




Lesson4 Obstacle Avoidance Car




Points of This Section

*The joy of learning lies not only in learning how to control your car, but also in learning how to protect your car.
So, keep your car far away from collision.*

Learning Objectives:

-  *Learn how to assemble the ultrasonic module*
-  *Be familiar with using steering*
-  *Learn about the principle of obstacle avoidance car*
-  *Use the program to make obstacle avoidance car come true*

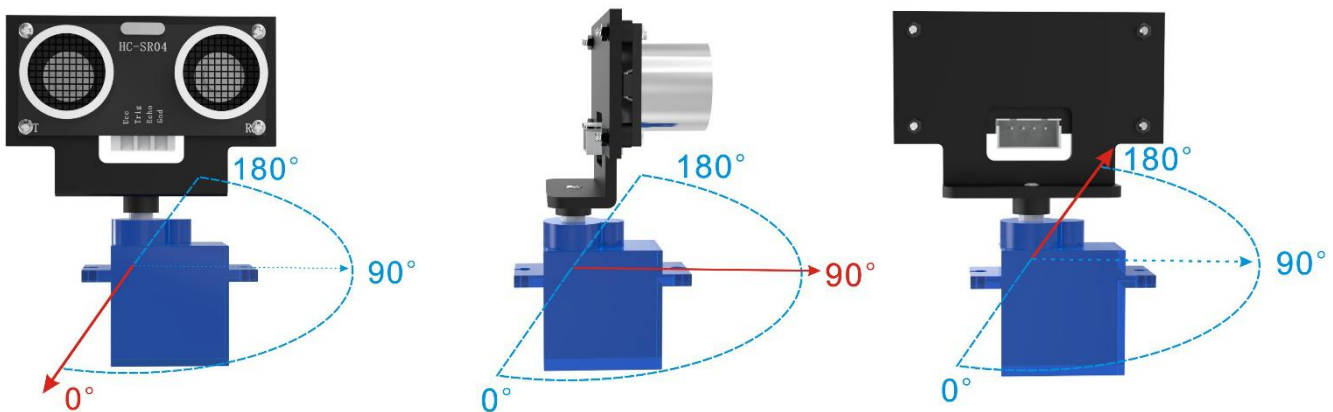
Preparations:

-  *A car (with battery)*
-  *A USB cable*
-  *A suit of ultrasonic cradle head*

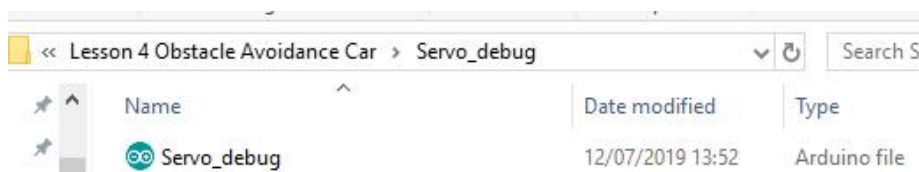
I . Connection

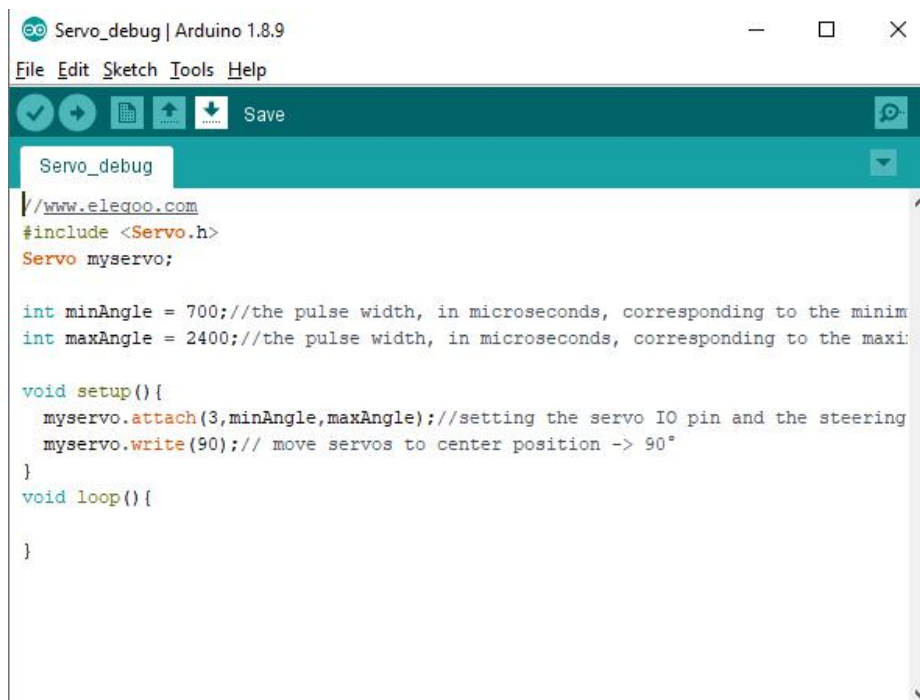
Tips: Because our products have been corrected accurately when they are manufactured, you can skip over the following step "Connection", if you don't remove the servo and the ultrasonic module.

When assemble the ultrasonic sensor module holder, the servo should also be debugged to ensure that the server can rotate 180 degrees.



*STEP1: Connect the UNO to the computer and open the Servo_debug code file in the path "**\Lesson 4 Obstacle Avoidance Car\Servo_debug\ Servo_debug.ino**".*





```

Servo_debug | Arduino 1.8.9
File Edit Sketch Tools Help

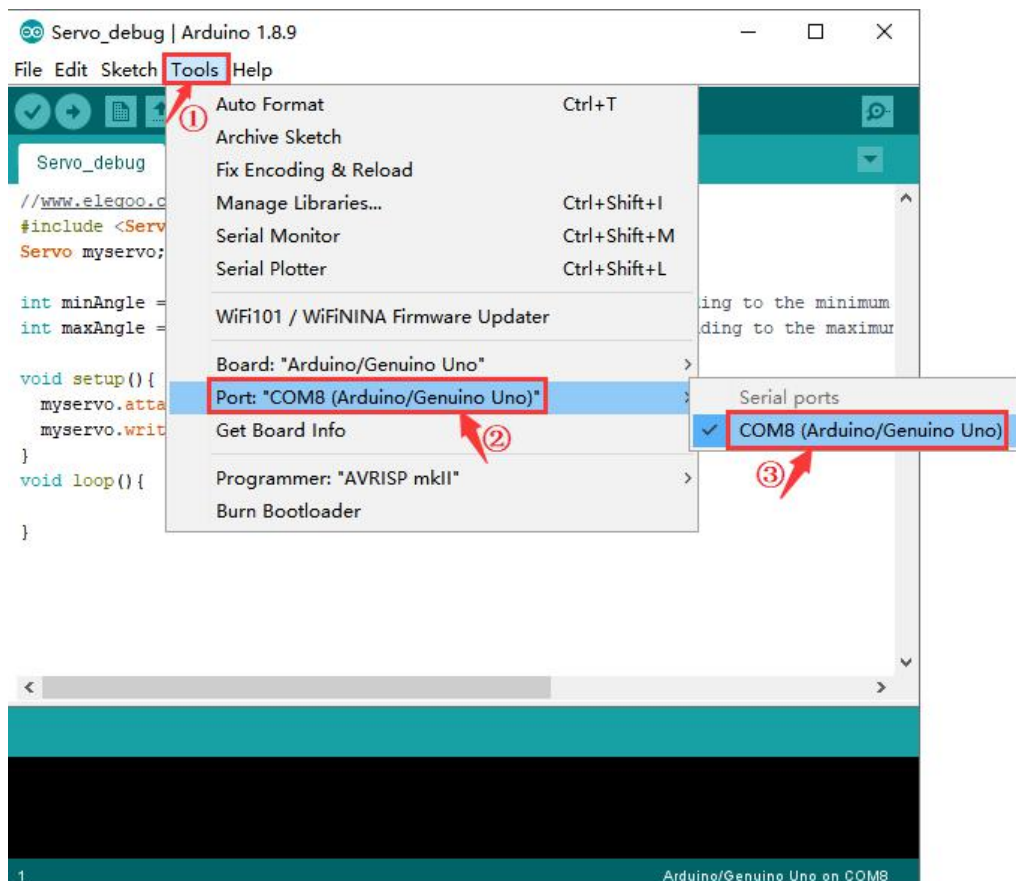
Servo_debug

//www.elegoo.com
#include <Servo.h>
Servo myservo;

int minAngle = 700;//the pulse width, in microseconds, corresponding to the minim
int maxAngle = 2400;//the pulse width, in microseconds, corresponding to the maxi

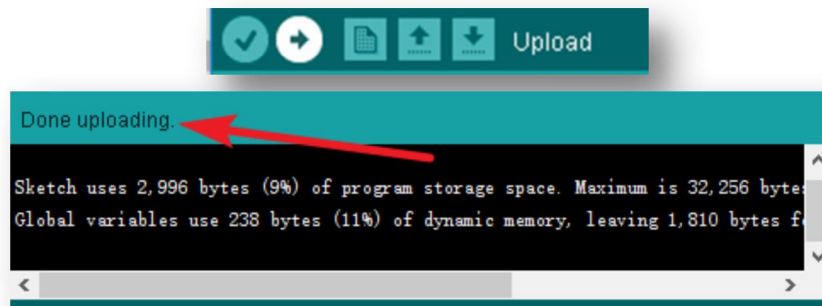
void setup() {
  myservo.attach(3,minAngle,maxAngle);//setting the servo IO pin and the steering
  myservo.write(90);// move servos to center position -> 90°
}
void loop() {
}
  
```

STEP2: Select "Tool" --> "Port" and "Board" in the Arduino IDE.



Tips: If you have any questions or run into any problems during assembling and testing Smart Robot Car please feel free to contact us at service@elegoo.com or euservice@elegoo.com (Europe customers).

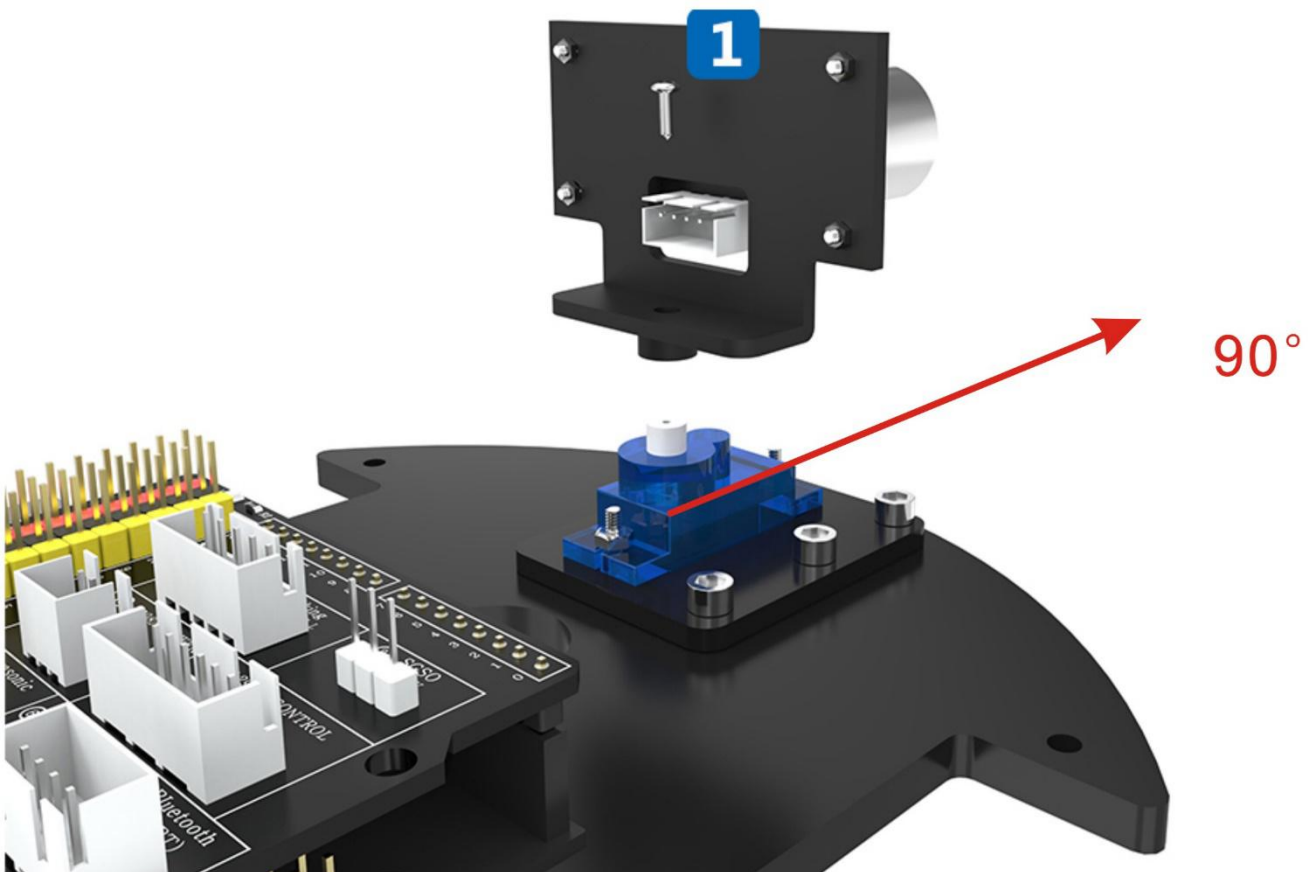
STEP3: Click the arrows button to upload the code to the UNO controller board.



After uploading, the servo will rotate to 90-degree angles and then become stationary.

STEP4: Assemble the ultrasonic sensor module at 90 degrees.

The angle of each teeth on micro servo is 15 degrees and if you install it on the middle of the direction of 90 degree, it will rotate to left or right by 15 degrees, which means the actual degree of installing the micro servo is 85 degrees or 105 degrees.



Tips: If you have any questions or run into any problems during assembling and testing Smart Robot Car please feel free to contact us at service@elegoo.com or euservice@elegoo.com (Europe customers).

Attention : FAQ about the servo motor.

1 Why does the micro servo rotate anticlockwise by 15 degrees each time I turn on the power?

This is normal for SG90 micro servo and it won't affect normal use of the program.

If you didn't control it with program, you can rotate it back to normal with your hand or plug off the wires, which connected with micro servo, before you turn on the power.

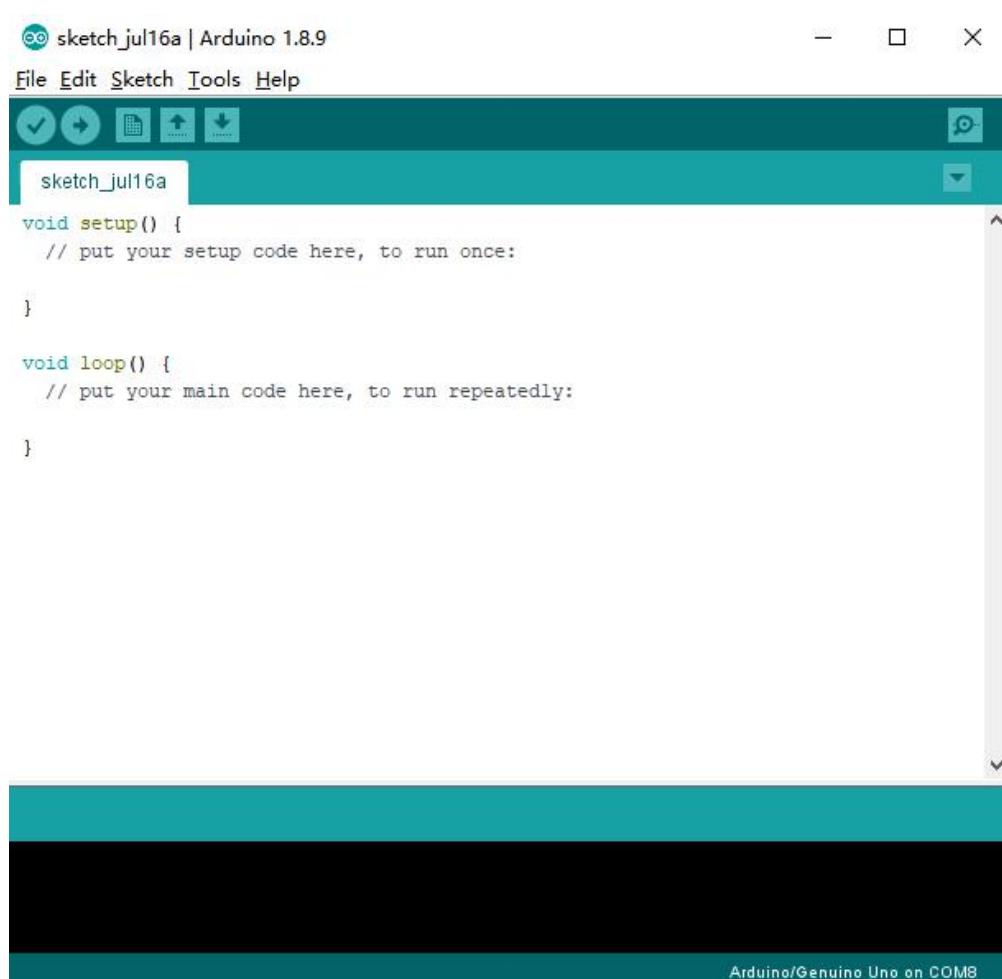
2 Why is the micro servo out of control and keeping rotating?

Use "myservo.attach(3,700,2400)" to command the micro servo turn to the specific angle which has a range from 0 to 180. If it exceed the range, the micro servo won't recognize this angle and will keep rotating.

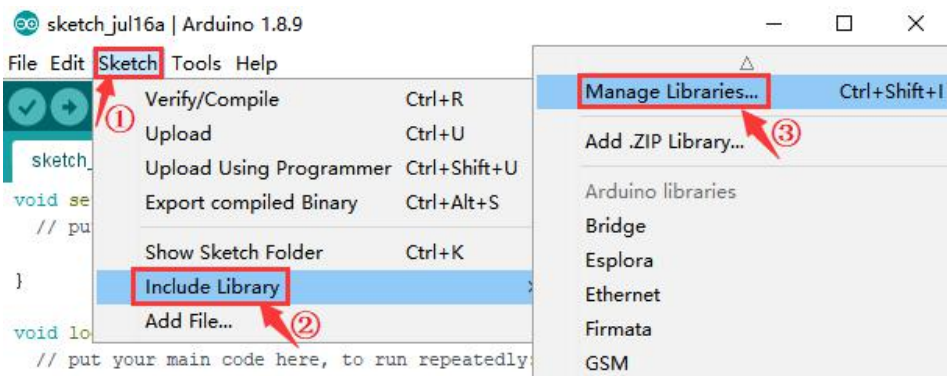
II. Upload program

Because the program uses the library <servo.h>, so we need to install the library at first.

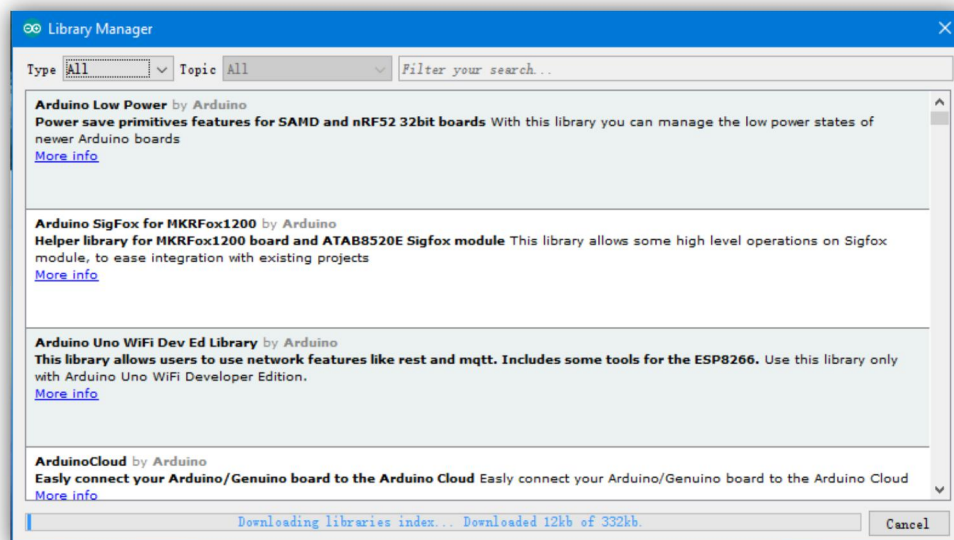
Open the Arduino software



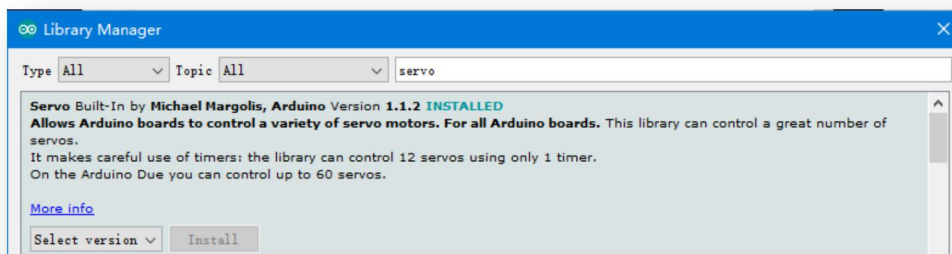
Select Sketch -> Include Library -> Manage Libraries



Waiting for “Downloading libraries index” to finish.

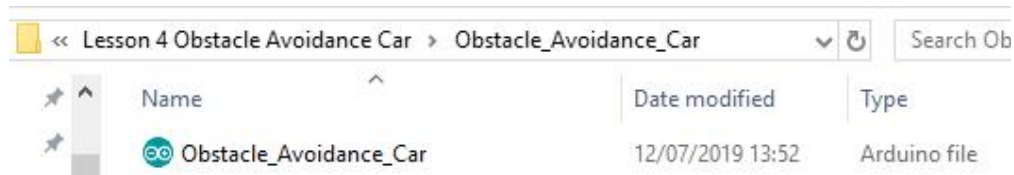


Search servo and then install the newest version. The following picture shows that the Servo library is already installed.



Connect the UNO controller board to the computer, open the code file in the path “\Lesson 4 Obstacle Avoidance Car\Obstacle_Avoidance_Car\Obstacle_Avoidance_Car.ino”.

Upload the program to the UNO board.



After uploading the program to the UNO control board, disconnect the cable, put the vehicle on the ground and switch on the power supply.

You will see that the vehicle will move forward and the cloud platform keeps rotating to make the distance measuring sensors operate continuously. If there are obstacles ahead, the cloud platform will stop and the vehicle will change its direction to bypass the obstacle. After bypassing the obstacle, the cloud platform will keep rotating again and the vehicle will also move on.

III. Introduction of principle

First of all, let's learn about the SG90 Servo:

SG90 Servo

180 angle steering gear

Rotation angle is from 0 to 180

Brown line —GND

Red line —SV

Orange line —signal(PWM)



Classification: 180 servo

Normally the servo has 3 controlling wires: power wire, ground wire and signal wire.

Definition of the servo pins: brown line——GND, red line——5V, orange line——signal.

How does servo work:

The signal modulation chip in the servo receives signals from the controller board then the servo will get the basic DC voltage. There is also a reference circuit inside the servo which will produce a standard voltage. These two voltages will compare to each other and the difference will be output. Then the motor chip will receive the difference and decide the rotational speed, direction and angel. When there is no difference between the two voltages, the servo will stop.

How to control the servo:

To control the servo rotation, you need to make the time pulse to be about 20ms and the high level pulse width to be about 0.5ms~2.5ms, which is consistent with the angle limited of the servo.

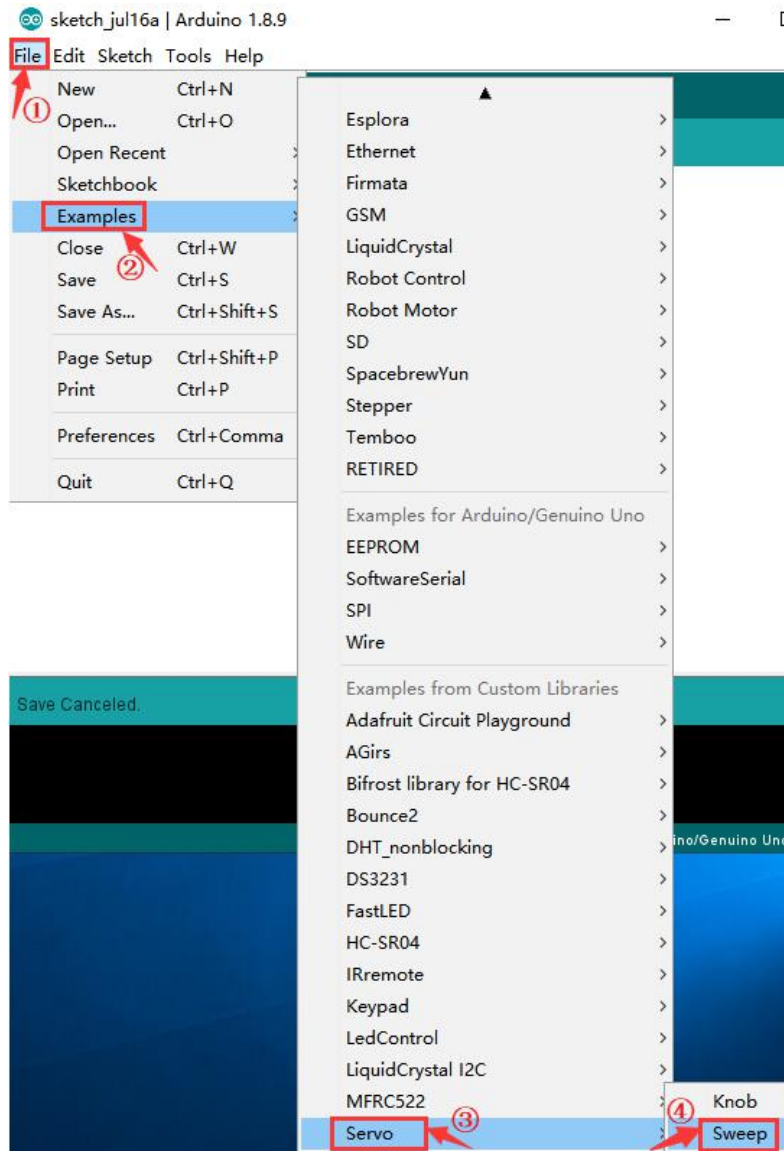
Taking 180 angle servo for example, corresponding control relation is as below:

0.5ms	0 degree
1.0ms	45 degree
1.5ms	90 degree
2.0ms	135 degree
2.5ms	180 degree

The example program:

Open Arduino IDE and select

“File->Examples->Servo->Sweep”



Next, let's have a look at the ultrasonic sensor module.



Feature of the module: testing distance, high precision module.

Application of the products: obstacle-avoidance robot、object distance testing、liquid testing、public security、 parking lot testing.

Main technical parameters

- (1): voltage used: DC---5V
- (2): static current: less than 2mA
- (3): level output: higher than 5V
- (4): level output: lower than 0
- (5): detection angle: not bigger than 15 degree
- (6): detecting distance: 2cm-450cm
- (7): high precision: up to 0.2cm

Method of connecting lines: VCC, trig (the end of controlling), echo (the end of receiving), GND

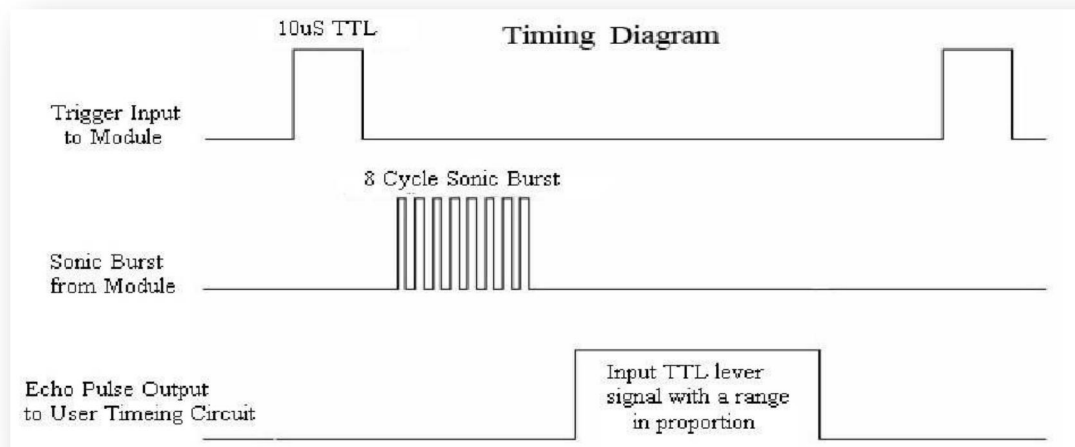
How does the module work:

- (1)Apply IO port of TRIG to trigger ranging, give high level signal, at least 10us one time;
- (2)The module sends 8 square waves of 40kHz automatically, tests if there are signals returned automatically;
- (3)If there are signals received, the module will output a high level pulse through IO port of ECHO, the duration time of high level pulse is the time between the wave sending and receiving. So the module can know the distance according to the time.

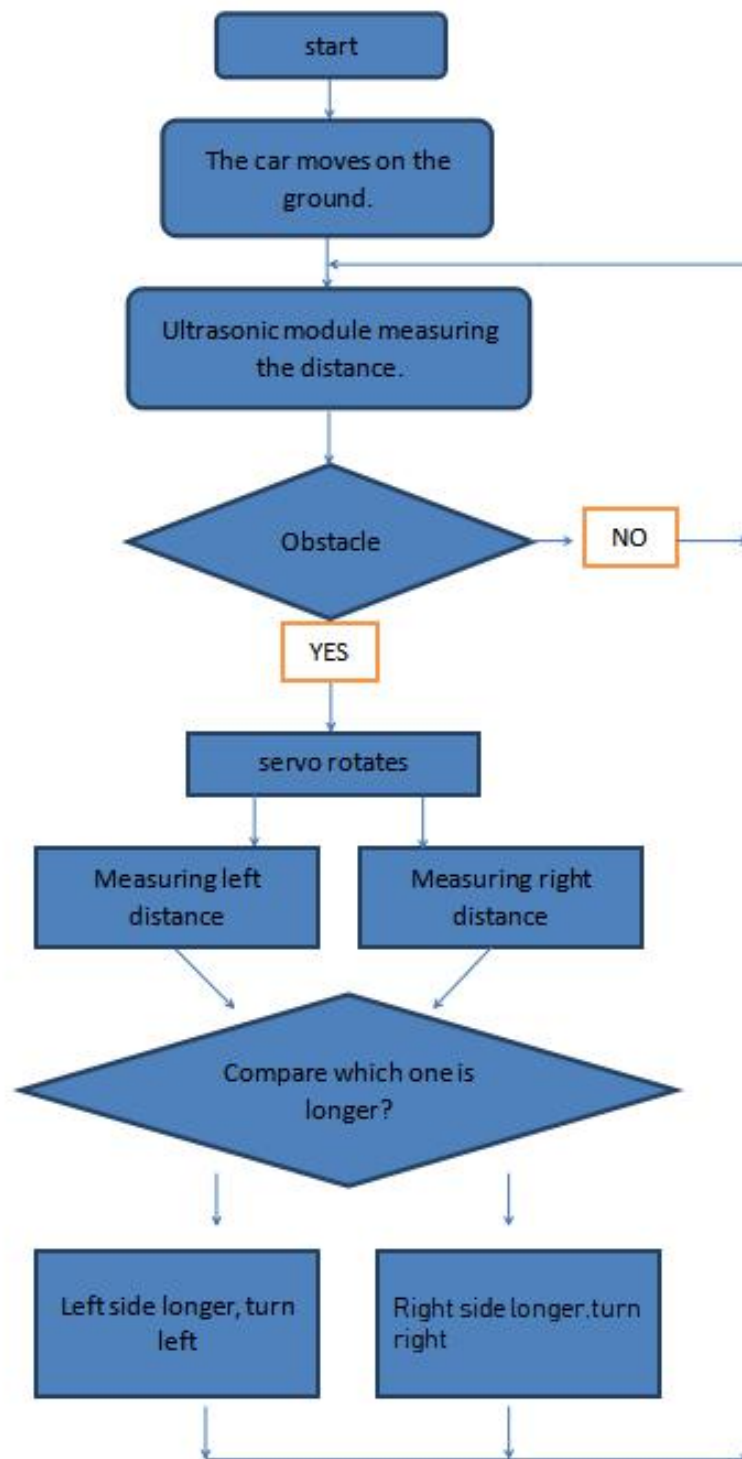
$$\text{Testing distance} = (\text{high level time} * \text{velocity of sound (340M/S)}) / 2;$$

Actual operation:

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; Or: The range = high level time * velocity (340M/S) / 2; we suggest you to use over 60ms measurement cycle, in order to prevent trigger signal back to the echo signal.



```
/*Ultrasonic distance measurement Sub function*/  
  
int Distance_test()  
{ digitalWrite(Trig, LOW);  
  
  delayMicroseconds(2);  
  
  digitalWrite(Trig, HIGH);  
  
  delayMicroseconds(20);  
  
  digitalWrite(Trig, LOW);  
  
  float Fdistance = pulseIn(Echo, HIGH);  
  
  Fdistance= Fdistance/58;  
  
  return (int)Fdistance;  
}
```



From the above picture, we can see that the principle of obstacle avoidance car is very simple. The ultrasonic sensor module will detect the distance between the car and the obstacles again and again and sending the data to the controller board, then the car will stop and rotate the servo to detect the left side and right side. After compared the distance from the different side, the car turn to the side which has a longer distance and move forward. Then the ultrasonic sensor module continue to detect the distance between the surrounding obstacles and itself.

Code preview:

```
if(rightDistance > leftDistance) {  
    right();  
    delay(360);  
}  
else if(rightDistance < leftDistance) {  
    left();  
    delay(360);  
}  
else if((rightDistance <= 40) || (leftDistance <= 40)) {  
    back();  
    delay(180);  
}  
else {  
    forward();  
}
```