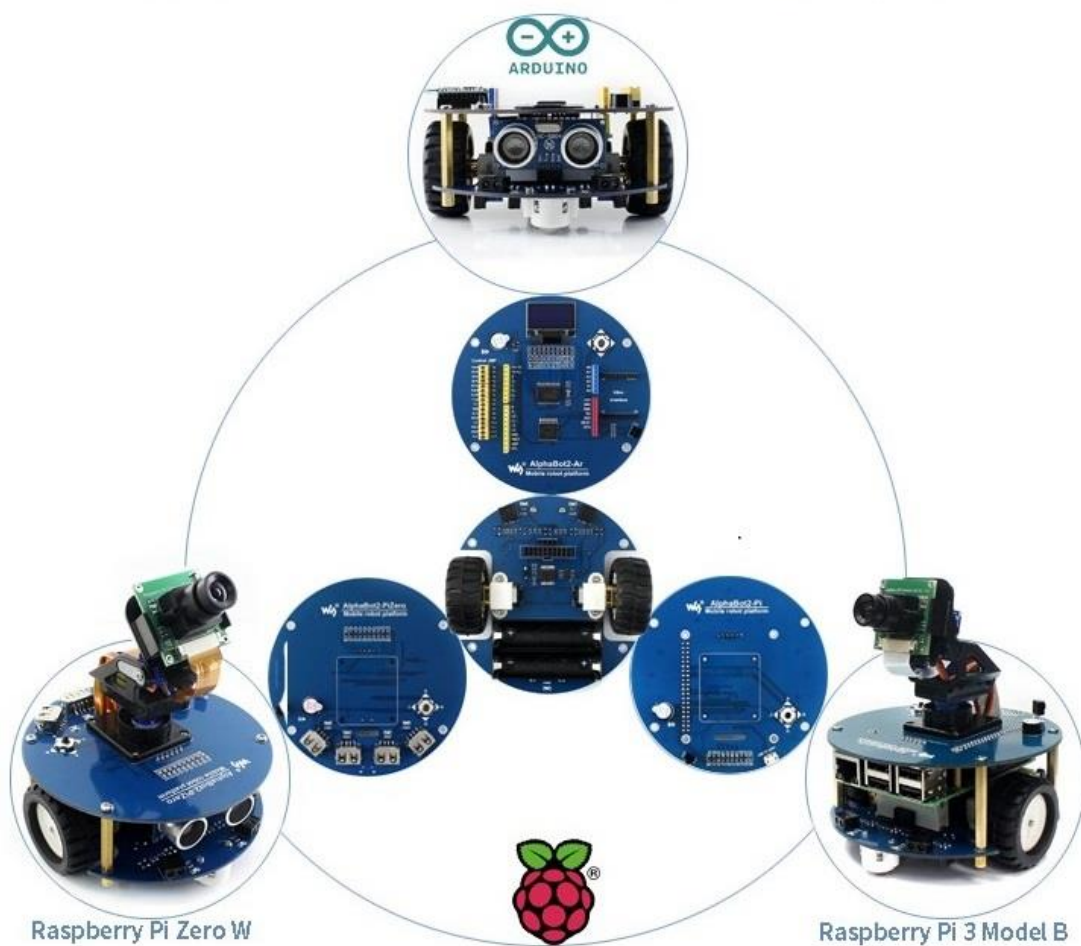# AlphaBot 2
User Manual

August 2017

V1.3

To the Reader

Selection Guide

Note: Accessories may be different in different product package. This manual describes all the compatible accessories but perhaps the product you bought doesn't contain some of them. The figure below shows most the compatible accessories:

Package Contents		Products				
Item	Description	AlphaBot2-Ar Acc Pack	AlphaBot2-Ar	AlphaBot2-Pi Acc Pack	AlphaBot2-Pi	AlphaBot2-PiZero Acc Pack
AlphaBot2-Base	Motor driver, integrates sensors for obstacle avoiding, line tracking	√	√	√	√	√
AlphaBot2-Ar	Adapter board, for connecting Arduino	√	√			
AlphaBot2-Pi	Adapter board, for connecting RPi3 B			√	√	
AlphaBot2-PiZero	Adapter board, for connecting RPi Zero W					√
RPi3 B	Raspberry Pi 3 Model B				√	
UNO PLUS	Enhanced Arduino compatible board		√			
Dual-mode Bluetooth	Dual-mode Bluetooth module		√			
Ultrasonic sensor	Ultrasonic obstacle avoiding, ranging	√	√			√
IR remote controller	remotely control the robot	√	√	√	√	√
RPi Camera (B)	Raspberry Pi camera, adjustable focus			√	√	√
SG90	Servo, working with the pan head, controlling the rotation of the camera			√	√	√
Micro SD Card 16GB	16GB Miro SD Card, class 10			√	√	√
5V 2.5A Power Adapter	RPi3 B requires 2.5A or above power supply			√	√	√

Note:

AlphaBot2-Ar Acc Pack requires an Arduino compatible controller, if you don't have one in hand, please consider the all-in-one kits: AlphaBot2-Ar.

AlphaBot2-Pi Acc Pack requires the controller Raspberry Pi, if you don't have one in hand, please consider the all-in-one kit: AlphaBot2-Pi.

AlphaBot2-PiZero Acc Pack requires the controller Raspberry Pi Zero/Zero W, which should be purchased separately.

Due to the structure design, ultrasonic sensor module can NOT be used with the AlphaBot2-Pi kit.

Recommendation

Reading this document can be a challenge for a freshman. We advise that the reader need to have some knowledge of the programming in Raspberry Pi Python or Arduino, which can be helpful for understanding the programming fundamentals. Due to space constraint, only a few of the programming fundamentals can be explored here. If you want to have a thorough understand of them, please check the source code. Here, we will describe more about the practical experience of smart robot development.

Raspberry Pi tutorial:

http://www.waveshare.com/wiki/Raspberry_Pi_Tutorial_Series

Due to the electrical characteristic limitation, user should be more careful to use this product in a safe and correct matter, to prevent the Raspberry Pi board or Arduino board from irreversible damage by the smart robot chassis. For your first time in using this product, please follow the steps described in the document.

Get details

Waveshare Wiki provides detailed data of AlphaBot2, including user manual, schematic diagram, Demos, data sheet and the likes. Users can apply the product in a safe and correct matter by reading them.

☺ Resource Link: www.waveshare.com/wiki/AlphaBot2

Contents

What's on board?

Introduction

AlphaBot2 robot kits includes a chassis (AlphaBot2-Base chassis) and an adapter board (one of the adapter boards AlphaBot2-Ar / AlphaBot2-Pi / AlphaBot2-PiZero). The robot supports Arduino / Raspberry Pi 3 Model B / Raspberry Pi Zero W with different adapter boards.

It features rich common robot functions including line tracking, obstacle avoiding, Bluetooth / infrared / WiFi remote control, video monitoring, etc.

Thanks to the highly integrated modular design, it is fairly easy to assemble by a snap, no soldering, no wiring. After a few minutes spent on hardware assembling, you're almost there, our open source demo codes is ready to help you get started fast.

Accessories may be different in different product package. This manual describes all the compatible accessories but perhaps the product you bought doesn't contain some of them.
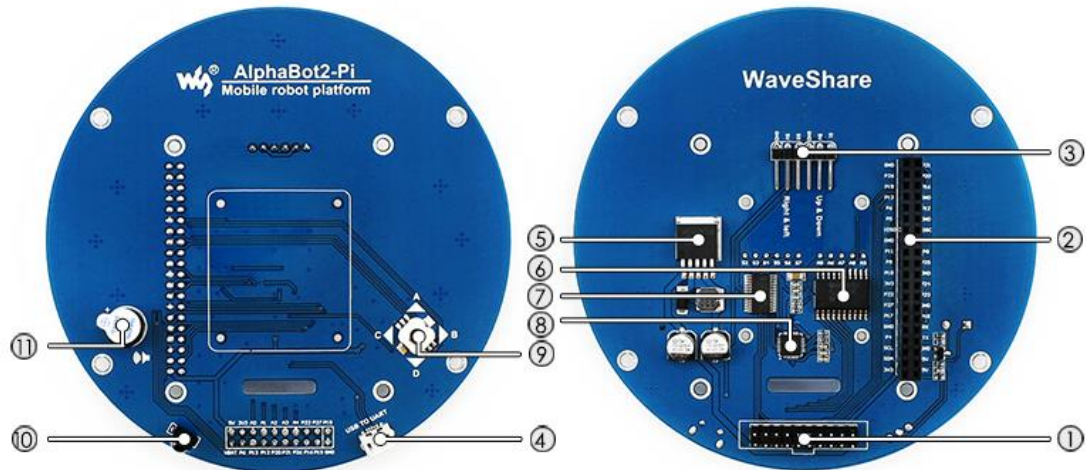
AlphaBot2-Base
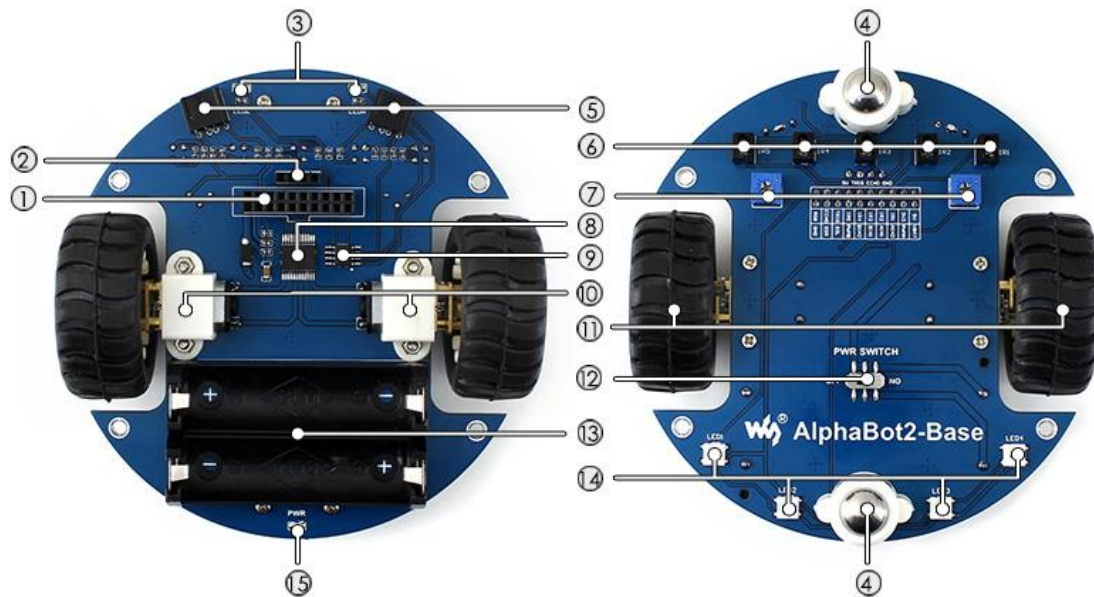
1. **AlphaBot2 control interface:** for connecting sorts of controller adapter board
2. **Ultrasonic module interface**
3. **Obstacle avoiding indicators**
4. **Omni-direction wheel**
5. **ST188**: reflective infrared photoelectric sensor, for obstacle avoiding
6. **ITR20001/T**: reflective infrared photoelectric sensor, for line tracking
7. **Potentiometer** for adjusting obstacle avoiding range
8. **TB6612FNG** dual H-bridge motor driver
9. **LM393** voltage comparator
10. **N20 micro gear motor** reduction rate 1:30, 6V/600RPM
11. **Rubber wheels** diameter 42mm, width 19mm
12. **Power switch**
13. **Battery holder**: supports 14500 batteries
14. **WS2812B:** true color RGB LEDs
15. **Power indicator**

AlphaBot2-Ar
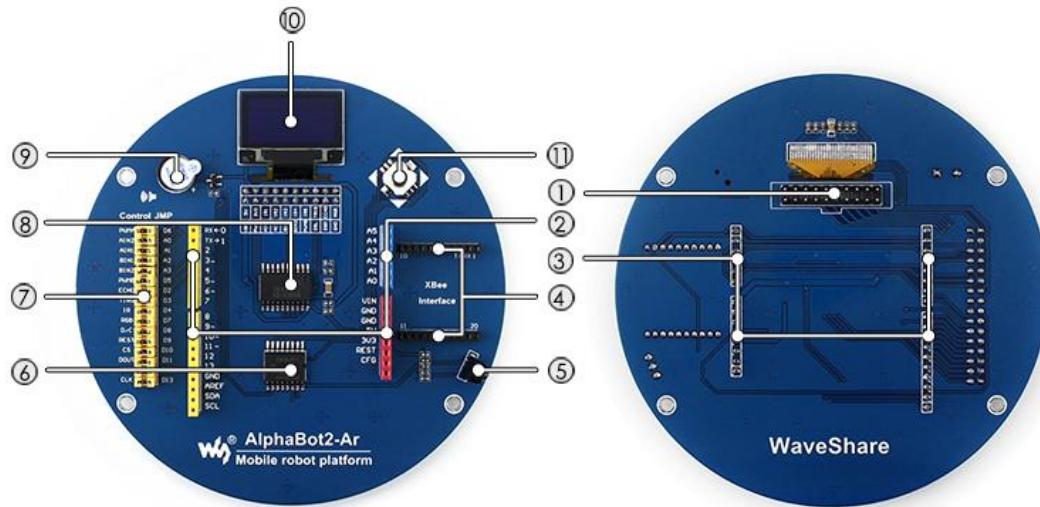
1. **AlphaBot2 control interface**: for connecting AlphaBot2-Base
2. **Arduino expansion header**: for connecting Arduino shields
3. **Arduino interface**: for connecting Arduino compatible controller
4. **Xbee connector**: for connecting dual-mode Bluetooth module, remotely control the robot via Bluetooth
5. **IR receiver**
6. **PC8574**: I/O expander, SPI interface
7. **Arduino peripheral jumpers**
8. **TLC1543**: 10-bit AD acquisition chip
9. **Buzzer**
10. **0.96inch OLED** SSD1306 driver, 128x64 resolution
11. **Joystick**

AlphaBot2-Pi

1. **AlphaBot2 control interface**: for connecting AlphaBot2-Base
2. **Raspberry Pi interface**: for connecting Raspberry Pi 3 Model B
3. **Servo interface**
4. **USB TO UART**: easy for controlling the Pi via UART
5. **LM2596**: 5V voltage regulator
6. **TLC1543**: 10-bit AD acquisition chip, allows the Pi to use analog sensors
7. **PCA9685**: servo controller, make it more smoothly to rotate the pan head
8. **CP2102**: USB TO UART converter
9. **Joystick**
10. **IR receiver**
11. **Buzzer**

AlphaBot2-PiZero
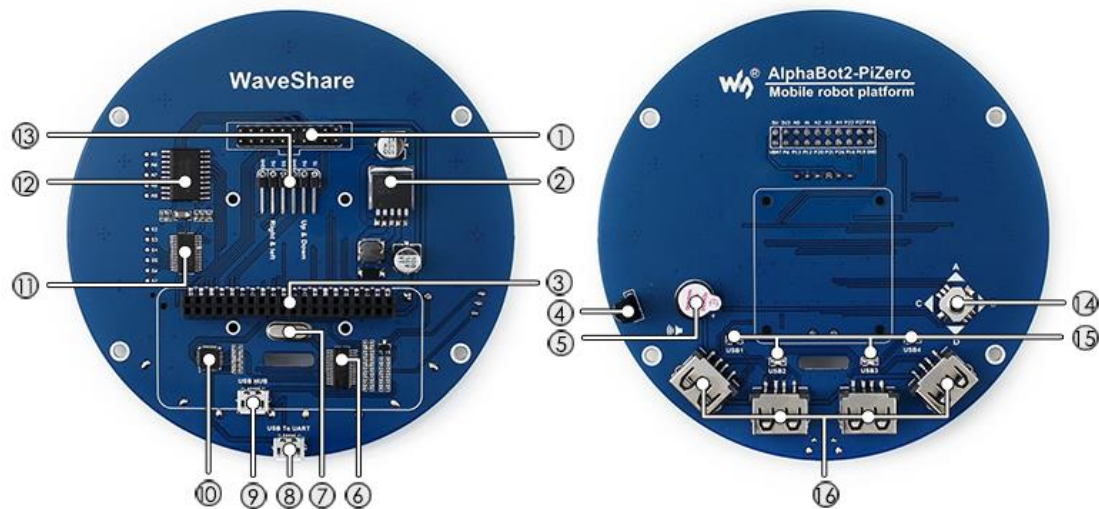
1. **AlphaBot2 control interface**: for connecting AlphaBot2-Base
2. **LM2596**: 5V voltage regulator
3. **Raspberry Pi interface**: for connecting Raspberry Pi Zero/Zero W
4. **IR receiver**
5. **Buzzer**
6. **FE1.1S**: USB HUB chip
7. **12M crystal**
8. **USB TO UART**: easy for controlling the Pi via UART
9. **USB HUB interface**: extends the USB port of Raspberry Pi Zero/Zero W
10. **CP2102**: USB TO UART converter
11. **PCA9685**: servo controller, make it more smoothly to rotate the pan head
12. **TLC1543**: 10-bit AD acquisition chip, allows the Pi to use analog sensors
13. **Servo interface**
14. **USB indicators**
15. **USB ports**: more USB capability

Quick start

We have provided the control programs for Arduino and Raspberry Pi, which can implement many functions, such as tracking, obstacle avoidance, remote control, video monitoring and so on.

You can get the example programs described in the following sections from the website of Waveshare Wiki. To apply these examples to your application, you need to copy the file libraries under the directory of Arduino into the folder libraries under the installation directory of Arduino IDE. Then, restart the Ariduino IDE, and click File-> Example to see if there is an option TRSensors. If yes, it means the libraries are imported successfully. When using the Raspberry Pi, you should extract the example programs and place them in the directory of /home/pi under the operation system of Raspbian.

Arduino examples

Motor testing

Compile and download the program Run-Test. Normally, the smart robot moves forward when the program runs and drives the motor to rotate. In case that the direction of motion is wrong, you may need to modify the motor wring or change the pin settings in your program. We recommend the latter one. If left wheels of the robot run in a wrong direction, you should exchange the settings of AIN1 and AIN2. For right wheels, exchange the settings of BIN1 and BIN2.

In the event that all the wheels run in a wrong direction, you should modify the

settings in the demos, like:

```
#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)
```

5-way Joystick

Compile and download the program Joystick. This Joystick has 5 control directions: Up, Down, Left, Right and Center (Enter). Open the serial monitor, and set the baud rate to 115200. Then, the monitor displays the current key press. When there is a key pressed down, the buzzer sounds and the motor rotates in the direction according to the key. If the wheel(s) runs in the wrong direction, you should modify the settings of the corresponding pins.
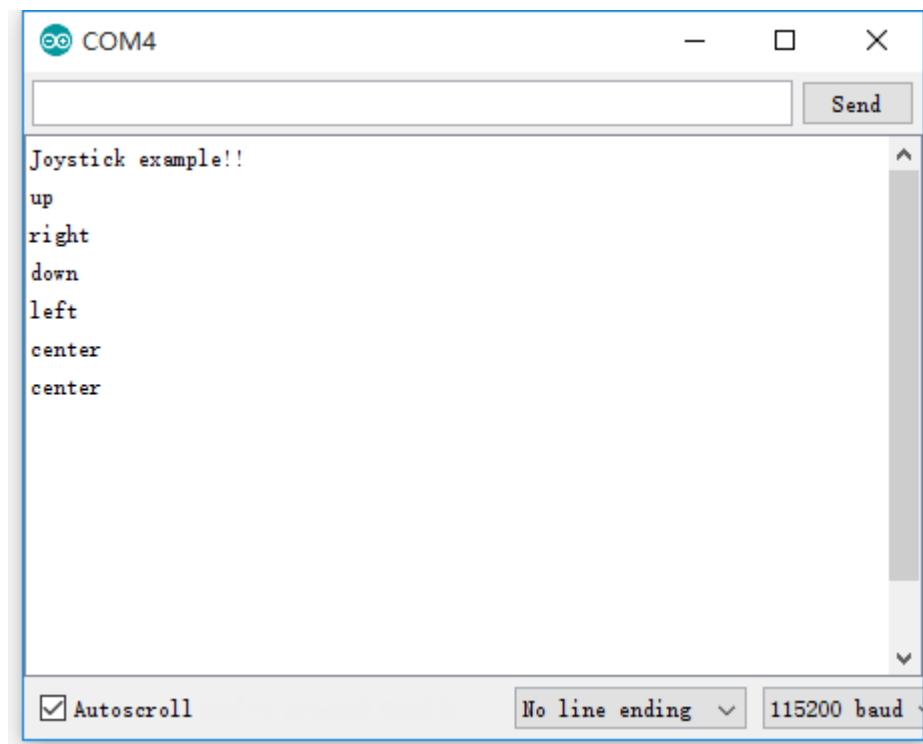
Infrared remote control

Compile and download the program IR. You can control the smart robot by pressing down the numeric keys on the infrared remote controller, in which, "2" is for go forwards, "8" for go backwards, "4" for turn left, "6" for turn right, and "5" for stop. Keys "-" and "+" can adjust the speed of the robot, and "EQ" can restore the default setting of the speed.

The key encoding may change with different Infrared remote controller. You can modify the program as required.

Infrared obstacle avoidance

Compile and download the program Infrared-Obstacle-Avoidance. When there is no obstacle ahead of the robot, the green LEDs on the robot are off. When the obstacle is detected, the green LEDs will light up. In case that the LEDs are always off or always on, you can try to adjusted the potentiometers on the bottom of the robot to make the LEDs work properly.

In this example, the robot goes straight when there is no obstacle, and turns right when obstacle is detected.

Ultrasonic distance measurement

Compile and download the program Ultrasonic_Ranging. Open the serial monitor, and set the baud rate to 115200. Then, the monitor displays the current measured distance.
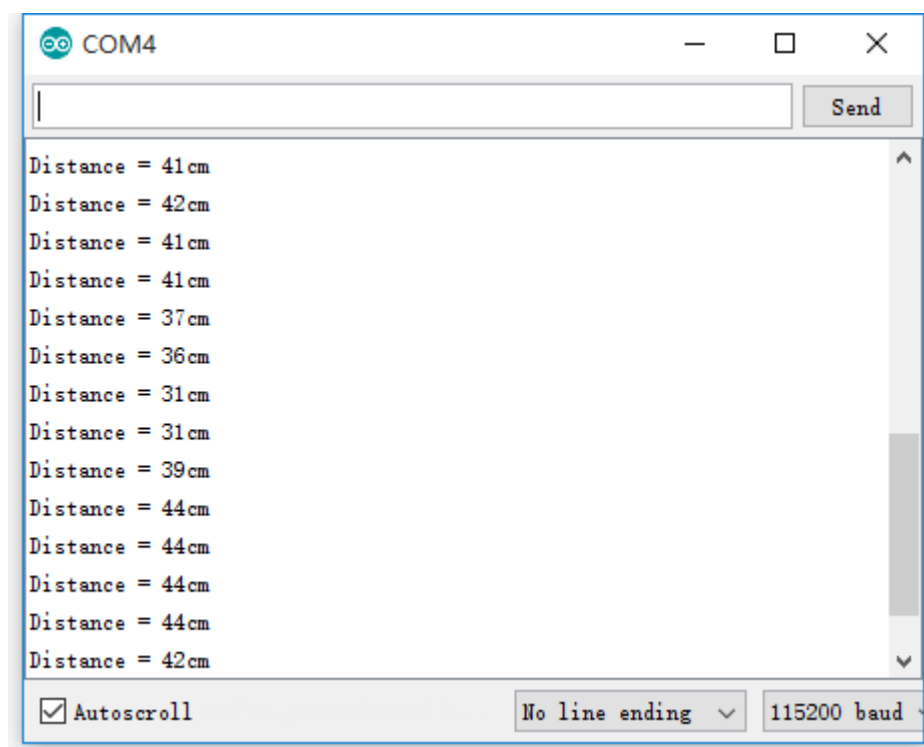
Ultrasonic obstacle avoidance

Compile and download the program Ultrasonic-Obstacle-Avoidance. In this example, the robot goes straight when there is no obstacle, and turns right when obstacle is detected. Since the ultrasonic can be reflected, if the obstacle is not in the front of the robot, but is at an angle with the robot, the distance measured between the robot and the obstacle may be inaccurate.

Ultrasonic infrared obstacle avoidance

Compile and download the program Ultrasonic-Infrared-Obstacle-Avoidance. In this example, the robot goes straight when there is no obstacle, and turns right when obstacle is detected.
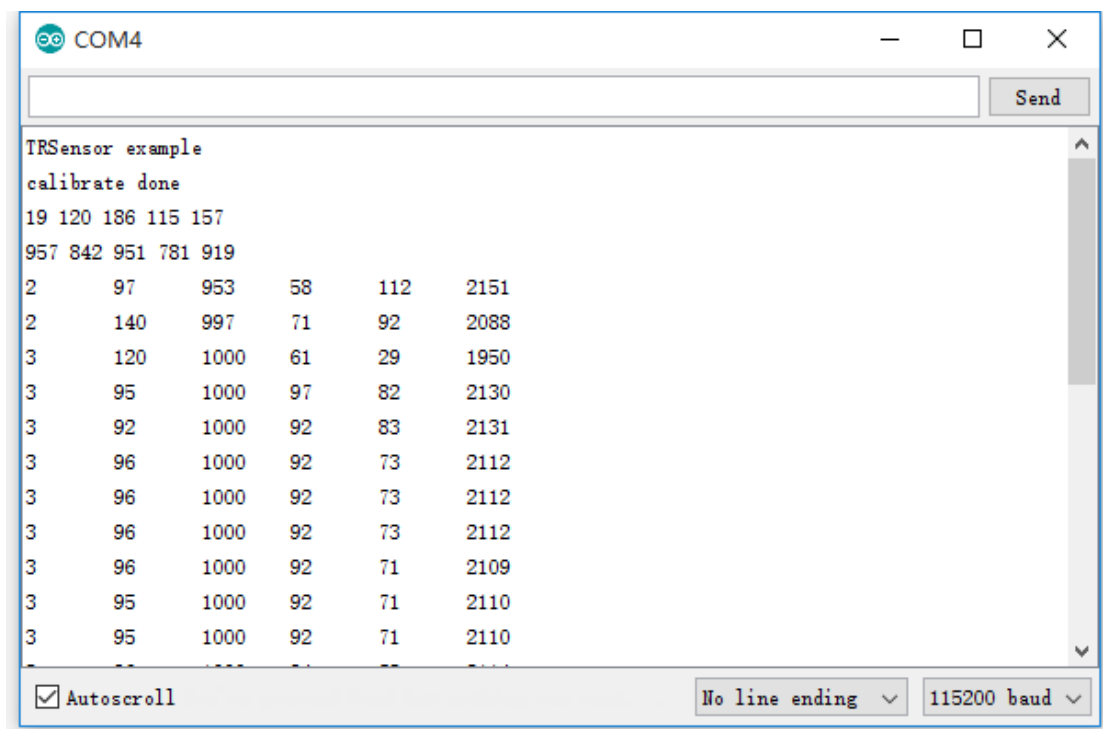
Tracker sensor testing

Import the library TRSensors to the file libraries under the installation directory of Arduino IDE. Then, click File -> Example -> TRSensors ->TRSensorExample to open the project. Compile and download the program to the development board Arduino. The tracker sensor can detect the black line in the background color of white. You can choose a black adhesive tape with 15cm width, and paste it to a white KT board to create the smart robot tracking line scenario. The dark background color will reduce the capability of the smart robot on the tracking line.

Open the serial monitor, and set the baud rate to 115200. When the monitor displays the characters "TRSensor example", you should hold the robot in the middle of the black line and close to the ground, then swing the robot from side to side. Then, the smart robot goes into the calibration mode, identifying the Max. and the Min. threshed of each sensor. Calibration can have a direct influence to the tracking line capability of the robot.

After the calibration, the monitor shows the Max. and the Min. threshed of each sensor. And then, the measured values of sensors are displayed as well. The larger

value means the deeper color. The last column of values indicate the position of the black line, in which "2000" means the black line is in the middle of the robot, "0" means the black line is on the leftmost side of the robot, and "4000" means the black line is on the rightmost side of the robot.

When this program runs well, the line tracking sensor works properly. Or else, the sensor cannot perform line tracking.

```
COM4                                              —   □   ×

                                                       Send

TRSensor example
calibrate done
19 120 186 115 157
957 842 951 781 919
2       97      953     58      112     2151
2       140     997     71      92      2088
3       120     1000    61      29      1950
3       95      1000    97      82      2130
3       92      1000    92      83      2131
3       96      1000    92      73      2112
3       96      1000    92      73      2112
3       96      1000    92      73      2112
3       96      1000    92      71      2109
3       95      1000    92      71      2110
3       95      1000    92      71      2110

☑ Autoscroll              No line ending ∨    115200 baud ∨
```

Infrared line tracking

Compile and download the program Infrared-Line-Tracking. Turn on the power adapter, and hold the robot in the middle of the black line and close to the ground, then swing the robot from side to side. The smart robot goes into the calibration mode, identifying the Max. and the Min. threshed of each sensor. Calibration can have a direct influence to the tracking line capability of the robot. When the wheels

of the robot begin to move, you can release the robot and it will run along with the

black line.

RGB LEDs

Compile and download the program W2812. There are 4 RGBn LEDs (red, green,

blue and yellow, respectively) on the bottom of the robot.

To run this program, you should import the corresponding library to Arduino IDE.

OLED

Compile and download the program OLED. The example program demonstrates the

basic functions of character display and drawing operation, including drawing line,

cycle, rectangle box and the likes. To run this program, you should import the

corresponding library to Arduino IDE.

Comprehensive program

Compile and download the program Line-Tracking. Turned on the power adapter,

you can see the OLED display the characters "AlphaBot2". Then, place the robot

on the black line and press down the key. Now, the smart robot goes into the

calibration mode, moving to left and then right. And the RGB LED is in green. When

the calibration finished, the RGB LED is in blue. OLED shows the information as the

figure below on the right, in which "**" indicates the current position of the black

line. When the smart robot moves to left or right, this symbol will be moved along.

Pressing down the key again, the smart robot runs along with the black line, and

the RGB LED shows different colors. When there is an obstacle detected, the smart

robot stops and the buzzer sounds. When the obstacle disappears, the smart robot

continue to run. If you take up the robot, it will stop as well. Then, place it on the

black line, it will run again.

Smart robot running mazes

Compile and download the program MazeSolver. Turned on the power adapter, you

can see the OLED display the characters "AlphaBot2". Then, place the robot on

the black line and press down the key. Now, the smart robot goes into the

calibration mode, moving to left and then right. And the RGB LED is in green. When

the calibration finished, the RGB LED is in blue. OLED shows the information as the

figure below on the right, in which "**" indicates the current position of the black

line. When the smart robot moves to left or right, this symbol will be moved along.

If the smart robot comes across the right angle, it will take a turn and look for the

end automatically. The smart robot stops when it arrives the end. Now, put the

smart robot to the start again, and press down the key, you will find that the smart

robot runs to the end along with the shortest route.

Here are some requirements in making the maze: the corner should be a right angle, and the maze should not has any loop.

About the program: when the smart robot detects any corner, it will slow down and detects if there is a corner once again. As the difference at motor and operation voltage in used, the smart robot may be unable to identify the corner. In this case, you can modify the speed setting and delay time in the program. Further, when the smart robot has to turn a corner, it need to increase delay time to judge the degree of the angle. Therefore, if there is any problem in turning a corner, you can modify the speed setting and delay time in the program as well.

```
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;
    }
  }
}
```

Bluetooth control smart robot

Compile and download the program Bluetooth. You should remove the Bluetooth module from the smart robot before downloading, or else the program cannot be loaded. When finished, you should insert the Bluetooth module to the smart robot again.

Then, scan the QR code below with your mobile phone, and download the relative APP according to your mobile phone OS. For iOS, Bluetooth BLE is applied to

communicate with the module, and iphone4S or above is required. For Android,

Bluetooth EDR is applied.

Startup the APP, and click the option "Scan" (Note: Turn on the Bluetooth on

your mobile phone). Then, insert the Bluetooth module to the smart robot. After a

few second, you can find the corresponding Bluetooth device in the list. For iOS,

select "Waveshare_BLE" device. For Android, select "Waveshare_ERD"

device. Then, enter the next page, and choose "remote control mode".

Note: If your Android mobile phone support Bluetooth BLE as well, it will scan two

devices. At this moment, please select the device with the MAC address of

00:0E:0E.

Select "Remote control mode", and click the option Settings to configure the

commands according to the program. When there is key pressed down, it will

trigger corresponding command. And when the key is released and bound up, it

will trigger the "Stop" command.

```
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;
```

When the color of the speed button changes from gray to black, it means the

program has been linked. Now, pressing down the key, you can control the smart

robot. At a meanwhile, the RGB LED display different colors.

For more information about **Dual-mode Bluetooth,** please refer to the corresponding

user manual

Chinese user manual: http://www.waveshare.net/wiki/Dual-mode_Bluetooth

English user manual: http://www.waveshare.com/wiki/Dual-mode_Bluetooth

Bluetooth control smart robot (JSON formatted command)

Remove the Bluetooth module, then, compile and download the program

Bluetooth-json. Next, startup the APP, and insert the Bluetooth module to the robot.

Then, select the option "Remote control mode" to control the robot, and choose

the option "Peripheral control" to control the buzzer to sound and RGD LED to

display different colors. (RGD LED may be unavailable in other robot. So you may

not control it on other robot.)

Raspberry Pi examples

Extract the package AlphaBot2.tar.gz and place it to the directory of /home/pi under raspbian system.

Before using this extension board, you should install the library python to your Raspberry Pi.

In case that the example is unable to be performed after installing the library, you can try to apply the command chmod +x filename to grant the program the executable permission.

Explanation: For the freshmen of Raspberry Pi, please refer to the Raspberry Pi Tutuiral. You can optionally read the following content.

① Raspberry Pi Tutuiral Serial: Access your Pi

② Libraries Installatin for RPi (Only python need to be operated)

Motor testing

Enter the Linux terminal, and input the following commands:

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

Expectation result: The smart robot moves forward when the program runs and drives the motor to rotate.

In case that the direction of motion is wrong, you may need to modify the motor wring or change the pin settings in your program. We recommend the latter one. If left wheels of the robot run in a wrong direction, you should exchange the settings of ain1 and ain2. For right wheels, exchange the settings of bin1 and bin2.
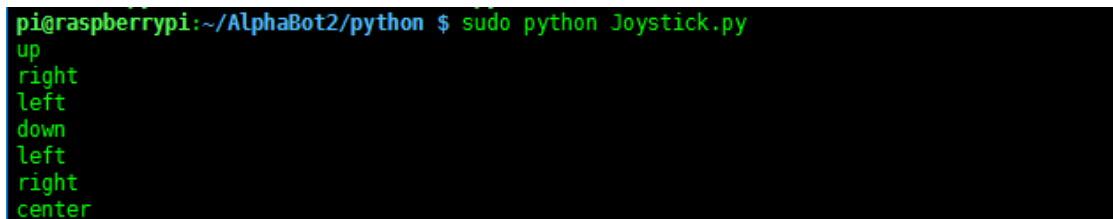
```
def __init__(self,ain1=12,ain2=13,ena=6,bin1=20,bin2=21,enb=26):
```

5-way joystick

Enter the Linux terminal, and input the following commands:

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

Expectation result: This Joystick has 5 control directions: Up, Down, Left, Right and Center (Enter). Open the serial monitor, and set the baud rate to 115200. Then, the monitor displays the current key press. When there is a key pressed down, the buzzer sounds and the motor rotates in the direction according to the key. If the wheel(s) runs in the wrong direction, you should modify the settings of the corresponding pins.

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

Infrared remote control

Enter the Linux terminal, and input the following commands:

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

Expectation result: You can control the smart robot by pressing down the numeric keys on the infrared remote controller, in which, "2" is for go forwards, "8" for go backwards, "4" for turn left, "6" for turn right, and "5" for stop. Keys "-" and "+" can adjust the speed of the robot, and "EQ" can restore the default setting of the speed.

(Note: The key encoding may change with different Infrared remote controller. You

can modify the program as required.)

Infrared obstacle avoidance

Enter the Linux terminal, and input the following commands:

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

When there is no obstacle ahead of the robot, the green LED on the robot is off.

When the obstacle is detected, the green LED will light up. In case that the LED is

always off or always on, you can try to adjusted the potentiometers on the bottom

of the robot to make the LED work properly.
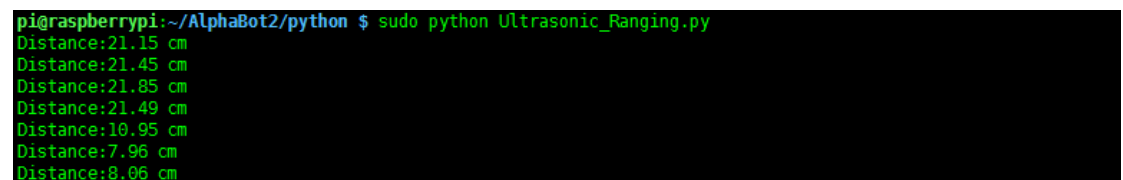
Expectation result: The robot goes straight when there is no obstacle detected by

the sensor on the left, and turns right when obstacle is detected.

Ultrasonic distance measurement (Raspberry Pi 3B kit does not contain ultrasonic unit)

Enter the Linux terminal, and input the following commands:

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

Expectation result: The terminal will display the current measured distance.

```
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
```

Ultrasonic distance measurement (Raspberry Pi 3B kit does not

contain ultrasonic unit)

Enter the Linux terminal, and input the following commands:

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

Expectation result: The robot goes straight when there is no obstacle detected by

the sensor on the left, and turns right when obstacle is detected.

RGB LED

Copy the file rpi_ws281x-master.zip into your Raspberry Pi, and run the following

command to extract the file.
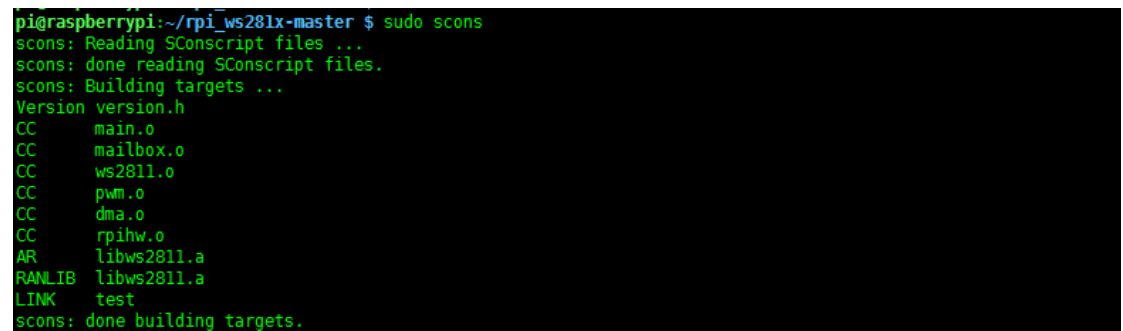
```
unzip rpi_ws281x-master.zip
```

Install the depended-upon software.

```
sudo apt-get build-essential python-dev install scons swig
```

Compile the program.

```
cd rpi_ws281x-master
sudo scons
```


Run the following command to perform the testing, you can see the RGB LED

flickering.

```
sudo ./test
```

Enter the commands below to install the library python.

```
cd python
sudo python setup.py install
```

Run the library python for testing, you can see the RGB LED flickering.

```
cd examples
sudo python lowlevel.py
```

Now, the library has been installed.

Enter the following commands under the terminal:

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

Expectation result: The four RGB LEDs on the bottom of the smart robot will be in

red, green, blue and yellow, respectively.

If they display the wrong color, you can find out the file /boot/config.txt and add

the lines below into it. Restart the module to make the lines take effect. The RGB

LED use DMA control, which can occupy the audio output DMA channels on the

Raspberry Pi. So, adding these lines may cause the earphone interface to be

unavailable.

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

Tracker sensor testing

Enter the Linux terminal, and input the following commands:

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

Expectation result: You can see five set of data printed on the terminal, they are corresponding to five sensors. When the smart robot is placed on the white KT board, the data value will be in the range of 800~900. When it is held in the air, the value can be in the range of 0~100. And, when it on the black line, the value can be in the range of 100~300.

Infrared line tracking

Enter the Linux terminal, and input the following commands:

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

Expectation result: While the program is running, the smart robot goes into the calibration mode, moving to left and then right. When the calibration finished, you can swing the smart robot side to side, and the terminal will display the current position of the black line and the values of five sensors. Place the smart robot on the black line, and press down the center key on the joystick. Now, you can see the smart robot run along with the black line, and the RGB LEDs on the bottom of the robot display different colors.

Servo

Enter the Linux terminal, and input the following commands:

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

Expectation result:: The servo will move to left and right, and its rotational range can reach 180 degree.

mjpg-streamer software real-time monitoring

"MJPG-streamer" takes JPGE from the web camera, and streams them over an IP-

 based network to various types of viewers.

a) Run the command raspi-config to start up the camera before using it:

```
sudo raspi-config
```

Select `Enable Camera`

b) For the camera with CSI interface, the system may be unable to find out the

 device under the directory of /dev/video0. In this case, you should add a line,

 bcm2835-v4l2, into the file /etc/modules.

```
sudo nano /etc/modules
Add the line: bcm2835-v4l2
```


c) After reboot, the system will load the module name stored in this file, and find

 out the device video0 under the directory of /dev. (Please take a notice that the

 letter l in the command bcm2835-v4l2 is the lower case of L.)

Explanation: After performing the steps a), b) and c) , if you still cannot find out the

device video0 under the directory of /dev, and all the operations and commands

are correct, you should run the command: rpi-update.

The software mjpg-streamer in the program has been compiled. You only need to

run the following lines to start up the mjpg-streamer:

```
cd AlphaBot2/lib/mjpg-streamer
```

```
sudo ./start.sh
```

Then, open the browser, and enter the IP address of RPi and the port number: 8080

into the address bar. (Notices: the port number here is 8080, other than 8000 what

we used before.) The website will show the information about the M-JPEG streamer.

Click the option Stream, it will real-time display the image captured by the camera.
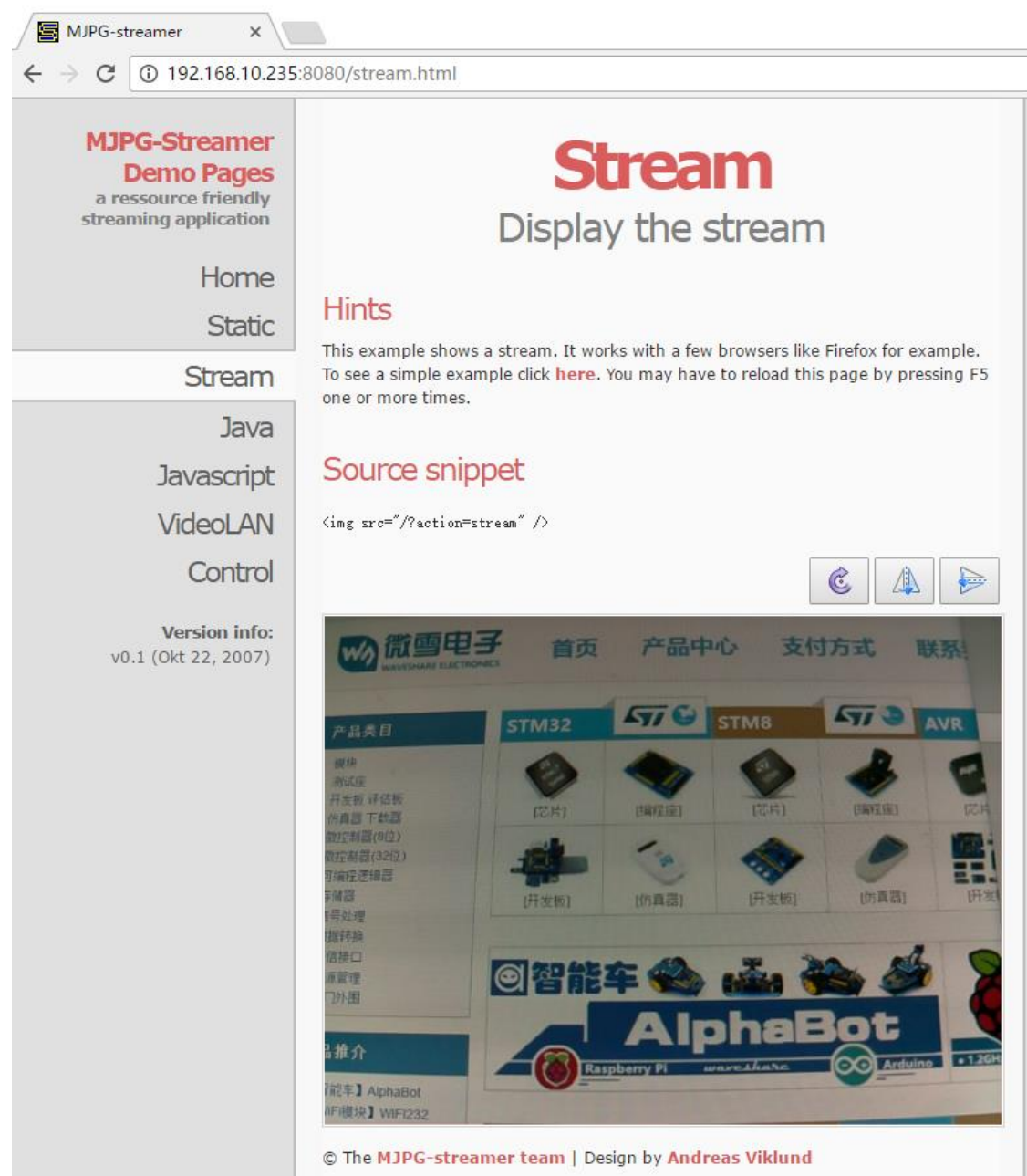
Under the terminal, pressing the keys Ctrl + C can end the program. However, we

recommend you to keep the program running, since we still need it in the following

operations.

Remote control via webiopi
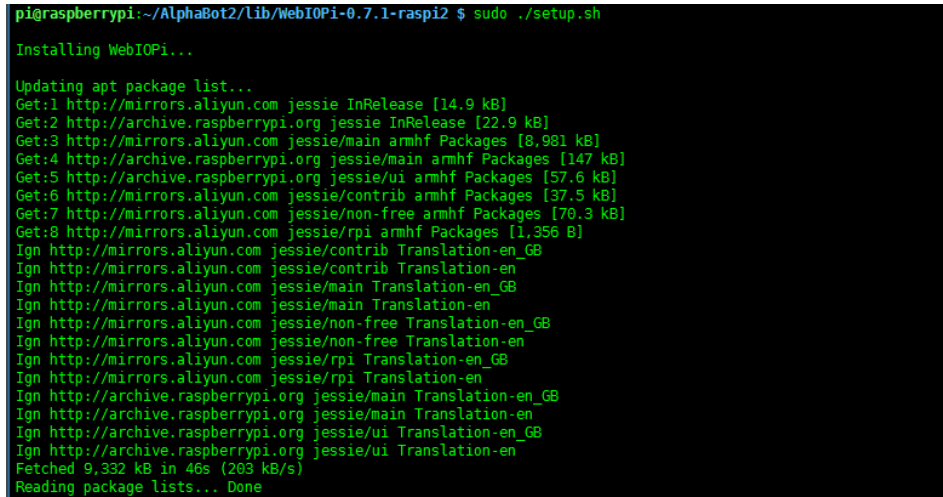
Webiopi is a web framework that can control GPIO of RPi. Here is the website:

http://webiopi.trouch.com/

However, this software is no longer to be updated after the version 0.7.1. The

software here we provided is the improvement version, able to support Raspberry

Pi 3B.

To install the webiopi, you should open a new terminal, and keep the route

described above running.

```
cd AlphaBot2/lib/WebIOPi-0.7.1-raspi2
sudo ./setuo.sh
```


Check if the library is installed successfully by running the line `webiopi -h`. if the

library has been installed, you can see the interface as the figure shows below:

```
pi@raspberrypi:~/AlphaBot2/lib/WebIOPi-0.7.1-raspi2 $ sudo webiopi -h
WebIOPi command-line usage
webiopi [-h] [-c config] [-l log] [-s script] [-d] [port]

Options:
  -h, --help         Display this help
  -c, --config file  Load config from file
  -l, --log file     Log to file
  -s, --script file  Load script from file
  -d, --debug        Enable DEBUG

Arguments:
  port               Port to bind the HTTP Server
```

Enter the terminal, and run the lines:

```
sudo webiopi -d -c /etc/webiopi/config
```

```
pi@raspberrypi:~/AlphaBot2/lib/WebIOPi-0.7.1-raspi2 $ sudo webiopi -c /etc/webiopi/config
2017-03-15 03:28:53 - WebIOPi - INFO - Starting WebIOPi/0.7.1/Python3.4
2017-03-15 03:28:53 - WebIOPi - INFO - GPIO - Native mapped to REST API /GPIO
2017-03-15 03:28:53 - WebIOPi - INFO - Loading configuration from /etc/webiopi/config
2017-03-15 03:28:53 - WebIOPi - INFO - Access protected using /etc/webiopi/passwd
2017-03-15 03:28:53 - WebIOPi - INFO - HTTP Server binded on http://192.168.10.235:8000/
2017-03-15 03:28:53 - WebIOPi - INFO - CoAP Server binded on coap://192.168.10.235:5683/
2017-03-15 03:28:53 - WebIOPi - INFO - CoAP Server binded on coap://224.0.1.123:5683/ (MULTICAST)
```

Then, open the browser on your PC or cell phone, and enter the IP address of RPi and the port number: 8000 into the address bar. For example, input http://192.168.1.16:8000 (the address can be changeable according to your actual condition). To login the website, you should enter your account and password. The default account name is webiopi, and the original password is raspberry. When finished the steps above, if you can enter the webpage WebIOPi Main Menu, it means the environment is configured well.
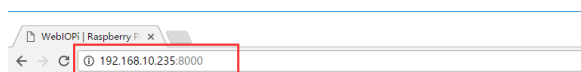
WebIOPi Main Menu

GPIO Header

Control and Debug the Raspberry Pi GPIO with a display which looks like the physical header.

GPIO List

Control and Debug the Raspberry Pi GPIO ordered in a single column.

Serial Monitor

Use the browser to play with Serial interfaces configured in WebIOPi.

Devices Monitor

Control and Debug devices and circuits wired to your Pi and configured in WebIOPi.

Click the option "GPIO Header", then, you can see the interface below. In this webpage, you can control the level of the GPIO.
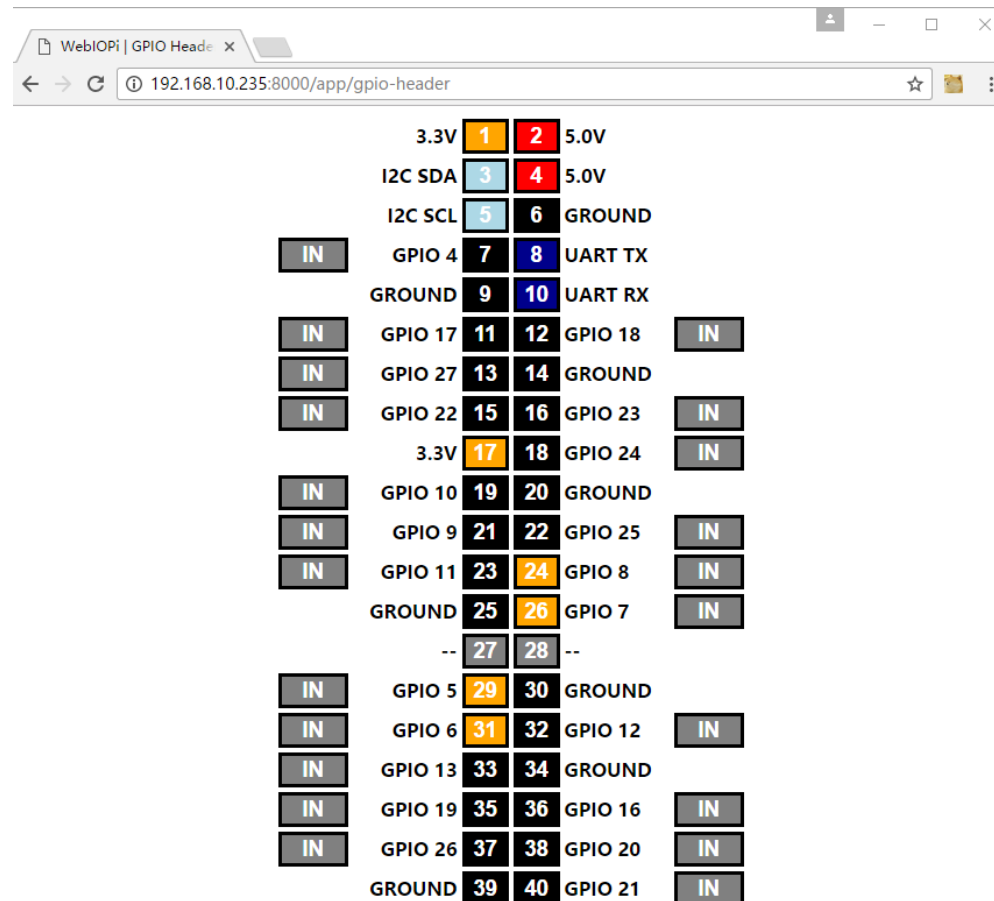
Enter the directory of mjpg-AlphaBot, and modify the video source IP.

```
cd ~/AlphaBot2/mjpg-AlphaBot/
sudo nano index.html
```

Find out the link: http://192.168.10.235:8080/?action=stream

In here, 192.168.10.235 is the IP address of RPi, and it can be changeable according to your actual condition. The port number should be set to 8080 (This port is used for transmitting stream).

Start up the Raspberry Pi camera program:

```
cd ~/AlphaBot2/mjpg-AlphaBot
sudo webiopi -c config
```

Then, open the browser, and enter the IP address of RPi and the port number: 8000

into the address bar. (Notices: the port number here is 8000, other than 8080.) The

image captured by the camera will be shown in the website. Clicking the direction

icon on the website can control the action of the robot.
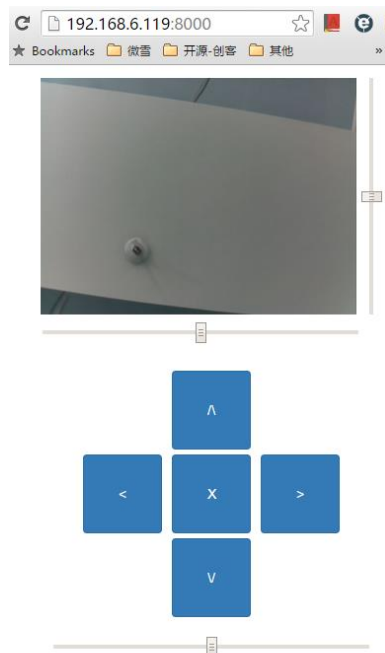
! If you cannot enter this webpage, we recommend you to use the Chrome browser or the Firefox browser. Furthermore, please make sure the current directory of Raspberry Pi is:

`~/AlphaBot2/mjpg-AlphaBot/mjpg-AlphaBot`

since there is a file index.html under this directory. And then, input the code:

`sudo webiopi -c config`

to enter the index.html. Or else, you will go to the WebIOPi Main Menu webpage.

Configure boot-up option:

Replace the file config

```
cd ~/AlphaBot2/mjpg-AlphaBot
sudo cp config /etc/webiopi/config
```

Run the command below to set the webiopi to be started up on boot.

```
sudo update-rc.d webiopi defaults
```

Open the file /etc/rc.local with your editor to add the following commands:

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

Add the commands following the comments. Please make sure the line exit 0 is the

last line. And then, save the modification and exit.

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

Remote control via Bottle

Bottle is a simple and effective micro python Web framework complying with WSGI. By Bottle, you can fast and easily implement Web control.

1) Install the library:

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

2) Perform web control RGB LED via Bottle
```
cd ~/AlphaBot2/Web-RGB
sudo python main
```

3) Then, enter the IP address of your Pi and the port number 8000 into the address bar of the browser. Clicking different position of the pallet, you can see RGB LED displays different colors. There are 3 display modes selectable, static, breath and flash.

(Note: This web-RGB program only support the mobile phone browser. It is unavailable for

PC browser.)
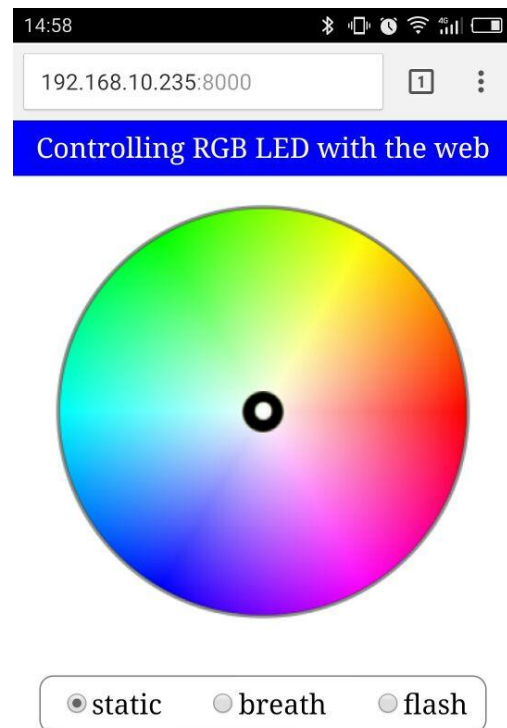
To implement web video control the smart robot by Bottle, you should modify the

IP address for the video stream.

```
cd ~/AlphaBot2/Web-Control
vi index.html
```

Find out the line below, and change the IP address to the actual RPI address and set

the port to the stream port.

```
<img width="320" height="240" src="http://192.168.10.235:8080/?action=stream">
```

Then, run the following command to start the program:

```
sudo python main
```

Then, open the browser, and enter the IP address of RPi and the port number: 8000

into the address bar. (Notices: the port number here is 8000, other than 8080.) Then,

you can see the following webpage. Clicking the direction icon on the website can

control the action of the robot.
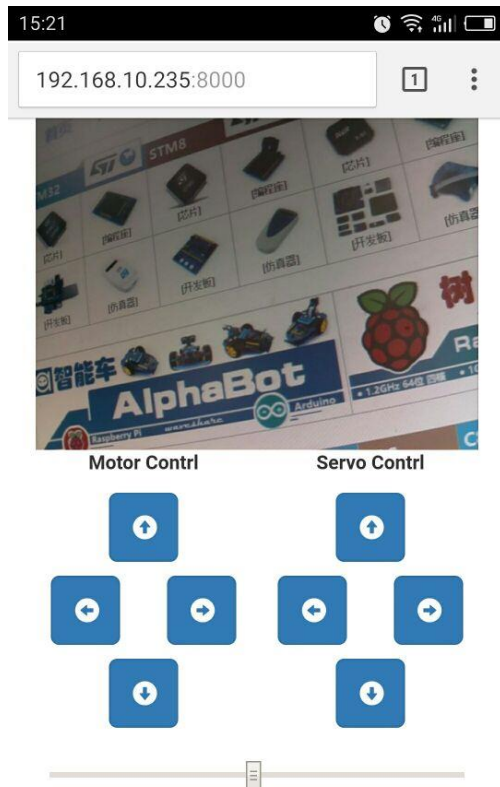
You can set the program to be started up on boot. Open the file /etc/rc.local with

your editor and add the following line into it.

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

Add the commands following the comments. Please make sure the line exit 0 is the

last line. And then, save the modification and exit.

```
cd /home/pi/AlphaBot2/lib/mjpg-streamer
sudo ./start.sh &
cd /home/pi/AlphaBot2/Web-Contrl
sudo python main &
```

Since the Raspberry Pi uses dynamic IP which may change each time the device restart, you can add the following lines under the directory of /etc/dhcpcd.comf to set the static IP:

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

In here, "192.168.1.1" is the router IP address, and "192.168.1.114" is the IP address to be set, following which the "/24" should not be removed.

Remote control via software

To implement video control the smart robot via the software on your mobile phone or PC, you should modify the IP address in the program firstly.

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

Find out the line below, and change the IP address to the actual RPI address and set the port to the stream port.

```
Host = '192.168.10.235'
```

Then, run the following command to start the program:

```
sudo python main
```

Open a new terminal to run the mjpg-streamer.

Run the program AlphaBot.exe on your PC, and enter the IP address, the port numbers of mjpg-stream and smart robot Soket sever.

Click the button Viedeo Connet to connect to the video. When the connection is built up, the button will turn to green. Then, click the button Cmd Connect to connect to the smart robot sever. You will see the button turn to green, if the connection is successful. Now, you can control the action of the smart robot by pressing down the keys on the left, and control the camera and steer by the keys on the right.
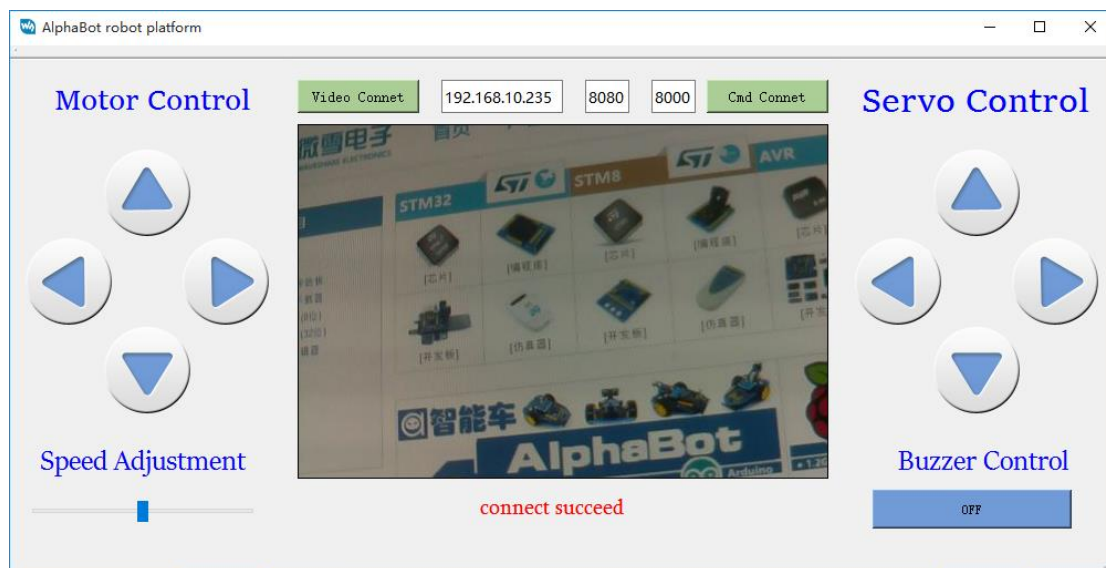
Then, scan the QR code below with your mobile phone, and download the relative APP.

Enter the actual IP address of RPi, and click the button Connect. Then, you can see the images captured by the camera. Next, click the button Connect under the image. When the control port has been connected to the program, this button will change from green to blue. Now, you can control the actions of the smart robot and the steer by pressing down the corresponding keys. By dragging the slider, you can change the speed of the robot.

Open the file /etc/rc.local with your editor to add the following commands:

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

Add the commands following the comments. Please make sure the line exit 0 is the last line. And then, save the modification and exit.

```
cd /home/pi/AlphaBot/lib/mjpg-streamer
sudo ./start.sh &
cd /home/pi/AlphaBot/App-Contrl
sudo python main &
```

You should connect the Raspberry Pi to the router at every restart. Of course, you can enable the AP function of the RPi wireless network adapter, taking the RPi as sever. For more information about the operation, please refer to the Raspberry Pi tutorial.

Remote control via RPi Bluetooth

1) Upgrade the relative packages of the Bluetooth, and install the depended-upon software.

```
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) Add the pi user to the Bluetooth group.

```
sudo usermod –G Bluetooth –a pi
```

3) Restart the Raspberry Pi.

```
sudo reboot
```

4) Start/Add the SPP, and turn on the Bluetooth device.

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

Modify two lines in the following file:

```
 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
```

After restarted the RPi, enter the command hciconfig (similar to the command ifconfig) to check the Bluetooth service.

```
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
```

If the Bluetooth has been working, you will see the device hci0 displayed on the screen. When the Bluetooth does not work, no Bluetooth device can be identified.

Note: If the line dtoverlay=pi3-miniuart-bt exists in the file /boot/config.txt, it should be commented out.

5) Connect to Bluetooth device

Input the following command to enter the Bluetooth control interface (you can input the command help to check the command list after entered the interface).

```
sudo bluetoothctl
```

```
agent on
```

After entered the interface, input the following lines (you can input the command

help to check the command list after entered the interface).

```
agent on
default-agent
```

Following the scan on, you can see the physical address of the scanned Bluetooth

device, similar to XX:XX:XX:XX:XX:XX.

For Bluetooth pairing, use the following command (In here, XX:XX:XX:XX:XX:XX can

be replaced to your device address. For Android mobile phone, you can check the

Bluetooth address by Setting-> About phone-> Status information)

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

When pairing is successful, add the device to the trust list.

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


Next, connect to the device.

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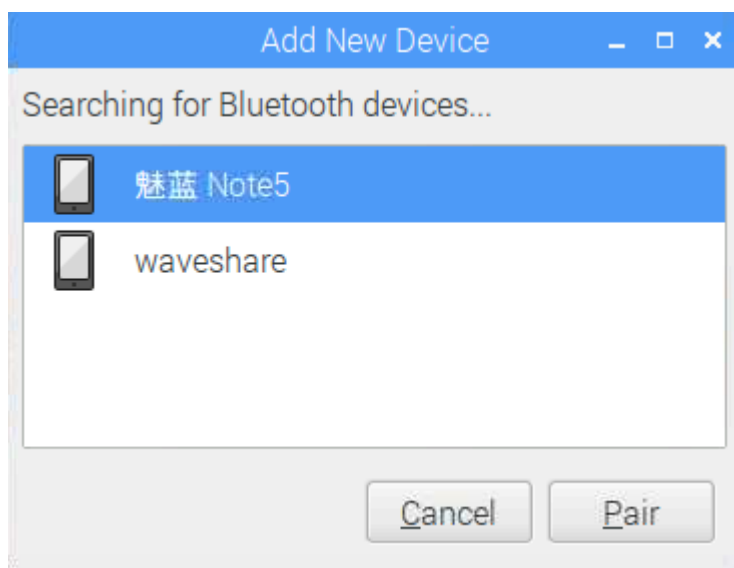

At last, enter the command quit to exit the interface. After the settings above, as

the Bluetooth device is turned on, the Raspberry Pi will automatically connect to it.

Note: If you are unable to build up the connection with the command line, you can
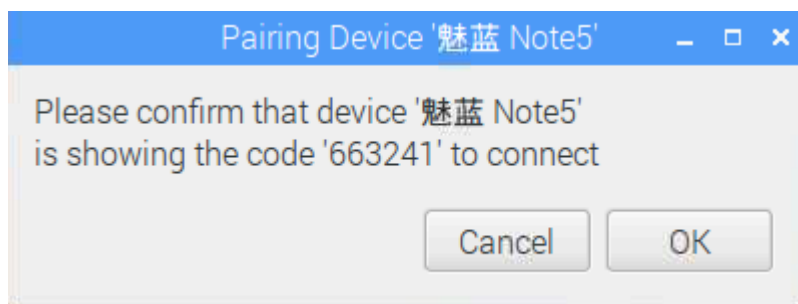
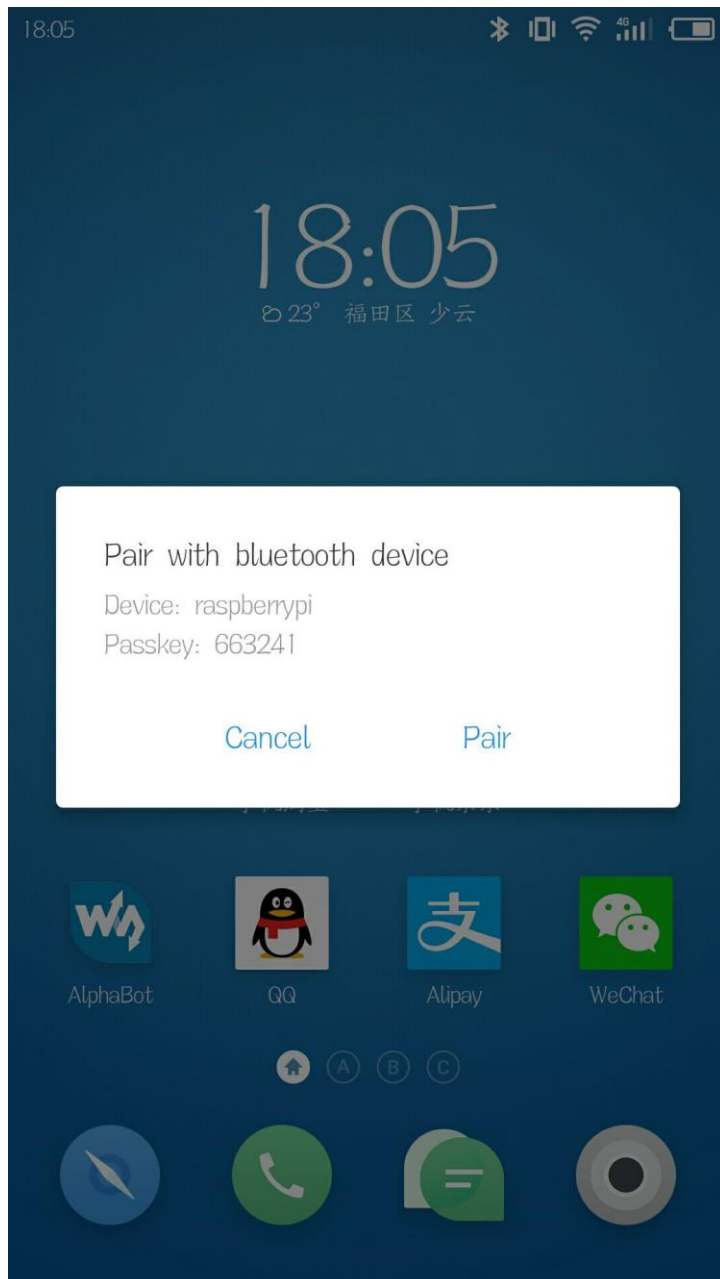enter the graphic interface to make the connection.
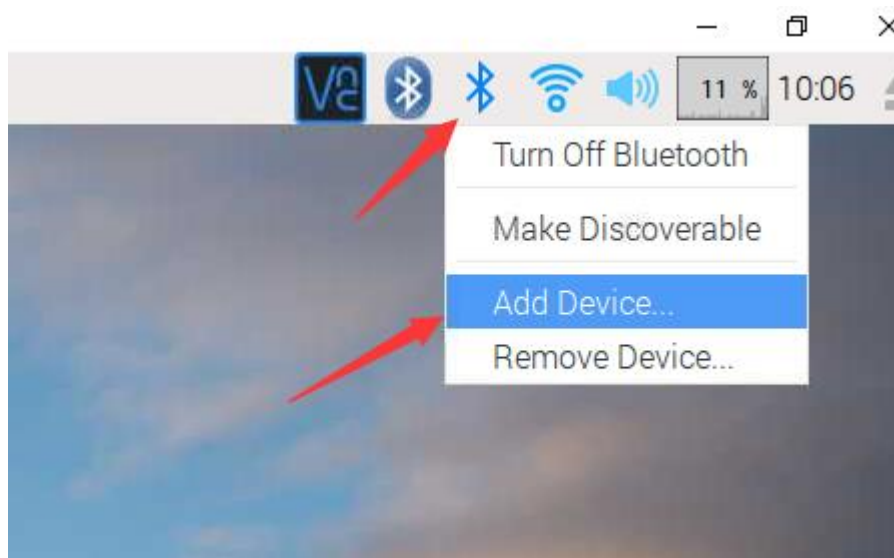

Click the Bluetooth icon and add new device.

Select the corresponding Bluetooth device. Please make sure the Bluetooth on the

mobile phone should be enable and set to discoverable.

On the RPi, select the option OK for pairing. And at the same time, on the mobile

phone, click the option Pair.

Then, you may see the prompt saying that "Paired successfully, but this device has

no services which can be used with Raspberry Pi."
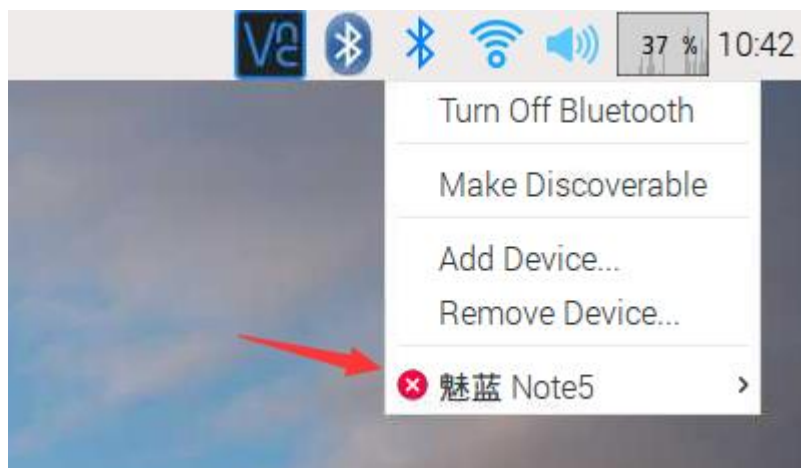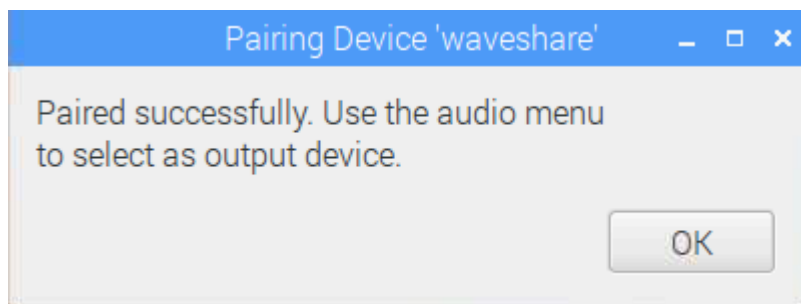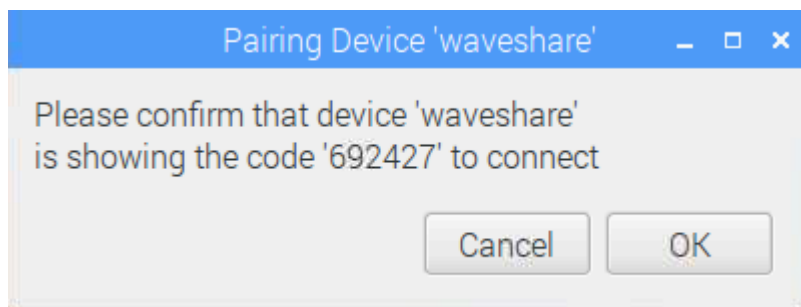
Now, the connection has been built up.

Run the following commands to start the program:

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

Then, scan the QR code below with your mobile phone, and download the relative

APP. (This is the software for controlling the Dual-mode Bluetooth module. If you

has installed this software in your Arduino mobile phone, you can skip this step)

Start the APP, and click the option Scan. (Note: please make sure the Bluetooth on

your mobile phone is turned on) Normally, you can see the relative Bluetooth

devices in the list. Then, select the option "raspberrypi". Go to the next page, and

select the option Remote control.
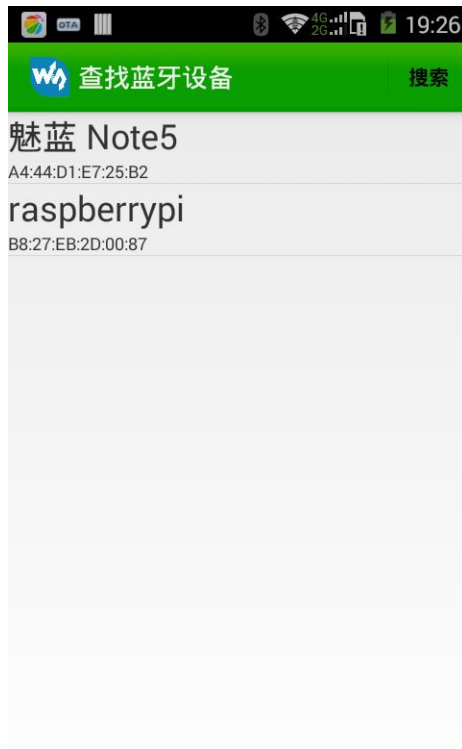
If the Bluetooth of the Raspberry PI can not be searched, you can run the following

lines to enable the Bluetooth and set it to discoverable.

FAQ

Question 1: Comparing with AlphaBot, is there any improvement on AlphaBot2?

Answer:

- AlphaBot2 has a smarter dimensions and more stable structure. Sensors are highly integrated in it.

- High-precision metal motor with low noise.

- TB6612FNG motor driver chip with high efficiency, smarter dimensions, and no heat produced, comparing with L298P.

- Increase more peripherals for debugging, such as keys, buzzer, RBG LED, USB converter and the likes.

- AlphaBot2-Ar adapter board adds OLED, and AlphaBot2-PiZero adapter board adds usb Hub.

- PCA9685 steer control chip, able to avoid jittering when using Raspberry Pi software to PWM control the steer.

- Provide much more programs. For Raspberry Pi, it has Andiron App video control and on-board Bluetooth remote control. For PC, it provide Qt software control.

Question 2: What kind of battery can be applied on Alphabot2 smart robot?

Answer: It uses 14500 type battery which is in the diameter of 14mm and length of 50mm. For a single battery, its rate voltage is 3.7V, full voltage is 4.2V, and its

lowest voltage should not be lower than 2.75V, or else the battery may be permanently damaged. This battery has a same dimensions with AA battery. But you should not use the AA battery to replace it, since the AA battery cannot provide enough voltage.

Question 3: Why the smart robot cannot perform infrared obstacle avoidance?

Answer: Firstly, you should check the whether the infrared detection module on the smart robot works properly. When there is no obstacle ahead of the robot, the green LEDs on the robot are off. When the obstacle is detected, the green LEDs will light up. In case that the LEDs are always off or always on, you can try to adjusted the potentiometers on the bottom of the robot to make the LEDs work properly.

Question 4: What can I do, if the motion of the smart robot is wrong?

Answer: In case that the direction of motion is wrong, you may need to change the pin settings in your program.

Question 5: The rotational range of the steer is abnormal.

Answer: The rotational range of the steer can reach 180 degree. In normal condition, the steer can turn to left at 90 degree, and turn to right at 90 degree as well. If there is a big difference between the both sides, you should reassemble the steer and make adjustment.

Question 6: How to know if the battery has power?

Answer: If there is low battery, the smart robot will move slowly. Normally, when the Raspberry Pi shuts down or become unable to connect, you should charge the battery. When the battery voltage is low, you should stop using the battery, or else it may be permanently damaged and cannot be applied any more.

Question 7: Can other MCUs control the base board of the smart robot?

Answer: Of course, they can. We have provided the schematic diagram and the dimension diagram of the smart robot for users to perform secondary development.

Safety Precautions

Notice

Please reserve the following information for future use.

Caution

To prevent AlphaBot2, a precision device, from breaking down or being damaged, please follow the requirement listed below.

● The product employs two series 14500 batteries for power supply with input voltage of 7.4V. When working with Arduino and Raspberry Pi at a same time, the external power supply should be applied. Any external power supply has to obey to the regulations and standards of power supply for your area.

● The product should be applied under a well-ventilated environment, to guarantee the heat dissipation of power supply and product performance.

● The disassembly chassis should be placed on the stable, flat and non-conducting surface, without touching any conductive materials.

● Installing the incompatible device to the chassis may cause damage and void the warranty.

● The product should not expose to water and moisture, or not be placed on conductive surface, in the course of any operations.

● AlphaBot2 should not touch any heat source, to make sure reliability service under normal room temperature environment

- Protect the printed circuit board and the components on it from mechanical or electrical damages, in the course of any operations.

- Avoid touching the energized printed circuit board in the course of any operations. Touching the edge of the board is allowed, which can reduce the risk electrostatic discharge.