
Forlinx Embedded RK3588 Documentation

Forlinx Embedded

Sep 02, 2025

HARDWARE

1	HARDWARE	2
1.1	User's Hardware Manual_V1.3	2
2	SOFTWARE	86
2.1	Linux5.10.209_User's Compilation Manual_V1.0	86
2.2	Forlinx Desktop22.04_User's Compilation Manual_V1.0	150
2.3	Android 12.0_User's Manual_V 1.3	207
2.4	Android 12.0_User's Compilation Manual_V 1.0	329
3	APPLICATION NOTE	394
4	Contact Us	395

 Note

To return to the **main Forlinx documentation site**, visit:

[Forlinx Documentation Main Page](#)

Official product overview:

[FET3588-C System On Module](#)

HARDWARE

1.1 User's Hardware Manual_V1.3

Document classification: Top secret Secret Internal information Open

1.1.1 Copyright

The copyright of this manual belongs to Baoding Folinx Embedded Technology Co., Ltd. Without the written permission of our company, no organizations or individuals have the right to copy, distribute, or reproduce any part of this manual in any form, and violators will be held legally responsible.

Forlinx adheres to copyrights of all graphics and texts used in all publications in original or license-free forms.

The drivers and utilities used for the components are subject to the copyrights of the respective manufacturers. The license conditions of the respective manufacturer are to be adhered to. Related license expenses for the operating system and applications should be calculated/declared separately by the related party or its representatives.

1.1.2 Overview

This manual is designed to help users quickly familiarize themselves with the product, understand interface functions and configuration, and primarily discusses the interface functions of the development board, interface introductions, product power consumption, and troubleshooting issues that may arise during use. Some commands were committed to make it easier for you to understand (Adequate and practical for the purpose). For information on pin function multiplexing, hardware troubleshooting methods, etc., please refer to Forlinx's "FET3588 Pin Function Comparison Table".

There are total four chapters:

- Chapter 1. is CPU overview, briefly introducing its performance and applications;
- Chapter 2. is comprehensive introduction to the SoM, including connector pins explanations and function introductions;
- Chapter 3. is comprehensive introduction to the development board, divided into multiple chapters, including both hardware principles and simple design ideas;
- Chapter 4. mainly describes the board's power consumption performance and other considerations.

A description of some of the symbols and formats associated with this manual:

Format	Meaning
	Note or information that requires special attention, be sure to read carefully.
	Relevant notes on the test chapters.
	Indicates the related path.

1.1.3 Revision History

Date	User Manual Version	SoM Version	Carrier Board Version	Revision History
12/01/2023	V1.3	V1.1	V1.1 and above	1. Updating the manual format; 2. Deleting the naming convention and ordering information of the SoM; 3. Deleting the section on SoM pin descriptions (categorized by function); 4. Adding the “Overview” and “Documentation Description” sections; 5. Updating the power consumption table.
24/01/2024	V1.3	V1.1	V1.1 and above	1. Updating the front photo of the development board; 2. Modifying the description of pull - up and pull - down in the boot item section.
23/04/2024	V1.2	V1.1	V1.1 and above	Adding an ESD characteristics section and increasing the description of the ESD characteristics of the SoM pins.
27/04/2024	V1.1	V1.1	V1.1 and above	Correcting the description of the SoM pin functions.
01/12/2023	V1.0	V1.1	V1.1	Initial Version

1.1.4 1. RK3588 Description

The RK3588 is a general-purpose SoC with ARM architecture, integrating quad-core Cortex-A76 and quad-core Cortex-A55 CPU in a typical size core architecture, and the GPU is equipped with G610 MP4 GPU to run complex graphics processing smoothly. The embedded 3D GPU makes the RK3588 fully compatible with OpenGL ES 1.1, 2.0, and 3.2, OpenCL up to 2.2, and Vulkan 1.2. The unique 2D hardware engine with MMU can maximize display performance and provide smooth operation. 6 TOPs NPU empowers various AI scenarios, offering a wide range of possibilities for applications such as local offline AI computing in complex scenarios and sophisticated video stream analysis. Built-in a variety of powerful embedded hardware engines, which supports 8K@60fps H.265 and VP9 decoder, 8K@30fps H.264 and 4K@60fps AV1 decoder, 8K@30fps H.264 and H.265 encoders, high-quality JPEG encoders/decoders, and specialized image pre-processors & post-processors.

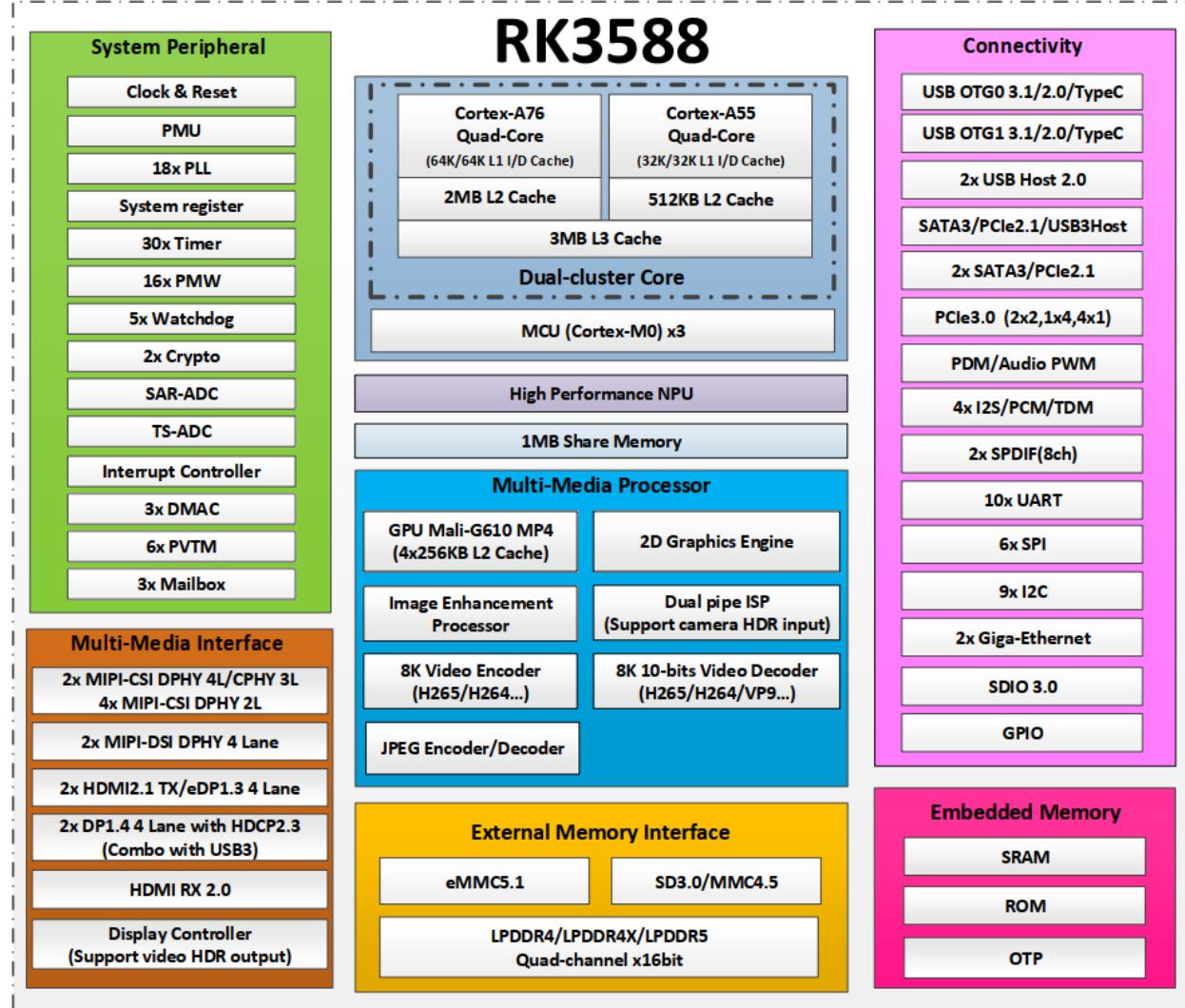
The RK3588 introduces a new generation of fully hardware-based maximum 48-megapixel ISP, which implements many computing accelerators such as HDR, 3A, LSC, 3DNR, 2DNR, sharpening, dehaze, fish-eye correction, gamma correction, etc., and is widely-used in graphics post-processing. The RK3588 integrates a new generation NPU processor of Rockchip, which can support INT4/INT8/INT16/FP16 mixed operation; its powerful compatibility makes it easy to convert a series framework network models based on TensorFlow/ MXNet/PyTorch/Caffe etc. The RK3588 features high-performance, 4-channel external memory connectivity (LPDDR4/LPDDR4X/LPDDR5), which can support harsh memory bandwidths.

Target Applications:

- Information Release Terminals
- Intelligent Cabin
- Smart Screen
- AR/VR
- Edge Computing
- High-end IPC
- Smart NVR
- Premium Pad

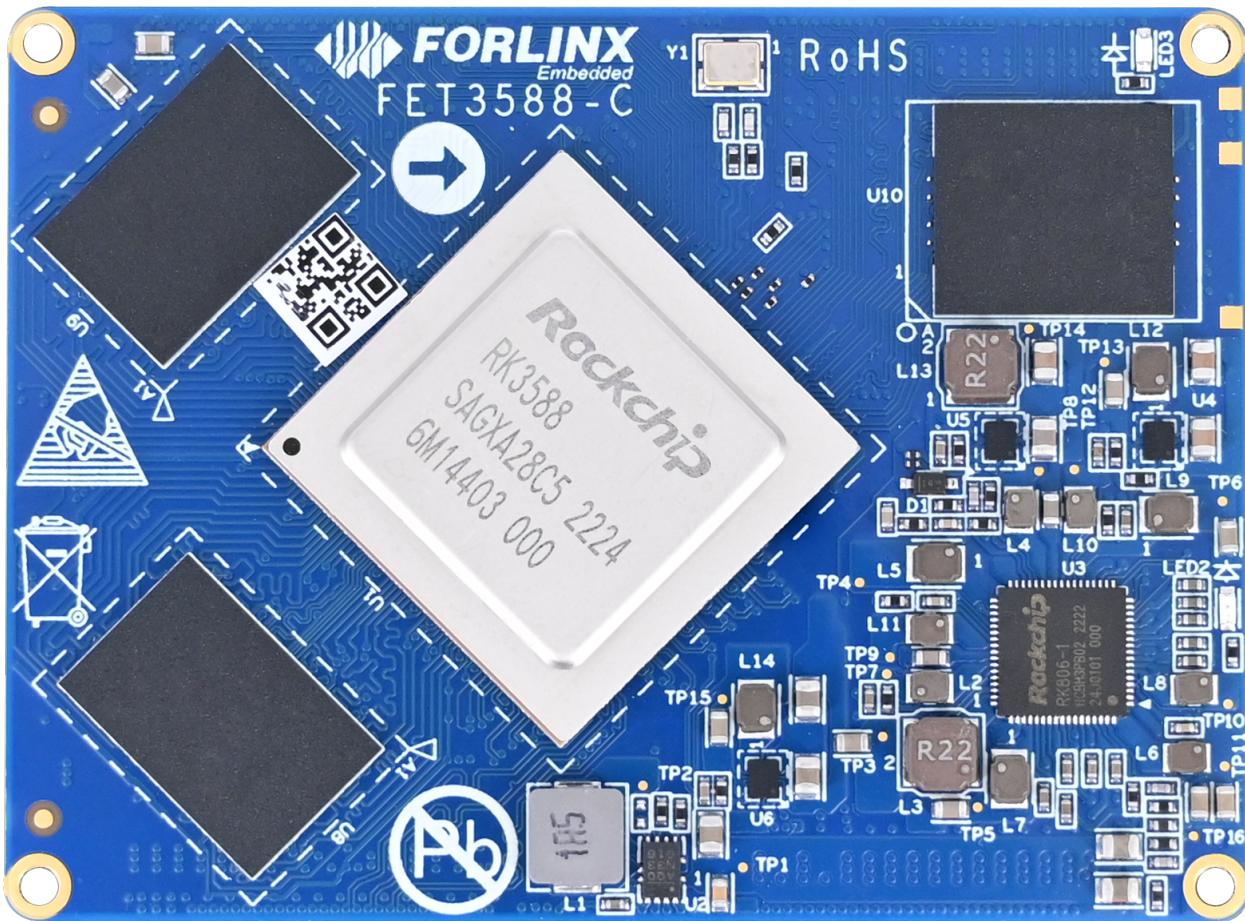
- ARM PC

RK3588 Block Diagram

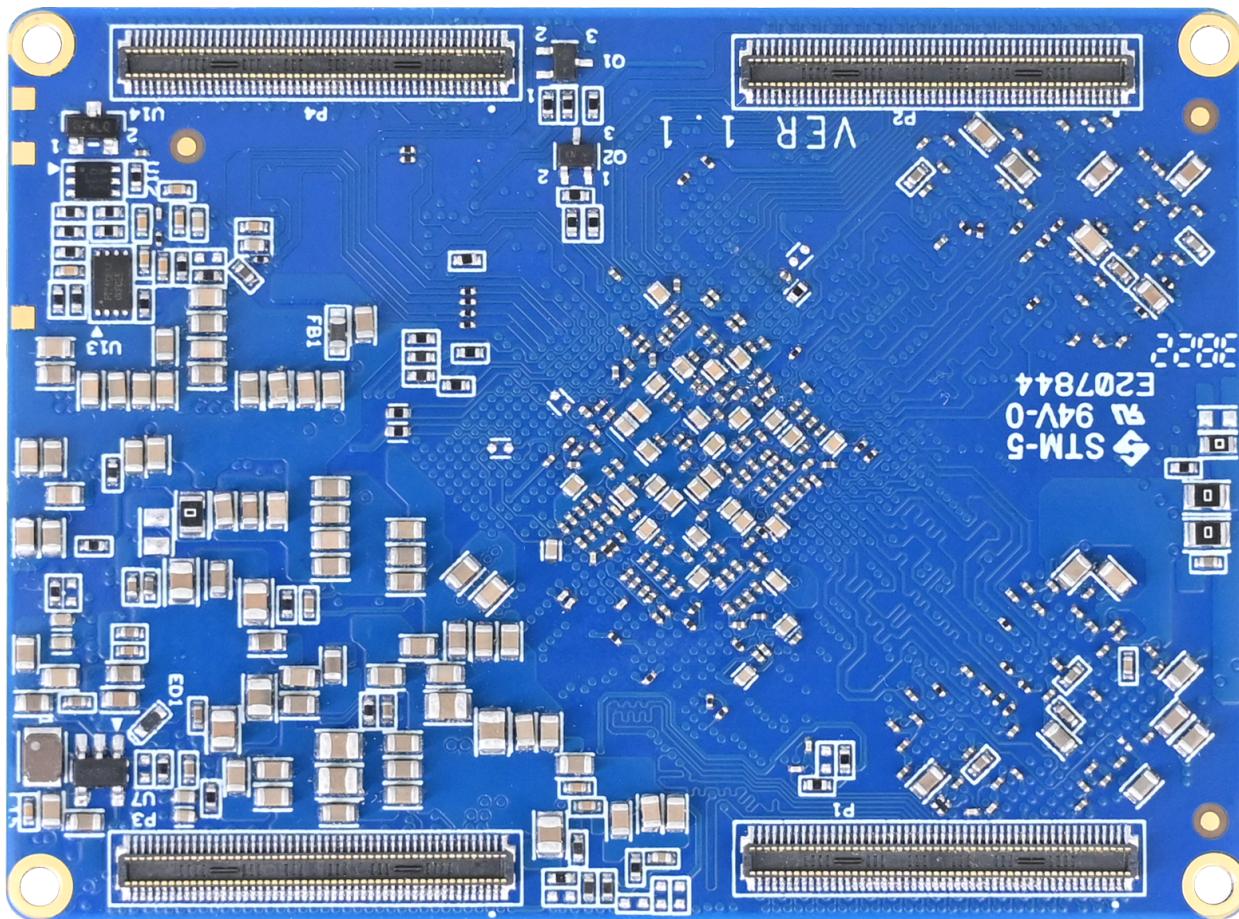


1.1.5 2. FET3588-C SoM Description

2.1 FET3588-C SoM

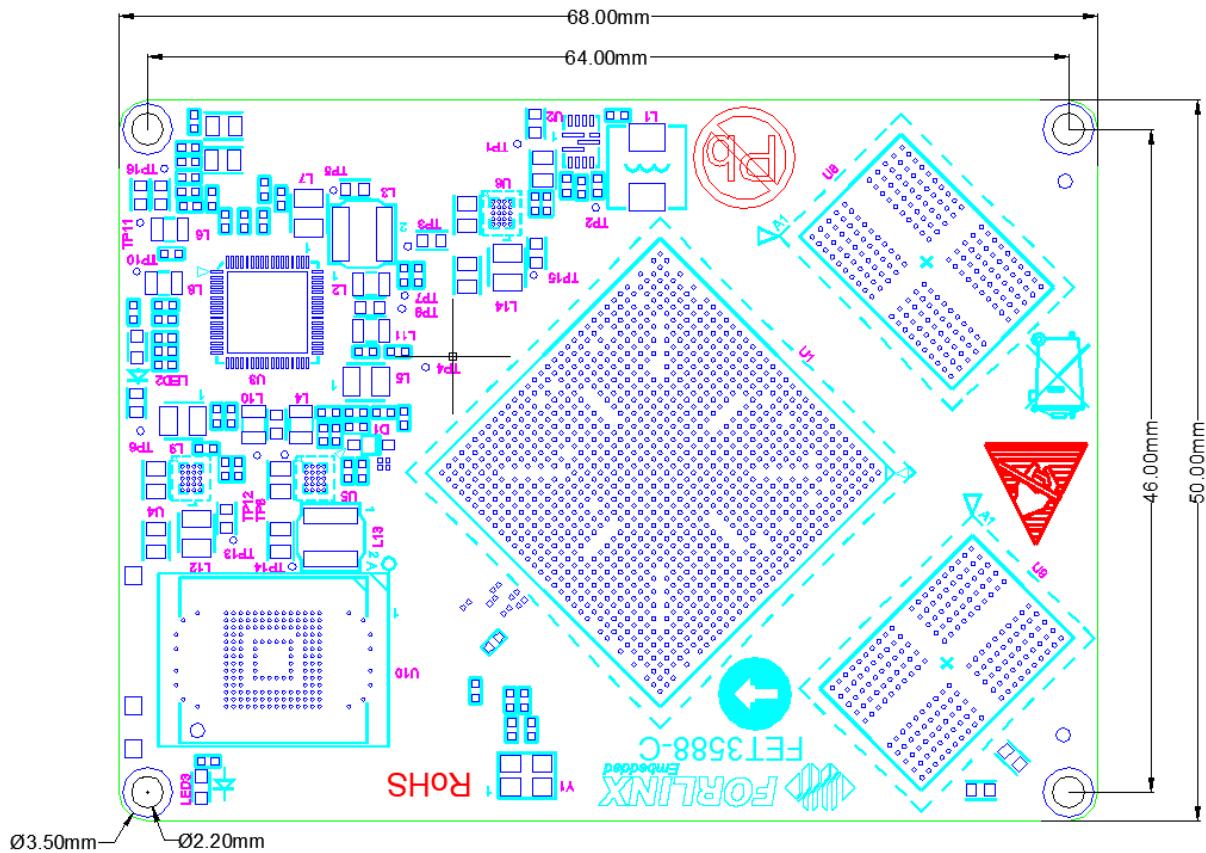


Front

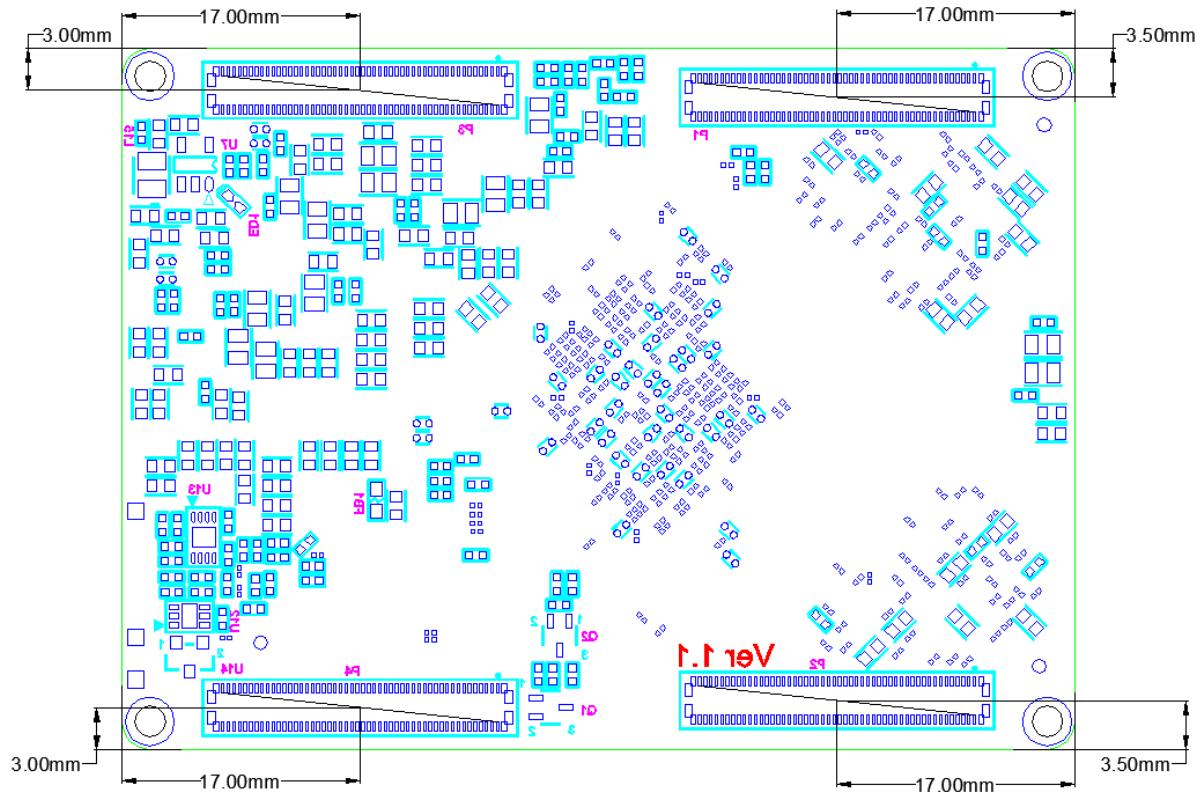


Back

2.2 FET3588-C SoM Dimension Diagram



Top Layer Dimension Diagram



Bottom Layer Dimension Diagram

Structure size: 68mm × 50mm, dimensional tolerance ± 0.15mm, refer to DXF file for more dimensional information.

Plate making process: 1.6mm thickness, 10-layer immersion gold PCB.

Connectors: Four 0.4mm pitch, 100pin board-to-board connectors. See the appendix for the connector dimension drawing.

Four mounting holes with a diameter of 2.2 mm are reserved at the four corners of the core board. For use in a vibrating environment, fixing screws can be installed to improve the reliability of the product connection.

You can refer to the development board design and use M2, L=1.5mm patch nuts on the carrier board.

2.3 Performance Parameters

2.3.1 System Main Frequency

Name	Specification				Description
	Minimum	Typical	Maximum	Unit	
System Frequency Arm® Cortex® -A76	-	-	2400	MHz	-
System Frequency Arm® Cortex® -A55	-	-	1800	MHz	
System Frequency Arm® Cortex® -M0	-	-	-	-	-

2.3.2 Power Parameter

Parameter	Pin Number	Specification			Description
		Minimum	Typical	Maximum	
Main Power Supply Voltage	12V	11.5	12	12.5	V

2.3.3 Operating Environment

Parameter	Specification				Description
	Minimum	Typi- cal	Maxi- mum	Unit	
Operating Temperature	Operating environment	0	25	+80	Commercial level
	Storage Environment	-40	25	+125	
Humidity	Operating Environment	-40	25	+85	Industrial-grade
	Storage Environment	-40	25	+125	
Humidity	Operating Environment	10	-	90	% RH No condensation
	Storage Environment	5	-	95	% RH

2.3.4 SoM Interface Speed

Parameter	Specification				Description
	Minimum	Typical	Maximum	Unit	
Serial Port Communication Speed	-	115200	4M	bps	-
SPI Clock Frequency	-	-	50	MHz	-
I2C Communication Speed	-	100	400	Kbps	-
USB3.0 Interface Speed	-	-	5	Gbps	-
USB2.0 Interface Speed	-	-	480	Mbps	-
CAN Communication Speed	-	-	1	Mbps	-
PCIe2.1	-	-	5	Gbps	-
PCIe3.0	-	-	8	Gbps	-

2.3.5 ESD Features

Parameter	Specification	Unit	Application Scope
	Minimum	Maximum	
ESD HBM(ESDA/JEDEC JS-001-2017)	-2000	2000	V Signals exported from SoM
ESD CDM(ESDA/JEDEC JS-002-2018)	-250	250	V Signals exported from SoM

Note:

1. The above data is provided by Rockchip;
2. As all the signals exported from SoM are electrostatic sensitive signals, the interfaces should be well protected from static electricity in the carrier board design and the SoM transportation, assembling, and use.

2.4 SoM Interface Speed

Function	Quantity	Parameter
MIPI DC PHY(DPHY *1	2	• Supports DPHY or CPHY; • 4-lane MIPI DPHY V2.0, with a maximum data rate of 4.5 Gbps per lane; • 3-lane MIPI CPHY V1.1, with a maximum data rate of 2.5 Gbps per lane;
MIPI CSI DPHY*1	4	• 2-lane MIPI DPHY V1.2, with a maximum data rate of 2.5 Gbps per lane; • Every two 2-lane DPHY can be combined into a single 4-lane DPHY for use with a parallel display interface, supporting a maximum resolution of WUXGA (1920 x 1200@60fps, with a 165 MHz pixel clock).
DVP	1	• 8/10/12/16-bit standard DVP interface, maximum data input of 150MHz; • Support BT.601/BT.656 and BT.1120 VI interface;
HDMI RX	1	• Supports 3.4Gbps~6Gbps HDMI 2.0; • Supports 250Mbps~3.4Gbps HDMI 1.4b; • Supports HDCP2.3 及 HDCP1.4;
HDMI/eDP TX	≤2	• Supports 2 x combined HDMI/eDP TX interfaces (HDMI and eDP cannot operate simultaneously), with each interface supporting x1, x2, and x4 configurations; • HDMI supports a resolution of 7680x4320@60Hz, bandwidths of 3, 6, 8, 10, and 12 Gbps, and HDCP 2.3; • eDP supports a resolution of 4K@60Hz, bandwidths of 1.62 Gbps, 2.7 Gbps, and 5.4 Gbps, and HDCP 1.3;
DP TX	2	• Supports 2 x DP TX 1.4a interfaces, which can be connected to USB 3.1 Gen1 and support 1/2/4 lanes; • Resolution up to 7680x4320@30Hz; • Supports DP Alt Mode under USB Type-C;
MIPI DSI	2	• Supports 2 x MIPI DPHY 2.0 or CPHY 1.1, with a resolution up to 4K@60Hz; • Supports dual MIPI displays in left - right mode and RGB/YUV formats (up to 10 bits);
BT.1120 Output	1	• Supports RGB format (up to 8 bit), with a data speed up to 150 MHz; • Resolution up to 1920x1080@60 Hz;
I2S	≤4	• The sending and receiving clocks can reach up to 50 MHz; • Supports Time - Division Multiplexing (TDM), Inter - IC Sound (I2C), and similar formats; • Supports digital audio interface transmission (SPDIF, IEC60958 - 1, and AES - 3 formats); • Supports an audio reference output clock;
SPDIF	2	• Supports 2 x 16bit audio data storage; • Supports dual - phase stereo output;
PDM	2	• Up to 8 channels, with audio resolution ranging from 16 to 24 bits and a sampling rate of up to 192 KHz; • Supports the PDM master receive mode;
DSM PWM	1	• Convert audio PCM data into direct bitstream digital coding to output 1bit signal data stream, and the output digital signal is filtered to obtain an audio signal;
Ethernet	2	• 2 x GMAC, with led out RGMII / RMII interfaces ; • Supports data transfer rates of 10/100/1000 Mbps;
USB3.1 Gen1*2	3	• The USB3.1 Gen1 data rate can reach up to 5 Gbps; • 2 x USB3.1 OTG are multiplexed with DP TX (USB3OTG_0 and USB3OTG_1). USB3OTG_0 and USB3OTG_1 support USB Type - C and DP Alt; • 1 x USB3.1 Host is multiplexed with PIPE PHY2 (USB3OTG_2);
USB 2.0 Host	2	• Supports 2 x USB2.0 Host;
PCIe 2.0*2	≤3	• Each PCIe 2.1 interface supports 1 lane, with data rate up to 5 Gbps;
PCIe 3.0*2	≤4	• Supports RC and EP, with data rate up to 8 Gbps • Supports four combination modes: 1 lane x4, 2 lanes x2, 4 lanes x1, 1 lane x2 + 2 lanes x1;
SD-MMC	1	• Integration of one SDMMC controller and one SDIO controller, both supporting SDIO 3.0 protocol, and MMC V4.51 protocol;
SDIO	1	
SPI	≤5	• Each controller supports two chip - select output channels; • Supports both serial master and serial slave modes, which can be configured via software;
I2C	≤9	• Supports both 7 - bit and 10 - bit address modes; • The data transfer rate can reach 100 k bits/s in standard mode and up to 400 k bits/s in fast mode;
UART	≤10	• 2 x built - in 64 - bit FIFO, which can be used for TX and RX respectively; • Supports the 5 - bit, 6 - bit, 7 - bit, and 8 - bit serial data sending and receiving, with a baud rate of up to 4 Mbps; • All 10
1.1. User's Hardware Manual V1.2		11
SATA*2	≤3	• 3 x SATA 3.0 controllers, which are multiplexed with PCIe and USB3_HOST2 controllers for PIPE PHY0/1/2; • Supports eSATA, with a maximum data rate of 6 Gbps;
PWM	≤16	• Support up to 16 on-chip PWM with interrupt-based operation and capture mode;

Note: The interface number listed in the table is the hardware design or CPU theoretical maximum quantity, and most of the function pins are multiplexed. Please refer to the PinMux table for easy configuration.

1. Supported MIPI camera combination:

2 MIPI DCPHY + 4 x 2 lanes MIPI CSI DPHY

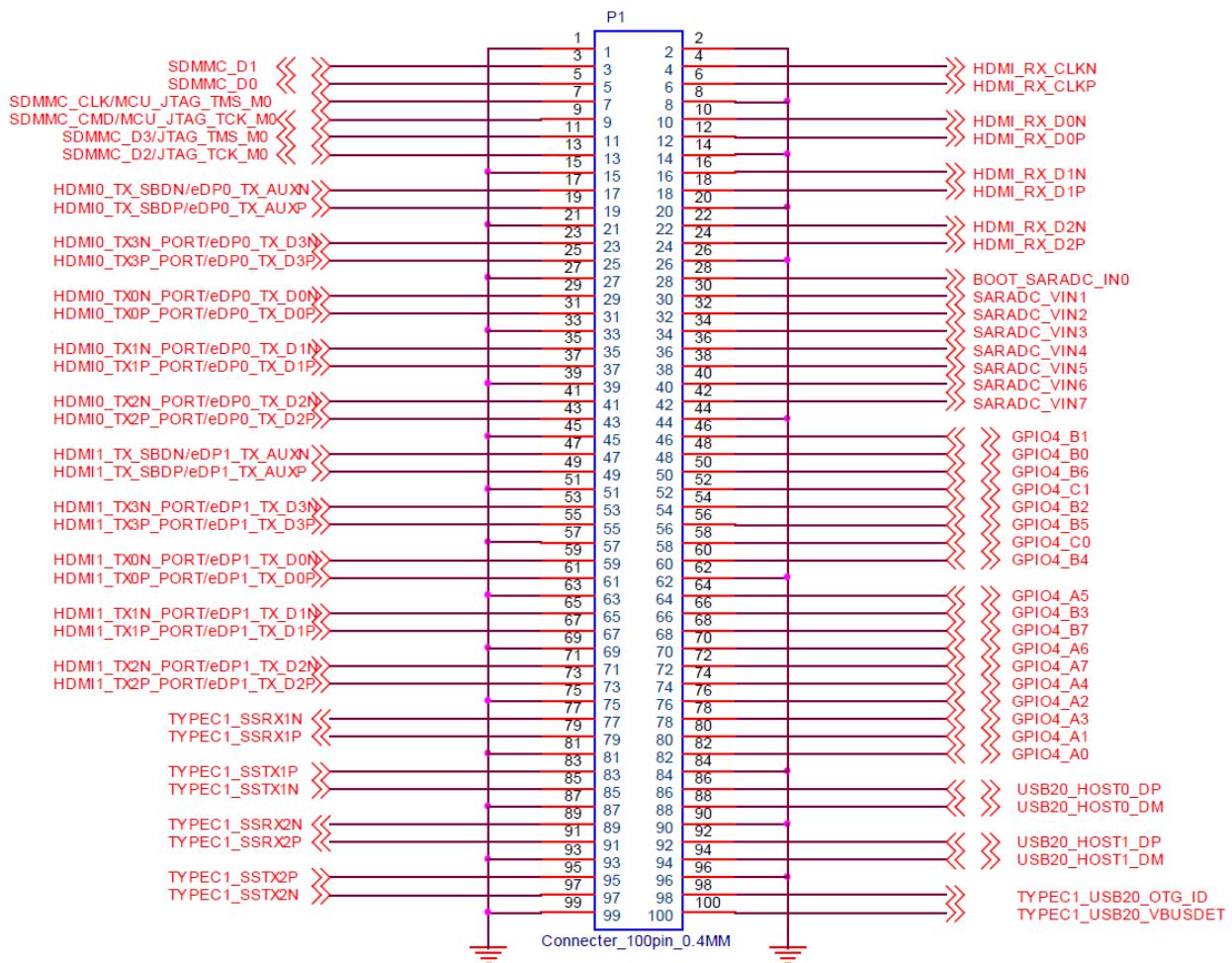
2 MIPI DCPHY + 1 x 4 lanes MIPI CSI DPHY

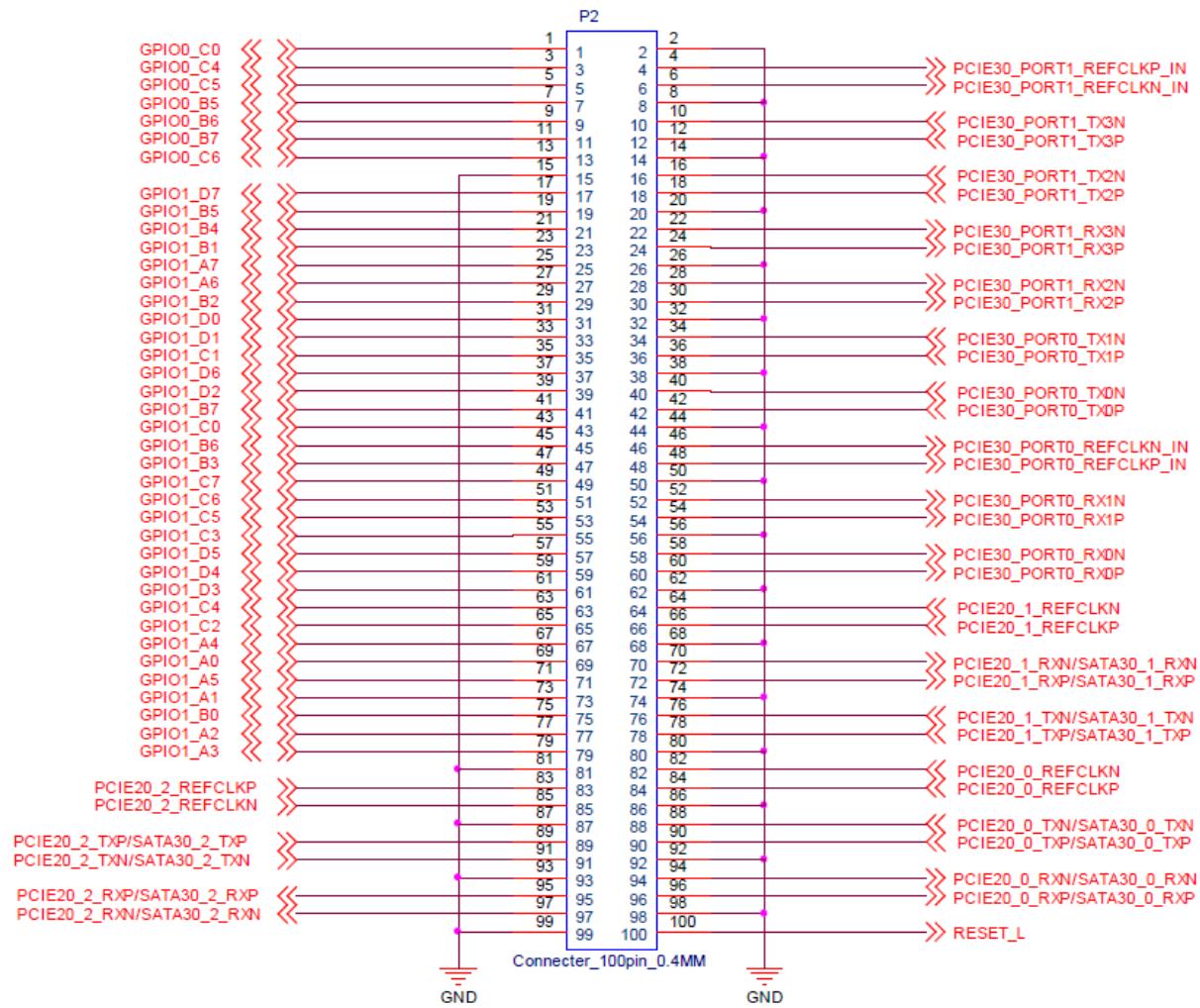
2 MIPI DCPHY + 2 x 4 lanes MIPI CSI DPHY

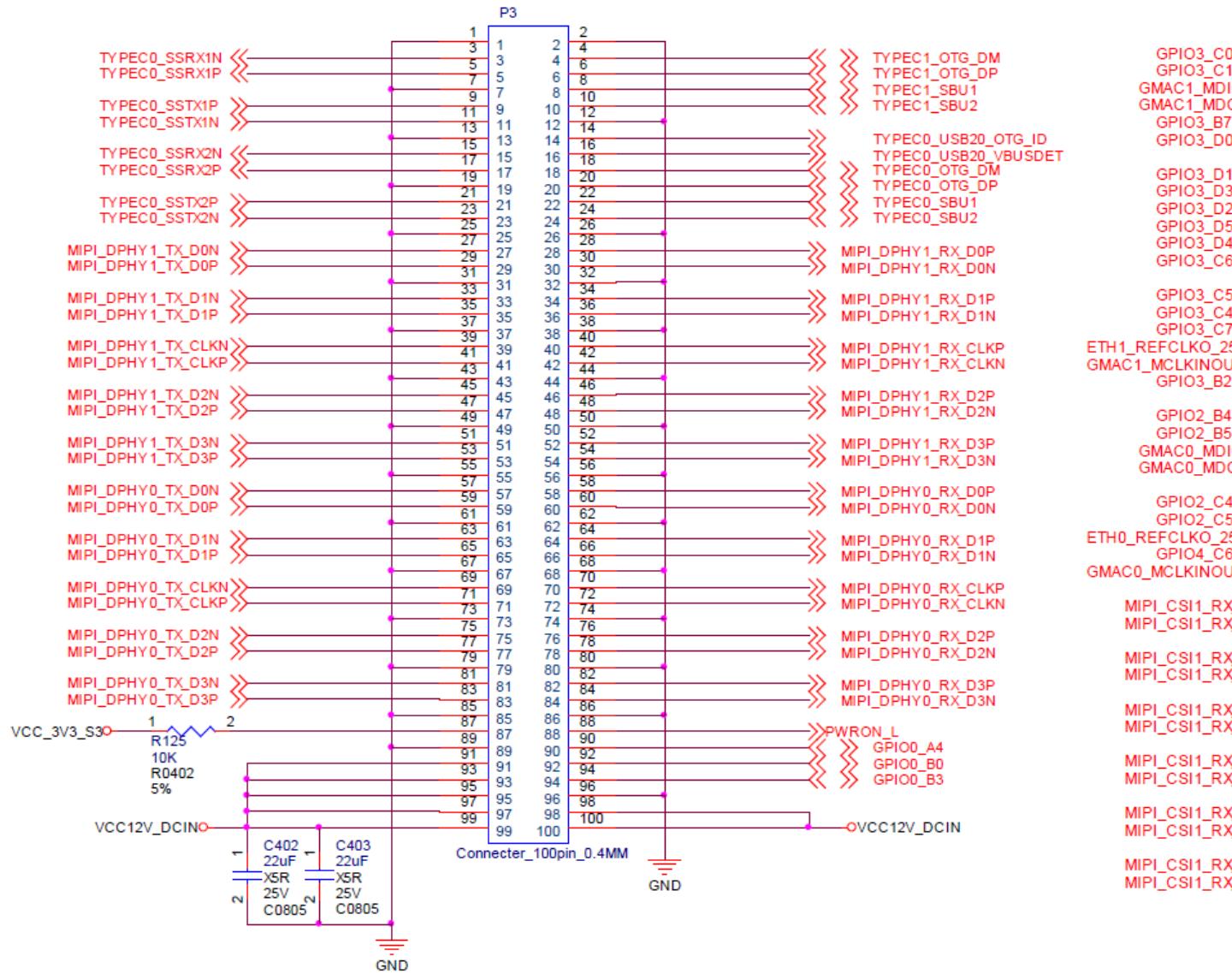
2. USB3.1, PCIe2.0 and SATA 3.0 are multiplexed, please refer to the following carrier board design chapters for more information.

2.5 FET3588-C SoM Pins Definition

2.5.1 FET3588-C SoM Pins Schematic







2.5.2 FET3588-C SoM Pins Description

Note 1:

Num — SoM connector pin no.:

Ball — CPU pin ball no.

GPIO — CPU pin general I/O port serial number

Vol — Pin signal level

Note 2:

Signal Name — SoM connector network name

Pin Description — SoM Pin Signal Descriptions

Default Function — Please don't make any modifications for all SoM pin functions regulated in the “default functions” of the following table, otherwise, it may have conflicts with the factory driver. Please contact us with any questions in time.

Note 3: The pins marked with “Do not use for carrier board” in the “Pin Description” are those used by the SoM, and should not be used in the carrier board design.

Table 1 RIGHT _ DOWN (P1) Connector Interface (Odd) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default I
1	-	GND	-	-	Ground	GND
3	AD1	SDMMC_D1	GPIO4_D1_u	3.3V	SD/MMC Interface data signal 1	SDMMC_
5	AD2	SDMMC_D0	GPIO4_D0_u	3.3V	SD/MMC Interface data signal 0	SDMMC_
7	AE1	SDMMC_CLK/MCU_JTAG_TMS_M0	GPIO4_D5_d	3.3V	SD/MMC Interface clock signal	SDMMC_
9	AE2	SDMMC_CMD/MCU_JTAG_TCK_M0	GPIO4_D4_u	3.3V	SD/MMC Interface order signal	SDMMC_
11	AF1	SDMMC_D3/JTAG_TMS_M0	GPIO4_D3_u	3.3V	SD/MMC Interface data signal 3	SDMMC_
13	AF2	SDMMC_D2/JTAG_TCK_M0	GPIO4_D2_u	3.3V	SD/MMC Interface data signal 2	SDMMC_
15	-	GND	-	-	Ground	GND
17	AG1	HDMI0_TX_SBDN/EDP0_TX_AUXN	-	-	HDMISBD signal-	HDMI0_TX
19	AG2	HDMI0_TX_SBDP/EDP0_TX_AUXP	-	-	HDMISBD signal+	HDMI0_TX
21	-	GND	-	-	Ground	GND
23	AH2	HDMI0_TX3N_PORT/EDP0_TX_D3N	-	-	HDMI differential signal 3-	HDMI_TX
25	AH3	HDMI0_TX3P_PORT/EDP0_TX_D3P	-	-	HDMI differential signal 3+	HDMI_TX
27	-	GND	-	-	Ground	GND
29	AJ1	HDMI0_TX0N_PORT/EDP0_TX_D0N	-	-	HDMI differential signal 0-	HDMI_TX
31	AJ2	HDMI0_TX0P_PORT/EDP0_TX_D0P	-	-	HDMI differential signal 0+	HDMI_TX
33	-	GND	-	-	Ground	GND
35	AK2	HDMI0_TX1N_PORT/EDP0_TX_D1N	-	-	HDMI differential signal 1-	HDMI_TX
37	AK3	HDMI0_TX1P_PORT/EDP0_TX_D1P	-	-	HDMI Differential signal 1+	HDMI_TX
39	-	GND	-	-	Ground	GND
41	AL1	HDMI0_TX2N_PORT/EDP0_TX_D2N	-	-	HDMI differential signal 2-	HDMI_TX
43	AL2	HDMI0_TX2P_PORT/EDP0_TX_D2P	-	-	HDMI differential signal 2+	HDMI_TX
45	-	GND	-	-	Ground	GND
47	AP2	HDMI1_TX_SBDN/EDP1_TX_AUXN	-	-	eDP auxiliary data-	EDP_TX1_
49	AN2	HDMI1_TX_SBDP/EDP1_TX_AUXP	-	-	eDP Auxiliary data +	EDP_TX1_
51	-	GND	-	-	Ground	GND
53	AN3	HDMI1_TX3N_PORT/EDP1_TX_D3N	-	-	eDP Differential signal 3-	EDP_TX1_
55	AM3	HDMI1_TX3P_PORT/EDP1_TX_D3P	-	-	eDP Differential signal 3+	EDP_TX1_
57	-	GND	-	-	Ground	GND
59	AP4	HDMI1_TX0N_PORT/EDP1_TX_D0N	-	-	eDP Differential signal 0-	EDP_TX1_
61	AN4	HDMI1_TX0P_PORT/EDP1_TX_D0P	-	-	eDP Differential signal 0+	EDP_TX1_
63	-	GND	-	-	Ground	GND
65	AN5	HDMI1_TX1N_PORT/EDP1_TX_D1N	-	-	eDP Differential signal 1-	EDP_TX1_
67	AM5	HDMI1_TX1P_PORT/EDP1_TX_D1P	-	-	eDP Differential signal 1+	EDP_TX1_
69	-	GND	-	-	Ground	GND
71	AP6	HDMI1_TX2N_PORT/EDP1_TX_D2N	-	-	eDP Differential signal 2-	EDP_TX1_
73	AN6	HDMI1_TX2P_PORT/EDP1_TX_D2P	-	-	eDP Differential signal 2+	EDP_TX1_
75	-	GND	-	-	Ground	GND
77	AP8	TYPEC1_SS_RX1N	-	-	TYPEC1 Receiving differential signals1-	TYPEC1_S
79	AN8	TYPEC1_SS_RX1P	-	-	TYPEC1 Receiving differential signals 1+	TYPEC1_S
81	-	GND	-	-	Ground	GND
83	AP9	TYPEC1_SS_TX1P	-	-	TYPEC1 Sending differential signals 1+	TYPEC1_S
85	AN9	TYPEC1_SS_TX1N	-	-	TYPEC1 Sending differential signals 1-	TYPEC1_S
87	-	GND	-	-	Ground	GND
89	AP10	TYPEC1_SS_RX2N	-	-	TYPEC1 Receiving differential signals2-	TYPEC1_S
91	AN10	TYPEC1_SS_RX2P	-	-	TYPEC1 Receiving differential signals2+	TYPEC1_S
93	-	GND	-	-	Ground	GND

continues on next page

Table 1 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
95	AP11	TYPEC1_SSTX2P	-	-	TYPEC1 Sending differential signals2+	TYPEC1_SSTX2P
97	AN11	TYPEC1_SSTX2N	-	-	TYPEC1 Sending differential signals2-	TYPEC1_SSTX2N
99	-	GND	-	-	Ground	GND

Table 2 RIGHT _ DOWN (P1) Connector Interface (Even) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
2	-	GND	-	-	Ground	GND
4	AF5	HDMI_RX_CLKN	-	-	HDMI Differential Clock Signals-	HDMI_RX_CLK_N
6	AF6	HDMI_RX_CLKP	-	-	HDMI Differential Clock Signals+	HDMI_RX_CLK_P
8	-	GND	-	-	Ground	GND
10	AG4	HDMI_RX_D0N	-	-	HDMI Receiving differential signals0-	HDMI_RX_D0_N
12	AG5	HDMI_RX_D0P	-	-	HDMI Receiving differential signals 0+	HDMI_RX_D0_P
14	-	GND	-	-	Ground	GND
16	AH5	HDMI_RX_D1N	-	-	HDMI Receiving differential signals1-	HDMI_RX_D1_N
18	AH6	HDMI_RX_D1P	-	-	HDMI Receiving differential signals 1+	HDMI_RX_D1_P
20	-	GND	-	-	Ground	GND
22	AJ4	HDMI_RX_D2N	-	-	HDMI Receiving differential signals2-	HDMI_RX_D2_N
24	AJ5	HDMI_RX_D2P	-	-	HDMI Receiving differential signals 2+	HDMI_RX_D2_P
26	-	GND	-	-	Ground	GND
28	AM16	BOOT_SARADC_IN0	-	1.8V	BOOT start configuration input	BOOT_SARADC_IN0
30	AL16	SARADC_VIN1	-	1.8V	General ADC1	SARADC_VIN1_KEY/R
32	AK16	SARADC_VIN2	-	1.8V	General ADC2	SARADC_VIN2
34	AN17	SARADC_VIN3	-	1.8V	General ADC3	SARADC_VIN3_HP_H
36	AM17	SARADC_VIN4	-	1.8V	General ADC4	SARADC_VIN4
38	AK15	SARADC_VIN5	-	1.8V	General ADC5	SARADC_VIN5
40	AL17	SARADC_VIN6	-	1.8V	General ADC6	SARADC_VIN6
42	AK17	SARADC_VIN7	-	1.8V	General ADC7	SARADC_VIN7
44	-	GND	-	-	Ground	GND
46	AL24	GPIO4_B1	GPIO4_B1_u	3.3V	HDMI0_TX ON signal	HDMI0_TX_ON_H
48	AK26	GPIO4_B0	GPIO4_B0_d	3.3V	TYPEC0_SBU2 signal	TYPEC0_SBU2_DC
50	AJ27	GPIO4_B6	GPIO4_B6_d	3.3V	PCIE30X4 Reset	PCIE30X4_PERSTn_M
52	AK24	GPIO4_C1	GPIO4_C1_d	3.3V	HDMICEC signal	HDMITX0_CEC_M0
54	AK25	GPIO4_B2	GPIO4_B2_u	3.3V	CAN1 data receiving	CAN1_RX_M1
56	AJ26	GPIO4_B5	GPIO4_B5_d	3.3V	PCIE30X4 link insertion detection	PCIE30X4_WAKEEn_M
58	AJ25	GPIO4_C0	GPIO4_C0_u	3.3V	HDMI serial data	HDMITX0_SDA_M0
60	AL26	GPIO4_B4	GPIO4_B4_u	3.3V	PCIE30X4_CLKREQn signal	PCIE30X4_CLKREQn_L
62	-	GND	-	-	Ground	GND
64	AK27	GPIO4_A5	GPIO4_A5_d	3.3V	PCIEx1 Reset	PCIEx1_0_PERSTn_M
66	AM25	GPIO4_B3	GPIO4_B3_u	3.3V	CAN1 data sending	CAN1_TX_M1
68	AJ28	GPIO4_B7	GPIO4_B7_u	3.3V	HDMI Serial clock	HDMITX0_SCL_M0
70	AL27	GPIO4_A6	GPIO4_A6_d	3.3V	I2C5 clock	I2C5_SCL_M2
72	AM27	GPIO4_A7	GPIO4_A7_d	3.3V	I2C5 Data	I2C5_SDA_M2
74	AL28	GPIO4_A4	GPIO4_A4_d	3.3V	PCIEx1 link insertion detection	PCIEx1_0_WAKEEn_M
76	AM29	GPIO4_A2	GPIO4_A2_d	3.3V	TYPEC1_SBU2 signal	TYPEC1_SBU2_DC
78	AL29	GPIO4_A3	GPIO4_A3_d	3.3V	PCIEx1_0_CLKREQn signal	PCIEx1_0_CLKREQn_M
80	AL30	GPIO4_A1	GPIO4_A1_d	3.3V	TYPEC1_SBU1 signal	TYPEC1_SBU1_DC
82	AK30	GPIO4_A0	GPIO4_A0_d	3.3V	TYPEC0_SBU1 signal	TYPEC0_SBU1_DC
84	-	GND	-	-	Ground	GND

continues on n

Table 2 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
86	AK6	USB20_HOST0_DP	-	-	USB20_HOST0 data+	USB20_HOST0_D_P
88	AL6	USB20_HOST0_DM	-	-	USB20_HOST0 data-	USB20_HOST0_D_N
90	-	GND	-	-	Ground	GND
92	AL7	USB20_HOST1_DP	-	-	USB20_HOST1 data+	USB20_HOST1_D_P
94	AM7	USB20_HOST1_DM	-	-	USB20_HOST1 data-	USB20_HOST1_D_N
96	-	GND	-	-	Ground	GND
98	AK8	TYPEC1_USB20_OTG_ID	-	-		x
100	AL8	TYPEC1_USB20_VBUSDET	-	-	TYPEC1_USB20_Insertion detection	TYPEC1_USB20_VBUSDET

Table 3 Left _ DOWN (P2) Connector Interface (Odd) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
1	T31	GPIO0_C0	GPIO0_C0_d	3.3V	I2C2 data	I2C2_SDA_M0
3	R30	GPIO0_C4	GPIO0_C4_d	3.3V	PWM2	PWM2_M0
5	P30	GPIO0_C5	GPIO0_C5_u	3.3V	PWM4	PWM4_M0
7	P29	GPIO0_B5	GPIO0_B5_d	3.3V	UART2 sending	UART2_TX_M0_DEBU
9	R29	GPIO0_B6	GPIO0_B6_d	3.3V	UART2 receiving	UART2_RX_M0_DEBU
11	T28	GPIO0_B7	GPIO0_B7_d	3.3V	I2C2 clock	I2C2_SCL_M0
13	T29	GPIO0_C6	GPIO0_C6_u	3.3V	PWM5	PWM5_M1
15	-	GND	-	-	Ground	GND
17	F25	GPIO1_D7	GPIO1_D7_u	1.8V	HDMI serial data	HDMI_RX_SDA_M2
19	E25	GPIO1_B5	GPIO1_B5_u	1.8V	GMAC1 Interrupt	GMAC1_INT
21	E24	GPIO1_B4	GPIO1_B4_u	1.8V	GMAC1 Reset	GMAC1_RESET
23	D25	GPIO1_B1	GPIO1_B1_d	1.8V	BT Link activation	BT_WAKE_HOST(1.8V)
25	C25	GPIO1_A7	GPIO1_A7_u	1.8V	WLAN Link activation	WLAN WAKE_1.8V_IN
27	C24	GPIO1_A6	GPIO1_A6_d	1.8V	BT Link activation	BT_WAKE_1.8V_IN
29	D26	GPIO1_B2	GPIO1_B2_d	1.8V	Headphone insertion detection	HP_DET_L
31	F26	GPIO1_D0	GPIO1_D0_d	1.8V	I2C7 clock	I2C7_SCL_M0
33	F27	GPIO1_D1	GPIO1_D1_d	1.8V	I2C7 data	I2C7_SDA_M0
35	G27	GPIO1_C1	GPIO1_C1_z	1.8V	I2C3 clock	I2C3_SCL_M0
37	F24	GPIO1_D6	GPIO1_D6_u	1.8V	HDMI Serial clock	HDMI_RX_SCL_M2
39	F28	GPIO1_D2	GPIO1_D2_d	1.8V	UART4 Sending data	UART4_TX_M0
41	E27	GPIO1_B7	GPIO1_B7_u	1.8V	HDMI_RXCEC signal	HDMI_RX_CEC_M2
43	G29	GPIO1_C0	GPIO1_C0_z	1.8V	I2C3 data	I2C3_SDA_M0
45	E26	GPIO1_B6	GPIO1_B6_d	1.8V	HDMI Receiving link detection	HDMI_RX_HPDOUT_M
47	D27	GPIO1_B3	PIO1_B3_d	1.8V	TYPEC1 Interrupt	TYPEC1_INT
49	E29	GPIO1_C7	GPIO1_C7_d	1.8V	I2S Data output	I2S0_SD00
51	D29	GPIO1_C6	GPIO1_C6_d	1.8V	4G/5G Reset	4G/5G RESET
53	D30	GPIO1_C5	GPIO1_C5_d	1.8V	I2S Sending frame clock	I2S0_LRCK_TX
55	E31	GPIO1_C3	GPIO1_C3_d	1.8V	I2S bit clock	I2S0_SCLK_TX
57	G26	GPIO1_D5	GPIO1_D5_d	1.8V	HDMIIRX insertion detection	HDMIIRX_DET_L
59	D28	GPIO1_D4	GPIO1_D4_d	1.8V	I2S data input	I2S0_SD10
61	E28	GPIO1_D3	GPIO1_D3_d	1.8V	UART4 receiving data:	UART4_RX_M0
63	E30	GPIO1_C4	GPIO1_C4_d	1.8V	4G/5G Reset	4G/5G_RESET_1V8
65	F30	GPIO1_C2	GPIO1_C2_d	1.8V	I2S main clock	I2S0_MCLK
67	B25	GPIO1_A4	GPIO1_A4_d	1.8V	IIC interrupt	IIC_GPIO_INT
69	A24	GPIO1_A0	GPIO1_A0_d	1.8V	UART6 receiving data:	UART6_RX_M1
71	B26	GPIO1_A5	GPIO1_A5_d	1.8V	HDMI Sending link detection	HDMITX0_HPDIN_M0
73	A25	GPIO1_A1	GPIO1_A1_d	1.8V	UART6 Sending data	UART6_TX_M1

continues on next page

Table 3 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
75	C27	GPIO1_B0	GPIO1_B0_u	1.8V	TYPEC0 Interrupt	TYPEC0_INT
77	A26	GPIO1_A2	GPIO1_A2_d	1.8V	UART6 request sending	UART6_RTSN_M1
79	A27	GPIO1_A3	GPIO1_A3_d	1.8V	UART6 clear sending	UART6_CTSN_M1
81	-	GND	-	-	Ground	GND
83	G31	PCIE20_2_REFCLKP	-	-		x
85	G30	PCIE20_2_REFCLKN	-	-		x
87	-	GND	-	-	Ground	GND
89	H30	PCIE20_2_TXP/SATA30_2_TXP	-	-	USB30_2 sending differential+	USB30_2_SSTX_P
91	H29	PCIE20_2_RXN/SATA30_2_RXN	-	-	USB30_2 sending differential-	USB30_2_SSTX_N
93	-	GND	-	-	Ground	GND
95	J31	PCIE20_2_RXP/SATA30_2_RXP	-	-	USB30_2 receiving differential+	USB30_2_SSRX_P
97	J30	PCIE20_2_RXN/SATA30_2_RXN	-	-	USB30_2 receiving differential-	USB30_2_SSRX_N
99	-	GND	-	-	Ground	GND

Table 4 Left _ DOWN (P2) Connector Interface (Even) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
2	-	GND	-	-	Ground	GND
4	A28	PCIE30_PORT1_REFCLKP_IN	-	-	PCIE3.0 clock input+	PCIE30_PORT1_REFCLK_IN_P
6	B28	PCIE30_PORT1_REFCLKN_IN	-	-	PCIE3.0 clock input-	PCIE30_PORT1_REFCLK_IN_N
8	-	GND	-	-	Ground	GND
10	B29	PCIE30_PORT1_TX3N	-	-	PCIE3.0 sending data 3-	PCIE30_PORT1_TX3_N
12	C29	PCIE30_PORT1_TX3P	-	-	PCIE3.0 data sending 3+	PCIE30_PORT1_TX3_P
14	-	GND	-	-	Ground	GND
16	A30	PCIE30_PORT1_TX2N	-	-	PCIE3.0 sending data 2-	PCIE30_PORT1_TX2_N
18	B30	PCIE30_PORT1_TX2P	-	-	PCIE3.0 data sending 2+	PCIE30_PORT1_TX2_P
20	-	GND	-	-	Ground	GND
22	B31	PCIE30_PORT1_RX3N	-	-	PCIE3.0 data receiving 3-	PCIE30_PORT1_RX3_N
24	C31	PCIE30_PORT1_RX3P	-	-	PCIE3.0 data receiving 3+	PCIE30_PORT1_RX3_P
26	-	GND	-	-	Ground	GND
28	A32	PCIE30_PORT1_RX2N	-	-	PCIE3.0 data receiving-	PCIE30_PORT1_RX2_N
30	B32	PCIE30_PORT1_RX2P	-	-	PCIE3.0 data receiving 2+	PCIE30_PORT1_RX2_P
32	-	GND	-	-	Ground	GND
34	C34	PCIE30_PORT0_TX1N	-	-	PCIE3.0 sending data 1-	PCIE30_PORT0_TX1_N
36	C33	PCIE30_PORT0_TX1P	-	-	PCIE3.0 data sending 1+	PCIE30_PORT0_TX1_P
38	-	GND	-	-	Ground	GND
40	D33	PCIE30_PORT0_RX0N	-	-	PCIE3.0 sending data 0-	PCIE30_PORT0_RX0_N
42	D32	PCIE30_PORT0_RX0P	-	-	PCIE3.0 data sending 0+	PCIE30_PORT0_RX0_P
44	-	GND	-	-	Ground	GND
46	E34	PCIE30_PORT0_REFCLKN_IN	-	-	PCIE3.0 clock input-	PCIE30_PORT0_REFCLK_IN_N
48	E33	PCIE30_PORT0_REFCLKP_IN	-	-	PCIE3.0 clock input+	PCIE30_PORT0_REFCLK_IN_P
50	-	GND	-	-	Ground	GND
52	F33	PCIE30_PORT0_RX1N	-	-	PCIE3.0 data receiving 1-	PCIE30_PORT0_RX1_N
54	F32	PCIE30_PORT0_RX1P	-	-	PCIE3.0 data receiving 1+	PCIE30_PORT0_RX1_P
56	-	GND	-	-	Ground	GND
58	G34	PCIE30_PORT0_RX0N	-	-	PCIE3.0 data receiving 0-	PCIE30_PORT0_RX0_N
60	G33	PCIE30_PORT0_RX0P	-	-	PCIE3.0 data receiving 0+	PCIE30_PORT0_RX0_P
62	-	GND	-	-	Ground	GND
64	H33	PCIE20_1_REFCLKN	-	-	PCIE2.0 clock input-	PCIE20_1_REFCLK_N

continues on next page

Table 4 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
66	H32	PCIE20_1_REFCLKP	-	-	PCIE2.0 clock input+	PCIE20_1_REFCLK_P
68	-	GND	-	-	Ground	GND
70	J34	PCIE20_1_RXN/SATA30_1_RXN	-	-	PCIE2.0 data receiving-	PCIE20_1_RX_N
72	J33	PCIE20_1_RXP/SATA30_1_RXP	-	-	PCIE2.0 data receiving+	PCIE20_1_RX_P
74	-	GND	-	-	Ground	GND
76	K34	PCIE20_1_TXN/SATA30_1_TXN	-	-	PCIE2.0 data sending-	PCIE20_1_TX_N
78	K33	PCIE20_1_TXP/SATA30_1_TXP	-	-	PCIE2.0 data sending+	PCIE20_1_TX_P
80	-	GND	-	-	Ground	GND
82	L33	PCIE20_0_REFCLKN	-	-	PCIE2.0 clock input-	PCIE20_0_REFCLK_N
84	L32	PCIE20_0_REFCLKP	-	-	PCIE2.0 clock input+	PCIE20_0_REFCLK_P
86	-	GND	-	-	Ground	GND
88	M33	PCIE20_0_RXN/SATA30_0_RXN	-	-	PCIE2.0 data sending-	PCIE20_0_RX_N
90	M34	PCIE20_0_RXP/SATA30_0_RXP	-	-	PCIE2.0 data sending+	PCIE20_0_RX_P
92	-	GND	-	-	Ground	GND
94	N34	PCIE20_0_RXN/SATA30_0_RXN	-	-	PCIE2.0 data receiving-	PCIE20_0_RX_N
96	N33	PCIE20_0_RXP/SATA30_0_RXP	-	-	PCIE2.0 data receiving+	PCIE20_0_RX_P
98	-	GND	-	-	Ground	GND
100		RESET_L	-	-	Reset	RESET_L

Table 5 Right_UP (P3) Connector Interface(Odd) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
1	-	GND	-	-	Ground	GND
3	AP13	TYPECO_SSRX1N	-	-	TYPECO Receiving differential signals1-	TYPECO_SSRX1_N
5	AN13	TYPECO_SSRX1P	-	-	TYPECO Receiving differential signals 1+	TYPECO_SSRX1_P
7	-	GND	-	-	Ground	GND
9	AP14	TYPECO_SSTX1P	-	-	TYPECO Sending differential signals 1+	TYPECO_SSTX1_P
11	AN14	TYPECO_SSTX1N	-	-	TYPECO Sending differential signals 1-	TYPECO_SSTX1_N
13	-	GND	-	-	Ground	GND
15	AP15	TYPECO_SSRX2N	-	-	TYPECO Receiving differential signals2-	TYPECO_SSRX2_N
17	AN15	TYPECO_SSRX2P	-	-	TYPECO Receiving differential signals2+	TYPECO_SSRX2_P
19	-	GND	-	-	Ground	GND
21	AP16	TYPECO_SSTX2P	-	-	TYPECO Sending differential signals2+	TYPECO_SSTX2_P
23	AN16	TYPECO_SSTX2N	-	-	TYPECO Sending differential signals2-	TYPECO_SSTX2_N
25	-	GND	-	-	Ground	GND
27	AP18	MIPI_DPHY1_TX_D0N	-	-	MIPI_DPHY 1 data sending 0-	MIPI_DPHY1_TXD0_N
29	AN18	MIPI_DPHY1_TX_D0P	-	-	MIPI_DPHY 1 data sending 0+	MIPI_DPHY1_TXD0_P
31	-	GND	-	-	Ground	GND
33	AP19	MIPI_DPHY1_TX_D1N	-	-	MIPI_DPHY 1 data sending 1-	MIPI_DPHY1_TXD1_N
35	AN19	MIPI_DPHY1_TX_D1P	-	-	MIPI_DPHY 1 data sending 1+	MIPI_DPHY1_TXD1_P
37	-	GND	-	-	Ground	GND
39	AP20	MIPI_DPHY1_TX_CLKN	-	-	MIPI_DPHY 1 clock sending-	MIPI_DPHY1_TXCLK_N
41	AN20	MIPI_DPHY1_TX_CLKP	-	-	MIPI_DPHY 1 clock sending+	MIPI_DPHY1_TXCLK_P
43		GND	-	-	Ground	GND
45	AP21	MIPI_DPHY1_TX_D2N	-	-	MIPI_DPHY 1 data sending 2-	MIPI_DPHY1_TXD2_N
47	AN21	MIPI_DPHY1_TX_D2P	-	-	MIPI_DPHY 1 data sending 2+	MIPI_DPHY1_TXD2_P
49	-	GND	-	-	Ground	GND
51	AP22	MIPI_DPHY1_TX_D3N	-	-	MIPI_DPHY 1 data sending 3-	MIPI_DPHY1_TXD3_N
53	AN22	MIPI_DPHY1_TX_D3P	-	-	MIPI_DPHY 1 data sending 3+	MIPI_DPHY1_TXD3_P

continues on next page

Table 5 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
55	-	GND	-	-	Ground	GND
57	AP24	MIPI_DPHY0_TX_D0N	-	-	MIPI_DPHY 0 data sending 0-	MIPI_DPHY0_RXD0_N
59	AN24	MIPI_DPHY0_TX_D0P	-	-	MIPI_DPHY 0 data sending 0+	MIPI_DPHY0_RXD0_P
61	-	GND	-	-	Ground	GND
63	AP25	MIPI_DPHY0_TX_D1N	-	-	MIPI_DPHY 0 data sending 1-	MIPI_DPHY0_RXD1_N
65	AN25	MIPI_DPHY0_TX_D1P	-	-	MIPI_DPHY 0 data sending 1+	MIPI_DPHY0_RXD1_P
67	-	GND	-	-	Ground	GND
69	AP26	MIPI_DPHY0_TX_CLKN	-	-	MIPI_DPHY 0 clock sending-	MIPI_DPHY0_RXCLK_N
71	AN26	MIPI_DPHY0_TX_CLKP	-	-	MIPI_DPHY 0 clock sending+	MIPI_DPHY0_RXCLK_P
73	-	GND	-	-	Ground	GND
75	AP27	MIPI_DPHY0_TX_D2N	-	-	MIPI_DPHY 0 data sending 2-	MIPI_DPHY0_RXD2_N
77	AN27	MIPI_DPHY0_TX_D2P	-	-	MIPI_DPHY 0 data sending 2+	MIPI_DPHY0_RXD2_P
79	-	GND	-	-	Ground	GND
81	AP28	MIPI_DPHY0_TX_D3N	-	-	MIPI_DPHY 0 data sending 3-	MIPI_DPHY0_RXD3_N
83	AN28	MIPI_DPHY0_TX_D3P	-	-	MIPI_DPHY 0 data sending 3+	MIPI_DPHY0_RXD3_P
85	-	GND	-	-	Ground	GND
87		CARRIER_BOARD_EN	-	-	CARRIER enable	CARRIER_BOARD_EN
89	-	GND	-	-	Ground	GND
91		VCC12V_DCIN	-	-	12V power input	VCC12V_DCIN
93		VCC12V_DCIN	-	-	12V power input	VCC12V_DCIN
95		VCC12V_DCIN	-	-	12V power input	VCC12V_DCIN
97		VCC12V_DCIN	-	-	12V power input	VCC12V_DCIN
99		VCC12V_DCIN	-	-	12V power input	VCC12V_DCIN

Table 6 Right_UP (P3) Connector Interface(Even) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
2	-	GND	-	-	Ground	GND
4	AL9	TYPEC1_OTG_DM	-	-	TYPEC1 data-	TYPEC1_OTG_D_N
6	AK9	TYPEC1_OTG_DP	-	-	TYPEC1 data+	TYPEC1_OTG_D_P
8	AL10	TYPEC1_SBU1	-	-	TYPEC1_SBU1 signal	TYPEC1_SBU1
10	AM10	TYPEC1_SBU2	-	-	TYPEC1_SBU2 signal	TYPEC1_SBU2
12	-	GND	-	-	Ground	GND
14	AL14	TYPEC0_USB20_OTG_ID	-	-		X
16	AM14	TYPEC0_USB20_VBUSDET	-	-	TYPEC0_USB20 insertion detection	TYPEC0_USB20_VBUSDET
18	AM12	TYPEC0_OTG_DM	-	-	TYPEC0 data-	TYPEC0_OTG_D_N
20	AL12	TYPEC0_OTG_DP	-	-	TYPEC0 data+	TYPEC0_OTG_D_P
22	AL15	TYPEC0_SBU1	-	-	TYPEC0_SBU1 signal	TYPEC0_SBU1
24	AM15	TYPEC0_SBU2	-	-	TYPEC0_SBU2 signal	TYPEC0_SBU2
26	-	GND	-	-	Ground	GND
28	AK18	MIPI_DPHY1_RX_D0P	-	-	MIPI_DPHY 1 data receiving 0+	MIPI_DPHY1_RXD0_P
30	AL18	MIPI_DPHY1_RX_D0N	-	-	MIPI_DPHY 1 data receiving 0-	MIPI_DPHY1_RXD0_N
32	-	GND	-	-	Ground	GND
34	AK19	MIPI_DPHY1_RX_D1P	-	-	MIPI_DPHY 1 data receiving 1+	MIPI_DPHY1_RXD1_P
36	AL19	MIPI_DPHY1_RX_D1N	-	-	MIPI_DPHY 1 data receiving 1-	MIPI_DPHY1_RXD1_N
38	-	GND	-	-	Ground	GND
40	AK20	MIPI_DPHY1_RX_CLKP	-	-	MIPI_DPHY 1 clock receiving+	MIPI_DPHY1_RXCLK_P
42	AL20	MIPI_DPHY1_RX_CLKN	-	-	MIPI_DPHY 1 clock receiving-	MIPI_DPHY1_RXCLK_N
44	-	GND	-	-	Ground	GND

continues on next page

Table 6 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
46	AK21	MIPI_DPHY1_RX_D2P	-	-	MIPI_DPHY 1 data receiving 2+	MIPI_DPHY1_RXD2_P
48	AL21	MIPI_DPHY1_RX_D2N	-	-	MIPI_DPHY 1 data receiving 2-	MIPI_DPHY1_RXD2_N
50	-	GND	-	-	Ground	GND
52	AK22	MIPI_DPHY1_RX_D3P	-	-	MIPI_DPHY 1 data receiving 3+	MIPI_DPHY1_RXD3_P
54	AL22	MIPI_DPHY1_RX_D3N	-	-	MIPI_DPHY 1 data receiving 3-	MIPI_DPHY1_RXD3_N
56	-	GND	-	-	Ground	GND
58	AN29	MIPI_DPHY0_RX_DOP	-	-	MIPI_DPHY 0 data receiving 0+	MIPI_DPHY0_RXD0_P
60	AP29	MIPI_DPHY0_RX_DON	-	-	MIPI_DPHY 0 data receiving 0-	MIPI_DPHY0_RXD0_N
62	-	GND	-	-	Ground	GND
64	AN30	MIPI_DPHY0_RX_D1P	-	-	MIPI_DPHY 0 data receiving 1+	MIPI_DPHY0_RXD1_P
66	AP30	MIPI_DPHY0_RX_D1N	-	-	MIPI_DPHY 0 data receiving 1-	MIPI_DPHY0_RXD1_N
68	-	GND	-	-	Ground	GND
70	AN32	MIPI_DPHY0_RX_CLKP	-	-	MIPI_DPHY 0 clock receiving+	MIPI_DPHY0_RXCLK_P
72	AP31	MIPI_DPHY0_RX_CLKN	-	-	MIPI_DPHY 0 clock receiving-	MIPI_DPHY0_RXCLK_N
74	-	GND	-	-	Ground	GND
76	AN33	MIPI_DPHY0_RX_D2P	-	-	MIPI_DPHY 0 data receiving 2+	MIPI_DPHY0_RXD2_P
78	AP32	MIPI_DPHY0_RX_D2N	-	-	MIPI_DPHY 0 data receiving 2-	MIPI_DPHY0_RXD2_N
80	-	GND	-	-	Ground	GND
82	AN34	MIPI_DPHY0_RX_D3P	-	-	MIPI_DPHY 0 data receiving 3+	MIPI_DPHY0_RXD3_P
84	AP33	MIPI_DPHY0_RX_D3N	-	-	MIPI_DPHY 0 data receiving 3-	MIPI_DPHY0_RXD3_N
86	-	GND	-	-	Ground	GND
88	-	PWRON_L	-	-	Power on control	PWRON_L
90	P31	GPIO0_A4	GPIO0_A4_u	1.8V	SDMMC card detection signal	SDMMC_DET_L
92	L30	GPIO0_B0	GPIO0_B0_z	1.8V	GMAC0 Reset	GMAC0_RESET
94	L29	GPIO0_B3	GPIO0_B3_z	1.8V	GMAC0 Interrupt	GMAC0_INT
96	-	GND	-	-	Ground	GND
98	-	VCC12V_DCIN	-	-	12V power input	VCC12V_DCIN
100	-	VCC12V_DCIN	-	-	12V power input	VCC12V_DCIN

Table7 LEFT_UP (P4) Connector Interface(Odd) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
1	Y29	GPIO3_C0	GPIO3_C0_d	3.3V	MIPI_DSI1 Interrupt	MIPI_DSI1_INT
3	Y27	GPIO3_C1	GPIO3_C1_d	3.3V	EDP_LED enable	EDP_LED_EN
5	Y30	GMAC1_MDIO	GPIO3_C3_d	3.3V	GMAC1 Serial Management Data	GMAC1_MDIO
7	Y31	GMAC1_MDC	GPIO3_C2_d	3.3V	GMAC1 Serial Management Clock	GMAC1_MDC
9	AA28	GPIO3_B7	GPIO3_B7_d	3.3V	MIPI_DSI1 Reset	MIPI_DSI1_RESET
11	AH24	GPIO3_D0	GPIO3_D0_u	3.3V	PCIE20 link activation signal	PCIE20X1_2_WAKEN_M0
13	-	GND	-	-	Ground	GND
15	AG23	GPIO3_D1	GPIO3_D1_d	3.3V	PCIE20 Reset	PCIE20X1_2_PERSTN_M0
17	AG24	GPIO3_D3	GPIO3_D3_d	3.3V	MIPI_DSI2 interrupt signal	MIPI_DSI2_INT
19	AG25	GPIO3_D2	GPIO3_D2_d	3.3V	MIPI_DSI2 reset signal	MIPI_DSI2_RESET
21	AB28	GPIO3_D5	GPIO3_D5_d	3.3V	UART9 sending	UART9_TX_M2
23	AA27	GPIO3_D4	GPIO3_D4_d	3.3V	UART9 receiving	UART9_RX_M2
25	AG26	GPIO3_C6	GPIO3_C6_u	3.3V	MIPI_DSI2 enable signal	MIPI_DSI2_EN
27	-	GND	-	-	Ground	GND
29	AH25	GPIO3_C5	GPIO3_C5_u	3.3V	CAN2 data sending	CAN2_TX_M0
31	AH26	GPIO3_C4	GPIO3_C4_u	3.3V	CAN2 data receiving	CAN2_RX_M0
33	AJ24	GPIO3_C7	GPIO3_C7_u	3.3V	PCIE20X1_2_CLKREQN signal	PCIE20X1_2_CLKREQN_M0

continues on next page

Table 7 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
35	AH27	ETH1_REFCLKO_25M	GPIO3_A6_d	3.3V	PHY 25MHz reference clock output	ETH1_REFCLKO_25M
37	AE29	GMAC1_MCLKINOUT	GPIO3_B6_d	3.3V	PHY 125MHz Sync Clock Input	GMAC1_MCLKINOUT
39	AE28	GPIO3_B2	GPIO3_B2_d	3.3V	MIPI_DSI1 enable signal	MIPI_DSI1_EN
41	-	GND	-	-	Ground	GND
43	AB31	GPIO2_B4	GPIO2_B4_u	1.8V	I2C4 data	I2C4_SDA_M1
45	AB30	GPIO2_B5	GPIO2_B5_u	1.8V	I2C4 clock	I2C4_SCL_M1
47	AB33	GMAC0_MDIO	GPIO4_C5_d	1.8V	GMAC0 Serial Management Data	GMAC0_MDIO
49	AB34	GMAC0_MDC	GMAC0_MDC	1.8V	GMAC0 Serial Management Clock	GMAC0_MDC
51	-	GND	-	-	Ground	GND
53	AC30	GPIO2_C4	GPIO2_C4_d	1.8V	eDP plug detection	EDP_HPD
55	AE30	GPIO2_C5	GPIO2_C5_d	1.8V	PCIE20X1 insertion detection	PCIE20X1_PRSNT_L_1V8
57	AD30	ETH0_REFCLKO_25M	GPIO2_C3_d	1.8V	PHY 25MHz reference clock output	ETH0_REFCLKO_25M
59	AF33	GPIO4_C6	GPIO4_C6_d	1.8V	PCIe30x4 Hot-plug detection	PCIe30x4_PRSNT_L_1V8
61	AF34	GMAC0_MCLKINOUT	GPIO4_C3_d	1.8V	PHY 125MHz Sync Clock Input	GMAC0_MCLKINOUT
63	-	GND	-	-	Ground	GND
65	AG32	MIPI_CSI1_RX_D0N	-	-	CSI1 data receiving 0-	MIPI_CSI1_RXD0_N
67	AG31	MIPI_CSI1_RX_D0P	-	-	CSI1 data receiving 0+	MIPI_CSI1_RXD0_P
69	-	GND	-	-	Ground	GND
71	AH32	MIPI_CSI1_RX_D1N	-	-	CSI1 data receiving 1-	MIPI_CSI1_RXD1_N
73	AH31	MIPI_CSI1_RX_D1P	-	-	CSI1 data receiving 1+	MIPI_CSI1_RXD1_P
75	-	GND	-	-	Ground	GND
77	AJ32	MIPI_CSI1_RX_CLKON	-	-	CSI1 clock 0-	MIPI_CSI1_RXCLK0_N
79	AJ31	MIPI_CSI1_RX_CLKOP	-	-	CSI1 clock 0+	MIPI_CSI1_RXCLK0_P
81	-	GND	-	-	Ground	GND
83	AK32	MIPI_CSI1_RX_D2N	-	-	CSI1 data receiving 2-	MIPI_CSI1_RXD2_N
85	AK31	MIPI_CSI1_RX_D2P	-	-	CSI1 data receiving 2+	MIPI_CSI1_RXD2_P
87	-	GND	-	-	Ground	GND
89	AL32	MIPI_CSI1_RX_D3N	-	-	CSI1 data receiving 3-	MIPI_CSI1_RXD3_N
91	AL31	MIPI_CSI1_RX_D3P	-	-	CSI1 data receiving 3+	MIPI_CSI1_RXD3_P
93	-	GND	-	-	Ground	GND
95	AM32	MIPI_CSI1_RX_CLK1N	-	-	CSI1 clock 1-	MIPI_CSI1_RXCLK1_N
97	AM31	MIPI_CSI1_RX_CLK1P	-	-	CSI1 clock 1+	MIPI_CSI1_RXCLK1_P
99	-	GND	-	-	Ground	GND

Table8 LEFT_UP (P4) Connector Interface(Even) Pin Definition

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
2	V31	GPIO0_C7	GPIO0_C7_d	3.3V	PWM6	PWM6_M0
4	W31	GPIO0_D0	GPIO0_D0_d	3.3V	TYPEC0 enable	TYPEC0_PWREN
6	U33	GPIO0_D3	GPIO0_D3_u	3.3V	TYPEC1 enable	TYPEC1_PWREN
8	-	GND	-	-	Ground	GND
10	AC34	GMAC0_TXD3	GPIO2_B2_u	1.8V	GMAC0 data sending 3	GMAC0_TXD3
12	AC33	GMAC0_TXD2	GPIO2_B1_u	1.8V	GMAC0 data sending 2	GMAC0_TXD2
14	AD34	GMAC0_TXD1	GPIO2_B7_d	1.8V	GMAC0 data sending 1	GMAC0_TXD1
16	AD33	GMAC0_TXD0	GPIO2_B6_d	1.8V	GMAC0 data sending 0	GMAC0_TXD0
18	AE34	GMAC0_TXEN	GPIO2_C0_d	1.8V	GMAC0 sending control	GMAC0_TXEN
20	AE33	GMAC0_TXCLK	GPIO2_B3_d	1.8V	GMAC0 clock sending	GMAC0_TXCLK
22	-	GND	-	-	Ground	GND
24	AC31	GMAC0_RXD3	GPIO2_A7_u	1.8V	GMAC0 data receiving 3	GMAC0_RXD3

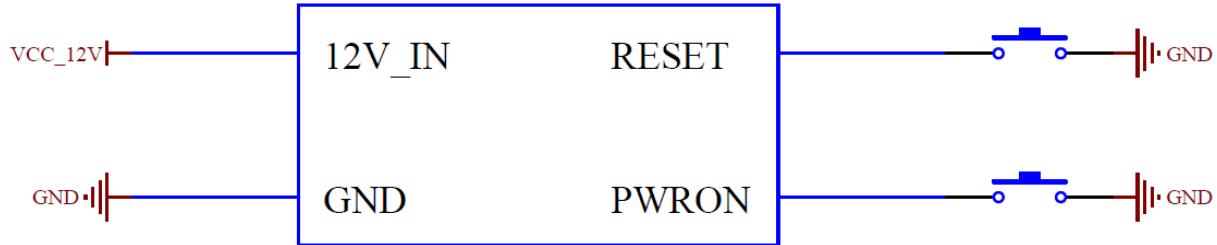
continues on next page

Table 8 – continued from previous page

NUM	BALL	Signal Name	GPIO	VOL	Pin Description	Default Function
26	AC32	GMAC0_RXD2	GPIO2_A6_u	1.8V	GMAC0 data receiving 2	GMAC0_RXD2
28	AD31	GMAC0_RXD1	GPIO2_C2_d	1.8V	GMAC0 data receiving 1	GMAC0_RXD1
30	AD32	GMAC0_RXD0	GPIO2_C1_d	1.8V	GMAC0 data receiving 0	GMAC0_RXD0
32	AE31	GMAC0_RXDV CRS	GPIO4_C2_d	1.8V	GMAC0 receiving control	GMAC0_RXDV CRS
34	AE32	GMAC0_RXCLK	GPIO2_B0_u	1.8V	GMAC0 clock receiving	GMAC0_RXCLK
36	-	GND	-	-	Ground	GND
38	AA30	GMAC1_TXD3	GPIO3_A1_u	3.3V	GMAC1 data sending 3	GMAC1_TXD3
40	AA29	GMAC1_TXD2	GPIO3_A0_u	3.3V	GMAC1 data sending 2	GMAC1_TXD2
42	AC29	GMAC1_TXD1	GPIO3_B4_u	3.3V	GMAC1 data sending 1	GMAC1_TXD1
44	AC28	GMAC1_TXD0	GPIO3_B3_u	3.3V	GMAC1 data sending 0	GMAC1_TXD0
46	AD29	GMAC1_TXEN	GPIO3_B5_u	3.3V	GMAC1 sending control	GMAC1_TXEN
48	AD28	GMAC1_TXCLK	GPIO3_A4_d	3.3V	GMAC1 clock sending	GMAC1_TXCLK
50	-	GND	-	-	Ground	GND
52	AE27	GMAC1_RXD3	GPIO3_A3_u	3.3V	GMAC1 data receiving 3	GMAC1_RXD3
54	AD27	GMAC1_RXD2	GPIO3_A2_u	3.3V	GMAC1 data receiving 2	GMAC1_RXD2
56	AG28	GMAC1_RXD1	GPIO3_A2_u	3.3V	GMAC1 data receiving 1	GMAC1_RXD1
58	AG29	GMAC1_RXD0	GPIO3_A7_u	3.3V	GMAC1 data receiving 0	GMAC1_RXD0
60	AH29	GMAC1_RXDV CRS	GPIO3_B1_d	3.3V	GMAC1 receiving control	GMAC1_RXDV CRS
62	AH30	GMAC1_RXCLK	GPIO3_A5_d	3.3V	GMAC1 clock receiving	GMAC1_RXCLK
64	-	GND	-	-	Ground	GND
66	AG33	MIPI_CSIO_RX_DOP	-	-	CSI0 data receiving 0+	MIPI_CSIO_RXD0_P
68	AG34	MIPI_CSIO_RX_DON	-	-	CSI0 data receiving 0-	MIPI_CSIO_RXD0_N
70	-	GND	-	-	Ground	GND
72	AH33	MIPI_CSIO_RX_D1P	-	-	CSI0 data receiving 1+	MIPI_CSIO_RXD1_P
74	AH34	MIPI_CSIO_RX_D1N	-	-	CSI0 data receiving 1-	MIPI_CSIO_RXD1_N
76	-	GND	-	-	Ground	GND
78	AJ33	MIPI_CSIO_RX_CLK0P	-	-	CSI0 clock 0+	MIPI_CSIO_RXCLK0_P
80	AJ34	MIPI_CSIO_RX_CLK0N	-	-	CSI0 clock 0-	MIPI_CSIO_RXCLK0_N
82	-	GND	-	-	Ground	GND
84	AK33	MIPI_CSIO_RX_D2P	-	-	CSI0 data receiving 2+	MIPI_CSIO_RXD2_P
86	AK34	MIPI_CSIO_RX_D2N	-	-	CSI0 data receiving 2-	MIPI_CSIO_RXD2_N
88	-	GND	-	-	Ground	GND
90	AL33	MIPI_CSIO_RX_D3P	-	-	CSI0 data receiving 3+	MIPI_CSIO_RXD3_P
92	AL34	MIPI_CSIO_RX_D3N	-	-	CSI0 data receiving 3-	MIPI_CSIO_RXD3_N
94	-	GND	-	-	Ground	GND
96	AM33	MIPI_CSIO_RX_CLK1P	-	-	CSI0 clock 1+	x
98	AM34	MIPI_CSIO_RX_CLK1N	-	-	CSI0 clock 1-	x
100	-	GND	-	-	Ground	GND

2.6 SoM Hardware Design Description

The FET3588-C SoM integrates power and storage circuits into a compact module, requiring minimal external circuitry. A minimum system can operate with just power supply and boot configuration, as shown in the figure below:



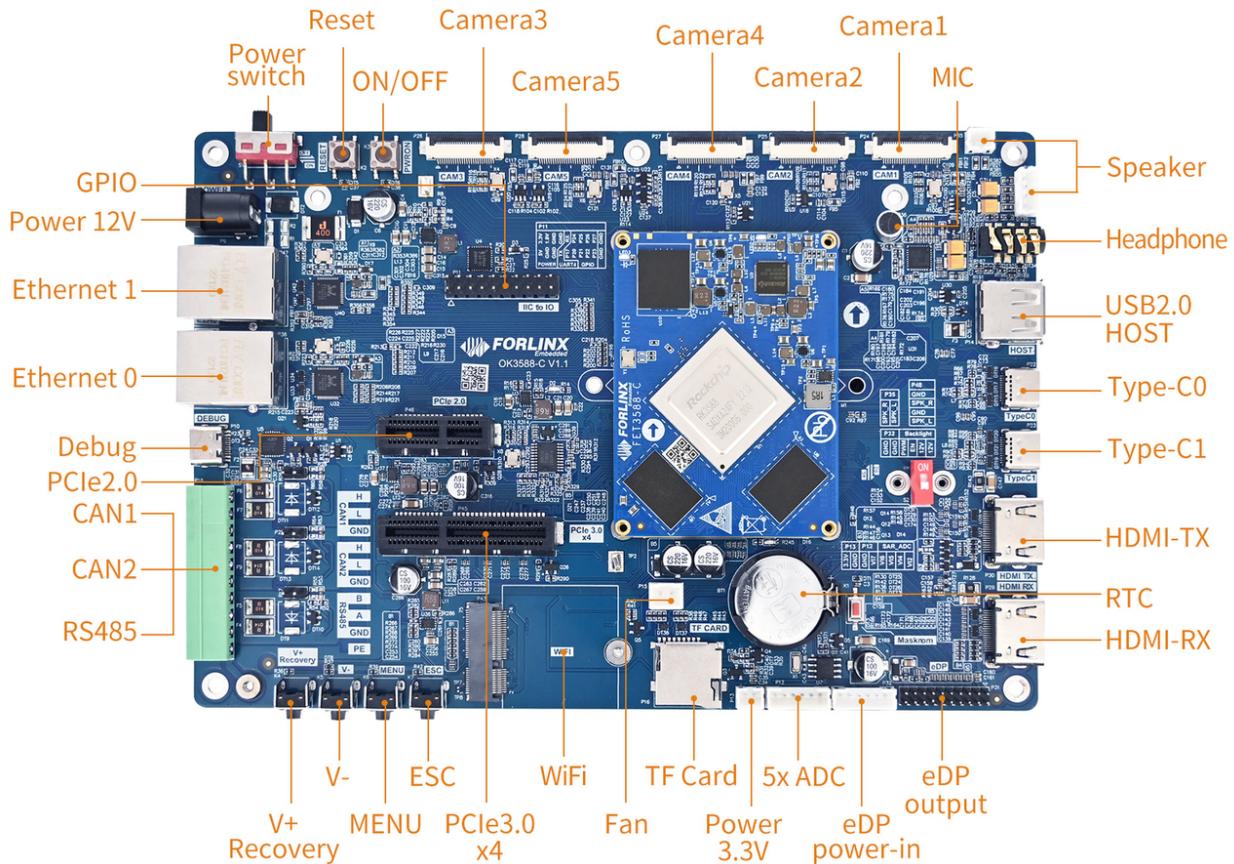
Please refer to “Appendix IV” for the minimal system schematic diagram. However, in most cases, it is recommended to connect some external devices in addition to the minimal system, such as a debugging serial port, otherwise, the user can not check whether the system is booted. After completing these steps, additional user-specific functions can be added based on the default interface definitions provided by Forlinx for the SoM.

Please refer to section 3.5 in “Chapter 3. OK3588-C Carrier Board Description” for the peripheral circuits.

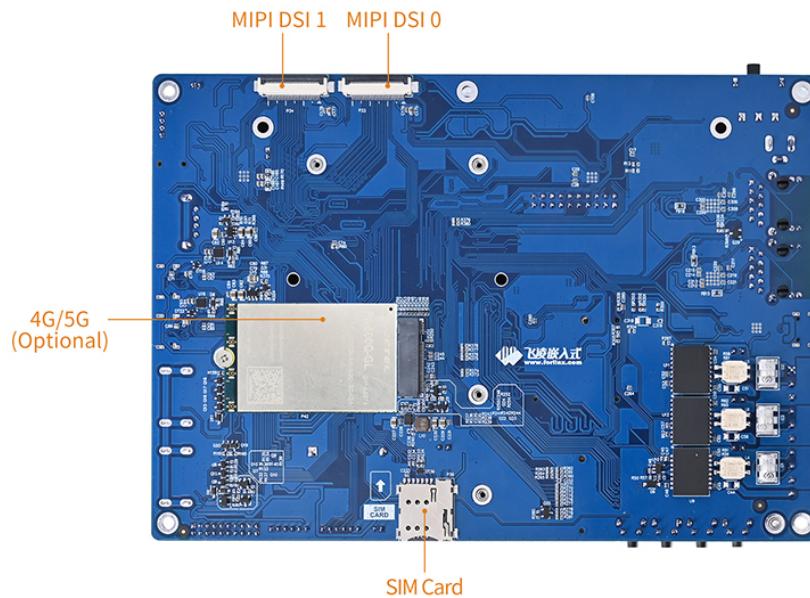
1.1.6 3. OK3588-C Development Platform Description

3.1 OK3588-C Development Board Interface Diagram

Connection method is board-to-board, and main interfaces are shown in the figure below:

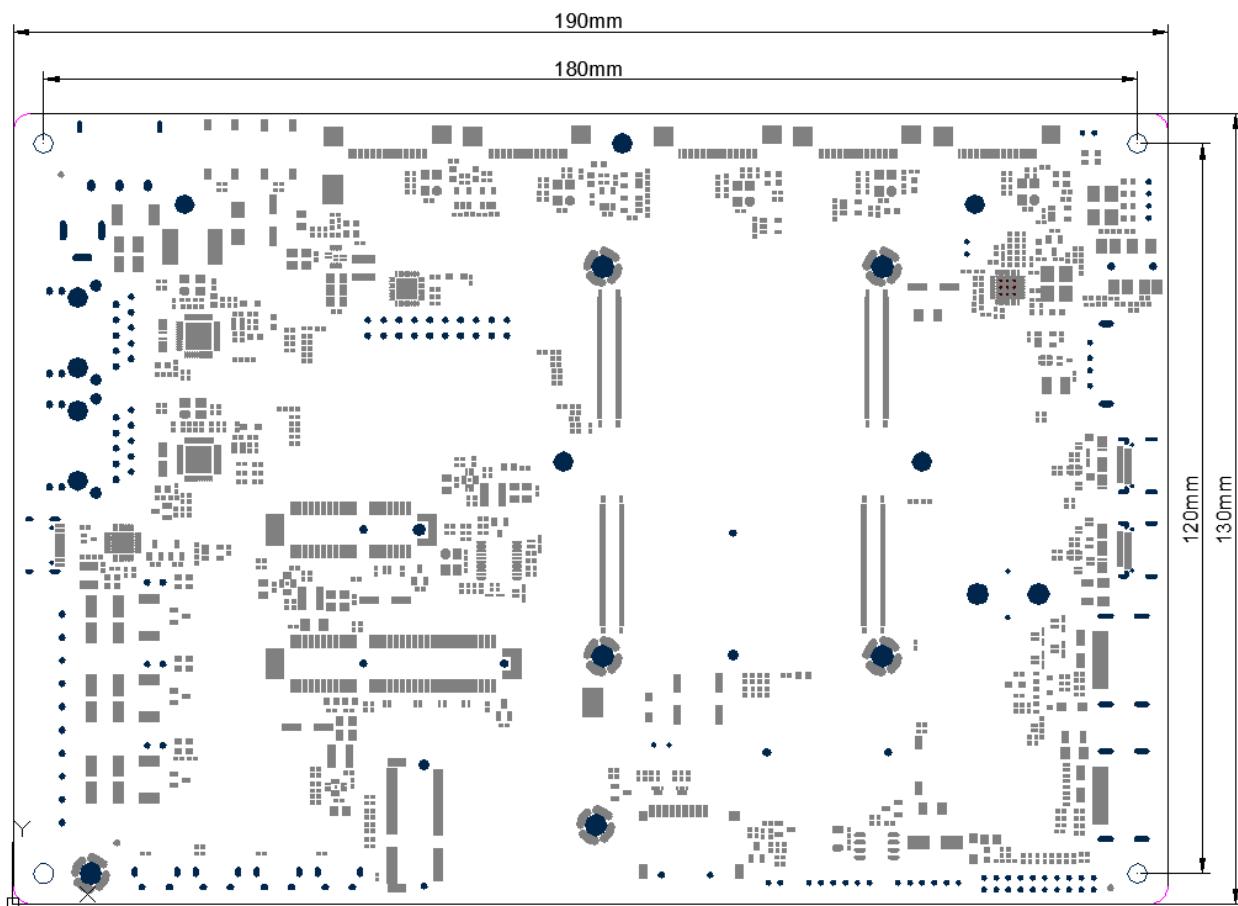


Front



Back

3.2 OK3588-C SoM Dimension Diagram

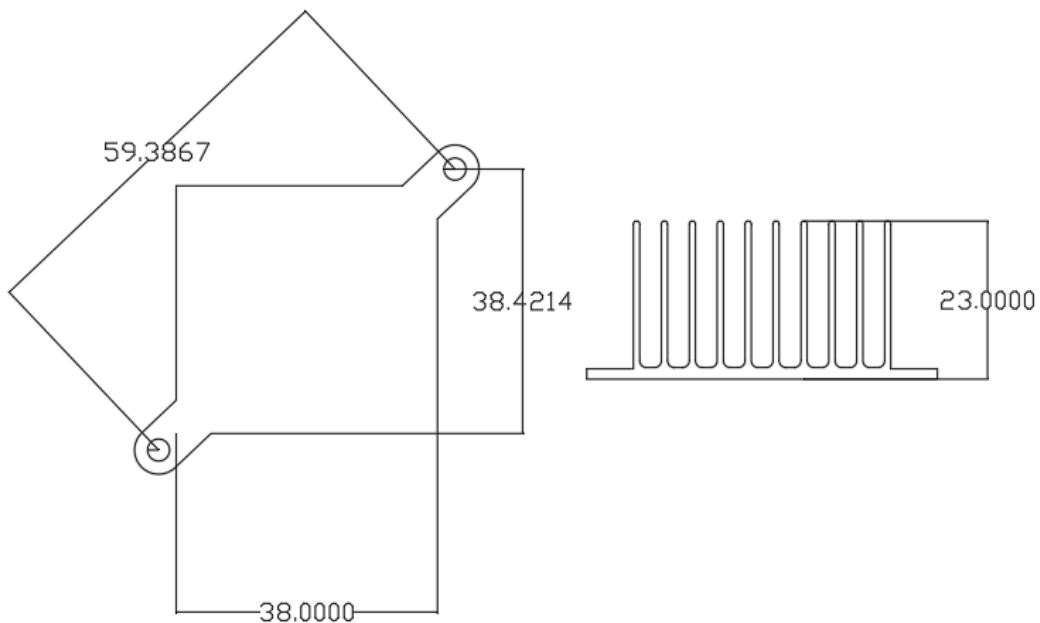


PCB Size: 190mm × 130mm.

Fixed hole size: spacing: 180mm × 120mm, hole diameter: 3.2mm.

Plate making process: thickness 1.6mm, 4-layer PCB.

The OK3588-C carrier board reserves an installation hole with a diameter of 3.2 mm for a heat sink. You can choose to install the heat sink according to the on - site environment. Please add a layer of insulating thermal conductive silicone pad on the contact surface between the heat sink and the SoM. 38Mm×38mm×23mm. For more detailed dimensions, please refer to the following figure.



3.3 Carrier Board Naming Rules

A - B - C + D E F : G - H

Field	Field Description	Value	Description
A	Product Line Identification	OK	Forlinx Embedded carrier board
-	Separator	-	The separator “-” is omitted between the product line
B	CPU Name	3588	RK3588
-	Segment Identification	-	Parameter segment sign
C	Connection	C	Board to Board Connector
+	Segment Identification	+	The configuration parameter section follows this identifier.
D	Type	M	Carrier board(Note: carrier board identification M, not filled by default)
E	Operating Temperature	C	0 to 80°C Commercial-grade
F	PCB Version	11	V1.1
:	Separator	:	It's followed by the manufacturer's internal identification.
G	Connector origin	1	Imported connector
-	Connector	-	Grade Mark Connector
H	Grade Identification	PC	Prototype Sample
	Blank		Mass Production
	SC		Special-purpose use: According to the customer's special requirements for special modifications (according to a function to modify a material,etc.)

3.4 Carrier Board Resources

Functionality	Quantity	Parameter
MIPI CSI	5	•2 × MIPI DPHY V1.2 4-lane interfaces, each lane supporting up to 2.5 Gbps; routed via 2 x 26-pin FPC connectors, with OV13850 cameras mounted by default; •2 × MIPI DPHY V1.2 2-lane interfaces, each lane supporting up to 2.5 Gbps; routed via three 26-pin FPC connectors, with OV5645 cameras mounted by default; •1 × MIPI DPHY V1.2 4-lane interface, each lane supporting up to 2.5 Gbps.
MIPI DSI	2	•Each MIPI interface supports 4lane output with a maximum resolution of 4K@60fps; •Compatible with Forlinx's 7-inch MIPI display, featuring a resolution of 1024x600@30fps;
HDMI RX	1	•Led out via standard HDMI socket; •Supports up to 4K @ 60Hz;
HDMI TX	1	•Led out via standard HDMI socket; •Supports up to 7680x4320 @ 60Hz;
eDP TX	1	•Adapt to 1080p @ 60Hz display; •Support up to 4K @ 60Hz;
DP TX	2	•2 x DP interfaces are used in combination with USB3.1 Gen1 and led out through Type-C interface; •Support up to 7680x4320 @ 30Hz;
USB3 Gen1	2	•Led out via Type-C interface; •Used with DP TX up to 5Gbps;
USB2 Host	1	•Led out via Type-A USB interface; •Supports three modes of high speed (480Mbps), full speed (12Mbps) and low speed (1.5M bps);
PCIe3 1	1	•1x 4lanes PCIe signals led out via PCIe X 4 slots; •Supports 2.5g bps (PCIe 1.1), 5Gbps (PCIe 2.1), 8Gbps (PCIe 3.0);
PCIe2 1	1	•Led out via PCIe X 1 slot; •Supports 5Gbps speed;
Ethernet	2	•Led out via 2 x RJ45 interfaces; •Support 10/100/1000 Mbps data transmission rate;
TF card	1	•TF card is available, rate up to 150MHz, support SDR104 mode;
Audio	1	•Codec chip on board, support headphone output, MIC input level Speaker;
RS485	1	•1 x RS485 CAN bus led out through RS485 transceiver;
UART	1	•Led out via 2.54 mm pitch; •Up to 4Mbps baud rate;
4G/5G	1	•Supports 4G/5G modules packaged in M.2 format;
WIFI	1	•Supports M.2-packaged Wi-Fi & BT modules; • Forlinx has adapted a module supporting Wi-Fi 6 SU/MU-MIMO + Bluetooth 5.3;
ADC	5	• Led out via PH2.0 socket; • 12-bit resolution and up to 1MS/s sampling rate;
RTC	1	•Onboard RTC chip and battery socket;
FAN	1	•Onboard fan connector;
GPIO	9	•9 x GPIO (3.3 V level) and 5V, 3.3 V, and 1.8 V power led out via a 2.54 mm pitch header pin;

Note: “TBD” means the function has not been developed in this phase;

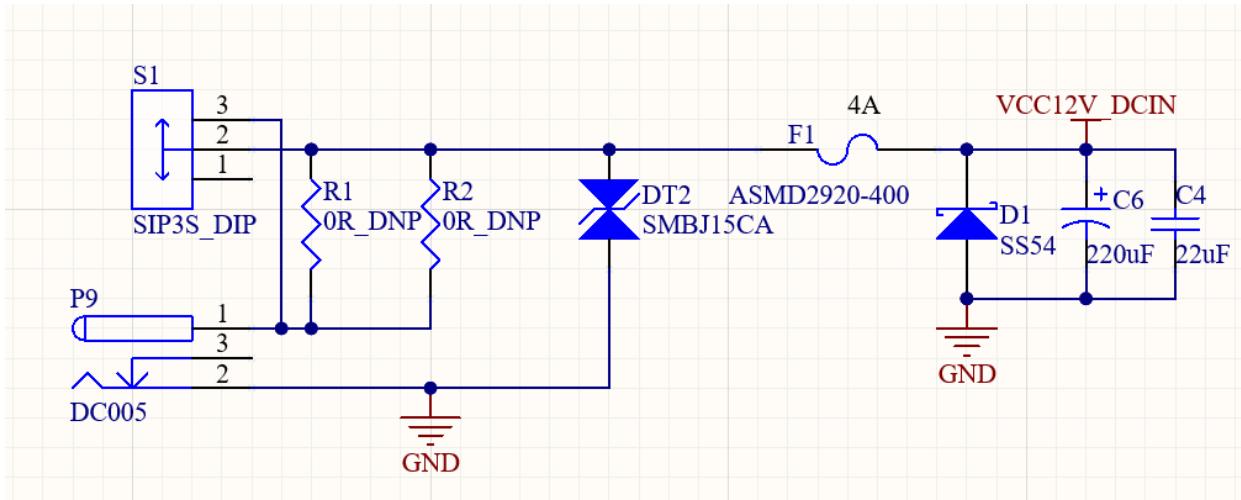
The parameters in the table are hardware design or theoretical CPU values.

3.5 OK3588-C Carrier Board Description

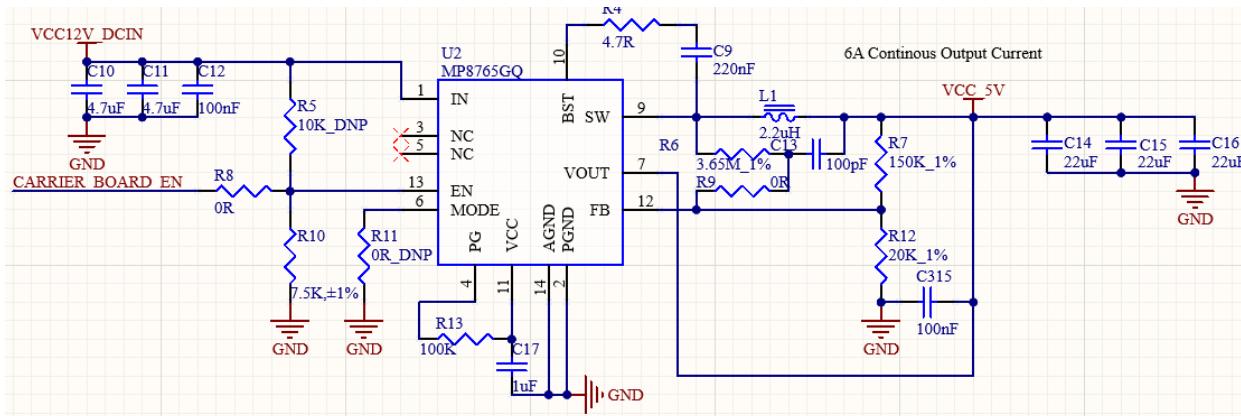
Note: The component UID with “_DNP” mark in the diagram below represents it is not soldered by default.

3.5.1 Carrier Board Power

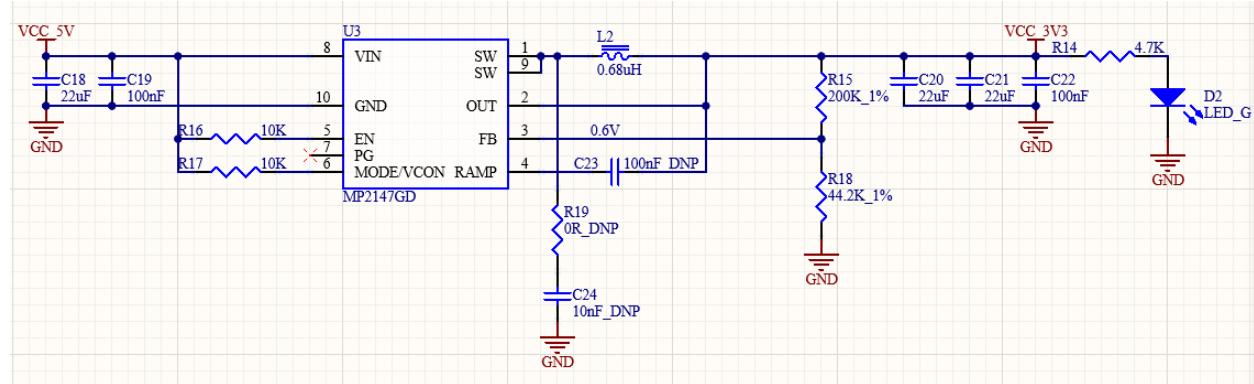
It uses a 12V power adapter for the power supply, and the power connector is a DC005 socket. S1(DIP switch) is the power switch, which moves according to the screen printing indication on the board. The rear of S1 has TVS for electrostatic protection, F1 for over-current protection, and D1 and F1 cooperate for anti-reverse connection protection. VCC12V_DCIN supplies power to SoM and carrier board at the same time.



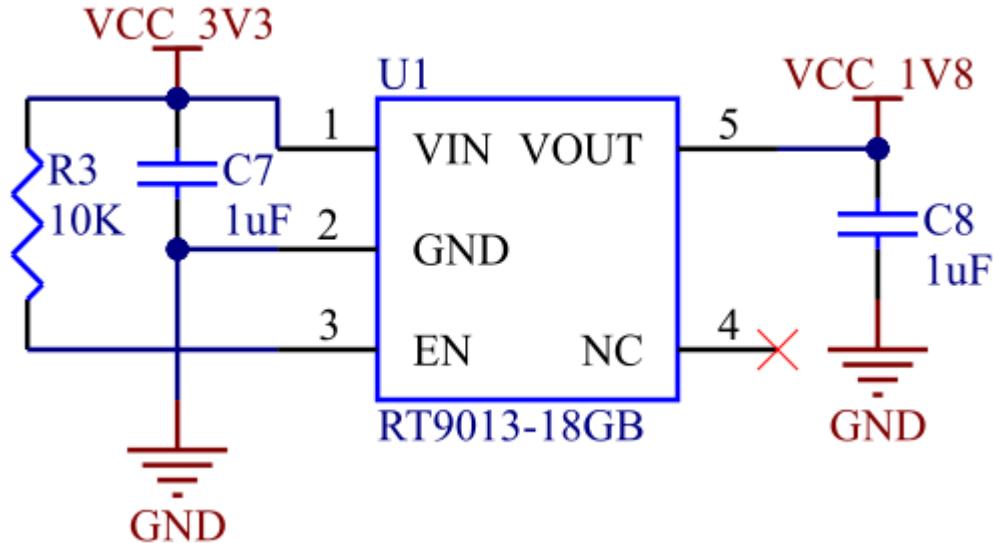
VCC12V_DCIN is decreased to VCC _ 5V via U2. VCC _ 5V supplies power for some of the peripherals on the carrier board. To meet the power-up timing, the carrier board power-up is controlled by CARRIER_BOARD_EN—a signal of SoM. (the electrical level is 3.3V, and the driving ability is 10K pull-up. If the required driving ability of the enabling pin of the enabling device exceeds this range, a buffer or gate circuit needs to be added for increasing the driving ability to ensure SoM and carrier board power-up normally).



VCC _ 5V is decreased to VCC _ 3V3 via U3 VCC _ 3V3 supplies power for partial carrier board configurations.



VCC_3V3 is decreased to VCC_1V8 via U1. VCC_1V8 supplies power for partial carrier board configurations.

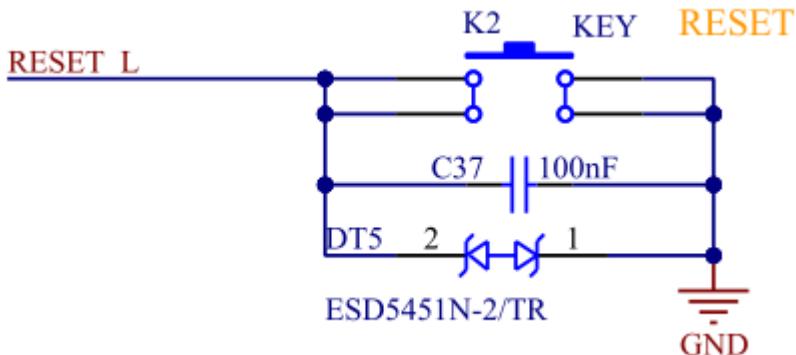


Note:

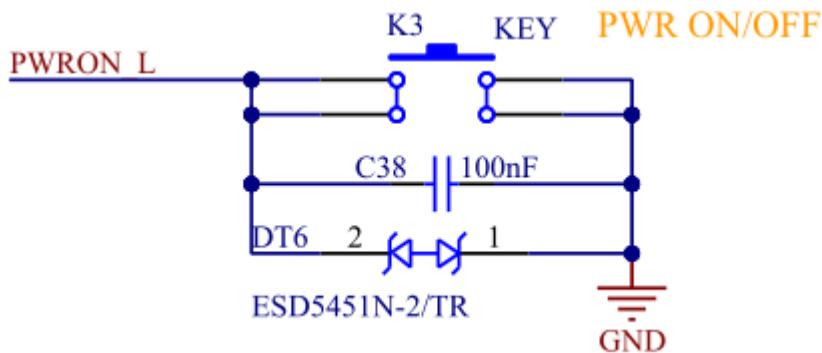
1. When undertaking independent design, it is crucial to ensure proper power sequencing during power-up; 2. For the device selection and external layout of the buck - boost chip, you need to refer to the corresponding chip manual to ensure a good power supply loop.

3.5.2 Reset and Startup/Shutdown Signal

RESET_L is SoM resetting signal input connected to the key for convenient debugging.



PWRON_L is an On/Off signal input connected to the key for convenient debugging.



3.5.3 Boot Configuration

RK3588 supports multiple boot modes. After resetting the chip reset, the boot code integrated inside the chip can be booted on the following interface devices, and the specific boot sequence can be selected according to actual application requirements:

- Serial Flash(FSPI)
- eMMC

SDMMC Card

If there is no boot code in the above devices, the system code can be downloaded to these devices by the USB2.0 OTG0 interface TYPECO_OTG_DM / TYPECO _OTG_DP signal.

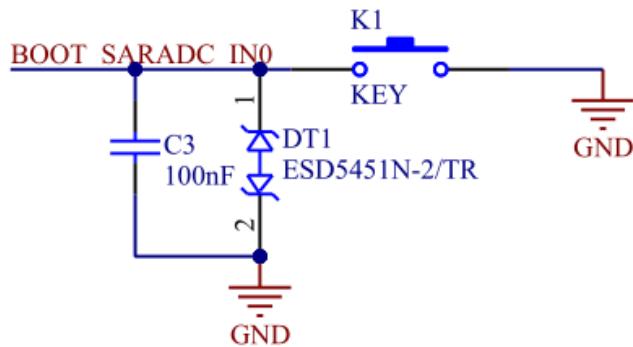
Boot Sequence Selection:

The boot sequence of RK3588 can be set by BOOT_SARADC_IN0 (PIN: P1_28), and it can boot from the peripheral corresponding to the different interfaces. As shown in the table below, the hardware is configured with different up and down resistance values to design the peripheral boot sequence of LEVEL1-LEVEL7 in seven modes that can be configurable according to actual application requirements

Table 3.5.2.1 Boot Sequence Configuration

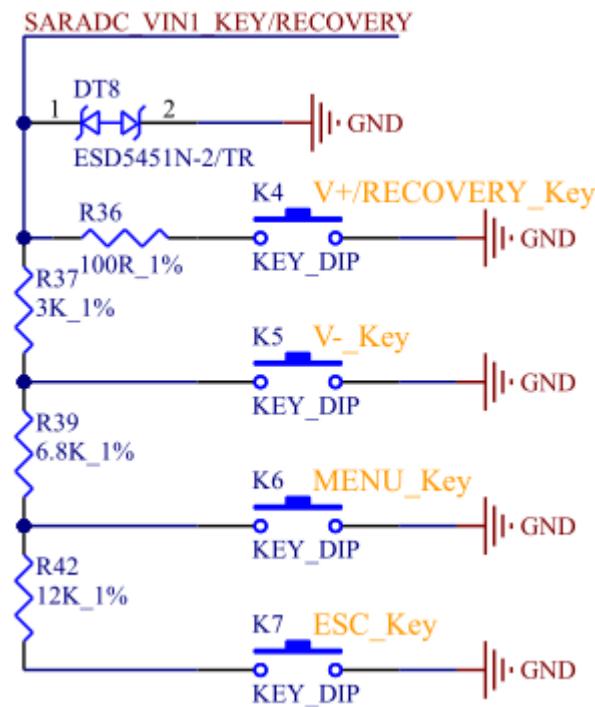
Item	Rup	Rdown	ADC	VOL	BOOT MODE
LEVEL1	DNP	1K	0	0V	USB (Maskrom mode)
LEVEL2	10K	2K	682	0.3V	SD Card-USB
LEVEL3	10K	5.1K	1365	0.6V	EMMC-USB
LEVEL4	10K	10K	2047	0.9V	FSPI M0-USB
LEVEL5	10K	20K	2730	1.2V	FSPI M1-USB
LEVEL6	10K	50K	3412	1.5V	FSPI M2-USB
LEVEL7	10K	DNP	4095	1.8V	FSPI M2-FSPI M1-FSPI M0-EMMC-SD Card-USB

BOOT_SARADC_IN0 on SoM is 10K pull-up, so the SoM defaults to start from eMMC. The pull-down resistor can be added to the carrier board to achieve other boot sequences. According to the above LEVEL1 setting, OK3588-C connects BOOT_SARADC_IN0 to GND by the touch key to achieve Maskrom mode.



USB (Maskrom mode)

SARADC_VIN1 is used to enter the recovery state due to a short circuit to the ground, and the SoM pulls it up to a 1.8V power supply using a 10K resistor. On RK3588, the key array adopts a parallel type, which can adjust the input key value by increasing or decreasing the keys and adjusting the proportion of the voltage divider resistor to achieve multi-key input to meet customer product requirements; It is recommended in the design that any two key values must be greater than ± 35 , i.e. the center voltage difference must be greater than 123mV. As shown below:



Note:

- When doing key acquisition, ESD protection is required near the keys. And 0 key value must be connected in series with a 100ohm resistor to strengthen the anti-static surge capacity (If there is only one button, ESD must be close to the button, ESD → 100ohm resistor → 1nF → chip pin);

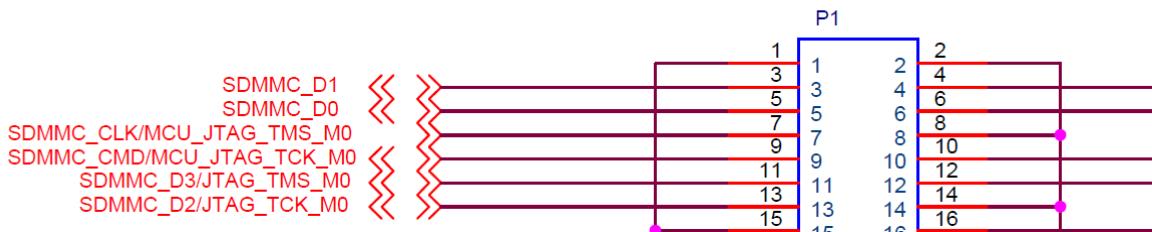
3.5.4 System Initialization Configuration Signal

There is one important signal in the FET3588 that affects the system boot configuration and needs to be configured and kept in a stable state before power-up:

GPIO0_A4 (PIN: P3_90) (default function is SDMMC_DET): determines whether P1_7, P1_9, P1_11, P1_13 are SDMMC or JTAG functions.

The ARM JTAG function of RK3588 is multiplexed with SDMMC function, and IOMUX function is switched via SD-

MMC_DET pin, so this pin also needs to be configured before power-up, otherwise, no output of the ARM JTAG function will affect the debugging of the boot stage, while no output of SDMMC will affect the SDMMC boot function.



- When this pin detects high level, the corresponding IO switches to the ARM JTAG function;
- When this pin detects low level (Most SD cards inserted will pull down this pin, if not need special treatment), the corresponding IO switches to SDMMC function;
- After the system is up, it can be switched to have registers to control IOMUX, then the pin can be released;
- For easy reference, the configuration status of this pin corresponds to its function shown as follows:

Table 3.5.4.1 FET3588 System Initialization Configuration Signal Description

Sig- signal	Internal Pull- up&down	Description
GPIOC	Pull-up	SDMMC/ARM JTAG pin multiplexing selection control signal:0: SD card insertion, SDMMC/ARM JTAG pin multiplexing as SDMMC function;1: SD card insertion, SDMMC/ARM JTAG pin multiplexing as ARM JTAG function (Default).

3.5.5 JTAG & UART Debug Circuit

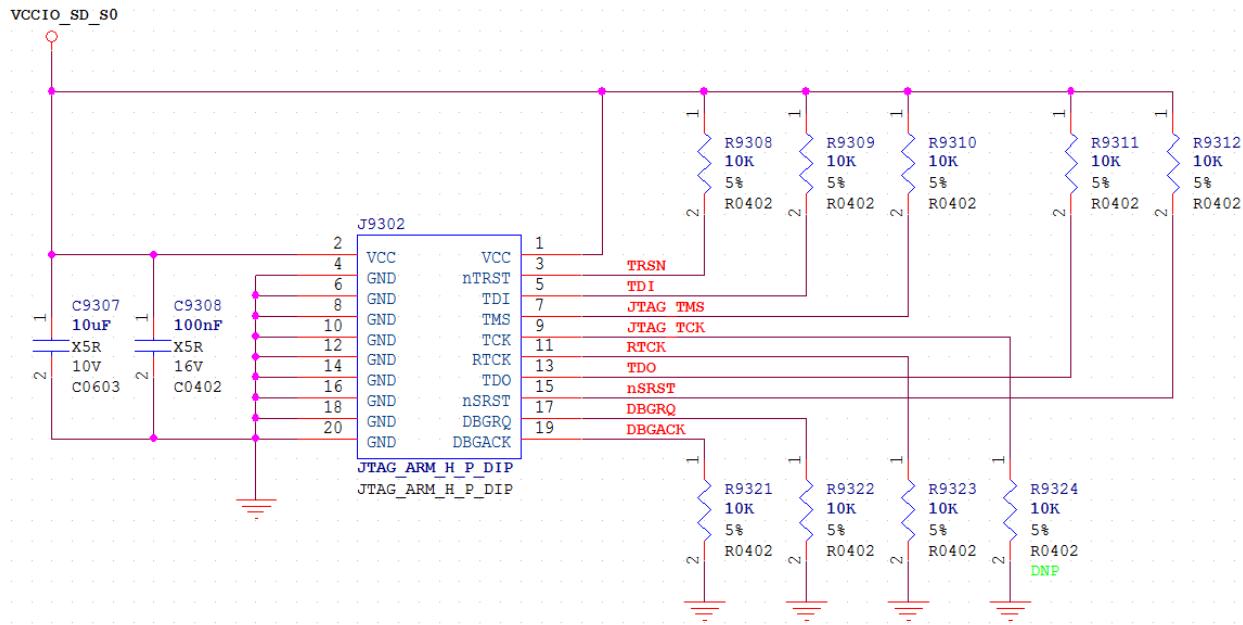
The ARM JTAG interface of the RK3588 chip complies with IEEE1149.1 standard, and the PC can connect to the DSTREAM emulator via SWD mode (two-wire mode) to debug the ARM Core inside the chip.

ARM JTAG interface description is as follows:

Table 3.5.5.1 RK3588 JTAG Debug

Signal	Description
JTAG_TCK_M0/M1	SWD mode clock input
JTAG_TMS_M0/M1	SWD mode data input and output

The JTAG connections and standard connector pin definitions are shown as follows:

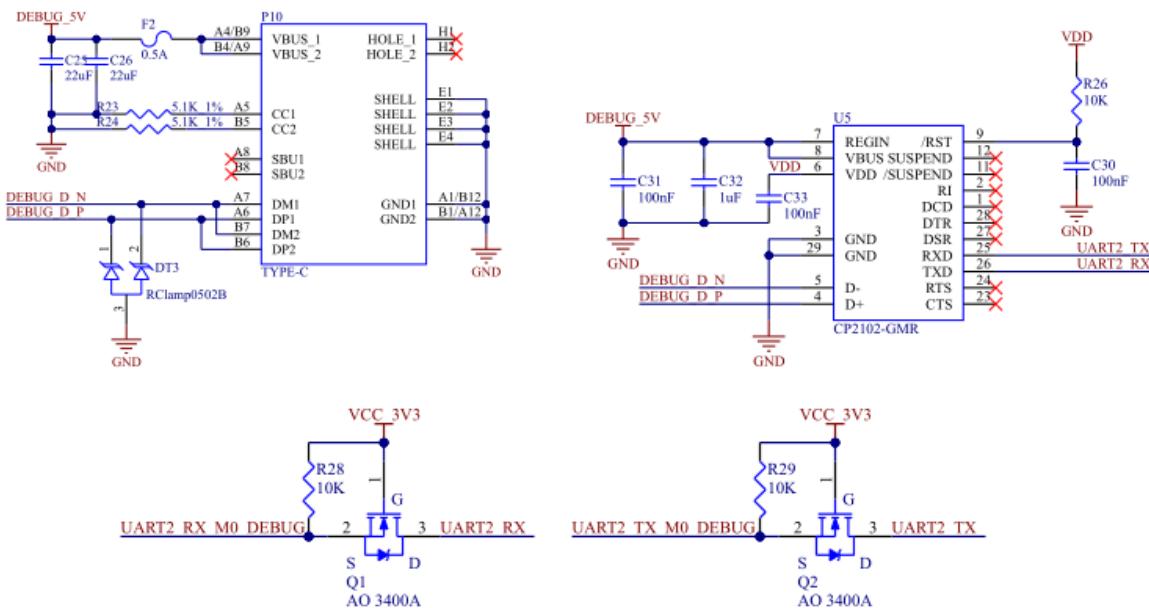


FET3588 UART Debug selects UART2_TX_M0_DEBUG (P2_7)/UART2_RX_M0_DEBUG (P2_9) by default. UART Debug signal needs to be connected with 100ohm resistor in series, if plug-in is used, and TVS tube needs to be added near the plug-in position.

To facilitate user debugging, OK3588 uses a USB to UART chip to convert the UART signal into a USB signal and leads it out through a Type-C socket. You can connect OK3588-C P10 to PC with USB Type-A to UAB Type-C cable and install a CP2102 driver.

The schematic is as follows:

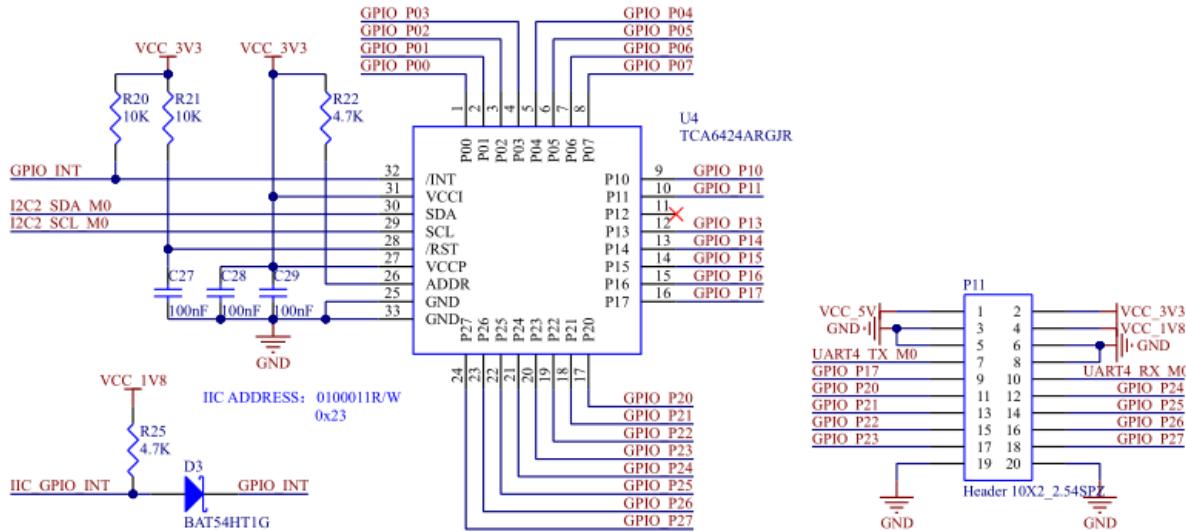
DEBUG



2. It is recommended to keep Q1 and Q2, which can effectively prevent the U5 current from flowing back to the CPU through UART2_TX/RX when the core board is not powered-up, affecting the startup and even causing damage;

3.5.6 IIC Extending IO

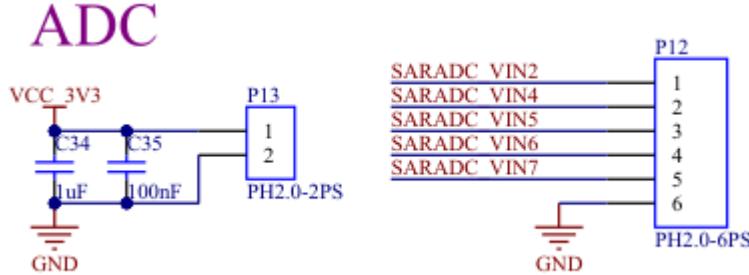
To introduce more diverse interfaces, the enable and reset signals of the carrier board are completed by the IIC to IO chip U4. At the same time, the U4 spare part of IO is led out by P11 to facilitate the expansion. The principle is as follows:



3.5.7 SARADC

The OK3588 introduces SARADC_VIN2/VIN4/VIN5/VIN6/VIN7 through P12, while P13 can provide 3.3V power output for peripherals.

As shown in the figure below:

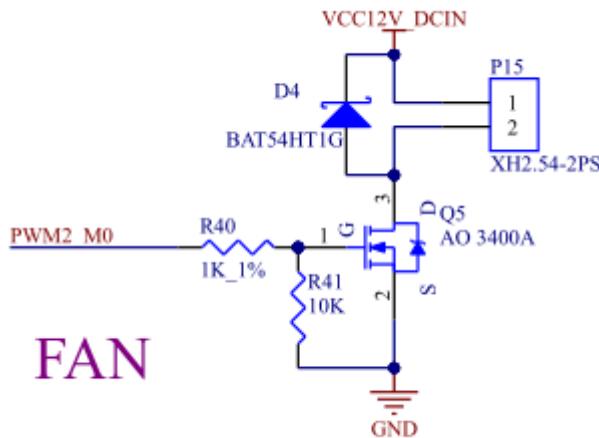


Note: When using the SARADC_VINx, 1nF capacitor must be added near the pin to eliminate jitter.

3.5.8 FAN Interface

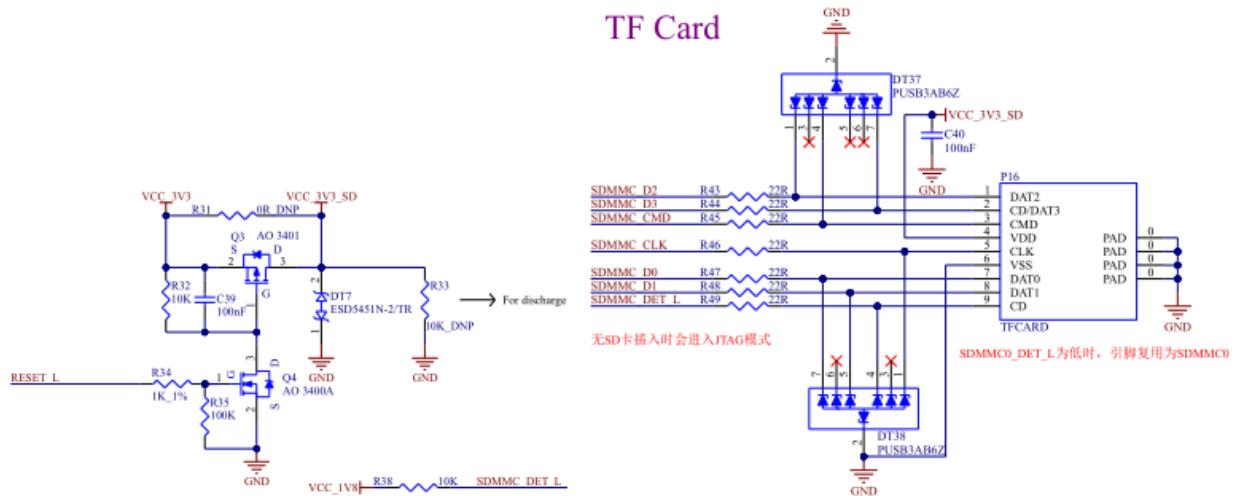
The OK3588-C has a fan power interface, which can adjust the power supply current through PWM to achieve fan speed adjustment.

As shown in the figure below:



3.5.9 TF Card

The carrier board P16 is a TF Card interface, which can support system boot and flash.



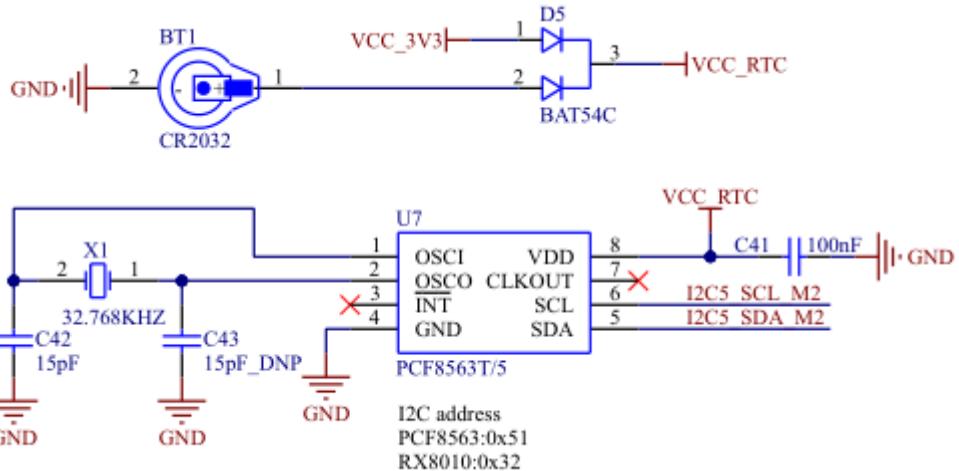
Note:

1. The power supply for the TF card must be controlled; refer to the carrier board circuit for implementation;
2. Impedance requirements: Single-ended 50ohm.

3.5.10 RTC Circuit

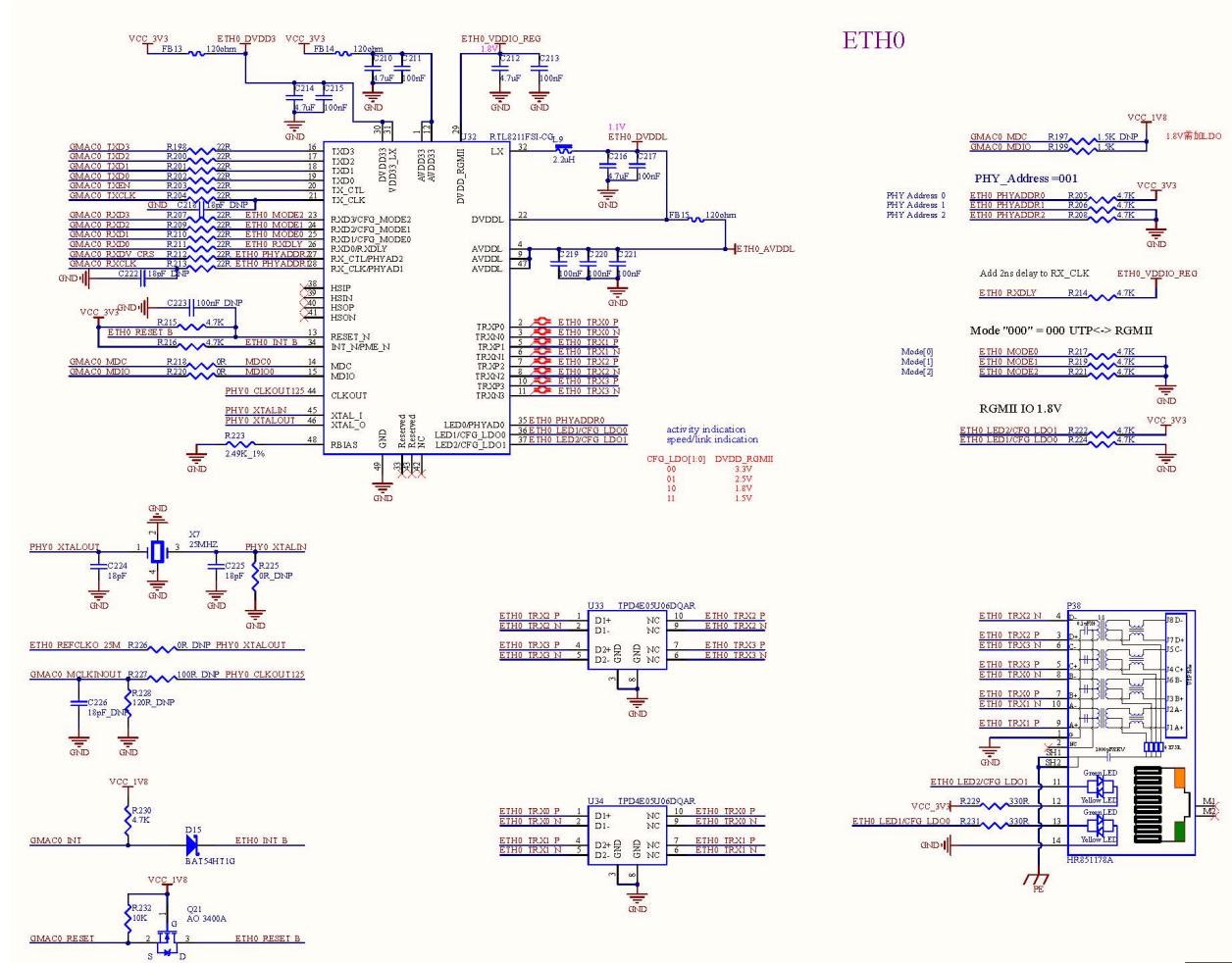
The OK3588 provides an on-board external RTC function for more accurate timing and lower power consumption. As shown in the figure below:

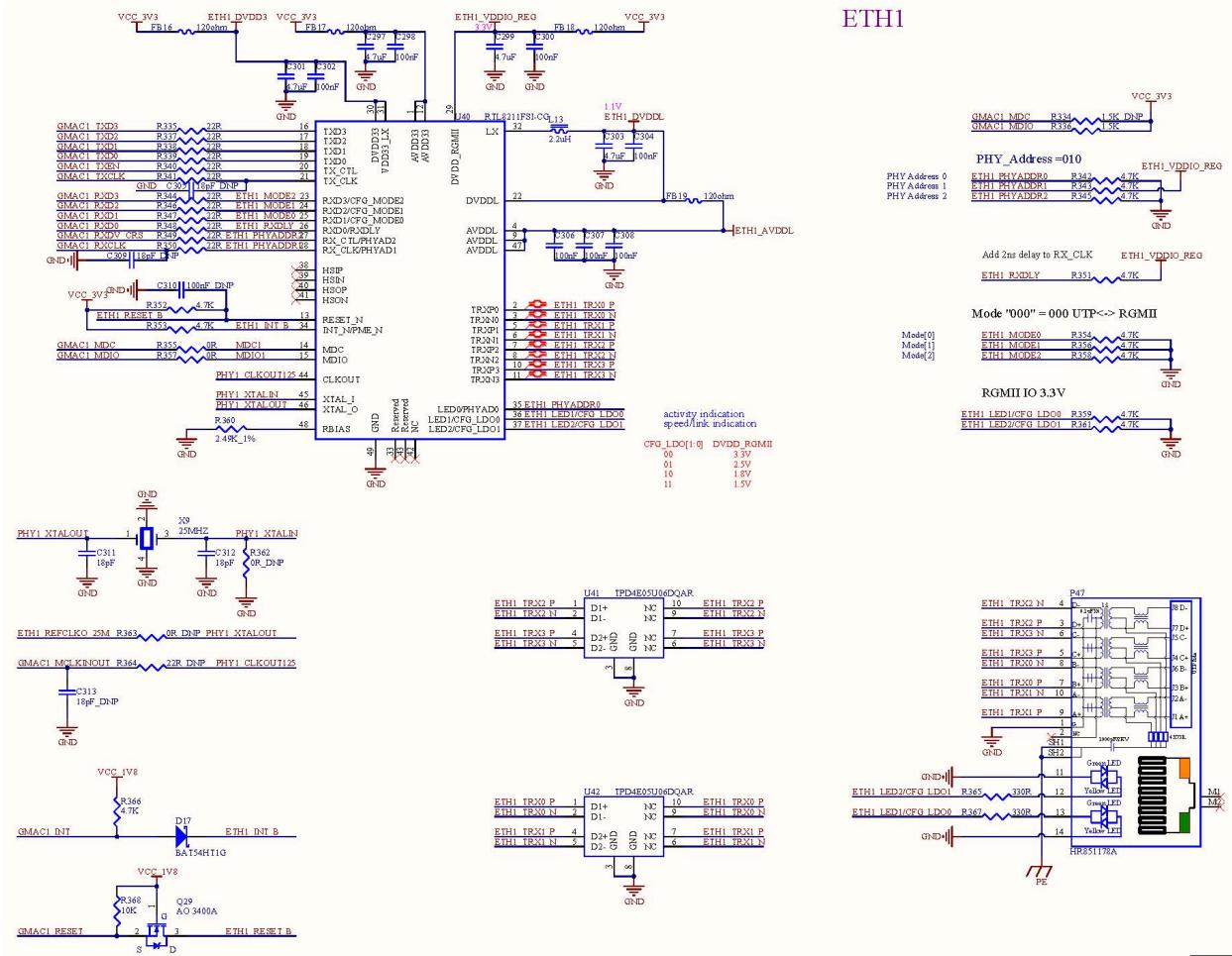
RTC



3.5.11 Ethernet Circuit

The carrier board supports dual 1000/100/10M Ethernet interfaces, which are led out via RJ45.





Note:

1.The following table shows the RK3588 RGMII/RMII interface design:

Table 3.5.11.1 RK3588 RGMII/RMII Interface

Signal	IO Type (Chip-side)	RGMII Interface	Signal Description	RMII Interface	Signal Description
GMACx_T	Output	RGMIIx_TXD[3:0]	Data sending	RMIIx_TXD[1:0]	Data sending
GMACx_T	Output	RGMIIx_TX	Reference clock for data sending	—	—
GMACx_T	Output	RGMIIx_TX	Data sending enable (rising edge) and data transmission error (falling edge)	RMIIx_TXEN	Data sending enable (
GMACx_R	Input	RGMIIx_RX	Data receiving	RMIIx_RXD[1:0]	Data receiving
GMACx_R	Input	RGMIIx_RX	Data receiving reference clock	—	—
GMACx_R	Input	RGMIIx_RX	Effective data receiving (rising edge) and receiving error (falling edge)	RMIIx_RXDV_CRS	Data receiving validity and carrier sense
GMACx_M	Input/Output	RGMIIx_MI	PHY sends 125MHz to MAC, optional	RMII_MCLKIN_50M or RMII_MCLKOUT_50	RMII data sending and data receiving reference clock
ETHx_REF	Output	ETHx_REF	RK3588 provides 25MHz clock to replace PHY crystal	ETHx_REFCLKO_25M	RK3588 provides 25MHz clock to replace PHY crystal
GMACx_M	Output	RGMIIx_MI	Manage the data clock	RMIIx_MDC	Manage the data clock
GMACx_M	Input/Output	RGMIIx_MI	Manage data output/input	RMIIx_MDIO	Manage data output/input

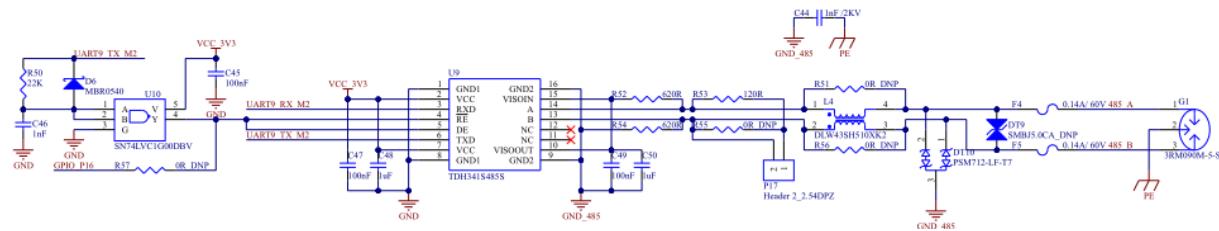
2. In RGMII mode, the TX/RX clock paths inside the RK3588 chip are integrated with a delay line, which supports adjustment. The default configuration in the reference diagram is as follows: The timing between TXCLK and data is controlled by the MAC, and the timing between RXCLK and data is controlled by the PHY. (For example, when using RTL8211F/FI, a 2ns delay for RXCLK is enabled by default. Please pay attention to other PHY configurations;
3. The GMAC0 interface level only supports 1.8V. The GMAC1 interface level is 3.3V by default (if it must be changed to 1.8V, please contact Forlinx). It should be noted whether the power - supply voltage of the RGMII signal power domain of the PHY chip matches the level of the GMACx interface;
4. The Reset signal of the Ethernet PHY needs to be controlled by GPIO. The level of the GPIO must match the PHY IO level. A 100nF capacitor must be added close to the PHY pins to enhance the anti - static ability. Note: The reset pin of RTL8211F/FI only supports a 3.3V level;
5. For TXD0 - TXD3, TXCLK, and TXEN, a 0ohm series resistor should be reserved at the FET3588 end to improve the signal quality conditionally according to the actual situation;
6. For RXD0 - RXD3, RXCLK, and RXDV, a 22ohm series resistor should be connected at the PHY end to improve the signal quality;
7. When the PHY uses an external crystal, the crystal capacitor should be selected according to the load capacitance value of the actually used crystal, and the frequency deviation should be controlled within ±20ppm;
8. The external resistor connected to the RBIAS pin of RTL8211F/FI is 2.49K ohms with an accuracy of 1%. Do not modify it casually;
9. An external pull - up resistor must be added to MDIO, with a recommended value of 1.5 - 1.8K ohm. The pull - up power supply must be consistent with the IO power supply;
- 10.The PCB layout needs to ensure the integrity of the RGMII signal reference plane and the integrity of the power supply reference plane around the PHY chip;

11.Equal - length requirement: The receiving and sending signals of RGMII can be grouped for equal - length processing, and the equal - length requirement is $\leq 12.5\text{mil}$;

12. Impedance requirement: 50 ohm for single - ended signals.

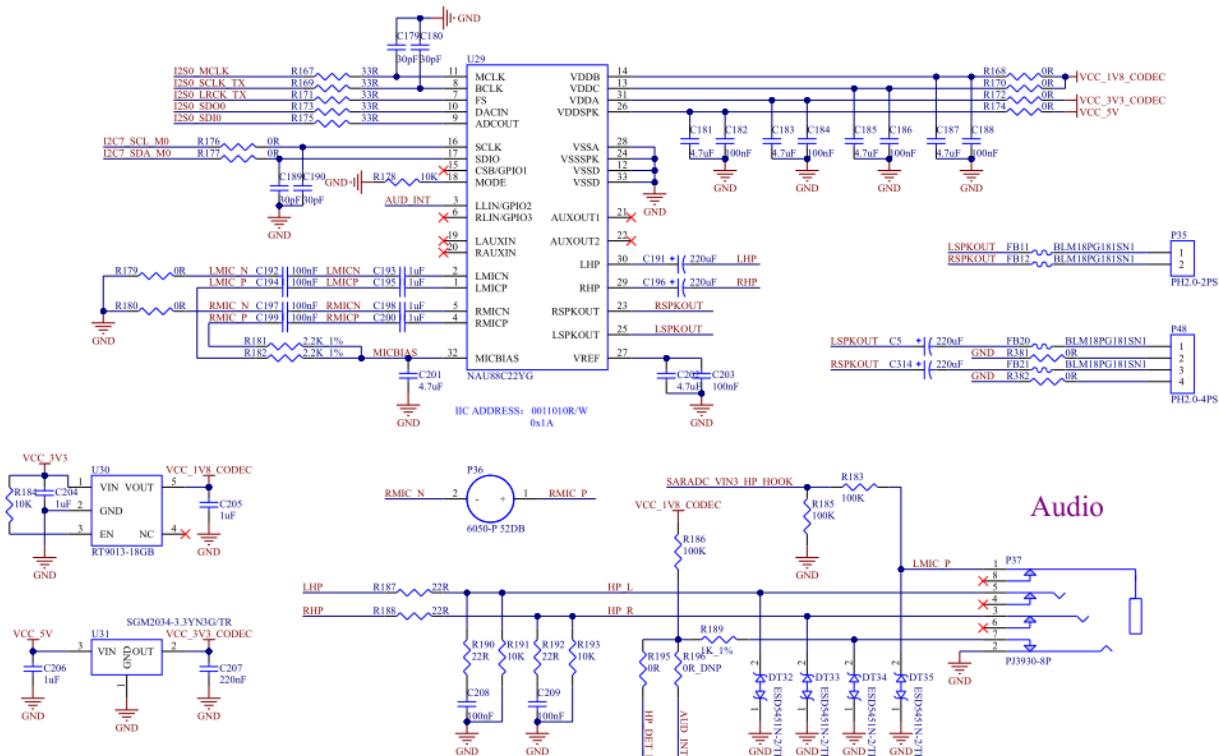
3.5.12 RS485 Interface

The OK3588 has an RS485 transceiver chip U9 on board, and its signal is TDH341S485S, with isolation withstand voltage up to 5000VDC, bus static protection up to 15kV (HBM), and transient immunity >25kV/us. Meanwhile, the OK3588 carrier board is compatible with a higher level of surge pulse group multi-level protection circuit, as shown in the following figure:



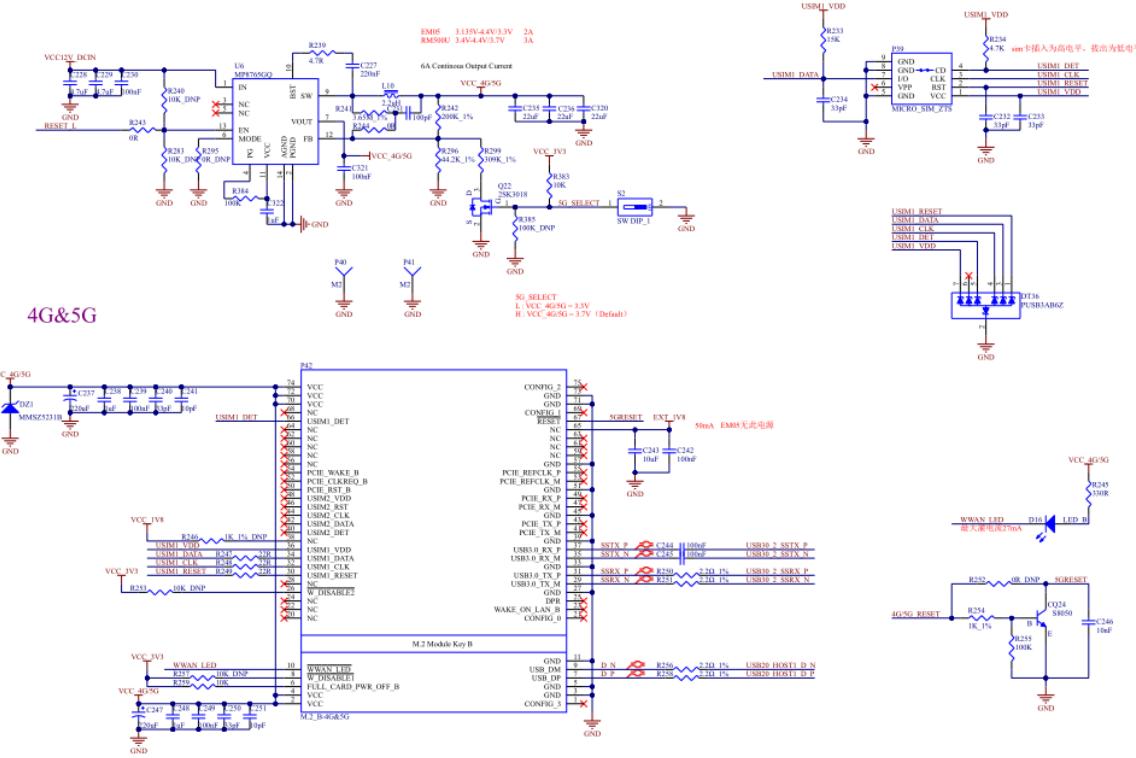
3.5.13 Audio

The OK3588 has an I2S interface Codec chip U29 on board, which supports MIC input, headphone output, and 1W 8Ω speaker output. The principle shown as follows:



3.5.14 4G&5G Interface

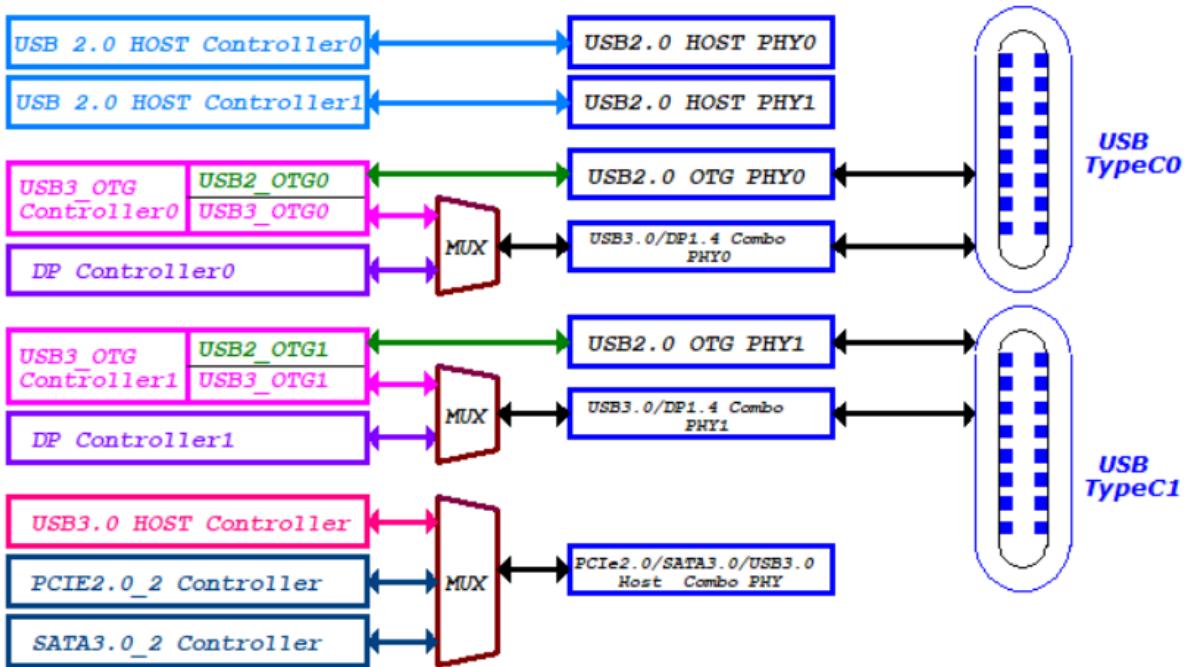
The OK3588 integrates an M.2 Key-B interface that is compatible with 4G and 5G modules. Since 4G and 5G modules have different power supply voltages, you need to toggle S2 to select the corresponding power supply voltage.



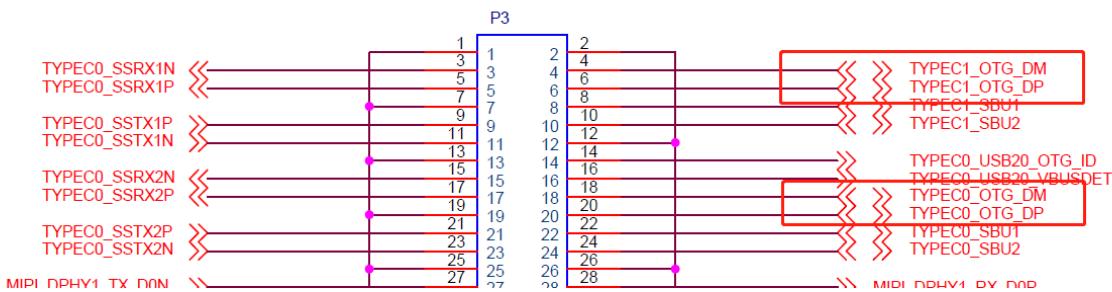
3.5.15 USB2.0/USB3.0 Circuit

The RK3588 chip has 2 x built-in USB 3.0 OTG controllers (with 2 x USB 2.0 OTG built-in, shown in green below), 1 x USB 3.0 HOST controller, and 2 x USB 2.0 HOST controllers.

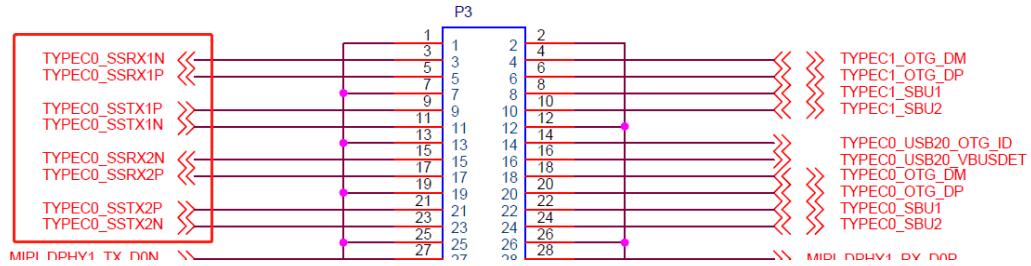
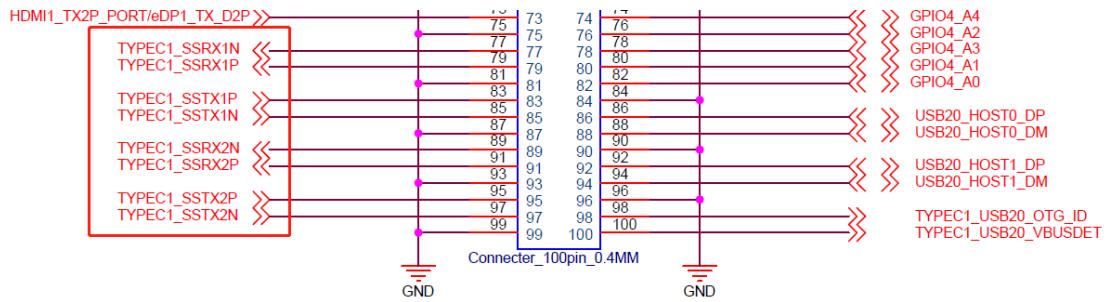
The internal multiplexing diagram of these controllers with the PHY is as follows:



USB3.0 OTGO controller supports SS/HS/FS/LS, and the embedded USB2.0 (HS/FS/LS) signal uses USB2.0 OTG PHY (the signal names are shown in the red box as follows); RK3588 currently only TYPECO_OTG_DM/TYPECO_OTG_DP supports Fireware Download, please be sure to reserve this interface in the application (at the same time TYPECO_USB20_VBUSDET must also be connected).



The SS signal (5Gbps) of USB3.0 is multiplexed with DP1.4, using the Combo PHY of USB/DP; the signal is in the red box as follows.



The multiplexing relationship of the USB and DP is as follows:Table

3.5.15.1 USB 3.0 and DP Multiplexing Relationship

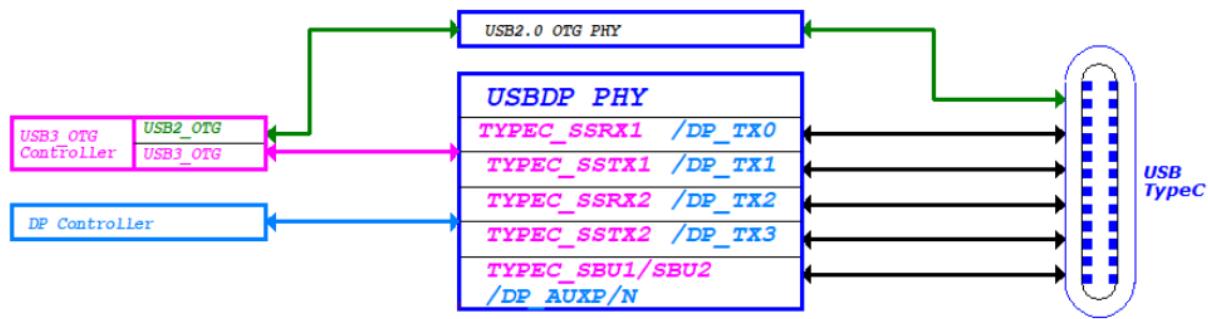
USB3.0	DP
TYPECx_SBU1	DPO_AUXP
TYPECx_SBU2	DPO_AUXN
TYPECx_SSRX1N	DPO_TXON
TYPECx_SSRX1P	DPO_TXOP
TYPECx_SSTX1P	DPO_TX1P
TYPECx_SSTX1N	DPO_TX1N
TYPECx_SSRX2N	DPO_TX2N
TYPECx_SSRX2P	DPO_TX2P
TYPECx_SSTX2P	DPO_TX3P
TYPECx_SSTX2N	DPO_TX3N

Since the USB3.0 OTG and USB2.0 OTG are the same USB3.0 controller, the USB3.0 and USB2.0 OTG can only do Device or HOST at the same time, not USB3.0 OTG for HOST, USB2.0 OTG for Device or USB3.0 OTG for Device and USB2.0 OTG to do HOST.

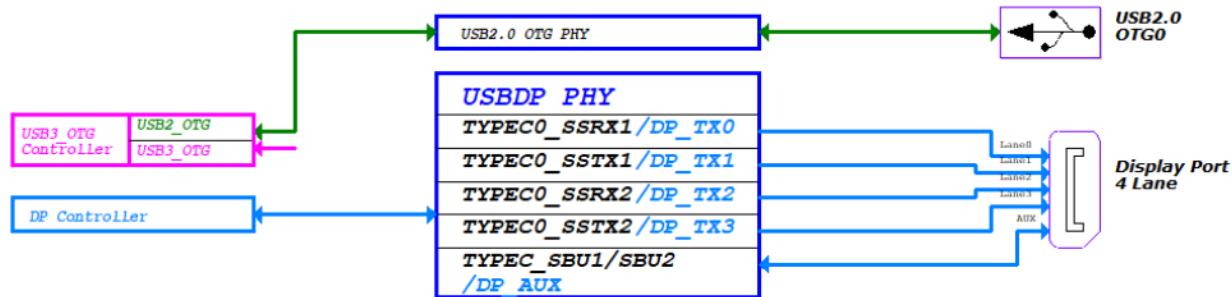
USB3.0 Controller and DP1.4 Controller are combined into a complete TYPEC port through USB3.0/DP1.4 Combo PHY, and this Combo PHY supports Display Alter mode. Lane0 and Lane2 do TX in DP mode and RX in USB mode; TX and RX share Lane0 and Lane2.

This USB3.0/DP1.4 Combo PHY supports inter-Lane switching (SWAP), so a TYPEC standard port can have the following five configurations:

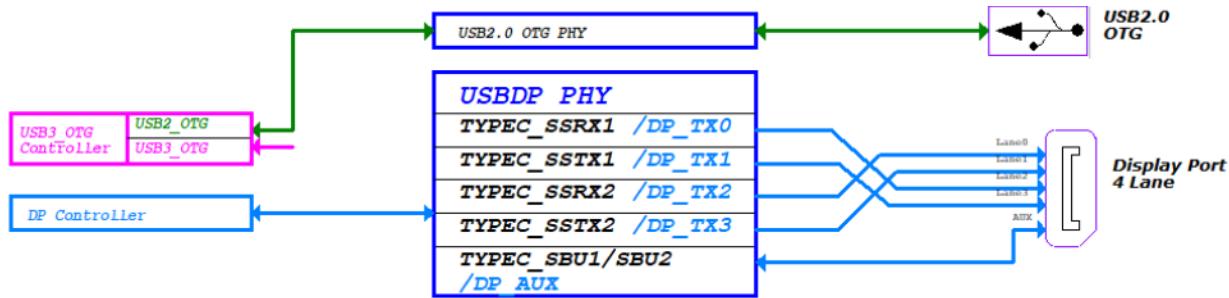
Configuration I: Type-C 4Lane (with DP function)



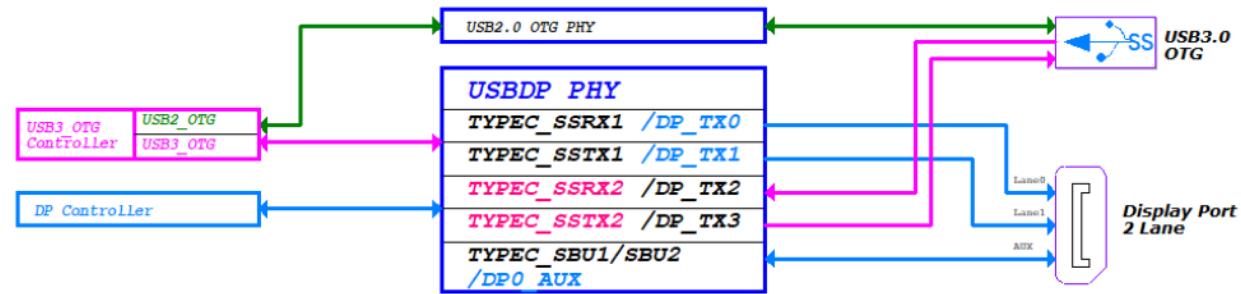
Configuration II: USB2.0 OTG+DP 4Lane (Swap OFF)



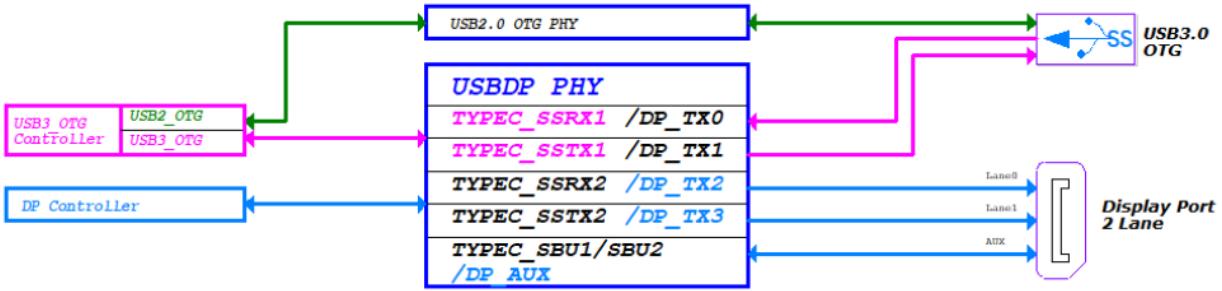
Configuration III: USB2.0 OTG+DP 4Lane(Swap ON)



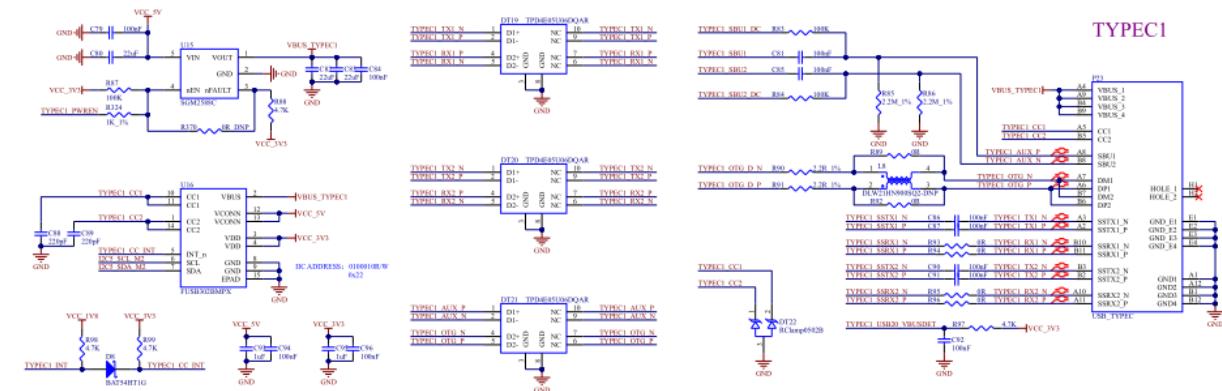
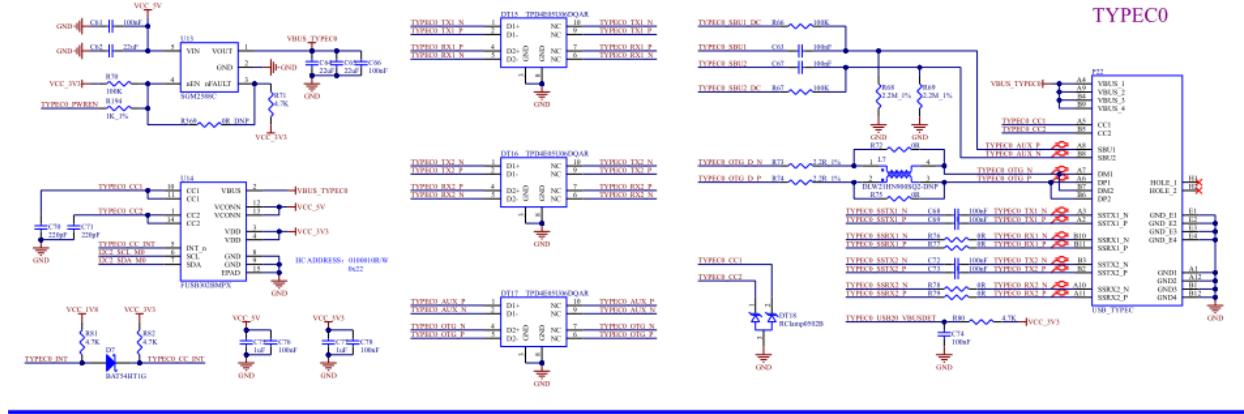
Configuration IV: USB3.0 OTG0+DP 2Lane (Swap OFF)



Configuration V: USB3.0 OTG+DP 2Lane (Swap ON)

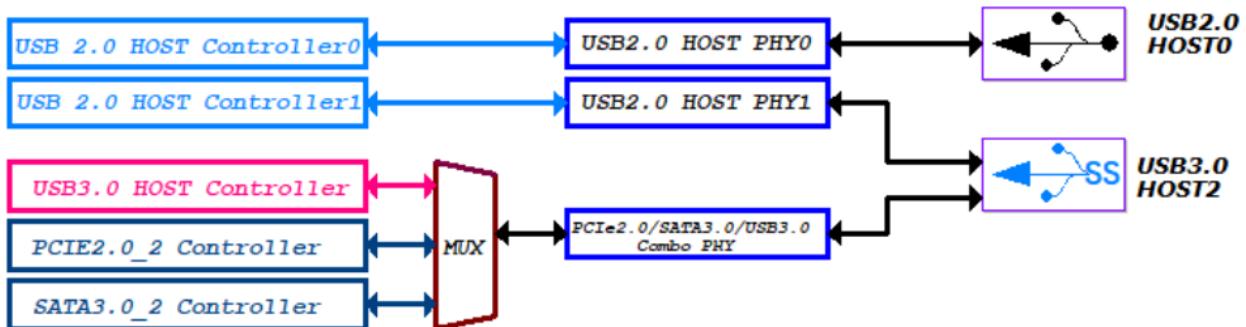


The default configuration of OK3588 is dual Type-C 4Lane (with DP function) , schematic is as follows:

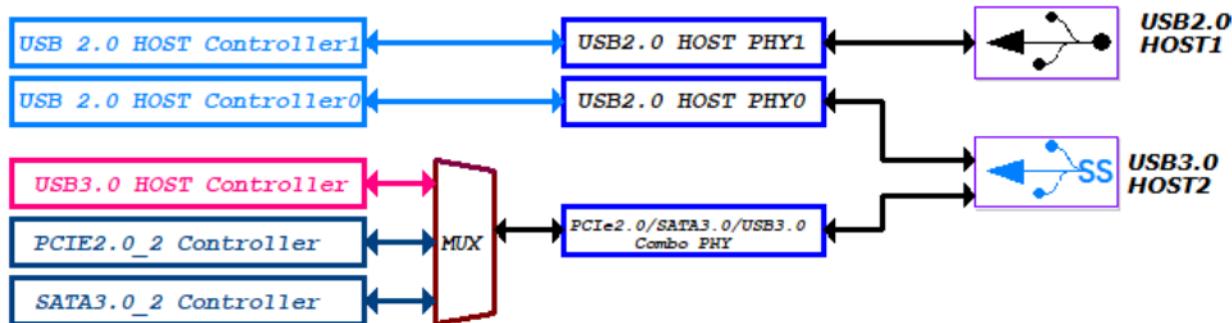


As the USB3.0 HOST controller only has USB3.0 HOST without built-in USB2.0, it needs to be combined with USB2.0 HOST Controller1 (configuration 1) or USB2.0 HOST Controller0 (configuration 2) to form a standard USB3.0 HOST. The internal link block diagram is as follows:

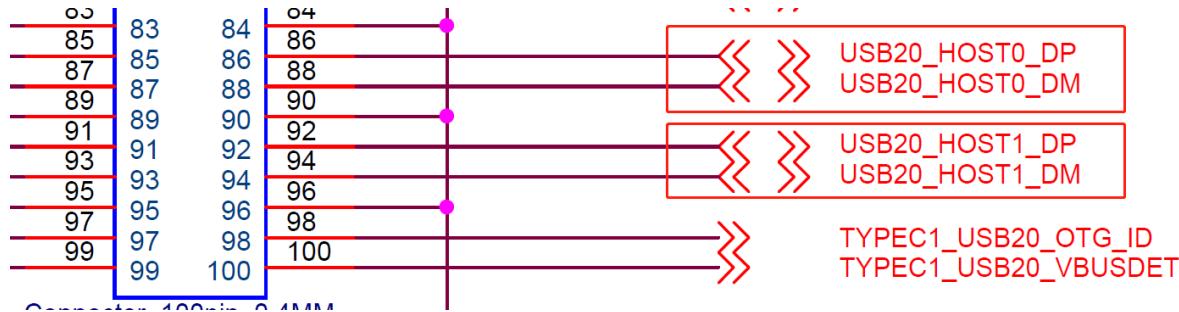
Configuration I: USB3.0 HOST2+USB2.0 HOST1



Configuration II: USB3.0 HOST2+USB2.0 HOST0



USB2.0 HOST controller, using USB2.0 HOST0PHY, the signals in the box below make up the USB2.0 HOST interface:



Note:

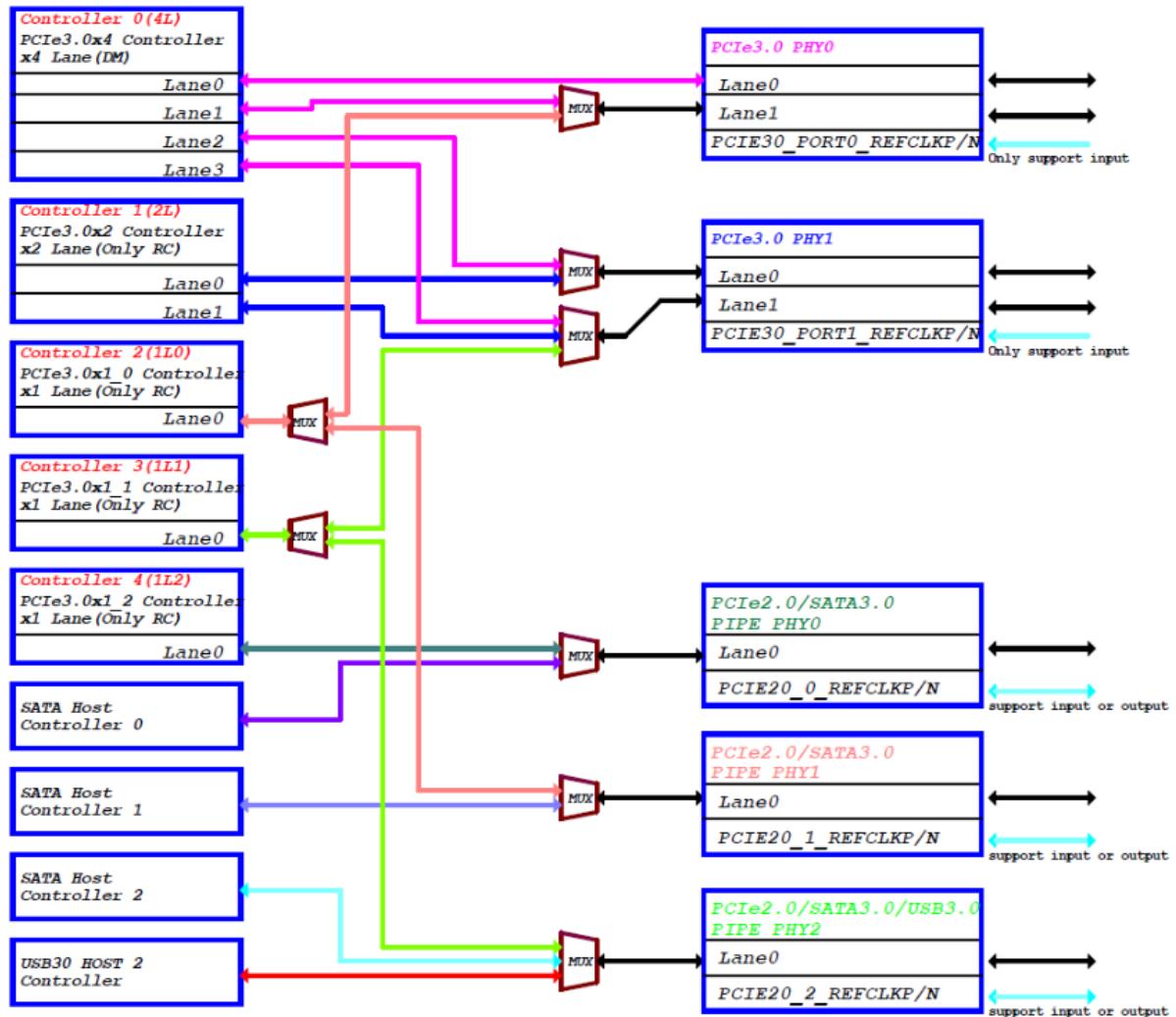
1. TYPEC0_OTG_DM/TYPEC0_OTG_DP is the system firmware programming port. If this interface is not used in the product, it must be reserved during the debugging and production processes. Otherwise, debugging and firmware programming during production will not be possible;
2. There is approximately a 200K ohm pull - up resistor to 1.8V inside the TYPEC0_USB20_OTG_ID;
3. TYPEC_USB20_VBUSDET is the detection pin for OTG and Device modes. It is active high, with a voltage range of 2.7 - 3.3V (Typical: 3.0V). It is recommended to place a 100nF capacitor at the pin;
4. The OTG mode can be set to the following three modes:
 - OTG mode: Automatically switch between the device mode and the HOST mode according to the state of the ID pin. When the ID is high, it is in the device mode; when the ID is pulled low, it is in the HOST mode. In the device mode, it will also check whether the VBUSDET pin is high (greater than 2.3V). Only when it is high will the DP be pulled high to start the enumeration process;
 - Device mode: When set to this mode, there is no need to use the ID pin. It only needs to check whether the VBUSDET pin is high (greater than 2.3V). Only when it is high will the DP be pulled high to start the enumeration process;
 - HOST mode: When set to this mode, there is no need to care about the states of the ID and VBUSDET pins. (In the case that the product only needs HOST mode, the system firmware burning port is only TYPEC0_OTG_DM/TYPEC0_OTG_DP, which is also used in the production and debugging, so when burning and abd debugging, we need to set it to device mode and connect TYPEC0_USB20_VBUSDET signal. If a TYPE - C interface is used, the TYPEC0_USB20_VBUSDE signals can be pulled high through a 4.7K resistor);
5. To enhance the anti - static and anti - surge capabilities, ESD devices must be reserved on the signals. The parasitic capacitance of the ESD for USB 2.0 signals shall not exceed 3pF. Additionally, a 2.2 - ohm resistor should be connected in series with the DP/DM of the USB 2.0 signals to enhance the anti - static and anti - surge capabilities;
6. To suppress electromagnetic radiation, a common - mode choke can be reserved on the signal lines. During the debugging process, a resistor or a common - mode choke can be selected according to the actual situation;
7. If the TYPECx_USB20_OTG_ID signal is used, to enhance the anti - static and anti - surge capabilities, ESD devices must be reserved on the signal, and a 100ohm resistor should be connected in series and shall not be removed;

8. When using the HOST function, it is recommended to add a current - limiting switch to the 5V power supply. The current - limiting value can be adjusted according to the application requirements. The current - limiting switch is controlled by a 3.3V GPIO. It is recommended to add capacitors of 22 μ F and over 100nF for filtering to the 5V power supply. If the USB port may be connected to a mobile hard disk, it is recommended to increase the filtering capacitor to over 100 μ F;
9. According to the TYPE - C protocol, a 100nF AC - coupling capacitor should be added to the SSTXP/N lines. It is recommended to use a 0201 package for the AC - coupling capacitor, which has lower ESR and ESL and can also reduce the impedance change on the line;
10. ESD devices must be added to all signals of the TYPE - C socket and should be placed close to the USB connector during layout. For the SSTXP/N and SSRXP/N signals, the parasitic capacitance of the ESD shall not exceed 0.3pF;
11. The differential impedance of USB 2.0 control signals should be 90 ohms \pm 10%, and the maximum time delay within the differential pair should be < 10mil;
12. The differential impedance of USB 3.0 control signals should be 90 ohms \pm 10%, and the maximum time delay within the differential pair should be < 3mil.

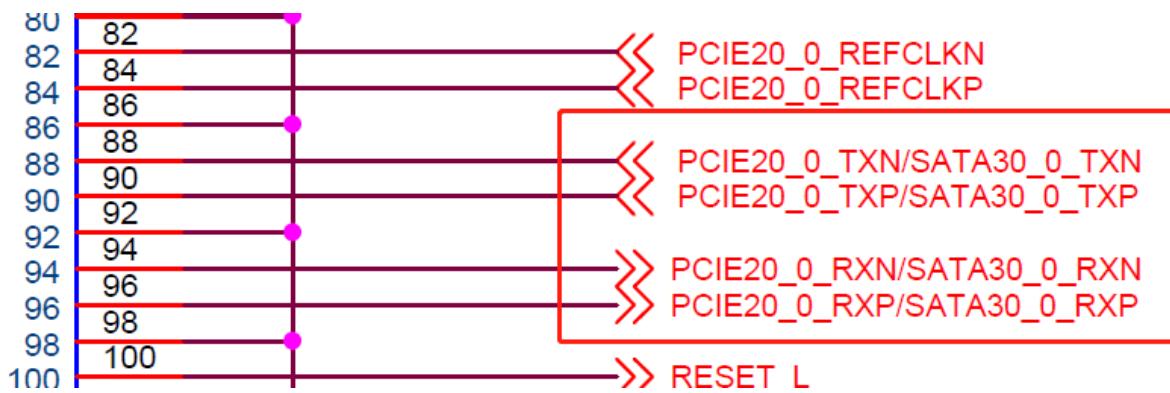
3.5.16 SATA3.0 Circuit

The RK3588 chip has 3 SATA3.0 controllers and multiplexes PIPE PHY0/1/2 with PCIe and USB3_HOST2 controllers. Please see the specific path as follows:

- Support SATA PM function, and each port can support up to 5 devices;
- Support SATA 1.5Gb/s, SATA 3.0Gb/s, SATA 6.0Gb/s speeds;
- Support eSATA.



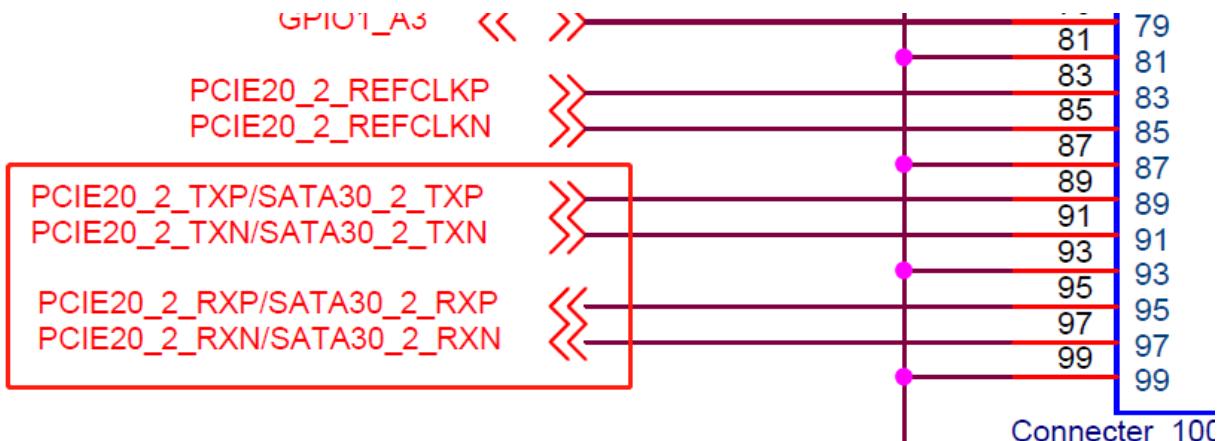
SATA0 controller uses PIPE_PHY0 (multiplexes with PCIe3.0x1_2 Controller).



SATA1 controller uses PIPE_PHY1 (multiplexes with PCIe3.0x1_0 Controller).



SATA2 controller uses PIPE_PHY2 (multiplexes with the PCIe3.0x1_1 Controller as well as the USB30 HOST2 Controller).



The control IO associated with the SATA0/1/2 controller are:

- SATA0_ACT_LED: SATA0 interface with LED flicker control output during data transfer;
- SATA1_ACT_LED: SATA1 interface with LED flicker control output during data transfer;
- SATA2_ACT_LED: SATA2 interface with LED flicker control output during data transfer;
- SATA_CP_DET: Plug detection input for SATA hot-plug devices;
- SATA_MP_SWITCH: Switch detection input for SATA hot-plug devices;
- SATA_CP POD: SATA control the power output switch of the hot plug device;

Among them, SATA_CP_DET, SATA_MP_SWITCH, and SATA_CP POD are shared interfaces for SATA0, SATA1, and SATA2. It can be configured through registers to select whether it is for SATA0, SATA1, or SATA2;

- SATA0_ACT_LED, SATA1_ACT_LED, SATA2_ACT_LED multiplexed pins are shown in the Pin Mux table.

Note:

During slot design, the peripheral circuitry and power supply must meet the specifications (Spec) requirements;

2. For the differential signals TXP/N and RXP/N of the SATA interface, a 10nF AC - coupling capacitor is connected in series. It is recommended to use a 0201 package for the AC - coupling capacitor, which has lower ESR and ESL and can also reduce the impedance change on the line;

3. ESD devices must be added to all signals of the eSATA interface socket. They should be placed close to the socket during layout, and the parasitic capacitance of the ESD shall not exceed 0.4pF.

3.5.17 PCIe2.0 and PCIe3.0 Circuit

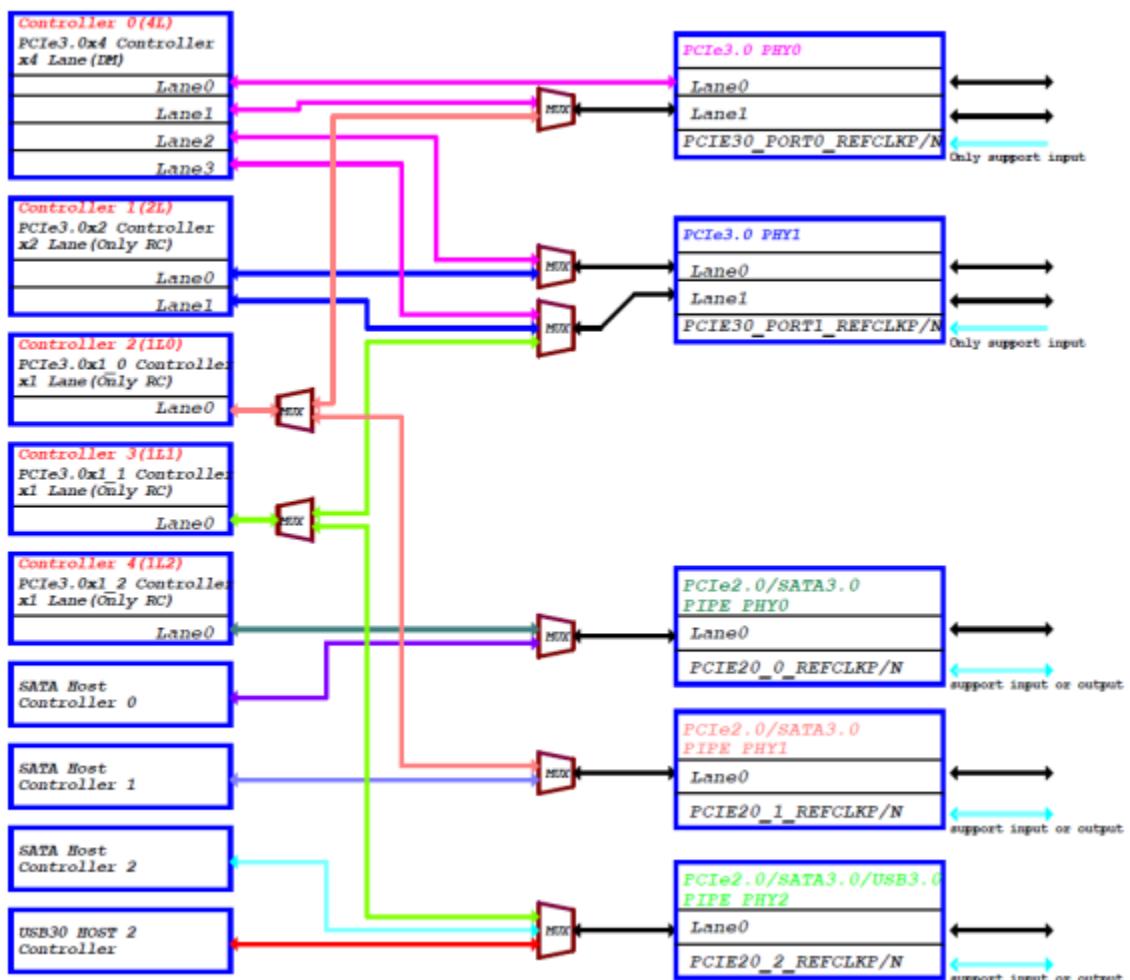
The RK3588 has 5 PCIe 3.0 controllers: (DM stands for Dual Mode and RC stands for Root Complex.)

1. Controller 0(4L), PCIe3.0x4 Controller x4 Lane(DM)
2. Controller 1(2L), PCIe3.0x2 Controller x2 Lane(Only RC)
3. Controller 2(1L0), PCIe3.0x1_0 Controller x1 Lane(Only RC)
4. Controller 3(1L1), PCIe3.0x1_1 Controller x1 Lane(Only RC)
5. Controller 4(1L2), PCIe3.0x1_2 Controller x1 Lane(Only RC)

2 PCIe3.0 PHY, data bit 2Lane, PCIe3.0 PHY0 and PCIe3.0 PHY1.

3 PCIe2.0 Combo PHY with data bit 1Lane, PCIe2.0/SATA3.0 Combo PHY0, PCIe2.0/SATA3.0 Combo PHY1 and PCIe2.0/SATA3.0/USB3.0 HOST Combo PHY2.

Mapping diagram between Controller and PHY:



Controller 0(4L) Lane0 can only be combined with PCIe3.0 PHY0 Lane0;

Controller 1(2L) Lane0 can only be combined with PCIe3.0 PHY1 Lane0;

Controller 0(4L) + PCIe3.0 PHY0 + PCIe3.0 PHY1 to form PCIe3.0 X4Lane RC or EP mode for 4Lane. It is compatible with PCIe3.0 X2Lane RC or EP mode and PCIe3.0 X1Lane RC or EP mode;

FET3588 PECL Signal	PCIe3.0 x 4Lane RC or EP	PCIe3.0 x 2Lane RC or EP	PCIe3.0 x 1Lane RC or EP
Port0	PCIE30_PORT0_TXOP/N ✓ PCIE30_PORT0_RXOP/N ✓	✓ ✓	✓ ✓
	PCIE30_PORT0_TX1P/N ✓	✓	?
	PCIE30_PORT0_RX1P/N ✓	✓	?
	PCIE30_PORT0_REFCLKI ✓	✓	✓
Port1	PCIE30_PORT1_TXOP/N ✓ PCIE30_PORT1_RXOP/N ✓	?	?
	PCIE30_PORT1_TX1P/N ✓	?	?
	PCIE30_PORT1_RX1P/N ✓	?	?
	PCIE30_PORT1_REFCLKI ✓	?	?

Controller 1(2L) + PCIe3.0 PHY1 to form 2Lane for PCIe3.0 X2Lane RC mode. It is compatible with PCIe3.0 X1Lane RC mode;

FET3588 PECL Signal	PCIe3.0 x 2Lane RC	PCIe3.0 x 1Lane RC
Port1	PCIE30_PORT1_TXOP/N ✓ PCIE30_PORT1_RXOP/N ✓	✓ ✓
	PCIE30_PORT1_TX1P/N ✓	?
	PCIE30_PORT1_RX1P/N ✓	?
	PCIE30_PORT1_REFCLKP/N_IN ✓	✓

Controller 2(1L0) + Lane1of PCIe3.0 PHY0 to form a PCIe3.0 X1Lane RC of 1Lane, or Controller 2(1L0) + PCIe2.0/SATA3.0 Combo PHY1 to form a PCIe2.0 X1Lane RC, so these two modes cannot be used at the same time;

? PCIe3.0 X1Lane RC mode corresponding signals in this mode:

FET3588 PECL Signal	PCIe3.0 x 1Lane RC
Port0	PCIE30_PORT0_TX1P/N ✓ PCIE30_PORT0_RX1P/N ✓
	PCIE30_PORT0_REFCLKP/N_IN ✓

? PCIe2.0 X1Lane RC mode corresponding signals in this mode:

FET3588 PECI Signal		PCIe3.0 x 1Lane RC
PCIe2.0/SATA3.0 Combo PHY1	PCIE20_1_TXP/N	✓
	PCIE20_1_RXP/N	✓
PCIE20_1_REFCLKP/N		✓

Controller 3(1L1) + Lane1 of PCIe3.0 PHY1 to form 1Lane of PCIe3.0 X1Lane RC mode, or Controller 3(1L1) + PCIe2.0/SATA3.0/USB3.0 HOST Combo PHY2 to form PCIe2.0 X1Lane RC mode, so these two modes cannot be used at the same time.

② PCIe3.0 X1Lane RC mode corresponding signals in this mode:

FET3588 PECI Signal		PCIe3.0 x 1Lane RC
Port1	PCIE30_PORT1_TX1P/N	✓
	PCIE30_PORT1_RX1P/N	✓
PCIE30_PORT1_REFCLKP/N_IN		✓

② PCIe2.0 X1Lane RC mode corresponding signals in this mode:

FET3588 PECI Signal		PCIe3.0 x 1Lane RC
PCIe2.0/SATA3.0/USB HOST Combo PHY2	PCIE20_2_TXP/N	✓
	PCIE20_2_RXP/N	✓
PCIE20_2_REFCLKP/N		✓

Controller 4(1L2)) + PCIe2.0/SATA3.0 Combo PHY0, to form the PCIe2.0 X1Lane RC mode for 1Lane;

② Corresponding signals in this mode:

FET3588 PECI Signal		PCIe3.0 x 1Lane RC
PCIe2.0/SATA3.0 Combo PHY0	PCIE20_0_TXP/N	✓
	PCIE20_0_RXP/N	✓
PCIE20_0_REFCLKP/N		✓

Multiple modes can be supported based on the above description. If all use the PCIe function, the RK3588 can support multiple combinations of PCIe modes(up to 5 modes can be used simultaneously).

The RK3588 PCIe multiple combination modes are as follows:

**eg1: 1 x PCIe3.0 X4Lane RC or EP
+ 3 x PCIe2.0 X1Lane RC**

PCIe3.0 x4Lane + PCIe2.0 x1Lane + PCIe2.0 x1Lane + PCIe2.0 x1Lane	Controller 0 (4L) RC or EP Controller 4 (1L2) RC Controller 2 (1L0) RC Controller 3 (1L1) RC	PCIe3.0 PHY0 Lane0+Lane1 + PCIe3.0 PHY1 Lane0+Lane1 PCIe2.0/SATA3.0 Combo PHY0 PCIe2.0/SATA3.0 Combo PHY1 PCIe2.0/SATA3.0/USB3.0 HOST Combo PHY2
---	---	--

**eg2: 1 x PCIe3.0 X2Lane RC or EP
+ 1 x PCIe3.0 X2Lane RC
+ 3 x PCIe2.0 X1Lane RC**

PCIe3.0 x2Lane + PCIe3.0 x2Lane + PCIe2.0 x1Lane + PCIe2.0 x1Lane	Controller 0 (4L) RC or EP Controller 1 (2L) RC Controller 4 (1L2) RC Controller 2 (1L0) RC Controller 3 (1L1) RC	PCIe3.0 PHY0 Lane0+Lane1 PCIe3.0 PHY1 Lane0+Lane1 PCIe2.0/SATA3.0 Combo PHY0 PCIe2.0/SATA3.0 Combo PHY1 PCIe2.0/SATA3.0/USB3.0 HOST Combo PHY2
---	--	---

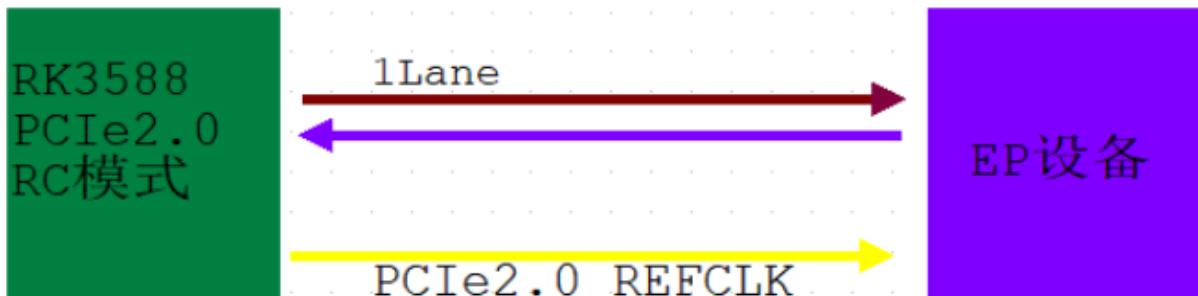
**eg3: 1 x PCIe3.0 X2Lane RC or EP
+ 1 x PCIe3.0 X1Lane RC
+ 3 x PCIe2.0 X1Lane RC**

PCIe3.0 x2Lane + PCIe3.0 x1Lane + PCIe2.0 x1Lane + PCIe2.0 x1Lane + PCIe2.0 x1Lane	Controller 0 (4L) RC or EP Controller 1 (2L) RC Controller 4 (1L2) RC Controller 2 (1L0) RC Controller 3 (1L1) RC	PCIe3.0 PHY0 Lane0+Lane1 PCIe3.0 PHY1 Lane0 PCIe2.0/SATA3.0 Combo PHY0 PCIe2.0/SATA3.0 Combo PHY1 PCIe2.0/SATA3.0/USB3.0 HOST Combo PHY2
---	--	---

**eg4: 4 x PCIe3.0 X1Lane RC
+ 1 x PCIe2.0 X1Lane RC**

PCIe3.0 x1Lane + PCIe3.0 x1Lane + PCIe3.0 x1Lane + PCIe2.0 x1Lane	Controller 0 (4L) RC Controller 2 (1L0) RC Controller 1 (2L) RC Controller 3 (1L1) RC Controller 4 (1L2) RC	PCIe3.0 PHY0 Lane0 PCIe3.0 PHY1 Lane1 PCIe3.0 PHY0 Lane0 PCIe3.0 PHY1 Lane1 PCIe2.0/SATA3.0 HOST Combo PHY0
---	--	--

PCIE20_REFCLKP/N can support either output or input, and the default output is provided to EP devices.

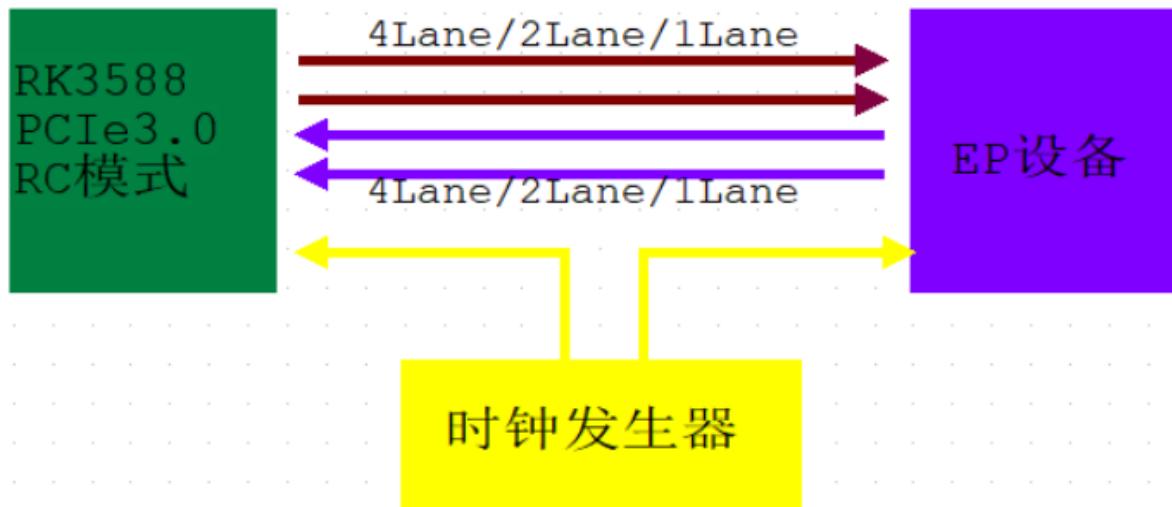


PCIE30_REF_CLKP/N only supports input

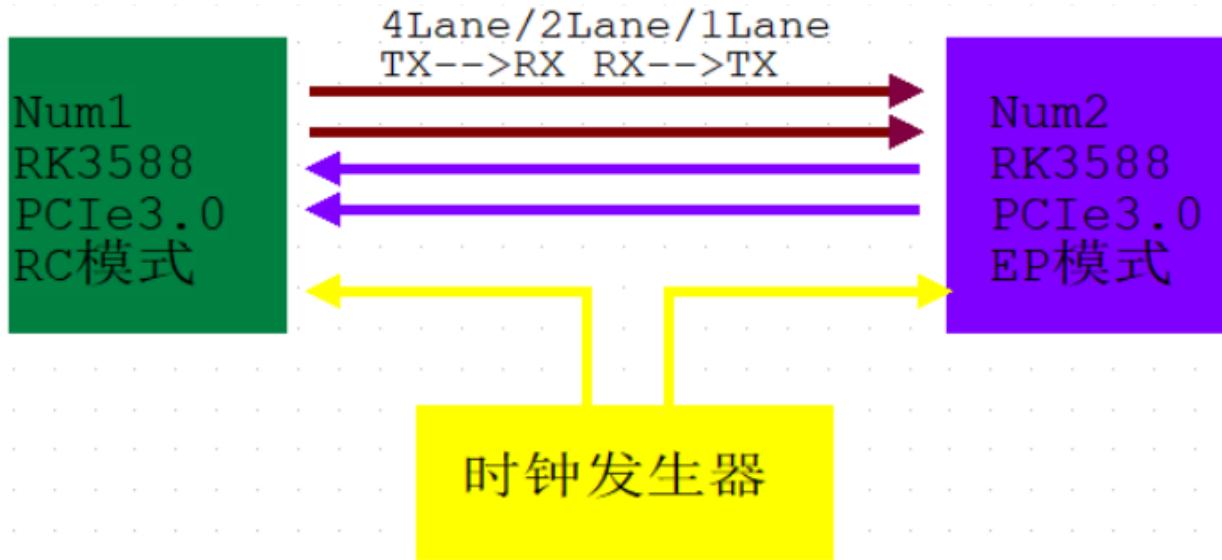
- HCSL electrical level clock input required

It is necessary to provide clocking requirements to meet PCIe 3.0 or higher

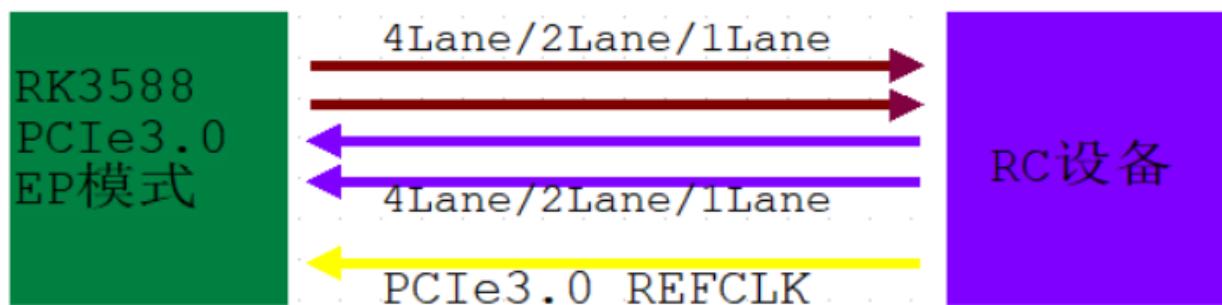
- RK3588 PCIe3.0 X4Lane RC mode. It is compatible with PCIe3.0 X2Lane RC mode and PCIe3.0 X1Lane RC mode. The reference clock path is as follows:



Another case, if it is 2 RK3588 cascade docking, equivalent to the above figure EP device is also RK3588. Reference clock path is the same, data Lane TX docking RX, RX docking TX.

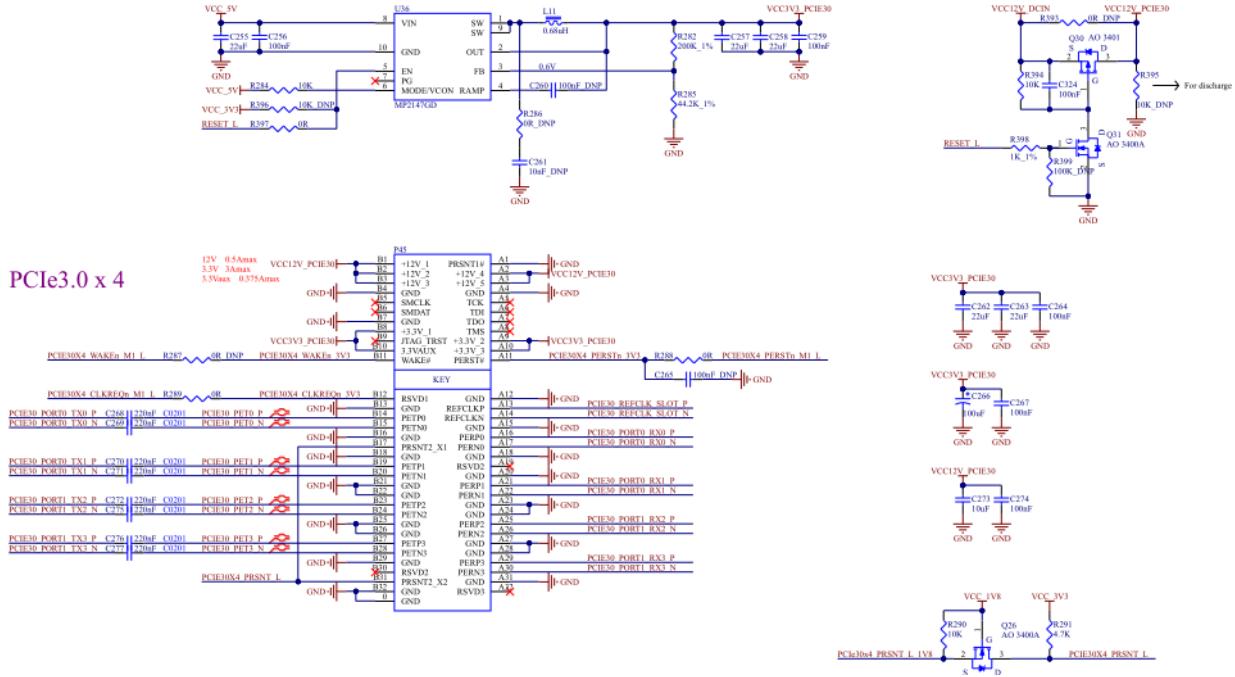


RK3588 PCIe 3.0 x4 Lane EP mode. It is compatible with PCIe3.0 X2Lane EP mode and PCIe3.0 X1Lane EP mode. The reference clock path is as follows:

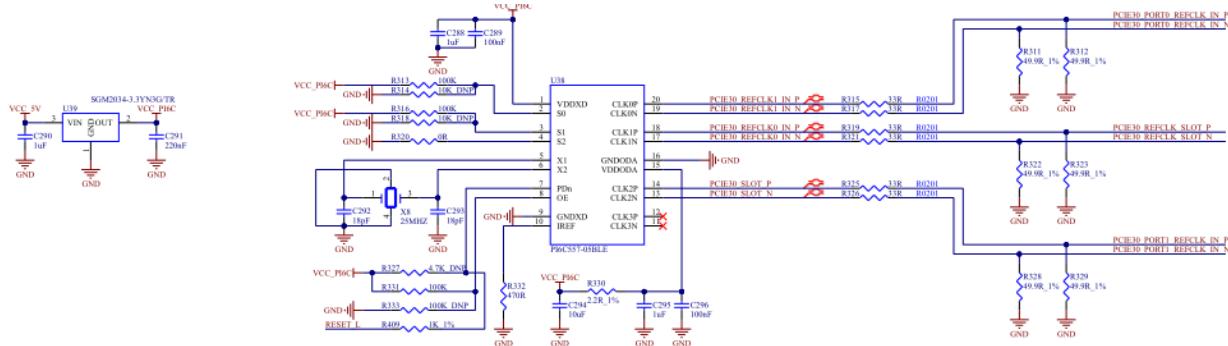


OK3588-C development board configuration is as follows:

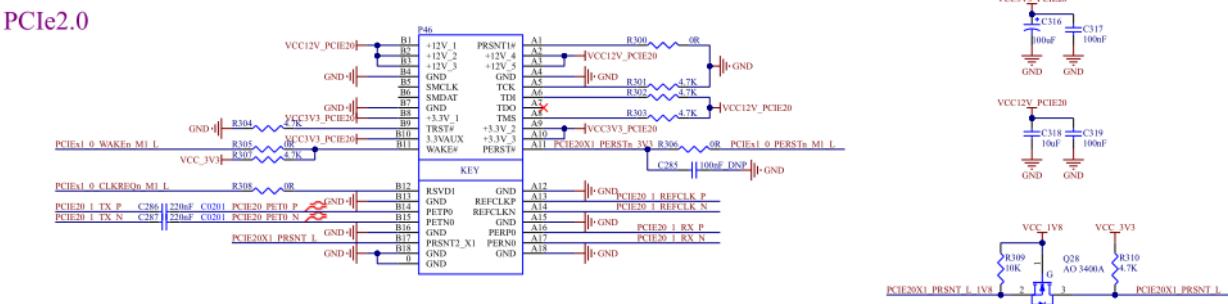
Controller 0(4L) + PCIe3.0 PHY0 + PCIe3.0 PHY1 to form PCIe3.0 X4Lane RC or EP mode for 4Lane. The principle is as follows:



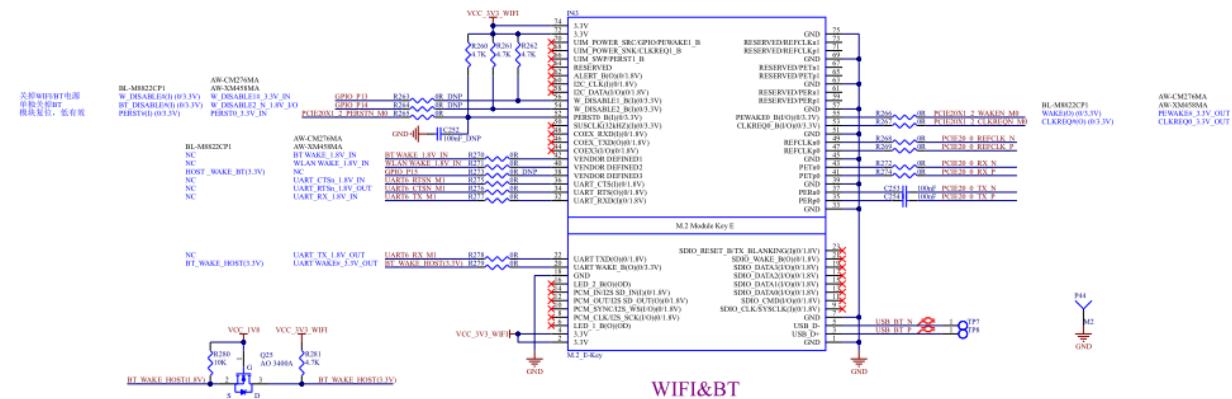
It should be noted that PCIE30_REFCLK_SLOT_P/N, PCIE30_PORT0_REFCLK_IN_P/N, PCIE30_PORT1_REFCLK_IN_P/N are generated by the clock chip U38, and the principle is as follows:



Controller 2(1L0) + PCIe2.0/SATA3.0 Combo PHY1 to form PCIe2.0 X1Lane RC mode. The principle is as follows:



Controller 4(1L2)) + PCIe2.0/SATA3.0 Combo PHY0, to form the PCIe2.0 X1Lane RC mode for 1Lane; The principle of a WIFI module compatible with PCIe interface led out by M.2 Key E , the principle is as follows:



PCIE2.0 Design Notes:

- During slot design, the peripheral circuitry and power supply must meet the specifications (Spec) requirements;
- For the TXP/N differential signals of the PCIe 2.0 interface, a 100 nF AC coupling capacitor is connected in series. It is recommended to use a 0201 package for the AC coupling capacitor due to its lower Equivalent Series Resistance (ESR) and Equivalent Series Inductance (ESL), which can also reduce impedance variations along the circuit;
- The PCIE2.0_CLKREQn and PCIE2.0_WAKEn signals must utilize the designated functional pins and cannot be substituted with GPIO pins. Special note: When selecting these pins, both must be chosen from the same group, i.e., either _M0, _M1, or _M2, and they cannot be mixed (e.g., one _M0 and one _M1);
- PCIE2.0_PERSTn can either utilize its designated functional pin or be substituted with a GPIO pin. However, when choosing the functional pin, it must be from the same _Mx group as PCIE2.0_CLKREQn and PCIE2.0_WAKEn;
- Voltage Levels:** PCIE2.0_CLKREQn, PCIE2.0_WAKEn, and PCIE2.0_PERSTn signals operate at a 3.3V level;
- PCIE2.0_PRSNT:** This is the Add In Card insertion detection pin and can be implemented using a GPIO;
- Multiplexing Consideration:** When utilizing PCIE2.0 functionality, the multiplexed SATA/USB30 functions cannot be used simultaneously. Refer to the corresponding functional module descriptions for SATA/USB30;
- If the PCIe 2.0 functional module is not in use, you can simply leave the data lines PCIE2.0_TXP/TXN, PCIE2.0_RXP/RXN and the reference clock lines PCIE2.0_REFCLKP/REFCLKN floating;
- Recommended Matching Design for PCIe2.0 Interface:**

Signal	Connection	Description
PCIE2.0_0/1/2_TXP/TX	Series Connection with 100nF Capacitor (0201 Package Recommended):	PCIe Data Output:
PCIE2.0_0/1/2_RXP/RXN	Direct Connection:	PCIe Data Input
PCIE2.0_0/1/2_REFCLK	Direct Connection:	PCIe Reference Clock:
PCIE2.0_CLKREQn	Series Connection with 0ohm Resistor:	PCIe Reference Clock Request Input (RC Mode):
PCIE2.0_WAKEn	Series Connection with 0ohm Resistor:	PCIe Wake Input (RC Mode):
PCIE2.0_PERSTn	Series Connection with 0ohm Resistor:	PCIe Global Reset Output (RC Mode):
PCIE2.0_PRSNT	Series Connection with 0ohm Resistor:	Add-In Card Insertion Detection Input (RC Mode):

- The impedance of the data traces should be controlled at a differential 85 ohms ±10%;
- The impedance of the clock traces should be controlled at a differential 100 ohms ±10%;

12. The maximum time - delay difference within differential pair should be < 3mil;

13. The spacing between differential pairs should be \geq 4 times the PCI-E trace width;

PCIe3.0 Design Notes:

1. During slot design, the peripheral circuitry and power supply must meet the specifications (Spec) requirements;
2. For PCIe3.0 interface differential signals TX0P/N, TX1P/N, a 220nF AC coupling capacitor should be connected in series. It is recommended to use a 0201 package for lower ESR and ESL, reducing impedance variations on the line;
3. The correspondence between PCIE30_CLKREQn, PCIE30_WAKEn, PCIE30_PERSTn, PCIE30X4_BUTTON_RSTN control signals and the controller should be as per the provided diagram:

PCIe Controller Configure Table

Controller Name	Data & Clk Lane Configure			Control GPIO
	OPTION	CLK LANE	DATA LANE	
PCIE30X4 RC & EP	OPTION1	PCIE30_PORT0_REF_CLKP PCIE30_PORT0_REF_CLKN	PCIE30_PORT0_TX0 PCIE30_PORT0_RX0	PCIE30X4_CLKREQ_M* PCIE30X4_WAKEN_M* PCIE30X4_PERSTN_M* PCIE30X4_BUTTON_RSTN
	OPTION2	PCIE30_PORT0_REF_CLKP PCIE30_PORT0_REF_CLKN	PCIE30_PORT0_TX0 PCIE30_PORT0_RX0 PCIE30_PORT0_TX1 PCIE30_PORT0_RX1	
	OPTION3	PCIE30_PORT0_REF_CLKP PCIE30_PORT0_REF_CLKN PCIE30_PORT1_REF_CLKP PCIE30_PORT1_REF_CLKN	PCIE30_PORT0_TX0 PCIE30_PORT0_RX0 PCIE30_PORT1_TX0 PCIE30_PORT1_RX0 PCIE30_PORT0_RX1 PCIE30_PORT1_RX1	
PCIE30X2 RC	OPTION1	PCIE30_PORT1_REF_CLKP PCIE30_PORT1_REF_CLKN	PCIE30_PORT1_TX0 PCIE30_PORT1_RX0	PCIE30X2_CLKREQ_M* PCIE30X2_WAKEN_M* PCIE30X2_PERSTN_M* PCIE30X2_BUTTON_RSTN
	OPTION2	PCIE30_PORT1_REF_CLKP PCIE30_PORT1_REF_CLKN	PCIE30_PORT1_TX0 PCIE30_PORT1_RX0 PCIE30_PORT1_TX1 PCIE30_PORT1_RX1	
PCIE30X1_0 RC	OPTION1	PCIE30_PORT0_REF_CLKP PCIE30_PORT0_REF_CLKN	PCIE30_PORT0_TX1 PCIE30_PORT0_RX1	PCIE30X1_0_CLKREQ_M* PCIE30X1_0_WAKEN_M* PCIE30X1_0_PERSTN_M* PCIE30X1_0_BUTTON_RSTN
	OPTION2	PCIE20_1_REFCLKP PCIE20_1_REFCLKN	PCIE20_1_TXP PCIE20_1_TXX PCIE20_1_RXP PCIE20_1_RXN	
PCIE30X1_1 RC	OPTION1	PCIE30_PORT1_REF_CLKP PCIE30_PORT1_REF_CLKN	PCIE30_PORT1_TX1 PCIE30_PORT1_RX1	PCIE30X1_1_CLKREQ_M* PCIE30X1_1_WAKEN_M* PCIE30X1_1_PERSTN_M* PCIE30X1_1_BUTTON_RSTN
	OPTION2	PCIE20_2_REFCLKP PCIE20_2_REFCLKN	PCIE20_2_TXP PCIE20_2_TXX PCIE20_2_RXP PCIE20_2_RXN	
PCIE20X1_2 RC	OPTION1	PCIE20_0_REFCLKP PCIE20_0_REFCLKN	PCIE20_0_TXP PCIE20_0_TXX PCIE20_0_RXP PCIE20_0_RXN	PCIE20X1_2_CLKREQ_M* PCIE20X1_2_WAKEN_M* PCIE20X1_2_PERSTN_M* PCIE20X1_2_BUTTON_RSTN

Note:

PCIE30_PORT*_REF_CLKP/N is input gpio

PCIE20_*_REFCLKP/N is output or input gpio

Note:

M*=Mean to M0 or M1, It's the same source, Just multiplex to M0 or M1, So, Only use one at the same time.

4. The PCIE30_CLKREQn and PCIE30_WAKEn must use the functional pins and cannot be replaced by GPIOs. Specifically, when making a selection, you must choose either all _M0, all _M1, or all _M2. You cannot choose one _M0 and one _M1;
5. PCIE30_PERSTn can either use its designated functional pin or be replaced with a GPIO pin. However, when choosing the functional pin, it must be from the same _Mx group as PCIE30_CLKREQn and PCIE30_WAKEn;
6. Standard PCIe: PCIE30X2_CLKREQn, PCIE30X1_WAKEn, and PCIE30_PERSTn signals operate at a 3.3V level;
7. The PCIE30_PRSNT is the Add In Card insertion detection pin and can be implemented using a GPIO;
8. The PCIE30_BUTTON_RSTN is for the external hardware reset and is reserved for future use and not to be used for now;

9. When two RK3588 PCIe are cascaded, the data lines should be cross - connected, that is, TX→RX and RX→TX. Control signals PCIE30_CLKREQn and PCIE30_PERSTn are connected one-to-one (e.g. Num1 and Num1 represent two RK3588 respectively).

Num1_PCIE30_CLKREQn connects to Num2_PCIE30_CLKREQn, Num1_PCIE30_PERSTn connects to Num2_PCIE30_PERSTn). three signals, PCIE30_WAKEn, PCIE30_PRSNT and PCIE30_BUTTON_RSTN may keep floating;

10. The PCIe30 functional module is not used. The data cables PCIE30_TXP/TXN and PCIE30_RXP/RXN are left floating. The reference clock lines PCIE30_REFCLKP/REFCLKN are either grounded or left floating;

11. **The REFCLKP/N of the PCIe30 PHY and Slot/peripherals need to meet the requirement of the same - source clock. For example, in the design shown in the reference diagram, the three - way REFCLKP/N of PHY0/PHY1 and Slot are output from the same clock generator;

12. Recommended Matching Design for PCIe3.0 Interface:

Signal	Connection	Description
PCIE30_TX0P/	Series Connection with 220nF Capacitor (0201 Package Recommended):	PCIe Data Output:
PCIE30_RX0P/	Direct Connection:	PCIe Data Input
PCIE30_TX1P/	Series Connection with 220nF Capacitor (0201 Package Recommended):	PCIe Data Output:
PCIE30_RX1P/	Direct Connection:	PCIe Data Input
PCIE30_REFCL	Direct Connection:	PCIe Reference Clock Input
PCIE30_CLKRE	Series Connection with 0ohm Resistor:	PCIe Reference Clock Request Input (RC Mode)PCIe Reference Clock Request Output (EP Mode)
PCIE30_WAKE	Series Connection with 0ohm Resistor:	PCIe Wake Input (RC Mode)PCIe Wake Output (EP Mode)
PCIE30_PERST	Series Connection with 0ohm Resistor:	PCIe Global Reset Output (RC Mode)PCIe Global Reset Input (EP Mode)
PCIE30_PRSNT	Series Connection with 0ohm Resistor:	Add-In Card Insertion Detection Input (RC Mode):
PCIE30_BTTC	Series Connection with 0ohm Resistor:	PCIe External Hardware Reset Output (RC Mode)PCIe External Hardware Reset Input (EP Mode)

13. Data Trace Impedance: Differential $85\text{ohm}\pm10\%$;

14. Clock Trace Impedance: Differential $100\text{ohm}\pm10\%$;

15. Maximum Skew Within Differential Pair: < 3mil;

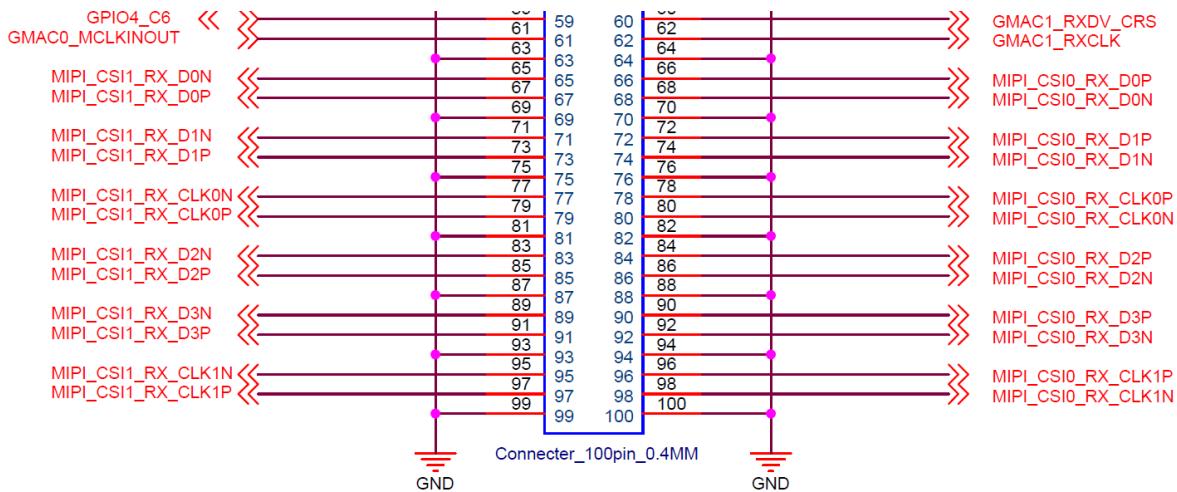
16. Recommended Spacing Between Differential Pairs: ≥ 5 times the PCI-E trace width.

3.5.18 Video Input Interface

RK3588 video input interface includes three kinds of interfaces: MIPI RX, CIF, HDMI. Among them, MIPI RX includes two groups of interfaces, RXMIPI DPHY CSI RX and MIPI_D/CPHY_RX. A detailed description of each interface function is as follows

3.5.18.1 MIPI DPHY CSI RX

RK3588 has two MIPI DPHY CSI RX, both support MIPI V1.2 version, the maximum data rate of each channel is 2.5Gbps. The FET3588-C MIPI DPHY CSI RX pin division is as follows:



MIPI DPHY CSI0 RX interface mode:

Support x4Lane mode, MIPI_CS10_D[3:0] data reference MIPI_CS10_CLK0

Support x2Lane+x2Lane mode:

- ② MIPI0_CSI_D[1:0] data reference MIPI_CS10_CLK0;
- ② MIPI0_CSI_D[3:2] data refer to MIPI_CS10_CLK1.

Two modes options:

Option1	Sensor1 x4Lane	MIPI_CS1_RX_D0-3MIPI_CS1_RX_CLK0
Option2	Sensor1 x2Lane+Sensor2 x2Lane	MIPI_CS1_RX_D0-1MIPI_CS1_RX_CLK0 MIPI_CS1_RX_D2-3MIPI_CS1_RX_CLK1

MIPI DPHY CSI1 RX interface mode:

- Support x4Lane mode, MIPI_CS1_D[3: 0] data reference MIPI_CS1_CLK0
 - Support x2Lane+x2Lane mode:
- ② MIPI1_CSI_D[1:0] data reference MIPI_CS1_CLK0;
 - ② MIPI1_CSI_D[3:2] data refer to MIPI_CS1_CLK1.

Option1	Sensor1 x4Lane	MIPI_CS1_RX_D0-3MIPI_CS1_RX_CLK0
Option2	Sensor1 x2Lane+Sensor2 x2Lane	MIPI_CS1_RX_D0-1MIPI_CS1_RX_CLK0 MIPI_CS1_RX_D2-3MIPI_CS1_RX_CLK1

3.5.18.2 MIPI_D/CPHY_RX

RK3588 has two MIPI D-PHY/C-PHY CSI RX Combo PHY

- D-PHY supports V1.2, D-PHY mode with 0/1/2/3 Lane, data transfer rate up to 2.5Gbps;
- C-PHY supports V1.1 version, C-PHY mode has 0/1/2 Trio, 3 lines per Trio A/B/C, data transfer rate up to 5.7Gbps/Trio (2.5Gbps).

The OK3588 is configured for D-PHY function by default and the signal pins are as follows. If you need to configure it for C-PHY, please check the Pin Mux table for the multiplexing key.

28		MIPI_DPHY1_RX_D0P
30		MIPI_DPHY1_RX_D0N
32		MIPI_DPHY1_RX_D1P
34		MIPI_DPHY1_RX_D1N
36		MIPI_DPHY1_RX_CLKP
38		MIPI_DPHY1_RX_CLKN
40		MIPI_DPHY1_RX_D2P
42		MIPI_DPHY1_RX_D2N
44		MIPI_DPHY1_RX_D3P
46		MIPI_DPHY1_RX_D3N
50		MIPI_DPHY0_RX_D0P
52		MIPI_DPHY0_RX_D0N
54		MIPI_DPHY0_RX_D1P
56		MIPI_DPHY0_RX_D1N
58		MIPI_DPHY0_RX_D2P
60		MIPI_DPHY0_RX_D2N
62		MIPI_DPHY0_RX_D3P
64		MIPI_DPHY0_RX_D3N
66		MIPI_DPHY0_RX_CLKP
68		MIPI_DPHY0_RX_CLKN
70		MIPI_DPHY0_RX_D0P
72		MIPI_DPHY0_RX_D0N
74		MIPI_DPHY0_RX_D1P
76		MIPI_DPHY0_RX_D1N
78		MIPI_DPHY0_RX_D2P
80		MIPI_DPHY0_RX_D2N
82		MIPI_DPHY0_RX_D3P
84		MIPI_DPHY0_RX_D3N
86		

DPHY and CPHY configuration support:

The RK3588 has two D/CPHY Combo PHY, both supporting RX and TX interfaces, CPU pins are as follows:



- MIPI D-PHY/C-PHY Combo PHY0 TX and RX can only be configured as DPHY0 TX, DPHY0 RX mode or CPHY0 TX, CPHY0 RX mode at the same time, and it does not support one configured as DPHY0 TX and another configured as CPHY0 RX;
- The TX and RX of MIPI D-PHY/C-PHY Combo PHY1 can only support being configured simultaneously in the DPHY1 TX and DPHY1 RX modes, or simultaneously in the CPHY1 TX and CPHY1 RX modes. It does not support one being configured as DPHY1 TX and the other as CPHY1 RX.

MIPI D/C-PHY0 support (In D-PHY time mode):

- Support x4Lane mode, MIPI_DPHY0_RX_D[3:0] data reference MIPI_DPHY0_RX_CLK;
- Not support splitting into x2Lane+x2Lane mode;

MIPI D/C-PHY0 support (In C-PHY time mode):

- Support 0/1/2 Trio, 3 wires per Trio A/B/C, MIPI_CPHY0_RX_TRI0[2:0]_A, MIPI_CPHY0_RX_TRI0[2:0]_B, MIPI_CPHY0_RX_TRI0[2:0]_C.

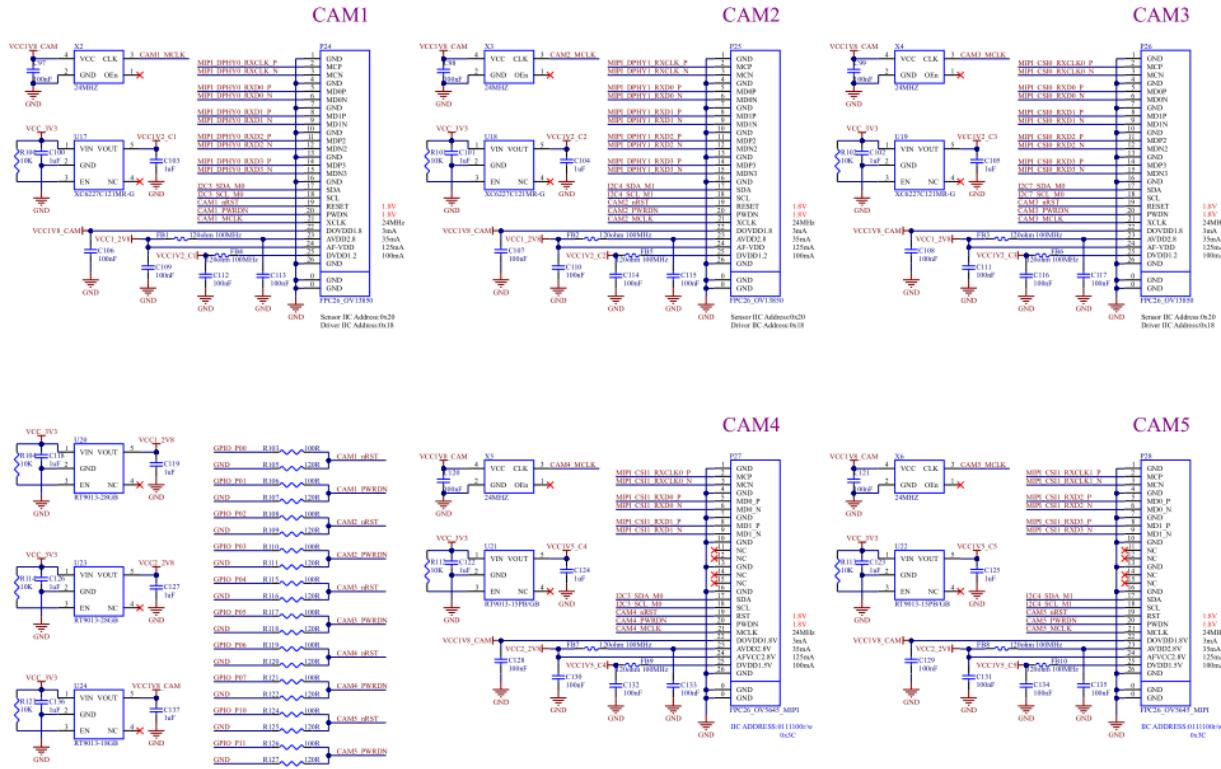
MIPI D/C-PHY1 support (In D-PHY time mode):

- Support x4Lane mode, MIPI_DPHY1_RX_D[3:0] data reference MIPI_DPHY1_RX_CLK;
- Not support splitting into x2Lane+x2Lane mode;

MIPI D/C-PHY1 support (In C-PHY time mode):

- Support 0/1/2 Trio, 3 wires per Trio A/B/C, MIPI_CPHY1_RX_TRI0[2:0]_A, MIPI_CPHY1_RX_TRI0[2:0]_B, MIPI_CPHY1_RX_TRI0[2:0]_C.

The OK3588 is configured with 5 Camera interfaces by default: MIPI_DPHY0_RX 4Lane, MIPI_DPHY1_RX 4Lane, MIPI_CSI0 4Lane, MIPI_CSI1 2Lane + MIPI_CSI1 2Lane. The principle is as follows:



MIPI RX Design Considerations:

1. Differential trace impedance requirement: $100 \Omega \pm 10\%$;
 2. Single-ended trace impedance requirement: $50 \Omega \pm 10\%$;
 3. Maximum time - delay difference within differential pair: < 3mil;
 4. Equal length between clock and data: < 6mil;
 5. Spacing between differential pairs should be $> 4 \times$ MIPI trace, and at a minimum, it should be 3 times the MIPI trace;
 6. Spacing between MIPI and other signals should be $> 4 \times$ MIPI trace, and at a minimum, it should be 3 times the MIPI trace;
 7. For CPHY configuration, maximum delay difference within the group (TRIO_A \ TRIO_B \ TRIO_C): < 3mil;
 8. Length matching requirement between groups (TRIO0\TRIO1\TRIO2) should be < 50mil.

3.5.18.3 CIF

Pin Multiplexing: Refer to the Pin Mux table for CIF interface reuse configurations. **Voltage Level:** The CIF interface operates at 3.3V; level shifting must be implemented to match the Camera module's actual IO power supply requirements.

CIF support format:

- Support BT601 YCbCr 422 8bit input;
 - Support BT656 YCbCr 422 8bit input;
 - Support RAW 8/10/12bit input;
 - Support BT1120 YCbCr 422 8/16bit input. single/dual-edge sampling;

- Support 2/4 mixed BT656/BT1120 YCbCr 422 8/16bit input;
- Support YUYV sequential configuration;

The 8/10/12/16bit data correspondence of CIF[15:0] is shown in the table below, using high bit alignment.

Mode	16bit	12bit	10bit	8bit
CIF_D0	D0	—	—	—
CIF_D1	D1	—	—	—
CIF_D2	D2	—	—	—
CIF_D3	D3	—	—	—
CIF_D4	D4	D0	—	—
CIF_D5	D5	D1	—	—
CIF_D6	D6	D2	D0	—
CIF_D7	D7	D3	D1	—
CIF_D8	D8	D4	D2	D0
CIF_D9	D9	D5	D3	D1
CIF_D10	D10	D6	D4	D2
CIF_D11	D11	D7	D5	D3
CIF_D12	D12	D8	D6	D4
CIF_D13	D13	D9	D7	D5
CIF_D14	D14	D10	D8	D6
CIF_D15	D15	D11	D9	D7

Data correspondence relationship in BT1120 16-bit mode, supporting YC Swap.

Pin Name	Default mode			Swap open:
	Pixel #0		Pixel #1	
CIF_D0	Y0[0]	Y1[0]	Cb0[0]	
CIF_D1	Y0[1]	Y1[1]	Cb0[1]	
CIF_D2	Y0[2]	Y1[2]	Cb0[2]	
CIF_D3	Y0[3]	Y1[3]	Cb0[3]	
CIF_D4	Y0[4]	Y1[4]	Cb0[4]	
CIF_D5	Y0[5]	Y1[5]	Cb0[5]	
CIF_D6	Y0[6]	Y1[6]	Cb0[6]	
CIF_D7	Y0[7]	Y1[7]	Cb0[7]	
CIF_D8	Cb0[0]	Cr0[0]	Y0[0]	
CIF_D9	Cb0[1]	Cr0[1]	Y0[1]	
CIF_D10	Cb0[2]	Cr0[2]	Y0[2]	
CIF_D11	Cb0[3]	Cr0[3]	Y0[3]	
CIF_D12	Cb0[4]	Cr0[4]	Y0[4]	
CIF_D13	Cb0[5]	Cr0[5]	Y0[5]	
CIF_D14	Cb0[6]	Cr0[6]	Y0[6]	
CIF_D15	Cb0[7]	Cr0[7]	Y0[7]	

The CIF interface pull-up and pull-down, and matching design recommendations are as follows:

Signal	Internal Pull-up&down	Connection	Description (Chip side)
CIF_D[1:0]	Pull-down	Direct connection, it is recommended to leave a series resistance near the device end	CIF data input
CIF_HRE	Pull-down	Direct connection, it is recommended to leave a series resistance near the device end	CIF data input
CIF_VSYI	Pull-down	Direct connection, it is recommended to leave a series resistance near the device end	CIF data input
CIF_CLKI	Pull-down	Connect 22ohm resistors in series, near the device side	CIF data input
CIF_CLKO	Pull-down	Connect 22ohm resistor in series, near the chip side	CIF clock output, which can be provided to the device as MCLK

Camera Design Considerations:

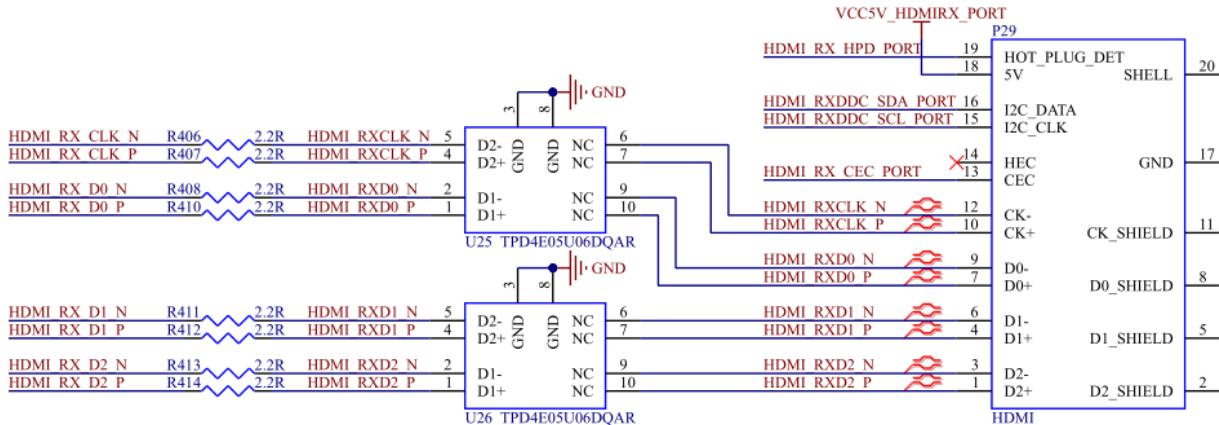
1. The Camera's DVDD power supply may vary (e.g., 1.2V/1.5V/1.8V). Provide the correct voltage as specified in the Camera datasheet (default reference: 1.2V);
2. For Cameras with high DVDD current (>100mA), it's recommended to use DCDC for power supply;
3. Follow the Camera's specified power-up sequence (default reference sequence: 1.8V → 1.2V → 2.8V). Adjust as per the datasheet;
4. The CIF interface defaults to 3.3 V, and level matching needs to be considered;
5. If the Camera includes an autofocus (AF) function, provide a separate VCC2V8_AF supply or share it with AVCC2V8_DVP using a ferrite bead for isolation;
6. All decoupling capacitors for the power supplies of the Camera must be retained and placed close to the connector (or socket) ;
7. The Camera's PWDN (power-down) signal must be controlled via a GPIO. Ensure the GPIO voltage level matches the Camera's IO level;
8. It is recommended to use GPIO to control the Reset signal of the Camera. The GPIO level must match the Camera IO level. The 100nF capacitor of the Reset signal should not be removed. Place it close to the connector to enhance the anti - static ability;
9. MCLK Clock Source Options:
 - (1) CIF_CLKOUT
 - (2) MIPI_CAMERA0_CLK
 - (3) MIPI_CAMERA1_CLK
 - (4) MIPI_CAMERA2_CLK
 - (5) MIPI_CAMERA3_CLK
 - (6) MIPI_CAMERA4_CLK

The clock level must match the Camera IO level. If not, the level must be matched by level conversion or resistance voltage division;
10. If two Cameras are of the same model, pay attention to whether the I2C address is the same. If the address is the same, two I2C buses are needed.

3.5.18.4 HDMI2.0 RX

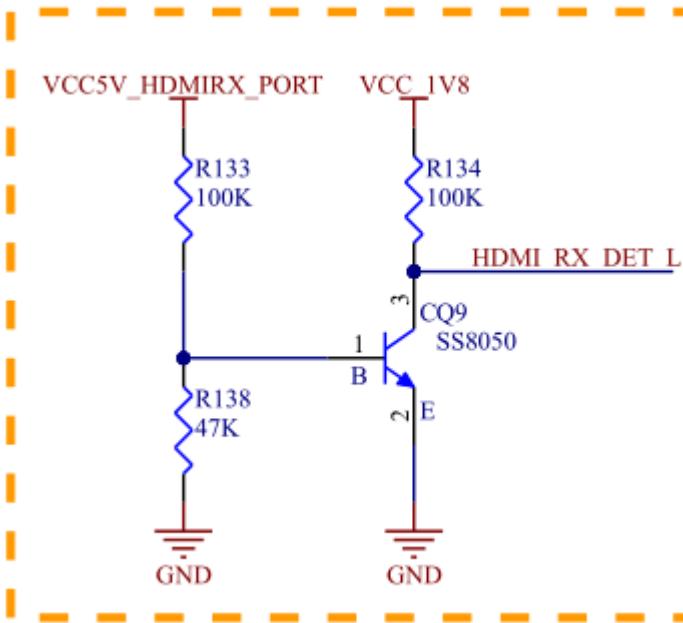
RK3588 chip supports HDMI 2.0 RX, downward compatible with HDMI 1.4b; supports RGB/YUV444/YUV422/YUV420 formats; up to 4K@60Hz input

HDMI RX TMDS signal is shown below and requires a 2.2ohm resistor to be reserved near the HDMI RX seat, which must not be removed to enhance anti-static surge capability.

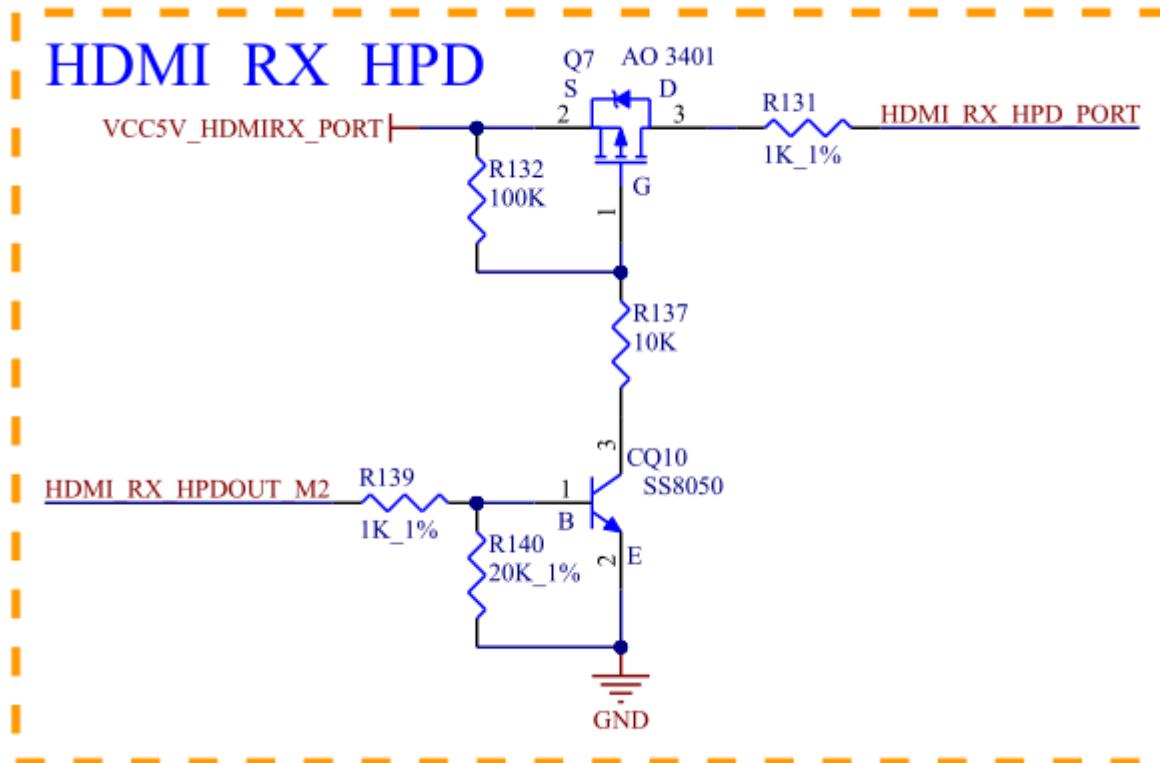


HDMI_RX_HPDOUT is a function where the HDMI RX controller is multiplexed to a general GPIO. Its level depends on the voltage of the power domain it's in. Check the Pin Mux table for specific multiplexed pins and levels.

As the HDMI RX controller does not support hardware detection Source plug-in, detection can only be done on software, the hardware circuit is as follows:

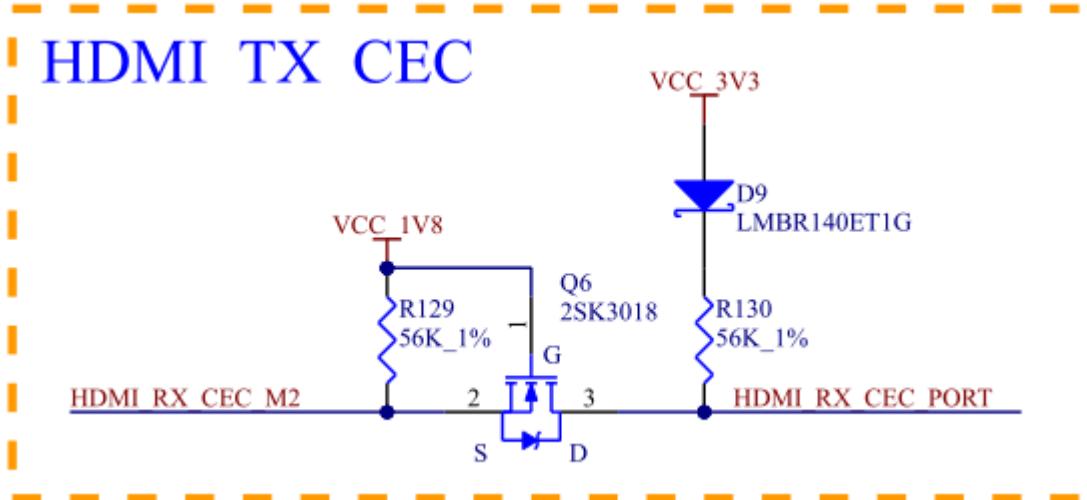


After detecting the HDMI_RX_DET_L pull-down action, HDMI_RX_HPDOUT_M2 outputs a high level, Q10 conducts, and VCC5V_HDMIRX_PORT sends 5V to HDMI_RX_HPD_PORT to complete the handshaking action with the Source side.



HDMI_RX_CEC is the HDMI controller CEC function multiplexed to the general GPIO, the specific pins, please check the Pin Mux table; level with the power domain voltage where the power domain supply voltage has changed, the peripheral circuit pull-up resistor power must also be adjusted synchronously.

The selected CEC signal for HDMI_RX_CEC_M2; the pin level is 1.8V, so the external need to add a level conversion circuit, Q6 default selection 2SK3018, if you want to change other models, the junction capacitance must be equivalent; if the junction capacitance is too large, not only will affect the work, the certification will not pass. Reference schematic diagram as follows:

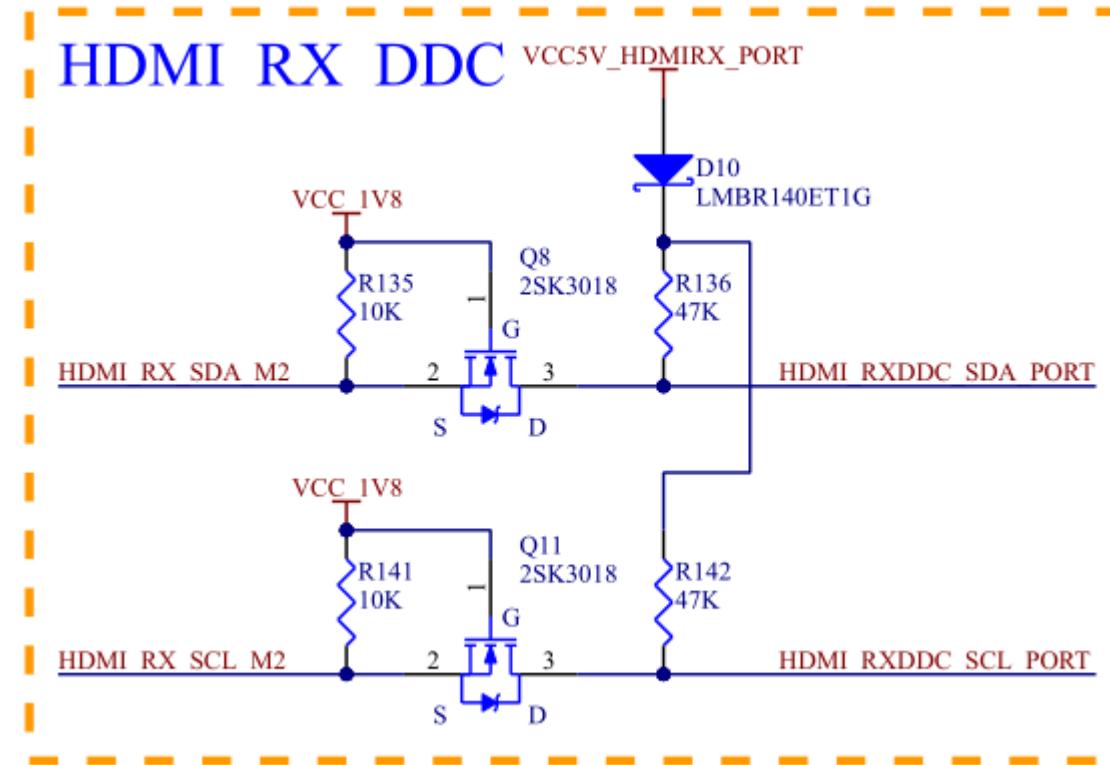


HDMI_RX_DDC_SCL/DDC_SDA is the HDMI RX controller I2C/DDC bus, which is led out from general GPIO, please refer to the Pin Mux table for the specific pins that can be multiplexed on IO; the level changes with the voltage of the power supply domain where it is located, and the pull-up resistor power supply of the peripheral circuit must also be

adjusted simultaneously.

Although the DDC_SCL/DDC_SDA protocol specifies a 5V level, the RK3588 IO does not support a 5V level, so the level conversion circuit need to be added and can not be deleted. The default is to use MOS tube level conversion, and the MOS type is 2SK3018; If the model needs to be changed, the junction capacitance must be equivalent, because the junction capacitance is too large, not only affecting the work and also affect the certification leading to failing certification.

It is recommended to refer to the default value for the pull-up resistance and can not be modified arbitrarily.



Design Considerations:

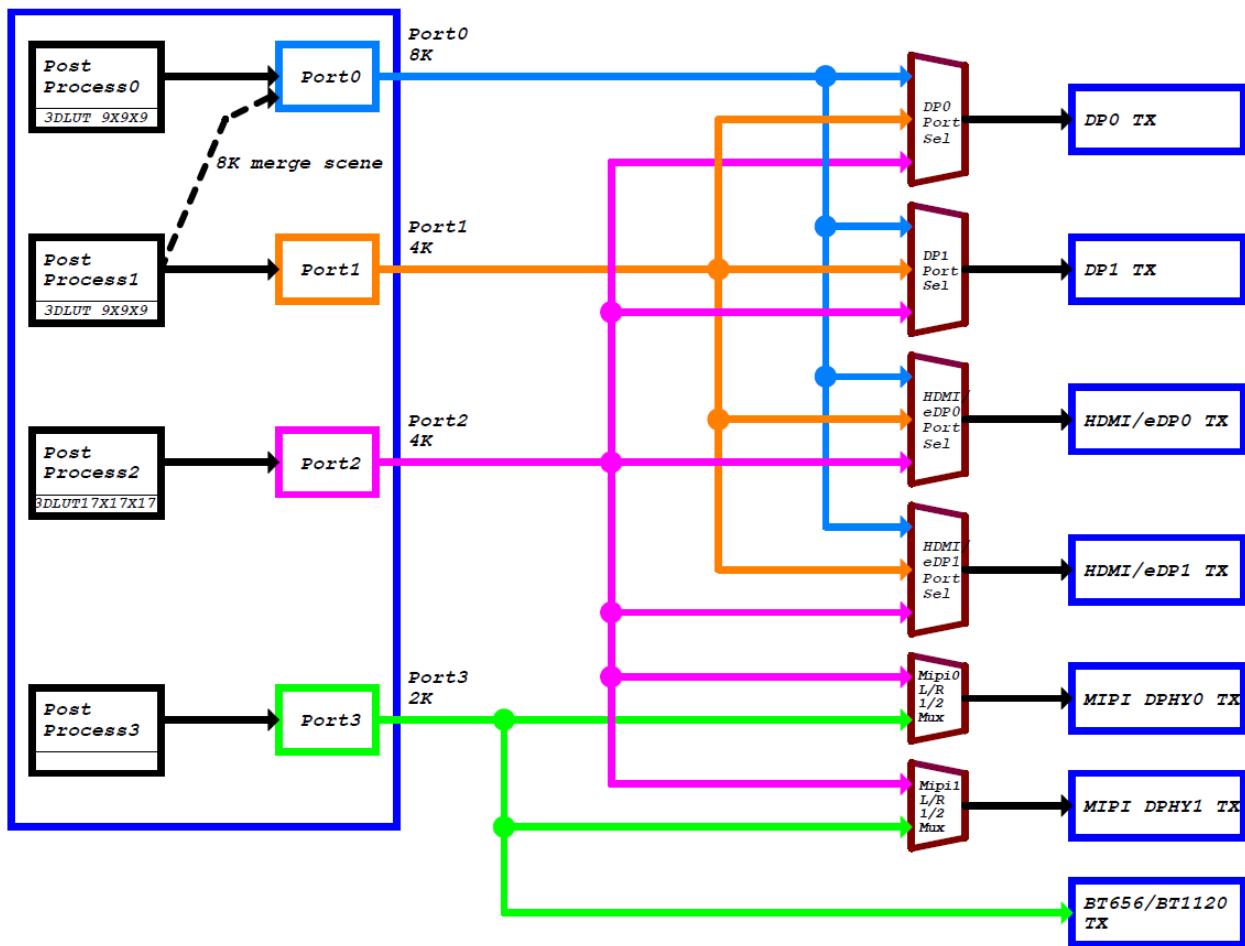
1. It is recommended to place a 0.1 uF decoupling capacitor on the Pin18 pin of the HDMI socket, and place it close to the HDMI socket pin when routing out. To strengthen the anti-static capability, ESD devices must be reserved on the signal. ESD parasitic capacitance of HDMI 2.0 signal should not exceed 0.4pF. It is recommended to use no more than 1pF for other signals of ESD parasitic capacitance.
2. The differential impedance of the traces should be controlled within 100 ohms ± 10%;
3. The maximum time - delay difference within differential pair should be < 3mil;
4. The clock and data equal length should be < 200 mil;
5. The spacing between differential pairs should be at least 5 times the HDMI trace width;
4. The spacing between HDMI signals and other signals should be at least 4 times the HDMI trace width.

3.5.19 Video Output Interface

The VOP controller of RK3588 chip, with four Port outputs, supports DP0/DP1/HDMI0/eDP0/HDMI1/eDP1/MIPI DS10/MIPI DS11/BT656/BT1120 video interface outputs.

Up to 4 screens are allowed to be displayed differently, such as 4K+4K+4K+2K, and if supports 8K, then only supports 8K+4K+2K (where 8K is achieved by Post Process0+ Post Process1 merging).

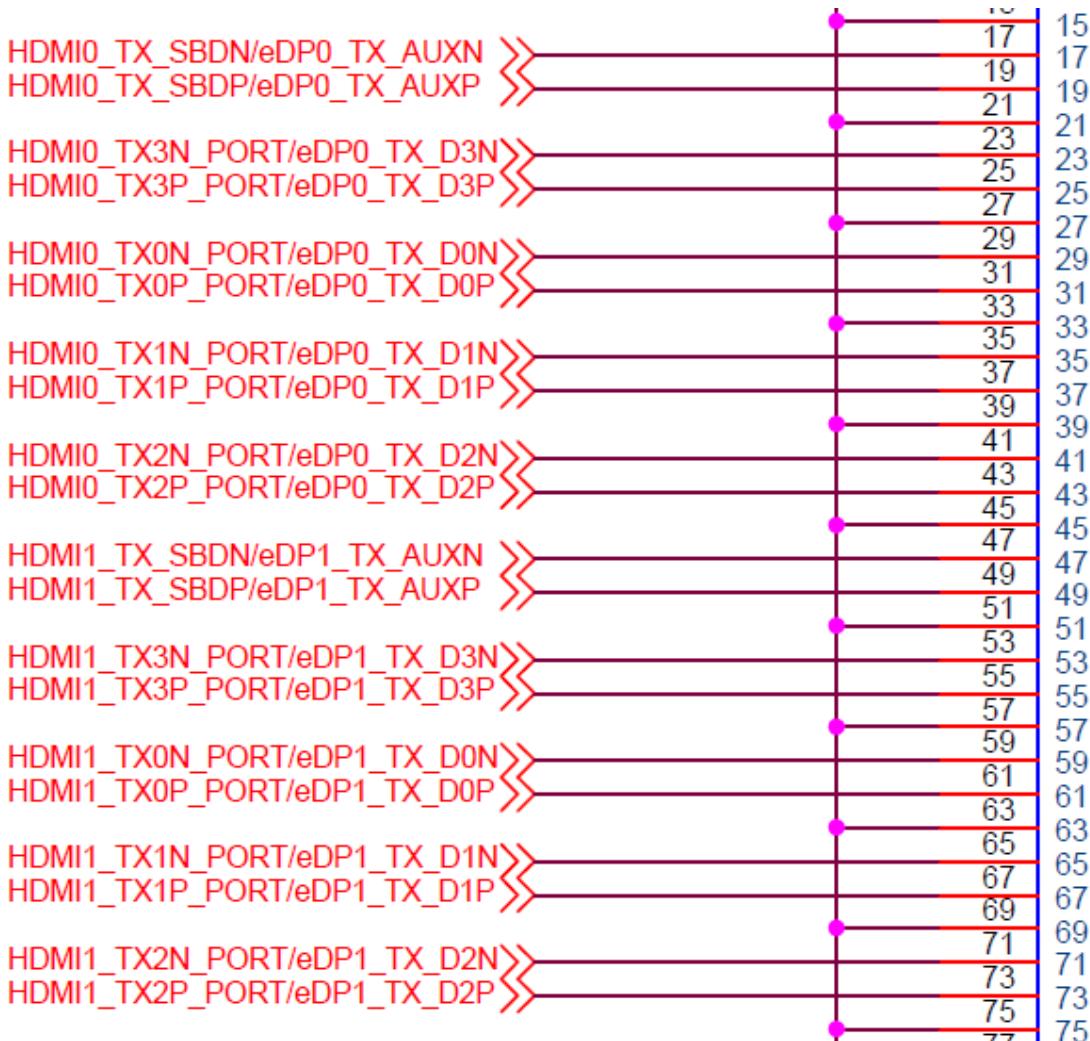
VOP and video interface output path diagrams are as follows:



3.5.19.1 HDMI2.1/eDP TX Interface

RK3588 has 2 x built-in HDMI/eDP TX Combo PHY.

- HDMI/eDP TX Combo PHY supports the following two modes:
- HDMI TX mode: maximum resolution support 8K@60Hz, support RGB/YUV444/YUV420 (Up to 10bit) format;
- eDP TX mode: Maximum resolution support 4K@60Hz, support RGB/YUV422(Up to 10bit) format.



· HDMI2.1 TX Mode

RK3588 supports HDMI 2.1 and downward for HDMI 2.0, compatible with HDMI 1.4. Because HDMI 2.1 works in FRL mode and works in TMDS mode, when switching to HDMI 2.0 and below, it will work in TMDS mode, so the AC coupled voltage mode driver is used.

As shown in the figure below, the AC coupling capacitor capacitance is 220nF, which cannot be changed at will; because the lower ESR and ESL can also reduce the impedance change on the line, it is recommended to use the 0201 packaging for the AC coupling capacitor.

Taking HDMI TX0 as an example, HDMI TX1 and HDMI TX0 are consistent.

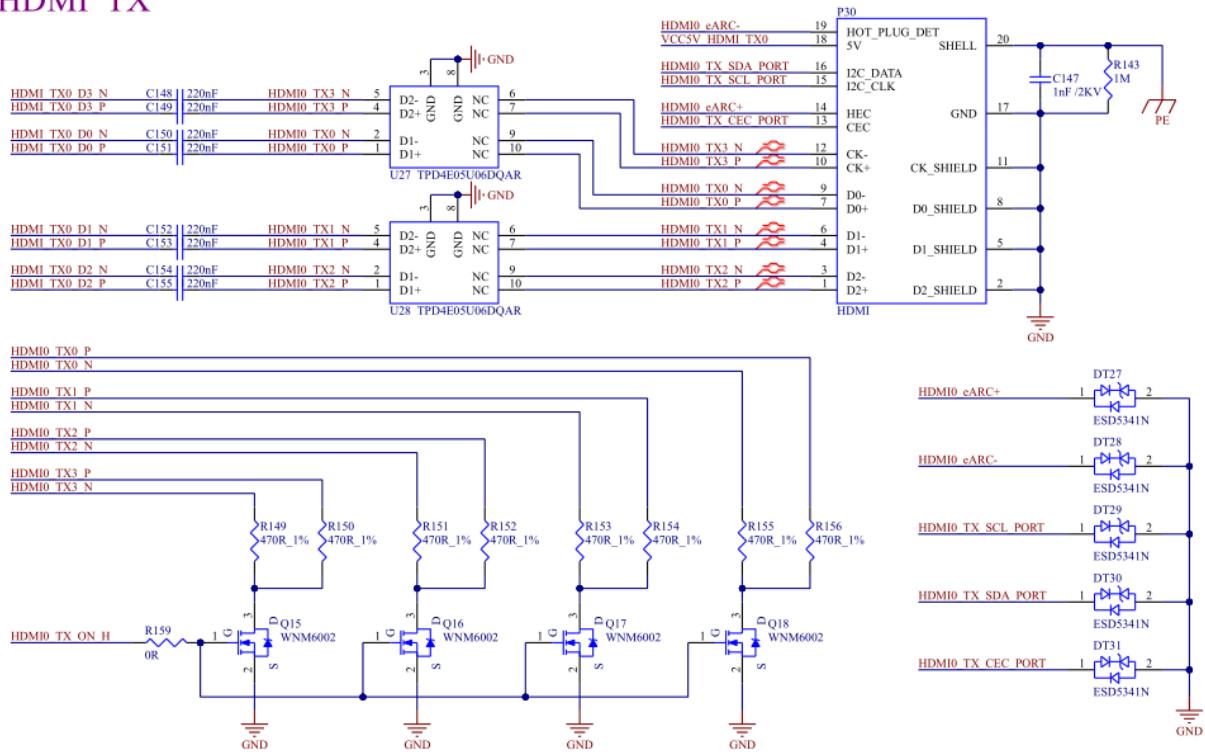
Operating in HDMI 2.1 mode, HDMI0_TX_ON_H is configured low, Q15, Q16, Q17, Q18 are not on.

When operating in HDMI 2.0 and below mode, HDMI0_TX_ON_H is configured high, Q15, Q16, Q17, and Q18 are on, and a DC bias of approximately 3V is formed by a 499ohm resistor to ground and a 50ohm pull-up resistor on the Sink side.

Design Considerations:

When only HDMI 2.0 and lower modes need to be supported, components Q15, Q16, Q17, and Q18 must not be omitted. It is essential to ensure that the transistors remain non-conductive when the device is powered off, as the HDMI CTS Test ID 7-3 TMDS Voff test requires that the Voff voltage stays within $\pm 10\text{mV}$ of AVcc when the Device Under Test (DUT) is unpowered; otherwise, this test item will fail.

HDMI TX



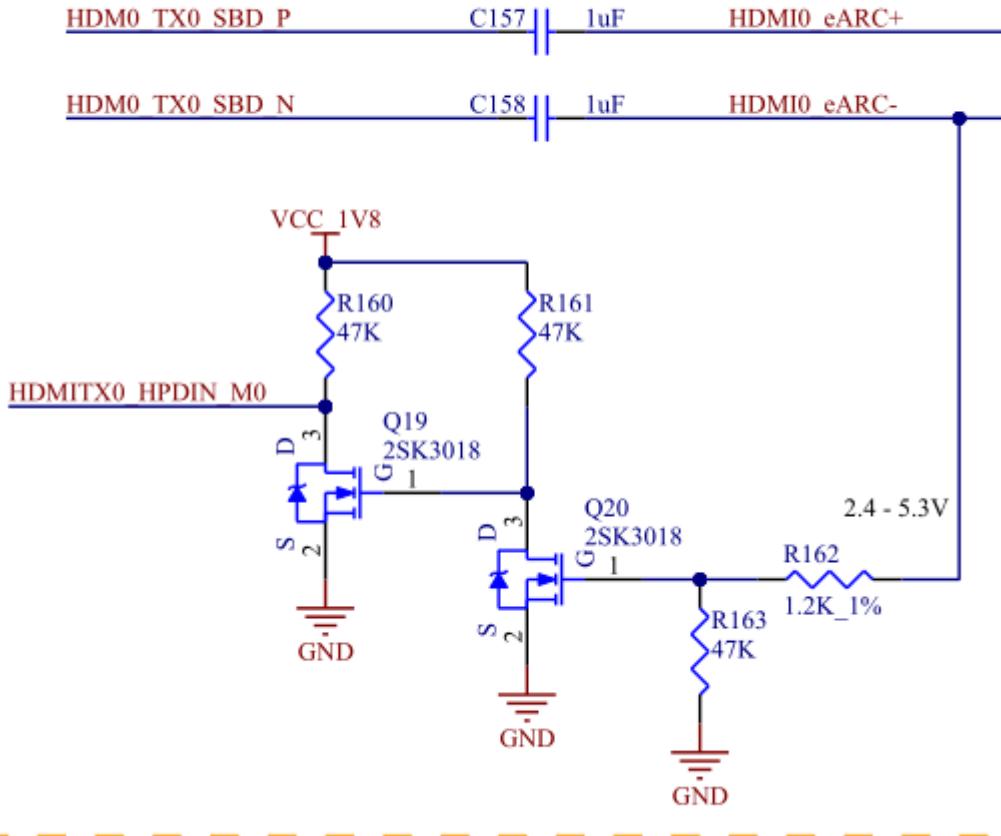
FRL mode: In the traditional TMDS architecture, a separate channel is used to transmit the Clock. But in the FRL architecture, the Clock is embedded in the Data channel, and the Clock is resolved at the Sink side through the Clock Recovery.

FRL rate vs. channel relationship:

Channel Rate	Channel Quantity
3Gbps	3
6Gbps	3
6Gbps	4
8Gbps	4
10Gbps	4
12Gbps	4

It supports ARC/eARC via HDMI0_TX_SBDP/HDMI0_TX_SBDN signal to parse out audio data inside RK3588.

HDMI TX eARC



HDMI_TX0_HPD is the HDMI TX controller multiplexed to the general GPIO; the level changes with the voltage of the power supply domain where it is located, and the pull-up resistor power supply of the peripheral circuit must also be adjusted simultaneously.

HDMI_TX0_CEC is the multiplexing of the HDMI controller's CEC function to a general GPIO. Its level varies with the power domain voltage. If the power domain supply voltage changes, the power supply of the pull-up resistor in the peripheral circuit must be adjusted synchronously.

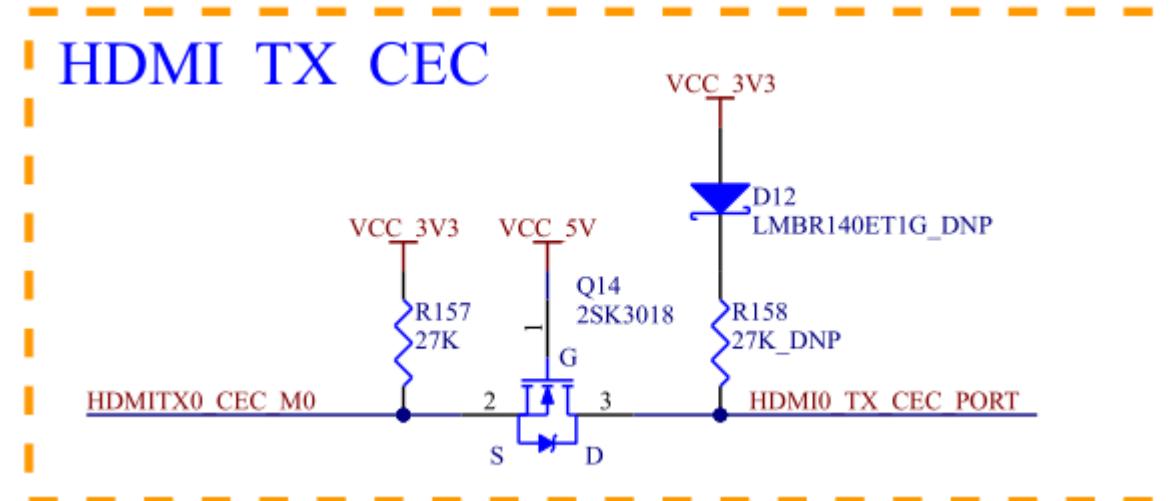
The CEC protocol specifies a 3.3V level. However, the protocol requires that the leakage should not exceed 1.8 μ A when adding 3.3V to the CEC pin through a 27K resistor.

Test ID 7-15: CEC Line Degradation

Reference	Requirement
[HDMI: Table 4-40] CEC line Electrical Specifications for all Configurations	A device with power removed (from the CEC circuitry) shall not degrade communication between other CEC devices (e.g. the line shall not be pulled down by the powered off device). Maximum CEC line leakage current must be $\leq 1.8\mu$ A

RK3588 IO Domain Leakage will occur if there is a voltage at IO in the power-down state. For example, the RK3588

is a power failure, and its HDMI cable is in connection to the Sink side (TV or monitor); meanwhile, the CEC at the Sink side has power and leaks through the HDMI cable to the RK3588 IO, which will cause the CEC to leak more than 1.8uA, so an external isolation circuit is necessary. We can not modify the R158 resistance at will, and we need to use 27Kohm, Q14 default, and selection 2SK3018. If needing to change other models, the junction capacitor must be the equivalent, if not, it will not only affect the work but will also affect the certification through.



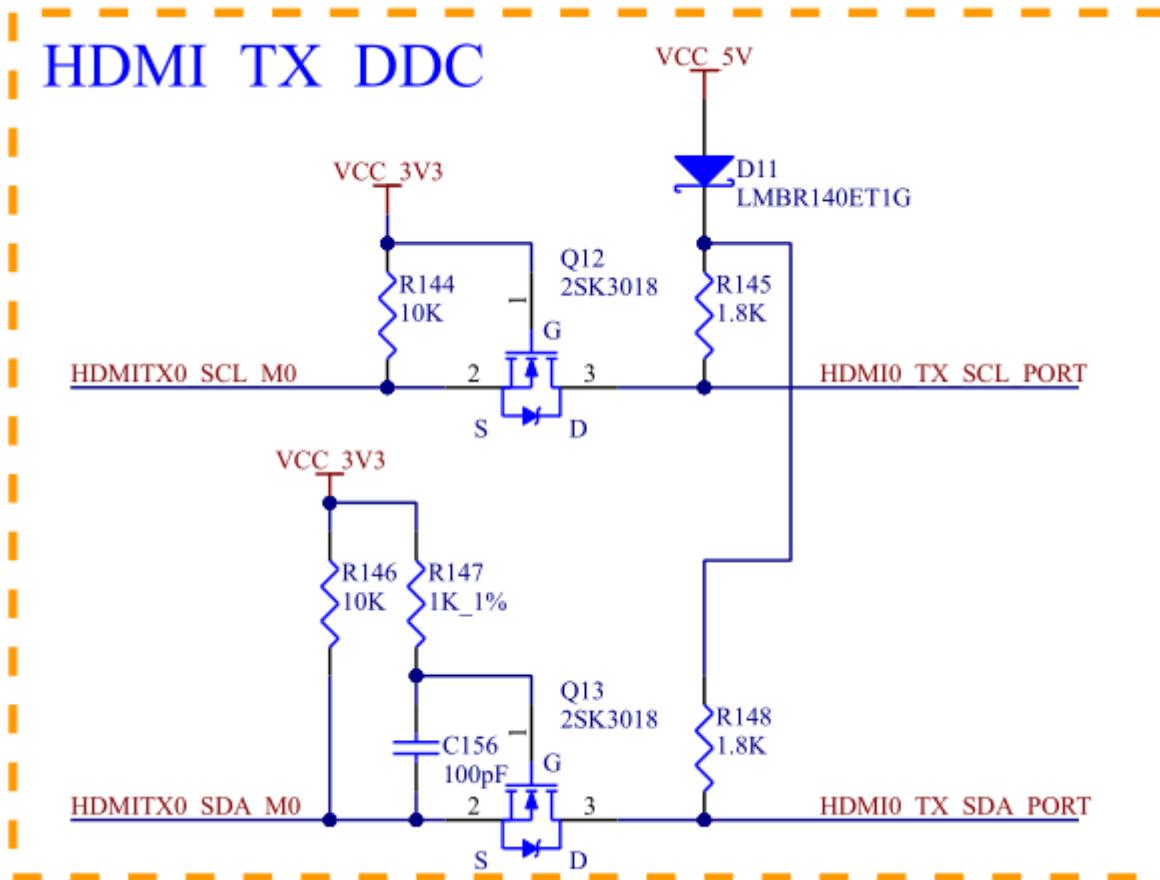
HDMI_TX0/1 DDC_SCL/DDC_SDA is the HDMI TX0/1 controller I2C/DDC bus, which is led out from general GPIO; the level changes with the voltage of the power supply domain where it is located, and the pull-up resistor power supply of the peripheral circuit must also be adjusted simultaneously.

Although the DDC_SCL/DDC_SDA protocol specifies a 5V level, the RK3588 IO does not support a 5V level, so the level conversion circuit need to be added and can not be deleted. The default is to use MOS tube level conversion, and the MOS type is 2SK3018; If the model needs to be changed, the junction capacitance must be equivalent, because the junction capacitance is too large, not only affecting the work and also affect the certification leading to failing certification.

It is recommended to refer to the default value for the pull-up resistance and not to modify it arbitrarily.

The D11 diode cannot be removed and is used to prevent leakage from the Sink side to VCC_5V0.

1K in series between MOS gate for SDA signal level conversion and power supply; A 100pF is connected in parallel between the MOS gate and source to improve the timing and can not be removed.



HDMI holder Pin18 voltage needs to be kept between 4.8-5.3V, 1uF decoupling capacitor needs to be placed on the pin, which must not be deleted, and the layout is placed close to the HDMI holder pin.

To strengthen the anti-static capability, ESD devices must be reserved on the signal. ESD parasitic capacitance of HDMI2.1 signal must not exceed 0.2pF.

ESD parasitic capacitance for other signals is recommended to use no more than 1pF.

Design Considerations:

1. The Coss of the controlling MOSFET should not be excessively high, as it may degrade signal quality. It is recommended to select a MOSFET model as per the reference design or with a corresponding Coss value;
2. Wire impedance control differential $100\text{ohm} \pm 10\%$;
3. Maximum Skew Within Differential Pair: < 3mil;
4. Equal Length Requirement Between Differential Pairs < 200mil;
5. Spacing Between Differential Pairs: no less than 7 times the width of the HDMI trace;
6. Spacing Between HDMI and other signals: ≥ 7 times the HDMI trace;
7. It is recommended to avoid vias;
8. I/O-to-ground capacitance: ≤ 0.2 pF.

-eDP TX Mode

Support eDP V1.3 version, total 4Lane, eDP TX maximum output resolution up to 4K@60Hz

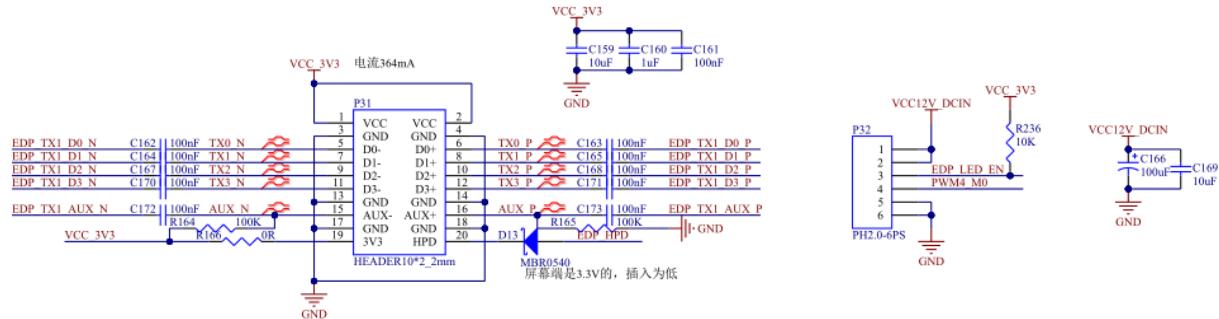
- Support 1.62/2.7/5.4Gbps per Lane rate;

- Support 1Lane or 2Lane or 4Lane mode;
- Support AUX channel with rate up to 1Mbps.

Take eDP TX0 as an example, eDP TX1 is the same as eDP TX0.

eDP_TX0_D0P/D0N, eDP_TX0_D1P/D1N, eDP_TX0_D2P/D2N, eDP_TX0_D3P/D3N need 100nF AC coupling capacitor in series; because the lower ESR and ESL can also reduce the impedance change on the line, it is recommended to use the 0201 packaging for the AC coupling capacitor, and the layout is placed close to the FET3588-C pin.

eDP



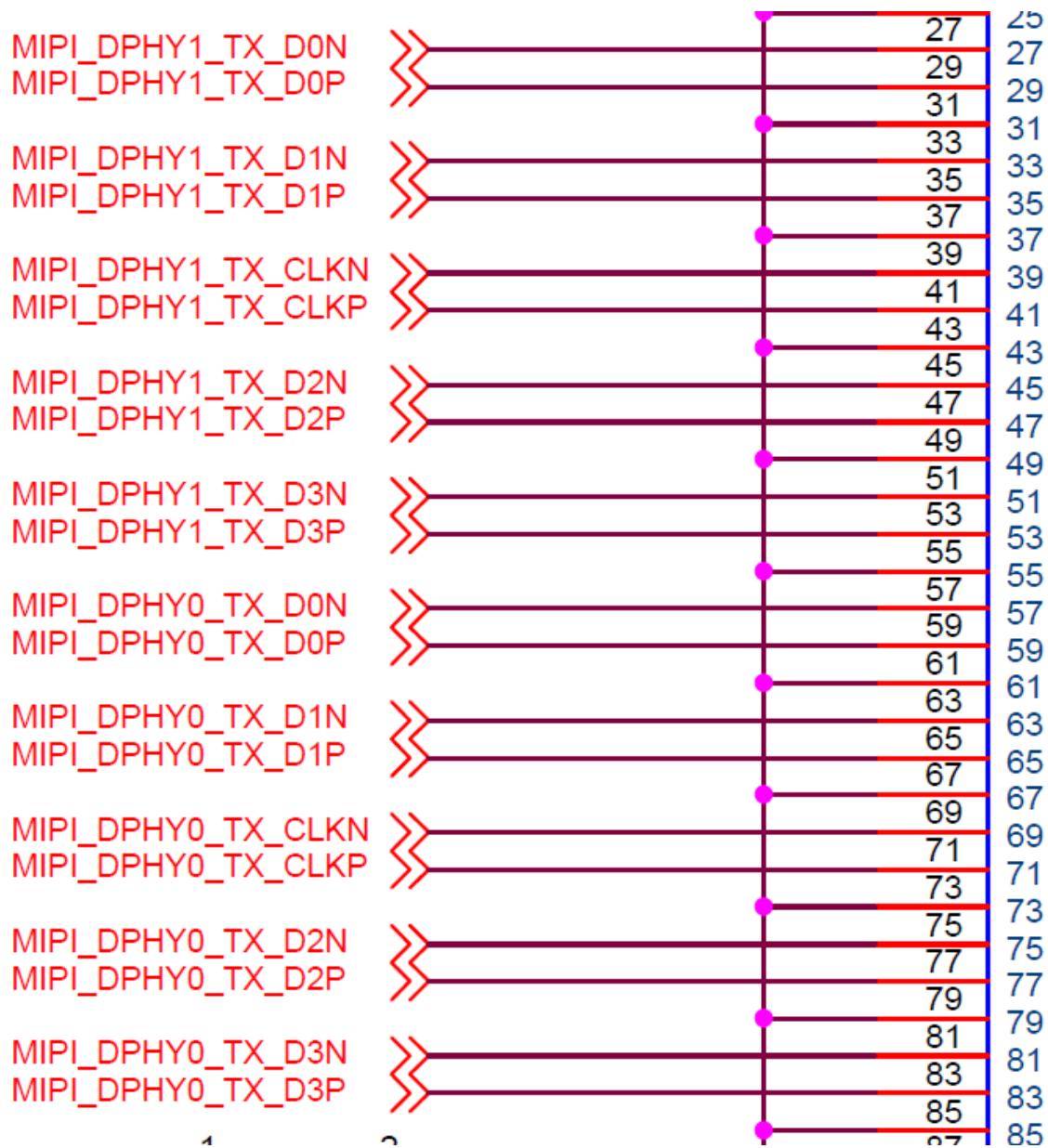
Design Considerations:

1. The differential impedance of the traces should be controlled within $85 \text{ ohms} \pm 10\%$;
2. The maximum delay difference within each differential pair should be $< 3\text{mil}$;
3. The spacing between differential pairs should be at least 4 times the EDP trace width;
4. The spacing between EDP signals and other signals should be at least 4 times the EDP trace width;
5. It is recommended that the number of vias allowed for each signal does not exceed 2.

3.5.19.2 MIPI_D/CPHY_TX

RK3588 has two MIPI D-PHY/C-PHY Combo PHY TX:

- D-PHY supports V1.2, D-PHY mode with 0/1/2/3 Lane, data transfer rate up to 2.5Gbps;
- C-PHY supports V1.1 version, C-PHY mode has 0/1/2 Trio, 3 lines per Trio A/B/C, data transfer rate up to 5.7Gbps/Trio (2.5Gps).



DPHY and CPHY configuration support:

- MIPI D-PHY/C-PHY Combo PHY0 TX and RX can only be configured as DPHY0 TX, DPHY0 RX mode or CPHY0 TX, CPHY0 RX mode at the same time, and it does not support one configured as DPHY0 TX and another configured as CPHY0 RX;
- The TX and RX of MIPI D-PHY/C-PHY Combo PHY1 can only support being configured simultaneously in the DPHY1 TX and DPHY1 RX modes, or simultaneously in the CPHY1 TX and CPHY1 RX modes. It does not support one being configured as DPHY1 TX and the other as CPHY1 RX.

MIPI D/C-PHY0 support (In D-PHY time mode):

- Support x4Lane mode, MIPI_DPHY0_TX_D[3: 0] data reference MIPI_DPHY0_TX_CLK;

MIPI D/C-PHY0 support (In C-PHY time mode):

- Support 0/1/2 Trio, 3 wires per Trio A/B/C, MIPI_CPHY0_TX_TRIO[2:0]_A, MIPI_CPHY0_TX_TRIO[2:0]_B, MIPI_CPHY0_TX_TRIO[2:0]_C.

MIPI D/C-PHY1 support (In D-PHY time mode):

- Support x4Lane mode, MIPI_DPHY1_TX_D[3: 0] data reference MIPI_DPHY1_TX_CLK;

MIPI D/C-PHY1 support (In C-PHY time mode):

- Support 0/1/2 Trio, 3 wires per Trio A/B/C, MIPI_CPHY1_TX_TRIO[2:0]_A, MIPI_CPHY1_TX_TRIO[2:0]_B, MIPI_CPHY1_TX_TRIO[2:0]_C.

Design Considerations:

1. Wire impedance control differential $100\text{ohm} \pm 10\%$;
2. Maximum Skew Within Differential Pair: $< 3\text{mil}$;
3. Equal length between clock and data $< 6\text{mil}$;
4. Recommended equal length between differential pairs: ≥ 4 times the MIPI trace width, and ≥ 3 times the MIPI trace width;
5. Spacing between MIPI signals and other signals: Spacing ≥ 4 times MIPI trace width and $\geq 3 \times$ MIPI trace width;
6. For CPHY, the single-ended trace impedance should be controlled at $50 \Omega \pm 10\%$;
7. The maximum time delay difference within a group (TRIO_A\TRIO_B\TRIO_C) should be < 3 mil;
8. The length matching requirement between groups (TRIO0\TRIO1\TRIO2) should be < 50 mil;
9. It is recommended that the number of vias allowed for each signal should ≤ 2 ;
10. It is recommended that the spacing between differential pairs should $\geq 4 \times$ MIPI trace width;
11. It is recommended that the spacing between MIPI and other signals should $\geq 4 \times$ MIPI trace width.

3.5.19.3 DP TX

RK3588 supports two DP1.4 TX PHY (and USB3.0 Combo), output resolution up to 8K@30Hz

- Each Lane rate can support 1.62/2.7G/5.4/8.1Gbps;
- Supports 1Lane or 2Lane or 4Lane mode;
- Supports RGB/YUV (Up to 10bit) format;
- Supports Single Stream Transport (SST).

Please refer to section 3.5.15 for the USB pin multiplexing.

Design Considerations:

1. The 100nF AC coupling capacitors need to be connected in series with DP0_TX_D0P/D0N, DP0_TX_D1P/D1N, DP0_TX_D2P/D2N, DP0_TX_D3P/D3N, DP1_TX_D0P/D0N, DP1_TX_D1P/D1N, DP1_TX_D2P/D2N, and DP1_TX_D3P/D3N. It is recommended to use the 0201 package for the AC coupling capacitors, as they have lower ESR and ESL and can also reduce the impedance variation on the line. During layout, place them close to the FET3588 - C pins;

DP1_TX_D0P/D0N、DP1_TX_D1P/D1N、DP1_TX_D2P/D2N、DP1_TX_D3P/D3N 需要串接的 100nF 交流耦合电容，交流耦合电容建议使用 0201 封装，更低的 ESR 和 ESL，也可减少线路上的阻抗变化，布局时，靠近 FET3588—C 管脚放置；

2. The trace impedance should be controlled at a differential $100 \text{ ohm} \pm 10\%$ (when used only as a DP interface without multiplexing), and a differential $95 \text{ ohm} \pm 10\%$ (when USB3.0/DP1.4 is multiplexed);
3. The delay difference within the differential pair should be $< 3\text{mil}$;
4. Equal Length Requirement Between Differential Pairs $< 500\text{mil}$;

5. Spacing Between Differential Pairs: no less than 6 times the width of the DP trace;
6. It is recommended that the spacing between DP and other signals should $\geq 6 \times$ DP trace width.
7. It is recommended that the number of vias allowed for each signal should ≤ 2 ;
8. I/O-to-ground capacitance: ≤ 0.2 pF.

3.5.19.4 BT1120 TX

The RK3588 supports 16bit BT1120 output interface with maximum output resolution up to 1920X1080@60Hz; it is compatible with 8bit BT656 interface, supporting PAL and NTSC.

The OK3588 multiplexes this group of signals into other functions such as GPIO. If we want match it to BT1120 TX interface, please refer to the Pin Mux table for the multiplexing relationship.

The BT1120 and BT656 multiplexing relationships are as follows:

Pin Name	BT656 (8bit)	BT1120 (16bit)
BT1120_CLKOUT	CLKOUT	CLKOUT
BT1120_D15	—	D15
BT1120_D14	—	D14
BT1120_D13	—	D13
BT1120_D12	—	D12
BT1120_D11	—	D11
BT1120_D10	—	D10
BT1120_D9	—	D9
BT1120_D8	—	D8
BT1120_D7	D7	D7
BT1120_D6	D6	D6
BT1120_D5	D5	D5
BT1120_D4	D4	D4
BT1120_D3	D3	D3
BT1120_D2	D2	D2
BT1120_D1	D1	D1
BT1120_D0	D0	D0

Data correspondence relationship in BT1120 output, supporting YC Swap.

Pin Name	Default mode		Swap open:
	Pixel #0	Pixel #1	
BT1120_D0	Y0[0]	Y1[0]	Cb0[0]
BT1120_D1	Y0[1]	Y1[1]	Cb0[1]
BT1120_D2	Y0[2]	Y1[2]	Cb0[2]
BT1120_D3	Y0[3]	Y1[3]	Cb0[3]
BT1120_D4	Y0[4]	Y1[4]	Cb0[4]
BT1120_D5	Y0[5]	Y1[5]	Cb0[5]
BT1120_D6	Y0[6]	Y1[6]	Cb0[6]
BT1120_D7	Y0[7]	Y1[7]	Cb0[7]
BT1120_D8	Cb0[0]	Cr0[0]	Y0[0]
BT1120_D9	Cb0[1]	Cr0[1]	Y0[1]
BT1120_D10	Cb0[2]	Cr0[2]	Y0[2]
BT1120_D11	Cb0[3]	Cr0[3]	Y0[3]
BT1120_D12	Cb0[4]	Cr0[4]	Y0[4]
BT1120_D13	Cb0[5]	Cr0[5]	Y0[5]
BT1120_D14	Cb0[6]	Cr0[6]	Y0[6]
BT1120_D15	Cb0[7]	Cr0[7]	Y0[7]

Design Considerations:

1. The default pin level of BT1120 output interface is 3.3V, which needs to be matched according to the actual IO power supply requirements of the peripheral device. If you need to change it to 1.8V, please contact Forlinx;
2. The following table is the recommended table for pull-up and pull-down and matching design of BT1120 output interface:

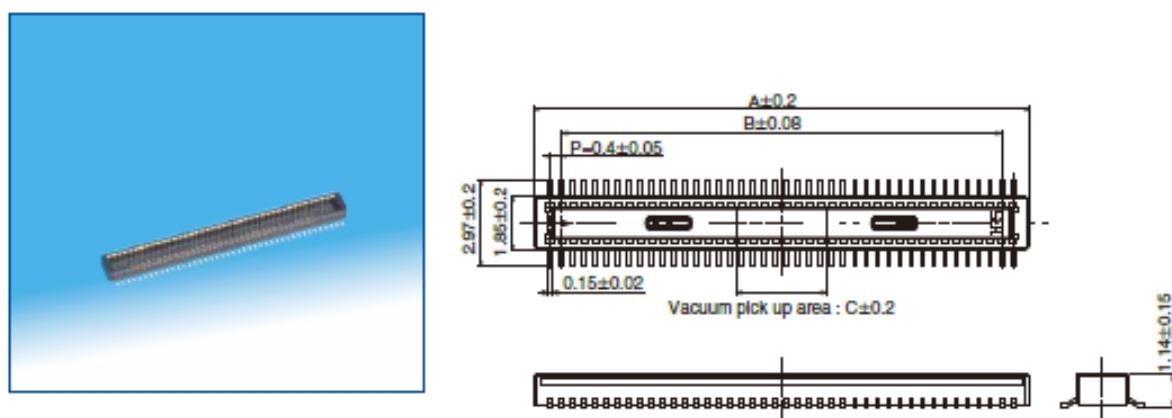
Signal	Internal Pull- up&down	Connection	Description (Chip side)
BT1120_D[1]	Pull-down	Direct connection is allowed. If feasible, reserve series resistors near the FET3588-C terminal.	BT1120 Data Output
BT1120_CLK	Pull-down	Series resistor: 22ohm, placed close to the device end.	BT1120 Clock Output

3. When implementing board-to-board connection, it is recommended to connect a 22 - 100 ohm resistor in series (the resistance value should meet SI test requirements), and reserve space for TVS devices.

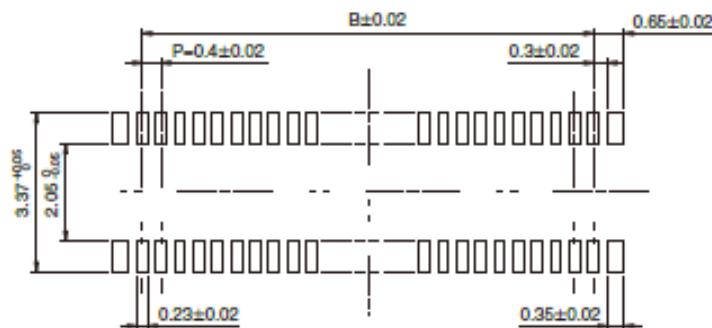
1.1.7 4. Connector Dimension Diagram

SoM Connector Dimension:

■Header



◆Recommended PCB layout



Part No.	HRS No.	No. of Contacts	A	B	C
DF40C-10DP-0.4V(51)	684-4035-0 51	10	3.52	1.6	1.0
DF40C-12DP-0.4V(51)	684-4149-9 51	12	3.92	2.0	
DF40C-20DP-0.4V(51)	684-4010-9 51	20	5.52	3.6	
DF40C-24DP-0.4V(51)	684-4011-1 51	24	6.32	4.4	
DF40C-30DP-0.4V(51)	684-4012-4 51	30	7.52	5.6	
DF40C-34DP-0.4V(51)	684-4024-3 51	34	8.32	6.4	
DF40C-40DP-0.4V(51)	684-4013-7 51	40	9.52	7.6	
DF40C-44DP-0.4V(51)	684-4077-0 51	44	10.32	8.4	
DF40C-50DP-0.4V(51)	684-4014-0 51	50	11.52	9.6	
DF40C-60DP-0.4V(51)	684-4003-3 51	60	13.52	11.6	
DF40C-70DP-0.4V(51)	684-4015-2 51	70	15.52	13.6	
DF40C-80DP-0.4V(51)	684-4001-8 51	80	17.52	15.6	
DF40C-90DP-0.4V(51)	684-4125-0 51	90	19.52	17.6	
DF40C-100DP-0.4V(51)	684-4032-1 51	100	21.52	19.6	

Note 1 : Please place orders by full reel.

Note 2 : There are no concave parts on the header side contact part to increase the mating strength for 60 pos. or above.

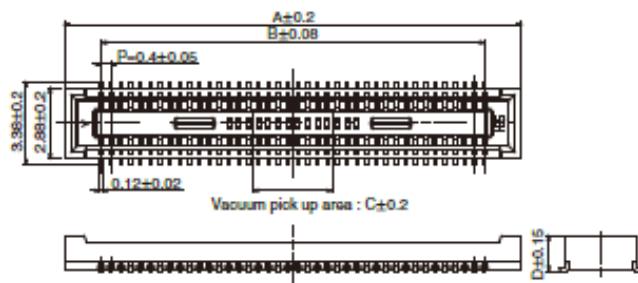
Note 3 : Use the contacts at the four corners as metal fittings. Do not use them as signal or power contacts.

Note 4 : The surface of the 90 pos and higher parts have a small, concave section that will not affect the vacuum pick up operation.

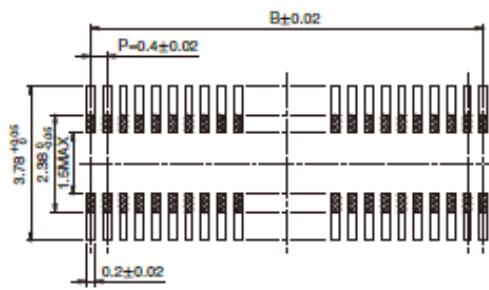
Note 5 : This connector is NOT polarized.

Carrier board Connector Dimension:

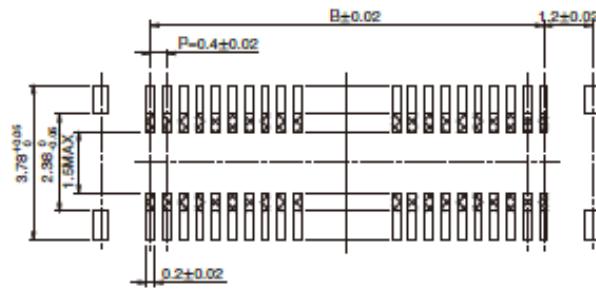
■ Receptacle (Stacking height 1.5mm)



◆ Recommended PCB layout



DF40C(Without reinforcing metal fitting)



DF40B(With reinforcing metal fitting)

■ Stacking height 1.5mm

[Specification No.]
(51) : Embossed package 5,000 pcs/reel

Part No.	HRS No.	No. of Contacts	A	B	C	D	Unit : mm
DF40B-10DS-0.4V(51)	684-4038-8 51	10	4.6	1.6			
DF40B-12DS-0.4V(51)	684-4152-3 51	12	5.0	2.0			
DF40B-30DS-0.4V(51)	684-4090-8 51	30	8.6	5.6	1.5		
DF40B-50DS-0.4V(51)	684-4018-0 51	50	12.6	9.6			
DF40B-60DS-0.4V(51)	684-4049-4 51	60	14.6	11.6			
DF40B-80DS-0.4V(51)	684-4052-9 51	80	18.6	15.6			
DF40C-20DS-0.4V(51)	684-4005-9 51	20	6.6	3.6	1.0		
DF40C-24DS-0.4V(51)	684-4006-1 51	24	7.4	4.4	1.2		
DF40C-30DS-0.4V(51)	684-4007-4 51	30	8.6	5.6	1.5		
DF40C-34DS-0.4V(51)	684-4023-0 51	34	9.4	6.4	2.3		
DF40C-40DS-0.4V(51)	684-4008-7 51	40	10.6	7.6			
DF40C-50DS-0.4V(51)	684-4009-0 51	50	12.6	9.6			
DF40C-60DS-0.4V(51)	684-4004-6 51	60	14.6	11.6			
DF40C-70DS-0.4V(51)	684-4016-5 51	70	16.6	13.6			
DF40C-80DS-0.4V(51)	684-4002-0 51	80	18.6	15.6			
DF40C-90DS-0.4V(51)	684-4124-8 51	90	20.6	17.6			
DF40C-100DS-0.4V(51)	684-4033-4 51	100	22.6	19.6			

Note 1 : Please place orders by full reel.

Note 2 : The surface of the 60 to 100 pos. parts have a small, concave section that will not affect the vacuum pick up operation.

Note 3 : Resist coating area.

Note 4 : This connector is NOT polarized.

1.1.8 5. OK3588-C Development Board Power Consumption Table

Table1. Android System Consumption

No.	Test Item	SoM Power (W)	Development board power (W)
1	No-load starting peak power	7.09	10.08
2	No-load standby peak power	1.63	4.10
3	Antutu running score	6.60	9.60
4	PWRON_L key sleep power consumption	1.32	3.56
5	PWRON_L key shutdown consumption	0.34	0.36

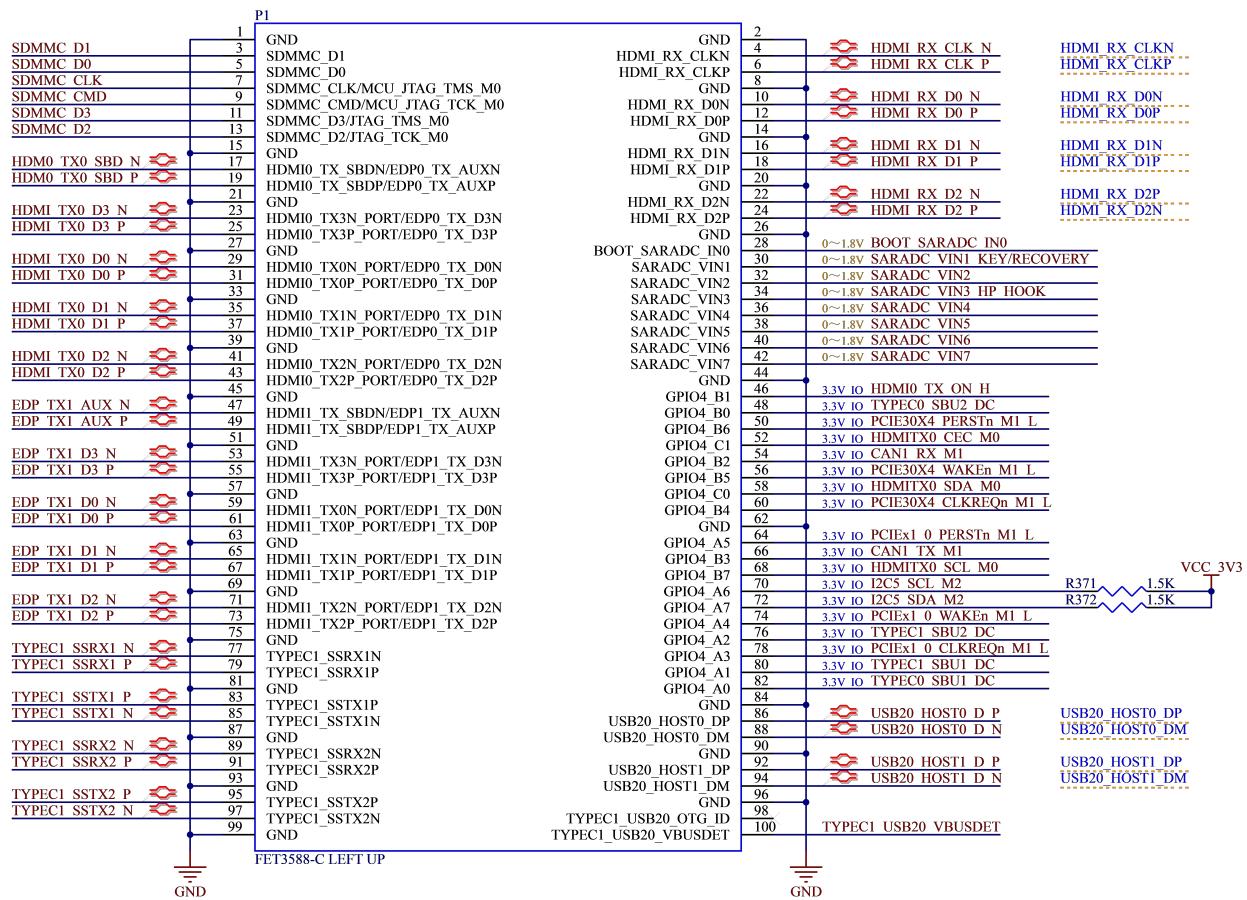
Table 2. Linux system power consumption

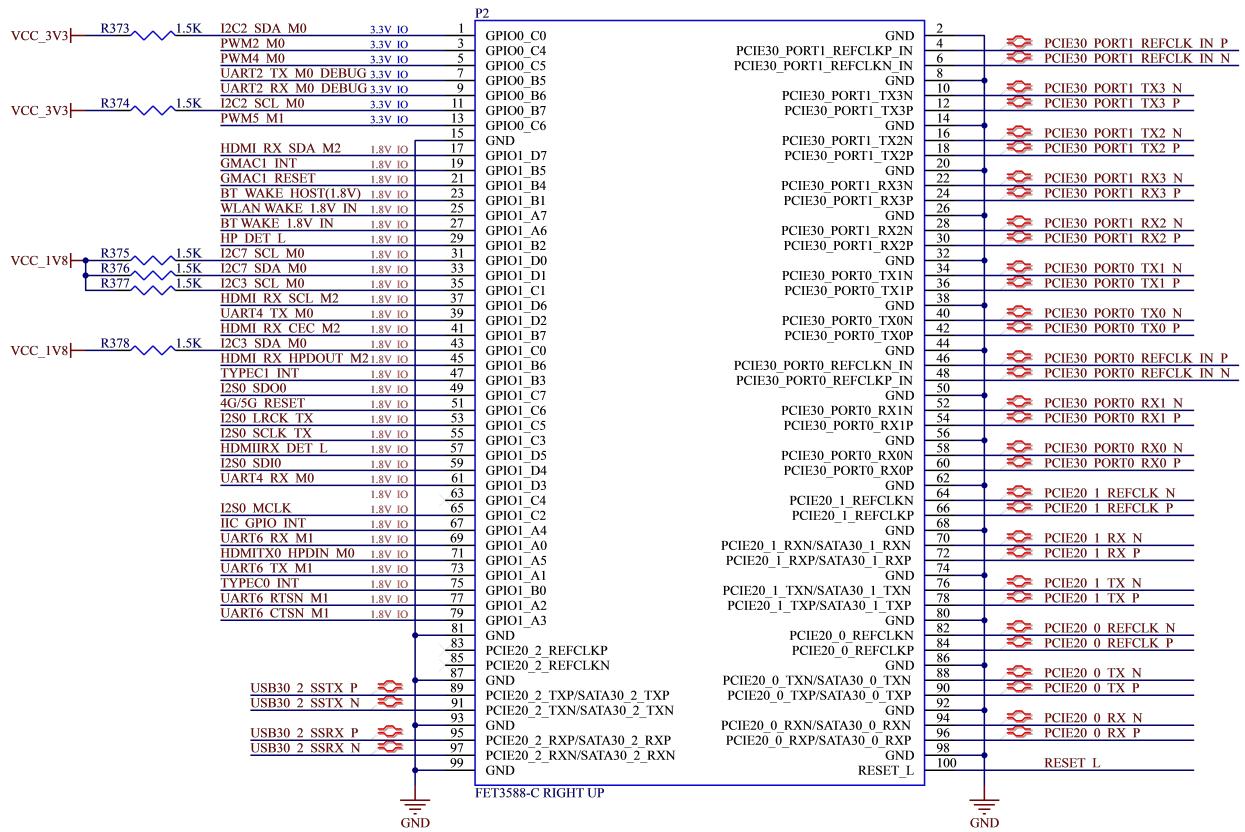
No.	Test Item	SoM Power (W)	Development board power (W)
1	No-load starting peak power	9.00	9.60
2	No-load standby peak power	2.37	3.60
3	CPU Stress + Memory + eMMC Read/Write Stress Test	6.98	10.00
4	PWRPN_L key sleep power consumption	0.59	2.53

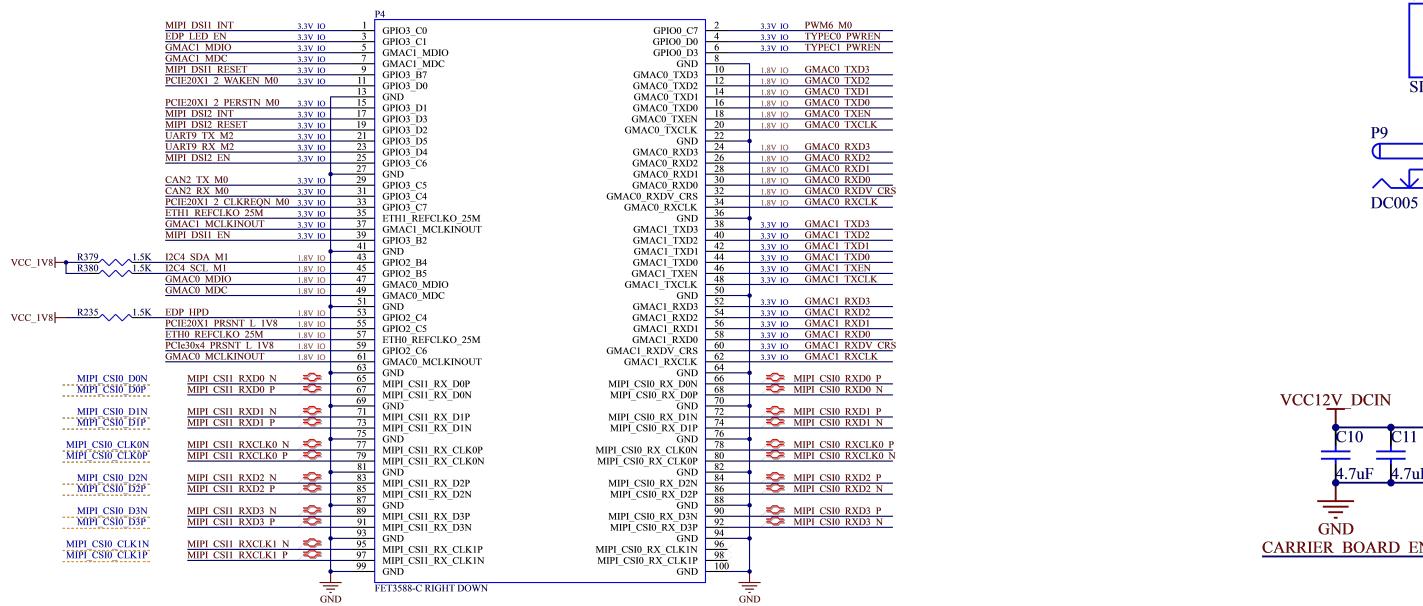
Note:

1. **Peak Current:** Maximum current value during booting;
 2. **Stable Value:** Current value stays on the boot screen after booting.
-

1.1.9 6. Minimum System Schematic







Note:

1. The minimum system includes SoM power supply, system flash circuit, and debugging serial port circuit;
 2. The factory image of OK3588 - C will load the PCIE3.0 driver during startup. At this time, it will detect the two external clock input signals PCIE30_PORT0\1_REFCLK_IN_P\N;
 3. If these two clock inputs are not available, the system will be stuck in the process and fail to start. When the PCIE3.0 clock circuit is not designed, you can simply disable the corresponding function in the device tree.

SOFTWARE

2.1 Linux5.10.209_User's Compilation Manual_V1.0

Document classification: Top secret Secret Internal information Open

2.1.1 Copyright Notice

The copyright of this manual belongs to Baoding Folinx Embedded Technology Co., Ltd. Without the written permission of our company, no organizations or individuals have the right to copy, distribute, or reproduce any part of this manual in any form, and violators will be held legally responsible.

Forlinx adheres to copyrights of all graphics and texts used in all publications in original or license-free forms.

The drivers and utilities used for the components are subject to the copyrights of the respective manufacturers. The license conditions of the respective manufacturer are to be adhered to. Related license expenses for the operating system and applications should be calculated/declared separately by the related party or its representatives.

2.1.2 Overview

This manual is designed to enable users of the Forlinx Embedded development board to quickly understand the compilation process of the products and familiarize themselves with the compilation methods of Forlinx products. The application needs to be cross-compiled on an ubuntu host before it can run on the development board. By following the methods provided in the compilation manual and performing practical operations, you will be able to successfully compile your own software code.

The manual will provide instructions for setting up the environment but there may be some unforeseen issues during the environment setup process. For beginners, it is recommended to use the pre-configured development environment provided by Forlinx. This will allow you to quickly get started and reduce development time.

Linux systems are typically installed in three ways: Dual system on a real machine, single system on a real machine, and virtual machine. Different installation methods have their advantages and disadvantages. This manual only provides methods to build ubuntu in a virtual machine.

Hardware Requirements: It is recommended to have at least 16GB memory or above. It allows for allocating a sufficient memory to the virtual machine (recommended to allocate 10GB or above), while still leaving enough resources for other operations on Windows. Insufficient memory allocation may result in slower performance on Windows.

The manual is mainly divided into four chapters:

- Chapter 1. Virtual Machine software installation - introduction to downloading and installing Vmware software;
- Chapter 2. provides the loading of the ubuntu system;
- Chapter 3. Building, setting up, and installing necessary tools for the Ubuntu system and common issues in development environments;
- Chapter 4. Compiling the kernel and Linux-related source code.

A description of some of the symbols and formats associated with this manual:

Format	Meaning
Note	Note or information that requires special attention, be sure to read carefully
?	Relevant notes on the test chapters
?	Indicates the related path.
Blue font on gray background	Refers to commands entered at the command line(Manual input required).
Black font	Serial port output message after entering a command
Bold black	Key information in the serial port output message
//	Interpretation of input instructions or output information
User- name@Hostname	forlinx @ ubuntu: Development environment ubuntu account information, which can be used to determine the environment in which the function operates.

After packaging the file system, you can use the “ls” command to view the generated files.

```
forlinx@ubuntu:~/3588$ ls
                   //List the files in this_
                   ↵directory
OK3588-linux-source  OK3588-linux-source.tar.bz2
```

- forlinx@ubuntu: the username is forlinx and the hostname is ubuntu, indicating that the operation is performed in the development environment ubuntu;
- //: Explanation of the instruction, no input required;
- ls: Blue font, indicating the relevant commands that need to be manually entered;
- **OK3588-linux-source**: Black font is the output information after entering the command; bold font is the key information; here is the packaged file system.

2.1.3 Application Scope

This manual is mainly applicable to the Linux5.10.209 operating system on the Forlinx OK3588-C platform. Other platforms can also refer to it, but there will be differences between different platforms. Please make modifications according to the actual conditions.

2.1.4 Revision History

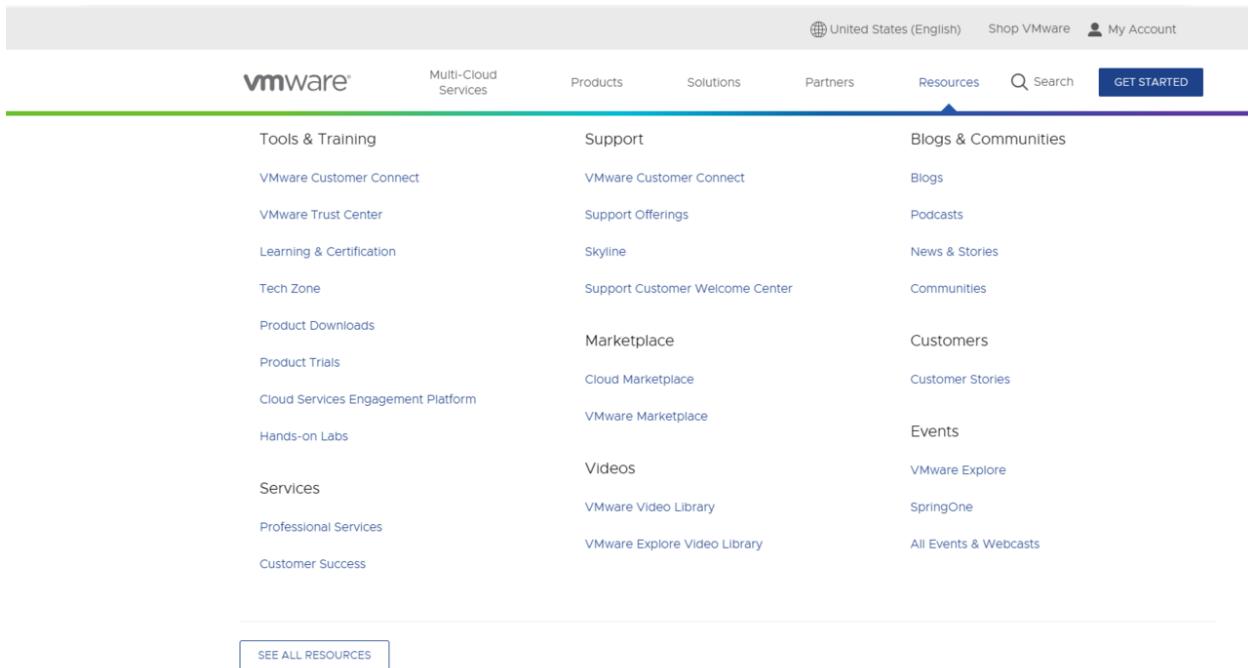
Date	Manual Version	SoM Version	Carrier Version	Board	Revision History
25/11/2024	V1.0	V1.1	V1.3 and Above		OK3588-C Linux User Compilation Manual Initial Version

2.1.5 1. VMware Virtual Machine Installation

This chapter mainly introduces the installation of VMware virtual machines, using VMware Workstation 17 Pro v17.0.0 as an example to demonstrate the installation and configuration process of the operating system.

1.1 VMware Software Downloads and Purchase

Go to the VMware website <https://www.vmware.com/cn.html> to download Workstation Pro and get the product key. VMware is a paid software that requires purchasing, or you can choose to use a trial version.



The screenshot shows the VMware website's navigation bar at the top with links for "United States (English)", "Shop VMware", and "My Account". Below the bar, the "Resources" menu is expanded, showing categories like "Tools & Training", "Support", and "Blogs & Communities". Each category has several sub-links. At the bottom left of the resources section, there is a button labeled "SEE ALL RESOURCES".

Category	Sub-links	Category	Sub-links
Tools & Training	VMware Customer Connect, VMware Trust Center, Learning & Certification, Tech Zone, Product Downloads, Product Trials, Cloud Services Engagement Platform, Hands-on Labs	Support	VMware Customer Connect, Support Offerings, Skyline, Support Customer Welcome Center, Marketplace, Cloud Marketplace, VMware Marketplace
Services	Professional Services, Customer Success	Videos	VMware Video Library, VMware Explore Video Library
			Blogs, Podcasts, News & Stories, Communities, Customer Stories, Events, VMware Explore, SpringOne, All Events & Webcasts

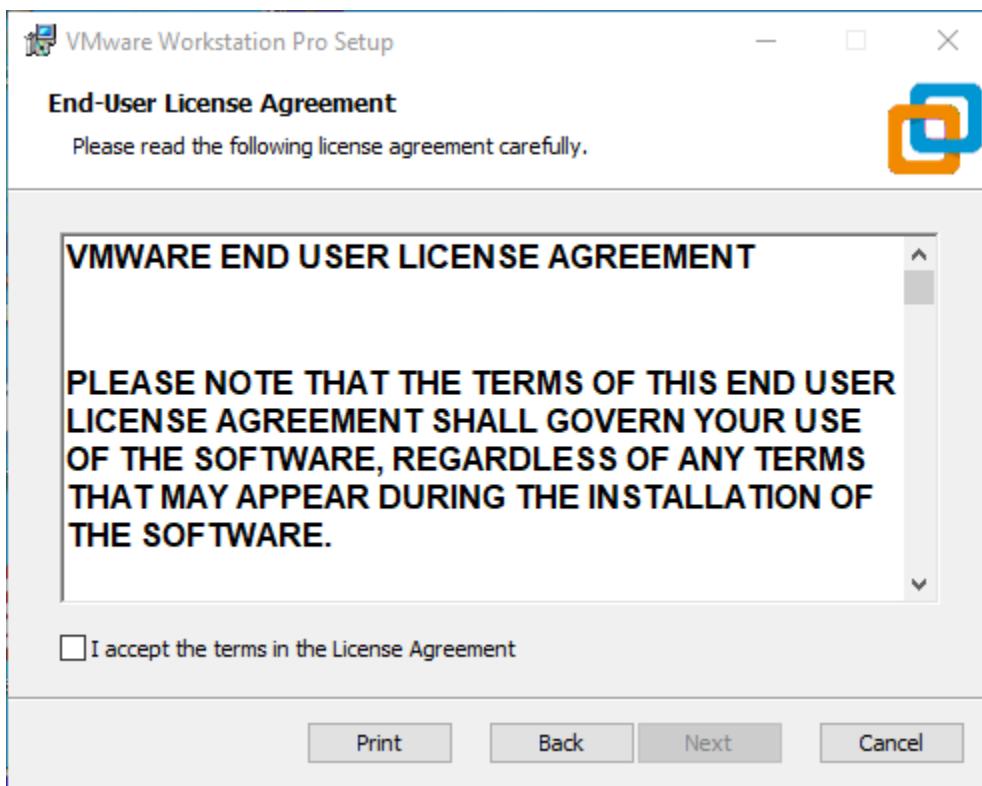
After the download is complete, double-click the startup file to start the installer.

1.2 VMware Software Installation

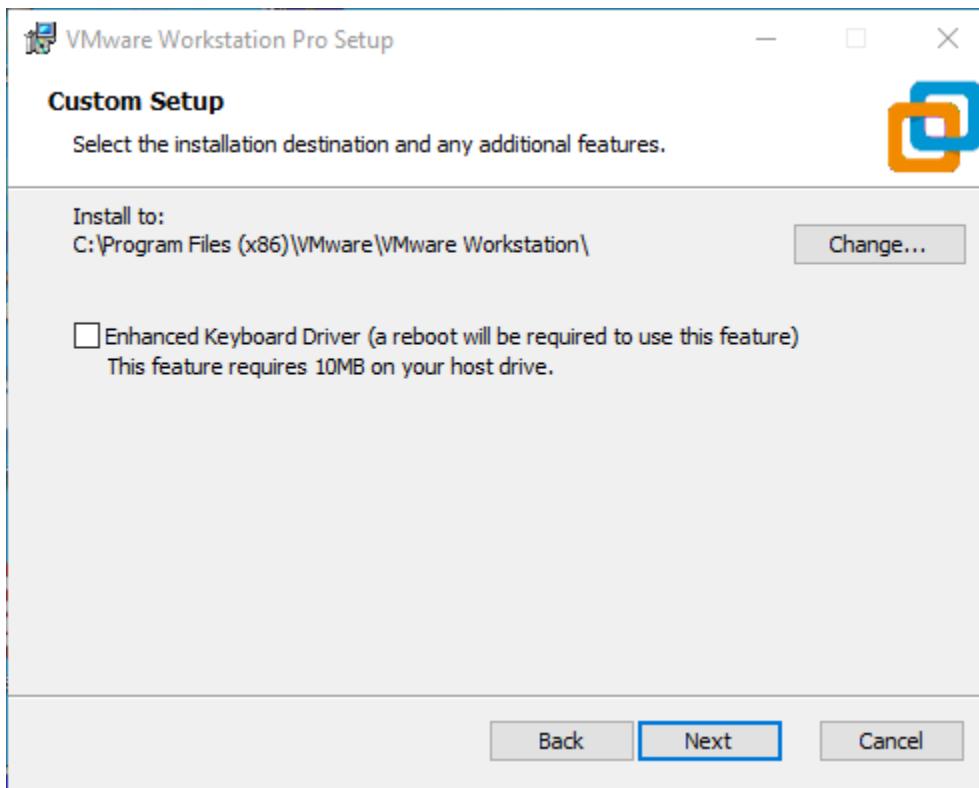
Double-click the startup program to enter the installation wizard, and click on “Next” .



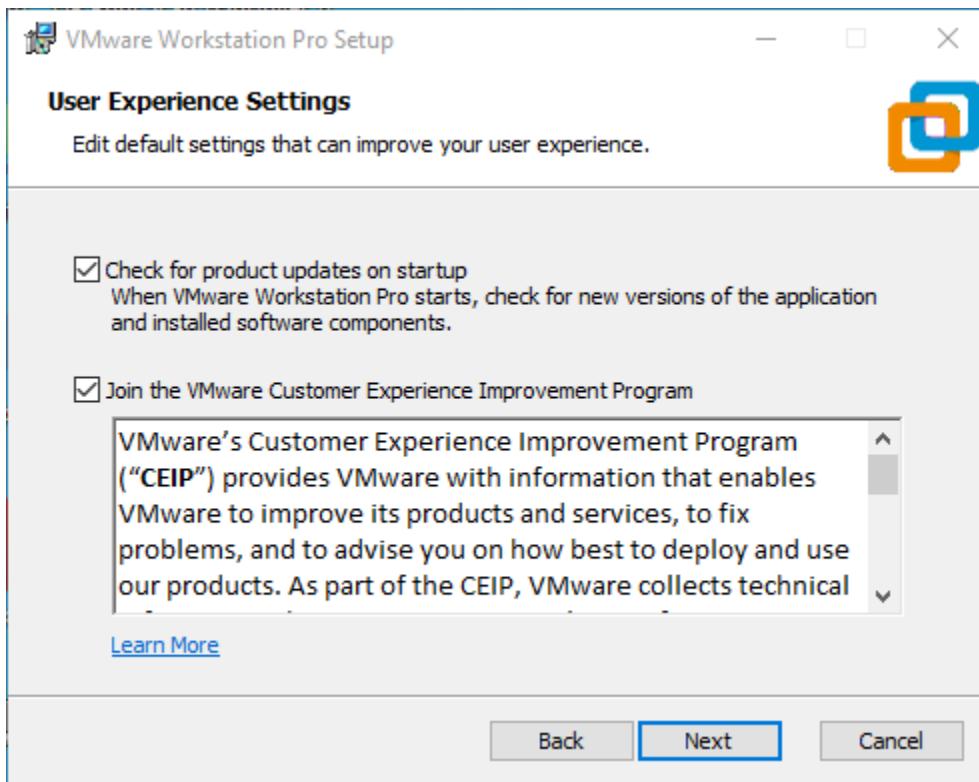
Check I accept the terms in the license agreement and click Next.



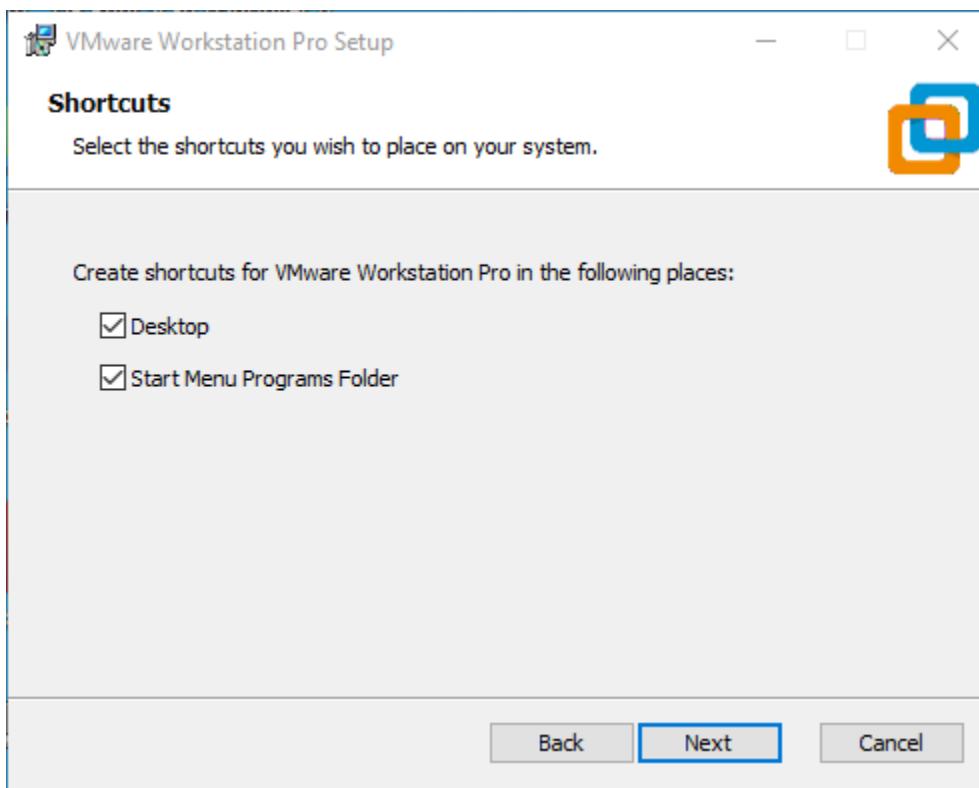
Modify the installation location to the partition of your computer where the software is installed, and click "Next" .



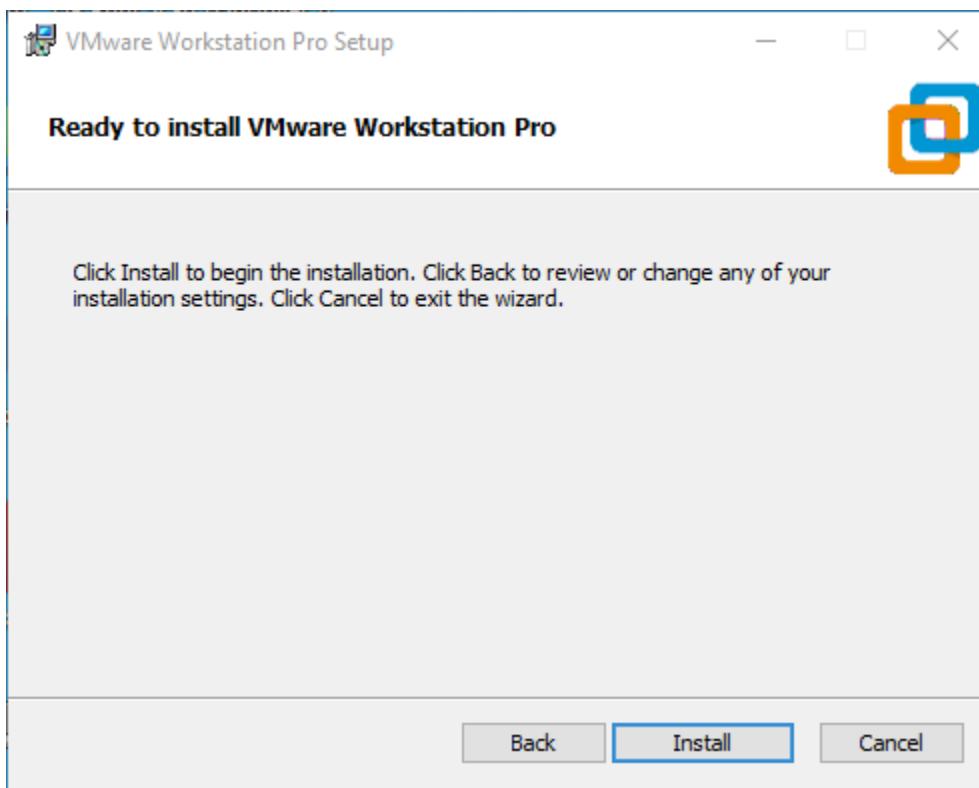
Uncheck and click on “Next” .



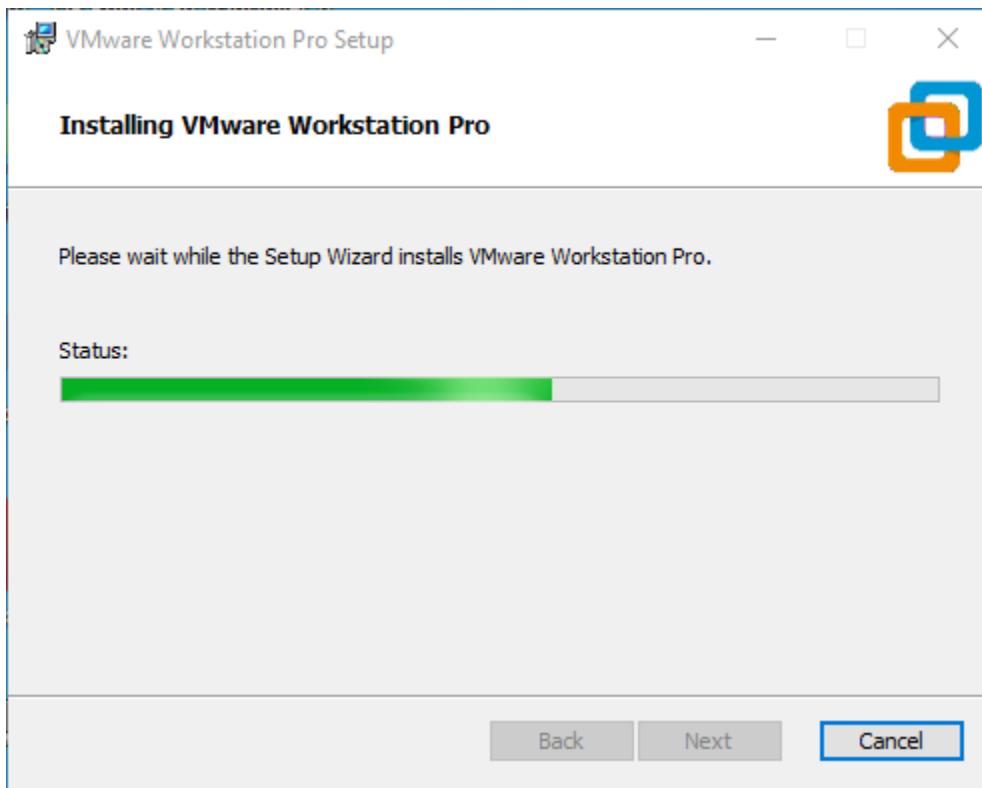
Check Add Shortcut and click “Next” .



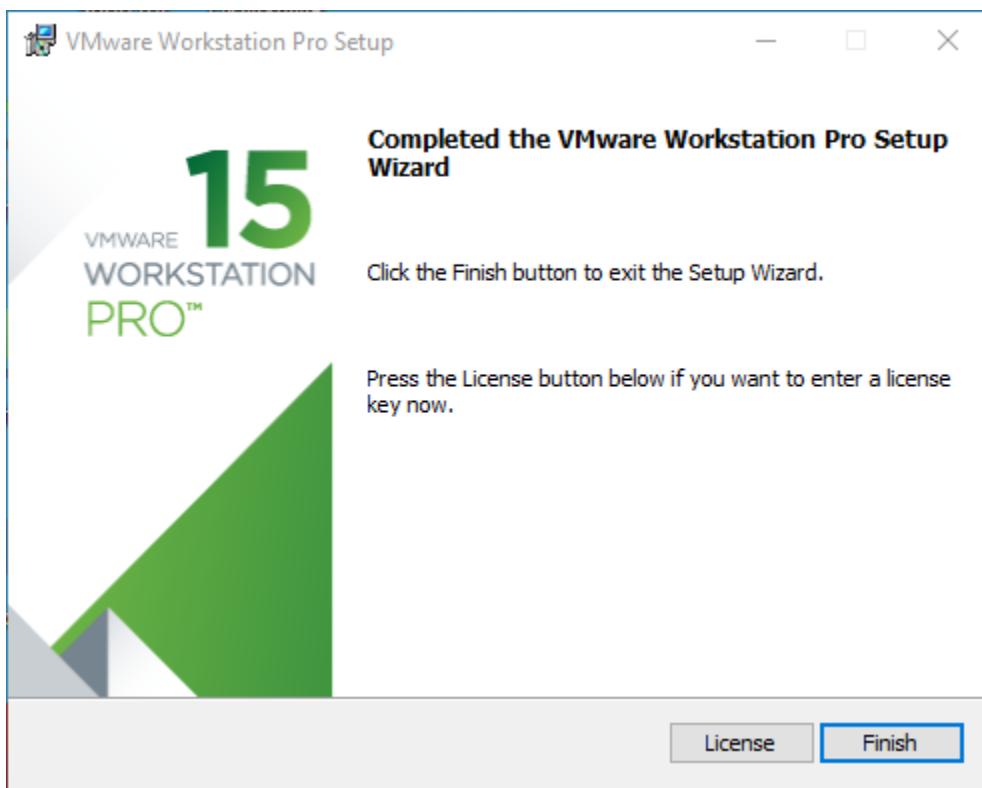
Click "Install"



Wait for the installation to complete.



Click “Finish” to try it out. If users need to use it for a long time, they need to buy it from the official and fill in the license.



2.1.6 2. Ubuntu Development Environment Loading

Note:

- It is recommended for beginners to directly use the pre-built virtual machine environment provided by Forlinx, which already includes installed cross-compiler and Qt environment. After understanding this chapter, you can directly jump to the compilation chapter for further study;
- The development environment provided for general users is: forlinx (username), forlinx (password);
- Please ask your sales representative for the download link.

There are two ways to use a virtual machine environment in VMware: One is to directly load an existing environment, and the other is to create a new environment. Let's first talk about how to load an existing environment.

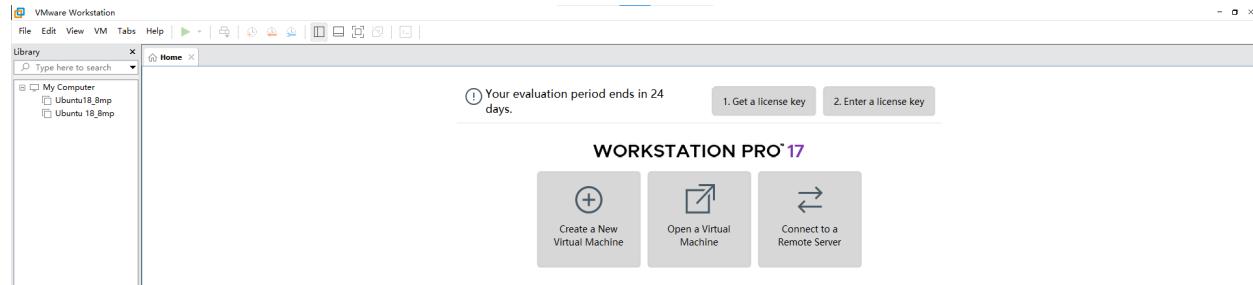
First, download the development environment provided by Forlinx. In the development environment documentation, there should be an MD5 checksum file. After downloading the development environment, you should verify the integrity of the compressed package using the MD5 checksum. (You can use an on-line MD5 checksum tool or download a specific MD5 checksum tool for this purpose). To check if the checksum in the verification file matches the checksum of the file itself. If they match, the file download is successful. If they don't match, it suggests that the file may be corrupt, and you should consider downloading it again.

 md5.txt	2025/7/18 15:33	文本文档	1 KB
 OK3588-Linux5.10.209-VM17-Ubuntu22.04.part01.rar	2025/7/18 15:19	WinRAR 压缩文件	2,097,152...
 OK3588-Linux5.10.209-VM17-Ubuntu22.04.part02.rar	2025/7/18 15:23	WinRAR 压缩文件	2,097,152...
 OK3588-Linux5.10.209-VM17-Ubuntu22.04.part03.rar	2025/7/18 15:27	WinRAR 压缩文件	1,514,868...
 readme.txt	2025/6/25 15:21	文本文档	1 KB

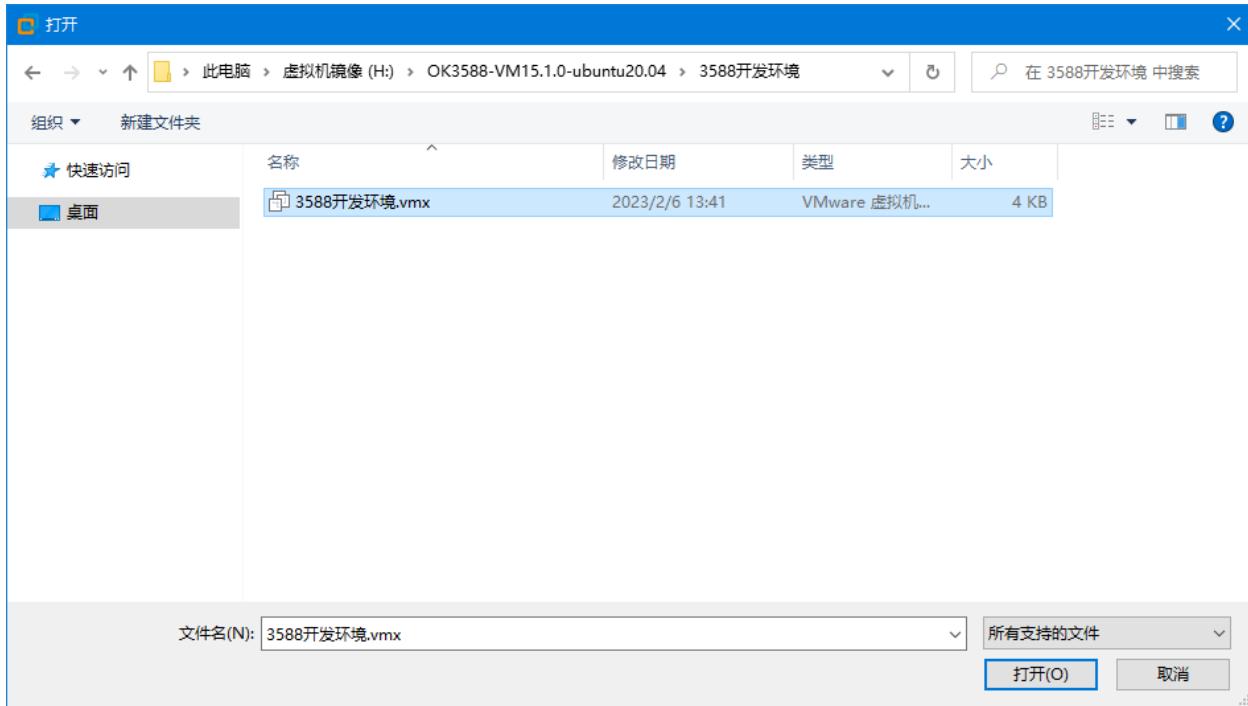
Select all compressed files, right-click and extract to the current folder or your own directory: Unzip it and get the development environment 3588.

3588 development environment.vmx in the 3588 folder is the file to be opened by the virtual machine.

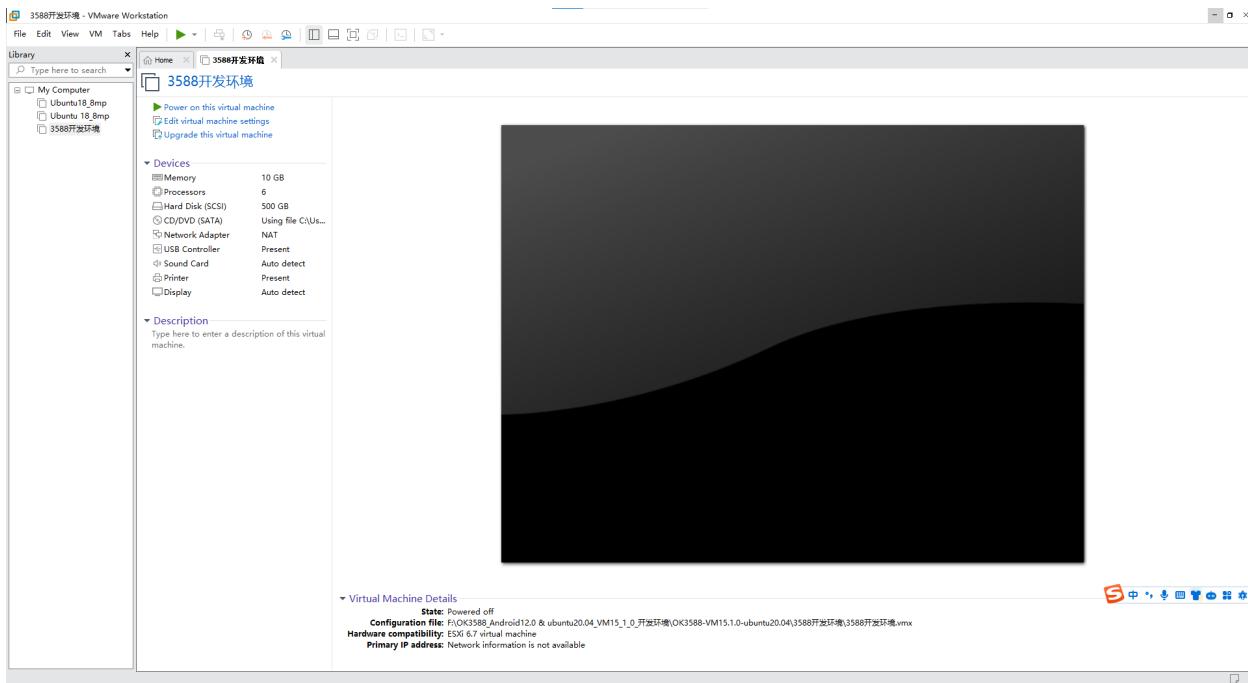
Open the installed virtual machine.



Navigate to the directory where the recently extracted OK3588-VM15.1.0-ubuntu20.04 virtual machine file is located, and double-click on the startup file to open it.



Turn on this virtual machine after loading is complete to run it and enter the system's interface.



The default automatic login account is “forlinx”, and the password is “forlinx” .

2.1.7 3. New Ubuntu Development Environment Setup

Note: Beginners are not recommended to set up a system on your own. It is recommended to use an existing virtual machine environment. If you do not need to set up the environment, you can skip this section.

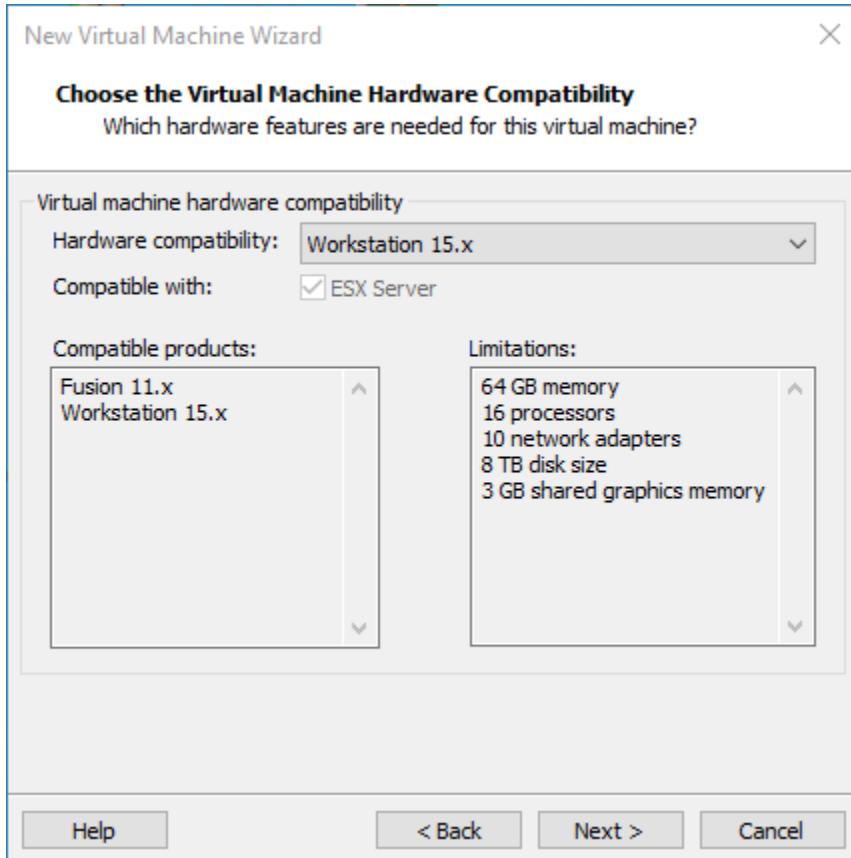
3.1 Ubuntu System Setup

3.1.1 Creating an Ubuntu Virtual Machine

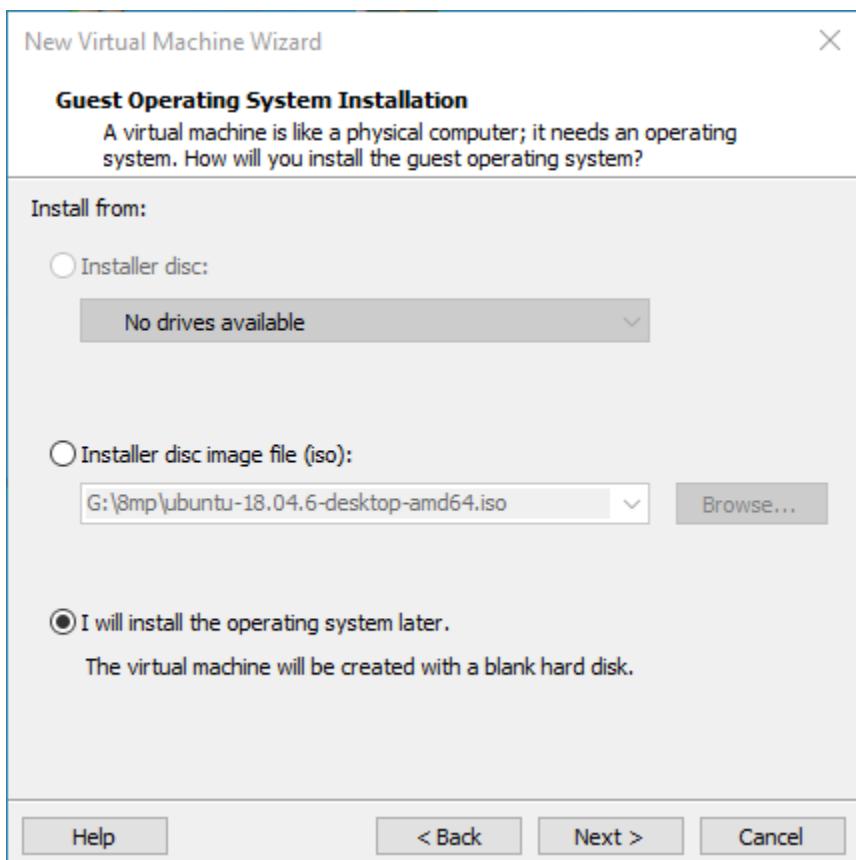
Step 1: Open the VMware software and click [File]/ [New Virtual Machine]. Enter the following interface:



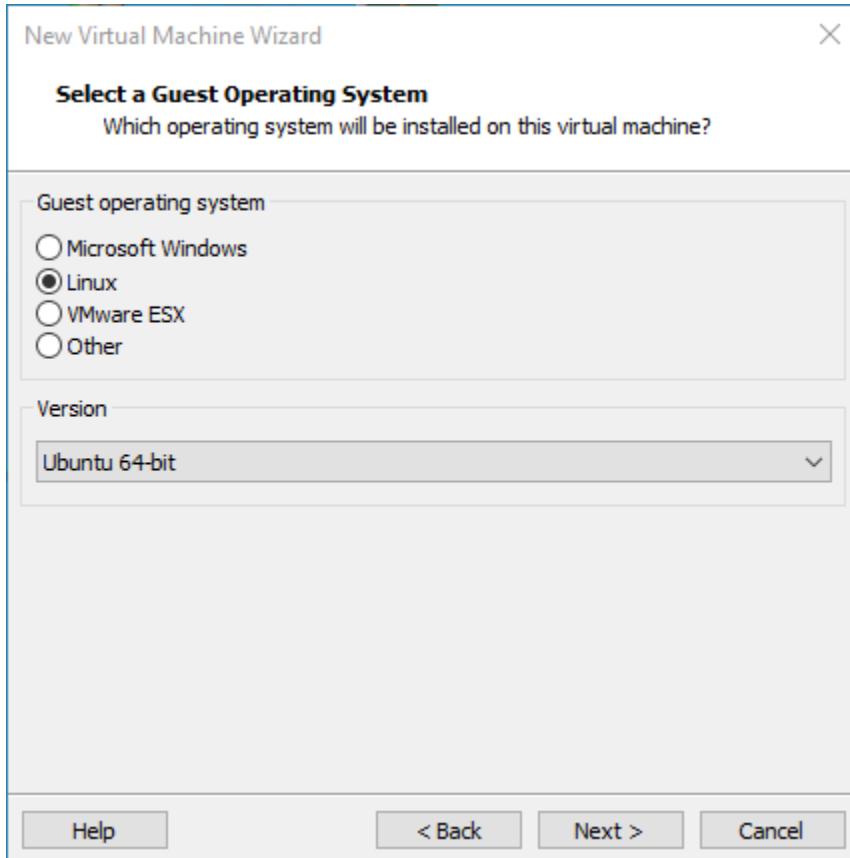
Step 2: Select Custom and click “Next” .



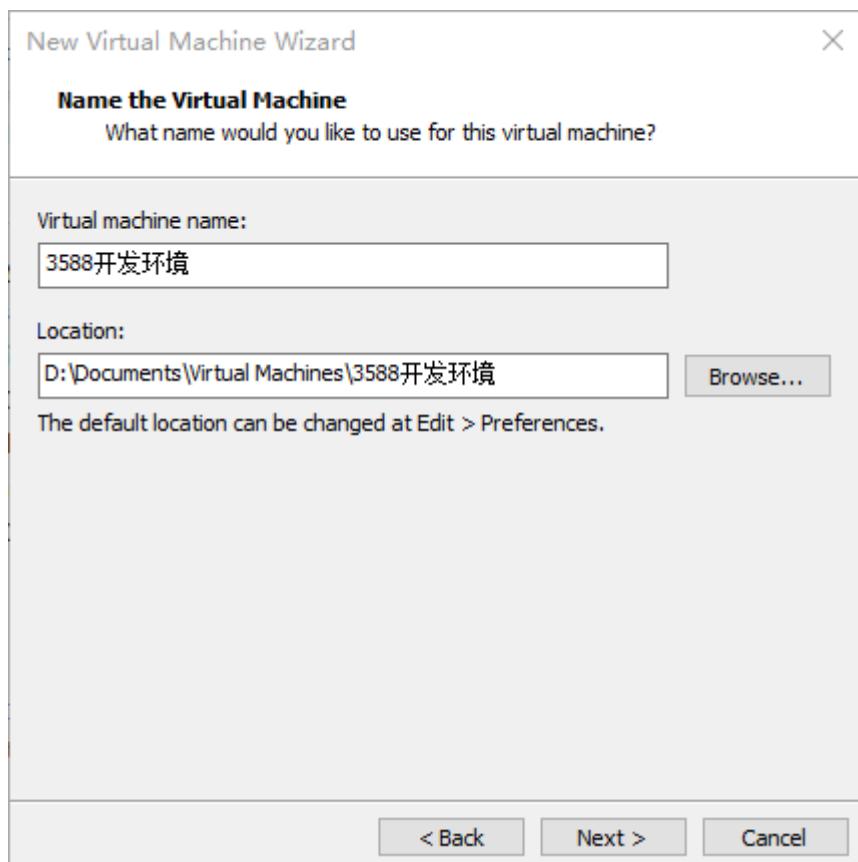
Select the compatibility with the corresponding version of VMware, which can be found in Help->About VMware Workstation, and click “Next” .



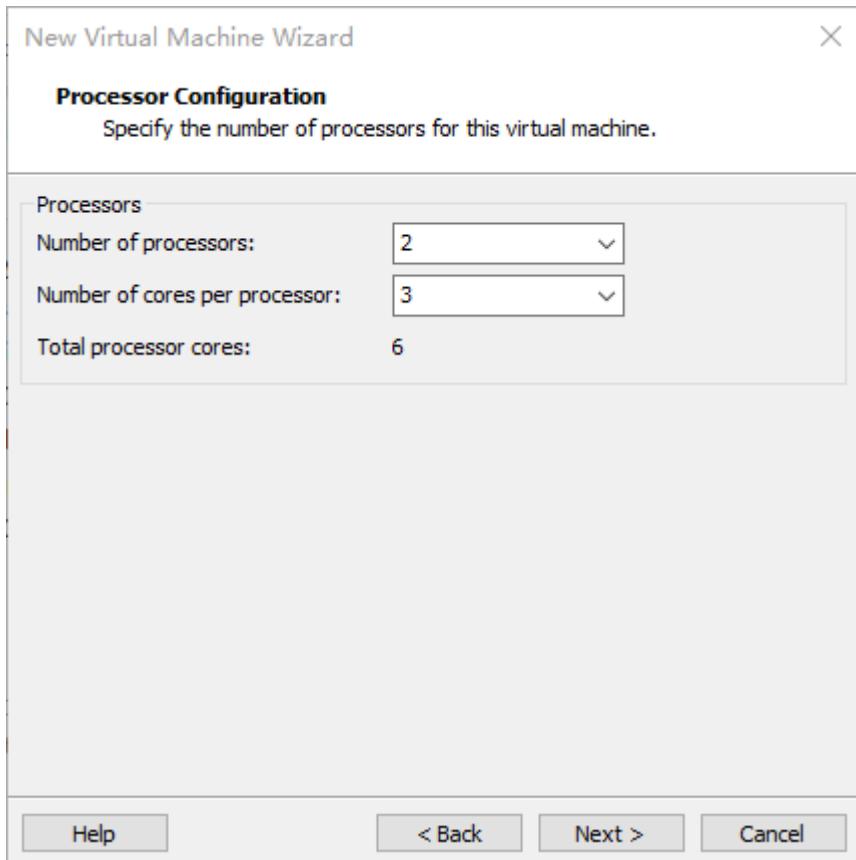
Select Install the operating system later and click “Next” .



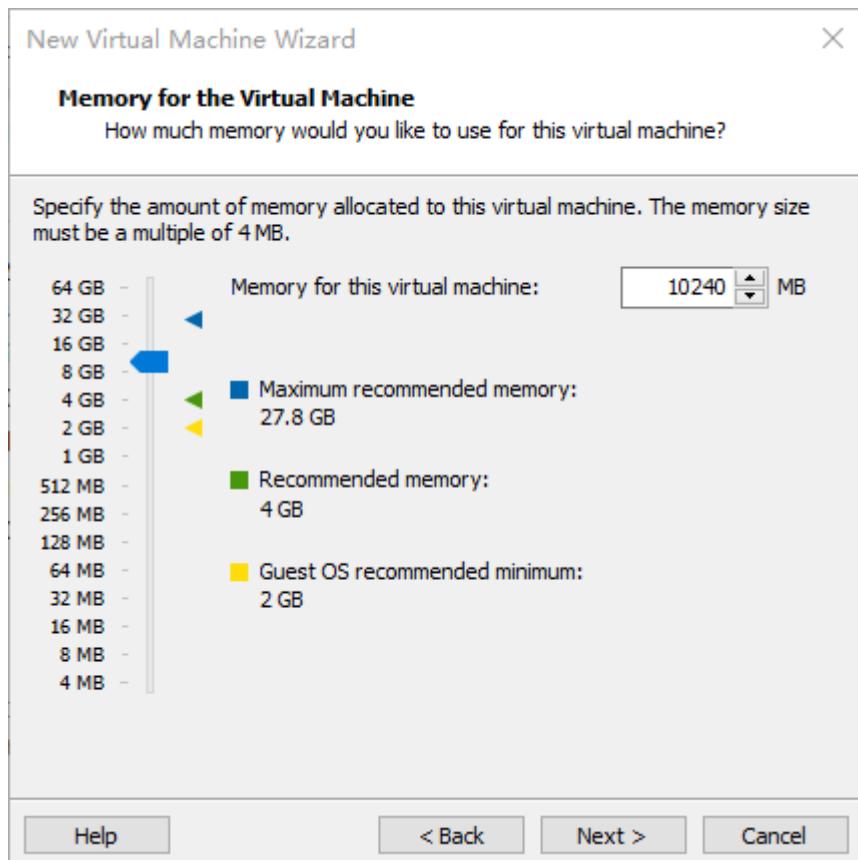
Keep the default settings and click “Next” .



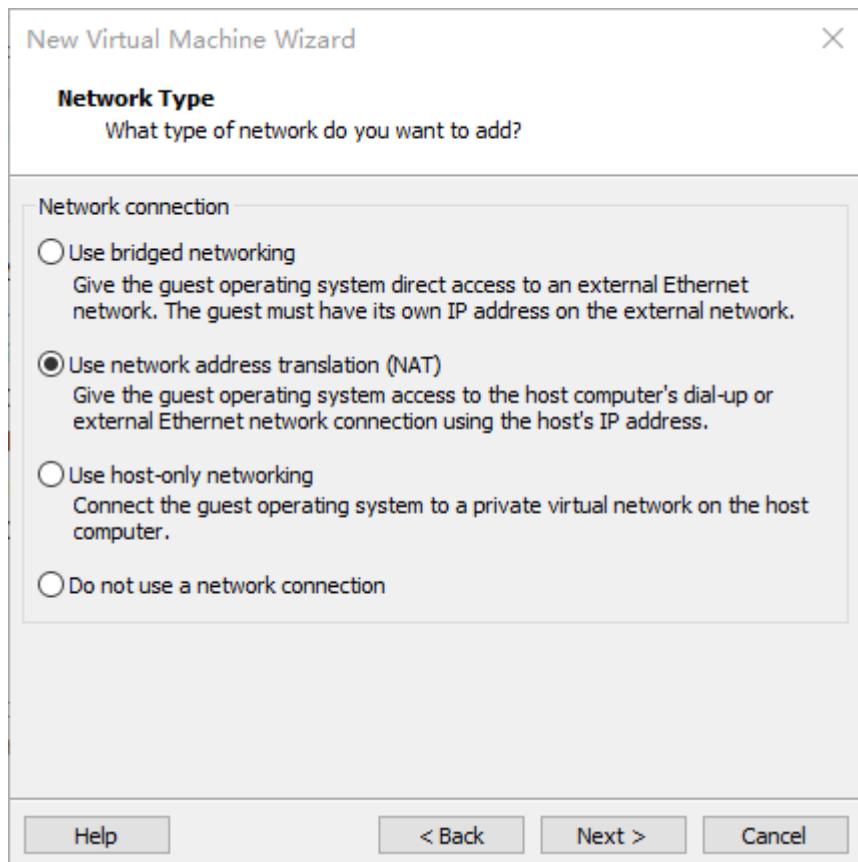
Modify the name and installation location of your virtual machine, and click “Next” .



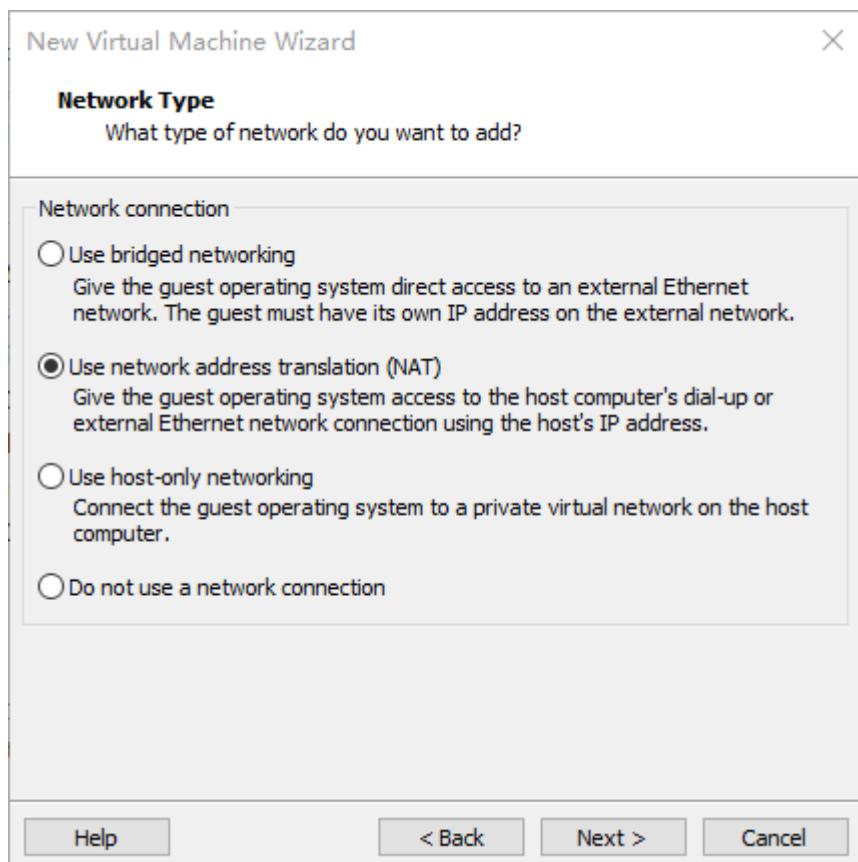
Configure the number of CPU based on your computer's actual specifications.



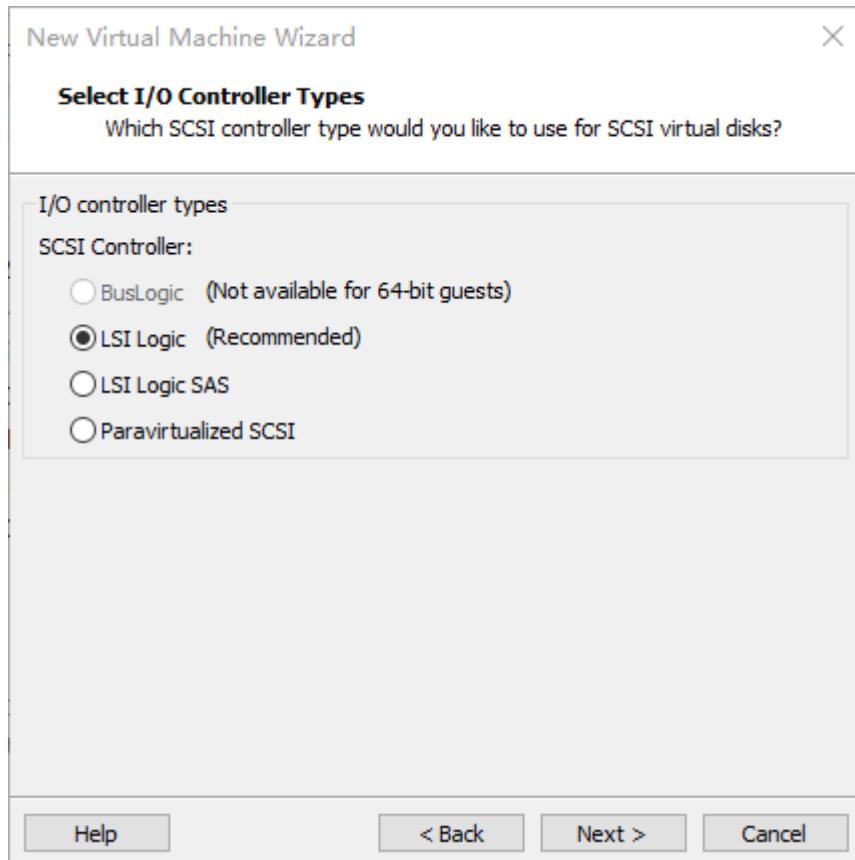
Set the memory size according to the actual situation. It is recommended to use 16G.



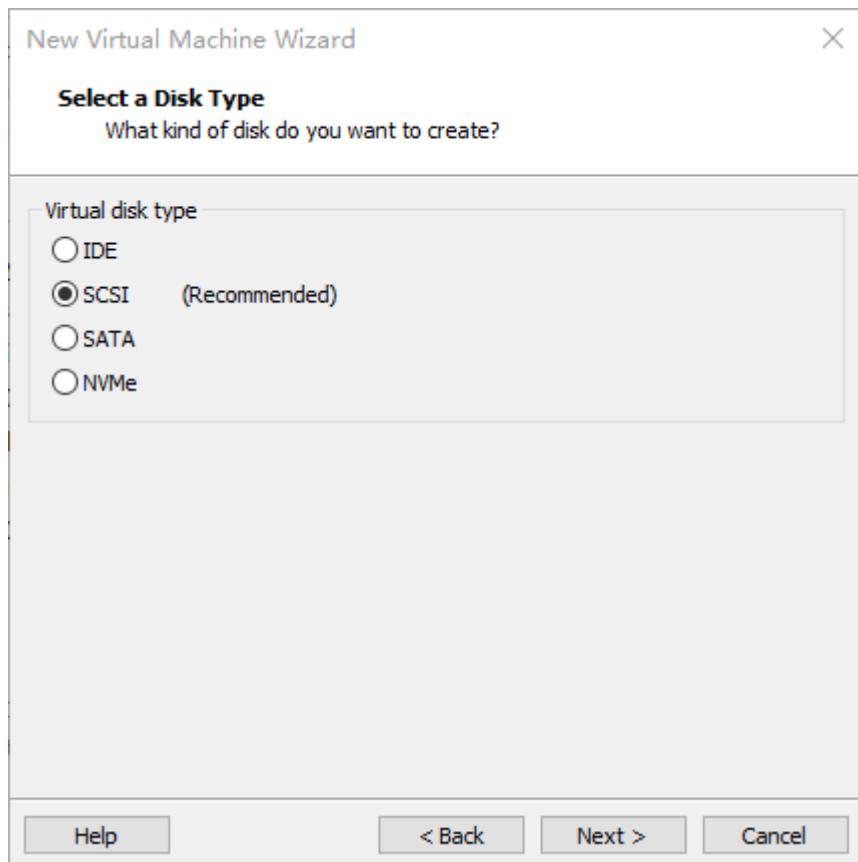
Set the network type, default to NAT mode then click “Next” . Keep the default values for the remaining steps until you reach the step to specify the disk capacity.



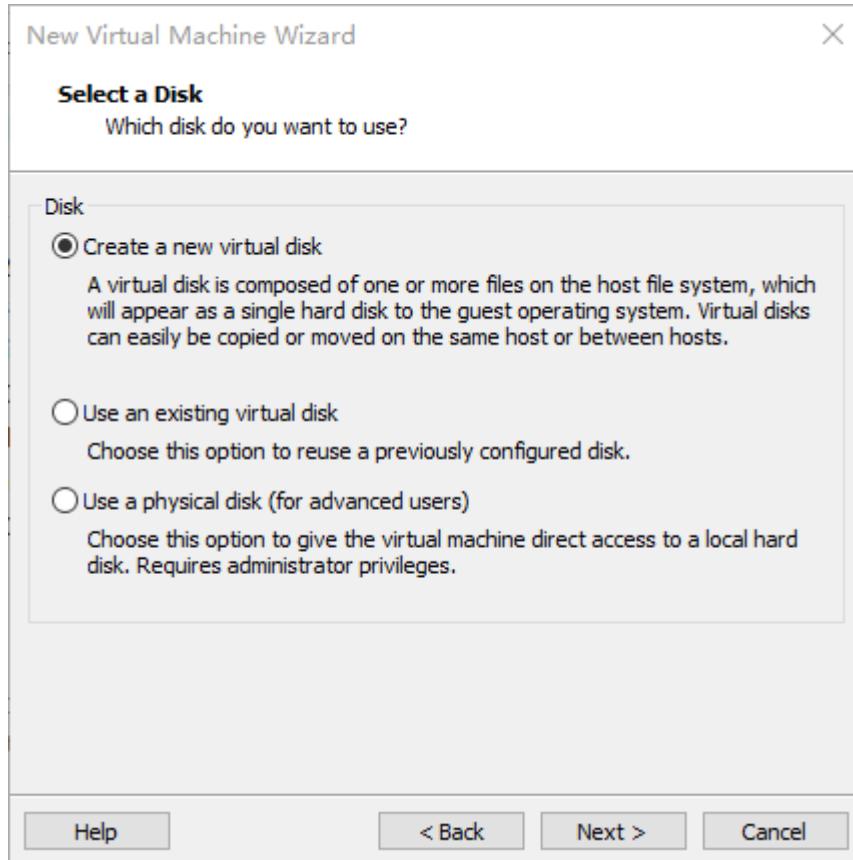
The default selection for the IO controller type here is LSI.



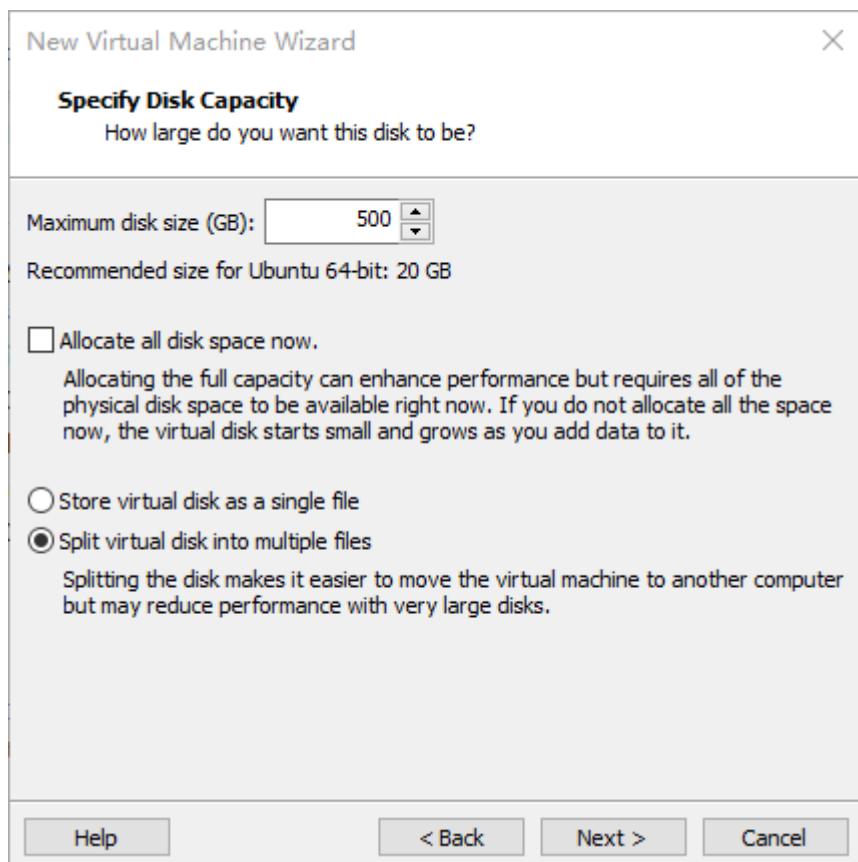
The default selection here is also SCSI.



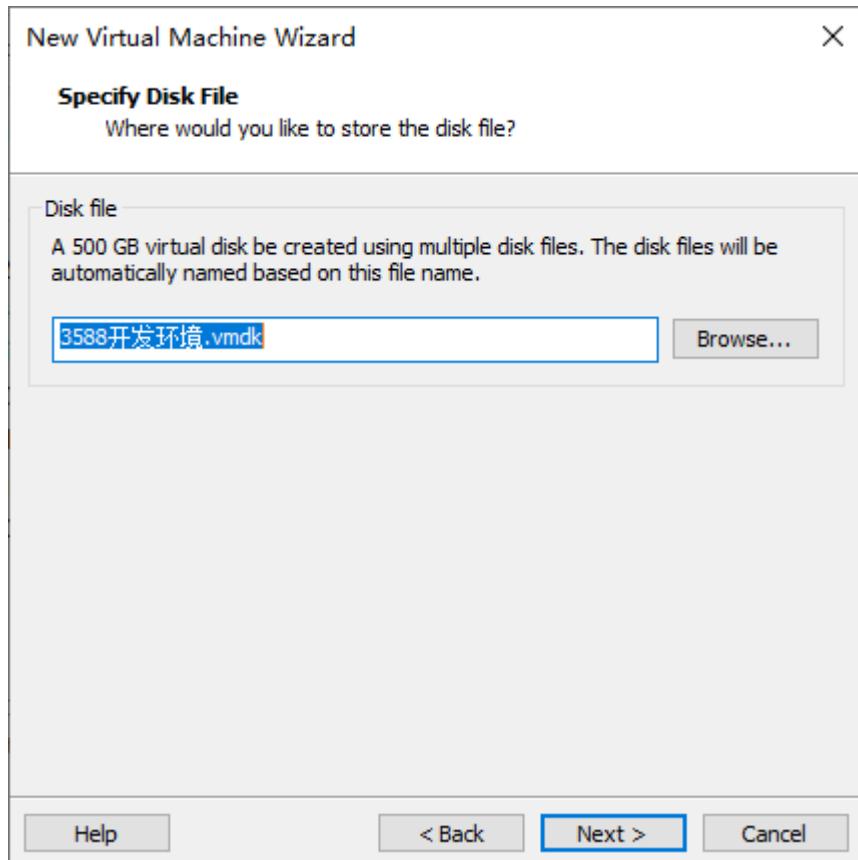
Choose to create a new virtual disk here.



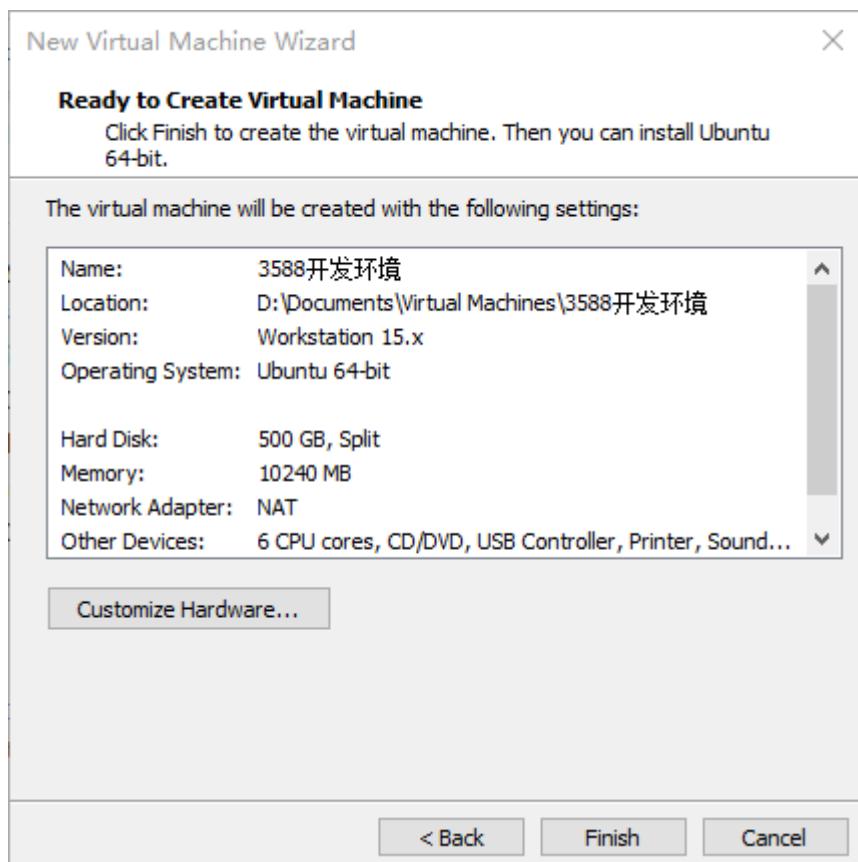
Set the disk size to 500GB, select the disk provisioning format, and then click “Next” .



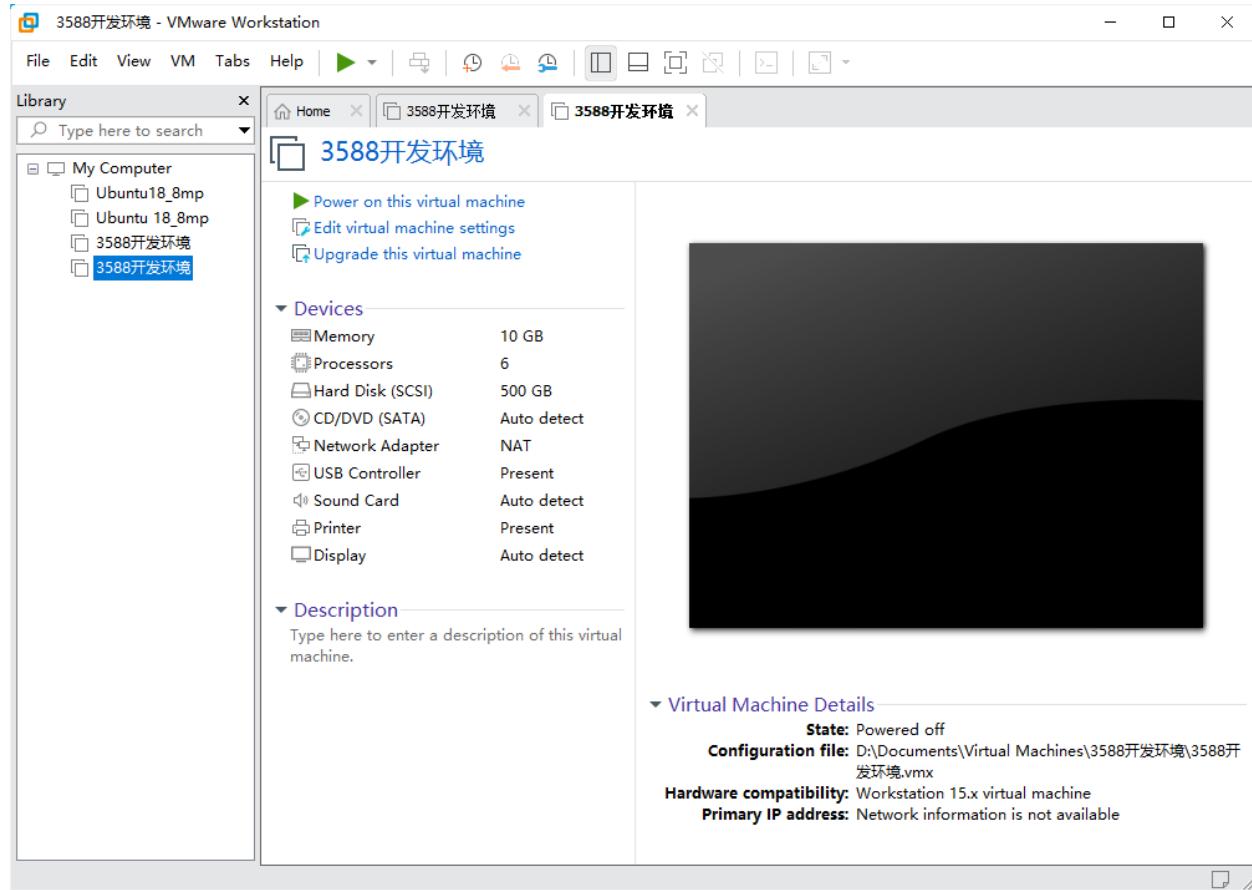
Specify the disk file, the default one here is OK.



Click “Finish” by default.



The virtual machine creation is now complete.



In the next section, the installation of the Ubuntu system in a virtual machine will be introduced. The installation method on a real machine is similar to that in a virtual machine. Here, the method of installing the Ubuntu system in a virtual machine is presented.

3.1.2 System Installation

In the previous section, a virtual machine was created, but no operating system was installed, so the virtual machine still cannot be started. Next, the Ubuntu operating system will be installed in the newly created virtual machine.

Step 1: First, go to the official Ubuntu website to obtain the 64-bit image of Ubuntu 22.04. The download address is:

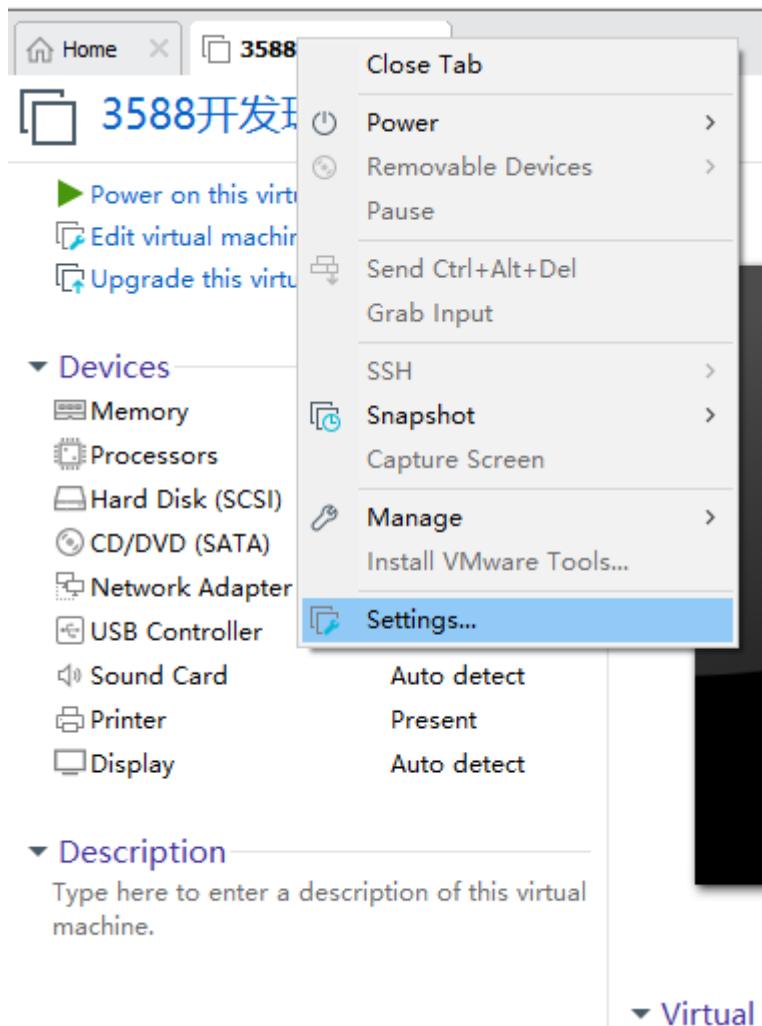
<https://old-releases.ubuntu.com/releases/22.04.4/>

The installed Ubuntu version is 22.04. The reason for choosing 22.04 is that all our source code compilation operations have been compiled and verified on this version. These operations may vary slightly between Ubuntu system versions.

Download “ubuntu-22.04.4-desktop-amd64.iso”

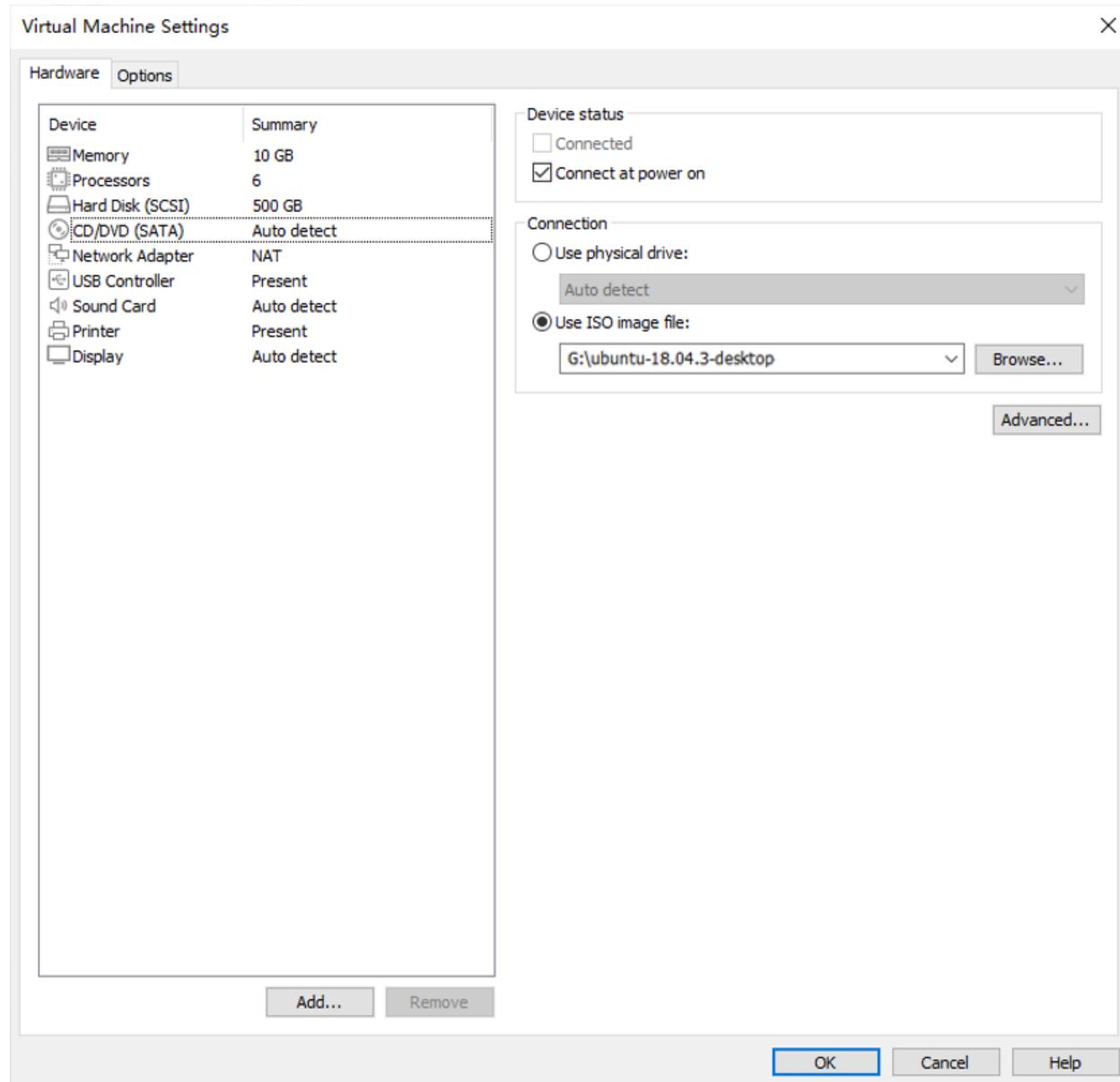
After downloading the image, you can proceed with the system installation operation.

Right-click on the created virtual machine name and select “Settings” from the pop-up menu.

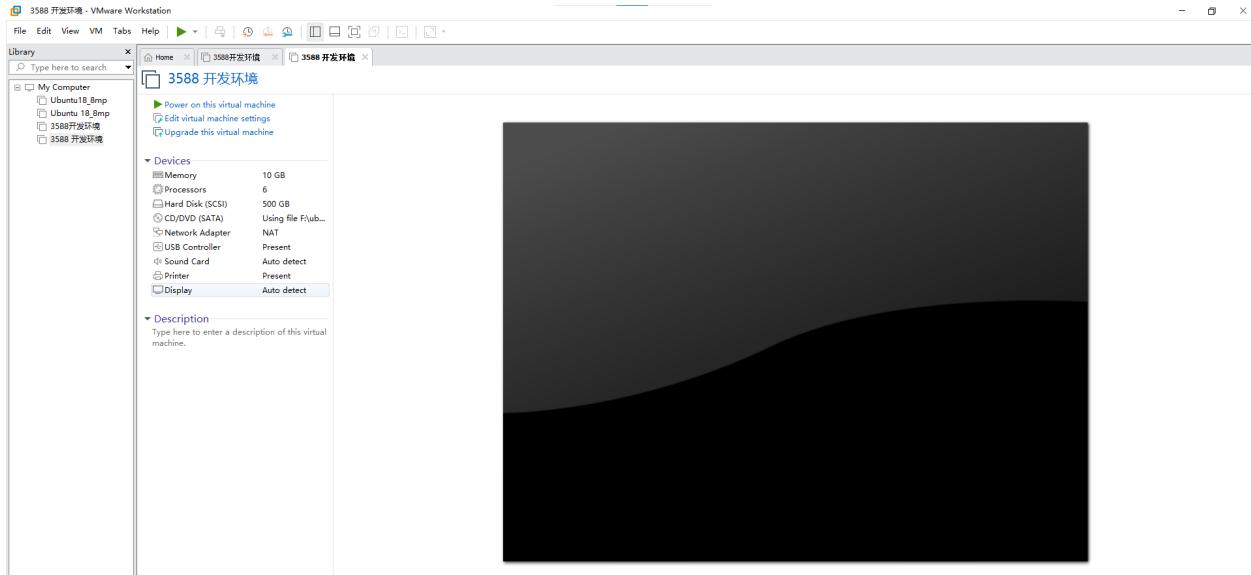


▼ Virtual I

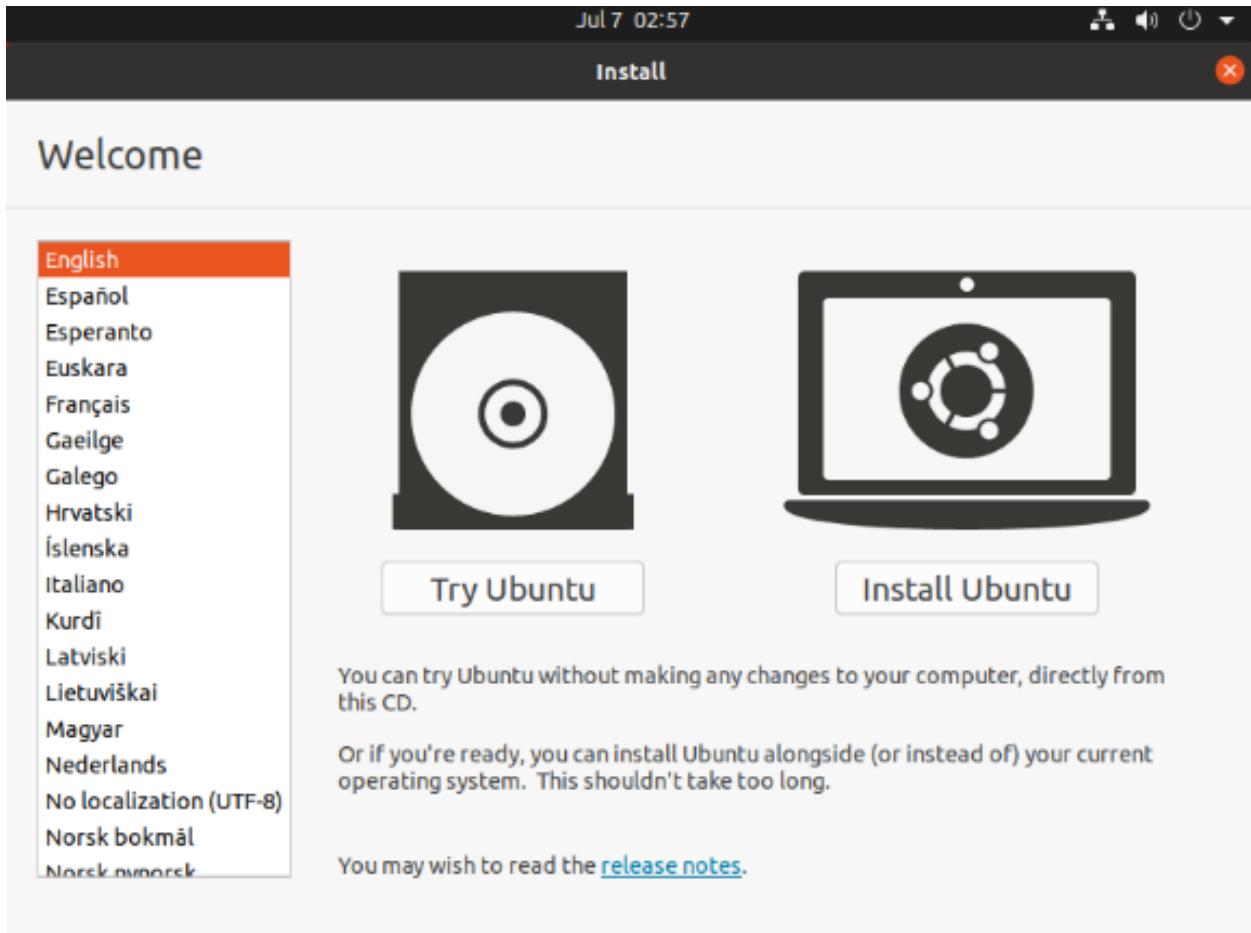
The “Virtual Machine Settings” menu will pop up. Click on CD/DVD (SATA), select “Use ISO image file,” browse and choose the previously downloaded Ubuntu image, then click “OK” to confirm.



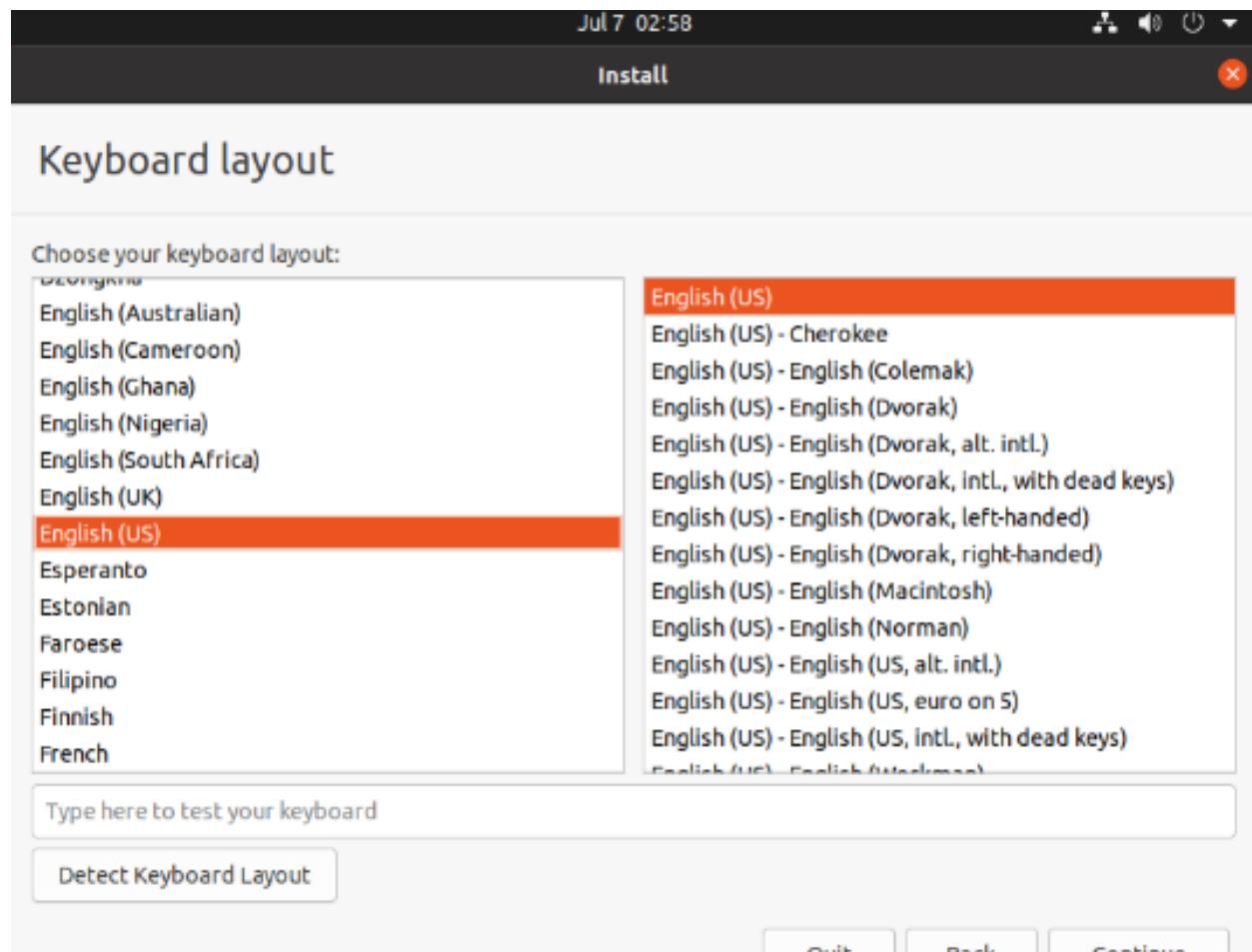
After setting up the image, ensure that the network is available. Then, start the virtual machine and proceed with the installation of the Ubuntu image.



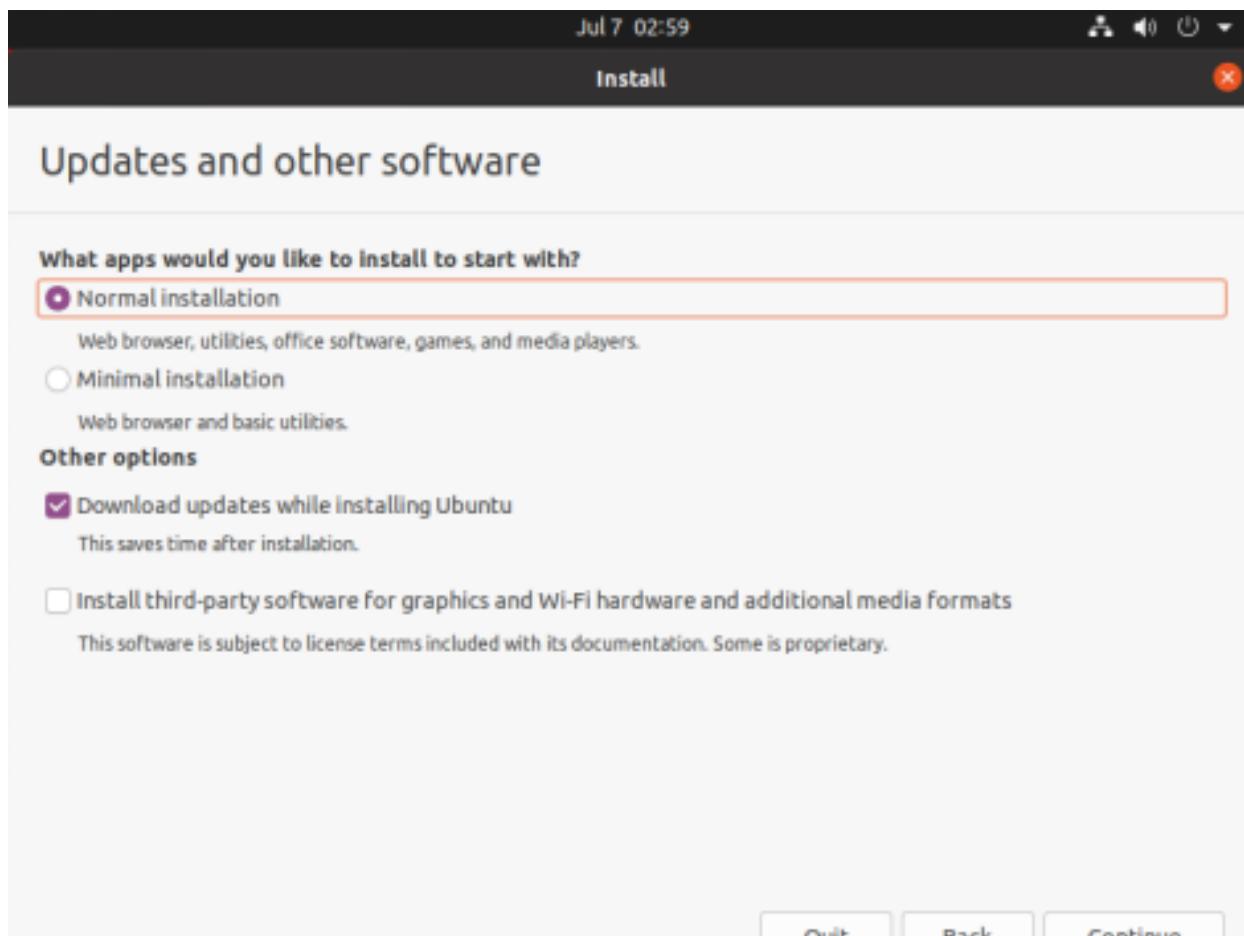
After starting the virtual machine, wait for the installation interface to appear as shown below:



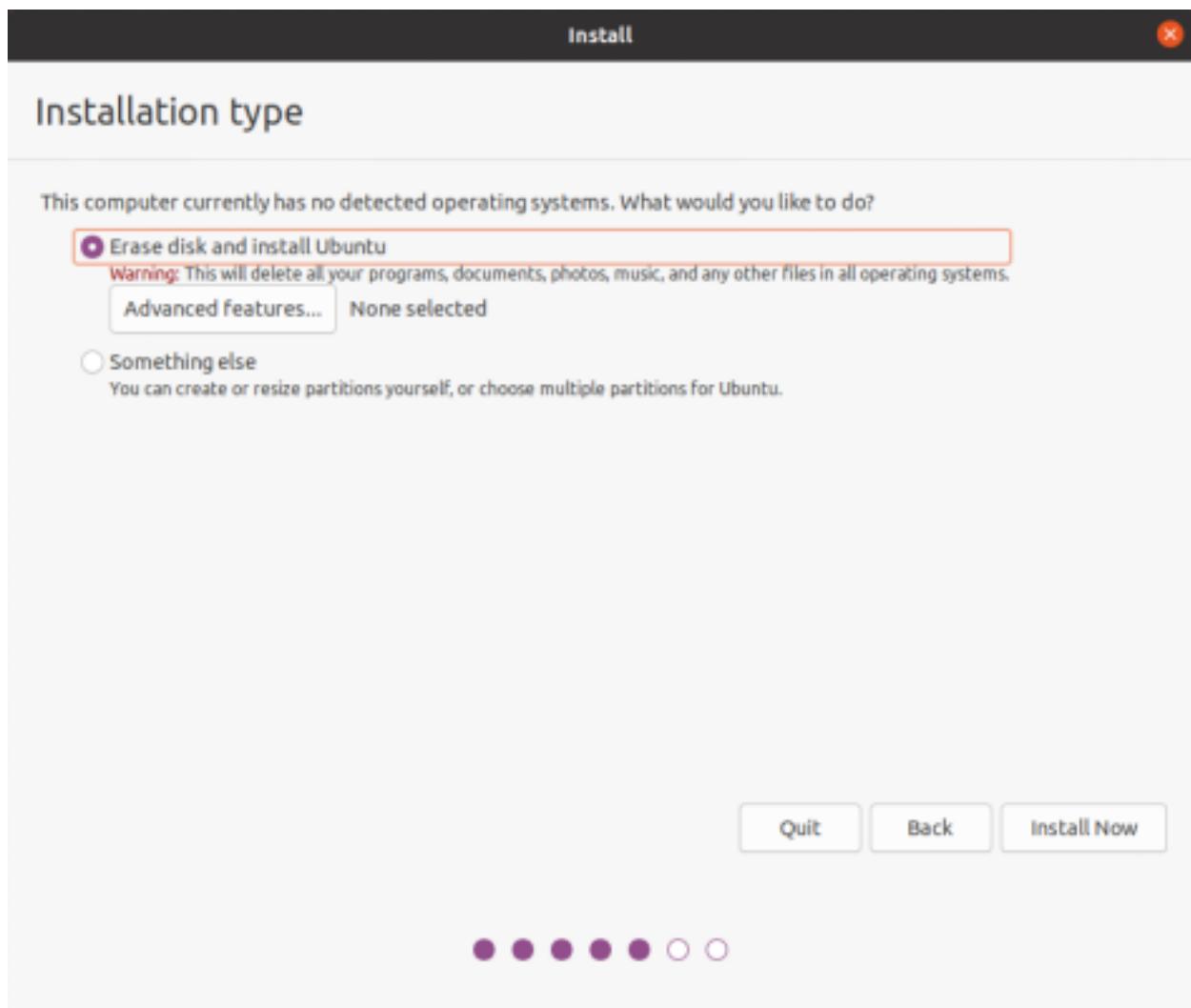
After selecting the language on the left side as shown in the image, click “Install Ubuntu”, and the language selection interface will pop up. The default language of Ubuntu is English, but of course, you can also choose others.



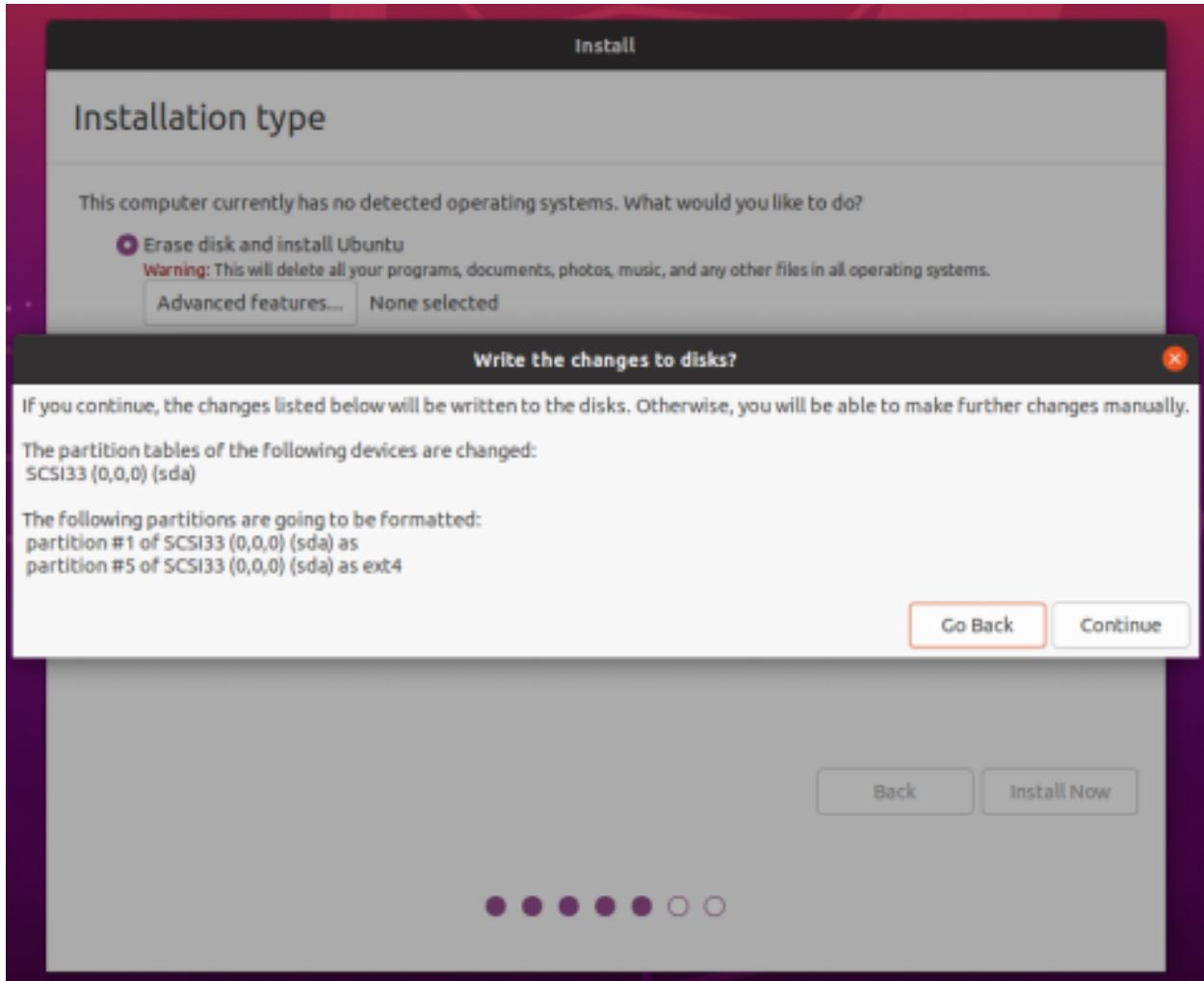
The default selected language can also be reset later, after the selection is complete continue.



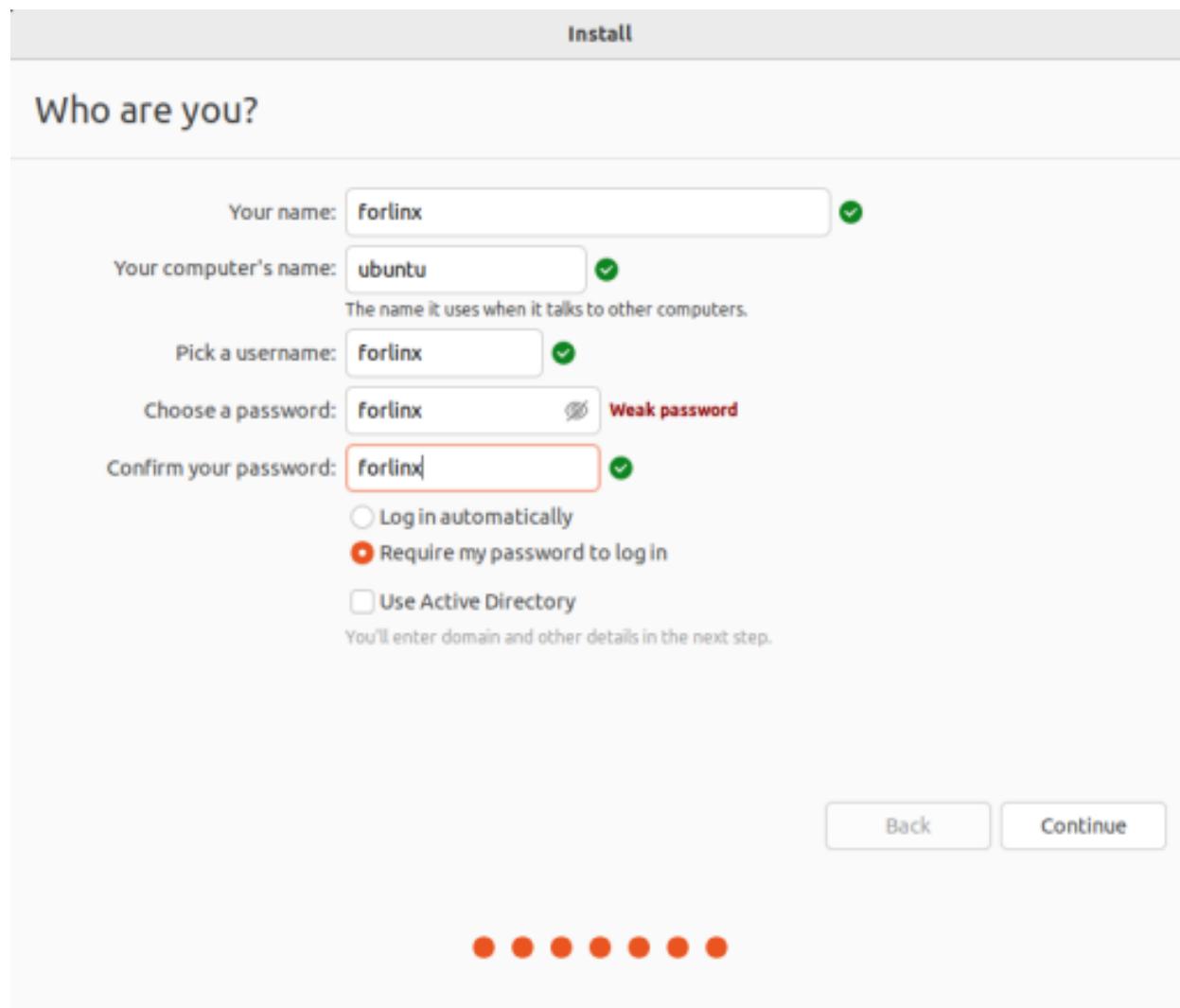
Next, select “Continue” as the default option to proceed with the installation. The installation process might be slow. Then, click “Continue” again.



By default, when you click on “Install Now”, a dialog box will appear as shown in the image. Simply click “Continue” to proceed.



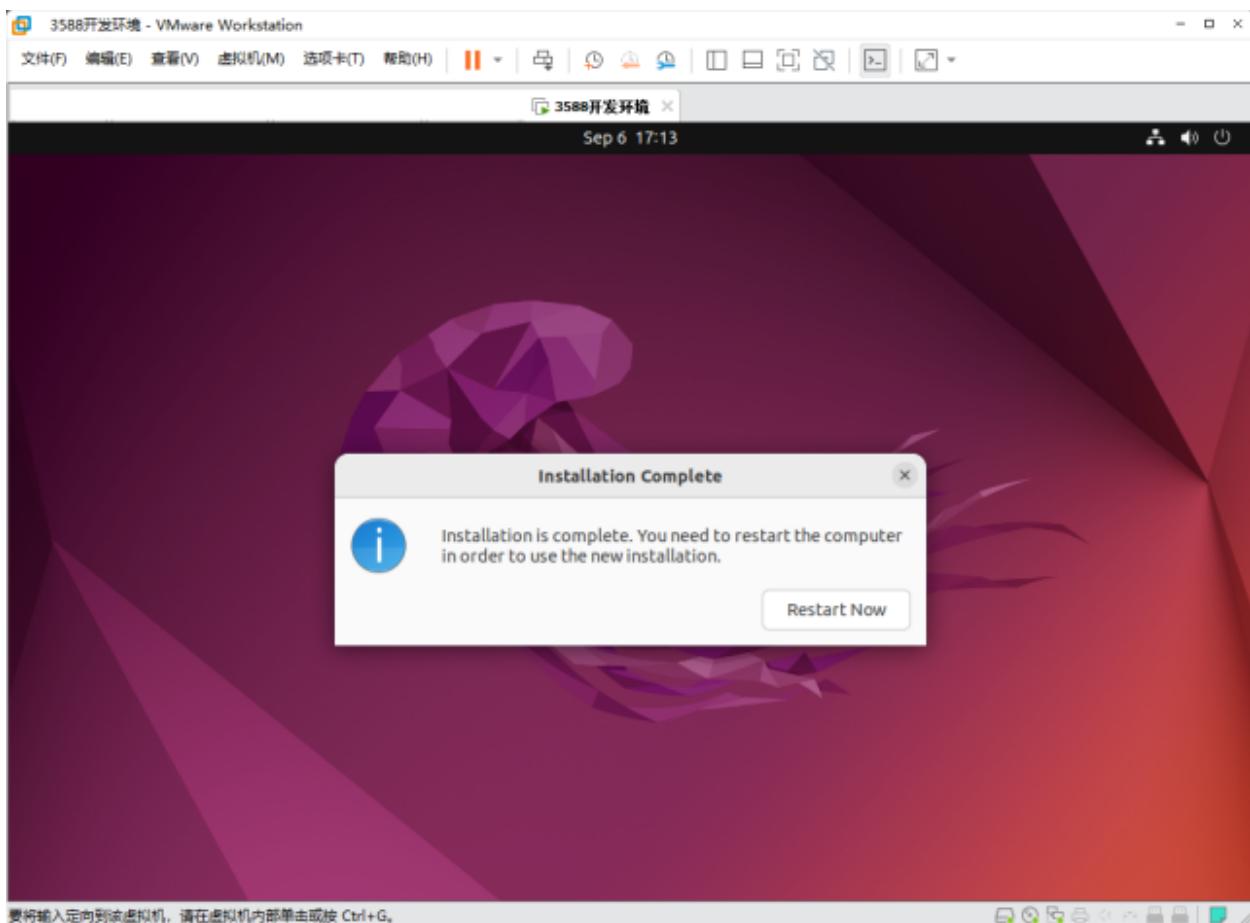
Next, select the timezone. You can either click on the Shanghai timezone or enter “Shanghai” (or choose the appropriate timezone based on your location). Then, click “Continue” to proceed. Finally, set your username and password. You can choose either automatic login or login with a username and password. Click “Continue” to start the automatic installation.



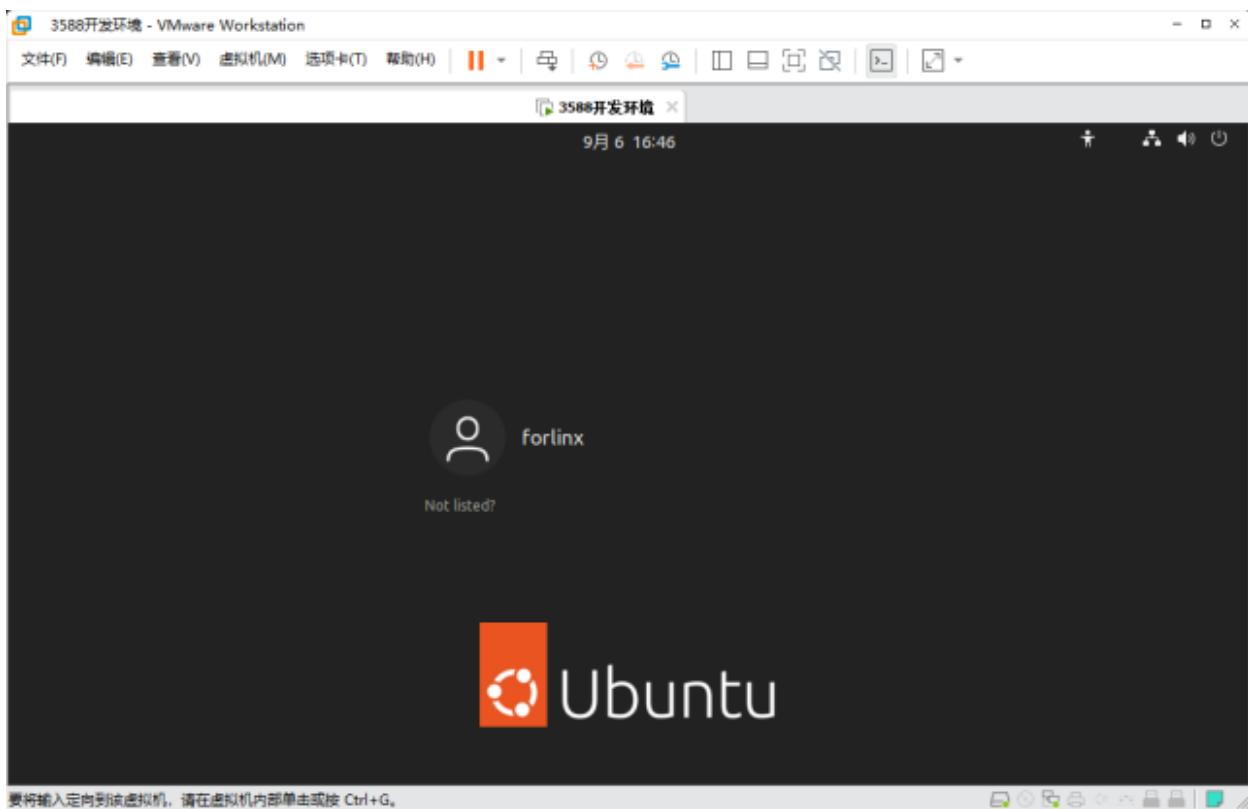
If the network is not good, it can be skipped without affecting the installation.



Click “Restart Now” to reboot.



The system interface after the reboot is complete.



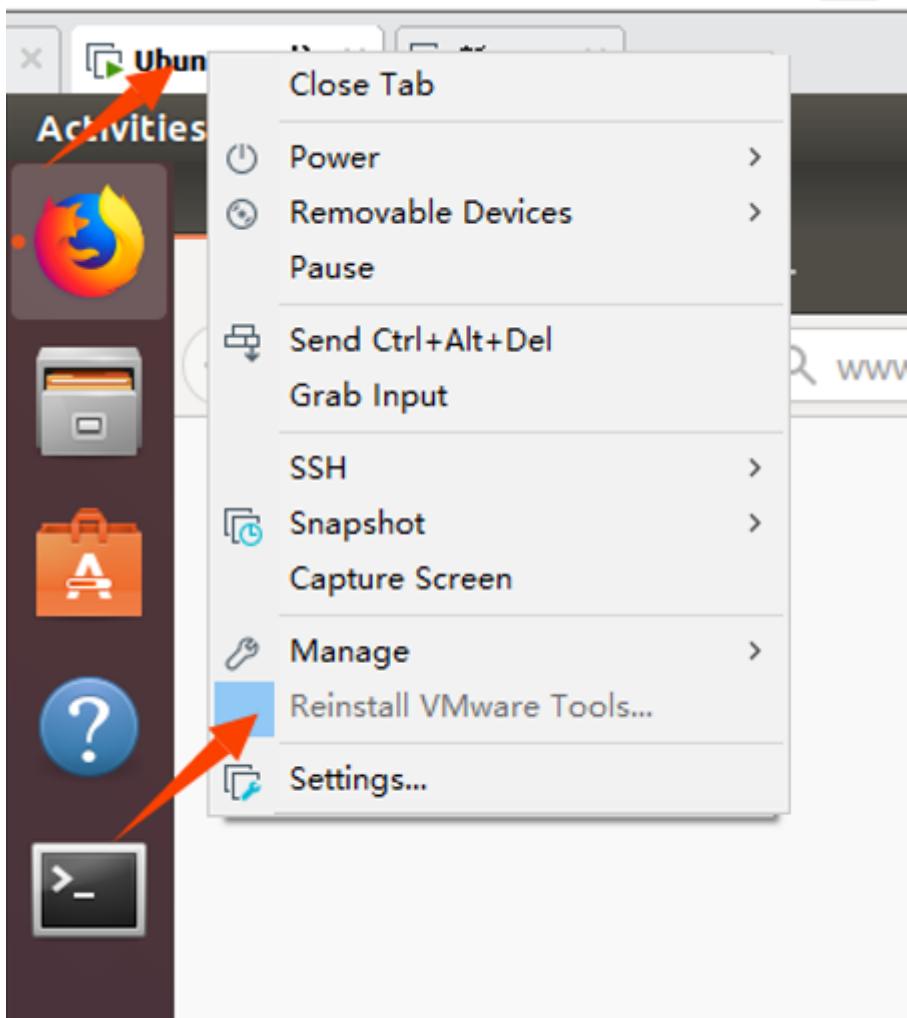
The ubuntu system installation is complete.

3.1.3 Ubuntu Basic Configuration

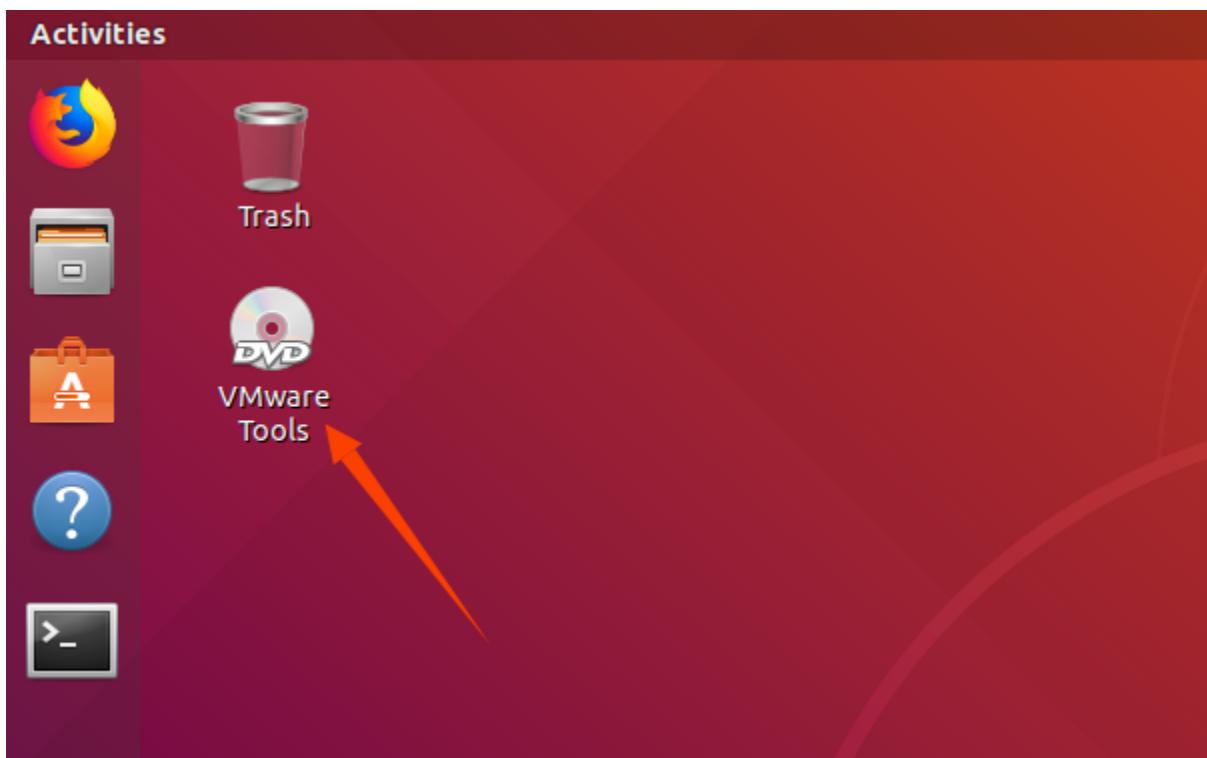
After installing the Ubuntu20.04 operating system, there are a few configurations to make.

VMware Tools Installation:

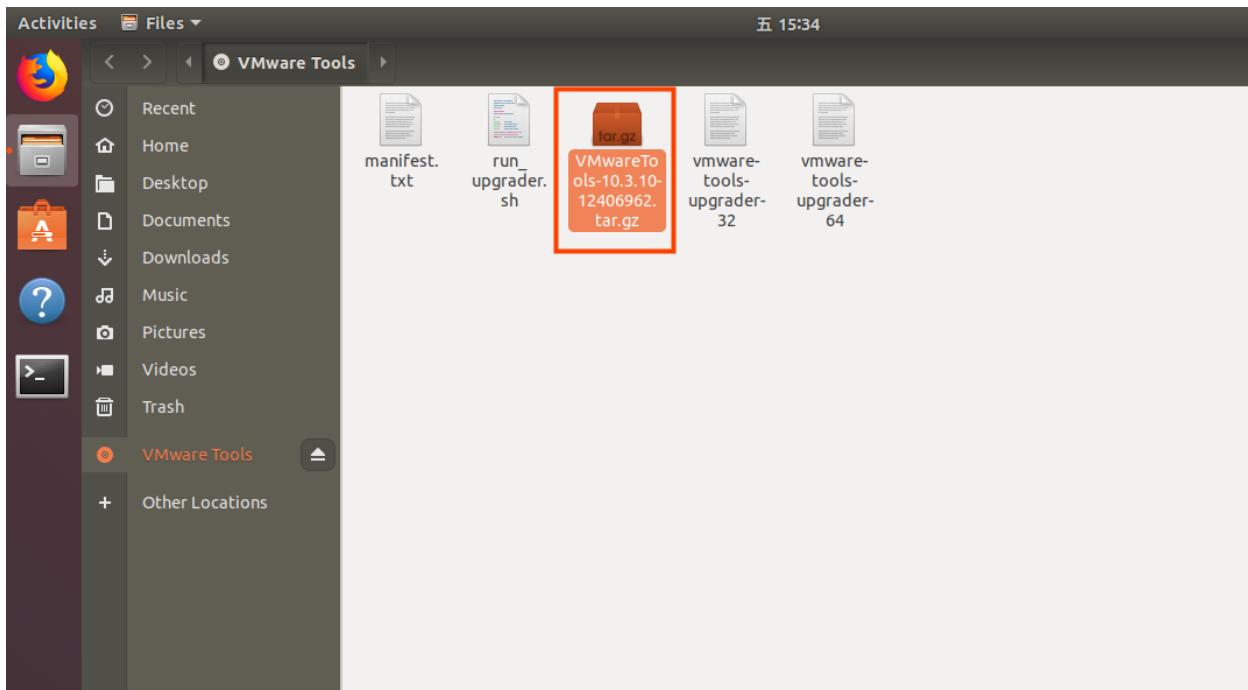
Next, install VMware Tools. Without installing this tool, you won't be able to copy and paste and drag file between the Windows host and the virtual machine. First click on "Virtual Machine" on the VMware navigation bar, then click "Install VMware Tools" in the drop-down box.



Once done, enter Ubuntu and the VMware Tools CD will appear on your desktop and click into it.



Enter and see a compressed file `VMwareTools-10.3.10-12406962.tar.gz` (it may be different for different VM versions); copy the file under the home directory (i.e. the directory with the home personal username)



Press [Ctrl+Alt+T] to bring up the Terminal Command Interface and enter the command:

```
forlinx@ubuntu:~$ sudo tar xvf VMwareTools-10.3.10-12406962.tar.gz
```

```
forlinx@ubuntu: ~
File Edit View Search Terminal Help
forlinx@ubuntu:~$ sudo tar xvf VMwareTools-10.3.10-12406962.tar.gz
[sudo] password for forlinx: [ ]
```

After the extraction is complete, a file named “vmware-tools-distrib” will appear.

```
vmware-tools-distrib/doc/open_source_licenses.txt
vmware-tools-distrib/doc/README
vmware-tools-distrib/vmware-install.pl
forlinx@linux:~$ ls
Desktop examples.desktop Public VMwareTools-10.3.10-12406962.tar.gz
Documents Music Templates vmware-tools-distrib
Downloads Pictures Videos
forlinx@linux:~$ [ ]
```

Go back to the terminal and type: cd vmware-tools-distrib to enter the directory.

Enter: sudo ./vmware-install.pl press enter to input the password, then the installation will begin. For any other inquiries, simply press Enter to go with the default installation settings.

```
forlinx@linux:~/vmware-tools-distrib$ sudo ./vmware-install.pl
open-vm-tools packages are available from the OS vendor and VMware recommends
using open-vm-tools packages. See http://kb.vmware.com/kb/2073803 for more
information.
Do you still want to proceed with this installation? [no] yes[ ]
```

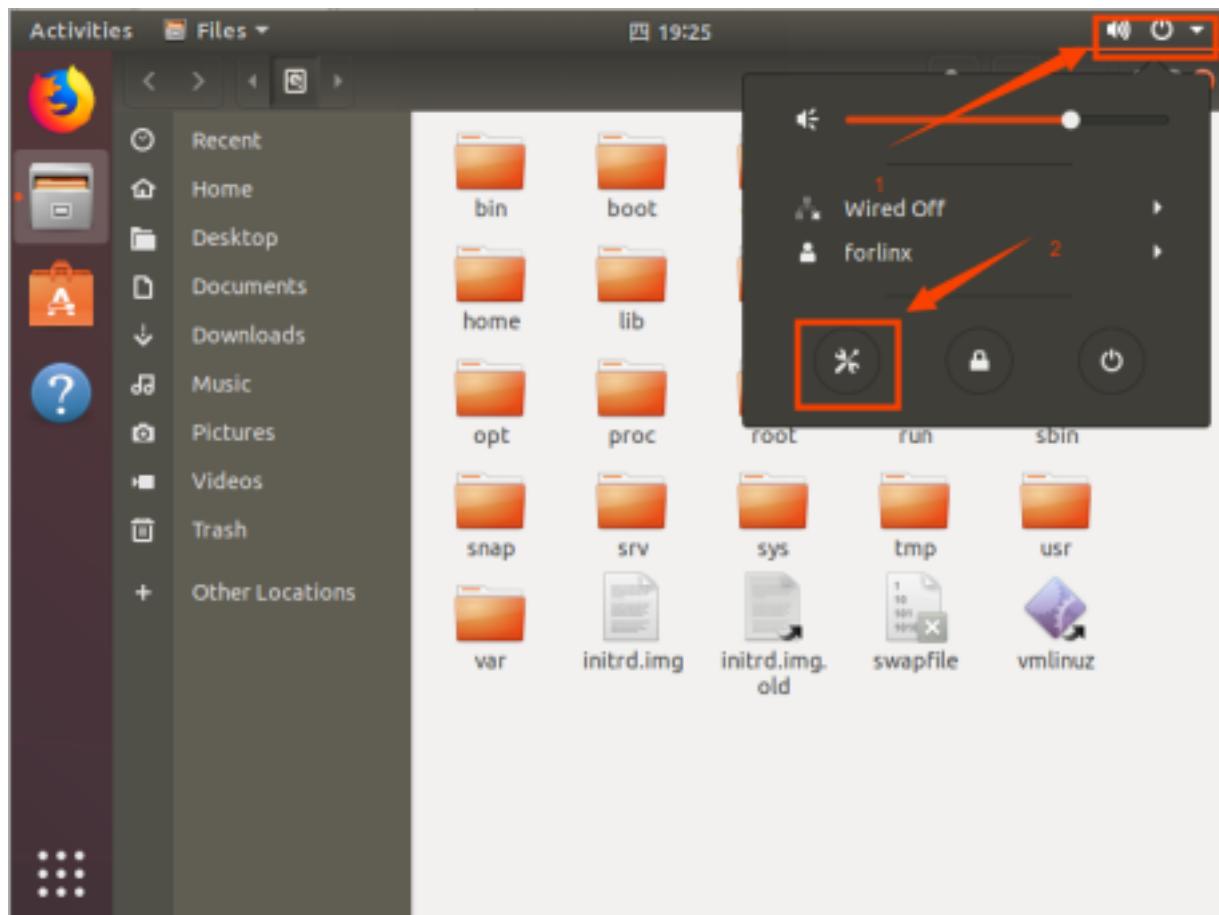
Once the VMware tools is complete, you can implement file copy and paste between Windows and Ubuntu.

The virtual machine is displayed full screen:

If the virtual machine is not able to be displayed in full screen, you can resolve this issue by clicking on “View” and selecting “Autofit Guest.” This will adjust the display to fit the screen automatically, enabling you to have a full-screen experience in the virtual machine.

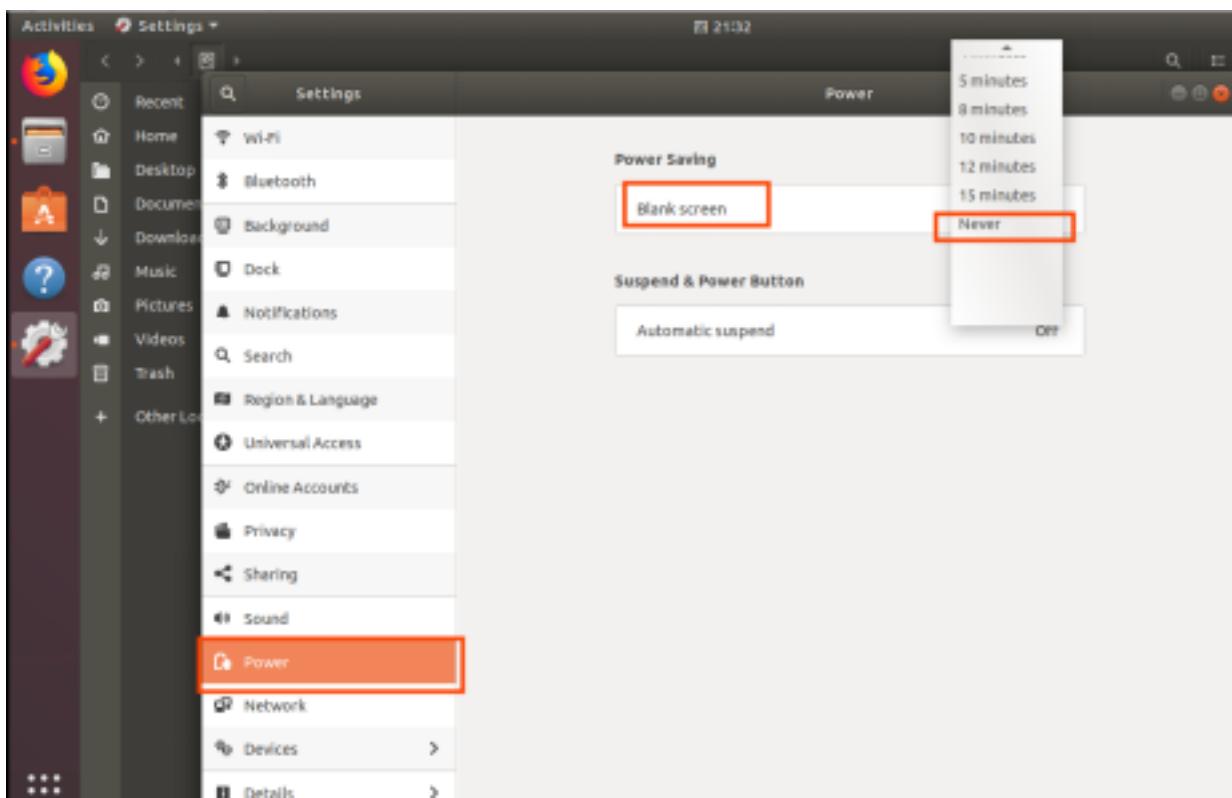


Make most of the system settings in the location shown. A lot of the setup requirements on Ubuntu can be done here.



Virtual machine hibernation settings:

Also, the default hibernation is 5min, if you don't want to set hibernation, just set it to Never by setting Power->Blank screen.



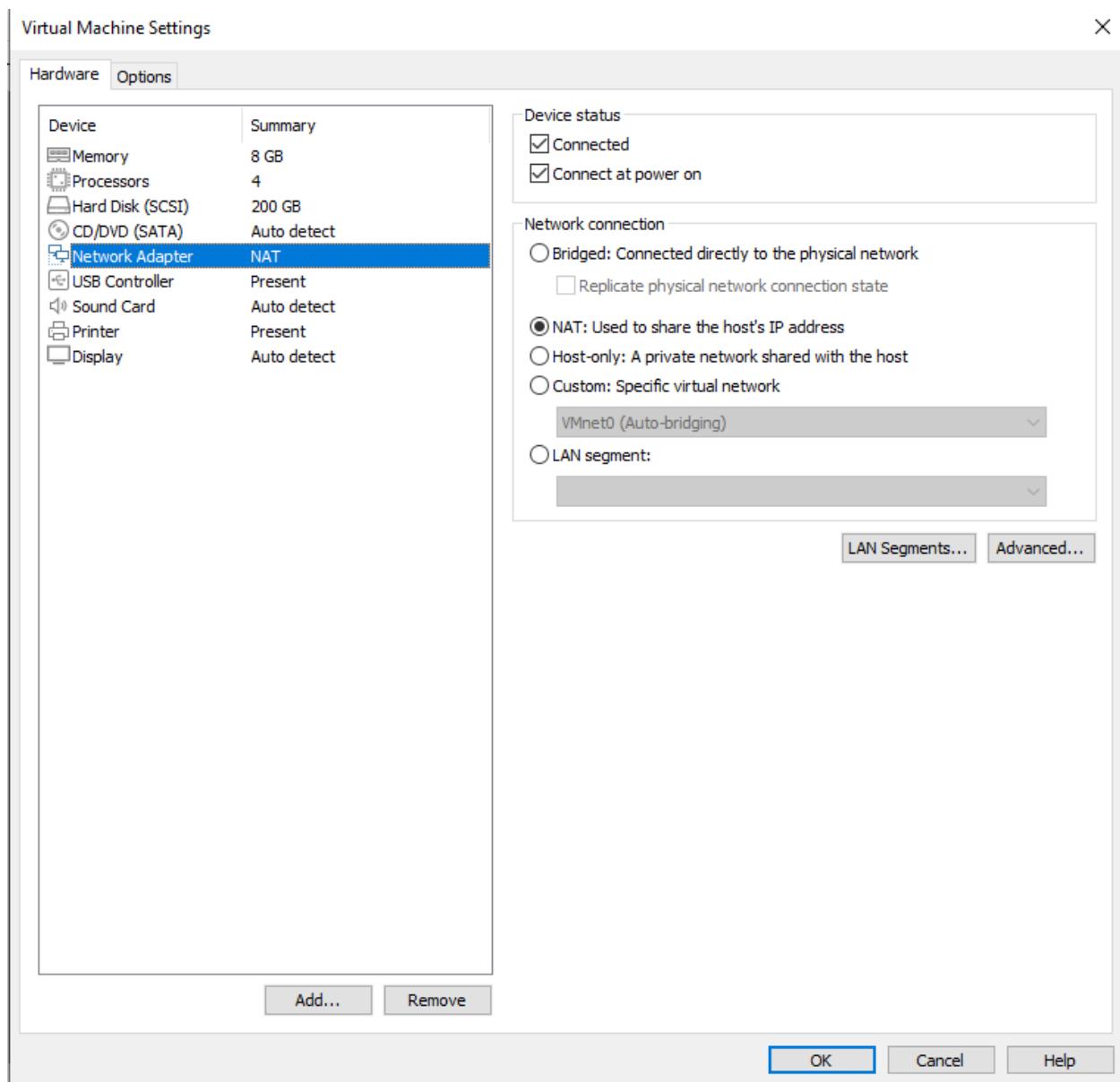
3.1.4 Network Settings for Virtual Machine

3.1.4.1 NAT Connection Method

By default, after the virtual machine is installed, the network connection method is set to NAT, which shares the host machine's IP address.

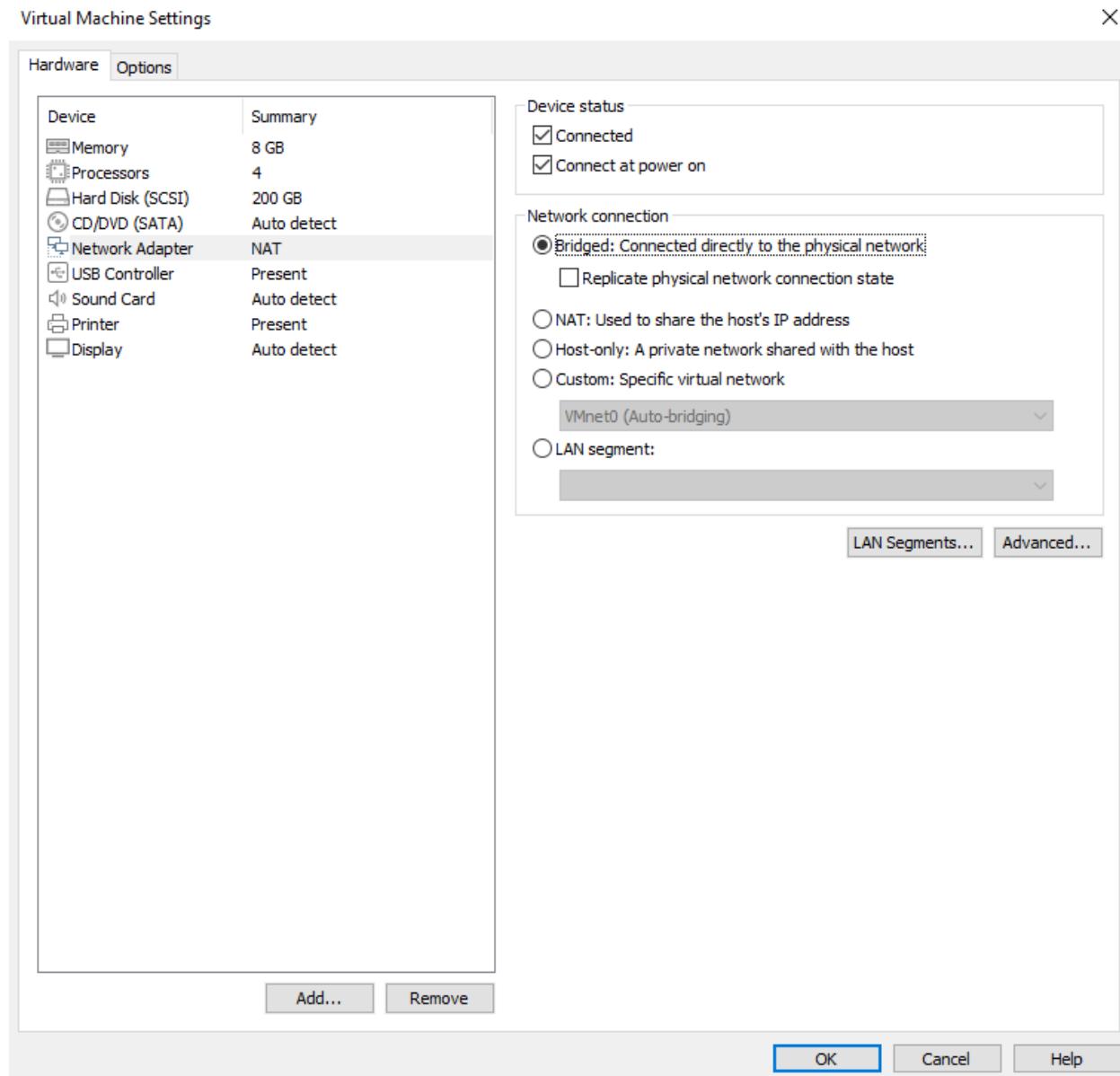
This configuration does not need to be changed when performing tasks like installing dependencies or compiling code.

When the VMware virtual NIC is set to NAT mode in a virtual machine, the network in the Ubuntu environment can be set to dynamic IP. In this mode the virtual NAT device and the host NIC are connected to communicate for Internet access. This is the most common way to access the external network.



3.1.4.2 Connections for Bridges

When the VMware virtual NIC device is in bridge mode, the host NIC and the virtual machine NIC communicate through the virtual bridge, and the network IP and the host need to be set in the same network segment in the Ubuntu environment. If accessing an external network, you need to set the DNS to be consistent with the host NIC. If TFTP, SFTP and other servers are used, the network contact mode of the virtual machine needs to be set as the bridge mode.



3.2 Libraries for Installing Linux Compilation System

Note: If you use the development environment provided by us, this section can be skipped directly.

Compiling for Linux requires the installation of a number of toolkits. Make sure that your computer or virtual machine can be connected to the Internet normally before the operation in this section. If the network is disconnected during the installation, please follow the following steps to install.

1. Install the necessary packages for compiling Linux.

```
forlinx@ubuntu:~$ sudo apt-get update
forlinx@ubuntu:~$ sudo apt-get install openssh-server vim git fakeroot      //Necessary
↳ toolkit installation
forlinx@ubuntu:~$ sudo apt-get install ssh make gcc libssl-dev liblz4-tool expect_
↳ expect-dev g++ patchelf chrpath gawk texinfo chrpath diffstat binfmt-support qemu-
↳ user-static live-build bison flex cmake gcc-multilib g++-multilib unzip device-tree-
(continues on next page)
```

(continued from previous page)

```
→compiler ncurses-dev libgucharmap-2-90-dev bzip2 expat gpgv2 cpp-aarch64-linux-gnu_
→libgmp-dev libmpc-dev bc python-is-python3 python2
```

2. The following libraries also need to be installed when using the Network Configuration Tool and menuconfig:

```
forlinx@ubuntu:~$ sudo apt-get update //Update the download_
→source information
forlinx@ubuntu:~$ sudo apt-get install libncurses* //For building text-
→based user interfaces
forlinx@ubuntu:~$ sudo apt-get install net-tools //Network_
→configuration tool
```

3.3 Cross-compilation Chain Installation

Use account information/2-Image and source code/2-Test procedure/aarch64-buildroot-linux-gnu_sdk-buildroot.tar.gz

Copy the above compressed package to the development environment/home/forlinx/, and decompress it in this directory:

```
forlinx@ubuntu:~$ tar -zvxf aarch64-buildroot-linux-gnu_sdk-buildroot.tar.gz
```

Go to the aarch64-buildroot-linux-gnu _ sdk -buildroot directory and execute relocate-sdk. Sh.

```
forlinx@ubuntu:~/aarch64-buildroot-linux-gnu_sdk-buildroot$ ./relocate-sdk.sh
```

3.4 Qt Creator Installation

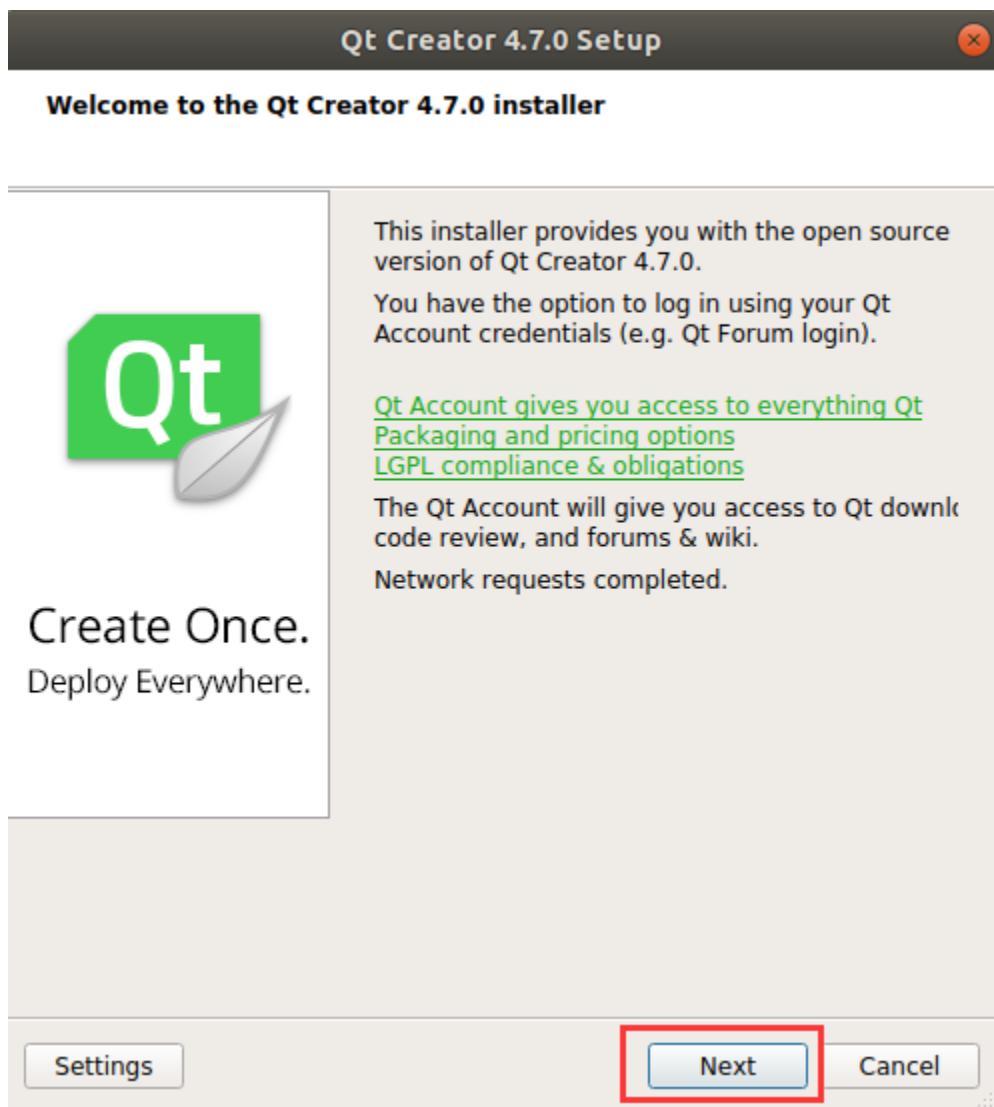
Copy qt-creator-opensource-linux-x86_64-4.1.0.run to any directory within the current user's home directory, and then run the following command.

Path: OK3588-C-C (Linux) user's profile\Linux\source code\qt-creator-opensource-linux-x86_64-4.7.0.run

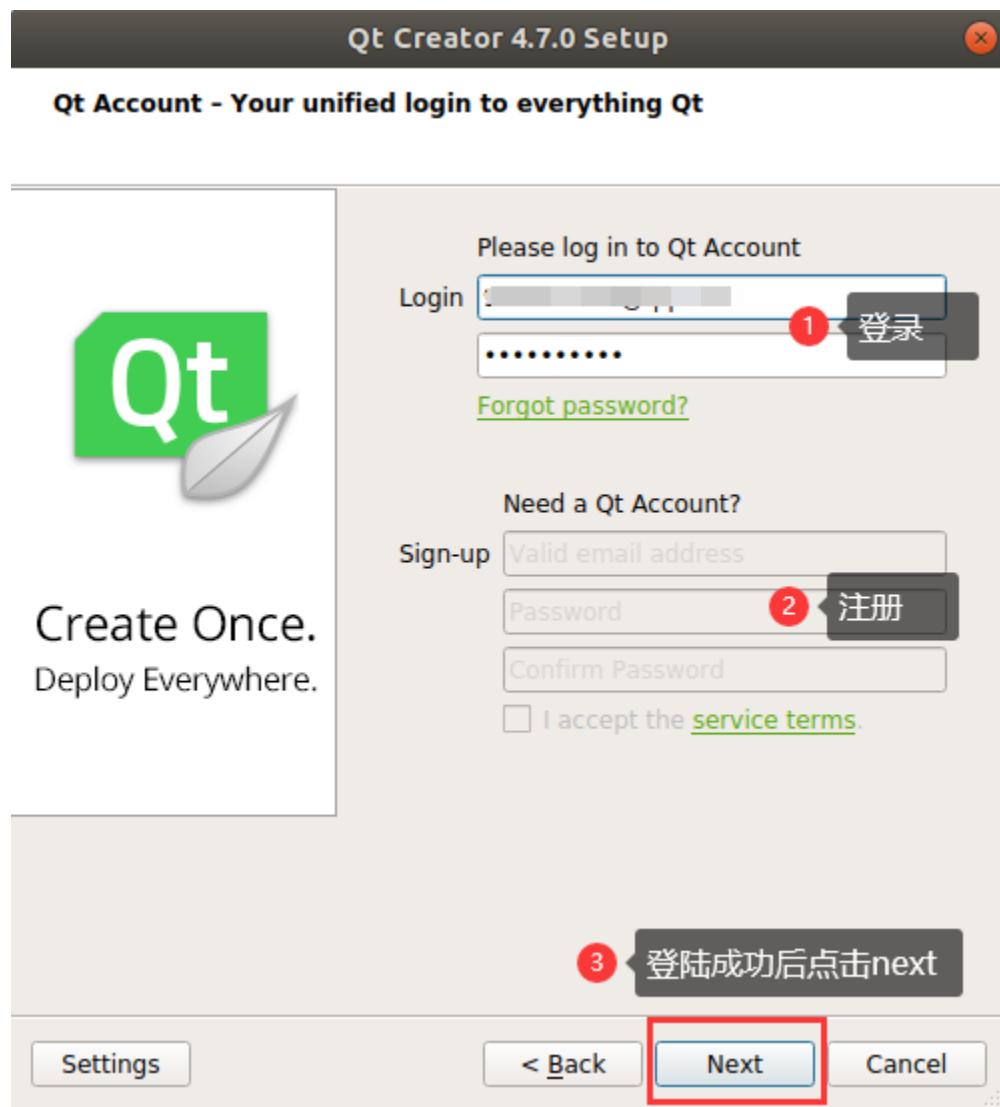
```
forlinx@ubuntu:~/3588$ ./qt-creator-opensource-linux-x86_64-4.7.0.run
```

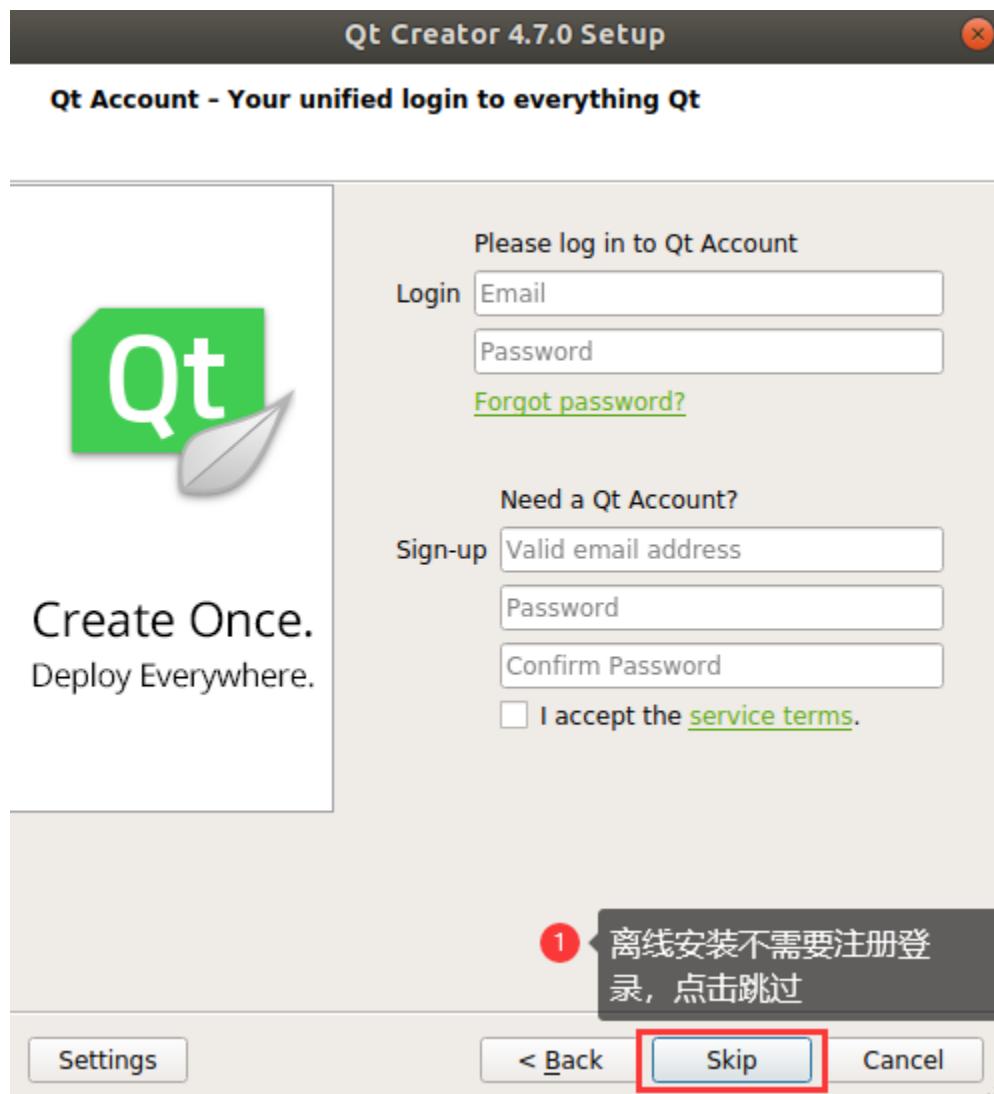
```
forlinx@ubuntu:~/3588$ ls qt-creator-opensource-linux-x86_64-4.7.0.run
qt-creator-opensource-linux-x86_64-4.7.0.run
forlinx@ubuntu:~/3588$
```

Then the installation window of the graphical interface will pop up, and install according to the instructions:

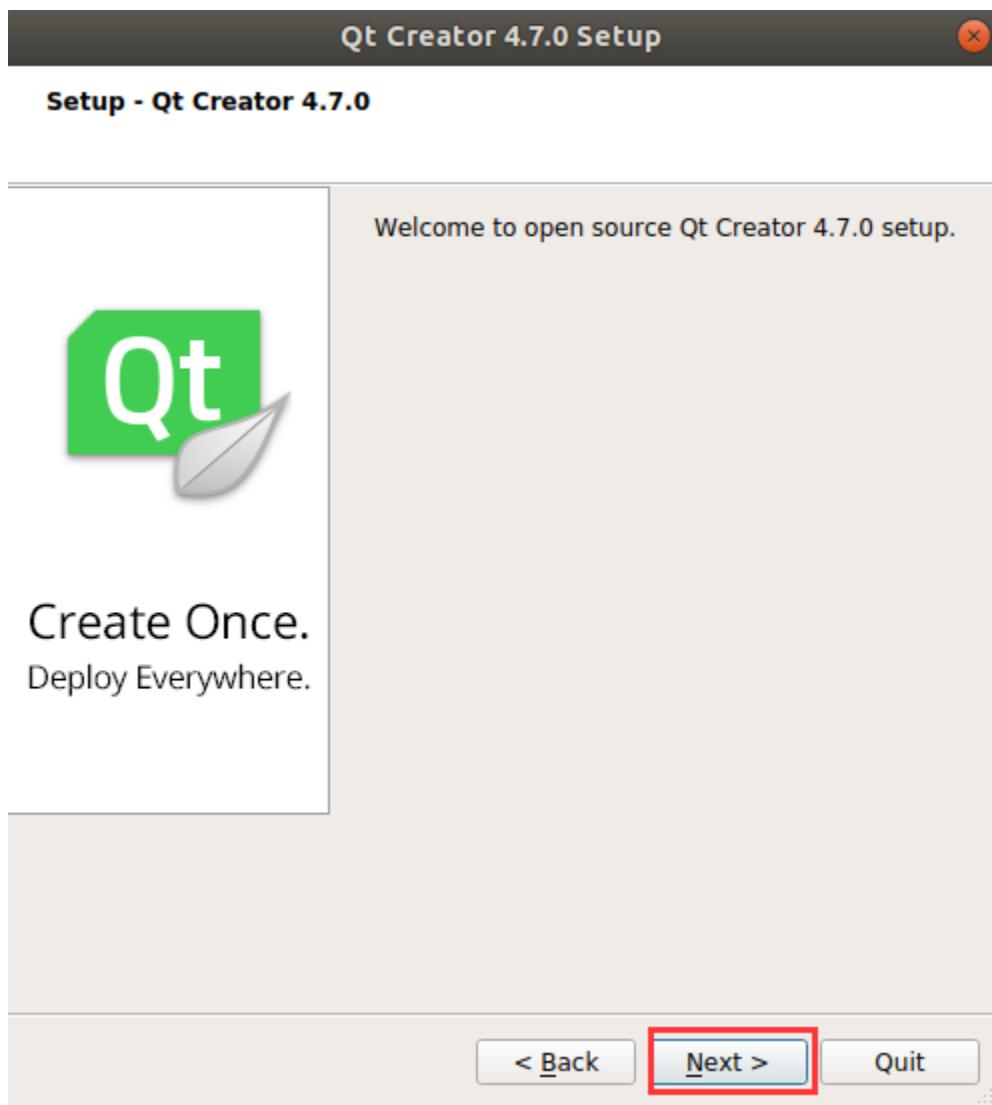




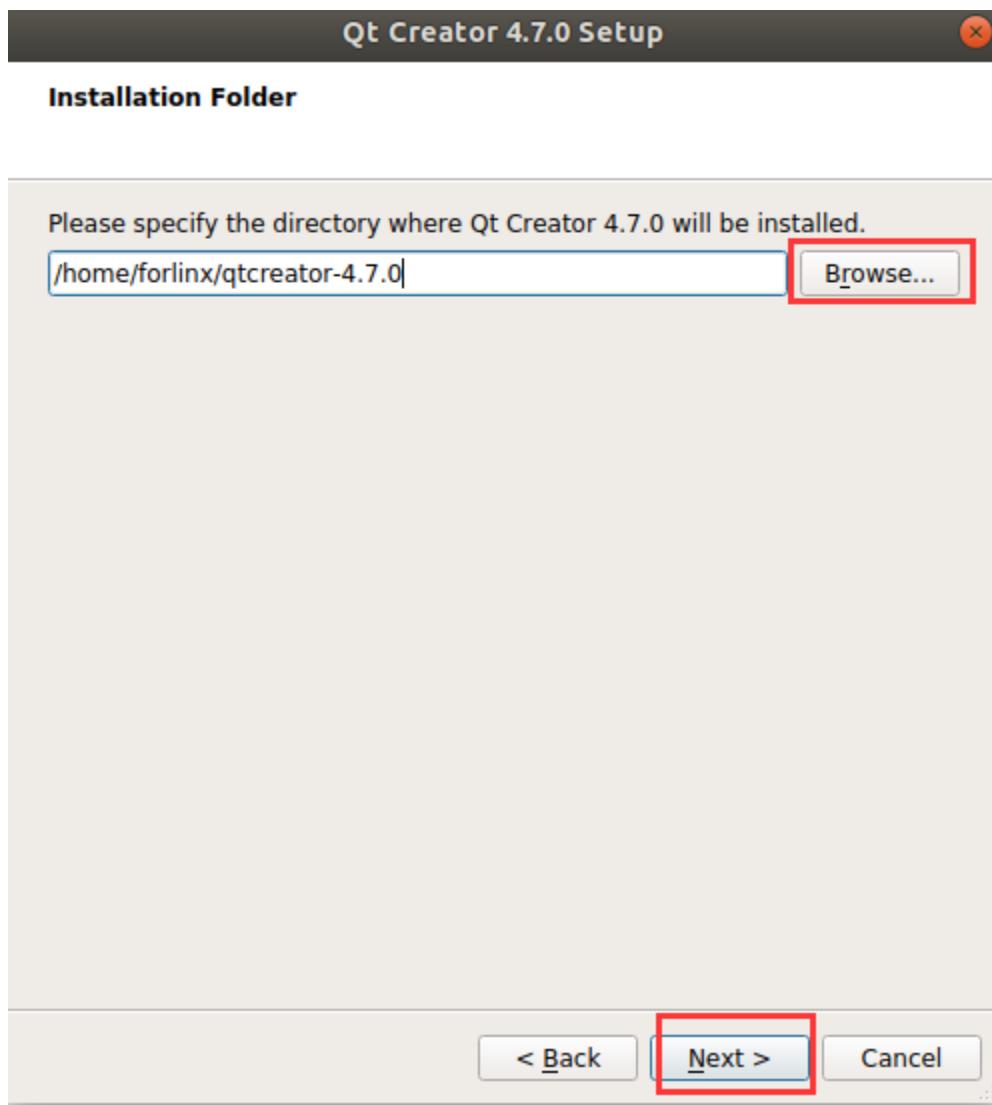




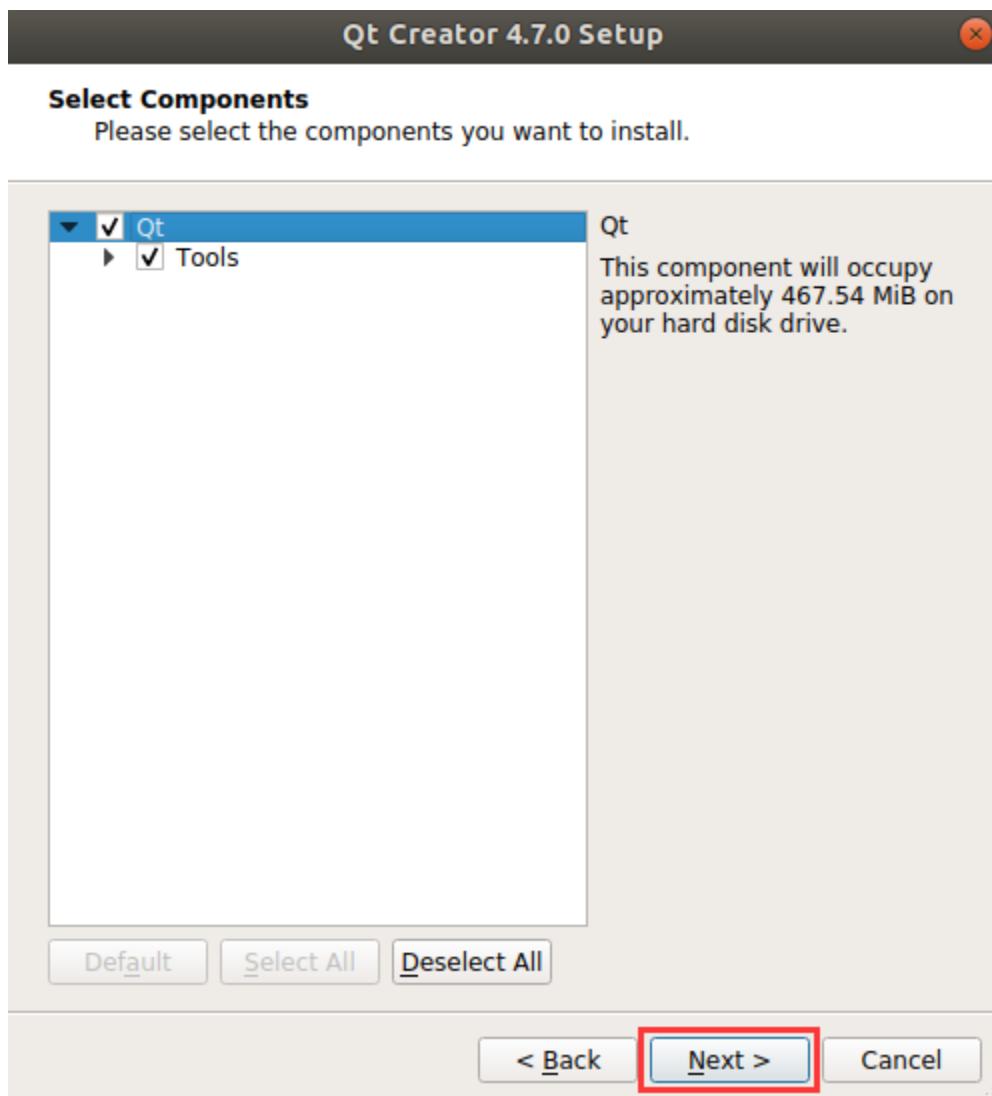
For online installation, register a Qt account (log in directly if you already have one). Password must include capital letters, uppercase, lowercase letters and numbers. After successful registration/login, click next. Skip this step for offline installation.



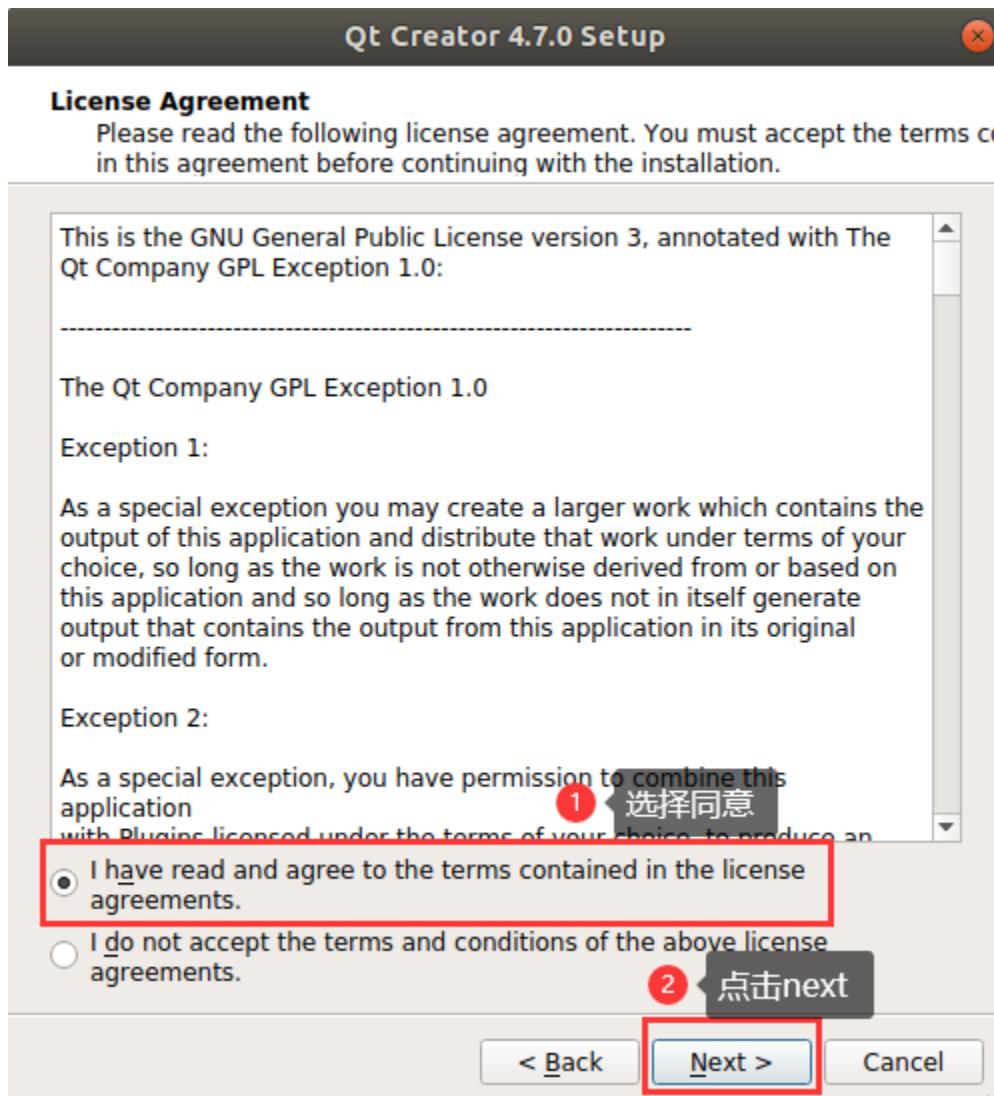
Click “Next” .

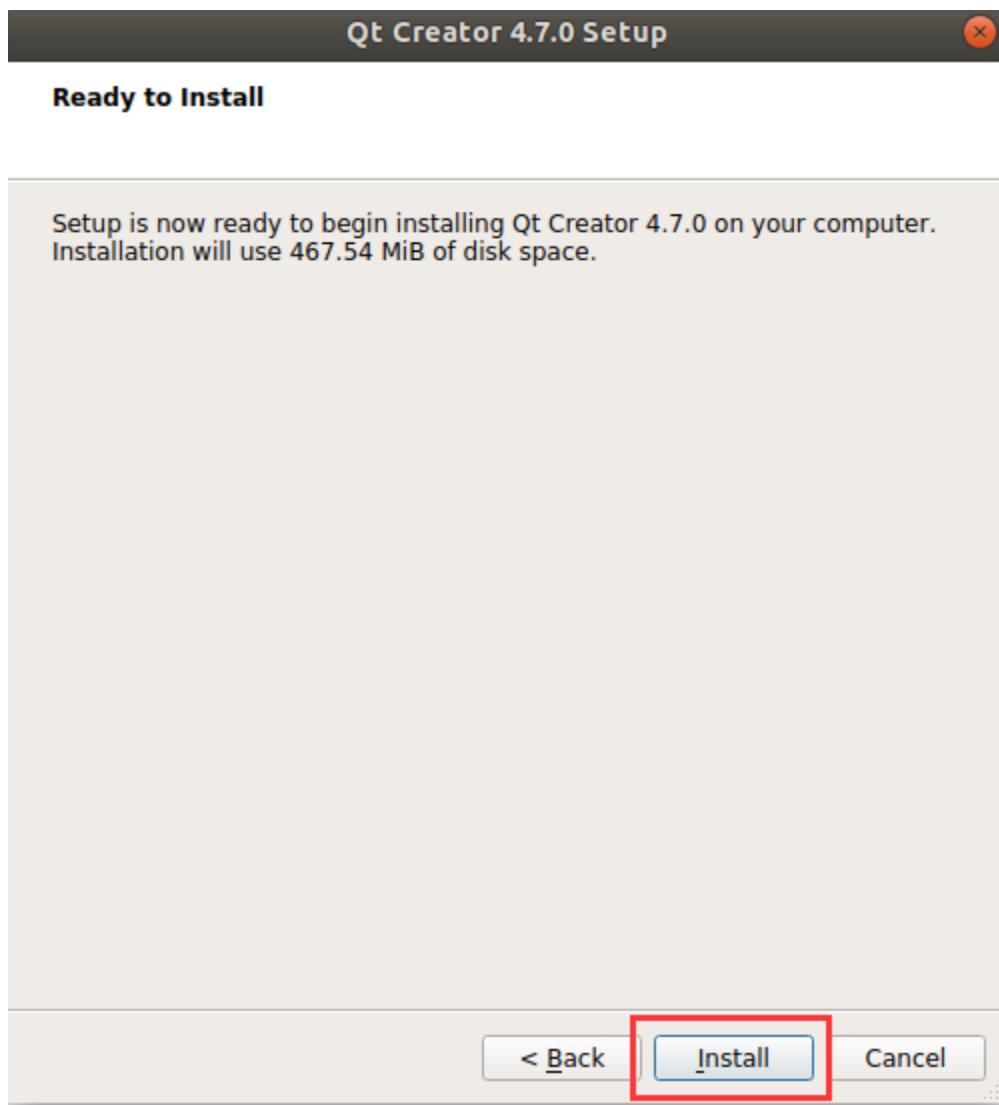


You can set the installation path according to your own habits. It is set by default here, so click “Next” .

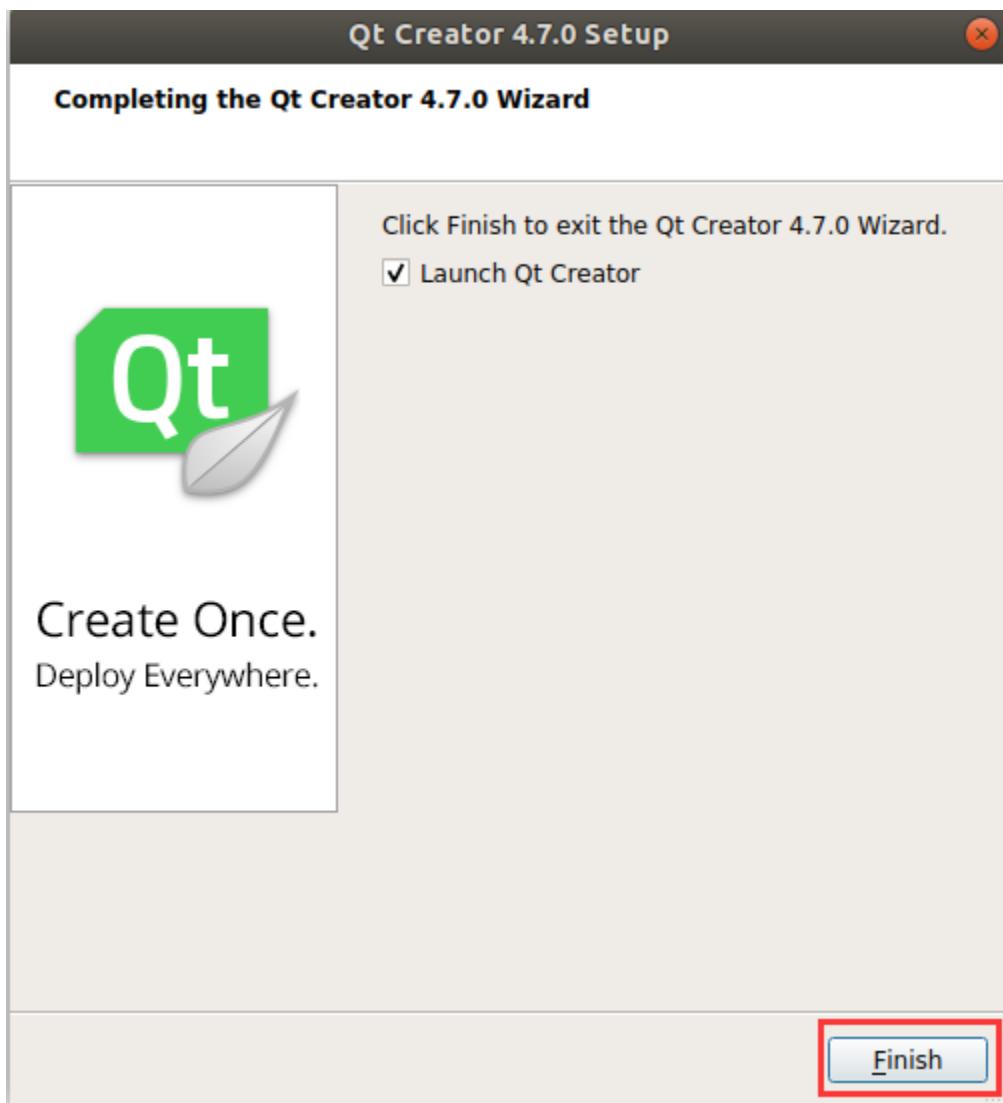


Installation completes, click “Next” .



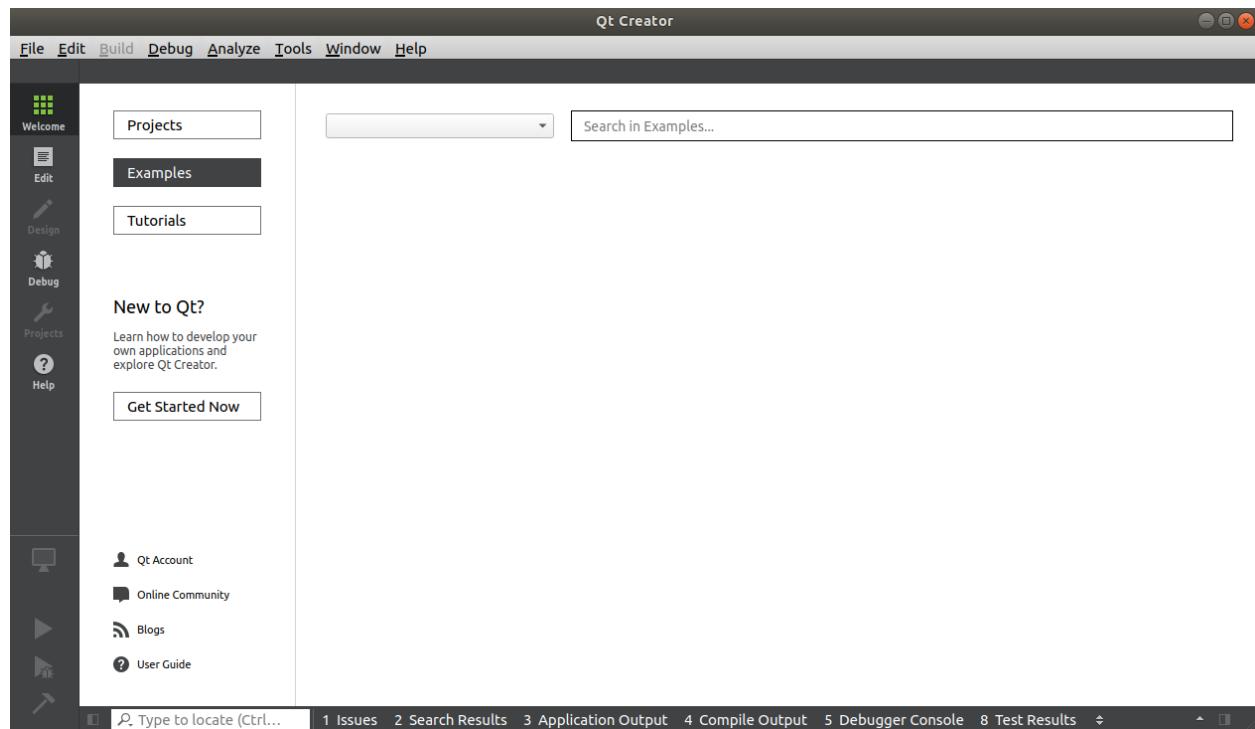


Click “Install” and wait for the installation to complete.



At this time, the Qt interface will be opened automatically. You can also start it through the command line. Execute the following command to open Qt Creator in the backstage. When you open it, the actual installation path shall prevail:

```
forlinx@ubuntu:~$ cd /home/forlinx/qtcreator-4.7.0/bin  
forlinx@ubuntu:~$ ./qtcreator &
```



The Qt Creator tool screen appears. Qt Creator is installed.

2.1.8 4. Related Code Compilation

Linux system is a desktop operating system built by Forlinx on the basis of buildroot.

This chapter mainly describes the compiling method of the source code related to the development board, including the kernel source code compilation and the application program compilation.

4.1 Preparation Before Compilation

4.1.1 Environment Description

- Development environment OS: Ubuntu22.04 64-bit version
- The board uses the Bootloader version: u-boot-2017.09
- Development Board Kernel: Linux-5.10.209
- Development board porting QT version: qt5.15.10

4.1.2 Source Code Copy

Source Code: User Information \ Linux \ Source Code \OK3588_Linux_fs.tar.bz2.0*

Create a working directory

```
forlinx@ubuntu:~$ cd /home/forlinx/
˓→3588                                         //Switch to_
˓→working directory
forlinx@ubuntu:~/3588$ cat OK3588-linux-source.tar.bz2.0* > OK3588-linux-source.tar.
˓→bz2
```

(continues on next page)

(continued from previous page)

```
forlinx@ubuntu:~/3588$ tar -xvf OK3588-linux-source.tar.bz2 //  
→Unzip the package in the current directory
```

Just run the command and wait for it to complete.

4.2 Source Code Compilation

Note:

- After the kernel source code is decompressed for the first time, the source code needs to be compiled as a whole;
- After compiling as a whole, you can compile separately according to the actual situation;
- The source code compilation requires a development environment with a running memory of 8G or above. Please do not modify the VM virtual machine image configuration provided by us.

4.2.1 Full Compilation Test

In the source code path, the compilation script build.sh is provided. Run the script to compile the entire source code. You need to switch to the decompressed source code path at the terminal and find the build.sh file.

```
forlinx@ubuntu:~$ cd /home/forlinx/3588/OK3588-linux-source  
forlinx@ubuntu: ~/3588/OK3588-linux-source$ rm output/defconfig
```

The following operations need to be operated under the source code directory, and the full compilation method is:

Full Compilation

```
forlinx@ubuntu: ~/3588/OK3588-linux-source$ ./build.sh
```

```
forlinx@ubuntu:~/3588/OK3588-linux-source$ ./build.sh  
Log colors: message notice warning error fatal  
  
Parsing supported commands...  
  
Log saved at /home/forlinx/3588/OK3588-linux-source/output/sessions/2024-09-06_13-46-51  
WARN: /home/forlinx/3588/OK3588-linux-source/output/defconfig not exists  
Pick a defconfig:  
  
1. OK3588_C_buildroot_defconfig  
2. OK3588_C_ubuntu22_defconfig  
Which would you like? [1]: 1
```

After successful compilation, the system image will be generated under the rockdev folder, as shown in the following figure:

```
forlinx@ubuntu:~/3588/OK3588-linux-source$ ls rockdev  
boot.img      misc.img  parameter.txt  rootfs.img  update.img  
MiniLoaderAll.bin  oem.img  recovery.img  uboot.img  userdata.img  
forlinx@ubuntu:~/3588/OK3588-linux-source$
```

Note: The update.img is packaged for full programming of OTG or TF card, and other files are programmed step by step.

4.2.2 Separate Compilation

Perform the operation in the kernel source code path.

```
forlinx@ubuntu: ~/3588/OK3588-linux-source$ ./build.sh kernel
```

```
INSTALL drivers/net/wireless/rockchip_wlan/rkwifi/bcmddhd/bcmddhd.ko
INSTALL drivers/net/wireless/rockchip_wlan/rkwifi/bcmddhd/dhd_static_buf.ko
DEPMOD 5.10.209
make: Leaving directory '/home/forlinx/3588/OK3588-linux-source/kernel'
Running 10-kernel.sh - build_kernel succeeded.
forlinx@ubuntu:~/3588/OK3588-linux-source$
```

The kernel in the update. img is not updated after successful compilation. Please flash the kernel/boot. img file step by step.

4.2.3 Clearing Compilation File

Perform the operation in the source code path.

```
forlinx@ubuntu: ~/3588/OK3588_Linux_fs$ sudo ./build.sh cleanall
```

```

forlinx@ubuntu:~/3588/OK3588-linux-source$ ./build.sh cleanall
Log colors: message notice warning error fatal

Log saved at /home/forlinx/3588/OK3588-linux-source/output/sessions/2024-09-07_16-43-15
Using last kernel version(5.10)
make: Entering directory '/home/forlinx/3588/OK3588-linux-source/kernel'
  CLEAN certs
  CLEAN drivers/firmware/efi/libstub
  CLEAN drivers/misc/lkdtm
  CLEAN drivers/scsi
  CLEAN drivers/tty/vt
fatal: not a git repository (or any of the parent directories): .git
  CLEAN fs/unicode
  CLEAN kernel
  CLEAN lib
  CLEAN net/wireless
  CLEAN usr
  CLEAN vmlinux.symvers modules-only.symvers modules.builtin modules.builtin.modinfo
  CLEAN scripts/basic
  CLEAN scripts/dtc
  CLEAN scripts/kconfig
  CLEAN scripts/mod
  CLEAN scripts
  CLEAN include/config include/generated .config .version Module.symvers
make: Leaving directory '/home/forlinx/3588/OK3588-linux-source/kernel'
make: Entering directory '/home/forlinx/3588/OK3588-linux-source/u-boot'
  CLEAN dts/.../arch/arm/dts
  CLEAN dts
  CLEAN examples/standalone
  CLEAN tools
  CLEAN tools/lib tools/common
  CLEAN spl/arch spl/board spl/cmd spl/common spl/disk spl/drivers spl/dts spl/env spl/fs spl/lib spl/u-boot-spl spl/u-boot-spl-dtb.bin spl/u-boot-spl-nodtb.bin spl/u-boot-spl.bin spl/u-boot-spl.dtb spl/u-boot-spl.lds spl/u-boot-spl.map spl/u-boot-spl.sym spl/u-boot.cfg tpl/arch tpl/board tpl/common tpl/disk tpl/drivers tpl/dts tpl/fs tpl/u-boot-spl.lds tpl/u-boot-tpl tpl/u-boot-nodtb.bin tpl/u-boot-tpl.bin tpl/u-boot-tpl.map tpl/u-boot-tpl.sym tpl/u-boot.cfg
  CLEAN u-boot u-boot-dtb.bin u-boot-nodtb.bin u-boot.bin u-boot.cfg u-boot.cfg.configs u-boot.dtb u-boot.lds u-boot.map u-boot.srec u-boot.sym System.map bl31_0x00040000.bin bl31_0x000f0000.bin bl31_0xffff100000.bin rk3588_spl_loader_v1.17.113.bin tee.bin u-boot-dtb.bin u-boot-nodtb.bin u-boot.bin uboot.img .cc
  CLEAN scripts/basic
  CLEAN scripts/dtc
  CLEAN scripts/kconfig
  CLEAN include/config include/generated spl tpl
  CLEAN .config include/autoconf.mk include/autoconf.mk.dep include/config.h
make: Leaving directory '/home/forlinx/3588/OK3588-linux-source/u-boot'
Running build.sh - cleanall succeeded.
forlinx@ubuntu:~/3588/OK3588-linux-source$
```

This operation clears all intermediate files. However, it does not affect the source files, including those that have already had changes made to them.

4.3 Image File Use

The update. img is packaged for full programming of OTG or TF card, and other files are programmed step by step. The Image file generated by separate compilation will not be updated in the update. img file, and it needs to be flashed by single-step. (see the software manual OTG flashing for details).

4.4 Qt Creator Environment Configuration

Qt is a cross-platform graphics development library, which supports many operating systems. Before compiling, you need to configure the compiling environment of Qt Creator.

4.4.1 Cross Compiler Configuration

Note: The cross-compilation chain has been installed in the default development environment. If you build your own development environment, you need to refer to 3.3 Cross-compilation Chain Installation (Default installation path: /home/forlinx/aarch64-buildroot-linux-gnu_sdk-buildroot).

Enter aarch64-buildroot-linux-gnu_sdk-buildroot to execute relocate-sdk.sh

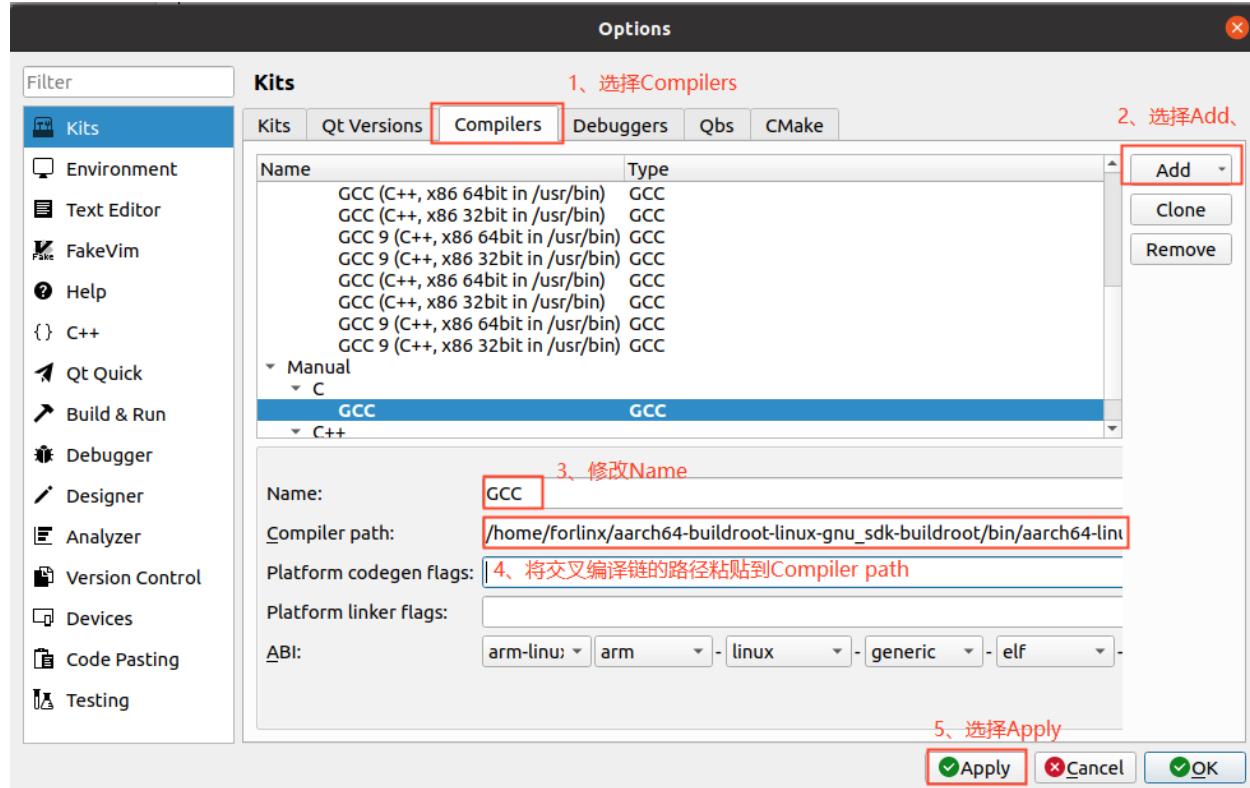
```
forlinx@ubuntu:~/aarch64-buildroot-linux-gnu_sdk-buildroot$ ./relocate-sdk.sh
```

1. Enter the installation path of qtcreator and open it;

```
forlinx@ubuntu: ~/qtcreator-4.7.0/bin$ ./qtcreator &
```

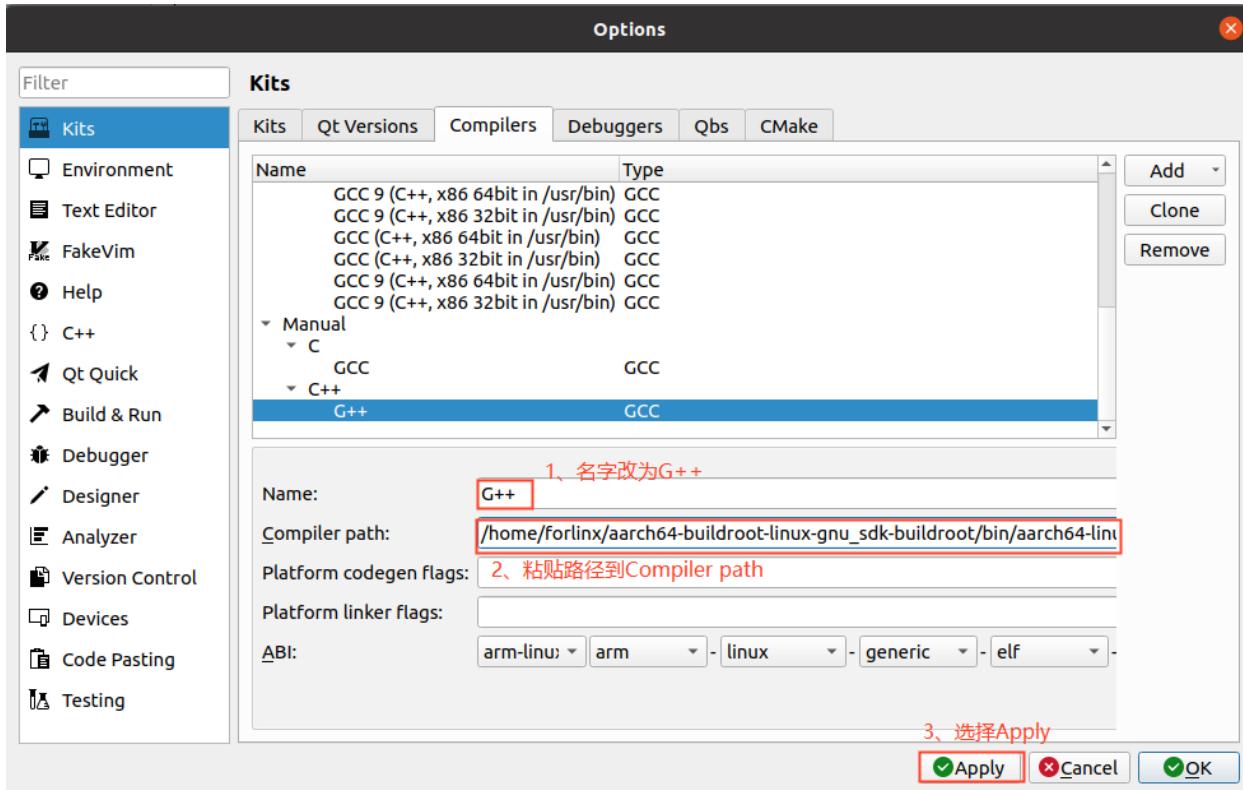
2. Click Tools-> Options-> Kits-> Compilers in Qt Creator, and then click Add-> GCC-> C;
3. Name enters GCC;
4. Paste the path of the compilation chain to the Compiler Path, as shown in the following figure:

Path: /home/forlinx/aarch64-buildroot-linux-gnu_sdk-buildroot/bin/aarch64-linux-gcc



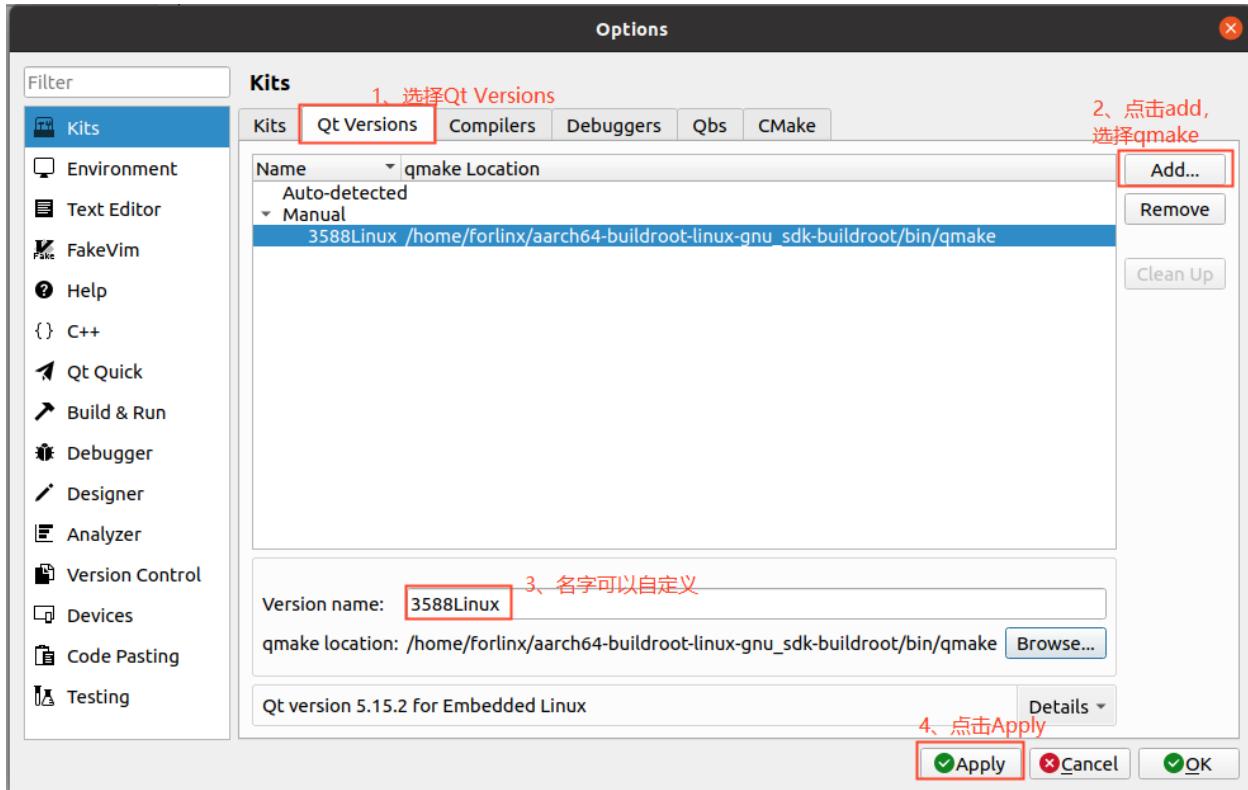
Follow the same method to add GCC compiler and click “Add->GCC->C” on the right side; as shown in the figure:

Path: /home/forlinx/aarch64-buildroot-linux-gnu_sdk-buildroot/bin/aarch64-linux-g++



4.4.2 Qt Versions Configuration

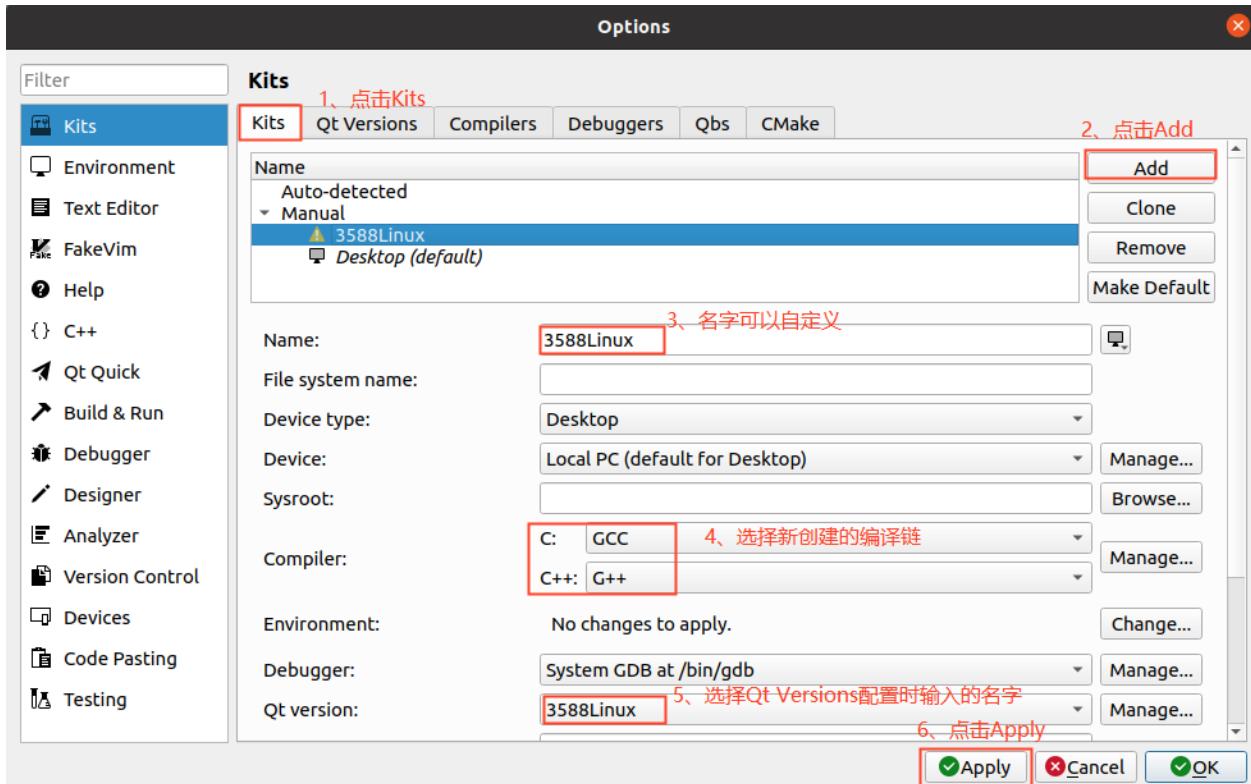
1. Click Tools -> Options -> Qt Versions in Qt Creator;
2. Then click Add to pop up a dialog box to select /home/forlinx/aarch64-buildroot-linux-gnu_sdk-buildroot/bin/qmake;
3. Click open to add;
4. Then it will return to the Qt Version configuration box, and the Version name can be changed by itself;
5. Then click Apply and OK.



4.4.3 Kits Configuration

Kits is a build kit for building and selecting development build environments useful for projects with multiple QT libraries. Add the previously added cross-compiler and QT Version to Kits to build a compilation environment suitable for the development board.

1. Click Tools->Options->Kits in Qt Creator, and then click Add to display the configuration section;
2. Name changes by itself;
3. Compiler selects GCC;
4. Qt version selects the name entered when the Qt version was created;
5. Then click Apply and OK.



4.5 Application Compilation and Operation

4.5.1 Command Line Application Compilation and Operation

This section uses the watchdog test program. By default, the source code is copied to the /home/forlinx/3588 directory.

1. Use the cd command to enter the test source code directory;

```
forlinx@ubuntu:~$ cd /home/forlinx/3588/OK3588_Linux_fs/app/forlinx/forlinx_cmd/
↳ fltest_watchdog
```

2. Add the cross-compiler path and use make to cross-compile;

```
forlinx@ubuntu: ~/3588/OK3588_Linux_fs/app/forlinx/forlinx_cmd/fltest_watchdog$_
↳ export PATH=/home/forlinx/aarch64-buildroot-linux-gnu_sdk-buildroot/bin/:$PATH
forlinx@ubuntu: ~/3588/OK3588_Linux_fs/app/forlinx/forlinx_cmd/fltest_watchdog$_
↳ aarch64-linux-gcc watchdog.c -o fltest_watchdog
generate fltest_watchdog success!!!
```

Use the file command to view the generated file information

```
forlinx@ubuntu:~/3588/OK3588_Linux_fs/app/forlinx/forlinx_cmd/fltest_watchdog$ file..
↳ fltest_watchdog
fltest_watchdog: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically
linked, interpreter /lib/ld-linux-aarch64.so.1, for GNU/Linux 3.7.0, not stripped
```

From the result, we can see that the compiled 64-bit ARM file.

3. Copy the fltest _ watchdog generated by compiling to the board through U disk or FTP, for example, under

the/forlinx path. Take the TF card as an example, and copy it to the development board and run the test;

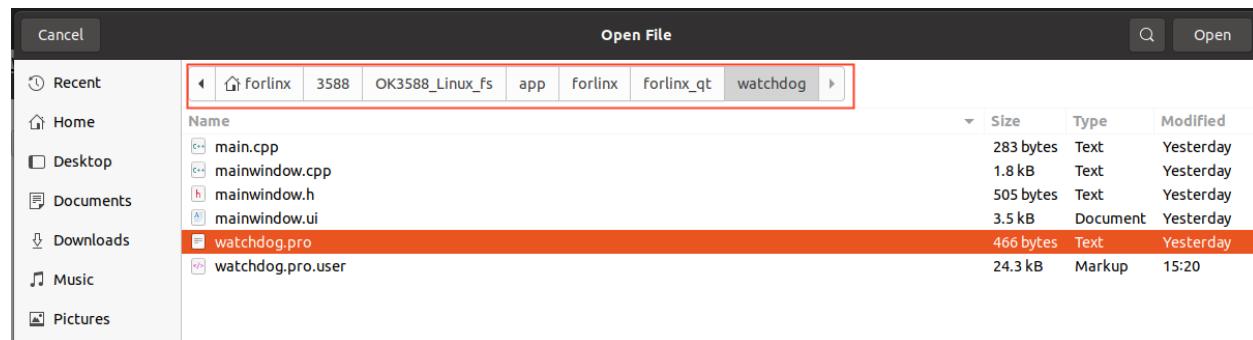
```
root@ok3588:/# cp /run/media/sda1/fltest_watchdog /home/forlinx/
root@ok3588:/# cd /home/forlinx/
root@ok3588:/home/forlinx# ./fltest_watchdog
Watchdog Ticking Away!
```

4. Refer to the chapter “Watchdog Test” in the user’s manual for the test.

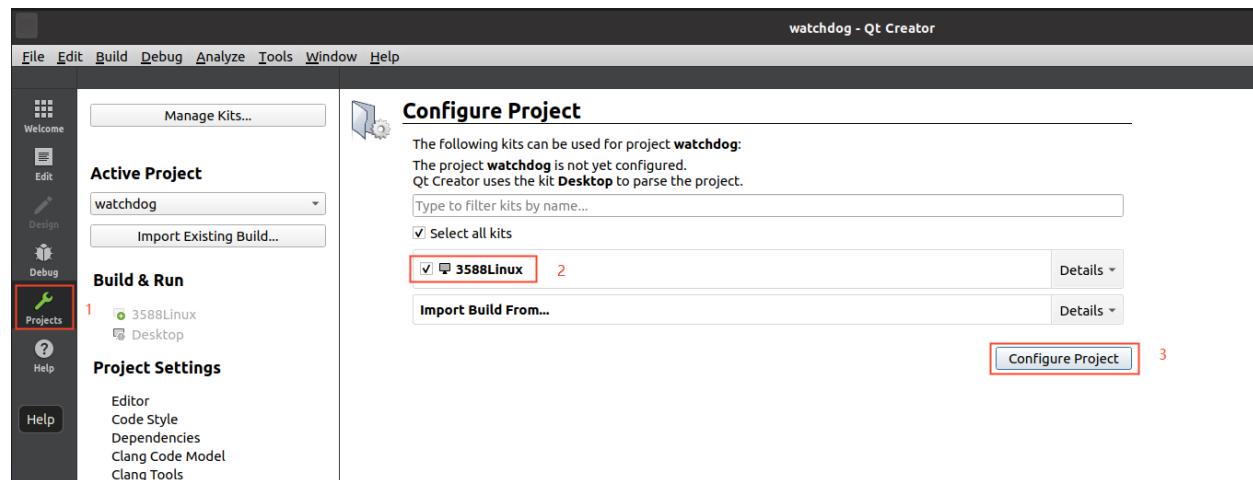
4.5.2 QT Application Compilation and Operation

Open Qt Creator in the development environment (the user opens it according to his actual path), click File-> Open File or Project of Qt Creator to pop up a window, and select/forlinx/3588/OK3588_Linux_fs/app/forlinx/forlinx_qt/watchdog/watchdog.pro

```
forlinx@ubuntu:~$ cd qtcreator-4.7.0/bin/
forlinx@ubuntu~/qtcreator-4.7.0/bin$ ./qtcreator &
```

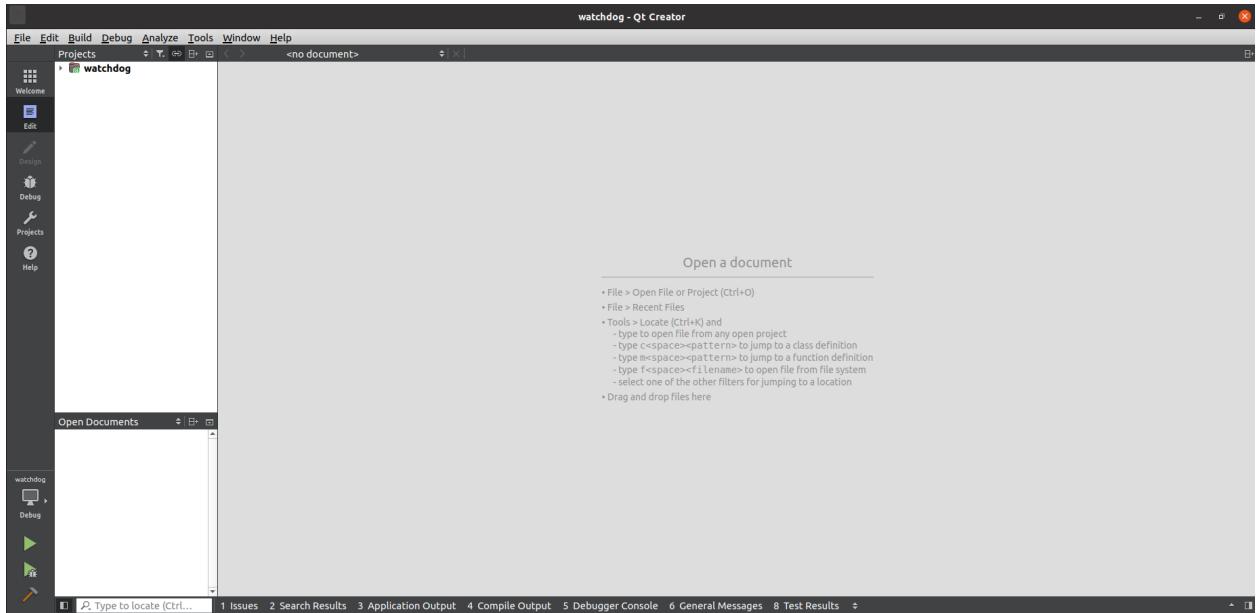


After opening the project, the interface is as follows: (If the page is not automatically changed, please select as shown in the screenshot).



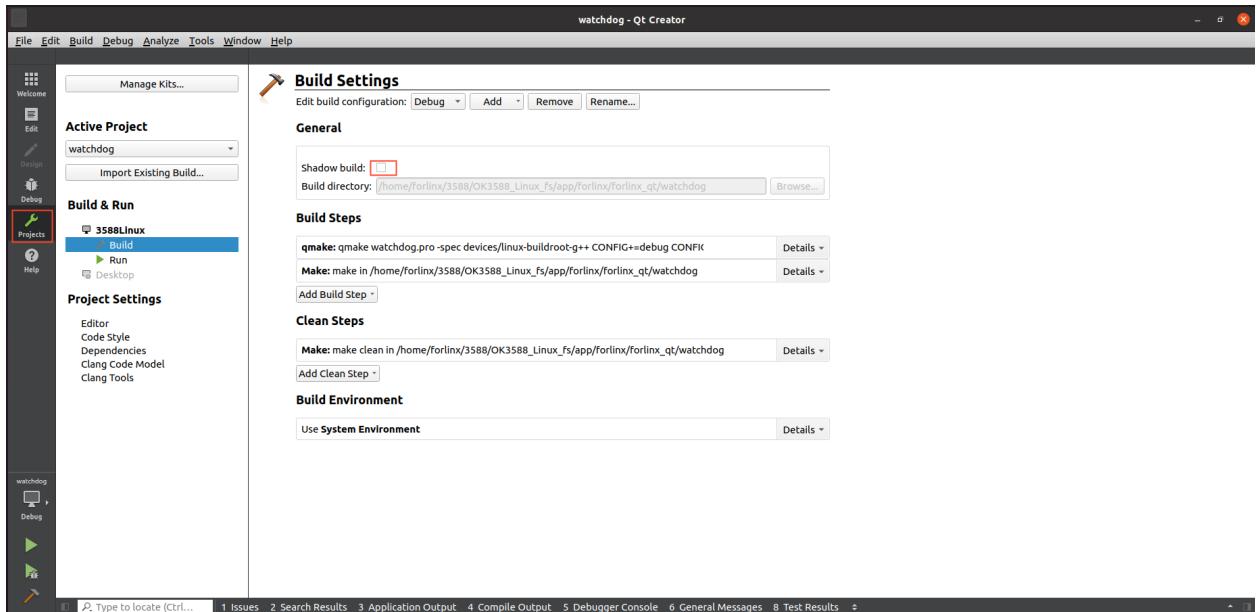
Click Configure Project to adapt to the build environment described in the “Qt Creator Environment Configuration” section of this manual.

After selection, the interface is as follows:

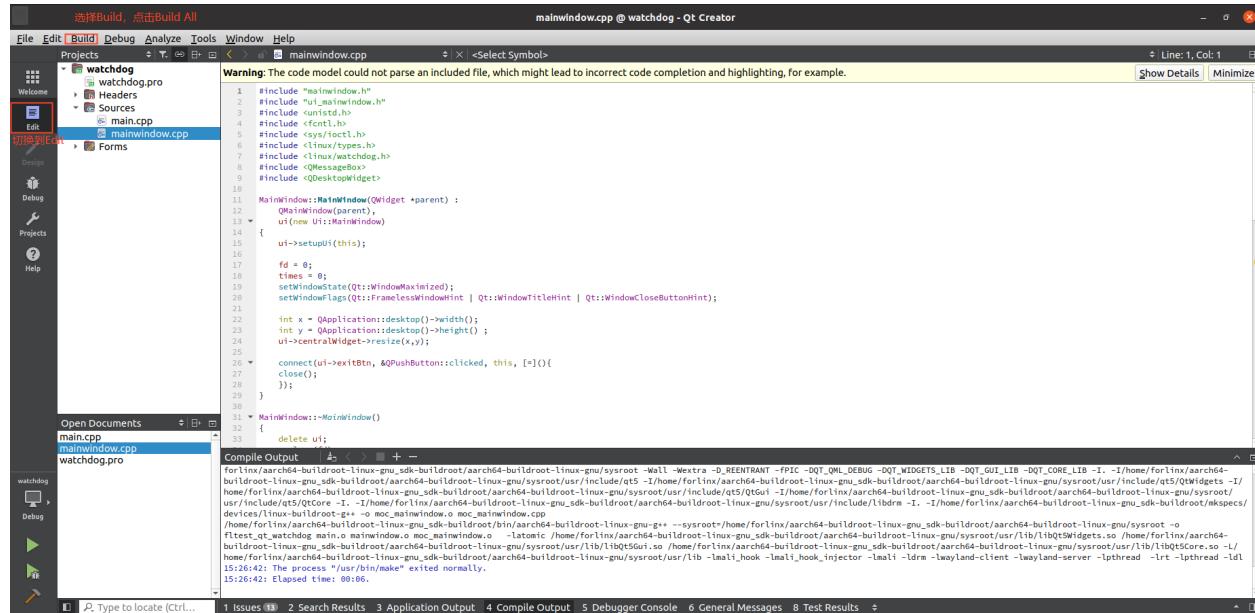


Click Build->Clean All to clear it. (If the intermediate file is not cleared, it can be deleted manually).

Click Projects to uncheck Shadow build.



Then click Build-> Build All to compile.



After the Build progress bar in the bottom right corner finishes running, it indicates that the compilation is complete. At this time, you will see the newly generated binary file fltest_qt_watchdog in the directory home/forlinx/3588/OK3588_Linux_fs/app/forlinx/forlinx_qt_out/, as follows:

```
forlinx@ubuntu:~/3588_u_r4/OK3588-Linux-source/app/forlinx/forlinx_qt_out/debug/bin$ ls -al
total 68
drwxrwxr-x 2 forlinx forlinx 4096 2月 24 10:13 .
drwxrwxr-x 3 forlinx forlinx 4096 2月 24 09:38 ..
-rwxrwxr-x 1 forlinx forlinx 59261 2月 24 10:13 fltest_qt_watchdog
```

Copy the executable file generated by compiling to the board through U disk or FTP, copy it to the development board, and run the test.

2.2 Forlinx Desktop22.04 User's Compilation Manual V1.0

Document classification: Top secret Secret Internal information [Open](#)

2.2.1 Copyright Notice

The copyright of this manual belongs to Baoding Folinx Embedded Technology Co., Ltd. Without the written permission of our company, no organizations or individuals have the right to copy, distribute, or reproduce any part of this manual in any form, and violators will be held legally responsible.

Forlinx adheres to copyrights of all graphics and texts used in all publications in original or license-free forms.

The drivers and utilities used for the components are subject to the copyrights of the respective manufacturers. The license conditions of the respective manufacturer are to be adhered to. Related license expenses for the operating system and applications should be calculated/declared separately by the related party or its representatives.

2.2.2 Overview

This manual is designed to enable users of the Forlinx Embedded development board to quickly understand the compilation process of the products and familiarize themselves with the compilation methods of Forlinx products. The application needs to be cross-compiled on an Ubuntu host before it can run on the development board. By following the methods provided in the compilation manual and performing practical operations, you will be able to successfully compile your own software code.

The manual will provide instructions for setting up the environment but there may be some unforeseen issues during the environment setup process. For beginners, it is recommended to use the pre-configured development environment provided by Forlinx. This will allow you to quickly get started and reduce development time.

Linux systems are typically installed in three ways: Dual system on a real machine, single system on a real machine, and virtual machine. Different installation methods have their advantages and disadvantages. This manual only provides methods to build ubuntu in a virtual machine.

Hardware Requirements: It is recommended to have at least 16GB memory or above. It allows for allocating a sufficient memory to the virtual machine (recommended to allocate 10GB or above), while still leaving enough resources for other operations on Windows. Insufficient memory allocation may result in slower performance on Windows.

The manual is mainly divided into four chapters:

- Chapter 1. Virtual Machine software installation - introduction to downloading and installing Vmware software;
- Chapter 2. provides the loading of the ubuntu system;
- Chapter 3. Building, setting up, and installing necessary tools for the Ubuntu system and common issues in development environments;
- Chapter 4. Compiling the kernel and Linux-related source code.

A description of some of the symbols and formats associated with this manual:

Format	Meaning
Note	Note or information that requires special attention, be sure to read carefully
?	Relevant notes on the test chapters
?	Indicates the related path.
Blue font on gray background	Refers to commands entered at the command line(Manual input required).
Black font	Serial port output message after entering a command
Bold black	Key information in the serial port output message
//	Interpretation of input instructions or output information
User-name@Hostname	forlinx @ ubuntu: Development environment ubuntu account information, which can be used to determine the environment in which the function operates.

After packaging the file system, you can use the “ls” command to view the generated files.

```
forlinx@ubuntu:~/3588$ ls
                   //List the files in this
                   ↵directory
OK3588-linux-source  OK3588-linux-source.tar.bz2
```

- forlinx@ubuntu: the username is forlinx and the hostname is ubuntu, indicating that the operation is performed in the development environment ubuntu.
- //: Explanation of the instruction, no input required.
- ls: Blue font, indicating the relevant commands that need to be manually entered.
- **OK3588-linux-source:** Black font is the output information after entering the command; bold font is the key information; here is the packaged file system.

2.2.3 Application Scope

This manual is mainly applicable to the Forlinx Desktop22.04 operating system on the Forlinx OK3588-C platform. Other platforms can also refer to it, but there will be differences between different platforms. Please make modifications according to the actual conditions.

2.2.4 Revision History

Date	User Manual Version	SoM Version	Carrier Board Version	Revision History
04/11/20	V1.0	V1.1	V1.1 and Above	OK3588-C Forlinx Desktop22.04 User's Compilation Manual Initial Version

2.2.5 1. VMware Virtual Machine Software Installation

This chapter mainly introduces the installation of VMware virtual machines, using VMware Workstation 15 Pro v15.1.0as an example to demonstrate the installation and configuration process of the operating system.

1.1 VMware Software Downloads and Purchase

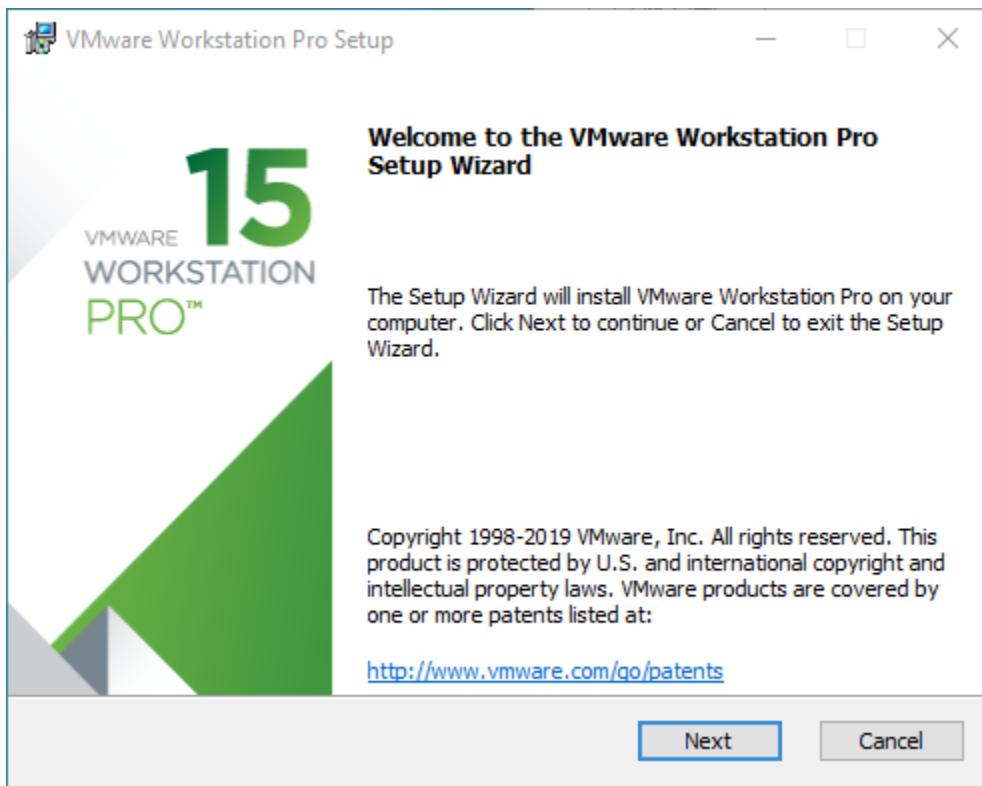
Go to the VMware website <https://www.vmware.com/cn.html> to download Workstation Pro and get the product key. VMware is a paid software that requires purchasing, or you can choose to use a trial version.

The screenshot shows the official VMware website homepage. At the top, there is a navigation bar with links for "United States (English)", "Shop VMware", and "My Account". Below the navigation bar is a search bar and a "GET STARTED" button. The main content area is organized into several sections: "Multi-Cloud Services" (with links to Tools & Training, VMware Customer Connect, VMware Trust Center, Learning & Certification, Tech Zone, Product Downloads, Product Trials, Cloud Services Engagement Platform, Hands-on Labs, Services, Professional Services, and Customer Success); "Products" (with links to Support, VMware Customer Connect, Support Offerings, Skyline, Support Customer Welcome Center, Marketplace, Cloud Marketplace, VMware Marketplace, Videos, VMware Video Library, VMware Explore Video Library, and VMware Explore); "Solutions" (with links to Blogs & Communities, Blogs, Podcasts, News & Stories, Communities, Customers, Customer Stories, Events, VMware Explore, SpringOne, and All Events & Webcasts); and "Partners" (with links to Resources, Search, and a "SEE ALL RESOURCES" button). A "GET STARTED" button is located at the bottom right of the main content area.

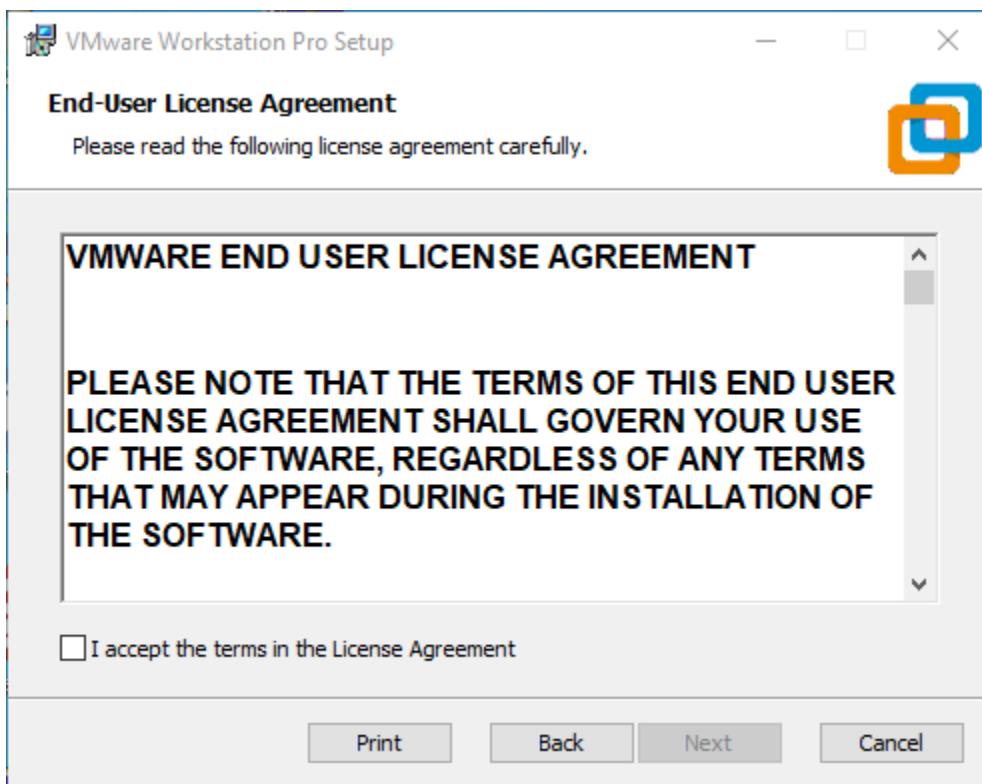
After the download is complete, double-click the startup file to start the installer.

1.2 VMware Software Installation

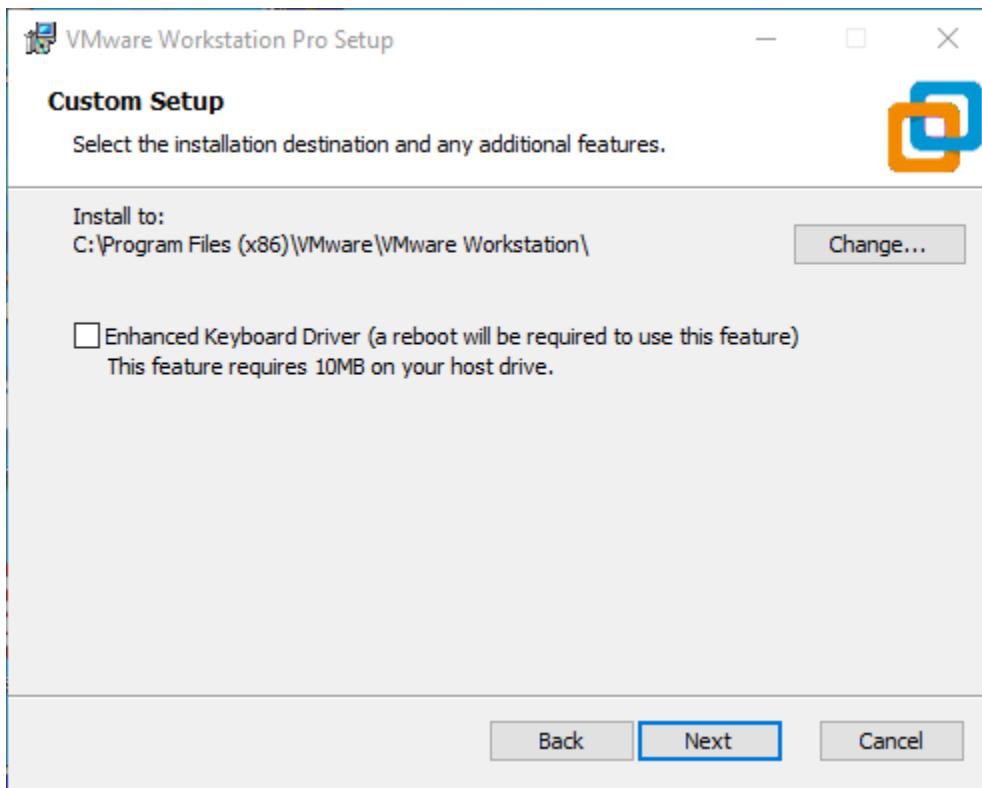
Double-click the startup program to enter the installation wizard, and click on “Next” .



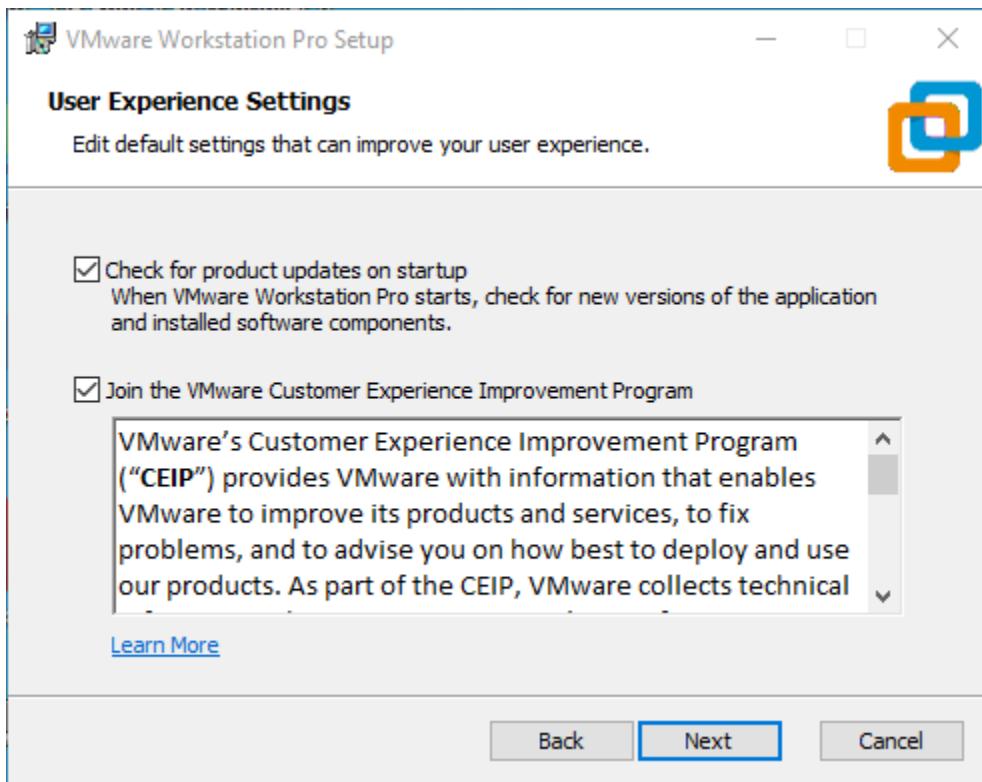
Check I accept the terms in the license agreement and click Next.



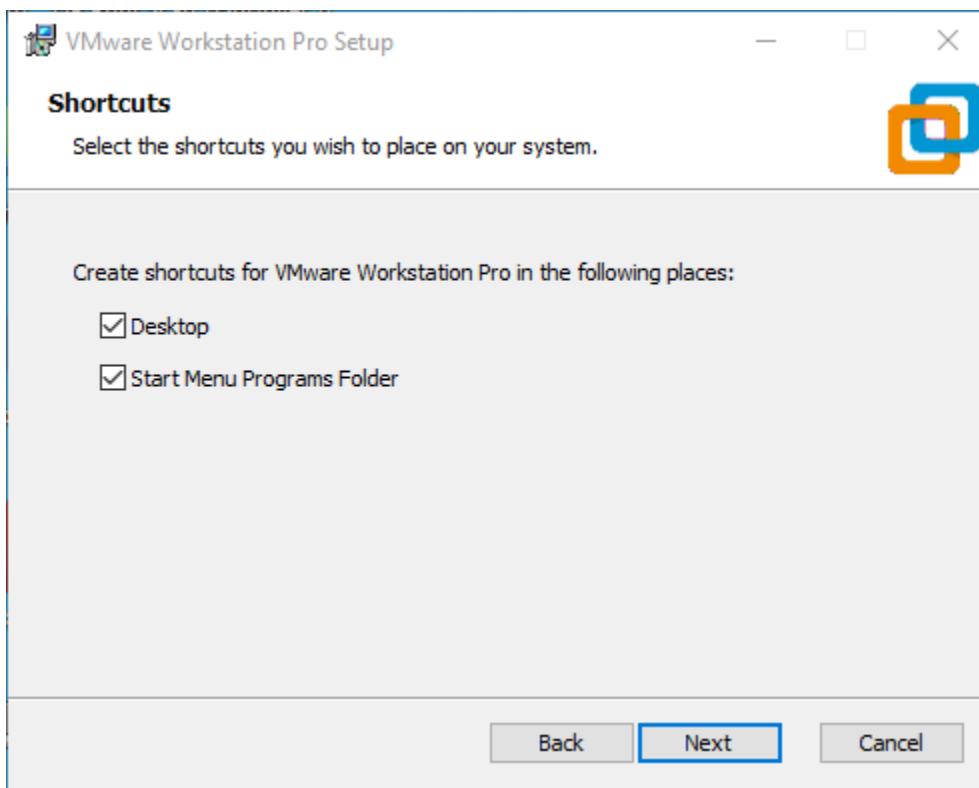
Modify the installation location to the partition of your computer where the software is installed, and click "Next" .



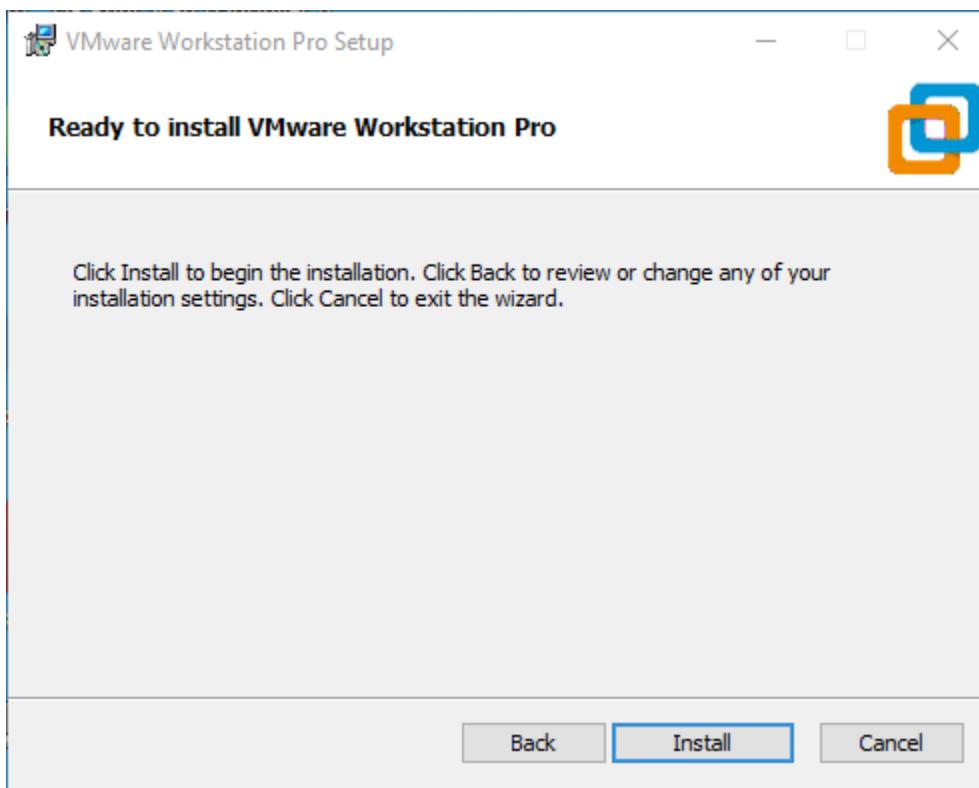
Uncheck and click on “Next” .



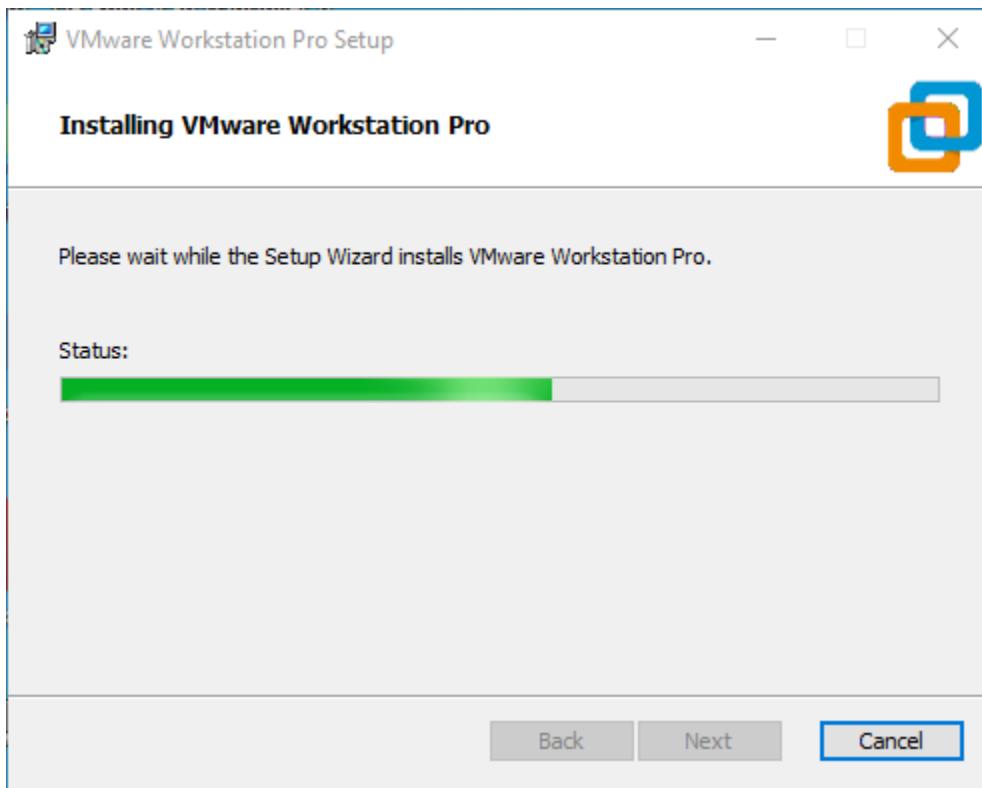
Check Add Shortcut and click “Next” .



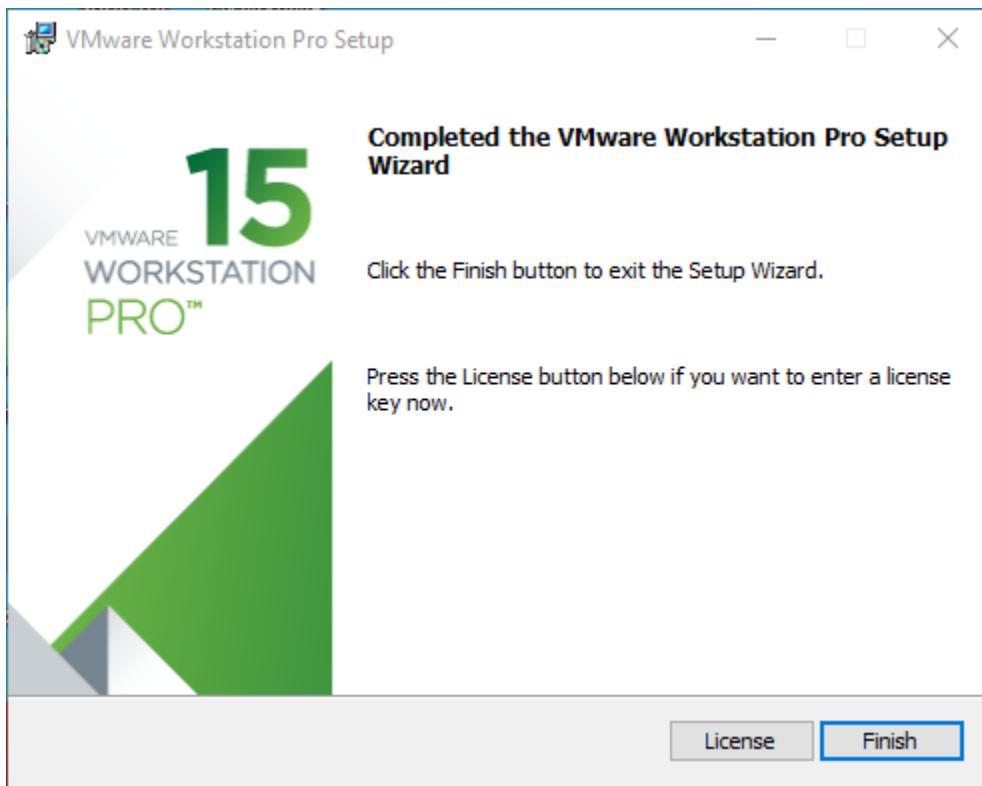
Click “Installation”



Wait for the installation to complete.



Click “Finish” to try it out. If users need to use it for a long time, they need to buy it from the official and fill in the license.



2.2.6 2. Loading the Existing Ubuntu Development Environment

Note:

- It is recommended for beginners to directly use the pre-built virtual machine environment provided by Forlinx, which already includes installed cross-compiler and Qt environment. After understanding this chapter, you can directly jump to the compilation chapter for further study;
- The development environment provided for general users is: forlinx (username), forlinx (password);
- Please ask your sales representative for the download link.

There are two ways to use a virtual machine environment in VMware: one is to directly load an existing environment, and the other is to create a new environment. First talk about how to load an existing environment.

First, download the development environment provided by Forlinx. In the development environment documentation, there should be an MD5 checksum file. After downloading the development environment, you should verify the integrity of the compressed package using the MD5 checksum. (You can use an on-line MD5 checksum tool or download a specific MD5 checksum tool for this purpose). To check if the checksum in the verification file matches the checksum of the file itself. If they match, the file download is successful. If they don't match, it suggests that the file may be corrupt, and you should consider downloading it again.

名称	修改日期	类型	大小
checksums.md5	2023/2/6 15:43	文件校验信息	1 KB
OK3588-VM15.1.0-ubuntu20.04.part1.rar	2023/2/6 13:45	WinRAR 压缩文...	1,945,600...
OK3588-VM15.1.0-ubuntu20.04.part2.rar	2023/2/6 13:47	WinRAR 压缩文...	1,786,526...

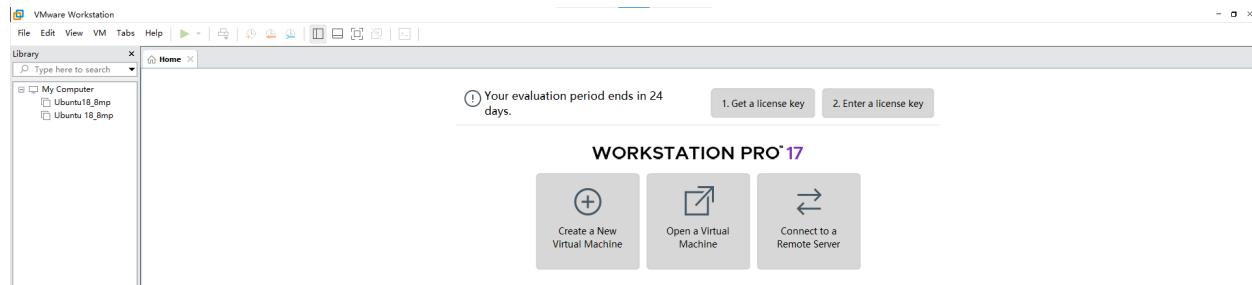
Select all compressed files, right-click and extract to the current folder or your own directory:

名称	修改日期	类型	大小
3588.zip.001	2024/9/9 11:12	001 文件	3,145,728...
3588.zip.002	2024/9/9 11:16	002 文件	3,145,728...
3588.zip.003	2024/9/9 11:16	003 文件	892,327 KB
md5.txt	2024/9/9 11:33	文本文档	1 KB
readme.txt	2024/9/9 11:35	文本文档	1 KB

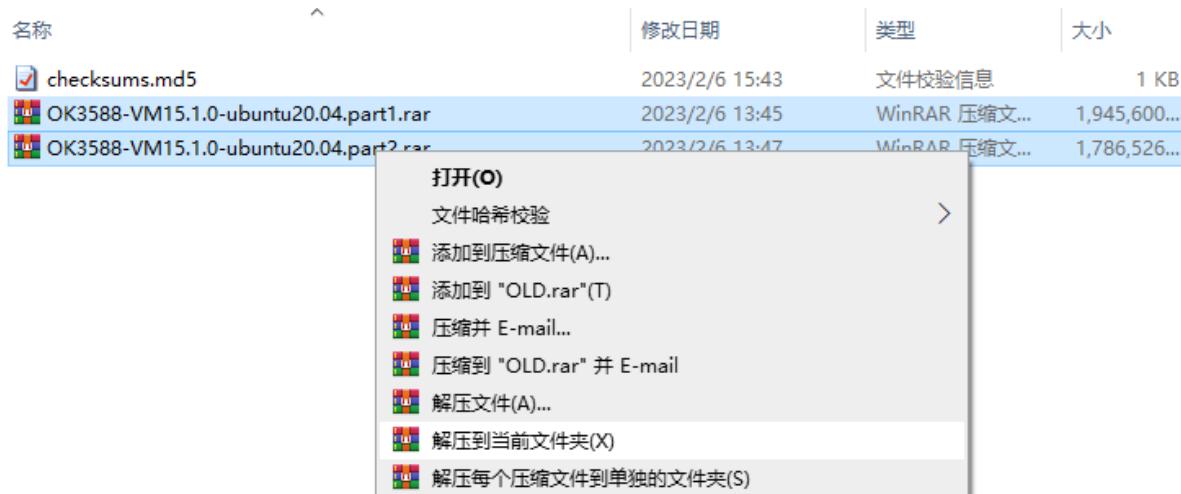
After the extraction is complete, you will obtain the development environment OK3588-VM15.1.0-ubuntu20.04.

The file “3588 development environment.vmx” in the OK3588-VM15.1.0-ubuntu20.04 folder is the file that you need to open to access the virtual machine.

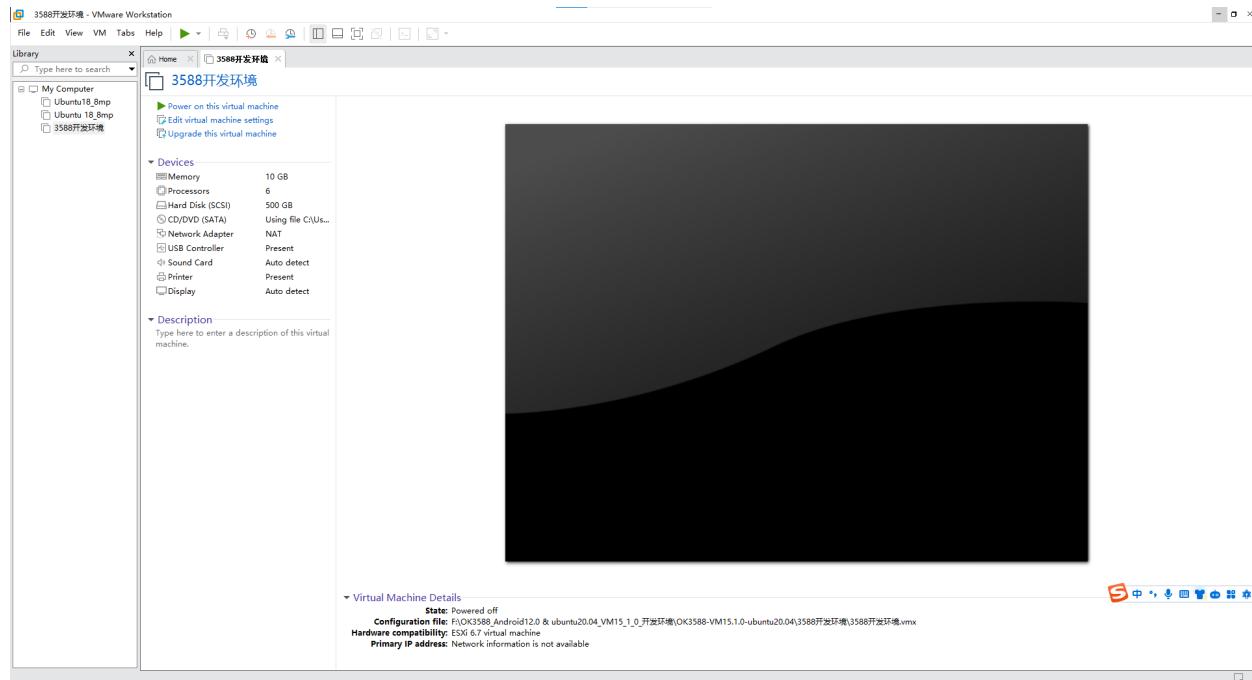
Open the installed virtual machine.



Navigate to the directory where the recently extracted OK3588-VM15.1.0-ubuntu20.04 virtual machine file is located, and double-click on the startup file to open it.



Turn on this virtual machine after loading is complete to run it and enter the system's interface.



The default automatic login account is forlinx, and the password is forlinx.

2.2.7 3. New Ubuntu Development Environment Setup

Note: Beginners are not recommended to set up a system on their own. It is recommended to use an existing virtual machine environment. If you do not need to set up the environment, you can skip this section.

This chapter mainly explains the building process of Ubuntu system.

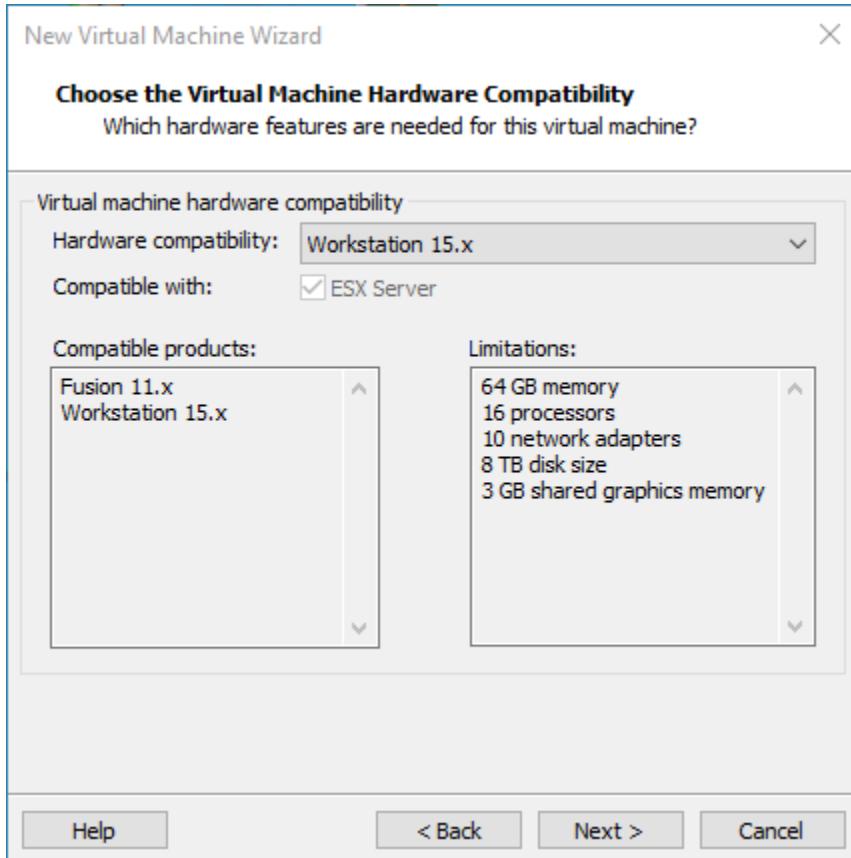
3.1 Ubuntu System Setting up

3.1.1 Ubuntu Virtual Machine Setup

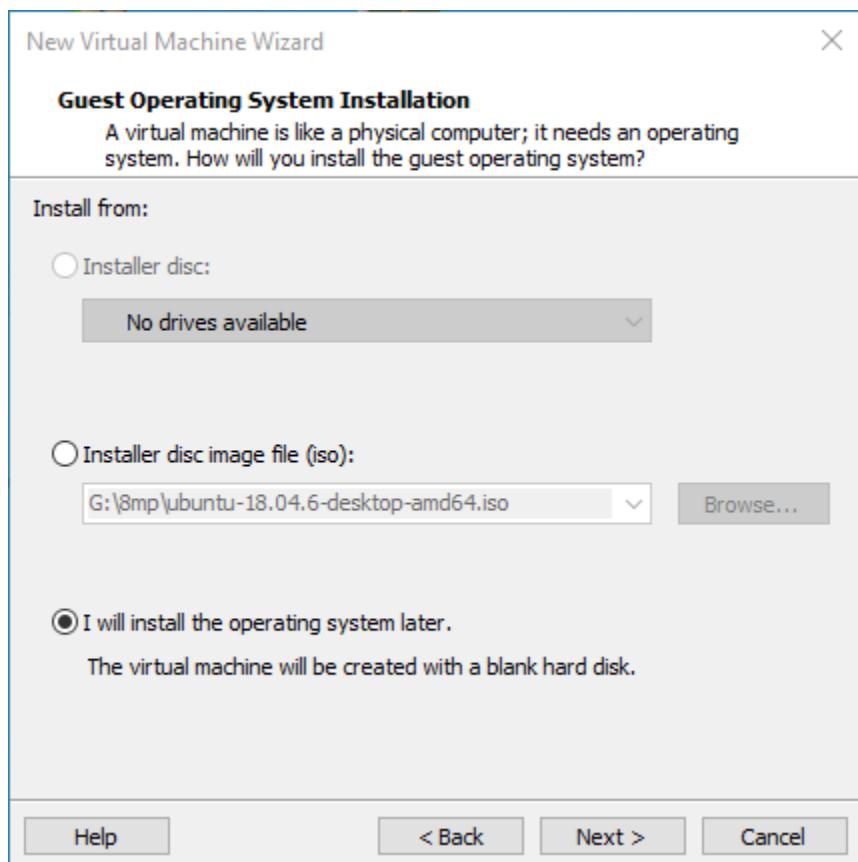
Step 1: Open the VMware software and click [File]/ [New Virtual Machine]. Enter the following interface.



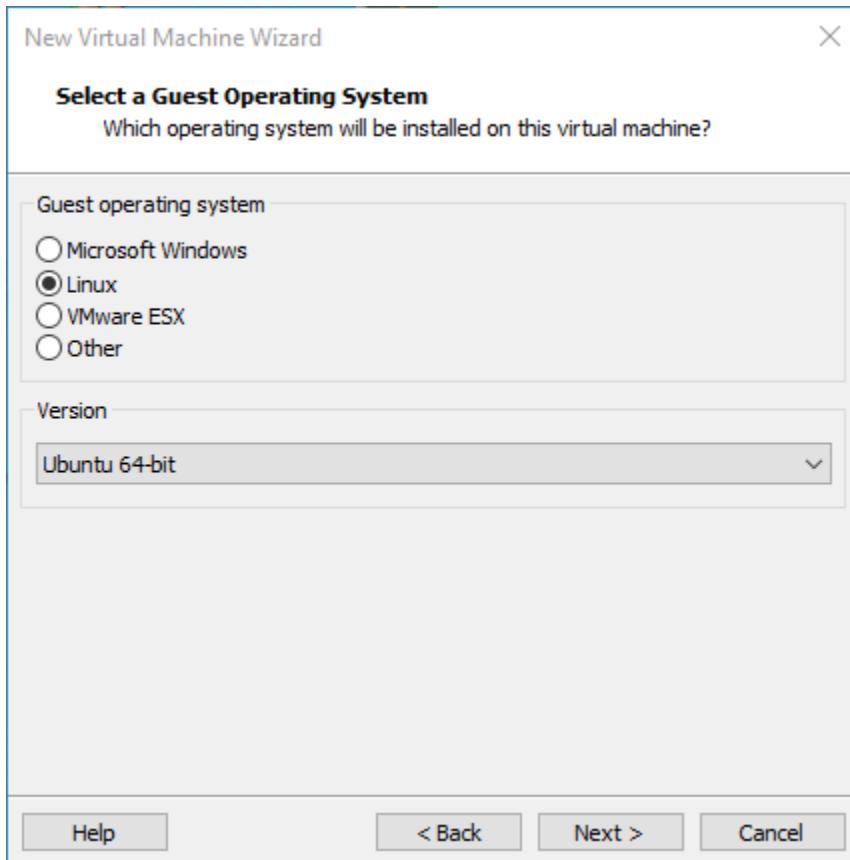
Step 2: Select Custom and click “Next” .



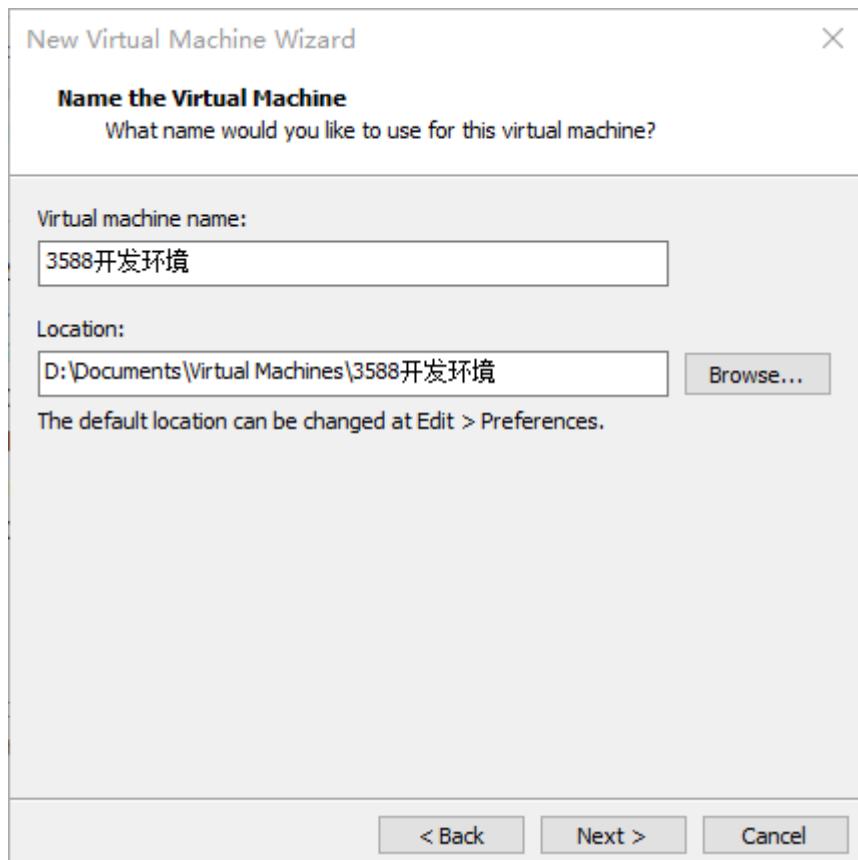
Select the compatibility with the corresponding version of VMware, which can be found in Help->About VMware Workstation, and click “Next” .



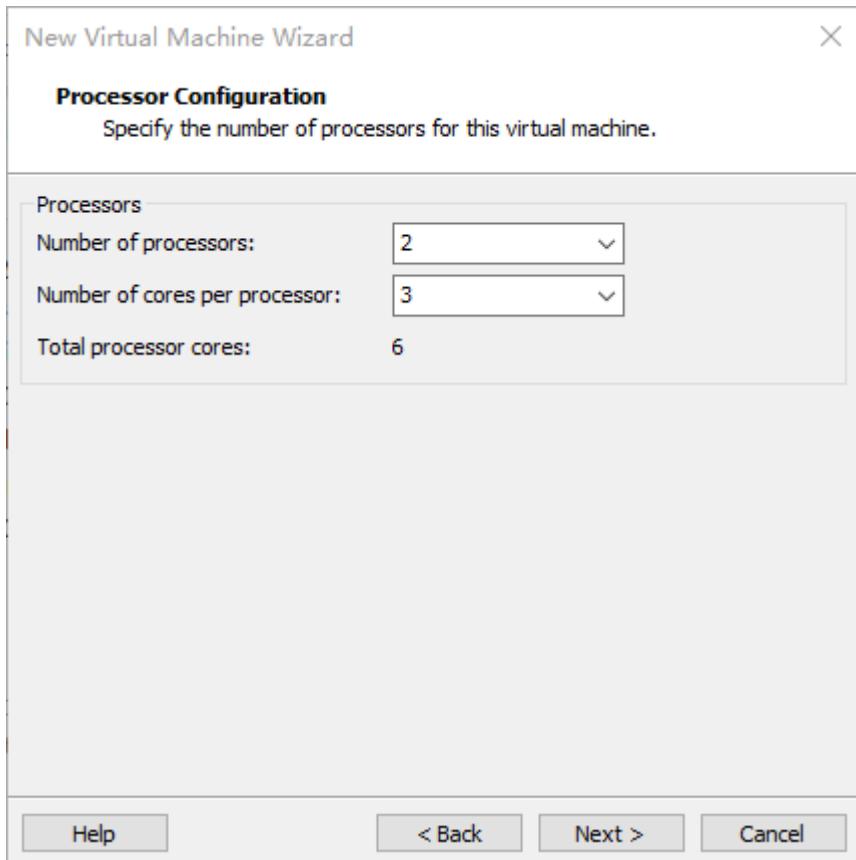
Select Install the operating system later and click “Next” .



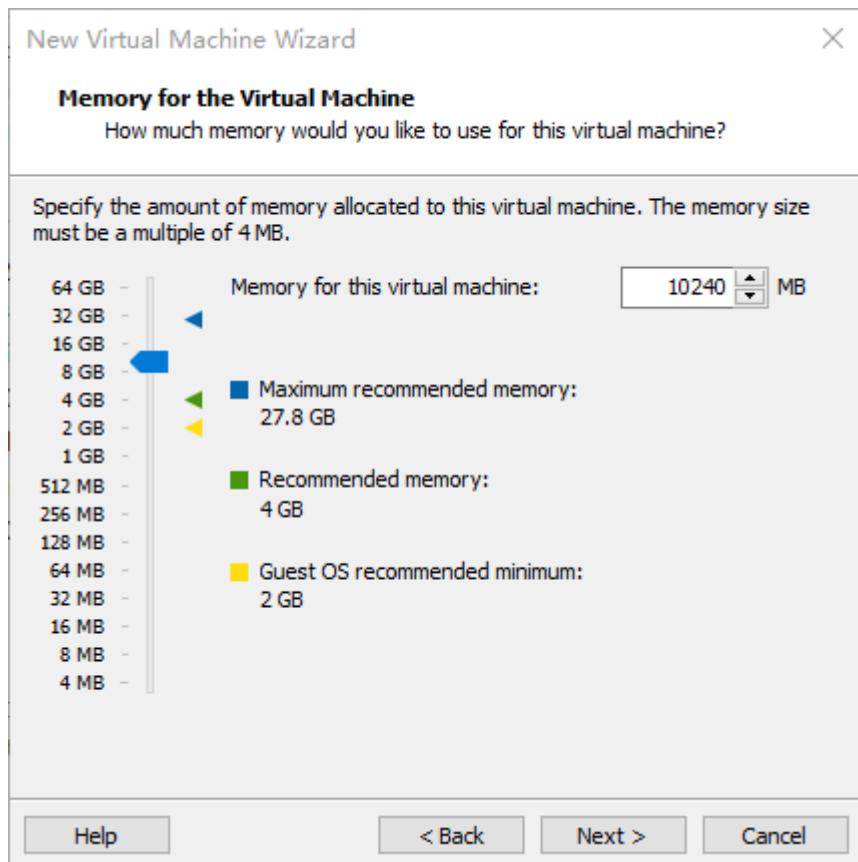
Keep the default settings and click “Next” .



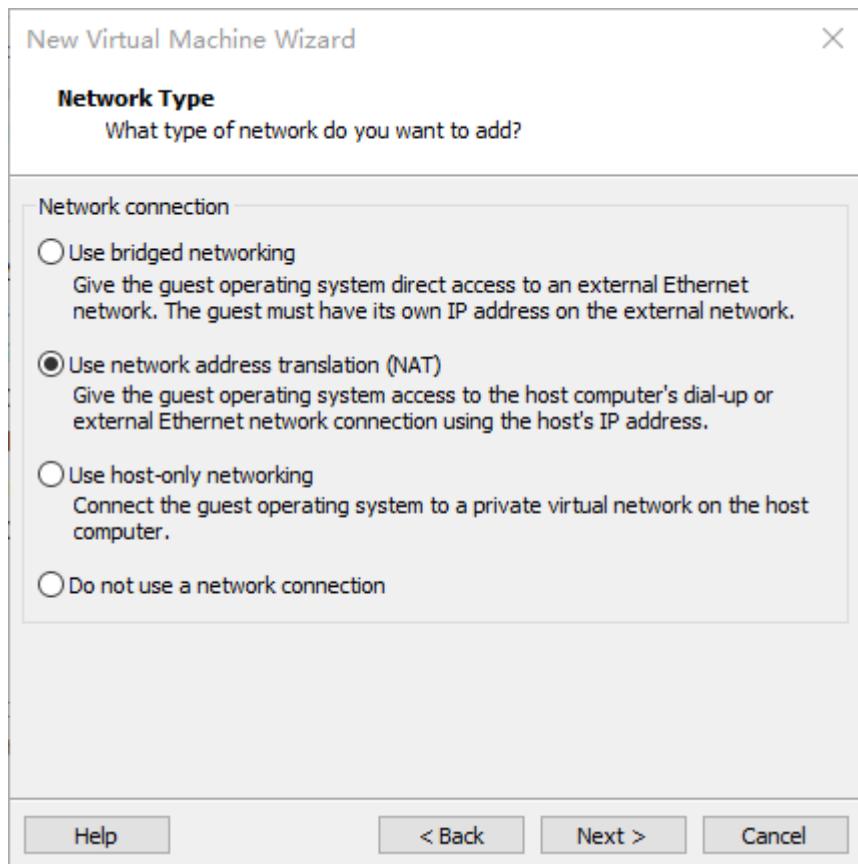
Modify the name and installation location of your virtual machine, and click “Next” .



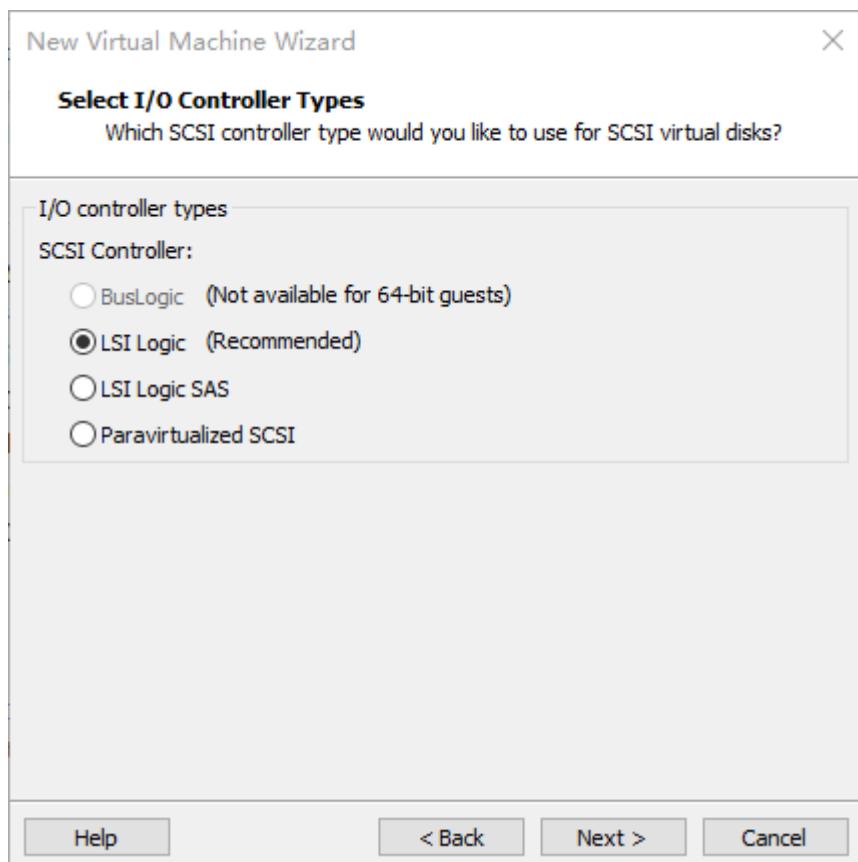
Configure the number of CPU based on your computer's actual specifications.



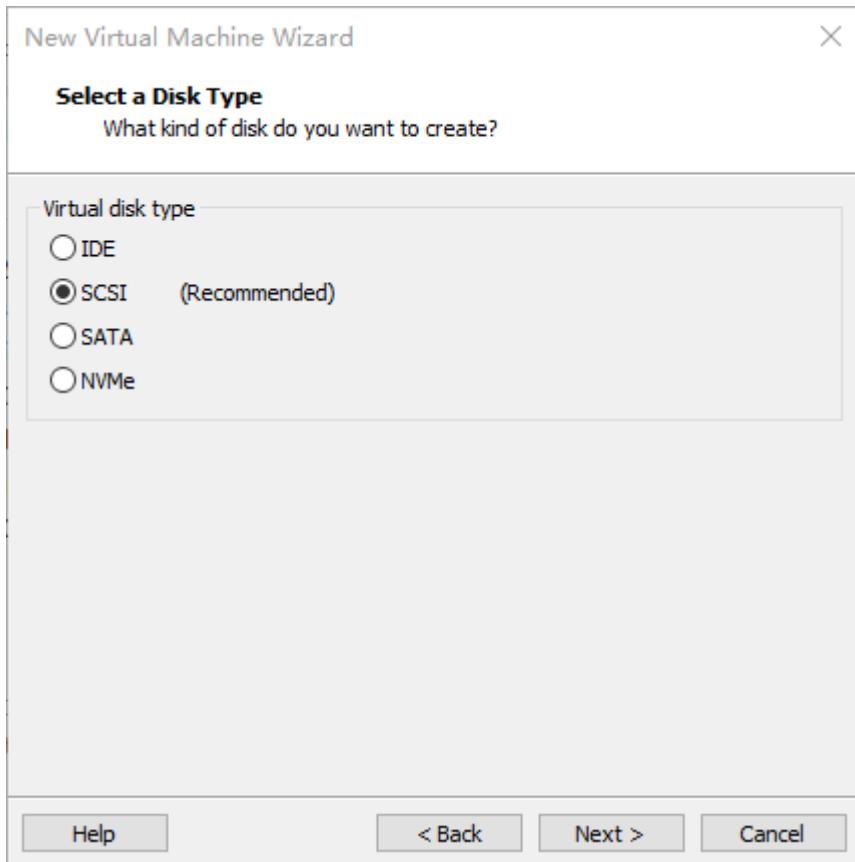
Set the memory size according to the actual situation. It is recommended to use 16g



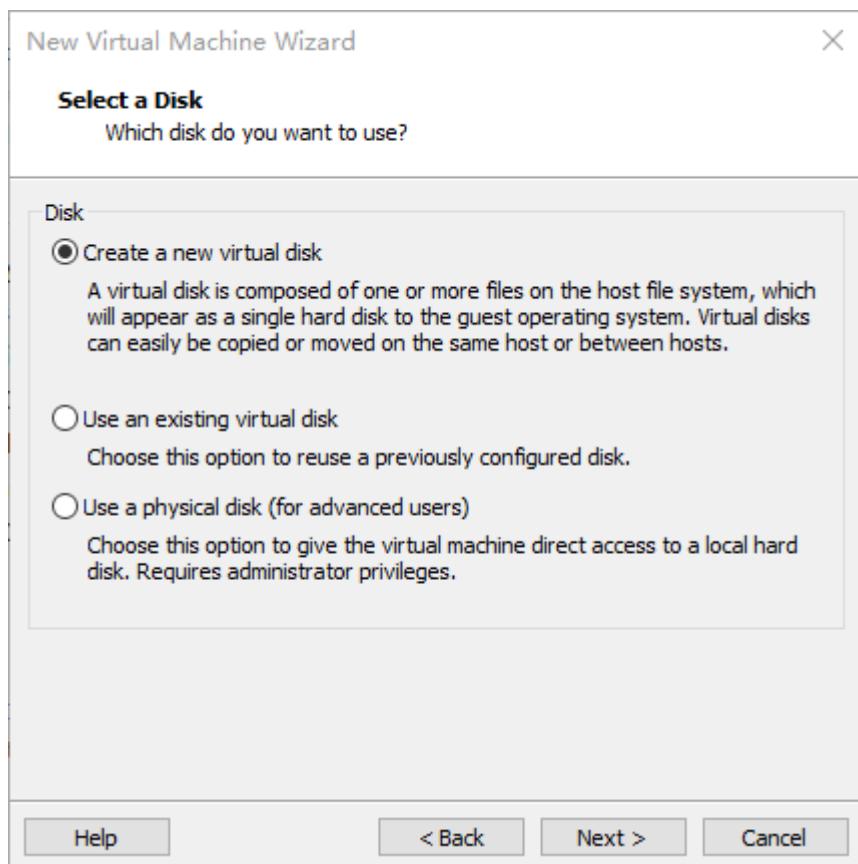
Set the network type, default to NAT mode then click “Next” . Keep the default values for the remaining steps until you reach the step to specify the disk capacity.



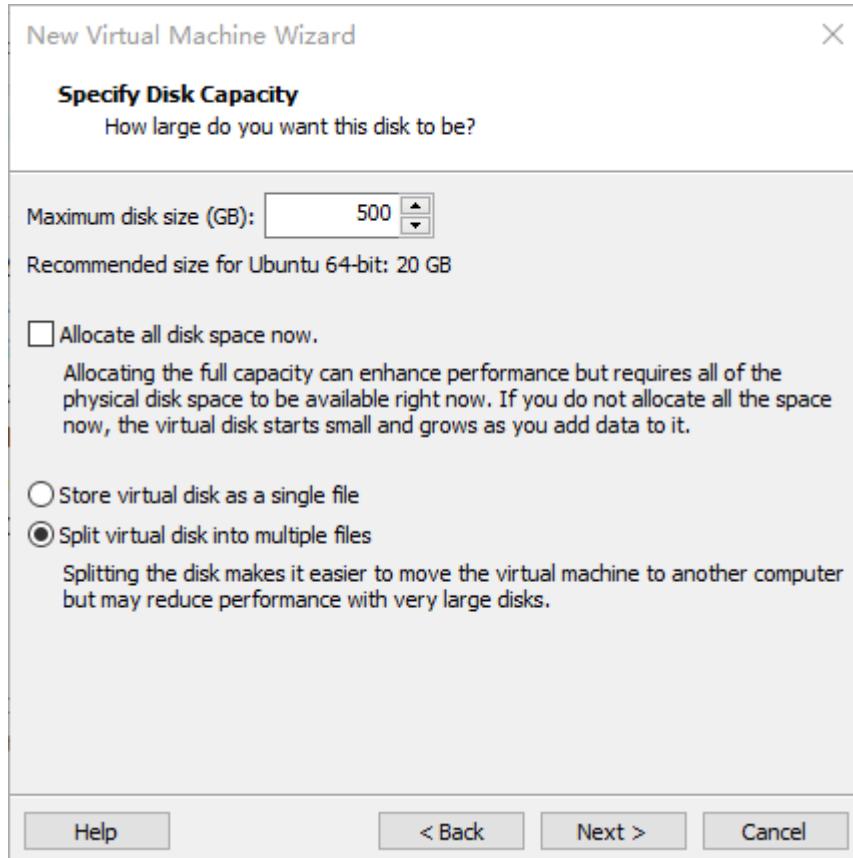
The default selection for the IO controller type here is LSI



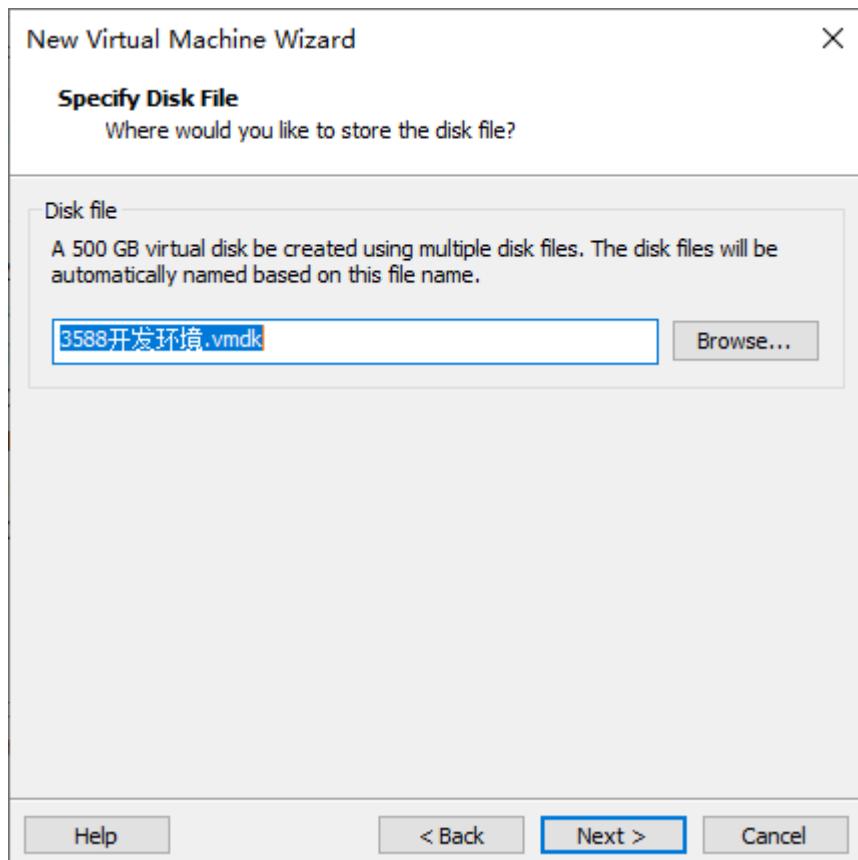
The default selection here is also SCSI.



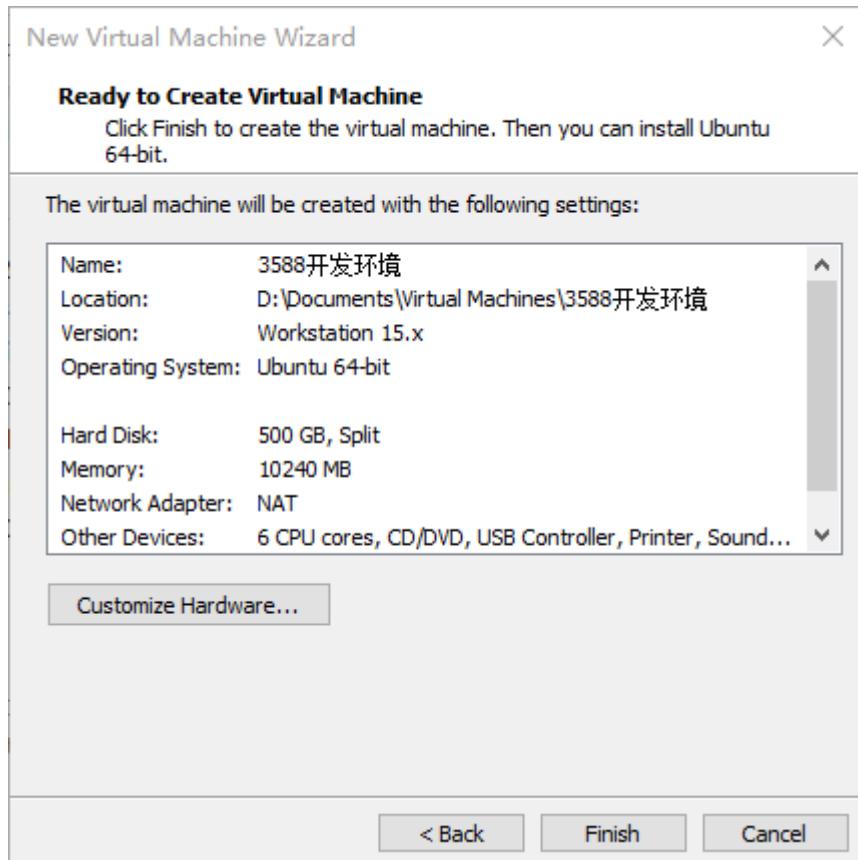
Choose to create a new virtual disk here.



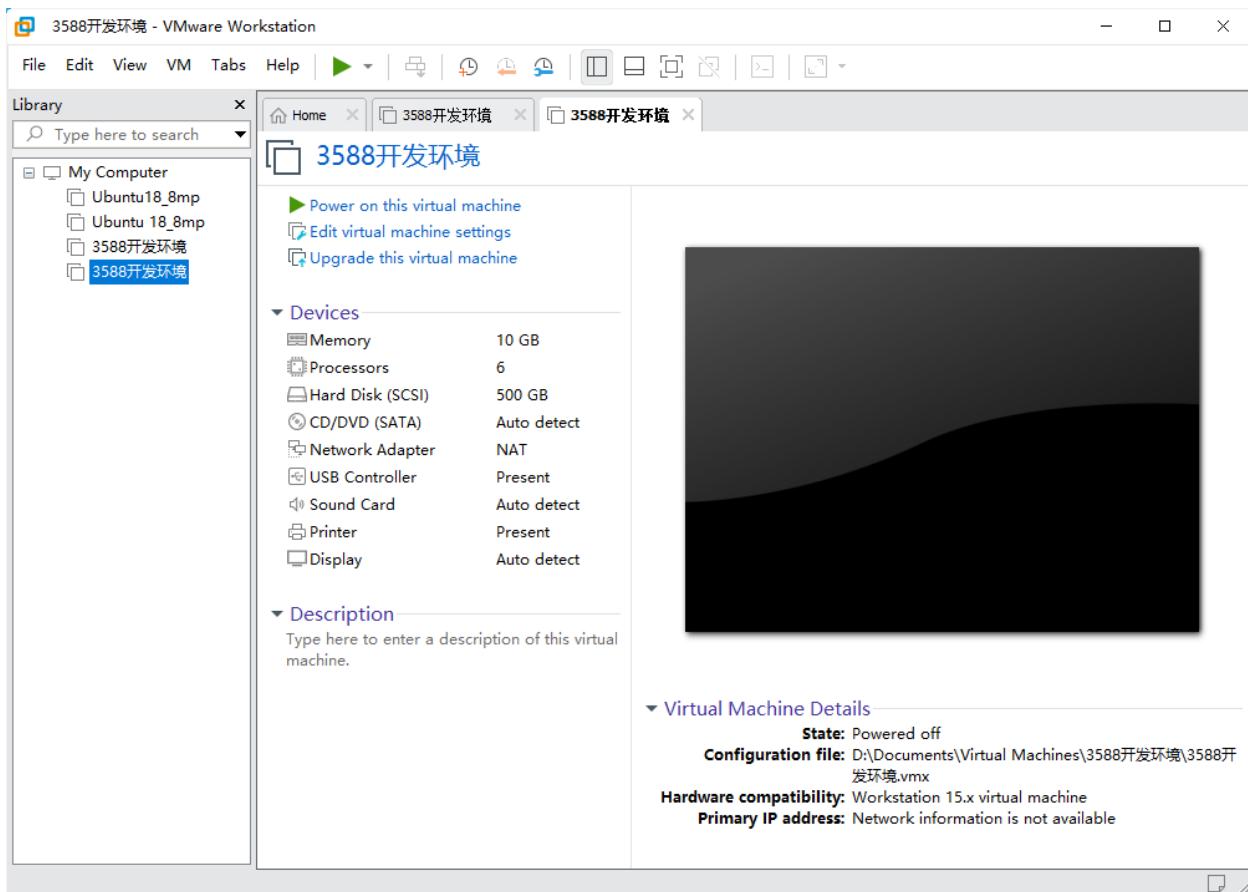
Set the disk size to 500G, select the disk provisioning format, and then click “Next” .



Specify the disk file, the default one here is fine.



Click “Finish” by default.



The virtual machine creation is now complete.

In the next section, it will introduce the installation of Ubuntu system in the virtual machine, which is similar to the installation method in the real machine. Here we describe the method of installing Ubuntu system in a virtual machine.

3.1.2 System Installation

In the previous section, we have created a virtual machine, but we haven't installed the operating system yet, so the virtual machine cannot be started. Next, we will install the Ubuntu operating system in the newly created virtual machine.

Step 1: Go to the Ubuntu official website to obtain the Ubuntu20.04.64.

The download address is: <https://old-releases.ubuntu.com/releases/20.04.3/>

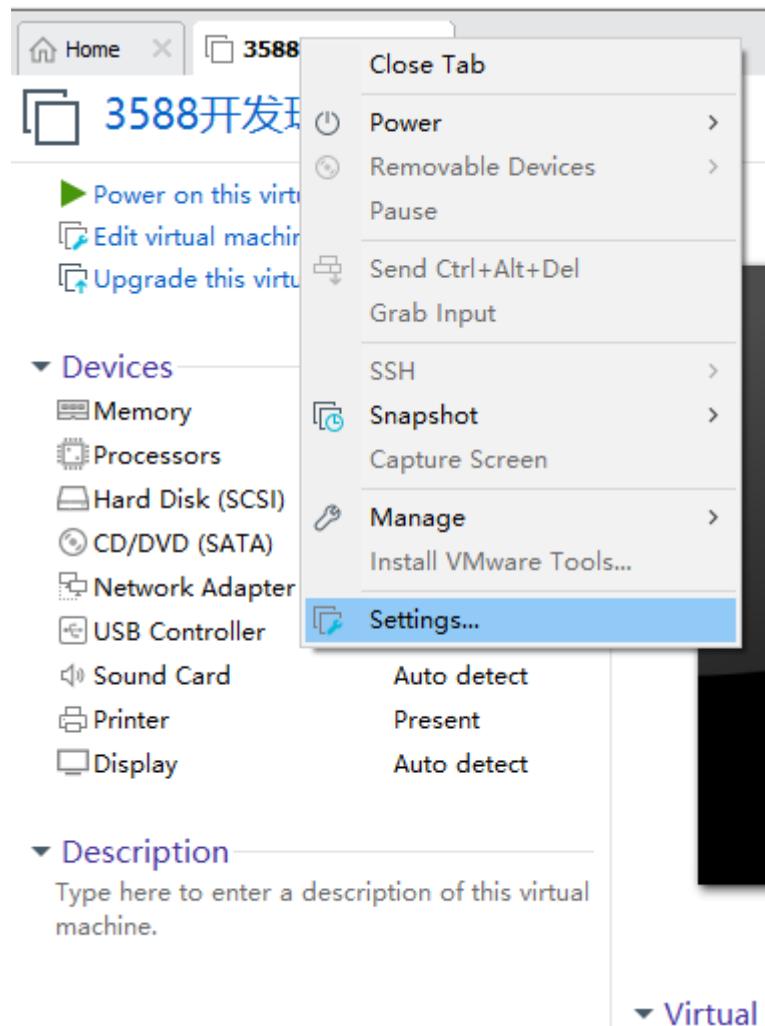
Because the source code is compiled and verified on the 20.04, select and install it. These operations may vary slightly between Ubuntu system versions.

Download “ubuntu-20.04.3-desktop-amd64.iso”

	ubuntu-20.04.3-desktop-amd64.iso	2021-08-19 11:06	2.9G
--	----------------------------------	------------------	------

After downloading the mirror image, you can proceed with the system installation operation.

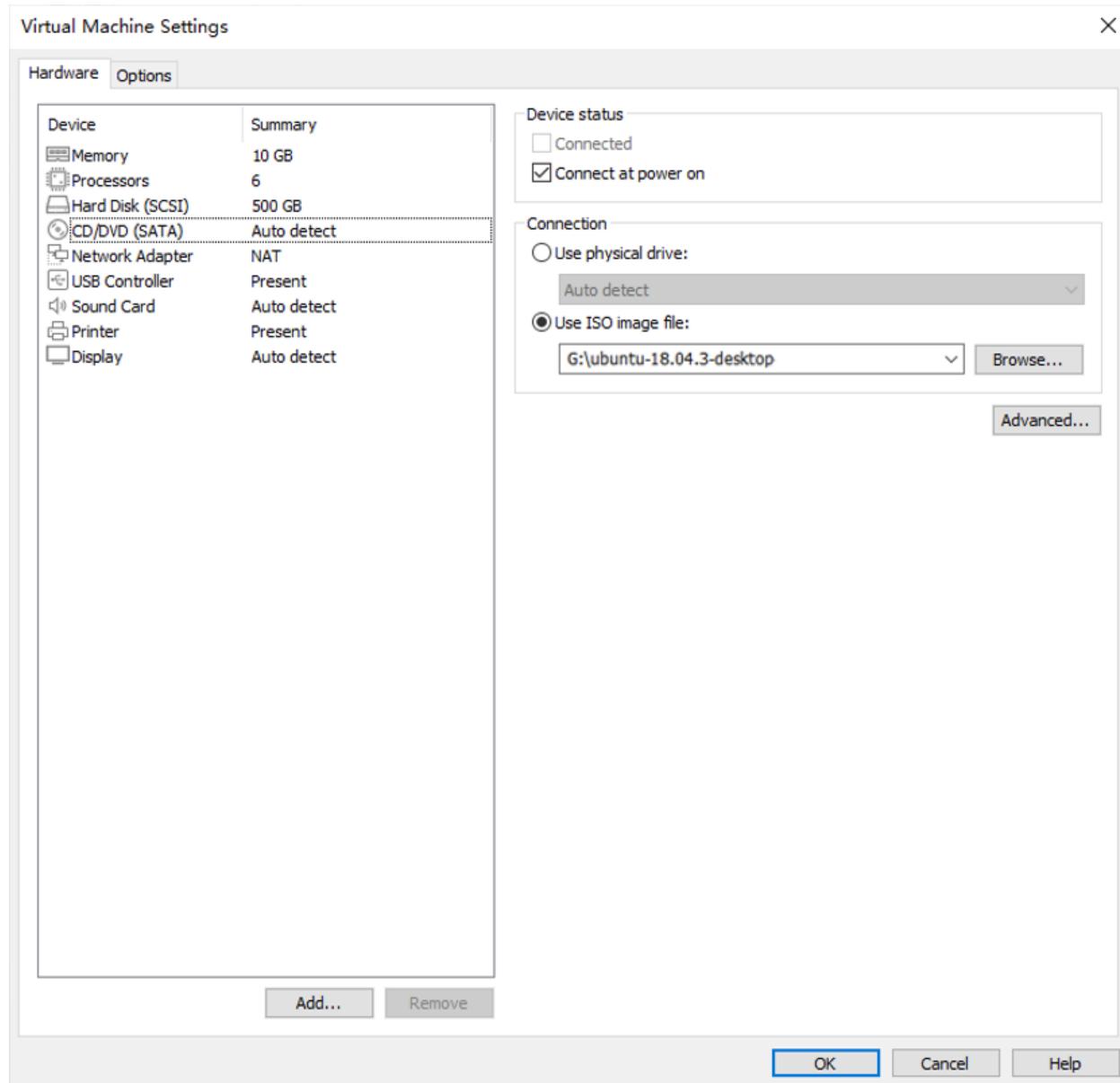
Right-click on the created virtual machine name and select “Settings” from the pop-up menu.



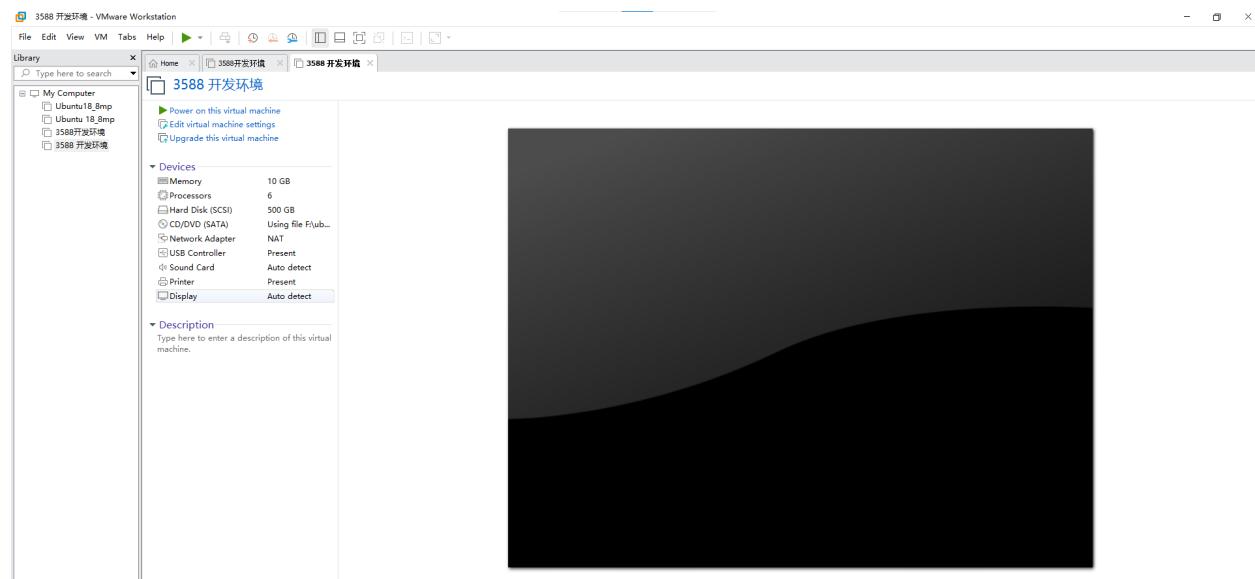
The “Virtual Machine Settings” menu will pop up.

Click on CD/DVD (SATA), select “Use ISO image file,” browse and choose the previously downloaded Ubuntu image, then click “OK” to confirm.

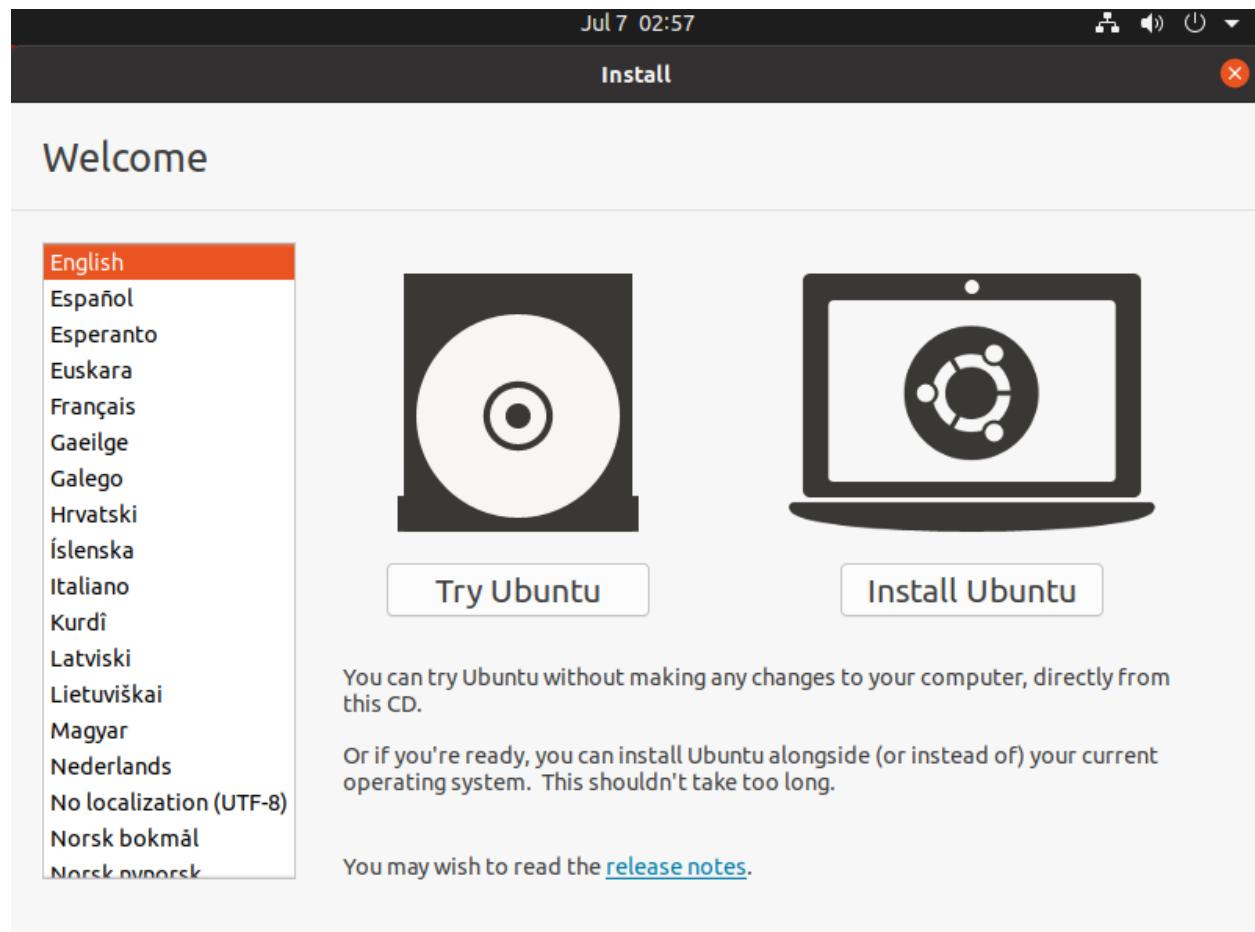
▼ Virtual I



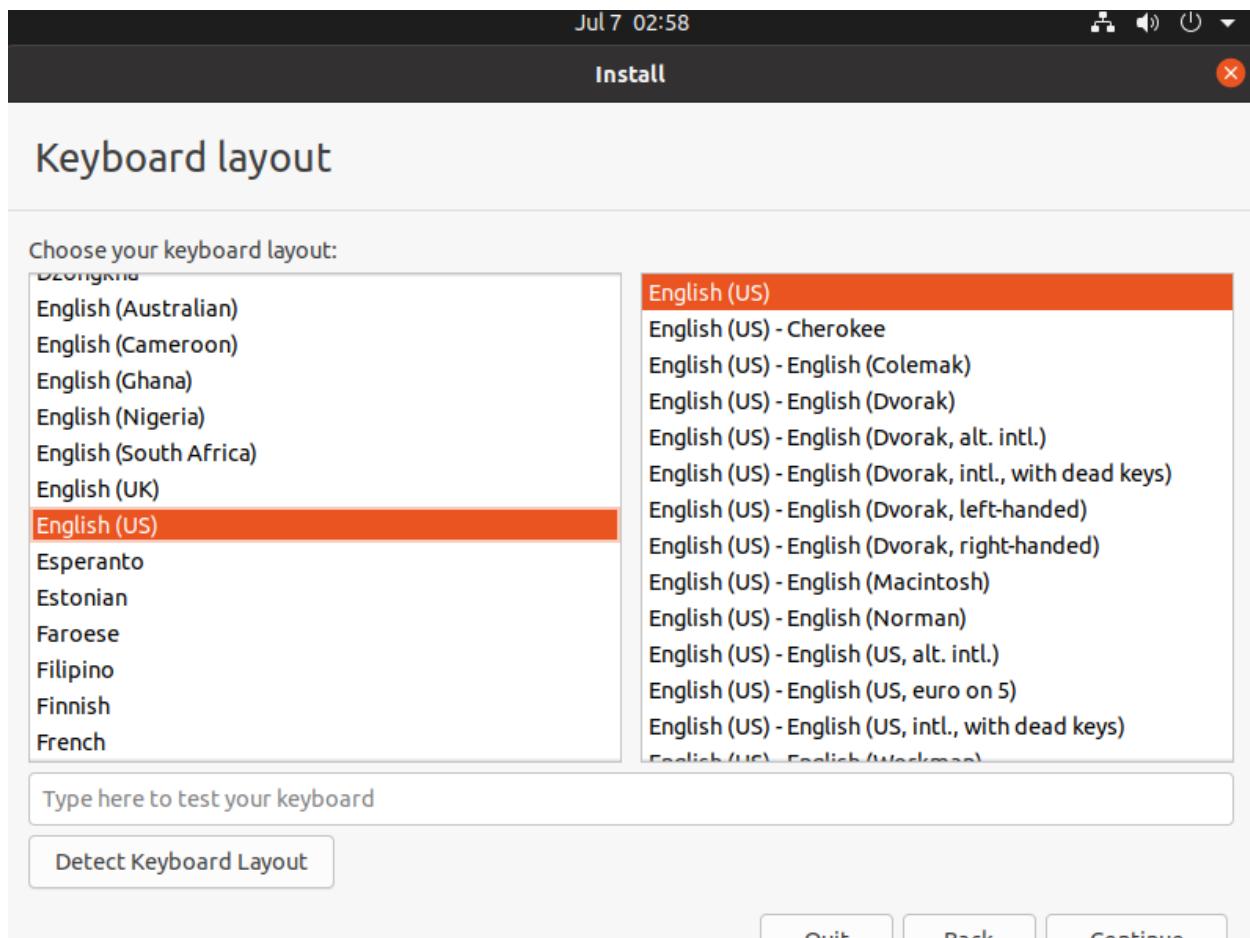
After setting up the image, ensure that the network is available. Then, start the virtual machine and proceed with the installation of the Ubuntu image.



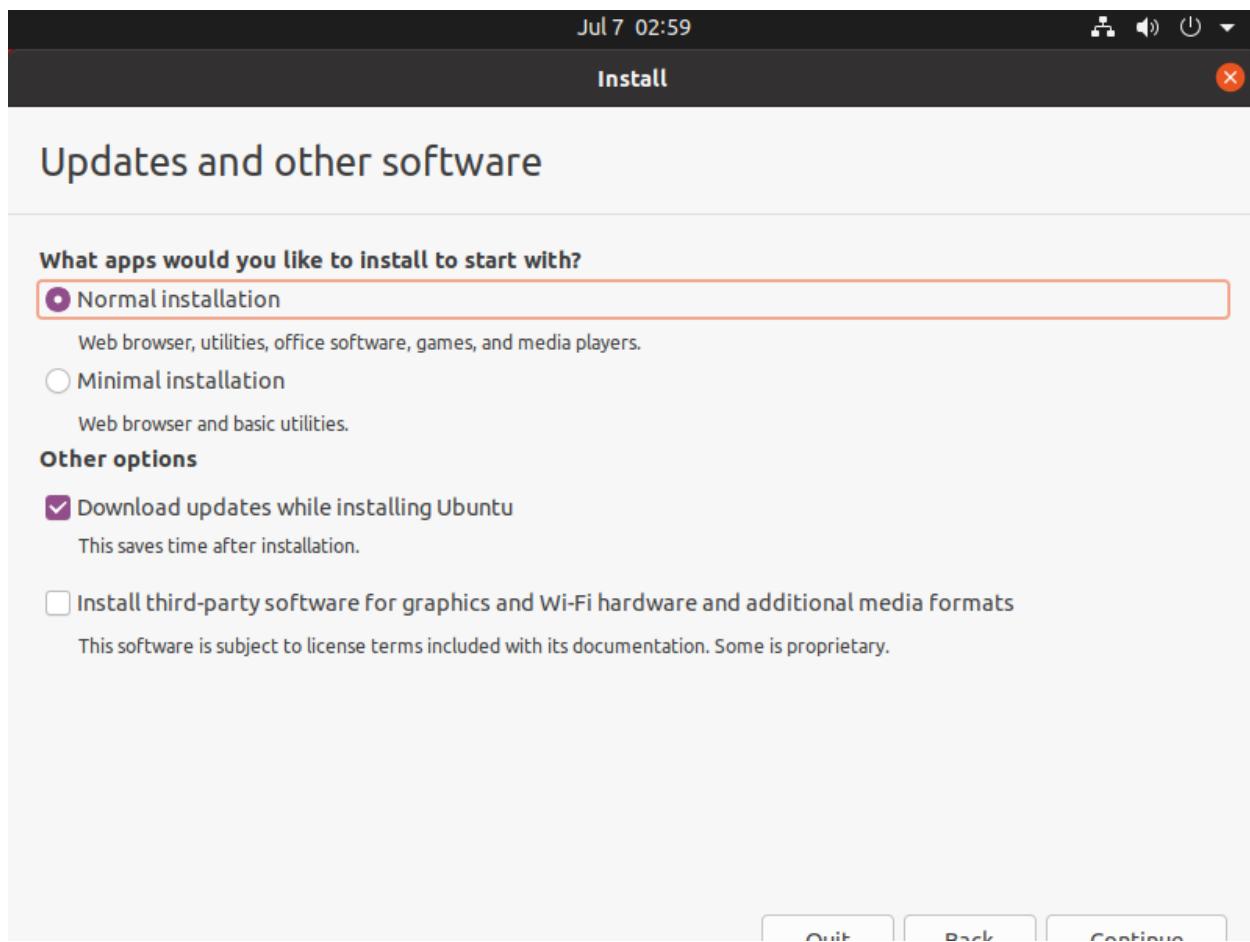
After starting the virtual machine, wait for the installation interface to appear as shown below.



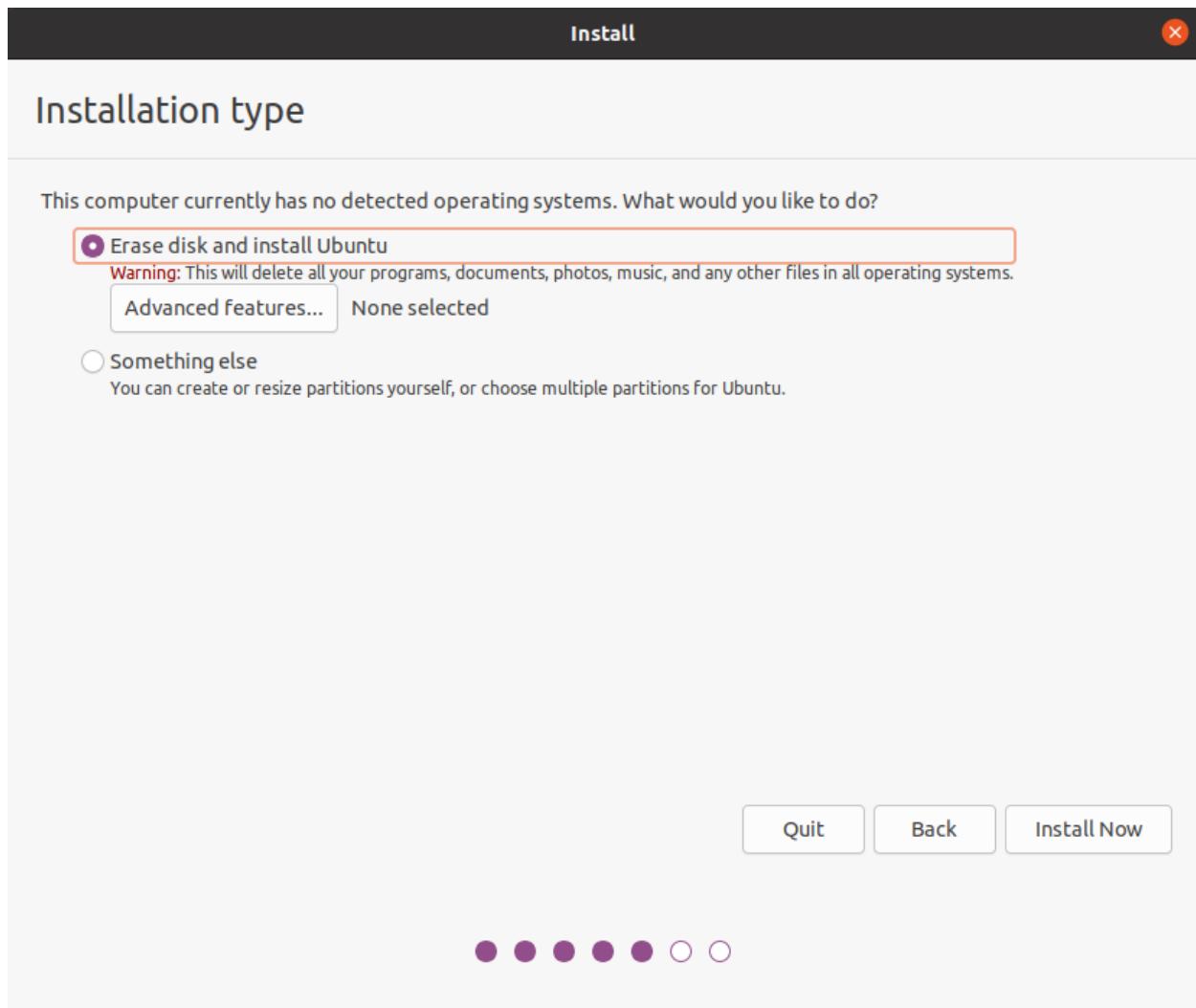
After selecting the language on the left side as shown in the image, click “Install Ubuntu”, and the language selection interface will pop up. The default language of Ubuntu is English, but of course, you can also choose Others.



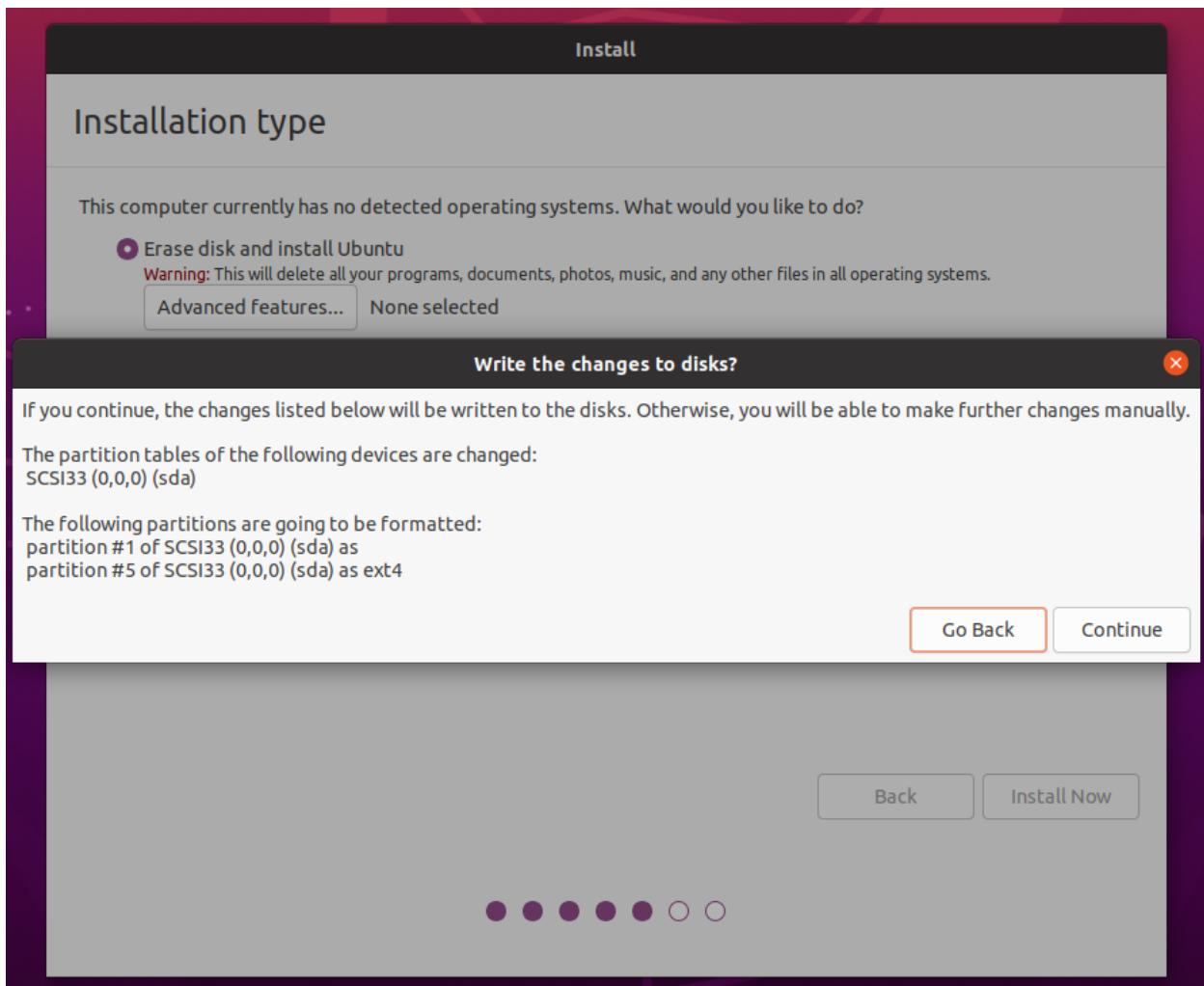
The default selected language can also be reset at a later stage, after the selection is complete continue.



Next, select “Continue” as the default option to proceed with the installation. The installation process might be slow. Then, click “Continue” again.



By default, when you click on “Install Now”, a dialog box will appear as shown in the image. Simply click “Continue” to proceed.



Next, select the timezone. You can either click on the Shanghai timezone or enter “Shanghai” (or choose the appropriate timezone based on your location). Then, click “Continue” to proceed.

Finally, set your username and password. You can choose either automatic login or login with a username and password. Click “Continue” to start the automatic installation.

Install

Who are you?

Your name: ✓

Your computer's name: ✓
The name it uses when it talks to other computers.

Pick a username: ✓

Choose a password: Weak password

Confirm your password: ✓

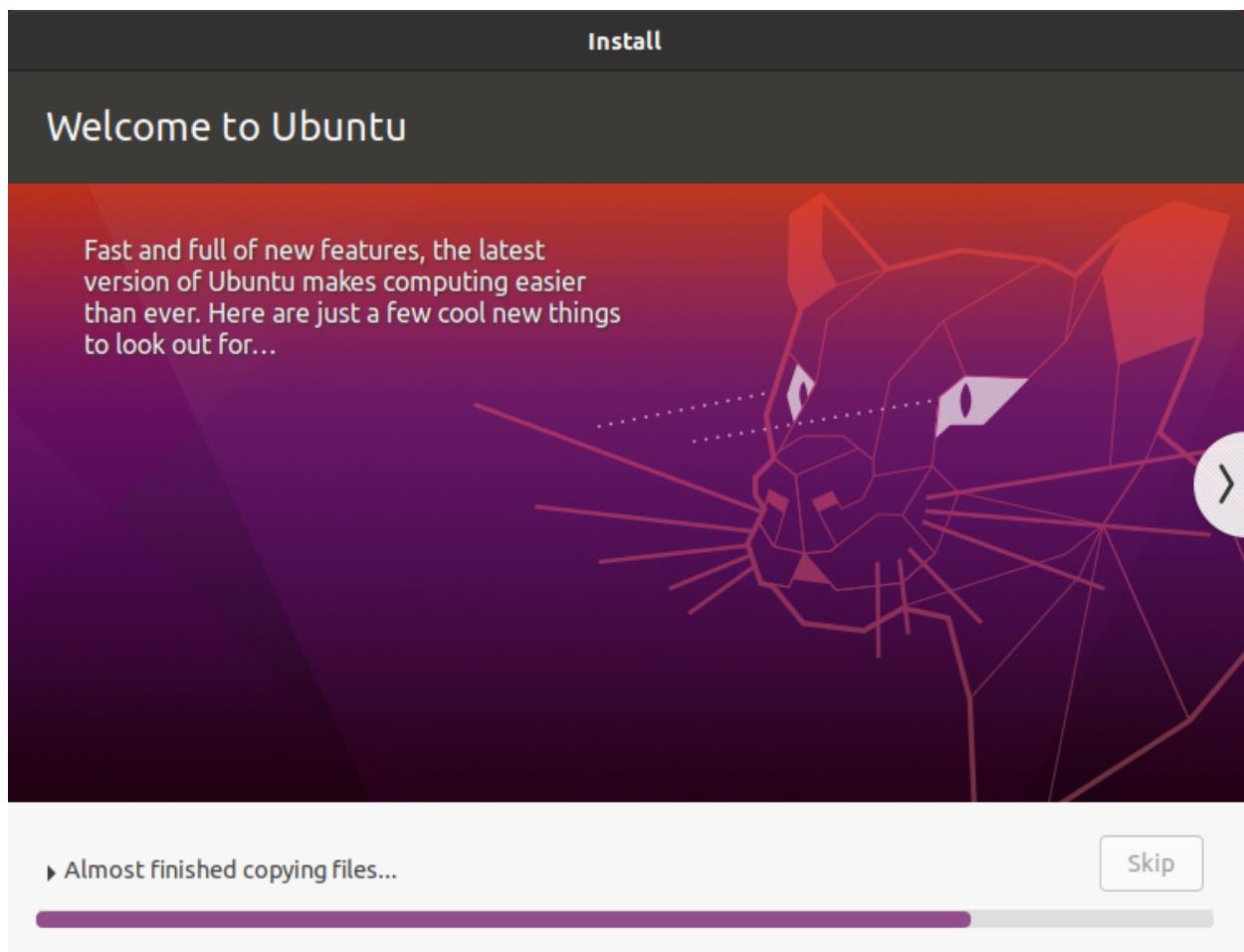
Log in automatically
 Require my password to log in
 Use Active Directory

You'll enter domain and other details in the next step.

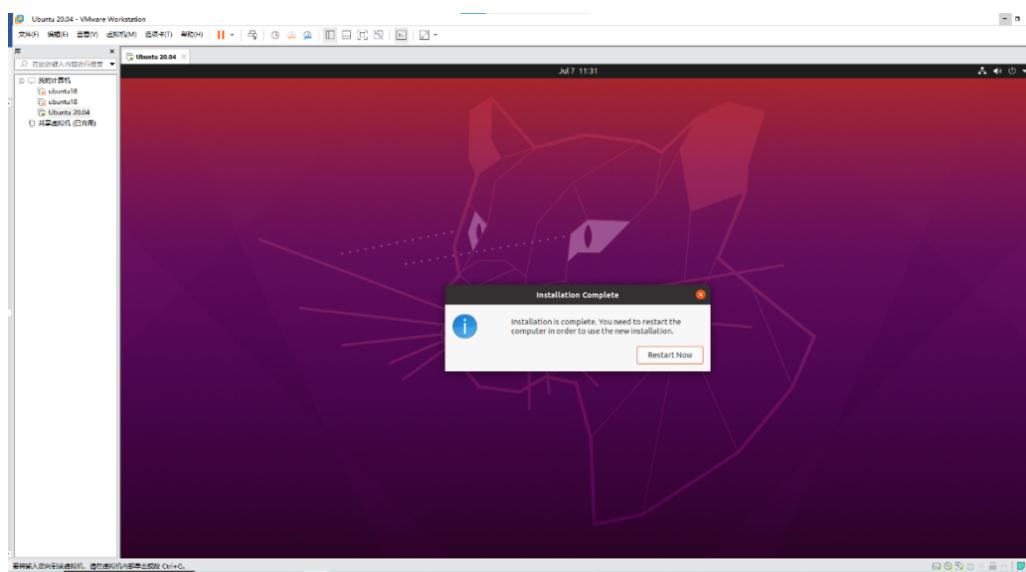
[Back](#) [Continue](#)

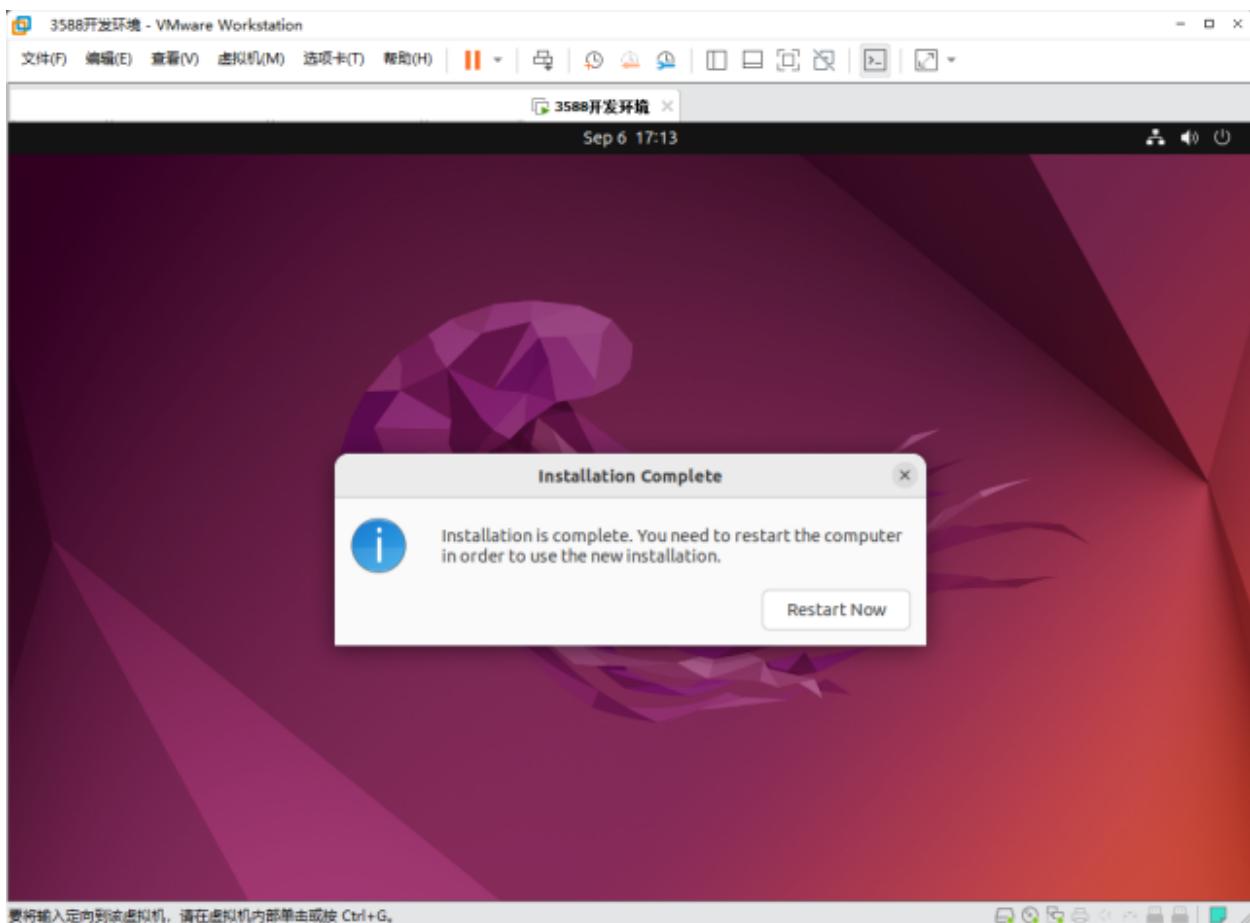
● ● ● ● ● ● ●

If the internet connection is poor, you can Skip without affecting the installation process.

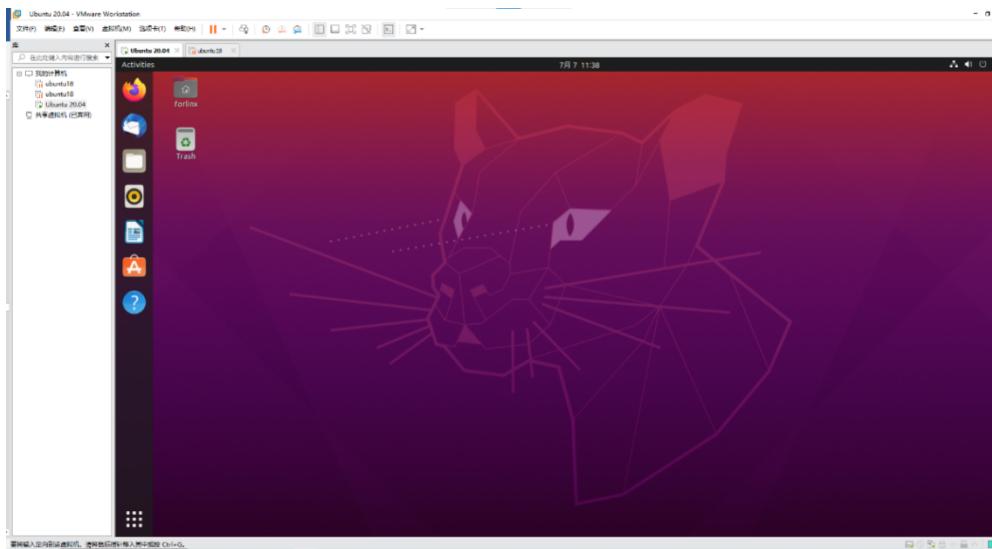


Click “Restart” Now to reboot.





The system interface after the reboot is complete.



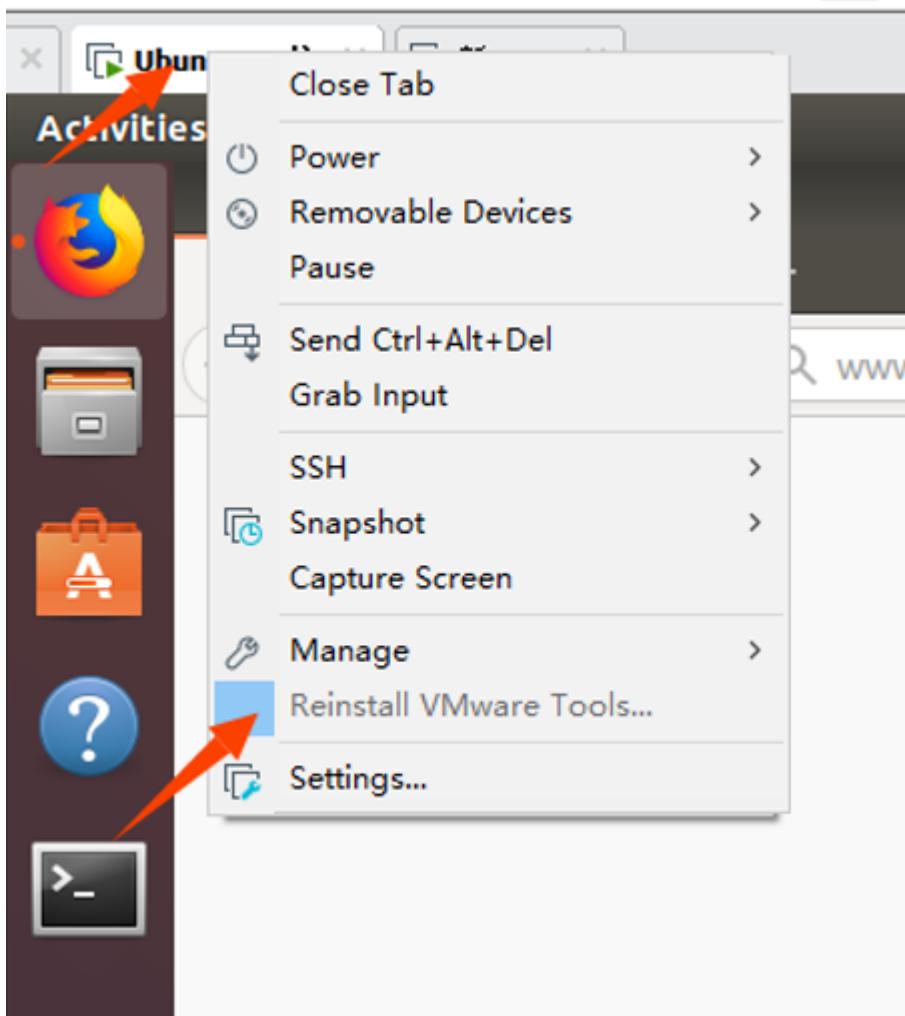
The ubuntu system installation is complete.

3.1.3 Ubuntu Basic Configuration

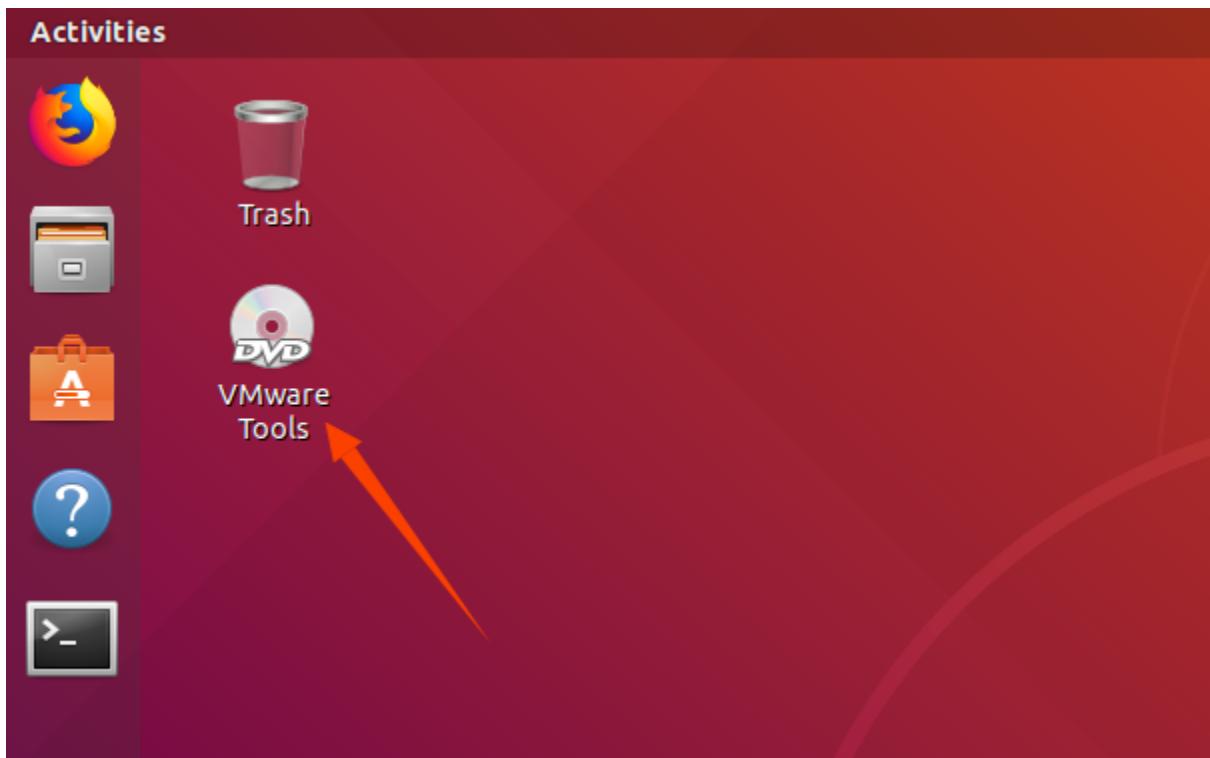
After installing the Ubuntu20.04 operating system, there are a few configurations to make.

VMware Tools Installation:

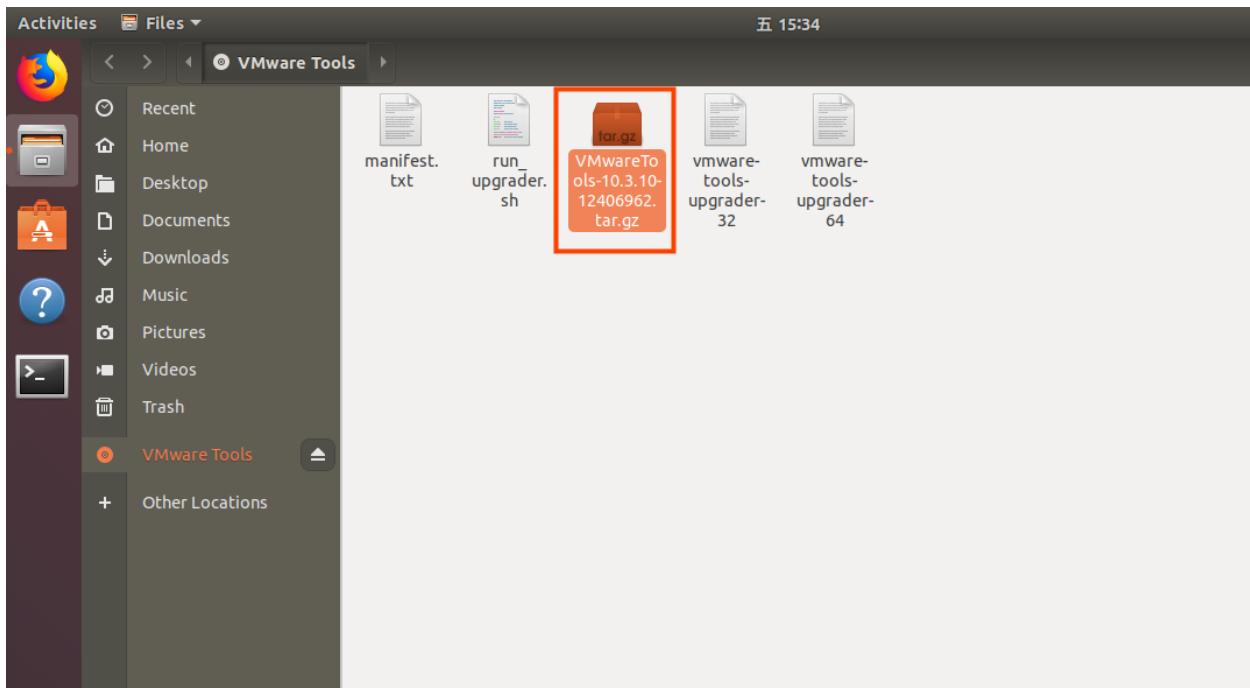
Next, install VMware Tools. Without installing this tool, you won't be able to copy and paste and drag file between the Windows host and the virtual machine. First click on "Virtual Machine" on the VMware navigation bar, then click "Install VMware Tools" in the drop-down box.



Once done, enter Ubuntu and the VMware Tools CD will appear on your desktop and click into it.

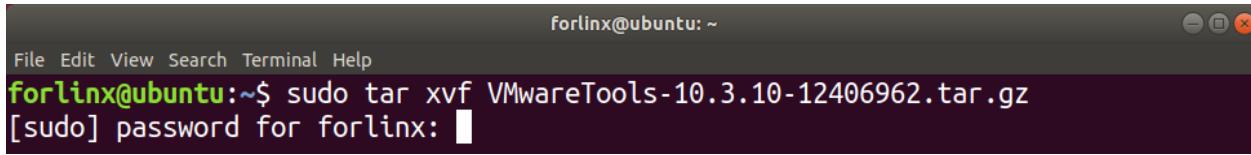


Enter and see a compressed file `VMwareTools-10.3.10-12406962.tar.gz` (it may be different for different VM versions); copy the file under the home directory (i.e. the directory with the home personal username)



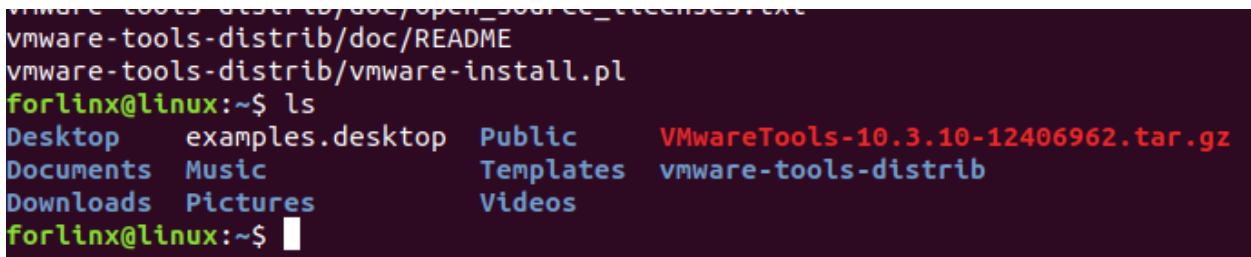
Press [Ctrl+Alt+T] to bring up the Terminal Command Interface and enter the command:

```
forlinx@ubuntu:~$ sudo tar xvf VMwareTools-10.3.10-12406962.tar.gz
```



```
forlinx@ubuntu: ~
File Edit View Search Terminal Help
forlinx@ubuntu:~$ sudo tar xvf VMwareTools-10.3.10-12406962.tar.gz
[sudo] password for forlinx: [REDACTED]
```

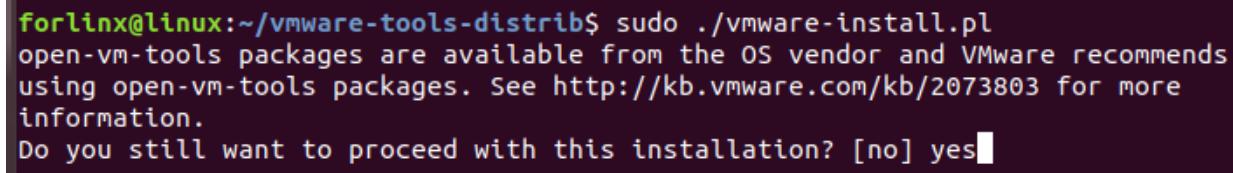
After the extraction is complete, a file named “vmware-tools-distrib” will appear.



```
forlinx@linux:~/vmware-tools-distrib$ ls
VMwareTools-10.3.10-12406962.tar.gz
forlinx@linux:~$ [REDACTED]
```

Go back to the terminal and type cd vmware-tools-distrib to enter the directory.

Enter: sudo ./vmware-install.pl followed by pressing Enter. Then, enter your password and the installation process will begin. When prompted, you can input “yes” and press Enter to proceed. For any other inquiries, simply press Enter to go with the default installation settings.



```
forlinx@linux:~/vmware-tools-distrib$ sudo ./vmware-install.pl
open-vm-tools packages are available from the OS vendor and VMware recommends
using open-vm-tools packages. See http://kb.vmware.com/kb/2073803 for more
information.
Do you still want to proceed with this installation? [no] yes [REDACTED]
```

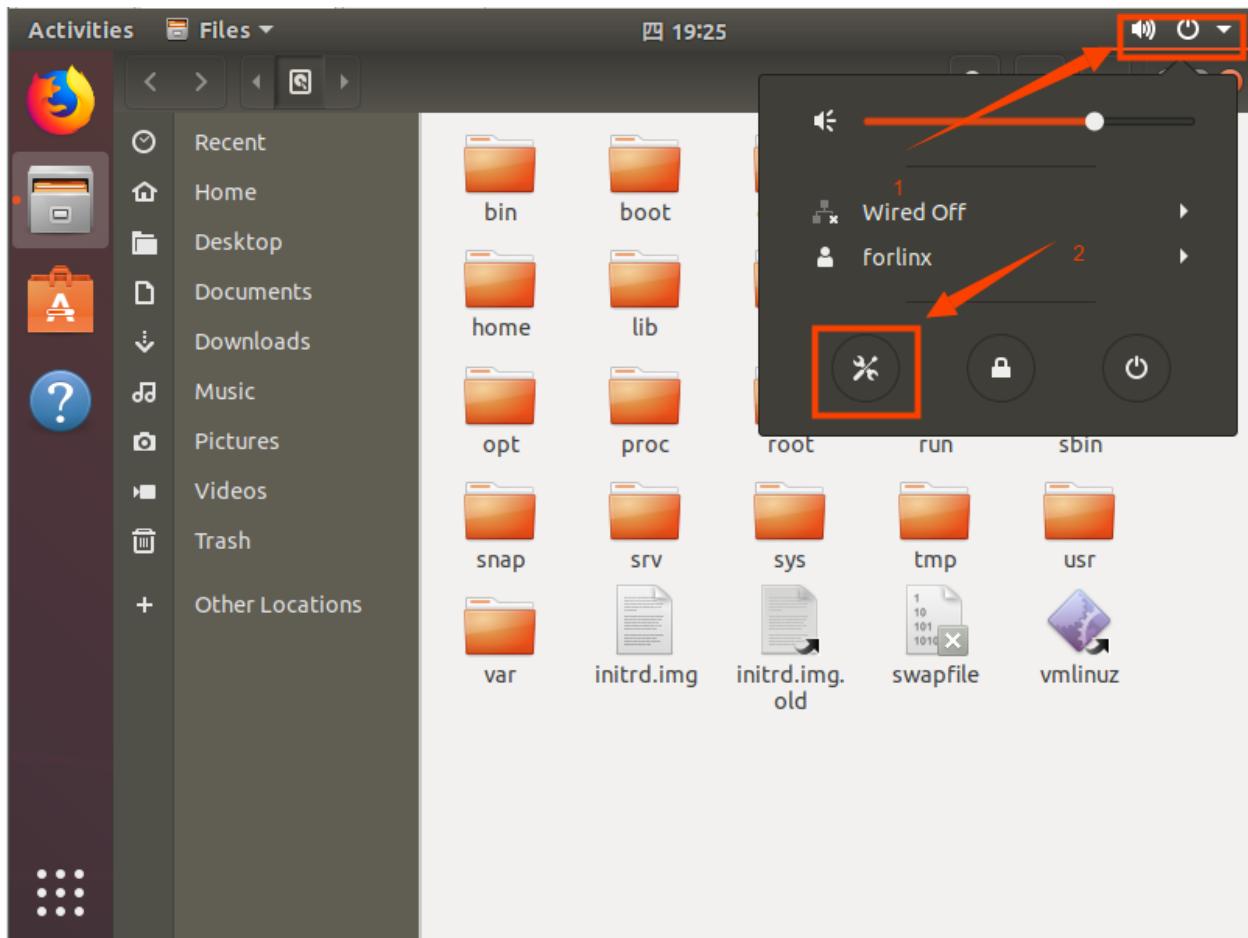
Once the VMware tools is complete, we can implement file copy and paste between Windows and Ubuntu.

The virtual machine is displayed full screen:

If the virtual machine is not able to be displayed in full screen, you can resolve this issue by clicking on “View” and selecting “Autofit Guest.” This will adjust the display to fit the screen automatically, enabling you to have a full-screen experience in the virtual machine.

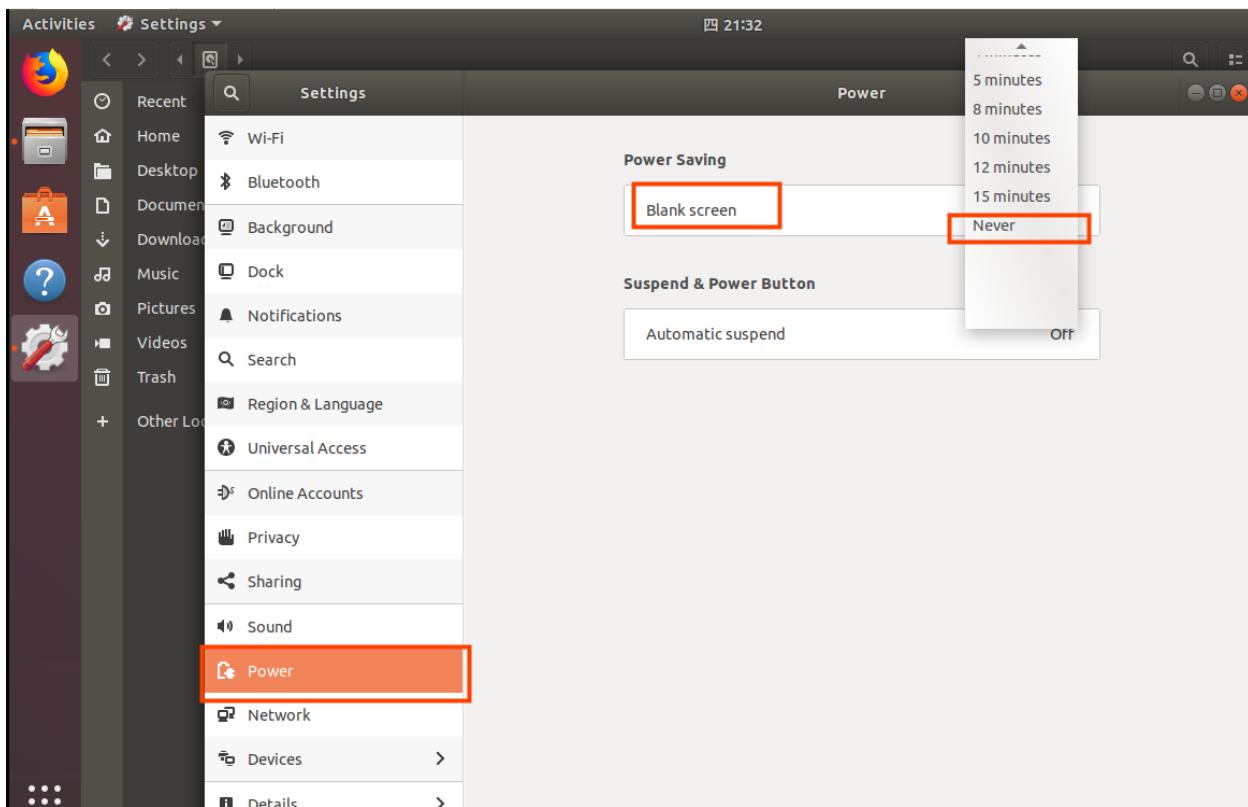


Make most of the system settings in the location shown. A lot of the setup requirements on Ubuntu can be done here.



Virtual machine hibernation settings:

Also, the default hibernation is 5min, if you don't want to set hibernation, just set it to Never by setting Power->Blank screen.

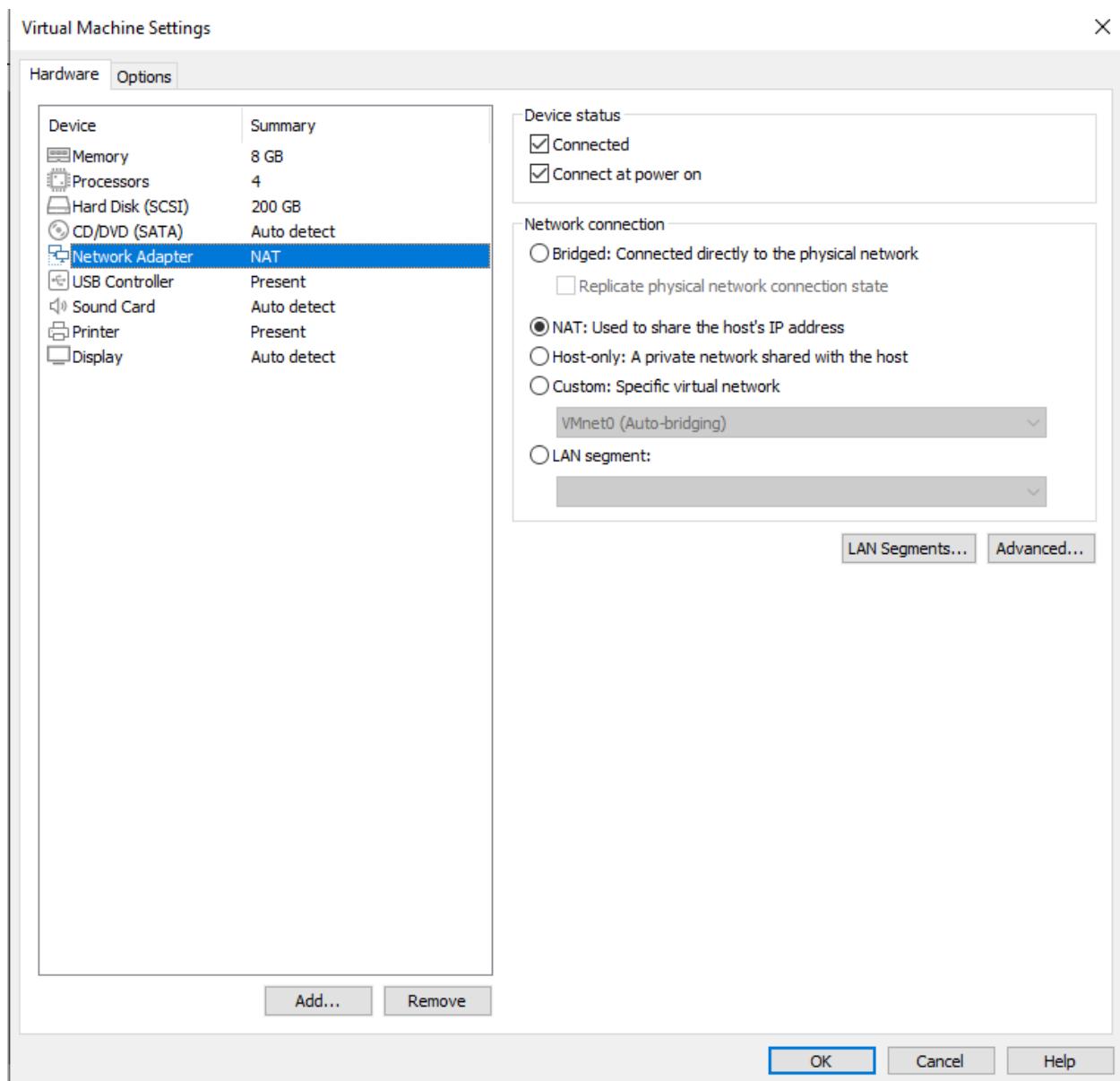


3.1.4 Network Settings for Virtual Machines

3.1.4.1 NAT Connection Method

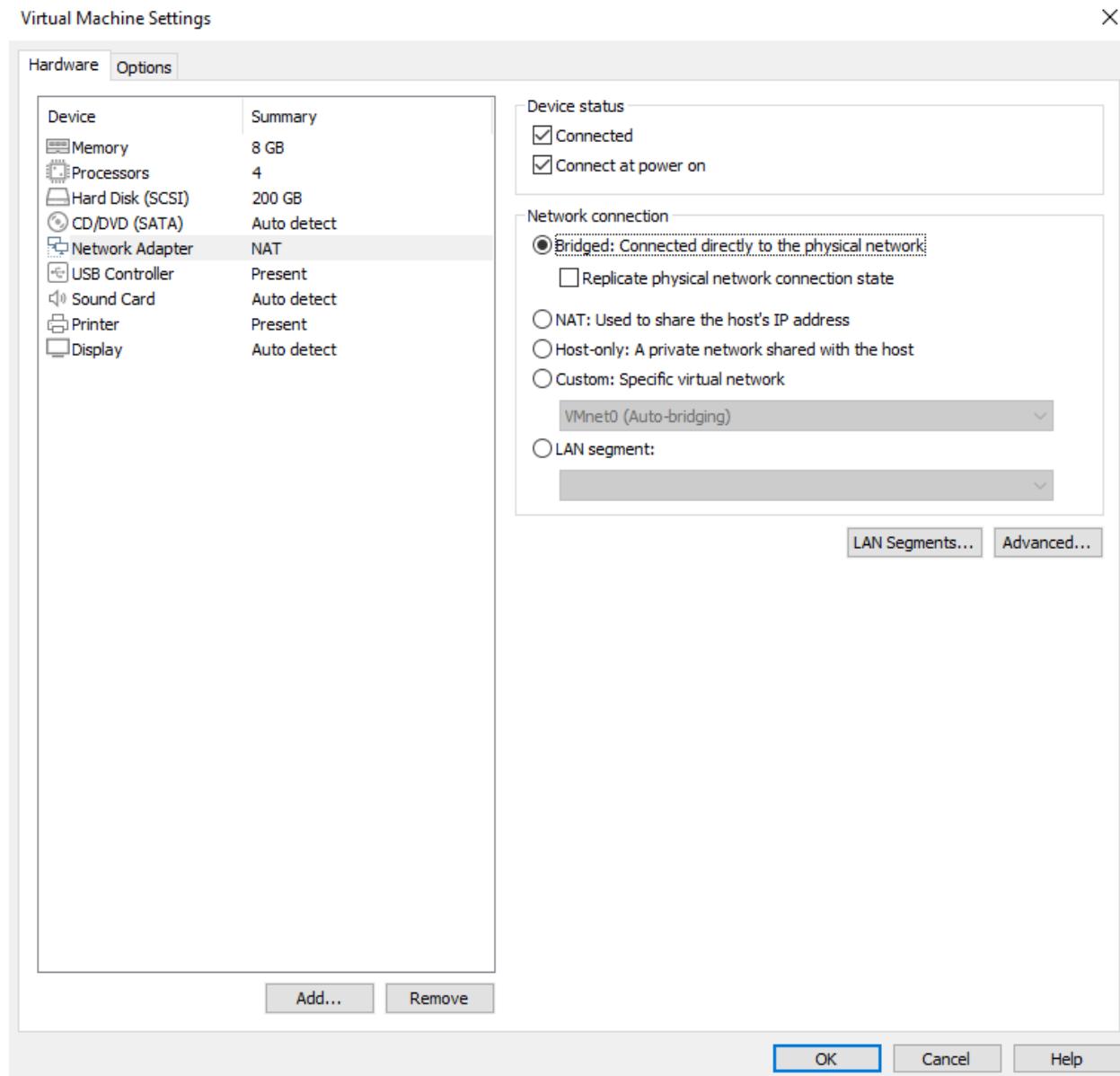
By default, after the virtual machine is installed, the network connection method is set to NAT, which shares the host machine's IP address. This configuration does not need to be changed when performing tasks like installing dependencies or compiling code.

When the VMware virtual NIC is set to NAT mode in a virtual machine, the network in the Ubuntu environment can be set to dynamic IP. In this mode the virtual NAT device and the host NIC are connected to communicate for Internet access. This is the most common way to access the external network.



3.1.4.2 Connections for Bridges

When the VMware virtual NIC device is in bridge mode, the host NIC and the virtual machine NIC communicate through the virtual bridge, and the network IP and the host need to be set in the same network segment in the Ubuntu environment. If accessing an external network, you need to set the DNS to be consistent with the host NIC. If TFTP, SFTP and other servers are used, the network contact mode of the virtual machine needs to be set as the bridge mode.



3.2 Libraries for Installing Linux Compilation System

Note: If you use the development environment provided by us, this section can be skipped directly.

Compiling for Linux requires the installation of a number of toolkits. Make sure that your computer or virtual machine can be connected to the Internet normally before the operation in this section. If the network is disconnected during the installation, please follow the following steps to install.

1. Install the necessary packages for compiling Linux.

```
forlinx@ubuntu:~$ sudo apt-get update
forlinx@ubuntu:~$ sudo apt-get install openssh-server vim git fakeroot      //
˓→ Necessary toolkit installation
forlinx@ubuntu:~$ sudo apt-get install repo git ssh make gcc libssl-dev liblz4-tool
˓→ expect g++ patchelf chrpath gawk texinfo chrpath diffstat binfmt-support qemu-user-
˓→ static live-build bison flex fakeroot cmake gcc-multilib g++-multilib unzip device-
(continues on next page)
```

(continued from previous page)

```
↪tree-compiler python-pip libncurses5-dev
forlinx@ubuntu:~$ sudo apt-get install -y expect time build-essential kmod python3.9
```

The following libraries also need to be installed when using the Network Configuration Tool and menuconfig:

```
forlinx@ubuntu:~$ sudo apt-get update                                //Update the download_
↪source information
forlinx@ubuntu:~$ sudo apt-get install libncurses*                  //For building text-
↪based user interfaces
forlinx@ubuntu:~$ sudo apt-get install net-tools                   //Network_
↪configuration tool
```

3.3 Qt Creator Installation

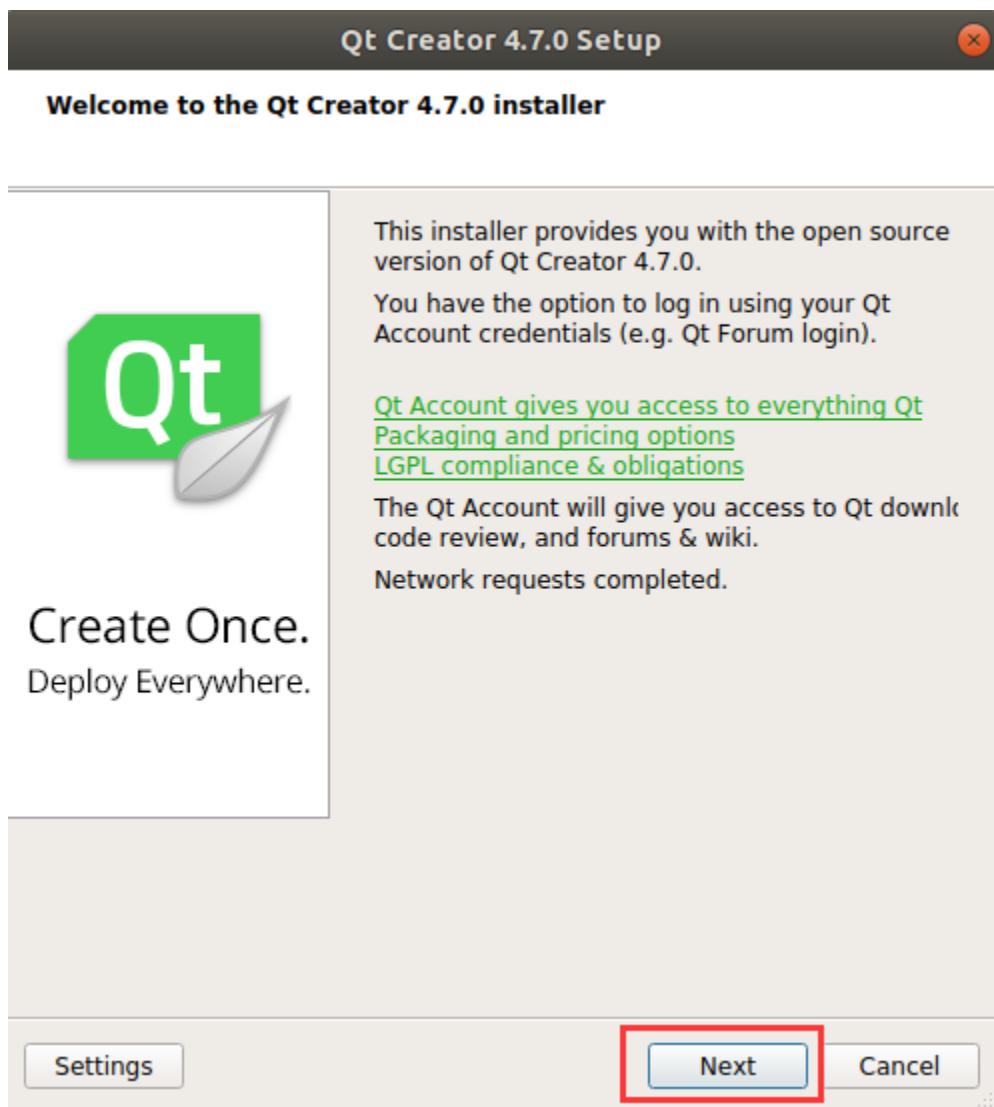
Copy qt-creator-opensource-linux-x86_64-4.1.0.run to any directory within the current user's home directory, and then run the following command.

Path: OK3588-C-C (Linux) user's profile\Linux\source code\qt-creator-opensource-linux-x86_64-4.7.0.run

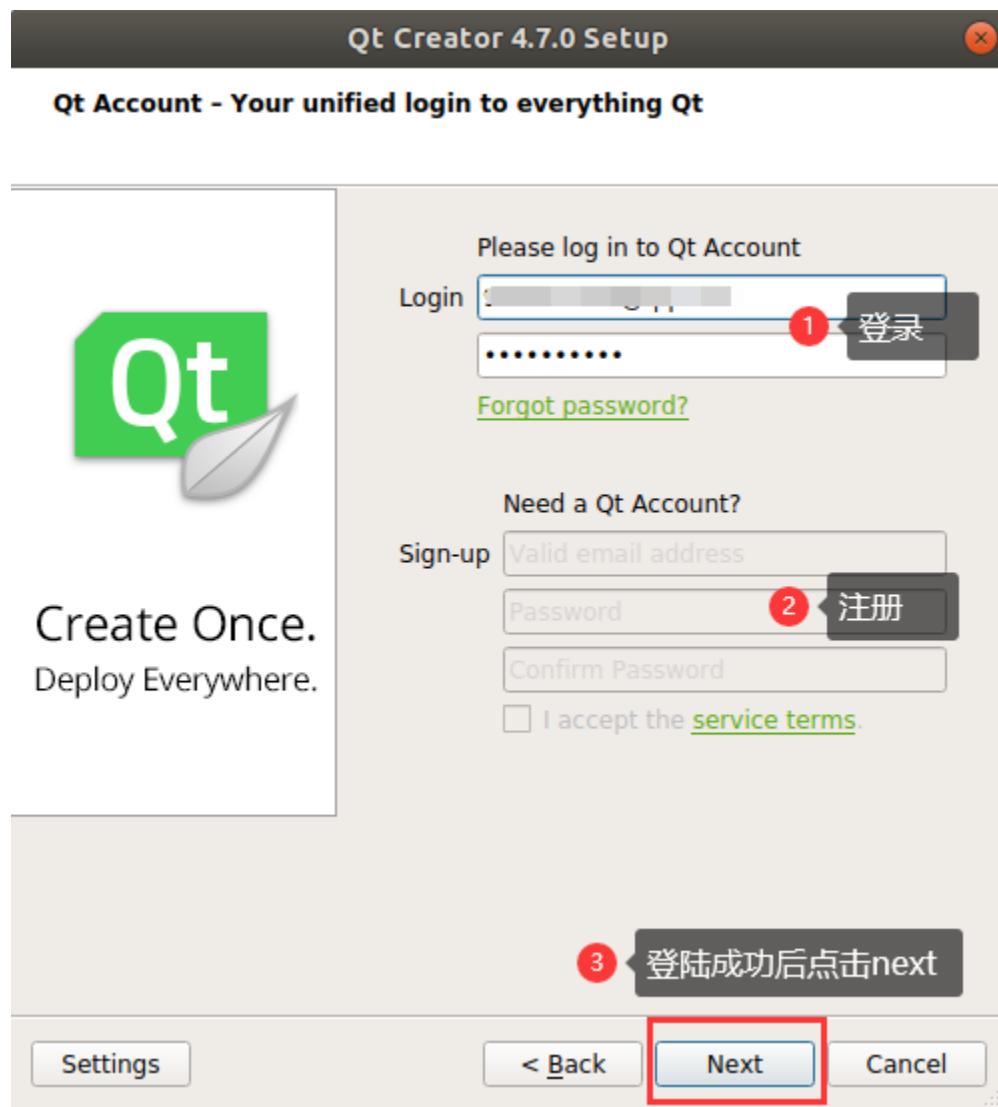
```
forlinx@ubuntu:~/3588$ ./qt-creator-opensource-linux-x86_64-4.7.0.run
```

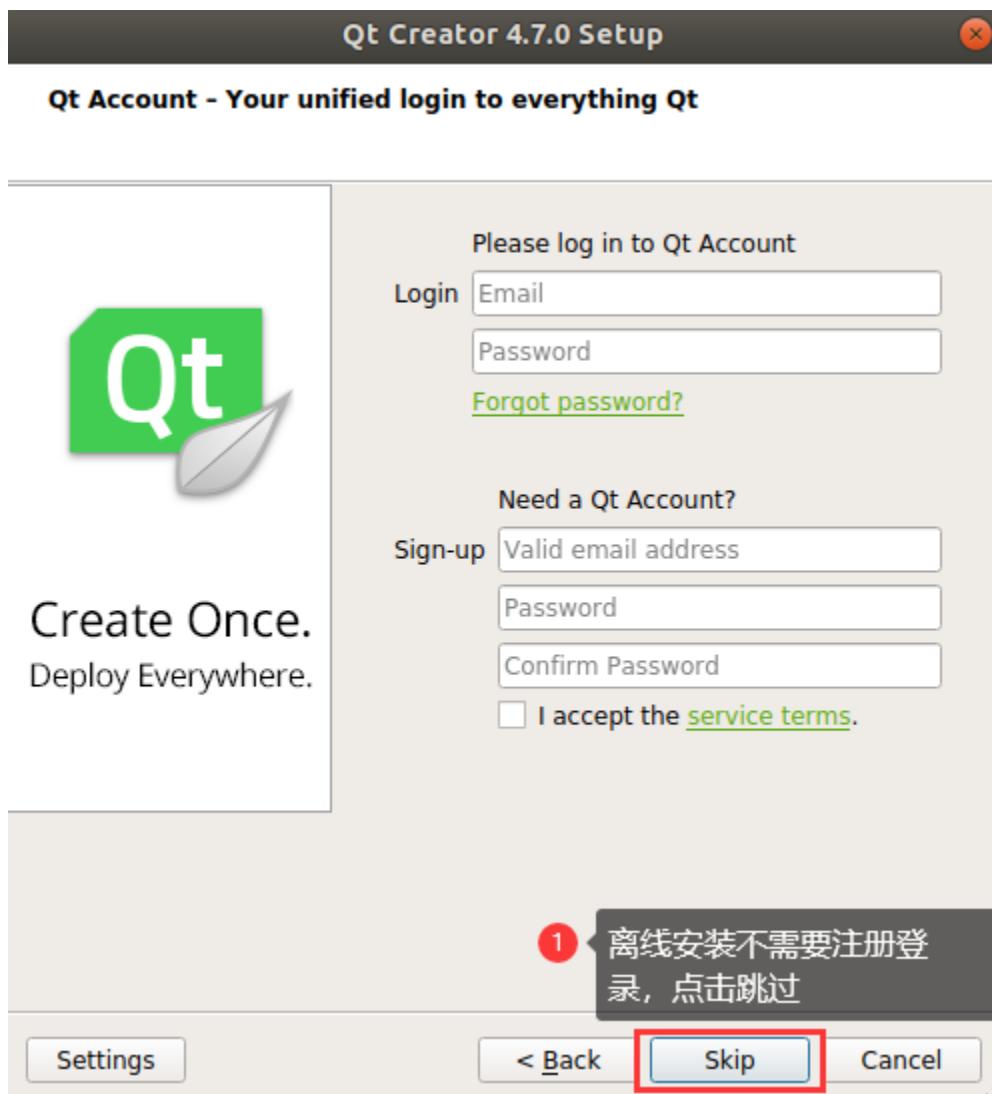
```
forlinx@ubuntu20:~/3588$ ls
qt-creator-opensource-linux-x86_64-4.7.0.run
forlinx@ubuntu20:~/3588$ █
```

Then the installation window of the graphical interface will pop up, and install according to the instructions:

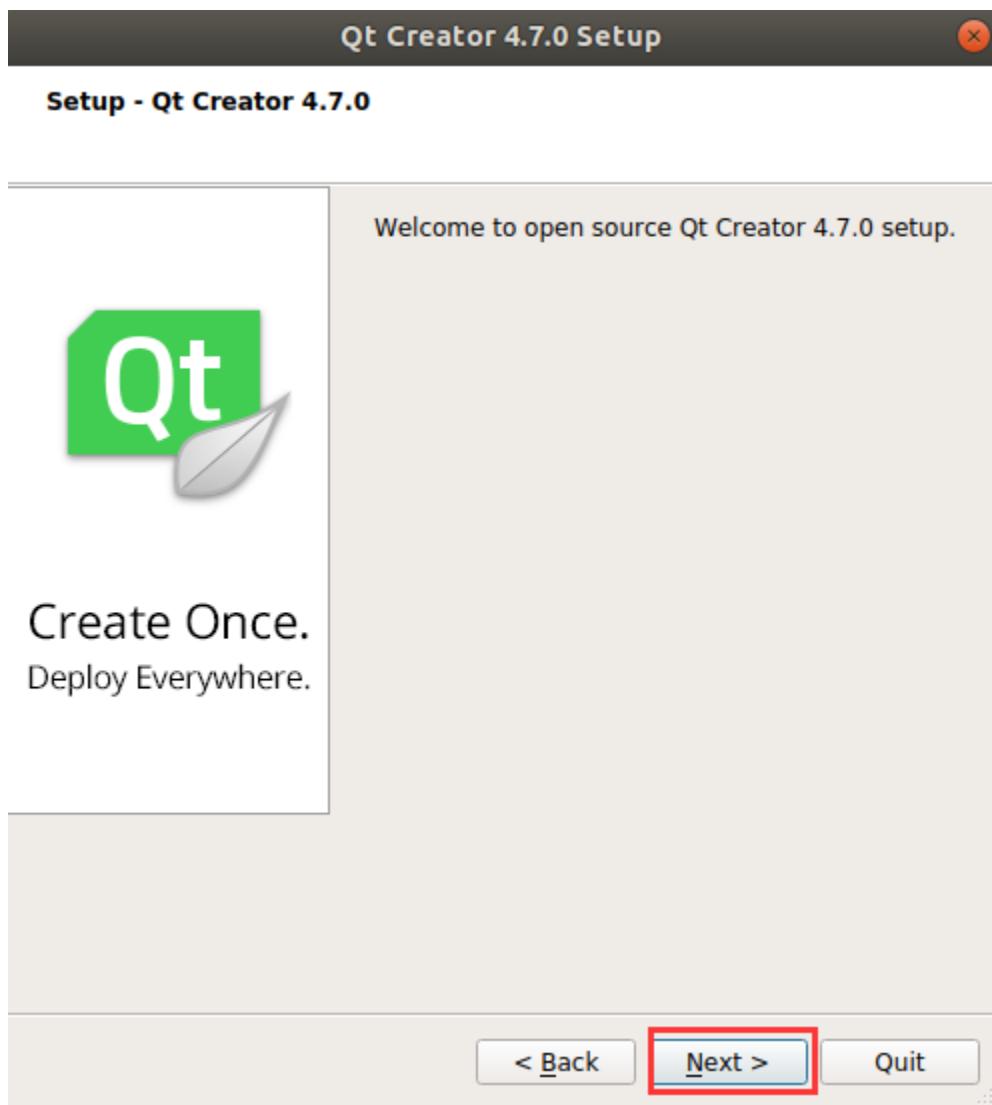




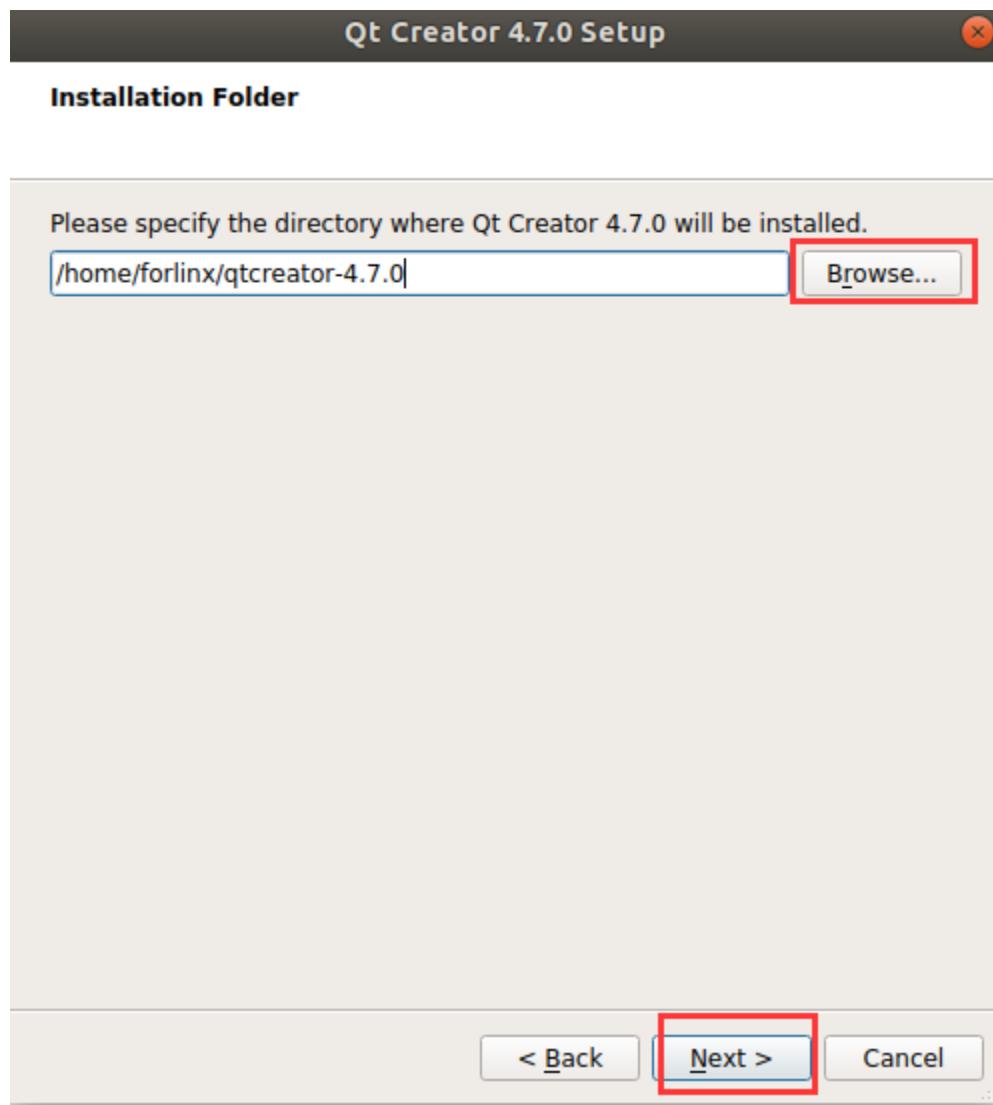




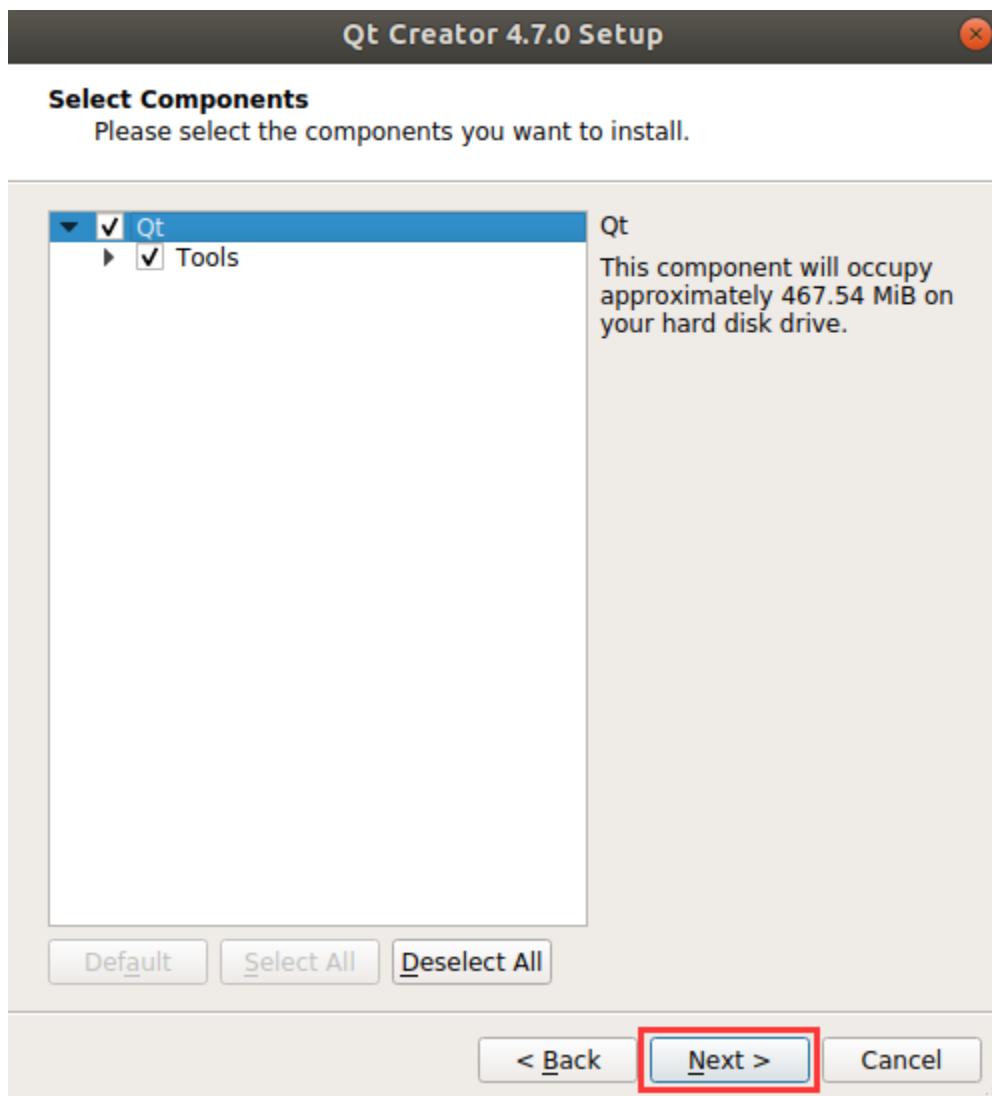
For online installation, register a Qt account (log in directly if you already have one). Password must include capital letters, uppercase, lowercase letters and numbers. After successful registration/login, click next. Skip this step for offline installation.



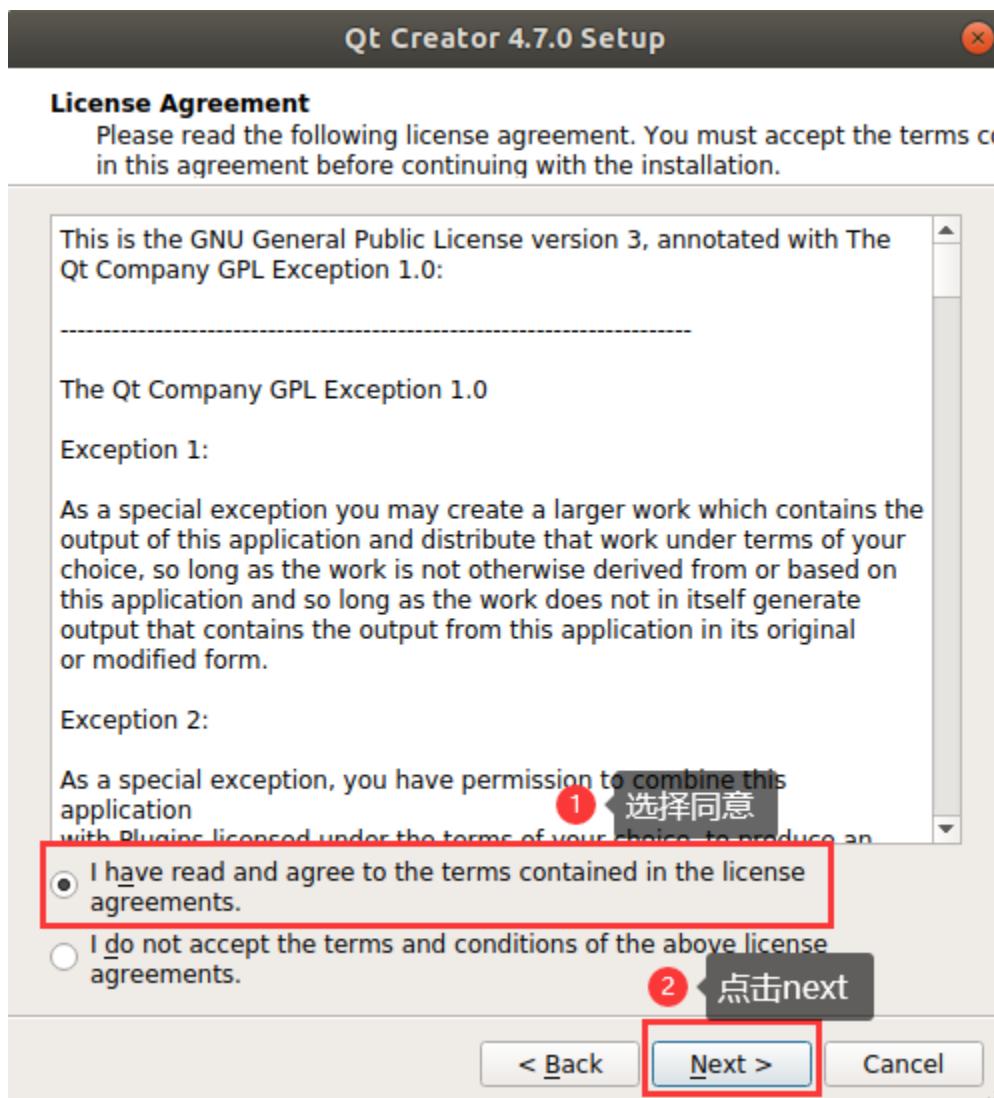
Click “Next” .

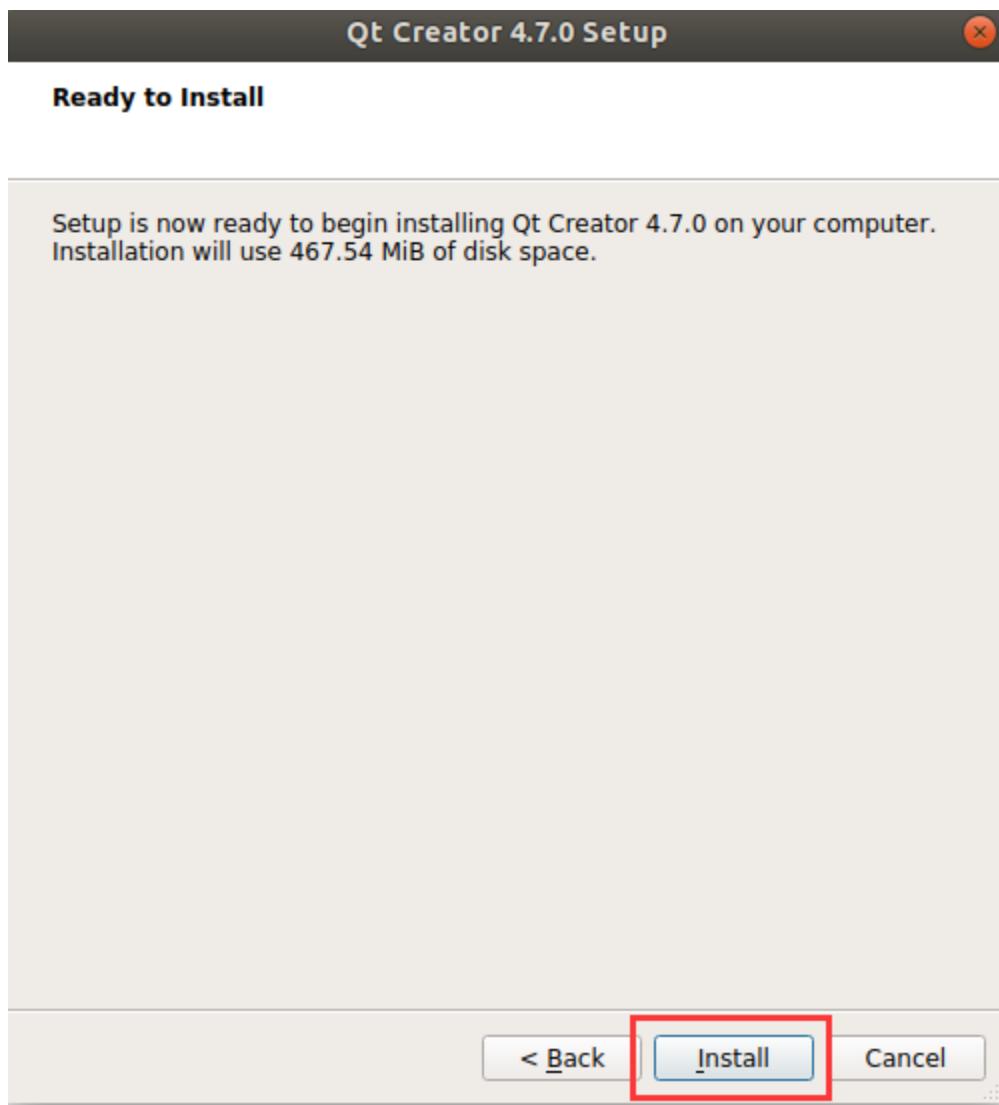


You can set the installation path according to your own habits. It is set by default here, so click “Next” .

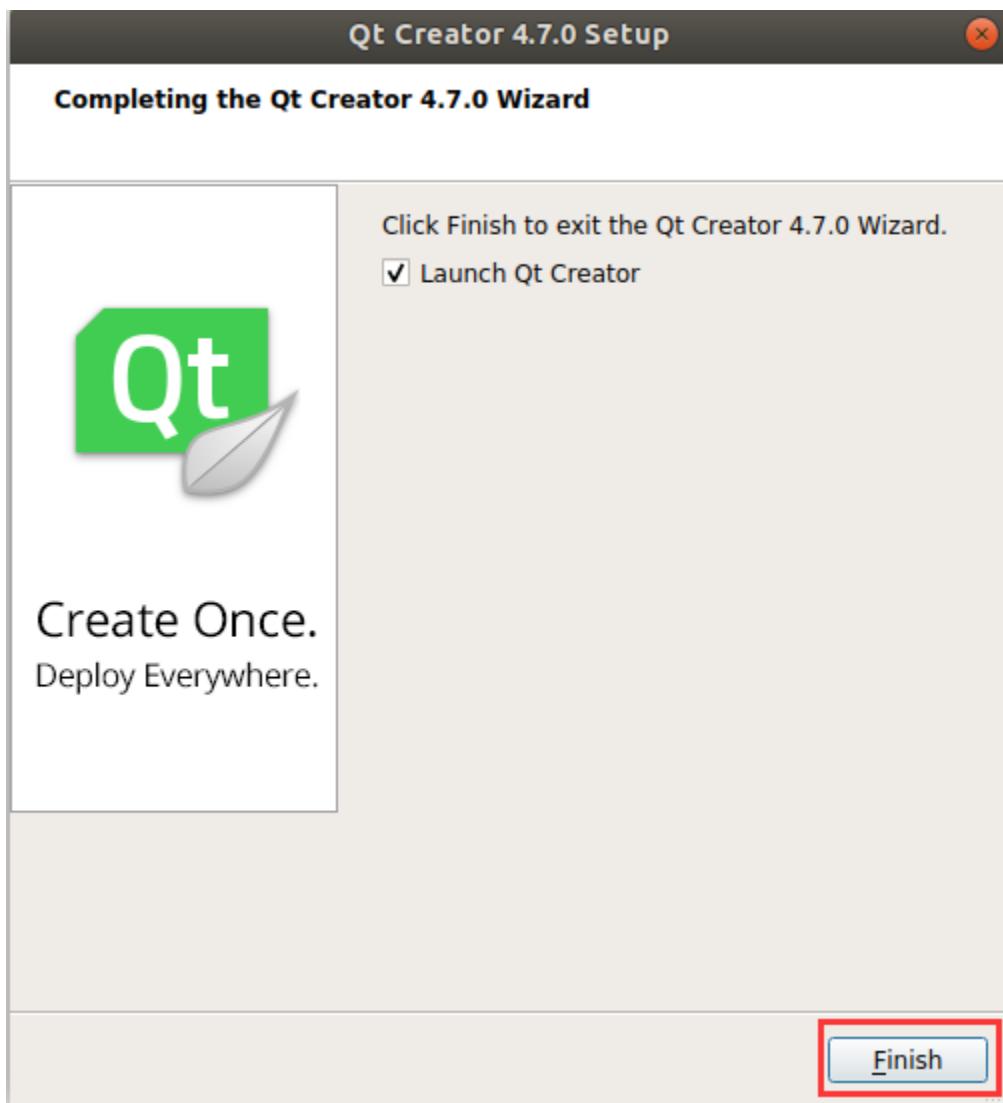


Installation completes, click “Next” .



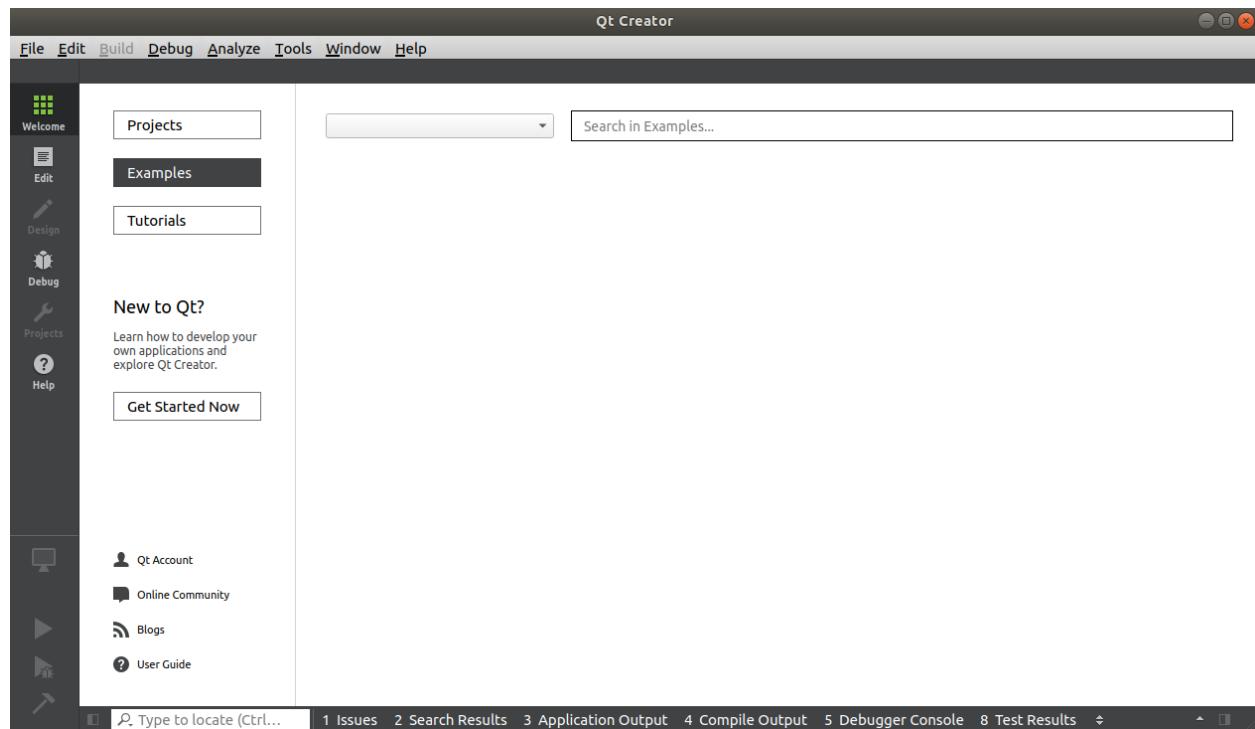


Click “Install” and wait for the installation to complete.



At this time, the Qt interface will be opened automatically. You can also start it through the command line. Execute the following command to open Qt Creator in the backstage. When you open it, the actual installation path shall prevail:

```
forlinx@ubuntu:~$ cd /home/forlinx/qtcreator-4.7.0/bin  
forlinx@ubuntu:~$ ./qtcreator &
```



The Qt Creator tool screen appears. Qt Creator is installed.

2.2.8 4. Related Code Compilation

This chapter mainly describes the compiling method of the source code related to the development board, including the kernel source code compilation and the application program compilation.

4.1 Preparation Before Compilation

4.1.1 Description of the Environment

- Development environment OS: Ubuntu 22.04 64-bit (R1-R3), Ubuntu 22.04 64-bit (R4);
- The board uses the Bootloader version: u-boot-2017.09;
- Development board kernel: linux-5.10.160 (R1-R3) \ linux-5.10.209 (R4).

4.1.2 Source Code Copy

Source Code: User Information \ Linux \ Source Code \OK3588_Linux_fs.tar.bz2.0*

Create a working directory.

```
forlinx@ubuntu:~$ mkdir -p /home/forlinx/3588 // Create the working directory in sequence
Copy the source code files `OK3588_Linux_fs.tar.bz2.0*` from the user documentation to the `/home/forlinx/3588` directory of the virtual machine.
forlinx@ubuntu:~$ cd /home/forlinx/3588 // Switch to the working directory
forlinx@ubuntu:~/3588$ cat OK3588_Linux_fs.tar.bz2.0* > OK3588_Linux_fs.tar.bz2
forlinx@ubuntu:~/3588$ tar -xvf OK3588_Linux_fs.tar.bz2 // Extract the compressed package at the current location
```

Just run the command and wait for it to complete.

4.2 Source Code Compilation

Note:

- After the kernel source code is decompressed for the first time, the source code needs to be compiled as a whole;
- R1-R3 source code compilation requires root permissions. The following operations use the forlinx user to call sudo permissions, and the password is also forlinx;
- The source code of R4 needs to be compiled as a forlinx user, and sudo cannot be added;
- After compiling as a whole, you can compile separately according to the actual situation;
- The source code compilation requires a development environment with a running memory of 8G or above. Please do not modify the VM virtual machine image configuration provided by us.

4.2.1 Full Compilation Test

In the source code path, the compilation script build.sh is provided. Run the script to compile the entire source code. You need to switch to the decompressed source code path at the terminal and find the build.sh file.

```
forlinx@ubuntu:~$ cd /home/forlinx/3588/OK3588_Linux_fs
```

The following operations need to be operated under the source code directory, and the full compilation method is:

1. Generate the configuration required for compilation (the forlinx user password is required at this time, and the password is also forlinx);

PS: linux5.10.209 (R4) kernel doesn't need this step.

```
forlinx@ubuntu: ~/3588/OK3588-linux-source$ ./build.sh BoardConfig-ubuntu-ok3588.mk
```

```
forlinx@ubuntu20:~/3588/OK3588_Linux_fs$ ./build.sh BoardConfig-ubuntu-ok3588.mk
switching to board: /home/forlinx/3588/OK3588_Linux_fs/device/forlinx/ok3588/BoardConfig-ubuntu-ok3588.mk
forlinx@ubuntu20:~/3588/OK3588_Linux_fs$
```

2. Full Compilation Test

- Full Compilation (R1-R3)

```
forlinx@ubuntu: ~/3588/OK3588-linux-source$ sudo ./build.sh
```

- Full Compilation (R4)

Note: Do not use sudo to compile the R4 materials. You don't need to use sudo for subsequent compilations either. Otherwise, you may encounter permission errors.

:::color2 If you are using linux5.10.209 R4 kernel, you need to delete the defconfig file in the output/ directory first.

:::

```
forlinx@ubuntu:~$ cd /home/forlinx/3588/OK3588-linux-source
forlinx@ubuntu: ~/3588/OK3588-linux-source$ rm output/defconfig
```

```
forlinx@ubuntu: ~/3588/OK3588-linux-source$ ./build.sh
```

When compiling for the first time, the following interface will appear. Compile Ubuntu and select 2:

```
Log colors: message notice warning error fatal

Log saved at /home/forlinx/3588/OK3588-linux-source/output/sessions/2025-01-20_09-45-46
Switching to chip: ok3588
Pick a defconfig:

1. OK3588_C_buildroot_defconfig
2. OK3588_C_ubuntu22_defconfig
Which would you like? [1]: 2
```

After successful compilation, the system image will be generated under the rockdev folder, as shown in the following figure:

```
forlinx@ubuntu20:~/3588/OK3588_Linux_fs/rockdev$ pwd
/home/forlinx/3588/OK3588_Linux_fs/rockdev
forlinx@ubuntu20:~/3588/OK3588_Linux_fs/rockdev$ ls
boot.img  MiniLoaderAll.bin  misc.img  oem.fs  oem.img  parameter.txt  recovery.img  rootfs.img  uboot.img
forlinx@ubuntu20:~/3588/OK3588_Linux_fs/rockdev$
```

Note: The update. img is packaged for full programming of OTG or TF card, and other files are programmed step by step.

4.2.2 Separate Compilation

Perform the operation in the kernel source code path.

Note: Do not use sudo to compile the R4 materials. You don't need to use sudo for subsequent compilations either. Otherwise, you may encounter permission errors.

```
forlinx@ubuntu: ~/3588/OK3588_Linux_fs$ sudo ./build.sh kernel
```

```
Hash value: 6e421d7ad8b0c2e7a50a0868309aca67e9866ec9831a5dbdbc6d9ed2695db6ec
Image 2 (resource)
Description: unavailable
Created: Sat Mar 18 16:54:52 2023
Type: Multi-File Image
Compression: uncompressed
Data Size: 893952 Bytes = 873.00 KiB = 0.85 MiB
Hash algo: sha256
Hash value: 0b38c222aba18364c57d95f6b73e86addaf3191dd4f0baaf683aa0bee5da51f1
Default Configuration: 'conf'
Configuration 0 (conf)
Description: unavailable
Kernel: kernel
FDT: fdt
Not Found io-domains in kernel/arch/arm64/boot/dts/rockchip/OK3588-C-Linux.dts
Running build_kernel succeeded.
forlinx@ubuntu20:~/3588/OK3588_Linux_fs$ 
```

The kernel in the update. img is not updated after successful compilation. Please flash the kernel/boot. img file step by step.

4.2.3 Clearing Compilation File

Perform the operation in the source code path.

Note: Do not use sudo to compile the R4 materials. You don't need to use sudo for subsequent compilations either. Otherwise, you may encounter permission errors.

```
forlinx@ubuntu: ~/3588/OK3588_Linux_fs$ sudo ./build.sh cleanall
```

```
forlinx@ubuntu20:~/3588/OK3588_Linux_fs$ sudo ./build.sh cleanall
processing option: cleanall
clean uboot, kernel, rootfs, recovery
make: Entering directory '/home/forlinx/3588/OK3588_Linux_fs/u-boot'
make: Nothing to be done for 'distclean'.
make: Leaving directory '/home/forlinx/3588/OK3588_Linux_fs/u-boot'
make: Entering directory '/home/forlinx/3588/OK3588_Linux_fs/kernel'
  CLEAN  certs
  CLEAN  drivers/firmware/efi/libstub
  CLEAN  drivers/misc/lkdtm
  CLEAN  drivers/scsi
  CLEAN  drivers/tty/vt
fatal: not a git repository (or any of the parent directories): .git
  CLEAN  fs/unicode
  CLEAN  kernel
  CLEAN  lib
  CLEAN  net/wireless
  CLEAN  usr
/bin/sh: 1: ./scripts/gcc-wrapper.py: not found
/bin/sh: 1: ./scripts/gcc-wrapper.py: not found
  CLEAN  vmlinux.symvers modules-only.symvers modules.builtin modules.builtin.modinfo
  CLEAN  scripts/basic
  CLEAN  scripts/dtc
  CLEAN  scripts/kconfig
  CLEAN  scripts/mod
  CLEAN  scripts
  CLEAN  include/config include/generated .config .version Module.symvers
make: Leaving directory '/home/forlinx/3588/OK3588_Linux_fs/kernel'
Running build_cleanall succeeded.
forlinx@ubuntu20:~/3588/OK3588_Linux_fs$
```

This operation clears all intermediate files. However, it does not affect the source files, including those that have already had changes made to them.

4.3 Image File Use

The update. img is packaged for full programming of OTG or TF card, and other files are programmed step by step. The Image file generated by separate compilation will not be updated in the update. img file, and it needs to be flashed by single-step. (see the software manual OTG flashing for details).

4.4 Application Compilation and Operation

4.4.1 Command Line Application Compilation and Operation

In this section, use the “hello world” test program and compile it in the development environment.

1. Set up the Ubuntu application development environment;

Install qemu and the corresponding tools.

```
forlinx@ubuntu:~$ sudo apt install qemu-user-static -y
forlinx@ubuntu:~$ sudo apt-get install git ssh make gcc libssl-dev liblz4-tool
```

2. Mount the Ubuntu system image

Note: You can add modification records or back up the Ubuntu system image before making any changes.

```
forlinx@ubuntu: ~/3588/OK3588-linux-sourc/ubuntu$ mkdir rootfs
forlinx@ubuntu: ~/3588/OK3588_Linux_sourc/ubuntu$ sudo jammy-rootfs.img rootfs
forlinx@ubuntu: ~/3588/OK3588_Linux_sourc/ubuntu$ cd rootfs/
```

3. Reset the root directory;

Set the root file system mounted from the Ubuntu image as the root directory.

```
forlinx@ubuntu: ~/3588/OK3588_Linux_sourc/ubuntu/rootfs$ sudo chroot ./
root@ubuntu20:/# ls
bin  boot  dev  etc  home  lib  lost+found  media  mnt  opt  proc  root  run  sbin  ↵
snap  srv  swapfile  sys  tmp  usr  var
```

4. Add the application hello_world.c for compilation

You can place it in the required location of this file system.

```
root@ubuntu20:/# vi hello_world.c
root@ubuntu20:/# cat hello_world.c
#include <stdio.h>

int main(){
    printf("hello world!\n");
    return 0;
}
root@ubuntu20:/# gcc hello_world.c -o hello_world
root@ubuntu20:/# ./hello_world
hello world!
root@ubuntu20:/# file hello_world
hello_world: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically-linked, interpreter /lib/ld-linux-aarch64.so.1, ↵
BuildID[sha1]=2ad869961eba250292af95bedbbf68a4b32fa0ce, for GNU/Linux 3.7.0, not ↵
stripped
```

Note: You can use the Ctrl + D key combination to exit the chroot environment.

You can see the program can be executed in the current command line. Using the file command, you can also see that the hello_world program is now of the ARM aarch64 architecture, which is also what the Ubuntu system requires.

5. Program transplantation.

- Merge into the file system

Unmount the rootfs. The modified content will be automatically merged into the system image of the file system. Then, perform a full recompilation.

```
forlinx@ubuntu: ~/3588/OK3588_Linux_source/ubuntu/$ sudo umount rootfs
forlinx@ubuntu: ~/3588/OK3588_Linux_source/$ ./build.sh
```

- Copy the executable program to the development board separately

Copy the executable program to a USB flash drive and then transfer it from the USB flash drive to the development board.

2.3 Android 12.0 User's Manual_V 1.3

Document classification: Top secret Secret Internal information [? Open](#)

2.3.1 Copyright Notice

The copyright of this manual belongs to Baoding Folinx Embedded Technology Co., Ltd. Without the written permission of our company, no organizations or individuals have the right to copy, distribute, or reproduce any part of this manual in any form, and violators will be held legally responsible.

Forlinx adheres to copyrights of all graphics and texts used in all publications in original or license-free forms.

The drivers and utilities used for the components are subject to the copyrights of the respective manufacturers. The license conditions of the respective manufacturer are to be adhered to. Related license expenses for the operating system and applications should be calculated/declared separately by the related party or its representatives.

2.3.2 Overview

This manual is designed to help users quickly familiarize themselves with the product, and understand the interface functions and testing methods. It primarily covers the testing of interface functions on the development board, the methods for flashing images, and troubleshooting procedures for common issues encountered in use. In the process of testing, some commands are annotated to facilitate the user's understanding, mainly for practical use. Please refer to "OK3588—C_Android12 User's Compilation Manual" provided by Forlinx for kernel compilation, related application compilation methods, development environment construction, etc.

The manual is primarily divided into five chapters:

- Chapter 1. focuses on the overall overview of the product, and briefly introduces the development board in the interface resources, the relevant driver path in the kernel source code, and the description of the key parts of the information;
- Chapter 2. mainly focuses on the fast booting of the product, which can be achieved through two methods: serial port login and network login, as well as the relevant introduction of the U-Boot menu;
- Chapter 3. Android Function Test;
- Chapter 4. focuses on the product's image update, mainly describing the method of updating the image to the storage device, and users can choose the corresponding flashing method according to the actual situation;
- Chapter 5. is mainly about the OTA upgrade test of the product system.

A description of some of the symbols and formats associated with this manual:

Format	Meaning
Note	Note or information that requires special attention, be sure to read carefully
?	Relevant notes on the test chapters
?	Indicates the related path.
Blue font on gray background	Refers to commands entered at the command line(Manual input required).
Black font	Serial port output message after entering a command
Bold black	Key information in the serial port output message
//	Interpretation of input instructions or output information
User-name@Hostname	console: Development board serial port login account information, which can be used to determine the environment for function operation

After packaging the file system, you can use the "ls" command to view the generated files.

```
forlinx@ubuntu:~/3588$ ls                                //List the files in this_
˓→directory
OK3588-android-source  OK3588-android-source.tar.bz2
```

- forlinx@ubuntu: the username is forlinx and the hostname is ubuntu, indicating that the operation is performed in the development environment ubuntu;
 - //: Explanation of the instruction, no input required;
 - Ls: Blue font on a gray background, indicating relevant commands that need to be entered manually;
 - OK3588-android-source: Black font is the output information after entering the command; bold font is the key information; here is the packaged file system.
-

2.3.3 Application Scope

This manual is mainly applicable to the Android12.0 operating system on the Forlinx OK3588-C platform. Other platforms can also refer to it, but there will be differences between different platforms. Please make modifications according to the actual conditions.

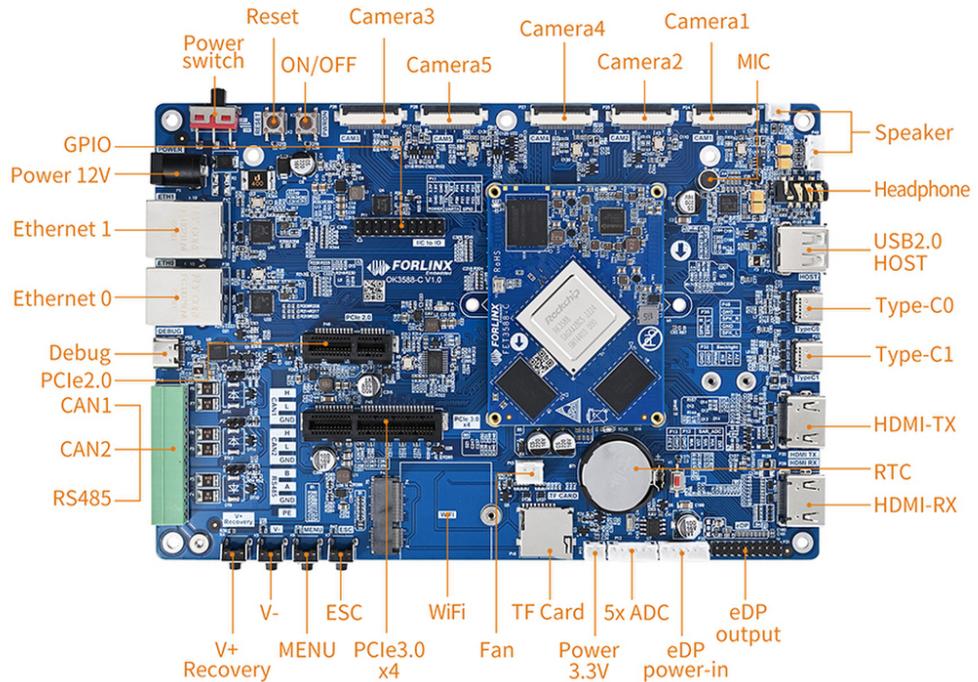
2.3.4 Revision History

Date	User Manual Version	SoM Version	Carrier Board Version	Revision History
01/12/2	V1.0	V1.1	V1.1	OK3588-C Android12.0 User's Manual Initial Version
04/11/2	V1.1	V1.1	and Above	1. Adding a description of flashing separate image in OTG flashing chapter; 2. Modifying maskroom flashing description.
16/01/2	V1.2	V1.1	and Above	Correcting the description regarding the OV5645; it now supports the OV5645 camera.
21/03/2	V1.3	V1.1	and Above	Adding CPU/GPU/NPU frequency description.

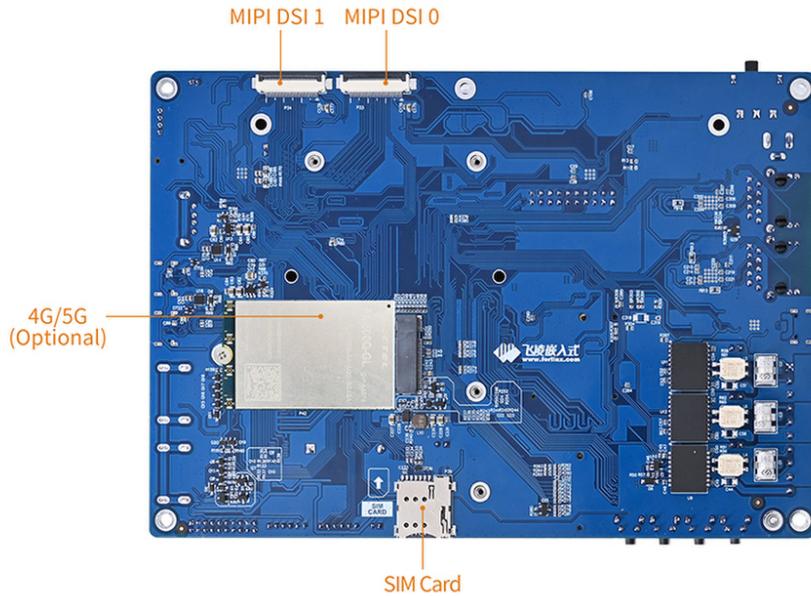
2.3.5 1. OK3588 Development Board Description

RK3588 is a low-power, high-performance processor based on ARM64 architecture, which includes 4-core Cortex-A55 and 4-core Conrtex-A76 as well as independent NEON processor and neural network processor NPU, and it can be applied to computers, cell phones, personal mobile Internet, and digital multimedia devices.

Connection method is board-to-board, and main interfaces are shown in the figure below:



Front



Back

Note: This software manual does not describe the hardware parameters. Before referring to this manual for software development, please read the “OK3588-C Hardware Manual _ V1.0” under “Hardware Data/User Manual” (the download method is the same as the software data) to understand the product naming rules and the hardware configuration of the product, which is helpful for you to use it.

1.2 CPU/GPU/NPU Frequency Description

RK3588J industrial grade SoM frequencies are described below:

**Note: For the industrial-grade RK3588J SoM, to better test the maximum performance of this SOC, starting from version R4 and subsequent versions, the SoM in the user materials will default to operate in overclocking mode (Without performance requirements, it is recommended to modify it to the normal mode). **

Refer to “Rockchip RK3588J Datasheet V1.1-03/08/2023.pdf ”

Table 3-2 Recommended operating conditions

Maximum CPU A76 frequency, normal mode ①	1.6GHz
Maximum CPU A76 frequency, overclocking mode ②	2.0GHz
Maximum CPU A55 frequency, normal mode ①	1.3GHz
Maximum CPU A55 frequency, overclocking mode ②	1.7GHz
Maximum GPU frequency, normal mode ①	700MHz
Maximum GPU frequency, overclocking mode ②	850MHz
Maximum NPU frequency, normal mode ①	800MHz
Maximum NPU frequency, overclocking mode ②	950MHz

1. Normal mode indicates that the chip is operating at a safe voltage and frequency. For industrial environments, it is highly recommended to keep it in normal mode to reasonably ensure longevity;
2. Overclocking mode will bring higher frequency, and the corresponding voltage will also increase. Running in overclocking mode in a long time may shorten the chip’s life, especially at high temperatures.

To switch to “normal mode”, you need to add # include “rk3588j.dtsi” to the reference in the kernel device tree. The path is:

OK3588-android-source/kernel-5.10/arch/arm64/boot/dts/rockchip/OK3588-C-common.dtsi

RK3588 commercial grade SoM frequencies are described below:

Refer to “Rockchip RK3588 Datasheet V1.7-17/11/2023.pdf”

Table 3-2 Recommended operating conditions

Maximum CPU A76 frequency	2.2-2.4 GHz
Maximum CPU A55 frequency	1.8GHz
Maximum GPU frequency	1GHz
Maximum NPU frequency	1GHz

1.3 Android12 System Software Resources Features

Device	Location of driver source code in the kernel	Device Name
LCD Backlight Driver	drivers/video/backlight/pwm_bl.c	/sys/class/backlight
USB Port	drivers/usb/storage/	
USB Mouse	drivers/hid/usbhid/	/dev/input/mice
Ethernet	drivers/net/ethernet/stmicro/stmmac	
SD/micro TF card driver	drivers/mmc/host/dw_mmc-rockchip.c	/dev/block/mmcblk1pX
EMMC Driver	drivers/mmc/host/dw_mmc-rockchip.c	/dev/block/mmcblk2pX
OV13850	drivers/media/i2c/ov13850.c	/dev/videoX
LCD Controller	drivers/gpu/drm/rockchip/rockchip_drm_vop.c	
MIPI CSI	drivers/phy/rockchip/phy-rockchip-mipi-rx.c	
MIPI DSI	drivers/phy/rockchip/phy-rockchip-inno-mipi-dphy.c	
LCD Touch Driver	drivers/input/touchscreen/goodix.c drivers/input/touchscreen/etft5x06.c	/dev/input/eventX
RTC Real Time Clock Driver	drivers/rtc/rtc-rx8010.c drivers/rtc/rtc-pcf8563.c	/dev/rtc0
serial port	drivers/tty/serial/8250/8250_dw.c	/dev/ttySX
Key Driver	drivers/input/keyboard/adc-keys.c	/dev/input/eventX
LED	drivers/leds/leds-gpio.c	
I2S	sound/soc/rockchip/rockchip_i2s.c	
Audio Driver	sound/soc/codecs/rk817_codec.c	/dev/snd/
PMIC	drivers/mfd/rk808.c	
PCIE	drivers/pci/controller/pcie-rockchip.c	
Watchdog	drivers/watchdog/dw_wdt.c	
SPI	drivers/spi/spi-rockchip.c	
PWM	drivers/video/backlight/pwm_bl.c	

1.4 eMMC Memory Partition Table

The following table is the eMMC memory partition information of Android operating system (the size of a block is 512bit when calculating):

Partition Index	Name	Offset / block	Size/block	content
N/A	security	0x00000000	0x00004000	MiniLoaderAll.bin
1	uboot	0x00004000	0x00004000	uboot.img
2	trust	0x00008000	0x00002000	misc.img
3	dtbo	0x0000a000	0x00002000	dtbo.img
4	vbmeta	0x0000c000	0x00000800	vbmeta.img
5	boot	0x0000c800	0x0001b800	boot.img
6	recovery	0x00020800	0x00030000	recovery.img
7	baseparameter	0x001d8800	0x00000800	baseparameter.img
8	super	0x001d9000		super.img

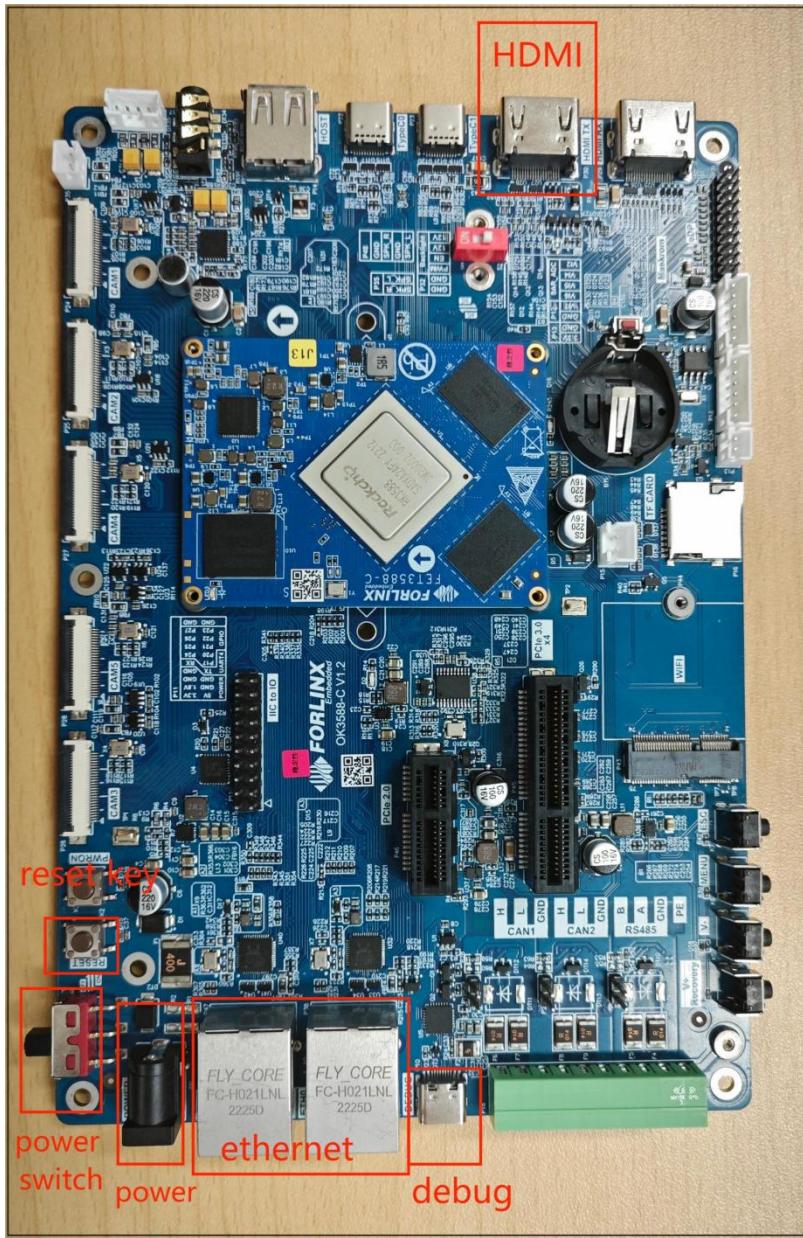
2.3.6 2. Fast Startup

2.1 Preparation Before Startup

The OK3588 development board has two system login methods, serial and network login. Hardware preparation before system startup:

- 12V2A or 12V3A DC Power Cable
- Debugging serial cable

The debug serial port on the development board is a Type-C socket, allowing users to connect the development board to a PC using a USB to Type-C cable to monitor the status of the development board.



2.2 Debugging Serial Driver Installation

The debugging serial port of the OK3588 - C platform uses a Type - C interface. There is an on - board USB to UART chip, so there's no need to purchase a USB to serial port debugging tool. It is extremely simple and convenient to use.

To install the driver, please use the driver package `DriverAssitant_v5.1.1.zip` provided in the User Profile `\Android\Tools\` directory.

Run `DriverInstall.exe` directly after the unzipping is completed; in order to ensure the driver is the latest version, please uninstall the driver first, then install again.

2.3 Serial Port Login

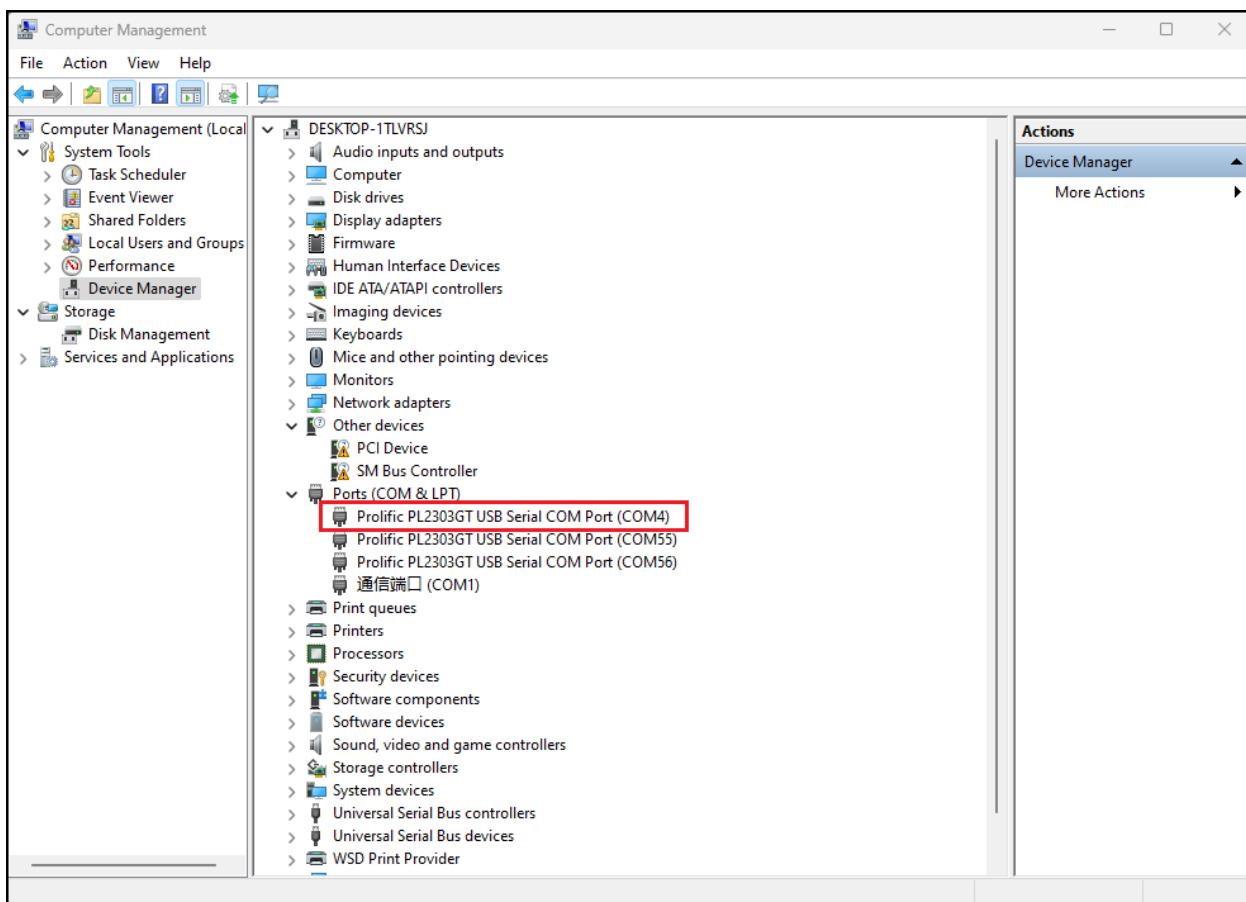
2.3.1 Serial Port Connection Settings

Note:

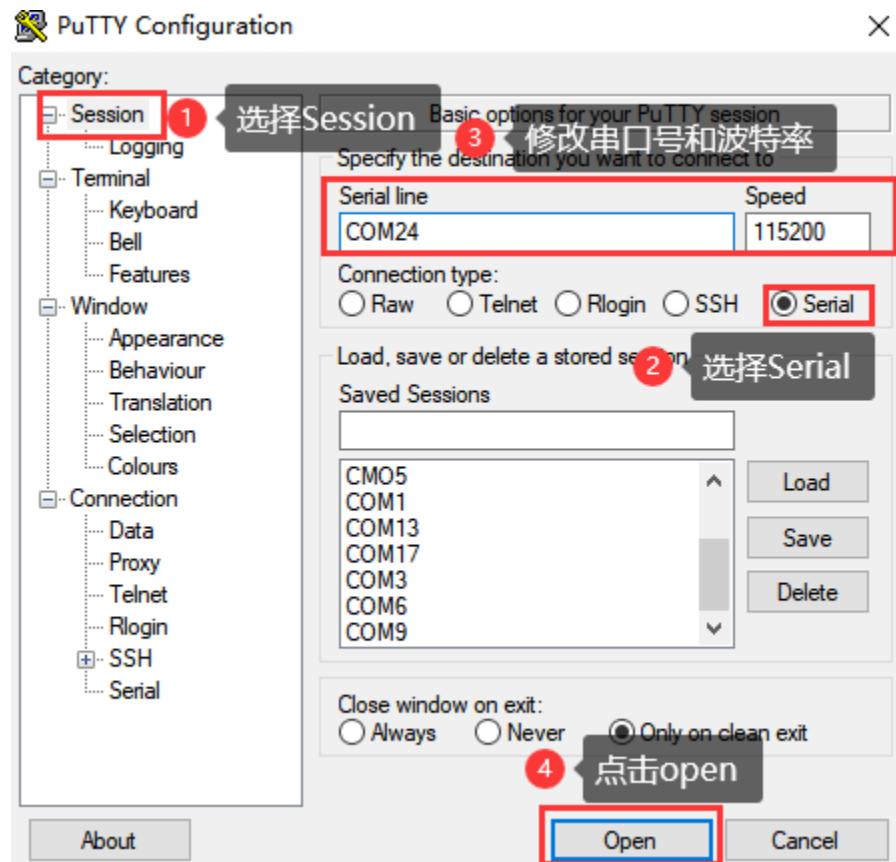
- **Serial port terminal login user:** Serial port terminal automatically logs in root user without password;
- **Serial port settings:** baud rate 115200, data bit 8, stop bit 1, no parity bit, no flow control;
- **Hardware Requirements:** Type-C cable required to connect PC and development boards;
- **Software requirements:** PC Windows system needs to install the super terminal software. Because the terminal software has many types, users can choose their familiar one.

In the following, we take the putty terminal software as an example to introduce the serial port login method:

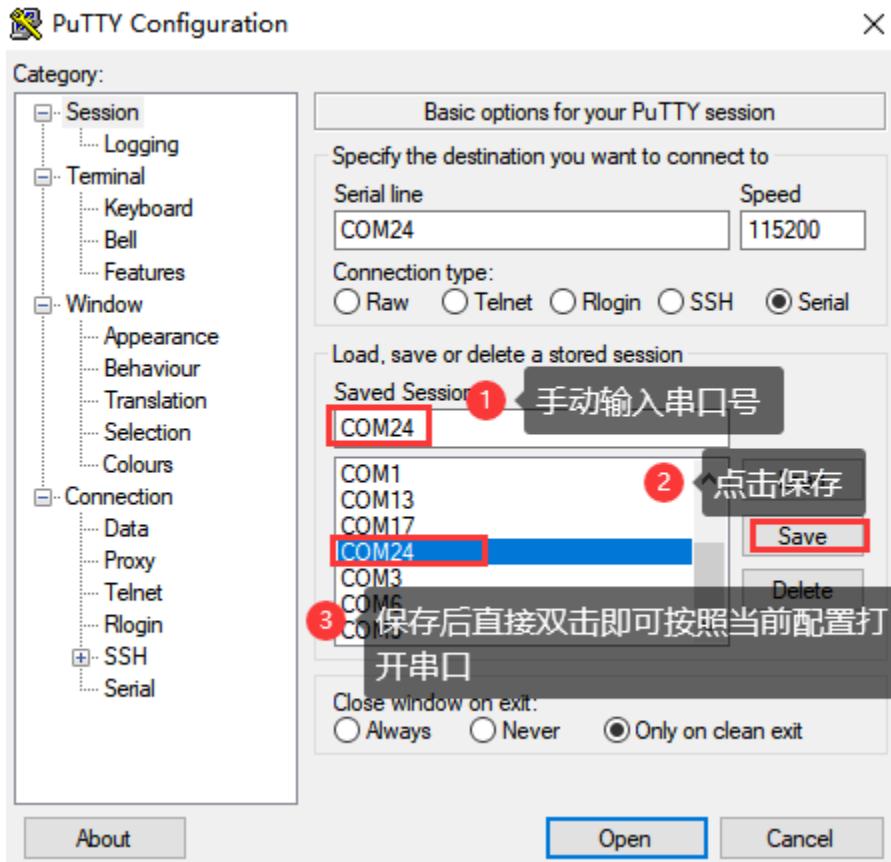
Step 1: Connect the serial port number of the computer—check the serial port number from the device manager (Based on the port actually recognized by the computer);



Step 2: Open and set up putty, then set the “line according to the COM port of the computer used, baud rate 115200;



Step 3: After the setting, input the COM port used by the computer in Saved Sessions. The following figure takes COM3 as an example, save the settings, open the serial port again later, and click on the saved port number;



2.3.2 Serial Login

After the terminal software on the PC side is set, connect the PC and the development board through the serial port cable, and power on after connecting the power supply. The startup information can be seen through the terminal software.

The following startup message indicates a successful start, allowing a new command line to be entered by pressing Enter:

```
[ 13.487596] [ T422] rk_gmac-dwmac fe1c0000.ethernet eth1: configuring for phy/rgmii-rxid link mode
[ 13.491887] [ T422] rk_gmac-dwmac fe1b0000.ethernet eth0: PHY [stmmac-0:01] driver [Generic PHY] (irq=POLL)
[ 13.493224] [ T422] dwmac4: Master AXI performs any burst length
[ 13.493239] [ T422] rk_gmac-dwmac fe1b0000.ethernet eth0: No Safety Features support found
[ 13.493249] [ T422] rk_gmac-dwmac fe1b0000.ethernet eth0: IEEE 1588-2008 Advanced Timestamp supported
[ 13.493362] [ T422] rk_gmac-dwmac fe1b0000.ethernet eth0: registered PTP clock
[ 13.493580] [ T422] rk_gmac-dwmac fe1b0000.ethernet eth0: configuring for phy/rgmii-rxid link mode
[ 13.528824] [ T365] Freeing drm_logo memory: 1404K
[ 13.528902] [ T365] dwhdmi-rockchip fde80000.hdmi: use tmds mode
[ 15.484581] [ T261] binder: undelivered transaction 21624, process died.
[ 15.485969] [ T1] init: service 'bootanim' (pid 330), exited with status 0 oneshot service took 8.817000 seconds in background
[ 15.486019] [ T1] init: Sending signal 9 to service 'bootanim' (pid 330) process group...
[ 15.486230] [ T1] libprocessgroup: Successfully killed process cgroup uid 1003 pid 330 in 0ms
[ 15.688711] [ T1] init: processing action (sys.boot_completed=1) from (/system/etc/init/hw/init.rc:1148)
[ 15.954062] [ T1] zram0: detected capacity change from 0 to 3985276928
[ 15.961170] [ T1] mkswap: Swapspace size: 3891868k, UUID=82ca8c18-885b-46b4-811f-bf5e94077e94
[ 15.990018] [ T1] Adding 3891868k swap on /dev/block/zram0. Priority:-2 extents:1 across:3891868k ss
console:/ $
```

2.4 Screen Switching

OK3588 supports various screen interfaces such as MIPI DSI, HDMI, eDP, DP, RGB, etc., and can simultaneously perform mirroring and independent display for up to four screens. Currently there are three screen switching methods: u-boot menu dynamic control; kernel device tree designation; DisplayHwConfig application control.

OK3588 contains 4 display controllers, i.e. 4 VP. Supports up to 4 screens simultaneously. The maximum resolution of

VP0 is 7680x4320; the maximum resolution of VP1 is 4096x4320; the maximum resolution of VP2 is 4096x4320; the maximum resolution of VP3 is 2048x1080.

2.4.1 Uboot Menu Dynamic Control

2.4.1.1 Display Type Settings

This method allows switching without recompiling and burn-in of existing supported screens.

During the uboot self-boot process, press the space bar at the serial terminal to bring up the control options:

```
Hit key to stop autoboot('Spacebar'): 0
```

```
-----  
0:Exit to console  
1:Reboot  
2:Display type  
3:Android config  
-----
```

Enter 2 at the terminal to access the Screen Control sub-menu:

```
-----  
hdmi0 and edp0 share same port, only one can be used.  
hdmi1 and edp1 share same port, only one can be used.  
only four VPs internally, so up to four interfaces can be activated  
hdmi edp dp can only be displayed on VP0 or VP1 or VP2.  
dsi0 dsi1 can only be displayed on VP2 or VP3.  
rgb can only be displayed on VP3.
```

```
Select display  
0:Exit  
1: hdmi0 => VP0  
2: hdmi1 =>  
3: edp0 =>  
4: edp1 =>  
5: dp0 =>  
6: dp1 =>  
7: mipi0 =>  
8: mipi1 =>  
9: rgb =>  
a: primary display => HDMI0  
b: primary display resolution => 1920x1080p50  
-----
```

According to the content of the comments in the uboot menu, you can get the uboot display menu setting rules:

1. Both hdmi0 and edp0 use the same port, and only one of them can be used at the same time;
2. HDMI1 and EDP1 share the same port, and only one of them can be used at a time;
3. There are only four VPs internally, so a maximum of four interfaces can be activated;
4. HDMI, EDP, and DP can only be displayed on VP0, VP1, or VP2;
5. DSI0 and DSI1 can only be displayed on VP2 or VP3;
6. RGB can only be displayed on VP3.

When setting up the display, enter the serial number corresponding to the display interface and VP will be assigned to the corresponding interface. If you input again, the system will switch the available VP (Virtual Ports) for this port one by one, or close the VP assigned to this port.

Enter a to switch the main screen display. Only the interface assigned with VP can be set as the main screen display.

Setup Examples:

As shown in the figure below, VP is assigned to hdmi0, hdmi1, dp0 and mipi0, so the main screen display can be selected from one of the above four interfaces. Because the HDMI interface is used, the EDP interface cannot be used. The primary display also selects HDMI0 with VP0 assigned and sets the resolution to 1920x1080.

Note: When the primary screen is HDMI or DP, you need to set the “primary display resolution” to 3840x2160 and connect the primary screen before the device is powered on. Otherwise, connecting the main screen after startup will result in resolution switching on the secondary screen. When the main screen is EDP or MIPI, the primary display resolution option will not take effect.

```
-----
Select display
0:Exit
1: hdmi0 => VP0
2: hdmi1 => VP1
3: edp0 =>
4: edp1 =>
5: dp0 => VP2
6: dp1 =>
7: mipi0 => VP3
8: mipi1 =>
9: rgb =>
a: primary display => HDMI0
b: primary display resolution => 1920x1080p50
-----
```

2.4.1.2 Android Setting

At the top of the uboot settings menu, type 3 to go to the android settings menu bar.

```
-----
android_navigationbar==>off android_topsliding==>off android_root_access==>off
Select android config
0:Exit
1:android navigationbar display off
2:android topsliding display off
3:android root access status off
-----
```

Here you can set the navigation bar, status bar, and root permissions of Android. Inputting the corresponding number can toggle the on/off status of the corresponding function.

The navigation bar display and top sliding display are set to on. After entering the system, the navigation bar and the top sliding bar will be shielded. Use the right mouse button to return to the function.

With the root access status set to on, other users can also gain root privileges at the terminal command line.

2.4.2 Kernel Device Tree Specification

This method does not require the connection of a serial terminal, and the system image defaults to the desired configuration selection, which is suitable for mass production. However, we need to manually modify the device tree and regenerate the system image once again.

Note: This method has higher priority than the uboot screen selection, and the uboot selection will not take effect after the device tree is modified.

The device tree path: kernel/arch/arm64/boot/dts/rockchip/OK3588-C-common.dtsi

In the kernel source code, open the device dtsi file and find the following node:

```
/*
 * hdmi0 and edp0 share same port, only one can be used.
 * hdmi1 and edp1 share same port, only one can be used.
 * only four VPs internally, so up to four interfaces can be activated
 * hdmi edp dp can only be displayed on VP0 or VP1 or VP2.
 * dsi0 dsi1 can only be displayed on VP2 or VP3.
 * rgb can only be displayed on VP3.
 */
/**/

/** if "disabled" , display is controlled by uboot ***/
status = "disabled";

// HDMI0 = "VP0";
// EDPM0 = "OFF";
// EDP1 = "VP1";
// DP0 = "OFF";
// DP1 = "OFF";
// MIPI0 = "VP2";
// MIPI1 = "VP3";
// RGB = "OFF";
primary_display = "MIPI0";
primary_display_resolution = "1920x1080p60";
};
```

The node has a default disabled state and needs to be changed to an okay enabled node. Change according to screen requirements.

Parameter Description:

Meaning	
status	Describe the node state: disabled is for off, okay is for on
HDMI0	Specify the VP assigned to HDMI0
HDMI1	Specify the VP assigned to HDMI1
EDP0	Specify the VP assigned to EDP0
EDP1	Specify the VP assigned to EDP1
DPO	Specify the VP assigned to DPO
DP1	Specify the VP assigned to DP1
MIPIO	Specify the VP assigned to MIPIO
MIPI1	Specify the VP assigned to MIPI1
RGB	Specify the VP assigned to RGB
primary_display	Specify the main screen display

Users need to change the setting parameters as required. After saving, it is necessary to recompile and generate an image.

An annotated description of the node:

1. Both hdmi0 and edp0 use the same port, and only one of them can be used at the same time;
2. HDMI1 and EDP1 share the same port, and only one of them can be used at a time;
3. There are only four VPs internally, so a maximum of four interfaces can be activated;
4. HDMI, EDP, and DP can only be displayed on VP0, VP1, or VP2;
5. DSI0 and DSI1 can only be displayed on VP2 or VP3;
6. RGB can only be displayed on VP3.

So the optional parameters for HDMI0/1, EDPO/1, DP0/1 are “VP0”, “VP1”, “VP2”, and “OFF” ;

MIPIO/1 optional parameters are: “VP2”, “VP3” ;

The RGB optional parameter is: “VP3” ;

The primary_display parameter depends on the actual display interface assigned to get the VP.

Note: When modifying the device tree, you need to follow the annotation rules to avoid using conflicts. The driver does not detect whether the forlinx-control configuration conforms to the rules. An error in the setting will cause abnormal display.

For the display interface set to “OFF”, blocking, deleting, or retaining is possible. It’s not necessary to set all four VP.

Examples:

Assign VP0 to HDMI0, VP1 to HDMI1, VP2 unused, and VP3 for RGB use. Set the main screen to HDMI0.

```
forlinx-control {
    /**
     * hdmi0 and edp0 share same port, only one can be used.
     * hdmi1 and edp1 share same port, only one can be used.
     * only four vps internally, so up to four interfaces can be activated
     * hdmi edp dp can only be displayed on VP0 or VP1 or VP2.
     * dsi0 dsi1 can only be displayed on VP2 or VP3.
     * rgb can only be displayed on VP3.
     */
    /**
     * if "disabled", display is controlled by uboot */
    status = "okay";

    HDMI0    = "VP0";
    HDMI1    = "VP1";
    RGB      = "VP3";
    primary_display = "HDMI0";
};
```

After saving, recompile to generate the image.

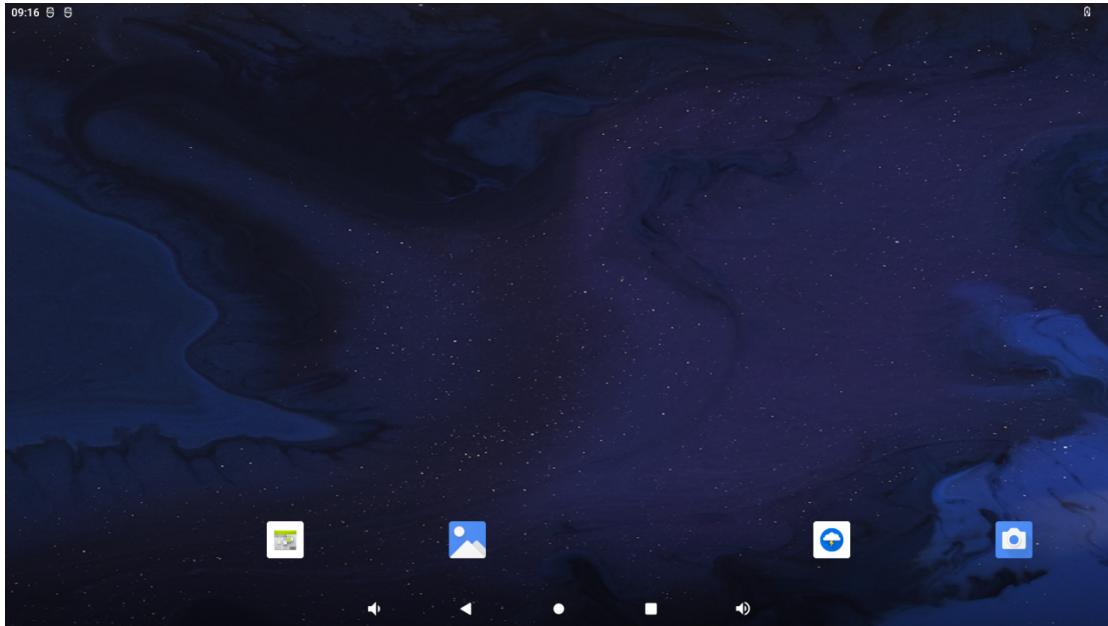
2.5 System Shutdown

In general, the power can be turned off directly. If there is data storage, function use, or other operations, avoid turning off the power arbitrarily during operation to prevent irreversible damage to the file. In such cases, only re-flashing the firmware can resolve the issue. To ensure that data is not completely written, enter the sync command to complete data synchronization before turning off the power.

Note: For products designed based on the SoM, if there are scenarios where accidental power loss causes the system to shut down unexpectedly, measures such as adding power-loss protection can be incorporated into the design.

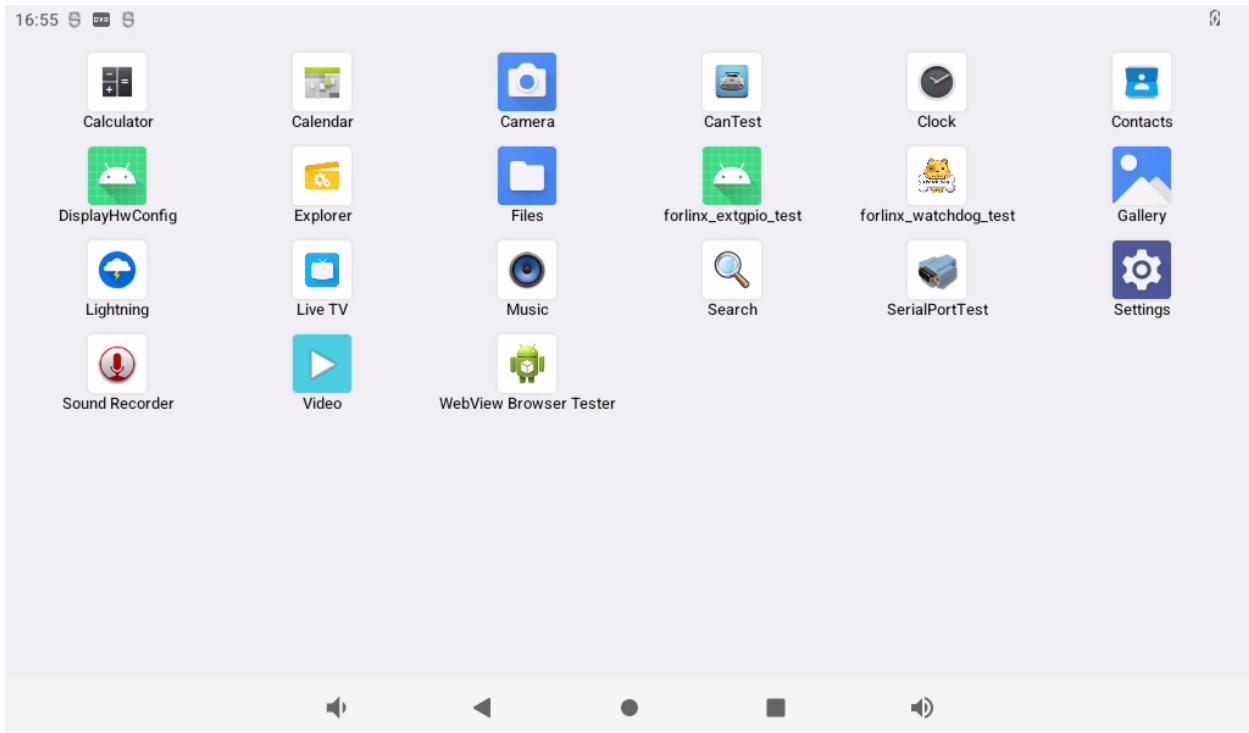
2.3.7 3. Android Function Use and Test

3.1 Main Interface Display



3.2 Application

Swipe up on the main screen to bring up the following screen.

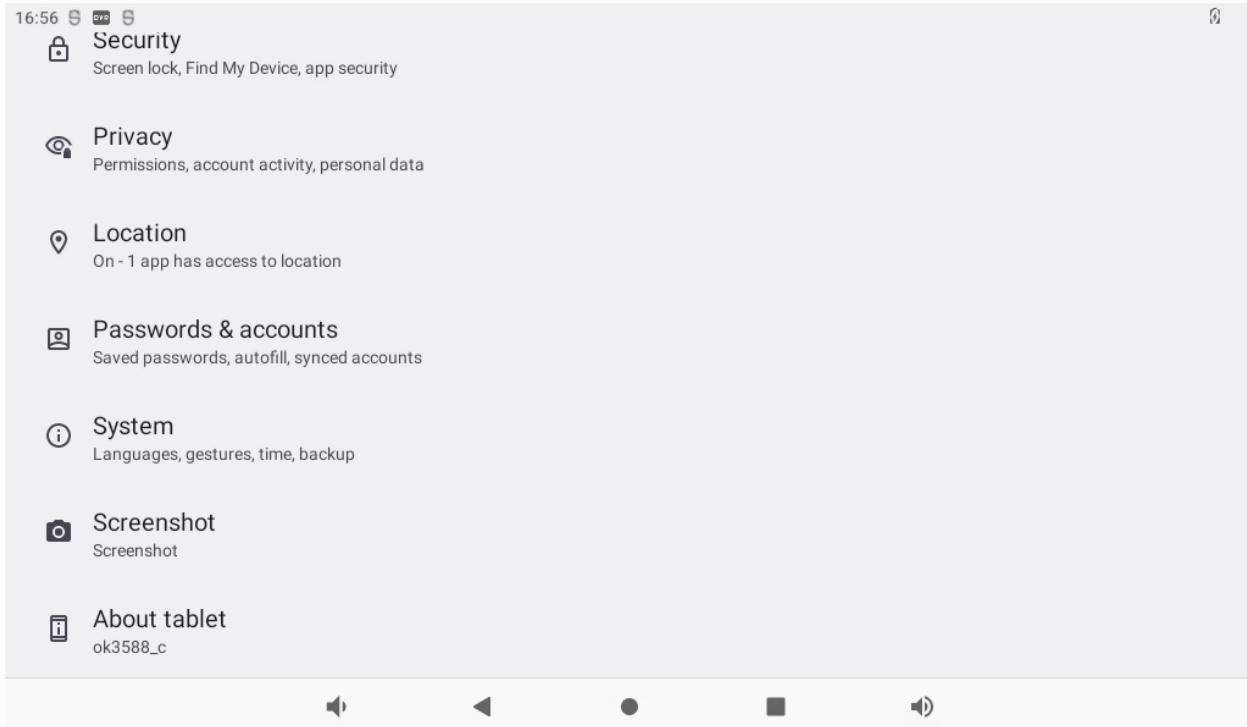


Note: After software version updates, there may be minor differences, which do not represent the actual images for each subsequent version update and are provided for reference only.

3.3 Language Setting



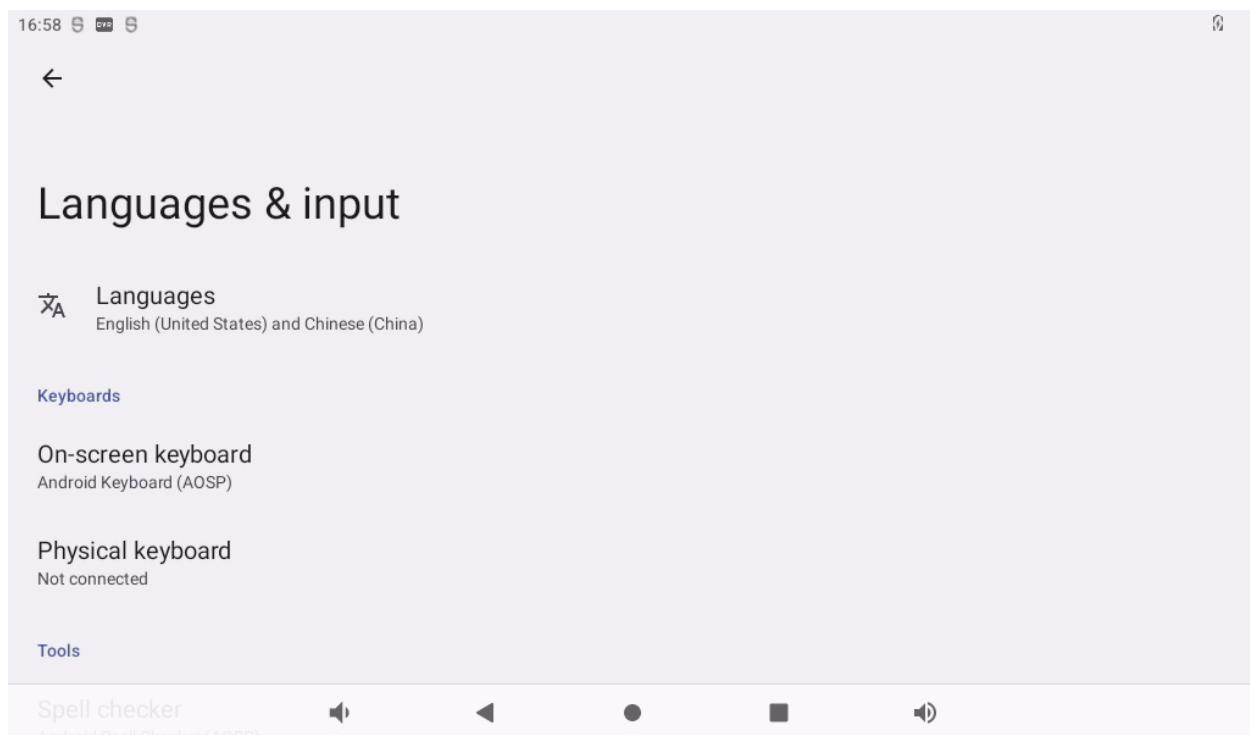
Click “ ” , on the application interface to enter the setting interface:



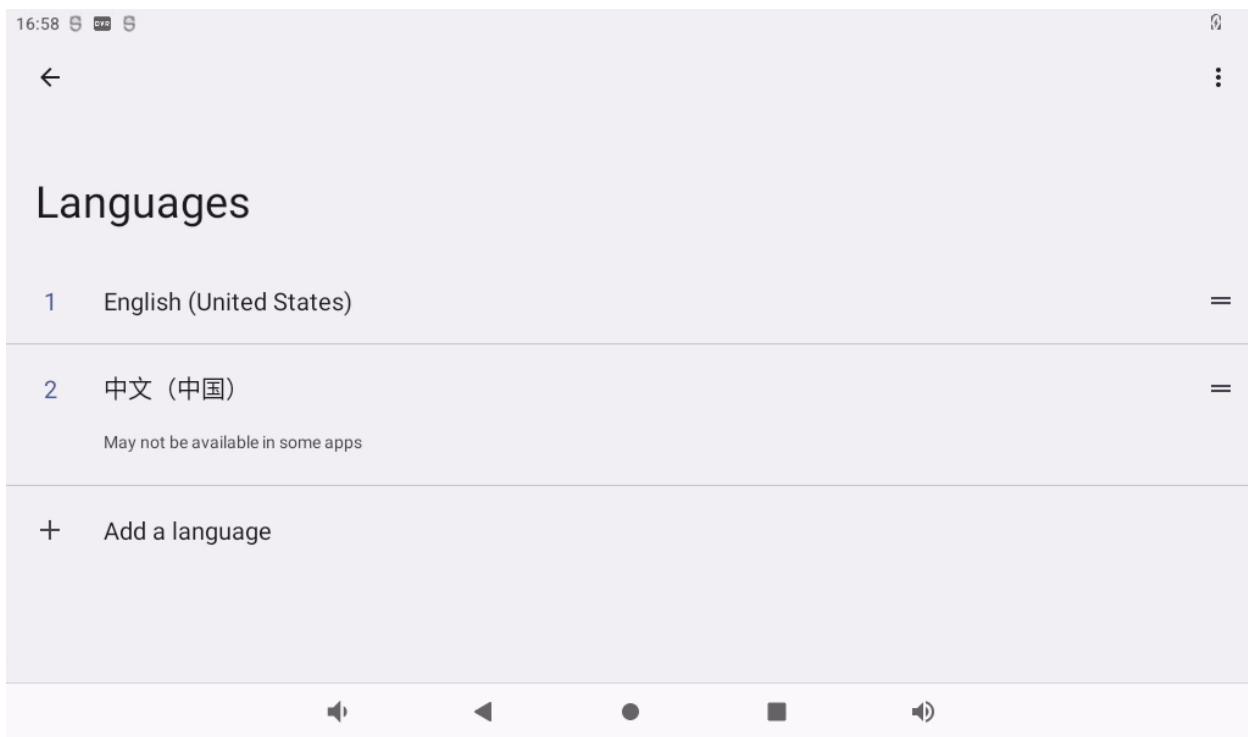
Click application interface to enter the system interface.



Click “Language and input method” to enter the language setting interface:



Click “Language” to enter the language selection interface:

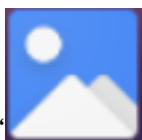


Click “Add Language” to add a new language.

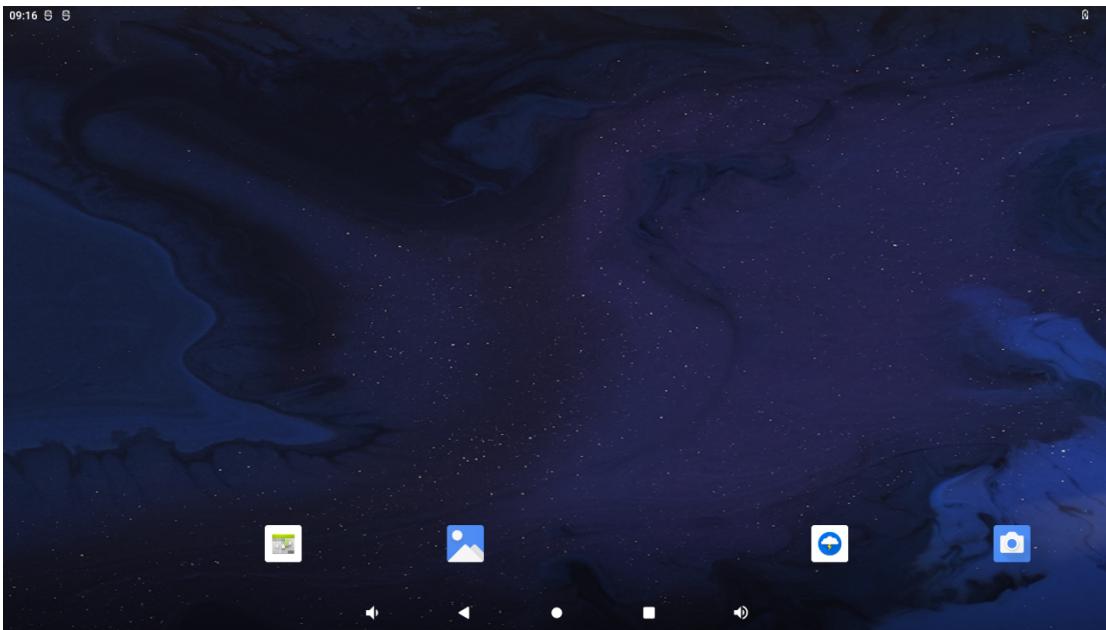
If you want to remove an installed language, you can click the icon with three dots in the upper right corner, select Remove, check the language you want to delete, click the trash can icon in the upper right corner, and a dialog box pops up, “Do you want to remove the selected language?” Click “Confirm” to deleted the language.

3.4 Picture and Audio View

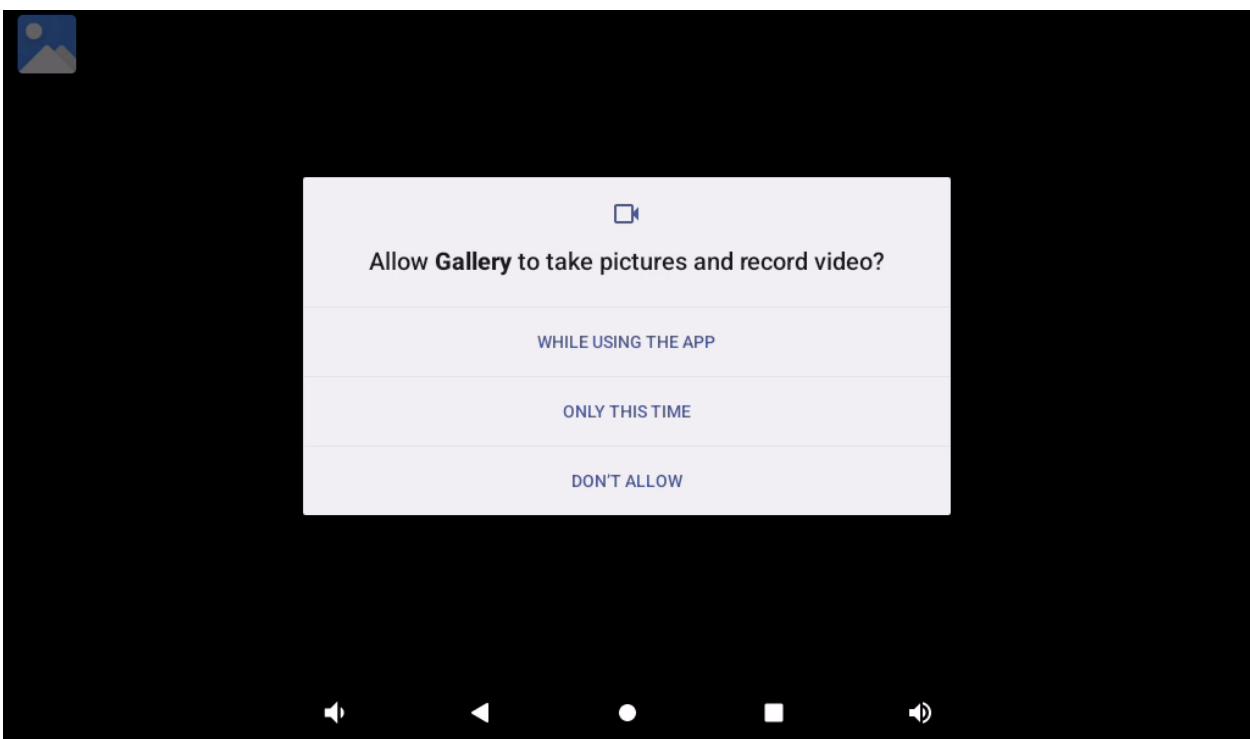
Store the picture and video files to be viewed into the TF card, and insert the TF card into the development board.

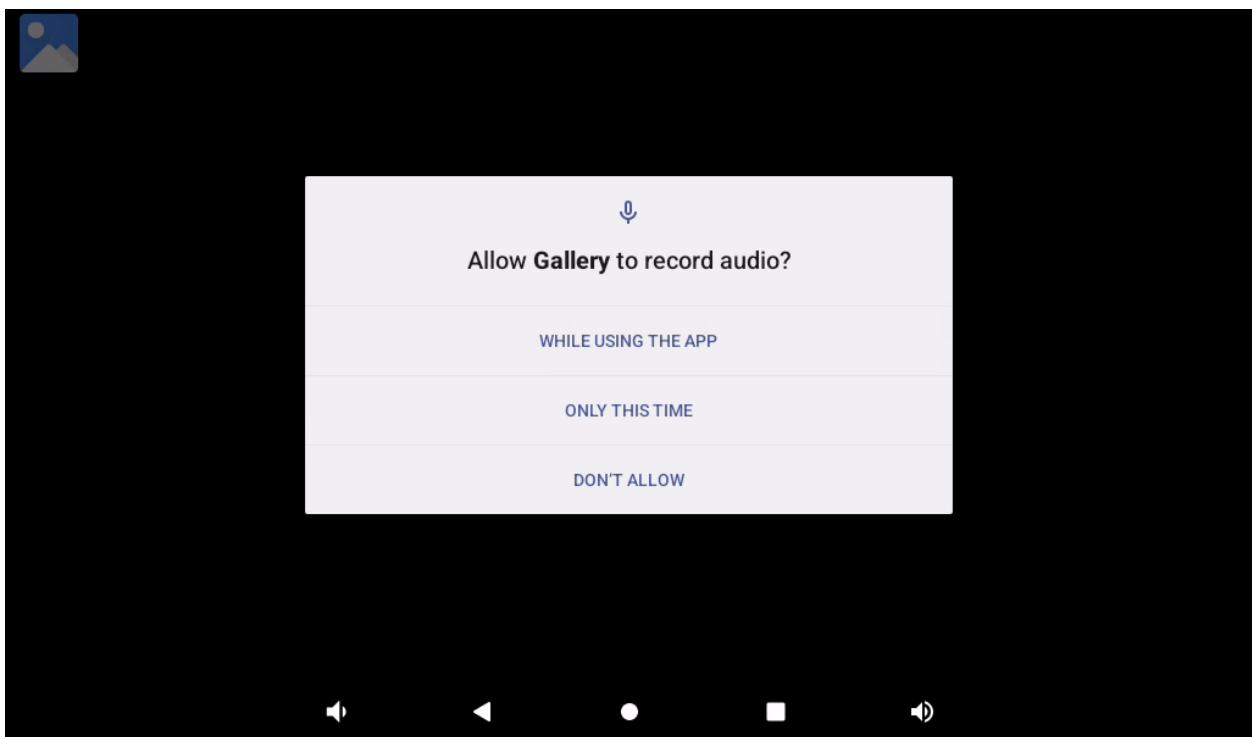


Click “” on the application interface to enter the TF card picture browsing interface.

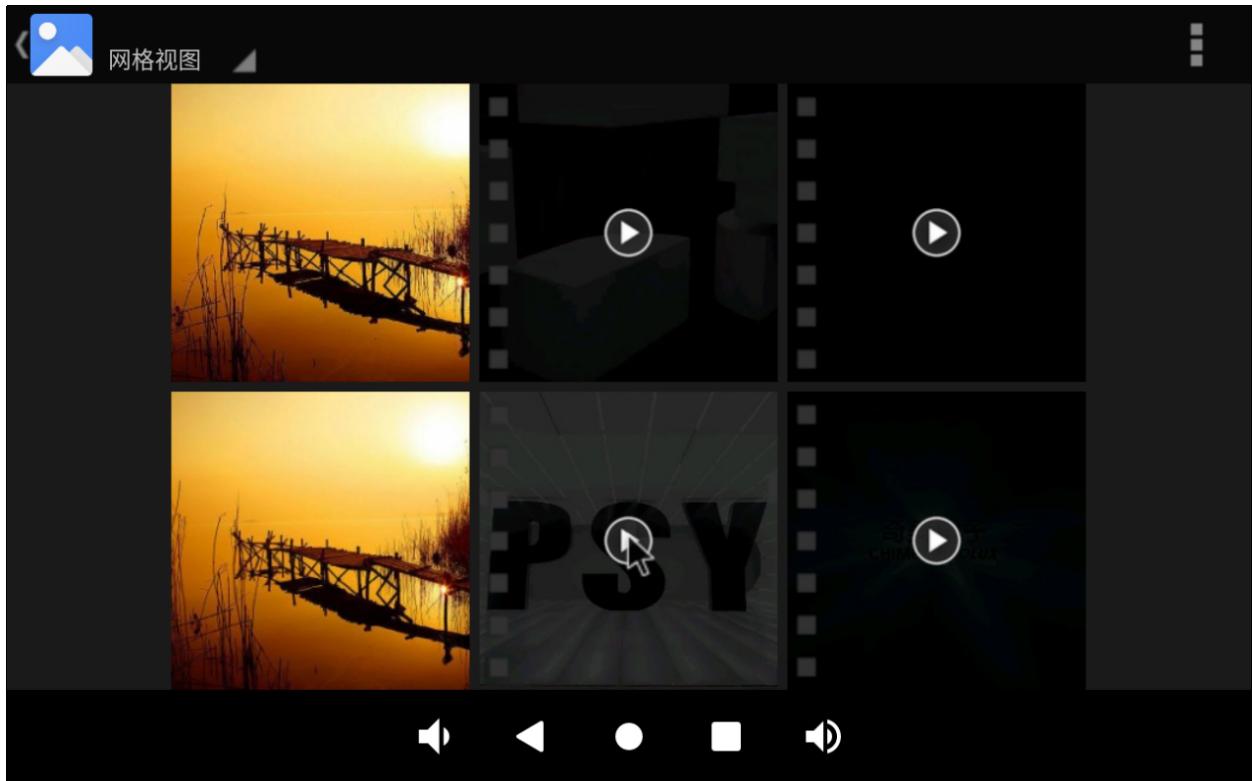


Configure permissions:





After configuration, enter the picture and video view:



Click on the pictures and videos to view:

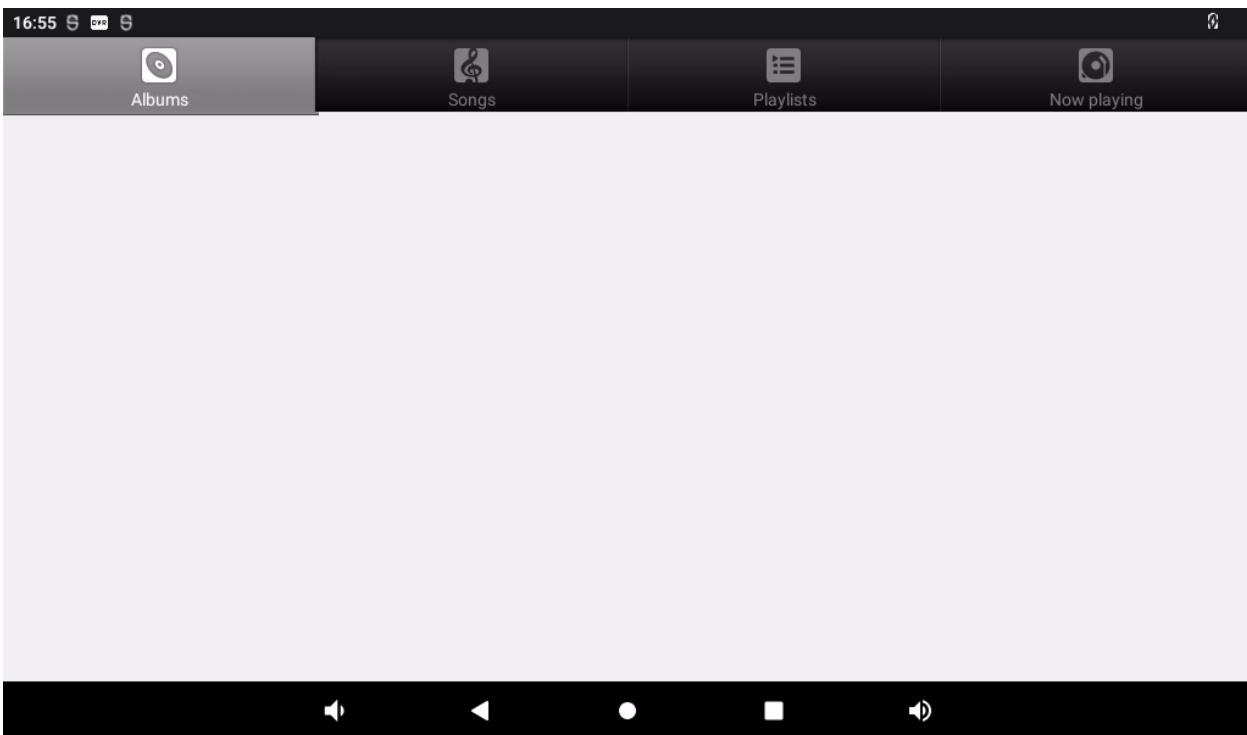


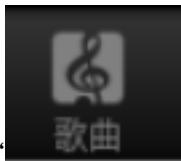
3.5 Multimedia Test

Store the audio file to be played into the TF card, and insert the TF card into the development board.



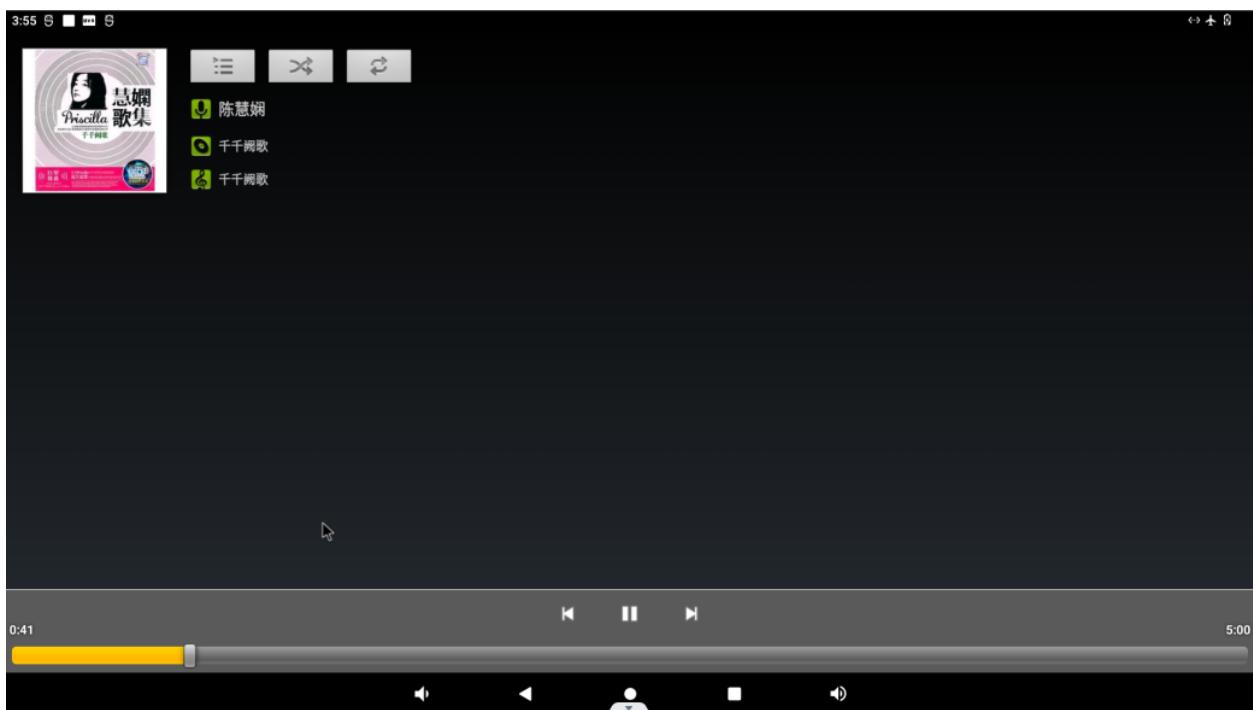
Click “ ” in the application interface to enter the music player interface.





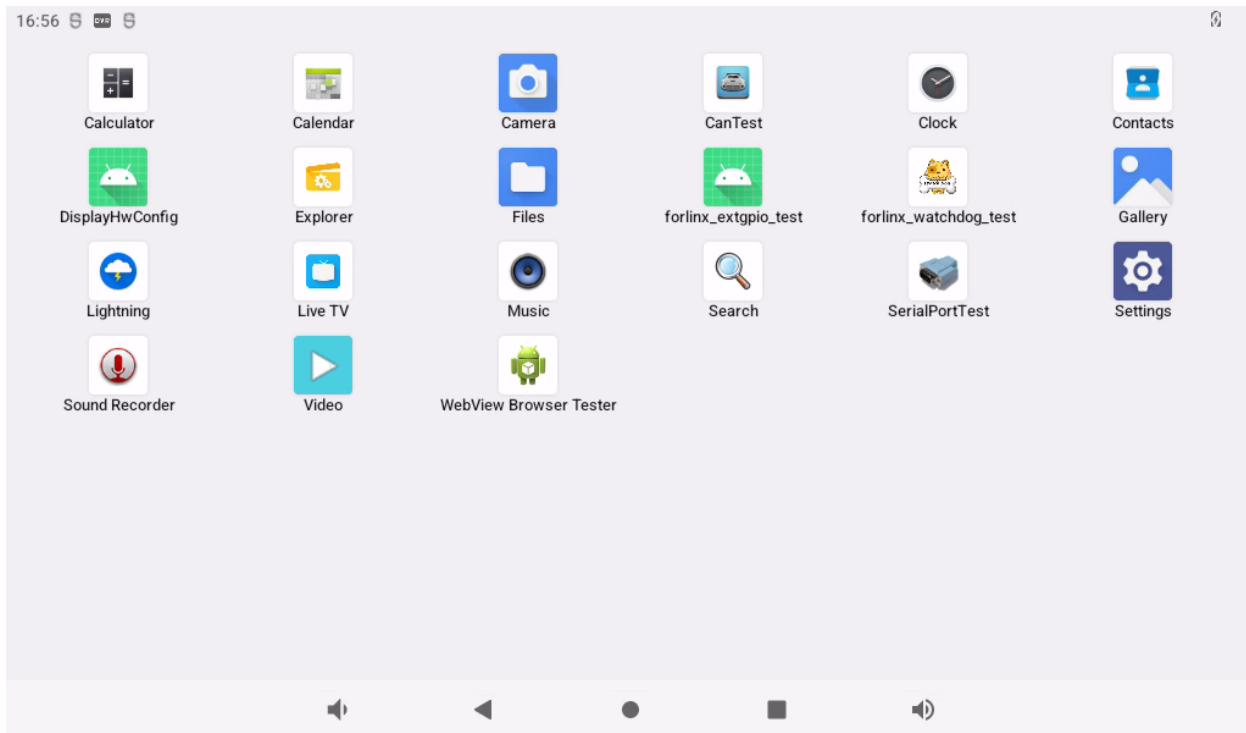
Click “

Click Play Music to enter the play interface.

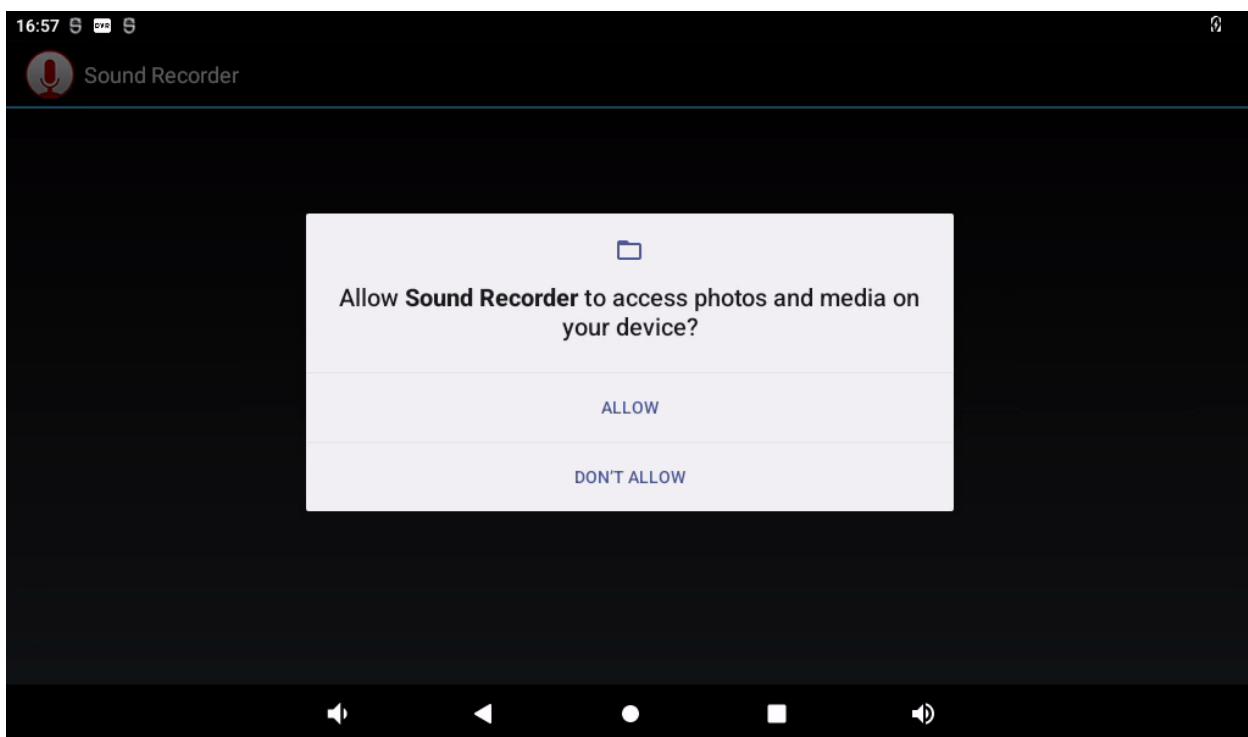


The priority of sound playback is headphone > HDMI audio > carrier board speaker, and the volume can be adjusted by pressing the physical keys VOL + and VOL- on the carrier board of the development board.

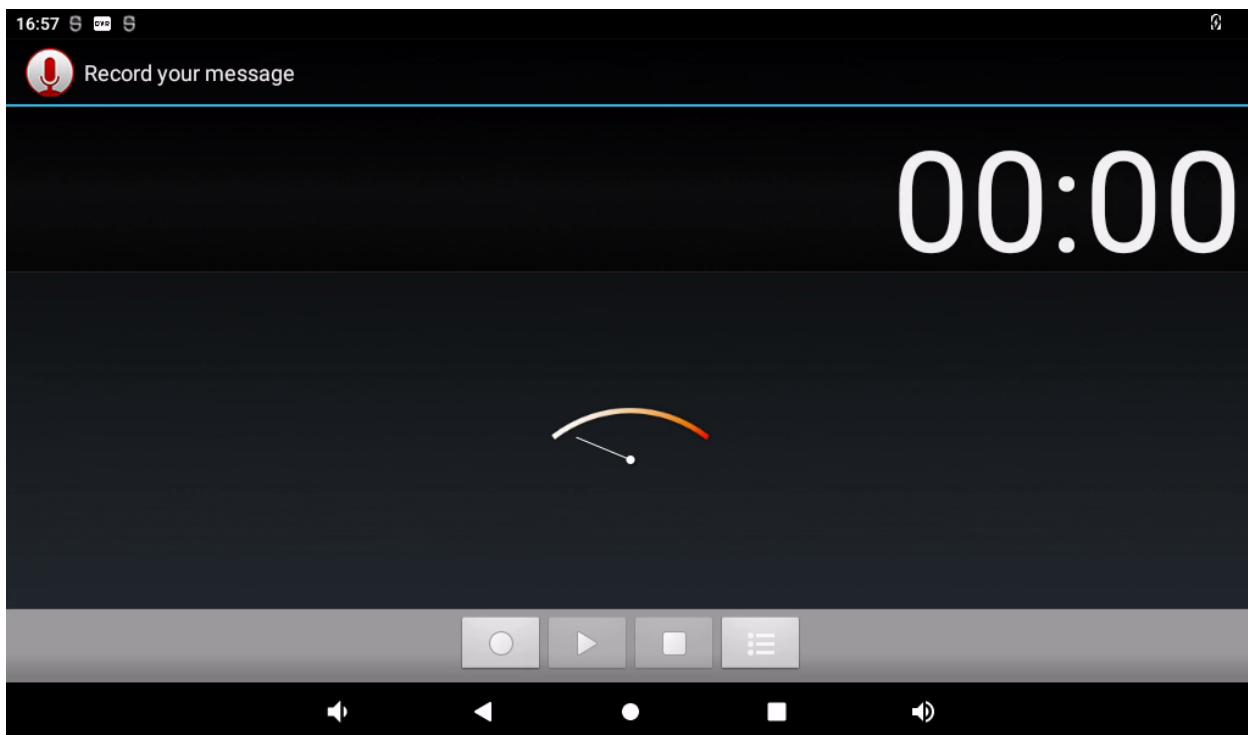
3.6 Recording (Supports Mic Input)

Click the video “A screenshot of the Forlinx Embedded RK3588 application interface. The screen shows a grid of app icons. In the bottom-left corner of the grid, there is a red microphone icon. The icons are arranged in four rows: Row 1: Calculator, Calendar, Camera, CanTest, Clock, Contacts. Row 2: DisplayHwConfig, Explorer, Files, forlinx_extgpio_test, forlinx_watchdog_test, Gallery. Row 3: Lightning, Live TV, Music, Search, SerialPortTest, Settings. Row 4: Sound Recorder, Video, WebView Browser Tester. Below the grid is a navigation bar with four circular icons: left arrow, right arrow, dot, and square. At the very bottom of the screen are physical control buttons: volume up, volume down, and a central button.

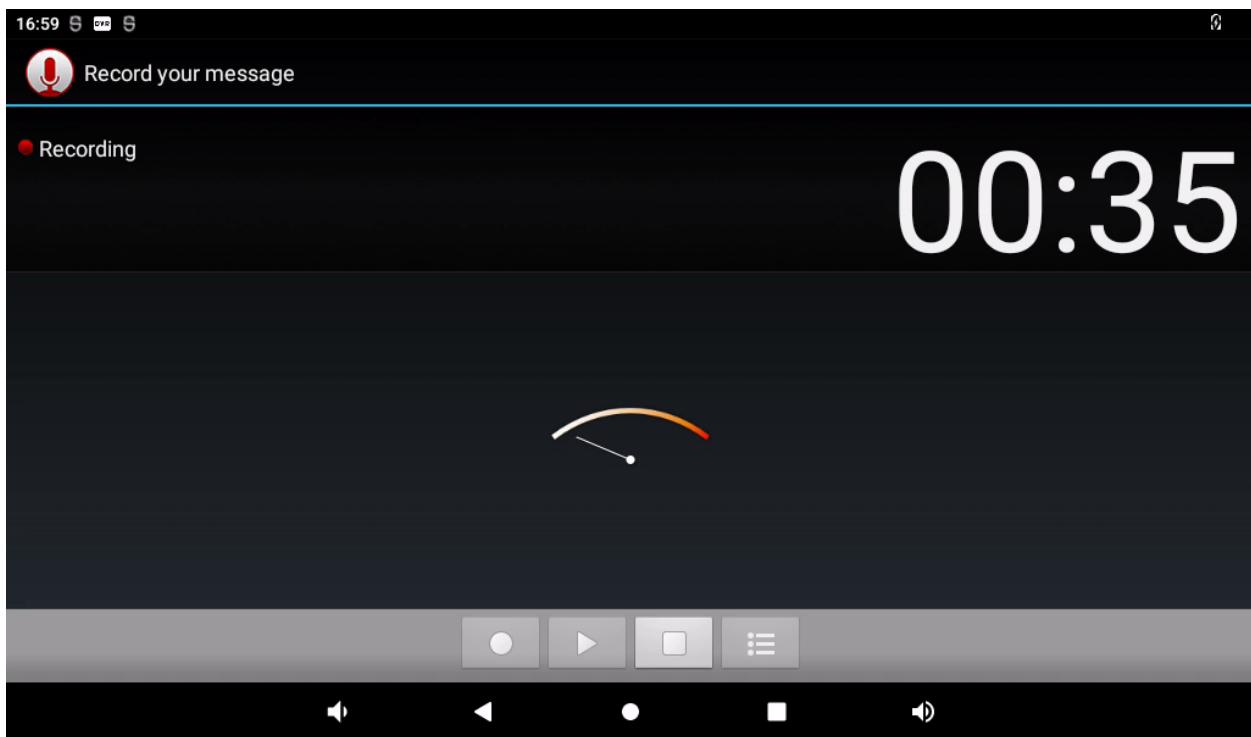
Configure permissions:



Click the round button to start recording: (Note: the pointer will swing according to the sound level during normal recording).

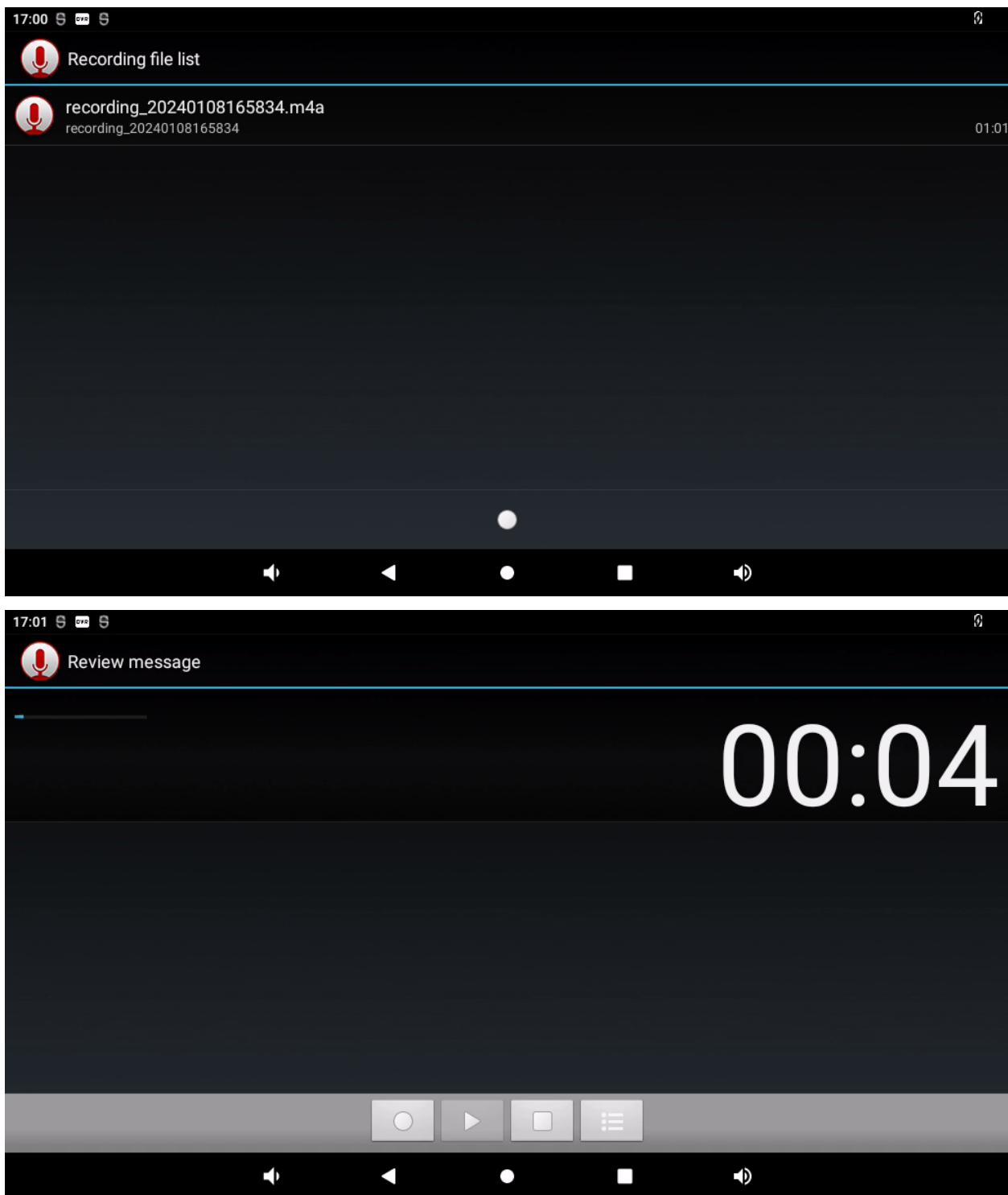


Click the square button to stop recording, and finally click the done button to save.



Click the “A screenshot of the same mobile application interface after a recording has been completed. At the top left is a timestamp "17:00". On the right are signal strength and battery icons. Below the timestamp is a red circular icon with a microphone symbol and the text "Message recorded". To the right is a large digital timer displaying "01:01". Below the timer are two buttons: "Discard" on the left and "Save" on the right. In the center is a list icon. Below the list icon is a control bar with five icons: a circle, a triangle pointing right, a square, and a list icon. At the bottom is a navigation bar with icons for volume, back, stop, play, and forward.

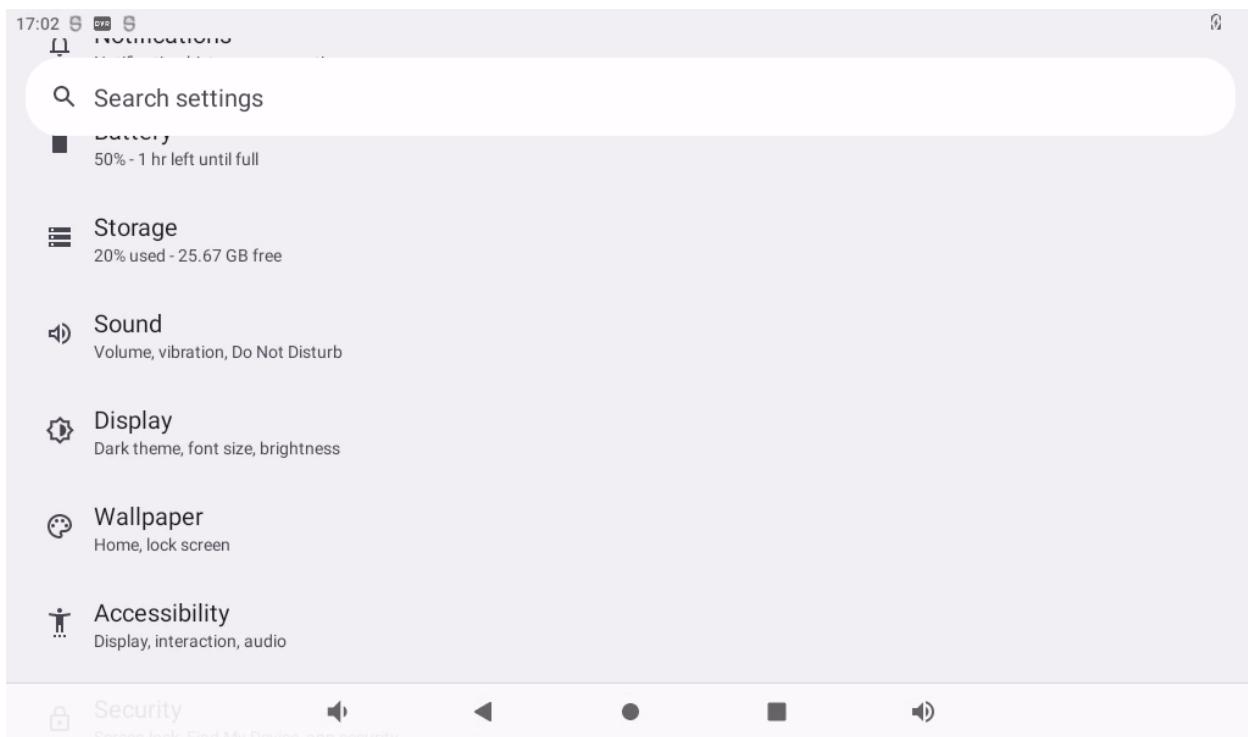
Click on the audio file you want to play, and the recording will start playing.



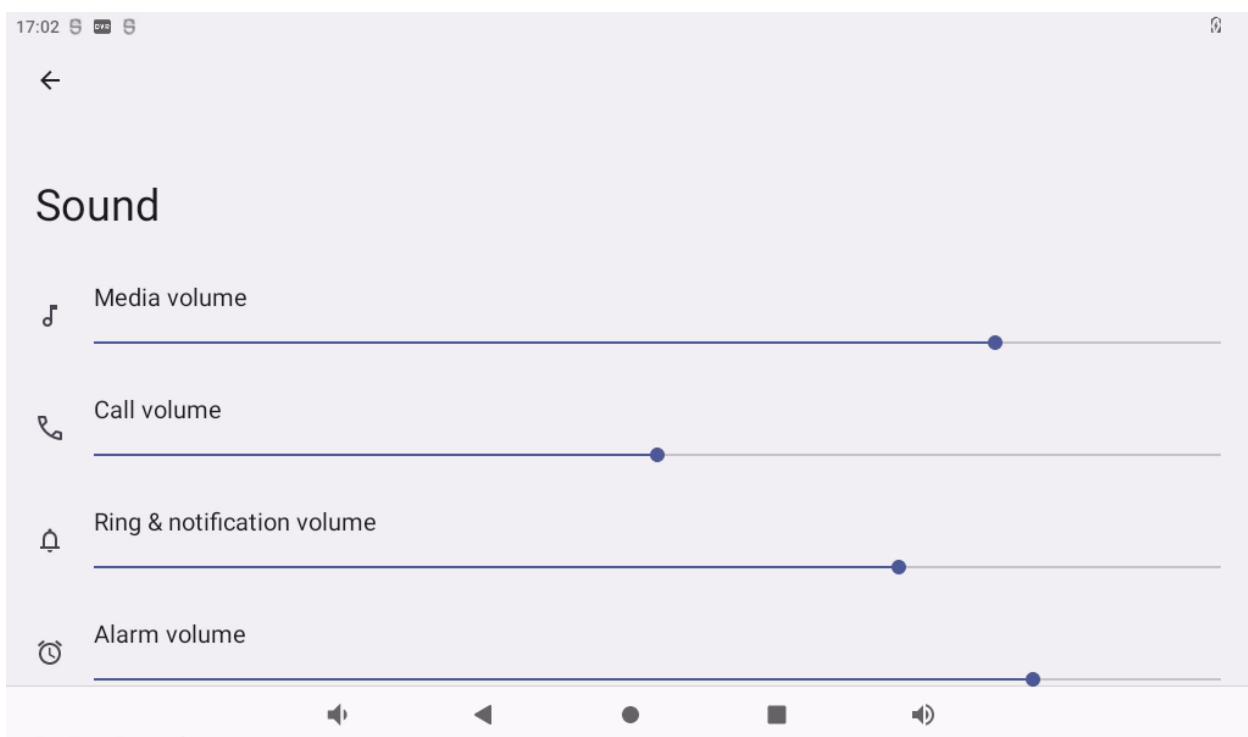
3.7 Volume Adjustment



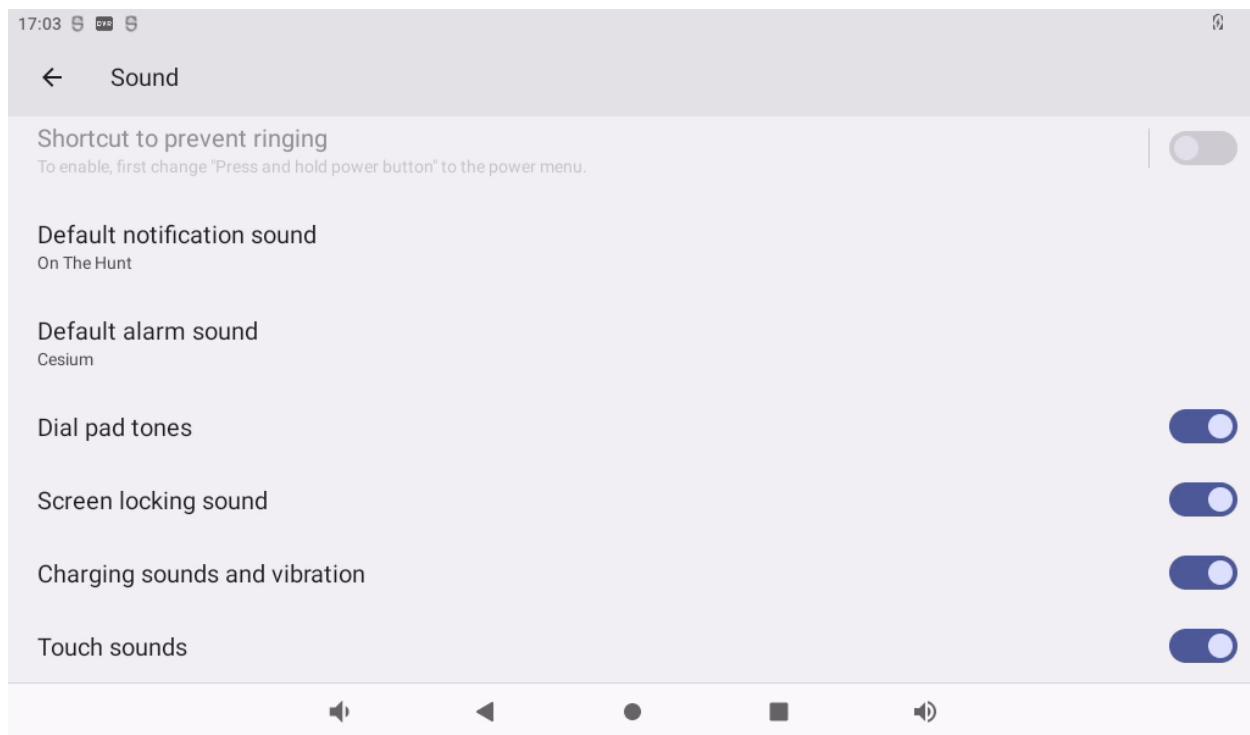
Click “ ” , on the application interface to enter the setting interface:



Click “Sound” in the settings interface to enter the volume settings interface.



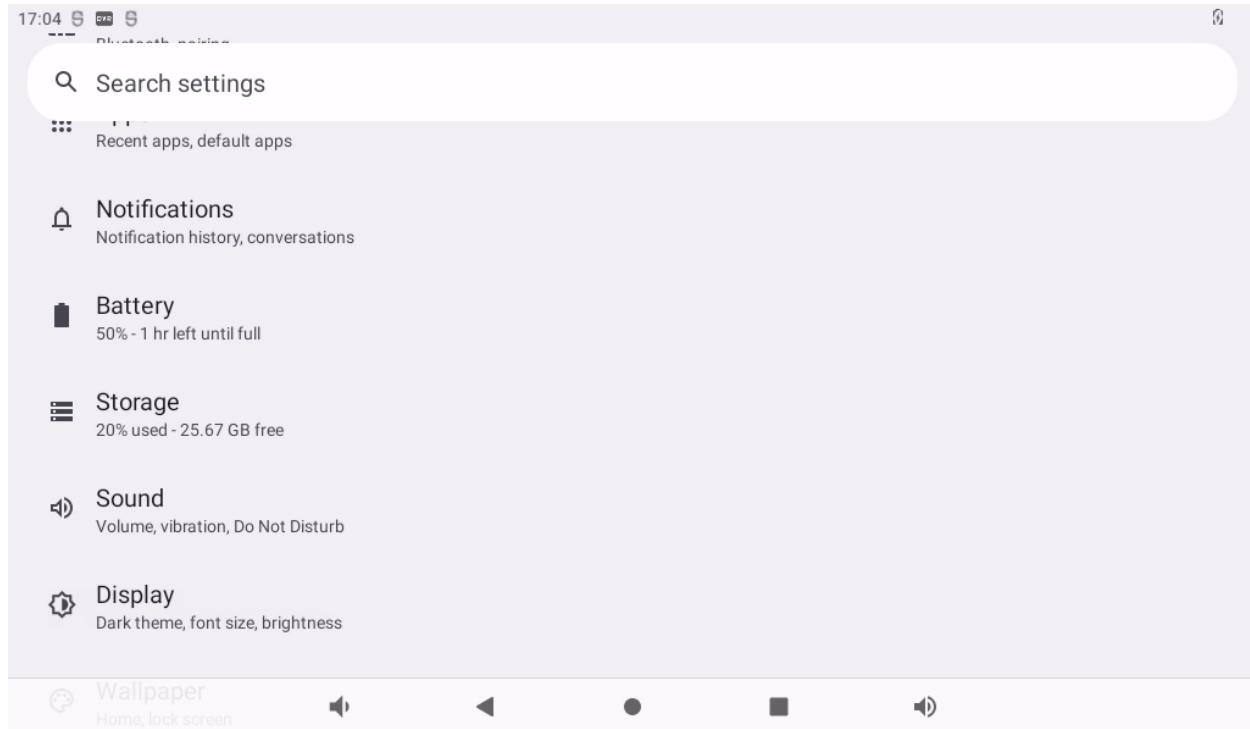
This interface allows you to adjust each section’s volume and supports media volume adjustment using the physical buttons VOL- and VOL+ on the base plate. The default alarm tone is Cesium. Click “Default Alarm Tone” to modify it.



3.8 Display Settings



Click “ ” , on the application interface to enter the setting interface:



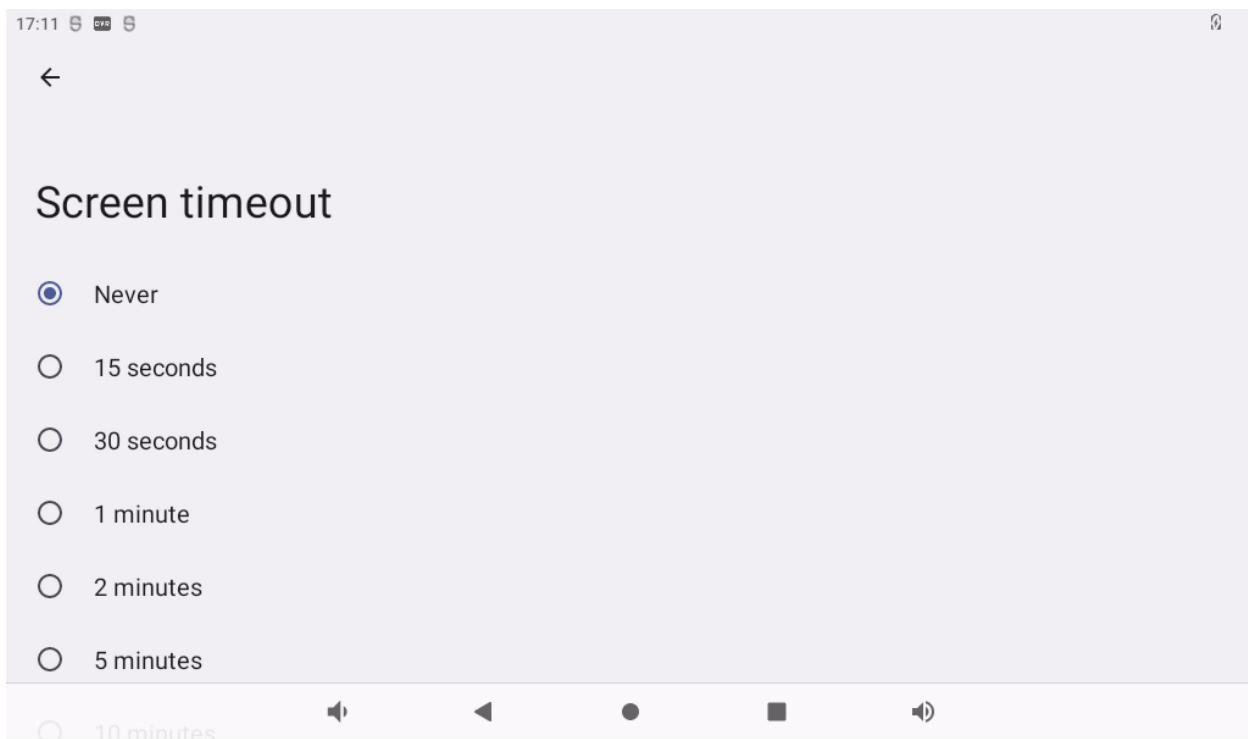
Click “Display” in the setting interface, enter the display setting interface, and select “Brightness” for the backlight setting, then the brightness adjustment slider will appear, adjust the brightness. Because the development board provided by Forlinx does not have a power sensing chip, the automatic screen rotation function in the advanced options does not work.



The default setting of OK3588 is to never turn off the screen. If you need to sleep and wake up, please click the “Screen timeout” option to select the sleep time.



Select screen sleep time.

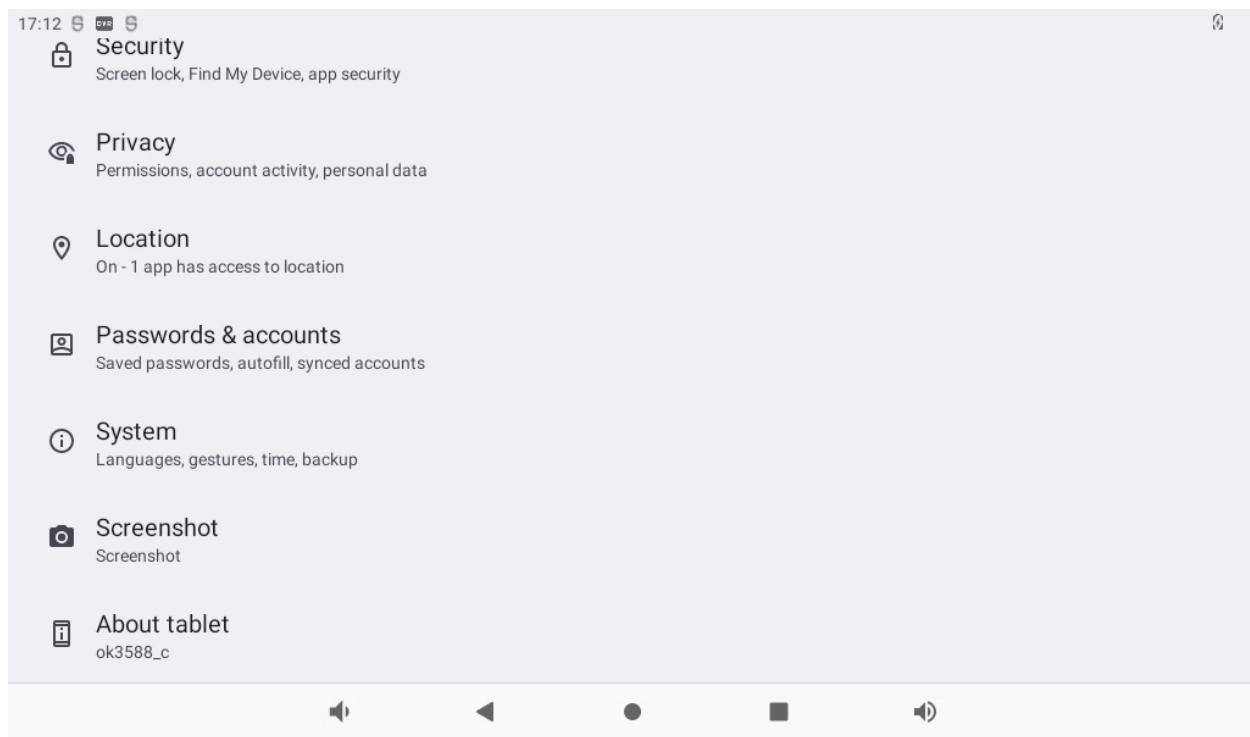


If there is no operation on the interface within the set sleep time, the screen will enter the sleep mode, and pressing the PWRON physical button on the carrier board will wake up the screen.

3.9 Time Setting (RTC)



Click “ ” , on the application interface to enter the setting interface:



Select “System,” where you can change the date and time, and even after power failure, the time can still be synchronized (ensure that the button battery is installed on the board).

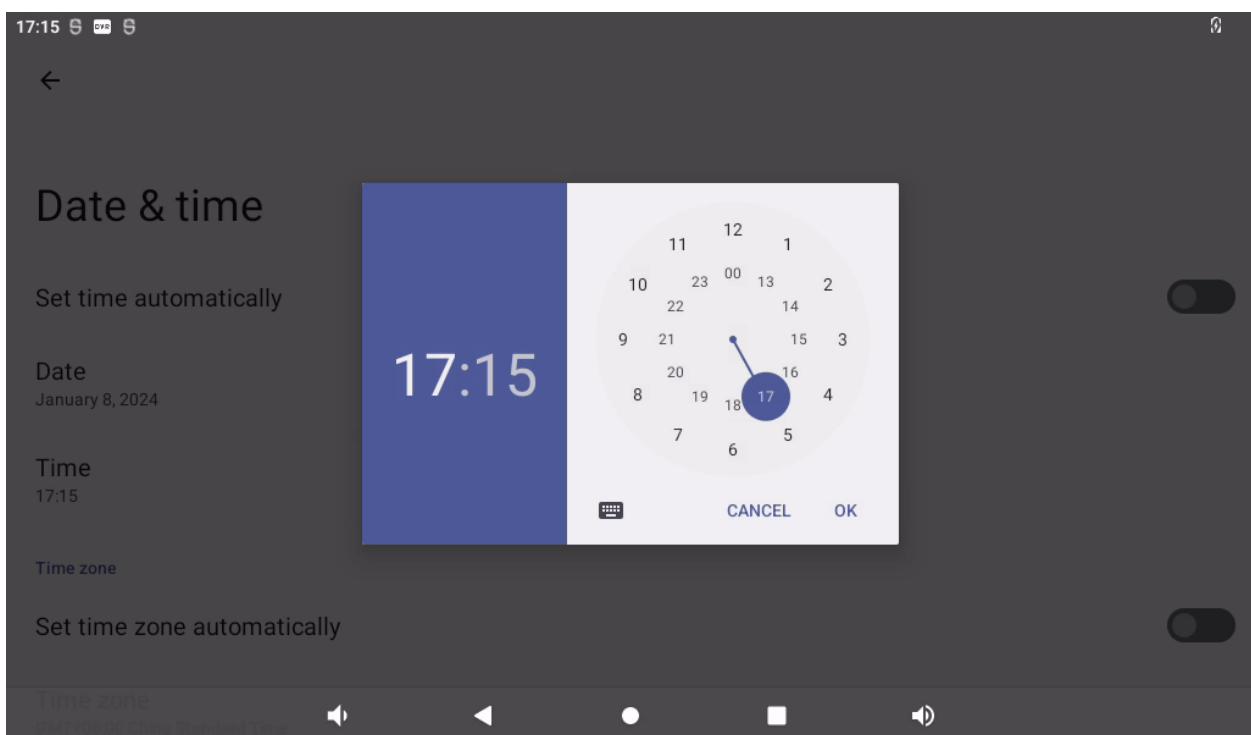


The default is “Turn off network-provided time” and the time format is 24 hours.

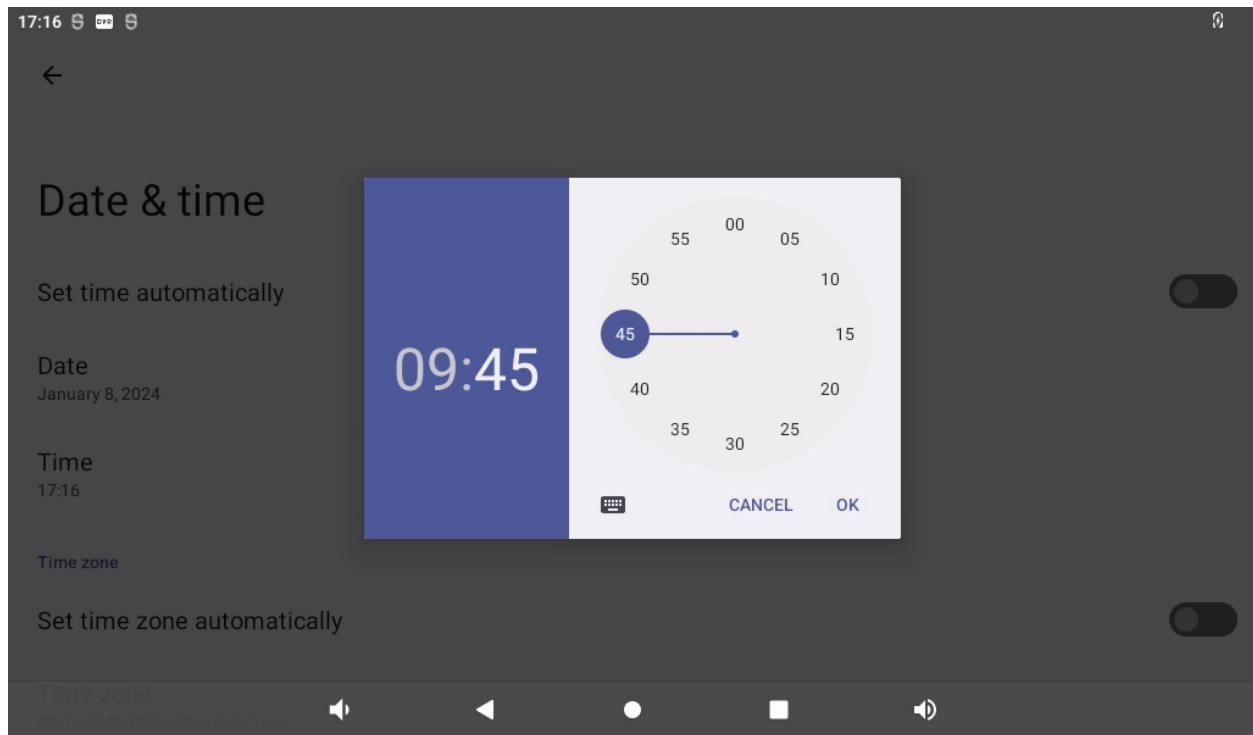
Set the date and time separately:



Click on “Set Date.”



Click on “Set Time.”



3.10 Ethernet Test

OK3588 has two Gigabit NICs on board (Ethernet ETH0 and Ethernet ETH1).

Description:

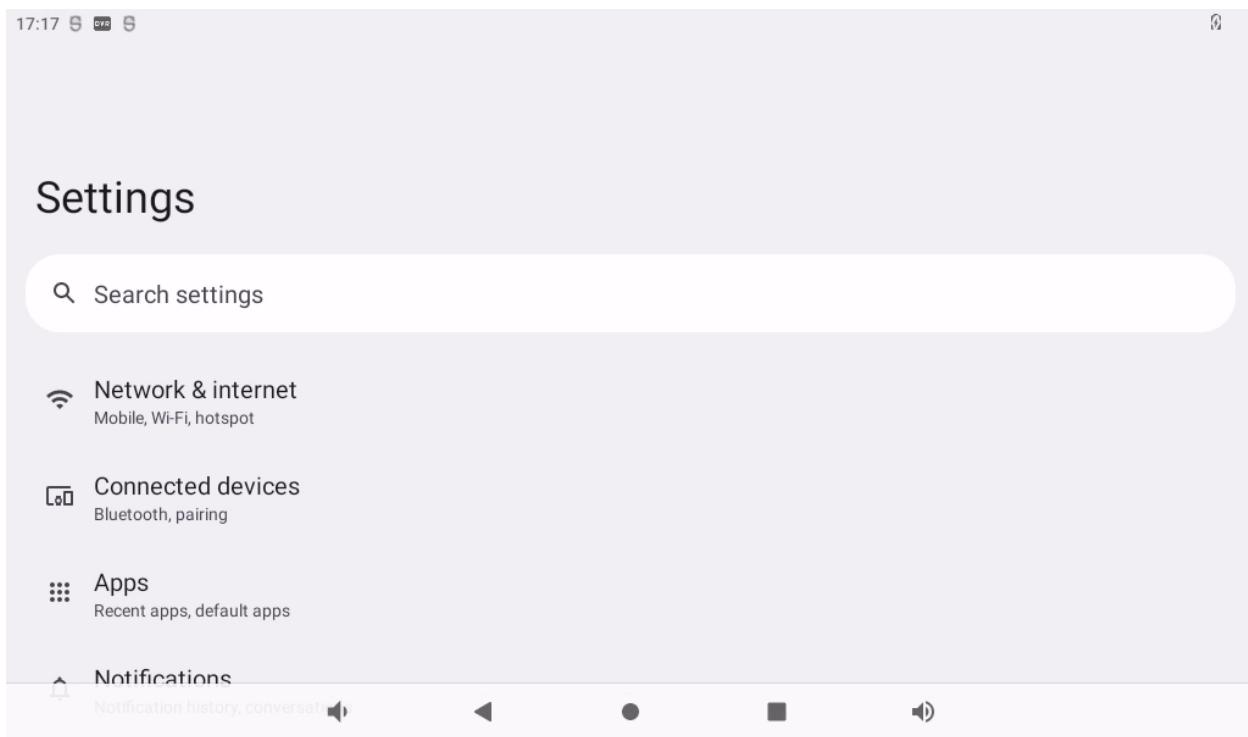
- When 4G and Ethernet exist at the same time, Ethernet is preferred by default; When 4G WIFI exists at the same time, WIFI is preferred by default. When both WiFi and Ethernet are present, Ethernet is prioritized by default.

1. Gigabit network port test:

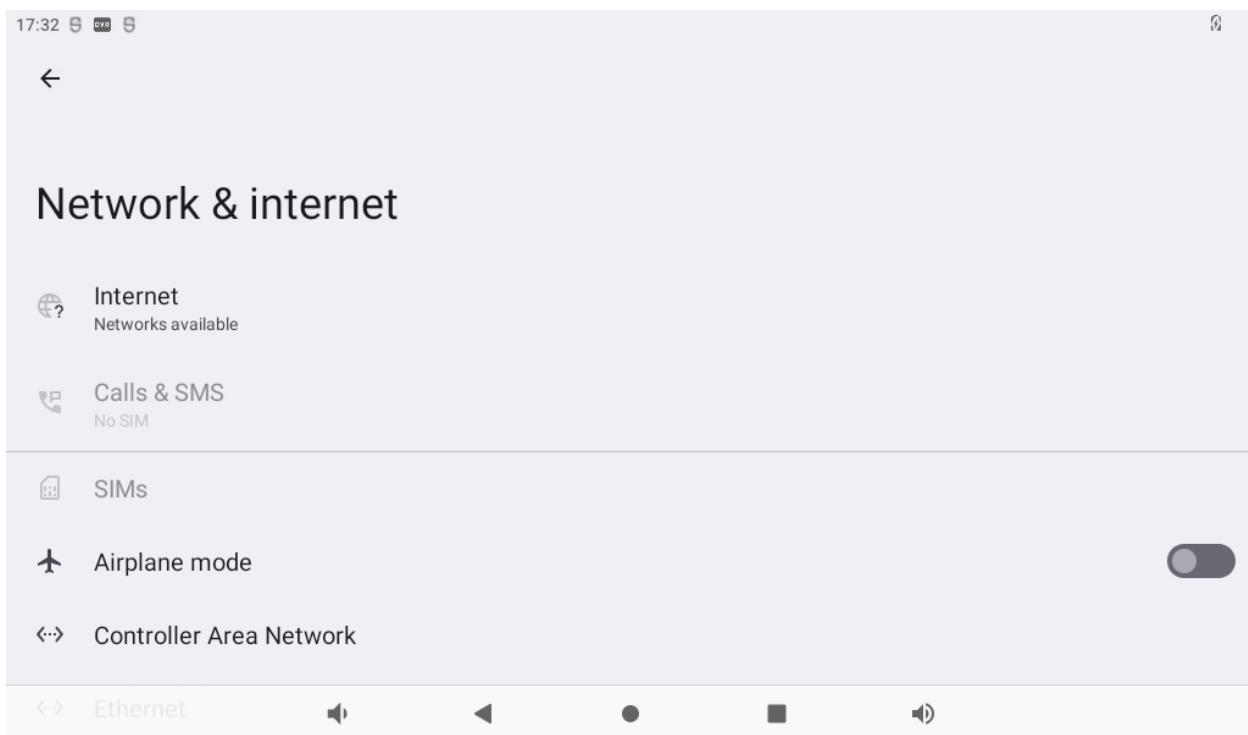
Prepare a router and a network cable that can be connected to the external network port.



After inserting the network cable, click “” on the application interface:



Click “Network and Internet” :

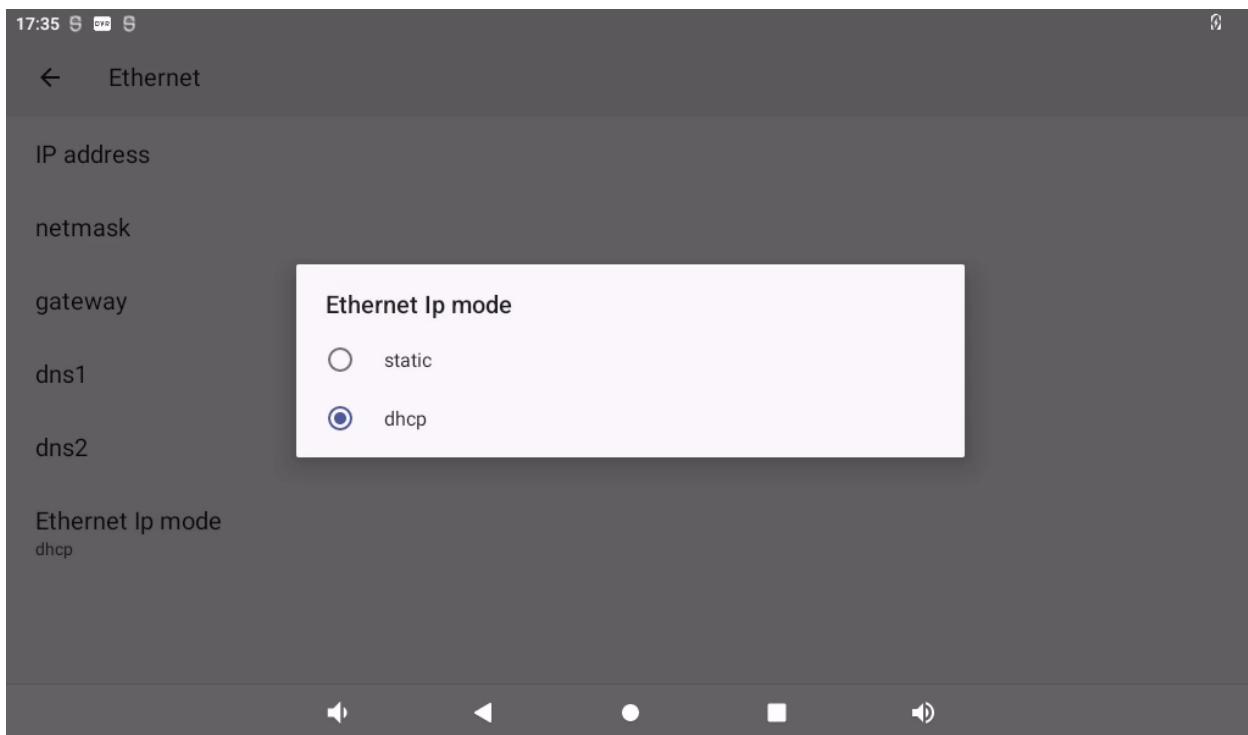


Click “Ethernet ETH0” to choose to automatically obtain IP DHCP or static IP. DHCP is recommended. If you set a static IP, make sure your network parameters are available.

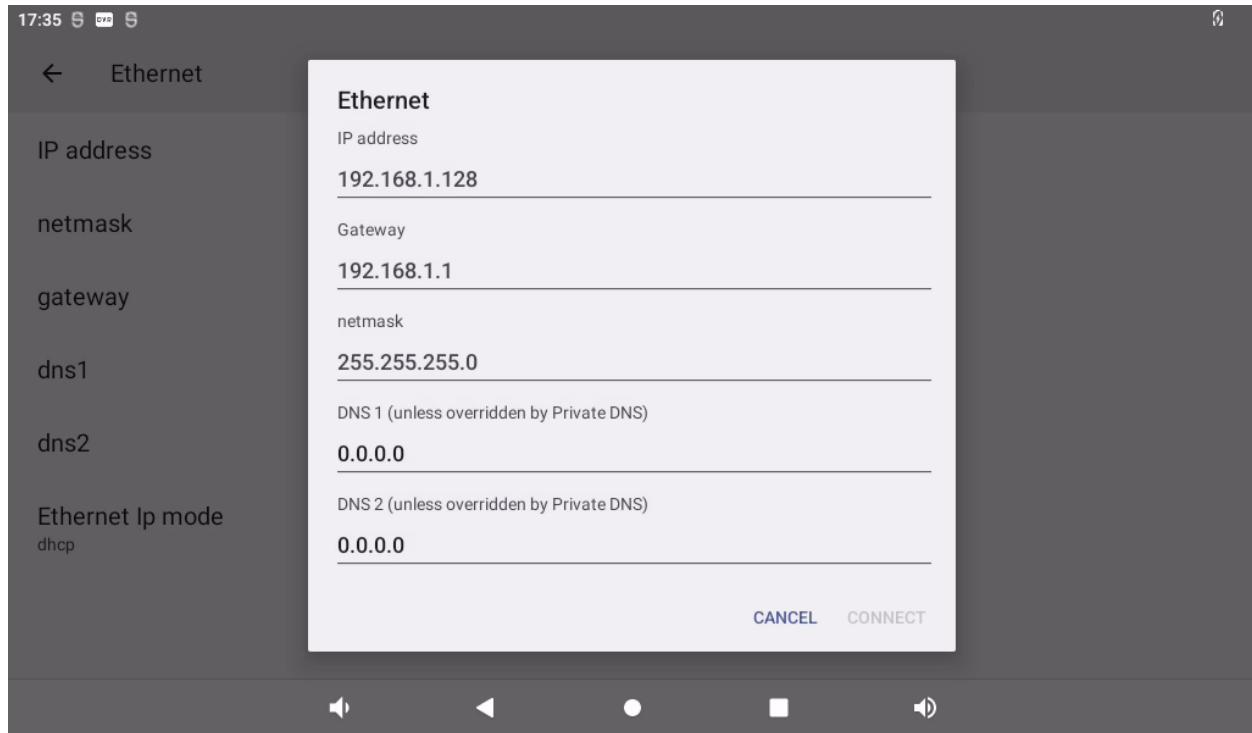
Click “Ethernet” :



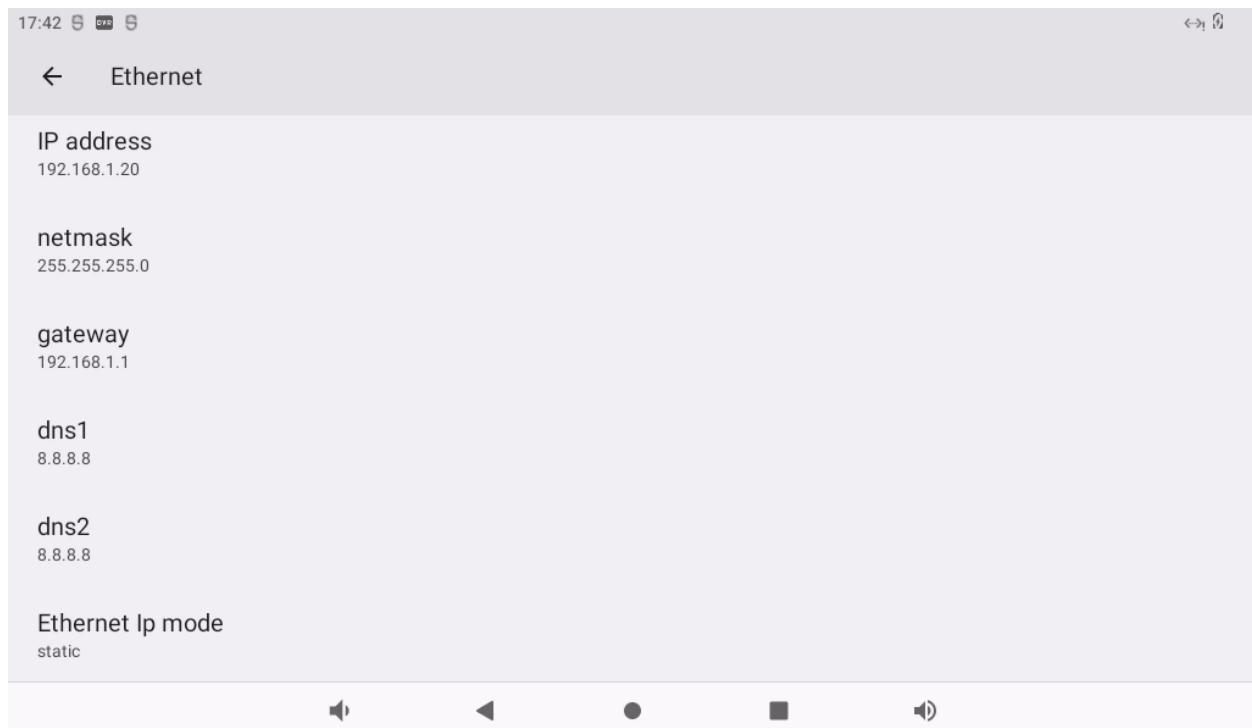
The default IP acquisition method is “dhcp” . If you want to set a static IP, click “Ethernet” Ip mode:



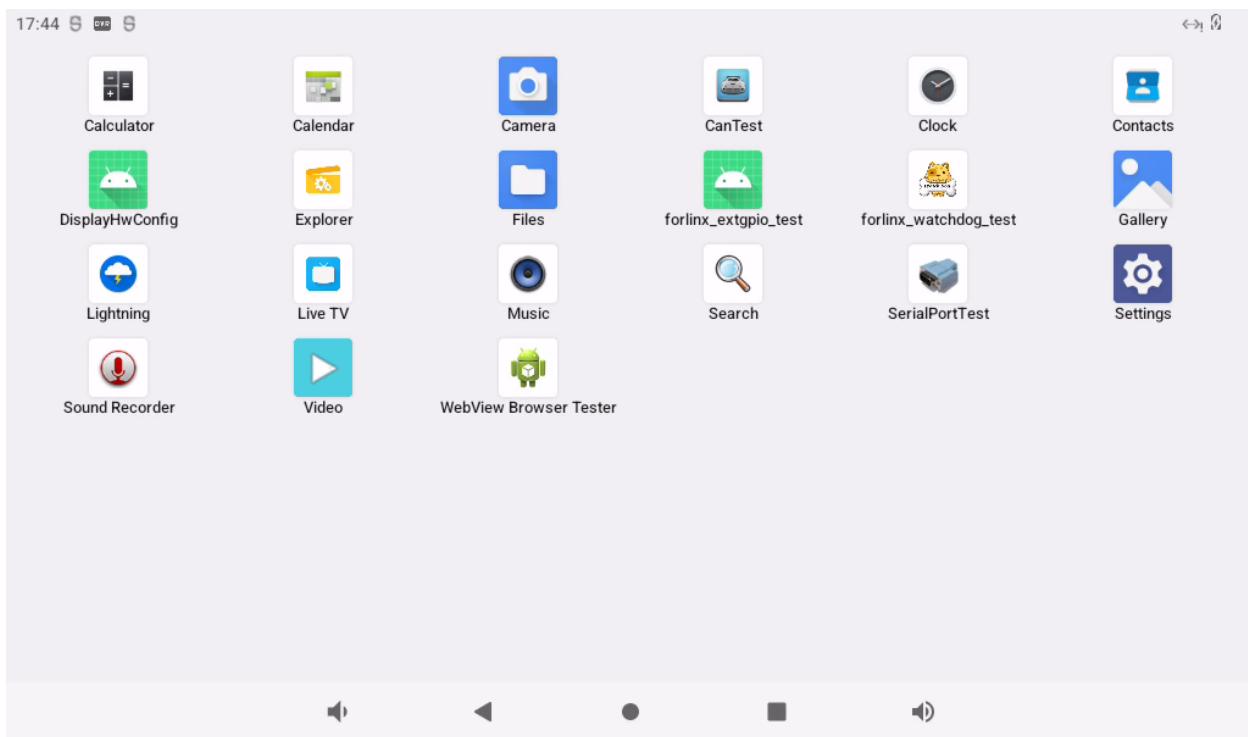
Select Static for static IP configuration:



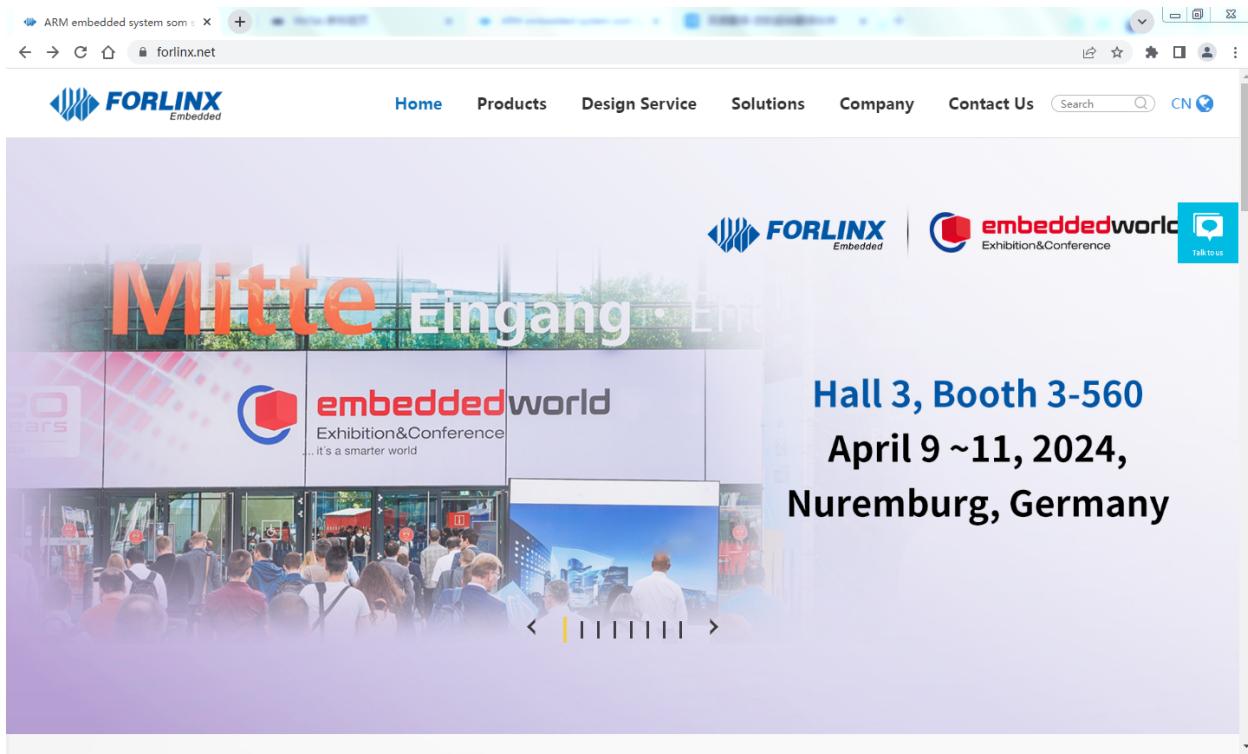
Click “CONNECT” to complete the configuration:



Click Lightning on the application interface for network test:



Enter “<http://www.forlinx.net>” in the domain name column and click “Start” to enter the official website of Forlinx.

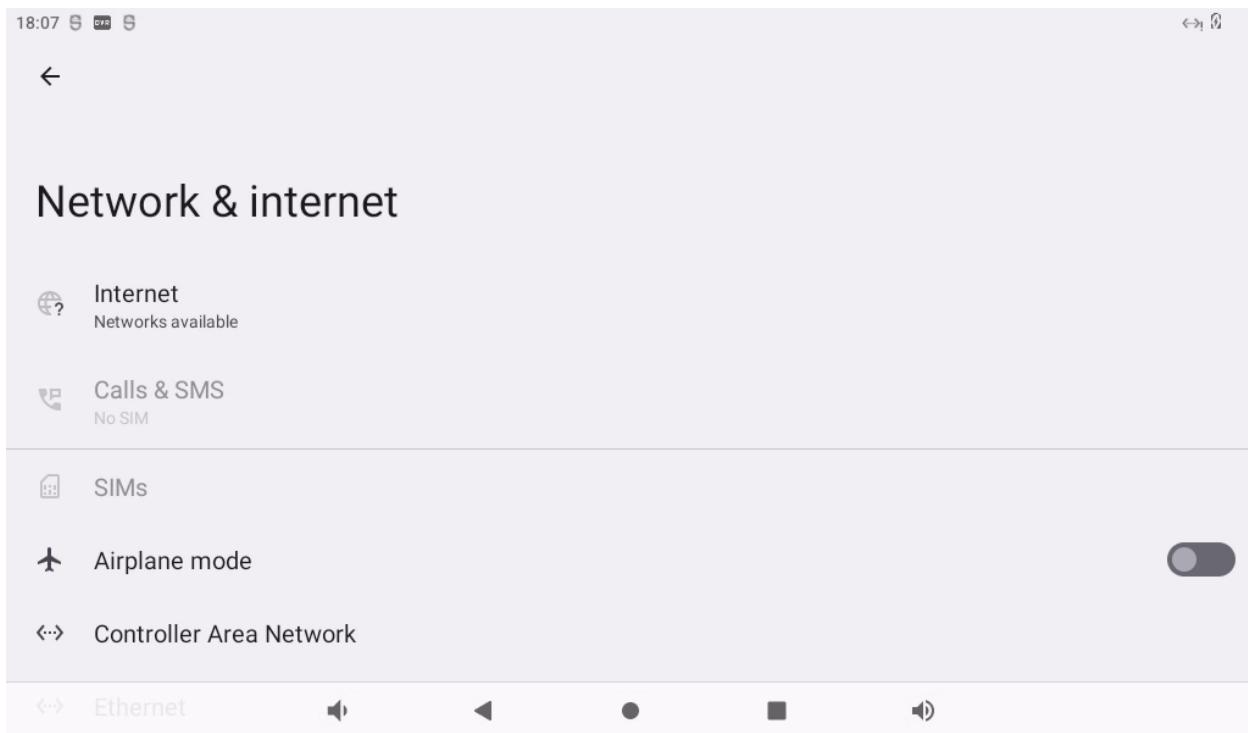


3.11 WiFi Internet

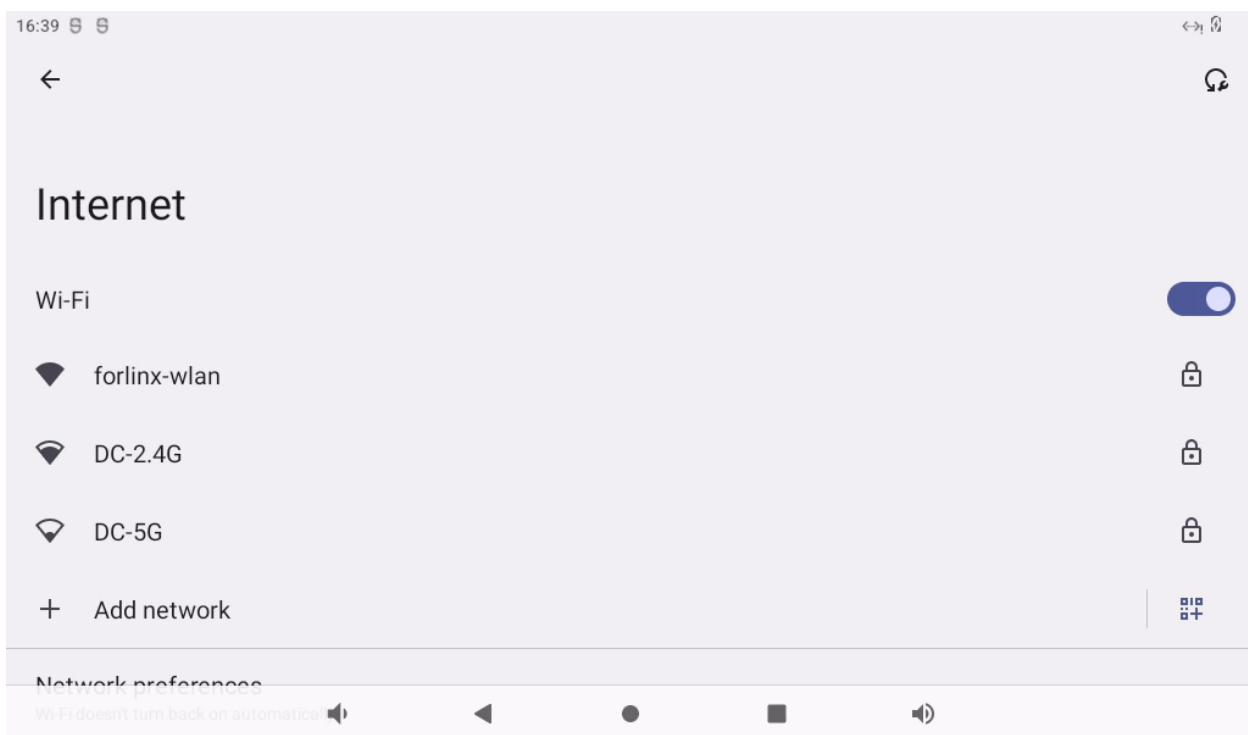
Description:

- When 4G and Ethernet exist at the same time, Ethernet is preferred by default; When 4G WIFI exists at the same time, WIFI is preferred by default. When both WiFi and Ethernet are present, Ethernet is prioritized by default;
- When testing WiFi, unplug the wired network.

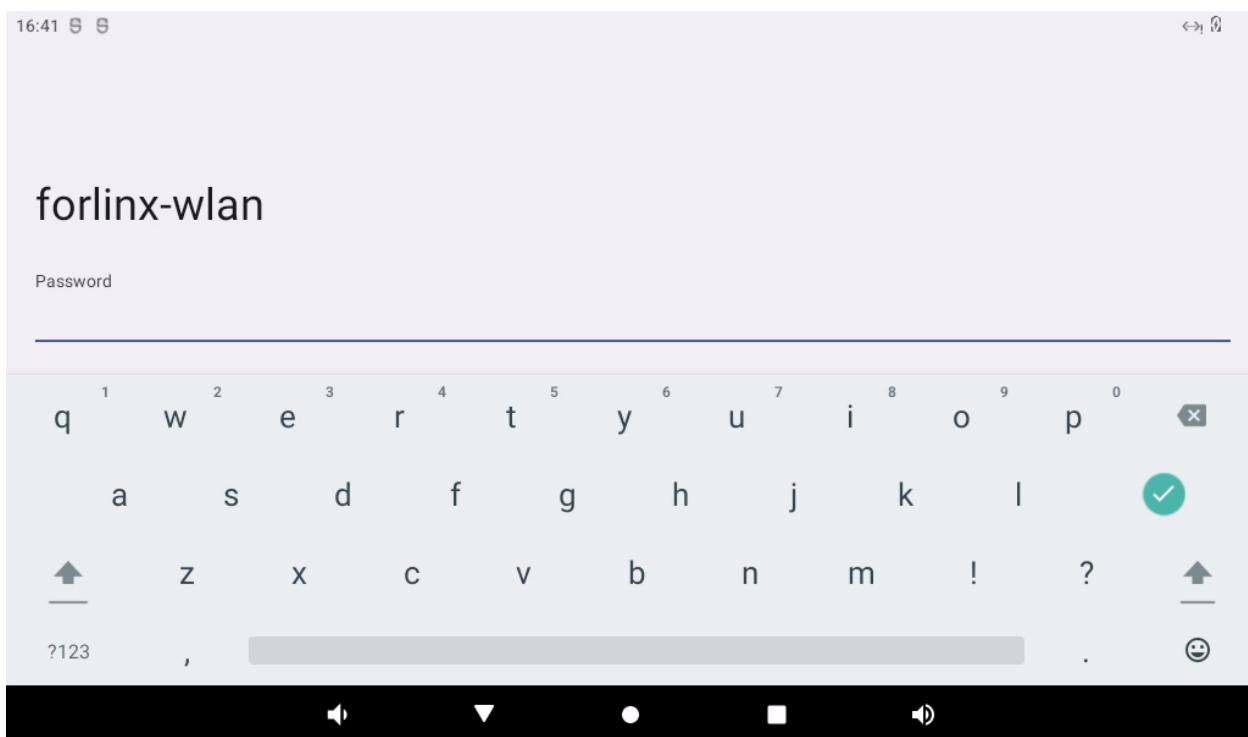
The OK3588 supports two modules onboard, the AW-CM276MA and the AW-XM458. Open Settings, select “Network & Internet”, and click “WLAN” :



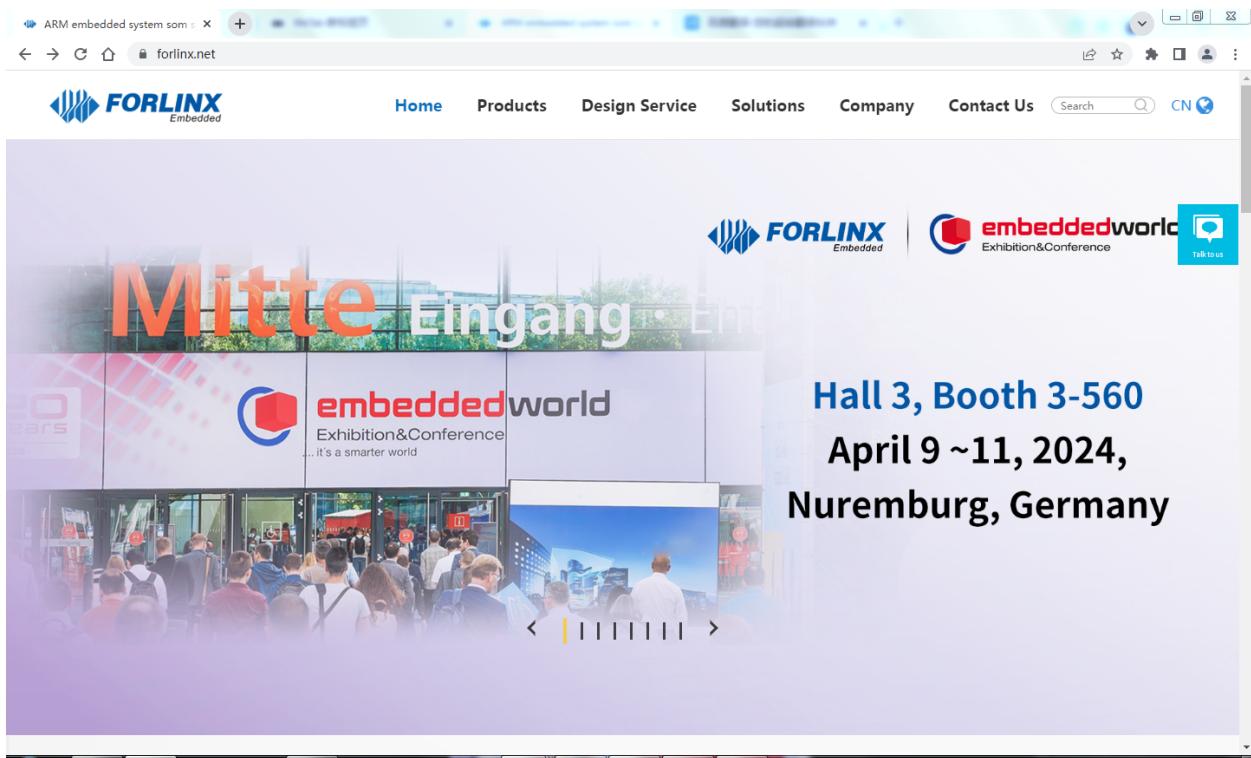
Click “Use WLAN” :



Click on the WIFI to be connected and enter the password:



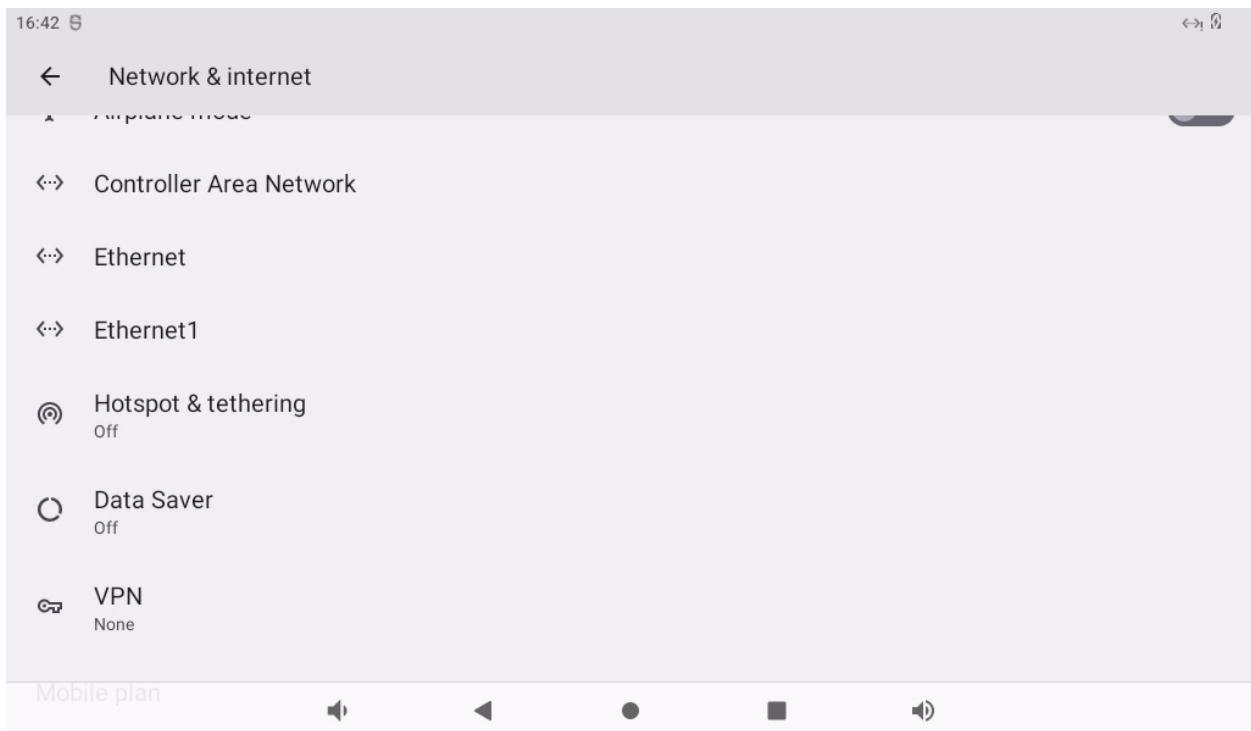
After successful connection, you can open the browser and enter the URL for network test:



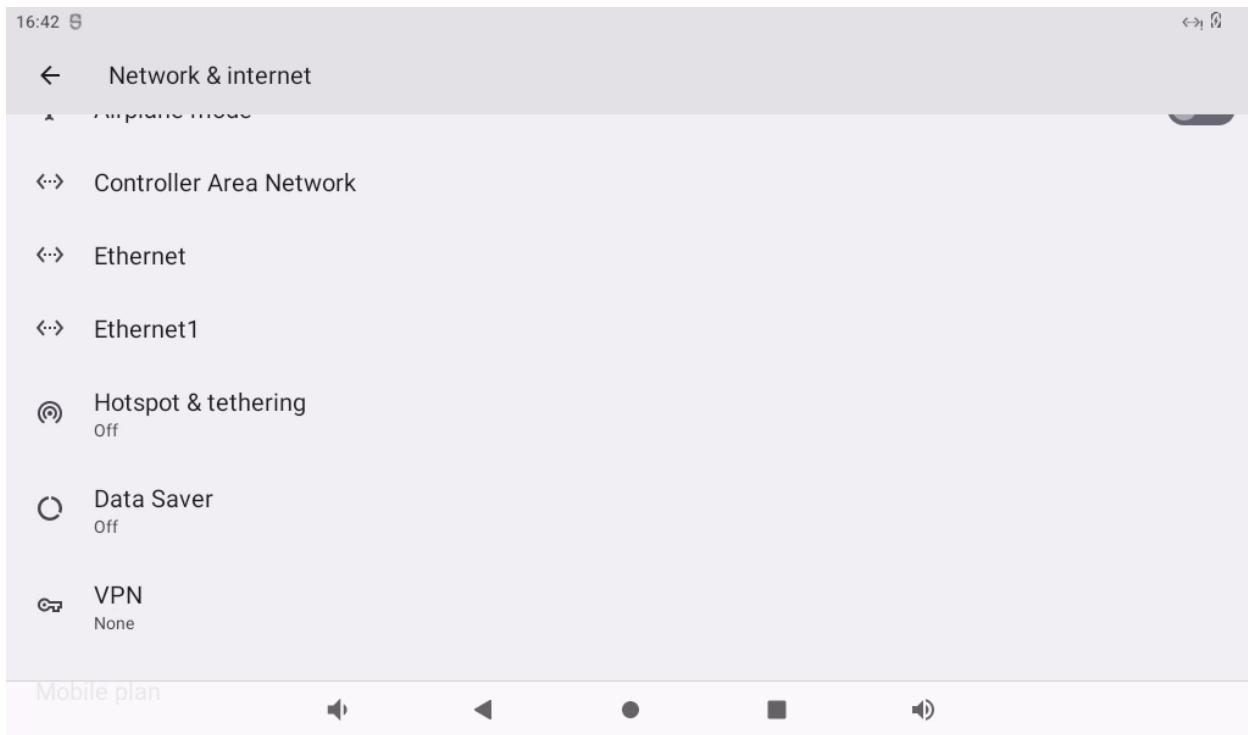
3.12 WiFi Hotspot Test

OK3588 supports the sharing of Ethernet or mobile networks through WIFI for WIFI hotspot testing. First, plug the network cable into the OK3588 ETH0 connector. Open Settings and click Network and Internet.

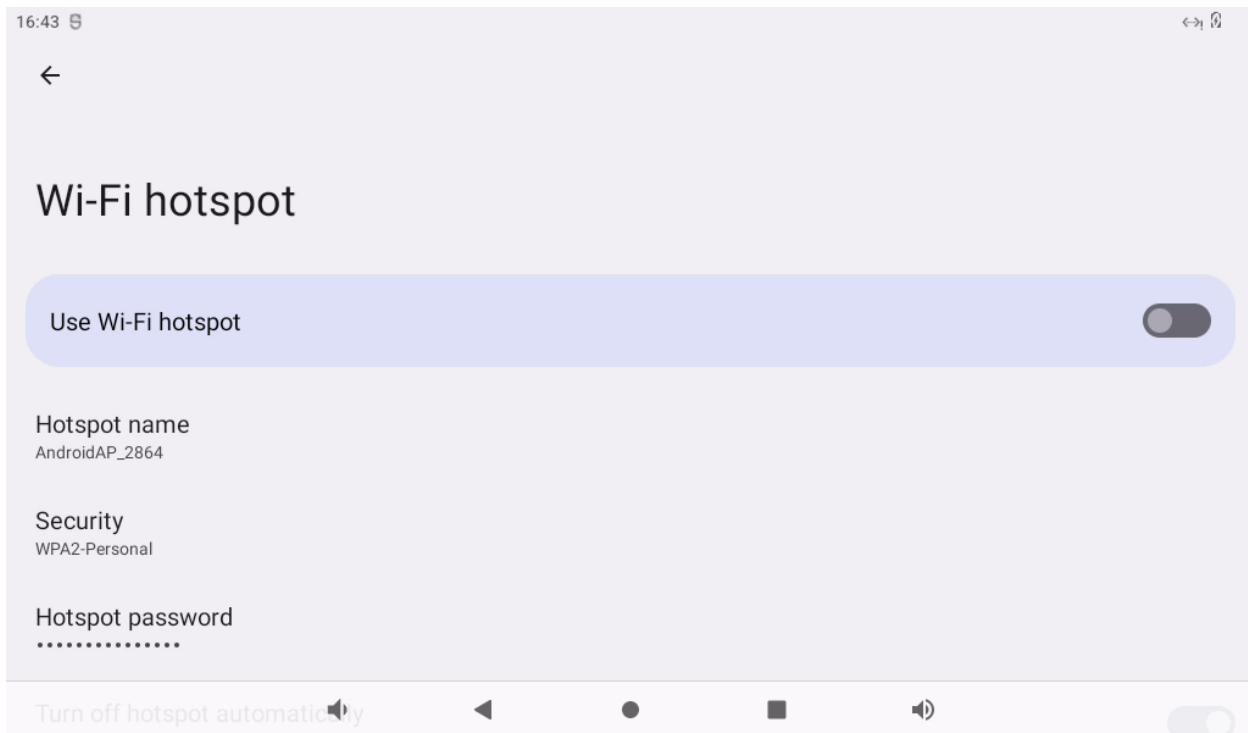
Click “Hotspot &tethering” :



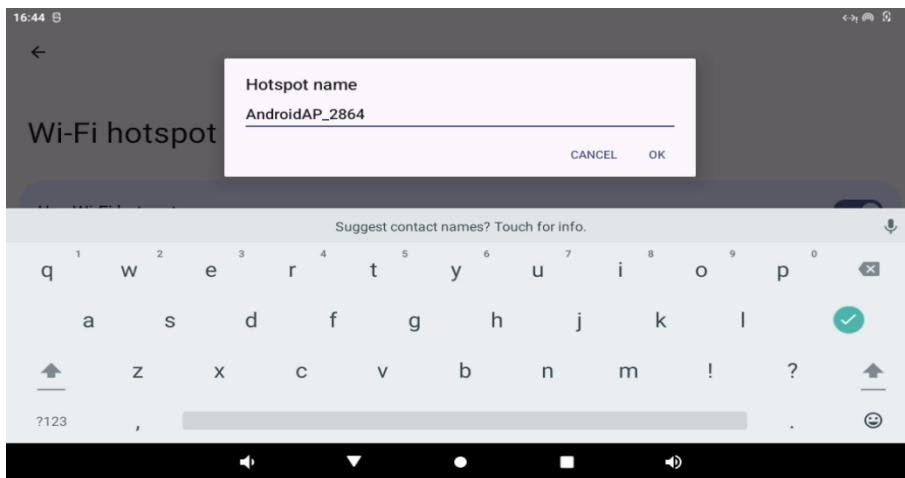
Click WLAN Hotspot:



Enable the WLAN hotspot and set the hot spot name and password:

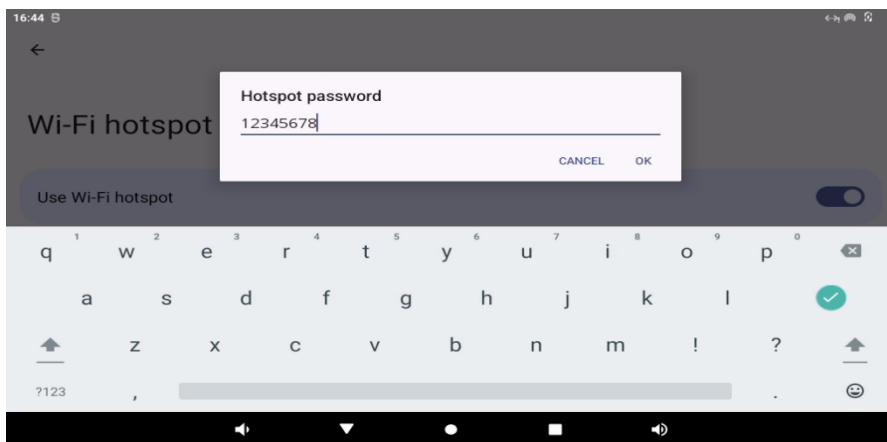


First set the hotspot name:



Click “Confirm” .

Set hotspot password:



Click “Confirm” .

After connecting to the hotspot through the mobile phone, you can surf the Internet normally.

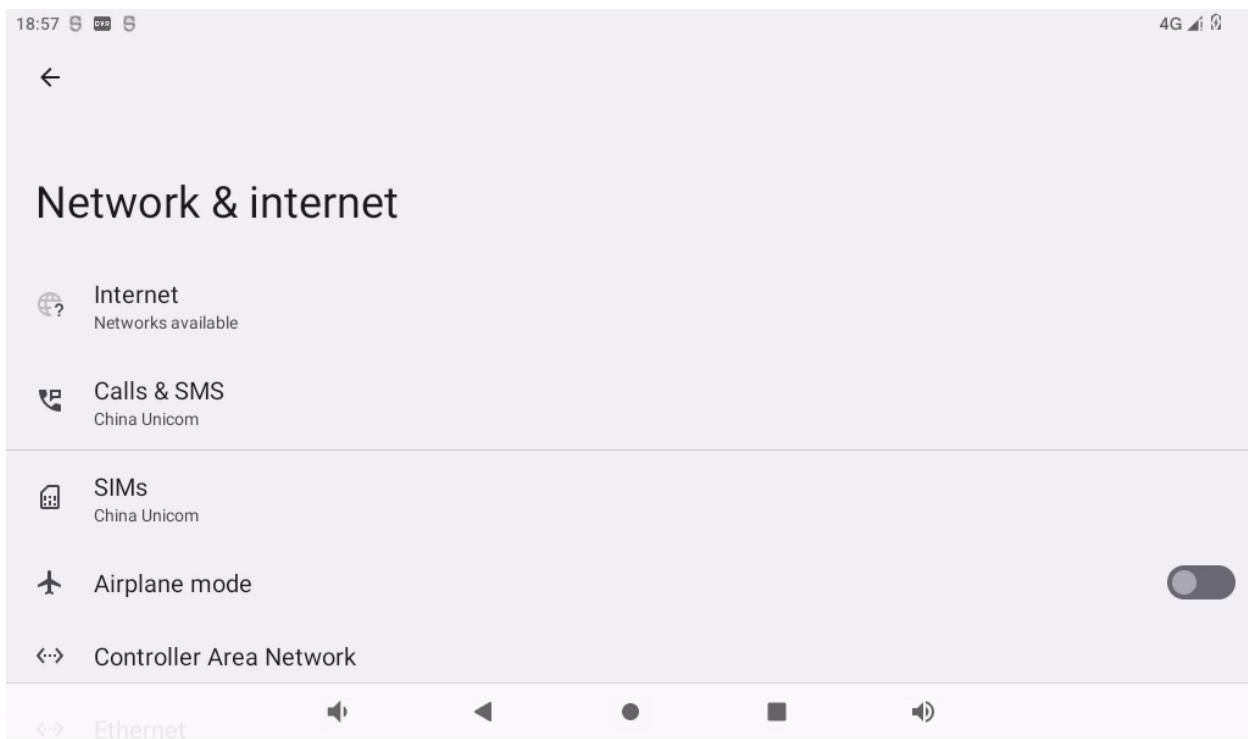
3.13 4G/4G Module Test

Description:

- When 4G and Ethernet exist at the same time, Ethernet is preferred by default; When 4G WIFI exists at the same time, WIFI is preferred by default. When both WiFi and Ethernet are present, Ethernet is prioritized by default;
- When testing 4G, unplug the wired network and turn off WiFi;
- When using the 4G module, dial the S2 to ON, and when using the 5G module, dial to the other end.

The OK3588 carrier board supports 4G modules (EM05) and 5G modules (RM500U, RM500Q). Before the test, please power off the development board, connect the 4G/5G module and insert the SIM card (pay attention to the direction of the SIM card), and start the development board.

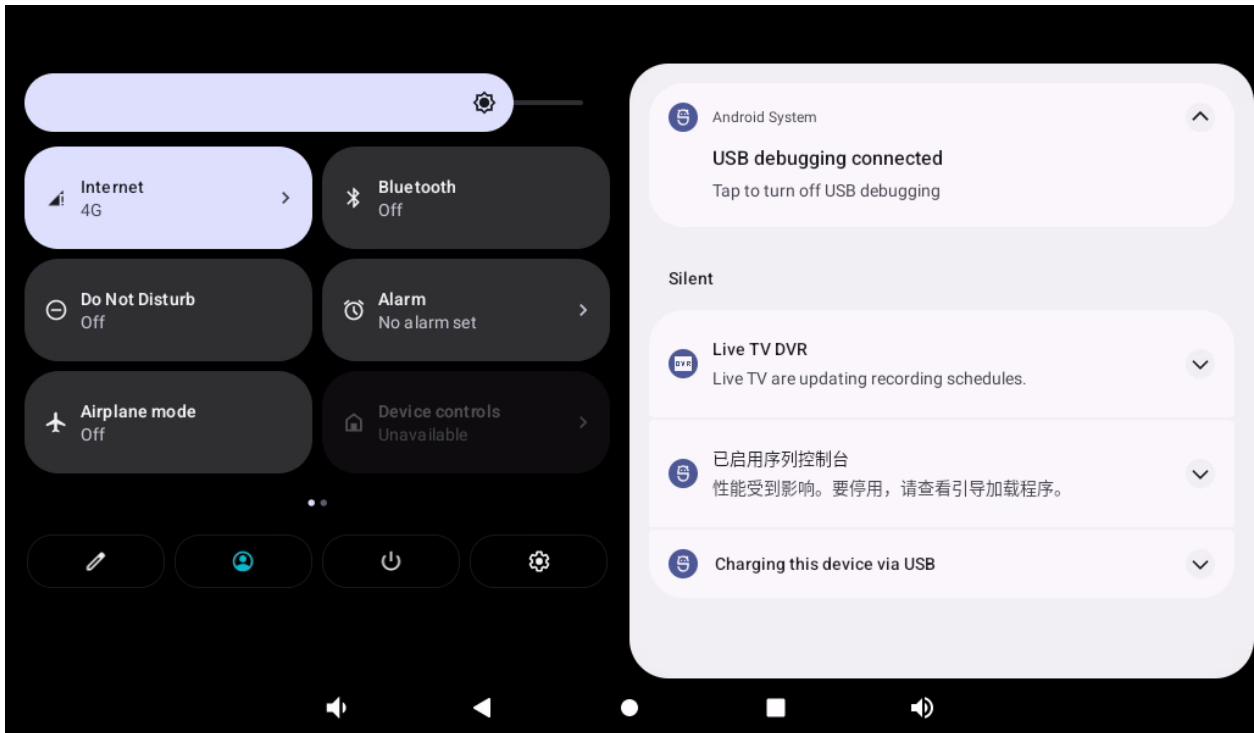
Open Settings, select “Network and Internet”, and click “SIM card” :



The default mobile network is on:



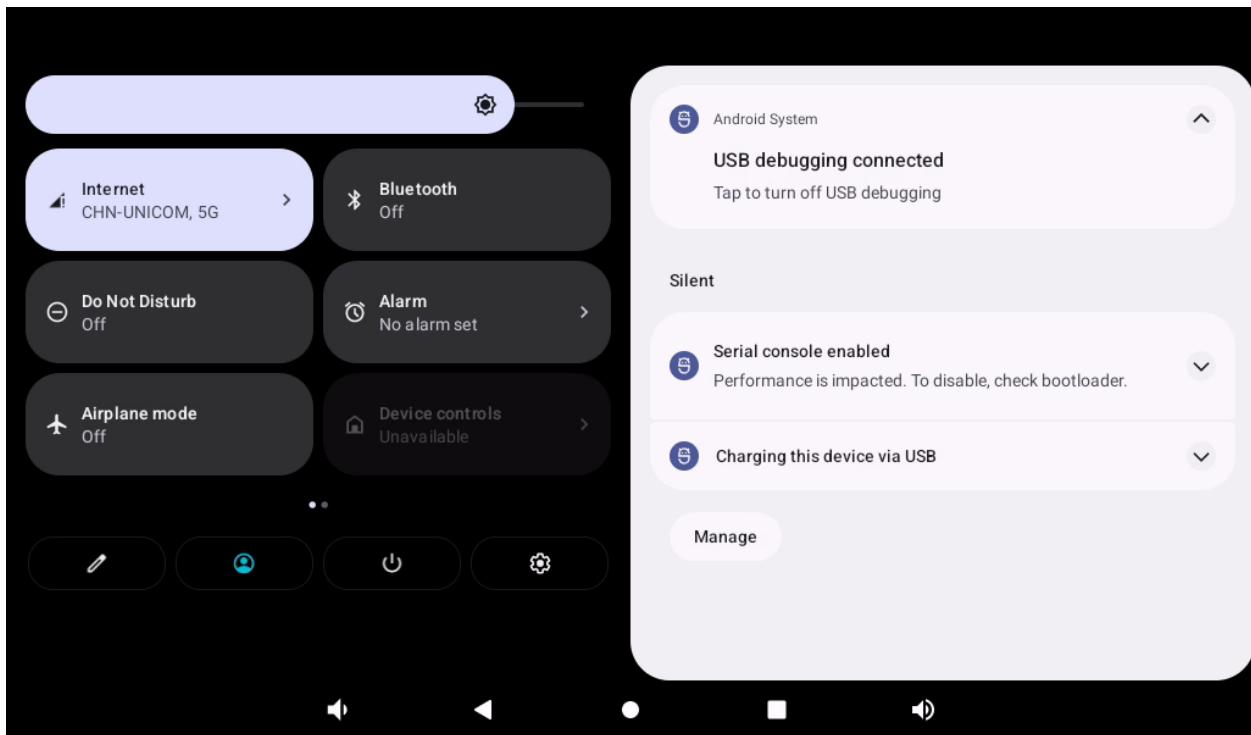
Link 4G is displayed in the drop-down menu for successful links.



After successful connection, you can open the browser for network test:



The test method of 5G is the same as that of 4G, and the difference is that the icon display is different:



3.14 Bluetooth Test

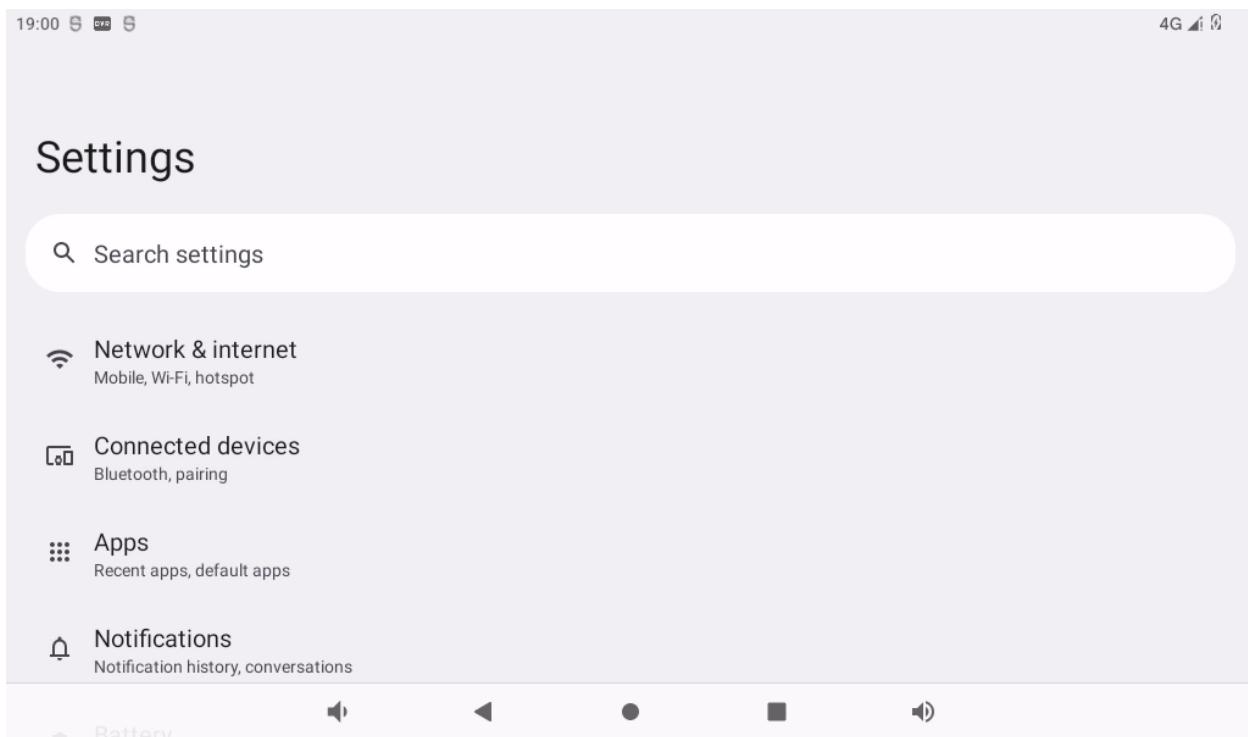
**Description: **

- The current system does not support iPhone Bluetooth connection;
- The Bluetooth function test of OK3588 platform uses the WiFi & Bluetooth integrated module, which supports the connection of Bluetooth devices as the main device to transmit/receive files.

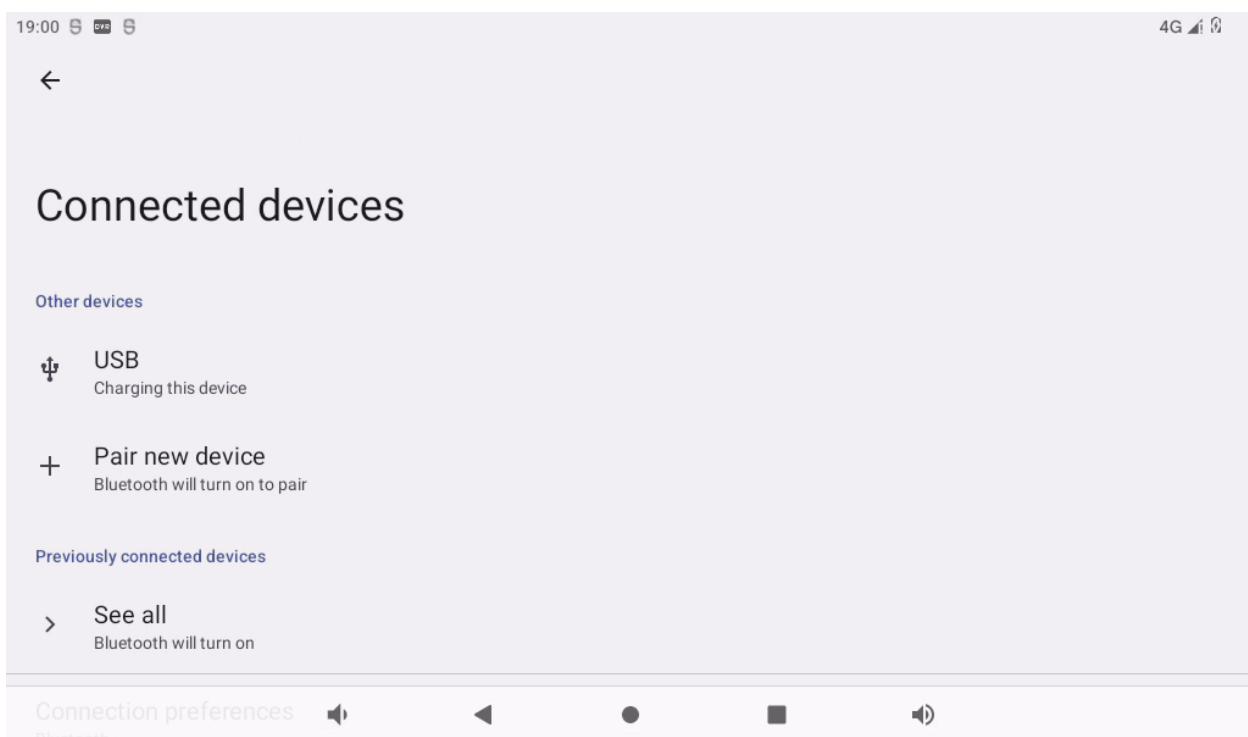
The testing method is as follows:



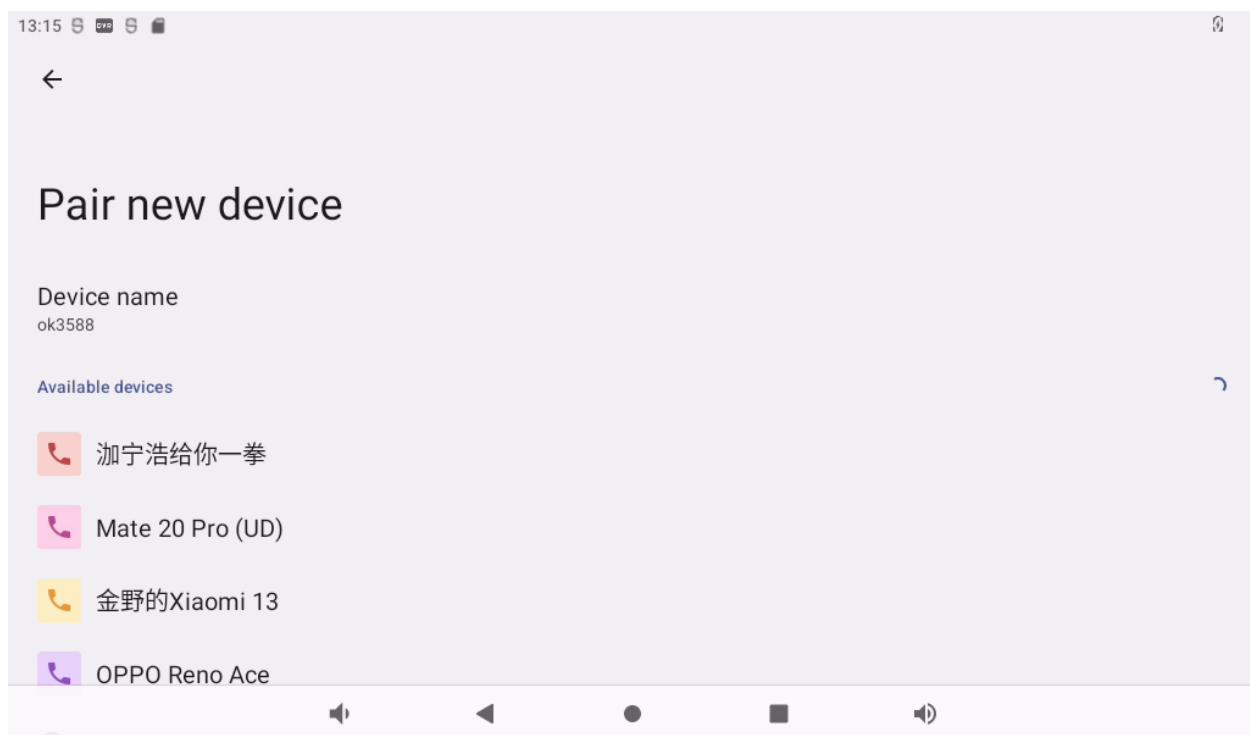
Click “ ” , on the application interface to enter the setting interface:



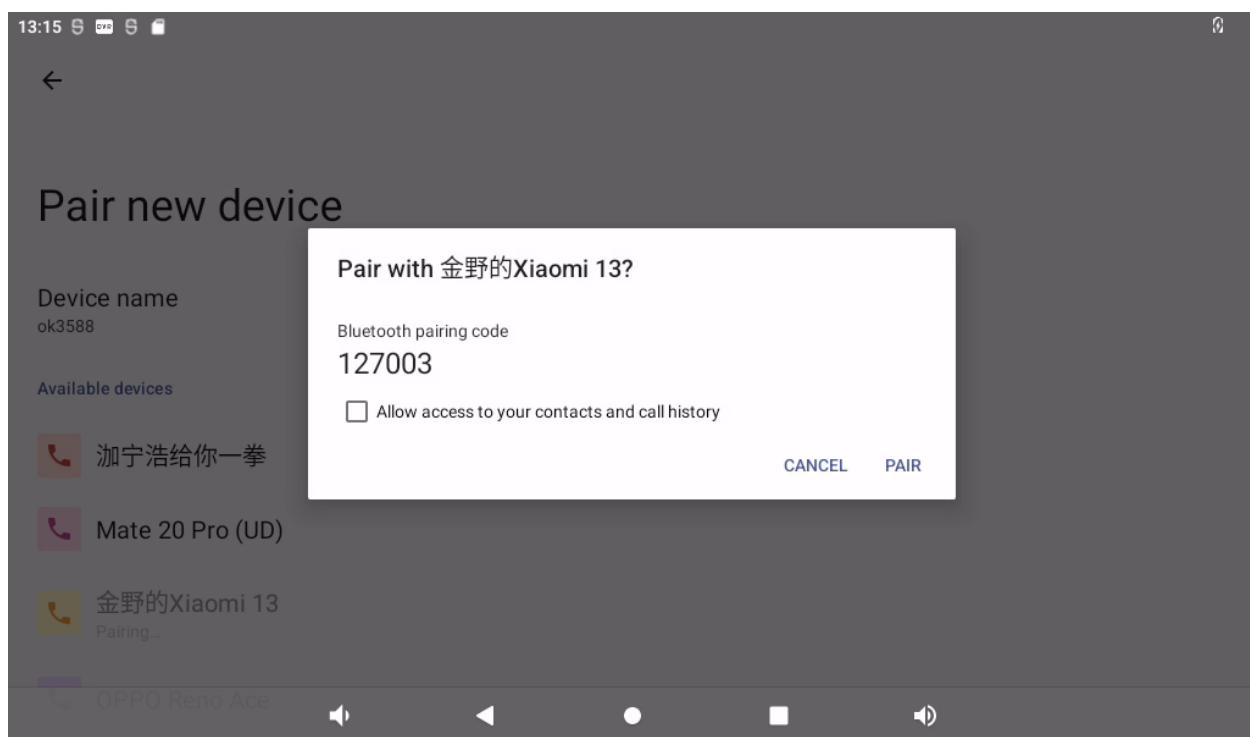
Click “Connected device” to enter the Bluetooth setting interface.

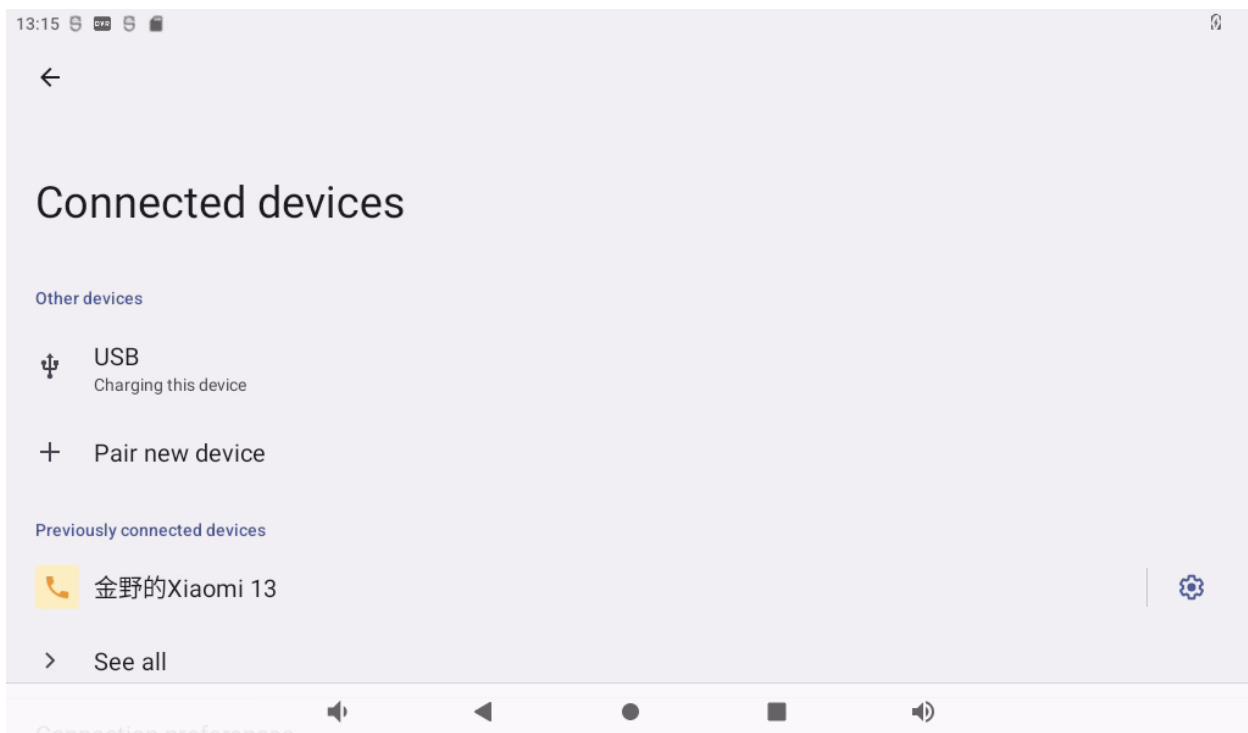


Click “+ pair with new device” , open PC Bluetooth to scan at the same time, and click the Bluetooth device to be connected.



Click “Pairing” , the mobile phone performs the corresponding pairing operation, and the interface of successful Bluetooth connection displays:

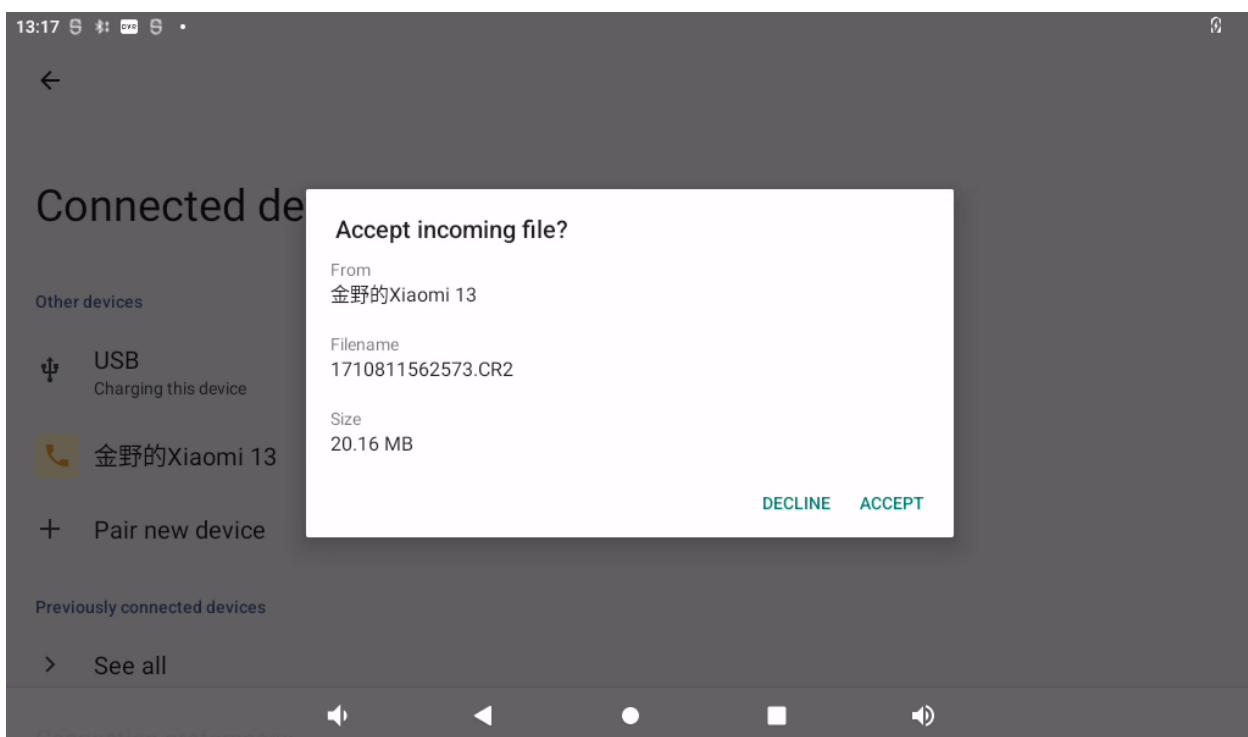




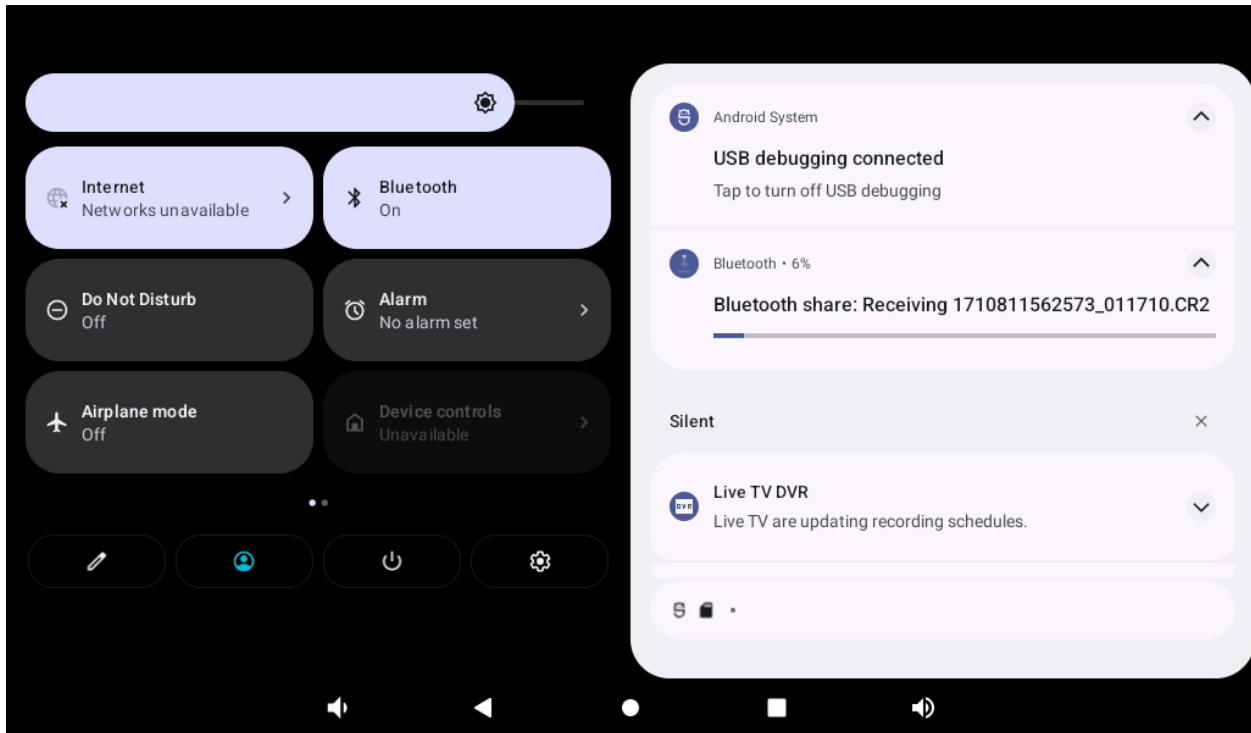
1. File transfer file test:

Accept the file:

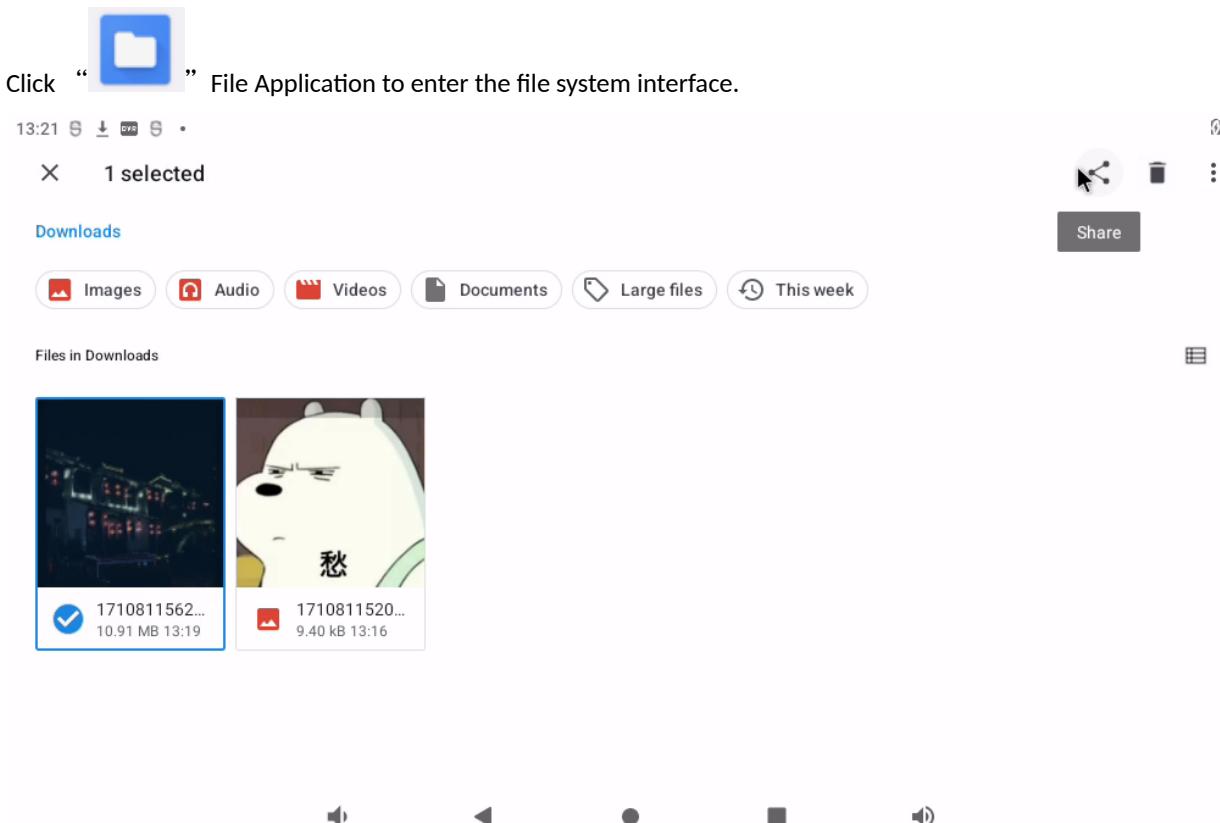
The mobile phone shares photos to the OK3588 using Bluetooth; click “Accept” will start the outgoing.



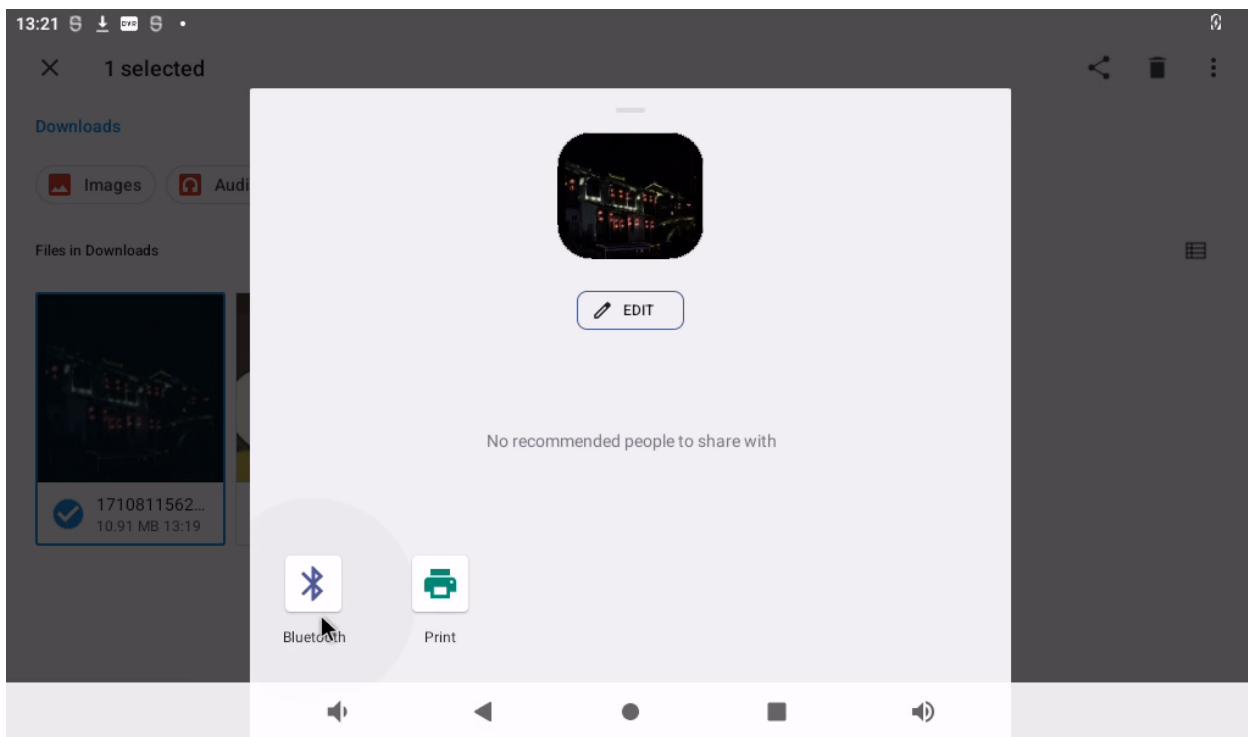
The transfer progress will be displayed in the prompt bar, and you can view the picture after the transfer is completed.



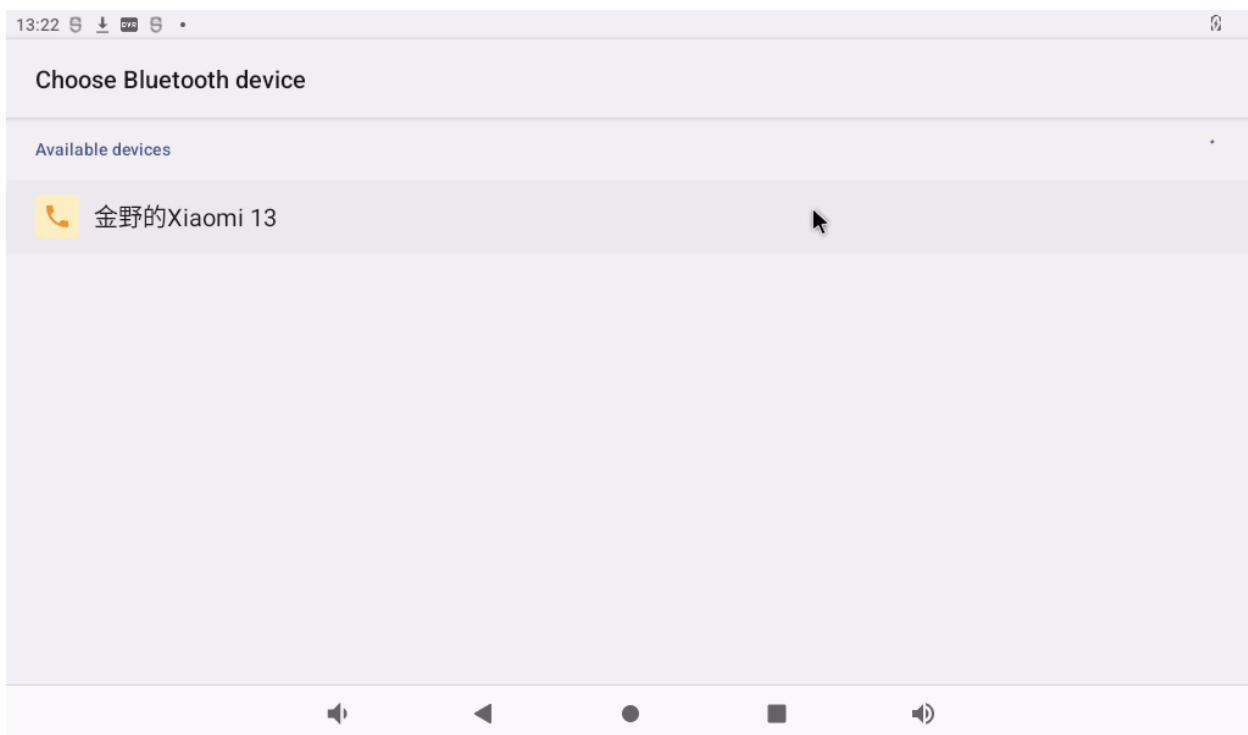
2. File transfer test:



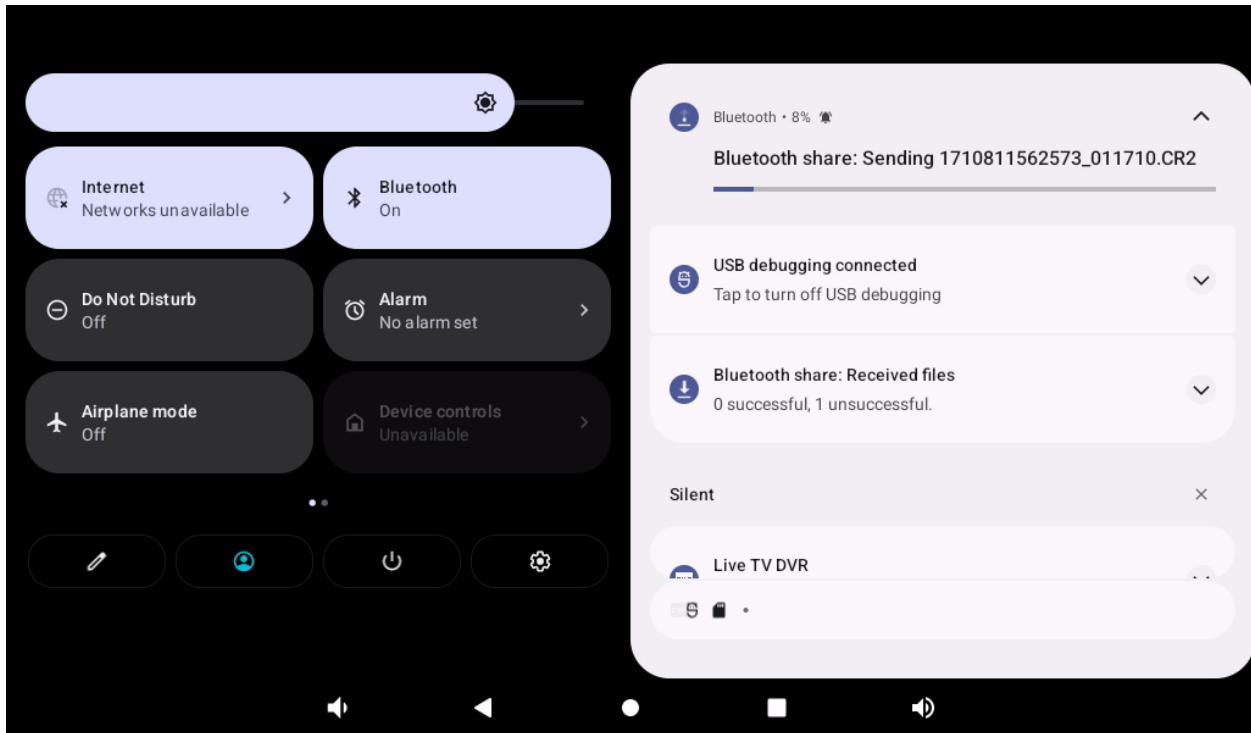
Select an image and click the share button in the top right corner “” .



Send via Bluetooth.



Select the previously paired device, select the phone to receive the file, and the Bluetooth transfer will start.



You can view the transfer progress in the notification bar, and after the transfer is complete, you can view the received images on your phone.

3.15 Key Test (Sleep Wake-up)

There are 8 keys on the development board, including VOL+, VOL-, MENU, ESC, HOME, PWRON, RESET and Maskroom.

Key	Function
Recovery/VOL+	VOL+
VOL-	VOL-
PWRON	Wake up from sleep and power on/off
Maskroom	Work with RESET to enter maskrom mode.
RESET	RESET
MENU	Pop-up menu Home screen settings, Widget, Wallpaper
ESC	Return

The default factory setting is the non-hibernation state. At this time, press the PWRON key lightly to turn off the screen and enter the hibernation state (note that the carrier board cannot be inserted into the wake-up source such as USBOTG). The hibernation print information is as follows:

```
de345678INFO: PMU1_PWR_CON(0x1) PMU1_CRU_PWR_CON(0x2f) PMU1_WAKEUP_INT_CON(0x100)
PMU2_BUS_IDLE_ST(0x27fffff 0x0) PMU2_BUS_IDLE_ACK(0x27fffff 0x0) PMU2_PWR_GATE_
→ST(0x6fffffff 0x0)
PMU2_BUS_IDLE_CON(0x0 0xfd80 0xf007) PMU2_BIU_AUTO_CON(0xfffff 0xfffff 0x7)
PMU2_PWR_GATE_CON(0x0 0x9000 0x3)
PMU2_VOL_GATE_CON(0x7 0x0 0x3)
PMU2_QCHANNEL_PWR_CON(0x0) PMU2_QCHANNEL_STATUS(0xfe0007f)
PMU1_DDR_PWR_CON(0x747 0x747 0x747 0x747)
PMU1_DDR_PWR_SFTCON(0x900 0x900 0x900 0x900)
```

(continues on next page)

(continued from previous page)

```

PMU1_PLLPD_CON(0xffff 0x3)
PMU2_DSU_PWR_CON(0x3)
PMU2_CORE_PWR_CON0(0x1 0x1)
PMU2_CORE_AUTO_PWR_CON0(0x0 0x0)
PMU2_CLUSTER_IDLE_CON(0x75)
INFO: PMU0_PWR_CON(0x0) PMU0_WAKEUP_INT_CON(0x0)
PMU0_DDR_RET_CON(0x0 0x0)
PMU1_GRF_SOC_CON2(0x7777) PMU0_GRF_OS_REGS9(0xf2acf6f4)
S

```

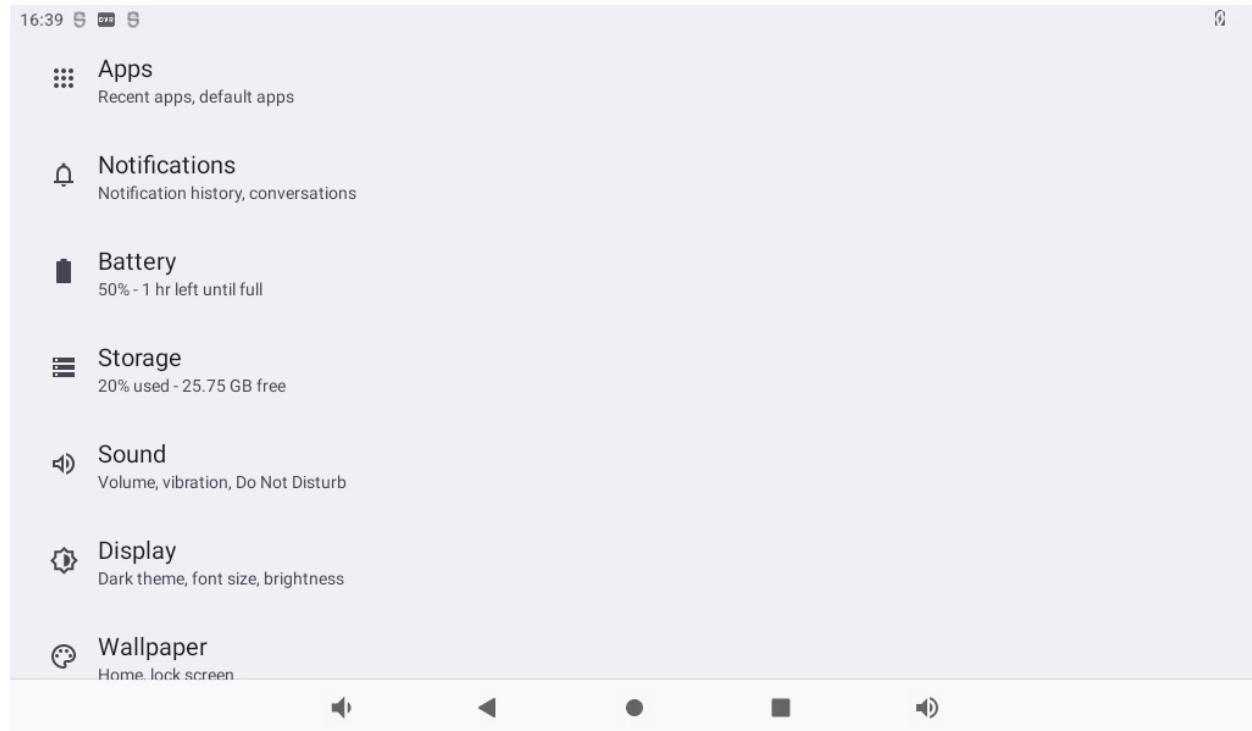
In the sleep state, press the PWRON key again to wake up the CPU. Press and hold PWRON to shut down the device. The other buttons have simpler functions, so please test them yourself.

3.16 TF Card and USB Storage Test

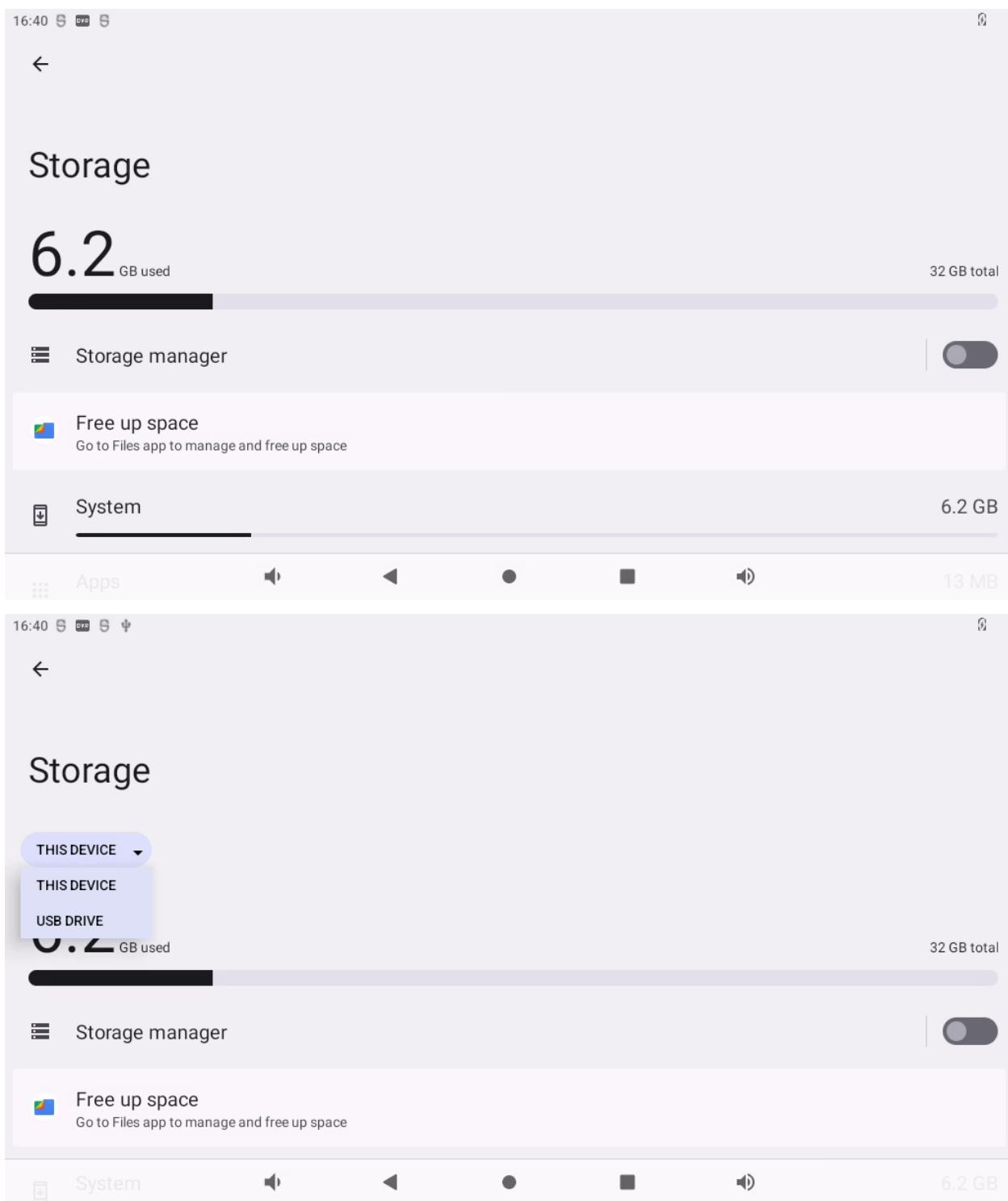
It is a test of TF card and USB storage device. Insert the USB device into the OK3588 USB Host port. The system will automatically detect the insertion of the USB flash drive.



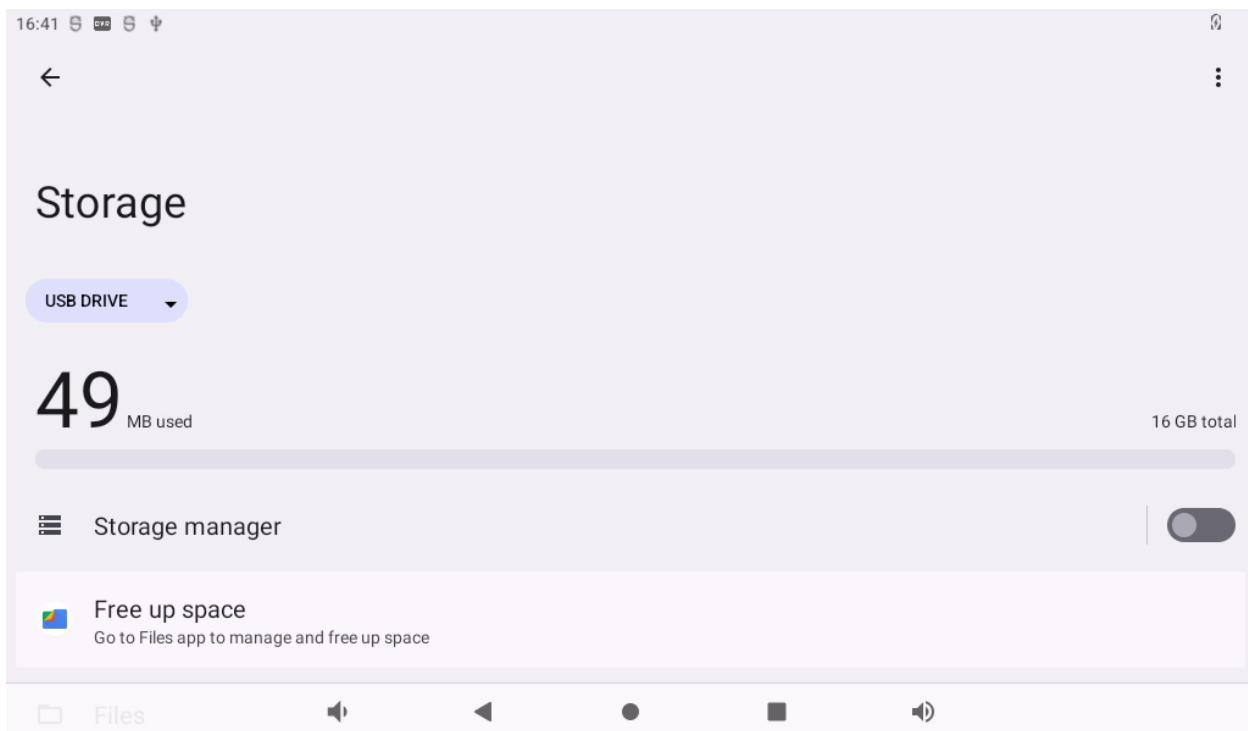
Click “ ”, on the application interface to enter the setting interface:



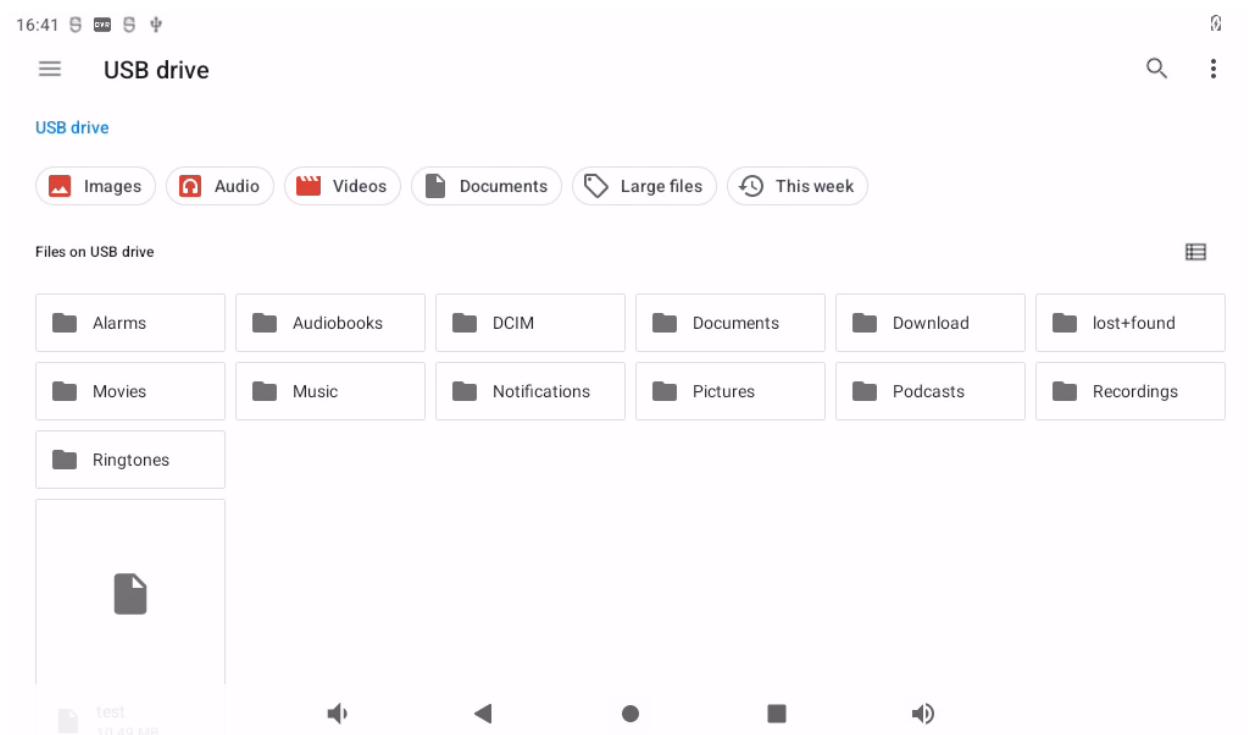
Click “Storage” to view the internal storage device and the inserted U disk device:



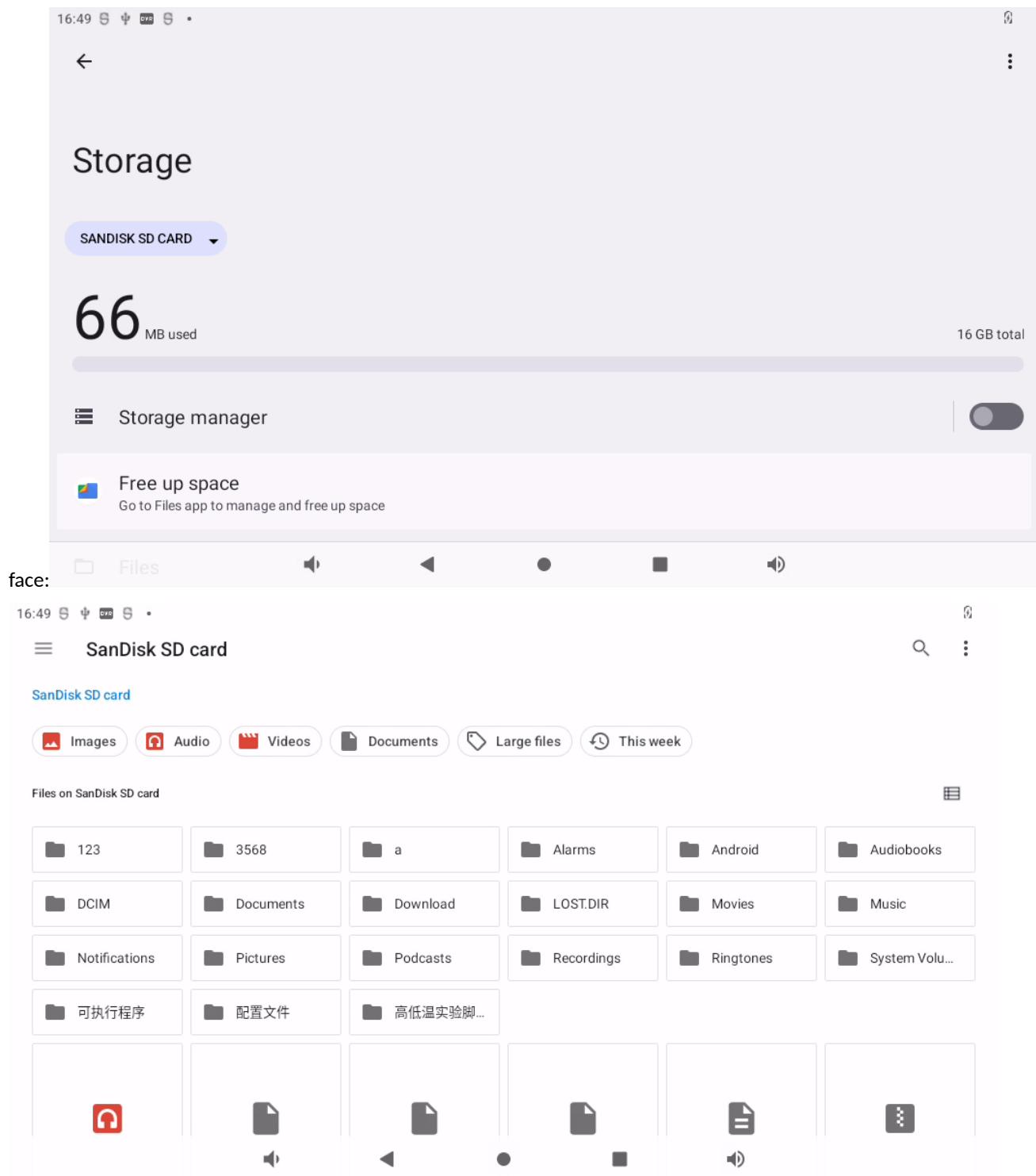
Click “MASS U disk” to view the contents of the U disk for reading and writing:



Click the “File” .



The TF card reading and writing method is the same as the U disk reading and writing test method. Insert the TF card into the TF card slot, and the system will automatically detect the insertion of the TF card. You can also view the contents of the TF card in the storage inter-



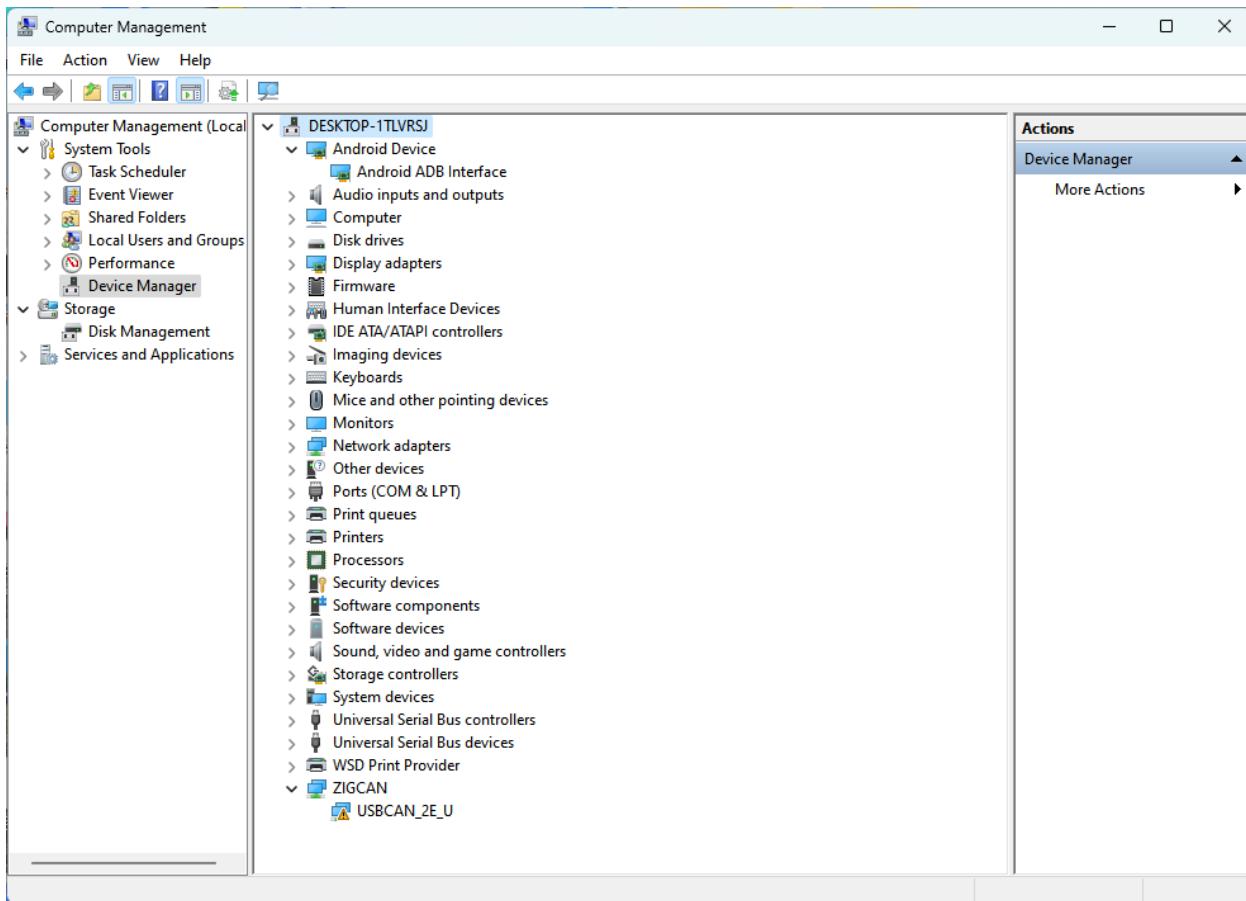
3.17 USB Mouse Test

Once the system is running, you can plug in a USB mouse into the USB host. You will then see the mouse cursor “”, within the interface, and you can navigate and operate the Android system using the mouse.

3.18 USB OTG Interface Test

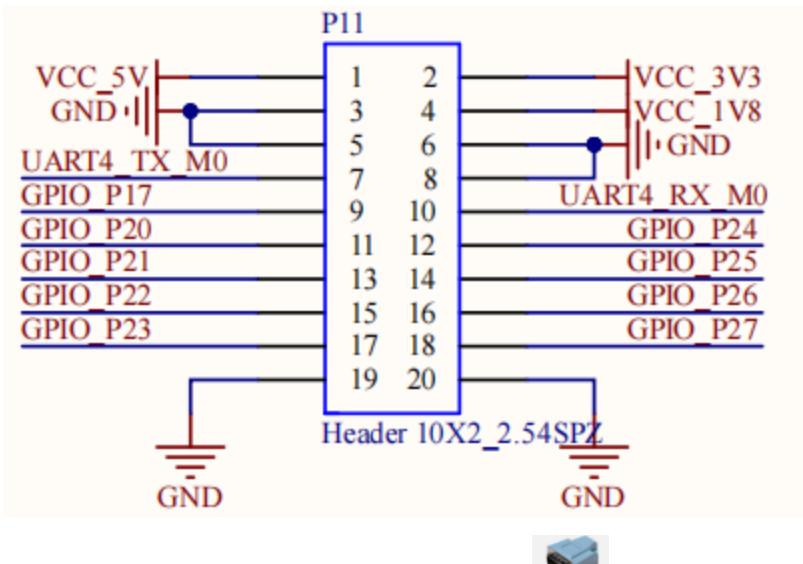
The OK3588 development board supports USB OTG functionality.

Typec0 of the development board is connected to the computer through the otg cable, and the computer will recognize the board as follows:

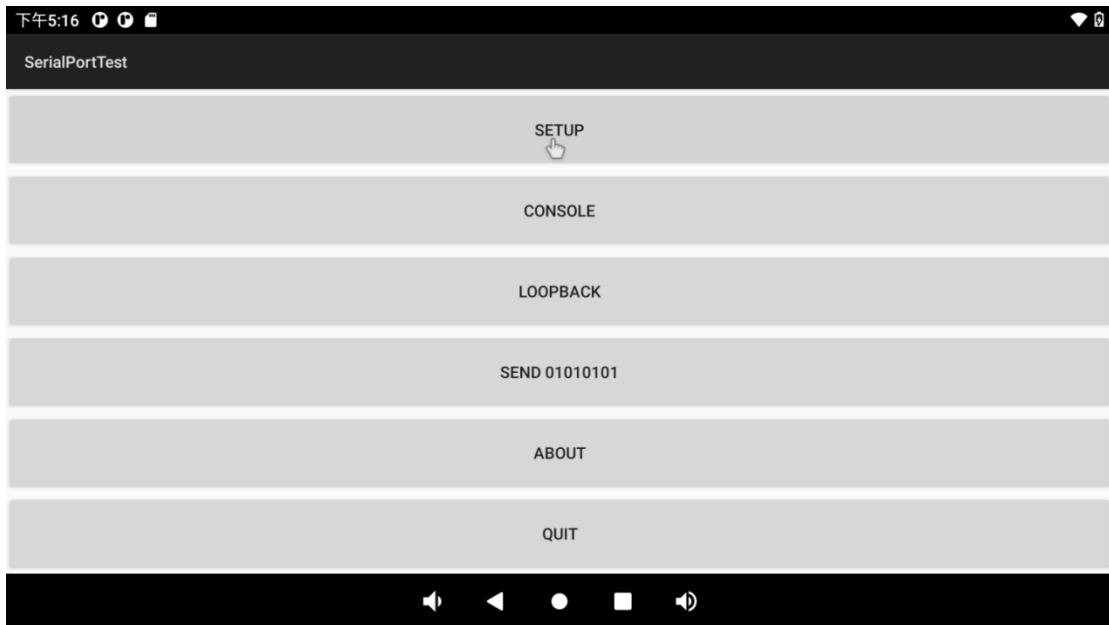


3.19 Serial Port Test

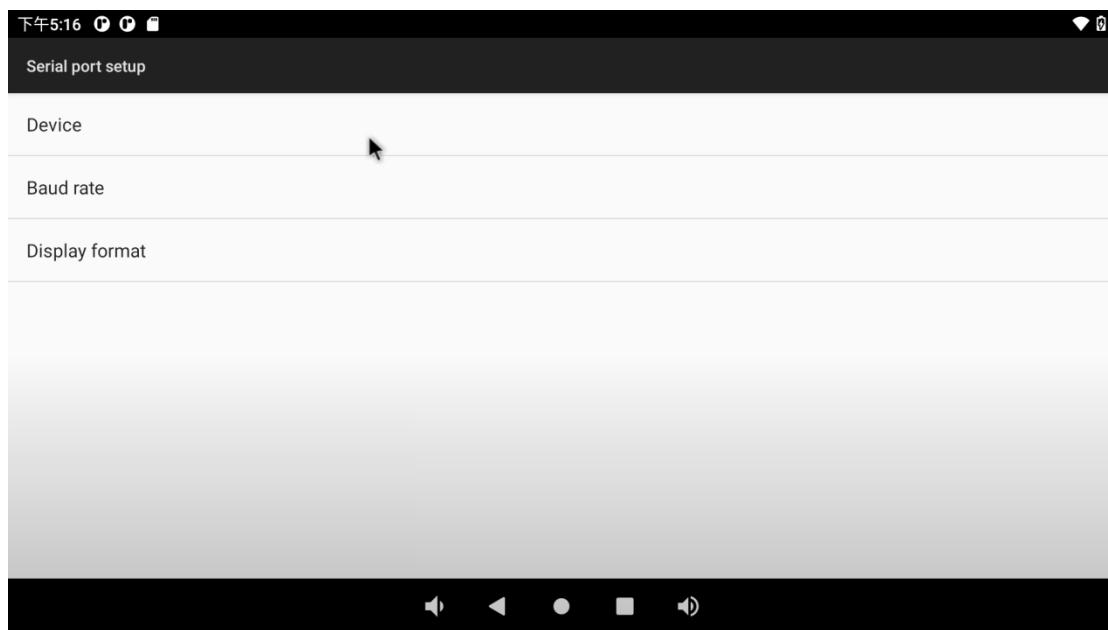
UART2, UART4, UART6, UART9, a total of four serial ports led out from the OK3588 carrier board; UART2 for debugging serial port, UART6 for Bluetooth serial port, and UART9 for 485 serial port. The default device names of UART4 and UART9 in the development board are ttyS4 and ttyS9 respectively. Take the test of UART4 serial port as an example, short-circuit the receiving and transmitting pins of UART4 according to the schematic diagram of the development board, corresponding to PIN7 and PIN10 respectively.



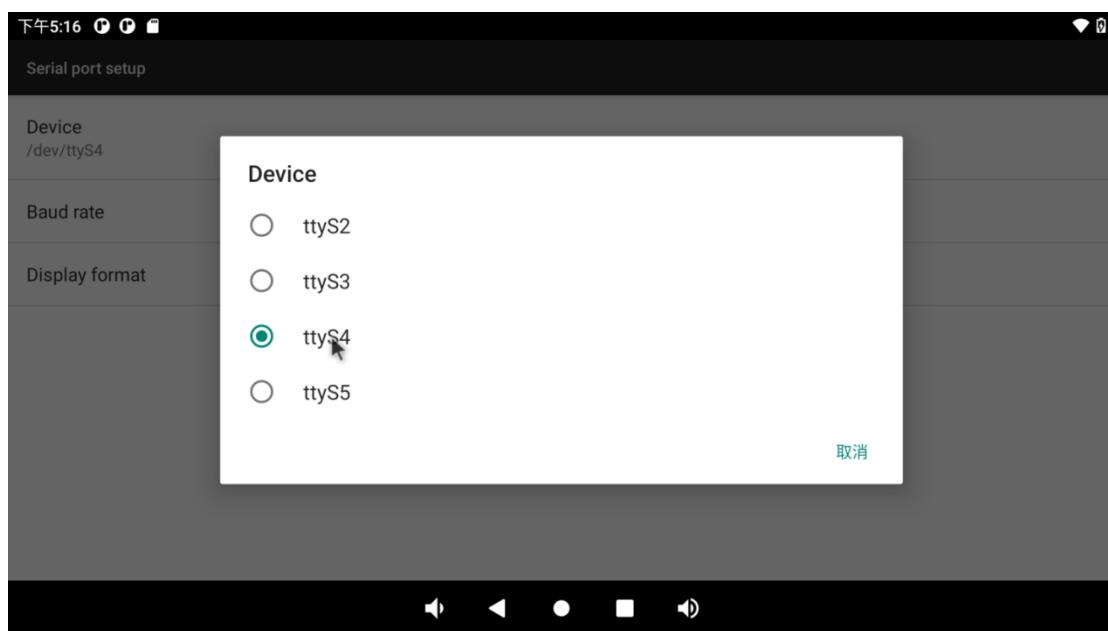
Open the serial port test program on the desktop [SerialPortTest](#).

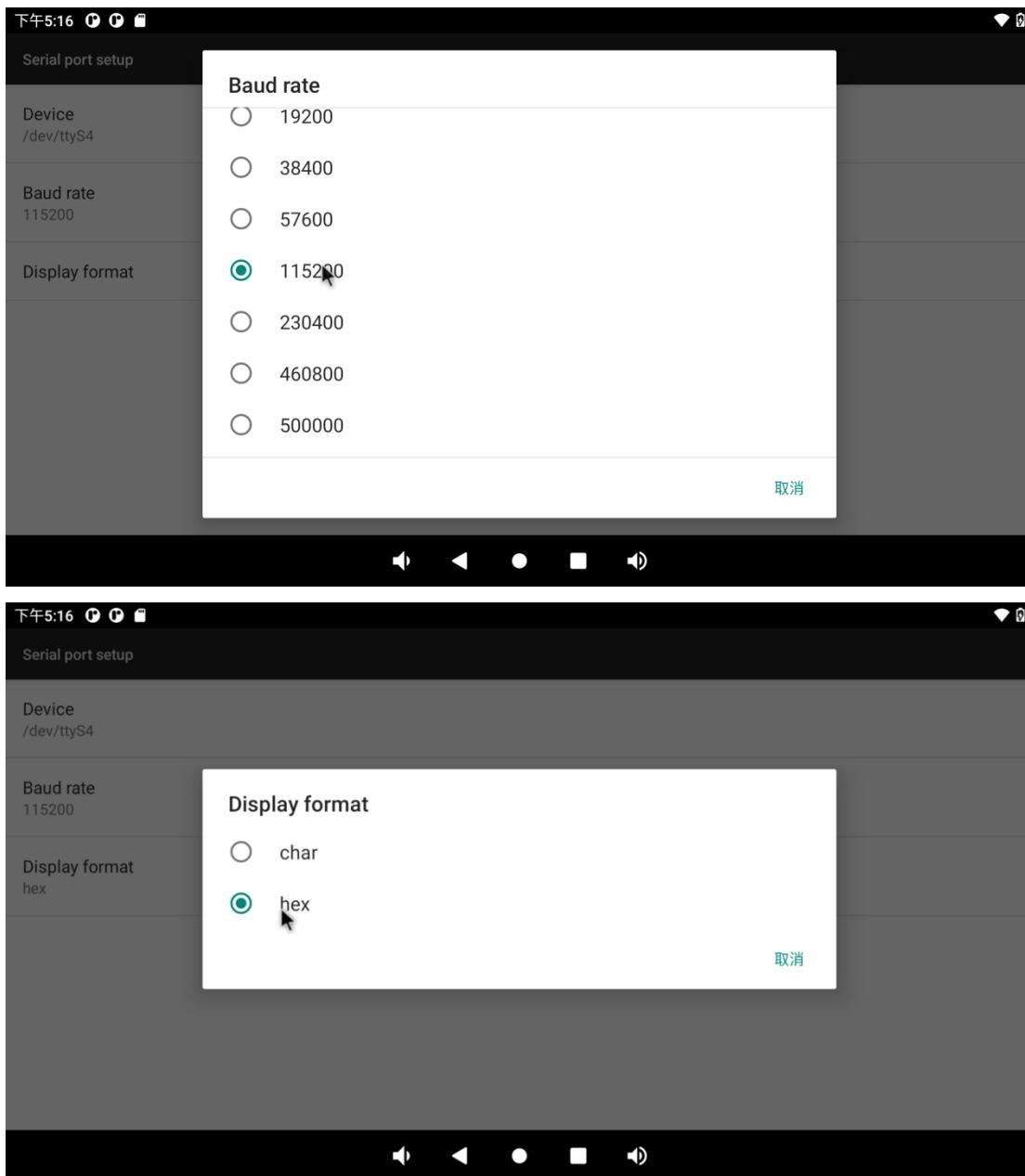


Click the “SETUP” button:

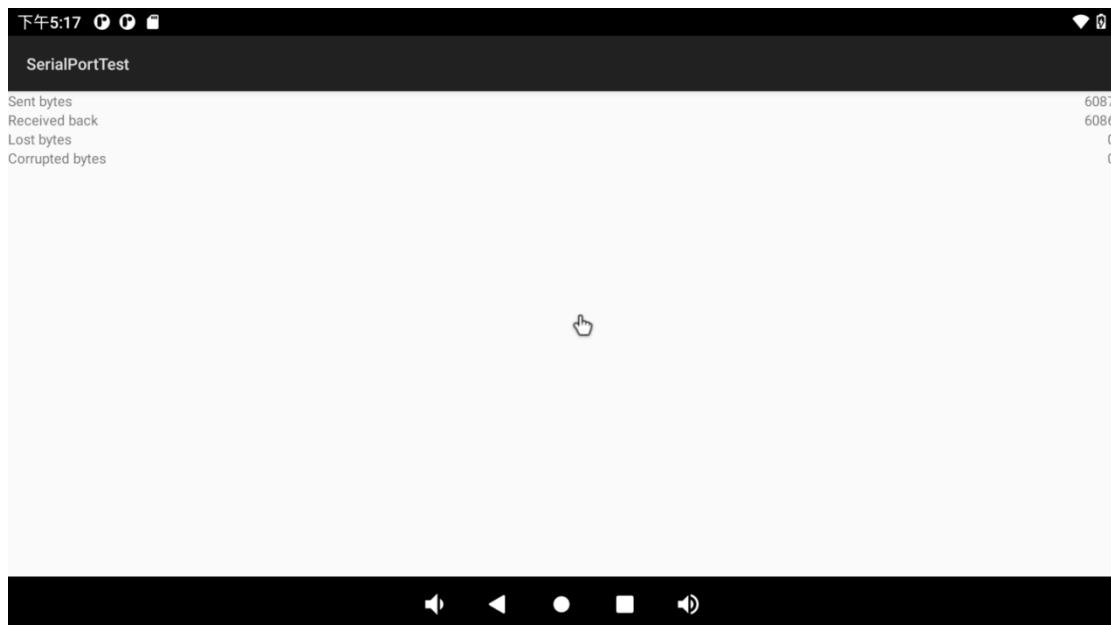


Set serial port device, baud rate and display format:

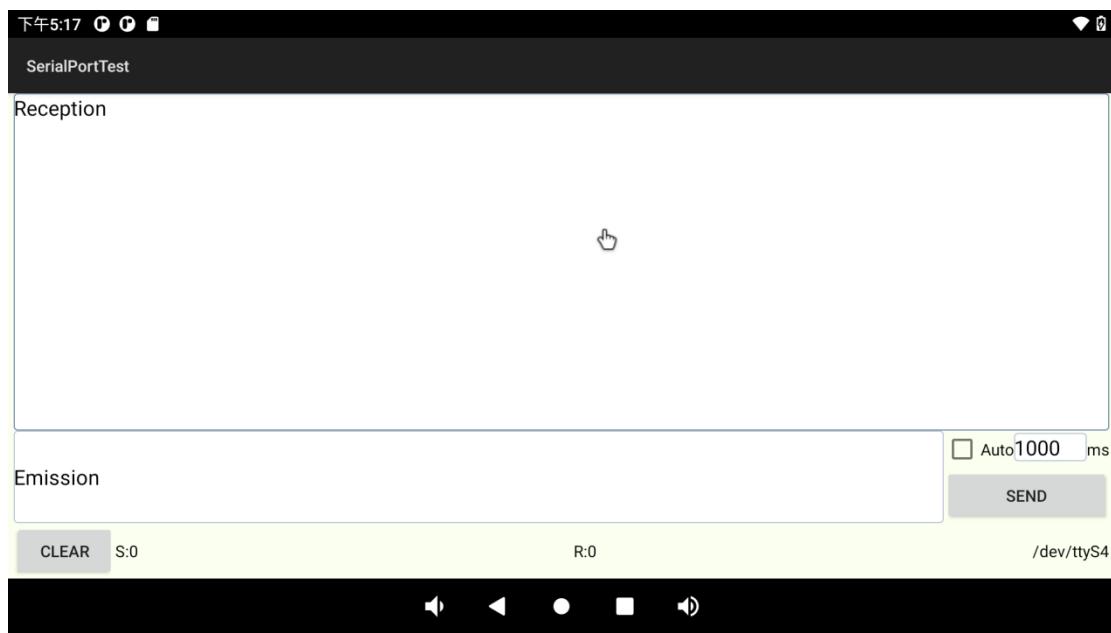


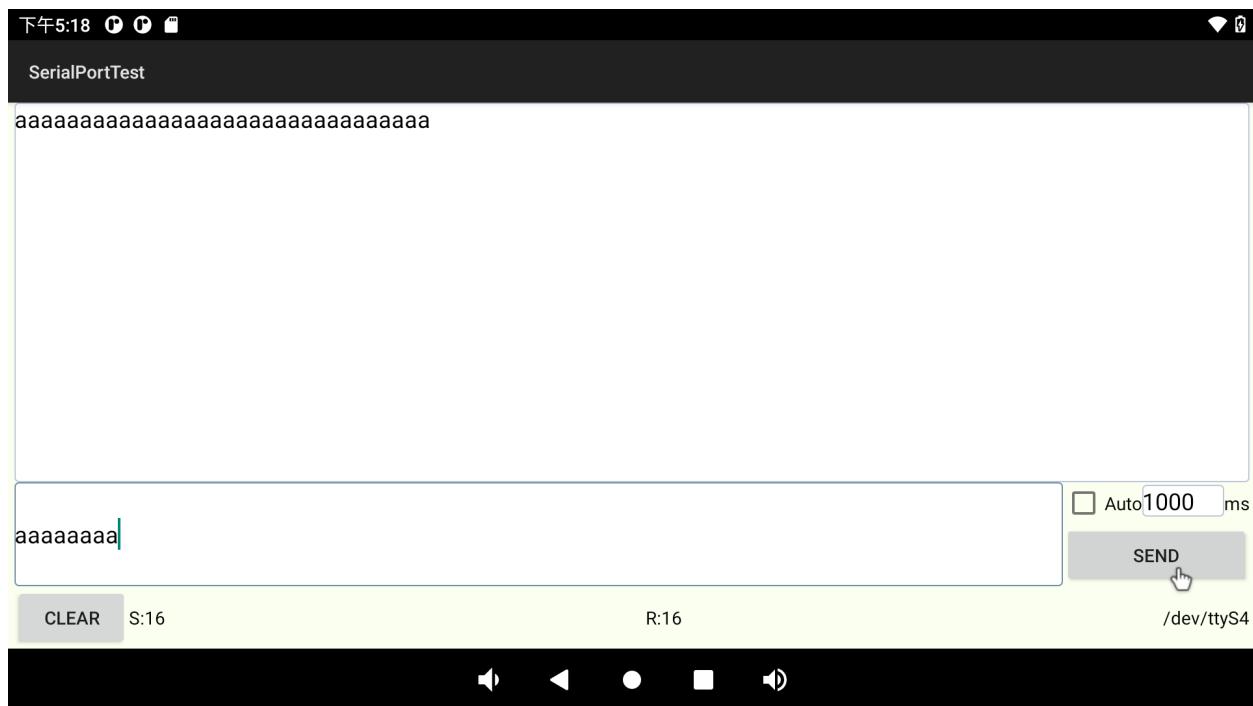


Then click the “Loopback” option in the previous menu to perform the loopback test.

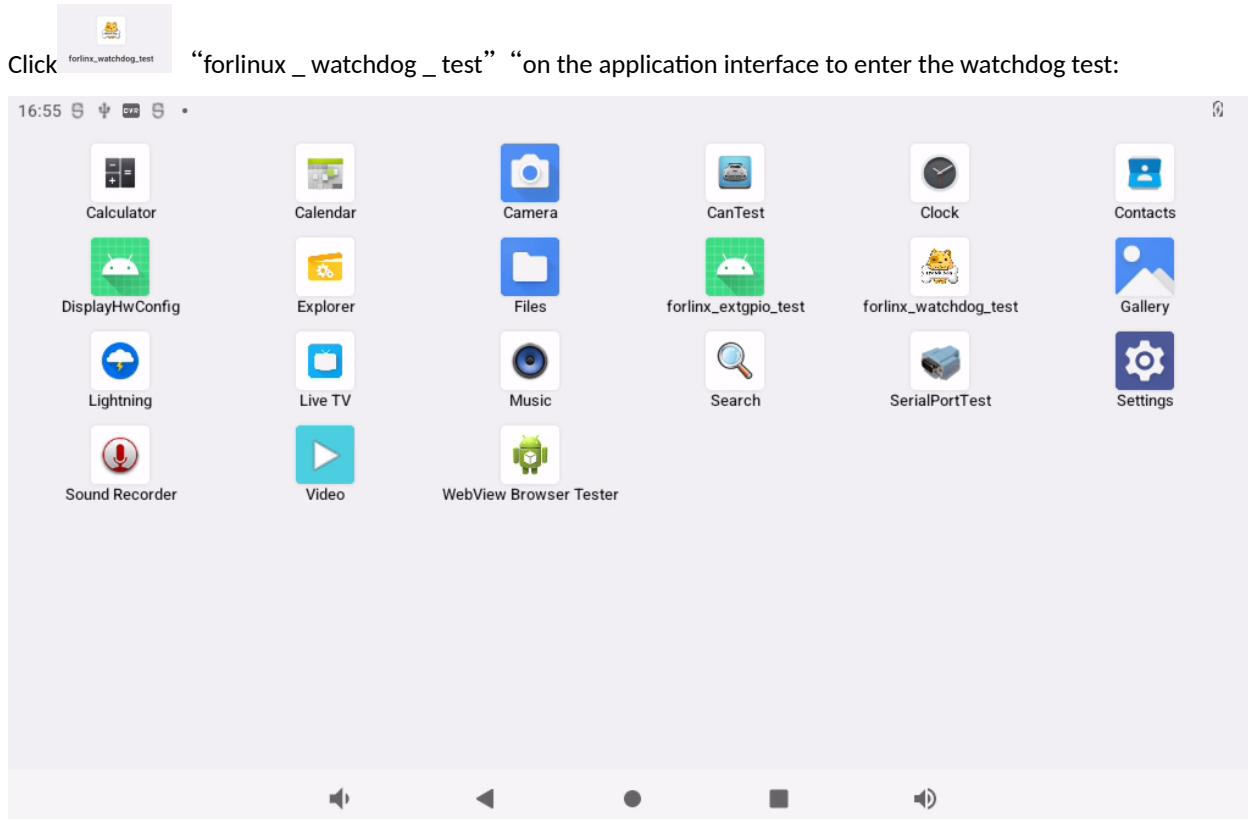


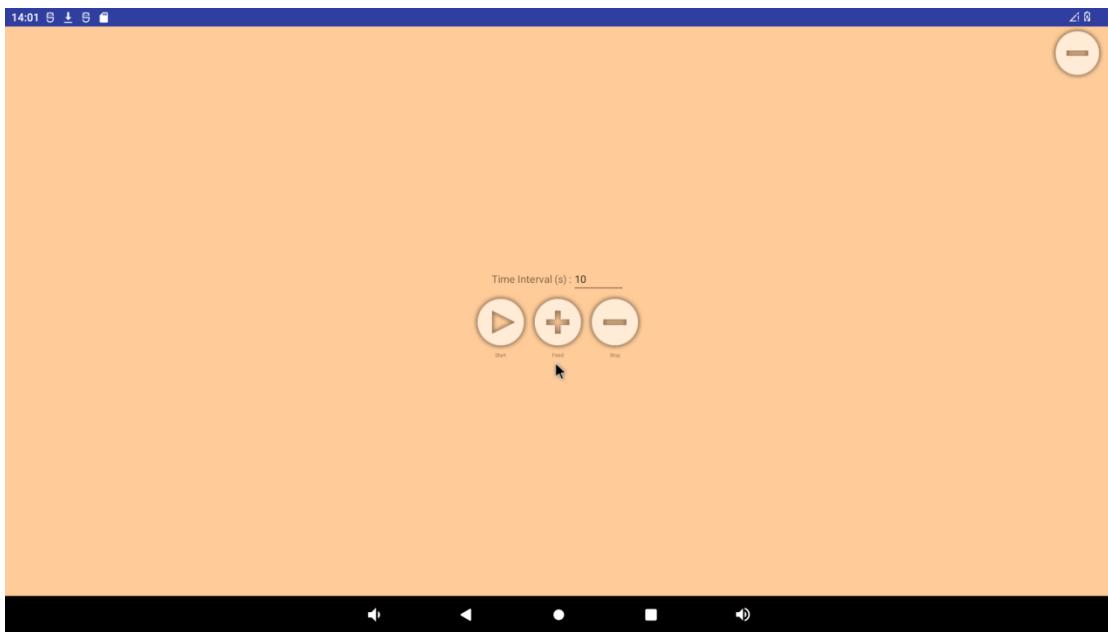
Click the “CONSOLE” option in the previous menu to perform the send-receive test:



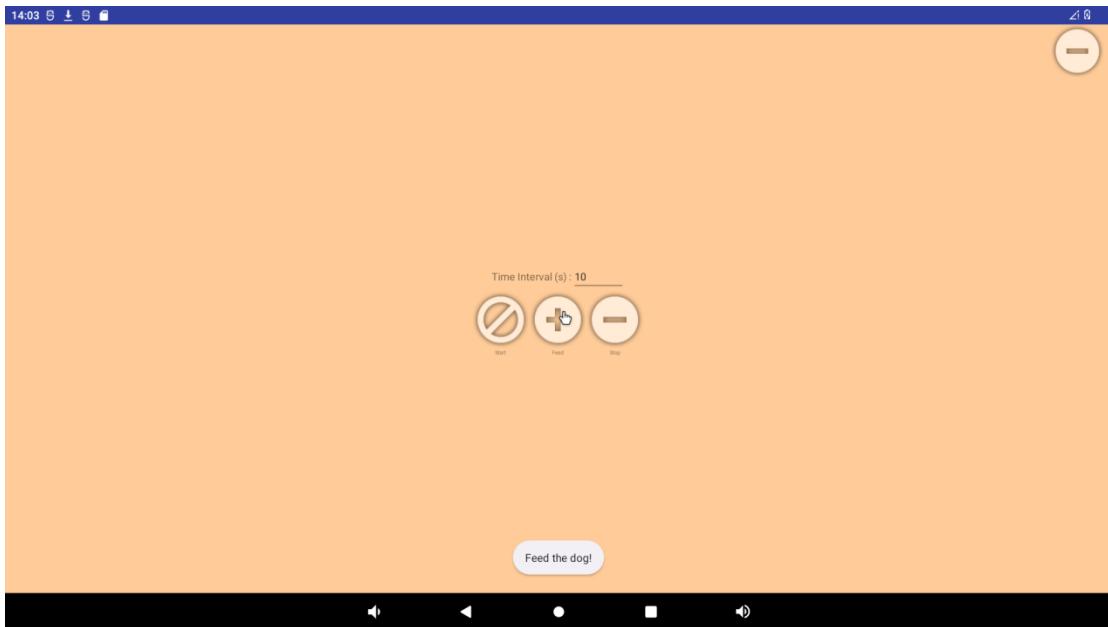


3.20 Watchdog Test

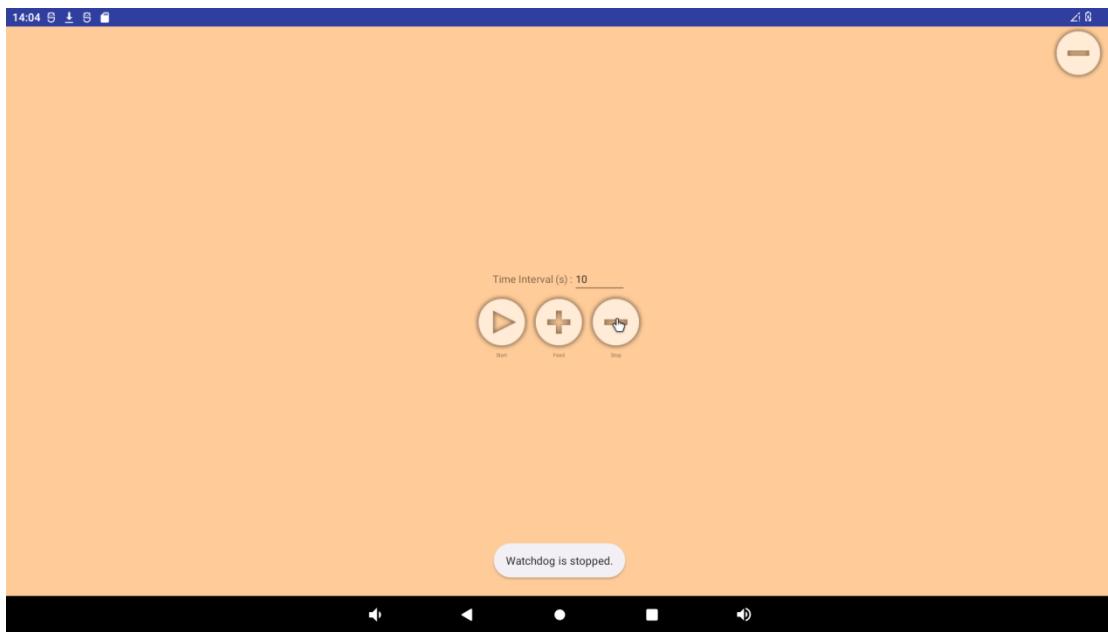




There are three buttons on the interface: “start” , “feed” and “stop” . Click “start” to see the dog and “feed” to feed the dog:



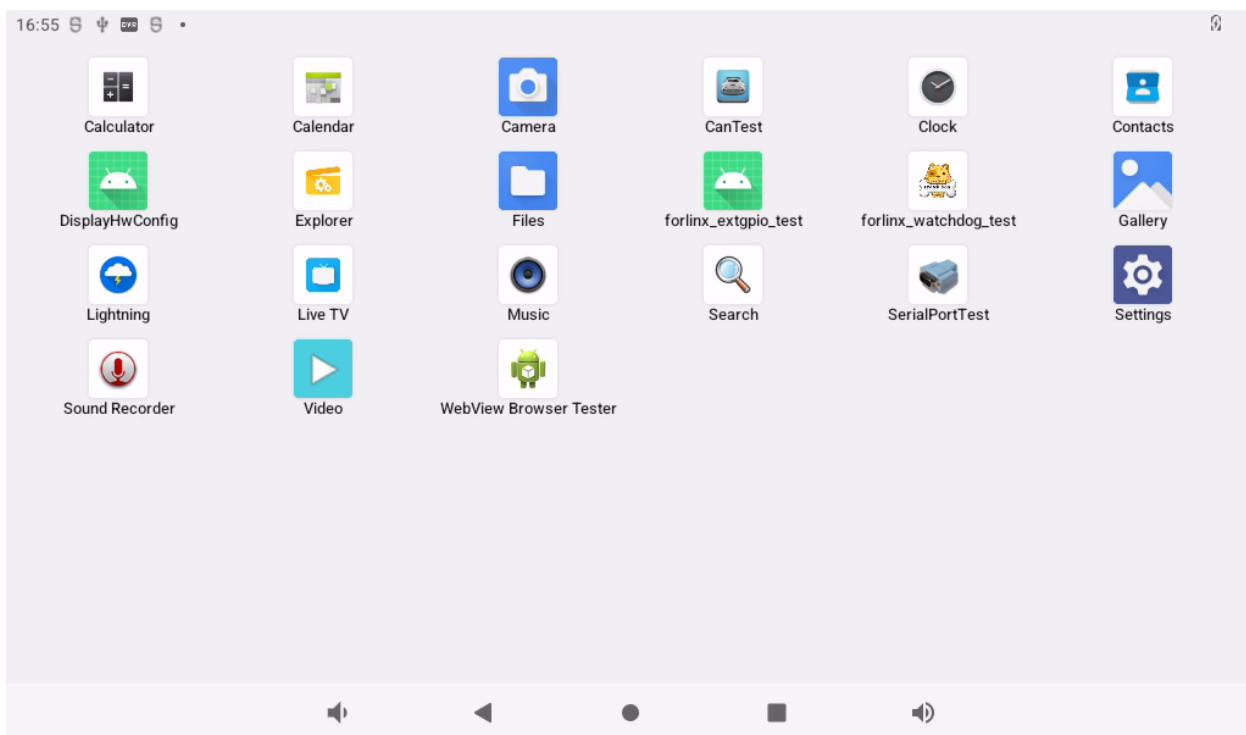
If a timeout (timeout of 10S) is not performed to feed the dog, the system reboots. Click “stop” to stop the watchdog test, and the system will not start:



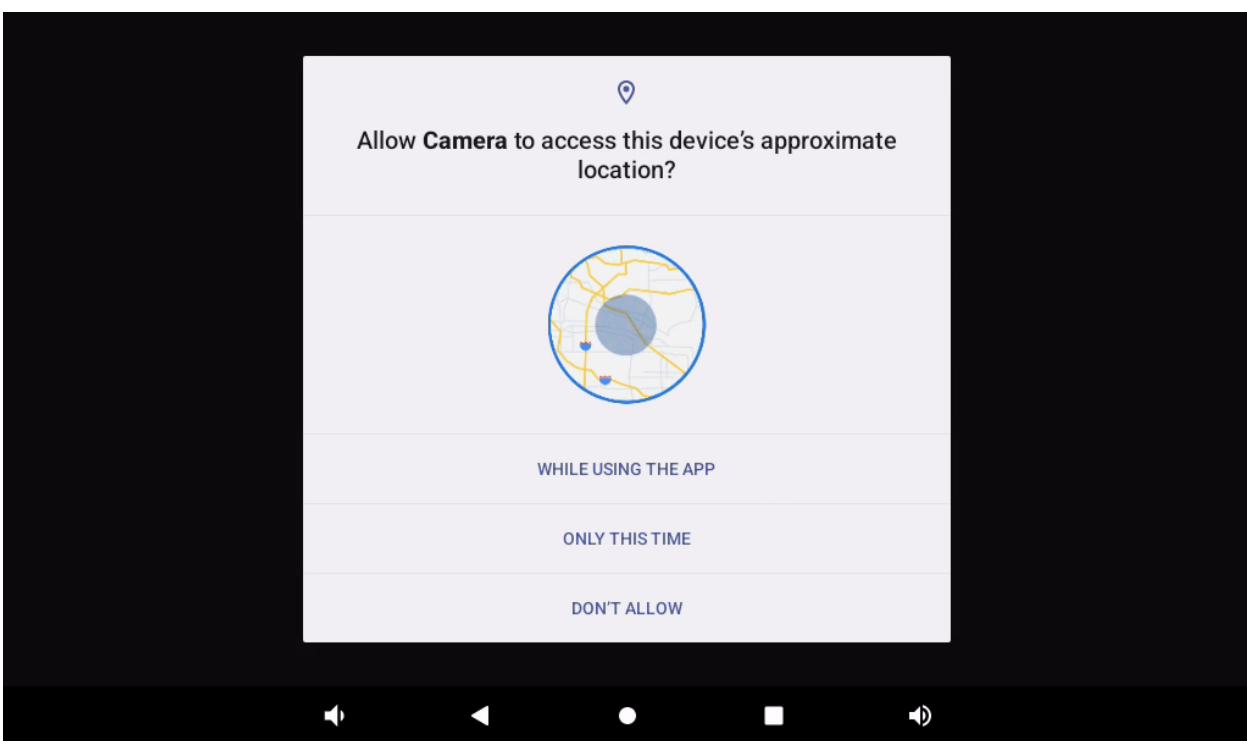
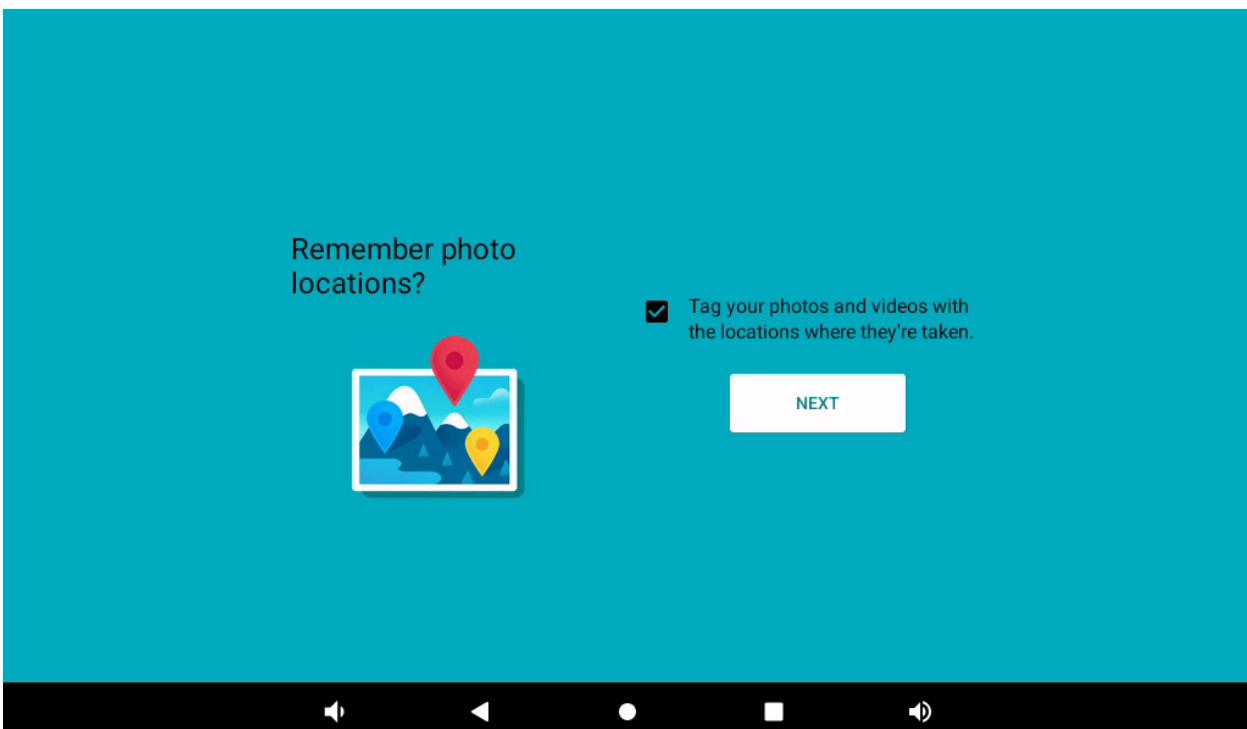
3.21 Camera Test

3.21.1 UVC Camera Test

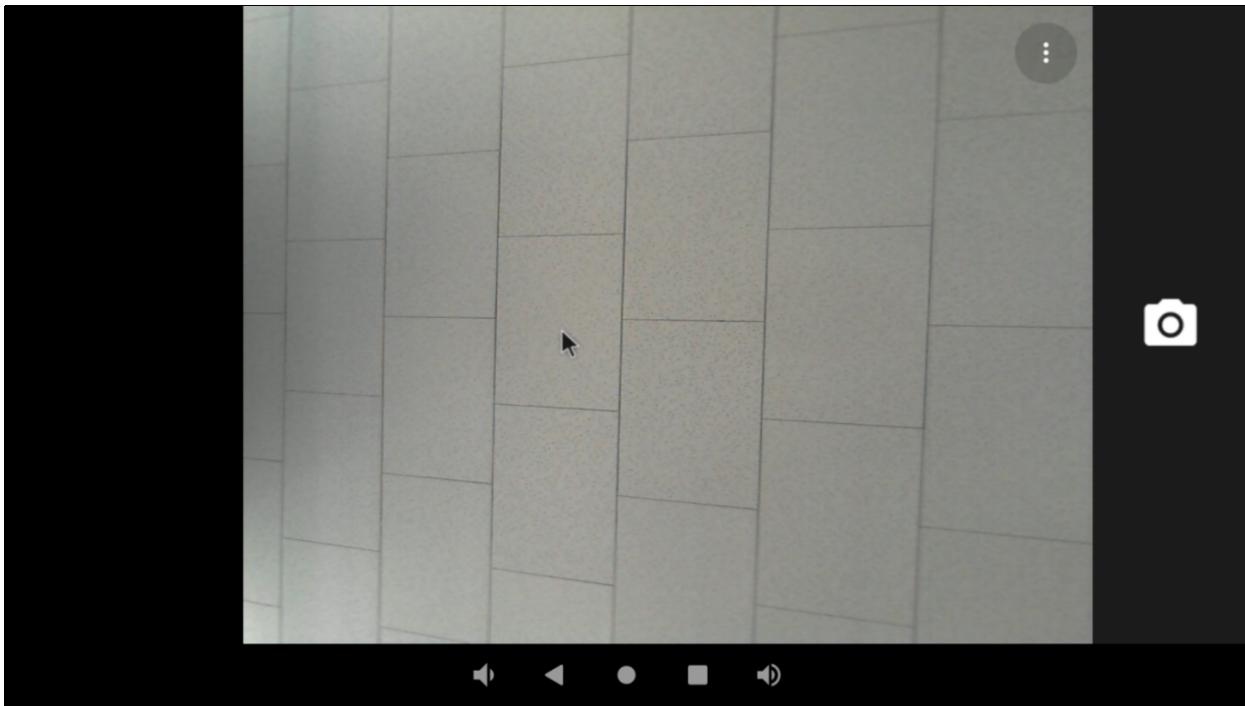
Click on the camera in the application interface:



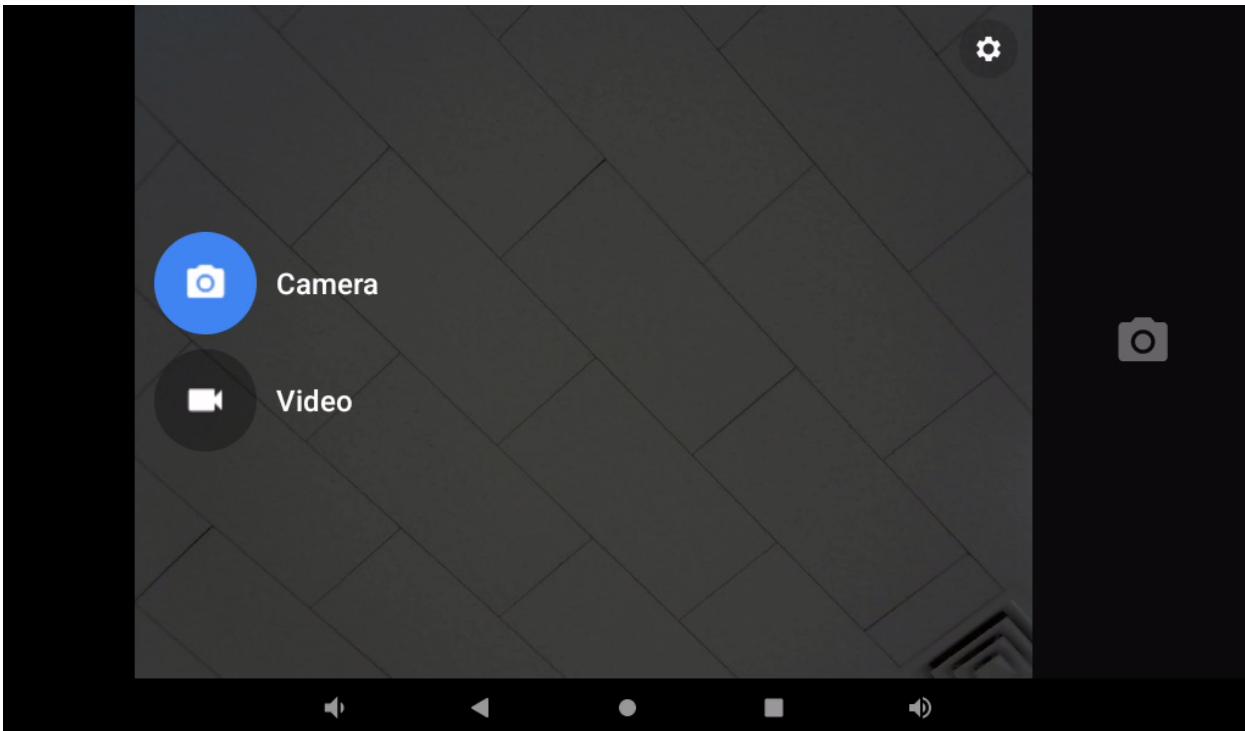
Configure permissions:



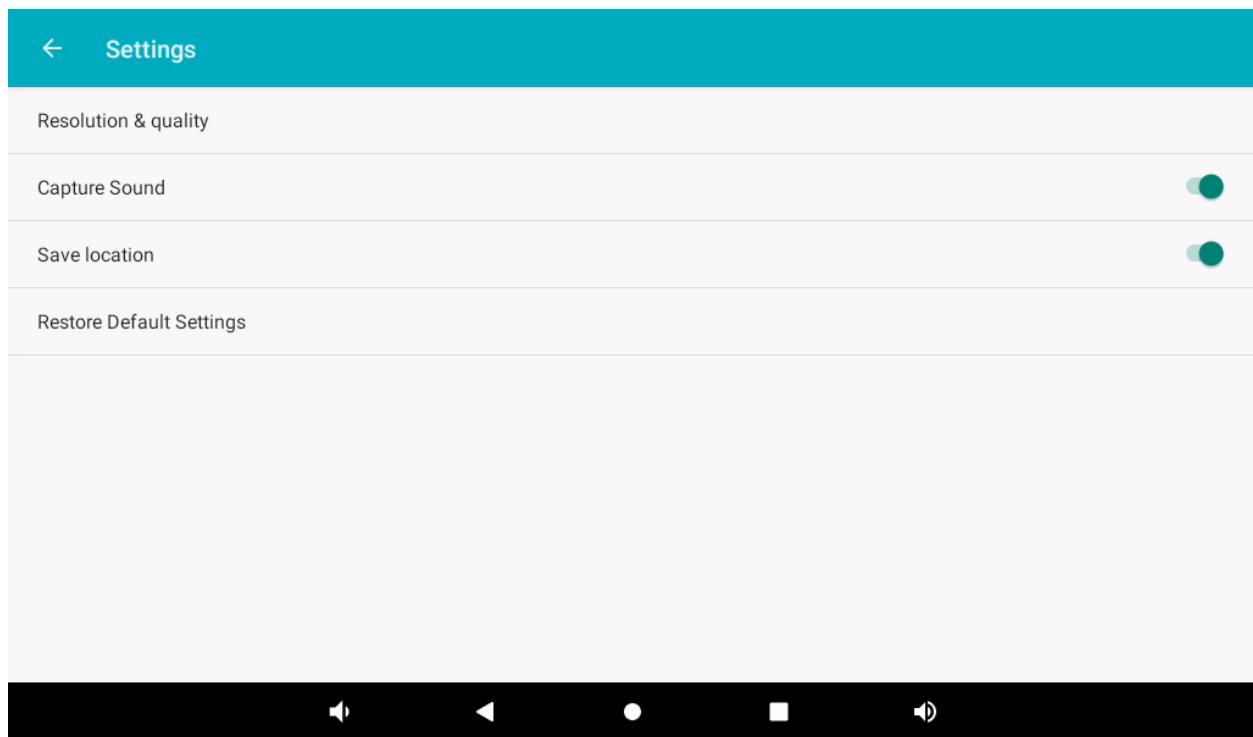
Enter the preview interface and click the photo button on the right to take a photo:



Swipe the screen to the right to open the options for switching between photo and video mode, as well as accessing settings.

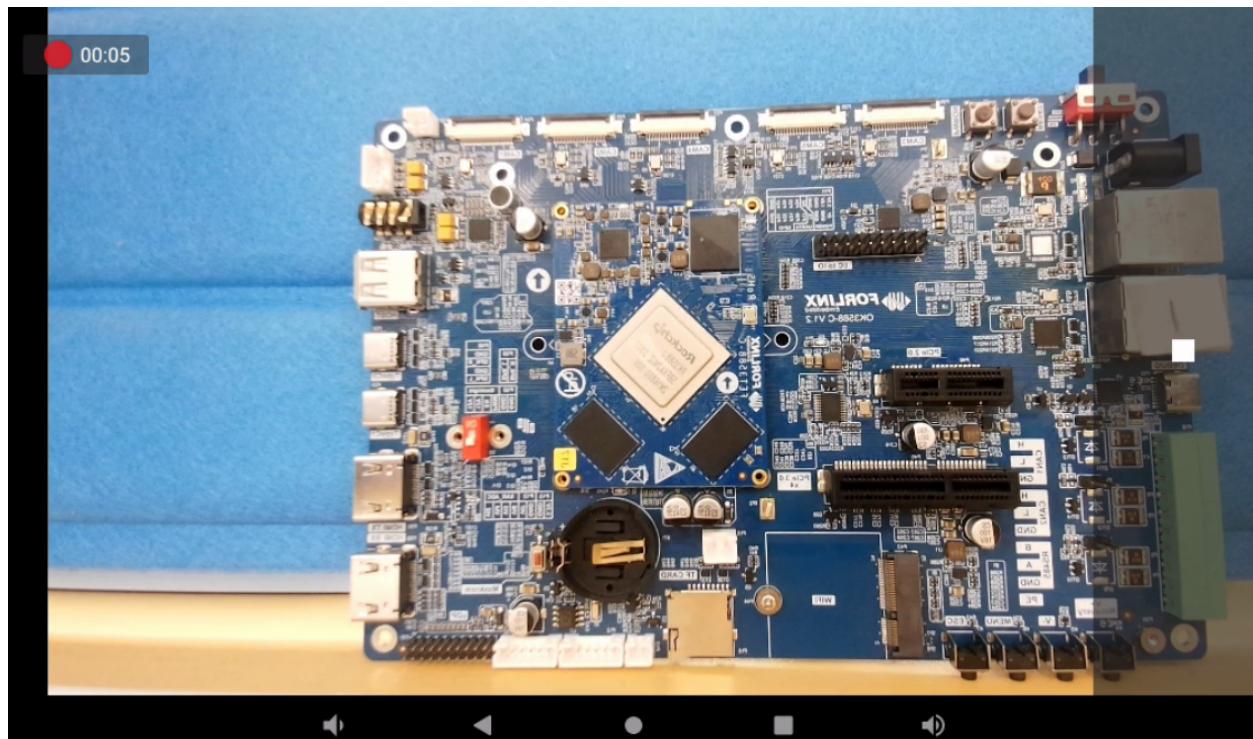


Tap on the settings button in the top right corner to adjust settings such as resolution and image quality.



Click the video button to enter the video preview interface:

Click the video button to record the video:

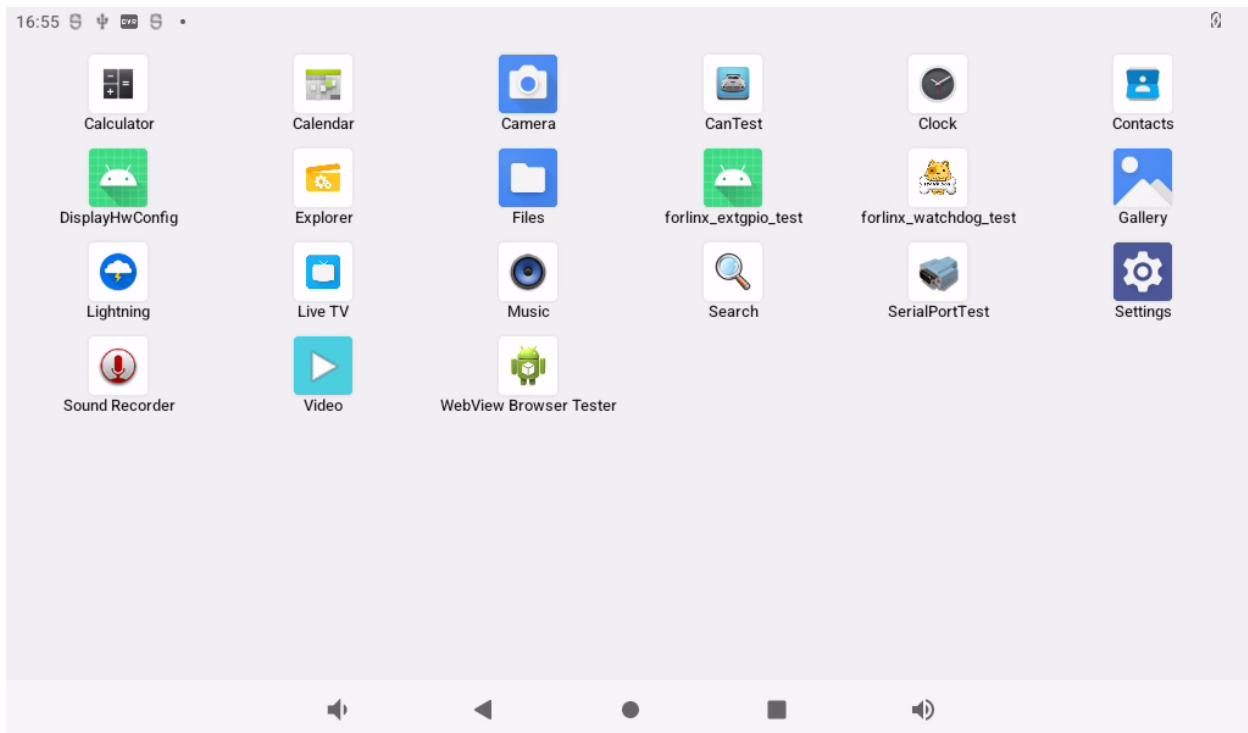


3.21.2 OV13850 Camera Test

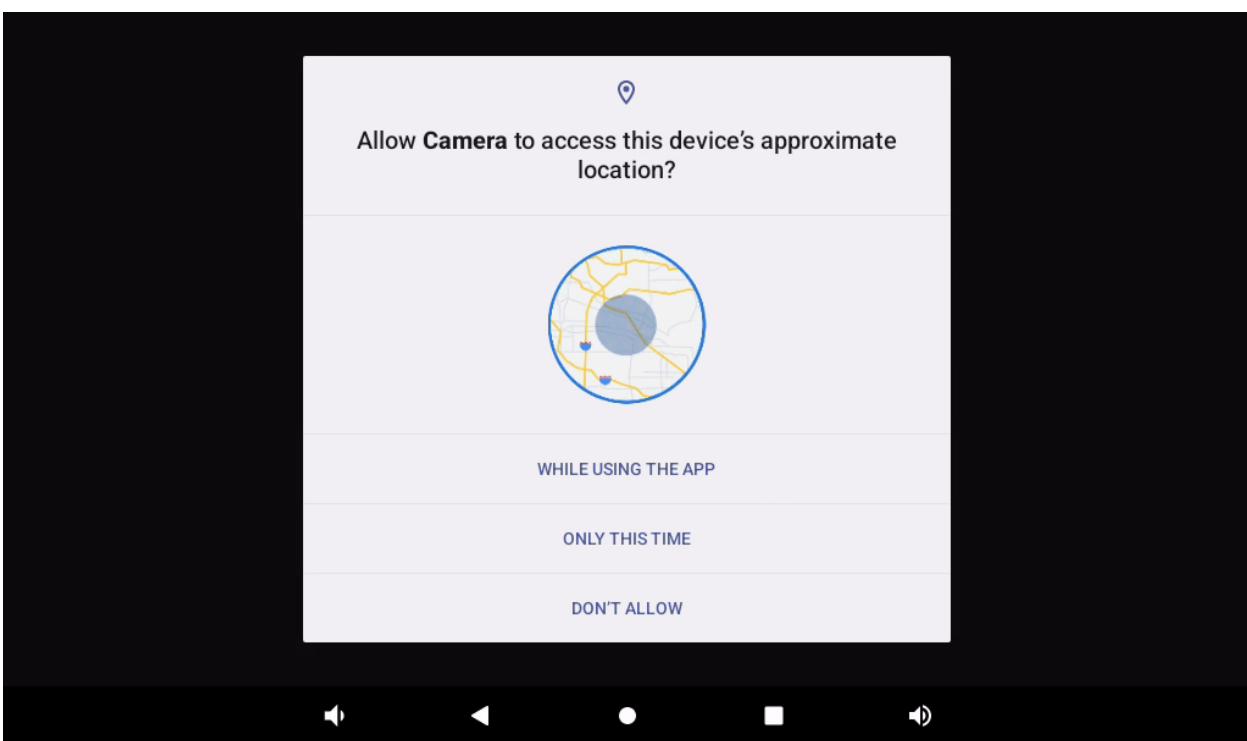
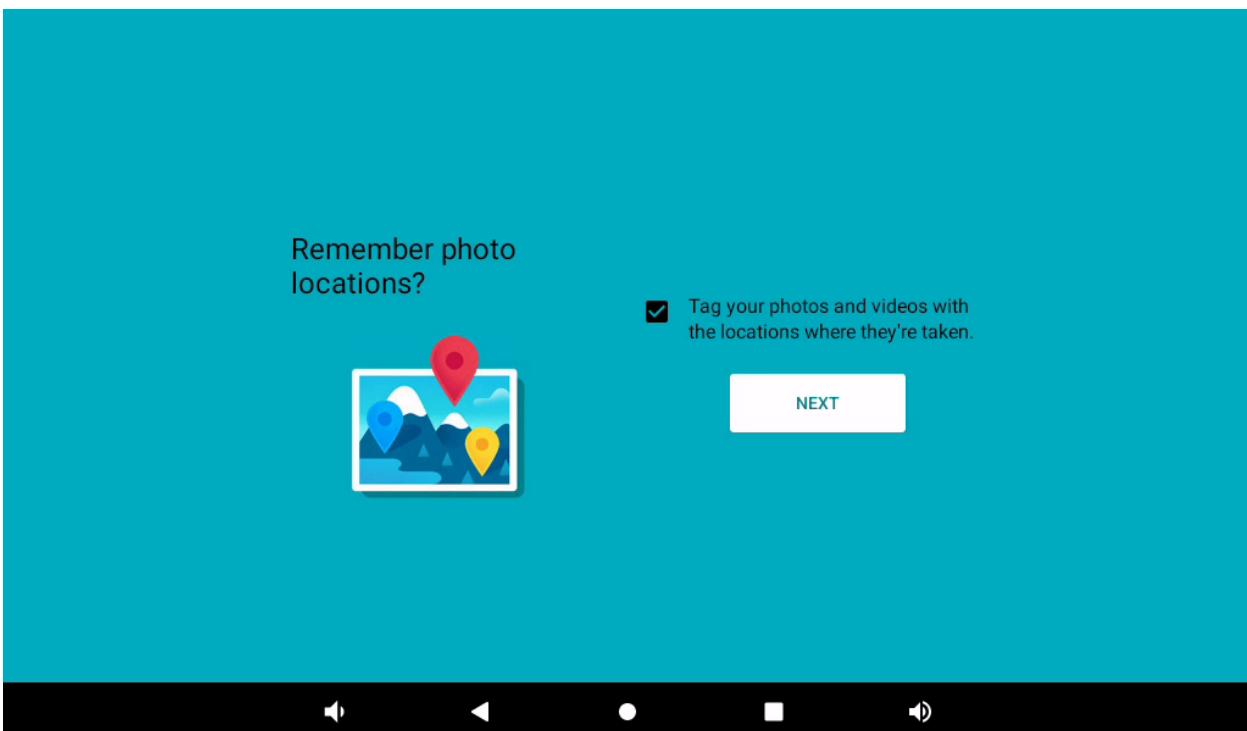
OK3588 supports 5 x mipi Camera, in which CAM1 and CAM2 are used for OV13850 and CAM3, CAM4 and CAM5 are used for MIPI OV5645.

Please power off first, plug in two OV13850 Cameras, and power on to start.

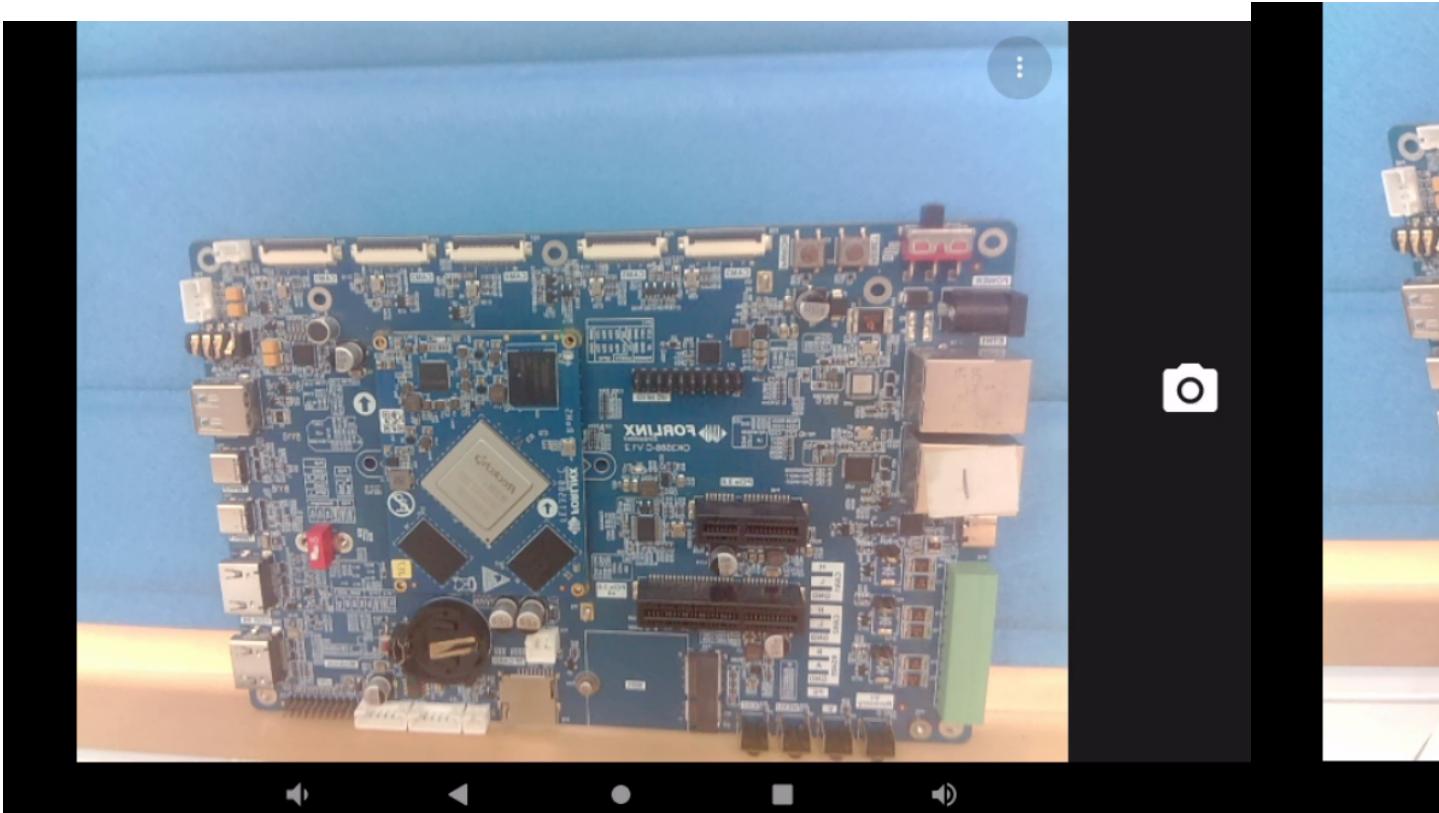
Click on the camera in the application interface:



Configure permissions:

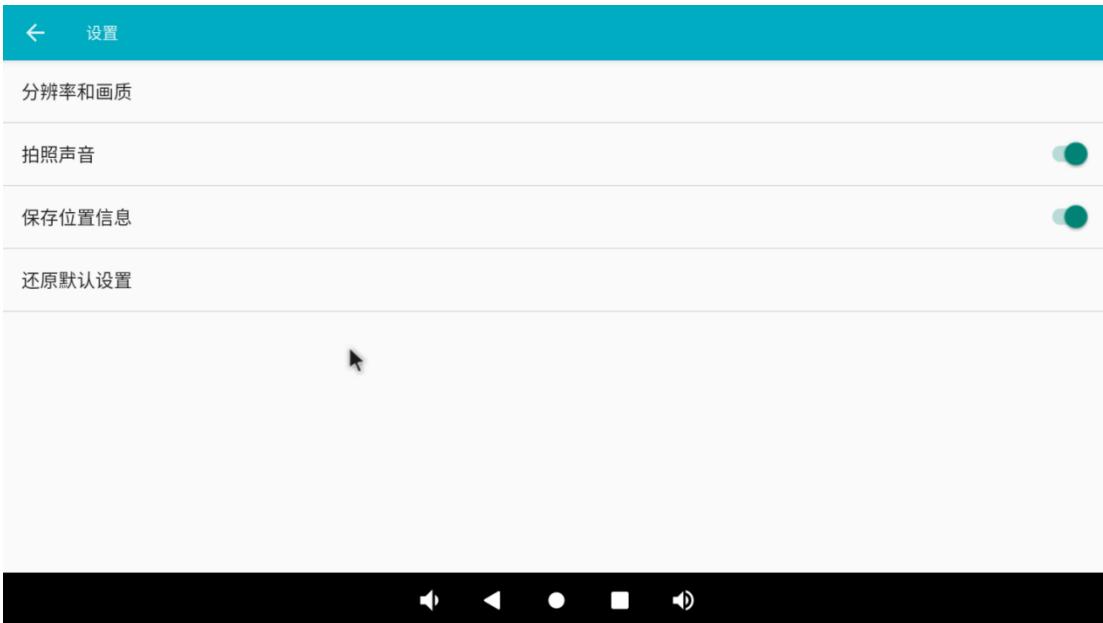


Enter the preview interface and click the photo button on the right to take a photo:



Swipe the screen to the right to open the options for switching between photo and video mode, as well as accessing settings.

Tap on the settings button in the top right corner to adjust settings such as resolution and image quality.



Click the video button to enter the video preview interface:



Click the video button to record the video:

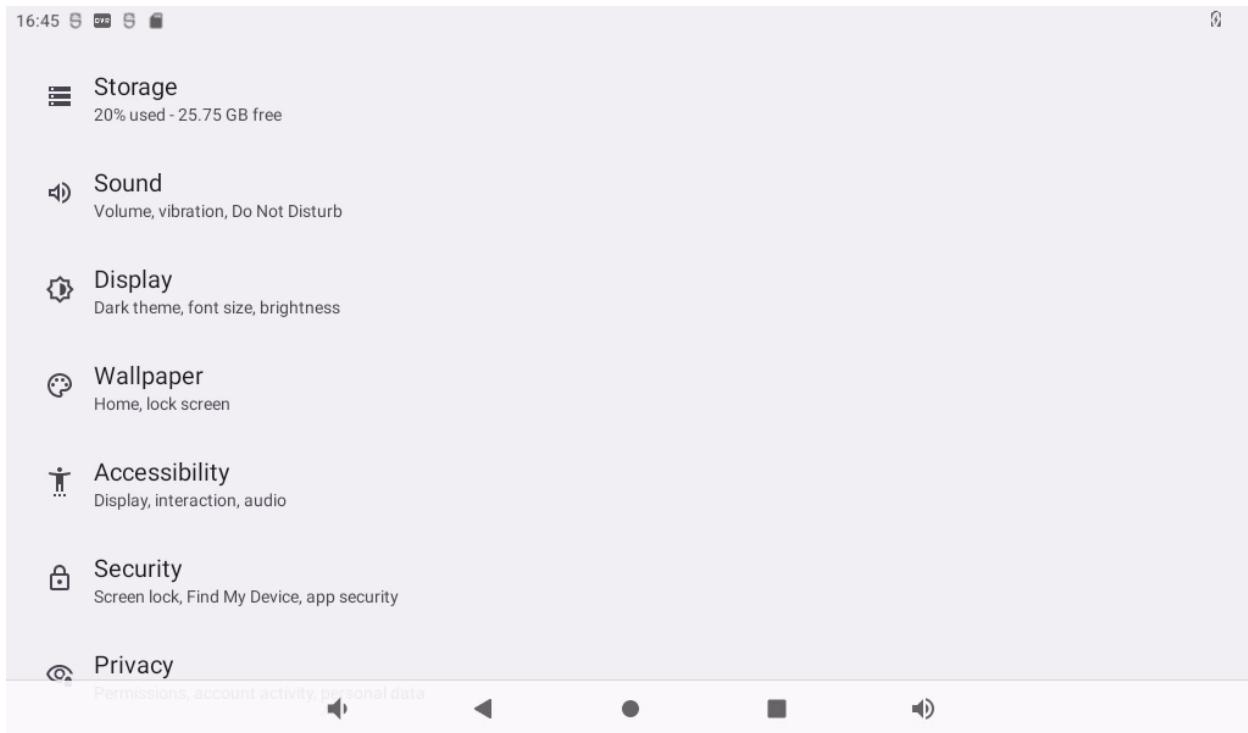


3.22 HDMI Resolution Setting Test

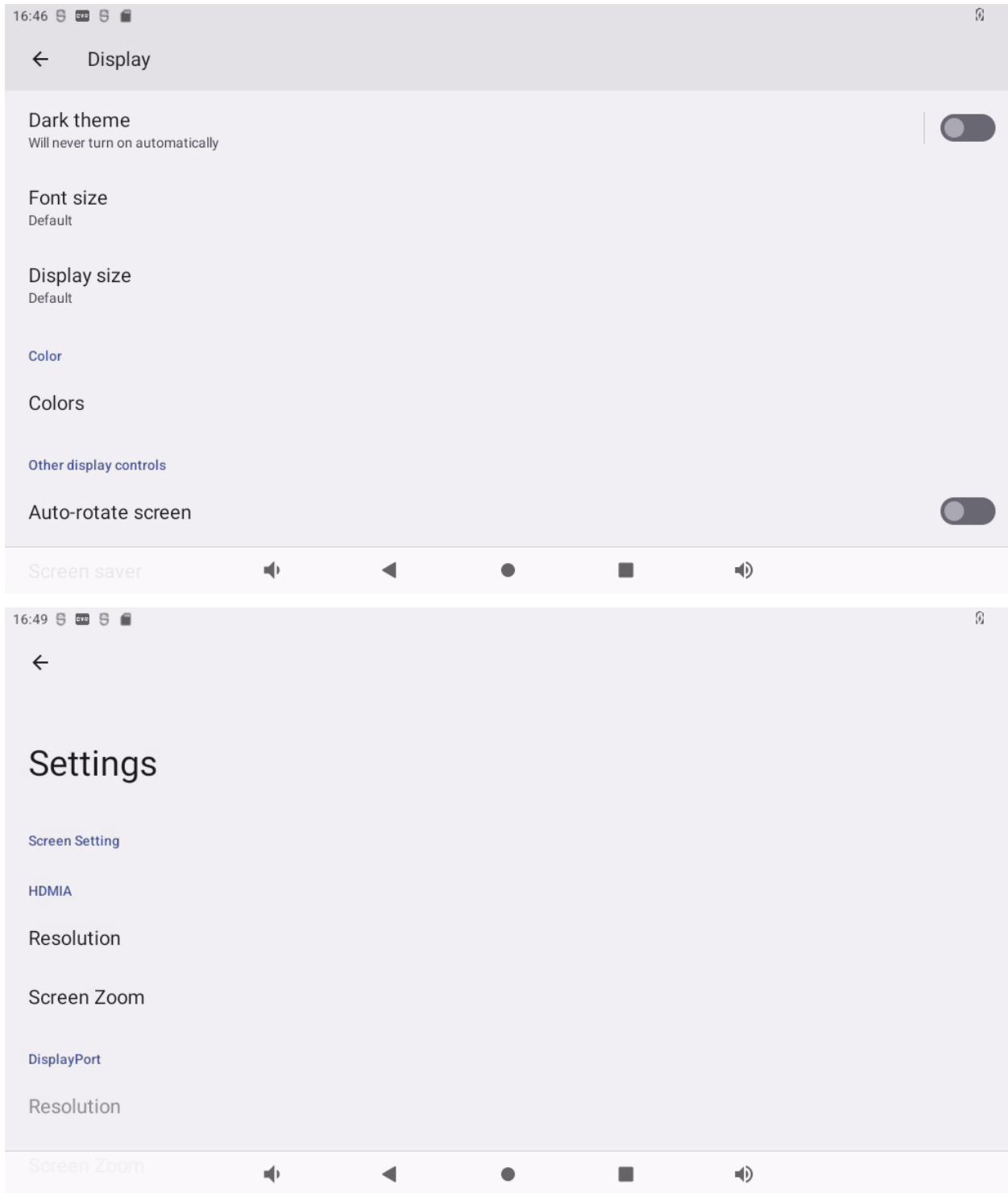
OK3588 platform supports dynamic setting of HDMI resolution.



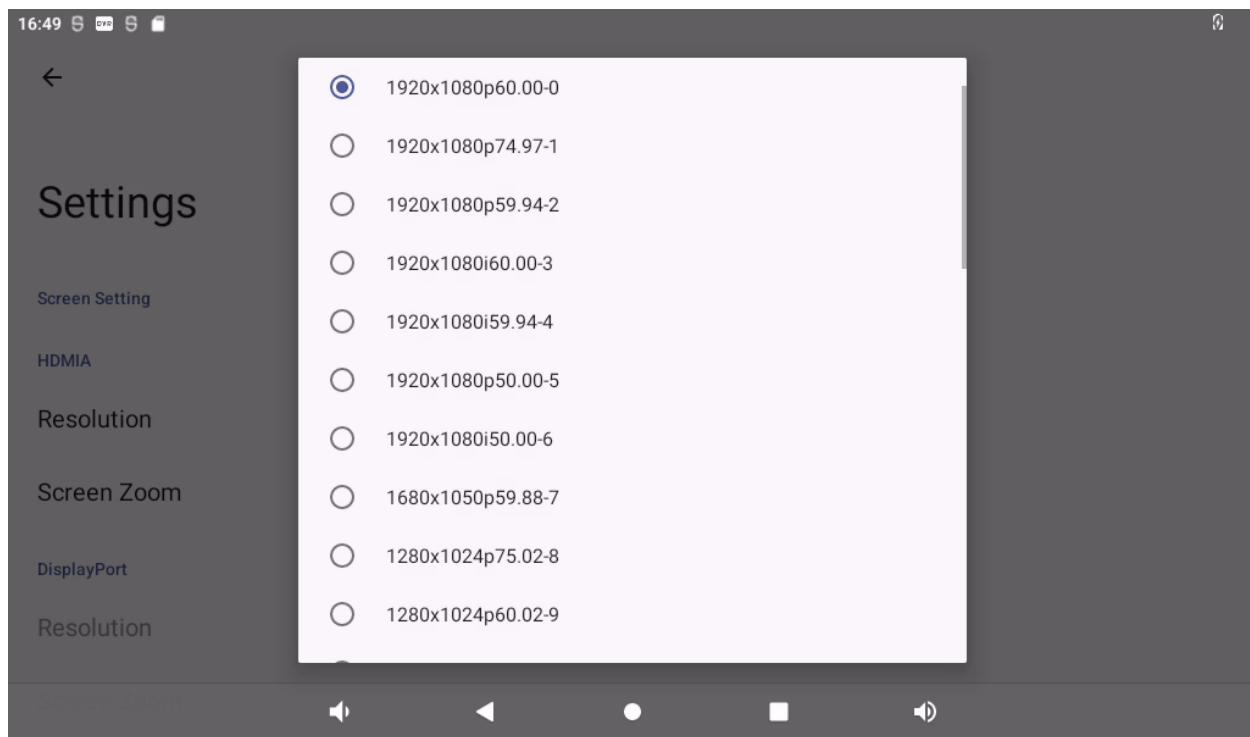
Click “ ” , on the application interface to enter the setting interface:



Click “Display” , select “Advanced” , and click “HDMI” to configure HDMI:



You can dynamically select the desired resolution based on the resolution supported by the current HDMI screen:



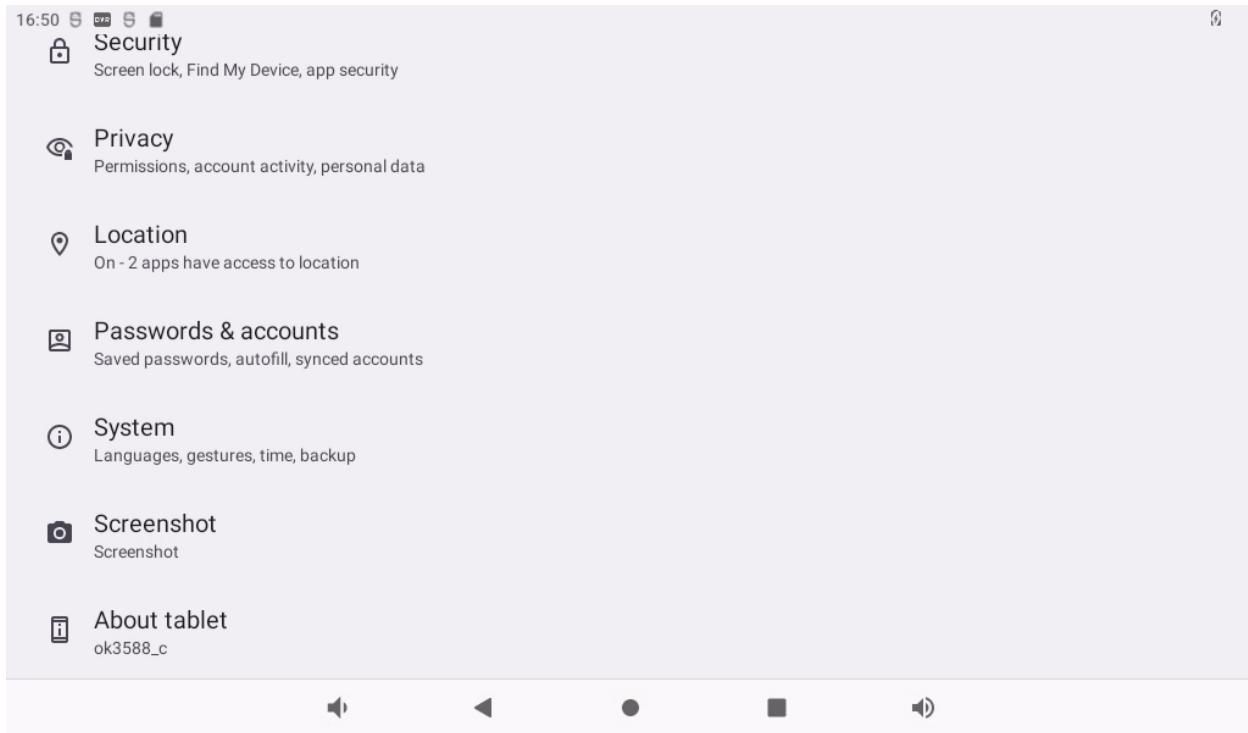
After restarting, the device will take effect.

3.23 Factory Reset

The OK3588 platform supports restoring factory settings.



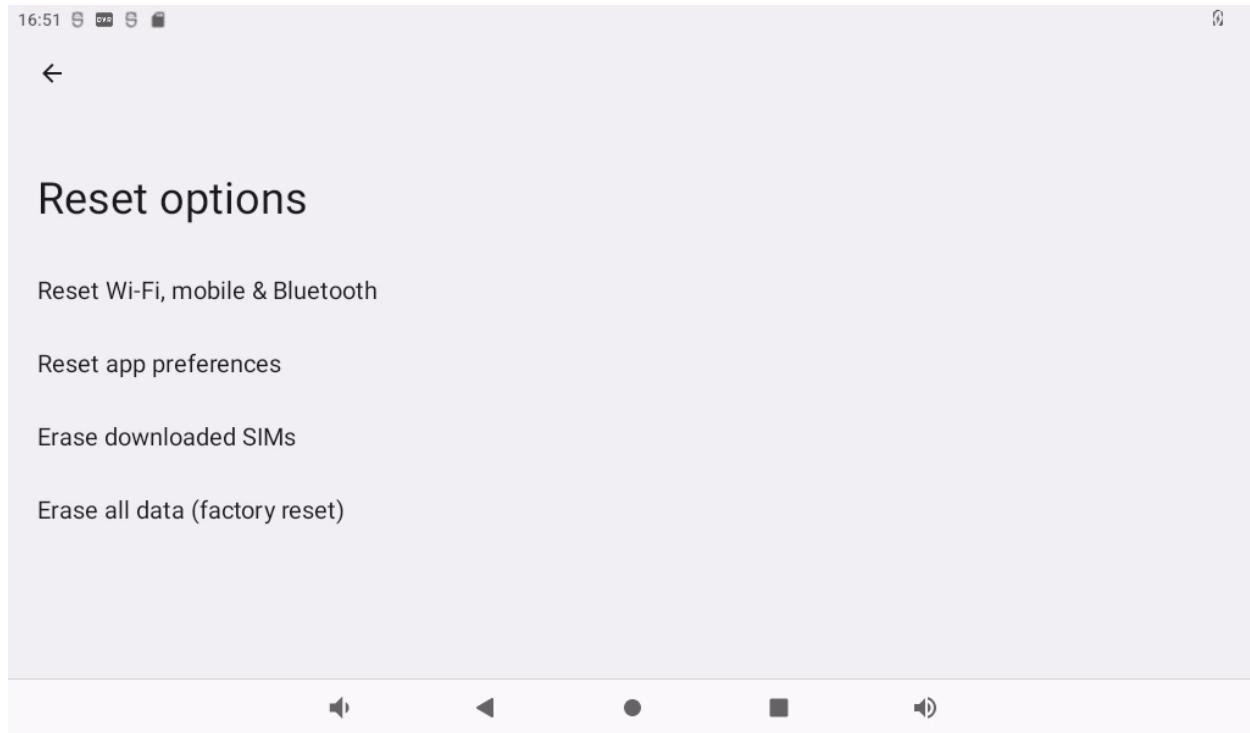
Click “ ” , on the application interface to enter the setting interface:



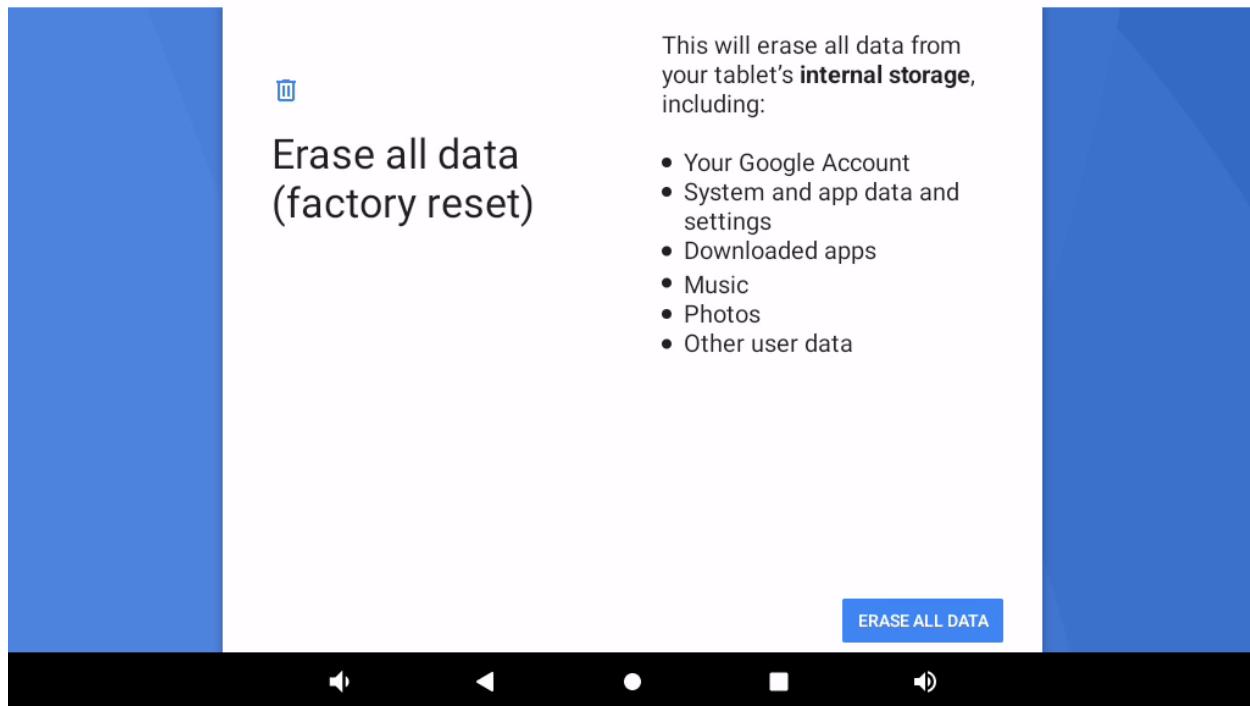
Click “System” :



Click “Reset option” and select “Clear all data (restore factory settings)” :



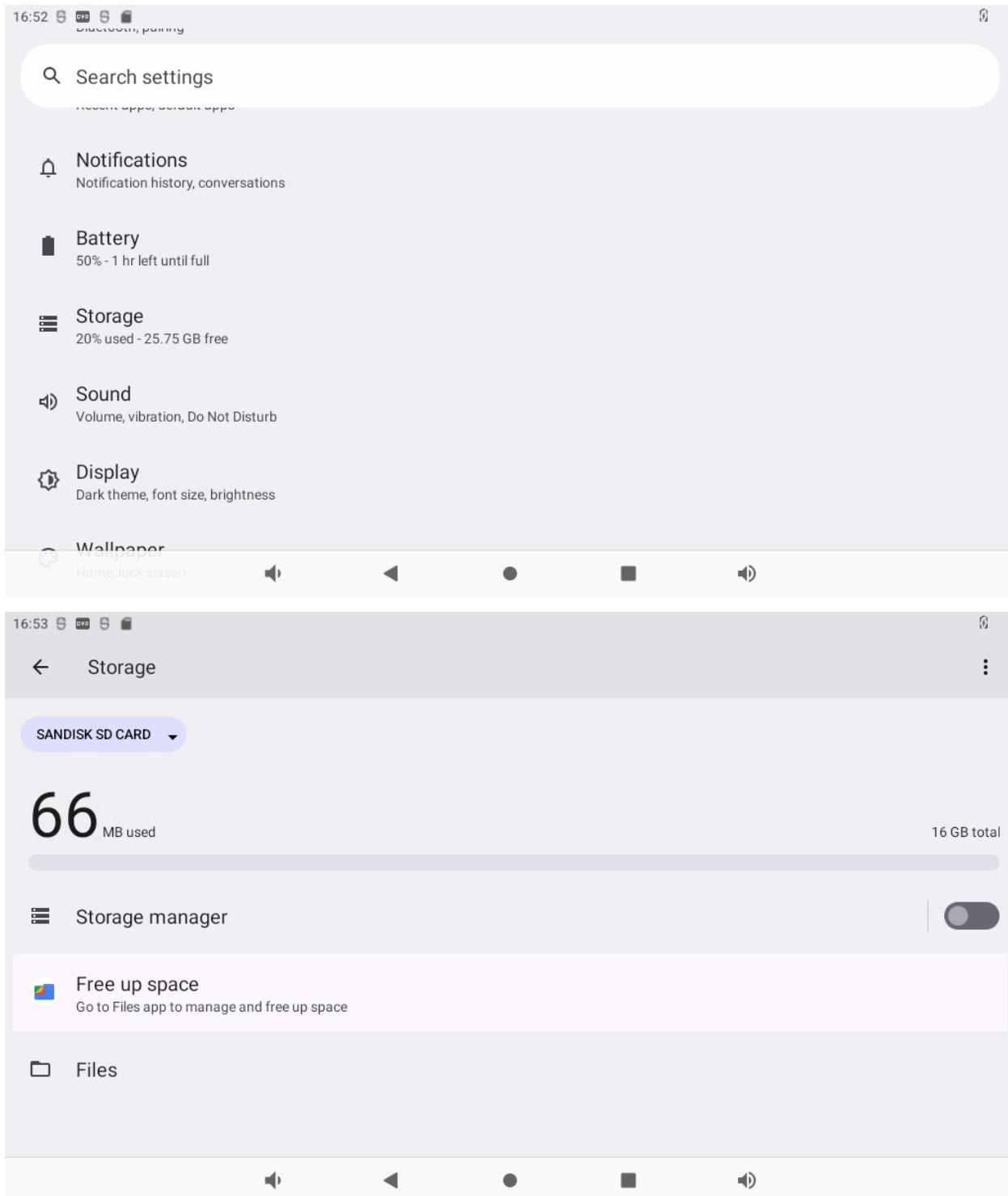
Then click “Clear All Data” .

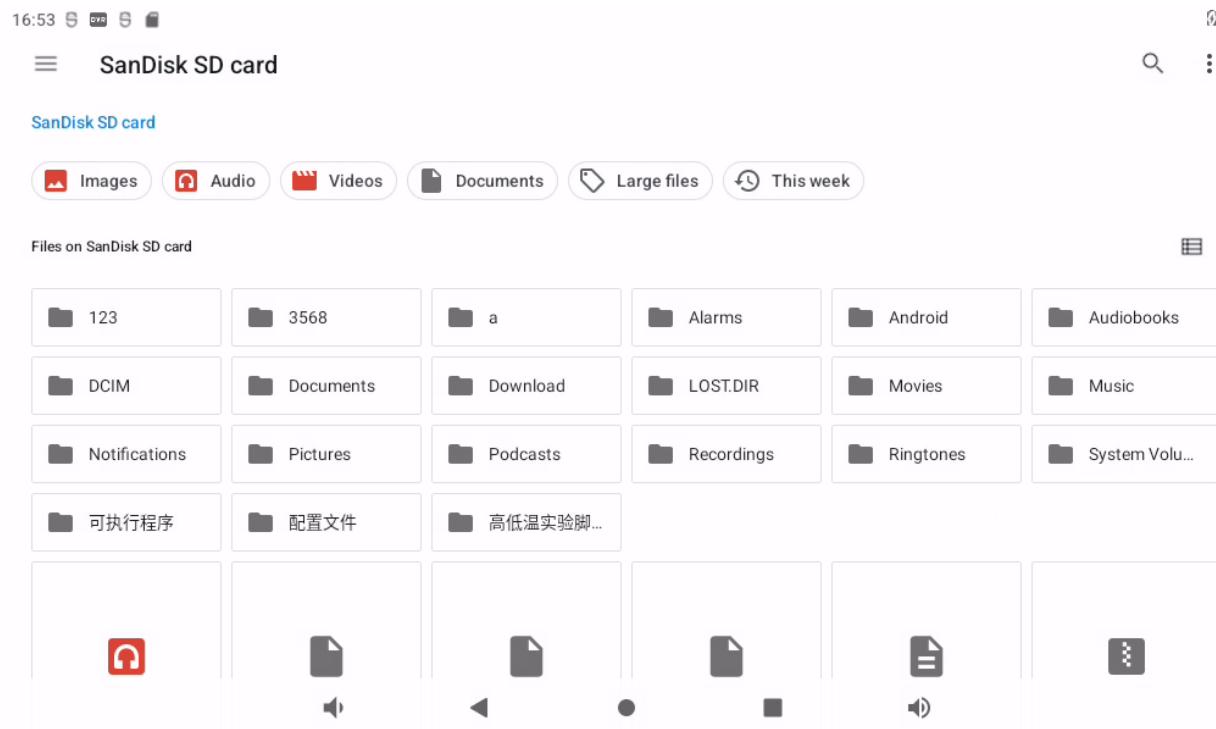


Wait for OK3588 to restore the default factory settings. Please do not power off during the process of restoring the factory settings.

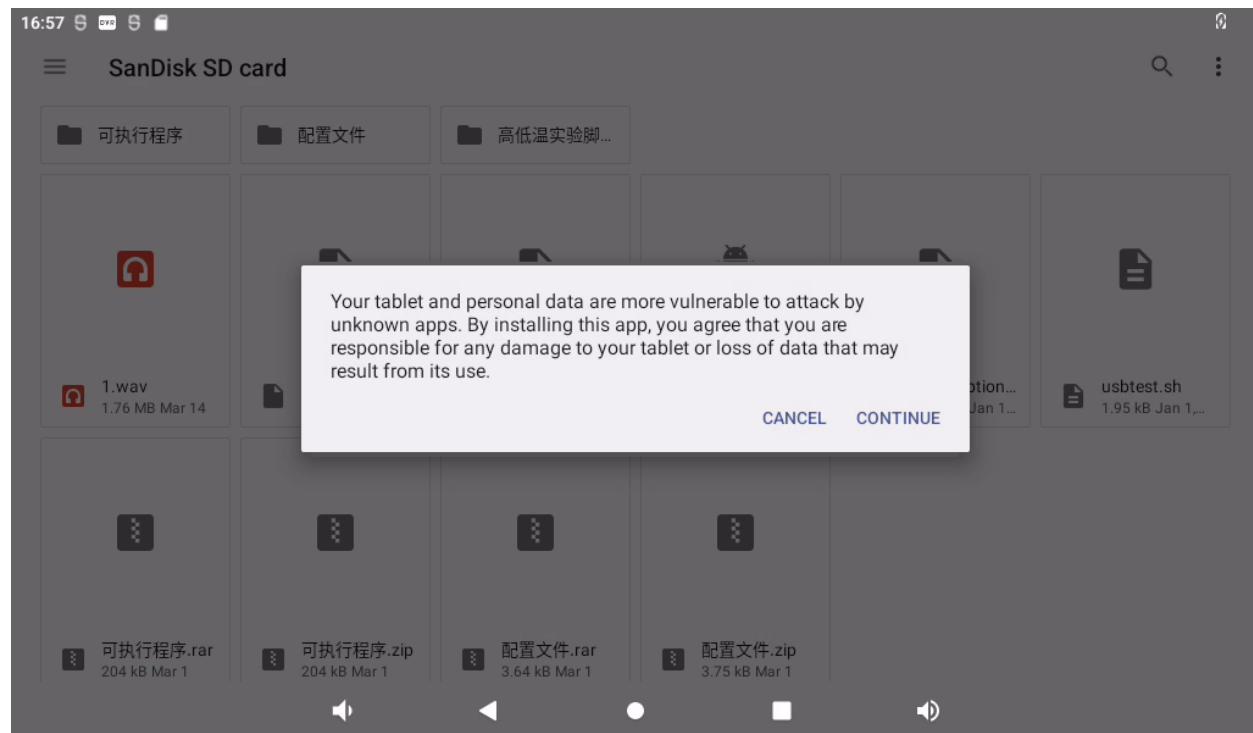
3.24 APK Installation with TF Card

After loading the TF card according to the previous steps, you can see an APK file after entering the TF card directory.

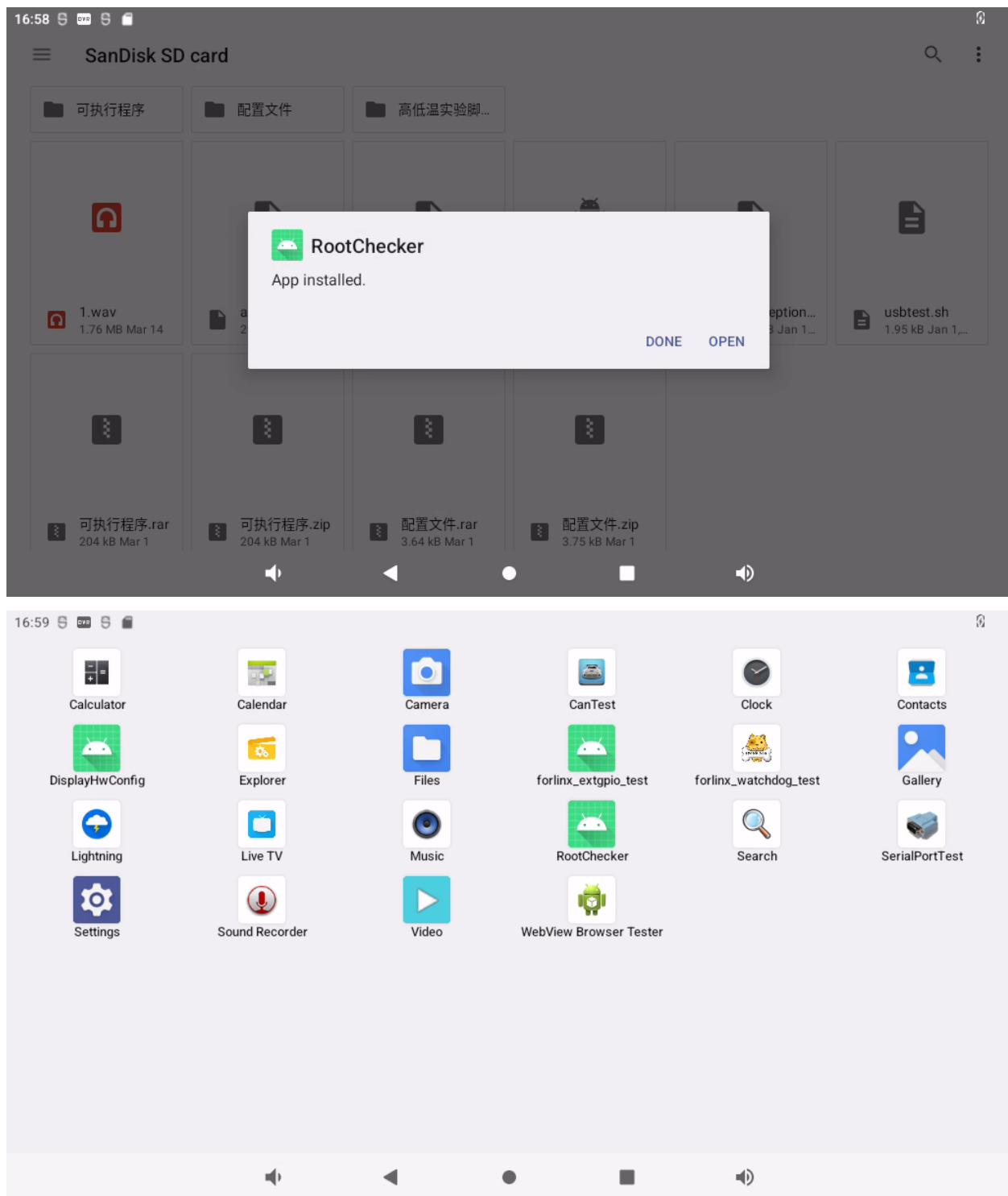




Double-click the APK file to install and configure permissions:



Click "Install" to complete the installation:

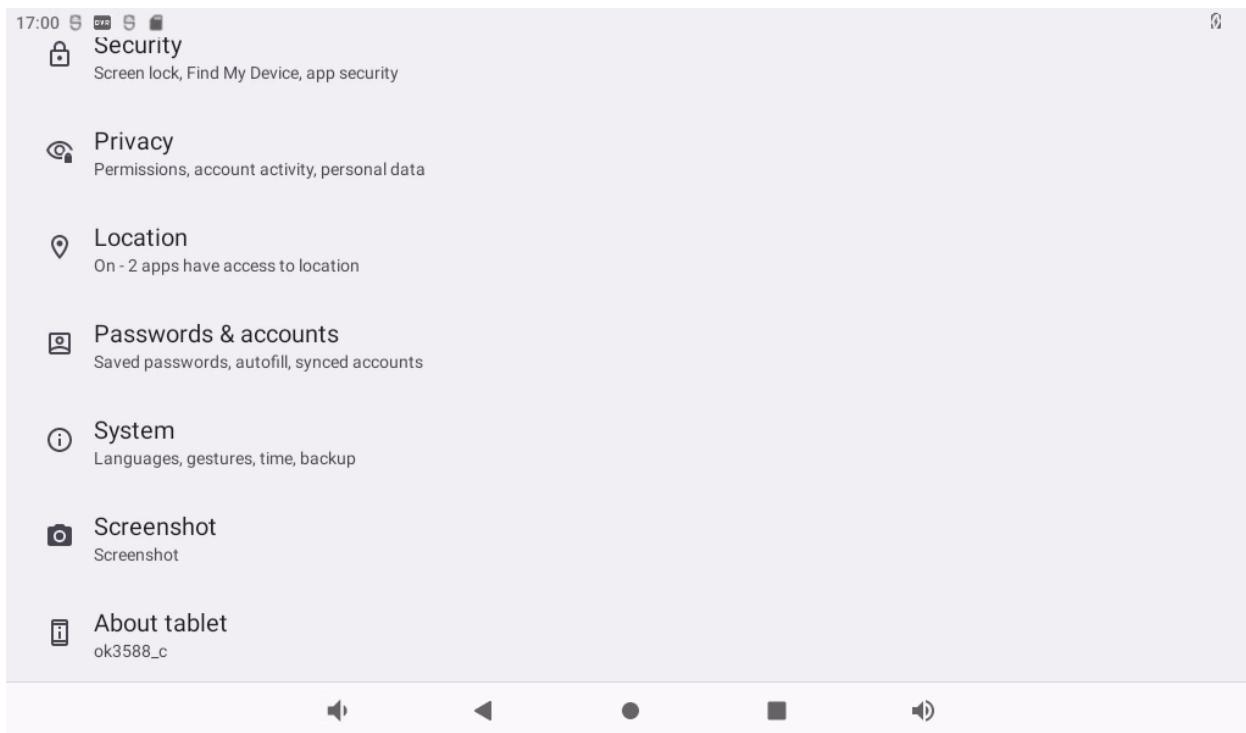


3.25 WiFi ADB Test

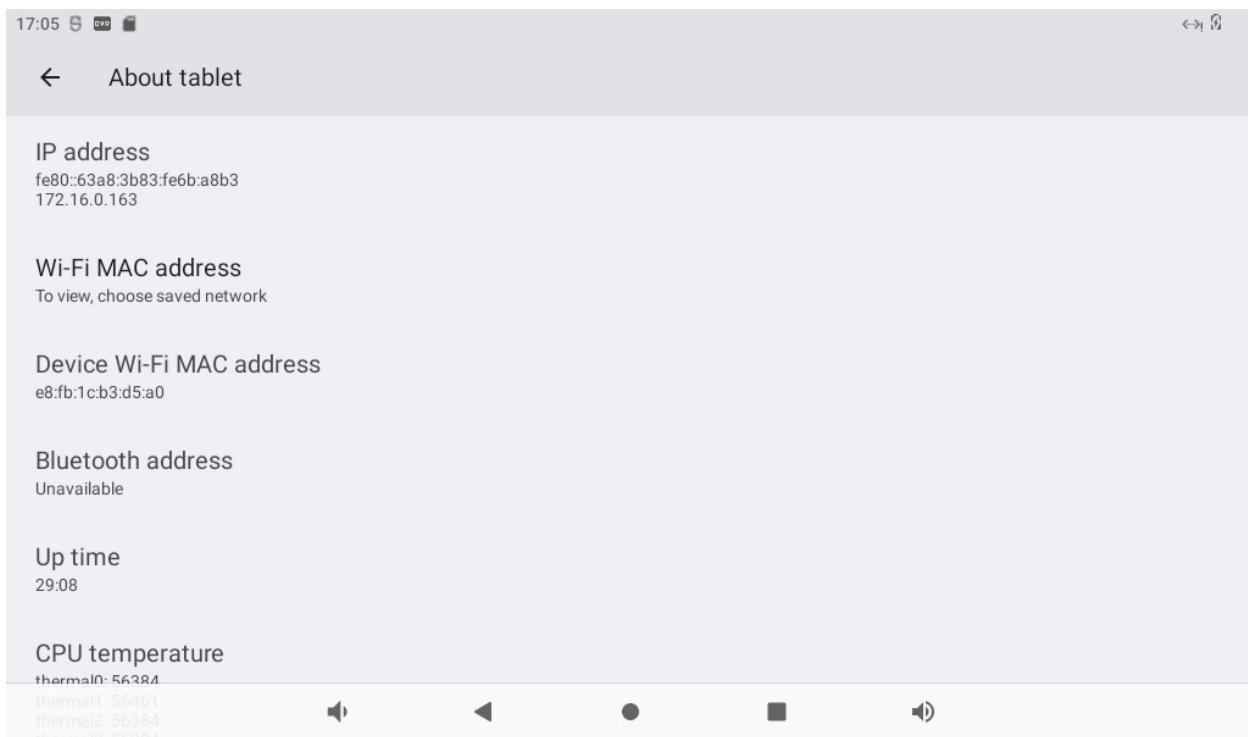
Note: Only one of USB ADB and WIFI ADB can be used at the same time.



Follow the previous section to connect to WIFI, click “ ” after successfully connecting to WIFI:



Click on “About Tablet” :



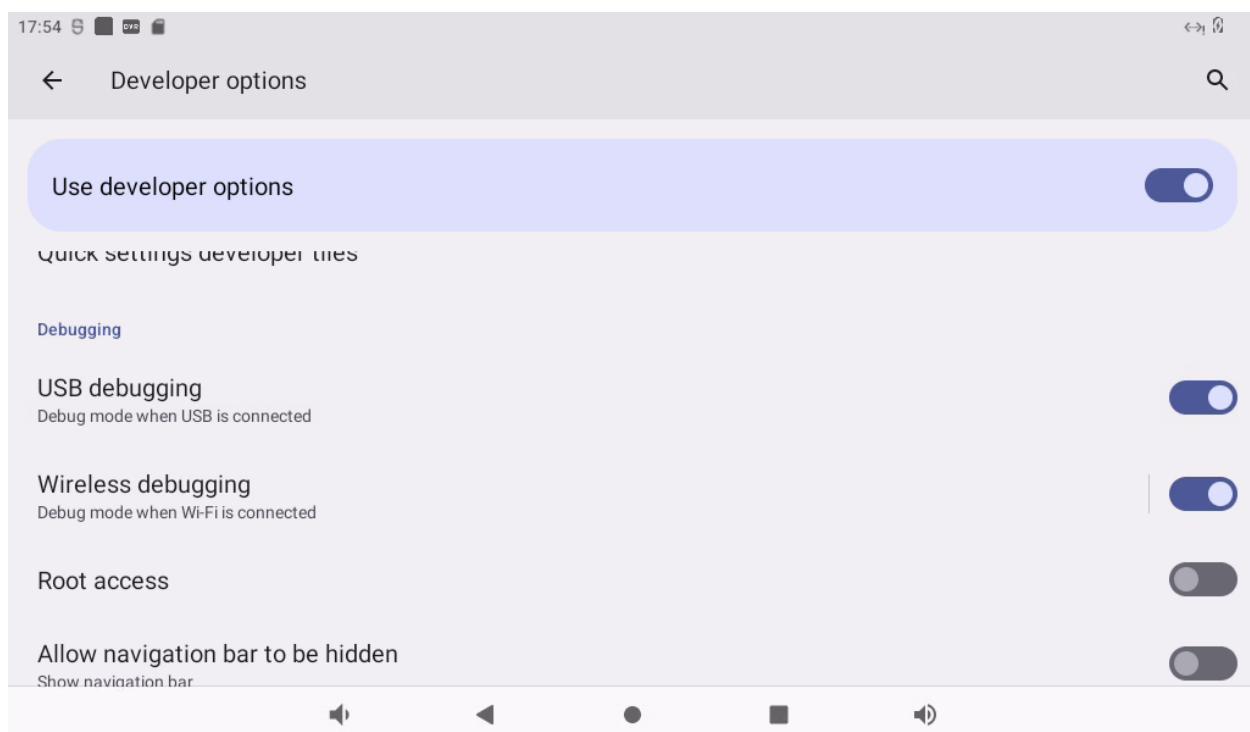
Continuously click the “version number” prompt to enter the development mode:

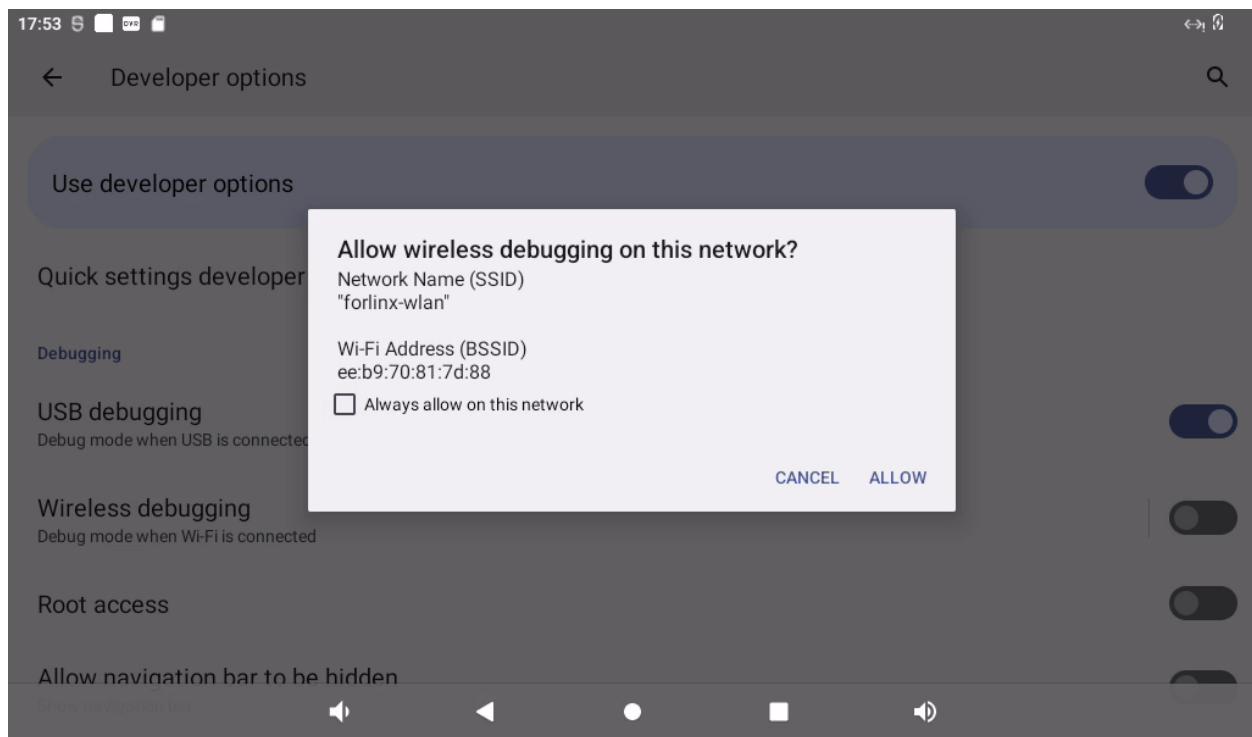


Return to the previous layer and select “System” in the setting interface:

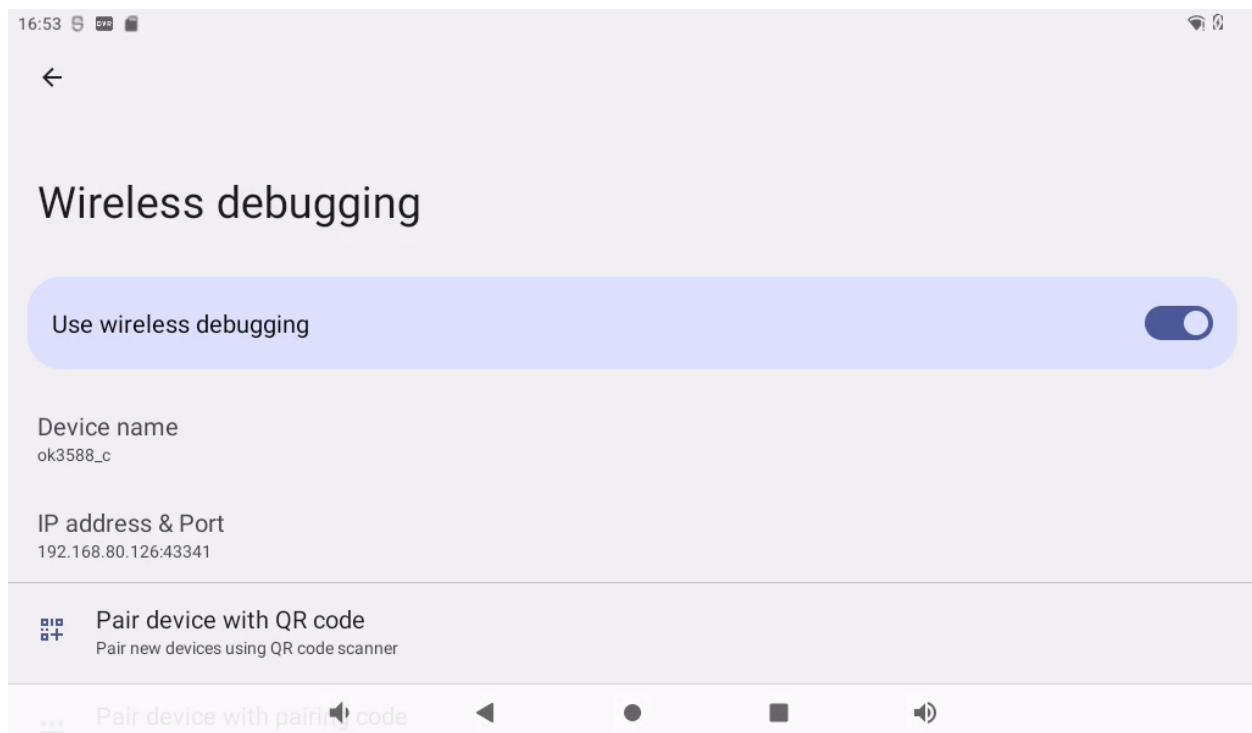


Select the developer option, turn off “USB Debugging” and turn on “Wireless Debugging” :

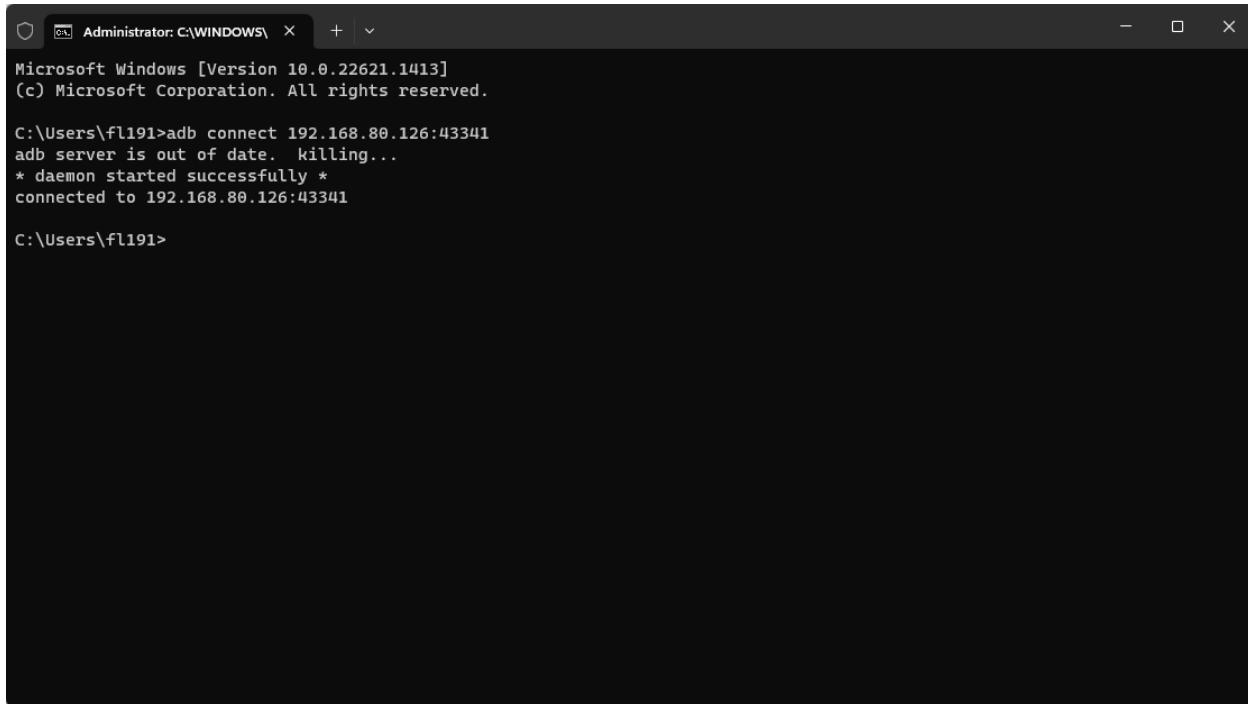




Record the IP address and port of the current WIFI



Open the window command control window and enter “adb connect 192.168.1.61:43985” to connect to wifi:

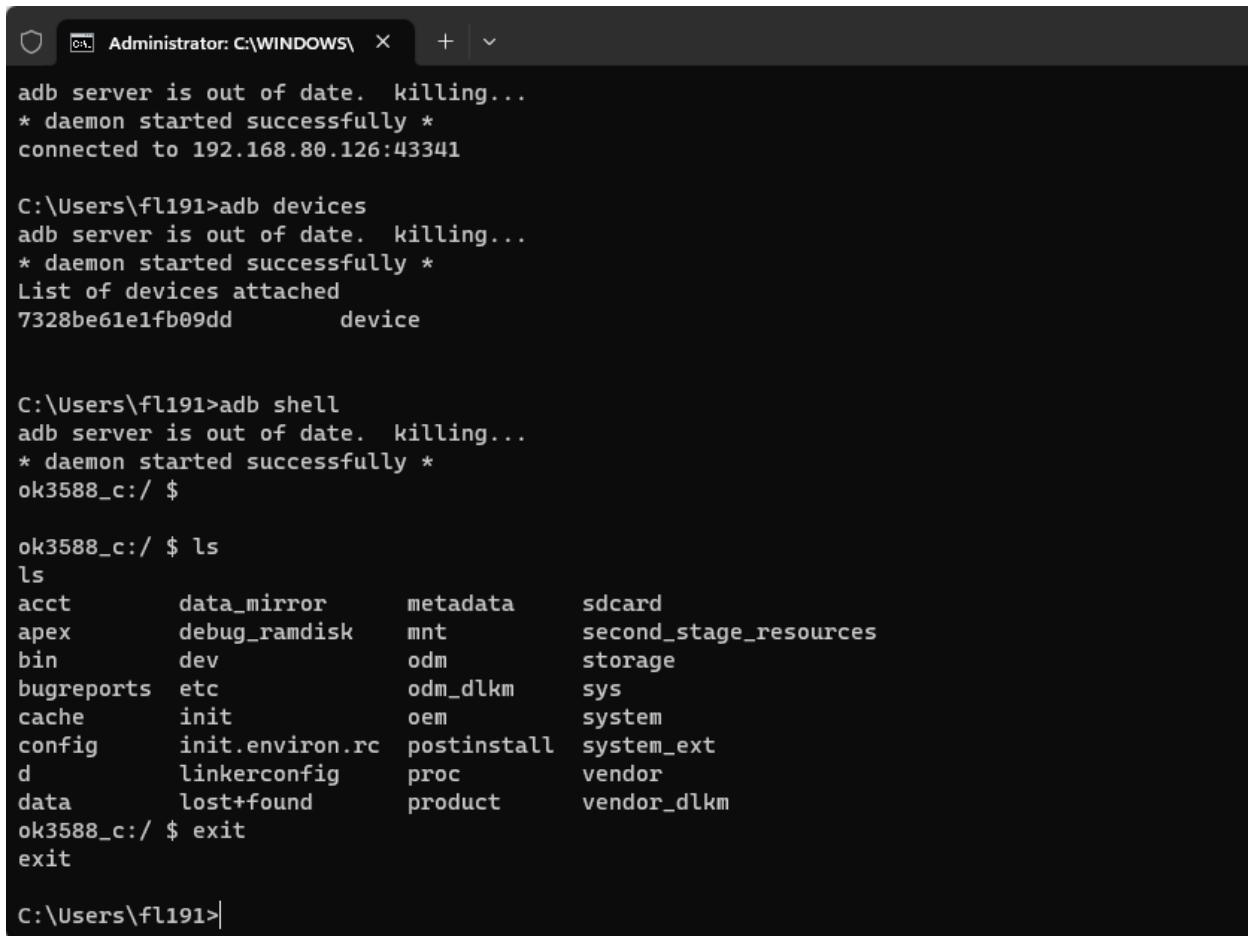


```
Administrator: C:\WINDOWS\ 
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\fl191>adb connect 192.168.80.126:43341
adb server is out of date. killing...
* daemon started successfully *
connected to 192.168.80.126:43341

C:\Users\fl191>
```

Type “adb devices” to see the connected devices. Enter “adb shell” to access the device terminal.



```
Administrator: C:\WINDOWS\ 
adb server is out of date. killing...
* daemon started successfully *
connected to 192.168.80.126:43341

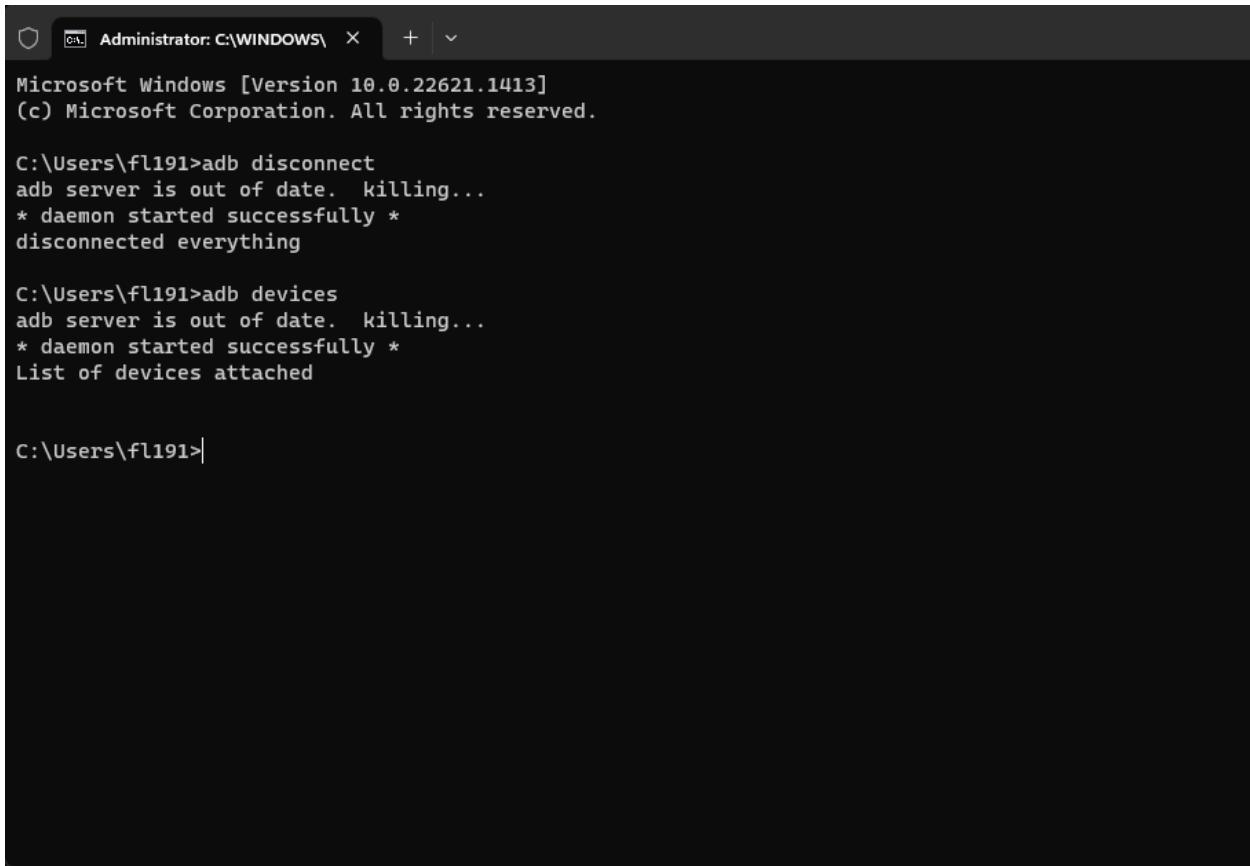
C:\Users\fl191>adb devices
adb server is out of date. killing...
* daemon started successfully *
List of devices attached
7328be61e1fb09dd      device

C:\Users\fl191>adb shell
adb server is out of date. killing...
* daemon started successfully *
ok3588_c:/ $

ok3588_c:/ $ ls
ls
acct      data_mirror    metadata    sdcard
apex      debug_ramdisk  mnt        second_stage_resources
bin       dev             odm        storage
bugreports etc            odm_dlkm   sys
cache     init            oem        system
config    init.environ.rc postinstall system_ext
d         linkerconfig    proc       vendor
data     lost+found      product    vendor_dlkm
ok3588_c:/ $ exit
exit

C:\Users\fl191>
```

Type “adb disconnect” to disconnect, and then type “adb devices” to see the devices.



A screenshot of a Windows Command Prompt window titled "Administrator: C:\WINDOWS\". The window shows the following text:

```
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

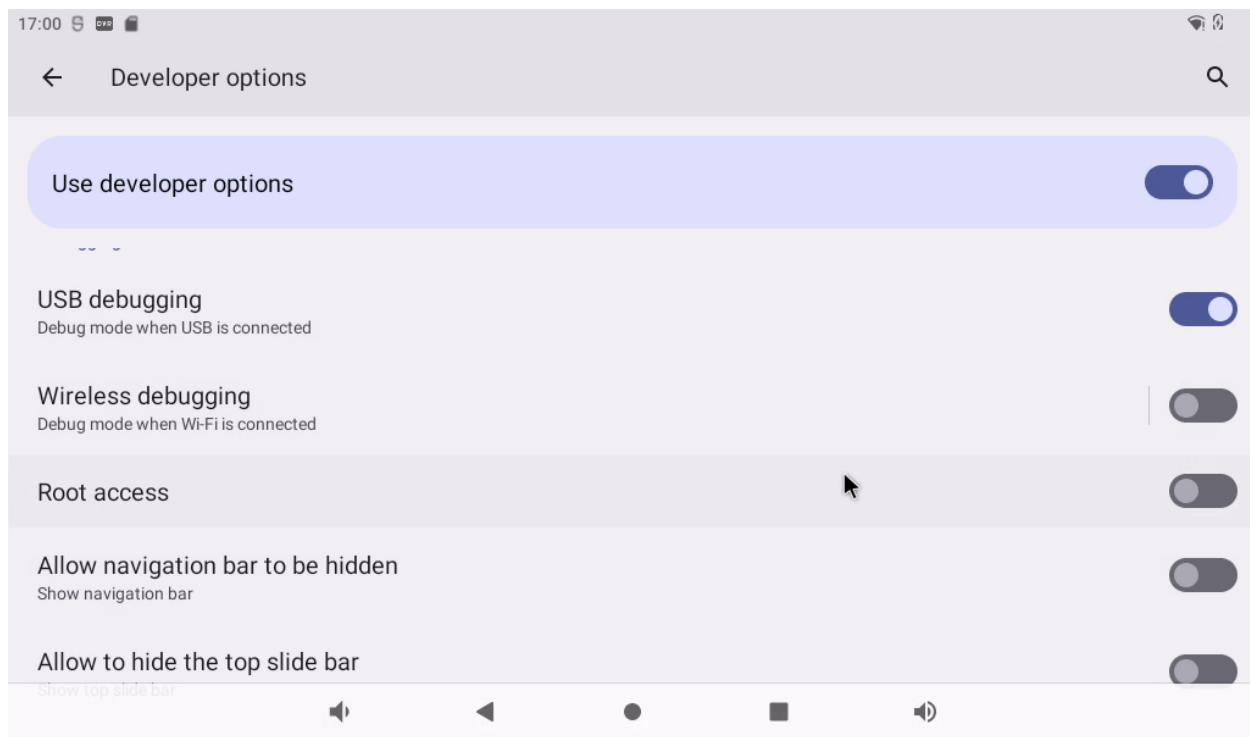
C:\Users\fl191>adb disconnect
adb server is out of date. killing...
* daemon started successfully *
disconnected everything

C:\Users\fl191>adb devices
adb server is out of date. killing...
* daemon started successfully *
List of devices attached

C:\Users\fl191>
```

3.26 ROOT Permission Test

Follow the previous sections to enter developer mode.



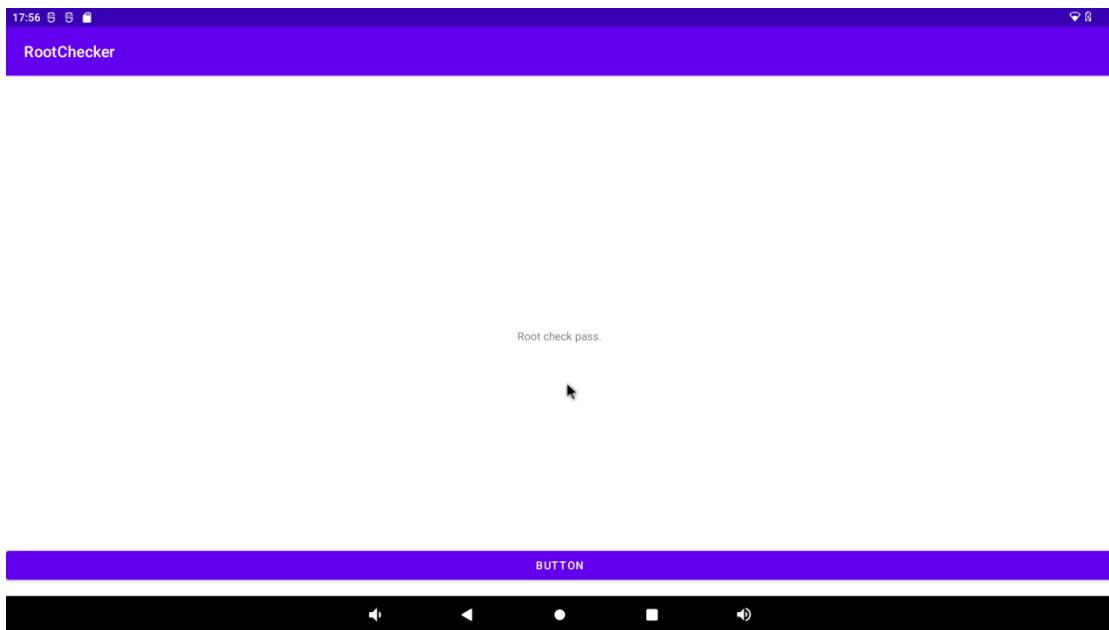
Click “Root Authorization” to restart the OK 3588 board.

Path: OK3588-C (Android) User Profile\Android\Tool\RootChecker.apk

Copy the RootChecker. Apk to any directory, open a window command window, and use ADB to install the RootChecker. Apk.

A screenshot of a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window displays the command 'adb install RootChecker.apk' being run, followed by the output 'Performing Streamed Install' and 'Success'. The command prompt window has a standard Windows interface with minimize, maximize, and close buttons.

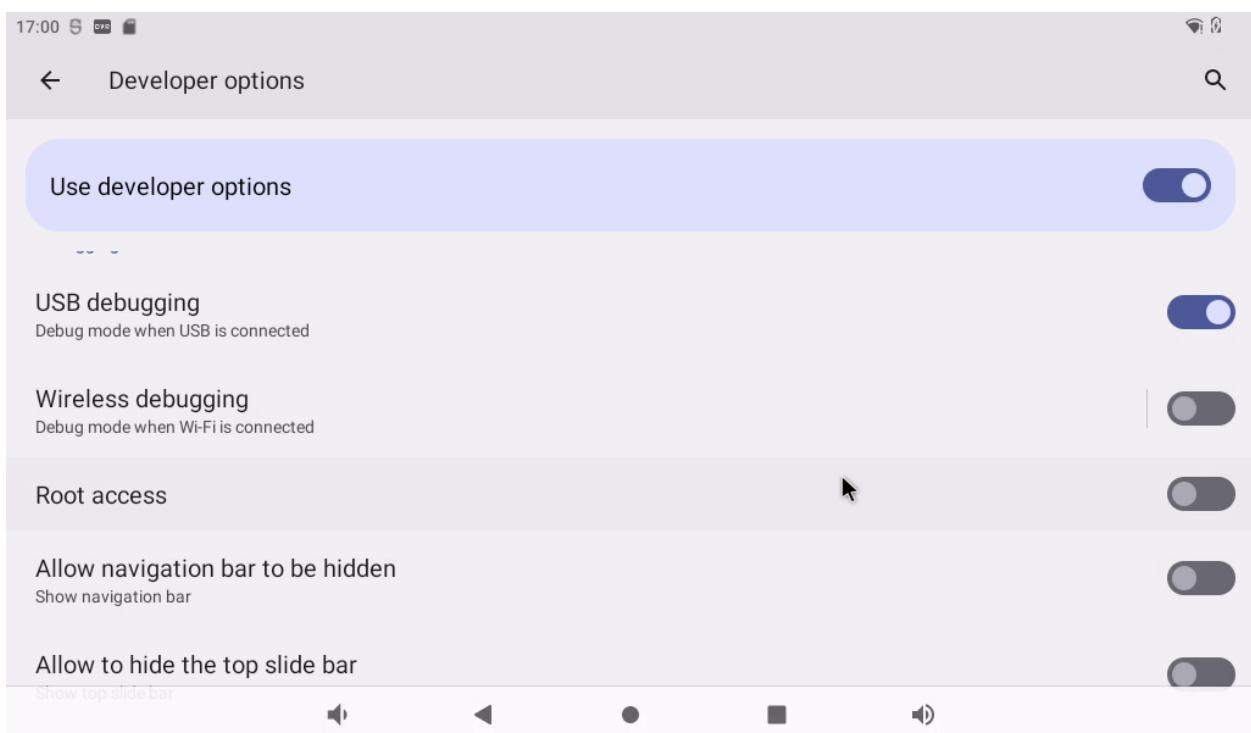
Slide to click the RootChecker icon “DisplayHwConfig”, on the OK3588 and click “BUTTON” .



3.27 Navigation Bar and Dropdown Bar Settings

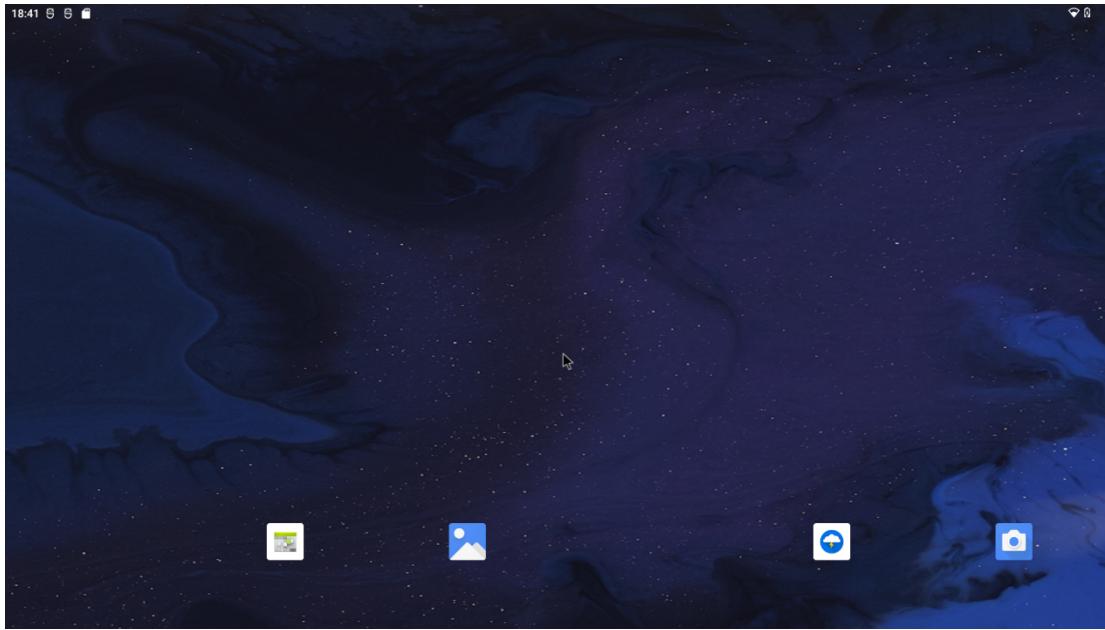
The current version supports showing/hiding navigation bar and slide bar.

Follow the previous sections to enter developer mode.



Select “Allow navigation bar to be hidden” and “Allow top slide bar to be hidden” .

Restart OK3588.



Block the navigation bar, top slide bar, and right mouse button for the return function.

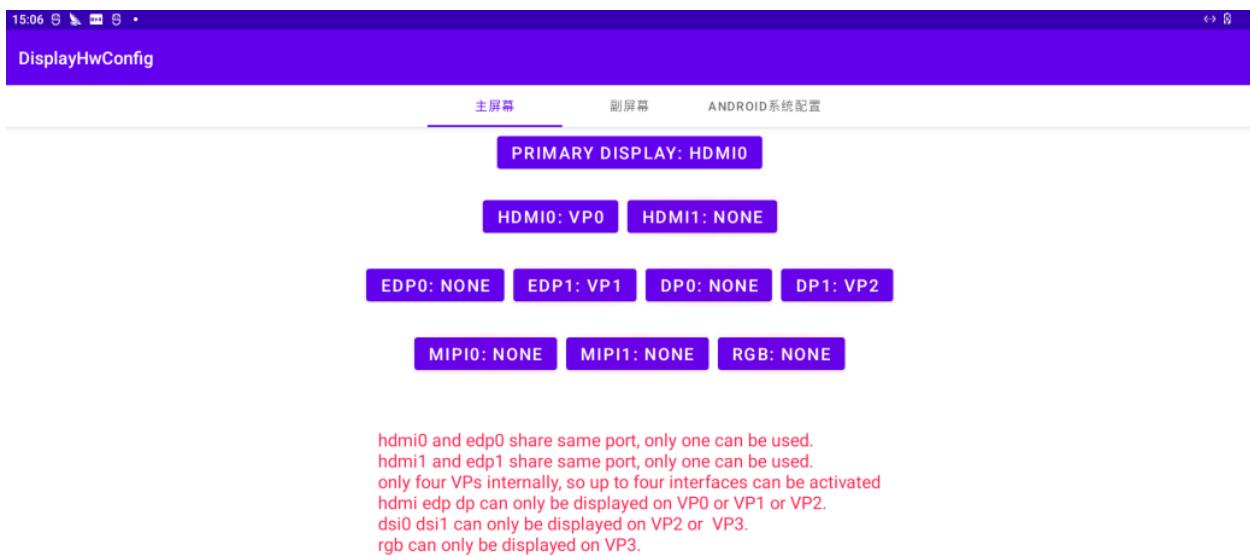
3.28 DisplayHwConfig Test



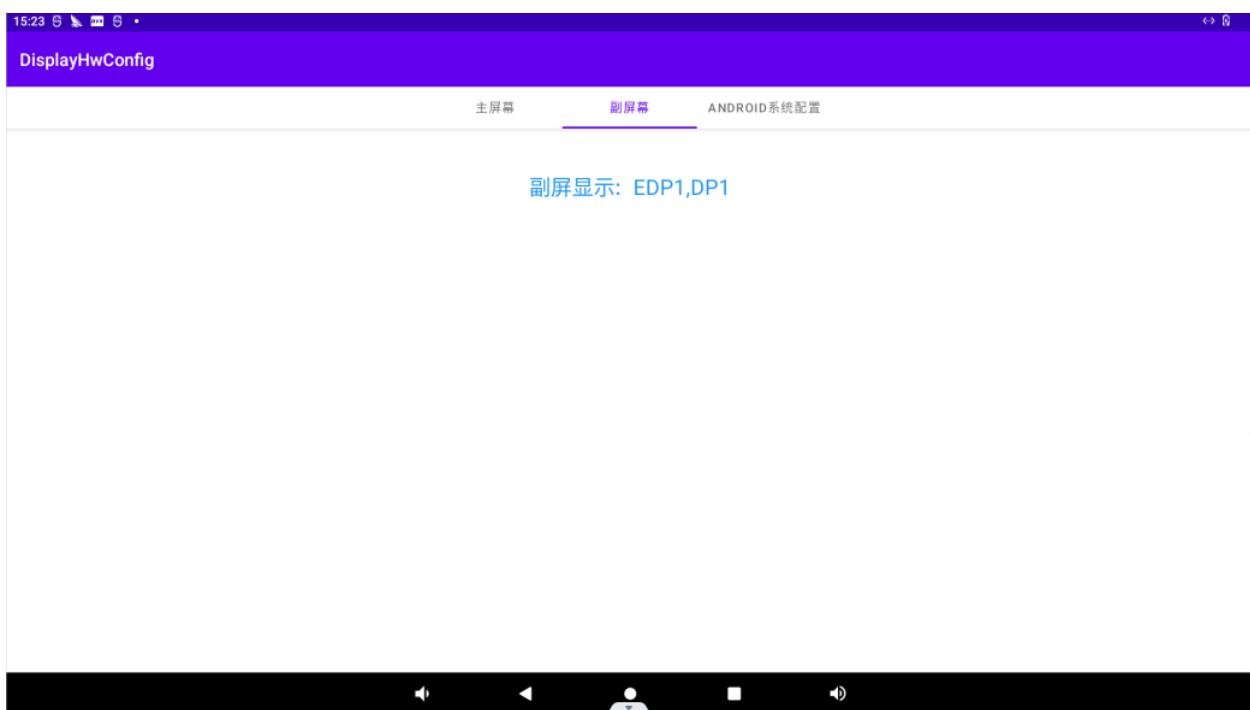
DisplayHwConfig can set the screen display and android system configuration. Click “DisplayHwConfig” program in the application interface to enter the program for display setting.

The program has three screens, the main screen, the secondary screen, and the ANDROID system configuration.

In the main screen interface, VP can be assigned to different interfaces according to the following rules. You can also specify which interface is the home screen.



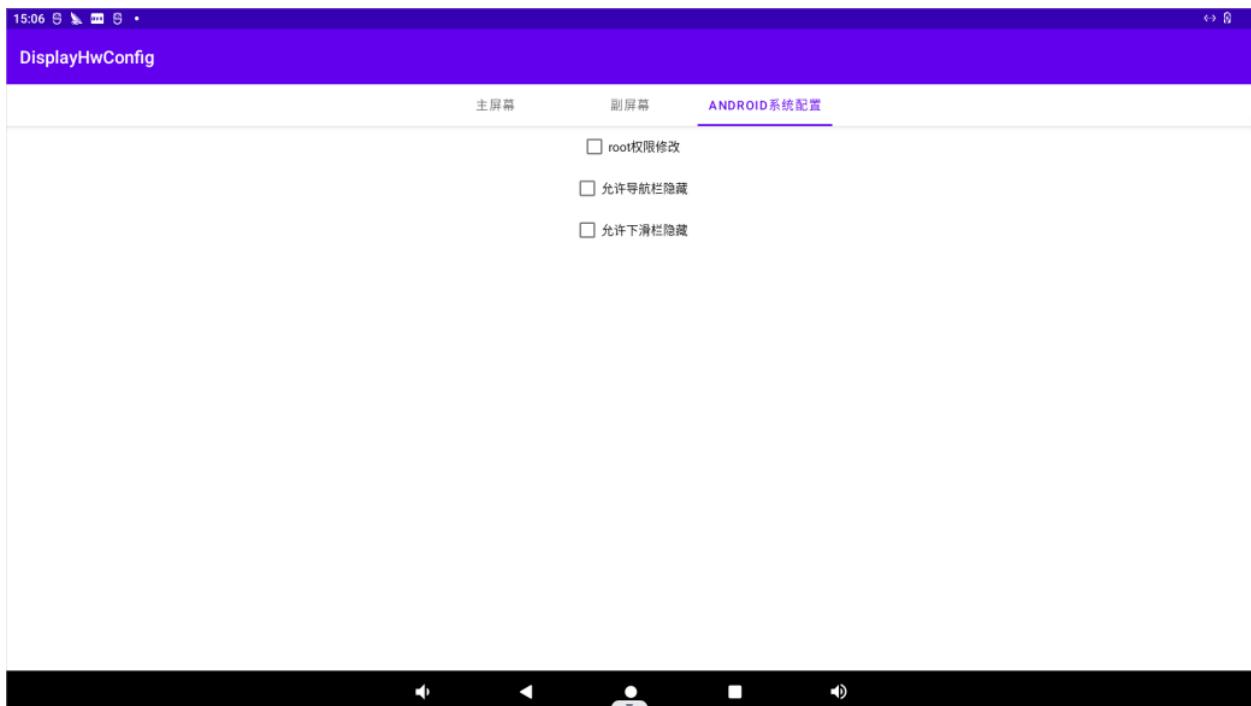
The secondary screen interface will show those previously displayed as secondary screen interfaces.



In the ANDROID system configuration interface, you can configure “modify root permission” , “allow the navigation bar to be hidden” , and “allow the sliding bar to be hidden” .

Check “Modify root permission” . After restarting, other users can also obtain root permission in the terminal.

Check “Allow the navigation bar to be hidden” and “Allow the sliding bar to be hidden” . After restarting, the system interface will not display the navigation bar and the sliding bar.



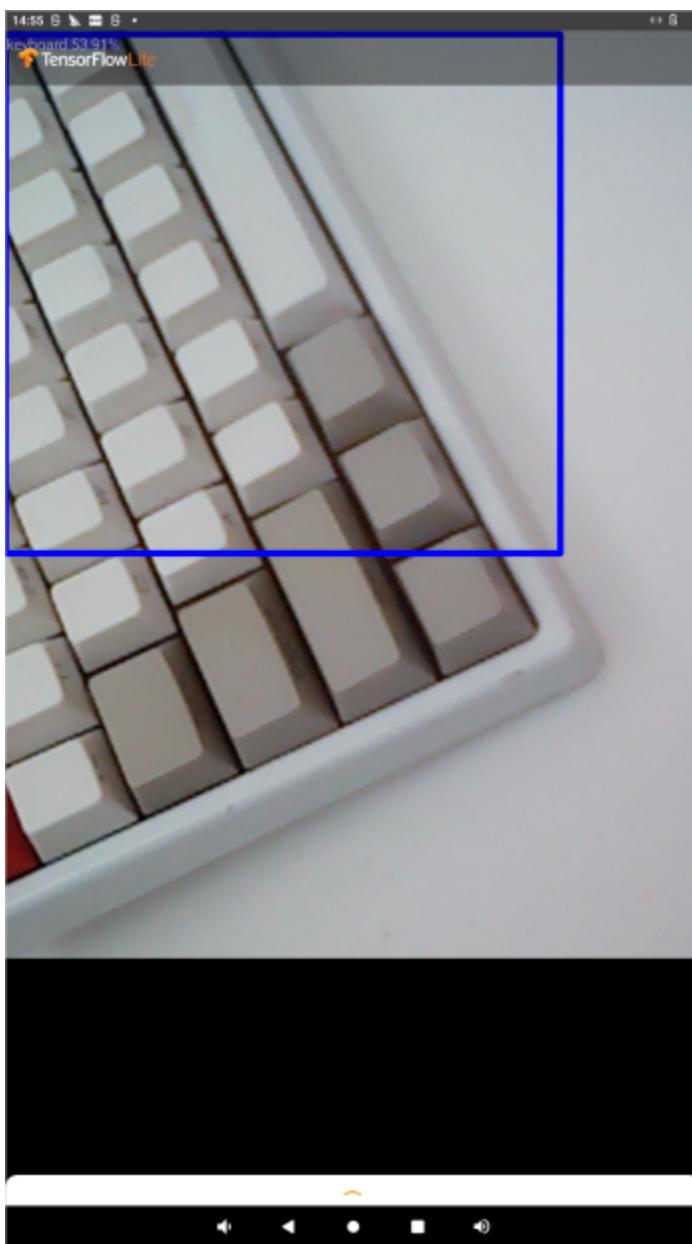
3.29 Artificial Intelligence Test

OK3588 android platform supports tensorflow lite and other mainstream AI frameworks. TFL Detect is used here to test target detection routines for customer reference. Routines such as accessories need to be installed by yourself.



The TFL Detect test routine is the official routine of tensorflow lite, which can run directly on the OK3588 platform without modification. It mainly uses the computing performance of the A-core, which occupies relatively high CPU, but it can often achieve higher detection frame rate by using multi-wire.

Test by placing the item in front of the camera will automatically recognize the item.



3.30 Locking Screen Test

Note: By default, Android will not lock the screen when it is started for the first time. If it is not modified, it will be opened in the lock screen state after restarting.

OK3588 does not lock the screen by default. If you need to lock the screen, you can modify it through Settings-> Security-> Screen Lock, as follows:



Under Screen Lock, you can specify how to lock the screen.

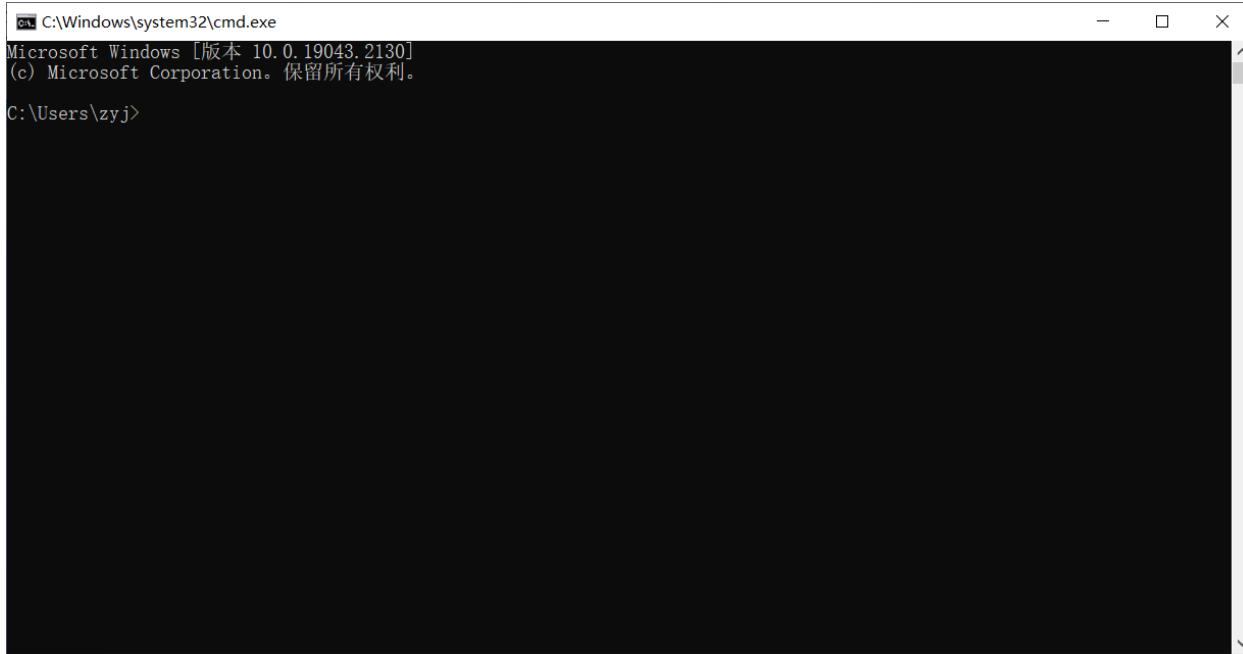
3.31 NPU Test

OK3588 The current test NPU examples are: rknn_mobilenet_demo_Android, rknn_ssd_demo_Android, and rknn_multiple_input_demo_Android, respectively, and the test routines are located at:

Path: OK3588-C (Android) User Profile\Android\Program\rknpu.tar.bz2

Here is an example of rknn_ssd_demo_Android to test the NPU:

Open the window command

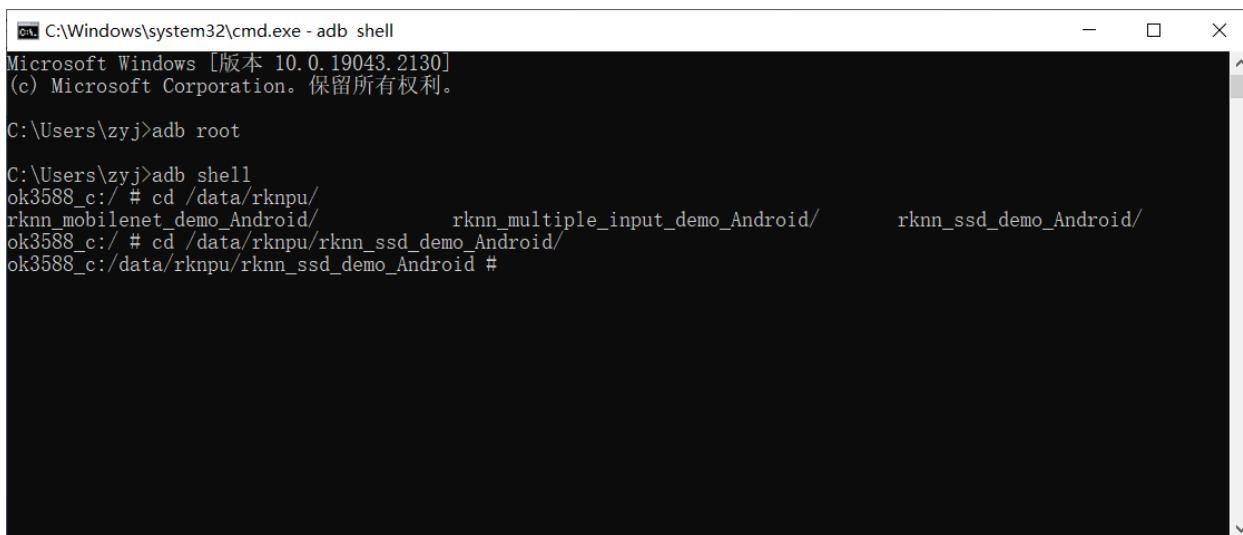


Enter the following command to upload the test routine to the development board file system.

```
adb root
adb push rknpu.tar.bz2 /data/
adb shell
cd /data
tar xvf rknpu.tar.bz2
```

Then enter the following command to test

```
cd /data/rknpu/rknn_ssd_demo_Android/
```



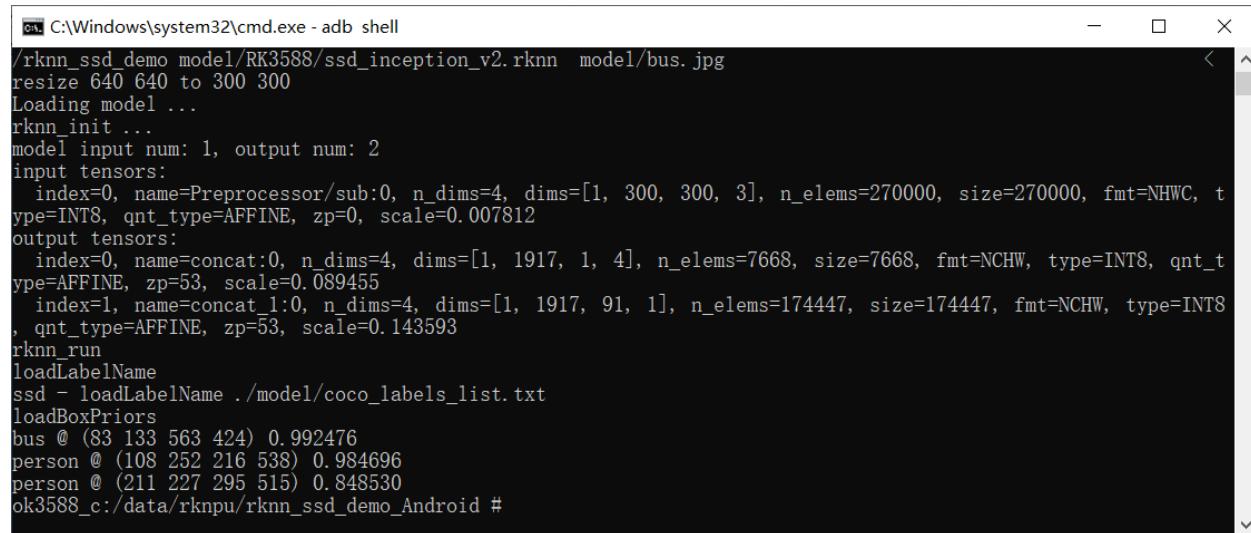
Run the rknn_ssd_demo as follows:

```
chmod +x rknn_ssd_demo
```

(continues on next page)

(continued from previous page)

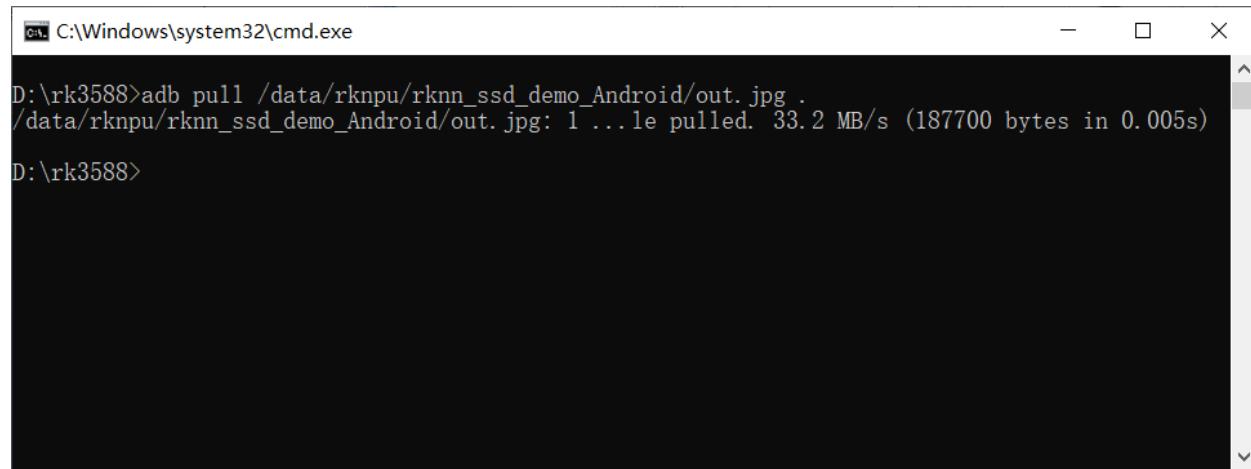
```
export LD_LIBRARY_PATH=../lib  
./rknn_ssd_demo model/RK3588/ssd_inception_v2.rknn model/bus.jpg
```



```
C:\Windows\system32\cmd.exe - adb shell  
/rknn_ssd_demo model/RK3588/ssd_inception_v2.rknn model/bus.jpg  
resize 640 640 to 300 300  
Loading model ...  
rknn_init ...  
model input num: 1, output num: 2  
input tensors:  
    index=0, name=Preprocessor/sub:0, n_dims=4, dims=[1, 300, 300, 3], n_elems=270000, size=270000, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=0, scale=0.007812  
output tensors:  
    index=0, name=concat:0, n_dims=4, dims=[1, 1917, 1, 4], n_elems=7668, size=7668, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=53, scale=0.089455  
    index=1, name=concat_1:0, n_dims=4, dims=[1, 1917, 91, 1], n_elems=174447, size=174447, fmt=NCHW, type=INT8, qnt_type=AFFINE, zp=53, scale=0.143593  
rknn_run  
loadLabelName  
ssd - loadLabelName ./model/coco_labels_list.txt  
loadBoxPriors  
bus @ (83 133 563 424) 0.992476  
person @ (108 252 216 538) 0.984696  
person @ (211 227 295 515) 0.848530  
ok3588_c:/data/rknpu/rknn_ssd_demo_Android #
```

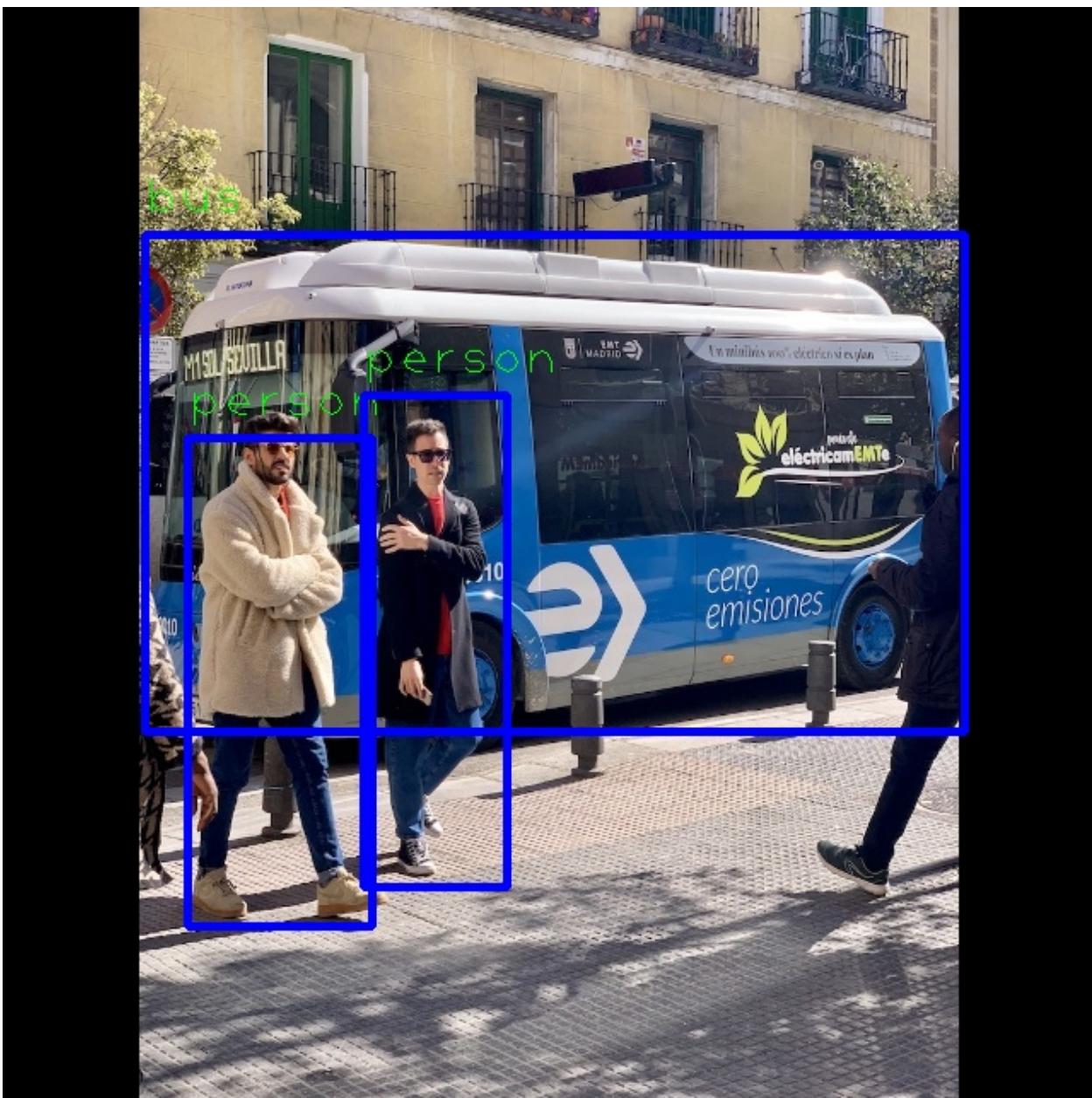
Pull the out.jpg file from the current directory to any directory in the window via adb.

```
adb pull /data/rknpu/rknn_ssd_demo_Android/out.jpg .
```



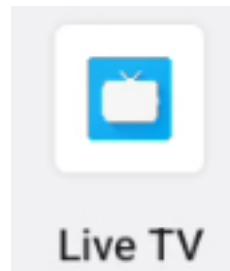
```
C:\Windows\system32\cmd.exe  
  
D:\rk3588>adb pull /data/rknpu/rknn_ssd_demo_Android/out.jpg .  
/data/rknpu/rknn_ssd_demo_Android/out.jpg: 1...1e pulled. 33.2 MB/s (187700 bytes in 0.005s)  
D:\rk3588>
```

Open the out.jpg file by using the diagram viewing software

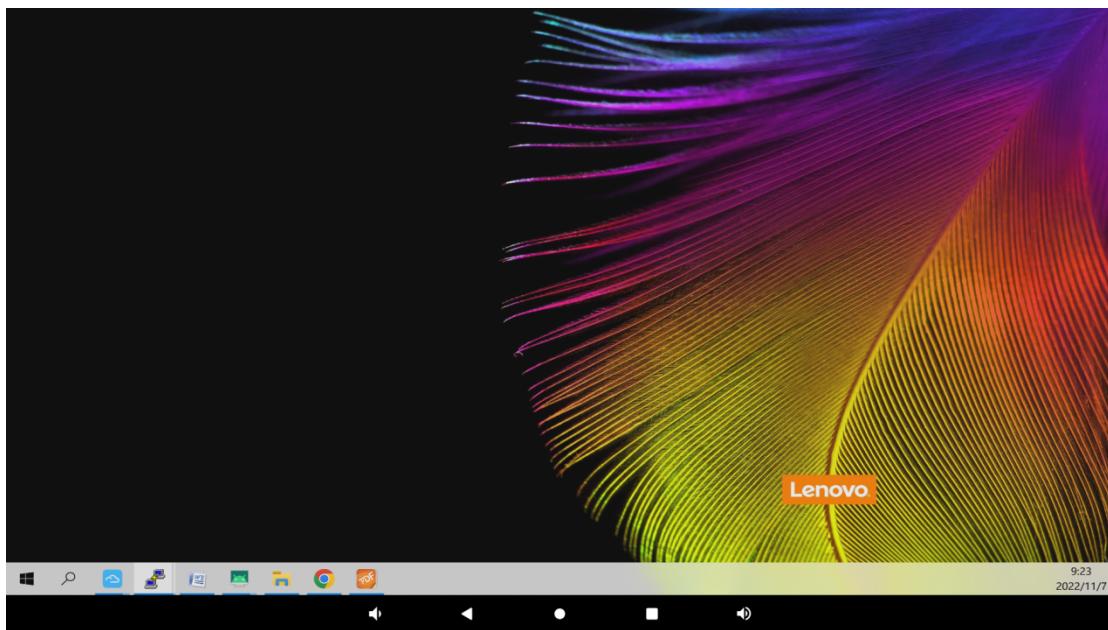


3.32 Hdmi rx Test

OK3588 HDMI RX supports resolutions up to 3840x2160 @ P60 and 4096x2160p @ P24.



After plugging in the HDMI cable, open “ ” Live TV in the application interface.



3.33 Forlinx-SDK Test

Forlinx provides a test application to demonstrate how to use some functions on the board.

The path of the test application is:

Path: OK3588-C (Android) User Profile\Android\Program\Forlinx-SDK.apk

You can install it manually through the file manager via USB flash drive or TF card, or use the following adb command for installation.

```
adb install Forlinx-SDK.apk
```

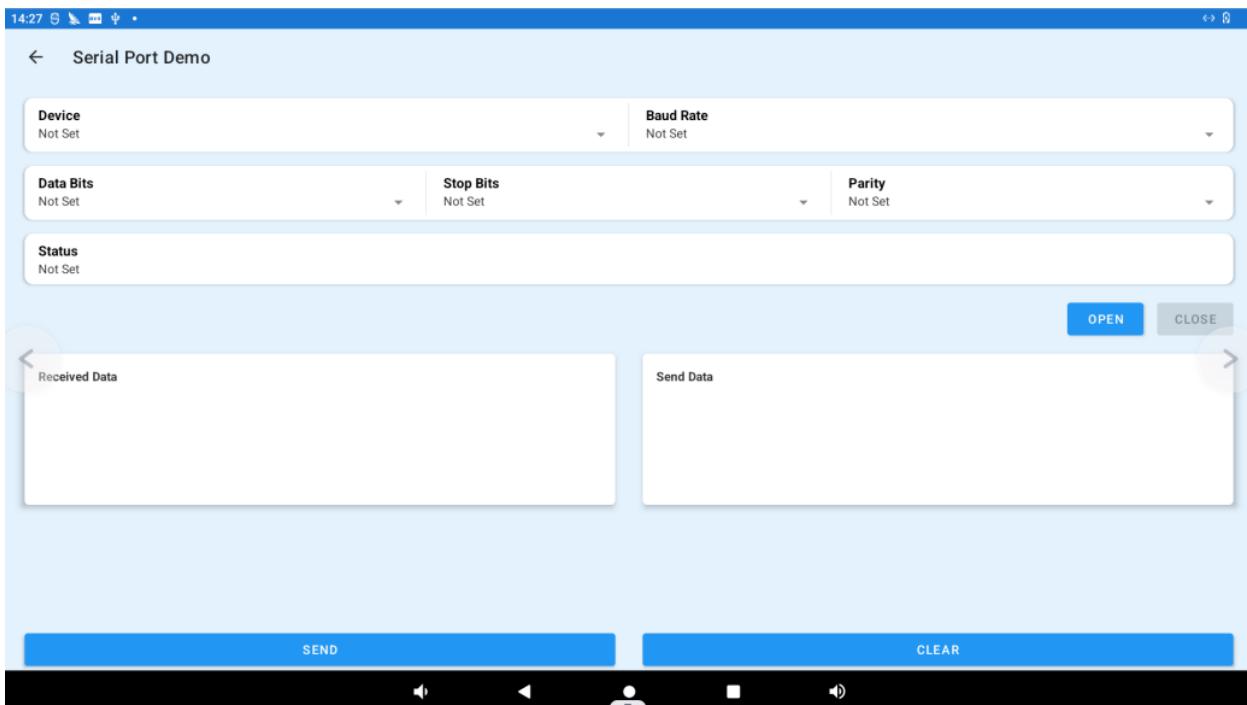
After installation, click “Forlinx-SDK” in the application interface to enter the program.

The content is shown in the figure below. The Forlinx-SDK program provides the test demo of serial port, SPI, I2C, GPIO, WATCHDOG, ADC, CAN, etc. These are described in turn below.



3.33.1 SERIAL PORT DEMO

Click “SERIAL PORT DEMO” “to enter the serial port test interface.



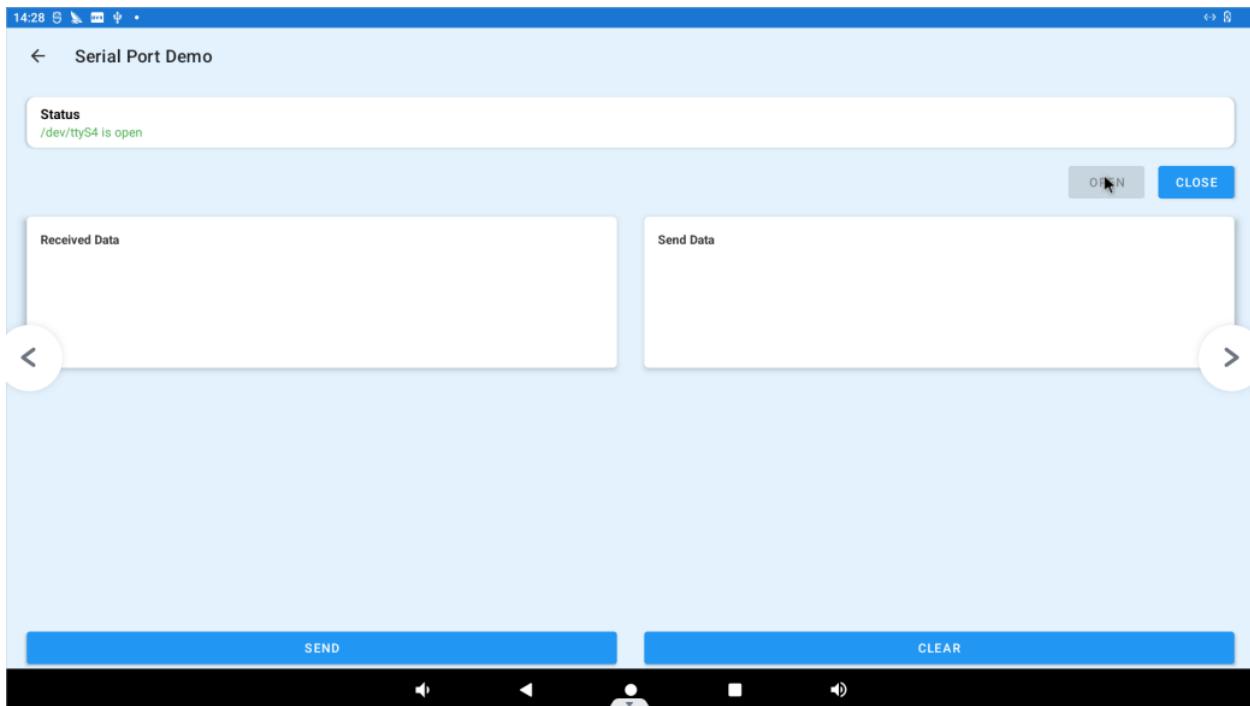
Device: Select the serial port to be tested.

Baud Rate: Select the baud rate of the serial port.

Data Bits: Specify serial port data bits.

Data Bits: Specify serial port stop bits.

Connect TX, RX of the corresponding uart4 serial port on the OK3588 development board (pin positions as shown in Figure 1 3.19 Section).



After setting the above contents, click the OPEN button to start the test

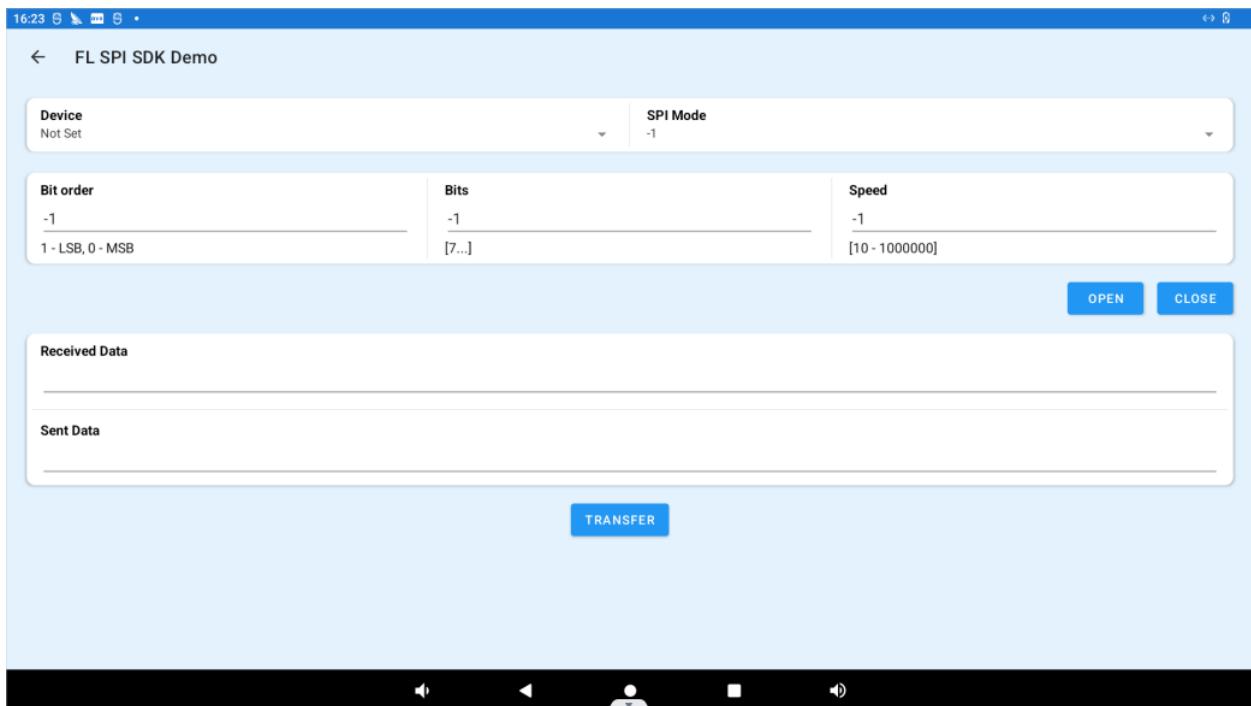
Display the set serial port information in the Status column.

Add the content to be sent in the Received Data column, and click SEND to send data.

Send data will be received in the Send Data column. Click CLEAR to clear the received data in the Send Data column.

3.33.2 SPI DEMO

Click “SPI DEMO” “to enter the SPI test interface (no SPI test interface is reserved on the 3588 board).



Device: Specify the SPI to use.

SPI Mode: Specify one of the four modes.

Bit order: Specify one of 1 (LSB), 0 (MSB).

Bits: Specify the number of bits in the data.

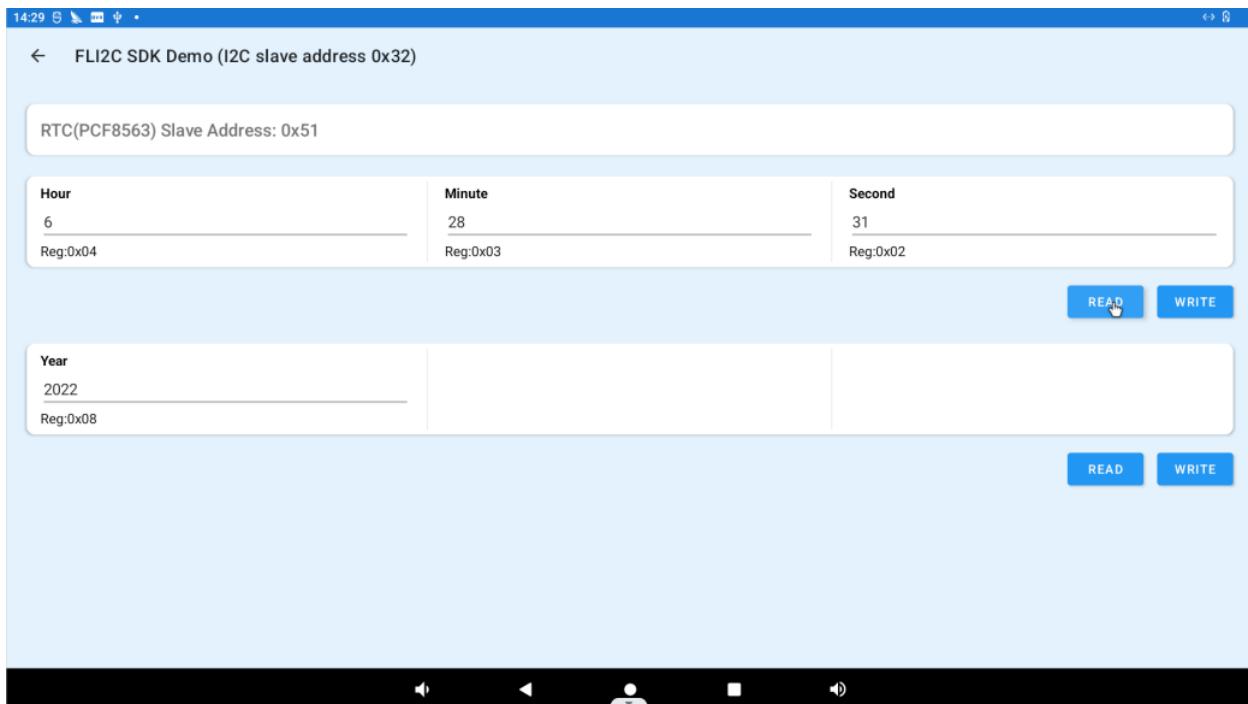
Speed: Specify the transmission rate in the range of 10-1000000.

After the above is set, click OPEN to start the test. Connect the miso and mosi pins of the selected spi on the OK3588.

Enter the data to be sent in the Sent Data field and receive the sent data in the Received Data field.

3.33.3 I2C DEMO

Click “I2C DEMO” to enter the i2c test interface.

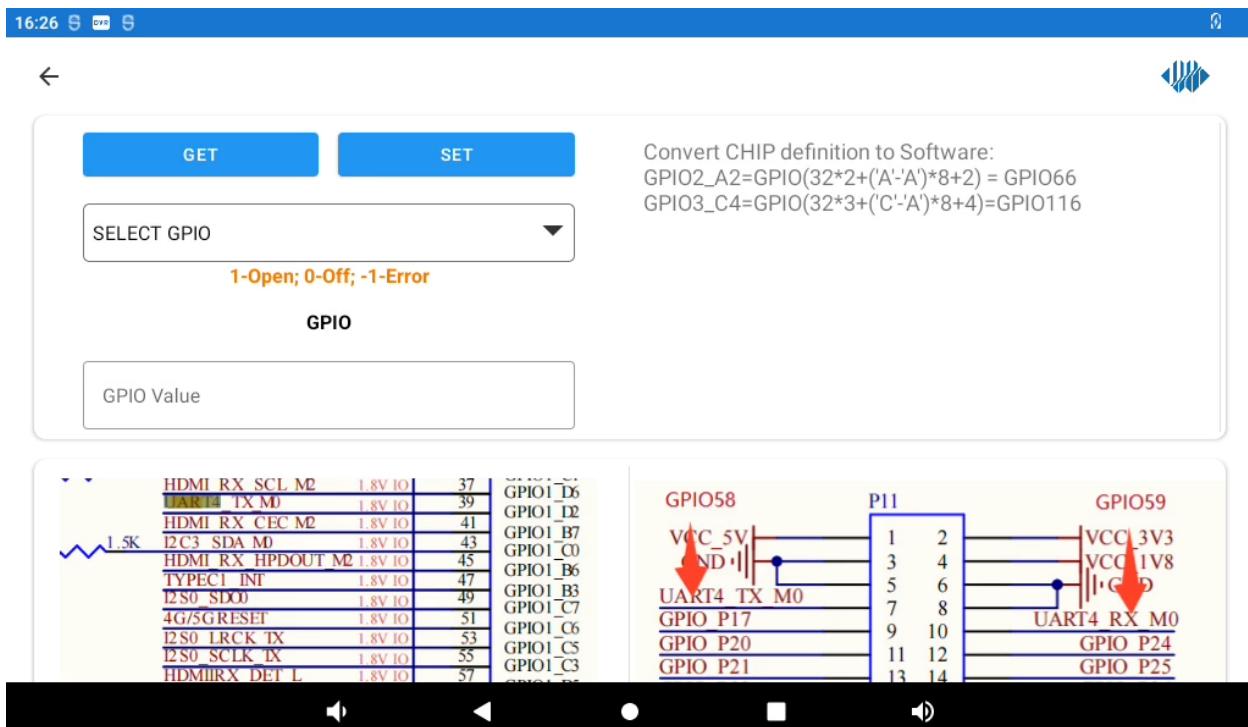


The RTC chip is connected to the I2C, and clicking the READ button will read the time in the RTC register.

Fill in the time in the columns of Year, Hour, Minute, and Second, and click the WRITE button to write the data into the corresponding registers of the RTC.

3.33.4 GPIO DEMO

Click GPIO DEMO “to enter the gpio test interface.



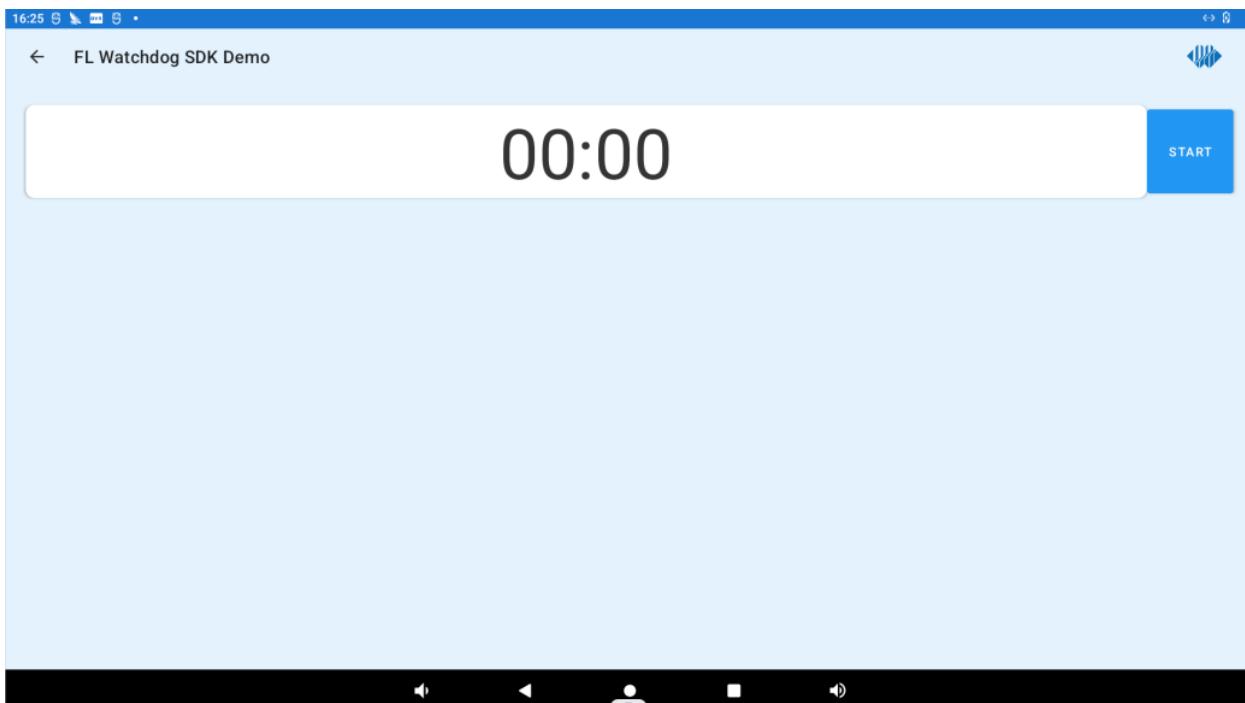
The formula in the upper right corner of the above figure, is the method of calculating the gpio serial number.

Select GPIO in the drop-down bar, click the GET button, and get the high level 1 or low level 0 in the GPIO Value bar.

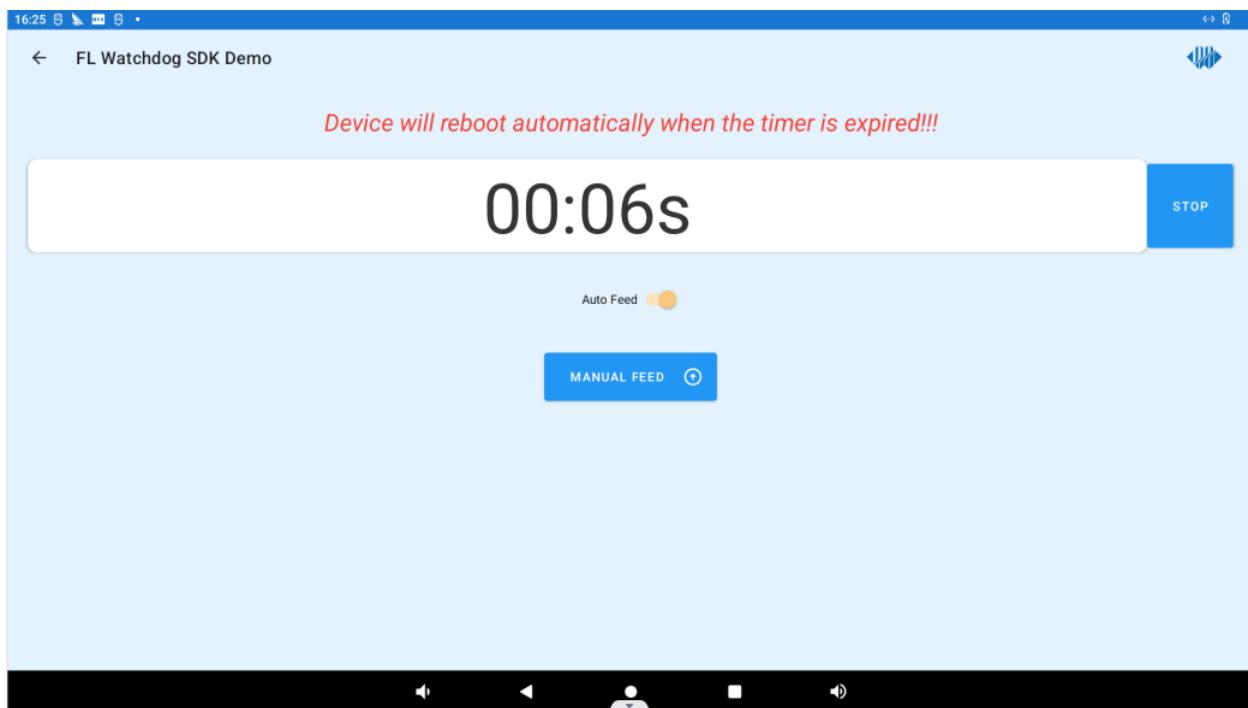
Enter a 1 or 0 in the GPIO Value field and click the SET button to set the GPIO output high or low specified in the drop-down field.

3.33.5 WATCHDOG DEMO

Click “WATCHDOG DEMO” to enter the Watchdog test interface.



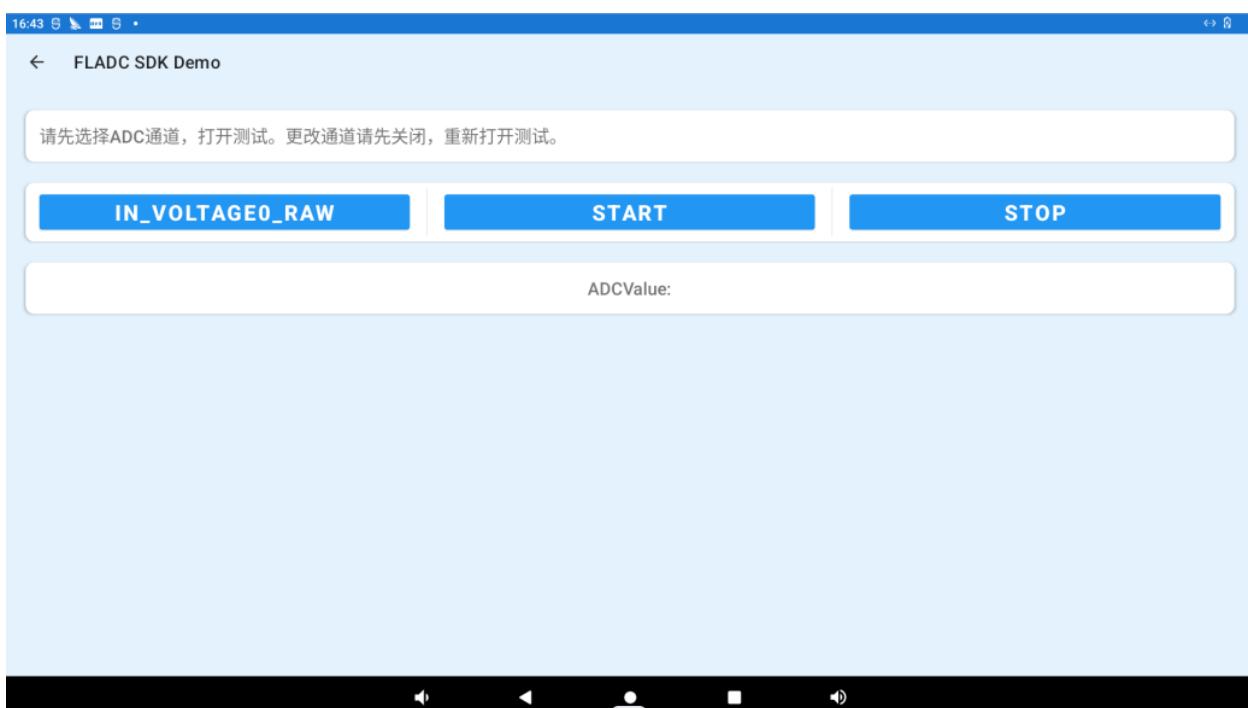
Clicking the START button will start the watchdog and start a 15-second countdown, when the countdown reaches 0, the device will reboot.



Turning on Auto Feed will enable automatic dog feeding; clicking on MANUAL FEED “will manually feed the watchdog once. Clicking the STOP button will stop the watchdog.

3.33.6 ADC DEMO

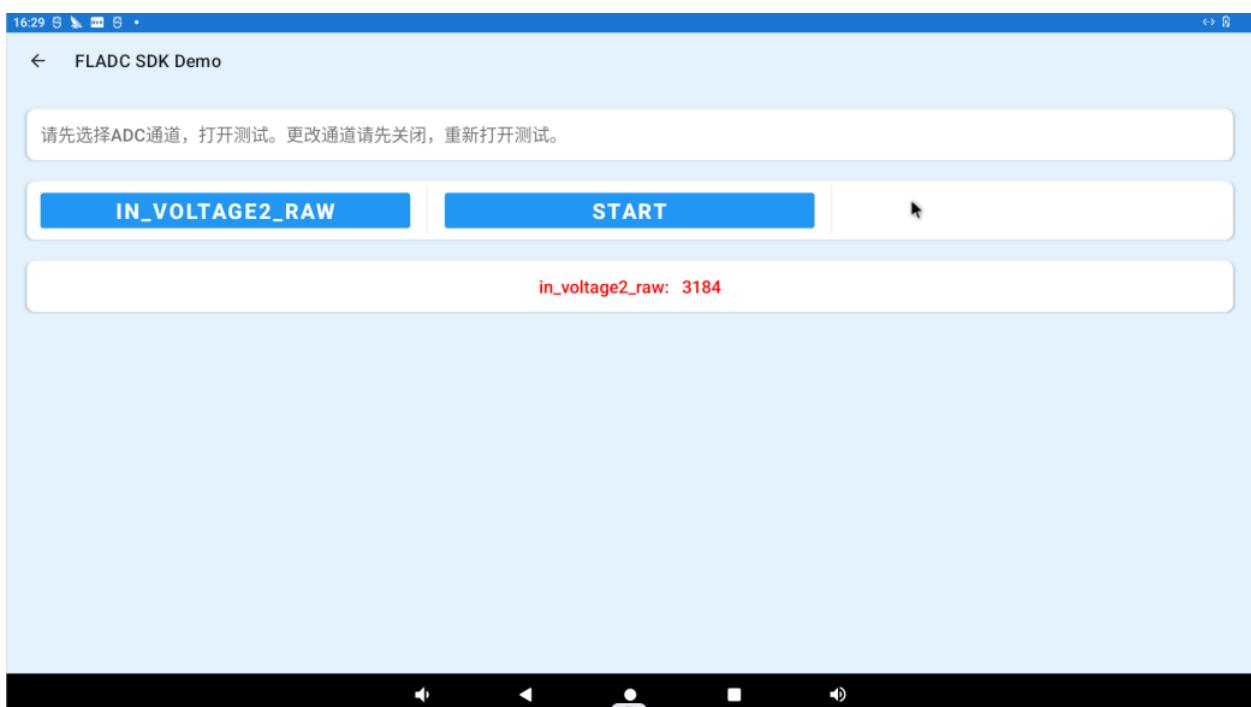
Click “ADC DEMO” to enter the adc test interface.



Click IN_VOLTAGE0_RAW to select the ADC channel, click the START button to start the test, the ADC value will be displayed in the “ADCValue” column.



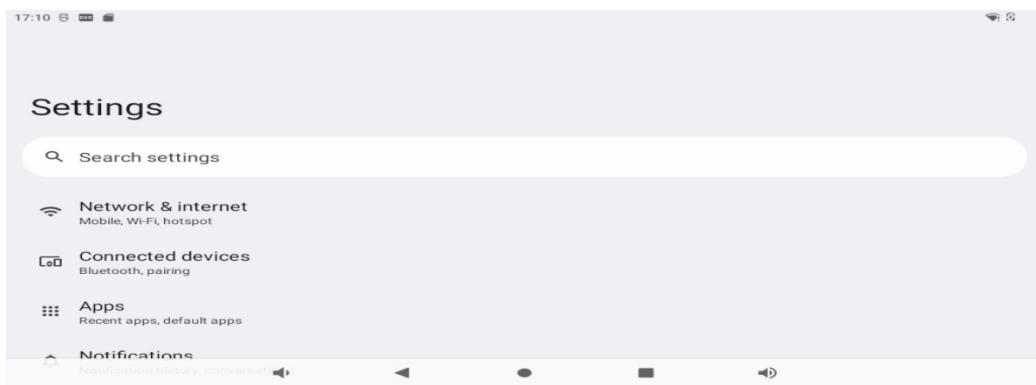
Click “STOP” to stop the test:



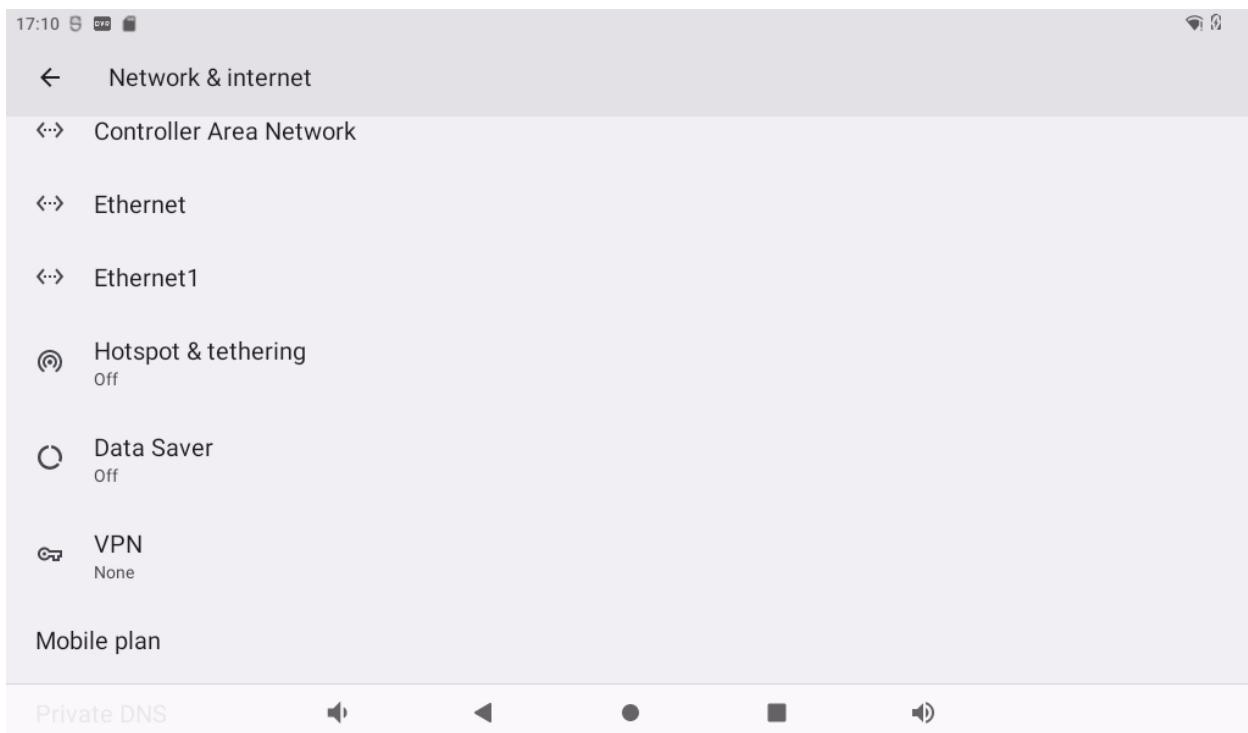
3.33.7 CAN DEMO



Click “ ” , on the application interface to enter the setting interface:



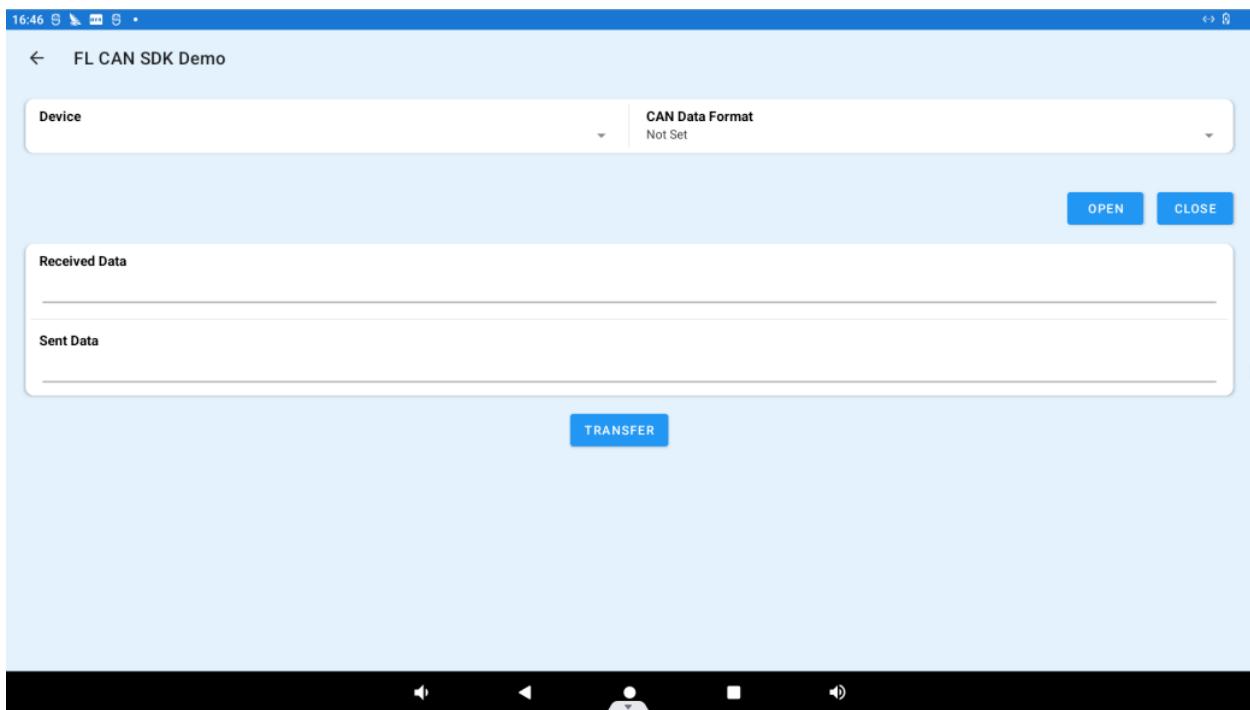
Click on “Network and Internet” to enter.



Click “Controller Area Network” to enter the can setting interface. Set the mode and use frequency of the two can in this interface.



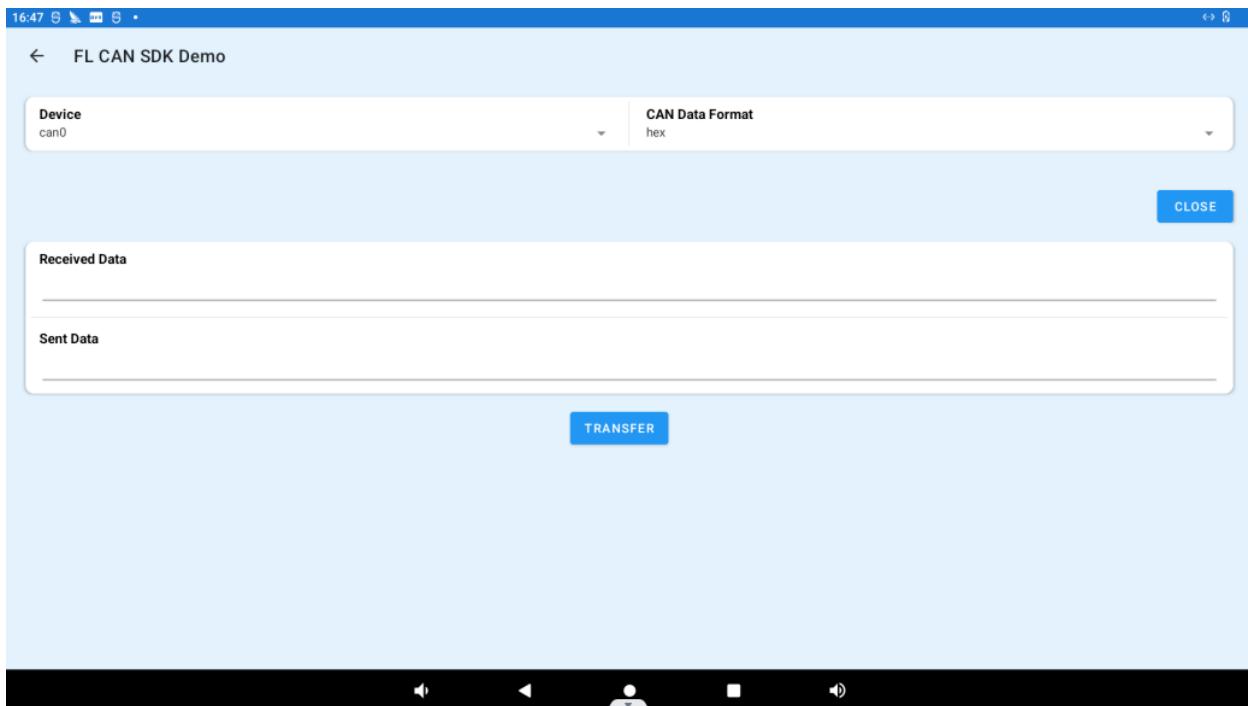
Open the “Forlinx-SDK” application and click CAN DEMO to enter the can test interface.



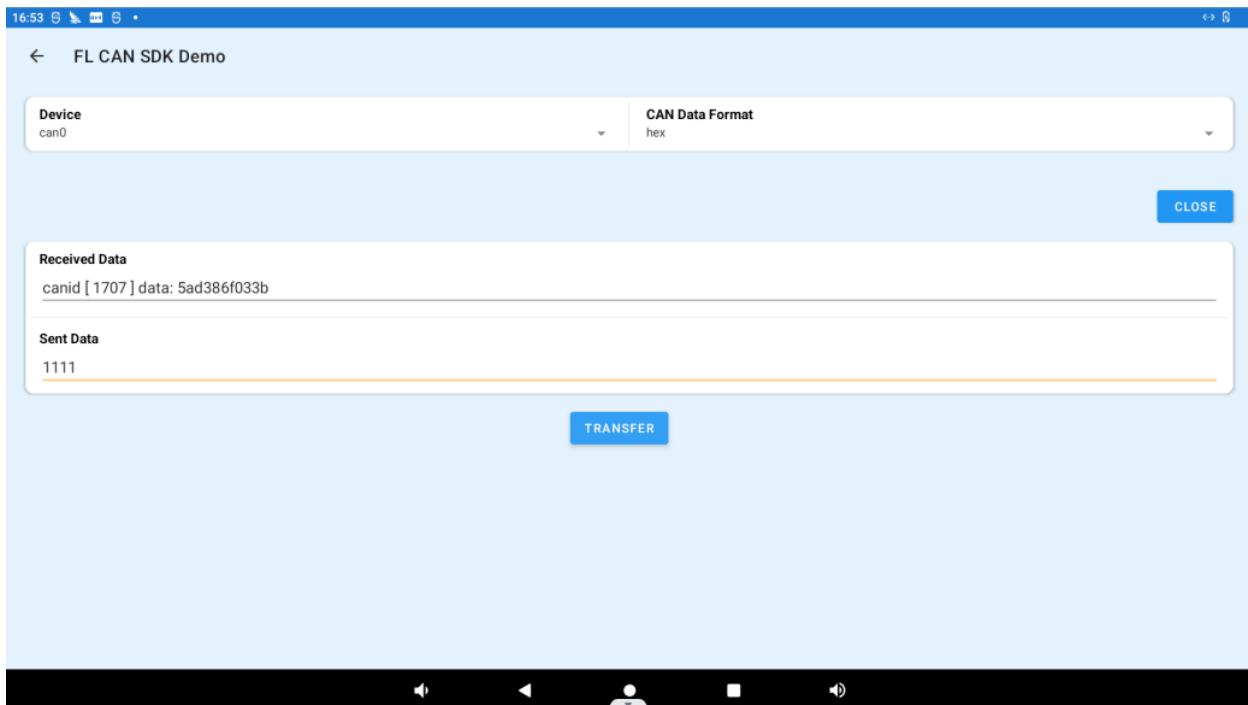
Device: Specify the can interface to test.

CAN Data Format: Specify the format of data transmitted by can, hexadecimal or character.

Click the OPEN button to start the selected can interface.



Short-circuit can0 and can1. Enter “candump can1” in the command line to receive data, write the data to be transmitted in the Sent Data column, click the TRANSFER button to send the data, and the command line will receive the sent content. Enter “cangen can1” on the command line, and the Received Data column will display the received data.



2.3.8 4. System Flashing

4.1 OTG System Flashing

4.1.1 OTG Driver Installation

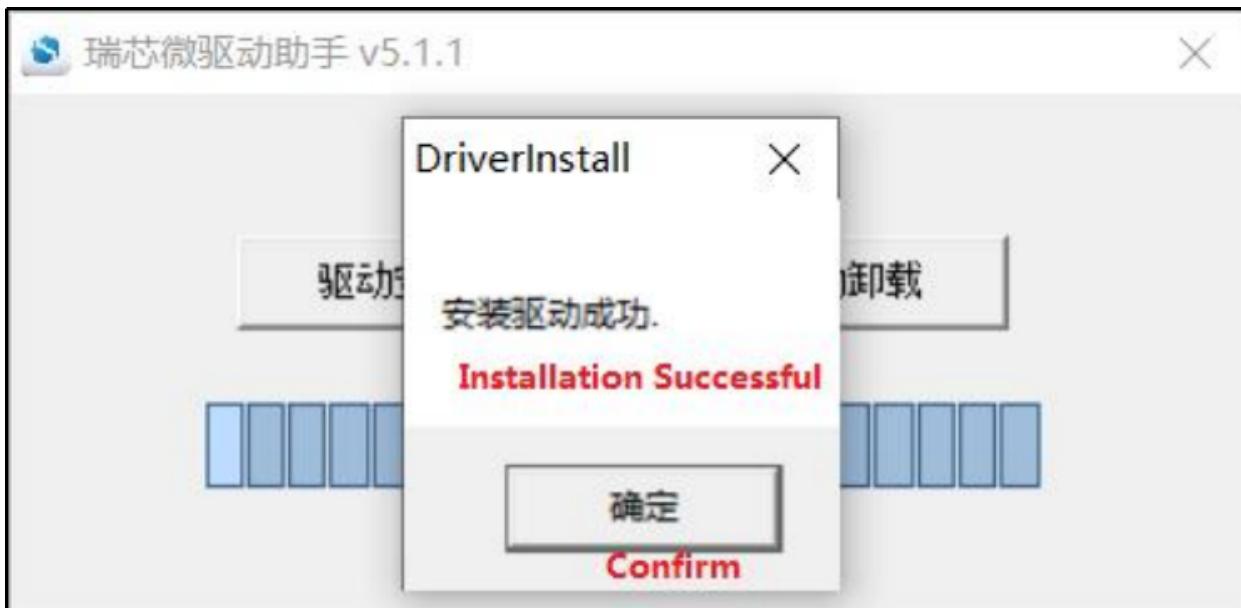
Path: OK3588-C (Android) User Profile\Android\Tool\DriverAssitant_v5.11.zip

Extract the above path file to any directory and run it with administrator privileges

Open DriverInstall.exe



Click “Driver Installation”



4.1.2 OTG Flashing Test

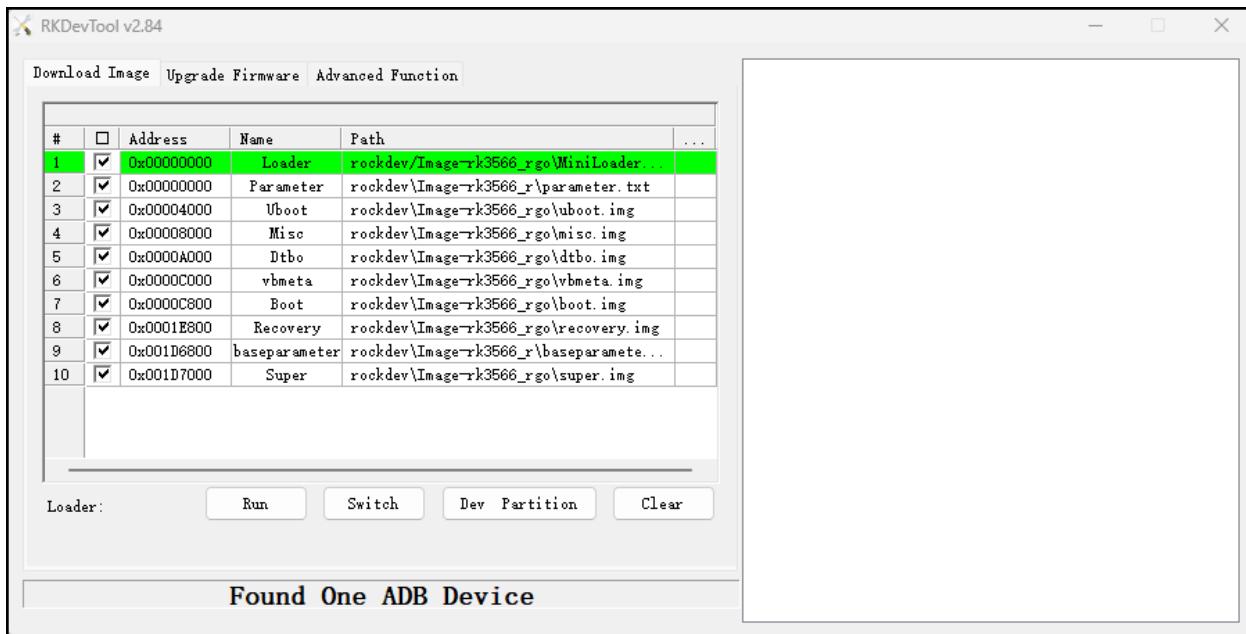
4.1.2.1 RKDevTool Flashing Test

Path: OK3588-C (Android) User Profile\Android\Tool>RKDevTool_Release_v2.84.zip

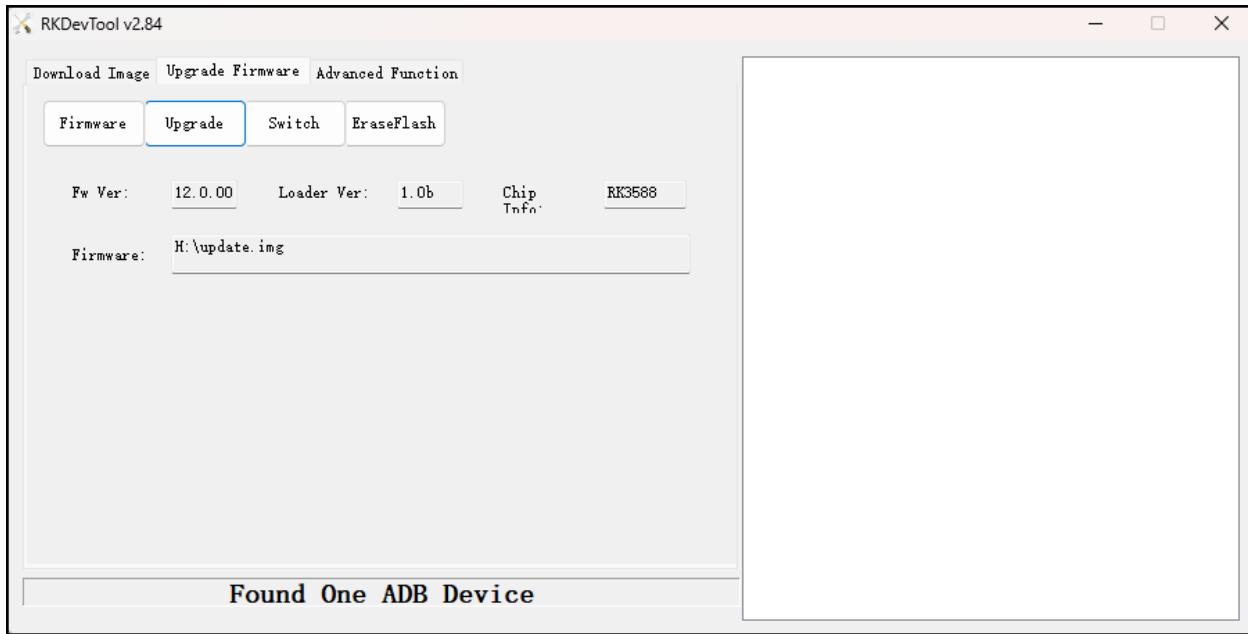
It is a development tool provided by Rockchip Micro. Unzip it to a full English path before use, connect the Typc0 port of the development board and the host computer with a Type-C cable, press and hold the recovery key of the development board and don't release it, then press the reset key to reset the system, and release the recovery key after about two seconds. There will be prompts on the Rockchip development tool : loader device found

Note: The operation to recognize the device is that the recovery button should be in the pressed state when the development board is powered on. Theoretically, Rockchip development tools have no requirements for the unzip directory. However, some users have feedback that the unzip directory should be in full English. If the tool doesn't match the following figure, please consider unzipping it in an English directory.

Open the Rockchip development tool:



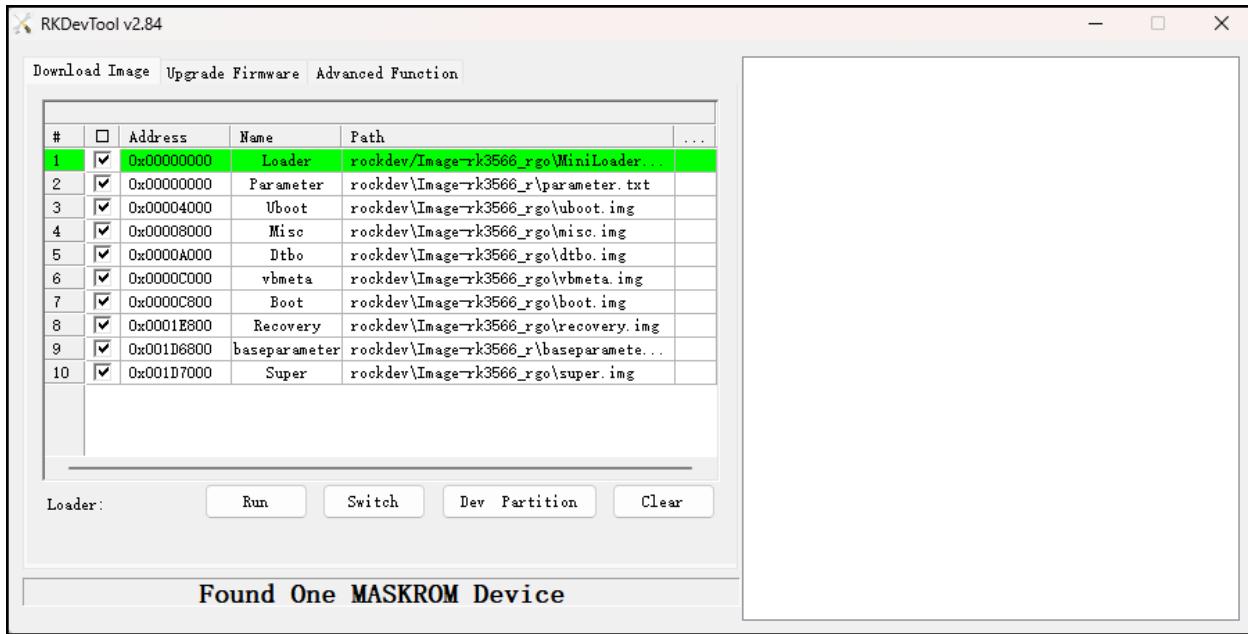
Click the “Upgrade Firmware” tab, click the “Firmware” button to select the full upgrade image update.img. The program will be parsing the firmware, so wait a while.



Click the “Upgrade” button to upgrade.

Introduction to MASKROM mode

If the loader is damaged and cannot enter the Loader mode, press and hold the red Maskrom key and then press the reset key to enter the maskrom mode for flashing.



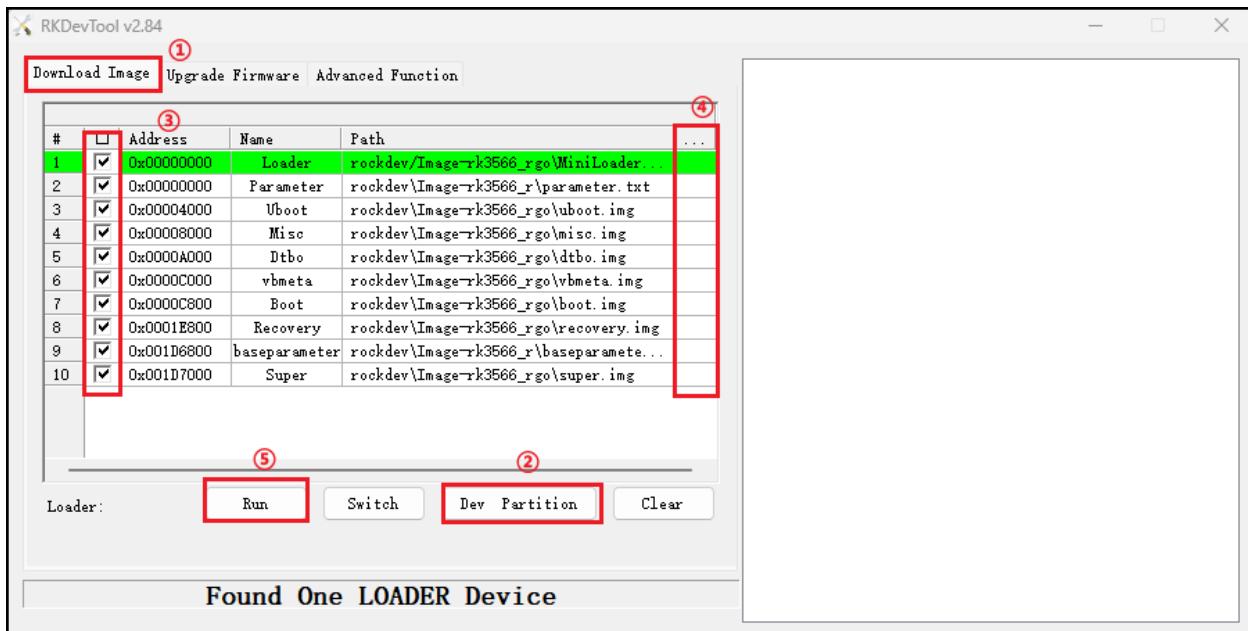
At this time, the system will prompt the discovery of a maskrom device. The flashing process is consistent with the loader mode, so it is best to use an update.img burning.

Note: Don't click “Device Partition Table” in maskrom mode, it is invalid. A separate burn in maskrom mode will not clear the UBOOT environment variables.

Introduction to Downloading the Individual Image Function

This feature is useful when you need to download a separate image. This function is only applicable in loader flashing

mode.



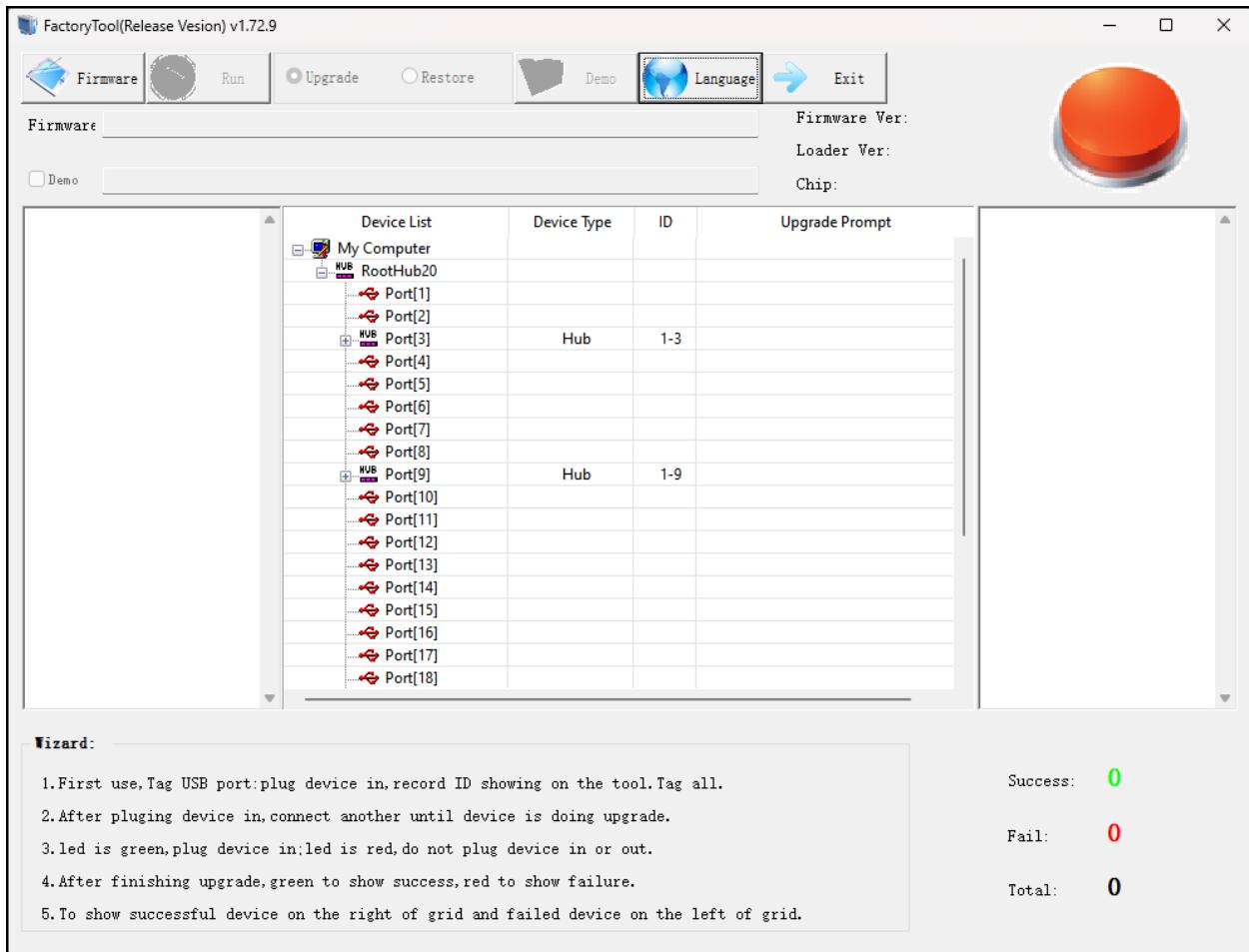
1. Click ① Download image tab
2. Click ② Device partition table to read the mirror partition location
3. Click the ③ check box to select the image to be flashed separately
4. Click ④ Here to select a image
5. Click ⑤ to execute for flashing
6. Restart after flashing.

4.1.2.2 Factory Tool Flashing Test

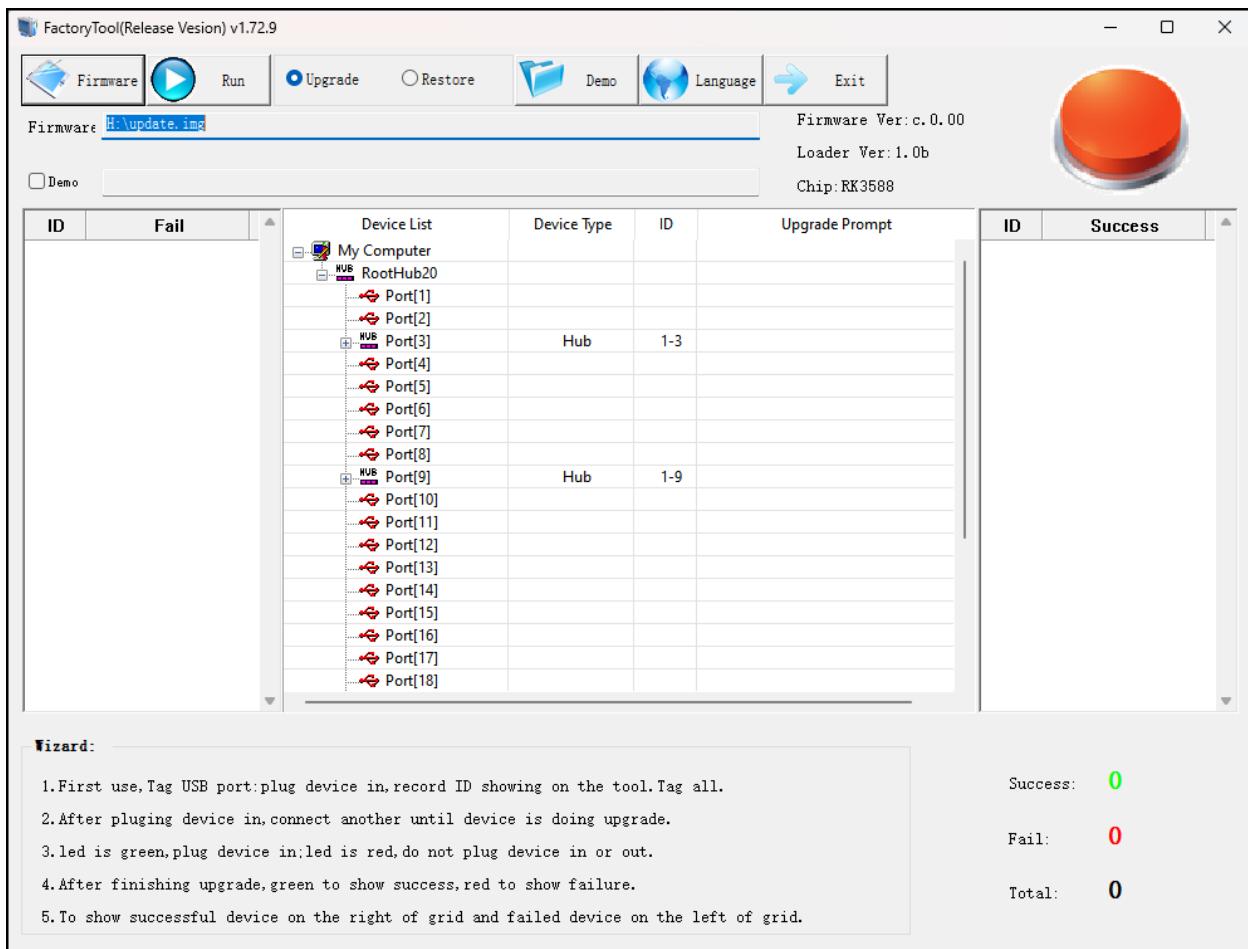
Factory Tool is a factory batch OTG flashing tool, which does not need to read the image and supports large file flashing. Use this tool if RKDevTool is not compatible. Before use, you need to decompress to the full English path, connect the development board to the host, press the recovery key, press the reset key to reset, and release the recovery key after two seconds. There will be prompts on the Rockchip development tool : loader device found

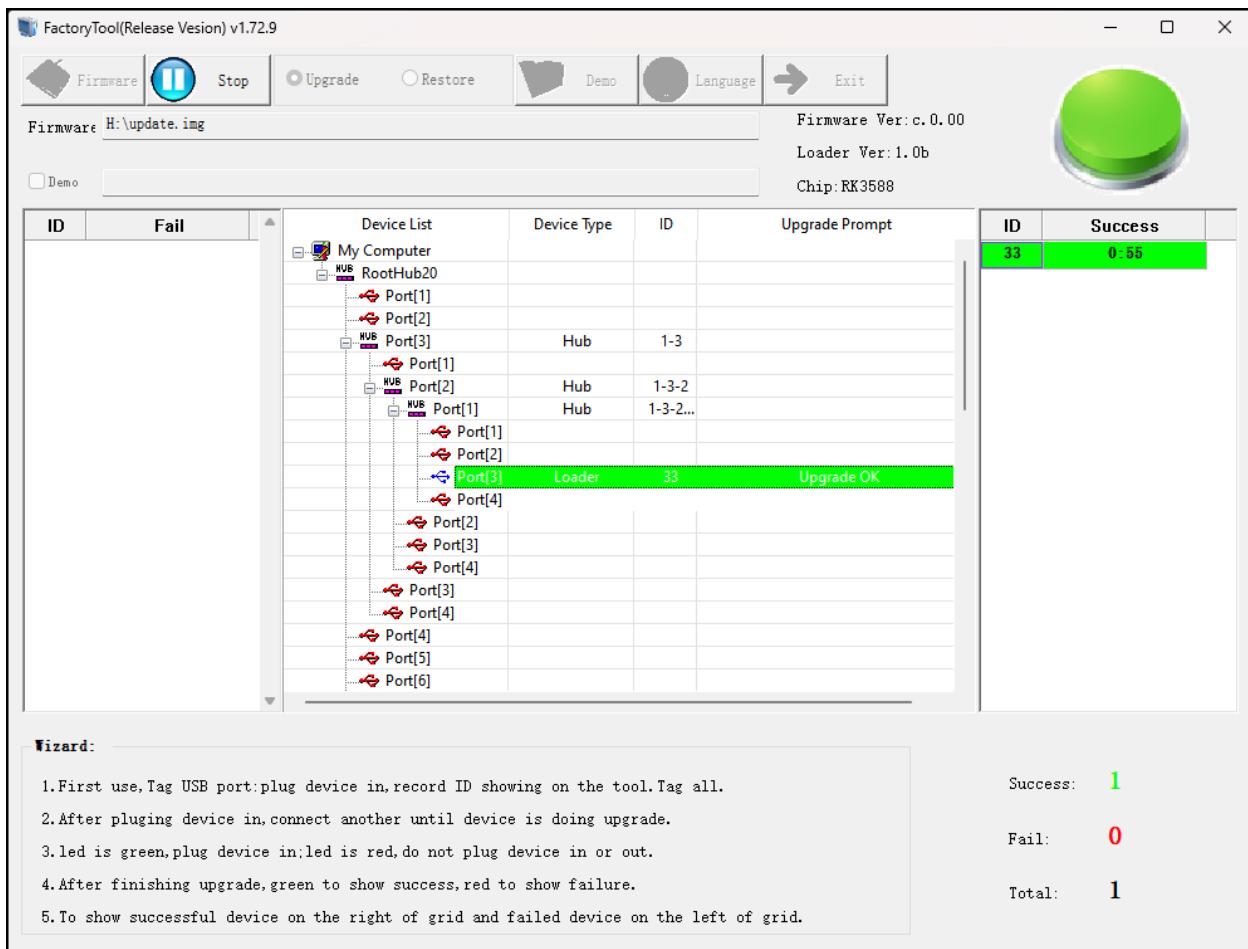
Note: The operation to recognize the device is that the recovery button should be in the pressed state when the development board is powered on. Theoretically, Rockchip development tools have no requirements for the unzip directory. However, some users have feedback that the unzip directory should be in full English. If the tool doesn't match the following figure, please consider unzipping it in an English directory.

Open the Rockchip development tool:



Click to select the firmware, and click to start. At this time to recognize the loader device will automatically start burning.



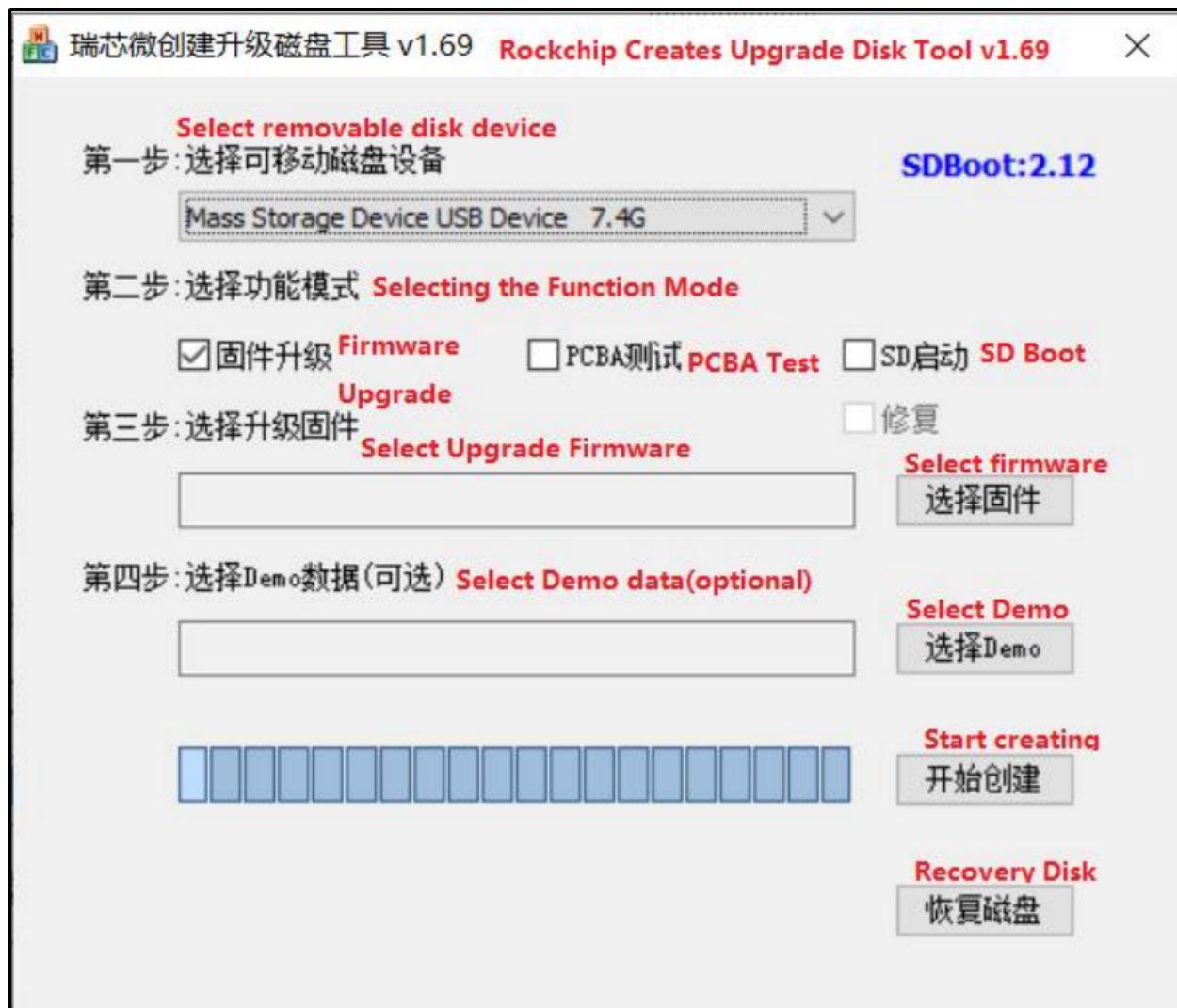


4.2 TF Card Flashing

TF card production, flashing and testing

Note: The tested TF card capacity is up to 16G, using 32G and above TF card may fail to burn.

Copy SDSDiskTool_v1.69.zip from the user profile tools directory to any directory on windows. Run SD_Firmware_Tool.exe with administrator privileges.



Select the disk device, check “Firmware Upgrade” and select update.img. Click Start Creating.





Insert the TF card into the development board and start, the system will automatically enter the flashing process. When the flashing is complete, both the screen and the serial port will prompt:

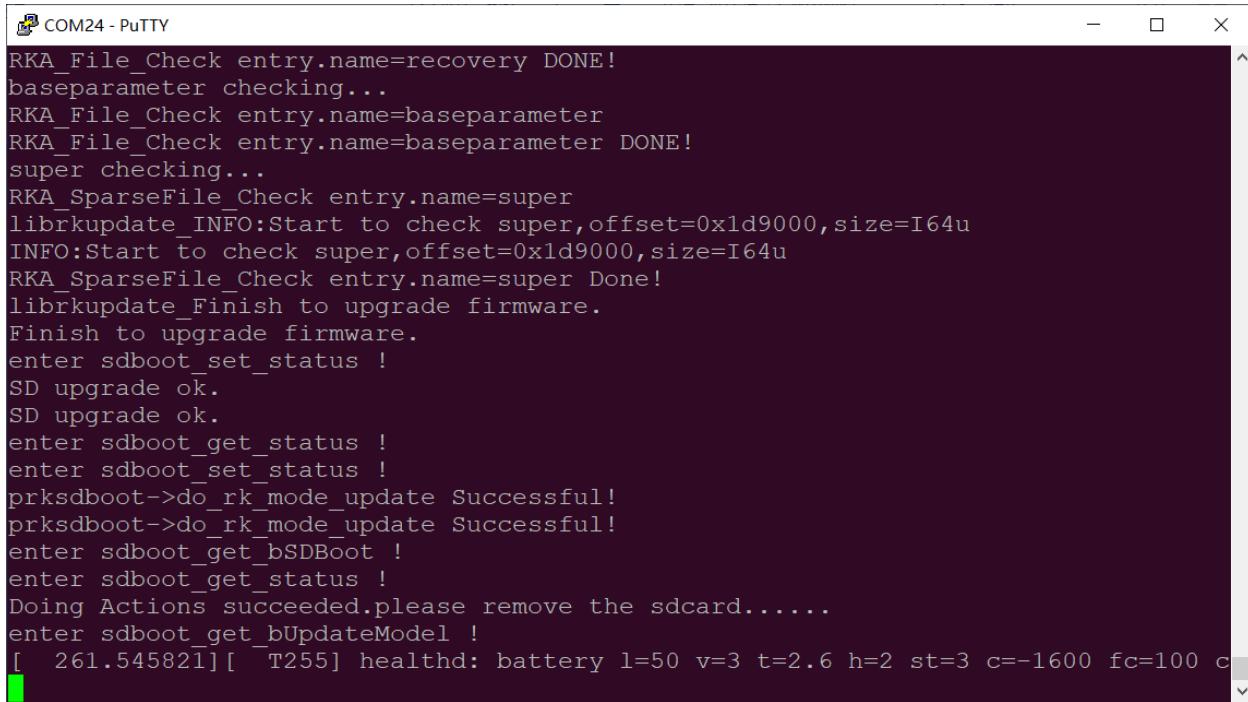
Please remove SD CARD!!!, wait for reboot.

At this time, pull out the TF card, the system automatically restarts (please do not power down directly).

During mass production, check the flashing status by SoM heartbeat light. Heartbeat light modes are as follows:

1. Kernel startup phase: Heartbeat light mode, regular intermittent flashes.
2. Flashing preparation phase: EMMC indicator light, off.
3. Flashing in progress phase: EMMC indicator light, on.
4. Flashing completion phase: Heartbeat light mode, regular intermittent flashes.

Serial port information during the burning process:



```

COM24 - PuTTY
RKA_File_Check entry.name=recovery DONE!
baseparameter checking...
RKA_File_Check entry.name=baseparameter
RKA_File_Check entry.name=baseparameter DONE!
super checking...
RKA_SparseFile_Check entry.name=super
librkupdate _INFO:Start to check super,offset=0x1d9000,size=I64u
INFO:Start to check super,offset=0x1d9000,size=I64u
RKA_SparseFile_Check entry.name=super Done!
librkupdate_Finish to upgrade firmware.
Finish to upgrade firmware.
enter sdboot_set_status !
SD upgrade ok.
SD upgrade ok.
enter sdboot_get_status !
enter sdboot_set_status !
prksdboot->do_rk_mode_update Successful!
prksdboot->do_rk_mode_update Successful!
enter sdboot_get_bSDBoot !
enter sdboot_get_status !
Doing Actions succeeded.please remove the sdcard.....
enter sdboot_get_bUpdateModel !
[ 261.545821] [ T255] healthd: battery l=50 v=3 t=2.6 h=2 st=3 c=-1600 fc=100 c

```

If the automatic restart does not occur after removing the TF card, a manual restart can also complete the burning. Please be patient during the burning process.

2.3.9 5. System OTA Upgrade Test

OTA (over the air) upgrade is a standard software upgrade method provided by Android system. It has powerful functions, and the current version of system OTA upgrade provides two methods of local complete package upgrade and network upgrade.

5.1 OTA Upgrade Package Compilation

The OTA upgrade package can be compiled by using forlinx.sh under OK3588-android-source file, please modify the android code first, then compile the OK3588 system upgrade package:

Note: Before compiling the OK3588 system upgrade package, make sure that the source code has been fully compiled using the./forlinx.sh all command.

```
forlinx@ubuntu20:~/OK3588-android-source$ ./forlinx.sh ota
```

Compilation process is too long, please be patient. Compilation completed firmware directory: rockdev/Image-ok3588_c/update.zip

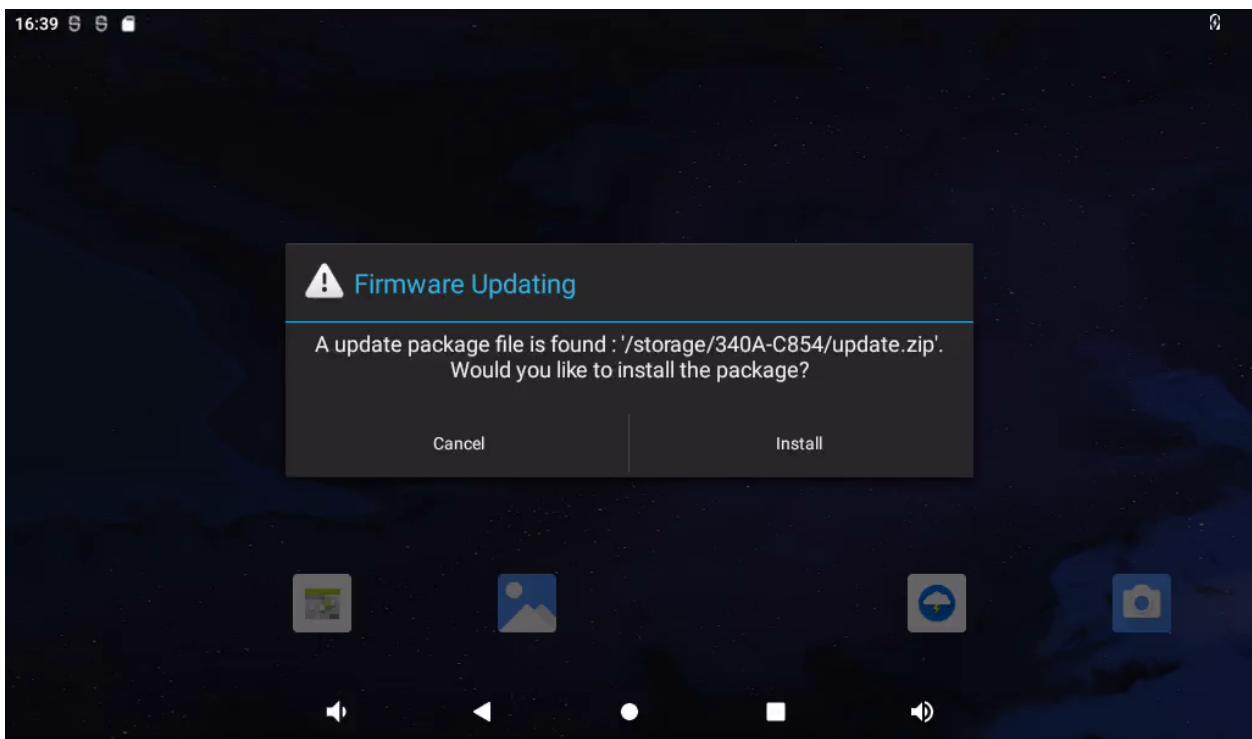
5.2 OTA Local Upgrade

Copy the update.zip generated in the previous section to the root directory of the USB or TF card, or the /data/media/0/ directory, the system will automatically detect the upgrade package and pop up the upgrade dialogue box.

The following is done in adb mode:

```
adb root
adb remount
adb push update.zip /data/media/0/
```

After waiting for a while, the interface prompts whether to install the upgrade package window.



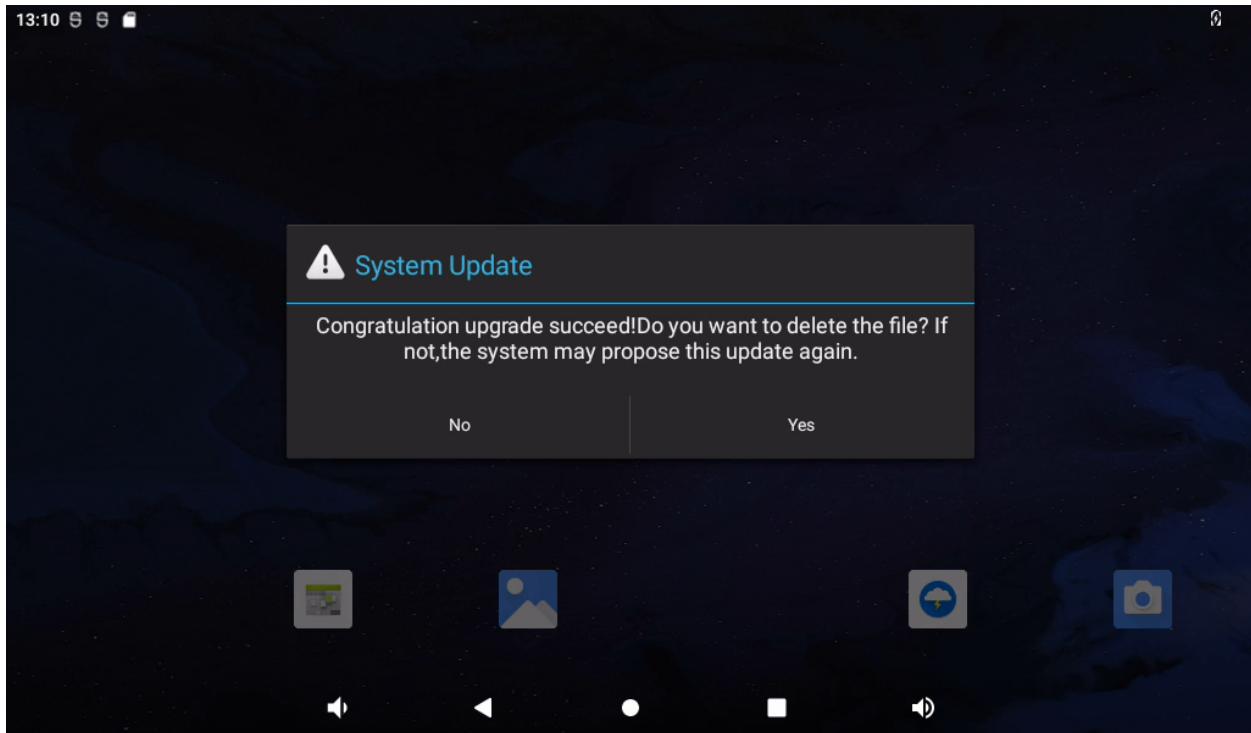
Click “Installation” .

The debugging window prints the following information:

After that, it will automatically restart and enter the Recovery system to automatically complete the OTA package upgrade. At this time, it cannot be powered off and wait for the upgrade.

When completed, it will automatically restart to the main Android interface.

After the system restarts, a dialog box pops up on the interface to prompt congratulations on the success of the upgrade.



Click “Yes” .

Finally, you can verify that the android system has been modified.

Note: The prompt for firmware upgrade operation after the system reboot is the correct prompt because the file /data/media/0/update.zip exists, so it will wait for a few minutes to pop up the prompt. The Firmware Prompt dialog will not pop up after deleting the /data/media/0/update.zip.

5.3 OTA Network Upgrade

1. Environment Setup

Edit the device/rockchip/rk3588/device.mk file and change the server IP address via ro.product.ota.host

```
PRODUCT_PROPERTY_OVERRIDES += \
-    ro.product.ota.host = 192.168.1.1:8888 \
+    ro.product.ota.host = 192.168.1.30:8888 \
        ro.vendor.sdkversion = $(CURRENT_SDK_VERSION) \
        vendor.gralloc.disable_afbc = 0
```

Execute ./for Linux.sh ota to compile the upgrade firmware, which is located at rockdev/Image-ok3588_c/update.zip.

Extract apache-tomcat-7.0.29.tar.gz to PC Ubuntu home directory, copy update.zip to /home/forlinx/apache-tomcat-7.0.29/webapps/OtaUpdater/WEB-INF/packages/ ok3588_c/1.0.0/1.0.1.zip

Enable apache-tomcat

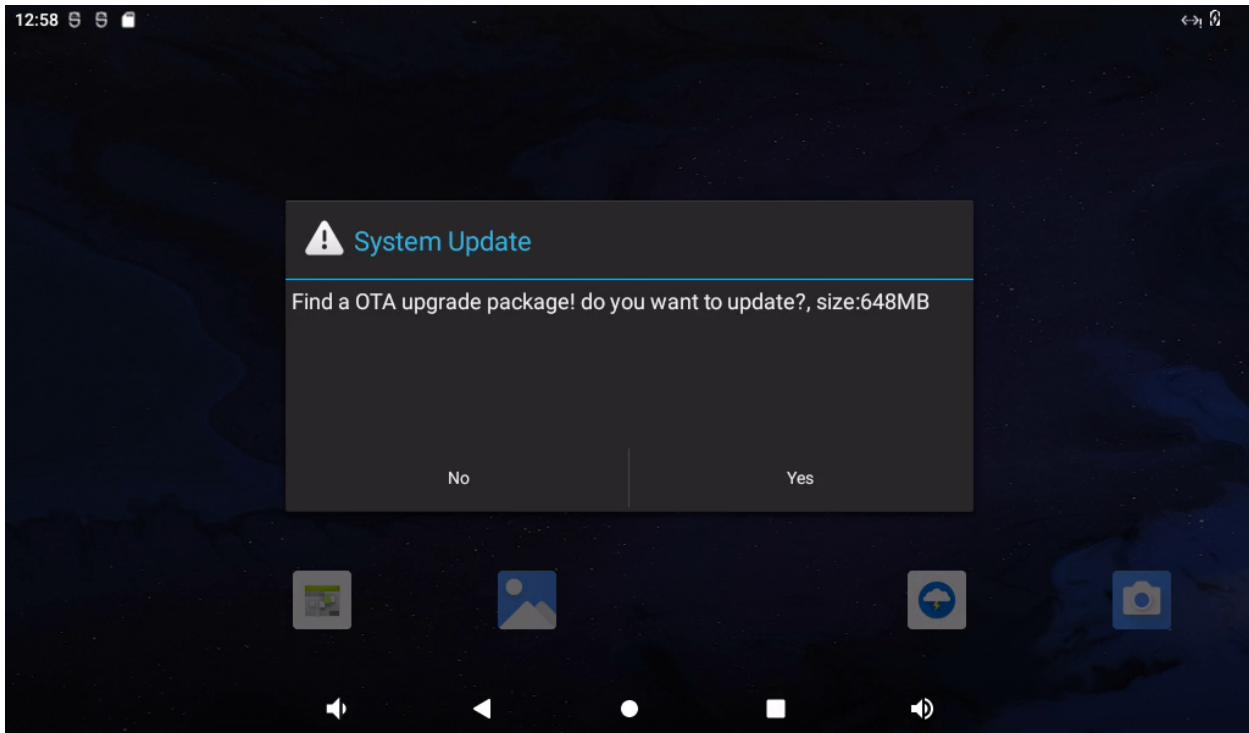
```
cd /home/forlinx/apache-tomcat-7.0.29  
./bin/startup.sh
```

Disable apache-tomcat

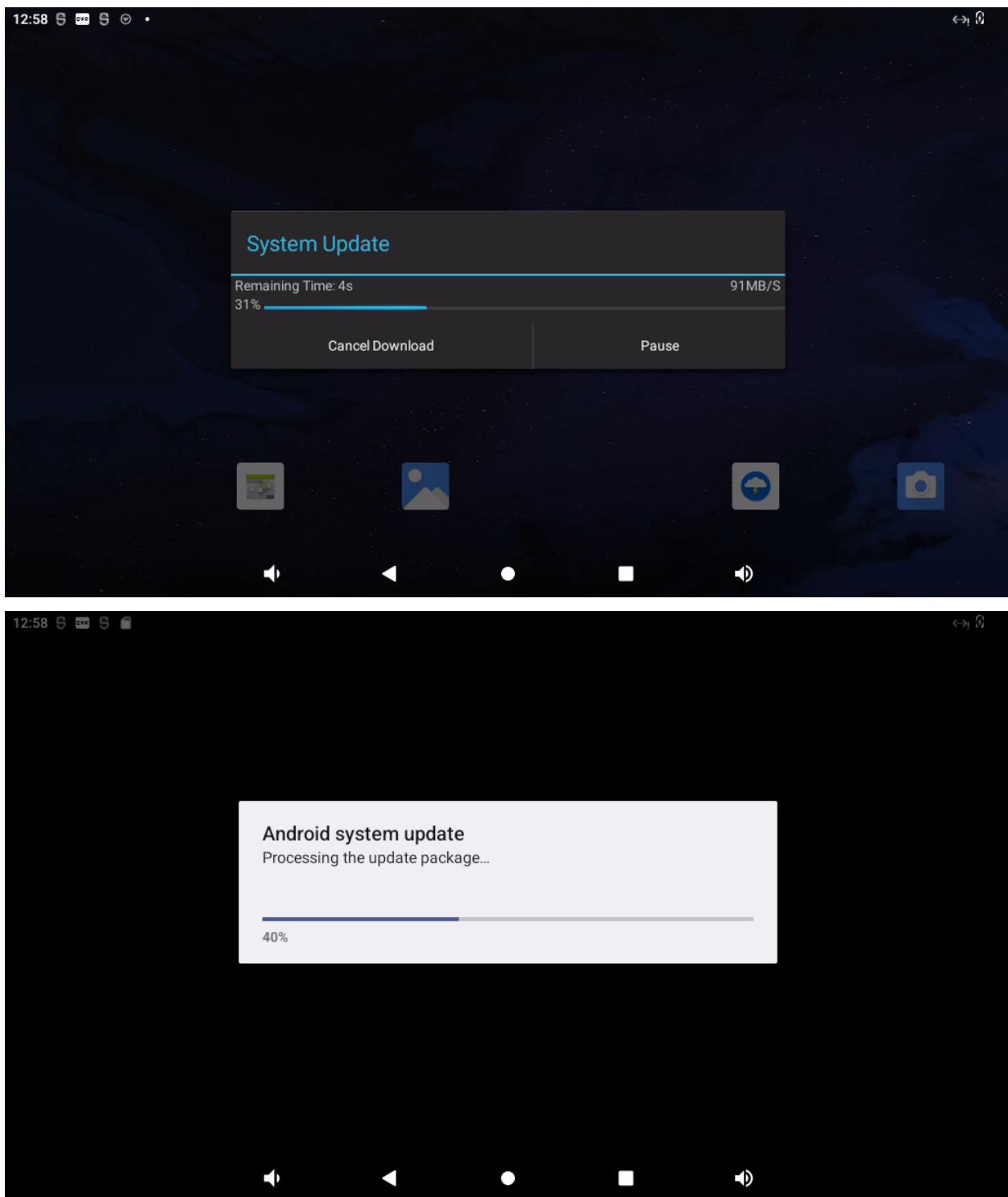
```
./bin/shutdown.sh
```

2. Network Upgrade Test

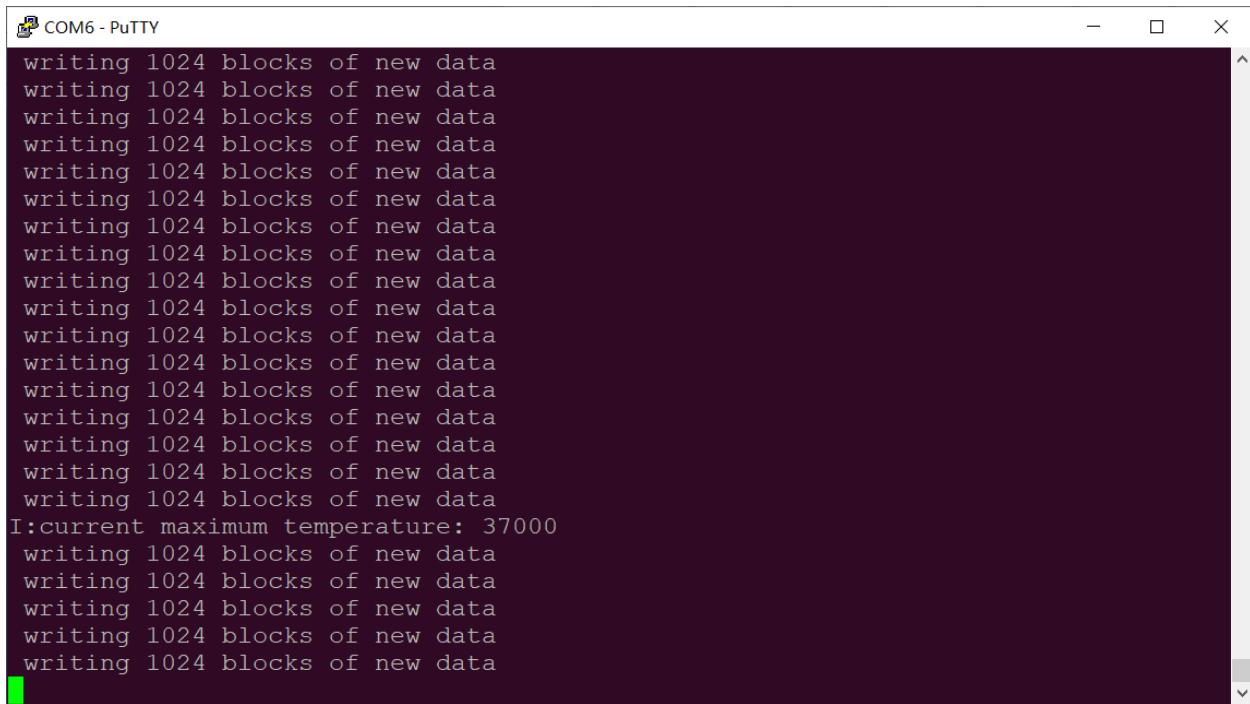
Connect network and power up OK3588 board, and the prompt dialog box will pop up to prompt system upgrading.



Click “Yes” , the upgrade 1.0.1.zip file will be downloaded via http protocol.

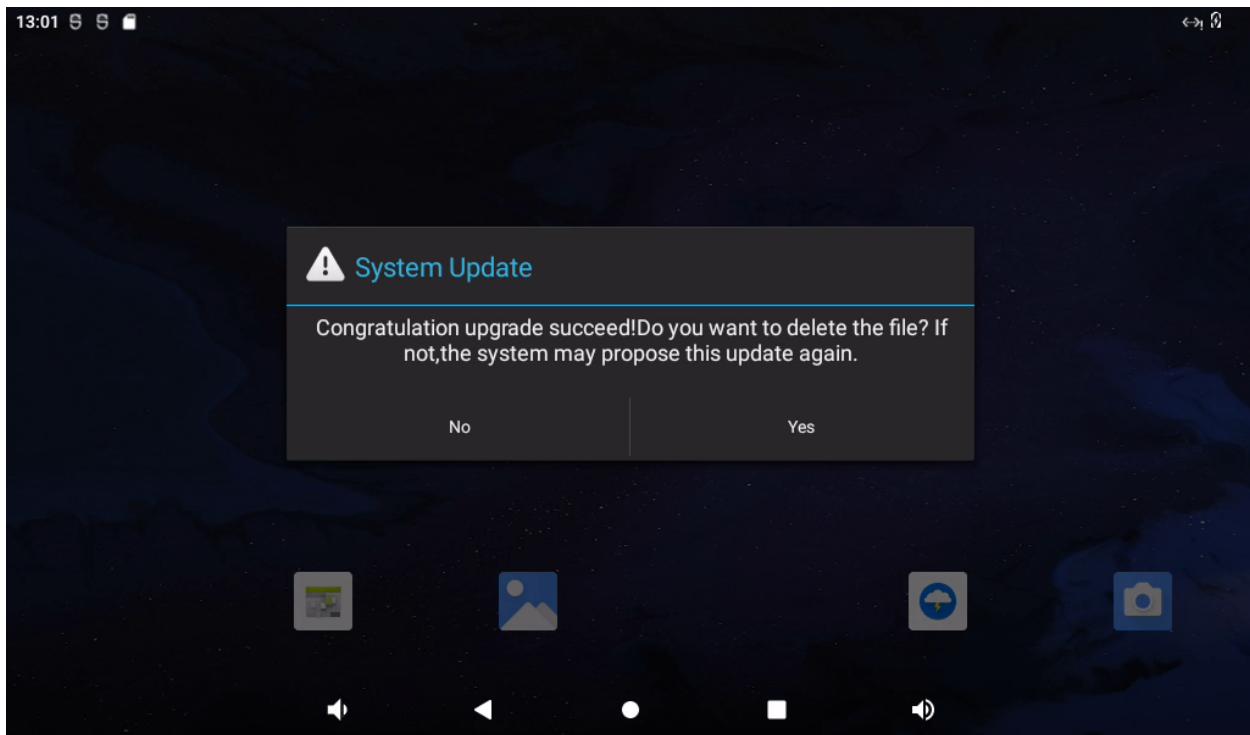


The system will restart automatically, and the serial port terminal will print as follows:



```
COM6 - PuTTY
writing 1024 blocks of new data
I:current maximum temperature: 37000
writing 1024 blocks of new data
```

After this reboot, the android interface prompts that the upgrade is complete.



Click “Yes” .

Note: If prompted to upgrade again, click “No” , because OK3588 can get the remote 1.0.1.zip upgrade package through the network, so it will remind you whether you need to upgrade or not.

2.4 Android 12.0 User's Compilation Manual_V 1.0

Document classification: Top secret Secret Internal information [Open](#)

2.4.1 Copyright Notice

The copyright of this manual belongs to Baoding Folinx Embedded Technology Co., Ltd. Without the written permission of our company, no organizations or individuals have the right to copy, distribute, or reproduce any part of this manual in any form, and violators will be held legally responsible.

Forlinx adheres to copyrights of all graphics and texts used in all publications in original or license-free forms.

The drivers and utilities used for the components are subject to the copyrights of the respective manufacturers. The license conditions of the respective manufacturer are to be adhered to. Related license expenses for the operating system and applications should be calculated/declared separately by the related party or its representatives.

2.4.2 Overview

This manual is designed to enable users of the Forlinx Embedded development board to quickly understand the compilation process of the products and familiarize themselves with the compilation methods of Forlinx products. The application needs to be cross-compiled on an Ubuntu host before it can run on the development board. By following the methods provided in the compilation manual and performing practical operations, you will be able to successfully compile your own software code.

The manual will provide instructions for setting up the environment but there may be some unforeseen issues during the environment setup process. For beginners, it is recommended to use the pre-configured development environment provided by us. This will allow you to quickly get started and reduce development time.

Linux systems are typically installed in three ways: Dual system on a real machine, single system on a real machine, and virtual machine. Different installation methods have their advantages and disadvantages. This manual only provides methods to build Ubuntu in a virtual machine.

Hardware Requirements: It is recommended to have at least 16GB memory or above. It allows for allocating a sufficient memory to the virtual machine (recommended to allocate 10GB or above), while still leaving enough resources for other operations on Windows. Insufficient memory allocation may result in slower performance on Windows.

The manual is mainly divided into five chapters:

- Chapter 1. Virtual Machine software installation - introduction to downloading and installing Vmware software;
- Chapter 2. provides the loading of the Ubuntu system;
- Chapter 3. Building, setting up, and installing necessary tools for the Ubuntu system and common issues in development environments;
- Chapter 4. mainly focuses on the deployment and usage of Docker containers, kernel compilation, as well as Android-related source code;
- Chapter 5. is the description of android application development.

A description of some of the symbols and formats in the manual:

Format	Meaning
Note	Note or information that requires special attention, be sure to read carefully
?	Relevant notes on the test chapters
?	Indicates the related path.
Blue font on gray background	Refers to commands entered at the command line(Manual input required).
Black font	Serial port output message after entering a command
Bold black	Key information in the serial port output message
//	Interpretation of input instructions or output information
User-name@Hostname	forlinx @ ubuntu: Development environment ubuntu account information, which can be used to determine the environment in which the function operates.

Example: Search the Docker-CE version:

```
forlinx@ubuntu:~$ apt-cache madison docker-ce          //查找 Docker-CE 的版本
docker-ce | 18.06.3~ce~3-0~ubuntu | http://mirrors.aliyun.com/docker-ce/linux/ubuntu/
→trusty/stable amd64 Packages
```

- forlinx@ubuntu: the username is forlinx and the hostname is ubuntu, indicating that the user forlinx is used on the development environment ubuntu for operations.
- //: Explanation of the instruction, no input required.
- apt-cache madison docker-ce: Blue font on a gray background, indicating the relevant commands that need to be manually entered.
- docker-ce | 18.06.3~ce~3-0~ubuntu: The black font with gray background is the output information after the input command, and the bold font is the key information.

2.4.3 Application Scope

This manual is mainly applicable to the Android12.0 operating system on the Forlinx OK3588-C platform. Other platforms can also refer to it, but there will be differences between different platforms. Please make modifications according to the actual conditions.

2.4.4 Revision History

Date	User Manual Version	Revision History
04/11/2022	V1.0	OK3588-C Android 12.0 User's Compilation Manual Initial Version

2.4.5 1. VMware Virtual Machine Software Installation

This chapter mainly introduces the installation of VMware virtual machine, and takes VMware workstation 16 Pro v16.2.3 as an example to show the installation and configuration process of the operating system.

1.1 VMware Software Downloads and Purchase

Go to the VMware website <https://www.vmware.com/cn.html> to downloadWorkstation Pro and get the product key. VMware is a paid software that requires purchasing, or you can choose to use a trial version.

The screenshot shows the VMware website's main navigation bar at the top. Below it, the 'Resources' section is expanded, displaying a grid of links categorized into three columns: Tools & Training, Support, and Blogs & Communities. Each category contains several specific links such as 'VMware Customer Connect', 'Support Offerings', 'Skyline', etc.

Category	Link 1	Link 2	Link 3
Tools & Training	VMware Customer Connect	Support Offerings	Blogs & Communities
	Learning & Certification	Skyline	Blogs
	Tech Zone	Support Customer Welcome Center	Podcasts
	Product Downloads	Marketplace	News & Stories
	Product Trials	Cloud Marketplace	Communities
	Cloud Services Engagement Platform	VMware Marketplace	Customers
	Hands-on Labs	Videos	Customer Stories
Services	Professional Services	VMware Video Library	Events
	Customer Success	VMware Explore Video Library	VMware Explore
			SpringOne
			All Events & Webcasts

[SEE ALL RESOURCES](#)

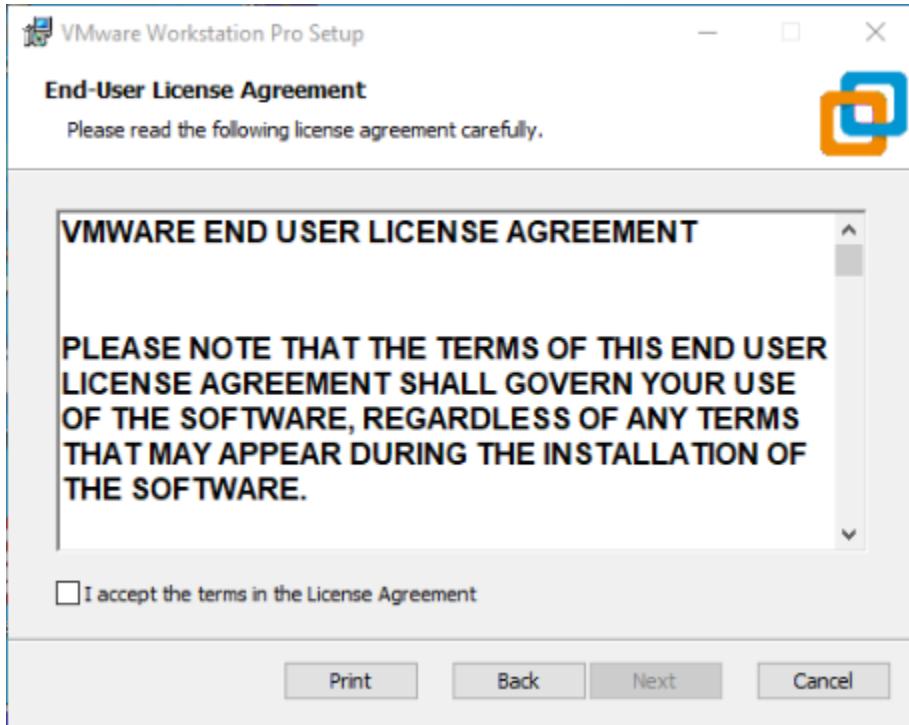
After the download is complete, double-click the startup file to start the installer.

1.2 VMware Software Installation

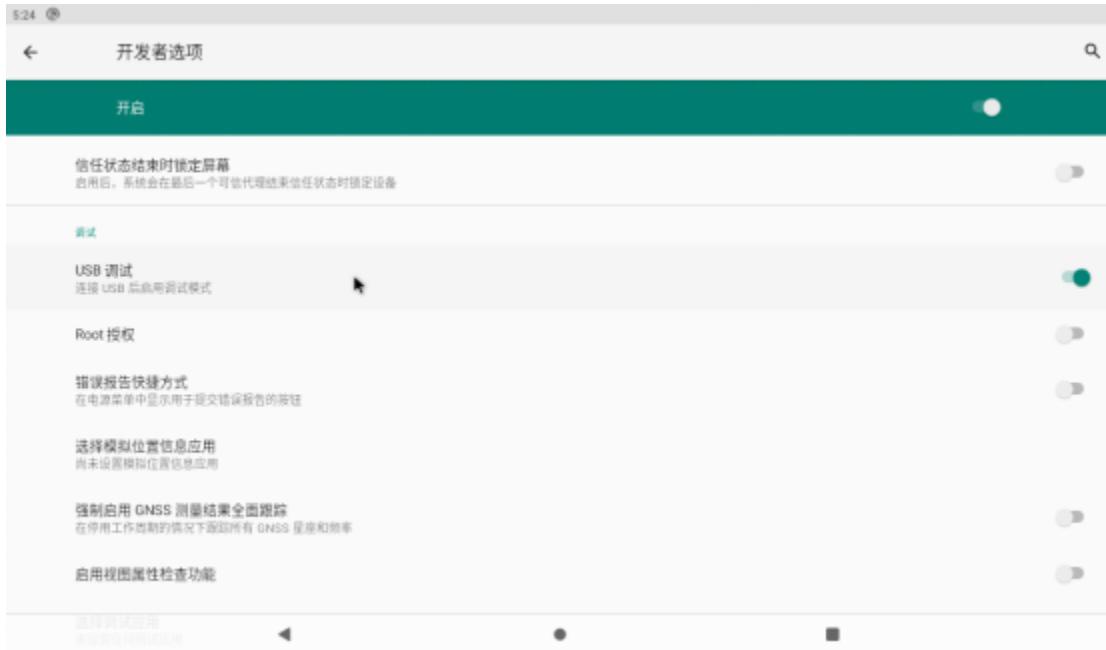
Double-click the startup program to enter the installation wizard, and click on “Next” .



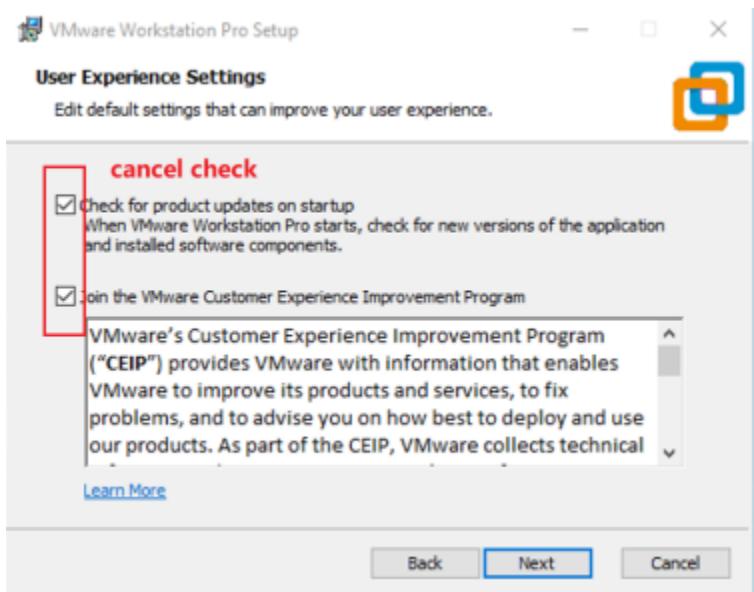
Check I accept the terms in the license agreement and click Next.



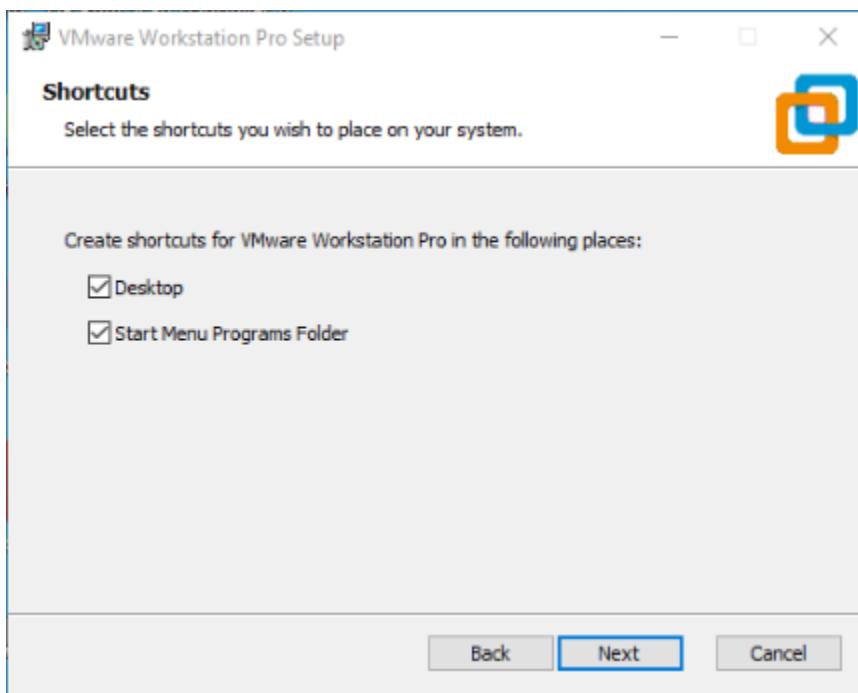
Modify the installation location to the partition of your computer where the software is installed, and click “Next” .



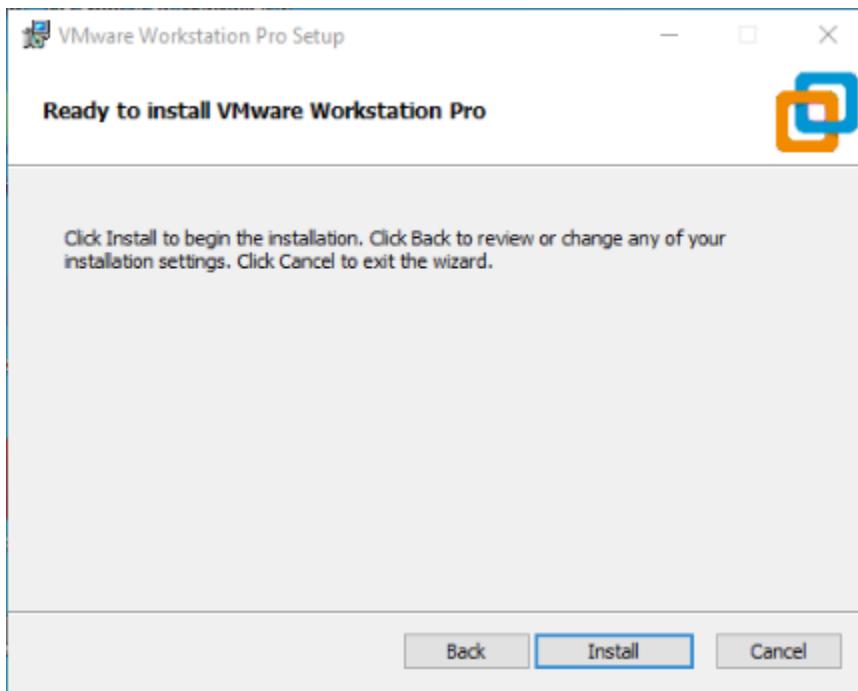
Uncheck and click on “Next” .



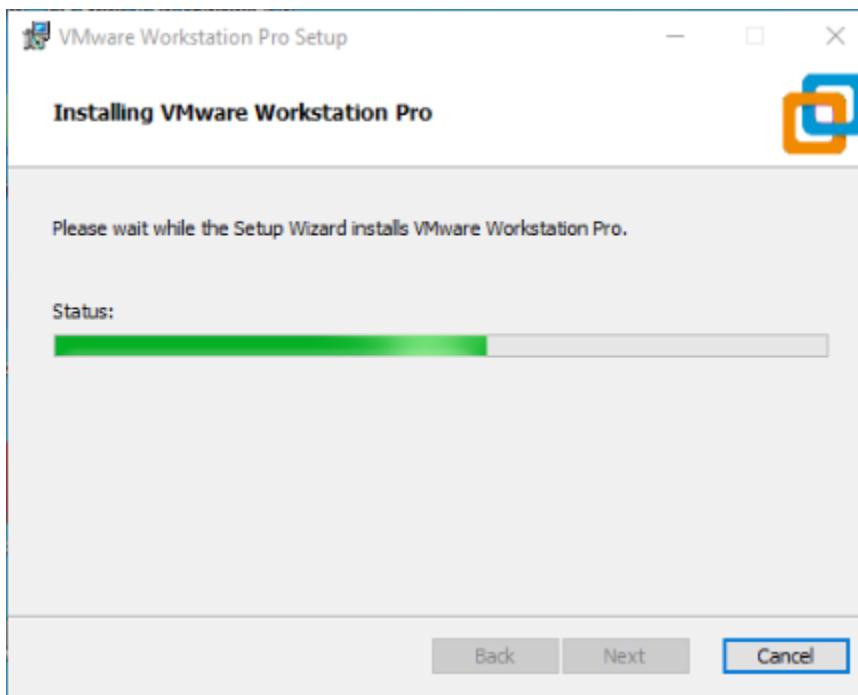
Check Add Shortcut and click “Next” .



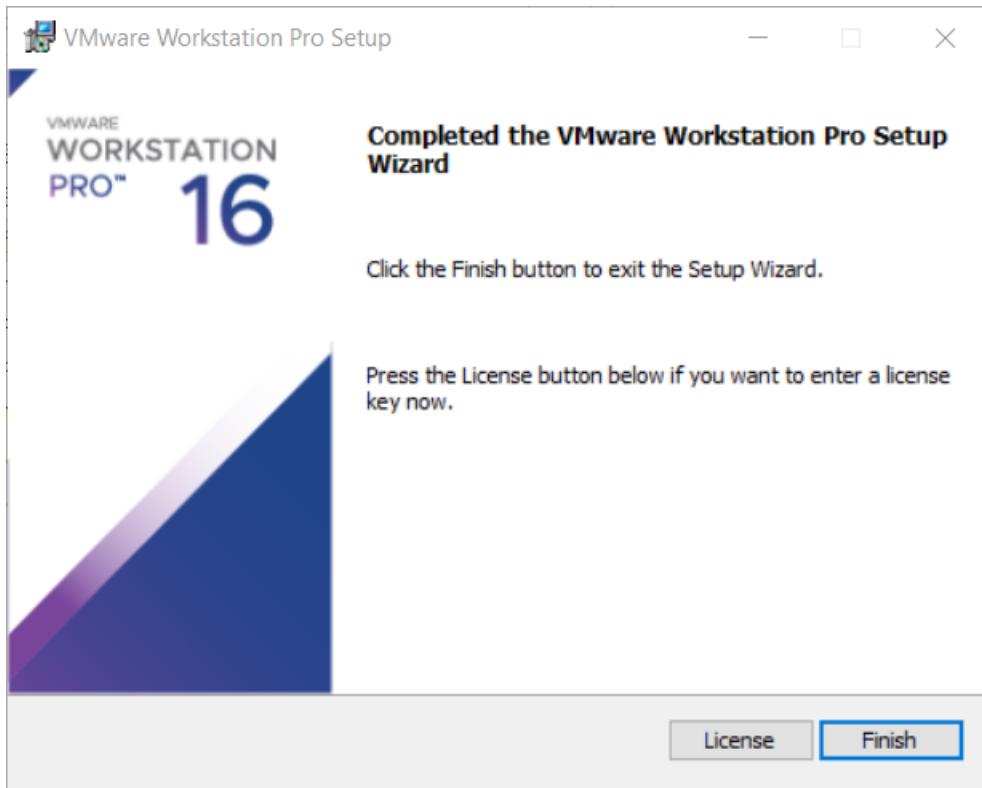
Click “Install”



Wait for the installation to complete.



Click “Finish” to try it out. If users need to use it for a long time, they need to buy it from the official and fill in the license.



2.4.6 2. Loading the Existing Ubuntu Development Environment

Note:

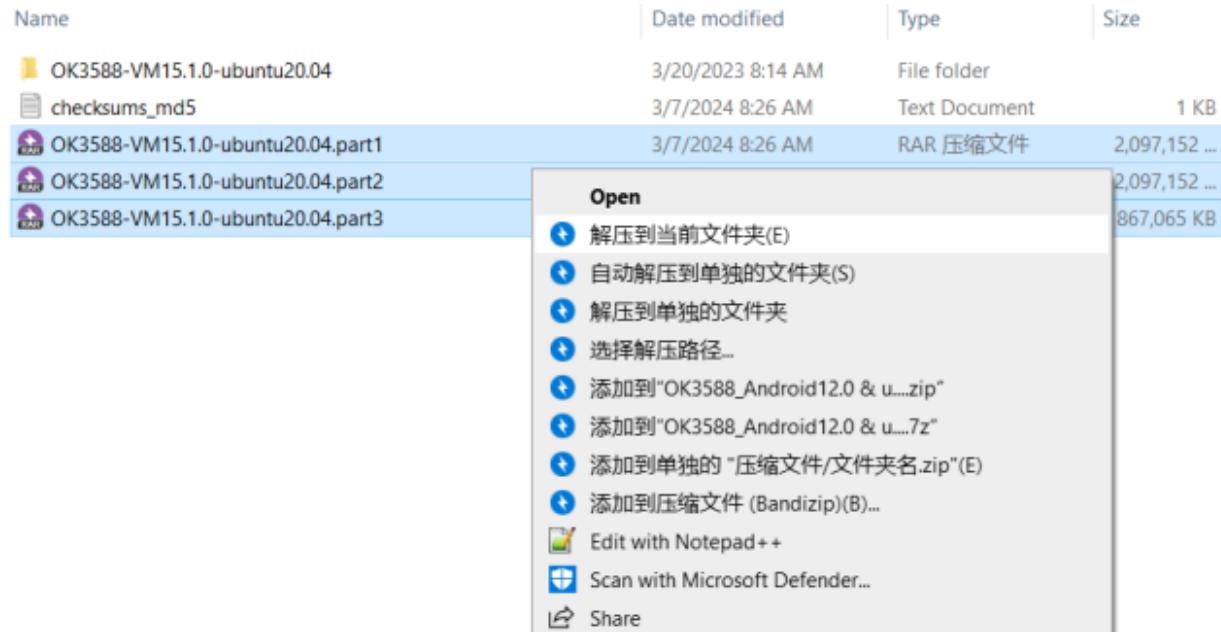
- It is recommended for beginners to directly use the pre-built virtual machine environment provided by Forlinx, which already includes installed cross-compiler and Qt environment. After understanding this chapter, you can directly jump to the compilation chapter for further study;
- The development environment provided for general users is: **forlinx (username), forlinx (password)**;
- Please ask your sales representative for the download link.

There are two ways to use a virtual machine environment in VMware: one is to directly load an existing environment, and the other is to create a new environment. First talk about how to load an existing environment.

First, download the development environment provided by Forlinx. In the development environment documentation, there should be an MD5 checksum file. After downloading the development environment, you should verify the integrity of the compressed package using the MD5 checksum. (You can use an on-line MD5 checksum tool or download a specific MD5 checksum tool for this purpose). To check if the checksum in the verification file matches the checksum of the file itself. If they match, the file download is successful. If they don't match, it suggests that the file may be corrupt, and you should consider downloading it again.

Name	Date modified	Type	Size
checksums_md5	2024/3/7 8:26	Text Document	1 KB
OK3588-VM15.1.0-ubuntu20.04.part1	2024/3/7 8:26	RAR 压缩文件	2,097,152 ...
OK3588-VM15.1.0-ubuntu20.04.part2	2024/3/7 8:26	RAR 压缩文件	2,097,152 ...
OK3588-VM15.1.0-ubuntu20.04.part3	2024/3/7 8:26	RAR 压缩文件	867,065 KB

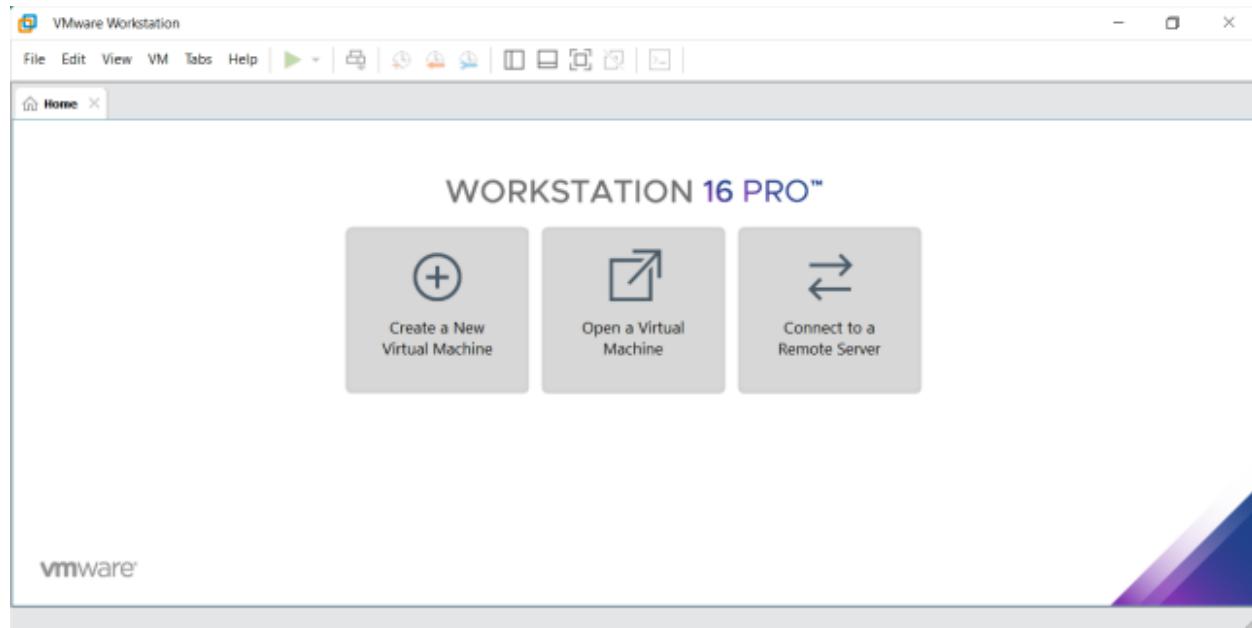
Select all compressed files, right-click and extract to the current folder or your own directory:



After the extraction is complete, you will obtain the development environment OK3588-VM16.2.3-ubuntu20.04.

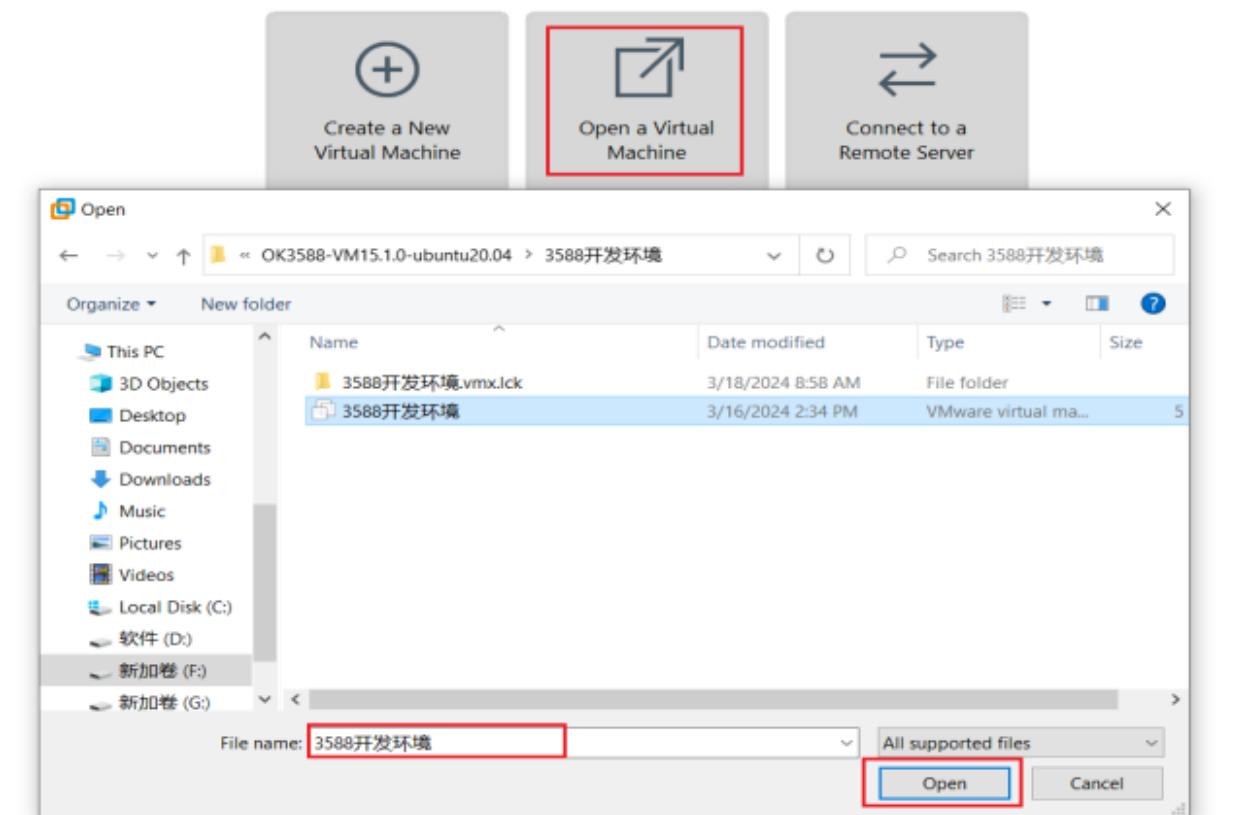
The file “3588 development environment.vmx” in the OK3588-VM16.2.3-ubuntu20.04 folder is the file that you need to open to access the virtual machine.

Open the installed virtual machine.

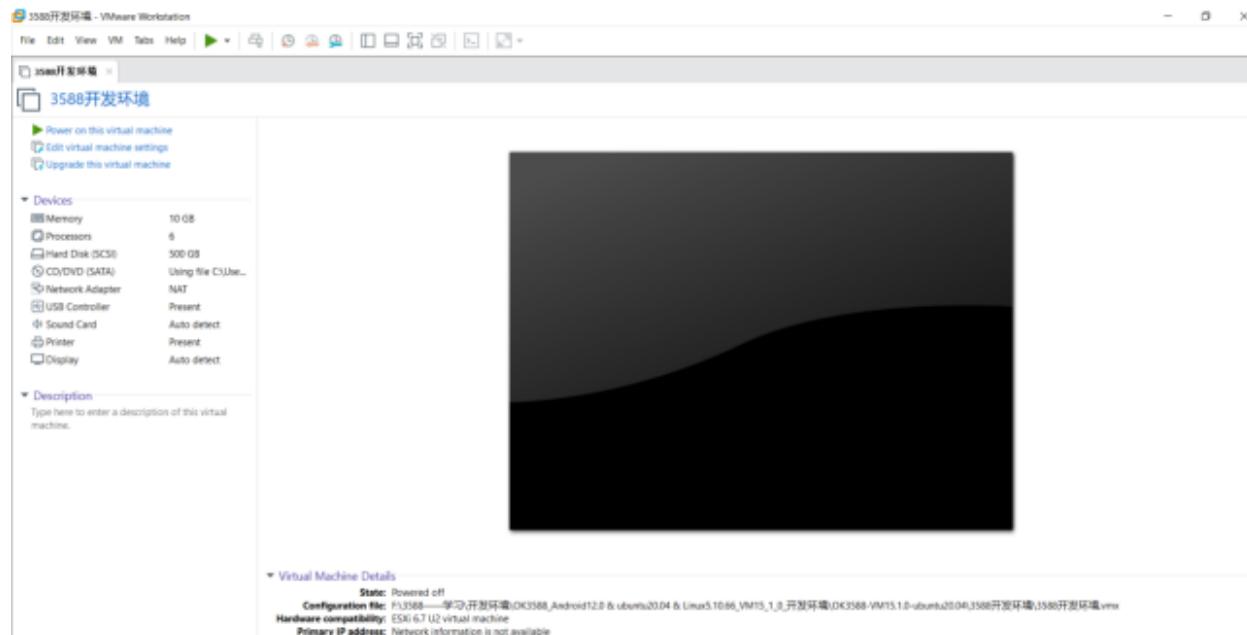


Navigate to the directory where the recently extracted OK3588-VM16.2.3-ubuntu20.04 virtual machine file is located, and double-click on the startup file to open it.

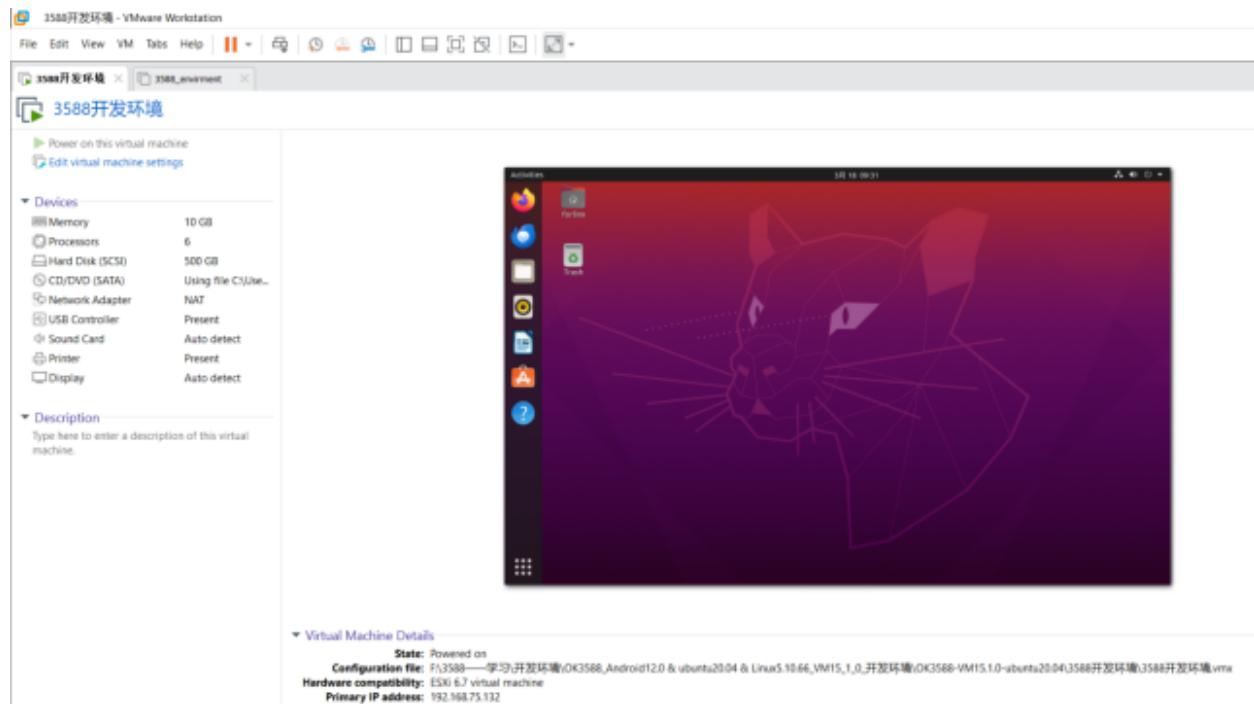
WORKSTATION 16 PRO™



Turn on this virtual machine after loading is complete to run it and enter the system's interface.



The default login account for automatic login in the development environment is “forlinx” .



2.4.7 3. New Ubuntu Development Environment Setup

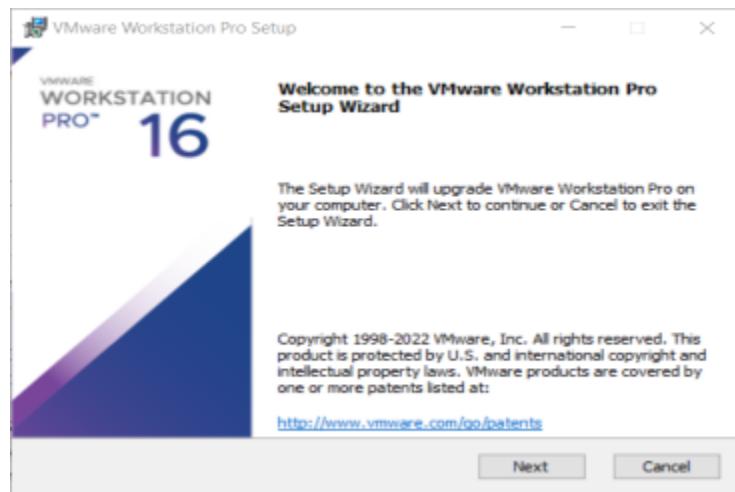
Note: Beginners are not recommended to set up a system on their own. It is recommended to use an existing virtual machine environment. If you do not need to set up the environment, you can skip this section.

This chapter mainly explains the building process of Ubuntu system.

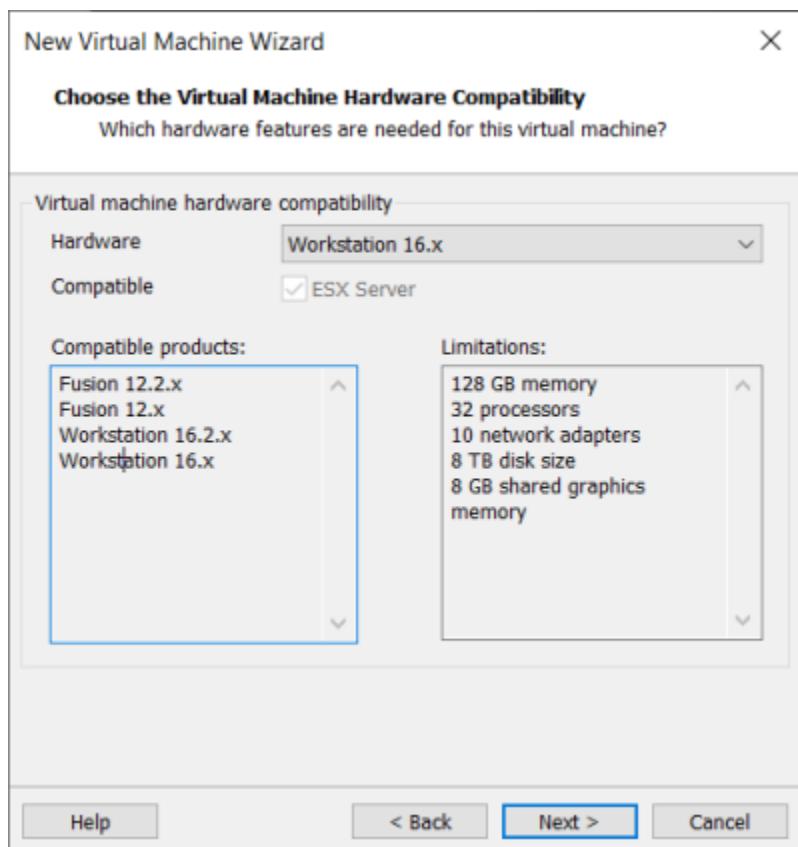
3.1 Ubuntu System Setup

3.1.1 Ubuntu Virtual Machine Setup

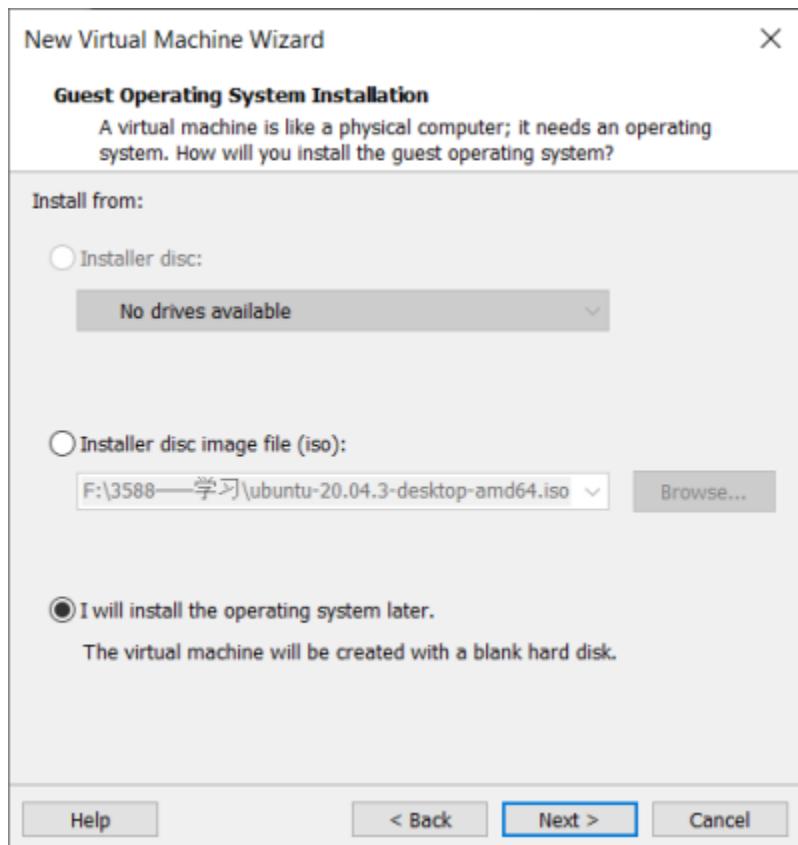
Step 1: Open the VMware software and click [File]/ [New Virtual Machine]. Enter the following interface:



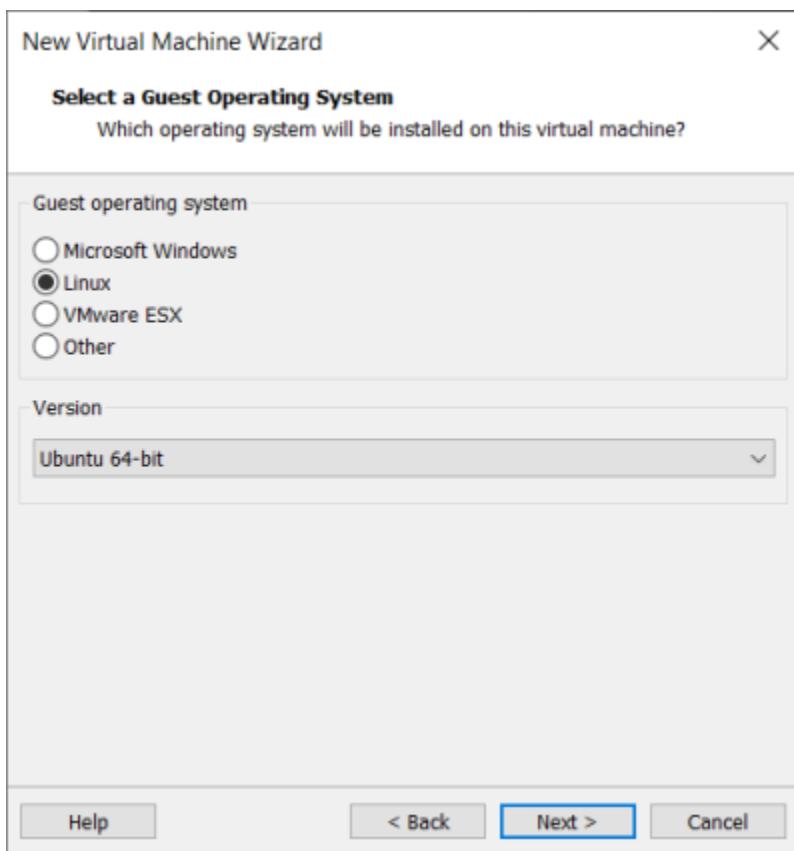
Step 2: Select Custom and click “Next” .



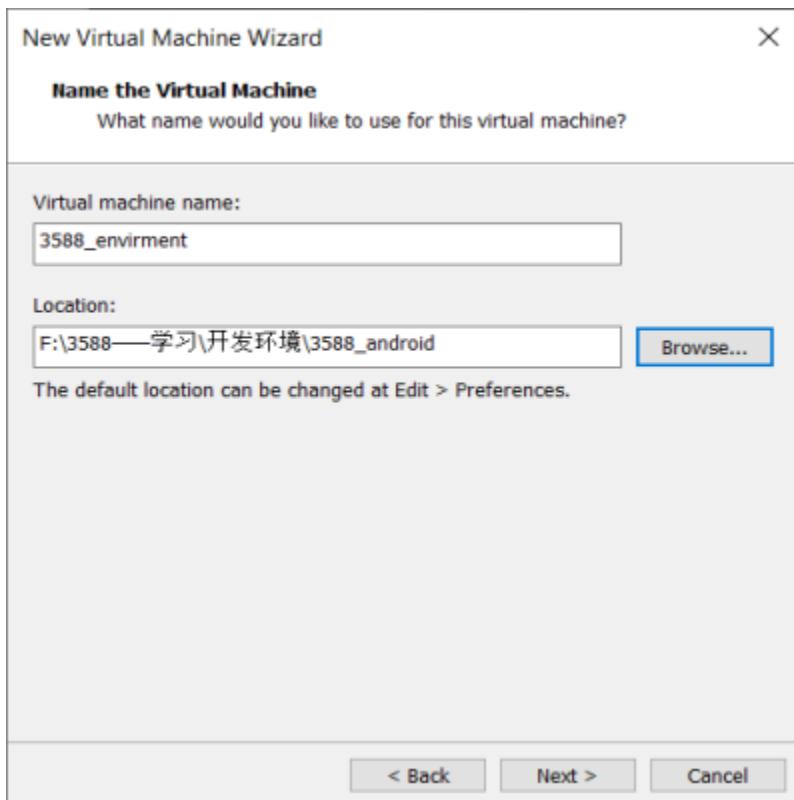
Select the compatibility with the corresponding version of VMware, which can be found in Help->About VMware Workstation, and click “Next” .



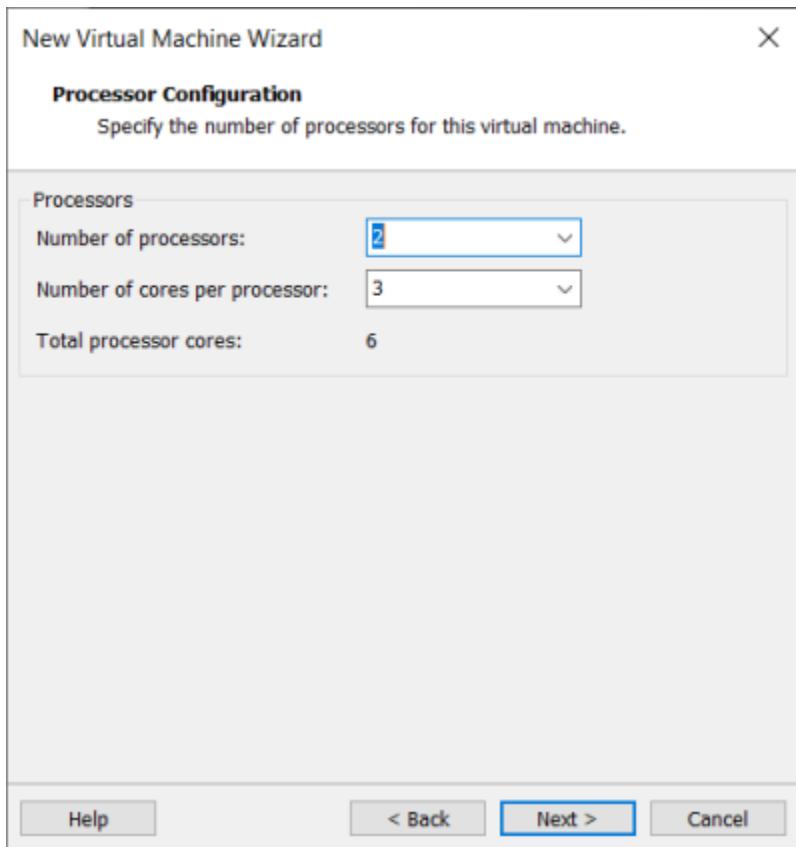
Select Install the operating system later and click “Next” .



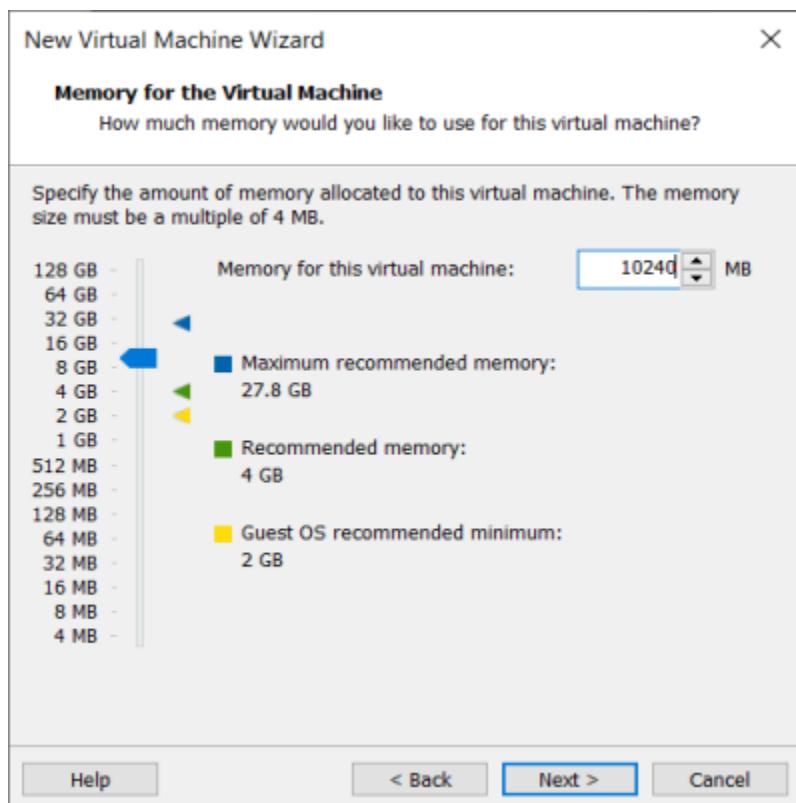
Keep the default settings and click “Next” .



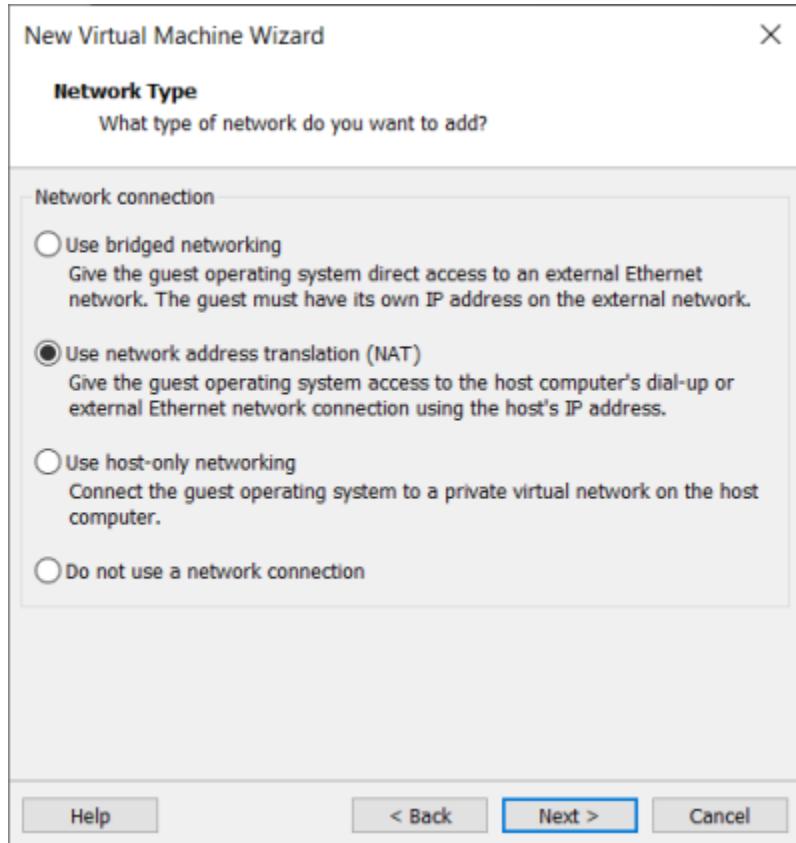
Modify the name and installation location of your virtual machine, and click “Next” .



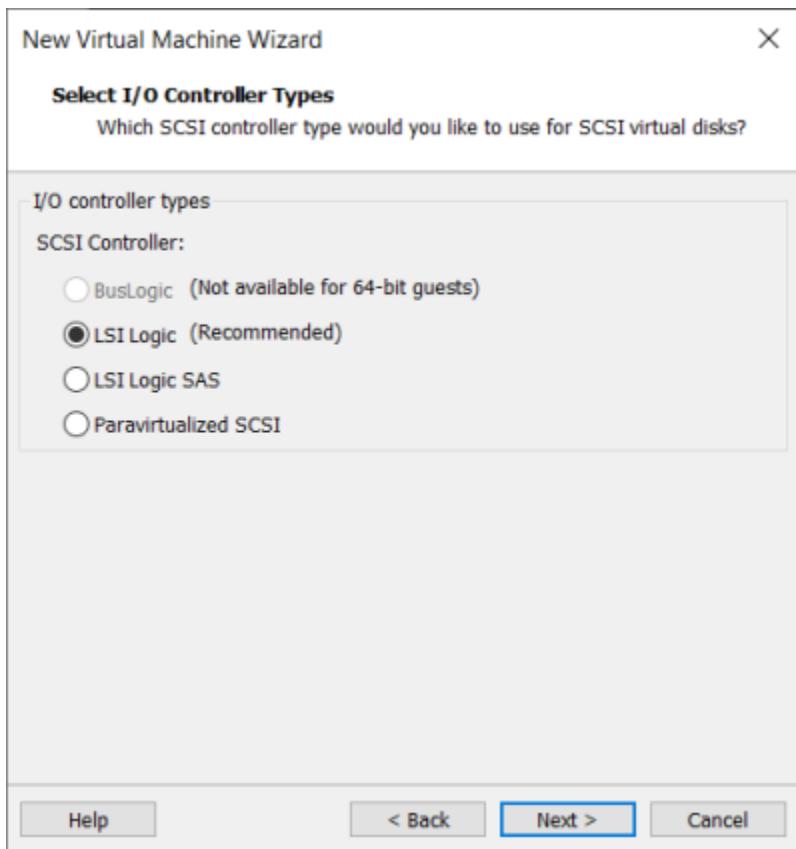
Configure the number of CPU based on your computer’s actual specifications.



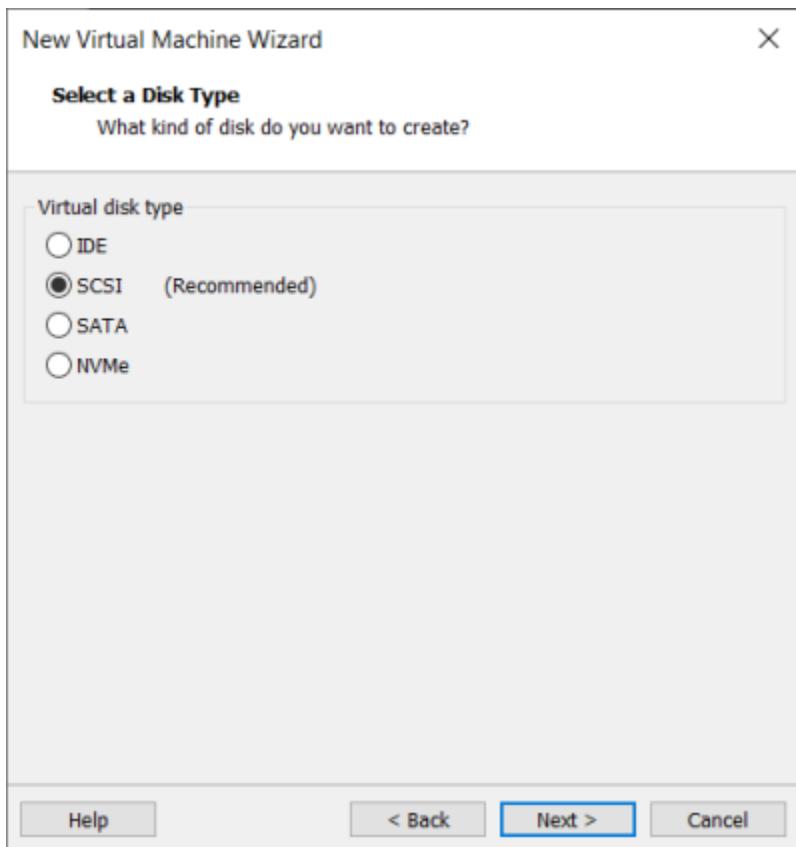
The memory size is also set according to the actual situation, and 16 gigabytes is recommended for Android 12 compilation.



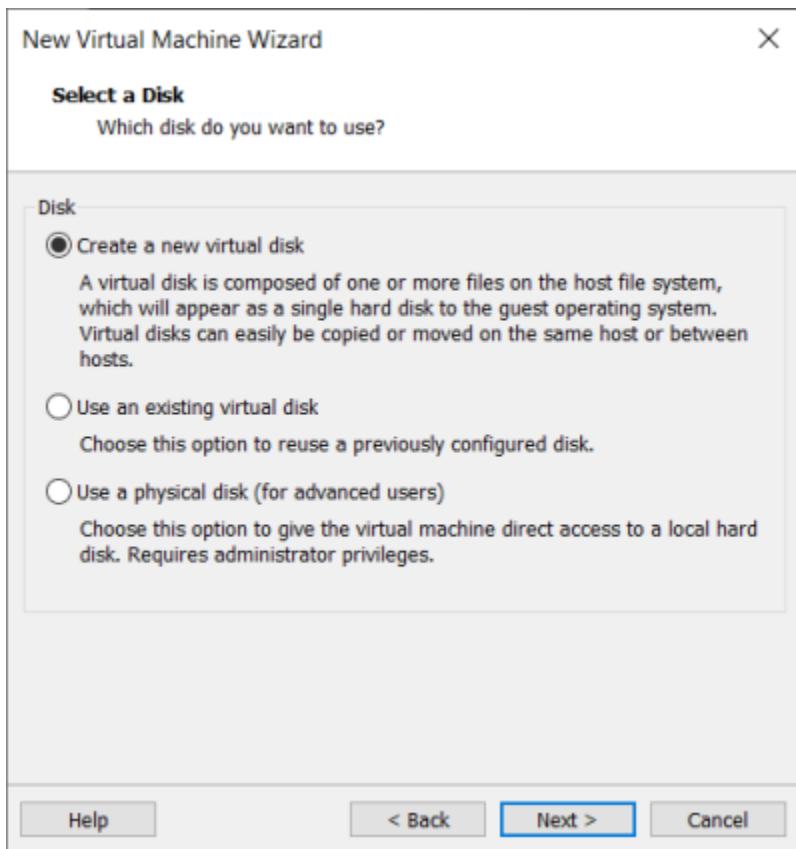
Set the network type, default to NAT mode then click “Next” . Keep the default values for the remaining steps until you reach the step to specify the disk capacity.



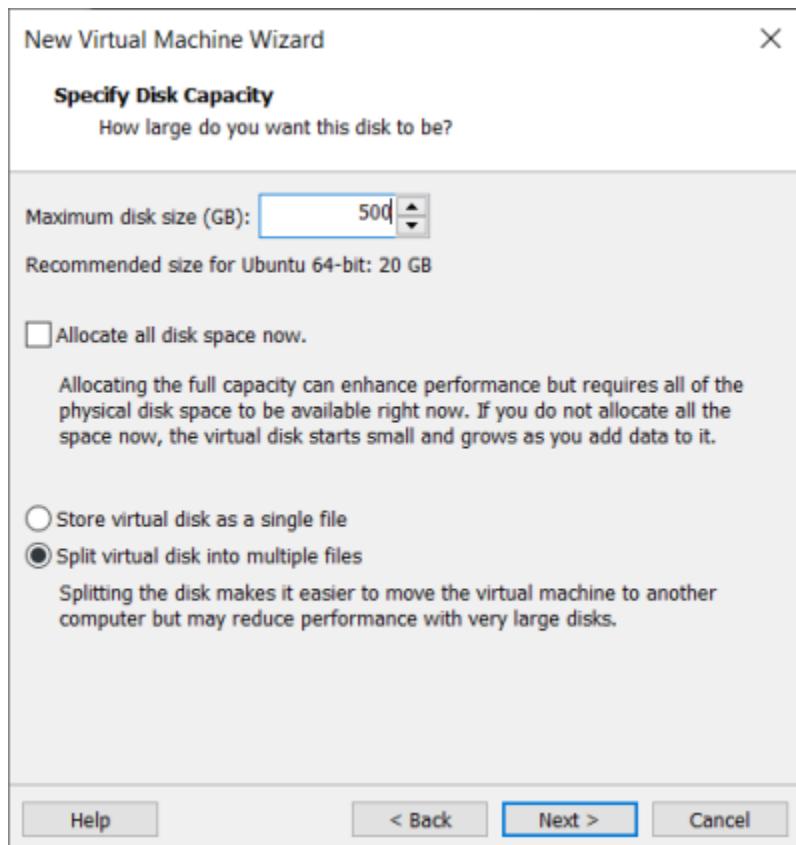
The default selection for the IO controller type here is LSI



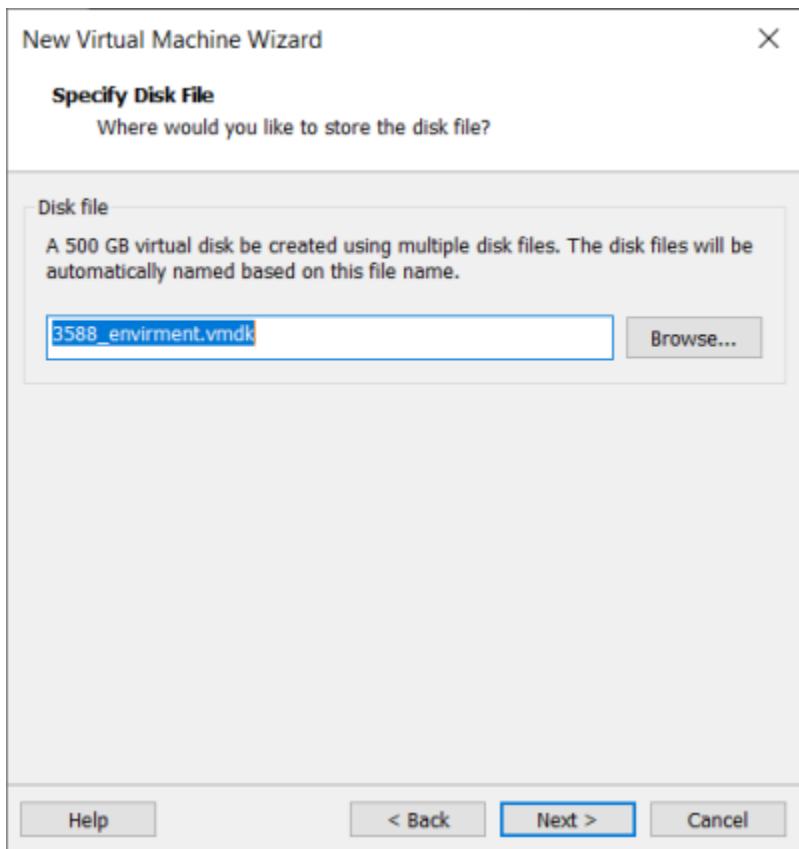
The default selection here is also SCSI.



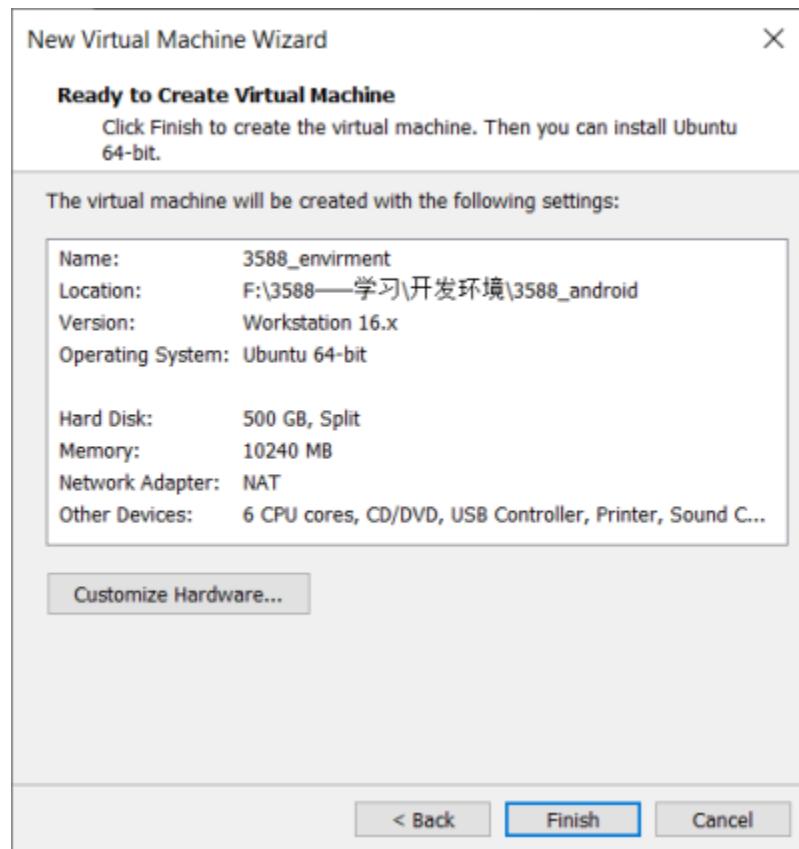
Choose to create a new virtual disk here.



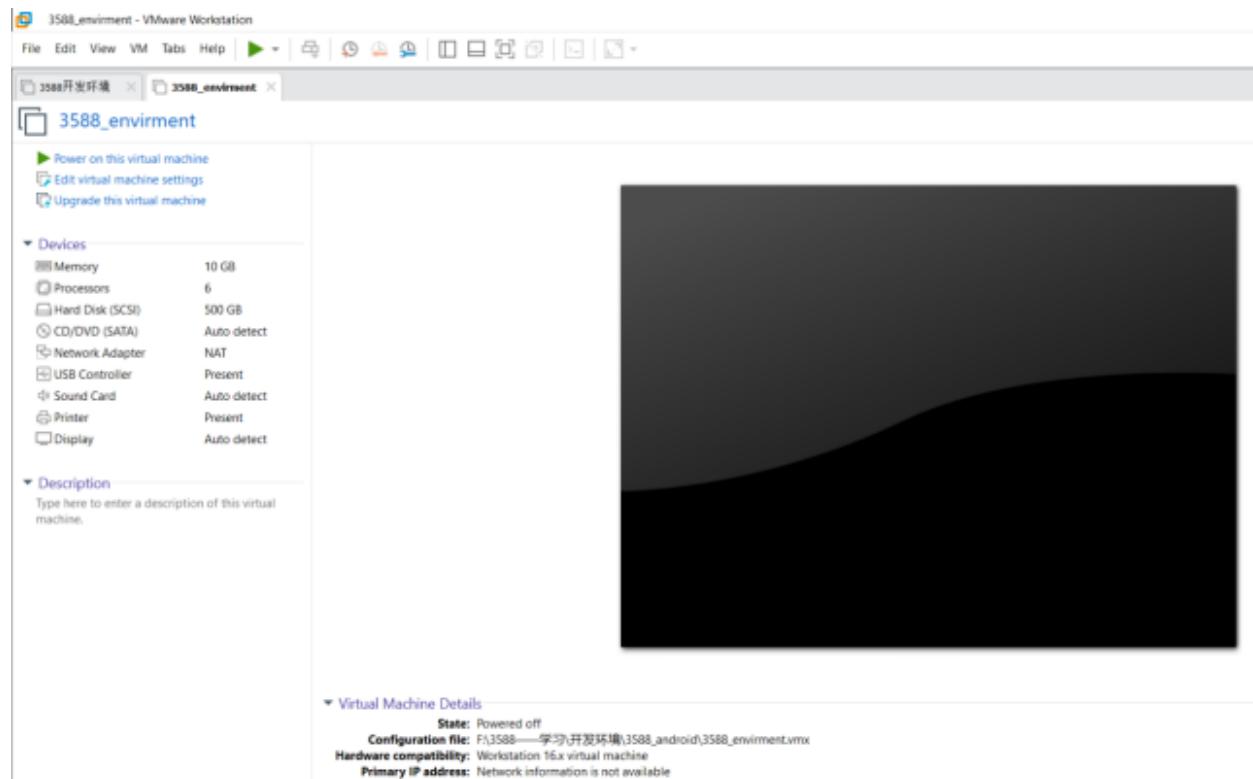
Set the disk size to 500G, select the disk provisioning format, and then click “Next” .



Specify the disk file, the default one here is fine.



Click “Finish” by default.



The virtual machine creation is now complete.

In the next section, we will introduce the installation of Ubuntu system in the virtual machine, which is similar to the installation method in the real machine. Here we describe the method of installing Ubuntu system in a virtual machine.

3.1.2 System Installation

In the previous section, a virtual machine was created, but no operating system was installed, so the virtual machine still cannot be started. Next, the Ubuntu operating system will be installed in the newly created virtual machine.

Step 1: Go to the Ubuntu official website to obtain the Ubuntu20.04 64. The download address is:

<https://old-releases.ubuntu.com/releases/20.04.3/>

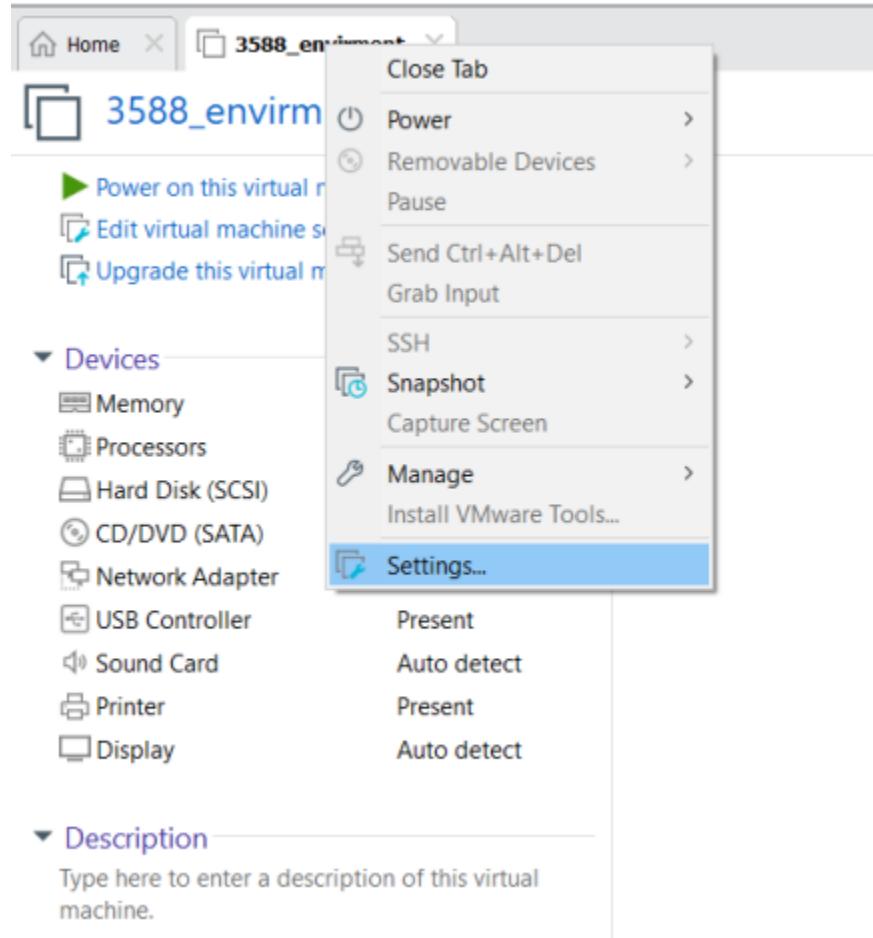
Because the source code is compiled and verified on the 20.04, select and install it. These operations may vary slightly between Ubuntu system versions.

Download “ubuntu-20.04.3-desktop-amd64.iso”

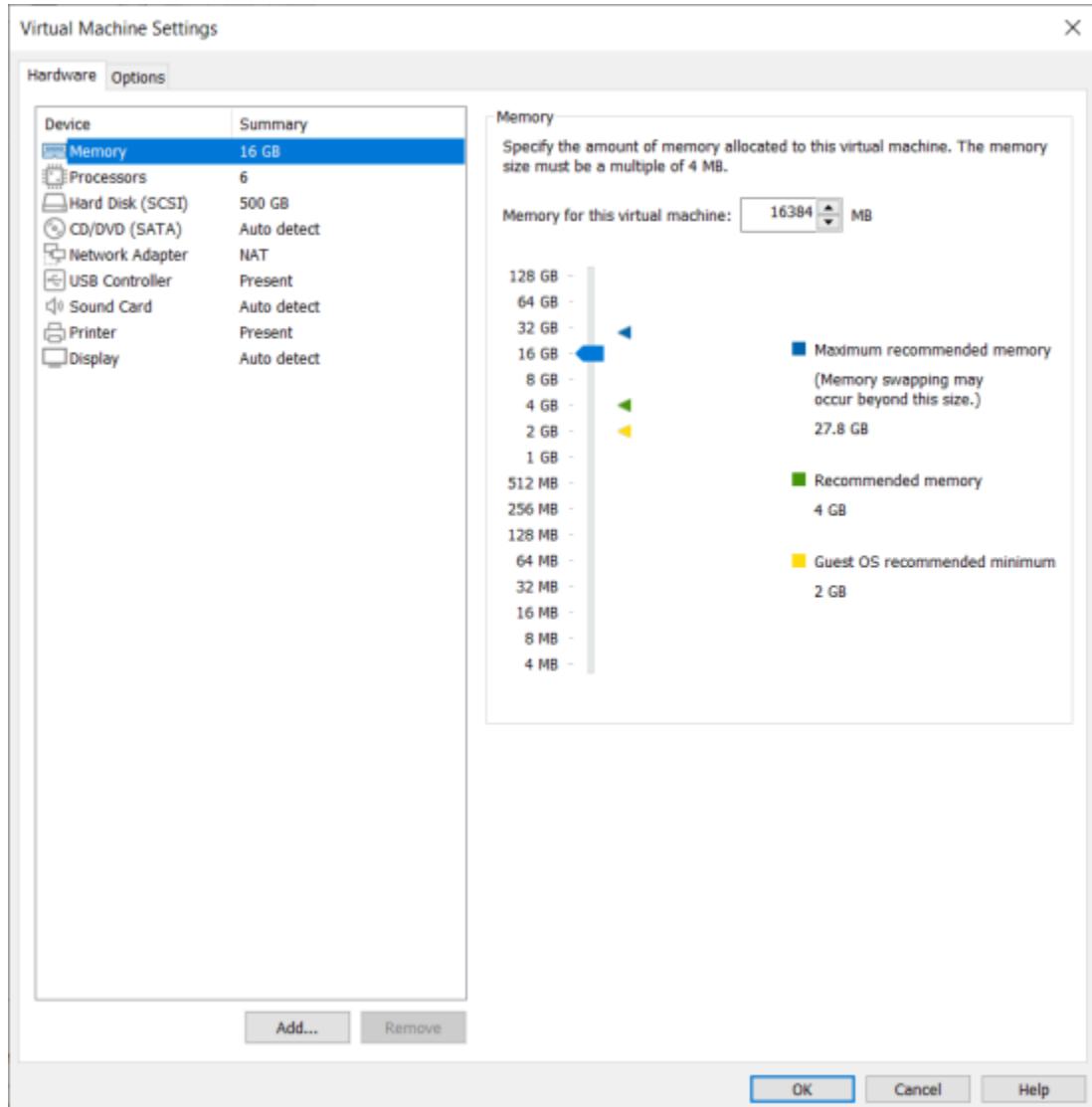
	ubuntu-20.04.3-desktop-amd64.iso	2021-08-19 11:06	2.9G
--	----------------------------------	------------------	------

After downloading the mirror image, you can proceed with the system installation operation.

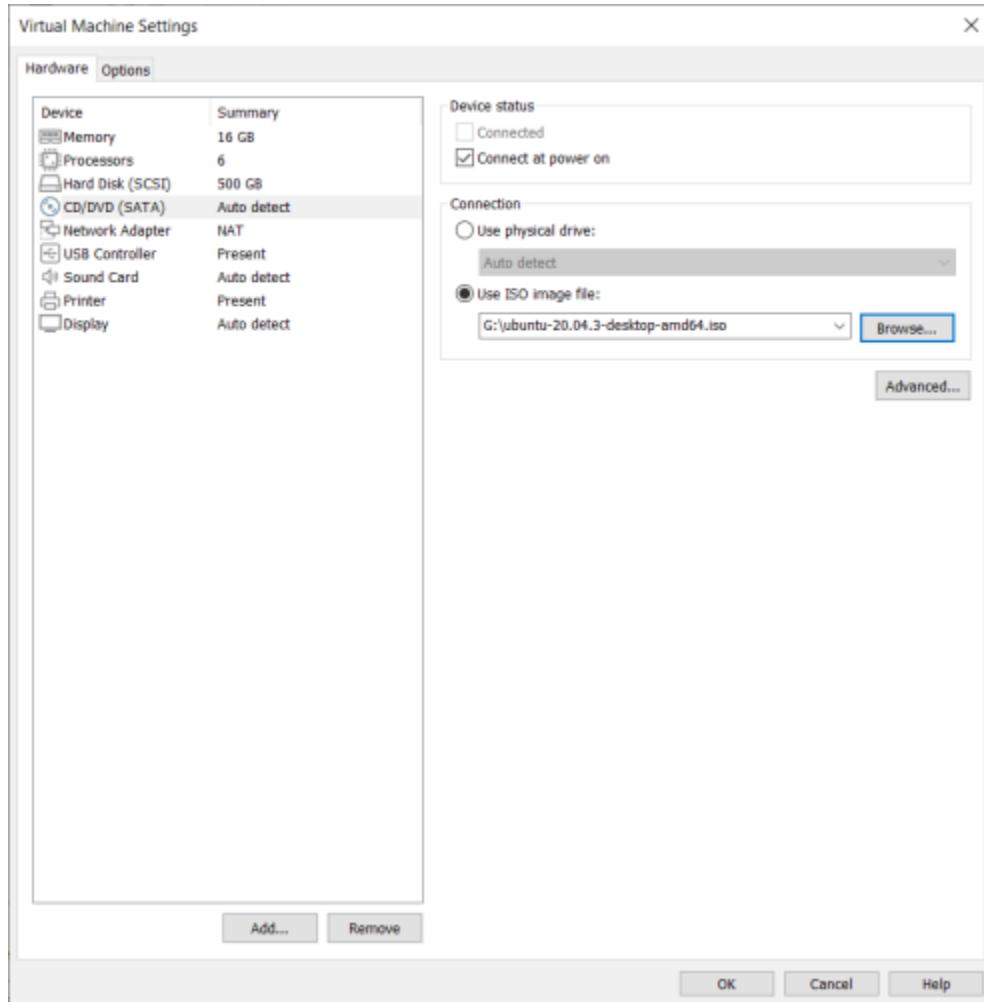
Right-click on the created virtual machine name and select “Settings” from the pop-up menu.



The “Virtual Machine Settings” menu will pop up.



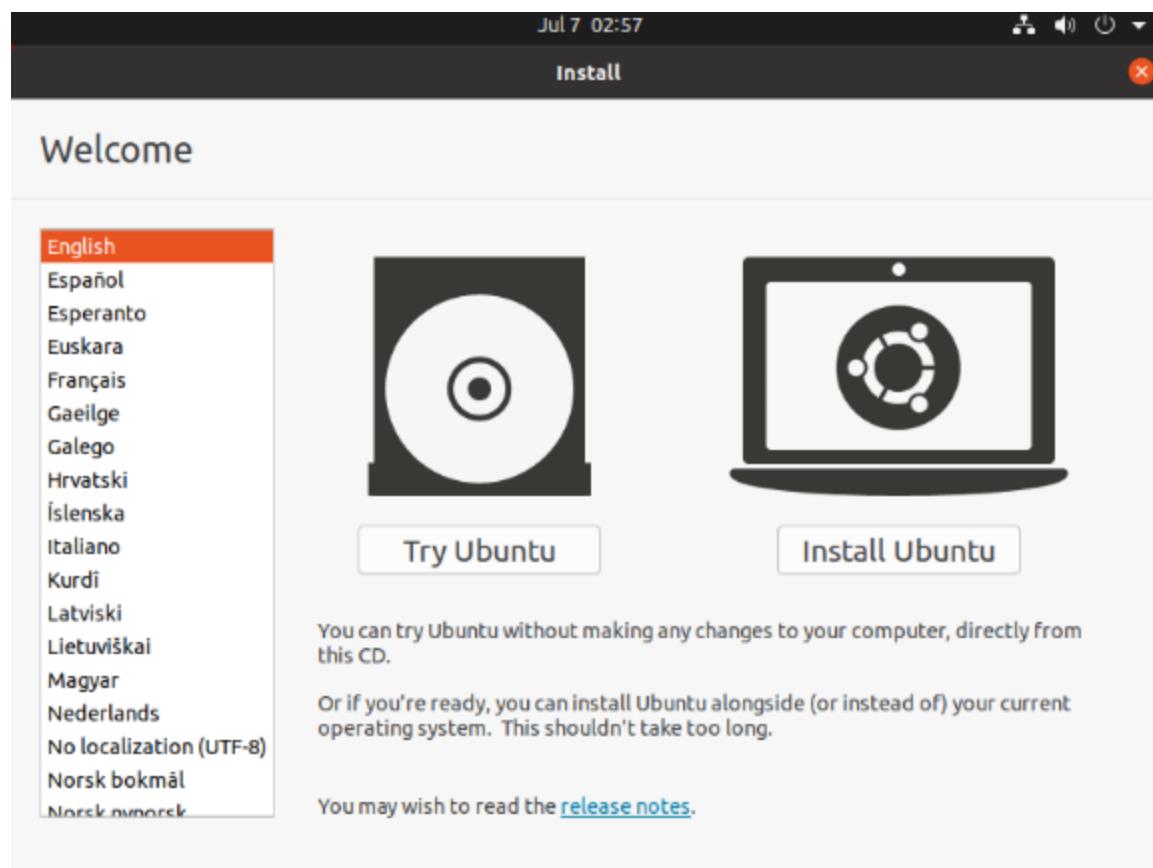
Click on CD/DVD (SATA), select “Use ISO image file,” browse and choose the previously downloaded Ubuntu image, then click “OK” to confirm.



After setting up the image, ensure that the network is available. Then, start the virtual machine and proceed with the installation of the Ubuntu image.

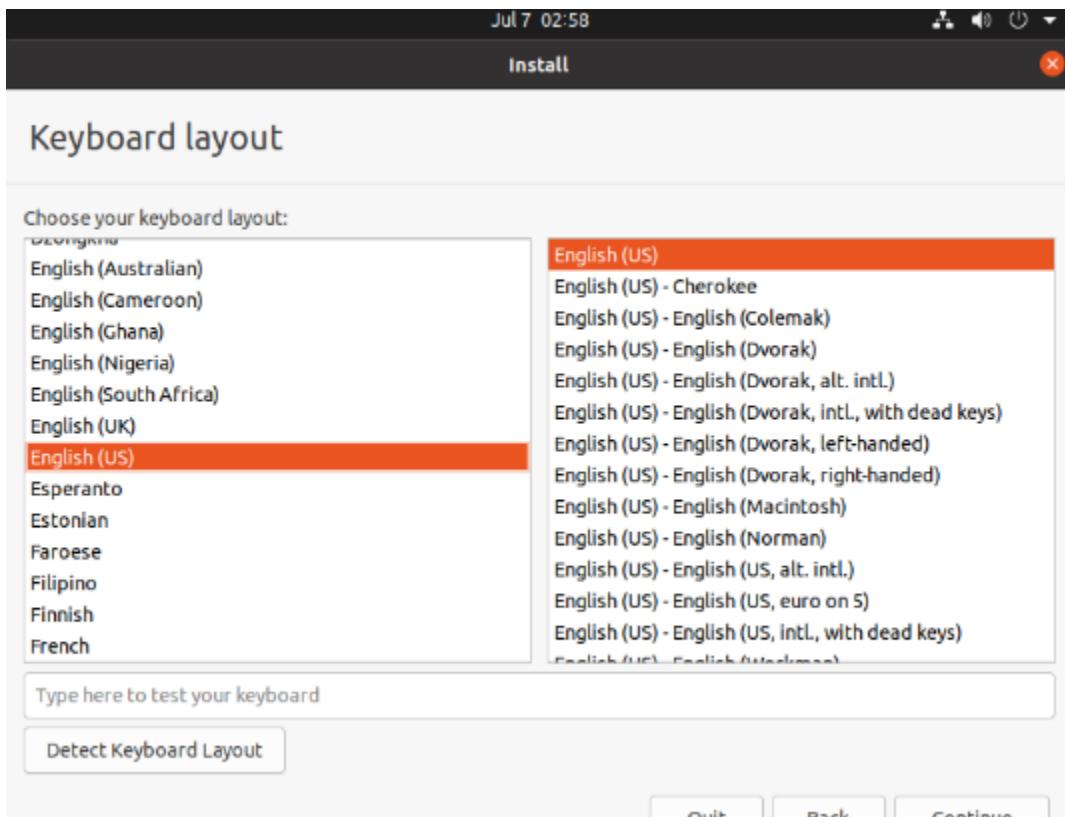


After starting the virtual machine, wait for the installation interface to appear as shown below.

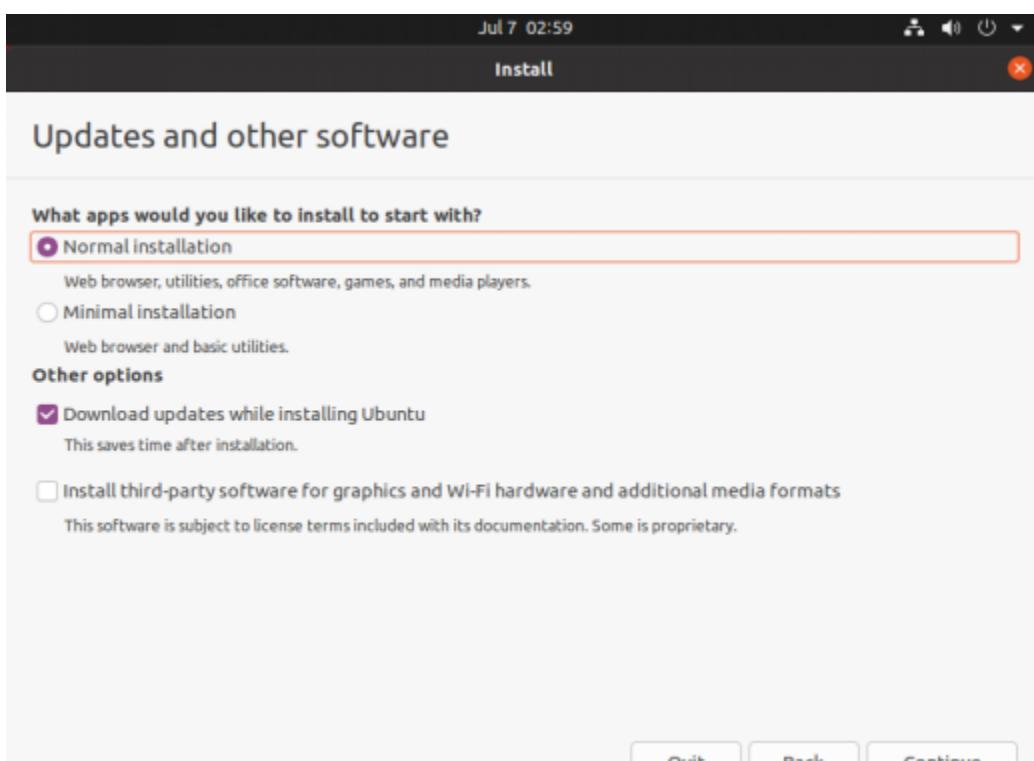


After selecting the language on the left side as shown in the image, click “Install Ubuntu”, and the language selection

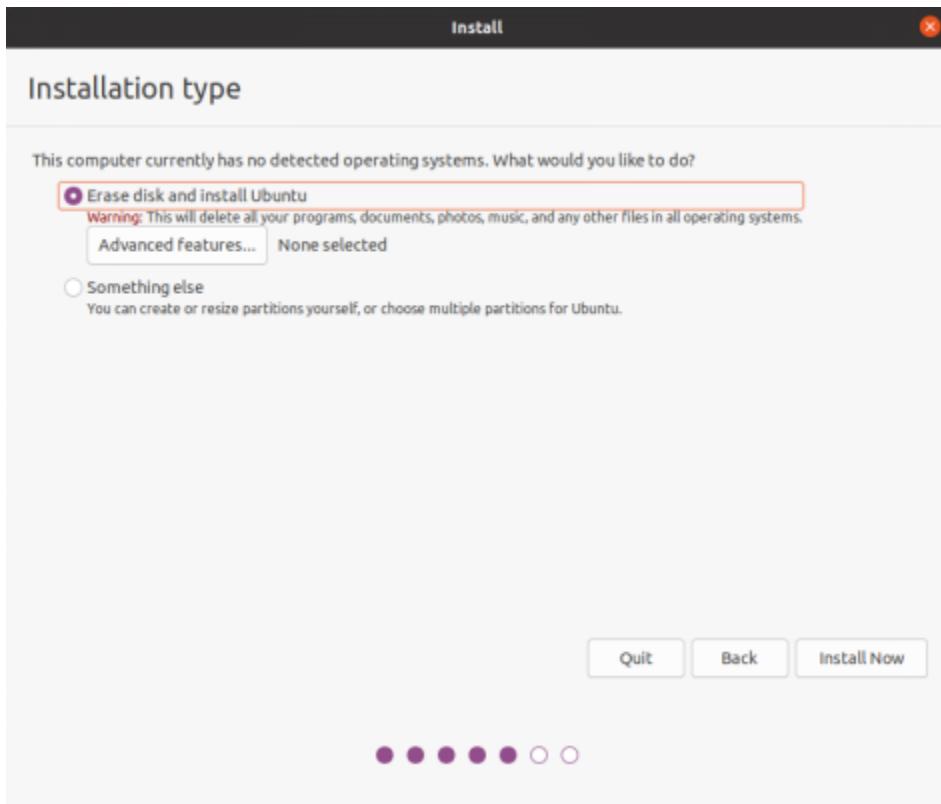
interface will pop up. The default language of Ubuntu is English, but of course, you can also choose Others.



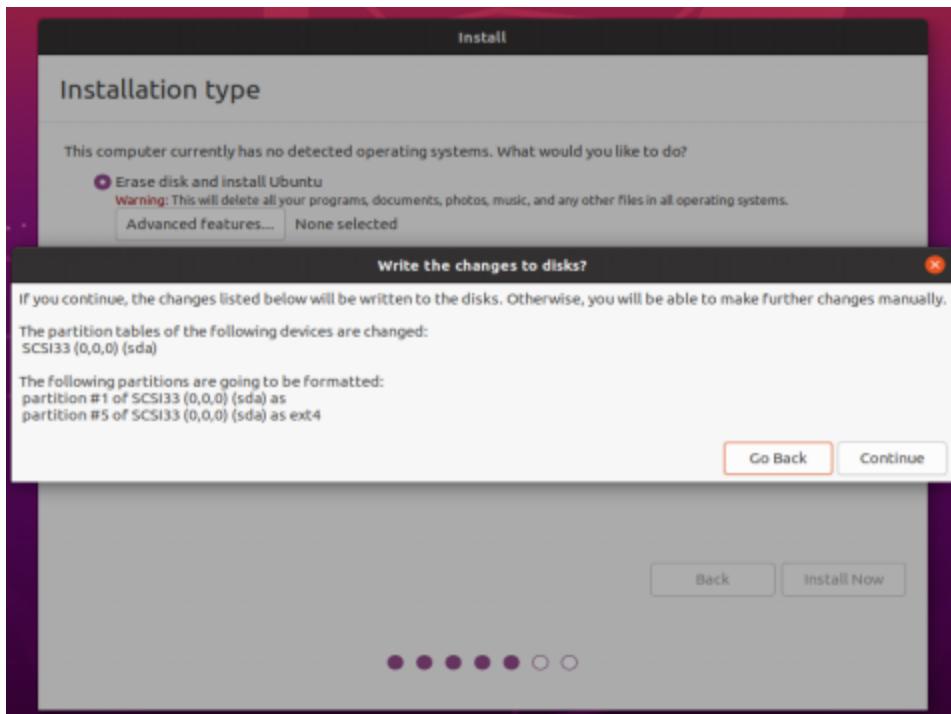
The default selected language can also be reset at a later stage, after the selection is complete continue.



Next, select “Continue” as the default option to proceed with the installation. The installation process might be slow. Then, click “Continue” again.

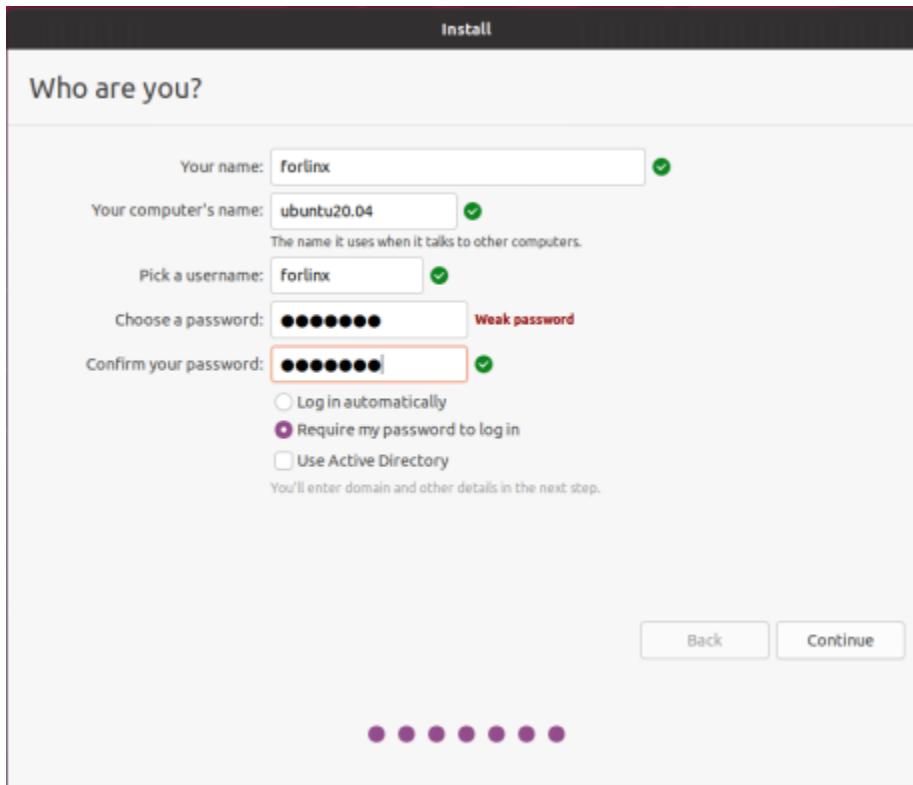


By default, when you click on “Install Now”, a dialog box will appear as shown in the image. Simply click “Continue” to proceed.

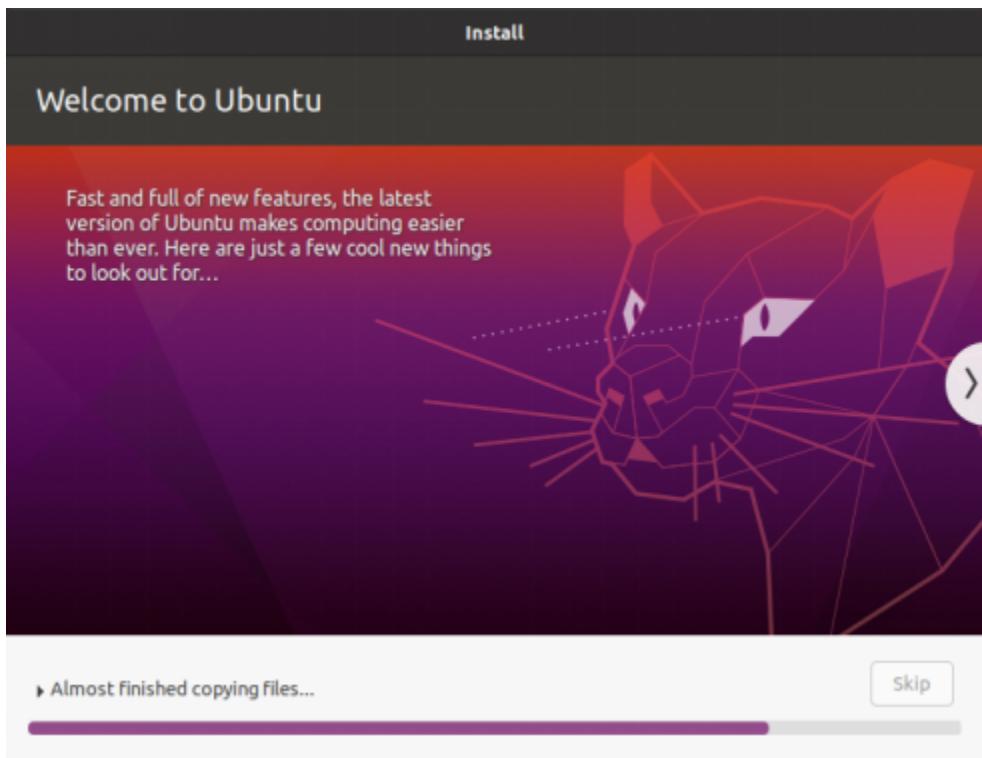


Next, select the timezone. You can either click on the Shanghai timezone or enter “Shanghai” (or choose the appropriate timezone based on your location). Then, click “Continue” to proceed.

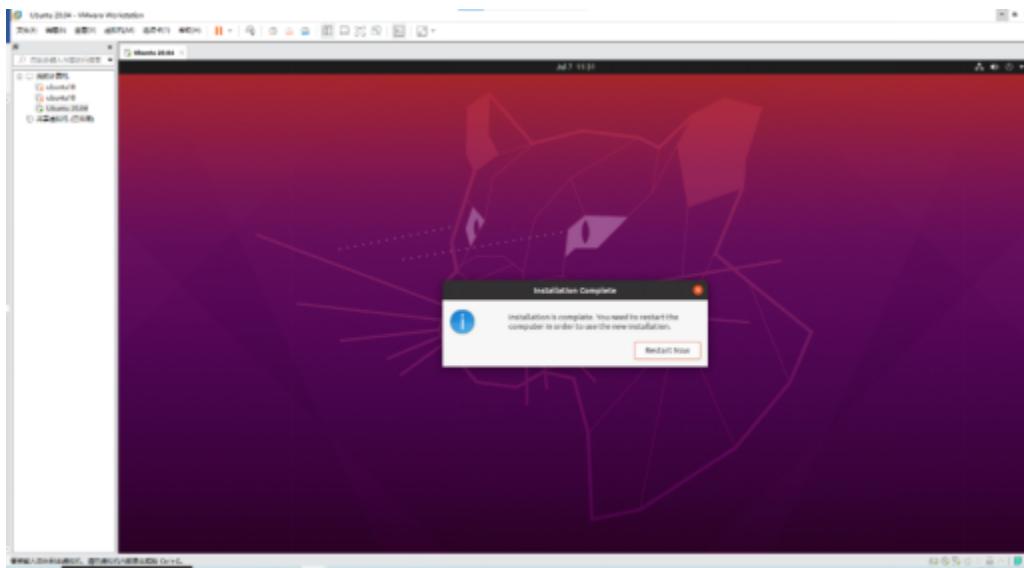
Finally, set your username and password. You can choose either automatic login or login with a username and password. Click “Continue” to start the automatic installation.

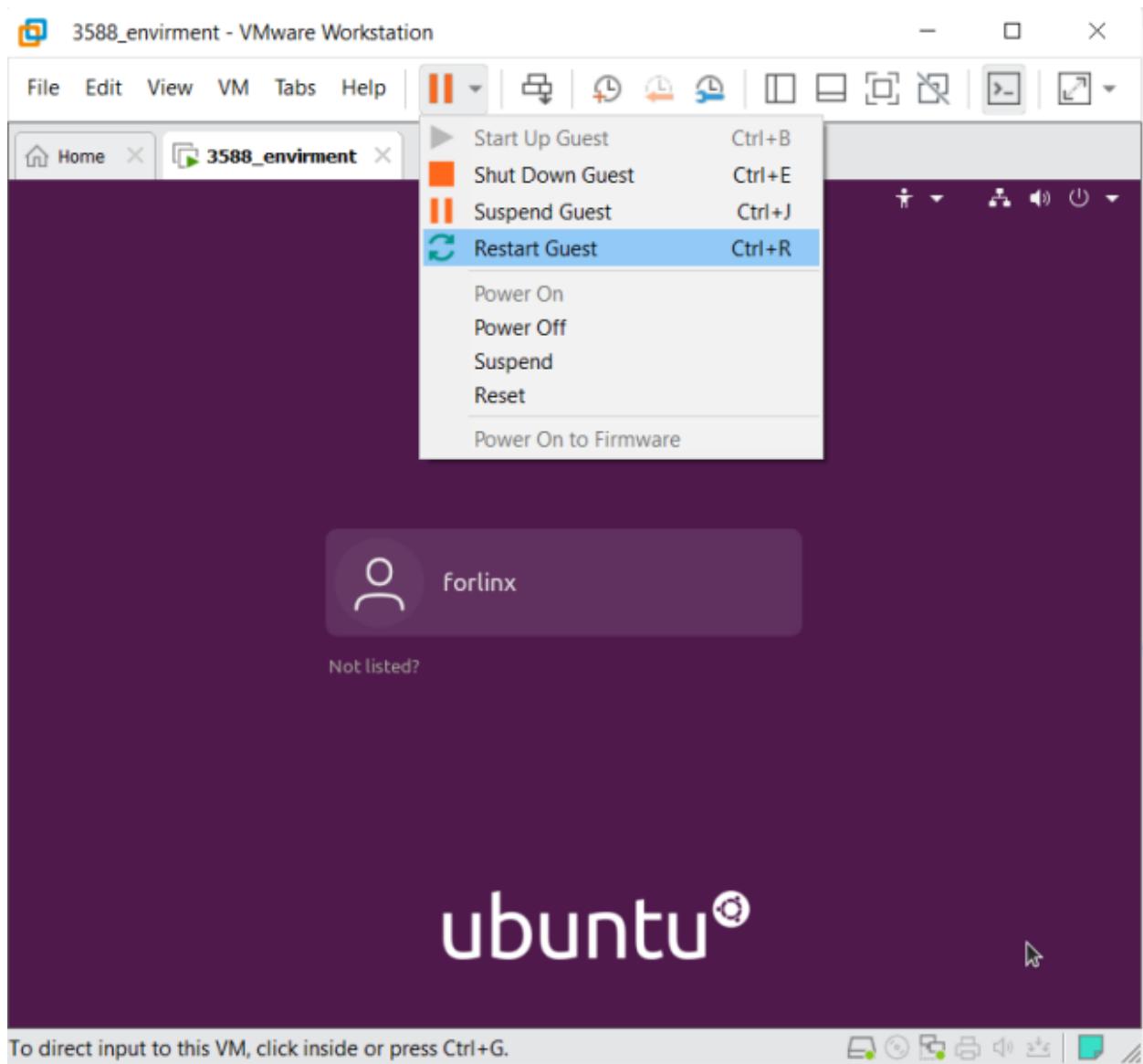


If the internet connection is poor, you can Skip without affecting the installation process.

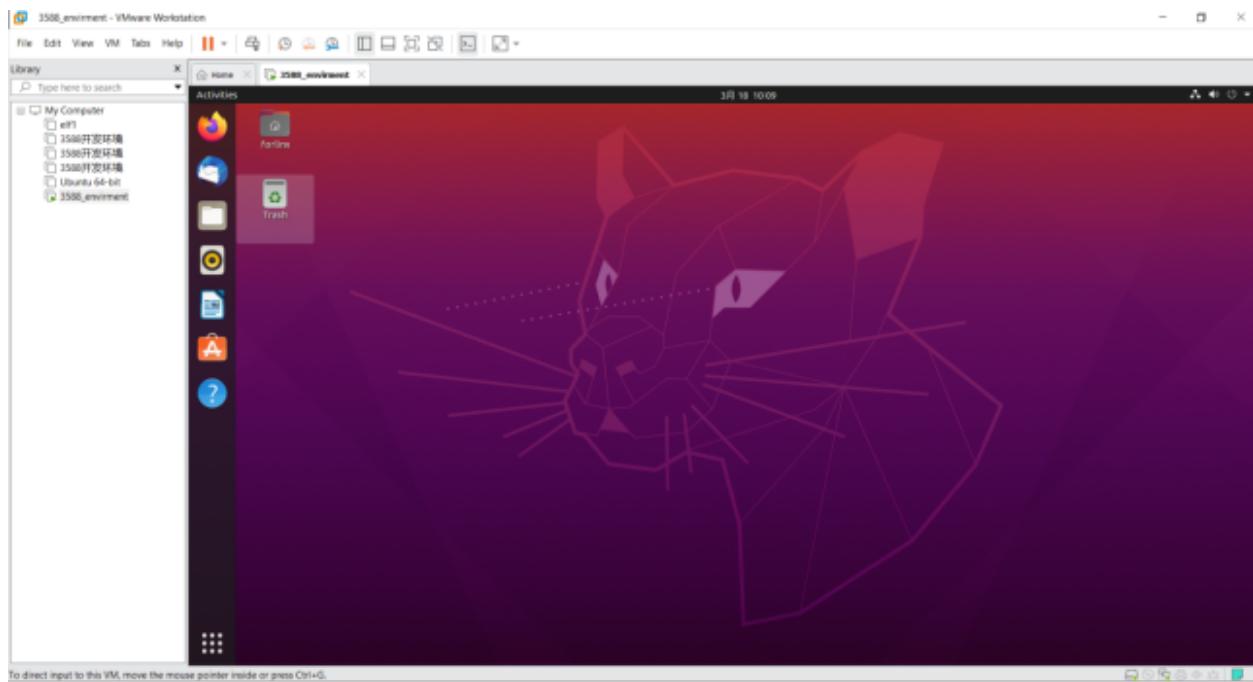


Click “Restart” Now to reboot.





The system interface after the reboot is complete.



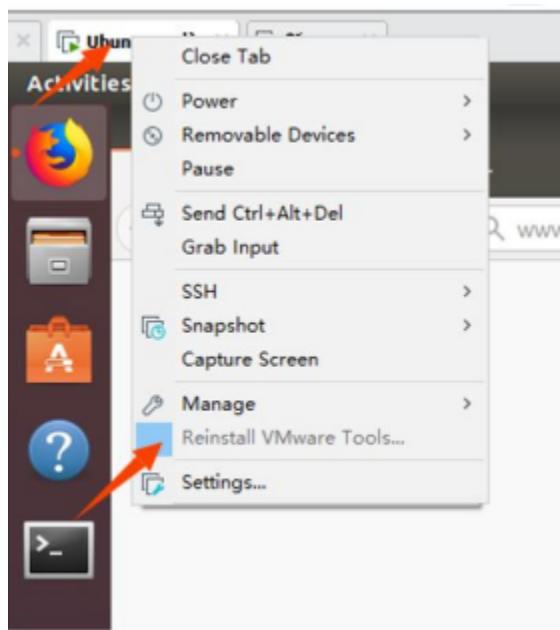
The ubuntu system installation is complete.

3.1.3 Ubuntu Basic Configuration

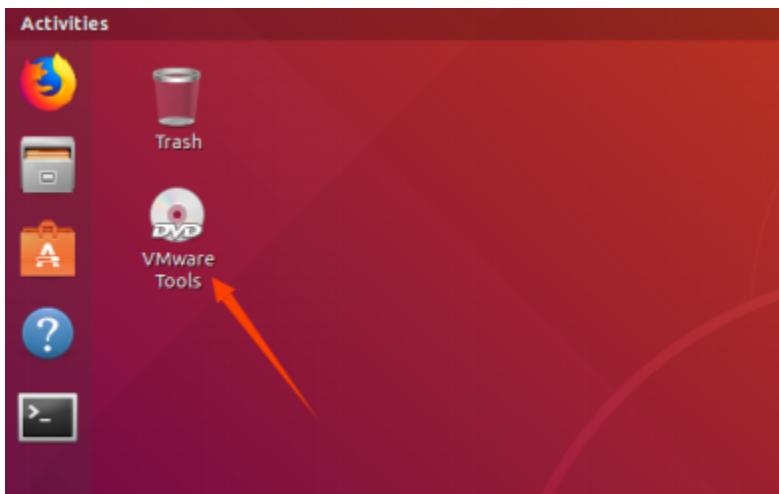
After installing the Ubuntu20.04 operating system, there are a few configurations to make.

VMware Tools Installation:

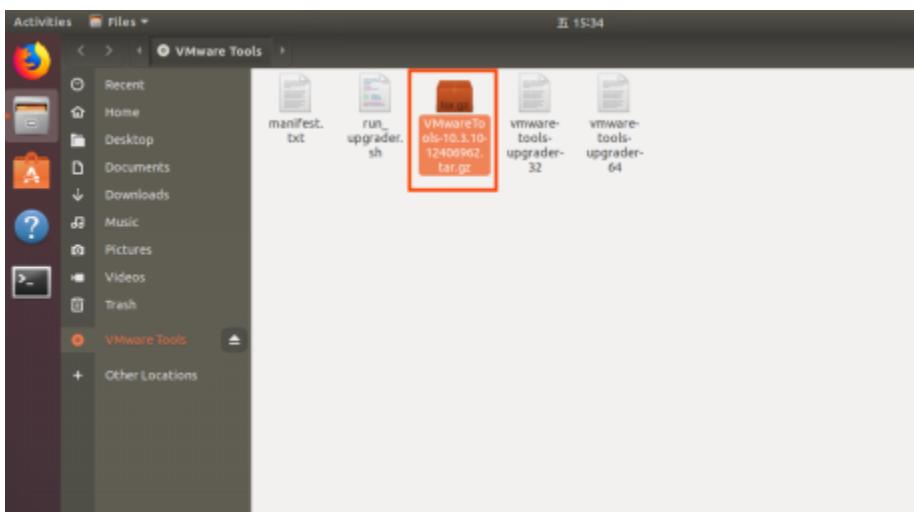
Next, install VMware Tools. Without installing this tool, you won't be able to copy and paste and drag file between the Windows host and the virtual machine. First click on “Virtual Machine” on the VMware navigation bar, then click “Install VMware Tools” in the drop-down box.



Once done, enter Ubuntu and the VMware Tools CD will appear on your desktop and click into it.

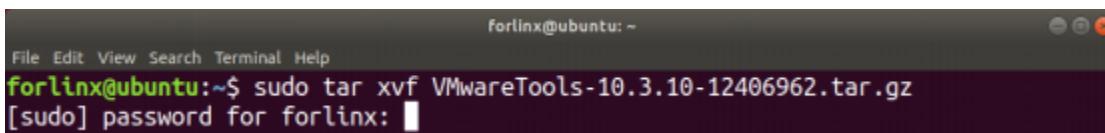


Enter and see a compressed file `VMwareTools-10.3.10-12406962.tar.gz` (it may be different for different VM versions); copy the file under the home directory (i.e. the directory with the home personal username)

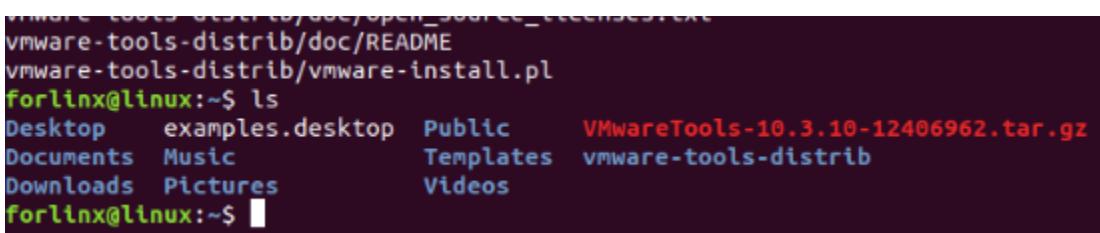


Press [Ctrl+Alt+T] to bring up the Terminal Command Interface and enter the command:

```
forlinx@ubuntu:~$ sudo tar xvf VMwareTools-10.3.10-12406962.tar.gz
```



After the extraction is complete, a file named “vmware-tools-distrib” will appear.



Go back to the terminal and type `cd vmware-tools-distrib` to enter the directory.

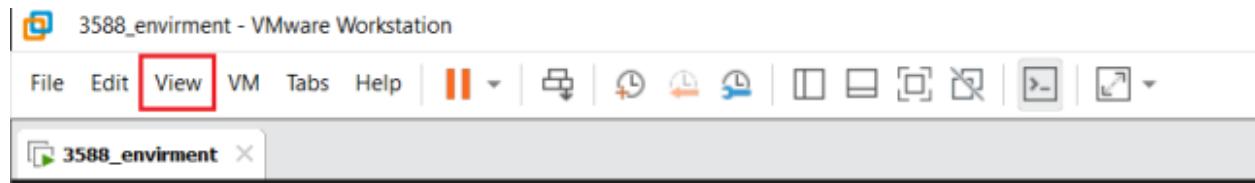
Enter: sudo ./vmware-install.pl followed by pressing Enter. Then, enter your password and the installation process will begin. When prompted, you can input “yes” and press Enter to proceed. For any other inquiries, simply press Enter to go with the default installation settings.

```
forlinx@linux:~/vmware-tools-distrib$ sudo ./vmware-install.pl
open-vm-tools packages are available from the OS vendor and VMware recommends
using open-vm-tools packages. See http://kb.vmware.com/kb/2073803 for more
information.
Do you still want to proceed with this installation? [no] yes
```

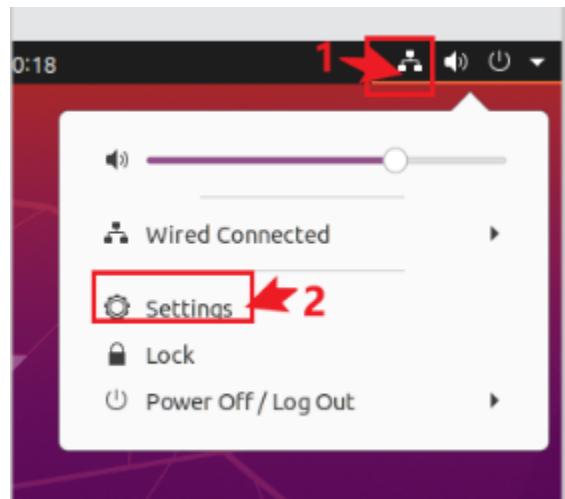
Once the VMware tools is complete, we can implement file copy and paste between Windows and Ubuntu.

The virtual machine is displayed full screen:

If the virtual machine is not able to be displayed in full screen, you can resolve this issue by clicking on “View” and selecting “Autofit Guest.” This will adjust the display to fit the screen automatically, enabling you to have a full-screen experience in the virtual machine.

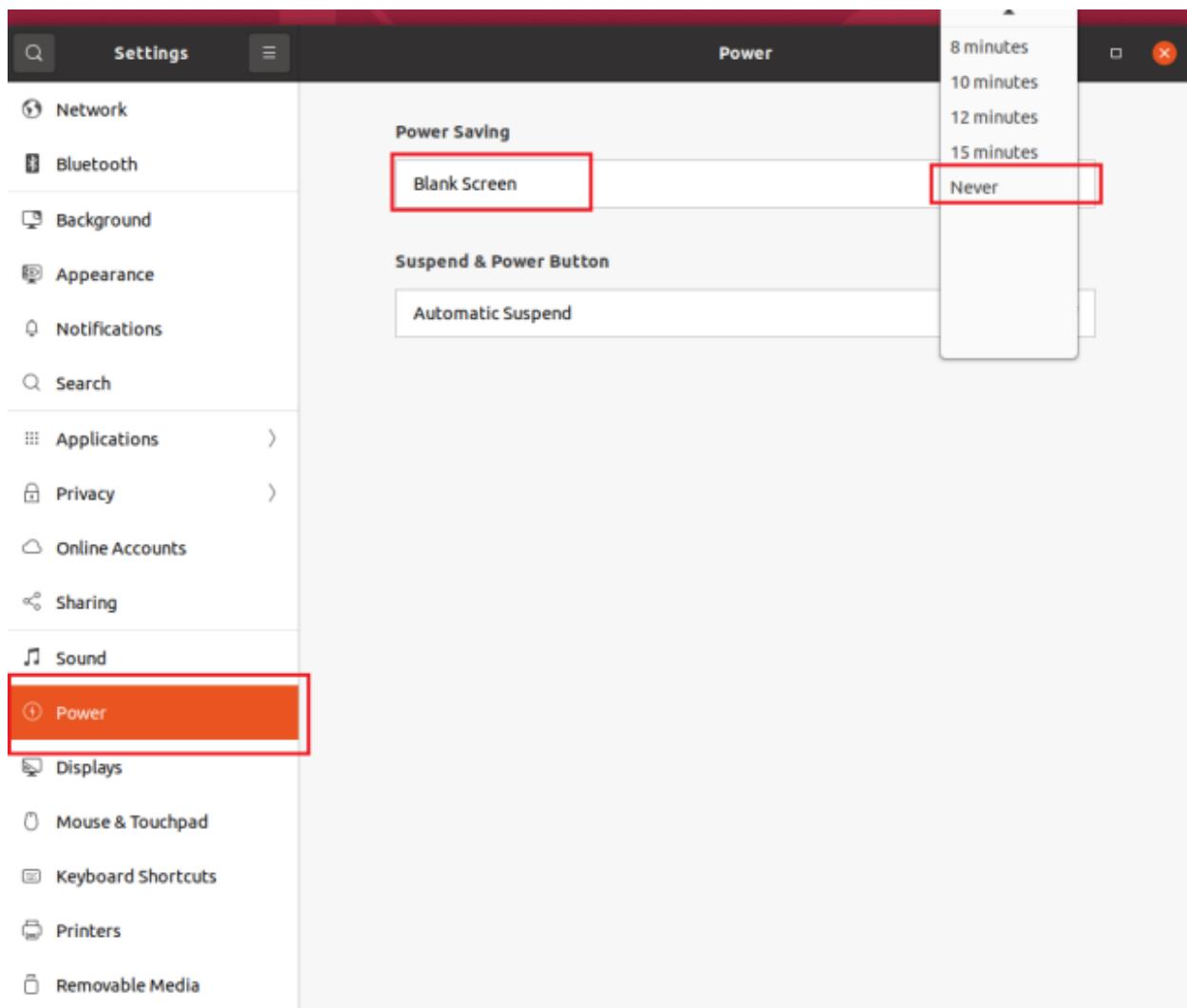


Make most of the system settings in the location shown. A lot of the setup requirements on Ubuntu can be done here.



Virtual machine hibernation settings:

Also, the default hibernation is 5min, if you don't want to set hibernation, just set it to Never by setting Power->Blank screen.

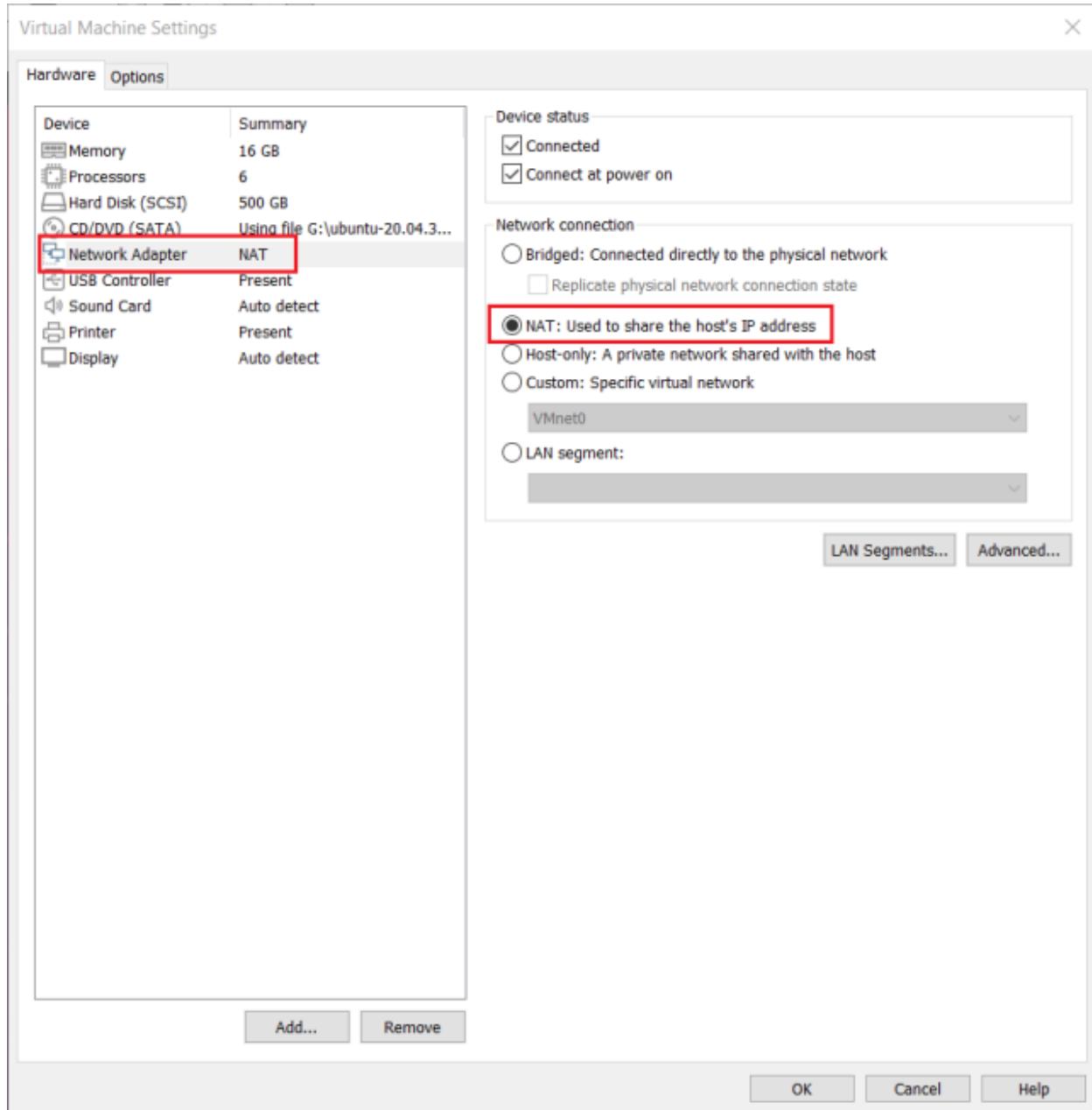


3.1.4 Network Settings for Virtual Machines

3.1.4.1 NAT Connection Method

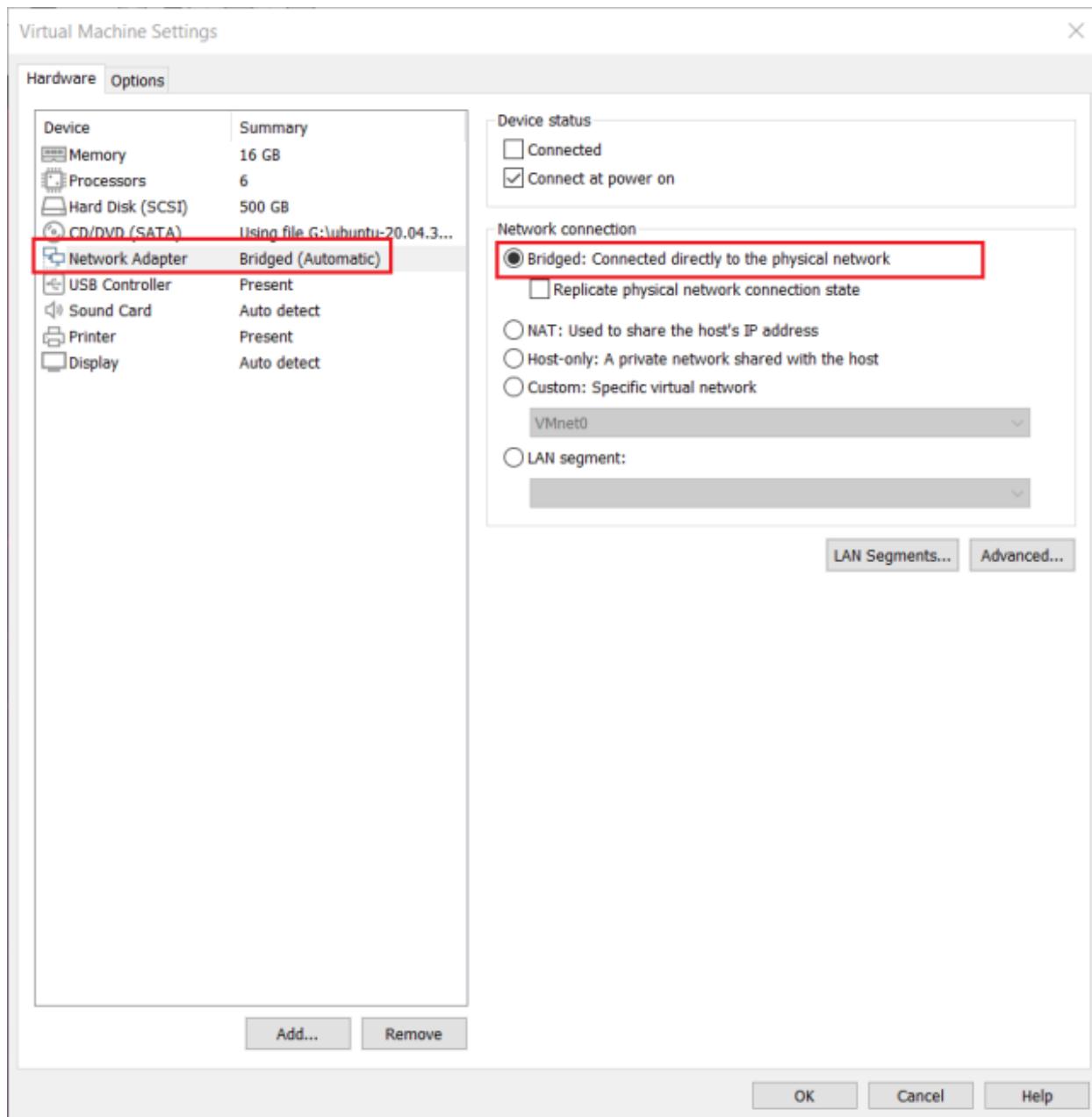
By default, after the virtual machine is installed, the network connection method is set to NAT, which shares the host machine's IP address. This configuration does not need to be changed when performing tasks like installing dependencies or compiling code.

When the VMware virtual NIC is set to NAT mode in a virtual machine, the network in the Ubuntu environment can be set to dynamic IP. In this mode the virtual NAT device and the host NIC are connected to communicate for Internet access. This is the most common way to access the external network.



3.1.4.2 Connections for Bridges

When the VMware virtual NIC device is in bridge mode, the host NIC and the virtual machine NIC communicate through the virtual bridge, and the network IP and the host need to be set in the same network segment in the Ubuntu environment. If accessing an external network, you need to set the DNS to be consistent with the host NIC. If TFTP, SFTP and other servers are used, the network contact mode of the virtual machine needs to be set as the bridge mode.



2.4.8 3.2 Installation of the Library Needed to Compile Android

Note: If you use the development environment provided by us, this section can be skipped directly.

Android compilation requires the installation of several toolkits. Make sure that your computer or virtual machine can be connected to the Internet normally before the operation in this section. If the network is disconnected during the installation, please follow the following steps to install.

1. Install and compile the necessary packages for Android.

```
forlinx@ubuntu:~$ sudo apt-get update
forlinx@ubuntu:~$ sudo apt-get install software-properties-common
forlinx@ubuntu:~$ sudo add-apt-repository ppa:openjdk-r/ppa
forlinx@ubuntu:~$ sudo apt-get update
```

(continues on next page)

(continued from previous page)

```
forlinx@ubuntu:~$ sudo apt-get install uuid \
    uuid-dev \
    zlib1g-dev \
    liblz-dev \
    liblzo2-2 \
    liblzo2-dev \
    lzop \
    git curl \
    u-boot-tools \
    mtd-utils \
    android-sdk-libsparse-utils \
    openjdk-8-jdk \
    device-tree-compiler \
    gdisk \
    m4 \
    make bc fakeroot unzip zip gawk busybox libstdc++6 lib32stdc++6 \
    bison flex python libssl-dev cpio lz4 rsync
```

The following libraries also need to be installed when using the Network Configuration Tool and menuconfig:

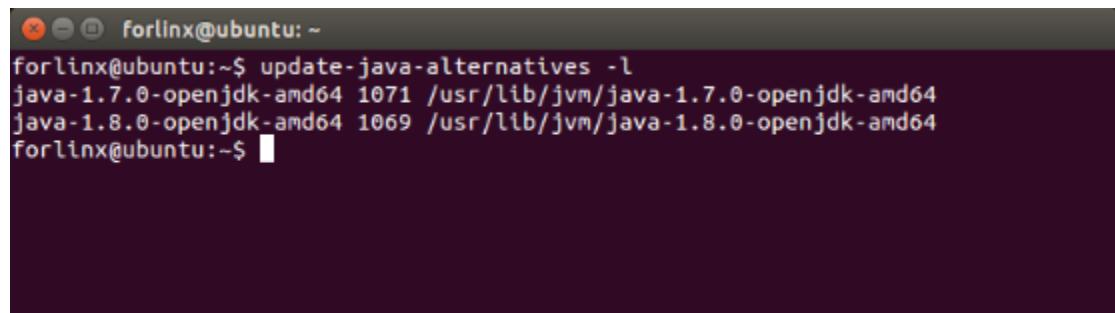
```
forlinx@ubuntu:~$ sudo apt-get update //Update the download
↳source information
forlinx@ubuntu:~$ sudo apt-get install libncurses* //For building text-
↳based user interfaces
forlinx@ubuntu:~$ sudo apt-get install net-tools //Network
↳configuration tool
```

Switch to the JDK version

If you have previously installed another version of the JDK, you will need to switch versions.

To view the installed version:

```
forlinx@ubuntu:~$ update-java-alternatives -l
```



```
forlinx@ubuntu:~$ update-java-alternatives -l
java-1.7.0-openjdk-amd64 1071 /usr/lib/jvm/java-1.7.0-openjdk-amd64
java-1.8.0-openjdk-amd64 1069 /usr/lib/jvm/java-1.8.0-openjdk-amd64
forlinx@ubuntu:~$
```

```
forlinx@ubuntu:~$ sudo update-alternatives --config java
```

```
forlinx@ubuntu:~$ update-alternatives -l
java-1.7.0-openjdk-amd64 1071 /usr/lib/jvm/java-1.7.0-openjdk-amd64
java-1.8.0-openjdk-amd64 1069 /usr/lib/jvm/java-1.8.0-openjdk-amd64
forlinx@ubuntu:~$ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                  Priority  Status
-----  -----
* 0        /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java  1071    auto m
1      /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java   1071    manual
* 2        /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java  1069    manual

Press enter to keep the current choice[*], or type selection number: 2
```

Enter the option corresponding to java8 and press Enter to confirm.

2.4.9 4. Android System Compilation

You can access the documentation and source code of the software and hardware through the web link provided by Forlinx.

The compilation process needs to be carried out in the docker container. The program should be modified outside the docker as far as possible. The vim command in the docker is not very convenient to use.

4.1 Software Configuration File Path

OK3588-C platform, the software configuration file path is as follows:

Docu- ment type	Path
Kernel configura- tion file	arch/arm64/configs/OK3588-Android_defconfig
Device tree file	arch/arm64/boot/dts/rockchip/FET3588-C.dtsiarch/arm64/boot/dts/rockchip/OK3588-C- Common.dtsiarch/arm64/boot/dts/rockchip/OK3588-C-Camera.dtsiarch/arm64/boot/dts/rockchip/ OK3588-C-Android.dts
android	device/rockchip/rk3588/ok3588_c/

4.2 System Compilation

4.2.1 Preparation Before Compilation

Please confirm the size of the swap partition of the current system. If the swap partition is insufficient, the compilation of the Android source code will fail. 10G is recommended. It is recommended to adjust the development environment memory to 16G.

1. View the swap partition:

```
forlinx@ubuntu:~$ cat /proc/swaps
```

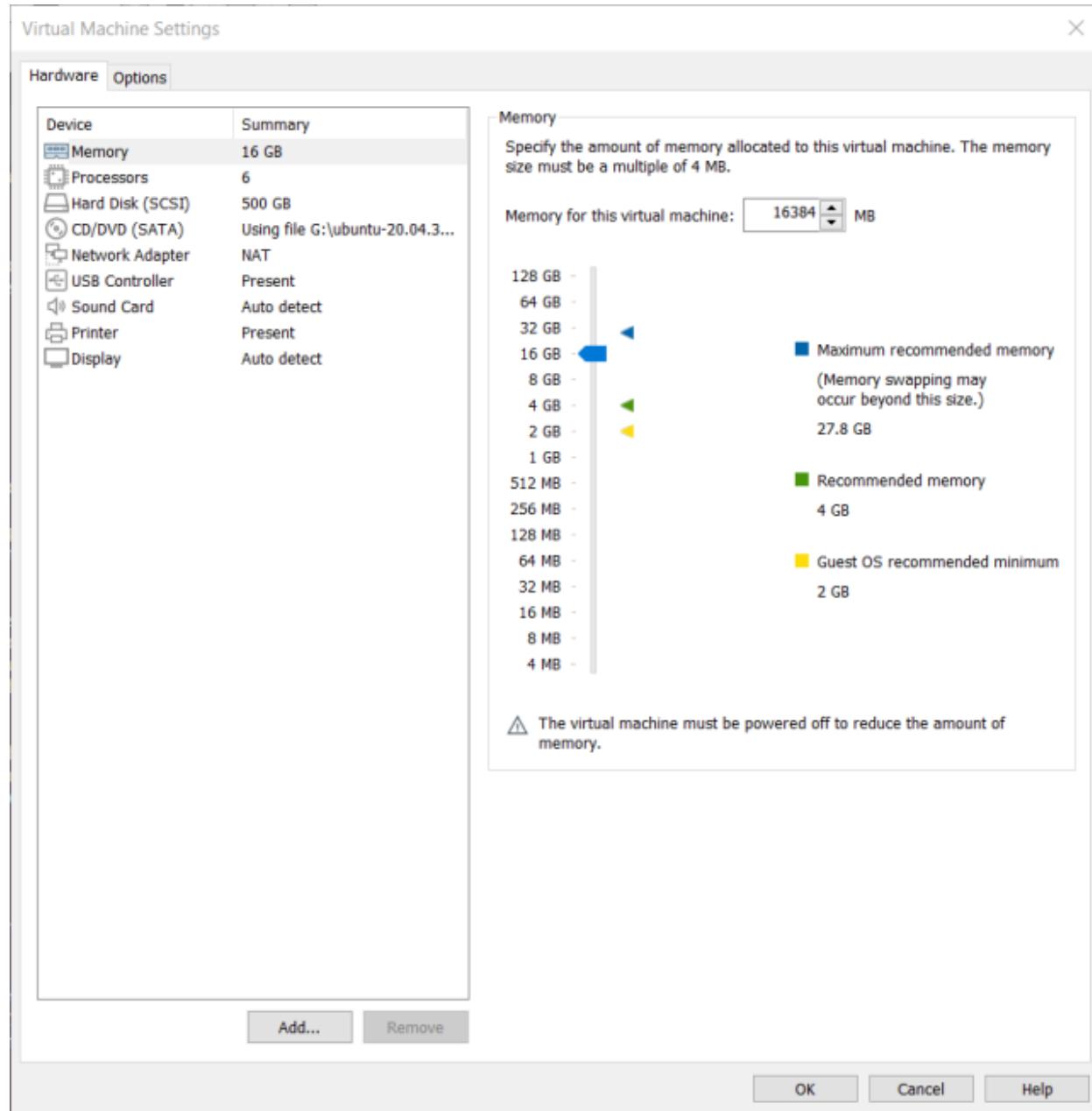
The virtual machine provided by Forlinx has been configured with the swap partition by default. If you use other virtual machines, you can increase the size of the swap partition by creating a swap file:

```
forlinx@ubuntu:~$ sudo swapoff /swapfile
forlinx@ubuntu:~$ sudo fallocate -l 10G /swapfile
forlinx@ubuntu:~$ sudo chmod 600 /swapfile
forlinx@ubuntu:~$ sudo mkswap /swapfile
forlinx@ubuntu:~$ sudo swapon /swapfile
forlinx@ubuntu:~$ sudo vim /etc/fstab
```

Adding the following at the end of the/etc/fstab file:

```
/swapfile none swap sw 0 0
```

2. It is recommended to adjust the memory of the development environment to 16G. Low memory may cause the compilation to fail.



The source code package of the Android 12 system is placed in the data in the form of a volume compression package. Copy all the compression packages to the /home/forlinx/work directory of the development environment, which is described below.

4.2.2 Source Code Copy and Release

Source Code Package Path: OK3588-C (Android) User Profile \ Android \ Source Code \

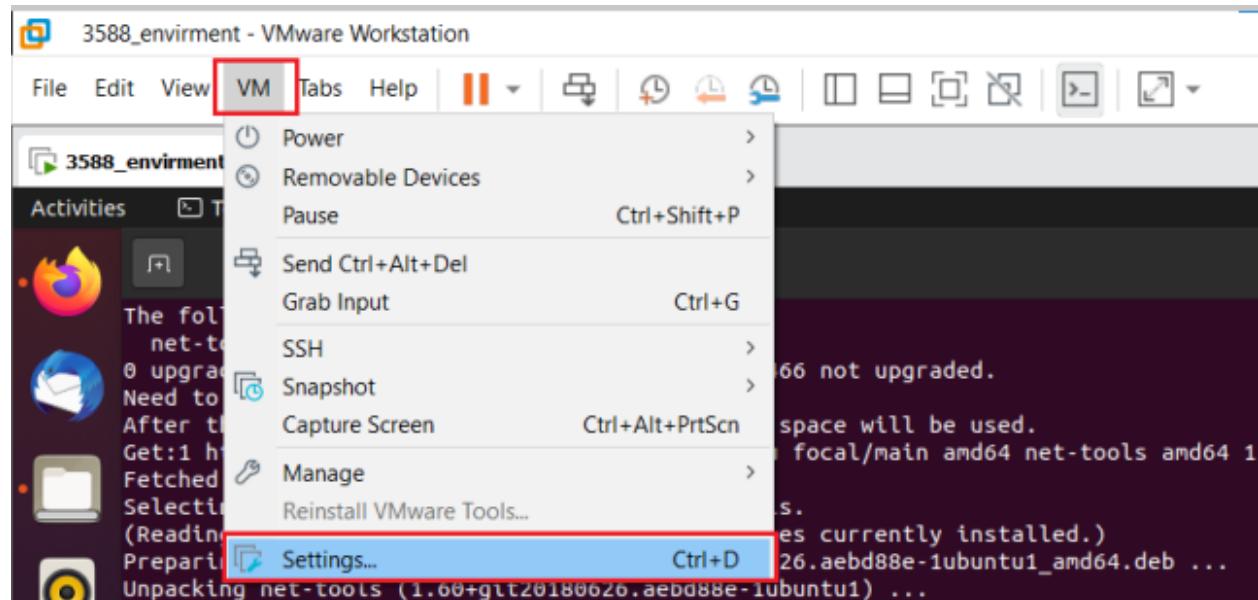
The source code in the data is a volume compression package, which needs to be merged first, and then decompressed and released.

1. Copy all source code compressed packages to the /home/forlinx/work directory of the development environment, and perform MD5 verification.

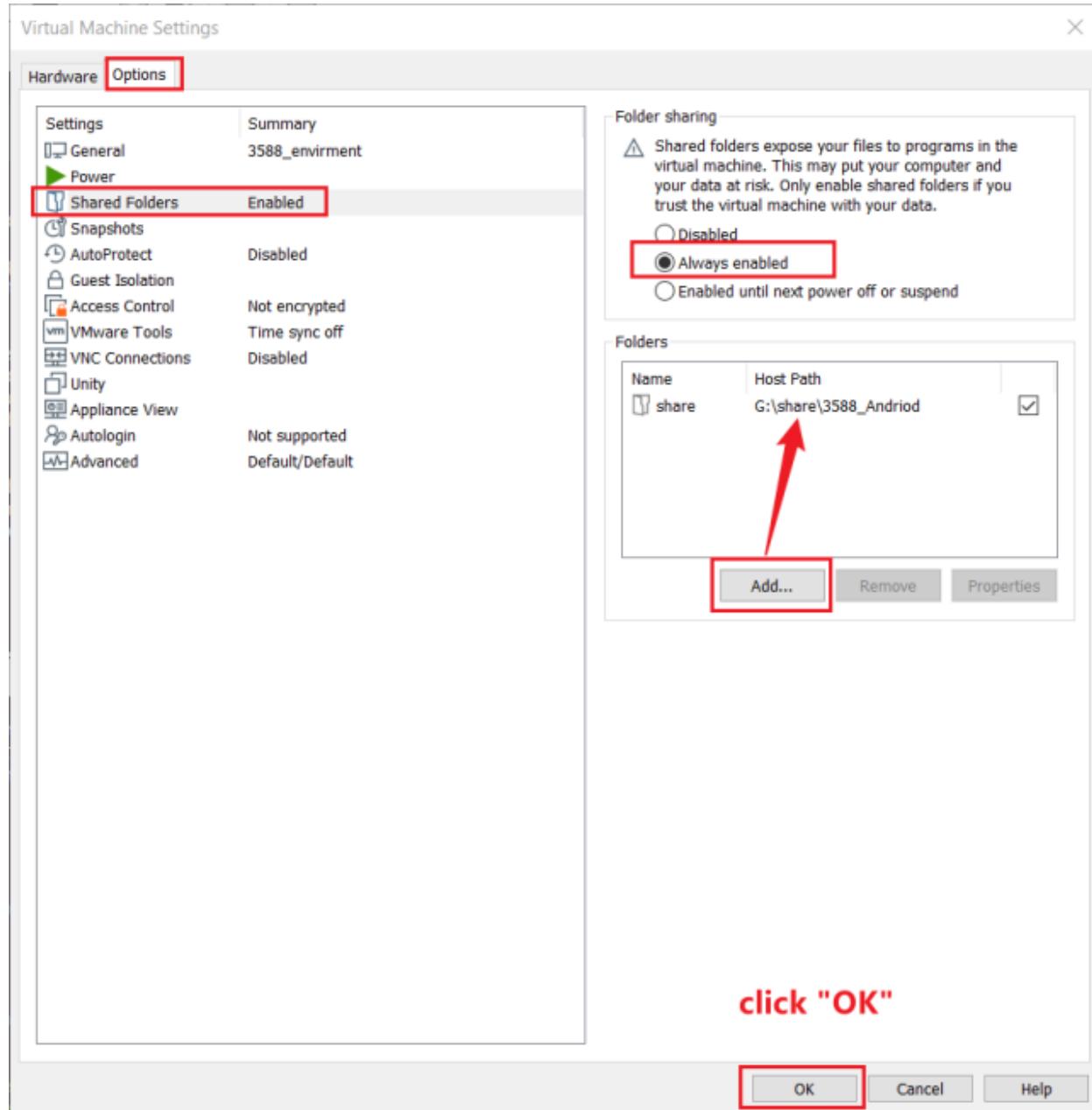
You can directly drag the source code package on the computer to the folder on the desktop of the virtual machine, or use the shared folder to use the command copy. Here we focus on the use of the shared folder.

There are many kinds of file transfers between ubuntu and Windows hosts. After installing VMware Tools, you can set up a virtual machine shared folder to mount the file directory of the Windows host to ubuntu for file sharing.

Click “Virtual Machine” on the menu bar and select “Settings” .



Click “Options”, enable “Shared Folders”, set the shared directory on the Windows host, and click “OK” .



After the file-sharing setup of the virtual machine is complete, place the source zip in the shared folder of the Windows host, here we name it “share” .

Shared folder in ubuntu in the mount directory /mnt/hgfs/share.

Copy the source code from the shared folder to ubuntu’s /home/forlinx/work directory for md5 checksum:

```
forlinx@ubuntu:~$ cp /mnt/hgfs/share/OK3588-android-source* /home/forlinx/work/
forlinx@ubuntu:~$ cd /home/forlinx/work
forlinx@ubuntu:~/work$ md5sum OK3588-android-source*
```

If the returned MD5 check code is consistent with the check code in the data, the source code can be merged.

2. Merge and unpack Android source code

First, merge the archives into a single archive:

```
forlinx@ubuntu:~/work$ cat OK3588-android-source.tar.bz2.* > OK3588-android-source.
tar.bz2
```

Decompress the combined compressed package

```
forlinx@ubuntu:~/work$ tar xvf OK3588-android-source.tar.bz2
```

3. Modify source code file group

View the current account ID. This example takes the development environment forlinx account provided by Forlinx as an example.

```
forlinx@ubuntu:~/work/$ id
uid=1000(forlinx) gid=1000(forlinx) groups=1000(forlinx),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),124
(sambashare)
```

Modify source code file group

```
forlinx@ubuntu:~/work/$ sudo chown 1000:1000 -R OK3588-android-source
forlinx@ubuntu:~/work/$ ls -l
drwxr-xr-x 4 forlinx forlinx 4096 4 月 19 10:50 OK3588-android-source
```

4. Add the Java runtime environment variable at the end of the.bashrc environment variable file in the user's home directory:

```
forlinx@ubuntu:~$ vim /home/forlinx/.bashrc
```

The additions are as follows:

```
export _JAVA_OPTIONS="-Xms64m -Xmx2g"
export MAVEN_OPTS="-Xms5120m -Xmx5120m"
```

Note: The development environment we provide modifies the.bashrc file as a hidden file, which can be viewed with the ls -a command and modified with the vim command.

```
forlinx@ubuntu:~$ source ~/.bashrc
```

4.2.3 Android System Compilation

The android system compilation part of OK3588 platform supports full and partial compilation. All contents will be automatically compiled and packaged into an image that can be directly used for flashing. Part of the compilation is mainly used in the product development phase to compile the Linux kernel device tree or android system.

1. Enter docker environment

To compile this system, you need to switch to the forlinx account

```
root@4d199700bc75:/# su forlinx
forlinx@4d199700bc75:$
```

2. Full compilation

Enter the longan directory and execute the following command:

```
forlinx@4d199700bc75:~$ cd /home/forlinx/work/OK3588-android-source/
forlinx@4d199700bc75:~/OK3588-android-source/$ ./forlinx.sh all
```

Because the android system is too large, the compilation time is slightly longer. Please wait patiently for the compilation to be completed. The compiled image is located in the rockdev/Image-ok3588_c/directory. The image file is a update. img

3. Compile the kernel separately

```
forlinx@4d199700bc75:~$ cd /home/forlinx/work/OK3588-android-source/
forlinx@4d199700bc75:~/OK3588-android-source/$ ./forlinx.sh kernel
forlinx@4d199700bc75:~/OK3588-android-source/$ ./forlinx.sh kernel_pack
```

The compiled boot. img is /out/target/product/ok3588_c\$.

2.4.10 5. Android Application Development

This chapter explains how to set up the Android application development environment, including downloading and installing the Android SDK and Android Studio integrated development environment, as well as using the OK3588 development board for on-device debugging. It is highly suitable for Android beginners to learn from and refer to.

5.1 Android Application Development Environment Setup

5.1.1 JDK (Java SE Development Kit) Download and Installation

Since the Android application code is written in Java, you need to install the JDK on Windows first. The JDK can be downloaded as follows:

It is recommended to install Java 8, visit the website <https://www.oracle.com/java/technologies/downloads>, and select the Windows x64 version of Java 8 in the page. You can also directly visit the website <https://www.oracle.com/java/technologies/downloads/#java8-windows> to enter the download page.

The screenshot shows the Oracle Java Downloads page. At the top, there are links for 'Java downloads', 'Tools and resources', and 'Java archive'. Below this, a section titled 'Java SE Development Kit 8u333' is shown. It includes a note that Java SE subscribers will receive JDK 8 updates until at least December of 2030. A note also states that the Oracle JDK 8 license changed in April 2019. Below this, a note about the Oracle Technology Network License Agreement for Oracle Java SE is provided, stating that it is substantially different from prior Oracle JDK 8 licenses. It mentions that this license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. A link to FAQs is provided. Further down, it says commercial license and support are available for a low cost with Java SE Subscription. It also notes that JDK 8 software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE. A 'JDK 8u333 checksum' link is present. At the bottom, there are tabs for 'Linux', 'macOS', 'Solaris', and 'Windows', with 'Windows' being the active tab. A table provides product/file descriptions, file sizes, and download links for x86 and x64 installers.

Product/file description	File size	Download
x86 Installer	159.21 MB	jdk-8u333-windows-i586.exe
x64 Installer	172.66 MB	jdk-8u333-windows-x64.exe

Accept the agreement and proceed with the download.

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

- I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE
Required

You will be redirected to the login screen in order to download the file.

[Download jdk-8u333-windows-x64.exe](#) 

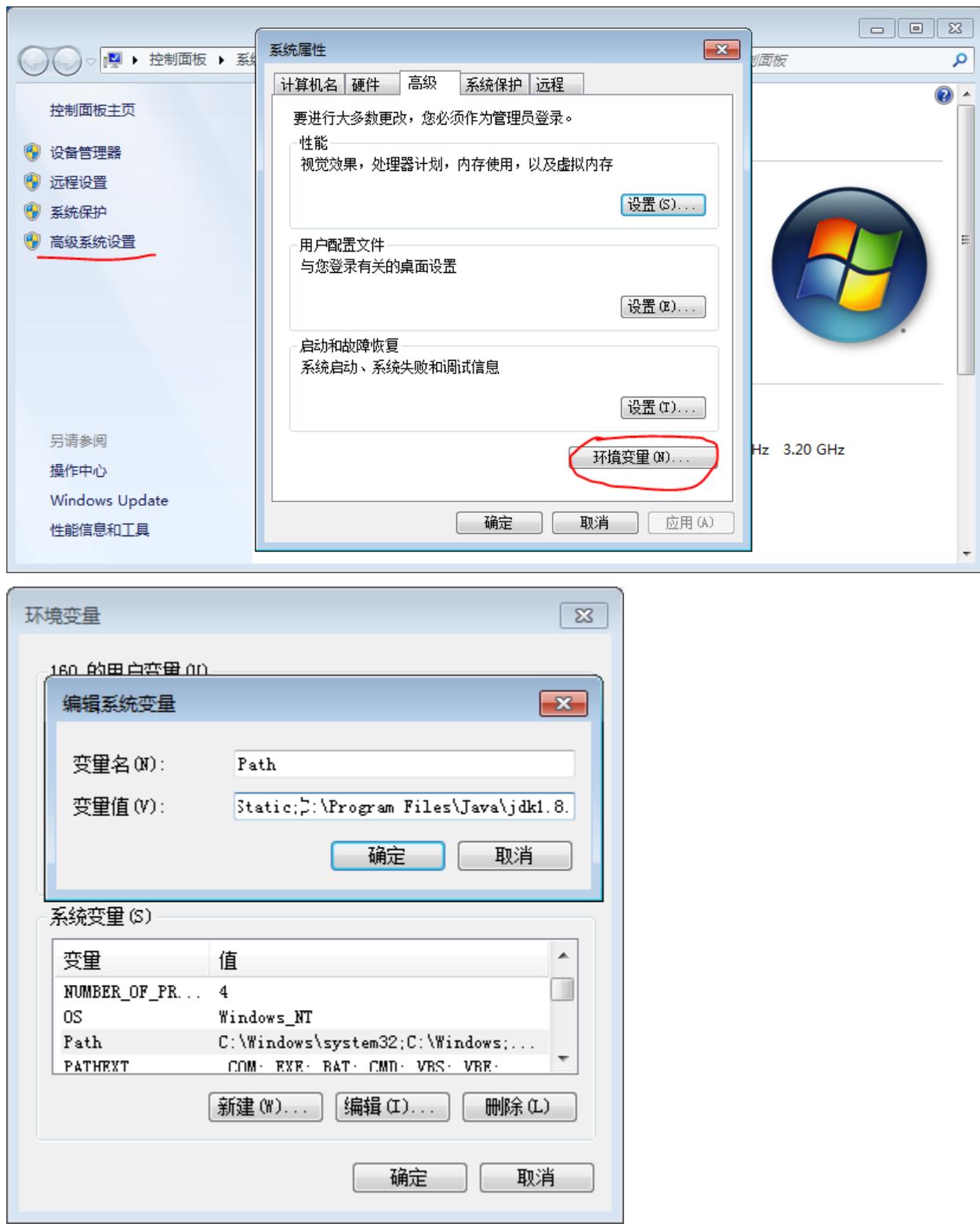
Currently, download JDK from the official Oracle website requires registration for an Oracle account. You may also choose alternative download methods.

After the download is completed, double-click the installer to complete the installation according to the wizard's prompts.

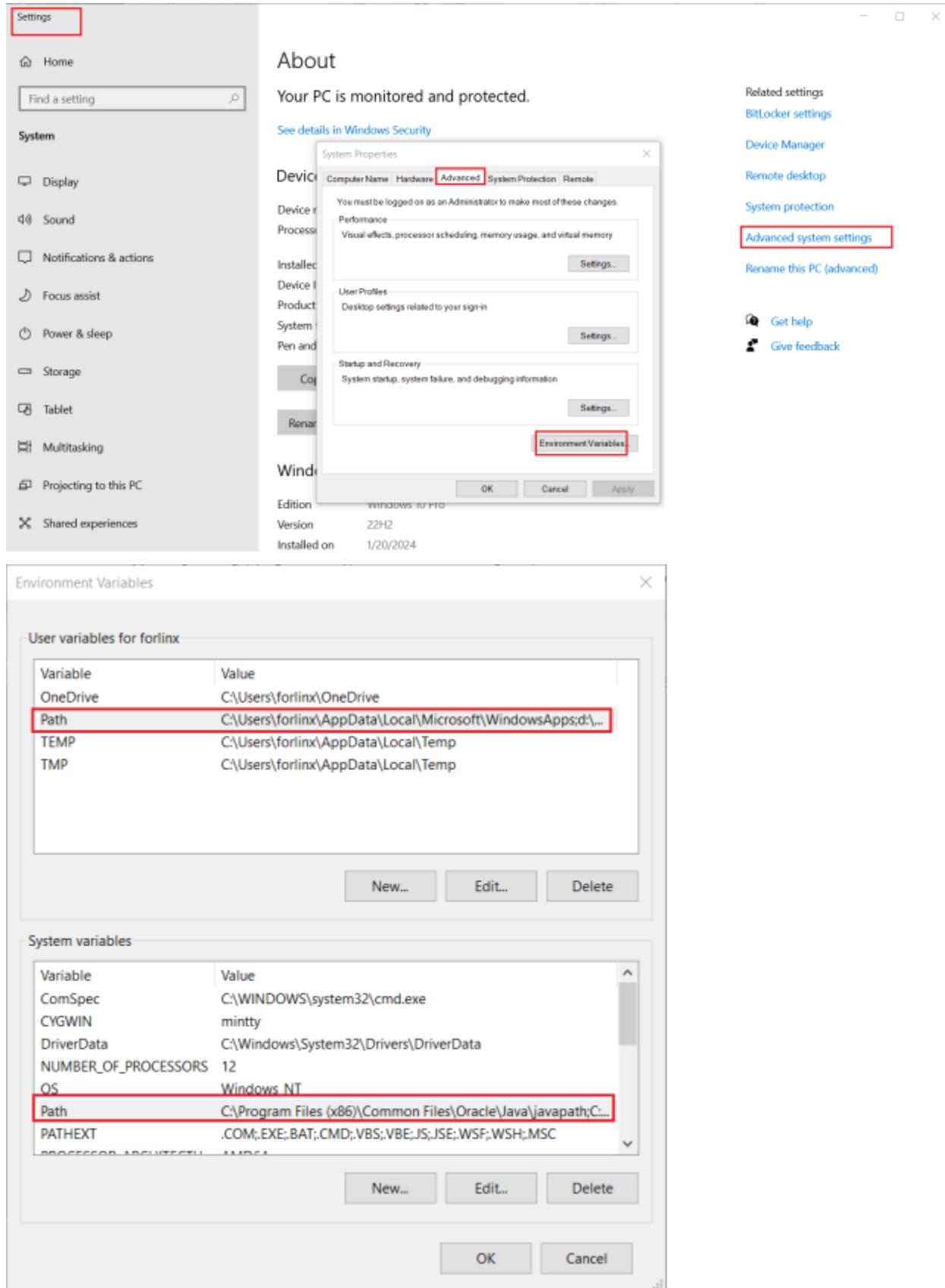
After the installation is complete, you need to add the JDK command to the Path environment variable. Use the following method to add the path of the JDK command to the Path environment variable:

- 1) Right-click on "My Computer" -> Properties, then select the "Advanced system settings" option on the left navigation;
- 2) Click the "Environment Variables" option in the lower right corner;
- 3) In "System Variables", find the Path environment variable, and double-click it; according to the actual installation path to set the java environment variable; the default installation adds the following content "C:\Program Files\Java\jdk1.8.0_211\bin" (depending on the customer's actual installation path);
- 4) Click "OK" to complete the environment variable setting.

The following is a screenshot of the win7 system:



The following is a screenshot of the win10 system:



- 5) Check that the installation was successful.

After completing the above settings, restart the computer, open the command prompt tool in the DOS window of the computer, and enter:

```
javac -version
```

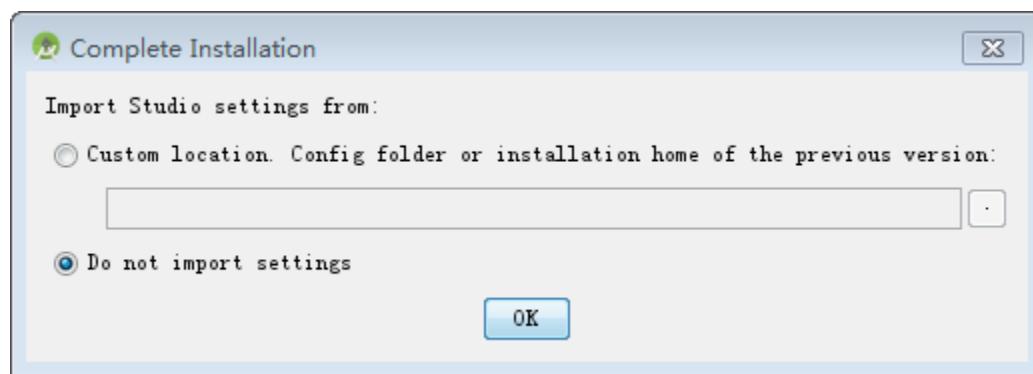
The screenshot shows a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window displays the following text:
Microsoft Windows [版本 6.1.7601]
版权所有 © 2009 Microsoft Corporation。保留所有权利。
C:\Users\160>javac -version
javac 1.8.0_211
C:\Users\160>

The correct display of the Java version indicates a successful installation.

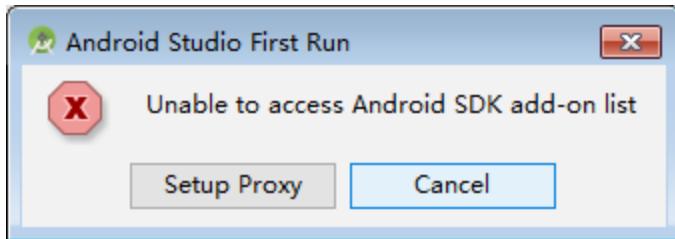
5.1.2 Android Studio Installation

Android Studio is a new development tool for Android launched by Google at I/O 2013. Please visit <http://www.android-studio.org/> to download. It is recommended to download version 3.5.2.

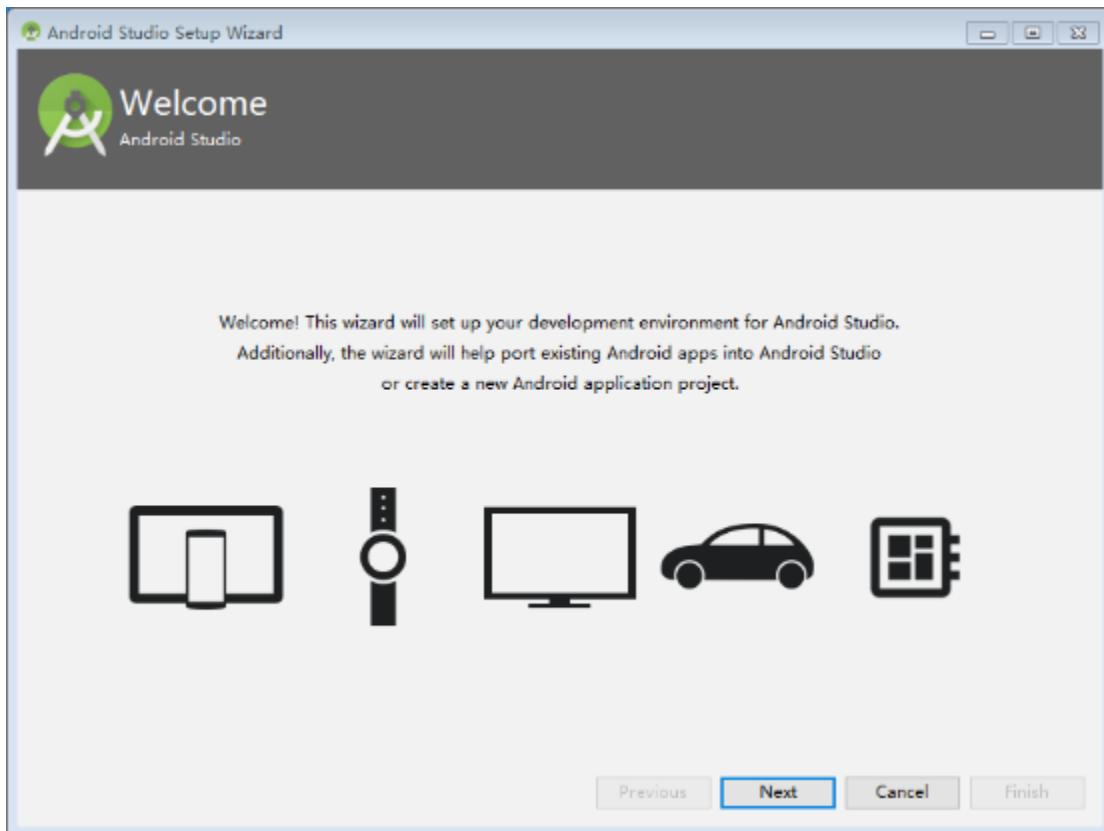
After downloading, follow the prompts to install it. When the installation is complete, the following figure appears:



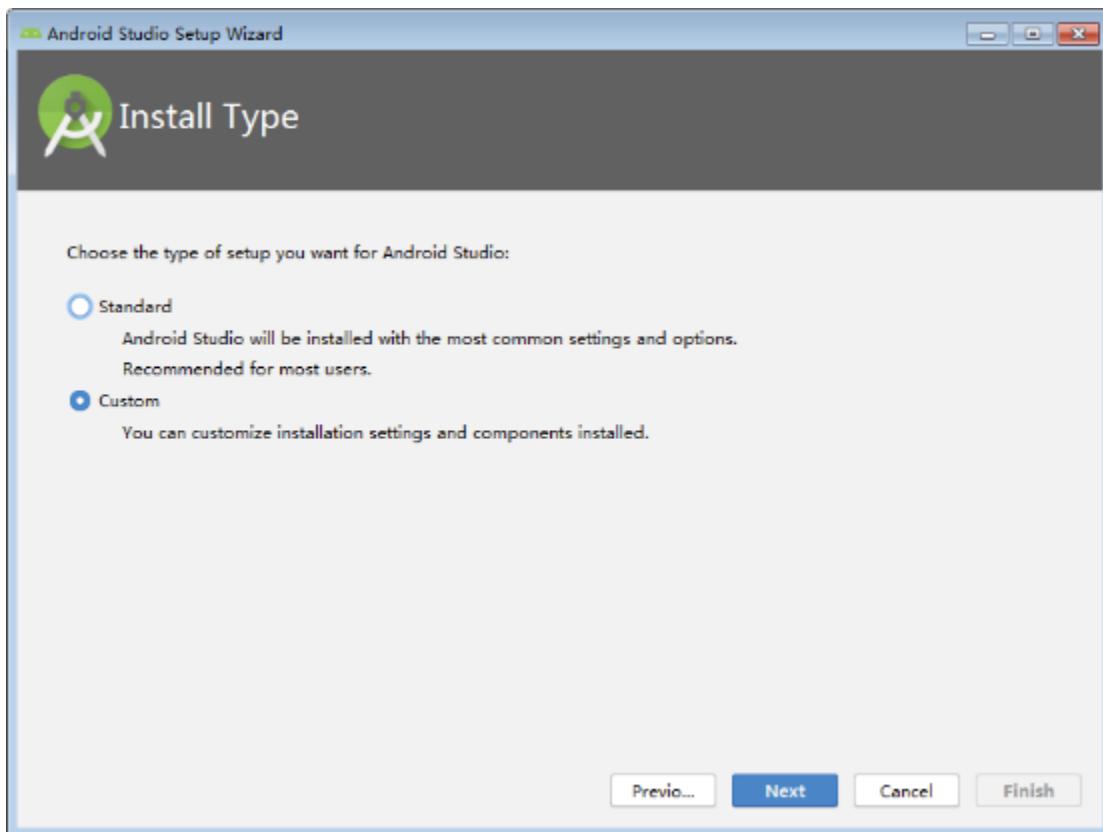
Select “Do not import settings”:



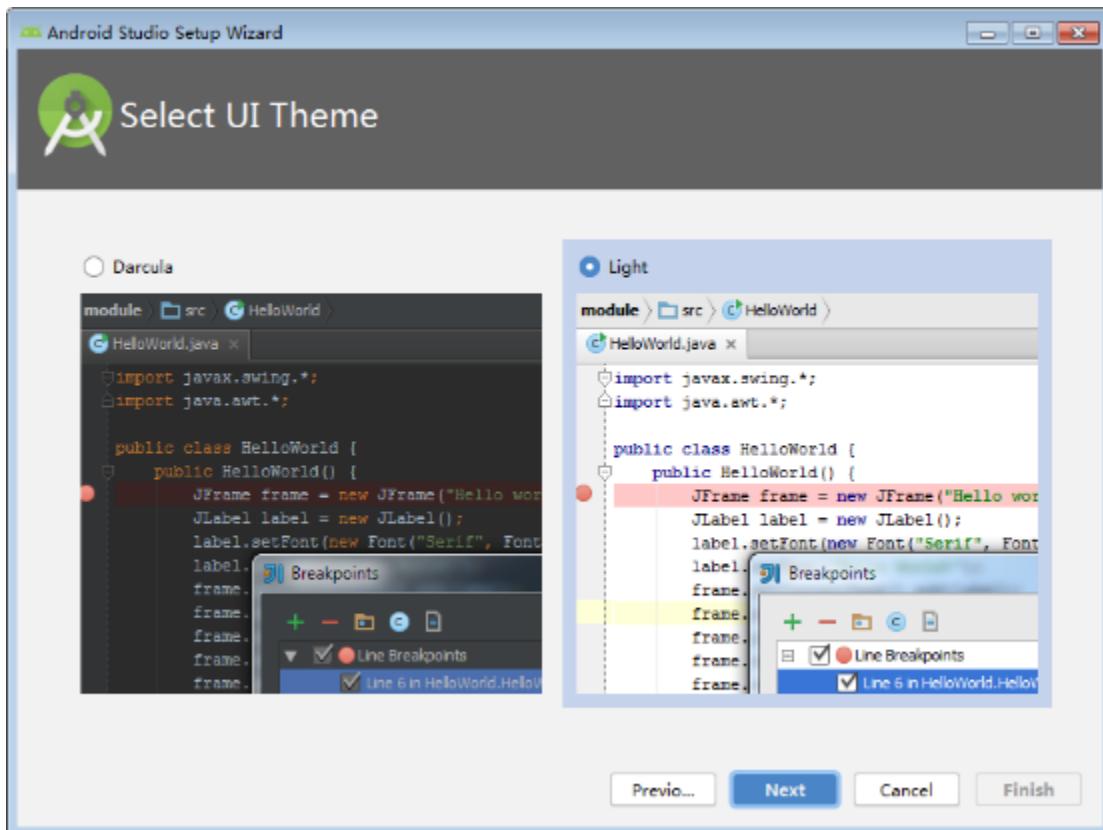
Select “Cancel”:



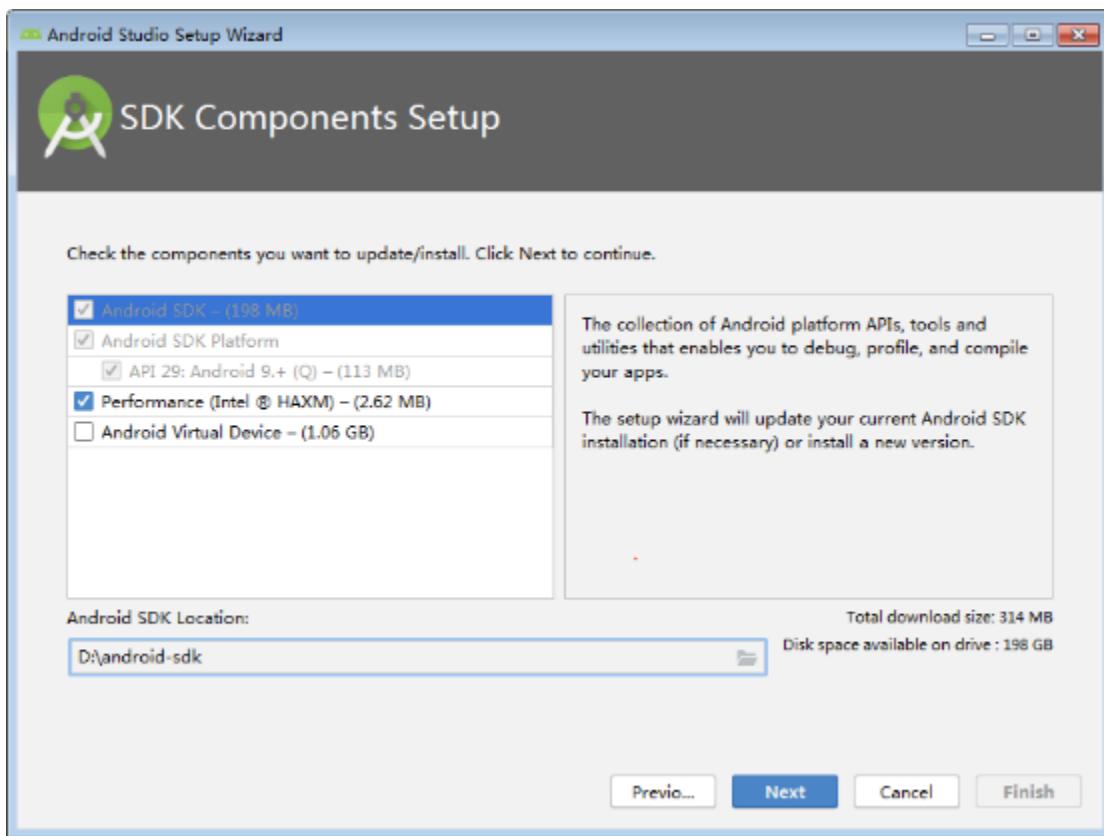
Click “Next” to go to the next step:



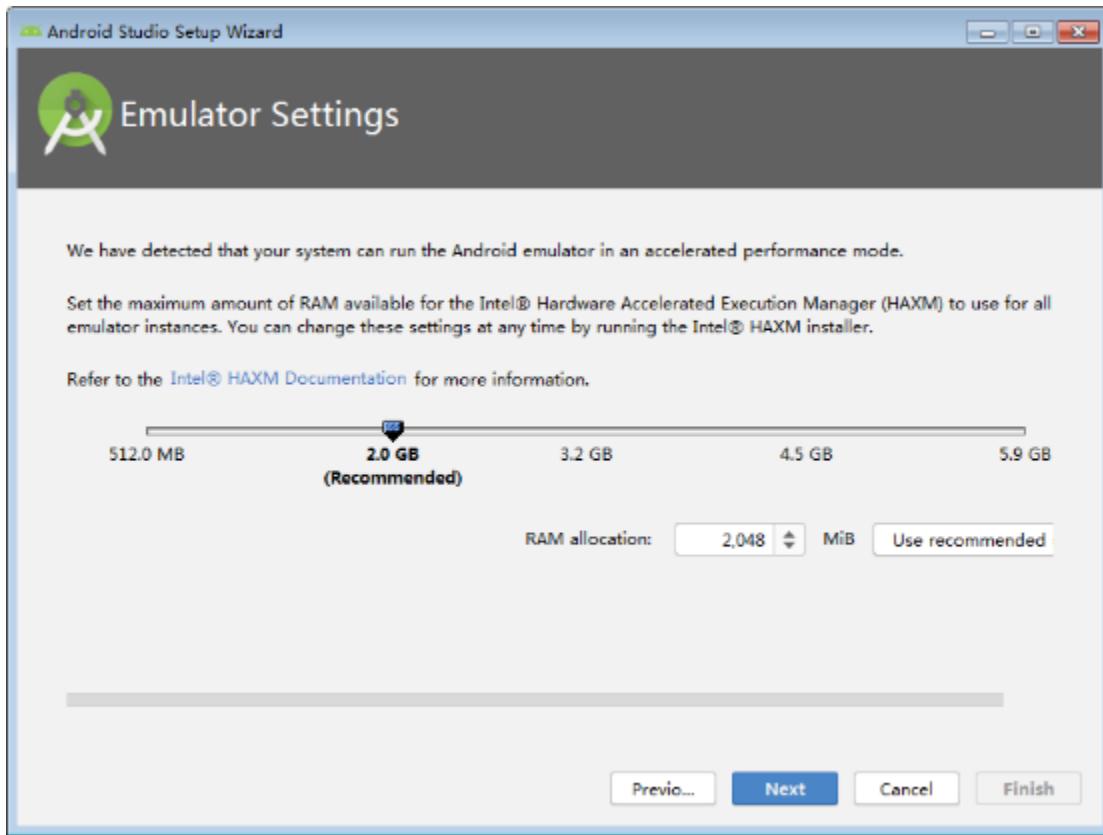
Select “custom” here:



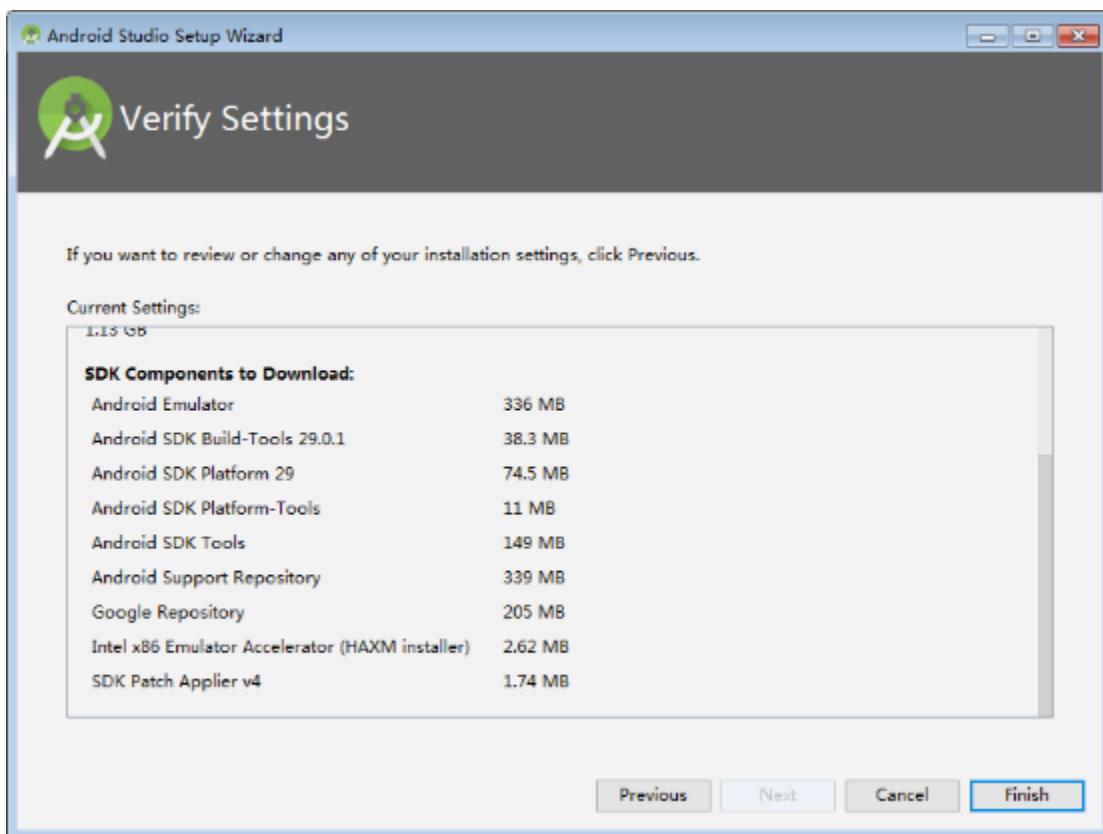
Choose a UI theme based on your personal preference:



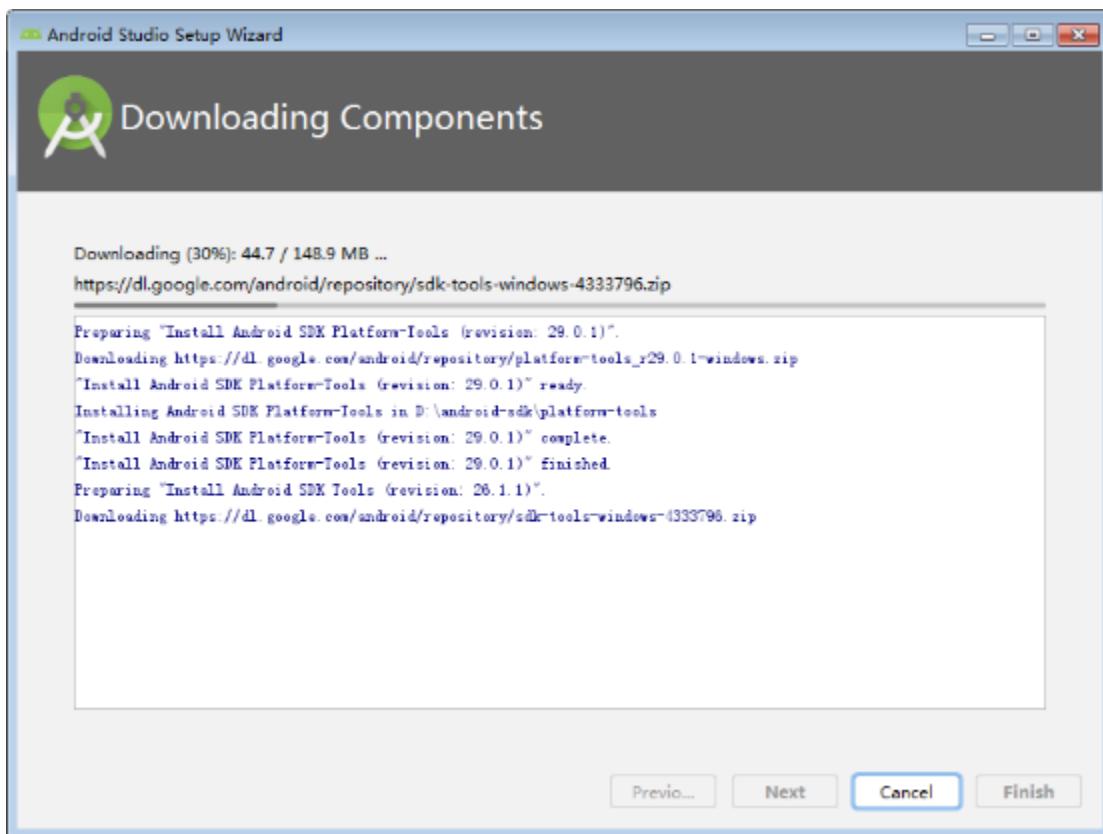
Select the installation path of Android SDK according to the actual situation:



Select the memory size for the emulator:



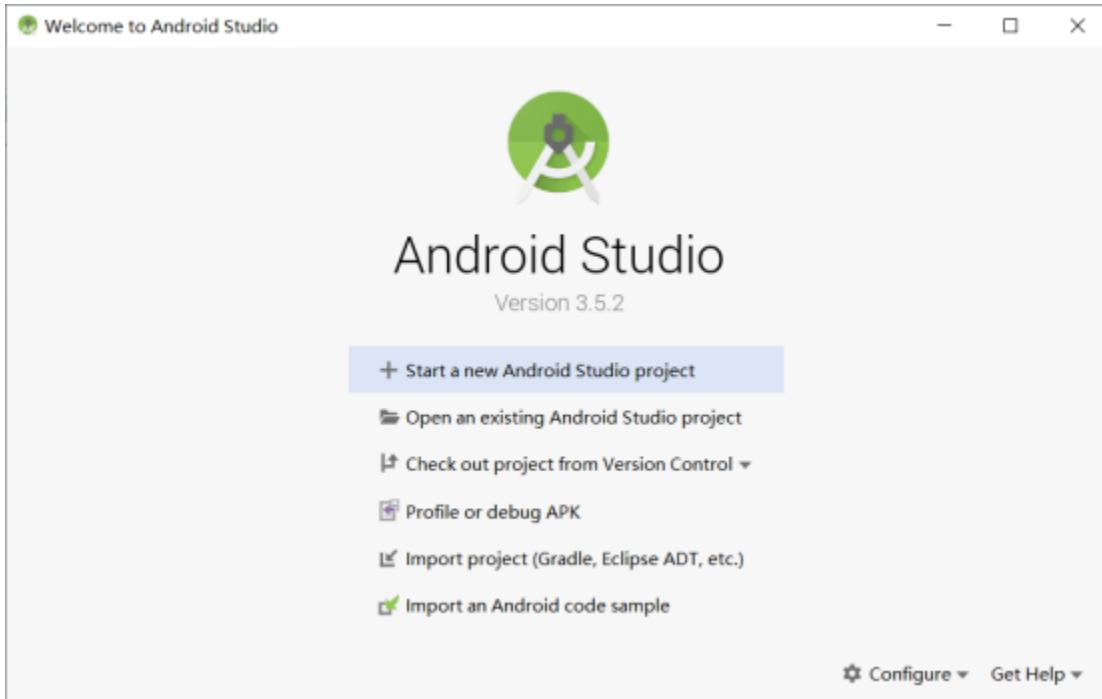
Click "Finish".



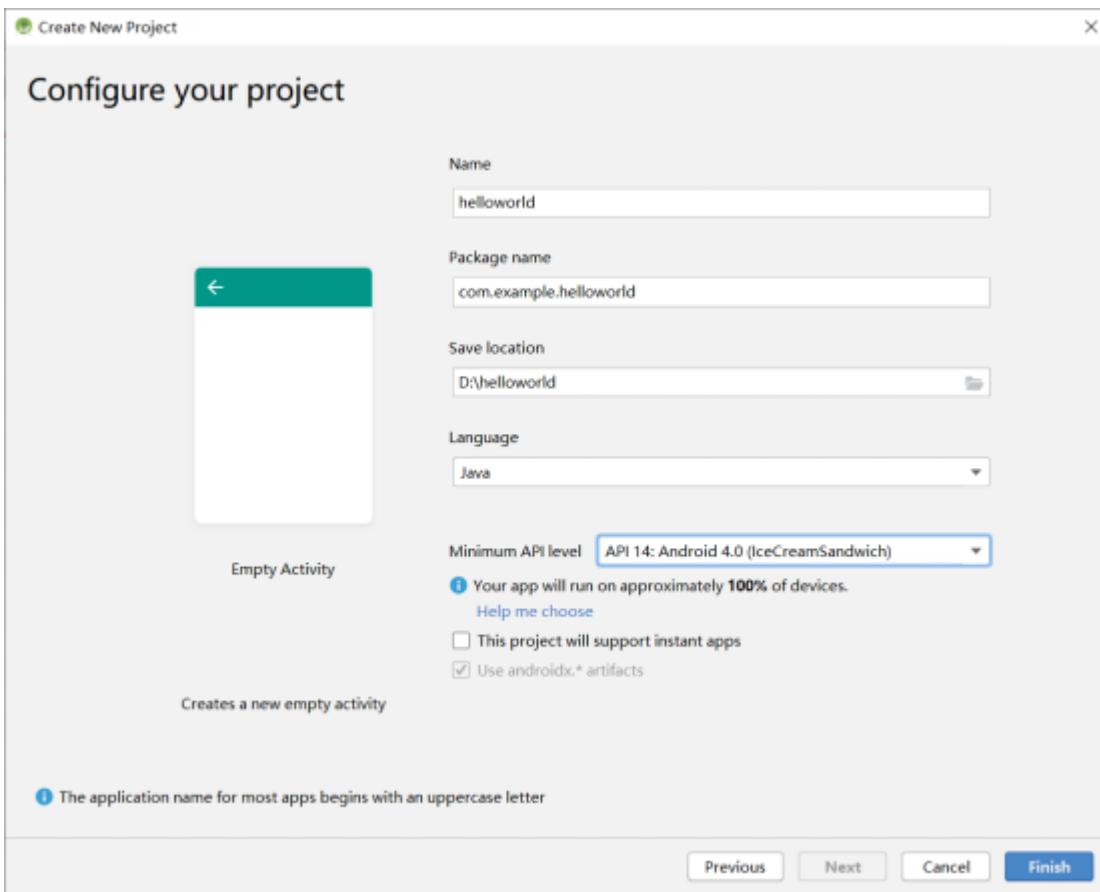
When the installation is complete, click “Finish” .

5.1.3 Helloworld Project Creation

1. Select “start a new android studio project” :



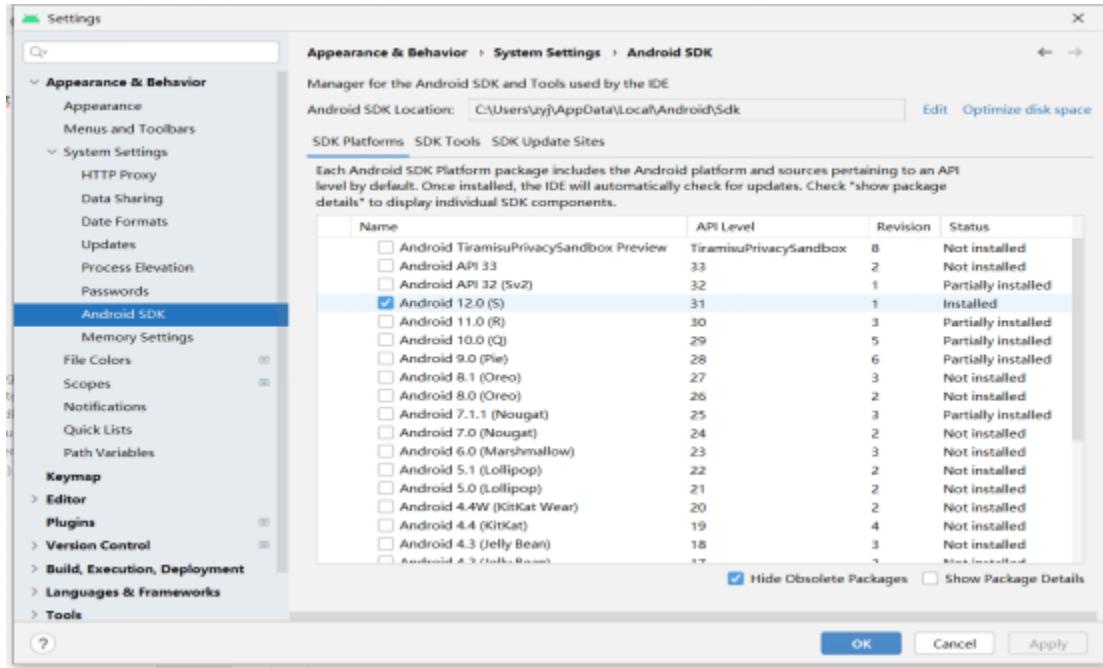
2. Choose “Empty Activity” :



Click “Next” to modify the project name and select the minimum supported android version:



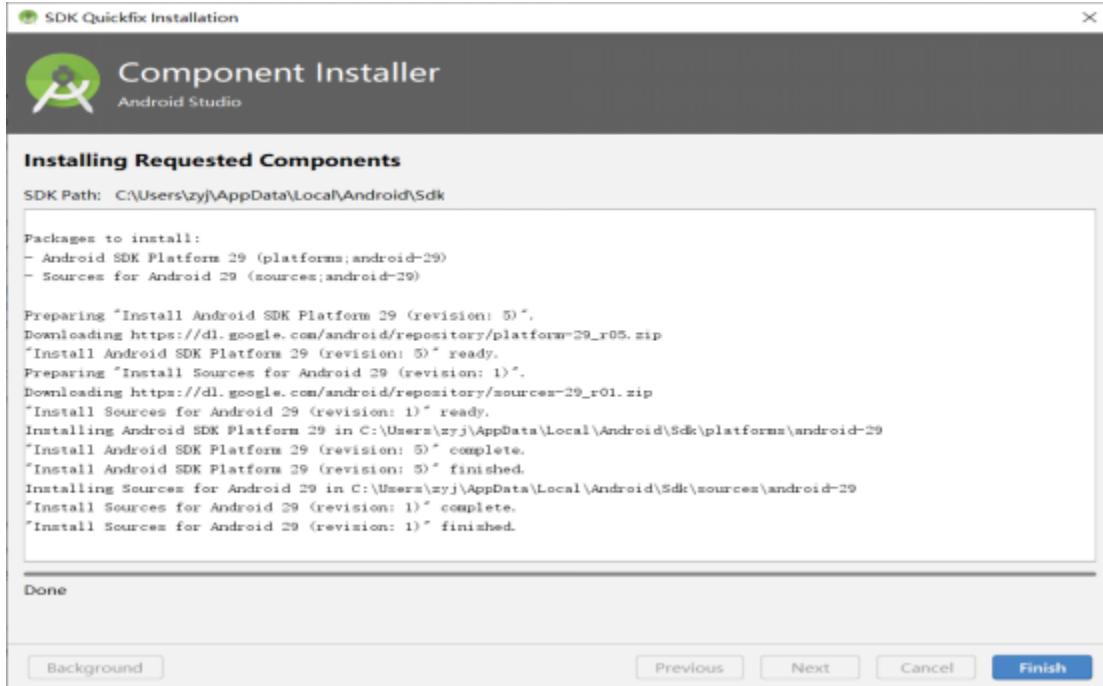
Click “Finish” .



Tools such as Gradle will be downloaded for the first use, so please be patient.

3. Install android 12.0 SDK

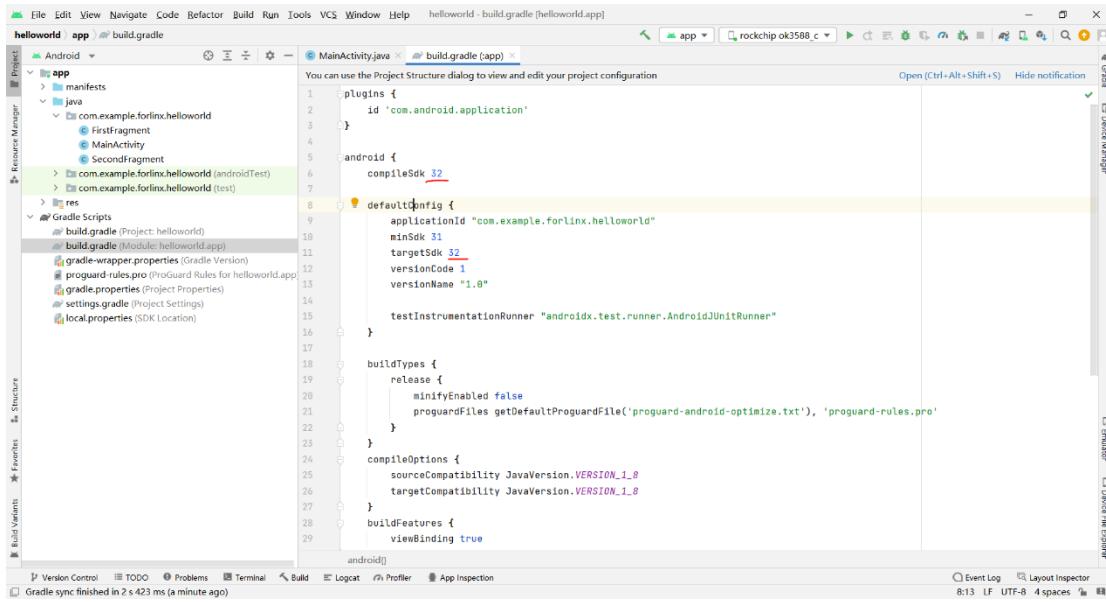
Click “File” -> “settings” to search SDK and open the interface as shown in the figure below:



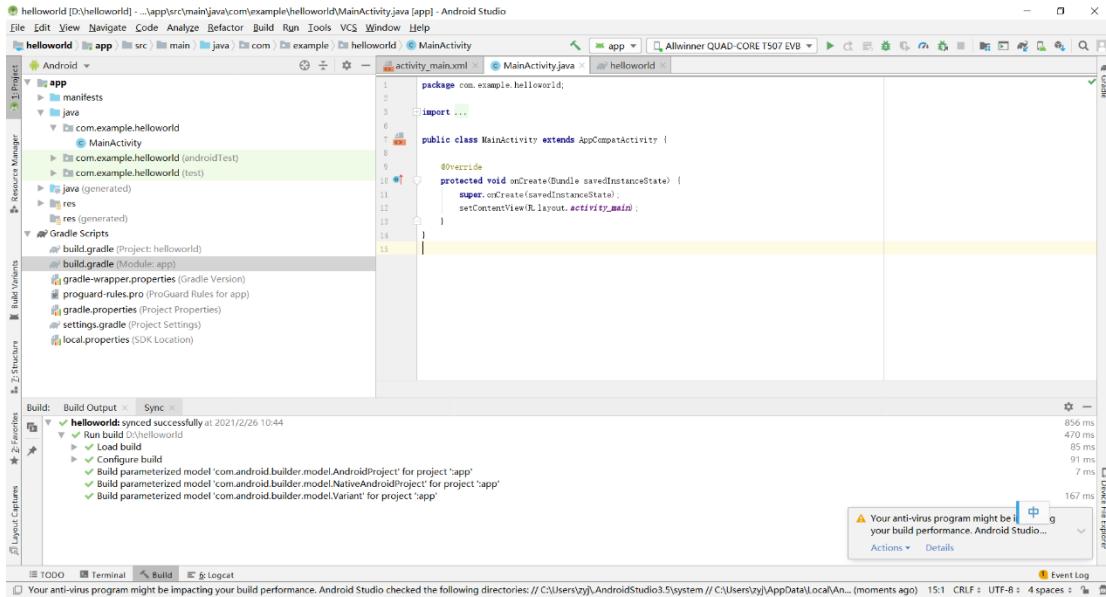
Check “Android 12.0” and click “OK” to install.

4. Modify “build.gradle” :

Modify all 32 in the file to 31, and press Ctrl + s to save the modified file.



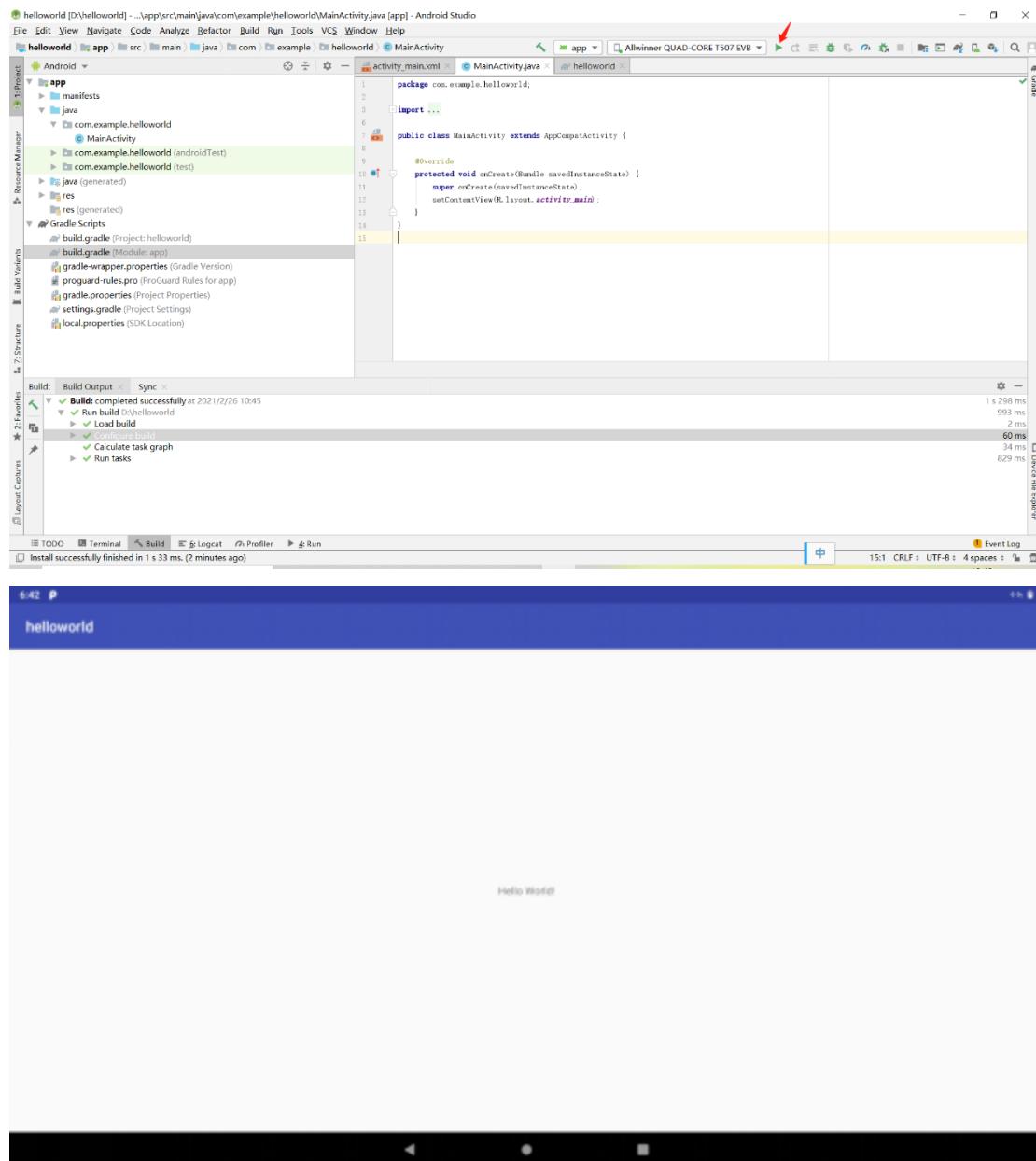
5. Compilation



Click “build” -> “Make Project” to recompile.

6. Run

After compiling, link the OTG cable to the USB port of your PC and click the green triangle icon in the menu bar.



5.2 Apk Platform Signature

Note: If there is no OpenSSL command in your windows system, please go to <http://slproweb.com/products/Win32OpenSSL.html> to down load and install it, and set environment variables.

In the Android platform, SELinux divides apps into three types, including untrusted_app which has no platform signature and system privileges, platform_app which has platform signature and no system privileges, and system_app which has platform signature and system privileges.

This chapter will introduce how to sign the apk and obtain system permissions.

1. Make signature document.

Copy build/target/product/security/platform.x509.pem and build/target/product/security/platform.pk8 in Android system to windows.

Open a command line window to execute:

```
openssl pkcs8 -in platform.pk8 -inform DER -outform PEM -out shared.priv.pem --nocrypt
openssl pkcs12 -export -in platform.x509.pem -inkey shared.priv.pem -out shared.pk12 -
    -name androiddebugkey
```

Enter password: android

```
keytool -importkeystore -deststorepass android -destkeypass android -destkeystore=
    -debug.keystore -srckeystore shared.pk12 -srcstoretype PKCS12 -srcstorepass android -
    -alias androiddebugkey
```

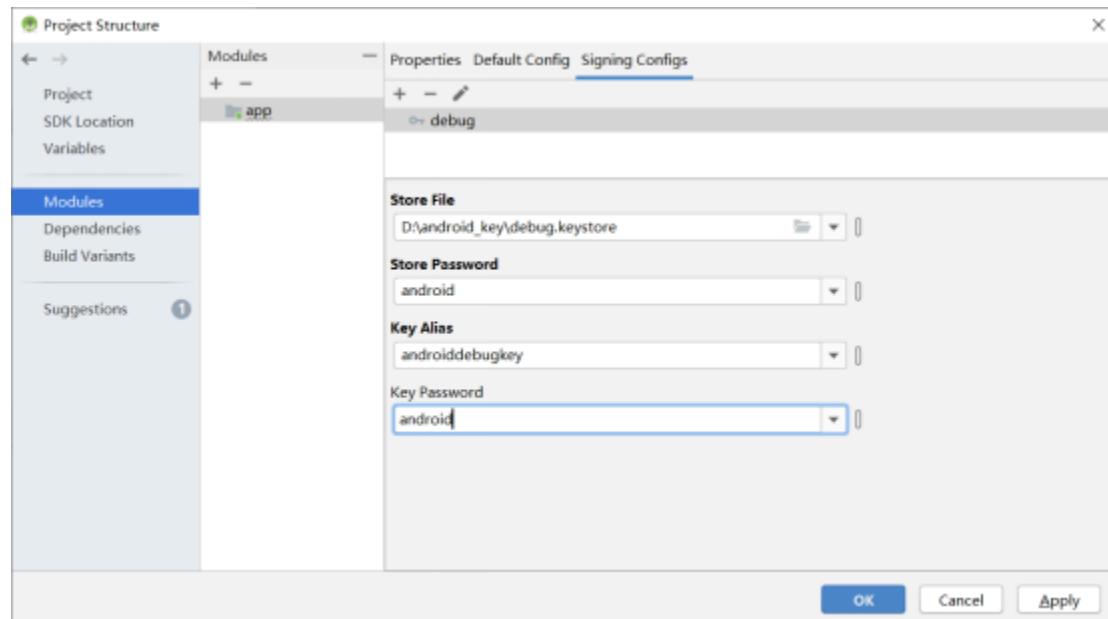
The key-alias and password can be modified to other contents as required. Save the signature file debug_keystore files to your usual directory.

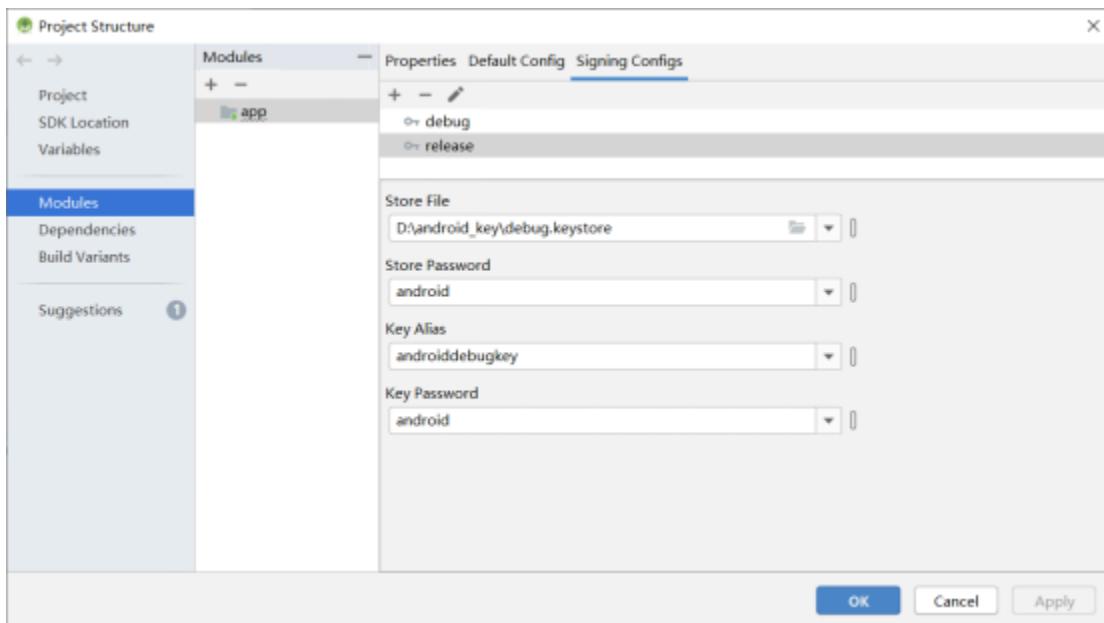
2. Set android studio

Open any android studio project and add the shared UID in the AndroidManifest.xml, for example:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.forlinx.helloworld"
    android:sharedUserId="android.uid.system">
```

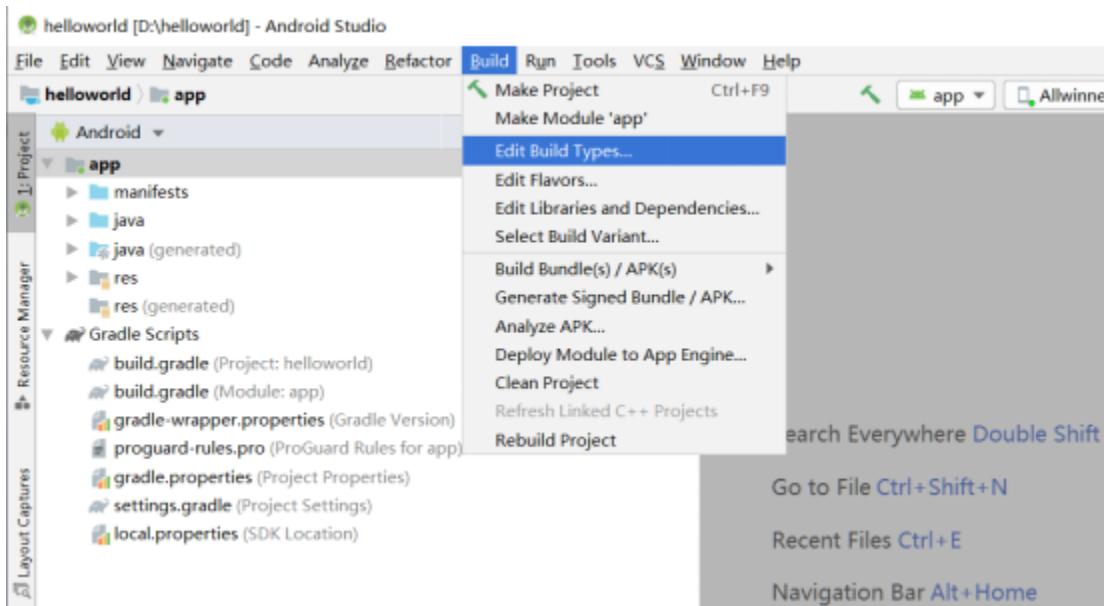
Click “File” -> “project structure” :

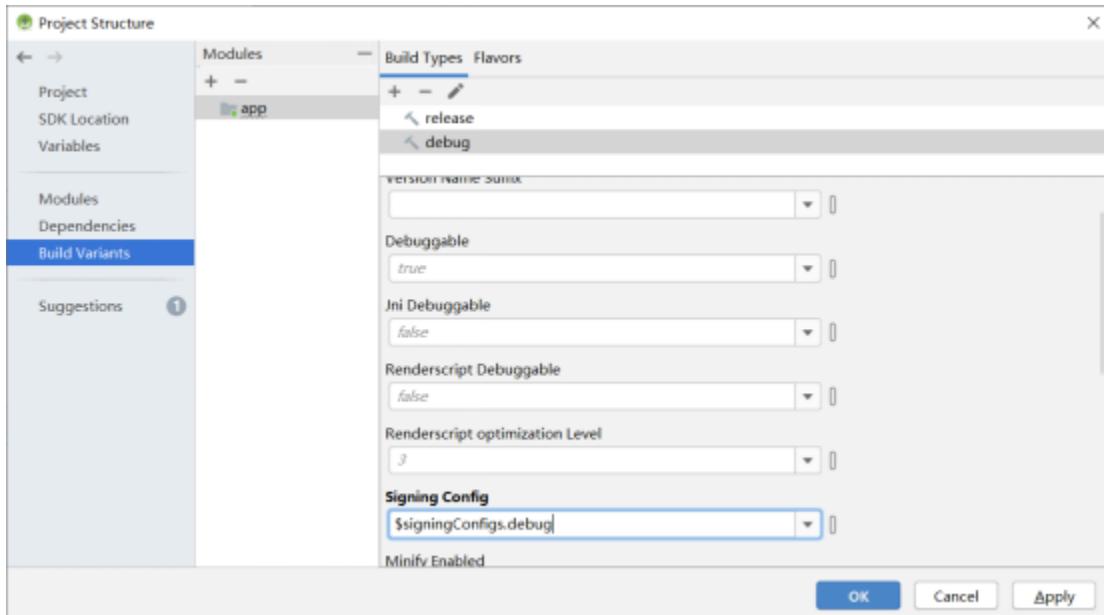
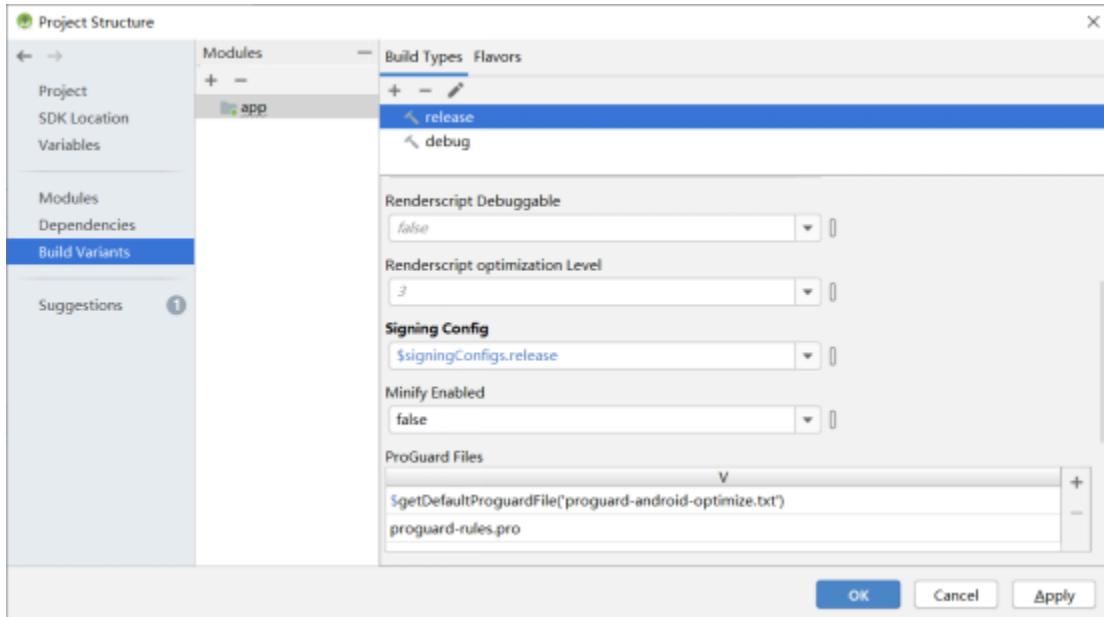




Click “OK” .

To add the signature configuration for the debug and release versions, click “Build” in the Android Studio menu bar and “Edit Build Type…” .





Click “OK” when the modification is completed.

Click the run button “ ” of android studio to start the app. Enter in the serial port:

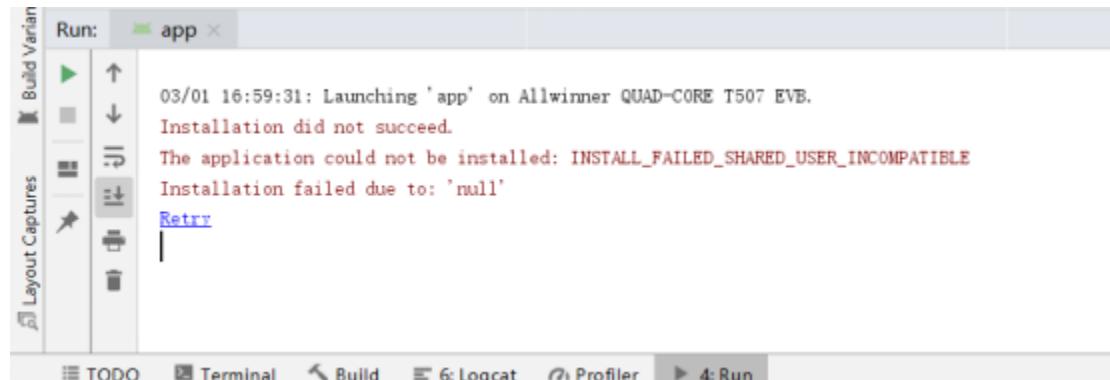
ps —AZ

```

u:r:kernel:s0          root      3222     2      0      0      0      0 s^
[worker/3:2]
u:r:platform_app:s0:c512,c768 u0_a52      3231   2283 1176640 192080 0      0 s
com.android.systemui
u:r:priv_app:s0:c512,c768    u0_a48      3465   2283 1062552 126028 0      0 s
com.android.launcher3
u:r:radio:s0              radio      3542   2283 1054664 96968 0      0 s
com.android.phone
u:r:kernel:s0              root      3643     2      0      0      0      0 s
[worker/3:3]
u:r:untrusted_app_27:s0:c60,c256,c512,c768 u0_a60 3834 2283 1035172 92956 0      0 s
com.android.inputmethod.latin
u:r:kernel:s0              root      3971     2      0      0      0      0 s
[kbase_event]
u:r:awdisplay:s0            system     4438     1    13060 4192 0      0 s
displayservice
u:r:secure_element:s0:c44,c260,c512,c768 secure_element 4515 2283 1021728 68600 0      0 s
com.android.secure_element
u:r:kernel:s0              root      4852     2      0      0      0      0 s
[kbase_event]
u:r:platform_app:s0:c512,c768 u0_a13      9759   2283 1029364 80008 0      0 s
com.android.permissioncontroller
u:r:system app:s0            system     9855   2283 1042596 107860 0      0 s
com.example.helloworld
u:r:kernel:s0              root      10006    2      0      0      0      0 s
[worker/3:3]
u:r:shell:s0               shell      16157   2198  9700 3144 0      0 R
ps
console:/ $
```

Confirm whether your app has become system _ app.

If the app prints the following error when running:



Solution: Uninstall the APP in the system, restart the board, and reinstall and run it.

5.3 Pre - installation Method of APK in System

1. Create a new directory in Android system:

```
forlinx@ubuntu:~/work/OK3588-android-source$ mkdir packages/apps/helloworld
```

Copy the apk that needs to be pre-installed (no signature is required) to the directory. Take the helloworld.apk as an example:

```
forlinx@ubuntu:~/work/OK3588-android-source$ cp helloworld.apk packages/apps/
↳helloworld
```

2. Create a new Android. mk at packages/apps/helloworld

```
forlinx@ubuntu:~/work/OK3588-android-source$ vi packages/apps/helloworld/Android.mk
```

Add the following:

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := helloworld
LOCAL_SRC_FILES := helloworld.apk
LOCAL_MODULE_CLASS := APPS
LOCAL_MODULE_SUFFIX := .apk
LOCAL_BUILT_MODULE_STEM := package.apk
LOCAL_CERTIFICATE := platform
LOCAL_DEX_PREOPT := false
LOCAL_PRIVILEGED_MODULE := true
include $(BUILD_PREBUILT)
```

3. Modify at the same time device/rockchip/rk3588/ok3588_c/ok3588_c.mk

```
forlinx@ubuntu:~/work/OK3588-android-source$ vi device/rockchip/rk3588/ok3588_c/
↳ ok3588_c.mk
```

Add the followings:

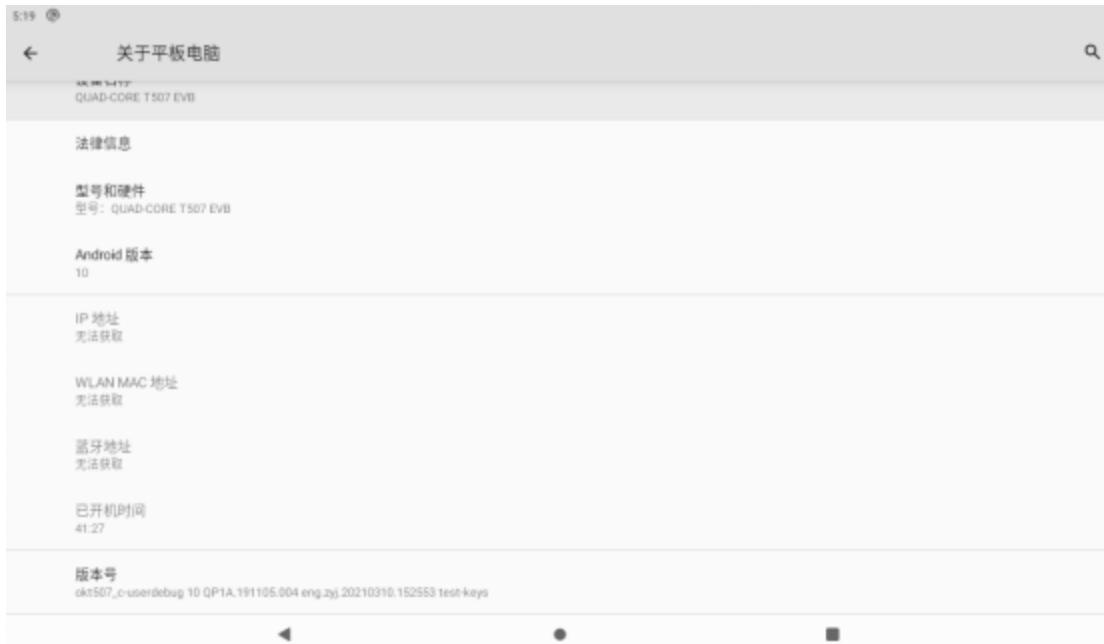
```
PRODUCT_PACKAGES += \
helloworld
```

4. Recompile the image

5.4 Method for App to Obtain Root Permission

The OK3588 platform supports app invocation of “su” to gain root privileges. By default, this feature is disabled. The method to enable it is as follows:

1. Open the Settings and click “About Tablet PC” to enter the interface as follows:

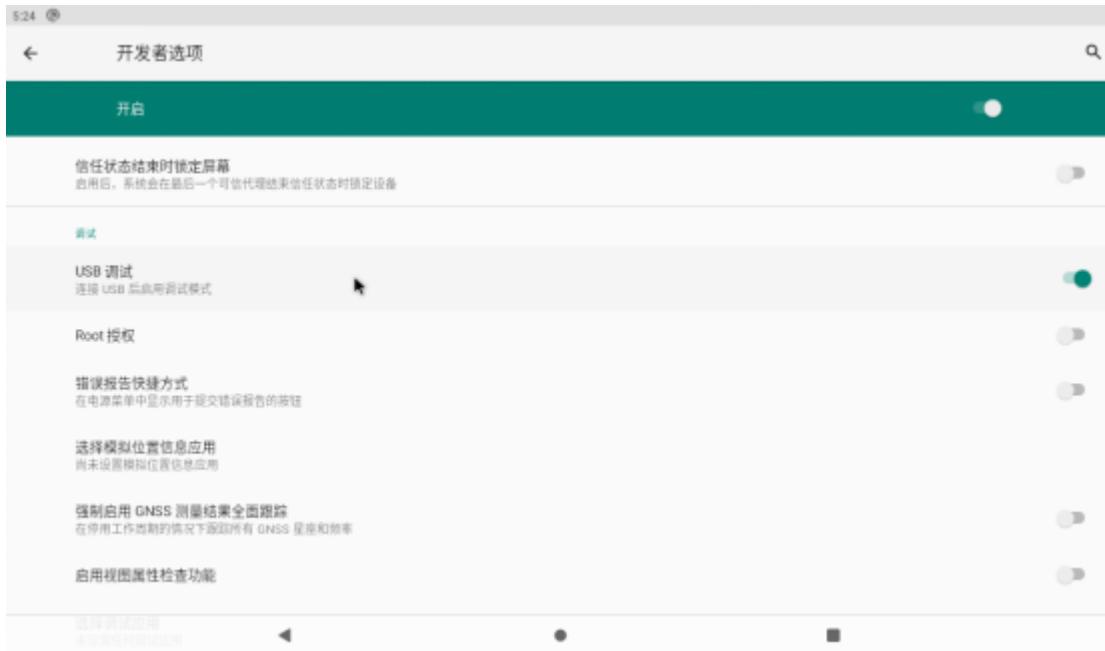


Click the “version number” continuously, and there will be a prompt at the bottom, “You are now in developer mode” .

2. Click “System”, “Advanced”, “Developer Options” :



3. Turn on the root authorization option:



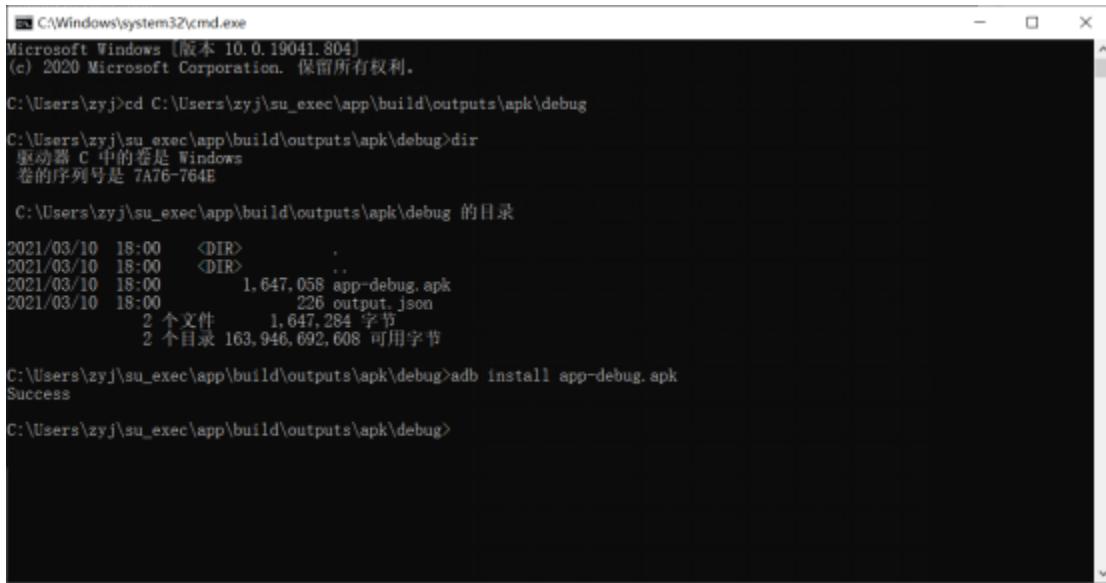
4. Restart the system.

5. App obtains root permission test.

Path: OK3588-C (Android) User Profile \ Android \ Testers \ su _ exec. Zip

Copy su_exec.zip to any directory in Windows and unzip it. Connect the OK3588 board to the PC via USB using a USB OTG cable. Navigate to the su_exec\app\build\outputs\apk\debug directory using the Windows Command Prompt

and install the apk software package using the adb install command.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19041.804]
(c) 2020 Microsoft Corporation. 保留所有权利。
C:\Users\zyj>cd C:\Users\zyj\su_exec\app\build\outputs\apk\debug
C:\Users\zyj\su_exec\app\build\outputs\apk\debug>dir
驱动器 C 中的卷是 Windows
卷的序列号是 7A76-764E

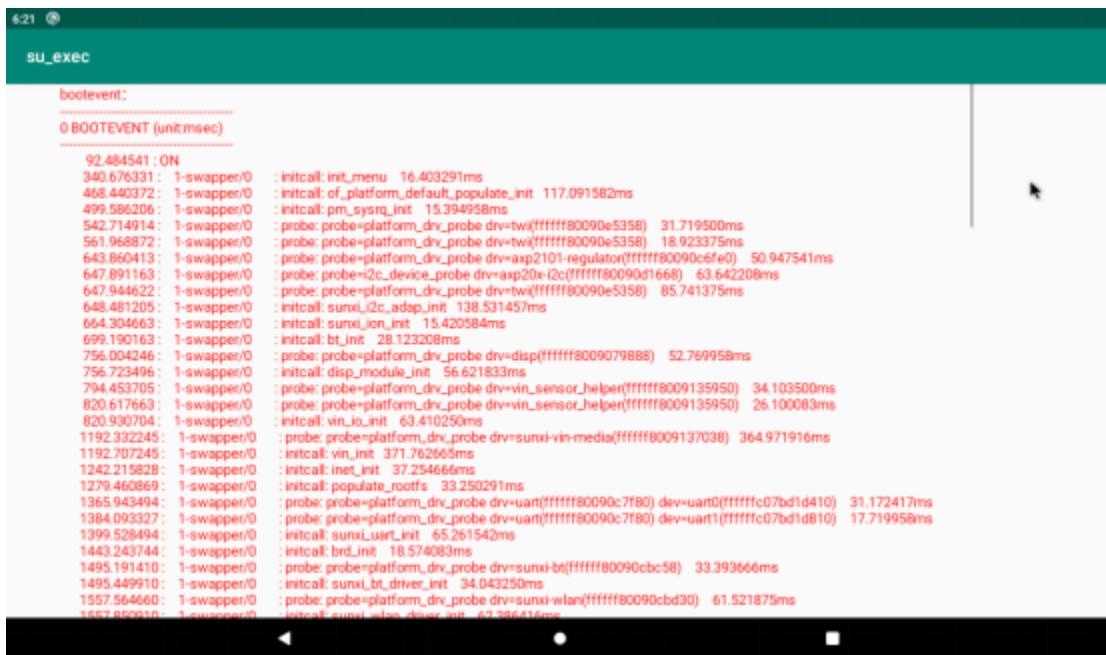
C:\Users\zyj\su_exec\app\build\outputs\apk\debug 的目录

2021/03/10 18:00 <DIR> .
2021/03/10 18:00 <DIR> ..
2021/03/10 18:00 1,647,058 app-debug.apk
2021/03/10 18:00 226 output.json
2 个文件 1,647,284 字节
2 个目录 163,946,692,608 可用字节

C:\Users\zyj\su_exec\app\build\outputs\apk\debug>adb install app-debug.apk
Success

C:\Users\zyj\su_exec\app\build\outputs\apk\debug>
```

After the installation is completed, click “su _ exec” “on the application interface.



Users can test if they have permission to view by typing cat /proc/bootevent in the OK3588 terminal.

**CHAPTER
THREE**

APPLICATION NOTE

CHAPTER
FOUR

CONTACT US

If you have any questions, please contact us at:

- **Contact Sales:** sales@forlinx.com
 - **Technical Support:** support@forlinx.com
 - **Website:** www.forlinx.net
-

Thank you for choosing Forlinx Embedded. We're committed to supporting your development journey.
