



openEuler

系统编译运行指南

文档版本 00B01

发布日期 2023-10-31

版权所有 © 海思技术有限公司2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为海思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。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

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。除非另有约定，本文档仅作为使用指导，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。

海思技术有限公司

地址：上海市青浦区虹桥港路2号101室 邮编：201721

网址：<https://www.hisilicon.com/cn/>

客户服务邮箱：support@hisilicon.com



前言

概述

本文档为使用海思Hi3093芯片进行二次开发的程序员而写，目的是供您在开发过程中查阅openEuler编译及升级方法。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3093	-

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。



符号	说明
须知	用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

日期	版本	修订说明
2023-10-31	00B01	第1次临时版本发布。



目录

前言.....	i
1 eMMC 镜像构建.....	1
1.1 概述.....	1
1.2 制作 eMMC 镜像.....	1
1.3 分模块编译参考.....	2
1.3.1 源码和编译环境准备.....	2
1.3.1.1 源码准备.....	2
1.3.1.2 编译环境准备.....	2
1.3.2 编译 Hi3093 SDK.....	3
1.3.2.1 Hi3093 SDK 编译准备.....	3
1.3.2.2 Hi3093 SDK 编译.....	3
1.3.3 uboot 制作.....	4
1.3.4 制作 uImage.....	4
1.3.5 文件系统的制作.....	4
2 eMMC 升级和启动.....	6
2.1 SFC 升级 eMMC 流程.....	6
2.2 eMMC 升级 eMMC 流程.....	7
2.3 eMMC 启动流程.....	8
3 SFC 镜像构建.....	9
4 SFC 启动混合部署与 Baremetal 方案.....	10
4.1 SFC 启动 openEuler 和混合部署系统方案.....	10
4.2 SFC 启动 Baremetal 方案.....	11



插图目录

图 2-1 PC 端 IP 配置参考图.....	7
图 2-2 文件传输示意图.....	7



1 eMMC 镜像构建

1.1 概述

eMMC镜像构建描述了生成eMMC镜像升级所需要的Hi3093_ext4fs.img、Hi3093_ext4fs_cms.bin和u-boot_rsa_4096.bin的方法。

由于openEuler Embedded构建过程需要基于openEuler操作系统，且需要安装较多系统工具和构建工具。为方便开发人员快速搭建构建环境，我们将构建过程所依赖的操作系统和工具封装到一个容器中，这就使得开发人员可以快速搭建一个构建环境，进而投入到代码开发中去，避免在准备环境阶段消耗大量时间。

参考文档：[容器环境下的快速构建指导 — openEuler Embedded在线文档 1.0.0 文档\(gitee.io\)](#)。

1.2 制作 eMMC 镜像

步骤1 下载release代码包mpu_solution.tgz并解压。

步骤2 将[1.1 概述](#)docker容器里面路径/usr1/openeuler/src/yocto-embedded-tools/config/arm64目录下的defconfig-kernel和编译生成的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz、zImage拷贝到mpu_solution/custom/open_euler目录下。

步骤3 执行如下编译脚本：

```
cd mpu_solution/  
./build.sh
```

出现编译选项提示后，输入oe或直接按回车（默认编译选项是openEuler）

步骤4 编译生成的文件在mpu_solution/build/build_hi3093/output目录下。

- **u-boot_rsa_4096.bin**: uboot。
- **Hi3093_ext4fs_cms.bin**: 文件系统签名。
- **Hi3093_ext4fs.img**: 文件系统，里面包含ulImage和设备树。

----结束



⚠ 注意

本解决方案不包括openEuler系统，为提升用户构建效率，一键构建脚本中会通过网络下载openEuler-22.03-LTS-SP1版本，用户可通过修改build.sh指定openEuler版本，openEuler系统部分的网络安全问题由用户负责。

须知

- 需要使用有sudo权限的账号去执行build.sh脚本。
- 执行build.sh过程中可能会让输入账户密码。
- 脚本会创建~/hi3093_tool目录，里面存放编译所依赖的工具和编译链。
- 如果有额外需要打包到二级文件系统的ko、bin、so、脚本等文件，需要将文件放到mpu_solution/custom/usr对应目录，然后再执行编译脚本。
- 如果有需要在开机启动时执行的脚本，将对应脚本文件放到mpu_solution/custom目录下，并命名为hi3093_product.sh，然后再执行编译脚本，系统启动时将会自动调用此脚本。
- mpu_solution/build/build_fs/cpio_fs保存了文件系统压缩前的文件及目录结构。
- 如果需要重复执行build.sh，需先执行clean.sh清除上一次的编译中间产物。

1.3 分模块编译参考

1.3.1 源码和编译环境准备

1.3.1.1 源码准备

步骤1 下载release代码包mpu_solution.tgz并解压。

步骤2 将1.1 概述docker容器里面路径/usr1/openeuler/src/yocto-embedded-tools/config/arm64目录下的defconfig-kernel和编译生成的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz、zImage拷贝到mpu_solution/custom/open_euler目录下。

----结束

1.3.1.2 编译环境准备

步骤1 从以下路径下载openEuler-22.03-LTS-SP1交叉编译链到~/hi3093_tool目录：

http://121.36.84.172/dailybuild/EBS-openEuler-22.03-LTS-SP1/embedded_img/arm64/aarch64-std/openeuler-glibc-x86_64-openeuler-image-aarch64-qemu-aarch64-toolchain-22.03.sh

步骤2 执行如下命令。

```
sh openeuler-glibc-x86_64-openeuler-image-aarch64-qemu-aarch64-toolchain-22.03.sh -d /home/${USER}/hi3093_tool/toolchain/ -y \
```

步骤3 配置openEuler交叉编译环境，执行如下命令。



```
./home/${USER}/hi3093_tool/toolchain/environment-setup-aarch64-openeuler-linux
```

----结束

须知

路径前是“点(.) 空格()”，点和空格都不能少。

1.3.2 编译 Hi3093 SDK

1.3.2.1 Hi3093 SDK 编译准备

步骤1 修改编译脚本权限。

```
cd mpu_solution/  
chmod u+x -R build
```

步骤2 从社区下载openEuler-22.03-LTS-SP1对应的内核放到mpu_solution/open_source目录下，并将目录名重命名成linux5.10，详细命令如下。

```
cd mpu_solution/open_source  
wget --no-check-certificate https://gitee.com/openeuler/kernel/repository/archive/openEuler-22.03-LTS-SP1.zip  
unzip openEuler-22.03-LTS-SP1.zip  
mv kernel-openEuler-22.03-LTS-SP1/ linux5.10
```

步骤3 配置编译时所需要的环境。

```
cd mpu_solution/build  
source build_prepare.sh  
source build_exports
```

步骤4 编译内核。

1. 更换内核默认配置。

```
cd mpu_solution  
cp custom/open_euler/defconfig-kernel open_source/linux5.10/arch/arm64/configs/defconfig
```

2. 打patch。

```
cd mpu_solution/open_source/linux5.10  
git apply ../../src/patches/openEuler/kernel-22.03-lts-sp1-mmc.patch
```

3. 编译内核。

```
cd mpu_solution/open_source/linux5.10  
make ARCH=arm64 defconfig  
make ARCH=arm64 CROSS_COMPILE=aarch64-openeuler-linux-gnu- -j32
```

----结束

1.3.2.2 Hi3093 SDK 编译

执行如下命令编译SDK：

```
cd mpu_solution/build/build_sdk/  
./build_sdk.sh
```

SDK编译完成后会打包生成两个文件：hi3093sdk.tar.gz，securec.tar.gz存放在mpu_solution/build/output目录下。



1.3.3 uboot 制作

步骤1 编译生成u-boot.bin。

```
cd mpu_solution/build/build_uboot  
./build_uboot.sh
```

生成的u-boot.bin在mpu_solution/build/output目录下

步骤2 将编译完成的u-boot.bin拷贝至build/build_sign/下。

```
cp mpu_solution/build/output/u-boot.bin mpu_solution/build/build_sign/
```

步骤3 创建rsacert.cer文件并在文件中输入任意一段字符，切记不能为空证书。

```
cd mpu_solution/build/build_sign/  
echo "hi3093" >> rsacert.cer
```

步骤4 执行如下命令生成u-boot_rsa_4096.bin。

```
./prepare_code_sign_data u-boot.bin  
./generate_sign_image u-boot_rsa_4096.cfg
```

命令执行完成后，在mpu_solution/build/build_sign目录下生成u-boot_rsa_4096.bin文件。

----结束

1.3.4 制作 ulmage

描述：用于openEuler的镜像的转换，zImage转换成ulmage。步骤如下：

步骤1 执行1.3.3 uboot制作的步骤后，将在mpu_solution/custom/open_euler目录生成mkimage可执行文件。

步骤2 将openEuler编译出的zImage制作成ulmage，执行如下命令。

```
cd mpu_solution/custom/open_euler  
./mkimage -A arm -O linux -T kernel -C none -a 0x98280000 -e 0x98280040 -n linux-5.10.0 -d zImage  
ulmage \
```

步骤3 执行完成后，在mpu_solution/custom/open_euler目录下生成ulmage文件。

----结束

1.3.5 文件系统的制作

步骤1 将/mpu_solution/custom/open_euler目录下的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz和1.3.4 制作ulmage生成的ulmage放到mpu_solution/build/output目录下。

步骤2 将mpu_solution/build/output目录下的openeuler-image-qemu-aarch64-*.rootfs.cpio.gz修改为initrd_boot.cpio.gz。

命令如下。

```
mv openeuler-image-qemu-aarch64-*.rootfs.cpio.gz initrd_boot.cpio.gz
```

步骤3 编译dtb。

```
cd mpu_solution/build/build_rtos  
dtc -I dts -O dtb -o hi1711_asic.dtb hi3093.dts
```

命令执行完成后，在mpu_solution/build/build_rtos生成hi1711_asic.dtb文件。

步骤4 进入mpu_solution/build/build_fs目录下，使用有sudo权限的账号执行如下命令：



```
./build_hi3093_ext4.sh
```

命令执行完成后，在mpu_solution/build/output目录下生成hi3093_ext4.tar.gz文件（压缩了Hi3093_ext4fs.cms.bin和Hi3093_ext4fs.img文件），在mpu_solution/build/build_fs/cpio_fs保存了文件系统压缩前的文件及目录结构。

- 步骤5** 如果想添加文件到文件系统中，修改./build_hi3093_ext4.sh中的update_cpio_for_ext4接口，添加将文件拷贝到对应目录命令（“\$TEMP_CPIO_UNPACKER_DIR”为单板起来后的根目录“/”），重新制作文件系统即可。
- 步骤6** 如果有额外需要打包到二级文件系统的ko、bin、so、脚本等文件，需要将文件放到mpu_solution/custom/usr对应目录，重新制作文件系统即可。
- 步骤7** 如果有需要在开机启动时执行的脚本，将对应脚本文件放到mpu_solution/custom目录下，并命名为hi3093_product.sh，重新制作文件系统，系统启动时将会自动调用此脚本。

----结束

须知

没有sudo权限制作出来的文件系统不可用。



2 eMMC 升级和启动

2.1 SFC 升级 eMMC 流程

步骤1 将SFC颗粒放入3093单板U12对应位置的底座中，注意颗粒1脚位置，需要将颗粒上标有小圆点的角对准单板上的小圆点，否则会引起SFC短路烧毁。

步骤2 检查拨码开关SW1，以on侧为1，记SW1 1~4为1110，即1~3开关拨至on侧，4开关拨至对侧。

步骤3 接通电源，串口接J26，网线接J12，打开电源开关。

步骤4 执行以下命令进行emmc gpp分区，默认为0-2分区384M，可在emmc_divide后加数字单位是MB，例emmc_divide 384。

```
emmc_divide
```

步骤5 单板上电并执行以下命令进行配网。

```
insmod /lib/net/gmac_drv.ko  
ifconfig eth2 192.168.0.11
```

步骤6 使用MobaXterm将第1.2 制作eMMC镜像中制作完成的Hi3093_ext4fs.img、Hi3093_ext4fs_cms.bin和u-boot_rsa_4096.bin和l0fw_rsa_4096.bin、l1fw_rsa_4096.bin传输至板端/tmp文件夹下。

步骤7 进入/tmp目录执行以下命令，执行成功后emmc镜像即已烧写成功，参考2.3 eMMC启动流程步骤1~2切换至emmc启动即可。

```
echo 0 > /sys/block/mmcblk0boot0/force_ro  
dd if=l0fw_rsa_4096.bin of=/dev/mmcblk0boot0 bs=1k count=256  
dd if=l1fw_rsa_4096.bin of=/dev/mmcblk0boot0 bs=1k count=512 seek=256  
dd if=u-boot_rsa_4096.bin of=/dev/mmcblk0boot0 bs=1k count=512 seek=768
```

```
echo 0 > /sys/block/mmcblk0boot1/force_ro  
dd if=l0fw_rsa_4096.bin of=/dev/mmcblk0boot1 bs=1k count=256  
dd if=l1fw_rsa_4096.bin of=/dev/mmcblk0boot1 bs=1k count=512 seek=256  
dd if=u-boot_rsa_4096.bin of=/dev/mmcblk0boot1 bs=1k count=512 seek=768
```

```
dd if=/dev/zero of=/dev/mmcblk0gp0 bs=1k count=393216  
dd if=Hi3093_ext4fs.img of=/dev/mmcblk0gp0 bs=1k  
dd if=Hi3093_ext4fs_cms.bin of=/dev/mmcblk0gp0 bs=1k count=36 seek=389120
```

```
dd if=/dev/zero of=/dev/mmcblk0gp1 bs=1k count=393216  
dd if=Hi3093_ext4fs.img of=/dev/mmcblk0gp1 bs=1k  
dd if=Hi3093_ext4fs_cms.bin of=/dev/mmcblk0gp1 bs=1k count=36 seek=389120
```

```
dd if=/dev/zero of=/dev/mmcblk0gp2 bs=1k count=393216  
dd if=Hi3093_ext4fs.img of=/dev/mmcblk0gp2 bs=1k  
dd if=Hi3093_ext4fs_cms.bin of=/dev/mmcblk0gp2 bs=1k count=36 seek=389120
```

----结束

2.2 eMMC 升级 eMMC 流程

步骤1 单板连接电脑，连接单板网口j12，连接单板串口j26，并且更改电脑对应以太网的TCP/IPv4 属性 ip地址：192.168.0.10 子网掩码：255.255.255.0。

图 2-1 PC 端 IP 配置参考图

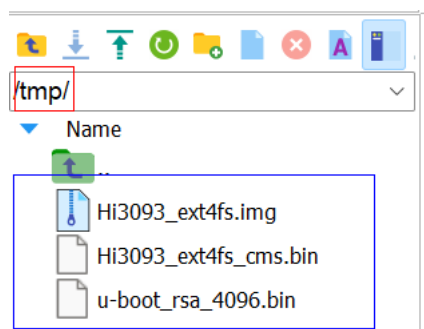


步骤2 成功进入openEuler系统，执行如下命令。

```
cd /  
sh /hi3093_init.sh (如果在系统启动阶段已自动执行过此命令，则不用再次执行)  
ifconfig eth2 192.168.0.11
```

步骤3 通过MobaXterm建立ssh连接，remote host设为192.168.0.11，把第1.2 制作eMMC镜像中制作完成的Hi3093_ext4fs.img、Hi3093_ext4fs_cms.bin和u-boot_rsa_4096.bin通过拖曳方式放在/tmp目录下，如图2-2所示。

图 2-2 文件传输示意图





步骤4 执行如下命令。

```
cd /tmp
sh /hi3093_upgrade.sh 4096
reboot
```

步骤5 在uboot控制台执行“env default -a”和“saveenv”重新设置环境变量，然后执行“run bootcmd”进入内核。

----结束

须知

首次升级eMMC执行**步骤3**时需将l0fw_rsa_4096.bin和l1fw_rsa_4096.bin文件也传输到单板/tmp目录下

2.3 eMMC 启动流程

步骤1 更改拨码开关SW1，以on侧为1，记SW1 1~4为1000，即1开关拨至on侧，2~4开关拨至对侧。

步骤2 串口接J26，网口接J12，打开电源开关。

步骤3 启动至出现第一个“ctrl + b”字样，按下“ctrl + b”打断启动，进入L1串口，未成功进入重新上下电重新操作即可。

步骤4 执行命令：mmc k 1。

步骤5 执行命令：reboot。

步骤6 进入操作系统即为启动完成。

----结束

须知

烧写SFC包后首次启动eMMC才需按照此流程操作，eMMC镜像升级无需重复此操作。



3 SFC 镜像构建

- 步骤1** 构建SFC烧片包前请确保按照[1.3.1 源码和编译环境准备](#)准备好编译环境并按照[1.3.2.1 Hi3093 SDK 编译准备](#)完成内核编译。
- 步骤2** 进入build/build_sfc目录，执行build_sfc.sh脚本，在sfc_output目录中的Hi3093_flash_ext4.bin即为SFC烧片包，此时uboot读取sfc中的内核和文件系统，可用于eMMC空片时启动，此时烧片包为64M；在完成64M烧片包编译后，可以执行build_sfc_16M.sh脚本生成16M烧片包，此时sfc_output目录中的Hi3093_flash_ext4.bin大小为16M。
- 步骤3** 若想从sfc启动Baremetal系统，执行build_sfc.sh baremetal即可，sfc_output目录中的Hi3093_flash_ext4.bin即为SFC烧片包。

----结束

须知

本文档当前仅提供SFC镜像构建方法，SFC烧写流程由客户自己完成。



4 SFC 启动混合部署与 Baremetal 方案

本章节描述从SFC启动混合部署系统镜像和Baremetal镜像方式。

4.1 SFC 启动 openEuler 和混合部署系统方案

- 步骤1** 将build/build_fs/init脚本中80和81行中的nvm_active替换为目标分区(如mmcblk0p1), 参考[openEuler eMMC镜像构建](#)或其它混合部署方案编译运行指南完成镜像的编译。
- 步骤2** 参考第三章[SFC镜像构建](#)步骤2完成SFC烧片包构建并烧写入SFC, 烧片后参考[2.1 SFC升级eMMC流程](#)的[步骤1](#)至[步骤3](#)切换至SFC启动。
- 步骤3** 若当前emmc usr区未进行分区, 在uboot串口执行“env default -a”后执行“saveenv”即可设置好加载SFC镜像的环境变量, 然后执行“run bootcmd”即可; 若emmc usr区已完成分区, 直接进入步骤8。
- 步骤4** 输入账号密码后进入系统, 执行“fdisk /dev/mmcblk0”进行分区创建(例: 输入“n”进行分区创建, 输入“p”选择创建主分区, 输入“1”设定分区编号为1, 输入“144”设定该分区起始sector, 输入“6324223”作为该分区终止sector, 输入“w”保存当前创建操作, 这里sector起止之差应大于786432, 即384MB大小), 执行“mkfs.ext4 -b 4096 /dev/mmcblk0pX”(例: mkfs.ext4 -b 4096 /dev/mmcblk0p1) 格式化X分区为ext4文件系统后, reboot重启至内核。
- 步骤5** 将编译好的烧片包hi3093_ext4.tar.gz传输至板端/tmp目录(可参考[2.2 eMMC升级eMMC流程](#)步骤3), 并完成解压。
- 步骤6** 执行“mkdir /mnt/update”创建镜像挂载目录, 执行“mount -t ext4 -o loop /tmp/Hi3093_ext4fs.img /mnt/update/”完成镜像挂载。
- 步骤7** 执行“cp -rf /mnt/update/* /run/media/mmcblk0pX”将编译好的镜像中的文件系统拷贝至X分区, 并执行“reboot”重启至uboot控制台。
- 步骤8** 执行“env delete bootargs”删除bootargs(欧拉和混合部署的bootargs使用dts中存储的), 执行“setenv boot_kernel_media 1”将镜像加载介质切换至emmc usr区(默认为0启动sfc镜像), 执行“setenv part2_user_offset Y”将emmc usr区读取的起始sector设置为想启动的usr分区的起始sector(例:mmcblk0p2起始sector设置为了6324224, 这里的Y为(6324224 -144)的十六进制, 为0x607f70), 最后执行“saveenv”保存以上环境变量并执行“run bootcmd”启动emmc usr区镜像。



步骤9 至步骤9已完成emmc usr区镜像加载，若想切换回加载sfc镜像，重新执行**步骤3**即可。

----结束

4.2 SFC 启动 Baremetal 方案

步骤1 参考**3 SFC镜像构建步骤4**完成SFC烧片包构建，烧片后使用SFC启动。

步骤2 参考《Baremetal系统编译运行指南》2.1 升级uboot章节5进行环境变量配置。

步骤3 参考《Baremetal系统编译运行指南》5.3Baremetal子系统编译章节内容进行镜像编译。

步骤4 参考《Baremetal系统编译运行指南》2.2升级用户业务镜像章节内容进行镜像升级。

----结束