树莓派智能小车PiRobot

**用户手册**

2016-12-1

自欺莫柒人

南京华世达科技有限公司

南京

玩转树莓派

v1.1

**阅读提示**

**本手册部分内容可能会让用户阅读起来稍微困难，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，C语言编程以及一些Linux操作系统的基本命令，以便深入理解示例程序的原理。由于篇幅所限，本手册不会涉及太多的程序原理。如果对这方面有更深入的探索，我希望您可以直接查看程序的源代码，因为智能小车开发的实战经验比理论知识更为重要。**

**由于小车的电器特性所限，相对于其他的开发板，用户应当更注意安全，正确的使用本产品，以避免小车配件对树莓派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。**

**PiRobot系统更新日志：**

**1. 基于最新的桌面环境PIXEL，全新的Raspbian桌面环境提供了更为养眼的界面设计、全新的图标、全新的应用菜单和全新的主题。  
2. 除了之前的C语言服务端代码这次添加了Python服务端源码。并配置了运行需要的环境。  
3. 进行了界面优化，安装了中文环境以及中文输入法。  
4. 预装谷歌浏览器以及Flash插件，可直接在线观看视频。  
5. 预装VCN、XRDP，安装了Qmmp音乐播放器。**

**主要功能** 手机电脑远程遥控，超声避障，黑线循迹，红外避障/物体追踪（本手册不针对某个特定的套餐）测试时请不要安装轮子。下面带大家熟悉一下树莓派的GPIO控制

## 常用开源工程简介

树莓派内核中已经编译自带了 gpio 的驱动，我们常通过一些第三方写好的库函数来完成具体的操作，比较常见的操作库函数有：

**python GPIO**

* **开发语言——python**
* **简单介绍——树莓派官方资料中推荐且容易上手。python GPIO 是一个小型的 python 库，可以帮助用户完成 raspberry 相关 IO 口操作，但是 python GPIO 库还没有支持 SPI、I2C 或者 1-wire 等总线接口。**
* **官方网站——**[**https://code.google.com/p/raspberry-gpio-python/**](https://code.google.com/p/raspberry-gpio-python/)

**wiringPi**

* **开发语言——C 语言**
* **简单介绍——wiringPi 适合那些具有 C 语言基础，在接触树莓派之前已经接触过单片机或者嵌入式开发的人群。wiringPi 的 API 函数和 arduino 非常相似，这也使得它广受欢迎。作者给出了大量的说明和示例代码，这些示例代码也包括 UART 设备，I2C 设备和 SPI 设备等。**
* **官方网站——**[**http://wiringpi.com/**](http://wiringpi.com/)

**BCM2835 C Library**

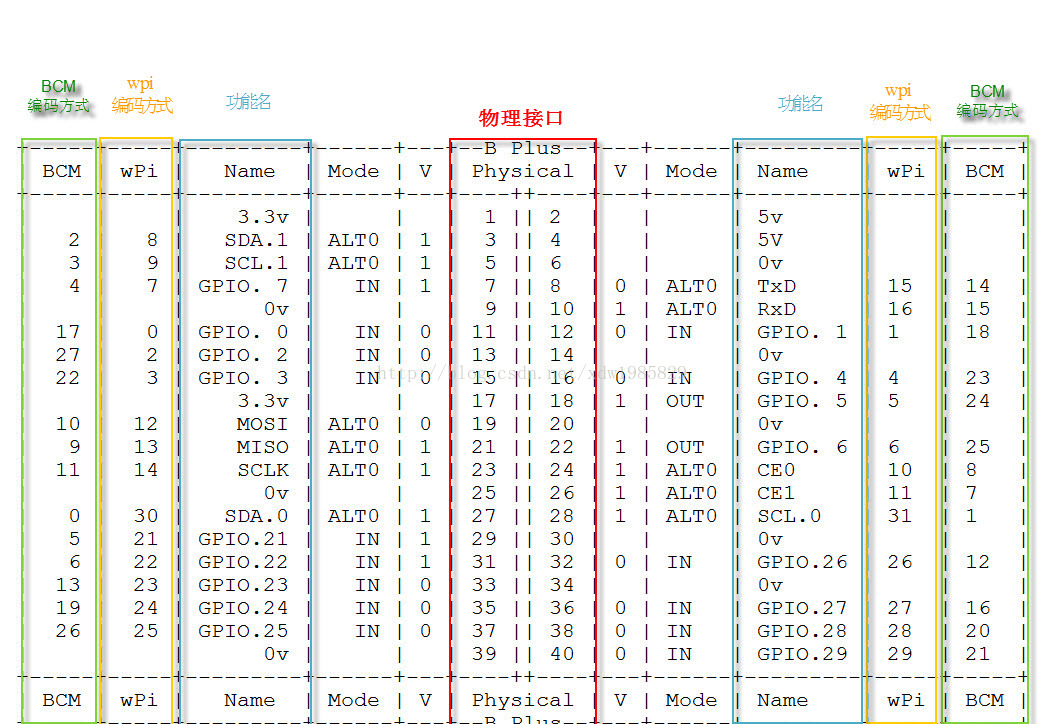
* **开发语言——C 语言**
* **简单介绍——BCM2835 C Library 可以理解为使用C语言实现的相关底层驱动，BCM2835 C Library 的驱动库包括 GPIO、SPI 和 UART 等，可以通过学习 BCM2835 C Library 熟悉 BCM2835 相关的寄存器操作。如果有机会开发树莓派上的 linux 驱动，或自主开发 python 或 PHP 扩展驱动，可以从 BCM2835 C Library 找到不少的“灵感”。**
* **官方网站——**[**http://www.airspayce.com/mikem/bcm2835/**](http://www.airspayce.com/mikem/bcm2835/)

**树莓派GPIO编号方式**

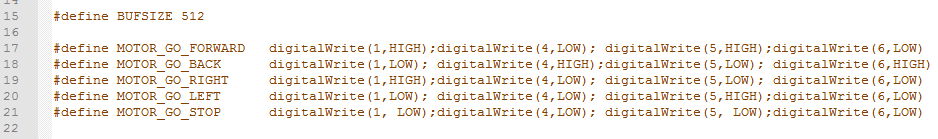
**功能物理引脚： 从左到右，从上到下：左边基数，右边偶数：1-40**

**BCM： 编号侧重 CPU 寄存器，根据 BCM2835 的 GPIO 寄存器编号。**

**wpi： 编号侧重实现逻辑，把扩展 GPIO 端口从 0 开始编号，这种编号方便编程。正如下图 WiringPi 一栏。**



树莓派引脚图

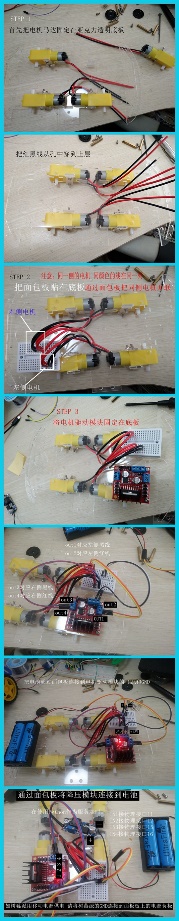




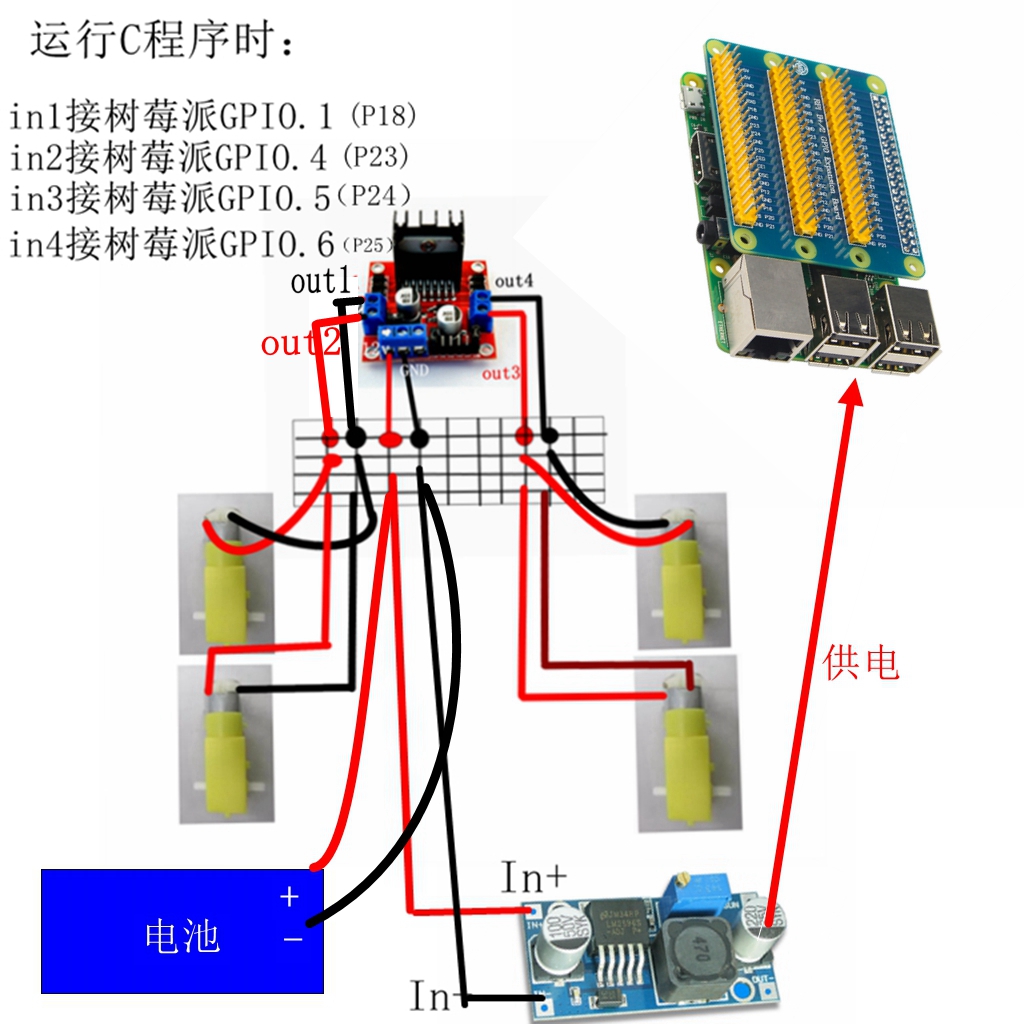
小车运行的PiRobot3.5系统已经安装了wiringPi和Python-Dev等环境,系统镜像和代码以及控制软件所有下载地址：[（12.20](http://pan.baidu.com/s/1dFhzDD3（12.20)日代码更新）地址到树莓派小车群

如果您希望自己动手安装请参照： <http://wiki.jikexueyuan.com/project/raspberry-pi/gpio.html>

**组装小车底板**

 图片存储位置在网盘：地址到树莓派智能小车群获取

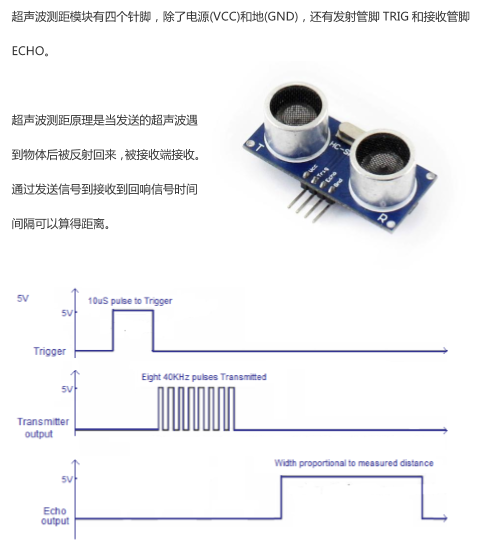
图片接线方式为Python对应的连接方式（IN1、2、3、4接在树莓派物理接口11 12 15 16），C语言服务端电机驱动模块IN1、2、3、4请接wPi编码方式的GPIO1、4、5、6。参照树莓派引脚图接线。

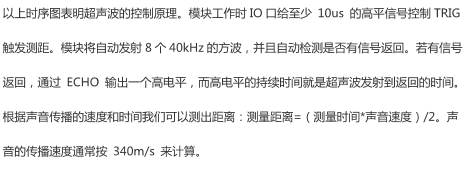


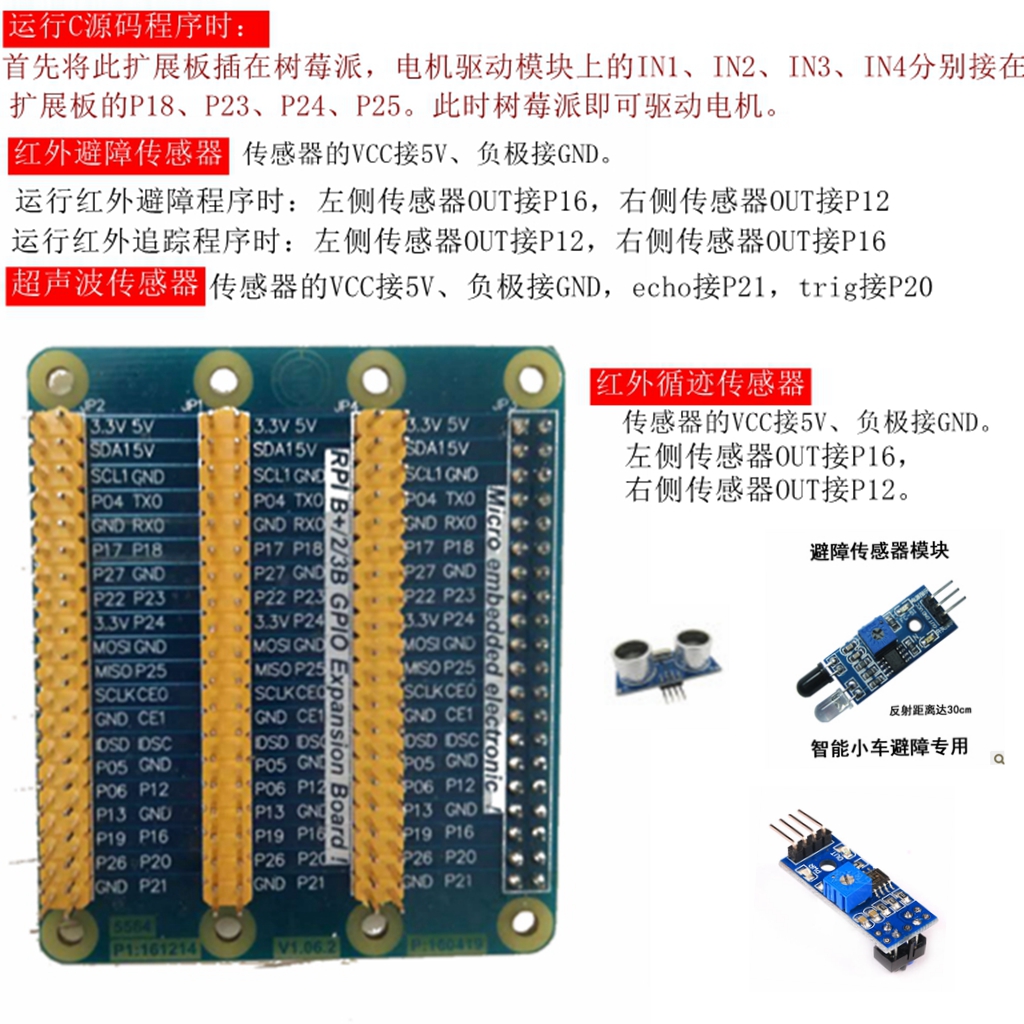
**电源部分**

小车采用两节18650串联供电，输入电压为7.4V（单节电池3.7V）,经过LM2596S稳压后给树莓派，传感器，电机供电。

**超声测距模块——自动避障**



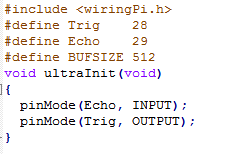




先看一下超声测距模块实现测距的源码（网盘中bizhang-pwm.c的文件），用filezilla上传到树莓派文件，编译命令如下：

gcc csb.c –o csb -lwiringPi –lpthread

运行程序： sudo ./csb

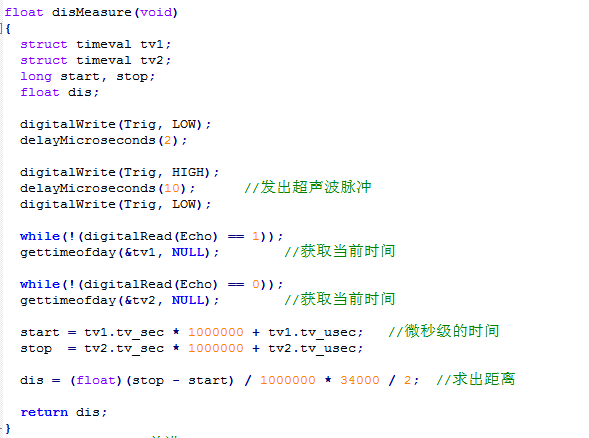
 

VCC接树莓派5V

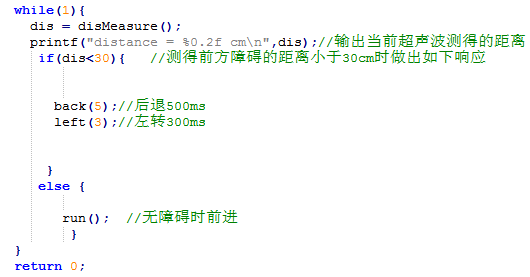
GND接树莓派GND

Trig 接树莓派GPIO28（wpi编码方式）

Echo 接树莓派GPIO29（wpi编码方式）



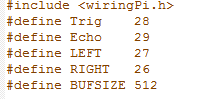
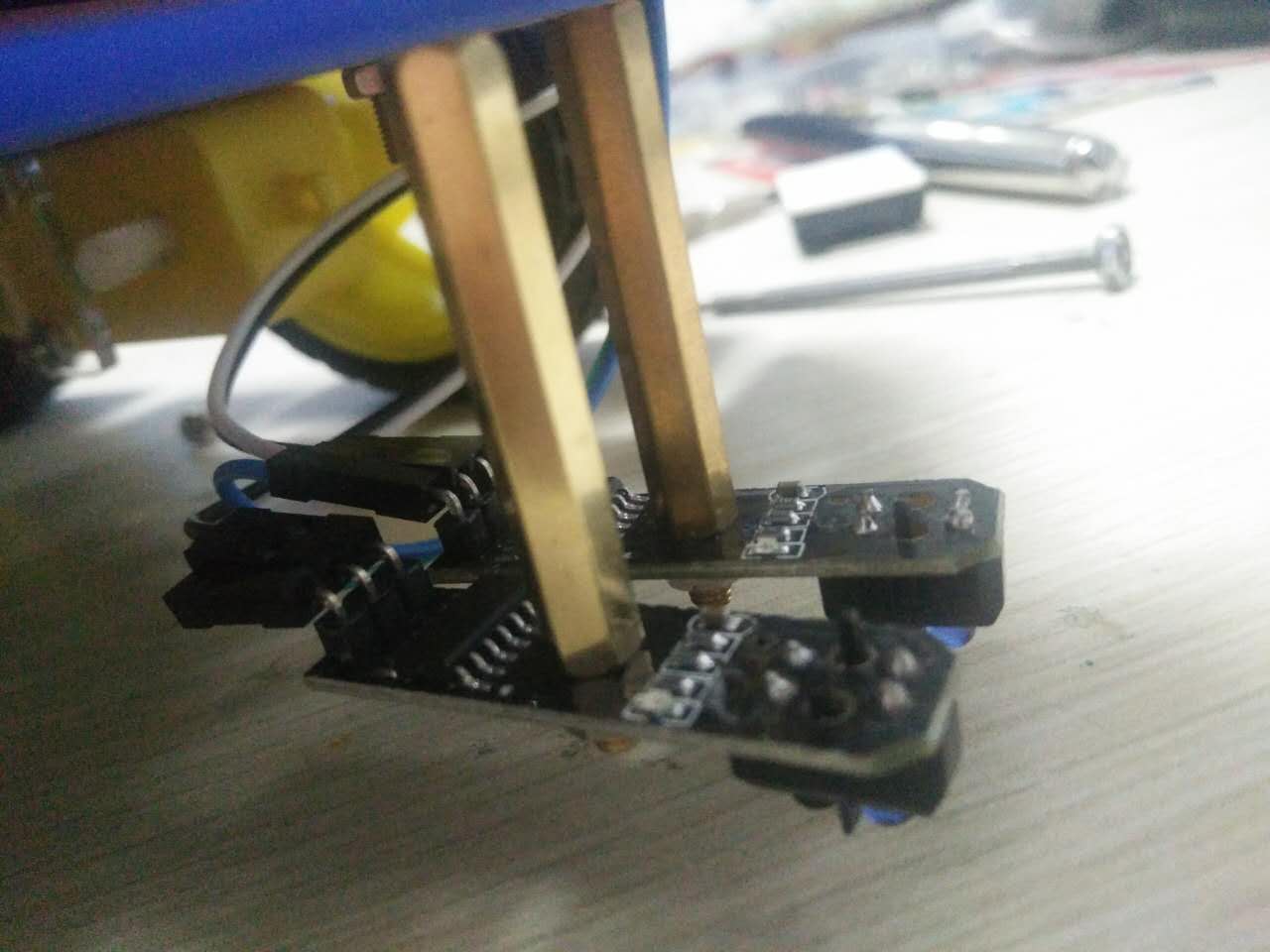
disMeasure函数返回距离值dis，main调用此函数，判断dis是否大于给定的参数。如小于30cm则直线运行，若测得距离小于30cm,后退然后左转。代码如下图



源码中各个动作的函数不再列出。可以根据自己的需要更改代码。函数调用了soft PWM库，可对电机转速进行控制。代码不做过多的分析。

**黑线循迹**

循迹模块和红外避障模块类似。循迹模块的红外发射二极管不断发射红外线，当放射出的红外线被物体反射时，被红外接收器接接收，并输出模拟值。输出模拟值和物体距离以及物体颜色有关。可以通过两路或者多路红外传感器判断黑线的位置，从而控制小车运动。可通过电位器旋钮调节点测距离。



代码分析：



只有两路循迹模块的返回值都为低时，小车前进。车子始终在黑线上行驶。代码不做过多描述。修改代码后编译

gcc xunji.c –o xun -lwiringPi -lpthread

运行 sudo ./xun

**红外避障和红外物体追随**

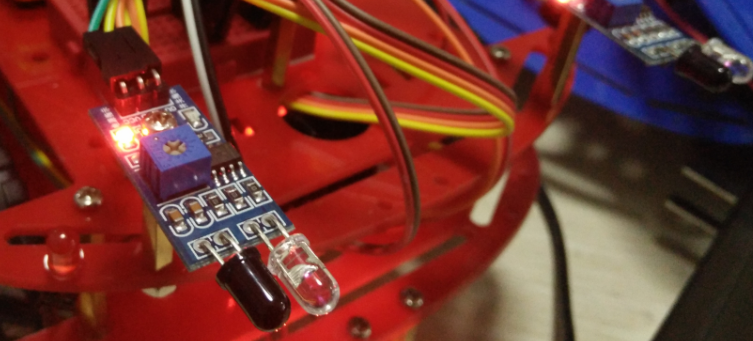
搭配两个红外避障传感器 既可以实现小车避障 也可以实现小车物体追随。我们先来看看红外模块避障的代码（hwbz.c）。可通过电位器旋钮调节点测距离。

**gcc hwbz.c -o hwbz -lwiringPi -lpthread （注意前面有空格 加中间的空格位置）**

**运行命令是 sudo ./hwbz**

定义引脚 VCC接树莓派5V ，GND接树莓派GND，左侧红红外模块的OUT接树莓派GPIO27（wpi编码方式），右侧接GPIO26



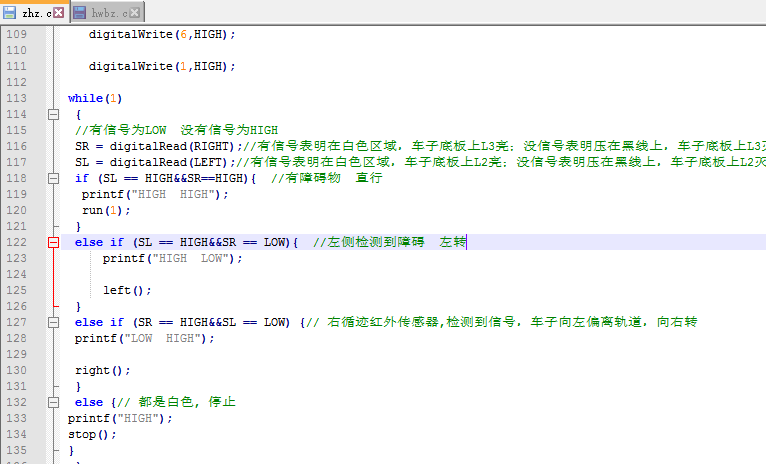


物体追踪的代码为 zhz.c可通过电位器旋钮调节点测距离。

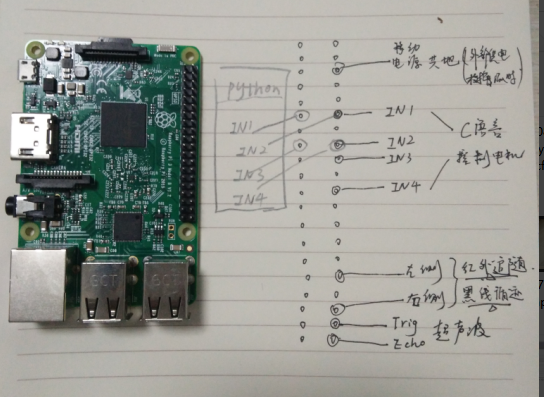
**gcc zhz.c -o zhz -lwiringPi -lpthread （注意前面有空格 加中间的空格位置）**

**运行命令是 sudo ./zhz**

定义引脚 VCC接树莓派5V ，GND接树莓派GND，左侧红外模块的OUT接树莓派GPIO26（wpi编码方式），右侧的接GPIO27 。



注意： 所有的C语言源程序运行时 ，IN1 IN2 IN3 IN 4 均接到树莓派的GPIO.1 GPIO.4 GPIO.5 GPIO.6 (wpi编码方式)



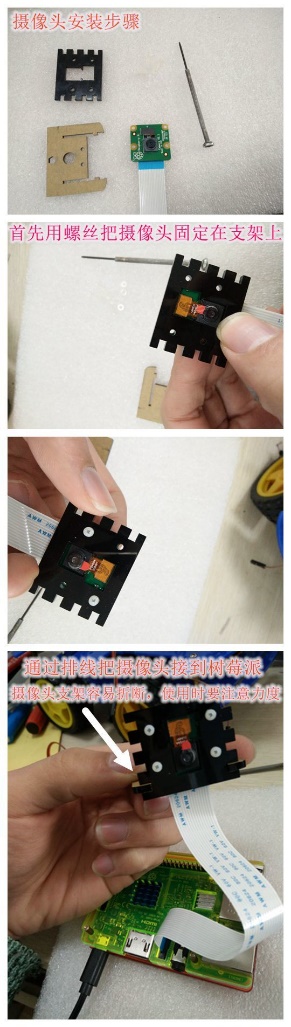




**WiFi视频远程遥控**

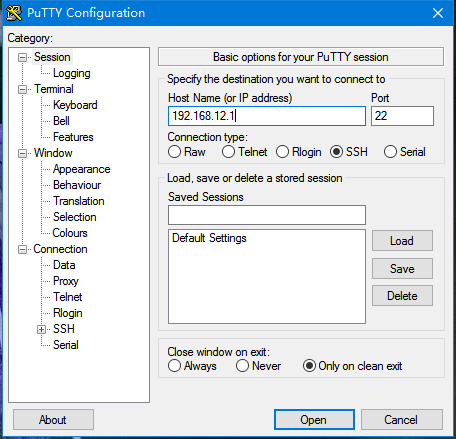
用树莓派作为主控时，可以用手机端或者网页实现WiFi视频监控，通过mjpg-streamer软件实时显示摄像头拍摄图片。MJPG-streamer适用于从webcam摄像头采集图像，把他们以流的形式通过基于Ip的网络传输到拥有浏览器的移动设备，通过mjpg-streamer软件可以实现将摄像头中采集到的图像通过网络传输到浏览器网页上显示。

**树莓派摄像头支架安装步骤**

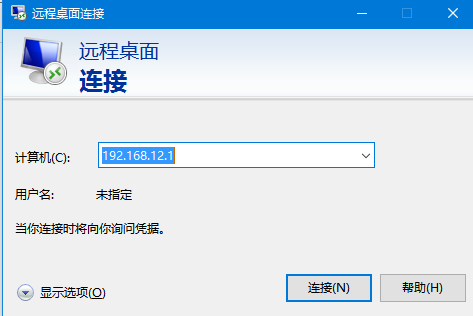
****大图在群文件和网盘

**WiFi视频远程控制程序C语言版**

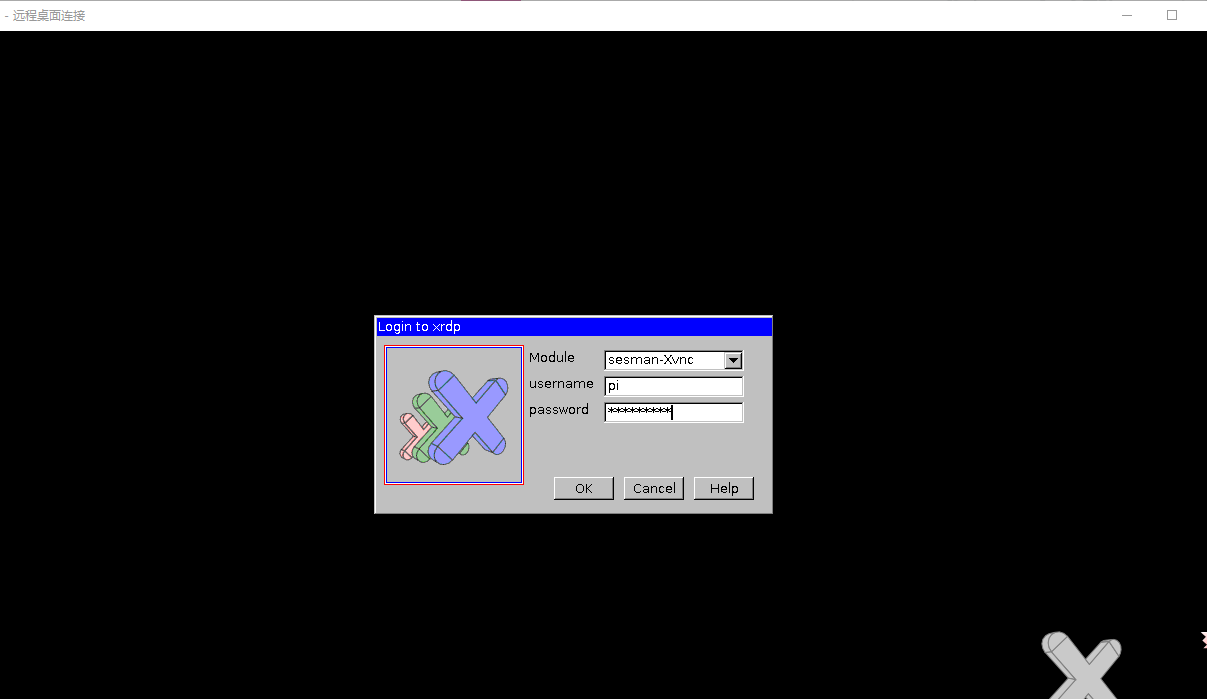
那我们就先把树莓派开机，进入系统树莓派自动开启热点。打开笔记本的WiFi连接，找到一个名为pi3的热点，密码为12345678。连接成功后，着打开putty （网盘链接中下载）输入192.168.12.1 端口号是22



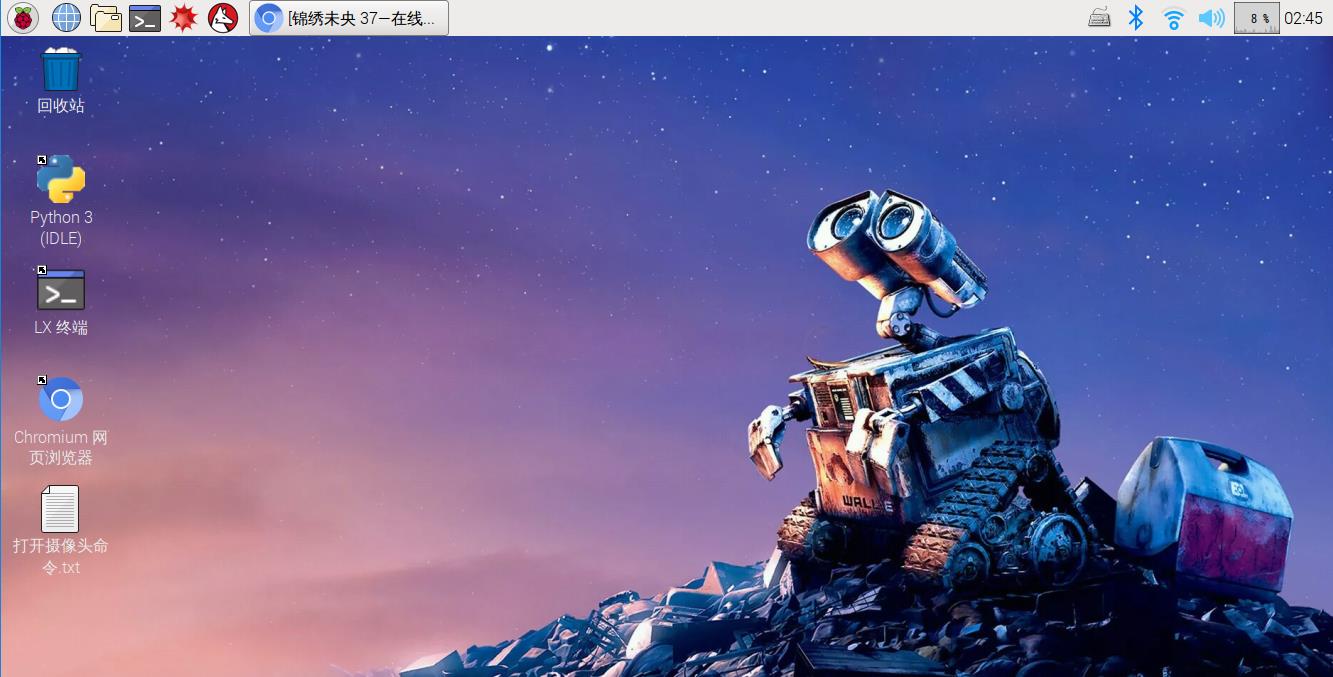
输入用户名pi，然后输入密码raspberry（不显示），回车在命令行输入tightvncserver 后回车。断电重启树莓派之后笔记本再次连接到树莓派的热点即可远程桌面到树莓派。 所有程序>附件>远程桌面连接。



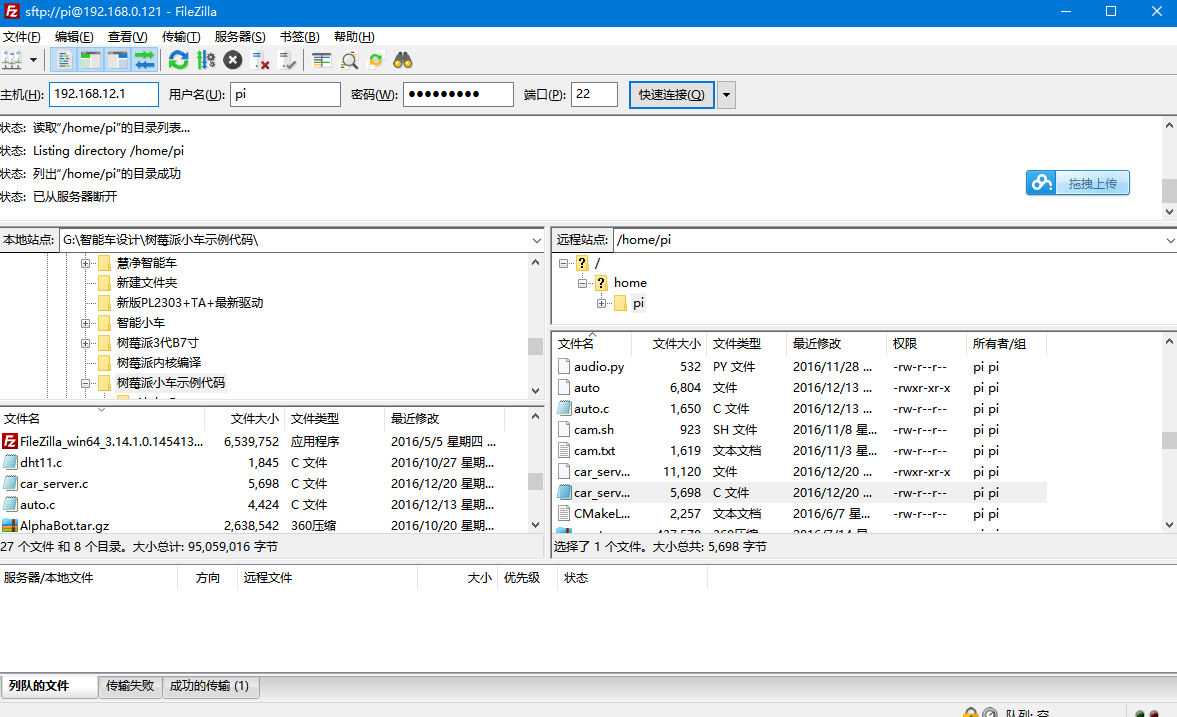
使用用户名 pi 密码 raspberry 登陆树莓派的系统桌面。



点击ok 或者回车就可以看到我们炫酷的桌面了

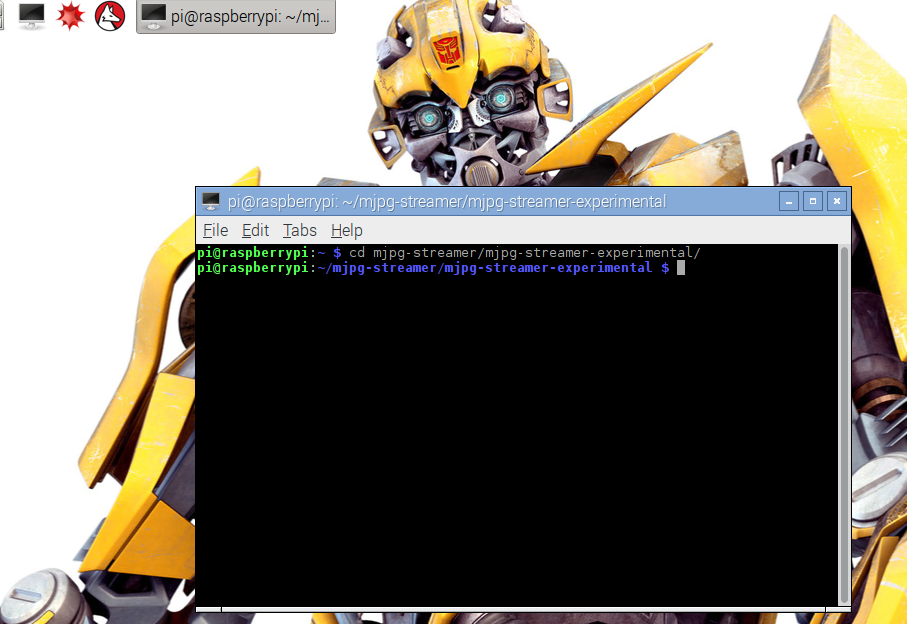


下载好的代码可以通过Filezilla上传到树莓派，也可以通过U盘拷贝到树莓派。软件截图如下



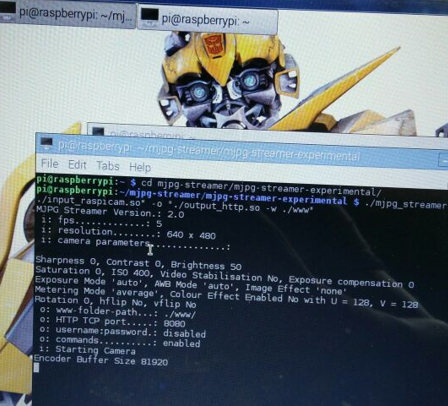
现在进入mjpg-streamer软件所在的目录，开启摄像头。打开终端输入

cd mjpg-streamer/mjpg-streamer-exprimental



进入此目录后输入

./mjpg\_streamer -i "./input\_raspicam.so" -o "./output\_http.so -w ./www" 回车



此时摄像头已经打开，不要关闭此终端窗口。

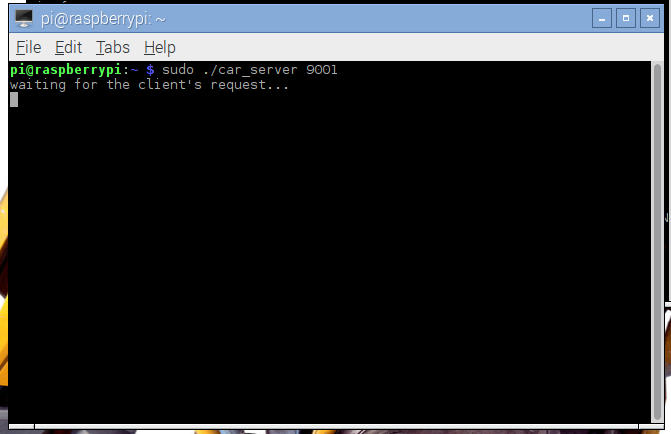
测试时请不要安装小车轮子

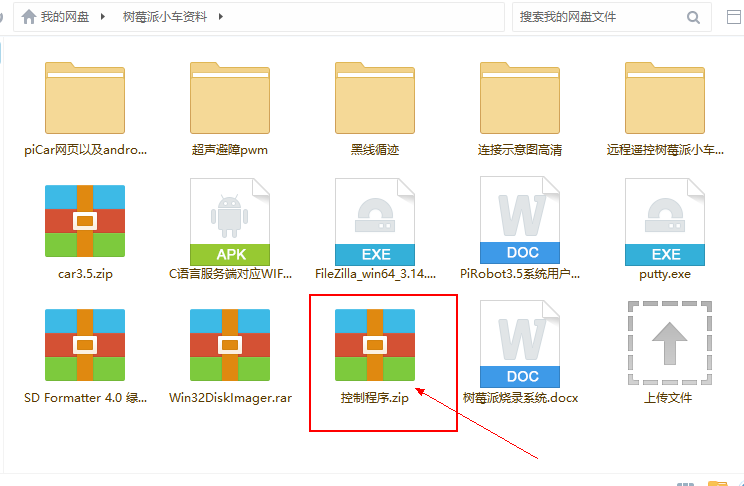
另启动一个终端窗口

系统里的car\_server.c文件已经编译过了，直接运行car\_server 9001是端口号。如果修改了car\_server.c文件那么就编译一下再运行，编译命令为

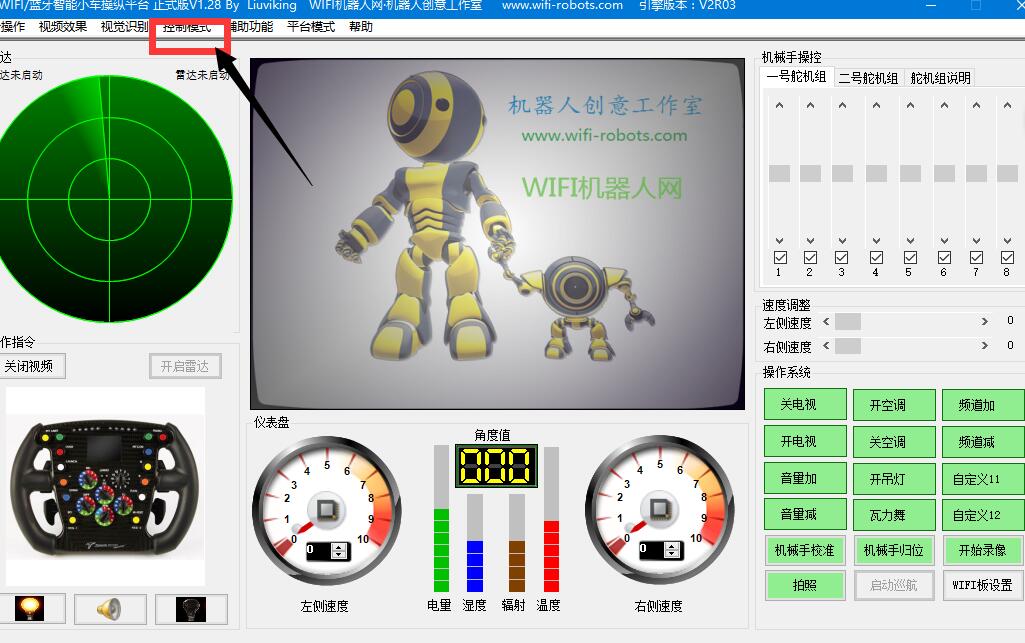
gcc car\_server.c -o car –lwiringPi –lwiringPiDev

运行则输入 sudo ./car\_server 9001





将电脑于树莓派置于同一局域网下，打开控制程序点击开启视频,控制模式选择wifi模式即可。



W——前 A——左

D——右 S——后

此系统开机创建热点，如果不需要，可以修改rc.local文件（sudo nano /ect/rc.local），将sudo ifdown wlan0 以及创建热点的命令删除。之后树莓派可以连WiFi

**WiFi视频远程控制程序Python语言**

此时我们调用python的代码，

注意 python的接线方式与C语言的接线方式不同。IN1 IN2 IN3 IN4分别接在树莓派物理接口的11 12 13 14

在终端进入树莓派根目录下的Server文件夹

cd server

直接运行server.py文件

sudo python server.py

打开安卓软件输入（前提是手机跟树莓派在同一局域网）

192.168.12.1

2000

192.168.12.1：8080



或者打开浏览器；输入地址

192.168.12.1：2000

**常用到的指令合集**

**打开摄像头软件位置**

**cd mjpg-streamer/mjpg-streamer-experimental**

**打开摄像头指令--复制到LX终端**

**./mjpg\_streamer -i "./input\_raspicam.so" -o "./output\_http.so -w ./www"**

**1、WIFI程序car\_server.c**

**编译指令**

**gcc car\_server.c -o car\_server -lwiringPi -lpthread （注意前面有空格 加中间的空格位置）**

**运行命令是**

**sudo ./car\_server 9001**

**2、循迹程序xj.c**

**编译指令**

**gcc xj.c -o xj -lwiringPi -lpthread （注意前面有空格 加中间的空格位置）**

**运行命令是**

**sudo ./xj**

**3、红外避障程序bz.c**

**编译指令**

**gcc bz.c -o bz -lwiringPi -lpthread （注意前面有空格 加中间的空格位置）**

**运行命令是**

**sudo ./bz**

**4、测距程序dis.c**

**编译指令**

**gcc dis.c -o dis -lwiringPi (注意前面有空格)**

**运行命令是 sudo ./cj**

**5、超声避障pwm csb.c**

**编译指令**

**gcc csb.c -o csb -lwiringPi -lpthread （注意前面有空格 加中间的空格位置）**

**运行命令是**

**sudo ./csb**

**停车.c**

**编译指令**

**gcc stop.c -o stop -lwiringPi -lpthread （注意前面有空格 加中间的空格位置）**

**运行命令是**

**sudo ./stop**

**运行编译出来的程序 sudo ./xxx**

**xxx.c 是你要编译的C文件 xxx是编译出来的执行文件**