## 树莓派安装与操作步骤

1. 安装系统
2. 将储存卡插入读卡器，读卡器插入电脑USB接口
3. 将…\大创\文档、成果、工具汇总\树莓派\开发工具 目录下SDFormatter-ha.zip文件解压
4. 双击解压出的文件SDFormatter.exe
5. 在Drive选项中选择电脑识别的储存卡，点击“格式化”按钮
6. 格式化完毕后点击“完成”
7. 解压同一目录下的Win32DiskImager-0.9.5-binary.zip文件
8. 双击打开解压文件中的Win32DiskImager.exe文件
9. 在Device选项中选择储存卡
10. 在Image File选项中选择同一目录下的2016-02-26-raspbian-jessie.img文件，点击“Write”按钮开始烧写系统
11. 烧写完成后点击“Exit”按钮，此时储存卡名称变为“boot”，拔出储存卡
12. 运行系统
13. 将储存卡拔出读卡器，插入树莓派开发板底侧卡槽
14. 将摄像头接口插入开发板中间Camera字样接口，向上拔动打开接口，插入摄像头，按下接口，注意方向
15. 将无线网卡，鼠标，键盘，接入开发板USB接口
16. 用HDMI高清输出线连接显示器与开发板
17. 插入microUSB电源，待系统启动
18. 基本环境搭建
19. 点击屏幕上方 黑色屏幕状 按钮“LXTerminal”，出现“终端”界面，输入一下指令后回车：

sudo raspi-config

此时会出现菜单

选择第一项“Expand Filesystem”回车，稍等1，2秒后会出现提示，再按一次回车回到菜单，按键盘ESC键退回终端 输入

sudo reboot

等待系统重启

1. （可选）显示输出切换
2. 将 …\大创\文档、成果、工具汇总\树莓派\开发工具 目录下LCD\_show.tar.gz文件拷贝进U盘
3. 将U盘插入开发板，系统自动识别弹窗，点击OK按钮，进入U盘将该文件复制到桌面
4. 将3.5寸LED显示屏对其开发板插于开发板上
5. 进入终端界面，依次输入一下指令(每输入一条回车，注意空格)：

cd Desktop

tar　–xzvf　LCD\_show.tar.gz

等待解压完成

cd LCD\_show

sudo .\ LCD35\_v2

完成后会系统重启，LCD屏不再显示内容，显示屏出现系统界面

若想切换回高清显示屏输出只需进入终端输入：

cd Deskop/LCD\_show

sudo .\LCD\_hdmi

1. 点击屏幕右上角 电脑状图标 选择WiFi连接
2. 更新软件源

进入终端，输入

sudo nano /etc/apt/sources.list

进入新界面 删除其中的三行内容

添加以下两行内容

deb http://mirror.sysu.edu.cn/raspbian/raspbian/ jessie main contrib non-free

deb-src http://mirror.sysu.edu.cn/raspbian/raspbian/ jessie main contrib non-free

按键盘ctrl+x键，再按y键，再按回车，返回终端界面输入

sudo apt-get update

等待刷新

1. 终端输入sudo raspi-config

选择第五项：Enable Camera回车，选择enable回车，选择finish回车，选择yes回车 等待系统重启

1. 安装opencv环境

<http://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/>

期间所有询问输入y 回车，

1. 打开终端输入，**逐条输入下列指令，待指令操作结束且成功后再输入下一条**

cd /etc/apt/source.list.d

sudo rm collabora.list

cd

sudo apt-get update

sudo apt-get upgrade

出现提示后输入y 回车，开始更新，更新结束后输入

sudo rpi-update

更新成功后，输入

sudo reboot

重启

sudo apt-get install build-essential git cmake pkg-config

sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev

sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

sudo apt-get install libxvidcore-dev libx264-dev

sudo apt-get install libgtk2.0-dev

sudo apt-get install libatlas-base-dev gfortran

sudo apt-get install python2.7-dev python3-dev

cd ~

wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.0.0.zip

unzip opencv.zip

wget -O opencv\_contrib.zip https://github.com/Itseez/opencv\_contrib/archive/3.0.0.zip

unzip opencv\_contrib.zip

wget <https://bootstrap.pypa.io/get-pip.py>

sudo python get-pip.py

确认该指令成功后再继续即出现successful字样，否则重新输入该指令

sudo pip install virtualenv virtualenvwrapper

sudo rm -rf ~/.cache/pip

sudo nano ~/.profile

会出现新界面，在该界面下添加三行：

# virtualenv and virtualenvwrapper

export WORKON\_HOME=$HOME/.virtualenvs

source /usr/local/bin/virtualenvwrapper.sh

添加完毕后ctrl+x，键盘y键，回车，回到终端指令界面

继续输入指令

source ~/.profile

mkvirtualenv cv

注：

此时每条指令钱都会出现（cv）字样，若中途退出可输入指令

source ~/.profile

workon cv

继续输入

pip install numpy

可能需要执行多次，知道成功为止，出现successful字样

workon cv

cd ~/opencv-3.0.0/

mkdir build

cd build

以下指令为一条：

cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

-D CMAKE\_INSTALL\_PREFIX=/usr/local \

-D INSTALL\_C\_EXAMPLES=ON \

-D INSTALL\_PYTHON\_EXAMPLES=ON \

-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_contrib-3.0.0/modules \

-D BUILD\_EXAMPLES=ON ..

结束

make -j4

sudo make install

sudo ldconfig

cd ~/.virtualenvs/cv/lib/python2.7/site-packages/

ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so

测试，输入一下指令若出现”3.0.0”即成功

workon cv

python

import cv2

cv2.\_\_version\_\_

成功后关闭该终端，重新打开输入

sudo apt-get install python-opencv

1. 运行程序
2. 将…\大创\文档、成果、工具汇总\树莓派\目录下，FaceRecognition文件夹拷入U盘
3. 将U盘插入开发板，弹出窗口，点击ok，或在左上角菜单中点击FileManager，进入路径/media/pi，即可找到U盘
4. 将FaceRecognition拷贝到桌面
5. 打开终端输入

cd Desktop/FaceRecognition

sudo python FaceRecognition.py

此时出现提示，若需要显示大窗口即输入

max

回车

若需要显示小窗口即输入

min

回车

随后即可出现视频显示界面，

相应指令：

q 退出程序

s 识别人脸，在此期间按c可捕捉识别到的人脸，终端出现提示，在终端中输入回车后，再输入人名，可记录人脸，可多次记录

r 验证人脸，将摄像头捕捉到的人脸与已存人脸对比，若符合会在人脸框左上方出现保存的名字，右上方出现匹配值，越低越准确

n 显示所有名称，控制台显示已存的所有名称

d删除名称，在终端中输入回车后，再输入人名，可删除已存的相应人脸

= 显示所有触发人脸验证的时间

t 以当前已存的人脸库，重新训练分类器

p 停止识别或验证

若无人体红外可将FaceRecognition/ FaceRecognition.py中

If ALIVE改为 if true