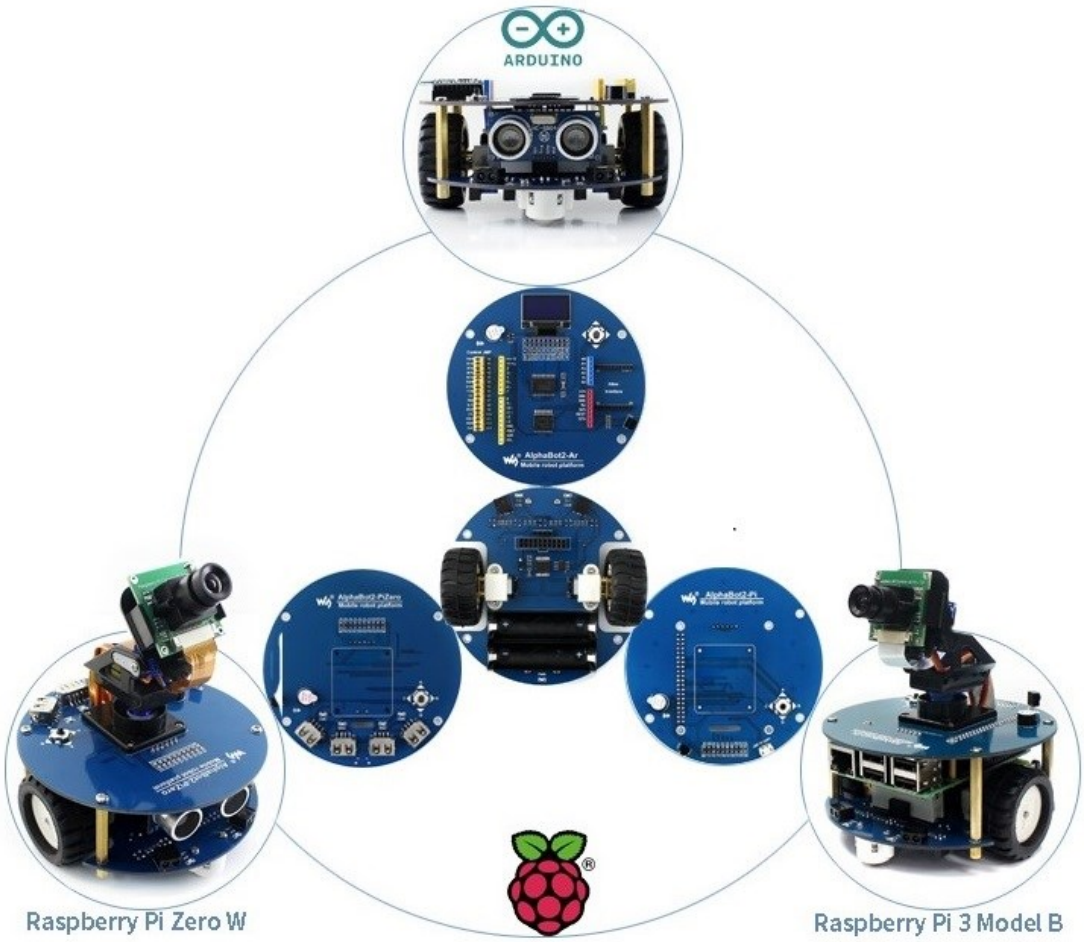
AlphaBot 2

用户手册

August 2017

V1.3



阅读提示

产品选型

说明：不同的套餐，配件可能是不同的。虽然本手册将介绍所有配件，但是您购买的套餐可能并不包含其中的某些配件。关于兼容的配件，请参见选型列表。

配置		套餐/配件包				
配件	说明	AlphaBot2-Ar 配件包	AlphaBot2-Ar	AlphaBot2-Pi 配件包	AlphaBot2-Pi	AlphaBot2-Pi Zero 配件包
AlphaBot2-Base	电机驱动板、集成避障，循迹传感器	√	√	√	√	√
AlphaBot2-Ar	适配板，可接入 Arduino	√	√			
AlphaBot2-Pi	适配板，可接入 RPi3 B			√	√	
AlphaBot2-PiZero	适配板，可接入 RPi Zero W					√
RPi3 B	Raspberry Pi 3 Model B				√	
UNO PLUS	Arduino 兼容板 增强板		√			
Dual-mode Bluetooth	双模蓝牙模块		√			
超声波传感器	超声波避障、测距	√	√			√
红外遥控	红外遥控小车	√	√	√	√	√

器						
RPi Camera (B)	树莓派摄像头，可调焦			√	√	√
SG90 舵机及云台	舵机，控制摄像头转动			√	√	√
Micro SD Card 16GB	Class10 16GB Micro SD 卡			√	√	√
5V 2.5A 电源	RPi3 B 须用 2.5A 以上电源			√	√	√

【说明】

1、AlphaBot2-Ar 配件包需要搭配 Arduino 使用，如果您没有，推荐选购套餐：AlphaBot2-Ar

2、AlphaBot2-Pi 配件包 需要搭配 Raspberry Pi 3 使用，如果您没有，推荐选购套餐：AlphaBot2-Pi

3、AlphaBot2-PiZero 配件包 需要搭配 Raspberry Pi Zero/Zero W 使用，您需要自备 Raspberry Pi Zero/Zero W

4、超声波传感器由于结构原因不能接到 AlphaBot2-Pi 套餐

前言

许多单片机爱好者都曾经想设计一台属于自己的智能小车。然而当真正动手时却发现并不像想象中简单，可能一知半解甚至毫无头绪。本套件能让您快速了解智能小车的结构框架、基本原理。让你快速入门智能小车的设计开发。

本手册主要介绍如果使用小车，如何运行示例程序实现各种功能。由于篇幅所限，本手册不会涉及太多的程序原理。部分内容可能会让用户阅读起来稍感困难，我们建议用户在使用本款小车之前，先稍微学习树莓派的 Python 编程或者 Arduino 编程，熟悉使用树莓派和 Arduino。建议结合智能车教程查看源代码，以便深入理解示例程序原理。内容深入浅出，从最小系统到整个小车的主控设计，从简单的小车直线运动到复杂的多功能复合实现。希望单片机初学者可以通过本套件对智能小车有清楚的认识。

树莓派教程：http://www.waveshare.net/study/portal.php?mod=list&catid=29

Arduino 教程：http://www.waveshare.net/study/portal.php?mod=list&catid=7

智能车教程：http://www.waveshare.net/study/portal.php?mod=list&catid=57

由于作者水平有限，教程中错误和不足之处在所难免，敬请读者批评、指正。

由于小车的电气特性所限，相对于其他的开发板，用户应当更加注意安全、正确地使用本产品，以避免小车底板对树莓派或 Arduino 板造成不可逆的电器损坏。因此首次使用本产品的用户，应当尽量按照本手册的操作步骤进行操作。

☺ 有任何疑问，都欢迎来到微雪学堂 http://www.waveshare.net/study 留言。

获取产品资料

用户可通过微雪百科获取 AlphaBot2 的详细资料，包括用户手册、原理图、示例程序、数据表等资源，以便正确、安全地使用本产品。

☺ 资料路径：www.waveshare.net/wiki/AlphaBot2

目录

AlphaBot 2 用户手册 1

阅读提示 2

产品选型 2

前言 4

获取产品资料 5

目录 6

板载资源 7

简介 7

AlphaBot2-Base 8

AlphaBot2-Ar 9

AlphaBot2-Pi 10

AlphaBot2-PiZero 11

快速入门 12

Arduino 示例程序 12

树莓派示例程序 23

常见问题解答 41

安全须知 43

注意 请保留此信息以供将来参考 43

板载资源

简介

AlphaBot2 智能车开发套件包含一个基板 AlphaBot2-Base 和一个适配板（AlphaBot2-Ar、AlphaBot2-Pi、AlphaBot2-PiZero 三者之一）。通过选择不同的适配板，小车可以支持接入 Arduino，树莓派 3B，树莓派 Zero W。

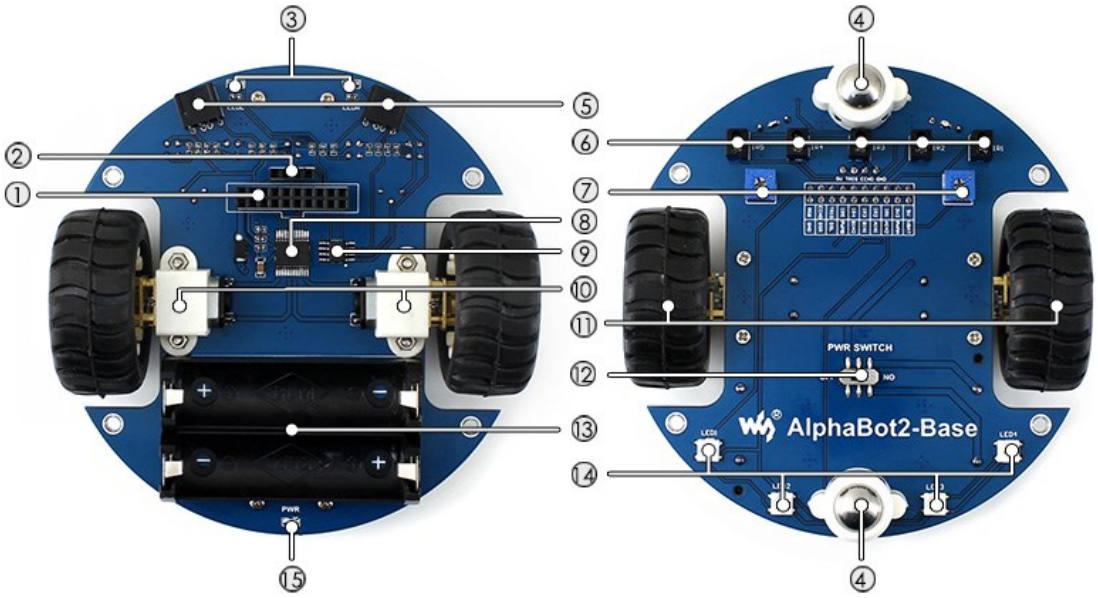
本智能车套件结构稳定，集成度高，而且还不用复杂的组装以及繁琐的接线。通过不同的适配板可以让 AlphaBot2 接入树莓派或 Arduino 控制，配合微雪提供的开源代码和学习教程，便能快速了解和探索智能小车的学习和开发。

这其中涉及到的技术要点：

循迹、避障、红外测距、手机/PC 视频监控、WIFI/蓝牙/红外无线遥控等。

注：此用户手册不针对某个特定的套餐，可能有些套餐不包含下文介绍到的部分模块，不能实现部分功能。

AlphaBot2-Base



[接口/器件简介]

1. AlphaBot2-Base 控制接口

方便接入 AlphaBot2 各种适配板

2. 超声波模块接口

3. 避障信号灯

4. 万向轮

5. ST188

反射式红外光电传感器，用于避障

6. ITR20001/T

反射式红外光电传感器，用于循迹

7. 电位器

调节避障感应距离

8. TB6612FNG

双 H 桥电机驱动芯片

9. LM393

电压比较器

10. N20 微型减速电机

减速比 1:30,6V/600 转

11. 橡胶轮

直径 42MM，宽 19MM

12. 电源开关

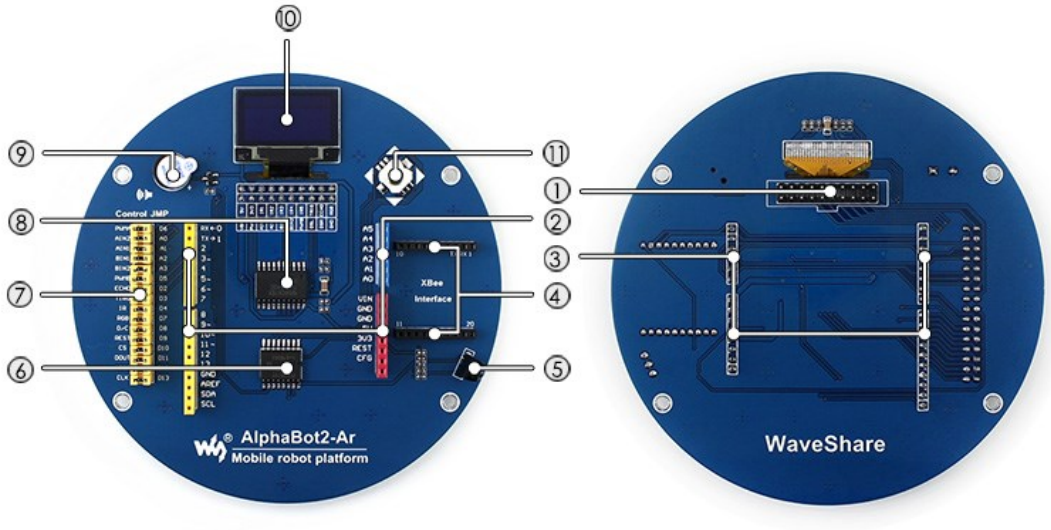
13. 14500 电池座

14. WS2812B

全彩 RGB LED 灯

15. 电源指示灯

AlphaBot2-Ar



[接口简介]

1. AlphaBot2-Base 控制接口

方便接入 Alphabot2-Base 基板

2. Arduino 扩展接口

方便接口 Arduino 扩展板

3. Arduino 连接口

方便接入 Arduino 控制板

4. Xbee 接口

方便接入蓝牙双模模块，蓝牙遥控小车

[器件/跳线说明]

5. 红外接收头

6. PC8574

I/O 扩展芯片，I2C 接口

7. Arduino 外设跳线

8. TLC1543

10 位 AD 采集芯片

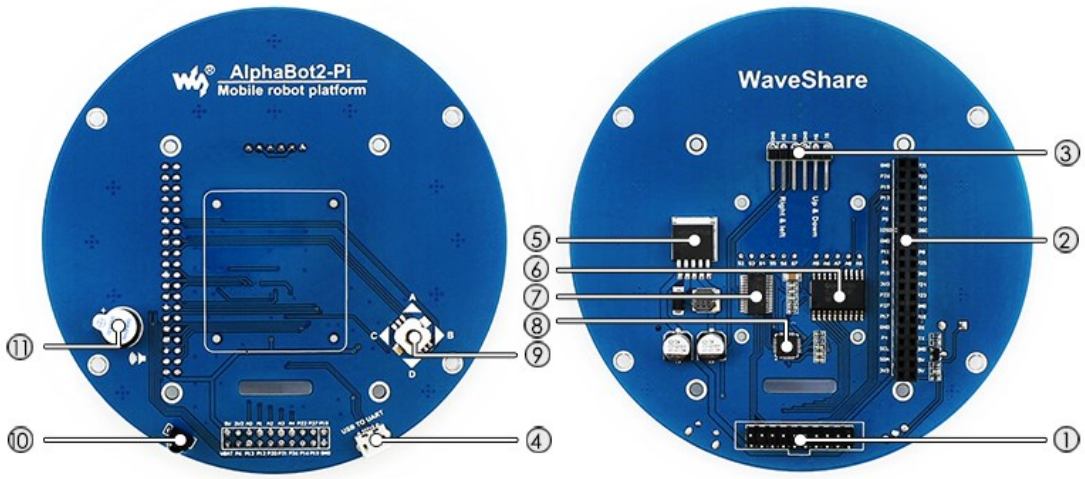
9. 蜂鸣器

10. 0.96 寸 OLED

驱动芯片为 SSD1306，128×64 分辨率

11. 五向摇杆

AlphaBot2-Pi



[接口简介]

1. AlphaBot2-Base 控制接口

方便接入 AlphaBot2-Base 基板

2. 树莓派连接口

方便接入 Raspberry Pi 3 Model B

3. 舵机接口

4. USB TO UART 接口

方便通过串口控制树莓派

方便接入 NRF24L01 无线模块

[器件说明]

5. LM2596

5V 电源稳压芯片

6. TLC1543

10 位 AD 采集芯片，方便树莓派接入模拟传感器

7. PCA9685

舵机控制芯片，控制舵机不抖动，更稳定

8. CP2102

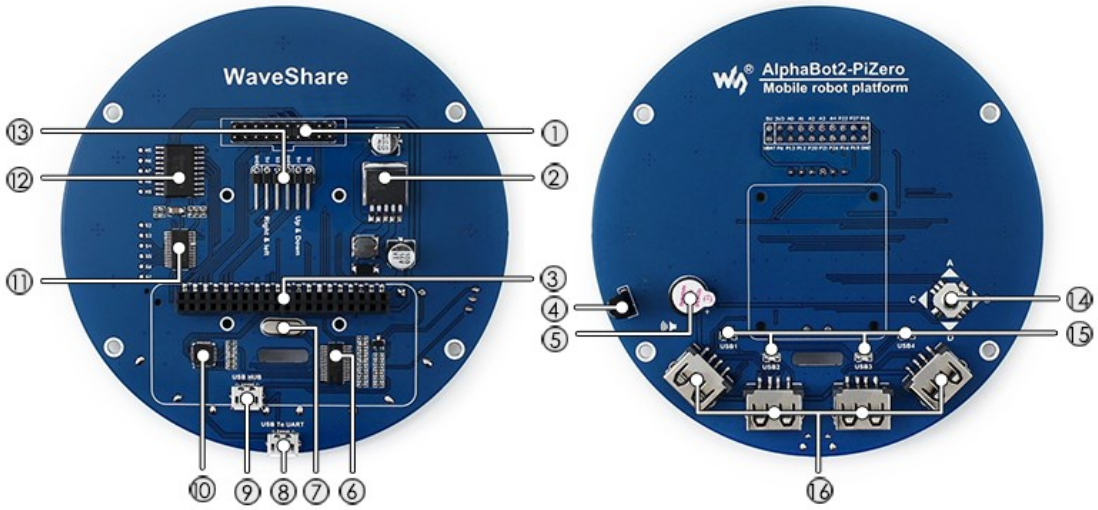
USB 转串口芯片

9. 五向摇杆

10. 红外接收头

11. 蜂鸣器

AlphaBot2-PiZero



[接口/器件简介]

1. AlphaBot2-Base 控制接口

方便接入 Alphabot2-Base 基板

2. LM2596

5V 电源稳压芯片

3. 树莓派连接口

方便接入 Raspberry Pi Zero/Zero W

4. 红外接收头

实现红外遥控小车

5. 蜂鸣器

6. FE1.1S

USB HUB 芯片

7. 12M 晶振

8. USB TO UART 接口

方便通过串口控制树莓派

9. USB HUB 接口

扩展 Raspberry Pi Zero/Zero W 的 USB 口

10. CP2102

USB 转串口芯片

11. PCA9685

舵机控制芯片，控制舵机不抖动，更稳定

12. TLC1543

10 位 AD 采集芯片，方便树莓派接入模拟传感器

13. 舵机接口

14. 五向摇杆

15. USB 指示灯

16. USB 端口

方便接入各种 USB 设备

快速入门

我们有提供 Arduino 和树莓派的控制程序。可以实现循迹，避障，遥控，视频等功能。

下面的示例程序都可以从微雪百科获取。将 Arduino 目录下的 libraires 库文件复制到 Arduino IDE 安装目录下 libraries 文件夹中。重启 Ariduino IDE，点击 File-> Example，查看是否有 TRSensors 选项，若有说明库导入成功。树莓派程序解压并放在 Raspbian 系统的/home/pi 目录下。

Arduino 示例程序

电机测试

编译并下载 Run-Test 程序，程序运行后电机转动，小车向前走。如果方向不对需要相应修改电机接线，或者修改程序中的管脚。建议采用第二种方法。左边轮子转动方向不对则对调 AIN1，AIN2，右边轮子转动方向不对则对调 BIN1，BIN2。

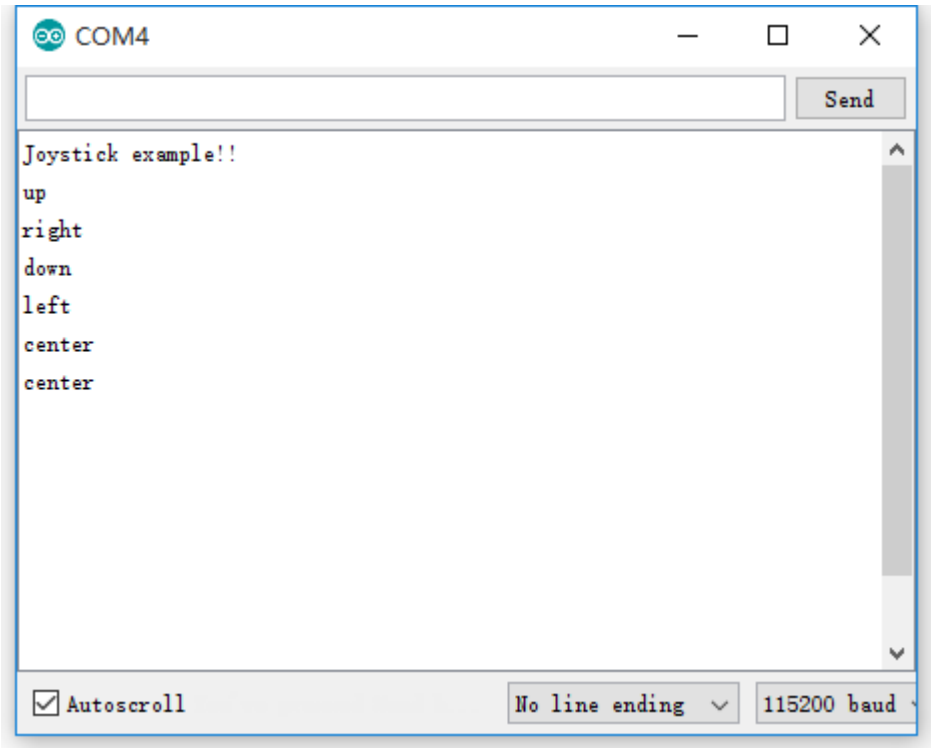
如果此程序轮子转动方向不对，以下全部程序都需要作相应修改。

```
#define PWMA   5          //Left Motor Speed pin (ENA)
#define AIN2   A0         //Motor-L forward (IN2).
#define AIN1   A1         //Motor-L backward (IN1)
#define PWMB   6          //Right Motor Speed pin (ENB)
#define BIN1   A2         //Motor-R forward (IN3)
#define BIN2   A3         //Motor-R backward (IN4)
```

五向摇杆

编译并下载 Joystick 程序，Joystick 5 个方向的操纵杆，包括上、下、左、右以及确定。打开串口监视器，波特率设置为 115200，串口监视器会显示当前按键的键位，电机会根据按键方向转动，当按键按下时蜂鸣器会响起。如果轮子转动方向和按键方向不一样，需修改

一下管脚。



红外遥控小车

编译并下载 IR 程序，按下红外遥控的数字键控制小车，2，8，4，6，5 分别代表前进，后退，左转，右转，停止。按下 – 或 + 可调节速度，按下 EQ 速度恢复默认值。

不同的红外遥控器可能按键编码不一样，如果不一样需相应修改程序。

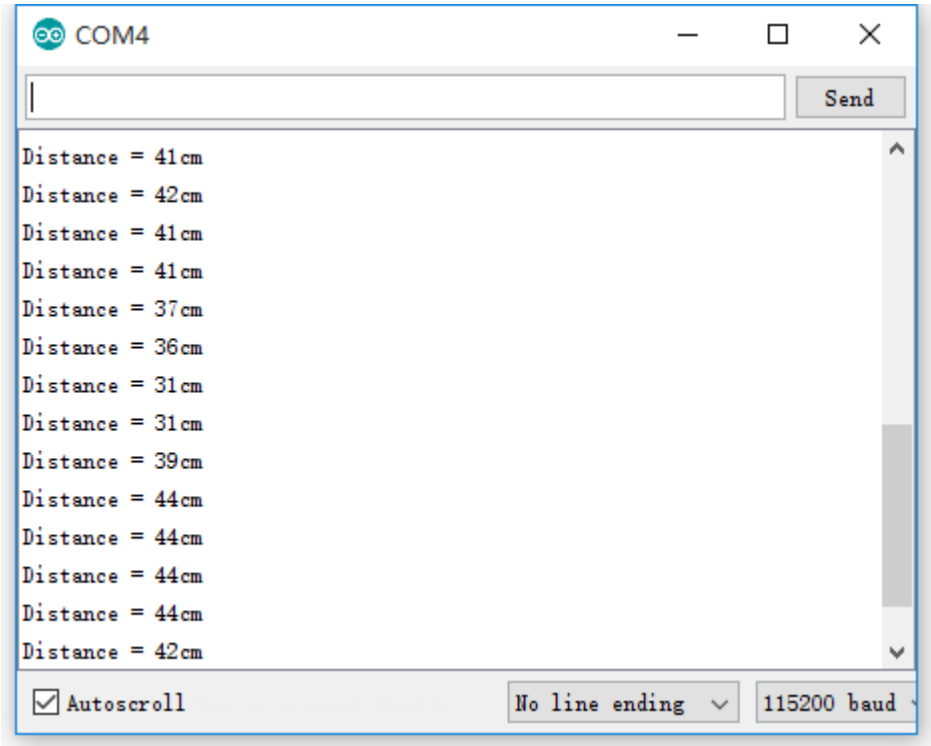
红外避障

编译并下载 Infrared-Obstacle-Avoidancec 程序，小车前面没有障碍物时，前面绿色的 LED 灯熄灭，小车遇到障碍物时，前面绿色的 LED 灯会亮。如果 LED 灯不亮或者一直亮可以调节小车底面的两个电位器，使 LED 处于刚好熄灭的状态。

程序现象是没有遇到障碍时小车直走，遇到障碍物时小车右转。

超声波测距

编译并下载 Ultrasionc_Ranging 程序，打开串口监视器，波特率设置为 115200，串口监视器会显示当前测量距离。



超声波避障

编译并下载 Ultrasionc-Obstacle-Avoidance 程序，没有遇到障碍时小车直走，遇到障碍物时小车右转。由于超声波会反射，当前方障碍物平面不是在小车正前方，与小车形成夹角时测量的距离有可能不准。

超声波红外避障

编译并下载 Ultrasionc-Infrared-Obstacle-Avoidance 程序，没有遇到障碍时小车直走，遇到障碍物时小车右转。

循迹传感器测试

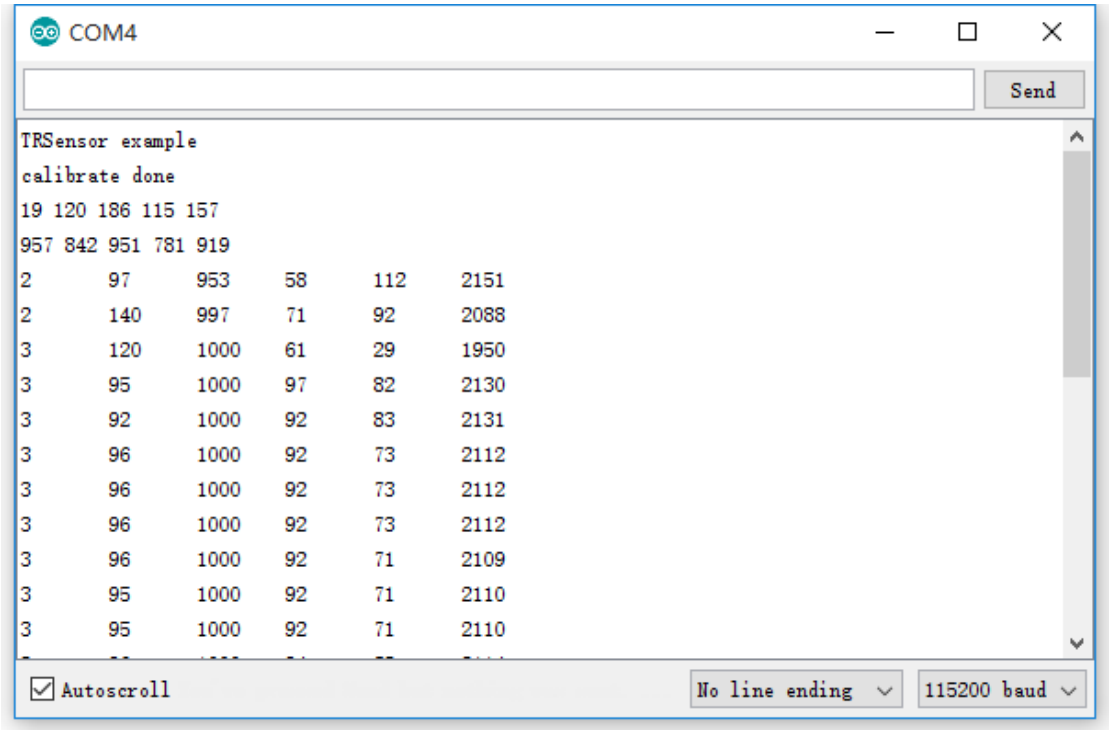
将 TRSensors 库导入到 Arduino IDE 安装目录下 libraries 中，点击 File -> Example -> TRSensors ->TRSensorExample 打开工程，编译并下载到 Arduino 开发板。

循迹传感器可以探测黑线，底色为白色（或者白线黑底，须修改程序）。可以用黑线胶布粘在白色的 KT 板中制作小车循迹场地，黑色轨迹宽度为 15cm。若底色颜色过深会影响循迹效果。

打开串口监视器，波特率设置为 115200，串口显示“TRSensor example”时，用手拿着小车在黑线中紧贴地面左右摇晃，此阶段为校准，使小车可以设别各个传感器的最大值和最小值。校准效果直接影响到循迹效果。

校准完成后会显示各个传感器的最小值和最大值。然后串口监视器会显示当前各个传感器的测量数值，数值越大表示颜色越深。最后一列表示黑色的位置，2000 表示黑线在正中，0 表示黑线在小车最左边，400 表示黑线在小车最右边。

如果此程序正常表示传感器没有问题，否则可能不能正常循迹。



红外循迹程序

编译并下载 Infrared-Line-Tracking 程序，打开电源开关，用手拿着小车在黑线中紧贴地面左右摇晃，此阶段为校准，使小车可以设别各个传感器的最大值和最小值。校准效果直接影响到循迹效果。直到小车轮子开始转动时开放小车，小车会沿着黑色线运行。

RGB 彩灯

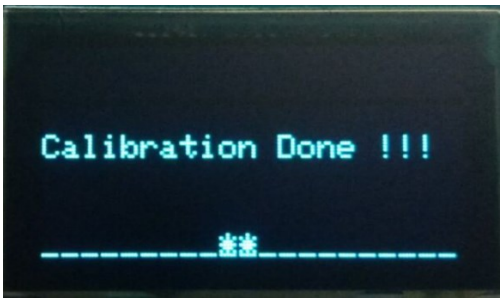
编译并下载 W2812 程序，小车底部四个彩色 LED 灯分别显示红色，绿色，蓝色，黄色。此程序需要将相应的库导入 Arduino IDE。

OLED

编译并下载 OLED 程序，示例程序展示基本的字符显示，绘图操作，包括划线，画圆，矩形框等。此程序需要将相应的库导入 Arduino IDE。

综合程序

编译并下载 Line-Tracking 程序，打开电源开关，OLED 显示 AlphaBot2,此时将小车放在黑线上按下按键，小车会左右转动自动校准，同时 RGB LED 显示绿色。校准完成后 RGB LED 显示蓝色。OLED 显示下右图。”**”表示当前黑线位置，左右移动小车此符号会跟着移动。再次按下按键小车沿黑线走。RGB LED 显示不同颜色，遇到障碍物时小车停止，同时蜂鸣器响。障碍物消失时小车继续运行。将小车拿起来小车也会停止，将小车继续放回到黑线中，小车继续运行。



智能小车走迷宫

编译并下载 MazeSolver 程序，打开电源开关，OLED 显示 AlphaBot2,此时将小车放在黑线上按下按键，小车会左右转动自动校准，同时 RGB LED 显示绿色。校准完成后 RGB LED 显示蓝色。OLED 显示下右图。”**”表示当前黑线位置，左右移动小车此符号会跟着移动。再次按下按键小车沿黑线走。小车遇到直角会自动转向寻找终点，直到终点小车停止。此时将小车拿到起始位置，按下按键，小车会沿着最短线路运行到终点。

关于迷宫制作要求：转向必须为直角，迷宫线路不能有循环。

关于程序：小车检测到有转角时，再减速运行一段时间再次检测是否有转角。由于电机和电压的差异，可能小车不能正常识别转角，此时可以修改程序中速度和延时时间。同理，小车

转角也是更加延时来判断角度大小，如果实际运行过程中转角出错，可以修改程序中转角的速度和延时时间

```
follow_segment();

// Drive straight a bit.  This helps us in case we entered the
// intersection at an angle.
// Note that we are slowing down - this prevents the robot
// from tipping forward too much.
SetSpeeds(30, 30);
delay(40);

// These variables record whether the robot has seen a line to the
// left, straight ahead, and right, whil examining the current
// intersection.
unsigned char found_left = 0;
unsigned char found_straight = 0;
unsigned char found_right = 0;

// Now read the sensors and check the intersection type.
trs.readLine(sensorValues);

// Check for left and right exits.
if (sensorValues[0] > 600)
  found_left = 1;
if (sensorValues[4] > 600)
  found_right = 1;

// Drive straight a bit more - this is enough to line up our
// wheels with the intersection.
SetSpeeds(30, 30);
delay(100);

// Check for a straight exit.
trs.readLine(sensorValues);
if (sensorValues[1] > 600 || sensorValues[2] > 600 || sensorValues[3] > 600)
  found_straight = 1;
```

```
void turn(unsigned char dir)
{
 // if(millis() - lasttime >500)
  {
    switch(dir)
    {
    case 'L':
      // Turn left.
      SetSpeeds(-100, 100);
      delay(190);
      break;
    case 'R':
      // Turn right.
      SetSpeeds(100, -100);
      delay(190);
      break;
    case 'B':
      // Turn around.
      SetSpeeds(100, -100);
      delay(400);
      break;
    case 'S':
      // Don't do anything!
      break;
    }
  }
```

蓝牙控制小车

编译并下载 Bluetooth 程序，下载程序时需要将蓝牙模块拔下，程序下载完成后才插上去否则程序不能下载。

手机扫描下面二维码下载相应的 APP。其中 iOS 使用蓝牙 BLE 与模块进行通信，需要使用 iphone4S 及以上版本。Android 使用蓝牙 EDR 与模块进行通信。

启动 APP，点击“扫描”（注意：手机需要打开蓝牙功能），插上蓝牙模块，大约几秒后，正常在列表中将显示相应的蓝牙设备。如果使用 iOS，选择“Waveshare_BLE”设备，如果使用 Android，选择“Waveshare_ERD”设备。进到下一个页面后选择“遥试模式”。

注意：Android 手机如果支持蓝牙 BLE，将扫描两个设备，注意需要选择 MAC 地址为 00:0E:0E 的设备

选择“遥控模式”，点击设置，根据程序设置命令，设置按键按下时为对应的命令，按键弹起时设置为“Stop”命令。

```
if(strcmp(command, "Forward") == 0)          //Forward
  forward();
else if(strcmp(command, "Backward") == 0)    //Backward
  backward();
else if(strcmp(command, "Left") == 0)        //Left
  left();
else if(strcmp(command, "Right") == 0)       //Right
  right();
else if(strcmp(command, "Stop") == 0)        //Stop
  stop();
else if(strcmp(command, "Low") == 0)         //Low
  Speed = 50;
else if(strcmp(command, "Medium") == 0)      //Medium
  Speed = 150;
else if(strcmp(command, "High") == 0)        //High
  Speed = 250;
```

当速度按键由灰色变为黑色则表示已经链接程序，按下按键可以遥控小车，同时 RGB LED 会亮各种颜色。

关于 **Dual-mode Bluetooth** 更多资料可以参考用户手册

中文：http://www.waveshare.net/wiki/Dual-mode_Bluetooth

英文：http://www.waveshare.com/wiki/Dual-mode_Bluetooth

蓝牙控制小车（JSON 格式命令）

拔下蓝牙模块，编译并下载 Bluetooth-json，启动 APP，插上蓝牙模块，点击“扫描”连接蓝牙。选择”遥控模式” 可以遥控小车，选择”外设控制”，可以控制蜂鸣器响，和 RGB LED 显示不同的颜色。（其他外设小车上没有，不能控制）。

树莓派示例程序

将 AlphaBot2.tar.gz 解压并放在 raspbian 系统的/home/pi 目录下。

在使用本扩展板之前，我们需要给树莓派安装 python 函数库。

如果安装完函数库之后依然无法执行以下的示例程序，那么您可以尝试使用 chmod +x filename 命令给程序赋予可执行权限。

说明：如果是刚开始接触树莓派的朋友可以参考微雪课堂上的《树莓派系列教程》，可选阅

① 树莓派系列教程 2：树莓派烧写镜像

② 树莓派系列教程 3：访问树莓派之 SSH 篇

③ 树莓派系列教程 3：访问树莓派之 WiFi 篇

④ 树莓派教程系列 6：文件共享(samba)

⑤ 树莓派教程系列 7：wiringPi、bcm2835、python 库安装 （只需操作 python 部分）

电机测试

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python AlphaBot2.py
```

预期结果：程序运行后电机转动，小车向前走。

如果方向不对需要相应修改电机接线，或者修改程序中的管脚。建议采用第二种方法。左边轮子转动方向不对则对 ain1，ain2，右边轮子转动方向不对则对调 bin1，bin2。

```
def __init__(self,ain1=12,ain2=13,ena=6,bin1=20,bin2=21,enb=26):
```

五向摇杆

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python Joystick.py
```

预期结果：Joystick 5 个方向的操纵杆，包括上、下、左、右以及确定。当按键按下时，控制端口会显示当前按键的键位，电机会根据按键方向转动，蜂鸣器会响起。如果轮子转动方向和按键方向不一样，需修改一下管脚。

```
pi@raspberrypi:~/AlphaBot2/python $ sudo python Joystick.py
up
right
left
down
left
right
center
```

红外遥控小车

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python IRremote.py
```

预期结果：按下红外遥控器的数字键控制小车。2，8，4，6，5 分别代表前进，后退，左转，右转，停止。按-或+可调节速度。（注：不同的红外遥控器可能按键编码不一样，如果不一样需修改相应程序）

红外避障

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python Infrared_Obstacle_Avoidance.py
```

小车前面没有障碍物时，前面绿色的 LED 灯熄灭，小车遇到障碍物时，前面绿色的 LED 灯会亮。如果 LED 灯不亮或者一直亮可以调节小车底面的两个电位器，使 LED 处于刚好熄灭的状态。

预期结果：小车左边传感器没有到障碍物小车会直走，检测到有障碍物时右转。

超声波测距（树莓派 3B 套餐不包含超声波模块）

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python Ultrasonic_Rangeing.py
```

预期结果：终端会显示当前测量距离。

```
pi@raspberrypi:~/AlphaBot2/python $ sudo python Ultrasonic_Ranging.py
Distance:21.15 cm
Distance:21.45 cm
Distance:21.85 cm
Distance:21.49 cm
Distance:10.95 cm
Distance:7.96 cm
Distance:8.06 cm
```

超声波测距（树莓派 3B 套餐不包含超声波模块）

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python Infrared_Obstacle_Avoidance.py
```

预期结果：小车左边传感器没有到障碍物小车会直走，检测到有障碍物时右转。

RGB 彩灯

安装库：

```
sudo gpt-get install python-pip
sudo pip install rpi_ws281x
```

在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python ws2812.py
```

预期结果：小车底部四个 RGB LED 分别显示红，绿，蓝，黄四种色。

如果显示颜色不对。可以在/boot/config.txt 文件中添加如下两个语句，然后重启生效。

由于 RGB LED 采用 DMA 控制，会树莓派上的音频输出 DMA 通道，添加这两个语句会导

致耳机接口不能用。

```
hdmi_force_hotplug=1
hdmi_force_edid_audio=1
```

循迹传感器测试

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python TRSensors.py
```

预期结果：终端打印五组数据，分别对应五个传感器，当小车放到白色底板上，数值约八九百左右，小车拿到空中，示数约一百几十左右，放到黑线上面，示数约一百多至三百。

红外循迹程序

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python LineFollow.py
```

预期结果：程序运行时会左右转动自动校准，校准完成后左右移动小车，终端会显示黑线当前位置，以及五个传感器的数值。摆正小车，按下五向摇杆中间的按键小车将会沿着黑线运动，同时小车底部 RGB LED 会显示不同的颜色。

舵机

开启 I2C 接口

```
sudo raspi-config
```

选择 Interfacing Options -> I2C -> Yes

进入 Linux 终端，在终端执行以下命令。

```
cd ~/AlphaBot2/python
sudo python PCA9685.py
```

预期结果：舵机会左右移动，转动范围约 180 度。

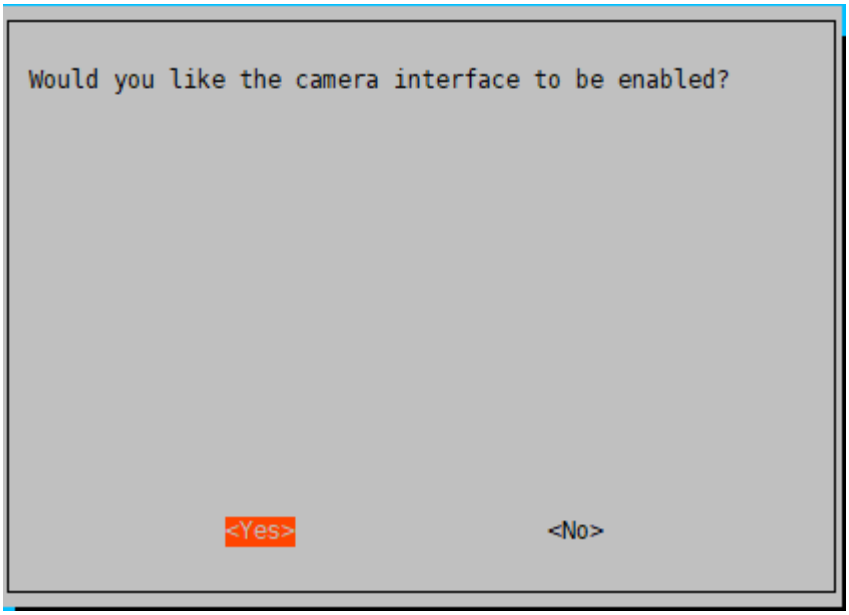
mjpg-streamer 软件实时监控

"MJPG-streamer",是用于从摄像头采集图像,把他们以流的形式通过基于 ip 的网络传输到浏览器。

a) 使用摄像头前必须运行 raspi-config 命令启用摄像头:

```
sudo raspi-config
```

选择 `Enable Camera`, 选择 `YES`



b) 如果是使用 CSI 接口的摄像头，那么系统找不到/dev/video0 的设备节点。需要在 `/etc/modules` 文件中添加一行 `bcm2835-v4l2`:

```
sudo nano /etc/modules
```

加上: `bcm2835-v4l2`

```
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

bcm2835-v4l2
```

（bcm2835-v4l2 中的 l 为 L 的小写，不是 1）

c) 那么系统启动之后，系统会加载这个文件中的模块名。

重启系统：sudo reboot

ls /dev/video*: 下面发现 video0 设备节点。

```
pi@raspberrypi:~/code/AlphaBot2/AlphaBot2 $ ls /dev/video*
/dev/video0  /dev/video10  /dev/video11  /dev/video12
```

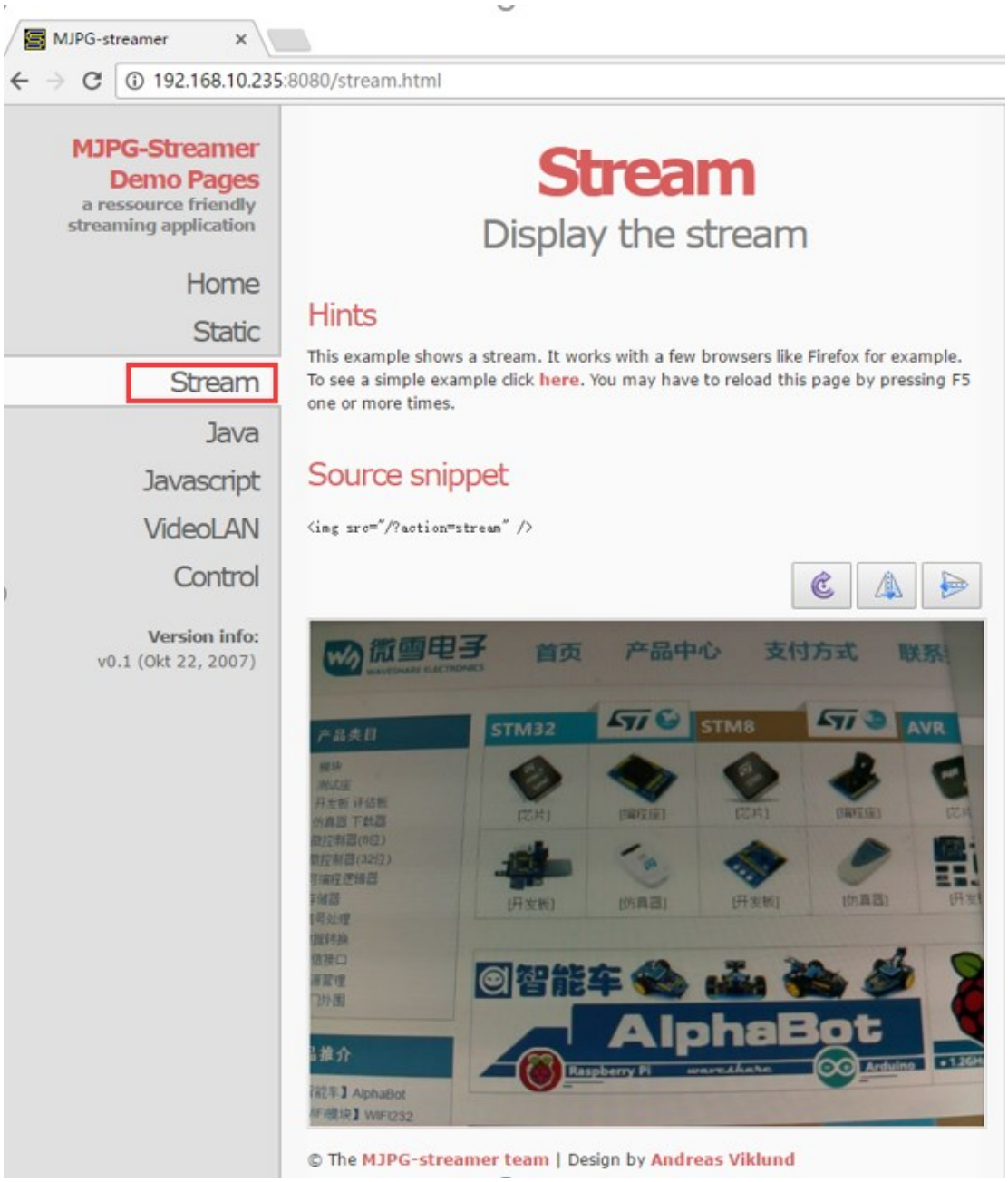
说明：如果执行了 a)、b)、c）之后，且确认操作和指令正确的情况下，仍无法在/dev 下面 发现 video0 设备，可以执行：rpi-update

d) 运行如下命令

```
sudo apt-get update
sudo apt-get install libjpeg8-dev cmake
cd AlphaBot2/mjpg-streamer-master/mjpg-streamer-experimental
sudo make clean all
sudo ./start.sh
```

在谷歌浏览器（其他浏览器可能会无法显示）地址栏内输入树莓派 ip 地址，端口号 8080，点击 Stream 会实时显示摄像头拍摄的内容。

如果你需要了解更多可以查看：https://github.com/jacksonliam/mjpg-streamer



通过 Bottle 实现远程遥控

Bottle 是一个简单高效的遵循 WSGI 的微型 python Web 框架.通过 Bottle 可以快速实现 web 控制。

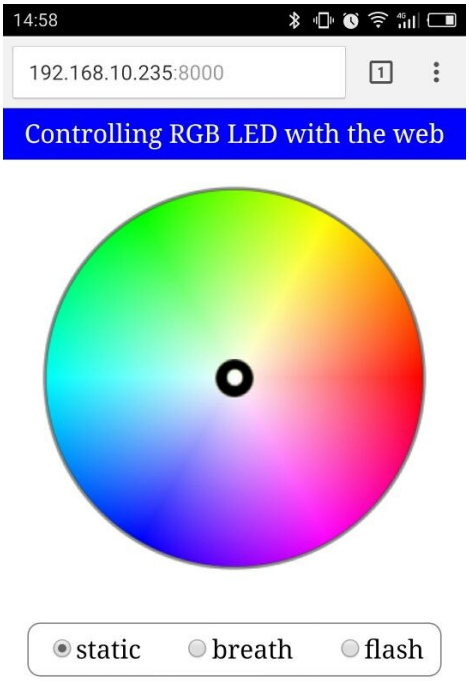
1) 安装库:

```
sudo apt-get install python-bottle
```

2) 通过 Bottle 实现 web 控制 RGB LED。

```
cd ~/AlphaBot2/Web-RGB
sudo python main.py
```

3) 在手机浏览器地址栏（不支持电脑浏览器）内输入树莓派 ip 地址，端口号 8002，会显示如下页面。通过点击图片不同的位置，RGB LED 会显示不同的颜色。"static"，"breath"，"flash" 分别对应三种显示模式。

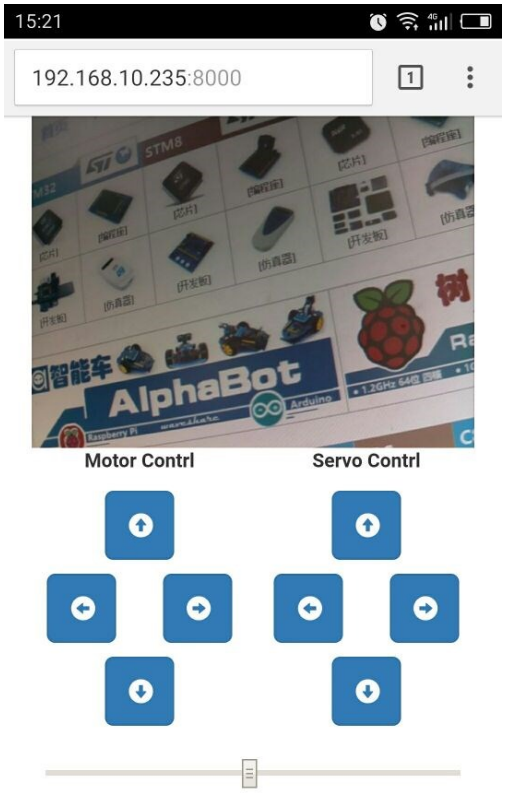


4) 通过 Bottle 实现 web 视频控制小车。

运行如下命令启动程序：

```
cd ~/AlphaBot2/Web-Control
sudo python main.py
```

在浏览器地址栏内输入树莓派 ip 地址，端口号 8000，会显示如下页面，点击按键可以遥控小车。



也可以设置此程序开机启动。用文本编辑器打开/etc/rc.local 文件添加命令。

```
sudo vi /etc/rc.local
```

在注释后面添加如下命令，但是要保证 exit 0 这几行代码在最后，然后保存文件退出。

```
cd /home/pi/AlphaBot2/Web-Contrl
sudo python main.py &
```

由于树莓派是动态获取 IP,可能每次启动时 IP 会变动，可以添加以下语句到 /etc/dhcpcd.comf 设置静态 IP：

```
interface wlan0
static ip_address = 192.168.1.114/24
static routers = 192.168.1.1
```

其中 192.168.1.1 为路由器 IP 地址，192.168.1.114 为要设置的 IP,后面/24 不能去掉。

通过软件远程遥控小车

通过手机端或 PC 端软件视频控制小车。首先修改程序中 IP 地址。

```
cd ~/AlphaBot2/App-Control
vi main.py
```

找到这个一句，将 ip 地址改为树莓派实际地址，端口为视频流端口

```
host = '192.168.10.235'
```

运行如下命令启动程序

```
sudo python main
```

另外开启一个终端，运行 mjpg-streamer.

```
cd /home/pi/AlphaBot2/mjpg-streamer/ mjpg-streamer-experimental
sudo ./start.sh
```

电脑端打开 AlphaBot.exe 程序，输入 IP 地址，mjpg-stream 的端口号，小车 Soket 服务端的端口号，

点击 Viedeo Connet 连接视频，按键变绿色即连接成功，点击 Cmd Connect,连接小车服务端。如果按键变绿则连接成功。左边的按键控制小车移动，右边按键控制摄像头舵机转动。

手机扫描下面二维码下载相应的 APP。

输入树莓派实际地址，点击链接可以看到摄像头拍摄的图像，点击连接，按键由绿色变蓝色后，表示控制端口连接程序，电机对应按键可以控制小车移动，舵机旋转，拉动滑条可以改变小车速度。

用文本编辑器打开/etc/rc.local 文件添加如下命令可以设置开机启动。

```
sudo vi /etc/rc.local
```

在注释后面添加如下命令，但是要保证 exit 0 这几行代码在最后，然后保存文件退出。

```
cd /home/pi/AlphaBot2/App-Contrl
sudo python main.py &
```

每次树莓派启动的时候都需要连接路由器，当然我们也可以开启树莓派无线网络适配器的AP 功能，将树莓派作为路由器。详见树莓派教程，这里不再详述。

通过树莓派蓝牙遥控小车

1) 升级安装蓝牙相关软件包，安装蓝牙依赖库

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get dist-upgrade -y
sudo apt-get install pi-bluetooth bluez bluez-firmware blueman
```

2) 添加 pi 用户到蓝牙组

```
sudo usermod -G Bluetooth -a pi
```

3) 重启树莓派

```
sudo reboot
```

4) 启动/增加 SPP，开启蓝牙设备

```
sudo vi /etc/systemd/system/dbus-org.bluez.service
```

修改文件中这两个语句。

```
1 [Unit]
2 Description=Bluetooth service
3 Documentation=man:bluetoothd(8)
4
5 [Service]
6 Type=dbus
7 BusName=org.bluez
8 ExecStart=/usr/lib/bluetooth/bluetoothd -C
9 ExecStartPost=/usr/bin/sdptool add SP
10 NotifyAccess=main
11 #WatchdogSec=10
12 #Restart=on-failure
13 CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE
14 LimitNPROC=1
15
16 [Install]
17 WantedBy=bluetooth.target
18 Alias=dbus-org.bluez.service
```

重启树莓派后，输入 hciconfig 命令(类似 ifconfig 命令)查看蓝牙服务

```
pi@raspberrypi:~ $ sudo hciconfig
hci0:   Type: BR/EDR  Bus: UART
        BD Address: B8:27:EB:2D:00:87  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:717 acl:0 sco:0 events:42 errors:0
        TX bytes:1532 acl:0 sco:0 commands:42 errors:0
```

如果有看到 hci0 设备则蓝牙已经开启工作。如果没有则没有识别到蓝牙设备。

注意：如果/boot/config.txt 文件中有 dtoverlay=pi3-miniuart-bt 这语句需要注释掉，

否则蓝牙设备不能正常工作。

5) 连接蓝牙设备

输入如下命令进入蓝牙控制界面，(可进入界面输入 help 查看命令列表)

```
sudo bluetoothctl
agent on
```

然后进入界面之后（可进入界面输入 help 查看命令列表）

```
agent on
default-agent
```

scan on 之后 就可以看到扫描得到的蓝牙设备的物理地址了，类似 XX:XX:XX:XX:XX:XX 之

类的地址

```
pi@raspberrypi:~ $ sudo bluetootctl
sudo: bluetootctl: command not found
pi@raspberrypi:~ $ sudo bluetoothctl
[NEW] Controller B8:27:EB:2D:00:87 raspberrypi [default]
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:2D:00:87 Discovering: yes
[NEW] Device 18:E2:9F:AC:E3:3C vivo X7
[NEW] Device A4:44:D1:E7:25:B2 A4-44-D1-E7-25-B2
[CHG] Device A4:44:D1:E7:25:B2 RSSI: -50
[CHG] Device A4:44:D1:E7:25:B2 Name: 魅蓝 Note5
[CHG] Device A4:44:D1:E7:25:B2 Alias: 魅蓝 Note5
```

配对使用以下命令（以下请把 XX:XX:XX:XX:XX:XX 替换为你自己设备的地址，Android 手

机可以在设置->关于手机->状态信息中看到蓝牙地址。）

```
pair XX:XX:XX:XX:XX:XX
```

```
[bluetooth]# pair A4:44:D1:E7:25:B2
Attempting to pair with A4:44:D1:E7:25:B2
[CHG] Device A4:44:D1:E7:25:B2 Connected: yes
```

```
[bluetooth]# pair 74:A5:28:F3:D0:31
Attempting to pair with 74:A5:28:F3:D0:31
[CHG] Device 74:A5:28:F3:D0:31 Connected: yes
Request confirmation
[agent] Confirm passkey 777190 (yes/no): yes
[CHG] Device 74:A5:28:F3:D0:31 Modalias: bluetooth:v000Fp1200d1436
[CHG] Device 74:A5:28:F3:D0:31 UUIDs:
        00001105-0000-1000-8000-00805f9b34fb
        0000110a-0000-1000-8000-00805f9b34fb
        0000110c-0000-1000-8000-00805f9b34fb
        0000110e-0000-1000-8000-00805f9b34fb
        00001112-0000-1000-8000-00805f9b34fb
        00001115-0000-1000-8000-00805f9b34fb
        00001116-0000-1000-8000-00805f9b34fb
        0000111f-0000-1000-8000-00805f9b34fb
        0000112f-0000-1000-8000-00805f9b34fb
        00001132-0000-1000-8000-00805f9b34fb
        00001200-0000-1000-8000-00805f9b34fb
        00001800-0000-1000-8000-00805f9b34fb
        00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 74:A5:28:F3:D0:31 Paired: yes
Pairing successful
```

配对之后把设备添加到信任列表

```
trust XX:XX:XX:XX:XX:XX
```

```
[bluetooth]# trust A4:44:D1:E7:25:B2
[CHG] Device A4:44:D1:E7:25:B2 Trusted: yes
Changing A4:44:D1:E7:25:B2 trust succeeded
```

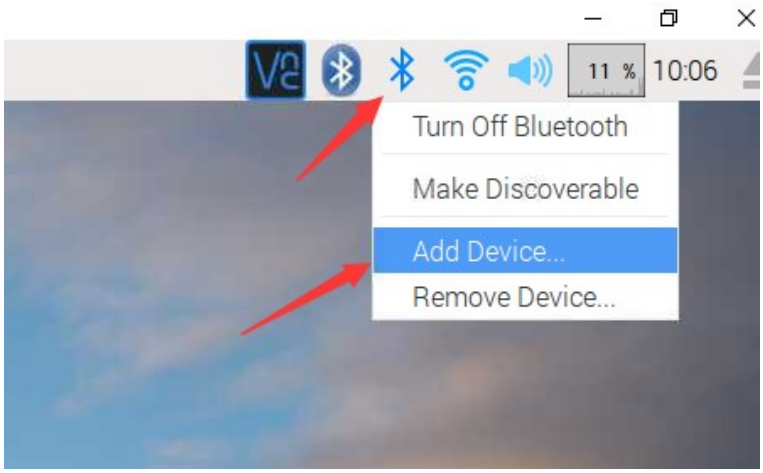
之后连接设备

```
connect XX:XX:XX:XX:XX:XX
```

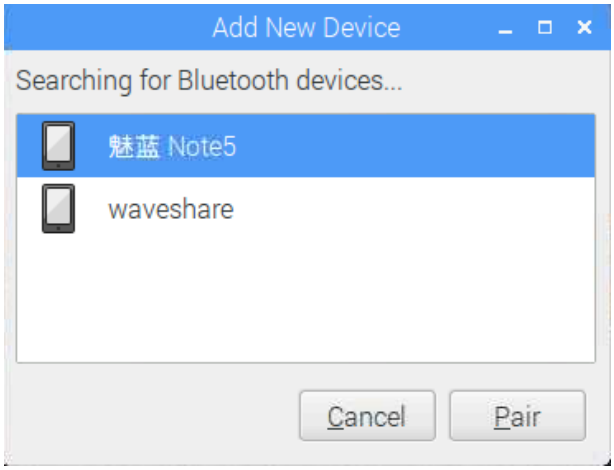
```
[bluetooth]# connect A4:44:D1:E7:25:B2
Attempting to connect to A4:44:D1:E7:25:B2
[CHG] Device A4:44:D1:E7:25:B2 Connected: yes
[CHG] Device A4:44:D1:E7:25:B2 Modalias: bluetooth:v000Fp1200d1436
[CHG] Device A4:44:D1:E7:25:B2 UUIDs:
        00001105-0000-1000-8000-00805f9b34fb
        0000110a-0000-1000-8000-00805f9b34fb
        0000110c-0000-1000-8000-00805f9b34fb
```

最后 quit 退出就可以了，以后蓝牙设备开机后树莓派会自动进行连接。

注意：如果命令行下链接不成功可以在图形界面下连接。

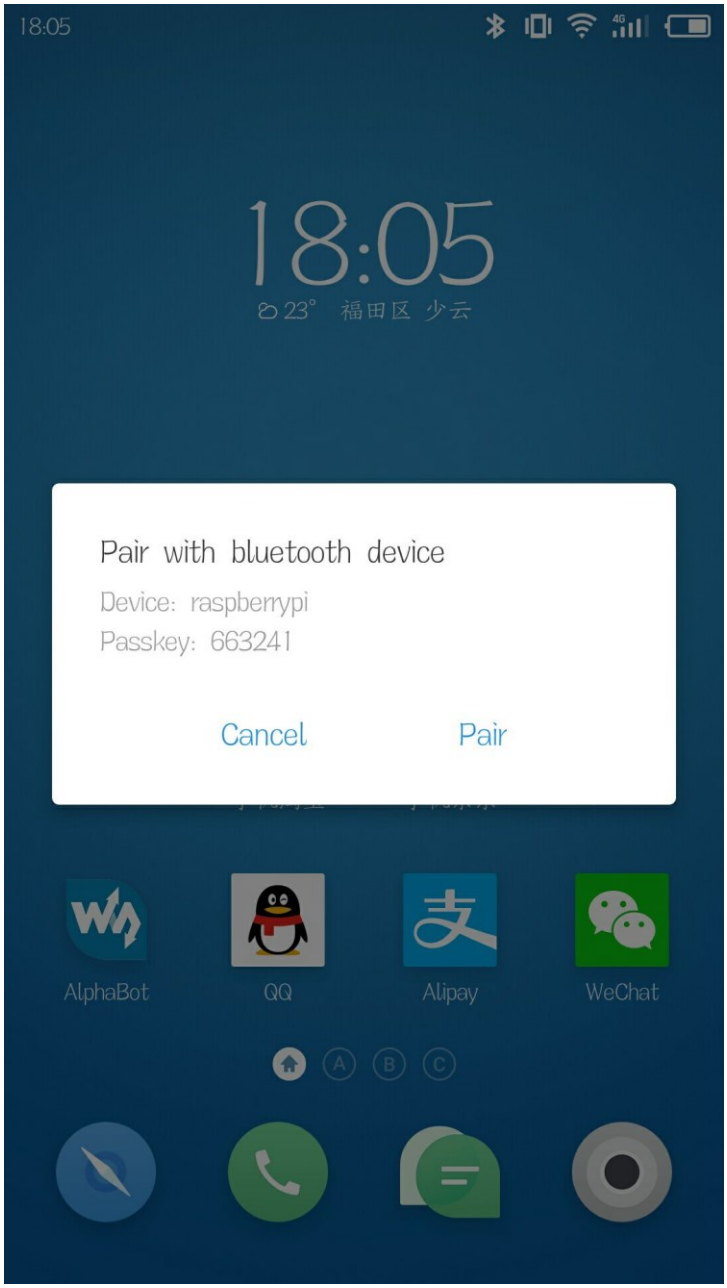


点击蓝牙图标，添加蓝牙设备。

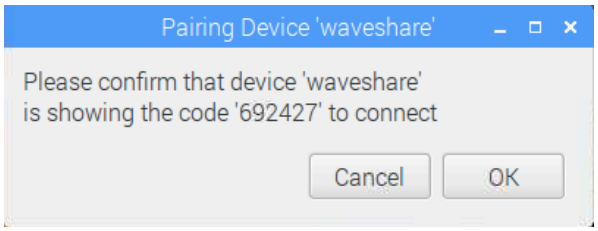


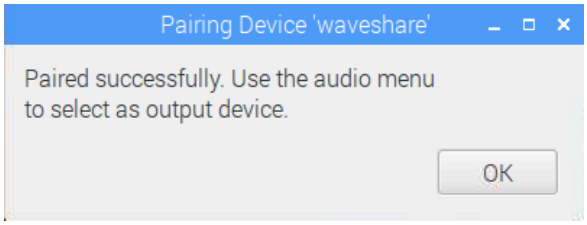
选择对应的手机蓝牙设备，注意手机蓝牙要开启可发现。

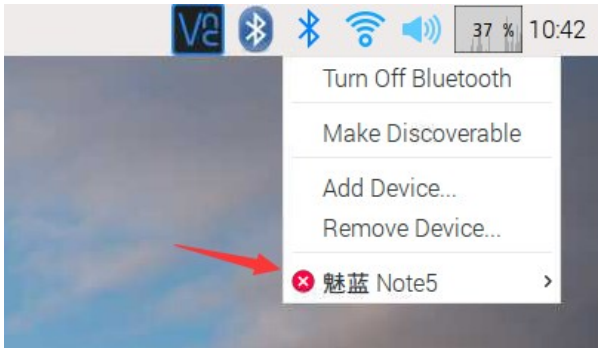
树莓派端选择 ok 配对，手机端也选择“Pair”配对。



提示配置成功，但是没有服务。







已经连接成功。

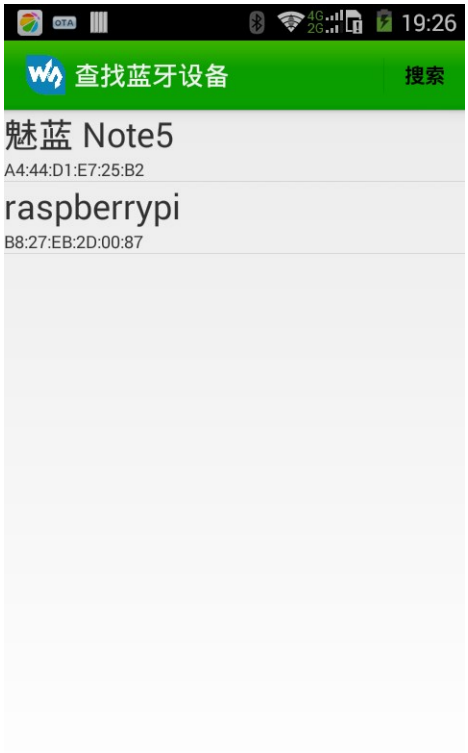
运行如下命令启动程序：

```
cd /AlphaBot2/Bluetooth-Control
sudo rfcomm watch hci0 1 python bluetooth.py
```

手机扫描下面二维码下载相应的 APP。(此软件为 Dual-mode Bluetooth 模块控制软件，

如果上面 Arduino 操作中已经安装此软件，则不需再次安装。)

启动 APP，点击“扫描”（注意：手机需要打开蓝牙功能），正常在列表中将显示相应的蓝牙设备。选择“raspberrypi”设备。进到下一个页面后选择“遥控模式”进入遥控模式。



如果没有搜索到树莓派蓝牙，可以运行如下命令开启蓝牙可发现。

```
pi@raspberrypi:~ $ sudo bluetoothctl
[NEW] Controller B8:27:EB:2D:00:87 raspberrypi [default]
[NEW] Device A4:44:D1:E7:25:B2 魅蓝 Note5
[NEW] Device 74:A5:28:F3:D0:31 PE-TL20
[NEW] Device F6:D0:10:0B:0E:2C waveshare
[bluetooth]# discoverable on
Changing discoverable on succeeded
[CHG] Controller B8:27:EB:2D:00:87 Discoverable: yes
[bluetooth]# quit
[DEL] Controller B8:27:EB:2D:00:87 raspberrypi [default]
```

常见问题解答

问题 1：相对于 AlphaBot 一代，AlphaBot2 有那些改进

答：

- AlphaBot2 尺寸更小，各种传感器高度集成，结构更加稳定。
- 采用高精度的全金属电机，噪声低精度高。
- 采用 TB6612FNG 电机驱动芯片，相对 L298P 高效，体积小，不发热。
- 添加按键，蜂鸣器，酷炫彩灯，usb 转串口等外设，更加方便调试。
- AlphaBot2-Ar 适配板增加 OLED ,AlphaBot2-PiZero 适配板增加 usb Hub
- 增加 PCA9685 舵机控制芯片，避免树莓派软件软件 PWM 控制舵机出现抖动现象。
- 程序更加丰富，树莓派增加安卓 App 视频控制，电脑端 Qt 软件控制。还增加树莓派板载蓝牙遥控。

问题 2：Alphabot2 系列智能车配什么电池?

答：配 14500 型电池，即直接为 14mm,长度为 50mm.单节标的电压一般为 3.7V，充满电电压一般为 4.2V，最低电压不能低于 2.75V，否则可能导致电池永久损坏。这种电池外形和 5 号电池大小一样，但是不能用 5 号电池代替。5 号电池电压不够。

问题 3：小车红外避障不正常是什么原因?

答:首先检查小车前面的红外检测模块是否正常，正常情况下两个绿色的 LED 灯是不亮的，检测前方有物体时才会亮。如果 LED 灯一直亮或者一直不亮，可以调节小车底部的电位器，使没有障碍物时 LED 刚好熄灭。

问题 4：小车转动方向不对。

答：电机转动方向不对时，可以通过修改程序中管脚设置改正。

问题 5：舵机转动的幅度范围不正。

答：舵机可以转动大约 180 度，正常情况下可以左右转动 90 度，如果左右两边转动的幅度相差太大时，可以重新组装调整。

问题 6：怎样确定电池是否有电。

答：电池电量过低时小车会变慢。一般情况下，树莓派关机断线不能连接时则电池需要充电了。电池电压过低时不应在继续使用，否则电池会永久损坏不能用了。

问题 7：小车基板可以用其他单片机控制吗。

答：可以，我们有提供原理图以及尺寸图，用户可以进行二次开发。

安全须知

注意

请保留此信息以供将来参考

警告

AlphaBot2 是精密器件，为避免 AlphaBot2 出现故障或者损坏，请遵守以下要求：

- 本产品通常使用两节 14500 电池供电，工作电压是 7.4 V。底板与 Arduino 和树莓派搭配使用时的任何外部电源，都应遵守所在地区适用的电源相关法规和标准。
- 本产品应在通风良好的环境下使用，以便电源散热以及保证产品性能。
- 底板拆卸下来之后，应该放置在稳定、平坦、不导电的表面上。不应接触导电物品。
- 将不兼容的设备连接到底板可能会导致设备损坏、质保失效。
- 切勿在操作过程中接触水分和湿气，或放置在导电的表面上。
- 切勿让 AlphaBot2 接触任何热源，以便该产品在正常室温环境下保持可靠运行。
- 操作过程应避免对印制电路板和其上的电子元器件造成机械或者电器损伤。
- 操作过程应避免触碰已通电的印制电路板，尽可能触碰其边缘，减少静电放电风险。