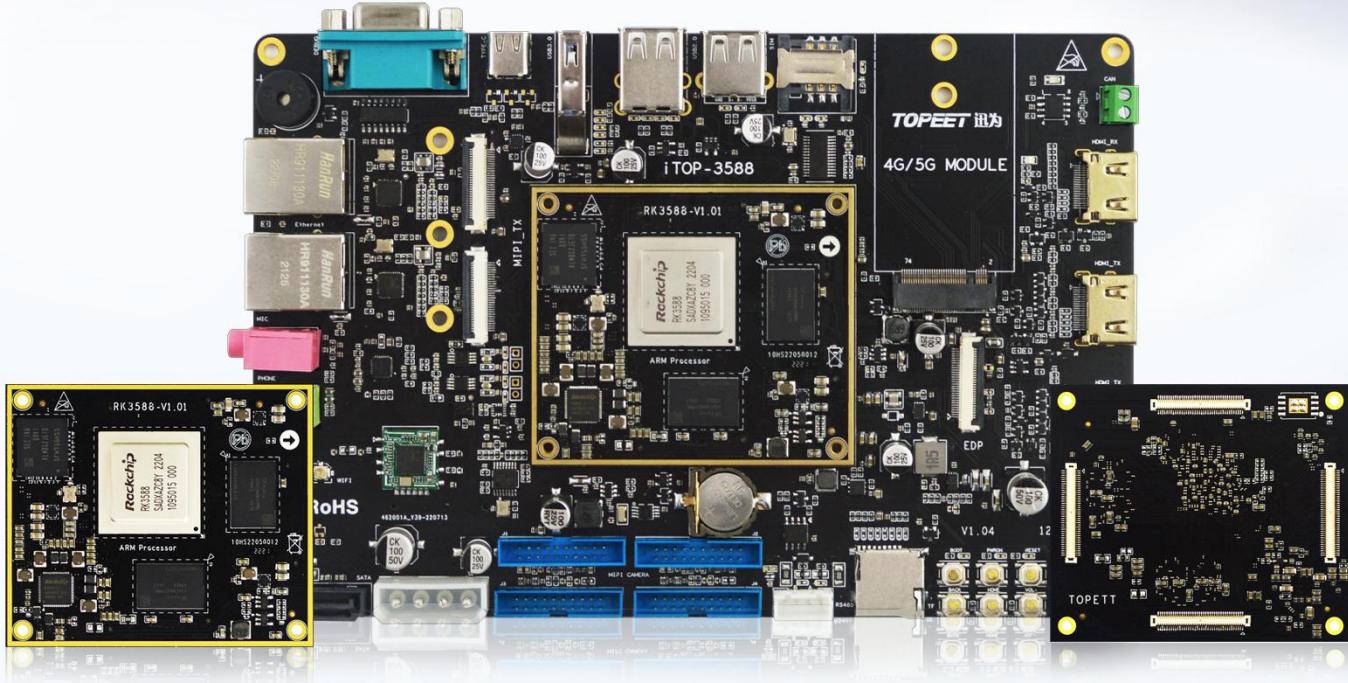


强大的 AI 能力 更快更强

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



官方微信公众号



迅为商城

更新记录

更新版本	修改内容
V1.0	初版

目录

更新记录	2
目录	3
版权声明	4
更多帮助	5
第1章 Linux 系统扩展目录	6
1.1 查看分区大小	6
1.2 修改分区大小	7

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/sydmfl/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 Linux系统扩展目录

1.1 查看分区大小

1 iTOP-RK3568 开发板以烧写 Ubuntu 系统为例，系统启动后如下图所示：

2 输入命令“df”查看系统根目录容量使用率为 23%，如下图所示：

```
root@topeet:~$ df -h
文件系统 容量 已用 可用 已用% 挂载点
/dev/root 22G 4.7G 16G 23% /
devtmpfs 3.9G 8.0K 3.9G 1% /dev
tmpfs 3.9G 0 3.9G 0% /dev/shm
tmpfs 793M 2.0M 791M 1% /run
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 3.9G 0 3.9G 0% /sys/fs/cgroup
tmpfs 3.9G 16K 3.9G 1% /tmp
/dev/mmcblk0p7 123M 13M 104M 11% /oem
/dev/mmcblk0p8 6.7G 36K 6.4G 1% /userdata
tmpfs 793M 32K 793M 1% /run/user/1000
tmpfs 793M 0 793M 0% /run/user/0
root@topeet:~$
```

3 输入命令 “`fdisk -l`” 再查看磁盘容量，32G 的 EMMC 实际可用 29.12G，我们的文件系统的分区为`/dev/mmcblk0p6`，可见该分区的大小为 22G，已经无法再扩大了，而`/dev/mmcblk0p8`

分区的大小为 6.8G，我们并不需要给 mmcblk0p8 分区给如此大的空间，那我们应该怎样来将 mmcblk0p8 分区的空间分配给 mmcblk0p6 呢。

```
root@topeet:~$ fdisk -l
Disk /dev/ram0: 4 MiB, 4194304 字节, 8192 个扇区
单位: 扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理): 512 字节 / 4096 字节
I/O 大小(最小/最佳): 4096 字节 / 4096 字节

Disk /dev/mmcblk0: 29.12 GiB, 31268536320 字节, 61071360 个扇区
单位: 扇区 / 1 * 512 = 512 字节
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: gpt
磁盘标识符: 516A0000-0000-446F-8000-69EE00001E88

设备      起点    末尾    扇区   大小  类型
/dev/mmcblk0p1  16384   24575   8192    4M 未知
/dev/mmcblk0p2  24576   32767   8192    4M 未知
/dev/mmcblk0p3  32768  163839  131072   64M 未知
/dev/mmcblk0p4  163840  425983  262144  128M 未知
/dev/mmcblk0p5  425984  491519  65536   32M 未知
/dev/mmcblk0p6  491520 46628863 46137344  22G 未知
/dev/mmcblk0p7  46628864 46891007 262144  128M 未知
/dev/mmcblk0p8  46891008 61071295 14180288  6.8G 未知
root@topeet:~$
```

1.2 修改分区大小

1.通过烧写工具解包系统 update.img 镜像，其中的 parameter.txt 文件就是我们烧写时系统的默认分区文件，如下图所示：

RKDevTool_Release_v2.84 > RKDevTool_Release_v2.84 > Output > Android > Image			
名称	修改日期	类型	大小
boot.img	2024/1/11 17:58	光盘映像文件	34,188 KB
MiniLoaderAll.bin	2024/1/11 17:58	BIN 文件	445 KB
misc.img	2024/1/11 17:58	光盘映像文件	48 KB
oem.img	2024/1/11 17:58	光盘映像文件	17,048 KB
parameter.txt	2024/1/11 17:58	文本文档	1 KB
recovery.img	2024/1/11 17:58	光盘映像文件	44,821 KB
rootfs.img	2024/1/11 17:58	光盘映像文件	5,193,728...
uboot.img	2024/1/11 17:58	光盘映像文件	4,096 KB
userdata.img	2024/1/11 17:58	光盘映像文件	4,368 KB

2.打开 parameter.txt 文件，这里跟分区有关的信息为“CMDLINE: mtdparts=**”。下面我们将对该部分分析

```

FIRMWARE_VER: 1.0
MACHINE_MODEL: RK3588
MACHINE_ID: 007
MANUFACTURER: RK3588
MAGIC: 0x5041524B
ATAG: 0x002000800
MACHINE: 0xffffffff
CHECK_MASK: 0x80
PWR_HLD: 0,0,A,0,1
TYPE: GPT
CMDLINE: mtdparts=rk29xxnand:0x00002000@0x00004000(uboot),0x00002000@
0x00006000(misc),0x00020000@0x00008000(boot),0x00040000@
0x00028000(recovery),0x00010000@0x00068000(backup),0x02c00000@
0x00078000(rootfs),0x00040000@0x02c78000(oem),-@0x02cb8000(userdata:grow)
uuid:rootfs=614e0000-0000-4b53-8000-1d28000054a9
uuid:boot=7A3F0000-0000-446A-8000-702F00006273

```

首先我们对该行内容进行分割，分割内容如下

```

CMDLINE: mtdparts=rk29xxnand:
0x00002000@0x00004000(uboot),
0x00002000@0x00006000(misc),
0x00020000@0x00008000(boot),
0x00040000@0x00028000(recovery),
0x00010000@0x00068000(backup),
0x02c00000@0x00078000(rootfs),
0x00040000@0x02c78000(oem),
-@0x02cb8000(userdata:grow)

```

@符号前是分区的大小

@符号后是分区的起始地址

括号中是分区的名字

单位都是 sector (512Bytes)

我们以 uboot 的分区信息为例子进行讲解，0x00002000@0x00004000(uboot)。首先对0x00002000 sector 化成十进制为 8192 sector， $8192 \times 512\text{Bytes} = 4096\text{KB} = 4\text{MB}$

另外 flash 最大的 block 是 4M (0x2000 sectors)，所以每个分区需要 4MB 对齐，即每个分区必须为 4MB 的整数倍。

3.我们在上面学会了每个分区参数的作用，我们如今的目的是使 rootfs 分区变大，则要改变 rootfs 的分区大小，我们这里修改为 25G，对应的分区大小的十六进制为 0x03200000，这里的分区大小改了，后面两个分区的起始地址同时也要进行修改，修改内容如下：

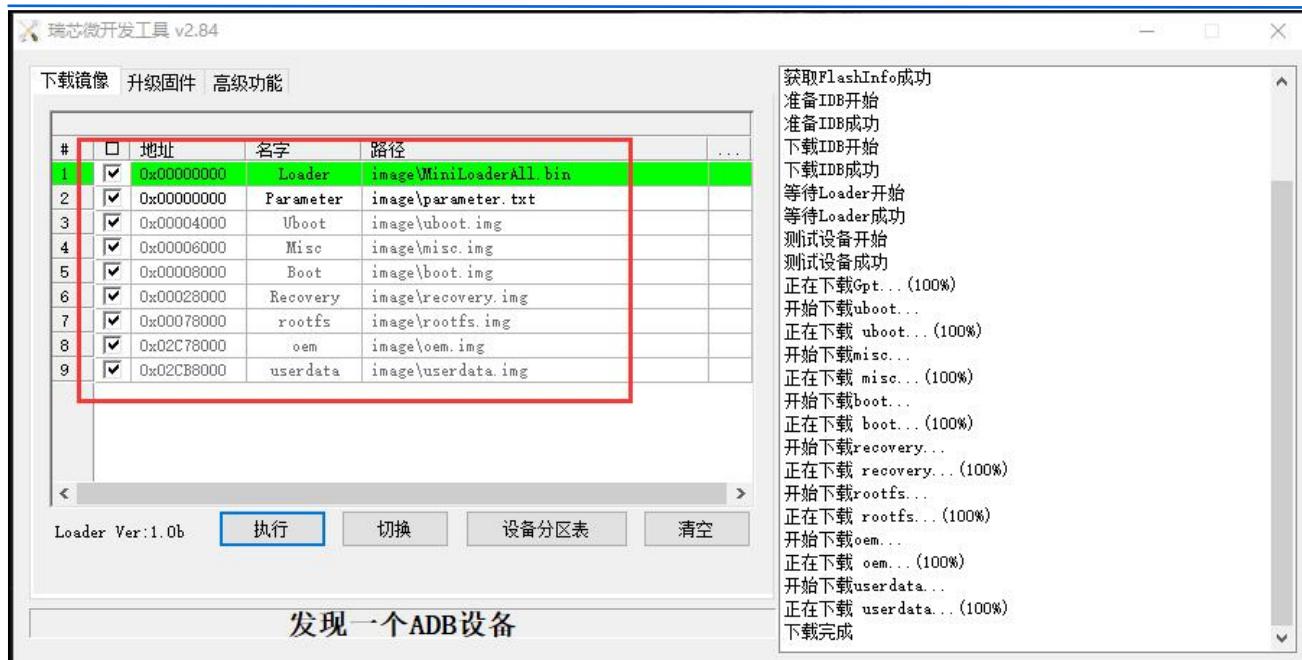
CMDLINE:

```

mtdparts=rk29xxnand:0x00002000@0x00004000(uboot),0x00002000@0x00006000(misc),0x00020
000@0x00008000(boot),0x00040000@0x00028000(recovery),0x00010000@0x00068000(backup),
0x03200000@0x00078000(rootfs),0x00040000@0x03278000(oem),-@0x032b8000(userdata:grow)

```

修改完成之后，我们重新烧写对应的镜像，如下图所示：



4. 我们首先输入命令 “`df -h`” 来查看分区的大小，如下图所示：

```
root@topeet:~$ df -h
文件系统 容量 已用 可用 已用% 挂载点
/dev/root 25G 4.7G 19G 20% /
devtmpfs 3.9G 8.0K 3.9G 1% /dev
tmpfs 3.9G 0 3.9G 0% /dev/shm
tmpfs 793M 2.0M 791M 1% /run
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 3.9G 0 3.9G 0% /sys/fs/cgroup
tmpfs 3.9G 16K 3.9G 1% /tmp
/dev/mmcblk0p8 3.7G 36K 3.6G 1% /userdata
tmpfs 793M 28K 793M 1% /run/user/1000
tmpfs 793M 0 793M 0% /run/user/0
root@topeet:~$
```

可以看到 root 的分区大小已经扩大为了 25G。

注：若默认没扩大输入命令 “`resize2fs /dev/mmcblk0p6`” 进行 EXT 文件系统扩容即可。

5. 若想编译时默认修改分区大小，修改源码中 `device/rockchip/rk3588/parameter.txt` 文件，修改完重新整体编译即可。