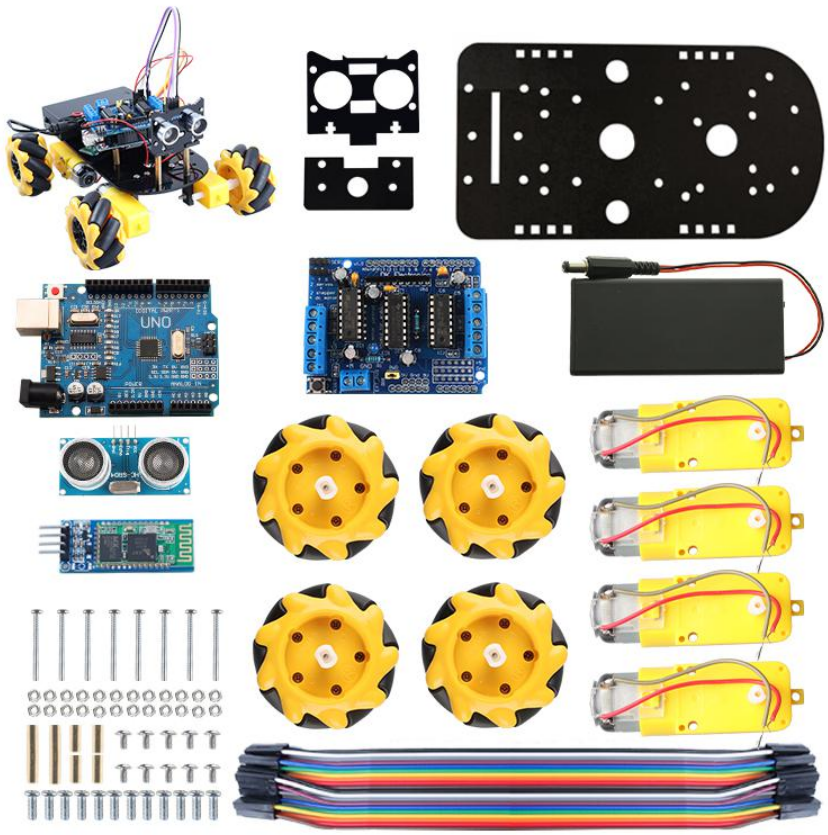
McNamum Wheel Car tutorial

Wiring and assemblymn

1. The basic components of smart car are as follows:



2. 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.

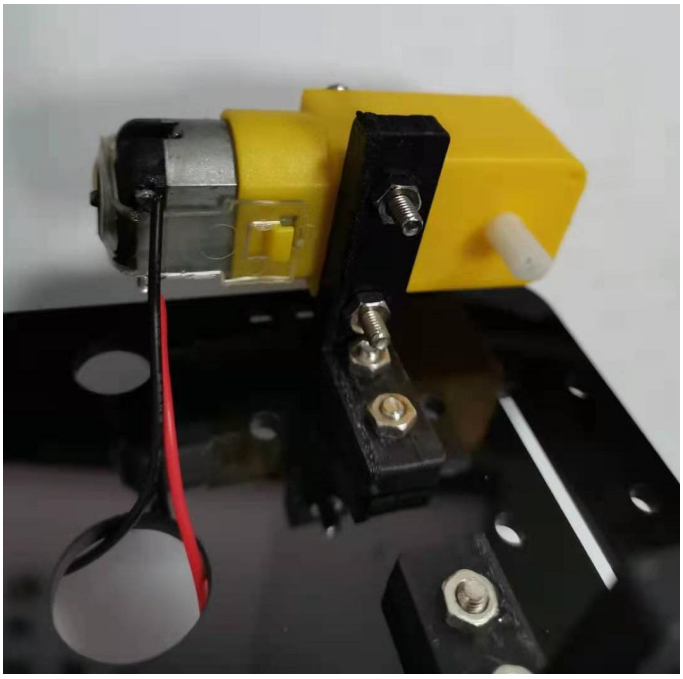
You can connect it to a PWM UNO pin and then use "analogWrite" to control the speed of the motor.

This pin is low activity, so we attach it to GND.

3、Fix the motor fixing bracket on the car chassis with M3 * 12 flat head screws

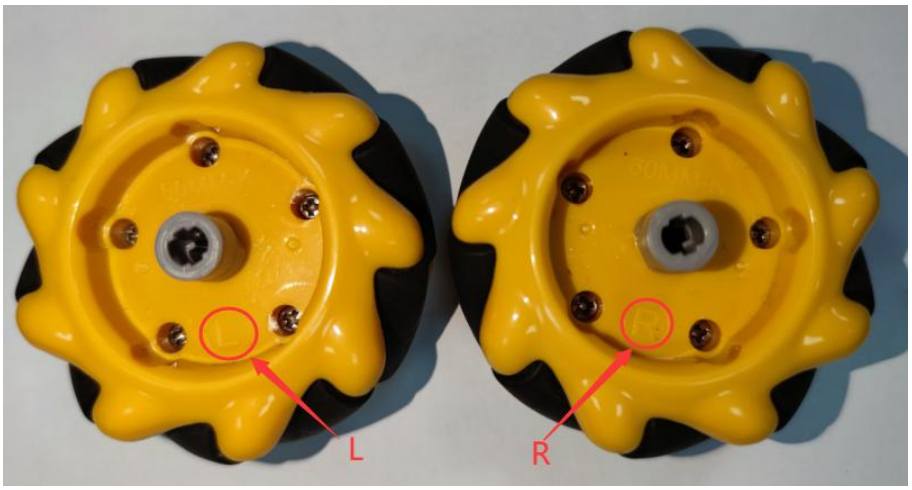
Secure the other three motor supports in the same way

4、 Fix the welded motor on the car chassis with screws.



Fix the other three motors in the same way.

5、Install McNamum wheels in AABB mode, and pay attention to distinguish left and right wheels;

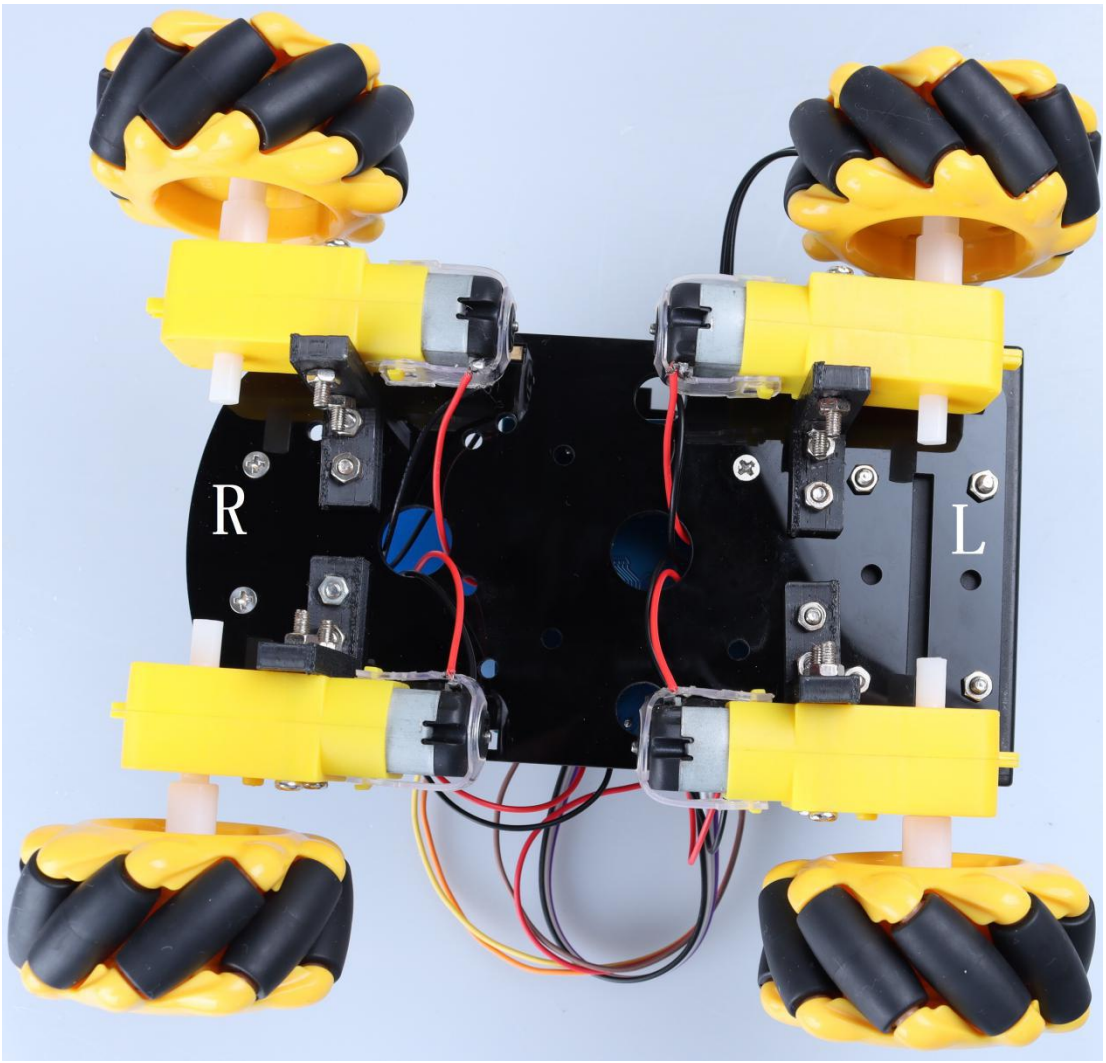


The motor shaft connects the wheels

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

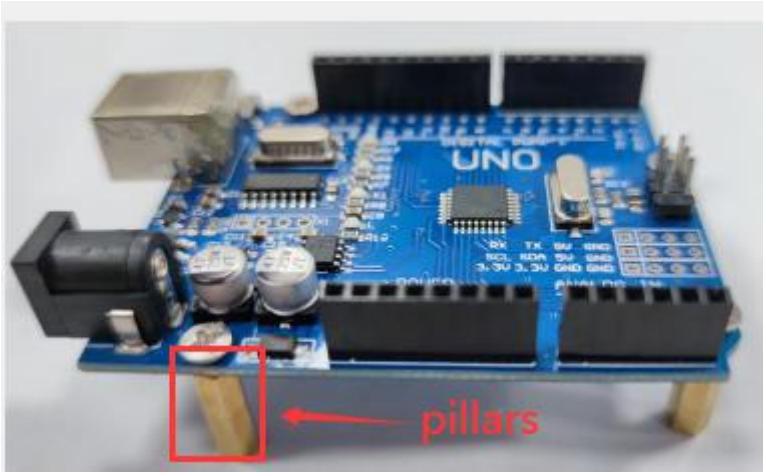
with long screws

Wheel installation completed:



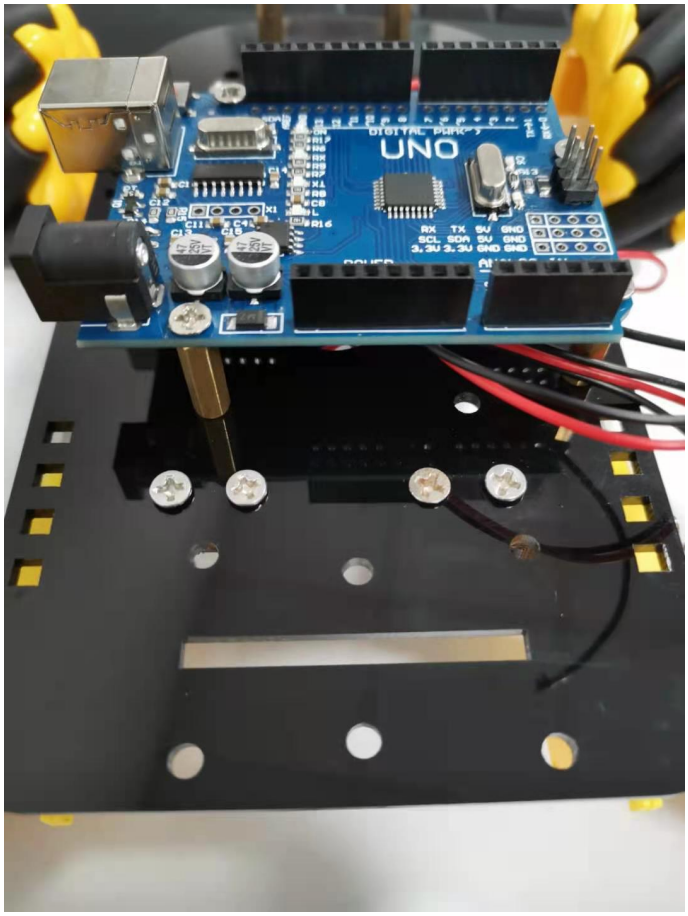
6、 Install 4 copper columns for fixing the main control board;

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.

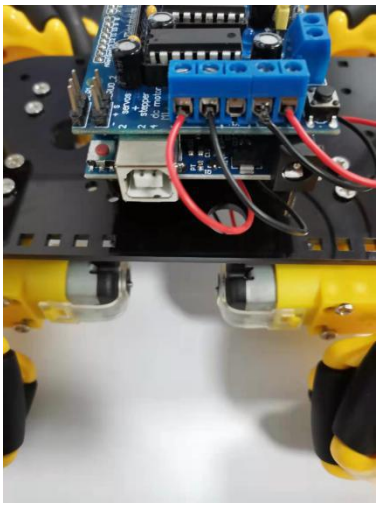


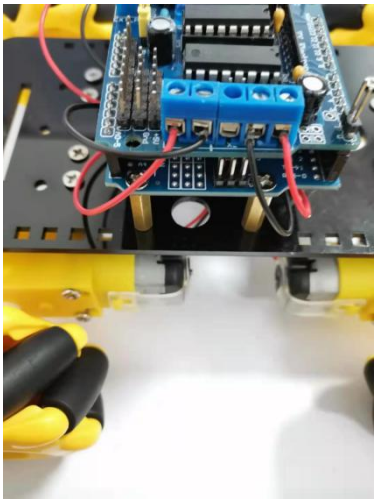
The installation method is the same for the other three copper columns.

7、There is only one positioning hole on the motherboard corresponding to the car bottom plate, and all the screws are used to connect the motherboard with the car bottom plate, so that users can play freely



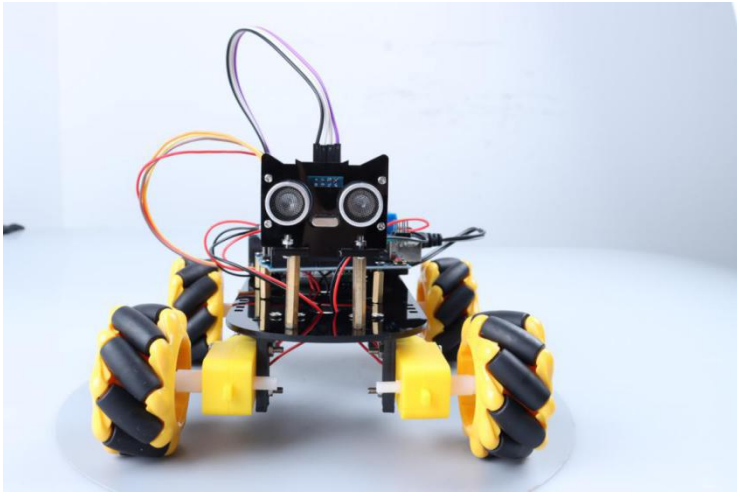
8、Then stack the L293D motor drive board on the control board, and pay attention to distinguish the direction to avoid reverse insertion. The key of the motor drive board is in the same direction as the power interface of the control board, as shown in the figure:



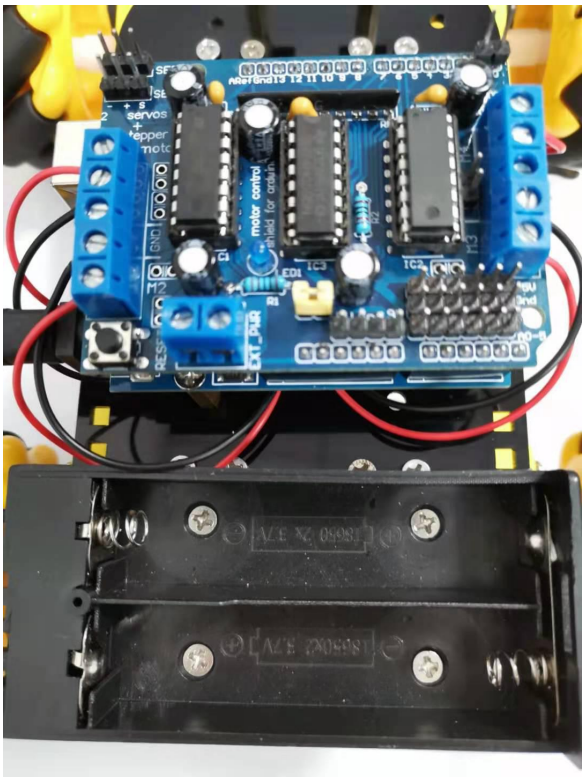


9、Fix the ultrasonic security sensor on the bracket, as shown in the figure:

Install the ultrasonic sensor and bracket to the metal chassis of the trolley.



10、Open the battery box, you will see four more screw holes, and fix the battery box in the corresponding four screw holes at the bottom of the car



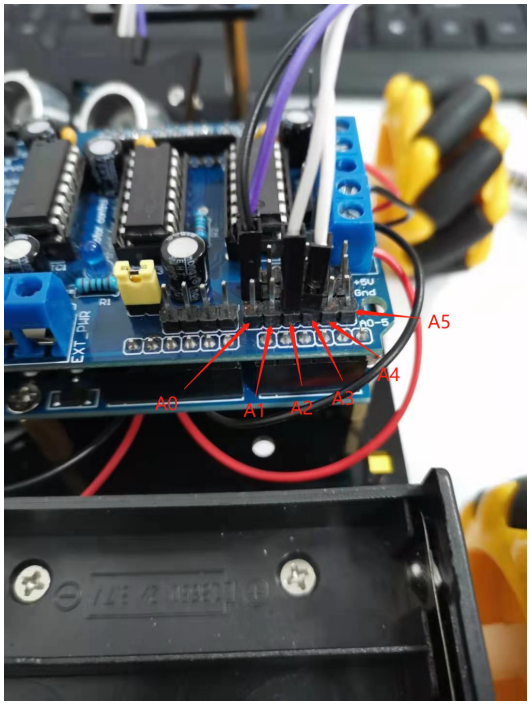
Wiring part

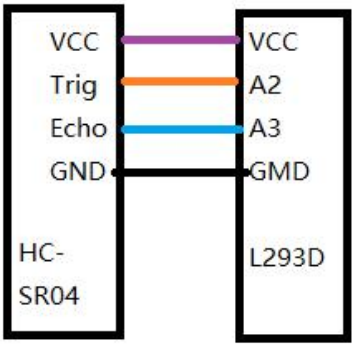
Pin wiring as defined by the program

1. 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



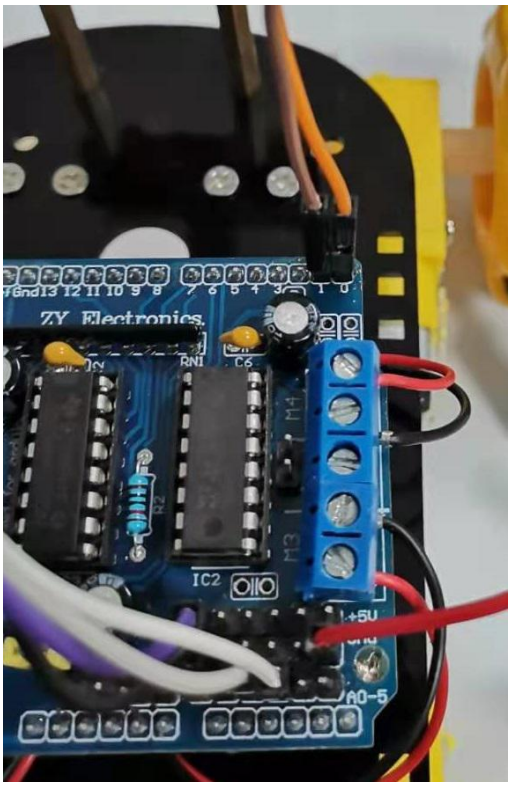


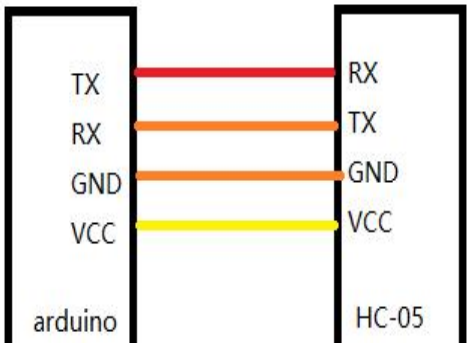
2. Bluetooth module (HC-05)wiring is respectively VCC,GND,TX and RX, which are connected to the motherboard respectively. TX and RX need cross connection.

Arduino's TX is connected to the RX of the Bluetooth module;

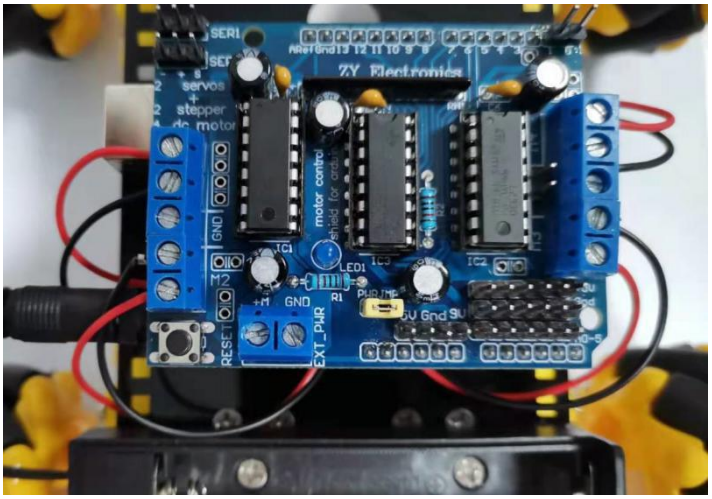
RX of Arduino is connected to TX of Bluetooth module.

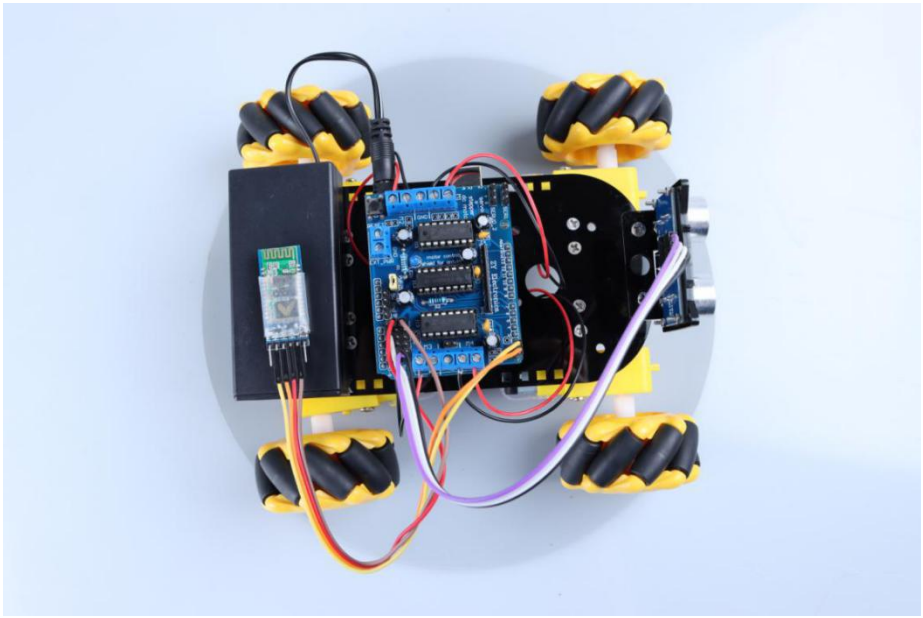
As shown in the figure:

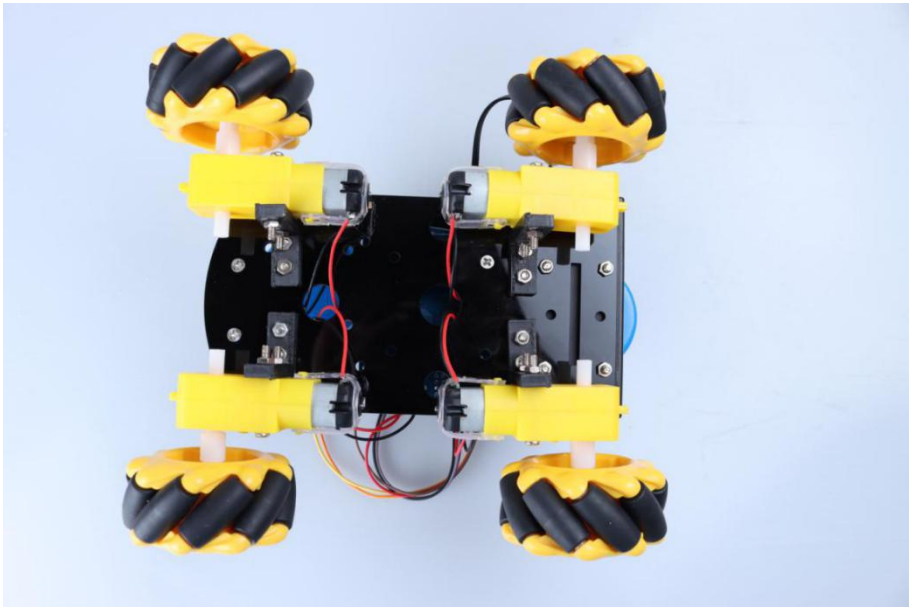




The M1,M2,M3 and M4 of the L293D drive plate are connected to four motors respectively. If the wiring sequence is different, the program must be modified, or the car will move in the wrong direction.







The motor wiring has been completed here

Three, program analysis

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

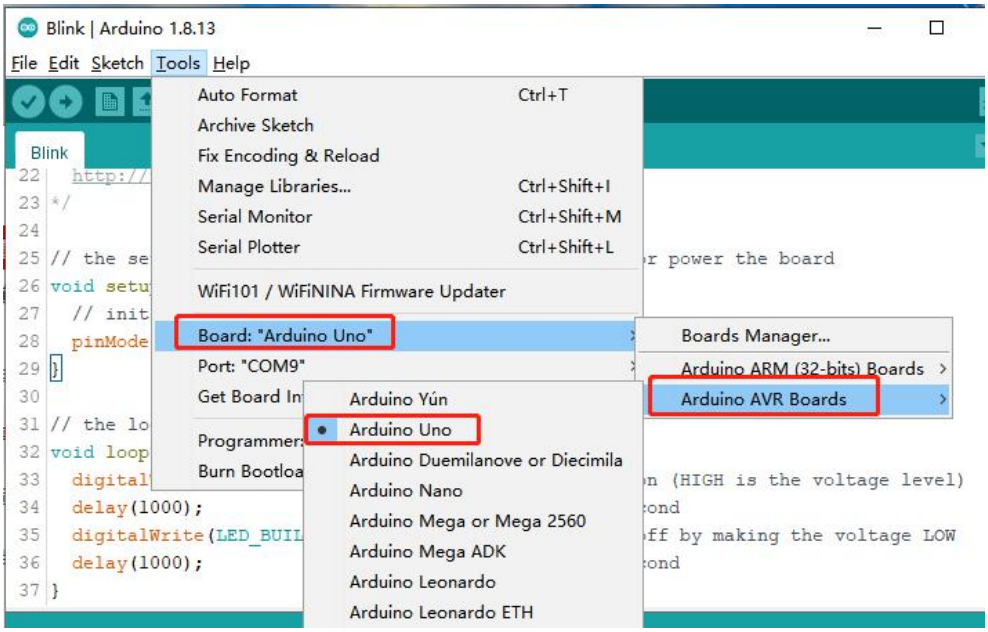
Each terminal port has two terminals that output high and low levels to enable forward and reverse motor rotation.

2. 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.

3. 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;

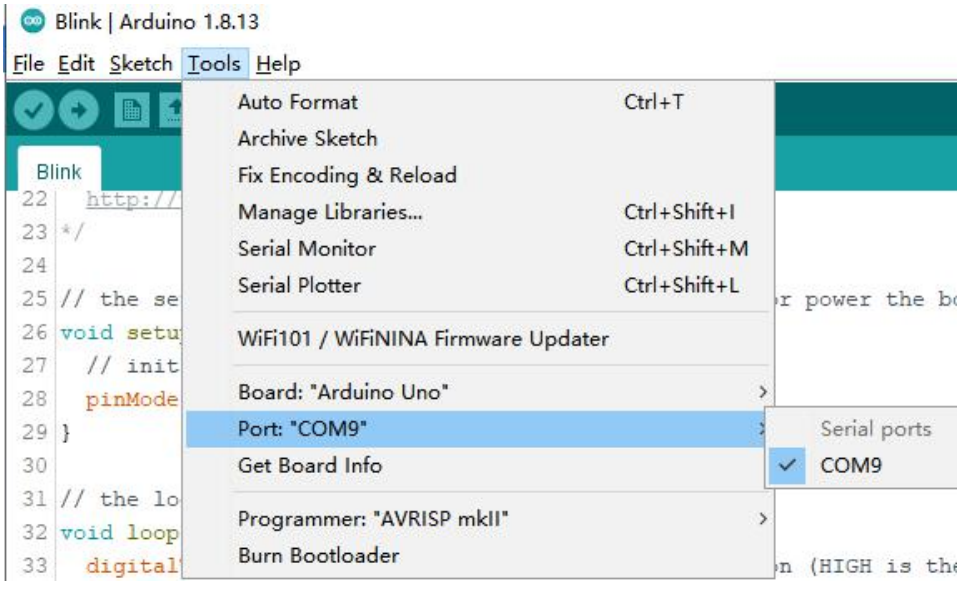
4. Select tools from the compiler, and select the development board: "Arduino Uno Option" from the following options, and then select the control board model: Arduino Uno;

As shown in the figure below.



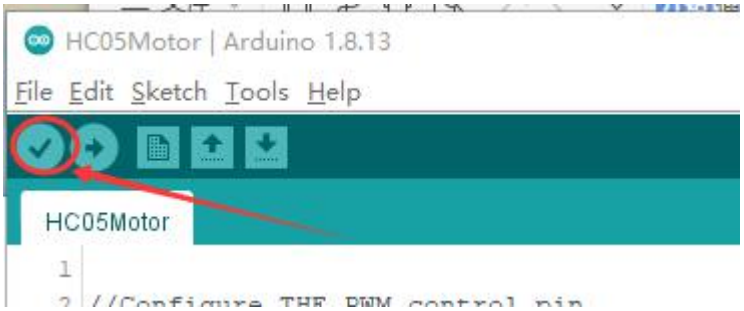
The upload code also needs to select the board port number. Our computer displays the port number of arduino as: "COM9 Arduio / Genuino Uno", you may be the other serial port number.

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

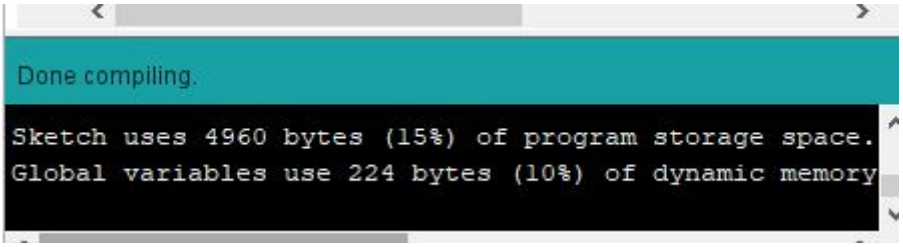


5. Verify the written program by selecting the verification button in the

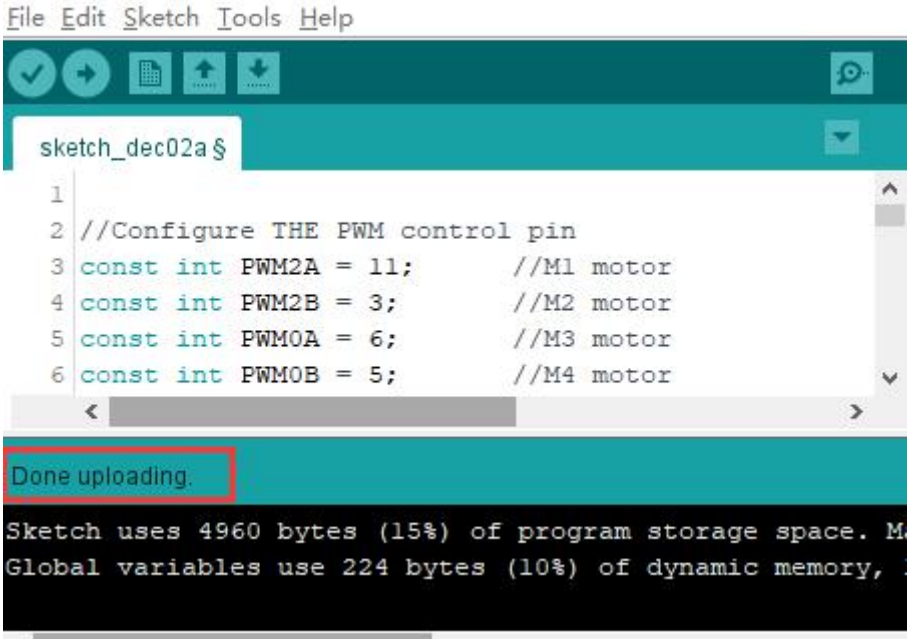
red circle below;



6. Complete compilation, no errors.



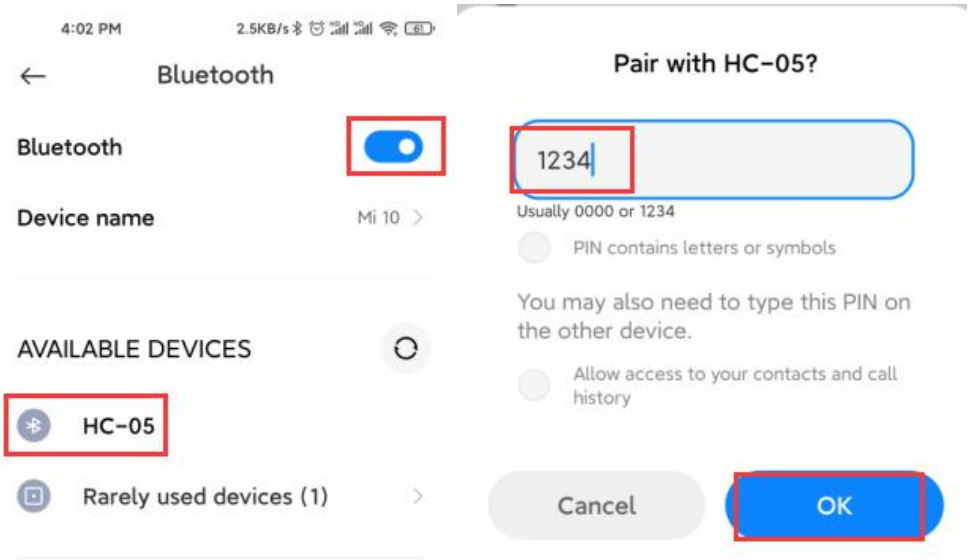
7. Select the "arrow to the right" upload button to upload the program to the Arduino Uno motherboard;



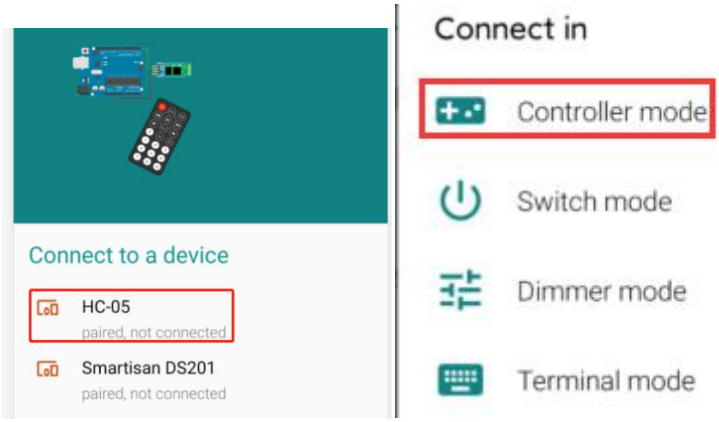
Equipped with bluetooth serial port assistant for remote

control of car;

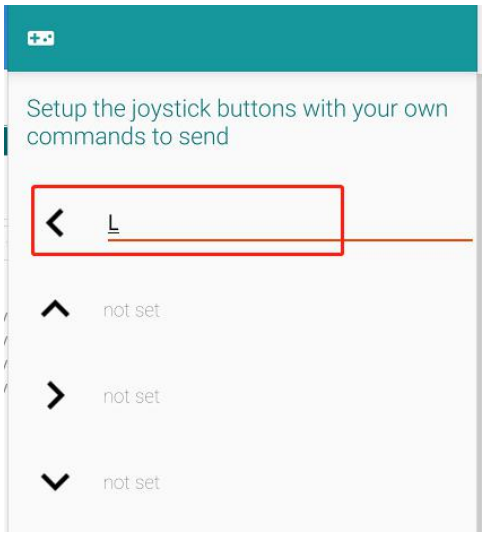
First, turn on bluetooth and search for Bluetooth devices to find hC-05 connection. The default connection pairing password is 1234.



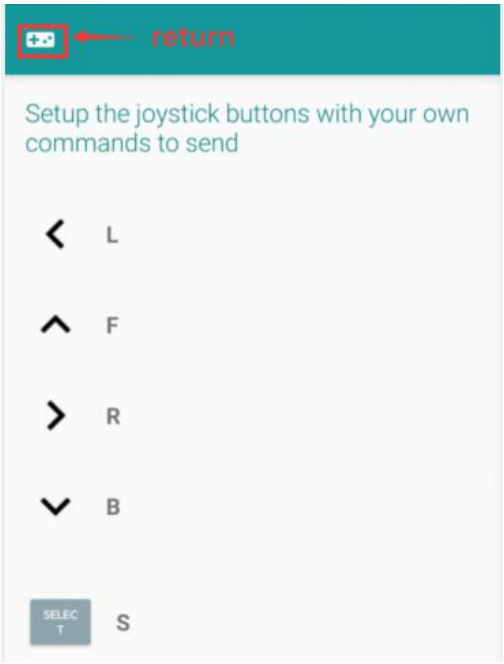
8. Open bluetooth Assistant and find the bluetooth module name hC-05 just configured. After successful connection, a remote control main interface will appear.



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

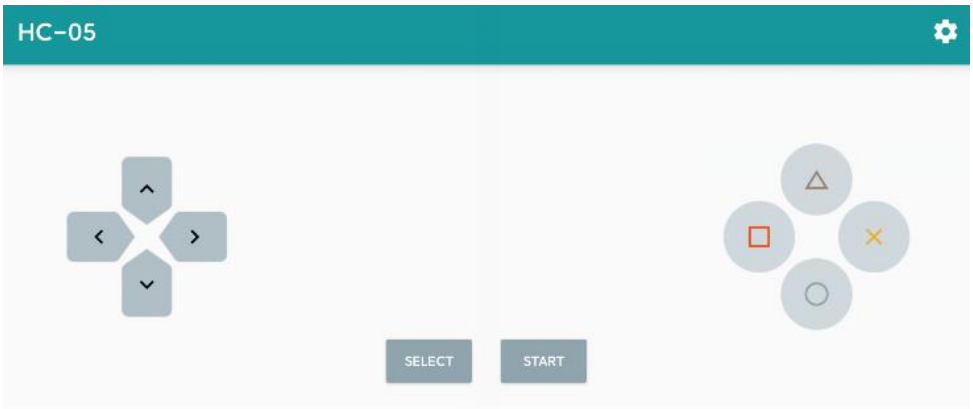


3. Just configured L,F,R,B and other commands, consistent with the program can be successful remote control car, otherwise the remote control fails;

```
//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);
  }
```

4. 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;



The code analysis

Pin configuration

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

5. Define the motion state parameters of the trolley

```
18
19 //Define motion state
20 const int Forward = 39;//39 is stored in the Forward variable
21 const int Back = 216;  //216 is stored in the Back variable
22 const int Left = 57;   //57 is stored in the Left variable
23 const int Right = 198; //The right amount of change
24 const int Stop = 0;    //Parking variable
```

6. 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.

```
38 void setup()
39 {
40     Serial.begin(9600);//Set the serial port baud rate 9600
41
42     //Configure for pin mode
43     pinMode(DIR_CLK,OUTPUT);
44     pinMode(DATA,OUTPUT);
45     pinMode(DIR_EN,OUTPUT);
46     pinMode(DIR_LATCH,OUTPUT);
47     pinMode(PWM0B,OUTPUT);
48     pinMode(PWM0A,OUTPUT);
49     pinMode(PWM2A,OUTPUT);
50     pinMode(PWM2B,OUTPUT);
51
52     //The Trig pin connected to the ultrasound is set to output mode
53     //The Echo pin connected to the ultrasound is set to input mode
54     pinMode(Trig,OUTPUT);
55     pinMode(Echo,INPUT);
56
57     void Motor(int Dri,int Speed1,int Speed2,int Speed3,int Speed4);
58     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.

7. Machine drive function

Void Motor()

Input parameters of the function:

Dir: motor motion direction parameter

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

AnalogWrite () function is called to speed PWM speed:

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

Dri values are explained as follows:

39 is for advance;

39 = 00100111

216 for astern;

216 = 11011000

57 represents the left translation;

57 = 00111001

198 means right translation;

198 = 11000110

Zero stands for stop

```
85 void Motor(int Dir,int Speed1,int Speed2,int Speed3,int Speed4)
86 {
87     analogWrite(PWM2A,Speed1); //1 Motor PWM speed regulation
88     analogWrite(PWM2B,Speed2); //2Motor PWM speed regulation
89     analogWrite(PWM0A,Speed3); //3Motor PWM speed regulation
90     analogWrite(PWM0B,Speed4); //4Motor PWM speed regulation
91
92   //Set the low level and write the direction of motion to prepare
93     digitalWrite(DIR_LATCH,LOW);
94   //Dir motion direction value writes
95     shiftOut(DATA,DIR_CLK,MSBFIRST,Dir);
96     //Set the high level and output the direction of motion
97     digitalWrite(DIR_LATCH,HIGH);
98 }
```

8. Adopt IO trigger ranging to give a high level signal of at least 10US;

The module automatically sends 8 40KHz square waves and automatically detects whether a signal is returned.

A signal returns and outputs a high level through IO. The duration of the high level is the duration of the ultrasonic wave from transmission to return.

Distance =(high level time * sound velocity (340m/s))/2;

When the ultrasonic ranging sensor pin "Trig" is high, it sends out ultrasonic waves. In order to ensure that 10 s sound waves are emitted, Trig pin needs to be pulled down and delayed for a short time before sending. The ranging function code is shown below:

```
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
}
```

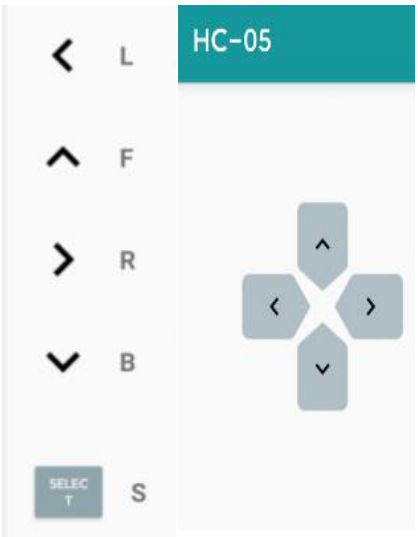
9. Serial. Available () returns the number of characters currently remaining in the Serial port buffer.

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.

```
131 void HC05()
132 {
133   // Determine whether the received data is greater than 0
134     if(Serial.available() > 0)
135     {
136         serialData = Serial.read(); // Receive function
137         // If the serial port receives data as character F,
138         //save F to CMD
139         if      ('F' == serialData )  cmd = 'F';
140         // If the data received by the serial port is character B,
141         //save F to CMD
142         else if('B' == serialData )  cmd = 'B';
143         // If the data received by the serial port is L,
144         //save F to CMD
145         else if('L' == serialData )  cmd = 'L';
146         // If the data received by the serial port is character R,
147         //save F to CMD
148         else if('R' == serialData )  cmd = 'R';
149         else if('S' == serialData )  cmd = 'S';
150
151         // Received string +, speed increases
```

10. 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;



```
184   //If Bluetooth receives the string F,
185   //the dolly moves forward and enables obstacle avoidance
186     if('F' == cmd)    //
187     {
188        AvoidingObstacles();//
189     }
190     //The ultrasonic obstacle avoidance function is called to
191     //realize the obstacle avoidance function
192     else if('B' == cmd)
193     {
194         Motor(Back,Speed1,Speed2,Speed3,Speed4);
195     }
196     // Bluetooth received string L, car left translation
197     else if('L' == cmd)
198     {
199         Motor(Left,Speed1,Speed2,Speed3,Speed4);
200     }
201     // When bluetooth receives the string R, dolly panalizes to the
202     else if('R' == cmd)
```

Call the AvoidingObstacles() function

```
213 void AvoidingObstacles()
214 {
215   // If the distance is greater than 20cm or
216   //bluetooth receives commands equal to F
217     if((distance > 20 ) || cmd == 'F')
218     {
219         delay(100);
220         // Again determine if the distance is really greater than 20cm
221         if(distance > 20)
222         {
223           // Call forward function
224             Motor(Forward,Speed1,Speed2,Speed3,Speed4);
225         }
226         // Otherwise the distance is less than 20
227         else
228         {
229             Motor(Back,Speed1,Speed2,Speed3,Speed4);
230             delay(500);
231             // Turn left to change the direction of the car
232             Motor(106,Speed1,Speed2,Speed3,Speed4);
233             delay(500);
```

11. Receive speed control

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.

```
151         // Received string +, speed increases
152         else if( serialData == '+' && Speed1 < 245)
153         {
154             Speed1 += 10;    //速度每次加10
155             Speed2 = Speed1;
156             Speed3 = Speed1;
157             Speed4 = Speed1;
158         }
159
160         //When I receive a string -- the speed decreases
161         else if( serialData == '-' && Speed1 > 30)
162         {
163           // The velocity decreases by 10 each time
164             Speed1 -= 10;
165             Speed2 = Speed1;
166             Speed3 = Speed1;
167             Speed4 = Speed1;
168         }
```

12. . Loop functions

```
63 void loop()
64 {
65     distance = SR04(Trig,Echo);// Obtain ultrasonic distance
66
67     HC05();// Call bluetooth car control function
68 }
69
```

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.