Compact3566 Android14 User Manual

V1.0

Boardcon Embedded Designer

Overview

The content of this document is intended solely for the Compact3566 development board, aiming to help users quickly understand, apply, and test the Compact3566 development board.

System Support

Development Board	Android14
Compact3566_V3	Y

Revision History

Version	Date	Author	Revision History
V1.0	2025-04-23	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

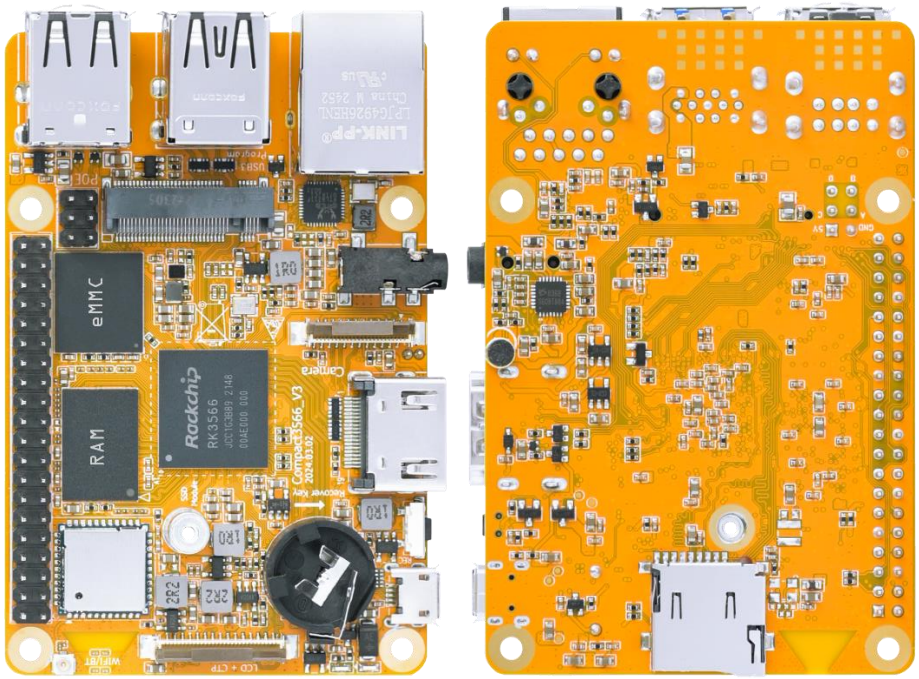
Content

1.Introduction

1.1 Overview

Compact3566 is a Raspberry Pi 3 B+ form factor single board computer (SBC) based on the Rockchip RK3566 Quad-core Cortex-A55 processor designed for IoT devices, such as home security system, Face Recognizing Robot, drones and HMI.

The small computing platform has a high-performance and low power processor with USB ports, Gigabit Ethernet, 2.4G/5G WiFi, M.2, micro-SD card slot, MIPI CSI camera connector, LVDS (or MIPI DSI) and HDMI port that support 4K monitors.



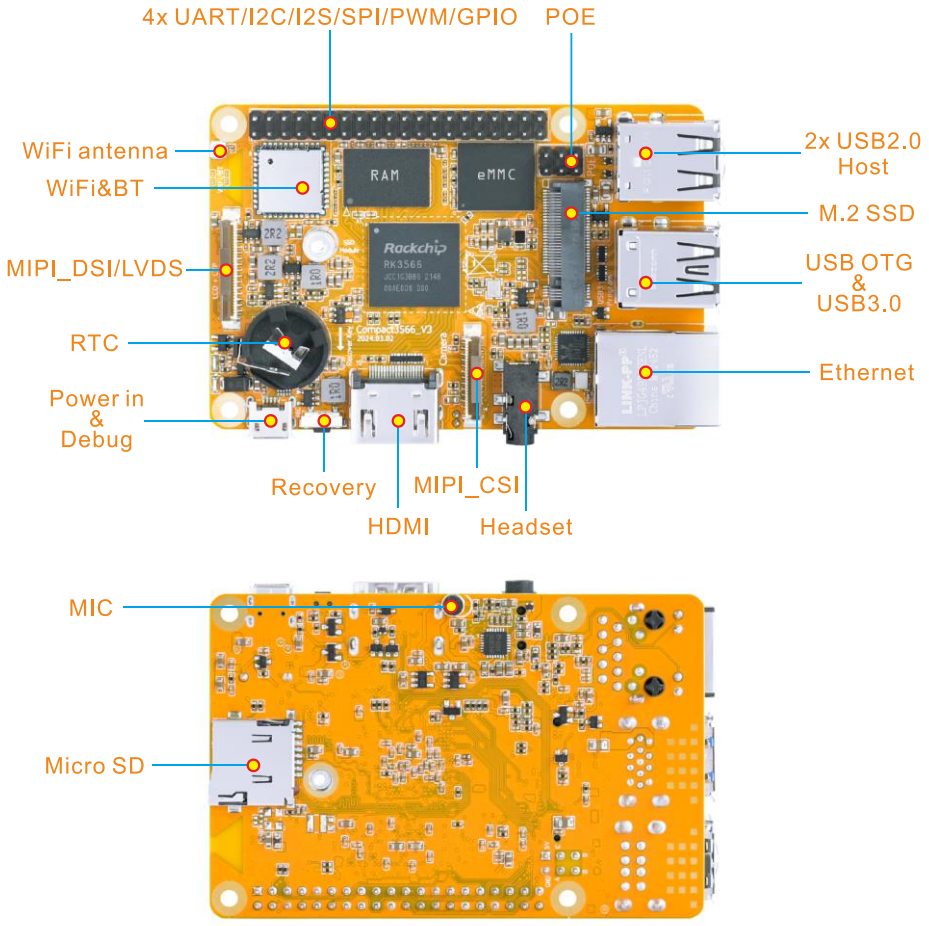
1.2 Product Parameters

Basic Parameters	
SOC	• RK3566
CPU	• Quad-core 64-bit ARM Cortex-A55@ up to 1.8GHz
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 MAC operations
Video	Decoder	• Support 4096x2306@60fps H.265/H.264/VP9 video decoding • Support 1920x1088@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		• 4GB LPDDR4X (up to 8GB)
ROM		• 8GB eMMC (up to 64GB)
Support system		Android, Debian, Buildroot
Hardware Parameters		
Extended Storage		• Support M.2 PCIe NVME SSD • Support MicroSD Card
Display		• Support HDMI TX 4K@60fps display • Support LVDS display
Audio		• Support HDMI TX audio output • Support Headphone output/input • Support MIC input
USB		• Support USB3.0 • Support 3x USB2.0
Network		• Support Gigabit Ethernet • Support WIFI/BT module
Camera		• Support 1x Camera (ov13850)

Peripheral communication	• Support SPI • Support 4xUART
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	
dimensions	85.0mm x 56.0mm

1.3 Hardware Interface Introduction



Interface parameters	
Power in & Debug	Micro USB interface, integration of power supply and serial port debugging
Recover	Recovery key
HDMI	HDMI TX interface
MIPI_CSI	MIPI CSI Camera interface
Headset	Headset output/input
Ethernet	Gigabit Ethernet interface
USB OTG&USB3.0	Dual-layer USB HOST interface
M.2 SSD	M.2 SSD interface
2X USB2.0 Host	Dual-layer USB2.0 HOST interface
POE	Power Over Ethernet interface
4xUART/I2C/I2S/SPI/PWM/GPIO	Expand GPIO interface
WIFI antenna	WIFI antenna interface
WIFI&BT	Realtek RTL8821CS module
RTC	RTC coin cell connector
MIC	Microphone
Micro SD	MicroSD card slot

2.Install Drivers and Tool

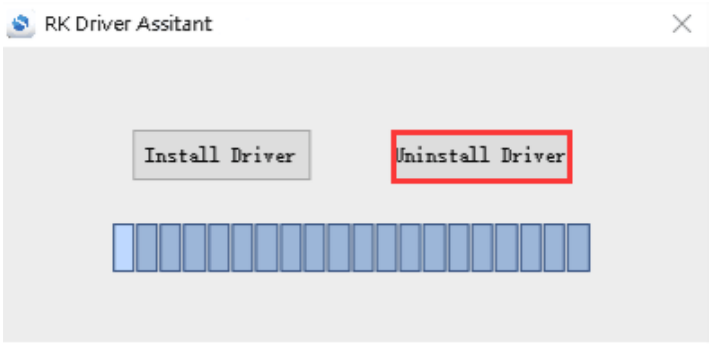
To download firmware and debug in the terminal, the following drivers and software need to be installed (for Windows computers):

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

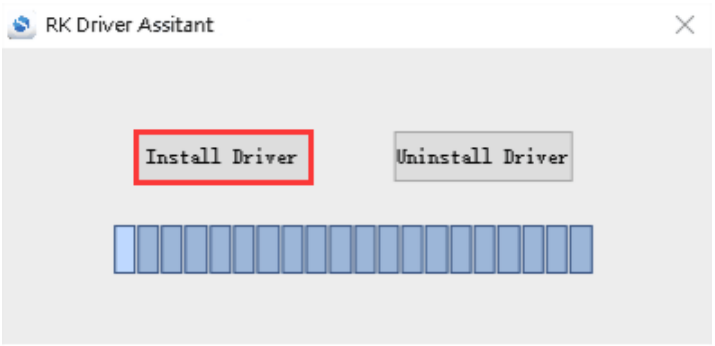
2.1 Install RK Driver Assitant

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

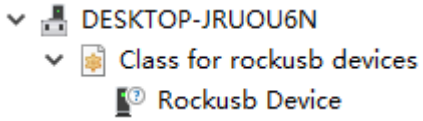
Step 2: To avoid driver conflicts, click **"Uninstall Driver"** to uninstall the driver.



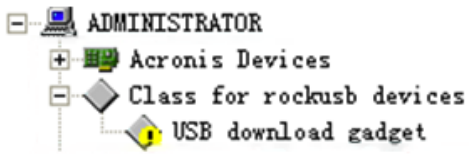
Step 3: Click button **"Install Driver"** to install.



Step 4: After the installation is complete, connect the board and PC with Type-A 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*.

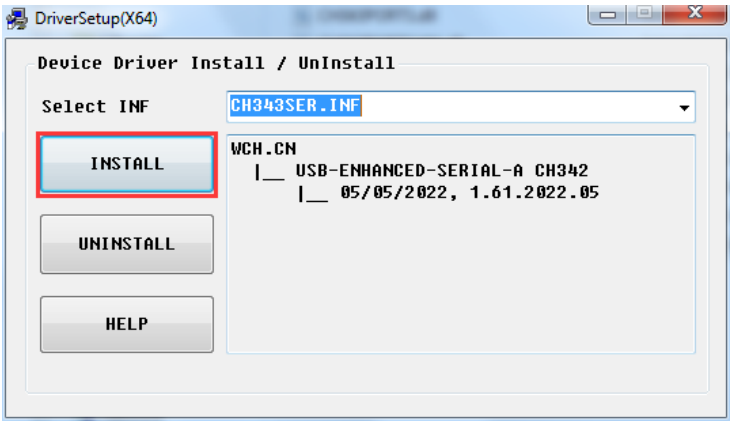
Step 7: After the installation is completed, the following device information can be seen in the Computer **Device Manager**.

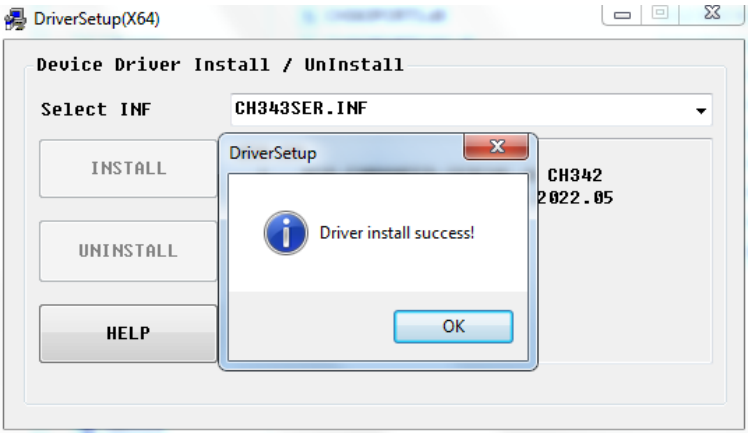
2.2 Install CH9102X Driver

Step 1: Connect the Micro USB cable (for power and debugging).

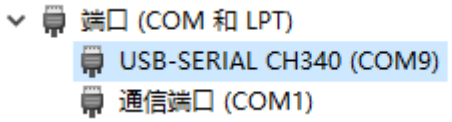
Step 2: Unzip *CH343SER.ZIP* on Windows.

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.

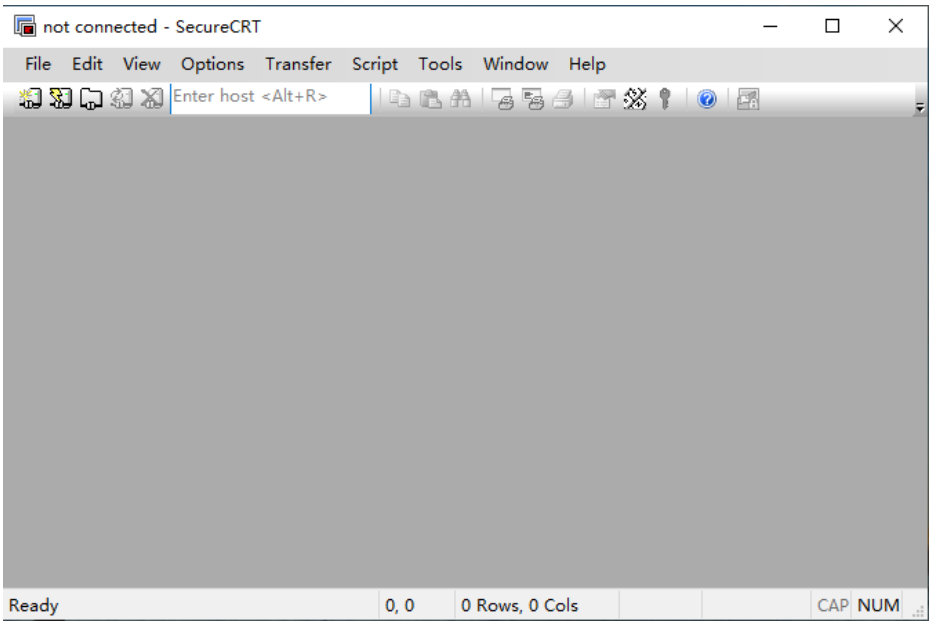


2.3 Install Serial Terminal Tool

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

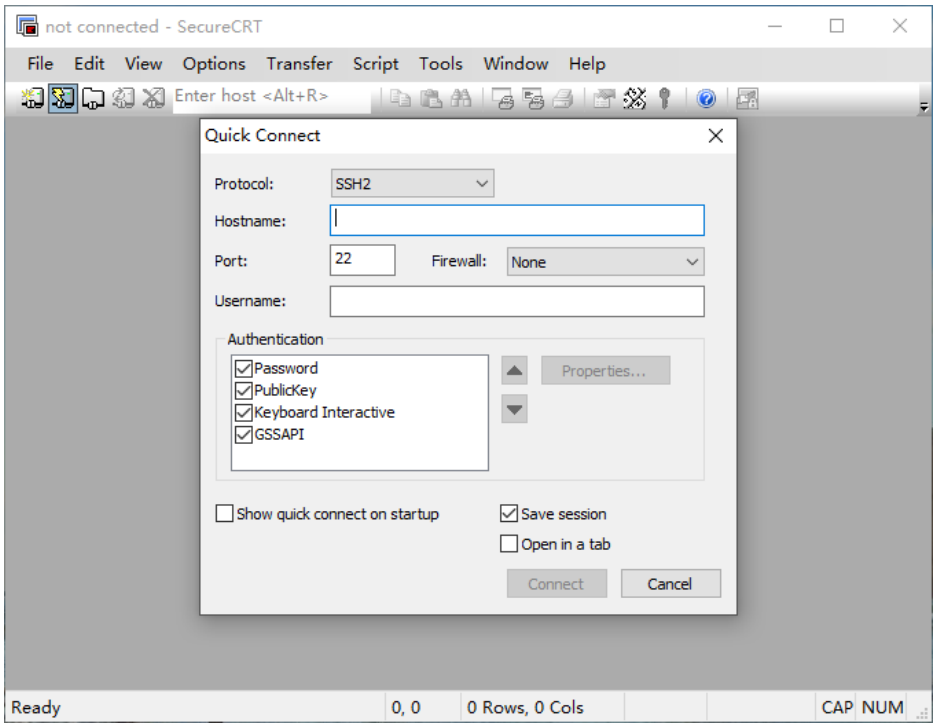
Step 1: Unzip *Platform/SecureCRT.rar* on PC.

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

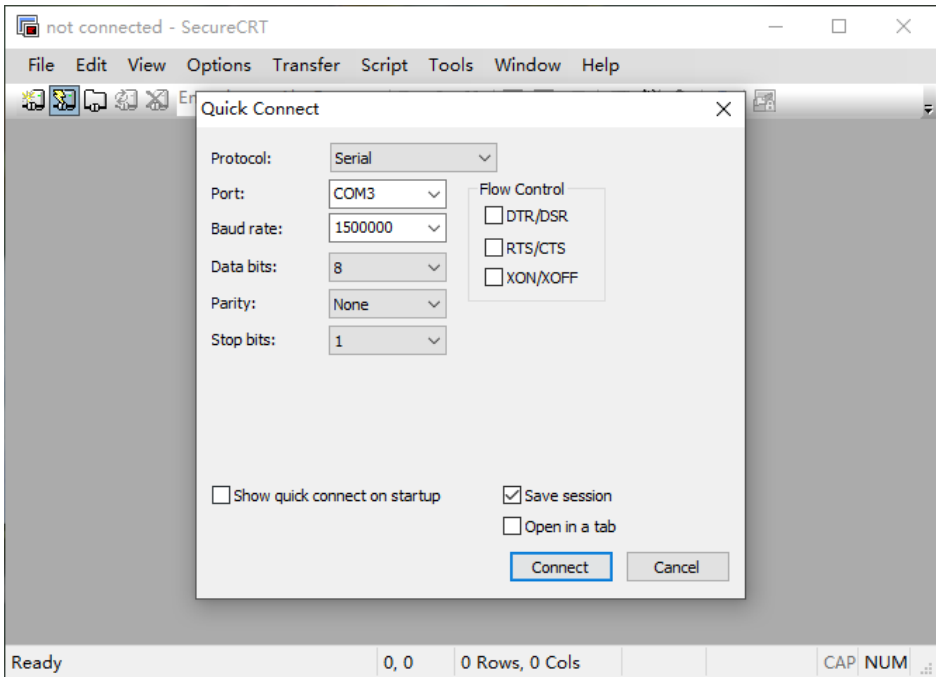


Step 3: Connect the Micro USB cable (for power and debugging).

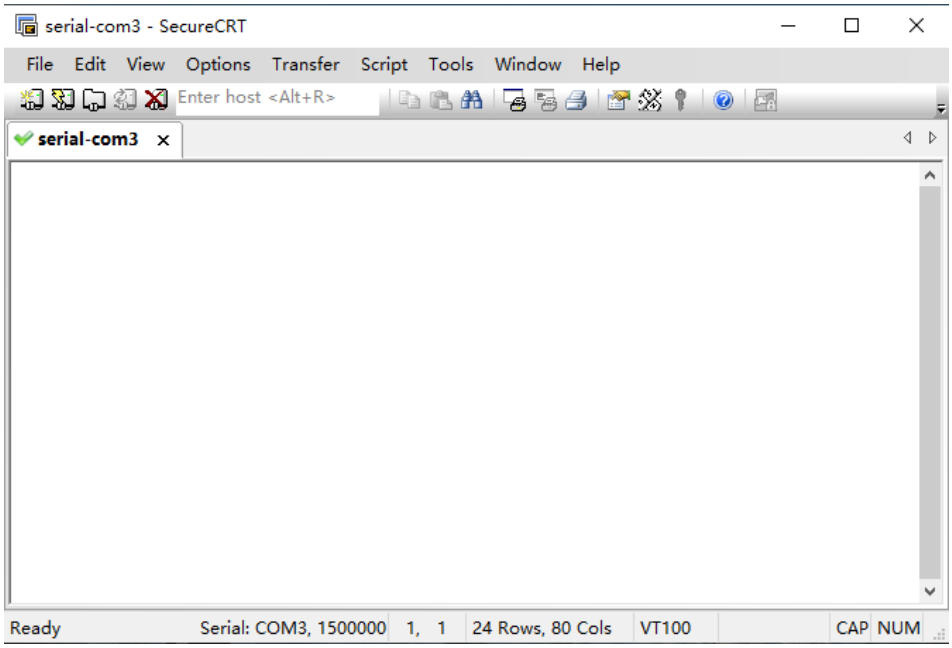
Step 4: Click the **"Quick Connect"** button to go to the Quick Connect configuration screen.



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

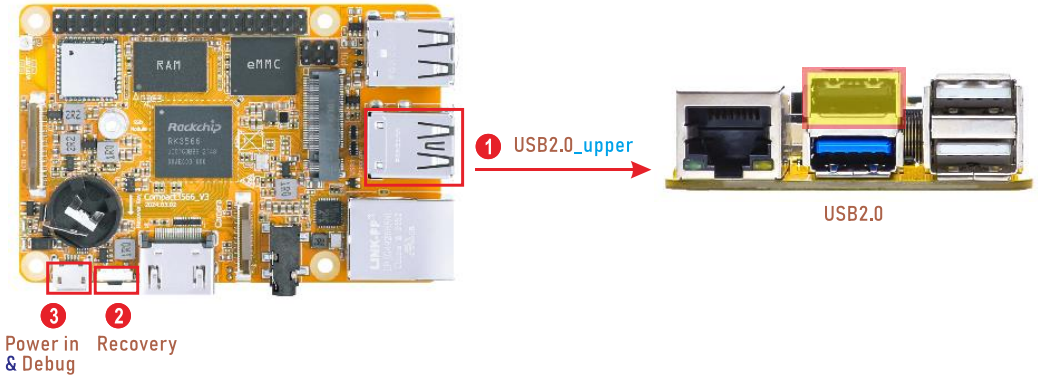
3.1.1.1 Hardware

Step 1: Disconnect the power adapter.

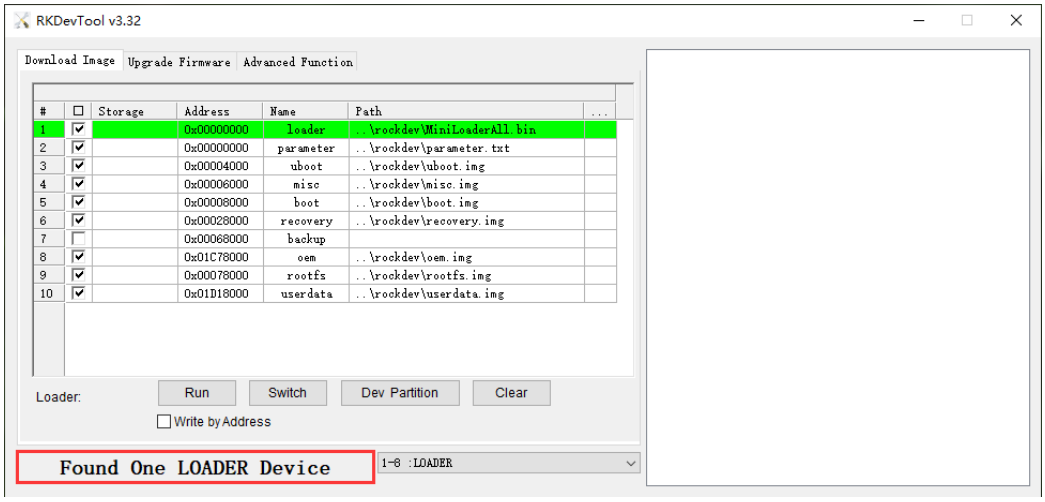
Step 2: Use a Type-A data cable to connect the personal computer and the development board.

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

Step 4: Connect the Micro USB cable (for power and debugging).



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

```
# reboot loader
```

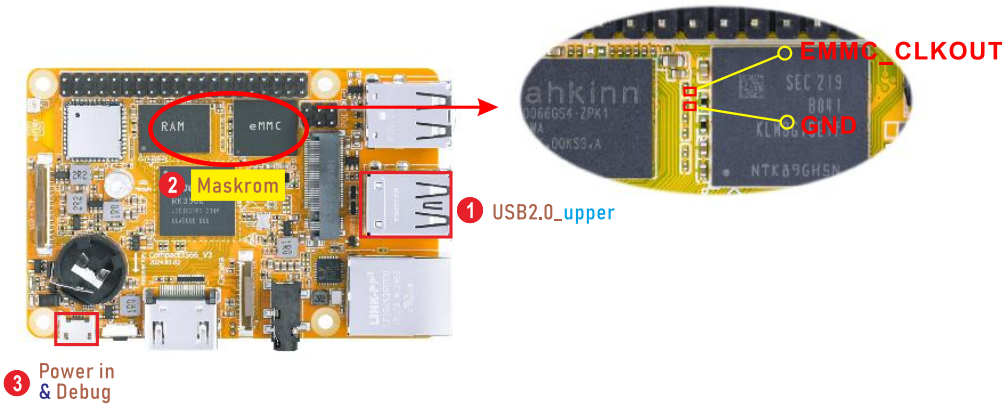
3.1.2 How to Enter MaskRom Mode

Step 1: Disconnect the power adapter.

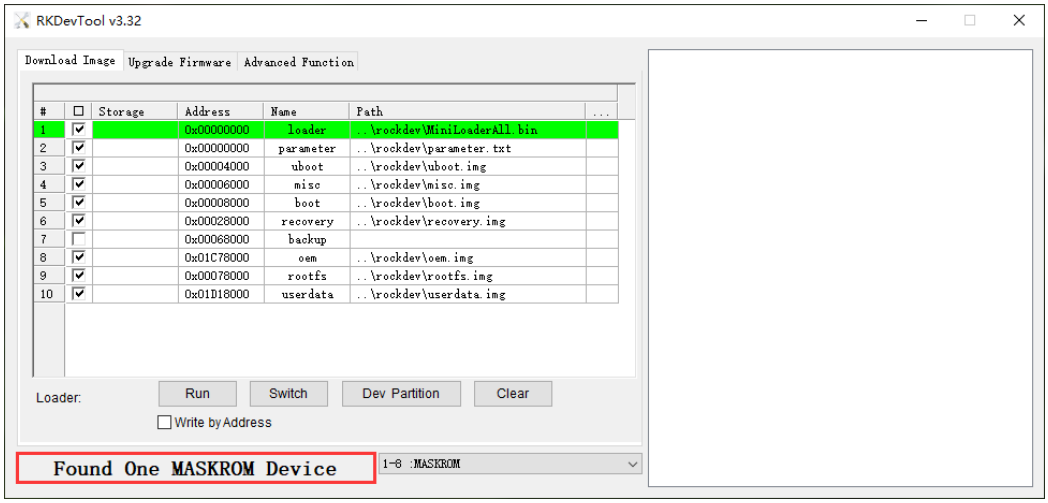
Step 2: Use a Type-A data cable to connect the personal computer and the development

board.

Step 3: Short EMMC_CLKOUT to GND.



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



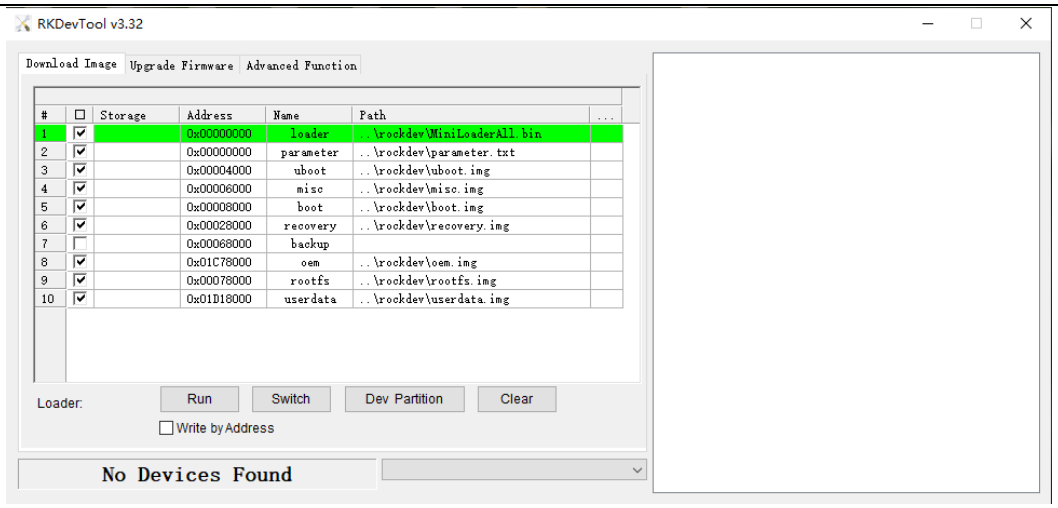
3.2 Burn firmware

Environment: Windows OS (Operating System).

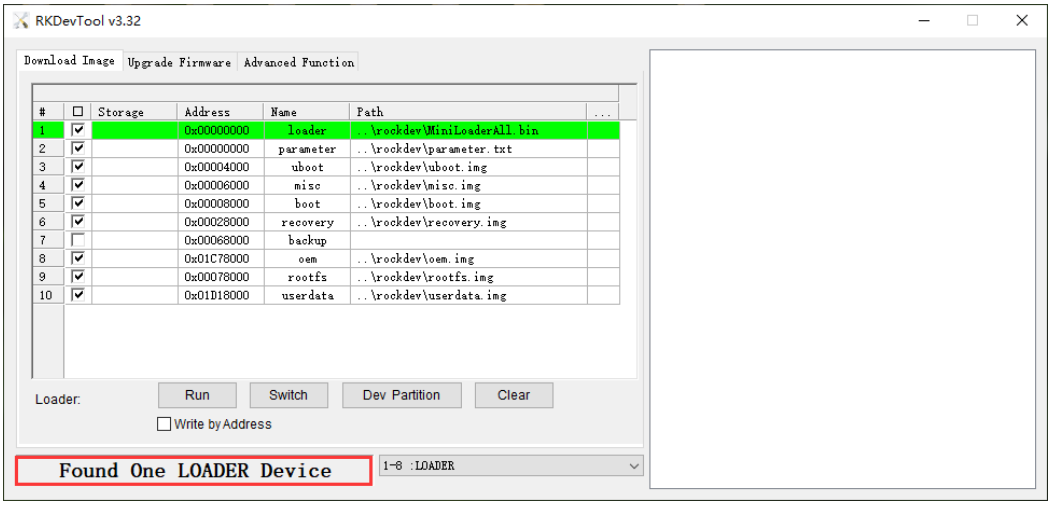
3.2.1 Burn Update.img Firmware

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

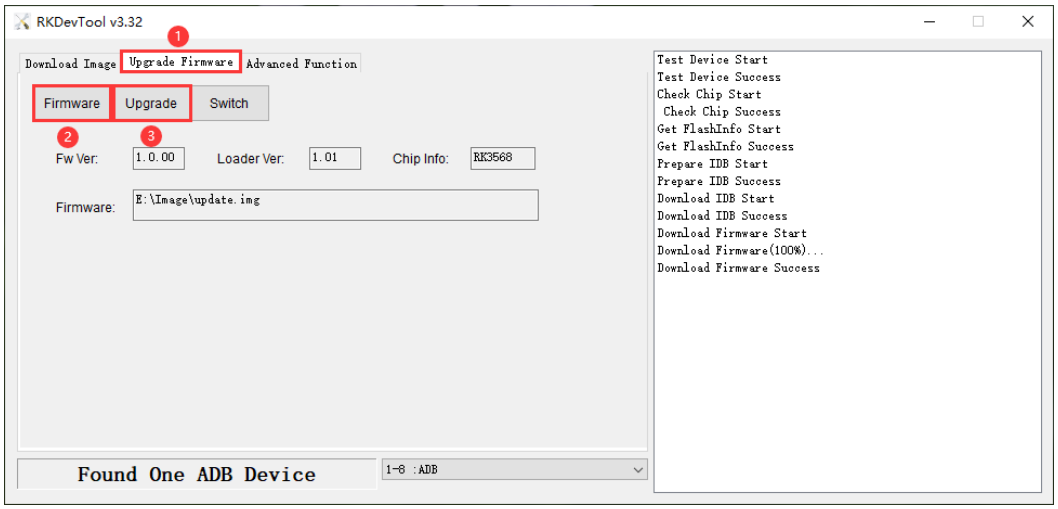
Step 2: Open *RKDevTool\RKDevTool_Release\RKDevTool.exe*.



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.

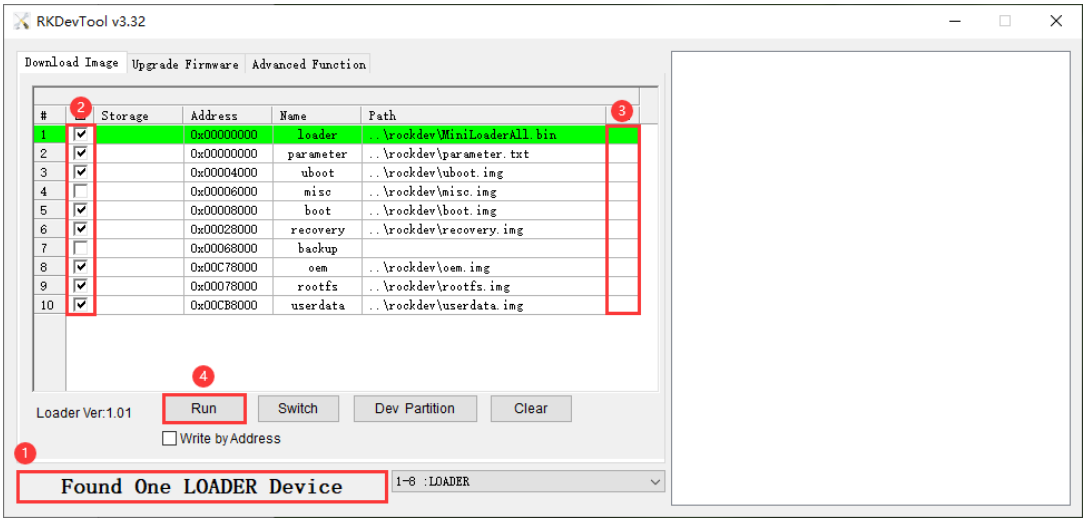
3.2.2 Burn Split Firmware

Step 1: Switch to **Loader mode**. (*How to Enter Loader Mode*)

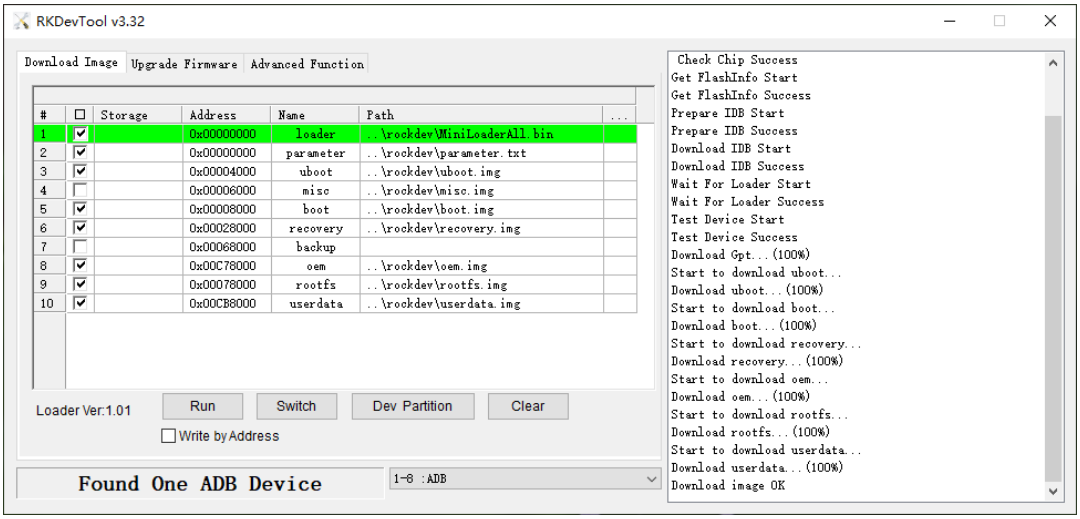
Step 2: Check the partitions to be flashed, multiple partitions can be selected.

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
```

5.Compile Source

Step 1: Unzip the Source

To extract the source files, execute the following commands:

```
$ tar xvf compact3566_android14-rkr*.tar.bz2
$ cd compact3566_android14-rkr6/
```

Step 2: Configure the Compiled Platform

To configure the board, execute:

```
$ source build/envsetup.sh
$ lunch rk3566_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-rk3566_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'
```

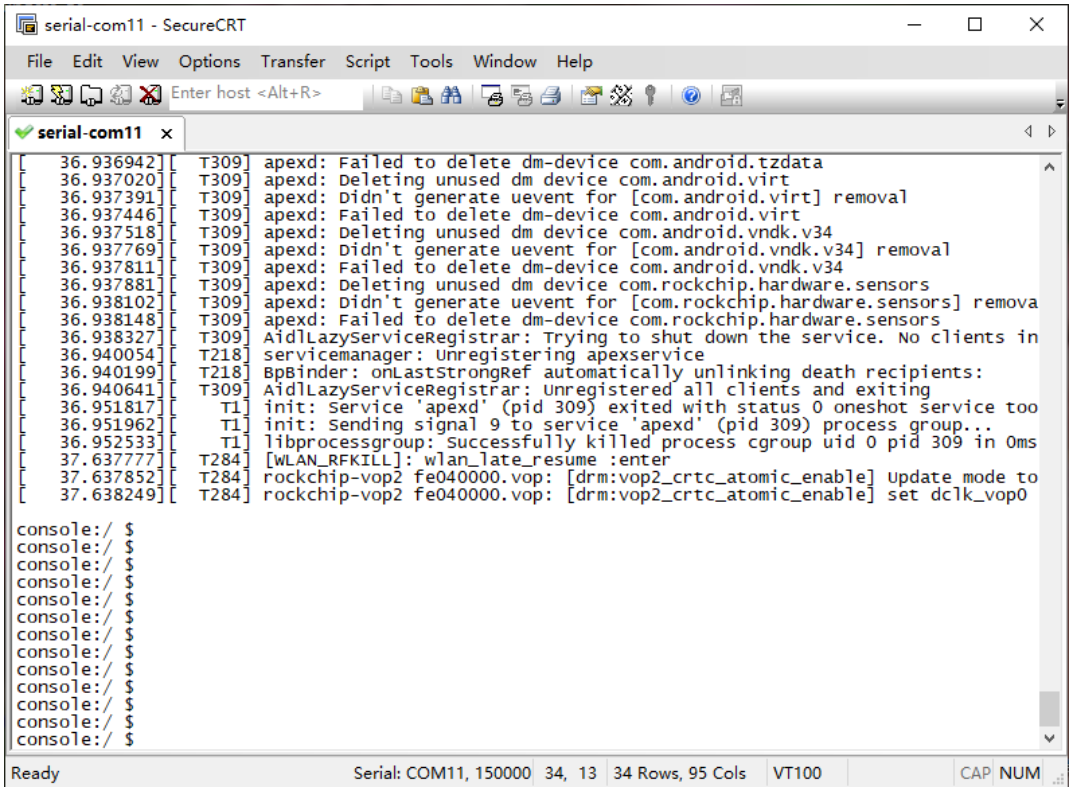
```
$ msk ARCH=arm64 rockchip_defconfig android-14.config rk356x.config
$ msk ARCH=arm64 BOOT_IMG=../rockdev/Image-rk3566_u/boot.img rk3566-evb2-lp4x-
v10.img -j32
```

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

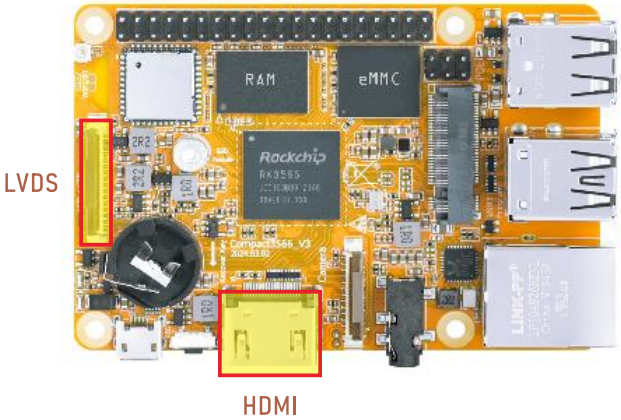
6.1 Serial Terminal

Connect the Micro USB cable (for power and debugging), the terminal will output boot information. The default baudrate is 1500000.

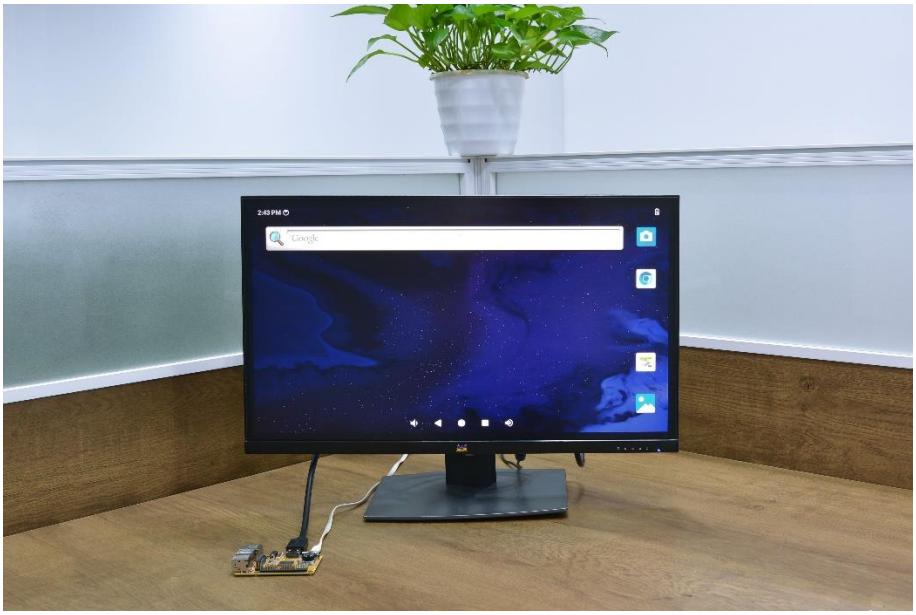


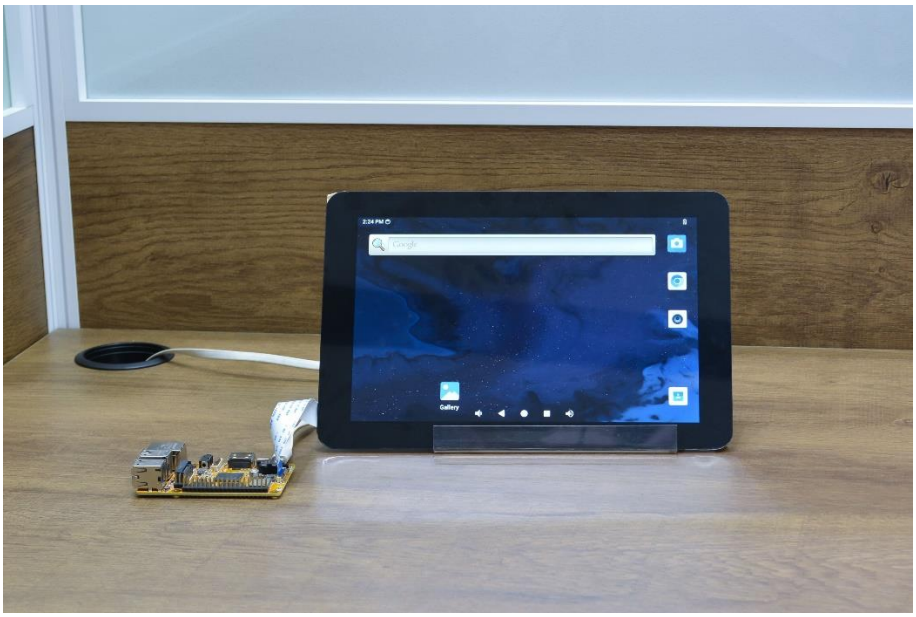
6.2 Display

The Compact566 can only output to a single display (either HDMI or LVDS) at a time.



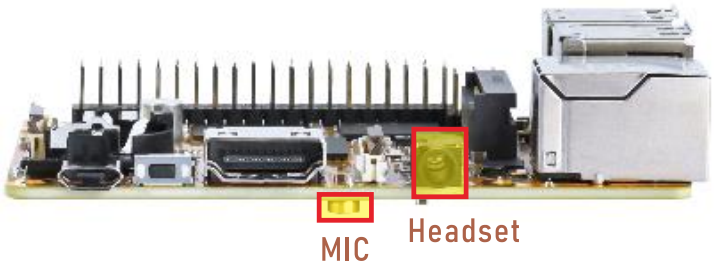
The display effect diagram is as follows:



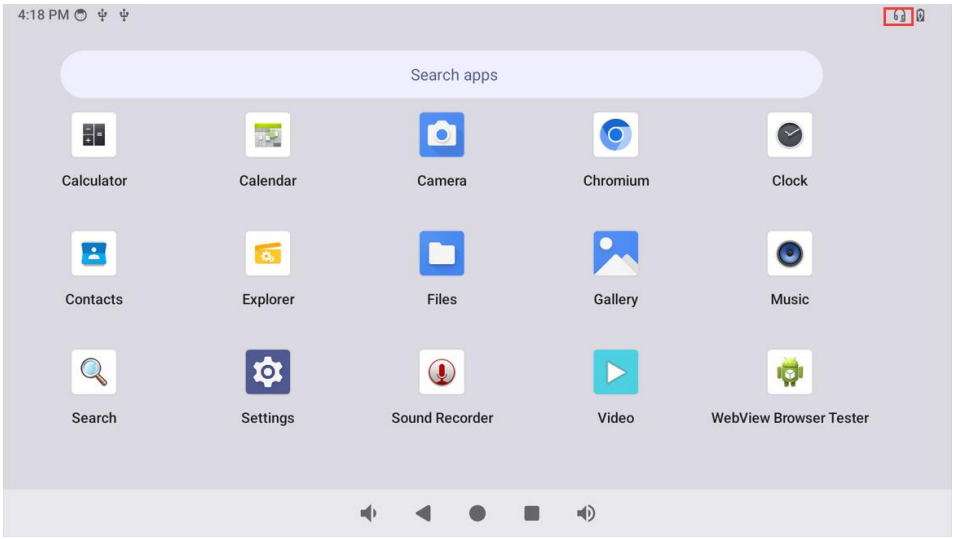


6.3 Audio

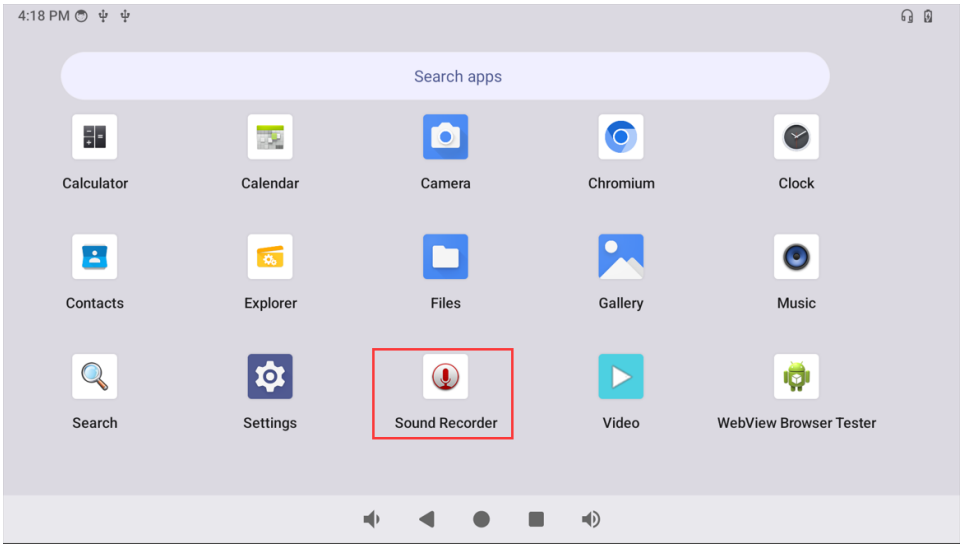
Step 1: Plug the headset into the headset jack.



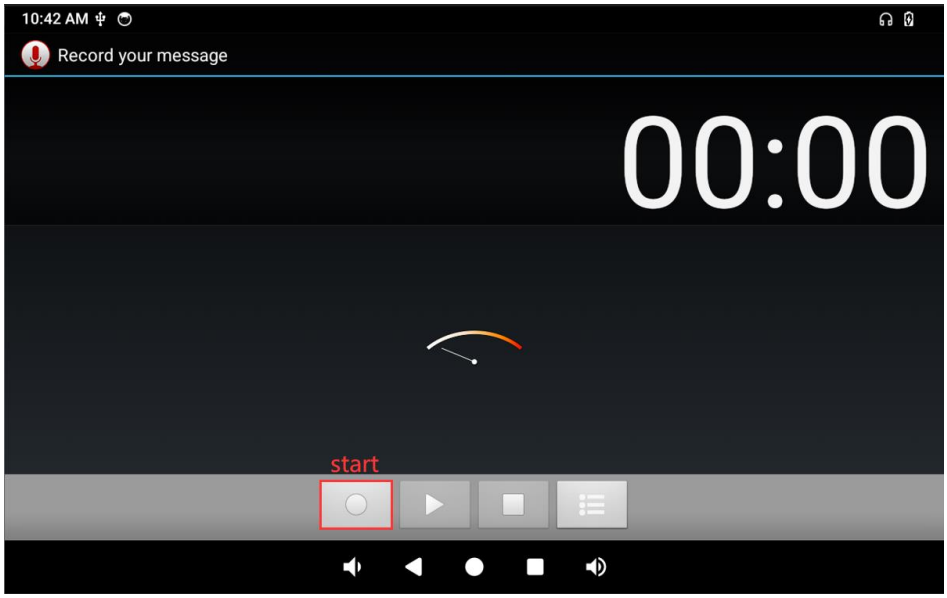
Status After Headset Insertion:



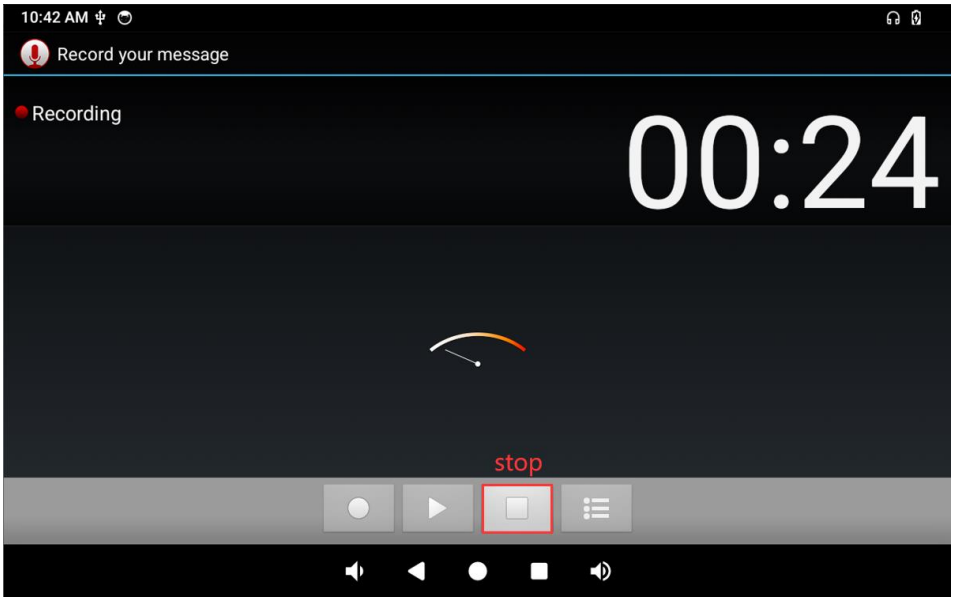
Step 2: Open the **Sound Recorder** app.



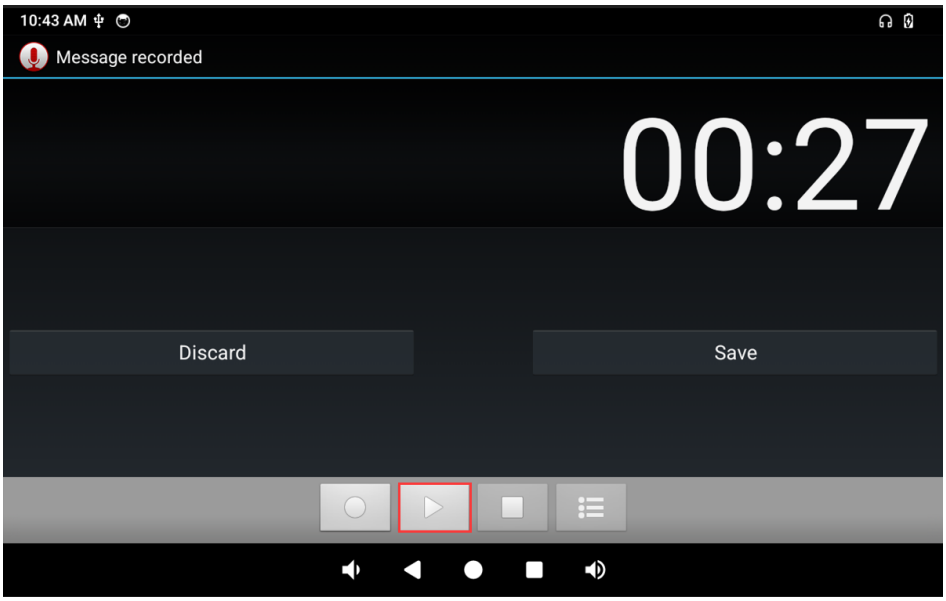
Step 3: Click the following button to start recording.



Step 4: Click the following button to stop recording.



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



Note:

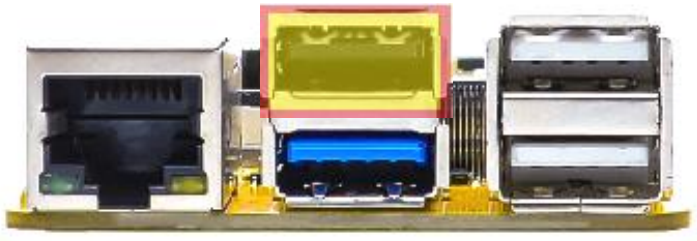
• Audio input level: Headset > MIC.

• Audio output level: Headset > HDMI.

6.4 USB

6.4.1 USB OTG

Compact3566 OTG defaults to Device mode on startup.



USB OTG

• Use the following command to switch to Host mode:

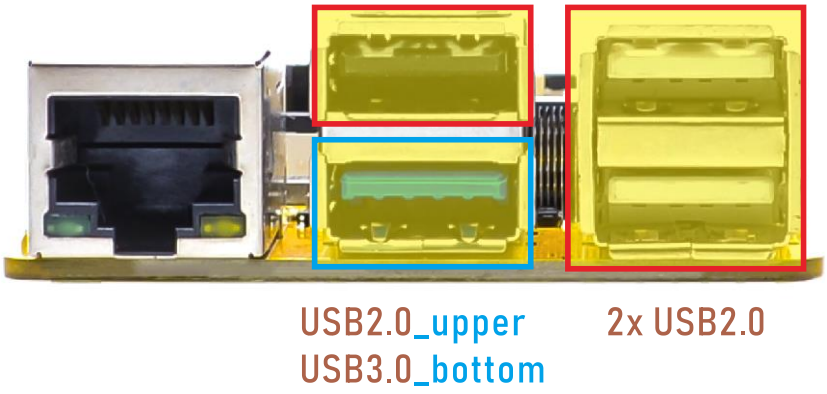
```
# echo host > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

• Use the following command to switch to Device mode:

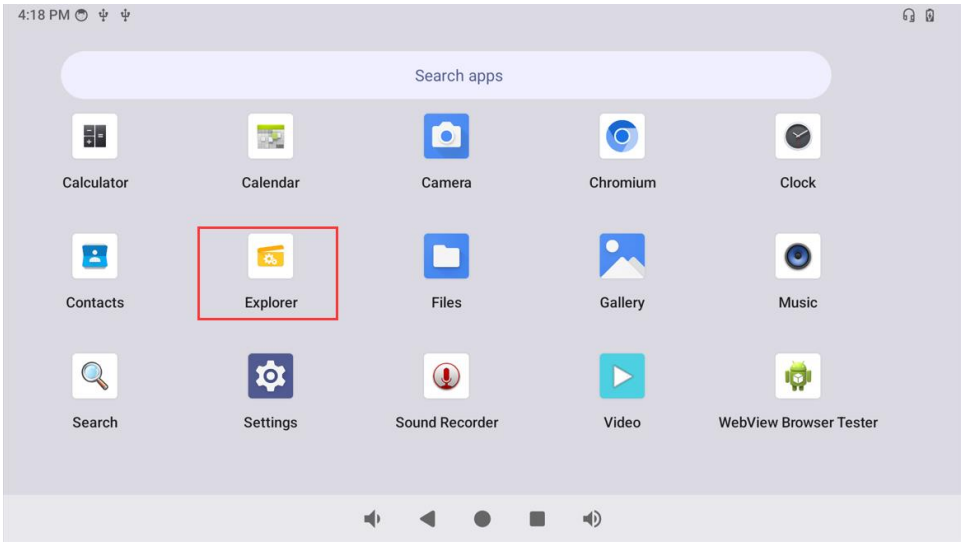
```
# echo peripheral > /sys/devices/platform/fe8a0000.usb2-phy/otg_mode
```

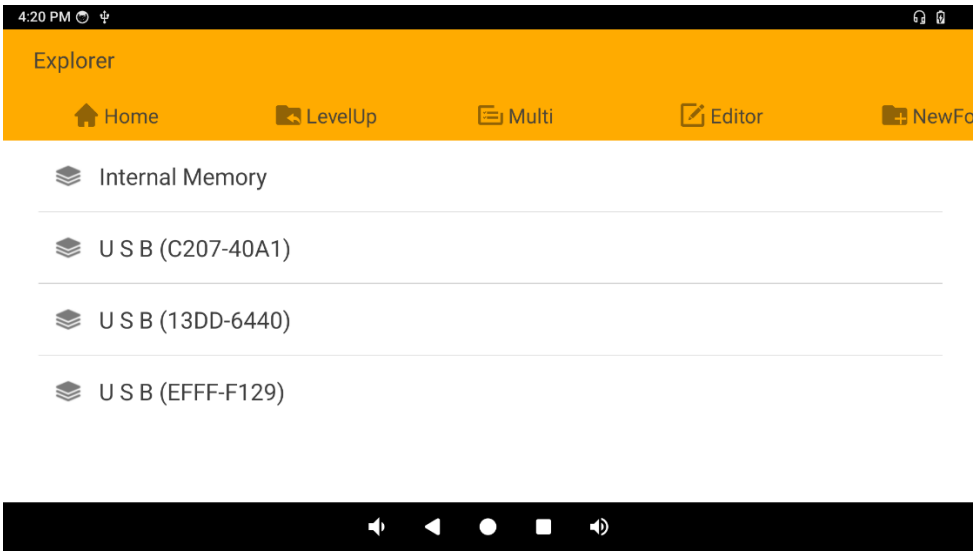
6.4.2 USB HOST

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.





The user can identify whether the mounted flash drive is USB 2.0 or USB 3.0 from the debug log.

• USB2.0 print information in high-speed mode.

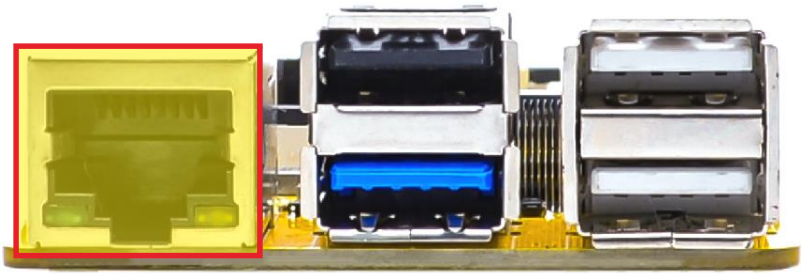
```
console:/ $ [   52.641257][  T116] usb 3-1: new high-speed USB device number 3 using ehci-platform
[   52.800305][  T116] usb 3-1: New USB device found, idVendor=346d, idProduct=5678, bcdDevice= 3.20
[   52.800430][  T116] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   52.800480][  T116] usb 3-1: Product: HIKSEMI
[   52.800555][  T116] usb 3-1: Manufacturer:
[   52.800598][  T116] usb 3-1: SerialNumber: FC04626F19B51
[   52.805981][  T116] usb-storage 3-1:1.0: USB Mass Storage device detected
[   52.808715][  T116] scsi host0: usb-storage 3-1:1.0
[   53.819576][   T10] scsi 0:0:0:0: Direct-Access            HIKSEMI         3.20 PQ: 0 ANSI: 4
[   53.825198][   T54] sd 0:0:0:0: [sda] 245760000 512-byte logical blocks: (126 GB/117 GiB)
[   53.825563][   T10] sd 0:0:0:0: Attached scsi generic sg0 type 0
[   53.826285][   T54] sd 0:0:0:0: [sda] Write Protect is off
[   53.827251][   T54] sd 0:0:0:0: [sda] No Caching mode page found
[   53.827305][   T54] sd 0:0:0:0: [sda] Assuming drive cache: write through
```

• USB3.0 print information in SuperSpeed mode.

```
130|console:/ $ [  110.565729][  T116] usb 2-1: new SuperSpeed USB device number 2 using xhci-hcd
[  110.586162][  T116] usb 2-1: LPM exit latency is zeroed, disabling LPM.
[  110.587354][  T116] usb 2-1: New USB device found, idVendor=346d, idProduct=5678, bcdDevice= 3.20
[  110.587458][  T116] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[  110.587509][  T116] usb 2-1: Product: HIKSEMI
[  110.587551][  T116] usb 2-1: Manufacturer:
[  110.587592][  T116] usb 2-1: SerialNumber: FC04626F19B51
[  110.596706][  T116] usb-storage 2-1:1.0: USB Mass Storage device detected
[  110.602854][  T116] scsi host0: usb-storage 2-1:1.0
[  111.610119][   T10] scsi 0:0:0:0: Direct-Access            HIKSEMI         3.20 PQ: 0 ANSI: 4
[  111.622689][   T54] sd 0:0:0:0: [sda] 245760000 512-byte logical blocks: (126 GB/117 GiB)
[  111.622935][   T10] sd 0:0:0:0: Attached scsi generic sg0 type 0
[  111.623362][   T54] sd 0:0:0:0: [sda] Write Protect is off
[  111.623763][   T54] sd 0:0:0:0: [sda] No Caching mode page found
```

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.

```
130|console:/ $ [  244.733209][  T110] rk_gmac-dwmac fe010000.ethernet eth0: Link is Up - 1Gbps/Full -
flow control rx/tx
[  244.733446][  T110] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

Step 2: View network interface information.

```
# ifconfig
```

```
console:/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 46:93:f4:d6:c9:79  Driver rk_gmac-dwmac
          inet addr:192.168.0.12  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::b626:2bf0:324b:30a7/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:144 errors:0 dropped:0 overruns:0 frame:0
          TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16668 TX bytes:5009
          Interrupt:51
```

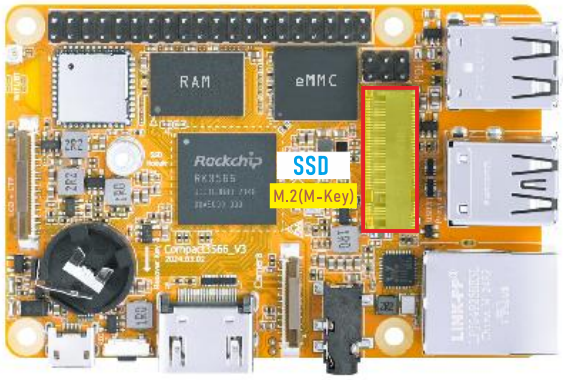
Step 3: Network connection test.

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

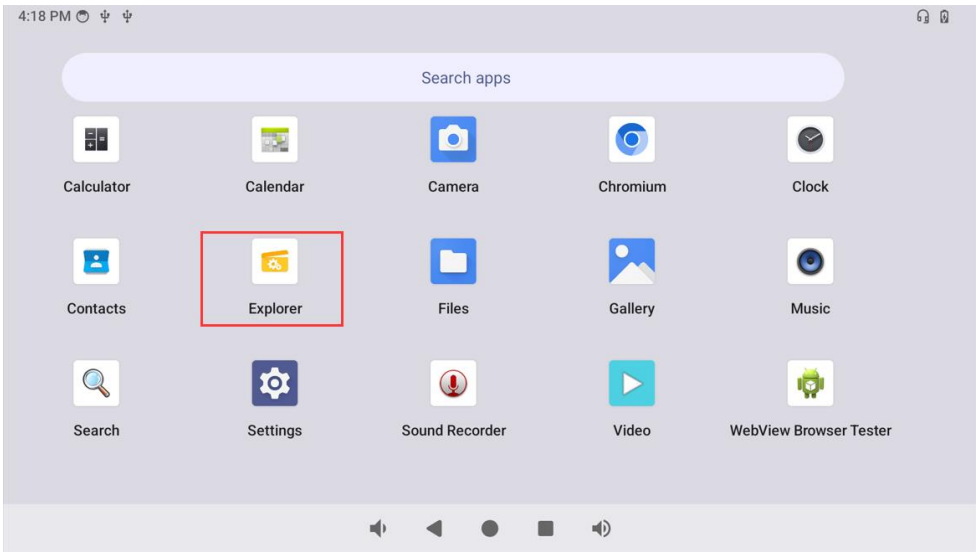
```
console:/ # ping -I eth0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.12 eth0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=48 time=183 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=48 time=183 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=48 time=183 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=48 time=183 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=48 time=183 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=6 ttl=48 time=183 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=7 ttl=48 time=183 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6003ms
```

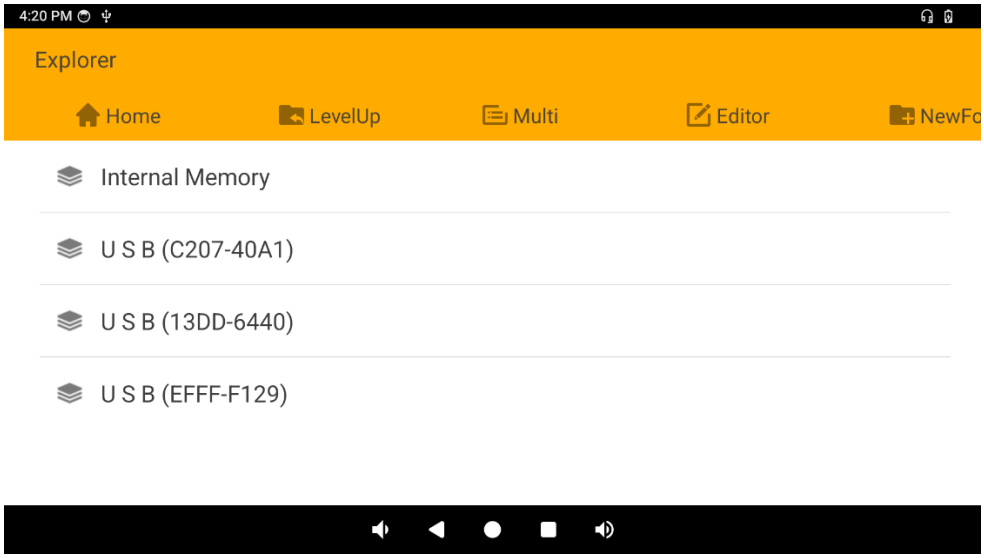
6.6 M.2 NVME SSD

Step 1: Connect the SSD, then power on.



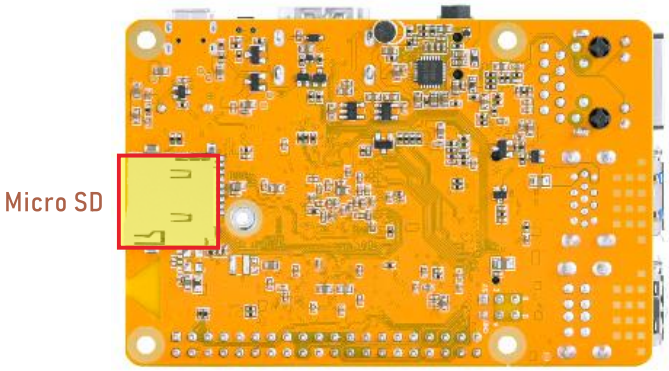
Step 2: If the SSD device is successfully recognized, the device will automatically mount, and it can be accessed directly through the Explorer app.



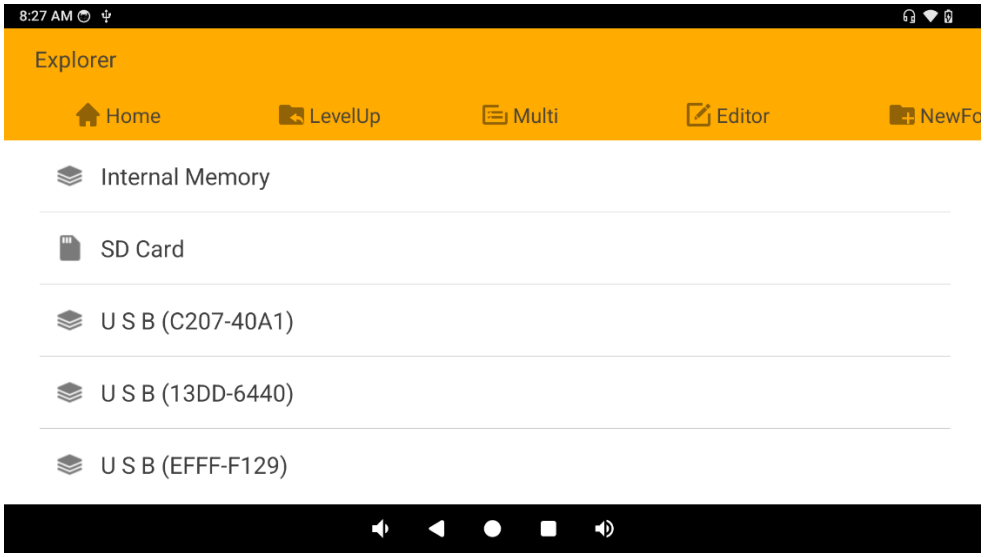


6.7 SD Card

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

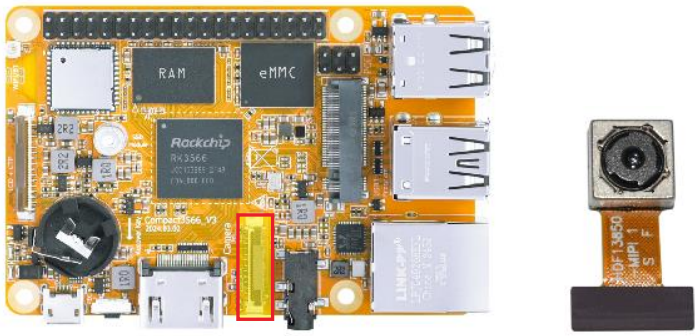


Step 2: After connecting the SD card drive, the device will automatically mount, and it can be accessed directly through the Explorer app.

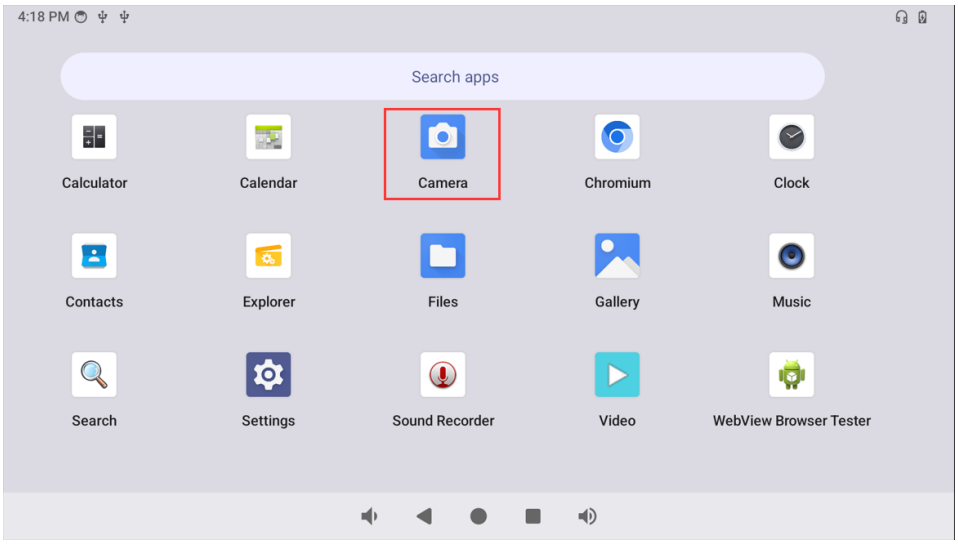


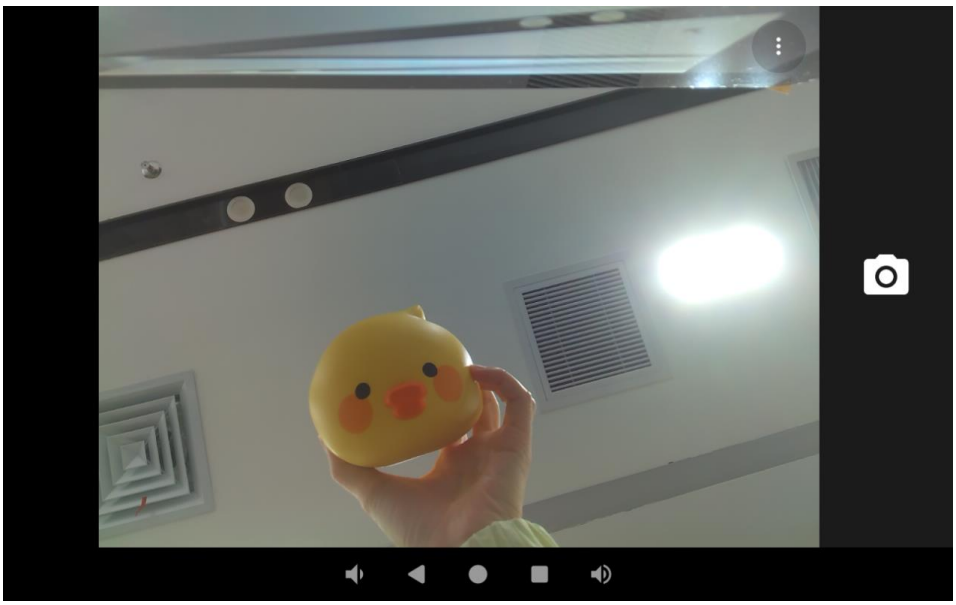
6.8 Camera

Step 1: Connect the camera (OV13850), and then power on the board.

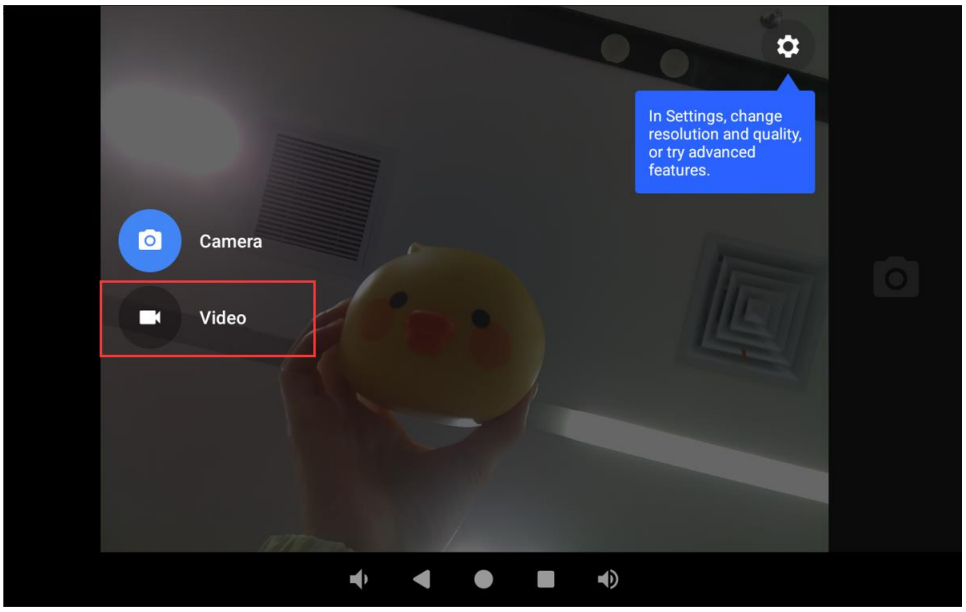


Step 2: Open the Camera app.

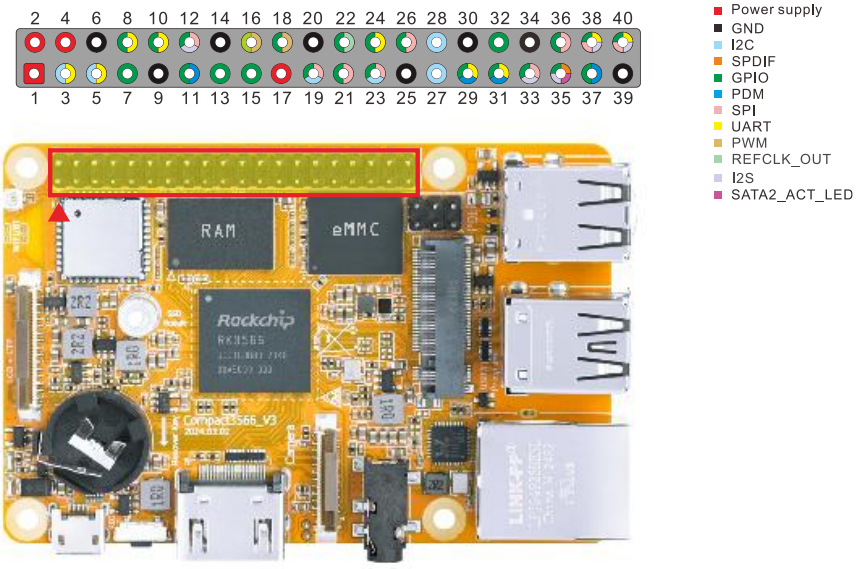




Step 6: Switch to video recording mode.

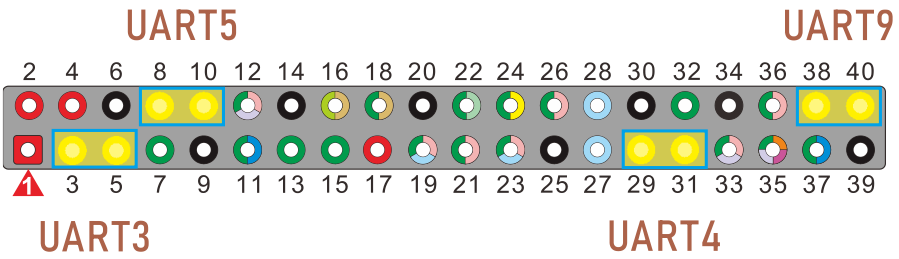

In Settings, change resolution and quality, or try advanced features.
Camera
Video

6.9 GPIO



6.9.1 UART

Short circuit RX and TX pins of UART.



• UART3 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
klklh
RECV: klklh
5555
RECV: 5555
jkuiui
RECV: jkuiui
```

• UART4 test:

```
# com /dev/ttyS4 115200 8 0 1
```

```
130|console:/ # com /dev/ttyS4 115200 8 0 1
port = /dev/ttyS4
baudrate = 115200
cs = 8
parity = 0
stopb = 1
52555555
RECV: 52555555
kjjjjjjj
RECV: kjjjjjjj
01010101010
RECV: 01010101010
```

• UART5 test:

```
# com /dev/ttyS5 115200 8 0 1
```

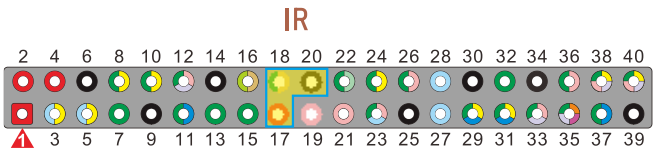
```
127|console:/ # com /dev/ttyS5 115200 8 0 1
port = /dev/ttyS5
baudrate = 115200
cs = 8
parity = 0
stopb = 1
jkjkjkjk
RECV: jkjkjkjk
uiuiuiui
RECV: uiuiuiui
8888888
RECV: 8888888
```

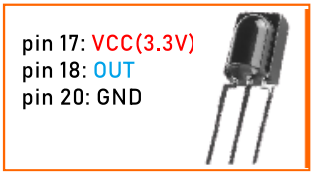
• UART9 test:

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

```
130|console:/ # com /dev/ttyS9 115200 8 0 1
port = /dev/ttyS9
baudrate = 115200
cs = 8
parity = 0
stopb = 1
oopopopopo
RECV: oopopopopo
666666
RECV: 666666
yytytytyty
RECV: yytytytyty
iii
RECV: iii
```

6.9.2 IR





Step 1: Open IR debugging print.

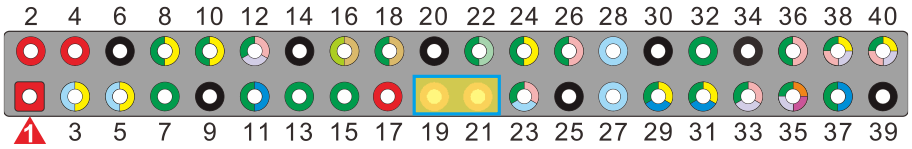
```
# echo 1 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
```

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:/ # [   75.164352][    C1] USERCODE=0x1818
[   75.191486][    C1] RMC_GETDATA=98
[   75.544325][    C1] USERCODE=0x1818
[   75.571457][    C1] RMC_GETDATA=98
[   76.008301][    C1] USERCODE=0x1818
[   76.035430][    C1] RMC_GETDATA=98
[   77.700606][    C1] USERCODE=0x1818
[   77.727784][    C1] RMC_GETDATA=9a
[   78.048390][    C1] USERCODE=0x1818
[   78.075554][    C1] RMC_GETDATA=99
[   78.436638][    C1] USERCODE=0x1818
[   78.463773][    C1] RMC_GETDATA=9b
[   80.664443][    C1] USERCODE=0x1818
[   80.691646][    C1] RMC_GETDATA=e6
```

6.9.3 SPI

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



SPI

Step 2: Execute the test script: **spidev0.0_test**.

```
# spidev-test
```

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

EE FF FF FF FF FF
C0 00 00 00 00 95
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
FF AD BF FF FE AD
F8 0D
```

6.10 RTC

Step 1: Install the coin cell battery.

Step 2: Set the system time.

```
# date -s "2025-04-11 16:02:00"
```

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

```
# hwclock -w
```

Step 4: Display the current hardware clock time.

```
# hwclock
```

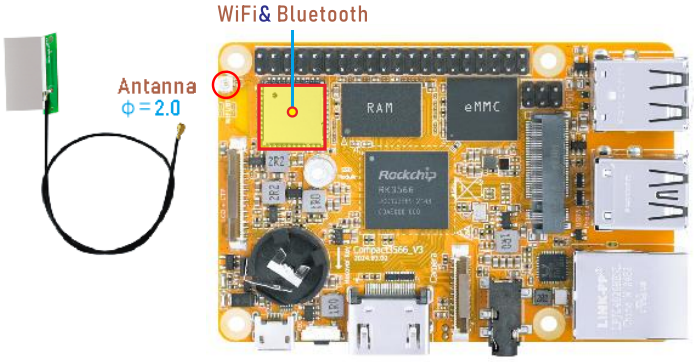
```
130|console:/ # date -s "2025-04-11 16:02:00"
Fri Apr 11 16:02:00 GMT 2025
console:/ # hwclock -w
console:/ # hwclock
2025-04-11 16:02:05+0000
console:/ # hwclock
2025-04-11 16:02:25+0000
console:/ # hwclock
2025-04-11 16:02:55+0000
console:/ # hwclock
2025-04-11 16:03:04+0000
```

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

```
console:/ # hwclock
2025-04-11 20:35:41+0000
console:/ # hwclock
2025-04-11 20:35:56+0000
console:/ # hwclock
2025-04-11 20:36:06+0000
console:/ # hwclock
2025-04-11 20:36:39+0000
console:/ # hwclock
2025-04-11 20:37:24+0000
```

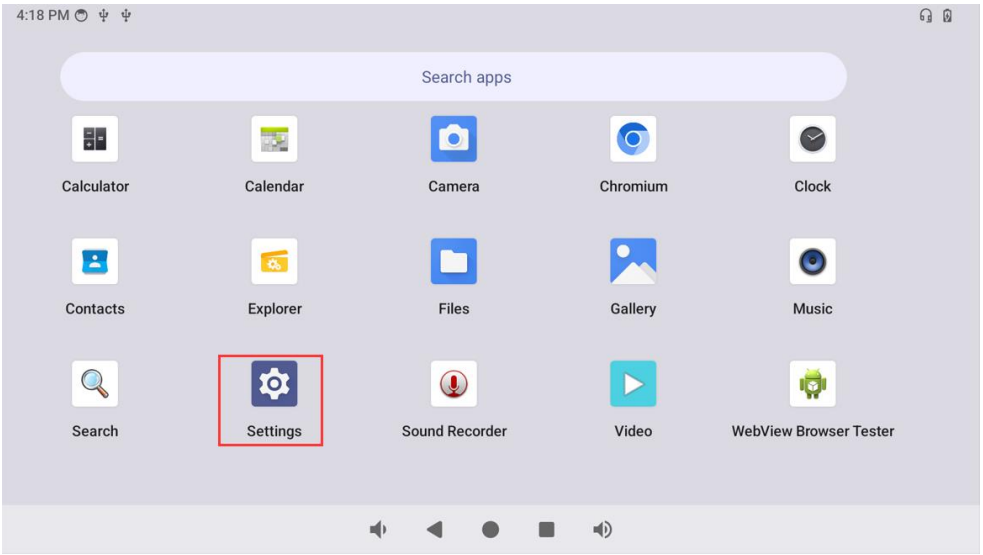
6.11 WiFi & Bluetooth

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

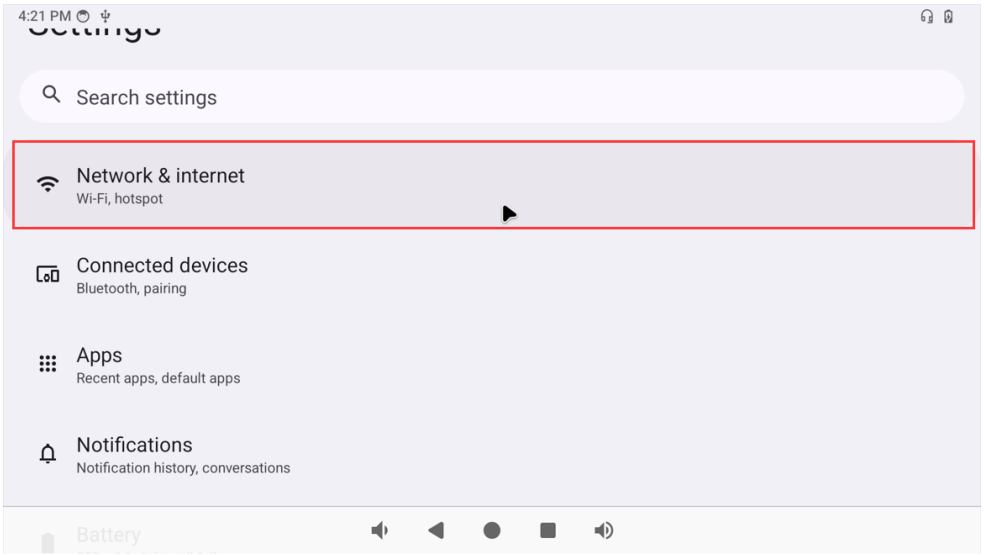


6.11.1 WiFi

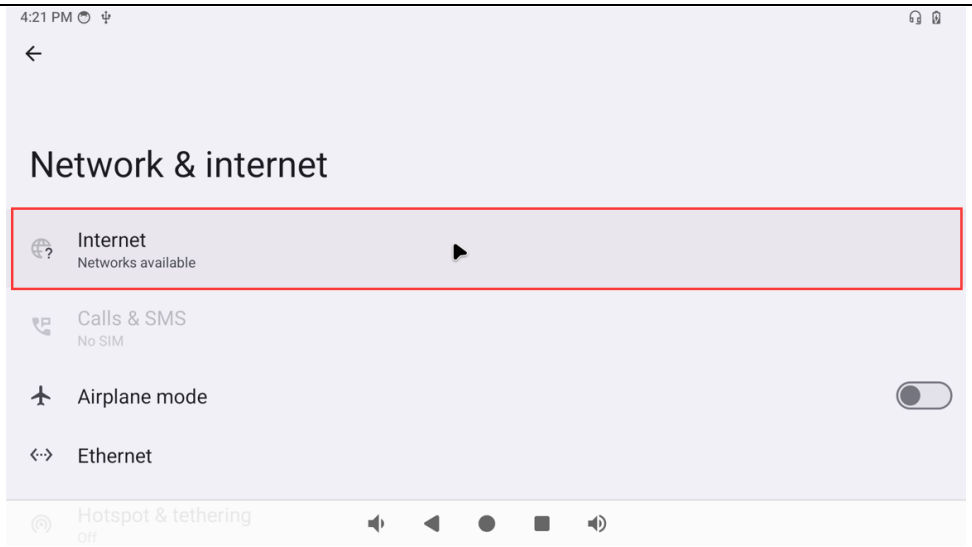
Step 1: Open the **Settings** app:



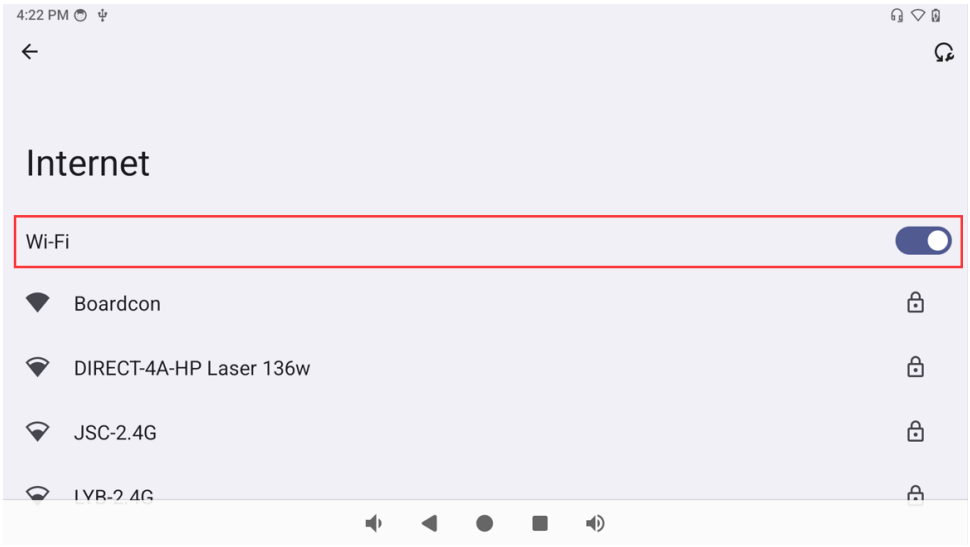
Step 2: Go to **Network & Internet**:



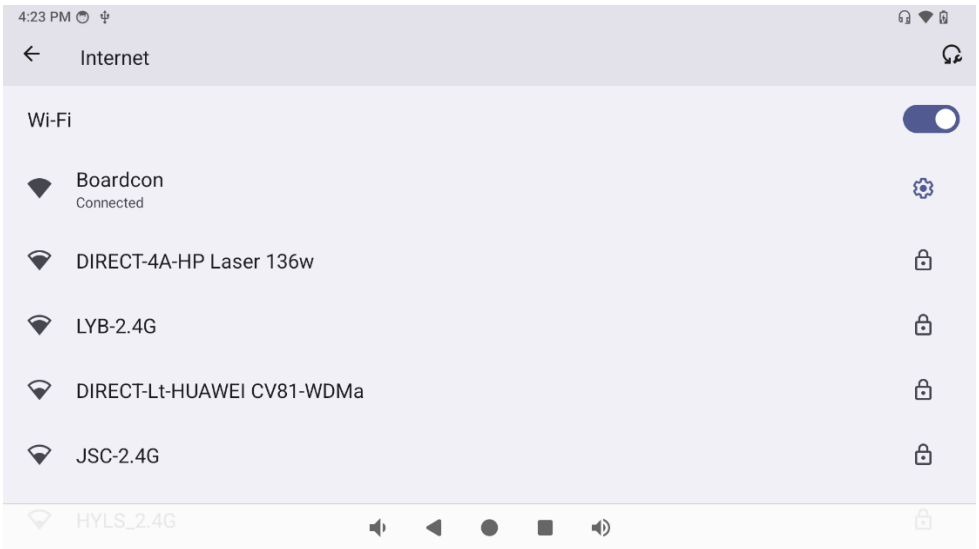
Step 3: Tap **Internet**:



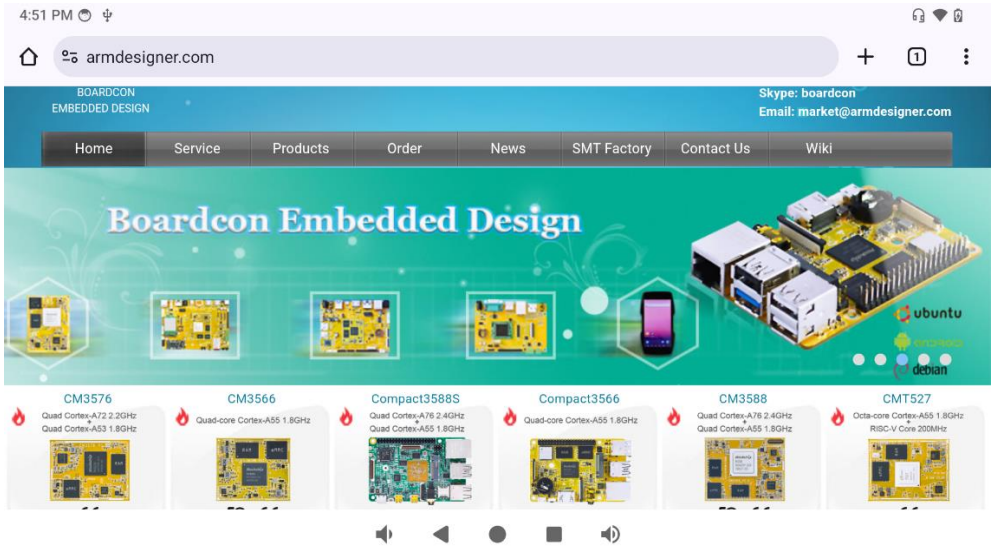
Step 4: Turn on the **Wi-Fi** switch:



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
```

```
127|console:/ # ifconfig
wlan0     Link encap:Ethernet  HWaddr 78:22:88:d9:62:91  Driver rtl8821cs
          inet addr:192.168.0.68  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::8fec:91b8:501:608e/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:210 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27283 TX bytes:8282
```

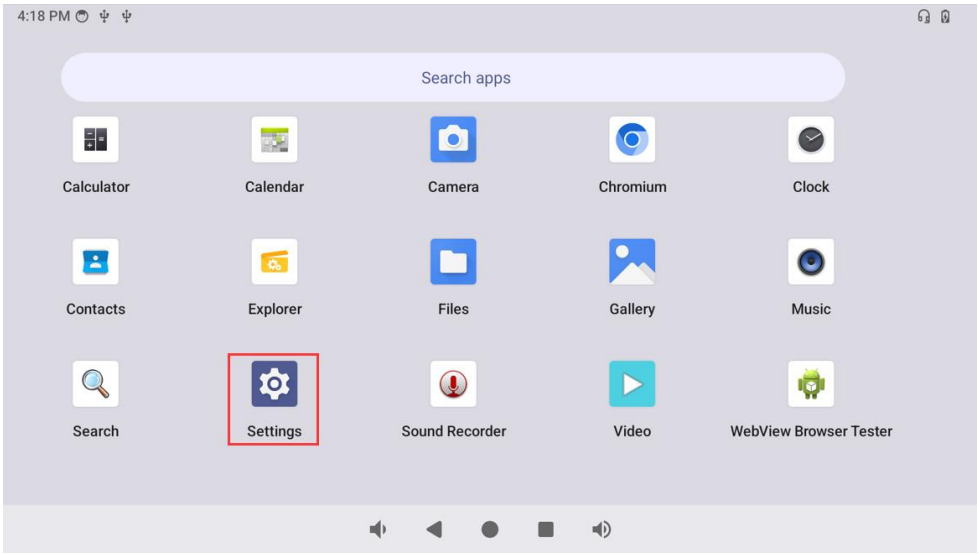
(2) Network connection test.

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

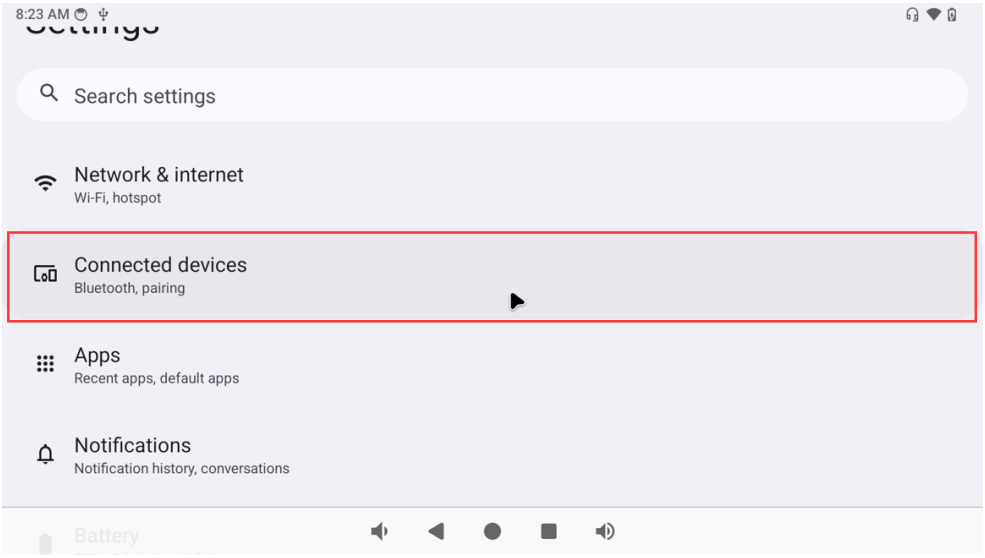
```
console:/ # ping -I wlan0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 192.168.0.68 wlan0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=1 ttl=48 time=185 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=2 ttl=48 time=184 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=3 ttl=48 time=184 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=4 ttl=48 time=185 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=5 ttl=48 time=481 ms
64 bytes from www.armdesigner.com (67.222.54.196): icmp_seq=6 ttl=48 time=185 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 6 received, 14% packet loss, time 6008ms
rtt min/avg/max/mdev = 184.336/234.656/481.228/110.273 ms
```

6.11.2 Bluetooth

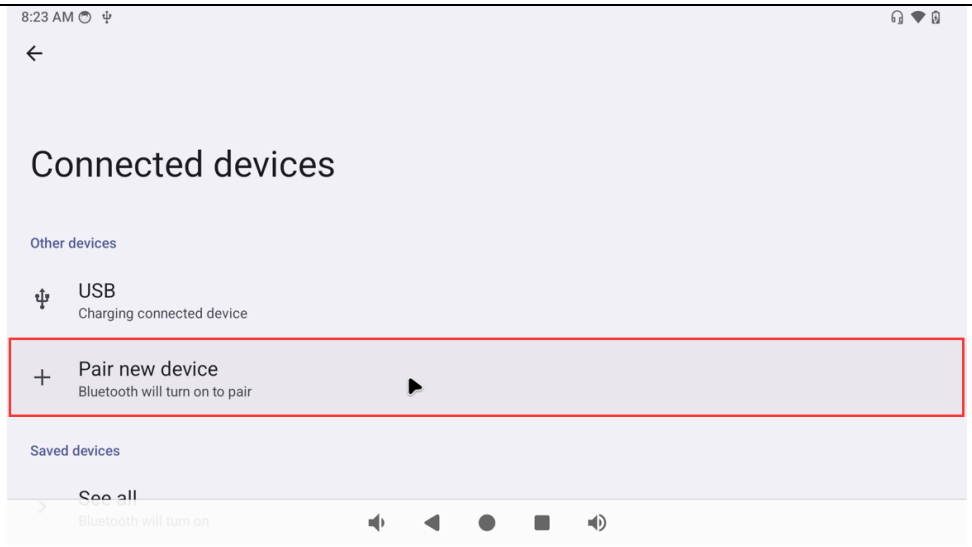
Step 1: Open the **Settings** app.



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



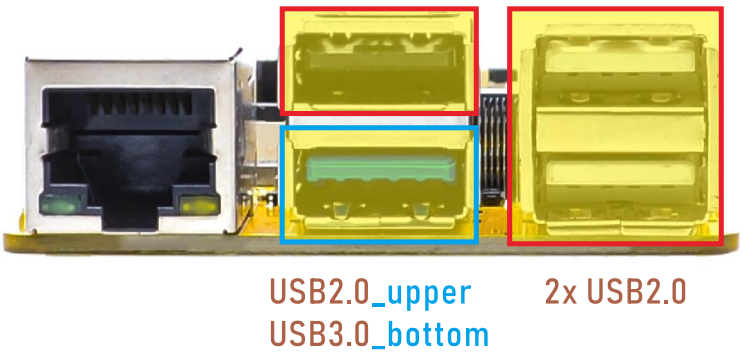
Step 3: Click on the option **Pair new device.**



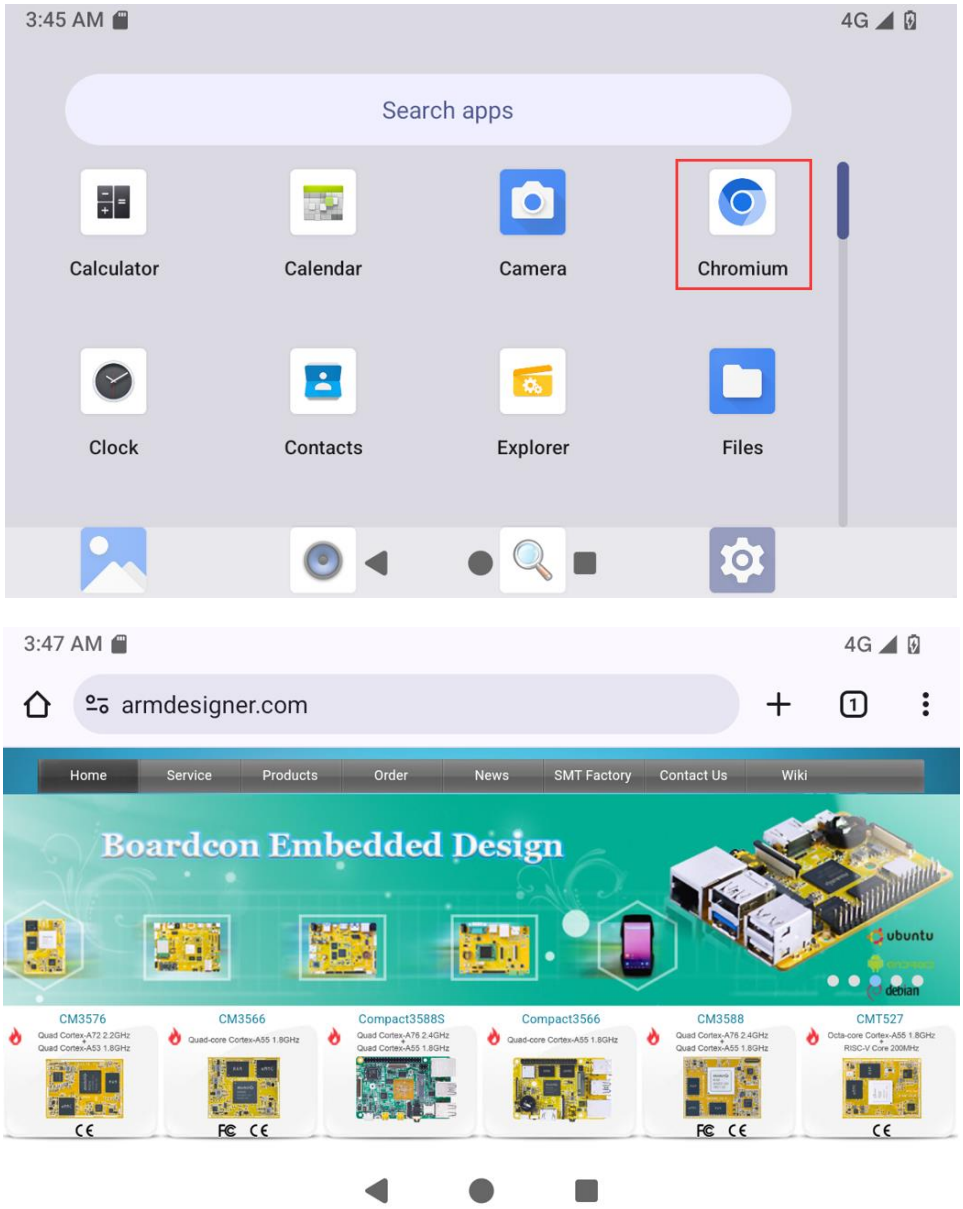
Step 4: User can pair themselves in the Bluetooth device list.

6.12 USB 4G

Step 1: Connect the USB 4G module to the USB host interface.



Step 2: 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
usb0      Link encap:Ethernet  HWaddr 02:0c:29:a3:9b:6d  Driver cdc ether
          inet addr:10.42.6.124  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::c:29ff:fea3:9b6d/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1922 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1863 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1832857 TX bytes:307399
```

(2) Network connection test.

```
# ping -I usb0 www.armdesigner.com
```

```
console:/ # ping -I usb0 www.armdesigner.com
PING www.armdesigner.com (67.222.54.196) from 10.42.6.124 usb0: 56(84) bytes of data.
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=1 ttl=46 time=352 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=2 ttl=46 time=316 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=3 ttl=46 time=276 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=4 ttl=46 time=555 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=5 ttl=46 time=516 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=6 ttl=46 time=475 ms
64 bytes from 67-222-54-196.unifiedlayer.com (67.222.54.196): icmp_seq=7 ttl=46 time=435 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6316ms
rtt min/avg/max/mdev = 276.154/418.256/555.862/97.843 ms
```