



Orange Pi Lite 2

用户手册





目录

一、 Orange Pi Lite 2 的基本特性-----	1
1. 什么是 Orange Pi Lite 2-----	1
2. Orange Pi Lite 2 的用途-----	1
3. Orange Pi Lite 2 是为谁设计的-----	1
4. Orange Pi Lite 2 的硬件特性-----	3
5. GPIO 规格-----	4
二、 开发板使用说明-----	5
1. 准备硬件与软件工具-----	5
2. 开发板的供电方式-----	5
三、 Android 编译环境搭建-----	6
1. 获取 SDK 源码压缩包-----	6
2. 搭建编译环境-----	6
3. 编译 SDK 源码-----	7
四、 Linux 编译环境搭建-----	10
1. 从 Github 获取 Linux SDK 的源码-----	10
2. 从百度云盘获取 Linux SDK 的源码-----	12
3. 搭建编译环境-----	12
4. 编译 Linux 和 U-boot 源码-----	12
5. Linux SDK 使用示例-----	14
五、 Android 固件烧录-----	18
1. Android 固件烧录步骤-----	18
六、 Linux 固件烧录-----	20
1. Etcher 的安装方式-----	20
2. 通过 Etcher 烧录 Linux 固件的方法-----	21
七、 Linux 系统的使用说明-----	22
1. Linux 启动亮灯情况说明-----	22
2. 登录账号和密码-----	22
3. 扩展 rootfs 分区-----	22
4. 录音放音测试方法-----	23
5. WIFI 的配置方法-----	23
八、 串口调试工具介绍-----	24



1. 基于 Windows 平台的使用-----	25
2. 基于 Linux 平台的使用-----	29



一、Orange Pi Lite 2 的基本特性

1. 什么是 Orange Pi Lite 2

香橙派是一款开源的单板卡片电脑，新一代的 arm64 开发板，它可以运行 Android 7.0、Ubuntu 和 Debian 等操作系统。香橙派单板电脑使用全志 H6 系统级芯片，同时拥有 1GB LPDDR3 内存。

2. Orange Pi Lite 2 的用途

我们可以用它搭建：

- 一台计算机
- 一个网络服务器
- 游戏机
- 高清视频播放器
- 扬声器
- Android
-

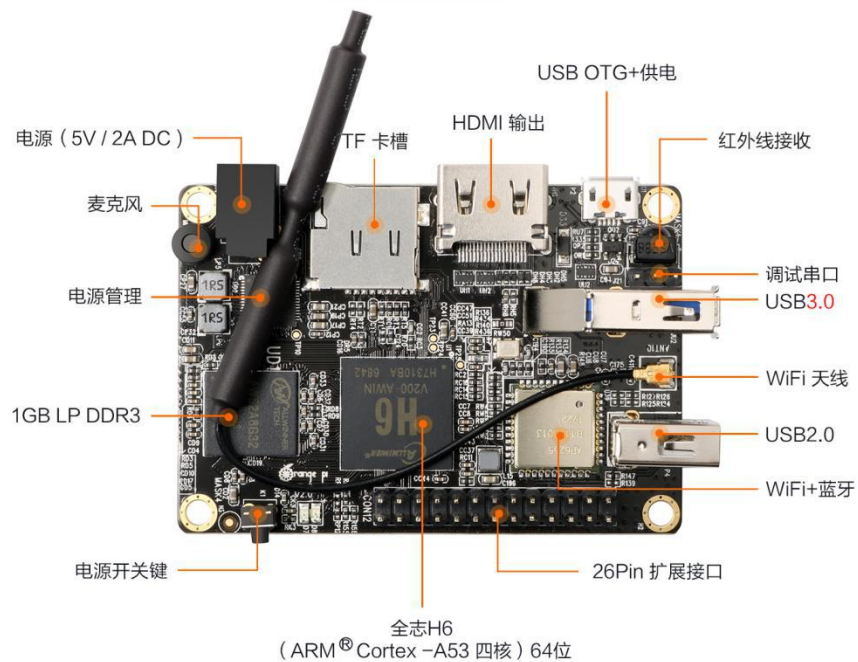
还有其他更多的功能，因为 Orange Pi Lite 2 是开源的。

3. Orange Pi Lite 2 是为谁设计的

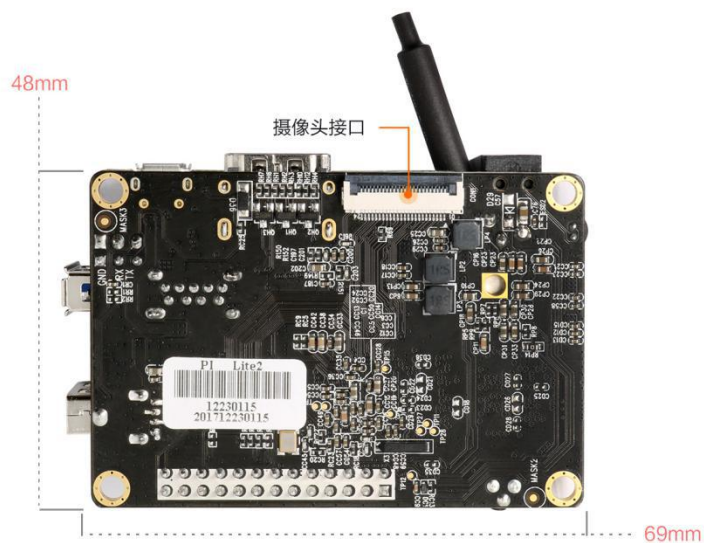
Orange Pi Lite 2 不仅仅是一款消费品，同时也是给任何想用技术进行创作创新的人设计的。它是一款非常简单、有趣、实用的工具，你可以用它去打造你身边的世界。



顶层视图



底层视图





4. Orange Pi Lite 2 的硬件特性

硬件特性介绍	
CPU	高性能全志 H6 芯片，4 核 64 位 Cortex-A53
GPU	高性能多核 GPU Mali T720
内存	1GB LPDDR3（与 GPU 共享）
板载存储	TF 卡（最大 32GB）
板载 WIFI+BT	AP6255, IEEE 802.11 AC/b/g/n/, BT4.1
音频输入	MIC
音频输出	HDMI
视频输入	具有 CSI 接口的摄像头模块
视频输出	HDMI 2.0a
电源	DC 输入，MicroUSB（OTG）可以用作电源输入
USB 2.0 端口	1 个 USB 2.0 HOST, 1 个 USB 2.0 OTG
USB 3.0 端口	1 个 USB 3.0 HOST
底层设备	40 pin 接头
GPI0(1x3) 口	UART, ground.
LED 灯	电源指示灯和状态指示灯
按键	电源（K1）
支持的操作系统	Android、Ubuntu 和 Debian 等操作系统
外观规格介绍	
产品尺寸	68mm × 48mm
重量	31.5 克
OrangePi	OrangePi™是深圳市迅龙软件有限公司的注册商标



5. GPIO 规格

下图是香橙派 Lite 2 的 GPIO 引脚功能图：



Orange Pi Lite 2 引脚对应表格		
CON12-P01	VCC-3.3V	VCC-I0
CON12-P02	VCC-5V	DCIN
CON12-P03	TWI1-SDA	PH06
CON12-P04	VCC-5V	DCIN
CON12-P05	TWI1-SCK	PH05
CON12-P06	GND	GND
CON12-P07	PWM1	PH04
CON12-P08	PD21	PD21
CON12-P09	GND	GND
CON12-P10	PD22	PD22
CON12-P11	UART3_RX	PD24
CON12-P12	PC09	PC09
CON12-P13	UART3_TX	PD23
CON12-P14	GND	GND
CON12-P15	UART3_CTS	PD26
CON12-P16	PC08	PC08
CON12-P17	VCC-3V3	VCC-I0
CON12-P18	PC07	PC07
CON12-P19	SPI0_MOSI	PC02
CON12-P20	GND	GND
CON12-P21	SPI0_MISO	PC03
CON12-P22	UART3_RTS	PD25
CON12-P23	SPI0_CLK	PC00
CON12-P24	SPI0_CS0	PC05
CON12-P25	GND	GND
CON12-P26	PH03	PH03



二、开发板使用说明

1. 准备硬件与软件工具

硬件需求：

- Orange Pi Lite 2 开发板
- TF 卡，最小 8GB 容量，class 10 级，建议使用品牌 TF 卡，如：闪迪 16G TF 卡
- 一台编译用的主机，配置最好满足以下条件：
 - 64 位 CPU
 - 8 GB 及以上内存
 - 100GB 以上的空闲磁盘空间
 - 操作系统最好为 Ubuntu14.04（用于编译 Android 源码）
Ubuntu18.04（用于编译 Linux 源码）

软件需求：

- Orange Pi Lite 2 SDK
- Orange Pi Lite 2 固件
- Android 和 Linux 烧录工具

以上软件可以通过 Github、Mega 网盘和百度云盘获取，详情参见中英文官网：

<http://www.orangepi.org/downloadresources/>

<http://www.orangepi.cn/downloadresourcescn/>

2. 开发板的供电方式

开发板的供电方式有两种：

- DC（5V 2A）供电：
插入 DC 适配器后即可开机。
- Micro USB（5V 2A）OTG 供电：
插入 Micro USB 适配器后即可开机。



三、Android 编译环境搭建

以下的操作都是在安装有 Ubuntu 14.04 的 PC 上进行的，其它版本的 Ubuntu 系统或者 Linux 发行版可能会有一些区别。

1. 获取 SDK 源码压缩包

Android 源码包下载完后，首先需要将多个压缩文件合并成一个，然后进行解压。

```
$ mkdir OrangePi_Lite2
$ cat H6-2018-1-2.tar.gz* > OrangePi_Lite2.tar
$ tar xf OrangePi_Lite2.tar -C OrangePi_Lite2
```

2. 搭建编译环境

● 安装 JDK

Android 7.0 开发只能使用 openjdk8 的版本，高于或低于此版本以及 oracle 的 JDK 都会导致编译失败。Openjdk-8 的安装命令如下：

```
$ sudo add-apt-repository ppa:openjdk-r/ppa
$ sudo apt-get update
$ sudo apt-get install openjdk-8-jdk
```

● 配置 JAVA 环境变量

比如安装路径为 `/usr/lib/jvm/java-8-openjdk-amd64`，可以在终端执行如下命令配置环境变量

```
$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
$ export PATH=$JAVA_HOME/bin:$PATH
$ export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar
```



● 安装平台支持软件

对于 Ubuntu14.04:

```
$ sudo apt-get update
$ sudo apt-get install git gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip

$ sudo apt-get install u-boot-tools
```

3. 编译 SDK 源码

SDK 解压后，解压的目录下会有 andorid 和 lichee 两个子目录，lichee 目录主要内容如下：

lichee/brandy/u-boot-2014.07	#uboot 代码目录
lichee/bootloader/uboot_2014_sunxi_spl	#boot0 代码目录
lichee/linux-3.10	#内核代码
lichee/tools	#方案硬件配置，打包工具等

● 内核编译流程

在 lichee 目录下输入以下命令：

```
$ cd OrangePi_Lite2/lichee
$ ./build.sh config

Welcome to mkscript setup progress
All available chips:
  0. sun50iw1p1
  1. sun50iw2p1
  2. sun50iw6p1
  3. sun8iw11p1
  4. sun8iw12p1
  5. sun8iw6p1
```



```
6. sun8iw7p1
7. sun8iw8p1
8. sun9iwlpl
Choice: 2
All available platforms:
0. android
1. dragonboard
2. linux
3. eyeseelinux
Choice: 0
All available business:
0. 5.1
1. 4.4
2. 7.x
Choice: 2
```

编译成功后输出内容如下:

```
regenerate rootfs cpio
15757 blocks
17099 blocks
build_ramfs
Copy boot.img to output directory ...
Copy modules to target ...

sun50iw6p1 compile Kernel successful

INFO: build kernel OK.

INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: build rootfs OK.
-----
build sun50iw6p1 android 7.x lichee OK
-----
```

内核代码在 `lichee/linux-3.10` 目录, 执行上述命令在编译前会自动将配置文件从 `lichee/linux-3.10/arch/arm64/configs/sun50iw6p1smp_android_7.x_defconfig` 拷贝到 `lichee/linux-3.10/.config` 作为默认配置, 下次编译时可以直接在 `lichee` 目录下运行 `./build.sh`, 将继续采用上一次的 `.config` 配置。



● uboot/boot0 编译流程（可选）

通常情况下无需重新编译 uboot，但如果对 uboot 有定制修改可以编译，编译方法如下：

```
cd lichee/brandy/u-boot-2014.07
make distclean && make sun50iw6p1_config && make -j5 #编译 uboot

cd lichee/brandy/u-boot-2014.07
make distclean && make sun50iw6p1_config && make spl #编译 boot0
```

如果没有编译 uboot/boot0 的话，默认是采用 lichee/tools/pack/chips/sun50iw6p1/bin 已经预编译好的结果，采用上述命令重编译后，将会自动替换掉上述文件。

● Android 代码编译流程

```
$ cd android
$ source ./build/envsetup.sh
$ lunch petrel_fvd_p1-eng
$ extract-bsp
$ make -j8 && pack
```

其中 pack 命令是打包生成固件，如果编译打包过程都顺利通过，会提示下面的信息：

```
Dragon execute image.cfg SUCCESS !
-----image is at-----

OrangePi_Lite2/lichee/tools/pack/sun50iw6p1_android_petrel-p1_uart0.img

pack finish
```

根据提示，可以在 OrangePi_Lite2/lichee/tools/pack/ 目录下看到生成的安卓固件 sun50iw6p1_android_petrel-p1_uart0.img，下面，请参照《Android 固件烧录》部分完成安卓固件的烧录。



四、Linux 编译环境搭建

1. 从 Github 获取 Linux SDK 的源码

● Orange Pi Linux 源码下载器

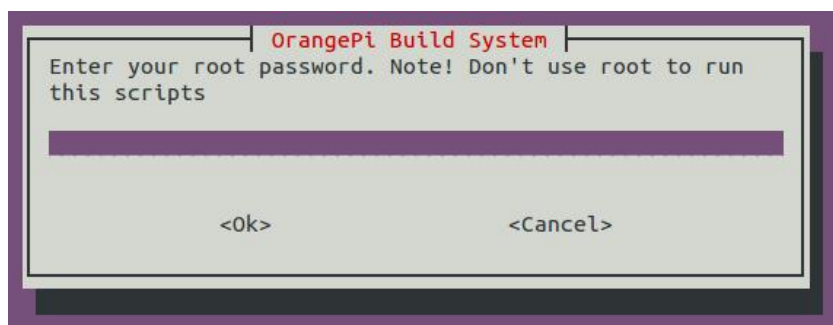
Orange Pi lite2 的 Linux 源码已经上传到 GitHub，目前支持的内核版本为 **Linux 4.9** 和主线内核（部分驱动还在开发中），我们可以使用 Orange Pi Linux 源码专用的下载器进行下载，获取下载器源码的方式如下所示：

```
$ sudo apt-get install git
$ git clone https://github.com/orangepi-xunlong/OrangePi_Build.git
$ cd OrangePi_Build
$ ls
Build_OrangePi.sh  lib  README.md
```

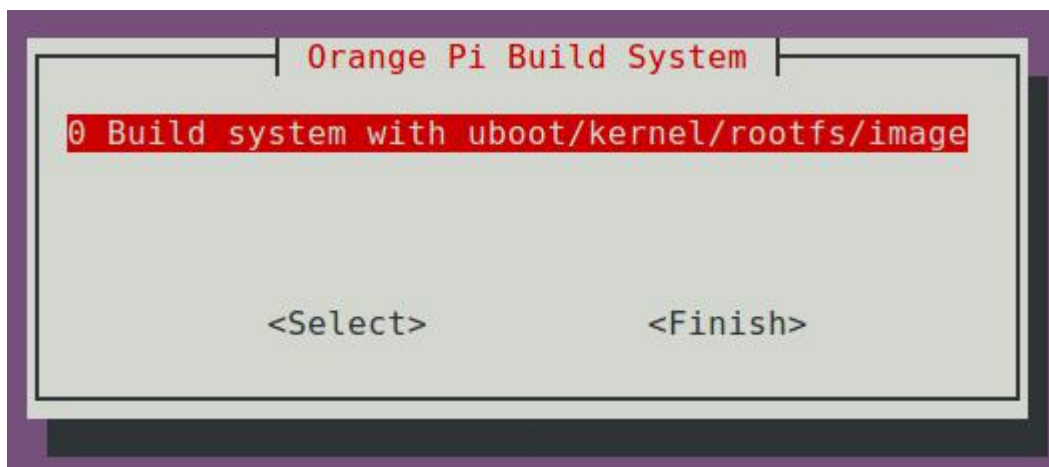
● 运行下载器

```
$ ./Build_OrangePi.sh
```

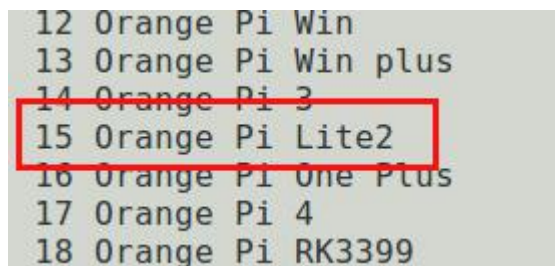
输入 root 密码，然后回车



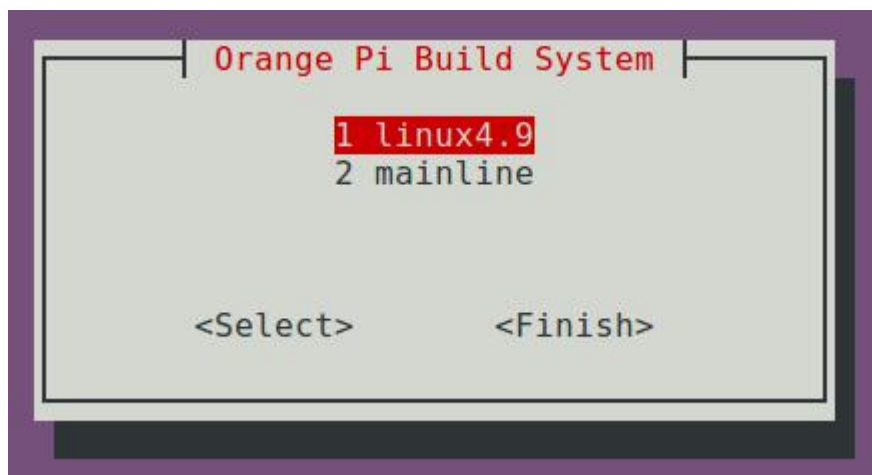
选择 0 Build system with uboot/kernel/rootfs/image 进入开发板型号选择的界面



选择 **orange pi lite2**，回车后进入内核版本的选择界面



目前 orange pi lite2 开发板支持 **Linux 4.9** 和 **mainline** 两个版本的内核代码，选择其中一个后回车就会开始下载对应的 SDK 源码



下载的源码会存放在 OrangePi_Build 的同级目录下

```
$ ls ../OrangePi_Build -l
OrangePi_Build
OrangePiH6_Linux4.9  (其中内核版本为 Linux4.9)
OrangePiH6_mainline  (目前内核版本为 Linux5.3.5)
```



2. 从百度云盘获取 Linux SDK 的源码

如果 GitHub 下载代码失败，可以从百度云盘直接下载 Linux SDK 的源码压缩包。
下载链接为：

https://pan.baidu.com/s/15NOF_eAwbN9ah3dfWx0ArQ

3. 搭建编译环境

Orange Pi H6 Linux SDK 目前只在 **ubuntu 18.04** 的 PC 中测试过。使用前请先准备好 **ubuntu 18.04** 的主机环境。

Orange Pi H6 的 Linux 源码目录结构如下所示：

```
$ cd OrangePiH6_Linux4.9
$ tree -L 1
.
├── build.sh -> scripts/build.s    编译启动脚本
├── external                      存放额外的配置文件
├── kernel                       Linux 内核源码
├── output                       存放输出文件，编译源码后才会生成
├── scripts                      编译过程使用的脚本文件
├── toolchain                    内核和 u-boot 使用的交叉编译工具链
└── uboot                       存放 boot0 和 u-boot 的源码

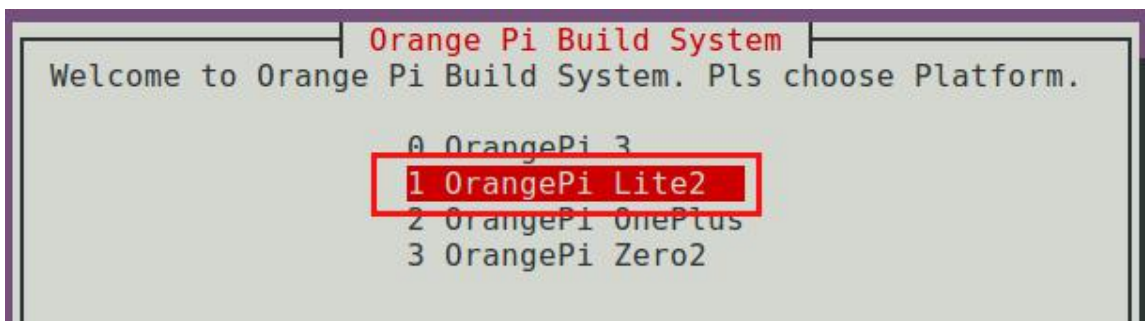
6 directories, 1 file
```

4. 编译 Linux 和 U-boot 源码

● 执行编译启动脚本

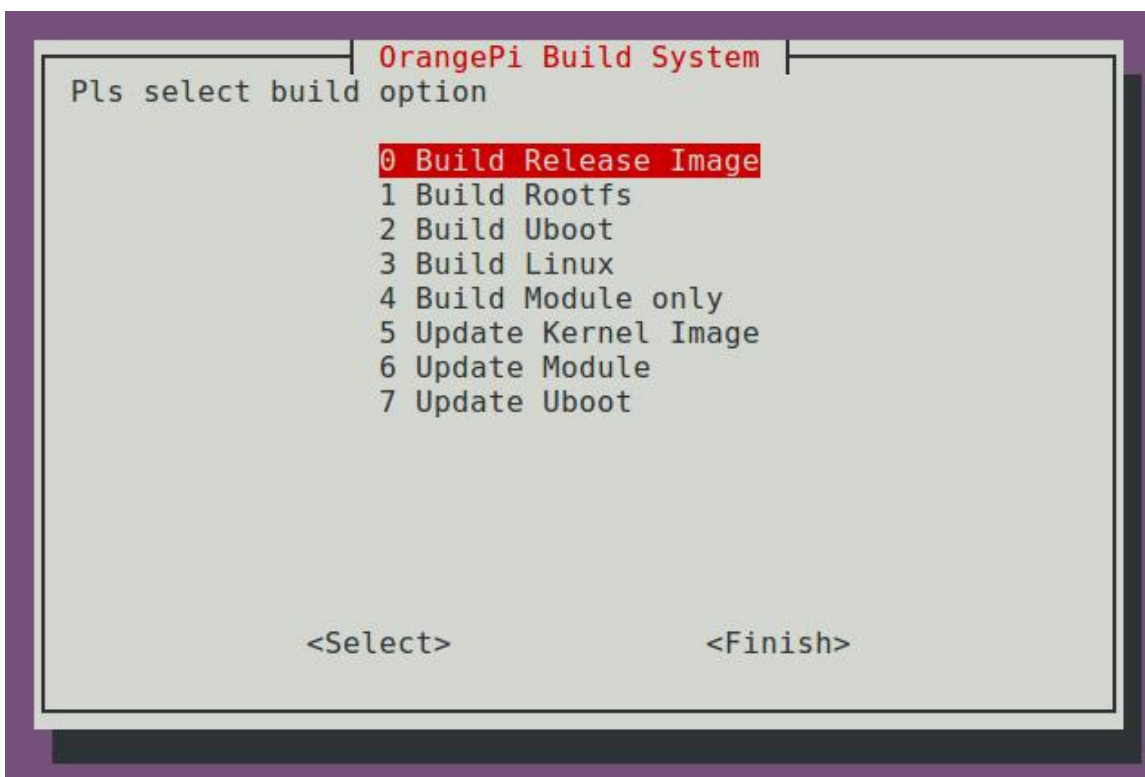
```
$ cd OrangePiH6_Linux4.9
$ sudo ./build.sh
```

选择 **orange pi lite2** 并回车



其中各选项的功能如下：

- 0 Build Release Image —— 编译 ubuntu 或 debian 的发行版镜像
- 1 Build Rootfs —— 编译 ubuntu 或 debian 的 rootfs
- 2 Build Uboot —— 编译 boot0 和 u-boot 的源码
- 3 Build Linux —— 编译 Linux 内核源码
- 4 Build Module only —— 编译 Linux 内核模块
- 5 Update kernel Image —— 更新 SD 卡 Linux 系统中的内核
- 6 Update Module —— 更新 SD 卡 Linux 系统中的内核模块
- 7 Update Uboot —— 更新 SD 卡 Linux 系统的 boot0 和 u-boot 以及 dtb 的配置



编译生成的最终文件会保存在 output 目录下



```
$ cd output
$ tree -L 2
.
├── images                //生成的镜像文件
│   └── OrangePi_3_ubuntu_xenial_server_linux4.9.118_v2.0.2.img
├── kernel                //编译生成的内核
│   └── uImage_3
├── rootfs                //编译生成的 rootfs
├── uboot                 //编译生成的 uboot 镜像
│   ├── boot0_sdcard_sun50iw6pl.bin
│   ├── H6.dtb
│   └── u-boot-sun50iw6pl.bin
└── xenial_arm64_server_rootfs.tar.gz
```

5. Linux SDK 使用示例

下面将以在内核源码中新增一个 **rtl8812AU** USB WIFI 内核模块的方式完整的演示下 Linux SDK 的使用方法。

● 从 github 获取 rtl8812AU 的源码

```
$ cd OrangePiH6_Linux4.9/kernel/drivers/net/wireless
$ git clone https://github.com/diederikdehaas/rtl8812AU.git
Cloning into 'rtl8812AU'...
remote: Counting objects: 2347, done.
Receiving objects: 100% (2347/2347), 7.87 MiB | 22.00 KiB/s, done.
Resolving deltas: 100% (1292/1292), done.
Checking connectivity... done.
```

● 添加 rtl8812AU 的配置

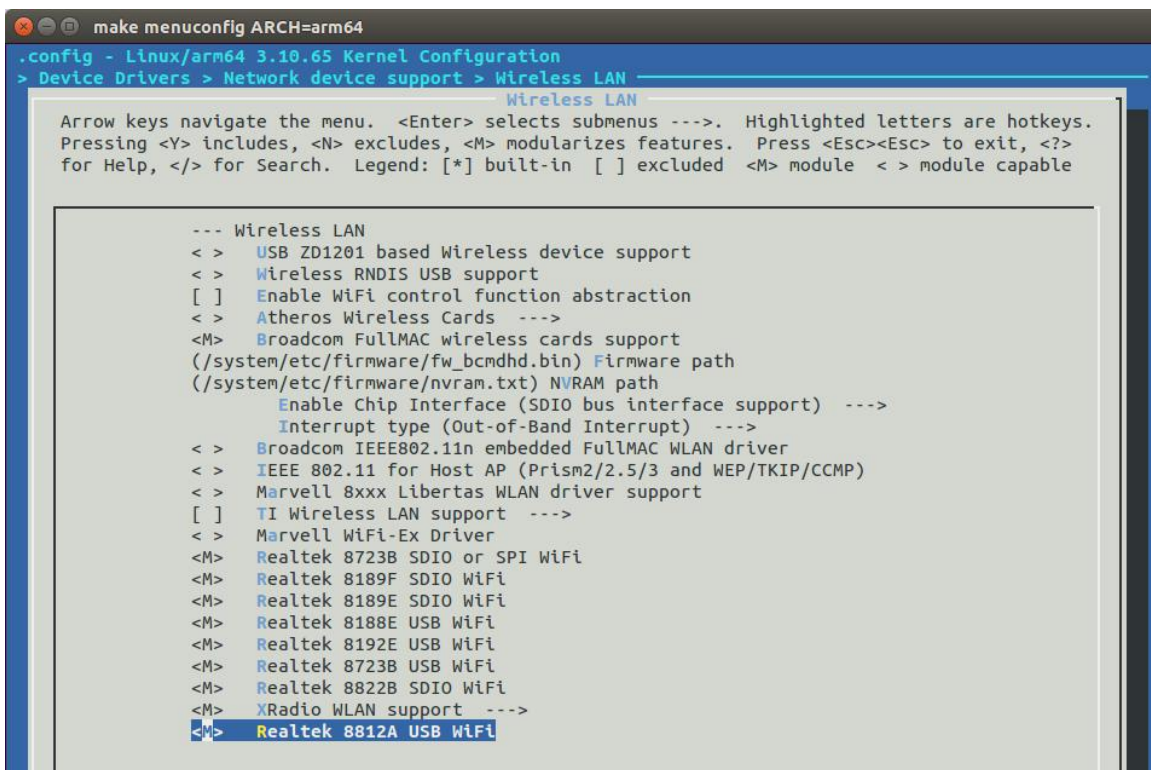
```
$ cd OrangePiH6_Linux4.9/kernel/drivers/net/wireless
$ git diff .
diff --git a/drivers/net/wireless/Kconfig b/drivers/net/wireless/Kconfig
index 373666b..b7ebd5c 100755
--- a/drivers/net/wireless/Kconfig
+++ b/drivers/net/wireless/Kconfig
@@ -294,4 +294,5 @@ source "drivers/net/wireless/rtl8192eu/Kconfig"
```



```
+source "drivers/net/wireless/rtl8812AU/Kconfig"
endif # WLAN

diff --git a/drivers/net/wireless/Makefile
b/drivers/net/wireless/Makefile
index fd8a466..3aef800 100755
--- a/drivers/net/wireless/Makefile
+++ b/drivers/net/wireless/Makefile
@@ -66,3 +66,4 @@ obj-$(CONFIG_XR_WLAN) += xradio/
+obj-$(CONFIG_RTL8812AU)      += rtl8812AU/
```

- 在内核配置中选中 **Realtek 8812A USB WiFi**，并编译成内核模块



- 根据《编译 Linux 和 U-boot 源码》一节的方法重新编译内核



```
0 Build Release Image
1 Build Rootfs
2 Build Uboot
3 Build Linux
4 Build Module only
5 Update Kernel Image
6 Update Module
7 Update Uboot
```

部分编译 Log 如下所示:

```
Start Compile.....
Start Compile Module
CC [M] drivers/net/wireless/rtl8812AU/core/rtw_cmd.o
CC [M] drivers/net/wireless/rtl8812AU/core/rtw_security.o
CC [M] drivers/net/wireless/rtl8812AU/core/rtw_debug.o
CC [M] drivers/net/wireless/rtl8812AU/core/rtw_io.o
CC [M] drivers/net/wireless/rtl8812AU/core/rtw_ioctl_query.o
CC [M] drivers/net/wireless/rtl8812AU/core/rtw_ioctl_set.o
```

编译完后, 可以在 **output/lib/modules/4.9.118+/kernel/drivers/net/wireless/rtl8812AU** 中可以找到编译好的内核模块

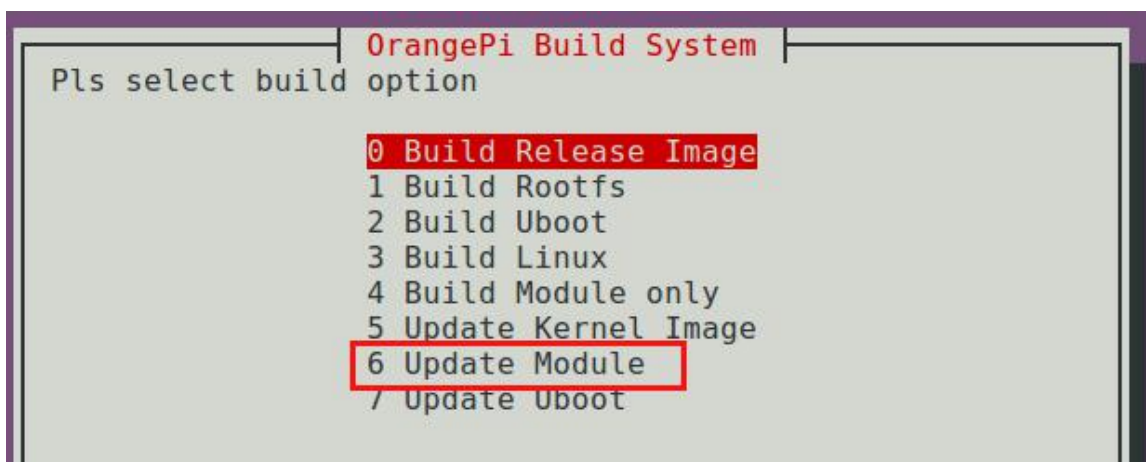
```
$ cd output/lib/modules/4.9.118+/kernel/drivers/net/wireless/rtl8812AU
$ ls
8812au.ko
```

● 更新内核模块

首先将已烧好 Linux 固件的 SD 卡插入用来编译 Linux 源码的 PC 机 (装有 Ubuntu 18.04 的实体机或者虚拟机) 中, 当系统识别并成功挂载插入的 SD 卡后, 我们就可以在 **/media/\$LOGNAME** 下看到对应分区名字

```
$ cd /media/$LOGNAME
$ ls
BOOT      存放内核
rootfs    根文件系统
```

然后根据《编译 Linux 和 U-boot 源码》一节的说明选择 **6 Update Module** 来更新内核模块



接下来就可以通过 SD 卡启动系统，并使用新增的 8812au.ko 内核模块驱动 USB WIFI 网卡



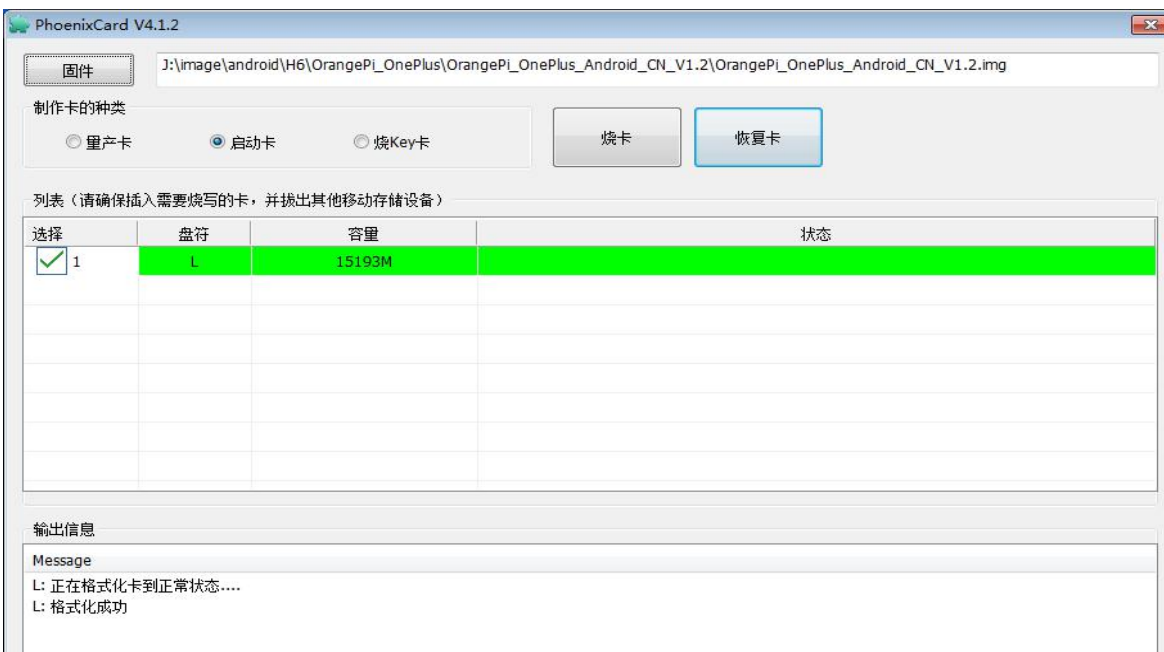
五、Android 固件烧录

Android 固件不能在 Linux 中通过 `dd` 命令或者在 Windows 中通过 Win32 Diskimager 工具来写入 TF 卡。需要使用工具 PhoenixCard 来写入, PhoenixCard 当前的最新版本为 **PhoenixCard V4.1.2**, 可从官网下载页的官方工具中下载。

1. Android 固件烧录步骤

● 格式化 TF 卡

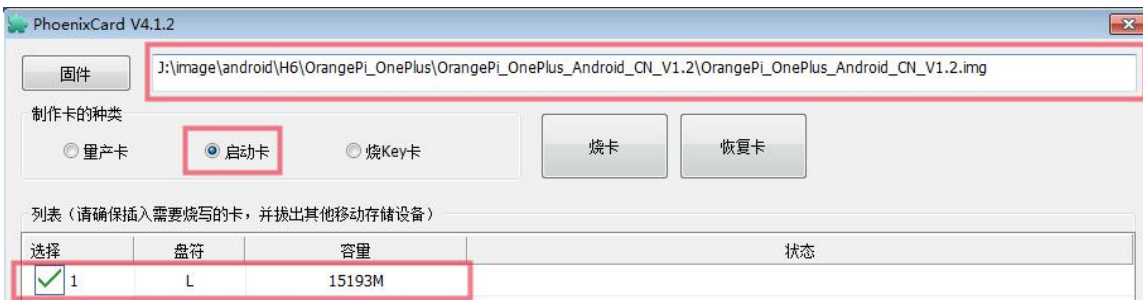
检查插入的 TF 卡是否与选择的盘符一致, 单击“恢复卡”按钮, 开始格式 TF



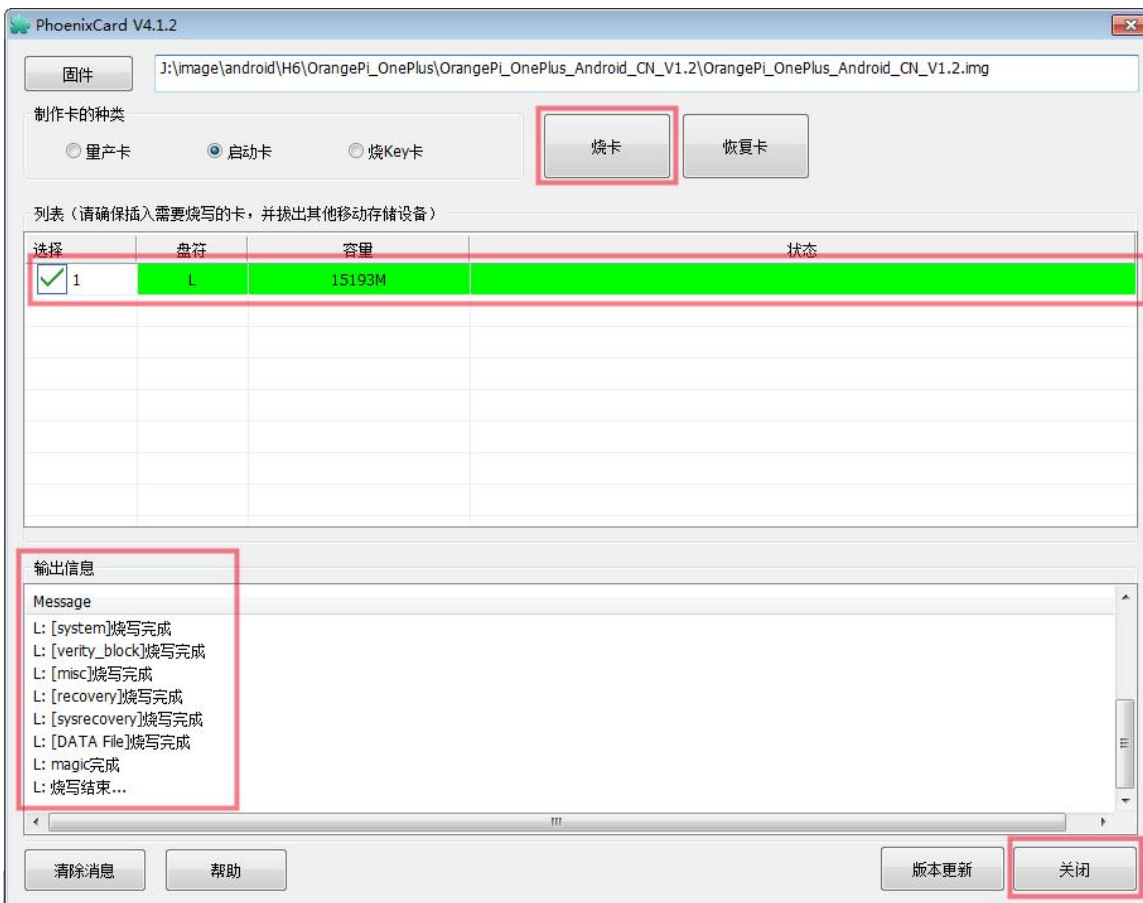


- 然后选择固件，选择启动卡

请注意下图红色标记的地方



- 点击“烧卡”，开始写入 TF 卡，等待烧录完成。



Android 固件成功烧写后，单击“关闭”按钮，然后就可以将 TF 卡插入开发板启动系统



六、Linux 固件烧录

我们可以通过 **Etcher** 将 Orange Pi Lite 2 的 Linux 固件烧录到 TF 卡中，由于 Orange Pi Lite 2 没有板载 eMMC，所以我们只能通过 TF 来启动系统。Etcher 支持如下的操作系统：

- Linux（大多数发行版，如 Ubuntu）
- MacOS 10.9 和后续的版本
- Windows 7 和后续的版本

Etcher 软件安装包可以在其官网 <https://etcher.io/> 中下载，也可以从 Orange Pi Lite 2 的官网下载页的**官方工具**中下载

1. Etcher 的安装方式

- Etcher 在 Windows 系统中的安装方式和普通软件的安装方式一样，这里不再赘述。
- Etcher 在 Ubuntu 和 Debian 系统中的安装方式如下

1. 增加 Etcher Debian 仓库：

```
$ echo "deb https://dl.bintray.com/resin-io/debian stable etcher" | sudo tee /etc/apt/sources.list.d/etcher.list
```

2. 下载 key

```
$ sudo apt-key adv --keyserver hkp://pgp.mit.edu:80 --recv-keys 379CE192D401AB61
```

3. 更新和安装

```
$ sudo apt-get update && sudo apt-get install etcher-electron
```

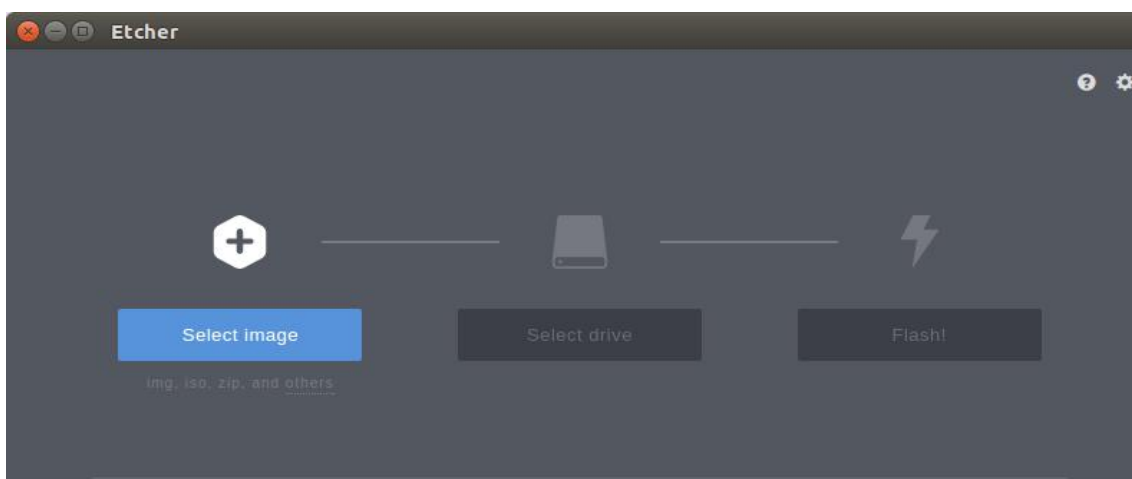



4. 卸载方式

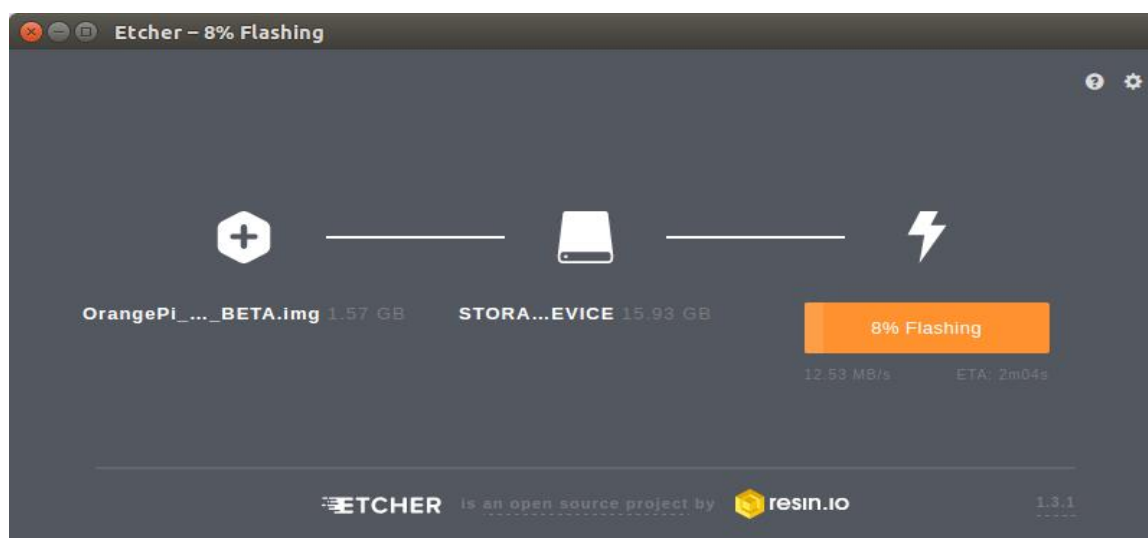
```
$ sudo apt-get remove etcher-electron  
$ sudo rm /etc/apt/sources.list.d/etcher.list && sudo apt-get update
```

2. 通过 Etcher 烧录 Linux 固件的方法

- 首先打开 Etcher，其界面如下图所示



- 然后通过 “Select image” 选择需要烧录的 Linux 固件
- 接着插入 TF 卡，Etcher 会自动识别相应驱动器
- 最后点击 “Flash!” 开始烧录，烧录完后，即可插入开发板启动系统





七、Linux 系统的使用说明

1. Linux 启动亮灯情况说明

- 启动后板载 LED 灯会先亮红灯，然后红灯灭，黄灯长亮

2. 登录账号和密码

- 用户名 root, 密码: orangepi
- 用户名 orangepi, 密码: orangepi

3. 扩展 rootfs 分区

做好系统运行卡之后应立即进行文件系统 rootfs 分区的扩展，这将能大大提升系统的性能，避免空间不足带来的各种繁琐问题

我们可以进入系统后使用系统内置的脚本 `resize_rootfs.sh` 进行扩容：

未扩容前系统可用空间的大小情况

```
root@OrangePi:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  1.1G  520M  488M   52% /
devtmpfs        481M    0  481M    0% /dev
tmpfs           489M    0  489M    0% /dev/shm
tmpfs           489M  6.6M  483M    2% /run
tmpfs           5.0M  4.0K  5.0M    1% /run/lock
tmpfs           489M    0  489M    0% /sys/fs/cgroup
/dev/mmcblk0p1   50M   29M   22M   58% /boot
```

运行系统内置的扩容脚本

```
root@OrangePi:~# resize_rootfs.sh
```

扩容后系统可用空间的大小情况

```
root@OrangePi:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  7.2G  539M  6.4G    8% /
```



devtmpfs	481M	0	481M	0%	/dev
tmpfs	489M	0	489M	0%	/dev/shm
tmpfs	489M	13M	477M	3%	/run
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	489M	0	489M	0%	/sys/fs/cgroup
/dev/mmcblk0p1	50M	29M	22M	58%	/boot

4. 录音放音测试方法

注：主线内核的 **Audio Codec** 音频驱动暂时还不可用

● 测试录音功能

```
root@orangepilite2:~# arecord -d 10 -f cd -D hw:1,0 -t wav Test.wav
Recording WAVE 'Test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@orangepilite2:~#
```

● 测试 HDMI 放音功能，直接使用 **aplay** 命令播放即可

```
root@orangepilite2:~# aplay Test.wav
```

5. WIFI 的配置方法

在 `/etc/network/interface` 中加入下面的配置，然后重启即可

```
auto wlan0
iface wlan0 inet dhcp
wpa-ssid orangepi //此处填入 WIFI 账号（现在是 orangepi）
wpa-psk orangepi //此处填入 WIFI 密码（现在是 orangepi）
```



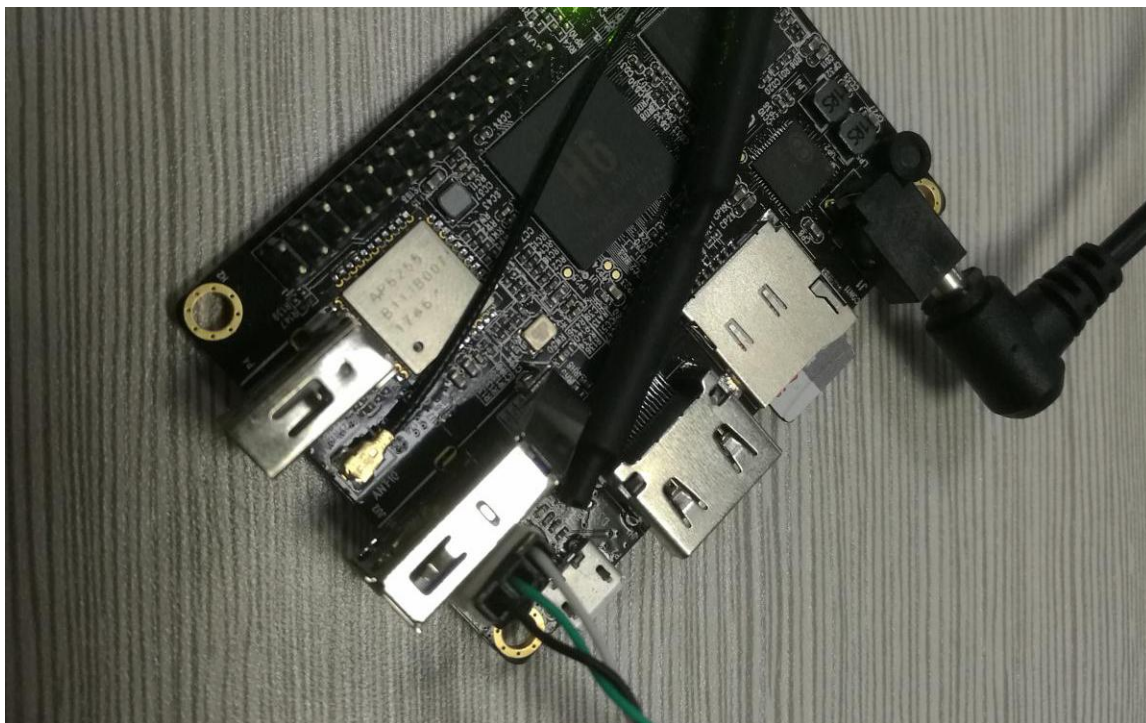
八、串口调试工具介绍

首先需要准备一根和下图类似的 USB 转 TTL 串口线：



按下图接好串口线，从板子的背面的丝印可以看到不同颜色的线对应的功能如下：

- 黑色——GND
- 绿色——RX
- 白色——TX





1. 基于 Windows 平台的使用

在使用 OrangePi 做项目开发过程中，为了获得更多的调试信息，OrangePi 默认支持串口信息调试。对于开发者而言，只需准备上面提到的材料，即可简单的获得串口调试信息。不同的上位机使用的串口调试工具大同小异，基本可以参考下文的方法进行部署。使用 Windows 平台进行串口调试的工具很多，通常使用的工具是 putty。本节以 putty 作为例子进行部署讲解。

● 安装 USB 驱动

下载最新版的驱动 PL2303_Prolific_DriverInstaller_v130.zip，下载解压。

PL2303_Prolific_DriverInstaller_v130	2010/7/15 10:41	应用程序	3,099 KB	← 解压之后的应用程序
PL2303_Prolific_DriverInstaller_v130	2016/8/3 9:20	WinRAR ZIP 压缩...	2,316 KB	← 下载的压缩包
releasenote	2010/7/22 10:14	文本文件	2 KB	

以管理员身份选择应用程序安装



等待安装完成



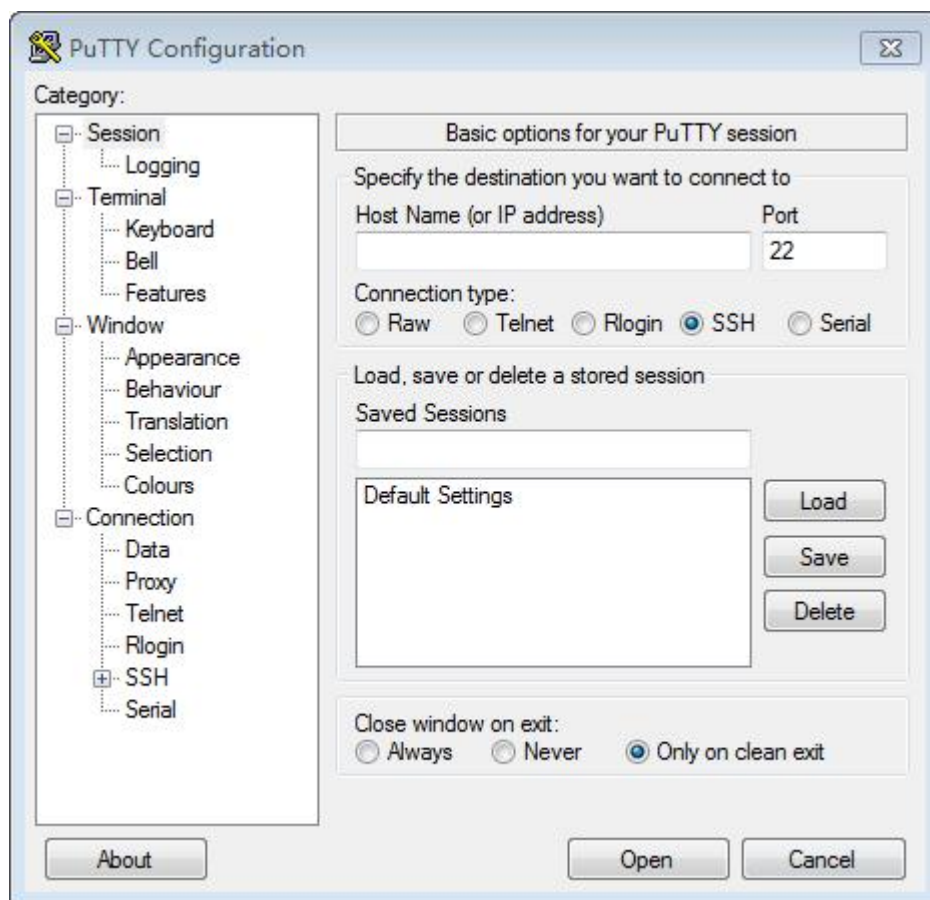


● 下载安装 Putty

Putty 可从下面的地址下载，请选择适合自己开发环境的版本。

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

直接双击下载的 putty.exe 即可打开 putty，软件界面如下图所示。





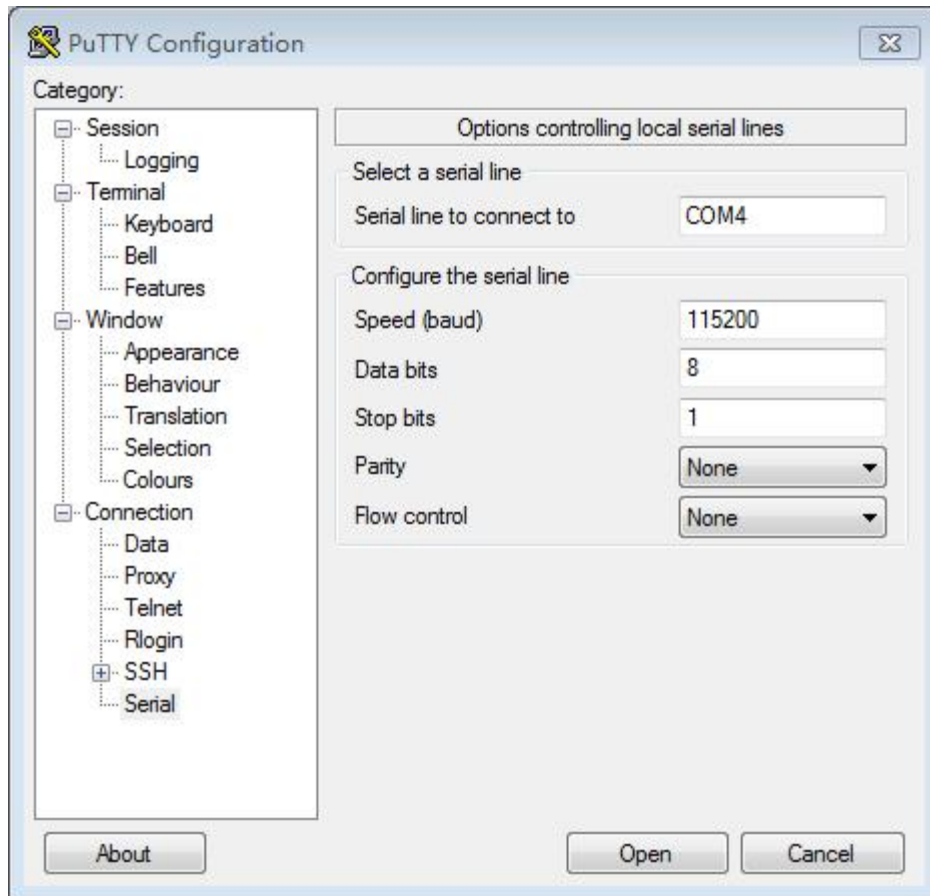
● 设备信息的获取

在 Windows7 中，我们可以通过设备管理器查看串口连接是否正常以及串口的设备号。如果设备没有正常识别，请检查驱动是否安装成功。如果驱动安装有问题，可以尝试使用 360 驱动大师扫描安装驱动。



● Putty 配置

串行口设置成相应的端口号 (COM4)，关闭流控，速度设置成 115200



● 启动调试串口输出

OrangePi 上电开机，putty 将会自动打印串口 log 信息



2. 基于 Linux 平台的使用

在 Linux 平台使用 putty 和 Windows 平台区别不太,下面主要说明有差异地方的操作步骤。所有操作都是基于 Ubuntu 14.04 系统。

● 安装并启动 Putty

```
$ sudo apt-get install putty  
$ sudo putty
```

● 配置 Putty

串口号可以通过 `ls /dev/ttyUSB*` 查看

波特率需要设置为 115200

并且关闭流控

