

内核源码的编译

- 首先需要把交叉编译工具gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar 解压到一个目录，修改内核源码中的set_env.sh脚本，里面的位置应该与交叉编译工具的目录一致。如下图，我是放在/opt/toolchains/目录下。

```
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ cat set_env.sh
export PATH=$PATH:"/opt/toolchains/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu/bin"
export CROSS_COMPILE=aarch64-linux-gnu-
export ARCH=arm64
```

- 每个新打开的终端，在执行编译前，先要执行一次 source set_env.sh配置环境，之后运行 build_kernel.sh脚本即可编译内核和设备树。

```
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ source set_env.sh
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ ./build_kernel.sh
#
# configuration written to .config
#
scripts/kconfig/conf --synccfg Kconfig
CALL    scripts/checksyscalls.sh
CHK     include/generated/compile.h
GZIP    kernel/config_data.gz
LOGO    drivers/video/logo/logo_linux_clut224.c
```

- 编译生成的内核镜像和设备树的位置如下

```
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ find -name "*phytium-pi-board.dtb"
./arch/arm64/boot/dts/phytium/phytium-pi-board.dtb
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ find -name "Image"
./arch/arm64/boot/Image
```

- 编译内核时，会把生成的modules放在如下位置，用户如果调试驱动，生成的ko驱动文件可以从该目录拷贝到飞腾派下的 /lib/modules/4.19.246-phytium-embedded+/下的对应位置

```
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ ls ./modules/lib/modules/4.19.246-phytium-embedded
build          modules.builtin          modules.dep.bin          modules.symbols
kernel        modules.builtin.alias.bin  modules.devname          modules.symbols.bin
modules.alias  modules.builtin.bin       modules.order            source
modules.alias.bin  modules.dep              modules.softdep
```

更新内核/设备树/模块

- 把编译出的内核/设备树替换到飞腾派文件系统的/boot 目录下即可。

以下是我用scp通过网络把 Image 和 phytium-pi-board.dtb 拷贝到飞腾派的过程，拷贝完成后 reboot 即可

```
root@Phytium-Pi:~# scp 86158@192.168.137.1:d:/share/Image /boot
86158@192.168.137.1's password:
Image                               100% 16MB 27.5MB/s 00:00
root@Phytium-Pi:~#
root@Phytium-Pi:~# scp 86158@192.168.137.1:d:/share/phytium-pi-board.dtb /boot
86158@192.168.137.1's password:
phytium-pi-board.dtb                100% 25KB 4.3MB/s 00:00
```

使用该方法，必须保证pc端和树莓派端在同一个网段，并且pc端已经开启了 openssh service 服务。

scp命令简单的解析如下

```
scp 86158@192.168.137.1:d:/share/Image /boot
```

86158为pc的用户名，192.168.137.1为pc的ip地址，d:/share/Image 为需要拷贝的文件名和路径，我这里把需要更新的文件放在pc端d盘share目录下， /boot表示拷贝到飞腾派文件系统的/boot路径下

设备树和 ko模块的操作同理。

- 也可以用其他方法，比如u盘等介质进行拷贝替换。

更换开机logo

首先客户把自己的logo图片(png格式)，命名为logo.png，替换源码下的logo.png文件。具体路径如下图。

之后，运行 ./generate_logo_ppm.sh 脚本，再运行编译内核脚本，使用上面的方法替换内核镜像，重启即可实现。

```
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ find -name "logo.png"
./drivers/video/logo/logo.png
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ ./generate_logo_ppm.sh
pnmcolormap: making histogram...
pnmcolormap: 1472 colors found
pnmcolormap: choosing 224 colors...
pnmremap: 219 colors found in colormap
zero@pc:~/work/temp/phytium-linux-kernel-V1.0.0$ ./build_kernel.sh
#
# configuration written to .config
#
scripts/kconfig/conf --syncconfig Kconfig
```

更新底层固件

- 一般开发不需要更新底层固件
- 选择匹配硬件的固件。例如 fip-all-v2.1hw-4GB-230919.bin，v2.1hw表示硬件版本，在板子背部的丝印上能找到，4GB表示板子的DDR内存大小，230919表示固件日期，也作为版本号识别。
- 取下飞腾派上的sd卡，插入pc端，会在pc端生成 /dev/sdx节点，可能是sdb,sdc...，以下我以sdb为例，使用dd命令更新固件到sd卡上。

```
zero@pc:~/work/temp$ sudo dd if=fip-all-v2.1hw-4GB-230919.bin of=/dev/sdb
[sudo] password for zero:
4545+1 records in
4545+1 records out
2327256 bytes (2.3 MB, 2.2 MiB) copied, 0.384609 s, 6.1 MB/s
zero@pc:~/work/temp$ sync
```

dd 命令简单解析如下

```
sudo dd if=fip-all-v2.1hw-4GB-230919.bin of=/dev/sdb
```

if=后面为需要写入的固件路径和名称 of=后面为需要写入的sd卡节点sdx，根据具体情况而定

写入之后运行 sync 指令，同步数据。

之后拔下sd卡，插入飞腾派板子上，上电即可。