# Cool Pi 4 Model B User Manual

| Revision Date Version | | |
|---|---|---|
| Number 2022/10/14 V1.0 | | Remarks Initial |
| 2023/01/29 V1.1 | | version Update version |

Contents

# Chapter 1 Getting to Know Cool Pi

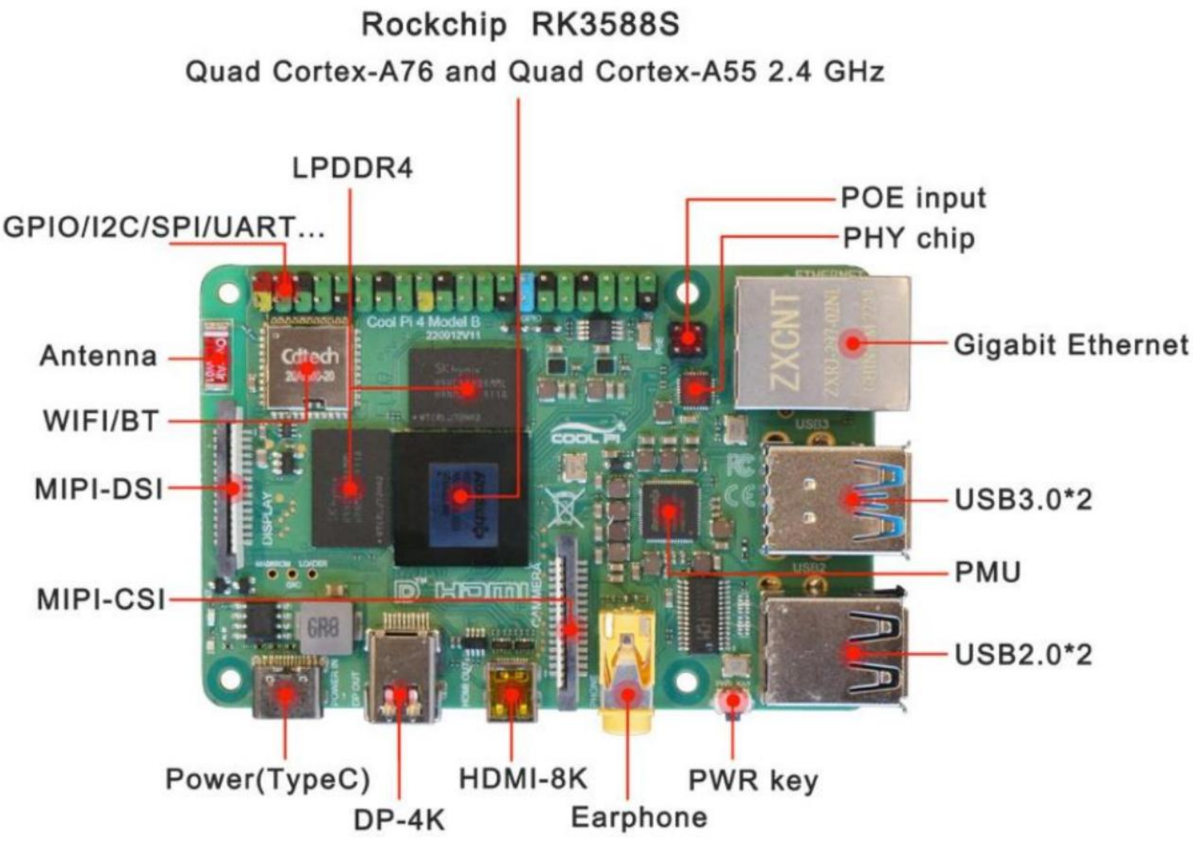
Embedded Linux development boards have been popular for many years, and choosing a suitable development board is particularly important for students and many embedded enthusiasts. A good discussion community and rich learning materials can lower the learning threshold, deepen interest in exploring new fields, and quickly get started with creative design.

Official technical forum: https://www.cool-pi.com

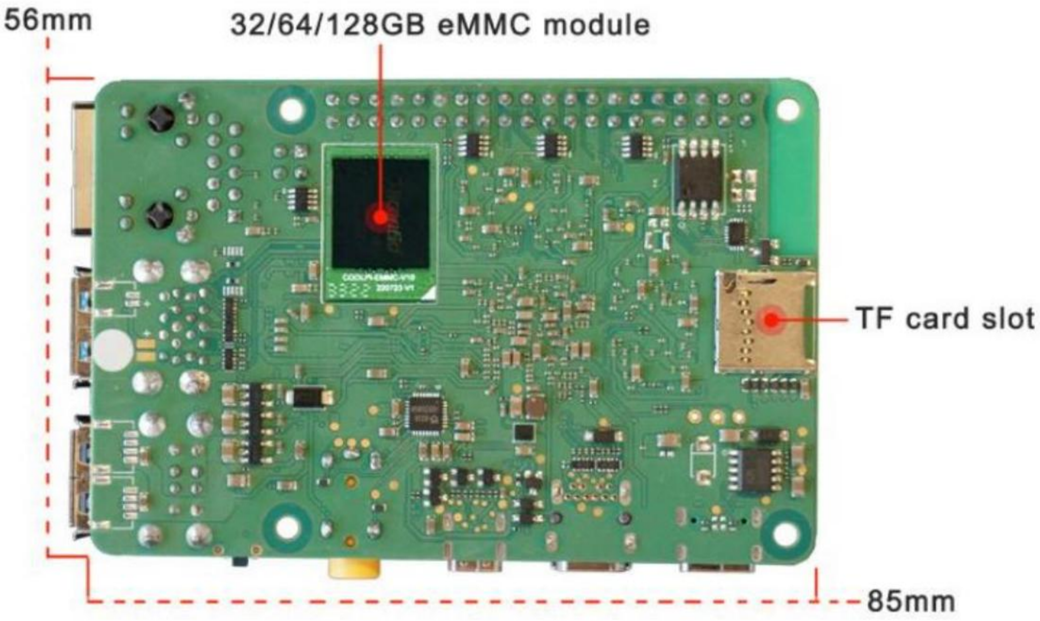
## 1.1 Hardware interface

CoolPi 4B has a wealth of hardware interfaces, covering common communication buses in the embedded industry, such as I2C, SPI, UART, CAN, etc. In terms of multimedia audio and video, micro HDMI supports up to 8K60 frame video output. The 8-core CPU up to 2.4Ghz can meet the daily multi-tasking requirements, such as server, gateway and other application scenarios. The built-in NPU has 6T computing power, which greatly enhances the realization of various model algorithms.

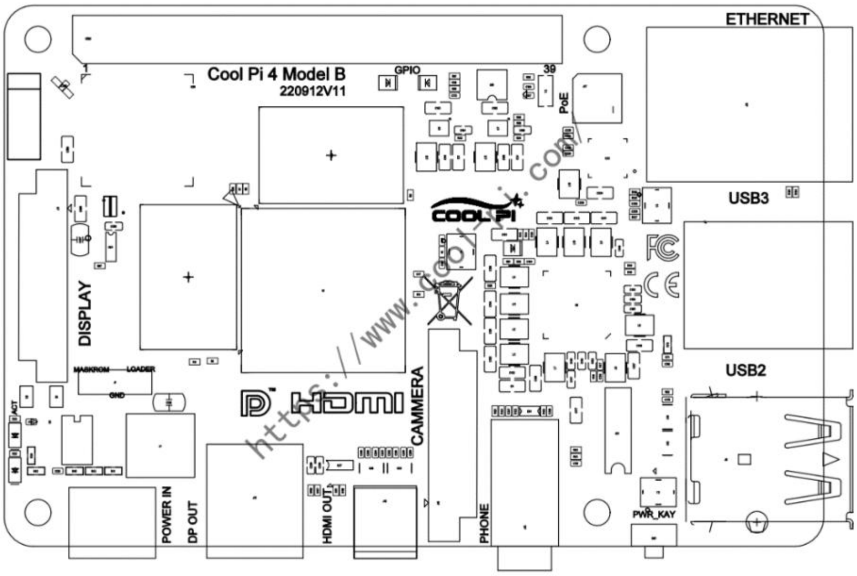
CoolPi 4B motherboard schematic diagram and related structural diagram files can be downloaded in the hardware area of the forum.



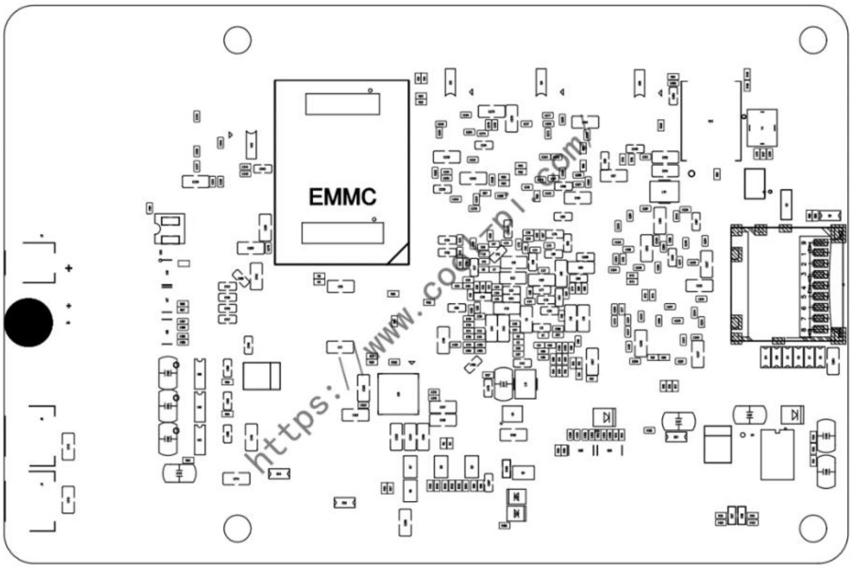
Motherboard reference picture front



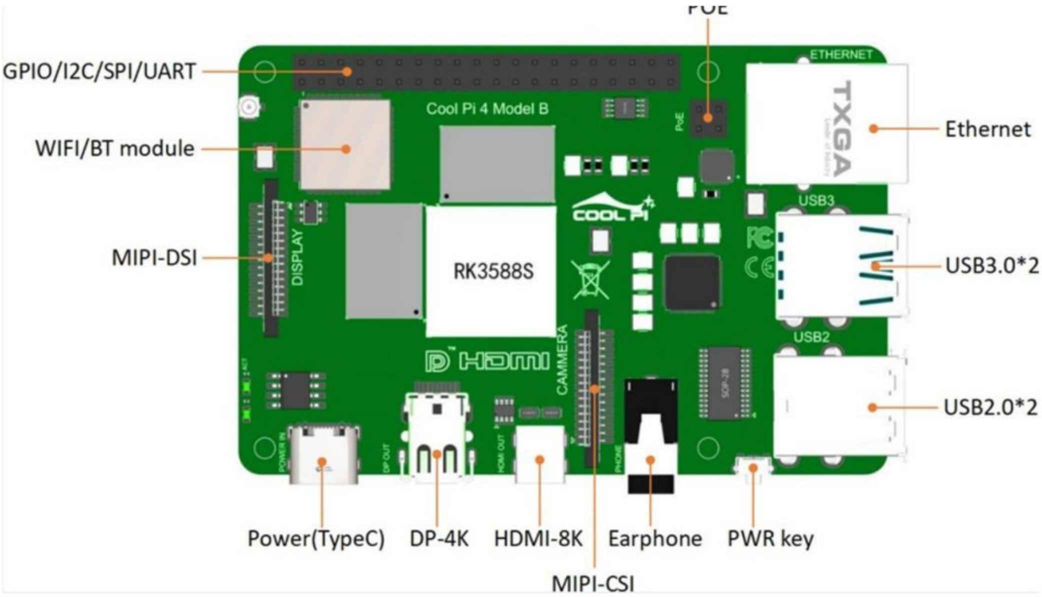
Motherboard Reference Picture Back



Motherboard bitmap front



Motherboard bitmap on the back



Reference diagram

(The lower left in the figure is No. 1, the upper left is No. 2, the lower right is No. 39, and the upper right is No. 40) The functions of the 40-pin female header in the above figure are defined in the following table:

| Default Signal Function Pin | | pin | Default signal function |
|---|---|---|---|
| | Number | number | |
| 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 eyes | 21 | 22 | gpio 40 |
| spi clk | 23 | 24 | spi cs 0 |
| 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 |

Remark:

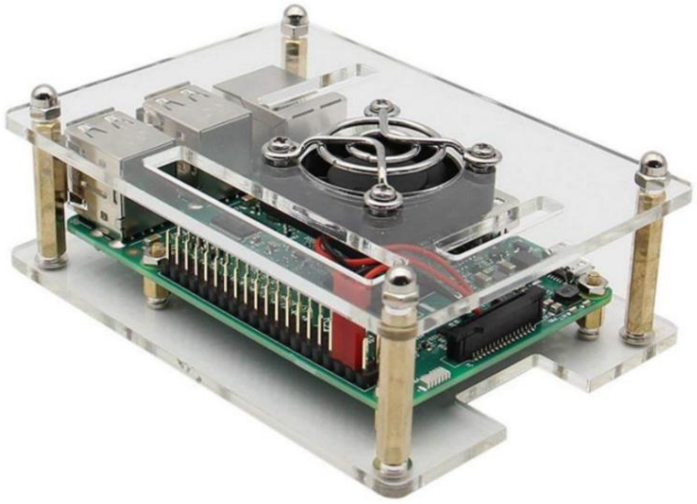
a) An RTC clock chip is integrated in the board, which is connected under the i2c6

node; b) The above gpio is the default code configuration, and some gpio can reuse other functions by changing the configuration, such as can, uart,

pwm, etc.; c) Power supply support Common TypeC interface adapter (DC5V~24V), please be careful if you use POE power supply Install POE expansion board to use (interface compatible with Raspberry Pi 4B).

## 1.2 Reference accessories

The appearance of the DP display interface on the mainboard of the case radiator is higher than that of the Raspberry Pi 4B micro HDMI interface. Users need to pay attention when choosing a case. In addition, our motherboard has an independent power key to facilitate normal power on and off. Acrylic shell https://m.tb.cn/h.UOpDuvC?tk=mujsdT1adnn



(It is recommended that the copper pillar choose a layer height of 30mm)

Share the 3D shell designed by the developer

(The forum has corresponding source files, thanks to the developers for sharing)

Display MIPI DSI

verification of a Weixue

5-inch screen https://m.tb.cn/h.UlHlw8Y?tk=XGe2dgBBkk1

Both interfaces of

Micro HDMI

cable are available https://m.tb.cn/h.UlD0BqE?tk=5E8kdT1coli

https://item.m.jd.com/product/674875.html

DP to HDMI cable

Mini DP interface can only support the standard DP protocol, the resolution can reach 4K P60, does not support INTEL DP++ protocol, so most of the ordinary Mini DP to HDMI cables on the market cannot be used. We verified two

cables for your reference https://item.jd.com/100021518367.html

(Pay attention to choose active)

https://item.jd.com/100018963014.html

Camera MIPI CSI verifies

an ov5647, accessories compatible with Raspberry Pi 4B

POE power supply

with fan interface

compatible with Raspberry Pi 4B https://m.tb.cn/h.UNRhcz8?tk=egHDdT1mbid

USB to TTL debugging

serial port reference is similar to

general-purpose transfer cables https://item.m.jd.com/product/52646835874.html

NVME to USB

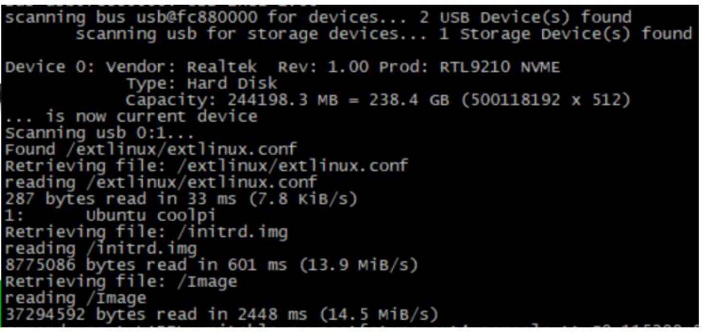
refer to similar adapter

https://item.m.jd.com/product/100014607376.html

# Chapter 2 Getting Started with Cool Pi

CoolPi 4B supports a variety of disk boots, such as SATA, NVME hard disk (via USB cable), U disk, TF card, eMMC, etc., can all be flashed offline. You can download the image file to any computer in advance, install the flash tool, and make a system disk. When you get the CoolPi 4B, plug in the boot disk and power it on to start it up, which is very convenient.

If you use USB3 to transfer the mobile hard disk, please pay attention to the hard disk power supply limit and increase the external power supply of the hard disk.



```
scanning bus usb@fc880000 for devices... 2 USB Device(s) found
       scanning usb for storage devices... 1 Storage Device(s) found

Device 0: Vendor: Realtek  Rev: 1.00 Prod: RTL9210 NVME
            Type: Hard Disk
            Capacity: 244198.3 MB = 238.4 GB (500118192 x 512)
... is now current device
Scanning usb 0:1...
Found /extlinux/extlinux.conf
Retrieving file: /extlinux/extlinux.conf
reading /extlinux/extlinux.conf
287 bytes read in 33 ms (7.8 KiB/s)
1:      Ubuntu coolpi
Retrieving file: /initrd.img
reading /initrd.img
8775086 bytes read in 601 ms (13.9 MiB/s)
Retrieving file: /Image
reading /Image
37294592 bytes read in 2448 ms (14.5 MiB/s)
```

## 2.1 Mirror address

CoolPi 4B provides reference images and related tools (via Baidu cloud download or Onedrive), and the open source code is hosted in the github repository. Developers are also welcome to share and exchange images.

Github　　百度网盘　　Onedrive

https://pan.baidu.com/s/1QV7RyMLqqK70ugYMxcXnbQ?pwd=qg2f

or

https://coolpi
my.sharepoint.com/:f:/g/personal/coolpi_coolpi_onmicrosoft_co m/EuWQQ9Cxt0pKs2-
UxgJjFFABVwsC916i49ZcjPIxM9wq8w?e=DFiNvC

Most of the system login user name coolpi default password coolpi or 123

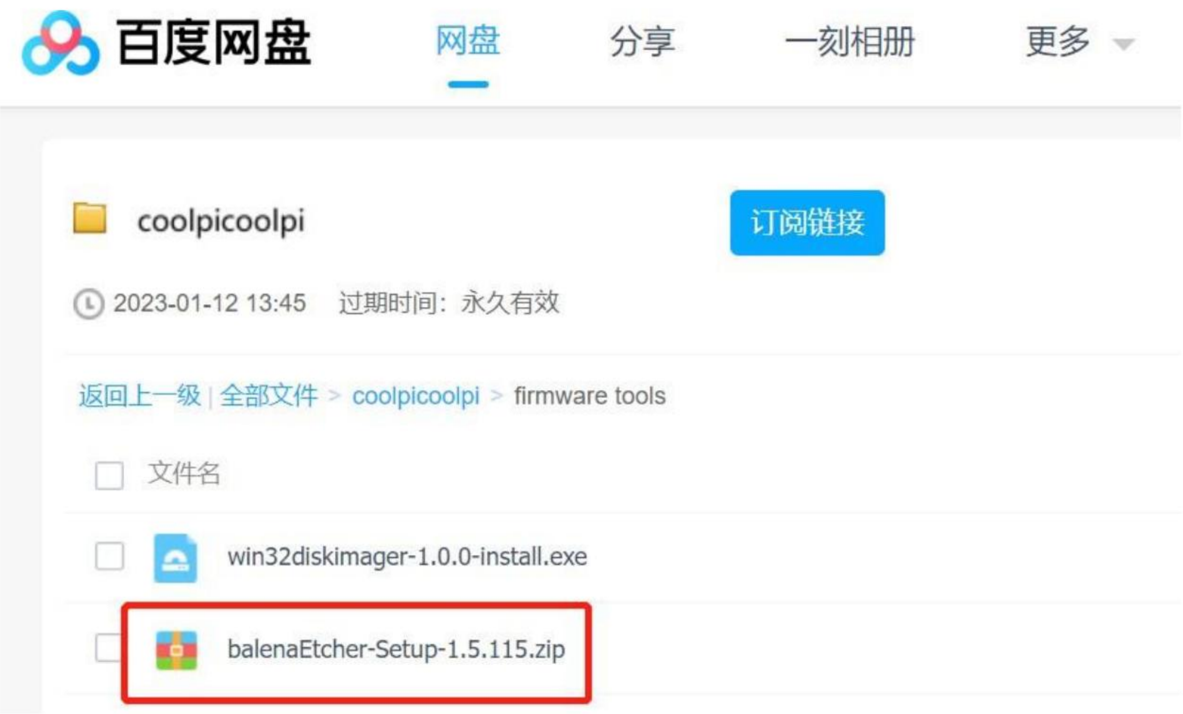
The system mirror is updated irregularly on the forum, please pay attention.

Now it supports various operating systems such as Armbian, Debian, Ubuntu, etc., and will continue to increase and expand support for other systems in the future.

## 2.2 Mirror flash

CoolPi 4B supports multiple flashing tools, such as Win32DiskImager, balenaEtcher. The flash tool has no dependencies and can be operated on any computer. The following uses balenaEtcher as an example.

Download the flashing tool installation package from Baidu Netdisk



coolpicoolpi

2023-01-12 13:45 过期时间：永久有效

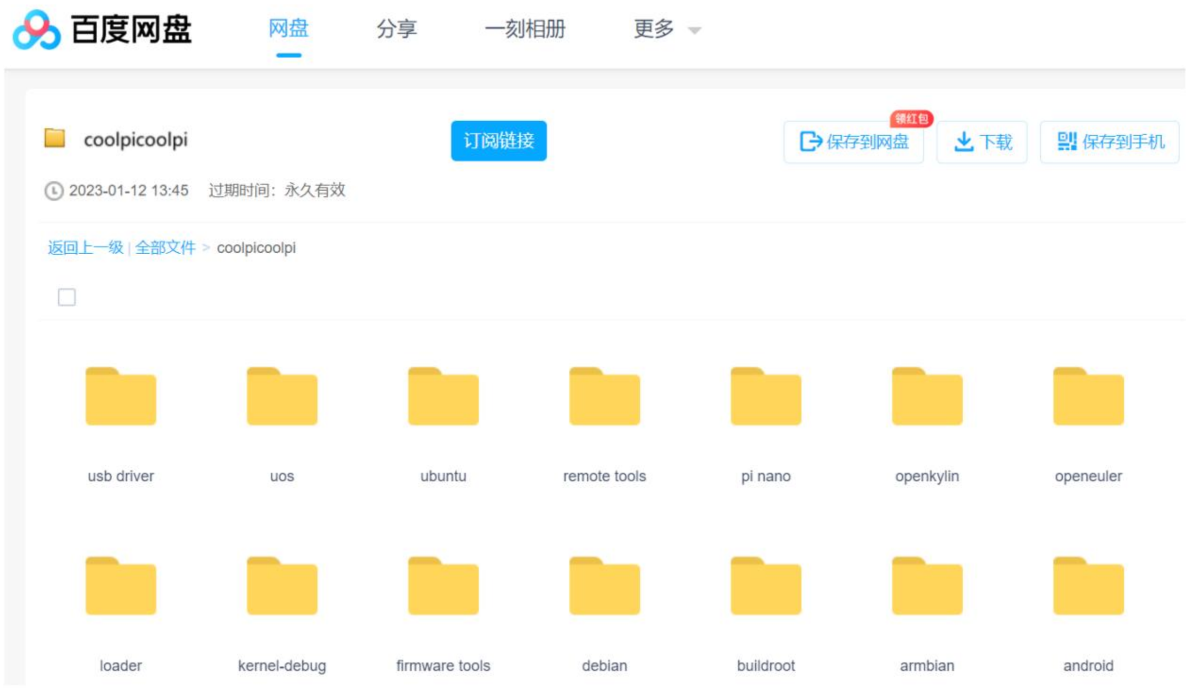
返回上一级 | 全部文件 > coolpicoolpi > firmware tools

□ 文件名

□ win32diskimager-1.0.0-install.exe

□ balenaEtcher-Setup-1.5.115.zip

Optional System Firmware



coolpicoolpi

2023-01-12 13:45 过期时间：永久有效

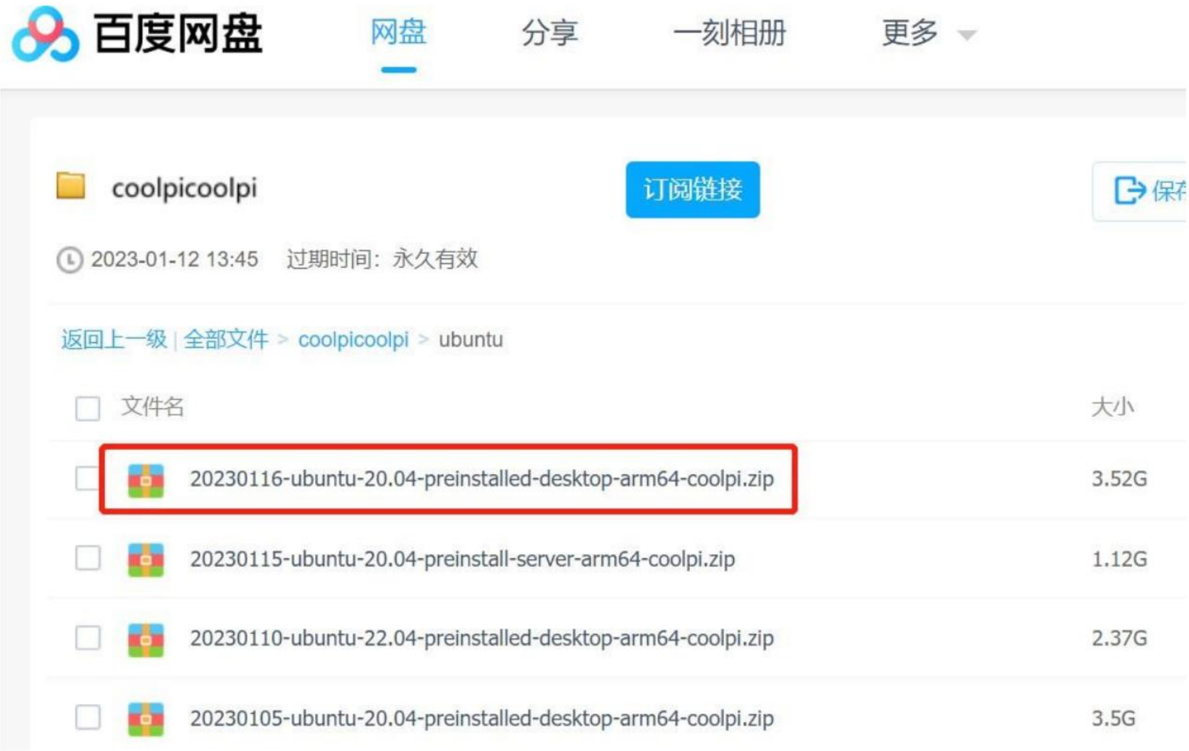
返回上一级 | 全部文件 > coolpicoolpi

□

usb driver, uos, ubuntu, remote tools, pi nano, openkylin, openeuler

loader, kernel-debug, firmware tools, debian, buildroot, armbian, android

We choose the Ubuntu20 image to introduce the operation process, download the image to the local and decompress it for backup.



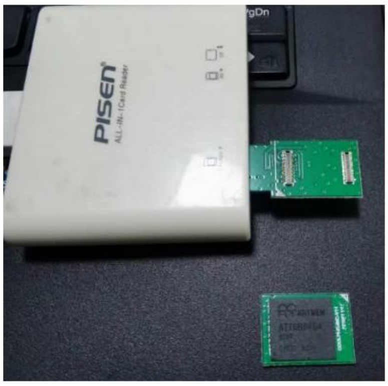
coolpicoolpi

2023-01-12 13:45 过期时间：永久有效

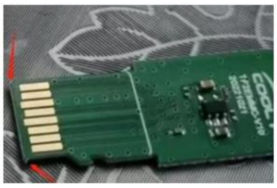
返回上一级 | 全部文件 > coolpicoolpi > ubuntu

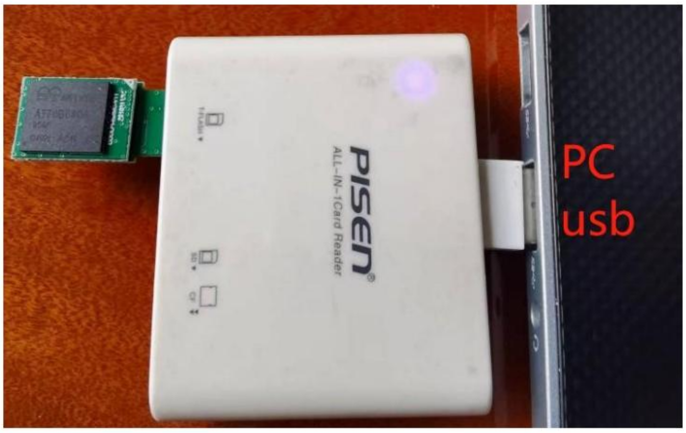
| □ 文件名 | 大小 |
|---|---|
| □ 20230116-ubuntu-20.04-preinstalled-desktop-arm64-coolpi.zip | 3.52G |
| □ 20230115-ubuntu-20.04-preinstall-server-arm64-coolpi.zip | 1.12G |
| □ 20230110-ubuntu-22.04-preinstalled-desktop-arm64-coolpi.zip | 2.37G |
| □ 20230105-ubuntu-20.04-preinstalled-desktop-arm64-coolpi.zip | 3.5G |

Buying CoolPi 4B will give you a small adapter board, which is convenient for converting eMMC modules to TF cards. Buckle the eMMC module according to the triangle mark and connect it to the card reader

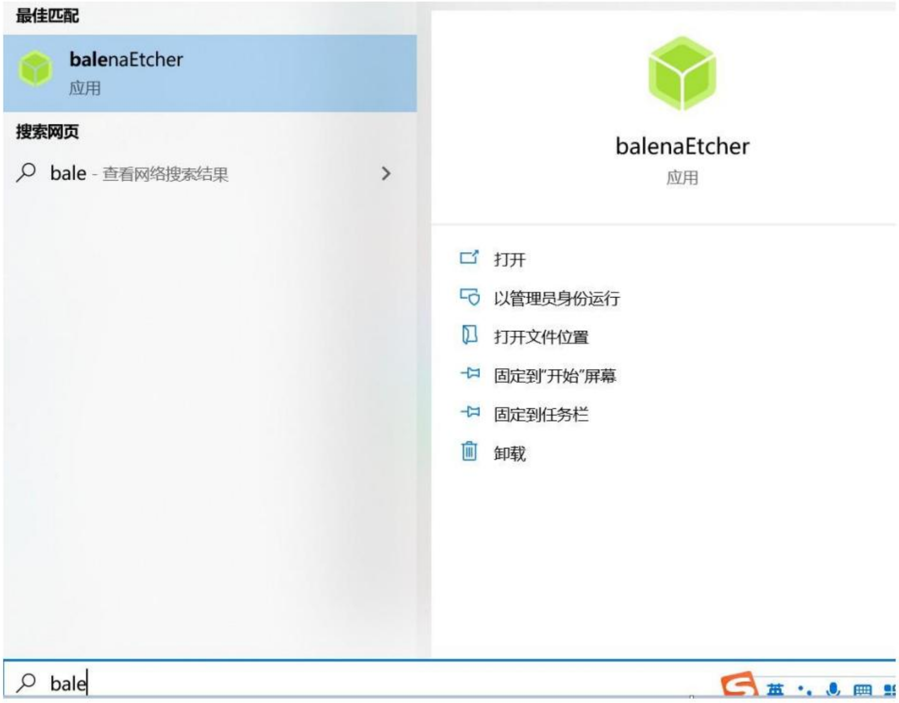


If it is not easy to insert into the card reader, grind the edge or cut it off with a knife and then insert it into the card reader

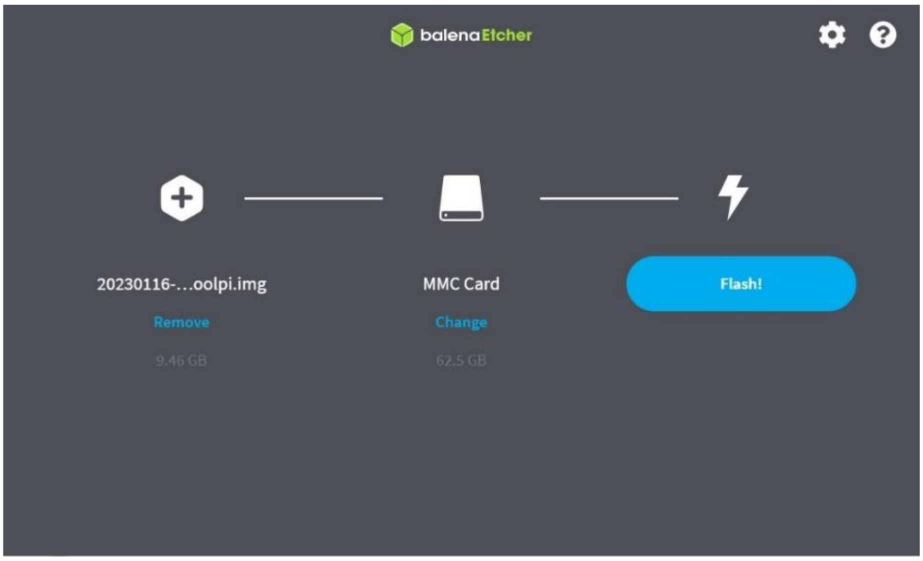




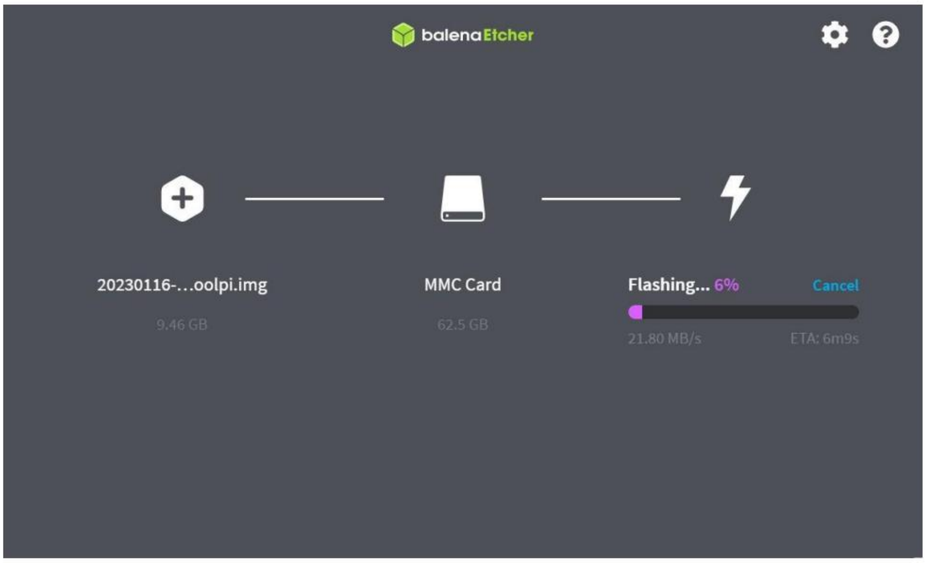
Open the burning software on Windows computer



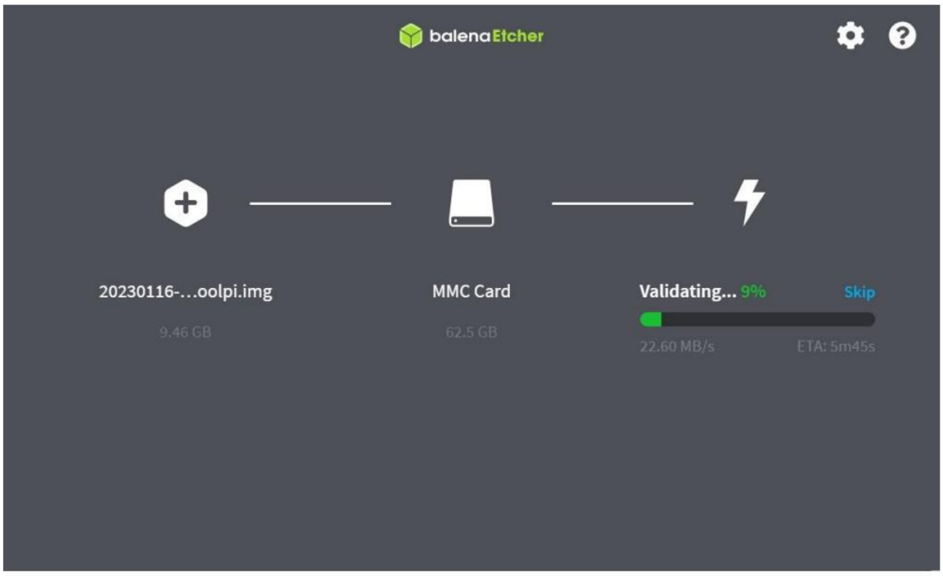
Select the image file, select the storage disk (be sure to choose carefully)

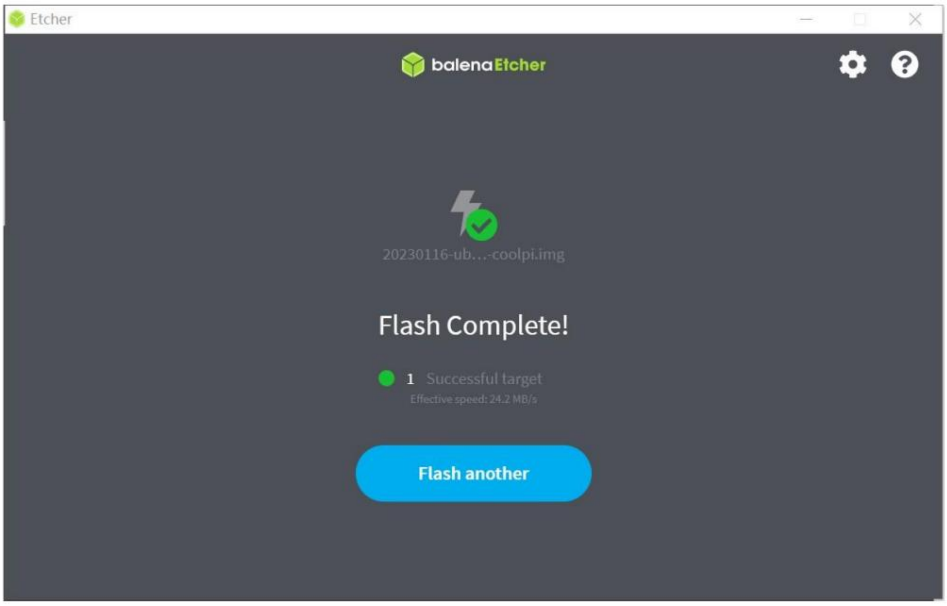


A pop-up window needs to obtain script batch processing permissions when starting, be sure to select Yes

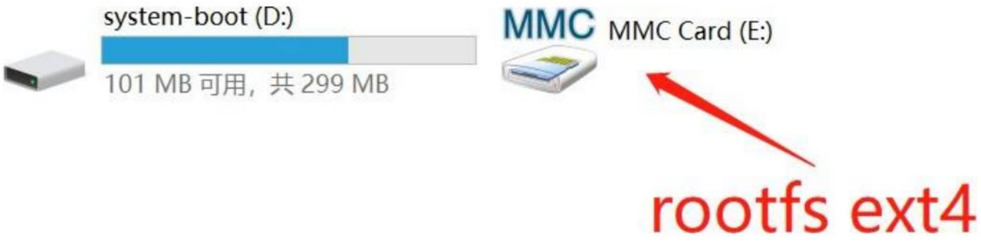


Waiting for burning verification

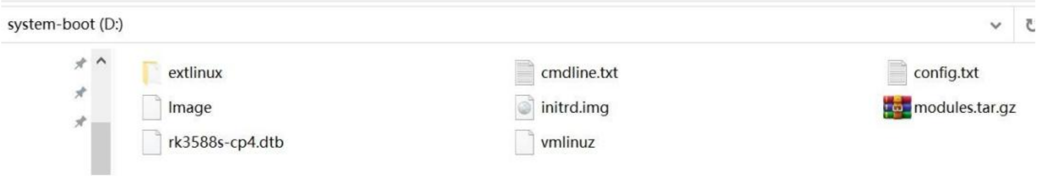




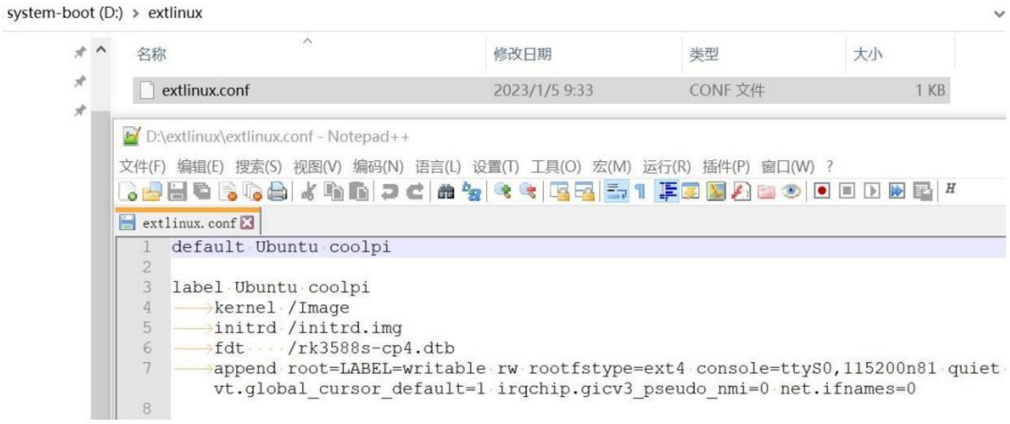
Re-plug the card reader, the computer automatically recognizes the eMMC partition, because Windows does not recognize the ext4 format partition, do not click to format the MMC partition here



The system-boot partition can edit and configure the linux kernel startup parameters according to the needs. Currently, extlinux.conf and cmdline.txt are supported, and extlinux.conf is read first.


system-boot (D:)

extlinux
Image
rk3588s-cp4.dtb
cmdline.txt
initrd.img
vmlinuz
config.txt
modules.tar.gz


system-boot (D:) > extlinux

| 名称 | 修改日期 | 类型 | 大小 |
|---|---|---|---|
| extlinux.conf | 2023/1/5 9:33 | CONF 文件 | 1 KB |

D:\extlinux\extlinux.conf - Notepad++

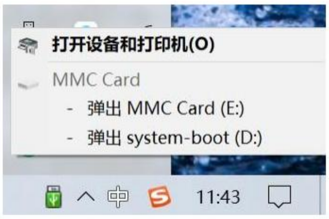
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
extlinux.conf

```
default Ubuntu coolpi

label Ubuntu coolpi
	kernel /Image
	initrd /initrd.img
	fdt    /rk3588s-cp4.dtb
	append root=LABEL=writable rw rootfstype=ext4 console=ttyS0,115200n81 quiet vt.global_cursor_default=1 irqchip.gicv3_pseudo_nmi=0 net.ifnames=0

```

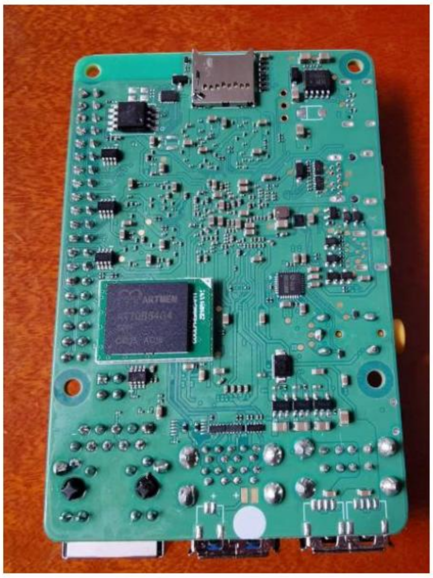
Save and exit after editing. Safely remove storage devices


打开设备和打印机(O)
MMC Card
- 弹出 MMC Card (E:)
- 弹出 system-boot (D:)

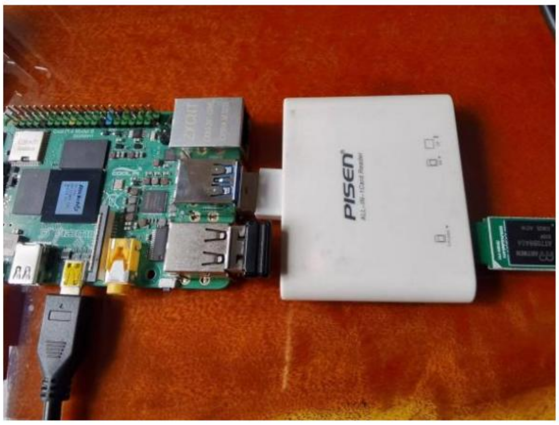
Windows 主进程 (Rundll32)

安全地移除硬件
“system-boot (D:)”设备现在可安全地从计算机移除。

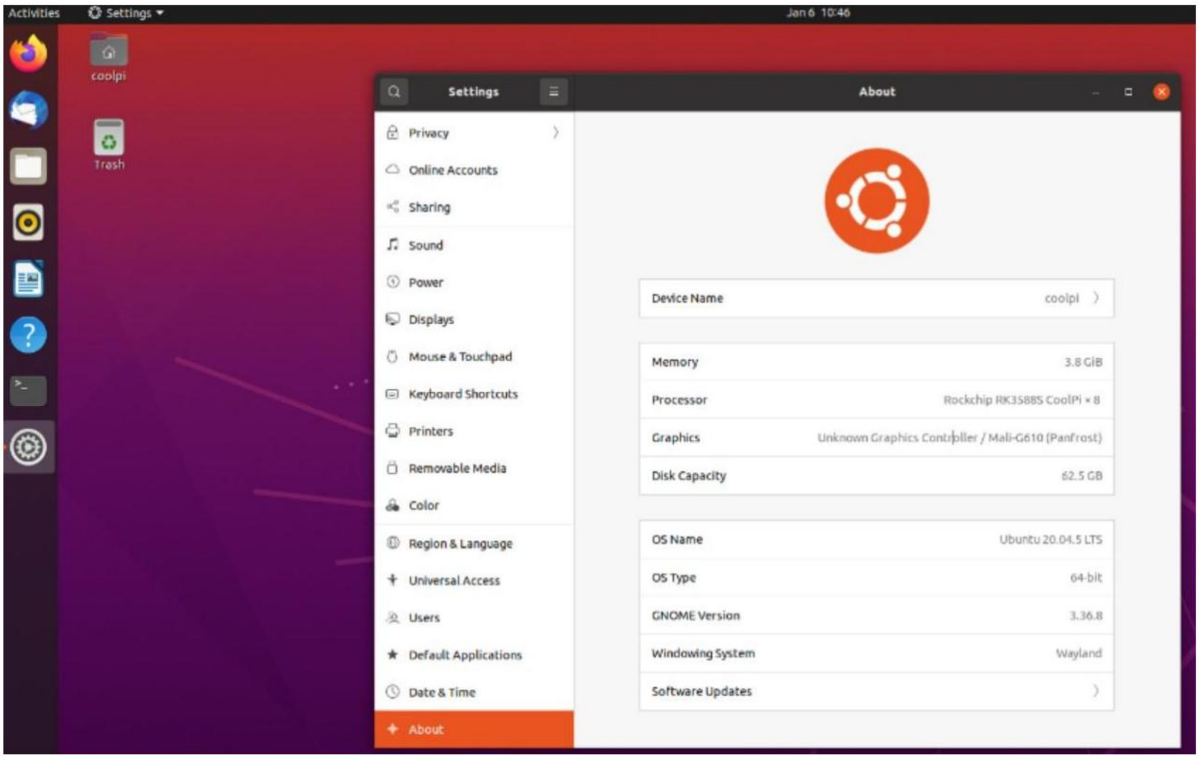
Install eMMC on CoolPi 4B



Can also boot directly from USB port



Start the system to view eMMC capacity



| | |
|---|---|
| Device Name | coolpi 〉 |
| Memory | 3.8 GiB |
| Processor | Rockchip RK3588S CoolPi × 8 |
| Graphics | Unknown Graphics Controller / Mali-G610 (Panfrost) |
| Disk Capacity | 62.5 GB |
| OS Name | Ubuntu 20.04.5 LTS |
| OS Type | 64-bit |
| GNOME Version | 3.36.8 |
| Windowing System | Wayland |
| Software Updates | 〉 |

Remark:

Please note that the device selects the disk to be flashed, and the data in the disk space will be cleared during the flashing process.

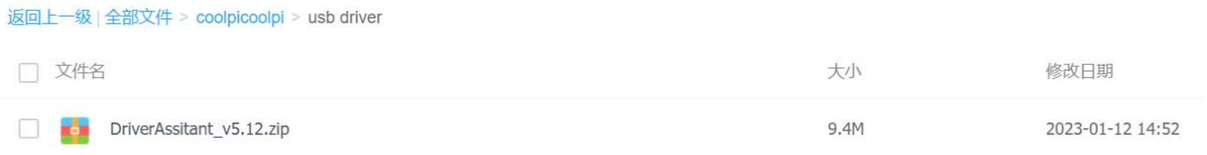
It is recommended to flash the machine in a linux system (Debian, Ubuntu, etc.) environment. Assuming that the disk node to be flashed is sdx, you can use the following command to complete it:

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

2.3 Loader burning Cool Pi

4B open u-boot source code, the user can download the source file from the github warehouse, modify and compile according to the needs, and then download and update through the following operations.

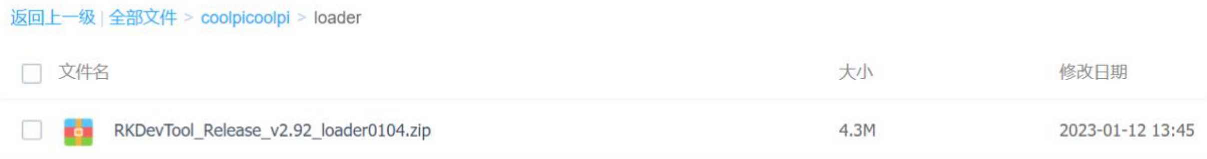
Windows computer installation driver



返回上一级 | 全部文件 > coolpicoolpi > usb driver

| □ 文件名 | 大小 | 修改日期 |
|---|---|---|
| □ DriverAssitant_v5.12.zip | 9.4M | 2023-01-12 14:52 |

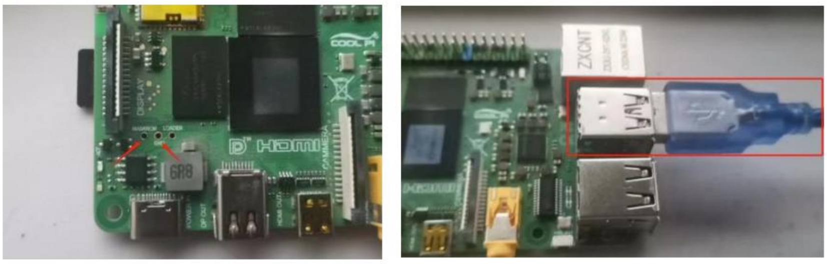
Download the unzip flashing tool



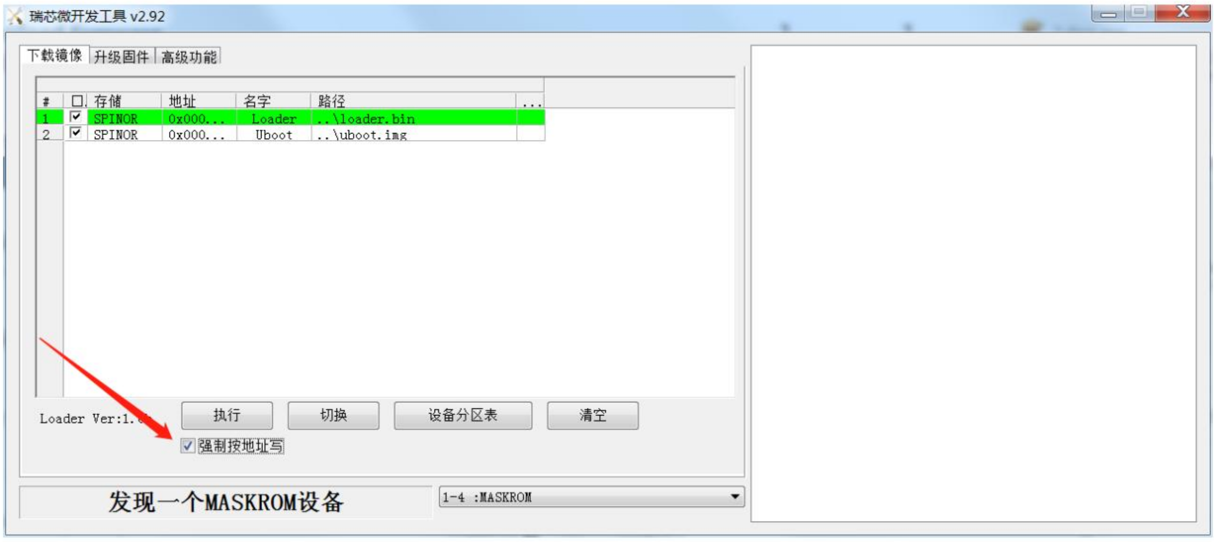
返回上一级 | 全部文件 > coolpicoolpi > loader

| □ 文件名 | 大小 | 修改日期 |
|---|---|---|
| □ RKDevTool_Release_v2.92_loader0104.zip | 4.3M | 2023-01-12 13:45 |

Use tweezers to short-circuit the MASKROM and GND positions, and connect the double-ended TYPE A USB cable to the Windows computer



Keep the tweezers shorted, power on Cool Pi 4B, and perform programming



瑞芯微开发工具 v2.92

下载镜像 | 升级固件 | 高级功能

| # | □ | 存储 | 地址 | 名字 | 路径 | ... |
|---|---|---|---|---|---|---|
| 1 | ☑ | SPINOR | 0x000... | Loader | ..\loader.bin | |
| 2 | ☑ | SPINOR | 0x000... | Uboot | ..\uboot.img | |

Loader Ver:1.

☑ 强制按地址写

发现一个MASKROM设备

1-4 :MASKROM

# Chapter 3 Develop Cool Pi

CoolPi 4B opens the linux source code, which is convenient for developers to debug the kernel driver.

The corresponding image making method can be exchanged and discussed in the forum www.cool-pi.com.

## 3.1 Kernel compilation

The linux kernel source code supports development and compilation in the X86-64 environment, and also supports direct development and compilation on the motherboard of this faction. The corresponding methods are the same, and the following uses the motherboard of this faction to run the Ubuntu system as an example:

download source code

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

Execute script compilation

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

After the compilation is complete, the files generated in the source code out directory can be used to update the system or create a new image.

Execute the script to generate the coolpi-boot.img image file

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

### 3.2 Mirror image production

We provide armbian system production warehouse, please visit the official forum www.cool-pi.com for details

The CoolPi 4B system image file is divided into two partitions (actually more than one can be used, currently only two are used), the basic partition information is as follows:

| Partition name label | Partition Format | Partition Size | Partition Content |
|---|---|---|---|
| system-boot | the first partition, FAT32 | 300MB | cmdline.txt config.txt initrd.img modules.tar.gz rk3588s-cp4.dtb vmlinuz Image extlinux/ extlinux.conf |
| | | | |
| writable | Second partition, ext4 | According to the size of the system | rootfs system file + application + driver package |

We create a 2GB image file as an example: a) Create img

and partition

```
~$ mkdir img_test` ~$ cd
img_test/` ~/img_test$
ls` ~/img_test$ dd if=/
dev/zero of=coolpi.img bs=1M count=2048`recorded 2048+0 read- in ` record 2048+0
wrote ` 2147483648 bytes
(2.1 GB, 2.0 GiB) copied,
2.1999 s, 976 MB/s` ~/img_test$ fdisk coolpi.img



Welcome to fdisk (util-linux 2.31.1). Changes will stay
in memory until you decide to write them to disk. Think twice
before using write commands.



Command (type m for help): p
Disk coolpi.img: 2 GiB, 2147483648 bytes, 4194304 sector units: sector / 1 *
512 = 512 bytes sector size (logical/
physical): 512 bytes / 512 bytes
I/O size (min/best): 512 bytes / 512 bytes Disk label type:
dos Disk identifier:
0xeeeeb672

Command (enter m for help): n
partition
    type p primary partition (0 primary partition, 0 extended
    partition, 4 free) e extended partition
(logical partition
container) selection (default
p): p partition number (1-4, Default 1): first sector
(2048-4194303, default 2048): last sector, +sectors or +size{K, M, G, T, P} (2048-4194303, default 4194303): +300M

Created a new partition 1 of type "Linux" with a size of 300 MiB.

Command (type m for help): t
Partition 1 selected
Hex code (enter L to list all codes): b
```

```
Changed the type of partition "Linux" to "W95 FAT32".

Command (enter m for help): n
partition
      type p primary partition (1 primary partition, 0 extended
      partition, 3 free) e extended partition
(logical partition
container) selection (default
p): p partition number (2-4, Default 2): first sector
(616448-4194303, default 616448): last sector, +sectors or +size{K, M, G, T, P} (616448-4194303, default 4194303):

Created a new partition 2 of type "Linux" with a size of 1.7 GiB.

Command (type m for help): p
Disk coolpi.img: 2 GiB, 2147483648 bytes, 4194304 sector units: sector / 1 *
512 = 512 bytes sector size (logical/
physical): 512 bytes / 512 bytes
I/O size (min/best): 512 bytes / 512 bytes Disk
label type: dos Disk
identifier: 0xeeeeb672

                      Start Start End Sector Size Id Type
device                             2048 616447 614400 300M b W95 FAT32
coolpi.img1 coolpi.img2 616448 4194303 3577856 1.7G 83 Linux

Command (type m for help): w Partition
table adjusted.
Synchronizing disks.

~/img_test$
```

b) Mount partition format write

```
~/img_test$ losetup -f /dev/loop18
~/img_test$
sudo losetup /dev/loop18 coolpi.img [sudo] xxx 's password: ~/
 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) Discarding device blocks: Finished creating filesystem with 447232 blocks (4k
each ) and
112000 inodes Filesystem UUID: c4c8cda5-77ae-4872 The backup of
the -9f50-4d4c20cf048f superblock is
stored in the following
blocks:


                32768, 98304, 163840, 229376, 294912
```

```
Allocating group table:
Done Writing to inode table:
Done creating log (8192 blocks)
Done writing superblock and file system account statistics: Done

~/img_test$ sudo e2label /dev/mapper/loop18p2 writable ~/img_test$ ~/
img_test$
sudo mount /dev/mapper/loop18p1 /mnt/ pre-prepare cmdline.txt
config.txt initrd.img modules.tar.gz rk3588s -cp4.dtb vmlinuz copy files to mount directory /mnt ~/img_test$
sudo umount /mnt/ ~/
img_test$ sudo mount /dev/mapper/
loop18p2 /mnt/ pre-prepare rootfs.tar.gz extract root file system
to Mount directory /mnt
Please pay attention to
decompress the driver package to /mnt/usr/
lib at the same time ~ /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$
Congratulations on completing the image creation!
```

At this point, you can insert a USB flash drive or a mobile hard disk, and use the dd method to flash the new firmware.

### 3.3 Mirror backup

The CoolPi 4B system can perform backup extraction or recreate a new image at any time. The method is relatively simple, as described below.

a) Insert the system disk into the linux system (Debian and Ubuntu are acceptable); b) Mount the two

partitions of the system disk, which will be automatically mounted by the general system, please confirm the corresponding mounting

directory; c) Open the terminal command line and copy the system -boot partition directory files are backed up to a local directory (such as /optÿÿ

d) Open the terminal command line, switch to the root authority, cd to the writable directory, and execute the compression command

command tar -czpvf /opt/rootfs.tar.gz *

e) Execute sync to refresh disk writing;

So far, the system image backup has been completed, and a new image can be created according to the instructions in Section 3.2.