Chipsee

Industrial PC

Buildroot Linux Qt 5.15 OS on RK3568 User Manual

For RK3568 Products

Content can change at anytime, check our website for latest information of this product.
www.chipsee.com

Contents

Buildroot Linux Qt 5.15 OS

Buildroot Linux Qt 5.15 OS on RK3568 User Manual

Flashing OS Image

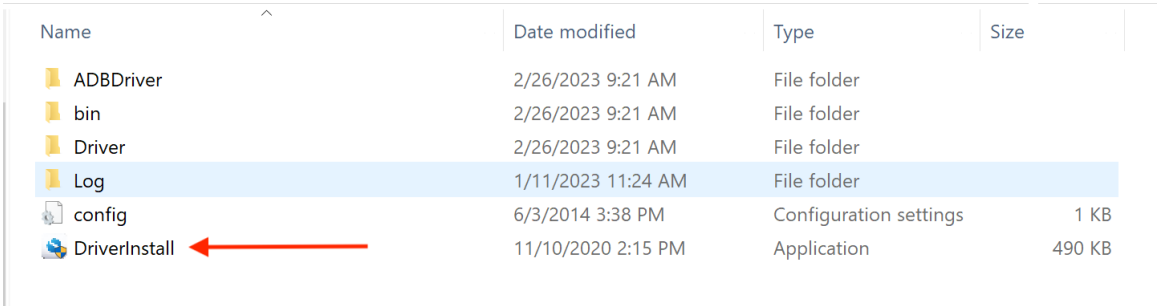
Download Required Tools

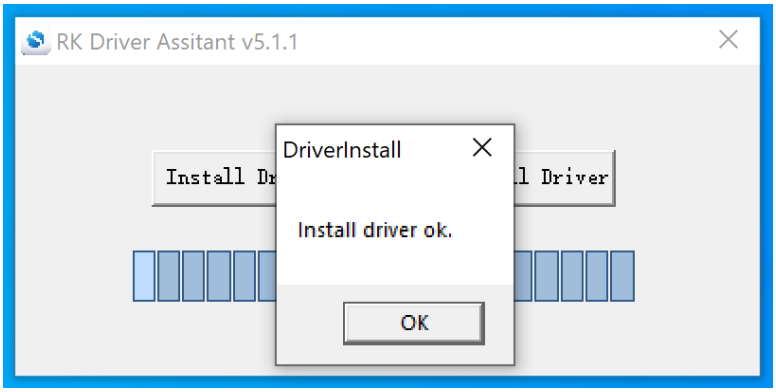
If you want a fresh OS, you can flash your Chipsee industrial PC.

You need two tools to flash the Buildroot Linux Qt 5.15 OS image to the RK3568 PC. The first is *DriverAssistant_v5.1.1*, the second is *RKDevTool_v2.93*, you can download all of them here.

These tools are Windows executables, please execute them on a Windows machine.

If you've been using a prior version of *DriverAssistant*, click uninstall before installing *DriverAssistant_v5.1.1*.



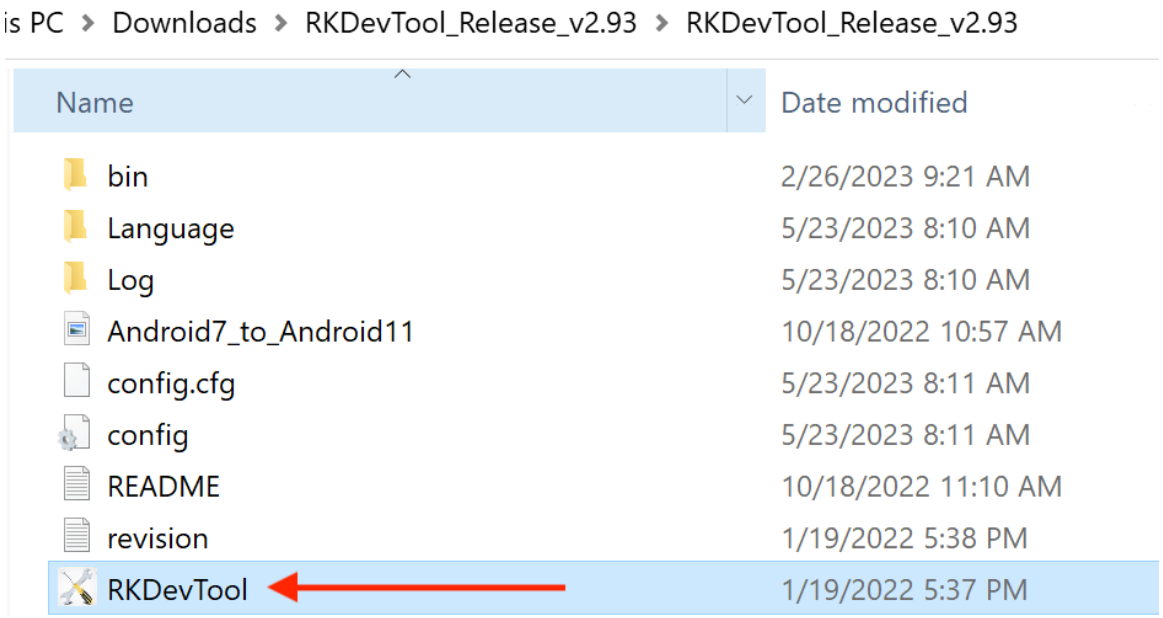


Download Prebuilt OS Images

If you haven't downloaded the prebuilt OS images, you can find one here.

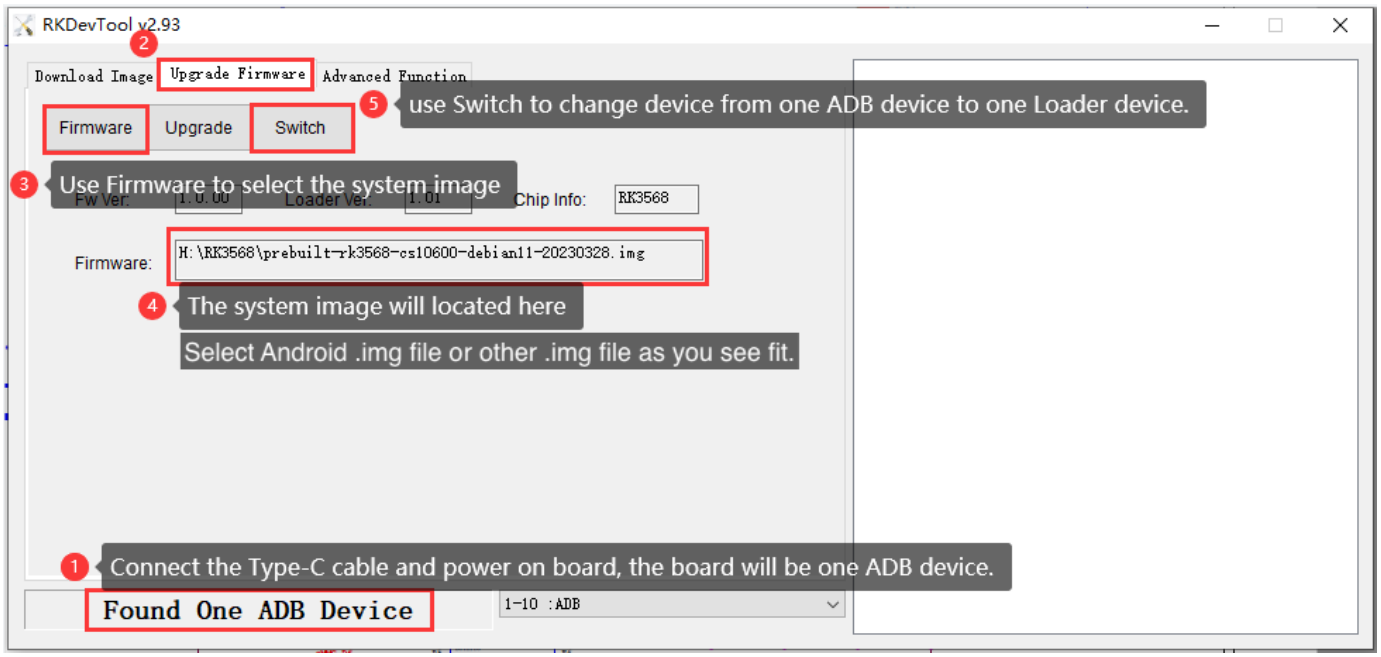
Start Flashing

After installing the DriverAssistant, you can now start to flash an OS image to the RK3568 board with *RKDevTool*. Double click the program to start flashing. The tool has English and Chinese language support.



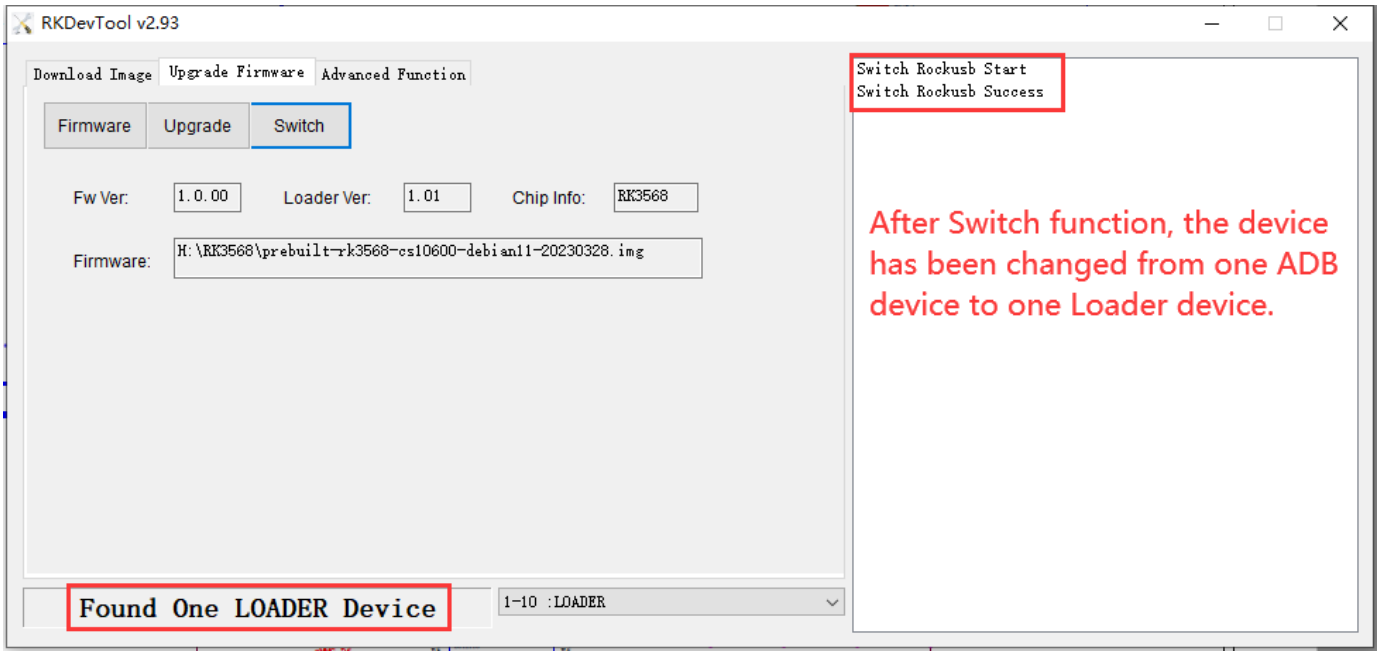
STEP 1:

1. Connect the Type-C cable and power on the board. (If unexpected messages occur at any of the following steps, try plugging the Type-C cable again.)
2. Click **Upgrade Firmware** tab.
3. Click **Firmware** button to select a .img Buildroot Linux Qt 5.15 image file. The screenshots show a debian11 img file is selected, but this is applicable to other OSes as well.



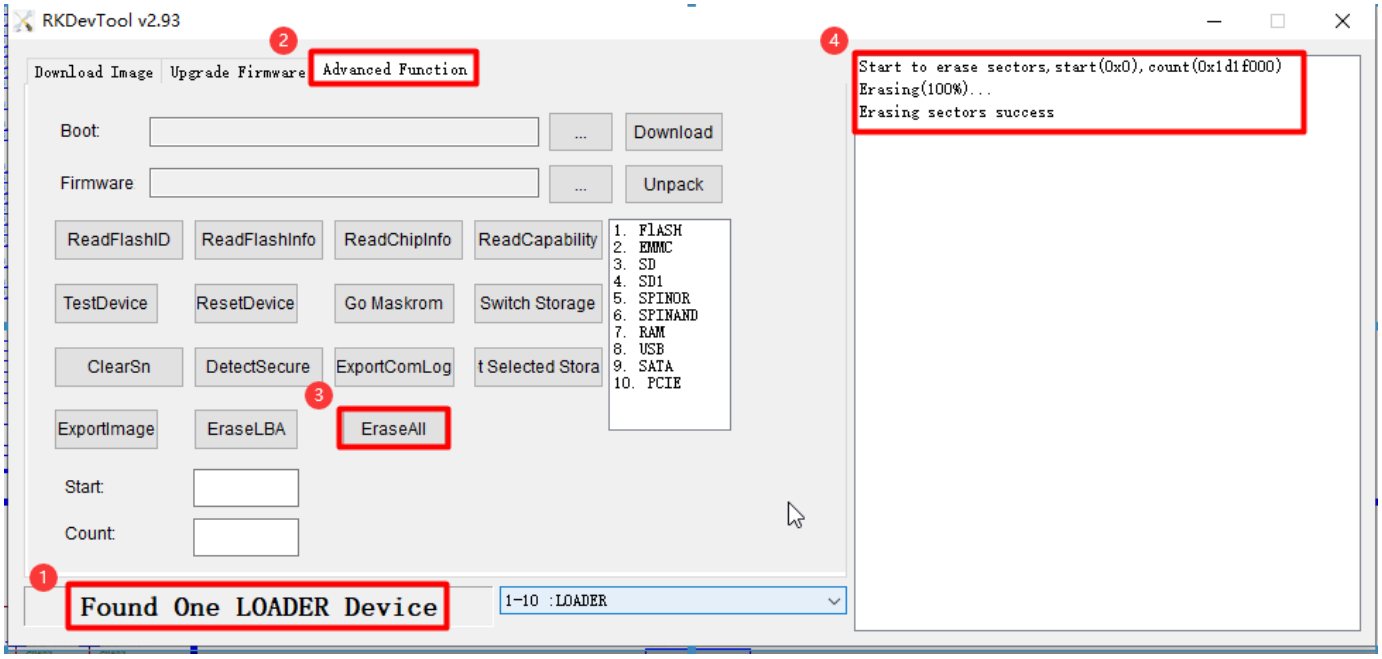
STEP 2:

1. Click **Switch** button to switch the device to a Loader device.



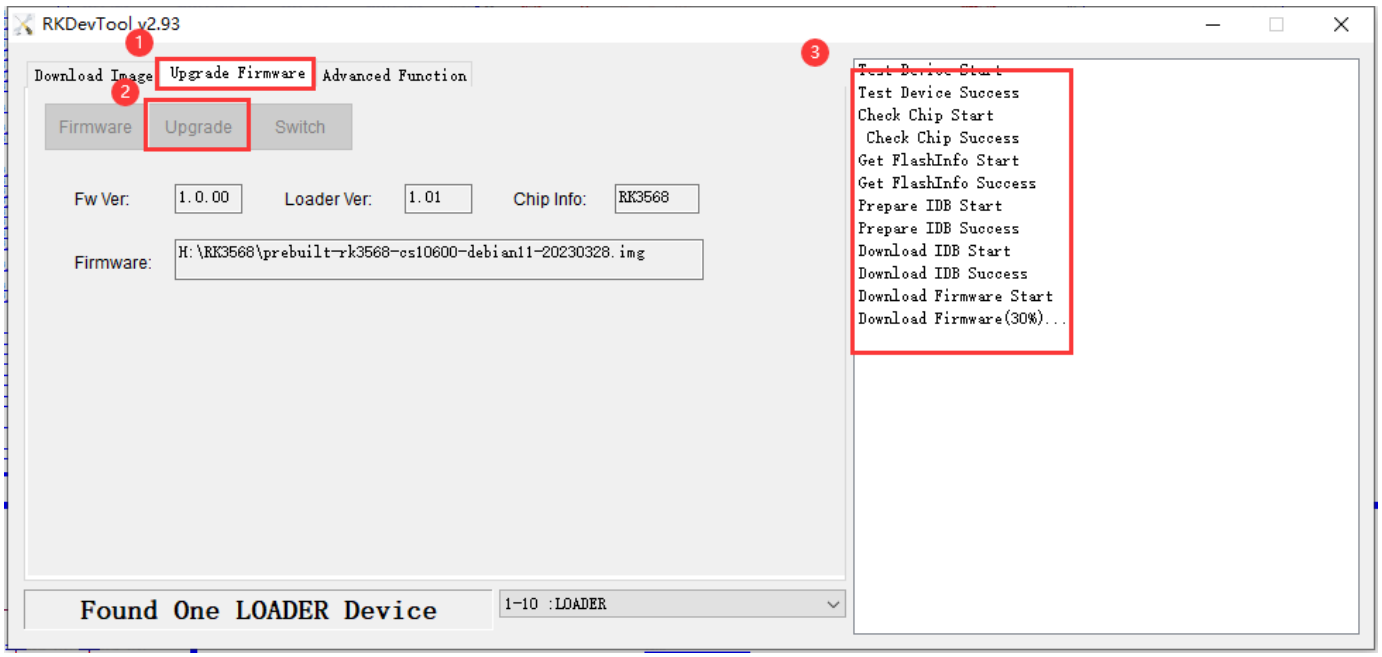
STEP 3:

1. You should see “Found One LOADER Device”.
2. Click **Advanced Function** tab.
3. Click **EraseAll** button.
4. You should see “Erasing sectors success” on the right side logs.



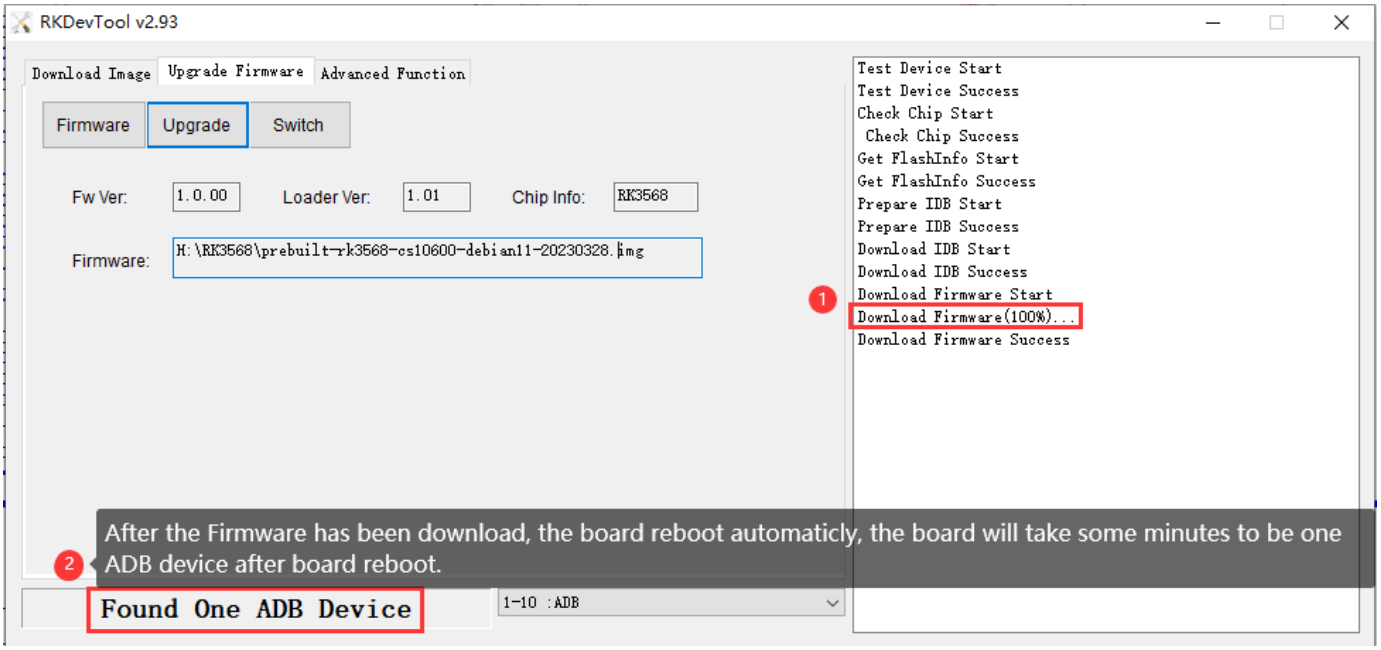
STEP 4:

1. Click **Upgrade Firmware** tab.
2. Click **Upgrade** button.
3. You should see Download Firmware progress on the right side logs.



STEP 5:

1. After the download firmware progress goes to 100%, the board reboots itself automatically.
2. After a few minutes, you should see “Found One ADB Device”.
3. Now your new OS is ready for use.



Video Tutorial for Flashing OS

Method 1: LOADER Mode

Here is a video tutorial we made demonstrating the OS installation process described above in Windows in the **LOADER** mode: https://www.youtube.com/watch?v=ufKDCJ1hpf4

The approach in the video above works best for devices that are still able to boot into the desktop, and when your workstation is a Windows machine. However, if you do not have a *Windows* machine in the room, you can use the approach below to flash an OS, in a Linux or Mac.

Method 2: MASKROM Mode

Apart from flashing in **LOADER** mode, when you're working on a *Linux(X86_64)* workstation or *MacOS(Intel and Apple Silicon)* machine, you can use another approach: **MASKROM** mode, to flash the OS. There is a PROG button on the Chipsee industrial PC, you can press the button before powering up the device, power up and hold the PROG button for 2~4 seconds, then use a X86_64/darwin_64 *upgrade_tool* program in the command line to flash the OS, here is a video we made to teach you how to do that in two minutes: https://www.youtube.com/watch?v=TDIHoQ9AuX4

The approach described in the second video works best for devices that are "bricked" (compared to the first approach), it can help rescue your device if your operating system is broken and cannot boot into the desktop. Even if your device is still functional, you can also use this approach to flash an OS, it works in Windows, Linux as well as MacOS.

The command used in the videos are:

For **Linux** workstation:

```
sudo ./upgrade_tool_linux_x86-64 ld # to list device
sudo ./upgrade_tool_linux_x86-64 uf ./prebuilt-rk3568-xxx.img # to upload
firmware
```

For **MacOS**:

```
./upgrade_tool_darwin64 ld # to list device
./upgrade_tool_darwin64 uf ./prebuilt-rk3568-xxx.img # to upload firmware
```

And that's all it takes.

The **upgrade_tool** used in the video can be download at:

1. `upgrade_tool_x86-64 (For Linux x86)`
2. `upgrade_tool_darwin64 (For MacOS Intel & Apple Silicon)`

We've tested that the MacOS upgrade_tool can execute in M1/Apple Silicon Macs, but you will need to install Rosetta to run this program. For Intel Macs, you do not need Rosetta, you can execute the binary program directly in your terminal.

Also, as noted in the video, do use a **absolute path** to the firmware file or **"./prebuilt-rk3568-xxx.img"**, rather than a relative path (e.g. your current directory contains the img file, and you directly use "upgrad_tool uf prebuilt-rk3568-xxx.img", this will not work). And make sure to use *sudo* in Linux.

Backup Your OS Image For Bulk Installation

If you have finished developing your software, and plan to “copy” the whole system to many other Chipsee industrial PCs, you can backup the OS to an image file, just like the **.img** file you downloaded from Chipsee, or the OS we installed in the factory for you before shipping. And then you can flash it to many more devices.

Prepare for backup

We will use **SDDiskTool** to flash a bootable SD card, let your Chipsee PC boot from this SD card, then use this system to backup your OS image (the whole content on eMMC rootfs partition). You will need:

- SDDiskTool (Click to download).
- 16GB or larger micro SD card.
- SD card reader (to be used on your HOST PC).
- A Windows PC to run the SDDiskTool.
- A (X86 or X86_64) Linux HOST PC or virtual machine to make a new img file (make sure there is 25GB or more free space on the disk for the following process).
- Two Chipsee prebuilt images, one is the image that you are developing your software on, the other is a prebuilt-xxx-sd-xx.img, if you cannot find the prebuilt-xxx-sd-xx.img of your device, you can use just the prebuilt-xxx-emmc-xx.img temporarily, we will release the sd image later.

> **Note**
>
> More on the two prebuilt images: the core idea of backup is to “swap” your data and the prebuilt data. So we will need to download a prebuilt image that you’re developing your software on (the OS image that you’re currently using on the Chipsee PC), unpack that image, swap the data, then repack the image.

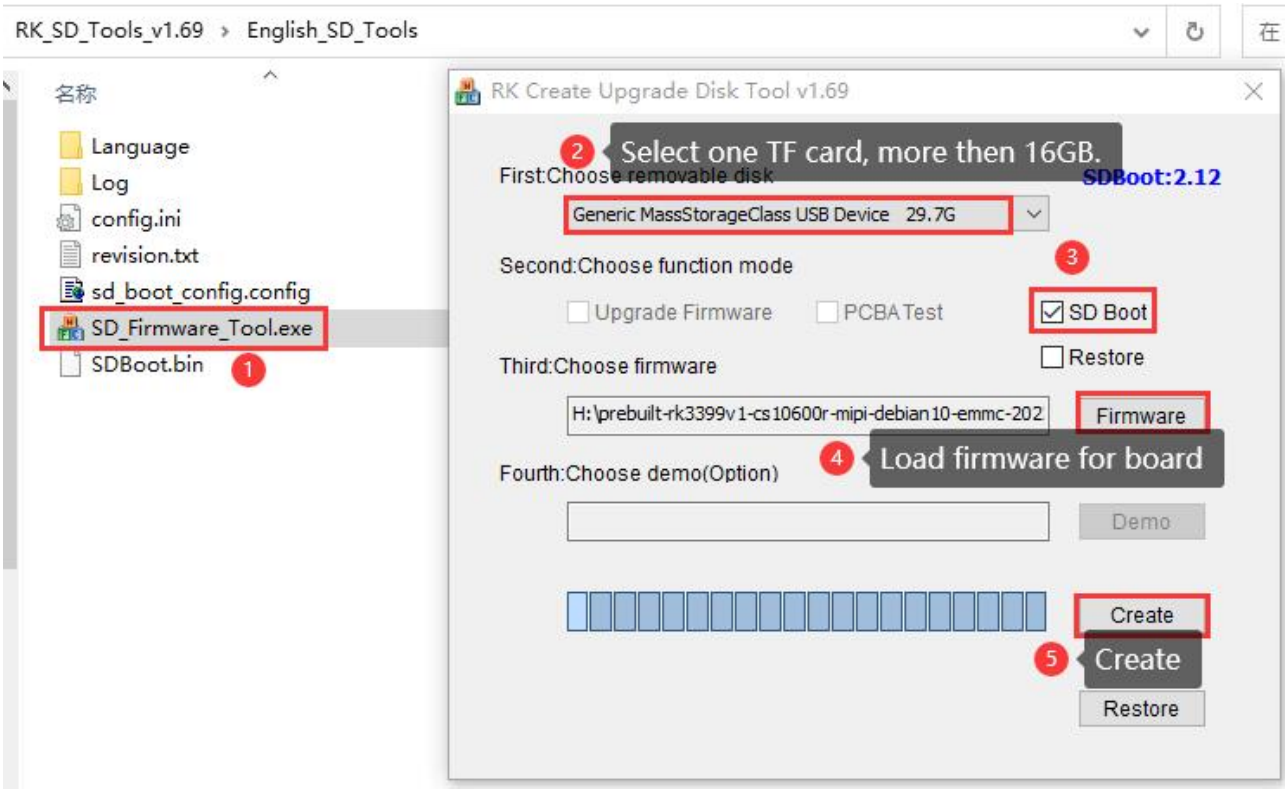
> **Note**
>
> As for the second prebuilt-xxx-sd-xx.img, we use it to make a bootable SD card (imagine the old time people use a WinPE USB stick to boot and backup Windows!).

Prepare a Bootable SD Card

On your Windows PC, we open SD_Firmware_Tool.exe to process 1,2,3,4,5 steps to create a bootable SD card.

You need to download the Chipsee prebuilt image as we mentioned earlier. Find the one that fits your screen size in Chipsee prebuilt images page.

Once the SD card is flashed, Windows will show a warning to let you format the unrecognized partition, **ignore or cancel** it because the SDDiskTool creates some partitions that Windows doesn't recognize.



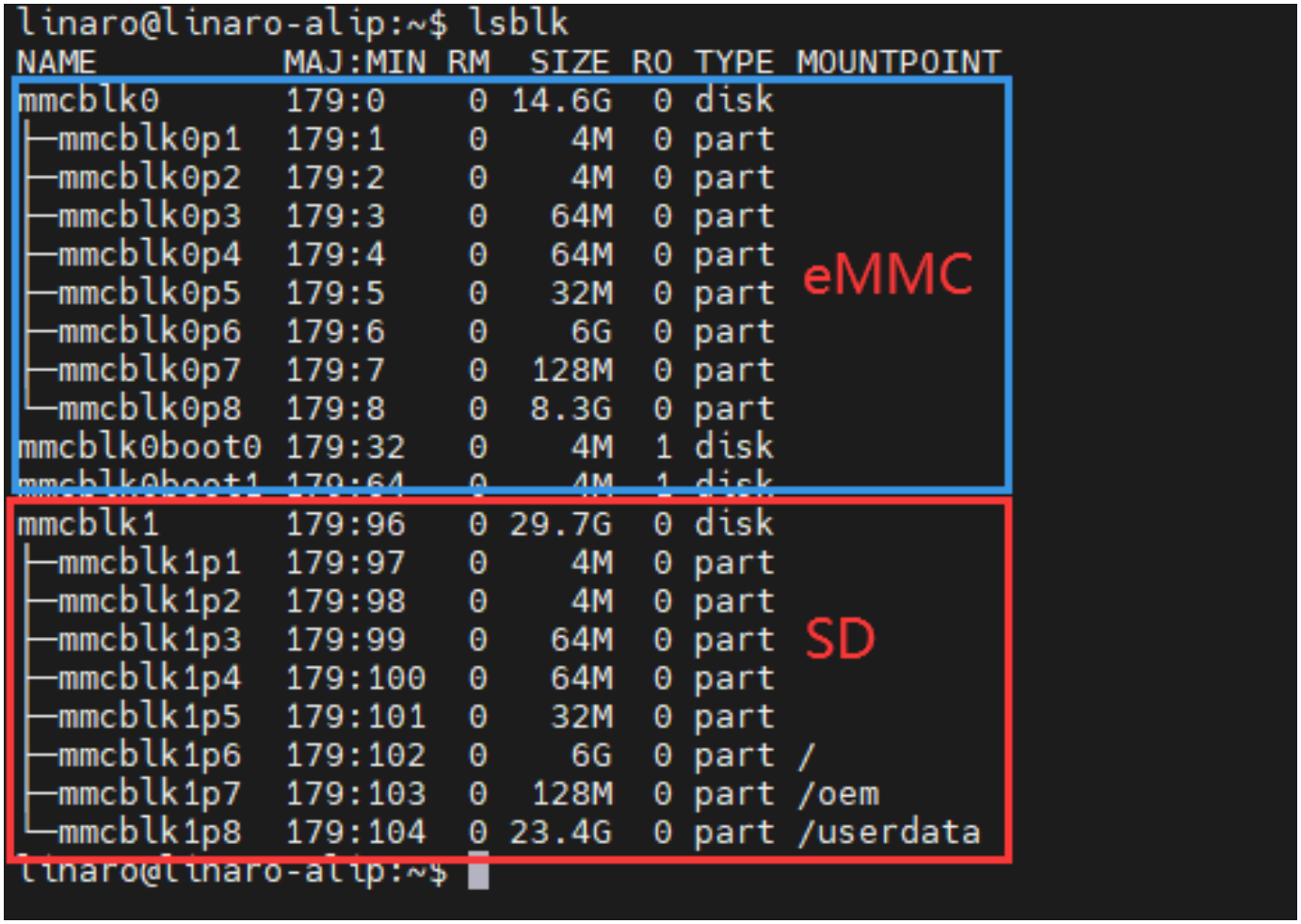
Follow the 5 steps on SDDisktool

Backup Your eMMC

Insert this SD card into the SD slot of the Chipsee PC and power it on, the Chipsee PC will boot into the system on the SD card, we can use this system to backup the whole contents on eMMC rootfs partitions.

Use the way you like to execute the following commands, for example, serial debug or ssh. You can connect a keyboard and mouse to the Chipsee device and run them in the command line as well.

The eMMC rootfs partition is **/dev/mmcblk0p6**. We will backup the contents in **/dev/mmcblk0p6**.



eMMC rootfs partition is **/dev/mmcblk0p6**

```
$ sudo su
# export ROOTFS_DEV=/dev/mmcblk0p6
# mkdir /mnt/backuprootfs
# mount $ROOTFS_DEV /mnt/backuprootfs/
# cd /mnt/

// sync would take an hour or more depending on the files in your system
# tar --numeric-owner -jcvpf backuprootfs.tar.bz2 backuprootfs && sync
# umount /mnt/backuprootfs
```

Now we have obtained the backup rootfs **backuprootfs.tar.bz2** in the SD card partition

Generate New Image File

Poweroff the Chipsee PC. Put the SD card into your Linux HOST PC (or virtual machine).

You should find a **/dev/sdX** in your Linux system, for example **/dev/sdb**, which is this SD card, **you should use your actual /dev/sdX here**, if you don't know which sdX is it, check with *df -h* and see which one's size is most likely your SD card.

Now we mount /dev/sdb6 to find backuprootfs.tar.bz2

```
# mount /dev/sdb6 /mnt/
```

It will be in **/mnt/mnt/backuprootfs.tar.bz2**, we will copy it out to our Linux PC later.

Run the following command to generate a new *.img* file. Make sure you have at least 25GB free space on your Linux PC, the process produces a lot of intermediate files.

```
$ sudo su
# git clone https://gitee.com/chipsee_admin/rk_pack_tools.git
# cd rk_pack_tools
# git checkout r510-rk3568

// copy the Chipsee prebuilt img file to this directory
# cp prebuilt-xxx.img .
# ./cs-unpack.sh prebuilt-xxx.img

// copy your backup rootfs from SD card to this directory
# cp /mnt/mnt/backuprootfs.tar.bz2 .

// generate rootfs.img file from backuprootfs.tar.bz2
# ./cs-mkrootfs.sh

// generate new img file
# ./cs-pack.sh prebuilt-new-xxx.img
```

Warning

If you see *checksum miss match error* or *Error:<AddFile> write file failed,err=28*, check your harddisk and make sure you have enough free space.

Now you have obtained your new img file prebuilt-new-xxx.img in the current folder, use this img file to flash other devices.

Notice

(This Buildroot Linux Qt 5.15 OS User Manual is under active development as of Feb 26th, 2024)

Disclaimer

This document is provided strictly for informational purposes. Its contents are subject to change without notice. Chipsee assumes no responsibility for any errors that may occur in this document. Furthermore, Chipsee reserves the right to alter the hardware, software, and/or specifications set forth herein at any time without prior notice and undertakes no obligation to update the information contained in this document.

While every effort has been made to ensure the accuracy of the information contained herein, this document is not guaranteed to be error-free. Further, it does not offer any warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document.

Despite our best efforts to maintain the accuracy of the information in this document, we assume no responsibility for errors or omissions, nor for damages resulting from the use of the information herein. Please note that Chipsee products are not authorized for use as critical components in life support devices or systems.

Technical Support

If you encounter any difficulties or have questions related to this document, we encourage you to refer to our other documentation for potential solutions. If you cannot find the solution you're looking for, feel free to contact us. Please email Chipsee Technical Support at **support@chipsee.com**, providing all relevant information. We value your queries and suggestions and are committed to providing you with the assistance you require.