**烧写系统到NanoPi S2的eMMC**

* 下载eflasher固件

固件文件名为：s5p4418-eflasher-sd8g-xxx-full.img.7z   
这个固件包含了一个可用SD卡启动的微型Ubuntu core系统，以及最新版本的 Debian, Android5, Android4.4和Ubuntu core 系统映象文件；  
将 s5p4418-eflasher-sd8g-xxx-full.img.7z 下载到电脑上，另外还需要下载Windows下的烧写工具: win32diskimager.rar；

* 用eflasher固件制作SD启动卡

将固件用7z软件解压，得到.img结尾的文件，在Windows下插入SDHC卡（限4G及以上的卡)，以管理员身份运行Win32DiskImager，选择 Image File载入固件，选择你的SD卡盘符，点 Write 即开始写Image到SD卡；  
如果你的电脑用的是Linux系统，你也可以用 dd 命令将解压后得到的 .img 文件直接写入SD卡完成启动卡的制作；

* 烧写系统到eMMC

将制作好的SD卡插入NanoPi S2，连接HDMI或LCD, 按住标住有“BOOT”字样的按键，上电，启动开发板后会看到系统选择，选择你需要烧写到eMMC的系统即可。

### 连接WiFi

FriendlyCore 使用 NetworkManager 工具来管理网络，其在命令行下对应的命令是 nmcli，要连接WiFi，相关的命令如下：

* 查看网络设备列表

$ **sudo** nmcli dev

注意，如果列出的设备状态是 unmanaged 的，说明网络设备不受NetworkManager管理，你需要清空 /etc/network/interfaces下的网络设置,然后重启.

* 开启WiFi

$ **sudo** nmcli r wifi on

* 扫描附近的 WiFi 热点

$ **sudo** nmcli dev wifi

* 连接到指定的 WiFi 热点

$ **sudo** nmcli dev wifi connect "SSID" password "PASSWORD"

请将 SSID和 PASSWORD 替换成实际的 WiFi名称和密码。  
连接成功后，下次开机，WiFi 也会自动连接。

串口模块安装

sudo apt install python3-pip

pip3 install –upgrade pip

sudo pip3 install pyserial

redis安装

pip3 install redis

wget http://download.redis.io/releases/redis-4.0.6.tar.gz

tar xzf redis-4.0.6.tar.gz

cd redis-4.0.6

make

启动服务端:

src/redis-server &

启动客户端并操作:

src/redis-cli

redis> set foo bar

OK

redis> get foo

"bar"

关闭持久化

1. #save 900 1
2. #save 300 10
3. #save 60 10000

或save ""

密码设置 #requirepass foobared。设置密码的方法就是去掉注释的#，把foobared替换成自己的密码即可，例如将密码设置为123456

Postgresql 的psycopg2安装

sudo vi /etc/resolv.conf

nameserver 127.0.0.1

search lan

sudo apt-get update

apt-get install libpq-dev python3-dev

pip3 install psycopg2

import psycopg2

conn = psycopg2.connect(database="materiel", user="pms", password="pms@pg123", host="58.214.232.165", port="5432")

cur = conn.cursor()

cur.execute("INSERT INTO tb\_epc\_record (id,device\_id,epc,report\_time) \

VALUES (0, '1234567890', '666666666', ‘’ )")

conn.commit()

conn.close()

一般都会推荐使用国内的镜像源，比如163或者阿里云的镜像服务器

将下列文本添加到/etc/apt/sources.list文件里

deb http://mirrors.aliyun.com/ubuntu/ raring main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ raring-security main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ raring-updates main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ raring-proposed main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ raring-backports main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ raring main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ raring-security main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ raring-updates main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ raring-proposed main restricted universe multiverse

deb-src http://mirrors.aliyun.com/ubuntu/ raring-backports main restricted universe multiverse

但是执行sudo apt-get update仍然报错，问题在于DNS没有配置好。

解决方法：

sudo vi /etc/resolv.conf

1

添加

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)

# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN

nameserver 127.0.1.1

#这里用的是阿里云的DNS服务器

nameserver 223.5.5.5

nameserver 223.6.6.6

之后执行sudo apt-get update就正常了。

Sudo ls –a /home/pi/

Sudo vim /home/pi/.profile

/opt/20180306/runrfid.sh

Sudo vim /home/pi/.bashrc

lxterminal

Sudo vim /root/.bashrc

//////////////////////////////////////////////////////使用

vim /etc/rc.local

exec /sbin/getty -n -l /opt/runrfid.sh -8 115200 tty1

chmod a+x xxx.sh

runrfid.sh

#! /bin/bash

# /bin/login –f root

pwd

#sudo python3 CUartThread.py

#sudo python3 CUart.py

cd /opt/redis-4.0.8/src/

sudo ./redis-server /opt/redis-4.0.8/src/redis.conf

sleep

cd /opt/20180316/

sudo python3 /opt/20180316/RfidReadAndWrite-Psql.py

//////////////////////////////////////////////////////使用