

# 飞腾派 OpenHarmony 用户指导手册 (V1.0)

### 飞腾信息技术有限公司

www.phytium.com.cn

2023 年 08 月 24 日

www.phytium.com.cn

版权所有 © 飞腾信息技术有限公司 2023。保留一切权利。 未经本公司同意,任何单位、公司或个人不得擅自复制、翻译、摘抄本文档内容 的部分或全部,不得以任何方式或途径进行传播和宣传。

#### 商标声明

Phytium 和其他飞腾商标均为飞腾信息技术有限公司的商标。 本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

#### 特别提示

本文档仅作为使用指导,飞腾对本文档内容不做任何明示或暗示的声明或保证。本文档中的所有陈述、 信息和建议不构成任何明示或暗示的担保。 由于产品版本升级或其他原因,本文档内容会不定期进行更新,如有变更,恕不另行通知。

#### 最新技术资源

飞腾嵌入式软件开源社区	https://gitee.com/phytium_embedded
或者访问飞腾软件开发者平台	https://service.phytium.com.cn/developer/

飞腾信息技术有限公司

www.phytium.com.cn

1 简介	3
1.1 文档目的	3
1.2 飞腾派芯片简介	3
1.3 OpenHarmony 简介	3
1.4 Phytiumpi OpenHarmony 开源仓库简介	3
2 搭建开发环境	. 4
2.1 交叉编译硬件环境	4
2.2 下载 repo 脚本文件	4
2.3 获取 OpenHarmony 标准系统源码	4
2.4 获取编译工具链	4
2.5 执行 prebuilts	5
3 代码下载与整合	5
3.1 下载 phytium OH_3.2_release phytiumpi-firefly 适配源码	5
3.2 整合 phytiym device vendor 相关源码	5
4 代码编译	. 6
4.1 OpenHarmony 编译	6
5 镜像烧写	. 6
5.1 SD 卡分区	7
5.2 system/vendor/userdata 分区	7
5.3 boot 分区	7
5.3.1 Uboot 启动	. 7
5.4 将 Uboot 镜像烧写到 SD 卡	8
6 设备启动	. 8
6.1 Uboot 启动	8
7 维护者邮箱	9
A. 更新记录	9

飞腾信息技术有限公司

www.phytium.com.cn

### 1 简介

#### 1.1 文档目的

该文档主要介绍,如何在飞腾信息技术有限公司的 phytiumpi-firefly 开发板平台上运行 OpenHarmony标准系统。

#### 1.2 飞腾派芯片简介

飞腾派是飞腾嵌入式四核处理器,该处理器兼容ARM V8指令集,包含2个FTC664核和2个FTC310核,其中FTC664核主频可达1.8GHz,FTC310核主频可达1.5GHz。

#### 1.3 OpenHarmony 简介

OpenHarmony 是由开放原子开源基金会(OpenAtom Foundation)孵化及运营的开源项目,目标 是面向全场景、全连接、全智能时代,搭建一个智能终端设备操作系统的框架和平台,促进万物互联产业 的繁荣发展。OpenHarmony 支持轻量系统、小型系统、标准系统三种类型系统。OpenHarmony 相关具 体内容可以参考 <u>OpenHarmony 官网</u>以及 <u>OpenHarmony 码云仓库</u>介绍。

#### 1.4 Phytiumpi OpenHarmony 开源仓库简介

Phytium 在飞腾派上对 OpenHarmony 3.2Release 标准系统已完成适配,项目代码已在码云 Gitee 上开源,开源仓库地址: Phytium-OpenHarmony-Phytiumpi。

该项目基于 phytiumpi-firefly 开发板开发,支持 Linux kernel 5.10,并已使能飞腾派内置提供的视频 解码硬件加速。

#### 代码仓库主要目录说明如下:

├── devic	e_board_phytium	#飞腾开发板代码仓库	

- device\_soc\_phytium #飞腾芯片代码仓库
- phytium\_env.sh #自动化合 patch 脚本
- vendor\_phytium #飞腾开发板方案定制代码仓库

www.phytium.com.cn

### 2 搭建开发环境

#### 2.1 交叉编译硬件环境

工欲善其事必先利其器,所以首先准备一台装有 ubuntu20.04 系统 X86 主机,内存最低配置要求 16G。 硬件配置肯定是多多益善了,配置越高,编译过程就更加的纵享丝滑了!

#### 2.2 下载 repo 脚本文件

- 1. 注册码云 gitee 账号。
- 2. 注册码云 SSH 公钥,请参考<u>码云帮助中心</u>。
- 3. 安装 git 客户端和 git-lfs 并配置用户信息。

git config --global user.name "yourname" git config --global user.email "your-email-address"

git config --global credential.helper store

 安装码云 repo 工具,可以执行如下命令。
 curl -s https://gitee.com/oschina/repo/raw/fork\_flow/repo-py3 > /usr/local/bin/repo chmod a+x /usr/local/bin/repo pip3 install -i https://repo.huaweicloud.com/repository/pypi/simple requests

#### 2.3 获取 OpenHarmony 标准系统源码

推荐版本 OpenHamony 3.2 release。

通过 repo + ssh 下载(需注册公钥,请参考码云帮助中心)。 export WORK\_SPACE=/home/xxx/OpenHarmony #替换成自己定义的 workspace 路径 export PROJ\_ROOT=\$WORK\_SPACE/3.2-release mkdir \$WORK\_SPACE mkdir \$PROJ\_ROOT cd \$PROJ\_ROOT repo init -u git@gitee.com:openharmony/manifest.git -b OpenHarmony-3.2-Release --no-repo-verify repo sync -c repo forall -c 'git lfs pull'

#### 2.4 获取编译工具链

使用安装包方式获取编译工具链。

飞腾信息技术有限公司

www.phytium.com.cn

sudo apt-get update && sudo apt-get install binutils git git-lfs 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 m4 bc gnutls-bin python3.8 python3-pip ruby

#### 2.5 执行 prebuilts

在源码根目录下执行脚本,安装编译器及二进制工具。 cd &PROJ\_ROOT bash build/prebuilts\_download.sh

### 3 代码下载与整合

#### 3.1 下载 phytium OH\_3.2\_release phytiumpi-firefly 适配源码

创建存放 phytium phytium OH\_3.2\_release 源码的目录,进到该目录,下载 phytium

OH\_3.2\_release phytiumpi-firefly 适配源码: export PHY\_DEV=\$WORK\_SPACE/phytium\_device

mkdir \$PHY\_DEV cd \$PHY\_DEV git clone https://gitee.com/phytium\_embedded/phytium-openharmony-phytiumpi-firefly-device.git

#### 3.2 整合 phytiym device vendor 相关源码

通过前面的步骤,我们下载好了相关飞腾部分的修改。是时候将其和入到鸿蒙主干了。这里我们贴心

的为开发者准备了一键构建脚本,我们这里执行脚本。

./phytium\_env.sh \$PROJ\_ROOT THE OHOS\_PATH\_ROOT /home/xxx/OpenHarmony/3.2-release #### sync phytium env start! #### #### sync device\_soc\_phytium #### #### sync device\_soc\_phytium end #### #### sync device\_board\_phytium end #### #### sync vendor\_phytium #### #### sync vendor\_phytium end #### #### sync vendor\_phytium end #### #### sync kernel #### #### sync kernel end #### #### applay 0001\_drivers\_peripheral.patch ####

飞腾信息技术有限公司

www.phytium.com.cn

#### applay 0001\_foundation\_arkui\_ace\_engine.patch ####
#### applay 0001\_foundation\_graphic\_graphic\_2d.patch ####
##### sync phytium end! ####

注意: 这里我们需要传入的参数是鸿蒙的根目录!

### 4 代码编译

#### 4.1 OpenHarmony 编译

```
cd $PROJ_ROOT
```

./build.sh --product-name phytiumpi-firefly --ccache --target-cpu arm64

编译成功提示:

post\_process

=====build phytiumpi-firefly successful.

编译完成,我们进入到 out 目录查看生成的文件:

cd out/phytiumpi-firefly/packages/phone/images/

tree

kernel

dtb

phytiumpi\_firefly.dtb

BOOT

BOOT

phytiumpi\_firefly.dtb

ph

可以看到,其中生成了内核和系统相关镜像文件!

### 5 镜像烧写

飞腾派开发板支持从 flash、SD 卡、EMMC 三种介质启动,通过拨码控制,具体设置方法请参考对 应的飞腾派开发板使用手册。OpenHarmony 仓库代码默认是从 SD 卡启动,这里分区和烧录以 SD 卡举 例。 修改成其他介质启动,需要修改代码,参考《OpenHarmony 多种存储介质配置说明》。

www.phytium.com.cn

#### 5.1 SD 卡分区

为了兼容从 SD 卡启动,前面会预留 64M 空间用于存放 uboot 固件,第一个分区的起始块偏移为 131072。由于不是从首地址开始分区,分区时都要手动指定起始地址,第一个分区地址地址为 131072, 后面分区起始地址是前一个分区结束地址基础上再加 1。

在 linux 环境下,使用 fdsik 命令对 SD 卡进行分区,创建 4 个分区,依次为 boot、system、vendor、 userdata,根据实际情况设定,比如可以设定为 500M、2G、500M、2G,分区参数可参考如下配置。

Device	Boot	Start	End	Sectors	Size Id Type
/dev/sdx1	1	31072	1155071	1024000	500M 83 Linux
/dev/sdx2	11	55072	534937	5 4194304	1 2G 83 Linux
/dev/sdx3	53	349376	637337	5 102400	0 500M 83 Linux
/dev/sdx4	63	373376	10567679	9 4194304	a 2G 83 Linux

注意: fidsk 命令详细使用方法可自行百度谷歌,或者参考飞腾嵌入式 LINUX 用户 手册中的磁盘分 区部分内容。

#### 5.2 system/vendor/userdata 分区

将这三个分区并格式化为 ext4, sdX 中的 X 烧写之前先用 df 命令确认一下是多少, 可能是 b, c…等,

#### 这里一定要注意不要烧错。

sudo mkfs.ext4 sdx2 sudo mkfs.ext4 sdx3 sudo mkfs.ext4 sdx4

使用 dd 命令将 4.1 OpenHarmony 编译章节编译生成的镜像文件烧写到对应分区中。

sudo dd if=system.img of=/dev/sdX2 bs=1M

sudo dd if=vendor.img of=/dev/sdX3 bs=1M

重要的事情说三遍,不要烧录 userdata 分区!只需要将 sdx4 格式化为 ext4 格式就 OK 了!

#### 5.3 boot 分区

boot 分区的烧录,启动方式不一样,烧录方法不一样,区分为 Uboot 启动和 UEFI 启动两种,推荐使用 Uboot 启动。

#### 5.3.1 Uboot 启动

1. 将第一个分区格式化为 ext4

飞腾信息技术有限公司

www.phytium.com.cn

sudo mkfs.ext4 sdx1

2. 挂载 boot 分区 mkdir ~/disk sudo mount /dev/sdx1 ~/disk

3. 将前面生成的 Kernel 镜像,设备树文件,以及 ramdisk 拷贝到 boot 分区。

sudo cp &PROJ\_ROOT/out/phytiumpi-firefly/packages/phone/images/kernel/Image ~/disk/

sudo cp &PROJ\_ROOT/out/phytiumpi-firefly/packages/phone/images/kernel/dtb/phytiumpi\_firefly.dtb ~/disk/

sudo cp &PROJ\_ROOT/out/phytiumpi-firefly/packages/phone/images/ramdisk.img ~/disk/ sync

4. 卸载 boot 分区

sudo umount ~/disk

#### 5.4 将 Uboot 镜像烧写到 SD 卡

如果板子不是设成从 SD 卡启动,这小节可以跳过。

当板子设成从 SD 卡启动时,首先将分区表信息保存为 table.bin,然后将从 FAE 获取到的 Uboot 镜

像<uboot.bin>烧写到 SD 卡,最后将 table.bin 再重新写回 SD 卡。 sudo dd if=/dev/sdx of=table.bin bs=1 skip=446 count=66 sudo dd if=<uboot.bin> of=/dev/sdx sudo dd if=table.bin of=/dev/sdx bs=1 seek=446 count=66

### 6 设备启动

将烧写好的 SD 卡插入 SD 卡槽,板子配置成合适的启动介质,连接串口线,连接到 phytiumpi-firefly 开发板上。上位机的串口调试工具波特率设置为 115200,上电开机。

#### 6.1 Uboot 启动

第一次开机,上电后,立即按回车,进入 uboot 后,按如下配置设置并保存 uboot 环境变量,最后输入 boot 进入系统。之后开机不需要再设置,会按这里保存的环境变量自动进入系统。

setenv bootargs console=ttyAMA1,115200 earlycon=p1011,0x2800d000 root=/dev/ram0 elevator=deadline rootwait rw loglevel=6 hardware=phytiumpi rootfstype=ext4 initrd=0x93000000,90M cma=256M setenv bootcmd "ext4load mmc 0 0x90100000 phytiumpi\_firefly.dtb;ext4load mmc 0 0x90200000 Image;ext4load mmc 0 0x93000000 ramdisk.img;booti 0x90200000 -

0×90100000"

saveenv

## Phytium飞腾 -

飞腾信息技术有限公司

www.phytium.com.cn

boot

### 7 维护者邮箱

飞腾信息技术有限公司

zhangjianwei@phytium.com.cn

tangkaiwen@phytium.com.cn

xiayan1086@phytium.com.cn

libowen1180@phytium.com.cn

chenzigui1762@phytium.com.cn

### A. 更新记录

发布日期	作者	
2023-08-24	夏炎	初稿