

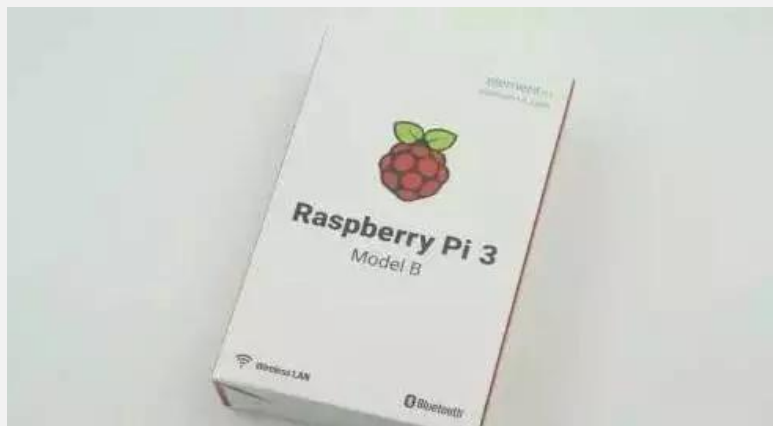


树莓派中如何使用ROS

作者： corvin

日期： 2017-8-23

一、安装树莓派jessie镜像流程



准备材料

1. 树莓派3 B型开发板，板载wifi和蓝牙；
2. 微雪7寸触摸显示屏；
3. 川宇microSD卡读卡器；
4. 至少16G的microSD卡；
5. 两根microUSB线，一条给树莓派供电，一条为触摸屏的触摸驱动线；
6. 一条HDMI线，连接树莓派与显示屏；

一、安装树莓派jessie镜像流程

1. 树莓派官网下载最新的jessie镜像，下载地址如下：

<https://www.raspberrypi.org/downloads/raspbian/>

2.准备好存储空间大于等于16G的SD卡，准备将镜像安装到SD卡中，操作流程：

<https://www.raspberrypi.org/documentation/installation/installing-images/linux.md>

3.安装完成后，将SD卡插入到树莓派主板中，开机启动；

4.此时**微雪的7寸触摸显示屏**无法正常显示，需要修改配置文件，才能正常加载微雪触摸屏的驱动。

用于树莓派的 Raspbian / Ubuntu mate 系统。当LCD工作于树莓派的 Raspbian / Ubuntu mate 系统时，必须手动设置分辨率，否则会显示黑白相间的条纹。编辑TF卡根目录下的config.txt文件。在config.txt文件末尾添加：

```
max_usb_current=1
```

```
hdmi_group=2
```

```
hdmi_mode=87
```

```
hdmi_cvt 1024 600 60 6 0 0 0
```

一、安装树莓派jessie镜像流程

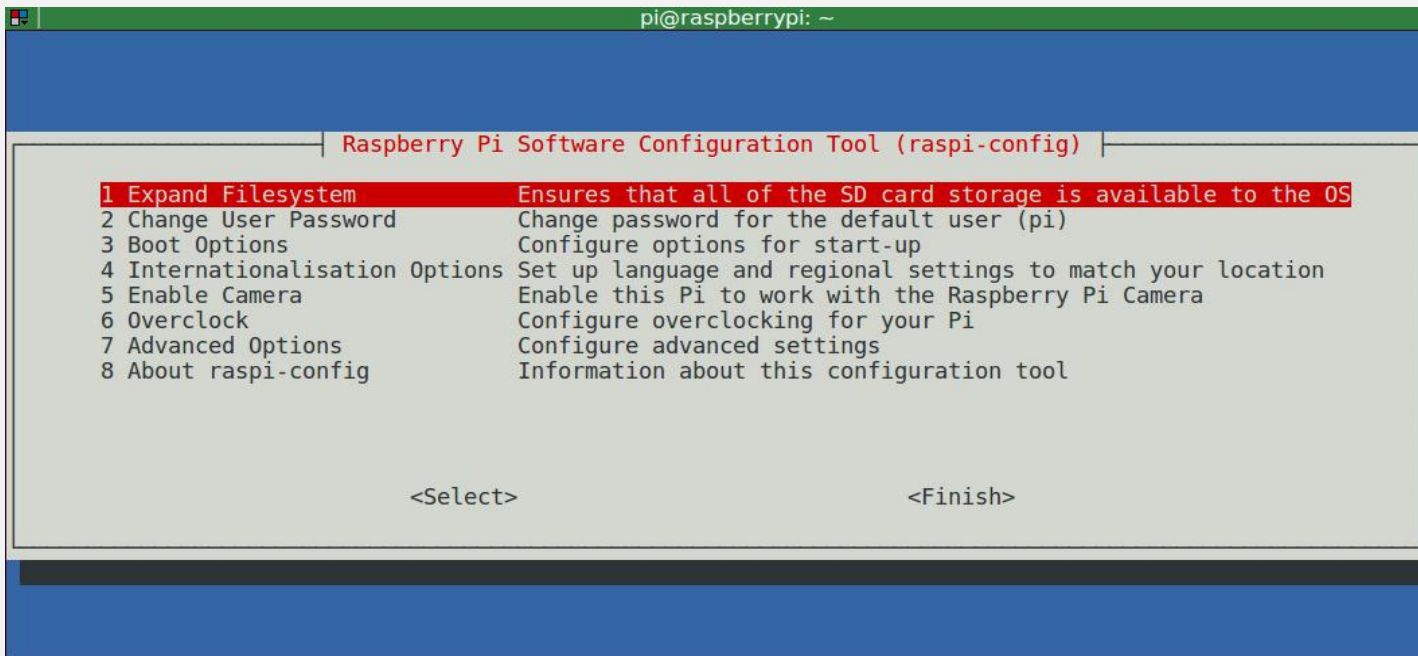
5.更新源为国内清华的源，编辑 /etc/apt/sources.list 文件，注释掉旧的路径，更新为如下：

```
deb http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ jessie main non-free contrib
```

```
deb-src http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ jessie main non-free contrib
```

6.树莓派第一次使用的时候需要进行一个简单的配置，在命令行模式下运行以下命令：

`sudo raspi-config`

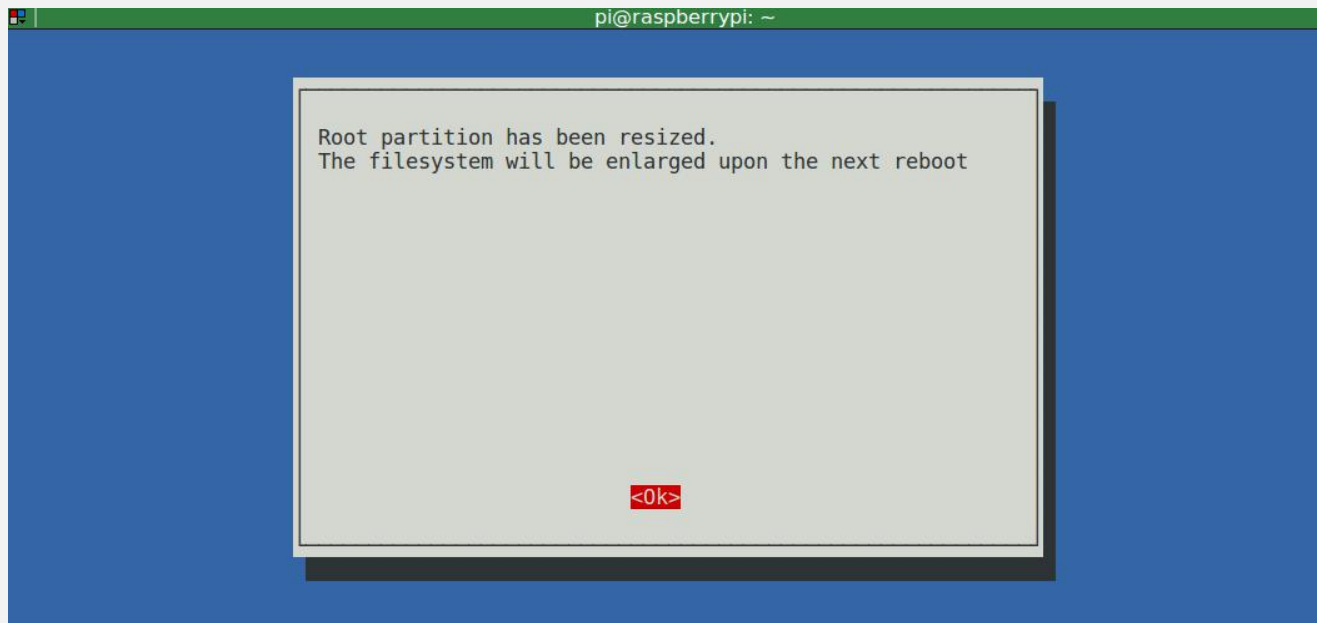


```
pi@raspberrypi: ~  
  
Raspberry Pi Software Configuration Tool (raspi-config)  
  
1 Expand Filesystem      Ensures that all of the SD card storage is available to the OS  
2 Change User Password   Change password for the default user (pi)  
3 Boot Options           Configure options for start-up  
4 Internationalisation Options Set up language and regional settings to match your location  
5 Enable Camera          Enable this Pi to work with the Raspberry Pi Camera  
6 Overclock              Configure overclocking for your Pi  
7 Advanced Options       Configure advanced settings  
8 About raspi-config     Information about this configuration tool  
  
<Select>                <Finish>
```

一、安装树莓派jessie镜像流程

7. 介绍各配置选项意义,首先介绍其中两个较为重要的;

(1) Expand Filesystem - 扩展文件系统,最新jessie镜像会自动扩展文件系统分区。可以运行, `df -h`命令进行查看, 如果/dev/root分区不是SD卡大小的话, 那么需要运行该功能进行扩展分区。



一、安装树莓派jessie镜像流程

7. 介绍各配置选项意义,介绍其中两个较为重要的;

(2) Advanced Options

SSH 是否运行SSH登录, 建议开户此选项, 以后操作PI方便, 有网络就行, 不用连接屏幕了.

```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan      You may need to configure overscan if black bars are present on display
A2 Hostname      Set the visible name for this Pi on a network
A3 Memory Split  Change the amount of memory made available to the GPU
A4 SSH           Enable/Disable remote command line access to your Pi using SSH
A5 VNC           Enable/Disable graphical remote access to your Pi using RealVNC
A6 SPI           Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiFace)
A7 I2C           Enable/Disable automatic loading of I2C kernel module
A8 Serial        Enable/Disable shell and kernel messages on the serial connection
A9 Audio         Force audio out through HDMI or 3.5mm jack
AA 1-Wire        Enable/Disable one-wire interface

                <Select>                                <Back>
```


一、安装树莓派jessie镜像流程

8. 介绍其他各配置选项意义；

- Change User Password -改变默认pi用户的密码，按回车后输入pi用户的新密码。

默认用户:pi, 默认密码是:raspberrypi

- Boot Options -启动时的选项

```
Raspberry Pi Software Configuration Tool (raspi-config)

B1 Desktop / CLI      Choose whether to boot into a desktop environment or the command line
B2 Wait for Network at Boot Choose whether to wait for network connection during boot
B3 Splash Screen      Choose graphical splash screen or text boot

<Select>              <Back>
```

一、安装树莓派jessie镜像流程

- Internationalisation Options 国际化选项，可以更改默认语言

I1 Change Locale - 语言和区域设置

I2 Change Timezone - 设置时区，如果不进行设置，PI的时间就显示不正常。选择Asia（亚洲）再选择Chongqing（重庆）即可。

I3 Change Keyboard Layout - 改变键盘布局

I4 Change Wi-Fi Country - 修改wifi热点所在国家

一、安装树莓派jessie镜像流程

- Enable Camera

启动PI的摄像头模块，如果想启用，选择Enable，禁用选择Disable就行了

- Advanced Options 高级设置

A1 Overscan 是否让屏幕内容全屏显示

A2 Hostname 在网上邻居或者路由器能看到的主机名称

A3 Memory Split 内存分配，选择给GPU多少内存

A4 SSH 是否运行SSH登录，建议开户此选项，以后操作PI方便，有网络就行，不用开屏幕了。

A5 VNC 是否可以远程桌面访问本地树莓派

... ..

A0 Update 把raspi-config这个工具自动升级到最新版本

二、在树莓派jessie中安装ROS indigo版本

1. 安装的教程网址：<http://wiki.ros.org/ROSberryPi>
2. 设置ROS的仓库，分为Wheezy镜像和Jessie镜像两个，这里只以Jessie镜像演示安装教程：
 - `sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu jessie main" > /etc/apt/sources.list.d/ros-latest.list'`
 - `wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -`

二、在树莓派jessie中安装ROS indigo版本

3. 确保Debian包都更新到最新版本(更新为国内源)

- `sudo apt-get update`
- `sudo apt-get upgrade`

3. 安装相关的辅助功能包（Jessie版本）

- `sudo apt-get install python-pip python-setuptools python-yaml python-distribute python-docutils python-dateutil python-six`
- `sudo pip install rosdep rosinstall_generator wstool rosinstall`

二、在树莓派jessie中安装ROS indigo版本

5. 初始化rosdep

- `sudo rosdep init`
- `rosdep update`

5. 创建catkin工作空间，准备下载和编译ROS indigo

- `mkdir ~/ros_catkin_ws`
- `cd ~/ros_catkin_ws`

接下来将要去获取核心包，然后编译。我们使用wstool这个工具完成这些操作，ROS 提供了两个代码包：

- ROS-Comm: 这是官方推荐的，但只包含基本的ROS通讯功能，没有GUI工具。
- Desktop: 完整的ROS 功能，适合所有机器人应用，但编译较为麻烦有各种问题。

二、在树莓派jessie中安装ROS indigo版本

7. 安装ROS-Comm包

- `rosinstall_generator ros_comm --rostdistro indigo --deps --wet-only --exclude roslisp --tar > indigo-ros_comm-wet.rosinstall`
- `wstool init src indigo-ros_comm-wet.rosinstall`

执行这两条命令可能会花费几分钟时间，执行完成后将会下载核心ROS功能包的源码到~/ros_catkin_ws/src这个文件夹中。

如果wstool init失败或者被中断了，通过运行以下命令来重新开始下载：

- `wstool update -t src`

二、在树莓派jessie中安装ROS indigo版本

8. 解决依赖关系

在可以编译catkin工作空间之前，需要确保已经安装了要求的依赖包，虽然我们可以使用rosdep工具来解决依赖包，但是仍然有一些包不可用，我们必须手动的去编译才行。 [Ros_Comm: libconsole-bridge-dev, liblz4-dev](#)

这些包可以通过下载源码在一个新目录中进行编译：

- `mkdir ~/ros_catkin_ws/external_src`
- `sudo apt-get install checkinstall cmake`
- `sudo sh -c 'echo "deb-src http://mirrordirector.raspbian.org/raspbian/ testing main contrib non-free rpi" >> /etc/apt/sources.list'`
- `sudo apt-get update`

二、在树莓派jessie中安装ROS indigo版本

9. 安装libconsole-bridge-dev功能包

- `cd ~/ros_catkin_ws/external_src`
- `sudo apt-get build-dep console-bridge`
- `apt-get source -b console-bridge`
- `sudo dpkg -i libconsole-bridge0.2*.deb libconsole-bridge-dev_*.deb`

10. 安装liblz4-dev功能包

- `cd ~/ros_catkin_ws/external_src`
- `apt-get source -b lz4`
- `sudo dpkg -i liblz4-*.deb`

二、在树莓派jessie中安装ROS indigo版本

11.使用rosdep解决剩余的依赖包

- `cd ~/ros_catkin_ws`
- `rosdep install --from-paths src --ignore-src --rosdistro indigo -y -r --os=debian:jessie`

12.编译catkin工作空间

若已经下载完成这些包，那么就解决了所有的依赖包，现在可以开始编译这些包了。

- `sudo ./src/catkin/bin/catkin_make_isolated --install -DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/indigo -j2`
- `echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc`

二、在树莓派jessie中安装ROS indigo版本

13. 测试ROS安装是否正常

通过在终端中启动roscore来测试，如果打印如下错误：

```
PARAMETERS
* /rostdistro: indigo
* /rosversion: 1.11.20

NODES

auto-starting new master
process[master]: started with pid [1452]
ROS_MASTER_URI=http://Robot:11311/

setting /run_id to 2d89bef4-d0e6-11e6-9dba-b827eba40603
process[rosout-1]: started with pid [1465]
started core service [/rosout]
terminate called after throwing an instance of 'std::runtime_error'
  what():  locale::facet::_S_create_c_locale name not valid
[rosout-1] process has died [pid 1465, exit code -6, cmd /opt/ros/indigo/lib/rosout/rosout __name:=rosout __log:=/home/pi/.ros/log/2d89bef4-d0e6-11e6-9dba-b827eba40603/rosout-1.log].
log file: /home/pi/.ros/log/2d89bef4-d0e6-11e6-9dba-b827eba40603/rosout-1*.log
[rosout-1] restarting process
process[rosout-1]: started with pid [1468]
terminate called after throwing an instance of 'std::runtime_error'
  what():  locale::facet::_S_create_c_locale name not valid
[rosout-1] process has died [pid 1468, exit code -6, cmd /opt/ros/indigo/lib/rosout/rosout __name:=rosout __log:=/home/pi/.ros/log/2d89bef4-d0e6-11e6-9dba-b827eba40603/rosout-1.log].
log file: /home/pi/.ros/log/2d89bef4-d0e6-11e6-9dba-b827eba40603/rosout-1*.log
[rosout-1] restarting process
process[rosout-1]: started with pid [1469]
terminate called after throwing an instance of 'std::runtime_error'
  what():  locale::facet::_S_create_c_locale name not valid
[rosout-1] process has died [pid 1469, exit code -6, cmd /opt/ros/indigo/lib/rosout/rosout __name:=rosout __log:=/home/pi/.ros/log/2d89bef4-d0e6-11e6-9dba-b827eba40603/rosout-1.log].
log file: /home/pi/.ros/log/2d89bef4-d0e6-11e6-9dba-b827eba40603/rosout-1*.log
```

出现如左图所示的错误是因为本地语言设置问题造成的，解决步骤如下：

1. 首先在 raspi-config 里的 Configuring locales 把下面两个语言打上勾：

[*] en_US.UTF-8 UTF-8

[*] zh_CN.UTF-8 UTF-8

2. Default Locale 选择 en_US.UTF-8。

3. 再次启动roscore即可正常启动，没有异常。

三、使用VNC远程桌面连接树莓派

1. VNC (Virtual Network Console)是虚拟网络控制台的缩写。它是一款优秀的远程控制工具软件，由著名的 AT&T 的欧洲研究实验室开发的。VNC是在基于UNIX 和 Linux 操作系统的免费开源软件，远程控制能力强大，高效实用，其性能可以和 Windows 和 MAC 中的任何远程控制软件媲美。

在 Linux 中，VNC 包括以下四个命令：vncserver, vncviewer, vncpasswd, 和 vncconnect。大多数情况下用户只需要其中的两个命令：vncserver 和 vncviewer。

三、使用VNC远程桌面连接树莓派

2.树莓派的jessie pixel版本默认安装了VNC server,只需打开即可。通过ssh连接至树莓派，然后运行命令：`sudo raspi-config`,即可看到如下图，选中“Interfacing Options”回车进入后，选中“P3 VNC”回车即可。

Raspberry Pi Software Configuration Tool (raspi-config)

- | | |
|------------------------------|--|
| 1 Expand Filesystem | Ensures that all of the SD card storage is available to the OS |
| 2 Change User Password | Change password for the default user (pi) |
| 3 Boot Options | Configure options for start-up |
| 4 Localisation Options | Set up language and regional settings to match your location |
| 5 Interfacing Options | Configure connections to peripherals |
| 6 Overclock | Configure overclocking for your Pi |
| 7 Advanced Options | Configure advanced settings |
| 8 About raspi-config | Information about this configuration tool |

<Select>

<Finish>

Raspberry Pi Software Configuration Tool (raspi-config)

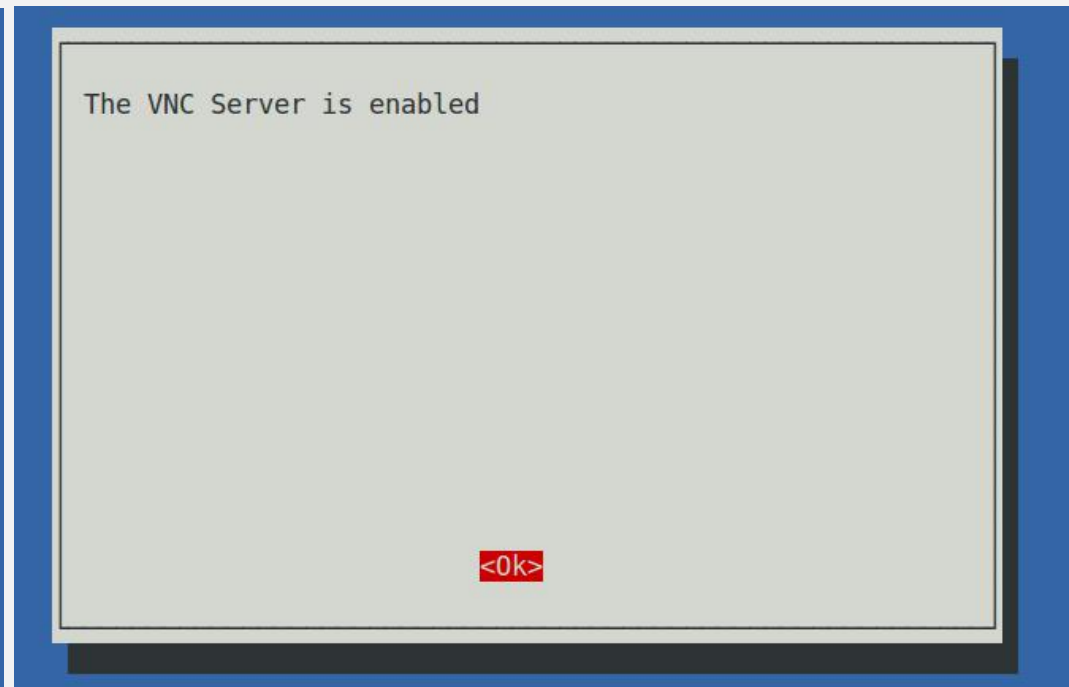
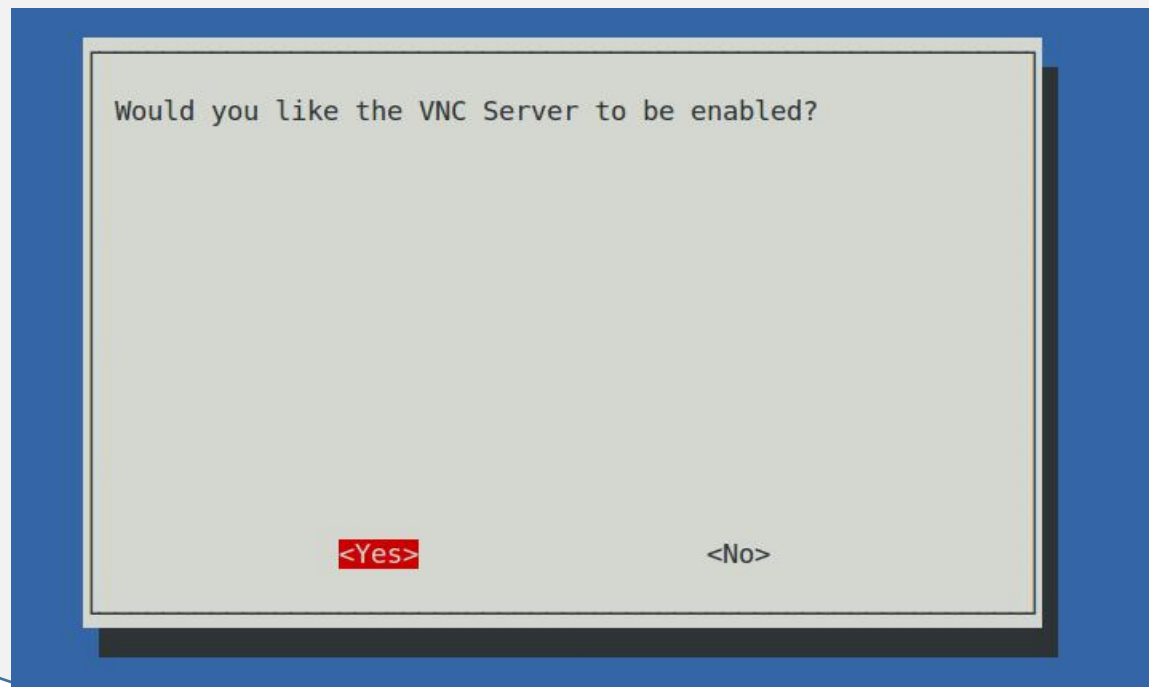
- | | |
|----------------|--|
| P1 Camera | Enable/Disable connection to the Raspberr |
| P2 SSH | Enable/Disable remote command line access |
| P3 VNC | Enable/Disable graphical remote access to |
| P4 SPI | Enable/Disable automatic loading of SPI k |
| P5 I2C | Enable/Disable automatic loading of I2C k |
| P6 Serial | Enable/Disable shell and kernel messages |
| P7 1-Wire | Enable/Disable one-wire interface |
| P8 Remote GPIO | Enable/Disable remote access to GPIO pins |

<Select>

<Back>

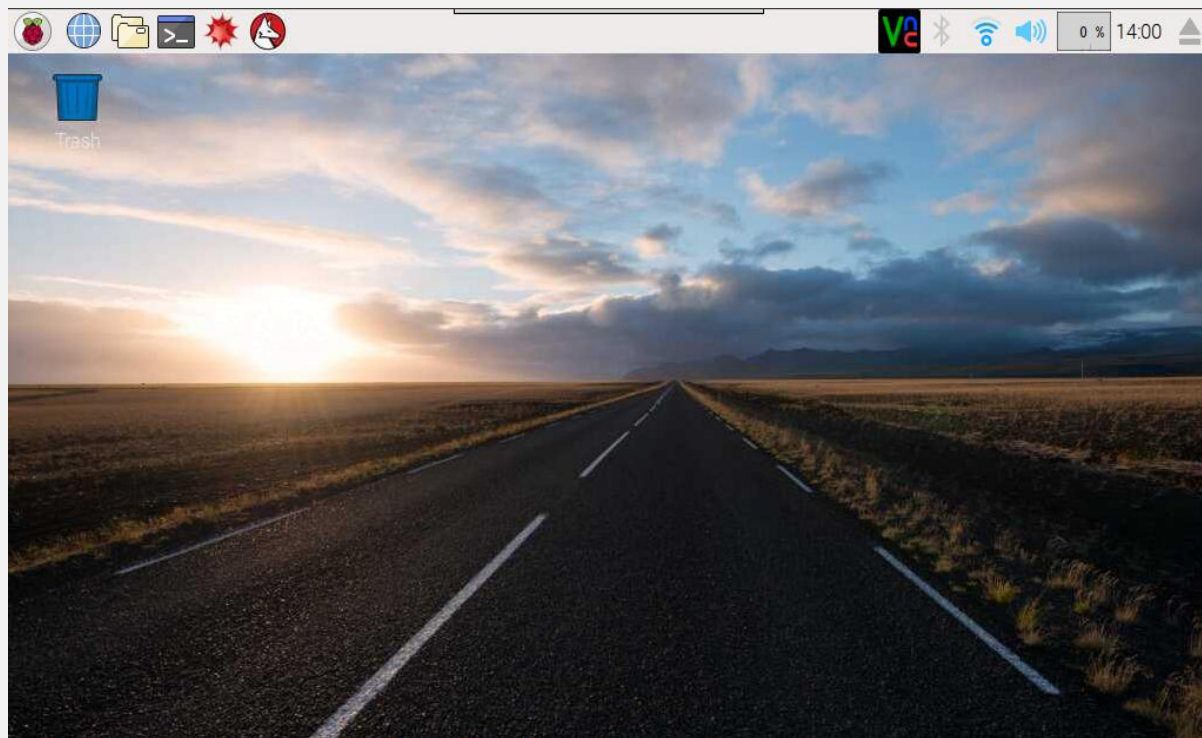
三、使用VNC远程桌面连接树莓派

3.选中“P3 VNC”回车后会出现是否启动VNC server服务的提示，选中Yes即可。出现“The VNC Server is enabled”，即右下图则说明树莓派的VNC server服务开启成功。VNC Server服务成功开启后，以后每次开机启动树莓派时都会自动开机启动。



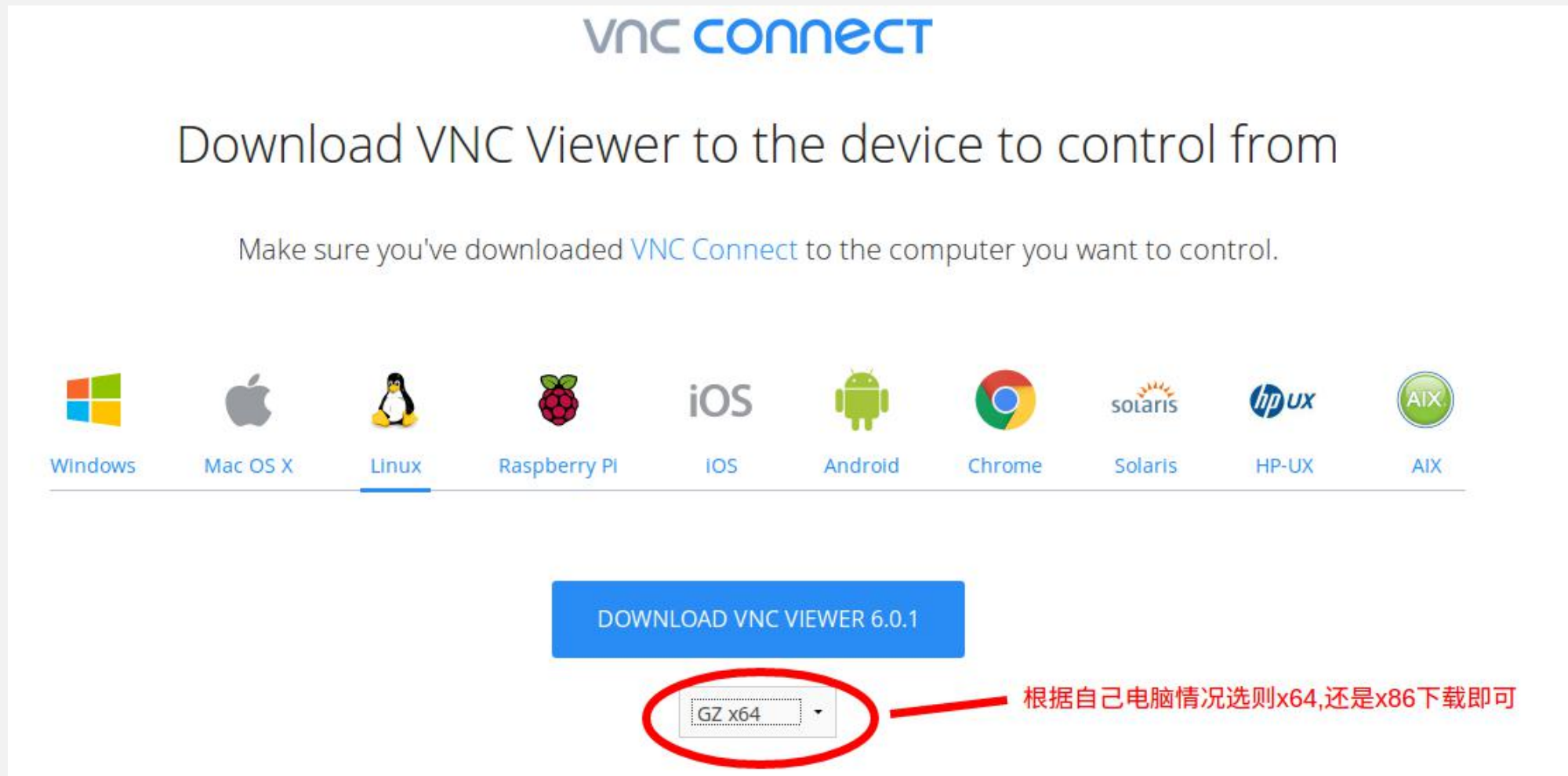
三、使用VNC远程桌面连接树莓派

4.树莓派上成功开启VNC server后，在树莓派的桌面**右上角**会出现VNC的小图标。现在树莓派已经可以作为被控制端电脑，可以被远程桌面连接控制，接下来需要在本地的电脑上（即控制端电脑-ubuntu电脑），安装vncviewer就可以远程连接树莓派了。



三、使用VNC远程桌面连接树莓派

5. 下载vncviewer的网站: <https://www.realvnc.com/download/viewer/linux/>

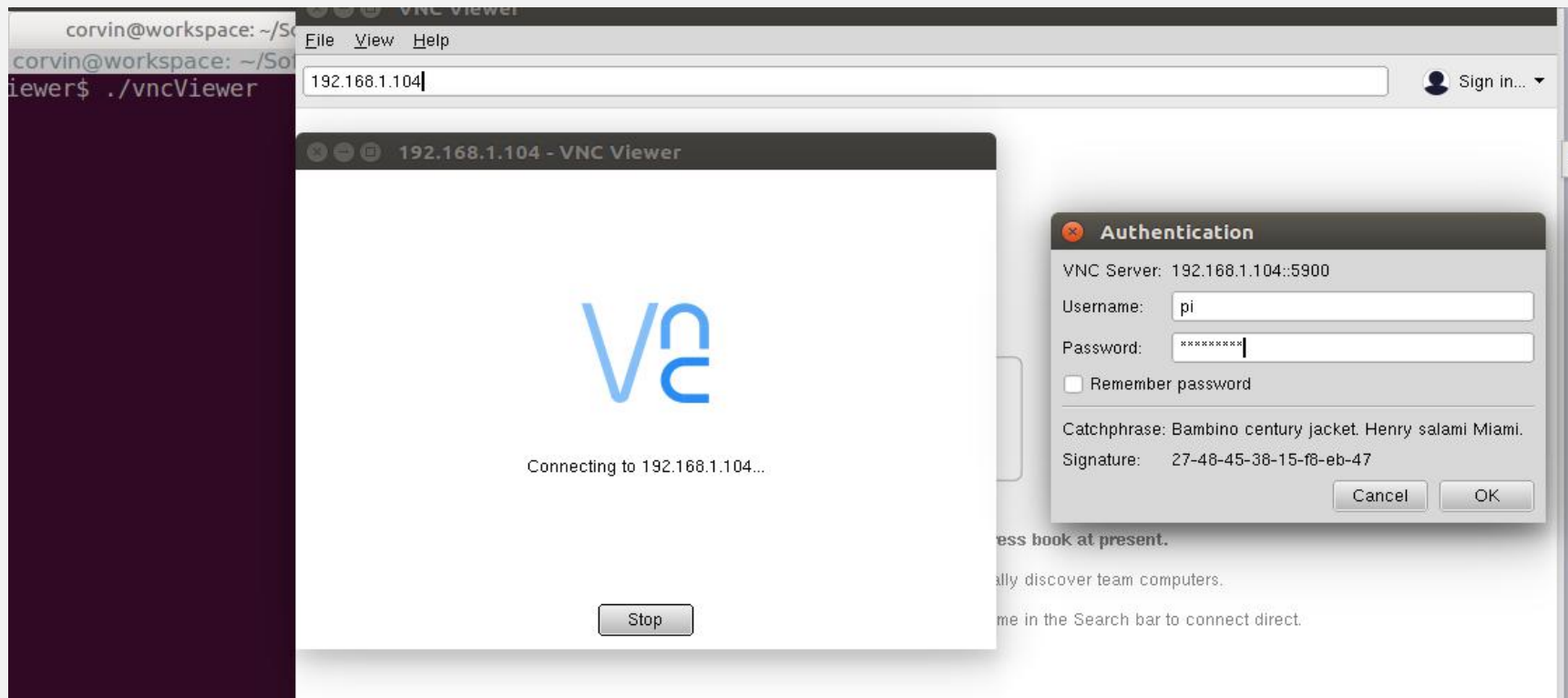


三、使用VNC远程桌面连接树莓派

6. 下载完成后，解压并增加执行权限，即可可以开始远程连接树莓派了。

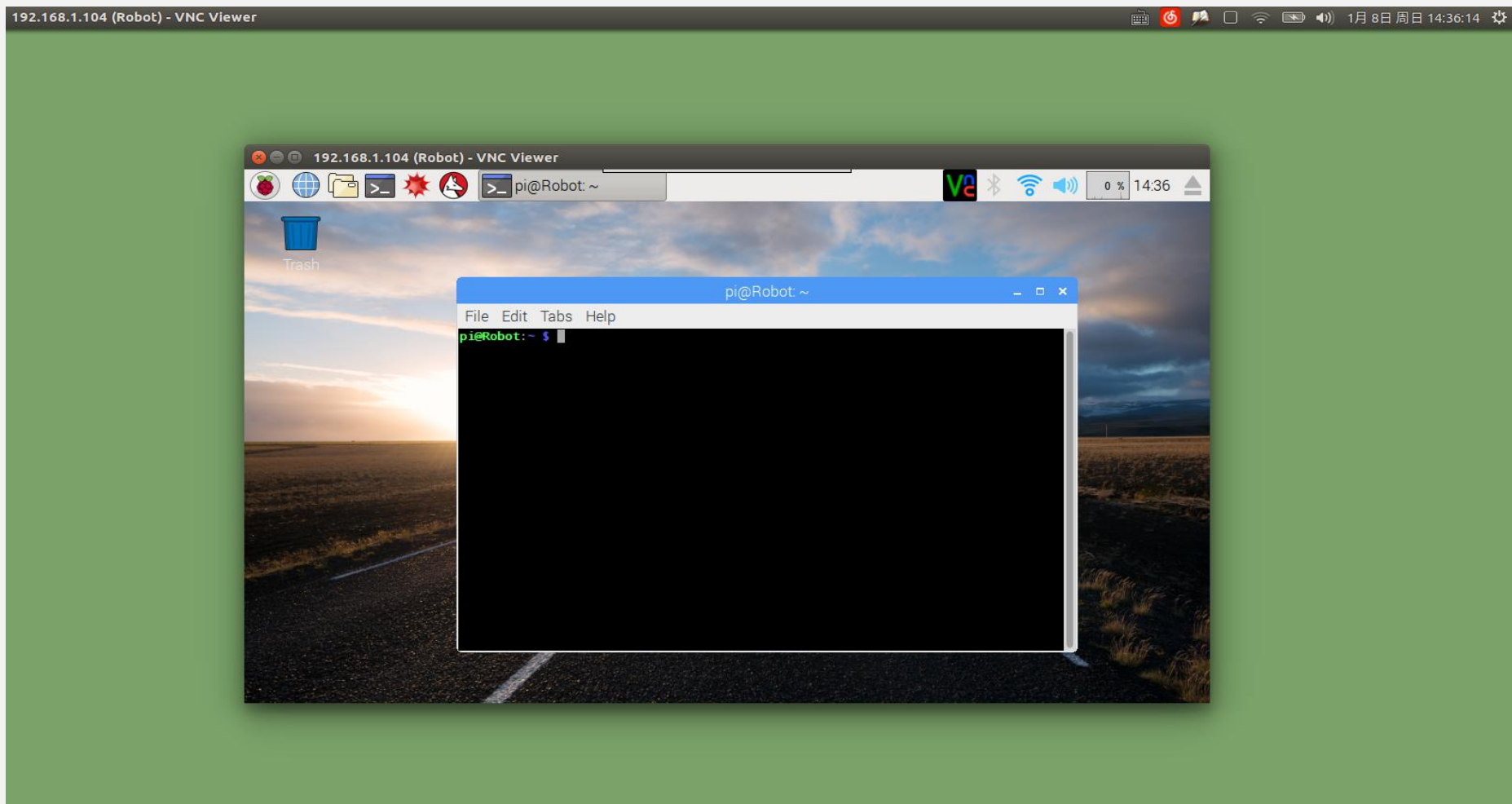
解压命令： `gunzip -c VNC-Viewer-6.0.1-Linux-x64.gz > vncViewer`

增加执行权限： `chmod +x vncViewer`



三、使用VNC远程桌面连接树莓派

7. 输入树莓派正确的用户名、密码即可远程界面连接控制树莓派。



四、搭建Arduino开发环境

1.在树莓派中进行arduino开发，首先需要下载对应版本的IDE，由于树莓派的CPU是ARM芯片，因此需要下载ARM版本的arduino IDE。目前截止到2017年1月21号，ARM版本的最新IDE版本是1.8.1，大小是74.92MB。

下载网站：<https://www.arduino.cc/en/Main/Software>

Download the Arduino IDE



ARDUINO 1.8.1

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

四、搭建Arduino开发环境

2.将下载完成的IDE发送到树莓派上，发送命令如下：

```
scp arduino-1.8.1-linuxarm.tar.xz pi@192.168.1.104:~/Downloads
```

命令行解析：

（1）scp就是secure copy，一个在linux下用来进行远程拷贝文件的命令。有时我们需要获得远程服务器上的某个文件，该服务器既没有配置ftp服务器，也没有做共享，无法通过常规途径获得文件时，只需要通过简单的scp命令便可达到目的。

（2）192.168.1.104 根据自己树莓派的IP地址进行修改；

（3）~/Downloads 是远程主机上保存文件的路径，这个路径可以根据需要自行修改；

四、搭建Arduino开发环境

3.在树莓派中将软件解压，解压命令分两条，解压完成后直接找到arduino执行文件就可以：

(1) `xz -d arduino-1.8.1-linuxarm.tar.xz`

(2) `tar -xvf arduino-1.8.1-linuxarm.tar`

4.在确保arduino IDE可以在树莓派中正常工作，且可以下载程序到arduino板子中，程序可以正常运行且可以查看串口打印输出信息。接下来配置roserial_arduino功能包，这样我们就可以直接在arduino IDE中直接编写ROS代码，roserial提供Arduino和ROS通讯的通信协议，这样的话我们的arduino板子就可以变成一个全双工的ROS节点，可以直接订阅和发布ROS节点消息。

四、搭建Arduino开发环境

5.在树莓派的ROS工作空间中，下载源码并编译该包：

(1). `cd <ws>/src`

(2). `git clone https://github.com/ros-drivers/roserial.git`

(3). `cd <ws>`

(4). `catkin_make -j2`

(5). `catkin_make install`

注意在调用catkin_make时，加上-j2选项，因为默认的是-j4，4个线程会导致树莓派的CPU100%利用率，会导致树莓派死机，因此使用-j2选项虽然编译慢点，但是可以保证可以编译完成。在编译完成后，会在<ws>/install目录下生成ros_lib库。最后不得不调用catkin_make install命令，否则部分ros_lib目录可能会丢失。

四、搭建Arduino开发环境

6.将ros_lib库安装到Arduino开发环境中，共有三条命令：

(1). `cd <sketchbook>/libraries`

(2). `rm -rf ros_lib`

(3). `roslaunch rosserial_arduino make_libraries.py .`

执行完以上命令，基本上就完成了树莓派下Arduino开发环境的搭建，剩下我们需要查看一下ros_lib库是否安装正确，打开IDE，如果可以在File->Examples->ros_lib找到ros_lib选项，则说明安装环境没问题。接下来需要编写代码进行测试。

四、搭建Arduino开发环境

7.需要在树莓派中开启roscore,然后在Arduino IDE中编写测试的代码,验证Arduino节点能否与master节点进行正常通信。测试代码路径:

<http://wiki.ros.org/roscpp/Arduino/Tutorials/Hello%20World>

该测试代码是Arduino作为ROS的一个节点,发布chatter话题,并向话题中发布消息“hello world”字符串。若我们在树莓派中通过rostopic echo /chatter即可查看到该话题中发布信息,那么说明环境搭建成功了。

四、搭建Arduino开发环境

8.启动arduino节点的命令是：

```
roslaunch rosserial_python serial_node.py /dev/ttyACM0
```

但是会启动失败，原因是缺少diagnostic_msgs.msg但是该消息又存在于common_msgs功能包中。需要去github中下载，下载命令如下：

```
git clone https://github.com/ros/common_msgs.git
```

```
pi@Robot:~/Downloads/arduino-1.8.1 $ roslaunch rosserial_python serial_node.py /dev/ttyACM0
Traceback (most recent call last):
  File "/home/pi/ros_catkin_ws/src/rosserial/rosserial_python/nodes/serial_node.py", line 39, in <module>
    from rosserial_python import SerialClient, RosSerialServer
  File "/home/pi/ros_catkin_ws/devel/lib/python2.7/dist-packages/rosserial_python/__init__.py", line 35, in <module>
    exec(__fh.read())
  File "<string>", line 1, in <module>
  File "/home/pi/ros_catkin_ws/src/rosserial/rosserial_python/src/rosserial_python/SerialClient.py", line 51, in <module>
    import diagnostic_msgs.msg
ImportError: No module named msg
```

四、搭建Arduino开发环境

9.在ros的工作空间目录中重新编译整个功能包。所有执行命令如下：

(1). catkin_make -j2

(2). catkin_make install

(3). source devel/setup.bash

(4). rosrun roserial_python serial_node.py /dev/ttyACM0

```
pi@Robot:~/ros_catkin_ws $ source devel/setup.bash
pi@Robot:~/ros_catkin_ws $ rosrun roserial_python serial_node.py /dev/ttyACM0
[INFO] [WallTime: 1484982504.903641] ROS Serial Python Node
[INFO] [WallTime: 1484982504.928680] Connecting to /dev/ttyACM0 at 57600 baud
[INFO] [WallTime: 1484982507.606871] Note: publish buffer size is 280 bytes
[INFO] [WallTime: 1484982507.609339] Setup publisher on chatter [std_msgs/String]

```


四、搭建Arduino开发环境

10.当arduino的节点正常运行后，即可通过rostopic list命令查看，所有的话题列表，通过rostopic echo命令可以查看到/chatter话题中消息。

```
pi@Robot:~/Downloads/arduino-1.8.1 $ rostopic list
/chatter
/diagnostics
/rosout
/rosout_agg
pi@Robot:~/Downloads/arduino-1.8.1 $ rostopic echo /chatter
data: hello world!
---
data: hello world!
---
data: hello world!
---
```

到这里可以查看到/chatter话题中的“hello world”消息，经过实际测试我们在树莓派上搭建Arduino开发环境，圆满完成。

五、增加科大讯飞的语音合成功能

1. 下载科大讯飞的linux版本在线语音合成代码，将其发送到树莓派中。下载地址是：<http://www.xfyun.cn/index.php/sdk/dispatcher>

2. 下载到本地电脑后，仍然通过scp命令将压缩包发送至树莓派系统中。注意那个下载的zip包名称每个人会不相同。

```
scp Linux_voice_1135_57207923.zip pi@192.168.1.104:~/Software
```

3. 在树莓派系统中的Software文件夹中，找到发送的zip包将其解压，然后进入到sample文件夹中，进入到tts_sample文件夹中，该文件夹是在线语音合成的示例代码。我们需要验证测试代码能否正常工作，然后才能将修改后的ROS语音合成代码运行，这样可以减少很多麻烦。

五、增加科大讯飞的语音合成功能

4. 由于我们下载的科大讯飞在线语音合成示例代码是基于x86 CPU的，但是树莓派却是ARM架构的CPU，因此我们若要编译该示例代码需要有ARM版本的动态库才行。下载树莓派版本的语音合成动态库在CSDN上有，下载地址如下：

<http://download.csdn.net/detail/yanghuan313/9616763?web=web>



下载完成后，需要将其发送到树莓派的语音合成示例代码libs文件夹中.然后修改tts_sample示例代码中的Makefile文件就可以正常编译了。修改如下：

```
13 ifdef LINUX64
14 LDFLAGS := -L$(DIR_LIB)/
15 else
16 LDFLAGS := -L$(DIR_LIB)/
17 endif
```

五、增加科大讯飞的语音合成功能

5.直接执行make，即可正常编译，编译完成后，在上层目录的bin文件夹中，可以发现tts_sample的执行文件。

```
pi@Robot:~/Software/voice/samples/tts_sample $ make
gcc -c -g -Wall -I../../include tts_sample.c -o tts_sample.o
gcc -g -Wall -I../../include tts_sample.o -o ../../bin/tts_sample -L../../libs
/ -lmsc -lrt -ldl -lpthread
pi@Robot:~/Software/voice/samples/tts_sample $ cd ../../bin/
pi@Robot:~/Software/voice/bin $ ls
gm_continuous_digit.abnf  msc  source.txt  tts_sample  userwords.txt  wav
```

在运行tts_sample之前，需要将刚才复制到libs文件夹中的libmsc.so复制到/usr/lib中，这样才可以在运行tts_sample时，动态加载到libmsc.so这个动态库。

复制命令：`sudo cp libmsc.so /usr/lib`



五、增加科大讯飞的语音合成功能

6.做好以上准备后，就可以直接开始调用tts_sample了。执行完成后，在当前文件夹中会生成tts_sample.wav文件，该文件就是合成的目标文件，我们只需要将其播放出来即可。

[illegible]

7.在树莓派中安装sox软件包，通过play命令即可播放wav文件。

```
sudo apt-get -y install sox
```

五、增加科大讯飞的语音合成功能

8.通过play命令即可播放合成的tts_sample.wav文件,到这里我们整个的语音合成的环境已经准备就绪。接下来就可以将ROS下的语音合成代码复制过来,在树莓派下编译即可正常运行了。

```
pi@Robot:~/Software/voice/bin $ play tts_sample.wav

tts_sample.wav:

  File Size: 525k      Bit Rate: 256k
  Encoding: Signed PCM
  Channels: 1 @ 16-bit
  Samplerate: 16000Hz
  Replaygain: off
  Duration: 00:00:16.42

In:100% 00:00:16.42 [00:00:00.00] Out:263k [      |      ] Hd:5.2 Clip:0
Done.
```


五、增加科大讯飞的语音合成功能

9.创建ROS代码的工作空间，首先创建catkin_ws/src文件夹，然后在src文件夹中执行catkin_init_workspace命令，这样就初始化工作空间了。以后就可以在catkin_ws目录下进行catkin_make编译ROS代码了。

```
pi@Robot:~ $ mkdir -p catkin_ws/src
pi@Robot:~ $ cd catkin_ws/src/
pi@Robot:~/catkin_ws/src $ catkin_init_workspace
Creating symlink "/home/pi/catkin_ws/src/CMakeLists.txt" pointing to "/opt/ros/indigo/share/catkin/cmake/toplevel.cmake"
pi@Robot:~/catkin_ws/src $ ls
CMakeLists.txt
pi@Robot:~/catkin_ws/src $ cd ..
pi@Robot:~/catkin_ws $ catkin_make
Base path: /home/pi/catkin_ws
Source space: /home/pi/catkin_ws/src
Build space: /home/pi/catkin_ws/build
Devel space: /home/pi/catkin_ws/devel
Install space: /home/pi/catkin_ws/install
####
#### Running command: "cmake /home/pi/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/pi/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/pi/catkin_ws/install -G Unix Makefiles" in "/home/pi/catkin_ws/build"
```

五、增加科大讯飞的语音合成功能

10.之所以不在ros_catkin_ws文件夹中编写自己的测试代码，是因为该工作空间只用来编译跟ROS相关的系统功能包，将需要增加的ROS功能包代码git clone至src文件中后，就可以通过以下命令编译并将编译后功能包安装到系统环境中。

```
sudo ./src/catkin/bin/catkin_make_isolated --install -
```

```
DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/indigo -j2
```

11.接下来将ROS下的语音系统相关的代码文件夹拷贝到刚才创建的catkin_ws工作空间中，注意需要修改一些其中的小错误，例如存储和播放合成文件的路径需要修改。注释掉CMakeLists.txt文件中其他节点的编译信息，目前只编译tts相关的代码。然后就可以直接调用catkin_make进行编译了。



五、增加科大讯飞的语音合成功能

12.编译完成后，就可以进行测试了。首先启动roscore，然后启动tts的节点。启动xf_tts_node节点完成后，会有语音播报提示。

```
pi@Robot:~/catkin_ws $ rosrn voice_system xf_tts_node
```

开始合成...

正在合成...

[illegible]

合成完毕

```
/home/pi/Music/voice.wav:
```

File Size: 131k Bit Rate: 256k

Encoding: Signed PCM

Channels: 1 @ 16-bit

Samplerate: 16000Hz

Replaygain: off

Duration: 00:00:04.10

```
In:100% 00:00:04.10 [00:00:00.00] Out:65.6k [  |  ] Hd:2.4 Clip:0
```

Done.

五、增加科大讯飞的语音合成功能

13.通过rostopic list命令即可查看话题列表，接下来测试语音合成模块是否可以正常工作。最后通过rostopic pub命令往话题中发送需要合成的文本即可。

```
pi@Robot:~/catkin_ws $ rostopic list
/rosout
/rosout_agg
/voice/xf_tts_topic
pi@Robot:~/catkin_ws $ rostopic info /voice/xf_tts_topic
Type: std_msgs/String

Publishers: None

Subscribers:
 * /xf_tts_node (http://Robot:35513/)
```

```
pi@Robot:~/catkin_ws $ rostopic pub -1 /voice/xf_tts_topic std_msgs/String "哼
！都怪你,也不哄哄人家,人家超想哭的,捶你胸口,老公！大坏蛋！！！咩,捶你胸口 你
好讨厌！要抱抱嘤嘤嘤哼,人家拿小拳拳捶你胸口！！！大坏蛋,打死你,人家超想哭的
。"
publishing and latching message for 3.0 seconds
```