

安装树莓派

【选择树莓派的三大理由】

1. 使用人多——各种软件会对它兼容适配
2. 省电——适合 7*24 小时开机
3. 性价比高——几百元，足够完成日常的各种学习与自动控制需求

【硬件准备】

- 树莓派 3B、3B+、4B



- 一张 16G 以上 tf 卡、一个 USB 口 tf 卡读写卡器



- 一台 Windows 或 MacOS 工作电脑

【操作步骤】

1. 下载树莓派镜像 (<https://www.raspberrypi.org/downloads/raspbian/>)，解压缩
2. 下载安装 etcher (<https://etcher.io/>)
3. 使用 etcher 将镜像写入 tf 卡
4. 配置打开 sshd，配置树莓派的 wifi

wpa_supplicant.conf 文件

```
country=CN
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="your_wifi_ssid"
    psk="your_wifi_password"
    key_mgmt=WPA-PSK
    priority=1
}
```

5. 启动树莓派，确认树莓派的 IP 地址
6. 通过 ssh 登录树莓派

MacOS 和 linux 上连接命令：ssh 树莓派 IP 地址

Windows 上使用 putty 软件连接

【参考】

- putty 下载地址
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

注：对于网络不好，或者希望偷懒的用户，可以直接使用 HACHina 的镜像文件，其中已经安装好了 HomeAssistant、NodeRed、JupyterNotebook、Samba 等软件，可以直接使用。

地址：<https://www.hachina.io/8408.html>

树莓派基础配置

【操作步骤】

1. 查看系统基础信息
2. 修改 pi 用户密码
3. 使用 raspi-config 修改时区、打开 vnc
4. 修改 APT 与 PIP 的配置
5. 安装一些常用的基础库

【参考】

- /etc/apt/sources.list 文件
`deb http://mirrors.aliyun.com/raspbian/raspbian/ buster main non-free contrib rpi`
`deb-src http://mirrors.aliyun.com/raspbian/raspbian/ buster main non-free contrib rpi`
- /etc/pip.conf 文件
`[global]`
`index-url=https://mirrors.aliyun.com/pypi/simple/`
`extra-index-url=https://www.piwheels.org/simple`
- 安装一些常用的基础库
`sudo apt-get install libevent-dev libjpeg-dev build-essential libssl-dev libffi-dev python3-dev libavahi-compat-libdnssd-dev libatlas-base-dev autoconf python3-venv`
- VNC Viewer
<https://www.realvnc.com/en/connect/download/viewer/>

安装 Samba 和 Jupyter Notebook

【操作步骤】

1. 安装、配置、使用 Samba
2. 安装、配置、使用 Jupyter Notebook

【参考】

- Samba 命令
安装: `sudo apt-get install samba samba-common`
增加用户 pi 并设置密码: `sudo smbpasswd -a pi`
重启 samba 服务: `sudo systemctl restart smbd`
- Samba 配置文件位置
`/etc/samba/smb.conf`
- 如果打开共享是空的 (未显示共享目录)
在 Windows 的 DOS 窗口中, 运行命令:
`net use z: \\raspberrypi password /USER:pi`
修改其中 password 为你的 password, 如果要保持 z 盘长久映射, 在命令中增加:
`/PERSISTENT:YES`
注: 很多 windows 缺省使用匿名访问共享, 而我们的共享是以 pi 用户共享的, 所以就看不到。
使用这个命令, 指定使用 pi 用户访问, 将共享的目录映射为 windows 下的 z 盘
- Jupyter Notebook 命令
安装: `sudo pip3 install jupyter`
生成配置文件: `jupyter notebook --generate-config`
设置访问密码: `jupyter notebook password`
- 修改 Jupyter Notebook 的配置 (文件位置 `~/jupyter/jupyter_notebook_config.py`)
在文件中找到下面的 4 个配置项, 去除前面的注释符号 (#), 并修改等号后的值为下面的值。
`c.NotebookApp.ip = '0.0.0.0'`
`c.NotebookApp.open_browser = False`
`c.NotebookApp.port = 8888`
`c.NotebookApp.notebook_dir = '/home/pi/'`

安装 HomeAssistant

1. 常见安装方式

- 在系统 python 环境中安装
- 在 python 虚拟环境中安装
- docker
- hass.io
- hassbian

2. 安装依赖关系

组件	依赖的 python 库
homeassistant (core)	aiohttp/astral/bcrypt/pyyaml/pytz/requests/voluptuous/.....
http	aiohttp_cors
api	
frontend	home-assistant-frontend
recorder	sqlalchemy
zeroconf	zeroconf
bitcoin	blockchain
.....

3. 安装 HomeAssistant

命令: `sudo pip3 install homeassistant`

执行内容: 下载并安装上表中蓝色部分

4. 启动 HomeAssistant

命令: `hass`

执行内容:

如果没有配置文件, 生成配置目录与缺省配置文件

加载配置文件中的组件 (以及其依赖的组件), 安装对应依赖的 python 库

5. 检查

进程: `ps -ef|grep hass`

网络: `netstat -an|grep 8123`

配置目录与配置文件: `/home/pi/.homeassistant/`

错误日志文件: `/home/pi/.homeassistant/home-assistant.log`

浏览器访问: `http://xx.xx.xx.xx:8123/`

HomeAssistant 配置、文件结构、升级

1. 第一次进入 HomeAssistant 界面
添加用户、设置地理位置
2. 当前用户配置
3. 前端配置组件（集成配置）——配置区域（zone）
4. 使用配置文件配置组件——配置比特币行情
5. 配置目录结构与内容

文件或目录	内容
configuration.yaml	用户编辑的配置文件
home-assistant.log	运行日志（每次重启会清空）
home-assistant_v2.db	数据库
.storage 目录	前端配置的各种内容
deps 目录	自动安装的各种 python 依赖包

6. 升级 HomeAssistant
命令：`sudo pip3 install homeassistant --upgrade`

【参考】

- yaml 格式
<https://www.hackplayers.io/docs/335.html>
- 比特币配置

```
sensor:  
  - platform: bitcoin  
    display_options:  
      - exchangerate  
      - trade_volume_btc
```

HomeAssistant 自启动

【操作步骤】

1. 自启动配置文件
2. 设置自启动
3. 查看日志输出
4. 手动控制服务
5. 同样操作，设置 Jupyter-notebook 的自启动

【参考】

- /etc/systemd/system/home-assistant@pi.service 文件

```
[Unit]
Description=Home Assistant
After=network.target
[Service]
Type=simple
User=%i
ExecStart=/usr/local/bin/hass
[Install]
WantedBy=multi-user.target
```
- /etc/systemd/system/jupyter-notebook@pi.service 文件

```
[Unit]
Description=Jupyter Notebook
[Service]
Type=simple
ExecStart=/usr/local/bin/jupyter-notebook
User=%i
[Install]
WantedBy=multi-user.target
```
- 自启动服务相关命令 (以 home-assistant@pi 服务为例)
 - 重载服务配置: `sudo systemctl --system daemon-reload`
 - 将服务加入自启动: `sudo systemctl enable home-assistant@pi`
 - 将服务移除自启动: `sudo systemctl disable home-assistant@pi`
 - 手工启动服务: `sudo systemctl start home-assistant@pi`
 - 手工停止服务: `sudo systemctl stop home-assistant@pi`
 - 手工重启服务: `sudo systemctl restart home-assistant@pi`
 - 查看服务输出: `sudo journalctl -f -u home-assistant@pi`
- 增加一些快捷命令

```
mkdir ~/bin/
echo sudo systemctl restart home-assistant@pi > ~/bin/ha-restart
echo sudo systemctl start home-assistant@pi > ~/bin/ha-start
echo sudo systemctl stop home-assistant@pi > ~/bin/ha-stop
echo sudo journalctl -fu home-assistant@pi > ~/bin/ha-log
chmod +x ~/bin/*ha-*
```

HomeAssistant 快速使用

1. 面临的问题
 - 网络访问国外资源困难
 - 耗费大量精力
 - 运行环境散乱
2. 使用 HACHina 的 HASSOS 镜像
 - 第一次正式启动无需访问 INTERNET
 - 已安装好所有官方组件依赖的 python 库，以及一些常用的 add-on
 - 配置了国内镜像站点，未来升级与安装比较快速
 - 官方推荐的标准运行环境
3. 镜像下载与烧写

https://pan.baidu.com/s/1INCX_0wkHnGdzJIBJyRuHQ (提取码 1024)

4. 配置网络与 ssh 访问

如果使用网线，不使用 wifi 连接，请整个删除 network 目录
如果不需要 ssh 访问，请删除 authorized_keys 文件
参考：如何生成自己的访问密钥 <https://developers.home-assistant.io/docs/operating-system/debugging#generating-ssh-keys>
5. 启动与访问
6. 一些常用操作
 - Supervisor 菜单
 - ssh 访问

参考：

https://github.com/zhujisheng/Home-Assistant-DIY/tree/master/hassos_image