SBC3588 Android16 SSI User Manual

V1.0

Boardcon Embedded Design

Overview

The content of this document is intended solely for the SBC3588 development board, aiming to help users quickly understand, apply, and test the SBC3588 development board.

System Support

Development Board	Android16
SBC3588 V2 CM3588 V2.0	Y

Revision History

Version	Date	Author	Revision History
V1.0	2026-03-19	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

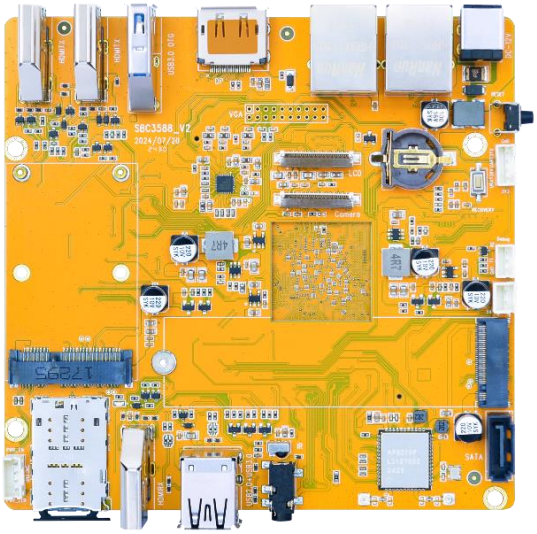
Tel: +86-755-26481393 | +86-755-27571591

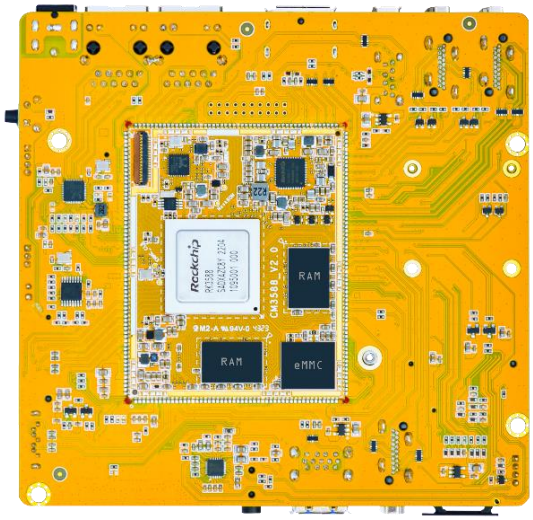
Contents

1.Introduction

1.1 Overview

The SBC3588 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.





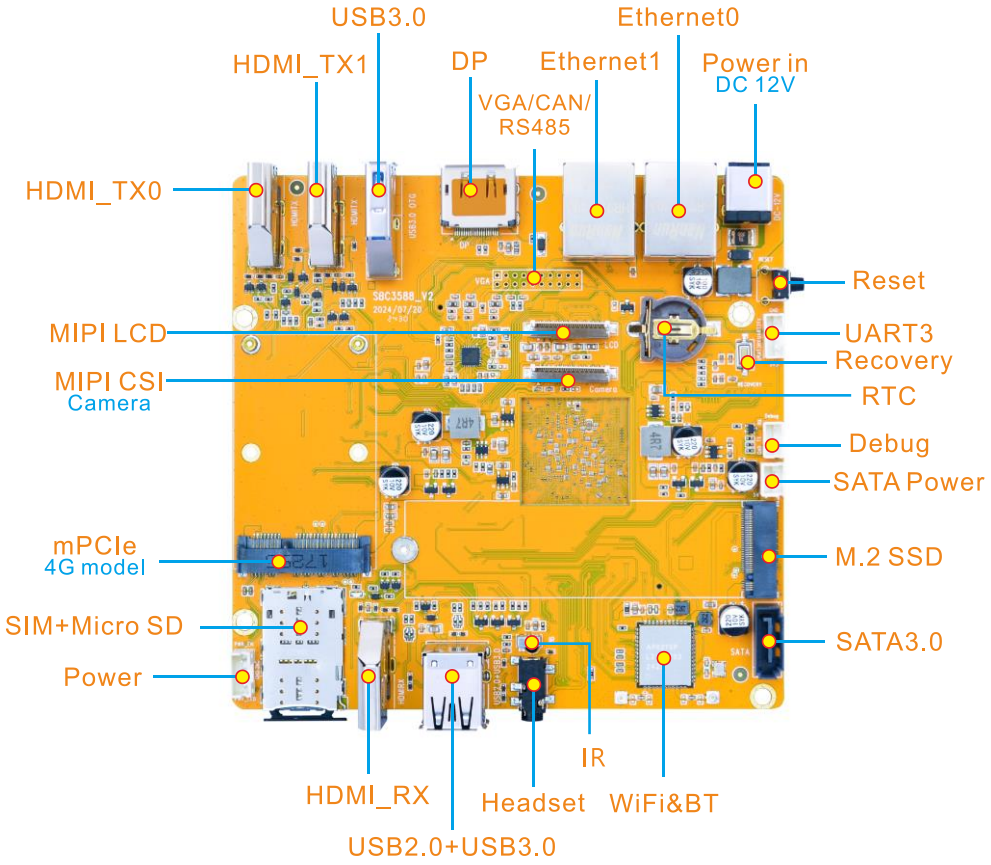
1.2 Product Parameters

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/AVS2 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 1x 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 TX 1.4a, up to 7680x4320@30fps • Support 1x MIPI DSI, up to 4K@60fps • Support 1x VGA
Audio		• Support 2x HDMI TX audio output • Support 1x DP audio output • Support 1x Headset output/input • Support 1x HDMI RX audio input

USB	• Support 2xUSB3.0 (1x OTG) • Support 1x USB2.0
Network	• Support 2x Gigabit Ethernet • Support 1x WIFI/BT module • Support 1x 4G module
Camera	• Support 1x MIPI Camera
Peripheral communication	• Support 1xCAN • Support 1xRS485 • Support 1xRS232
Other parameters	Support 1xDebug, 1xRTC, 1xIR
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	67mm x 53mm
Motherboard dimensions	125mm x 121.7mm

1.3 Hardware Interface Introduction



Interface parameters	
Power in DC 12V	12V DC power input interface
Ethernet0	Gigabit Ethernet0 RJ45 interface
Ethernet1	Gigabit Ethernet1 RJ45 interface
VGA/CAN/RS485	VGA display / CAN / RS485 interface
DP	DP display interface
USB3.0	USB3.0 OTG interface
HDMI_TX1	HDMI TX1 interface
HDMI_TX0	HDMI TX0 interface
MIPI LCD	MIPI display interface
MIPI CSI Camera	MIPI camera interface

mPCIe 4G module	4G model interface
SIM + Mirco SD	All-in-one card holder
Power	Sleep/Wake GPIO control interface
HDMI_RX	HDMI RX interface
USB2.0 + USB3.0	Dual-layer USB HOST interface
Headset	Headset output/input interface
IR	Infrared receiver
WIFI&BT	WIFI&Bluetooth module
SATA	SATA3.0 interface
M.2 SSD	PCIe M.2 SSD interface
SATA Power	SATA Power interface, 5V
Debug	Debug the serial port
RTC	RTC coin cell connector
Recovery	Recovery key
UART3	UART3 serial interface
Reset	Reset key

2.Install Drivers and Tool

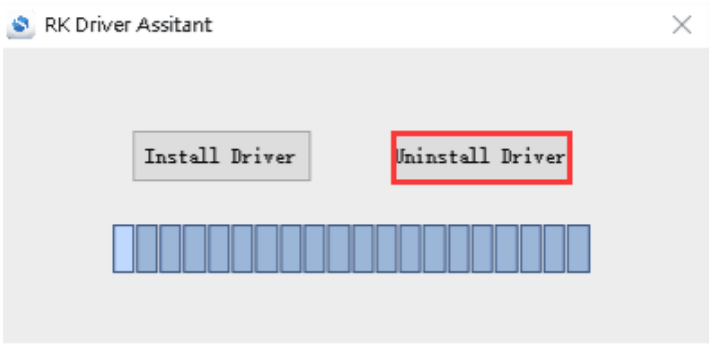
To download firmware and debug in the terminal, the following drivers and software need to be installed (for Windows computers):

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

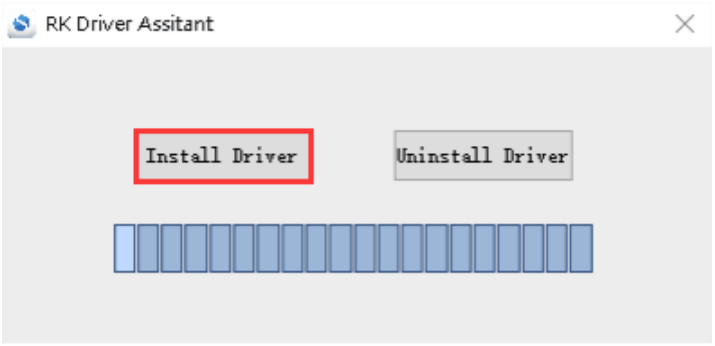
2.1 Install RK Driver Assitant

Step 1: Open *DriverAssitant_v5.1.1/DriverInstall.exe*.

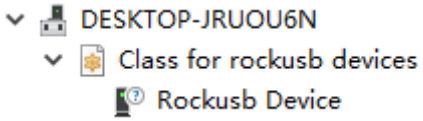
Step 2: To avoid driver conflicts, click **"Uninstall Driver"** to uninstall the driver.



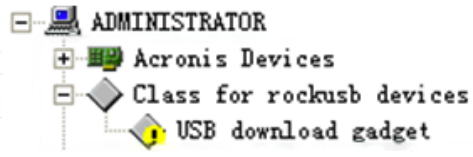
Step 3: Click button **"Install Driver"** to install.



Step 4: After the installation is complete, connect the board and PC with Type_A 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.

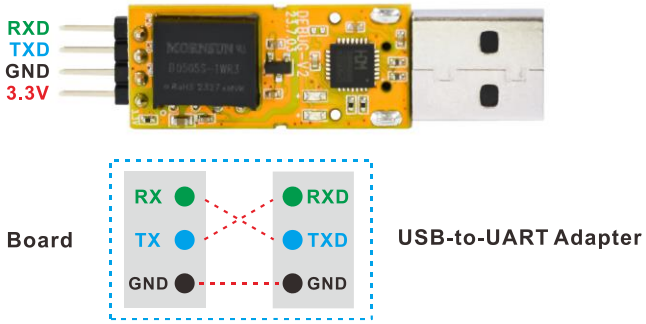


Step 6: The WINDOW will pop up found New Hardware Wizard dialog box, choose to install from the specified location, and then select: *DriverAssitant_v5.1.1/ADBDriver.*

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



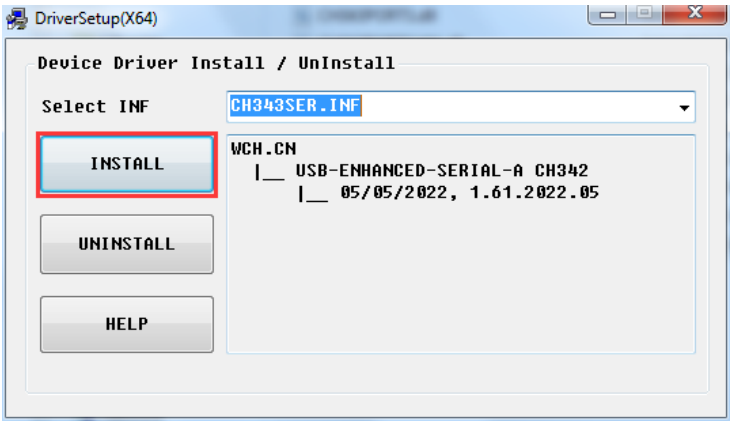
Pin	Connection Description
RXD	Receive, connect to TX pin of the board.
TXD	Transmit, connect to RX pin of the board.
GND	Ground, connect to GND pin of the board.
3V3	No need to connect.

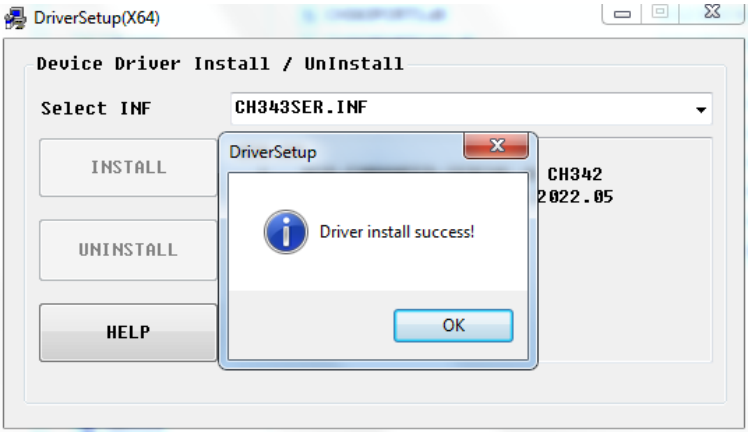
2.2.2 Install Driver

Step 1: Plug the CH9102X Module to the PC

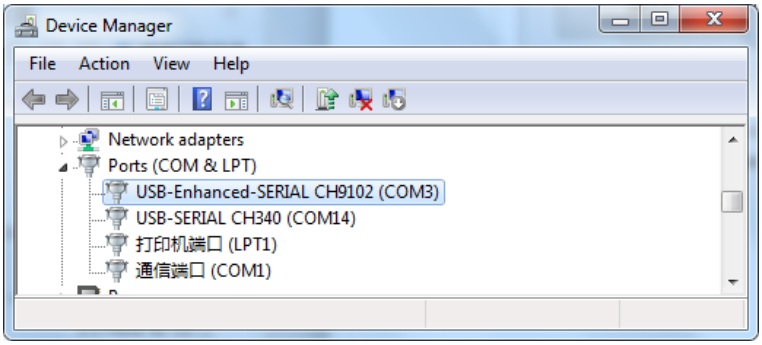
Step 2: Unzip *CH343SER.ZIP* on Windows.

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.

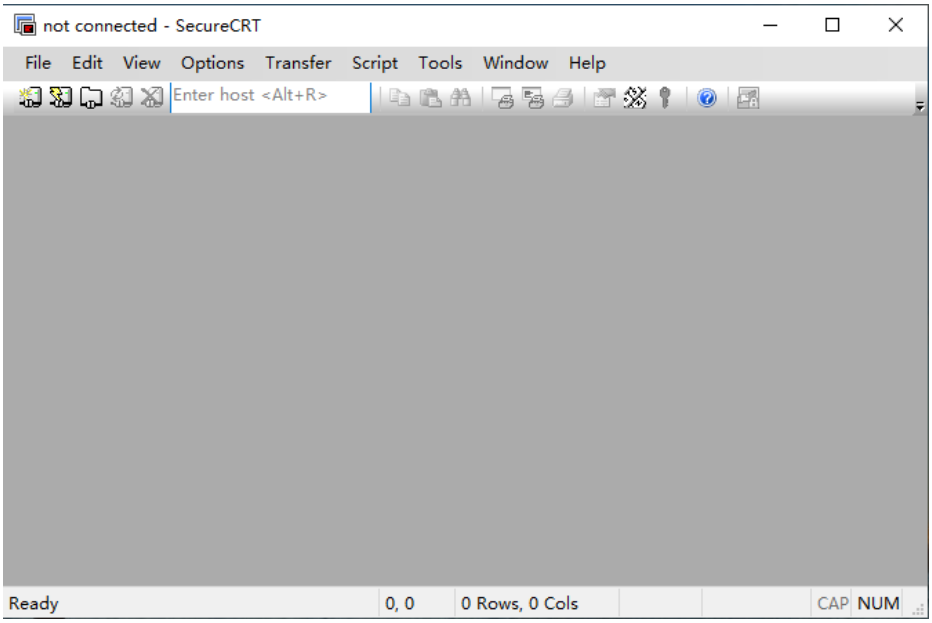


2.3 Install Serial Terminal Tool

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

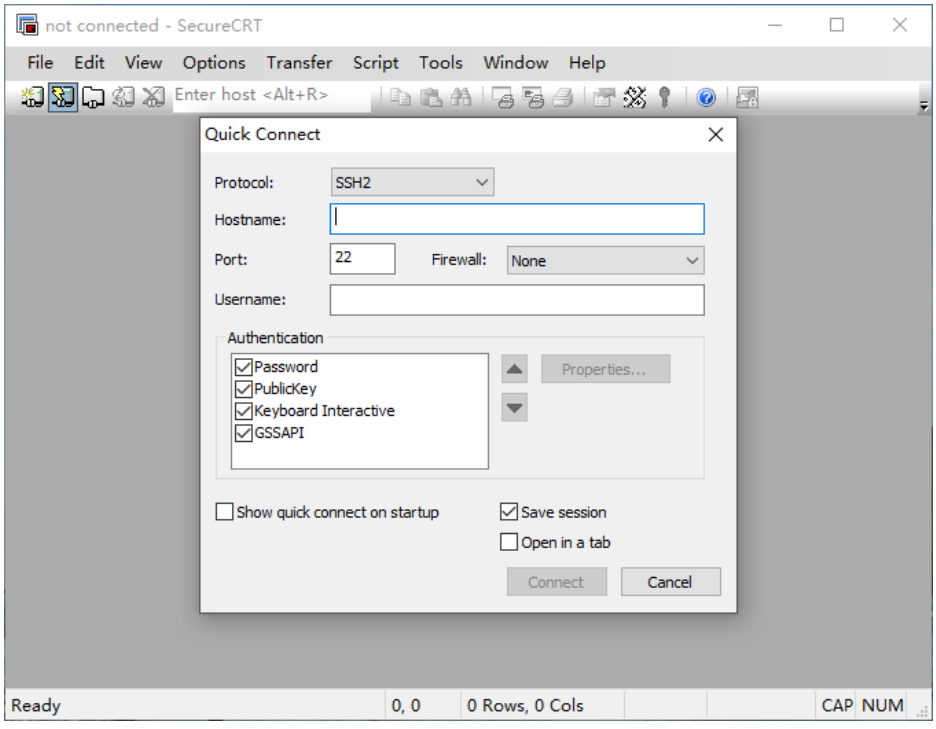
Step 1: Unzip *Platform/SecureCRT.rar* on PC.

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

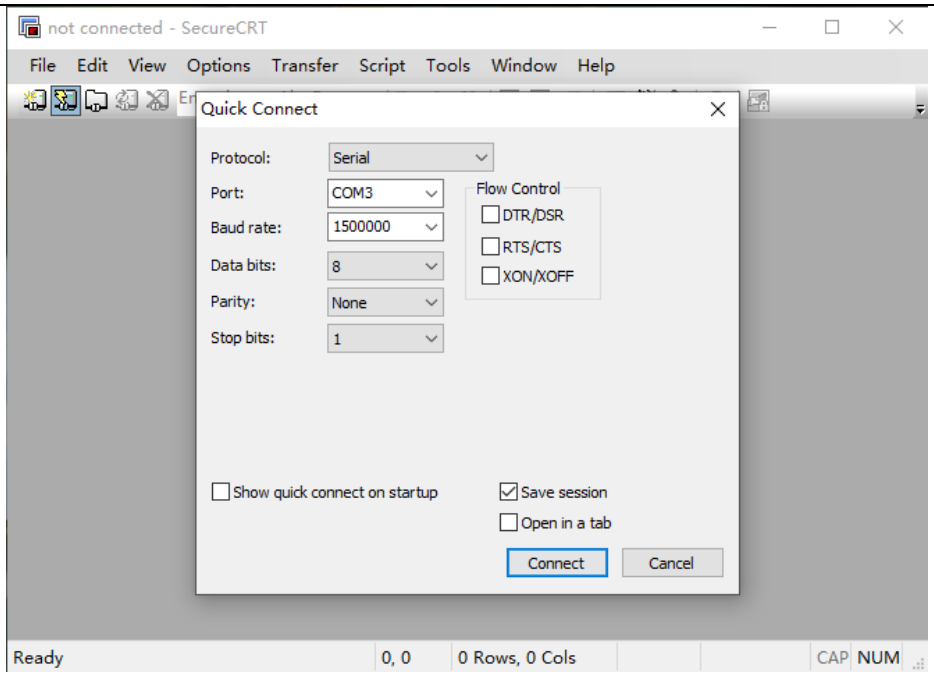


Step 3: Confirm that the CH9102X driver has been installed and the CH9102X module is connecting to the PC.

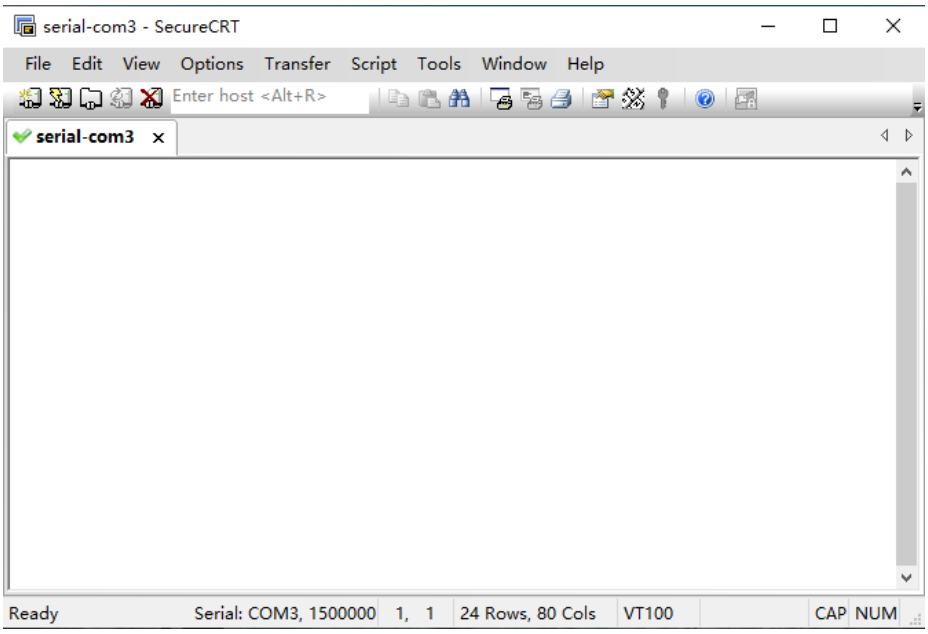
Step 4: Click the **"Quick Connect"** button to go to the Quick Connect configuration screen.



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

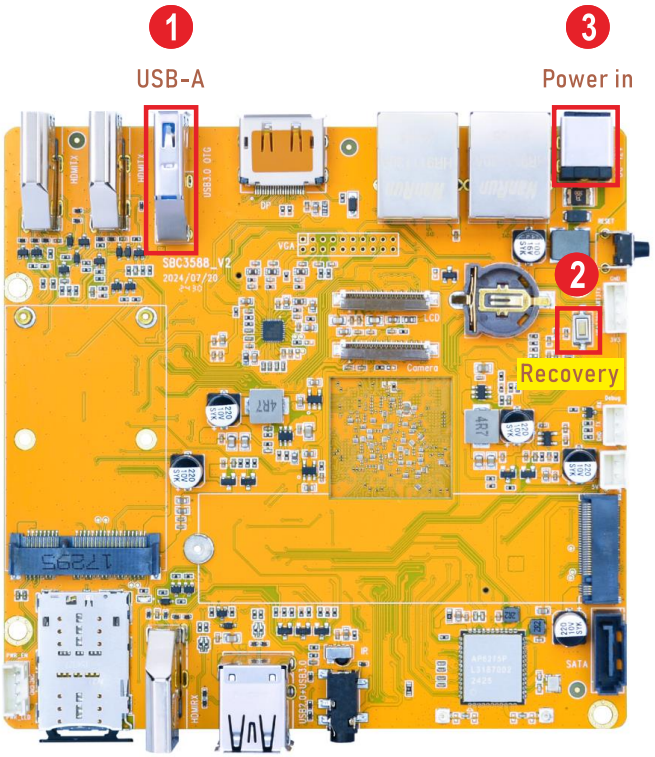
3.1.1.1 Hardware

Step 1: Disconnect the power adapter.

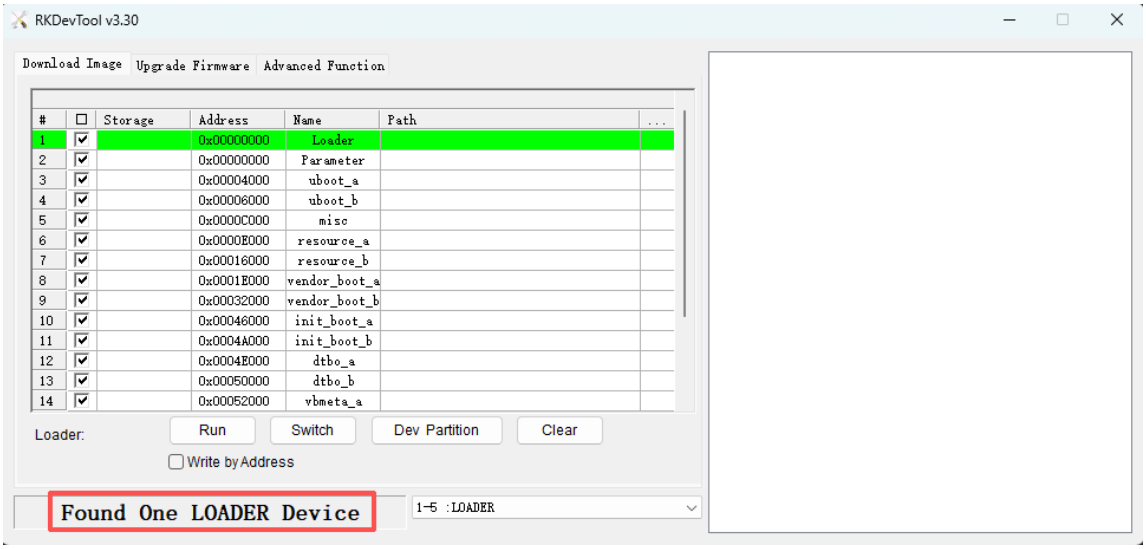
Step 2: Connect one end of the Type_A cable to the host and the other end to the development board.

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 Devic"**.



3.1.1.2 Software

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

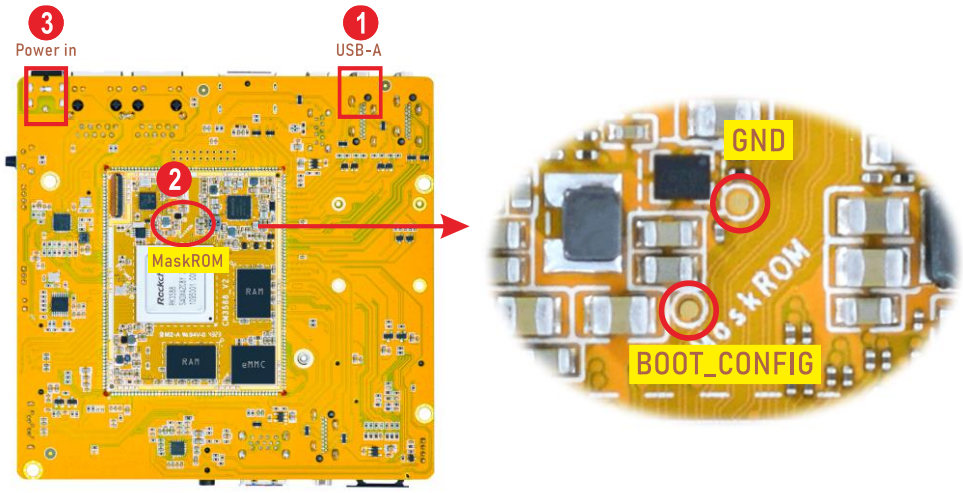
```
# reboot loader
```

3.1.2 How to Enter MaskRom Mode

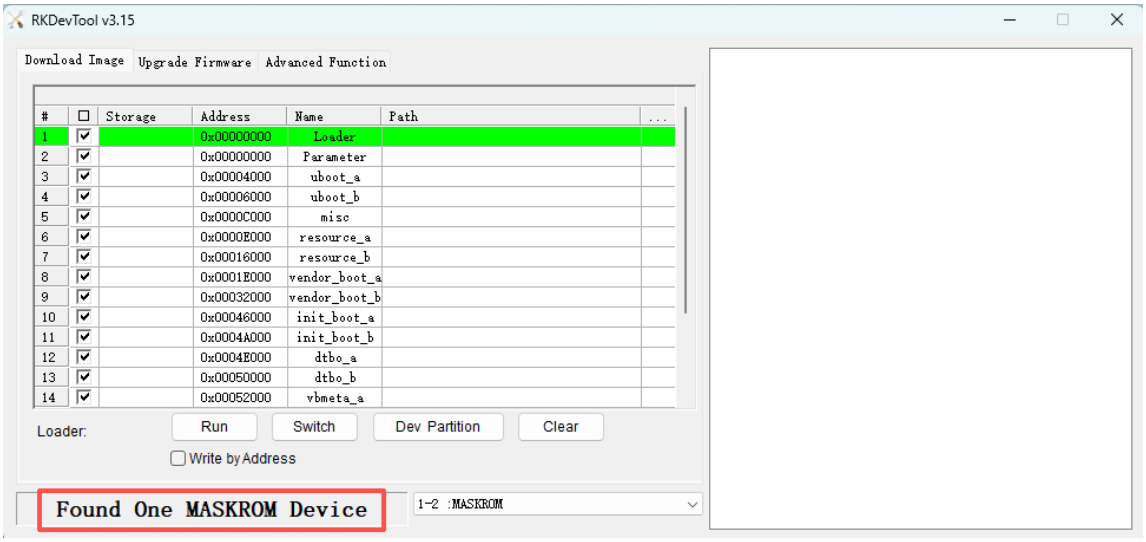
Step 1: Disconnect the power adapter.

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

Step 3: Use tweezers to short the two test points on the CM3588.



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



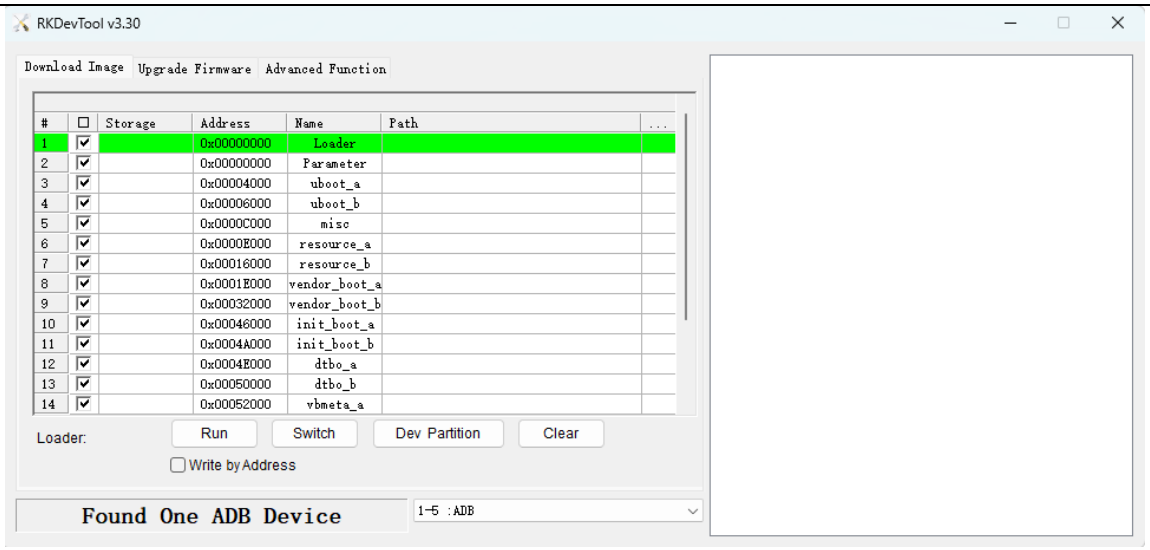
3.2 Firmware Flashing

Environment: Windows OS (Operating System).

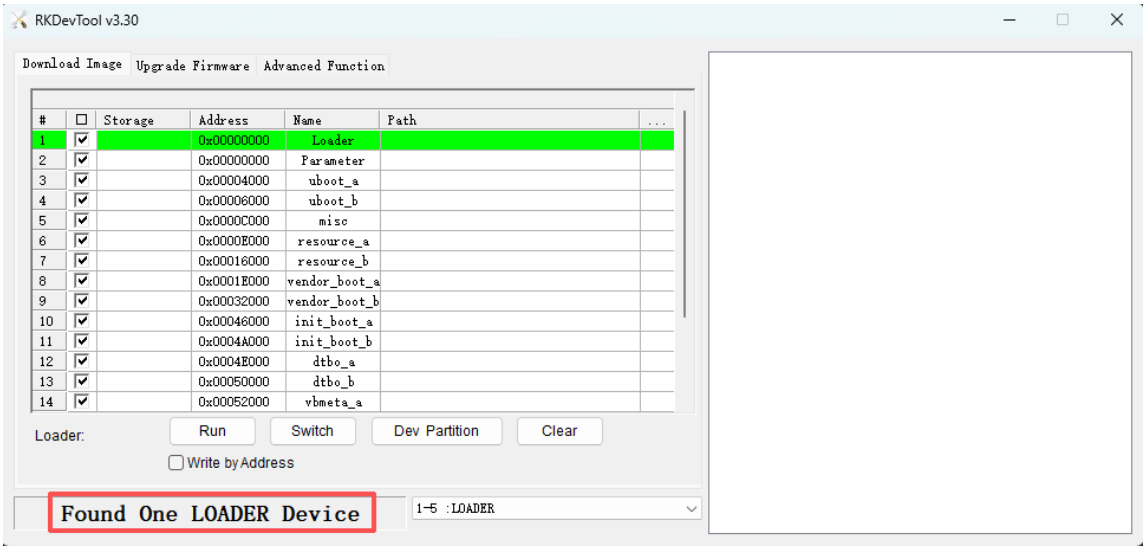
3.2.1 Flash Full Firmware Image

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

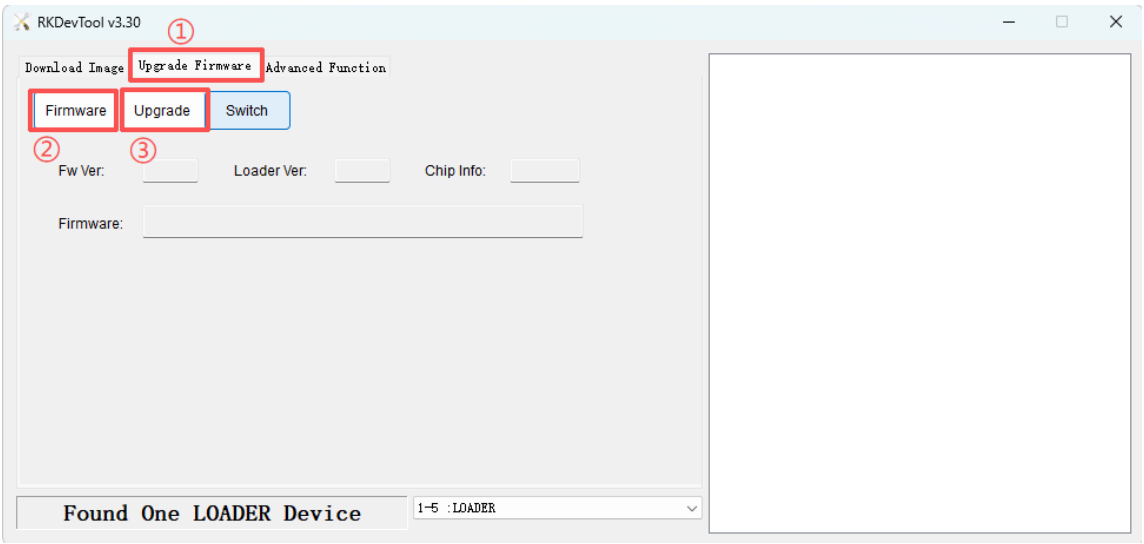
Step 2: Open *RKDevTool_v3.30_for_window\RKDevTool.exe*.



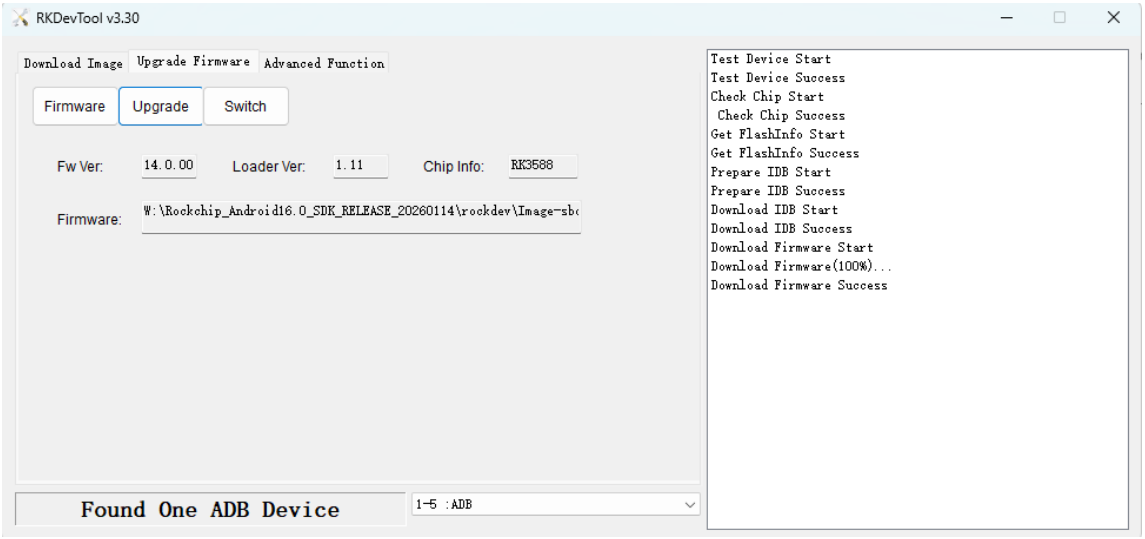
Step 3: Switch to **loader mode**. (*How to Enter Loader Mode*)



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



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



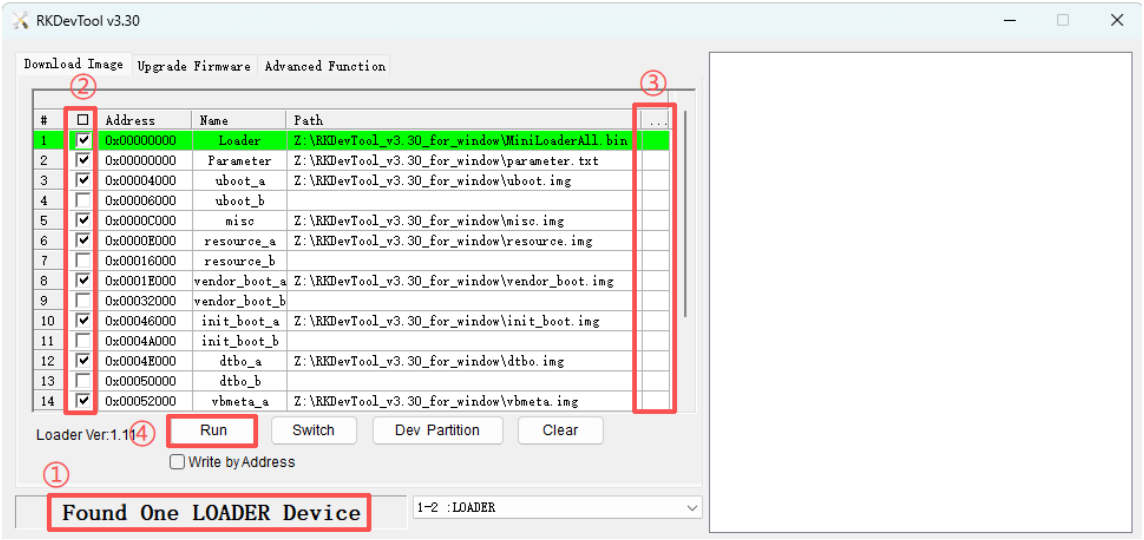
3.2.2 Flash Separate Partition Images

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

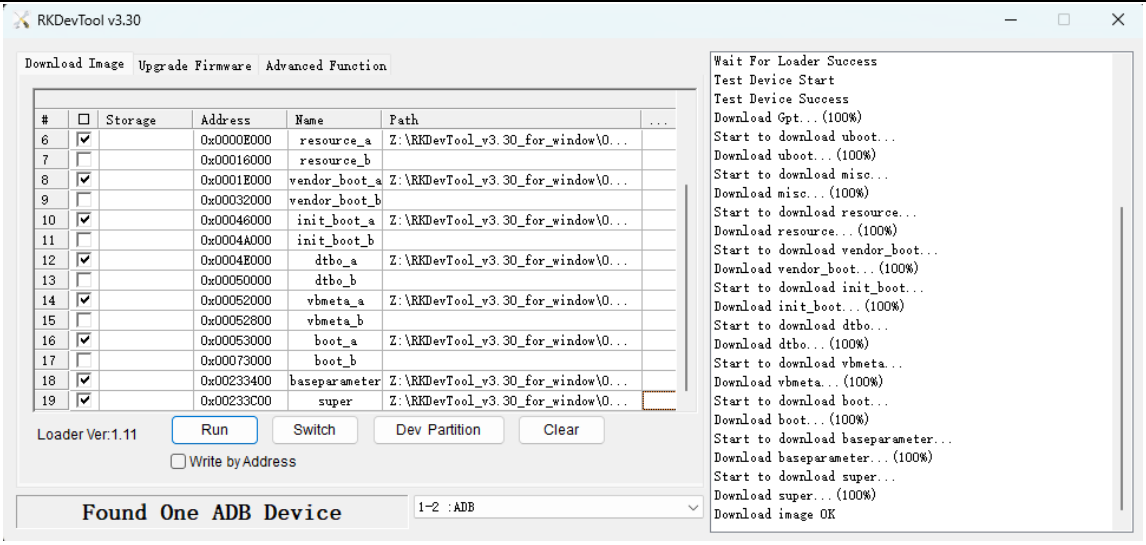
Step 2: Check the partitions to be flashed, multiple partitions can be selected.

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
```

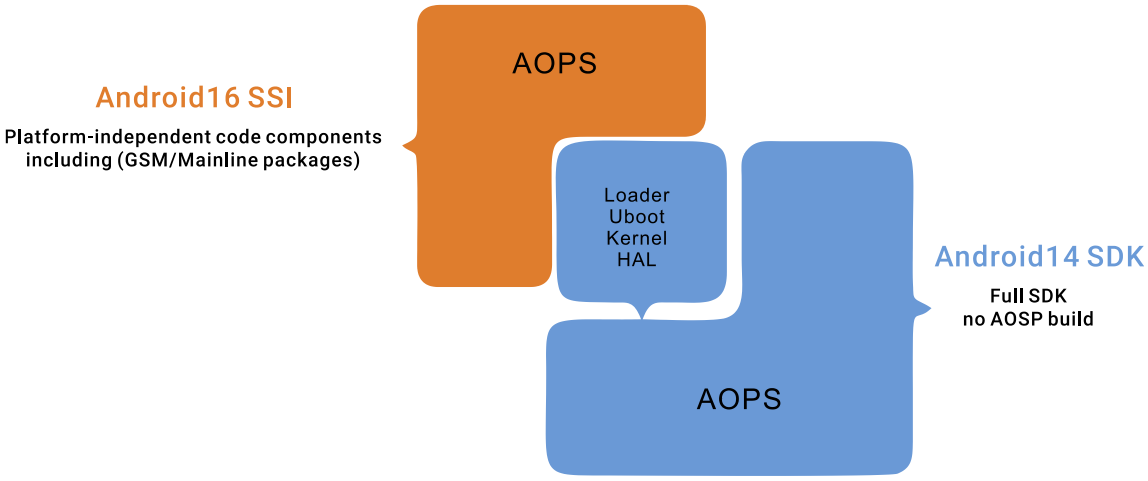
4.3 Installing the JDK

```
$ 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):sbc3588_u-userdebug

Do you need to specify the build number? (Press Enter to automatically generate, or directly enter the
required build number):46

Current PRODUCT_KERNEL_DTS=sbc3588-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
```

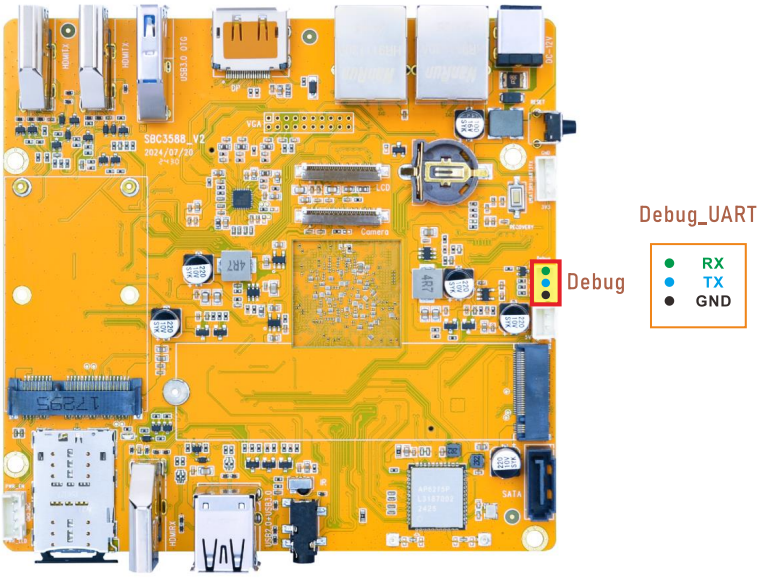
For partial compilation, need to perform merge operation after compilation:

```
$ rkbuild_merge

$ rkbuild_mkupdate
```

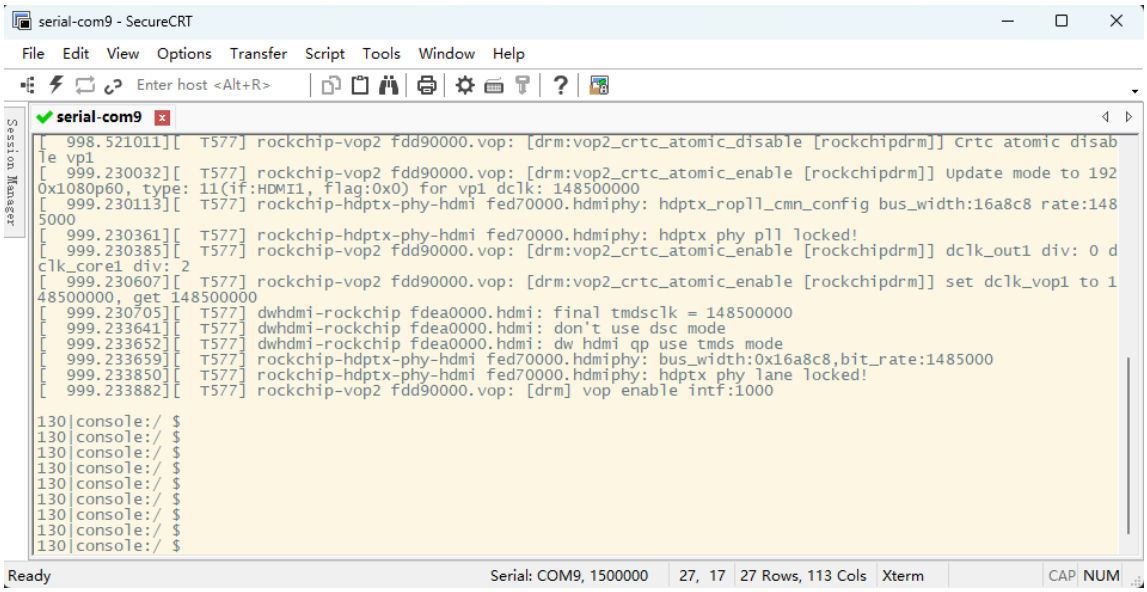
The path of the compiled firmware is /Rockchip_Android16.0_SDK_RELEASE_20260114/rockdev/Image-sbc3588_u/.

6. Android16 Test

6.1 Serial Terminal

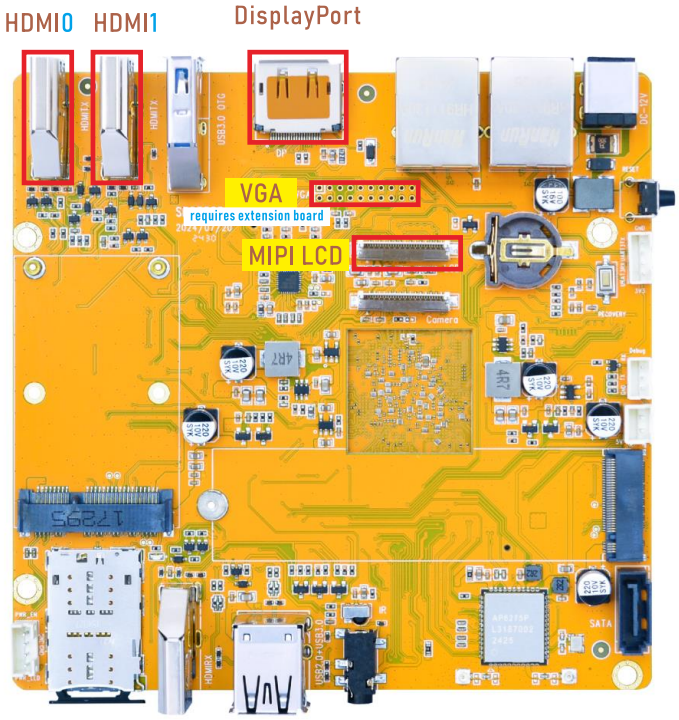


Connect the board and PC with USB Serial cable, then power on, the terminal will output boot information. The default baudrate is 1500000.

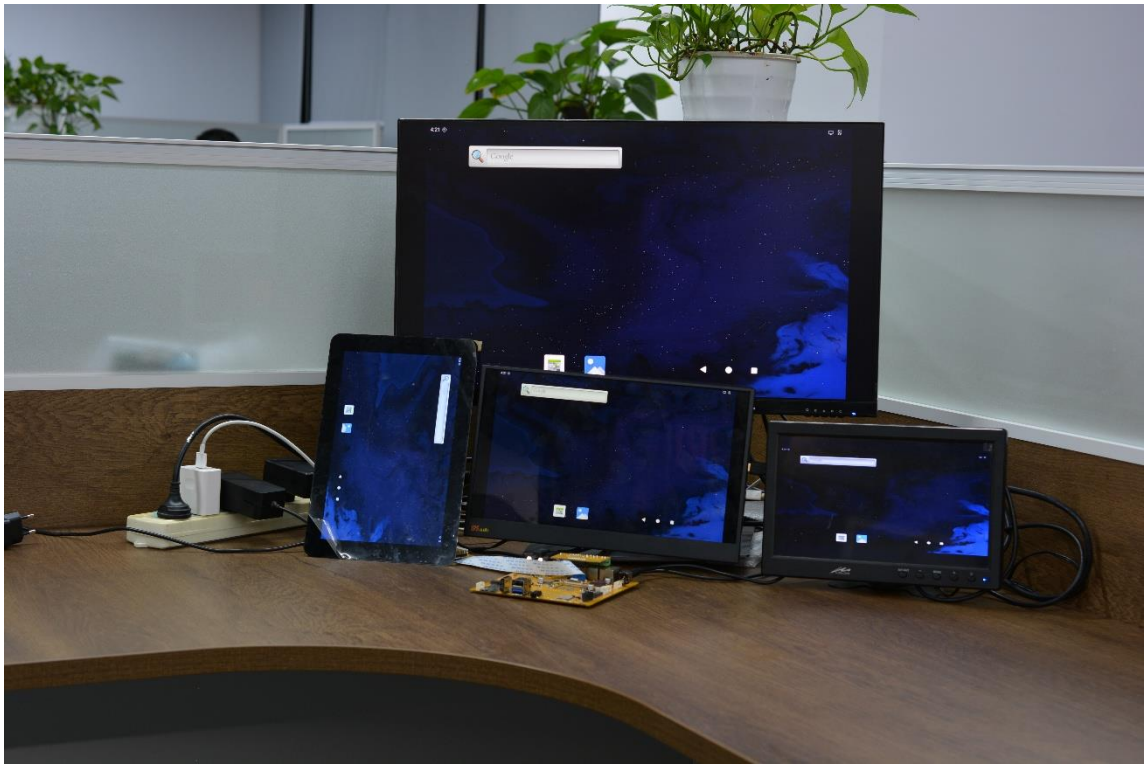


6.2 Display

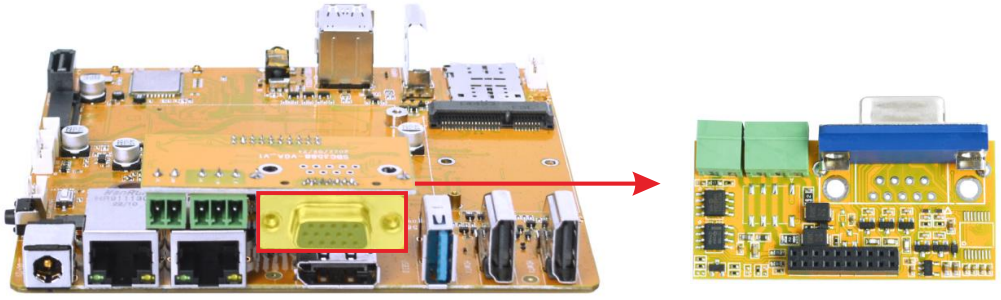
The SBC3588 supports quad-display output on the default Android16 image, with interfaces including HDMI0, HDMI1, DP/VGA (selectable), and MIPI DSI.



The display effect diagram is as follows:



Note: DP and VGA share the same VOP and cannot be used simultaneously with different display timings—only one output will be active at a time.

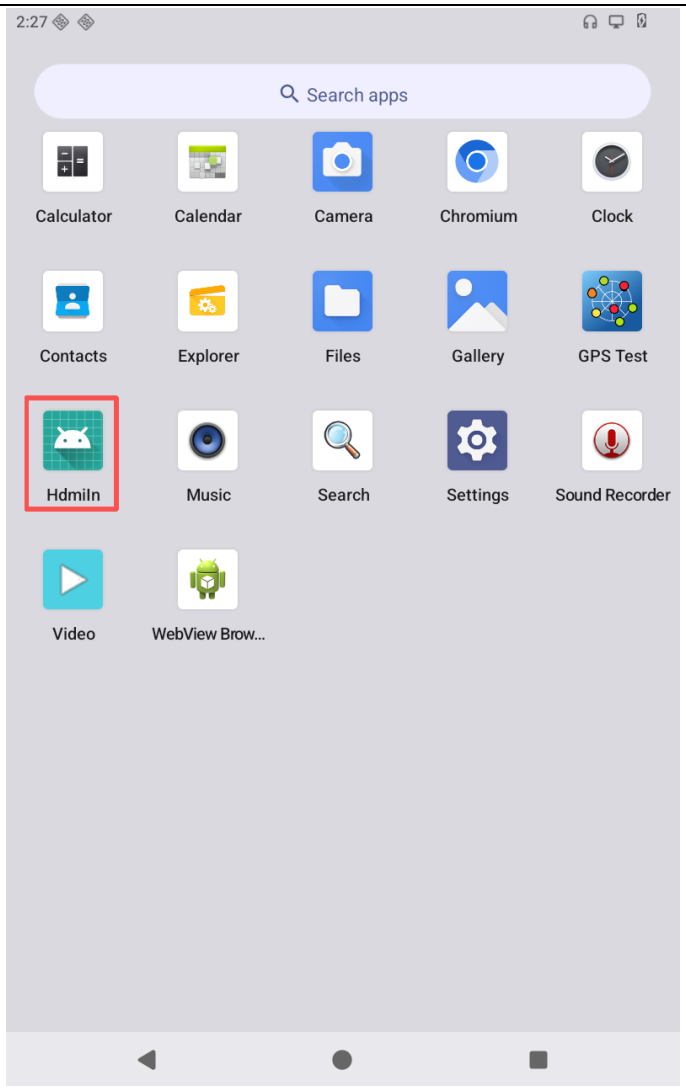


Preview effect:

6.3 HDMI RX

Step 1: Connect the HDMI OUT port of the source device to the HDMI IN port of the SBC3588.

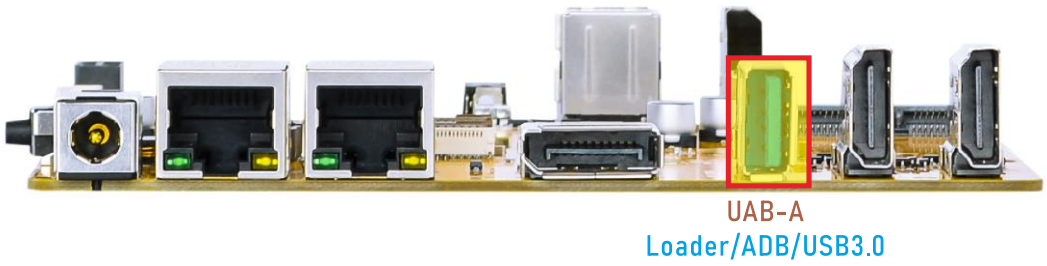
Step 2: Open the **HdmiIn** app：



6.4 USB

6.4.1 USB OTG

Default mode on boot: Device mode, functioning as a USB device for ADB debugging and file transfer.



6.4.1.1 Host mode

In this mode, the board functions as a USB host, allowing connection to:

- USB keyboard
- USB mouse
- USB flash drive

To switch to host mode, execute the following command:

```
$ su
# echo host > /sys/devices/platform/fd5d0000.syscon/fd5d0000.syscon:usb2-
phy@0/otg_mode
```

```
1|console:/ $ su
sys/devices/platform/fd5d0000.syscon/fd5d0000.syscon:usb2-phy@0/otg_mode     <
console:/ # [ 2105.409657][  T951] dwc3 fc000000.usb: request 0000000000000000 was not queued to ep0out
[ 2105.410441][ T2623] android_work: sent uevent USB_STATE=DISCONNECTED
[ 2105.458345][  T943] read descriptors
[ 2105.458378][  T943] read strings
[ 2105.520129][  T951] phy phy-fd5d0000.syscon:usb2-phy@0.8: illegal mode
[ 2105.520199][  T951] xhci-hcd xhci-hcd.16.auto: xHCI Host Controller
[ 2105.520256][  T951] xhci-hcd xhci-hcd.16.auto: new USB bus registered, assigned bus number 8
[  2105.520488][    T951]  xhci-hcd  xhci-hcd.16.auto:  hcc  params  0x0220fe64  hci  version  0x110  quirks
0x0000008002010010
[ 2105.520560][  T951] xhci-hcd xhci-hcd.16.auto: irq 76, io mem 0xfc000000
[ 2105.521051][  T951] xhci-hcd xhci-hcd.16.auto: xHCI Host Controller
[ 2105.521100][  T951] xhci-hcd xhci-hcd.16.auto: new USB bus registered, assigned bus number 9
[ 2105.521137][  T951] xhci-hcd xhci-hcd.16.auto: Host supports USB 3.0 SuperSpeed
[ 2105.521550][  T951] usb usb8: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice= 6.01
[ 2105.521589][  T951] usb usb8: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 2105.521720][  T951] usb usb8: Product: xHCI Host Controller
[ 2105.521748][  T951] usb usb8: Manufacturer: Linux 6.1.145-android14-11-g4f184da362d5-ab14200214 xhci-
hcd
[ 2105.521776][  T951] usb usb8: SerialNumber: xhci-hcd.16.auto
[ 2105.526073][  T951] hub 8-0:1.0: USB hub found
[ 2105.526182][  T951] hub 8-0:1.0: 1 port detected
[ 2105.533715][  T951] usb usb9: We don't know the algorithms for LPM for this host, disabling LPM.
[ 2105.534282][  T951] usb usb9: New USB device found, idVendor=1d6b, idProduct=0003, bcdDevice= 6.01
[ 2105.534340][  T951] usb usb9: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 2105.534384][  T951] usb usb9: Product: xHCI Host Controller
[ 2105.534423][  T951] usb usb9: Manufacturer: Linux 6.1.145-android14-11-g4f184da362d5-ab14200214 xhci-
hcd
[ 2105.534463][  T951] usb usb9: SerialNumber: xhci-hcd.16.auto
[ 2105.539542][  T951] hub 9-0:1.0: USB hub found
[ 2105.539671][  T951] hub 9-0:1.0: 1 port detected
[ 2105.898524][  T791] healthd: battery l=50 v=3300 t=2.6 h=2 st=3 c=-1600 fc=100 chg=au
```

6.4.1.2 Device mode

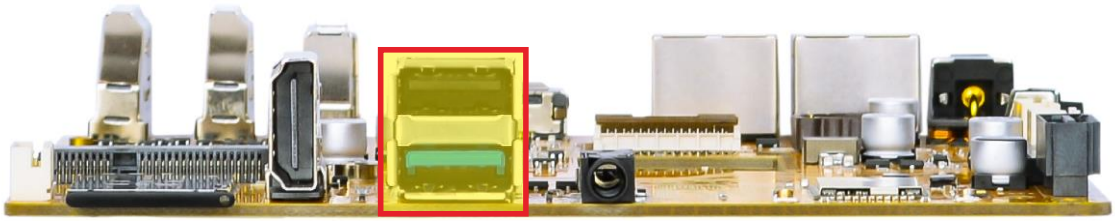
To switch back to Device mode, execute the following command:

```
# echo peripheral > /sys/devices/platform/fd5d0000.syscon/fd5d0000.syscon:usb2-
phy@0/otg_mode
```

```
m/fd5d0000.syscon/fd5d0000.syscon:usb2-phy@0/otg_mode                    <
console:/ # [ 1471.971256][ T2849] xhci-hcd xhci-hcd.16.auto: remove, state 4
[ 1471.971291][ T2849] usb usb9: USB disconnect, device number 1
[ 1471.974851][ T2849] xhci-hcd xhci-hcd.16.auto: USB bus 9 deregistered
[ 1471.975257][ T2849] xhci-hcd xhci-hcd.16.auto: remove, state 4
[ 1471.975278][ T2849] usb usb8: USB disconnect, device number 1
[ 1471.976376][ T2849] xhci-hcd xhci-hcd.16.auto: USB bus 8 deregistered
[ 1472.215218][ T2297] android_work: did not send uevent (0 0 0000000000000000)
[ 1472.425037][ T3025] dwc3 fc000000.usb: device reset
[ 1472.517585][ T2297] android_work: sent uevent USB_STATE=CONNECTED
[ 1472.548004][ T2297] android_work: sent uevent USB_STATE=CONFIGURED
[ 1473.237169][ T3025] dwc3 fc000000.usb: device reset
[ 1473.292668][  T723] dwc3 fc000000.usb: request 0000000000000000 was not queued to ep0out
[ 1473.292763][ T2426] android_work: sent uevent USB_STATE=DISCONNECTED
[ 1473.399018][ T3045] dwc3 fc000000.usb: device reset
[ 1473.575020][ T3045] dwc3 fc000000.usb: device reset
[ 1473.668766][ T2297] android_work: sent uevent USB_STATE=CONNECTED
[ 1473.700813][ T2297] android_work: sent uevent USB_STATE=CONFIGURED
```

6.4.2 USB Host

The USB host interface provides connectivity for standard peripherals, including USB keyboards, mice, flash drives, and other USB-compatible devices.



USB version identification based on kernel log output:

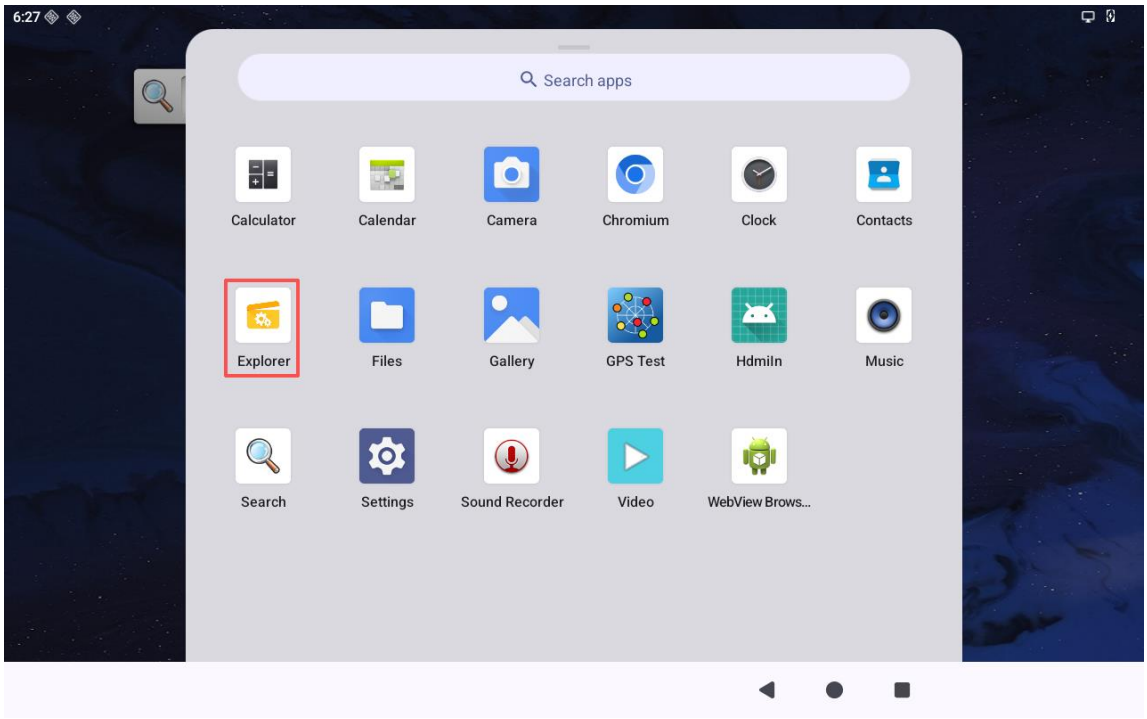
• **USB 2.0**: Displays as high-speed.

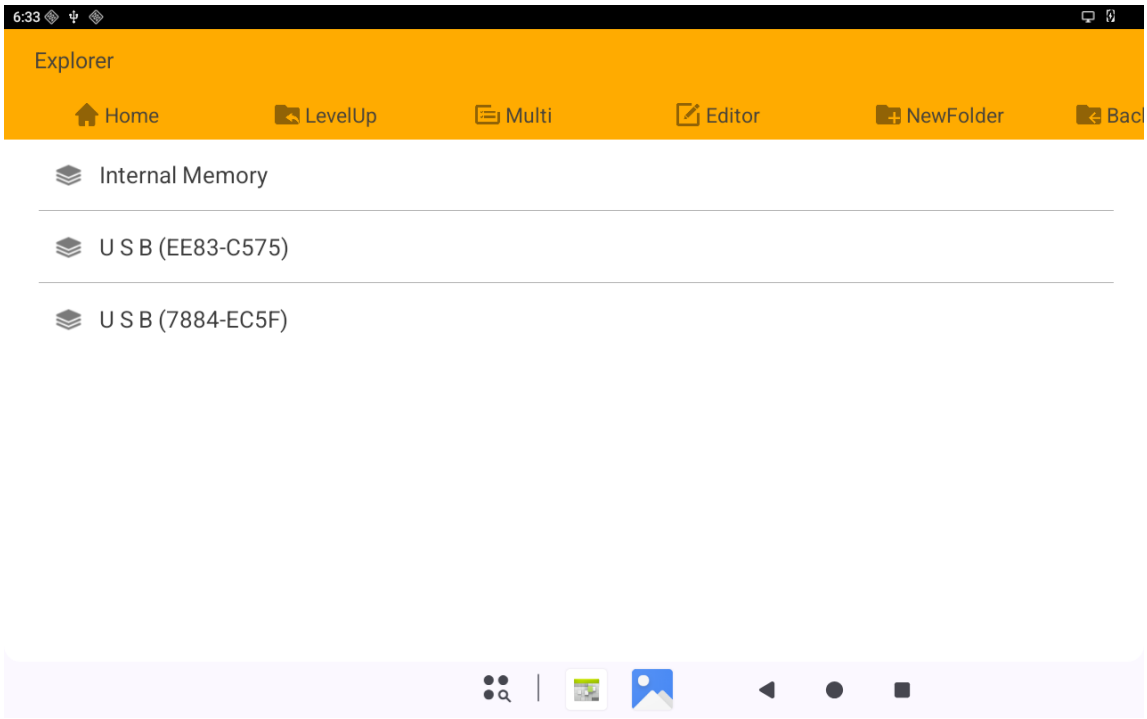
```
console:/ # [ 2404.113642][ T2849] usb 3-1: new high-speed USB device number 2 using xhci-hcd
[ 2404.263958][ T2849] usb 3-1: New USB device found, idVendor=346d, idProduct=5678, bcdDevice= 3.20
[ 2404.264057][ T2849] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 2404.264101][ T2849] usb 3-1: Product: HIKSEMI
[ 2404.264137][ T2849] usb 3-1: Manufacturer:
[ 2404.264173][ T2849] usb 3-1: SerialNumber: FC054E5F74301
[ 2404.266718][ T2849] usb-storage 3-1:1.0: USB Mass Storage device detected
[ 2404.268074][ T2849] scsi host1: usb-storage 3-1:1.0
[ 2405.299650][ T2304] scsi 1:0:0:0: Direct-Access            HIKSEMI         3.20 PQ: 0 ANSI: 4
[ 2405.306476][ T2304] sd 1:0:0:0: Attached scsi generic sg0 type 0
[ 2405.308170][ T2319] sd 1:0:0:0: [sda] 245760000 512-byte logical blocks: (126 GB/117 GiB)
[ 2405.308561][ T2319] sd 1:0:0:0: [sda] Write Protect is off
[ 2405.308894][ T2319] sd 1:0:0:0: [sda] No Caching mode page found
[ 2405.308911][ T2319] sd 1:0:0:0: [sda] Assuming drive cache: write through
 [ 2405.312679][ T2319]  sda: sda1
[ 2405.313122][ T2319] sd 1:0:0:0: [sda] Attached SCSI removable disk
 [ 2405.995540][ T1558] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please
run fsck.
```

• **USB 3.0**: Displays as SuperSpeed.

```
console:/ # [ 2471.950185][ T3141] usb 2-1: new SuperSpeed USB device number 3 using xhci-hcd
[ 2471.981398][ T3141] usb 2-1: New USB device found, idVendor=0dd8, idProduct=3b00, bcdDevice= 0.02
[ 2471.981646][ T3141] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 2471.981694][ T3141] usb 2-1: Product: OnlyDisk
[ 2471.981730][ T3141] usb 2-1: Manufacturer: Netac
[ 2471.981766][ T3141] usb 2-1: SerialNumber: C0E8BFA3EC38F796
[ 2471.991364][ T3141] usb-storage 2-1:1.0: USB Mass Storage device detected
[ 2471.994222][ T3141] scsi host2: usb-storage 2-1:1.0
[ 2473.428951][ T3139] scsi 2:0:0:0: Direct-Access     Netac    OnlyDisk        8.01 PQ: 0 ANSI: 6
[ 2473.430738][ T3139] sd 2:0:0:0: Attached scsi generic sg1 type 0
[ 2473.433481][ T2304] sd 2:0:0:0: [sdb] 121610240 512-byte logical blocks: (62.3 GB/58.0 GiB)
[ 2473.433786][ T2304] sd 2:0:0:0: [sdb] Write Protect is off
[ 2473.434002][ T2304] sd 2:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO
or FUA
[ 2473.440559][ T2304]  sdb: sdb1
[ 2473.441365][ T2304] sd 2:0:0:0: [sdb] Attached SCSI removable disk
```

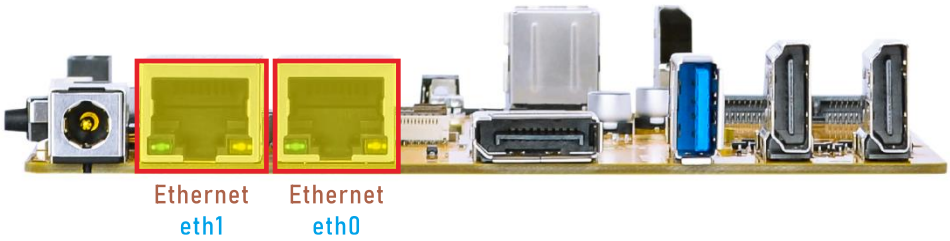
After connection, a USB flash drive is mounted automatically and becomes accessible via the File Explorer.





6.5 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:/ # [   35.711227][  T152] rk_gmac-dwmac fe1b0000.ethernet eth1: Link is Up - 1Gbps/Full - flow
control rx/tx
[   35.711304][  T152] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
console:/ # [   49.023202][  T156] rk_gmac-dwmac fe1c0000.ethernet eth0: Link is Up - 1Gbps/Full - flow
control rx/tx
[   49.023279][  T156] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

Step 2: View network interface information.

```
# ifconfig
```

```
console:/ # ifconfig
eth0      Link encap:Ethernet  HWaddr fe:5d:9a:19:c0:aa  Driver rk_gmac-dwmac
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:160 errors:0 dropped:27 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19254 TX bytes:3733
          Interrupt:73

eth1      Link encap:Ethernet  HWaddr fa:5d:9a:19:c0:aa  Driver rk_gmac-dwmac
          inet addr:192.168.0.195  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::da6b:4622:5ec6:1160/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:206 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30520 TX bytes:4925
          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.

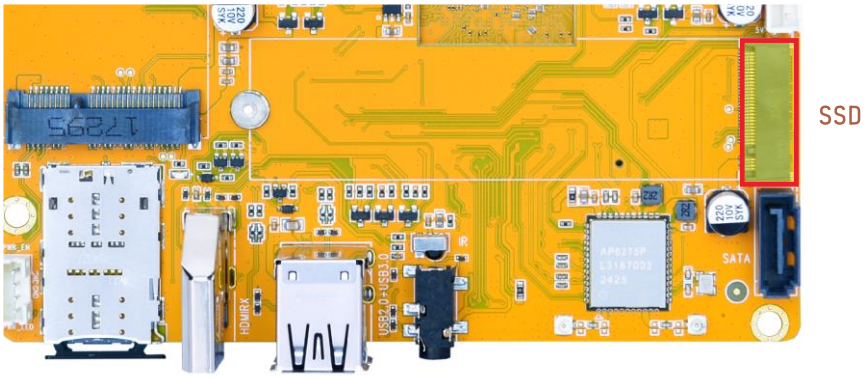
Step 3: Network connection test.

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

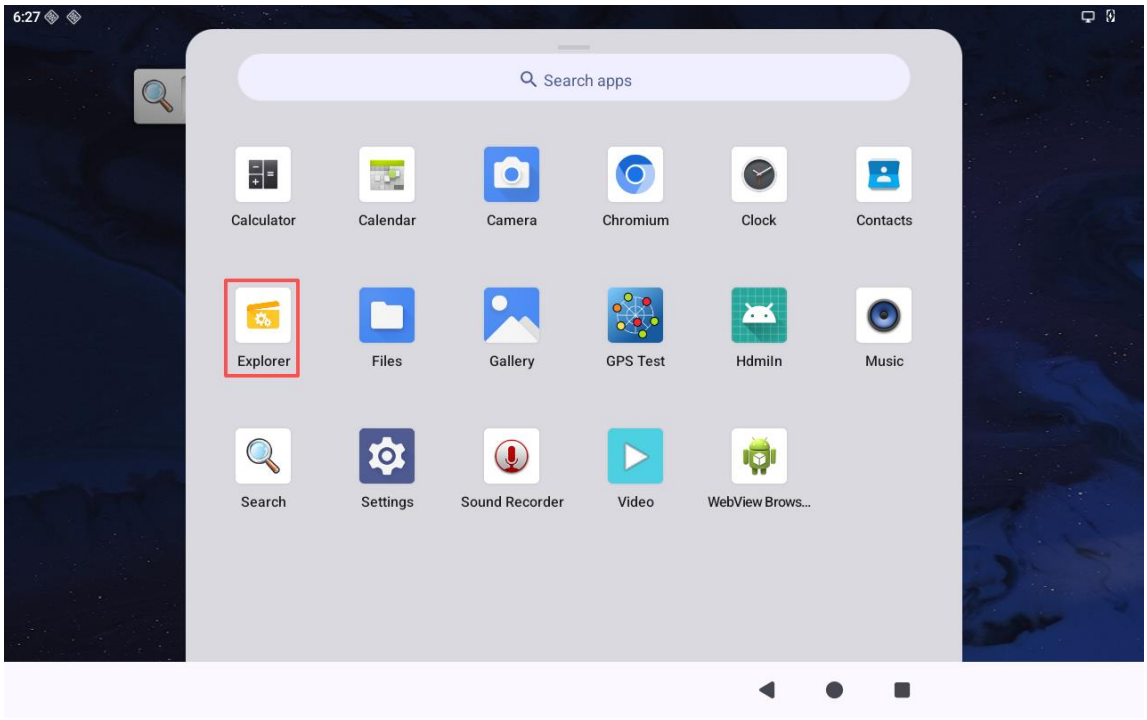
```
console:/ # ping -I eth0 192.168.0.1
ping: Warning: source address might be selected on device other than eth0.
PING 192.168.0.1 (192.168.0.1) from 192.168.0.195 eth0: 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=3.61 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=0.558 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=0.421 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=255 time=0.452 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=255 time=0.485 ms
64 bytes from 192.168.0.1: icmp_seq=6 ttl=255 time=0.374 ms
^C
--- 192.168.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5072ms
rtt min/avg/max/mdev = 0.374/0.984/3.618/1.179 ms
console:/ #
console:/ # ping -I eth1 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.195 eth1: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=50 time=189 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=50 time=189 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=50 time=189 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=50 time=189 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=50 time=189 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=6 ttl=50 time=189 ms
^C
--- www.armdesigner.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 189.001/189.282/189.739/0.434 ms
```

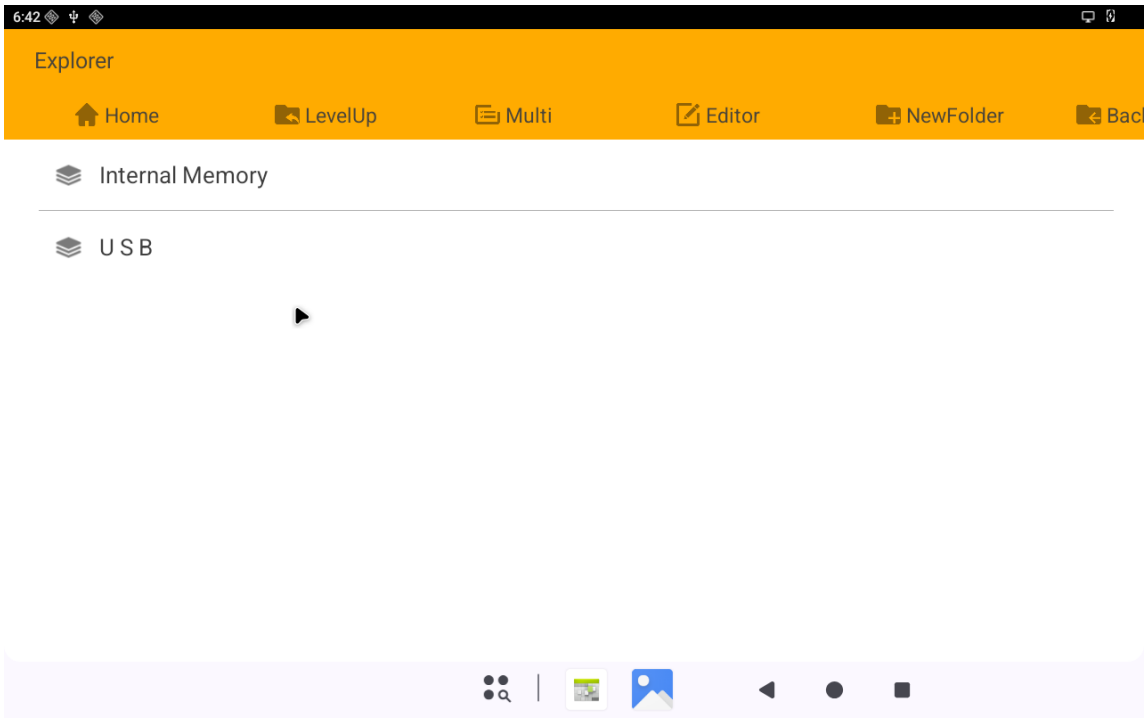
6.6 PCIe M.2 SSD

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



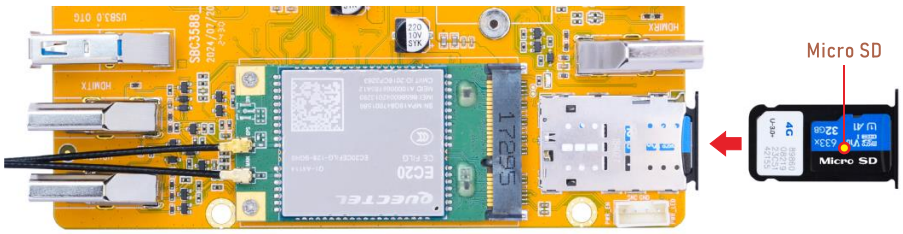
Step 2: The system will automatically detect and mount the SSD, and it can be accessed directly through the **Explorer** app.



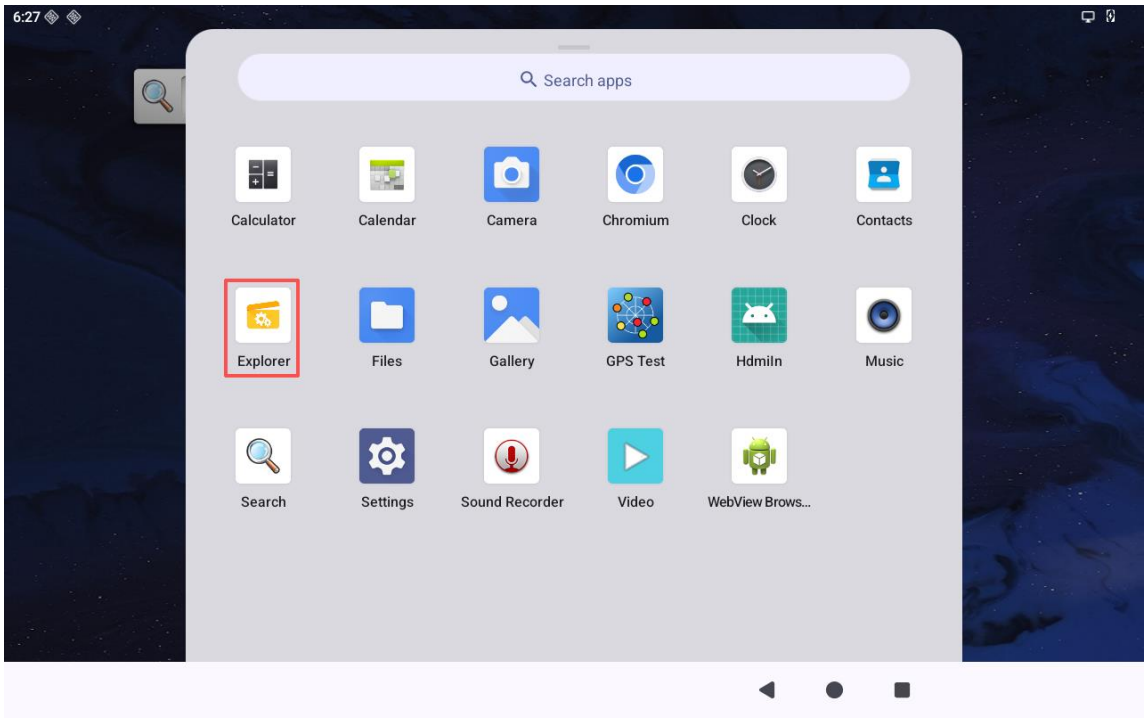


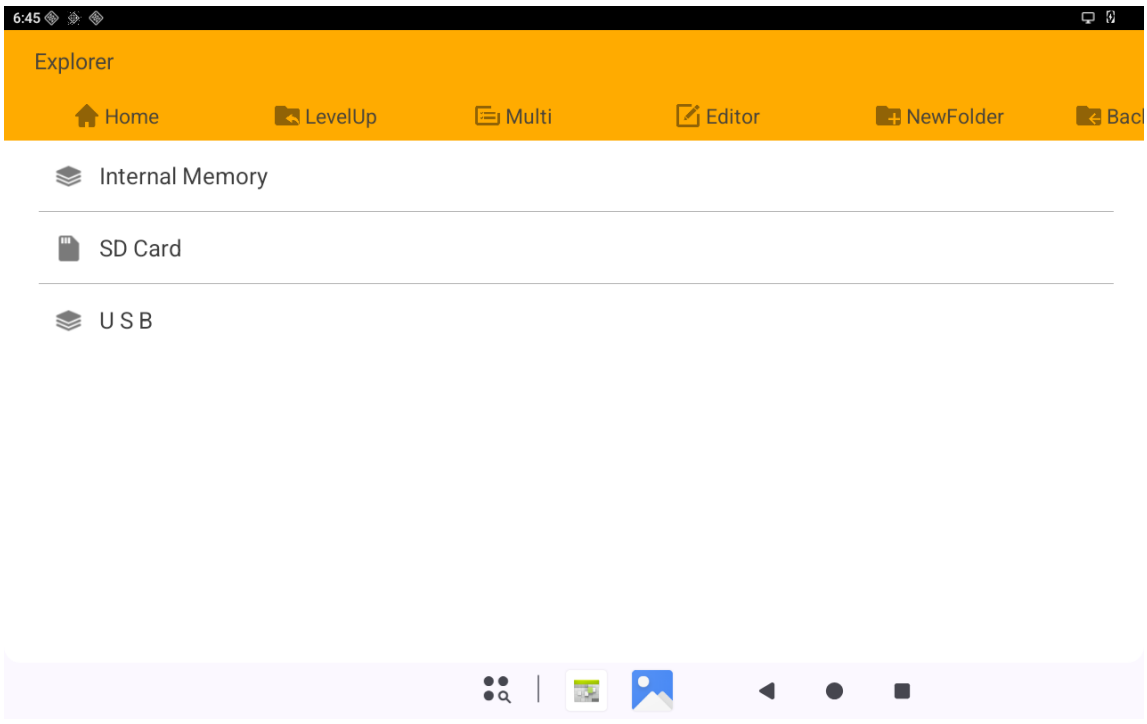
6.7 SD Card

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



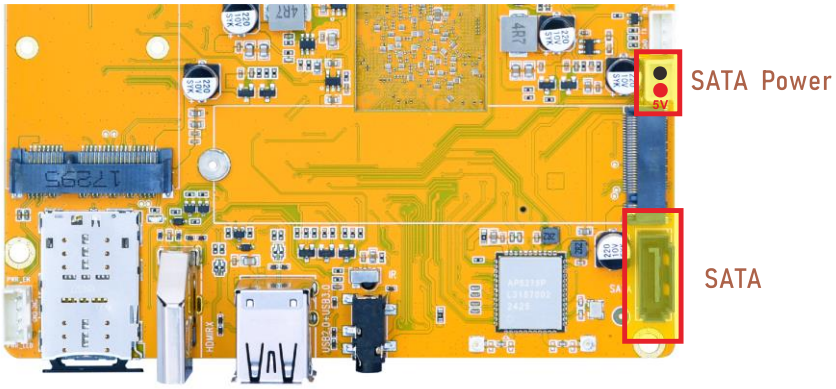
Step 2: After connecting the SD card drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.



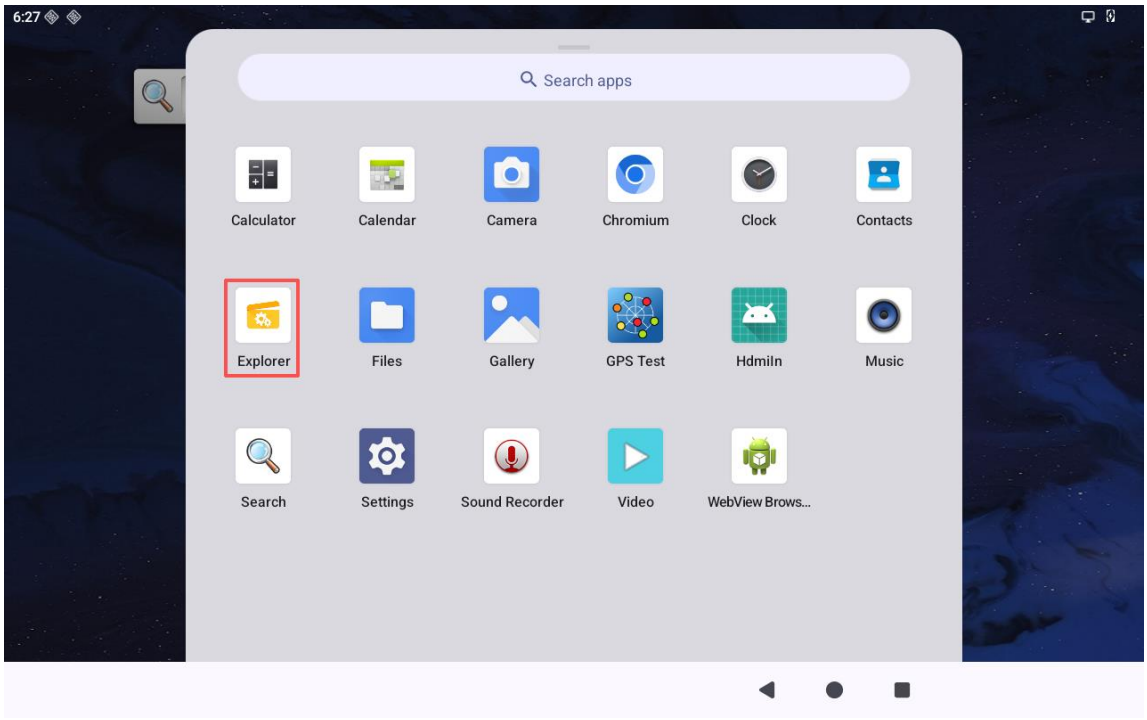


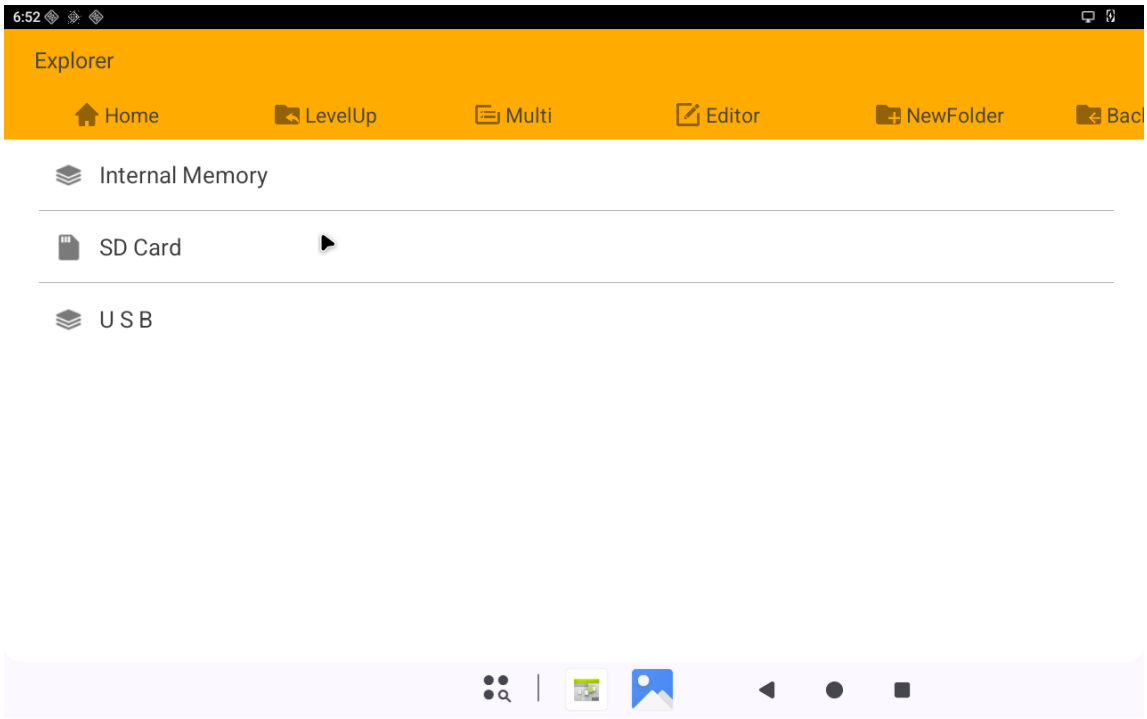
6.8 SATA

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



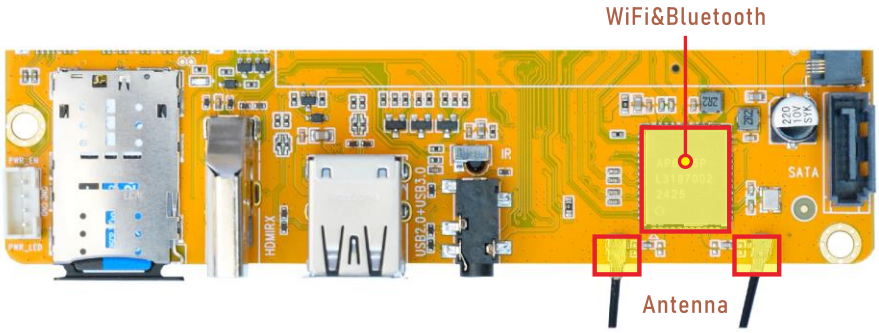
Step 2: After connecting the sata drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.





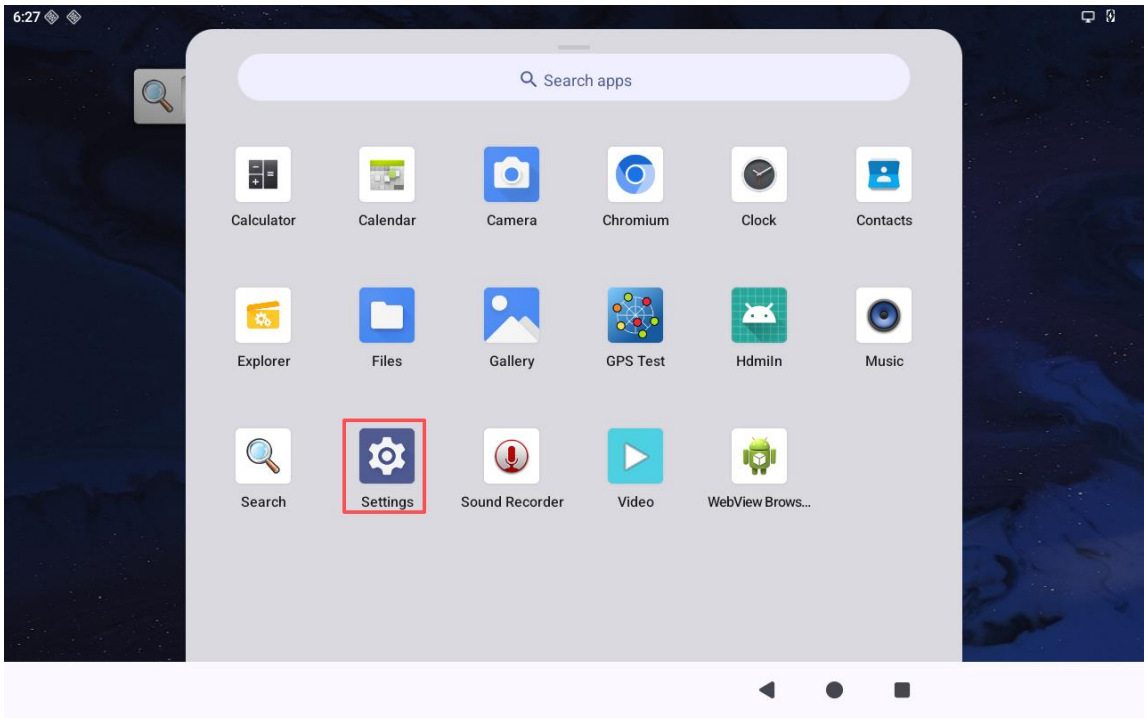
6.9 WiFi & Bluetooth

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

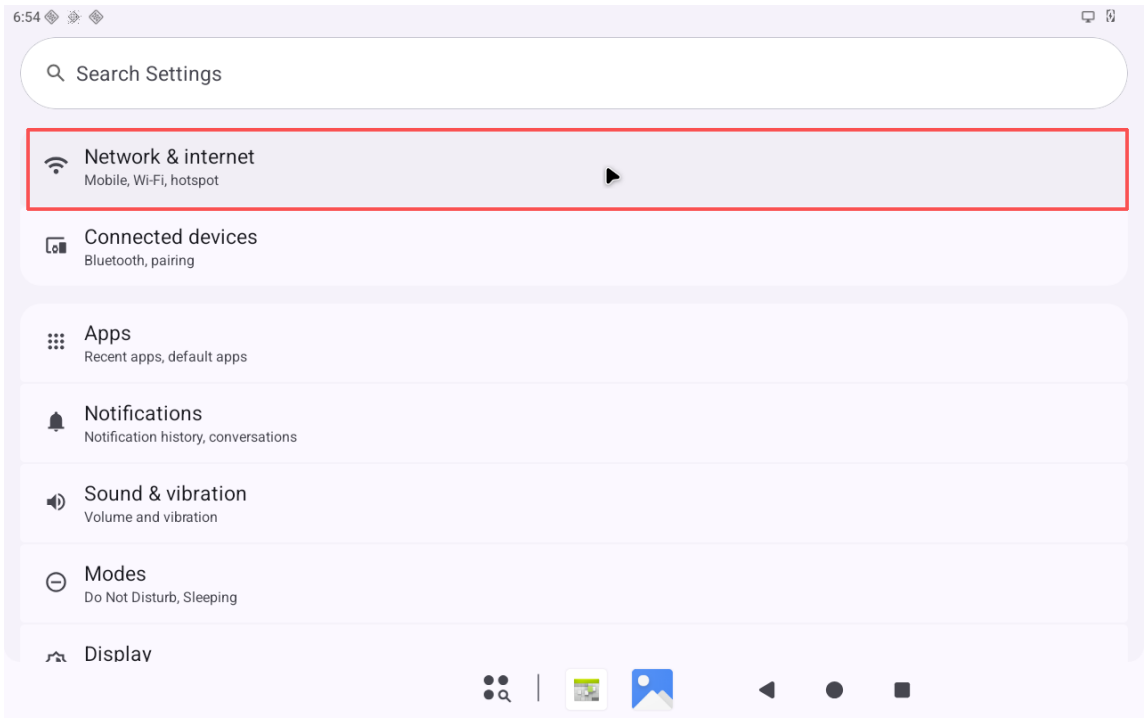


6.9.1 WiFi test

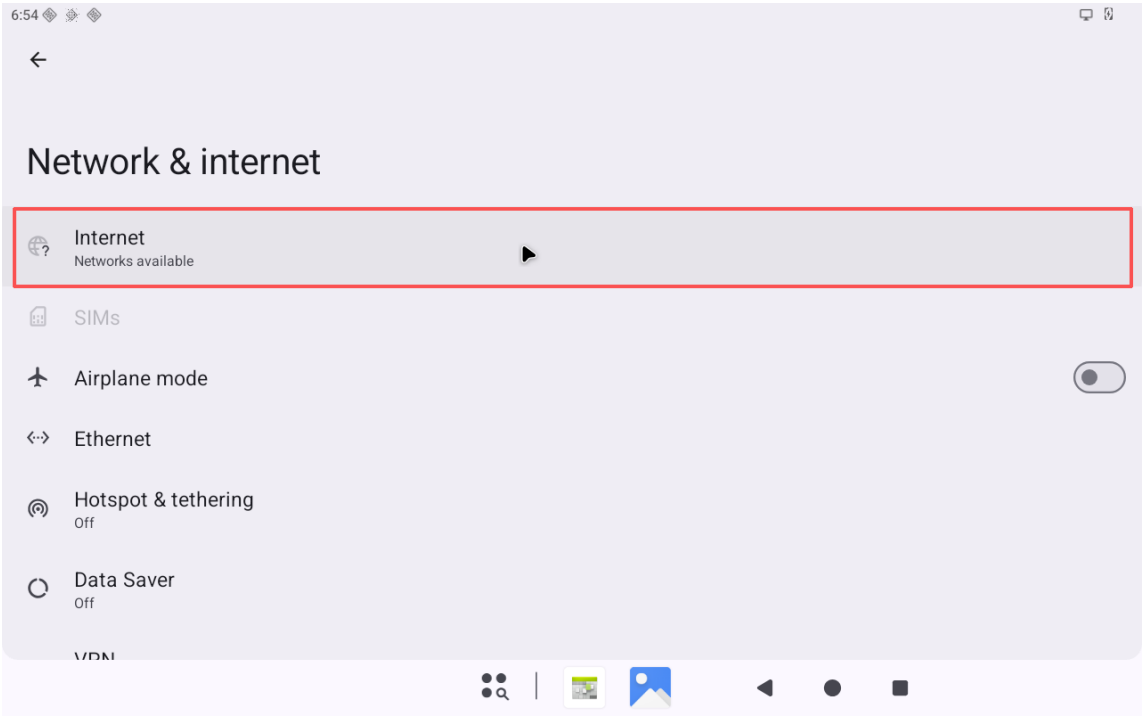
Step 1: Open the **Settings** app：



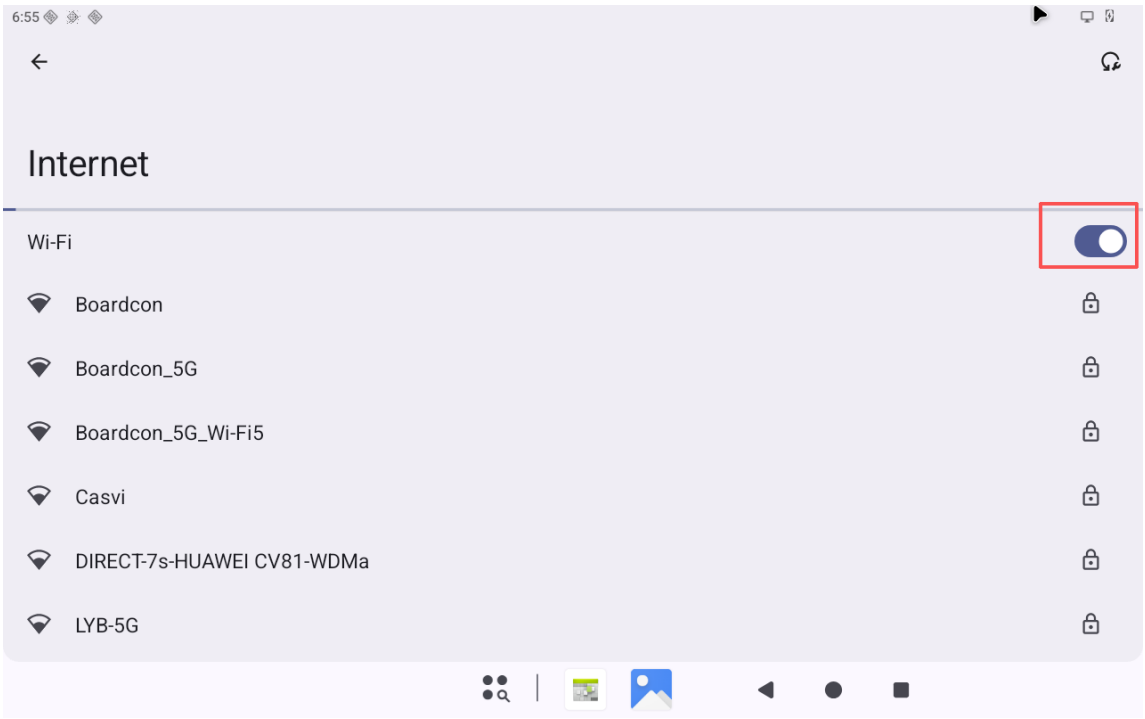
Step 2: Go to **Network & Internet**:



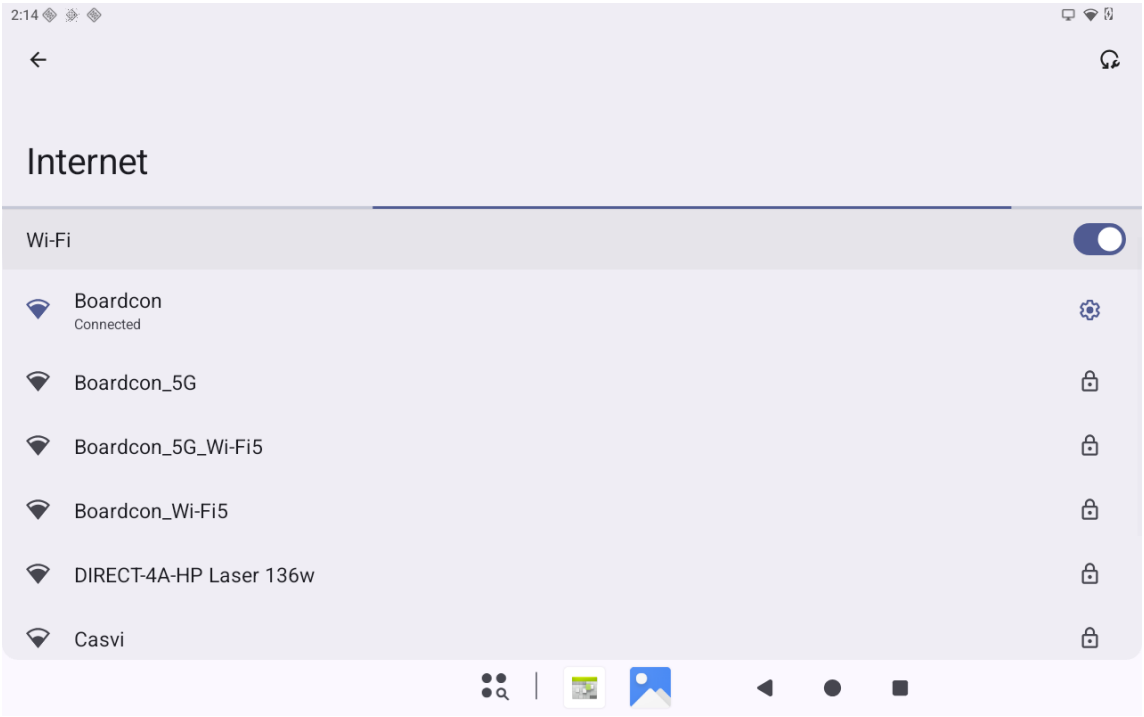
Step 3: Tap **Internet**:



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.

```
# ifconfig
```

```
console:/ # ifconfig
 wlan0     Link encap:Ethernet  HWaddr 54:78:c9:88:4b:34
          inet addr:192.168.0.47  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::851d:7db4:219d:5435/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:51 errors:0 dropped:27 overruns:0 frame:0
          TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12076 TX bytes:4280
```

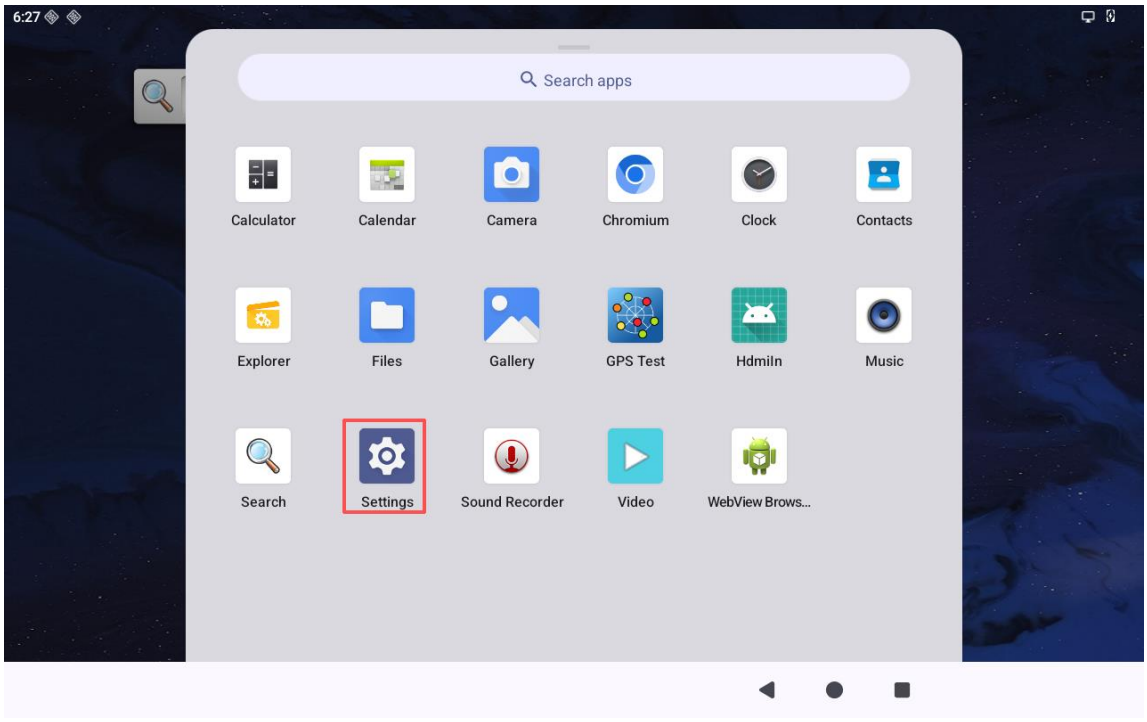
(2) Network connection test.

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

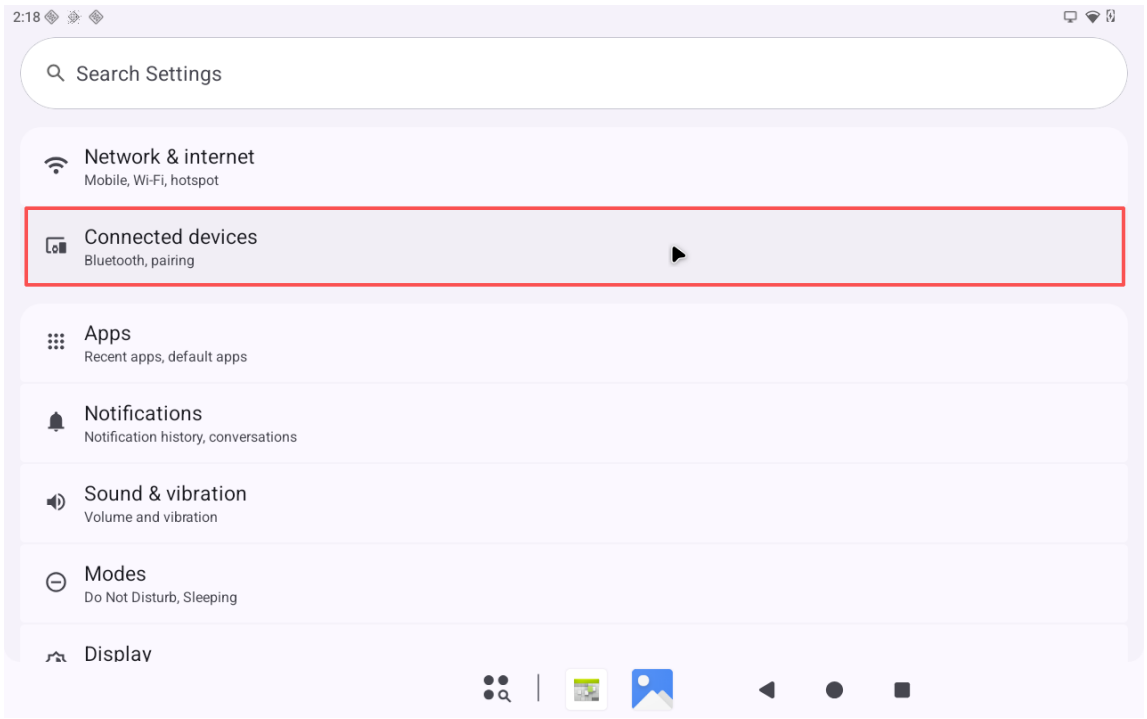
```
console:/ # ping -I wlan0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.47 wlan0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=50 time=198 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=50 time=193 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=50 time=197 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=50 time=200 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=50 time=194 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=6 ttl=50 time=416 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=7 ttl=50 time=197 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 193.611/228.356/416.423/76.809 ms
```

6.9.2 Bluetooth test

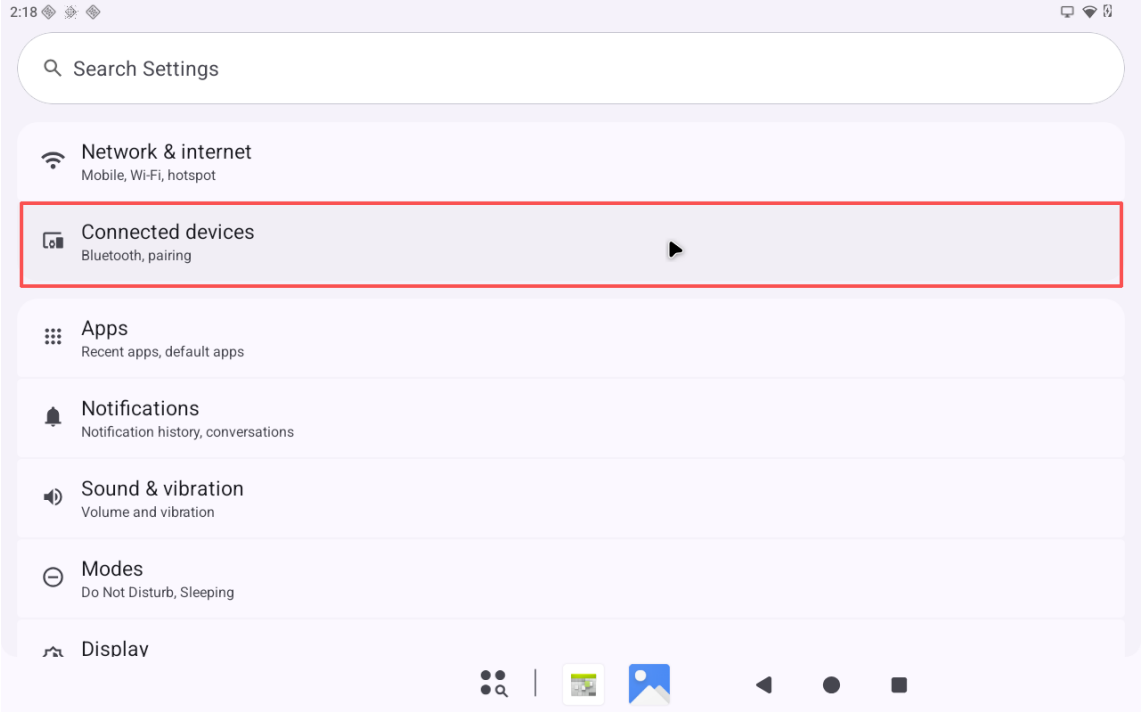
Step 1: Open the **Settings** app.



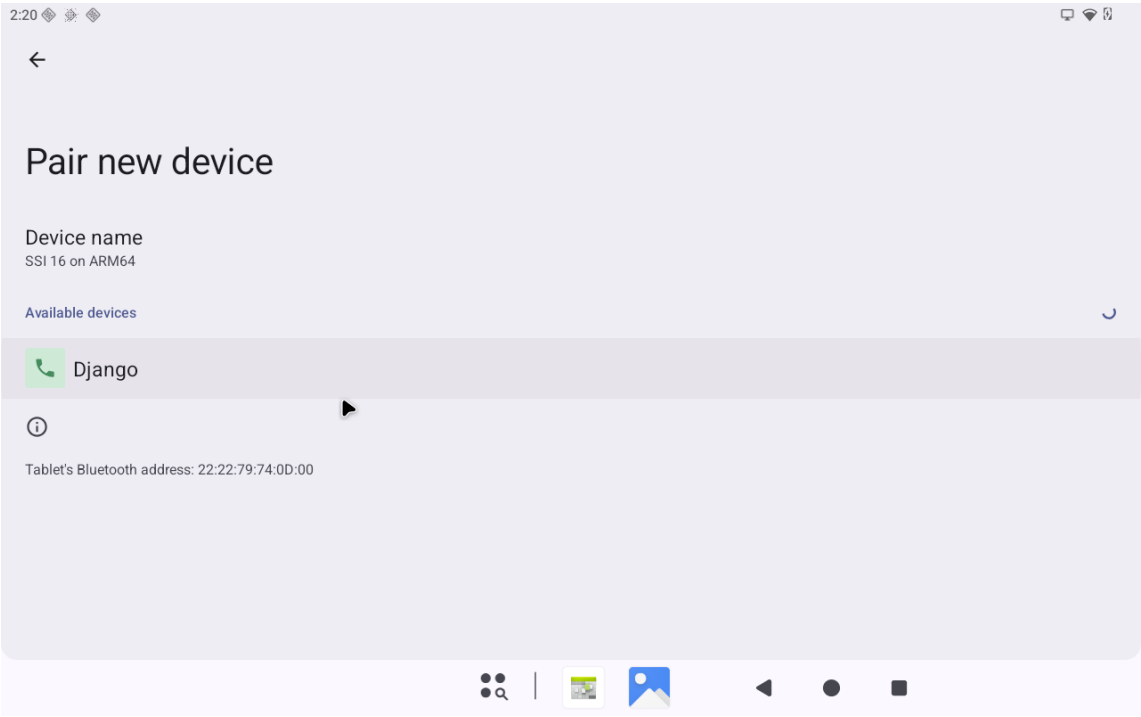
Step 2: Click on the option **Connected devices**



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



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

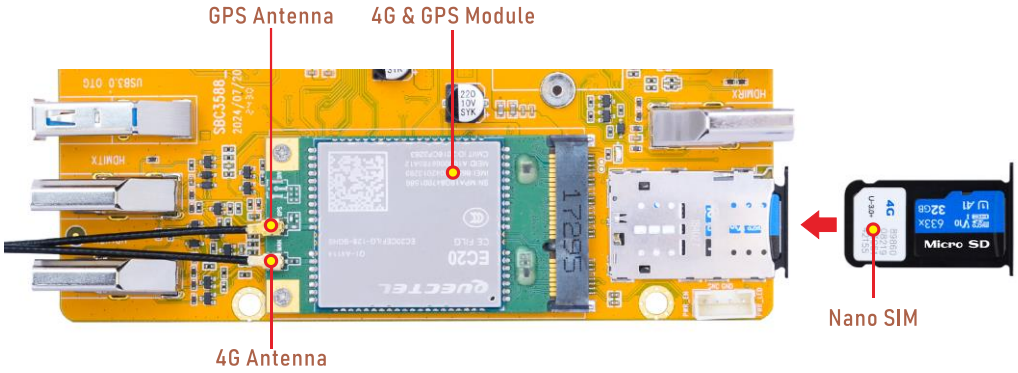


6.10 4G & GPS

Step 1: Insert 4G module to PCIe socket (4G model: EC20/EC25).

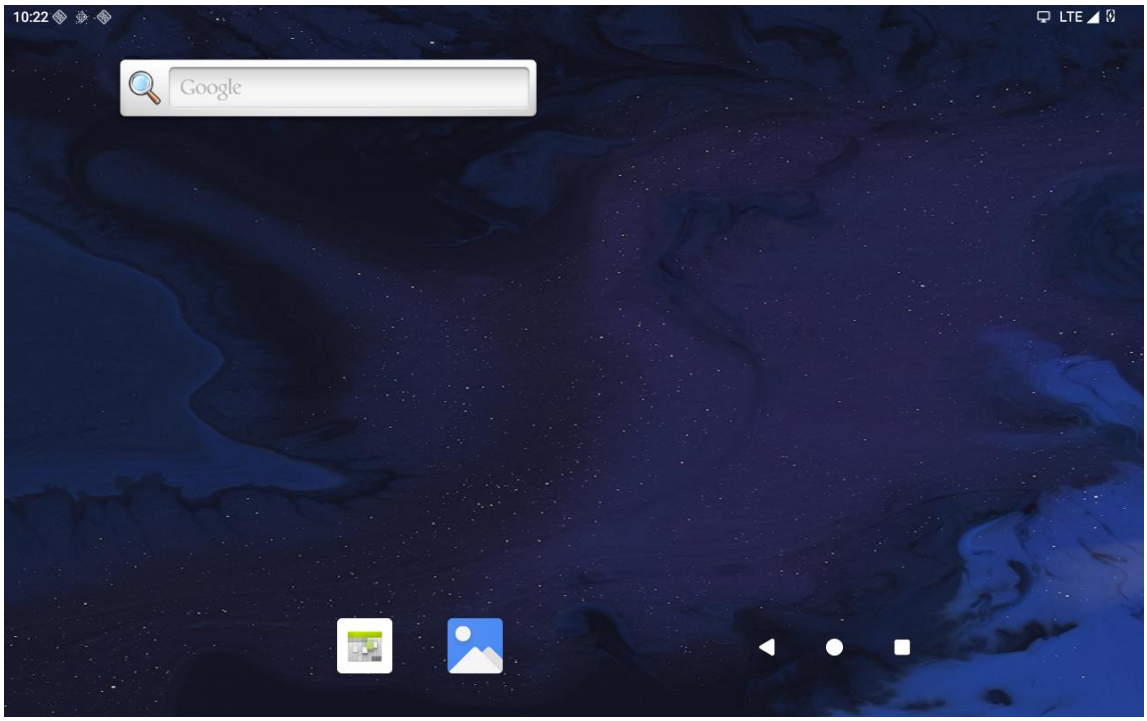
Step 2: Connect antenna and insert SIM card.

Step 3: Power on.

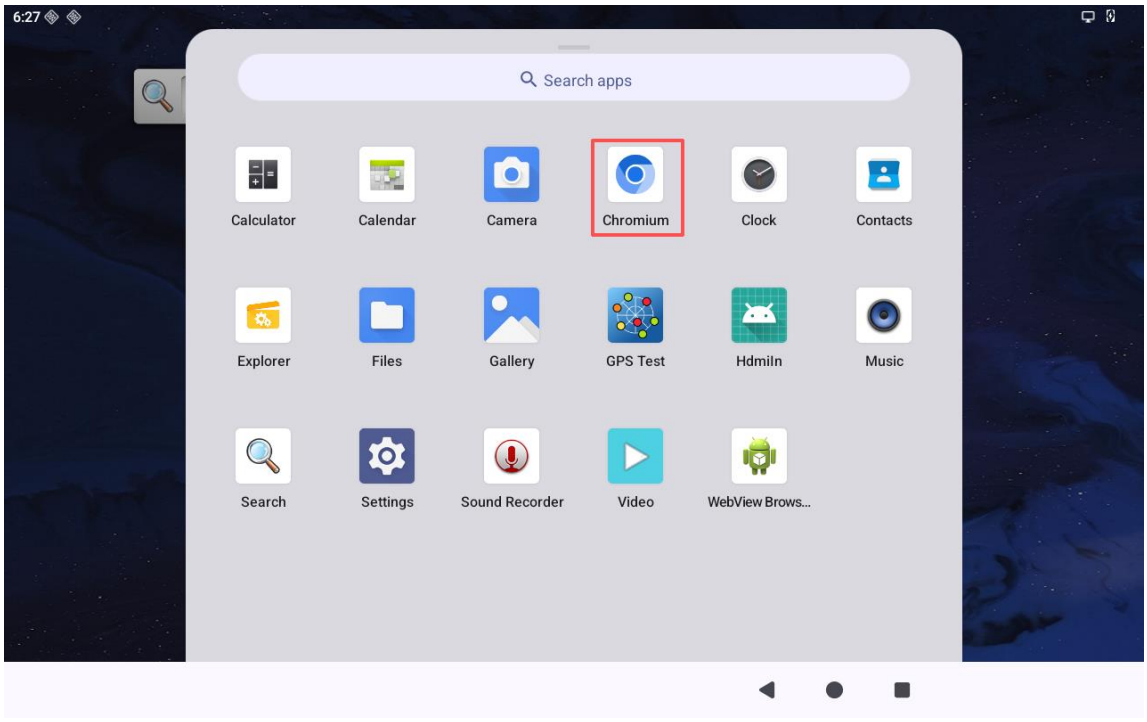


6.10.1 4G Test

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 wwan0
```

```
console:/ $ ifconfig wwan0
wwan0     Link encap:Ethernet  HWaddr 42:b7:bd:c9:cb:fb  Driver qmi_wwan_q
          inet addr:10.4.216.208  Mask:255.255.240.0
          inet6 addr: 240e:47c:3000:233:2c53:ff5a:917a:835e/64 Scope: Global
          inet6 addr: fe80::40b7:bdff:fec9:cbfb/64 Scope: Link
          inet6 addr: 240e:47c:3000:233:40b7:bdff:fec9:cbfb/64 Scope: Global
          UP RUNNING NOARP  MTU:1500  Metric:1
          RX packets:163 errors:0 dropped:0 overruns:0 frame:0
          TX packets:293 errors:0 dropped:0 overruns:0 carrier:0
```

(2) Network connection test.

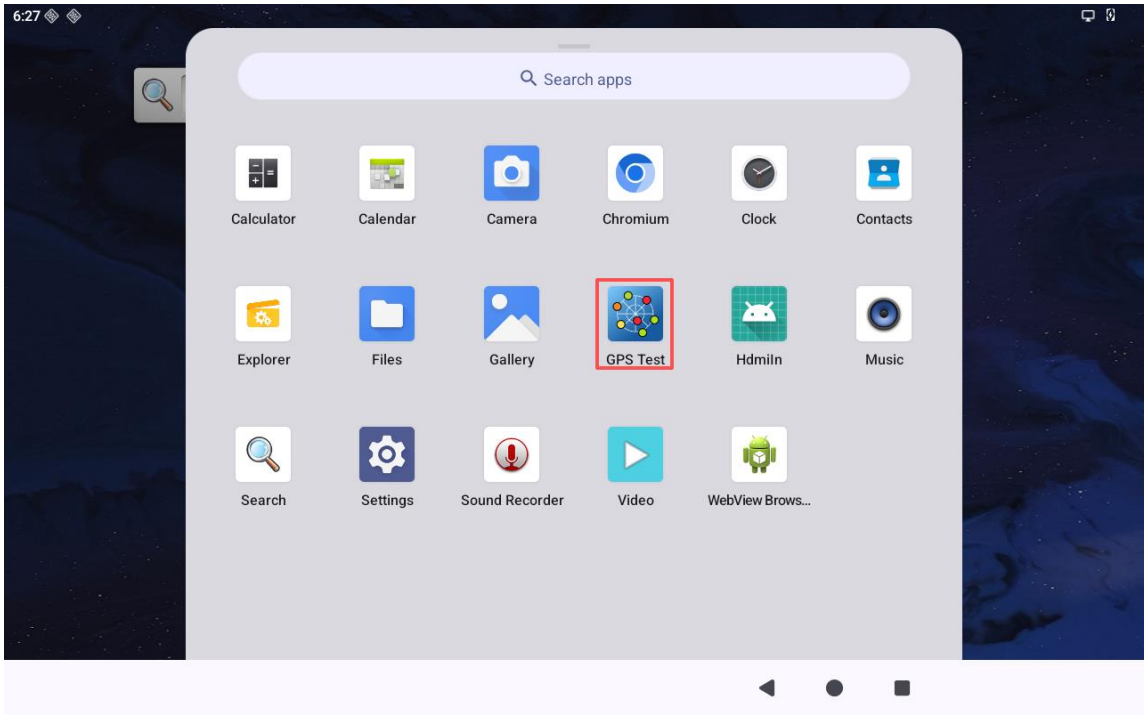
```
$ su
# ping -I wwan0 www.armdesigner.com
```

```
PING www.armdesigner.com (67.222.54.196) from 10.4.216.208 wwan0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=47 time=614 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=47 time=676 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=47 time=300 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=47 time=275 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=47 time=556 ms
^C
--- www.armdesigner.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 8720ms
rtt min/avg/max/mdev = 275.926/484.844/676.664/165.041 ms
```

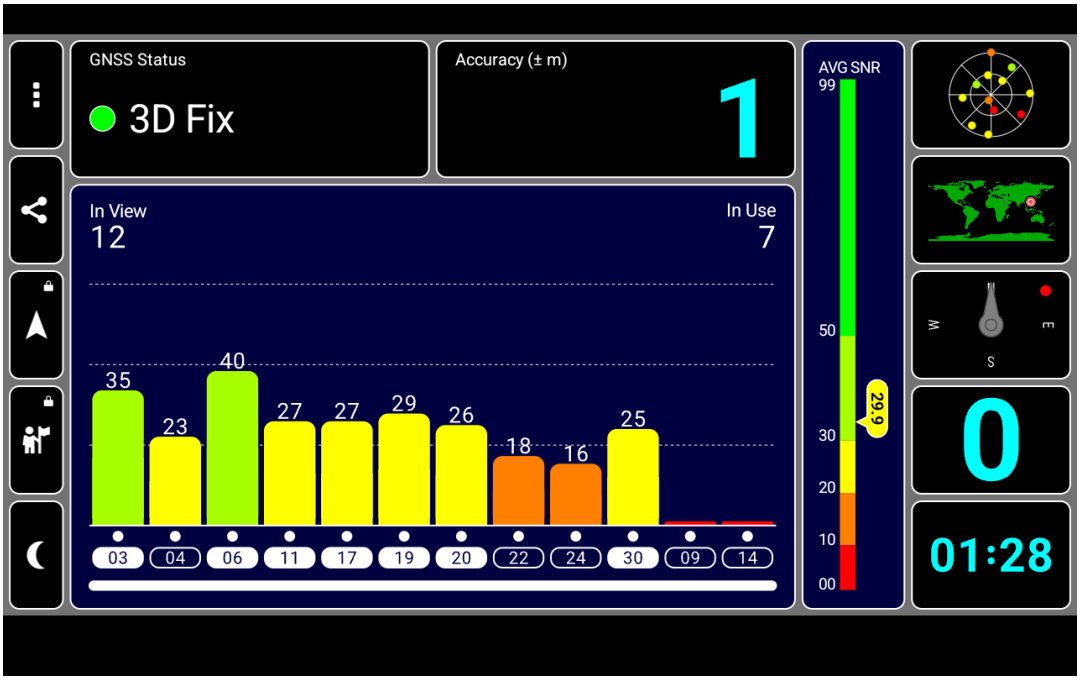
6.10.2 GPS Test

GPS positioning needs to be tested in open outdoor areas.

Step 1: Open **GPS Test** APP.

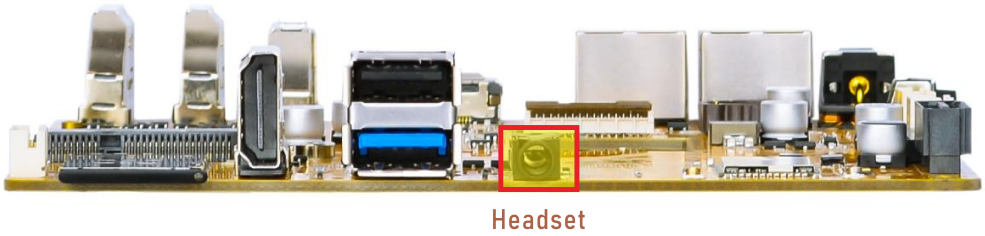


Step 2: Wait a moment to successfully obtain the location.

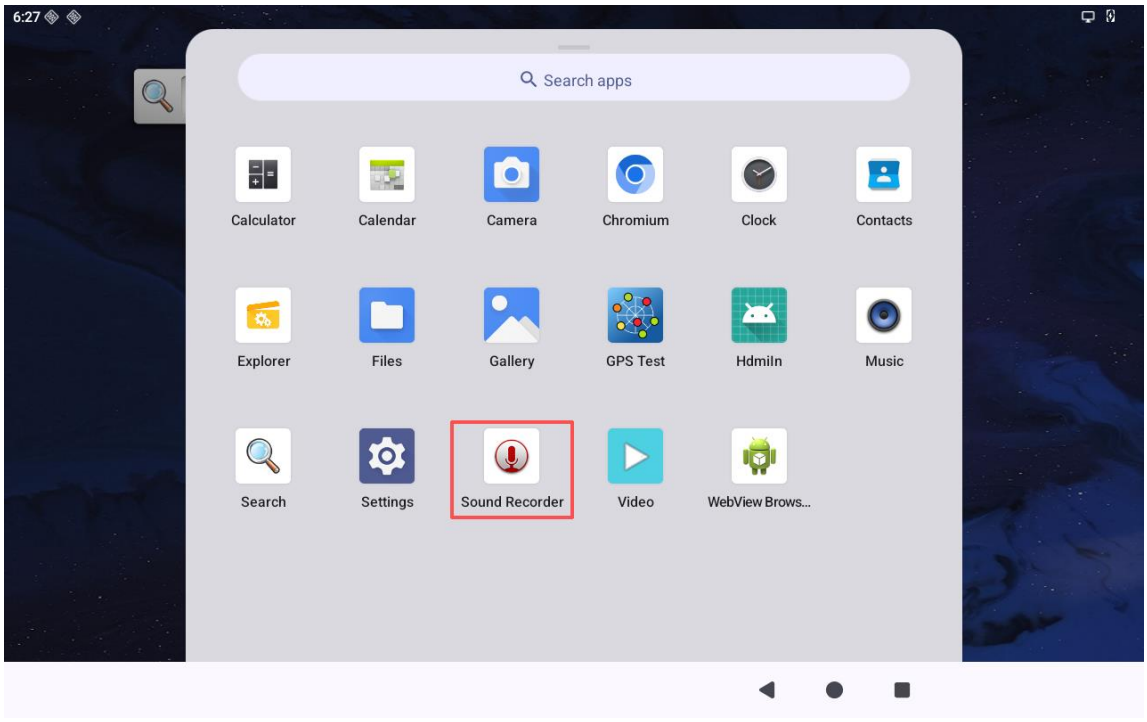


6.11 Headset

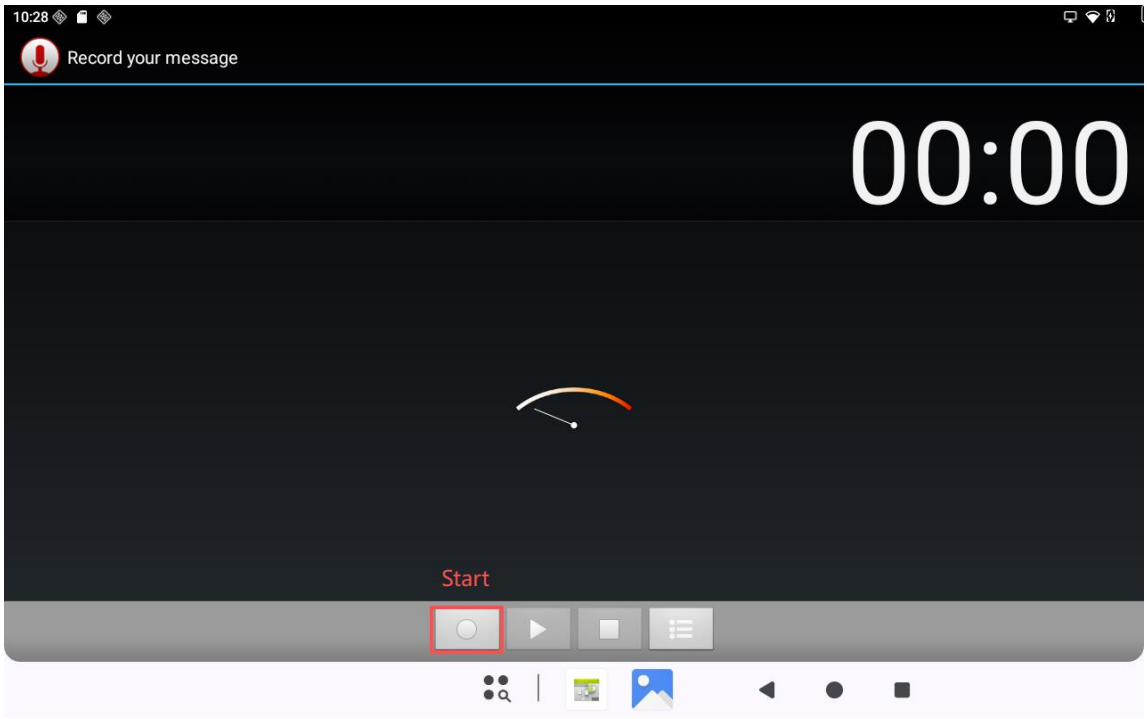
Step 1: Plug the headset into the headset jack.



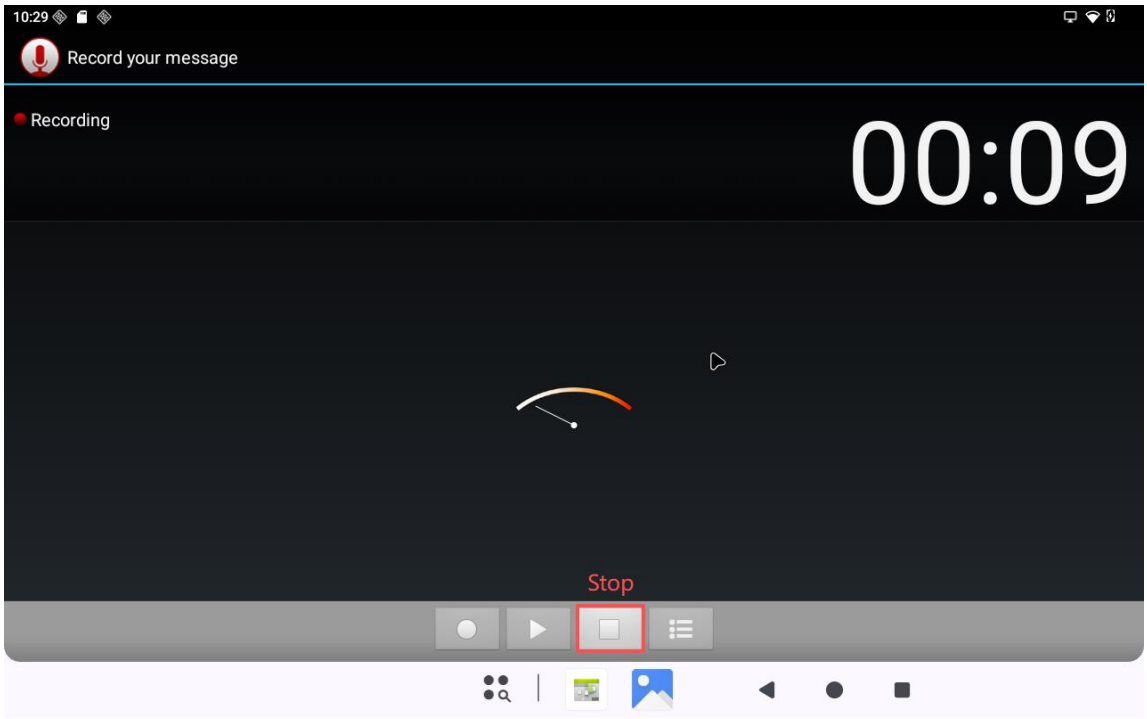
Step 2: Open the **Sound Recorder** app.



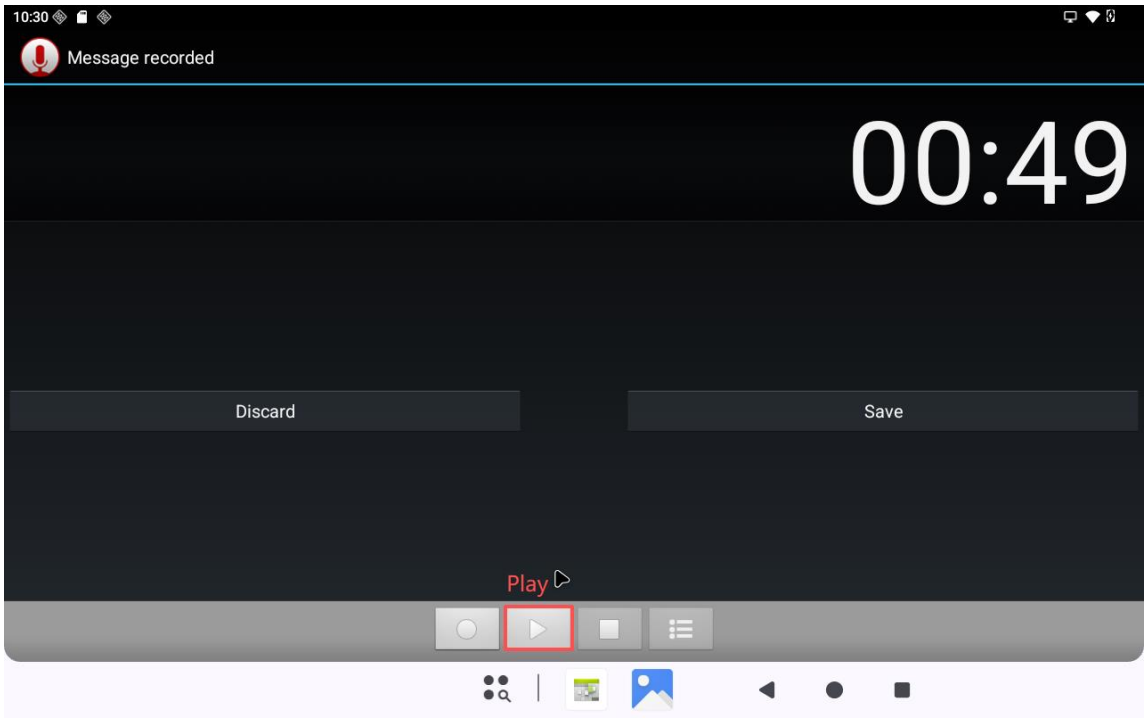
Step 3: Click the following button to start recording.



Step 4: Click the following button to stop recording.

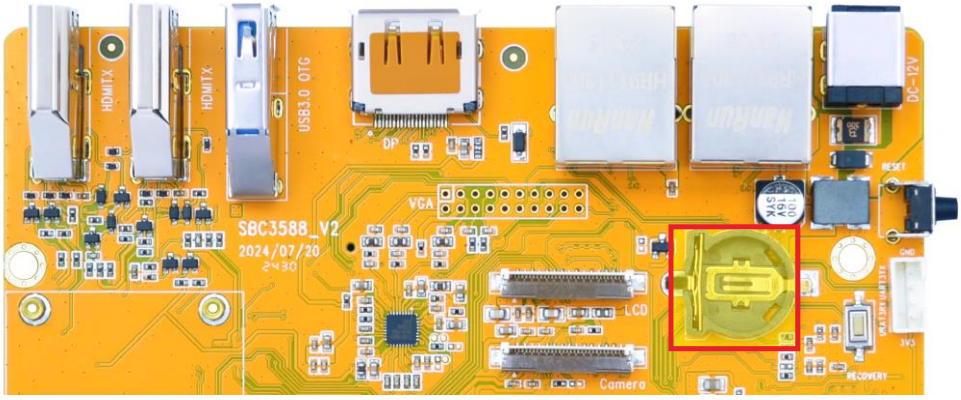


Step 5: Click the following button to play back the recording.



6.12 RTC

Step 1: Install the coin cell battery.



RTC

Step 2: Set the system time.

```
$ su
# date -s "2026-08-05 18:48:00"
```

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

```
# hwclock -w
```

Step 4: Display the current hardware clock time.

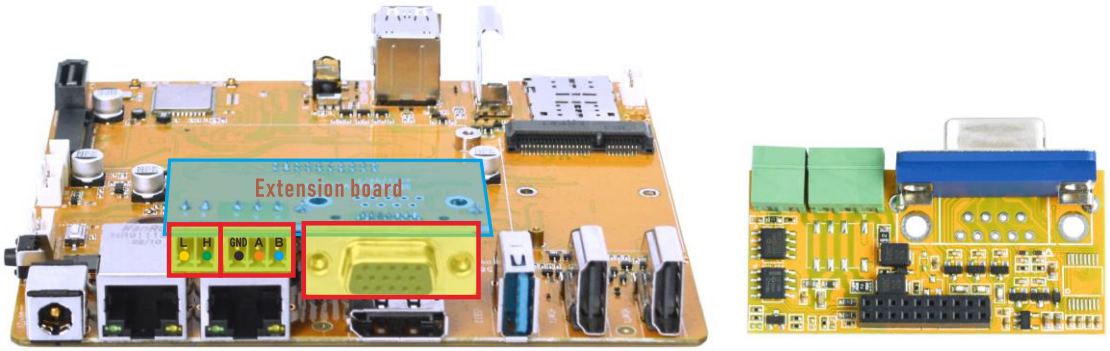
```
# hwclock
```

```
1|console:/ $ su
console:/ # date -s "2026-08-05 18:48:00"
Wed Aug  5 18:48:00 GMT 2026
console:/ # hwclock -w
console:/ # hwclock
2026-08-05 18:48:02+0000
console:/ # hwclock
2026-08-05 18:48:13+0000
console:/ # hwclock
2026-08-05 18:48:16+0000
```

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

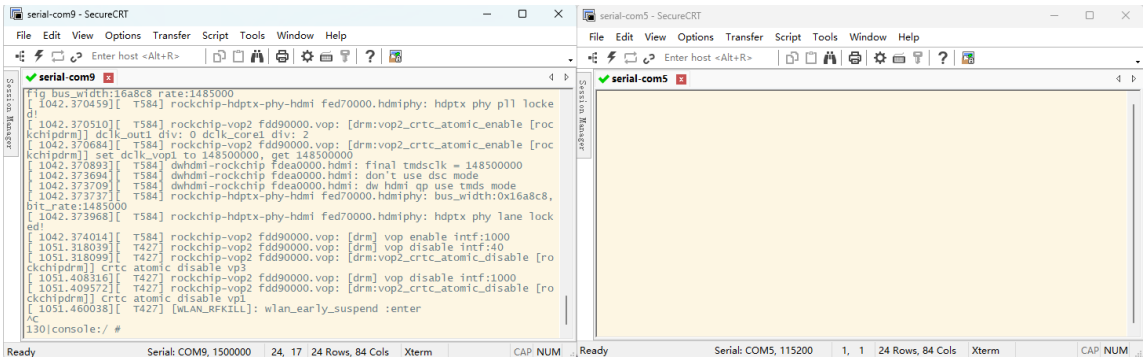
```
1|console:/ $ su
console:/ # hwclock
2026-08-05 18:49:15+0000
console:/ # hwclock
2026-08-05 18:49:19+0000
console:/ # hwclock
2026-08-05 18:49:21+0000
```

6.13 RS485



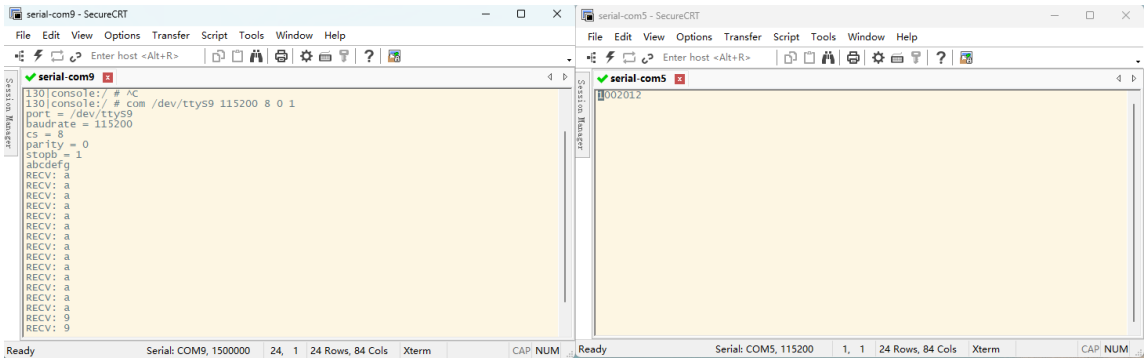
Step 1: As shown in the diagram, connect the RS485 test tool to the development board.

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.

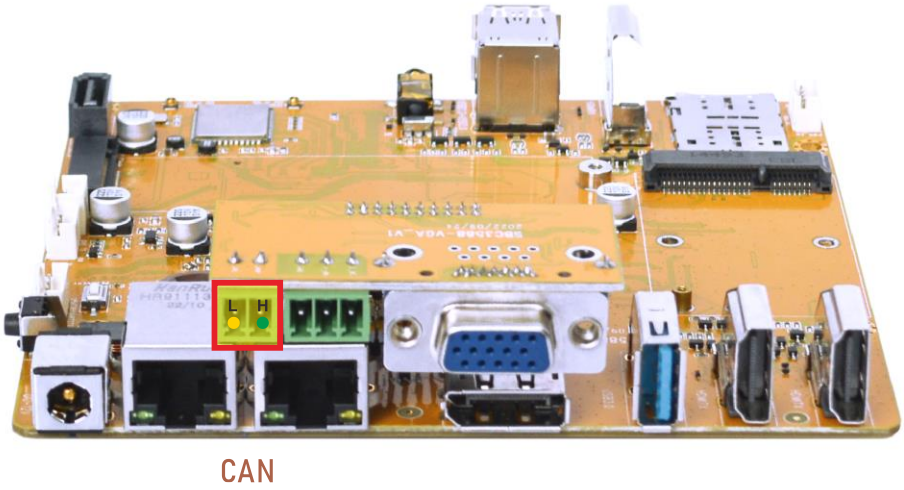


Step 3: Execute the following command on the board to test the RS485 transmission and reception functionality.

```
# com /dev/ttyS9 115200 8 0 1
```

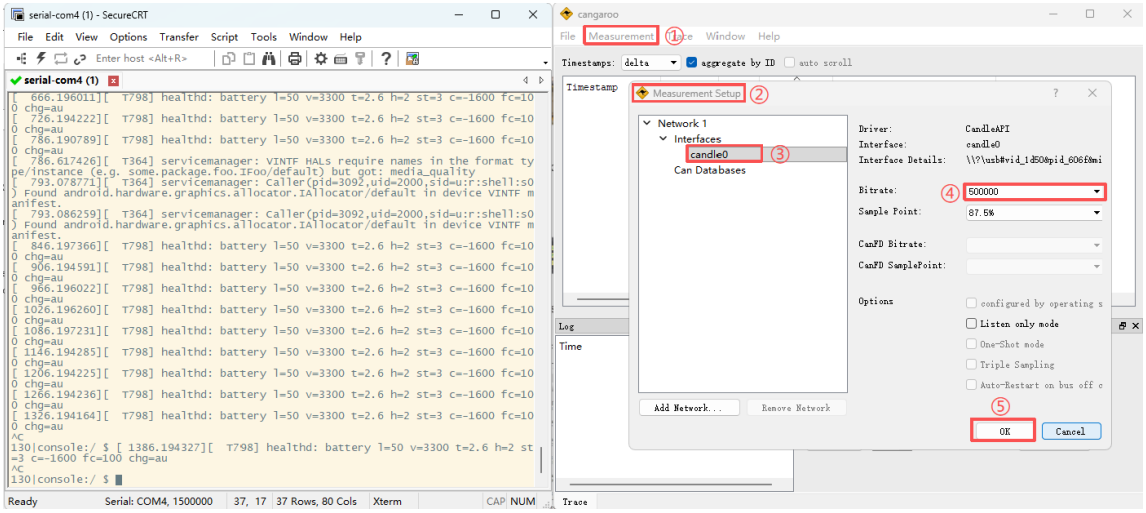


6.14 CAN



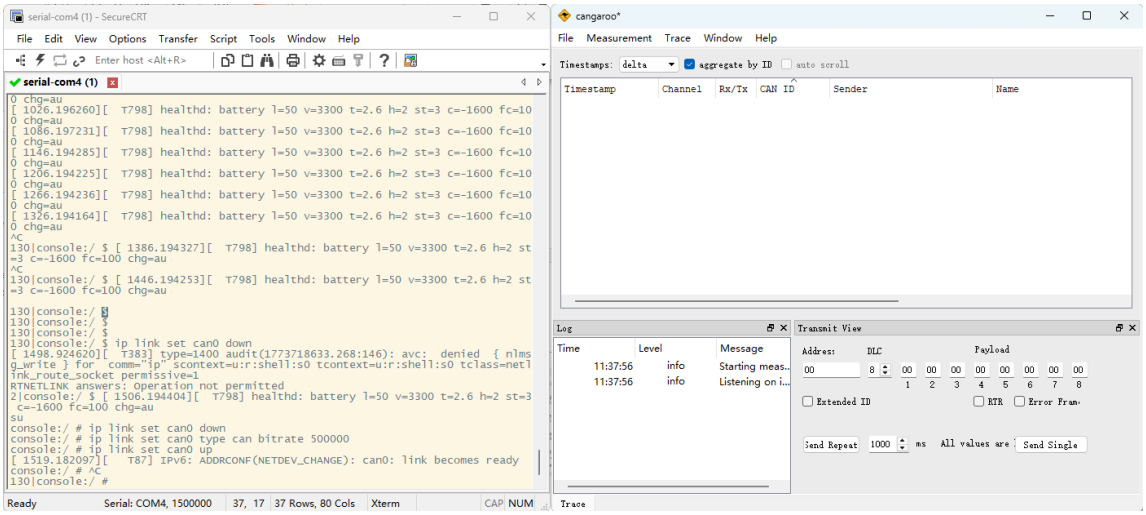
Step 1: Connect the CAN test tool to the board as shown in the diagram below.

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



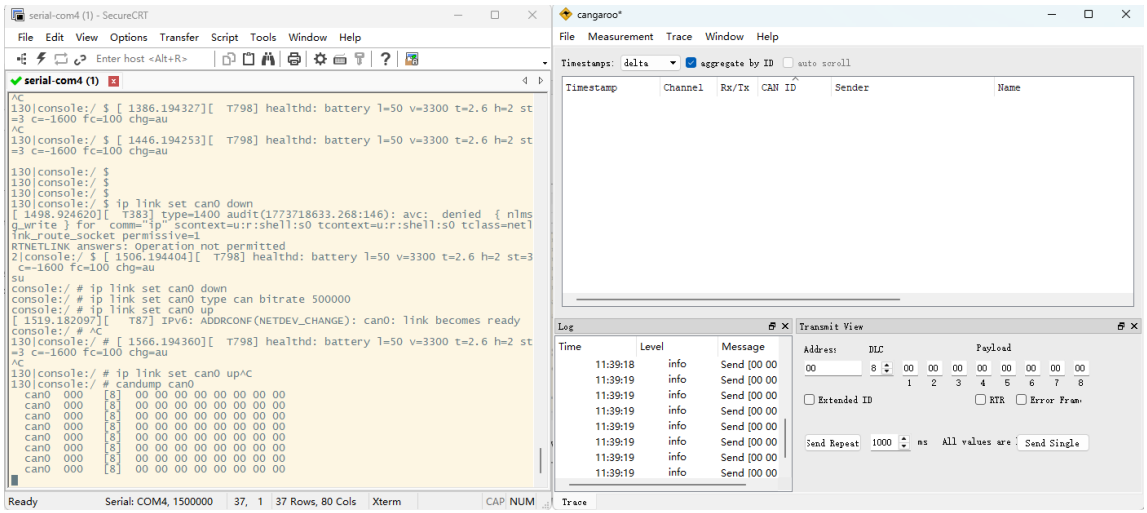
Step 3: Configure and activate the CAN network, setting the bitrate to 500000.

```
$ su
# ip link set can0 down
# ip link set can0 type can bitrate 500000
# ip link set can0 up
```



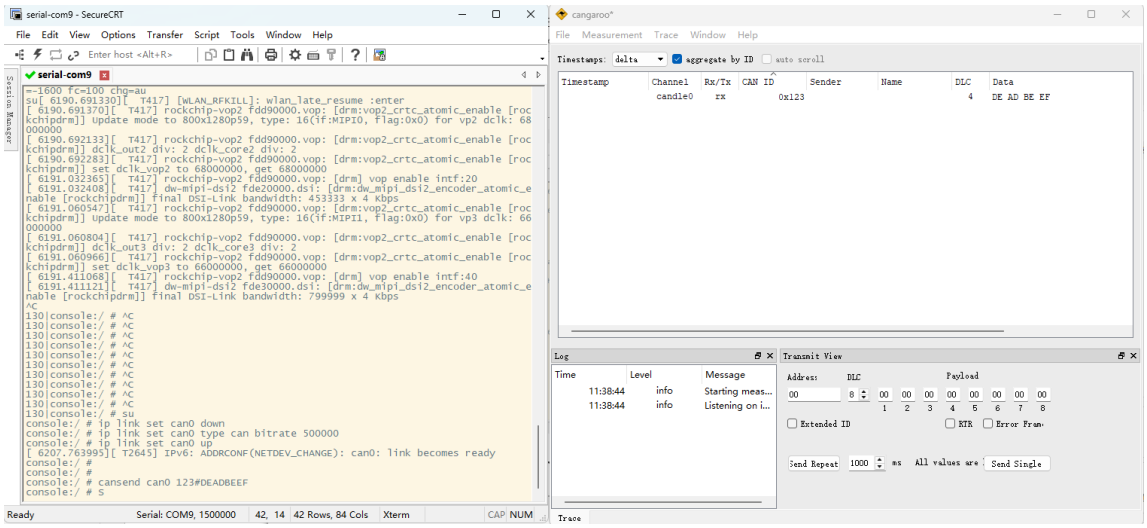
Step 4: Configure CAN as the receiver.

```
# candump can0
```



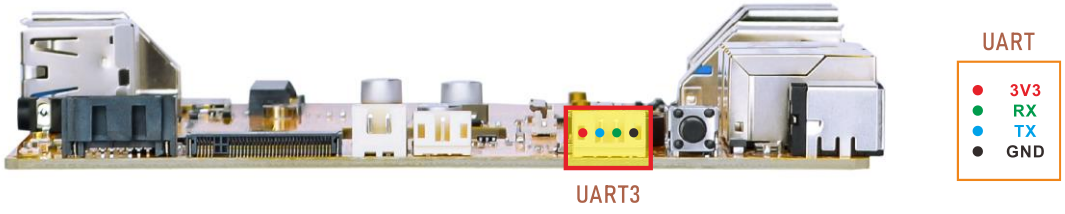
Step 5: Configure CAN as the sender.

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



6.15 RS232

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



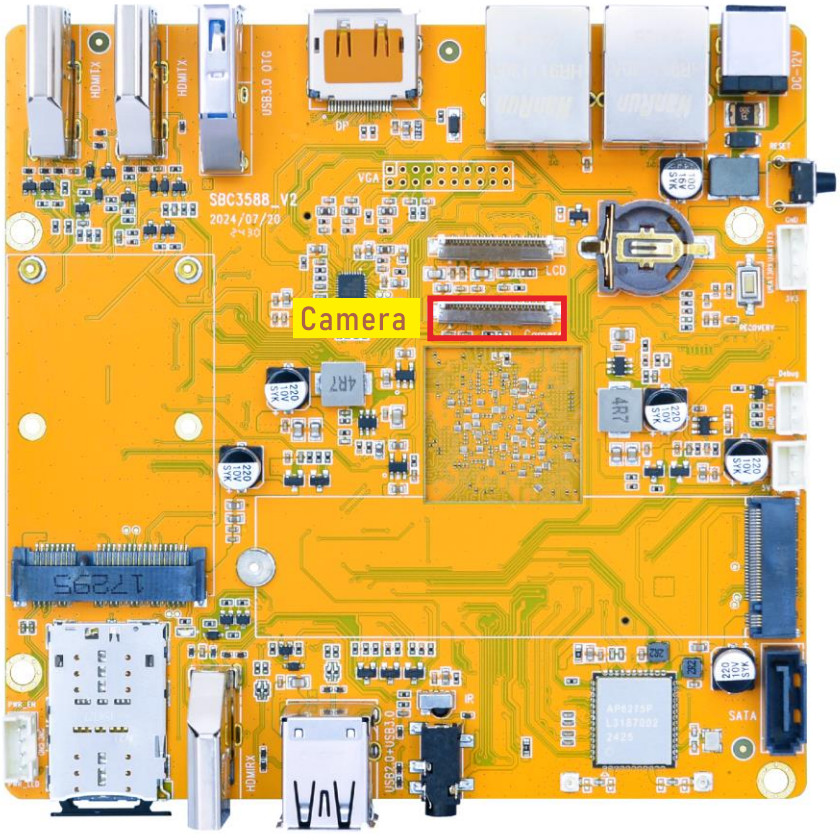
Step 2: RS232 test.

```
# com /dev/ttyS3 115200 8 0 1
```

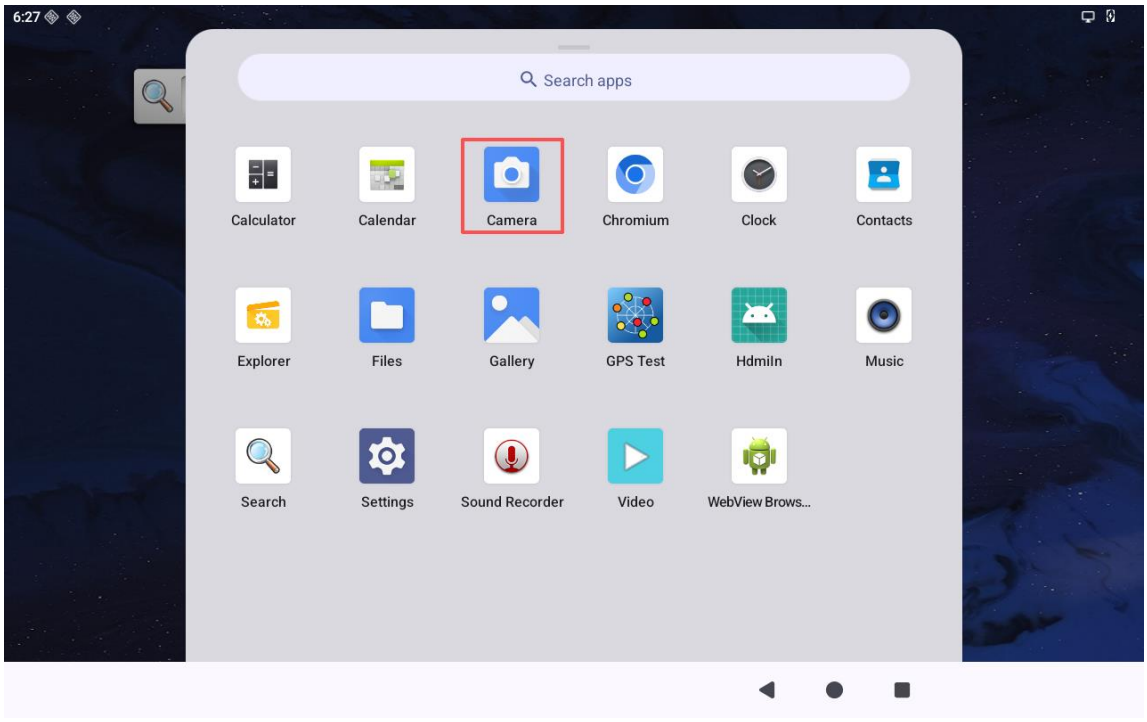
```
130|console:/ $ com /dev/ttyS3 115200 8 0 1
port = /dev/ttyS3
baudrate = 115200
cs = 8
parity = 0
stopb = 1
[  299.814407][ T3633] of_dma_request_slave_channel: dma-names property of node '/serial@feb60000' missing
or empty
[  299.814424][ T3633] dw-apb-uart feb60000.serial: failed to request DMA

[  306.952097][  T804] healthd: battery l=50 v=3300 t=2.6 h=2 st=3 c=-1600 fc=100 chg=au
RECV:
abcdefg
RECV: abcdefg
1233
RECV: 1233
4564
RECV: 4564
qoqpqp
RECV: qoqpqp
```

6.16 Camera

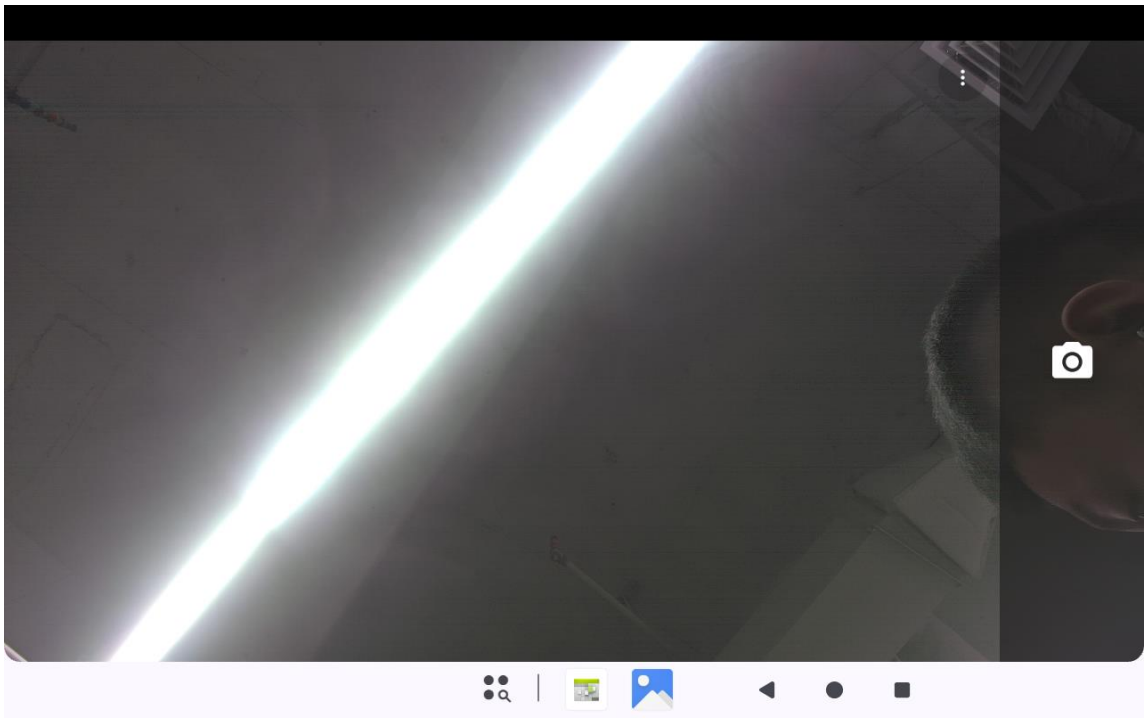
Step 1: Connect the camera (IMX415), and then power on the board.



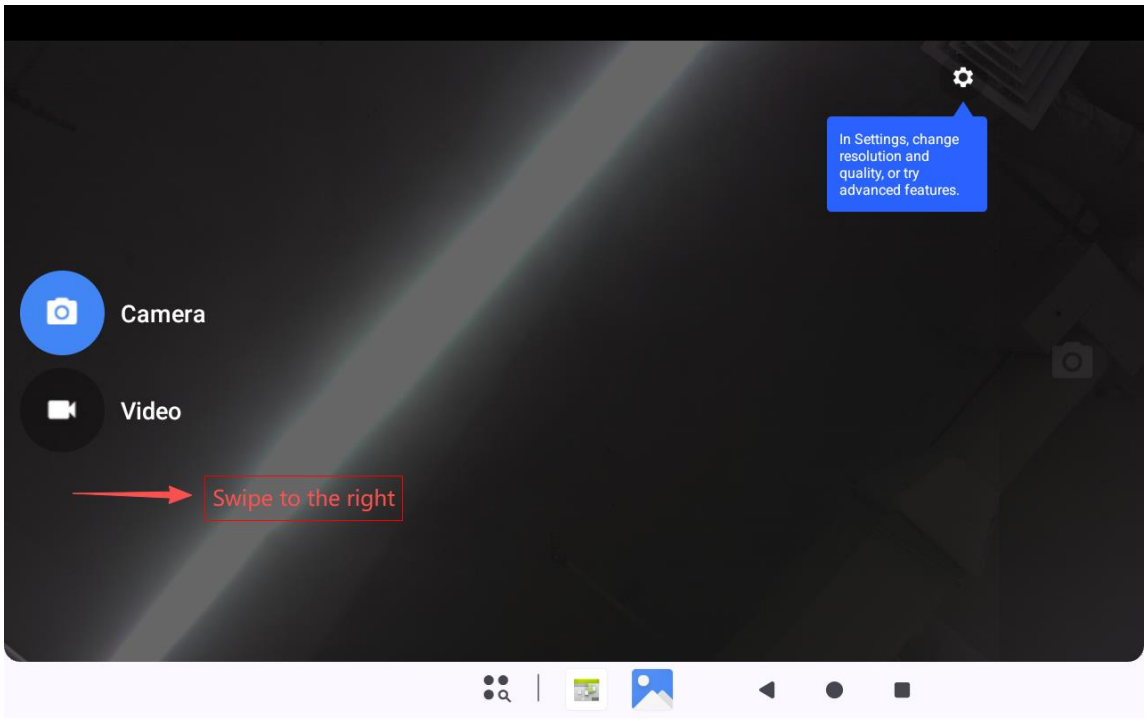
Step 2: Open the **Camera** app.

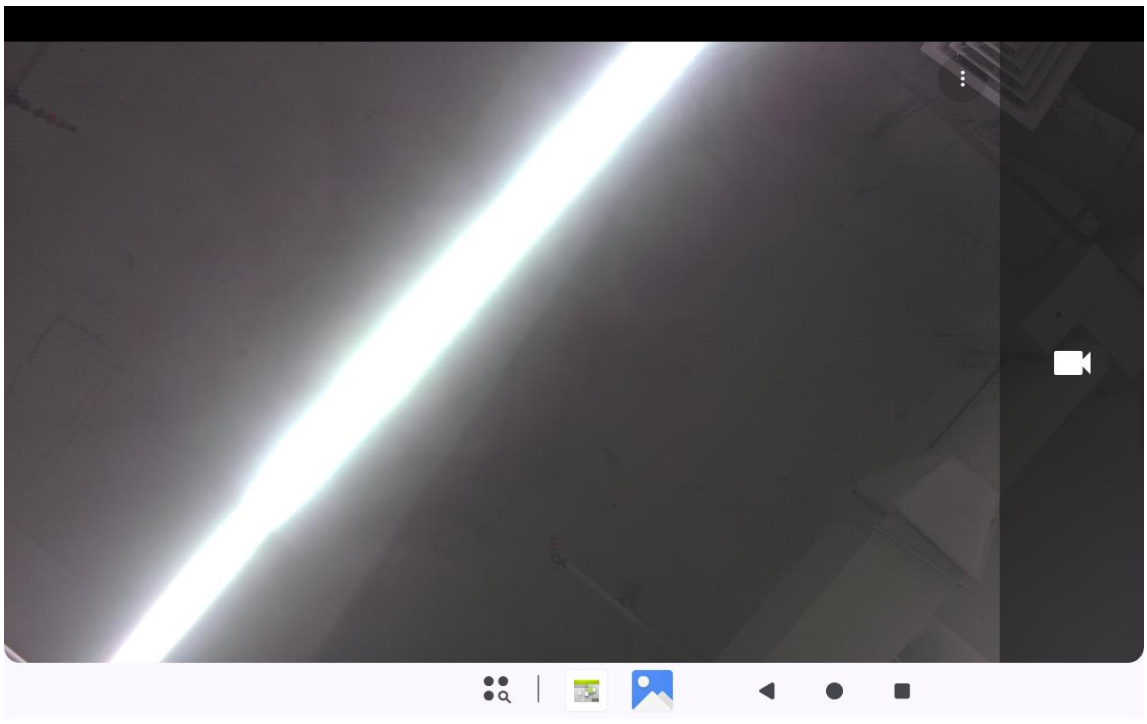


Step 3: Default photo mode.

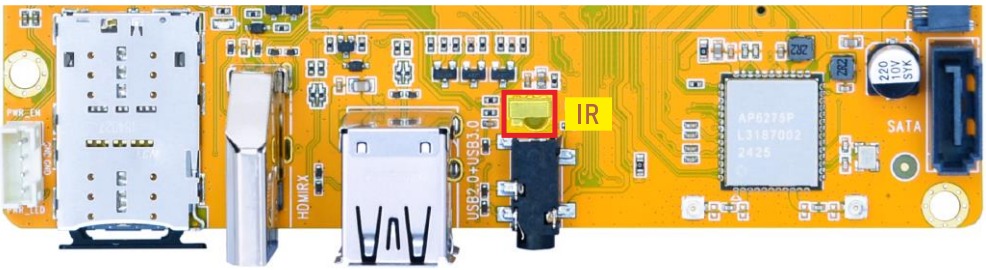


Step 4: Switch to video recording mode.


In Settings, change resolution and quality, or try advanced features.
Camera
Video
Swipe to the right



6.17 IR



Step 1: Open IR debugging print.

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

Step 2: When pressing a button on the remote control towards the IR receiver, the key value will be printed to the log.

```
130|console:/ $ su
console:/ #
console:/ # echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
console:/ # [   32.644976][    C1] USERCODE=0xd728
[   32.672066][    C1] RMC_GETDATA=e8
[   32.910822][    C1] USERCODE=0xd728
[   32.937925][    C1] RMC_GETDATA=f8
[   33.288221][    C1] USERCODE=0xd728
[   33.315412][    C1] RMC_GETDATA=f8
[   33.776692][    C1] USERCODE=0xd728
[   33.803890][    C1] RMC_GETDATA=e9
[   34.111948][    C1] USERCODE=0xd728
[   34.139093][    C1] RMC_GETDATA=e1
[   34.539303][    C1] USERCODE=0xd728
[   34.566447][    C1] RMC_GETDATA=f2
[   34.887858][    C1] USERCODE=0xd728
[   34.914956][    C1] RMC_GETDATA=f8
```