Chipsee

Industrial PC

Debian 11 OS on RK3568 User Manual

For RK3568 Products

Content can change at anytime, check documentation website for latest information.
www.chipsee.com

Contents

Debian 11 OS 4

1. System Information 6

2. Prepare for Developing 7

2.1. Prepare the Hardware 7

2.2. Prepare the Software 7

3. Connecting to the Chipsee industrial PC 8

3.1. Serial Port Debugging 9

3.2. SSH Connect 12

3.3. Connect with VNC Remote Desktop 15

4. Hardware Resources in the OS 17

4.1. Network 18

4.2. Serial Port RS232 and RS485 19

4.3. GPIO 21

4.4. BUZZER 22

4.5. Backlight 22

4.6. CAN Bus 23

5. FAQ 25

6. Flashing OS Image 27

6.1. Download Required Tools 27

6.2. Download Prebuilt OS Images 27

6.3. Start Flashing 27

6.4. Video Tutorial for Flashing OS 31

6.4.1. Method 1: LOADER Mode 31

6.4.2. Method 2: MASKROM Mode 31

7. Backup Your OS Image For Bulk Installation 33

7.1. Prepare for backup 33

7.2. Prepare a Bootable SD Card 33

7.3. Backup Your eMMC 34

7.4. Generate New Image File 35

8. Disclaimer 37

9. Technical Support 37

Debian 11 OS

Debian 11 OS on RK3568 User Manual

This is the software manual for RK3568 Chipsee industrial PC. If you've never developed on this hardware with a Debian 11 OS, this manual can get you started quickly.

Supported Chipsee PCs: all Chipsee RK3568 based industrial PCs, including but not limited to:

- CS12720-RK3568-050P
- CS10600-RK3568-070P
- CS12800-RK3568-101P
- CS10768-RK3568-121P
- CS19108-RK3568-133P
- CS10768-RK3568-150P
- CS19108-RK3568-156P
- CS12102-RK3568-170P
- CS19108-RK3568-185P
- CS12102-RK3568-190P
- CS12800-RK3568-215P
- CS19108-RK3568-236P
- CS-RK3568-BOX

When you develop software for the Chipsee industrial PC, you can open the hardware document beside this software document, to aid you in wiring your devices.

In this document, main topics are:

- How to connect to the hardware from your workstation.
- How to use the hardware resources such as RS232, RS485 and GPIO, etc.
- How to install a new operating system.

System Information

Out of Box System

	Description
Kernel	5.10.110
Bootloader	U-Boot 2017.09
OS	Debian GNU/Linux 11 (bullseye) aarch64
Python	3.9.2
Qt	5.15.2, QMake 3.1
GCC	10.2.1
username/password	[linaro/linaro]
Window Manager	Xfwm4
Desktop Environment	Xfce 4.16

Prepare for Developing

To get started, you first need to power on the Chipsee industrial PC, then you may want to connect to this PC from your own laptop or computer to control it. Let's prepare some hardware and software to start developing.

Prepare the Hardware

To power on and connect to Chipsee industrial PC, we need:

1. A power adaptor. For products with a screen of 7 inch and below, a power adaptor with 6V ~ 36V DC output is required; for 10.1 inch and above products, you need one with 12V ~ 36V.
2. A USB to serial cable (if you need serial debug).
3. An Ethernet cable (if you want to SSH into the Chipsee industrial PC). You may also use WiFi if your Chipsee industrial PC supports WiFi, in this case you don't need the Ethernet cable.
4. A USB type-C cable (if you want to flash a new OS).

Prepare the Software

Thanks to the Debian 11 OS, developing on Chipsee industrial PC isn't really different from developing on any other PCs, you can use any developer software you're comfortable with, with only one exception for flashing OS.

1. To flash a new operating system, you need RKDevtool, you can refer to the Flashing OS Image section.

   The software listed below are not mandatory, they're recommendations because we find them easy to use:

2. To SSH into a Chipsee industrial PC, you may find **PuTTY** on Windows handy; for Linux and macOS users, a terminal app should come with your OS out of box, like **Terminal/ iTerm2** on macOS and **xterm** on Linux.
3. To use a remote desktop, you can download **VNC Viewer** on your laptop or PC.

Connecting to the Chipsee industrial PC

When a power supply is plugged, the Chipsee industrial PC should boot itself automatically, next let's connect to it from our laptop or computer.

There are three different approaches: serial port debugging, SSH with Ethernet/WiFi and remote desktop with VNC.

Serial Port Debugging

[*Cheatsheet for experienced developer*: username: **linaro**, password **linaro**.]

Physically, your laptop connects to your Chipsee industrial PC with a USB to serial cable. To connect to the Chipsee industrial PC in the terminal program, you first need to know what port your laptop or computer is using in the device tree.

The best way to find this is check what **tty** devices you have before connecting the Chipsee industrial PC to your computer, then find out what is changed after you connect it to your laptop or PC.

Take macOS as an example:

```
# on your laptop
ls /dev/tty.*
```

You may see:

```
# ➜  ~ ls /dev/tty.*
# /dev/tty.Bluetooth-Incoming-Port /dev/tty.wlan-debug
# ➜  ~
```

It means there is a Bluetooth and a wlan port **before** you plug the Chipsee industrial PC to your Macbook.

Now you can plug the USB end to your laptop or computer, connect the serial port end to your Chipsee industrial PC's RS232 port. For different products the pins may vary, you can check the pin definition table to find out which pins are the RS232 TX, RX, GND for serial debugging.

And let's check again what has changed **after** we connect the two:

```
# on your laptop, check again
ls /dev/tty.*
```

You might see:

```
# ➜  ~ ls /dev/tty.*
# /dev/tty.Bluetooth-Incoming-Port /dev/tty.wlan-debug /dev/tty.usbmodem2812
# ➜  ~
```

You can tell the last */dev/tty.usbmodem2812* appeared right after we plug in the cable, and this is the serial port that connects to your Chipsee industrial PC with your USB to serial cable.

For Windows and Linux users the approach is almost the same, on Windows you may check the Microsoft Windows Device Manager. And check which COM port appears/disappears when you plug/unplug the USB to serial cable.

Now that we know the port on our OS, let's use the terminal program to connect to it.

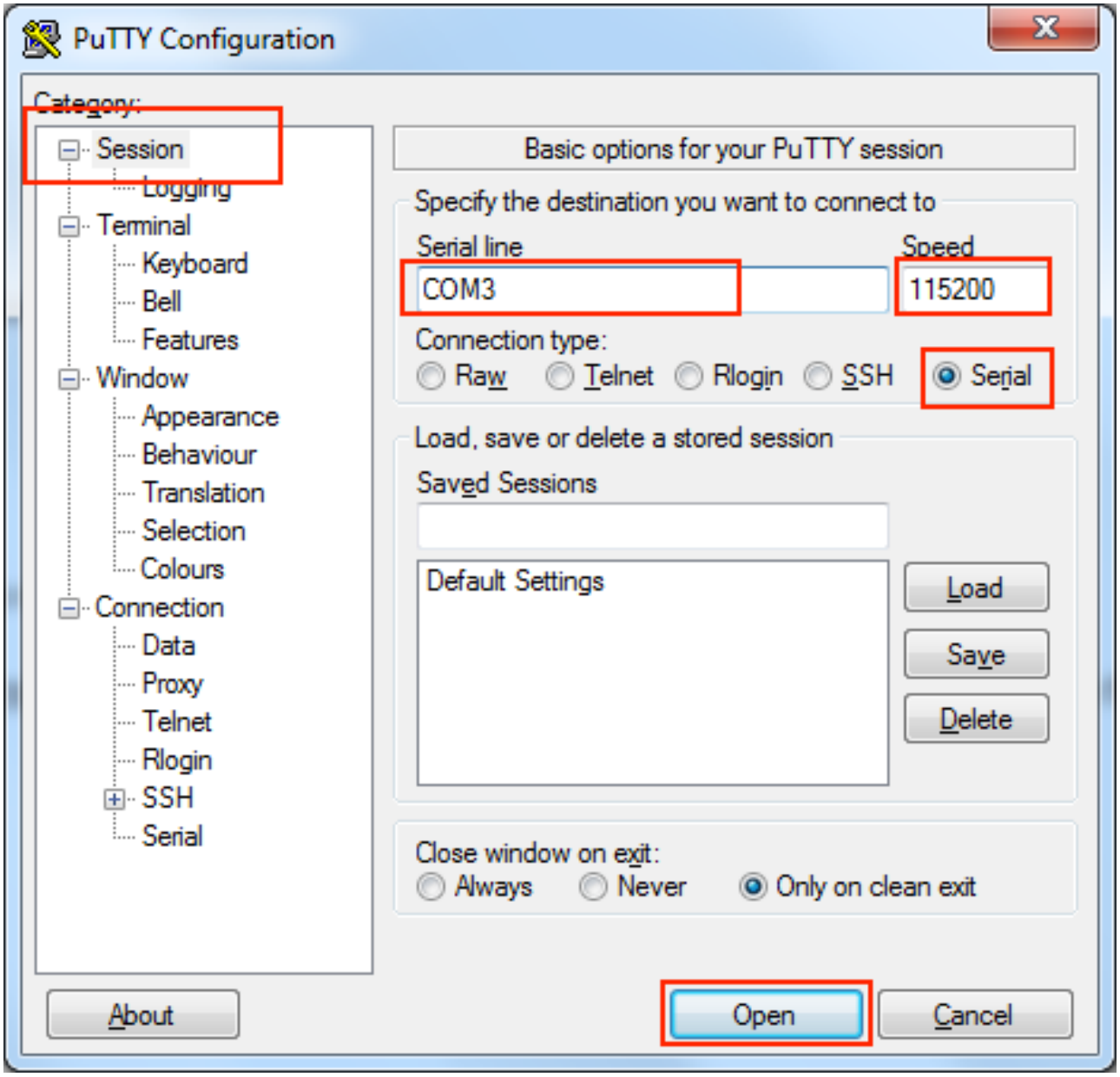
Take macOS as an example:

```
# on your laptop
screen /dev/tty.usbmodem2812 115200
```

Use the **screen** command followed by the port name (in our case usbmodem2812), and a baud rate to connect to this serial port.

After a few seconds, in your terminal you should see some text, asking for a username and password. In our RK3568 based system, we should input *linaro* for the username, and also *linaro* for the password.

For Windows users who use PuTTY, you should select Session, choose Serial in the radio buttons, and choose the COM port you found in the previous procedure, say COM3, and choose 115200 as baud rate, then click Open. When it asks for a username and password, give **linaro** for username and also **linaro** for password.



Input your-com in the Serial Line field, in our case COM3

That's it for connecting the Chipsee industrial PC to your machine with a serial debug port.

SSH Connect

[*Cheatsheet for experienced developer*: username: **linaro**, password: **linaro**, IP check: **sudo ifconfig**.]

If you have a network connecting the Chipsee industrial PC and your laptop or computer, you may find SSH comes handy.

To connect to your Chipsee industrial PC with SSH, you need to plug an Ethernet cable to the RJ45 port, or connect through WiFi if your product supports WiFi.

You will need to know the IP address of your Chipsee industrial PC to use SSH, to find out which, you can use a mouse and keyboard temporarily and type:

```
# type this in your Chipsee PC's xterm program to find out the IP address of this device:
$ sudo ifconfig
```

You should see a bunch of outputs, look for the keyword **eth** and **inet**, they will turn out to be your IP address, for example:

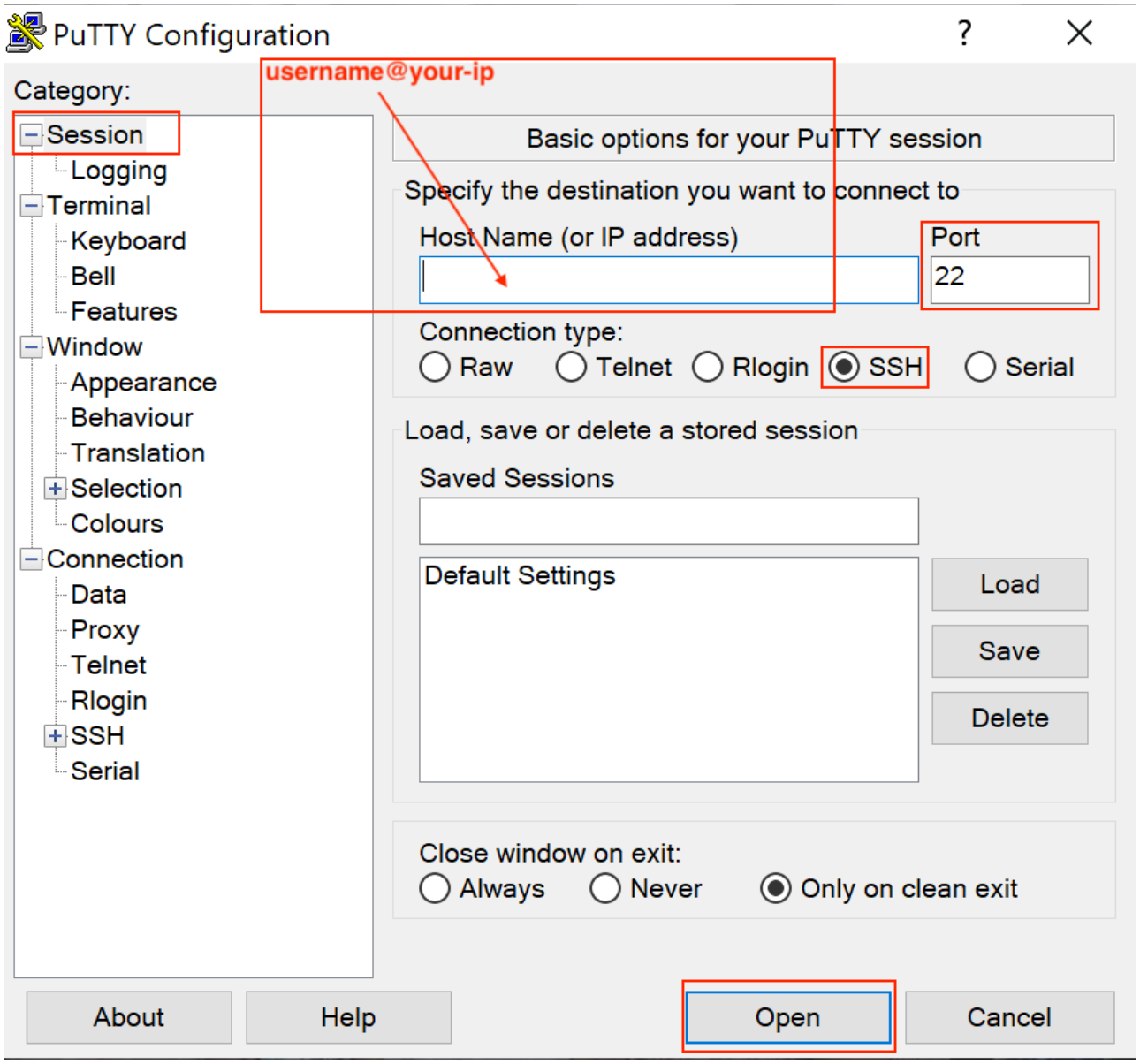
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
     inet 192.168.6.134  netmask 255.255.255.0  broadcast 192.168.6.255
     inet6 fe80::c3be:d086:e810:ee8e  prefixlen 64  scopeid 0x20<link>
     ......

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
     inet 127.0.0.1  netmask 255.0.0.0
     inet6 ::1  prefixlen 128  scopeid 0x10<host>
     ......

p2p0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
     ......
```

In our example, the IP address is **192.168.6.134**.

Now that we have the IP of our Chipsee industrial PC, we can detach the mouse and keyboard, and go back to our laptop or computer to proceed.

We can now open a terminal app, like PuTTY on Windows, or Terminal/xterm on macOS/Linux.

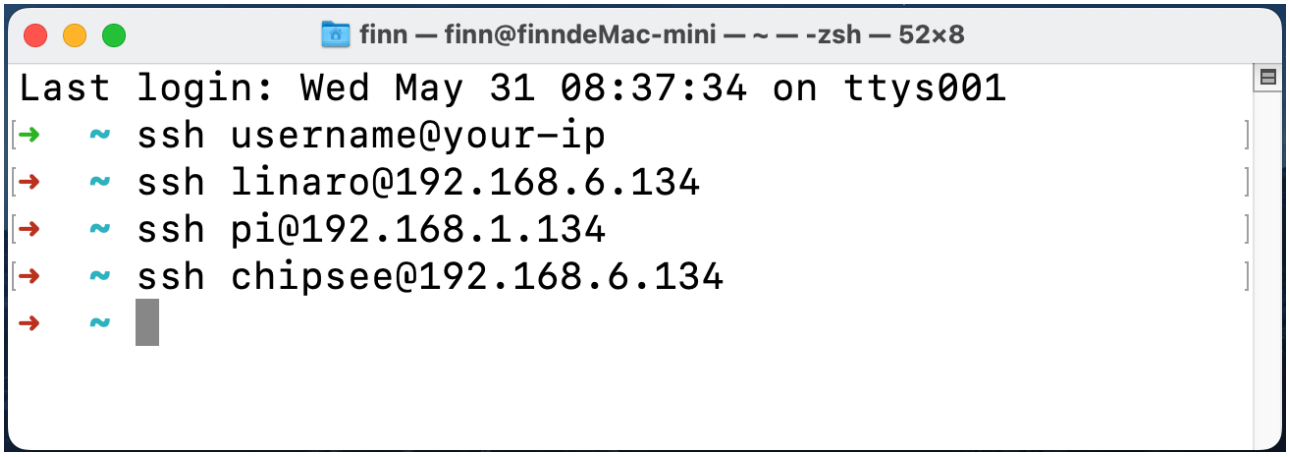
For Windows users using PuTTY, you can choose Session, input **username@your-ip** (in our case *linaro@192.168.6.134*, yours should be different) in the Host Name field. Port Number should remain 22, choose SSH as the connection type in the radio buttons, click “Open”. When the prompt asks for the password, type **linaro** and hit enter.



Input username@your-ip in the Host Name field, in our case linaro@192.168.6.134

For macOS/Linux, in your terminal program, type:

```
# change 192.168.6.134 to your Chipsee PC's IP we got earlier
ssh linaro@your-ip
# in our case, we are
ssh linaro@192.168.6.134
```



Change the command to ssh username@your-ip, in our case ssh linaro@192.168.6.134

When the prompt asks for your password, input **linaro**, the password will not show up on the screen when you type it, it's OK, just hit enter when you finish typing.

For both Windows and Linux/macOS users, we should have successfully ssh'd into our Chipsee industrial PC now.

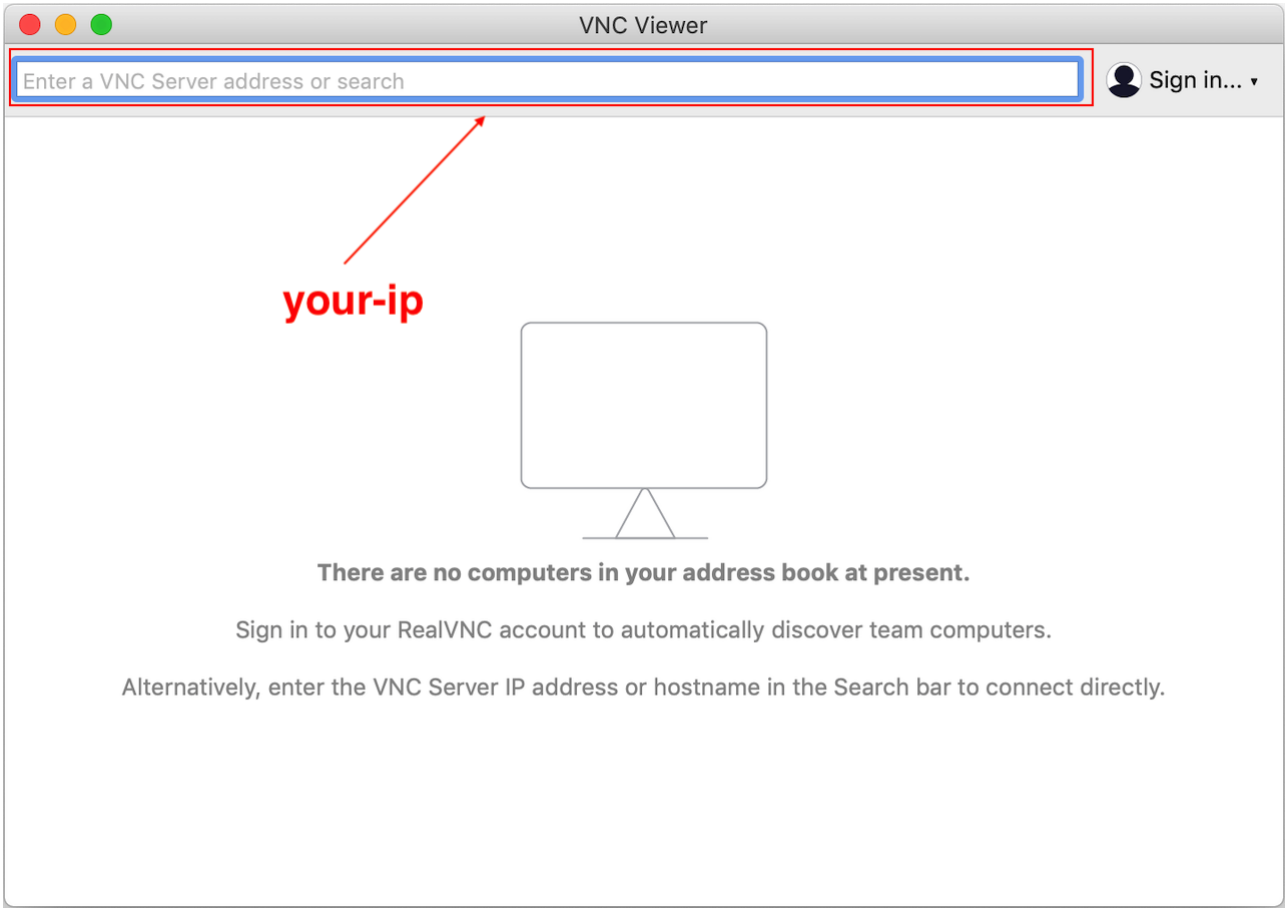
Connect with VNC Remote Desktop

[*Cheatsheet for experienced developer*: username: **linaro**, password: **linaro**, IP check: **sudo ifconfig**, VNC port: **5900** (default port).]

We can have a graphical user interface if we use VNC to connect to our Chipsee industrial PC. This is handy if you're developing an HMI and want to test from your own laptop or computer. Here is how:

First please refer to the previous section to learn how to find out your Chipsee industrial PC's IP address. In our case we get **192.168.6.134**.

Then, for either Windows, macOS or Linux users, please open the VNC Viewer software on your laptop or PC. In the input area, which has a placeholder "Enter a VNC server address or search", you should input the IP address we obtained earlier: **your-ip**, in our case: **192.168.6.134** (yours should be different), and hit enter.



Change the input to your-ip, in our case 192.168.6.134

VNC Viewer might ask for your credentials, you should input our username and password, in our case: username: **linaro**, password: **linaro**. And then continue, VNC Viewer might tell you the connection is not encrypted, it's OK if you're in your local network.

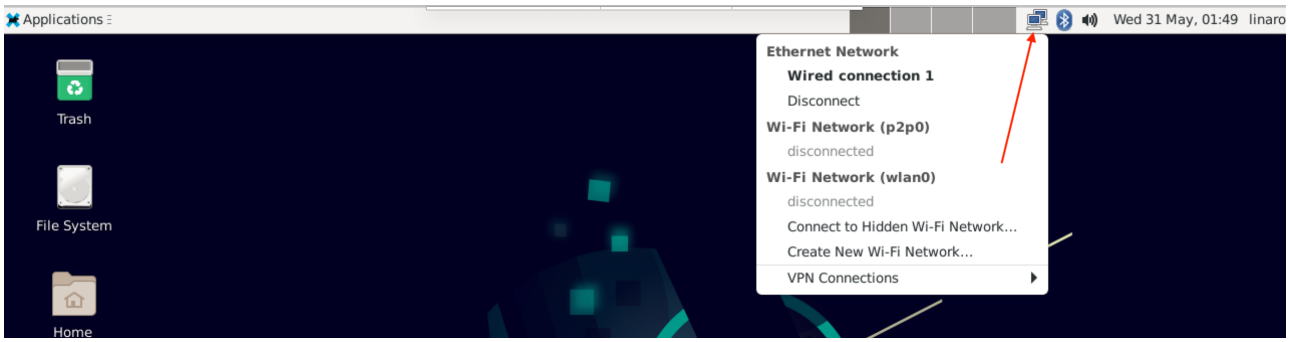
Now we should see our Chipsee industrial PC's desktop GUI in the VNC Viewer software.

Hardware Resources in the OS

Now that you have successfully booted the Chipsee industrial PC and connected the Chipsee industrial PC to your laptop/computer, this section will tell you how to control this Chipsee industrial PC from its OS desktop itself, or from your PC.

Network

From the Chipsee industrial PC's desktop GUI, you can click the Network icon in the menu bar to set your network.



Network Settings

From the terminal, you can use **nmcli** to configure the network. You can refer to the documents of the **nmcli** program for more information. **nmcli** is not a Chipsee specific tool, it's widely used in Linux distributions.

```
$ nmcli -o

eth0: connected to Wired connection 1
     "eth0"
     ethernet (rk_gmac-dwmac), 42:0B:6F:67:B0:E3, hw, mtu 1500
     ip4 default
     inet4 192.168.6.134/24

p2p0: disconnected
        "p2p0"
        wifi (rtl8821cs), B6:6D:C2:13:A3:AE, hw, mtu 1500

wlan0: disconnected
        "wlan0"
        wifi (rtl8821cs), B4:6D:C2:13:A3:AE, hw, mtu 1500
```

Serial Port RS232 and RS485

The RK3568 based Chipsee industrial PC supports RS232 and RS485, here are the mapping from the port name to the system tree device:

5 inch product

Name	Node	Protocol
RS232_0	/dev/ttyS0	RS232
RS232_2	/dev/ttyFIQ0	RS232, Serial Debug
RS485_3	/dev/ttyS3	RS485
RS485_5	/dev/ttyS4	RS485

Table 186 RS232/485 for **5 inch** product(CS12720-RK3568-050P)

7 inch product and Box product

Name	Node	Protocol
RS232_0	/dev/ttyS0	RS232
RS232_2	/dev/ttyFIQ0	RS232, Serial Debug
RS485_3	/dev/ttyS3	RS485
RS485_4	/dev/ttyS4	RS485
RS485_5	/dev/ttyS5	RS485

Table 187 RS232/485 for **7 inch** product(CS10600-RK3568-070P) and **box** product(CS-RK3568-BOX)

10.1+ inch products

Name	Node	Protocol
RS232_0	/dev/ttyS0	RS232
RS232_2	/dev/ttyFIQ0	RS232, Serial Debug
RS485_3	/dev/ttyS3	RS485
RS485_4	/dev/ttyS4	RS485

Table 188 RS232/485 for **10.1+ inch** products(CS12800-RK3568-101P, CS10768-RK3568-121P, CS19108-RK3568-133P, CS10768-RK3568-150P, CS19108-RK3568-156P, CS12102-RK3568-170P, CS19108-RK3568-185P, CS12102-RK3568-190P, CS12800-RK3568-215P, CS19108-RK3568-236P)

The 120 Ohm match resistor is already mounted on the RS485 port. RS485 ports are half-duplex, the hardware can switch the Tx/Rx direction automatically. RS232 ports are full-duplex.

You can install **cutecom** to test the serial port:

```
$ sudo apt-get install cutecom
```

Only root user and use the serial port:

```
$ sudo cutecom
```

You can then use two Chipsee industrial PCs to test each other.

If you've got only one Chipsee industrial PC, you can test that with your computer: you can use a USB to serial cable, connect the USB end to your computer, and connect the serial end to any of the Chipsee industrial PC's serial port. You may also wish to install a UART/ COM Assistant software which has a GUI for testing.

If you're an experienced engineer, you can also use a programming language to test the serial ports, like C, C++, Python, Javascript. They have their libraries for controlling serial port devices.

GPIO

There are 8 GPIOs, 4 Output, and 4 Input, they are all isolated. You can control the output or input pin voltage by feeding the VDD_ISO suite voltage. The pin voltage should be from 5V to 24V. Refer to the tables below for a detailed port definition:

Function	Device Node
IN1	/dev/chipsee-gpio5
IN2	/dev/chipsee-gpio6
IN3	/dev/chipsee-gpio7
IN4	/dev/chipsee-gpio8
OUT1	/dev/chipsee-gpio1
OUT2	/dev/chipsee-gpio2
OUT3	/dev/chipsee-gpio3
OUT4	/dev/chipsee-gpio4

Table 189 GPIO Device Node

- **Set** *OUT1* **to high or low**

```
$ echo 1 > /dev/chipsee-gpio1    # set OUT1 to high

$ echo 0 > /dev/chipsee-gpio1    # set OUT1 to low
```

- **Get** *IN1* **value**

```
$ cat /dev/chipsee-gpio5    # value 1 indicates high, value 0 indicates
low
```

BUZZER

The Chipsee industrial PC has one buzzer. We have created one symbol link to */dev/buzzer* . You can control it as follows:

```
$ echo 1 > /dev/buzzer    # enable buzzer

$ echo 0 > /dev/buzzer    # disable buzzer
```

Backlight

We use one PWM to control the backlight of RK3568 boards, You can use the following commands:

- **Get the supported max brightness:**

  ```
  $ cat /sys/class/backlight/backlight/max_brightness
  ```

- **Get the current brightness:**

  ```
  $ cat /sys/class/backlight/backlight/actual_brightness
  ```

- **Set brightness:**

  ```
  # might require write permission to the brightness file if you're not root
  $ sudo chmod a+w /sys/class/backlight/backlight/brightness
  # then set the new brightness
  $ echo 50 > /sys/class/backlight/backlight/brightness
  ```

CAN Bus

There are two CAN bus channels on the RK3568 based Chipsee industrial PC. You can install *can-utils* and use them to test CAN. But you must add one 120Ω resistor between CAN_H and CAN_L on one of the two Boards, as shown on the figure below.

> **Note**
>
> The Chipsee IPC does not mount the 120Ω matched resistor on all CAN signals by default.

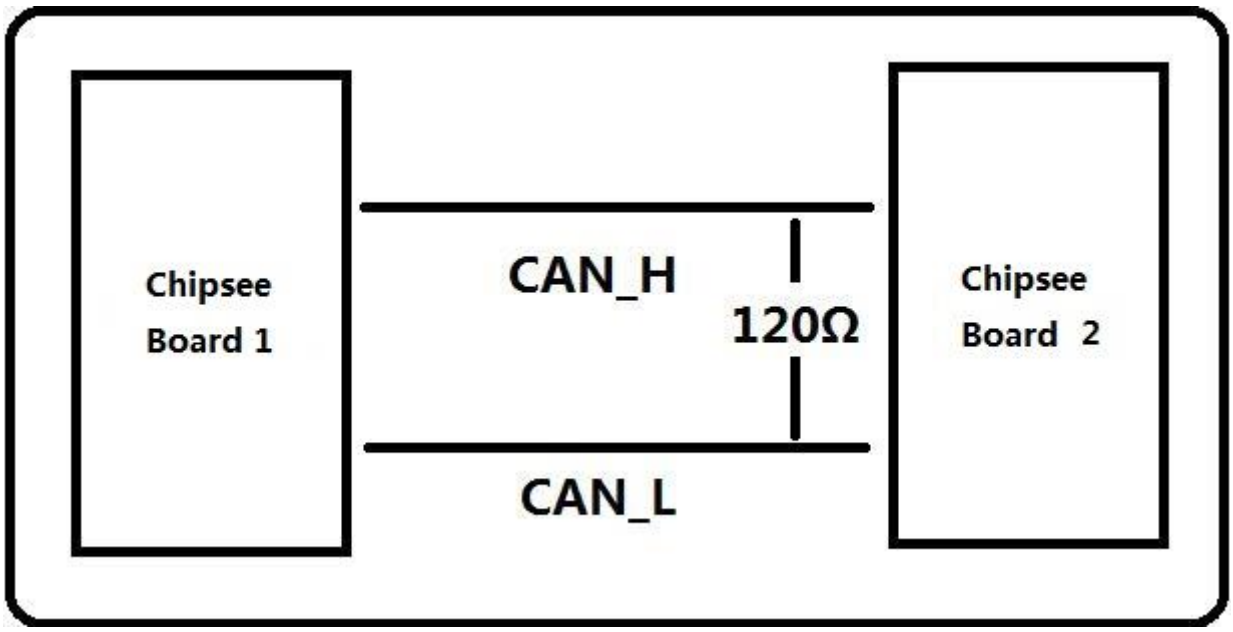


Figure 654: Connecting CAN

Here are a few examples to test CAN by using CAN units.

- **Install** *can-utils* **if you haven't.**

```
$ sudo apt install can-utils
```

- **Set the bit-rate to 50Kbits/sec with triple sampling using the following command (use ROOT user):**

```
$ sudo ip link set can0 down
$ sudo ip link set can0 type can bitrate 50000 triple-sampling on
```

- **Bring up the device using the command:**

```
$ sudo ip link set can0 up
```

- **Transfer packets**

```
$ sudo cansend can0 5A1#11.2233.44556677.88
```

- **Receive data from CAN bus**

```
$ sudo candump can0
```

- **Bring down the device**

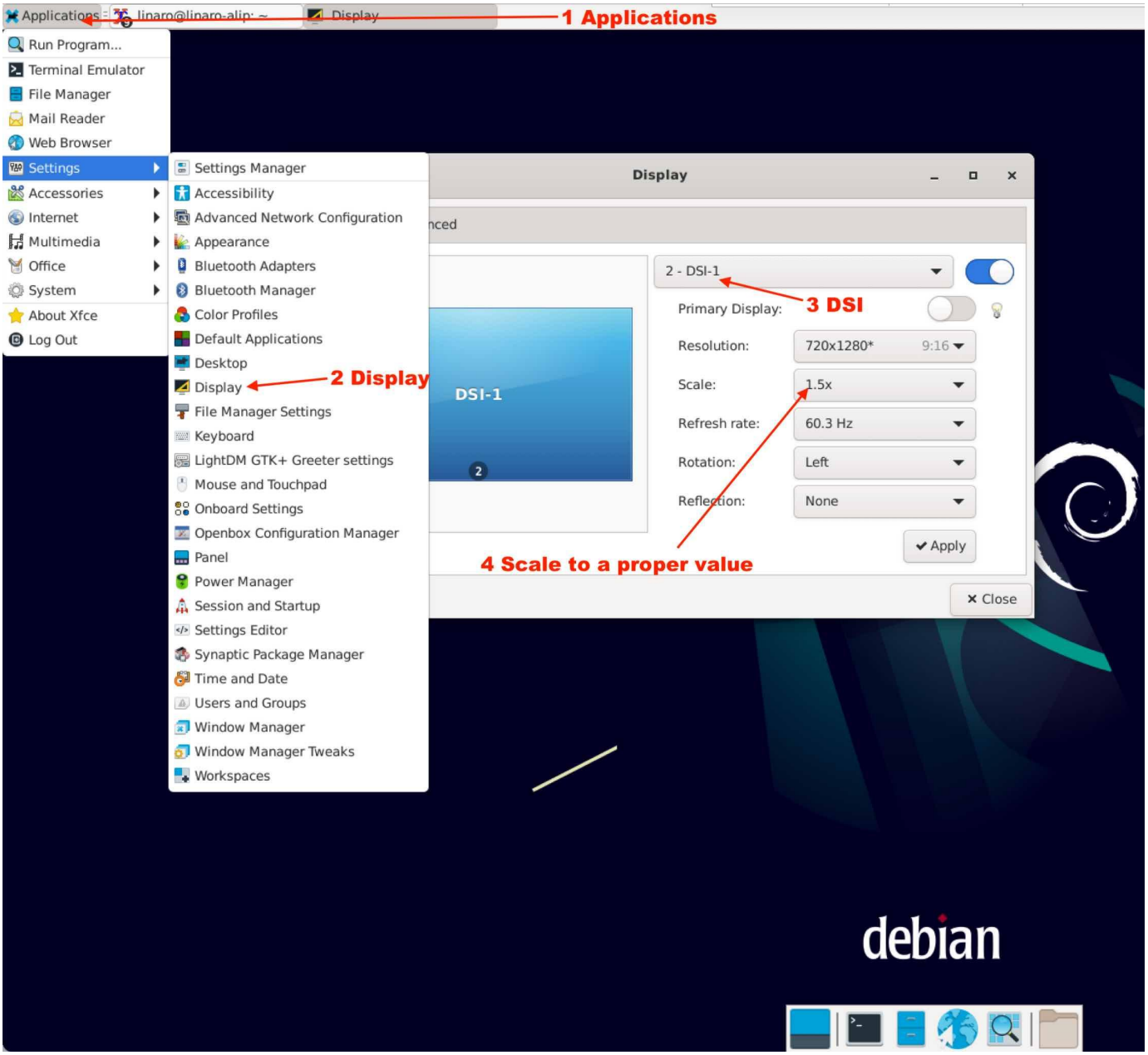
```
$ sudo ip link set can0 down
```

FAQ

External Display Interferes with Touch Screen

When you connect to an external display, such as a 1920 x 1080 resolution display from the HDMI out port, the internal display needs to be scaled, otherwise the touch screen are not aligned correctly. You can set the touch screen and internal display in Display Setting.

For example, on the 5 inch device, the original display is 720x1080(rotated left), if you wish to connect to a 1920x1080 external display, *external height/internal height = 1080/720 = 1.5; external width/internal width = 1920/1080=1.5*. You need to set DSI scale to 1.5 like in the image below.



Scale Internal Display when Attach an External Display

Alternatively, if you need a custom scale setting, you can use command line:

```
$ # Set the resolution according to your needs
$ # For example, if you use 7 inch RK3568 product with a 2K external display,
then set it to 2560x1440
$ su linaro -c "DISPLAY=:0 xrandr --output DSI-1 --same-as HDMI-1 --scale-from
2560x1440"
```

```
linaro@linaro-alip:~$ su linaro -c "DISPLAY=:0 xrandr --output DSI-1 --same-as HDMI-1 --scale-from 1920x1080"
Password:
bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
linaro@linaro-alip:~$ su linaro -c "DISPLAY=:0 xrandr --output DSI-1 --same-as HDMI-1 --scale-from 1080x1920"
Password:
bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
```

Scale Internal Display when Attach an External Display with Command Line

Flashing OS Image

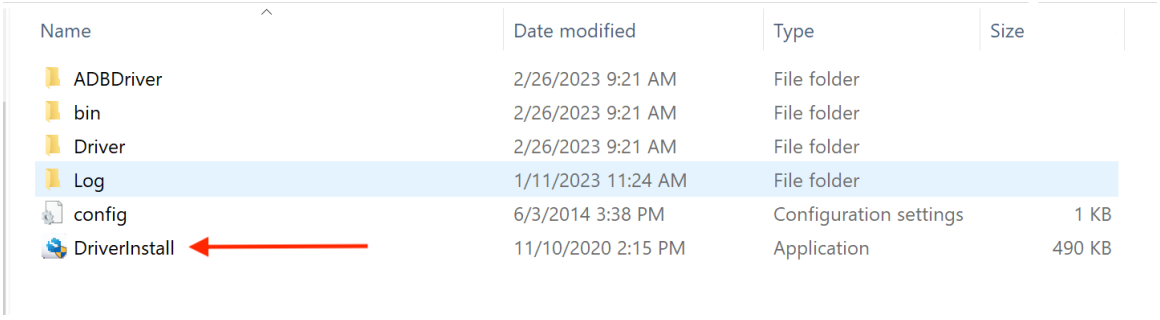
Download Required Tools

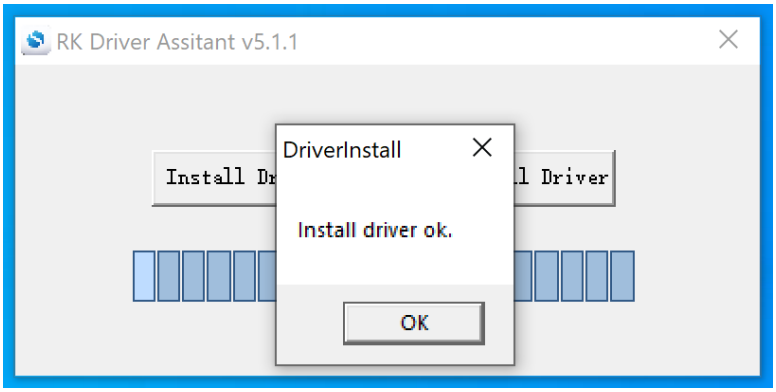
If you want a fresh OS, you can flash your Chipsee industrial PC.

You need two tools to flash the Debian 11 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*.



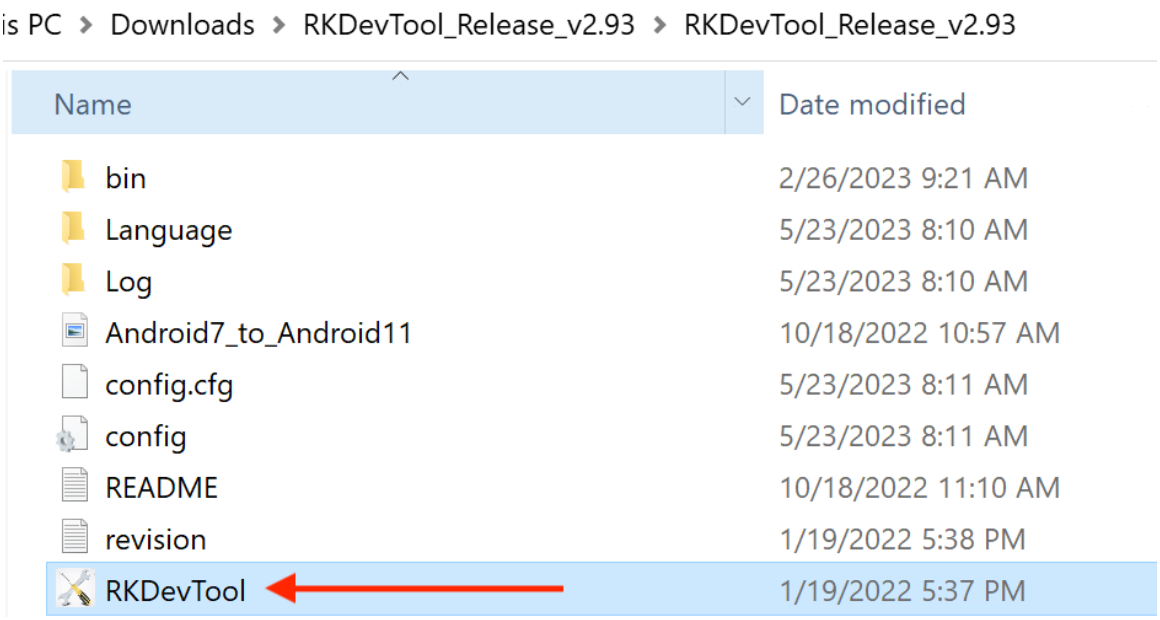


Download Prebuilt OS Images

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

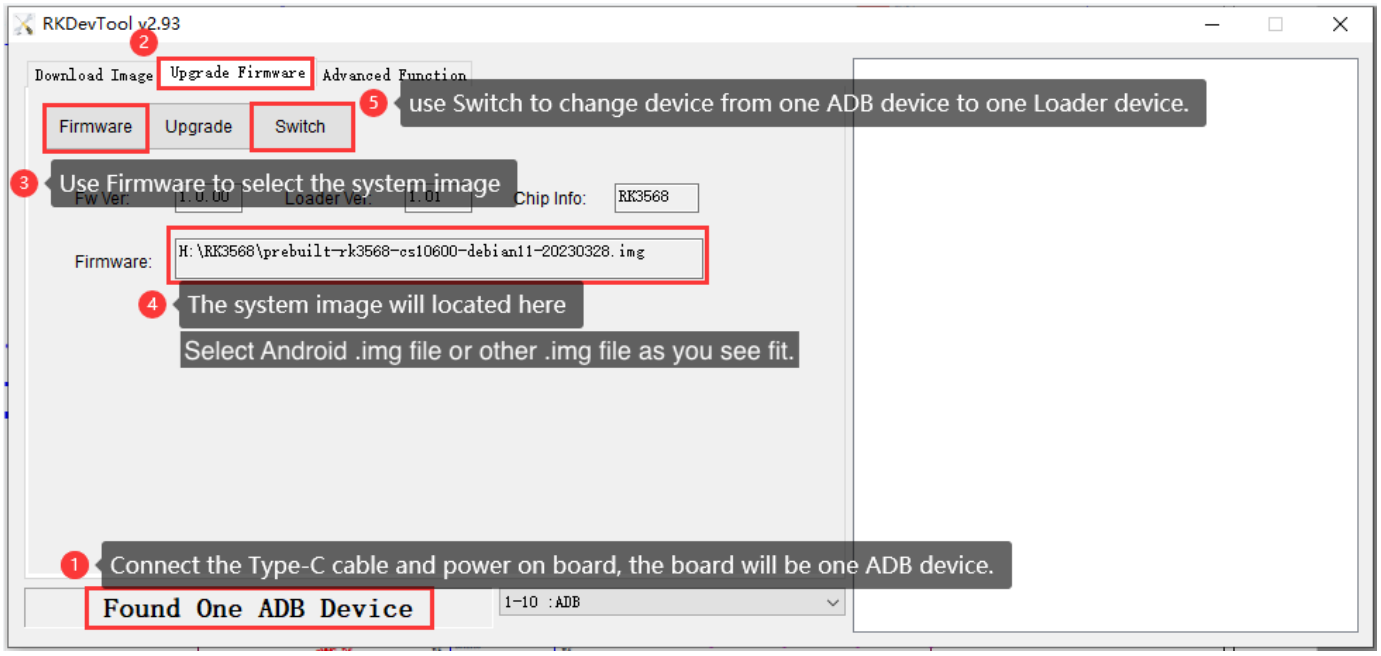
Start Flashing

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.



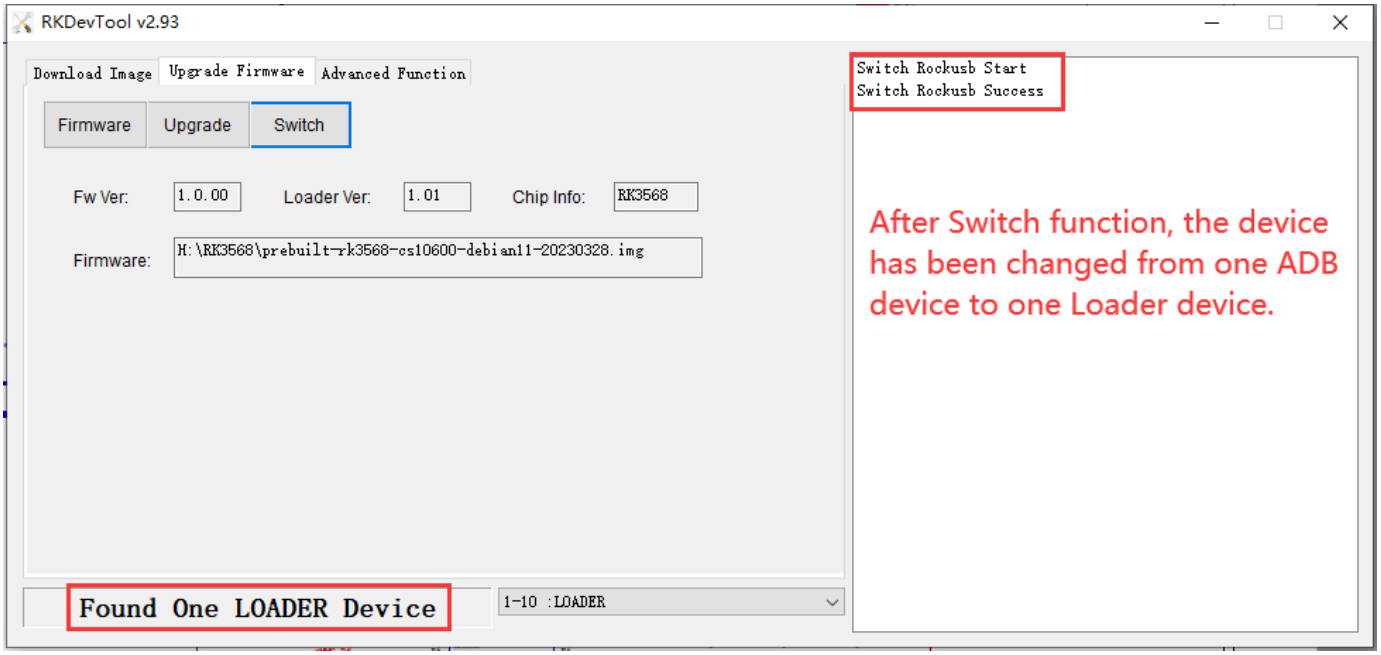
STEP 1:

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 Debian 11 image file. The screenshots show a debian11 img file is selected, but this is applicable to other OSes as well.



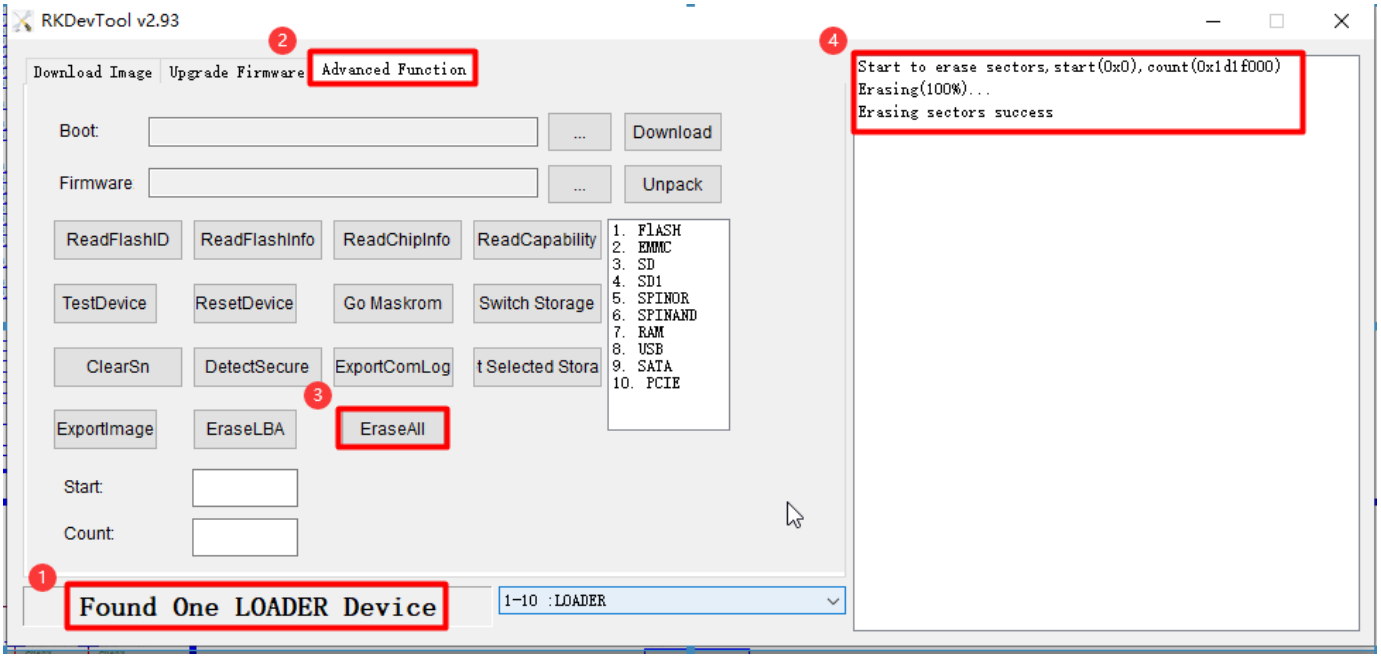
STEP 2:

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



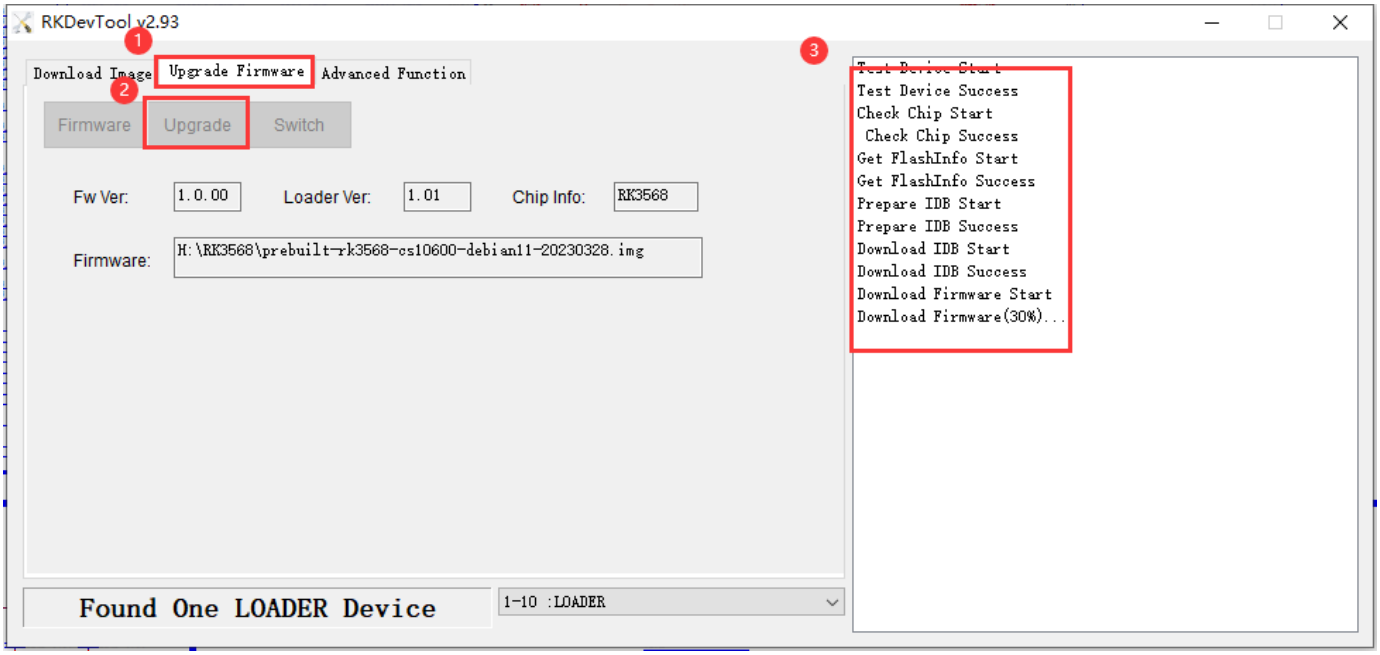
STEP 3:

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.



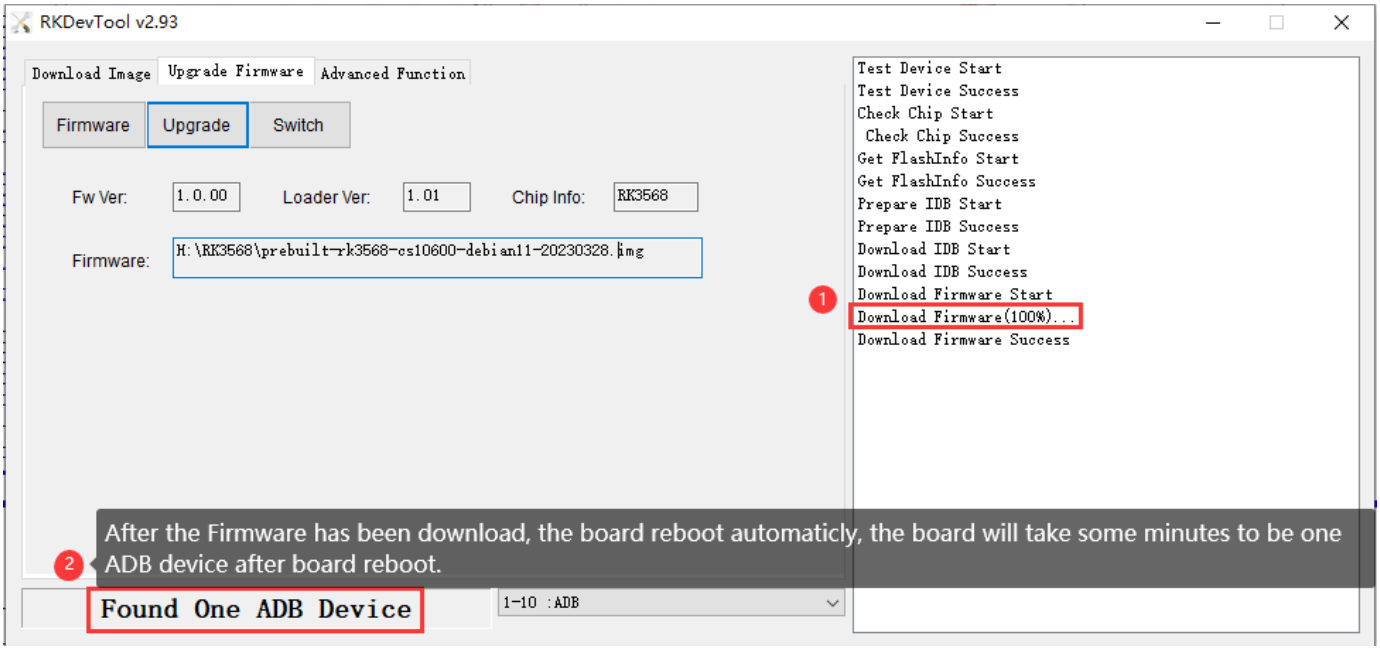
STEP 4:

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.

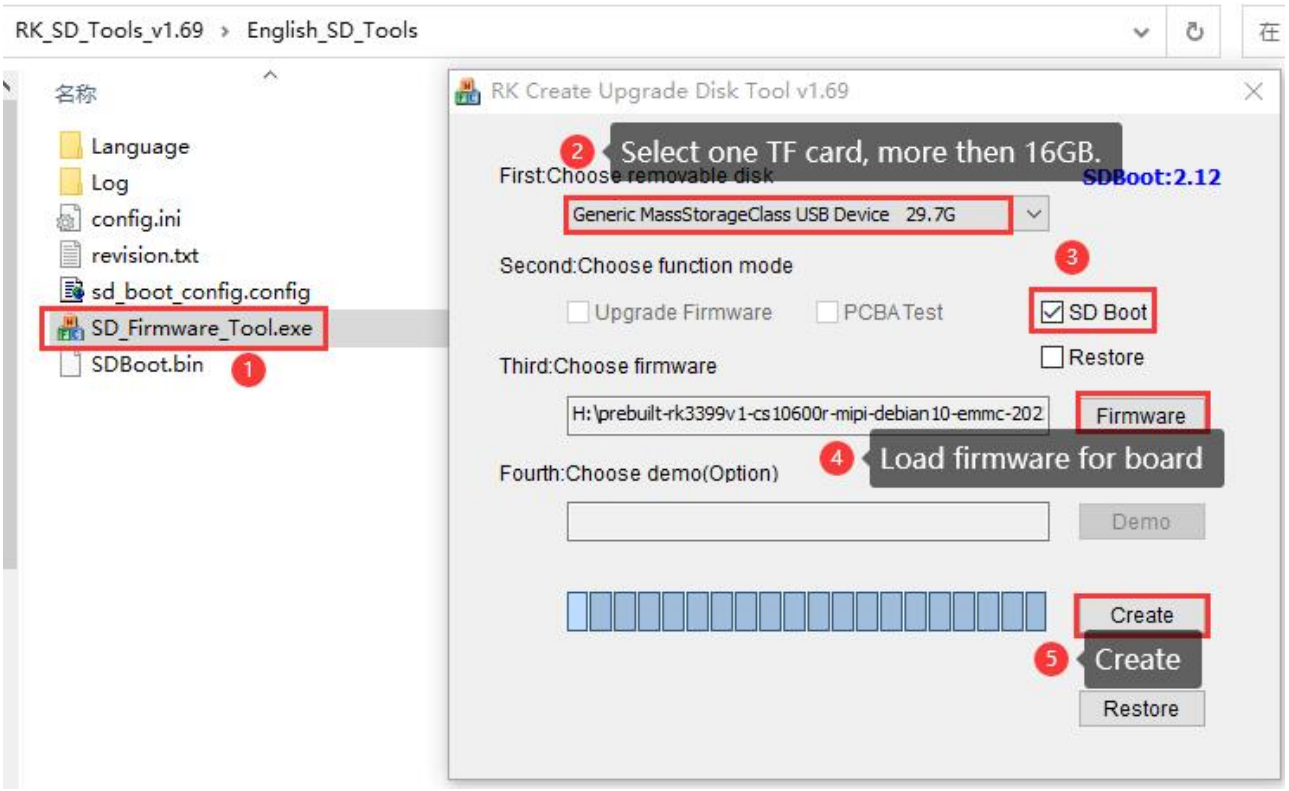
> **Note**
>
> As for the second prebuilt-xxx-sd-xx.img, we use it to make a bootable SD card (imagine the old time people use a WinPE USB stick to boot and backup Windows!).

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.



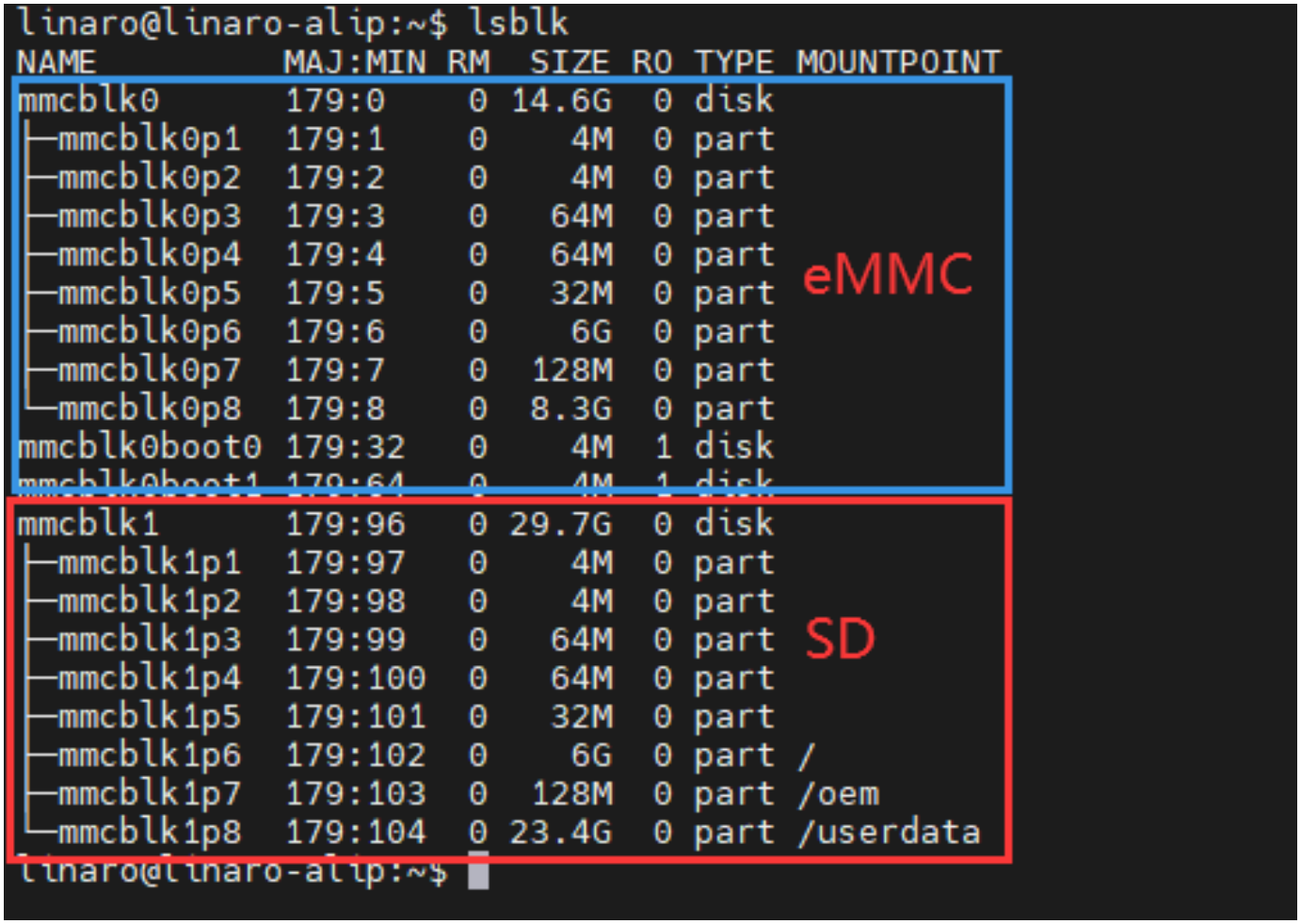
Follow the 5 steps on SDDisktool

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.

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.