9.Raspberry Pi platform -------Bluetooth_control

1)Introduction of experimental

In this experiment, we control car by Bluetooth App by Android Mobile phone. The mobile phone sends commands through the serial port to control the advance, backward, turn left, turn right , stop, any angle control of the servo,whistle, speed of car.

At the same time, the status of various sensors on the car and the distance measured by the ultrasonic wave are displayed in real time on the Bluetooth APP interface by the serial port.

2)Experimental Steps

2.1) Android users scan the following QR code by browser or search "YahboomRobot" in Play Store to download APP;

iOS users scan the following QR code by camera or search "YahboomRobot" in App Store to download APP.

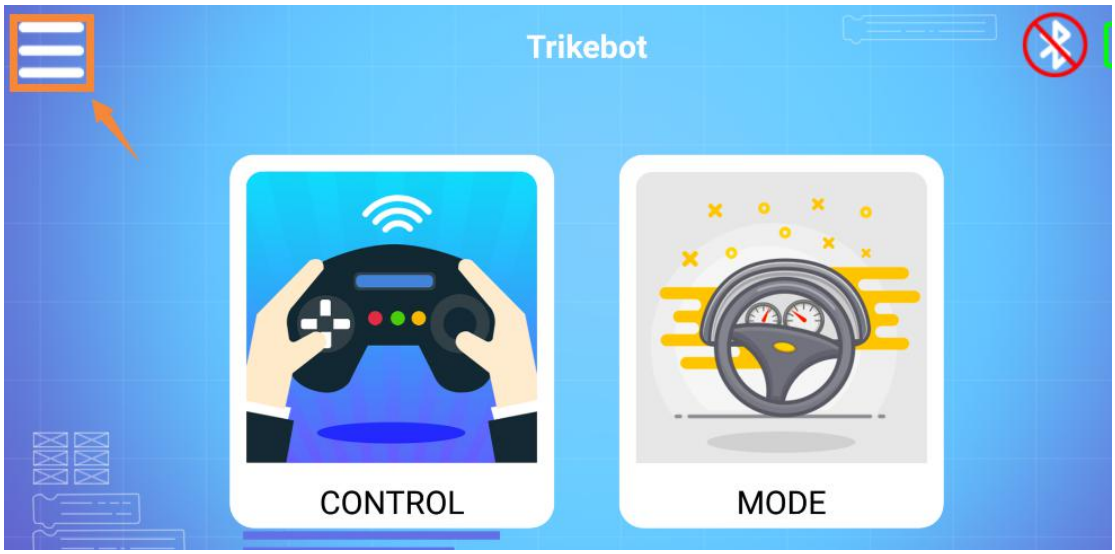
As shown in figure below.

!!Note:Because the software is relatively large, the download takes a certain amount of time, please be patient.

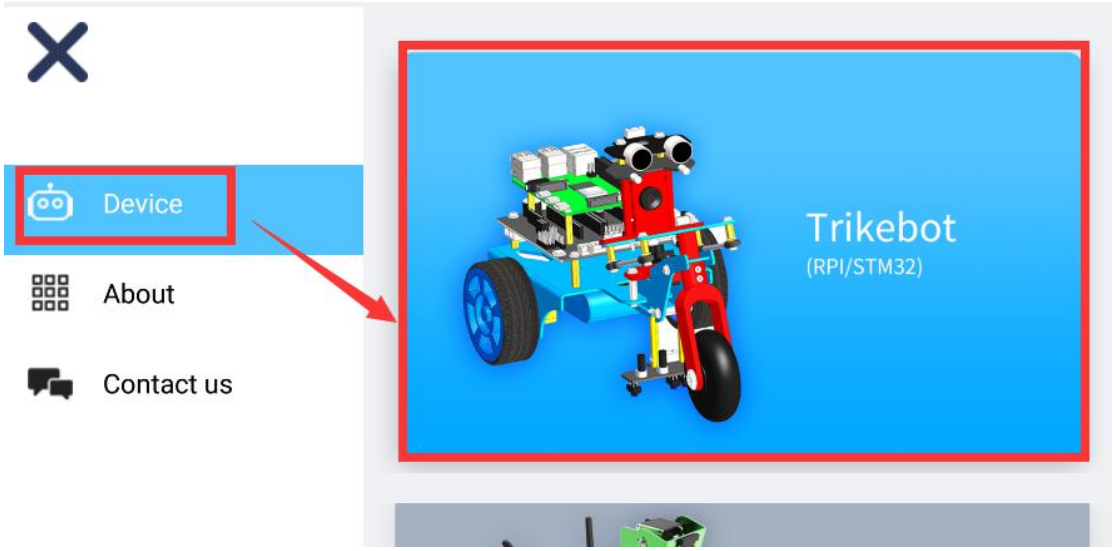
Note:During installation, If you find any prompts on your phone (for example: location permissions of your phone). You must select "Yes".

2.2) After the APP is installed, open the Bluetooth of the your phone, open the power switch of the Car, the red indicator of the Bluetooth module keeps flashing.

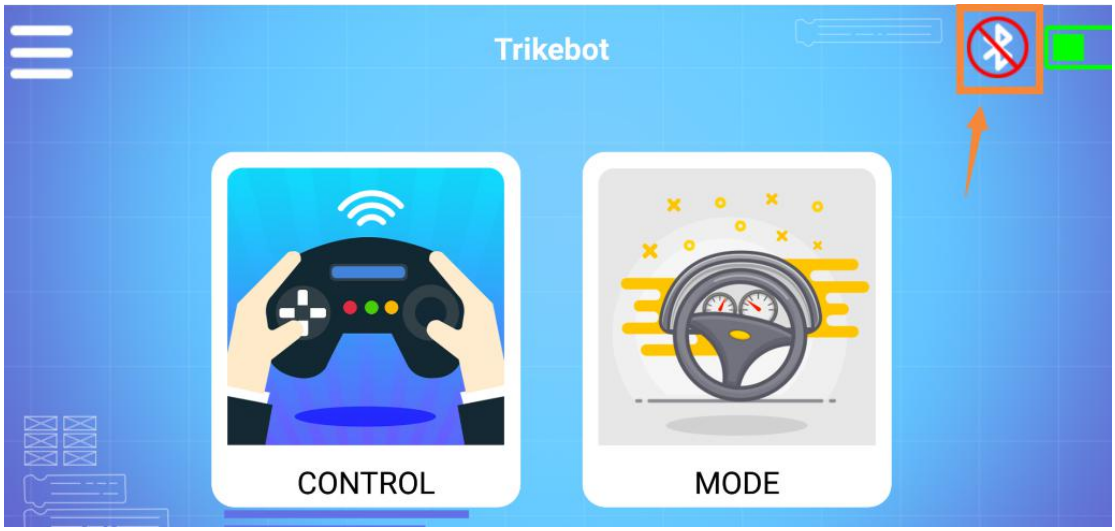
2.3) Then, open the **YahboomRobot** APK. You will see the APK interface and we need to click on the top left corner of the APK to select the device as shown below.



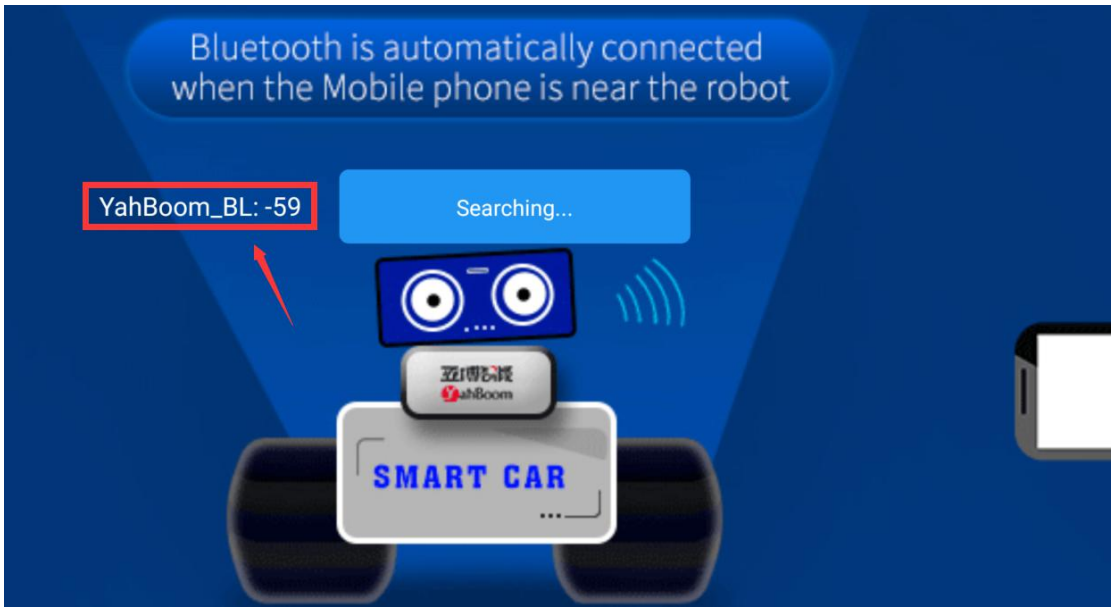
2.4) Select 【Trikebot】to enter the remote control interface, as shown below:



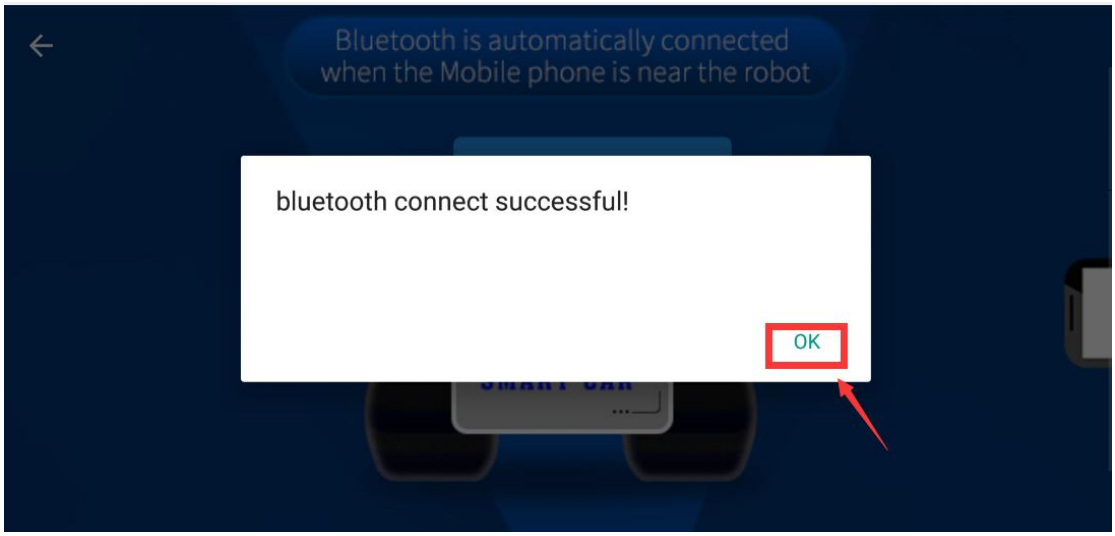
2.5) You will this interface as shown below. Click on the top right corner of the APK to connect Bluetooth.



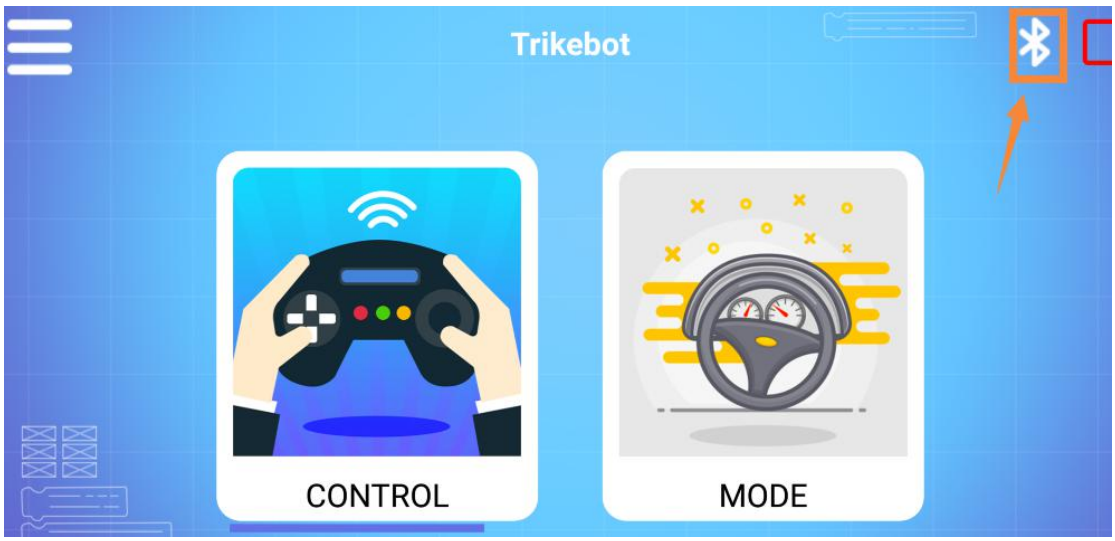
2.6) You can see bluetooth signal. Wait patiently, the phone will automatically connect to the Bluetooth near the Car.



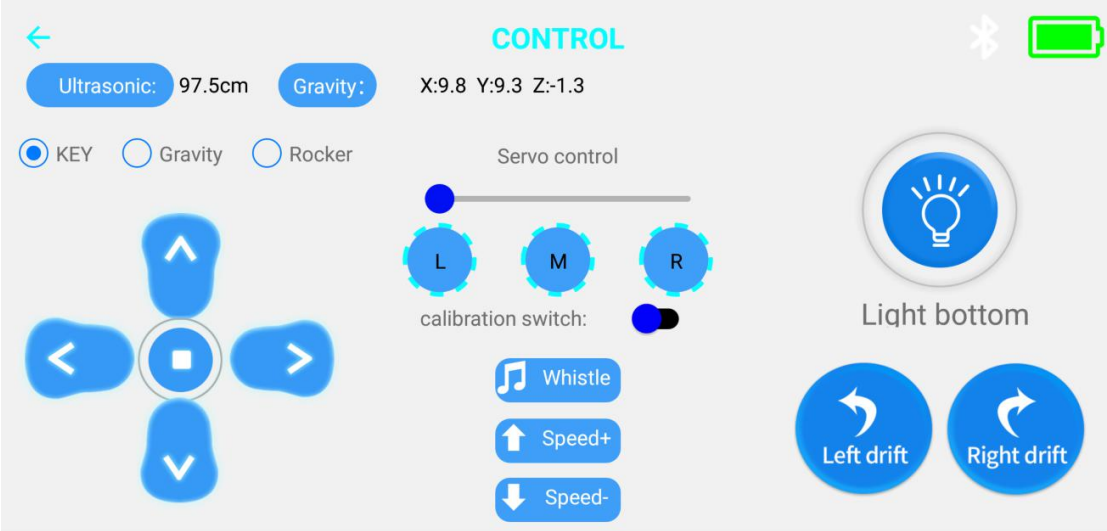
2.7) Bluetooth can be successfully connected, and the APP will enter the interface as shown below. At the same time, the red indicator of the Bluetooth module will be keep on. You need to click”OK”.



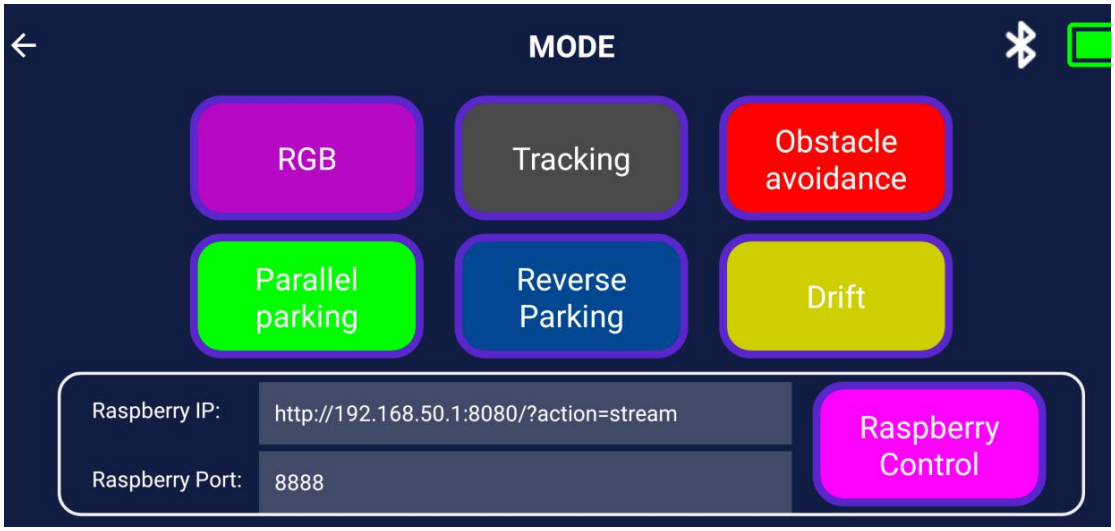
You will enter the interface as shown below.



2.8) Click “CONTROL” to enter interface as shown below. Wait for the ultrasonic data to change, it prove that Bluetooth starts to transmit data normally. You can start to control the car.



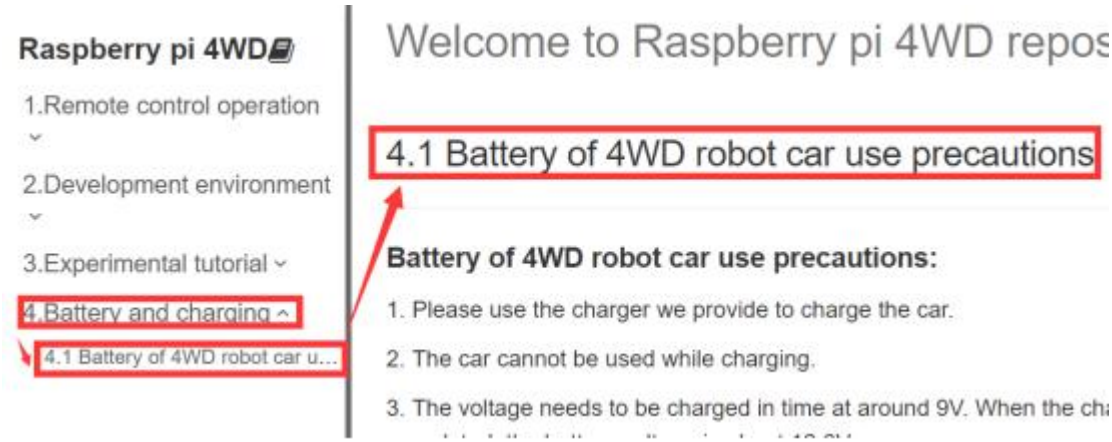
2.9) Click “MODE” to enter interface as shown below.



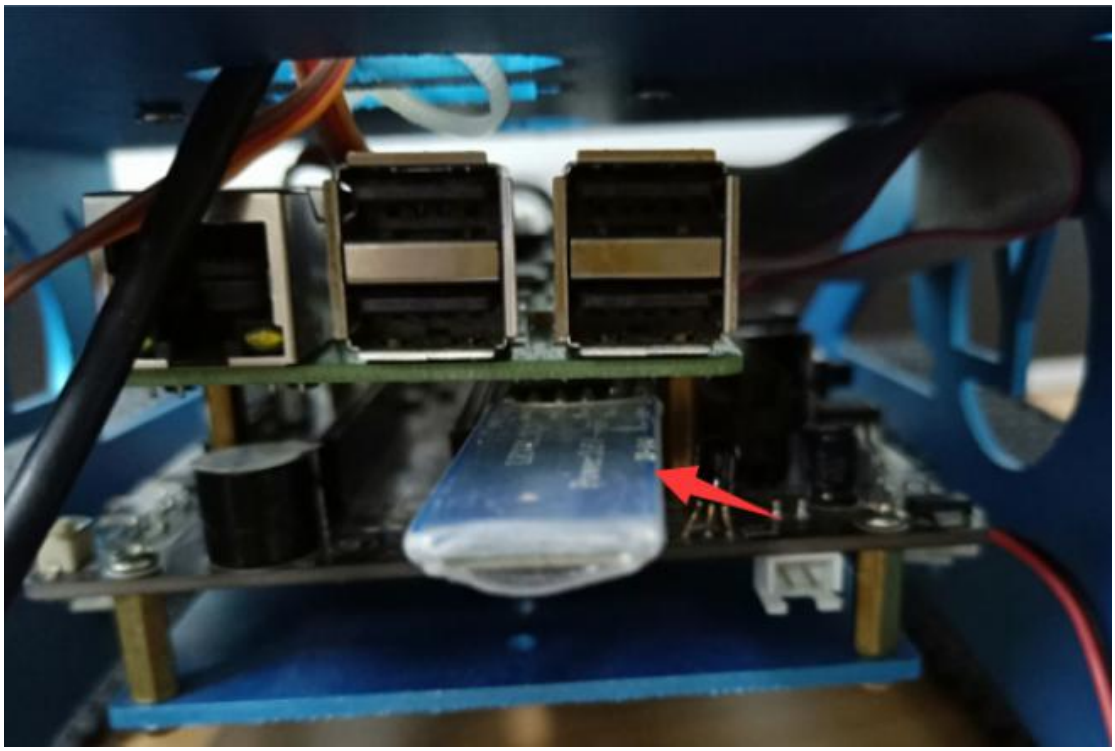
You need to pay attention to the points, otherwise the Bluetooth remote control function will have problems.

Note:

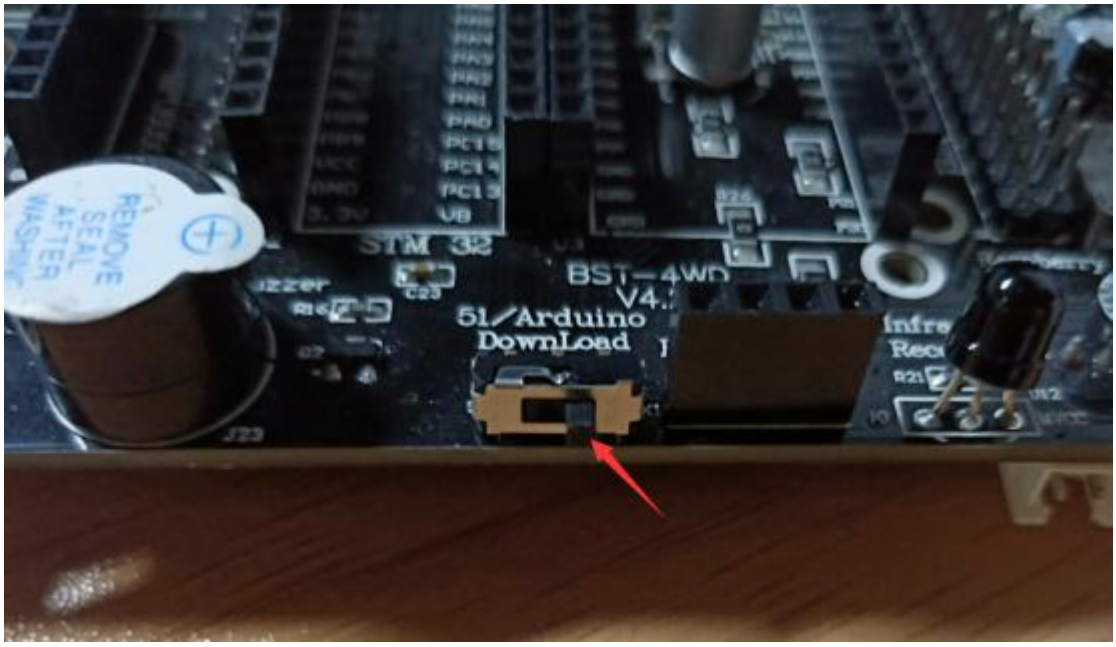
(1)The robot Car needs to have enough voltage to work properly. Please refer to the following figure for the charging method and battery usage:



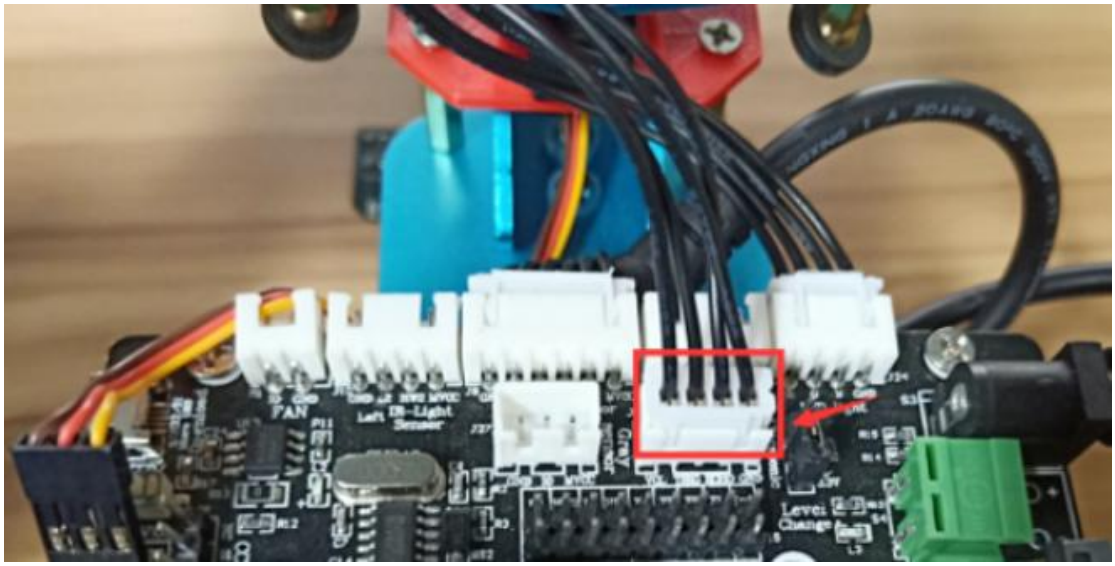
(2)The Bluetooth module needs to be properly inserted into the expansion board of the Car. As shown in the figure below.



(3)51/Arduino Download Switch on the expansion board must be set to [OFF]. As shown in the figure below.



(4) The ultrasonic module must be inserted. As shown in the figure below.



Please read our manual for introductions of Bluetooth remote control interface.

About camera:

If you want to use camera, you need to connect the camera and the Raspberry Pi motherboard correctly.

Your phone must connect WiFi of the Car. Aa shown below.

Name: Yahboom_TrikeBot

Password: 12345678

This WiFi is only used to transmit video and cannot be accessed online.

When you connect to WiFi, Click "**RaspberryControl**" you can see the picture taken by the camera on your mobile phone.

Video as shown below.

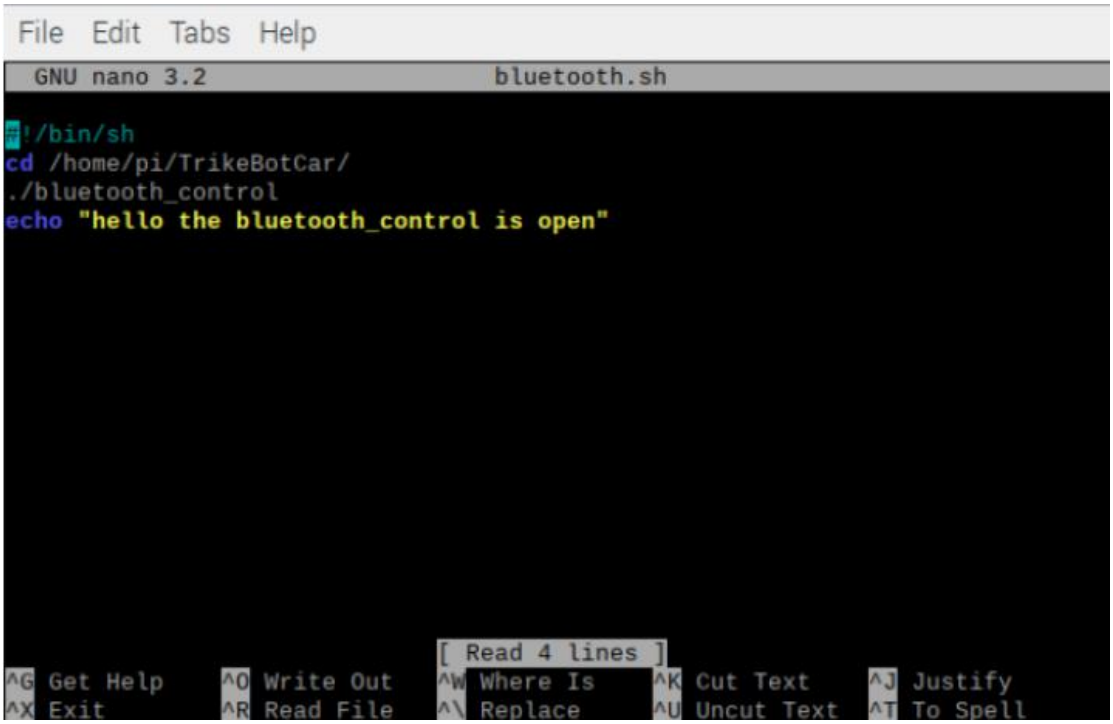
3) About the code

If you are using Yahboom_TrikeBot car_image we provided, you need to complete following steps.

Input command:

sudo nano bluetooth_control.sh

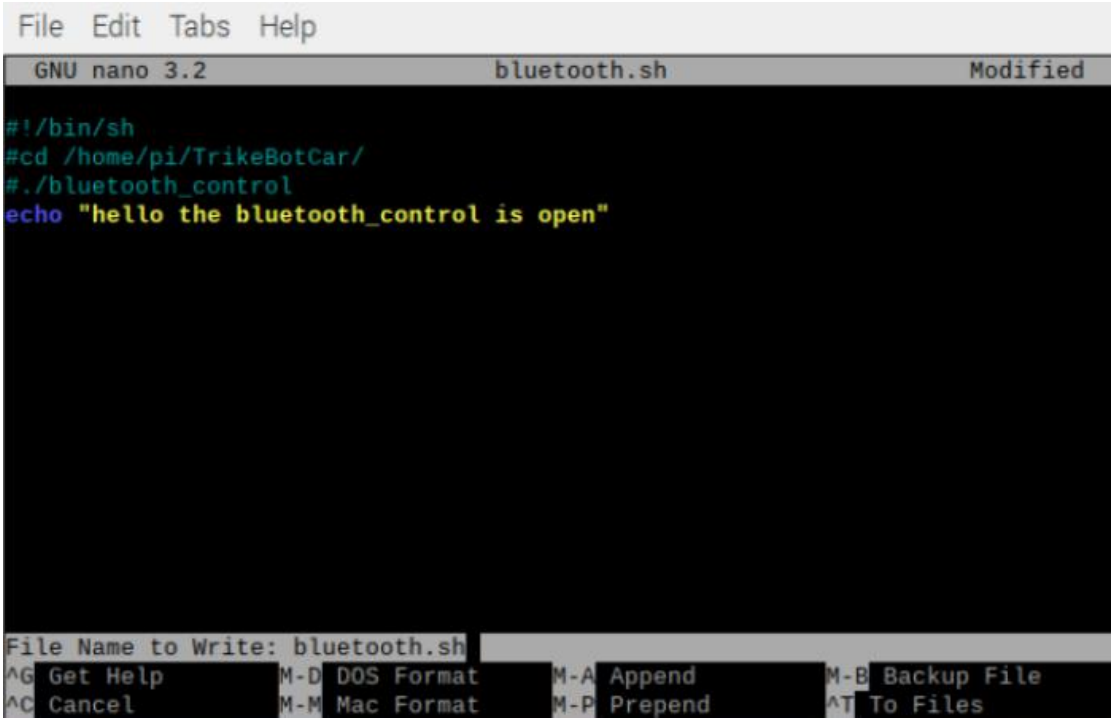
You will see following interface.



You need to make the following modifications

#cd /home/pi/TrikeBotCar

#./bluetooth_control



Save ans exit.

A. For .c code

(1) We need to compile this file in the Raspberry Pi system. (Note: we need to add -lwiringPi and -lpthread to the library file.)

We need to input: gcc bluetooth_control.c -o bluetooth_control -lwiringPi -lpthread
(2) We need to run the compiled executable file in the Raspberry Pi system.We need to input: ./bluetooth_control

```
gcc bluetooth_control.c -o bluetooth_control -lwiringPi -lpthread
./bluetooth_control
```

(3)We can input: ctrl+c to stop this process, which mean is send a signal to the linux kernel to terminate the current process, but the state of the relevant pin is uncertain at this time, we also need to run a script to initialize all pins.
(Note:The initpin.sh script file is included in the SmartCar/python directory.)
You need to input: sudo chmod 777 initpin.sh
./initpin.sh

```
pi@yah~ $ sudo chmod 777 initpin.sh
pi@y $ ./initpin.sh
```

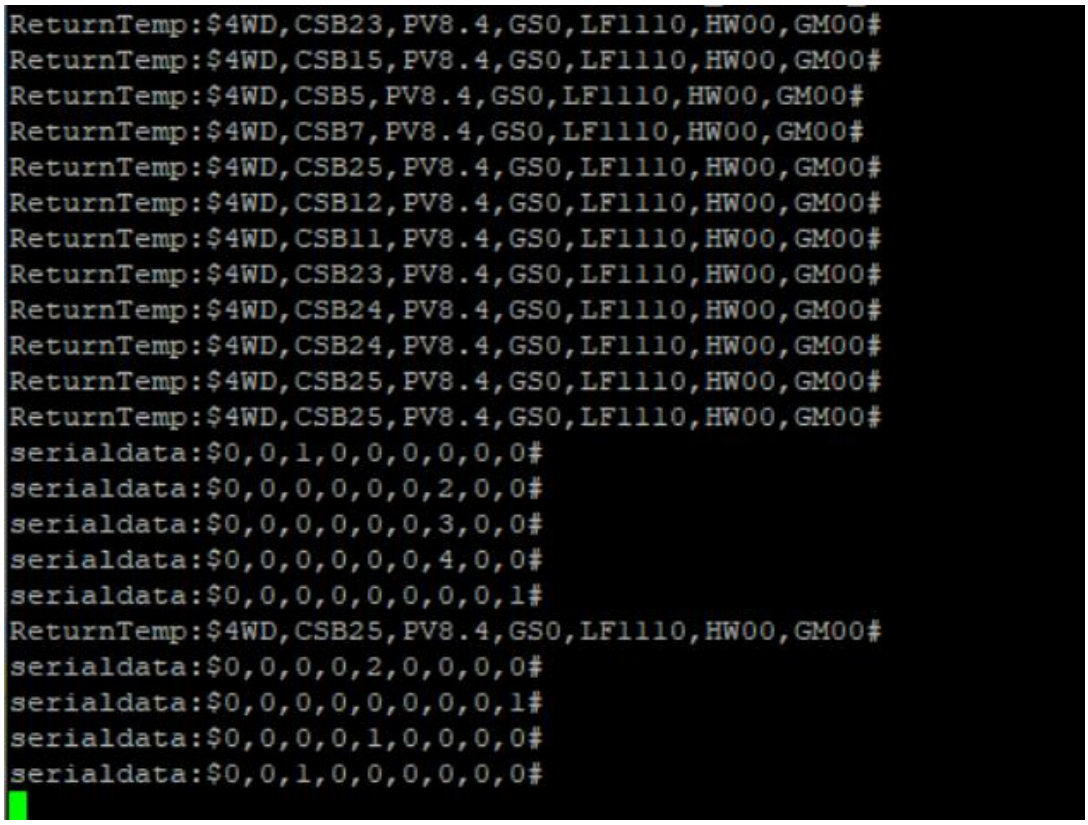
B. For python code
(1) We need to input following command to run python code.
python bluetooth_control.py

```
python bluetooth_control.py
```

(2) We can input: ctrl+c to stop this process, which mean is send a signal to the linux kernel to terminate the current process, but the state of the relevant pin is uncertain at this time, we also need to run a script to initialize all pins.
(3) You need to input: sudo chmod 777 initpin.sh
./initpin.sh

```
pi@yah~ $ sudo chmod 777 initpin.sh
pi@y $ ./initpin.sh
```

In normally circumstances, when you click the button on the APP, to control the car, these data will be printed here.


ReturnTemp:$4WD,CSB23,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB15,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB5,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB7,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB25,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB12,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB11,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB23,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB24,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB24,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB25,PV8.4,GS0,LF1110,HW00,GM00#
ReturnTemp:$4WD,CSB25,PV8.4,GS0,LF1110,HW00,GM00#
serialdata:$0,0,1,0,0,0,0,0,0#
serialdata:$0,0,0,0,0,0,2,0,0#
serialdata:$0,0,0,0,0,0,3,0,0#
serialdata:$0,0,0,0,0,0,4,0,0#
serialdata:$0,0,0,0,0,0,0,0,1#
ReturnTemp:$4WD,CSB25,PV8.4,GS0,LF1110,HW00,GM00#
serialdata:$0,0,0,0,2,0,0,0,0#
serialdata:$0,0,0,0,0,0,0,0,1#
serialdata:$0,0,0,0,1,0,0,0,0#
serialdata:$0,0,1,0,0,0,0,0,0#