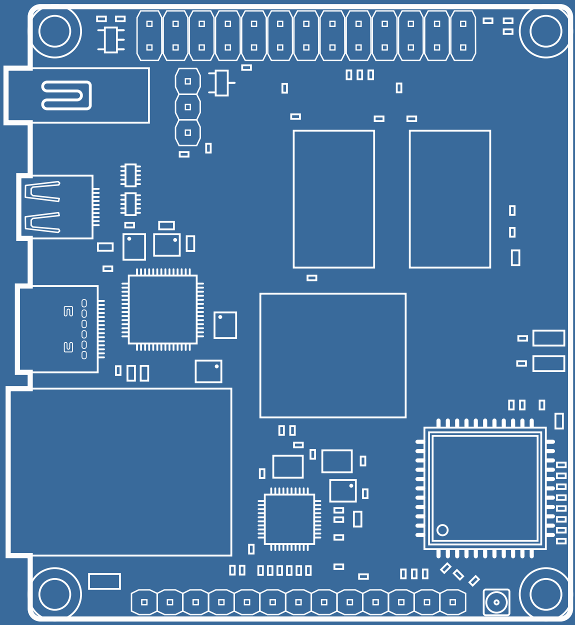
1st Edition

A USER GUIDE FOR **ORANGE PI ZERO 2**

THE OFFICIAL Orange Pi User's Guide



Orange Pi

THE OFFICIAL Orange Pi User's Guide

Contents

APPENDICES

Chapter 1

Getting familiar with your Orange Pi

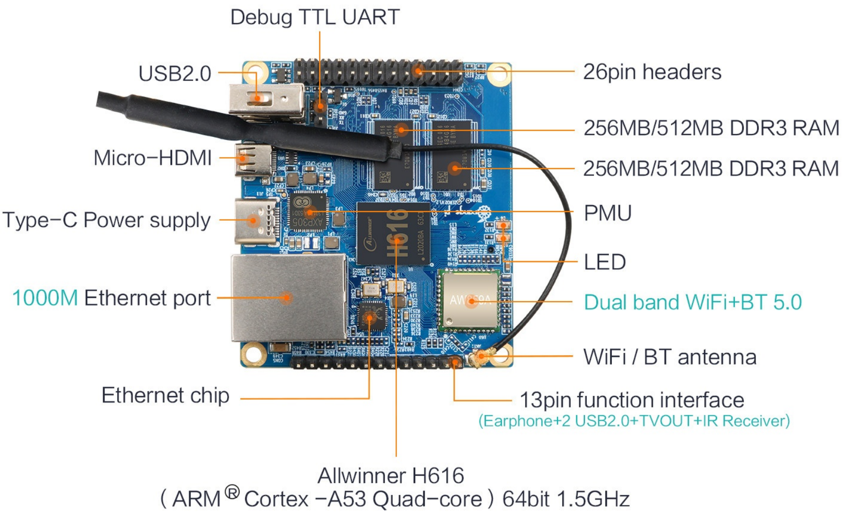
Know your new pocket-sized Orange Pi computer by taking a walkthrough over its various components and features.

Orange Pi is an open-source single board computer, a new generation of arm64 development board, which can run systems such as Android 10, Ubuntu and Debian and so on. Orange Pi Zero 2 uses the Allwinner H616 system-on-chip and has 512MB/1GB DDR3 memory.

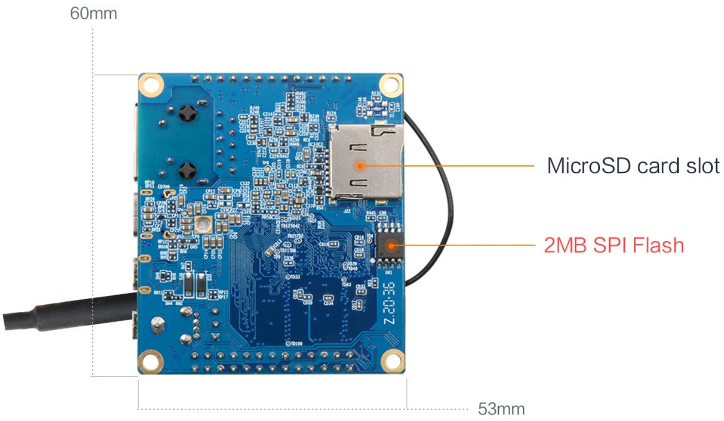
Orange Pi Zero 2 is for anyone who wants to start creating with technology not just consuming it. It's a simple, fun, useful tool that you can use to start taking control of the world around you.

Hardware Overview

Top view



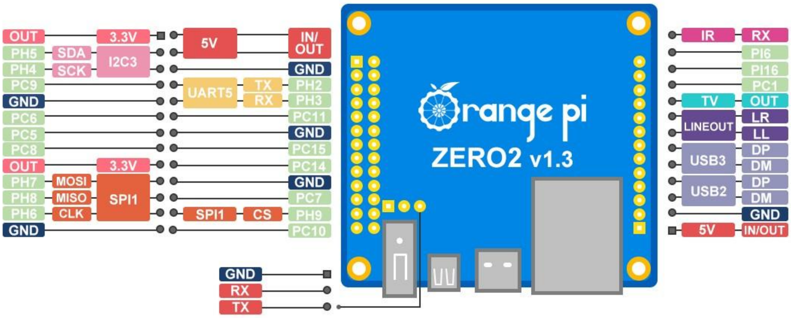
Bottm view



Hardware Specification

- Allwinner H616, Quad core Cortex-A53 64-bit SoC @ 1.5GHz
- 512MB or 1GB DDR3 SDRAM (depending on model)
- H.265 (4kp60 decode), H264 (4kp30 decode, 4kp25 encode or 1080p60 encode)
- ARM G31 GPU Supports OpenGL ES 3.2/2.0/1.0, Vulkan 1.1 OpenCL 2.0
- AW859A module Support IEEE 802.11 a/b/g/n/ac wireless, Bluetooth 5.0
- 1000M/100M/10M Ethernet
- Micro-HDMI ports (up to 4kp60 supported)
- 3 USB 2.0 ports. (Two of them are via 13pin interface board)
- 26 pin GPIO header with I2C, SPI, UART and multiple GPIO ports
- 13 pin GPIO header with 2 USB Host, IR pin, TV-out, Audio and GPIO ports
- 5V DC via USB-C connector (minimum 2A)
- OS: Android10, Ubuntu and Debian
- Dimension: 85mm×56mm
- Weight: 30g

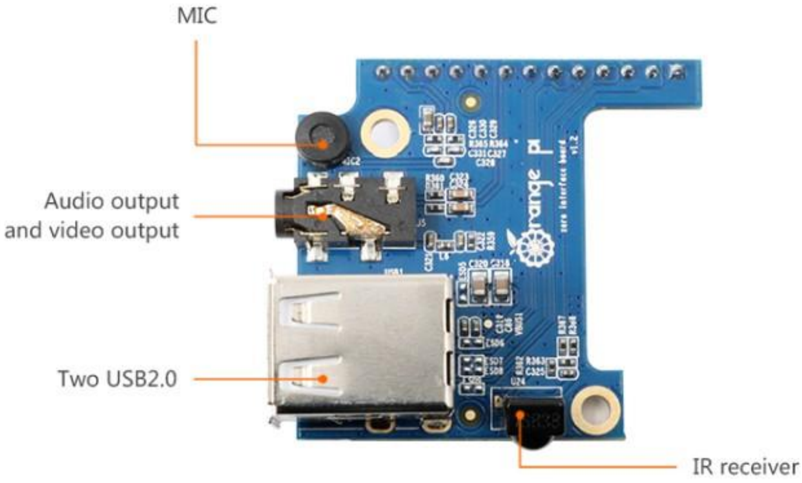
Pinout Diagram



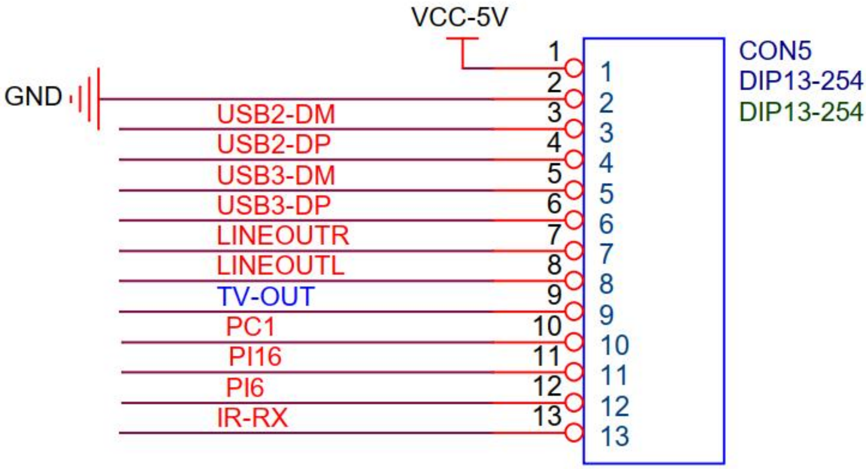
Expansion board

The expansion board is an easy way to expand the functionalities of the Orange Pi Zero2 board by providing two extra USB ports, One IR receiver, and an audio/video composite port.

Note: The Orange Pi Zero2 does not support MIC Input on the expansion board.



Expansion board schematic



Chapter 2

Getting started with your Orange Pi

Find out what items you'll need for Orange Pi and how to set up everything to get it running.

Orange Pi boards are easy to set up and beginners friendly. The Orange Pi boards are mostly self-contained and only required a few extra components to get it working. This mini-computer only needs a computer monitor or TV with an HDMI connection for display. If you want it to run as a mini headless server then you don't need a display screen either.

The Orange Pi Zero2 is very compact and offers only two USB ports out-of-box. The USB Type-C port used for powering the board and the USB Type-A port is available for use. We can get additional two USB Type-A ports with the Orange Pi Zero Addon board.

As we generally have only one USB port is available to use. We recommend using the Orange Pi Zero Addon board to get access to more USB ports and other features like an Audio/Video composite port and an IR receiver.

Optionally, you can use a USB Hub to get more USB ports to connect different peripherals.

Peripheral Requirements

If you have only bought the Orange Pi Zero2 board then you will need the following items.

USB power supply – A 5V 3amps(3A) power supply with a USB Type-C connector. The official OrangePi power supply is recommended for this board.

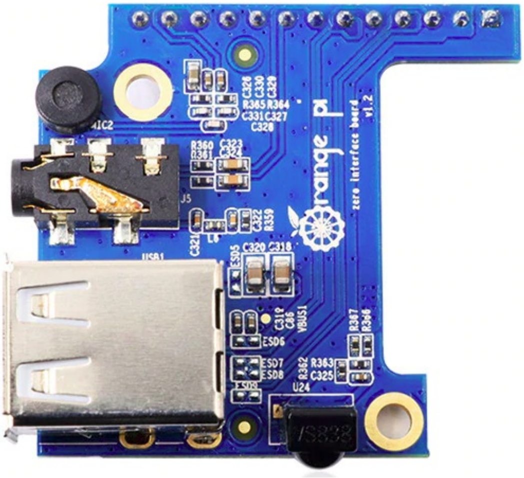
microSD card with OrangePi OS – The microSD card acts as an primary permanent storage for the OrnagePi Zero2 board. A minimum of 8GB class 10 card is required. Although 16GB microSD card is recommended. You need to write the OrangePi OS into this blank microSD card. follow Appendix A for instructions.

A keyboard and mouse – The keyboard and mouse are used to control your Orange Pi. Any wired or wireless keyboard and mouse will work. Although wired USB keyboard and mouse are preferred as they are most likely to work without any driver issues.

Micro-HDMI cable – A Micro-HDMI to HDMI cable is needed for getting Audio/Video from Orange Pi to your TV or Monitor.

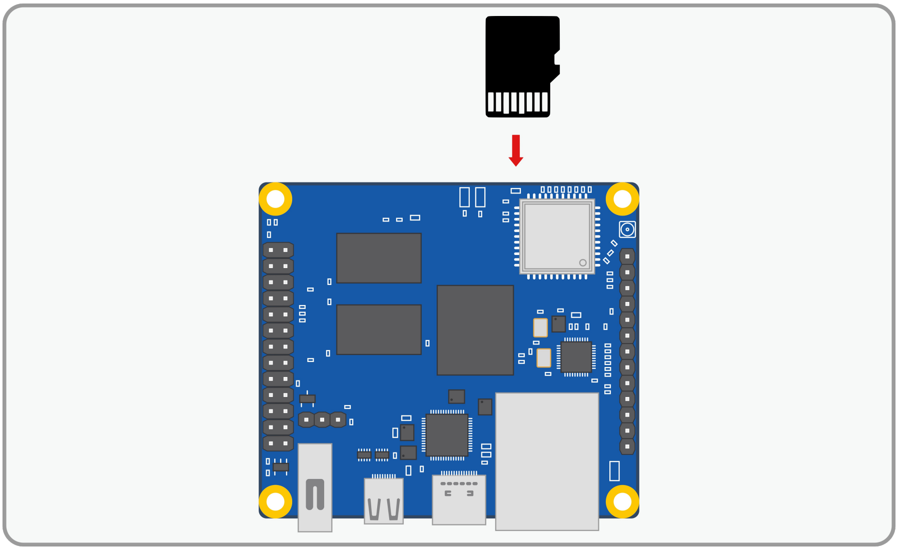
USB Hub – The Orange Pi Zero2 has only one USB Type-A port that you may need to use a USB Hub if you want to use more the one USB device.

Orange Pi Zero expansion board(Optional) – The expansion board can provide two extra USB Type-A ports and some other functionalities.

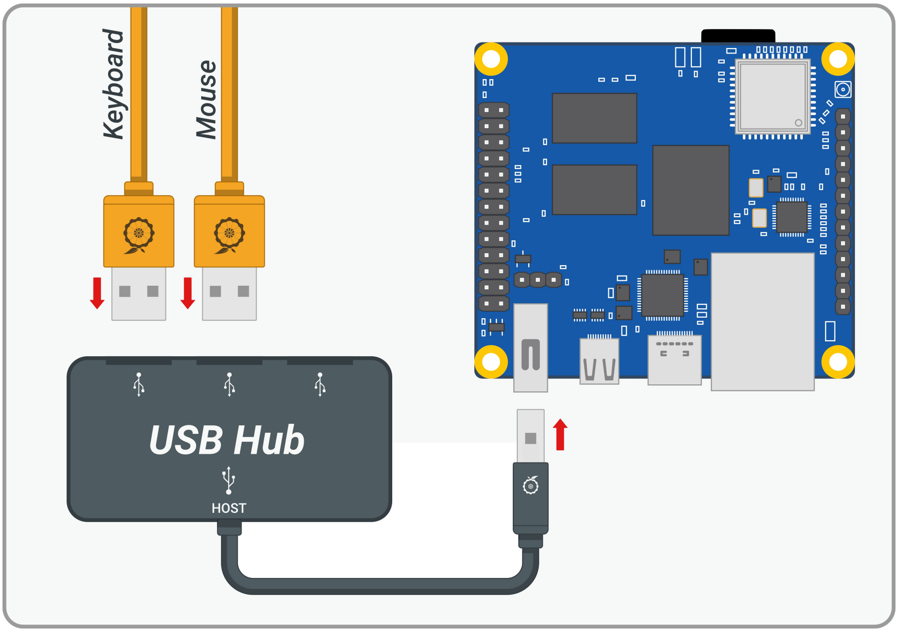


Connect your OrangePi Zero2

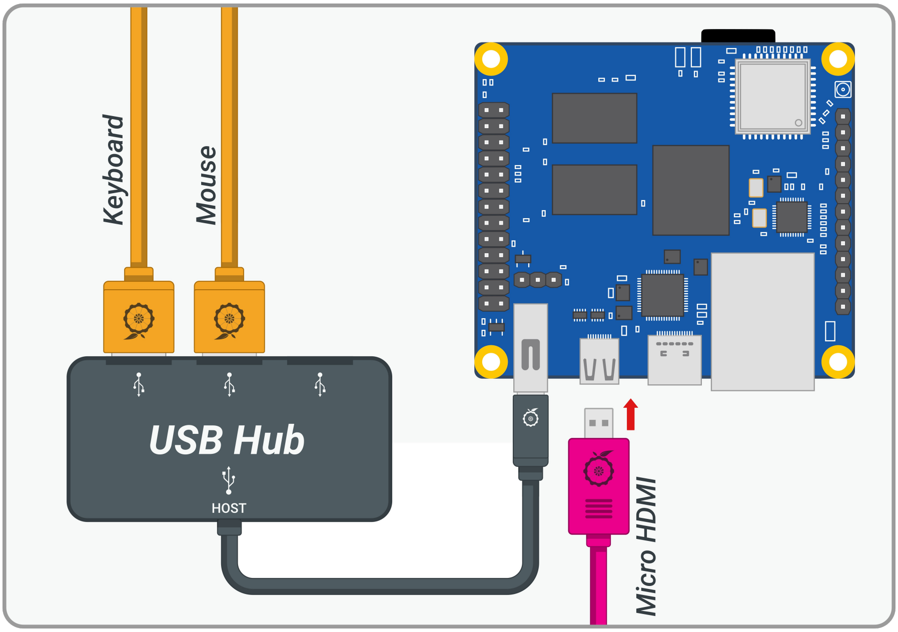
Insert the SD card – Insert the SD card into the microSD card slot on the underside of your OrangePi Zero2.



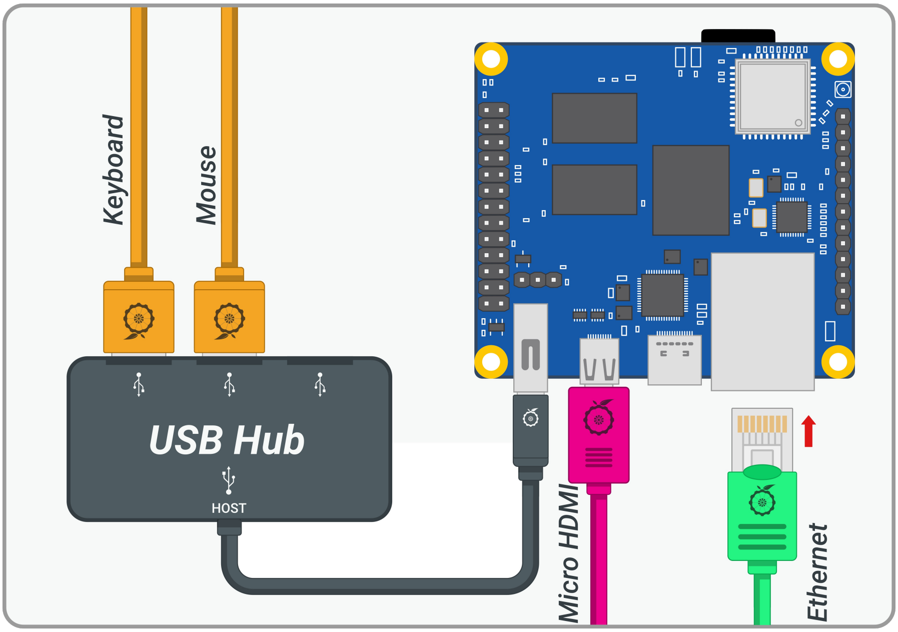
Connect USB Mouse & Keyboard – Connect the mouse to a USB port on OrangePi Zero2 (You can use either a USB Hub or OrangePi Addon board).



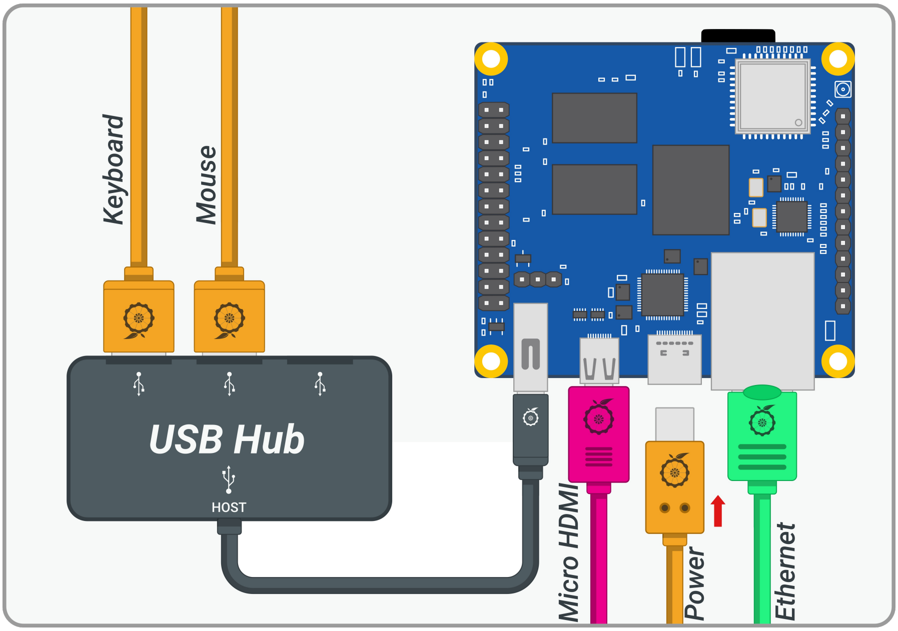
Connect HDMI cable – Connect your moniter to the Micro HDMI port of OrangePi Zero2.



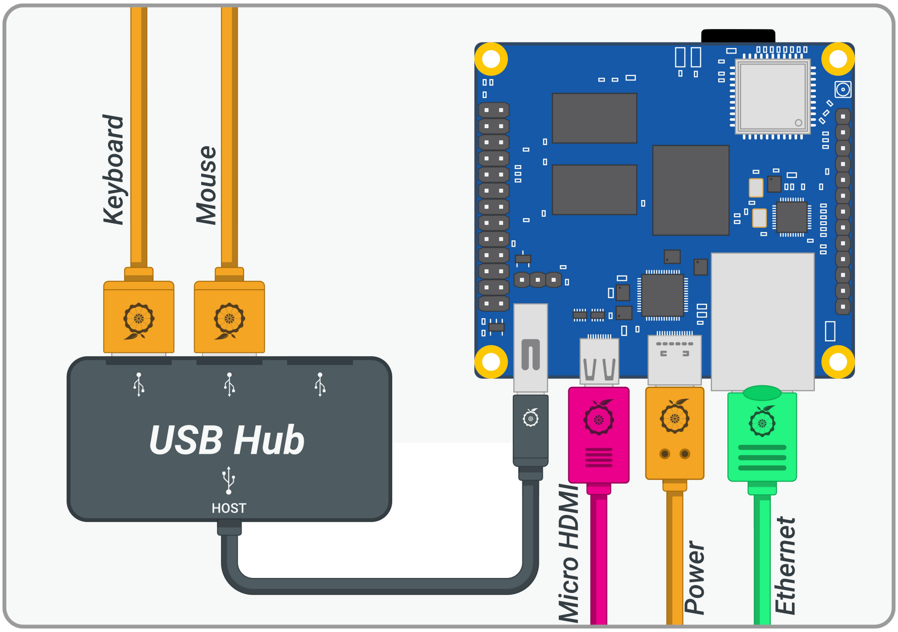
Connect Ethernet cable(Optional) – If you want to connect your OrangePi Zero2 to the internet via Ethernet then use an Ethernet cable to connect the Ethernet port on OrangePi to your internet router.



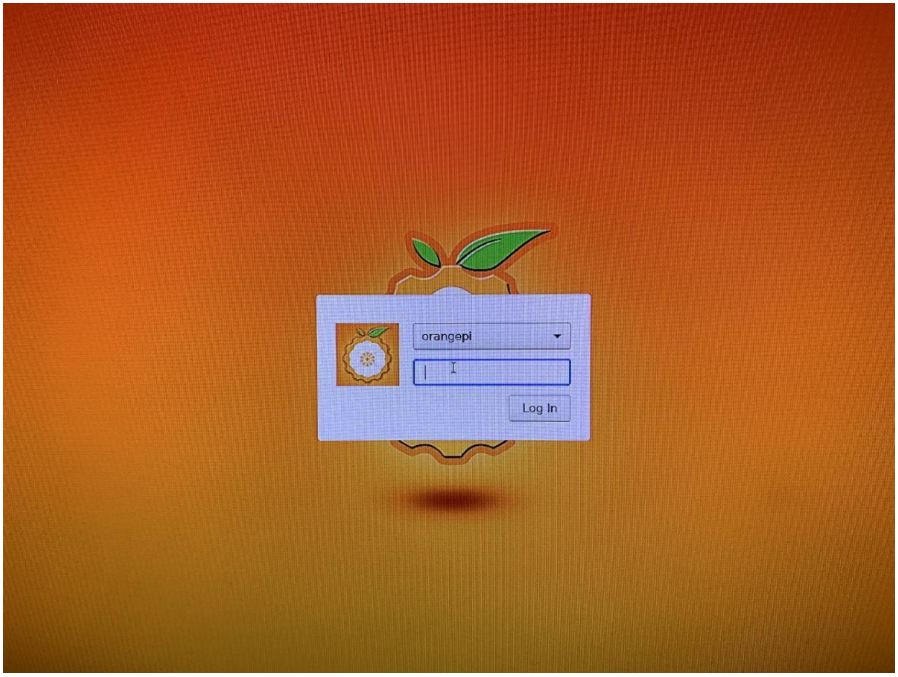
Connect USB-C Power Supply – Connect a USB-C Power supply to the Orange Pi Zero2. This board doesn't have a power switch, so it will start booting as soon as you connect it to a power supply.



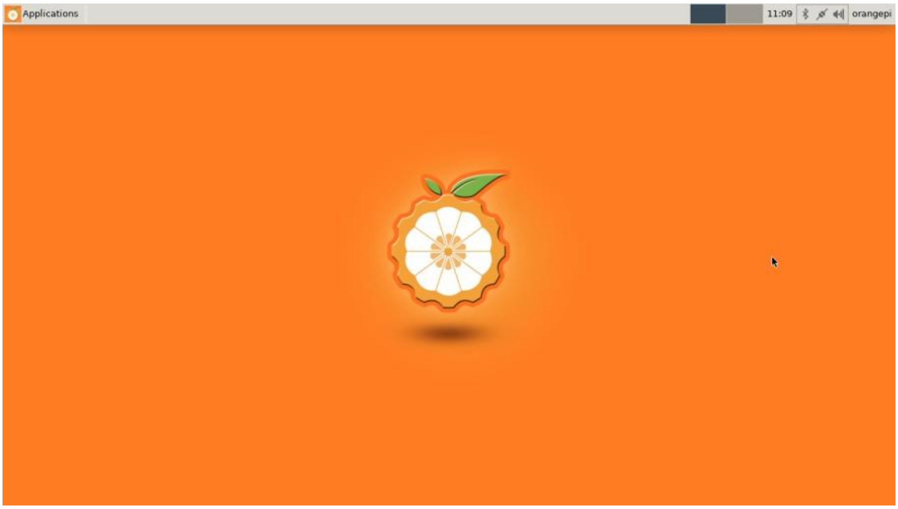
Finally, supply power to the USB-C power adapter, and OrangePi Zero2 will start booting.



After boot-up is complete, you will be greeted with the login screen. The default password for OrangePi OS is "orangepi". Enter the password and hit return.



Congratulations! You have booted your first Operating System on the OrangePi Zero2.



Chapter 3

Using Linux on your Orange Pi

Learn about the Orange Pi Operating System.

Orange Pi Zero2 can run a wide range of Operating Systems like Ubuntu, Debian, and Android 10, including server editions of Ubuntu and Debian.

In this chapter, we will learn some basic configurations like changing the screen resolution, adjusting Linux log levels, setting up an SSH connection, etc.

Changing the Linux log level

The log level of the Linux system is set to 1 by default. When using the serial port to view the boot logs, it only shows minimal information. We can increase the log levels to get more detailed system logs for debugging.

The following command will change the log level to 7.

```
root@orangepi:~# sed -i "s/verbosity=1/verbosity=7/" /boot/orangepiEnv.txt
```

Changing the screen resolution.

The screen resolution can be changed by choosing a different mode for the disp_mod variable and adjusting the width/height values of the frame buffer.

We need to change the values of fb0_width, fb0_height, and disp_mode in the **/boot/orangepiEnv.txt** file according to the following table.

disp_mode	fb0_width/fb0_height	Frame rate
480i	720x480	60
576i	720x480	50
480p	720x480	60
576p	720x576	60
720p50	1280x720	50
720p60	1280x720	60
1080i50	1920x1080	50
1080i60	1920x1080	60
1080p24	1920x1080	24
1080p50	1920x1080	50
1080p60	1920x1080	60

Setup SSH remote access

SSH remote login development board under Ubuntu

1) Get the IP address of the development board
2) Then you can log in to the linux system remotely through the ssh command

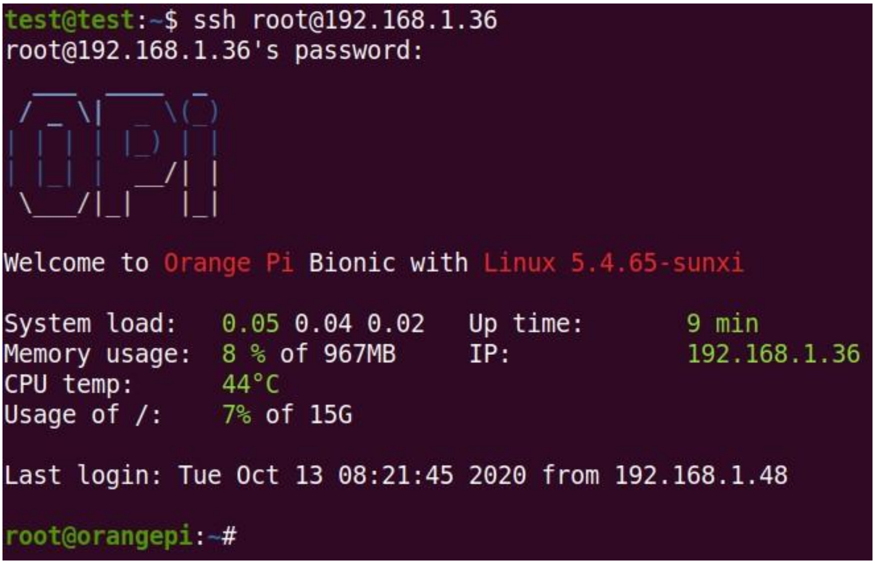
Note: Need to be replaced with the IP address of the development board

```
$ ssh root@192.168.1.36
```

Next: Enter the password here, the default password is **orangepi**

```
root@192.168.1.36's password:
```

After a successful SSH login system, you can get access to the system remotely.



```
test@test:~$ ssh root@192.168.1.36
root@192.168.1.36's password:
  ___  ____  _
 / _ \|  _ \(_)
| | | | |_) | |
| |_| |  __/| |
 \___/|_|   |_|

Welcome to Orange Pi Bionic with Linux 5.4.65-sunxi

System load:   0.05 0.04 0.02   Up time:       9 min
Memory usage:  8 % of 967MB     IP:            192.168.1.36
CPU temp:      44°C
Usage of /:    7% of 15G

Last login: Tue Oct 13 08:21:45 2020 from 192.168.1.48

root@orangepi:~#
```

SSH remote login development board under Windows

MobaXterm can be used to remotely log in to the development board under windows, first create a new ssh session

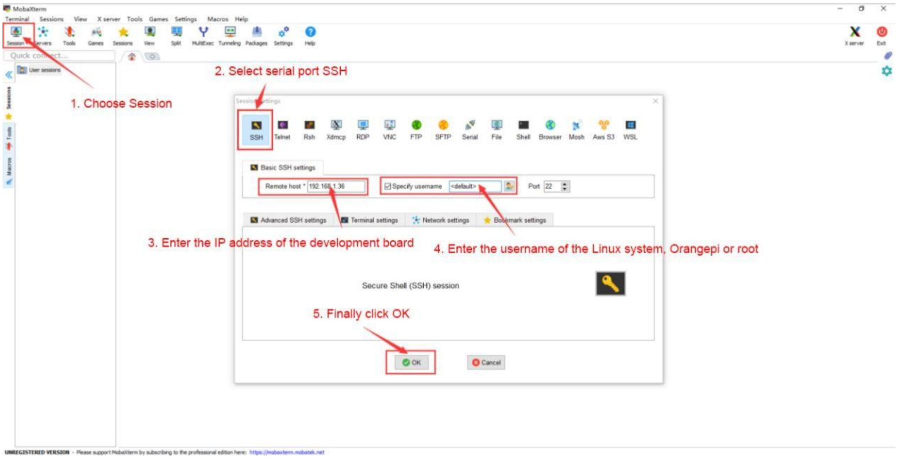
a. Open Session

b. Then select SSH in Session Setting

c. Then enter the IP address of the development board in Remote host

d. Then enter the username root or orangepi of the Linux system in Specify username

e. Finally click OK


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
User sessions
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL
Basic SSH settings
Advanced SSH settings Terminal settings Network settings Bookmark settings
Secure Shell (SSH) session
OK Cancel
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

1. Choose Session

2. Select serial port SSH

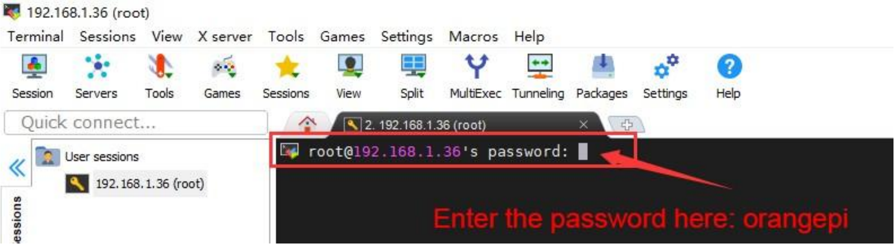
3. Enter the IP address of the development board

4. Enter the username of the Linux system, Orangepi or root

5. Finally click OK

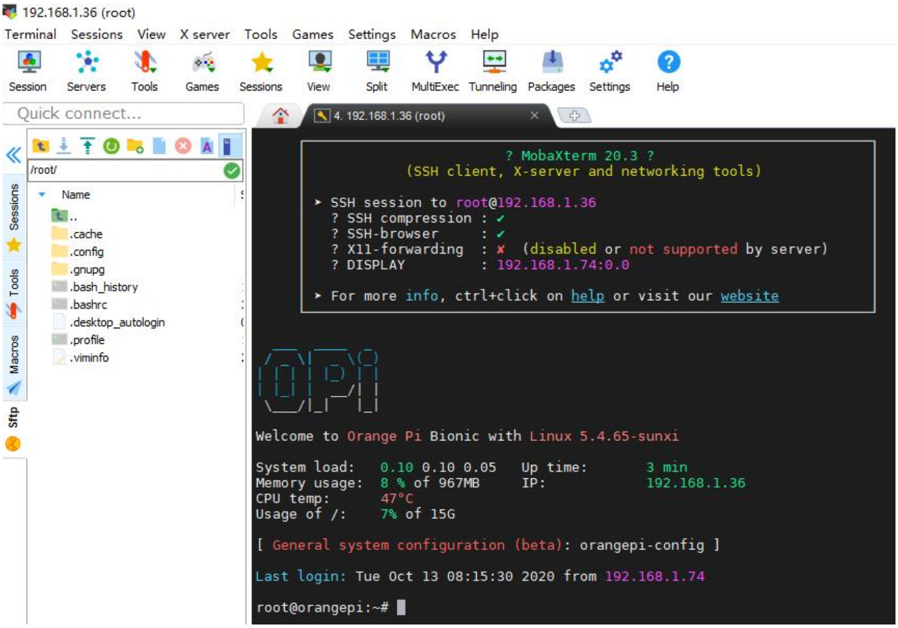
Remote host * 192.168.1.36 ☑ Specify username <default> Port 22

You will be prompted to enter a password, the default passwords for root and orangepi users are **orangepi**


192.168.1.36 (root)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 192.168.1.36 (root)
User sessions
192.168.1.36 (root)
Sessions

```
root@192.168.1.36's password:
```

Enter the password here: orangepi

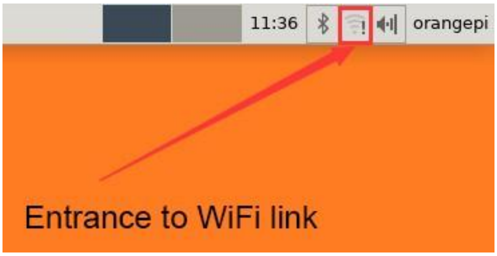


```
                    ? MobaXterm 20.3 ?
        (SSH client, X-server and networking tools)

 ➤ SSH session to root@192.168.1.36
   ? SSH compression : ✔
   ? SSH-browser     : ✔
   ? X11-forwarding  : ✘  (disabled or not supported by server)
   ? DISPLAY         : 192.168.1.74:0.0

 ➤ For more info, ctrl+click on help or visit our website

Welcome to Orange Pi Bionic with Linux 5.4.65-sunxi

System load:   0.10 0.10 0.05   Up time:       3 min
Memory usage:  8 % of 967MB     IP:            192.168.1.36
CPU temp:      47°C
Usage of /:    7% of 15G

[ General system configuration (beta): orangepi-config ]

Last login: Tue Oct 13 08:15:30 2020 from 192.168.1.74

root@orangepi:~#
```

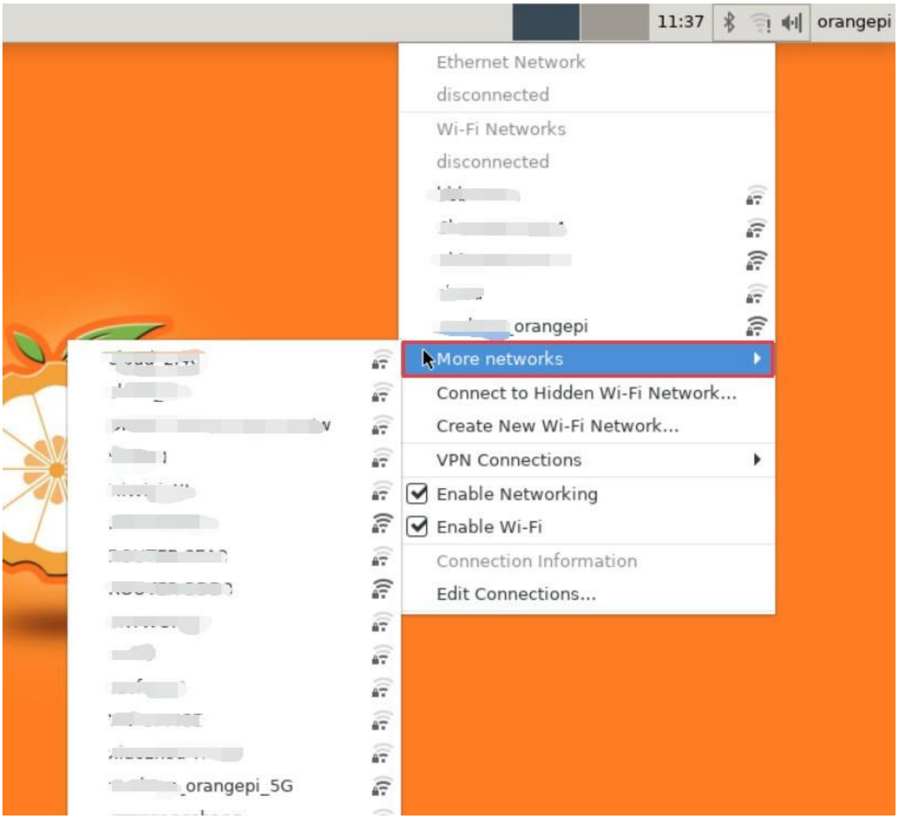
Connecting to the WiFi

Connect to the WiFi in the Desktop edition

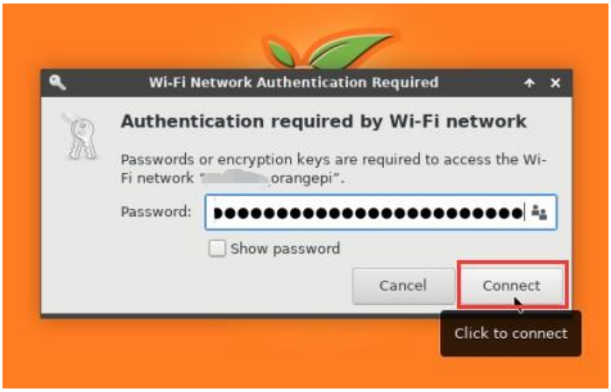
Click the network configuration icon in the upper right corner of the desktop (please do not connect the network cable when testing WIFI)



Click More networks in the pop-up drop-down box to see all scanned WIFI hotspots, then select the WIFI hotspot you want to connect to



Then enter the password of the WiFi hotspot, and then click Connect to start connecting to WiFi



Wi-Fi Network Authentication Required

Authentication required by Wi-Fi network

Passwords or encryption keys are required to access the Wi-Fi network "[redacted] orangepi".

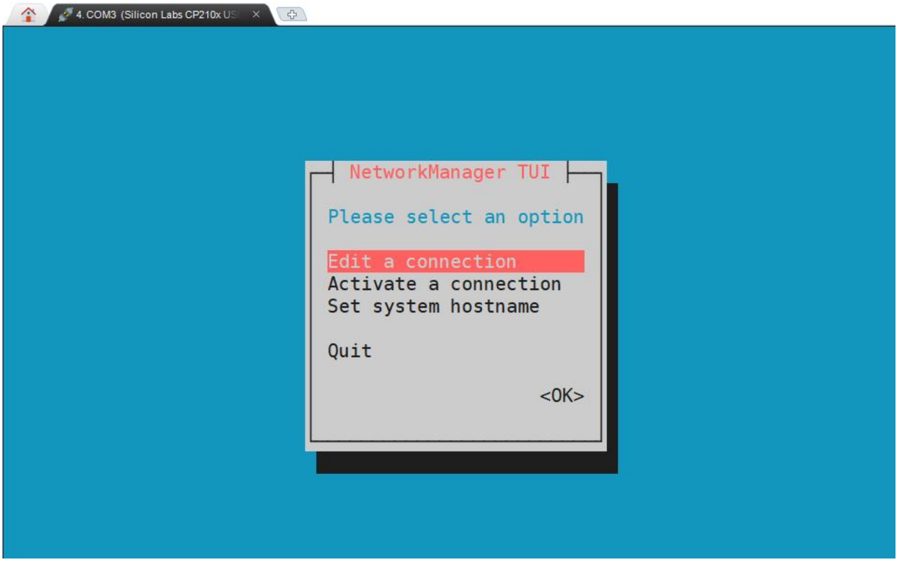
Password: ●●●●●●●●●●●●●●●●●●●●●●●●●

□ Show password

Connect to the WiFi in the Server edition

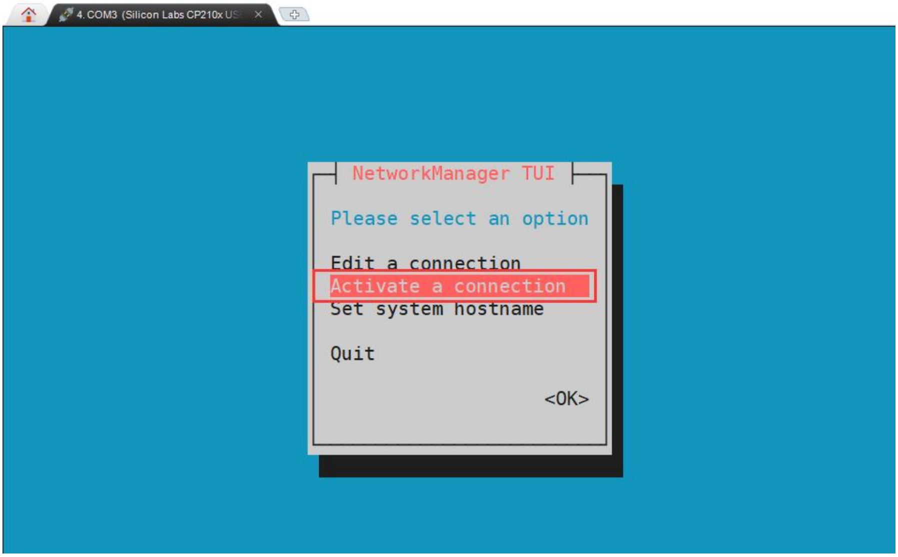
Open the command terminal and enter the following command

```
root@orangepi:~# nmtui
```



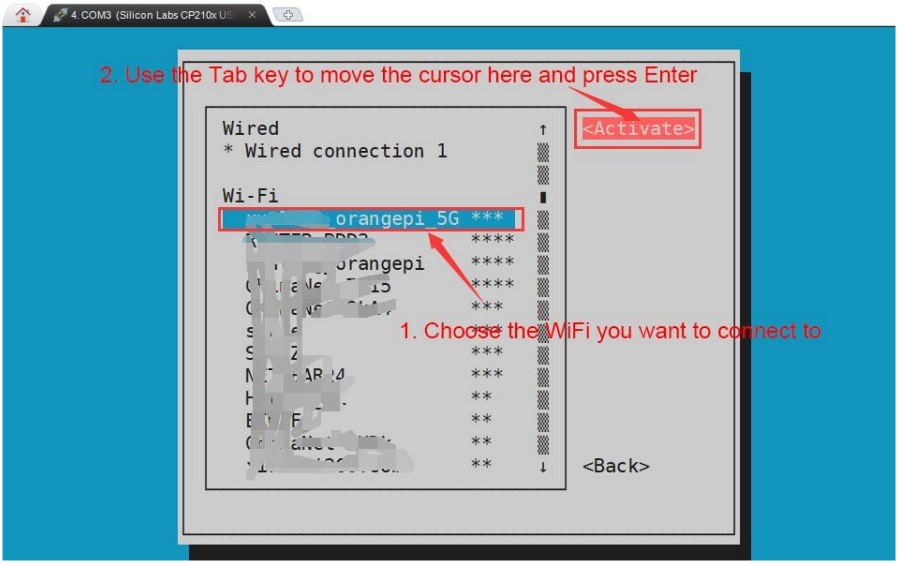
```
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

Quit

                       <OK>
```

Select **Activate a connection** and press Enter.



```
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

Quit

                         <OK>
```

Select the WiFi hotspot you want to connect to, then use the Tab key to position the cursor on Activate and press Enter

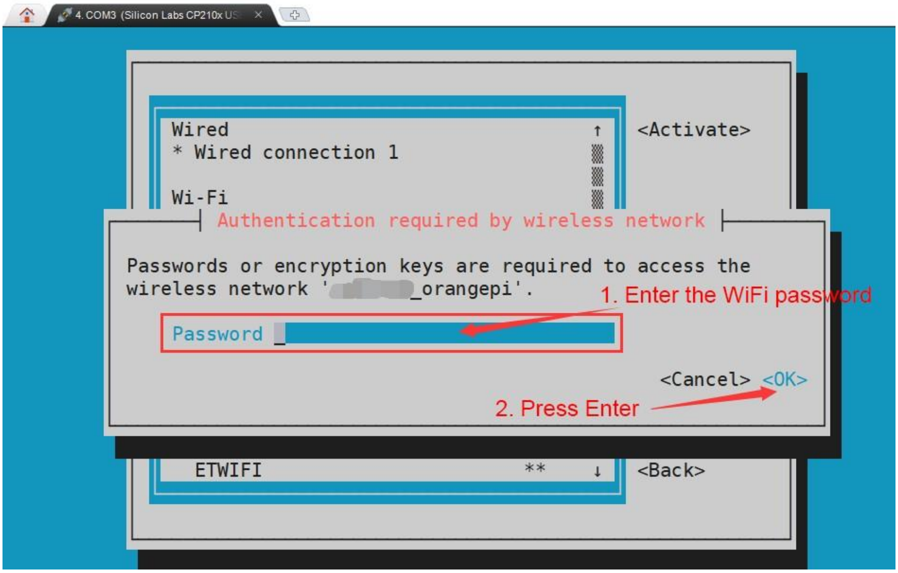


2. Use the Tab key to move the cursor here and press Enter

```
Wired                                ↑   <Activate>
  * Wired connection 1

Wi-Fi
  [redacted]_orangepi_5G  ***
  [redacted]              ****
  [redacted]orangepi      ****
  [redacted]              ****
  [redacted]              ***
  [redacted]              ***
  [redacted]              ***
  [redacted]              ***
  [redacted]              **
  [redacted]              **
  [redacted]              **
  [redacted]              **      ↓   <Back>
```

1. Choose the WiFi you want to connect to

A dialog box for entering the password will pop up, enter the corresponding password in Password and press Enter to start connecting to WiFi.

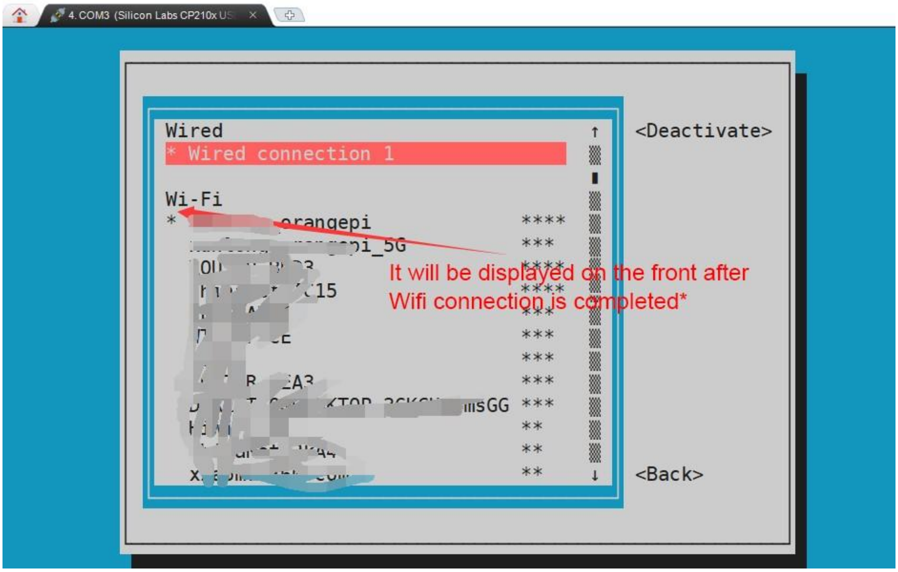


```
Wired                                          ↑   <Activate>
* Wired connection 1
Wi-Fi
  ┤ Authentication required by wireless network ├

 Passwords or encryption keys are required to access the
 wireless network '[redacted]_orangepi'.

       Password _

                                              <Cancel> <OK>

    ETWIFI                                  **   ↓   <Back>
```

1. Enter the WiFi password

2. Press Enter

After the WiFi connection is successful, a "*" will be displayed before the connected WiFi name

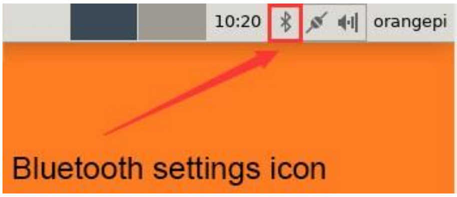


```
Wired                                          ↑   <Deactivate>
* Wired connection 1
Wi-Fi
* [redacted]orangepi                     ****
  [redacted]_5G                          ***
  [redacted]                             ****
                                         ****
                                         ***
                                         ***
                                         ***
                                         ***
                                         ***
                                         **
                                         **
                                         **    ↓   <Back>
```

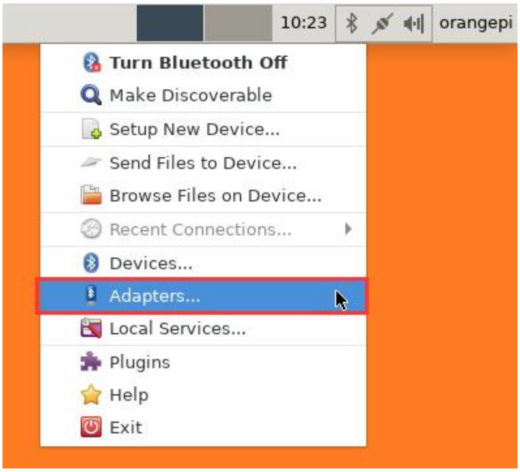
It will be displayed on the front after Wifi connection is completed*

Connecting to the Bluetooth

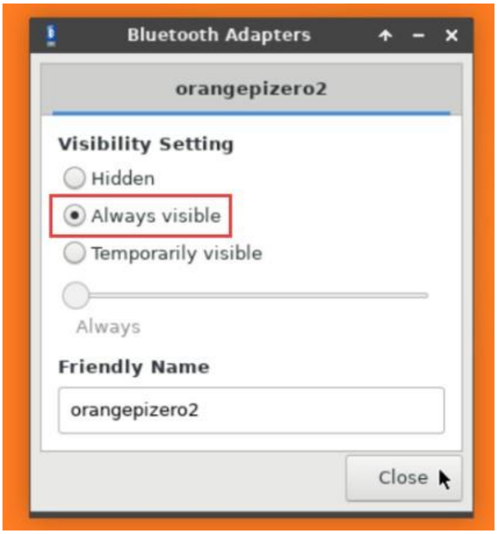
Click the Bluetooth icon in the upper right corner of the desktop



Then select the adapter



Set Visibility Setting to Always visible in the Bluetooth adapter setting interface, and then click close



Visibility Setting

○ Hidden

● Always visible

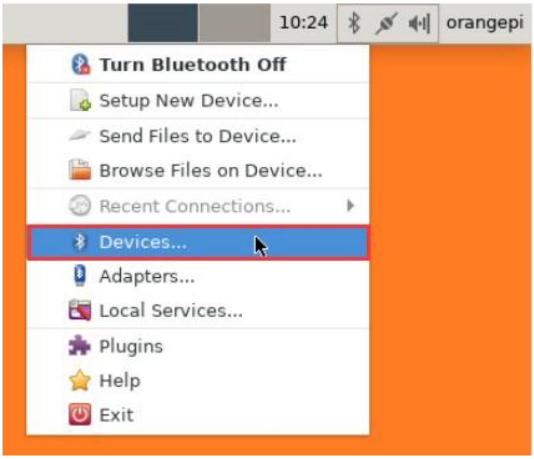
○ Temporarily visible

Always

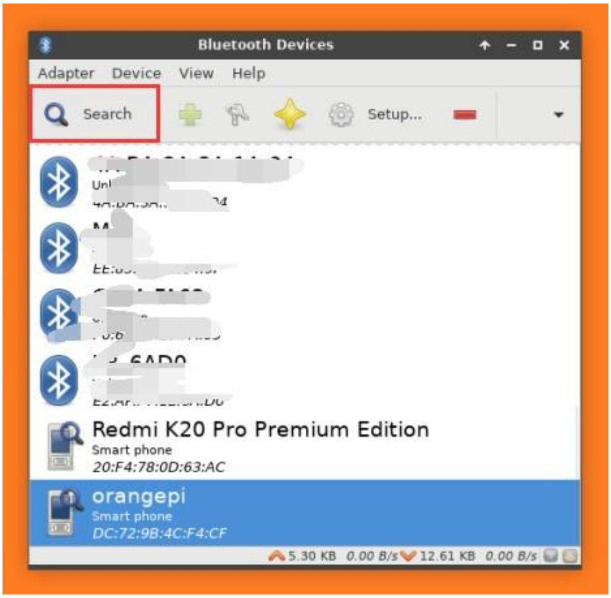
Friendly Name

orangepizero2

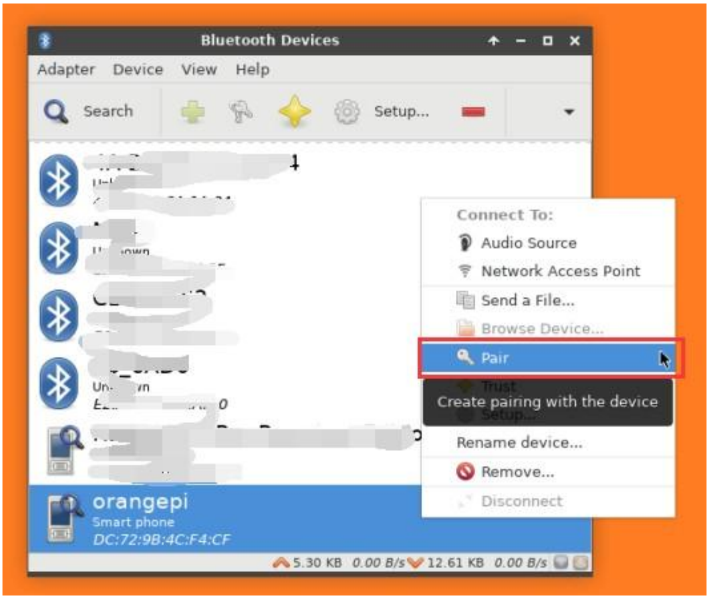
Then open the configuration interface of the Bluetooth device



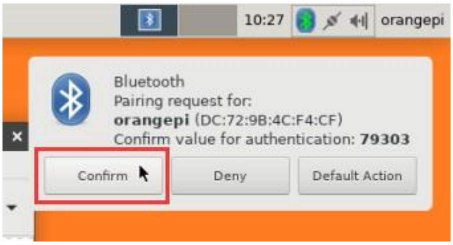
Click Search to start scanning surrounding Bluetooth devices



Then select the Bluetooth device you want to connect to, and then click the right mouse button to pop up the operation interface of the Bluetooth device. Select Pair to start pairing. Here is a demonstration of pairing with an Android phone

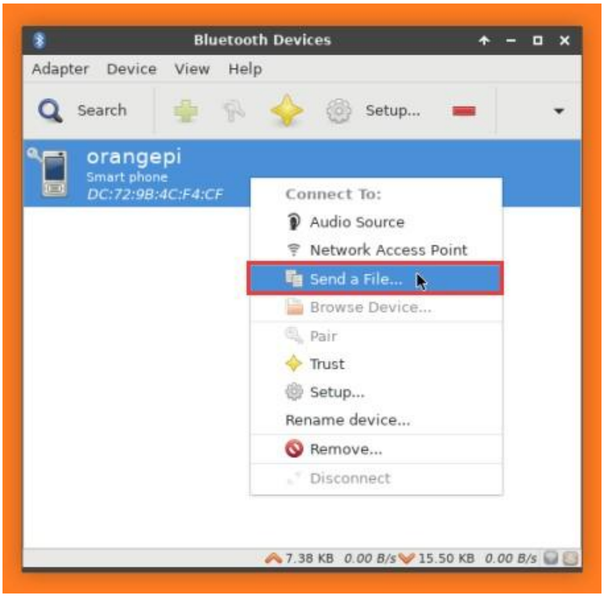


When pairing with a mobile phone, a pairing confirmation box will pop up in the upper right corner of the desktop, select Confirm to confirm. At this time, the mobile phone also needs to be confirmed

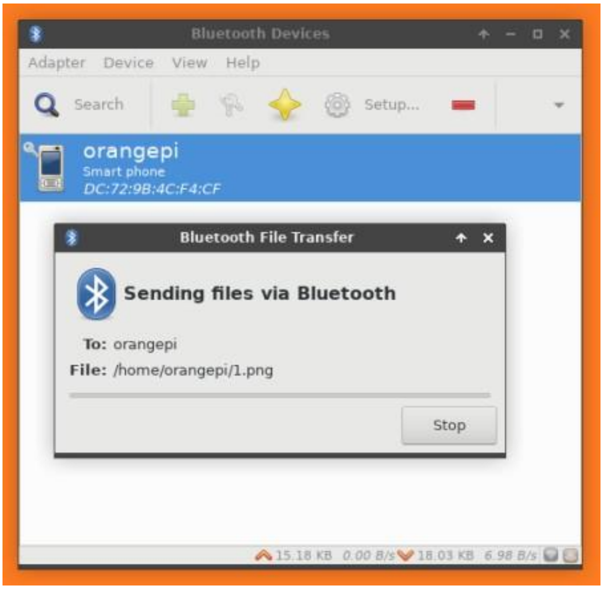


Bluetooth
Pairing request for:
orangepi (DC:72:9B:4C:F4:CF)
Confirm value for authentication: **79303**

After pairing with the phone, you can select the paired Bluetooth device, then right-click and select Send a File to start sending a picture to the phone



The interface for sending pictures is as follows



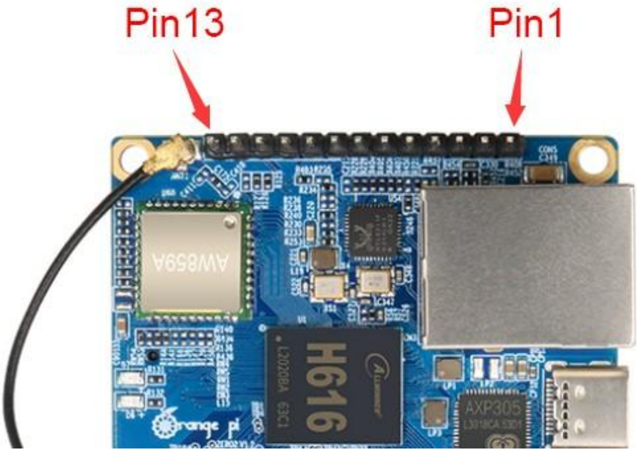
Sending files via Bluetooth

To: orangepi

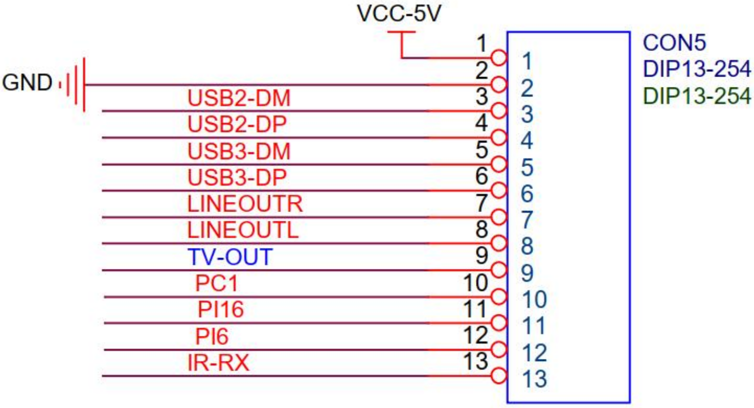
File: /home/orangepi/1.png

13 pin header for Addon board

Please refer to the figure below for the sequence of the Orange Pi Zero 2 dev board 13 pin adapter board interface pins

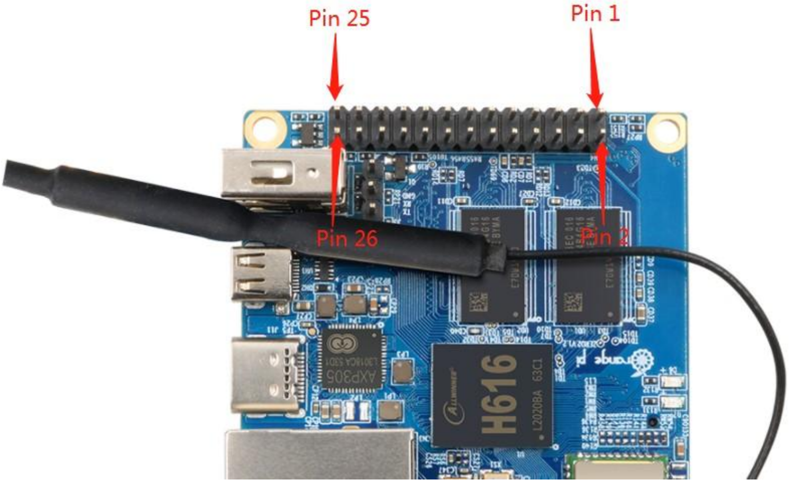


The schematic diagram of the 13pin interface of the Orange Pi Zero 2 development board is shown below



26 pin GPIO expension header

Please refer to the figure below for the sequence of the 26 pin of the Orange Pi Zero2 development board



The function of the 26 pin of the Orange Pi Zero2 development board is shown in the table below

GPIO No.	GPIO	Function	Pin	Pin	Function	GPIO	GPIO No.
		3.3V	1	2	5V		
229	PH5	TWI3-SDA	3	4	5V		
228	PH4	TWI3-SCK	5	6	GND		
73	PC9	PC9	7	8	UART5_TX	PH2	226
		GND	9	10	UART5_RX	PH3	227
70	PC6	PC6	11	12	PC11	PC11	75
69	PC5	PC5	13	14	GND		
72	PC8	PC8	15	16	PC15	PC15	79
		3.3V	17	18	PC14	PC14	78
231	PH7	SPI1_MOSI	19	20	GND		
232	PH8	SPI1_MISO	21	22	PC7	PC7	71
230	PH6	SPI1_CLK	23	24	SPI1_CS	PH9	233
		GND	25	26	PC10	PC10	74

Chapter 4

Using Android on your Orange Pi

Learn about the Android 10 Operating System.

Orange Pi Zero2 can run a wide range of Operating Systems like Ubuntu, Debian, and Android 10, including server editions of Ubuntu and Debian.

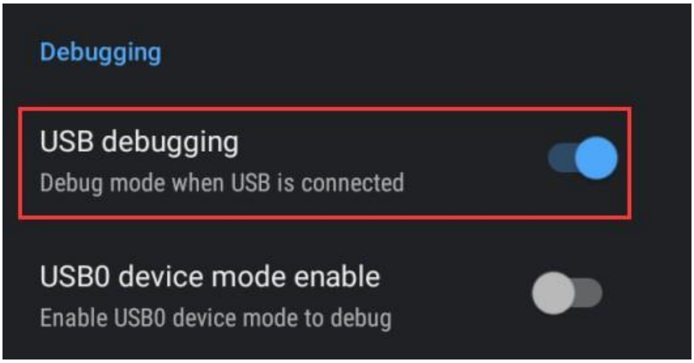
In this chapter, we will learn some basic configurations in Android 10 Operating system.

Onboard LED light display description

	Green light	Red light
u-boot startup phase	OFF	ON
Kernel boot to enter system	ON	OFF
GPIO 口	PC13	PC12

Using USB debugging.

Goto **Settings -> Device Preferences -> Developer options**
Find **USB debugging**, make sure it is turned on



Debugging

USB debugging
Debug mode when USB is connected

USB0 device mode enable
Enable USB0 device mode to debug

Use data cable to connect adb debugging

Prepare a USB Typc C interface data cable. One end of the USB interface is inserted into the USB interface of the computer, and the other end of the Type C interface is inserted into the power interface of the development board. In this case, the USB interface of the computer supplies power to the development board, so please ensure that the USB interface of the computer can provide the most power to drive the development board.

Install adb tool on Ubuntu using following commands

```
$ sudo apt update
$ sudo apt install adb
```

View the identified ADB device

```
$ adb devices
List of devices attached
8c00141167058911ccd device
```

Then you can log in to the android system through adb shell on the Ubuntu PC

```
$ adb shell
cupid-p2:/ #
```

Use network connection adb debugging

1) Using the network adb does not require a USB Typc C interface data cable to connect the computer and the development board, but communicates through the network, so first make sure that the wired or wireless network of the development board is connected, and then obtain the IP address of the development board , To be used later

2) Make sure that the **USB debugging** option is turned on

3) Make sure that the **service.adb.tcp.port** of the Android system is set to port number 5555

```
cupid-p2:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

4) If service.adb.tcp.port is not set, you can use the following command to set the port number of the network adb

```
cupid-p2:/ # setprop service.adb.tcp.port 5555
cupid-p2:/ # stop adbd
cupid-p2:/ # start adbd
```

5) Install adb tool on Ubuntu PC

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install adb
```

6) Then connect to the network adb on the Ubuntu PC

test@test:~$ adb connect 192.168.1.xxx (The IP address needs to be modified to the IP address of the development board)

```
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xxx:5555

test@test:~$ adb devices
List of devices attached
192.168.1.xxx:5555 device
```

7) Then you can log in to the android system through adb shell on the Ubuntu PC

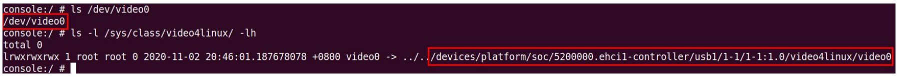
```
test@test:~$ adb shell
cupid-p2:/ #
```

How to use USB camera

1) First insert the USB camera into the USB interface of the development board, and then confirm that the kernel module related to the USB camera has been loaded normally

```
console:/ # lsmod
Module              Size    Used by
sprdwl_ng           405504 0
sprdbt_tty          36864 2
uwe5622_bsp_sdio    274432 2 sprdwl_ng,sprdbt_tty
uvcvideo            102400 0
videobuf2_v4l2      28672 1 uvcvideo
videobuf2_vmalloc   16384 1 uvcvideo
videobuf2_memops    16384 1 videobuf2_vmalloc
videobuf2_core      49152 2 uvcvideo,videobuf2_v4l2
mali_kbase          532480 7
```

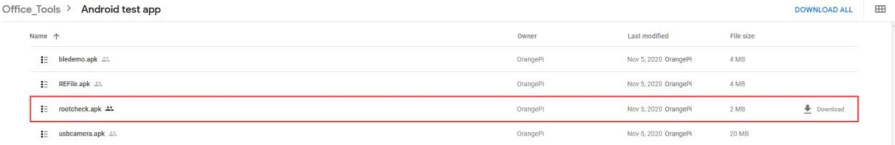
2) If the USB camera is recognized normally, the corresponding video device node will be generated under **/dev**



```
console:/ # ls /dev/video0
/dev/video0
console:/ # ls -l /sys/class/video4linux/ -lh
total 0
lrwxrwxrwx 1 root root 0 2020-11-02 20:46:01.187678078 +0800 video0 -> ../../devices/platform/soc/5200000.ehci1-controller/usb1/1-1/1-1:1.0/video4linux/video0
console:/ #
```

3) Then make sure that the adb connection between the Ubuntu PC and the development board is normal

4) Download the USB camera test APP from the official tool on the page below the Orange Pi Zero 2 information



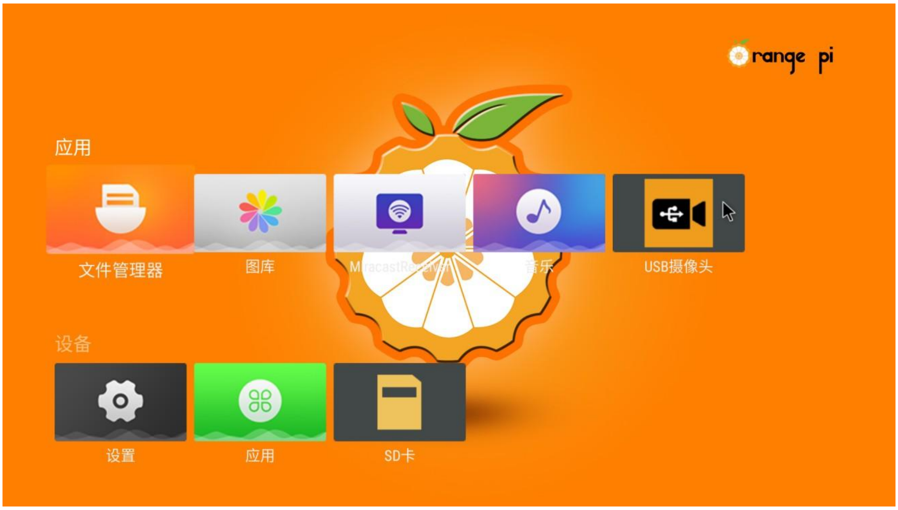
Office_Tools › Android test app

Name ↑	Owner	Last modified	File size	
bledemo.apk	OrangePi	Nov 5, 2020 OrangePi	4 MB	
REFile.apk	OrangePi	Nov 5, 2020 OrangePi	4 MB	
rootcheck.apk	OrangePi	Nov 5, 2020 OrangePi	2 MB	Download
usbcamera.apk	OrangePi	Nov 5, 2020 OrangePi	20 MB	

5) Then use the adb command to install the USB camera test APP to the Android system, of course, you can also use the U disk copy method to install

test@test:~$ adb install usbcamera.apk

6) After installation, you can see the startup icon of the USB camera on the Android desktop



7) Then double-click to open the USB camera APP and you can see the output video of the USB camera

Android system ROOT description

The Android 10.0 system released by Orange Pi is already ROOT, you can use the following method to test

1) Download rootcheck.apk from the official tool on the Orange Pi Zero 2 data download page

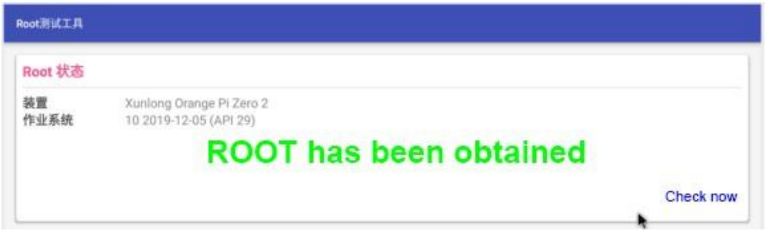
2) Then make sure that the adb connection between the Ubuntu PC and the development board is normal

3) Then use the adb command to install rootcheck.apk to the Android system, of course, you can also use the U disk copy to install

```
test@test:~$ adb install rootcheck.apk
```

4) After installation, you can see the startup icon of the ROOT test tool on the Android desktop

6) Open the **ROOT test tool** and click "Check now" to start the inspection of the ROOT status of the Android system. The display after the inspection is as follows, you can see that the Android system has obtained ROOT permission



装置 Xunlong Orange Pi Zero 2
作业系统 10 2019-12-05 (API 29)

ROOT has been obtained

Chapter 5

Building Linux from source

Build your own Linux image with customizations

Linux images for Orange Pi Zero2 can be easily built by the build script specially made for Orange Pi boards. The build script can make server and desktop versions of Debian and Ubuntu.

Orange Pi build script based upon Armbian build script. We will make a few different images as an example.

Get the Build script source

The compilation of the Linux is done on a PC with **Ubuntu 18.04** installed. Other versions of Ubuntu systems may have some differences

We will need Git tool for download the source repository. You can install Git using following commands

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install git
```

Download the source repository using the following command.

```
test@test:~$ git clone https://github.com/orangepi-xunlong/orangepi-build.git
```

Orangepi-build will contain the following files and folders after downloading

build.sh: Compile the startup script

external: Contains the configuration files needed to compile the image, specific scripts, and the source code of some programs, etc.

LICENSE: GPL 2 license file

README.md: orangepi-build instruction file

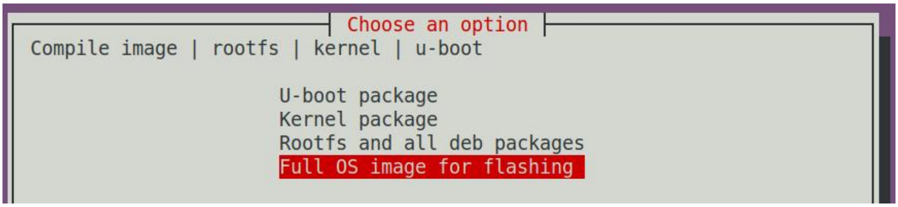
scripts: general scripts for compiling linux images

Build a Server Linux

1) Run the build.sh script, remember to add sudo permissions

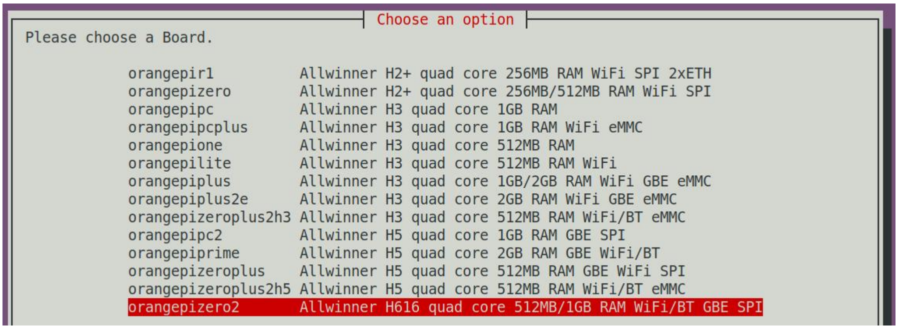
test@test:~/orangepi-build$ sudo ./build.sh

2) Select **Full OS image for flashing**, then press Enter



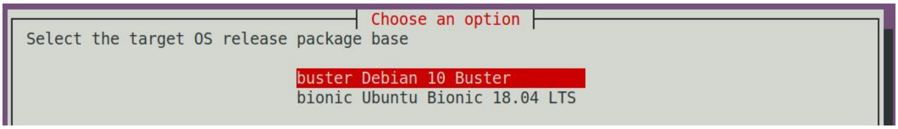
```
                          Choose an option
 Compile image | rootfs | kernel | u-boot

                          U-boot package
                          Kernel package
                          Rootfs and all deb packages
                          Full OS image for flashing
```

3) Then select the model of the development board



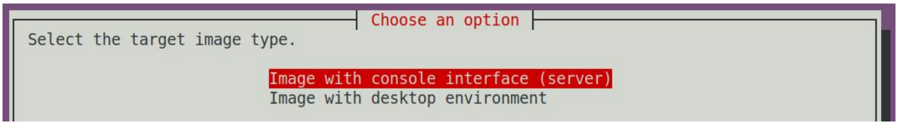
```
                              Choose an option
 Please choose a Board.

            orangepir1          Allwinner H2+ quad core 256MB RAM WiFi SPI 2xETH
            orangepizero        Allwinner H2+ quad core 256MB/512MB RAM WiFi SPI
            orangepipc          Allwinner H3 quad core 1GB RAM
            orangepipcplus      Allwinner H3 quad core 1GB RAM WiFi eMMC
            orangepione         Allwinner H3 quad core 512MB RAM
            orangepilite        Allwinner H3 quad core 512MB RAM WiFi
            orangepiplus        Allwinner H3 quad core 1GB/2GB RAM WiFi GBE eMMC
            orangepiplus2e      Allwinner H3 quad core 2GB RAM WiFi GBE eMMC
            orangepizeroplus2h3 Allwinner H3 quad core 512MB RAM WiFi/BT eMMC
            orangepipc2         Allwinner H5 quad core 1GB RAM GBE SPI
            orangepiprime       Allwinner H5 quad core 2GB RAM GBE WiFi/BT
            orangepizeroplus    Allwinner H5 quad core 512MB RAM GBE WiFi SPI
            orangepizeroplus2h5 Allwinner H5 quad core 512MB RAM WiFi/BT eMMC
            orangepizero2       Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
```

4) Then select the type of rootfs



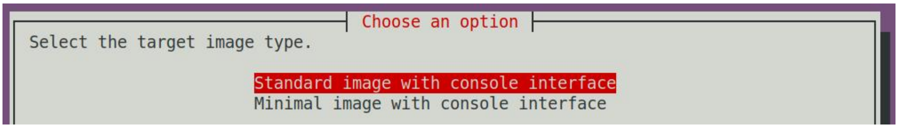
```
                              Choose an option
 Select the target OS release package base

                              buster Debian 10 Buster
                              bionic Ubuntu Bionic 18.04 LTS
```

5) Then select the type of Image you wanna build.



```
Choose an option
Select the target image type.

        Image with console interface (server)
        Image with desktop environment
```

6) Select between Standard and Minimal Image.



```
Choose an option
Select the target image type.

        Standard image with console interface
        Minimal image with console interface
```

7) After compiling the image, the following information will be prompted

A. The storage path of the compiled image

[o.k.] Done building [output/images/Orangepizero2_2.0.8_ubuntu_bionic_server_linux4.9.170/ Orangepizero2_2.0.8_ubuntu_bionic_server_linux4.9.170.img]

B. Compilation time

[o.k.] Runtime [19 min]

C. Repeat the command to compile the image, use the following command without selecting through the graphical interface, you can directly start to compile the image

[o.k.] Repeat Build Options [sudo ./build.sh BOARD=orangepizero2 BRANCH=legacy BUILD_OPT=image RELEASE=bionic BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes]

Chapter 5

Building Android from source

Build your own Android image with customizations

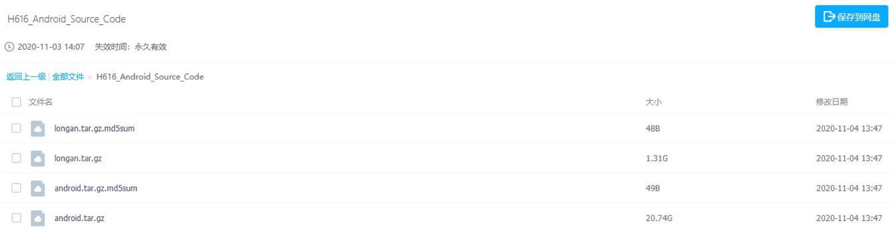
Android images for Orange Pi Zero2 can be easily built by the vendor provided BSP sources. This features a latest version of Android with all drivers for GPU and media playback.

1) The compilation of the Android SDK is performed on a PC with Ubuntu 14.04 installed, and other versions of Ubuntu systems may have some differences

2) Android SDK is the original SDK released by the chip manufacturer. If you want to use the Android image compiled by the Android SDK on the Orange Pi development board, you need to adapt to different boards to ensure that all functions are used normally

Download the source code of android sdk

1) The android source code of H616 contains the following 4 files
a. android.tar.gz: android mirror source code
b. android.tar.gz.md5sum: MD5 checksum file of android.tar.gz
c. longan.tar.gz: Contains u-boot, linux kernel source code (does not include boot0 source code)
d. longan.tar.gz.md5sum: MD5 checksum file of longan.tar.gz



H616_Android_Source_Code

2020-11-03 14:07 失效时间：永久有效

□ 文件名	大小	修改日期
□ longan.tar.gz.md5sum	48B	2020-11-04 13:47
□ longan.tar.gz	1.31G	2020-11-04 13:47
□ android.tar.gz.md5sum	49B	2020-11-04 13:47
□ android.tar.gz	20.74G	2020-11-04 13:47

2) After downloading the android source code, first check whether the MD5 checksum is correct, if not, please download the source code again

```
test@test:~$ md5sum -c android.tar.gz.md5sum
android.tar.gz: confirm
test@test:~$ md5sum -c longan.tar.gz.md5sum
longan.tar.gz: confirm
```

3) Then unzip the android source code

a. android: store android-related source code

b. longan: store the source code of the linux kernel and u-boot (not including the source code of boot0), and other configuration files

```
test@test:~$ tar -zxf android.tar.gz
test@test:~$ tar -zxf longan.tar.gz
test@test:~$ ls
android longan
```

Build android compilation environment

1) Install JDK

```
test@test:~$ sudo add-apt-repository ppa:openjdk-r/ppa
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install openjdk-8-jdk
```

2) Configure JAVA environment variables

a. First determine the installation path of java, generally

```
test@test:~$ ls /usr/lib/jvm/java-8-openjdk-amd64
ASSEMBLY_EXCEPTION bin docs include THIRD_PARTY_README
```

b. Then use the following command to export java environment variables

```
test@test:~$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
test@test:~$ export PATH=$JAVA_HOME/bin:$PATH
test@test:~$ export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar
```

3) Use Ubuntu 14.04 to compile the source code of android 10, you need to ensure that Ubuntu 14.04 uses the linux 4.4 kernel, otherwise an error will be reported when compiling, if the kernel is not linux 4.4, please upgrade the kernel

```
test@test:~$ uname -a
Linux ubuntu 4.4.0-142-generic #168~14.04.1-Ubuntu SMP Sat Jan 19 11:26:28
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
```

4) Install platform support software

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install git gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip
test@test:~$ sudo apt-get install u-boot-tools
```

Compile android image

1) First configure the compilation environment

```
test@test:~$ cd longan
test@test:~/longan$ ./build.sh config
Welcome to mkscript setup progress
All available platform:
0. android
1. linux
Choice [android]: 0
All available ic:
0. h313
1. h616
2. h700
Choice [h616]: 1
```

```
All available board:
0. fpga
1. ft
2. p1
3. p2
4. perf1
5. perf1_axp152
6. perf2
7. perf3
8. qa
Choice [p2]: 3
INFO: kernel defconfig: generate /wspace2/H616/Android_10/longan/kernel/linux-
4.9/.config/wspace2/H616/Android_10/longan/kernel/linux-4.9/arch/arm64/
configs/sun50iw9p1smp_h616_android_defconfig
*** Default configuration is based on 'sun50iw9p1smp_h616_android_defconfig'
#
# configuration written to .config
```

2) Then start compiling

```
test@test:~/longan$ ./build.sh
```

3) The output after compilation is as follows

```
sun50iw9p1 compile Kernel successful

INFO: build kernel OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: ----------------------------------------
INFO: build lichee OK.
INFO: ----------------------------------------
```

Compile android source code

1) The command to compile android is as follows

```
test@test:~$ cd android
test@test:~/android$ source build/envsetup.sh
test@test:~/android$ lunch cupid_p2-eng
test@test:~/android$ extract-bsp
test@test:~/android$ make -j8
```

2) After compiling, the following information will be printed

```
#### build completed successfully (01:51 (mm:ss)) ####
```

3) Then use the pack command to package and generate the android image

```
test@test:~/android$ pack
......
----------image is at----------
longan/out/h616_android10_p2_uart0.img
pack finish
use pack4dist for release
```

4) The storage path of the generated Android image is

```
longan/out/h616_android10_p2_uart0.img
```

Appendix A

Install an OS to a microSD card

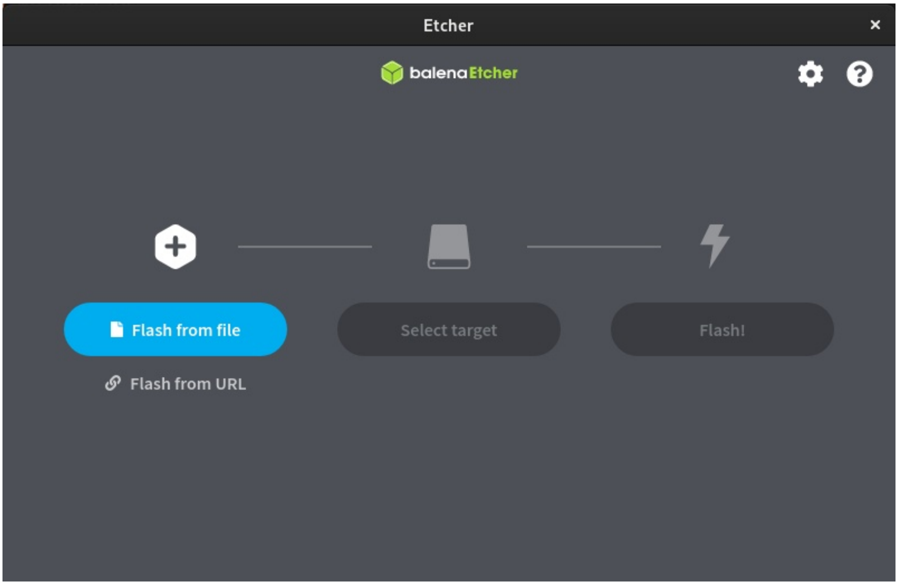
1. Download the tools.

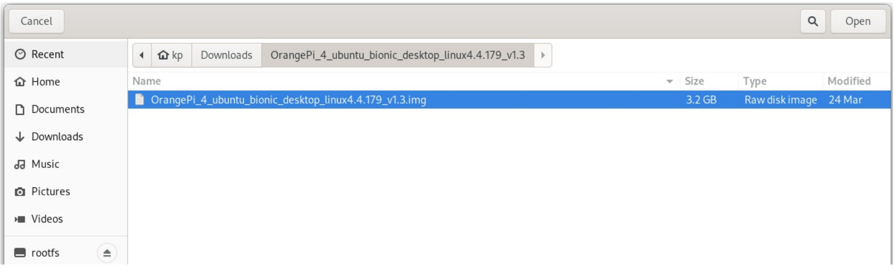
We are going to use a Open Source tool called **balenaEtcher** for writing Orange Pi OS images to MicroSD card. Download the tool using following link.

Download balenaEtcher from here.

2. Select an Orange Pi OS image.

Click on **Flash from file** and select a OS Image.

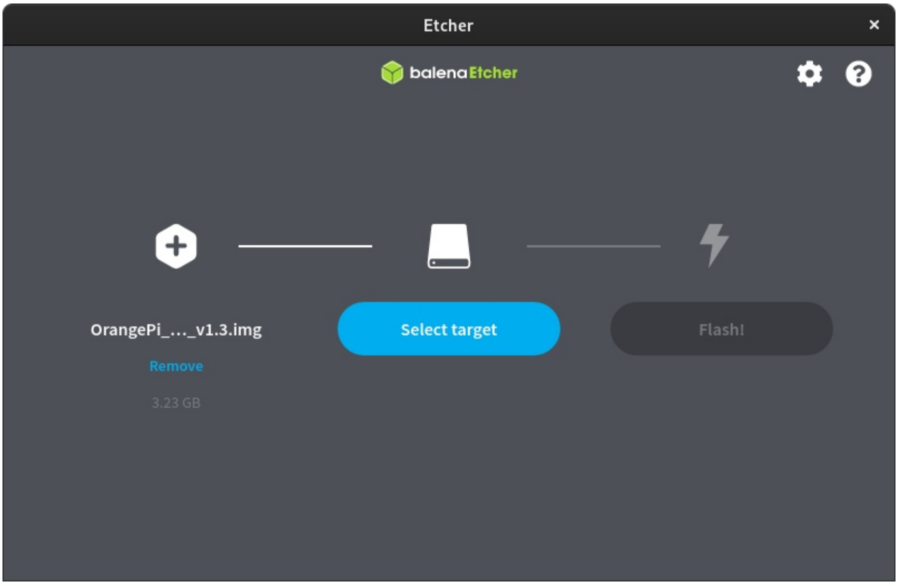


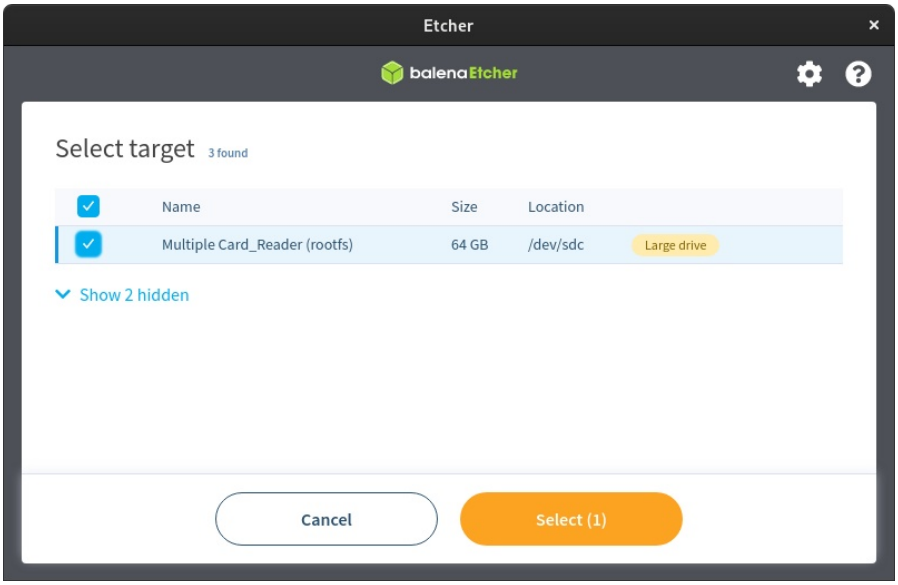


Name	Size	Type	Modified
OrangePi_4_ubuntu_bionic_desktop_linux4.4.179_v1.3.img	3.2 GB	Raw disk image	24 Mar

3. Select a MicroSD card drive.

Click on **Select target** and select your **MicroSD** drive.



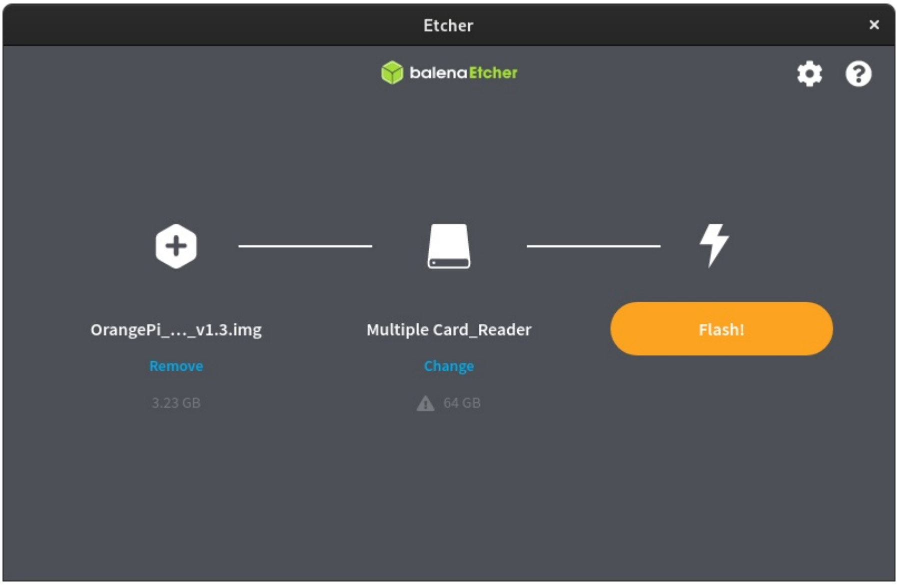


Select target 3 found

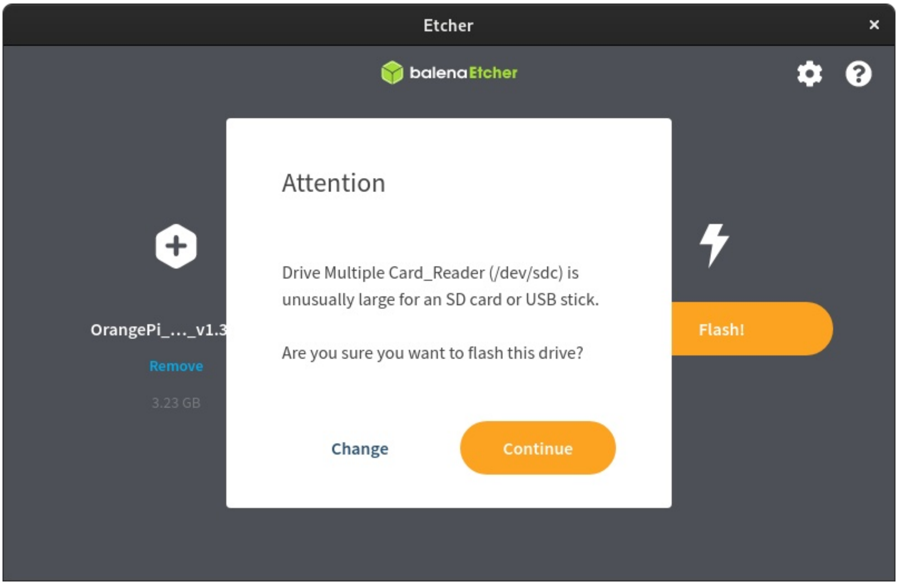
☑	Name	Size	Location	
☑	Multiple Card_Reader (rootfs)	64 GB	/dev/sdc	Large drive

4. Perform the Write opration

Click on the **Flash!** button to write OS image to MicroSD card.



Sometimes Etcher gives you a warnning about MicroSD card being unusually large if you use a large size MicroSD card.

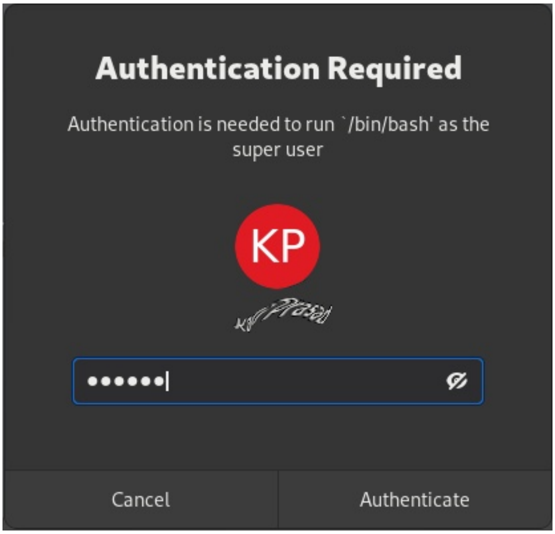


Attention

Drive Multiple Card_Reader (/dev/sdc) is unusually large for an SD card or USB stick.

Are you sure you want to flash this drive?

You may have to give Root permission to perform the write opration.

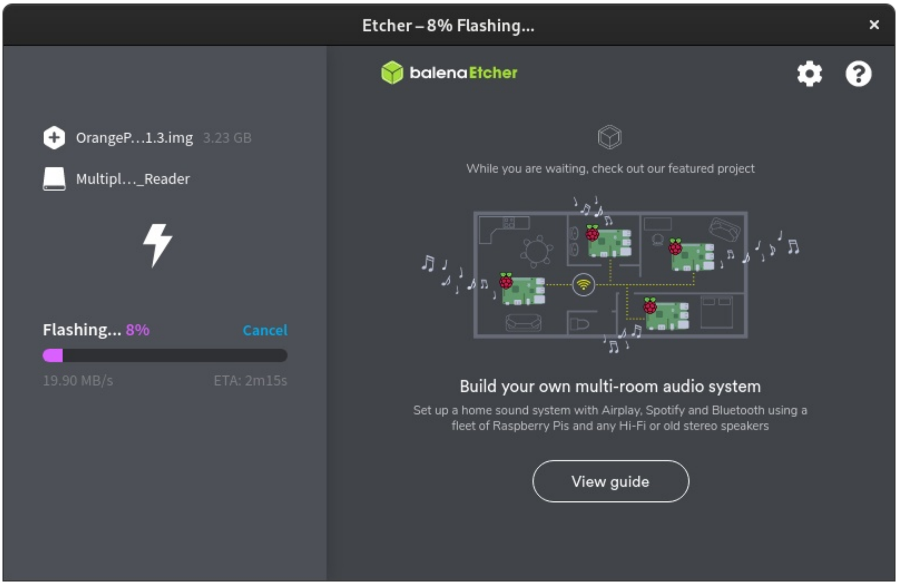


Authentication Required

Authentication is needed to run `/bin/bash' as the super user

••••••

Wait for Etcher to finish the write opration.



Etcher – 8% Flashing...

OrangeP...1.3.img 3.23 GB

Multipl..._Reader

Flashing... 8%

19.90 MB/s ETA: 2m15s

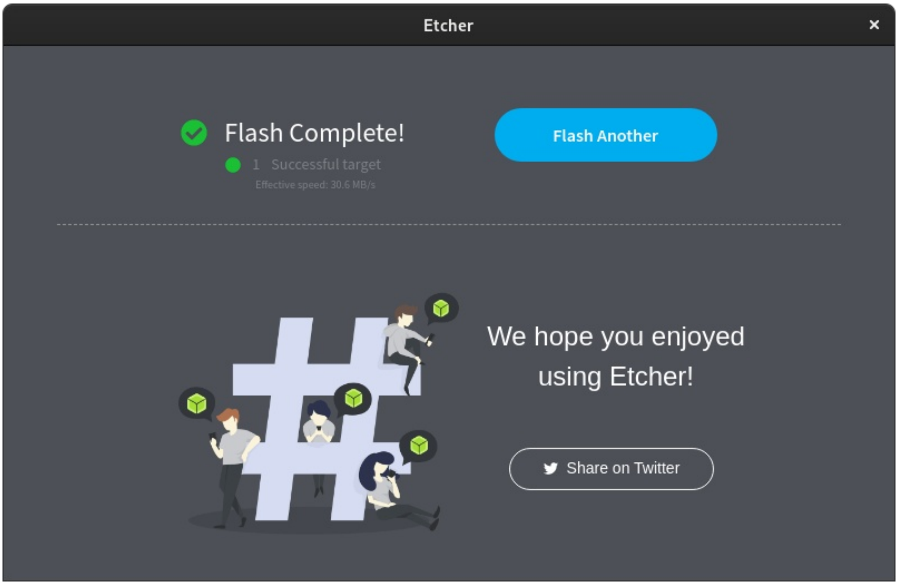
While you are waiting, check out our featured project

Build your own multi-room audio system

Set up a home sound system with Airplay, Spotify and Bluetooth using a fleet of Raspberry Pis and any Hi-Fi or old stereo speakers

5. Success!

If you see **1 Successful target** then this MicroSD card is ready to boot on Orange Pi SBC.



Etcher

Flash Complete!

1 Successful target

Effective speed: 30.6 MB/s

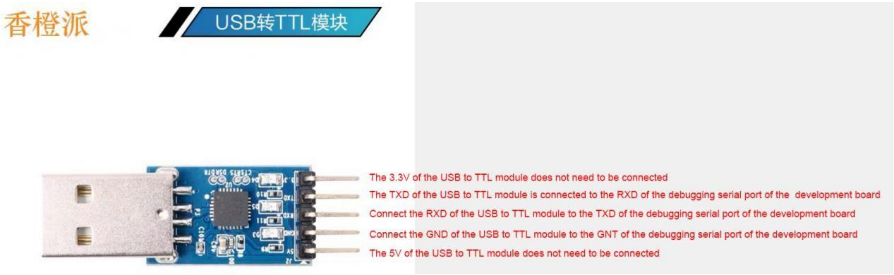
We hope you enjoyed using Etcher!

Appendix B

Connect to the Serial Port.

Connect USB Serial Adapter

First, you need to prepare a USB to TTL module. This module can be bought in Orange Pi store. If there are other similar USB to TTL modules, you can also insert the USB end of the USB to TTL module into the USB port of the computer



The 3.3V of the USB to TTL module does not need to be connected

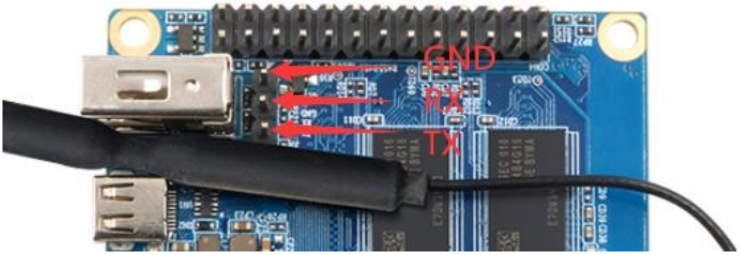
The TXD of the USB to TTL module is connected to the RXD of the debugging serial port of the development board

Connect the RXD of the USB to TTL module to the TXD of the debugging serial port of the development board

Connect the GND of the USB to TTL module to the GNT of the debugging serial port of the development board

The 5V of the USB to TTL module does not need to be connected

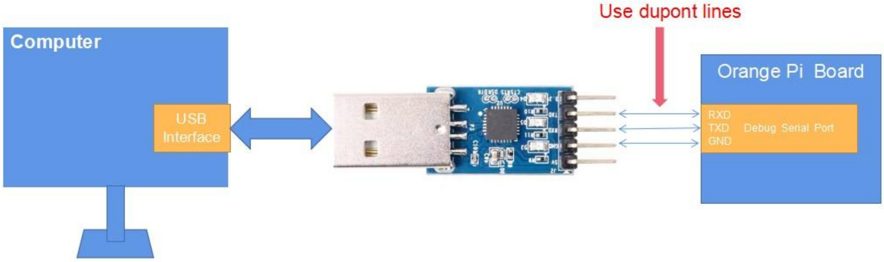
The corresponding relationship between the debug serial port GND, TX, and RX pins of the development board is shown in the figure below



The GND, TX, and RX pins of the USB to TTL module need to be connected to the debug serial port of the development board through a Dupont cable

a. Connect the GND of the USB to TTL module to the GND of the board
b. Connect the RX of the USB to TTL module to the TXD of the board
c. Connect the TX of the USB to TTL module to the RX of the board

The schematic diagram of connecting the USB to TTL module to the computer and the Orange Pi development board is shown below



Schematic diagram of connecting USB to TTL module to computer and Orange Pi development board

How to use the debug serial port on the Ubuntu platform.

1) If the USB to TTL module is connected normally, you can see the corresponding device node name under /dev of Ubuntu PC, remember this node name, you will use it when setting up the serial port software later

```
test@test:~$ ls /dev/ttyUSB*
/dev/ttyUSB0
```

2) Many serial debugging tools that can be used under Linux, such as putty, minicom, etc. The following shows how to use putty

3) First, install putty on Ubuntu PC

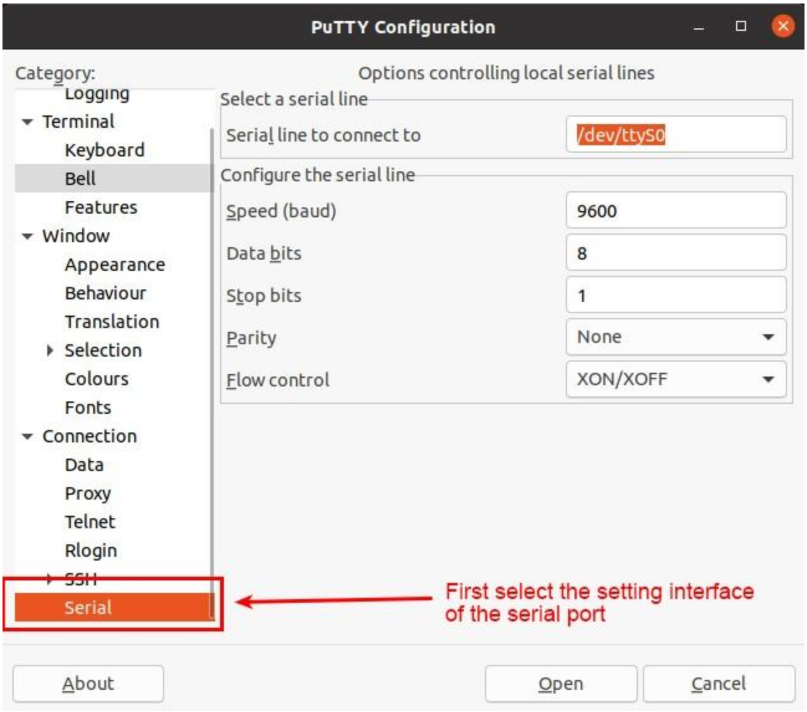
```
test@test:~$ sudo apt update
test@test:~$ sudo apt install putty
```

4) Then run putty, remember to add sudo permissions

```
test@test:~$ sudo putty
```

5) After executing the putty command, the following interface will pop up

6) First, select the setting interface of the serial port



Options controlling local serial lines

Select a serial line

Serial line to connect to: /dev/ttyS0

Configure the serial line

Speed (baud): 9600

Data bits: 8

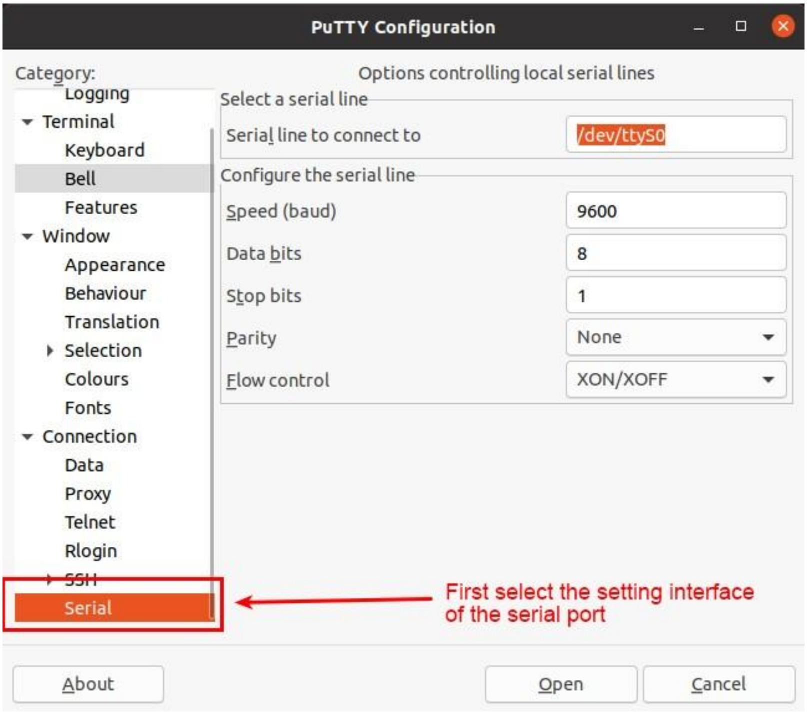
Stop bits: 1

Parity: None

Flow control: XON/XOFF

First select the setting interface of the serial port

7) Then set the parameters of the serial port



Options controlling local serial lines

Select a serial line

Serial line to connect to: /dev/ttyS0

Configure the serial line

Speed (baud): 9600

Data bits: 8

Stop bits: 1

Parity: None

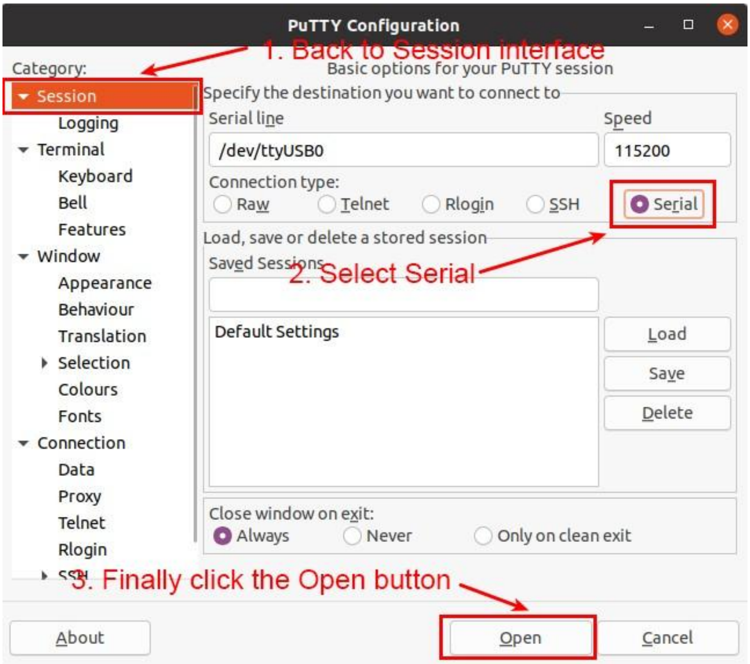
Flow control: XON/XOFF

First select the setting interface of the serial port

8)After setting the serial port setting interface, return to the Session interface

a. First, select the Connection type as Serial

b. Then click the Open button to connect to the serial port



1. Back to Session interface

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: /dev/ttyUSB0

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

2. Select Serial

Load, save or delete a stored session

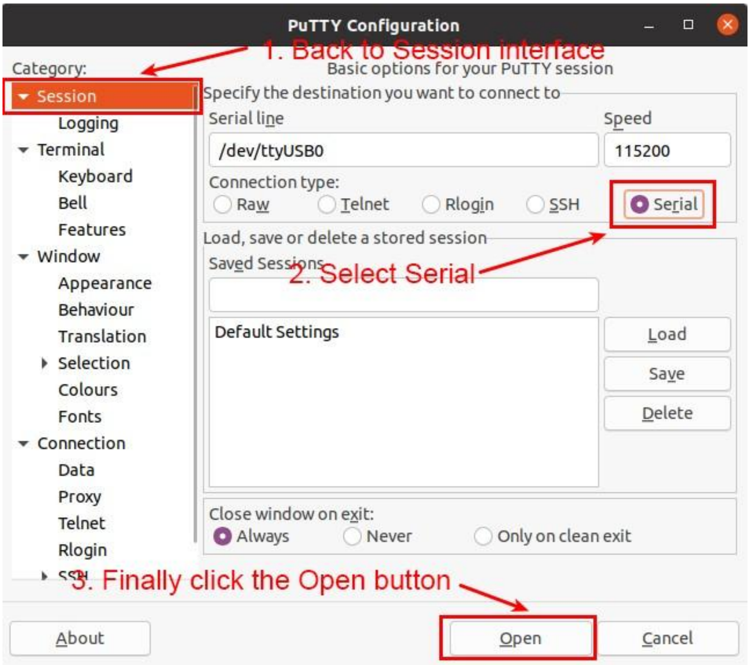
Saved Sessions

Default Settings

Close window on exit: ● Always ○ Never ○ Only on clean exit

3. Finally click the Open button

9)After starting the development board, you can see the Log information output by the OS from the opened serial terminal


1. Back to Session interface

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: /dev/ttyUSB0

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

2. Select Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ● Always ○ Never ○ Only on clean exit

3. Finally click the Open button

How to use the debug serial port on the Windows platform?

1) Many serial debugging tools that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following shows how to use MobaXterm. This software has a free version and can be used without purchasing a serial number.

2) Download MobaXterm

Download MobaXterm from here.

3) After opening the software, the steps to set the serial connection are as follows

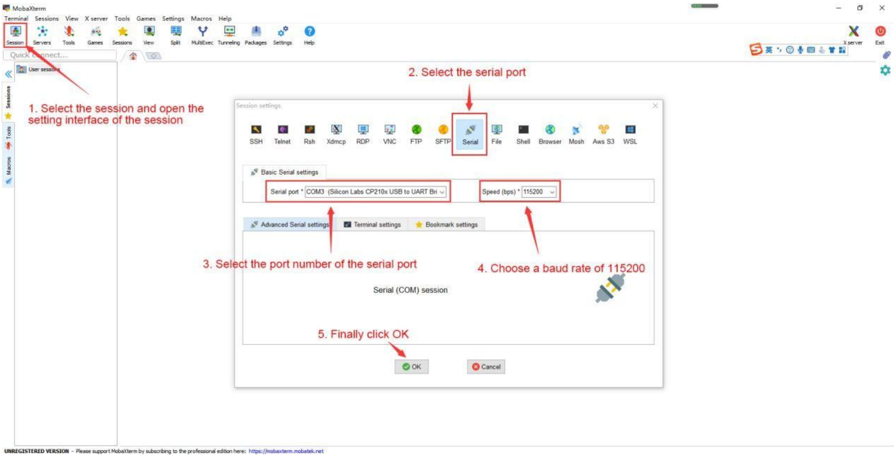
a. Open the session setting interface

b. Select the serial port type

c. Select the port number of the serial port (choose the corresponding port number according to the actual situation), if you cannot see the port number, please use the 360 driver master to scan and install the driver for the USB to TTL serial chip

d. Select the baud rate of the serial port to be 115200

e. Finally click the "OK" button to complete the setting



1. Select the session and open the setting interface of the session

2. Select the serial port

3. Select the port number of the serial port

4. Choose a baud rate of 115200

5. Finally click OK

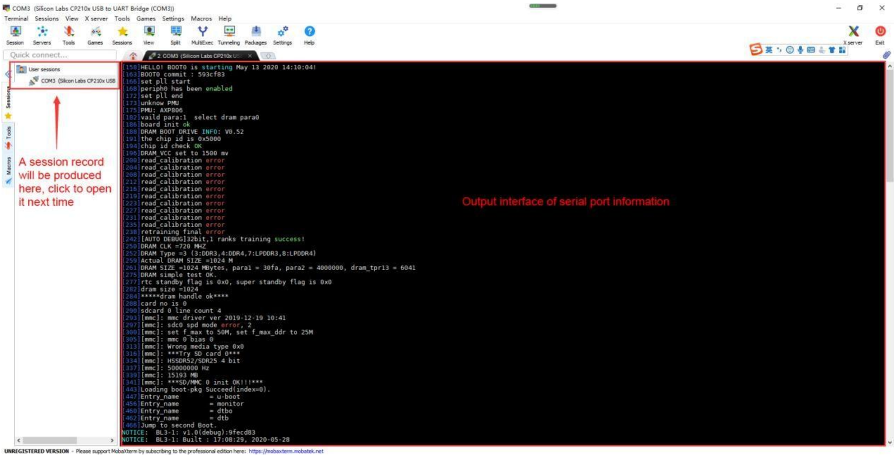
Serial port * COM3 (Silicon Labs CP210x USB to UART Bri

Speed (bps) * 115200

Serial (COM) session

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

4)After clicking the "OK" button, you will enter the following interface, and you can see the output information of the serial port after starting the development board


COM3 (Silicon Labs CP210x USB to UART Bridge (COM3))
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. COM3 (Silicon Labs CP210x US
User sessions
COM3 (Silicon Labs CP210x USB
Sessions
Tools
Macros
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

A session record will be produced here, click to open it next time

Output interface of serial port information

```
[158]HELLO! BOOT0 is starting May 13 2020 14:10:04!
[163]BOOT0 commit : 593cf83
[166]set pll start
[168]periph0 has been enabled
[172]set pll end
[173]unknow PMU
[175]PMU: AXP806
[182]vaild para:1  select dram para0
[186]board init ok
[188]DRAM BOOT DRIVE INFO: V0.52
[191]the chip id is 0x5000
[194]chip id check OK
[196]DRAM_VCC set to 1500 mv
[200]read_calibration error
[204]read_calibration error
[208]read_calibration error
[212]read_calibration error
[216]read_calibration error
[219]read_calibration error
[223]read_calibration error
[227]read_calibration error
[231]read_calibration error
[235]read_calibration error
[238]retraining final error
[242][AUTO DEBUG]32bit,1 ranks training success!
[250]DRAM CLK =720 MHZ
[252]DRAM Type =3 (3:DDR3,4:DDR4,7:LPDDR3,8:LPDDR4)
[259]Actual DRAM SIZE =1024 M
[261]DRAM SIZE =1024 MBytes, para1 = 30fa, para2 = 4000000, dram_tpr13 = 6041
[275]DRAM simple test OK.
[277]rtc standby flag is 0x0, super standby flag is 0x0
[282]dram size =1024
[284]*****dram handle ok****
[288]card no is 0
[290]sdcard 0 line count 4
[293][mmc]: mmc driver ver 2019-12-19 10:41
[297][mmc]: sdc0 spd mode error, 2
[300][mmc]: set f_max to 50M, set f_max_ddr to 25M
[305][mmc]: mmc 0 bias 0
[313][mmc]: Wrong media type 0x0
[316][mmc]: ***Try SD card 0***
[334][mmc]: HSSDR52/SDR25 4 bit
[337][mmc]: 50000000 Hz
[339][mmc]: 15193 MB
[341][mmc]: ***SD/MMC 0 init OK!!!***
[443]Loading boot-pkg Succeed(index=0).
[447]Entry_name        = u-boot
[456]Entry_name        = monitor
[460]Entry_name        = dtbo
[462]Entry_name        = dtb
[466]Jump to second Boot.
NOTICE:  BL3-1: v1.0(debug):9fecd83
NOTICE:  BL3-1: Built : 17:08:29, 2020-05-28
```

Other awesome Orange Pi boards

Orange Pi 4

Orange Pi 4 is a high-performance SBC outfitted with a 6-core RK3399 64-Bit processor, suitable for everyday use and perfect for demanding embedded tasks.

CPU: RK3399 GPU: Mali-T864 eMMC: 16GB
RAM: 4GB DDR4 OS: Linux/Android Wireless: WiFi/BT4.0
Network: 10/100/1000Mbps Ethernet

Orange Pi 4B

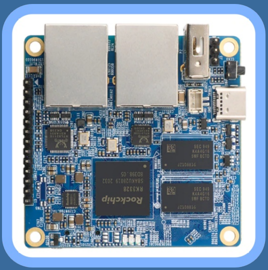
Orange Pi 4B is a high-performance SBC specifically designed for Artificial intelligence tasks, outfitted with a 6-core RK3399 64-Bit processor and a powerful 9.3Tops NPU suitable for AI-related projects and perfect for demanding embedded tasks.

CPU: RK3399 GPU: Mali-T864 NPU: SPR2801S
RAM: 4GB DDR4 OS: Linux/Android eMMC: 16GB
Network: 10/100/1000Mbps Ethernet Wireless: WiFi/BT4.0

Orange Pi 3

Orange Pi 3 is an All-in-One SBC suitable for most embedded needs, comes with plenty of USB-3 ports for peripherals. It comes with a modern PCIe interface for high-speed devices.

CPU: Allwinner H6 GPU: Mali-T720 eMMC: 8GB
RAM: 2GB DDR3 OS: Linux/Android Wireless: WiFi/BT5.0
Network: 10/100/1000Mbps Ethernet



Orange Pi R1+

Orange Pi R1+ is a high-performance IoT SBC with dual Gigabit Ethernet and Quad-core 64bit ARM CPU, suitable for most demanding IoT tasks.

CPU: RK3328 GPU: Mali-450MP2
RAM: 1GB DDR4 OS: Linux/Android/OpenWRT
Network: Dual 10/100/1000Mbps Ethernet

Orange Pi