树莓派课件

wuchao

树莓派（Raspberry Pi）实战指南——手把手教你掌握100个精彩案例

www.tup.tsinghua.edu.cn/bookscenter/book\_06410801.html#

## 硬件

树莓派概述。应用实例

树莓派3B+的定价是35美元，主要特性如下：

1. 1.4GHz 64位4核 ARM Cortex-A53 CPU
2. 双频 802.11ac 无线网卡和蓝牙 4.2
3. 以太网（千兆以太网 over USB 2.0）
4. 1G LPDDR2
5. PoE 支持（Power-over-Ethernet，with PoE HAT）
6. 改进 PXE 网络与 USB 大容量存储启动

主板

1.树莓派3B+主板 BCM2837B0 型号 CPU

2.散热片

配件

3.主板配套的外壳

4.microSD卡

5.microSD卡的读卡器

6.电源插头及电源线

7.HDMI线

8.飞鼠：无线键盘鼠标二合一（先usb鼠标和键盘 ）

9.树莓派官方摄像头

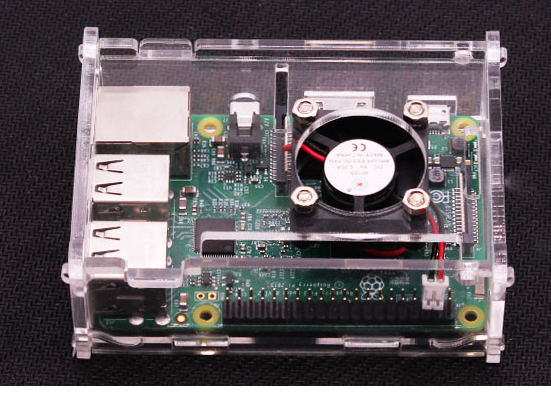
10.传感器套件

11.显示屏

所有外壳安装前都必须把SD卡从树莓派取出，否则安装过程可能会损坏SD卡。

* 散热风扇的接口安装。

第一排的从右往左数第二和第三个接口（注意正负极）





## 系统安装

### 指示灯

1. 正常情况下红色电源灯常亮，绿色信号灯不规律闪烁。
2. 红灯偶尔出现灭的情况可能是供电不足。
3. ACT灯闪烁:SD卡正在活动（类似电脑的硬盘灯）。
4. PWR灯闪烁:电源电压不正常。

### 系统烧录

系统镜像里包含了 bootloader、kernel、文件系统。

树莓派由于其开源特性，支持非常多的系统类型。



树莓派镜像和工具下载地址：

<http://shumeipai.nxez.com/download>

树莓官方系统镜像默认是不开启SSH远程服务的，这样没有屏幕的小伙伴会比较难以使用树莓派。使用开启SSH远程服务的系统镜像，烧录这款镜像以后可直接远程登录树莓派系统。

**准备：**

1. SD卡及读卡器，建议最好4G以上，否则后续开发会使用经常不够用。
2. 专门的格式化工具SDFormatter格式化内存卡。
3. windows系统下安装烧写镜像的工具：Win32DiskImager.zip

**烧写步骤：**

1. 解压下载的系统压缩文件，得到img镜像文件。

2. 将SD使用卡托或者读卡器后，连上电脑。

3. 解压并运行win32diskimager工具。

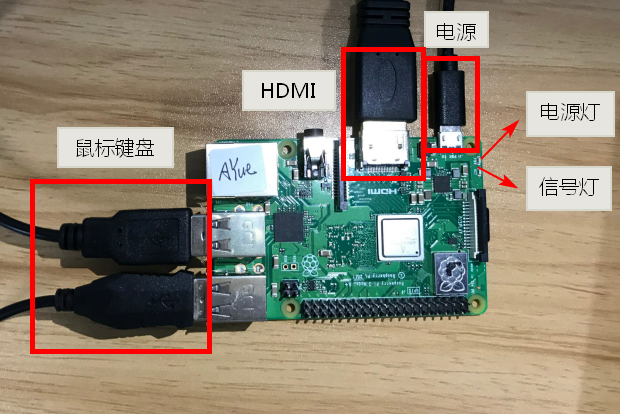
4. 在软件中选择img（镜像）文件，“Device”下选择SD的盘符，然后选择“Write”。

5.安装结束后会弹出完成对话框，说明安装就完成了。

树莓派烧写成功。

* 树莓派启动

**树莓派启动需要几十秒时间，打开电源后1分钟内不可以关闭电源，会影响树莓派的使用寿命。**



烧写完后把SD卡直接插入树莓派，另外连接显示器、电源、鼠标、键盘，打开电源开关以后，就可以进入树莓派系统。

树莓派raspbian系统用户名为pi，密码默认为raspberry；root用户权限密码为raspberry。

**树莓派3种使用方式：**

1. 直接给树莓派外接一个HDMI显示屏和鼠标键盘，然后上电直接操作；
2. 通过SSH或串口来访问树莓派，我们这里借助putty工具；
3. 通过VNC服务远程登录树莓派（需要配置）。

### IP地址设置

修改/etc/network/interfaces文件设置

原先网卡IP是从DHCP服务器获取的

iface eth0 inet dhcp

改为静态IP

iface eth0 inet static

#固定IP地址

address 192.168.1.201

#掩码，可以登录路由器查看

netmask 255.255.255.0

#网关，可以登录路由器查看

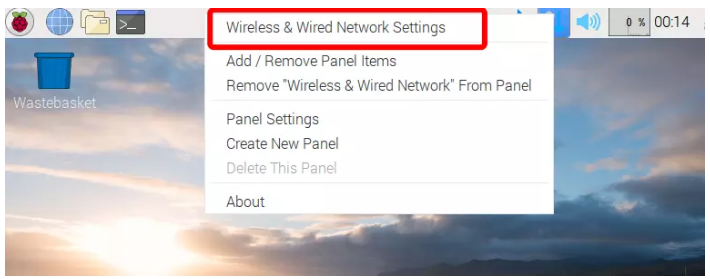
gateway 192.168.1.1

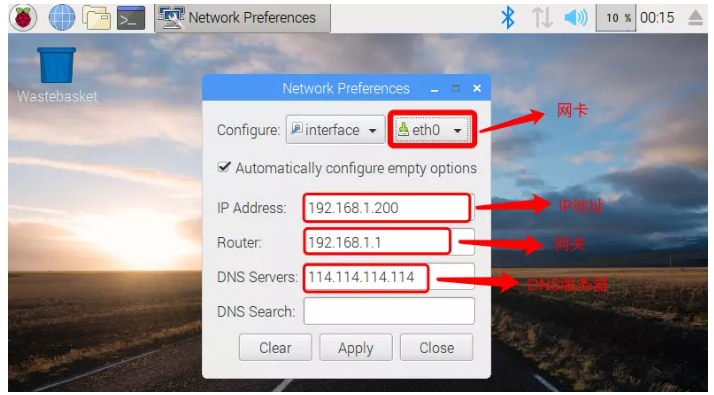
#DNS服务器

dns-nameservers 114.114.114.114

控制面板修改

右键点击菜单栏上面网络的图标





* WIFI设置

1. **在右上角的菜单栏里面选择wifi，输入密码就可以了。**
2. **使用命令行：**

1.查看树莓派已经识别的wifi

每一个cell是一个网络，其中的ESSID是我们关注的，它是wifi的名称，找到自己路由wifi的名字。

sudo iwlist wlan0 scan

2.用nano工具配置wifi信息：

sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf

#添加wifi的名字和密码

network={

ssid="XXXX"

psk="XXXX"

}

## 网络服务

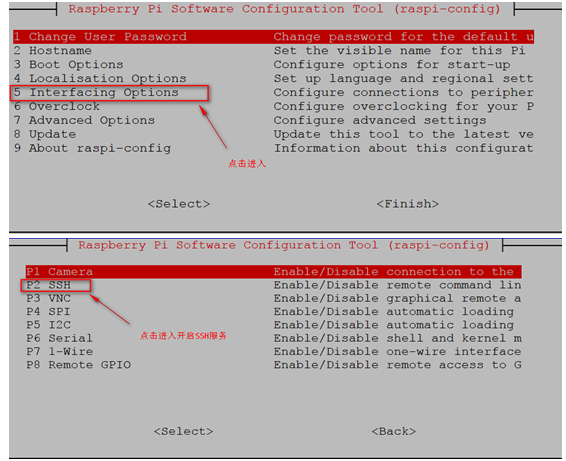
周2

树莓派联网

### 连接SSH

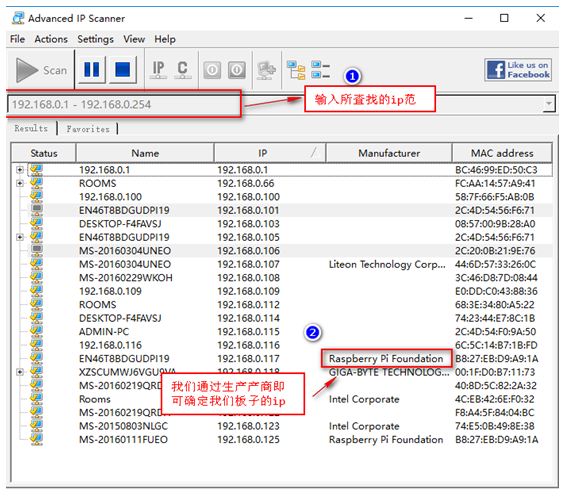
官方的树莓派系统没有开启SSH服务，需要我们开启SSH服务。

需要在命令行终端上输入sudo raspi-config进入到树莓派系统配置界面。



现在可以通过SSH服务远程登录树莓派操作系统。下次启动时可能由于路由器采取的是DHCP自动分配ip地址。

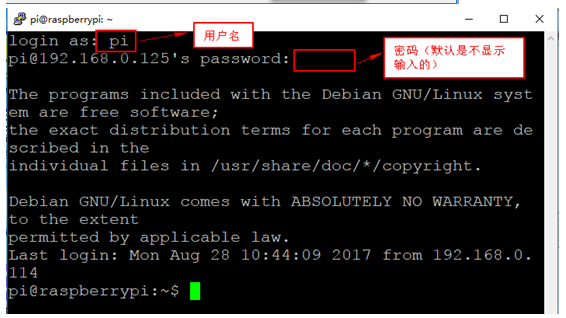
Advanced IP Scanner来查找我们树莓派的ip地址。



Ssh客户端使用

通过putty软件SSH服务远程登录树莓派操作系统。





### ssh设置

生成SSH密钥过程：

1.查看是否已经有了ssh密钥：cd ~/.ssh

如果没有密钥则不会有此文件夹，有则备份删除。

$ git config --global user.name "yourname"

$ git config --global user.email "youremail"

2.生存密钥：

$ ssh-keygen -t rsa -C “haiyan.xu@gmail.com”

如果生成到本地只需要: $ ssh-keygen -t rsa -C

按3个回车，密码为空。

最后得到了两个文件：id\_rsa和id\_rsa.pub

3.添加密钥到ssh：ssh-add 文件名

需要之前输入密码。

4.在github上添加ssh密钥，这要添加的是“id\_rsa.pub”里面的公钥。

打开https://github.com/ ，登陆，然后添加ssh。

### 使用VNC

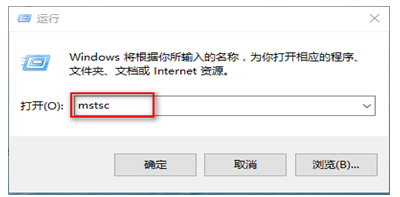
在树莓派命令行下

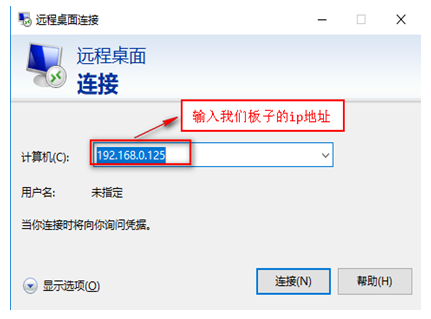
sudo apt-get install xrdp

sudo apt-get install tightvncserver

reboot重启

在windows下的运行界面输入mstsc来打开远程桌面





成功的登录界面：



## 系统管理

周3

### 树莓派设置

Raspberry pi 系统中的Raspbian默认用户是pi 密码为raspberry

sudo passwd root

系统会提示输入两遍的root密码，输入你想设的密码即可

root用户解锁

sudo passwd --unlock root

* **把根目录扩大到整个SD卡**

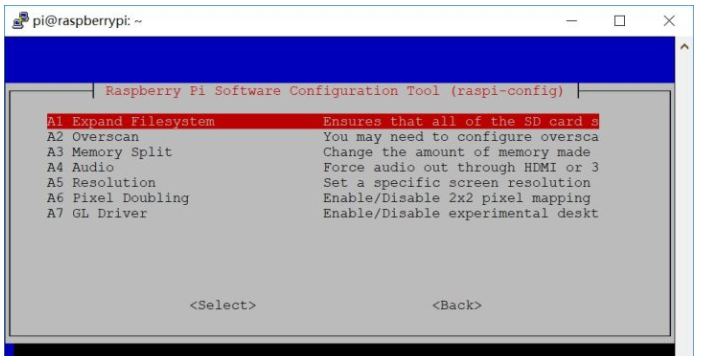
sudo raspi-config

Advanced Options

选择Expand Filesystem，将根目录扩展到这个SD卡，充分利用SD卡的存储空间。

退出设置界面，重启树莓派。

sudo reboot



中文输入法

sudo apt-get install ttf-wqy-zenhei

将安装文泉驿的开源中文字体

sudo apt-get install scim-pinyin

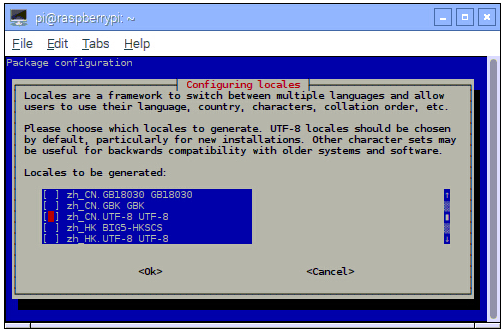
Linux下的中文输入法SCIM

快捷键是Ctrl+空格。或者直接点击屏幕右上角的输入法图标选择。

配置中文操作系统

sudo raspi-config

选择Internationalisation Options-->Change Locale-->zh\_CN.UTF-8-->OK，然后重启系统，设置为中文操作系统。



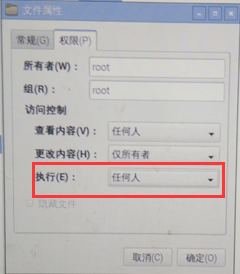
树莓派Raspbian怎样禁用休眠。

/etc/profile.d文件夹，新建screen.sh文件

xset dpms 0 0 0

xset s off

为该文件(screen.sh)增加所有人可执行权限，重新登录Raspbian就可以生效了。



### 树莓派源管理

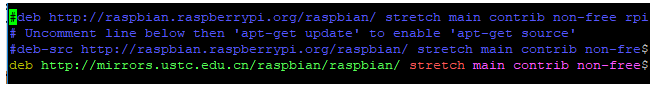
1. 修改sources.list文件。
2. 修改raspi.list 文件。

sudo nano /etc/apt/sources.list

将初始化中的代码中默认的官方软件源注释掉 # 添加下面的软件源（中国科技大学的软件源 ）

deb http://mirrors.ustc.edu.cn/raspbian/raspbian/ stretch main contrib non-free rpi

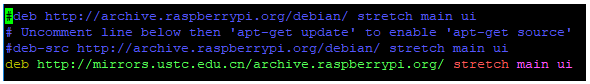
保存并退出：Ctrl+X 按Y（保存数据） 再按回车



sudo nano /etc/apt/sources.list.d/raspi.list

将初始化中的代码中默认的官方软件源注释掉 # 添加下面的软件源

deb http://mirrors.ustc.edu.cn/archive.raspberrypi.org/ stretch main ui



1. 更新本地的软件源

sudo apt-get update

sudo apt-get dist-upgrade -y

### 树莓派安装软件

树莓派是基于Debian的系统

* apt-get管理器

apt-get工具主要是处理软件包的各种依赖关系并获取软件安装过程需要的各种安装包，dkpg负责实际的软件安装工作。

# 安装软件

sudo apt-get install xxx

# 卸载软件

sudo apt-get remove xxx

# 清除无用软件包

sudo apt-get autoremove

# 更新软件源的软件包信息，在配置新的软件源后运行

sudo apt-get update

搜索软件包 apt-cache search softname1

* dpkg命令

dpkg -l 查看系统中安装的所有软件。

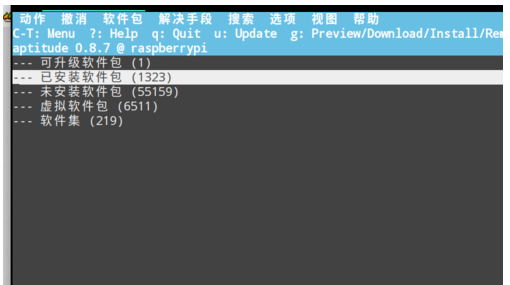
dpkg -l | grep vim

dpkg -L vim 要查看已安装软件的安装位置。

* aptitude管理工具

查看已安装的软件，直接通过方位键选择“已安装软件包”，按回车键直接就可以查看到所有的已安装软件包。

查看单个软件可以通过搜索命令来进行搜索。



问题：树莓派如何应用，解决实际问题。

## Python基础

python语言的优点：

1. 简单：Python是一种代表简单主义思想的语言。
2. 易学
3. 速度快：Python 的底层是用 C 语言写的
4. 免费、开源：Python是FLOSS（自由/开放源码软件）之一
5. 高层语言：无需考虑很多底层细节。
6. 可移植性
7. 面向对象：Python既支持面向过程的编程也支持面向对象的编程。
8. 可嵌入性：可以把Python嵌入C/C++程序，从而向程序用户提供脚本功能。
9. 规范的代码：Python采用强制缩进的方式使得代码具有较好可读性。

Python3 在线速查手册

<https://www.w3xue.com/manual/py3/>

wiki.ubuntu.org.cn/Python\_官方简明教程





test1.py

*# Filename : test.py 生成随机数***import** random  
  
**for** j **in** range(0,3):  
 print(random.randint(0, 50))

**语法小结**

变量类型

赋值

字符串

格式化字符串

三引号

字符串内建函数

数据格式

列表

列表函数方法

元组

字典

内置方法

map

条件语句

if

switch case实现

循环语句

While

for

函数

参数传递

参数

必备参数

关键字参数

缺省参数

不定长参数

匿名函数

函数装饰器

模块

import 语句

文件IO

标准输入输出

打印到屏幕

读取键盘输入

raw\_input函数

input函数

文件操作

open

write

read

文件定位

重命名和删除文件

目录

文件方法

面向对象

类

类的属性

垃圾回收

继承

重载

类属性与方法

类的私有属性

类的方法

类的私有方法

异常

正则表达式

---



### 安装Python3

python依赖环境

$ sudo apt-get install build-essential libsqlite3-dev sqlite3 bzip2 libbz2-dev

下载python3.6版本源码并解压

$ wget https://www.python.org/ftp/python/3.6.1/Python-3.6.1.tgz

$ tar zxvf Python-3.6.1.tgz

$ cd Python-3.6.1

编译安装

$ sudo ./configure && sudo make && sudo make install

建立软连接

pi@raspberrypi:~ $ python --version

Python 3.4.2

pi@raspberrypi:~ $ python3 --version

Python 3.6.1

pi@raspberrypi:~ $ which python

/usr/bin/python

pi@raspberrypi:~ $ which python3

/usr/local/bin/python3

pi@raspberrypi:~ $ sudo mv /usr/bin/python /usr/bin/python3.4.2

pi@raspberrypi:~ $ sudo ln -s /usr/local/bin/python3 /usr/bin/python

测试,树莓派上的python已经成功升级到Python3.6.1

pi@raspberrypi:~ $ python --version

Python 3.6.1

### 开发IDE



PyCharm是一种Python IDE，其带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如， 调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制等等。

Pycharm 专业版

使用SSH连接树莓派

打开File -> Settings -> Project Interpreter，点旁边的小齿轮，选择Add Romote,进入环境配置。

选择SSH Credentials，Host中填写树莓派的IP地址，User name 和 Password默认为pi 和 raspberry



**配置上传路径**

打开Tool -> Deployment -> Configure,在打开的页面中Type选择SFTP协议，然后继续填写树莓派的IP地址以及用户名和密码，其余默认。

点开Mapping选项卡，配置工程目录，这里主要是设置Deployment path on server，该选项会将电脑上的文件上传到树莓派的对应位置，一般建议目录为/Desktop/py，其他可以默认。

上传工程

点击Tool -> Upload to xxx,点击调试按钮自动安装调试插件，安装完成后自动开始调试。

这时候你就可以看见你的工程已经自动上传到树莓派中。

## 编程环境（c/c++ 选学）

周4

### 编译器gcc

GCC 的意思也只是 GNU C Compiler 而已。经过了这么多年的发展，GCC 已经不仅仅能支持 C 语言；它现在还支持 Ada 语言、C++ 语言、Java 语言、Objective C 语言、Pascal 语言、COBOL语言等等。

安装gcc

apt-get install gcc g++

GCC编译程序

* 1. 预处理（Pre-Processing）
  2. 编译（Comping）
  3. 汇编（Assembing）
  4. 连接（Linking）

GCC基本用法

gcc [optiopn] [filenames]

[option]编译的选项

-o 目标文件

-s 汇编语言文件

-c只编译,不连接可执行文件生成.o

-g产生调试工具所需要的信息,如果要进行调试,就必须加此参数.

-Wall 生成警告信息

-w 不生成警告信息

-E：仅执行编译预处理；

-S：将C代码转换为汇编代码；

[filename] 要编译的文件名

测试

新建文件 hello.c

#include <stdio.h>

int main(int argc, char \*argv[])

{

printf("Hello World!\n");

return 0;

}

编译

gcc -o hello hello.c

hello.c文件编译为hello文件

运行：

./hello

终端输出：Hello World!

### 嵌入式工程

一种经典的嵌入式工程文件目录



小灯闪烁工程组织结构



**编辑工具vim**

**调试工具gdb**

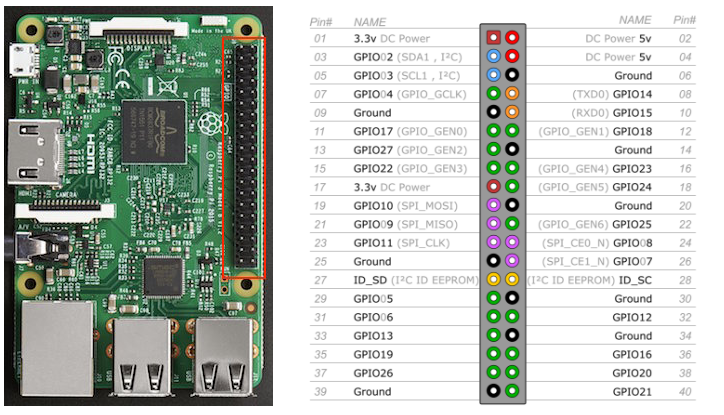
## 树莓派GPIO

周5

在树莓派3上，GPIO接口由40个针脚（PIN）组成。每个针脚都可以用导线和外部设备相连。

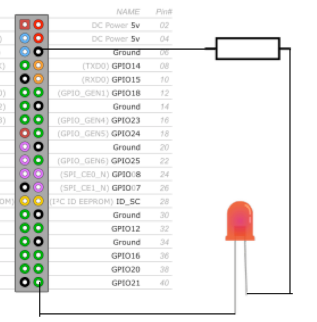
这组GPIO接口大大拓展了树莓派的能力。GPIO不仅能实现通信，还能直接控制电子元器件，从而让用户体验到硬件编程的乐趣。

40个PIN中，有固定输出的5V（2、4号PIN）、3.3V（1、17号PIN）和地线（Ground，6、9、14、20、25、30、34、39）。如果一个电路两端接在，5V和地线之间，该电路就会获得5V的电压输入。27和28号PIN标着ID\_SD和ID\_SC。它们是两个特殊的PIN。它们属于ID EEPROM (Electrically Erasable Programmable Read-Only Memory) 接口，用于和拓展树莓派功能的附加电路板通信。其他的PIN大多编程GPIOX的编号，如GPIO14。树莓派的操作系统中，会用GPIO的编号14来指代这个PIN，而不是位置编号的8。有一些PIN除了GPIO功能外，还提供了高级端口功能。比如说，GPIO14和GPIO15就同时可以充当UART端口。此外，GPIO上还能找到I2C和SPI端口。



### 控制LED灯

我们在GPIO21和地线之间接了一个串联电路。电路上有一个LED灯，还有一个用于防止短路的330欧电阻。当GPIO21位于高电平时，将有电流通过电路，从而点亮LED灯。



在Linux中，外部设备经常被表示成文件。向文件写入或读取字符，就相当于向设备输出或者从设备输入字符。树莓派上的GPIO端口也是如此，其代表文件位于/sys/class/gpio/下。

激活GPIO21:

echo 21 > /sys/class/gpio/export

/sys/class/gpio/下面增加了代表GPIO21的一个目录，目录名就是gpio21。

echo out > /sys/class/gpio/gpio21/direction

文件/sys/class/gpio/gpio21/direction用于控制GPIO21的方向。把GPIO21置于输出状态。

echo 1 > /sys/class/gpio/gpio21/value

LED灯亮了起来。向GPIO21写入1，从而让PIN处于高电压。

如果想关掉LED灯。

echo 0 > /sys/class/gpio/gpio21/value

使用完毕GPIO21，可以删除该端口：

echo 21 > /sys/class/gpio/unexport

/sys/class/gpio/gpio21随即消失。

* python编程

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。

安装 python-dev

sudo apt-get install python-dev

$ sudo apt-get update

$ sudo apt-get dist-update

$ sudo apt-get install python-rpi.gpio

源程序中安装

$ wget http://raspberry-gpio-python.googlecode.com/files/RPi.GPIO-0.6.3.tar.gz

$ sudo tar -zxvf RPi.GPIO-0.6.3.tar.gz

$ cd RPi.GPIO-0.6.3

$ python setup.py install

控制led.py

# -\*- coding: utf-8 -\*-

import RPi.GPIO as GPIO

import time

pin = 40 ## 使用40号引脚

GPIO.setmode(GPIO.BOARD) ## 使用BOARD引脚编号，此外还有 GPIO.BCM

GPIO.setup(pin, GPIO.OUT) ## 设置40号引脚输出

while: ## 重复

GPIO.output(pin, GPIO.HIGH) ## 打开 GPIO 引脚（HIGH）

time.sleep(1) ## 等1秒

GPIO.output(pin, GPIO.LOW) ## 关闭 GPIO 引脚（LOW）

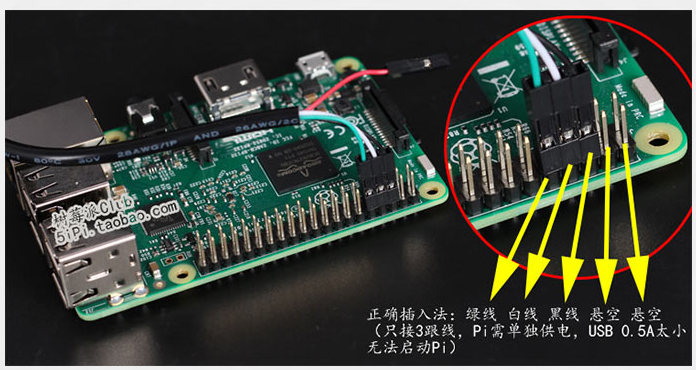
time.sleep(1) ## 等1秒

PIO.cleanup() ## 清除

### 串口



树莓派 PL2303TA USB to TTL 转接线



usb转ttl线（淘宝上很多，10几块钱一根）连接树莓派与电脑，即可通过串口控制调试树莓派。

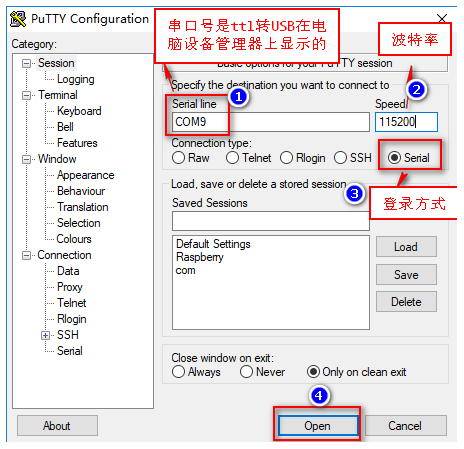
使用的是putty工具

用户名和密码：

默认用户名：pi

默认密码：raspberry

即可成功登录。



## 摄像头

周6

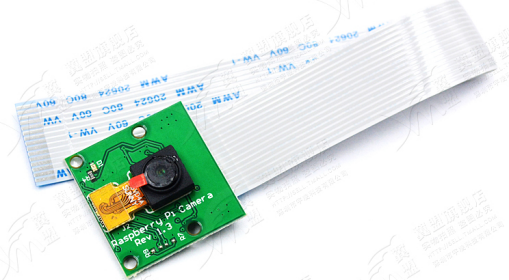
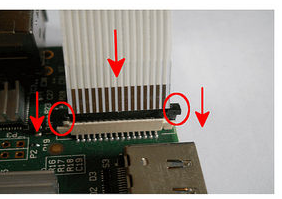
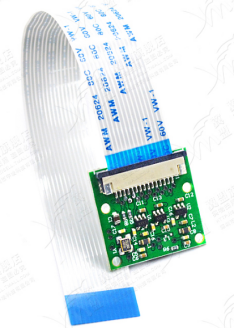
### 拍照和录制影片

* 硬件连接



把摄像头的排线插入到网口和HDMI口之间的排线插槽，银色亮面朝向 HDMI口。

需要先把树莓派板载的排线插槽的扣子拨开，才能插入排线。排线需要紧密插入排线插槽，同时应注意避免排线折弯。排线插入之后，需要把插槽的扣子重新扣上。

* 配置摄像头

1. 使用 raspi-config 配置摄像头。

sudo raspi-config

光标移动到 Camera 项，回车。



1. 启用树莓派摄像头：

Enable support for Raspberry Pi camera?

选择：Enable

1. 重启系统：

Would you like to reboot now?

选择：Yes

* 拍照和摄像

配置并且连接完摄像头，只要给树莓派上电即可使用拍照和摄像功能。

拍照请终端执行以下语句

raspistill -o image.jpg

摄像请终端执行以下语句

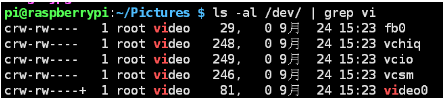
raspivid -o video.h264 -t 10000

其中 -t 10000 表示录制 10 秒，用户各根据自己的需要进行调整。

* 检查故障

查看/dev下面是否存在摄像头设备。

ls -al /dev/ | grep video



摄像头还是不能正常的使用或者驱动，请先检查硬件的连接的问题，可能是排线没有很好的插稳，或者是摄像头本身的问题。

没有/dev/video0那么在/etc/modules中添加bcm2835-v4l2

sudo nano /etc/modules

在文件末尾添加，注意v4l2里面是小写字母L

bcm2835-v4l2

### USB摄像头

1. 安装motion程序

$ sudo apt-get install motion

1. 配置motion程序

sudo vim /etc/default/motion

把no改成yes，开启motion的daemon一直检测设备。

$ sudo vim /etc/motion/motion.conf

把daemon off改成daemon on

确认视频流的接口是8081

把stream\_localhost on改成stream\_localhost off，关闭localhost本地的限制。

把sdl\_threadnr注释掉。

保存文件，退出。

1. 启动motion程序的daemon

$ sudo motion

只要这个motion一直开着，就支持热插拔。

关闭重启motion：

$ sudo killall -TERM motion

$ sudo motion

1. 打开浏览器查看树莓派的摄像头影像

地址是：http://树莓派IP地址:8081

### USB驱动

摄像头usb驱动源码

Camera Usb driver

<\*> USB Video Class (UVC)

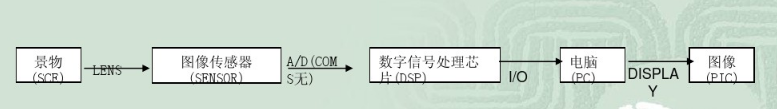
[\*] UVC input events device support

GSPCA是一个万能摄像头驱动程序。

设备名称是：/dev/video0

* 工作原理

景物通过镜头(LENS)生成的光学图像投射到图像传感器表面上，然后转为电信号，经过A/D(模数转换)转换后变为数字图像信号，再送到数字信号处理芯片(DSP)中加工处理，再通过USB接口传输到电脑中处理，通过显示器就可以看到图像。



硬件上来说主要是：镜头、主控芯片与感光芯片。



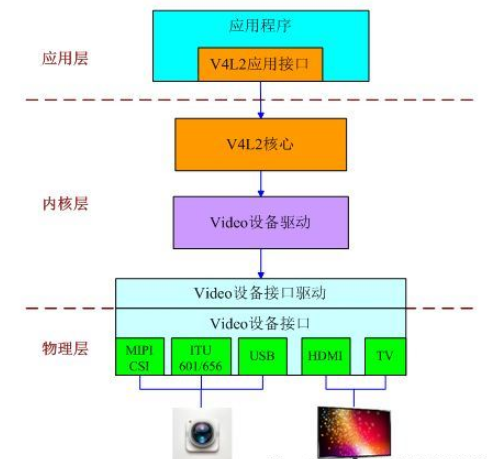
* 摄像头指标

1. 图像解析度/分辨率
2. 图像格式(imageFormat/Colorspace)
3. 自动白平衡调整(AWB)
4. 图像压缩方式
5. 彩色深度
6. 图像噪音。图像中的杂点干扰。表现为图像中有固定的彩色杂点。
7. 视角
8. 输出/输入接口

USB摄像头一般都是基于V4L2架构的，需要编写V4L2架构的程序来操作摄像头。V4L2（Video4Linux的缩写）是Linux下关于视频采集相关设备的驱动框架，为驱动和应用程序提供了一套统一的接口规范。

V4L2让驱动开发者依靠其驱动模型轻松的完成video设备的驱动，让视频应用程序可以轻松的使用其接口完成应用程序开发。

V4L2的架构

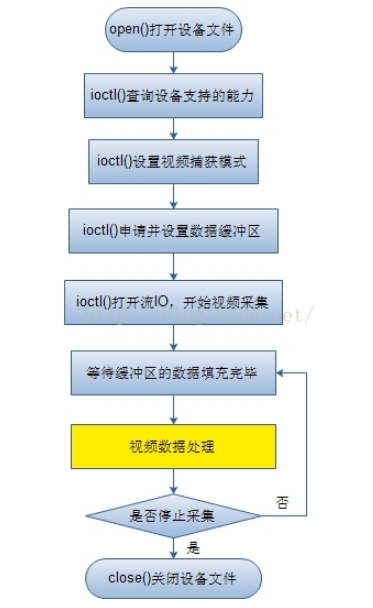


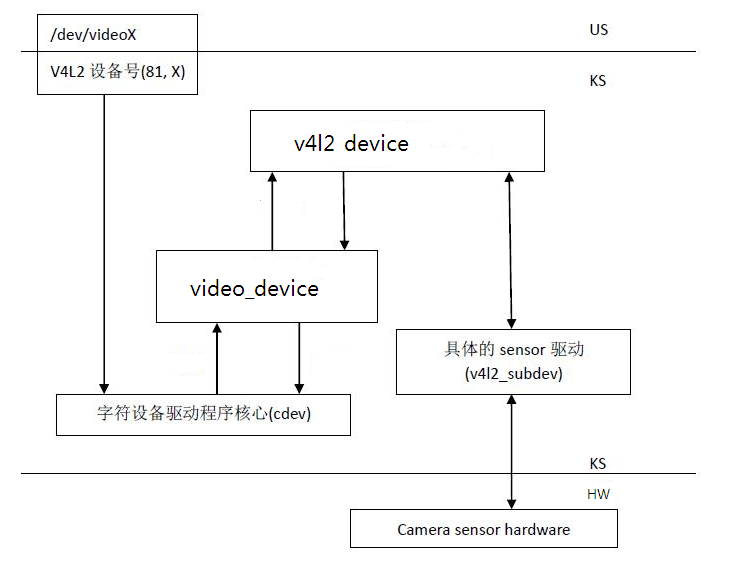
Linux的V4L2驱动开发的示例源码：“/drivers/media/video/vivi.c”

V4L2核心，为驱动开发者提供统一的视频设备驱动开发模型，对上为应用开发者提供操作视频设备的统一接口。“/drivers/media/video/v4l2-dev.c”。

应用程序通过V4L2提供read()、write()、ioctl()编程接口，来控制操作视频设备，如：设置图像分辨率、视频数据格式、开始/结束视频捕获等等。

* 应用程序编写





**字符设备驱动**程序核心：V4L2本身就是一个字符设备，提供接口给用户空间。

**V4L2驱动核心**：主要是构建一个内核中标准视频设备驱动的框架。

**平台V4L2设备驱动**：在V4L2框架下，根据平台自身的特性实现与平台相关的V4L2驱动部分。

**sensor驱动**：主要上电、提供工作时钟、视频图像裁剪、流IO开启等，实现各种设备控制方法供上层调用并注册v4l2\_subdev。



编译：

gcc webcam-stable.c -lv4l2 -o webcam-stable

程序运行

3个参数，分别是输出图片文件的路径、输出图片的宽度、输出图片的高度。

./webcam-stable /home/codelast/1.ppm 320 240

在输出图片之后，再将其转为jpg格式。

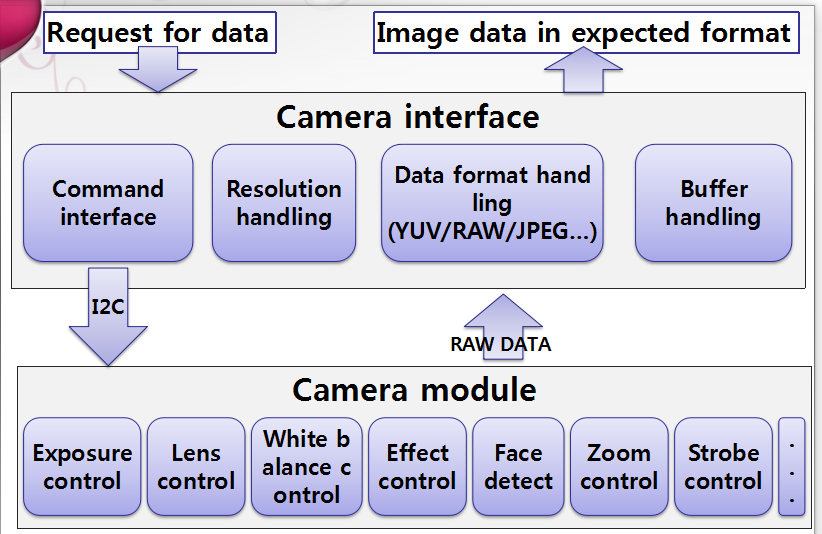
* USB摄像头设备驱动

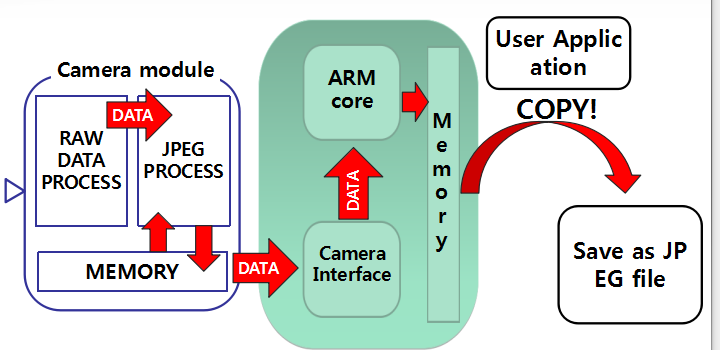
Platform是CPU上的Camera Interface驱动（由CPU厂商提供）

Sensor是具体的Camera模组驱动（由模组厂商提供）

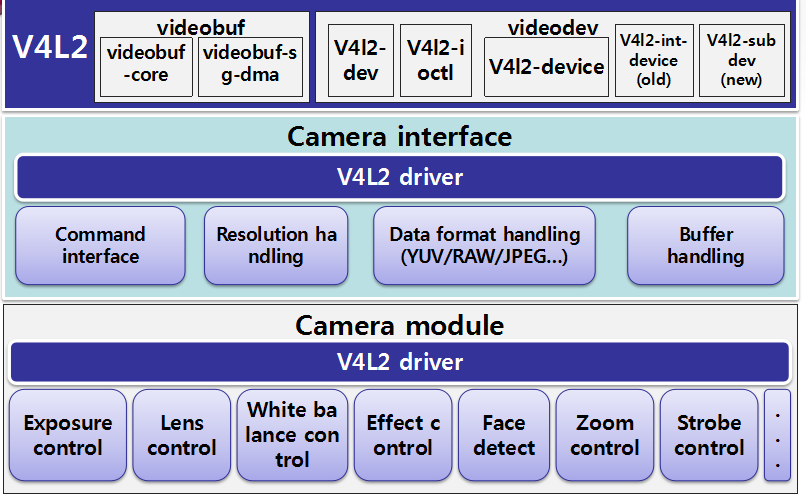
Machine驱动是具体自定义PCB的主板相关，将Platform和Sensor驱动连接起来。

SoC camera subsystem





Camera subsystem in Linux kernel



### python编程

fswebcam命令

$ sudo apt-get install fswebcam

$ fswebcam /dev/video0 ~/image.jpg

脚本1

import os

import sys

import re

import commands

a=commands.getoutput("fswebcam --no-banner -r 640x480 image3.jpg")

print a

脚本2

from picamera import PiCamera

import time

"""set camera"""

camera = PiCamera()

camera.resolution = (1920,1080)

camera.framerate = 60

# 打开预览

camera.start\_preview()

camera.capture('/home/pi/testme.jpg')

camera.stop\_preview()

## 数据库

周7

### 安装和测试mysql

**安装mysql**

sudo apt-get install mysql-server

安装过程中需要输入root管理员的密码，该密码之后用于访问数据库系统。

sudo netstat -tap | grep mysql

有mysql 的socket处于 listen 状态则表示安装成功。

**重启数据库**

sudo service mysql restart

**Mysql配置**

MySQL 默认只监听本地ip访问，如果需要远程IP访问还需要修改系统配置文件。

修改 /etc/mysql/my.cnf

bind-address = 127.0.0.1 把 127.0.0.1 改成 0.0.0.0 就可以远程访问了。

### 如果my.cnf文件没有bind-address

vi /etc/mysql/mariadb.conf.d/50-server.cnf

注释掉 bind-address = 127.0.0.1

修改root密码

sudo mysqladmin -u root -p password

Enter password:

New password: 填你的密码

Confirm new password:

**测试MySQL**

Sudo apt-get isntall mysql-client

sudo apt-get install libmysqlclient-dev

输入在安装过程中设置的密码。

$ sudo mysql -u root -p

或者以空密码登陆

$ sudo mysql -u root



### windows客户端

（windows）使用实例

测试数据库。 访问域名为：db4free.net 端口为：3306。

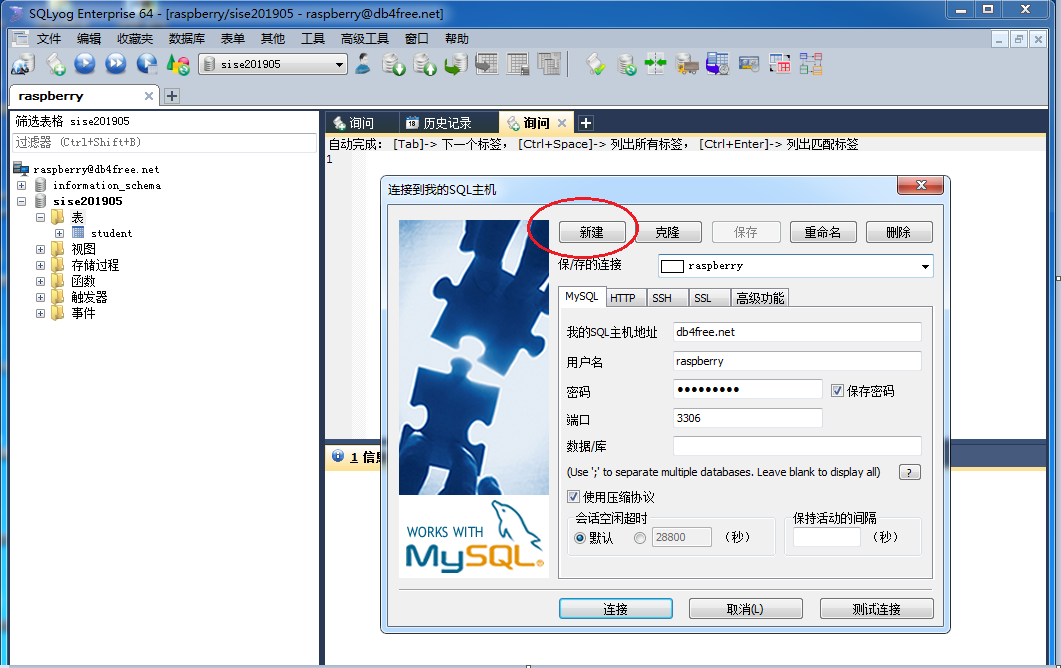
数据库: sise201905

用户名: raspberry

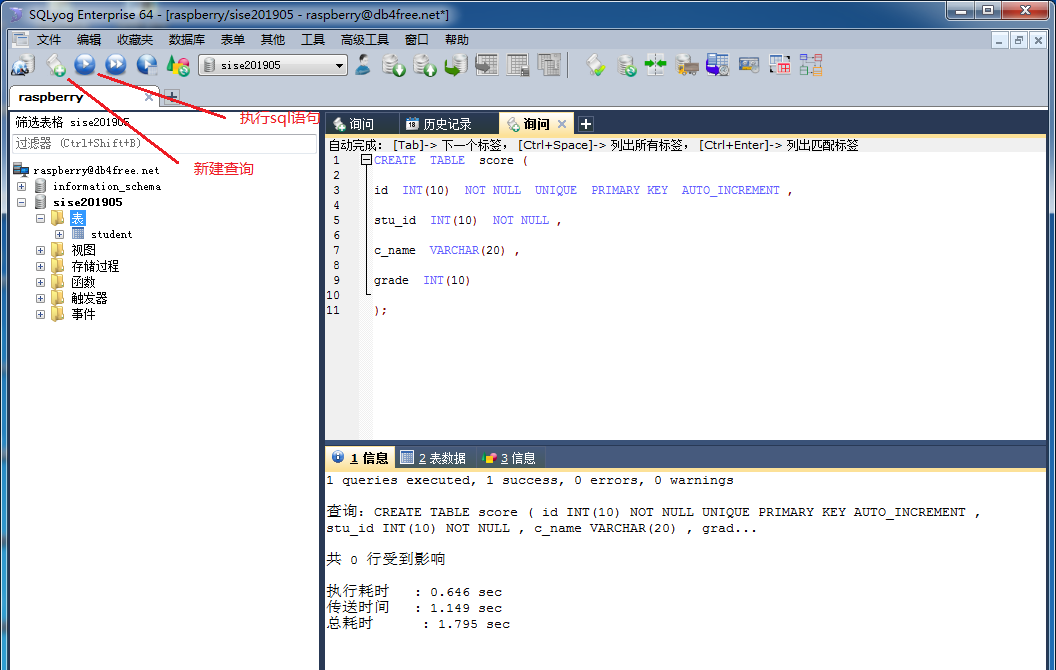
* SQLyog 12.0.8.0 中文

SQLyog是一个MySQL数据库全能管理工具。其社区版为自由及开源软件，遵循GPL许可协议。开发者在使用MySQL时所需的多数功能都可以通过简单的点击鼠标完成，通过标签界面可以查看查询结果集、查询分析器(query profiler)、服务器消息、表数据、表信息和查询历史等。

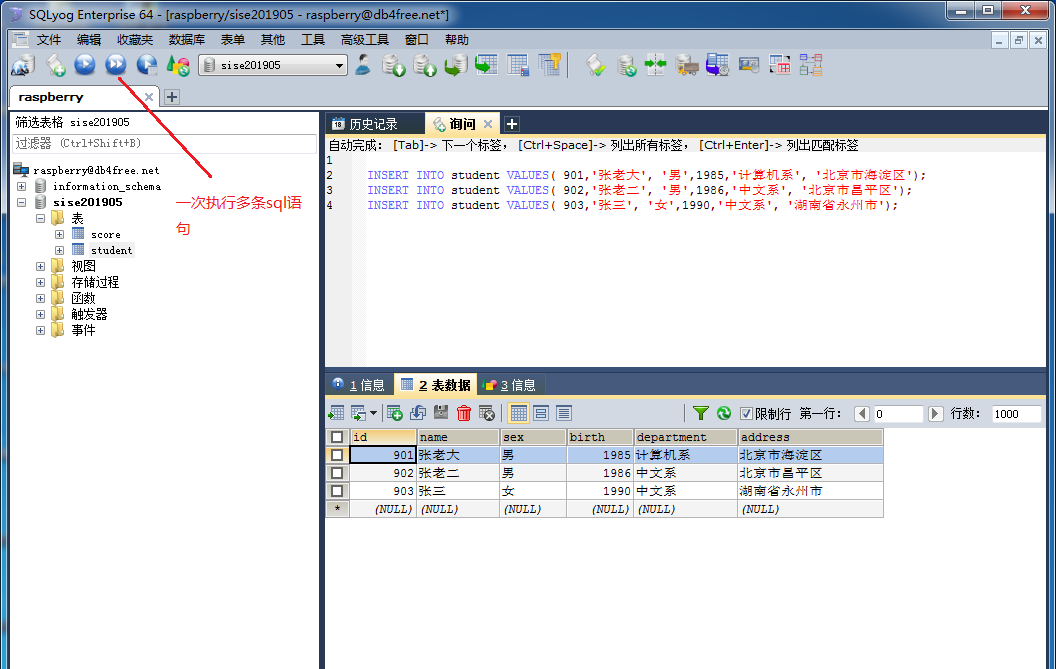
另外，开发者可以轻松创建视图和存储过程。



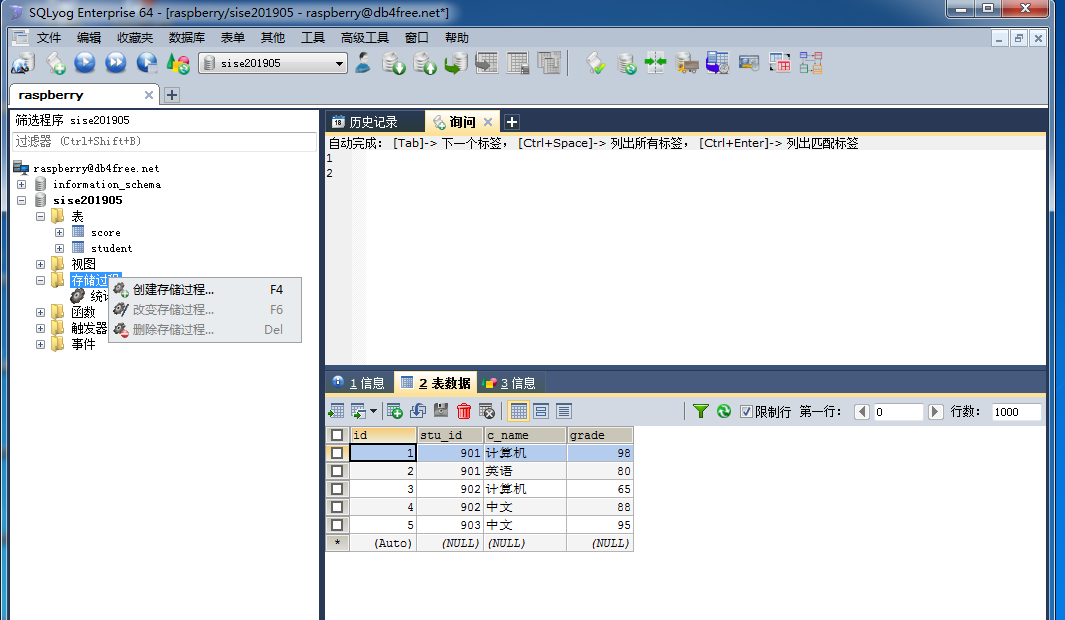
写sql语句



Student表插入多条数据。



创建存储过程。



### 数据库编程

C编写mysql客户端程序

初始化 =>

建立连接 =>

执行SQL语句 =>

获取结果集（针对不同的SQL操作，获取不同数据） =>

遍历结果集，以行的形式(也可以是随机访问) =>

关闭连接 =>

清理资源

*mysqltest0.cpp*

#include "mysql.h"

#include <cstdlib>

#include <cstdio>

int main()

{

MYSQL mysql;

MYSQL\_RES \*res;

MYSQL\_ROW row;

char sql[100];

mysql\_library\_init(0, NULL, NULL);

mysql\_init(&mysql);

if (!mysql\_real\_connect(&mysql,"localhost","root","root","test",0,NULL,0)){

fprintf(stderr, "无法连接到数据库，错误原因：%s\n", mysql\_error(&mysql));

}

else {

fprintf(stdout, "连接数据库成功\n");

sprintf(sql, "insert into students(name) values('eric')");

mysql\_query(&mysql, sql);

sprintf(sql, "select name from students");

int t = mysql\_query(&mysql, sql);

if (0 != t) {

fprintf(stderr, "查询数据库失败%s\n", mysql\_error(&mysql));

}

else {

res = mysql\_store\_result(&mysql);

while (row = mysql\_fetch\_row(res)) {

for (int r = 0;r < mysql\_num\_fields(res); r++) {

printf("%s\n", row[r]);

}

}

}

}

return 0;

}

编译链接执行

g++ -c -I /usr/include/mysql mysqltest0.cpp

g++ -o mysqltest0 mysqltest0.o -L /usr/lib/mysql -lmysqlclient

./mysqltest0

### python编程

import MySQLdb

sql='show databases';

conn=MySQLdb.connect(host='localhost',user='test',passwd='test',db='test',port=3306)

cur=conn.cursor()

cur.execute(sql)

print( cur.fetchone() )

# python mysql\_test1.py

安装mysql的Python3包

pip3 install pymysql

获取CPU温度并存入数据库

**import** time

**import** pymysql

*#获取CPU温度*

**def** **get\_temp**():

**with** open("/sys/class/thermal/thermal\_zone0/temp", "r") **as** f:

temp **=** float(f**.**read()) **/** 1000

**return** temp

*#检查是否写数据库成功,如果不成功,写入本地文件***def** **intoFile**(dt, temp\_now):

**with** open("./temp.txt", 'a+') **as** f:

f**.**write(dt **+** " " **+** str(temp\_now) **+** '\n')

*#连接数据库并写入数据库*

**def** **conn\_mysql**(dt, temp\_now):

**try**:

*#打开数据库*

db **=** pymysql**.**connect(host**=**"host", port**=**3306, user**=**"name",passwd **=**"password",db **=** "temp", charset**=**'utf8')

*# 使用 cursor() 方法创建一个游标对象 cursor*

cursor **=** db**.**cursor()

sql **=** "INSERT INTO temp(time,temp) VALUES ('%s','%f')" **%** (dt, temp\_now)

cursor**.**execute(sql)

cursor**.**close()

*#更新数据库*

db**.**commit()

db**.**close()

**except**:

**return** "Errors"

**def** **insert2mysql**(dt, temp\_now):

*#存入Mysql并返是否成功,没有成功就存入本地*

sta **=** conn\_mysql(dt, temp\_now)

**if** sta **is** **None**:

**pass**

**else**:

intoFile(dt, temp\_now)

**def** **main**():

**while** **True**:

*#获取当前时间,并格式化成可以存放在MYSQL中*

dt **=** time**.**strftime("%Y-%m-%d %H:%M:%S", time**.**localtime())

*#获取当前CPU温度*

temp\_now **=** get\_temp()

insert2mysql(dt, temp\_now)

time**.**sleep(10)

**if** \_\_name\_\_ **==** '\_\_main\_\_':

main()

## 项目实践（机器视觉）

用OpenCV控制摄像头采集视频，以及进行运动检测（Motion Detection）的实现。

### OpenCV

OpenCV是一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。

OpenCV参考手册

wiki.opencv.org.cn

**特性：**

常用的标准图

图像文件读入和显示

图像创建、保存和复制

Canny边缘检测

Canny边缘检测2

轮廓(contour)检测

轮廓(contour)检测2

图像旋转与缩放

读视频文件和运动物体检测

Hough线段检测

鼠标绘图

snake轮廓例子

BMP与IplImage相互转换

yml文件的读取与写入

离散傅立叶变换(DFT)

人脸检测

QR分解

绘制贝赛尔Bezier曲线

使用DirectShow采集图像

设定跟踪目标图片的改进camshift例子

摄像头标定

施密特正交化

分形 fractal 演示

高斯背景建模

图像缩放

高级图像处理初步

图像颜色分布直方图

灰度模板匹配

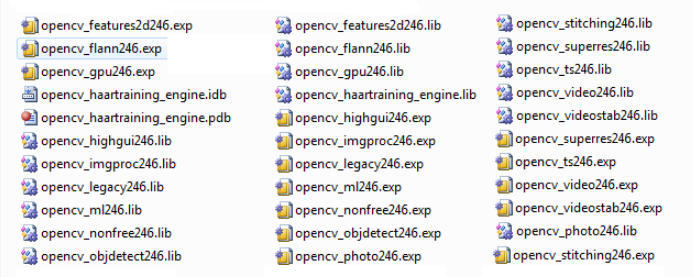
特征点匹配

支持向量机

OpenCV 拥有包括 300 多个C函数的跨平台的中、高层 API。它不依赖于其它的外部库——尽管也可以使用某些外部库。

OpenCV 对非商业应用和商业应用都是免费（FREE）的。

**OpenCV2.X库模块**



opencv\_modules.hpp文件里面存放的是OpenCV2中与新模块构造相关的说明代码

#define HAVE\_OPENCV\_CALIB3D 相机校准和三维重建相关内容。

#define HAVE\_OPENCV\_CONTRIB 新型人脸识别、立体匹配、人工视网膜模型等。

#define HAVE\_OPENCV\_CORE opencv 基本数据结构、动态数据结构、绘图函数、数组

#define HAVE\_OPENCV\_FEATURES2D 2D功能框架

#define HAVE\_OPENCV\_FLANN 快速近似最近邻搜索、聚类

#define HAVE\_OPENCV\_GPU 用GPU加速的计算机视觉模块

#define HAVE\_OPENCV\_HIGHGUI 高层GUI图形用户界面

#define HAVE\_OPENCV\_IMGPROC 图像处理模块

#define HAVE\_OPENCV\_LEGACY 一些已经废弃的代码库，保留是为了向下兼容

#define HAVE\_OPENCV\_ML 机器学习模块

#define HAVE\_OPENCV\_NONFREE 具有专利的算法模块,特征检测和GPU相关的

#define HAVE\_OPENCV\_OBJDETECT 目标检测模块

#define HAVE\_OPENCV\_OCL 用OpenCL加速的计算机视觉组件模块

#define HAVE\_OPENCV\_PHOTO 图像修复和图像去噪两部分

#define HAVE\_OPENCV\_STITCHING images stitching 图片拼接模块

#define HAVE\_OPENCV\_SUPERRES 超分辨率技术的相关功能模块

#define HAVE\_OPENCV\_TS Opencv 测试相关代码

#define HAVE\_OPENCV\_VIDEO 视频分析组件

#define HAVE\_OPENCV\_VIDEOSTAB 视频稳定相关的组件

### Opencv编程

* 安装相关软件包

运行OpenCV程序，仅需安装libcv1，libcvaux1，libhighgui1：

apt-get install libcv1 libcvaux1 libhighgui1

使用OpenCV来编写程序，那还需要安装libcv-dev，libcvaux-dev，libhighgui-dev包。

apt-get install libcv-dev libcvaux-dev libhighgui-dev

* **运动检测**

说明：程序打开本电脑上的摄像头作为视频输入设备，或者将命令行的输入参数作为文件名来打开的视频文件。

不断的循环一帧一帧地处理，涉及到的图像处理有 背景擦除，平滑滤波，二值化等。



编译和运行

$ gcc test.cpp -o test `pkg-config --cflags --libs opencv`

$ ./test //运行，显示图片

* **用cmake测试**



CMakeLists.txt中的源代码：

cmake\_minimum\_required(VERSION 2.8)

project( DisplayImage )

find\_package( OpenCV REQUIRED )

add\_executable( DisplayImage DisplayImage.cpp )

target\_link\_libraries( DisplayImage ${OpenCV\_LIBS} )

编译命令：

cmake .

make

运行命令：

./DisplayImage

### python编程

安装Python3上的OpenCV

sudo pip3 install opencv-python imutils numpy scipy pillow

sudo apt-get install libatlas-base-dev libjasper-dev

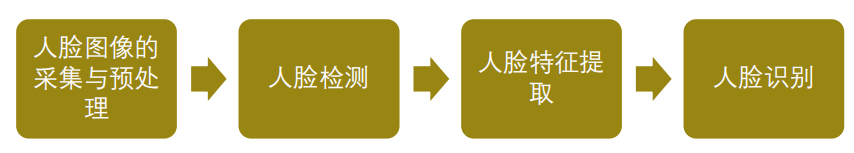
sudo apt-get install libgstreamer1.0-0

sudo apt-get install libgstreamer-plugins-base1.0-0

sudo apt-get install libqtgui4 libqt4-test

sudo apt install python-opencv

## 项目实践（人脸定位）



人脸检测是指应用一定的策略对给出的图片或者视频来进行检索，判断是否存在着人脸。

如果存在则定位出每张人脸的位置、大小与形态的过程。

人脸检测方法

1. 基于肤色模型的检测。
2. 基于边缘特征的检测。
3. 基于统计理论方法。

检测指标

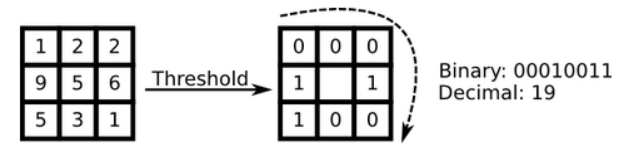
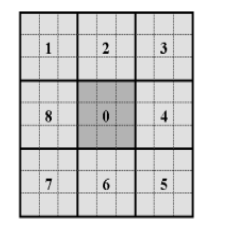
1. 检测率：识别正确的人脸/图中所有的人脸。
2. 误检率：识别错误的人脸/识别出来的人脸。
3. 漏检率：未识别出来的人脸/图中所有的人脸。
4. 速度：从采集图像完成到人脸检测完成的时间。

经典算法

* 特征脸法（Eigenface）
* 局部二值模式（Local Binary Patterns，LBP）

OpenCV 的人脸检测系统设计

我们需要将分类器的XML文件haarcascade\_frontalface\_default.xml从opencv文件夹/sources/data/haarcascades/下复制到我们的项目目录下。

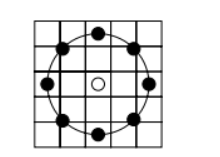
 

**LBP特征**

最初的LBP是定义在像素3x3邻域内的，以邻域中心像素为阈值，将相邻的8个像素的灰度值与其进行比较，若周围像素值大于中心像素值，则该像素点的位置被标记为1，否则为0。

这样，3x3邻域内的8个点经比较可产生8位二进制数（通常转换为十进制数即LBP码，共256种），用这个值来反映该区域的纹理信息。

改进后的 LBP 算子允许在半径为 R 的圆形邻域内有任意多个像素点。从而得到了诸如半径为R的圆形区域内含有P个采样点的LBP算子。比如下图定了一个5x5的邻域：





## 项目实践（语音处理）

## 作业

1. Shell脚本。

ftp 下载文件。

遍历ftp服务器文件目录，并且保存日志。

1. 源码分析 linux 命令 ls

linux中ls源码简单分析

coreutils源代码包看看

查看一个软件工具属于哪一个包

# dpkg -S 'ls'

## 附录

https://github.com/raspberrypi

树莓派的GPIO编程 https://www.cnblogs.com/vamei/p/6751992.html

USB设备驱动开发之扩展（利用USB虚拟总线驱动模拟USB摄像头

用STM32F407实现的USB摄像头UVC

Firefly-RK3399

Camera硬件及基于V4L2驱动源码分析

摄像头方案总结

USB摄像头驱动实现源码分析

LINUX 下USB摄像头设置

Linux系统下USB摄像头驱动开发

linux 下 USB 摄像头实现分析

摄像头的工作原理说明加电路图

在Raspberry Pi(树莓派)上调用V4L2来操纵摄像头拍照

Navicat如何写sql语句

Opencv实现运动检测

http://www.cnblogs.com/tanhangbo/p/4293800.html

使用树莓派控制摄像头采集视频及运动检测

http://v.eepw.com.cn/video/play/id/14050

树莓派家庭监控的运动检测和追踪系统

An OpenCV Framework of Automated Sign language to Text Translator

OpenCV: High-level design overview

树莓派升级（安装）Python3.6

一文总结 Python 3 的 14 张思维导图

远程调试树莓派（PyCharm实现）

《Python语言参考》对Python语言给出了更正式的定义。

《扩展和嵌入Python解释器》以及《Python/C API参考手册》。

树莓派---GPIO应用详解