

EM3588 Android16 SSI User Manual

V1.0



Boardcon Embedded Design

Overview

The content of this document is intended solely for the EM3588 development board, aiming to help users quickly understand, apply, and test the EM3588 development board.

System Support

Development Board	Android16
MINI3588 V1 EM3588 V1	Y

Revision History

Version	Date	Author	Revision History
V1.0	2026-03-14	Xue Junchao	Initial version

Disclaimer

The information in this manual is provided for reference only. While Boardcon has made every effort to ensure the accuracy and reliability of the content, no guarantee is given regarding its completeness or correctness. All information in this manual is subject to change without prior notice.

Boardcon reserves the right to revise or update the contents of this manual at any time without prior notification. Boardcon shall not be held liable for any direct or indirect loss or damage arising from the use of this document or the products described herein.

Boardcon embedded design limited

2007-11 Haofang Tianji Plaza, 11008 Beihuan Avenue, Nanshan District,
Shenzhen, Guangdong, China. 518051

URL: www.armdesigner.com | www.boardcon.com

Email: market@armdesigner.com

Technical Support Inquiries: support@armdesigner.com

Tel: +86-755-26481393 | +86-755-27571591

Content

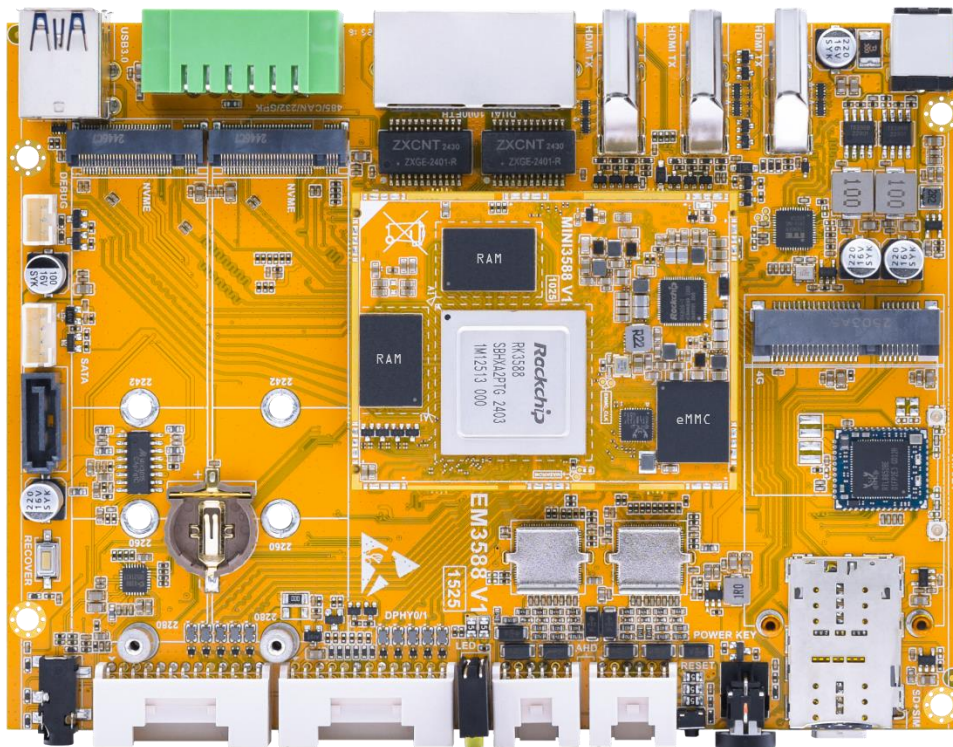
1.Introduction.....	4
1.1 Overview.....	4
1.2 Product Parameters	4
1.3 Hardware Interface Introduction.....	7
2.Install Drivers and Tool	8
2.1 Install RK Driver Assitant.....	9
2.2 Install CH9102X Driver.....	10
2.2.1 How to Connect the Serial Port Tool	10
2.2.2 Install Driver	10
2.3 Install Serial Terminal Tool.....	11
3.Upgrade Introduction	13
3.1 Upgrade Mode	13
3.1.1 How to Enter Loader Mode	14
3.1.2 How to Enter MaskRom Mode.....	15
3.2 Firmware Flashing	16
3.2.1 Flash Full Firmware Image.....	16
3.2.2 Flash Separate Partition Images.....	18
4.Development Environment	19
4.1 Preparing the Development Environment.....	19
4.2 Installing Libraries and Toolkits	19
4.3 Installing the JDK	20
5.Compile Source.....	20
6. Android16 Test.....	23
6.1 Serial Terminal.....	23
6.2 Display	24
6.3 USB Type-C.....	25
6.3.1 ADB	25

6.3.2 Type-C to USB3.0.....	26
6.3.3 DP Alt Mode	28
6.4 Ethernet.....	29
6.5 USB Host.....	30
6.7 SATA.....	33
6.8 WiFi & Bluetooth.....	35
6.8.1 WiFi test.....	35
6.8.2 Bluetooth test	40
6.9 4G & GPS	42
6.9.1 4G Test	42
6.9.2 GPS Test.....	45
6.10 Headset.....	46
6.11 RTC	48
6.12 RS485.....	50
6.13 CAN.....	51
6.14 RS232.....	53
6.15 Camera	54
6.16 PCIe M.2 SSD.....	56

1.Introduction

1.1 Overview

The EM3588 is a high-performance, low-power processor platform built on an octa-core 64-bit architecture, featuring four Cortex-A76 and four Cortex-A55 cores. It integrates a powerful NPU with up to 6 TOPS computing power, supports 8K video encoding and decoding, and includes advanced image processing capabilities with a 48MP ISP. The EM3588 provides extensive multimedia support, flexible display interfaces, rich peripheral connectivity, and high-speed communication interfaces, making it ideal for edge computing, intelligent vision systems, industrial control, and other embedded AI applications.



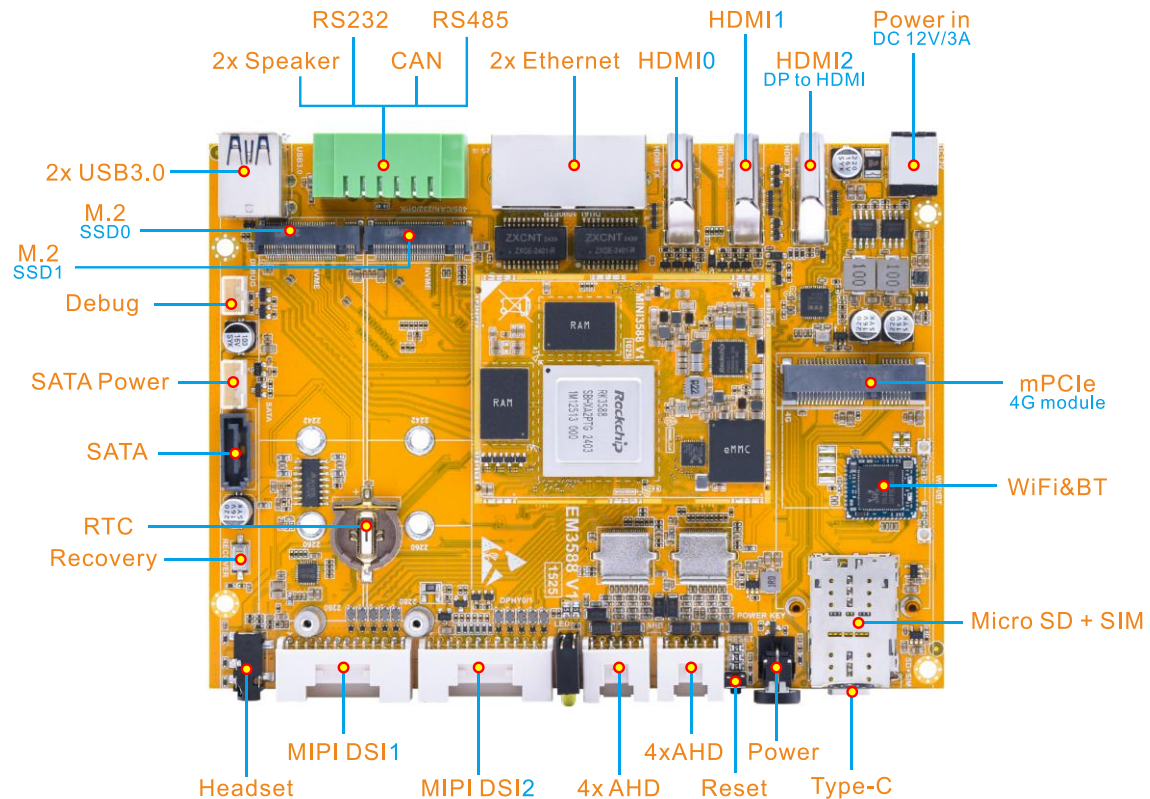
1.2 Product Parameters

Basic Parameters	
SOC	RK3588

CPU		• Octa-core 64-bit architecture: Quad-core Cortex-A76 + Quad-core Cortex-A55
GPU		• ARM Mali-G610 MC4 • OpenGL ES 1.1/2.0/3.2 • Vulkan 1.2 • OpenCL 2.2 • Embedded high performance 2D image acceleration module
NPU		• Up to 6 TOPS AI computing power • Supports int4, int8, int16, FP16, BF16, TF32 operation
Video	Decoder	• Support 8K@60fps H.265/VP9 video decoding • Support 8K@30fps H.264 video decoding • Support 4K@60fps AV1 video decoding • Support 1080P@60fps MPEG-2/MPEG-1/VC-1/VP8 video decoding
	Encoder	• Support H.264/H.265 video encoding, up to 8K@30fps
RAM		8GB LPDDR4X
ROM		32GB eMMC
Support system		Android16
Hardware Parameters		
Extended Storage		• Support 2x M.2 PCIe3.0 SSD • Support 1xSATA3.0 • Support 1x MicroSD Card
Display		• Support 2x HDMI2.1 TX, up to 7680x4320@60fps • Support 1x DP to HDMI2.0, up to 4K@60fps • Support 2x DSI MIPI, up to 4K@60fps • Support 1x DP Alt mode, up to 4K@60fps
Audio		• Support 2x HDMI TX audio output

	• Support 2x DP audio output • Support 1x Stereo speaker output • Support 1x Headphone output/input
USB	• Support 2xUSB3.0 • Support 1x USB3.1 Type-C (ADB/USB3.0/DP)
Network	• Support 2x Gigabit Ethernet • Support 1x WIFI/BT module • Support 1x 4G module
Camera	• Support 8xAHD
Peripheral communication	• Support 1xCAN • Support 1xRS485 • Support 1xRS232
Other parameters	Support 1xDebug, 1xRTC
Electrical Parameters	
Power supply input voltage	12V/3A
RTC input voltage	3V/0.6uA
Operating temperature	0~70°
Storage temperature	-40~85°
Structural Parameters	
Core board dimensions	62mm x 41mm
Motherboard dimensions	155mm x 115mm

1.3 Hardware Interface Introduction



Interface parameters	
Power in DC 12V	12V DC power input interface
HDMI2	DP1 TX to HDMI interface
HDMI1	HDMI TX1 interface
HDMI0	HDMI TX0 interface
2x Ethernet	2x Gigabit Ethernet RJ45 interface
RS485	RS485 communication interface
CAN	CAN communication interface
RS232	RS232 communication interface
2x Speaker	Stereo speaker output via 2x speaker channels (L/R)
2XUSB3.0	2x USB3.0 Host interface
M.2 SSD0	M.2 SSD0 interface

M.2 SSD1	M.2 SSD1 interface
Debug	Debug the serial port
SATA Power	SATA Power interface, 5V
SATA	SATA3.0 interface
RTC	RTC coin cell connector
Recovery	Recovery key
Headset	Headset output/input interface
MIPI DIS1	MIPI0 display interface
MIPI DIS2	MIPI1 display interface
4x AHD	AHD camera interface
4x AHD	AHD camera interface
Reset	Reset key
Power	Power key
Type-C	USB Type-C interface, OTG download interface
Mirco SD + SIM	All-in-one card holder
WIFI&BT	WIFI&Bluetooth module
mPCIe 4G module	4G model interface

2.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 assitant
2	CH9102x	SETUP.EXE	Serial port debugging driver
3	Serial Terminal Tool	SecureCRT.exe	Debugging tool

2.1 Install RK Driver Assitant

Step 1: Open [DriverAssitant_v5.1.1/DriverInstall.exe](#).

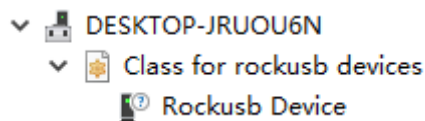
Step 2: To avoid driver conflicts, click “**Uninstall Driver**” to uninstall the driver.



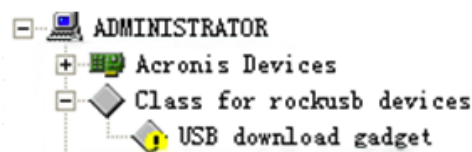
Step 3: Click button “**Install Driver**” to install.



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 in the **Device Manager** after the operation in Step 4, user need to proceed to the next step.



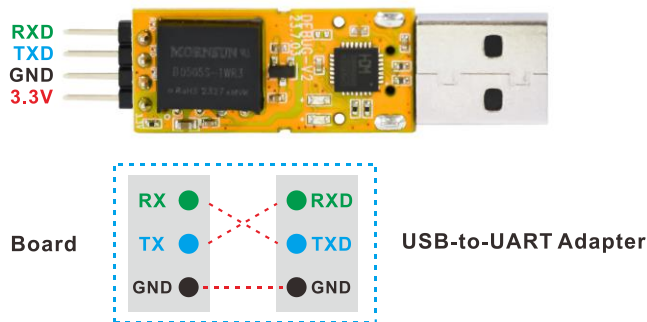
Step 6: The WINDOW will pop up found New Hardware Wizard dialog box, choose to install from the specified location, and then select: [DriverAssitant_v5.1.1/ADBDriver](#).

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



2.2 Install CH9102X Driver

2.2.1 How to Connect the Serial Port Tool



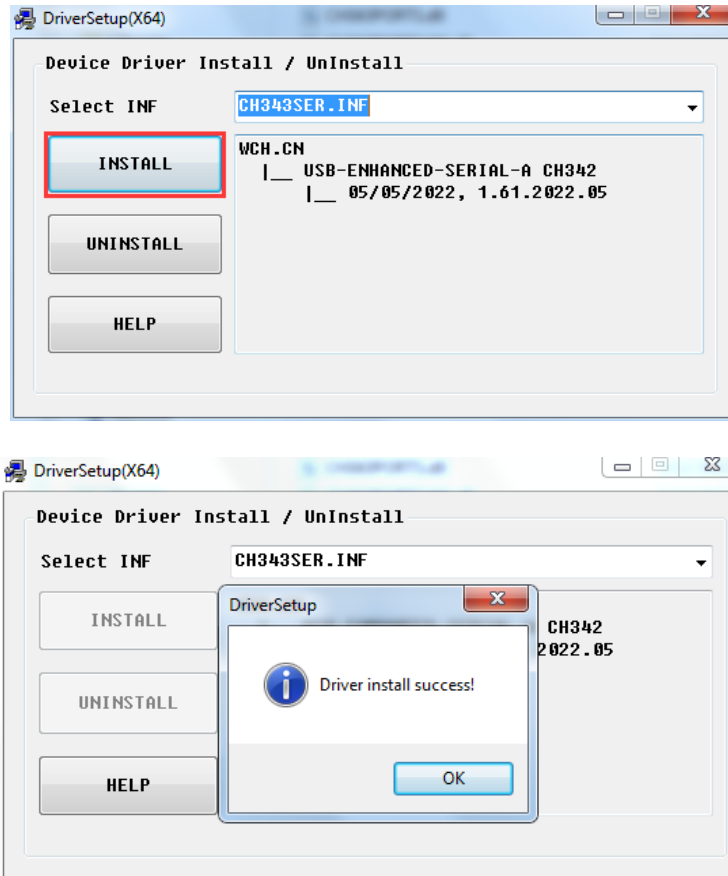
Pin	Connection Description
RXD	Receive, connect to TX pin of the board.
TXD	Transmit, connect to RX pin of the board.
GND	Ground, connect to GND pin of the board.
3V3	No need to connect.

2.2.2 Install Driver

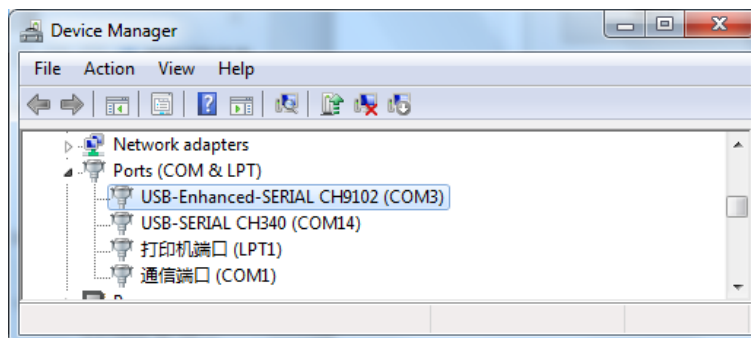
Step 1: Plug the CH9102X Module to the PC

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.

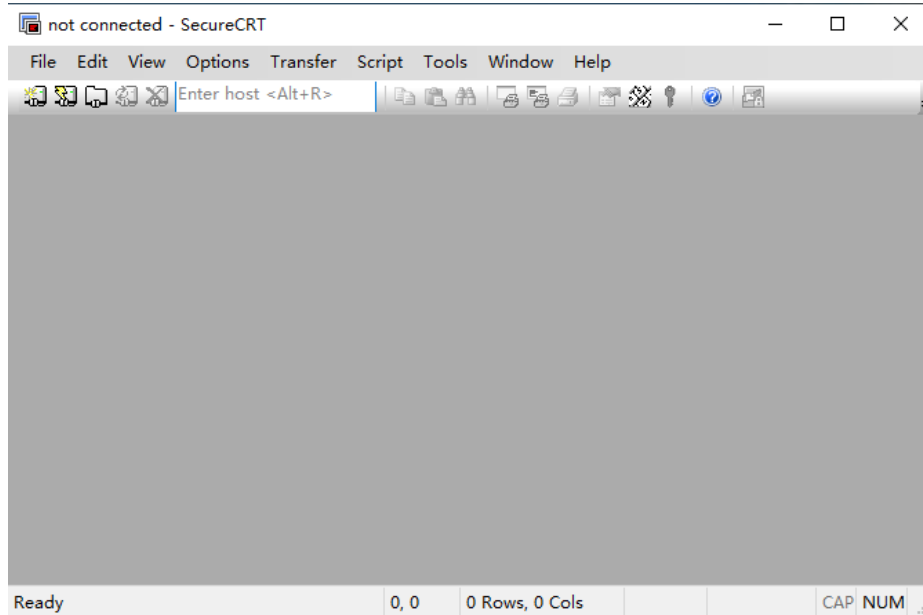


2.3 Install Serial Terminal Tool

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

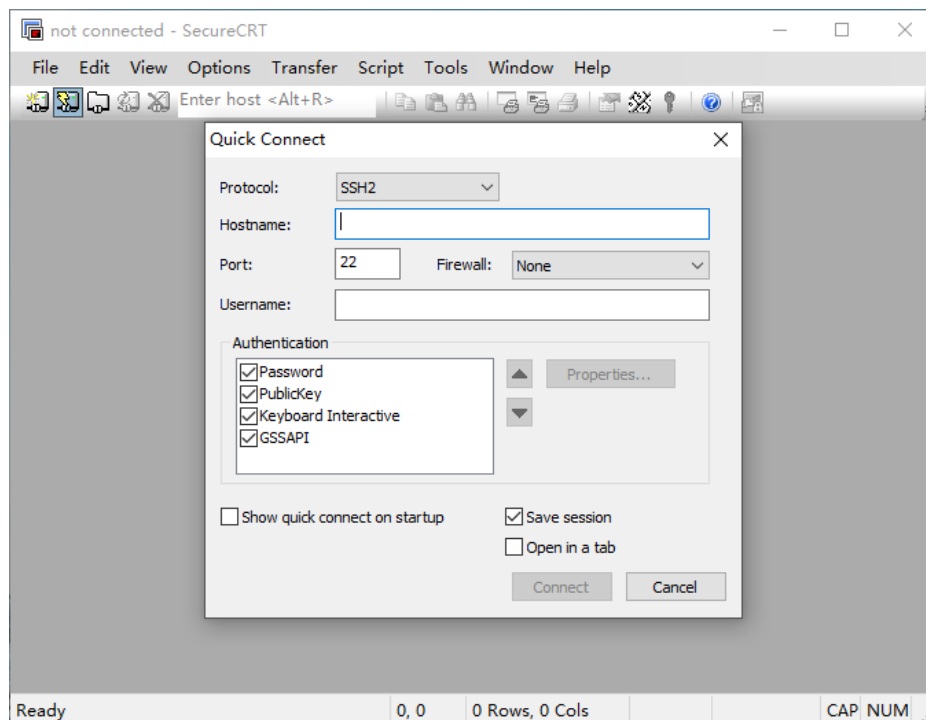
Step 1: Unzip *Platform/SecureCRT.rar* on PC.

Step 2: Click *SecureCRT/SecureCRT.exe* open the SecureCRT.

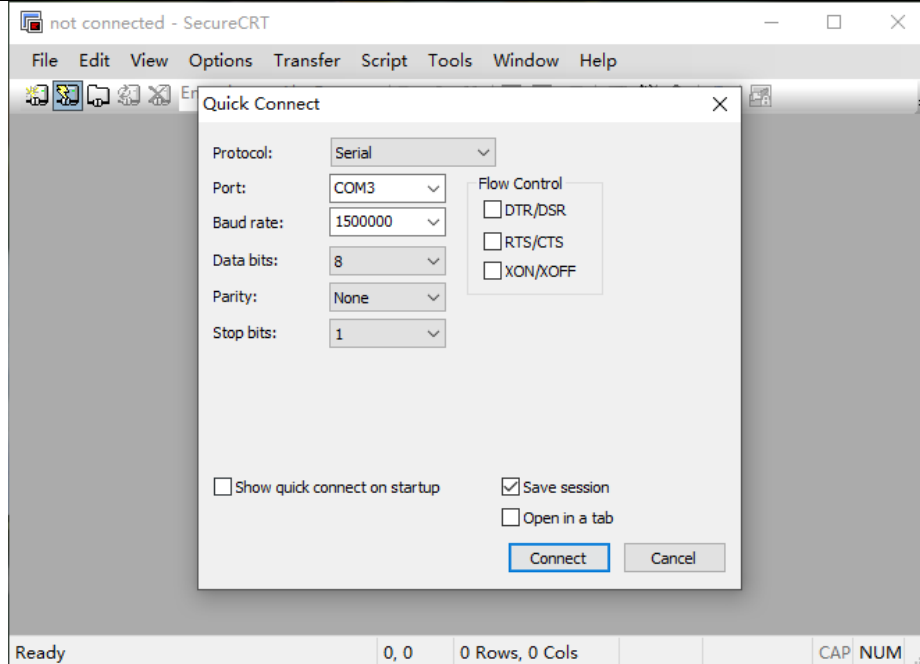


Step 3: Confirm that the CH9102X driver has been installed and the CH9102X module is connecting to the PC.

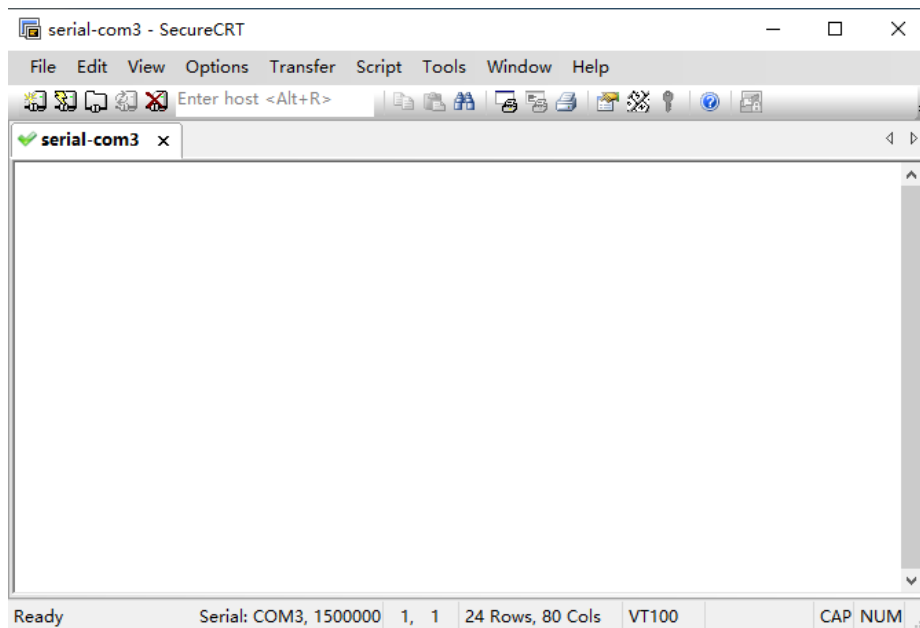
Step 4: Click the “**Quick Connect**” button to go to the Quick Connect configuration screen.



Step 5: Configure as shown in the following figure.



Step 6: After clicking “**Connect**” button, the terminal serial interface will be successfully accessed.



3.Upgrade Introduction

3.1 Upgrade Mode

The firmware can be upgraded via USB cable in two modes:

1. Loader Mode:

The standard mode used for firmware upgrades.

2. MaskRom Mode:

A last-resort mode used when the device is bricked. Entering MaskRom mode requires hardware manipulation, which involves certain risks. It should only be attempted if Loader mode is unavailable.

• Prerequisite

Before upgrading the firmware via USB cable, ensure that the necessary drivers are installed. For installation instructions, refer to the section [Install RK Driver Assistant](#).

3.1.1 How to Enter Loader Mode

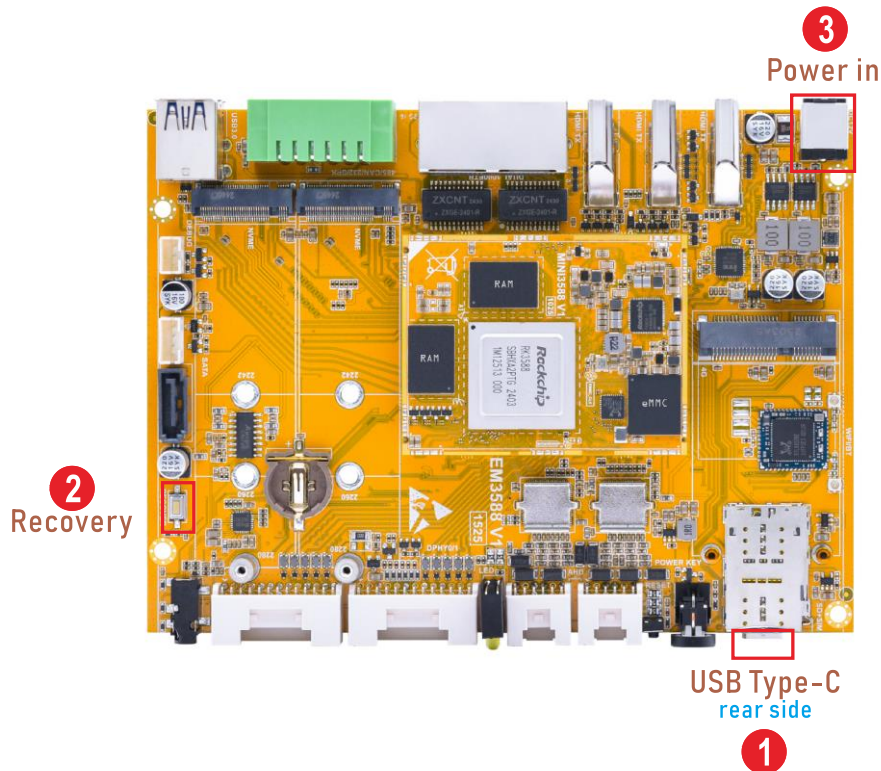
3.1.1.1 Hardware

Step 1: Disconnect the power adapter.

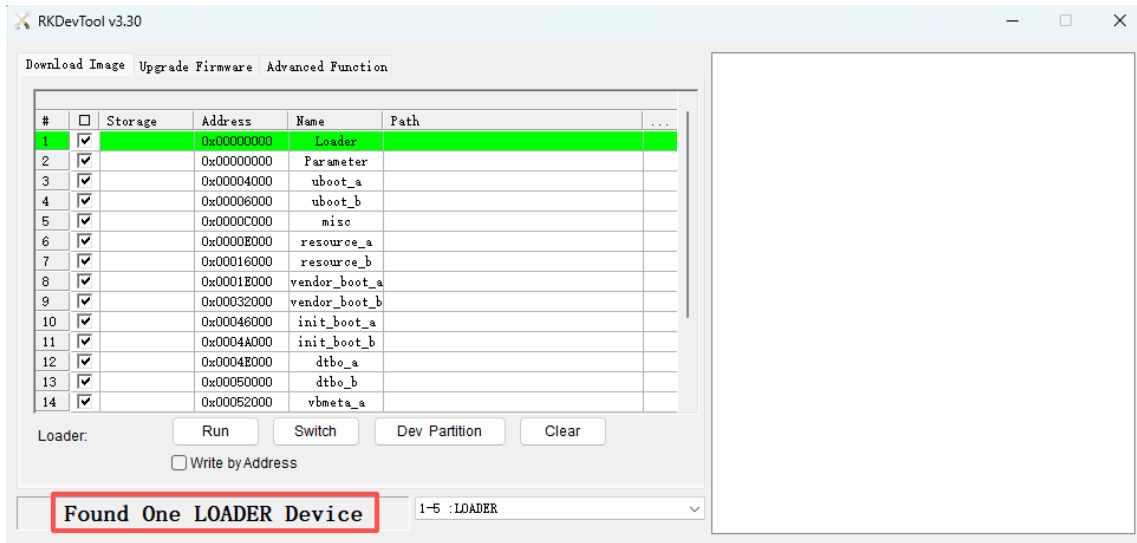
Step 2: Connect one end of the Type_C cable to the host and the other end to the development board.

Step 3: Press and hold the **Recovery** button on the board

Step 4: Connect the power supply.



Step 5: After a few seconds, release the **Recovery** button when the flashing tool shows **“Found one LOADER Device”**.



3.1.1.2 Software

After connecting the Type_C USB cable, execute the following command in the serial debug terminal or adb shell.

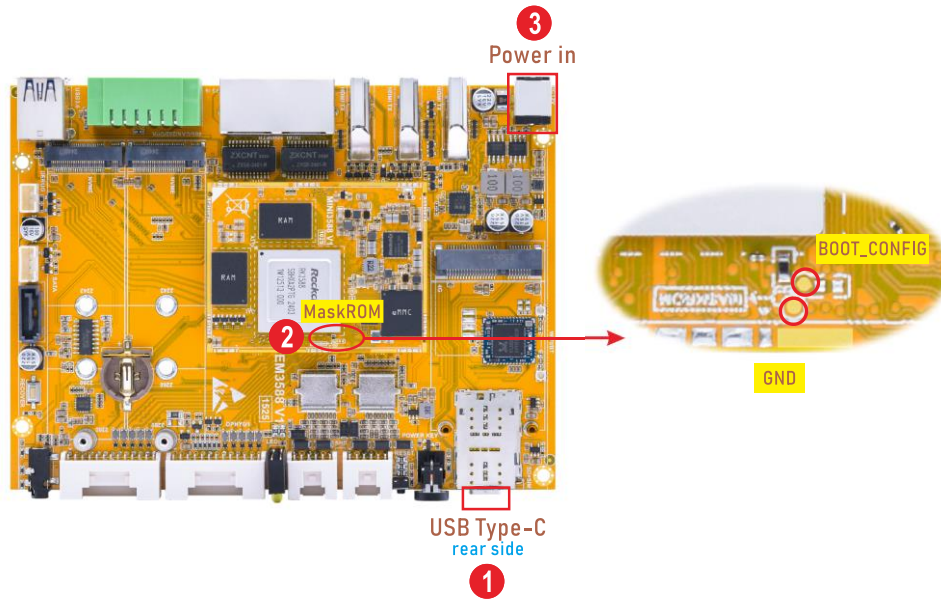
```
# reboot loader
```

3.1.2 How to Enter MaskRom Mode

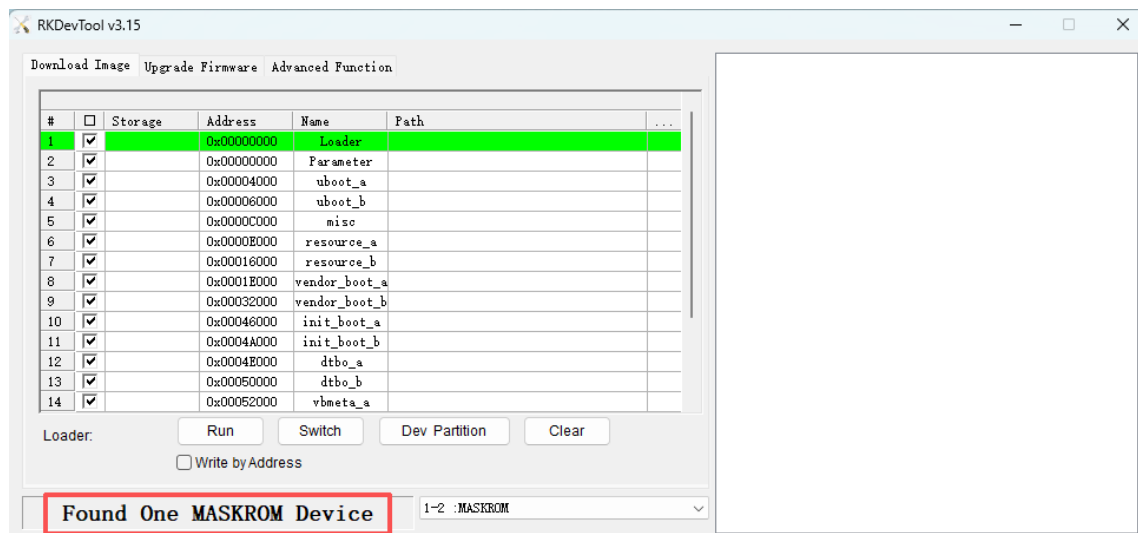
Step 1: Disconnect the power adapter.

Step 2: Connect one end of the Type_C cable to the host and the other end to the development board.

Step 3: Use tweezers to short the two test points on the MINI3588.



Step 4: After connecting the power cable, the device will enter MaskRom mode.



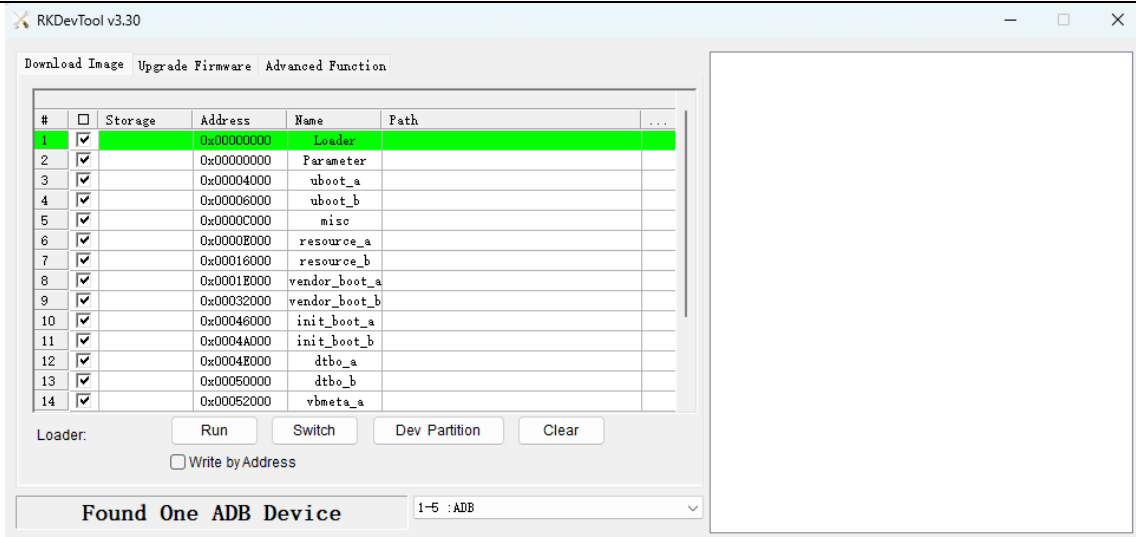
3.2 Firmware Flashing

Environment: Windows OS (Operating System).

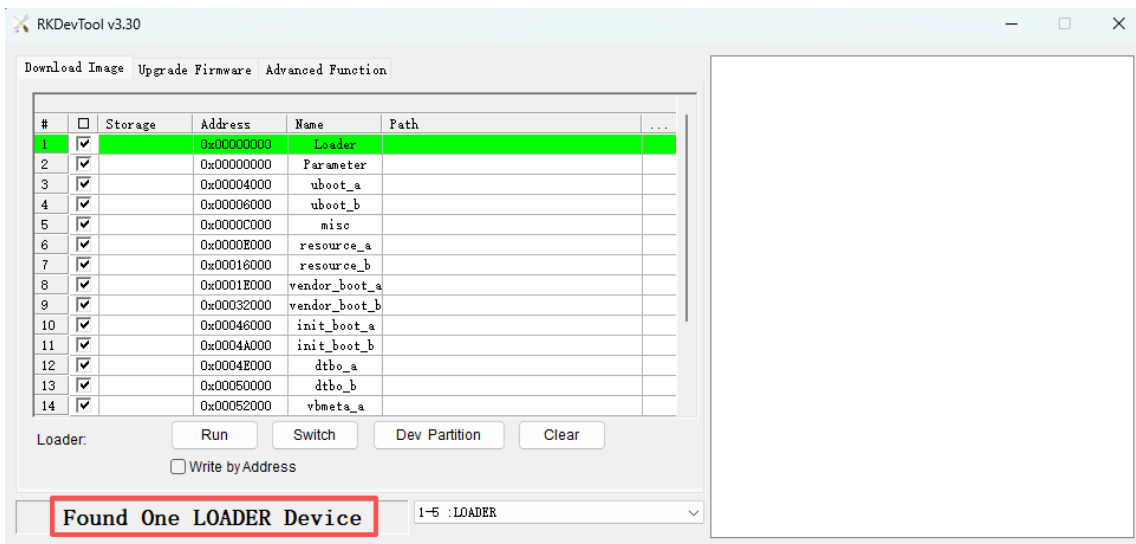
3.2.1 Flash Full Firmware Image

Step 1: Unzip *RKDevTool_v3.30_for_window.rar* on Windows.

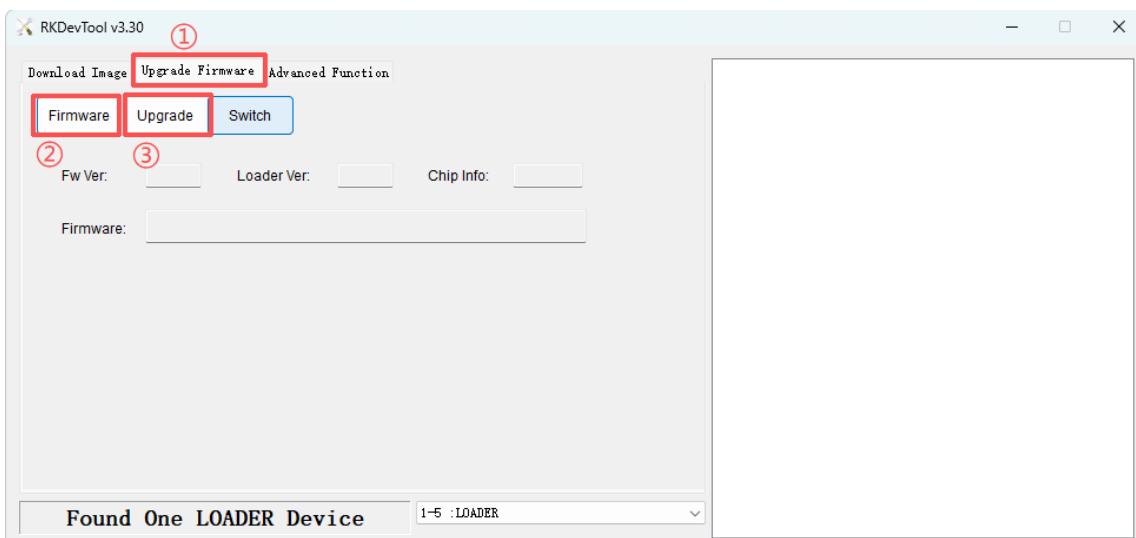
Step 2: Open *RKDevTool_v3.30_for_window\RKDevTool.exe*.



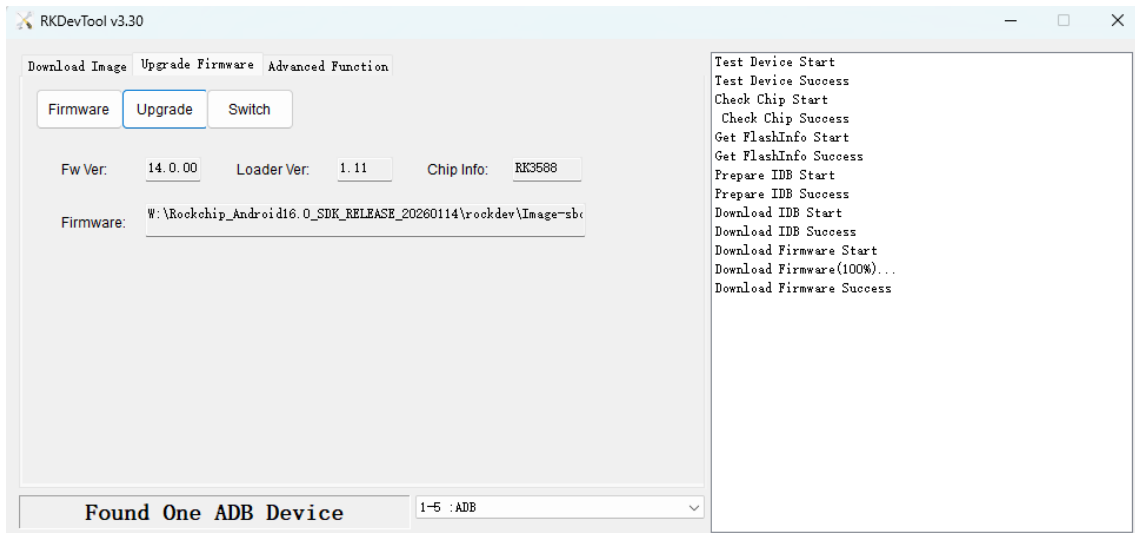
Step 3: Switch to loader mode. ([How to Enter Loader Mode](#))



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



After the flashing is complete, the board will automatically reboot.



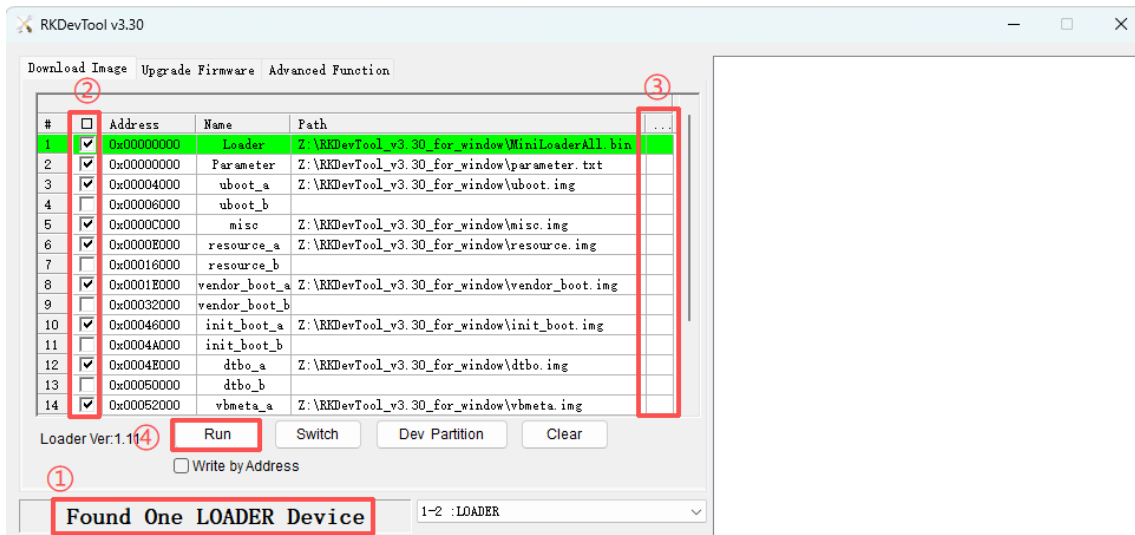
3.2.2 Flash Separate Partition Images

Step 1: Switch to **Loader mode**.

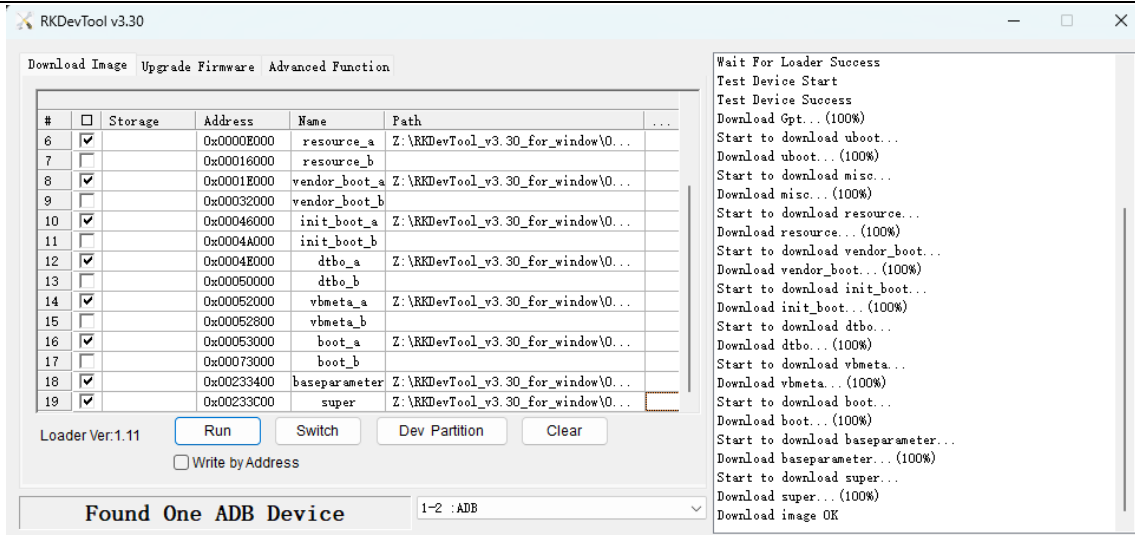
Step 2: Check the partitions to be flashed, multiple partitions can be selected.

Step 3: Ensure the image file path is correct. If necessary, click the blank cell next to the path to reselect it.

Step 4: Click the **Run** button to flash the image.



After the flashing is complete, the board will automatically reboot.



4. Development Environment

4.1 Preparing the Development Environment

It is recommended to use Ubuntu 22.04 or higher version for compilation. If you encounter an error during compilation, user 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 22.04

4.2 Installing Libraries and Toolkits

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	Network	Permission
Ubuntu 22.04	online	root

To install the required tools, execute the following commands:

```
$ sudo apt-get install git ssh make gcc libssl-dev liblz4-tool libmpc-dev
$ sudo apt-get install expect g++ patchelf chrpath gawk texinfo chrpath diffstat
$ sudo apt-get install binfmt-support live-build bison flex fakeroot libgmp-dev
$ sudo apt-get install cmake gcc-multilib g++-multilib unzip device-tree-compiler
$ sudo apt-get install ncurses-dev libgucharmap-2-90-dev bzip2 expat gpgv2
$ sudo apt-get install cpp-aarch64-linux-gnu g++-aarch64-linux-gnu
$ sudo apt install python2 python-is-python3
```

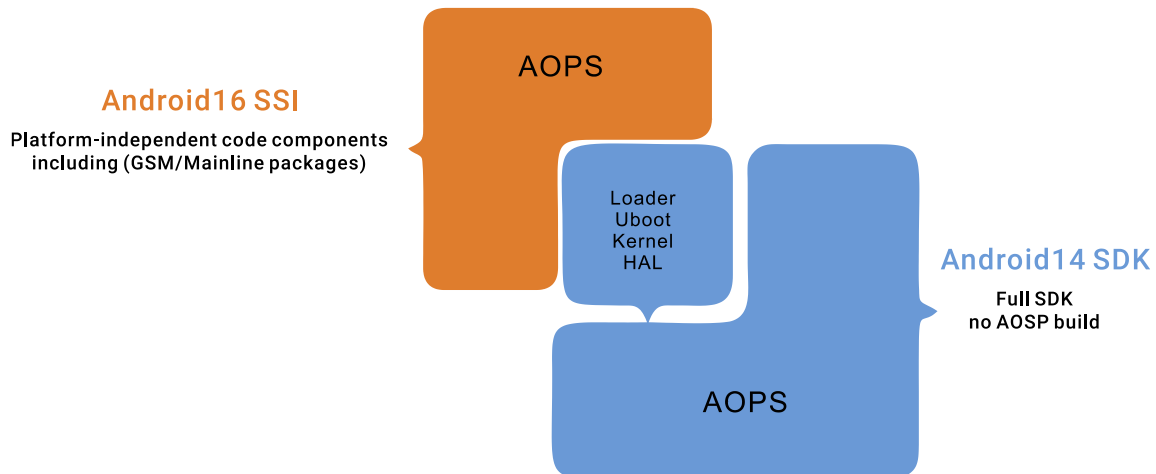
4.3 Installing the JDK

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

5.Compile Source

Rockchip Android 16.0 SSI SDK only releases aosp code, and after compilation it only generates system related images such as system.img, system_ext.img, product.img, which are identical to Google's GSI firmware. Other parts of the code need to be compiled using the Android 14 SDK, so Rockchip Android 16.0 SSI SDK needs to be used in conjunction with the Android 14 SDK.

The architecture diagram of the complete SDK composed of Rockchip Android 16.0 SSI SDK and Android14 SDK is as follows:



Step 1: To extract the source files, execute the following commands:

```
$ tar xvf Rockchip_Android16.0_SDK_RELEASE_20260114.tar.bz2
$ tar xvf Rockchip_Android14.0_SDK_RELEASE_20260115.tar.bz2
$ cd Rockchip_Android16.0_SDK_RELEASE_20260114
```

Step 2: Select the corresponding lunch, execute the following commands:

```
$ source build/envsetup.sh
$ lunch ssi_16_arm64-bp2a-userdebug
```

```
xuejunchao@boardcon:~/Rockchip_Android16.0_SDK_RELEASE_20260114$ source build/envsetup.sh
including device/rockchip/rockchip_generic/vendorsetup.sh
xuejunchao@boardcon:~/Rockchip_Android16.0_SDK_RELEASE_20260114$ lunch ssi_16_arm64-bp2a-userdebug
build/make/core/product_config.mk:489: warning: PRODUCT_INSTALL_DEBUG_POLICY_TO_SYSTEM_EXT is set but
PRODUCT_NAME (ssi_16_arm64) doesn't look like a GSI for compliance testing. This is a special
configuration for compliance GSI, so do make sure you understand the security implications before setting
this option. If you don't know what this option does, then you probably shouldn't set this.
build/make/core/product_config.mk:489: warning: PRODUCT_INSTALL_DEBUG_POLICY_TO_SYSTEM_EXT is set but
PRODUCT_NAME (ssi_16_arm64) doesn't look like a GSI for compliance testing. This is a special
configuration for compliance GSI, so do make sure you understand the security implications before setting
this option. If you don't know what this option does, then you probably shouldn't set this.
```

```
=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=16
TARGET_PRODUCT=ssi_16_arm64
TARGET_BUILD_VARIANT=userdebug
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=generic
HOST_OS=linux
HOST_OS_EXTRA=Linux-6.8.0-101-generic-x86_64-Ubuntu-22.04.5-LTS
HOST_CROSS_OS=windows
BUILD_ID=BP2A.250605.031.A3
OUT_DIR=out
SOONG_ONLY=false
=====
```

Step 3: Linking Android14 SDK and configuring build parameters:

```
$ rkbuild_setup
```

```
xuejunchao@boardcon:~/Rockchip_Android16.0_SDK_RELEASE_20260114$ rkbuild_setup
[WARN] rkbuild_setup -d <kernel_dts> -l <vendor_lunch> -f <build_number> -p <vendor_sdk_path> -j
<build_thread>
Enter parameters in the specified format or press Enter to input as prompted: Enter

Please enter the directory of Android_VENDOR_SDK (for example: /home/hcq/4_Android14_0409):
/Rockchip_Android14.0_SDK_RELEASE_20260115

[WARN] The optional lunch projects for Android_VENDOR_SDK are as follows:

You're building on Linux

Lunch menu .. Here are the common combinations:
 1. PX30_u-user
 2. PX30_u-userdebug
 3. aosp_arm-eng
 4. aosp_arm64-eng
 5. aosp_x86-eng
 6. aosp_x86_64-eng
 7. em3588_u-user
 8. em3588_u-userdebug
 9. idea3588_u-user
10. idea3588_u-userdebug
.....
45. sbc3588_u-user
46. sbc3588_u-userdebug

Please enter the lunch option for Android_VENDOR_SDK (for example: rk3588_u-user):em3588_u-userdebug

Do you need to specify the build number? (Press Enter to automatically generate, or directly enter the
required build number):8

Current PRODUCT_KERNEL_DTS=em3588-u, If you need to modify the DTS, please enter. Press Enter to use the
default value: Enter
```

Step 4: Start compilation.

Full Compilation:

```
$ rkbuild_build all
```

Partial Compilation:

```
$ rkbuild_build { uboot | kernel | ssi | vendor | all }
```

- uboot: Build the uboot in Android_VENDOR_SDK
- kernel: Build the kernel modules in Android_VENDOR_SDK
- ssi: Build the system SDK
- vendor: Build the GRF SDK
- all: Build the uboot & kernel & ssi & vendor, then pack the images to
update.img

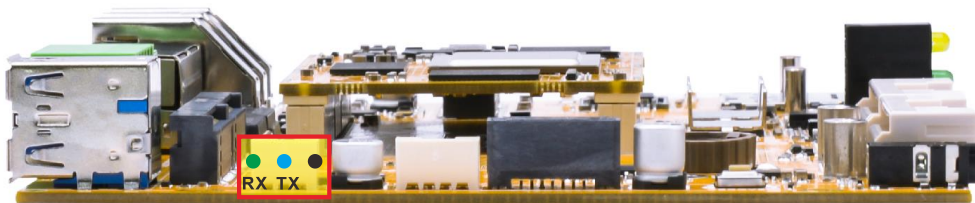
For partial compilation, need to perform merge operation after compilation:

```
$ rkbld_merge
$ rkbld_mkupdate
```

The path of the compiled firmware is /Rockchip_Android16.0_SDK_RELEASE_20260114/rockdev/Image-em3588_u/.

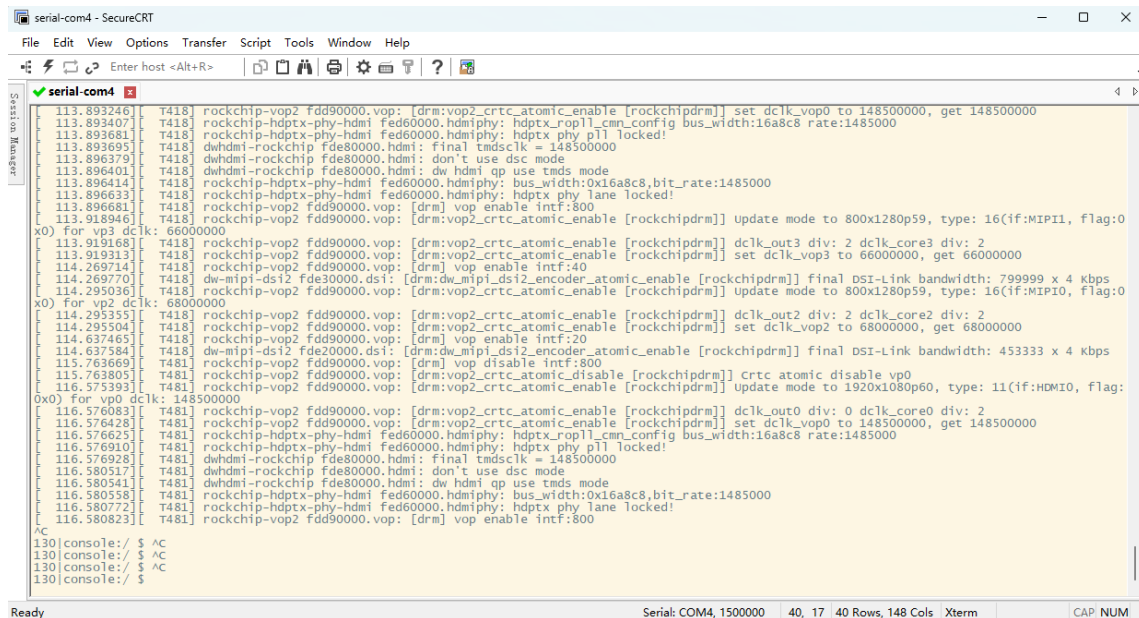
6. Android16 Test

6.1 Serial Terminal



Debug

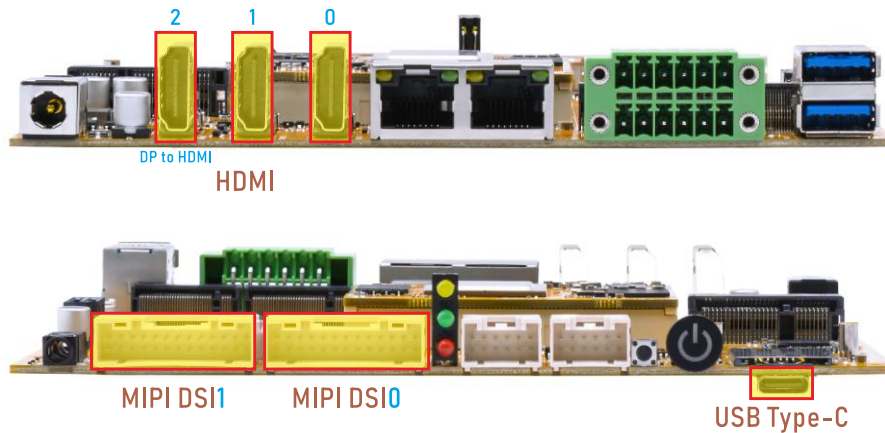
Connect the board and PC with USB Serial cable, then power on, the terminal will output boot information. The default baudrate is 1500000.



```
serial-com4 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
[+] serial-com4
113.893246 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] set dclk_vop0 to 148500000, get 148500000
113.893407 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: hdp1x_rop11_cm config bus_width:16a8c8 rate:1485000
113.893681 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: hdp1x_rop11_cm config bus_width:16a8c8 rate:1485000
113.893695 [T418] dw-hdmi-rockchip fde80000.hdmi: final tmdsclk = 148500000
113.896379 [T418] dw-hdmi-rockchip fde80000.hdmi: don't use dsc mode
113.896401 [T418] dw-hdmi-rockchip fde80000.hdmi: dw hdmi qp use tmds mode
113.896414 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: bus_width:0x16a8c8,bit_rate:1485000
113.896633 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: hdp1x_rop11_cm config bus_width:16a8c8 rate:1485000
113.896681 [T418] rockchip-vop2 fdd90000.vop: [drm] vop enable intf:800
113.918946 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] update mode to 800x1280p59, type: 16(if:MIP1, flag:0
X0) for vp3 dclk: 66000000
113.919168 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] dclk_out3 div: 2 dclk_core3 div: 2
113.919313 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] set dclk_vop3 to 66000000, get 66000000
114.269714 [T418] rockchip-vop2 fdd90000.vop: [drm] vop enable intf:40
114.269770 [T418] dw-mipi-dsi2 fde30000.dsi: [drm:dw_mipi_dsi2_encoder_atomic_enable [rockchipdrm]] final DSI-Link bandwidth: 799999 x 4 kbps
114.295036 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] update mode to 800x1280p59, type: 16(if:MIP10, flag:0
X0) for vp2 dclk: 68000000
114.295355 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] dclk_out2 div: 2 dclk_core2 div: 2
114.295504 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] set dclk_vop2 to 68000000, get 68000000
114.637465 [T418] rockchip-vop2 fdd90000.vop: [drm] vop enable intf:20
114.637584 [T418] dw-mipi-dsi2 fde20000.dsi: [drm:dw_mipi_dsi2_encoder_atomic_enable [rockchipdrm]] final DSI-Link bandwidth: 453333 x 4 kbps
115.763669 [T418] rockchip-vop2 fdd90000.vop: [drm] vop disable intf:800
115.763805 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_disable [rockchipdrm]] crtc atomic disable vp0
116.573393 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] update mode to 1920x1080p60, type: 11(if:HDMI0, flag:
0x0) for vp0 dclk: 148500000
116.576083 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] dclk_out0 div: 0 dclk_core0 div: 2
116.576428 [T418] rockchip-vop2 fdd90000.vop: [drm:vop2_crtc_atomic_enable [rockchipdrm]] set dclk_vop0 to 148500000, get 148500000
116.576625 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: hdp1x_rop11_cm config bus_width:16a8c8 rate:1485000
116.576910 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: hdp1x_rop11_cm config bus_width:16a8c8 rate:1485000
116.576928 [T418] dw-hdmi-rockchip fde80000.hdmi: final tmdsclk = 148500000
116.580517 [T418] dw-hdmi-rockchip fde80000.hdmi: don't use dsc mode
116.580541 [T418] dw-hdmi-rockchip fde80000.hdmi: dw hdmi qp use tmds mode
116.580558 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: bus_width:0x16a8c8,bit_rate:1485000
116.580772 [T418] rockchip-hdptx-phy-hdmi fed60000.hdmi: hdp1x_rop11_cm config bus_width:16a8c8 rate:1485000
116.580823 [T418] rockchip-vop2 fdd90000.vop: [drm] vop enable intf:800
AC
130|console:/ $ ^C
130|console:/ $ ^C
130|console:/ $ ^C
130|console:/ $ ^C
Ready Serial: COM4, 1500000 40, 17 40 Rows, 148 Cols Xterm CAP NUM
```


6.2 Display

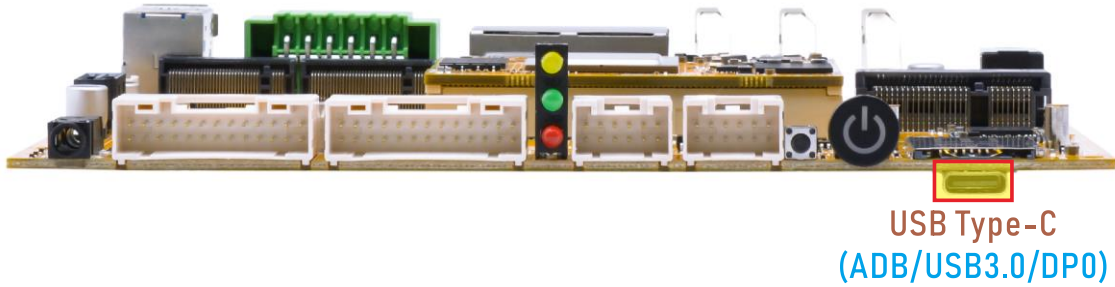
EM3588 supports up to 6-screen synchronous display by default under Android 16. To achieve synchronized output on similar interfaces (such as HDMI0/HDMI1 or HDMI2/DP Alt Mode), identical display timing must be used. In this case, the maximum supported resolution is 4K@60fps.



The display effect diagram is as follows:



6.3 USB Type-C

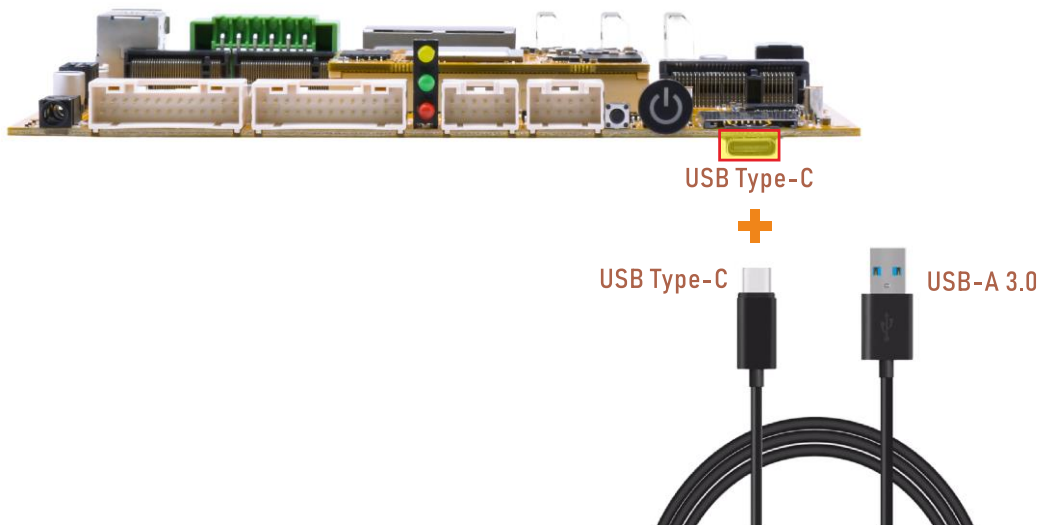


Features Supported by the Type-C Interface:

- **Reversible Design:** Allows insertion in either orientation for user convenience.
- **Data Transmission Roles:** Supports flexible designation of host and device roles.
- **Multiple Protocol Support:** Compatible with USB 3.0 and DisplayPort, enabling high-speed data transmission.

6.3.1 ADB

Step 1: Connect the board and PC host with Type-C cable.



Step 2: Install ADB driver on Windows system.

Step 3: Press **Windows + R** to open the Run program. Type “cmd” and press Enter.

Step 4: Execute the following command to enable ADB.

```
# adb shell
```

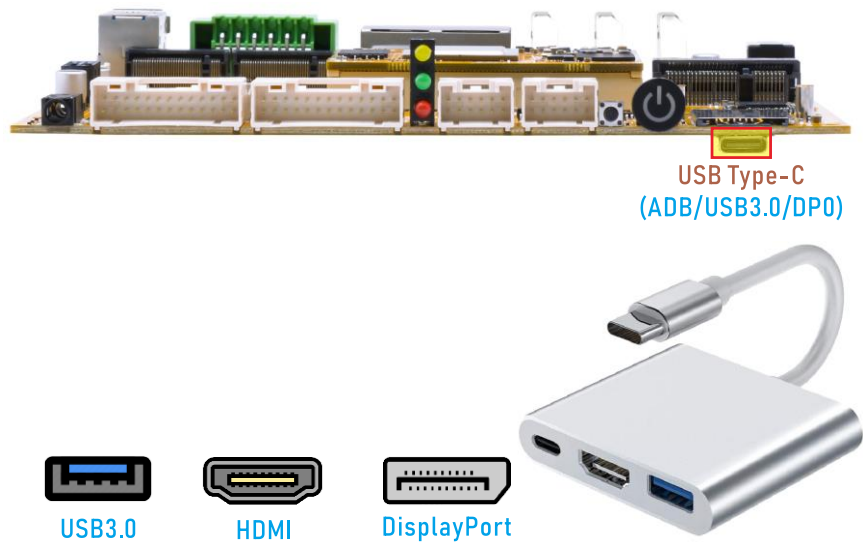
```
C:\WINDOWS\system32\cmd. X + v

Microsoft Windows [版本 10.0.26200.8037]
(c) Microsoft Corporation。保留所有权利。

C:\Users\28675>adb shell
ssi_16_arm64:/ $ |
```

6.3.2 Type-C to USB3.0

Step 1: Connect the Type-C to USB3.0 converter to the Type-C port of the development board, and then insert the USB3.0 flash drive into the USB3.0 port of the docking station.



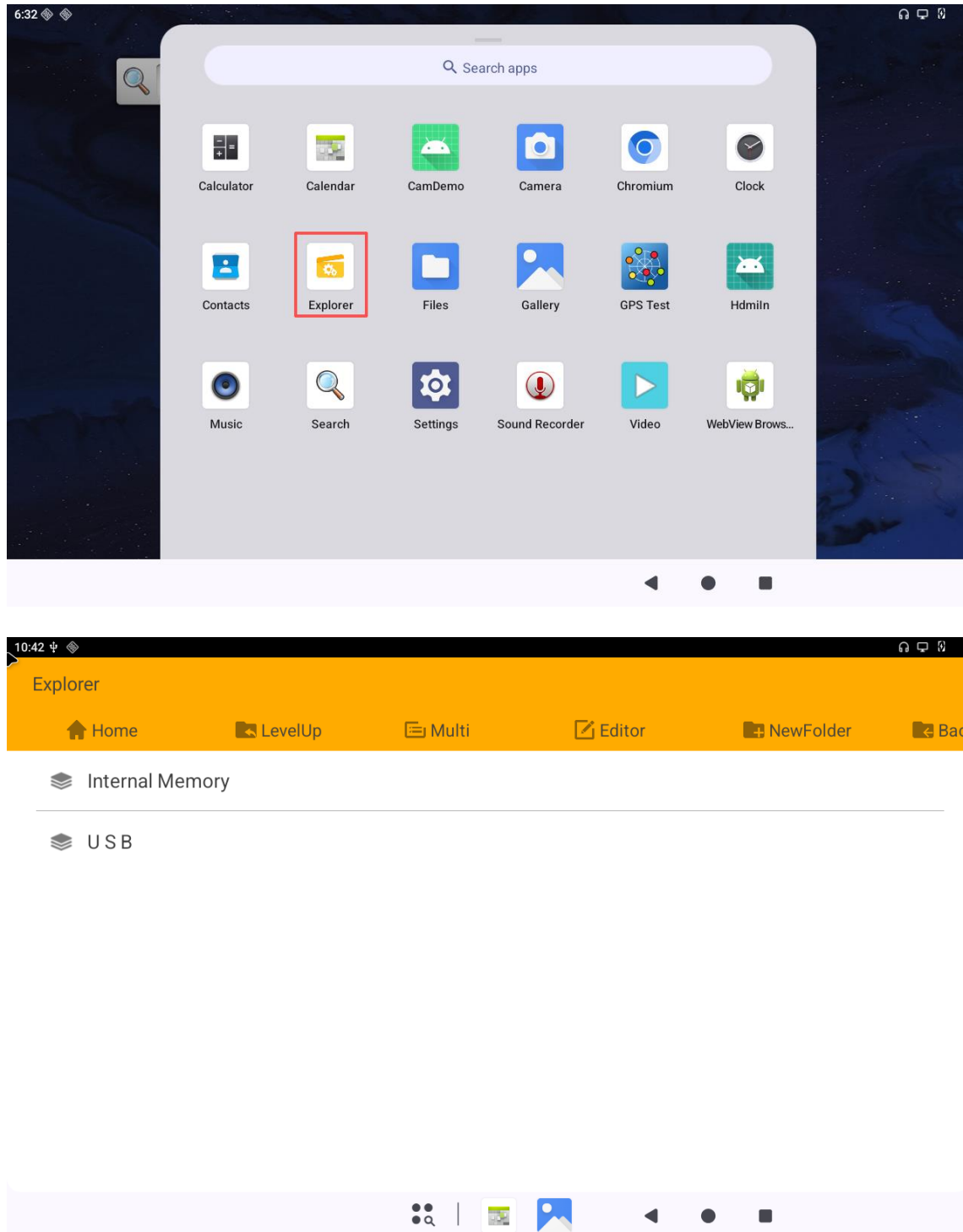
The serial port of the terminal will display the following message, indicating that the USB3.0 device is successfully recognized.

```

130|console:/ $ [ 111.777936][ T121] dwc3 fc000000.usb: request 0000000000000000 was not queued to
ep0out
[ 111.778050][ T120] android_work: did not send uevent (0 0 0000000000000000)
[ 111.828641][ T945] read descriptors
[ 111.828759][ T945] read strings
[ 111.887299][ T121] phy phy-fd5d0000.syscon:usb2-phy@0.8: illegal mode
[ 111.887330][ T121] xhci-hcd xhci-hcd.15.auto: xHCI Host Controller
[ 111.887351][ T121] xhci-hcd xhci-hcd.15.auto: new USB bus registered, assigned bus number 9
[ 111.887472][ T121] xhci-hcd xhci-hcd.15.auto: hcc params 0x0220fe64 hci version 0x110 quirks
0x00000008002010010
[ 111.887502][ T121] xhci-hcd xhci-hcd.15.auto: irq 79, io mem 0xfc000000
[ 111.887677][ T121] xhci-hcd xhci-hcd.15.auto: xHCI Host Controller
[ 111.887695][ T121] xhci-hcd xhci-hcd.15.auto: new USB bus registered, assigned bus number 10
[ 111.887708][ T121] xhci-hcd xhci-hcd.15.auto: Host supports USB 3.0 SuperSpeed
[ 111.888194][ T121] usb usb9: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice= 6.01
[ 111.888212][ T121] usb usb9: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 111.888224][ T121] usb usb9: Product: xHCI Host Controller
[ 111.888234][ T121] usb usb9: Manufacturer: Linux 6.1.145-android14-11-g4f184da362d5-ab14200214 xhci-
hcd
[ 111.888245][ T121] usb usb9: SerialNumber: xhci-hcd.15.auto
[ 111.888913][ T121] hub 9-0:1.0: USB hub found
[ 111.889185][ T121] hub 9-0:1.0: 1 port detected
[ 111.890911][ T121] usb usb10: We don't know the algorithms for LPM for this host, disabling LPM.
[ 111.890001][ T121] usb usb10: New USB device found, idVendor=1d6b, idProduct=0003, bcdDevice= 6.01
[ 111.890014][ T121] usb usb10: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 111.890024][ T121] usb usb10: Product: xHCI Host Controller
[ 111.890034][ T121] usb usb10: Manufacturer: Linux 6.1.145-android14-11-g4f184da362d5-ab14200214 xhci-
hcd
[ 111.890044][ T121] usb usb10: SerialNumber: xhci-hcd.15.auto
[ 111.890967][ T121] hub 10-0:1.0: USB hub found
[ 111.891250][ T121] hub 10-0:1.0: 1 port detected
[ 112.876530][ T945] read descriptors
[ 112.876652][ T945] read strings
[ 112.999077][ T169] usb 9-1: new SuperSpeed USB device number 2 using xhci-hcd
[ 113.149677][ T169] usb 9-1: New USB device found, idVendor=346d, idProduct=5678, bcdDevice= 2.00
[ 113.149781][ T169] usb 9-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 113.149826][ T169] usb 9-1: Product: HIKSEMI
[ 113.149864][ T169] usb 9-1: SerialNumber: FC0767D6251F4
[ 113.152798][ T169] usb-storage 9-1:1.0: USB Mass Storage device detected
[ 113.154167][ T169] usb-storage 9-1:1.0: Quirks match for vid 346d pid 5678: 420
[ 113.154350][ T169] scsi host1: usb-storage 9-1:1.0
[ 113.919092][ T945] read descriptors
[ 113.919137][ T945] read strings
[ 114.179295][ T169] scsi 1:0:0:0: Direct-Access HIKSEMI 3.00 PQ: 0 ANSI: 4
[ 114.179805][ T169] scsi 1:0:0:0: Attached scsi generic sg0 type 0
[ 114.187441][ T87] sd 1:0:0:0: [sda] 122880000 512-byte logical blocks: (62.9 GB/58.6 GiB)
[ 114.187625][ T87] sd 1:0:0:0: [sda] Write Protect is off
[ 114.187806][ T87] sd 1:0:0:0: [sda] No Caching mode page found
[ 114.187819][ T87] sd 1:0:0:0: [sda] Assuming drive cache: write through
[ 114.190552][ T87] sda: sda1
[ 114.190814][ T87] sd 1:0:0:0: [sda] Attached SCSI removable disk

```

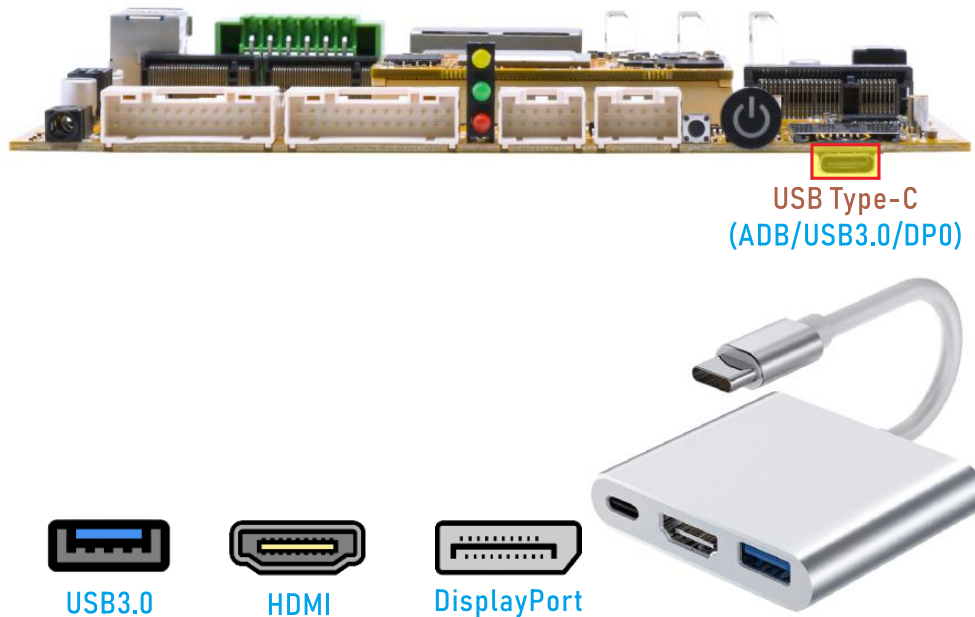
Step 2: After connecting the USB flash drive, the device will automatically mount, and it can be accessed directly through the Explorer app.



6.3.3 DP Alt Mode

DisplayPort Alternate Mode (DP Alt Mode) is a technology that facilitates the transmission of DisplayPort video signals through a USB Type-C interface. It enables devices to output video and audio via a USB-C connection without requiring a dedicated

DisplayPort connector. This allows users to connect a display using a single USB-C port while transmitting video, audio, and additional data.

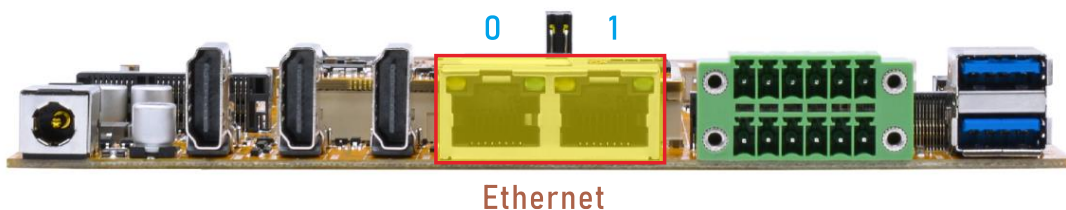


Currently, Boardcon has conducted the following output tests:

- **Type-C to HDMI:** Supports 3840x2160p60 resolution.
- **Type-C to DisplayPort (DP):** Supports 3840x2160p60 resolution.

6.4 Ethernet

Step 1: Connect the network cable to the Ethernet port.



According to the log, it can be seen that the Gigabit Ethernet recognition is successful.

```
console:/ $ [ 546.277137][ T2864] rk_gmac-dwmac fe1c0000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[ 546.277333][ T87] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 557.540887][ T139] rk_gmac-dwmac fe1b0000.ethernet eth1: Link is Up - 1Gbps/Full - flow control rx/tx
[ 557.540969][ T87] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
```

Step 2: View network interface information.

```
# ifconfig
```



```
console:/ $ ifconfig
eth0      Link encap:Ethernet HWaddr 5a:0d:96:f9:98:96 Driver rk_gmac-dwmac
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:304 errors:0 dropped:35 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:35018 TX bytes:7527
          Interrupt:73

eth1      Link encap:Ethernet HWaddr 56:0d:96:f9:98:96 Driver rk_gmac-dwmac
          inet addr:192.168.0.55 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::398d:8357:8b24:cbc4/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:270 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30396 TX bytes:4753
          Interrupt:75
```

Note: When eth0 and eth1 are enabled simultaneously, eth0 is dedicated to internal network communication, while eth1 is dedicated to external network communication.

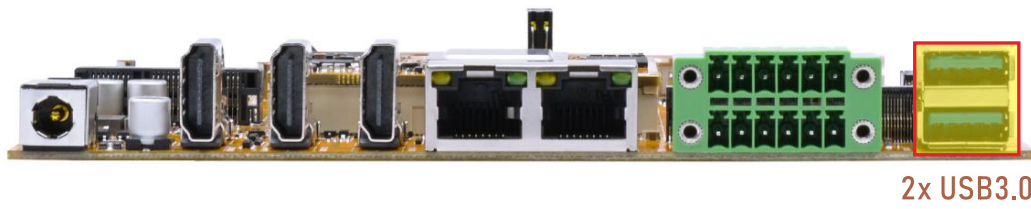
Step 3: Network connection test.

```
# ping -I eth0 192.168.0.1
# ping -I eth1 www.armdesigner.com
```

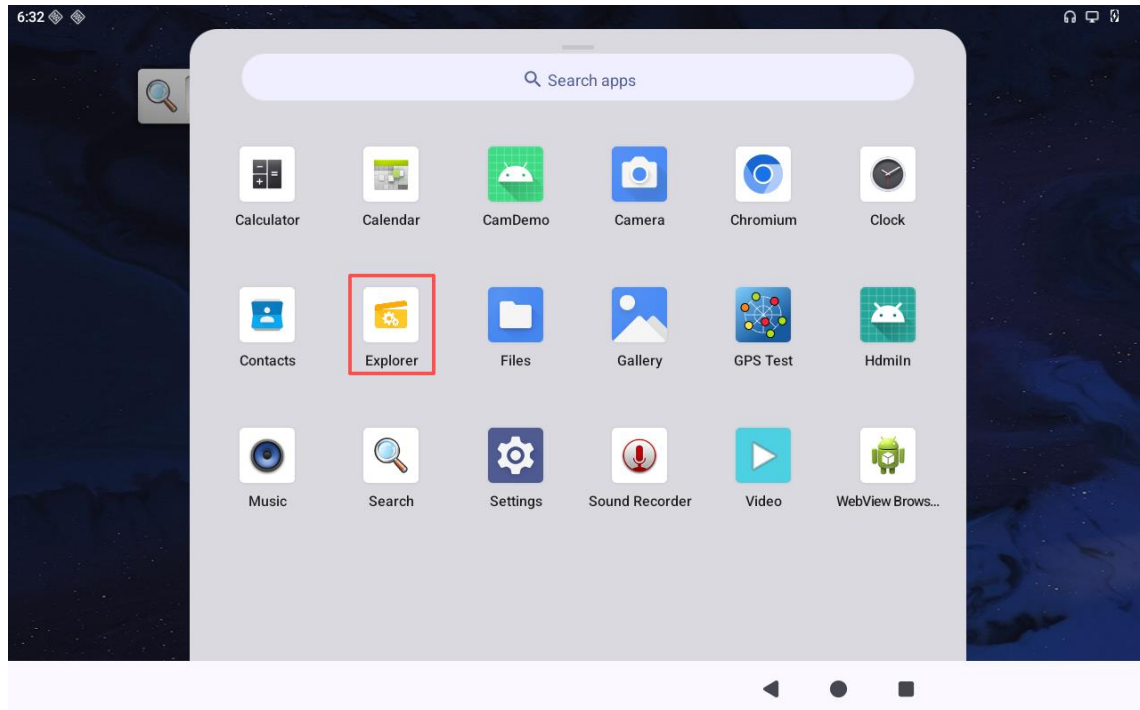
```
console:/ $ ping -I eth0 192.168.0.1
ping: Warning: source address might be selected on device other than eth0.
PING 192.168.0.1 (192.168.0.1) from 192.168.0.55 eth0: 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=3.03 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=0.420 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=0.300 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=255 time=0.343 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=255 time=0.604 ms
^C
--- 192.168.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4046ms
rtt min/avg/max/mdev = 0.300/0.940/3.035/1.052 ms
console:/ $
console:/ $ ping -I eth1 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.55 eth1: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=50 time=190 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=50 time=190 ms (DUP!)
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=50 time=192 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=50 time=192 ms (DUP!)
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=50 time=191 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=50 time=191 ms (DUP!)
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=50 time=190 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=50 time=190 ms (DUP!)
^C
--- www.armdesigner.com ping statistics ---
4 packets transmitted, 4 received, +4 duplicates, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 190.428/191.140/192.173/0.676 ms
```

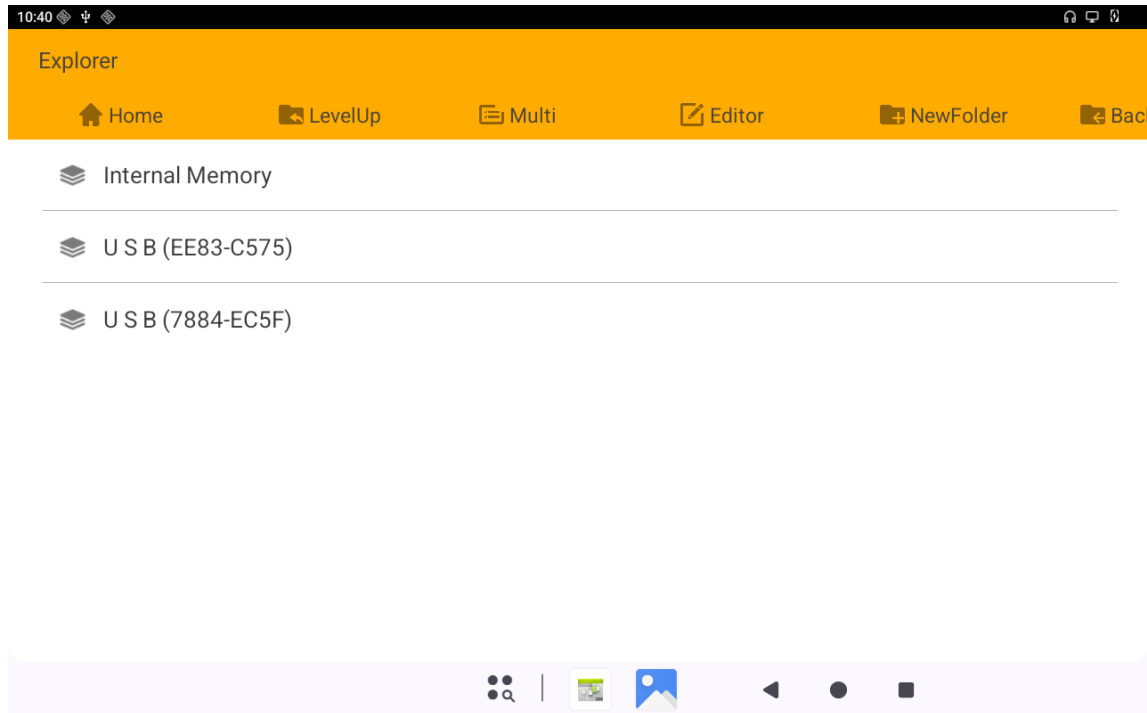
6.5 USB Host

The USB host can be used to connect devices such as USB mouse, USB keyboards, USB flash drives, and other USB peripherals.



After connecting the USB flash drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.

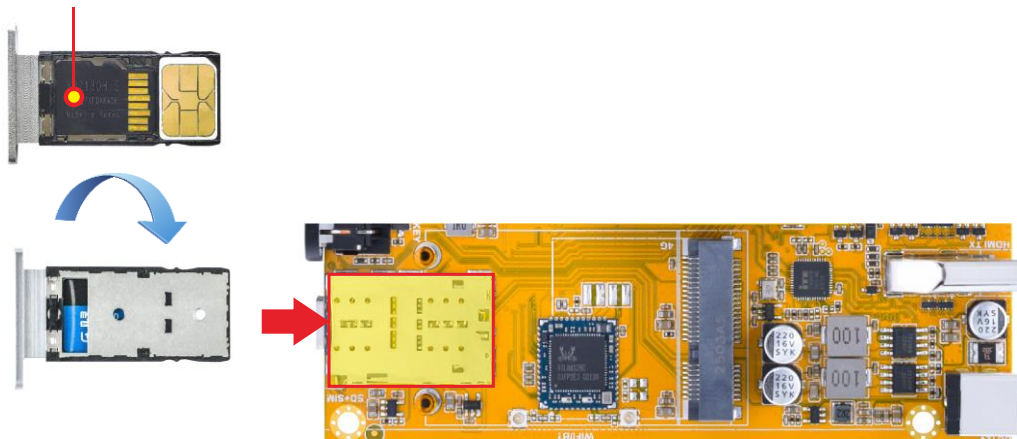




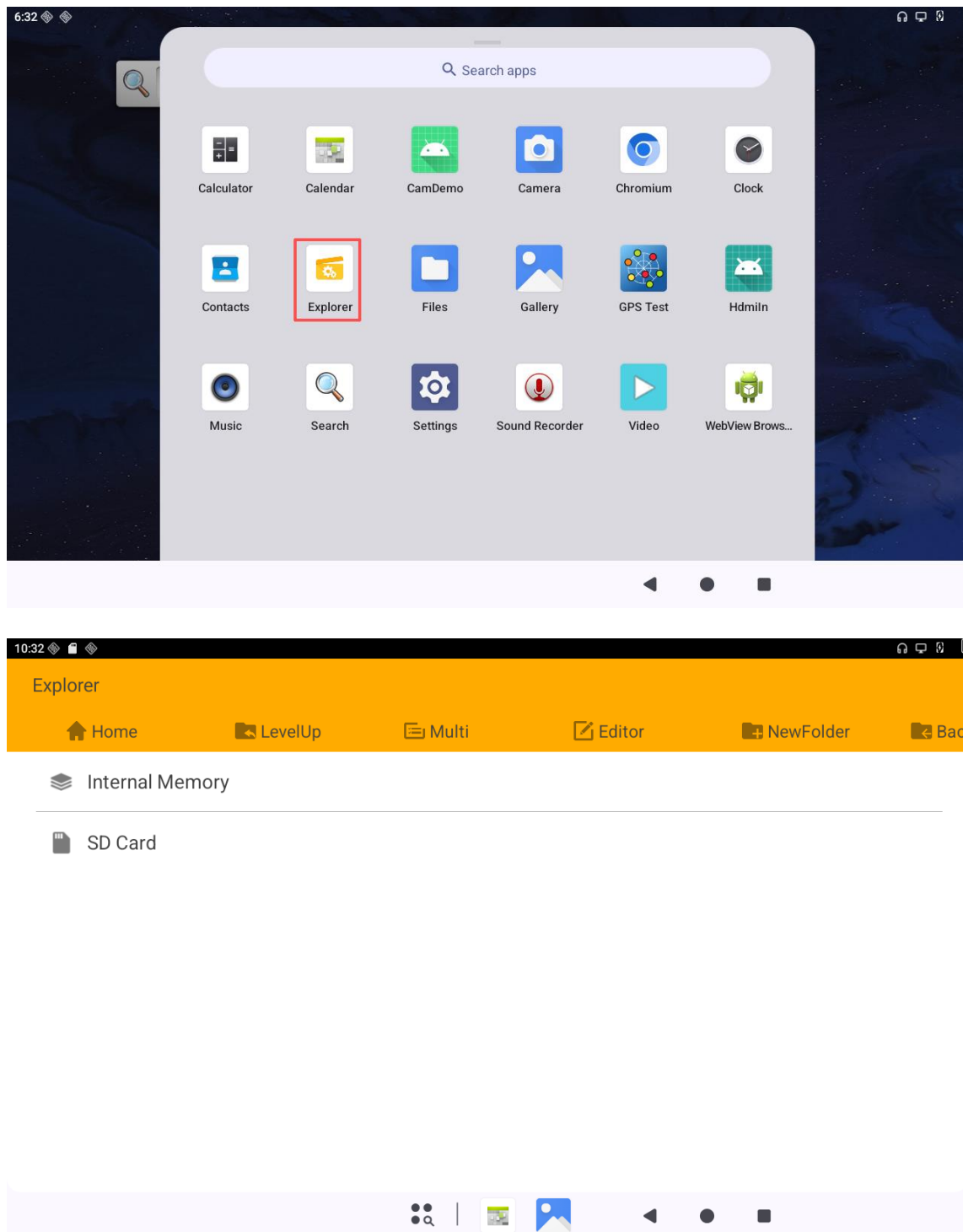
6.6 SD Card

Step 1: Insert the micro SD card into the card slot.

Micro SD



Step 2: After connecting the SD card drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.

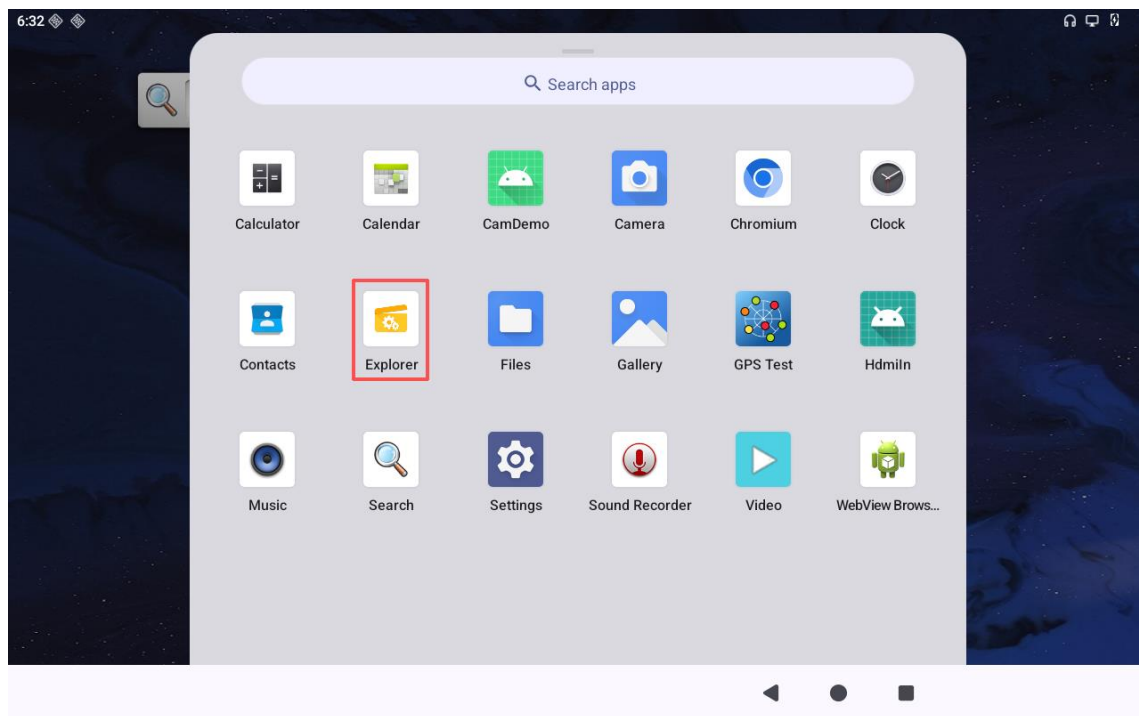


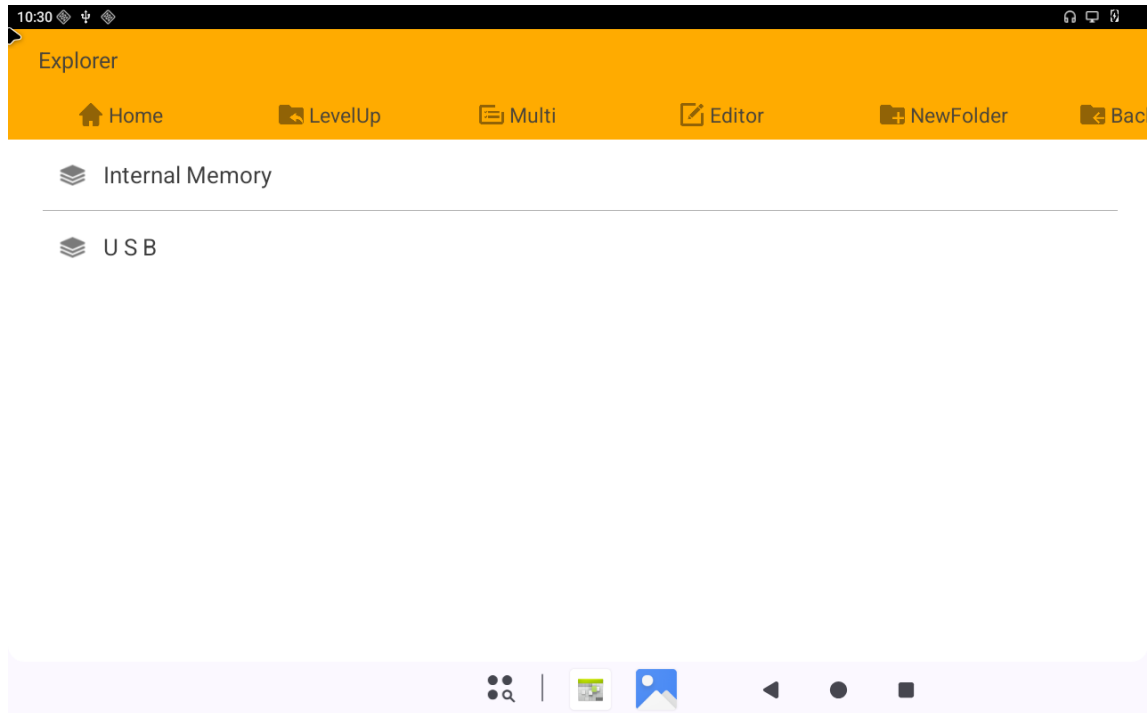
6.7 SATA

Step 1: Connect the sata and sata power, then power on.



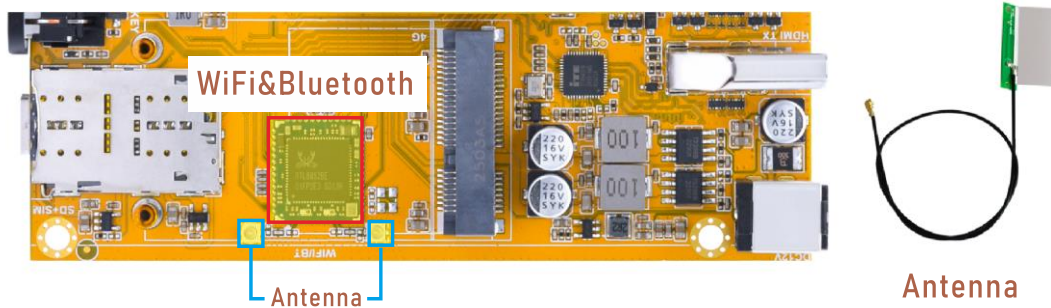
Step 2: After connecting the sata drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.





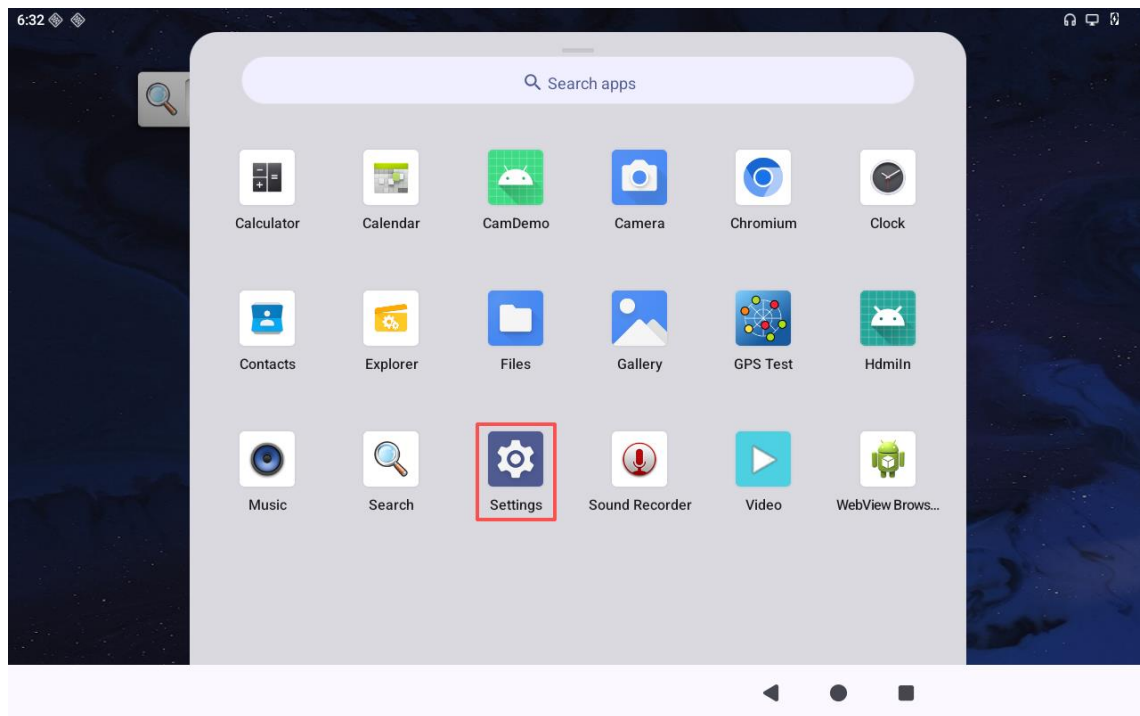
6.8 WiFi & Bluetooth

To use Wi-Fi and Bluetooth functions properly, the antenna needs to be connected.

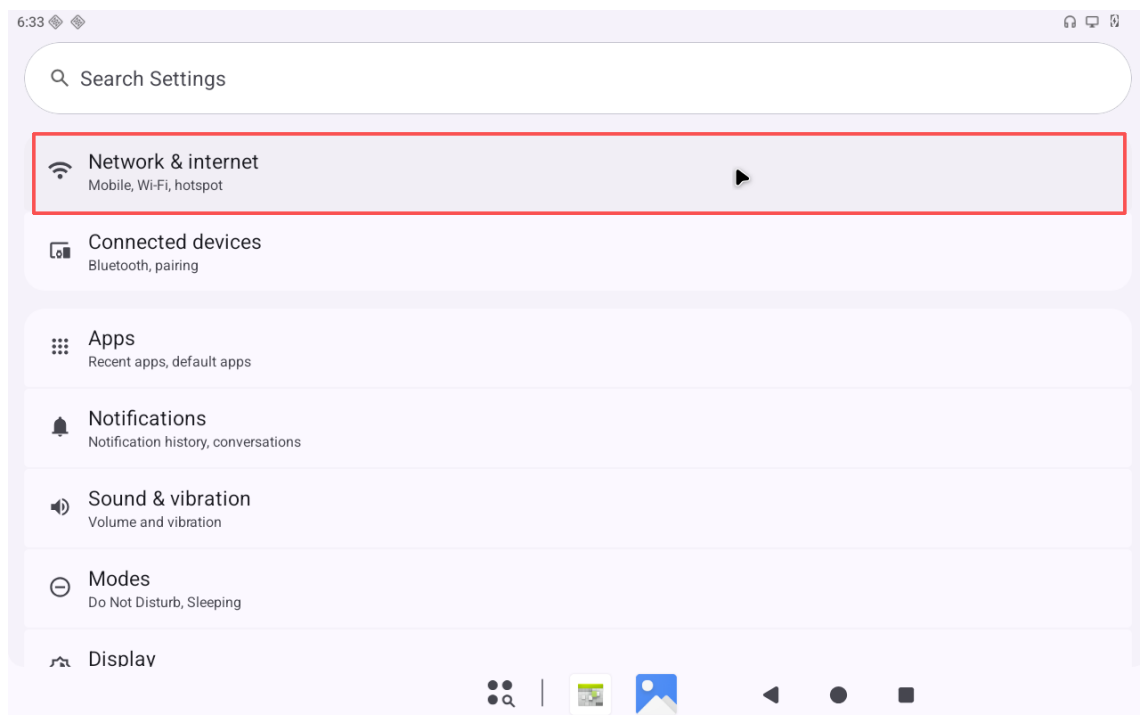


6.8.1 WiFi test

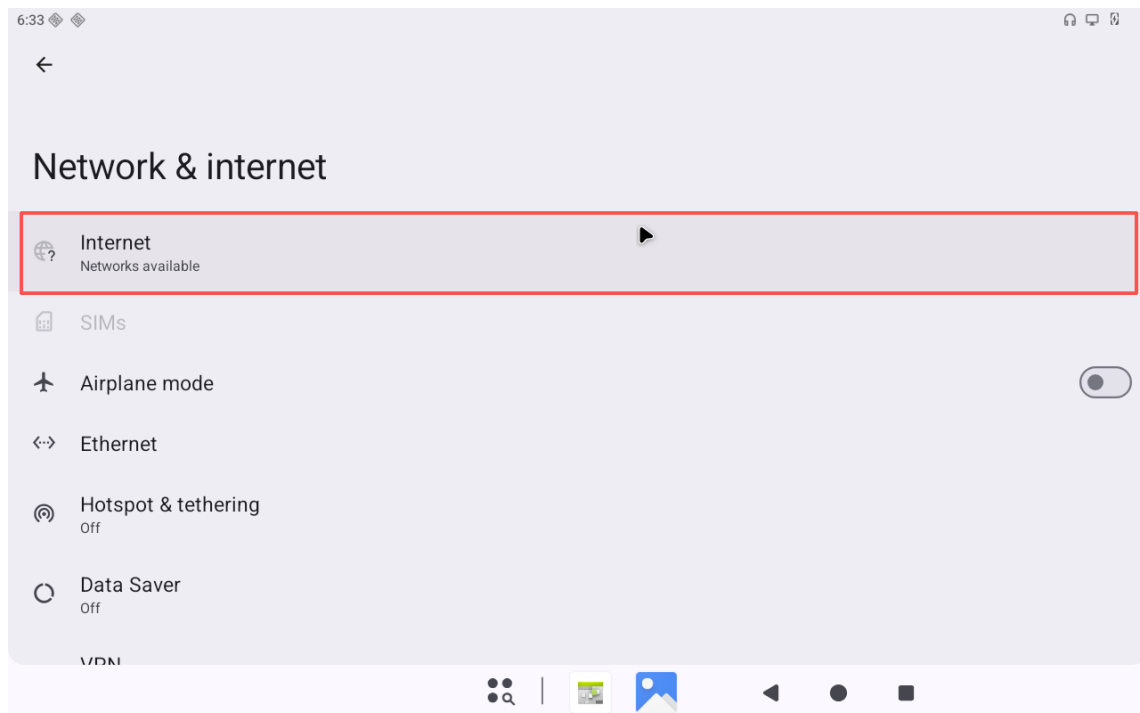
Step 1: Open the **Settings** app:



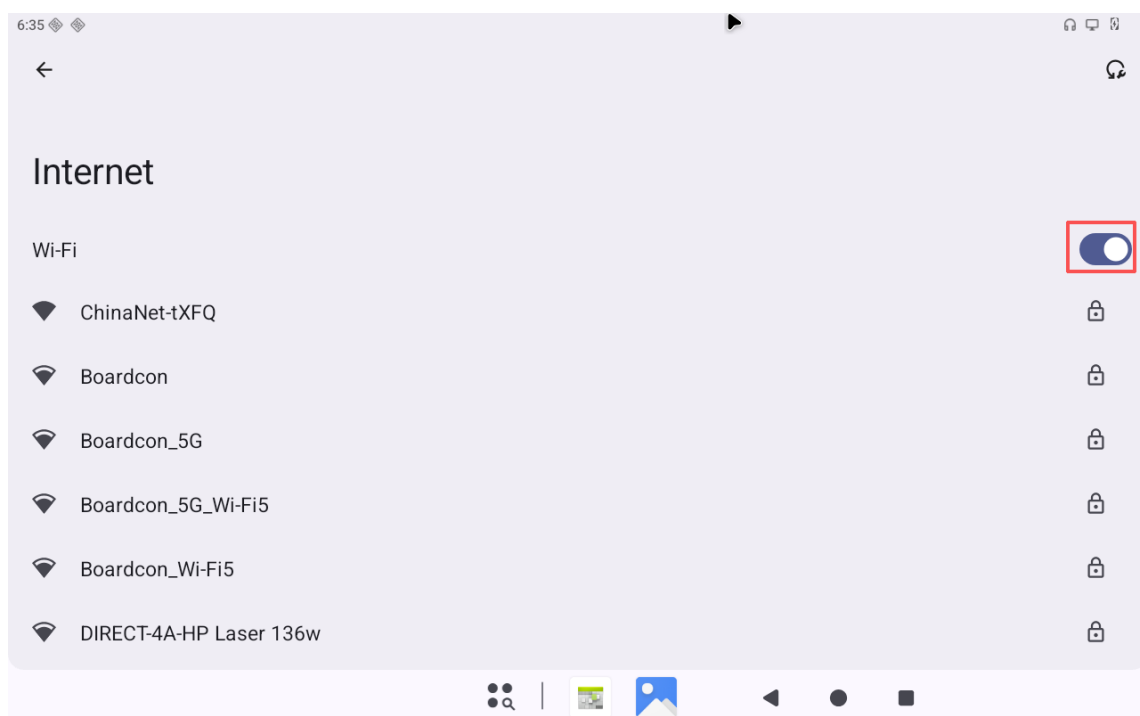
Step 2: Go to Network & Internet:



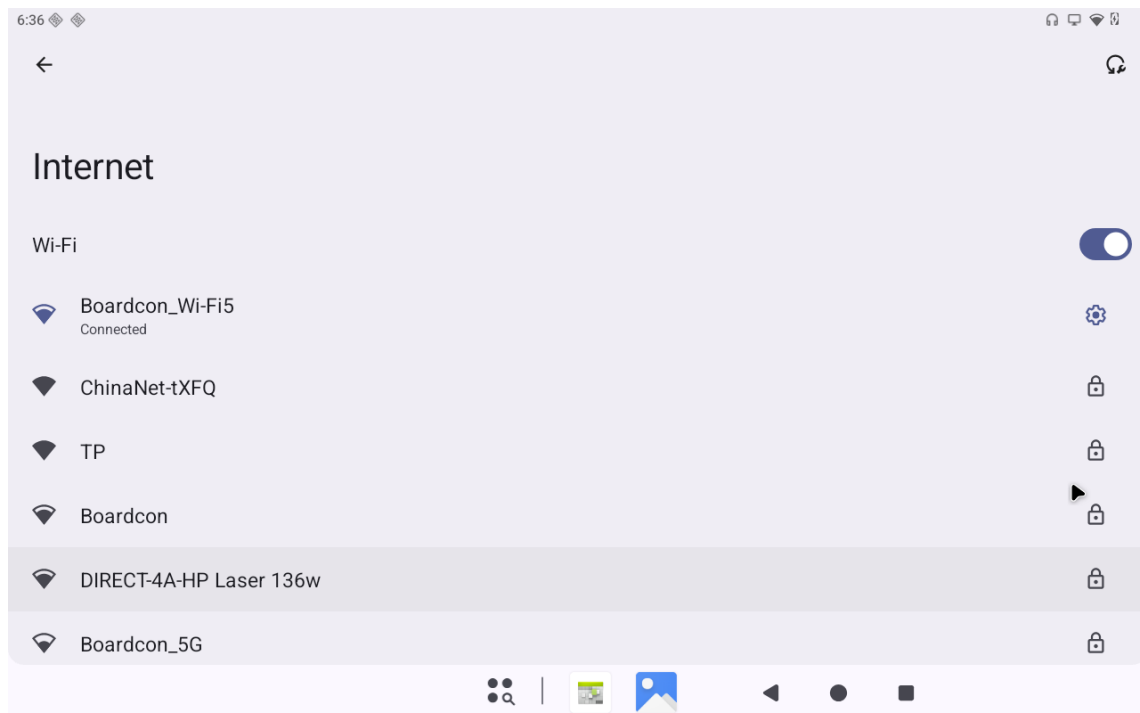
Step 3: Tap Internet:



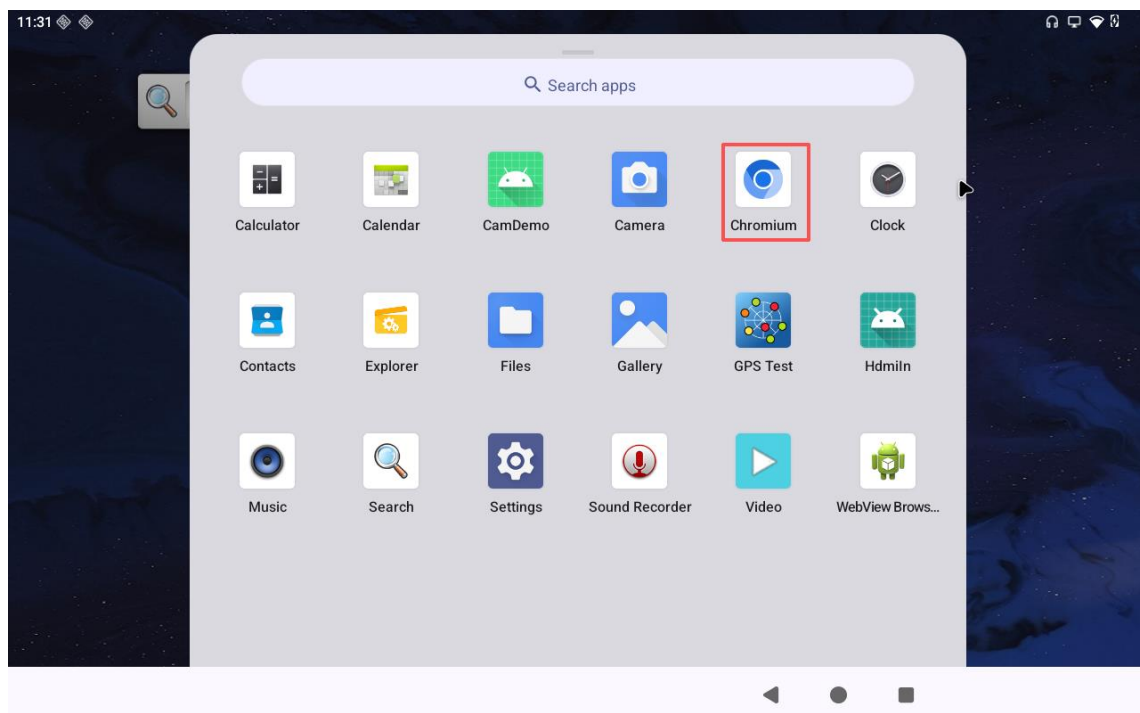
Step 4: Turn on the **Wi-Fi** switch:

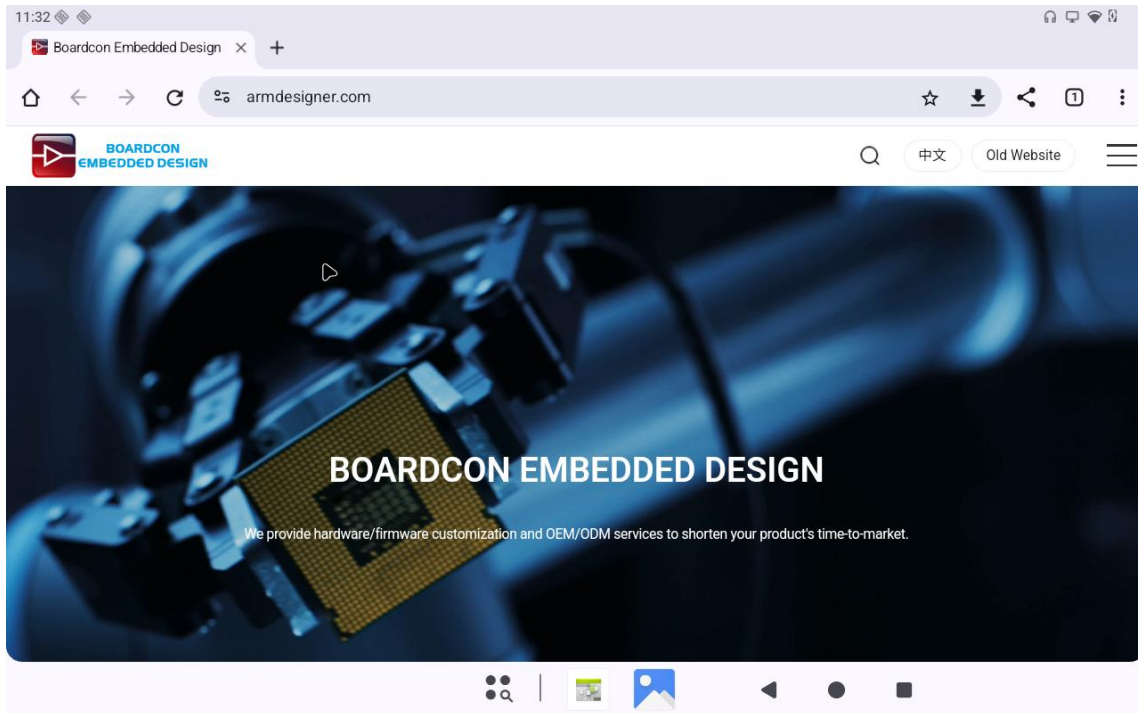


Step 5: Select an available Wi-Fi hotspot from the list and connect:



Step 6: Internet test.





Users can also choose to use the ping command to test the connectivity of the wifi, as shown below:

(1) View network interface information.

```
$ su
# ifconfig wlan0
```

```
console:/ $ ifconfig wlan0
wlan0    Link encap:Ethernet  HWaddr cc:64:1a:78:ee:6a  Driver rtl8852be
         inet addr:192.168.0.42  Bcast:192.168.0.255  Mask:255.255.255.0
         inet6 addr: fe80::f2c3:f8a4:1f47:8770/64  Scope: Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:2396 errors:0 dropped:0 overruns:0 frame:0
         TX packets:1523 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:2030800 TX bytes:203805
         Interrupt:141
```

(2) Network connection test.

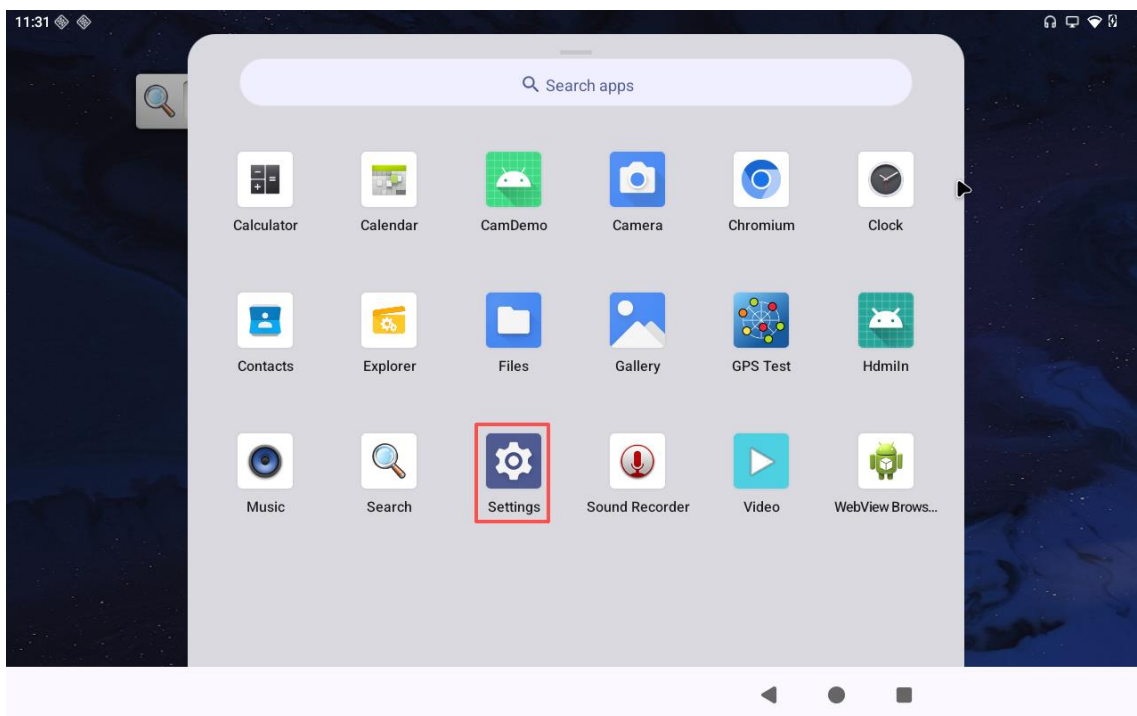
```
# ping -I wlan0 www.armdesigner.com
```



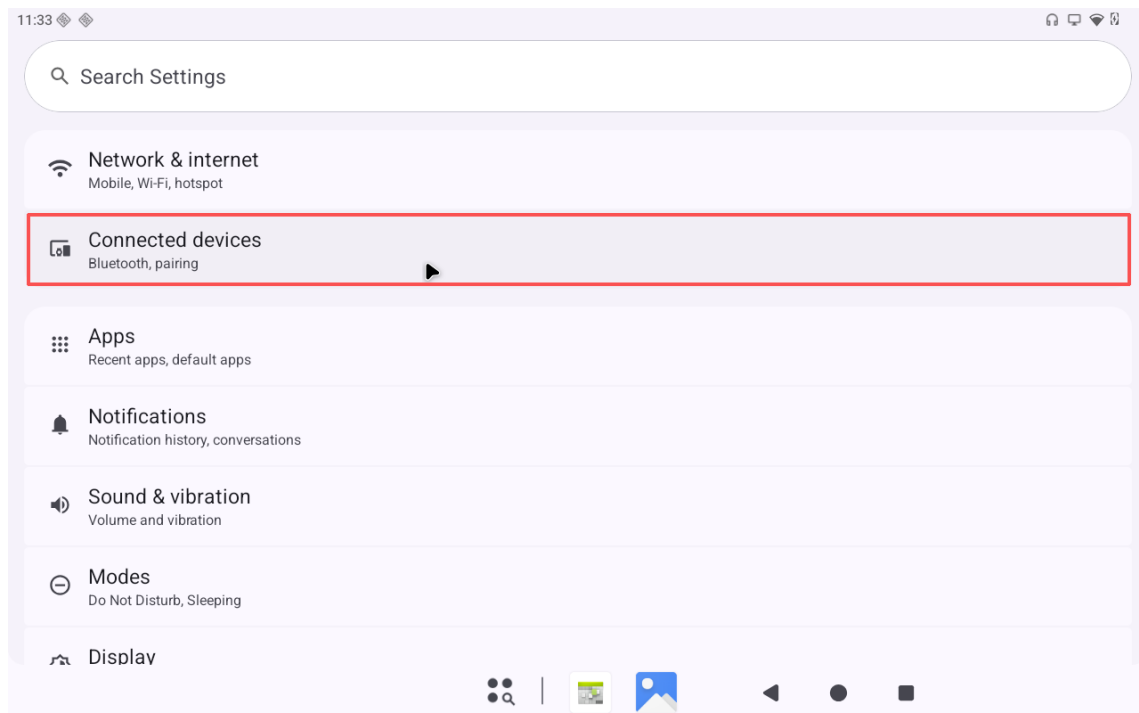
```
ssi_16_arm64:/ $ ping -I wlan0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.42 wlan0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=49 time=199 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=49 time=189 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=49 time=193 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=49 time=196 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=49 time=324 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=6 ttl=49 time=251 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 6 received, 14% packet loss, time 6007ms
rtt min/avg/max/mdev = 189.282/225.811/324.579/48.909 ms
```

6.8.2 Bluetooth test

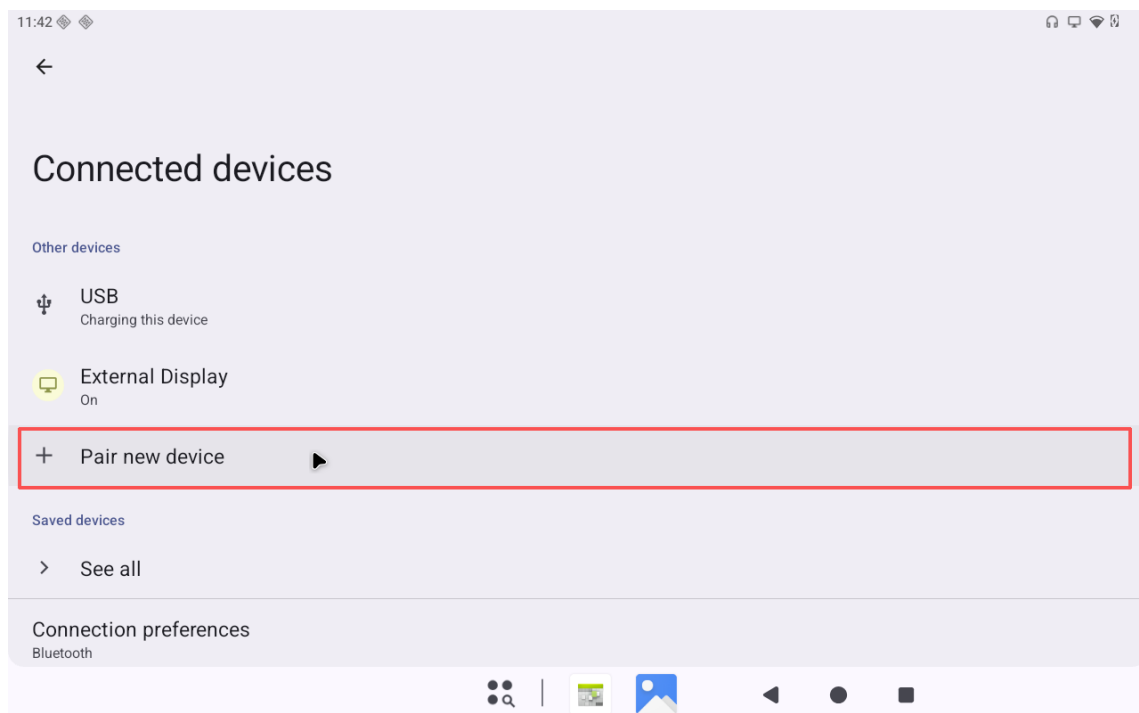
Step 1: Open the Settings app.



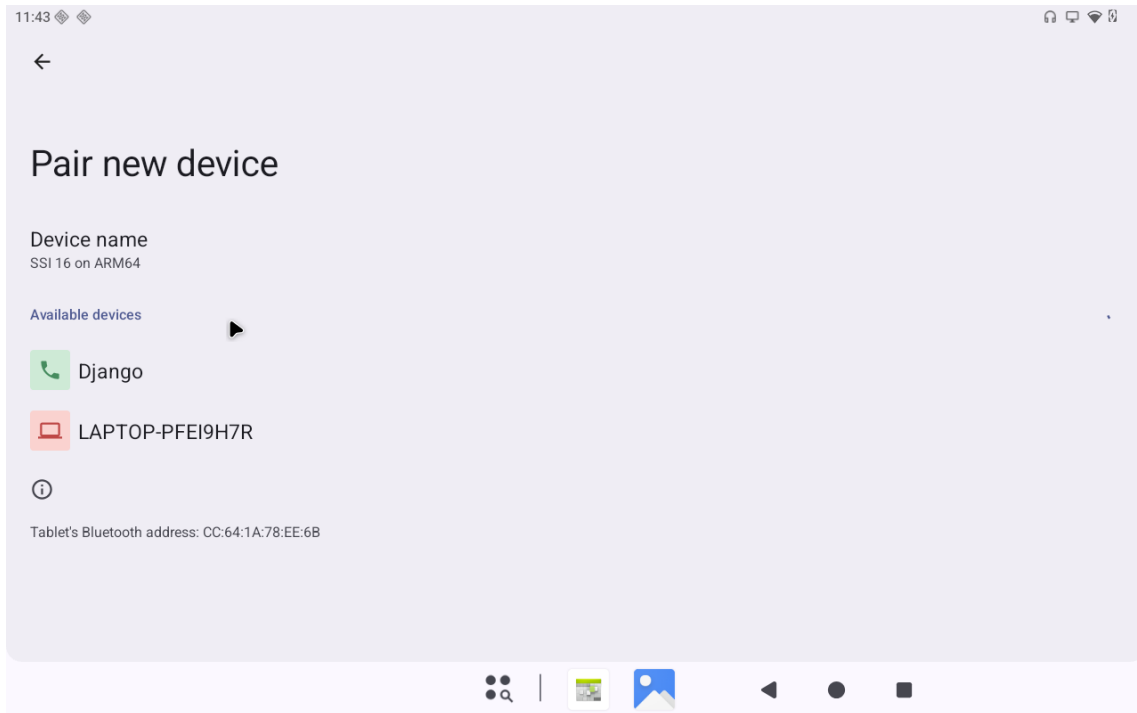
Step 2: Click on the option **Connected devices**



Step 3: Click on the option **Pair new device**.



Step 4: User can pair themselves in the Bluetooth device list.

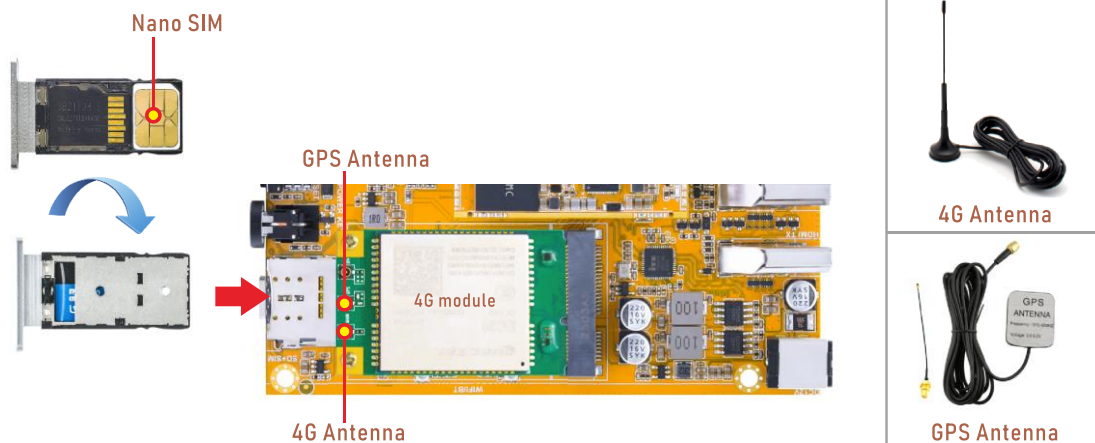


6.9 4G & GPS

Step 1: Insert 4G module to PCIe socket (4G model: EC20).

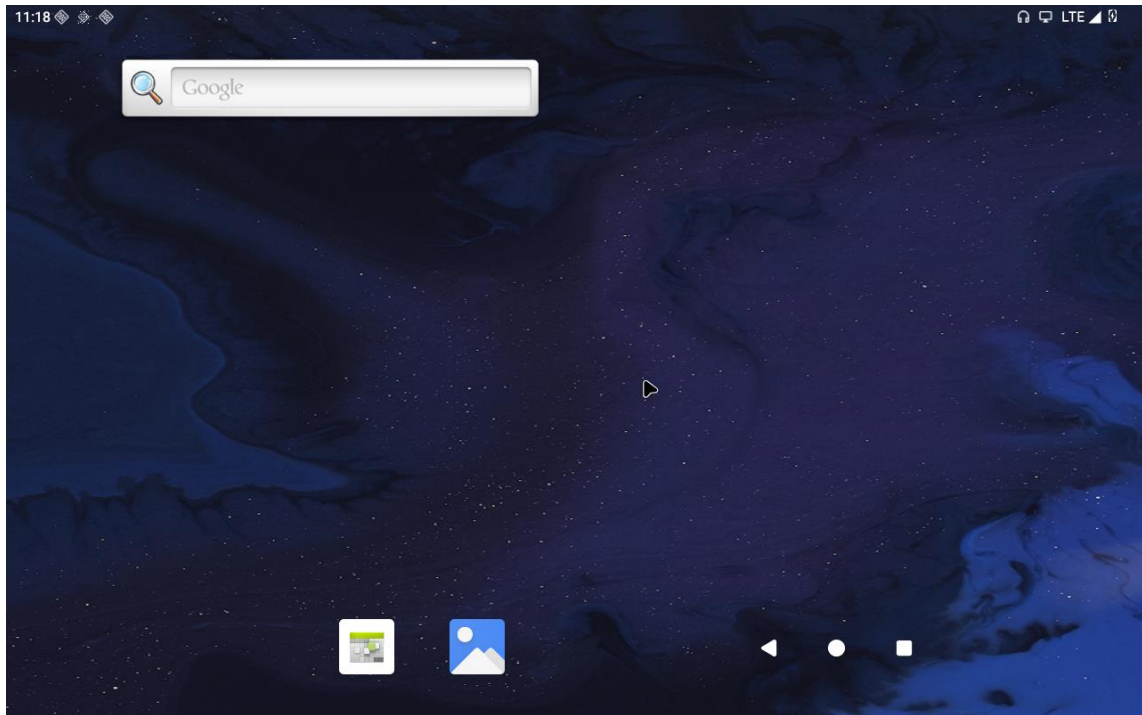
Step 2: Connect antenna and insert SIM card.

Step 3: Power on.

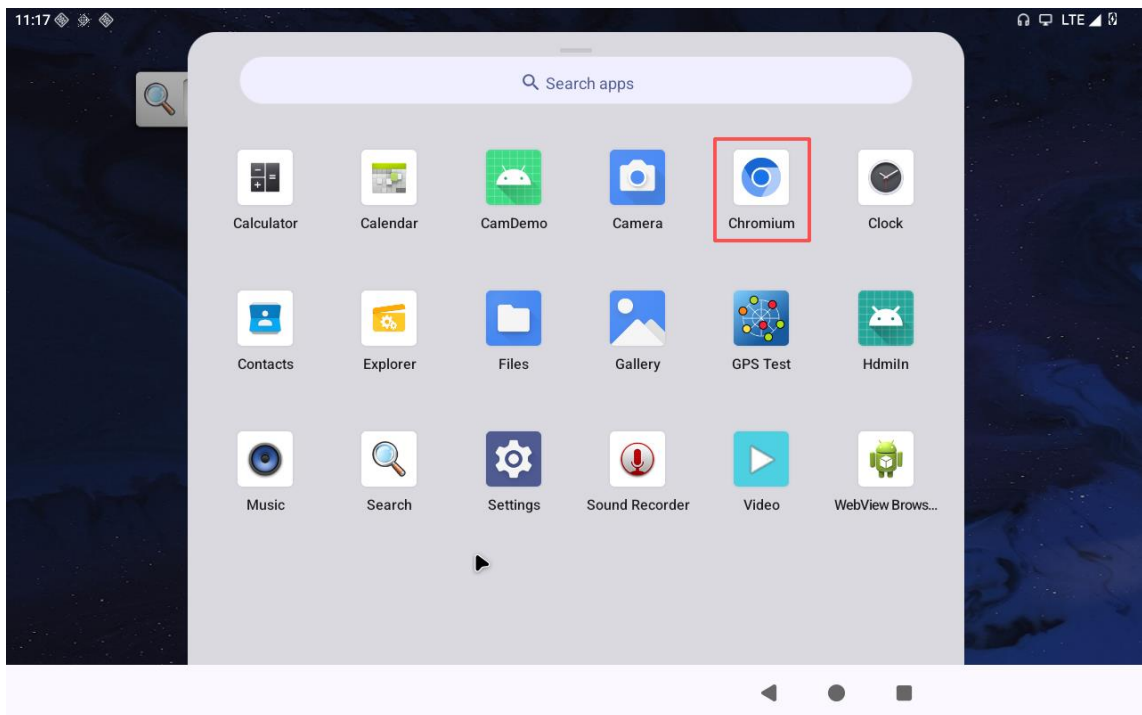


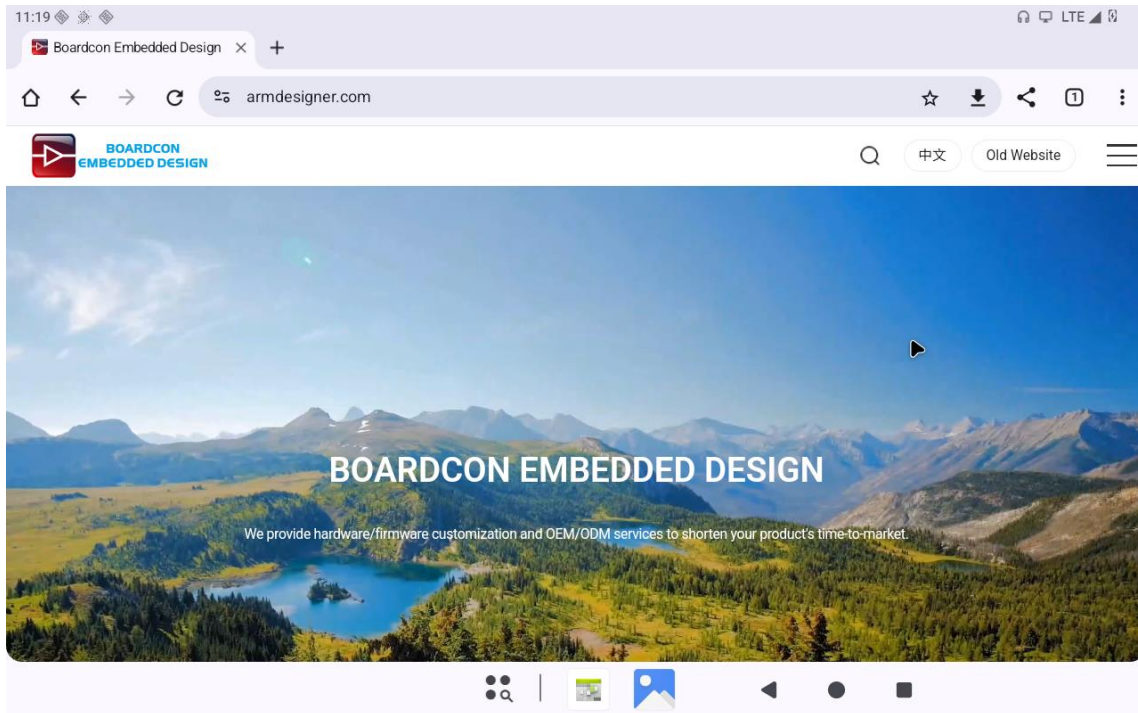
6.9.1 4G Test

After boot, the device defaults to 4G mode, and the corresponding icon will appear in the upper right corner of the desktop.



Test the 4G connection:





Users can also choose to use the ping command to test the connectivity of the 4G, as shown below:

(1) View network interface information.

```
# ifconfig
```

```
130|console:/ $ ifconfig wwan0
wwan0      Link encap:Ethernet  HWaddr 8e:fd:44:46:9d:44  Driver qmi_wwan_q
           inet addr:10.4.48.151 Mask:255.255.240.0
           inet6 addr: fe80::8cfd:44ff:fe46:9d44/64 Scope: Link
           inet6 addr: 240e:47c:3080:339:8cfd:44ff:fe46:9d44/64 Scope: Global
           inet6 addr: 240e:47c:3080:339:f040:42ec:4517:6463/64 Scope: Global
           UP RUNNING NOARP MTU:1500 Metric:1
           RX packets:753 errors:0 dropped:0 overruns:0 frame:0
           TX packets:775 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:642395 TX bytes:132950
```

(2) Network connection test.

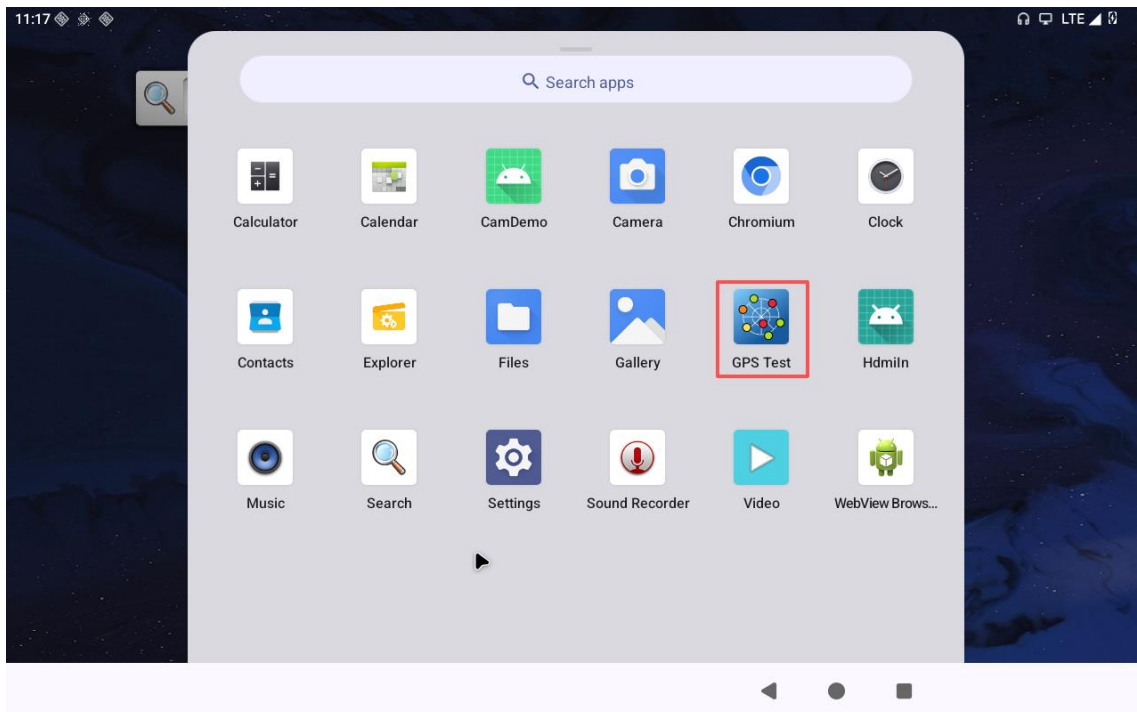
```
# su
# ping -I wwan0 www.armdesigner.com
```

```
console:/ # ping -I wwan0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 10.4.48.151 wwan0: 56(84) bytes of data.
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=1 ttl=47 time=429 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=2 ttl=47 time=319 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=3 ttl=47 time=236 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=4 ttl=47 time=220 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=5 ttl=47 time=524 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=6 ttl=47 time=476 ms
^C
--- www.armdesigner.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5064ms
rtt min/avg/max/mdev = 220.226/367.679/524.764/116.537 ms
```

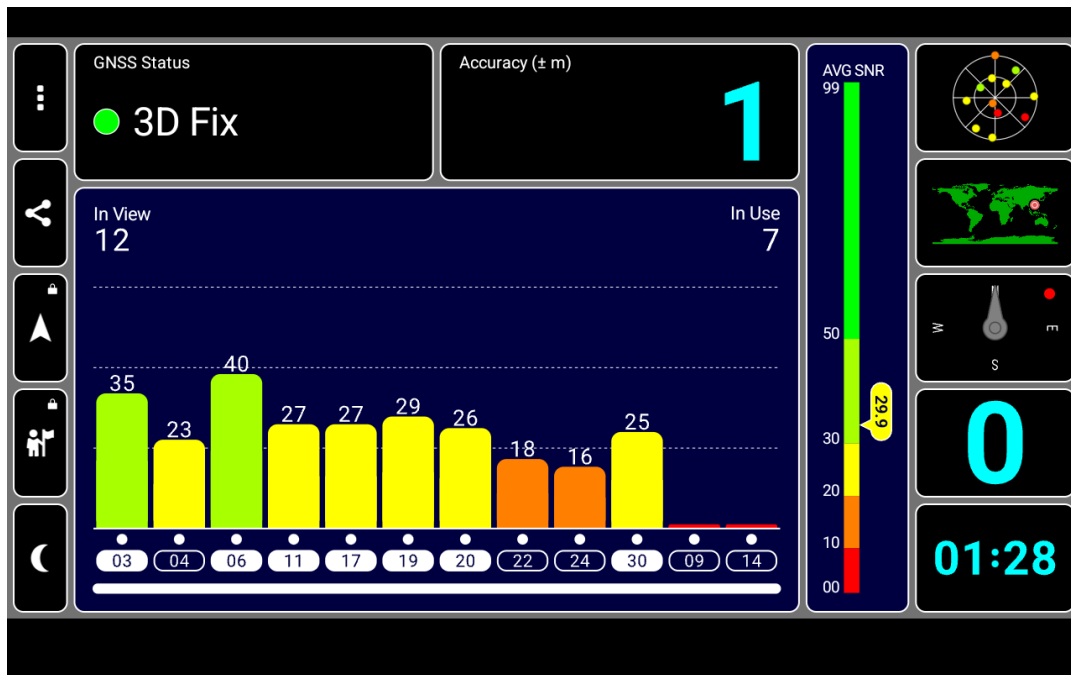
6.9.2 GPS Test

GPS positioning needs to be tested in open outdoor areas.

Step 1: Open GPS Test APP.

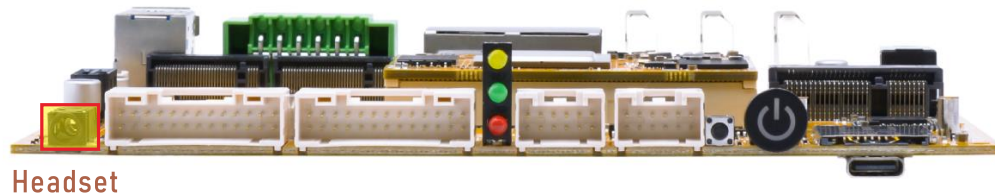


Step 2: Wait a moment to successfully obtain the location.

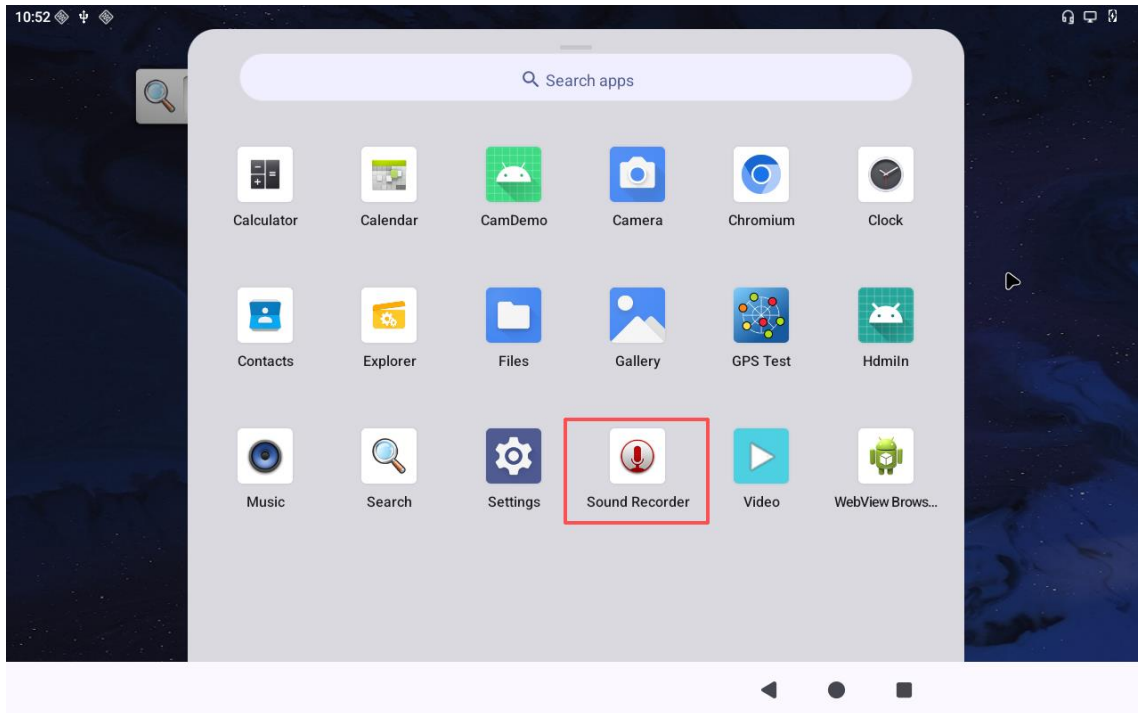


6.10 Headset

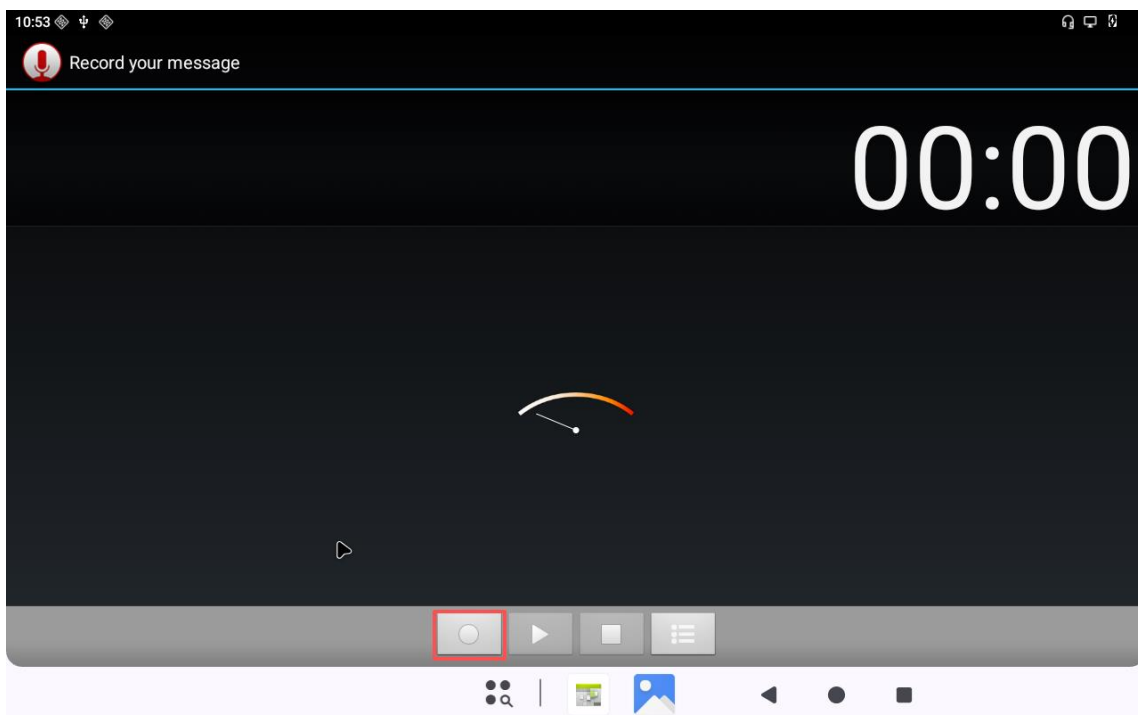
Step 1: Plug the headset into the headset jack and connect the speaker.



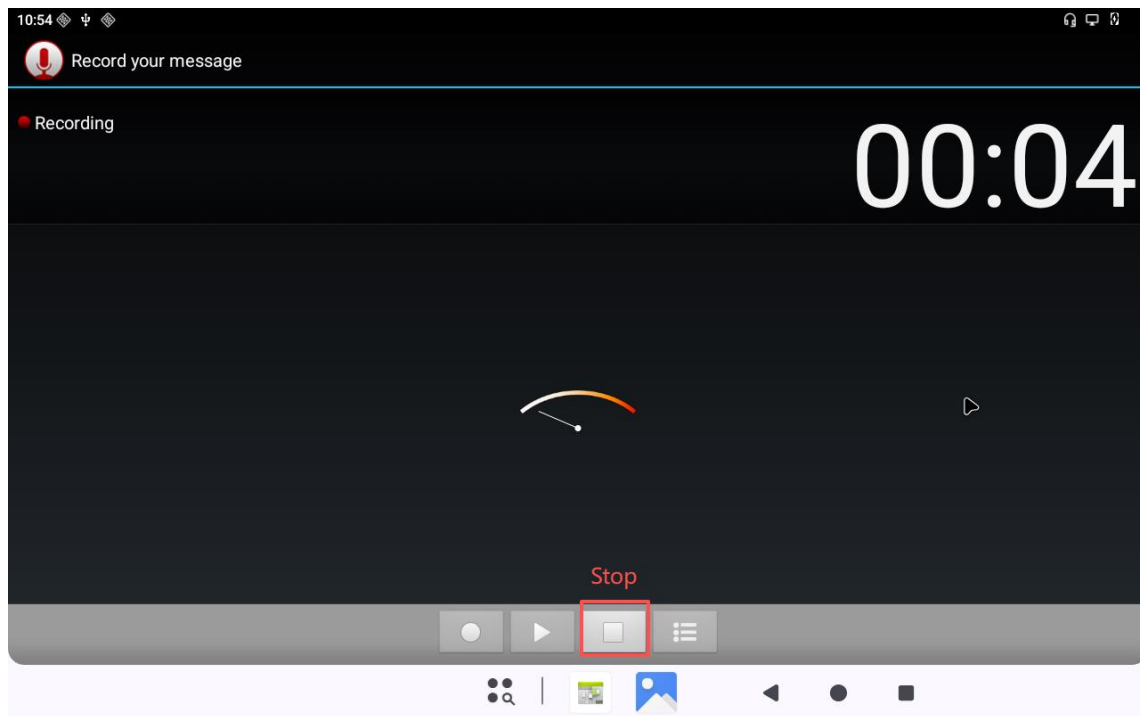
Step 2: Open the **Sound Recorder** app.



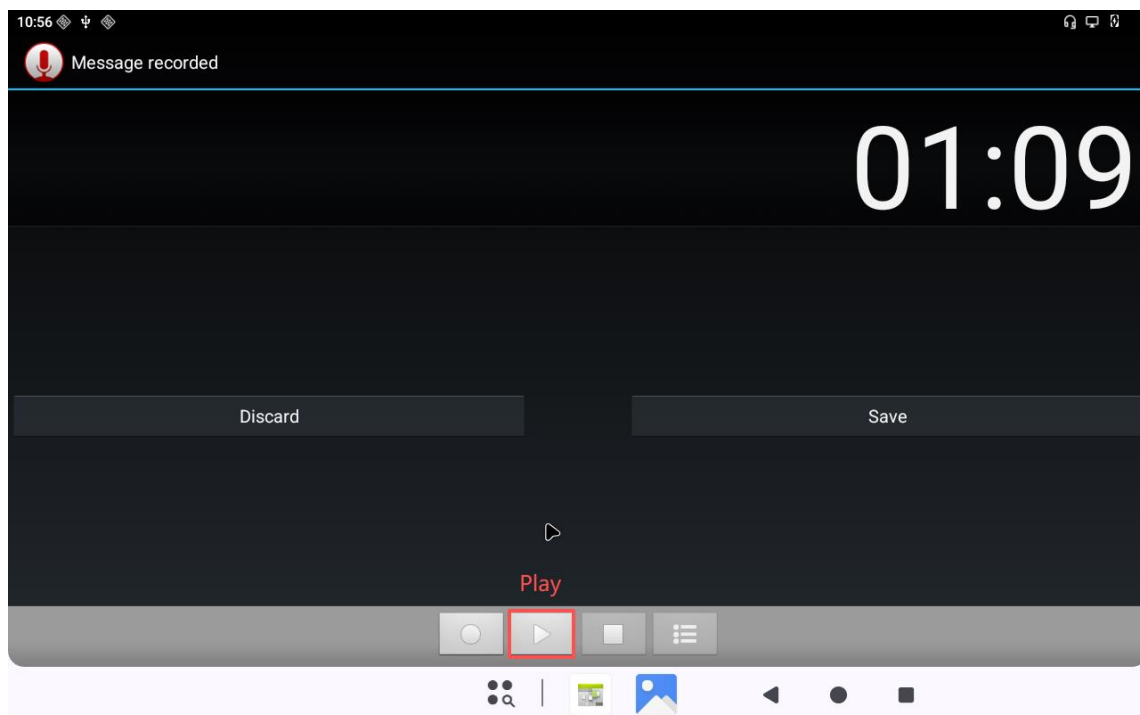
Step 3: Click the following button to start recording.



Step 4: Click the following button to stop recording.



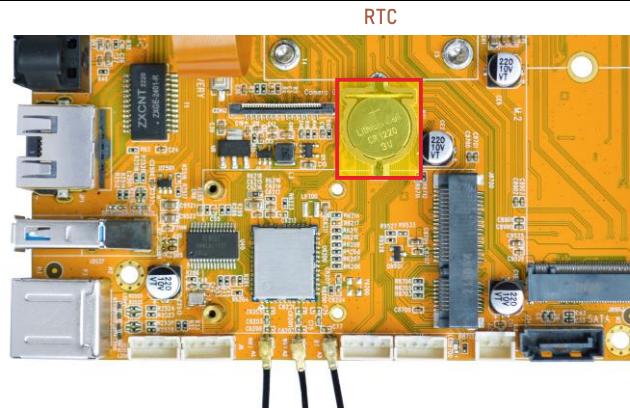
Step 5: Click the following button to play back the recording.



Note: Audio output level: Headset > HDMI > Speaker.

6.11 RTC

Step 1: Install the coin cell battery.



Step 2: Set the system time.

```
# su
# date -s "2026-12-12 9:42:00"
```

Step 3: Write the system time to the hardware clock.

```
# hwclock -w
```

Step 4: Display the current hardware clock time.

```
# hwclock
```

```
130|console:/ # date -s "2026-12-12 9:42:00"
Sat Dec 12 09:42:00 CST 2026
console:/ # hwclock -w
console:/ # hwclock
2026-12-12 01:42:08+0000
console:/ # hwclock
2026-12-12 01:42:14+0000
console:/ # hwclock
2026-12-12 01:42:19+0000
```

Step 5: Power off, after a period of time to turn on the power again, check whether the time is saved.

```
console:/ # hwclock
2026-12-12 01:43:36+0000
console:/ # hwclock
2026-12-12 01:43:40+0000
console:/ # hwclock
2026-12-12 01:43:46+0000
```

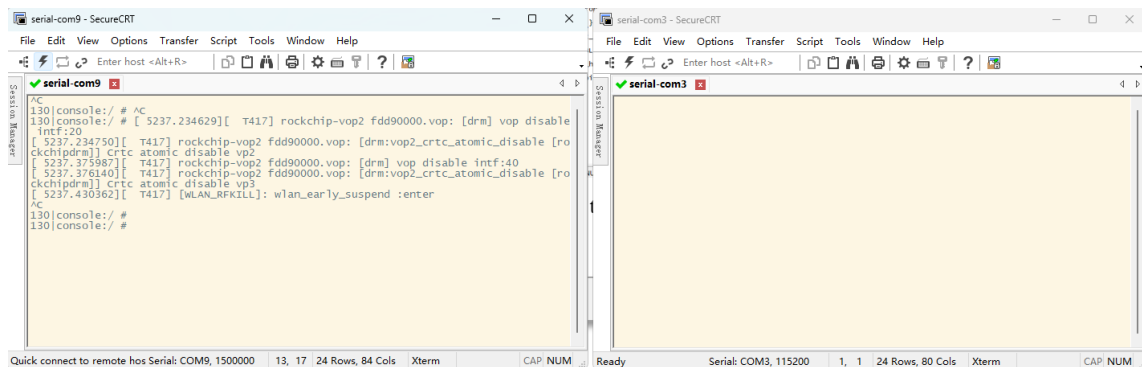
6.12 RS485



Step 1: As shown in the diagram, connect the RS485 test tool to the development board.



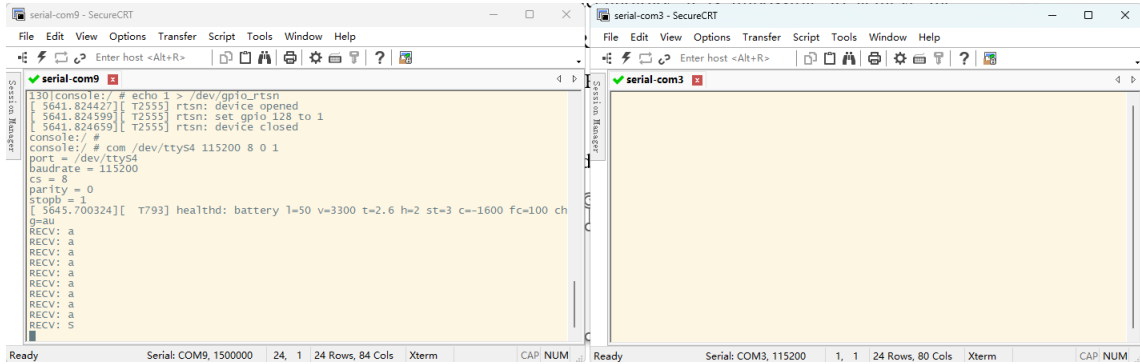
Step 2: Open the corresponding serial terminal, set the baud rate of the board to 1500000, and set the baud rate of the RS485 test tool to 115200.



Step 3: Since this image employs GKI technology, it is impossible to achieve the automatic transmission and reception of RS485 through software or hardware traffic control. The transmission and reception of RS485 can only be realized through manual control of RTSN.

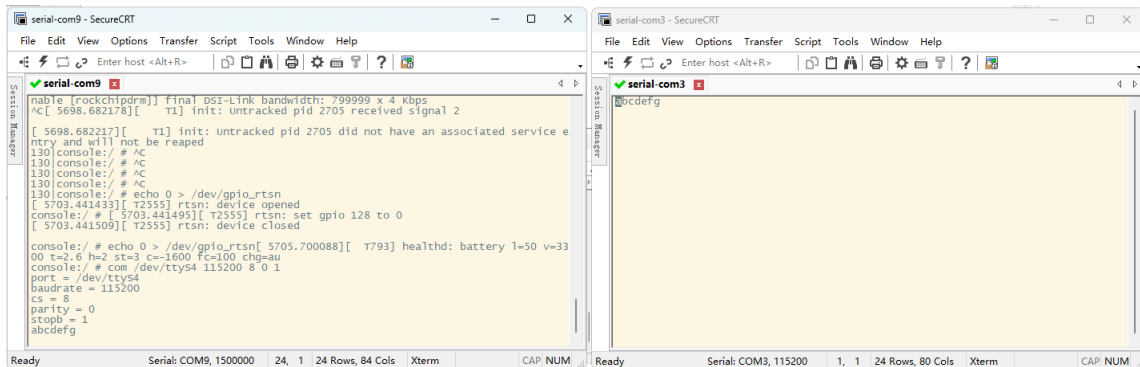
Execute the following command on the board to test the RS485 reception functionality.

```
# echo 1 > /dev/gpio_rtsn
# com /dev/ttyS4 115200 8 0 1
```



Execute the following command on the board to test the RS485 transmission functionality.

```
# echo 0 > /dev/gpio_rtsn
# com /dev/ttyS4 115200 8 0 1
```



6.13 CAN

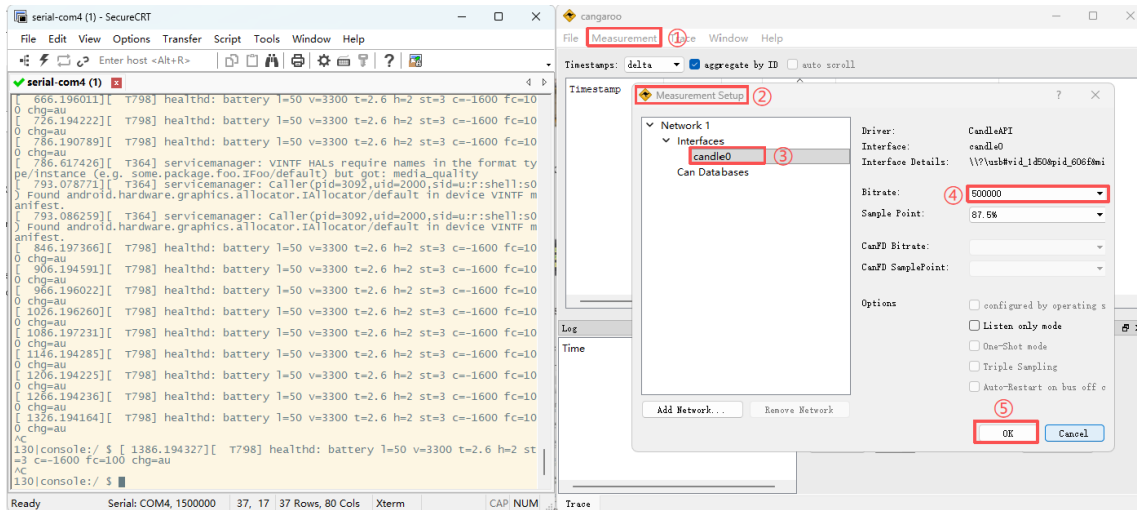


CAN

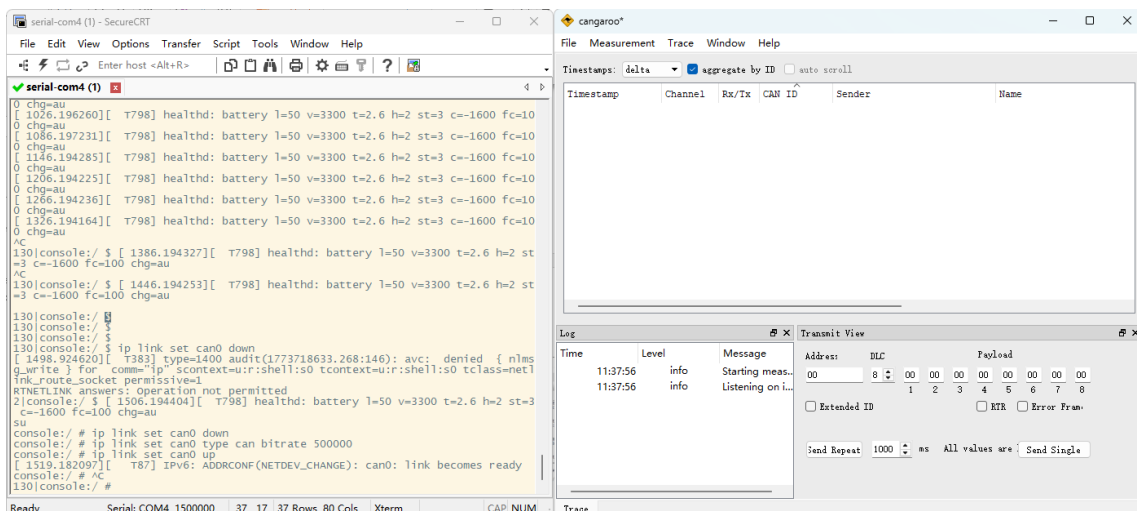
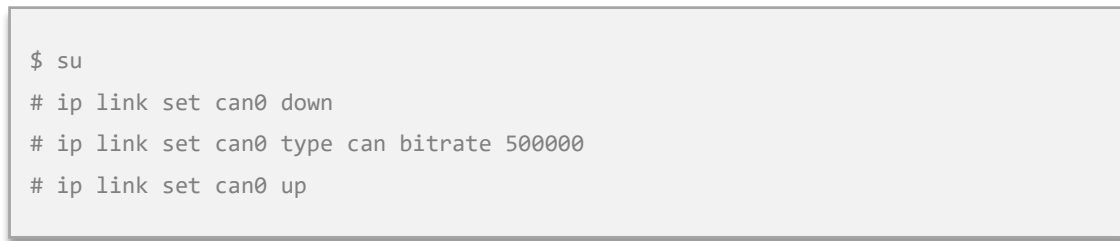
Step 1: Connect the CAN test tool to the board as shown in the diagram below.



Step 2: Open the CAN test software and set the baud rate to 500000.

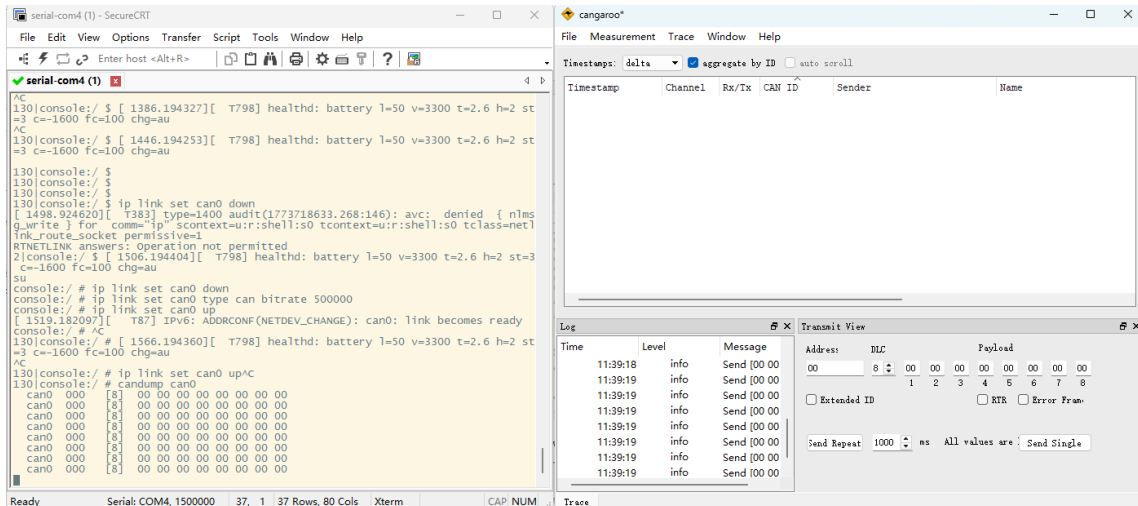


Step 3: Configure and activate the CAN network, setting the bitrate to 500000.



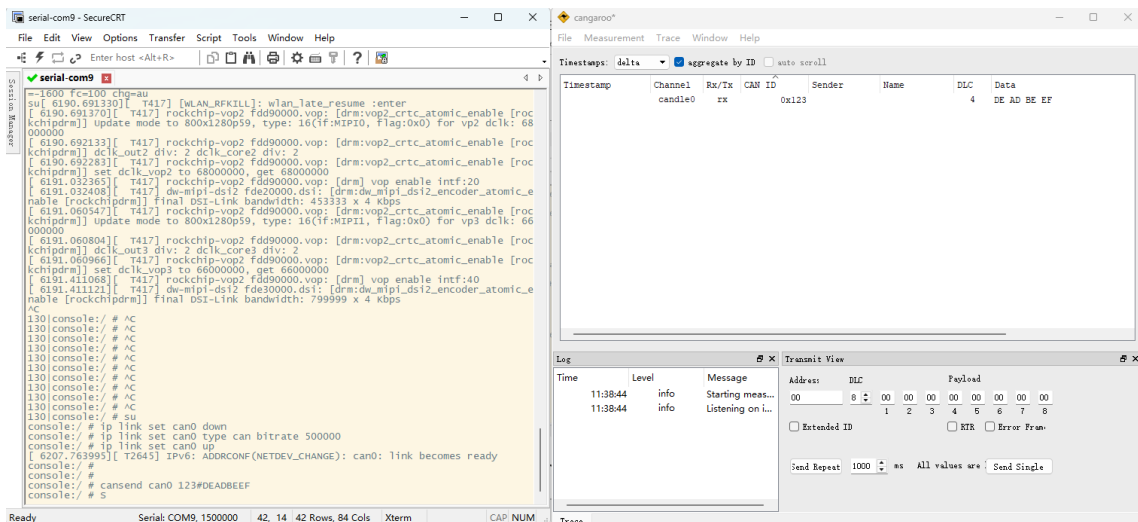
Step 4: Configure CAN as the receiver.





Step 5: Configure CAN as the sender.

```
# cansend can0 123#DEADBEEF
```



6.14 RS232

Step 1: Short circuit RX and TX pins of UART.



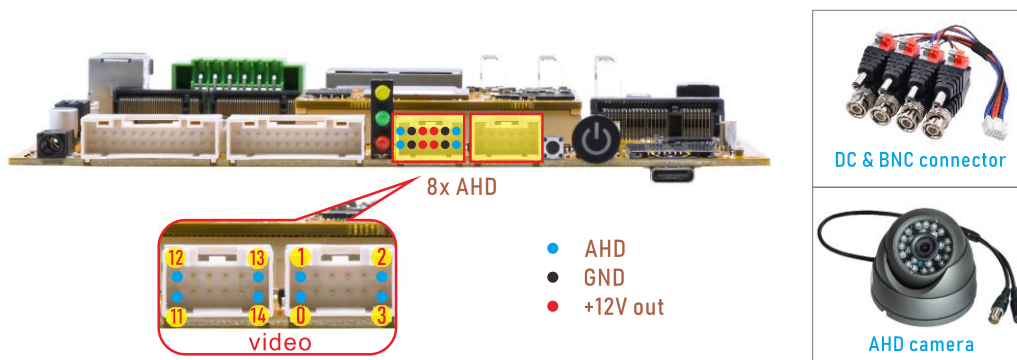
RS232

Step 2: RS232 test.

```
$ com /dev/ttyS7 115200 8 0 1
```

```
130|console:/ $ com /dev/ttyS7 115200 8 0 1
port = /dev/ttyS7
baudrate = 115200
cs = 8
parity = 0
stopb = 1
123456789
RECV: 123456789
asdffjhghkhl;l;
RECV: asdffjhghkhl;l;
```

6.15 Camera



Note: This device only provides a video capture interface. Preview functionality must be implemented by the customer through custom application development based on their specific requirements.

- Single-camera preview, using video11 as an example.

```
# v4l2-ctl --verbose -d /dev/video11 --set-fmt-
video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --set-
selection=target=crop,flags=0,top=0,left=0,width=1920,height=1080 --stream-
to=/mnt/out0.yuv
```

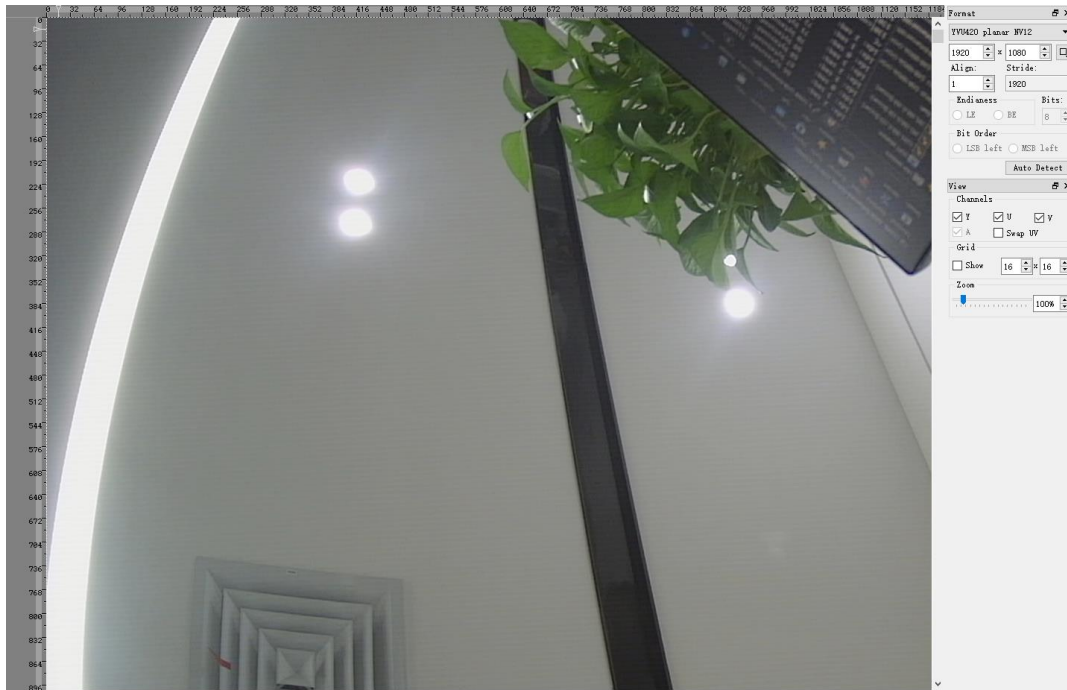



```
console:/ # su
tream-mmap=4 --set-selection=target=crop,flags=0,top=0,left=0,width=1920,height=1080 --stream-
to=/mnt/out0.yuv <
VIDIOC_QUERYCAP: ok
VIDIOC_G_FMT: ok
VIDIOC_S_FMT: ok
Format Video Capture Multiplanar:
  Width/Height      : 1920/1080
  Pixel Format       : 'NV12'
  Field              : None
  Number of planes   : 1
  Flags              :
  Colourspace        : Default
  Transfer Function   : Default
  YCbCr Encoding     : Default
  Quantization       : Default
  Plane 0            :
    Bytes per Line   : 1920
    Size Image       : 3110400
VIDIOC_G_SELECTION: ok
VIDIOC_S_SELECTION: ok
VIDIOC_REQBUFS: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_STREAMON: ok
idx: 0 seq:      0 bytesused: 3110400 ts: 124.590904
idx: 1 seq:      1 bytesused: 3110400 ts: 124.630922 delta: 40.018 ms
idx: 2 seq:      2 bytesused: 3110400 ts: 124.670946 delta: 40.024 ms
idx: 3 seq:      3 bytesused: 3110400 ts: 124.710935 delta: 39.989 ms
idx: 0 seq:      4 bytesused: 3110400 ts: 124.750934 delta: 39.999 ms fps: 25.00
idx: 1 seq:      5 bytesused: 3110400 ts: 124.790929 delta: 39.995 ms fps: 25.00
idx: 2 seq:      6 bytesused: 3110400 ts: 124.830877 delta: 39.948 ms fps: 25.00
idx: 3 seq:      7 bytesused: 3110400 ts: 124.870926 delta: 40.049 ms fps: 25.00
idx: 0 seq:      8 bytesused: 3110400 ts: 124.910923 delta: 39.997 ms fps: 25.00
idx: 1 seq:      9 bytesused: 3110400 ts: 124.950920 delta: 39.997 ms fps: 25.00
idx: 2 seq:     10 bytesused: 3110400 ts: 124.990919 delta: 39.999 ms fps: 25.00
idx: 3 seq:     11 bytesused: 3110400 ts: 125.030916 delta: 39.997 ms fps: 25.00
idx: 0 seq:     12 bytesused: 3110400 ts: 125.070915 delta: 39.999 ms fps: 25.00
idx: 1 seq:     13 bytesused: 3110400 ts: 125.110915 delta: 40.000 ms fps: 25.00
idx: 2 seq:     14 bytesused: 3110400 ts: 125.150911 delta: 39.996 ms fps: 25.00
idx: 3 seq:     15 bytesused: 3110400 ts: 125.190838 delta: 39.927 ms fps: 25.00
idx: 0 seq:     16 bytesused: 3110400 ts: 125.230901 delta: 40.063 ms fps: 25.00
idx: 1 seq:     17 bytesused: 3110400 ts: 125.270899 delta: 39.998 ms fps: 25.00
idx: 2 seq:     18 bytesused: 3110400 ts: 125.310896 delta: 39.997 ms fps: 25.00
idx: 3 seq:     19 bytesused: 3110400 ts: 125.350895 delta: 39.999 ms fps: 25.00
idx: 0 seq:     20 bytesused: 3110400 ts: 125.390890 delta: 39.995 ms fps: 25.00
idx: 1 seq:     21 bytesused: 3110400 ts: 125.430888 delta: 39.998 ms fps: 25.00
idx: 2 seq:     22 bytesused: 3110400 ts: 125.470793 delta: 39.905 ms fps: 25.00
idx: 3 seq:     23 bytesused: 3110400 ts: 125.510882 delta: 40.089 ms fps: 25.00
idx: 0 seq:     24 bytesused: 3110400 ts: 125.550882 delta: 40.000 ms fps: 25.00
idx: 1 seq:     25 bytesused: 3110400 ts: 125.590879 delta: 39.997 ms fps: 25.00
idx: 2 seq:     26 bytesused: 3110400 ts: 125.630881 delta: 40.002 ms fps: 25.00
idx: 3 seq:     27 bytesused: 3110400 ts: 125.670878 delta: 39.997 ms fps: 25.00
^C
```

The file will save as under the current directory.

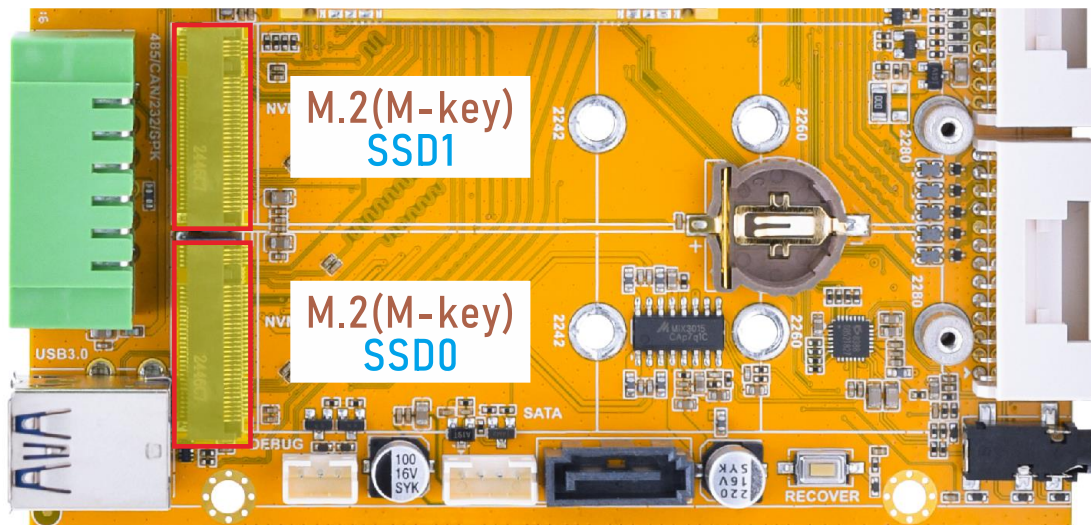
```
console:/ # ls /mnt/
androidwritable  expand      media_rw  out1_0.yuv  runtime  secure
appfuse          installer  obb       pass_through  scratch  user
asec             iso        out0.yuv  product     sdcard   vendor
```

Copy the file to PC then play it by **7yuv**.



6.16 PCIe M.2 SSD

Step 1: Insert the M.2 SSD into the M.2 card slot, then power on the device.



Step 2: The system will automatically detect and mount the SSD, and it can be accessed directly through the **Explorer** app.

