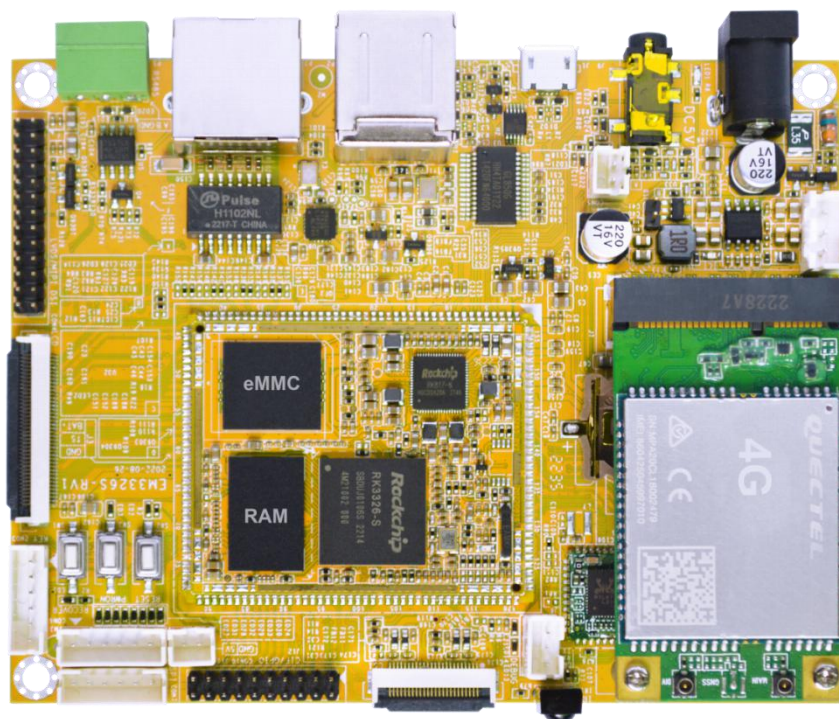


# EM3326S User Manual

---

V2.0



**Boardcon Embedded Design**

[www.boardcon.com](http://www.boardcon.com)

## Preface

### Overview

The content of this document is only described for the development board EM3326S, aiming to help users quickly understand, apply and develop the EM3326S development board.

### System Support

Development Board	Android12	Debian11	Buildroot
EM3326S-RV3 2023-08-09 CM3326S	Y	Y	Y

### Revision History

Version	Date	Author	Revision History
V1.0	2023-03-17	Boardcon Team	Initial version
V2.0	2023-12-25	Boardcon Team	Added support for Debian 11

### Version update instructions

Due to product version upgrades or other reasons, the content of this document will be updated from time to time. At the same time, it will be synchronized to the Boardcon website ([www.boardcon.com](http://www.boardcon.com) or [www.armdesigner.com](http://www.armdesigner.com)).

If you have any questions, concerns or comments about what is stated in the document, please feel free to contact us.

- **Company:** Boardcon embedded design limited.
- **Address:** 2508-2509 Haofang Tianji Plaza, 11008 Beihuan Avenue, Nanshan District, Shenzhen, GuangDong, China. 518051.
- **URL:** <https://www.boardcon.com> <https://www.armdesigner.com/>
- **Email:** [market@armdesigner.com](mailto:market@armdesigner.com)
- **Technical Support Inquiries:** [support@armdesigner.com](mailto:support@armdesigner.com)
- **Tel:** +86-755-26481393  
+86-755-27571591
- **Fax:** +86-755-27571552

## **Content**

Introduction.....	5
1. Overview.....	5
2. Product Parameters.....	5
3. Hardware Interface Introduction.....	7
Install Drivers and Tool.....	8
1.Install RK Driver Assitant.....	8
2.Install CH9102X Driver.....	9
2.1 How to connect the serial port tool.....	9
2.2 Install driver.....	10
3.Install Serial Terminal Tool.....	11
Android12.....	13
1.Compiler Environment.....	13
2.Install Tools.....	13
2.1 Install dependent packages.....	14
2.2 Install JDK.....	14
3.Compile Source.....	14
4.Images Operation.....	15
4.1 Pack image.....	15
4.2 Unzip firmware.....	17
5.Burn Images.....	18
5.1 Burn update firmware.....	18
5.2 Burn split firmware.....	20
6.Android12 Application.....	20
6.1 Normal display.....	20
6.2 USB OTG.....	20
6.3 SD card.....	21
6.4 USB host.....	22
6.5 Video player.....	22
6.6 Ethernet.....	23
6.7 Record.....	24
6.8 RTC.....	25
6.9 WiFi.....	26
6.10 Bluetooth.....	26
6.11 Camera.....	26
6.12 RS485.....	27
6.13 SPI.....	28
6.14 4G.....	30
6.15 IR.....	31
Debian11.....	32
1.Compiler Environment.....	32



2.Install Tools.....	32
3.Compile Source.....	32
4.Images Operation.....	34
4.1 Pack image.....	34
4.2 Unzip firmware.....	35
5.Burn Images.....	36
5.1 Burn update firmware.....	36
5.2 Burn split firmware.....	38
6.Debian11 Application.....	39
6.1 Display.....	39
6.2 USB OTG.....	39
6.3 SD/USB device.....	40
6.4 Audio.....	42
6.5 Video player.....	43
6.6 Ethernet.....	45
6.7 RTC.....	47
6.8 4G(EC20).....	47
6.9 GPS(EC20).....	48
6.10 WiFi.....	49
6.11 Bluetooth.....	50
6.12 Camera.....	52
6.13 RS485.....	53
6.14 SPI.....	54
6.15 IR.....	57
Tool Usage.....	58
1.Driver.....	58
1.1 USB driver.....	58
1.2 Debug serial port driver.....	58
2.Debug Tool.....	58
3.Development Flashing Tool.....	59
3.1 Loader mode.....	59
3.2 Maskrom mode.....	60
4.Tool to Implement SD Upgrading and Boot.....	62
4.1 How to format sdcard.....	63
4.2 How to make SD upgrade card.....	64
4.3 How to make SD boot card.....	66
Common Issues.....	67
1.How do open the kernel configuration file.....	67
2.Error about lz4.....	68
Configuration list.....	68

# Introduction

## 1. Overview

The EM3566S CPU integrates quad-core Quad-core Cortex-A35@1.5GHz, RK3326-S supports almost full-format H.264/H.265 decoder by 1080p@60fps, H.264 encoder by 1080p@30fps, high-quality JPEG encoder/decoder. Embedded ARM G31-2EE GPU makes RK3326-S completely compatible with OpenGL ES 1.1/2.0/3.2, DirectX 11 FL9\_3, OpenCL 2.0 and Vulkan 1.0. Special 2D hardware engine will maximize display performance and provide very smoothly operation.

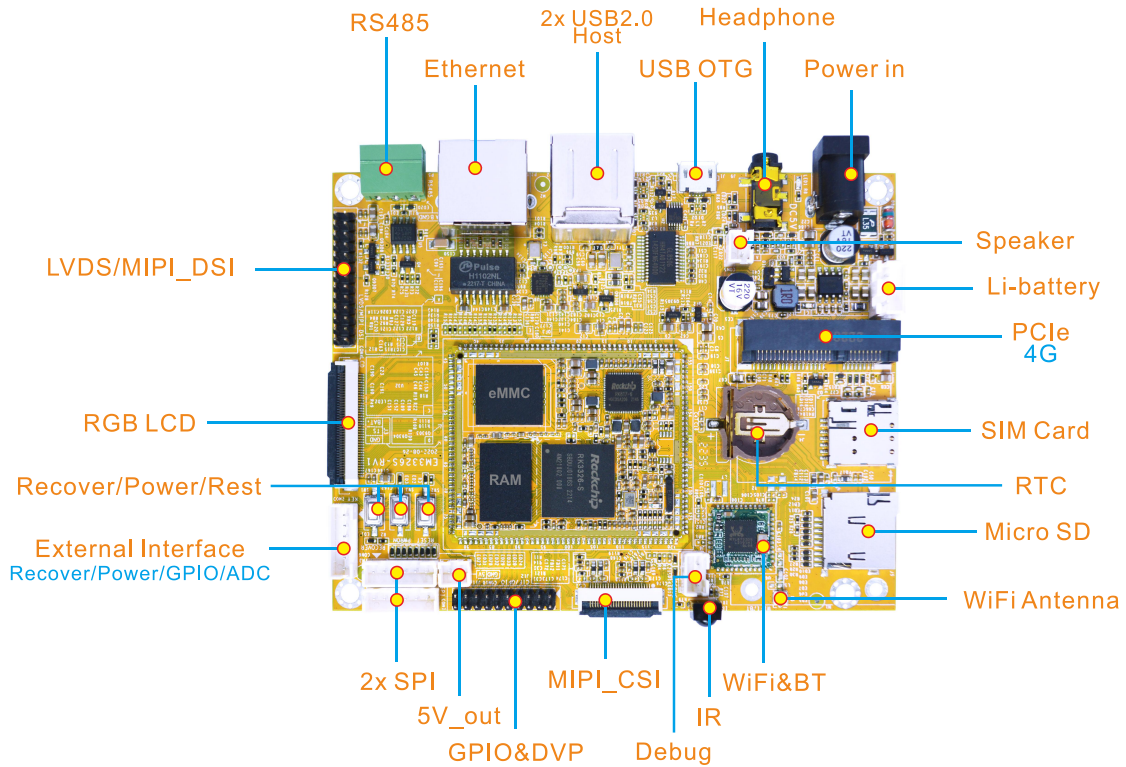
## 2. Product Parameters

Basic Parameters		
SOC	RK3326-S	
CPU	Rockchip RK3326-S Quad-core Cortex-A35 @1.5GHz	
GPU	<ul style="list-style-type: none"> <li>• ARM G31-2EE</li> <li>• OpenGL ES 1.1/2.0/3.2</li> <li>• Vulkan 1.0</li> <li>• DirectX 11 FL9_3, OpenCL 2.0</li> <li>• Special 2D hardware engine</li> </ul>	
Video	Decoder	<ul style="list-style-type: none"> <li>• Support almost full-format H.264 decoder by 1080p@60fps</li> <li>• Support H.265 decoder by 1080p@60fps</li> </ul>
	Encoder	<ul style="list-style-type: none"> <li>• Support H.264 encoder by 1080p@30fps, high-quality JPEG encoder/decoder.</li> </ul>
RAM	1GB LPDDR4 (support upgrade)	
ROM	8GB eMMC (support upgrade)	
Support system	Android12	
Hardware Parameters		
Extended Storage	<ul style="list-style-type: none"> <li>• Support 1x MicroSD Card</li> </ul>	
Display	<ul style="list-style-type: none"> <li>• Support MIPI_DSI interface/LVDS interface MIPI_DSI Up to 1080p@60fps display output. LVDS Up to 1280x800@60fps.</li> </ul>	



	<ul style="list-style-type: none"><li>• Support RGB Parallel Display interface</li></ul>
Audio	<ul style="list-style-type: none"><li>• Support 1x speaker audio output</li><li>• Support 1x Headphone output/input</li></ul>
USB	<ul style="list-style-type: none"><li>• Support 2xUSB2.0</li><li>• Support 1x USB OTG 2.0 (ADB)</li></ul>
Network	<ul style="list-style-type: none"><li>• Support 1x 100M Ethernet</li><li>• Support 1x WIFI/BT module</li><li>• Support 1x 4G module</li></ul>
Camera	Support 1x Camera
Peripheral communication	<ul style="list-style-type: none"><li>• Support 1x RS485</li></ul>
Other parameters	Support 1x Debug, 1x IR, 1x RTC, 2x SPI
<b>Electrical Parameters</b>	
Power supply input voltage	5V/3A
RTC input voltage	3V/0.6uA
Operating temperature	0~70°
Storage temperature	-40~85°
<b>Structural Parameters</b>	
Core board dimensions	45mm x 37mm
Motherboard dimensions	110mm x 85mm

### 3. Hardware Interface Introduction



Interface parameters	
Power in	5V DC power input interface
Reset	Reset key
USB OTG	OTG download interface (multiplexed with USB Host/Ethernet/4G model)
Ethernet	100M Ethernet RJ45 interface
2XUSB2.0 Host	Dual-layer USB2.0 HOST interface
Recover	Recovery key
WIFI&BT	WIFI&Bluetooth module
PCIe2.0 4G model	4G model interface
RTC	RTC coin cell connector
MIPI_CSI	Camera interface
Audio I/O	Earphone output/input and Speaker output
Micro SD	SD card interface (multiplexed with debug port)
IR	infrared receiver

Power	Power key
Debug	debug the serial port
RS485	RS485 communication interface
MIPI_DSI/LVDS	DSI/LVDS display interface

# Install Drivers and Tool

To download firmware and debug in the terminal, the following drivers and software need to be installed(for Windows computers):

Number	Driver name	Driver	Use
1	RK Driver Assitant	DriverInstall.exe	OTG USB driver installation assistant
2	CP210x	PreInstaller.exe	Serial port debugging driver
3	Serial Terminal Tool	SecureCRT.exe	Debugging tool

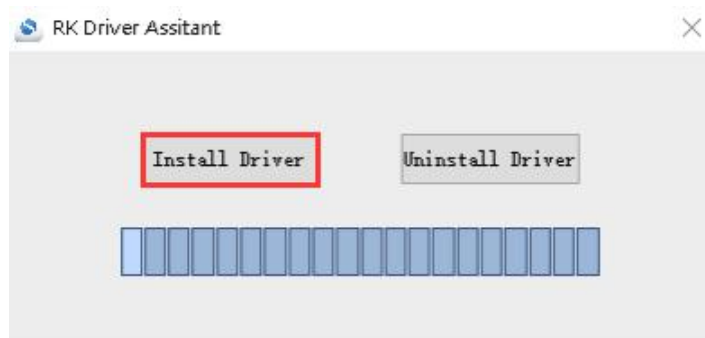
## 1.Install RK Driver Assitant

**Step 1**, open DriverAssitant\_v5.1.1/DriverInstall.exe.

**Step 2**, to avoid driver conflicts, click **Uninstall Driver** to uninstall the driver.

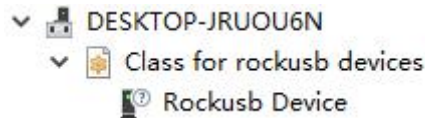


**Step 3**, click button **Install Driver** to install.

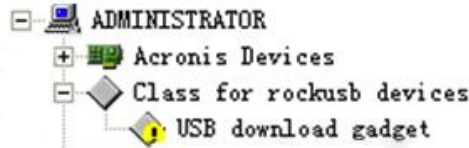


**Step 4**, after the installation is complete, connect the board and PC with Type-C USB cable and press the **Recovery** key and hold then power the board, the following information is displayed in the

Computer Device Manager, indicating that the USB driver was successfully installed.

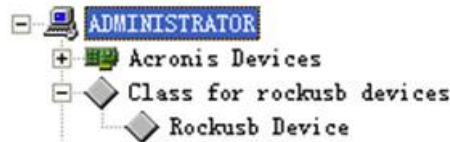


**Step 5**, if the following device information appears on the computer device manager after the operation in step 4, you need to proceed to the next step.



**Step 6**, the WINDOW will pop up found New Hardware Wizard dialog box, choose to install from the specified location, and then select: \DriverAssitant\_v5.1.1\ADBDriver.

**Step 7**, after the installation is completed, the following device information can be seen in the Computer Device Manager.

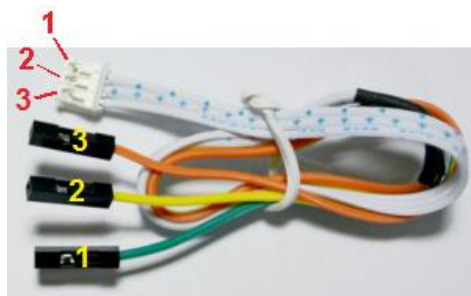


## 2.Install CH9102X Driver

### 2.1 How to connect the serial port tool



Pin	Connection Description
RXD	Receive, connect to RX pin of the board.
TXD	Transmit, connect to TX pin of the board.
GND	Ground, connect to GND pin of the board.
3V3	No need to connect.



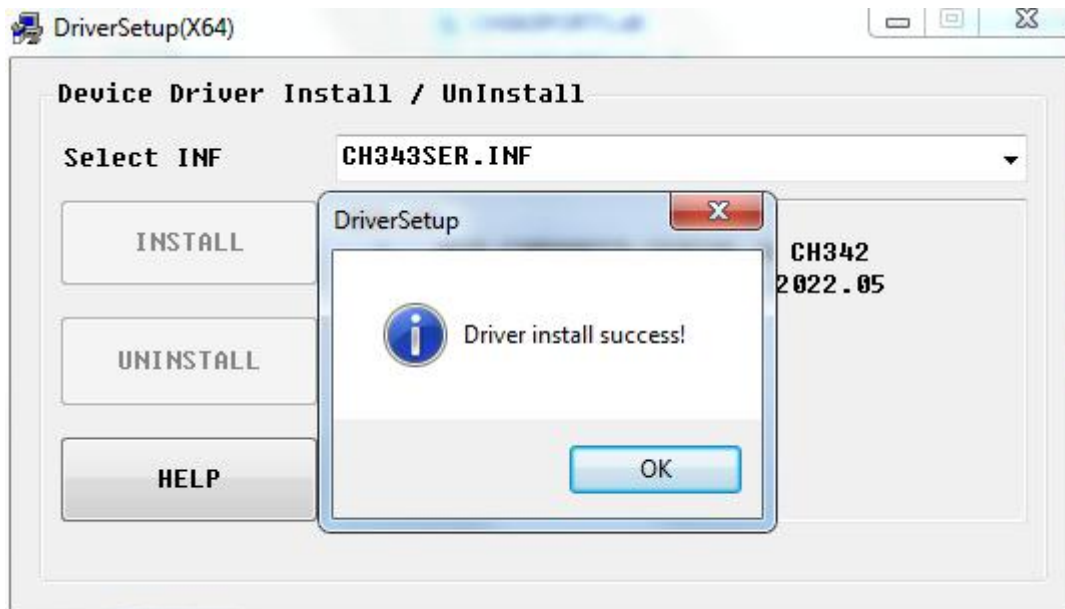
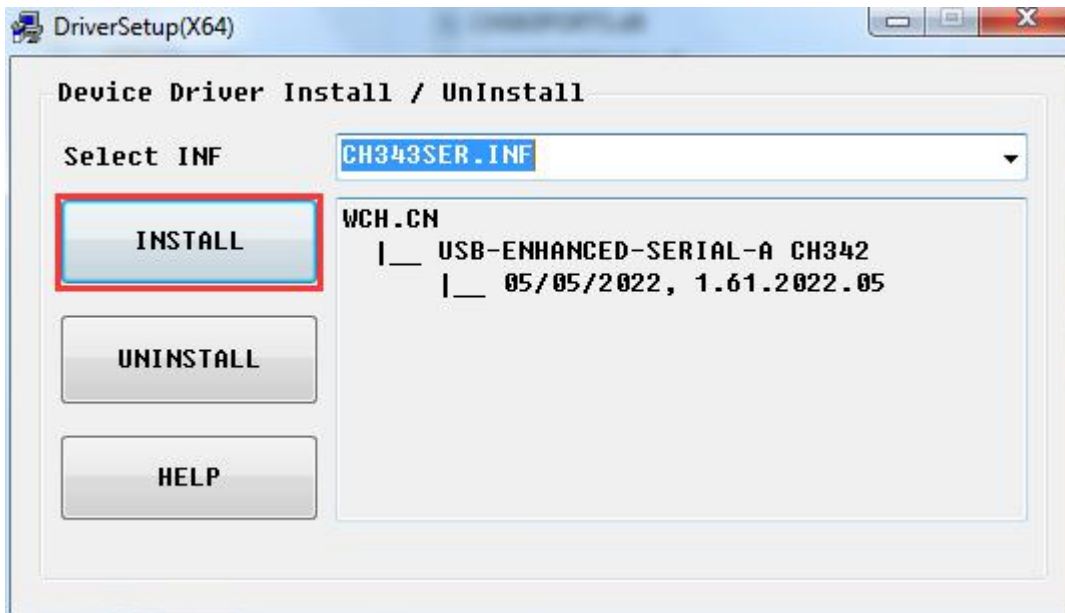
Pin	Connection Description
1	RX, connect to TXD pin of the CH9102X Module.
2	TX, connect to RXD pin of the CH9102X Module.
3	Ground, connect to GND pin of the CH9102X Module.

## 2.2 Install driver

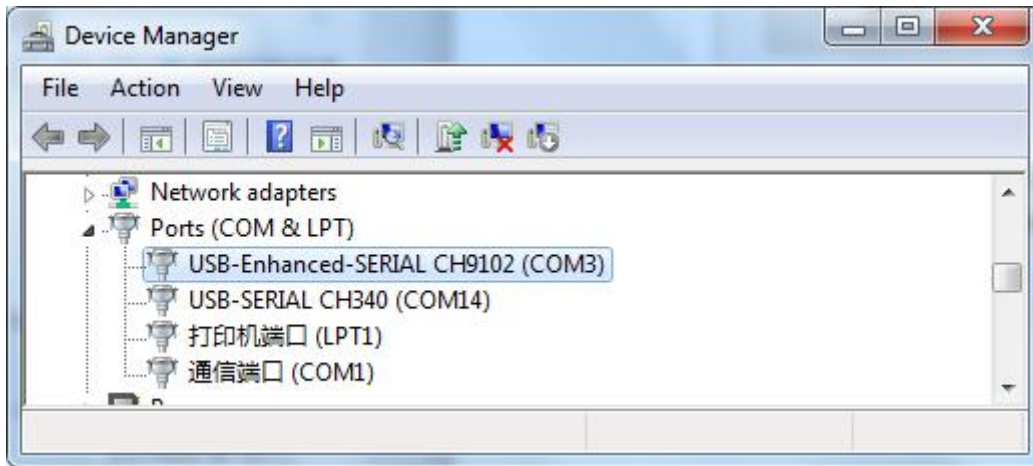
*Step 1*, plug the CH9102X Module to the PC.

*Step 2*, unzip CH343SER.ZIP on Windows.

*Step 3*, select and install the corresponding SETUP.EXE according to the computer properties..



*Step 4*, after the installation is completed, the device will be listed under **Device Manager** -> **ports** with unique serial port assigned.

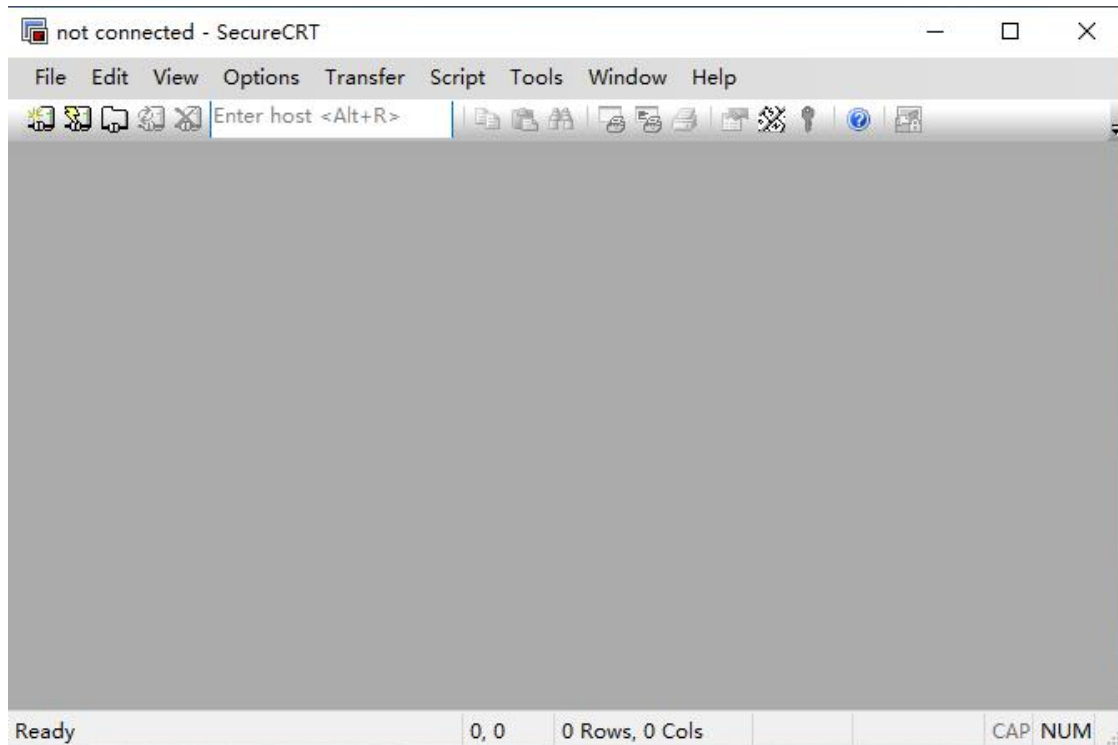


### 3.Install Serial Terminal Tool

The serial terminal SecureCRT is used for debugging in Windows. It can be used directly after decompression.

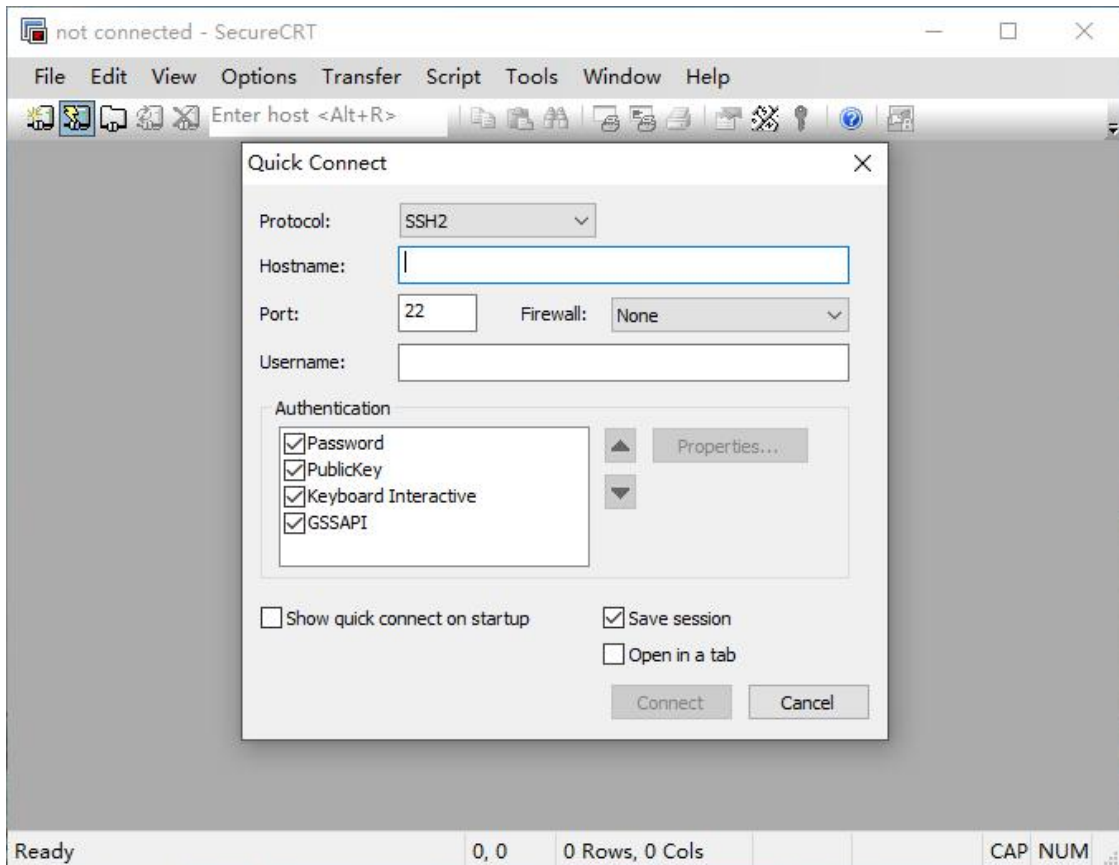
*Step 1*, unzip Platform/SecureCRT.rar on PC.

*Step 2*, click SecureCRT/SecureCRT.exe open the SecureCRT.

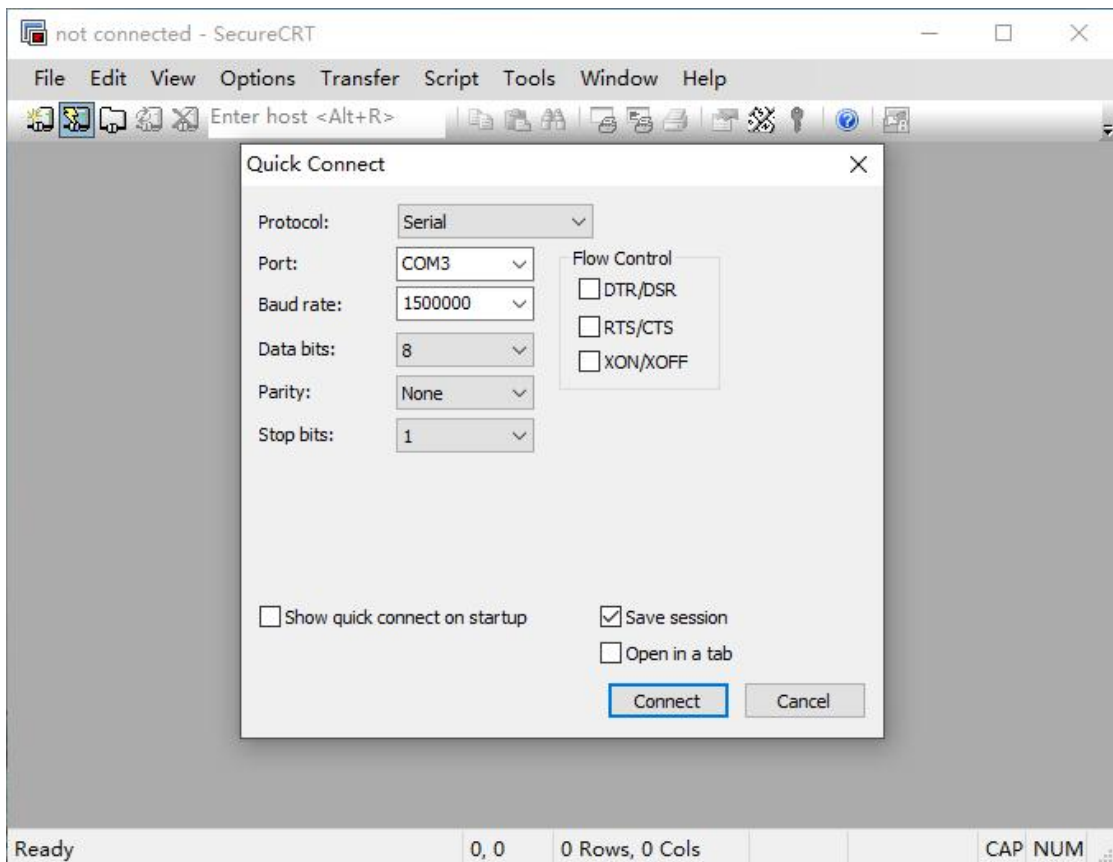


*Step 3*, confirm that the CP210x driver has been installed and the CP2102 module is connecting to the PC.

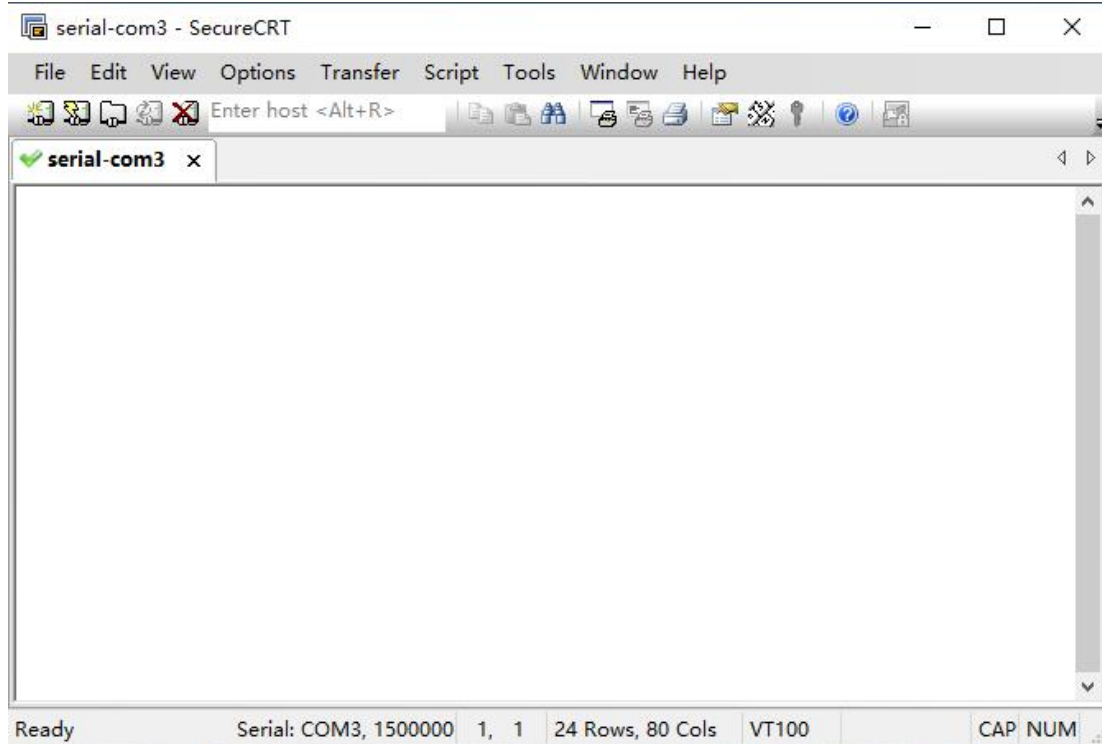
*Step 4*, click the **Quick Connect** button to go to the Quick Connect configuration screen.



Step 5, configure as shown in the following figure:



Step 6, after clicking connect button, the terminal serial interface will be successfully accessed.



# Android12

## 1. Compiler Environment

It is recommended to use Ubuntu 16.04 or higher version for compilation . If you encounter an error during compilation, you can check the error message and install the corresponding software packages accordingly. Other Linux versions may need to adjust the software package accordingly. In addition to the system requirements, there are other hardware and software requirements.

Hardware requirements	Software requirements
64-bit system, hard disk space should be greater than 200G. If you do multiple builds, you will need more hard drive space.	Ubuntu 16.04 system

## 2. Install Tools

The contents of this directory only provide the software package installation commands that are needed to build the compiled SDK environment. Please install other tools such as samba and ssh yourself.

PC OS: ubuntu system

Network: online

Permission: root

## 2.1 Install dependent packages

```
# sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl
zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 lib32ncurses5-dev x11protocore-dev libx11-dev
lib32z-dev ccache libgl1-mesa-dev libxml2-utils xsltproc
unzip python-pyelftools python3-pyelftools device-tree-compiler libbfd-dev
libbfd1 libssl-dev liblz4-tool python-dev
```

If the compilation encountered error, please according to the error message install the corresponding software package.

## 2.2 Install JDK

```
$ sudo apt-get update
$ sudo apt-get install openjdk-8-jdk
```

# 3. Compile Source

*Step 1*, unzip the source and set the compile board

```
$ tar xvf em3326_android-12.0.tar.gz
$ cd android-12.0
```

*Step 2*, compile uboot

```
$ cd u-boot
$ ./make.sh rk3326
```

*Step 3*, compile the kernel

```
$ cd kernel-4.19
$ make ARCH=arm64 rockchip_defconfig android-11-go.config
$ make ARCH=arm64 rk3326-evb-lp3-v10-avb.img -j8
```

Note: If only want to update kernel, must be compiled kernel as follow:

```
$ cd kernel-4.19
$ make ARCH=arm64 BOOT_IMG=./rockdev/Image-rk3326_s/boot.img
rk3326-evb-lp3-v10-avb.img -j8
```

You can flash boot.img under the catalogue of kernel-4.19 directly to boot position of machine after compiling, and please load the partition table (parameter.txt) when flashing, for fear of flashing to the wrong place.

*Step 4*, compile Android

```
$ source build/envsetup.sh
$ lunch rk3326_s-userdebug
$ make -j8
```

*Step 5*, generated image file

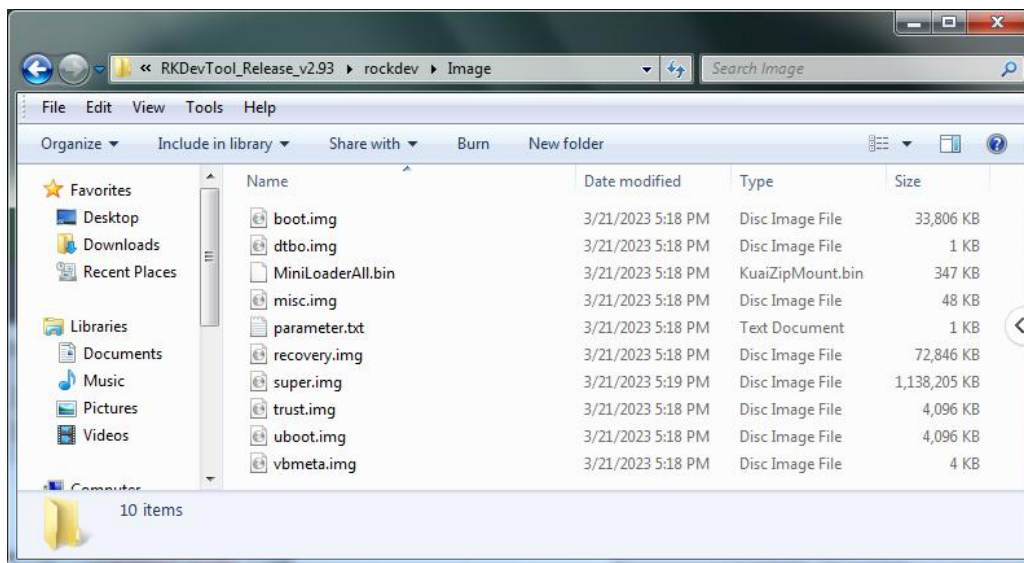
```
$. /mkimage.sh  
$. /build.sh -u  
$. cd rockdev  
$. ls
```

Images and update.img are generated in current directory.

## 4.Images Operation

### 4.1 Pack image

*Step 1*, copy the firmware file to be packaged to AndroidTool/rockdev /Image.



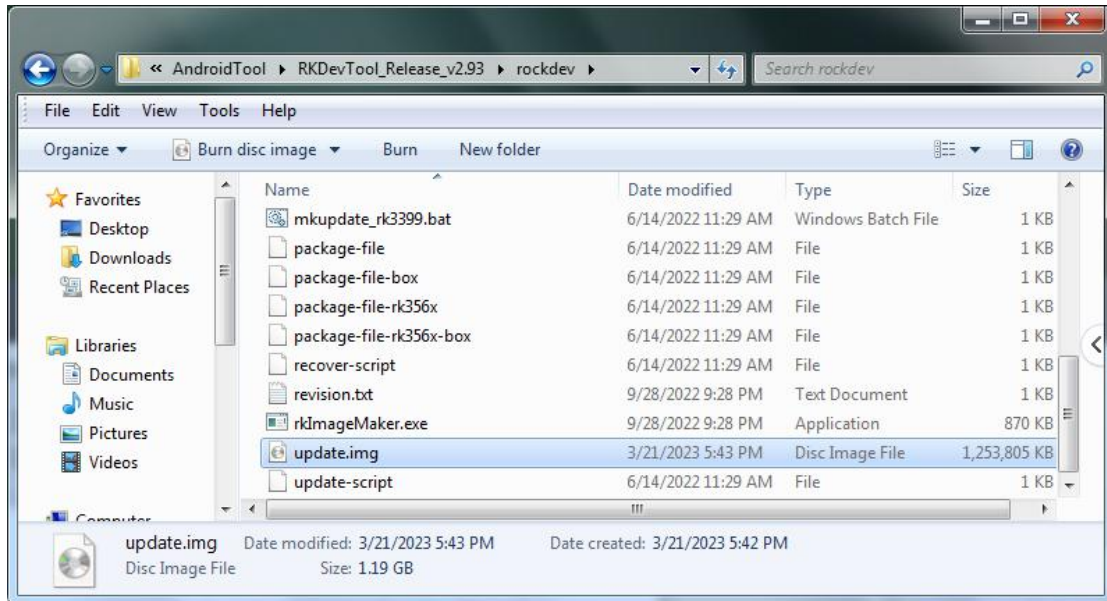
*Step 2*, enter AndroidTool/rockdev, double-click to run **mkupdate\_rk3326.bat**.



```
C:\Windows\system32\cmd.exe
E:\CD\3326\em3326-k101\AndroidTool\RKDevTool_Release_v2.93\rockdev
Android Firmware Package Tool v2.0
----- PACKAGE -----
Add file: .\backupimage\package-file
package-file,Add file: .\backupimage\package-file done,offset=0
Add file: .\backupimage\../Image/MiniLoaderAll.bin
bootloader,Add file: .\backupimage\../Image/MiniLoaderAll.bin
Add file: .\backupimage\../Image/parameter.txt
parameter,Add file: .\backupimage\../Image/parameter.txt done
Add file: .\backupimage\../Image/trust.img
trust,Add file: .\backupimage\../Image/trust.img done,offset=
Add file: .\backupimage\../Image/uboot.img
uboot,Add file: .\backupimage\../Image/uboot.img done,offset=
Add file: .\backupimage\../Image/misc.img
misc,Add file: .\backupimage\../Image/misc.img done,offset=0x
Add file: .\backupimage\../Image/kernel.img
Error:<AddFile> open file failed,err=2?
----- FAILED -----

E:\CD\3326\em3326-k101\AndroidTool\RKDevTool_Release_v2.93\rockdev
Android Firmware Package Tool v2.0
----- PACKAGE -----
Add file: .\package-file
package-file,Add file: .\package-file done,offset=0x800,size=0x
Add file: .\Image/MiniLoaderAll.bin
bootloader,Add file: .\Image/MiniLoaderAll.bin done,offset=0x10
Add file: .\Image/parameter.txt
parameter,Add file: .\Image/parameter.txt done,offset=0x58000,s
Add file: .\Image/trust.img
trust,Add file: .\Image/trust.img done,offset=0x58800,size=0x40
Add file: .\Image/uboot.img
uboot,Add file: .\Image/uboot.img done,offset=0x458800,size=0x4
Add file: .\Image/misc.img
misc,Add file: .\Image/misc.img done,offset=0x858800,size=0xc00
Add file: .\Image/boot.img
boot,Add file: .\Image/boot.img done,offset=0x864800,size=0x210
Add file: .\Image/dtbo.img
dtbo,Add file: .\Image/dtbo.img done,offset=0x2968000,size=0x27
Add file: .\Image/vbmeta.img
vbmeta,Add file: .\Image/vbmeta.img done,offset=0x2968800,size=
Add file: .\Image/recovery.img
recovery,Add file: .\Image/recovery.img done,offset=0x2969800,s
Add file: .\Image/super.img
super,Add file: .\Image/super.img done,offset=0x708d000,size=0x
Add CRC...
Make firmware OK!
----- OK -----
```

Step 3, the **update.img** will be generated in **rockdev** directory.

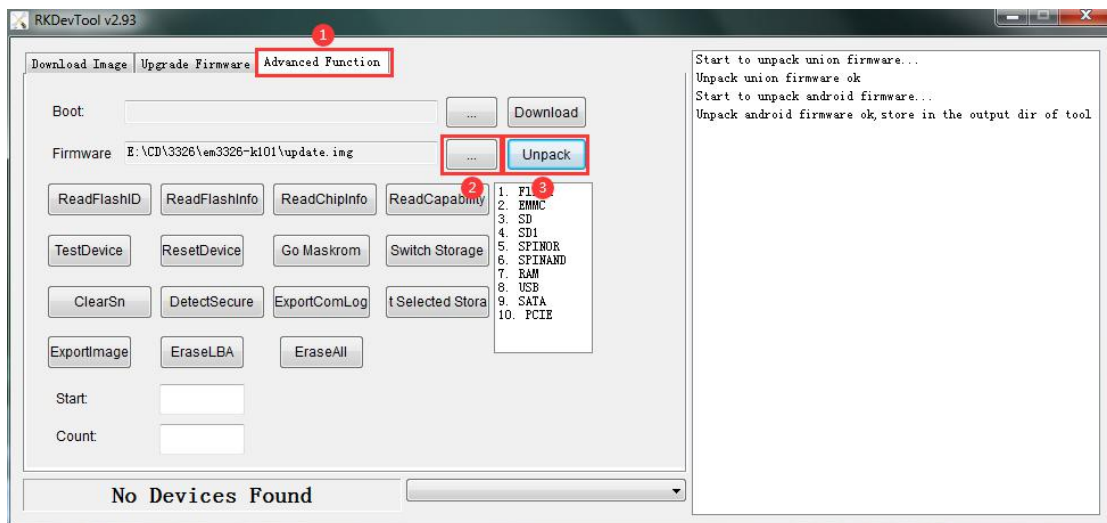


## 4.2 Unzip firmware

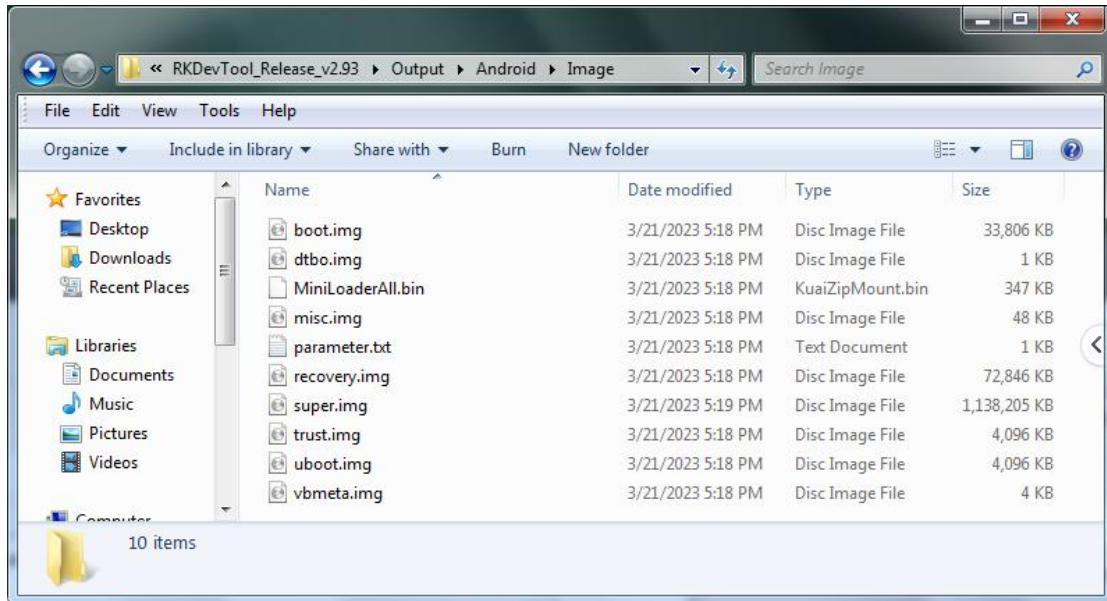
**Step 1**, unzip RKDevTool\_Release\_v2.93.zip on Windows.

**Step 2**, open RKDevTool\_Release\_v2.93\RKDevTool.exe.

**Step 3**, click **Advanced Function** -> **Firmware**, select **update.img**. Click **Unpack** to Unzip.



**Step 4**, the unzip files will be generated in RKDevTool\_Release\_v2.93\Output\Android\image directory.

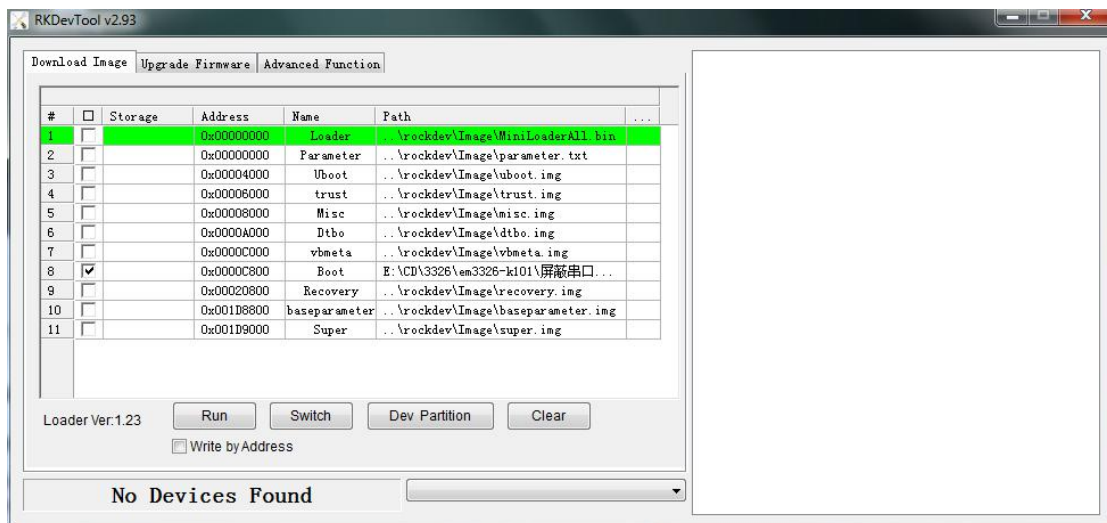


## 5. Burn Images

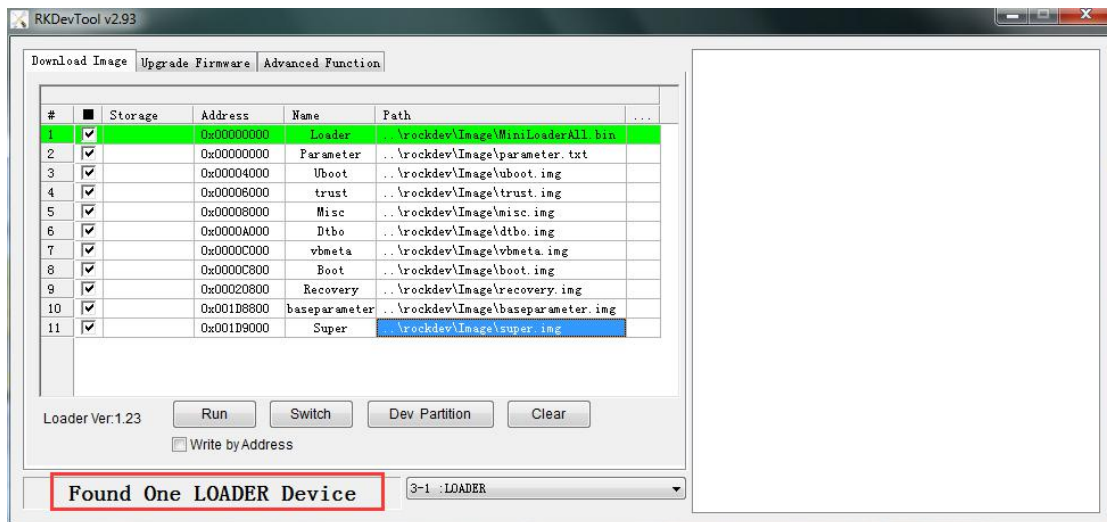
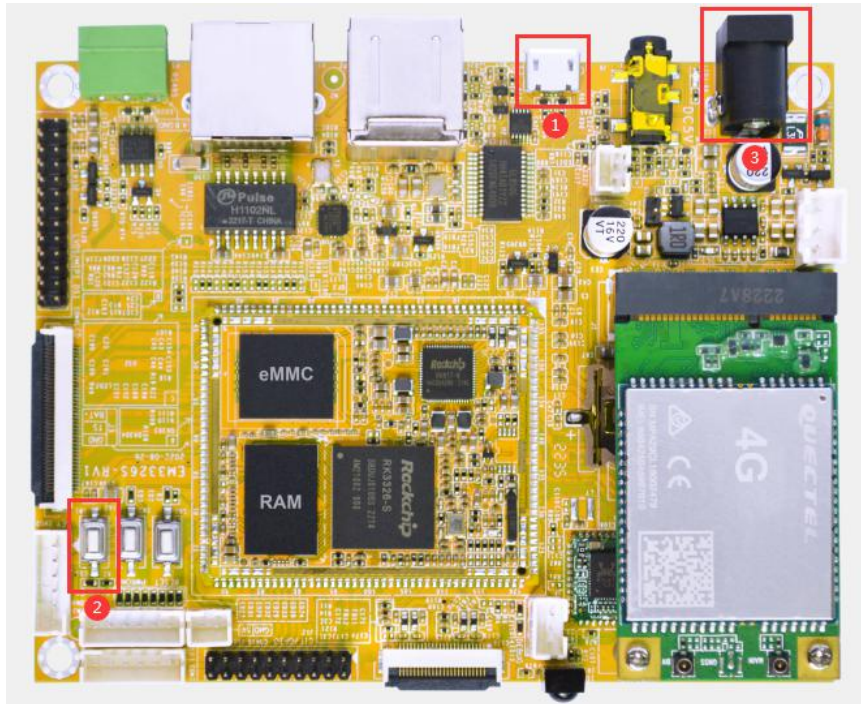
### 5.1 Burn update firmware

*Step 1*, unzip RKDevTool\_Release\_v2.93.zip on Windows.

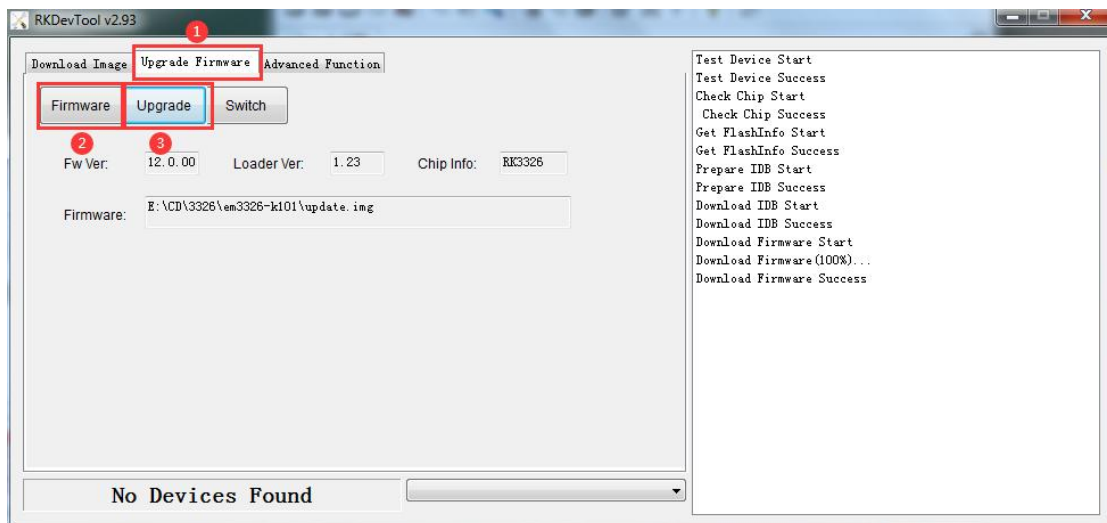
*Step 2*, open RKDevTool\_Release\_v2.93\RKDevTool.exe.



*Step 3*, connect PC and development board with OTG USB cable, keep pressing the **Recover Key** and power the board until the windows PC shows **Found one LOADER Device**.



Step 4, click **Upgrade Firmware** -> **Firmware**, select **update.img**, then click **Upgrade** to flash.

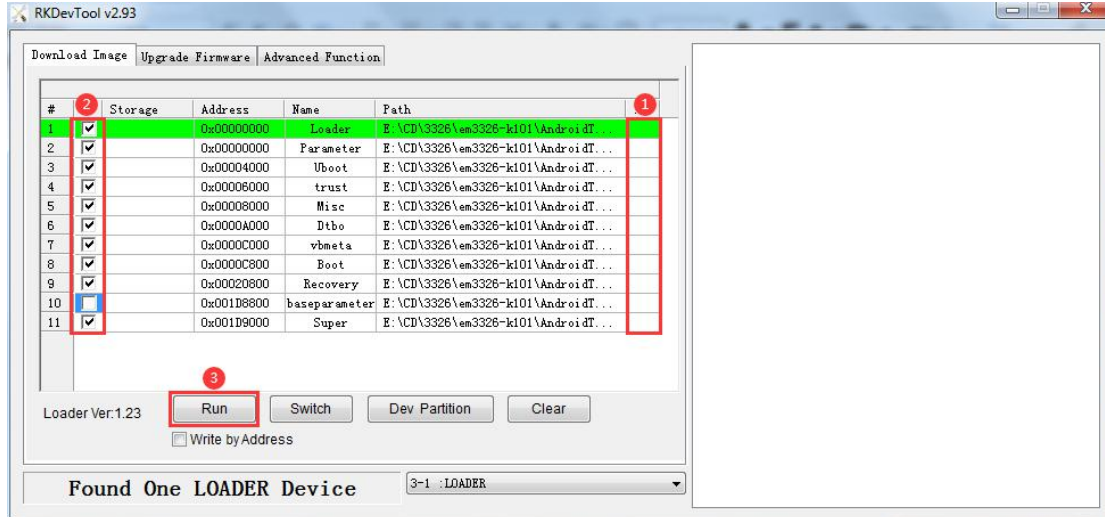


## 5.2 Burn split firmware

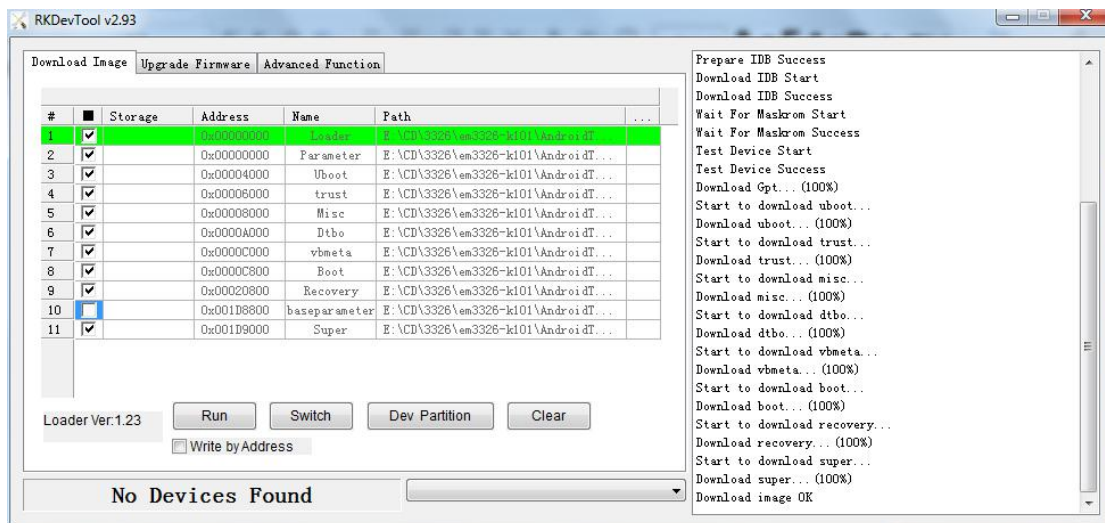
**Step 1**, click the column on the right side for the path of the file want to flash.

**Step 2**, select the checkbox on the left.

**Step 3**, click “run” to flash the image.



**Step 4**, wait for burning to complete.



## 6.Android12 Application

### 6.1 Normal display

EM3326s Android12 supports 10.1 inch MIPI lcd display.

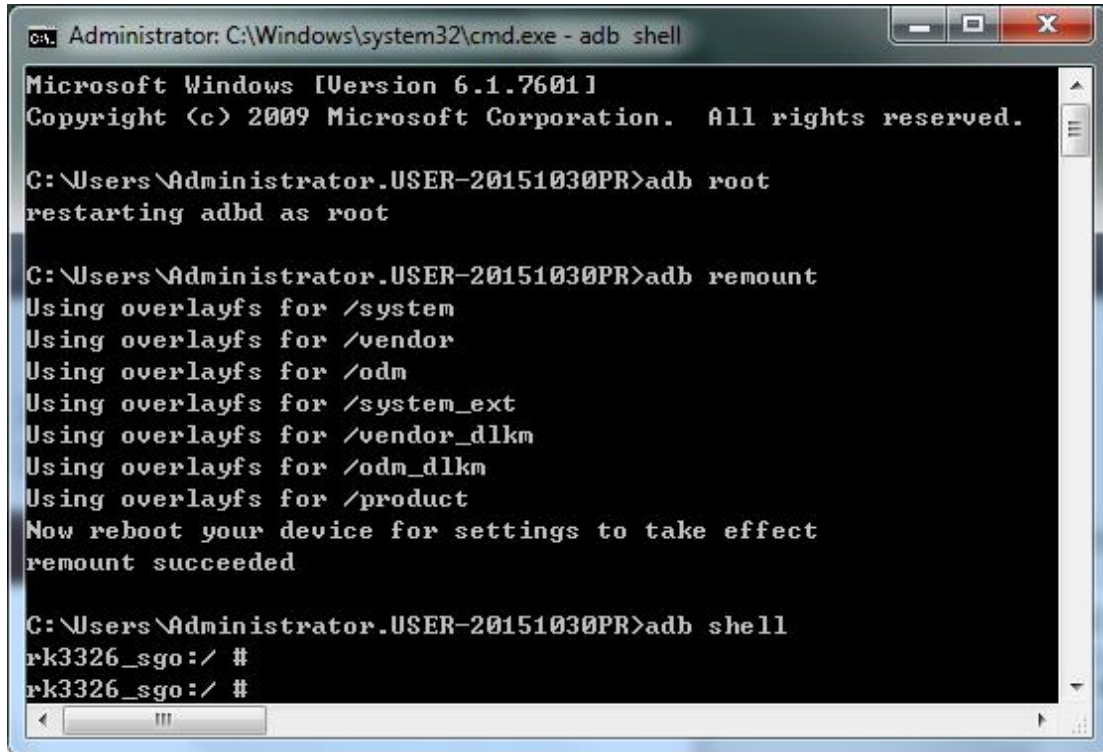
### 6.2 USB OTG

USB OTG support ADB, Type-c to

(1) ADB is the command-line debugging tool for Android, and it can use for system logs, uploading and downloading the files, installing the applications, etc. Connect the board USB OTG and PC

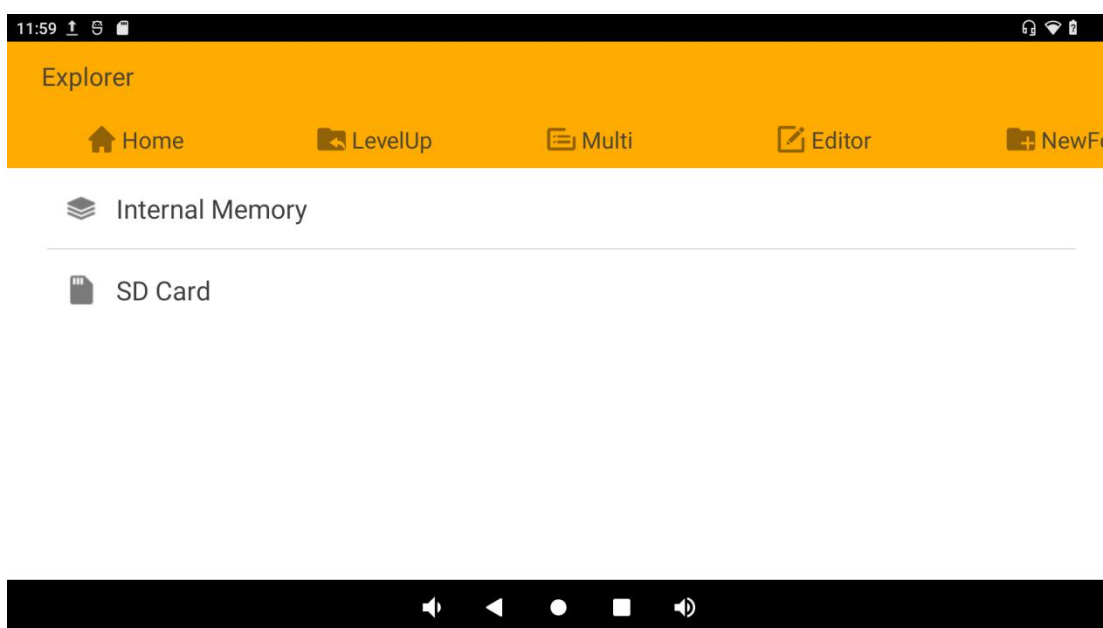
host with USB OTG data cable; Install the adb driver and commands on the windows system. Execute the follow commands to enable ADB.

```
# adb root
# adb remount
# adb shell
```



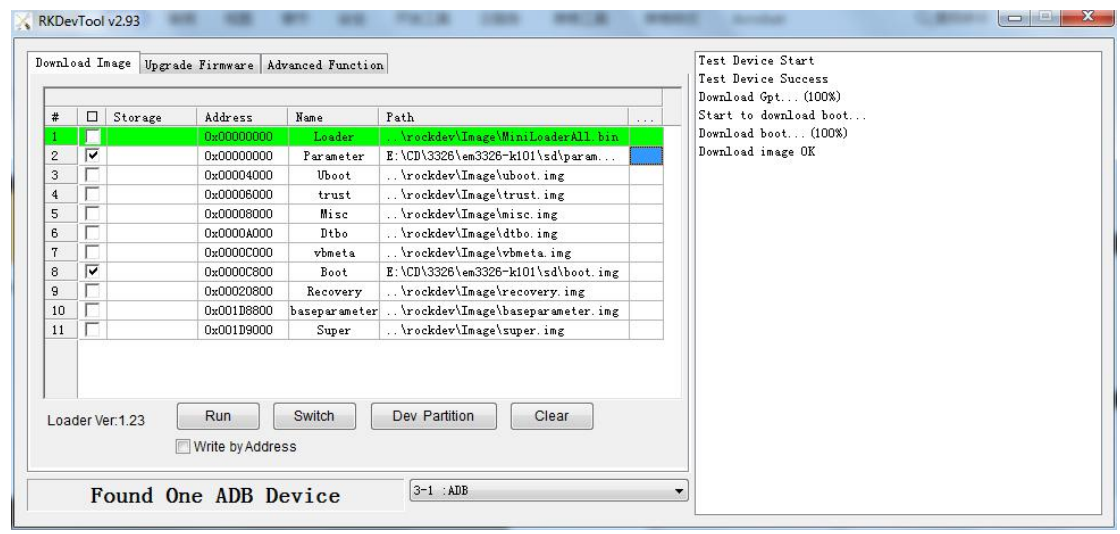
**Note:** USB OTG 2.0 multiplexed with USB Host/Ethernet/4G model, Using one of these functions requires unplugging the other device

## 6.3 SD card



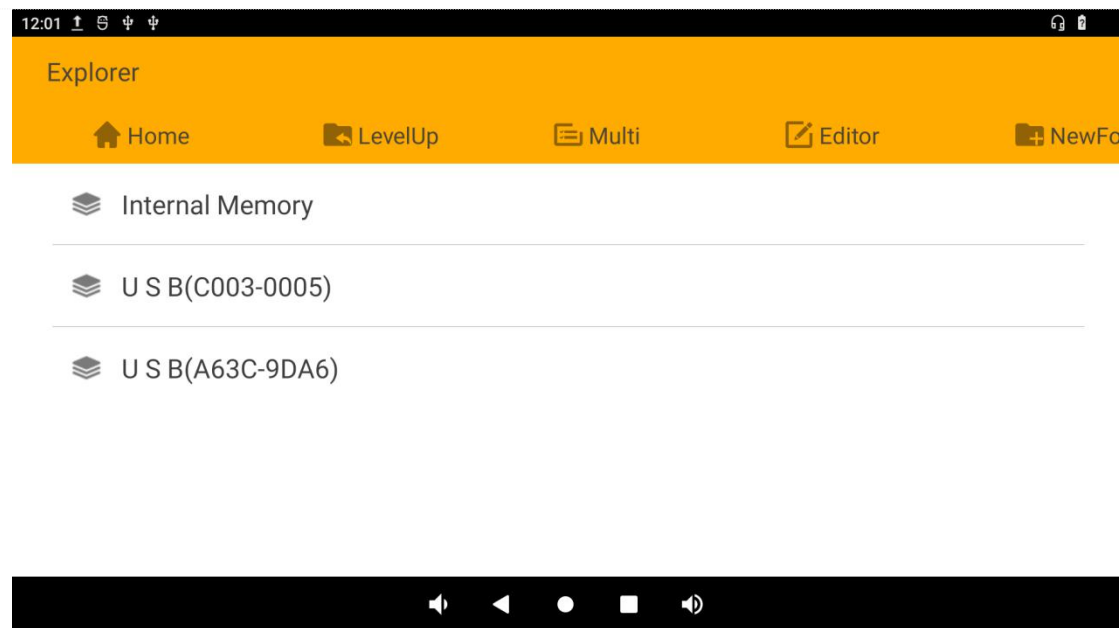
**Note:** The SD card multiplexed with debug port. If use SD card need reflash sd/boot.img and

parameter.



## 6.4 USB host

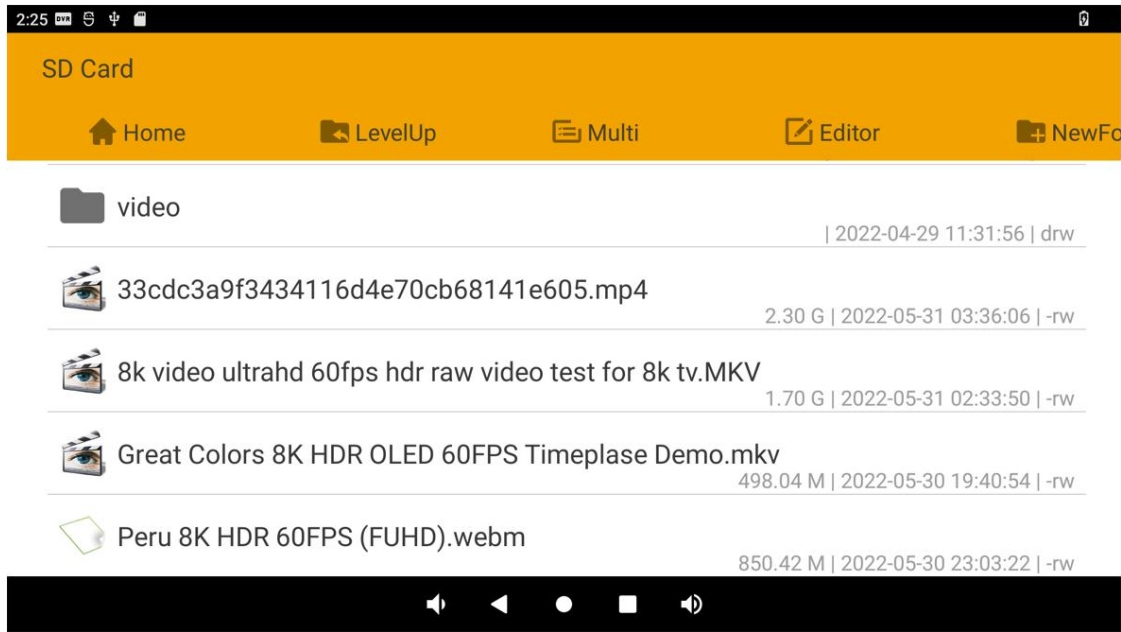
USB2.0:



The USB Host can be used to connect USB mouse, USB keyboard, U-Disk or other USB devices.

## 6.5 Video player

Copy video file to sdcard/udisk then insert it to the board, after system boot open sdcard/udisk.



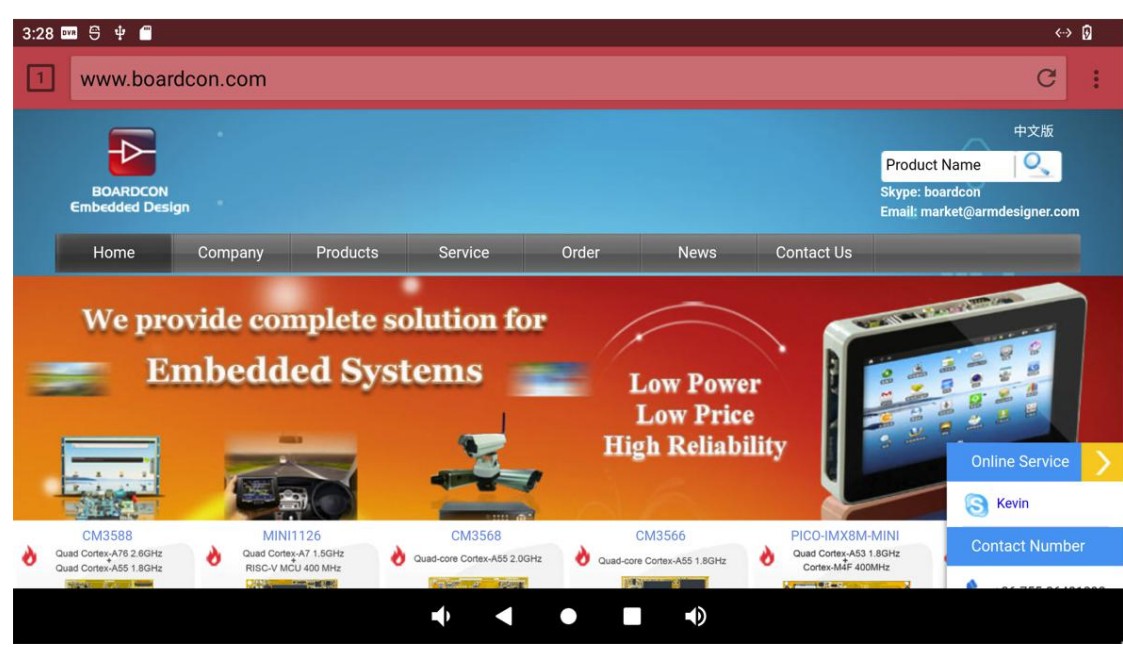
Click video file to play.



Audio output priority: headset>speaker.

## 6.6 Ethernet

```
# ifconfig  
# ping www.boardcon.com  
Or open browse www.boardcon.com to test ethernet
```

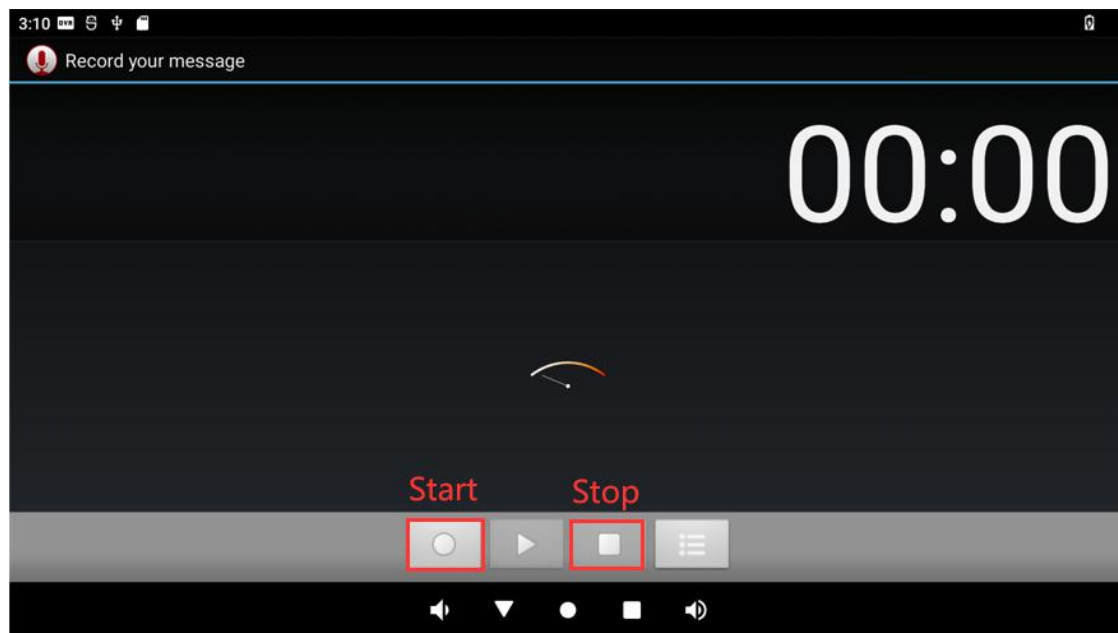


## 6.7 Record

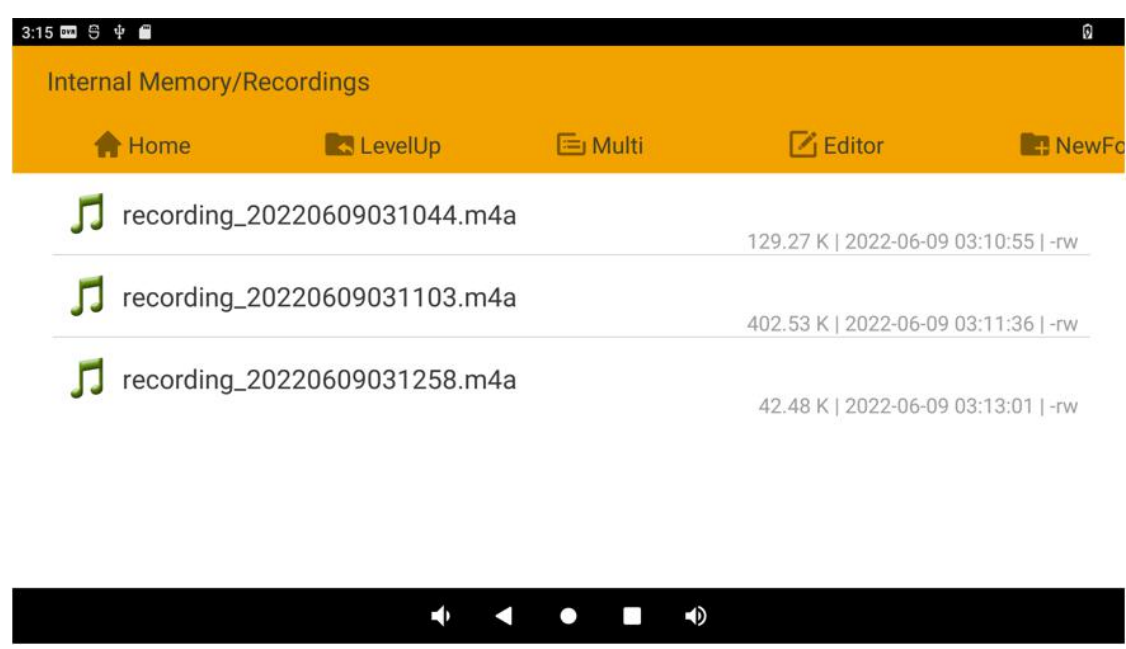
*Step 1*, open the APP **Sound Recorder** in Android.

*Step 2*, click on the APP to **start** recording, speech in front of the microphone then can record.

*Step 3*, after finish recording, click **stop** menu and select **Save** to store file.

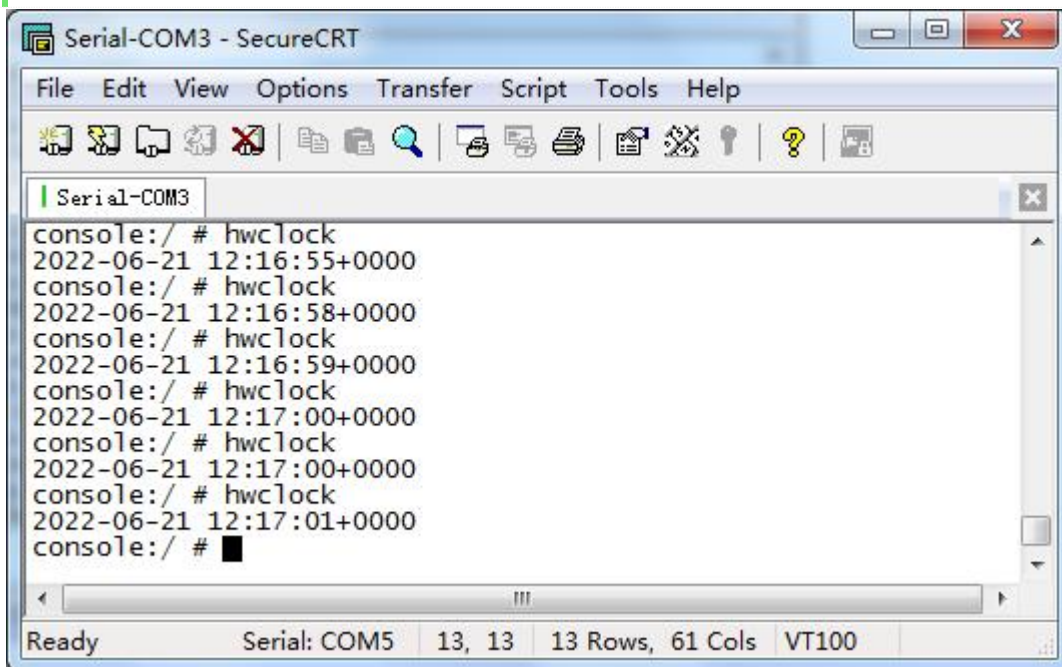


*Step 5*, the default storage path is **Internal Memory/Recordings**.



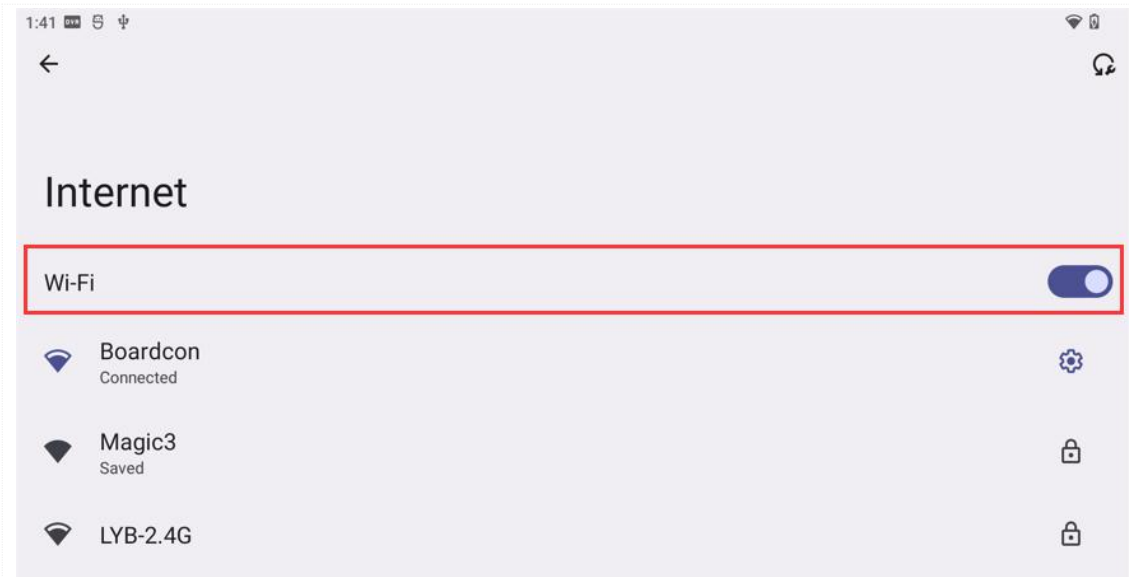
## 6.8 RTC

```
# date "2023-03-22 10:16:00"
# hwclock -w
# hwclock
```



As shown above, the RTC operates normally. If the network is connected, the RTC can synchronize with the network time. If there is a button battery on the board, the time can also be saved.

## 6.9 WiFi



After connected, user can ping URL/IP at terminal or open the browser to test Network.

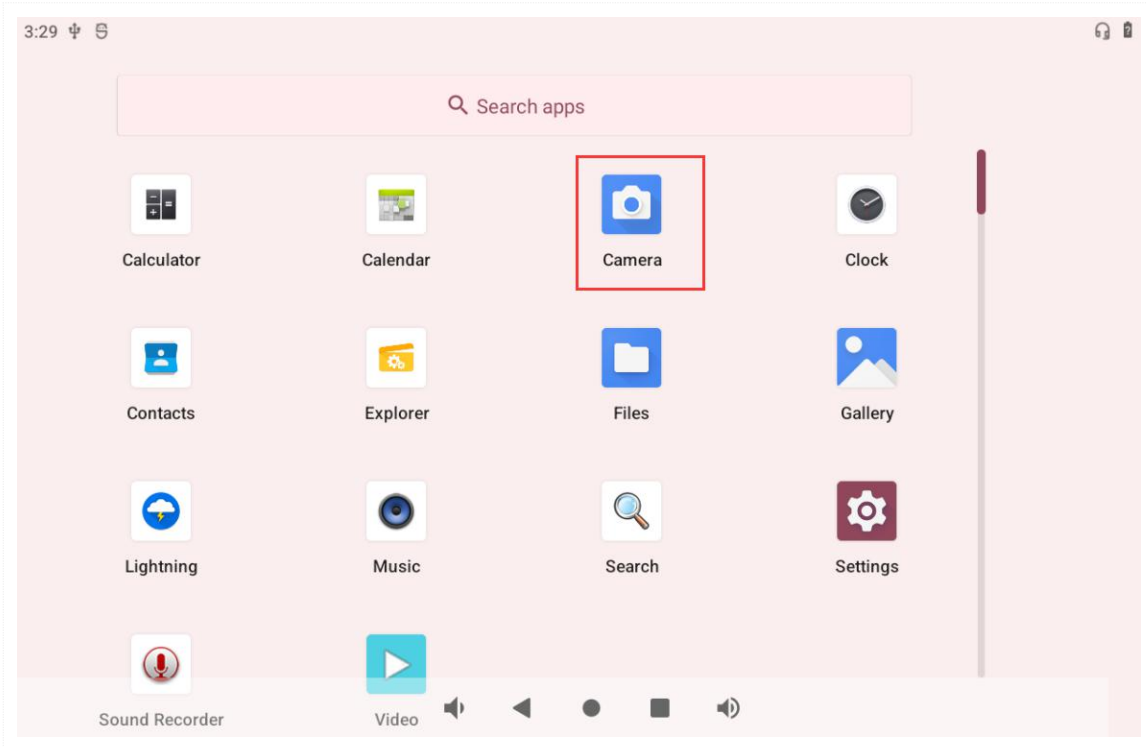
## 6.10 Bluetooth



After pairing, Bluetooth communication can be conducted between devices.

## 6.11 Camera

The camera of EM3326S is connected to CIF. Click **Camera** to test.



## 6.12 RS485

**Step 1**, connect the RS485 ports of board A and board B as follows:



**Step 2**, use ADB to push the file com into board A and board B.

```
# adb root
# adb remount
# adb push \\xx\com /system/bin // \\xx\ is the absolute path to store com files
# adb shell
# chmod 777 /system/bin/com // modify com file properties
```

**Step 3**, execute the following commands on the terminal serial ports of board A and board B respectively.

```
# echo 13 > /sys/class/gpio/export
# echo "out" > /sys/class/gpio/gpio13/direction
# echo "1" > /sys/class/gpio/gpio13/value //set RS485 for receiver
Or
# echo "0" > /sys/class/gpio/gpio13/value //set RS485 for send
# com /dev/ttyS0 115200 8 0 1
```

**Step 4**, at this time, you can test the RS485 communication.

```

Serial-COM3
Serial-COM3
console:/ #
console:/ # echo 13 > /sys/class/gpio/export
console:/ # echo "out" > /sys/class/gpio/gpio13/direction
console:/ # echo "1" > /sys/class/gpio/gpio13/value
console:/ # com /dev/ttyS0 115200 8 0 1
port = /dev/ttyS0
baudrate = 115200
cs = 8
parity = 0
stopb = 1
RECV: d
RECV: d
RECV: d
RECV: d
RECV: d
RECV: d
RECV: d
RECV: d
RECV: d
RECV: e
RECV: e
RECV: e
RECV: e
RECV: e
RECV: e
RECV: e
RECV: e
RECV: e
RECV: e
RECV: y
RECV: y
RECV: y
RECV: y
█
Ready      Serial: COM5  33, 1  33 Rows, 62 Cols  VT100  NUM

```

The method for testing RS485 is not unique, and the above method is only for reference.

## 6.13 SPI

*Step 1*, use ADB to push the spidev\_test0.0 and spidev\_test1.0 file into the test board.

```

# adb root
# adb remount
# adb push \\xx\spidev_test0.0 spidev_test1.0 /system/bin
// \\xx\ is the absolute path to store files

```

```

G:\em3326-tools>adb root
* daemon not running. starting it now at tcp:5037 *
* daemon started successfully *

G:\em3326-tools>adb remount
Using overlayfs for /system
Using overlayfs for /vendor
Using overlayfs for /odm
Using overlayfs for /system_ext
Using overlayfs for /vendor_dlkm
Using overlayfs for /odm_dlkm
Using overlayfs for /product
Now reboot your device for settings to take effect
remount succeeded

G:\em3326-tools>
G:\em3326-tools>adb push spidev_test0.0 spidev_test1.0 /system/bin
spidev_test0.0: 1 file pushed. 0.0 MB/s (4220 bytes in 0.093s)
spidev_test1.0: 1 file pushed. 0.3 MB/s (4220 bytes in 0.015s)
2 files pushed. 0.1 MB/s (8440 bytes in 0.122s)

G:\em3326-tools>
  
```

**Step 2**, short circuit SPI0\_MOSI and SPI0\_MISO (CON6 PIN3&PIN4) of SPI0. short circuit SPI1\_MOSI and SPI1\_MISO (CON7 PIN3&PIN4) of SPI1.

**Step 3**, SPI0 test: Execute follow command to view changes of SPI values when not short circuit SPI0\_MOSI & SPI0\_MISO and short circuit SPI0\_MOSI & SPI0\_MISO.

```
# spidev_test0.0
```

```

console:/ # spidev_test0.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF

console:/ #
console:/ # spidev_test0.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
40 00 00 00 00 95
FF FF FF FF FF FF
FF FF FF FF FF FF
DE AD BE EF BA AD
F0 0D

console:/ #
  
```

**Step 4**, SPI0 test: Execute follow command to view changes of SPI values when not short circuit SPI1\_MOSI & SPI1\_MISO and short circuit SPI1\_MOSI & SPI1\_MISO.



```
# spidev_test0.0
```

```
console:/ # spidev_test1.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF

console:/ # spidev_test1.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
40 00 00 00 00 95
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
DE AD BE EF BA AD
F0 0D
console:/ # █
```

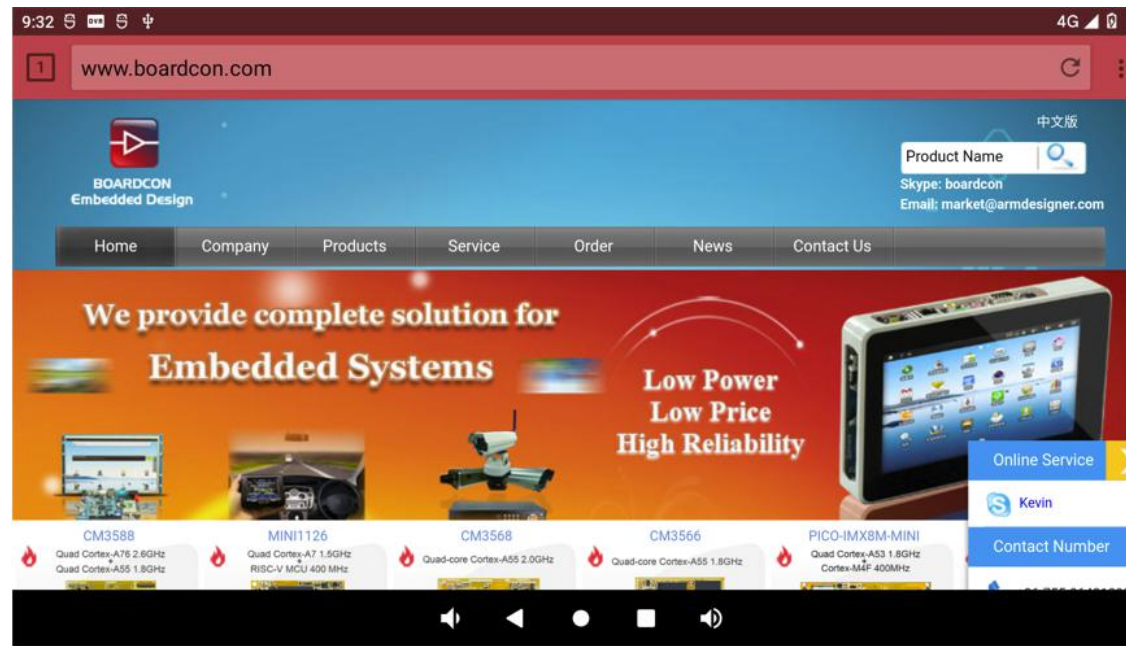
## 6.14 4G

*Step 1*, insert 4G module to PCIe slot (4G model:EC20).

*Step 2*, connect antenna and insert SIM card.

*Step 3*, the default connection is 4G network after power on.

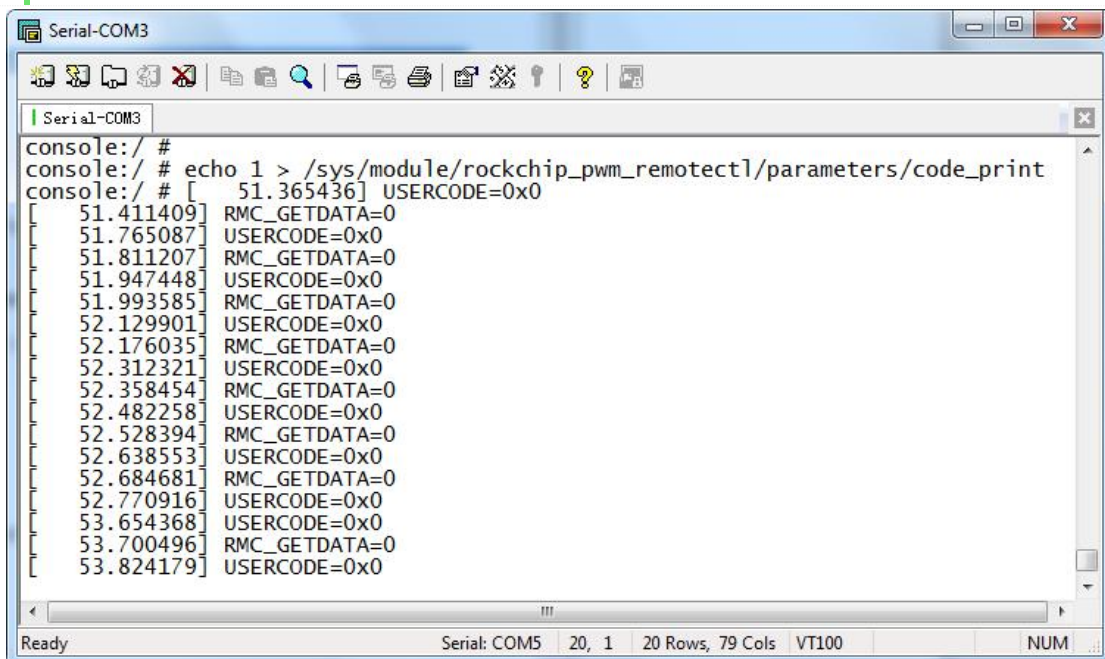
[www.boardcon.com](http://www.boardcon.com)



## 6.15 IR

Connect IR receiver to the IR port, execute the following command to turn on IR receiving data information printing:

```
# echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```



# Debian11

## 1.Compiler Environment

It is recommended to use Ubuntu 20.04 or above. If you encounter an error during compilation, you can check the error message and install the corresponding software packages accordingly. Other Linux versions may need to adjust the software package accordingly. In addition to the system requirements, there are other hardware and software requirements.

Hardware requirements	Software requirements
64-bit system, hard disk space should be greater than 40G. If you do multiple builds, you will need more hard drive space.	Ubuntu 20.04 system

## 2.Install Tools

The contents of this directory only provide the software package installation commands that are needed to build the compiled SDK environment. Please install other tools such as samba and ssh yourself.

PC OS: ubuntu 20.04 system

Network: online

Permission: root

```
# sudo apt-get install binfmt-support bison bzip2 chrpath cmake \
  cpp-aarch64-linux-gnu device-tree-compiler diffstat expat expect \
  fakeroot flex g++ g++-aarch64-linux-gnu gawk gcc gcc-multilib git \
  g++-multilib gpgv2 libgucharmap-2-90-dev liblz4-tool libssl-dev live-build \
  make ncurses-dev patchelf qemu-user-static ssh texinfo unzip
```

## 3.Compile Source

**Step 1**, unzip the source

```
$ tar xvf em3326s_linux5.1_23_07_15.tar.gz
$ cd rk3326_linux5.1
```

**Step 2**, configure the compiled board

```
$ ./build.sh lunch
```

```
wuguankang@boardcon:~/opt/rk3326_linux5.1_gen_rk4.1/rk3326_linux5.1$ ./build.sh lunch
```

Log saved at

```
/home/wuguankang/opt/rk3326_linux5.1_gen_rk4.1/rk3326_linux5.1/output/log/2023-06-13_11-42-18
```

Pick a defconfig:

1. rockchip\_defconfig



2. rockchip\_rk3326\_evb\_lp3\_v10\_32bit\_defconfig
3. rockchip\_rk3326\_evb\_lp3\_v10\_defconfig
4. rockchip\_rk3326\_evb\_lp3\_v12\_32bit\_defconfig
5. rockchip\_rk3326\_evb\_lp3\_v12\_defconfig
6. rockchip\_rk3326\_robot64\_defconfig
7. rockchip\_rk3326\_robot64\_no\_gpu\_defconfig

Which would you like? [1]: 3 // rockchip\_rk3326\_evb\_lp3\_v10\_defconfig

Switching to defconfig:

```
/home/wuguankang/opt/rk3326_linux5.1_gen_rk4.1/rk3326_linux5.1/output/rockchip_rk3326_evb_lp3_v10_defconfig
```

#### Step 3, compile uboot

```
$ ./build.sh uboot
```

#### Step 4, compile the kernel

```
$ ./build.sh kernel
```

#### Step 5, compile recovery

```
$ ./build.sh recovery
```

#### Step 6, compile debian

Description: re-install the depend packages

```
# cd debian
# sudo apt-get install binfmt-support qemu-user-static live-build
# sudo dpkg -i ubuntu-build-service/packages/*
# sudo apt-get install -f
```

Run the following command with the root permission:

```
# RELEASE=bullseye TARGET=desktop ARCH=arm64 ./mk-base-debian.sh
# VERSION=debug ARCH=arm64 ./mk-rootfs-bullseye.sh
# ./mk-image.sh
```

After compilation, generate linaro-rootfs.img in the Debian directory.

#### Step 7, generate and check firmwares

```
$ ./build.sh firmware
```

```
export RK_ROOTFS_SYSTEM=debian
```

#### Step 8, build update image

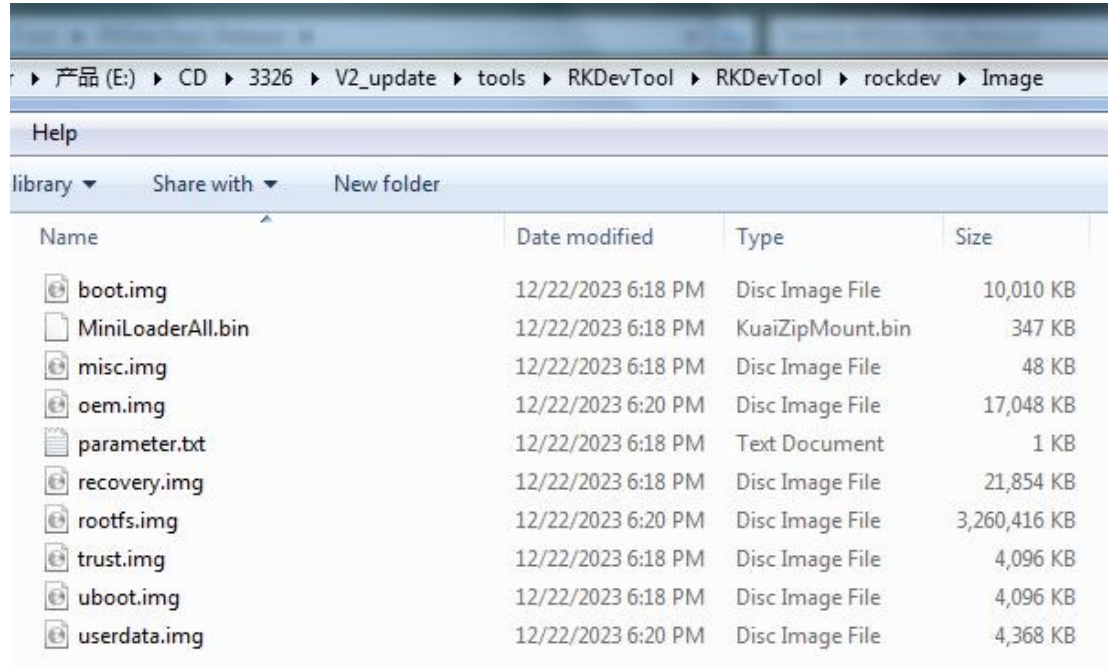
```
$ ./build.sh updateimg
```

Images and update.img are generated in rockdev/ directory.

## 4. Images Operation

### 4.1 Pack image

*Step 1*, copy the firmware file to be packaged to windows RKDevTool/rockdev/Image.



*Step 2*, enter RKDevTool/rockdev/, double-click to run **mkupdate.bat**

```

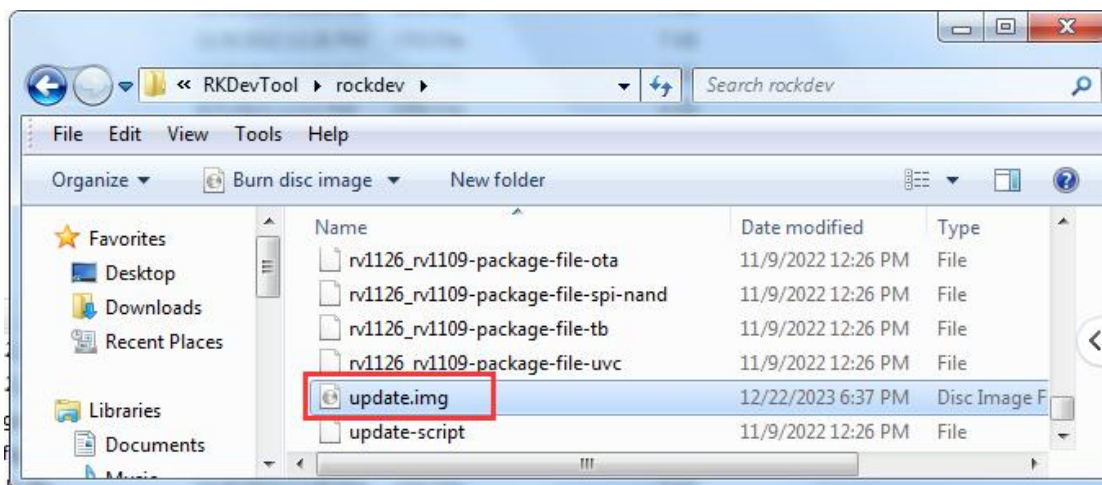
C:\> Android Firmware Package Tool v1.65
E:\CD\3326\U2_update\tools\RKDevTool\RKDevTool\rockdev>afptool -pack ./ Image\update.img
Android Firmware Package Tool v1.65
----- PACKAGE -----
Add file: .\package-file
Add file: .\package-file done,offset=0x800,size=0x28a,userspace=0x1
Add file: .\Image\MiniLoaderAll.bin
Add file: .\Image\MiniLoaderAll.bin done,offset=0x1000,size=0x5694e,userspace=0xae
Add file: .\Image\parameter.txt
Add file: .\Image\parameter.txt done,offset=0x58000,size=0x209,userspace=0x1
Add file: .\Image\trust.img
Add file: .\Image\trust.img done,offset=0x58800,size=0x40000,userspace=0x801
Add file: .\Image\uboot.img
Add file: .\Image\uboot.img done,offset=0x459000,size=0x40000,userspace=0x801
Add file: .\Image\misc.img
Add file: .\Image\misc.img done,offset=0x859800,size=0xc000,userspace=0x19
Add file: .\Image\boot.img
Add file: .\Image\boot.img done,offset=0x866000,size=0x9c6800,userspace=0x138e
Add file: .\Image\recovery.img
Add file: .\Image\recovery.img done,offset=0x122d000,size=0x1557800,userspace=0x2ab0
Add file: .\Image\rootfs.img
Add file: .\Image\rootfs.img done,offset=0x2785000,size=0xc700000,userspace=0x18e001
Add file: .\Image\oem.img
Add file: .\Image\oem.img done,offset=0xc9785800,size=0x10a6000,userspace=0x214d
Add file: .\Image\userdata.img
Add file: .\Image\userdata.img done,offset=0xca82c000,size=0x444000,userspace=0x889
Add CRC...
Make firmware OK!
----- OK -----

E:\CD\3326\U2_update\tools\RKDevTool\RKDevTool\rockdev>RKImageMaker.exe -RK3326 Image\MiniL
*****RKImageMaker ver 1.66 *****
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!

E:\CD\3326\U2_update\tools\RKDevTool\RKDevTool\rockdev>rem update.img is new format, Image\
E:\CD\3326\U2_update\tools\RKDevTool\RKDevTool\rockdev>del Image\update.img

E:\CD\3326\U2_update\tools\RKDevTool\RKDevTool\rockdev>pause
Press any key to continue . . .
  
```

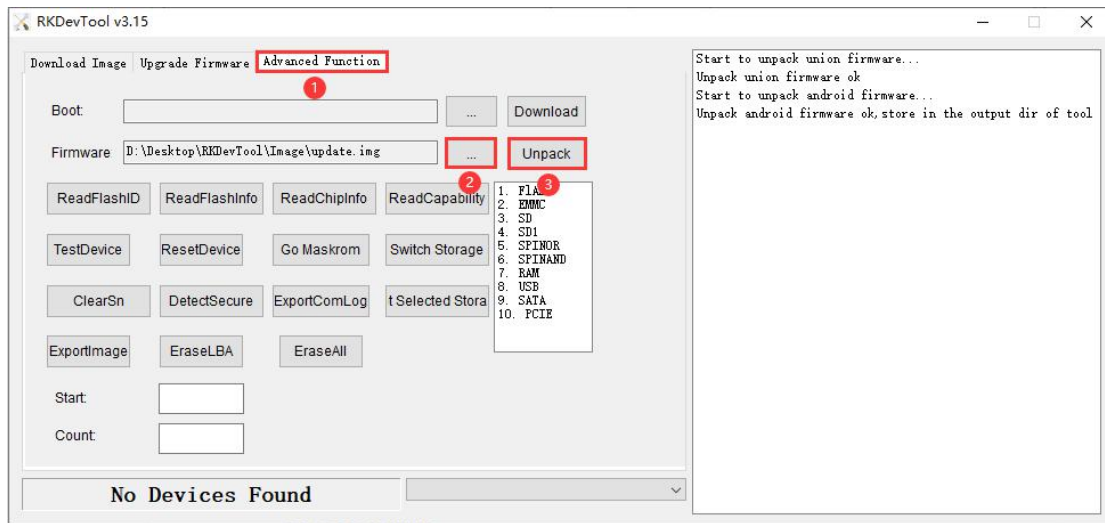
Step 3, the **update.img** will be generated in rockdev/ directory.



## 4.2 Unzip firmware

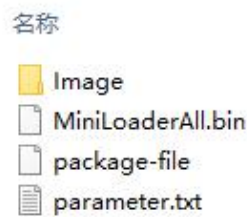
Step 1, open \RKDevTool\RKDevTool\_Release\RKDevTool.exe.

**Step 2**, click **Advanced Function** -> **Firmware**, select **update.img**, then click **Unpack** to unzip.



**Step 3**, the unzip files will be generated in `\RKDevTool\RKDevTool_Release\Output\Android` and `\RKDevTool\RKDevTool_Release\Output\Android\Image` directory.

RKDevTool > RKDevTool\_Release > Output > Android >



CD > 3326 > V2\_update > tools > RKDevTool > RKDevTool > RKDevTool\_Release > Output > Android > Image

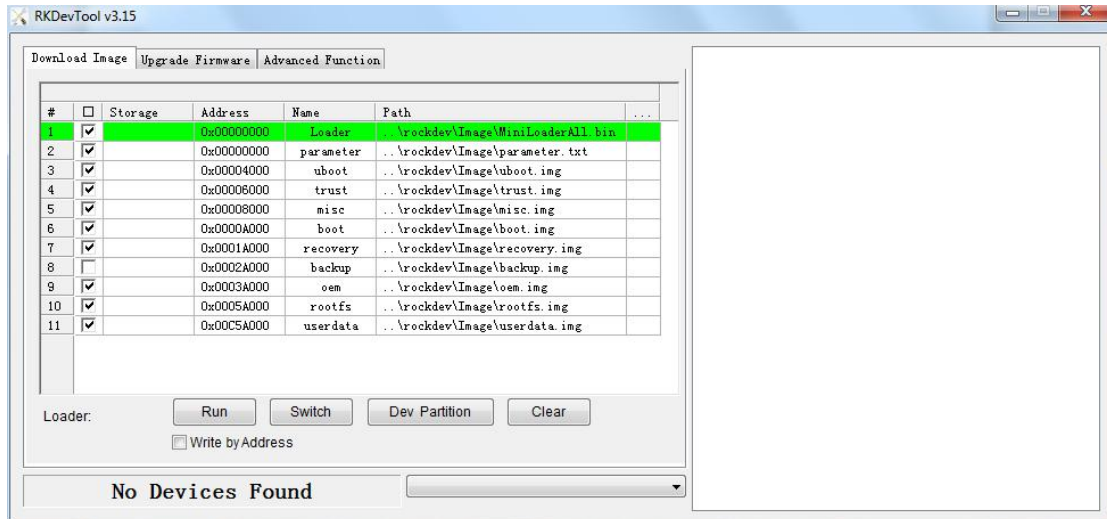
Name	Date modified	Type	Size
boot.img	12/22/2023 7:23 PM	Disc Image File	10,010 KB
misc.img	12/22/2023 7:23 PM	Disc Image File	48 KB
oem.img	12/22/2023 7:25 PM	Disc Image File	17,048 KB
recovery.img	12/22/2023 7:23 PM	Disc Image File	21,854 KB
rootfs.img	12/22/2023 7:25 PM	Disc Image File	3,260,416 KB
trust.img	12/22/2023 7:23 PM	Disc Image File	4,096 KB
uboot.img	12/22/2023 7:23 PM	Disc Image File	4,096 KB
userdata.img	12/22/2023 7:25 PM	Disc Image File	4,368 KB

## 5. Burn Images

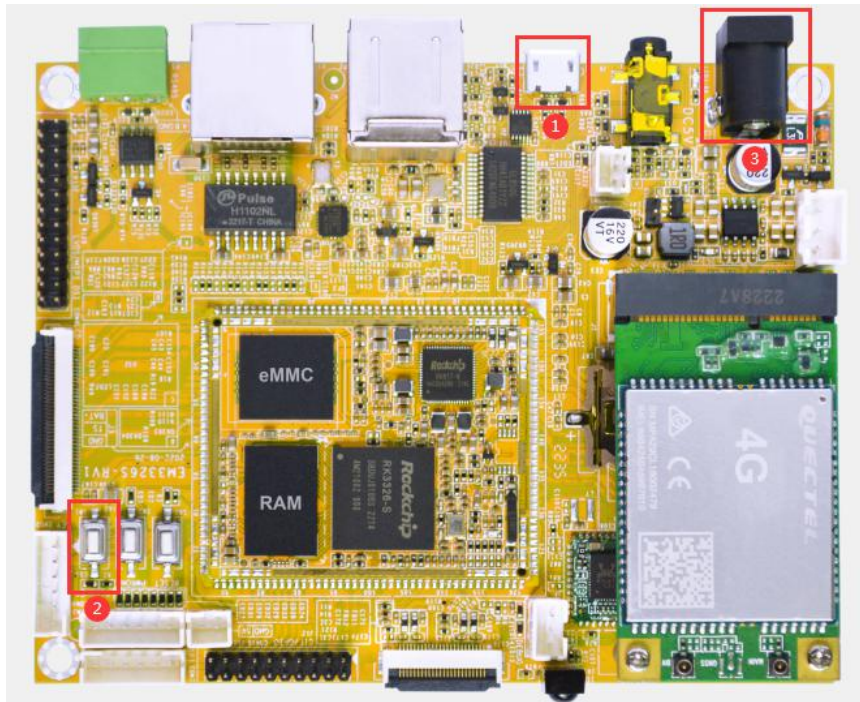
### 5.1 Burn update firmware

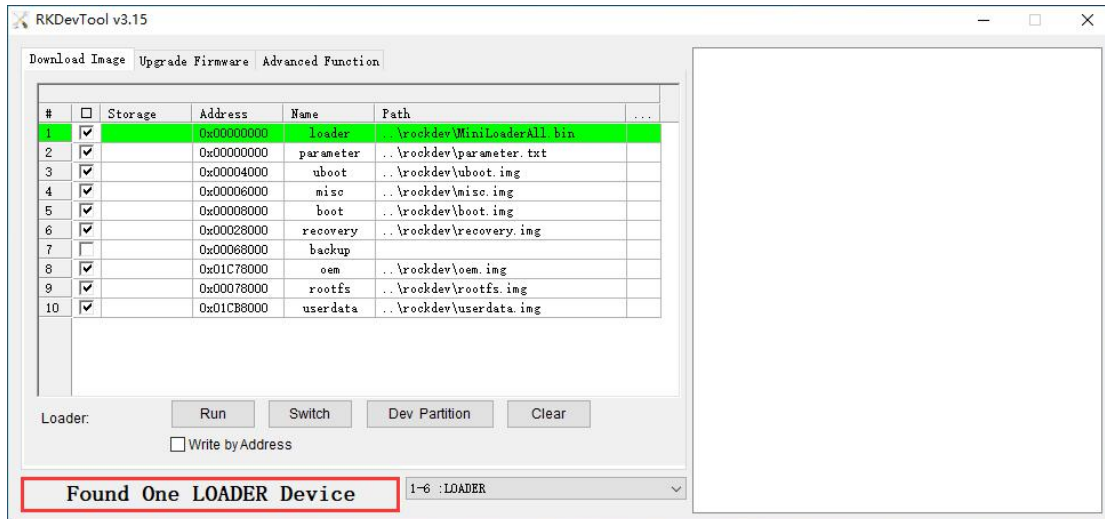
**Step 1**, unzip RKDevTool.rar on Windows.

**Step 2**, open `\RKDevTool\RKDevTool_Release\RKDevTool.exe`.

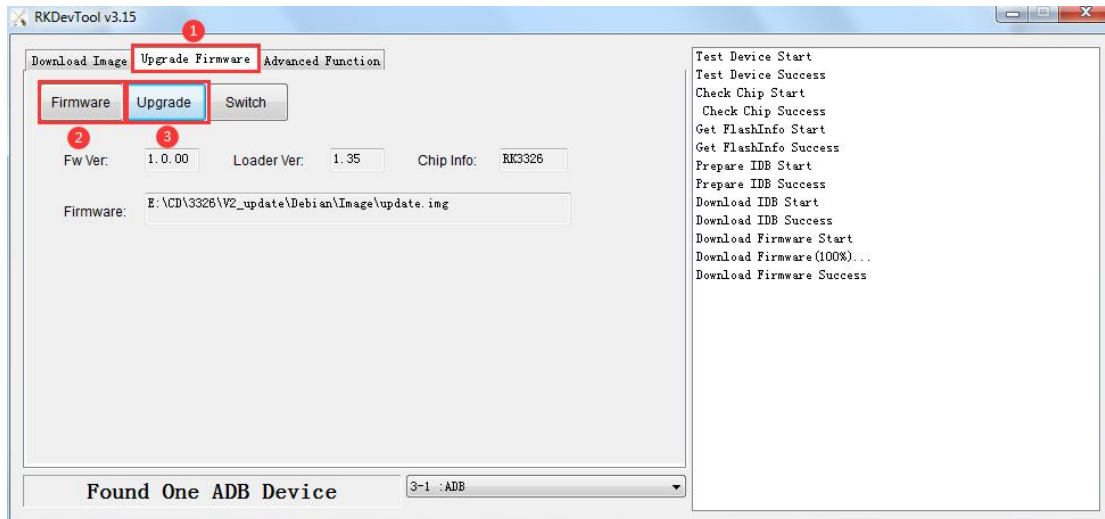


**Step 3**, connect PC and development board with Micro USB cable, keep pressing the **Recovery Key** and power the board until the windows PC shows **Found one LOADER Device** release the **Recovery Key**.





Step 4, click **Upgrade Firmware** -> **Firmware**, select **update.img**, then click **Upgrade** to flash.

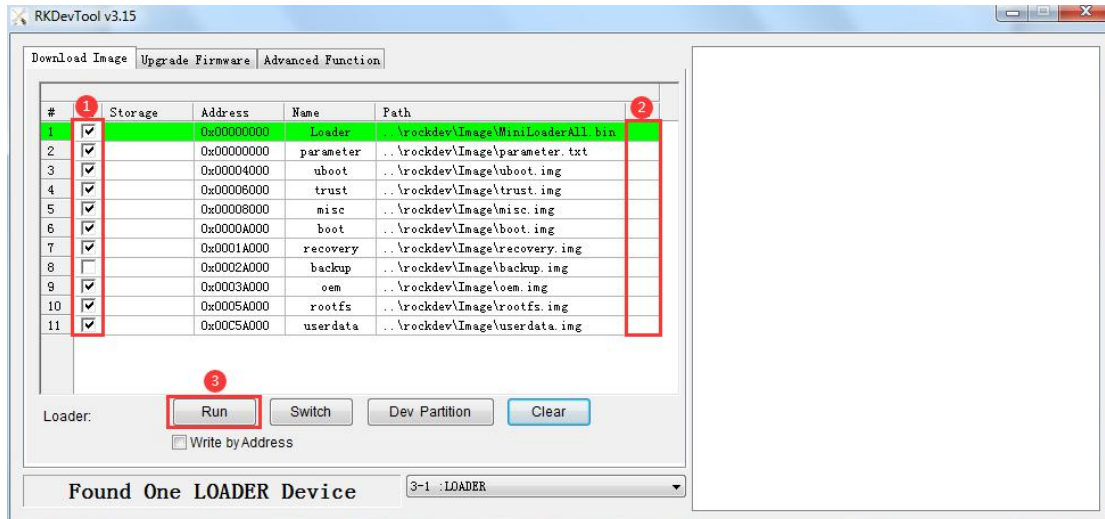


## 5.2 Burn split firmware

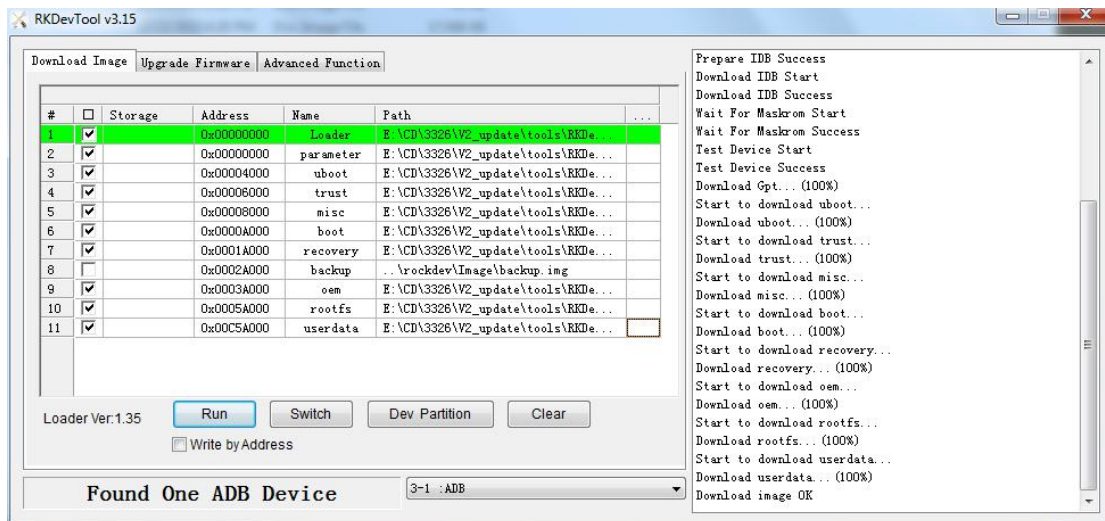
Step 1, select the checkbox on the left.

Step 2, click the column on the right side for the path of the file want to flash.

Step 3, click **run** button to flash the image.



Step 4, wait for the completion of burning.



## 6. Debian11 Application

### 6.1 Display

EM3326S debian11 support 10.1 inch MIPI LCD display or 4.3 inch RGB LCD display. Function multiplexing cannot be displayed at the same time by switching through the following resistors:

#### LVDS/MIPI display:

Soldering: R137, R138, R139, R140, R135, R136, R141, R142, R143, R144 solder oR.

NC: R145, R146, R147, R148, R149, R150, R151, R152, R153, R154.

#### RGB display:

Soldering: R145, R146, R147, R148, R149, R150, R151, R152, R153, R154 solder 0R.

NC: R137, R138, R139, R140, R135, R136, R141, R142, R143, R144.

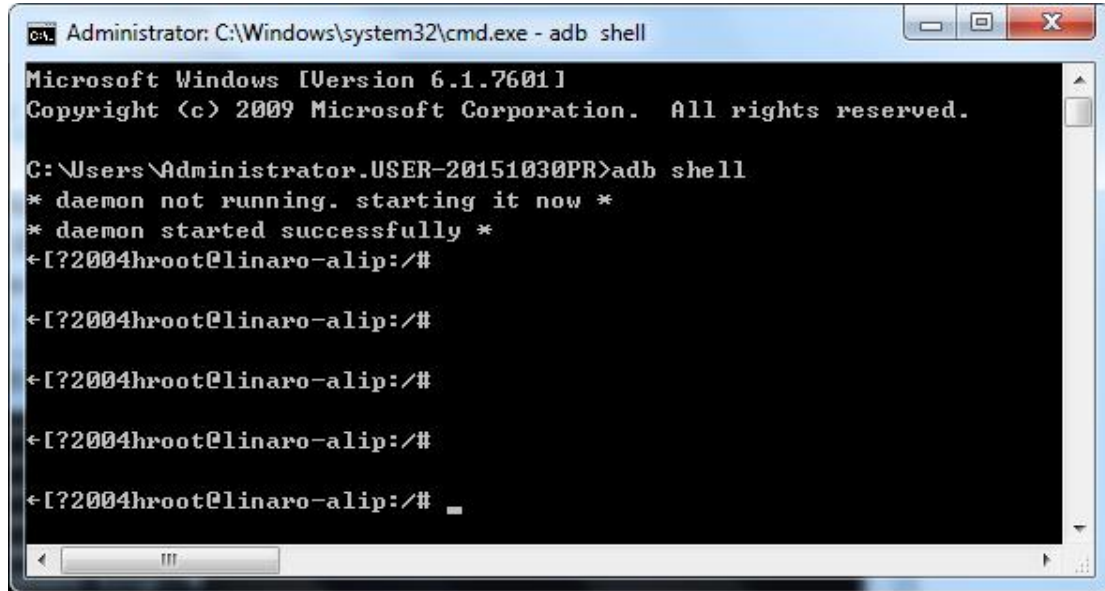
### 6.2 USB OTG

USB OTG support ADB.

(1) ADB is the command-line debugging tool for debian, and it can use for system logs, uploading

and downloading the files, installing the applications, etc. Connect the board USB OTG and PC host with USB OTG data cable; Install the adb driver and commands on the windows system. Execute the follow commands to enable ADB.

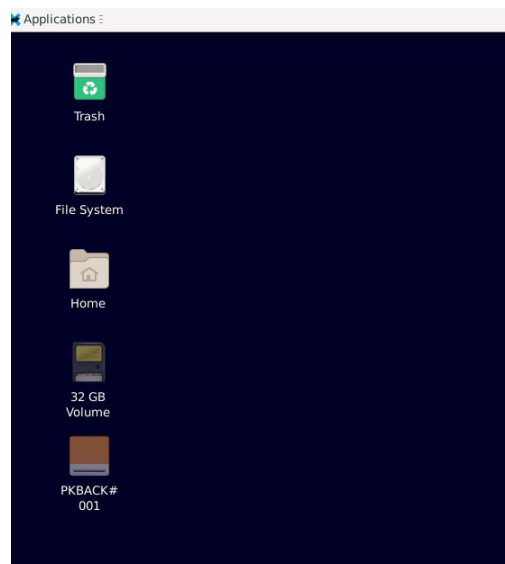
```
# adb shell
```



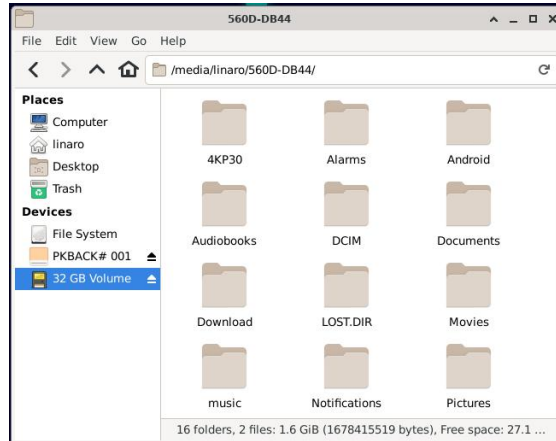
**Note:** USB OTG 2.0 multiplexed with USB Host/Ethernet/4G model, Using one of these functions requires unplugging the other device.

## 6.3 SD/USB device

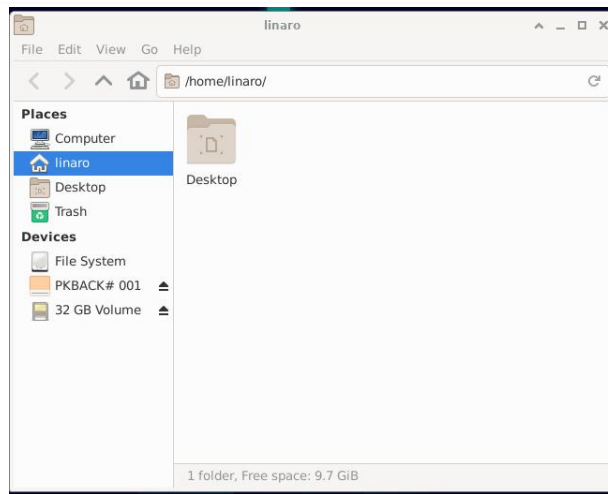
After inserting an SD card or USB device, a corresponding icon will be formed on the desktop. Automatic mounting of SD/USB storage devices can be achieved through the following two methods.



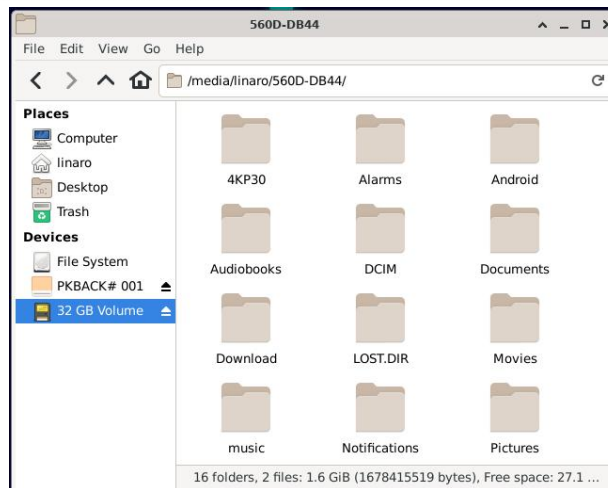
**Step 1,** double click on the desktop icon and the SD/USB storage device will automatically mount:



**Step 2**, double click on the **Home** icon to enter the main directory:



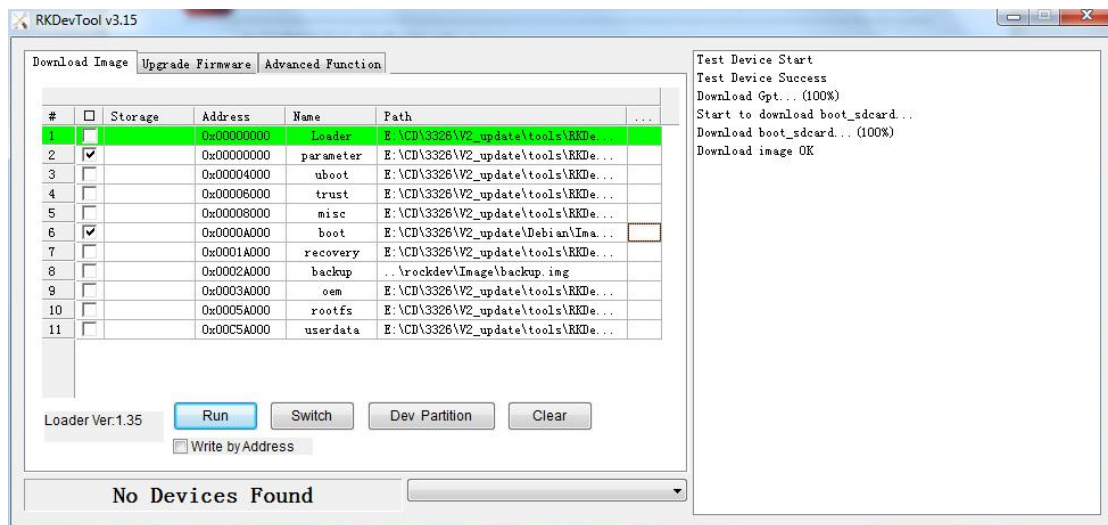
Then click on the **Devices** under the devices directory to achieve automatic mounting:



```
root@linaro-alip:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        5.9G  4.1G  1.6G   73% /
devtmpfs        979M   8.0K 979M   1% /dev
tmpfs           990M   18M  972M   2% /dev/shm
tmpfs           396M   1.1M 395M   1% /run
tmpfs           5.0M   4.0K 5.0M   1% /run/lock
tmpfs           990M   16K  990M   1% /tmp
/dev/mmcblk0p7   60M   12M   44M  22% /oem
/dev/mmcblk0p9  1.1G   36K 1007M   1% /userdata
tmpfs           198M   44K  198M   1% /run/user/1000
/dev/sda1        3.6G   3.3G  305M  92% /media/linaro/U-DISK
/dev/sdb1        29G   26G   3.6G  88% /media/linaro/88E7-4C8E
```

The USB Host can be used to connect USB mouse, USB keyboard, U-Disk or other USB devices.

**Note:** The SD card multiplexed with debug port. If use SD card need reflash sd/boot.img and parameter.



## 6.4 Audio

**Step 1**, view sound card:

```
# cat /proc/asound/cards
```

```
root@linaro-alip:/# cat /proc/asound/cards
0 [rockchiprk817 ]: rockchip-rk817 - rockchip-rk817
rockchip-rk817
root@linaro-alip:/# █
```

**Step 2**, headset recording:

```
# amixer -c 0 cset numid=2 1
```

```
# arecord -Dhw:0,0 -f cd record.wav
```

```

root@linaro-alip:/# amixer -c 0 cset numid=2 1
numid=2,iface=MIXER,name='Capture MIC Path'
; type=ENUMERATED,access=rw-----,values=1,items=4
; Item #0 'MIC OFF'
; Item #1 'Main Mic'
; Item #2 'Hands Free Mic'
; Item #3 'BT Sco Mic'
; values=1
root@linaro-alip:/# arecord -Dhw:0,0 -f cd record.wav
Recording WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
^CAborted by signal Interrupt...
root@linaro-alip:/# ls
bin      lib      'record.wav' '$'\240'      sha256sum.README  userdata
boot    lost+found 'record.wav' '$'\240\240'  sha256sum.txt     usr
data    media    rockchip-test             srv               var
dev      mnt      root                      sys              vendor
etc      oem      run                       system
home    opt      sbin                      tmp
info    proc     sdcard                    udisk
root@linaro-alip:/#

```

**Step 3, play audio:**

```

# amixer -c 0 cset numid=1 3
# aplay -Dhw:0,0 record.wav // Headset
# amixer -c 0 cset numid=1 2
# aplay -Dhw:0,0 record.wav // Speaker OUT

```

```

serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
root@linaro-alip:/# amixer -c 0 cset numid=1 3
numid=1,iface=MIXER,name='Playback Path'
; type=ENUMERATED,access=rw-----,values=1,items=11
; Item #0 'OFF'
; Item #1 'RCV'
; Item #2 'SPK'
; Item #3 'HP'
; Item #4 'HP_NO_MIC'
; Item #5 'BT'
; Item #6 'SPK_HP'
; Item #7 'RING_SPK'
; Item #8 'RING_HP'
; Item #9 'RING_HP_NO_MIC'
; Item #10 'RING_SPK_HP'
; values=3
root@linaro-alip:/# aplay -Dhw:0,0 record.wav
Playing WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@linaro-alip:/# amixer -c 0 cset numid=1 2
numid=1,iface=MIXER,name='Playback Path'
; type=ENUMERATED,access=rw-----,values=1,items=11
; Item #0 'OFF'
; Item #1 'RCV'
; Item #2 'SPK'
; Item #3 'HP'
; Item #4 'HP_NO_MIC'
; Item #5 'BT'
; Item #6 'SPK_HP'
; Item #7 'RING_SPK'
; Item #8 'RING_HP'
; Item #9 'RING_HP_NO_MIC'
; Item #10 'RING_SPK_HP'
; values=2
root@linaro-alip:/# aplay -Dhw:0,0 record.wav
Playing WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@linaro-alip:/# █
Ready Serial: COM31, 150000 35, 21 35 Rows, 81 Cols Linux CAP NUM

```

## 6.5 Video player

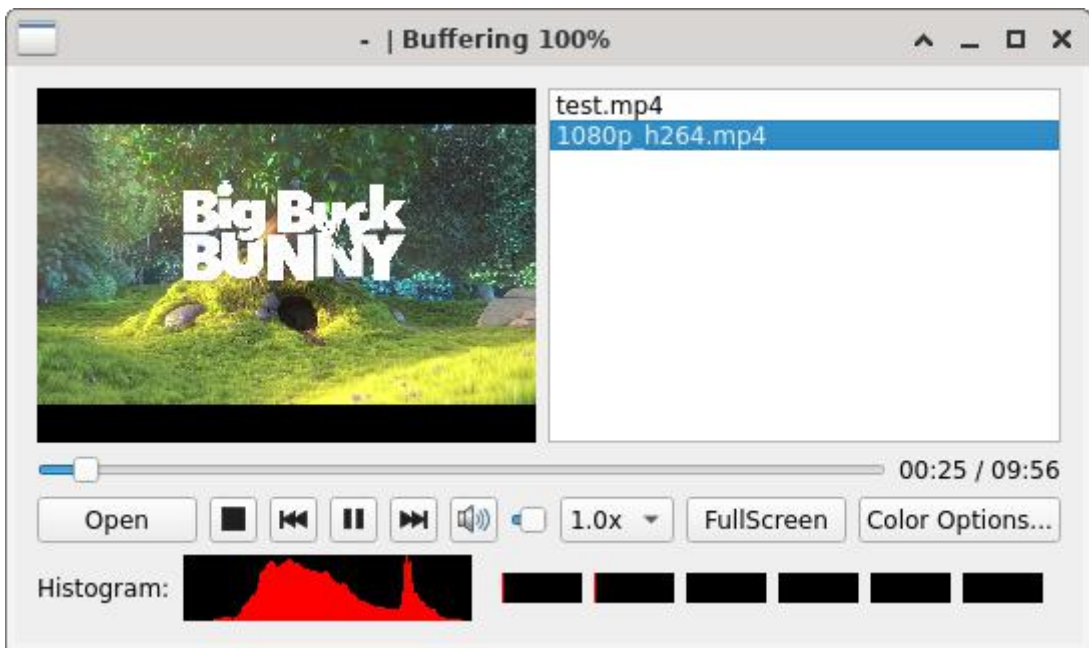
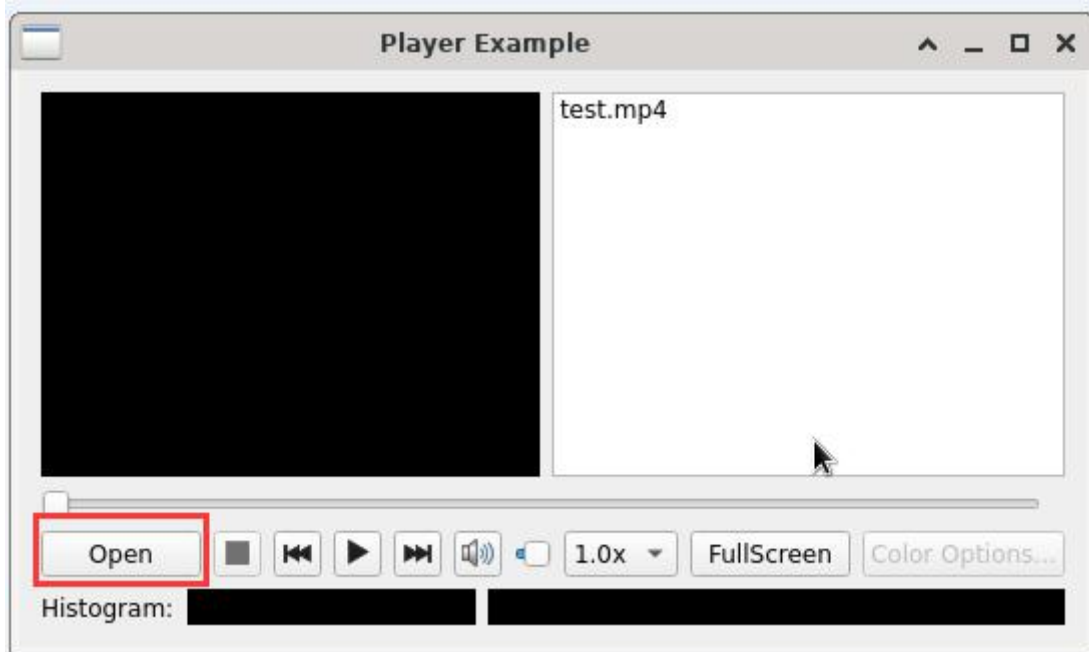
```

# /rockchip-test/video/test_dec-qt.sh arm64 // Open the player

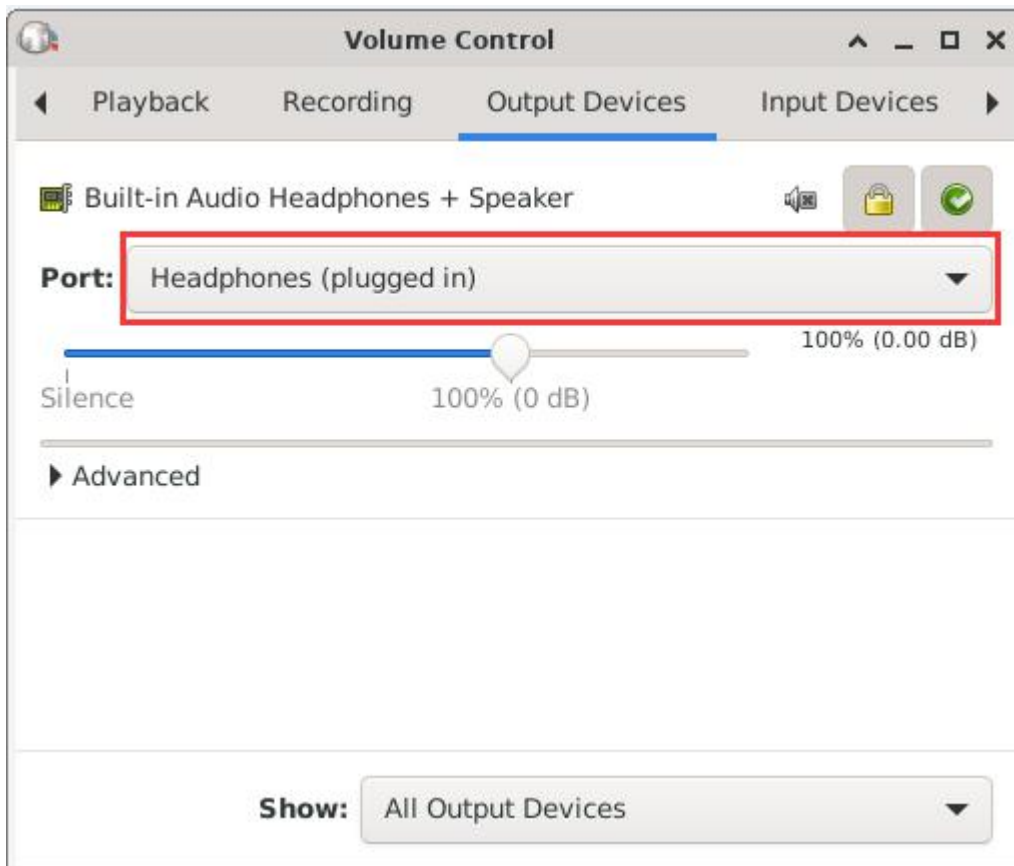
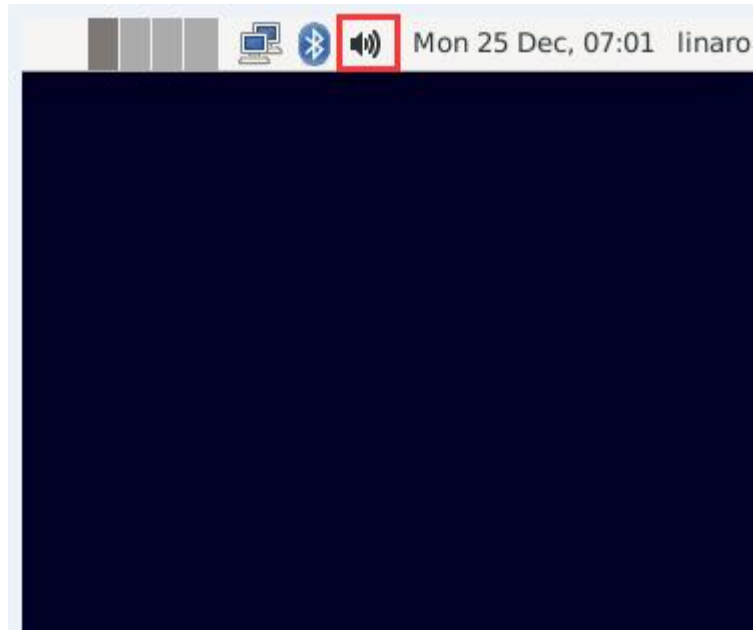
```

```
root@linaro-alip:/# /rockchip-test/video/test_dec-qt.sh arm64
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
Supported audio roles:
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
```

Click "Open" to select the file for playback.



After the video opens, click on the **speaker icon** in the upper right corner to select the audio output channel:



## 6.6 Ethernet

```
# ifconfig
```



```
root@linaro-alip:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.195 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::dcb2:7bbd:ede2:8a40 prefixlen 64 scopeid 0x20<link>
ether 82:17:69:32:92:80 txqueuelen 1000 (Ethernet)
RX packets 181 bytes 23286 (22.7 KiB)
RX errors 0 dropped 4 overruns 0 frame 0
TX packets 61 bytes 5788 (5.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

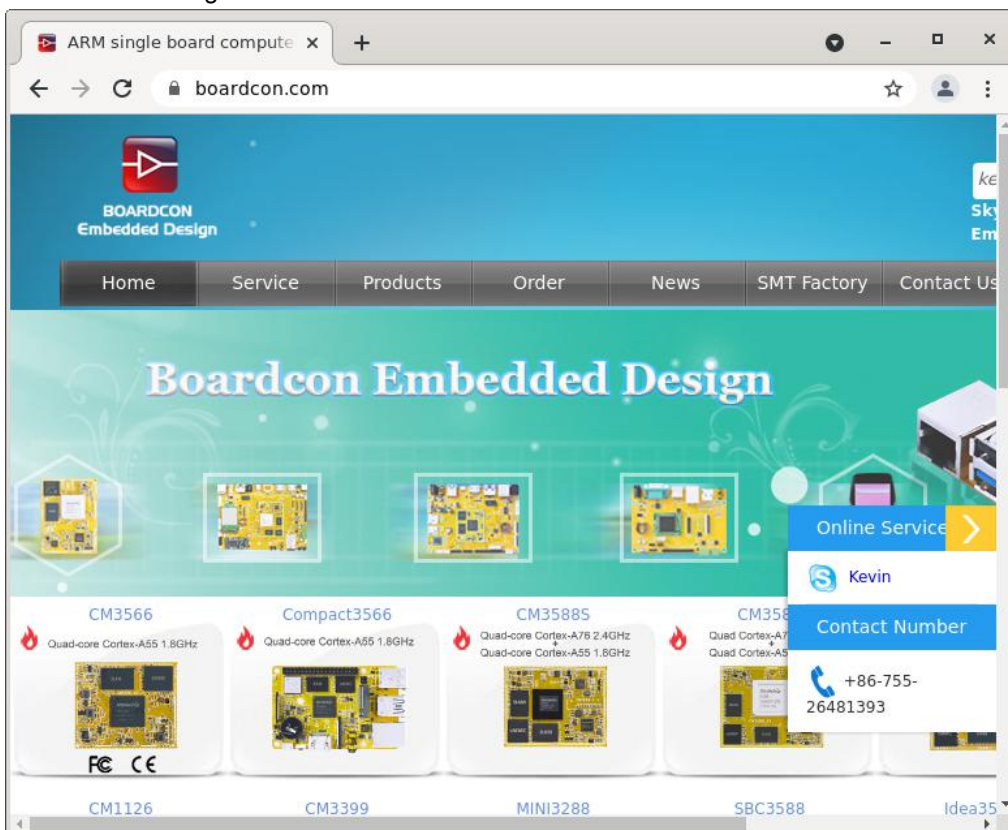
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@linaro-alip:/#
```

```
# ping -I eth0 www.boardcon.com
```

```
root@linaro-alip:/# ping -I eth0 www.boardcon.com
PING www.boardcon.com (67.222.54.196) from 192.168.0.195 eth0: 56(84) bytes of data.
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=1 ttl=47 time=189 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=2 ttl=47 time=242 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=3 ttl=47 time=254 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=4 ttl=47 time=279 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=5 ttl=47 time=268 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=6 ttl=47 time=198 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=7 ttl=47 time=195 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=8 ttl=47 time=198 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=9 ttl=47 time=188 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=10 ttl=47 time=190 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=11 ttl=47 time=196 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=12 ttl=47 time=202 ms
AC
--- www.boardcon.com ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11020ms
rtt min/avg/max/mdev = 188.379/216.729/279.434/32.509 ms
root@linaro-alip:/#
```

Browse the web through the browser.



## 6.7 RTC

Execute the follow command to set the RTC time:

```
# date -s "2023-12-25 12:00:00"
# hwclock -w
# hwclock
```

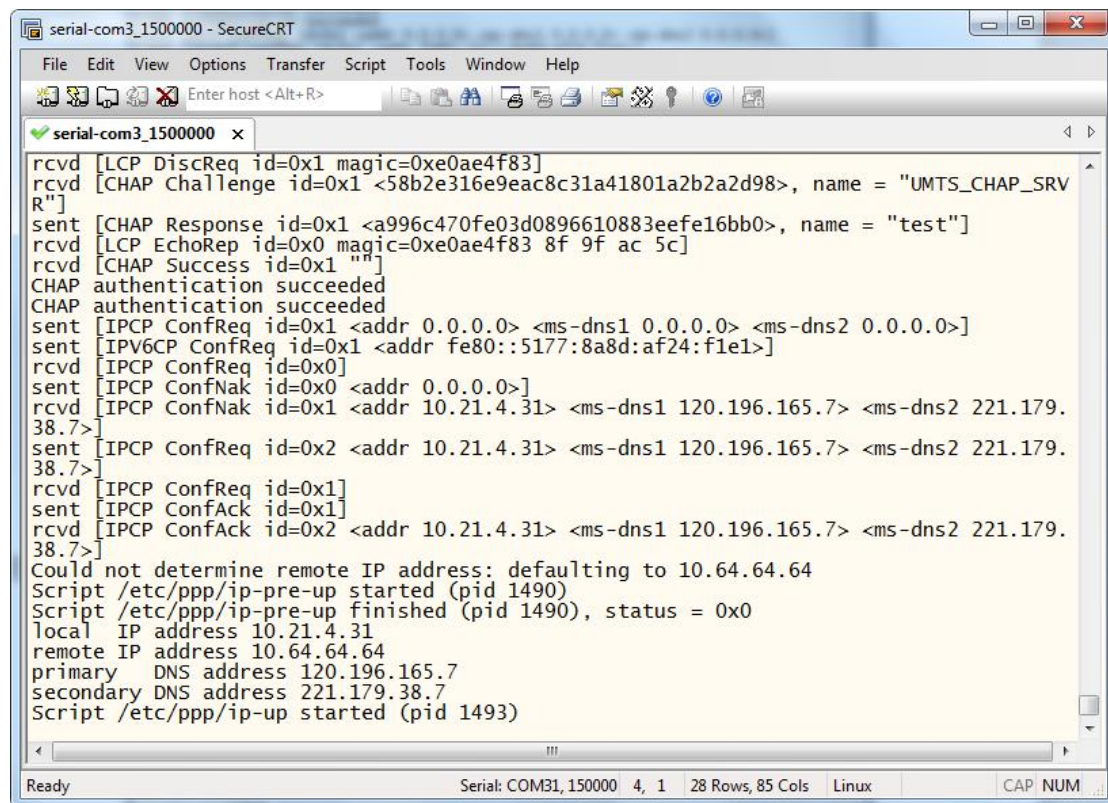
```
root@linaro-alip:/# date -s "2023-12-25 12:00:00"
Mon Dec 25 12:00:00 UTC 2023
root@linaro-alip:/# hwclock -w
root@linaro-alip:/# hwclock
2023-12-25 12:00:06.918571+00:00
root@linaro-alip:/# hwclock
2023-12-25 12:00:10.615157+00:00
root@linaro-alip:/# hwclock
2023-12-25 12:00:11.847907+00:00
root@linaro-alip:/# hwclock
2023-12-25 12:00:12.829042+00:00
root@linaro-alip:/# hwclock
2023-12-25 12:00:13.718299+00:00
root@linaro-alip:/# hwclock
2023-12-25 12:00:14.632908+00:00
root@linaro-alip:/#
```

After the board is turned off, the RTC supports battery power supply, and RTC clock to be saved.

## 6.8 4G(EC20)

*Step 1*, execute the following command to realize ppp dialing:

```
# pppd call quectel-ppp &
```



```
serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
rcvd [LCP DiscReq id=0x1 magic=0xe0ae4f83]
rcvd [CHAP Challenge id=0x1 <58b2e316e9eac8c31a41801a2b2a2d98>, name = "UMTS_CHAP_SRV
R"]
sent [CHAP Response id=0x1 <a996c470fe03d0896610883eefe16bb0>, name = "test"]
rcvd [LCP EchoRep id=0x0 magic=0xe0ae4f83 8f 9f ac 5c]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
sent [IPV6CP ConfReq id=0x1 <addr fe80::5177:8a8d:af24:f1e1>]
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfNak id=0x1 <addr 10.21.4.31> <ms-dns1 120.196.165.7> <ms-dns2 221.179.
38.7>]
sent [IPCP ConfReq id=0x2 <addr 10.21.4.31> <ms-dns1 120.196.165.7> <ms-dns2 221.179.
38.7>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfAck id=0x2 <addr 10.21.4.31> <ms-dns1 120.196.165.7> <ms-dns2 221.179.
38.7>]
Could not determine remote IP address: defaulting to 10.64.64.64
Script /etc/ppp/ip-pre-up started (pid 1490)
Script /etc/ppp/ip-pre-up finished (pid 1490), status = 0x0
local IP address 10.21.4.31
remote IP address 10.64.64.64
primary DNS address 120.196.165.7
secondary DNS address 221.179.38.7
Script /etc/ppp/ip-up started (pid 1493)
Ready Serial: COM31,150000 4, 1 28 Rows, 85 Cols Linux CAP NUM
```

Step 2, execute the following command to view the network interface status:

```
# ifconfig
```

```
root@linaro-alip:/# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 16:1f:bf:a6:4f:f5 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.106.168.138 netmask 255.255.255.255 destination 10.64.64.64
    ppp txqueuelen 3 (Point-to-Point Protocol)
    RX packets 4 bytes 52 (52.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 156 (156.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@linaro-alip:/# sent [IPV6CP ConfReq id=0x1 <addr fe80::e126:737d:2578:a510>]
sent [IPV6CP ConfReq id=0x1 <addr fe80::e126:737d:2578:a510>]
```

Step 3, execute the following to check the connectivity of the detection network:

```
# ping -I ppp0 www.boardcon.com
```

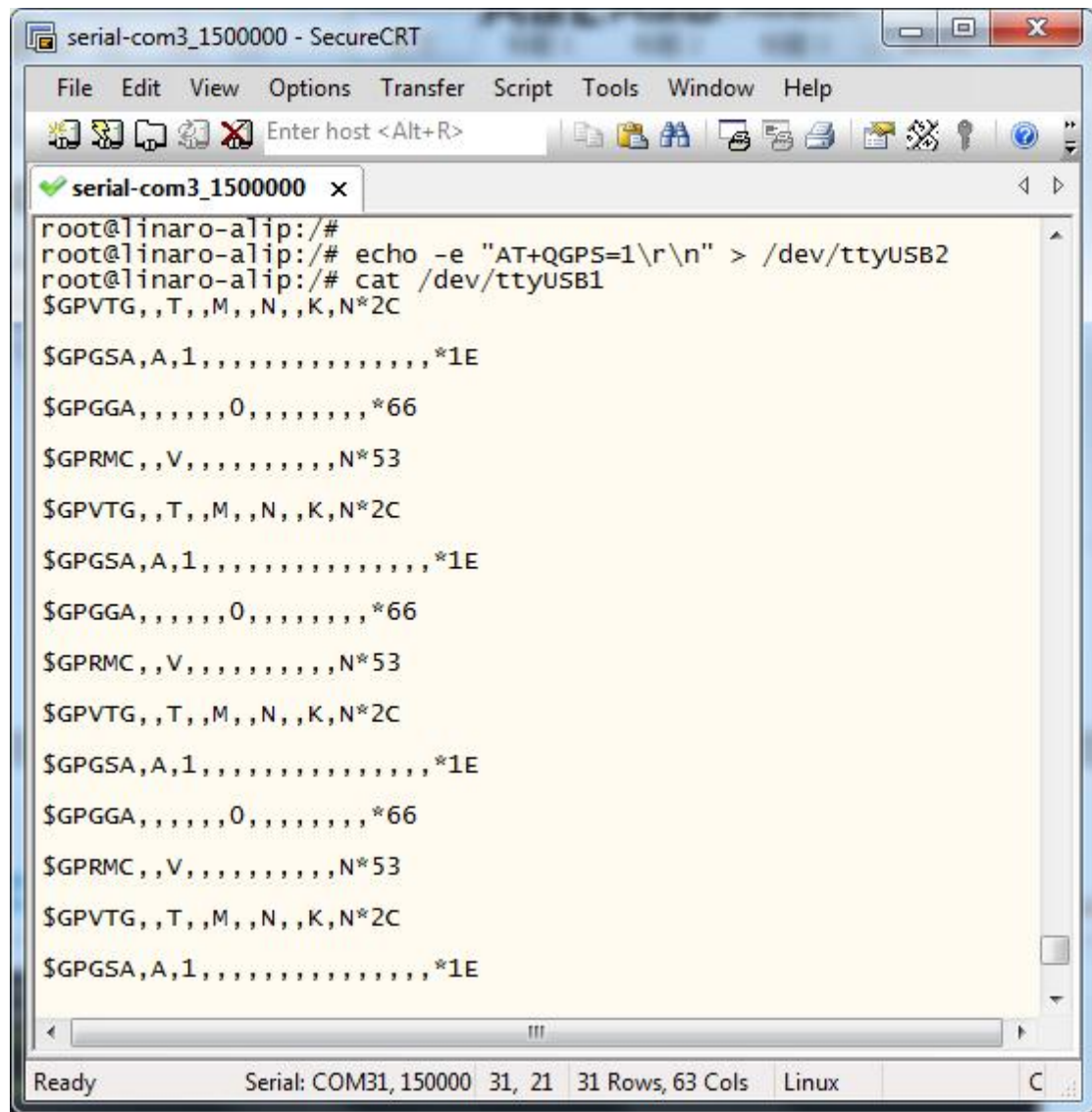
```
root@linaro-alip:/# ping -I ppp0 www.boardcon.com
PING www.boardcon.com (67.222.54.196) from 10.106.168.138 ppp0: 56(84) bytes of data.
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=2 ttl=45 time=264 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=3 ttl=45 time=274 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=4 ttl=45 time=263 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=6 ttl=45 time=253 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=7 ttl=45 time=252 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=10 ttl=45 time=259 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=11 ttl=45 time=259 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=13 ttl=45 time=261 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=15 ttl=45 time=264 ms
^C
--- www.boardcon.com ping statistics ---
16 packets transmitted, 9 received, 43.75% packet loss, time 15152ms
rtt min/avg/max/mdev = 252.040/260.951/273.580/6.049 ms
root@linaro-alip:/#
```

## 6.9 GPS(EC20)

Plug the EC20 module and connect GPS antenna, then power on and execute the follow command:

```
# echo -e "AT+QGPS=1\r\n" > /dev/ttyUSB2
```

```
# cat /dev/ttyUSB1
```



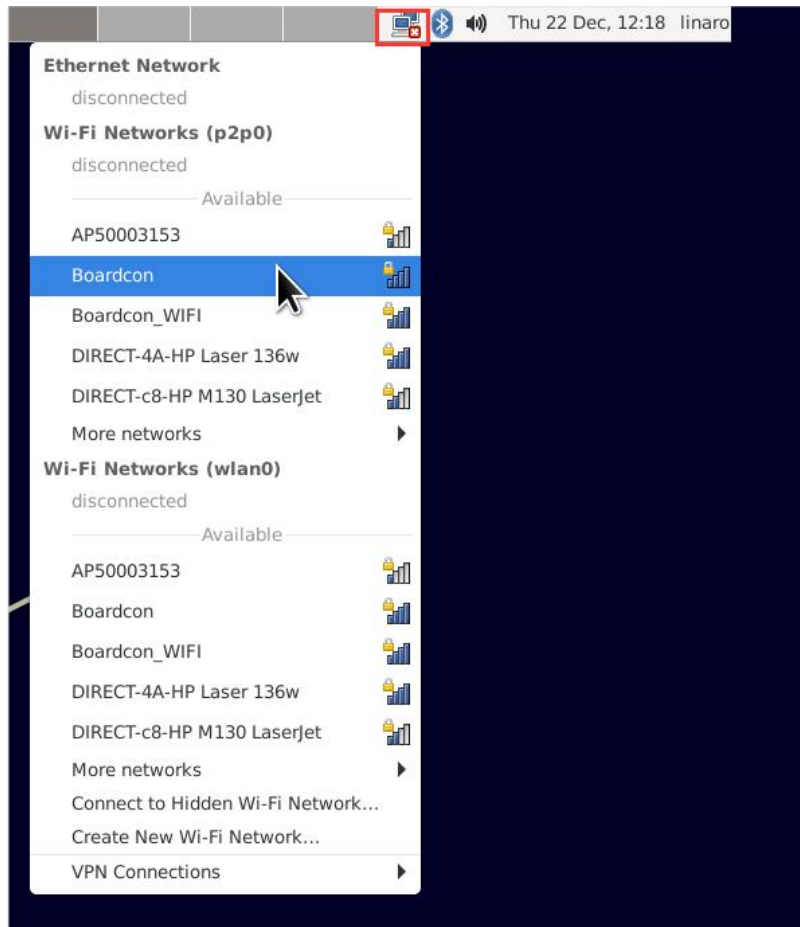
```
serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
root@linaro-alip:/#
root@linaro-alip:/# echo -e "AT+QGPS=1\r\n" > /dev/ttyUSB2
root@linaro-alip:/# cat /dev/ttyUSB1
$GPVTG,,T,,M,,N,,K,N*2C

$GPGSA,A,1,,,,,,,,,,,,*1E
$GPGGA,,,,,0,,,,,,,,*66
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,T,,M,,N,,K,N*2C
$GPGSA,A,1,,,,,,,,,,,,*1E
$GPGGA,,,,,0,,,,,,,,*66
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,T,,M,,N,,K,N*2C
$GPGSA,A,1,,,,,,,,,,,,*1E
$GPGGA,,,,,0,,,,,,,,*66
$GPRMC,,V,,,,,,,,,N*53
$GPVTG,,T,,M,,N,,K,N*2C
$GPGSA,A,1,,,,,,,,,,,,*1E

Ready Serial: COM31, 150000 31, 21 31 Rows, 63 Cols Linux C
```

## 6.10 WiFi

*Step 1*, connect the WiFi antenna, then click the network icon in the top right corner of the UI interface, select the SSID from the list of available networks and enter the password.



**Step 2**, execute the following to view the network interface status:

```
# ifconfig
```

```
p2p0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.187 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::d1e:6789:bc14:f830 prefixlen 64 scopeid 0x20<link>
ether c6:3c:b0:43:03:4a txqueuelen 1000 (Ethernet)
RX packets 91 bytes 19584 (19.1 KiB)
RX errors 0 dropped 11 overruns 0 frame 0
TX packets 14 bytes 2110 (2.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

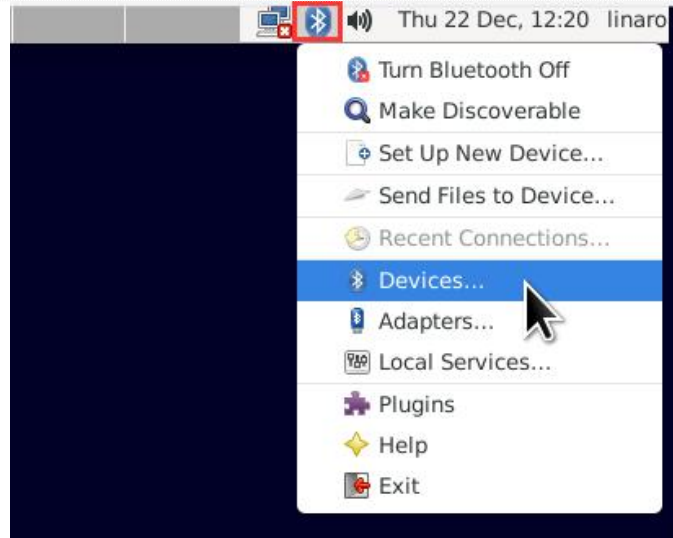
**Step 3**, execute the following to check the connectivity of the detection network:

```
# ping -I p2p0 www.boardcon.com
```

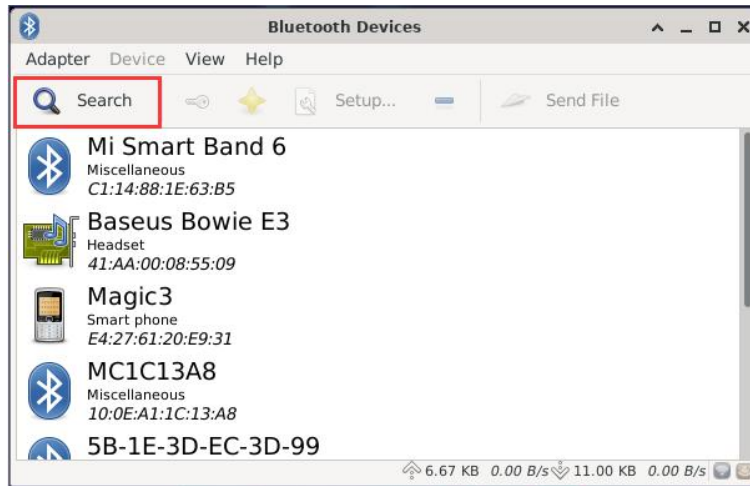
```
root@linaro-alip:/# ping -I p2p0 www.boardcon.com
PING www.boardcon.com (67.222.54.196) from 192.168.0.187 p2p0: 56(84) bytes of data.
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=1 ttl=47 time=204 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=2 ttl=47 time=195 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=3 ttl=47 time=248 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=4 ttl=47 time=192 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=5 ttl=47 time=205 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=6 ttl=47 time=194 ms
^C
--- www.boardcon.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5002ms
rtt min/avg/max/mdev = 192.172/206.514/248.351/19.390 ms
```

## 6.11 Bluetooth

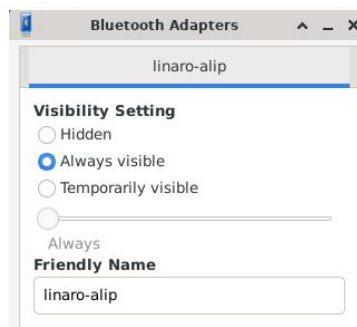
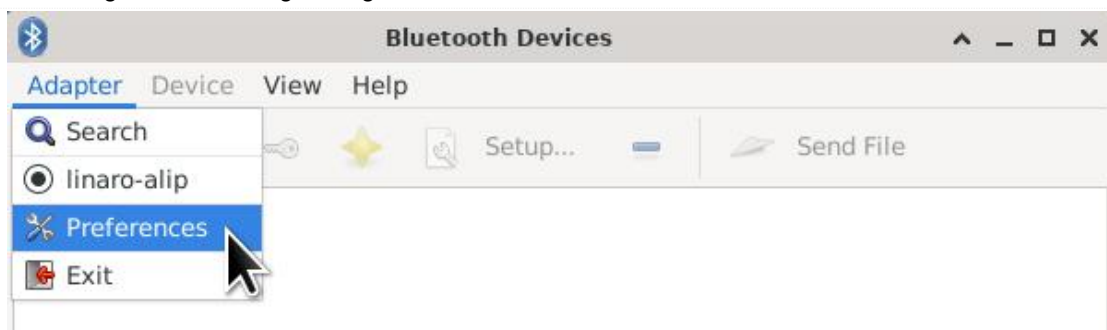
**Step 1**, click the Bluetooth icon in the top right corner of the UI interface, Select **“Devices”** enter the Bluetooth Devices.



Step 3, click the “Search” button to start searching and select the available device in the list to pair.

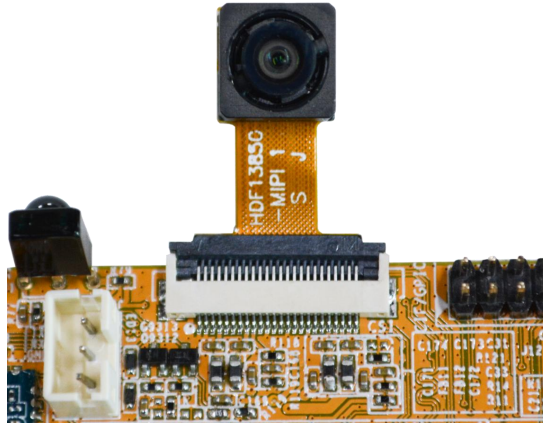


Step 4, the Bluetooth device name is hidden by default. Set the Bluetooth device name to be visible according to the following settings:



## 6.12 Camera

The EM3566S debain system the camera module used is OV13850. If you need to preview camera, please connect camera to CIF interface.



**Step 1**, execute the following command to view the device channel:

```
# grep "" /sys/class/video4linux/v*/name | grep mainpath
```

```
root@linaro-alip:/# grep "" /sys/class/video4linux/v*/name | grep mainpath
/sys/class/video4linux/video0/name:rkisp1_mainpath
root@linaro-alip:/#
```

```
# grep "" /sys/class/video4linux/v*/name | grep selfpath
```

```
root@linaro-alip:/# grep "" /sys/class/video4linux/v*/name | grep selfpath
/sys/class/video4linux/video1/name:rkisp1_selfpath
root@linaro-alip:/#
```

**Step 2**, execute the following command to preview camera:

```
# gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=NV12,width=1920,height=1080,framerate=30/1 ! xvimagesink
```

```
# gst-launch-1.0 v4l2src device=/dev/video1 ! video/x-raw,format=NV12,width=1920,height=1080,framerate=30/1 ! xvimagesink
```

Or

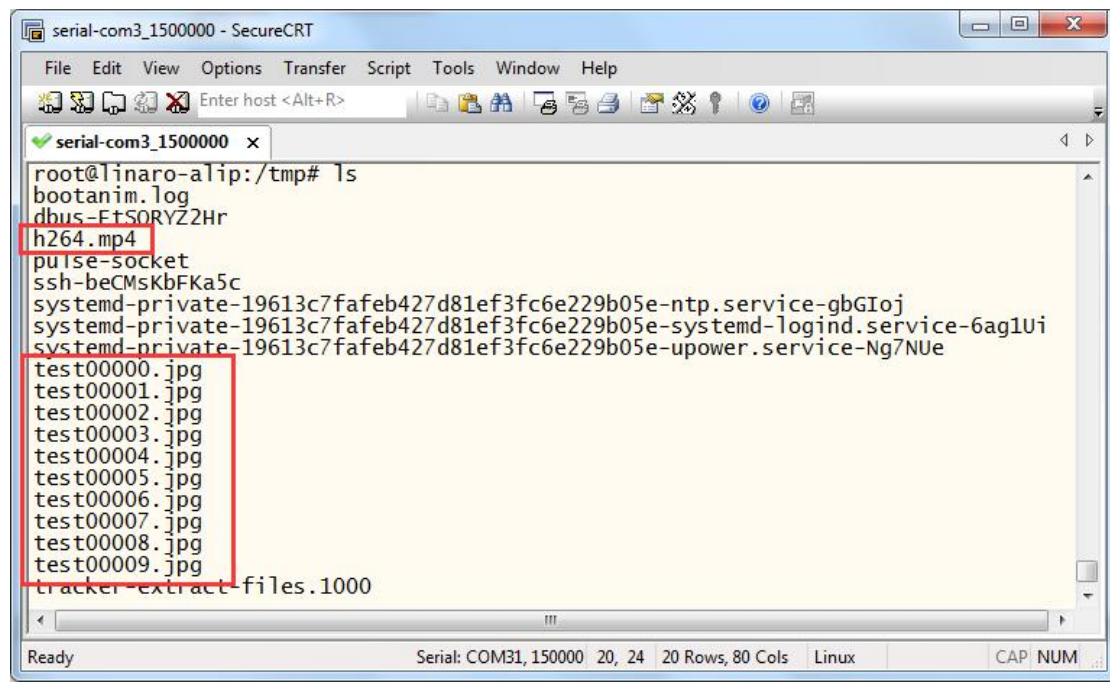
```
# /rockchip-test/camera/camera_rkaiq_test.sh
```

**Step 3**, execute the following command to record the video:

```
# gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=100 ! \
video/x-raw,format=NV12,width=1920,height=1088,framerate=30/1 ! \
videoconvert ! mpph264enc ! h264parse ! mp4mux ! \
filesink location=/tmp/h264.mp4
```

**Step 4**, execute the following command to take photos:

```
# gst-launch-1.0 -v v4l2src device=/dev/video0 num-buffers=10 ! \
video/x-raw,format=NV12,width=1280,height=800 ! mppjpegenc ! \
multifilesink location=/tmp/test%05d.jpg
```



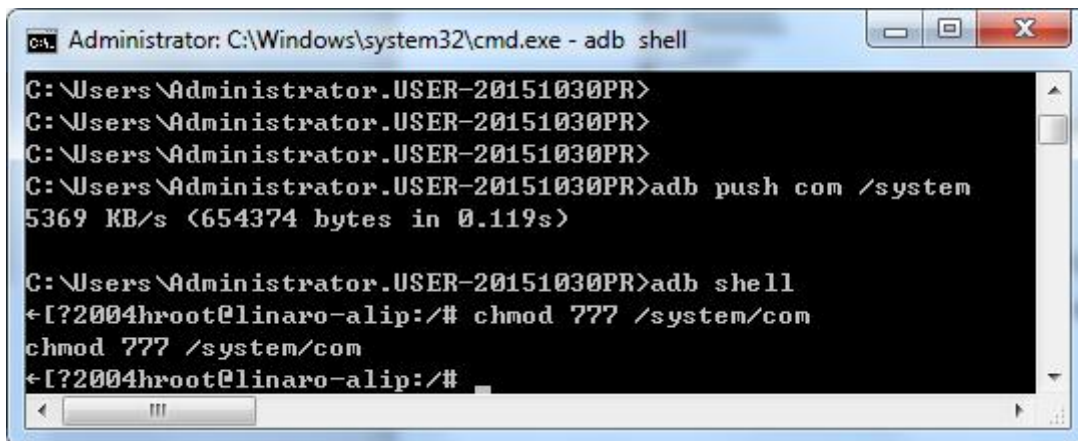
## 6.13 RS485

**Step 1**, connect the RS485 ports of board A and board B as follows:



**Step 2**, execute the following commands to push com to the board by ADB.

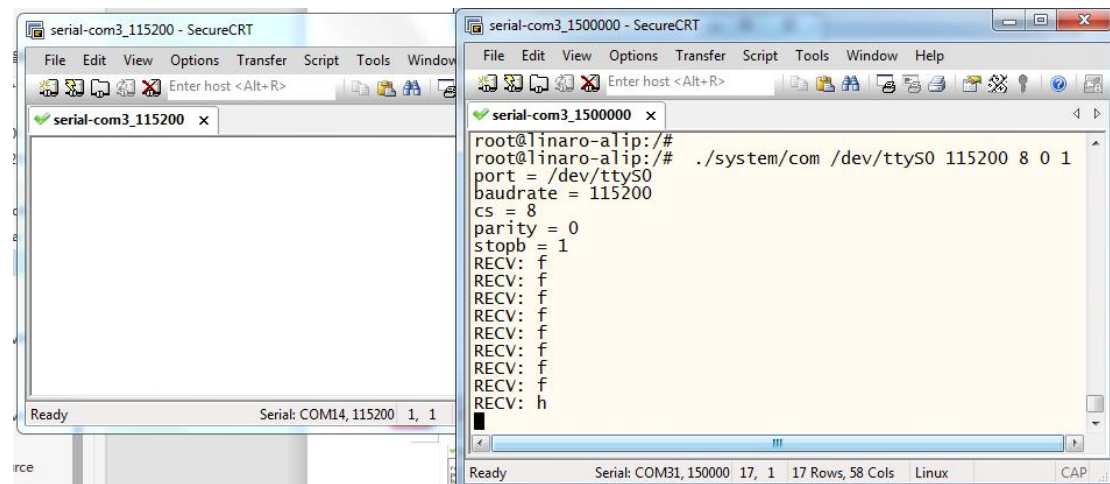
```
# adb push com /system
# adb shell
# chmod 777 /system/com
```



**Step 3**, execute the following commands on the terminal serial ports of board A and board B respectively.

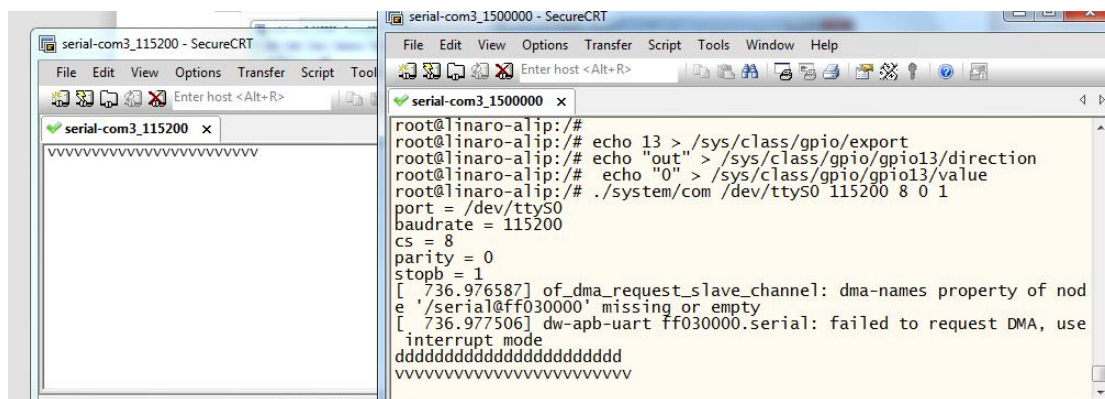
```
# ./system/com /dev/ttyS0 115200 8 0 1
```

**Step 4**, at this time, can test the RS485 communication, The RS485 is used as the receiver by default.



**Step 5**, Execute following command to switch RS485 as the sender:

```
# echo 13 > /sys/class/gpio/export
# echo "out" > /sys/class/gpio/gpio13/direction
# echo "0" > /sys/class/gpio/gpio13/value
# ./system/com /dev/ttyS0 115200 8 0 1
```



Execute following command to switch RS485 as the receiver again :

```
# echo "1" > /sys/class/gpio/gpio13/value
```

The method for testing RS485 is not unique, and the above method is only for reference.

## 6.14 SPI

**Step 1**, execute the following commands to push spidev\_test0.0 and spidev\_test1.0 to the board by ADB.

```
# adb push spidev_test0.0 /system
# adb push spidev_test1.0 /system
# adb shell
# chmod 777 /system/spidev_test*
```

```

Administrator: C:\Windows\system32\cmd.exe - adb shell
C:\Users\Administrator.USER-20151030PR>adb push spidev_test0.0 /system
1087 KB/s <20040 bytes in 0.018s>

C:\Users\Administrator.USER-20151030PR>adb push spidev_test1.0 /system
978 KB/s <20040 bytes in 0.020s>

C:\Users\Administrator.USER-20151030PR>adb shell
<[?2004hroot@linaro-alip:/# chmod 777 /system/spi *
chmod 777 /system/spidev_test*
<[?2004hroot@linaro-alip:/# chmod 777 /system/spidev_test*
chmod 777 /system/spidev_test*
<[?2004hroot@linaro-alip:/#
  
```

**Step 2**, SPI0 test: after the terminal serial port executes the following instructions, the test can be performed.

```
# ./system/spidev_test0.0
```

```

serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
root@linaro-alip:/# ./system/spidev_test0.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF
root@linaro-alip:/#
  
```

Short circuit SPI0\_MISO and SPI0\_MOSI pins of SPI0(CON6 pin3&pin4). Execute follow command again.

```
# ./system/spidev_test0.0
```

```

serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
root@linaro-alip:/# ./system/spidev_test0.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
40 00 00 00 00 95
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
DE AD BE EF BA AD
F0 0D
root@linaro-alip:/#
Ready Serial: COM31, 150000 13, 21 13 Rows, 49 Cols Linux

```

Step 3, SPI1 test: after the terminal serial port executes the following instructions, the test can be performed.

# ./system/spidev\_test1.0

```

serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
root@linaro-alip:/# ./system/spidev_test1.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF
FF FF
root@linaro-alip:/# █
Ready Serial: COM31, 150000 13, 21 13 Rows, 49 Cols Linux

```

Short circuit SPI1\_MISO and SPI1\_MOSI pins of SPI1(CON7 pin3&pin4). Execute follow command again.

# ./system/spidev\_test1.0

```

serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
root@linaro-alip:/# ./system/spidev_test1.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
40 00 00 00 00 95
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
DE AD BE EF BA AD
F0 0D
root@linaro-alip:/# █
Ready Serial: COM31, 150000 13, 21 13 Rows, 49 Cols Linux

```

## 6.15 IR

Connect IR receiver to the IR port, execute the following command to turn on IR receiving data information printing:

```
# echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

```

serial-com3_1500000 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com3_1500000 x
root@linaro-alip:/#
root@linaro-alip:/#
root@linaro-alip:/# echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
root@linaro-alip:/# [ 2612.096450] USERCODE=0x0
[ 2654.874270] USERCODE=0xfe01
[ 2654.901309] RMC_GETDATA=b8
[ 2655.140532] USERCODE=0xfe01
[ 2655.167599] RMC_GETDATA=b8
[ 2655.347653] USERCODE=0xfe01
[ 2655.374709] RMC_GETDATA=b8
[ 2655.537997] USERCODE=0xfe01
[ 2655.565053] RMC_GETDATA=b8
█
Ready Serial: COM31, 150000 13, 1 13 Rows, 85 Cols Linux CAP_NUM

```

# Tool Usage

## 1.Driver

### 1.1 USB driver

Rockchip USB driver install package includes ADB and image flashing driver.

RKTools\windows\DriverAssitant\_v5.1.1.zip

### 1.2 Debug serial port driver

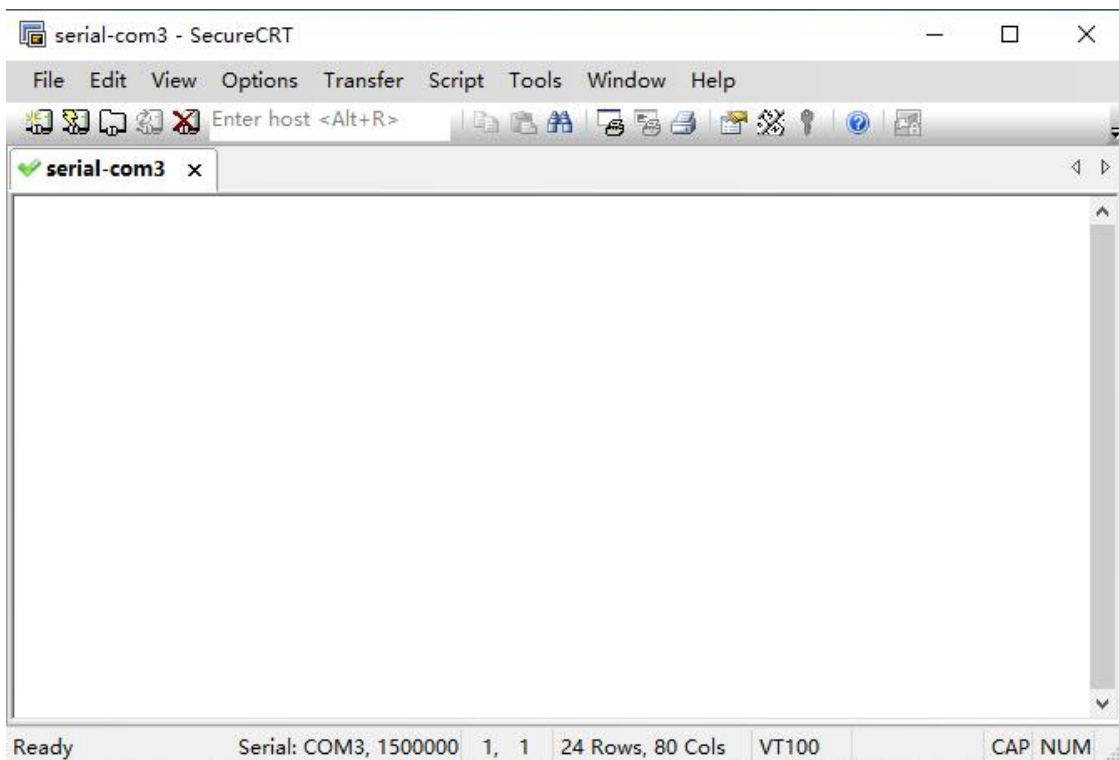
Boardcon will provide debugging serial port drivers and tools. If you need to debug on the serial port, you need to install this driver.

CP210x\_Windows\_Drivers\_with\_Serial\_Enumeration.zip

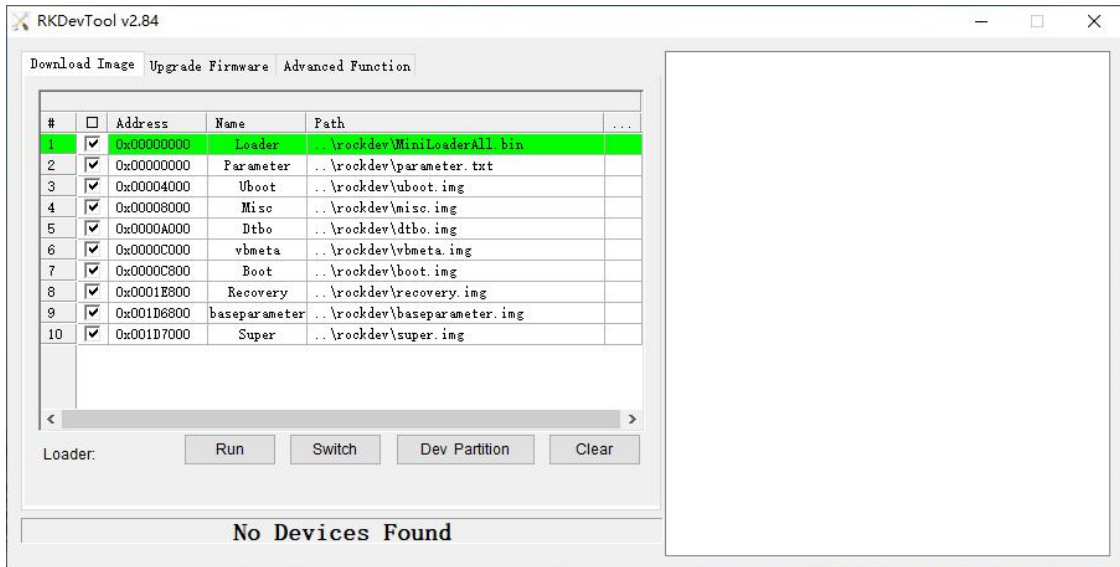
## 2.Debug Tool

Boardson recommends using SecureCRT. If you have a better choice, you can use other tools instead.

Platform/SecureCRT.rar



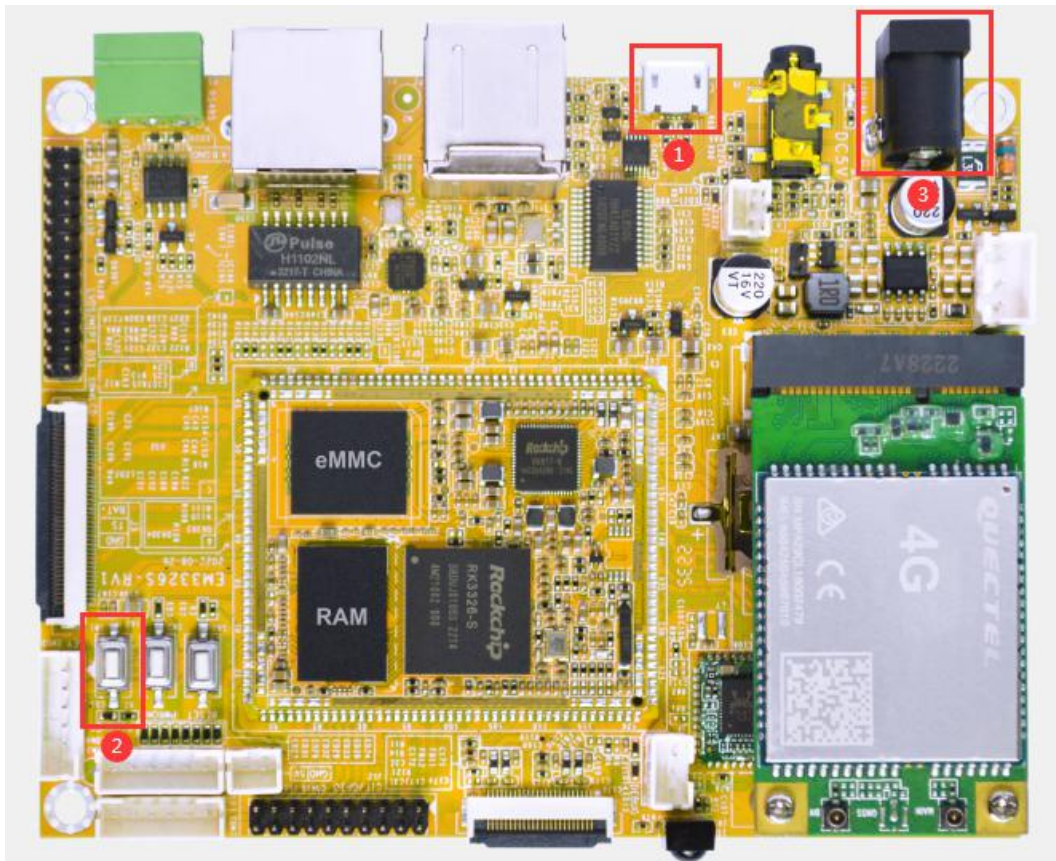
# 3. Development Flashing Tool



## 3.1 Loader mode

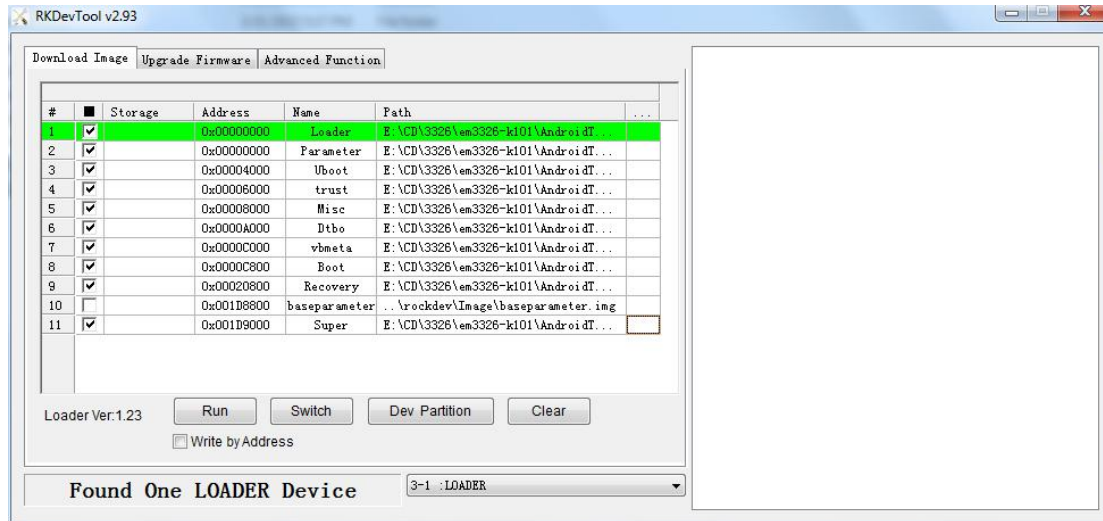
The normal burning mode is the loader mode. You can enter the loader mode through the following methods:

- (1) Connect PC and development board with OTG USB cable, keep pressing the **Recover Key** and power the board until the windows PC shows **Found one LOADER Device**.



(2) When USB OTG cable and serial port cable have been connected, enter the following command in the debugging serial port tool to enter the loader mode:

\$: reboot loader

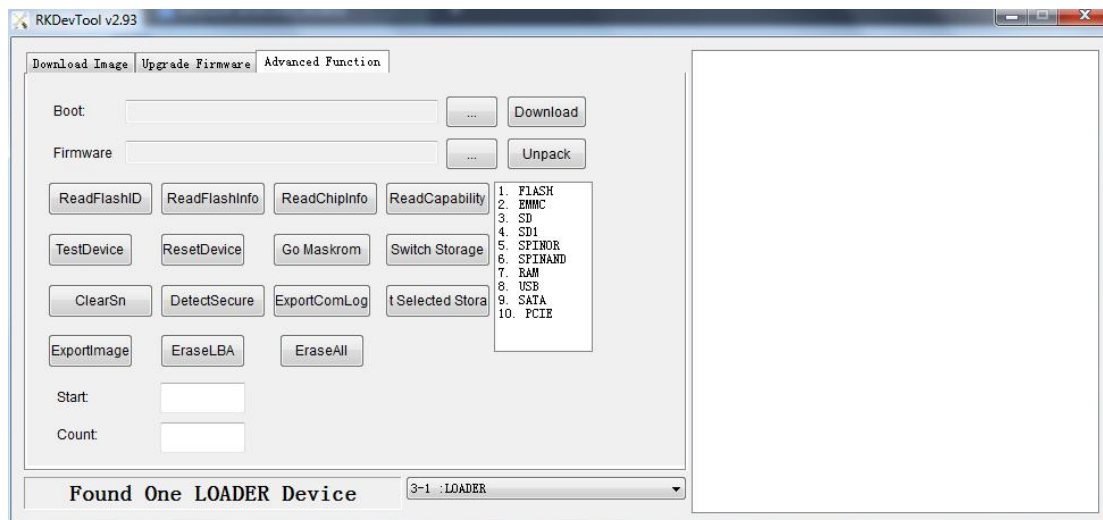


### 3.2 Maskrom mode

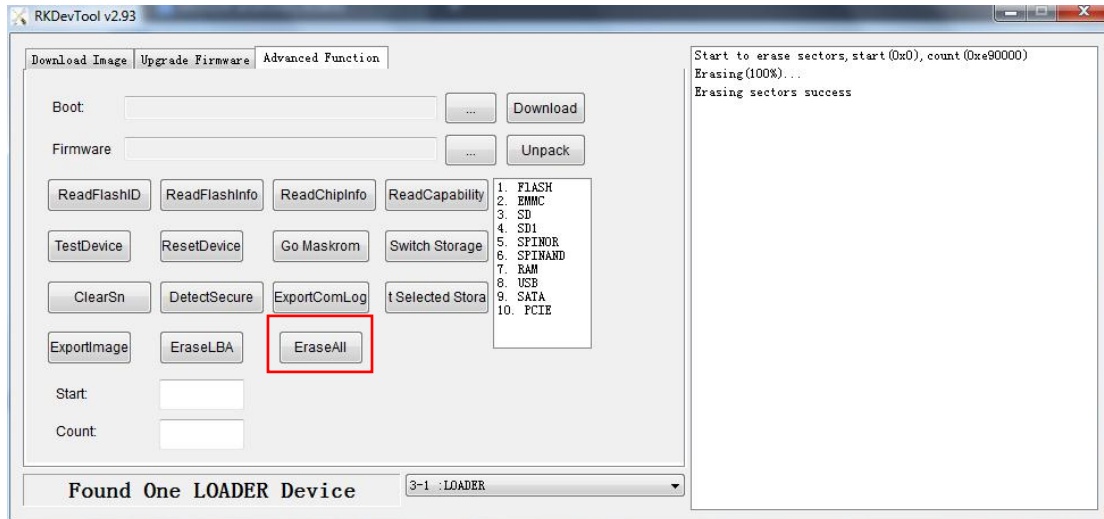
MaskRom mode is mainly used to repair the system when bootloader is damaged. Generally, you do not need to enter MaskRom mode. Only when bootloader verification fails (cannot read IDR block or bootloader is damaged), will choose to use markrom mode to burn firmware. Here are two ways to enter the maskrom mode:

(1) In the Loader mode, use the firmware to erase the emmc before entering the maskrom mode.

**Step 1**, enter loader mode.



**Step 2**, click **Advanced Function** . Click **EraseAll** button to erase flash. Then repower enter the maskrom mode.

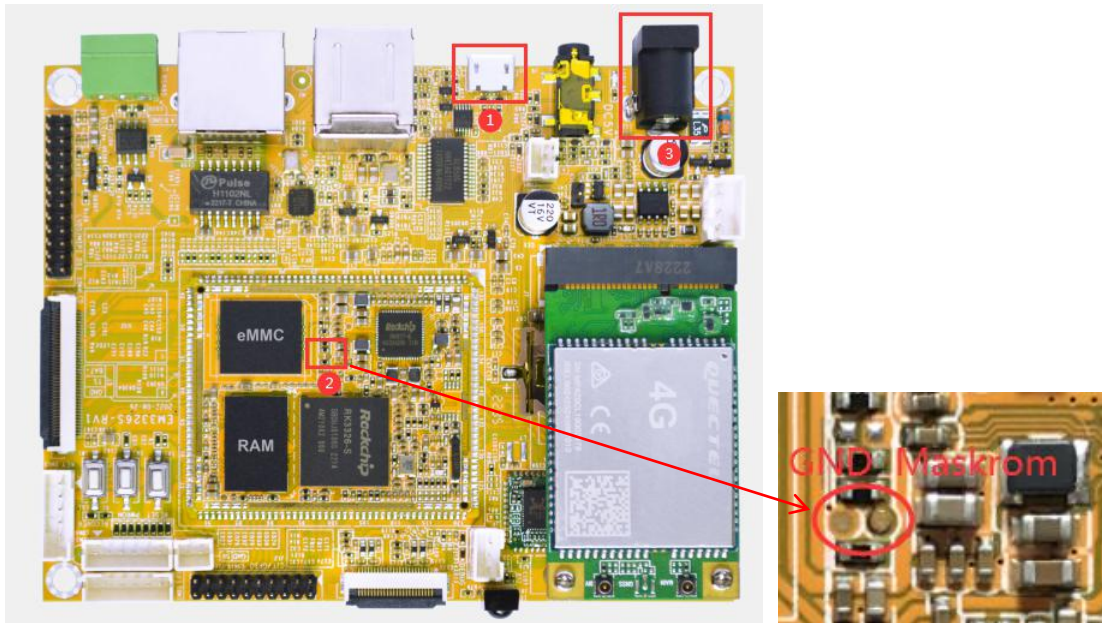


(2) Two solder joints are reserved in the hardware design. Just short the two solder joints and then power to enter the maskrom mode.

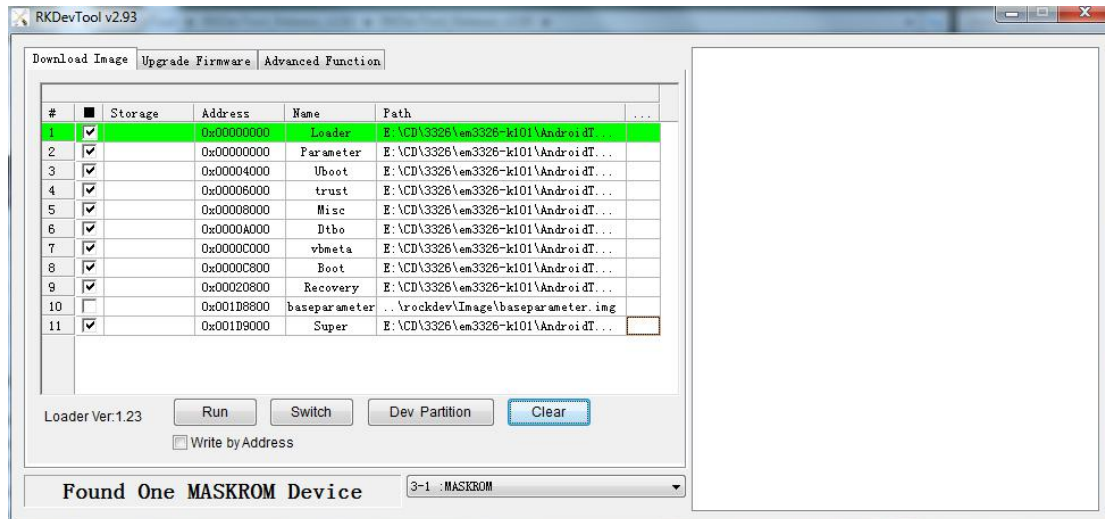
**Step 1**, connect PC and development board with OTG USB cable.

**Step 2**, short circuit the two test points near eMMC to enter MASKROM mode.

**Step 3**, connect power cables.

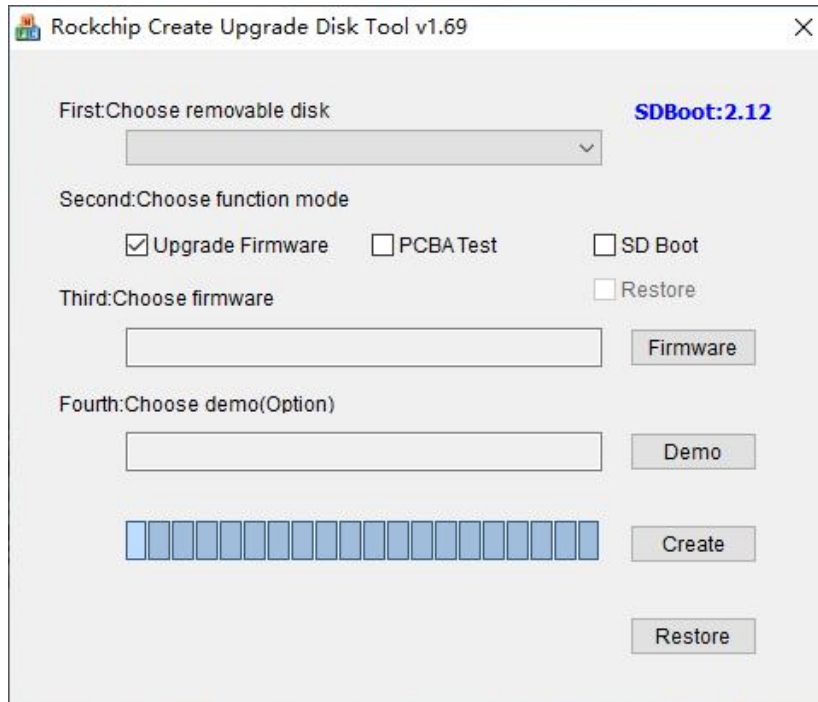


Step 4, after entering the maskrom mode, cancel the short circuit.



## 4.Tool to Implement SD Upgrading and Boot

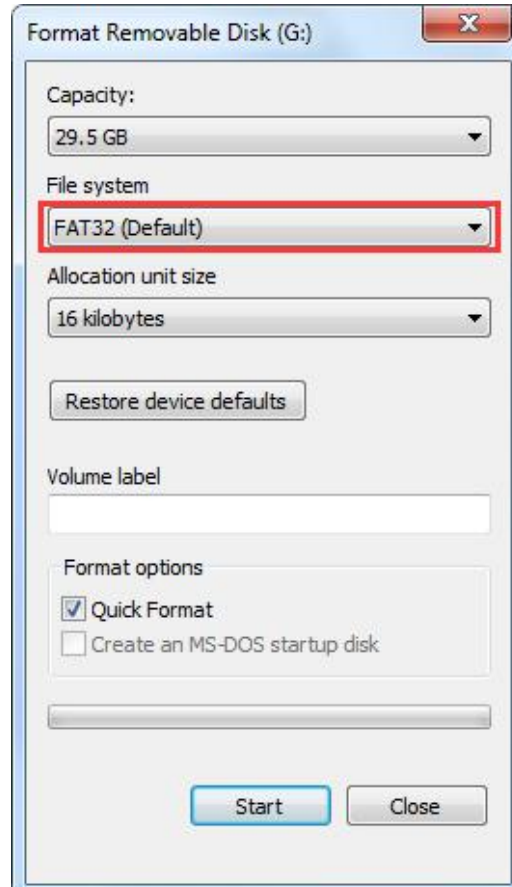
RKTools\windows\SDDiskTool\_v1.69.zip



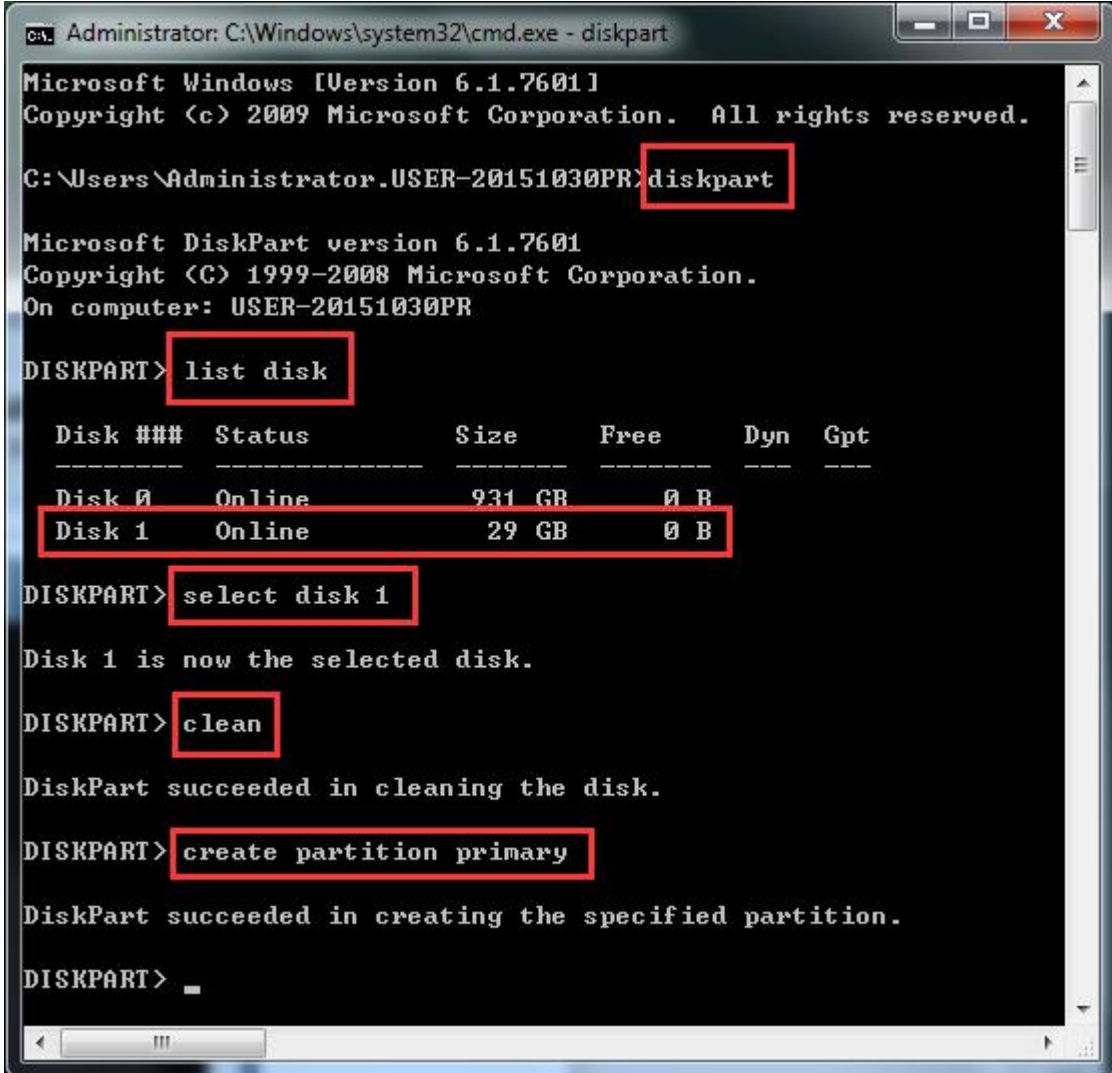
## 4.1 How to format sdcard

The SD card must be formatted before make SD upgrade card, and make sure that the SD card has only one partition, otherwise it may cause the board fail to boot.

- (1) If your sdcard has only one partition, just format it as FAT32 in WINDOWS.



(2) If your sdcard has multiple partitions, need to delete the partitions in CMD of WINDOWS as follow:



```
Administrator: C:\Windows\system32\cmd.exe - diskpart
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator.USER-20151030PR>diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: USER-20151030PR

DISKPART> list disk

   Disk ###  Status         Size           Free           Dyn  Gpt
   -----  -
   Disk 0    Online         931 GB         0 B
   Disk 1    Online         29 GB         0 B

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> clean

DiskPart succeeded in cleaning the disk.

DISKPART> create partition primary

DiskPart succeeded in creating the specified partition.

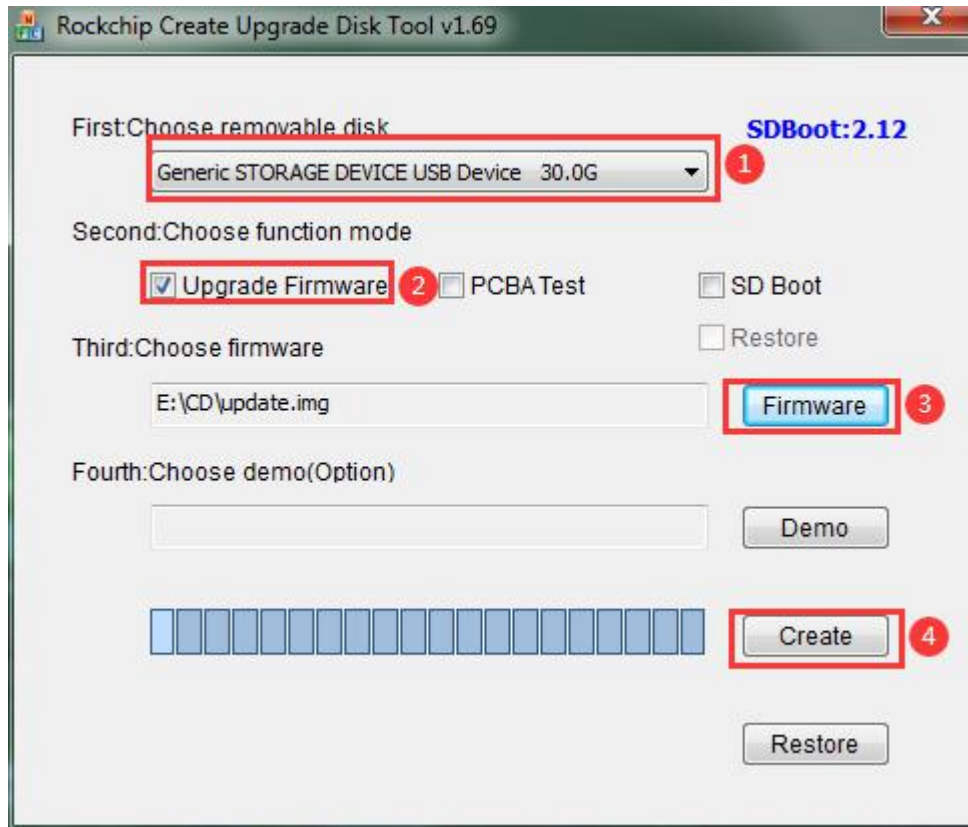
DISKPART> _
```

And then format it as FAT32.

## 4.2 How to make SD upgrade card

Step 1, open SD\_Firmware\_Tool.exe (SDDiskTool\_v1.69\SD\_Firmware\_Tool.exe).

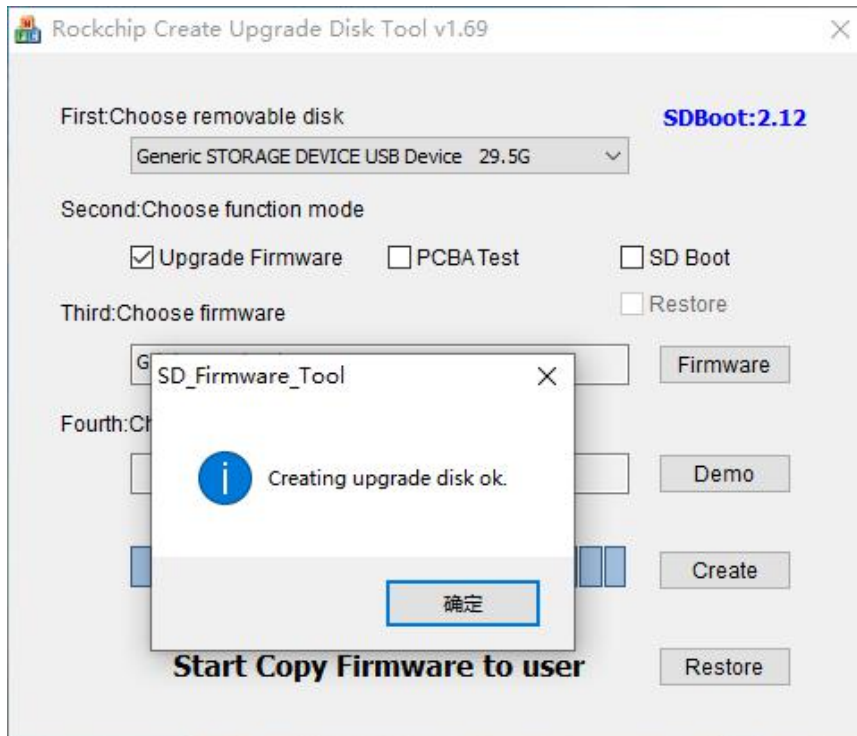
Step 2, as shown in the figure, make SD upgrade card.



Parameter Description:

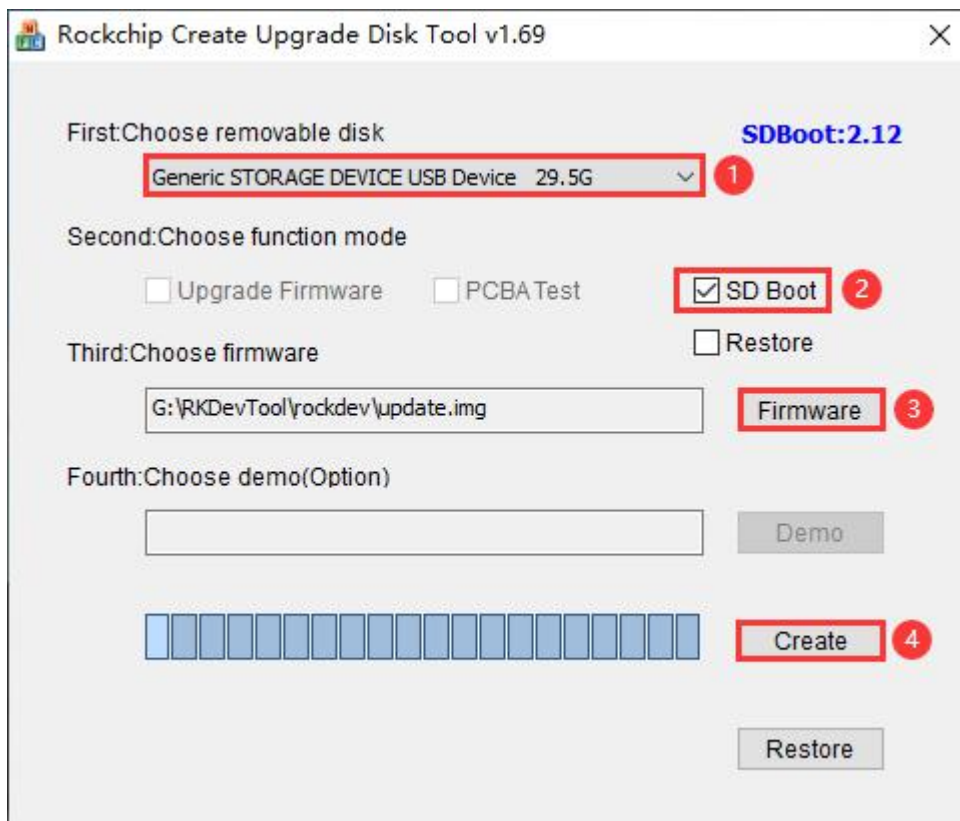
Step	Describe
1	Select an SD card or USB flash drive from the <b>drop-down list</b> .
2	Select " <b>Upgrade Firmware</b> ".
3	Click Select <b>Firmware</b> button and select <b>update.img</b> to upgrade the firmware.
4	Click <b>Create</b> button to start creation.

*Step 3*, wait for make SD upgrade card to finish.



### 4.3 How to make SD boot card

The card printing mode of SD card boot is similar to that of SD card upgrade, but the only difference is that the Choose function mode is different, as shown below:



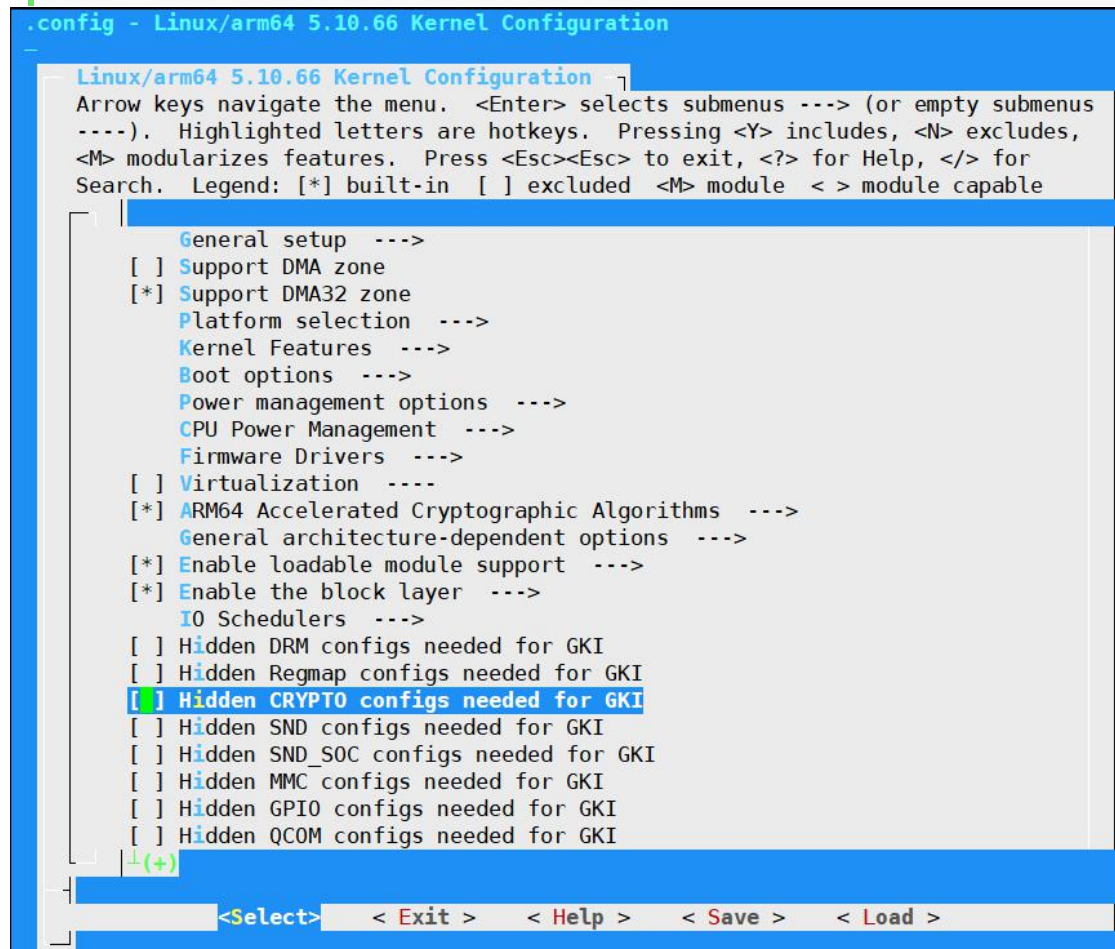
The SD card boot firmware is incompatible with the firmware burned to flash. You need to modify the SDK related configuration. If need to do so, please contact boardcon.

# Common Issues

## 1. How do open the kernel configuration file

Use the following command to open the kernel configuration:

```
$: cd kernel
$: make ARCH=arm64 menuconfig
```



```
.config - Linux/arm64 5.10.66 Kernel Configuration
Linux/arm64 5.10.66 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus
----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

General setup --->
[ ] Support DMA zone
[*] Support DMA32 zone
Platform selection --->
Kernel Features --->
Boot options --->
Power management options --->
CPU Power Management --->
Firmware Drivers --->
[ ] Virtualization ----
[*] ARM64 Accelerated Cryptographic Algorithms --->
General architecture-dependent options --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
IO Schedulers --->
[ ] Hidden DRM configs needed for GKI
[ ] Hidden Regmap configs needed for GKI
[+] Hidden CRYPTO configs needed for GKI
[ ] Hidden SND configs needed for GKI
[ ] Hidden SND_SOC configs needed for GKI
[ ] Hidden MMC configs needed for GKI
[ ] Hidden GPIO configs needed for GKI
[ ] Hidden QCOM configs needed for GKI

<Select> < Exit > < Help > < Save > < Load >
```

If the save directory is not modified, the configuration content is saved in the `kernel/.config` file by default, you need to manually update the latest configuration content to `kernel/arch/arm64/configs/rockchip_defconfig` file, and then compile the kernel again.

**Note:**

- (1) Some SDK have multiple kernel directories (`kernel`, `kernel-4.19`, `kernel-5.10`). You can know which kernel directory is used according to kernel compilation instructions.
- (2) Different systems used by default `rockchip_defconfig` configuration file may not be the same, according to the kernel compilation directives can know the specific use which configuration files in the `kernel/arch/arm64/configs/` directory.

## 2.Error about lz4

If the following error occurs during compilation:

```
yy@yy-virtual-machine:/home/idea3588/buildroot$ ./build.sh kernel
processing option: kernel
=====Start building kernel=====
TARGET_ARCH      =arm64
TARGET_KERNEL_CONFIG =rockchip_linux_defconfig
TARGET_KERNEL_DTS  =rk3588-evb2-lp4-v10-edp-linux
TARGET_KERNEL_CONFIG_FRAGMENT =
=====
#
# No change to .config
#
CALL    scripts/atomic/check-atomics.sh
CALL    scripts/checksyscalls.sh
CHK     include/generated/compile.h
fatal: your current branch 'master' does not have any commits yet
LZ4C    arch/arm64/boot/Image.lz4
Incorrect parameters
Usage :
    lz4 [arg] [input] [output]

input  : a filename
        with no FILE, or when FILE is - or stdin, read standard input
Arguments :
  -1    : Fast compression (default)
  -9    : High compression
  -d    : decompression (default for .lz4 extension)
  -z    : force compression
  -f    : overwrite output without prompting
  -h/-H : display help/long help and exit
arch/arm64/boot/Makefile:31: recipe for target 'arch/arm64/boot/Image.lz4' failed
make[2]: *** [arch/arm64/boot/Image.lz4] Error 1
make[2]: *** Deleting file 'arch/arm64/boot/Image.lz4'
arch/arm64/Makefile:170: recipe for target 'Image.lz4' failed
make[1]: *** [Image.lz4] Error 2
make[1]: *** Waiting for unfinished jobs....
arch/arm64/Makefile:214: recipe for target 'rk3588-evb2-lp4-v10-edp-linux.img' failed
make: *** [rk3588-evb2-lp4-v10-edp-linux.img] Error 2
ERROR: Running build_kernel failed!
ERROR: exit code 2 from line 656:
make ARCH=$RK_ARCH $RK_KERNEL_DTS.img -j$RK_JOBS
```

The built-in lz4 version is too low, requiring 1.8.3 or later versions, view the version number of the current lz4:

```
$ lz4 -v
```

### Solution:

Method (1) : Directly copy Android compiled lz4 overwrite system lz4:

```
# sudo cp /out/host/linux-x86/bin/lz4 /usr/bin/lz4
```

Method (2) : Remove the built-in lz4 and download and install a new lz4:

```
# sudo apt-get remove liblz4-tool
# sudo git clone https://github.com/lz4/lz4.git
# cd /lz4
# sudo make
# sudo make install
```

## Configuration list

Configuration name	note
--------------------	------



EM3326S	Bare board, built-in WIFI&BT module, no button battery, no display
Power adapter	5V/3A
USB Micro cable	Used to download
Serial port module	Serial port debugging
Ethernet cable	1.5M
WIFI Antenna	Used for signal enhancement
U disk	Contains CD data