SBC3568 Android14 User Manual

V1.0

Boardcon Embedded Designer

Overview

The content of this document is intended solely for the SBC3568 development board, aiming to help users quickly understand, apply, and test the SBC3568 development board.

System Support

Development Board	Android14
Mini3568 RV2 SBC3568 V3	Y

Revision History

Version	Date	Author	Revision History
V1.0	2025-05-22	Liu Yuan	Initial version

Disclaimer

The information in this manual is for reference only. While Boardcon strives to ensure its accuracy, no guarantees are made regarding its completeness or correctness. All content is subject to change without prior notice. Boardcon reserves the right to revise the content of this manual without prior notification.

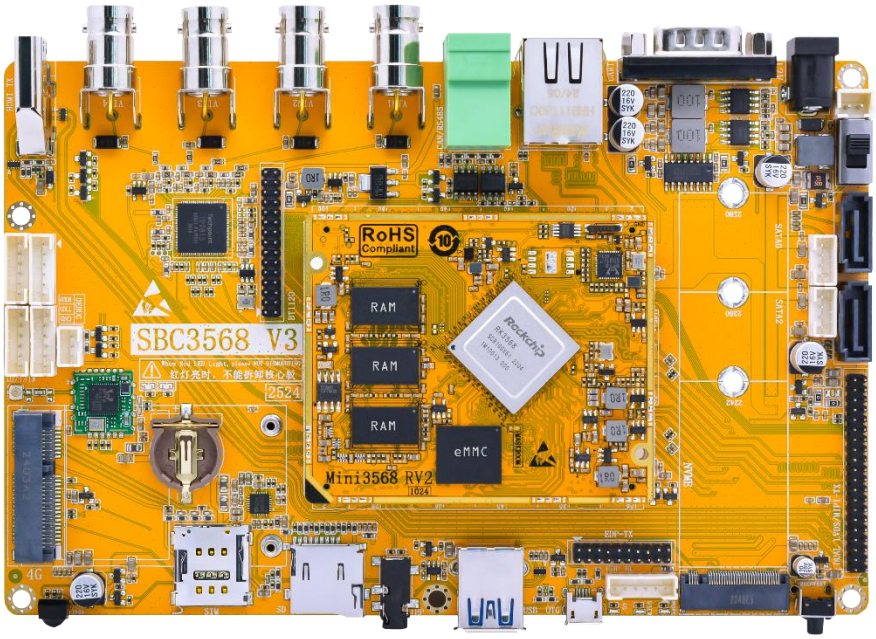
Boardcon embedded design limited
2508 Haofang Tianji Plaza, 11008 Beihuan Avenue, Nanshan District,
Shenzhen, Guangdong, China. 518051
URL: www.armdesigner.com | www.boardcon.com
Email: market@armdesigner.com
Technical Support Inquiries: support@armdesigner.com
Tel: +86-755-26481393 | +86-755-27571591

Content

1.Introduction

1.1 Overview

The SBC3568 development board is equipped with the RK3568 quad-core ARM Cortex-A55 processor. This processor delivers high performance with low power consumption, supporting nearly all H.264 decoding at 4K@60fps, H.265 decoding at 4K@60fps, H.264/H.265 encoding at 1080p@60fps, and high-quality JPEG encoding/decoding. The RK3568 is ideal for personal mobile internet devices and AIoT applications.



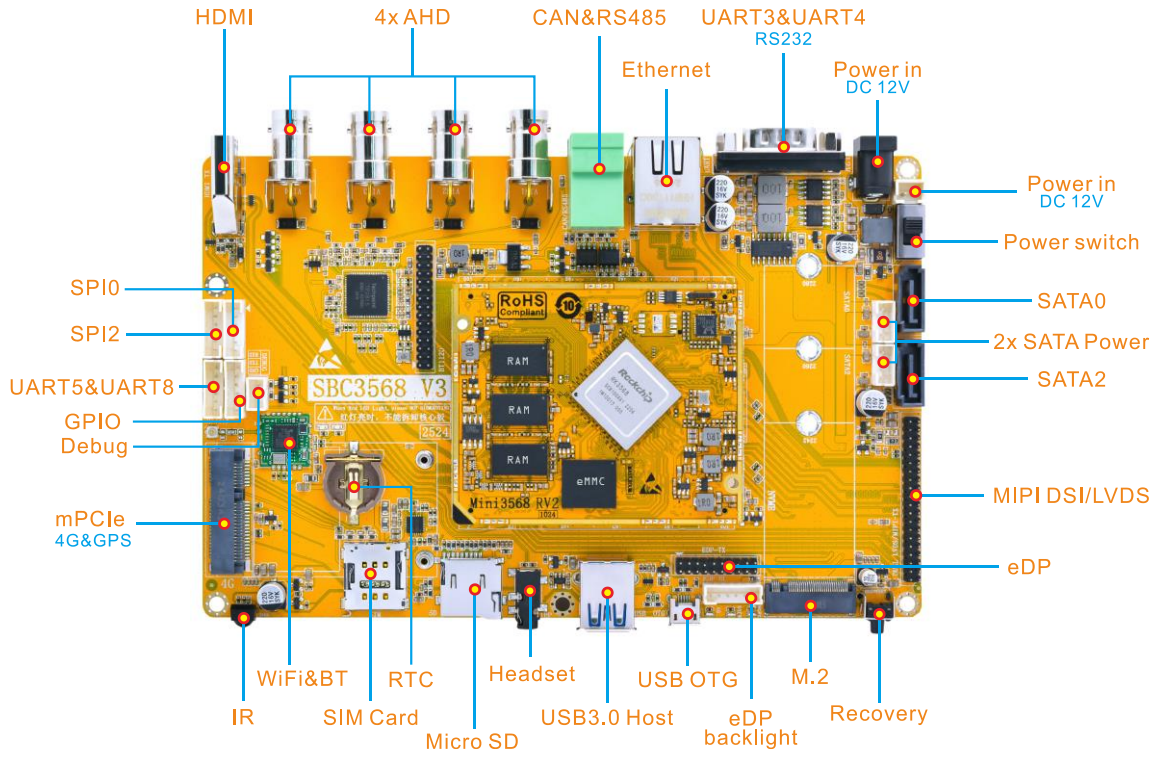
1.2 Product Parameters

Basic Parameters	
SOC	• RK3568
CPU	• Quad-core 64-bit ARM Cortex-A55@ up to 2.0GHz
GPU	• OpenCL 2.0 • OpenGLES 1.1/2.0/3.2 • Vulkan 1.1

NPU		• 1 TOPS AI computing power • Supports int8/int16/fp16/bfp16 hybrid MAC operations
Video	Decoder	• Support 4K@60fps H.265/H.264/VP9 video decoding • Support 1080P@60fps VP8/VC1/MPEG-4,2,1 video decoding • Support 720x576@60fps H.263 video decoding
	Encoder	• Support 1080P@60fps H.265/H.264 video encoding • Support YUV/RGB video source with rotation and mirror
RAM		• 2GB LPDDR4X (up to 8GB)
ROM		• 8GB eMMC (up to 64GB)
Support system		Android, Debian, Buildroot
Hardware Parameters		
Extended Storage		• Support 2x SATA • Support M.2 SSD • Support MicroSD Card
Display		• Support HDMI TX 4K@60fps display • Support MIPI DSI or LVDS display
Audio		• Support HDMI TX audio output • Support Headphone output/input
USB		• Support USB3.0
Network		• Support Gigabit Ethernet • Support WIFI/BT module • Support 4G module
Camera		• Support 4x AHD
Peripheral communication		• Support CAN • Support RS485 • Support 2xUART • Support 2xRS232

	• Support 2xSPI
Other parameters	Support Debug, IR, RTC, OTG.
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	60.0mm x 45.0mm
Motherboard dimensions	135.0mm x 100.0mm

1.3 Hardware Interface Introduction



Interface parameters	
Power in DC 12V	12V DC power input interface
UART3&UART4 RS232	RS232 communication interface
Ethernet	Gigabit Ethernet RJ45 interface
CAN&RS485	CAN&RS485 communication interface
4xAHD	AHD camera interface
HDMI	HDMI2.0 TX interface
BT1120	BT1120 interface
SPI0	SPI0 interface
SPI2	SPI2 interface
UART5&UART8	UART5&UART8, TTL level interface
GPIO	GPIO interface
Debug	UART2, debug the serial port
mPCIe 4G module	4G&GPS module interface
IR	Infrared receiver
WIFI&BT	Realtek RTL8723DU module
SIM card	SIM card port
RTC	RTC coin cell connector
Micro SD	MicroSD card slot
Headset	Headset output/input
USB3.0	USB3.0 Host interface
USB OTG	OTG download interface
eDP backlight	eDP backlight interface
M.2	M.2 SSD interface
Recovery	Recovery key
eDP	eDP display interface
MIPI DSI/LVDS	LVDS/MIPI display interface

SATA2	SATA2 interface
2x SATA Power	SATA 5V Power Connector
SATA0	SATA0 interface
Power switch	Power rocker switch
Power in DC 12V	12V DC power input 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):

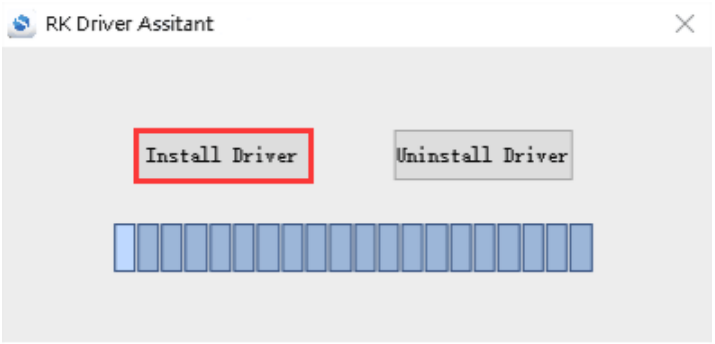
Number	Driver name	Driver	Use
1	RK Driver Assitant	DriverInstall.exe	OTG USB driver installation assitant
2	CH9102x	SETUP.EXE	Serial port debugging driver
3	Serial Terminal Tool	SecureCRT.exe	Debugging tool

2.1 Install RK Driver Assitant

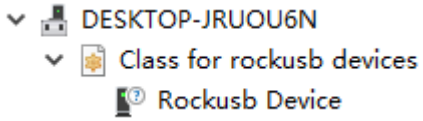
Step 1: Open *DriverAssitant_v5.1.1/DriverInstall.exe*.

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

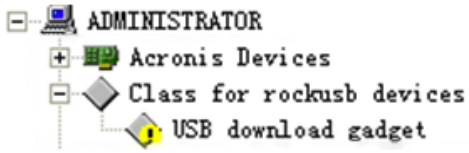
Step 3: Click button **"Install Driver"** to install.



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

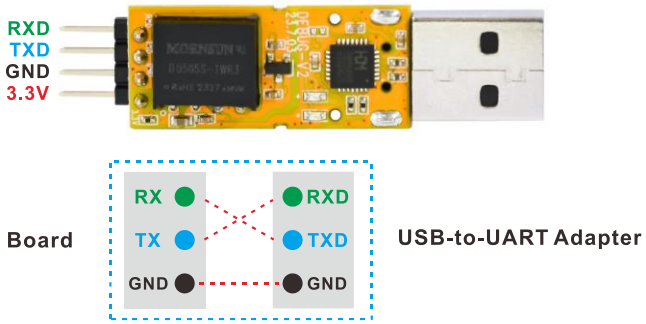


Step 6: The WINDOW will pop up found New Hardware Wizard dialog box, choose to install from the specified location, and then select: *DriverAssitant_v5.1.1/ADBDriver.*

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



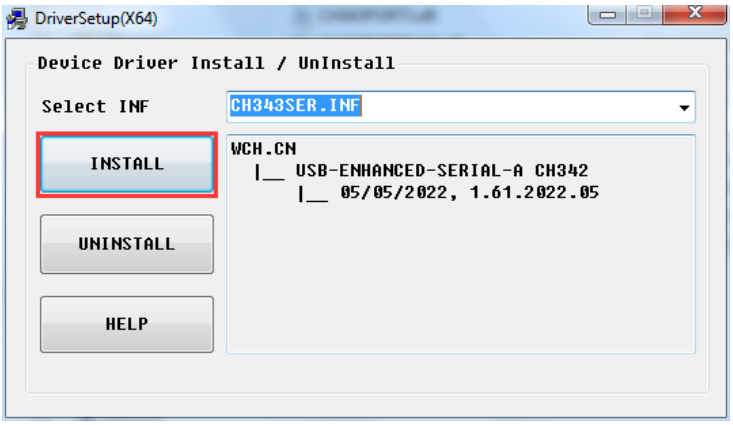
Pin	Connection Description
RXD	Receive, connect to TX pin of the board.
TXD	Transmit, connect to RX pin of the board.
GND	Ground, connect to GND pin of the board.
3V3	No need to connect.

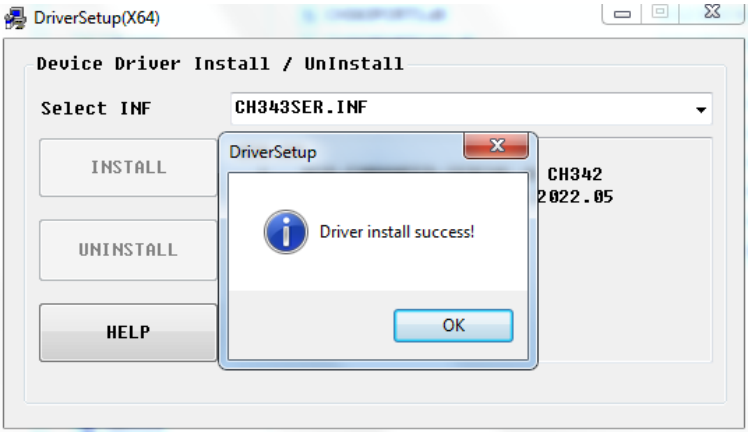
2.2.2 Install Driver

Step 1: Plug the CH9102X Module to the PC

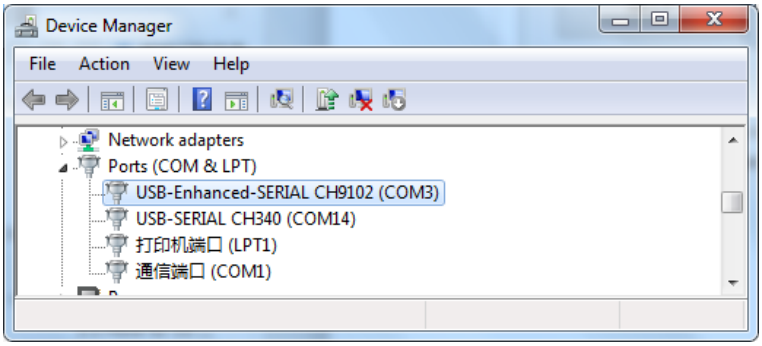
Step 2: Unzip *CH343SER.ZIP* on Windows.

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.

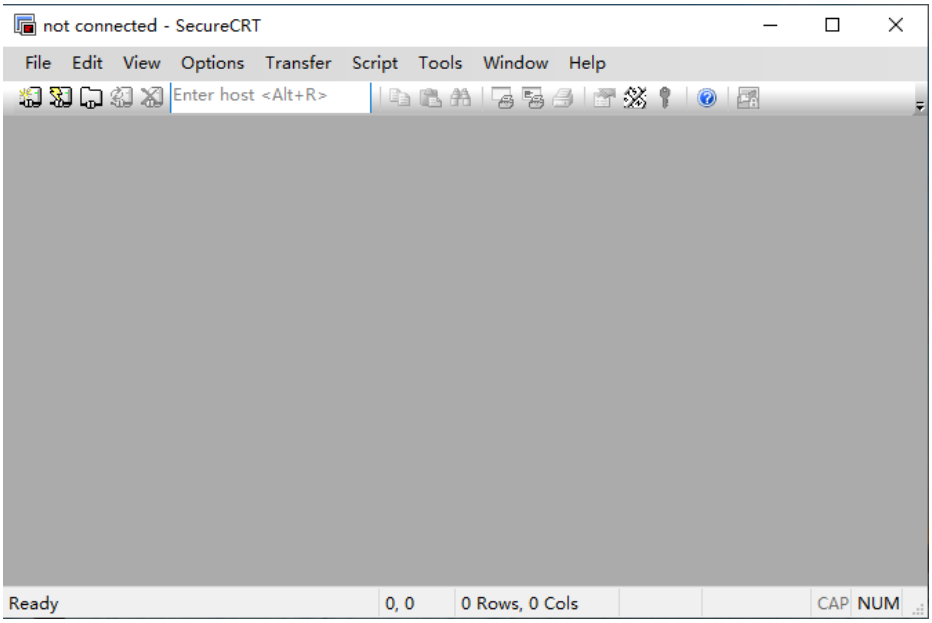


2.3 Install Serial Terminal Tool

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

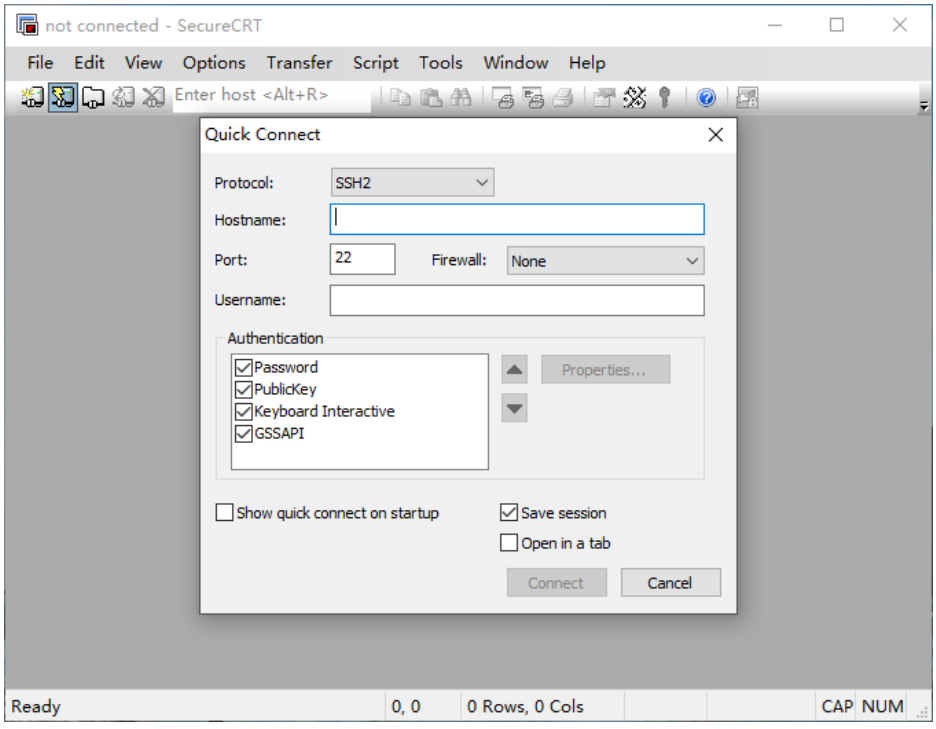
Step 1: Unzip *Platform/SecureCRT.rar* on PC.

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

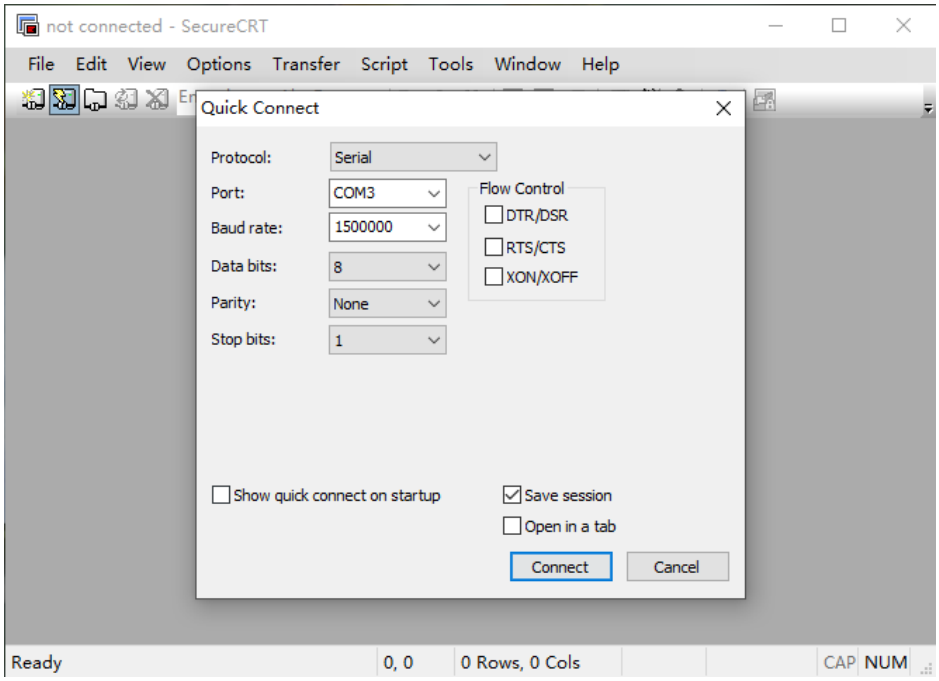


Step 3: Confirm that the CH9102X driver has been installed and the CH9102X module is connecting to the PC.

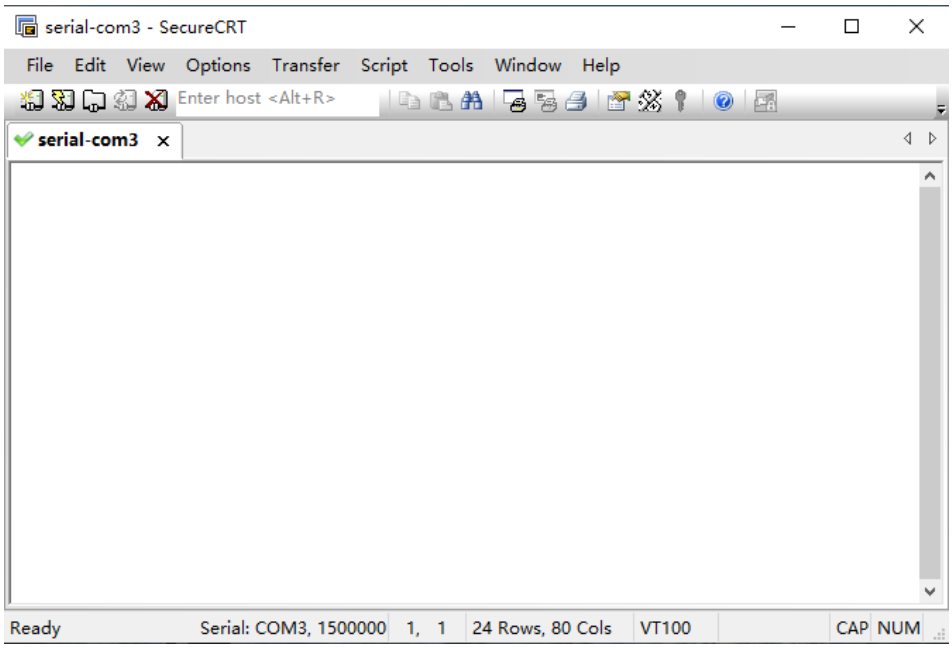
Step 4: Click the **"Quick Connect"** button to go to the Quick Connect configuration screen.



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

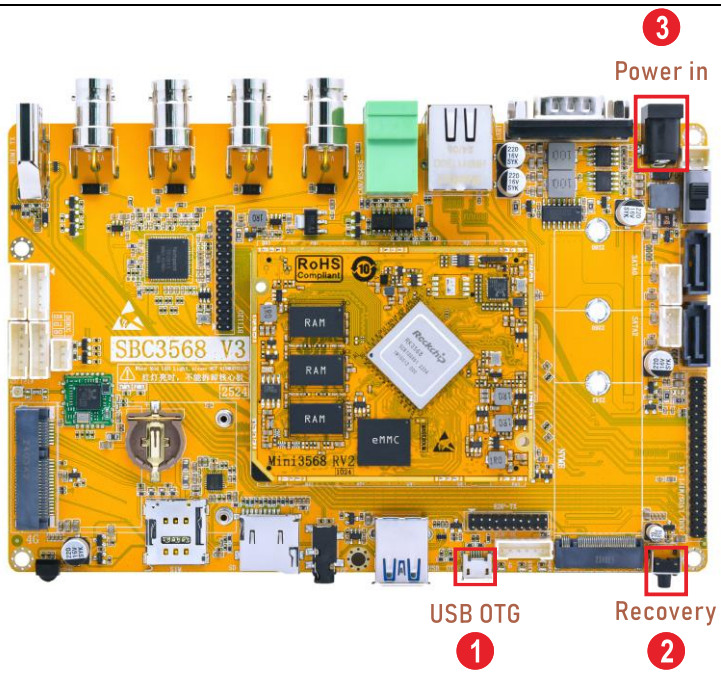
3.1.1.1 Hardware

Step 1: Disconnect the power adapter.

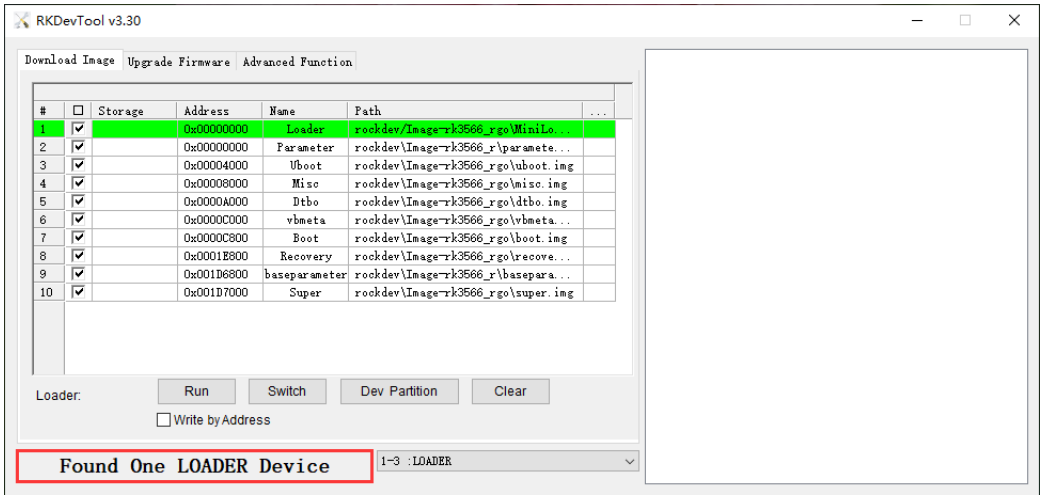
Step 2: Connect one end of the Micro cable to the host and the other end to the development board.

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 Micro cable, execute the following command in the serial debug terminal or adb shell.

```
# reboot loader
```

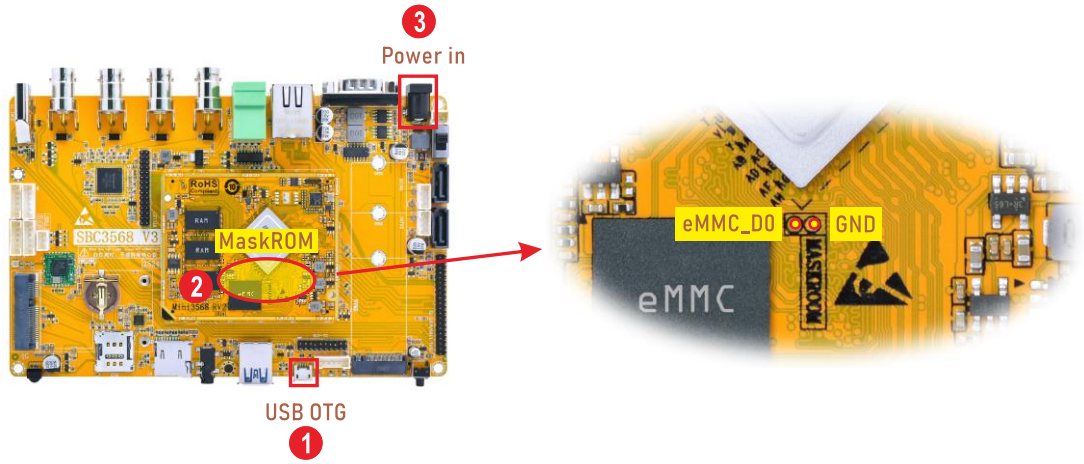
3.1.2 How to Enter MaskRom Mode

Step 1: Disconnect the power adapter.

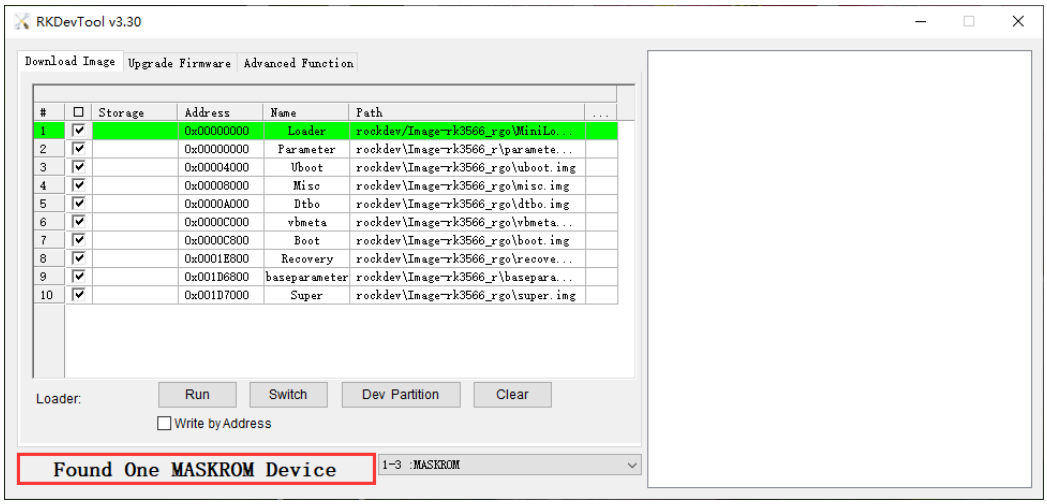
Step 2: Connect one end of the Micro cable to the host and the other end to the

development board.

Step 3: Use tweezers to short the two test points on the Mini3568.



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



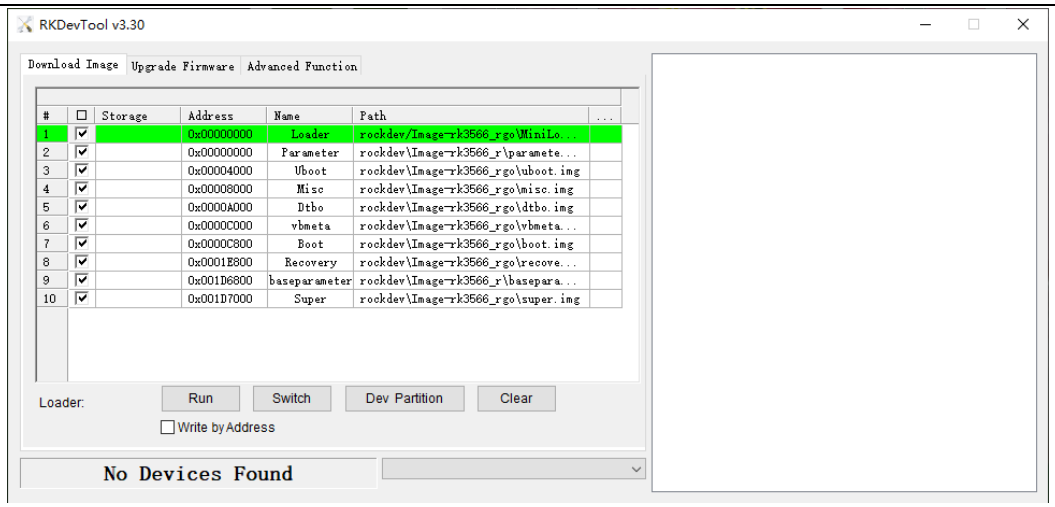
3.2 Burn firmware

Environment: Windows OS (Operating System).

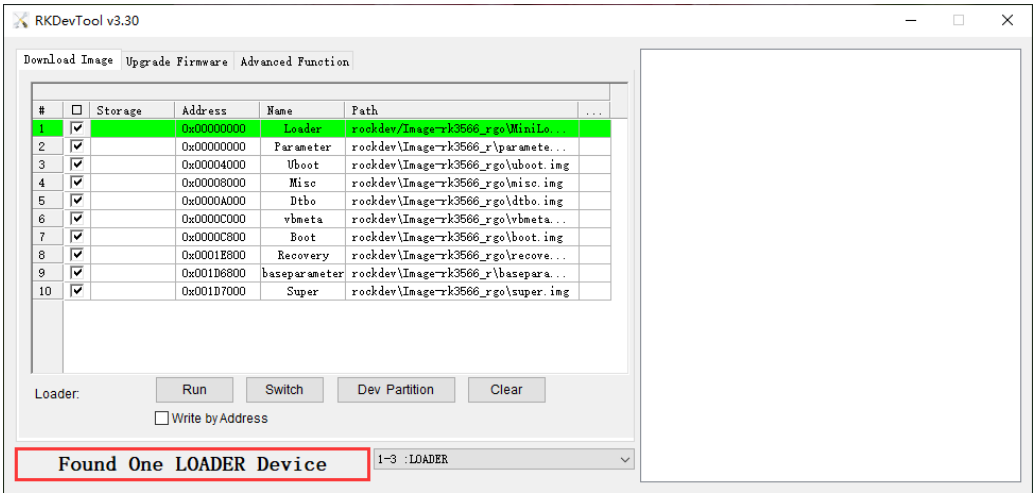
3.2.1 Burn Update.img Firmware

Step 1: Unzip *RKDevTool.rar* on Windows.

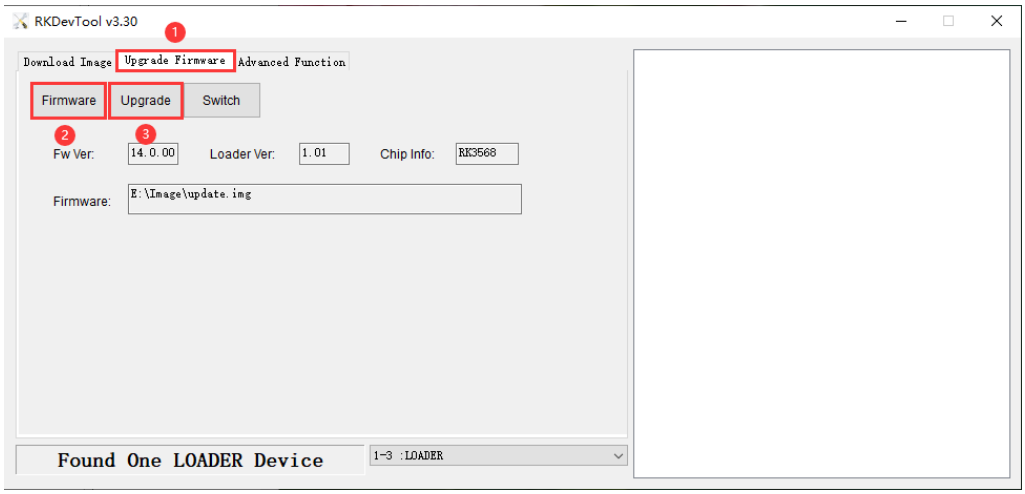
Step 2: Open *RKDevTool\RKDevTool_Release\RKDevTool.exe*.



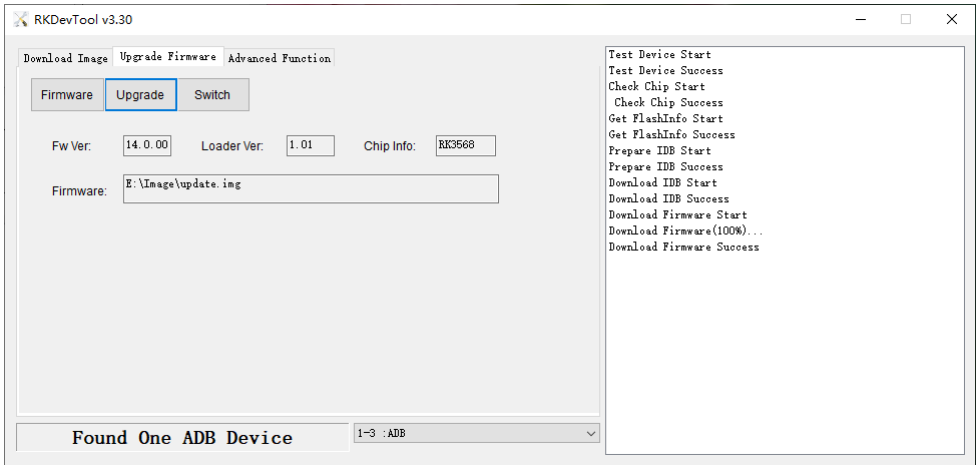
Step 3: Switch to **loader mode**. (*How to Enter Loader Mode*)



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



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



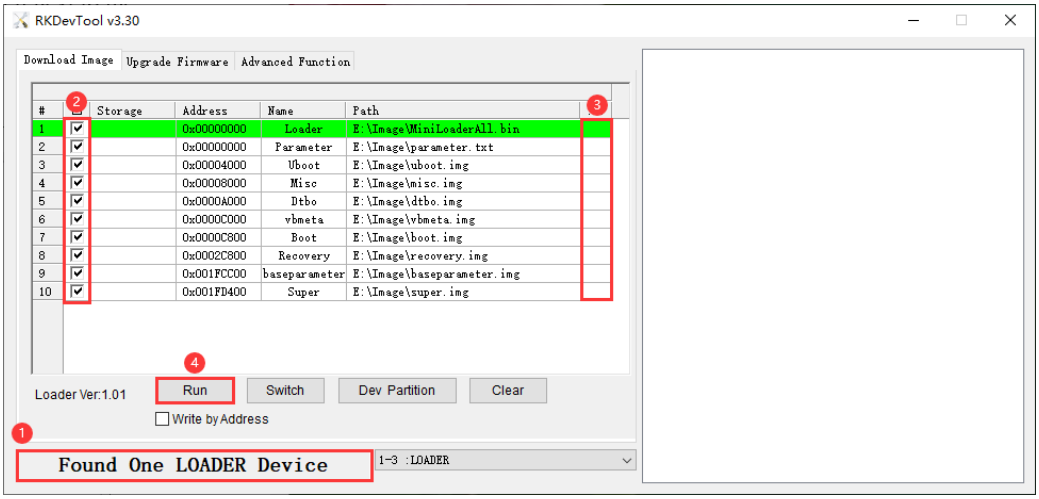
3.2.2 Burn Split Firmware

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

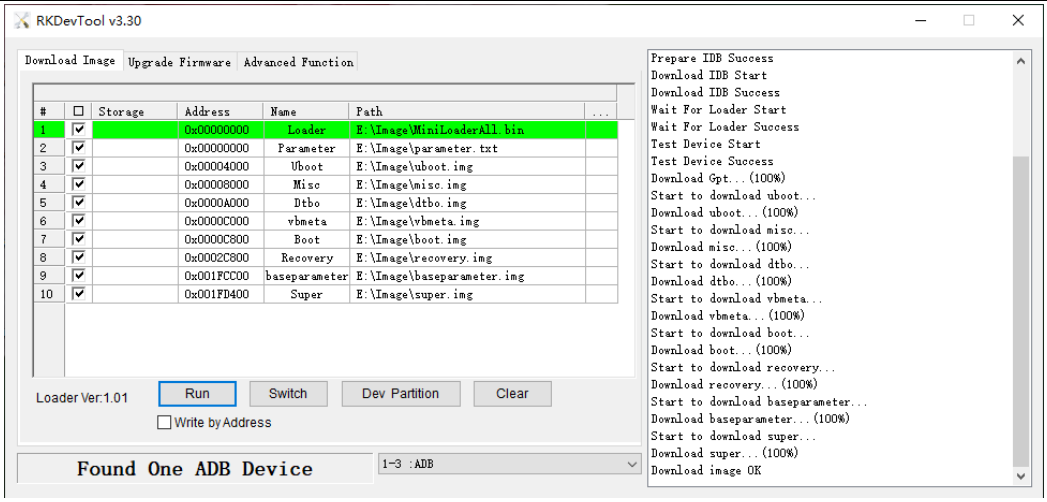
Step 2: Check the partitions to be flashed, multiple partitions can be selected.

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 24.04 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 24.04

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 24.04	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

Step 1: Unzip the Source

To extract the source files, execute the following commands:

```
$ tar xvf rk3568-em_sbc-android14-rkr*.tar.bz2
$ cd rk3568_android14-rkr6/
```

Step 2: Configure the Compiled Platform

To configure the board, execute:

```
$ source build/envsetup.sh
$ lunch sbc3568_u-userdebug
```

Step 3: One key compiling command

To compile uboot, kernel, and Android, execute the following command:

```
$ ./build.sh -AUCKu
```

```
./build.sh -UKAup
( WHERE: -U = build uboot
     -C = build kernel with Clang
     -K = build kernel
     -A = build android
     -p = will build packaging in IMAGE
     -o = build OTA package
     -u = build update.img
     -v = build android with 'user' or 'userdebug'
     -d = huild kernel dts name
     -V = build version
     -J = build jobs
------------you can use according to the requirement, no need to record
uboot/kernel compiling commands------------------
)
============================================================
Please remember to set the environment variable before using the one key
compiling command, and select the platform to be compiled, for example:
source build/envsetup.sh
lunch rk3588_t-userdebug
============================================================
```

Images and **update.img** are generated in *rockdev/Image-rk3568_u* directory.

Other compiling instruction:

Android14 cannot directly flash **kernel.img** and **resource.img**. Using ./build.sh -AK will recompile the entire Android system, which is time-consuming. If the user only needs to compile the kernel, it is recommended to use the following command:

Export clang to the environment:

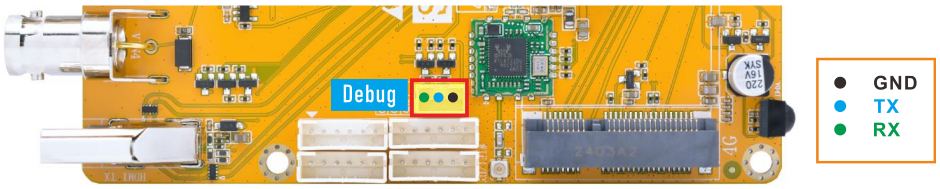
```
$ cd kernel-6.1
$ export PATH=../prebuilts/clang/host/linux-x86/clang-r487747c/bin:$PATH
$ alias msk='make CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1'
```

```
$ msk ARCH=arm64 rockchip_defconfig android-14.config rk356x.config
$ msk ARCH=arm64 BOOT_IMG=../rockdev/Image-sbc3568_u/boot.img boardcon-sbc3568-
rv2-v3.img -j32
```

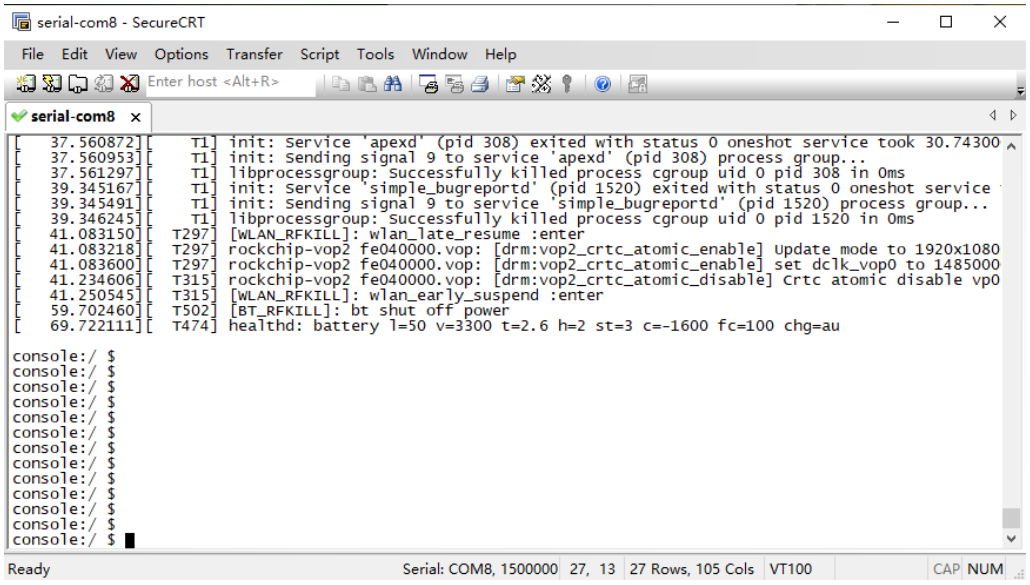
The user can flash **boot.img** under the catalogue of *kernel-6.1/* directly to boot position of machine after compiling, and please load the partition table（**parameter.txt**） when flashing, for fear of flashing to the wrong place.

6.Android14 Test

6.1 Serial Terminal

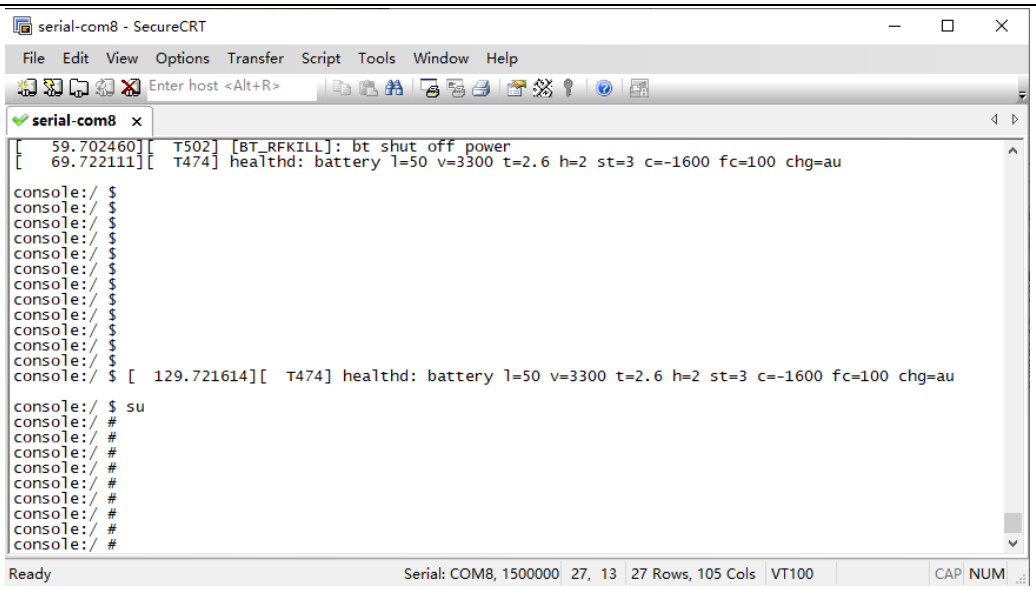


Connect the board and PC with USB Serial cable, then power on, the terminal will output boot information. The default baudrate is 1500000.



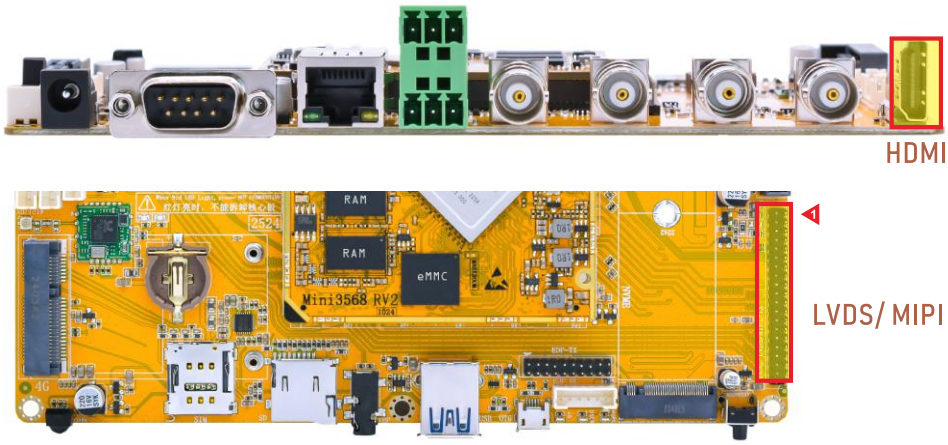
Execute the following command to switch to root mode:

```
$ su
```

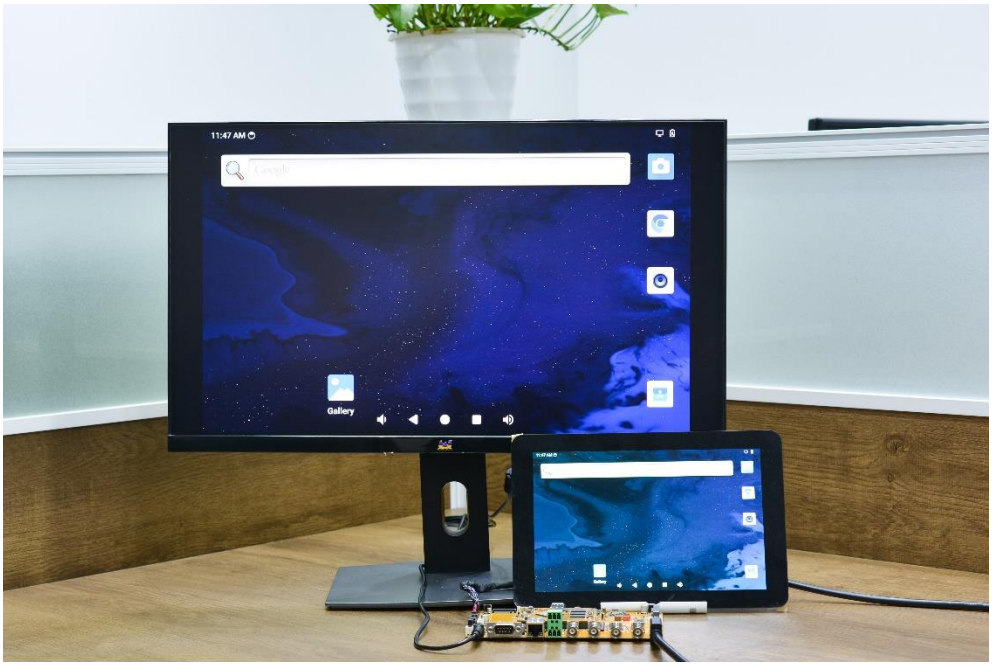


6.2 Display

The SBC3568 Android14 system supports simultaneous display on both LVDS and HDMI screens with mirrored output.

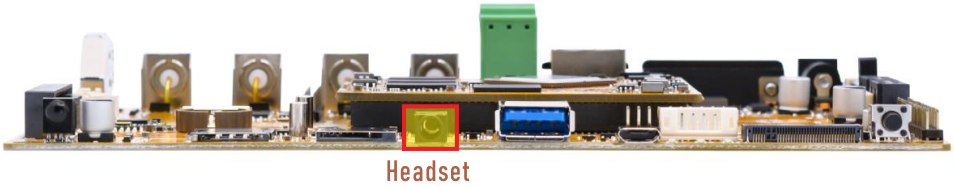


The display effect diagram is as follows:

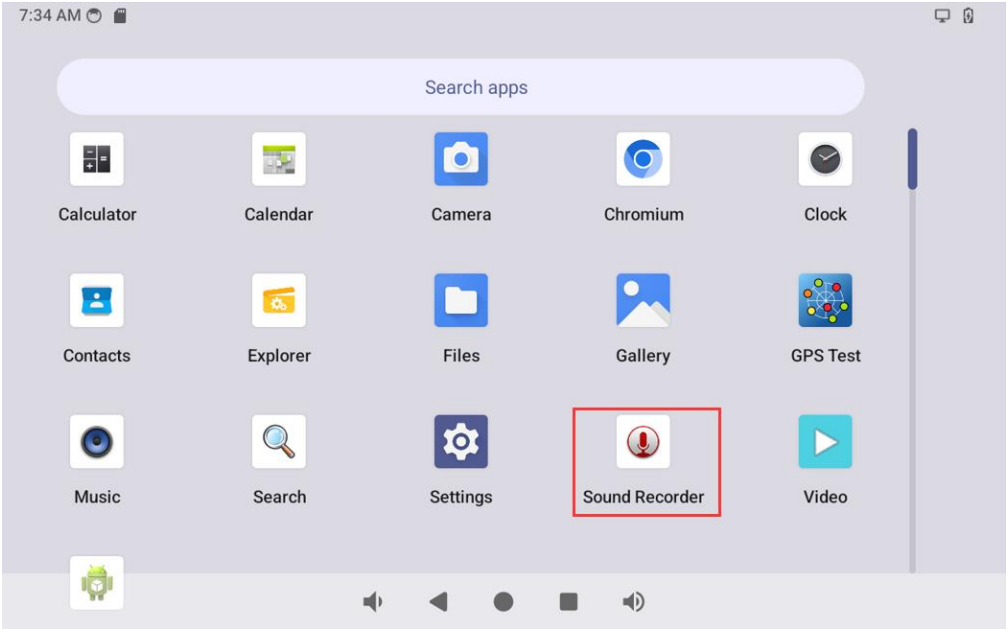


6.3 Audio

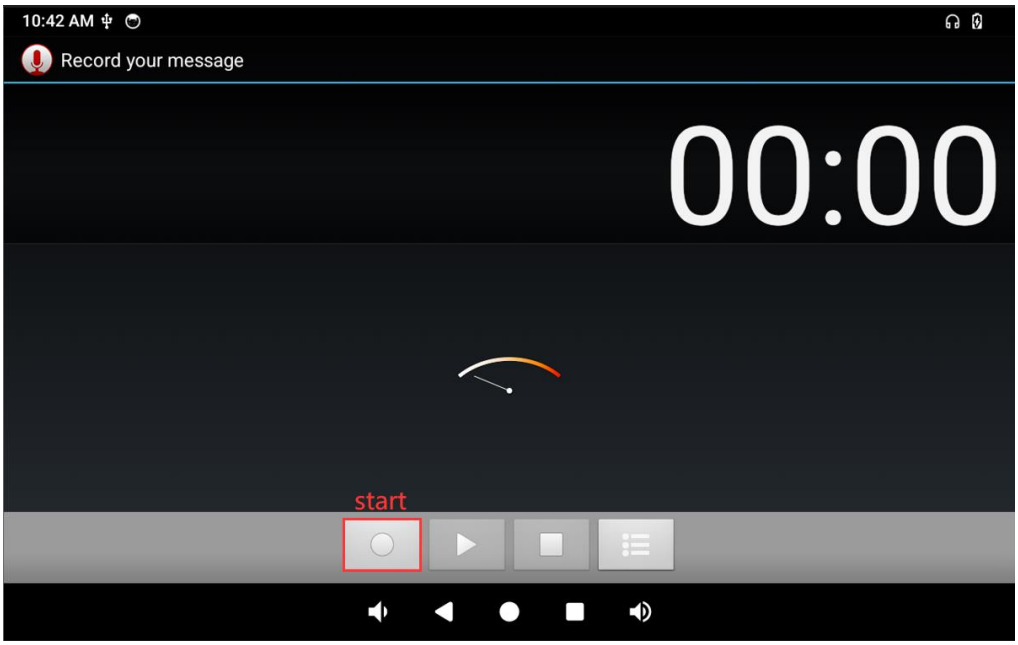
Step 1: Plug the headset into the headset jack.



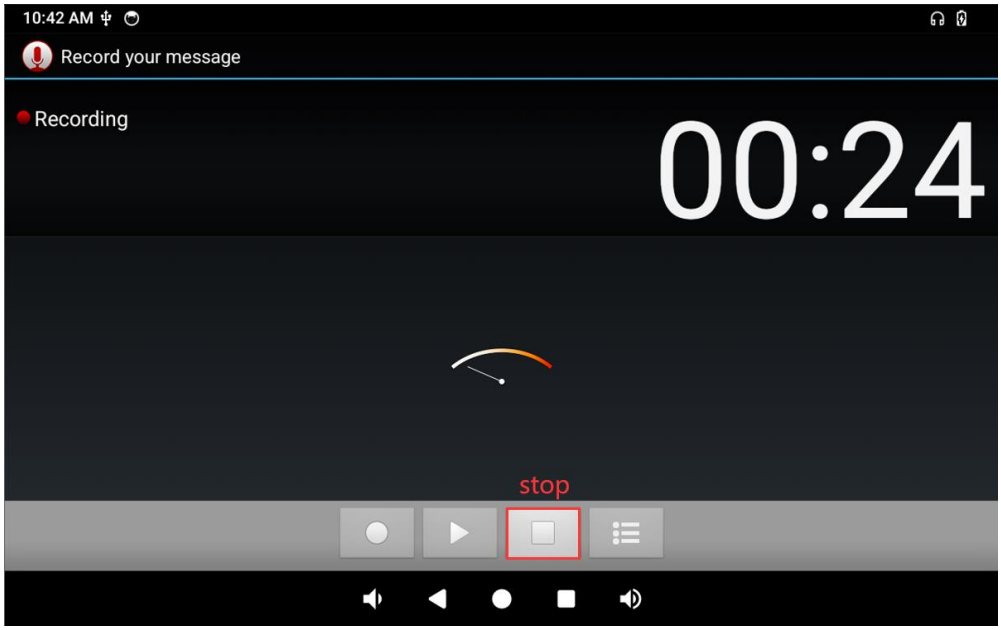
Step 2: Open the **Sound Recorder** app.



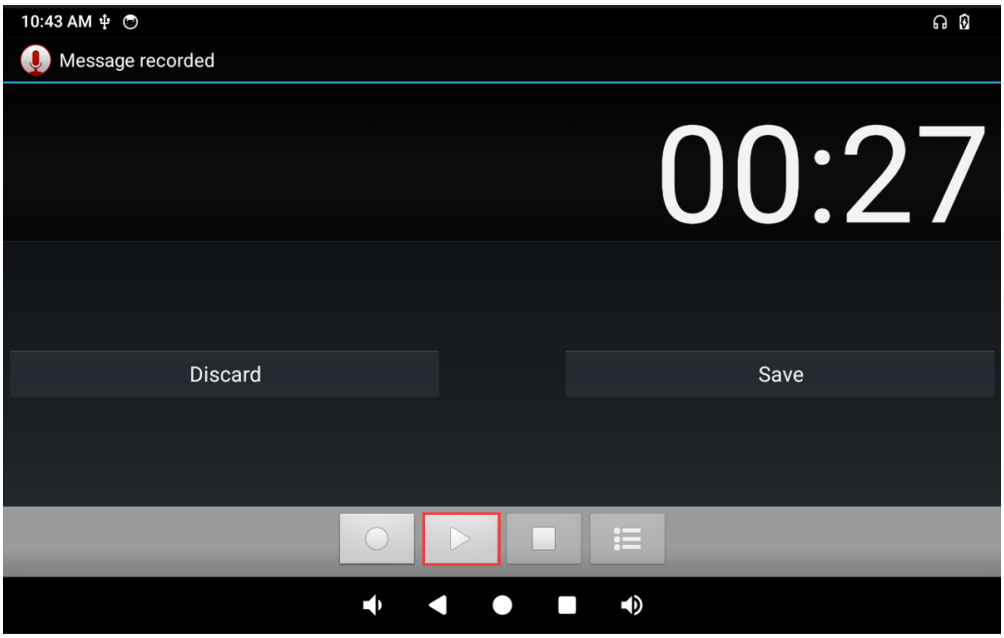
Step 3: Click the following button to start recording.



Step 4: Click the following button to stop recording.



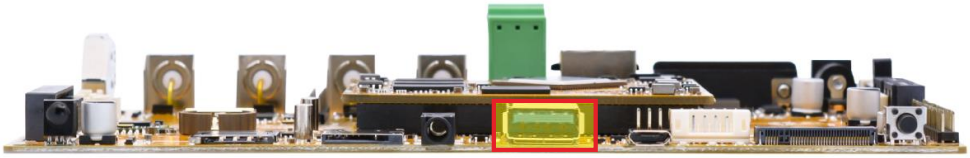
Step 5: Click the following button to play back the recording.



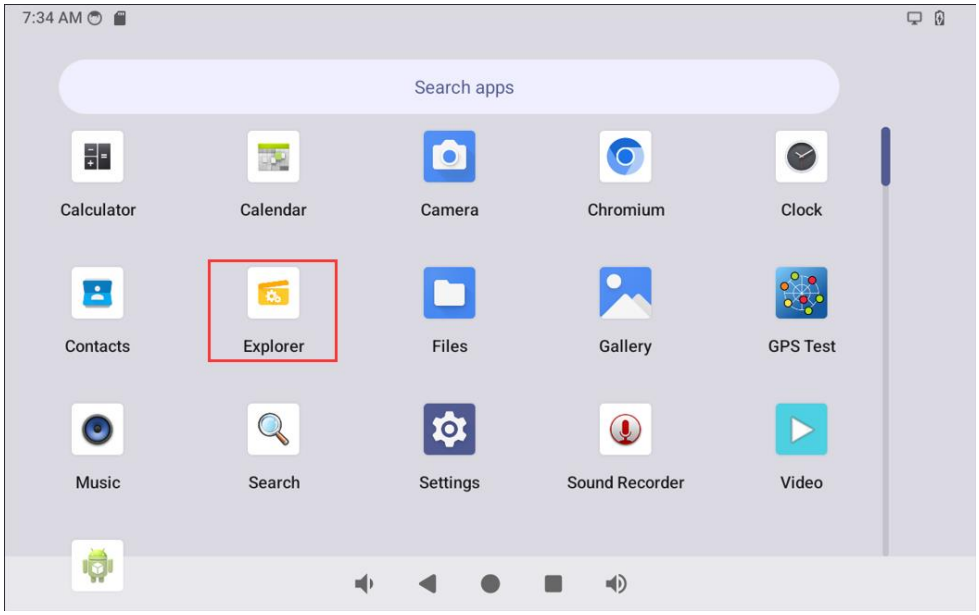
Note: Audio output level: HDMI > Headset.

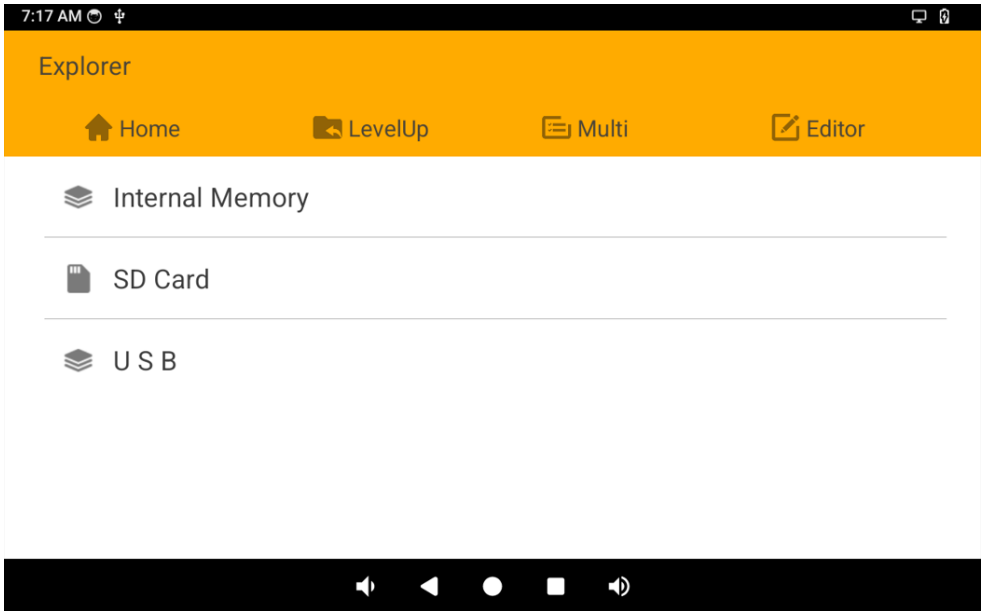
6.4 USB Host

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



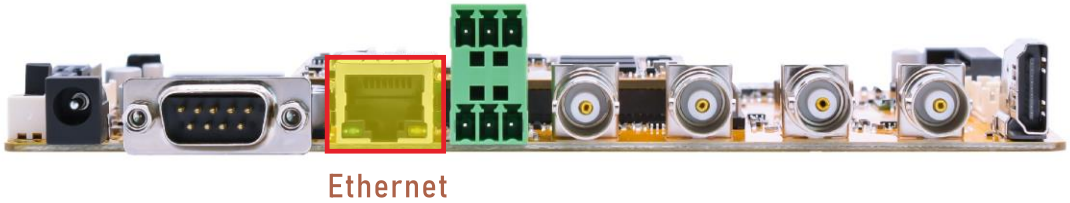
After connecting the USB flash drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.





6.5 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:/ # [  573.011713][ T2115] rk_gmac-dwmac fe2a0000.ethernet eth0: Link is Up - 1Gbps/Full - flow
control rx/tx
[  573.011871][ T2115] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

Step 2: View network interface information.

```
# ifconfig
```

```
console:/ # ifconfig
eth0      Link encap:Ethernet  HWaddr a6:eb:00:5b:9c:06  Driver rk_gmac-dwmac
          inet addr:192.168.0.82  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::b847:db1c:69c1:e2e4/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16332 TX bytes:4853
          Interrupt:54
```

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

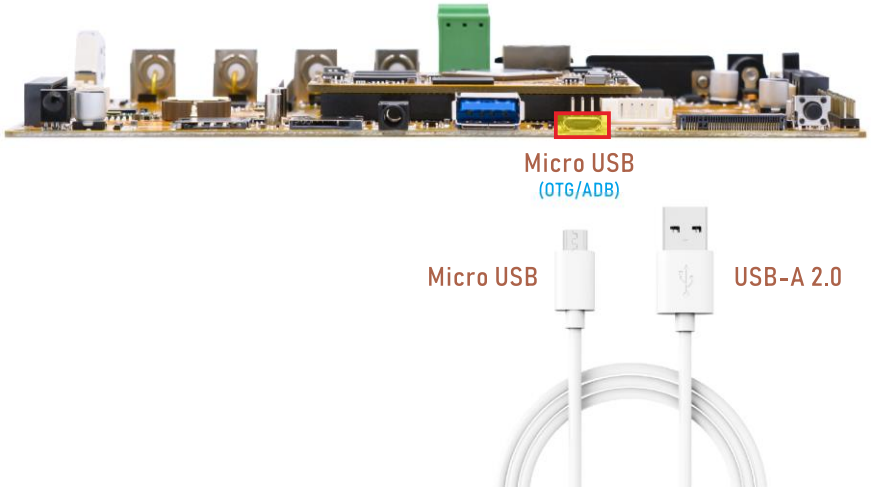
Step 3: Network connection test.

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

```
console:/ # ping -I eth0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.82 eth0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=50 time=182 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=50 time=182 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=50 time=182 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=50 time=183 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=50 time=182 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=6 ttl=50 time=182 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=7 ttl=50 time=182 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=8 ttl=50 time=183 ms
^C
--- www.armdesigner.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7010ms
rtt min/avg/max/mdev = 182.201/182.839/183.856/0.764 ms
```

6.6 ADB

Step 1: Connect the board and PC host with Micro usb cable.

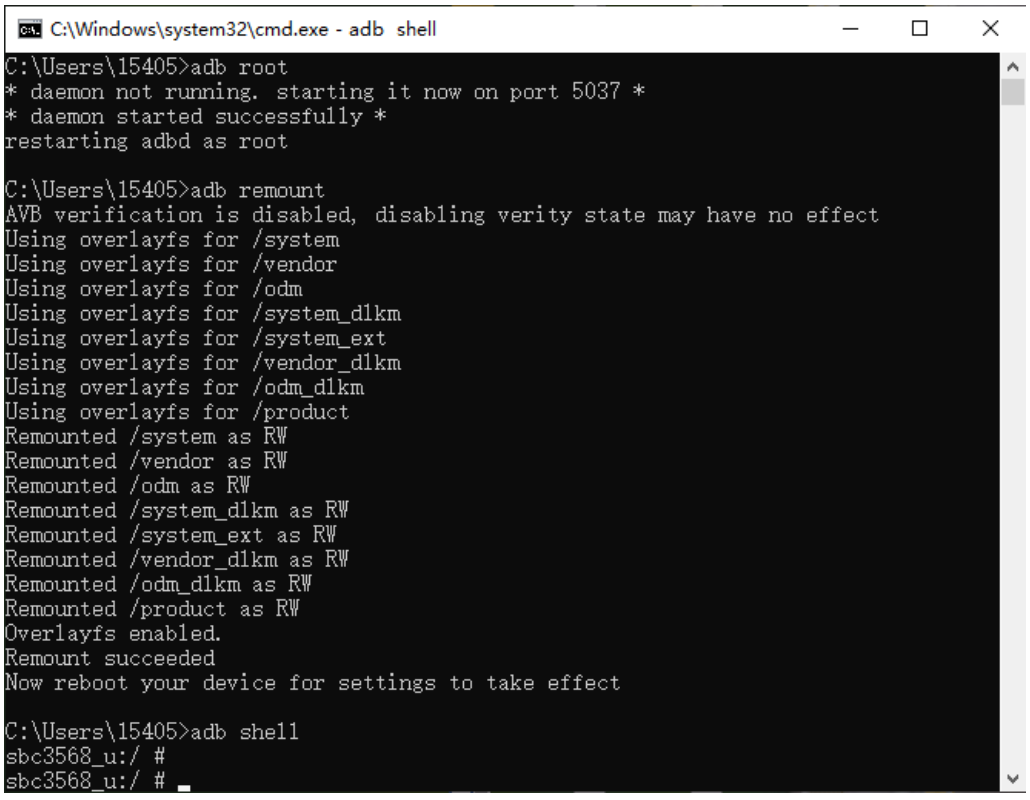


Step 2: Install ADB driver on Windows system.

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 root
# adb remount
# adb shell
```

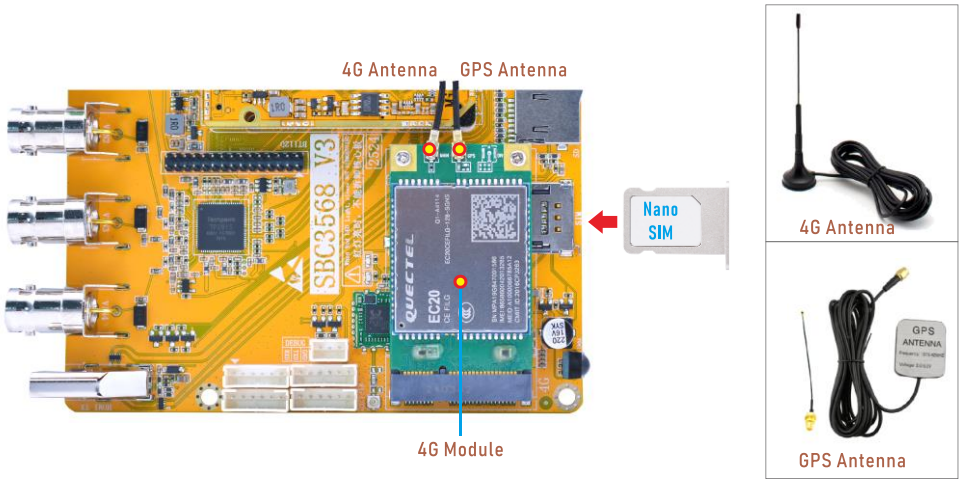


6.7 4G & GPS

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

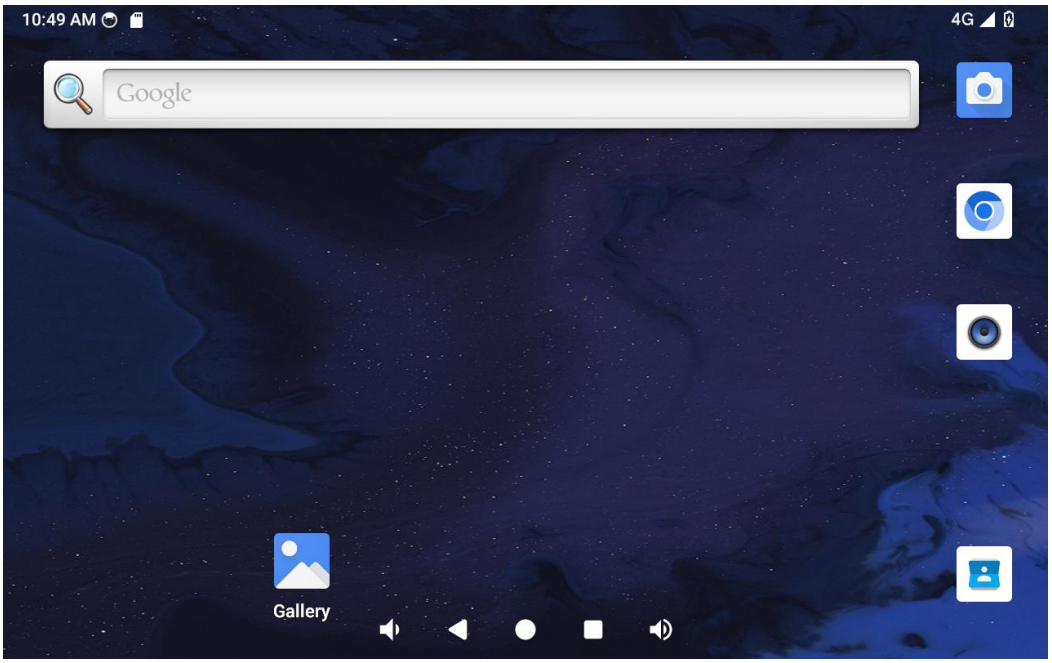
Step 2: Connect antenna and insert SIM card.

Step 3: Power on.

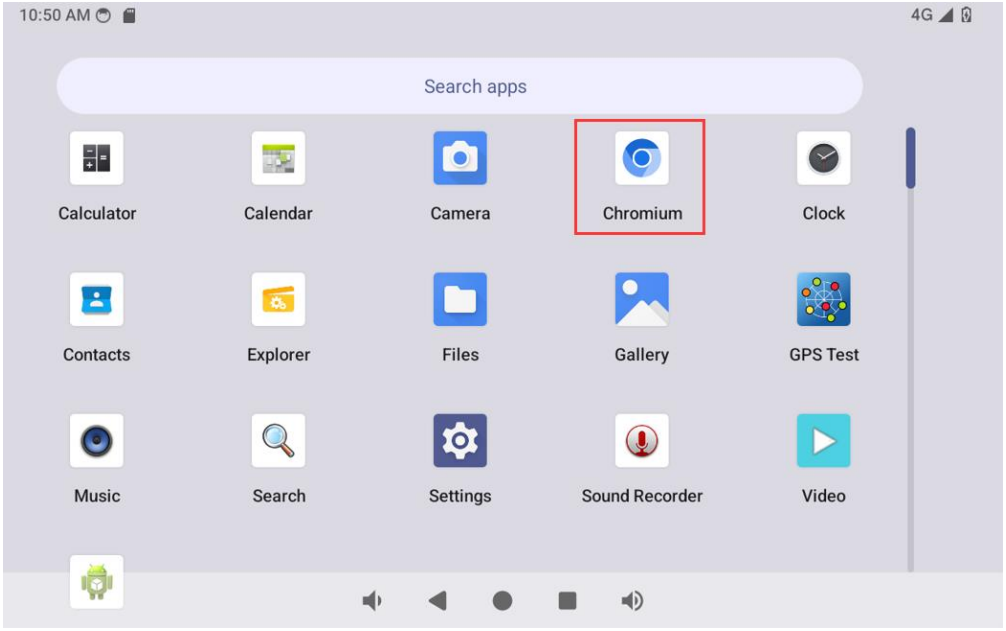


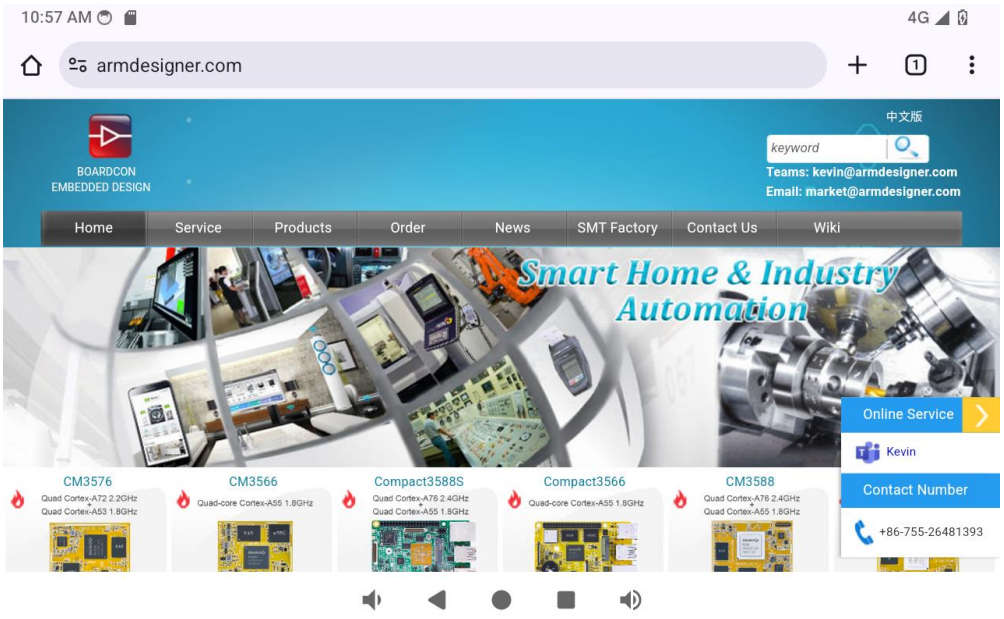
6.7.1 4G Test

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
```

```
console:/ # ifconfig
wwan0     Link encap:Ethernet  HWaddr 22:2d:30:80:15:4b  Driver qmi_wwan_q
          inet addr:10.8.43.14  Mask:255.255.240.0
          inet6 addr: fe80::202d:30ff:fe80:154b/64 Scope: Link
          UP RUNNING NOARP  MTU:1500  Metric:1
          RX packets:5729 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2937 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7282893 TX bytes:324536
```

(2) Network connection test.

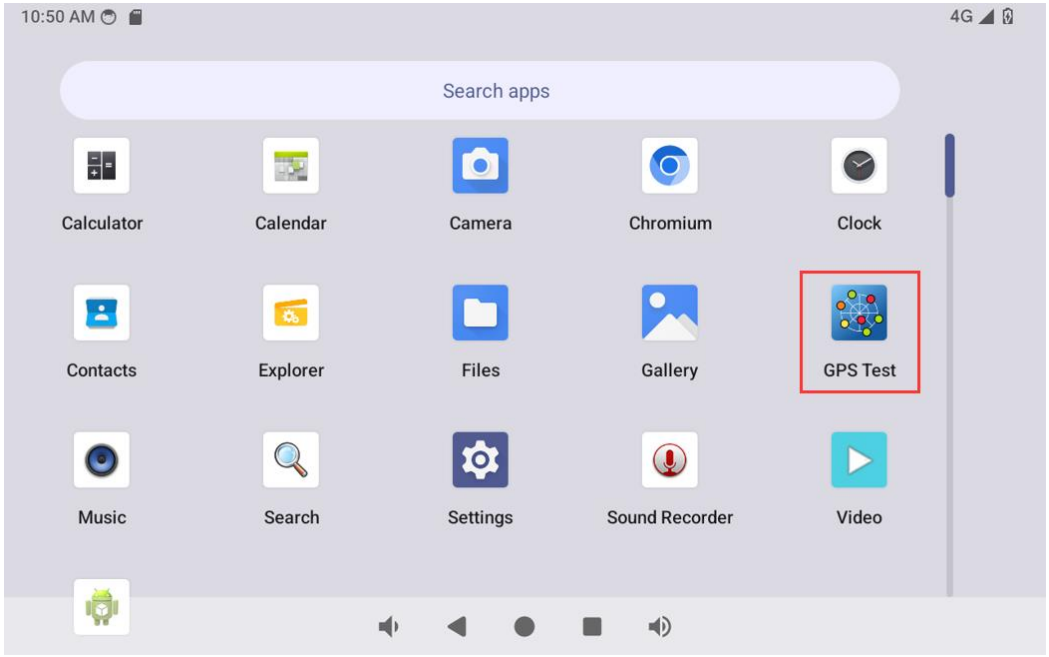
```
# ping -I wwan0 www.armdesigner.com
```

```
console:/ # ping -I wwan0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 10.8.43.14 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=223 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=2 ttl=47 time=211 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=3 ttl=47 time=204 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=4 ttl=47 time=203 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=5 ttl=47 time=202 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=6 ttl=47 time=204 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=7 ttl=47 time=204 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 10629ms
rtt min/avg/max/mdev = 202.321/207.817/223.637/6.971 ms
```

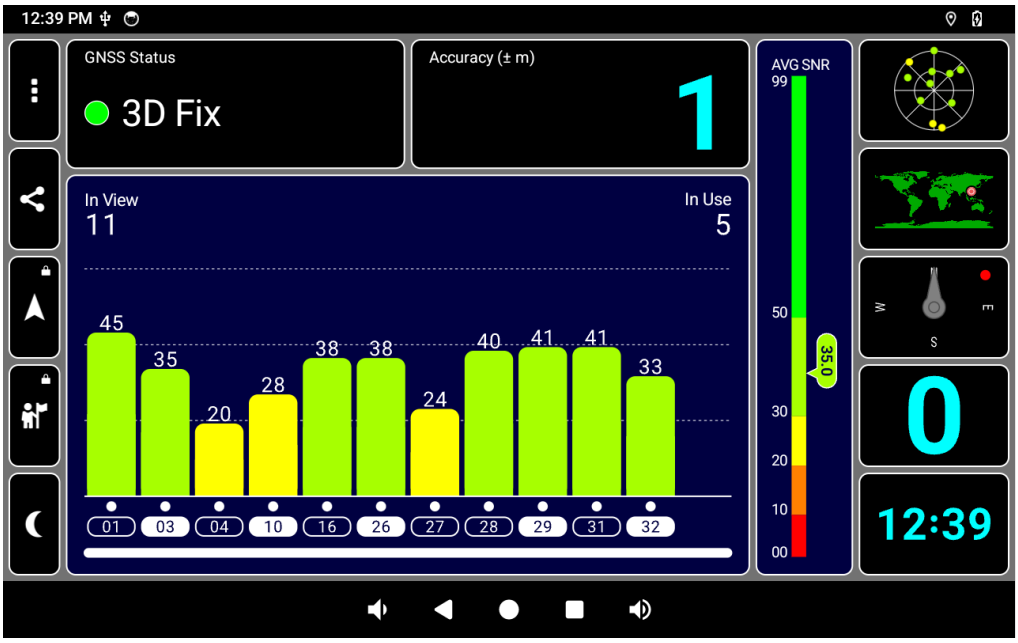
6.7.2 GPS Test

GPS positioning needs to be tested in open outdoor areas.

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

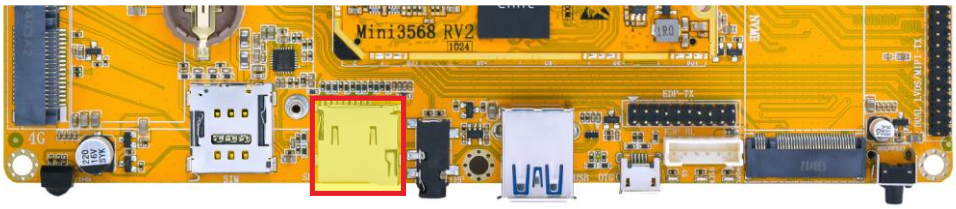


Step 2: Wait a moment to successfully obtain the location.

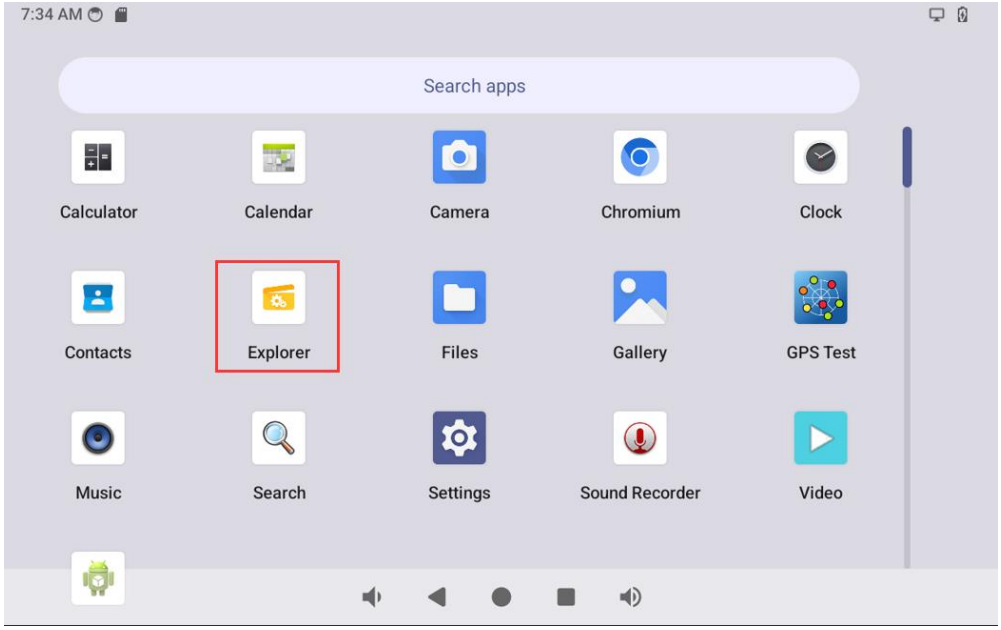


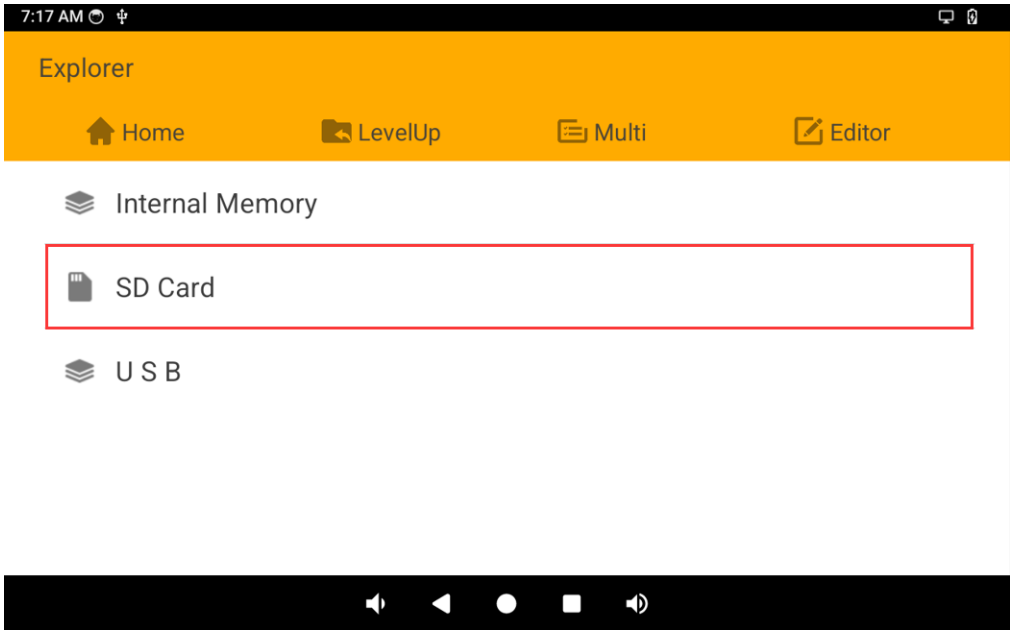
6.8 SD Card

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



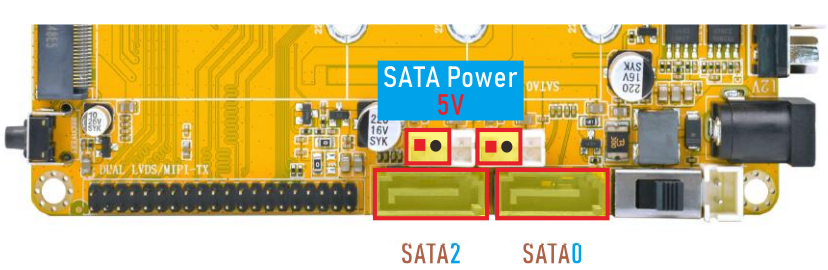
Step 2: After connecting the SD card drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.



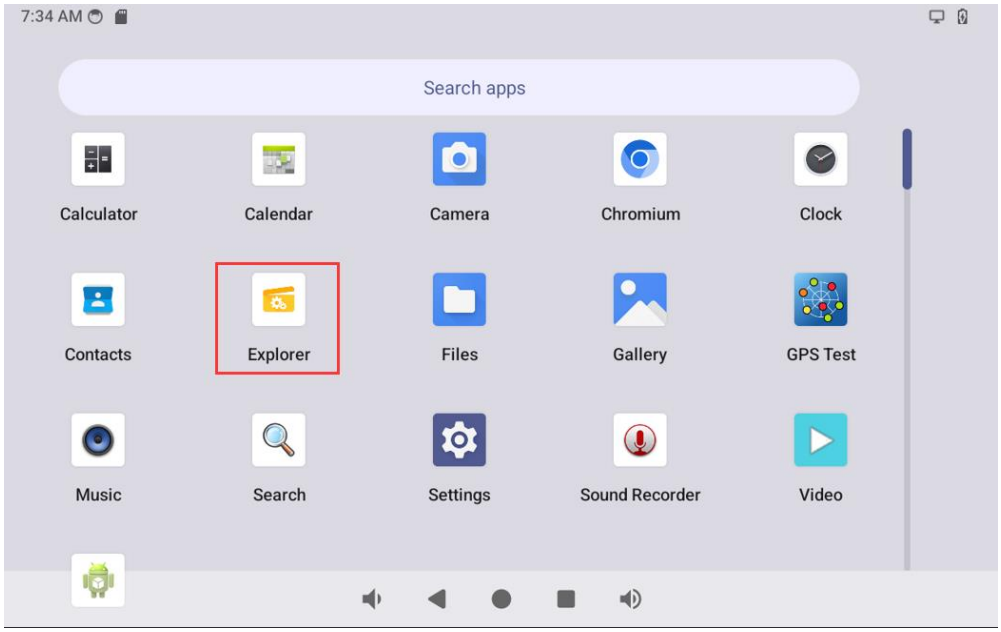


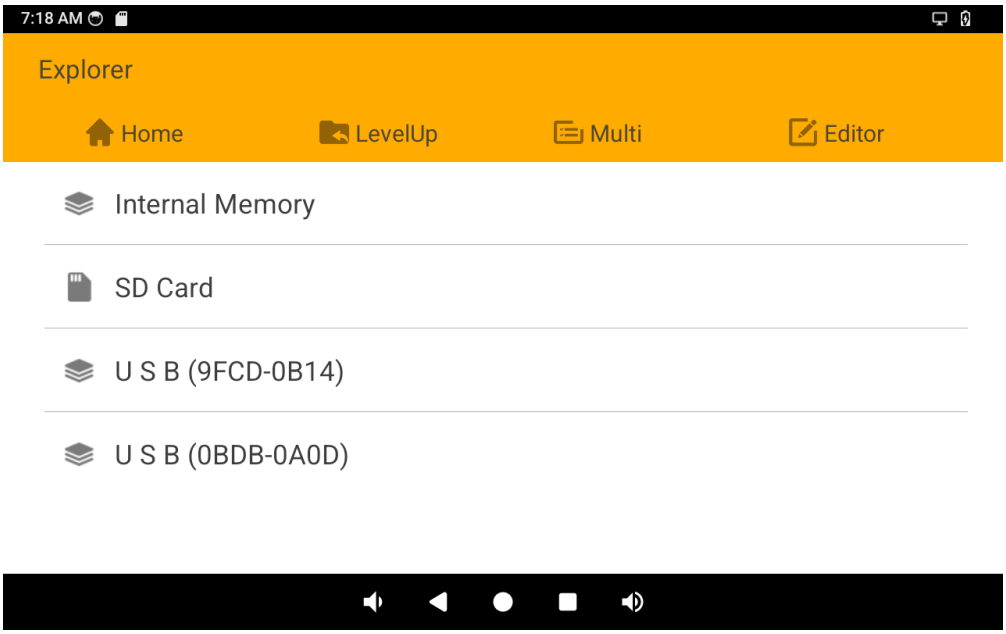
6.9 SATA

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

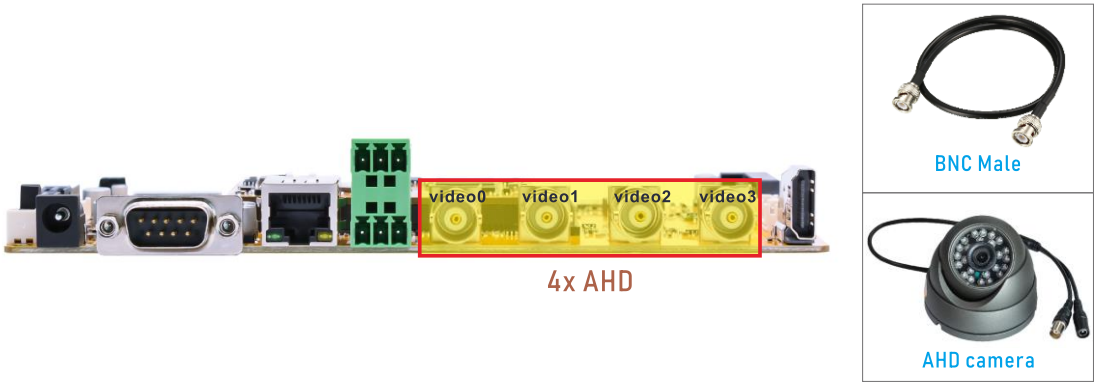


Step 2: After connecting the sata drive, the device will automatically mount, and it can be accessed directly through the **Explorer** app.





6.10 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 video0 as an example.

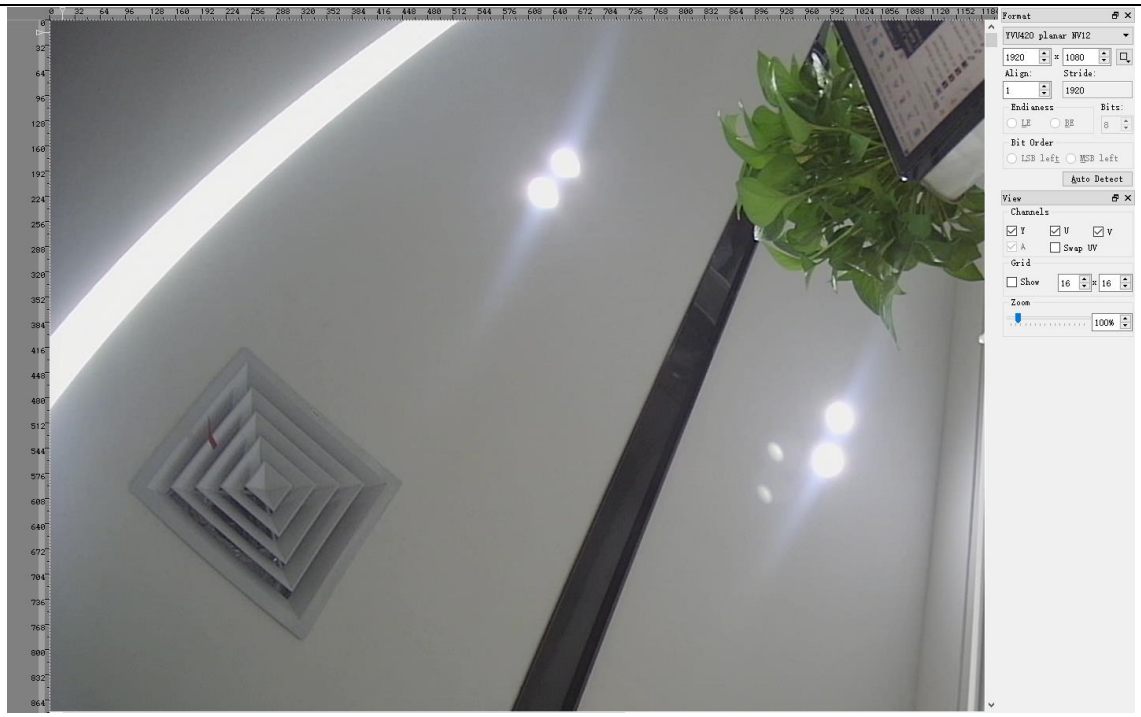
```
# v4l2-ctl --verbose -d /dev/video0 --set-fmt-
video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4 --set-
selection=target=crop,flags=0,top=0,left=0,width=1920,height=1080 --stream-
to=/mnt/out1_0.yuv
```

```
console:/ #
ags=0,top=0,left=0,width=1920,height=1080 --stream-to=/mnt/out1_0.yuv        <
VIDIOC_QUERYCAP: ok
VIDIOC_G_FMT: ok
VIDIOC_S_FMT: ok
[ 7482.685603][ T8538] rkcif_mipi_lvds: static crop, S_SELECTION(1920x1080@0:0) target: 0
Format Video Capture Multiplanar:
       Width/Height      : 1920/1080
       Pixel Format      : 'NV12'
       Field             : None
       Number of planes  : 1
       Flags             :
       Colorspace        : 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
[ 7482.698284][ T8538] rkcif_mipi_lvds: stream[0] start streaming
[ 7482.701722][ T8538] rkcif_mipi_lvds: Allocate dummy buffer, size: 0x003f5000
[ 7482.705655][ T8538] rockchip-mipi-csi2 mipi-csi2: stream on, src_sd: 000000001d19f2d9,
sd_name:rockchip-csi2-dphy0
[ 7482.705695][ T8538] rockchip-mipi-csi2 mipi-csi2: stream ON
[ 7482.705747][ T8538] rockchip-csi2-dphy0: dphy0, data_rate_mbps 594
[ 7482.705795][ T8538] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:1, dphy0, ret 0
[ 7482.707098][ T8538] techpoint 2-0047: detect channel 0 1080P_25
[ 7482.707884][ T8538] techpoint 2-0047: set channel 0 1080P_25
[ 7482.748405][ T8538] techpoint 2-0047: detect channel 1 is not supported, default 1080P_25
[ 7482.749469][ T8538] techpoint 2-0047: set channel 1 1080P_25
[ 7482.796549][ T8538] techpoint 2-0047: detect channel 2 is not supported, default 1080P_25
[ 7482.797627][ T8538] techpoint 2-0047: set channel 2 1080P_25
[ 7482.841944][ T8538] techpoint 2-0047: detect channel 3 is not supported, default 1080P_25
[ 7482.843026][ T8538] techpoint 2-0047: set channel 3 1080P_25
[ 7482.900085][    C0] (0xfdfb0000)MIPI_CSI2 ERR1:0x700 (f_seq,vc: 0 1 2)
[ 7482.903092][    C0] (0xfdfb0000)MIPI_CSI2 ERR1:0x770 (fs/fe mis,vc: 0 1 2) (f_seq,vc: 0 1 2)
[ 7482.903850][    C0] (0xfdfb0000)MIPI_CSI2 ERR1:0x800 (f_seq,vc: 3)
VIDIOC_STREAMON: ok
idx: 0 seq:      0 bytesused: 3110400 ts: 7482.361913
idx: 1 seq:      1 bytesused: 3110400 ts: 7482.401921 delta: 40.008 ms
idx: 2 seq:      2 bytesused: 3110400 ts: 7482.441963 delta: 40.042 ms
idx: 3 seq:      3 bytesused: 3110400 ts: 7482.481962 delta: 39.999 ms
idx: 0 seq:      4 bytesused: 3110400 ts: 7482.521968 delta: 40.006 ms fps: 24.99
idx: 1 seq:      5 bytesused: 3110400 ts: 7482.561968 delta: 40.000 ms fps: 24.99
idx: 2 seq:      6 bytesused: 3110400 ts: 7482.601958 delta: 39.990 ms fps: 25.00
idx: 3 seq:      7 bytesused: 3110400 ts: 7482.641970 delta: 40.012 ms fps: 24.99
idx: 0 seq:      8 bytesused: 3110400 ts: 7482.681975 delta: 40.005 ms fps: 25.00
idx: 1 seq:      9 bytesused: 3110400 ts: 7482.721925 delta: 39.950 ms fps: 25.00
idx: 2 seq:     10 bytesused: 3110400 ts: 7482.761914 delta: 39.989 ms fps: 25.00
idx: 3 seq:     11 bytesused: 3110400 ts: 7482.801966 delta: 40.052 ms fps: 25.00
idx: 0 seq:     12 bytesused: 3110400 ts: 7482.841973 delta: 40.007 ms fps: 25.00
idx: 1 seq:     13 bytesused: 3110400 ts: 7482.881979 delta: 40.006 ms fps: 25.00
```

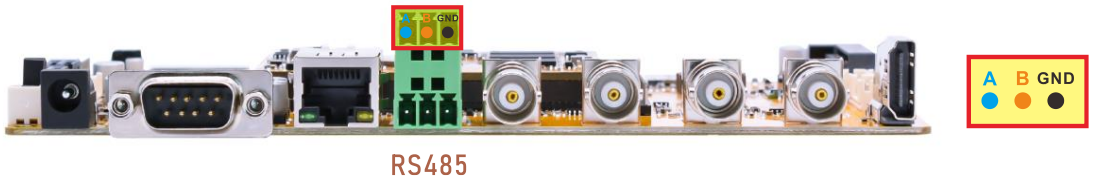
The file will save as under the current directory.

```
console:/ # ls /mnt/
androidwritable  expand     media_rw    out1_0.yuv    runtime  secure
appfuse          installer  obb         pass_through  scratch  user
asec             iso        out0_0.yuv  product       sdcard   vendor
```

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

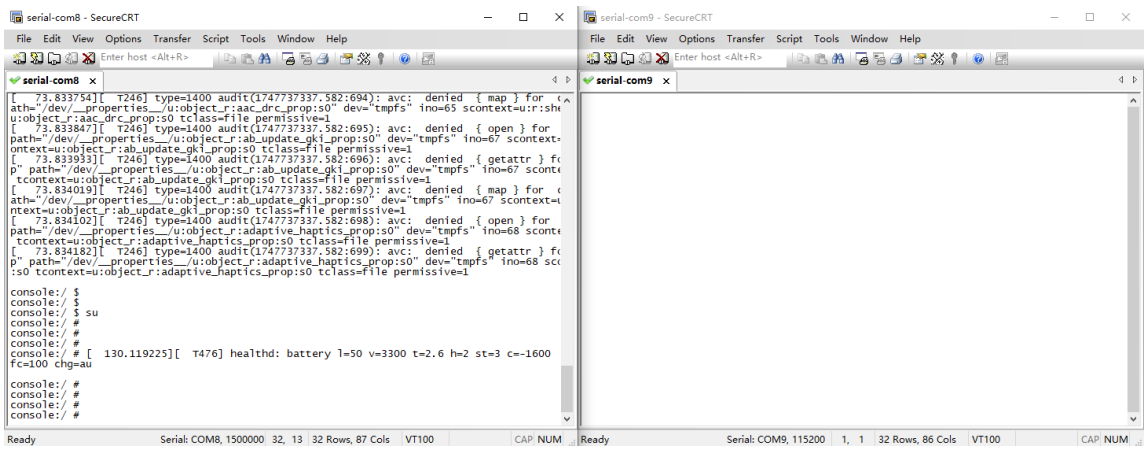


6.11 RS485



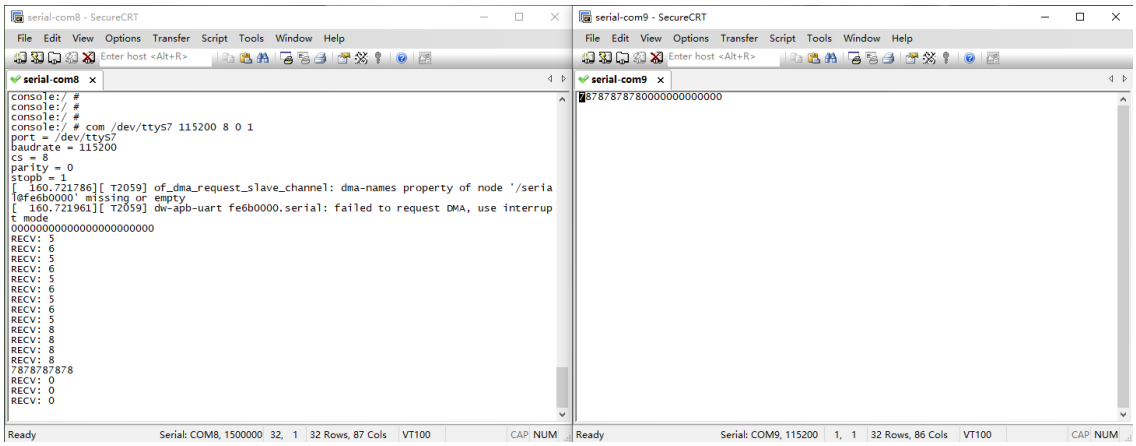
Step 1: As shown in the diagram, connect the RS485 test tool to the development board.

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.

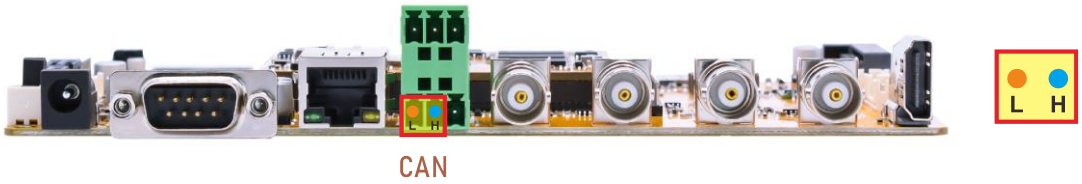


Step 3: Execute the following command on the board to test the RS485 transmission and reception functionality.

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

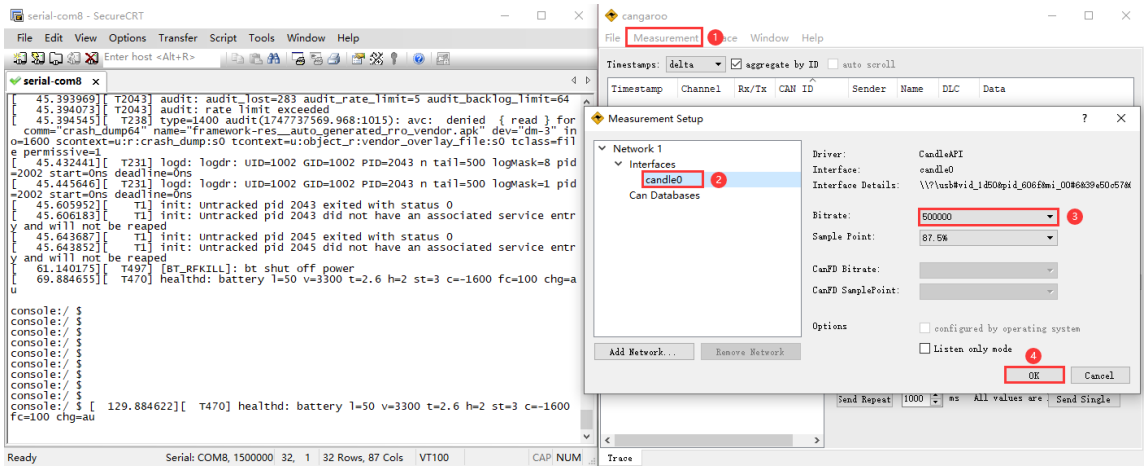


6.12 CAN



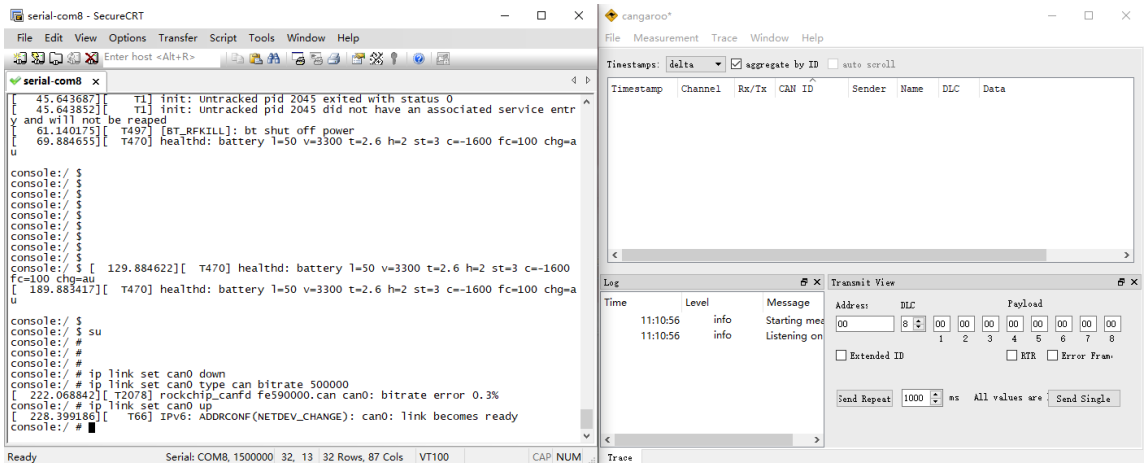
Step 1: Connect the CAN test tool to the board as shown in the diagram below.

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



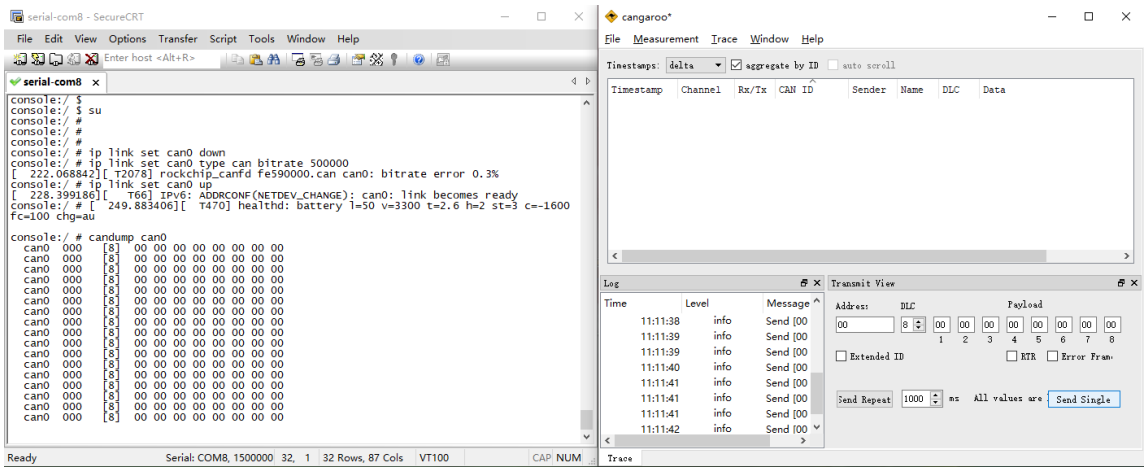
Step 3: Configure and activate the CAN network, setting the bitrate to 500000.

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



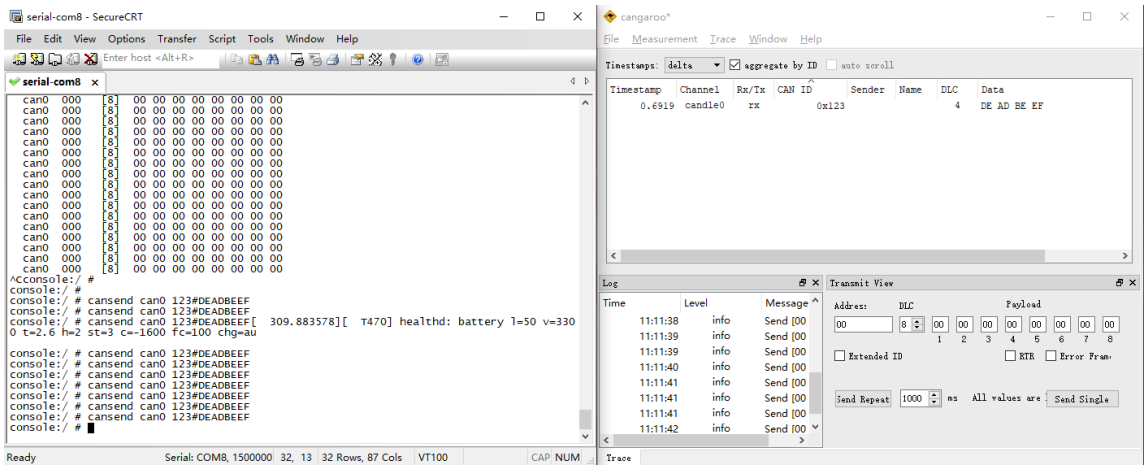
Step 4: Configure CAN as the receiver.

```
# candump can0
```



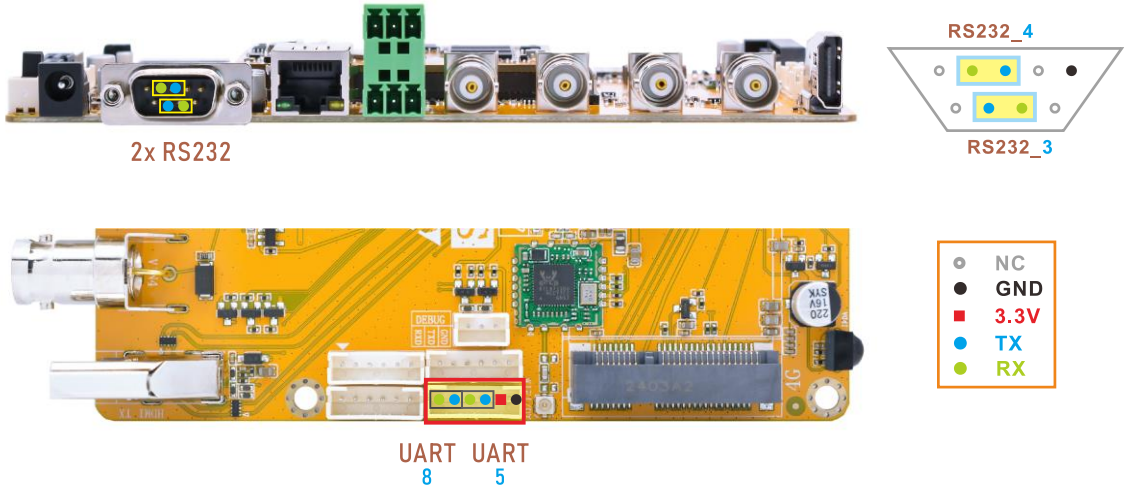
Step 5: Configure CAN as the sender.

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



6.13 UART

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



Step 2: UART Test Command.

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

Parameter Description:

```
/dev/ttySx: UART device node. (e.g., /dev/ttyS3 for UART3)
115200: Baud rate.
8: Data bits.
0: Parity (0 = None).
1: Stop bit.
```

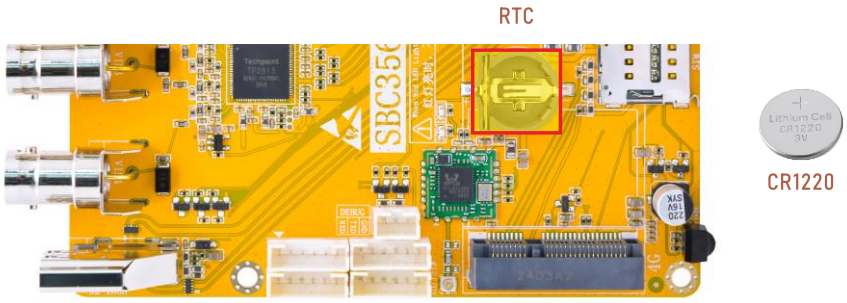
Step 3: UART test. (Using UART3 as a Reference)

```
# 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
uuuuuuuuuuuuuuuuu
RECV: uuuuuuuuuuuuuuuuu
hjhjhjhjhjh
RECV: hjhjhjhjhjh
0000000kkkkkk
RECV: 0000000kkkkkk
mmmm
RECV: mmmm
```

6.14 RTC

Step 1: Install the coin cell battery.



Step 2: Set the system time.

```
# date -s "2025-05-21 11:36:00"
```

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

```
# hwclock -w
```

Step 4: Display the current hardware clock time.

```
# hwclock
```

```
console:/ # date -s "2025-05-21 11:36:00"
Wed May 21 11:36:00 GMT 2025
console:/ # hwclock -w
console:/ # hwclock
2025-05-21 11:36:05+0000
console:/ # hwclock
2025-05-21 11:36:51+0000
console:/ # hwclock
2025-05-21 11:37:01+0000
```

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

```
console:/ # hwclock
2025-05-21 11:59:53+0000
console:/ # hwclock
2025-05-21 12:00:01+0000
console:/ # hwclock
2025-05-21 12:00:21+0000
console:/ # hwclock
2025-05-21 12:01:05+0000
```

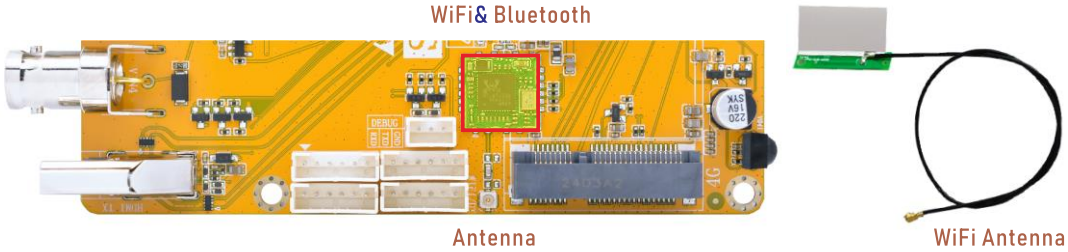
6.15 WiFi & Bluetooth

Note: The WiFi/Bluetooth module shares the USB differential data lines (**USB_HOST_OTGDM / USB_HOST_OTGDP**) with the USB OTG interface.

Do not connect a Micro USB cable when using WiFi or Bluetooth, as this may cause

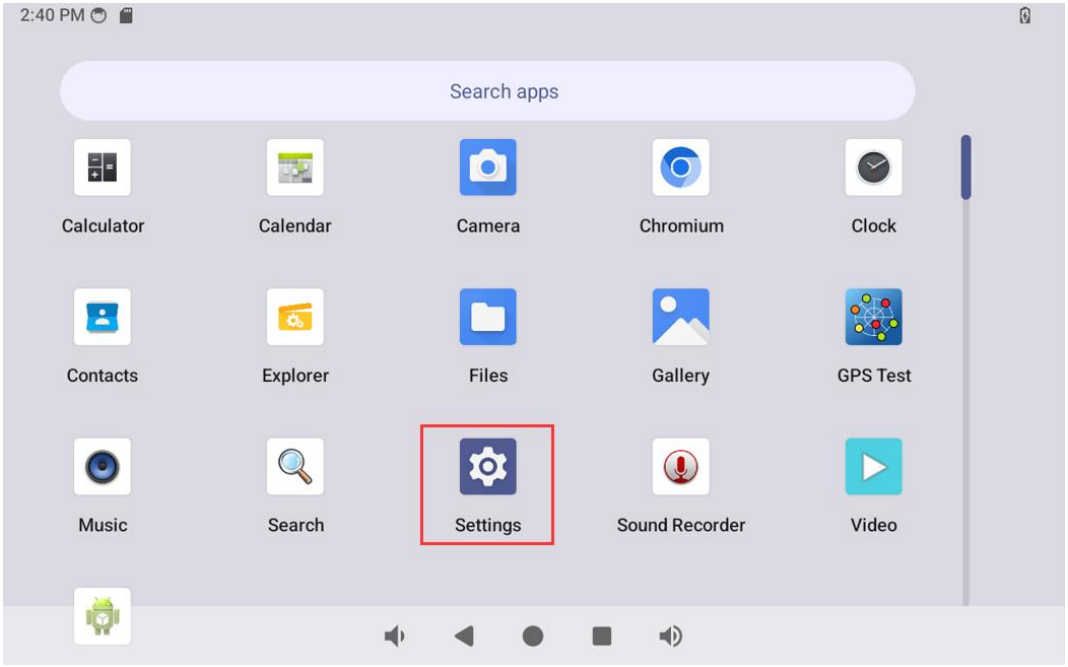
signal conflicts and lead to wireless communication issues.

Additionally, to ensure proper functionality of WiFi and Bluetooth, **an external antenna must be connected.**

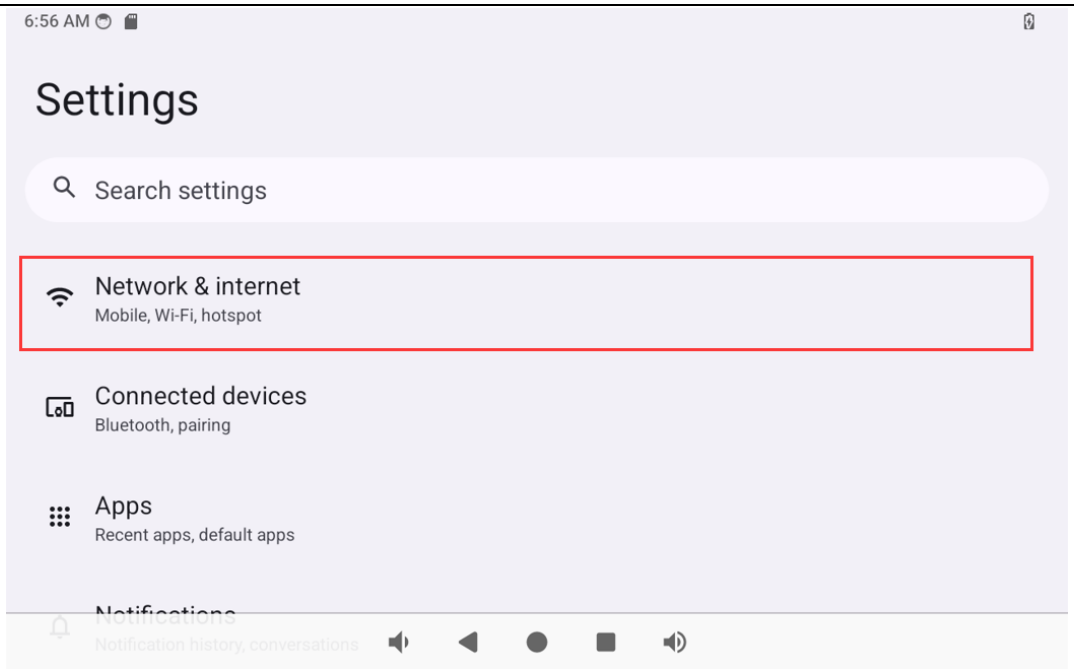


6.15.1 WiFi test

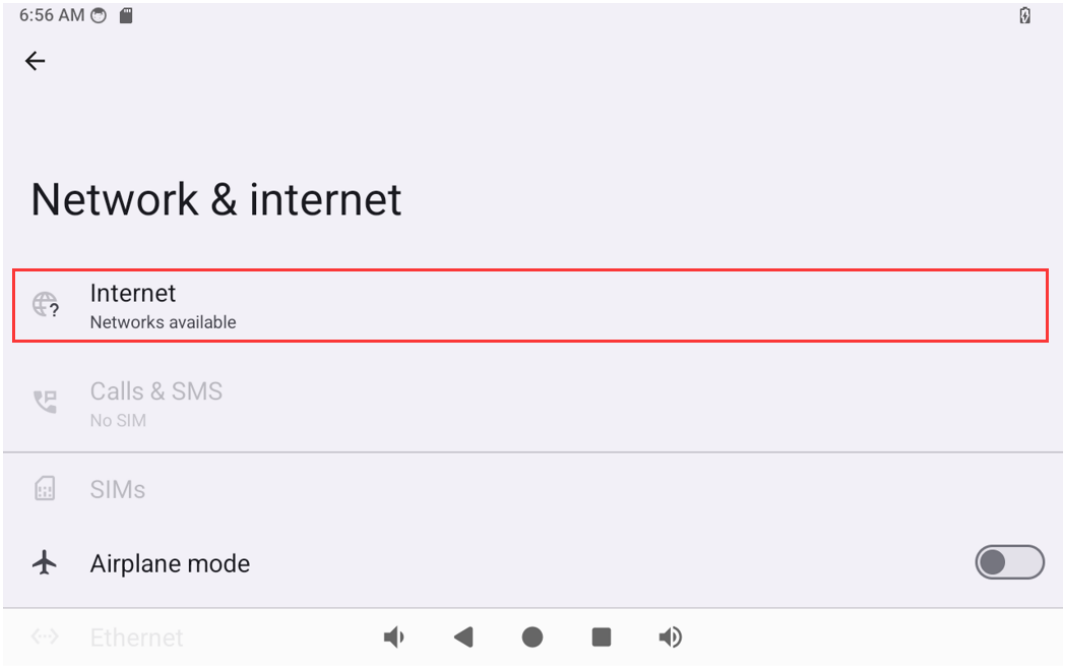
Step 1: Open the **Settings** app：



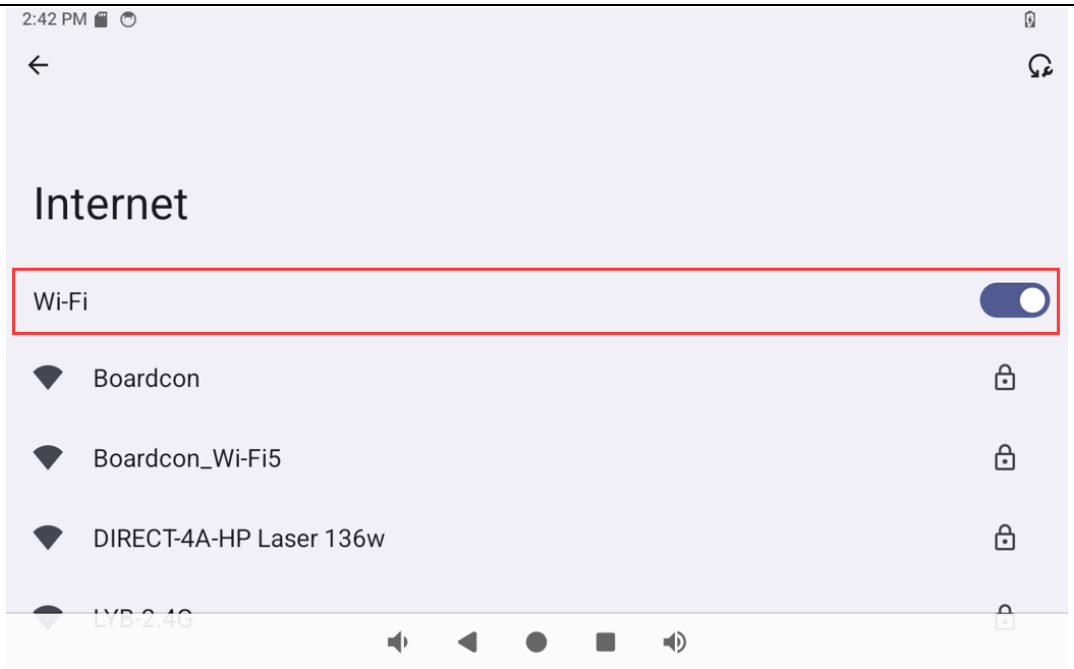
Step 2: Go to **Network & Internet**:



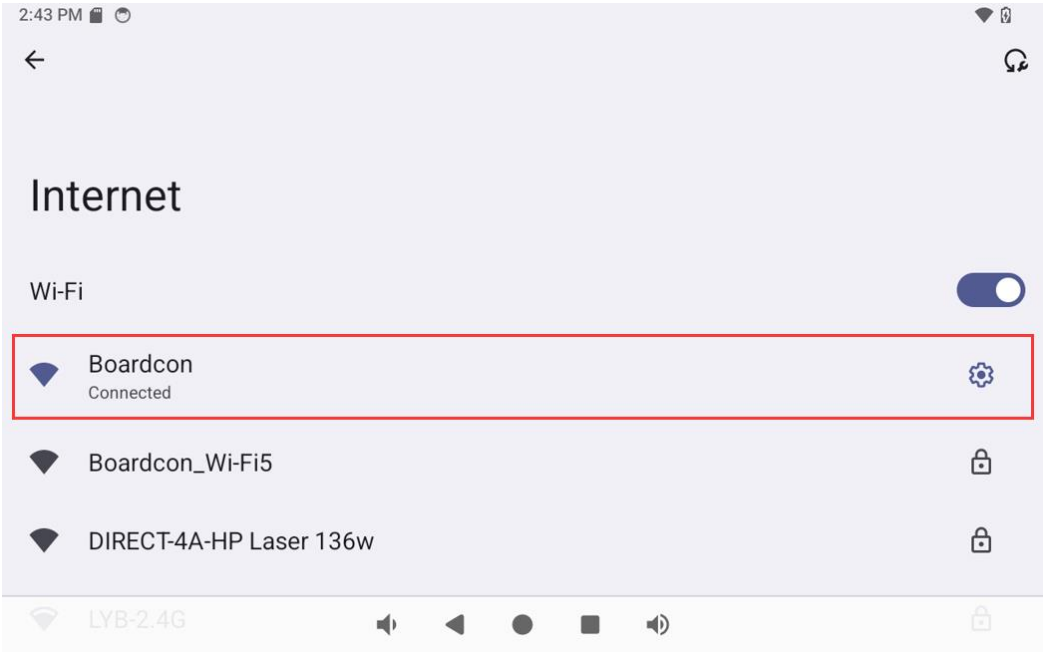
Step 3: Tap **Internet**:



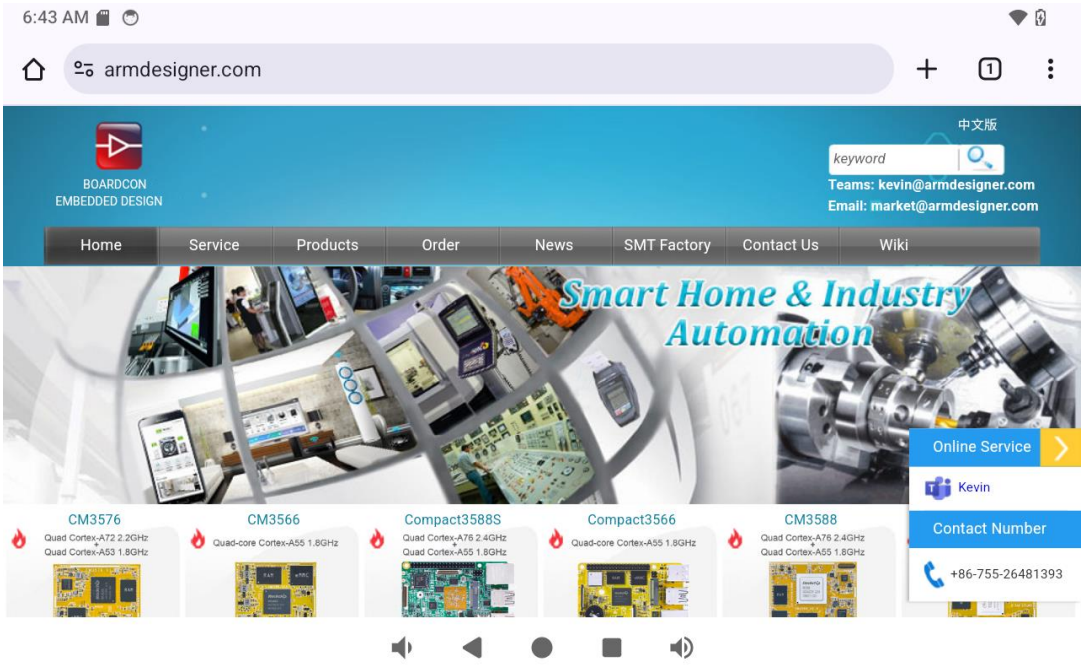
Step 4: Turn on the **Wi-Fi switch**:



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 c8:fe:0f:02:49:98  Driver rtl8723du
          inet addr:192.168.0.166  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::bb17:2351:34f7:f96a/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:106 errors:0 dropped:15 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:932358 TX bytes:221991
```

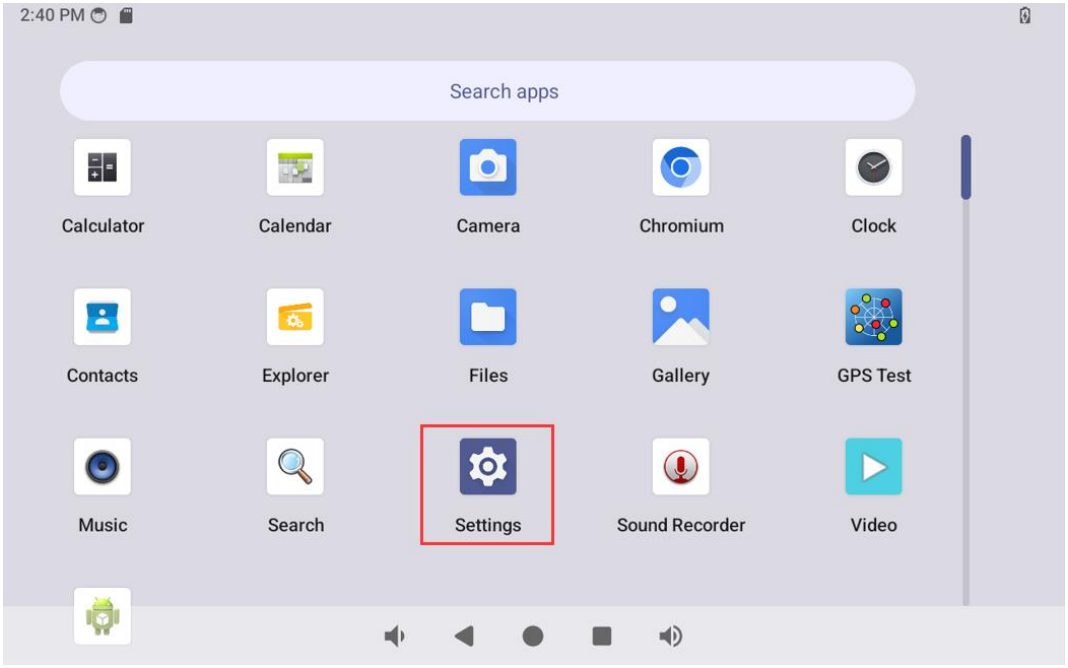
(2) Network connection test.

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

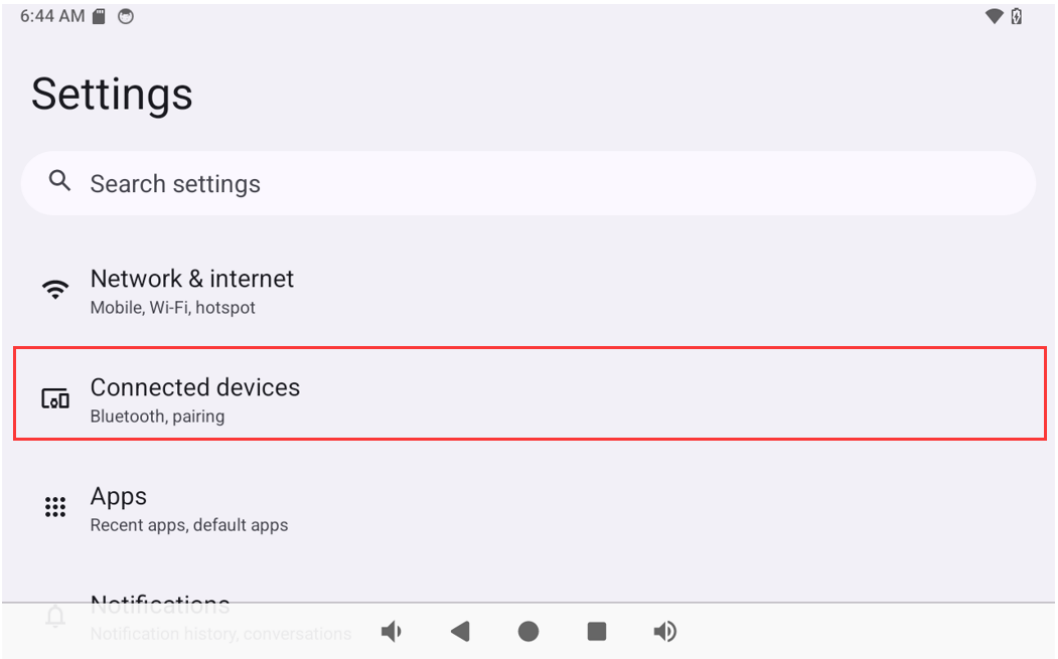
```
console:/ # ping -I wlan0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.166 wlan0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=50 time=551 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=50 time=190 ms
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=4 ttl=50 time=1319 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=50 time=316 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=6 ttl=50 time=189 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=7 ttl=50 time=213 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6015ms
rtt min/avg/max/mdev = 189.893/424.588/1319.079/384.818 ms, pipe 2
```

6.15.2 Bluetooth test

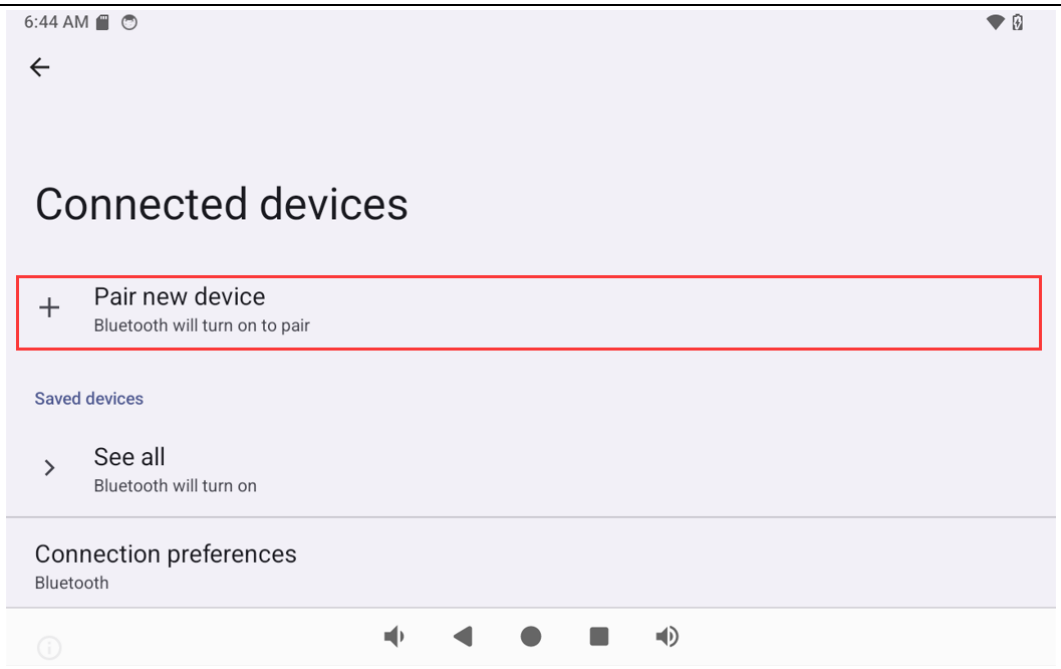
Step 1: Open the **Settings** app.



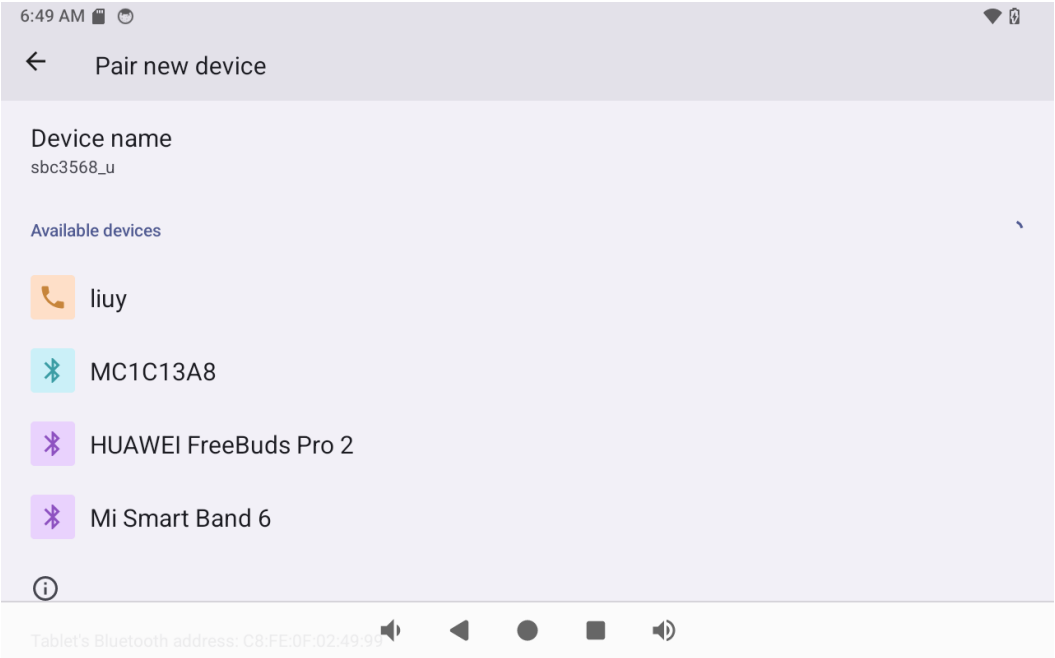
Step 2: Click on the option **Connected devices**



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

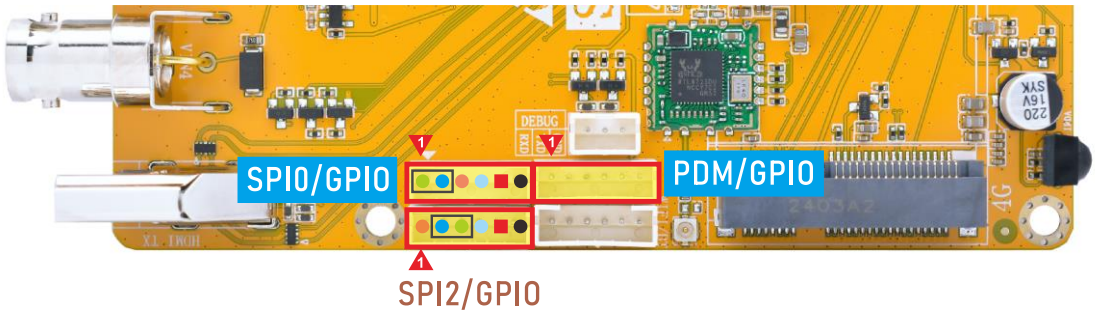


Step 4: User can pair themselves in the Bluetooth device list.



6.16 SPI

Step 1: short circuit MISO_M1 and MOSI_M1 pins of SPI.



Step 2: Execute the test script: **spidev0.0-test** or **spidev2.0-test**.

• **spidev0.0-test** corresponds to the SPI0 device.

• **spidev2.0-test** corresponds to the SPI2 device.

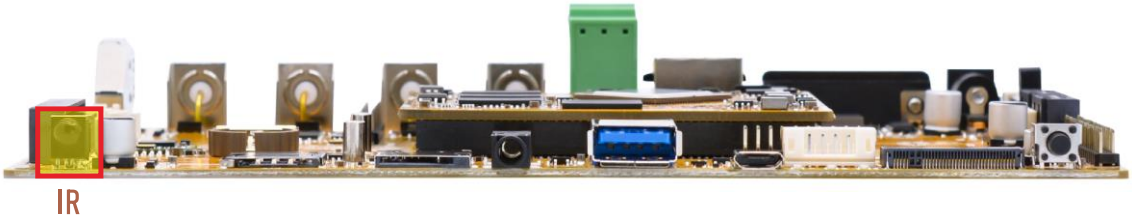
• The following test uses SPI0 as an example.

```
# spidev0.0-test
```

```
console:/ # spidev0.0-test
spi mode: 0
bits per word: 8
max speed: 10000000 Hz (10000 KHz)

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

6.17 IR



Step 1: Open IR debugging print.

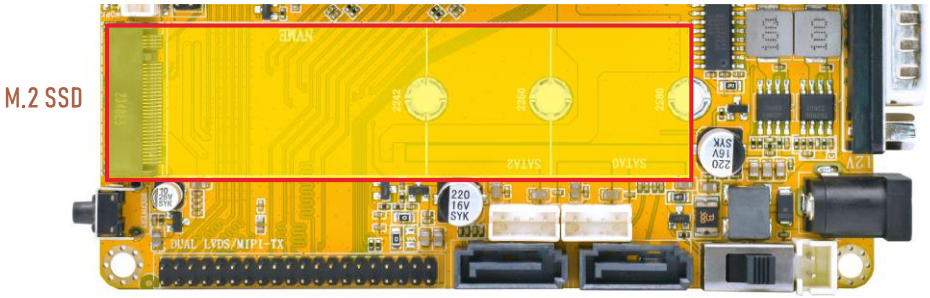
```
# echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

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

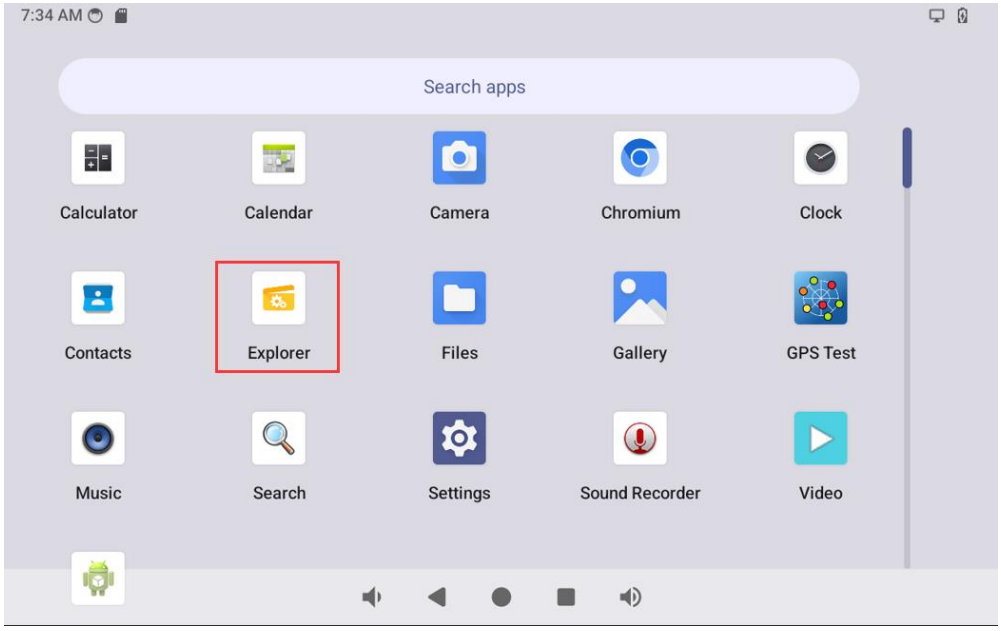
```
console:/ # echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
console:/ #
console:/ # [   46.094606][    C1] USERCODE=0x1818
[   46.121895][    C1] RMC_GETDATA=9a
[   46.466599][    C1] USERCODE=0x1818
[   46.493658][    C1] RMC_GETDATA=98
[   46.862325][    C1] USERCODE=0x1818
[   46.889386][    C1] RMC_GETDATA=99
[   47.242728][    C1] USERCODE=0x1818
[   47.269909][    C1] RMC_GETDATA=9b
[   48.474946][    C1] USERCODE=0x1818
[   48.502082][    C1] RMC_GETDATA=e4
[   49.566664][    C1] USERCODE=0x1818
[   49.593725][    C1] RMC_GETDATA=e5
```

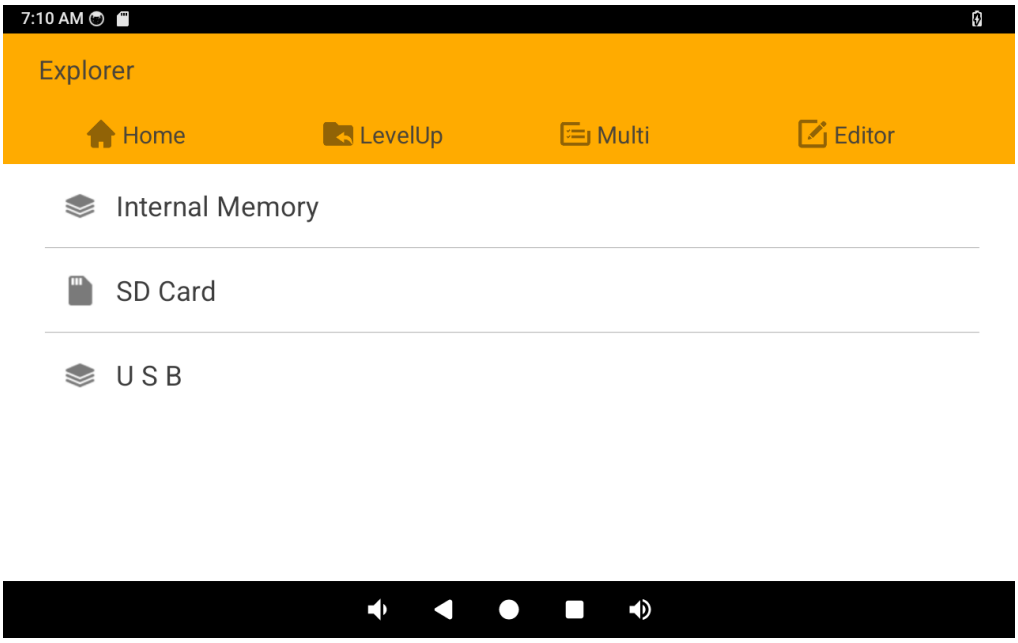
6.18 M.2 SSD

Step 1: Connect the SSD, 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.




7:10 AM
Explorer
Home
LevelUp
Multi
Editor
Internal Memory
SD Card
U S B