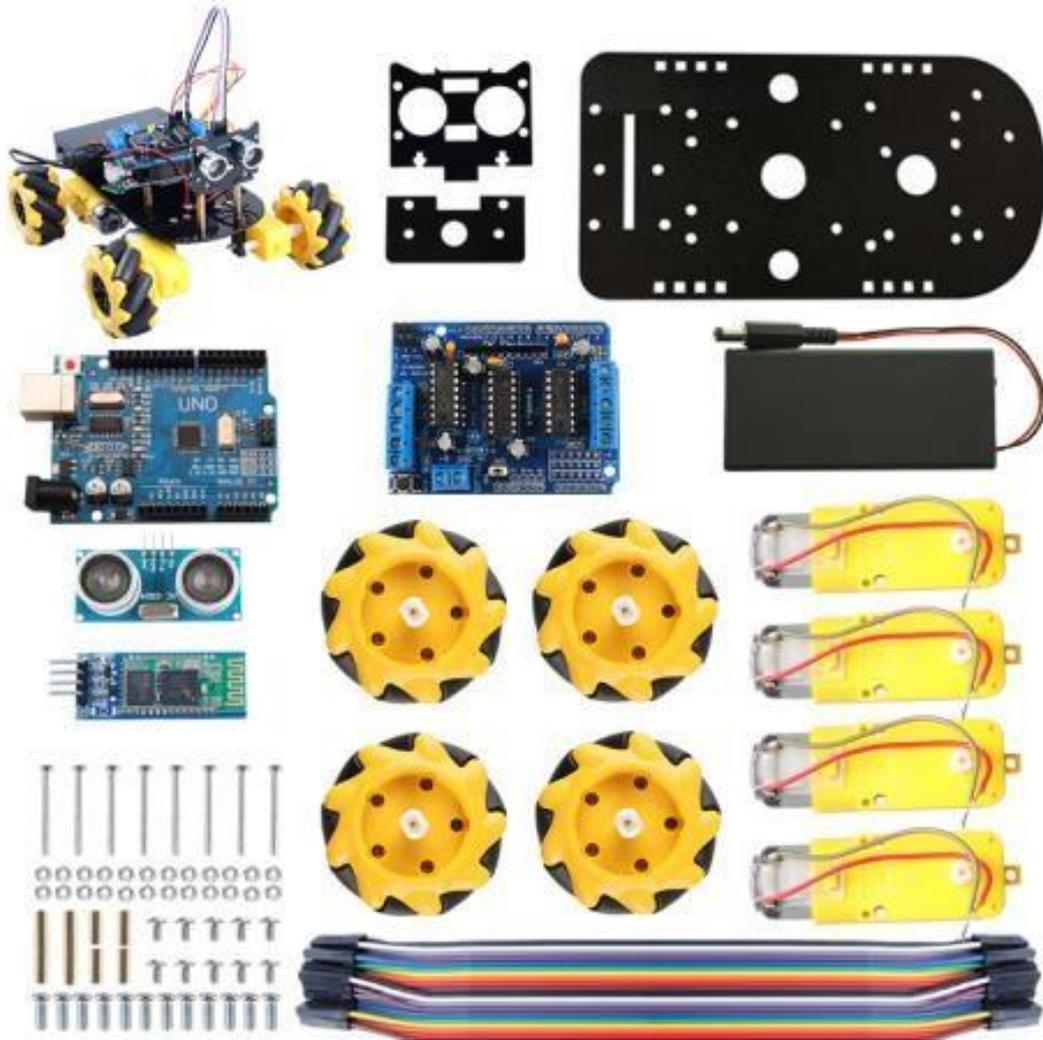


car introduction	1
1 Installing IDE	2
1.1 Introduction	2
1.2 Click Windows Installer	4
1.3 Click JUST DOWNLOAD	4
1.4 Click Install to initiate installation.....	6
1.5 Installing Arduino(Mac OS X)	7
1.6 Installing Arduino(Linux).....	7
2 Add Libraries	8
2.1 Installing Additional Arduino Libraries	8
2.2 What are Libraries	8
2.3 How to Install a Library	8
2.4 Example:SR04	9
2.5 Add method two:.....	9
3 Ultrasonic transducer	10
3.1 Introduction	10
3.2 Obstacle avoidance principle.....	11
3.3 Acquisition of measurement data	12
4 car assembly	15
4.1 motor wiring	15
4.2 Motor fixing screw installation	15
4.3 wheels installed	15
4.5 Fixation of control panel.....	17
4.6 Installing ultrasonic sensors	18
4.7 Install the battery box.....	19
5 Wiring part.....	20
5.1 Ultrasonic sensor wiring	20
5.2 Bluetooth module wiring.....	21

5.3 L293D wiring	22
6 program analysis.....	24
6.1 Motor drive	24
6.2 Upload code.....	25
6.3 Bluetooth remote control.....	27
6.5 Command function corresponding table.....	29
6.6 Virtual serial port	30
7 code analysis.....	31
7.1 Pin configuration.....	31
7.2 Ultrasonic sensor ranging	35
7.3 serial port communication	36
7.4 Set speed	38
7.5 Loop functions	39

car introduction

The basic components of smart car are as follows:



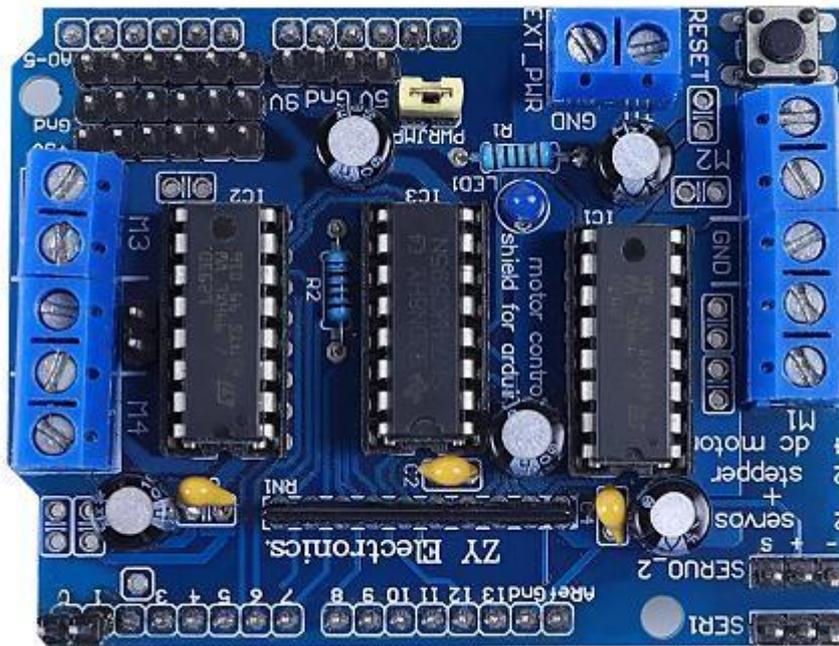
Motor drive module: L293D is selected as the motor drive module, which can drive four-way DC motor;

The L293D motor drive board uses a chip-series-parallel converter called 74HC595. The chip has eight outputs (perfect) and three inputs, and you use it to input data in one bit at a time. At each pulse, if the data pin is high, then a 1 is pushed into the shift register; Otherwise a 0.

When all 8 pulses are received, make the latch needle copy the 8 values to the latch register. This is necessary; Otherwise, when the data is loaded into the shift register, the wrong output, not as you want to control the car's motion.

The chip also has an output enable (OE) pin, which is used to enable or disable all outputs at once.

You can connect it to a PWM UNO pin and then use "analogWrite" to control the speed of the motor. This pin is low activity, so we attach it to GND.



1 Installing IDE

1.1 Introduction

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform.

In this lesson, you will learn how to setup your computer to use Arduino and how to set about the lessons that follow.

The Arduino software that you will use to program your Arduino is available for Windows, Mac and Linux. The installation process is different for all three platforms and unfortunately there is a certain amount of manual work to install the software.

STEP 1: Go to <https://www.arduino.cc/en/software>.

 **Arduino IDE 1.8.13**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE
Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

[Release Notes](#)
[Checksums \(sha512\)](#)

The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

STEP2 : Download the development software that is compatible with the operating system of your computer. Take Windows as an example here.

DOWNLOAD OPTIONS

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

[Release Notes](#)
[Checksums \(sha512\)](#)

1.2 Click Windows Installer.

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



1.3 Click JUST DOWNLOAD

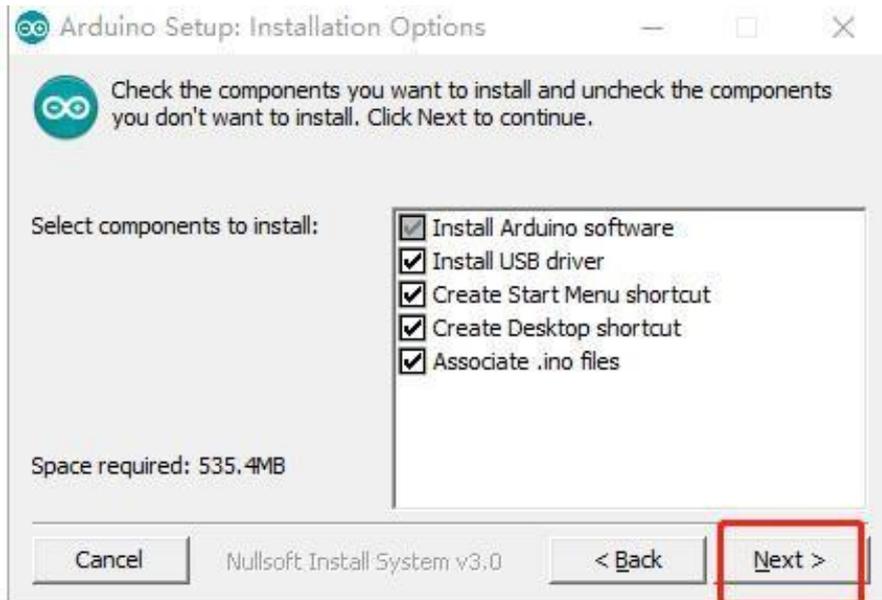
Also version 1.8.9 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.

Installing Arduino(Windows)

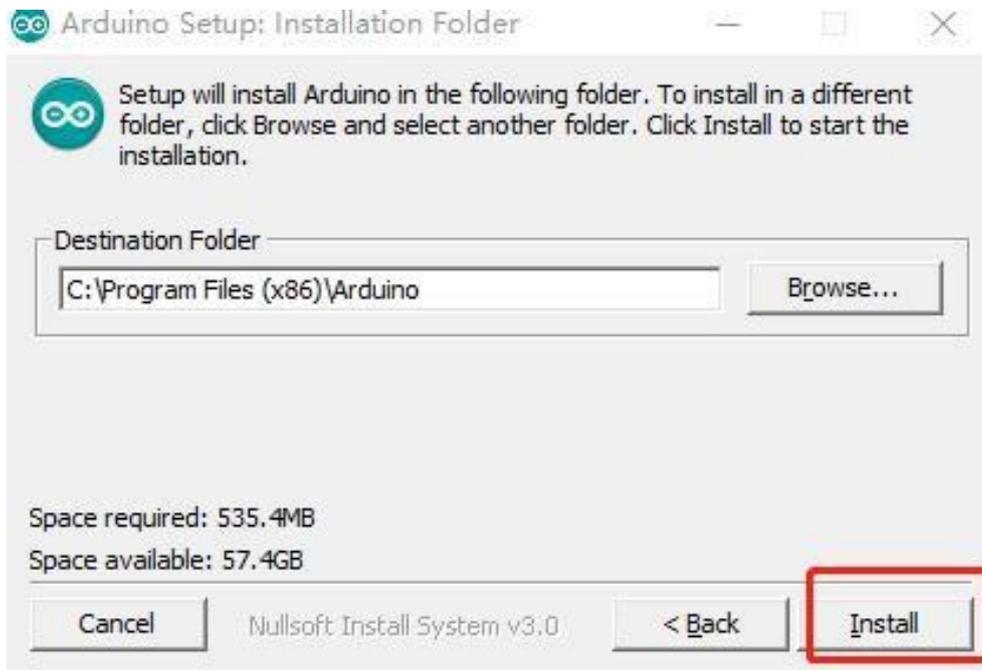
 arduino-1.8.13-windows



Click I Agree to see the following interface

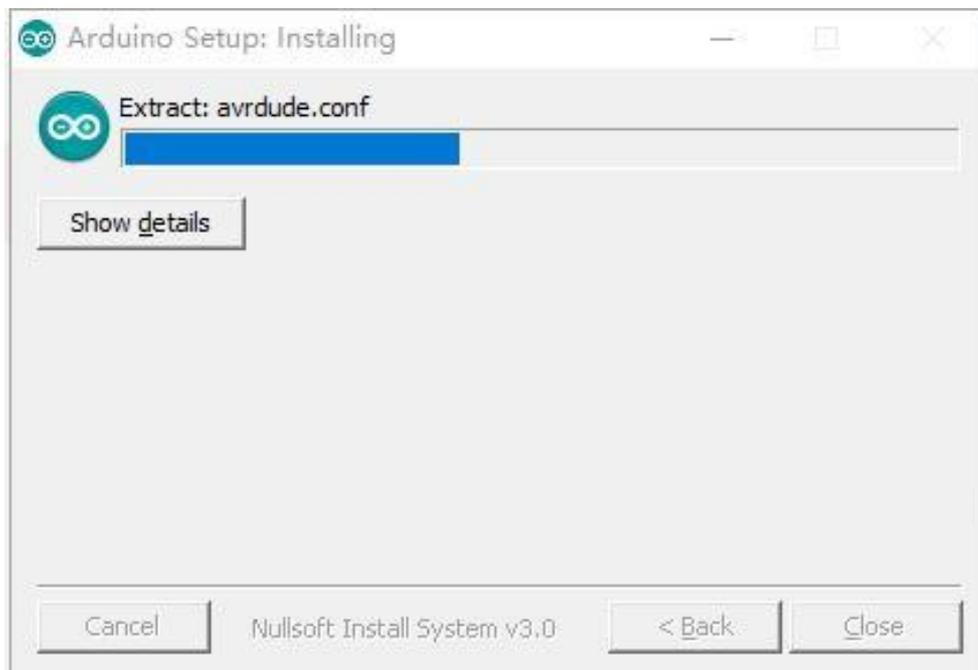


Click Next



You can press Browse...to choose an installation path or directly type in the directory you want.

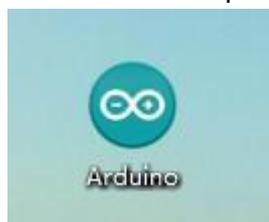
1.4 Click Install to initiate installation



Finally, the following interface appears, click Install to finish the installation.



Next, the following icon appears on the desktop



Double-click to enter the desired development environment



```
sketch_nov13a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_nov13a
1
2 void setup()
3 {
4   pinMode(13, OUTPUT);
5 }
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(500);
10  digitalWrite(13, LOW);
11  delay(500);
12 }
```

1.5 Installing Arduino(Mac OS X)

Download and Unzip the zip file,double click the Arduino.app to enter Arduino IDE;the system will ask you to install Java runtime library if you don't have it in your computer.Once the installation is complete you can run the Arduino IDE.

1.6 Installing Arduino(Linux)

You will have to use the make install command.If you are using the Ubuntu system,it is recommended to install Arduino IDE from the software center of Ubuntu.

2 Add Libraries

2.1 Installing Additional Arduino Libraries

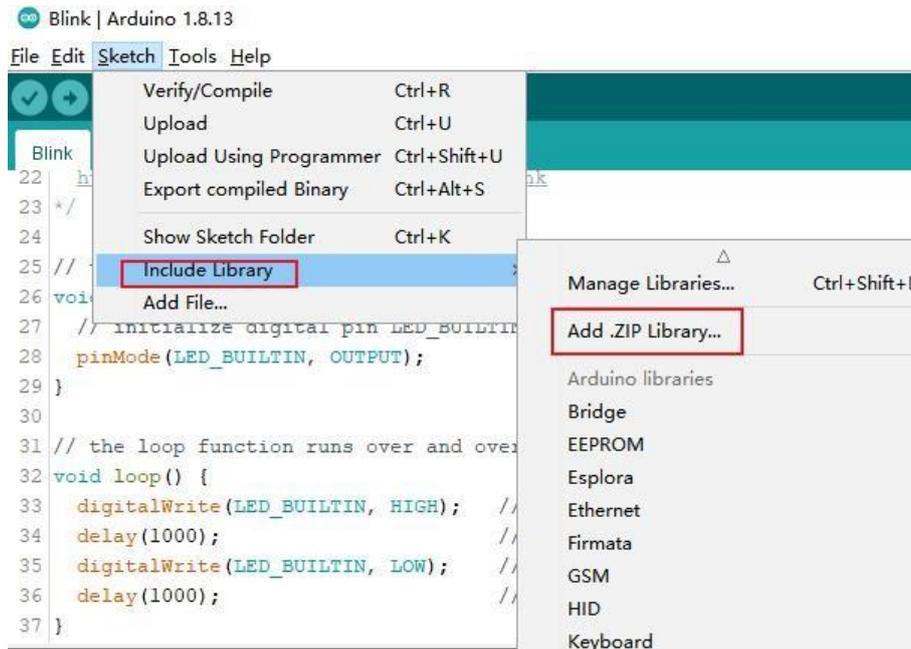
Once you are comfortable with the Arduino software and using the built-in functions, you may want to extend the ability of your Arduino with additional libraries.

2.2 What are Libraries

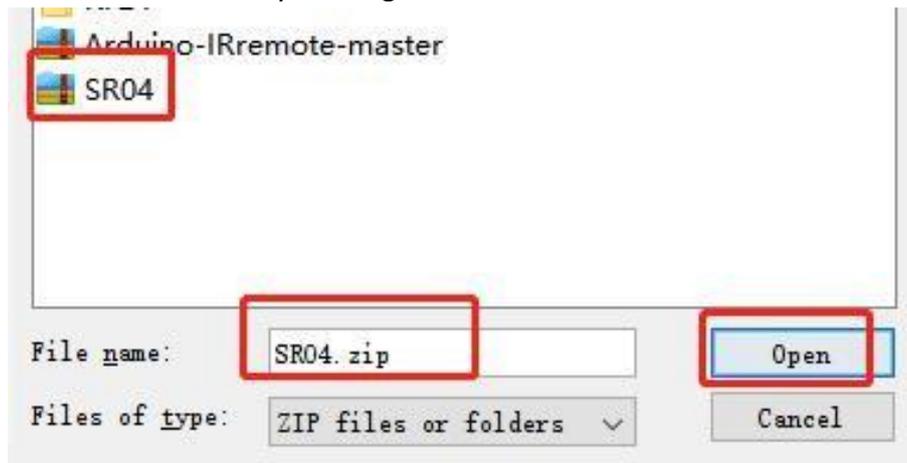
Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are listed in the reference. To use the additional libraries, you will need to install them.

2.3 How to Install a Library

Using the Library Manager. To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.8.13). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.



and then Include Library>Manage Libraries.



2.4 Example:SR04

Open arduino software-project-load library-add a.zip library

2.5 Add method two:

Copy the library folder to the Libraries folder in the Arduino installation directory. Restart Arduino and the added library will take effect.

3 Ultrasonic transducer

3.1 Introduction

Ultrasonic waves are produced by mechanical vibration and can travel at different speeds in different media.

Ultrasonic wave is often used to measure distance because of its strong directivity, slow energy loss and long propagation distance in the medium.

Such as rangefinder and level measuring instrument can be realized through ultrasonic.

Ultrasonic ranging is a non-contact detection method.

Compared with other methods, such as electromagnetic or optical methods, it is not affected by light, the color of the object under test, etc.

For the measured objects in the dark, dust, smoke, electromagnetic interference, toxic and other harsh environment has a certain adaptability.

Therefore, it is widely used in liquid level measurement, manipulator control, vehicle automatic navigation, object recognition and other aspects.

Especially applied to air ranging, because of the slow wave velocity in the air, the echo signal contained along the propagation direction of the structure information is easy to detect, has a very high resolution, so its accuracy is higher than other methods;

And ultrasonic sensor has the advantages of simple structure, small size, Signal processing reliability and other characteristics.

The use of ultrasonic detection is often more rapid, convenient, simple calculation, easy to achieve real-time control, and in terms of measurement accuracy can meet the requirements of industrial practical.

There are many methods of ultrasonic ranging, the principle of the system in the ultrasonic measurement is: detect the ultrasonic from the ultrasonic transmitter, the transmission time to the receiver through the gas medium.

Multiply this time by the speed of sound in the gas to give the distance the sound travels.

Ultrasonic transmitter to a certain direction of the transmission of ultrasonic wave, at the same time the MCU began to time, ultrasonic wave in the air transmission, encountered obstacles on the way to return immediately, ultrasonic receiver received the reflected wave immediately stop timing.

According to the time T recorded by the timer, the distance (s) between the launching point and the obstacle can be calculated.

formula: = —

Four factors limit the maximum measurable distance of an ultrasonic system: the amplitude of the ultrasonic wave, the texture of the reflector, the Angle between the reflected and incident sound waves, and the sensitivity of the receiving transducer.

The ability of the receiving transducer to receive the acoustic pulse directly will determine the minimum measurable distance.

3.2 Obstacle avoidance principle

The trigger signal input end (TRIG) will input a high-level signal of more than 10 microseconds, and the ultrasonic transmitter will automatically send 8 40Hz square waves upon receiving the signal. At the same time, the timer will be started. When the sensor receives the echo, the timing will be stopped and the echo signal will be output.

According to the time interval, the distance can be calculated by the formula:

distance =(high level time * sound velocity)/2, and the speed of sound propagation in the air is 340m/s.

3.3 Acquisition of measurement data

The measured distance is obtained through the SR04() function.

```
int SR04(int Trig,int Echo)
{
    float cm = 0;
    digitalWrite(Trig,LOW);    //Trig Set to low level
    delayMicroseconds(2);     //Wait 2 microseconds
    digitalWrite(Trig,HIGH);  //Trig Set to high level
    delayMicroseconds(15);    //Wait 15 microseconds
    digitalWrite(Trig,LOW);   //Trig Set to low level
    cm = pulseIn(Echo,HIGH)/58.8; //Convert the ranging time to CM cm
    = (int(cm * 100.0))/100.0; //Leave 2 as a decimal
    //Character Distance displayed in serial port monitor window:
    Serial.print("distance:");
    Serial.print(cm);
    Serial.println("cm");
    return cm;    //Returns cm value ranging data
}
```

The measurement results can be viewed in the Aduino IDE by printing the information through the serial port.

```
distance = 42
distance = 43
dista
```

When the distance detected by ultrasonic wave is less than the set distance, the car will be judged to encounter obstacles, and then the obstacle avoidance program will be written to make the car turn backward or turn left to avoid obstacles, so as to realize the car's automatic driving around obstacles.

How to make car automatic obstacle avoidance more flexible?

Of course, the ultrasonic sensor is installed on the steering gear. Through the rotation of the steering gear, the distance between the left and the right and the middle of the car is measured. Comparing the three distances, the maximum value is obtained, and the car is driven to the maximum measured direction.

The direction of the car to avoid obstacles will be more accurate.

Car follow function using ultrasonic ranging principle, when in the set distance, the car will follow the front of the items driving, when less than the set range, it will reverse.

Trigger the left infrared obstacle avoidance sensor, the car and servo motor will turn left, trigger the right infrared obstacle avoidance sensor, the car and servo motor will turn right, to follow.

Part of the procedure is as follows:

```

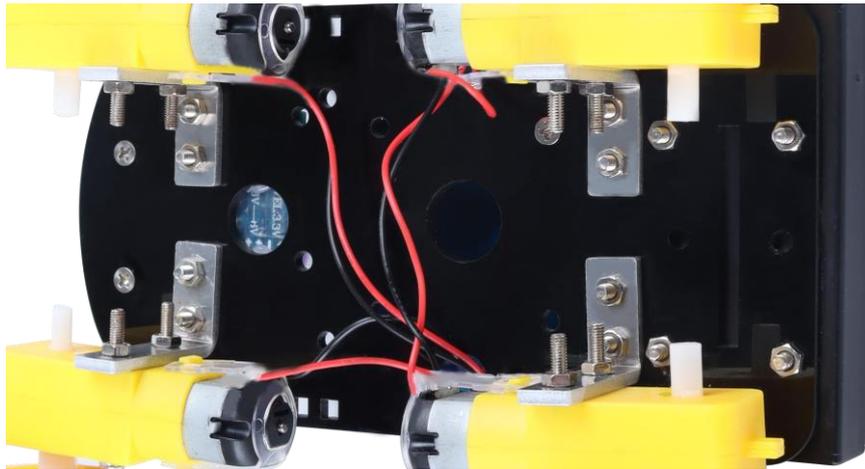
void AvoidingObstacles()
{
  // If the distance is greater than 20cm or
  //bluetooth receives commands equal to F
  if((distance > 20 ))
  {
    delay(10);
    // Again determine if the distance is really greater than 20cm
    if(distance > 20)
    {
      // Call forward function
      Motor(Forward, Speed1, Speed2, Speed3, Speed4);
    }
    // Otherwise the distance is less than 20
  else
  {
    Motor(Back, Speed1, Speed2, Speed3, Speed4);
    delay(500);
    // Turn left to change the direction of the car
    Motor(Left, Speed1, Speed2, Speed3, Speed4);
    delay(500);
  }
}

```

4 car assembly

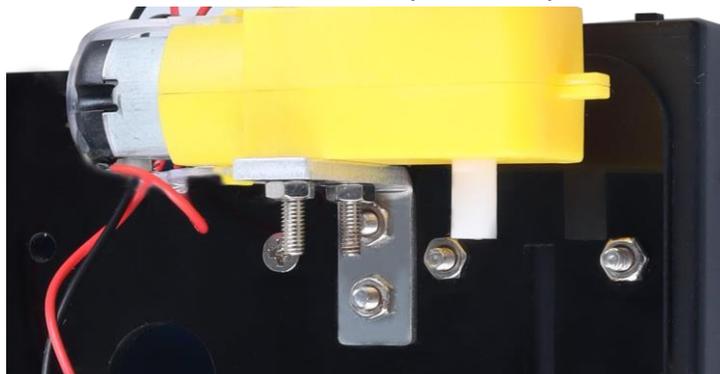
4.1 motor wiring

The motor in the kit is not connected, so the prepared power cord needs to be welded to the second end of the motor, as shown in the figure ;



4.2 Motor fixing screw installation

Fix the welded motor on the metal trolley chassis by screws.



Fix the other three motors in the same way.

4.3 wheels installed

Wheel installation, pay attention to distinguish between the left and right wheels;

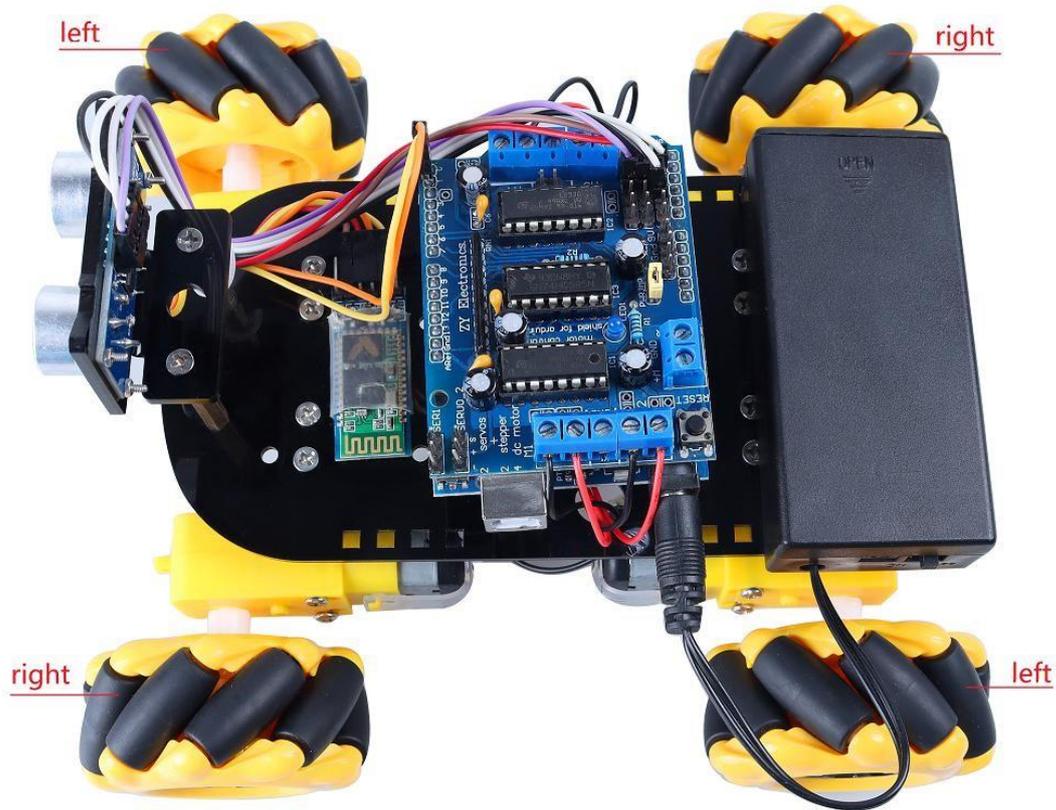


The motor shaft connects the wheels.

Then place a screw in the middle of the wheel and tighten it with a screwdriver.

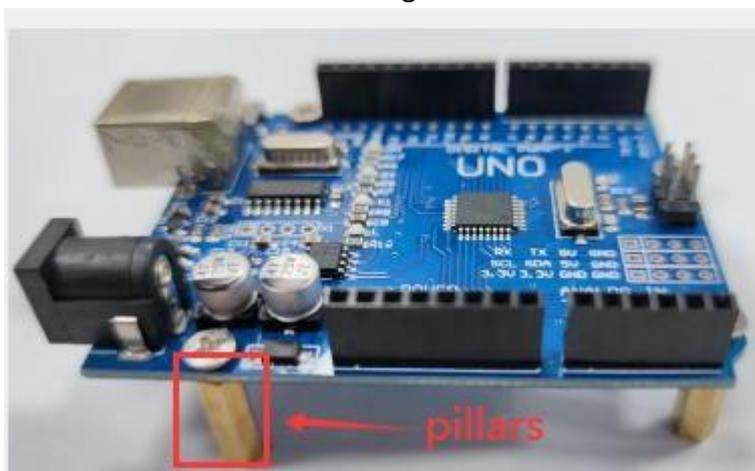


Wheel installation completed:



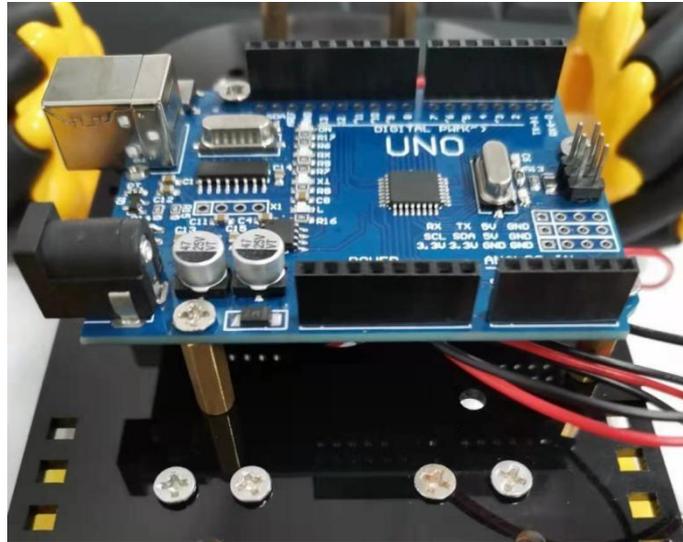
4.5 Fixation of control panel

Install 4 copper columns for fixing the main control board;First of all, put a M3*8 screw into the positioning hole of the main control board, then take a copper column and connect it with the screw and tighten it.

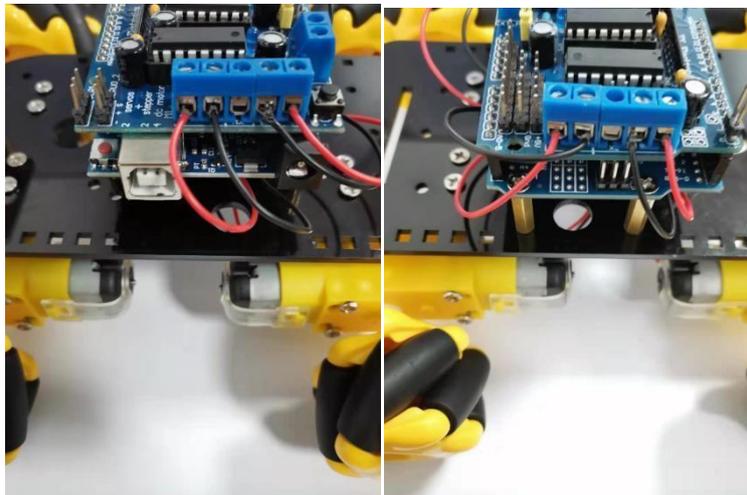


The installation method is the same for the other three copper columns.

Then align the main board with the holes inside the copper column and the holes in the chassis of the trolley. Screw the screws in the copper column and fix the main control board well.

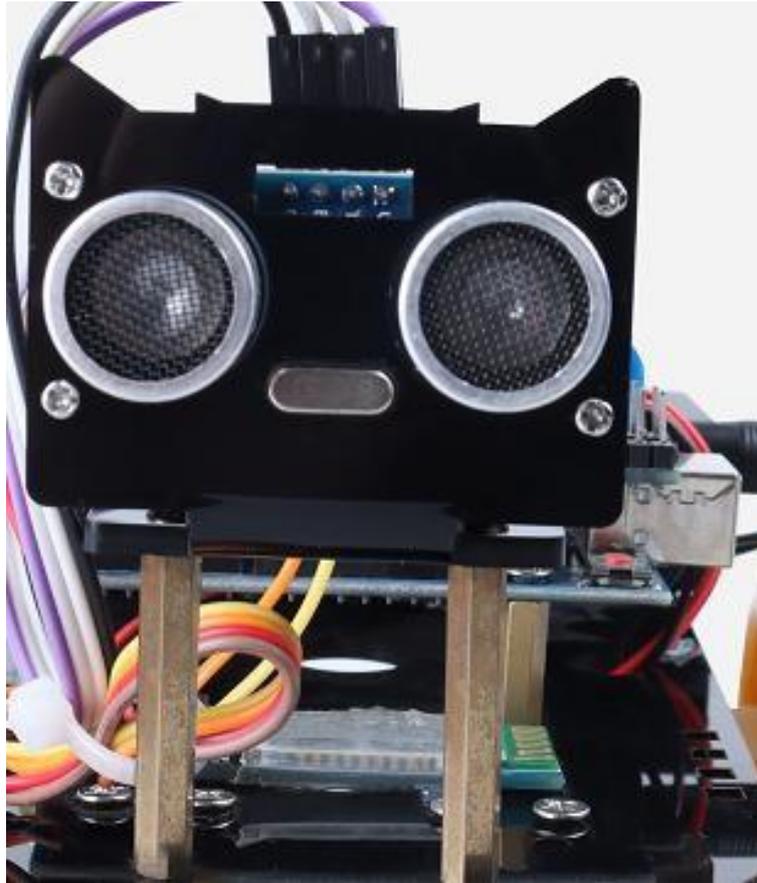


Then stack the L293D motor drive board on the control board, and pay attention to distinguish the direction to avoid reverse insertion. The key of the motor drive board is in the same direction as the power interface of the control board, as shown in the figure:



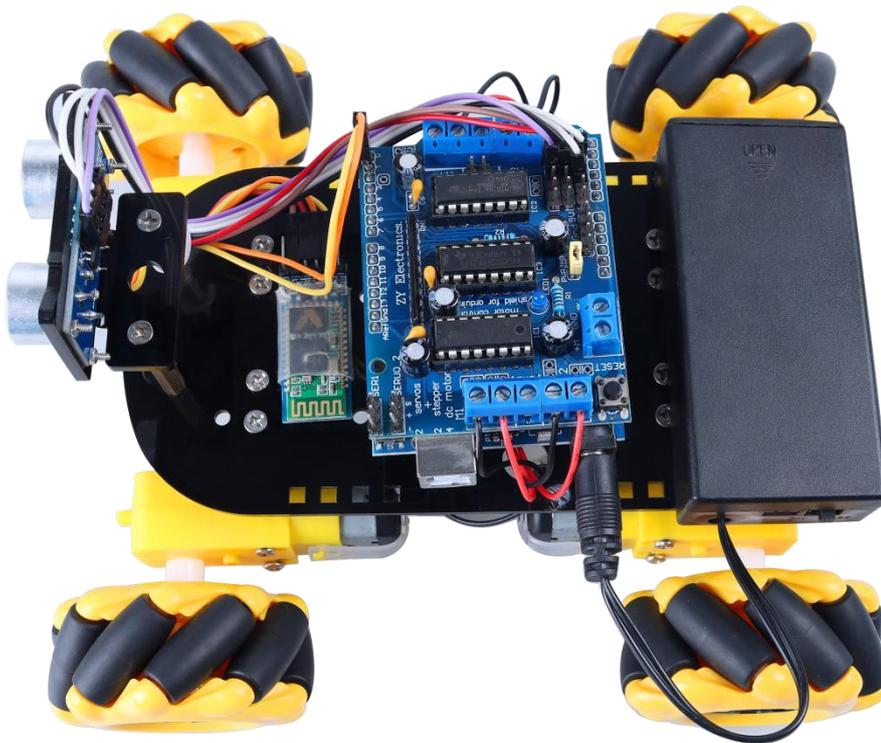
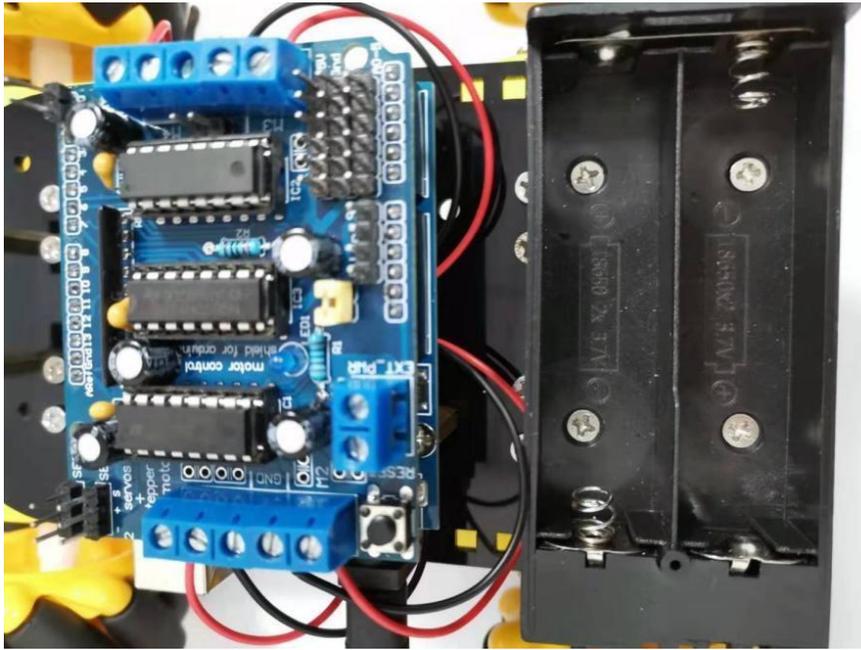
4.6 Installing ultrasonic sensors

Fix the ultrasonic security sensor on the bracket, as shown in the figure:



4.7 Install the battery box

Hit the lid of the battery box, one of which has four screw holes, and put the battery box in the parking space. After the screw hole of the box is aimed at the hole of the chassis, put the screw into the box to pass through the chassis, and then screw the nut into the screw, as shown in the figure.



5 Wiring part

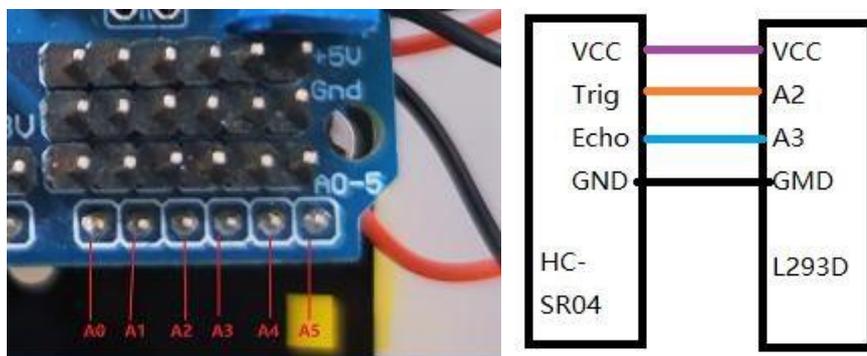
5.1 Ultrasonic sensor wiring

Ultrasonic sensor(HC-SR04) is connected. In the program, the Trig definition of

the sensor is connected to the A2 pin of the main control board, Echo and A3 are connected, while VCC is connected to the 5V pin of the main control board, and GND is connected to GND.

```
13 //Define the pin of ultrasonic obstacle avoidance sensor
14 //A2 is the pin Trig connected to the ultrasonic sensor
15 //A3 is the pin Echo connected to the ultrasonic sensor
16 const int Trig = A2;
17 const int Echo = A3;
```

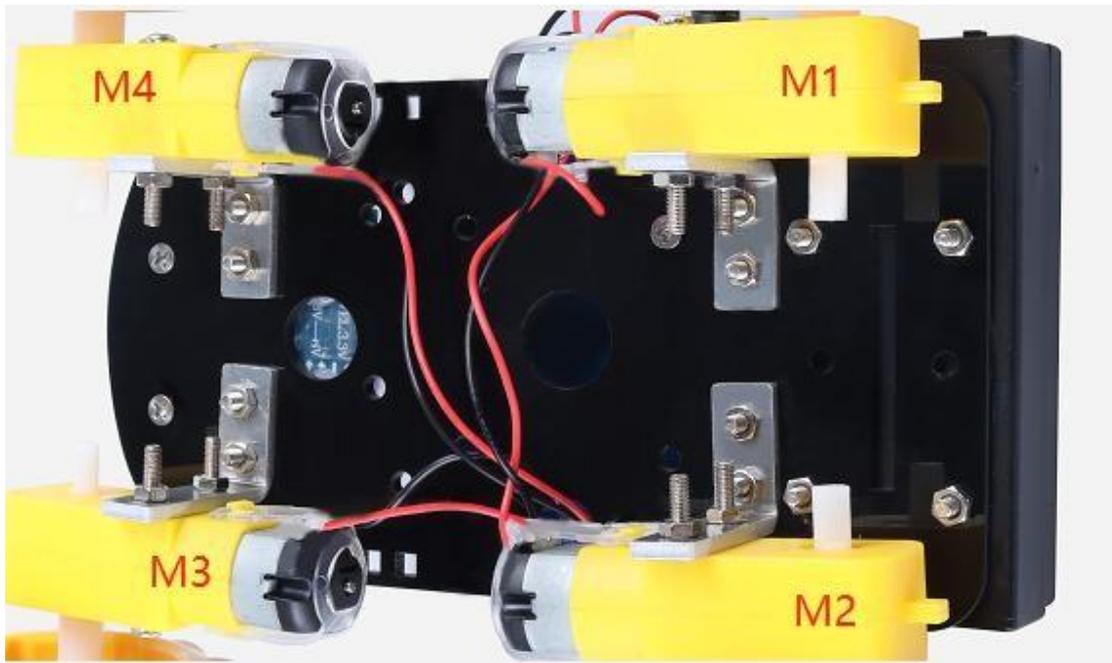
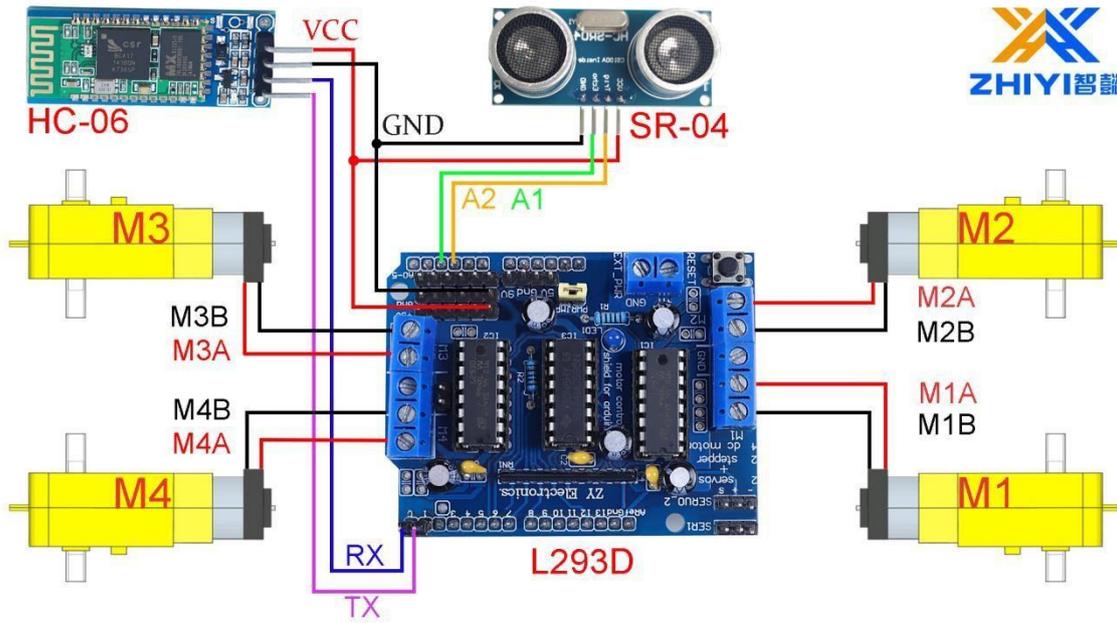
Connect the ultrasonic sensor to the main control board through the dupont line.



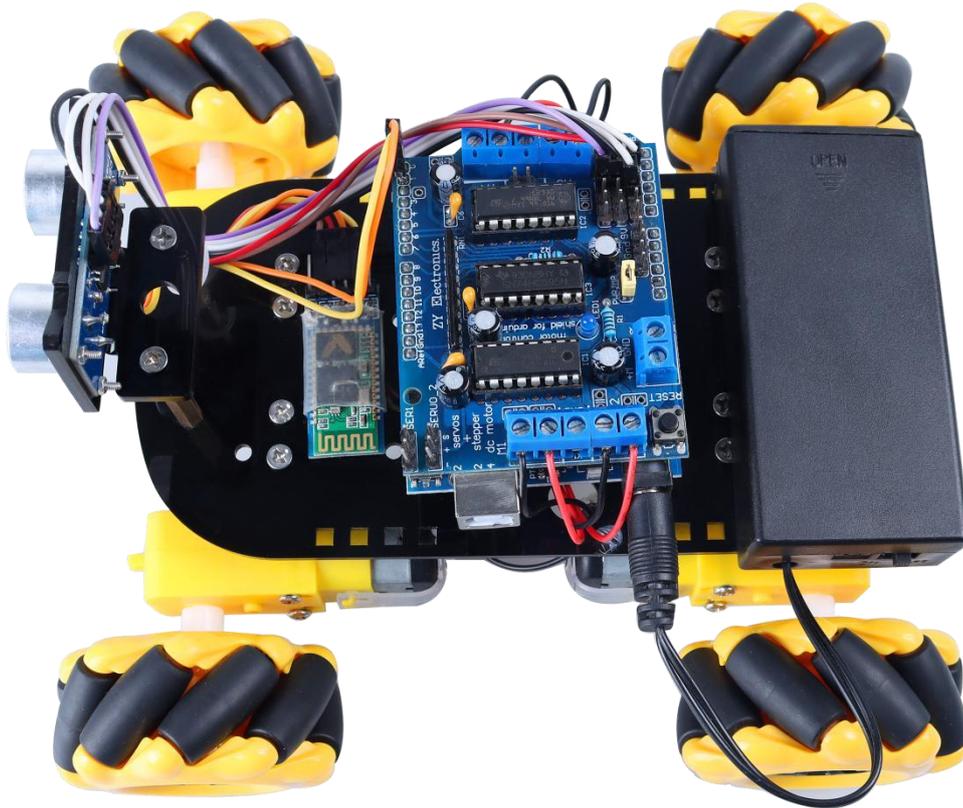
5.2 Bluetooth module wiring

Bluetooth module (HC-05) wiring is respectively VCC, GND, TX and RX, which are connected to the motherboard respectively. TX and RX need cross connection.

Arduino's TX is connected to the RX of the Bluetooth module; RX of Arduino is connected to TX of Bluetooth module. As shown in the figure:



The motor wiring has been completed here



6 program analysis

6.1 Motor drive

Motor 1 is connected to M1 terminal, motor 2 to M2 terminal, motor 3 to M3 terminal, motor 4 to M4 terminal.

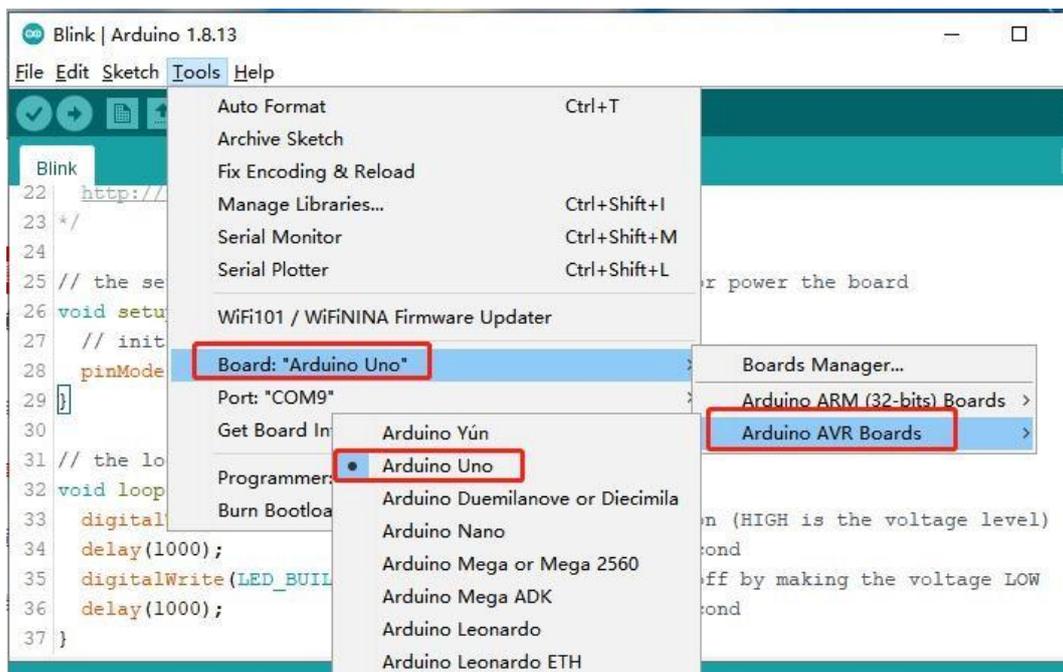
Each terminal port has two terminals that output high and low levels to enable forward and reverse motor rotation.

Power supply mode: Two 18650 batteries of 3.7V are connected in series in the battery box to obtain a voltage of 7.4V. The power output of the battery box is connected to the power input end of the Arduino main control board, which is combined with the arduino main control board and the stacking mode.

Connect the wires of the four motors to each M port of the L293D motor drive board, and download the program of the Arduino motherboard through the USB data wire, so that the car can run;

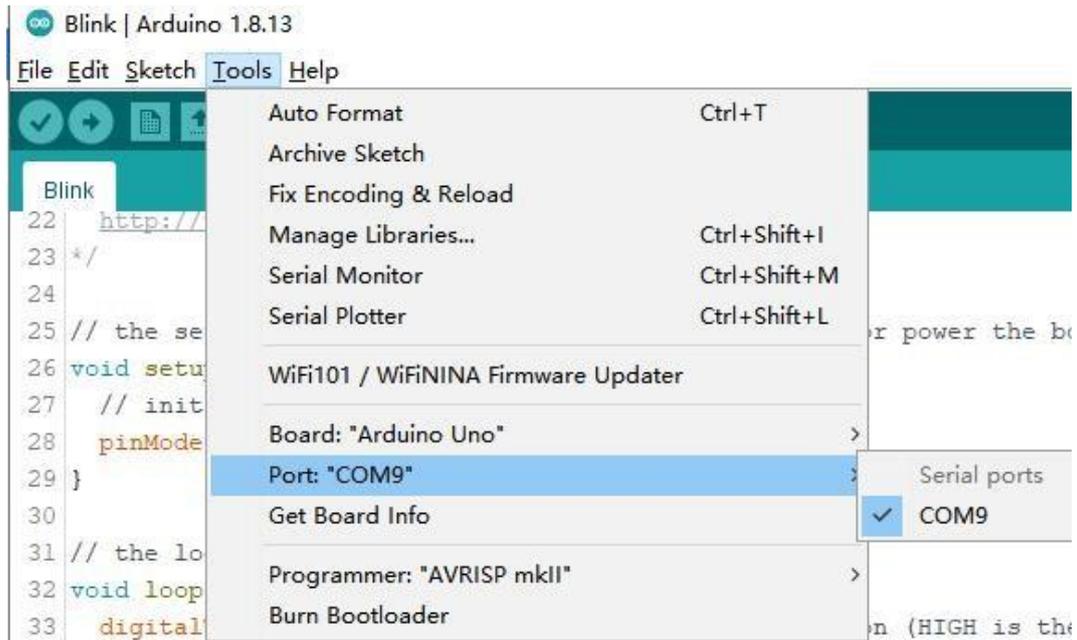
6.2 Upload code

Select tools from the compiler, and select the development board: "Arduino Uno Option" from the following options, and then select the control board model: Arduino Uno;As shown in the figure below.



The upload code also needs to select the board port number. Our computer displays the port number of arduino as: "COM9 Arduio / Genuino Uno", you may be the other serial port number.

Note: A correct COM port should be COMX (arduino) XXX), which is certified by the standard.

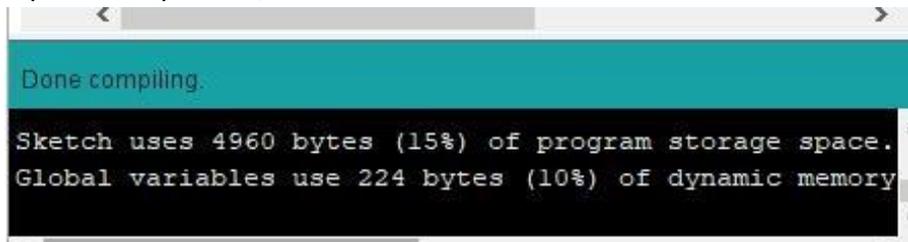


Verify the written program by selecting the verification button in the red circle

below;



Complete compilation, no errors.



Select the "arrow to the right" upload button to upload the program to the Arduino Uno motherboard;

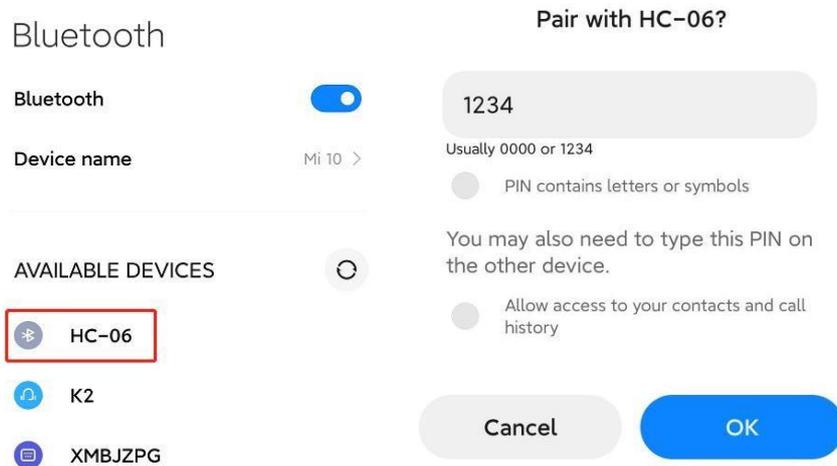
```

File Edit Sketch Tools Help
sketch_dec02a $
1
2 //Configure THE PWM control pin
3 const int PWM2A = 11; //M1 motor
4 const int PWM2B = 3; //M2 motor
5 const int PWM0A = 6; //M3 motor
6 const int PWM0B = 5; //M4 motor
Done uploading.
Sketch uses 4960 bytes (15%) of program storage space. M
Global variables use 224 bytes (10%) of dynamic memory,

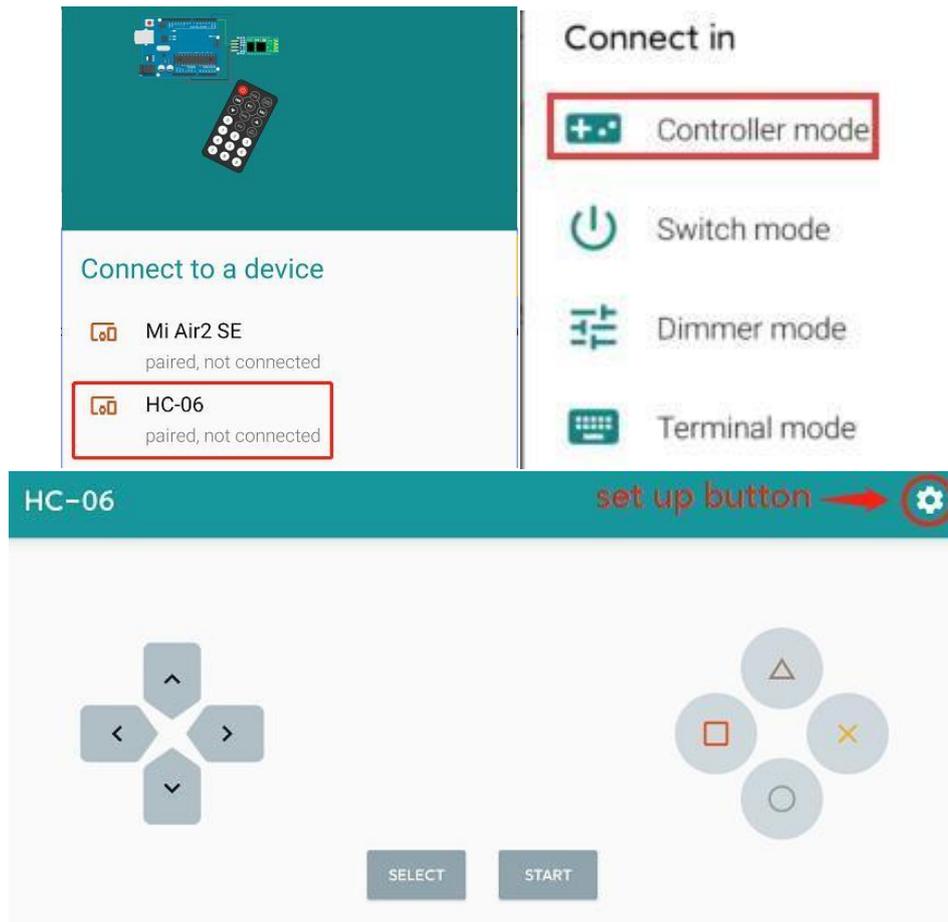
```

6.3 Bluetooth remote control

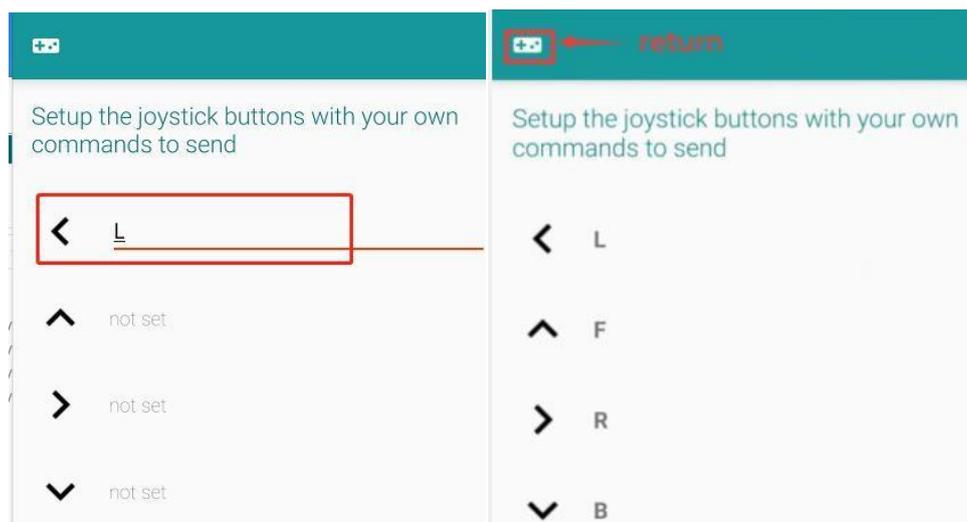
Equipped with bluetooth serial port assistant for remote control of car;First, turn on bluetooth and search for Bluetooth devices to find hC-05 connection. The default connection pairing password is 1234.



Open bluetooth Assistant and find the bluetooth module name hC-05 just configured. After successful connection, a remote control main interface will appear.



Command Settings, such as L for trolley moving to the left. Select the icon in the upper left corner to return to the control interface.



Just configured L,F,R,B and other commands, consistent with the program can be successful remote control car, otherwise the remote control fails;

```

//If Bluetooth receives the string F,
//the dolly moves forward and enables obstacle avoidance
if('F' == cmd)  //
{
    AvoidingObstacles();//
}
//The ultrasonic obstacle avoidance function is called to
//realize the obstacle avoidance function
else if('B' == cmd)
{
    Motor(Back, Speed1, Speed2, Speed3, Speed4);
}

```

Press the forward button and the car will move forward. Release it and it will still move forward. Control the car in other directions is the same operation.



6.5 Command function corresponding table

command	function
F	forward
B	back
L	Turn left
R	Turn right
S	stop
A	Obstacle avoidance
C	Left rotation

D	Right rotation
---	-------------------

6.6 Virtual serial port

Through the virtual window communication, the Bluetooth module is connected to the pin of the virtual serial port. When downloading programs, there is no need to disconnect the data line of the Bluetooth module, which makes our operation more convenient. The use method is the same as the default serial port.

Virtual window communication routines:

```
#include <SoftwareSerial.h>

// Serial is a virtual port, Rx is port A0, Tx is port
A1 SoftwareSerial serial(A0, A1); void setup()
{
  // Initialize the virtual serial port serial.begin(9600);
  // Initialize the default serial port
  Serial.begin(9600);
}

char  value;

void loop()
{
  // The virtual serial port is used in much the same way as the default serial port
  if (serial.available())
  {
    value = serial.read();
    Serial.print(value);
  }
}
```

7 code analysis

7.1 Pin configuration

```
//Configure THE PWM control pin
const int PWM2A = 11;    //M1 motor
const int PWM2B = 3;    //M2 motor
const int PWM0A = 6;    //M3 motor
const int PWM0B = 5;    //M4 motor

const int DIR_CLK = 4;  //Data input clock line
const int DIR_EN = 7;  //Equip the L293D enabling pins
const int DATA = 8;   // cable
const int DIR_LATCH = 12; //Output memory latch clock

//Define the pin of ultrasonic obstacle avoidance sensor
//A2 is the pin Trig connected to the ultrasonic sensor
//A3 is the pin Echo connected to the ultrasonic sensor
const int Trig = A2;
const int Echo = A3;
```

Define the motion state parameters of the trolley

```
//Define motion state
const int Forward = 39; //39 is stored in the Forward variable
const int Back = 216;  //216 is stored in the Back variable
const int Left = 116;  //116 is stored in the Left variable
const int Right = 139; //The right amount of change 139
const int Stop = 0;    //Parking variable
```

Define PWM speed control value. The default speed control value is within the range of 1~255.

Under normal conditions, try not to set the speed too low to avoid motor rotation;

```
26 //Set the default speed between 1 and 255
27 int Speed1 = 180; //PWM0B -M3
28 int Speed2 = 180; //PWM0A -M4
29 int Speed3 = 180; //PWM2A -M3
30 int Speed4 = 180; //PWM2B -M4
```

Void setup() function will only execute once during the program run, and only configure once for the baud rate of serial port. Baud rate is set to 9600, which is the same as the Baud rate of bluetooth module and serial port monitor, so that data can be transmitted to each other.

```
void setup()
{
    Serial.begin(9600); //Set the serial port baud rate 9600

    //Configure for pin mode
    pinMode(DIR_CLK, OUTPUT);
    pinMode(DATA, OUTPUT);
    pinMode(DIR_EN, OUTPUT);
    pinMode(DIR_LATCH, OUTPUT);
    pinMode(PWM0B, OUTPUT);
    pinMode(PWM0A, OUTPUT);
    pinMode(PWM2A, OUTPUT);
    pinMode(PWM2B, OUTPUT);

    //The Trig pin connected to the ultrasound is set to output mode
    //The Echo pin connected to the ultrasound is set to input mode
    pinMode(Trig, OUTPUT);
    pinMode(Echo, INPUT);

    void Motor(int Dri, int Speed1, int Speed2, int Speed3, int Speed4);
    int SR04(int Trig, int Echo);
}
```

The previously defined pin pattern is also configured here:

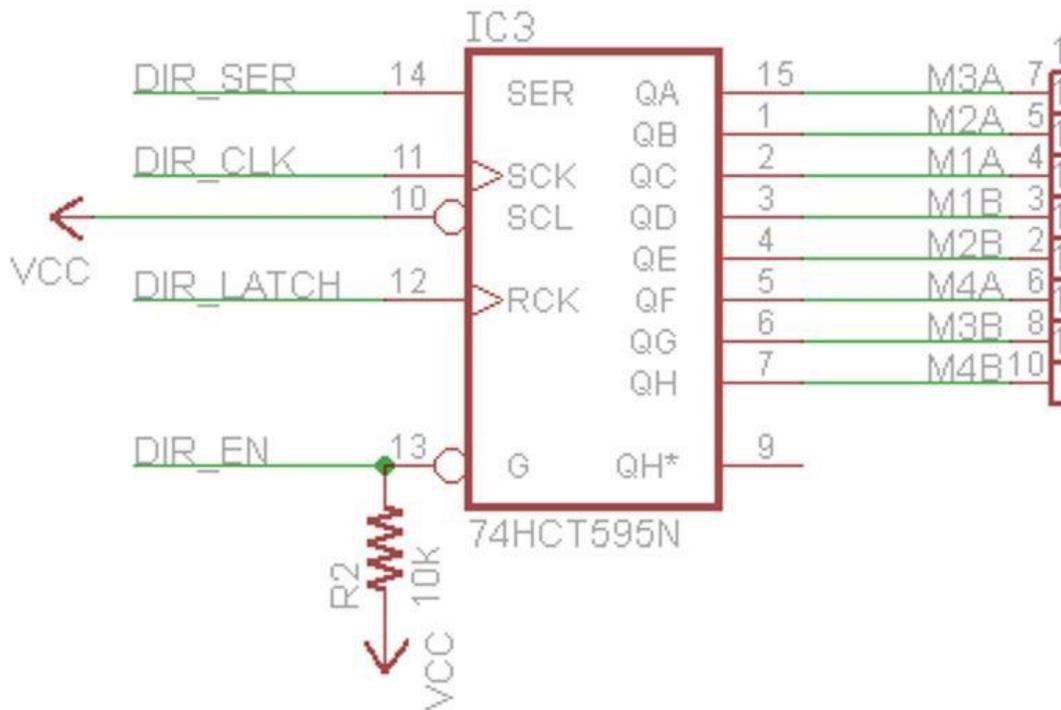
The pins of the Arduino UNO motherboard output signals to the motor driver module to control the movement of the motor, so the pins such as PWM0B, PWM0A, PWM1B and PWM1A need to be set as the output mode, and the pinMode() function is called.

74HC595 is a displacement cache with 8-bit serial input and parallel output: the parallel output is tri-state output.

At the rising edge of the SCK, single-row data is input from SDL to the internal 8-bit shift cache and output by Q7', while parallel output is input from the 8-bit shift cache to the 8-bit parallel output cache at the rising edge of the LCK.

When the control signal of the serial data input OE is low enabled, the output value of the parallel output terminal is equal to the value stored in the parallel output cache.

When OE is high potential, that is, the output is off, the parallel output remains in the high impedance state.



dir	value	M3A	M2A	M1A	M1B	M2B	M4A	M3B	M4B
forward	39	0	0	1	0	0	1	1	1
back	216	1	1	0	1	1	0	0	0
left	116	0	1	1	1	0	1	0	0
right	139	1	0	0	0	1	0	1	1
stop	0	0	0	0	0	0	0	0	0

Convert binary to decimal:

$$216 = 11011001 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0;$$

$$128=1 \times 2^7=2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2;$$

$$216=128+64+0+16+8+0+0+0;$$

Because the shiftOut (DATA, DIR_CLK MSBFIRST, Dir);

In the function, MSBFIRST is used, the first input data, the last output, for example, input 11010001, the output result is 10001011.

The enable pins of the L293D, 1A and 3A, are at high levels. 2A and 3A are low for the motor to start

input					output			
PWM	MA1	M1B	M2A	MAB	M1		M2	
EN1-4	1A	2A	3A	4A	1Y	2Y	3Y	4Y
H	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	L	L
H	H	H	L	L	H	H	L	L
H	L	L	H	H	L	L	H	H
L	H	L	H	L	L	L	L	L

(1) Machine drive function

(2) Void Motor()

(3) Input parameters of the function:

(4) Dir: motor motion direction parameter

(5) Speed1~Speed4 represents PWM speed control parameters of the motor, which have been previously introduced;

(6) AnalogWrite () function is called to speed PWM speed:

(7) Call shiftOut() function as a function of motor motion direction, change motion direction through Dir; Convert from decimal to binary;

(8) Dri values are explained as follows:

Forward:39 = 00100111

Back:216 = 11011000

Left:116= 01110100

Right:139 = 10001011

Stop:0 = 00000000

```
void Motor(int Dir,int Speed1,int Speed2,int Speed3,int Speed4)
{
    analogWrite(PWM2A,Speed1); //1 Motor PWM speed regulation
    analogWrite(PWM2B,Speed2); //2Motor PWM speed regulation
    analogWrite(PWM0A,Speed3); //3Motor PWM speed regulation
    analogWrite(PWM0B,Speed4); //4Motor PWM speed regulation

    //Set the low level and write the direction of motion to prepare
    digitalWrite(DIR_LATCH,LOW);
    //Dir motion direction value writes
    shiftOut(DATA,DIR_CLK,MSBFIRST,Dir);
    //Set the high level and output the direction of motion
    digitalWrite(DIR_LATCH,HIGH);
}
```

7.2 Ultrasonic sensor ranging

The module automatically sends 40KHz sound waves and automatically detects the time required for the signal to return. When the ultrasonic ranging sensor pin "Trig" is high, it sends out a 40khz ultrasonic Sound wave. In order to ensure that sound waves are emitted for only 15-microseconds(us), the trig pin needs to be pulled down for a short time before sending. When a signal returns the module outputs a high level through the echo pin. The duration of the high level is the duration of time from the start of the ultrasonic wave transmission to return. Distance = (high level time * sound velocity (340m/s))/2 = (high level time / 58.8) cm/s The ranging function code is shown below:

```

int SRO4(int Trig,int Echo)
{
    float cm = 0;

    digitalWrite(Trig,LOW);    //Trig Set to low level
    delayMicroseconds(2);     //Wait 2 microseconds
    digitalWrite(Trig,HIGH);  //Trig Set to high level
    delayMicroseconds(15);    //Wait 15 microseconds
    digitalWrite(Trig,LOW);   //Trig Set to low level

    cm = pulseIn(Echo,HIGH)/58.8; //Convert the ranging time to CM
    cm = (int(cm * 100.0))/100.0; //Leave 2 as a decimal
    //Character Distance displayed in serial port monitor window:
    Serial.print("Distance:");
    Serial.print(cm);
    Serial.println("cm");

    return cm;    //Returns cm value ranging data
}

```

7.3 serial port communication

Serial.available () returns the number of characters currently remaining in the Serial port buffer.

In this program, through this function to determine whether the Serial port buffer data, when serial.available ();

Serial.read() means to extract and read a Byte data from the buffer of the Serial port. For example, in the Bluetooth Assistant APP, press the forward button, then serial.read () can read the sent data, judge it as "F", and then store F in the CMD variable to wait for the call.

```

void bluetooth()
{
  // Determine whether the received data is greater than 0
  if(Serial.available() > 0)
  {
    serialData = Serial.read(); // Receive function
    Serial.println(serialData);
    // If the serial port receives data as character F,
    //save F to CMD
    if ('F' == serialData ) cmd = 'F';
    // If the data received by the serial port is character B,
    //save F to CMD
    else if('B' == serialData ) cmd = 'B';
    // If the data received by the serial port is L,
    //save F to CMD
    else if('L' == serialData ) cmd = 'L';
    // If the data received by the serial port is character R,
    //save F to CMD
    else if('R' == serialData ) cmd = 'R';
    else if('S' == serialData ) cmd = 'S';
  }
}

```

If (' F '== CMD) means that if the CMD equals' F' condition is true, then execute the program at line 155, call the AvoidingObstacles() function, and perform the forward and ultrasonic avoidance program;



```

//If Bluetooth receives the string F,
//the dolly moves forward and enables obstacle avoidance
  if('F' == cmd)  //
  {
    AvoidingObstacles();//
  }
  //The ultrasonic obstacle avoidance function is called to
  //realize the obstacle avoidance function
  else if('B' == cmd)
  {
    Motor(Back, Speed1, Speed2, Speed3, Speed4);
  }
  // Bluetooth received string L, car left translation
  else if('L' == cmd)
  {
    Motor(Left, Speed1, Speed2, Speed3, Speed4);
  }
  // When bluetooth receives the string R, dolly panalizes to the
  else if('R' == cmd)

```

Call the Avoiding Obstacles() function

```

void AvoidingObstacles()
{
  // If the distance is greater than 20cm or
  //bluetooth receives commands equal to F
  if((distance > 20 ) || cmd == 'F')
  {
    delay(100);
    // Again determine if the distance is really greater than 20cm
    if(distance > 20)
    {
      // Call forward function
      Motor(Forward, Speed1, Speed2, Speed3, Speed4);
    }
    // Otherwise the distance is less than 20
    else
    {
      Motor(Back, Speed1, Speed2, Speed3, Speed4);
      delay(500);
      // Turn left to change the direction of the car
      Motor(106, Speed1, Speed2, Speed3, Speed4);
      delay(500);
    }
  }
}

```

7.4 Set speed

If the command '+' character is received and Speed1 is less than 245, the condition is true, execute the statement in curly braces, and modify the PWM speed variable Speed1+10;

Speed2=Speed1 means to assign the modified Speed1 value to Speed2, so the data of Speed2, Speed3 and Speed4 are the same as the data of Speed1, and the following PWM speed regulation is the same.

```
// Received string +, speed increases
else if( serialData == '+' && Speed1 < 245)
{
    Speed1 += 10;    //We're going to increase our velocity by 10
    Speed2 = Speed1;
    Speed3 = Speed1;
    Speed4 = Speed1;
    Serial.print("Speed = ");
    Serial.println(Speed1);
}

//When I receive a string -- the speed decreases
else if( serialData == '-' && Speed1 > 30)
{
    // The velocity decreases by 10 each time
    Speed1 -= 10;
    Speed2 = Speed1;
    Speed3 = Speed1;
    Speed4 = Speed1;
    Serial.print("Speed = ");
    Serial.println(Speed1);
}
```

7.5 Loop functions

```
void loop()
{
    distance = SR04(Trig,Echo); // Obtain ultrasonic distance

    bluetooH(); // Call bluetooth car control function
}
```

The functions of ultrasonic sensor ranging, Bluetooth receiving command, Bluetooth controlling the forward and backward of the car, PWM speed regulation, omnidirectional movement, obstacle avoidance, etc., are all executed in loop() function.