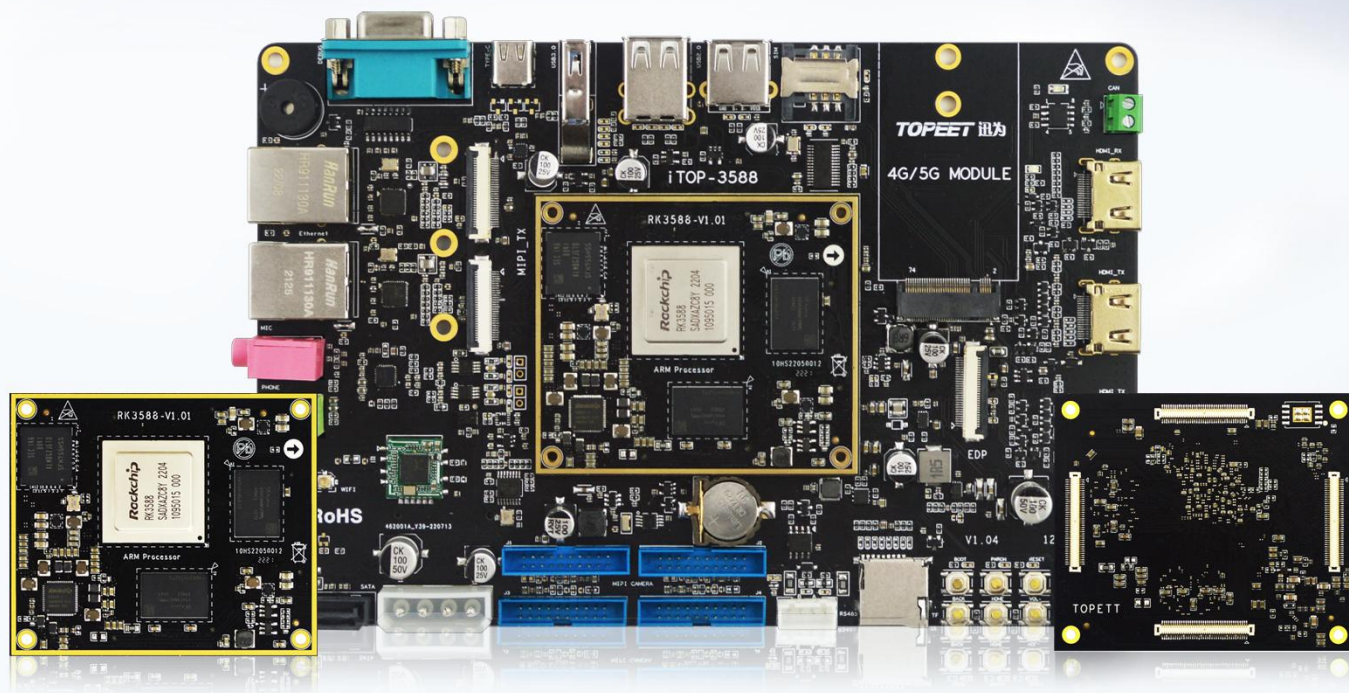


强大的 AI 能力 更快更强

超长供货周期 | 7X24 小时稳定运行 | 8K 视频编解码



iTOP-RK3588 开发板使用手册

八核 64 位 CPU | 主频 2.4GHz | NPU 算力 6T | 4800 安防级别 ISP

更新记录

更新版本	修改内容
V1.0	初版
V1.1	更新系统备份方式

目录

更新记录	2
目录	3
版权声明	4
更多帮助	5
第 1 章 EMMC 镜像导出	6
1.1 EMMC 系统备份	6
1.2 打包 update.img	9

版权声明

本文档版权归北京迅为电子有限公司所有。未经本公司书面许可，任何单位和个人无权以任何形式复制、传播、转载本文档的任何内容，违者将被追究法律责任。

更多帮助

注意事项与维护

- ❖ 请注意和遵循标注在产品上的所有警示和指引信息；
- ❖ 请勿带电插拔核心板及外围模块；
- ❖ 使用产品之前，请仔细阅读本手册，并妥善保管，以备将来参考；
- ❖ 请使用配套电源适配器，以保证电压、电流的稳定；
- ❖ 请勿在冷热交替环境中使用本产品，避免结露损坏元器件；
- ❖ 请保持产品干燥，如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干；
- ❖ 请勿使用有机溶剂或腐蚀性液体清洗本产品；
- ❖ 请勿在多尘、脏乱的环境中使用本产品，如果长期不使用，请包装好本产品；
- ❖ 如果在震动场景使用，请做好核心板与底板的固定，避免核心板跌落损坏；
- ❖ 请勿在通电情况下，插拔核心板及外围模块(特别是串口模块)；
- ❖ 请勿自行维修、拆解本产品，如产品出现故障应及时联系本公司进行维修；
- ❖ 请勿自行修改或使用未经授权的配件，由此造成的损坏将不予保修；

资料的更新

为了确保您的资料是最新状态，请密切关注我们的动态，我们将会通过微信公众号和 QQ 群推送。

关注“迅为电子”微信公众号，不定期分享教程、资料和行业干货及产品一线资料。

迅为新媒体账号

官网: <https://www.topeetboard.com>

知乎 <https://www.zhihu.com/people/topeetabc123>

CSDN: <https://blog.csdn.net/BeiJingXunWei>



售后服务政策

1. 如产品使用过程中出现硬件故障可根据售后服务政策进行维修
2. 服务政策：参见官方网售后服务说明
<https://www.topeetboard.com/sydymfl/Service/bx.html>

送修地址：

1. 地址：北京市海淀区永翔北路 9 号中国航发大厦三层
2. 联系人：迅为开发板售后服务部
3. 电话：010-85270716
4. 邮编：100094
5. 邮寄须知：建议使用顺丰、圆通或韵达，且不接受任何到付

技术支持范围

1. 了解产品的软、硬件资源提供情况咨询
2. 产品的软、硬件手册使用过程中遇到的问题
3. 下载和烧写更新系统过程中遇到的问题
4. 产品用户的资料丢失、更新后重新获取
5. 产品的故障判断及售后维修服务。

PS: (由于嵌入式系统知识范围广泛, 我们无法保证对各种问题都能一一解答, 部分内容无法供技术支持, 只能提供建议。)

技术支持

1. 周一至周五: (法定节假日除外)
上午 9:00 ~ 11:30 / 下午 13:30 ~ 17:30
2. QQ 技术交流群: 824412014
822183461
95631883
861311530

第 1 章 EMMC 镜像导出

在实际开发过程中，系统默认提供的功能和环境通常很难全面满足项目需求。为了实现特定功能或满足项目运行的个性化要求，开发人员往往需要对系统进行深入的定制化处理。这包括安装必要的依赖库、创建项目运行所需的配置文件、优化启动流程等。这些定制化操作不仅需要满足项目的当前需求，还需要保证系统的稳定性和运行效率。

当项目调试完成后，为了能够将已经优化好的系统环境高效地推广到多个设备上，避免手动重复配置的繁琐和可能出现的人为错误，开发人员需要将整个系统的当前状态制作成一个完整的镜像文件。通过导出系统镜像，可以将所有定制内容完整地打包。这种方式不仅大幅减少了后续批量生产中的工作量，还可以确保每台设备都拥有一致的运行环境，降低潜在的部署风险，提升生产效率。

在本篇手册中，将会讲解如何导出 EMMC 当前运行的系统镜像，并打包为可以整体烧写的 update.img 镜像，从而将其应用到批量生产的实际场景中。

1.1 EMMC 系统备份

本篇手册适合备份全部的 Linux 系统镜像，包括 Buildroot、Ubuntu 和 Debian，这里我们选用 ubuntu22 系统进行 EMMC 镜像备份演示，首先将“iTOP-RK3588\02_【iTOP-RK3588 开发板】开发资料\09_Linux 系统开发配套资料\06_Linux 系统备份”路径下的 backup.zip 拷贝到外置存储设备例如 U 盘、TF 卡、NVME 固态硬盘上，这里选用最方便的 U 盘(建议格式化为 NTFS 格式，FAT32 格式会有 4G 大小的限制)，拷贝完成如下图所示：

<input type="checkbox"/> 名称	修改日期	类型	大小
 backup.zip	2024/11/21 17:43	360压缩 ZIP 文件	1,819 KB

然后进入开发板的系统终端，如下图所示：

[illegible]

然后插入外置存储设备，使用 `mount` 命令自行手动挂载，如下图所示：

```
root@topeet:~# mount /dev/sda /mnt/
root@topeet:~# cd /mnt/
root@topeet:/mnt# ls
backup.zip 'System Volume Information'
root@topeet:/mnt#
```

然后进入挂载目录下，可以看到刚刚拷贝的 **backup.zip** 压缩包，使用 **unzip** 命令进行解压，解压完成如下图所示：

```
root@topeet:/mnt# ls
backup.zip 'System Volume Information'
root@topeet:/mnt# unzip backup.zip
Archive: backup.zip
  creating: backup/
  creating: backup/rkbin/
  creating: backup/rkbin/rk3562/
  inflating: backup/rkbin/rk3562/parameter.txt
  inflating: backup/rkbin/rk3562/MiniloaderAll.bin
  creating: backup/rkbin/rk3576/
  inflating: backup/rkbin/rk3576/parameter.txt
  inflating: backup/rkbin/rk3576/MiniloaderAll.bin
  creating: backup/rkbin/rk3568_5.10/
  inflating: backup/rkbin/rk3568_5.10/parameter.txt
  inflating: backup/rkbin/rk3568_5.10/MiniloaderAll.bin
  creating: backup/rkbin/rk3568_4.19/
  inflating: backup/rkbin/rk3568_4.19/parameter.txt
  inflating: backup/rkbin/rk3568_4.19/MiniloaderAll.bin
  creating: backup/rkbin/rk3588/
  inflating: backup/rkbin/rk3588/parameter.txt
  inflating: backup/rkbin/rk3588/MiniloaderAll.bin
  creating: backup/Image/
  creating: backup/script/
  inflating: backup/script/backup.sh
  inflating: backup/script/post-build.sh
  inflating: backup/script/mk-image.sh
  inflating: backup/script/.backup.sh.swp
root@topeet:/mnt# ls
backup backup.zip 'System Volume Information'
root@topeet:/mnt#
```

然后进入 `backu` 文件夹的 `script` 目录下，这里的 `backup.sh` 就是备份脚本，如下图所示：

```
root@topeet:/mnt/backup# ls
Image  rkbin  script
root@topeet:/mnt/backup# cd script/
root@topeet:/mnt/backup/script# ls
backup.sh  mk-image.sh  post-build.sh
root@topeet:/mnt/backup/script#
```

接下来运行 backup.sh 进行备份，运行之后会选择当前 SOC 类型，现在要备份的是 rk3588 版本的系统，所以这里输入 4 即可开始备份，备份过程如下所示：

```
root@topeet:/mnt/backup/script# ./backup.sh

请输入要构建CPU的序号：
[0] 退出
[1] rk3562
[2] rk3568_4.19
[3] rk3568_5.10
[4] rk3588

选择：4
设置 SOC=rk3588.....
开始备份分区...
正在备份 /dev/mmcblk0p1 到 ../Image/uboot.img (稀疏模式) ...
记录了1+0 的读入
记录了1+0 的写出
4194304字节 (4.2 MB, 4.0 MiB) 已复制, 0.0913655 s, 45.9 MB/s
正在备份 /dev/mmcblk0p2 到 ../Image/misc.img (稀疏模式) ...
记录了1+0 的读入
记录了1+0 的写出
4194304字节 (4.2 MB, 4.0 MiB) 已复制, 0.0996875 s, 42.1 MB/s
正在备份 /dev/mmcblk0p3 到 ../Image/boot.img (稀疏模式) ...
记录了16+0 的读入
记录了16+0 的写出
67108864字节 (67 MB, 64 MiB) 已复制, 0.877103 s, 76.5 MB/s
正在备份 /dev/mmcblk0p4 到 ../Image/recovery.img (稀疏模式) ...
83886080字节 (84 MB, 80 MiB) 已复制, 1 s, 81.3 MB/s
记录了32+0 的读入
记录了32+0 的写出
134217728字节 (134 MB, 128 MiB) 已复制, 1.63009 s, 82.3 MB/s
正在备份 /dev/mmcblk0p5 到 ../Image/backup.img (稀疏模式) ...
记录了8+0 的读入
记录了8+0 的写出
33554432字节 (34 MB, 32 MiB) 已复制, 0.446823 s, 75.1 MB/s
开始备份 rootfs 分区...
挂载 /dev/mmcblk0p6 到 ../binary...
清理 rootfs 不需要的数据...
运行 mk-image.sh 制作 rootfs 镜像...
```

等待备份完成，备份完成如下图所示：

```
创建含有 1265152 个块 (每块 4k) 和 316368 个 inode 的文件系统
文件系统 UUID: 216f978c-5b7b-41c3-a584-6d049300f188
超级块的备份存储于下列块：
32768, 98304, 163840, 229376, 294912, 819200, 884736

正在分配组表： 完成
正在写入 inode表： 完成
创建日志 (16384 个块)： 完成
将文件复制到设备： 完成
写入超级块和文件系统账户统计信息： 已完成

Rootfs Image: ../Image/rootfs.img
卸载 ../binary...
所有分区和文件系统备份完成！文件保存在 ../Image
root@topeet:/mnt/backup/script#
```

备份完成的镜像会存放在上一级目录的 Image 目录下，如下图所示：

```
root@topeet:/mnt/backup/script# cd ..
root@topeet:/mnt/backup# ls Image/
backup.img  MiniLoaderAll.bin  parameter.txt  rootfs.img
boot.img   misc.img            recovery.img   uboot.img
root@topeet:/mnt/backup#
```

可以看到分区文件、rkbin 镜像、uboot 镜像、内核镜像、文件系统镜像都已进行了导出，至此，关于 EMMC 系统备份就讲解完成了。

1.2 打包 update.img

在上一个小节中已经将 EMMC 中的全分区镜像进行了导出，而为了方便期间和统一烧写，我们一般都需要将这些镜像文件打包成一个整体的镜像 update.img，在本小节中将对镜像打包的过程进行讲解。

首先将上一小节 U 盘 Image 目录下的全部文件拷贝到对应 Linux SDK 的 output/firmware 目录下(该目录默认不存在，可以编译一个 uboot 镜像来生成)，拷贝完成如下图所示：

```
topeet@topeet:~/rockchip/rk3588-linux$
topeet@topeet:~/rockchip/rk3588-linux$ ls
Makefile  build.sh  device  kernel  rkbin      script_run_flag  ubuntu
README.md buildroot docs    output  rkflash.sh tools
app       debian   external prebuilts rockdev    u-boot
topeet@topeet:~/rockchip/rk3588-linux$ ls output/firmware/
MiniloaderAll.bin boot.img parameter.txt rootfs.img
backup.img        misc.img recovery.img uboot.img
topeet@topeet:~/rockchip/rk3588-linux$
```

然后使用 “./build.sh updateimg” 命令进行打包，打包过程如下所示：

```
Android Firmware Package Tool v2.2
----- PACKAGE -----
Add file: ./package-file
package-file,Add file: ./package-file done,offset=0x800,size=0xbf,userspace=0x1
Add file: ./parameter.txt
parameter,Add file: ./parameter.txt done,offset=0x1000,size=0x1ec,userspace=0x1,flash_address=0x00000000
Add file: ./MiniloaderAll.bin
bootloader,Add file: ./MiniloaderAll.bin done,offset=0x1800,size=0x771c0,userspace=0xef
Add file: ./uboot.img
uboot,Add file: ./uboot.img done,offset=0x79000,size=0x400000,userspace=0x800,flash_address=0x00004000
Add file: ./misc.img
misc,Add file: ./misc.img done,offset=0x479000,size=0x400000,userspace=0x800,flash_address=0x00006000
Add file: ./boot.img
boot,Add file: ./boot.img done,offset=0x879000,size=0x400000,userspace=0x8000,flash_address=0x00008000
Add file: ./recovery.img
recovery,Add file: ./recovery.img done,offset=0x4879000,size=0x8000000,userspace=0x10000,flash_address=0x00028000
Add file: ./rootfs.img
rootfs,Add file: ./rootfs.img done,offset=0xc879000,size=0x134e0000,userspace=0x269c00,flash_address=0x00078000
Add CRC...
Make firmware OK!
----- OK -----
*****rkImageMaker ver 2.23*****
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
Running 90-updateimg.sh - build_updateimg succeeded.
topeet@topeet:~/rockchip/rk3588-linux$
```

打包完成之后会在 output/update/Image/目录下生成整体镜像 update.img。

```
topeet@topeet:~/rockchip/rk3588-linux$ ls output/update/Image/ -lah
total 11G
drwxr-xr-x 2 topeet topeet 4.0K Nov 22 13:05 .
drwxr-xr-x 3 topeet topeet 4.0K Nov 22 13:05 ..
lrwxrwxrwx 1 topeet topeet 32 Nov 22 13:04 MiniLoaderAll.bin -> ../../firmware/MiniLoaderAll.bin
lrwxrwxrwx 1 topeet topeet 25 Nov 22 13:04 backup.img -> ../../firmware/backup.img
lrwxrwxrwx 1 topeet topeet 23 Nov 22 13:04 boot.img -> ../../firmware/boot.img
lrwxrwxrwx 1 topeet topeet 23 Nov 22 13:04 misc.img -> ../../firmware/misc.img
-rw-r--r-- 1 topeet topeet 191 Nov 22 13:04 package-file
lrwxrwxrwx 1 topeet topeet 28 Nov 22 13:04 parameter.txt -> ../../firmware/parameter.txt
lrwxrwxrwx 1 topeet topeet 27 Nov 22 13:04 recovery.img -> ../../firmware/recovery.img
lrwxrwxrwx 1 topeet topeet 25 Nov 22 13:04 rootfs.img -> ../../firmware/rootfs.img
lrwxrwxrwx 1 topeet topeet 24 Nov 22 13:04 uboot.img -> ../../firmware/uboot.img
-rw-r--r-- 1 topeet topeet 5.1G Nov 22 13:05 update.img
-rw-r--r-- 1 topeet topeet 5.1G Nov 22 13:05 update.raw.img
topeet@topeet:~/rockchip/rk3588-linux$
```

至此关于打包 update.img 镜像的步骤就讲解完成了，后续直接使用 RK 提供的烧写工具进行整体烧写即可。