

Idea1126B-P Linux6.1 User Manual

V2.0



Boardcon Embedded Design

Overview

This document applies only to the Idea1126B-P development board. It is intended to help users quickly understand the hardware interfaces of the board and provides guidance for environment setup, source code compilation, firmware flashing, and functional testing of onboard hardware interfaces.

System Support

Development Board	Debian12	Buildroot
CM1126 V2 + Idea1126 V2	Y	Y

Revision History

Version	Date	Author	Revision History
V2.0	2026-07-07	Liu Yuan	Software updated to v1.2

Disclaimer

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

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

Boardcon embedded design limited

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

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

Email: market@armdesigner.com

Technical Support Inquiries: support@armdesigner.com

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

Content

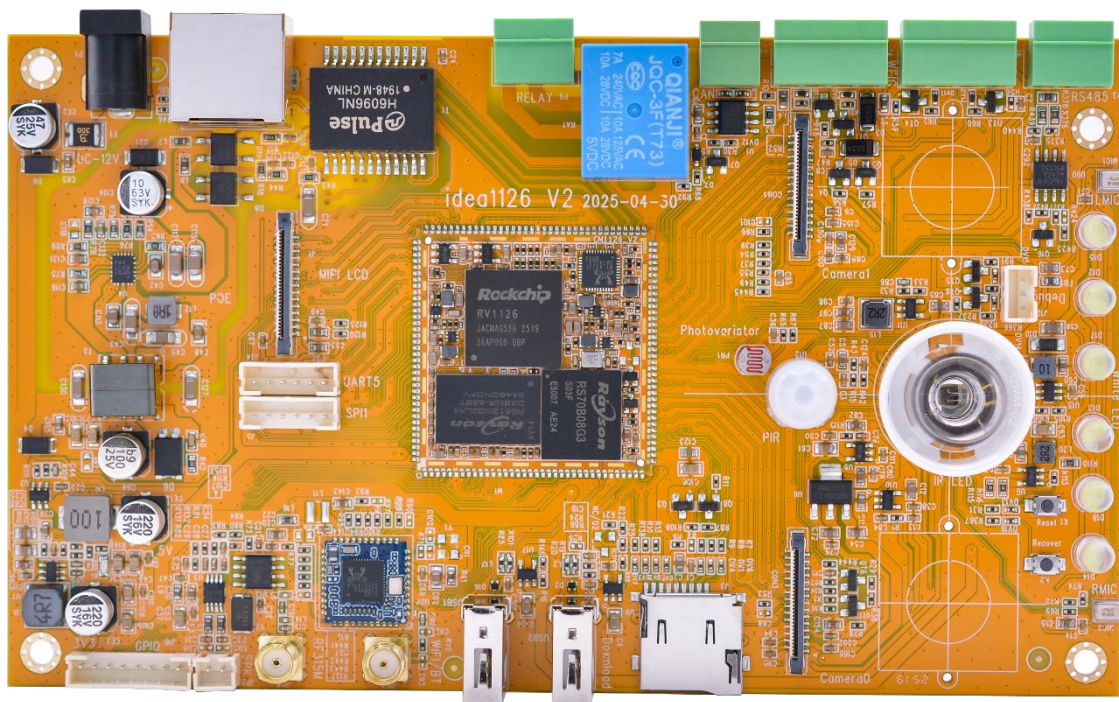
1. Introduction.....	4
1.2 Product Parameters	4
1.3 Hardware Interface Introduction.....	6
2. Install Drivers and Tools	8
2.1 Install RK Driver Assistant	9
2.2 Install CH9102X Driver.....	10
2.2.1 How to Connect the Serial Port Tool	10
2.2.2 Install Driver	10
2.3 Install Serial Terminal Tool.....	11
3. Upgrade Introduction.....	14
3.1 Upgrade Mode	14
3.1.1 How to Enter Loader Mode	15
3.1.2 How to Enter MaskRom Mode.....	17
3.2 Firmware Flashing	18
3.2.1 Flash Full Firmware Image.....	18
3.2.2 Flash Separate Partition Images.....	20
4. Development Environment.....	22
4.1 Preparing the Development Environment.....	22
4.2 Installing Libraries and Toolkits	22
5. Compile Source Code	23
6. Buildroot Test.....	26
6.1 Serial Terminal.....	26
6.2 Display.....	27
6.2.1 GStreamer Version	28
6.2.2 Rokit Version.....	28
6.3 USB.....	28
6.3.1 USB OTG	28

6.3.2 USB HOST	29
6.4 Audio.....	30
6.5 Ethernet.....	32
6.6 SD Card.....	33
6.7 WiFi & Bluetooth.....	34
6.7.1 WiFi	34
6.7.2 Bluetooth.....	36
6.8 IR LED.....	39
6.9 PIR	40
6.10 Photoresistor	41
6.11 Relay	41
6.12 RF315.....	42
6.13 Wiegand	43
6.14 RS485.....	45
6.15 UART.....	46
6.16 CAN.....	47
6.17 SPI.....	48
6.18 LED.....	49
6.19 Camera	50
6.19.1 GStreamer Version	50
6.19.2 Rokit Version.....	53
6.20 Video Playback	58
6.20.1 GStreamer Version	58
6.20.2 Rokit Version.....	59

1. Introduction

Idea1126B-P is a high-performance intelligent vision development platform based on the Rockchip RV1126B-P AI SoC. It integrates a quad-core Cortex-A53 processor and a 3TOPS NPU, making it suitable for edge AI computing, video analytics, and image recognition applications.

The board supports dual camera input, MIPI DSI display output, H.264/H.265 video codec, Gigabit Ethernet, Wi-Fi/BT, USB 2.0, audio, MicroSD storage, and multiple peripheral interfaces such as CAN, RS485, RS232, SPI, and Wiegand. It can be widely used in smart cameras, AI edge boxes, security surveillance, and industrial vision applications.



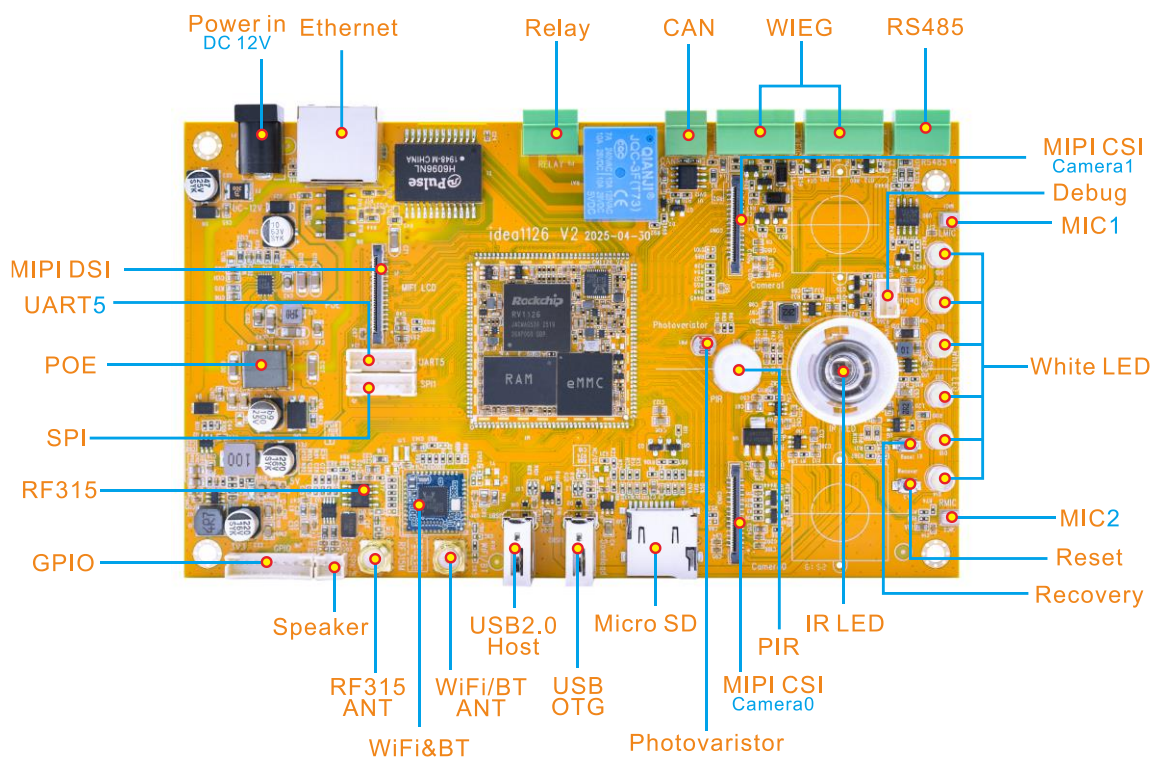
1.2 Product Parameters

Basic Parameters	
SOC	RV1126B-P

CPU		• Quad-core ARM Cortex-A53 • 32KB I-Cache / 32KB D-Cache per core • 512KB shared L2 Cache
NPU		• Up to 3 TOPS AI computing power • Supports INT4, INT8, INT16, FP16, BF16, TF32 • Optimized for CNN, Transformer, and other AI models
Video	Decoder	• H.265/H.264 up to 3840×2160@30fps • JPEG/MJPEG, VP8, VP9 decoding
	Encoder	• H.265/H.264 up to 3840×2160@30fps
AI-ISP		• Built-in 12MP AI Image Signal Processor • Supports multi-channel HDR, denoise, geometric correction
RAM		2GB LPDDR4
ROM		8GB eMMC
Support system		Debian, Buildroot
Hardware Parameters		
Extended Storage		• Support 1x MicroSD Card
Display		• Support 1x MIPI DSI
Audio		• Support 1x Speaker • Support 2x Digital MEMS MIC
USB		• Support 2x USB2.0
Network		• Support 1x Gigabit Ethernet • Support 1x WIFI/BT module
Camera		• Support 2x Camera
Peripheral communication		• Support 1xWiegand • Support 1xSPI • Support 1xRS485 • Support 1xRS232

	• Support 1xCAN
Other parameters	Support 1xDebug UART, 1xPIR, 1xIR LED ,6xLED, 1xPhotoresistor, 1xRF315
Electrical Parameters	
Power supply input voltage	12V/3A
Operating temperature	0~70°
Storage temperature	-40~85°
Structural Parameters	
Core board dimensions	35.0mm x 34.0mm
Motherboard dimensions	165.0mm x 95.0mm

1.3 Hardware Interface Introduction



Interface parameters	
Power in DC 12V	12V DC power input interface
Ethernet	Gigabit Ethernet RJ45 interface
Relay	Relay output interface for external device control
CAN	CAN communication interface
WIEG	Wiegand interface (input/output supported)
RS485	RS485 communication interface
MIPI CSI Camera1	MIPI CSI camera interface 1
Debug	Debug the serial port
MIC1	Microphone input channel 1
White LED	White LED
MIC2	Microphone input channel 2
Reset	Reset key
Recovery	Recovery key
IR LED	IR LED
PIR	Passive Infrared (PIR) motion sensor
MIPI CSI Camera0	MIPI CSI camera interface 0
Photovaristor	Light sensor
Mirco SD	MicroSD card slot
USB OTG	USB OTG interface (supports device/host switching)
USB2.0 Host	USB 2.0 host interface
WIFI&BT ANT	Wi-Fi/Bluetooth antenna interface
WIFI&BT	WIFI&Bluetooth module
RF315 ANT	RF 315MHz wireless antenna connector
RF315	RF 315MHz wireless module
Speaker	Stereo speaker output
GPIO	General-purpose input/output pins

SPI	SPI interface
POE	Power over Ethernet
UART5	UART5 serial communication interface
MIPI DIS	MIPI display interface

2. Install Drivers and Tools

To download firmware and debug in the terminal, the following drivers and software need

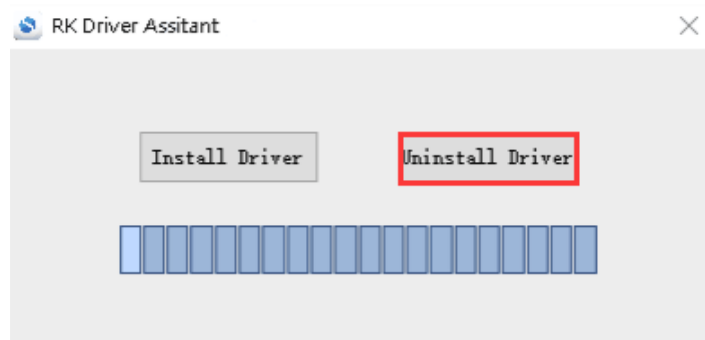
to be installed (for Windows computers):

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

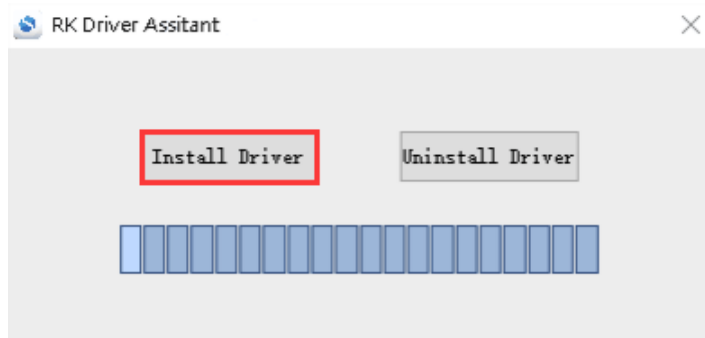
2.1 Install RK Driver Assistant

Step 1: Open *DriverAssistant/DriverInstall.exe*.

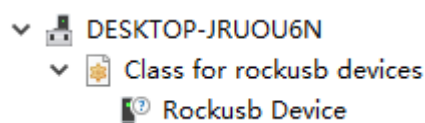
Step 2: To avoid driver conflicts, click “**Uninstall Driver**” to uninstall the driver.



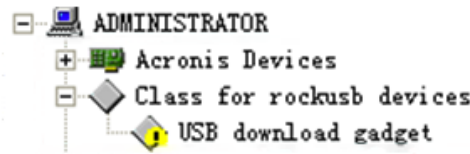
Step 3: Click button “**Install Driver**” to install.



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

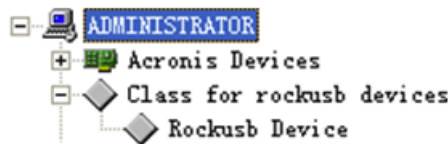


Step 5: If the following device information appears in the **Device Manager** after the operation in Step 4, user need to proceed to the next step.



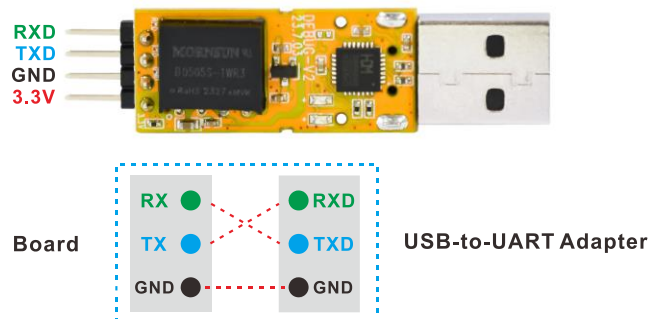
Step 6: The WINDOW will pop up found New Hardware Wizard dialog box, choose to install from the specified location, and then select: *DriverAssistant/ADBDriver*.

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



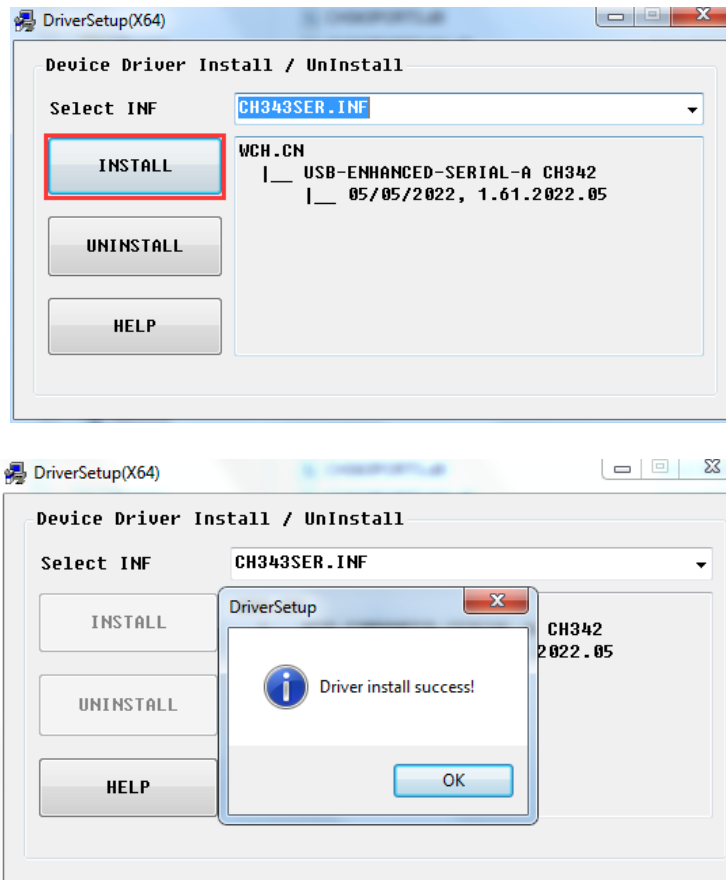
Pin	Connection Description
RXD	Receive, connect to TX pin of the board.
TXD	Transmit, connect to RX pin of the board.
GND	Ground, connect to GND pin of the board.
3V3	No need to connect.

2.2.2 Install Driver

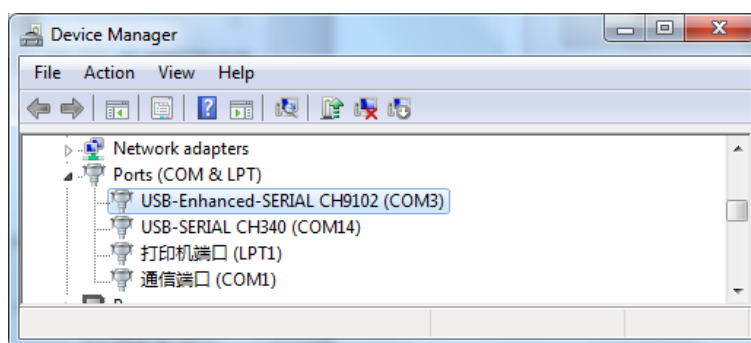
Step 1: Connect the CH9102X module to the PC.

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.

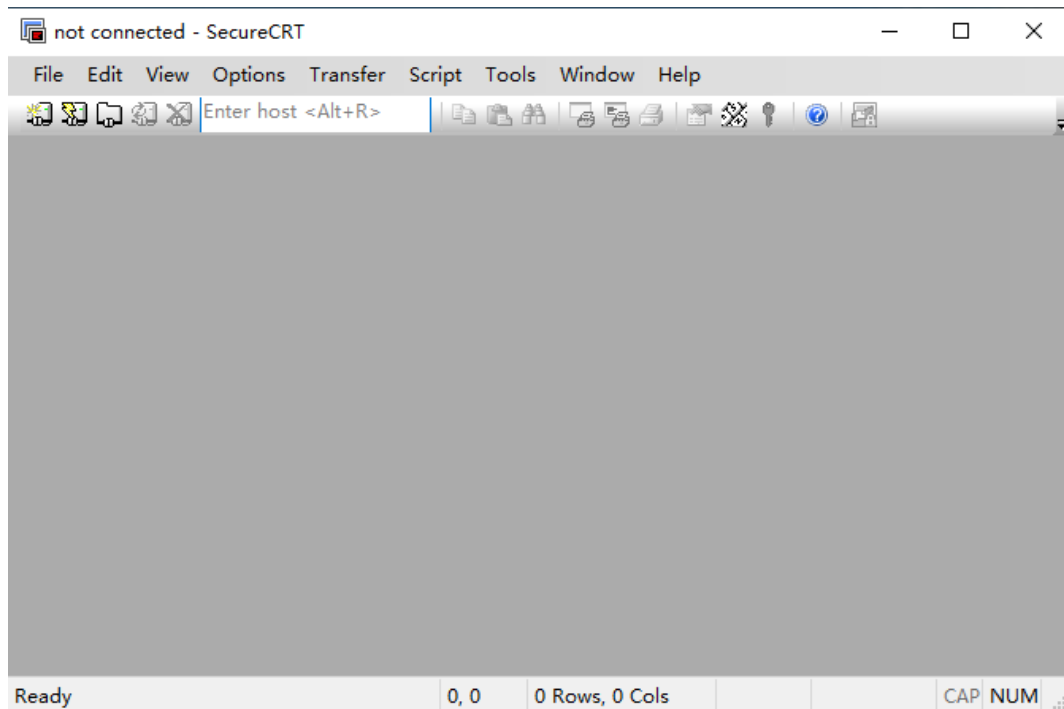


2.3 Install Serial Terminal Tool

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

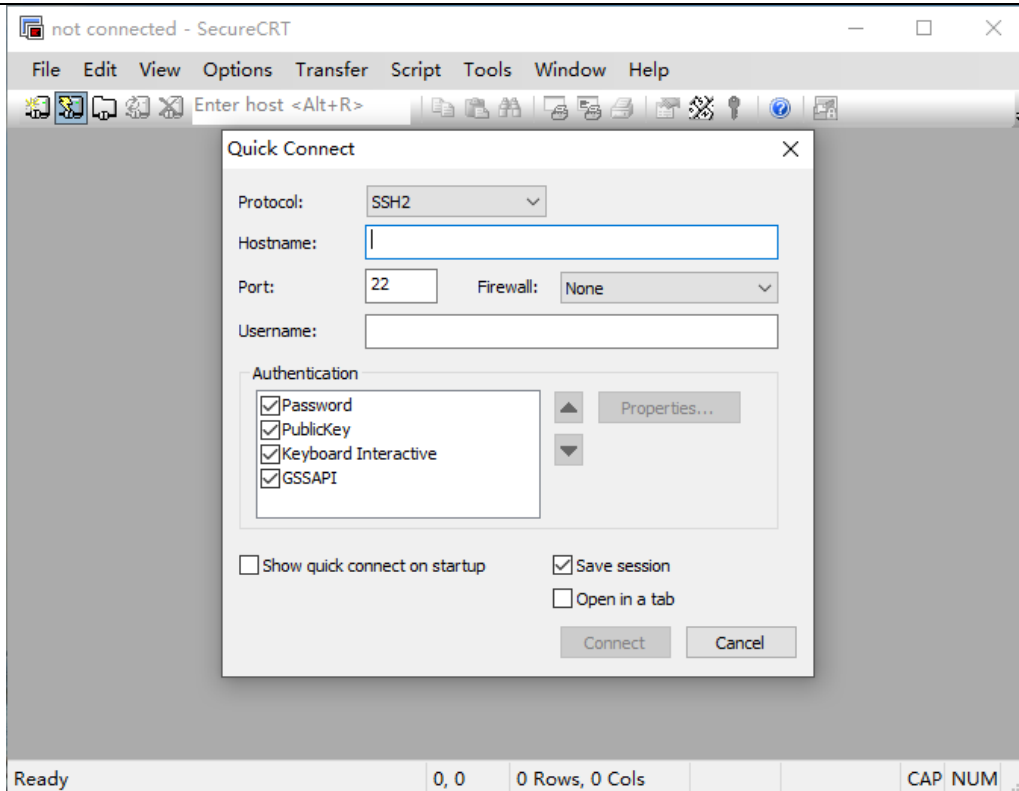
Step 1: Unzip *Platform/SecureCRT.rar* on PC.

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

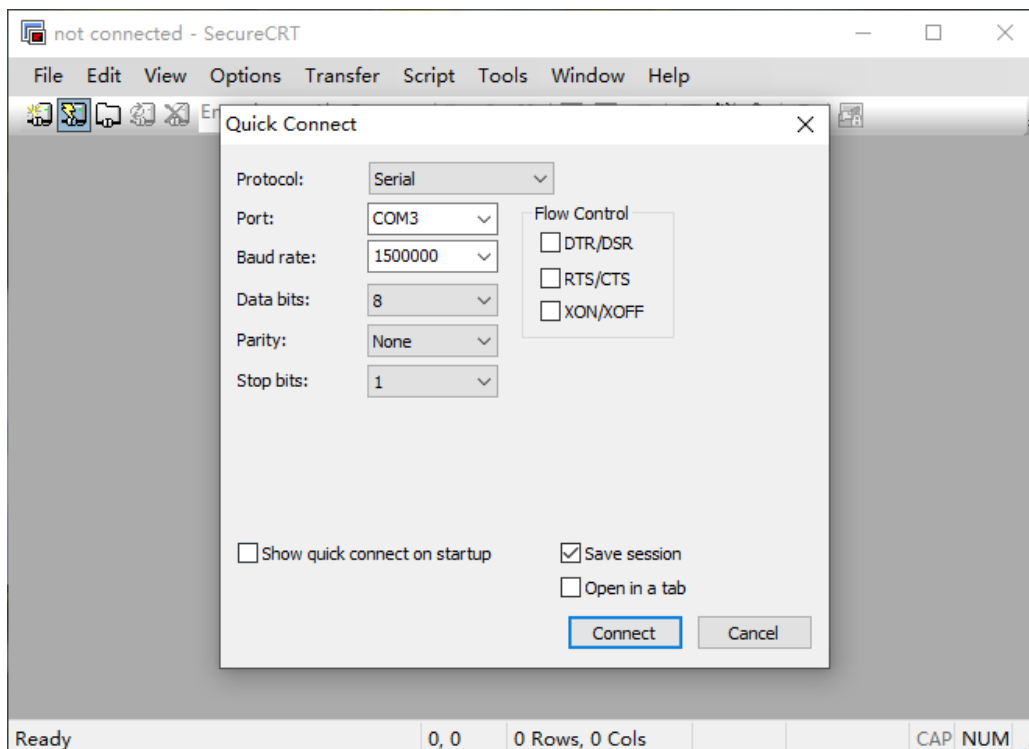


Step 3: Confirm that the CH9102X driver has been installed and the CH9102X module is connected to the PC.

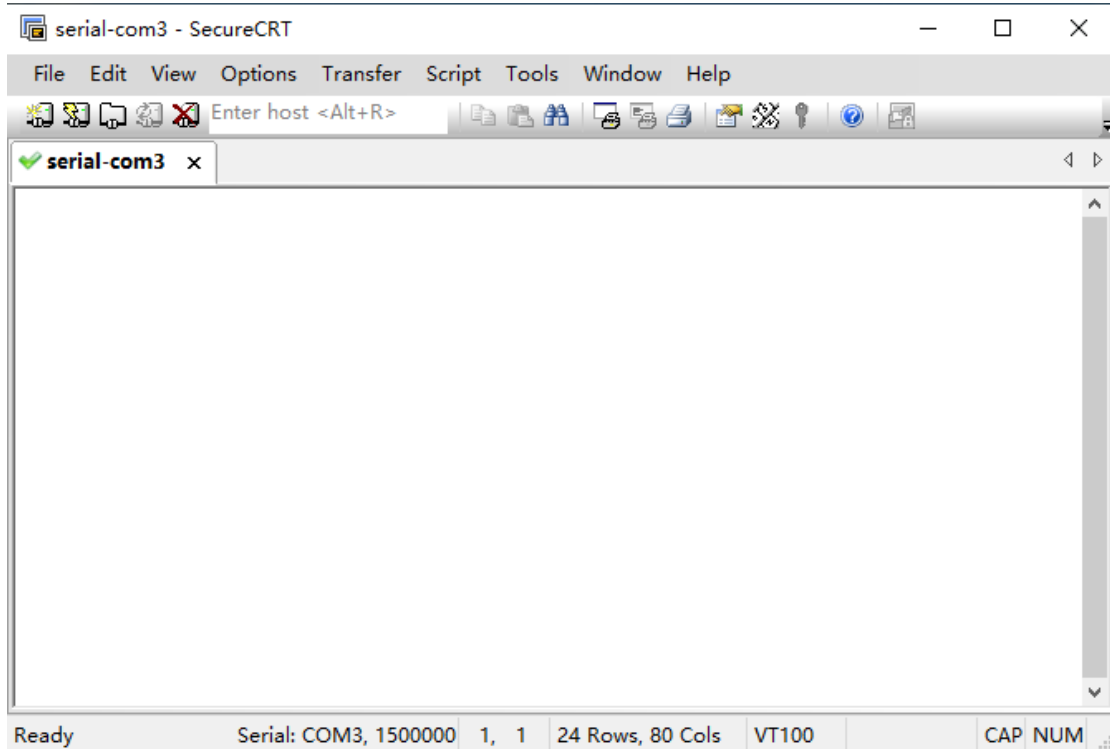
Step 4: Click the “**Quick Connect**” button to go to the Quick Connect configuration screen.



Step 5: Configure as shown in the following figure.



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



3. Upgrade Introduction

3.1 Upgrade Mode

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

1. Loader Mode:

Loader Mode is the standard mode used for firmware upgrade.

2. MaskRom Mode:

MaskRom Mode is a recovery mode used when the device cannot enter Loader Mode or fails to boot properly. Entering MaskRom Mode requires hardware operation and may involve certain risks. It is recommended to use this mode only when Loader Mode is unavailable.

Prerequisite:

Before upgrading the firmware via USB cable, make sure the required USB driver has been installed on the PC. For driver installation instructions, refer to the section [2.1 Install](#)

RK Driver Assistant.

3.1.1 How to Enter Loader Mode

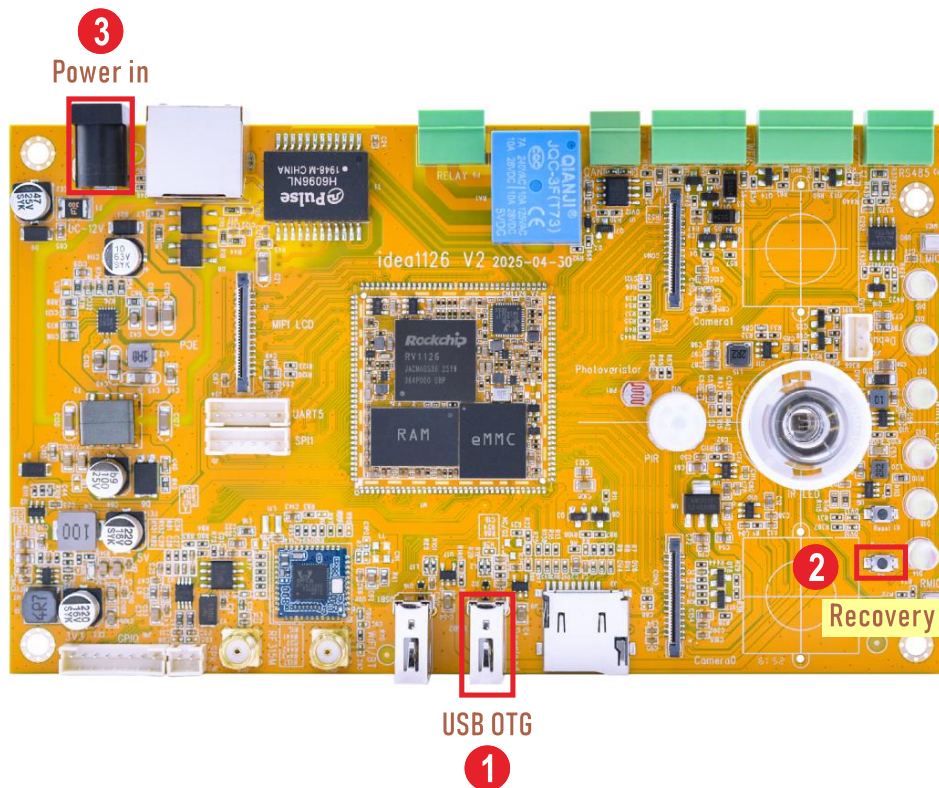
3.1.1.1 Hardware Method

Step 1: Disconnect the power adapter.

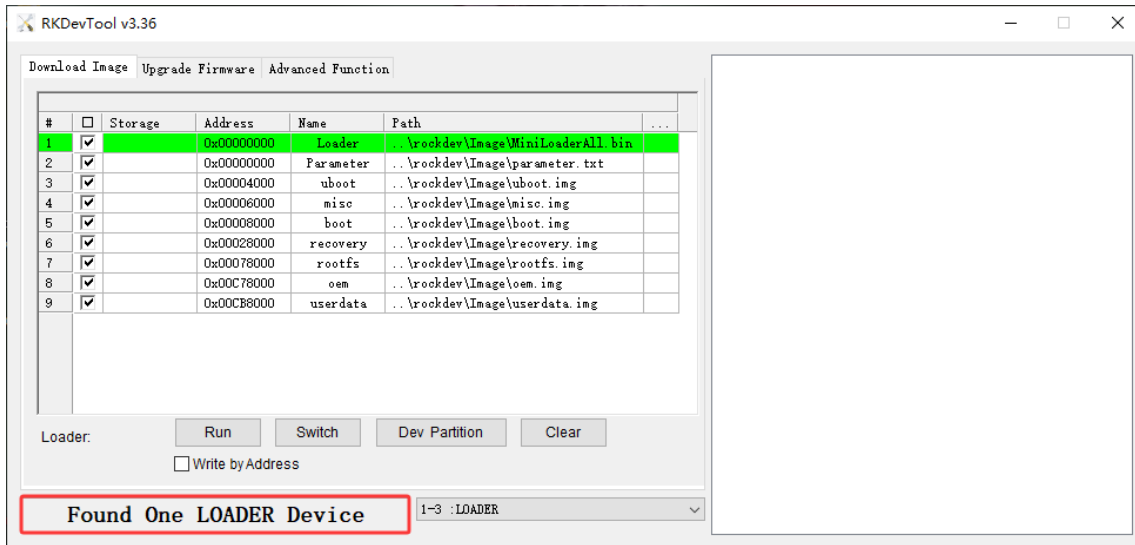
Step 2: Connect one end of the USB_A cable to the host PC and the other end to the development board.

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

Step 4: Connect the power supply.



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



3.1.1.2 Software Method

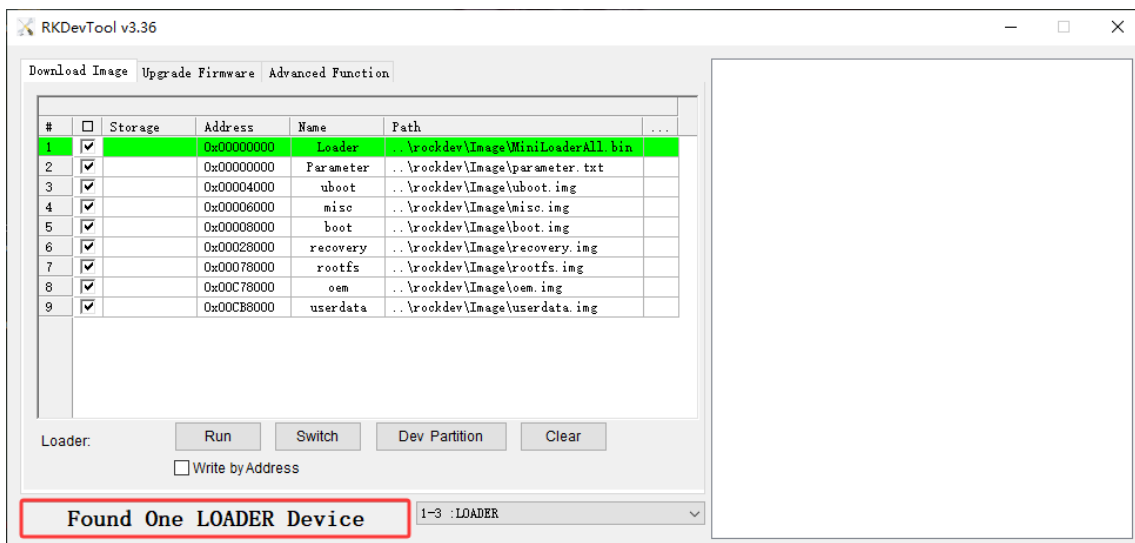
After the Micro USB cable is connected and the board is powered on, run the following command from the serial debug terminal or ADB shell:

```
# reboot loader
```

3.1.1.3 Serial Terminal Method

Step 1: Connect one end of the USB_A cable to the host PC and the other end to the development board, then open the serial debug terminal.

Step 2: Press and hold **Ctrl + D** in the serial terminal, then power on the board. Release **Ctrl + D** when the flashing tool shows “**Found one LOADER Device**”.



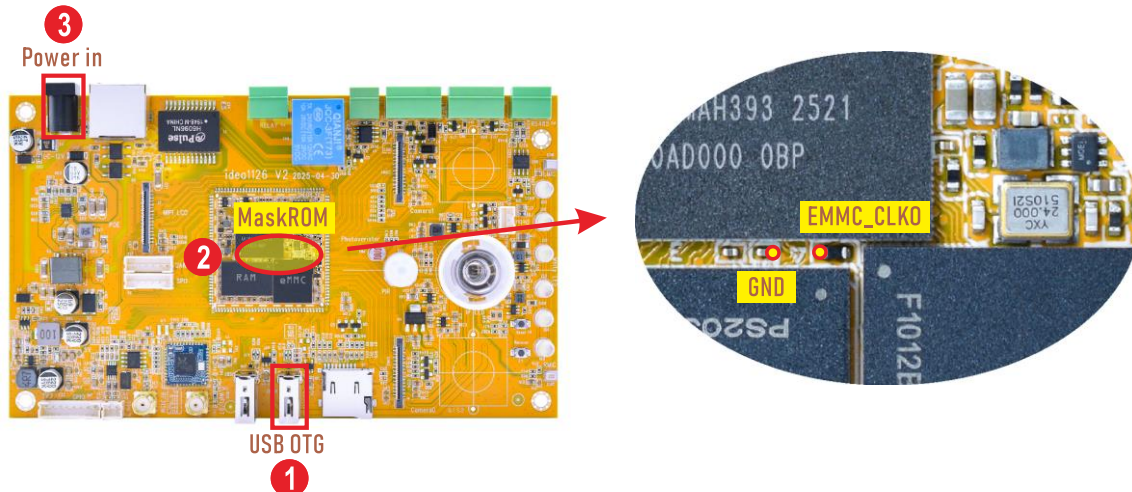
3.1.2 How to Enter MaskRom Mode

3.1.2.1 Hardware Method

Step 1: Disconnect the power adapter.

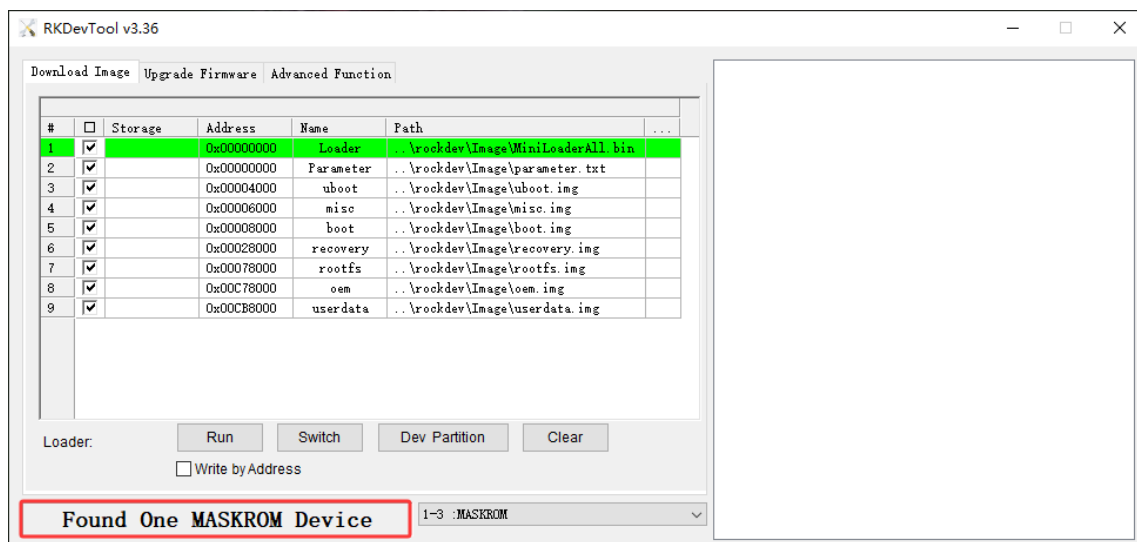
Step 2: Connect one end of the USB_A cable to the host PC and the other end to the development board.

Step 3: Use tweezers to short the two MaskRom test points on the CM1126.



Step 4: While keeping the test points shorted, connect the power supply.

Step 5: After the flashing tool shows “**Found one MASKROM Device**”, release the tweezers. The device has now entered MaskRom mode.

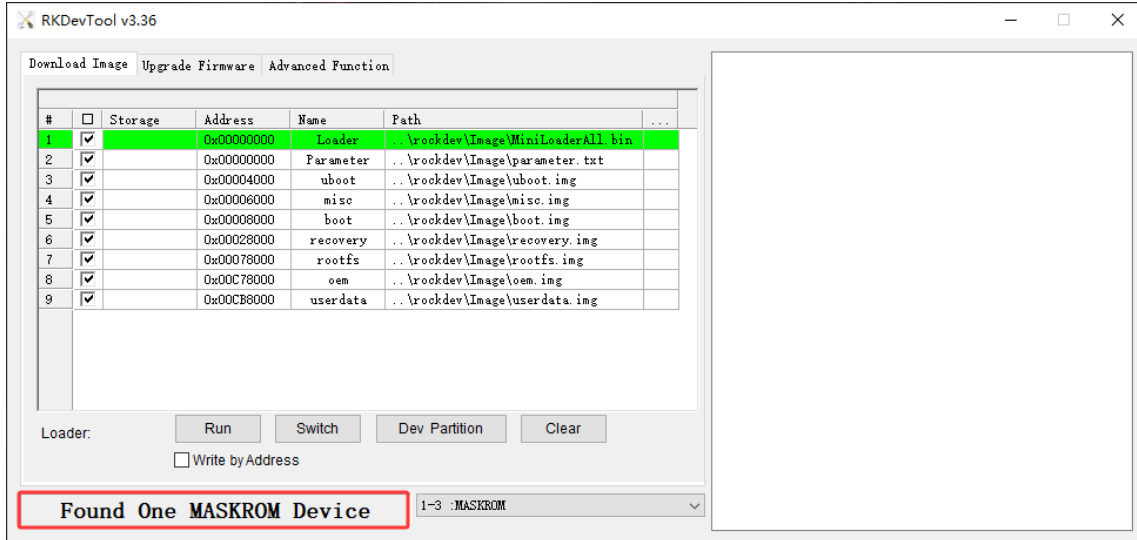


3.1.2.2 Serial Terminal Method

Step 1: Connect one end of the USB_A cable to the host PC and the other end to the

development board, then open the serial debug terminal.

Step 2: Press and hold **Ctrl + B** in the serial terminal, then power on the board. Release **Ctrl + B** when the flashing tool shows “Found one MASKROM Device”.

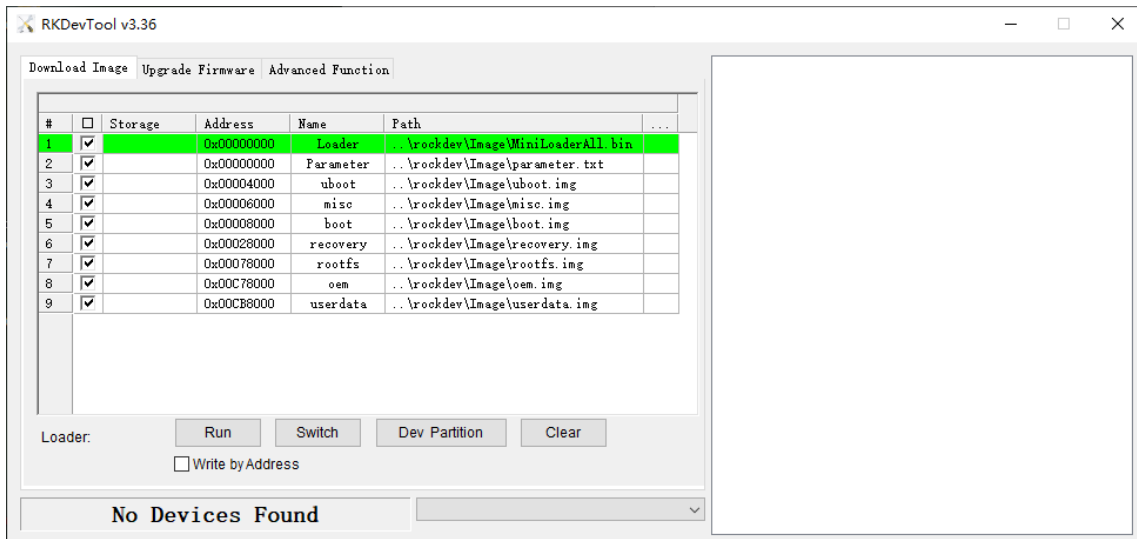


3.2 Firmware Flashing

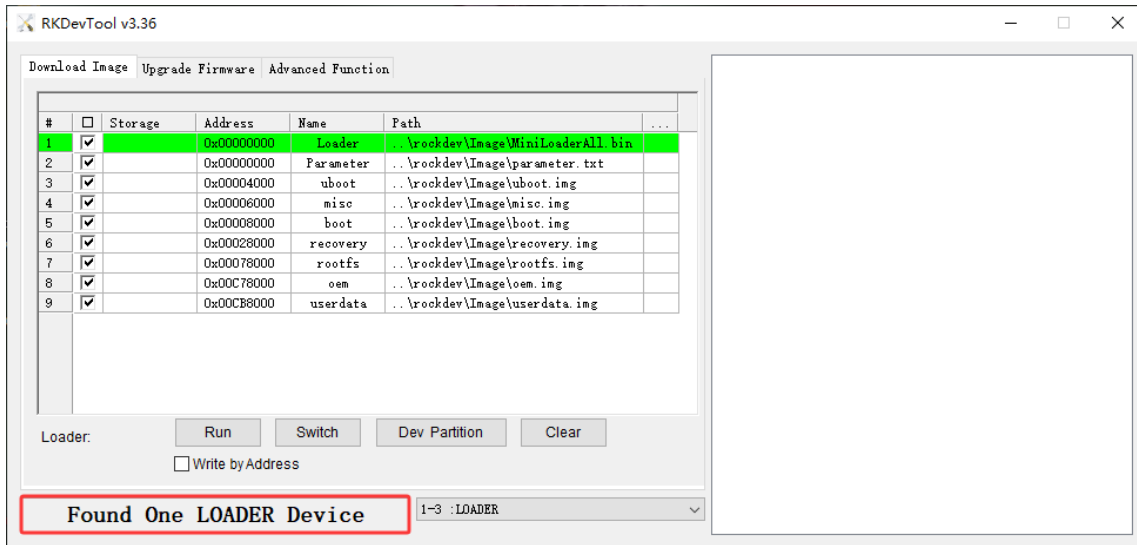
Environment: Windows OS

3.2.1 Flash Full Firmware Image

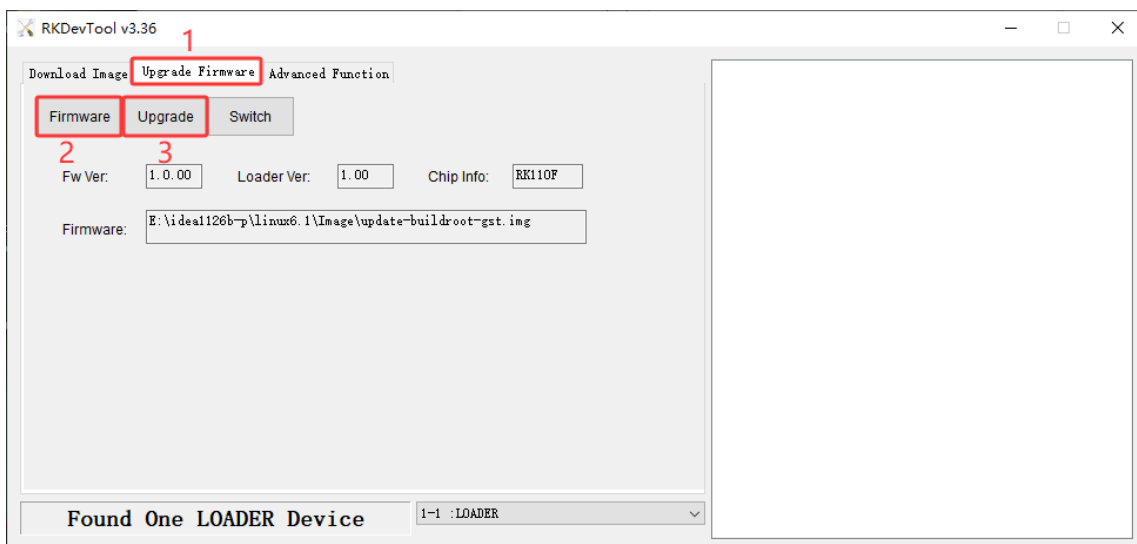
Step 1: Open *RKDevTool\RKDevTool_Release\RKDevTool.exe*.



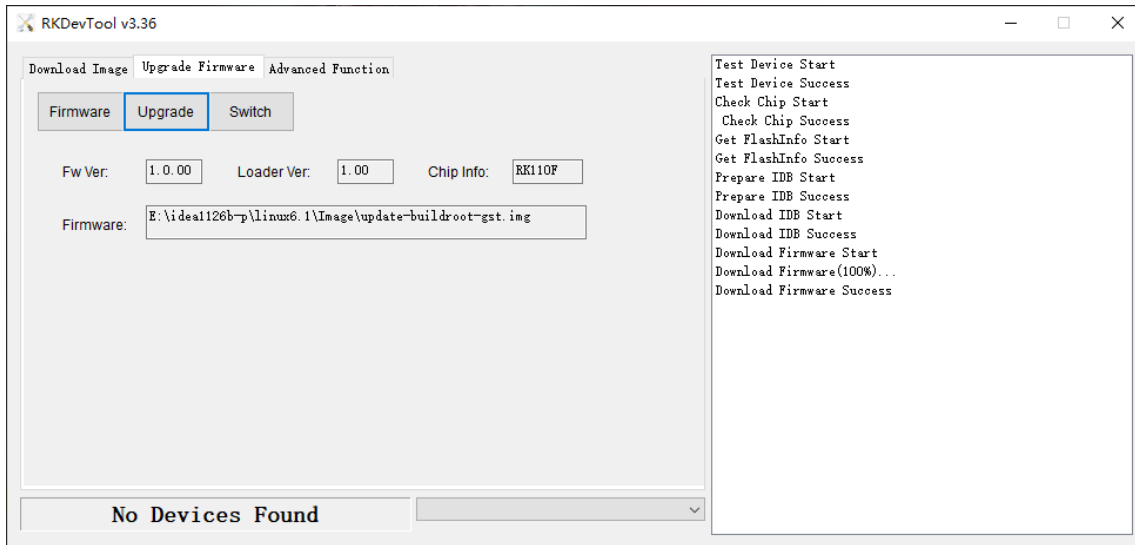
Step 2: Put the board into Loader mode. For details, refer to Section [3.1.1 How to Enter Loader Mode](#).



Step 3: Click **Upgrade Firmware** -> **Firmware**, select **update.img**, and then click **Upgrade** to start flashing.



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



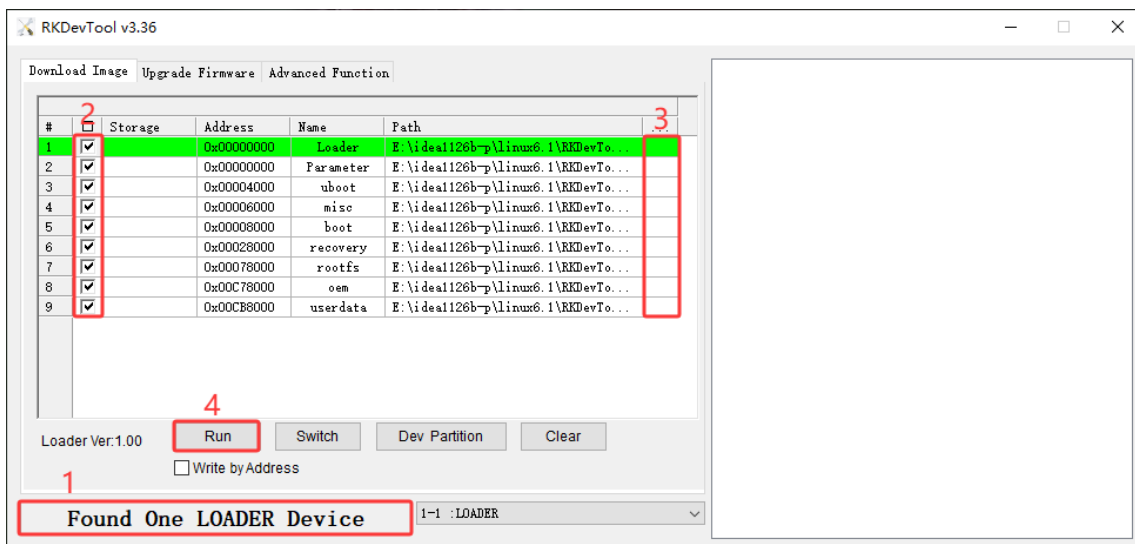
3.2.2 Flash Separate Partition Images

Step 1: Put the board into Loader mode. For details, refer to Section [3.1.1 How to Enter Loader Mode](#).

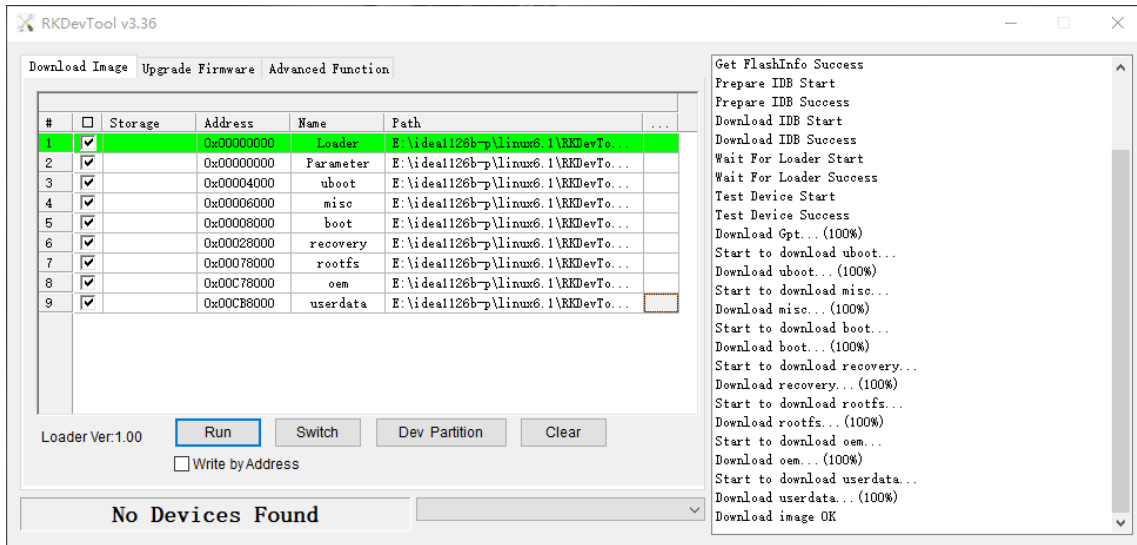
Step 2: Select the partitions to be flashed. Multiple partitions can be selected.

Step 3: Make sure the image file paths are correct. If necessary, click the blank cell next to each path to reselect the corresponding image file.

Step 4: Click **Run** to start flashing.



After flashing is complete, the board will automatically reboot.



4. Development Environment

4.1 Preparing the Development Environment

Ubuntu 22.04 or later is recommended for SDK compilation. If an error occurs during compilation, check the error message and install the required software packages accordingly. For other Linux distributions, the package names or installation commands may need to be adjusted.

In addition to the operating system requirements, the following hardware and software requirements should be met:

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

This section provides the commands for installing the software packages required to build the SDK compilation environment. Other tools, such as Samba and SSH, should be installed as needed.

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 Code

Step 1: Extract the Source Code

To extract the source code package, execute the following commands:

```
$ tar xvf RV1126B_LINUX6.1_V1.2.0.tar.gz
$ cd RV1126B_LINUX6.1_V1.2.0
```

Step 2: Configure the Target Board

To configure the target board, execute the following command:

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

After running **./build.sh lunch**, the system will list available defconfig files.

For the Buildroot system, select the corresponding configuration according to the required version:

- **GStreamer version:**

4. [boardcon_idea1126bp_defconfig](#)

- **Rockit version:**

5. [boardcon_idea1126bp_ipc_defconfig](#)

Note:

Only the Buildroot system provides both the GStreamer version and the Rockit version.

For the Debian system, use the GStreamer version.

The following steps use the GStreamer version as an example:

```
$ ./build.sh lunch
##### Rockchip Linux SDK #####
Manifest: rv1126b_linux6.1_release_v1.2.0_20251220.xml
GIT commit: "4236b4b boardcon: fix IPC defconfig to use em1126bp_ipc buildroot config"
Log colors: message notice warning error fatal
Log saved at /home/chenyu/RV1126bp/RV1126B_LINUX6.1_V1.2.0/output/sessions/2026-01-21_16-41-16
Pick a defconfig:
1. rockchip_defconfig
2. boardcon_em1126bp_defconfig
3. boardcon_em1126bp_ipc_defconfig
4. boardcon_idea1126bp_defconfig
5. boardcon_idea1126bp_ipc_defconfig
6. rockchip_rv1126b_dv_64_evb1_v10_defconfig
7. rockchip_rv1126b_dv_64_evb1_v12_defconfig
8. rockchip_rv1126b_evb1_v10_defconfig
9. rockchip_rv1126b_evb4_v10_defconfig
10. rockchip_rv1126b_fastboot_defconfig
11. rockchip_rv1126b_ipc_32_evb1_v10_defconfig
12. rockchip_rv1126b_ipc_64_evb1_v10_defconfig
13. rockchip_rv1126b_robot_defconfig
14. rockchip_rv1126b_tiny_32_evb2_v10_defconfig
15. rockchip_rv1126bp_evb1_v10_defconfig
16. rockchip_rv1126bp_fastboot_defconfig
17. rockchip_rv1126bp_ipc_32_evb1_v10_defconfig
18. rockchip_rv1126bp_ipc_64_evb1_v10_defconfig
19. rockchip_rv1126bp_robot_defconfig
Which would you like? [1]: 4
```

Step 3: Compile U-Boot

To compile U-Boot, execute the following command:

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

Step 4: Compile the Kernel

To compile the kernel, execute the following command:

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

Step 5: Compile Recovery

To compile recovery, execute the following command:

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

Step 6: Compile rootfs

To compile buildroot, execute the following command:

```
$ ./build.sh buildroot
```

Step 7: Generate and Package Firmware Images

To generate and package the firmware images, execute the following command:

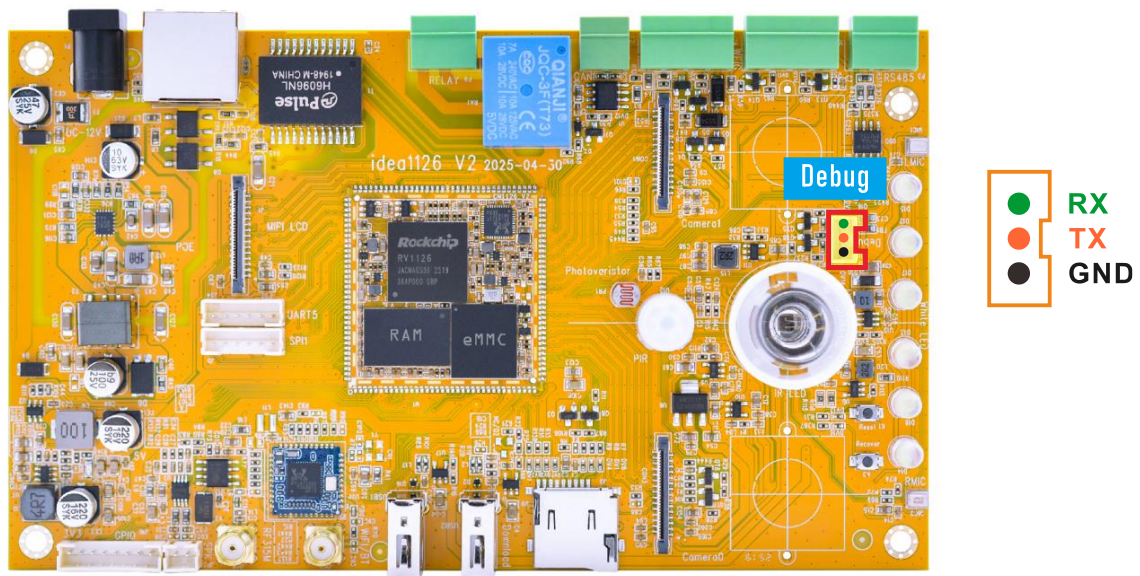
```
$ ./build.sh firmware
```

After the command is completed, the image files and **update.img** will be generated in the *rockdev/* directory.

6. Buildroot Test

6.1 Serial Terminal

Step 1: Connect the USB-to-serial module to the PC and connect the other end to the Debug UART interface on the board.



Step 2: Open a serial terminal tool on the PC, select the corresponding serial port, and set the baud rate to 1500000.

Step 3: Power on the board and check whether the boot log is displayed in the serial terminal.

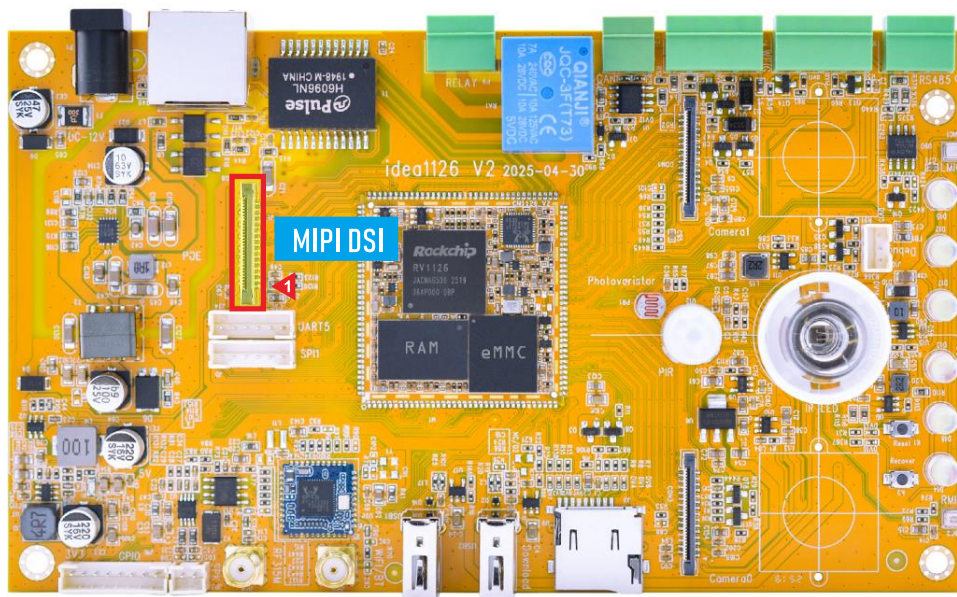
```
serial-com6 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com6 x
Realtek Bluetooth :Init Process finished
[ 10.216484] Bluetooth: h5_open
[ 10.216513] Bluetooth: hci_uart_register_dev
Realtek Bluetooth :Realtek Bluetooth post process
[ 10.217506] rtk_btcoex: open BTCOEX
Realtek Bluetooth :Device setup complete
[ 10.219659] rtk_btcoex: BTCOEX hci_rev 0xaa8
[ 10.219699] rtk_btcoex: BTCOEX lmp_subver 0x2df5
[ 10.246554] Bluetooth: MGMT ver 1.22
bluealsa: [876] D: bluez.c:1199: Signal: org.freedesktop.DBus.ObjectManager.InterfacesAdded()
HACK: Remove /dev/rfkill to disable external BT power operations.
[ 10.260310] rtk_btcoex: BTCOEX hci_rev 0xaa8
[ 10.260349] rtk_btcoex: BTCOEX lmp_subver 0x2df5
bluealsa: [876] D: bluez.c:783: Registering battery provider: /org/bluez/hci0/battery
Successfully init BT for RTL8723DS!
bluealsa: [876] D: bluez.c:598: Creating media endpoint object: /org/bluez/hci0/A2DP/SBC/sink/1
bluealsa: [876] D: bluez.c:509: Registering media endpoint: /org/bluez/hci0/A2DP/SBC/sink/1
bluealsa: [876] D: bluez.c:598: Creating media endpoint object: /org/bluez/hci0/A2DP/SBC/sink/2
bluealsa: [876] D: bluez.c:509: Registering media endpoint: /org/bluez/hci0/A2DP/SBC/sink/2
[ 10.321061] rtk_btcoex: rtk_handle_cmd_complete_evt: status 01, profile info cmd 0
[ 10.321118] Bluetooth: hci0: HCI set profile report cmd (-56)
[ 10.321129] rtk_btcoex: check_profileinfo_cmd: check profile info done
[ 16.873404] rockchip-cpuinfo cpuinfo: Soc : 11260b10
[ 16.873436] rockchip-cpuinfo cpuinfo: Serial : 075b198785832b01
[ 16.873799] platform mtd_vendor_storage: deferred probe pending

root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/#
```

For details about installing and using the serial terminal tool, please refer to [Section 2.2](#)
[Install CH9102X Driver](#) and [2.3 Install Serial Terminal Tool](#).

6.2 Display

The default MIPI display resolution of the Idea1126B-P is 800×1280@60 Hz.



The Buildroot system provides two versions: the GStreamer version and the Rockit

version. The display output may vary depending on the firmware version.

6.2.1 GStreamer Version

After the GStreamer version is started, the display output is shown in the following figure:



6.2.2 Rokit Version

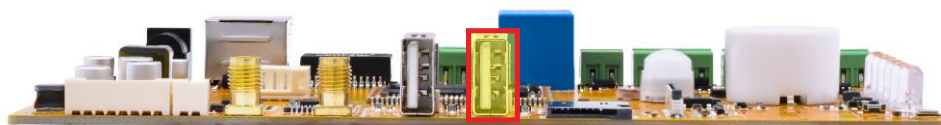
After the system boots with the Rokit version, if two cameras are connected before power-on, the system displays a combined preview of the two camera channels on the screen by default.

If the two cameras are not connected, the desktop UI will not be displayed after boot-up.

6.3 USB

6.3.1 USB OTG

The USB OTG port of Idea1126B-P is configured as host mode by default at startup.



USB OTG
28

- To switch the USB OTG port to USB device mode, execute the following commands:

```
# echo device > /sys/kernel/debug/usb/21500000.usb/mode
# usbdevice start
```

```
root@rv1126b-buildroot:/# echo device > /sys/kernel/debug/usb/21500000.usb/mode
root@rv1126b-buildroot:/# [ 372.623990] xhci-hcd xhci-hcd.0.auto: remove, state 4
[ 372.624037] usb usb2: USB disconnect, device number 1
[ 372.625200] xhci-hcd xhci-hcd.0.auto: USB bus 2 deregistered
[ 372.625588] xhci-hcd xhci-hcd.0.auto: remove, state 4
[ 372.625623] usb usb1: USB disconnect, device number 1
[ 372.627020] xhci-hcd xhci-hcd.0.auto: USB bus 1 deregistered

root@rv1126b-buildroot:/# usbdevice start
[1970-01-01 00:06:20] usbdevice [1358]: Sending start request to service...
[1970-01-01 00:06:20] usbdevice [1358]: Check /var/log/usb-gadget.log for usb-service log...
root@rv1126b-buildroot:/# [ 382.619079] file system registered
[ 382.845601] read descriptors
[ 382.845639] read strings
[ 383.024014] dwc3 21500000.usb: device reset
[ 383.111685] android_work: sent uevent USB_STATE=CONNECTED
[ 383.142080] android_work: sent uevent USB_STATE=CONFIGURED
```

- To switch the USB OTG port back to USB Host mode, execute the following command:

```
# echo host > /sys/kernel/debug/usb/21500000.usb/mode
```

```
root@rv1126b-buildroot:/# echo host > /sys/kernel/debug/usb/21500000.usb/mode
root@rv1126b-buildroot:/# [ 452.635893] dwc3 21500000.usb: request 00000000e209ac6a was not queued
to ep0out
[ 452.636082] android_work: sent uevent USB_STATE=DISCONNECTED
[ 452.745008] phy phy-21400000.usb2-phy.2: illegal mode
[ 452.745057] xhci-hcd xhci-hcd.0.auto: xHCI Host Controller
[ 452.745342] xhci-hcd xhci-hcd.0.auto: new USB bus registered, assigned bus number 1
[ 452.745508] xhci-hcd xhci-hcd.0.auto: hcc params 0x0220fe64 hci version 0x110 quirks
0x00000008022010010
[ 452.745560] xhci-hcd xhci-hcd.0.auto: irq 82, io mem 0x21500000
[ 452.745702] xhci-hcd xhci-hcd.0.auto: xHCI Host Controller
[ 452.745935] xhci-hcd xhci-hcd.0.auto: new USB bus registered, assigned bus number 2
[ 452.745968] xhci-hcd xhci-hcd.0.auto: Host supports USB 3.0 SuperSpeed
[ 452.746147] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice= 6.01
[ 452.746162] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 452.746171] usb usb1: Product: xHCI Host Controller
[ 452.746178] usb usb1: Manufacturer: Linux 6.1.141 xhci-hcd
[ 452.746185] usb usb1: SerialNumber: xhci-hcd.0.auto
[ 452.746765] hub 1-0:1.0: USB hub found
[ 452.746822] hub 1-0:1.0: 1 port detected
[ 452.747239] usb usb2: We don't know the algorithms for LPM for this host, disabling LPM.
[ 452.747369] usb usb2: New USB device found, idVendor=1d6b, idProduct=0003, bcdDevice= 6.01
[ 452.747383] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 452.747392] usb usb2: Product: xHCI Host Controller
[ 452.747399] usb usb2: Manufacturer: Linux 6.1.141 xhci-hcd
[ 452.747407] usb usb2: SerialNumber: xhci-hcd.0.auto
[ 452.747871] hub 2-0:1.0: USB hub found
[ 452.747914] hub 2-0:1.0: 1 port detected
[ 452.800532] read descriptors
[ 452.800582] read strings
```

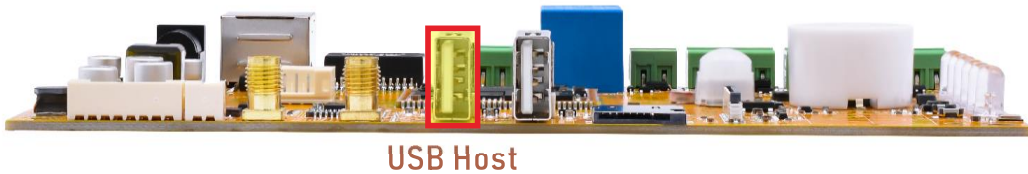
6.3.2 USB HOST

The Idea1126B-P provides one USB 2.0 Host interface and one USB 2.0 OTG interface.

The USB Host interface can be used to connect USB peripherals, such as a USB mouse, USB keyboard, USB flash drive, and other USB devices.

Note:

The USB 2.0 OTG interface supports OTG mode. To use it as a Host interface, switch it to Host mode first. For details, refer to Section [6.3.1 USB OTG](#).



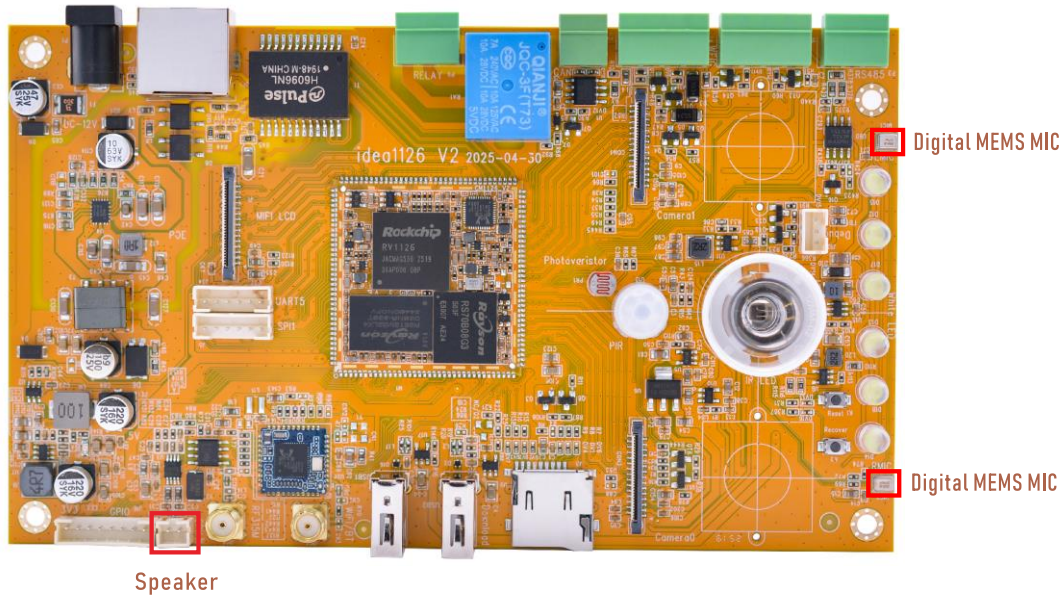
After a USB flash drive is connected, it will be mounted automatically. Run the following command to check the mount path of the USB flash drive:

```
# df -h
```

```
root@rv1126b-buildroot:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        5.9G  450M   5.3G   8% /
devtmpfs         959M    0   959M   0% /dev
tmpfs            987M  528K   987M   1% /run
tmpfs            987M  216K   987M   1% /var/log
tmpfs            987M    0   987M   0% /dev/shm
tmpfs            987M  140K   987M   1% /tmp
/dev/mmcblk0p7   124M   5.3M   118M   5% /oem
/dev/mmcblk0p8   881M  324K   880M   1% /userdata
/dev/sda1        28G   20G   8.7G  69% /mnt/udisk
/dev/sda2        448M  327M  121M  74% /media/udisk1
```

6.4 Audio

Step 1: Connect the speaker.



Step 2: Check the sound card information.

Execute the following command:

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

```
root@rv1126b-buildroot:/# cat /proc/asound/cards
0 [rockchiprv1126b]: rockchip_rv1126 - rockchip,rv1126b-acodec
rockchip,rv1126b-acodec
1 [rockchipppdmica]: simple-card - rockchip,pdm-mic-array
rockchip,pdm-mic-array
```

Step 3: Record audio through the Digital MEMS MIC.

Note:

For the **Rockit** version, the audio recording test may conflict with the rkipc service. Before performing the audio input test, stop rkipc first.

Execute the following command:

```
# killall -9 rkipc
```

To start rkipc again after the test, execute the following command:

```
# rkipc &
```

Execute the following command:

```
# arecord -D hw:1,0 -f s16_le -c 2 -r 16000 cap.wav
```

```
root@rv1126b-buildroot:/# arecord -D hw:1,0 -f s16_le -c 2 -r 16000 cap.wav
Recording WAVE 'cap.wav' : Signed 16 bit Little Endian, Rate 16000 Hz, Stereo
```

Step 4: Play audio through the Speaker.

Configure the speaker output path and volume:

```
# amixer -c 0 cset name='Power Amplifier' 1
# amixer -c 0 cset name='Speaker Switch' on
# amixer -c 0 cset name='DAC Digital Volume' 250,250
```

```
root@rv1126b-buildroot:/# amixer -c 0 cset name='Power Amplifier' 1
numid=40,iface=MIXER,name='Power Amplifier'
; type=ENUMERATED,access=rw-----,values=1,items=2
; Item #0 'Off'
; Item #1 'On'
: values=1
root@rv1126b-buildroot:/# amixer -c 0 cset name='Speaker Switch' on
numid=44,iface=MIXER,name='Speaker Switch'
; type=BOOLEAN,access=rw-----,values=1
: values=on
root@rv1126b-buildroot:/# amixer -c 0 cset name='DAC Digital Volume' 250,250
numid=37,iface=MIXER,name='DAC Digital Volume'
; type=INTEGER,access=rw---R--,values=2,min=0,max=510,step=0
: values=250,250
| dBscale-min=-956.25dB,step=3.75dB,mute=0
```

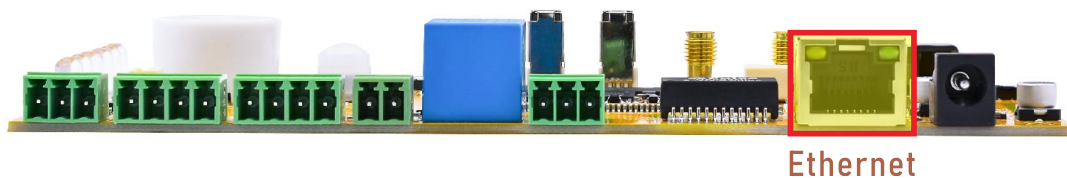
Then play the recorded audio file:

```
# aplay -Dhw:0,0 cap.wav
```

```
root@rv1126b-buildroot:/# aplay -Dhw:0,0 cap.wav
Playing WAVE 'cap.wav' : Signed 16 bit Little Endian, Rate 16000 Hz, Stereo
```

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.

```
root@rv1126b-buildroot:/# [ 2678.825363] rk_gmac-dwmac 21c70000.ethernet eth0: Link is Up - 1Gbps/Full -
flow control rx/tx
[ 2678.825419] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

Step 2: Check the network interface information.

Execute the following command:

```
# ifconfig eth0
```

```
root@rv1126b-buildroot:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 7A:B1:AC:2D:53:6F
          inet addr:192.168.0.102  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::3ffa:7401:7324:d8e2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:175 errors:0 dropped:23 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15472 (15.1 KiB)  TX bytes:1410 (1.3 KiB)
          Interrupt:80
```

Step 3: Test the network connection.

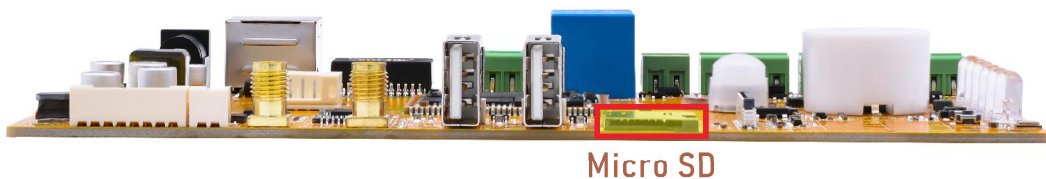
Execute the following command:

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

```
root@rv1126b-buildroot:/# ping -I eth0 www.armdesigner.com
PING www.armdesigner.com (3.174.46.103) from 192.168.0.102 eth0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=1 ttl=246 time=186 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=2 ttl=246 time=193 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=3 ttl=246 time=187 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=4 ttl=246 time=185 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=5 ttl=246 time=187 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=6 ttl=246 time=193 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=7 ttl=246 time=187 ms
^C
--- www.armdesigner.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 185.361/188.288/193.056/2.981 ms
```

6.6 SD Card

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



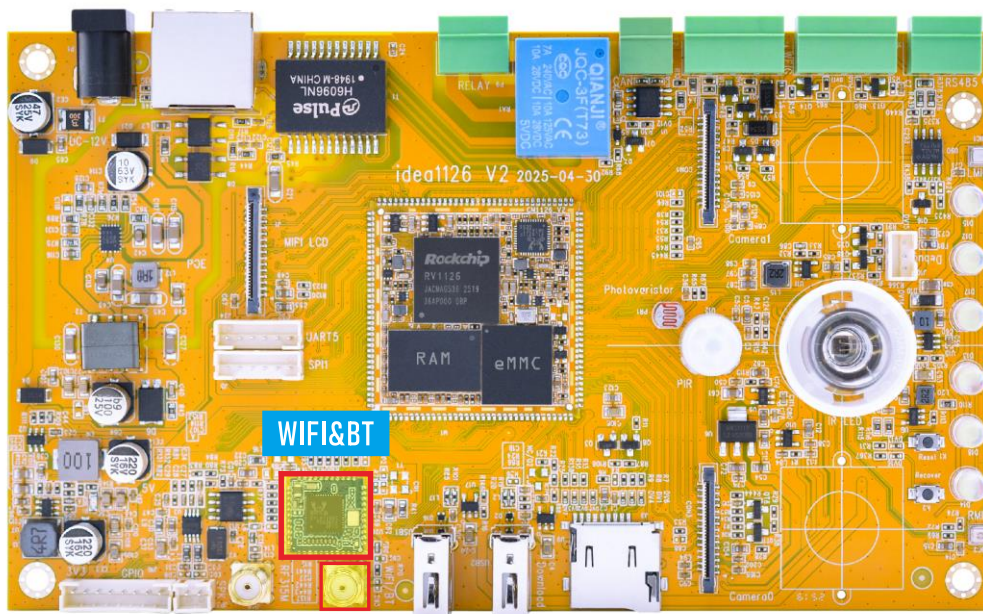
After the SD card is recognized, it will be mounted automatically. Run the following command to check the mount path of the SD card.

```
# df -h
```

```
root@rv1126b-buildroot:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        5.9G  450M  5.3G   8% /
devtmpfs        959M    0  959M   0% /dev
tmpfs           987M  532K  987M   1% /run
tmpfs           987M  196K  987M   1% /var/log
tmpfs           987M    0  987M   0% /dev/shm
tmpfs           987M  140K  987M   1% /tmp
/dev/mmcblk0p8   881M  396K  864M   1% /userdata
/dev/mmcblk0p7  124M   5.3M  115M   5% /oem
/dev/sda1        58G   38G   21G  65% /mnt/udisk
/dev/mmcblk1p1   60G   13G   48G  21% /mnt/sdcard
```

6.7 WiFi & Bluetooth

To use the WiFi and Bluetooth functions properly, make sure the antenna is connected.



Antenna

6.7.1 WiFi

Step 1: Check the Wi-Fi device information.

Execute the following command:

```
# ifconfig wlan0
```

```
root@rv1126b-buildroot:/# ifconfig wlan0
wlan0      Link encap:Ethernet HWaddr A8:B5:8E:CA:27:50
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Step 2: Scan for available WiFi hotspots.

• GStreamer Version

Execute the following command:

```
# iwlist wlan0 scan
```

```
root@rv1126b-buildroot:/# iwlist wlan0 scan
wlan0    Scan completed :
  Cell 01 - Address: C8:3A:35:51:1E:98
            ESSID:"Tenda_511E98"
            Protocol:IEEE 802.11bgn
            Mode:Master
            Frequency:2.412 GHz (Channel 1)
            Encryption key:on
            Bit Rates:150 Mb/s
            Extra:wpa_ie=dd180050f20101000050f20401000050f20401000050f2020c00
            IE: WPA Version 1
                Group Cipher : CCMP
                Pairwise Ciphers (1) : CCMP
                Authentication Suites (1) : PSK
            Quality=96/100  Signal level=30/100
            Extra:fm=0003
  Cell 02 - Address: B4:F1:8C:6D:D1:24
            ESSID:"Boardcon"
            Protocol:IEEE 802.11bgn
            Mode:Master
            Frequency:2.462 GHz (Channel 11)
            Encryption key:on
            Bit Rates:300 Mb/s
            Extra:wpa_ie=dd1a0050f20101000050f20202000050f2040050f20201000050f202
```

• Rockit Version

Execute the following command:

```
# wpa_cli -i wlan0 scan
# wpa_cli -i wlan0 scan_results
```

```
root@rv1126b-buildroot:/# wpa_cli -i wlan0 scan
OK
root@rv1126b-buildroot:/# wpa_cli -i wlan0 scan_results
ssid / frequency / signal level / flags / ssid
b4:f1:8c:6d:d1:24      2462    -50    [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]    Boardcon
b4:f1:8c:6d:d1:29      2462    -50    [WPA2-PSK-CCMP][WPS][ESS]
b4:f1:8c:fd:d1:29      2462    -58    [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]    Boardcon_Wi-
Fi5
d8:32:14:25:b7:a8      2437    -69    [WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]    LYB-2.4G
b2:22:7a:5a:b6:4a      2462    -51    [WPA2-PSK-CCMP][ESS]    DIRECT-4A-HP Laser 136w
44:6a:2e:20:83:60      2412    -67    [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]    ydsz
9c:a6:15:10:e5:57      2437    -74    [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]    TP
b4:f1:8c:6d:d1:25      2462    -58    [WPS][ESS]
38:20:28:50:1e:a1      2437    -68    [WPA2-PSK-CCMP][ESS][UTF-8]    Casvi
50:0f:f5:a7:bb:e0      2417    -78    [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]    1960
38:20:28:50:0f:81      2412    -70    [WPA2-PSK-CCMP][ESS][UTF-8]    Casvi
d8:fe:42:ce:ee:d3      2432    -81    [WPA2-PSK-CCMP][WPS][ESS][P2P]    DIRECT-zb-HUAWEI CV81-WDMa
9c:a6:15:10:da:c5      2437    -71    [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]    TP
9c:a6:15:10:dd:6f      2412    -81    [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]    TP
8c:ee:fd:1a:a4:fb      2437    -77    [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]    ChinaNet-AQ74
d8:fe:42:ce:f2:58      2437    -75    [WPA2-PSK-CCMP][WPS][ESS]    DIRECT-A1-HUAWEI CV81-WDMa
```

Step 3: Connect to the WiFi hotspot.

Execute the following command:

```
# wifi-connect.sh SSID PSK
```



```
root@rv1126b-buildroot:/# wifi-connect.sh Boardcon Boardcon43435656
connect to WiFi ssid: Boardcon, Passwd: Boardcon43435656
Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
[ 148.528961] NOHZ tick-stop error: local softirq work is pending, handler #08!!!
[ 148.533191] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
[ 148.796912] NOHZ tick-stop error: local softirq work is pending, handler #08!!!
[ 148.832911] NOHZ tick-stop error: local softirq work is pending, handler #08!!!
[ 148.853020] NOHZ tick-stop error: local softirq work is pending, handler #08!!!
Obtaining IP address...
WiFi connected successfully!
IP address: 192.168.0.160
DNS server: 192.168.0.1
```

Step 4: Check the network interface status.

Execute the following command:

```
# ifconfig wlan0
```

```
root@rv1126b-buildroot:/# ifconfig wlan0
wlan0      Link encap:Ethernet HWaddr 84:FC:14:8A:A6:45
           inet addr:192.168.0.160 Bcast:192.168.0.255 Mask:255.255.255.0
           inet6 addr: fe80::6f7b:f84c:8a62:ec99/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
           RX packets:272 errors:0 dropped:38 overruns:0 frame:0
           TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:35070 (34.2 KiB) TX bytes:4072 (3.9 KiB)
```

Step 5: Test the WiFi network connection.

Execute the following command:

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

```
root@rv1126b-buildroot:/# ping -I wlan0 www.armdesigner.com
PING d3dsoj86mhwkk6.cloudfront.net (3.174.46.103) from 192.168.0.160 wlan0: 56(84) bytes of data.
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=1 ttl=246 time=219 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=2 ttl=246 time=1278 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=3 ttl=246 time=265 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=4 ttl=246 time=343 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=5 ttl=246 time=215 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=6 ttl=246 time=229 ms
64 bytes from www.armdesigner.com (3.174.46.103): icmp_seq=7 ttl=246 time=204 ms
^C
--- d3dsoj86mhwkk6.cloudfront.net ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6027ms
rtt min/avg/max/mdev = 203.982/393.118/1277.518/363.724 ms, pipe 2
```

6.7.2 Bluetooth

Bluetooth is configured to work as a Bluetooth speaker by default.

Step 1: Start the BlueALSA A2DP sink service.

Execute the following command:

```
# bluealsa -p a2dp-sink &
```

```
root@rv1126b-buildroot:/# bluealsa -p a2dp-sink &
[1] 1341
root@rv1126b-buildroot:/# bluealsa: [1341] D: storage.c:84: Initializing persistent storage:
/var/lib/bluealsa
bluealsa: [1341] D: main.c:604: Starting main dispatching loop
bluealsa: [1341] E: main.c:133: Couldn't acquire D-Bus name. Please check D-Bus configuration. Requested
name: org.bluealsa
```

Step 2: Enter the Bluetooth control tool.

Execute the following command:

```
# bluetoothctl
```

```
root@rv1126b-buildroot:/# bluetoothctl
hci0 new_settings: powered bondable ssp br/edr le secure-conn
Agent registered
[CHG] Controller 84:FC:14:8A:A6:46 Pairable: yes
[bluetooth]#
```

Step 3: Enable Bluetooth and set it to discoverable mode.

Execute the following commands in bluetoothctl:

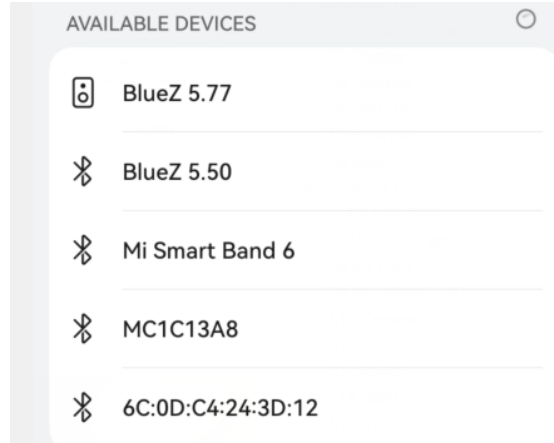
```
[bluetooth]# power on
[bluetooth]# discoverable on
```

```
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# discoverable on
[bluetooth]# [ 410.688521] Bluetooth: hu 00000000b31dfb1 retransmitting 1 pkts
hci0 new_settings: powered connectable bondable ssp br/edr le secure-conn
hci0 new_settings: powered connectable discoverable bondable ssp br/edr le secure-conn
Changing discoverable on succeeded
[CHG] Controller 84:FC:14:8A:A6:46 Discoverable: yes
[bluetooth]#
```

After this step, the Bluetooth device can be discovered by the phone.

Step 4: Pair and connect the Bluetooth device.

On the phone, find the Bluetooth device named “**BlueZ 5.77**” in the Bluetooth device list, then tap it to pair and connect.



If a pairing confirmation message appears, confirm it on both the phone and the board.

```
[bluetooth]# [ 432.300087] rtk_btcoex: hci accept conn req
[ 432.493911] rtk_btcoex: connected, handle 0002, status 0x00
[ 432.493952] rtk_btcoex: Page success
[ 432.548448] rtk_btcoex: io capability request
hci0 A8:35:12:9A:EB:4D type BR/EDR connected eir_len 11
[bluetooth]# bluealsa: [854] D: bluez.c:1199: Signal: org.freedesktop.DBus.ObjectManager.InterfacesAdded()
[NEW] Device A8:35:12:9A:EB:4D liuy
Request confirmation
[agent] Confirm passkey 585345 (yes/no): yes
[liuy]# [ 439.265025] rtk_btcoex: link key notify
hci0 new_link_key A8:35:12:9A:EB:4D type 0x05 pin_len 0 store_hint 1
hci0 device_flags_changed: A8:35:12:9A:EB:4D (BR/EDR)
supp: 0x00000000 curr: 0x00000000
[CHG] Device A8:35:12:9A:EB:4D INFO: 0x0007 (7)
[CHG] Device A8:35:12:9A:EB:4D Bonded: yes
[CHG] Device A8:35:12:9A:EB:4D UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[ 440.083762] rtk_btcoex: PSM(0x0001) do not need parse
[liuy]# [ 440.100325] rtk_btcoex: l2cap op 3, len 20, out 0
[ 440.100393] rtk_btcoex: RX l2cap conn rsp, hndl 0x0002, dcid 0x0052, scid 0x0041, result 0x0000
bluealsa: [854] D: bluez.c:1199: Signal: org.freedesktop.DBus.ObjectManager.InterfacesAdded()
[CHG] Device A8:35:12:9A:EB:4D Modalias: bluetooth:v010Fp107Ed1436
[CHG] Device A8:35:12:9A:EB:4D ServicesResolved: yes
[CHG] Device A8:35:12:9A:EB:4D INFO: 0x000f (15)
[CHG] Device A8:35:12:9A:EB:4D Paired: yes
Authorize service
[agent] Authorize service 00001108-0000-1000-8000-00805f9b34fb (yes/no): [ 440.996577] rtk_btcoex:
count_pan_packet_timeout: pan_packet_count 10
[agent] Authorize service 00001108-0000-1000-8000-00805f9b34fb (yes/no): yes
[liuy]# [ 448.562980] rtk_btcoex: l2cap op 2, len 16, out 0
Authorize service
[agent] Authorize service 0000110d-0000-1000-8000-00805f9b34fb (yes/no): [ 448.764668] rtk_btcoex: TX
l2cap conn rsp, hndl 0x0002, dcid 0x0041, scid 0x005f, result 0x0001
W: [pulseaudio] module-loopback.c: Configured latency of 200.00 ms is smaller than minimum latency, using
minimum instead
W: [pulseaudio] module-loopback.c: Cannot set requested sink latency of 66.67 ms, adjusting to 250.00 ms
W: [pulseaudio] module-loopback.c: Cannot set requested source latency of 16.00 ms, adjusting to 39.91 ms
W: [pulseaudio] module-loopback.c: Configured latency of 200.00 ms is smaller than minimum latency, using
minimum instead
W: [pulseaudio] module-loopback.c: Cannot set requested sink latency of 35.20 ms, adjusting to 39.91 ms
W: [pulseaudio] module-loopback.c: Cannot set requested source latency of 66.67 ms, adjusting to 250.00 ms
[agent] Authorize service 0000110d-0000-1000-8000-00805f9b34fb (yes/no): yes
[liuy]#
```

Check the log output to confirm the MAC address. In this example, the MAC address of the mobile phone is **A8:35:12:9A:EB:4D**.

Step 5: Exit bluetoothctl.

Execute the following command:


```
[bluetooth]# exit
```

Step 6: Start Bluetooth audio playback.

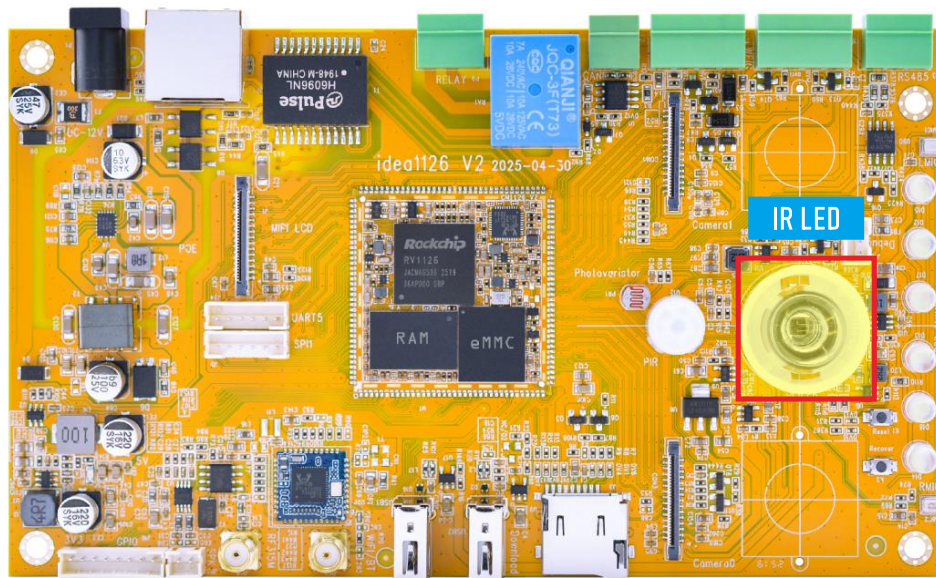
Execute the following command:

```
# bluealsa-aplay A8:35:12:9A:EB:4D &
```

```
[liuy]# exit
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/# bluealsa-aplay A8:35:12:9A:EB:4D &
[1] 1360
root@rv1126b-buildroot:/# bluealsa-aplay: [1360] D: aplay.c:1194: Starting main loop
```

After the Bluetooth connection is established, play audio on the phone. The audio should be output from the board.

6.8 IR LED



Step 1: Turn on the IR LED.

Execute the following command:

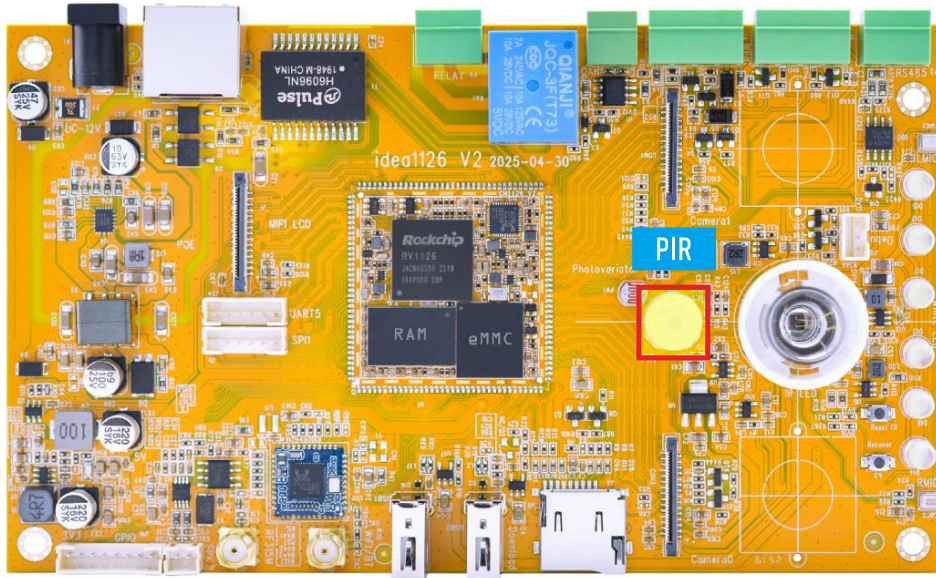
```
# echo 1 > /sys/class/leds/ir_led/brightness
```

Step 2: Turn off the IR LED.

Execute the following command:

```
# echo 0 > /sys/class/leds/ir_led/brightness
```

6.9 PIR



The PIR sensor is used to detect human motion by sensing changes in infrared radiation.

Execute the following command to test the PIR sensor:

```
# pir_test.sh
```

```
The output value indicates the detection status:  
1: Motion detected  
0: No motion detected
```

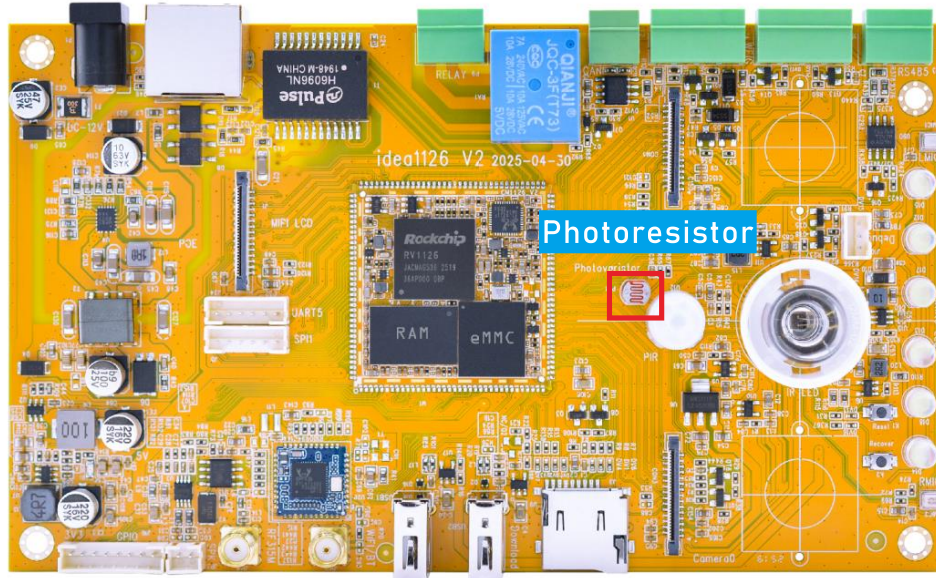
Example output:

```
root@rv1126b-buildroot:/# pir_test.sh  
1  
1  
1  
1  
1  
1  
1  
1  
1  
0  
0  
1  
0  
0
```

When human motion is detected, the output is 1. When no motion is detected, the output is 0.

6.10 Photoresistor

The photoresistor is used to detect ambient light intensity.



Execute the following command to read the raw ADC value:

```
# cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
```

```
root@rv1126b-buildroot:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
2318
root@rv1126b-buildroot:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
2323
root@rv1126b-buildroot:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
1017
root@rv1126b-buildroot:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
636
root@rv1126b-buildroot:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
438
```

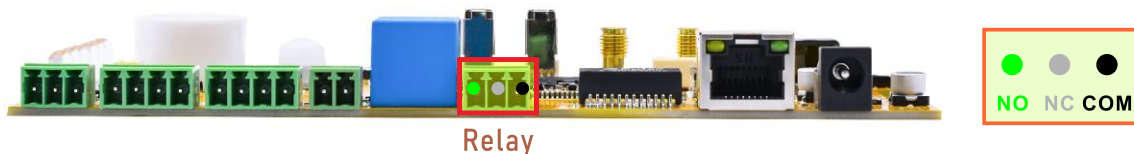
The output value changes with the ambient light intensity.

6.11 Relay

The relay has three terminals: **NO**, **NC**, and **COM**.

By default, the relay is energized after power-on, and **NO** is connected to **COM**.

To release the relay, execute the following command:




```
# echo 1 > /sys/class/leds/raley_power/brightness
```

```
root@rv1126b-buildroot:/# echo 1 > /sys/class/leds/raley_power/brightness
root@rv1126b-buildroot:/#
```

After execution, the relay is released, and NC is connected to **COM**.

To energize the relay again, execute the following command:

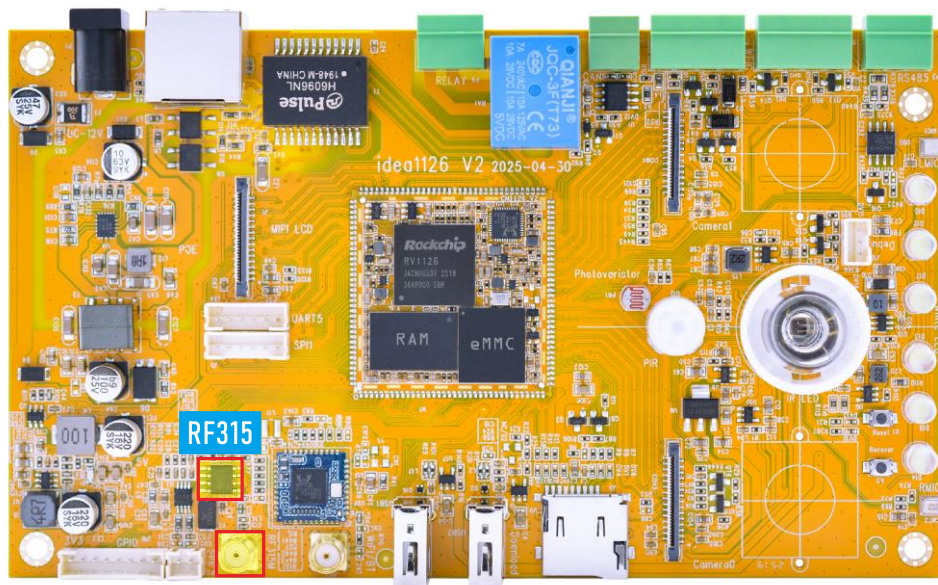
```
# echo 0 > /sys/class/leds/raley_power/brightness
```

```
root@rv1126b-buildroot:/# echo 0 > /sys/class/leds/raley_power/brightness
root@rv1126b-buildroot:/#
```

After execution, **NO** is connected to **COM** again.

6.12 RF315

Step 1: Connect the RF315 antenna.



Step 2: Enable RF315 debug output:

```
# echo 1 > /sys/module/rf315_decode/parameters/code_print
```

Step 3: Press a key on the RF315 remote control.



If the signal is received correctly, the received key value will be printed in the serial console or system log.

```
troot@rv1126b-buildroot:/# echo 1 > /sys/module/rf315_decode/parameters/code_print
root@rv1126b-buildroot:/# [ 8909.453775] RF315 data=0xadf488
[ 8909.490630] RF315 data=0xadf488
[ 8911.531192] RF315 data=0xadf484
[ 8911.568093] RF315 data=0xadf484
[ 8913.331031] RF315 data=0xadf482
[ 8913.367931] RF315 data=0xadf482
[ 8914.712698] RF315 data=0xadf481
[ 8914.749613] RF315 data=0xadf481
[ 8916.782097] RF315 data=0xadf484
[ 8917.765258] RF315 data=0xadf482
[ 8917.802131] RF315 data=0xadf482
[ 8923.601151] RF315 data=0xadf488
```

6.13 Wiegand

The Wiegand interface is commonly used for access control devices, such as card readers and keypads.



The board provides two Wiegand interfaces: **Wieg_OUT** and **Wieg_IN**. A loopback test can be performed by connecting **Wieg_OUT** to **Wieg_IN**.

Step 1: Connect the Wiegand pins as follows:

Wieg_OUT	Wieg_IN
Data0	Data0
Data1	Data1

Step 2: Open a terminal through the serial port, run the Wiegand output test program:

```
# wg_out_test
```

```
root@rv1126b-buildroot:/# wg_out_test
hid is 0xa26f
pid is 0x8b5c
####Setup Wiegand output 26bit or 34bit.####
[1] Wiegand output 26bit
[2] Wiegand output 34bit
[0] quit
-----Select-----
Enter your Selection:
```

Step 3: Open another terminal through ADB and run the Wiegand input test program:

```
# wg_input_test
```

```
C:\Users\15405>
C:\Users\15405>adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
root@rv1126b-buildroot:/#
root@rv1126b-buildroot:/# wg_input_test
Listening for Wiegand data on /dev/input/by-path/platform-rockchip-wg-input-event...
(Press Ctrl+C to exit)
```

Step 4: Check whether the data received by wg_input_test matches the data sent by wg_out_test.

<pre>root@rv1126b-buildroot:/# root@rv1126b-buildroot:/# wg_out_test hid is 0xa26f pid is 0x8b5c ####Setup Wiegand output 26bit or 34bit.#### [1] Wiegand output 26bit [2] Wiegand output 34bit [0] quit -----Select----- Enter your Selection: 1 [16500.358813] wg_output_26bit: HID:6f PID:8b5c ####Setup Wiegand output 26bit or 34bit.#### [1] Wiegand output 26bit [2] Wiegand output 34bit [0] quit -----Select----- Enter your Selection: 2 [16502.850816] wg_output_34bit: HID:a26f PID:8b5c ####Setup Wiegand output 26bit or 34bit.#### [1] Wiegand output 26bit [2] Wiegand output 34bit [0] quit -----Select----- Enter your Selection: 1 [16506.554836] wg_output_26bit: HID:6f PID:8b5c ####Setup Wiegand output 26bit or 34bit.#### [1] Wiegand output 26bit [2] Wiegand output 34bit [0] quit -----Select----- Enter your Selection: 1</pre>	<pre>C:\Users\15405> C:\Users\15405>adb shell * daemon not running. starting it now on port 5037 * * daemon started successfully * root@rv1126b-buildroot:/# root@rv1126b-buildroot:/# wg_input_test Listening for Wiegand data on /dev/input/by-path/platform-rockchip-wg-input-event... (Press Ctrl+C to exit) 26-bit: HID:6f PID:8b5c Raw=0x2df16b8 34-bit: HID:a26f PID:8b5c Raw=0x344d16b9 26-bit: HID:6f PID:8b5c Raw=0x2df16b8</pre>
--	---

Parameter Description:

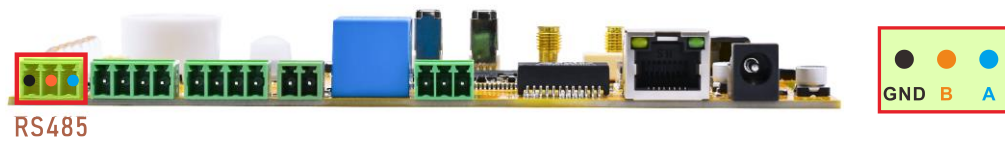
The following uses the 26-bit Wiegand data HID: 6f PID: 8b5c Raw=0x2df16b8 as an example to describe the meaning of each parameter.

```

26-bit: HID:6f  PID:8b5c  Raw=0x2df16b8

(1)Bit Grouping Structure:
1 | 01101111 | 1000101101011100 | 0
^          Facility      Card Number      ^
(2)Breakdown:
P1 (Leading parity bit): 1
Facility Code/HID: 01101111 = 0x6F
Card Number/PID: 1000101101011100 = 0x8B5C
P2 (Trailing parity bit): 0
  
```

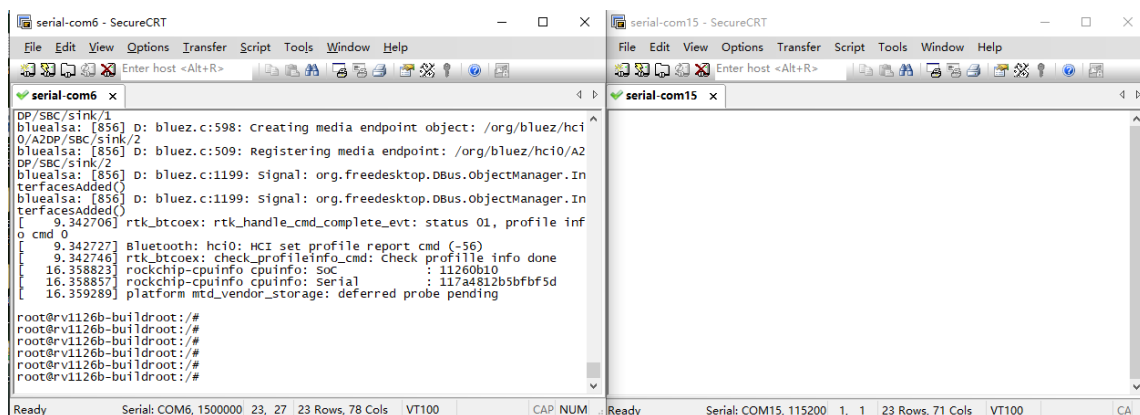
6.14 RS485



Step 1: Connect the RS485 test tool to the development board as shown in the diagram.

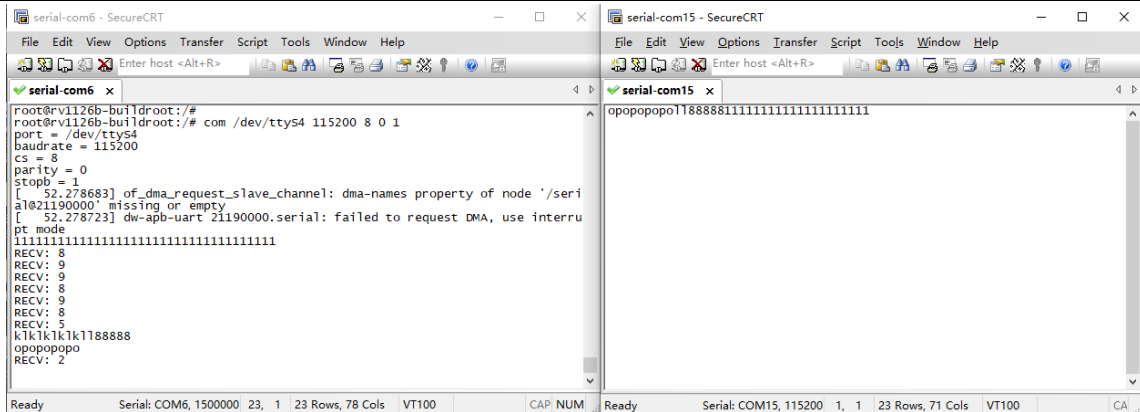


Step 2: Open two serial terminal windows. Set the debug serial port baud rate to 1500000, and set the RS485 serial port baud rate to 115200.



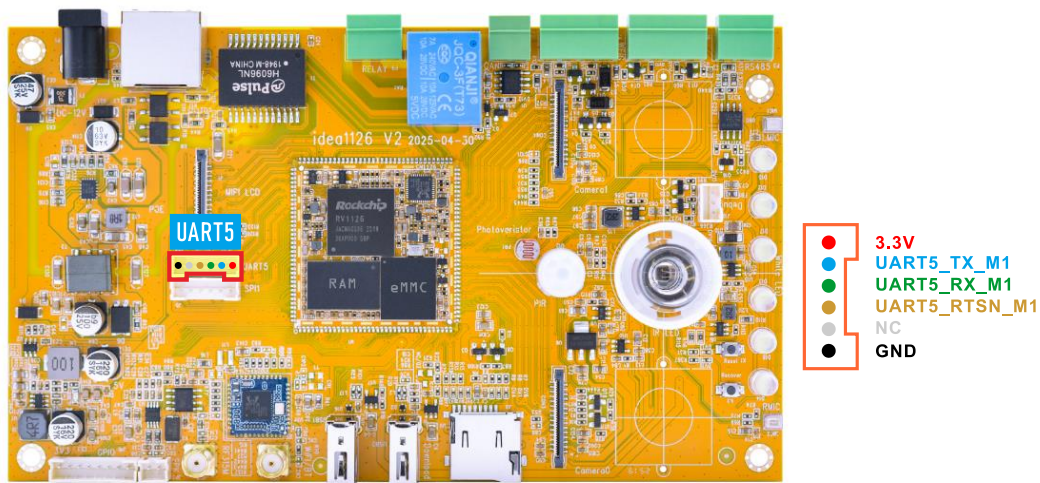
Step 3: Execute the following command on the board to test the RS485 data transmission and reception.

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



6.15 UART

Step 1: Connect the RX and TX pins of the UART interface together to create a loopback connection.



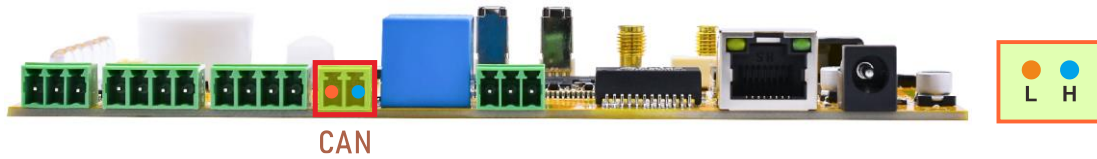
Step 2: Test UART loopback communication.

Execute the following command:

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

```
root@rv1126b-buildroot:/# com /dev/ttyS5 115200 8 0 1  
port = /dev/ttyS5  
baudrate = 115200  
cs = 8  
parity = 0  
stopb = 1  
klk1klk1klk1k  
RECV: klk1klk1klk1k  
opopopop  
RECV: opopopop  
8989898  
RECV: 8989898  
uiuiuiui  
RECV: uiuiuiui
```

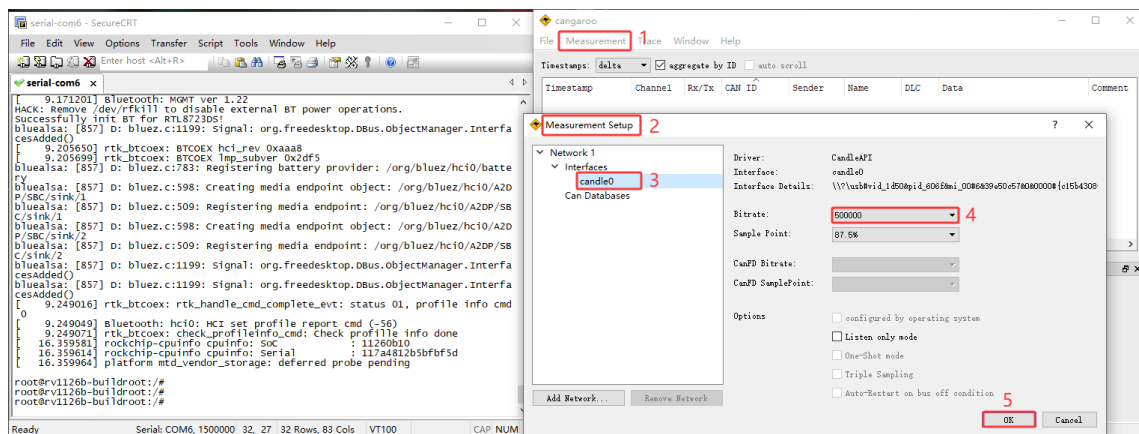

6.16 CAN



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 enable the CAN0 interface on the board with a bitrate of 500000.

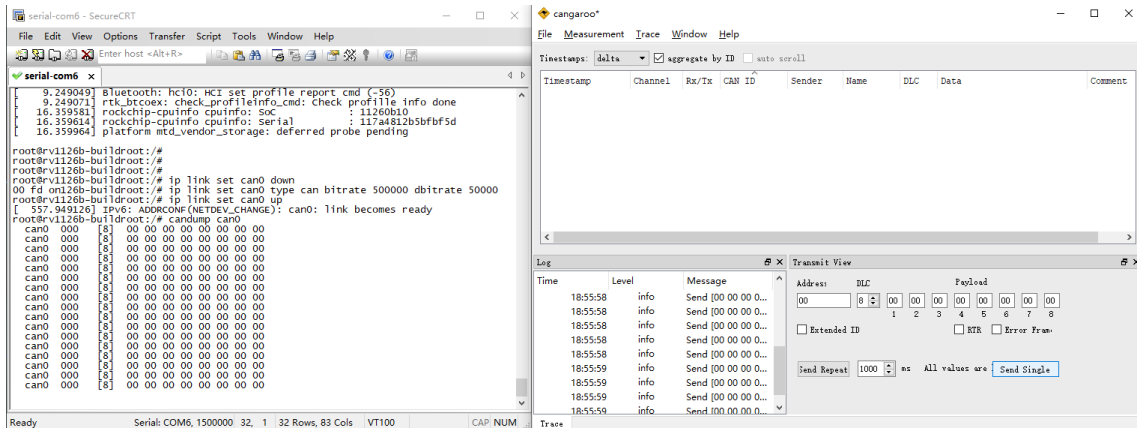
Execute the following command:

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

Step 4: Configure the CAN test tool as the sender, and configure CAN0 on the board as the receiver.

Execute the following command:

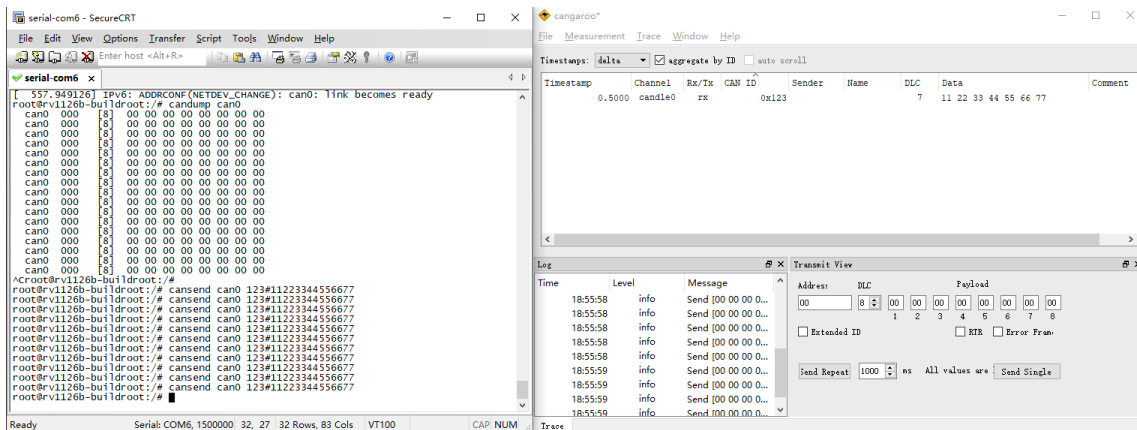
```
# candump can0
```



Step 5: Configure the CAN test tool as the receiver, and configure CAN0 on the board as the sender.

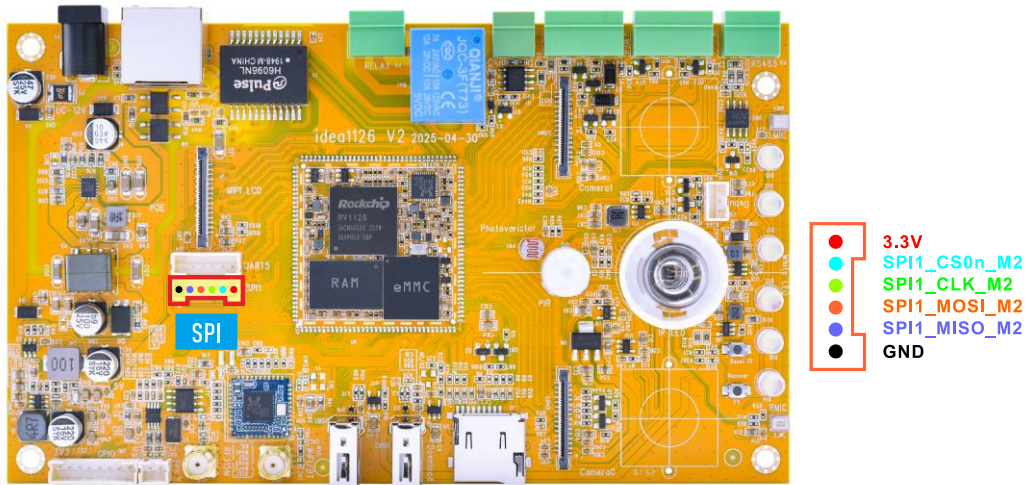
Execute the following command:

```
# cansend can0 123#11223344556677
```



6.17 SPI

Step 1: Short the **MOSI** and **MISO** pins of the SPI interface to create a loopback connection.



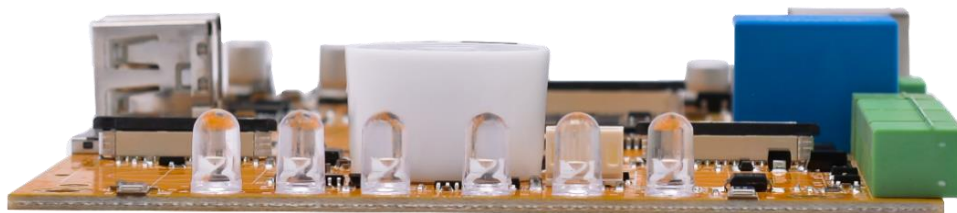
Step 2: Execute the following command to perform the SPI loopback test:

```
# spidev_test -D /dev/spidev1.0 -v
```

```
root@rv1126b-buildroot:/# spidev_test -D /dev/spidev1.0 -v
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 kHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0D
|.....@.....|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0D
|.....@.....|
root@rv1126b-buildroot:/#
```

6.18 LED

The board provides user LEDs for function testing. The LED can be controlled by writing values to the brightness node.



Step 1: Execute the following command to turn on the LED:

```
# echo 1 > /sys/class/leds/user_led/brightness
```

```
root@rv1126b-buildroot:/# echo 1 > /sys/class/leds/user_led/brightness
root@rv1126b-buildroot:/#
```

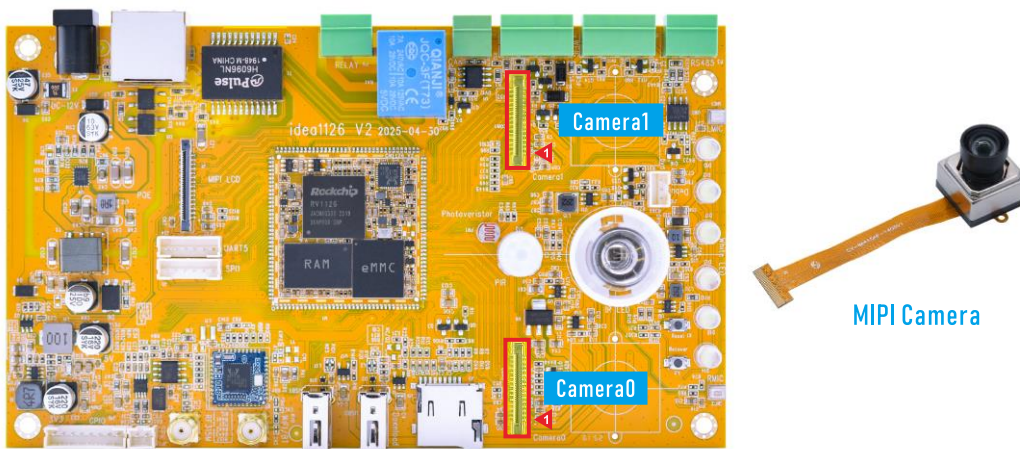
Step 2: Execute the following command to turn off the LED.

```
# echo 0 > /sys/class/leds/user_led/brightness
```

```
root@rv1126b-buildroot:/# echo 0 > /sys/class/leds/user_led/brightness
```

6.19 Camera

Connect the camera module, then power on the board.



6.19.1 GStreamer Version

Step 1: Check the video device nodes.

Execute the following command:

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

```
root@rv1126b-buildroot:/# grep "" /sys/class/video4linux/v*/name | grep mainpath
/sys/class/video4linux/video23/name:rkisp_mainpath4linux/v*/name | grep mainpath
/sys/class/video4linux/video31/name:rkisp_mainpath
```

The output shows that camera0 uses the device node: **/dev/video23**, and camera1 uses the device node: **/dev/video31**.

The following steps use camera0 as an example.

Step 2: Preview the camera.

```
# gst-launch-1.0 v4l2src device=/dev/video23 ! video/x-raw,format=NV16,width=1920,height=1080, framerate=25/1 ! waylandsink
```

```
root@rv1126b-buildroot:/#
aw,format=NV16,width=1920,height=1080, framerate=25/1 ! waylandsink3 ! video/x-ra
Setting pipeline to PAUSED ...
Using mplane plugin for capture
Pipeline is live and does not need PREROLL ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[ 16.141467] rkisp_hw 21d00000.isp: set isp clk = 500000000Hz
[ 16.149818] rkCIF-mipi-lvds: stream[0] start streaming
[ 16.150116] rockchip-mipi-csi2 mipi0-csi2: stream on, src_sd: 00000000f26425db, sd_name:rockchip-csi2-
dphy0
[ 16.150139] rockchip-mipi-csi2 mipi0-csi2: stream ON
[ 16.150190] rockchip-csi2-dphy0: dphy0, data_rate_mbps 892
[ 16.150223] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:1, dphy0, ret 0
[ 16.150237] imx415 5-0036: s_stream: 1. 3864x2192, hdr: 0, bpp: 10
[ 16.255038] rkCIF-mipi-lvds: Warning: vblank need >= 1000us if isp work in online, cur 859 us
[ 16.290084] rkCIF-mipi-lvds: Warning: vblank need >= 1000us if isp work in online, cur 859 us
Redistribute latency...
[00:00:14.772] seeing the first app
[ 16.359269] rockchip-cpuinfo cpuinfo: SoC : 11260b10
[ 16.359383] rockchip-cpuinfo cpuinfo: Serial : 117a4812b5bfbf5d
[ 16.359989] platform mtd_vendor_storage: deferred probe pending
0:00:04.0 / 99:99:99.
```

Step 4: Record a video.

```
# gst-launch-1.0 v4l2src device=/dev/video23 num-buffers=100 ! \
video/x-raw,format=NV12,width=1920,height=1080,framerate=30/1 ! \
videoconvert ! mpph264enc ! h264parse ! mp4mux ! \
filesink location=/tmp/h264.mp4
```

```

root@rv1126b-buildroot:/#
=100 ! \126b-buildroot:/# gst-launch-1.0 v4l2src device=/dev/video23 num-buffers=
> video/x-raw,format=NV12,width=1920,height=1080,framerate=30/1 ! \
> videoconvert ! mpph264enc ! h264parse ! mp4mux ! \
> filesink location=/tmp/h264.mp4
mpp[1321]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1321]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1321]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1321]: mpp: unable to create enc vp8 for soc rv1126b unsupported
mpp[1321]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
Setting pipeline to PAUSED ...
Using mplane plugin for capture
mpp[1321]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1321]: mpp: Only rk3588's h264/265/jpeg and rk3576's h264/265 encoder can use frame parallel
Pipeline is live and does not need PREROLL ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
mpp[1321]: mpp_enc: set prep cfg w:h [1920:1080] stride [1920:1088] fmt 0 rotate 0 mirror 0
mpp[1321]: mpp_enc: set rc cbr bps [7776000:8262000:7290000] fps [30:1:fix] - [30:1:fix] gop 30
[ 44.066939] rkisp_hw 21d00000.isp: set isp clk = 500000000Hz
mpp[1321]: mpp_enc: mode cbr bps [7290000:7776000:8262000] fps fix [30/1] -> fix [30/1] gop i [30] v [0]
[ 44.082704] rkisp rkisp-vir0: first params buf queue
[ 44.082966] rkCIF-mipi-lvds: stream[0] start streaming
[ 44.083243] rockchip-mipi-csi2 mipi0-csi2: stream on, src_sd: 00000000f26425db, sd_name:rockchip-csi2-
dphy0
[ 44.083260] rockchip-mipi-csi2 mipi0-csi2: stream ON
[ 44.083302] rockchip-csi2-dphy0: dphy0, data_rate_mbps 892
[ 44.083334] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:1, dphy0, ret 0
[ 44.083347] imx415 5-0036: s_stream: 1. 3864x2192, hdr: 0, bpp: 10
[ 44.187982] rkCIF-mipi-lvds: Warning: vblank need >= 1000us if isp work in online, cur 859 us
[ 44.223030] rkCIF-mipi-lvds: Warning: vblank need >= 1000us if isp work in online, cur 859 us
Redistribute latency...
Redistribute latency...
Got EOS from element "pipeline0".
Execution ended after 0:00:03.489676751
Setting pipeline to NULL ...
[ 47.545954] rkCIF-mipi-lvds: stream[0] start stopping, total mode 0x8, cur 0x8
[ 47.587545] rockchip-mipi-csi2 mipi0-csi2: stream off, src_sd: 00000000f26425db, sd_name:rockchip-csi2-
dphy0
[ 47.587589] rockchip-mipi-csi2 mipi0-csi2: stream OFF
[ 47.587619] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream_stop stream stop, dphy0
[ 47.587641] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:0, dphy0, ret 0
[ 47.587675] imx415 5-0036: s_stream: 0. 3864x2192, hdr: 0, bpp: 10
[ 47.588217] rkCIF-mipi-lvds: stream[0] stopping finished, dma_en 0x0
Freeing pipeline ...

```

Step 5: Capture JPEG images.

```

# gst-launch-1.0 -v v4l2src device=/dev/video23 num-buffers=10 ! \
video/x-raw,format=NV12,width=1920,height=1080 ! mppjpegenc ! \
multifilesink location=/tmp/test%05d.jpg

```



```

ers=10 ! \6b-buildroot:/# gst-launch-1.0 -v v4l2src device=/dev/video23 num-buffe
> video/x-raw,format=NV12,width=1920,height=1080 ! mppjpegenc ! \
> multifilesink location=/tmp/test%05d.jpg
mpp[1340]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1340]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1340]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1340]: mpp: unable to create enc vp8 for soc rv1126b unsupported
mpp[1340]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
Setting pipeline to PAUSED ...
Using mplane plugin for capture
mpp[1340]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1340]: mpp: Only rk3588's h264/265/jpeg and rk3576's h264/265 encoder can use frame parallel
Pipeline is live and does not need PREROLL ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
/GstPipeline:pipeline0/GstV4l2Src:v4l2src0: crop-bounds = < (int)0, (int)0, (int)3840, (int)2160 >
New clock: GstSystemClock
/GstPipeline:pipeline0/GstV4l2Src:v4l2src0.GstPad:src: caps = video/x-raw, format=(string)NV12,
width=(int)1920, height=(int)1080, pixel-aspect-ratio=(fraction)1/1, framerate=(fraction)120/1, interlace-
mode=(string)progressive, colorimetry=(string)bt709
/GstPipeline:pipeline0/GstCapsFilter:capsfilter0.GstPad:src: caps = video/x-raw, format=(string)NV12,
width=(int)1920, height=(int)1080, pixel-aspect-ratio=(fraction)1/1, framerate=(fraction)120/1, interlace-
mode=(string)progressive, colorimetry=(string)bt709
[ 92.815225] rkisp_hw 21d00000.isp: set isp clk = 500000000Hz
mpp[1340]: mpp_enc: set prep cfg w:h [1920:1080] stride [1920:1088] fmt 0 rotate 0 mirror 0
mpp[1340]: mpp_enc: set rc cbr bps [31104000:33048000:29160000] fps [120:1:fix] - [120:1:fix] gop 120
mpp[1340]: mpp_enc: set jpeg qfactor [80:1:99]
/GstPipeline:pipeline0/GstMppJpegEnc:mppjpegenc0.GstPad:src: caps = image/jpeg, width=(int)1920,
height=(int)1080, pixel-aspect-ratio=(fraction)1/1, framerate=(fraction)120/1, interlace-
mode=(string)progressive, colorimetry=(string)bt709
/GstPipeline:pipeline0/GstMultiFileSink:multifilesink0.GstPad:sink: caps = image/jpeg, width=(int)1920,
height=(int)1080, pixel-aspect-ratio=(fraction)1/1, framerate=(fraction)120/1, interlace-
mode=(string)progressive, colorimetry=(string)bt709
/GstPipeline:pipeline0/GstMppJpegEnc:mppjpegenc0.GstPad:sink: caps = video/x-raw, format=(string)NV12,
width=(int)1920, height=(int)1080, pixel-aspect-ratio=(fraction)1/1, framerate=(fraction)120/1, interlace-
mode=(string)progressive, colorimetry=(string)bt709
/GstPipeline:pipeline0/GstCapsFilter:capsfilter0.GstPad:sink: caps = video/x-raw, format=(string)NV12,
width=(int)1920, height=(int)1080, pixel-aspect-ratio=(fraction)1/1, framerate=(fraction)120/1, interlace-
mode=(string)progressive, colorimetry=(string)bt709
[ 92.830810] rkisp rkisp-vir0: first params buf queue
[ 92.831081] rkCIF-mipi-lvds: stream[0] start streaming
[ 92.831350] rockchip-mipi-csi2 mipi0-csi2: stream on, src_sd: 00000000f26425db, sd_name:rockchip-csi2-
dphy0
[ 92.831370] rockchip-mipi-csi2 mipi0-csi2: stream ON
[ 92.831414] rockchip-csi2-dphy0: dphy0, data_rate_mbps 892
[ 92.831447] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:1, dphy0, ret 0
[ 92.831460] imx415 5-0036: s_stream: 1. 3864x2192, hdr: 0, bpp: 10
[ 92.935791] rkCIF-mipi-lvds: Warning: vblank need >= 1000us if isp work in online, cur 859 us
[ 92.970200] rkCIF-mipi-lvds: Warning: vblank need >= 1000us if isp work in online, cur 859 us
Redistribute latency...

```

6.19.2 Rokit Version

Step 1: Stop the default rkipc service.

Execute the following command:

```
# killall -9 rkipc
```



```

root@rv1126b-buildroot:/# killall -9 rkipp
root@rv1126b-buildroot:/# [ 22.785495] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 3 hnd
00000000d04057c online 0 combo -1 mst -1
[ 22.785683] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 3 done
[ 22.785700] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 1 hnd 000000001eb426f7 online 0
combo -1 mst -1
[ 22.785954] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 1 done
[ 22.785970] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 0 hnd 00000000cb164542 online 0
combo -1 mst -1
[ 22.786193] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 0 done
[ 22.790189] kmpp_venc_chan_put_frm: invalid chan id 0
[ 22.790239] kmpp_venc_chan_put_frm: invalid chan id 1
[ 22.823493] kmpp_venc_chan_put_frm: invalid chan id 0
[ 22.823560] kmpp_venc_chan_put_frm: invalid chan id 1
[ 22.856816] kmpp_venc_chan_put_frm: invalid chan id 0
[ 22.856880] kmpp_venc_chan_put_frm: invalid chan id 1
[ 22.890137] kmpp_venc_chan_put_frm: invalid chan id 0
[ 22.890202] kmpp_venc_chan_put_frm: invalid chan id 1
[ 22.923471] kmpp_venc_chan_put_frm: invalid chan id 0
[ 22.923561] kmpp_venc_chan_put_frm: invalid chan id 1
[ 22.956804] kmpp_venc_chan_put_frm: invalid chan id 0
[ 22.956887] kmpp_venc_chan_put_frm: invalid chan id 1
[ 22.990148] kmpp_venc_chan_put_frm: invalid chan id 0
[ 22.990229] kmpp_venc_chan_put_frm: invalid chan id 1
[ 23.023529] kmpp_venc_chan_put_frm: invalid chan id 0
[ 23.023618] kmpp_venc_chan_put_frm: invalid chan id 1
[ 23.056913] kmpp_venc_chan_put_frm: invalid chan id 0
[ 23.057009] kmpp_venc_chan_put_frm: invalid chan id 1
[ 23.090231] kmpp_venc_chan_put_frm: invalid chan id 0
[ 23.090330] kmpp_venc_chan_put_frm: invalid chan id 1
[ 23.105516] rkvpss-vir0: rkvpss_stream_stop id:3 timeout
[ 23.123589] kmpp_venc_chan_put_frm: invalid chan id 0
[ 23.156746] rkCIF-mipi-lvds: stream[0] start stopping, total mode 0x8, cur 0x8
[ 23.203605] rockchip-mipi-csi2 mipi0-csi2: stream off, src_sd: 000000002ca583ee, sd_name:rockchip-csi2-
dphy0
[ 23.203631] rockchip-mipi-csi2 mipi0-csi2: stream OFF
[ 23.203653] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream_stop stream stop, dphy0
[ 23.203666] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:0, dphy0, ret 0
[ 23.203689] imx415 5-0036: s_stream: 0. 3864x2192, hdr: 0, bpp: 10
[ 23.204181] rkCIF-mipi-lvds: stream[0] stopping finished, dma_en 0x0
[ 24.225283] rkisp-vir0: waiting on params stream off event timeout
[ 24.545278] rkvpss-vir0: rkvpss_stream_stop id:0 timeout
[ 24.552568] ivs_release 20
[ 24.552880] vi_release 27
[ 24.552927] rockit vpss stream off
[ 24.552992] rockit vpss stream off
[ 24.553029] rockit vpss stream off
[ 24.553107] venc_release 19
[ 24.553235] vrgn_release 30
[ 24.554337] vsys_release, 6
[ 24.613778] dw9714 5-000c: dw9714_set_power(1046) on(0)

```

Step 2: Check the network status and obtain the Ethernet IP address of the board.

Execute the following command:

```
# ifconfig eth0
```

```

root@rv1126b-buildroot:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 1A:25:9A:BA:66:BF
          inet addr:192.168.0.205  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::4c00:9dc6:a350:5a53/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17 errors:0 dropped:1 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1614 (1.5 KiB)  TX bytes:1200 (1.1 KiB)
          Interrupt:80

```

Step 3: Start the camera demo.

Execute the following command:

```
# sample_demo_dual_camera -s 0 -W 1920 -H 1080 -w 720 -h 576 -f 30 -r 0 -s 1 -W
1920 -H 1080 -w 720 -h 576 -f 30 -r 0 -n 0 -b 1
```

```
root@rv1126b-buildroot:/#
576 -f 30 -r 0 -s 1 -W 1920 -H 1080 -w 720 -h 576 -f 30 -r 0 -n 0 -b 1-w 720 -h
sensor 0: main:1920*1080, sub:720*576
sensor 1: main:1920*1080, sub:720*576
#CodecName:H265
#IQ Path: /oem/usr/share/iqfiles
#Rkaiq XML DirPath: /oem/usr/share/iqfiles
#bMultictx: 1

rkaiq log level ff1
XCORE:K:rk_aiq_init_lib, ISP HW ver: 35

***** VERSION INFOS *****
version release date: 2025-11-07
AIQ: AIQ v6.0x32.0
git logs:
b99f06f boardcon: update IMX415 IQ file for EM1126BP
bd99027 uAPI: add bSingleExp for API setOnetoMultiMode

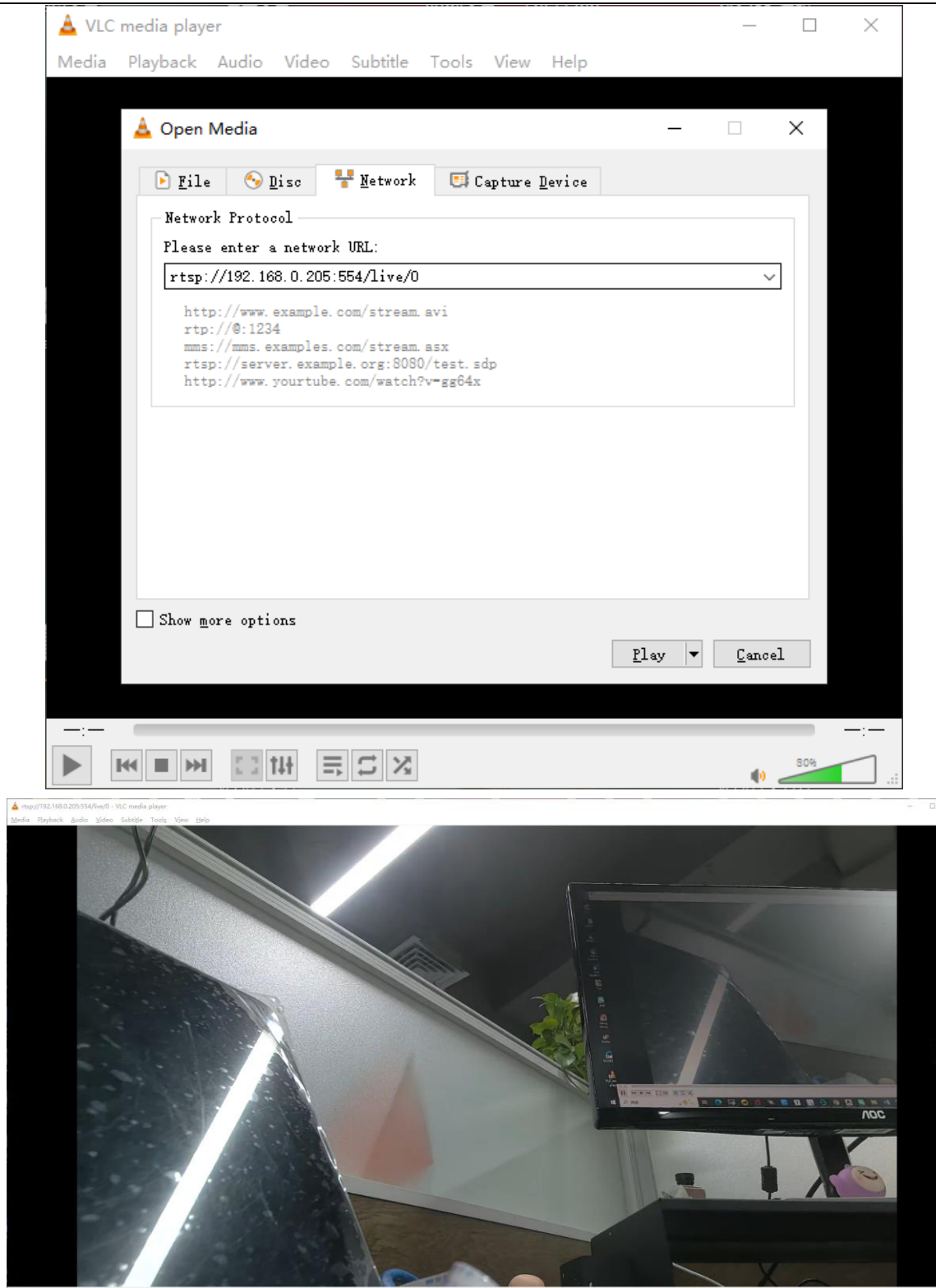
***** VERSION INFOS END *****
ID: 0, sensor_name is m00_b_imx415 5-0036, iqfiles is /oem/usr/share/iqfiles
XCORE:K:rk_aiq_uapi2_sysctl_preInit_scene: main_scene: normal, sub_scene: day
XCORE:K:m00_b_imx415 5-0036: create preinitCfg success !
```

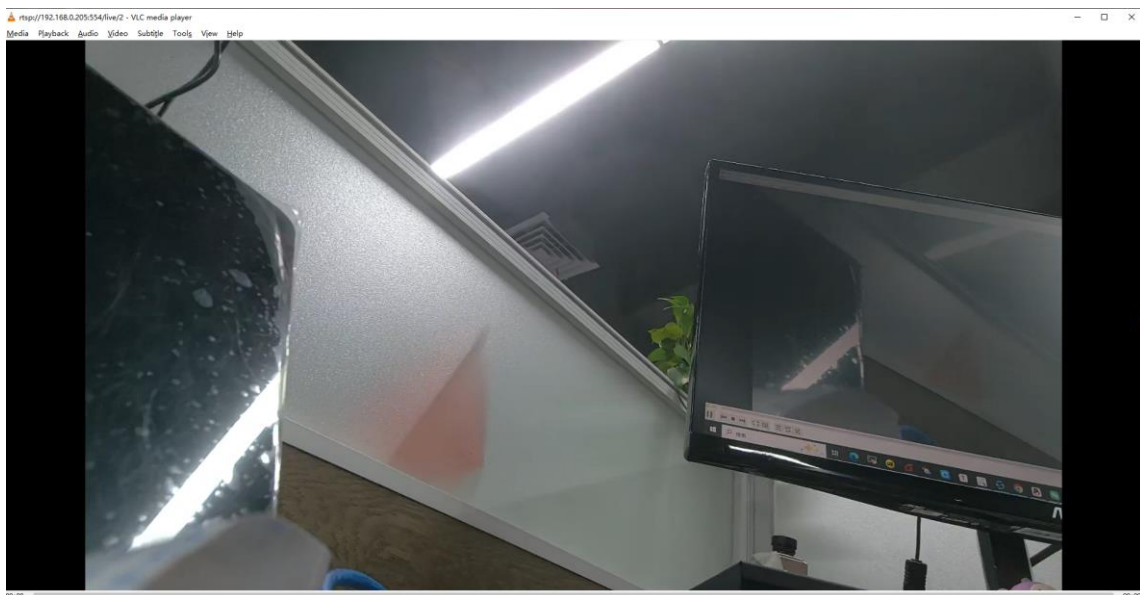
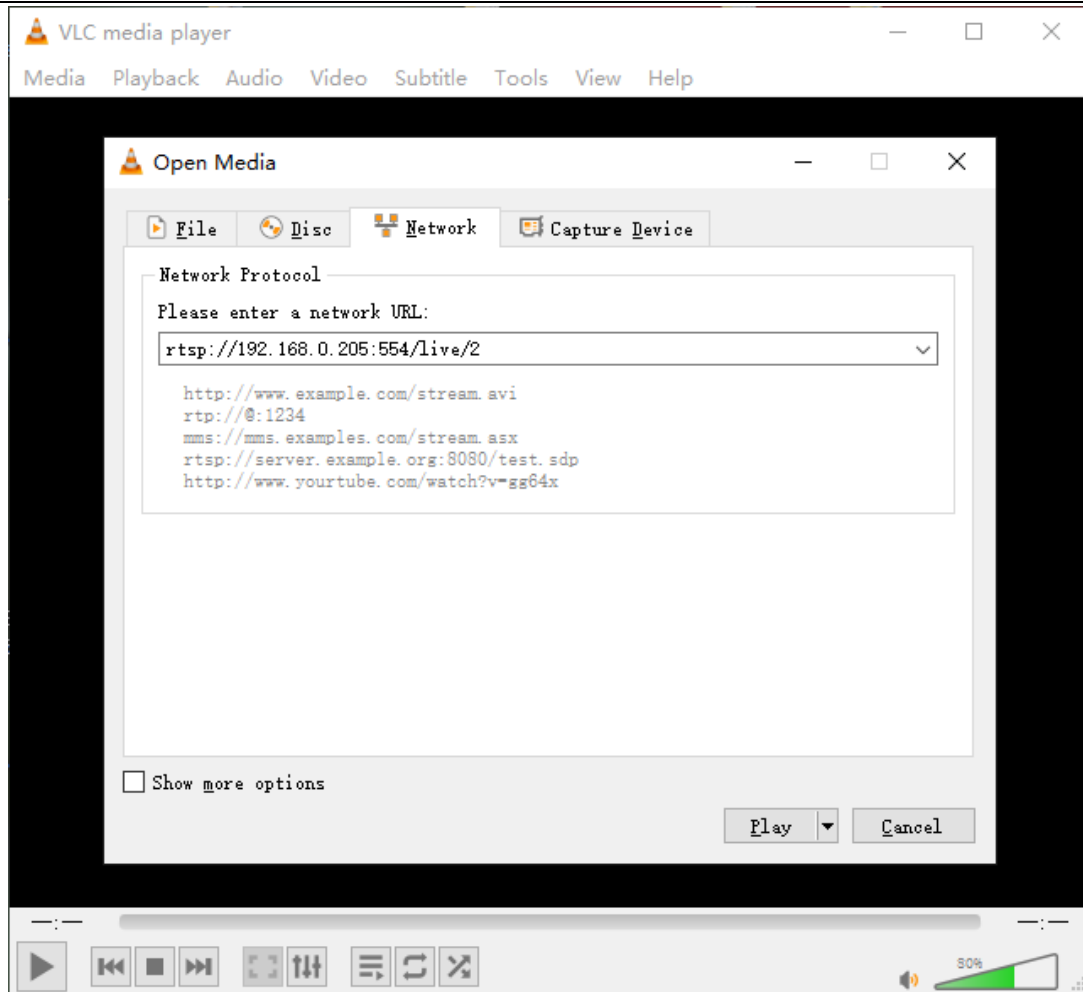
Step 4: Preview the camera stream on the PC.

Open VLC media player on the PC, select **Media -> Open Network Stream...**, and enter one of the following RTSP URLs.

Camera0:

```
rtsp://192.168.0.205:554/live/0
```



**Note:**

Replace 192.168.0.205 with the actual Ethernet IP address of the board.

6.20 Video Playback

6.20.1 GStreamer Version

(1) Use the built-in video test scripts.

Execute the following command:

```
# /rockchip-test/video/test_gst_video.sh
```

```
root@rv1126b-buildroot:/# /rockchip-test/video/test_gst_video.sh
Setting pipeline to PAUSED ...
Pipeline is PREROLLING ...
Redistribute latency...
mpp[2290]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out_lowdelay mode
mpp[2290]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out_lowdelay mode
mpp[2290]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out_lowdelay mode
mpp[2290]: mpp: unable to create enc vp8 for soc rv1126b unsupported
mpp[2290]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out_lowdelay mode
mpp[2290]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out_lowdelay mode
Redistribute latency...
mpp[2290]: h264d_api: is_avcC=1
mpp[2290]: mpp_buf_slot: mismatch h_stride_by_pixel 1472 - 1280
mpp[2290]: mpp_buf_slot: mismatch h_stride_by_byte 1472 - 1280
mpp[2290]: mpp_buf_slot: mismatch size_total 1589760 - 1843200
mpp[2290]: mpp_buf_slot: mismatch h_stride_by_pixel 1472 - 1280
mpp[2290]: mpp_buf_slot: mismatch h_stride_by_byte 1472 - 1280
mpp[2290]: mpp_buf_slot: mismatch size_total 1589760 - 1843200
Pipeline is PREROLLED ...
Prerolled, waiting for async message to finish...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
Redistribute latency...
0:00:03.9 / 0:00:29.5 (13.3 %)
```

(2) Play videos using the gst-play-1.0 command.

Execute the following command:

```
# export GST_MPP_VIDEODEC_DEFAULT_ARM_AFBC=1
# gst-play-1.0 --videosink="waylandsink fullscreen=true"
/mnt/udisk/video/4KP30/4K30P-BLACKPINK-DDU-DU_DDU-DU.mp4 --audiosink="alsasink
device=hw:0,0"
```



```
root@rv1126b-buildroot:/# export GST_MPP_VIDEODEC_DEFAULT_ARM_AFBC=1
device=hw:0,0"de0/4KP30/4K30P-BLACKPINK-DDU-DU_DDU-DU.mp4" --audiosink="alsasink d
Press 'k' to see a list of keyboard shortcuts.
Now playing /mnt/udisk/video/4KP30/4K30P-BLACKPINK-DDU-DU_DDU-DU.mp4
Redistribute latency...
Redistribute latency...
mpp[1346]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1346]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1346]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1346]: mpp: unable to create enc vp8 for soc rv1126b unsupported
mpp[1346]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1346]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
Redistribute latency...
Redistribute latency...
mpp[1346]: h264d_api: is_avcC=1
mpp[1346]: mpp_buf_slot: mismatch h_stride_by_pixel 4032 - 3840
mpp[1346]: mpp_buf_slot: mismatch h_stride_by_byte 4032 - 3840
mpp[1346]: mpp_buf_slot: mismatch size_total 13063680 - 16588800
mpp[1346]: mpp_buf_slot: mismatch h_stride_by_pixel 4032 - 3840
mpp[1346]: mpp_buf_slot: mismatch h_stride_by_byte 4032 - 3840
mpp[1346]: mpp_buf_slot: mismatch size_total 13063680 - 16588800
Redistribute latency...
[00:05:08.484] seeing the first app
0:00:20.9 / 0:03:28.8
```

Command explanation:

- export GST_MPP_VIDEODEC_DEFAULT_ARM_AFBC=1: [Open AFBC](#).
- --videosink="kmssink plane-id=74": [Specifies the plane](#). Which can be viewed using the command: `cat /sys/kernel/debug/dri/0/state | grep "plane\[\"`.
- /mnt/udisk/video/4KP30/4K30P-BLACKPINK-DDU-DU_DDU-DU.mp4: [The media file path to be played](#).
- --audiosink="alsasink device=hw:0,0": [Specifies the audio output device as hw:0,0](#).

6.20.2 Rockit Version

Step 1: Stop the default rkipc service.

Execute the following command:

```
# killall -9 rkipc
```

```

root@rv1126b-buildroot:/# killall -9 rkipc
root@rv1126b-buildroot:/# [ 19.533598] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 3 hnd
000000003ddc32e2 online 0 combo -1 mst -1
[ 19.533841] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 3 done
[ 19.533868] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 1 hnd 0000000030ab6293 online 0
combo -1 mst -1
[ 19.547082] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 1 done
[ 19.547137] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 0 hnd 00000000b16c964 online 0
combo -1 mst -1
[ 19.547374] mpp_vcodec_chan: mpp_vcodec_chan_destory: destroy chan 0 done
[ 19.554306] kmpp_venc_chan_put_frm: invalid chan id 0
[ 19.554912] kmpp_venc_chan_put_frm: invalid chan id 1
[ 19.587637] kmpp_venc_chan_put_frm: invalid chan id 0
[ 19.620837] rkCIF-mipi-lvds: stream[0] start stopping, total mode 0x8, cur 0x8
[ 19.667880] rockchip-mipi-csi2 mipi0-csi2: stream off, src_sd: 00000000a2f5faab, sd_name:rockchip-csi2-
dphy0
[ 19.667919] rockchip-mipi-csi2 mipi0-csi2: stream OFF
[ 19.667950] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream_stop stream stop, dphy0
[ 19.667972] rockchip-csi2-dphy csi2-dphy0: csi2_dphy_s_stream stream on:0, dphy0, ret 0
[ 19.668007] imx415 5-0036: s_stream: 0. 3864x2192, hdr: 0, bpp: 10
[ 19.668542] rkCIF-mipi-lvds: stream[0] stopping finished, dma_en 0x0
[ 20.706397] rkisp-vir0: waiting on params stream off event timeout
[ 21.026378] rkvpss-vir0: rkvpss_stream_stop id:0 timeout
[ 21.035507] ivs_release 20
[ 21.035833] vi_release 27
[ 21.035882] rockit vpss stream off
[ 21.035943] rockit vpss stream off
[ 21.035985] rockit vpss stream off
[ 21.036059] venc_release 19
[ 21.036191] vrgn_release 30
[ 21.049362] vsys_release, 6
[ 21.138784] dw9714 5-000c: dw9714_set_power(1046) on(0)

```

Step 2: The Rockit version supports video playback using the `simple_vdec_bind_vo` command.

Execute the following command:

```
# simple_vdec_bind_vo -i /mnt/udisk/testh264.h264
```



```

root@rv1126b-buildroot:/# simple_vdec_bind_vo -i /mnt/udisk/testh264.h264
Input: /mnt/udisk/testh264.h264, Bind: 1
rockit load start
v4l2 tx probe successv4l2_rx_probe success
rockit_load end
rockit log path (null), log_size = 0
log_file = (nil)
(null) 02:48:12-602 {log_level_init :209}

please use echo name=level > /tmp/rt_log_level set log level
name: all cmpi mb sys vdec venc rgn vpss vgs tde avs wbc vo vi ai ao aenc adec
log_level: 0 1 2 3 4 5 6

rockit default level 4, can use export rt_log_level=x, x=0,1,2,3,4,5,6 change
(null) 02:48:12-602 {read_log_level :083} text is all=4
(null) 02:48:12-602 {read_log_level :085} module is all, log_level is 4
(null) 02:48:12-602 {dump_version :060} -----
----
(null) 02:48:12-603 {dump_version :061} rockit version: git-5057bd373 Thu Dec 11 09:34:24
2025 +0800
(null) 02:48:12-603 {dump_version :062} rockit building: built-Chu 2025-12-15 10:15:00
(null) 02:48:12-603 {dump_version :063} -----
----
(null) 02:48:12-603 {monitor_log_level :141} #Start monitor_log_level thread, arg:(nil)
vsys dev open 4
[ 54.203995] vsys dev open 6
[ 54.215177] rkvpss-offline: CLK_CORE_VPSS set to 396000000 Hz (requested 400000000 Hz)
load library(librga.so) in reulative path
Video resolution: 800x1280, Codec: H.264
mpp[1465]: mpp_info: mpp version: 53ff3d39 author: Yandong Lin 2025-12-23 fix[hal_vepu]: poll max set to
1 on split out lowdelay mode
mpp[1465]: hal_264d_com: control info: fmt 0, w 800, h 1280
mpp[1465]: mpp_dec: mpp_dec_proc_cfg found MPP_DEC_SET_FRAME_INFO fmt 0
mpp[1465]: mpp_buf_slot: mismatch h_stride_by_pixel 960 - 800
mpp[1465]: mpp_buf_slot: mismatch h_stride_by_byte 960 - 800
mpp[1465]: mpp_buf_slot: mismatch size_total 1843200 - 2048000
mpp[1465]: mpp_buf_slot: set frame info: w 800 h 1280 hor 800 ver 1280
mpp[1465]: mpp_dec: setting default w 800 h 1280 h_str 800 v_str 1280
Video codec: H.264/AVC
Video: 800x1280, timebase: 1/1200000
[SEND] Frame 100, PTS=0 us
get a hdl = 0x558c383870
[ 54.529486] phys = 53278000
get a hdl = 0x558c3839d0
[ 54.533331] phys = 4af1f000
get a hdl = 0x558c3841e0
load library(librga.so) in reulative path
rga_api version 1.10.5_[9]

```

Command explanation:

- `simple_vdec_bind_vo`: Video decoding and display test tool.
- `-i`: Specifies the input video file.
- `/mnt/udisk/testh264.h264`: Input H.264 file path.

Note:

The Rockit version supports H.264 and H.265 elementary stream files, such as .h264 and .h265 files. This command does not support standard MP4 container files.