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

}

1. 启动树莓派，确认树莓派的IP地址
2. 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: \\raspiberry\_ip\pi 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 |
| …… | …… |

1. 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

1. 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

1. 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

1. 当前用户配置
2. 前端配置组件（集成配置）——配置区域（zone）
3. 使用配置文件配置组件——配置比特币行情
4. 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 |

1. 升级HomeAssistant

命令：sudo pip3 install homeassistant --upgrade

【参考】

* yaml格式

<https://www.hachina.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](mailto:/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](mailto:/etc/systemd/system/home-assistant@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）

1. 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>

1. 启动与访问
2. 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>