D-Robotics RDK Suite

This document is the user manual for the D-Robotics RDK Suite, providing developers with instructions and development guidance for products such as RDK X3, RDK X3 Module, and RDK Ultra. The content covers various aspects such as hardware design, system customization, application development, and algorithm toolchains. Users are welcome to update and experience, please refer to the Quick Start section for specific instructions.

- All **RDK X3** mentioned in this document use the Ubuntu 2.0 and 3.0 version operating system.

To check the system version number, you can use the following command `cat /etc/version`.

Overview of D-Robotics RDK Suite

D-Robotics Developer Kits, abbreviated as D-Robotics RDK Suite, is a robot development kit built on the D-Robotics intelligent chip, including RDK X3, RDK X3 Module. In combination with the TogetheROS.Bot robot middleware, the D-Robotics RDK Suite can help developers quickly build robot prototypes and carry out evaluation and verification work.

This document will provide detailed instructions on how to use the D-Robotics RDK Suite, including setting up the development environment, running example tutorials, developing applications, and customizing system images. Regardless of which hardware you choose to use, the content described in this document will provide you with a consistent user experience.

Product Introduction

RDK X3 (Sunrise X3) is a full-featured development board with 5Tops edge inference power. It provides developers with flexible hardware expansion and connection options by supporting a variety of sensors and expansion components.

RDK X3 Module (Sunrise X3 Module) is a compact core module that maintains the same specifications as RDK X3 and is compatible with the Raspberry Pi CM4 module in terms of size and interface. By combining with the expansion board, it can provide efficient computing and communication capabilities for various application scenarios.

RDK X3 Module

RDK X3

Version Release History

Version: 2.1.0

New Features:

- Improved srpi-config system configuration tool, supporting Wi-Fi connections, enabling/disabling SSH and VNC, enabling/disabling peripheral buses on the 40-pin connector, language localization configuration, CPU overclocking, ION memory size configuration, and more.
- Support for /boot/config.txt system configuration file, supporting options such as dtoverlay, CPU overclocking, and IO boot state configuration.
- Added yolov5s v6/v7 model examples.

Enhancements:

- Support for outputting boot logs and entering the user command-line interface on HDMI displays to facilitate user use.
- Support for more HDMI display resolutions, greatly enhancing compatibility.
- Optimized pre-installed software lists for Desktop and server versions, removing redundant items and adding necessary software, such as VLC.
- Optimized layout of the Desktop menu bar, simplifying options.
- Bluetooth functionality is enabled by default.
- Added C++ interface for post-processing, improving post-processing efficiency.
- Automatically mount USB flash drives using udisk2, solving the problem of not being able to access NTFS file systems after automatic mounting.
- Support for retaining VNC password file.
- VNC service is not automatically started by default to reduce system resource consumption. Users can enable it through the srpi-config tool.
- RDK X3 v2.1 and RDK Module development board's CPU can run at a maximum frequency of 1.5GHz in normal mode and 1.8GHz after overclocking.

Bug Fixes:

- Remove redundant kernel logs for Wi-Fi drivers.
- Modify apt source domain to sunrise.D-Robotics.cc.

Other updates:

- Support for the Chromium browser, users can install and use it with `sudo apt install chromium`.

Version: 2.0.0

This release brings many anticipated features and improvements, aiming to provide a better development experience and broader application support. Here are the main highlights of this version release:

Open-source:

- We have fully open-sourced the source code of the operating system, including the source code of system core modules and functional modules. Developers can freely view and modify the source code, providing greater flexibility for customization and optimization.
- Detailed code documentation and comments will be provided to developers to help them better understand and use the source code.
- We welcome developers to participate in code contribution and discussions through the open-source community, together driving the improvement and optimization of the operating system. The source code is maintained on D-Robotics.

Support for RDK X3 Module:

- We introduce a brand new core board development kit, the RDK X3 Module.
- The RDK X3 Module has a smaller size and is compatible with the Raspberry Pi CM4 interface.
- Developers can choose compatible third-party carrier boards according to their needs to expand the functionality and application scenarios of the core board.

Other updates:

- We have optimized existing functions, fixed known issues and vulnerabilities, improving the stability and performance of the operating system.
- Revised the documentation and help documents, providing more comprehensive and accurate technical information and guidelines.
- We provide lower-level APIs to facilitate developers for secondary development and integration, enabling them to customize software more flexibly.

1.1 Preparation

```
import Tabs from '@theme/Tabs';
import TabItem from '@theme/TabItem';
```

Before using the RDK X3 development board, the following preparations need to be made.

Development Preparation

Power Supply

The RDK X3 development board is powered by a USB Type C interface. It requires a power adapter that supports **5V/3A** to power the board. It is recommended to use the recommended power adapter model in the Basic Accessories List section.

Please do not power the development board through the USB interface of a computer, as it may cause abnormal power failure and repeated restarts due to insufficient power supply.

For more troubleshooting, please refer to the Common Questions section.

Storage

The RDK X3 development board uses a Micro SD card as the system boot medium, and it is recommended to use a storage card with a capacity of at least 8GB to meet the storage requirements of the Ubuntu system and application software.

Display

The RDK X3 development board supports HDMI display interface, and connects the development board and monitor via HDMI cable to support graphical desktop display.

Network Connection

The RDK X3 development board supports Ethernet and Wi-Fi network interfaces, and users can use either interface for network connection.

Frequently Asked Questions

Here are some common issues when using the development board for the first time:

- **Power on failure**: Please ensure that the recommended power adapter is used for power supply; please also ensure that the Micro SD card or eMMC of the development board has been flashed with the Ubuntu system image.
- **No response from USB Host interface**: Please make sure that no data cable is connected to the Micro USB interface of the development board.
- **Hot-plugging storage card during usage**: The development board does not support hot-plugging of Micro SD storage cards. If an accidental operation occurs, please restart the development board.

Important Notices

- Do not plug and unplug any devices other than USB, HDMI, and Ethernet cables when they are powered on.
- The Type C USB interface of RDK X3 is only used for power supply.
- Use USB Type C power cables from reputable brands; otherwise, power supply abnormalities may occur, leading to system power failure.

For more problem-solving, please refer to the Frequently Asked Questions section, and you can also visit the D-Robotics Developer Forum for assistance.

1.2 System burning

```
import Tabs from '@theme/Tabs';
import TabItem from '@theme/TabItem';
```

The RDK suite currently provides the Ubuntu 20.04 system image, which supports desktop graphical interaction.

Image download {#img_download}

Click **Download image** to enter the version selection page, select the corresponding version directory, and enter the file download page. Take downloading the 2.0.0 version of the system image as an example:

After downloading, unzip the Ubuntu system image file, such as `ubuntu-preinstalled-desktop-arm64.img`.

Version description:

- Version 2.0: Made based on the RDK Linux open source code package, supporting the full range of hardware such as RDK X3 and X3 module.

Click **Download image** to enter the version selection page, select the corresponding version directory, and enter the file download page. Take downloading the 2.0.0 version of the system image as an example:

After downloading, unzip the Ubuntu system image file, such as `ubuntu-preinstalled-desktop-arm64.img`

Version description:

- Version 2.0: Made based on the RDK Linux open source code package, supporting the full range of hardware such as RDK X3 and X3 module.

- desktop: Ubuntu system with a desktop, can be connected to an external screen and operated with a mouse
- server: Ubuntu system without a desktop, can be operated remotely through serial or network connection

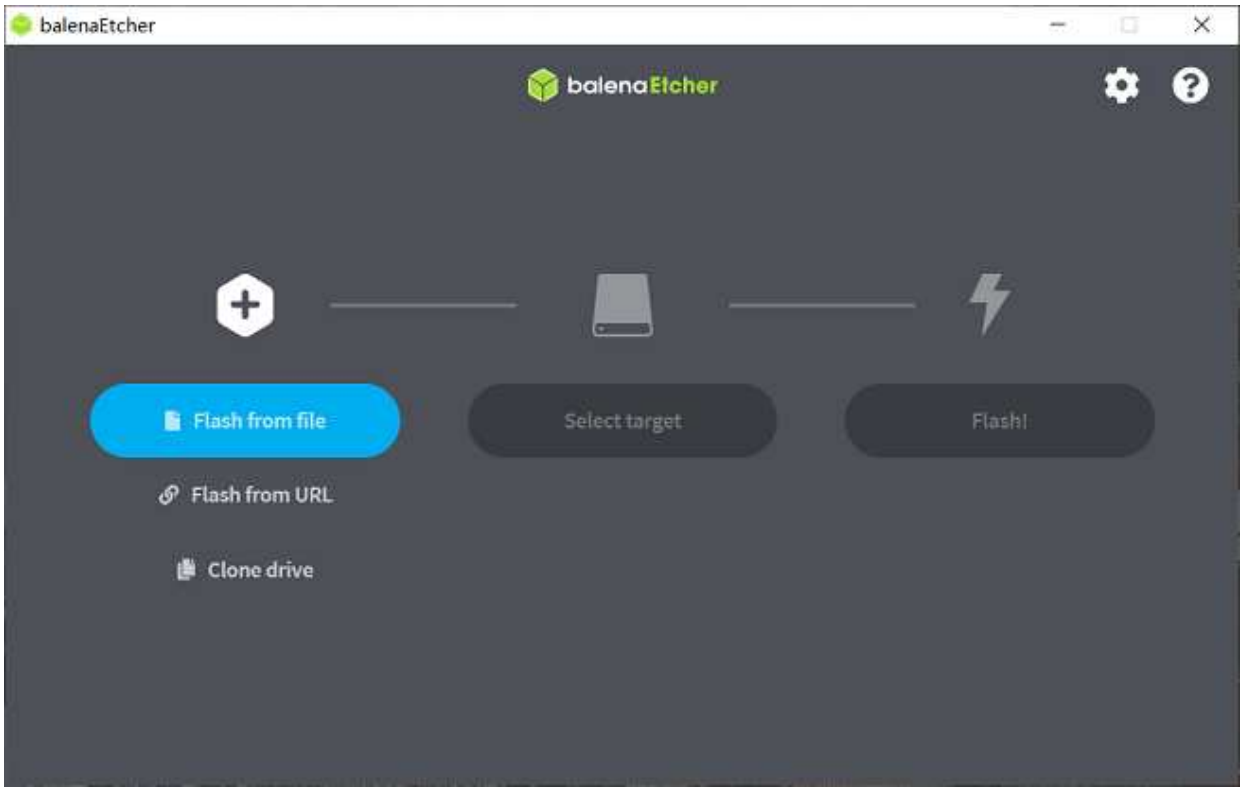
System Burning

Before burning the Ubuntu system image, please make the following preparations:

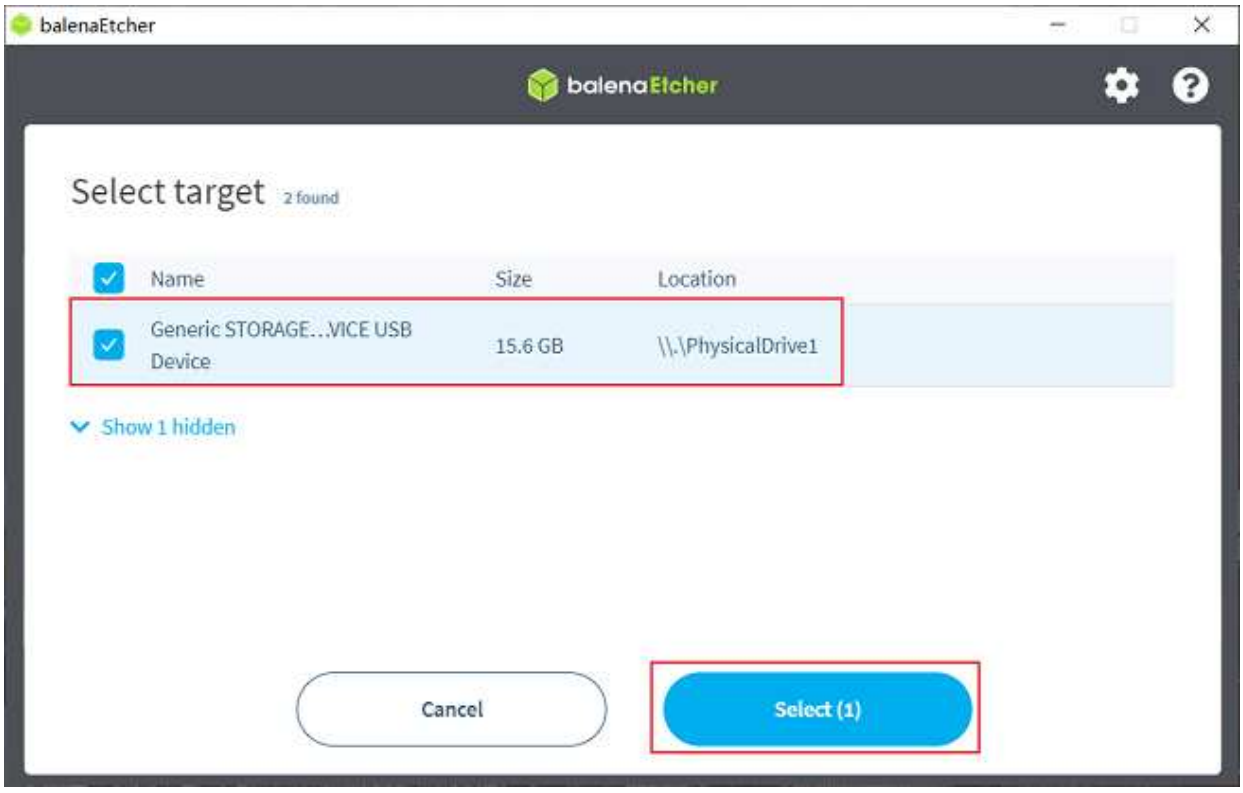
- Prepare a Micro SD card with a capacity of at least 8GB
- SD card reader
- Download the image burning tool balenaEtcher (available for download here)

balenaEtcher is a PC-side boot disk creation tool that supports multiple platforms such as Windows/Mac/Linux. The process of creating an SD boot card is as follows:

1. Open the balenaEtcher tool, click the `Flash from file` button, and select the extracted `ubuntu-preinstalled-desktop-arm64.img` file as the burning image.



2. Click the `Select target` button and select the corresponding Micro SD storage card as the target storage device.



3. Click the `Flash` button to start burning. When the tool prompts `Flash Complete`, it means the image burning is complete. You can close balenaEtcher and remove the storage card.

Start the system

First, keep the development board powered off, then insert the prepared memory card into the Micro SD card slot of the development board, and connect the development board to a monitor using an HDMI cable. Finally, power on the development board.The default environment configuration will be performed when the system starts up for the first time. The entire process takes about 45 seconds, and after the configuration is completed, the Ubuntu system desktop will be displayed on the monitor.

- **Red** indicator light: When it is on, it indicates normal hardware power-on.
- **Green** indicator light: When it is on, it indicates that the system is booting up. When it is off or flashing, it indicates that the system booting process is complete.

If there is no display output on the development board for a long time (more than 2 minutes) after power-on, it means that the development board failed to start. In this case, users can check the system status through the indicator lights using the following methods:

- **Green light** stays on: It indicates that the system failed to start. Users can check if the power adapter meets the requirements and try to remake the system image.
- **Green light** is off or flashing: It indicates that the system has started successfully, but the display service failed to start. Users need to confirm if the connected display meets the specification requirements.

After the Ubuntu Desktop version system is fully booted, the system desktop will be displayed on the monitor via the HDMI interface

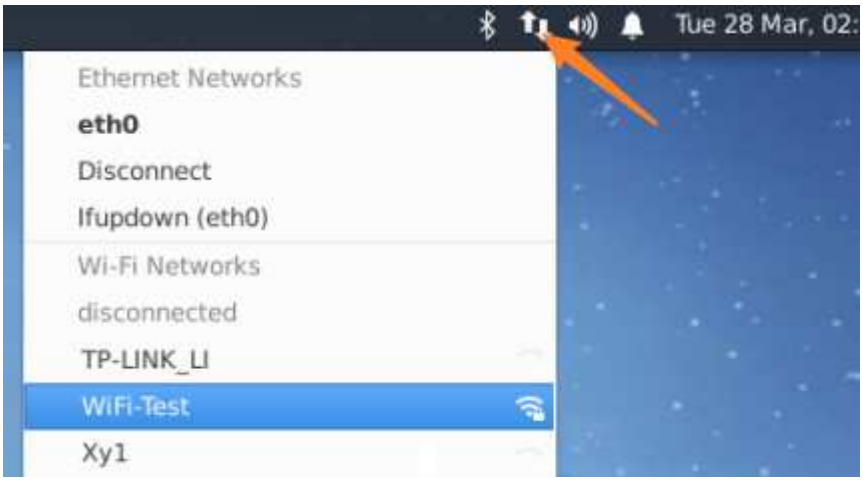
1.3 Getting Started Configuration

The getting started configuration methods described in this chapter are only supported on RDK X3 and RDK X3 Module models.

The system version should be no lower than `2.1.0`.

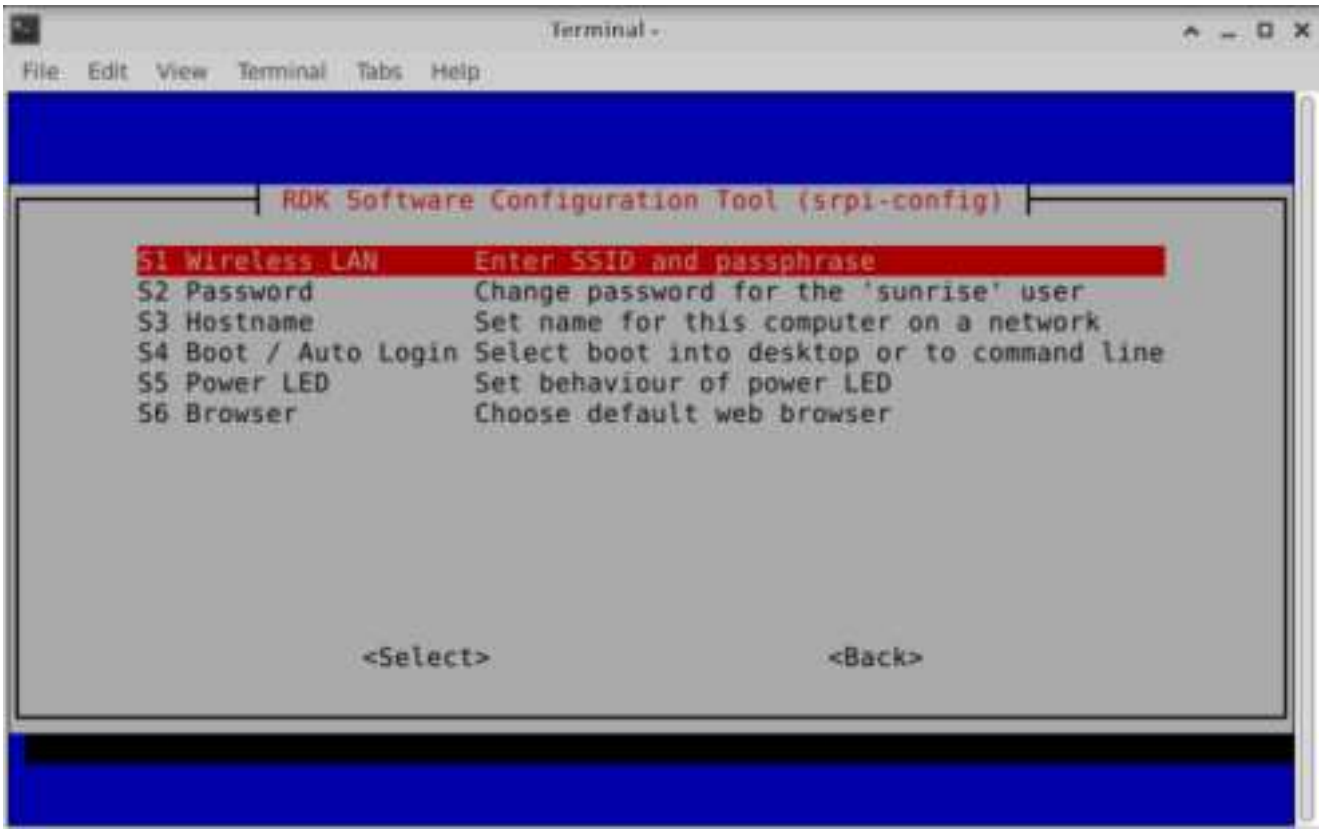
Connecting to Wi-Fi

Use the Wi-Fi management tool in the top-right corner of the menu bar to connect to Wi-Fi. As shown in the following figure, click on the Wi-Fi name you need to connect to, and then enter the Wi-Fi password in the pop-up dialog box.



Use the srpi-config tool to connect to Wi-Fi.

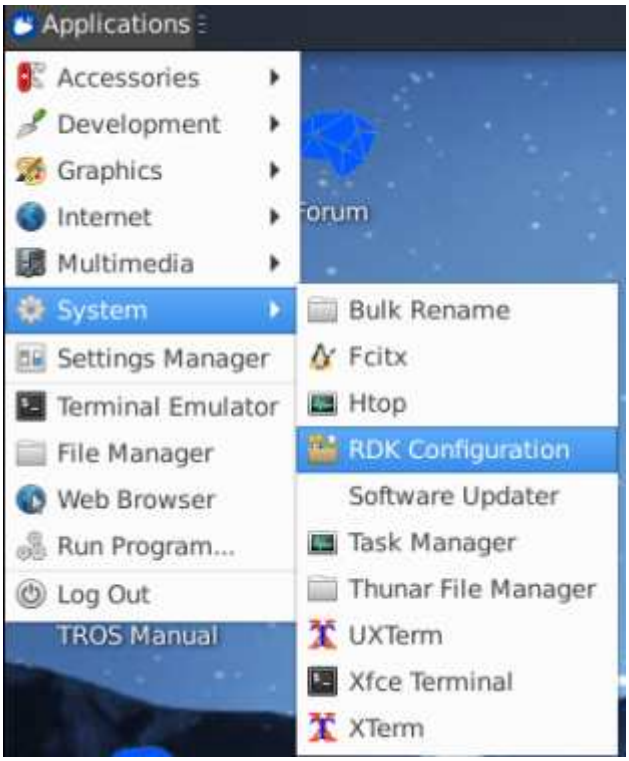
Execute the command `sudo srpi-config`, select System Options -> Wireless LAN, and enter the Wi-Fi name (`SSID`) and password (`passwd`) as prompted.



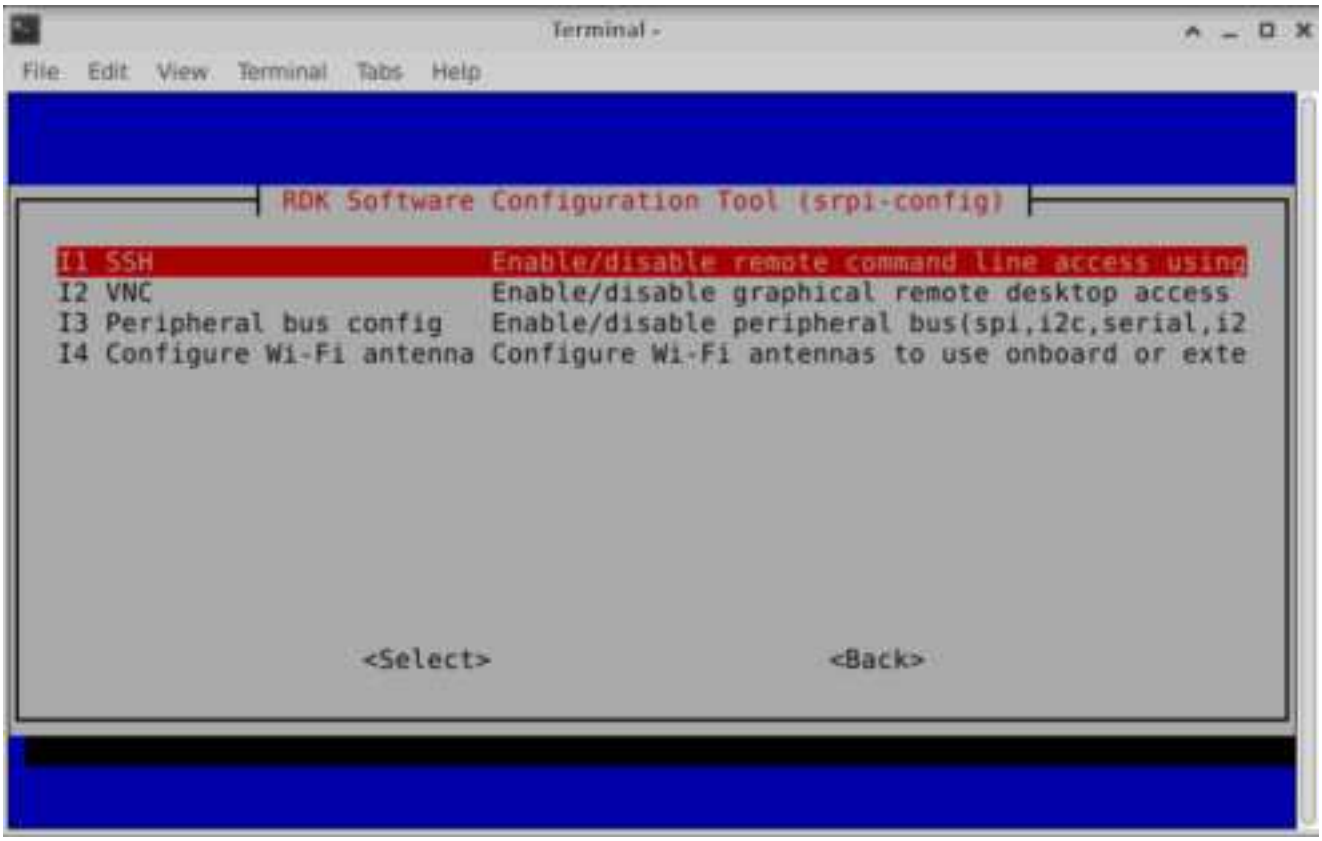
Enabling SSH Service

The SSH login service is enabled by default in the current system version. Users can use this method to toggle the service.

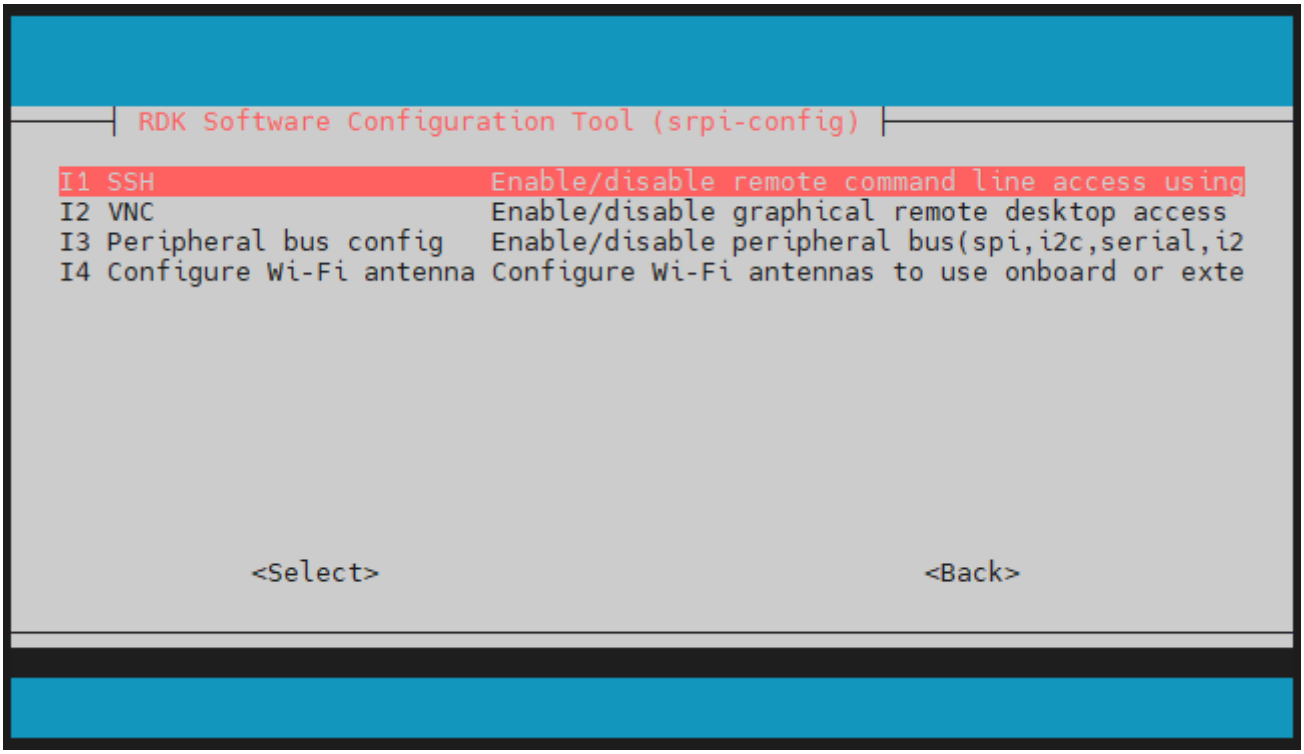
Find the `RDK Configuration` option through the menu bar and click to open it.



Select Interface Options -> SSH, and follow the prompts to enable or disable the SSH service.



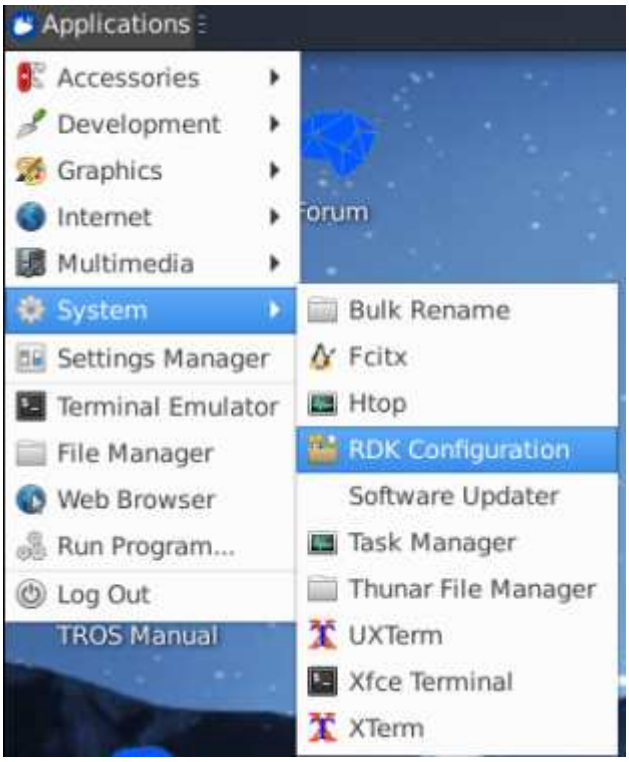
Execute the `sudo srpi-config` command to enter the configuration menu. Select Interface Options -> SSH, and follow the prompts to enable or disable the SSH service.



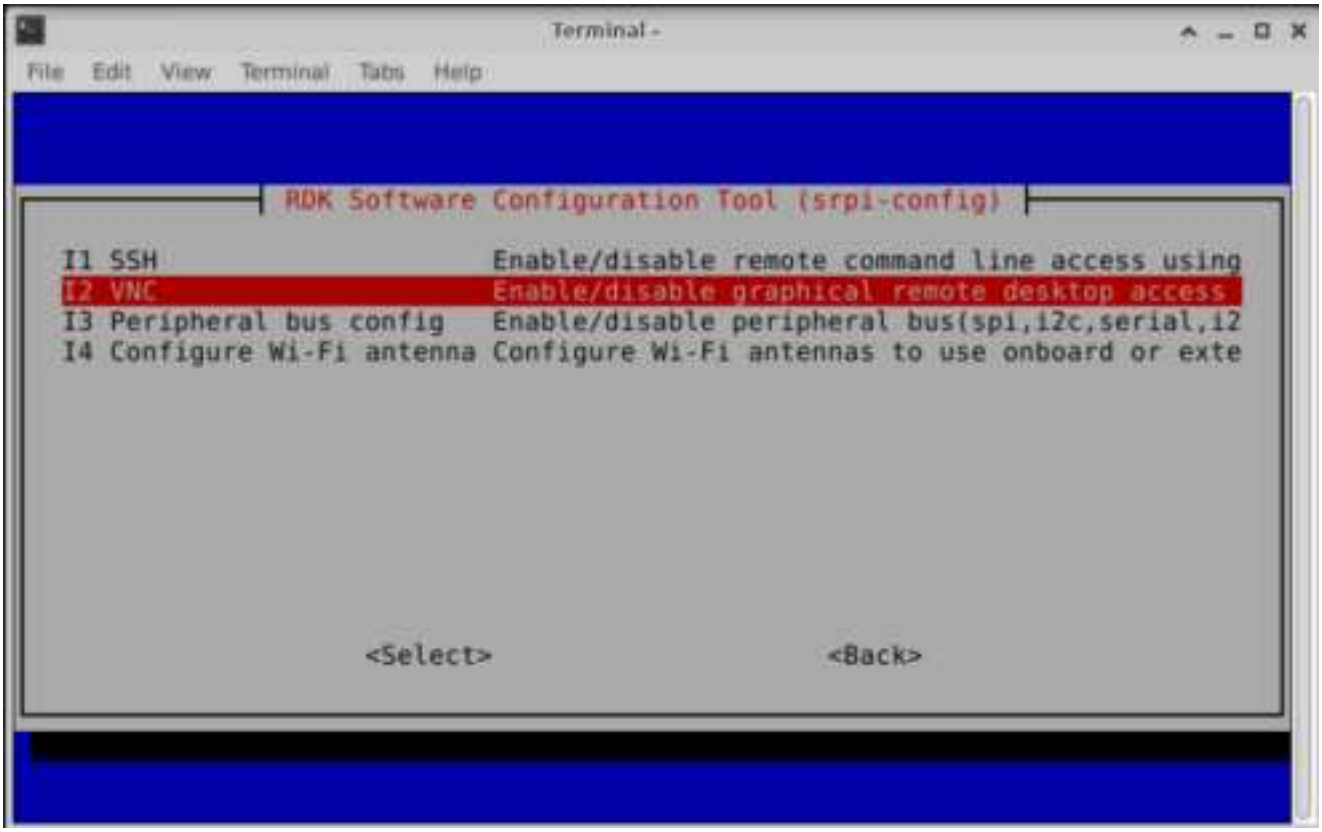
Please refer to Remote Login - SSH Login for how to use SSH.

Enable VNC Service

Find the `RDK Configuration` option through the menu bar and click to open it.



Select Interface Options -> VNC, and follow the prompts to enable or disable the VNC service. When enabling VNC, you need to set a login password, which must be an 8-character string composed of numbers and characters.



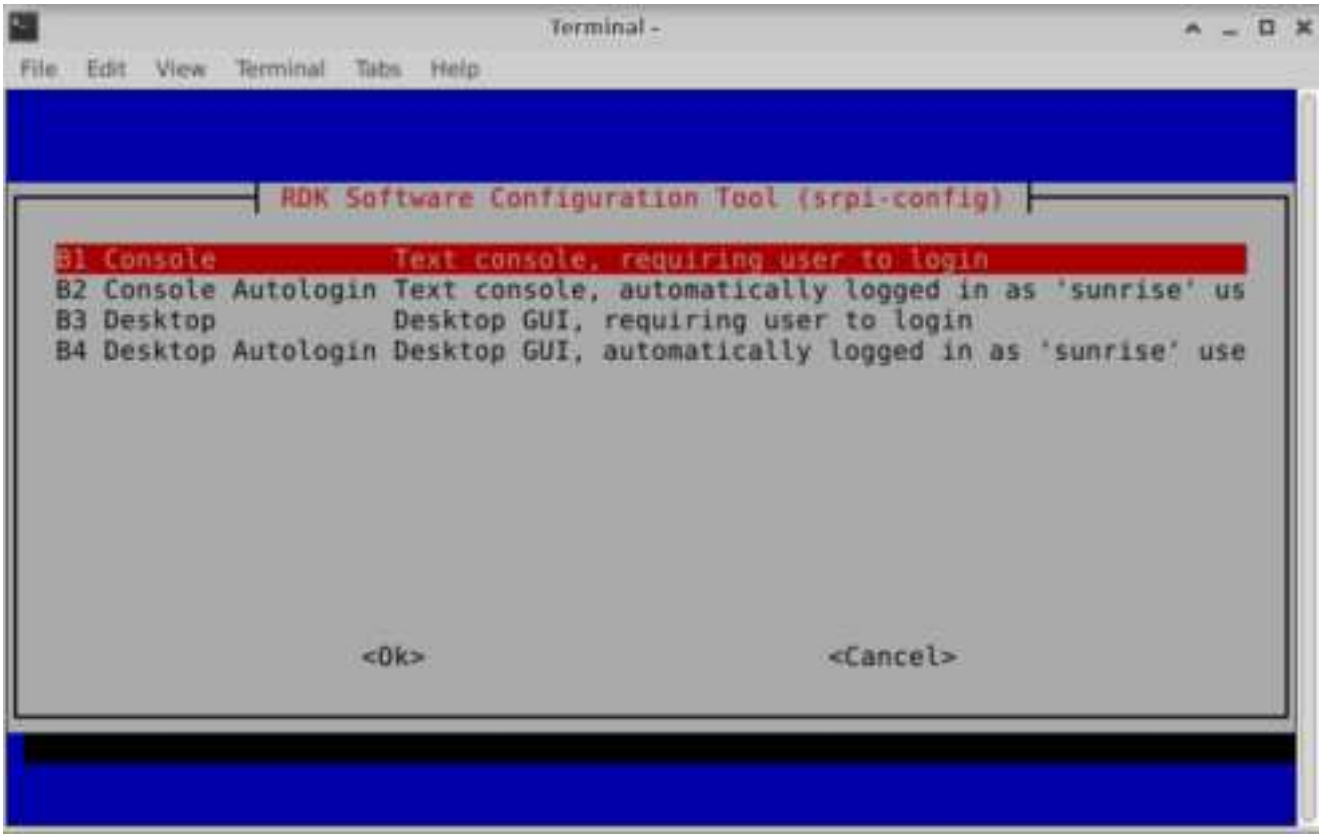
Please refer to Remote Login - VNC Login for how to use VNC.

Set Login Mode

For the desktop graphical system, there are four login modes available:

1. Start the graphical interface and automatically log in.
2. Start the graphical interface and require manual login by the user.
3. Character terminal, automatically log in.
4. Character Terminal, User Manual Login

Open `RDK Configuration` through the menu bar. Select System Options -> Boot / Auto Login to enter the following configuration options. Select the corresponding options according to your needs.



It will take effect after restarting.

Server system, supporting four login modes:

1. Character Terminal, Automatic Login
2. Character Terminal, User Manual Login

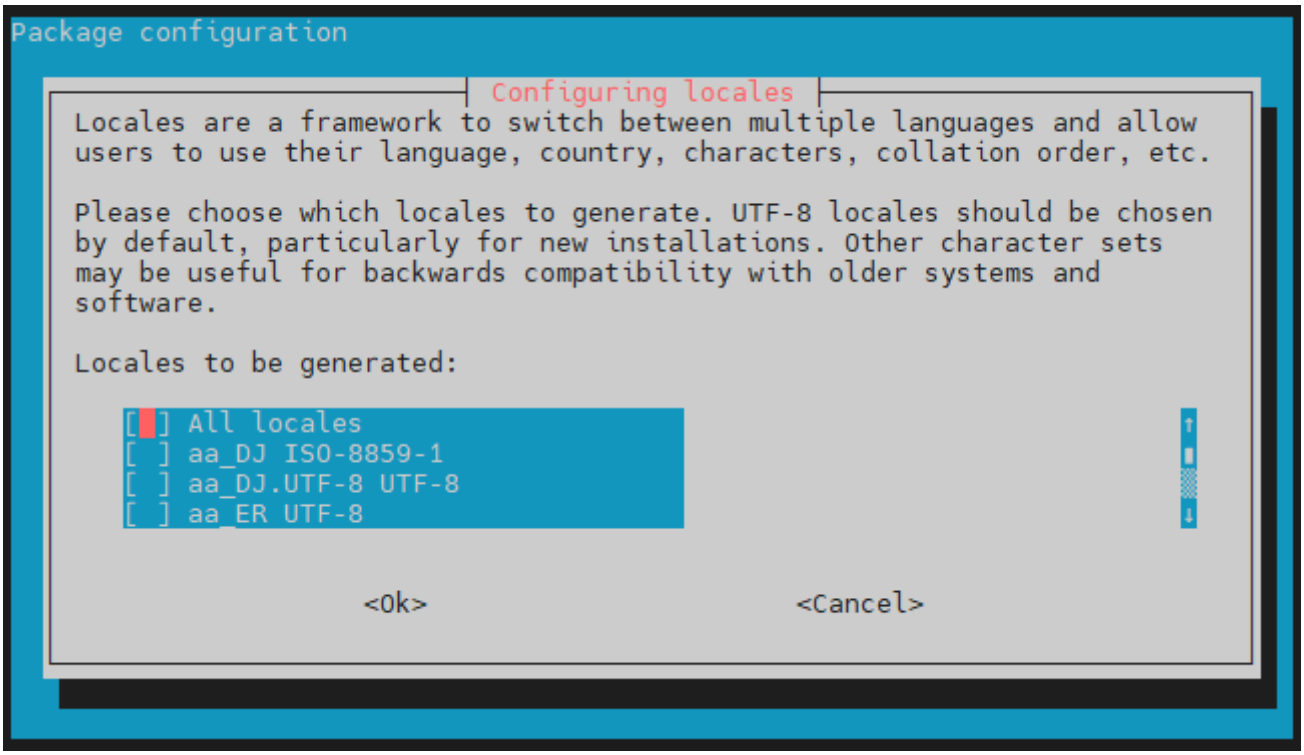
Execute the command `sudo srpi-config` to enter the configuration menu. Select System Options -> Boot / Auto Login to enter the following configuration options. Select the corresponding options according to your needs.

It will take effect after restarting.

Set up Chinese Environment

Open `RDK Configuration` through the menu bar. Select Localisation Options -> Locale to enter the following configuration.

Step 1: Select the language environment(s) you need (multiple choices), generally choose `en_US.UTF-8 UTF-8` and `zh_CN.UTF-8 UTF-8`. Press Enter to confirm and proceed to the next step.



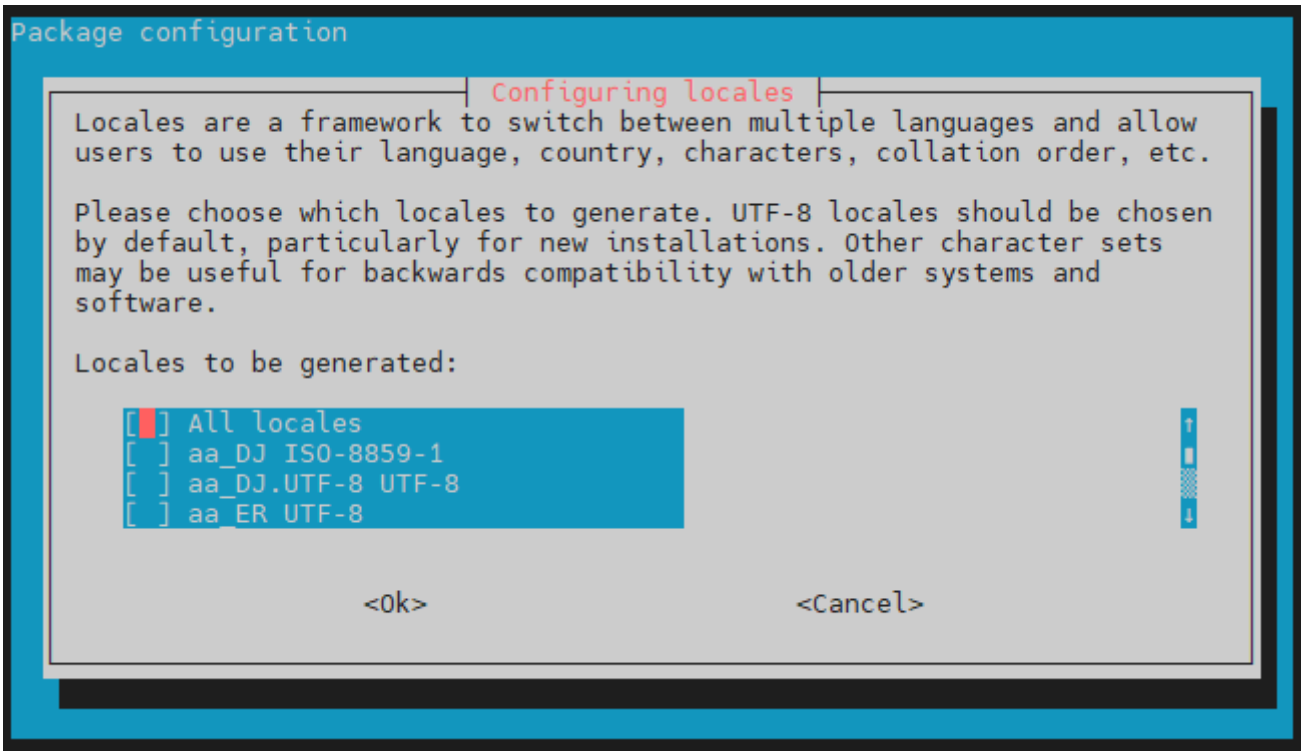
Step 2: Select the default language environment, choose `zh_CN.UTF-8 UTF-8` for Chinese environment. Press Enter and wait for a while to complete the configuration.

Step 3: Restart the machine to apply the latest configuration. `sudo reboot`

After booting, you will be prompted whether to update the names of several commonly used folders in the home directory.
It is recommended to choose `Don't ask me again` `Keep Old Name`, so that the directory names such as `Desktop` `Documents` `Downloads` under the user's working directory will not change with the language environment.

To enter the configuration menu, execute the command `sudo srpi-config`. Select the option "Localisation Options -> Locale" to enter the following configuration.

Step 1: Select the desired language environment(s) (multiple selection). Usually, selecting both `en_US.UTF-8 UTF-8` and `zh_CN.UTF-8 UTF-8` is sufficient. Press Enter to confirm and proceed to the next step.



Step 2: Select the default language environment. For the Chinese language environment, select `zh_CN.UTF-8 UTF-8`. Press Enter to confirm and wait for the configuration to complete.

Step 3: Restart the machine to apply the latest configuration. Execute `sudo reboot`.

1.4 Remote Login

This chapter is intended to introduce users who need to remotely access the development board through their personal computer (PC) on how to remotely login via serial port and network (VNC, SSH) methods.

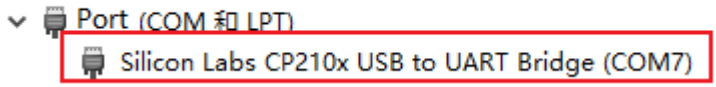
Before remote login via network, the development board needs to be connected to the network through wired Ethernet or wireless WiFi and configure the IP address of the development board. For the IP address information under both connection methods, refer to the following descriptions:

- Wired Ethernet: The development board defaults to static IP mode, with an IP address of `192.168.1.10`, subnet mask of `255.255.255.0`, and gateway of `192.168.1.1`.
- Wireless WiFi: The development board's IP address is generally assigned by the router and can be viewed in the device command line using the `ifconfig` command for the wlan0 network.

Serial Port Login {#login_uart}

Before using serial port login, it is necessary to confirm that the serial port cable of the development board is correctly connected to the computer. The connection method can refer to the Serial Port Debugging chapter. Serial port login requires a PC terminal tool. The commonly used tools are `Putty`, `MobaXterm`, etc. Users can choose according to their own habits. The port configuration process is similar for different tools. Below is an example with `MobaXterm` to explain the process of creating a new serial port connection:

- When the USB-to-Serial adapter is first plugged into the computer, the serial port driver needs to be installed. The driver can be obtained from the Tools sub-column of the Resource Center. After the driver is installed, the Device Manager can recognize the serial port board port normally, as shown in the figure below:



- Open the `MobaXterm` tool, click `Session`, and then select `Serial`.
- Configure the port number, for example, `COM3`. The actual serial port number used depends on the serial port number recognized by the PC.
- Set the serial port configuration parameters as follows:

Configuration Item	Parameter Value
Baud rate	921600
Data bits	8
Parity	None
Stop bits	1
Flow Control	None

- Click `OK`, enter the username: `root`, password: `root` to log in to the device

At this point, you can use the `ifconfig` command to query the IP address of the development board, where eth0 and wlan0 represent the wired and wireless networks respectively:

```
root@ubuntu:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast
192.168.1.255
        inet6 fe80::211:22ff:feaa:7637  prefixlen 64  scopeid 0x20<link>
        ether 00:11:22:aa:76:37  txqueuelen 1000  (Ethernet)
        RX packets 767  bytes 54006 (54.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5766  bytes 246466 (246.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 43  base 0xa000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 3847  bytes 339115 (339.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3847  bytes 339115 (339.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 08:e9:f6:ae:f8:8a  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Network Status Confirmation

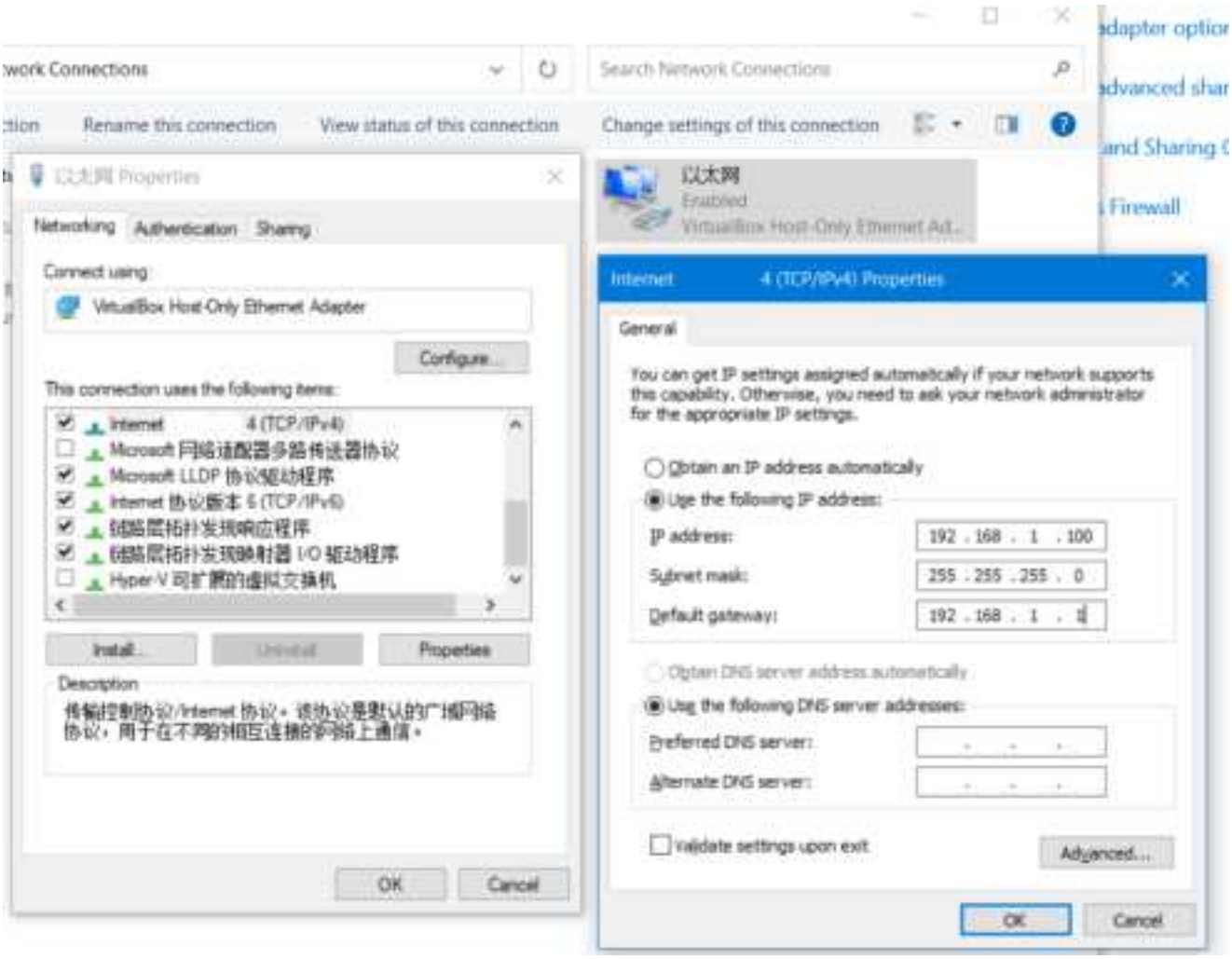
Before using remote login, it is necessary to ensure that the communication between the computer and the development board is normal. If it cannot be pinged, please follow the steps below to confirm:

- Confirm the IP address configuration of the development board and the computer. Generally, the first three segments need to be the same. For example, the development board: `192.168.1.10` and the computer: `192.168.1.100`.
- Confirm that the subnet mask and gateway configuration of the development board and the computer are consistent.
- Confirm whether the network firewall of the computer is turned off.

The wired Ethernet of the development board is set to use the static IP mode by default, and the IP address is `192.168.1.10`. For the case where the development board and the computer are directly connected to the network, only need to configure the computer as a static IP to ensure that it is in the same network segment as the development board. Taking the WIN10 system as an example, the method to modify the static IP of the computer is as follows:

- Find the corresponding Ethernet device in the network connection and double-click to open it.
- Double-click to open the Internet Protocol Version 4 option.

- Enter the corresponding network parameters in the red box in the figure below and click OK.



If you want to configure the wired network of the development board to obtain IP dynamically via DHCP mode, please refer to the Wired Network chapter for configuration.

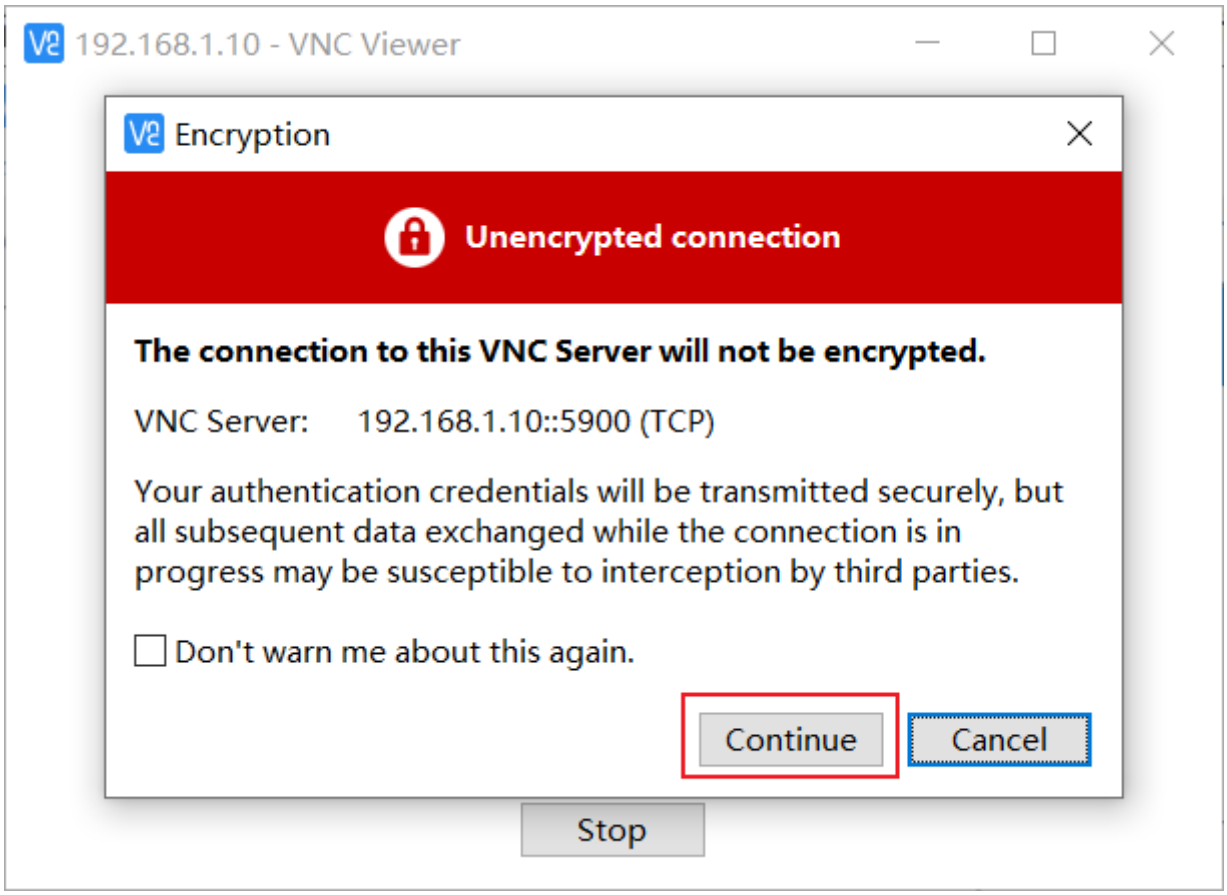
VNC Login

This section is for users using the Ubuntu Desktop system version, and it explains how to use "VNC Viewer" to achieve remote desktop login. "VNC Viewer" is a graphical desktop sharing software that allows you to remotely log in and control the desktop of the device on your computer. With this software, you can preview the system desktop of the development board on your computer screen and use your computer's mouse and keyboard for remote operation. By using VNC Viewer, you can achieve the same effect as local operation on the development board. You can download VNC Viewer from the following link: VNC Viewer.

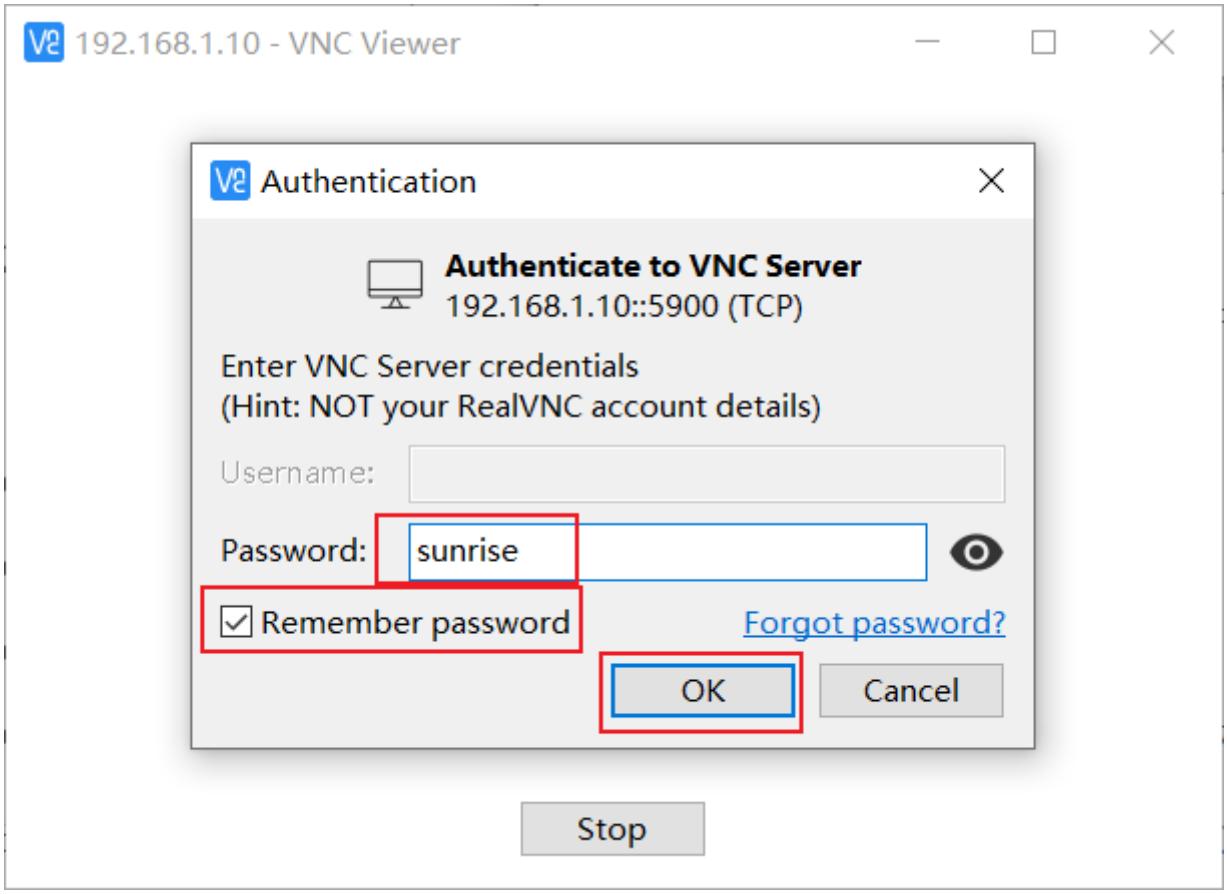
Connect to the Development Board Currently, VNC supports two connection methods: direct connection and cloud connection. Users can choose according to their own needs. This article recommends using the direct connection method. The connection steps are as follows:- Enter the IP

address of the input device, for example: 192.168.1.10

- After entering the IP address, press Enter, a prompt for an unencrypted connection will appear, click `Continue`



- Enter the password `sunrise`, check `Remember password`, and click `OK` to connect



SSH Login {#ssh}

In addition to VNC login for remote desktop, you can also connect to the development board via SSH. The following steps describe how to create SSH connections using terminal software and terminal command line methods.

Terminal Software

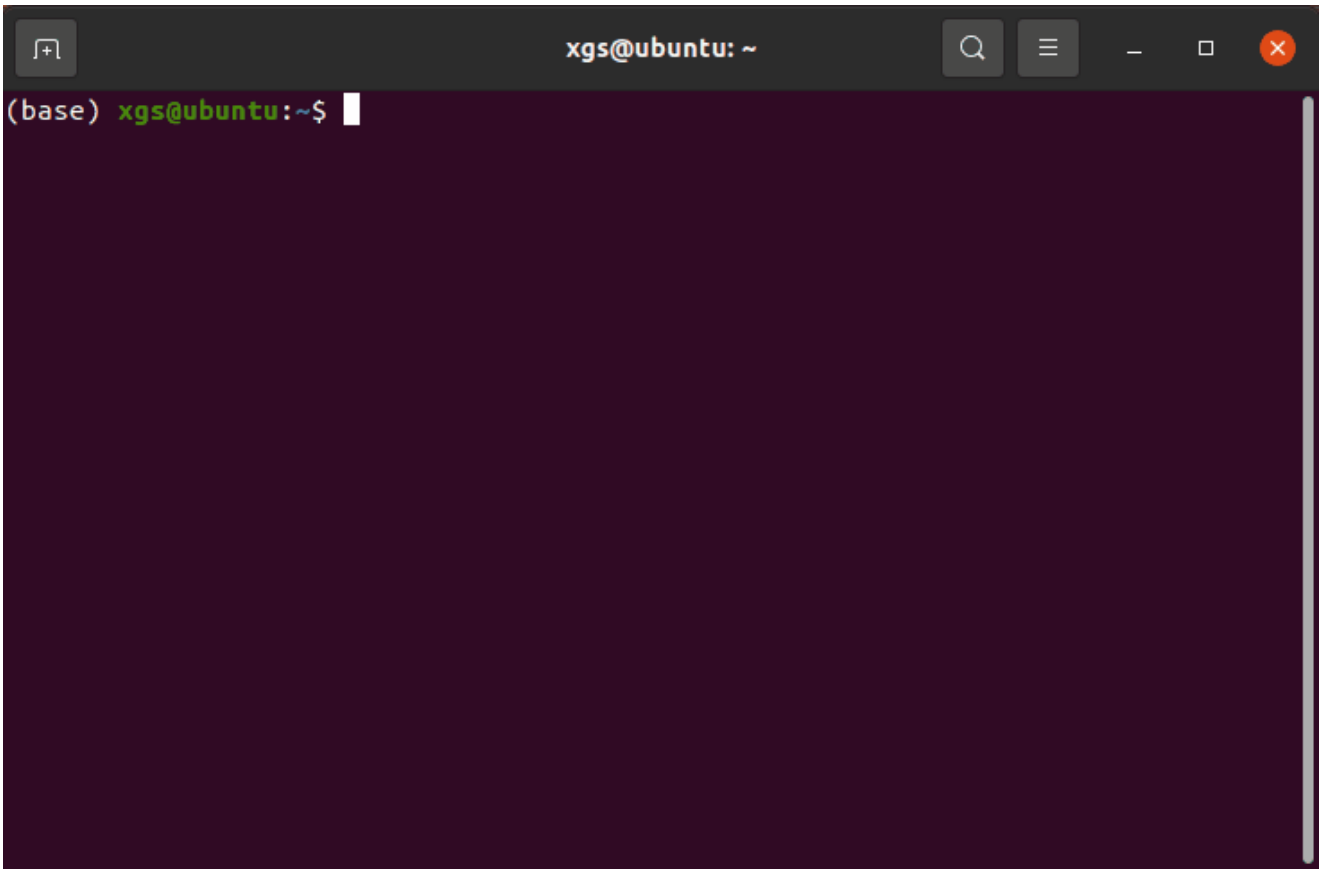
Commonly used terminal tools include `Putty`, `MobaXterm`, etc. Users can choose according to their own preferences. The configuration process for different tools is similar. The following example shows how to create a new SSH connection using `MobaXterm`:

1. Open the `MobaXterm` tool, click on `Session`, then select `SSH`.
2. Enter the development board IP address, for example: `192.168.1.10`.
3. Select `specify username`, enter `sunrise`.
4. After clicking OK, enter the username (sunrise) and password (sunrise) to complete the login.

Command Line on PC

Users can also use the command line to log in via SSH. The steps are as follows:

1. Open the terminal window and enter the SSH login command, for example: `ssh sunrise@192.168.1.10`.
2. A connection confirmation prompt will appear, enter YES.
3. Enter the password (sunrise) to complete the login.



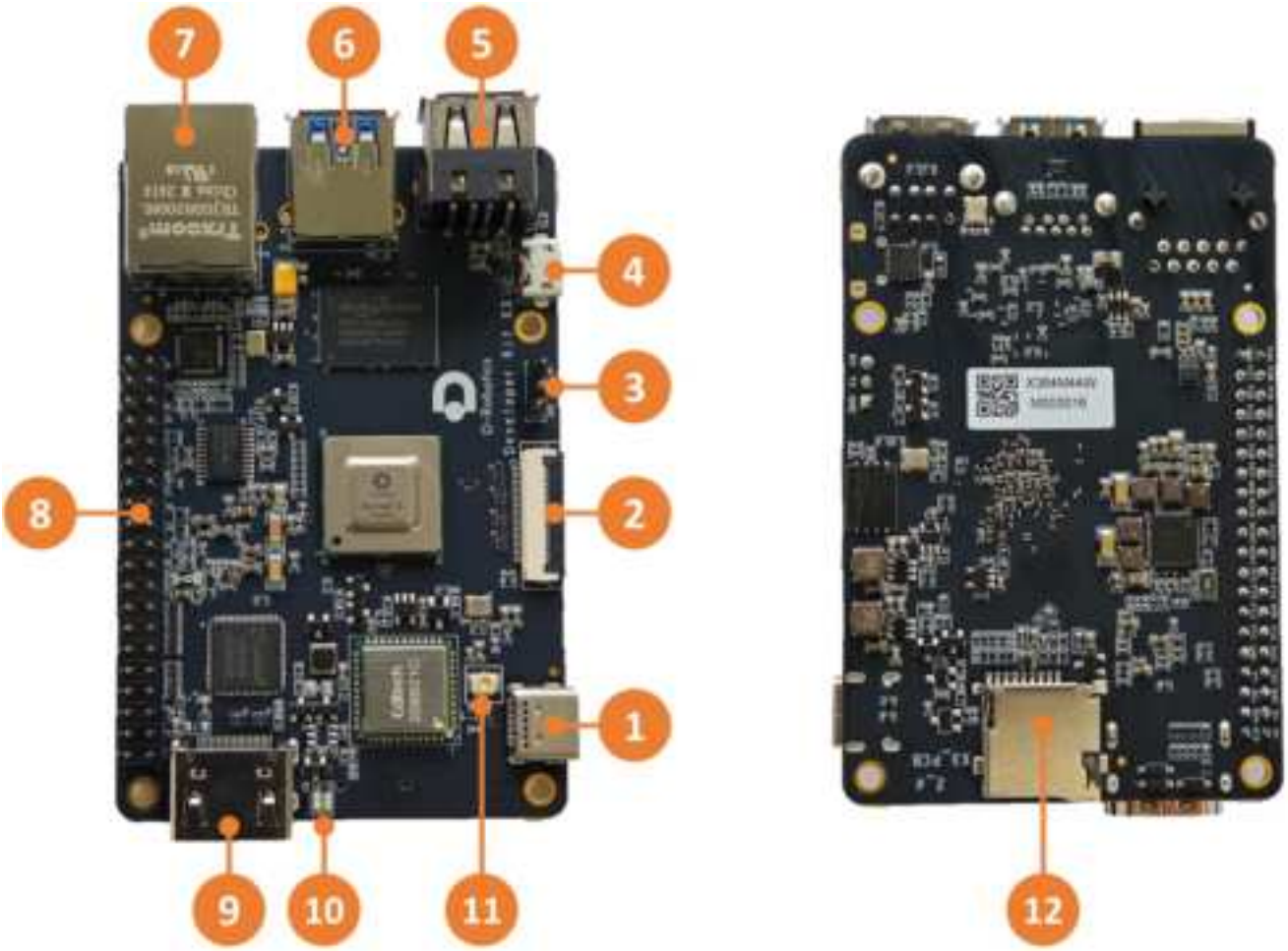
1.5 Hardware Interface Specification

```
import Tabs from '@theme/Tabs';
import TabItem from '@theme/TabItem';
```

Interface Overview

RDK X3 provides interfaces such as Ethernet port, USB, camera, LCD, HDMI, 40PIN, etc., which facilitate the development and testing of image multimedia and deep learning algorithms. The interface layout

of the development board is as follows:



No.	Function	No.	Function	No.	Function
1	Type C Power Supply	2	reserved interface	3	reserved interface
4	reserved interface	5	Two USB 2.0 Type A interfaces	6	USB 3.0 Type A interface
7	Gigabit Ethernet port	8	reserved interface	9	HDMI interface
10	reserved interface	11	reserved interface	12	reserved interface

Power interface

The development board provides one USB Type C interface (Interface 1) as the power supply interface, which requires the use of a power adapter supporting 5V/3A to power the development board. After connecting the power adapter to the development board, the development board's red power indicator light illuminates, indicating that the development board is powered properly.

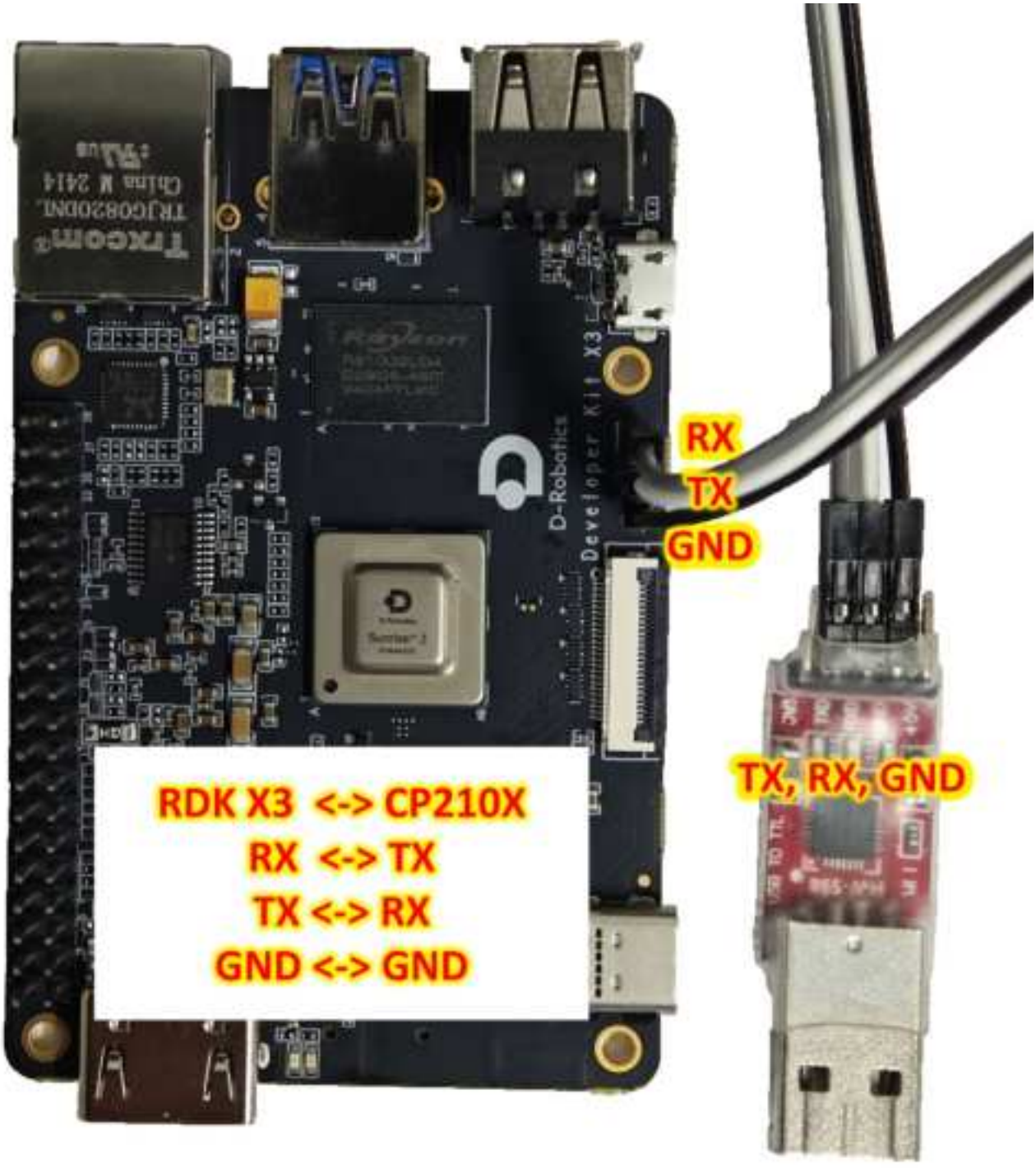
Do not use the USB interface of the computer to power the development board, as it may cause insufficient power supply and lead to abnormal power failure and repeated restarts of the development board.

Debug UART {#debug_uart}

The development board provides one debug UART (Interface 3) for serial port login and debugging functions. The parameter configuration of the computer serial port tool is as follows:

- Baud rate: 921600
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow Control: None

When connecting the UART, the DuPont wire needs to be connected to Interface 3 of the development board, and the USB serial adapter needs to be connected to the computer. After the connection is completed, it should look like the following figure:



The RDK X3 Module carrier board provides one debugging interface (Interface 15). Hardware-wise, the core module debugging serial port is converted into a USB interface via the `CH340` chip, allowing users to perform various debugging tasks using this interface. The parameters of the computer serial port tool should be configured as follows:

- Baud rate: 921600
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow Control: None

Typically, when users use this interface for the first time, they need to install the CH340 driver on the computer. Users can search for the keyword `CH340 serial port driver` for downloading and installation.

Ethernet Port

The development board provides one Gigabit Ethernet interface (Interface 7), supporting the 1000BASE-T and 100BASE-T standards. It defaults to static IP mode with the IP address `192.168.1.10`. To confirm the IP address of the development board, you can log in to the device via serial port and use the `ifconfig` command to view the configuration of the `eth0` interface:

```
sunrise@ubuntu:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast
192.168.1.255
        inet6 fe80::211:22ff:fe6f:de17  prefixlen 64  scopeid 0x20<link>
        ether 00:11:22:6f:de:17  txqueuelen 1000  (Ethernet)
        RX packets 112  bytes 7327 (7.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 197  bytes 8678 (8.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 39  base 0xc000
```

If the system displays the following printout after inserting the Ethernet cable into the development board, it indicates that the Ethernet connection is normal:

```
[13293.952661] st_gmac a5014000.ethernet eth0: Link is Down
[13295.062996] st_gmac a5014000.ethernet: Link is Up - 1000/Full
[13296.000750] st_gmac a5014000.ethernet eth0: Link is Up - 1Gbps/Full -
flow control rx/tx
```

The development board provides one Gigabit Ethernet interface (Interface 6), supporting the 1000BASE-T and 100BASE-T standards. It defaults to static IP mode with the IP address `192.168.1.10`. To confirm the IP address of the development board, you can log in to the device via serial port and use the `ifconfig` command to view the configuration of the `eth0` interface:

```
sunrise@ubuntu:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast
192.168.1.255
        inet6 fe80::211:22ff:fe6f:de17  prefixlen 64  scopeid 0x20<link>
        ether 00:11:22:6f:de:17  txqueuelen 1000  (Ethernet)
        RX packets 112  bytes 7327 (7.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 197  bytes 8678 (8.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 39  base 0xc000
```

If the system displays the following printout after inserting the Ethernet cable into the development board, it indicates that the Ethernet connection is normal:

```
[13293.952661] st_gmac a5014000.ethernet eth0: Link is Down
[13295.062996] st_gmac a5014000.ethernet: Link is Up - 1000/Full
[13296.000750] st_gmac a5014000.ethernet eth0: Link is Up - 1Gbps/Full -
flow control rx/tx
```

HDMI Interface {#hdmi_interface}

The development board provides one HDMI display interface (Interface 9), supporting a maximum resolution of 1080P. The development board outputs the Ubuntu system desktop (Ubuntu Server version displaying logo icons) on the monitor via the HDMI interface. Additionally, the HDMI interface supports real-time display of camera and network stream images.

The currently supported display resolutions for the HDMI interface are as follows:

- 1920x1080
- 1280x720
- 1024x600
- 800x480

USB Interface

Due to the X3 chip providing only one USB interface, the development board implements multiple USB interface extensions through hardware circuits to meet the needs of multiple USB device connections. The interface descriptions are as follows:

Interface Type	Interface Number	Quantity	Interface Description
Micro USB 2.0	Interface 4	1	USB Device mode, used for connecting to the host to implement functions such as ADB, Fastboot, UVC, etc.

Interface Type	Interface Number	Quantity	Interface Description
USB 2.0 Type A	Interface 5	2	USB Host mode, used for connecting USB 2.0 peripherals
USB 3.0 Type A	Interface 6	1	USB Host mode, used for connecting USB 3.0 peripherals

USB host and device mode switching is fully implemented by hardware circuits. Users only need to connect devices according to the logic in the table above.

The development board's USB Host and Device functions are mutually exclusive. When a device is connected to the Device interface, the Host interface will automatically become inactive.

Using a USB Flash Drive

```
sunrise@ubuntu:/media/sda1$ sudo touch test
[sudo] password for sunrise:
sunrise@ubuntu:/media/sda1$ ls -l test
-rwxr-xr-x 1 root root 0 Apr 30 21:19 test
sunrise@ubuntu:/media/sda1$
```

Using a USB Serial Port Adapter

The USB Type A interfaces (Interfaces 5 and 6) on the development board support USB serial port adapter functionality. They can automatically detect the USB serial port adapter and create device nodes `/dev/ttyUSB*` or `/dev/ttyACM*` (the asterisk represents a number starting from 0). Users can refer to the UART Usage section for instructions on using the serial port.

The RDK X3 core module only supports one USB3.0 interface. Therefore, the carrier board expands to 4 USB3.0 host interfaces and 1 Micro USB2.0 device interface through peripheral circuits and USB hubs, meeting the diverse USB interface requirements of users. The interface descriptions are as follows:

Interface Type	Interface Number	Quantity	Description
USB3.0 Type A Host	Interface 3	4	Used for USB peripherals
Micro USB2.0 Device	Interface 7	1	Used for adb debugging and fastboot flashing

The USB host/device mode switch is completely implemented by the hardware circuit. Users only need to connect the devices according to the logic in the above table.

The USB host and device functions on the development board are mutually exclusive. After connecting a device to the device interface, the host interface will be automatically disabled.

USB Camera

The development board's USB Type A interface supports USB camera functionality and can automatically detect the connection of a USB camera and create a device node `/dev/video8`. Users can use the `v4l2` command to confirm if the USB camera is working properly, as shown below:

```
sunrise@ubuntu:/media/sda1$ sudo v4l2-ctl -d /dev/video8 --all
Driver Info:
        Driver name      : uvcvideo
        Card type        : FHD Camera: FHD Camera
        Bus info         : usb-xhci-hcd.0.auto-1.2
        Driver version   : 4.14.87
        Capabilities     : 0x84200001
                Video Capture
                Streaming
                Extended Pix Format
                Device Capabilities
        Device Caps      : 0x04200001
                Video Capture
                Streaming
                Extended Pix Format
Media Driver Info:
```Driver name: uvcvideo
Model: FHD Camera: FHD Camera
Serial: 12345
Bus info: 1.2
Media version: 4.14.87
Hardware revision: 0x00000020 (32)
Driver version: 4.14.87

Interface Info:
ID: 0x03000002
Type: V4L Video

Entity Info:
ID: 0x00000001 (1)
Name: FHD Camera: FHD Camera
Function: V4L2 I/O
Pad 0x01000004: Sink
Link 0x0200000a: from remote pad 0x1000007 of entity 'Processing 2':
Data, Enabled, Immutable

Priority: 2
Video input: 0 (Camera 1: ok)

Format Video Capture:
Width/Height: 1920/1080
Pixel Format: 'MJPG' (Motion-JPEG)
Field: None
Bytes per Line: 0
Size Image: 4147200
```
```

```
Colorspace: Default
Transfer Function: Default (maps to Rec. 709)
YCbCr/HSV Encoding: Default (maps to ITU-R 601)
Quantization: Default (maps to Full Range)
Flags:

Crop Capability Video Capture:
Bounds: Left 0, Top 0, Width 1920, Height 1080
Default: Left 0, Top 0, Width 1920, Height 1080
Pixel Aspect: 1/1

Selection Video Capture: crop_default, Left 0, Top 0, Width 1920, Height
1080, Flags:
Selection Video Capture: crop_bounds, Left 0, Top 0, Width 1920, Height
1080, Flags:

Streaming Parameters Video Capture:
Capabilities: timeperframe
Frames per second: 30.000 (30/1)
Read buffers: 0
```

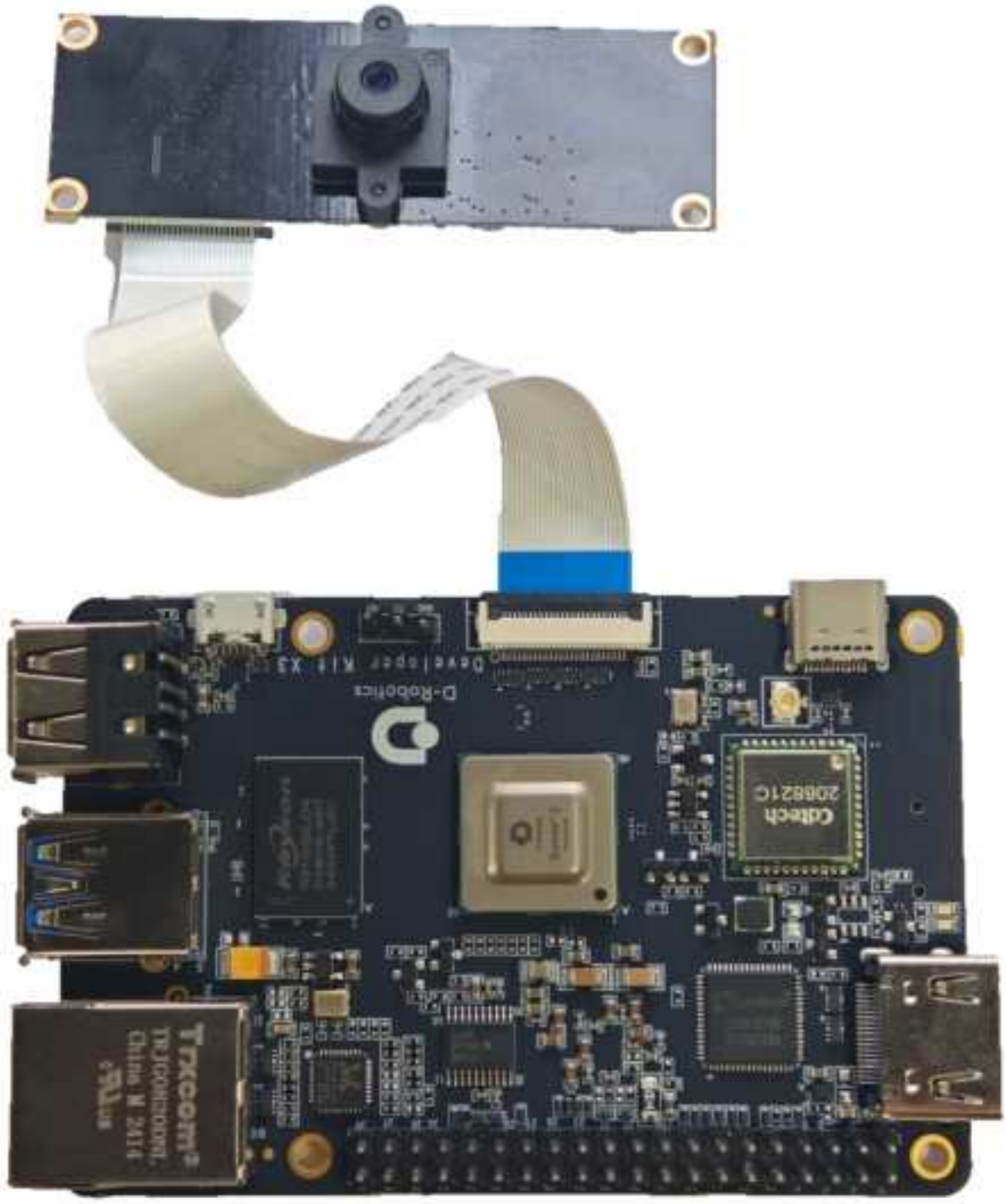
MIPI CSI {#mipi_port}

The development board provides 1 MIPI CSI interface (Interface 2), which allows for the connection of MIPI cameras. Currently, the development board is compatible with multiple specifications of camera modules. The module models and specifications are as follows:

No.	Sensor	Resolution	FOV	I2C Device Address
1	GC4663	400W	H:104 V:70 D:113	0x29
2	JXF37	200W	H:62 V:37 D:68	0x40
3	IMX219	800W	H:62 V:37 D:68	0x10
4	IMX477	1200W	H:62 V:37 D:68	0x1a
5	OV5647	500W	H:62 V:37 D:68	0x36

The camera module is connected to the development board through a FPC cable. Please note that the blue side of the cable should face up when inserting it into the connector.

Taking the JXF37 camera module as an example, after installation, it should look like the following picture:



After installation, users can use the i2cdetect command to verify if the I2C address of the module can be detected correctly. If it cannot be detected, users need to check if the FPC cable connection is correct. When the I2C address is successfully detected, the log will print as follows:

```
# First, enable the 24MHz master clock for the sensor
sunrise@ubuntu:~# sudo bash -c "echo 1 >
/sys/class/vps/mipi_host0/param/snrclk_en"
sunrise@ubuntu:~# sudo bash -c "echo 24000000 >
/sys/class/vps/mipi_host0/param/snrclk_freq"
# Execute the i2cdetect command, and the displayed 40 is the I2C device
address of the JXF37 sensor, indicating that the camera is connected
correctly
sunrise@ubuntu:~# sudo i2cdetect -y -r 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

The purchase method for the above Camera module can be referred to on the purchase link.

The RDK X3 Module carrier board provides three sets of MIPI CSI interfaces for CAM 0/1/2, allowing for the simultaneous connection of three camera modules to meet the diverse needs of users. The specific descriptions are as follows:

1. CAM 0/2 (Interfaces 16/17), using 15-pin FPC connectors, can directly connect to various camera modules such as Raspberry Pi's OV5647, IMX219, IMX477, etc.
2. CAM 1 (Interface 11), using a 24-pin FPC connector, supports various camera modules such as F37, GC4663, IMX415, etc.

The basic specifications of the camera modules are as follows:

Number	Sensor	Resolution	FOV	I2C Device Address
1	GC4663	4MP	H:104 V:70 D:113	0x29
2	JXF37	2MP	H:62 V:37 D:68	0x40
3	IMX219	8MP	H:62 V:37 D:68	0x10
4	IMX477	12MP	H:62 V:37 D:68	0x1a
5	OV5647	5MP	H:62 V:37 D:68	0x36

The purchase links for the above camera modules can be found here.

Important: It is strictly prohibited to plug or unplug the camera while the development board is powered on, as it may damage the camera module.

Micro SD Interface

The development board provides 1 Micro SD card interface (Interface 12). It is recommended to use a storage card with a capacity of at least 8GB to meet the installation requirements of the Ubuntu operating system and related packages.

Hot-plugging the TF storage card is prohibited during the use of the development board, as it may cause abnormal system operation or even damage to the storage card file system.

Wi-Fi Antenna Interface

The development board supports both onboard and external antennas for wireless networking. In most cases, the onboard antenna can meet the user's needs. However, when the development board is installed with a metal casing, an external antenna needs to be connected to Interface 11 to enhance signal strength.

To convert the onboard antenna to an external antenna, use the following command: sed -i 's/trace/cable/g' /etc/init.d/hobot-wifi. Restart the system for the changes to take effect.
To revert to the onboard antenna, use the following command: sed -i 's/cable/trace/g' /etc/init.d/hobot-wifi. Restart the system for the changes to take effect.

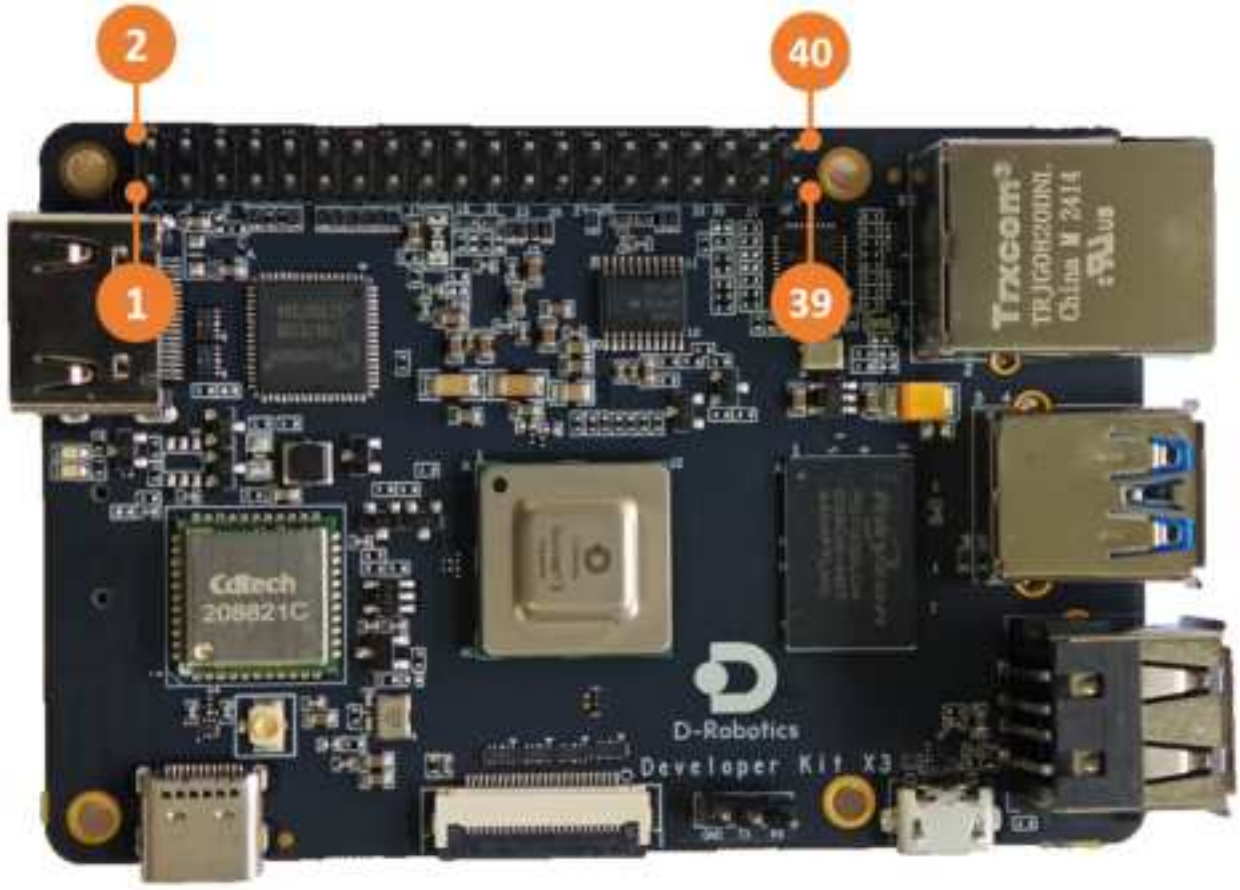
40-pin header interface

The RDK X3 development board provides a 40-pin header interface with IO signals using a 3.3V logic level design. The pin definitions are compatible with products such as Raspberry Pi. The detailed pin definitions and multiplex relationships are as follows:

RDK X3 40Pin Pinout Diagram

Function Description	X3 Pin Number	BCM Code	CVM Function Name	Physical Pin BOARD Code		CVM Function Name	BCM Code	X3 Pin Number	Function Description
3.3V Power Signal			VDD_3V3	1	2	VDD_5V			5V Power Signal
I2C0 Data Signal	9	2	I2C0_SDA	3	4	VDD_5V			5V Power Signal
I2C0 Clock Signal	8	3	I2C0_SCL	5	6	GND			GND Signal
I2S0 MCLK Clock Signal	101	4	I2S0_MCLK	7	8	UART_TXD	14	111	UART3 Transmit Signal
GND Signal			GND	9	10	UART_RXD	15	112	UART3 Receive Signal
GPIO17 Signal	12	17	GPIO17	11	12	I2S0_BCLK	18	102	I2S0 BCLK Clock Signal
GPIO27 Signal	13	27	GPIO27	13	14	GND			GND Signal
GPIO22 Signal	30	22	GPIO22	15	16	GPIO23	23	27	GPIO23 Signal
3.3V Power Signal			VDD_3V3	17	18	GPIO24	24	22	GPIO24 Signal
SPI1 MOSI Signal	6	10	SPI1_MOSI	19	20	GND			GND Signal
SPI1 MISO Signal	7	9	SPI1_MISO	21	22	GPIO25	25	29	GPIO25 Signal
SPI1 CLK Signal	3	11	SPI1_SCLK	23	24	SPI1_CSN	8	5	SPI1 CS Signal
GND Signal			GND	25	26	GPIO7	7	28	GPIO7 Signal
I2C3 Data Signal	15	0	I2C3_SDA	27	28	I2C3_SCL	1	14	I2C3 Clock Signal
GPIO5 Signal	119	5	GPIO5	29	30	GND			GND Signal
GPIO6 Signal	118	6	GPIO6	31	32	PWM4	12	25	PWM4 Signal
PWM0 Signal	4	13	PWM0	33	34	GND			GND Signal
I2S0 LRCK Signal	103	19	I2S0_LRCK	35	36	GPIO16	16	20	GPIO16 Signal
GPIO26 Signal	117	26	GPIO26	37	38	I2S1_SDIO	20	108	I2S1_SDIO Signal
GND Signal			GND	39	40	I2S0_SDIO	21	104	I2S0_SDIO Signal

The development board has silk screen marking for the 40-pin interface, making it convenient for users to operate. The positions of PIN1 and PIN40 are as follows:



For the usage of each function of the 40-pin interface, please refer to the 40PIN Function Usage section.

The RDK X3 Module carrier board provides a set of 40-pin header interfaces (Interface 9), and the interface voltage level is determined by the IO level switching header (Interface 14), supporting both 1.8V and 3.3V modes. The pin definitions are compatible with products such as Raspberry Pi. The detailed pin definitions and multiplex relationships are as follows:

By default, the RDK X3 Module core module firmware and carrier board voltage configuration are set to 3.3V. If you need to switch the IO level, please refer to the IO Level Selection Header Interface.

IO Level Switching Interface

This interface is not available.

It is important to note that **this interface cannot be left unconnected, otherwise the core module will not be able to start up due to power supply abnormalities**.

By default, the firmware of the core module of RDK X3 Module and the level configuration of the carrier board are set to 3.3V. If you need to switch the IO level, please follow the steps below:

1. Download the boot firmware that supports the 1.8V level configuration from the firmware download address.
2. Use the official programming tool `hbupdate` to update the boot firmware of the core board. For detailed instructions, please refer to Image Flash.
3. Power off the device, short connect the `vref` and `1.8V` signals using a jumper cap, and then power on again.

Bluetooth

Initialization

The Bluetooth function on the development board is not enabled by default. You need to execute the `/usr/bin/startbt6212.sh` script to initialize it. The script completes the following tasks:

- Reset the Bluetooth
- Create the `messagebus` user and group, which are required for the operation of the `dbus-daemon` program
- Run `brcm_patchram_plus` to load the Bluetooth driver and firmware
- Continuously check if the `/sys/class/bluetooth/hci0` directory exists to confirm that the Bluetooth driver is running properly
- The appearance of **Done setting line discpline** indicates that the Bluetooth has been successfully enabled
- Execute `hciconfig hci0 up` to bring up the Bluetooth Link
- Execute `hciconfig hci0 piscan` to perform Bluetooth scanning (this step can be excluded depending on the situation)

The log after the successful execution of the script is as follows:

```
root@ubuntu:~# sudo startbt6212.sh
Waiting for bluetooth initialize.....Done setting line discpline
.Done
Check Bluetooth State...unblocked
Set Bluetooth Up...
Set Bluetooth piscan...
hci0: Type: Primary  Bus: UART
 BD Address: 08:E9:F6:AE:F8:8B  ACL MTU: 1021:8  SCO MTU: 64:1
 UP RUNNING PSCAN ISCAN
 RX bytes:759 acl:0 sco:0 events:48 errors:0
 TX bytes:2522 acl:0 sco:0 commands:48 errors:0

root@ubuntu:~#
```

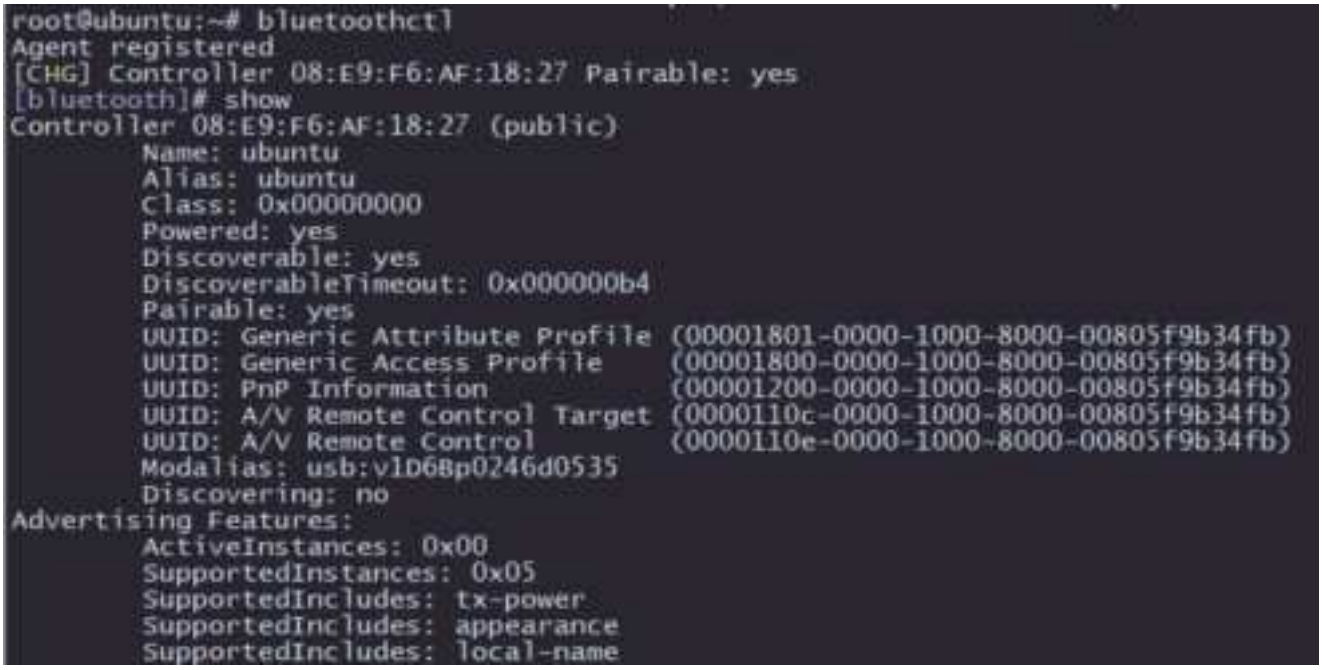
In addition, users can use the following command to check if the Bluetooth process is functioning properly:

```
ps ax | grep "/usr/bin/dbus-daemon\|/usr/lib/bluetooth/bluetoothd"
/usr/bin/dbus-daemon

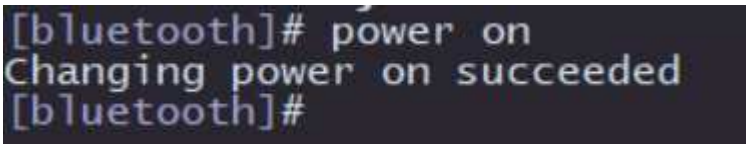
/usr/lib/bluetooth/bluetoothd
```

Network Configuration and Connection

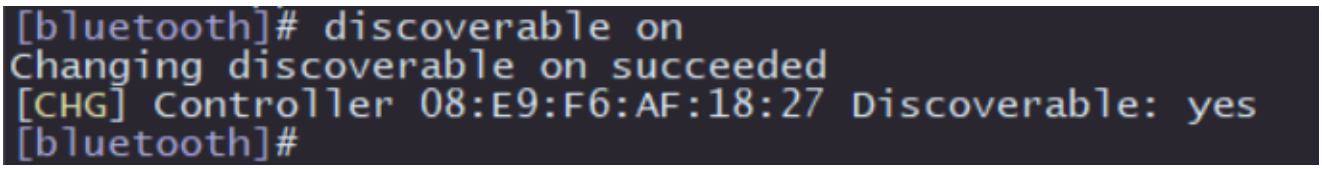
Execute `sudo bluetoothctl` to enter the interactive mode of Bluetooth configuration. If device information similar to the image below appears, it means that the Bluetooth has been recognized. Then, use `show` to view the Bluetooth information and pay attention to the `powered` and `discoverable` statuses of the Bluetooth.



Execute `power on` to enable the Bluetooth, as shown in the image below:



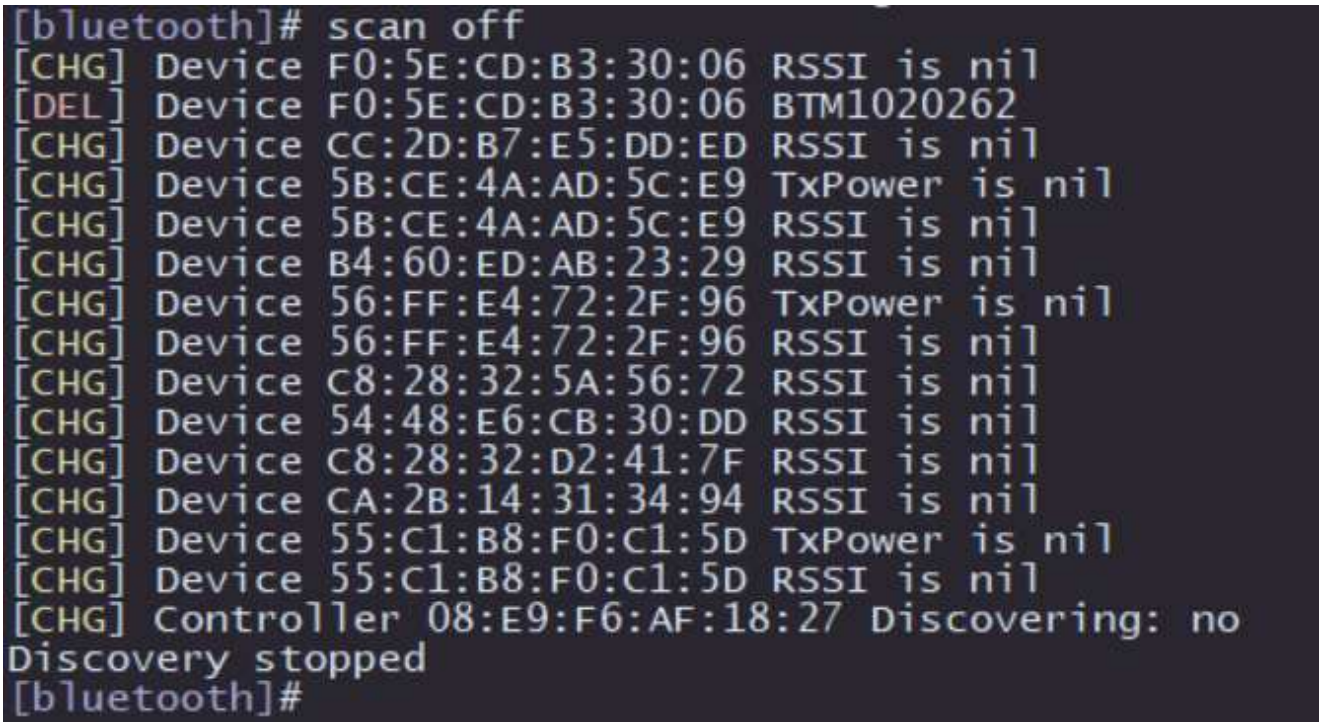
In order to make the Bluetooth discoverable to nearby devices, execute `discoverable on` to enable the Bluetooth and open the discoverable attribute of the Bluetooth, as shown in the image below:



Now, you can use a mobile phone or computer to scan for the Bluetooth device with the name `ubuntu`, as shown in the image below:


Settings
Bluetooth
Bluetooth
This iPhone is discoverable as "iPhone" while Bluetooth Settings is open.
DEVICES
ubuntu
To pair an Apple Watch with your iPhone, go to the Apple Watch app.

Next, test the active scanning function of Bluetooth. In the interactive interface of `bluetoothctl`, enter `scan on` to enable active scanning. It will periodically print nearby devices. You can see that my mobile phone device has been discovered. Enter `scan off` to disable the scanning function and summarize the scanned Bluetooth devices.



Then comes the pairing with other Bluetooth devices:

- Pairing command: `pair [targetMAC]`, after entering this command, follow the prompts to enter `yes`, and the remote Bluetooth device will select the `Pair` option to complete the pairing.
- After successful pairing, you can use `trust [targetMAC]` to automatically connect next time.

```
[bluetooth]# pair CC:2D:B7:E5:DD:ED
Attempting to pair with CC:2D:B7:E5:DD:ED
[CHG] Device CC:2D:B7:E5:DD:ED Connected: yes
Request confirmation
[agent] Confirm passkey 125453 (yes/no): yes
[CHG] Device CC:2D:B7:E5:DD:ED Modalias: bluetooth:v004Cp7109d0E70
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 00000000-deca-fade-deca-deafdecacafe
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 00001000-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device CC:2D:B7:E5:DD:ED UUIDs: 02030302-1d19-415f-86f2-22a2106a0a77
[CHG] Device CC:2D:B7:E5:DD:ED ServicesResolved: yes
[CHG] Device CC:2D:B7:E5:DD:ED Paired: yes
Pairing successful
[CHG] Device CC:2D:B7:E5:DD:ED ServicesResolved: no
[CHG] Device CC:2D:B7:E5:DD:ED Connected: no
[bluetooth]# trust  CC:2D:B7:E5:DD:ED
[CHG] Device CC:2D:B7:E5:DD:ED Trusted: yes
Changing CC:2D:B7:E5:DD:ED trust succeeded
[bluetooth]#
```

After the above operations, the basic functions of Bluetooth scanning and pairing are completed. For more functions, please refer to the official help documentation of `BlueZ`.

Information on Disposal for Users of Waste Electrical & Electronic Equipment

This symbol on the product or on its packaging indicates that used electrical and electronic products should not be mixed with unsorted municipal waste. For proper treatment, it is your responsibility to dispose of your waste equipment by arranging to return it to designated collection points.

Disposing of this product correctly will help save valuable resources and prevent any potential negative effects

on human health and the environment, which could otherwise arise from inappropriate waste handling.

To return your used device, please use the return and collection systems or contact the retailer where the product was purchased, which is free of charge, please contact your local authority for further details of your nearest designated collection point.

Penalties may be applicable for incorrect disposal of this waste, in accordance with your national legislation.

FCC COMPLIANCE NOTICE

This equipment must be installed and operated in accordance with provided instructions and the antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter. End-users and installers must be provided with antenna installation instructions and transmitter operating conditions for satisfying RF exposure compliance.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation.

Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

– Reorient or relocate the receiving antenna.

– Increase the separation between the equipment and receiver.

– Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.

– Consult the dealer or an experienced radio/TV technician for help.

Integration instructions for host product manufacturers according to KDB 996369 D03 OEM Manual v01

For RDK X3，FCC ID: 2BGUG-RDKX3K

2.2 List of applicable FCC rules

CFR 47 FCC PART 15 SUBPART C&E has been investigated. It is applicable to the modular.

2.3 Specific operational use conditions

This module is stand-alone modular. If the end product will involve the Multiple simultaneously transmitting condition or different operational conditions for a stand-alone modular transmitter in a host, host manufacturer have to consult with module manufacturer for the installation method in end system.

2.4 Limited module procedures

Not applicable

2.5 Trace antenna designs

Not applicable

2.6 RF exposure considerations

To maintain compliance with FCC's RF Exposure guidelines, this equipment should be installed and operated with minimum distance of 20cm from your body.

2.7 Antennas

This radio transmitter FCC ID: 2BGUG-RDKX3K has been approved by Federal Communications Commission to operate with the antenna types listed below, with the maximum permissible gain indicated. Antenna types not included in this list that have a gain greater than the maximum gain indicated for any type listed are strictly prohibited for use with this device.

maximum Gain / Antenna type	2.4G	5.2G	5.8G
PCB antenna	2.27dBi	2.25dBi	2.24dBi
Dipole antenna	2.76dBi	2.33dBi	1.56dBi

2.8 Label and compliance information

The final end product must be labeled in a visible area with the following " Contains FCC ID: 2BGUG-RDKX3K"

2.9 Information on test modes and additional testing requirements

Host manufacturer is strongly recommended to confirm compliance with FCC requirements for the transmitter when the module is installed in the host.

2.10 Additional testing, Part 15 Subpart B disclaimer

Host manufacturer is responsible for compliance of the host system with module installed with all other applicable requirements for the system such as Part 15 B.

Integration instructions for host product manufacturers according to KDB 996369 D03 OEM Manual v01

2.2 List of applicable FCC rules

CFR 47 FCC PART 15 SUBPART C&E has been investigated. It is applicable to the modular.

2.3 Specific operational use conditions

This module is stand-alone modular. If the end product will involve the Multiple simultaneously transmitting condition or different operational conditions for a stand-alone modular transmitter in a host, host manufacturer have to consult with module manufacturer for the installation method in end system.

2.4 Limited module procedures

Not applicable

2.5 Trace antenna designs

Not applicable

2.6 RF exposure considerations

To maintain compliance with FCC's RF Exposure guidelines, this equipment should be installed and operated with minimum distance of 20cm from your body.

2.7 Antennas

This radio transmitter FCC ID: 2BGUG-RDKX3M has been approved by Federal Communications Commission to operate with the antenna types listed below, with the maximum permissible gain indicated. Antenna types not included in this list that have a gain greater than the maximum gain indicated for any type listed are strictly prohibited for use with this device.

Antenna type	Maximum Antenna gain
Dipole Antenna	2.76 dBi

2.8 Label and compliance information

The final end product must be labeled in a visible area with the following " Contains FCC ID: 2BGUG-RDKX3M"

2.9 Information on test modes and additional testing requirements

Host manufacturer is strongly recommended to confirm compliance with FCC requirements for the transmitter when the module is installed in the host.

2.10 Additional testing, Part 15 Subpart B disclaimer

Host manufacturer is responsible for compliance of the host system with module installed with all other applicable requirements for the system such as Part 15 B.