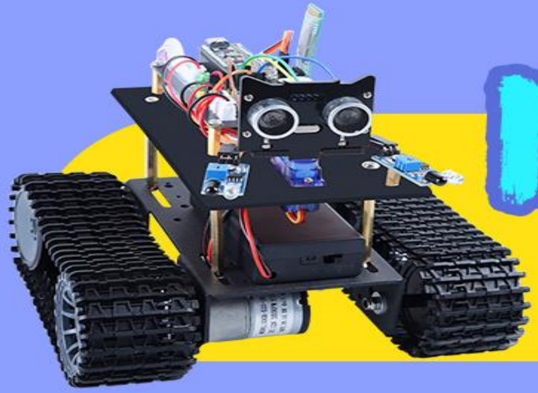


NANOBOT
ASSEMBLY TUTORIAL

Lesson 2 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.

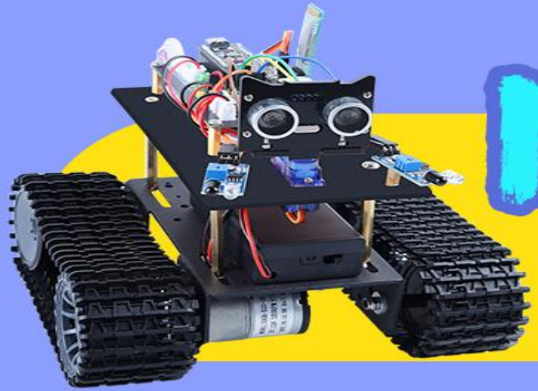
Learning Objectives:

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

智懿 ZHIYI

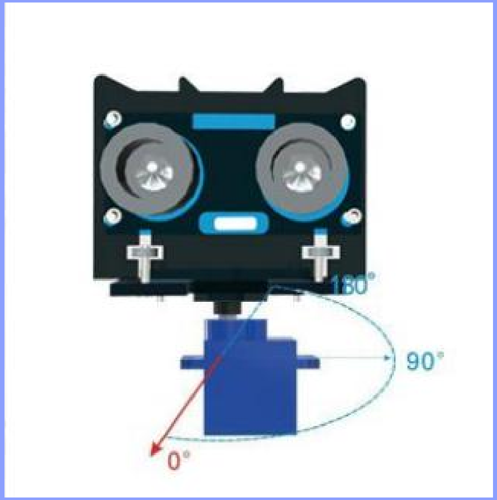


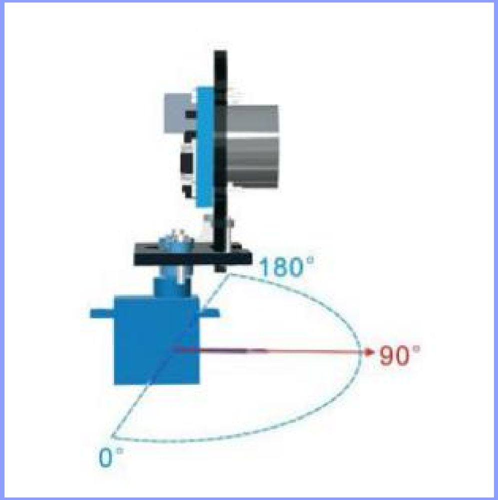
NANOBOT ASSEMBLY TUTORIAL

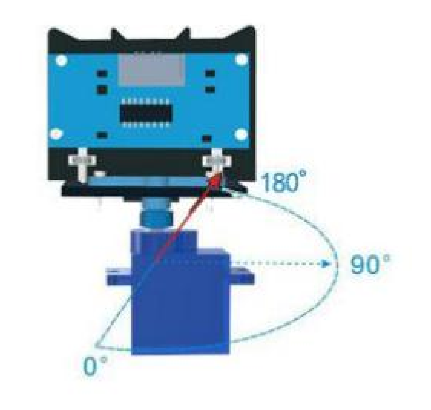
① Connection

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.

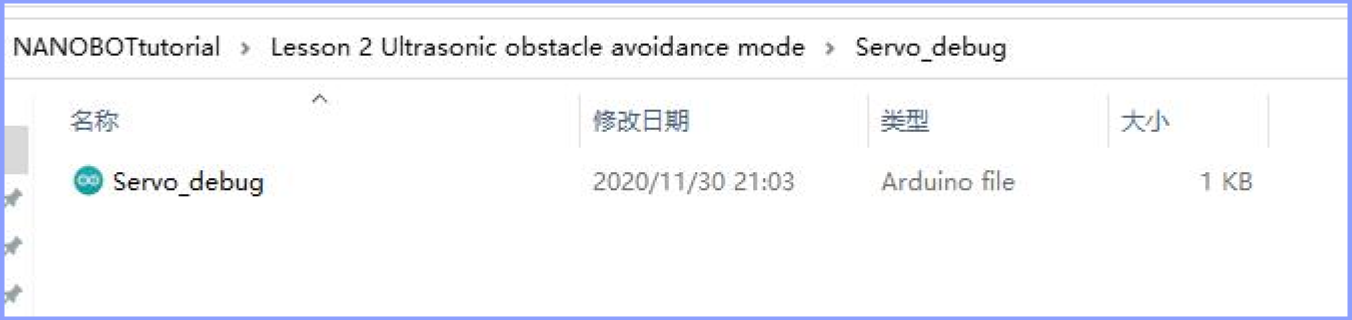






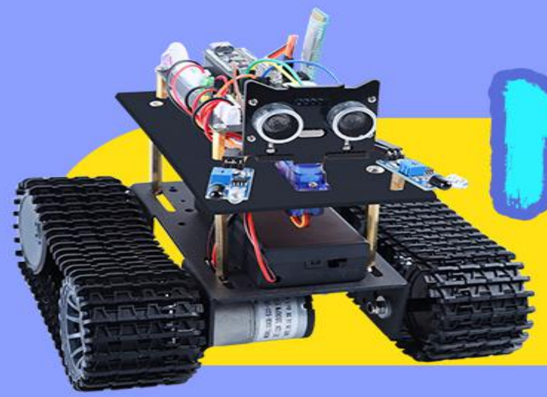
智懿 ZHIYI

STEP1: Connect the RF-NANO to the computer and open the Servo_debug code file in the path “\Lesson 2 Ultrasonic obstacle avoidance mode\Servo_debug\Servo_debug.ino”.

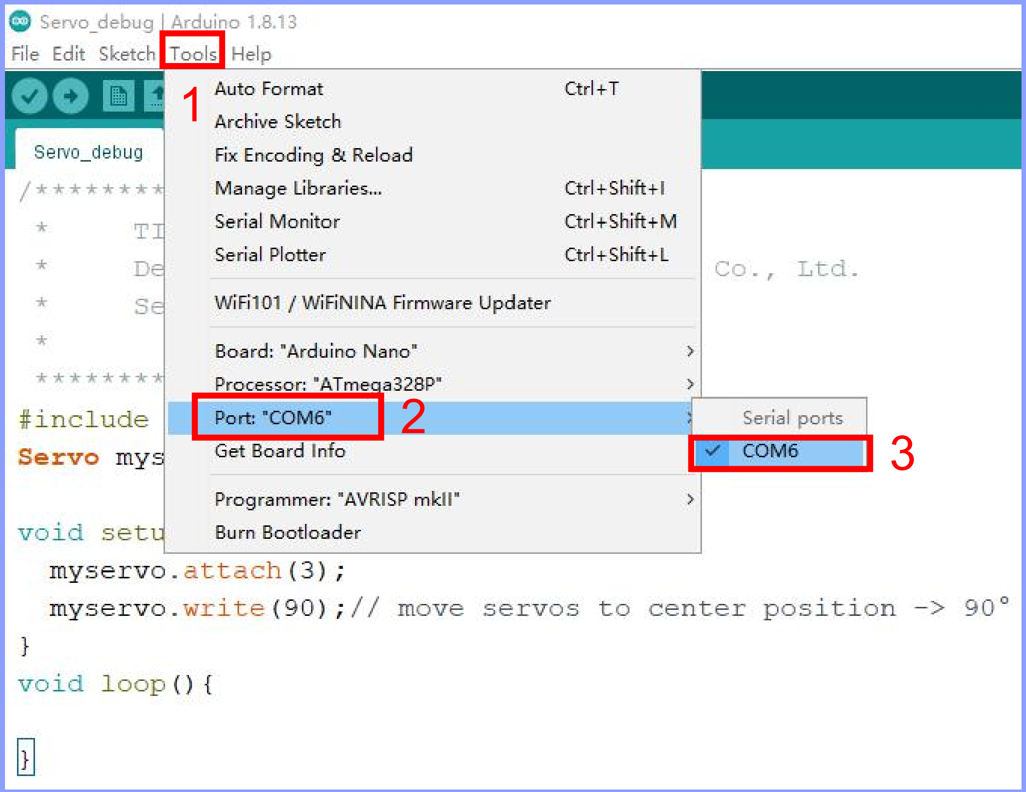


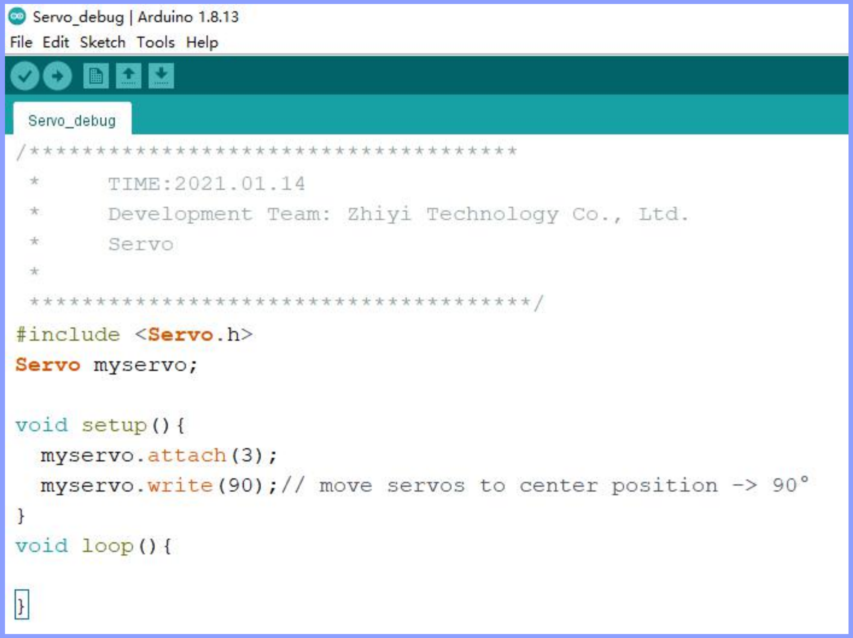
NANOBOTtutorial › Lesson 2 Ultrasonic obstacle avoidance mode › Servo_debug

名称	修改日期	类型	大小
Servo_debug	2020/11/30 21:03	Arduino file	1 KB

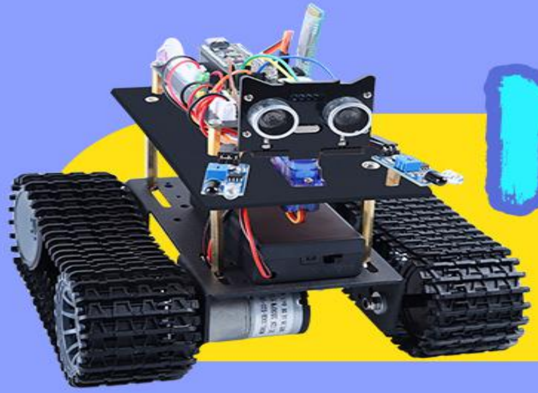


STEP2: Select “Tool” --> ” Port” and “Board” in the Arduino IDE.


Servo_debug | Arduino 1.8.13
File Edit Sketch Tools Help
1
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Nano"
Processor: "ATmega328P"
Port: "COM6"
2
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM6
3
Servo_debug



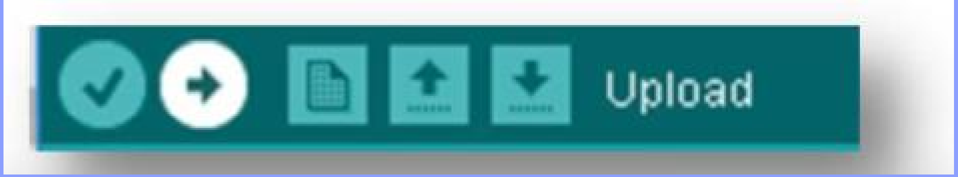
```
/***************************************
 *     TIME:2021.01.14
 *     Development Team: Zhiyi Technology Co., Ltd.
 *     Servo
 *
 ***************************************/
#include <Servo.h>
Servo myservo;

void setup(){
  myservo.attach(3);
  myservo.write(90);// move servos to center position -> 90°
}
void loop(){

}
```

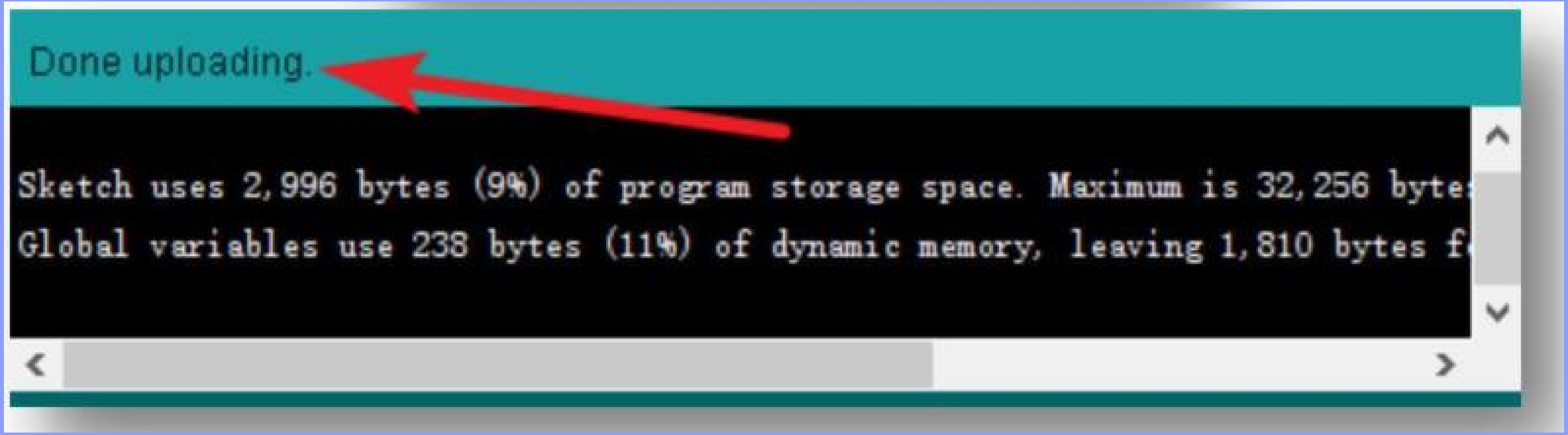


NANOBOT

ASSEMBLY TUTORIAL

STEP3: Click the arrows button to upload the code to the RF-NANO controller board.

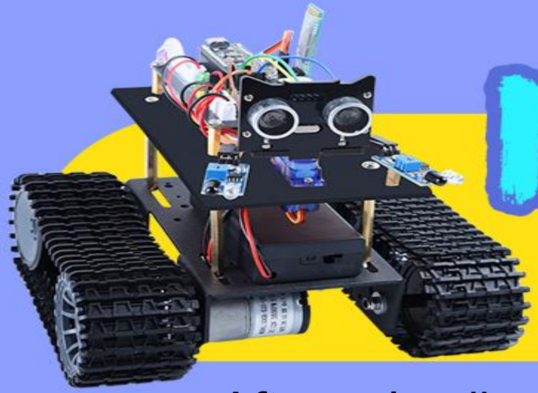




Done uploading.

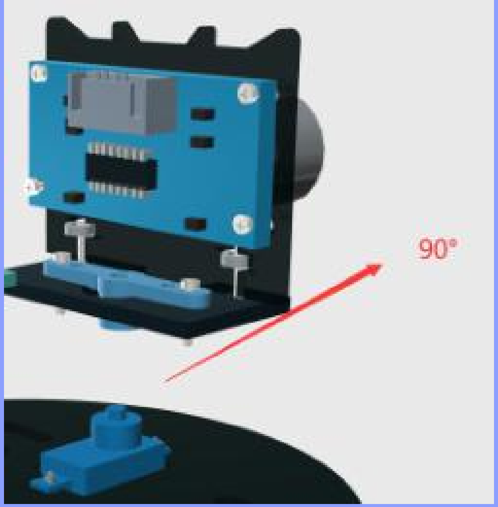
```
Sketch uses 2,996 bytes (9%) of program storage space. Maximum is 32,256 byte
Global variables use 238 bytes (11%) of dynamic memory, leaving 1,810 bytes f
```

智懿 ZHIYI

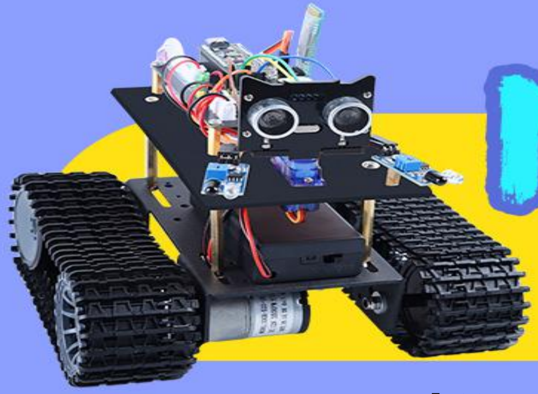


After uploading, the servo will rotate to 90-degree angles and then become stationary.

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.



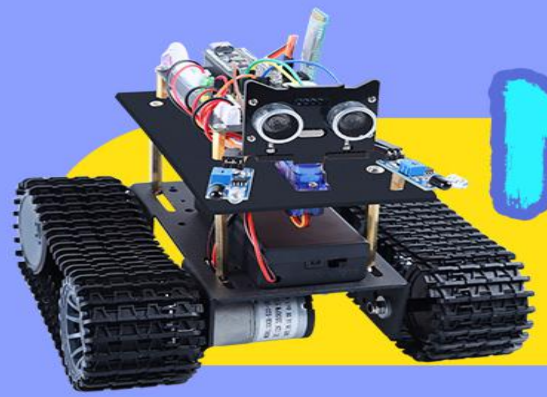
Attention : FAQ about the servo motor.

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.

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

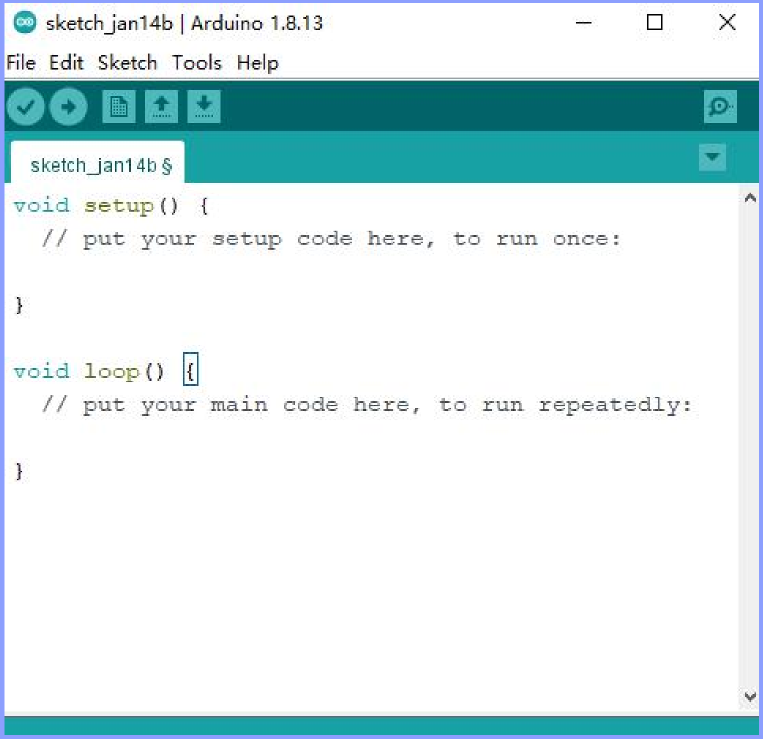
Use “myservo.write(pos);” to command the micro servo turn to the specific angle which has a range from 0 to 180. If it exceed the range, the micro servo won't recognize this angle and will keep rotating.



② Upload program

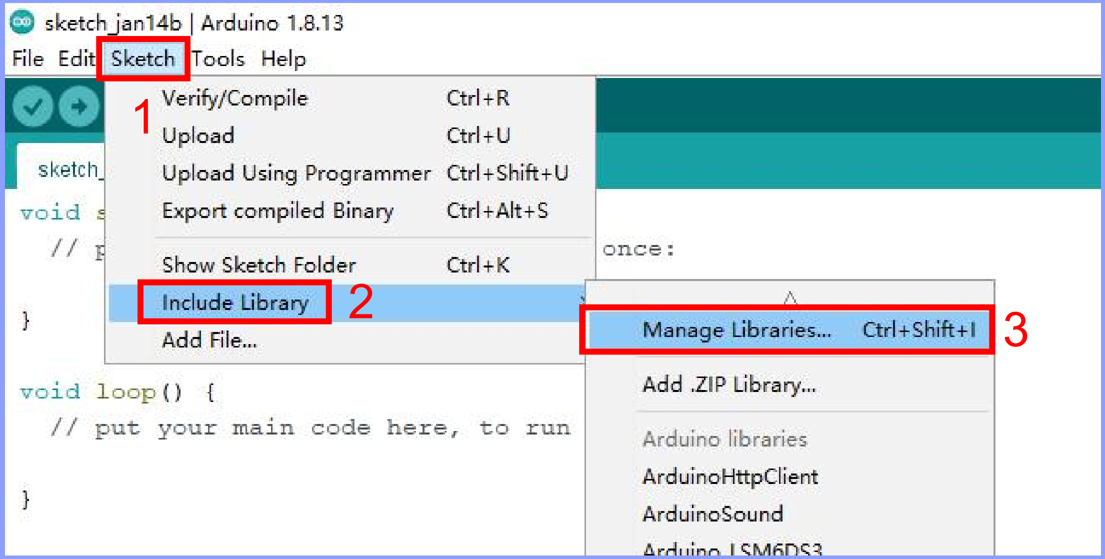
Because the program uses the library <servo.h>, so we need to install the library at first.

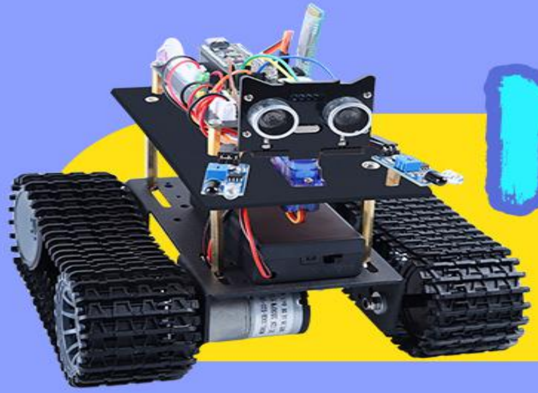
Open the Arduino software



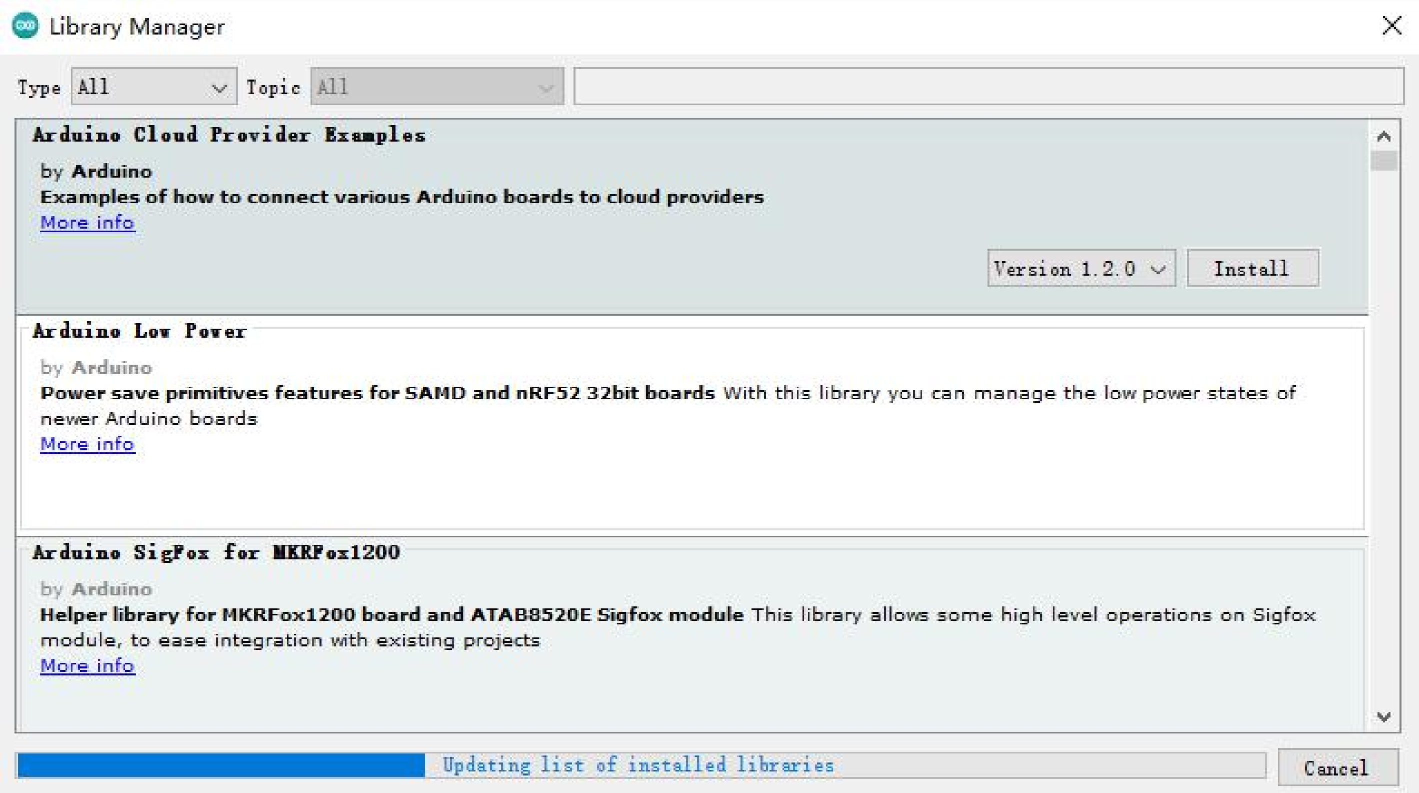
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Select Sketch -> Include Library -> Manage Libraries


sketch_jan14b | Arduino 1.8.13
File Edit Sketch Tools Help
1
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
2
Add File...
Manage Libraries... Ctrl+Shift+I
3
Add .ZIP Library...
Arduino libraries
ArduinoHttpClient
ArduinoSound
Arduino_LSM6DS3



Waiting for “Updating list of installed libraries” to finish.



Library Manager

Type All Topic All

Arduino Cloud Provider Examples

by **Arduino**

Examples of how to connect various Arduino boards to cloud providers

More info

Version 1.2.0 Install

Arduino Low Power

by **Arduino**

Power save primitives features for SAMD and nRF52 32bit boards With this library you can manage the low power states of newer Arduino boards

More info

Arduino SigFox for MKRFox1200

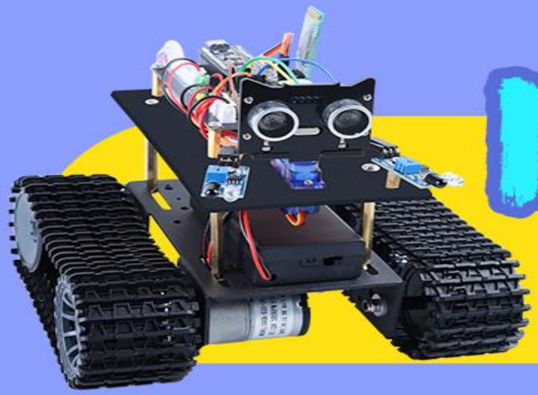
by **Arduino**

Helper library for MKRFox1200 board and ATAB8520E Sigfox module This library allows some high level operations on Sigfox module, to ease integration with existing projects

More info

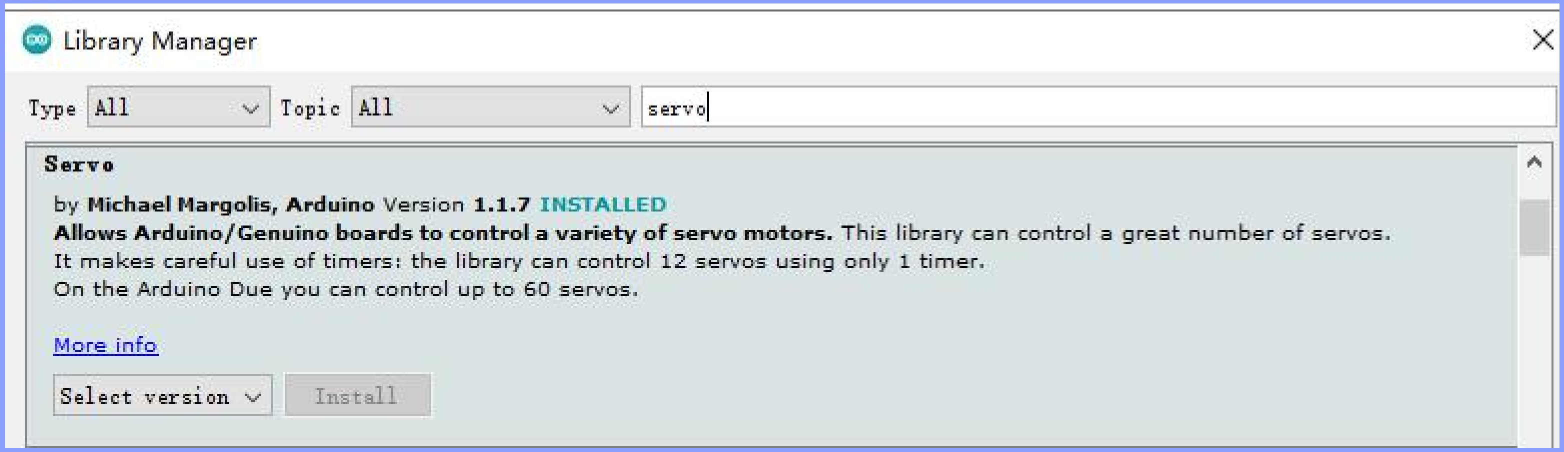
Updating list of installed libraries

Cancel



Search servo and then install the newest version.

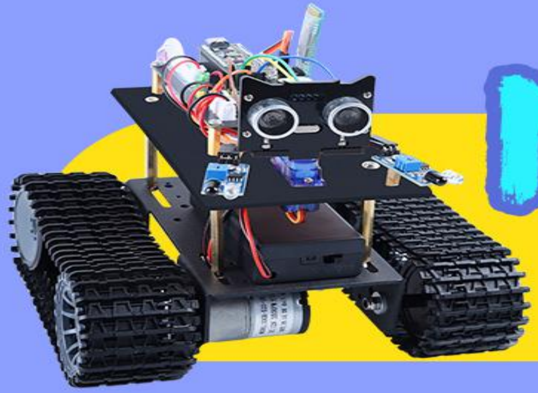
The following picture shows that the Servo library is already installed.



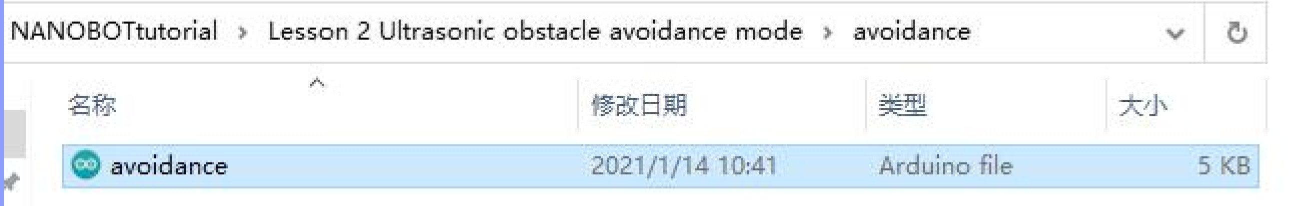
servo

Servo

by **Michael Margolis, Arduino** Version **1.1.7** INSTALLED
Allows Arduino/Genuino boards to control a variety of servo motors. This library can control a great number of servos.
It makes careful use of timers: the library can control 12 servos using only 1 timer.
On the Arduino Due you can control up to 60 servos.



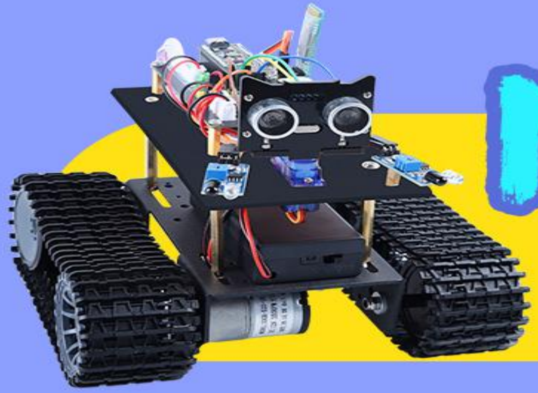
Connect the RF-NANO board to the computer, open the code file in the path
"\Lesson 2 Ultrasonic obstacle avoidance mode\avoidance\avoidance.ino.
Upload the program to the RF-NANO board.



NANOBOTtutorial › Lesson 2 Ultrasonic obstacle avoidance mode › avoidance

名称	修改日期	类型	大小
avoidance	2021/1/14 10:41	Arduino file	5 KB

After uploading the program to the RF-NANO 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.



③ Introduction of principle

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

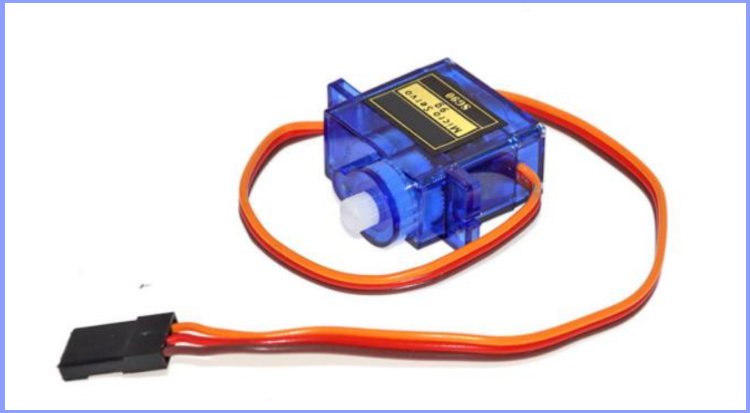
SG90 Servo

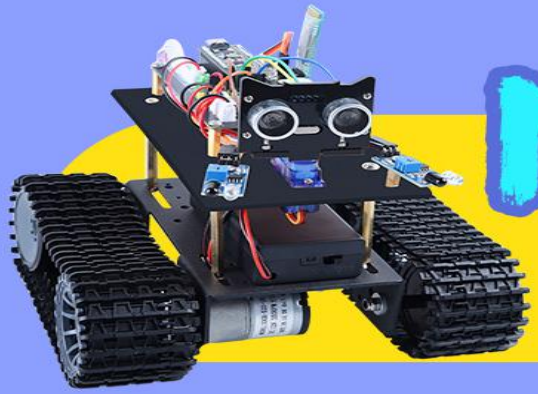
180 angle steering gear Rototion angle is from 0 to 180

Brown line --- GND

Red line --- SV

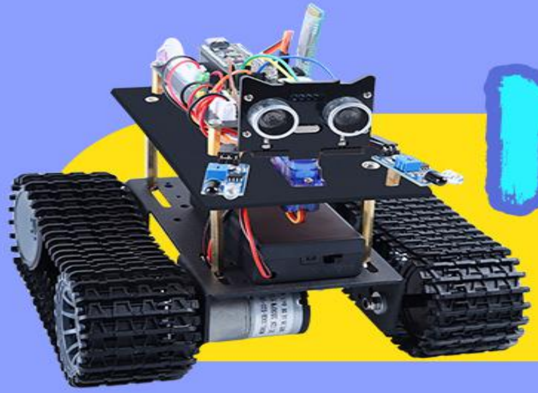
Orange line --- signal(PWM)





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.

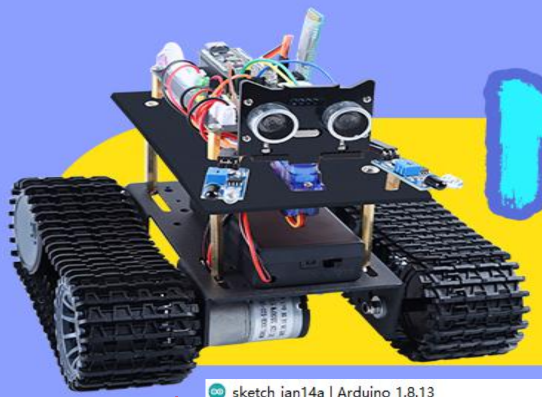


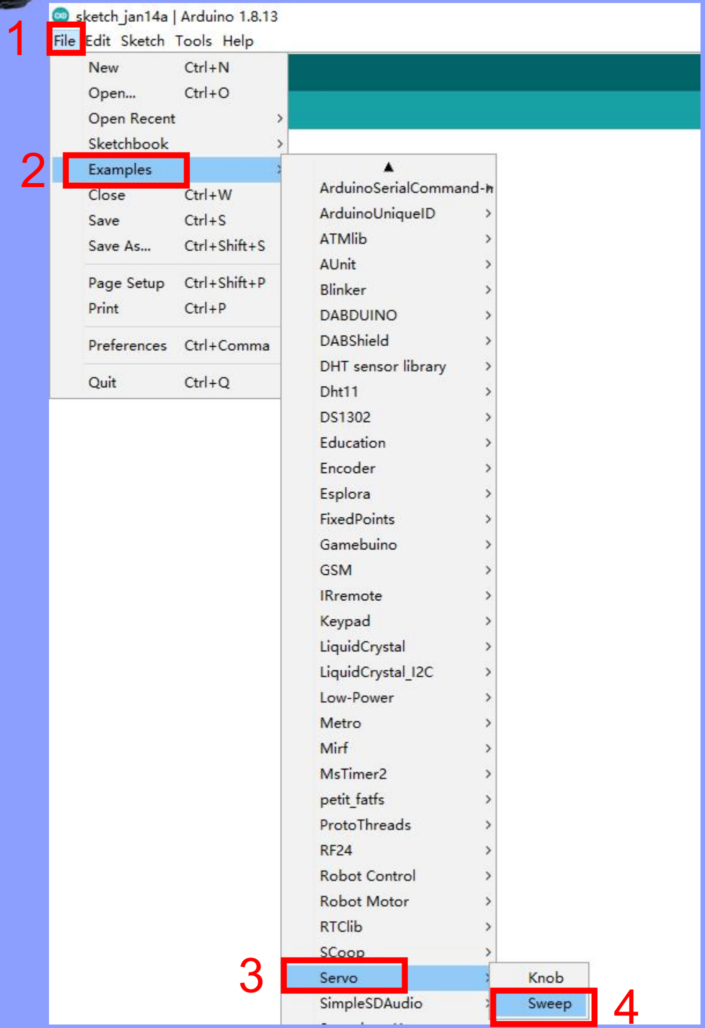
How to control the servo:

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**

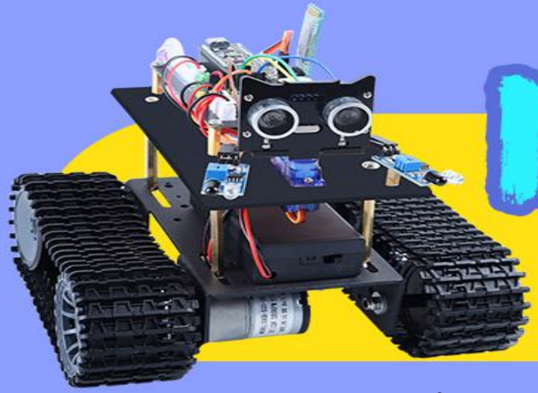




The example program:

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.

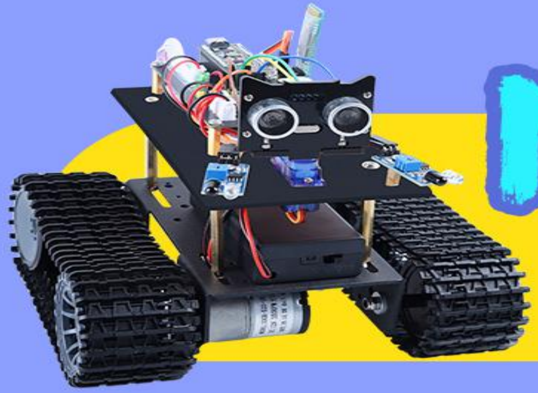
Main technical parameters:

(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

智懿
ZHIYI



How does the module work:

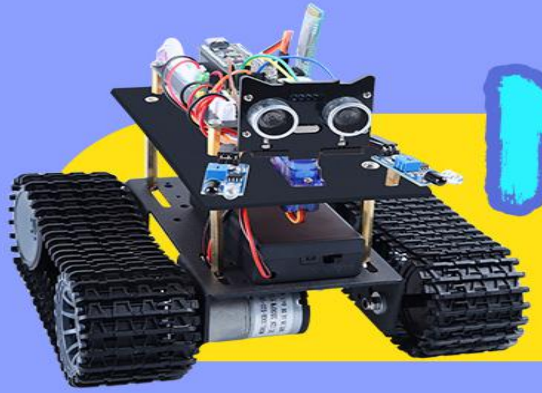
(1)Apply IO port of TRIG to trigger ranging, give high level signal, at least 10us one time;

(2)The module sends 8 square waves of 40kz 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.

Testing distance= (high level time* velocity of sound (340M/S))/2);

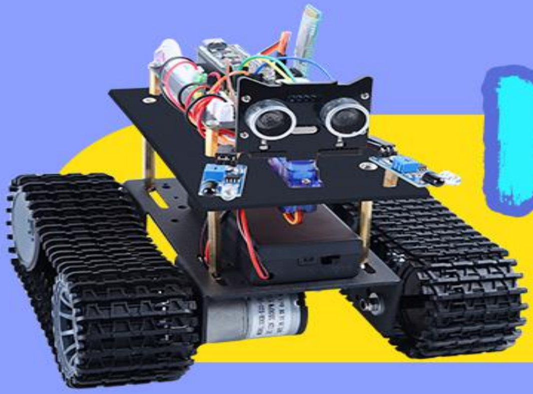


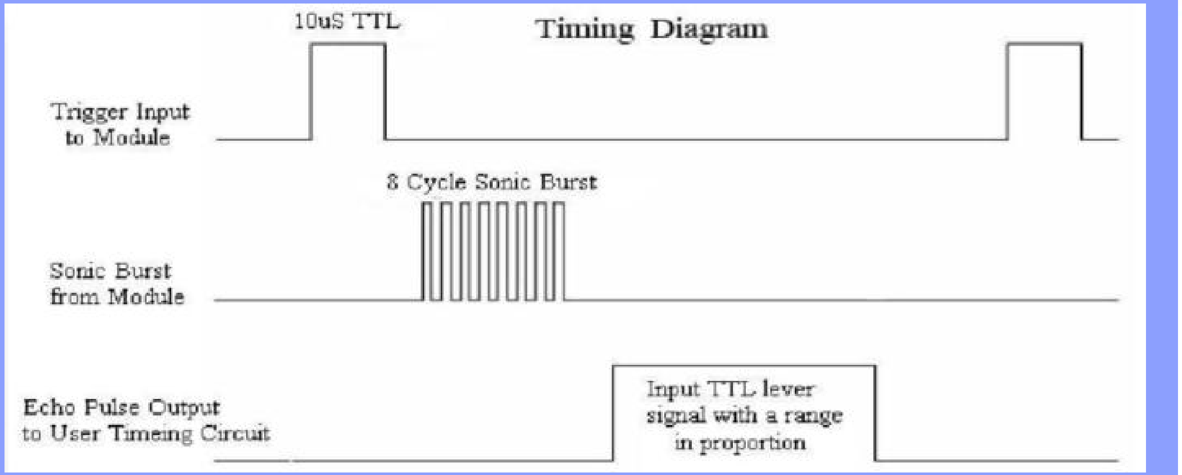
Actual operation:

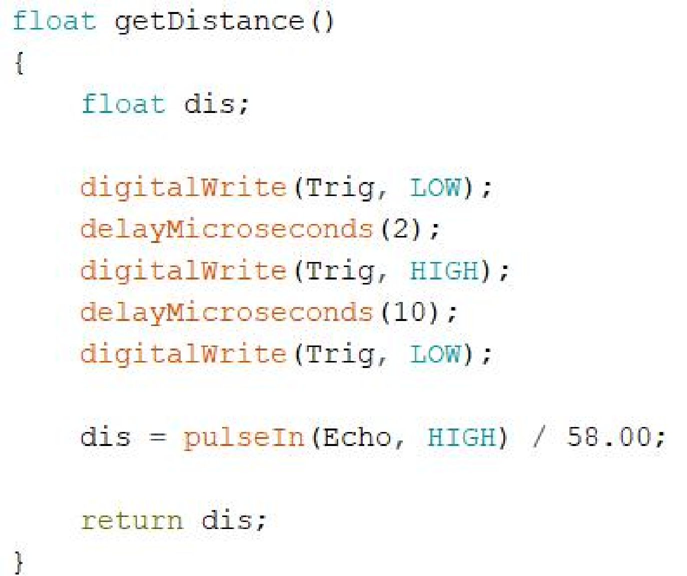
The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal.

Formula: uS / 58 = centimeters or uS / 148 =inch;
Or: The range = high level time *velocity (340M/S) / 2;

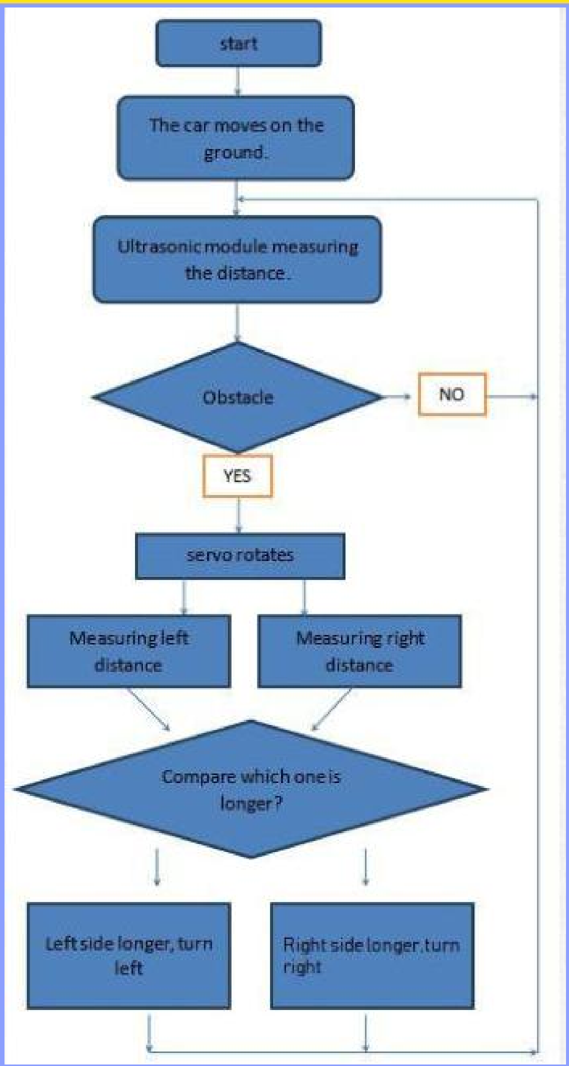
we suggest you to use over 60ms measurement cycle, in order to prevent trigger signal back to the echo signal.

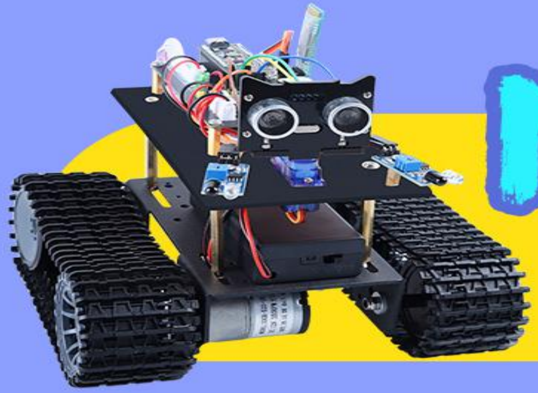



10uS TTL
Timing Diagram
Trigger Input to Module
8 Cycle Sonic Burst
Sonic Burst from Module
Echo Pulse Output to User Timeing Circuit
Input TTL lever signal with a range in proportion

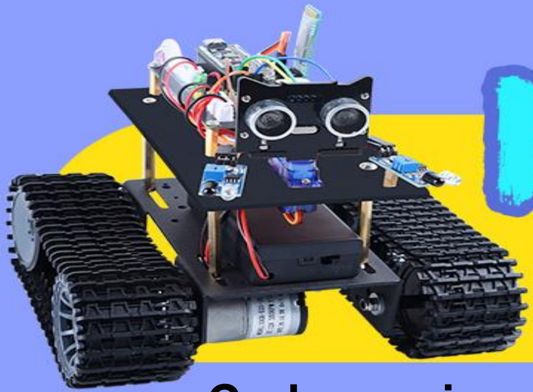


```
float getDistance()
{
    float dis;

    digitalWrite(Trig, LOW);
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);

    dis = pulseIn(Echo, HIGH) / 58.00;

    return dis;
}
```

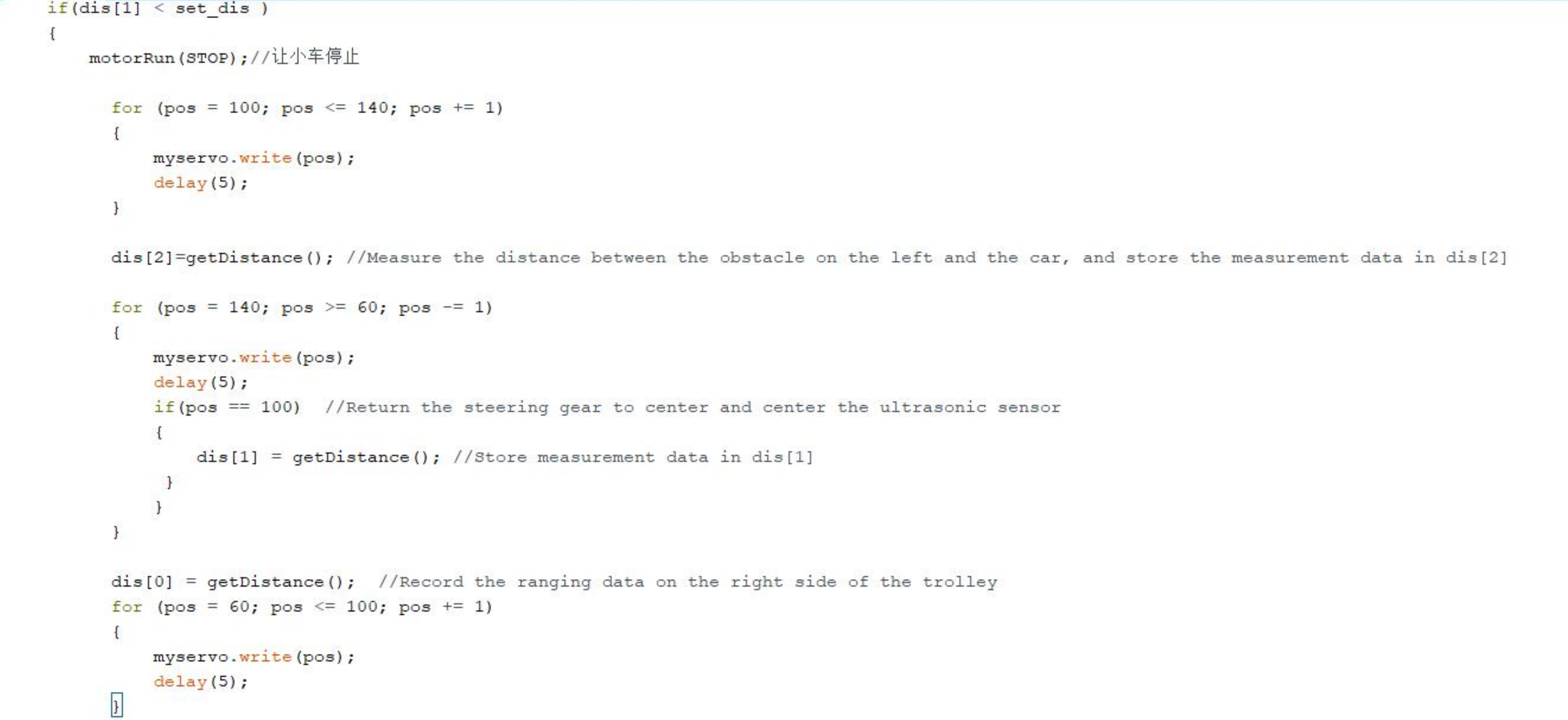

start
The car moves on the ground.
Ultrasonic module measuring the distance.
Obstacle
NO
YES
servo rotates
Measuring left distance
Measuring right distance
Compare which one is longer?
Left side longer, turn left
Right side longer, turn right



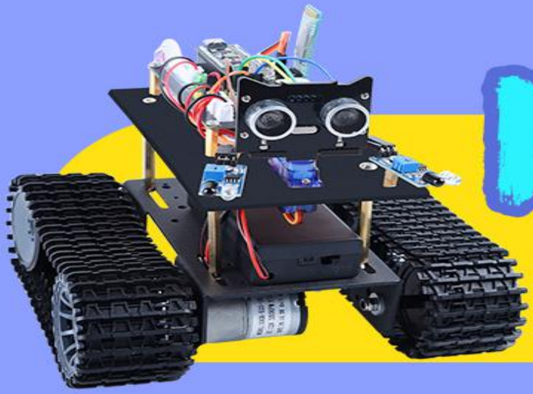
From the above picture, we can see that the principle of obstacle avoidance car is very simple. The ultrasonic sensor module will detect the distance between the car and the obstacles again and again and sending the data to the controller board, then the car will stop and rotate the servo to detect the 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.

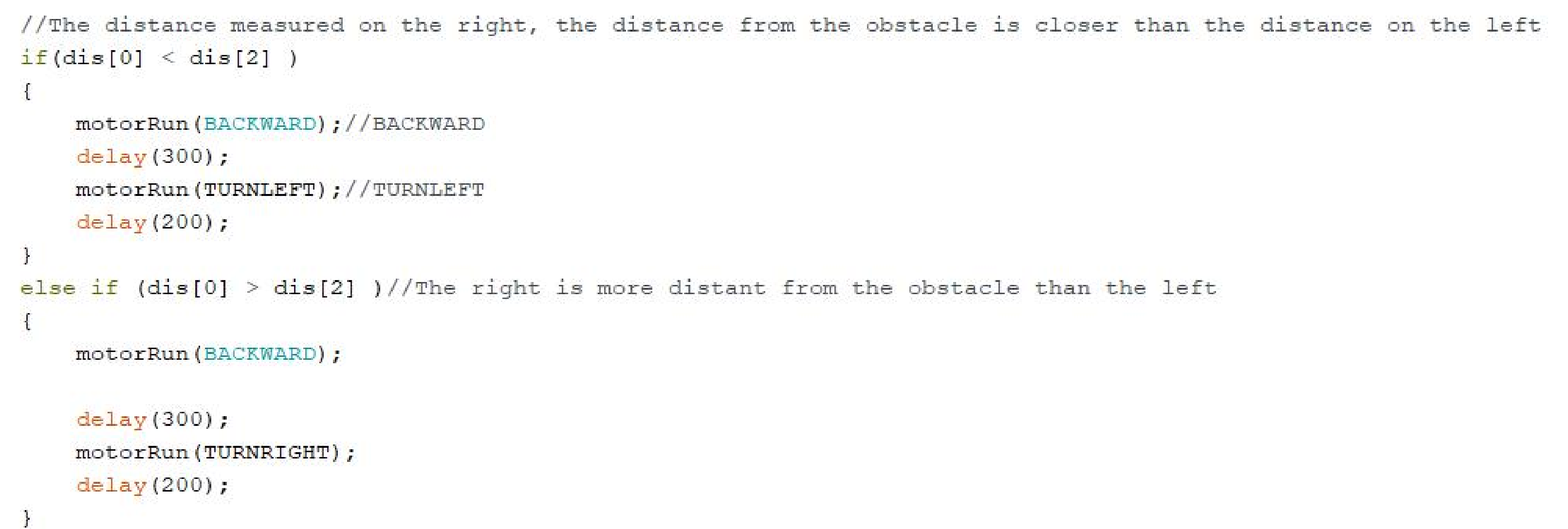


Code preview:

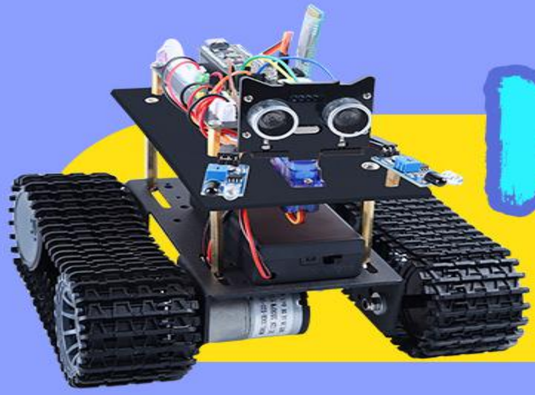


```
if(dis[1] < set_dis )
{
    motorRun(STOP);//让小车停止

      for (pos = 100; pos <= 140; pos += 1)
      {
          myservo.write(pos);
          delay(5);
      }

      dis[2]=getDistance(); //Measure the distance between the obstacle on the left and the car, and store the measurement data in dis[2]

      for (pos = 140; pos >= 60; pos -= 1)
      {
          myservo.write(pos);
          delay(5);
          if(pos == 100)  //Return the steering gear to center and center the ultrasonic sensor
          {
              dis[1] = getDistance(); //Store measurement data in dis[1]
           }
          }
      }

      dis[0] = getDistance();  //Record the ranging data on the right side of the trolley
      for (pos = 60; pos <= 100; pos += 1)
      {
          myservo.write(pos);
          delay(5);
      }
```





```
//The distance measured on the right, the distance from the obstacle is closer than the distance on the left
if(dis[0] < dis[2] )
{
    motorRun(BACKWARD);//BACKWARD
    delay(300);
    motorRun(TURNLEFT);//TURNLEFT
    delay(200);
}
else if (dis[0] > dis[2] )//The right is more distant from the obstacle than the left
{
    motorRun(BACKWARD);

    delay(300);
    motorRun(TURNRIGHT);
    delay(200);
}
```



NANOBOT ASSEMBLY TUTORIAL

Thanks for watching!

智懿 ZHIYI