
Cool Pi 4 model B 用户手册

目录

第一章 初识.....	2
1.1 硬件接口.....	2
第二章 上手.....	4
2.1 镜像地址.....	4
2.2 镜像刷机.....	5
第三章 开发.....	6
3.1 内核编译.....	6
3.2 镜像制作.....	7
3.3 镜像备份.....	9

第一章 初识

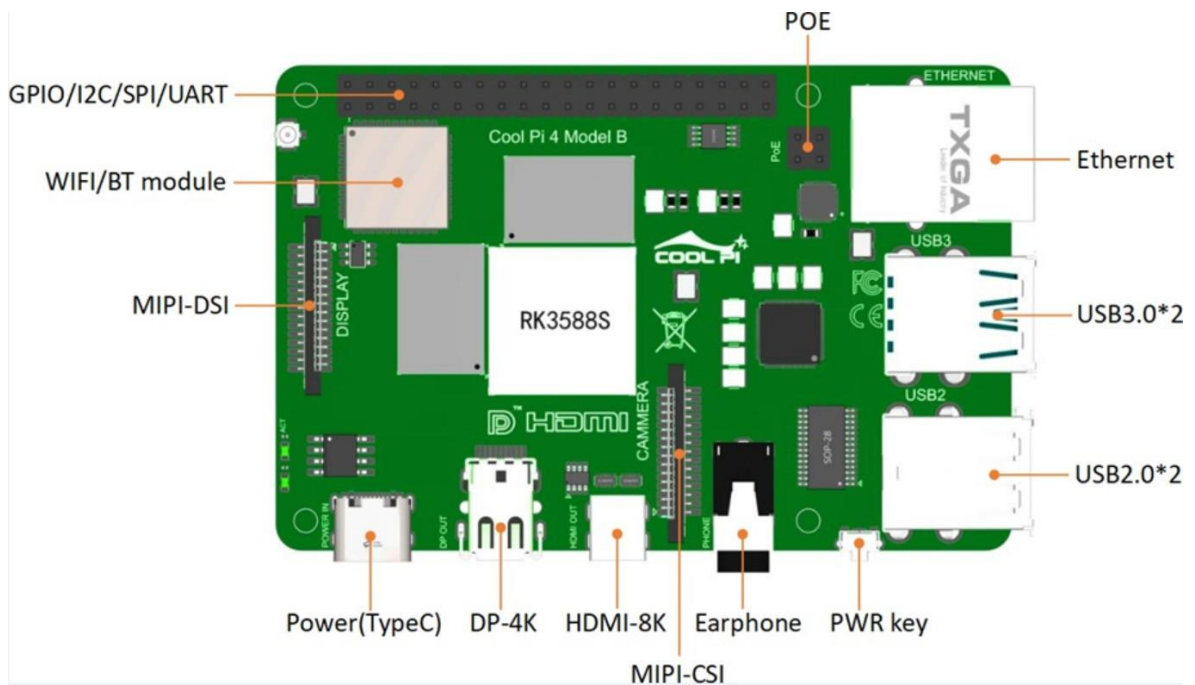
嵌入式Linux开发单板已经普及多年，选择一款合适的开发板对广大学生以及众多嵌入式爱好者尤为重要。良好的讨论社区以及丰富的学习资料，能够降低学习门槛，加深对新领域的探索兴趣，快速上手创意设计。

官方技术论坛: <https://www.cool-pi.com>

1.1 硬件接口

本派系具有丰富的硬件接口，涵盖嵌入式行业常用通信总线，如I2C、SPI、UART、CAN等。在多媒体音视频方面，能够支持8K 60帧视频输出。8核芯CPU能够满足日常多任务需求，如服务器、网关等应用场景。





上图中40 pin排母功能定义如下表（图中左下为编号1，左上为编号2，右下为编号39，右上为编号 40）

默认信号功能	引脚编号	引脚编号	默认信号功能
3.3V	1	2	5V
/dev/i2c1 sda	3	4	5V
/dev/i2c1 scl	5	6	GND
gpio 47	7	8	/dev/ttyS0 uart txd 3.3V TTL
GND	9	10	/dev/ttyS0 uart rxd 3.3V TTL
gpio 128	11	12	gpio 39
gpio 129	13	14	GND
gpio 130	15	16	/dev/ttyS2 uart txd 3.3V TTL
3.3V	17	18	/dev/ttyS2 uart rxd 3.3V TTL
spi mosi	19	20	GND
spi miso	21	22	gpio 40
spi clk	23	24	spi cs 0

默认信号功能	引脚编号	引脚编号	默认信号功能
GND	25	26	spi cs 1
/dev/i2c6 sda	27	28	/dev/i2c6 scl
gpio 131	29	30	GND
gpio 132	31	32	pwm2
gpio 133	33	34	GND
gpio 134	35	36	gpio 138
gpio 135	37	38	gpio 139
GND	39	40	gpio 115

备注:

- a) 板内集成一颗RTC时钟芯片，连接在 i2c6节点下;
- b) 以上gpio为默认代码配置，部分gpio可以通过更改配置复用其他功能，如can、uart、pwm等;
- c) 电源供电支持常见TypeC接口适配器，若使用POE供电请安装POE扩展板使用。

第二章 上手

本派系主板支持多种磁盘启动，如SATA硬盘（通过USB转接线）、U盘、TF卡、EMMC等，均可以离线刷机。您可以提前下载镜像文件到任意电脑，安装刷机工具，制作系统盘。当您拿到本派系主板，插入启动盘通电即可开机使用，非常方便。

2.1 镜像地址

本派系提供参考镜像（通过百度云下载），也欢迎开发者分享交流镜像。

Debian11

链接: https://pan.baidu.com/s/1Nig5flHkWghwwgb7_Gmabg?pwd=hqi4

提取码: hqi4

登陆用户名 coolpi 默认密码 coolpi

其他系统镜像在论坛上不定期更新，请关注。

2.2 镜像刷机

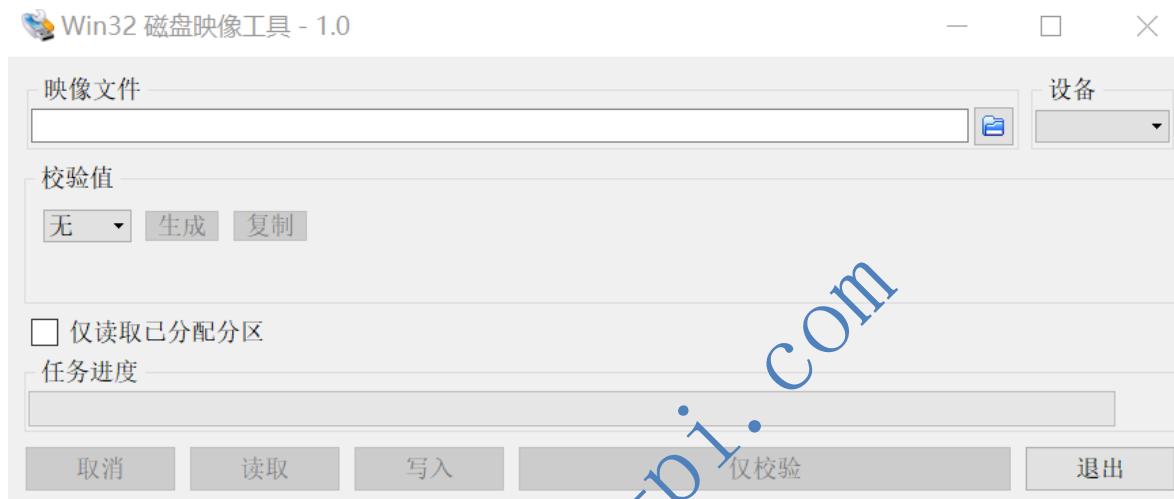
本派系支持多种刷机工具，如 Win32DiskImager、 balenaEtcher 。刷机工具无相关依赖，任意电脑均可以操作。

可选工具一

Win32DiskImager

链接: https://pan.baidu.com/s/1iPhicMFgt688md_iTbCIIA?pwd=nbxg

提取码: nbxg



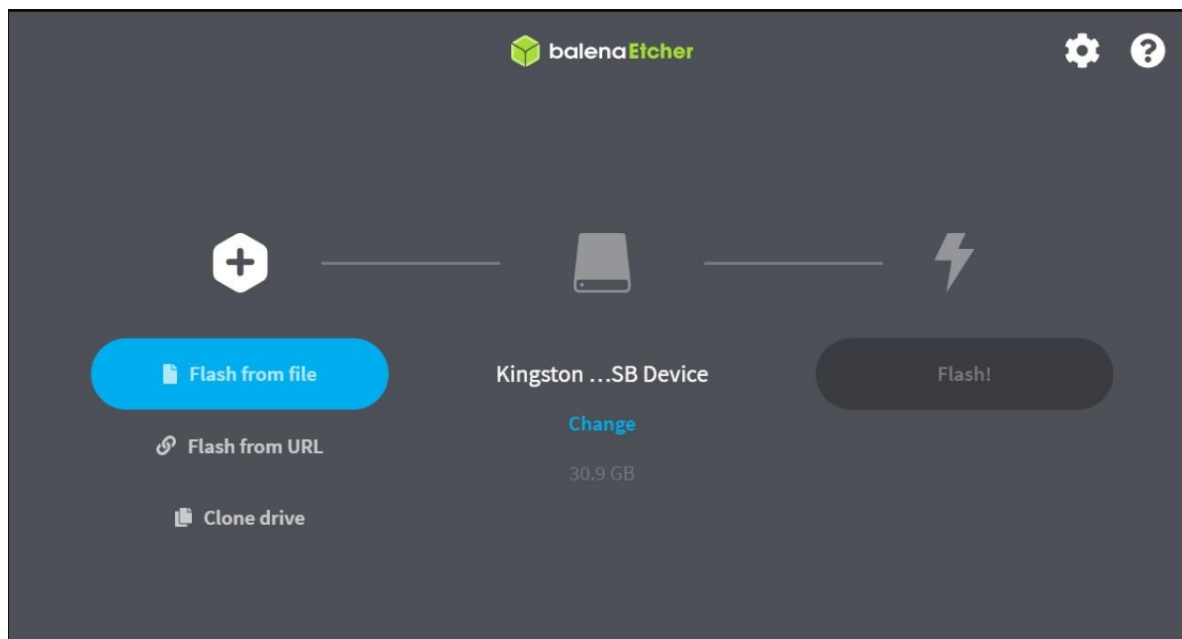
备注: 请注意设备选择待刷机的磁盘, 刷机执行过程中会清除该磁盘空间内容数据。

可选工具二

balenaEtcher

链接: https://pan.baidu.com/s/1CbkId6_TeeRnaKWYxolzSA?pwd=zkjm

提取码: zkjm



备注：请注意设备选择待刷机的磁盘，刷机执行过程中会清除该磁盘空间内容数据。

推荐在linux系统（Debian、Ubuntu等）环境中刷机，假定待刷机的磁盘节点是sdx，可以使用如下指令完成：

```
dd if=镜像.img of=/dev/sdx bs=1M status=progress;sync
```

第三章 开发

本派系主板开放linux源码，方便开发者调试内核驱动。相应镜像制作方法可以在论坛www.cool-pi.com交流探讨。

3.1 内核编译

linux内核源码支持在X86-64环境开发编译，也支持在本派系主板上直接开发编译。相应方法一致，如下以本派系主板运行Ubuntu系统为例说明：

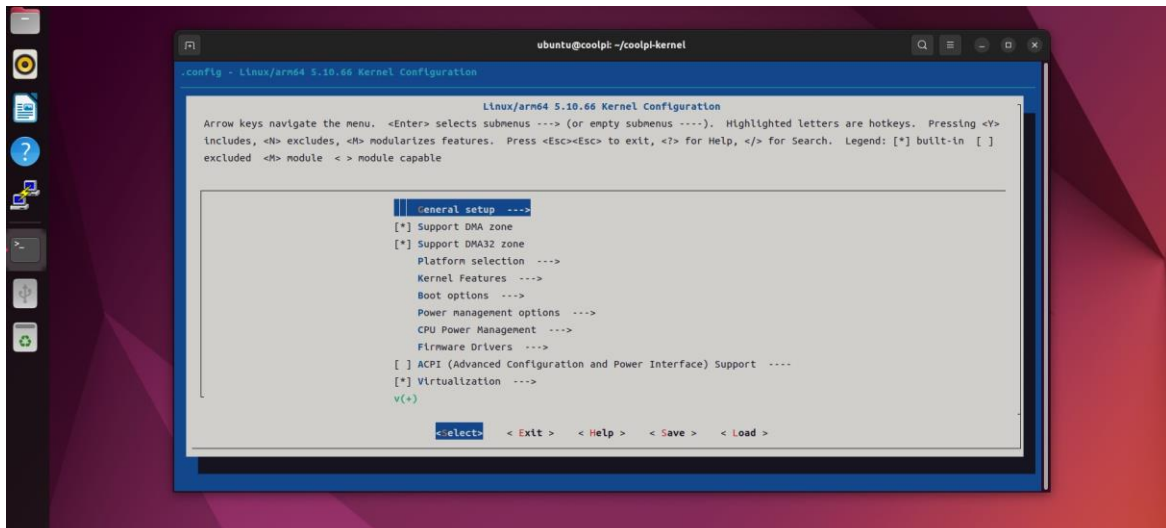
下载源码

```
ubuntu@coolpi:~$ mkdir test
ubuntu@coolpi:~$ cd test/
ubuntu@coolpi:~/test$ git clone https://github.com/yanyitech/coolpi-kernel.git
正克隆到 'coolpi-kernel'...
remote: Enumerating objects: 88605, done.
remote: Counting objects: 100% (88605/88605), done.
remote: Compressing objects: 100% (75148/75148), done.
```

执行脚本编译

```
ubuntu@coolpi:~/test/$ ./build-kernel.sh
```

编译完成后，在源码out目录下生成的文件，可用于更新系统或用于制作新镜像。



3.2 镜像制作

本派系镜像文件划分为两个分区（实际多个也可以，目前只使用两个），基本分区信息如下表：

分区名称label	分区格式	分区大小	分区内容
system-boot	第一个分区, FAT32	300MB	cmdline.txt config.txt initrd.img modules.tar.gz rk3588s-cp4.dtb vmlinuz
writable	第二个分区, ext4	根据系统大小决定	rootfs系统文件+应用程序+驱动包

我们创建一个2GB镜像文件作为范例：

a) 创建img并分区

```
~$ mkdir img_test`
~$ cd img_test/`
~/img_test$ ls`
~/img_test$ dd if=/dev/zero of=coolpi.img bs=1M count=2048`
记录了2048+0 的读入`
记录了2048+0 的写出`
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 2.1999 s, 976 MB/s`
~/img_test$ fdisk coolpi.img

欢迎使用 fdisk (util-linux 2.31.1)。
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

命令(输入 m 获取帮助): p
Disk coolpi.img: 2 GiB, 2147483648 字节, 4194304 个扇区
单元: 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: dos
磁盘标识符: 0xeeeeb672

命令(输入 m 获取帮助): n
分区类型
  p  主分区 (0个主分区, 0个扩展分区, 4空闲)
  e  扩展分区 (逻辑分区容器)
选择 (默认 p): p
分区号 (1-4, 默认 1):
第一个扇区 (2048-4194303, 默认 2048):
上个扇区, +sectors 或 +size{K,M,G,T,P} (2048-4194303, 默认 4194303): +300M

创建了一个新分区 1, 类型为“Linux”, 大小为 300 MiB。

命令(输入 m 获取帮助): t
已选择分区 1
Hex 代码(输入 L 列出所有代码): b
```

已将分区“Linux”的类型更改为“w95 FAT32”。

命令(输入 m 获取帮助): n

分区类型

p 主分区 (1个主分区, 0个扩展分区, 3空闲)

e 扩展分区 (逻辑分区容器)

选择 (默认 p): p

分区号 (2-4, 默认 2):

第一个扇区 (616448-4194303, 默认 616448):

上个扇区, +sectors 或 +size{K,M,G,T,P} (616448-4194303, 默认 4194303):

创建了一个新分区 2, 类型为“Linux”, 大小为 1.7 GiB。

命令(输入 m 获取帮助): p

Disk coolpi.img: 2 GiB, 2147483648 字节, 4194304 个扇区

单元: 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: dos

磁盘标识符: 0xeeeeb672

设备	启动	起点	末尾	扇区	大小	Id	类型
coolpi.img1		2048	616447	614400	300M	b w95	FAT32
coolpi.img2		616448	4194303	3577856	1.7G	83	Linux

命令(输入 m 获取帮助): w

分区表已调整。

正在同步磁盘。

~/img_test\$

b) 挂载分区格式化写入

```
~/img_test$ losetup -f /dev/loop18
~/img_test$ sudo losetup /dev/loop18 coolpi.img
[sudo] xxx 的密码:
~/img_test$
~/img_test$ sudo kpartx -av /dev/loop18
add map loop18p1 (253:0): 0 614400 linear 7:18 2048
add map loop18p2 (253:1): 0 3577856 linear 7:18 616448
~/img_test$
~/img_test$ sudo mkfs.vfat -F 32 /dev/mapper/loop18p1
mkfs.fat 4.1 (2017-01-24)
~/img_test$ sudo fatlabel /dev/mapper/loop18p1 system-boot
fatlabel: warning - lowercase labels might not work properly with DOS or windows
~/img_test$
~/img_test$ sudo mkfs.ext4 /dev/mapper/loop18p2
mke2fs 1.44.1 (24-Mar-2018)
丢弃设备块: 完成
创建含有 447232 个块 (每块 4k) 和 112000 个inode的文件系统
文件系统UUID: c4c8cda5-77ae-4872-9f50-4d4c20cf048f
超级块的备份存储于下列块:
    32768, 98304, 163840, 229376, 294912
```



```
正在分配组表： 完成
正在写入inode表： 完成
创建日志（8192 个块） 完成
写入超级块和文件系统账户统计信息： 已完成
```

```
~/img_test$ sudo e2label /dev/mapper/loop18p2 writable
~/img_test$
~/img_test$ sudo mount /dev/mapper/loop18p1 /mnt/
预先准备好cmdline.txt config.txt initrd.img modules.tar.gz rk3588s-cp4.dtb vmlinuz
拷贝文件到挂载目录/mnt
~/img_test$ sudo umount /mnt/
~/img_test$ sudo mount /dev/mapper/loop18p2 /mnt/
预先准备好rootfs.tar.gz
解压根文件系统到挂载目录/mnt
请注意同时将驱动包一起解压到/mnt/usr/lib
~/img_test$ sudo umount /mnt/
~/img_test$ sync
~/img_test$ sudo kpartx -dv /dev/loop18
del devmap : loop18p2
del devmap : loop18p1
~/img_test$ sudo losetup -d /dev/loop18
~/img_test$ losetup -f
/dev/loop18
~/img_test$
恭喜您完成镜像制作！
```

此时您可以插入U盘或移动硬盘，使用dd方式愉快刷写新固件。

3.3 镜像备份

本派系主板系统可以随时进行备份提取或重新制作新的镜像，方法比较简单，按照如下描述即可。

- a) 将系统盘插入linux系统（Debian、Ubuntu均可）环境；
- b) 挂载系统盘两个分区，一般系统都会自动挂载，请确认对应挂载目录；
- c) 打开终端命令行，拷贝system-boot分区目录文件备份到本地目录（如/opt）；
- d) 打开终端命令行，切换到root权限，cd 到writable目录，执行压缩命令tar -czpvf /opt/rootfs.tar.gz *
- e) 执行sync刷新磁盘写入；

至此已完成系统镜像备份，可按照3.2章节说明进行新镜像制作。