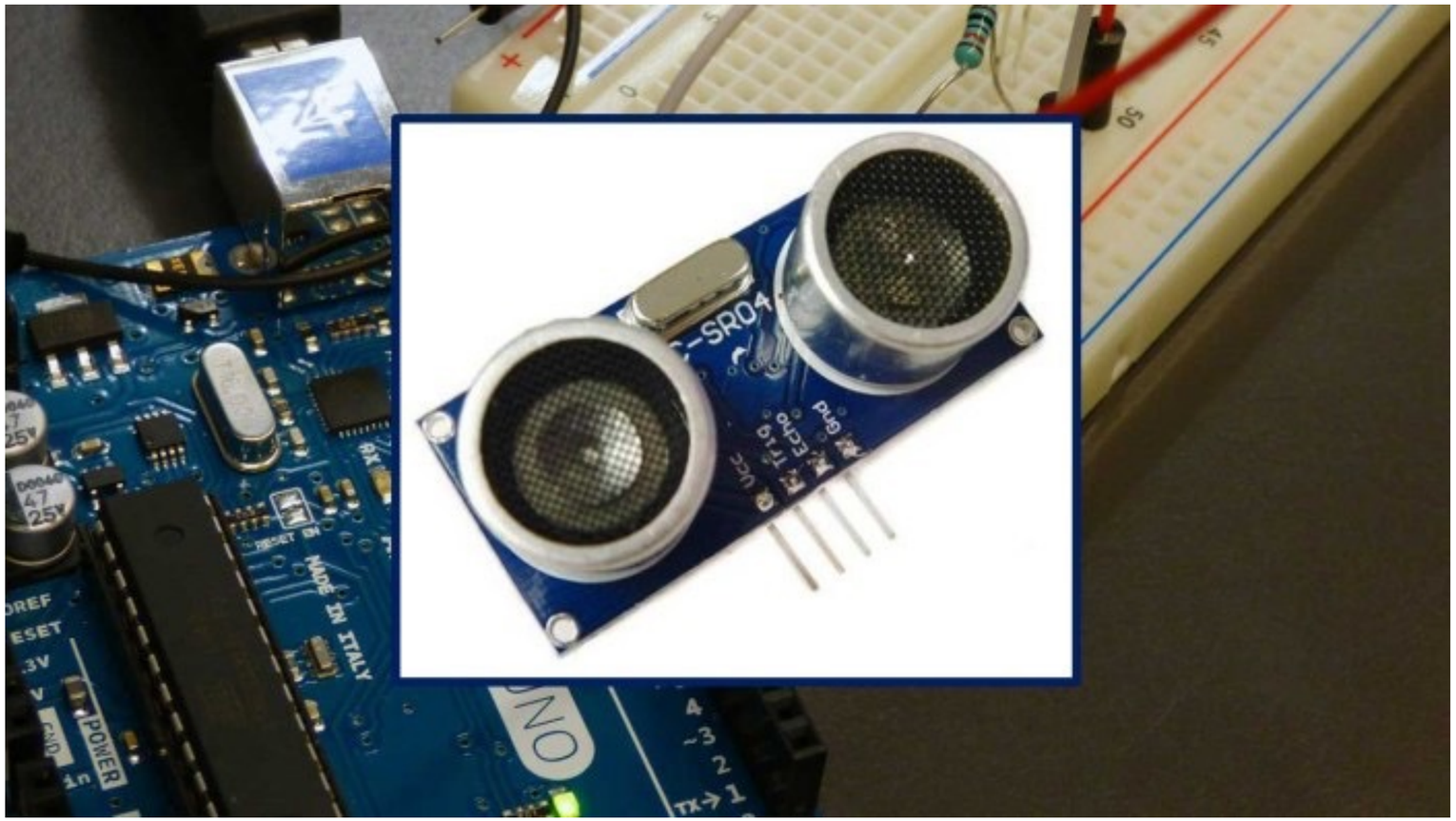
driver's chip helps us manage the change of the current direction

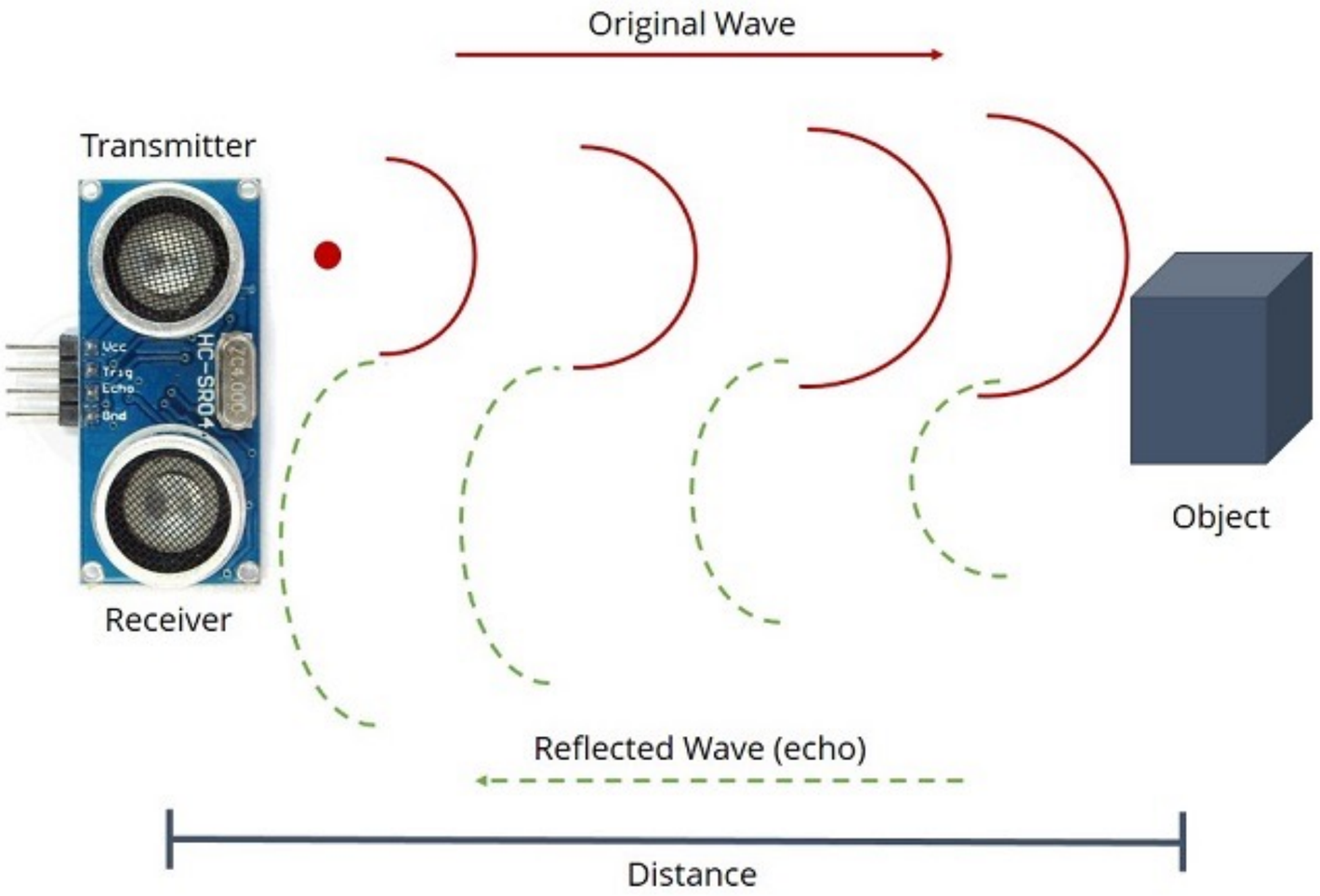


IN1->GPIO21
IN2-> GPIO19
IN3-> GPIO18
IN4-> GPIO5
5V -> VIN
GND -> GND

```
#include <Arduino.h>
#include <Stepper.h>



const int stepsPerRevolution = 2048; // change this to fit the number
of steps per revolution


// ULN2003 Motor Driver Pins
#define IN1 21
#define IN2 19
#define IN3 18
#define IN4 5


// initialize the stepper library
Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4);


void setup() {


myStepper.setSpeed(5);
}


void loop() {
 // step one revolution in one direction:
 Serial.println("clockwise");
 myStepper.step(stepsPerRevolution);
 delay(1000);


 // step one revolution in the other direction:
 Serial.println("counterclockwise");
 myStepper.step(-stepsPerRevolution);
 delay(1000);



delay(1000); // wait for a second before the next loop
}
```

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

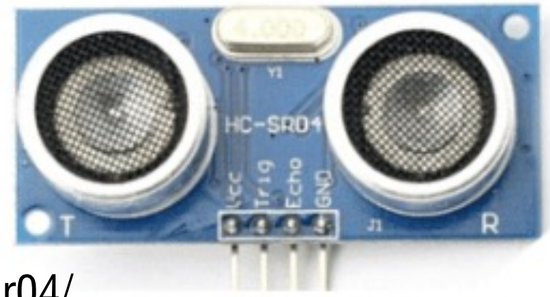
VCC: +5VDC
Trig : Trigger (**OUTPUT**) -> GPIO5
Echo: Echo (**INPUT**) -> GPIO18
GND: GND

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

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.

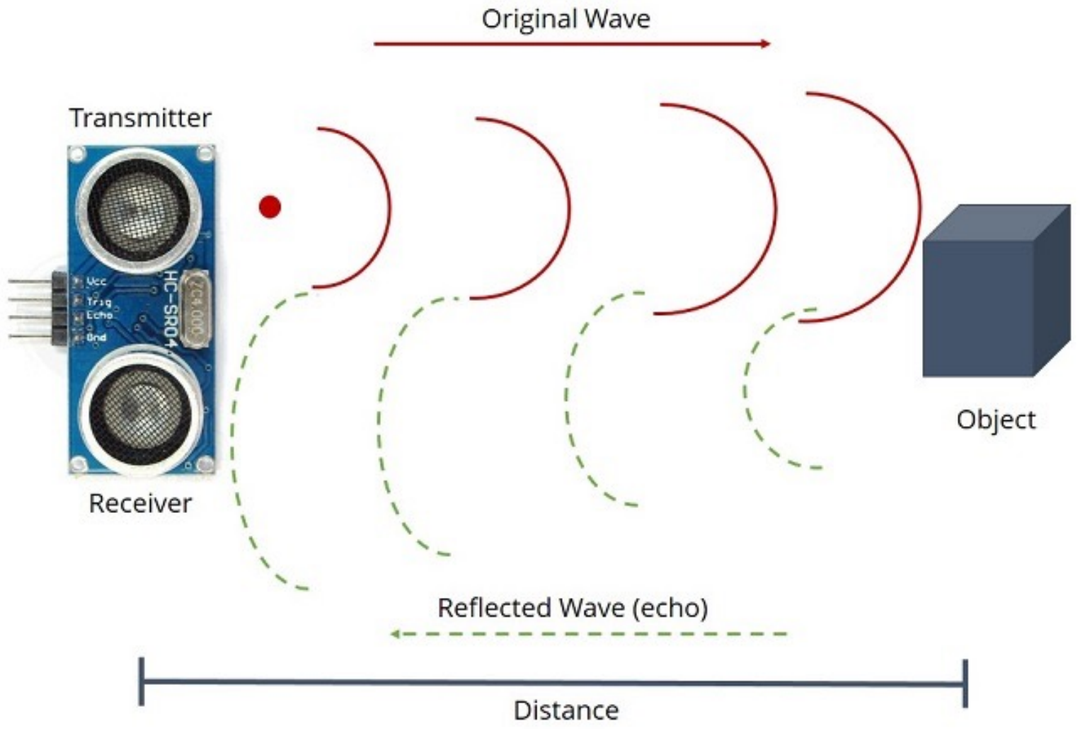
The pulseIn() function in Arduino is used to measure the duration of a pulse on a specified pin. It listens for either a HIGH or LOW pulse on that pin and returns the time in microseconds that the pulse lasted.



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




```
void distance() {
  long duration=0, distance = -1;
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);


  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, 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");
  }

}
```

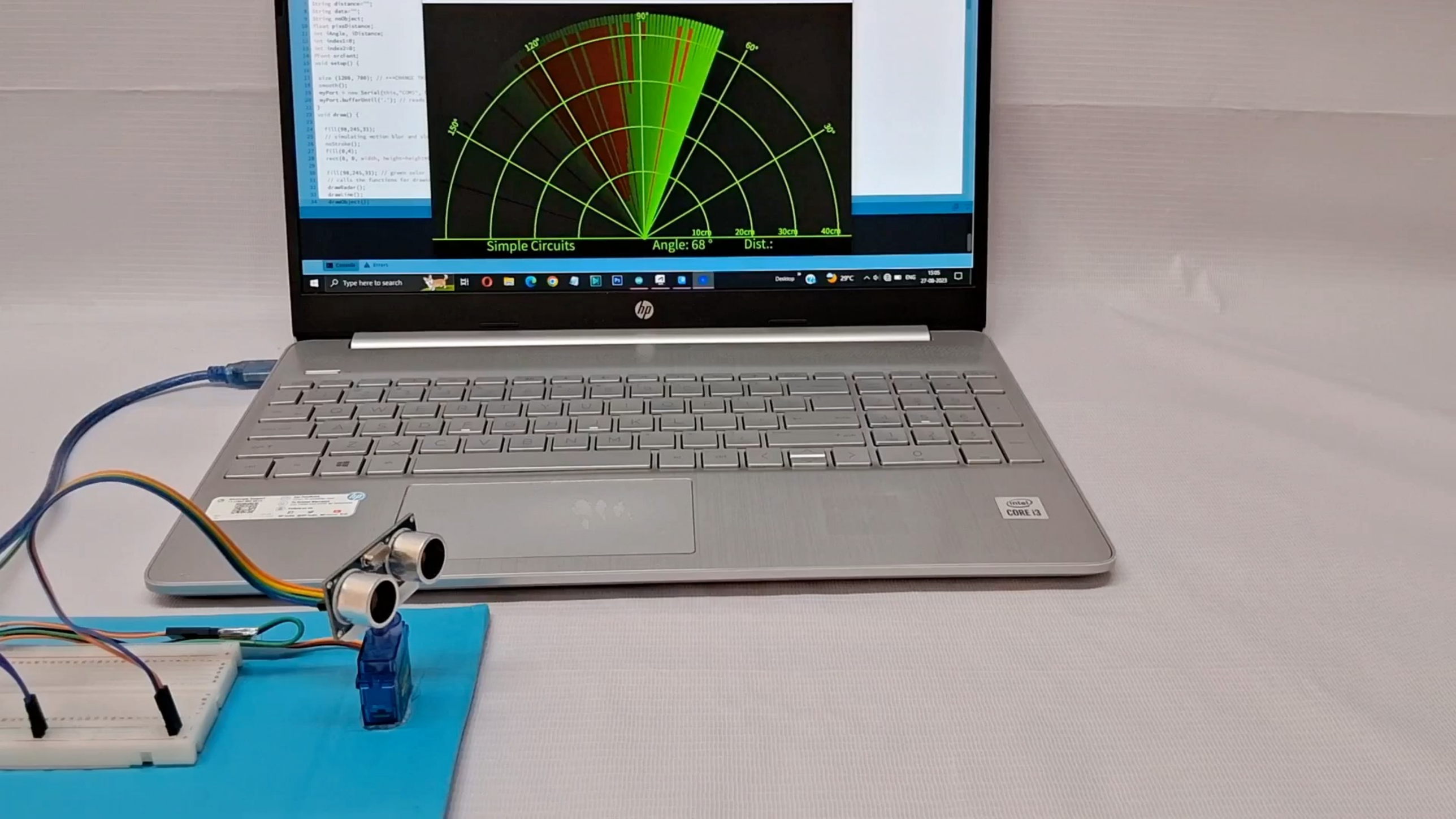
Assignment:

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

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.

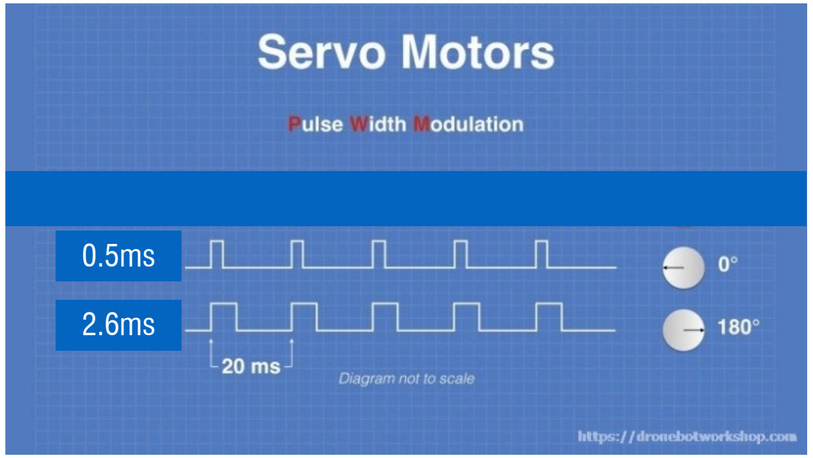
Optional: write a GUI.


90°
120°
60°
150°
30°
10cm
20cm
30cm
40cm
Simple Circuits
Angle: 68 °
Dist.:
Type here to search
CORE i3

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);

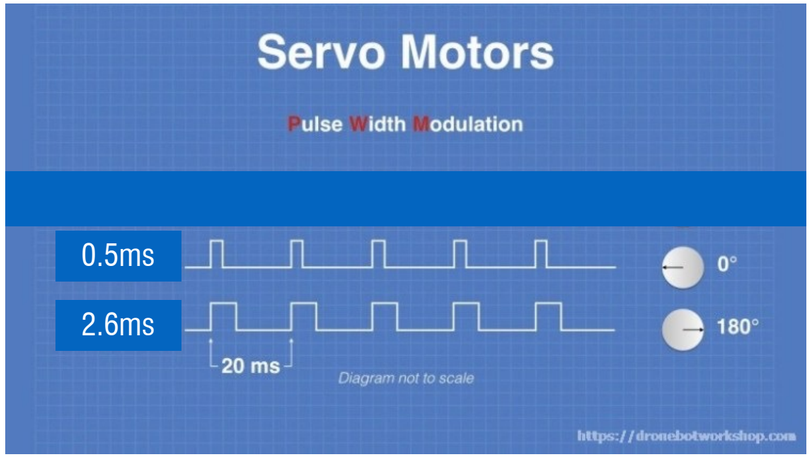
```
#include <Arduino.h>
#include <ESP32Servo.h>

int pos = 0;
int servoPin = 13;
Servo myservo;

void setup() {
  myservo.attach(servoPin, 500, 2600);
}

void loop() {
  for (pos = 0; pos<180; pos+=1) {
    myservo.write(pos);
    delay(15);
  }

  delay(500);

    for (pos = 180; pos>0; pos-=1) {
    myservo.write(pos);
    delay(15);
  }

  delay(500);
}
```

```arduino
#include <Arduino.h>

int pos = 0;
int servoPin = 13;

void angle(int a);

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

void loop() {
  for (pos = 0; pos<180; pos+=1) {
    angle(pos);
    delay(15);
  }

  delay(500);

    for (pos = 180; pos>0; pos-=1) {
    angle(pos);
    delay(15);
  }

  delay(500);
}

void angle(int a) {

}
```

Supporting functions you might need:

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

delayMicroseconds(x);



```
void angle(int a) {
  int pulseWidth = map(a, 0, 180, 500, 2600);

  digitalWrite(servoPin, HIGH);
  delayMicroseconds(pulseWidth);
  digitalWrite(servoPin,LOW);
  delayMicroseconds(20000-pulseWidth);
}
```

Supporting functions you might need:

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

delayMicroseconds(x);