

互联网思维与技术

实验 6 树莓派智能小车



目 录

1 初识树莓派	1
2 树莓派智能车组装与系统安装.....	2
2.1 树莓派智能车.....	2
2.2 操作系统.....	3
2.3 树莓派的配置.....	4
3 树莓派智能车实践指导.....	5
3.1 小车的供电和使用.....	5
3.2 Linux 基础操作	5
3.3 Python 基础	7
3.3.1 Python 简介	8
3.3.2 Python 中的函数与模块	8
3.3.3 Python 面向对象编程	9
3.4 小车的硬件模块控制.....	10
3.4.1 车轮控制.....	10
3.4.2 按键控制.....	12
3.4.3 初始化函数.....	12
3.4.4 主程序.....	13
3.5 注意事项.....	14

1 初识树莓派

树莓派是什么？树莓派的英文名为 Raspberry Pi（简称为 RPi，或者 RasPi）是为学生计算机编程教育而设计，只有信用卡大小的嵌入式产品，其系统基于 Linux。树莓派主要由系统芯片、CPU、GPU、输入输出原件等组成，且不断迭代更新，如图 1-1 所示。可以把树莓派理解为一台小的“电脑主机”，连接显示器、键盘和鼠标后可以和我们日常生活中的普通电脑一样工作。

树莓派各版本参数比较：RPI Zero、RPI A+、RPI B、RPI B+、RPI2 B、RPI3 B

项目	Zero	A+型	B型	B+型	2代B型	3代B型
SoC(系统芯片)	BCM2835 (CPU, GPU DSP和SDRAM)				BCM2836	BCM2837
CPU(中央处理)	ARM1176JZF-S核心(ARM11系列)700MHz 单核				ARM Cortex-A7 900MHz 四核	ARM Cortex-A53 1.2GHz 四核
GPU(图形处理)	Broadcom VideoCore IV, OpenGL ES 2.0, 1080p 30 h.264/MPEG-4 AVC 高清解码器					
内存	512MB	256MB	512MB		1GB	
USB 2.0	1 (注: Zero为micro USB口)		2	4		
	支持USB hub扩展					
视频输出	RCA视频接口输出 (仅1代B型有此接口), 支持PAL和NTSC制式, 支持HDMI (1.3和1.4), 分辨率为640 x 350 至 1920 x 1200 支持PAL 和NTSC制式。					
音频输出	3.5mm 插孔(Zero无此项), HDMI (高清晰度多音频视频接口)					
SD卡接口	Micro SD卡接口		标准SD卡接口		Micro SD卡接口	
网络介入	没有 (需通过USB)		10/100 以太网接口 (RJ45接口), 3代B型还支持WiFi和蓝牙。			
扩展接口	40		26		40	
额定功率	未知, 但更低		700毫安(为3.5 W)		600毫安(为3.0 W)	1000毫安(为5.0 W)
电源输入	5v, 通过MicroUSB或GPIO引脚					
总体尺寸	65 x 30 x 5 mm	65 x 56 mm	85.60 x 53.98 mm		85 x 56 x 17 mm	
操作系统	Debian GNU/Linux、Fedora、Arch Linux、RISC OS 2代B型和3代B型还支持Windows10和Snappy Ubuntu Core, 官方会持续更新以支持更多操作系统, 敬请期待!					



图 1-1 树莓派的组成和不同版本

树莓派可以用来做很多意想不到的事情：<http://shumeipai.nxez.com/what-raspi-used-for>。

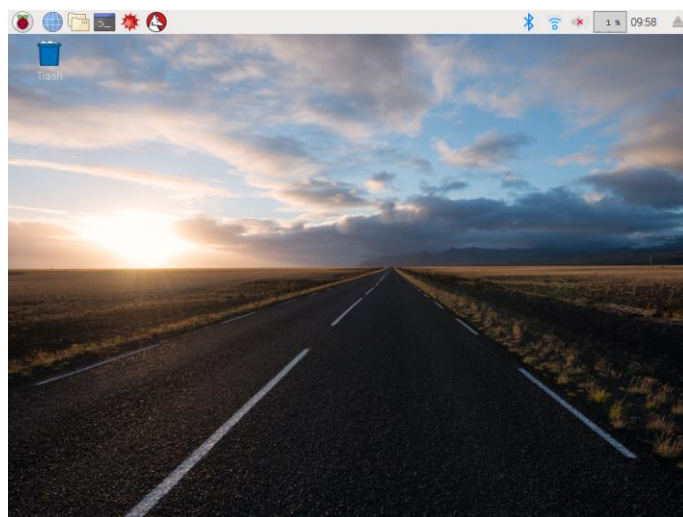


图 1-2 树莓派桌面

2 树莓派智能车组装与系统安装

2.1 树莓派智能车

本次实验为大家提供了智能车硬件、配套代码以及全套视频教程。其中硬件部分组装后成品图如下：

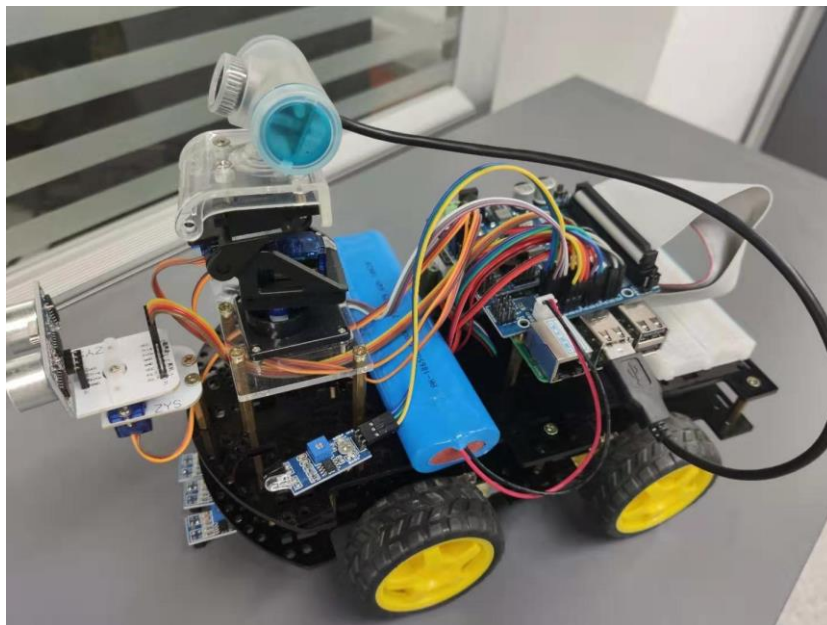


图 2-1 树莓派智能车

提供的配套代码包列表如下：

- 1、蜂鸣器实验
- 2、按键启动程序
- 3、前进后退左转右转
- 4、循迹实验
- 5、红外壁障实验
- 6、超声波避障程序（不带摇头）
- 7、超声波避障（带舵机）
- 8、红外遥控实验
- 9、IIC LCD1602液晶显示实验（扩展 选配）

图 2-2 配套代码包

同时也为大家配套提供了树莓派相关书籍，用作开发时的参考，配套提供的书籍清单如下：

 Raspberry Pi Python 编程入门 2014.02.pdf	2015/11/13 16:22	PDF 文件	22,469 KB
 Raspberry_Pi树莓派实作应用_PDF电子书下载 ...	2015/7/11 22:20	PDF 文件	7,724 KB
 Raspberry_Pi用户指南_ (英) EBEN_UPTON...	2015/11/13 16:34	PDF 文件	20,979 KB
 wiringPi用户手册V001.pdf	2017/9/24 16:49	PDF 文件	765 KB
 树莓派 (Raspberry+Pi) 实战指南+手把手教...	2017/8/29 11:42	PDF 文件	38,750 KB
 树莓派编程入门.pdf	2017/8/29 11:45	PDF 文件	14,301 KB
 玩转树莓派 RASPBERRY PI 王江伟, 刘青 北京...	2016/11/17 16:27	PDF 文件	21,687 KB

图 2-2 配套书籍清单

同学们按照组装视频组装智能车即可，所需的一切材料均在材料箱中。

注意 1：需严格检查插线，尤其是电源插线。反插可能会烧毁树莓派或引发电池过热。

注意 2：使用智能车时需注意电压表上显示的电池电压，当电压低于 7v 时需要开始充电，以避免过放。

注意 3：给电池充电时需注意关闭电路开关。

2.2 操作系统

树莓派没有硬盘，取而代之的是 TF 卡。只需一个电源一张 TF 卡即可启动树莓派，操作系统[官网下载地址](http://www.raspbian.org)。

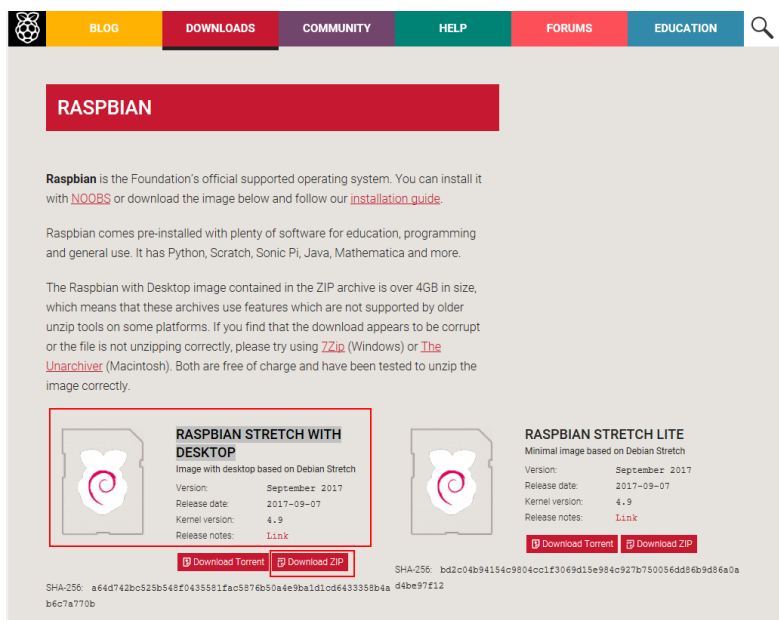


图 2-4 树莓派操作系统下载地址

对于初学者建议推荐选择 Raspbian Stretch with desktop，这个操作系统与 Linux 类似，带图形界面，解压后约 4GB 左右；而 Raspbian Stretch LITE 是裁剪过的轻量系统，没有图形界面，解压后约 1.3GB 左右。

具体系统安装过程可参见：

https://blog.csdn.net/weixin_41656968/article/details/79592624

注：若烧录过程中始终提示“Error5: 拒绝访问”，则需对 tf 卡的分区进行删除。详可参见：

<https://www.jianshu.com/p/47a5c8e2fd5b>

2.3 树莓派的配置

树莓派 `raspi-config` 设置。第一次使用树莓派的时候需要进行一些简单的配置，在终端运行如下命令进入配置界面。

```
sudo raspi-config
```

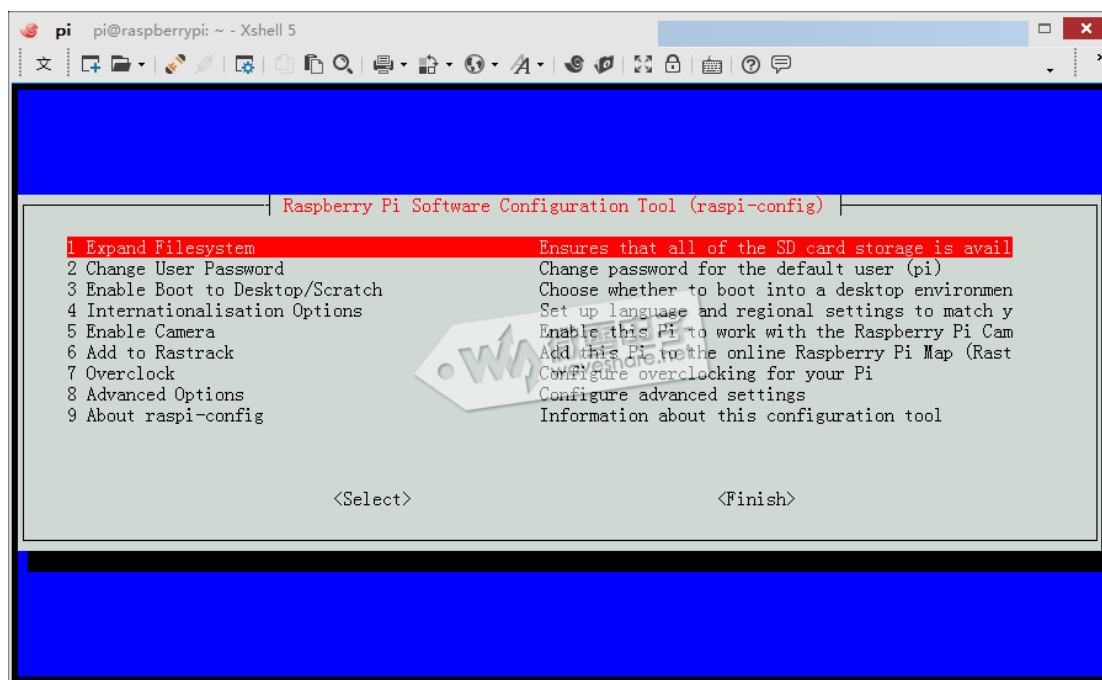


图 2-5 树莓派配置界面

Internationalisation Options 国际化选项

I1 Change Timezone

设置时区，如果不进行设置，Pi 的时间就显示不正常。

I2 Change Keyboard Layout

改变键盘布局,国内多使用 PC104 标准键盘，一般设置如下：

- >Generic 104-key PC
- >English(US)
- >The default for the keyboard layout
- >No compose key
- >ctrl_alt_backspace Yes

linux 系统常用 apt(Advanced Package Tool)高级软件工具来安装软件、



```
sudo apt-get install xxx  安装软件
sudo apt-get update  更新软件列表
sudo apt-get upgrade  更新已安装软件
sudo apt-get remove xxx  删除软件
```

在使用 **apt** 前，修改镜像源为国内阿里云镜像源，否则下载更新会很慢。具体做法如下：

1) 编辑 `/etc/apt/sources.list` 文件。这里推荐用 **vim** 命令编辑。命令如下：

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak  # 备份为 sources.list.bak
sudo vim /etc/apt/sources.list  # 编辑 sources.list 文件
```

2) 将网址换为 <http://mirrors.aliyun.com/raspbian/raspbian/>

我们需要安装 **python** 图像处理包 **opencv**，启用终端输入以下代码：

```
sudo apt-get install libopencv-dev
sudo apt-get install python-opencv
```

3 树莓派智能车实践指导

3.1 小车的供电和使用

小车有两种供电方式，一种是有线供电，使用变压器将 220V 交流转 5V 直流进行供电；另一种是电池供电，在小车外接锂电池，打开车顶的开关，可启动小车。树莓派的使用有两种方式：

- (1) 连接显示器、键盘和鼠标等设备，用于显示和输入。关闭系统需要点击左上角的树莓派图标，选择“shutdown”选项。
- (2) 连接笔记本电脑，详可参见：

<https://blog.csdn.net/u013895577/article/details/79579161>

3.2 Linux 基础操作

树莓派系统多都是基于 **Linux**，而 **Linux** 多数情况下都是在命令行下输入命令操作。在树莓派中，点击左上角的终端图标可以打开终端，

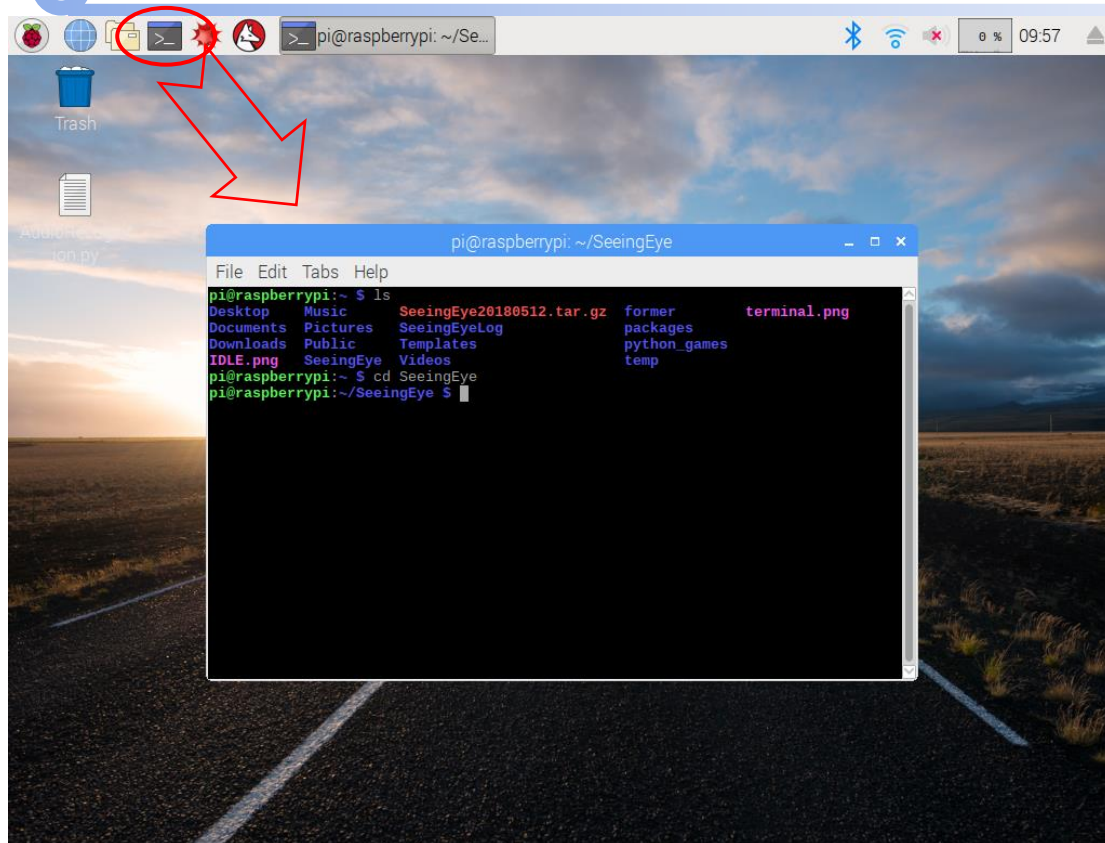


图 3-1 打开终端

在终端中，可以输入控制命令。例如，输入“pwd”查看当前终端所在目录，输入“cd+目录”可以进入指定目录页，在终端中还可以输入命令安装 python 包或安装需要的软件，例如，输入“pip install sklearn”安装 sklearn 机器学习包，输入“apt-get install python-opencv”安装 opencv。在命令行之前行添加“sudo”可以执行超级权限。

Unix/Linux 命令参考

FOSSwire.com

文件命令	系统信息
ls - 列出目录	date - 显示当前日期和时间
ls -al - 使用格式化列出隐藏文件	cal - 显示当月的日历
cd dir - 更改目录到 dir	uptime - 显示系统从开机到现在所运行的时间
cd - 更改到 home 目录	w - 显示登录的用户
pwd - 显示当前目录	whoami - 查看你的当前用户名
mkdir dir - 创建目录 dir	finger user - 显示 user 的相关信息
rm file - 删除 file	uname -a - 显示内核信息
rm -r dir - 删除目录 dir	cat /proc/cpuinfo - 查看 cpu 信息
rm -f file - 强制删除 file	cat /proc/meminfo - 查看内存信息
rm -rf dir - 强制删除目录 dir *	man command - 显示 command 的说明手册
cp file1 file2 - 将 file1 复制到 file2	df - 显示磁盘占用情况
cp -r dir1 dir2 - 将 dir1 复制到 dir2; 如果 dir2 不存在则创建它	du - 显示目录空间占用情况
mv file1 file2 - 将 file1 重命名或移动到 file2; 如果 file2 是一个存在的目录则将 file1 移动到目录 file2 中	free - 显示内存及交换区占用情况
ln -s file link - 创建 file 的符号连接 link	
touch file - 创建 file	
cat > file - 将标准输入添加到 file	
more file - 查看 file 的内容	
head file - 查看 file 的前 10 行	
tail file - 查看 file 的后 10 行	
tail -f file - 从后 10 行开始查看 file 的内容	
进程管理	压缩
ps - 显示当前的活动进程	tar cf file.tar files - 创建包含 files 的 tar 文件 file.tar
top - 显示所有正在运行的进程	tar xf file.tar - 从 file.tar 提取文件
kill pid - 杀掉进程 id pid	tar czf file.tar.gz files - 使用 Gzip 压缩创建 tar 文件
killall proc - 杀掉所有名为 proc 的进程	tar xzf file.tar.gz - 使用 Gzip 提取 tar 文件
bg - 列出已停止或后台的作业	tar cjf file.tar.bz2 - 使用 Bzip2 压缩创建 tar 文件
fg - 将最近的作业带到前台	tar xjf file.tar.bz2 - 使用 Bzip2 提取 tar 文件
fg n - 将作业 n 带到前台	gzip file - 压缩 file 并重命名为 file.gz
	gzip -d file.gz - 将 file.gz 解压缩为 file
文件权限	网络
chmod octal file - 更改 file 的权限	ping host - ping host 并输出结果
<ul style="list-style-type: none"> 4 - 读 (r) 2 - 写 (w) 1 - 执行 (x) 	whois domain - 获取 domain 的 whois 信息
示例:	dig domain - 获取 domain 的 DNS 信息
chmod 777 - 为所有用户添加读、写、执行权限	dig -x host - 逆向查询 host
chmod 755 - 为所有者添加 rwx 权限, 为组和其他用户添加 rx 权限	wget file - 下载 file
更多选项参阅 man chmod.	wget -c file - 断点续传
SSH	安装
ssh user@host - 以 user 用户身份连接到 host	从源代码安装:
ssh -p port user@host - 在端口 port 以 user 用户身份连接到 host	./configure
ssh-copy-id user@host - 将密钥添加到 host 以实现无密码登录	make
	make install
	dpkg -i pkg.deb - 安装包 (Debian)
	rpm -Uvh pkg.rpm - 安装包 (RPM)
搜索	快捷键
grep pattern files - 搜索 files 中匹配 pattern 的内容	Ctrl+C - 停止当前命令
grep -r pattern dir - 递归搜索 dir 中匹配 pattern 的内容	Ctrl+Z - 停止当前命令, 并使用 fg 恢复
command grep pattern - 搜索 command 输出中匹配 pattern 的内容	Ctrl+D - 注销当前会话, 与 exit 相似
	Ctrl+W - 删除当前行中的字
	Ctrl+U - 删除整行
	!! - 重复上次的命令
	exit - 注销当前会话
	* 小心使用。
	翻译/Toy <http://LinuxTOY.org>

图 3-2 Linux 常用命令参考

3.3 Python 基础

为了使同学们更快上手, 这里简单介绍了 Python 最基本的原理和操作, 方便大家实现想要的功能。计算机程序的编写由顺序、选择和循环三种结构组成, 本实验中, 使用 Python2 实现所需的功能。

3.3.1 Python 简介

树莓派智能车的控制是通过 Python 编程语言实现的，Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。Python 的设计具有很强的可读性，相比其他语言经常使用英文关键字，其他语言的一些标点符号，它具有比其他语言更有特色语法结构，是目前数据挖掘和人工智能等领域最流行的语言之一。

树莓派中 Python 的编辑需要使用 IDLE 软件。系统安装完成后自带 Python 的 IDLE- Thonny，其界面如下图所示。

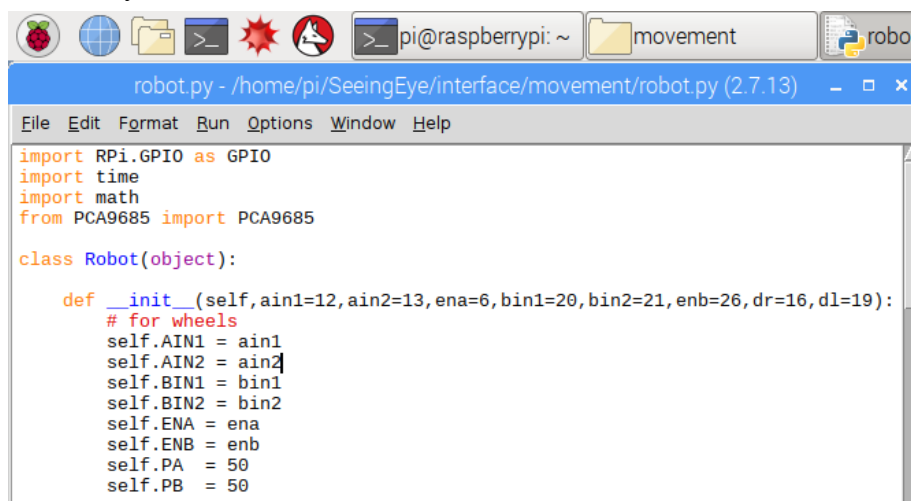


图 3-3 树莓派的 Python 编辑器 IDLE

3.3.2 Python 中的函数与模块

函数是组织好的，可重复使用的，用来实现单一，或相关联功能的代码段。函数能提高应用的模块性，和代码的重复利用率。你已经知道 Python 提供了许多内建函数，比如 `print()`。但你也可以自己创建函数，这被叫做用户自定义函数。

你可以定义一个由自己想要功能的函数，以下是简单的规则：

- 1) 函数代码块以 `def` 关键词开头，后接函数标识符名称和圆括号()。
- 2) 任何传入参数和自变量必须放在圆括号中间。圆括号之间可以用于定义参数。
- 3) 函数的第一行语句可以选择性地使用文档字符串—用于存放函数说明。
- 4) 函数内容以冒号起始，并且缩进。
- 5) `return [表达式]` 结束函数，选择性地返回一个值给调用方。不带表达式 `return` 相当于返回 `None`。

例如编写求取两个数均值的函数：

```

def median(a, b):
    c = (a+b)/2
    
```



```
return c
```

调用函数，使用语句：

```
m = median(3,5)
print(m)
```

输出结果为“4”

Python 模块(Module)，是一个 Python 文件，以“.py”结尾，包含了 Python 对象定义和 Python 语句。模块让你能够有逻辑地组织你的 Python 代码段。把相关的代码分配到一个模块里能让你的代码更好用，更易懂。模块能定义函数，类和变量，模块里也能包含可执行的代码。

模块定义好后，我们可以使用 import 语句来引入模块，比如要引用模块 math，就可以在文件最开始的地方用 import math 来引入。在调用 math 模块中的 sin() 函数时可以这样引用：a=math.sin(3.14)。

如果使用自己定义的模块，需要使用其中的包“median”，该包所在的 python 文件名字也是“median”，python 文件的内容为上述函数，该包的功能是求取两个数的均值，那么调用时需要在新的 python 文件中使用的语句：

```
from median import median
c = median(3,5)
print(c)
```

输出结果为“4”。

当你导入一个模块，Python 解析器对模块位置的搜索顺序是：

- 1、当前目录
- 2、如果不在当前目录，Python 则搜索在 shell 变量 PYTHONPATH 下的每个目录。
- 3、如果都找不到，Python 会查看默认路径。UNIX 下，默认路径一般为 /usr/local/lib/python/。模块搜索路径存储在 system 模块的 sys.path 变量中。变量里包含当前目录，PYTHONPATH 和由安装过程决定的默认目录。

3.3.3 Python 面向对象编程

Python 面向对象编程有几个重要的术语：

类(Class): 用来描述具有相同的属性和方法的对象的集合。它定义了该集合中每个对象所共有的属性和方法。对象是类的实例。

数据成员：类变量或者实例变量用于处理类及其实例对象的相关的数据。

方法：类中定义的函数。

对象：通过类定义的数据结构实例。对象包括两个数据成员（类变量和实例

变量) 和方法。

实例化：创建一个类的实例，类的具体对象。

3.4 小车的硬件模块控制

智能小车的可操控硬件包括车轮、相机转动、相机、红外测距传感器等。其中，车轮由四个电机进行控制，每个电机都有两个转动方向，且转动速度可通过程序调节；相机转动模块也由两个舵机控制，性能与车轮舵机相同，但设置了转动范围；相机是图像采集模块，可以采集可调分辨率的图像，并传送树莓派进行处理；红外测距模块能够采集小车前端是否有障碍物，并向树莓派传送布尔信息号，帮助小车完成避障功能。

除以上功能之外，还可以为树莓派进行扩展，例如增加声音采集模块(话筒)和音频输出模块(蓝牙音箱)，使小车能够完成更加智能的功能。

本模块中，我们以树莓派智能车的红外避障行驶功能为例，向大家介绍树莓派智能车的基本控制原理和编程实现方式。

3.4.1 车轮控制

一台小车首先要有电机轮子，通过控制电机转动，小车实现不同方向行走。电机可以是直流电机或者步进电机。步进电机是以步阶方式分段移动，也就是每次只能控制转动一小段。所以步进电机的控制精度会很高，可以精确的控制转动距离，但是转速会比较慢。在控制精度要求很高的情况会用，例如机械手臂等。而直流电机则相反，转速快，但是控制精度相对差。直流电机是通过电压控制，通电后电机就可以转，电压越大转速越快，电流方向决定电机转动方向。智能车采用直流电机控制。

车轮控制的函数如下，使小车针对不同的输入值作出不同的反应。

```
#运动函数
#前进
def t_up(speed,t_time):
    L_Motor.ChangeDutyCycle(speed)#左侧电机速度控制
    GPIO.output(AIN2,False)#AIN2 正转
    GPIO.output(AIN1,True) #AIN1 正转

    R_Motor.ChangeDutyCycle(speed)#右侧电机速度控制
    GPIO.output(BIN2,False)#BIN2 正转
    GPIO.output(BIN1,True) #BIN1 正转
    time.sleep(t_time)#运动时间
```



#停止

```
def t_stop(t_time):  
    L_Motor.ChangeDutyCycle(0)  
    GPIO.output(AIN2,False)#AIN2  
    GPIO.output(AIN1,False) #AIN1  
  
    R_Motor.ChangeDutyCycle(0)  
    GPIO.output(BIN2,False)#BIN2  
    GPIO.output(BIN1,False) #BIN1  
    time.sleep(t_time)
```

#后退

```
def t_down(speed,t_time):  
    L_Motor.ChangeDutyCycle(speed)  
    GPIO.output(AIN2,True)#AIN2  
    GPIO.output(AIN1,False) #AIN1  
  
    R_Motor.ChangeDutyCycle(speed)  
    GPIO.output(BIN2,True)#BIN2  
    GPIO.output(BIN1,False) #BIN1  
    time.sleep(t_time)
```

#左转

```
def t_left(speed,t_time):  
    L_Motor.ChangeDutyCycle(speed)  
    GPIO.output(AIN2,True)#AIN2  
    GPIO.output(AIN1,False) #AIN1  
  
    R_Motor.ChangeDutyCycle(speed)  
    GPIO.output(BIN2,False)#BIN2  
    GPIO.output(BIN1,True) #BIN1  
    time.sleep(t_time)
```

#右转

```
def t_right(speed,t_time):  
    L_Motor.ChangeDutyCycle(speed)  
    GPIO.output(AIN2,False)#AIN2
```



```
GPIO.output(AIN1,True) #AIN1

R_Motor.ChangeDutyCycle(speed)
GPIO.output(BIN2,True)#BIN2
GPIO.output(BIN1,False) #BIN1
time.sleep(t_time)
```

3.4.2 按键控制

```
#按键控制
def keysacn():
    val = GPIO.input(BtnPin)
    while GPIO.input(BtnPin) == False:
        val = GPIO.input(BtnPin)
    while GPIO.input(BtnPin) == True:
        time.sleep(0.01)
        val = GPIO.input(BtnPin)
        if val == True:
            GPIO.output(Rpin,1)
            while GPIO.input(BtnPin) == False:
                GPIO.output(Rpin,0)
        else:
            GPIO.output(Rpin,0)
```

3.4.3 初始化函数

```
def setup():
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)      # Numbers GPIOs by physical location
    GPIO.setup(Gpin, GPIO.OUT)   # Set Green Led Pin mode to output
    GPIO.setup(Rpin, GPIO.OUT)   # Set Red Led Pin mode to output
    GPIO.setup(BtnPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup(SensorRight,GPIO.IN)
    GPIO.setup(SensorLeft,GPIO.IN)

    GPIO.setup(AIN2,GPIO.OUT)
```




```
GPIO.setup(AIN1,GPIO.OUT)
GPIO.setup(PWMA,GPIO.OUT)

GPIO.setup(BIN1,GPIO.OUT)
GPIO.setup(BIN2,GPIO.OUT)
GPIO.setup(PWMB,GPIO.OUT)
```

3.4.4 主程序

主程序，先进行系统初始化，后控制小车根据红外传感器的返回数据进行障碍行驶。当按钮被按下时，终止程序。

```
if __name__ == '__main__':
    setup()#初始化
    keysacn()#按键控制
    L_Motor= GPIO.PWM(PWMA,100)
    L_Motor.start(0)
    R_Motor = GPIO.PWM(PWMB,100)
    R_Motor.start(0)
    try:
        while True:
            SR_2 = GPIO.input(SensorRight)#右侧传感器
            SL_2 = GPIO.input(SensorLeft)#左侧传感器
            if SL_2 == True and SR_2 == True:
                print ("t_up")
                t_up(50,0)#直行
            elif SL_2 == True and SR_2 ==False:
                print ("Left")
                t_left(50,0)#左转
            elif SL_2==False and SR_2 ==True:
                print ("Right")
                t_right(50,0)#右转
            else:
                t_stop(0.3)#停机
                t_down(50,0.4)#后退
                t_left(50,0.5)#转向
```

```
except KeyboardInterrupt: # 使用'Ctrl+C'退出
    GPIO.cleanup()
```

本案例仅以简单功能作课堂示范,该程序也可以用同样的方式在其他 python 文件中调用,这样逐层封装,通过函数的嵌套实现高级的功能,最终搭建一个系统,这也是程序的魅力之一。

3.5 注意事项

使用摄像头模块时,每次调用需要将之前的 Shell 关闭,否则可能出现资源不足的错误,如图所示,这时需要点击左上角的图标重启系统。

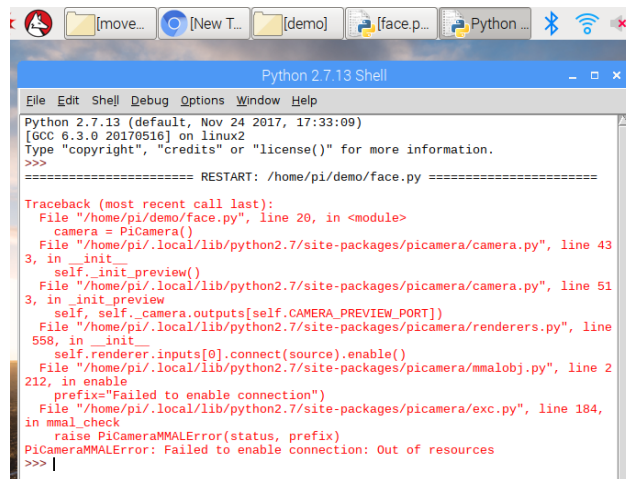


图 3-4 资源不足报错“out of resources”

当需要从树莓派上拷贝文件到 U 盘时,需要将 U 盘格式化,分区方式设置为 FAT32,插入 U 盘后该 U 盘路径在 media/pi/下。

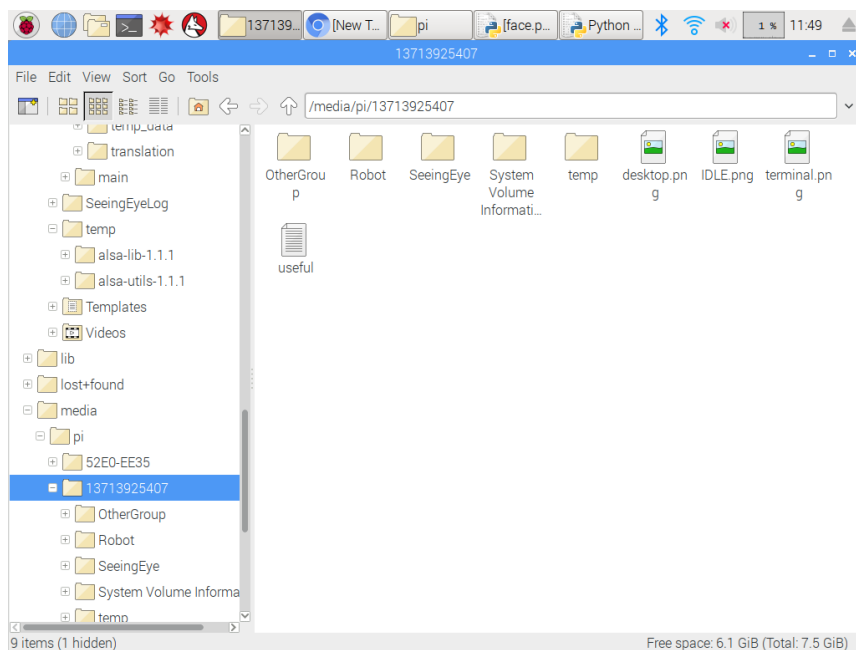


图 3-5 U 盘路径

Python2 包的安装路径在“/usr/lib/python2.7/disk-sitpackages”中，需要安装的包可以在终端使用“pip instgall [包名]”安装。

3.6 异常情况处理办法

若在树莓派使用过程中有任何疑问可参考提供的开发代码和视频教程，或联系助教。

若实验期间发生任何危险情况，包括但不限于：电池异常发热、电路板异常高温、电路板冒烟或起火、电机异常堵转发热等，请立即停止实验并及时联系助教。

若发生实验材料损坏，包括但不限于：树莓派损坏、机器人或智能车结构性损坏、断线、电池过放等，可联系助教维修或更换相应部分。

实验结束后材料将被回收，请大家爱护实验机器人！