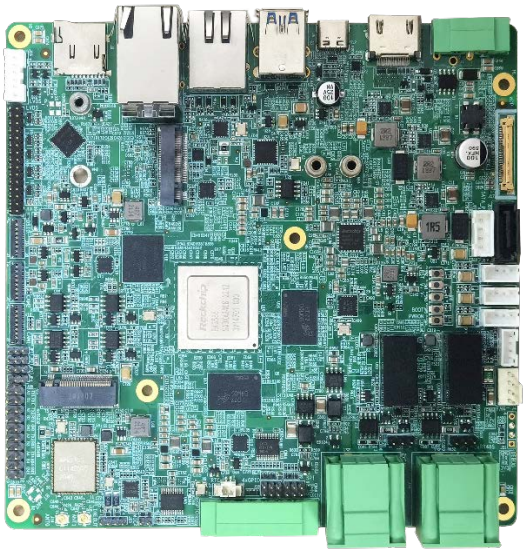
VT-SBC-3588 Single Board Computer



User Manual

Version: 1.5

© Vantron Technology, Inc. All rights reserved.

Revision History

No.	Version	Description	Date
1	V1.0	First release	Jun. 14, 2022
2	V1.1	Updated pinout description of certain connectors	Aug. 30, 2022
3	V1.2	Added instructions for serial port debugging	May 6, 2023
4	V1.3	Differentiate the commands for serial debugging and GPIO debugging per operating systems	Oct. 17, 2023
5	V1.4	Aligned the serial port debugging commands of Debian system to the other systems	Dec. 8, 2023
6	V1.5	Separated the serial port debugging commands of Ubuntu and Debian systems from the Android system	Dec. 15, 2023

Table of Contents

Foreword

Thank you for purchasing VT-SBC-3588 single board computer ("the Board" or "the Product"). This manual intends to provide guidance and assistance necessary on setting up, operating or maintaining the Product. Please read this manual and make sure you understand the functionality of the Product before putting it into use.

Intended Users

This manual is intended for:

- Embedded software developer
- Custom development software engineer
- Other technically qualified personnel

Copyright

Chengdu Vantron Technology Co., Ltd. ("Vantron") reserves all rights of this manual, including the right to change the content, form, product features, and specifications contained herein at any time without prior notice. An up-to-date version of this manual is available at www.vantrontech.com.

The trademarks in this manual, registered or not, are properties of their respective owners. Under no circumstances shall any part of this user manual be copied, reproduced, translated, or sold. This manual is not intended to be altered or used for other purposes unless otherwise permitted in writing by Vantron. Vantron reserves the right of all publicly released copies of this manual.

Disclaimer

While all information contained herein has been carefully checked to assure its accuracy in technical details and typography, Vantron does not assume any responsibility resulting from any error or features of this manual, nor from improper uses of this manual or the software.

It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without notice.

Technical Support and Assistance

Should you have any question about the Product that is not covered in this manual, contact your sales representative for solution. Please include the following information in your question:

- Product name and PO number;
- Complete description of the problem;
- Error message you received, if any.

Vantron Technology, Inc.

Address: 48434 Milmont Drive, Fremont, CA 94538
Tel: (650) 422-3128
Email: sales@vantrontech.com

Symbology

This manual uses the following signs to prompt users to pay special attention to relevant information.

	Caution for latent damage to system or harm to personnel
	Attention to important information or regulations

General Safety Instructions

The Product is supposed be installed by knowledgeable, skilled persons familiar with local and/or international electrical codes and regulations. For your safety and prevention of damage to the Product, please read and observe carefully the following safety instructions prior to installation and operation. Keep this manual well for future reference.

- Do not disassemble or otherwise modify the Product. Such action may cause heat generation, ignition, electronic shock, or other damages including human injury, and may void your warranty.
- Keep the Product away from heat source, such as heater, heat dissipater, or engine casing.
- Do not insert foreign materials into any opening of the Product as it may cause the Product to malfunction or burn out.
- To ensure proper functioning and prevent overheating of the Product, do not cover or block the ventilation holes of the Product.
- Follow the installation instructions with the installation tools provided or recommended.
- The use or placement of the operation tools shall comply with the code of practice of such tools to avoid short circuit of the Product.
- Cut off the power before inspection of the Product to avoid human injury or product damage.

Precautions for Power Cables and Accessories

⚠ Use proper power source only. Make sure the supply voltage falls within the specified range.

⚠ Place the cables properly at places without extrusion hazards.

⚠ There is a coin cell battery for powering the RTC. Therefore, please avoid short circuit of the battery during transportation or operation at high temperatures.

⚠ Cleaning instructions:

- Power off before cleaning the Product
- Do not use spray detergent
- Clean with a damp cloth
- Do not try to clean exposed electronic components unless with a dust collector

⚠ Power off and contact Vantron technical support engineer in case of the following faults:

- The Product is damaged
- The temperature is excessively high
- Fault is still not solved after troubleshooting according to this manual

⚠ Do not use in combustible and explosive environment:

- Keep away from combustible and explosive environment
- Keep away from all energized circuits
- Unauthorized removal of the enclosure from the device is not allowed. Do not change components unless the power cable is unplugged. In some cases, the device may still have residual voltage even if the power cable is unplugged. Therefore, it is a must to remove and fully discharge the device before replacement of the components.

CHAPTER 1 INTRODUCTION

1.1 Product Overview

Vantron VT-SBC-3588 Single Board Computer is powered by Rockchip latest flagship RK3588 AIoT chipset that is equipped with an 8-core 64-bit CPU, an ARM Mali-G610 MP4 quad-core GPU, and a built-in AI acceleration NPU, capable of providing 6 TOPS computing power and supporting mainstream deep learning frameworks. With the development of the technology, there definitely will be a rising demand for AI-based products from the industrial control market, including but not limited to industrial robots, automated control, drones, etc., and VT-SBC-3588 comes into being.

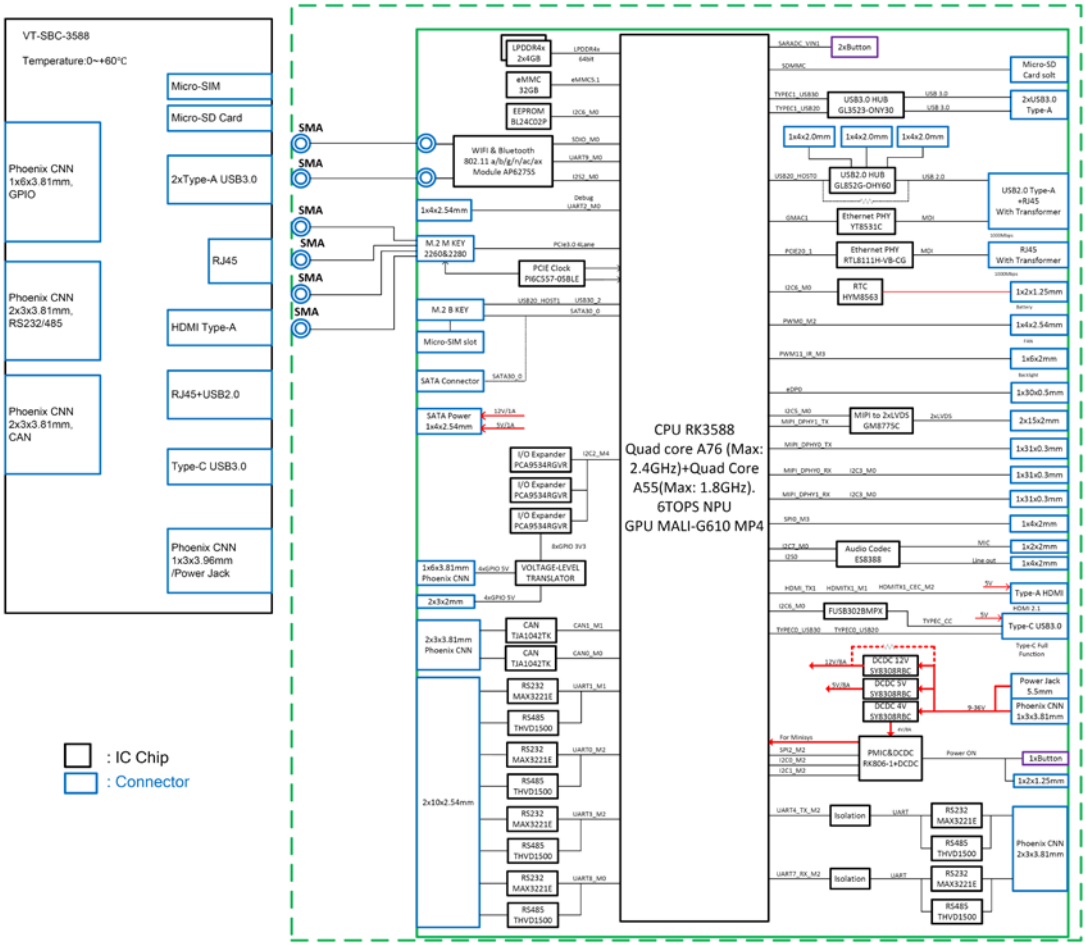
The single board computer offers two Gigabit Ethernet ports, supports 2.4GHz/5GHz Wi-Fi 6 and Bluetooth 5.0, and provides an M.2 slot for 4G/5G expansion to keep communication uninterrupted. It also supports 8K video decoding and encoding to deliver optimized display performance.

Since the single board computer provides rich interfaces, a wide range of peripherals can be connected for extended applications like ARM PC, edge computing, cloud server, smart NVR, and other fields. Moreover, the different operating systems provide a stable and secure system environment for users.

1.2 Terminology

Terminology	Description
NC	No connection
VCC	Voltage common collector
GND	Ground
/	Active low signal
+	Positive of difference signal
-	Negative of difference signal
I	Input
O	Output
I/O	Input/output
P	Power or ground
A	Analog
OD	Open drain
CMOS	3.3 V CMOS
LVCMOS	Low Voltage CMOS
LVTTL	Low Voltage TTL
3.3V	3.3 V signal level
5V	5V signal level
USB	5V tolerant signal
PCIe	PCI Express signal, not 3.3 V tolerant
MMC	Multimedia Card

1.3 Block Diagram



1.4 Specifications

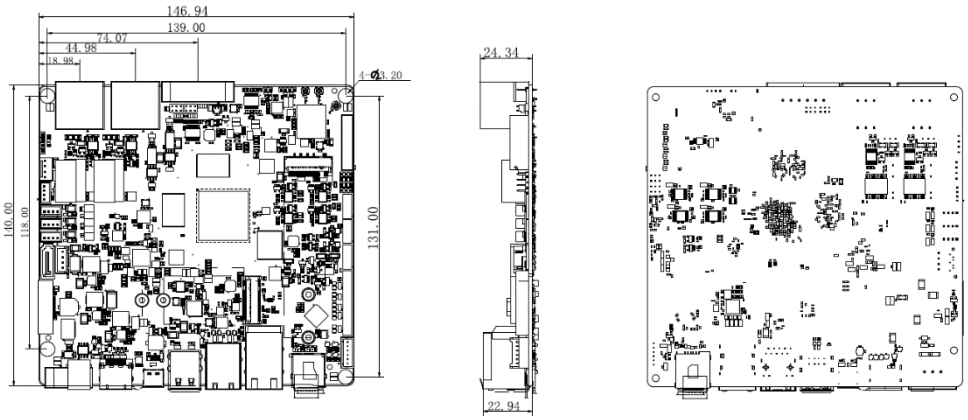
VT-SBC-3588			
System	CPU	RK3588 Quad-core Cortex-A76 + Quad-core Cortex-A55, Max. 2.4GHz	
	GPU	ARM Mali-G610 MP4, Max. 1GHz	
	NPU	6 TOPS	
	Memory	8GB LPDDR4 (Optional: 16GB)	
	Storage	3 2GB eMMC V5.1, up to 128GB	1 x Micro SD slot
		SSD supported by an M.2 M-Key/M.2 B-Key (256GB ~ 1TB)	
Communication	Ethernet	2 x RJ45, 1000Mbps	
	4G/5G	Supported (expansion by an M.2 B-Key)	
	Wi-Fi & Bluetooth	Wi-Fi 802.11 a/b/g/n/ac/ax + BT 5.0	
Media	Display	1 x HDMI 2.1 (4096 x 2160 @60Hz) 1 x Dual-channel LVDS (1920 x 1200 @60Hz)	1 x eDP (1920 x 1080 @60Hz) 1 x MIPI DSI (1920 x 1080 @60Hz)
	Camera	2 x MIPI CSI	
	Audio	1 x Mic in connector	1 x Line out connector
I/Os	USB	2 x USB 3.0 Host, Type-A 1 x USB 2.0 Host, Type-A	1 x USB Type-C OTG 3 x USB 2.0 Host connector
	Serial port	2 x RS232/RS485 on the Phoenix terminal	4 x RS232/RS485 connector
	TTL UART	1 x TTL UART, for debugging	
	Fan	1 x CPU fan connector	
	SIM slot	1 x Micro SIM slot	
	RTC	Supported	
	Watchdog	Supported	
	Button	1 x Power button connector	
Expansion	M.2 slot	1 x M.2 M-Key (2260/2280), PCIe 3.0 x 4, for SSD	
		1 x M.2 B-Key (2242/3052), USB3.0/SATA3.0, for 4G/5G/SSD	
	GPIO	4 x GPIO on the Phoenix terminal	3 x GPIO connector
	SPI	1 x SPI	
	CAN	2 x CAN on the Phoenix terminal	
Software	OS	Debian, Android, Ubuntu	
	Language	English (default), Chinese	
	Device management platform	BlueSphere MDM (Android device only)	
	OTA tool	BlueSphere OTA	
Power	Input	12V/5A, 24V/3A	1 x Power terminal
Mechanical	Dimensions	146.94mm x 140mm x 24.34mm	
Environment condition	Temperature	Operating: 0℃ ~ +60℃	Storage: -40℃ ~ +85℃
	Humidity	RH 0~95% (non-condensing)	
	Certification	FCC, CCC	

1.5 Operating system

VT-SBC-3588 supports Debian, Android and Ubuntu operating systems.

1.6 Mechanical Dimensions

- 146.94mm x 140mm x 24.34mm



1.7 Power Supply and Consumption

VT-SBC-3588 works with 12V/24V DC power supply.

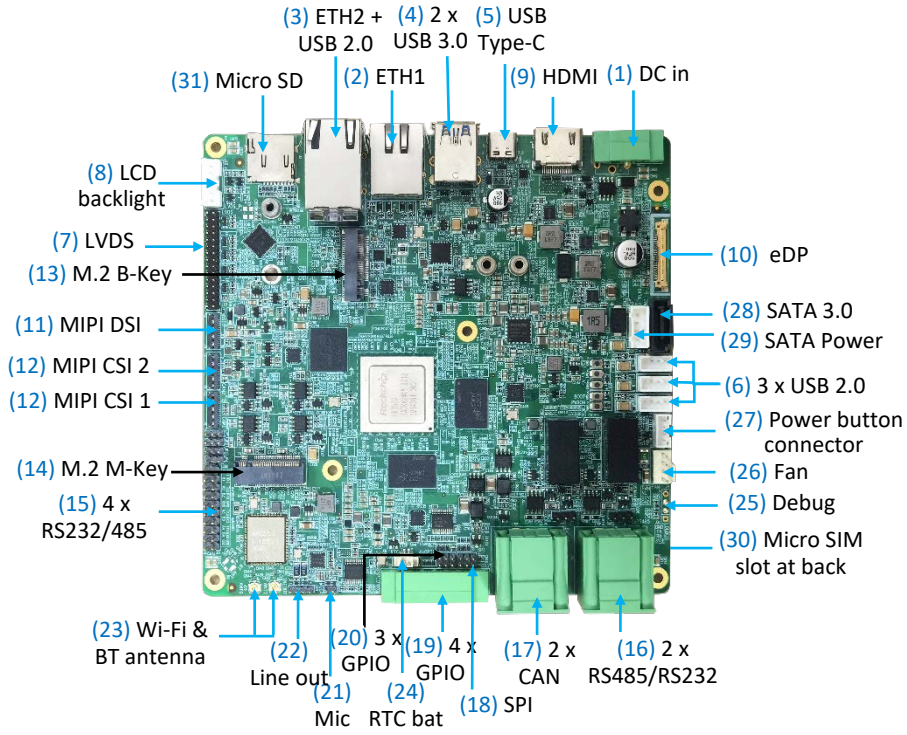
The power consumption of the Board is 30W at the maximum. It should be pointed out that power consumption is largely determined by the RAM, storage capacity, and other configurations of the board.

1.8 Environmental Specifications

VT-SBC-3588 works at a temperature ranging from 0℃ to +60℃ and at relative humidity of no more than RH 96% for non-condensing purpose. It is designed to be stored at a temperature ranging from -40°C to +85°C.

CHAPTER 2 CONNECTOR DESCRIPTION

2.1 Product Layout



The board I/Os will be described in detail in 2.4 Connectors and Jumpers following the sequencing numbers provided here.

2.2 Memory and Storage

2.2.1 LPDDR4 RAM

VT-SBC-3588 is equipped with an 8GB LPDDR4 RAM by default, and users also have the option of 16GB RAM.

2.2.2 eMMC flash

VT-SBC-3588 provides an eMMC 5.1 flash up to 128 GB, and the default capacity is 32GB. It is used as the default boot and storage device.

2.2.3 Storage expansion

VT-SBC-3588 implements a Micro SD slot for expansion of the storage capacity. In addition, it also offers an M.2 M-Key and an M.2 B-Key that support SSD expansion.

2.3 Identification of Pin 1

Unless otherwise stated, pin 1 of a connector is seated on a square pad that is different from the round pads used for other pins. Sometimes, pin 1 is next to a trigonal mark on the board. When there are two rows of pins on a connector, the row with pin 1 is composed of odd numbers and the other is composed of even numbers.

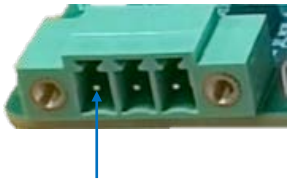
Usually, there will be numbers or marks next to the pins of a connector on the board to indicate the pinouts.

2.4 Connectors and Jumpers

This section is going to brief on the connectors/jumpers on the Board with corresponsive pinout description.

2.4.1 J1 Power terminal (1)

VT-SBC-3588 supports 12V/24V DC power input (12V 5A recommended) and implements a 3-pin power terminal (1 x 3 x 3.81mm) to supply power for the Board.



Pinout description of the power terminal:

Pin	Name	Type	Description
1	GND	P	Ground
2	-VDC	P	DC-IN POWER -
3	+VDC	P	DC-IN POWER +

2.4.2 J22 Ethernet jack (2)

VT-SBC-3588 implements two RJ45 Ethernet jacks each with two LED indicators, green for activity indication and yellow for link indication. The ports support 10/100/1000 Mbps transmission rate. J22 is a standalone Ethernet jack, while the other is integrated with a USB 2.0 Type-A interface.

2.4.3 J21 Ethernet jack + USB 2.0 (3)

The other Ethernet jack is stacked in layers with a USB 2.0 Type-A interface, and the maximum output of the USB 2.0 Type-A interface is 5V/0.5A.

2.4.4 U27 USB 3.0 Type-A (4)

VT-SBC-3588 offers two USB 3.0 Type-A interfaces for connecting peripherals. The maximum output of each interface is 5V/1A.

2.4.5 J6 USB Type-C (5)

VT-SBC-3588 implements a USB Type-C interface, supporting USB 3.0 OTG and DP 1.4 video output, with a maximum output of 5V/1.5A.

2.4.6 J7/J8/J9 USB 2.0 Host connectors (6)

VT-SBC-3588 has three on-board USB 2.0 connectors (1 x 4 x 2.0mm) set in the host mode by default. The maximum output of each connector is 5V/0.5A.

Pin 1

Pinout description of J7:

Pin	Name	Type	Description
1	HUB_HOST_1	P	5V power output, current limit 0.5A
2	HUB_HOST1_DM	I/O	USB Host1 DATA-
3	HUB_HOST1_DP	I/O	USB Host1 DATA+
4	GND	P	Ground

Pinout description of J8:

Pin	Name	Type	Description
1	HUB_HOST_2	P	5V power output, current limit 0.5A
2	HUB_HOST2_DM	I/O	USB Host2 DATA-
3	HUB_HOST2_DP	I/O	USB Host2 DATA+
4	GND	P	Ground

Pinout description of J9:

Pin	Name	Type	Description
1	HUB_HOST_3	P	5V power output, current limit 0.5A
2	HUB_HOST3_DM	I/O	USB Host3 DATA-
3	HUB_HOST3_DP	I/O	USB Host3 DATA+
4	GND	P	Ground

2.4.7 J13 LVDS (7)

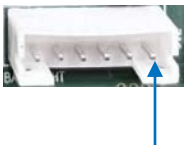
VT-SBC-3588 offers a dual-channel LVDS to connect high-definition displays (resolution up to 1920 x 1080 @60Hz). It is recommended that you use LVDS shielded twisted pairs to connect the interface for improved transmission reliability.

Pinout description:

Pin	Name	Type	Description
1	VCC5V0_LCD_LVDS	P	5V power output
2	VCC5V0_LCD_LVDS	P	5V power output
3	VCC5V0_LCD_LVDS	P	5V power output
4	GND	P	Ground
5	GND	P	Ground
6	GND	P	Ground
7	TXA0N	O	Channel A LVDS data 0 differential output -
8	TXA0P	O	Channel A LVDS data 0 differential output +
9	TXA1N	O	Channel A LVDS data 1 differential output -
10	TXA1P	O	Channel A LVDS data 1 differential output +
11	TXA2N	O	Channel A LVDS data 2 differential output -
12	TXA2P	O	Channel A LVDS data 2 differential output +
13	GND	P	Ground
14	GND	P	Ground
15	TXACN	O	Channel A LVDS clock differential output -
16	TXACP	O	Channel A LVDS clock differential output +
17	TXA3N	O	Channel A LVDS data 3 differential output -
18	TXA3P	O	Channel A LVDS data 3 differential output -
19	TXB0N	O	Channel B LVDS data 0 differential output -
20	TXB0P	O	Channel B LVDS data 0 differential output +
21	TXB1N	O	Channel B LVDS data 1 differential output -
22	TXB1P	O	Channel B LVDS data 1 differential output +
23	TXB2N	O	Channel B LVDS data 2 differential output -
24	TXB2P	O	Channel B LVDS data 2 differential output +
25	GND	P	Ground
26	GND	P	Ground
27	TXBCN	O	Channel B LVDS clock differential output -
28	TXBCP	O	Channel B LVDS clock differential output +
29	TXB3N	O	Channel B LVDS data 3 differential output -
30	TXB3P	O	Channel B LVDS data 3 differential output -

2.4.8 J15 Backlight connector (8)

VT-SBC-3588 implements a backlight connector (1 x 6 x 2.0mm) that is designed to connect a backlight to increase readability of the LCD in low light conditions.



Pin 1

Pinout description:

Pin	Name	Type	Description
1	VCC12V_LCD	P	12V power supply for backlight
2	VCC12V_LCD	P	12V power supply for backlight
3	PANEL_BKLTEN	O	LCD backlight power control output, 5V level
4	PANEL_BL_PWM	O	LCD backlight PWM control output, 5V level
5	GND	P	Ground
6	GND	P	Ground

2.4.9 J9 HDMI (9)

VT-SBC-3588 has a standard HDMI Type-A interface for image output with a resolution up to 4096 x 2160 @60Hz, and it supports HDMI 2.1 protocol.

2.4.10 J17 eDP (10)

VT-SBC-3588 implements an eDP interface (1 x 30 x 0.5mm) for image output with a resolution up to 4096 x 2160 @60Hz, and it supports HDMI 2.1 protocol. It is recommended that you use LVDS shielded twisted pairs to connect the interface for improved transmission reliability.

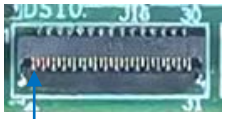
Pin 1

Refer to the next page for the pinout description of the interface.

Pin	Name	Type	Description
1	VCC3V3_EDP	P	3.3V power output
2	VCC3V3_EDP	P	3.3V power output
3	VCC3V3_EDP	P	3.3V power output
4	NC		
5	GND	P	Ground
6	eDP0_TX_D0N	O	eDP TX differential data lane 0 -
7	eDP0_TX_D0P	O	eDP TX differential data lane 0 +
8	GND	P	Ground
9	eDP0_TX_D1N	O	eDP TX differential data lane 1 -
10	eDP0_TX_D1P	O	eDP TX differential data lane 1+
11	GND	P	Ground
12	eDP0_TX_D2N	O	eDP TX differential data lane 2 -
13	eDP0_TX_D2P	O	eDP TX differential data lane 2 +
14	GND	P	Ground
15	eDP0_TX_D3N	O	eDP TX differential data lane 3 -
16	eDP0_TX_D3P	O	eDP TX differential data lane 3 +
17	GND	P	Ground
18	eDP_TX_AUXN		eDP auxiliary differential data lane -
19	eDP_TX_AUXP		eDP auxiliary differential data lane +
20	GND	P	Ground
21	GND	P	Ground
22	LCD_BL_EN_H	O	LCD backlight control output, 3.3V level
23	LCD_BL_PWM1	O	LCD backlight brightness control output, 3.3V level
24	GND	P	Ground
25	GND	P	Ground
26	GND	P	Ground
27	VCC12V_LCD_EDP	P	12V backlight power output
28	VCC12V_LCD_EDP	P	12V backlight power output
29	VCC12V_LCD_EDP	P	12V backlight power output
30	VCC12V_LCD_EDP	P	12V backlight power output

2.4.11 J16 MIPI DSI (11)

VT-SBC-3588 implements a MIPI DSI connector (1 x 31 x 0.3mm) for connecting displays, supporting a resolution up to 1920 x 1080 @60Hz.



Pin 1

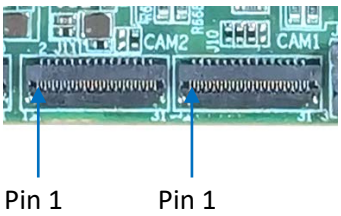
Pinout description:

Pin	Name	Type	Description
1	VCC_LED0_A_MIPI	P	LED +
2	VCC_LED0_A_MIPI	P	LED +
3	VCC_LED0_A_MIPI	P	LED +
4	NC		
5	VCC_LED0_K_MIPI	P	LED -
6	VCC_LED0_K_MIPI	P	LED -
7	VCC_LED0_K_MIPI	P	LED -
8	VCC_LED0_K_MIPI	P	LED-
9	GND	P	Ground
10	GND	P	Ground
11	MIPI_DPHY0_TX_D2P	O	MIPI DPHY0 TX differential data lane 2 +
12	MIPI_DPHY0_TX_D2N	O	MIPI DPHY0 TX differential data lane 2 -
13	GND	P	Ground
14	MIPI_DPHY0_TX_D1P	O	MIPI DPHY0 TX differential data lane 1 +
15	MIPI_DPHY0_TX_D1N	O	MIPI DPHY0 TX differential data lane 1 -
16	GND	P	Ground
17	MIPI_DPHY0_TX_CLKP	O	MIPI DPHY0 TX differential clock +
18	MIPI_DPHY0_TX_CLKN	I/O	MIPI DPHY0 TX differential clock -
19	GND	P	Ground
20	MIPI_DPHY0_TX_D0P	O	MIPI DPHY0 TX differential data lane 0 +
21	MIPI_DPHY0_TX_D0N	O	MIPI DPHY0 TX differential data lane 0 -
22	GND	P	Ground
23	MIPI_DPHY0_TX_D3P	O	MIPI DPHY0 TX differential data lane 3 +
24	MIPI_DPHY0_TX_D3N	O	MIPI DPHY0 TX differential data lane 3 -
25	GND	P	Ground
26	NC		
27	MIPI_RST_L	O	MIPI LCD reset output, 1.8V
28	NC		
29	VCC1V8_LCD	P	1.8V power output

30	VCC3V3_LCD	P	3.3V power output
31	VCC3V3_LCD	P	3.3V power output

2.4.12 J10/J11 MIPI CSI (12)

VT-SBC-3588 has two MIPI CSI connectors (1 x 31 x 0.3mm) for connecting cameras.



Pinout description of J10 (CAM 1):

Pin	Name	Type	Description
1	GND	P	Ground
2	MIPI_DPHY0_RX_D3N	I	MIPI DPHY0 RX differential data lane 3 -
3	MIPI_DPHY0_RX_D3P	I	MIPI DPHY0 RX differential data lane 3 +
4	GND	P	Ground
5	MIPI_DPHY0_RX_D2N	I	MIPI DPHY0 RX differential data lane 2 -
6	MIPI_DPHY0_RX_D2P	I	MIPI DPHY0 RX differential data lane 2 +
7	GND	P	Ground
8	MIPI_DPHY0_RX_D1N	I	MIPI DPHY0 RX differential data lane 1 -
9	MIPI_DPHY0_RX_D1P	I	MIPI DPHY0 RX differential data lane 1 +
10	GND	P	Ground
11	MIPI_DPHY0_RX_D0N	I	MIPI DPHY0 RX differential data lane 0 -
12	MIPI_DPHY0_RX_D0P	I	MIPI DPHY0 RX differential data lane 0 +
13	GND	P	Ground
14	MIPI_DPHY0_RX_CLKN	I	MIPI DPHY0 RX differential clock -
15	MIPI_DPHY0_RX_CLKP	I	MIPI DPHY0 RX differential clock +
16	GND	P	Ground
17	I2C3_SCL_M0_CAM1	O	I2C serial clock, voltage domain 1.8V
18	I2C3_SDA_M0_CAM1	I/O	I2C serial data, voltage domain 1.8V
19	MIPI_CAM1_RESET	O	CAM1 reset output, voltage domain 1.8V
20	MIPI_CAM1_PDN	O	CAM1 power down control output, voltage domain 1.8V
21	GND	P	Ground
22	MIPI_CAM1_MCLK	O	CAM1 master clock output, voltage domain 1.8V
23	GND	P	Ground
24	NC		
25	VCC_1V8_CAM1	P	1.8V power output

26	VCC_1V8_CAM1	P	1.8V power output
27	VCC_1V2_CAM1	P	1.2V power output, voltage level adjustment supported by hardware
28	VCC_2V8_AF_CAM1	P	2.8V motor power output
29	VCC_2V8_CAM1	P	2.8V power output
30	NC		
31	GND	P	Ground

Pinout description of J11 (CAM 2):

Pin	Name	Type	Description
1	GND	P	Ground
2	MIPI_DPHY1_RX_D3N	I	MIPI DPHY1 RX differential data lane 3 -
3	MIPI_DPHY1_RX_D3P	I	MIPI DPHY1 RX differential data lane 3 +
4	GND	P	Ground
5	MIPI_DPHY1_RX_D2N	I	MIPI DPHY1 RX differential data lane 2 -
6	MIPI_DPHY1_RX_D2P	I	MIPI DPHY1 RX differential data lane 2 +
7	GND	P	Ground
8	MIPI_DPHY1_RX_D1N	I	MIPI DPHY1 RX differential data lane 1 -
9	MIPI_DPHY1_RX_D1P	I	MIPI DPHY1 RX differential data lane 1 +
10	GND	P	Ground
11	MIPI_DPHY1_RX_D0N	I	MIPI DPHY1 RX differential data lane 0 -
12	MIPI_DPHY1_RX_D0P	I	MIPI DPHY1 RX differential data lane 0 +
13	GND	P	Ground
14	MIPI_DPHY1_RX_CLKN	I	MIPI DPHY1 RX differential clock -
15	MIPI_DPHY1_RX_CLKP	I	MIPI DPHY1 RX differential clock +
16	GND	P	Ground
17	I2C3_SCL_M0_CAM2	O	I2C serial clock, voltage domain 1.8V
18	I2C3_SDA_M0_CAM2	I/O	I2C serial data, voltage domain 1.8V
19	MIPI_CAM2_RESET	O	CAM2 reset output, voltage domain 1.8V
20	MIPI_CAM2_PDN	O	CAM2 power down control output, voltage domain 1.8V
21	GND	P	Ground
22	MIPI_CAM2_MCLK	O	CAM2 master clock output, voltage domain 1.8V
23	GND	P	Ground
24	NC		
25	VCC_1V8_CAM2	P	1.8V power output
26	VCC_1V8_CAM2	P	1.8V power output
27	VCC_1V2_CAM2	P	1.2V power output, voltage level adjustment supported by hardware
28	VCC_2V8_AF_CAM2	P	2.8V motor power output
29	VCC_2V8_CAM2	P	2.8V power output

30	NC		
31	GND	P	Ground

2.4.13 J28 M.2 B-Key (13)

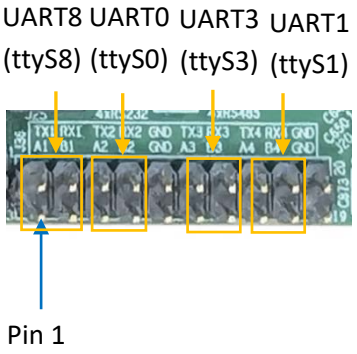
VT-SBC-3588 offers an M.2 B-Key socket that supports a size of 2242. It is compatible with the SATA 3.0 bus. You can connect an SSD for huge data transfer and storage, but please note that it's not available simultaneously with the SATA signal of port J26. Additionally, the slot also supports a size of 3052 and is compatible with PCIe x1/USB 3.1 to connect a 5G module for high-speed wireless communication.

2.4.14 J25 M.2 M-Key (14)

VT-SBC-3588 also offers an M.2 M-Key socket that supports the sizes of 2242 and 2280 and is compatible with the PCIe 3.0 x4 to connect an SSD for storage expansion.

2.4.15 J36 RS485/RS232 connector (15)

VT-SBC-3588 has 4 RS232/RS485 multiplexers (2 x 10 x 2mm), namely UART8, UART0, UART3 and UART1 which are identified as /dev/ttyS8, /dev/ttyS0, /dev/ttyS3 and /dev/ttyS1, respectively in the software system. The serial mode is switchable by software commands.

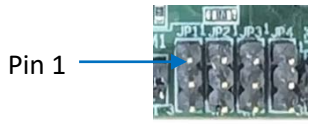


Pinout description of J36:

Pin	Name	Type	Description
1	485_A_1		RS485_1 A
2	232_TX_1	O	RS232_1 data output
3	485_B_1		RS485_1 B
4	232_RX_1	I	RS232_1 data input
5	485_A_2		RS485 Bus_2 A
6	232_TX_2	O	RS232_2 data output
7	485_B_2		RS485_2 B
8	232_RX_2	I	RS232_2 data input
9	GND	P	Ground

10	GND	P	Ground
11	485_A_3		RS485_3 A
12	232_TX_3	O	RS232_3 data output
13	485_B_3		RS485_3 B
14	232_RX_3	I	RS232_3 data input
15	485_A_4		RS485_4 A
16	232_TX_4	O	RS232_4 data output
17	485_B_4		RS485_4 B
18	232_RX_4	I	RS232_4 data input
19	GND	P	Ground
20	GND	P	Ground

When the connector is in RS485 mode, the jumpers (JP1, JP2, JP3, JP4) to the left of the connector are used to connect terminal resistors (120Ω) in series respectively for RS485_1, RS485_2, RS485_3, RS485_4 to prevent signal reflection and improve the signal quality.



Pinout description of JP1:

Pin	Name	Type	Description
1	485_B_1		RS485_1 B, 120Ω resistor connection in series
2	485_A_1		RS48_1 A
3	NC		

Pinout description of JP2:

Pin	Name	Type	Description
1	485_B_2		RS485_2 B, 120Ω resistor connection in series
2	485_A_2		RS48_2 A
3	NC		

Pinout description of JP3:

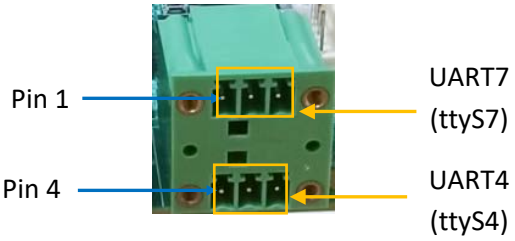
Pin	Name	Type	Description
1	485_B_3		RS485_3 B, 120Ω resistor connection in series
2	485_A_3		RS48_3 A
3	NC		

Pinout description of JP3:

Pin	Name	Type	Description
1	485_B_4		RS485_4 B, 120Ω resistor connection in series
2	485_A_4		RS48_4 A
3	NC		

2.4.16 J34 RS232/RS485 (Phoenix terminal) (16)

VT-SBC-3588 implements a 6-pin Phoenix terminal (2 x 3 x 3.81mm) configured as two isolated RS232/RS485 multiplexers (UART4 and UART7). The two connectors are identified as /dev/ttyS7 and /dev/ttyS4 in the software system.



Pinout description:

Pin	Name	Type	Description
1	RS485_A_2 / RS232_TX2		RS485_2 A / RS232_2 data output
2	RS485_B_2/ RS232_RX2		RS485_2 B / RS232_2 data input
3	ISO_GND_2	P	RS485_2 / RS232_2 ground_2
4	RS485_A_1/ RS232_TX1		RS485_1 A / RS232_1 data output
5	RS485_B_1/ RS232_RX1		RS485_1 B / RS232_1 data input
6	ISO_GND_1	P	RS485_1/ RS232_1 ground_1

When the connector is in RS485 mode, the jumpers (J33, J35) behind the Phoenix terminal are used to connect terminal resistors (120Ω) in series respectively for UART 4 and UART 7 to prevent signal reflection and improve the signal quality.

Pinout description of J33:

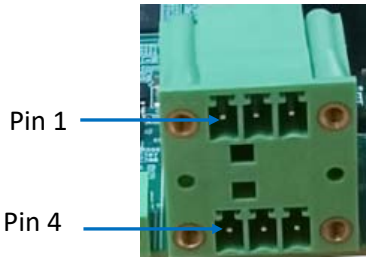
Pin	Name	Type	Description
1	RS485_B_1		RS485_1 B, 120Ω resistor connection in series
2	RS485_A_1		RS485_1 A
3	NC		

Pinout description of J35:

Pin	Name	Type	Description
1	RS485_B_2		RS485_2 B, 120Ω resistor connection in series
2	RS485_A_2		RS485_2 A
3	NC		

2.4.17 J32 CAN (17)

VT-SBC-3588 also implements another 6-pin Phoenix terminal (2 x 3 x 3.81mm) configured as two CAN connectors.



Pinout description:

Pin	Name	Type	Description
1	CAN0_H		High level CAN0
2	CAN0_L		Low level CAN0
3	GND	P	Ground
4	CAN1_H		High level CAN1
5	CAN1_L	P	Low level CAN1
6	GND	P	Ground

2.4.18 J39 SPI (18)

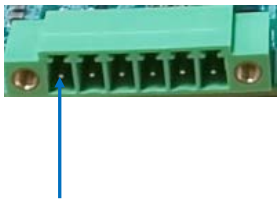
VT-SBC-3588 implements an SPI connector (2 x 3 x 2mm) for connecting a peripheral for data transfer and communication.

Pinout description:

Pin	Name	Type	Description
1	SPI0_MISO_M3_3V3	I/O	SPI MISO, 3.3V level
2	SPI0_MOSI_M3_3V3	I/O	SPI MOSI, 3.3V level
3	SPI0_CLK_M3_3V3	I/O	SPI CLK, 3.3V level
4	SPI0_CS0_M3_3V3	I/O	SPI CS, 3.3V level
5	VCC_3V3_S0	P	3.3V power output
6	GND	P	Ground

2.4.19 J37 GPIO (Phoenix terminal) (19)

VT-SBC-3588 offers 4 GPIOs on a 6-pin Phoenix terminal (1 x 6 x 3.81mm) to users to connect and control external devices.

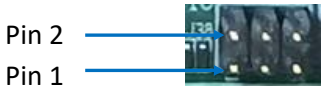


Pinout description:

Pin	Name	Type	Description
1	GPIO8_5V	I/O	GPIO 5V level
2	GPIO7_5V	I/O	GPIO 5V level
3	GPIO6_5V	I/O	GPIO 5V level
4	GPIO5_5V	I/O	GPIO 5V level
5	VCC_5V0	P	5V power output
6	GND	P	Ground

2.4.20 J38 GPIO header (20)

There is a GPIO header on the Board, providing 3 GPIOs for users to customize.

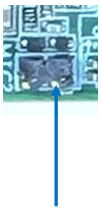


Pinout description:

Pin	Name	Type	Description
1	GPIO4_5V	I/O	GPIO 5V level
2	GPIO3_5V	I/O	GPIO 5V level
3	GPIO2_5V	I/O	GPIO 5V level
4	NC		
5	VCC_5V0	P	5V power output
6	GND	P	Ground

2.4.21 J24 Microphone connector (21)

VT-SBC-3588 provides a microphone connector (1 x 2 x 2mm) for the audio input.

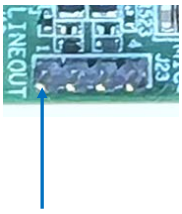


Pinout description:

Pin	Name	Type	Description
1	MIC2P	I	MIC input
2	MIC2N	I	MIC input

2.4.22 J23 Line out connector (22)

There is a line out connector (1 x 4 x 2mm) next to the microphone connector for connecting speakers to enable audio output.



Pin 1

Pinout description:

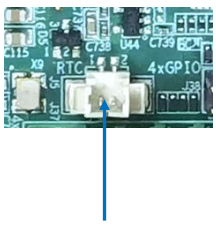
Pin	Name	Type	Description
1	GND	P	Ground
2	ROUT1	O	Right channel output
3	LOUT1	O	Left channel output
4	GND	P	Ground

2.4.23 J18/J19 Wi-Fi and BT antenna connectors (3)

J18 is a Wi-Fi (2.4G/5GHz) antenna connector while J19 is for both Wi-Fi (2.4G/5GHz) & Bluetooth antennas. Both are male connectors.

2.4.24 J5 RTC battery connector (24)

The RTC battery connector is designed to connect a battery to power the RTC and ensure accurate timekeeping. A 3V RTC battery is recommended.



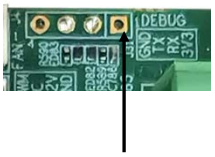
Pin 1

Pinout description:

Pin	Name	Type	Description
1	RTC_P	PI	RTC power input, 3V
2	GND	P	Ground

2.4.25 J31 Debug port (25)

VT-SBC-RK3588 implements a serial debug port, for debugging or configuring the Board. It is recommended to set the baud rate of the port to 1.5Mbps before use.



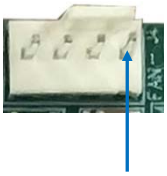
Pin 1

Pinout description:

Pin	Name	Type	Description
1	VCC_3V3_S3	P	3.3V power output
2	UART2_RX_M0_DEBUG	I	Data input, default baud rate 1.5Mbps, TTL 3.3V
3	UART2_TX_M0_DEBUG	O	Data output, default baud rate 1.5Mbps, TTL 3.3V
4	GND	P	Ground

2.4.26 J4 Fan connector (26)

The fan connector (1 x 4 x 2.54mm) is for connecting a fan to provide active dissipation for the Board. It is recommended to connect a fan with power consumption less than 6W (12V/0.5A).



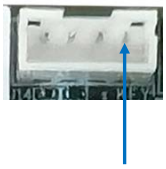
Pin 1

Pinout description:

Pin	Name	Type	Description
1	GND	P	Ground
2	FAN_12V	PO	12V power output, current limit 0.5A
3	NC		
4	PWM0_M2_FAN	O	PWM output, voltage domain 5V

2.4.27 J14 Power button connector (27)

The power button connector (1 x 4 x 2mm) provides control signals for the power button, to control the power of the Board.



Pin 1

Pinout description:

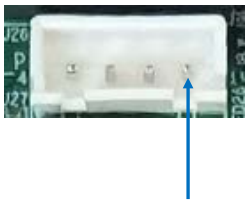
Pin	Name	Type	Description
1	LED+	P	LED +
2	GND	P	Ground
3	PWRON	I	Power button control input
4	GND	P	Ground

2.4.28 J26 SATA 3.0 (28)

The SATA 3.0 interface is designed to connect an SSD for expansion of the storage capacity. As mentioned before, the SATA signals of this interface and the M.2 B-Key socket cannot be used simultaneously. If the SATA signal of this interface is utilized for connecting an SSD, then the M.2 B-Key socket's SATA signal will not be available. Conversely, if the M.2 B-Key socket's SATA signal is utilized, then the SATA 3.0 interface will be disabled.

2.4.29 J27 SATA power connector (29)

VT-SBC-RK3588 implements a SATA power connector (1 x 4 x 2.54mm) for supplying power to the SATA storage device, enabling reliable and efficient storage performance.



Pin 1

Pinout description:

Pin	Name	Type	Description
1	VCC_5V0	P	5V power output
2	GND	P	Ground
3	GND	P	Ground
4	VCC12V_DCIN	P	12V power output

2.4.30 J29 Micro SIM slot (30)

There is a Micro SIM slot on the back of VT-SBC-3588.

Specifications: Micro SIM, push-push, hot plug supported

2.4.31 J3 Micro SD slot (31)

VT-SBC-3588 implements a Micro SD slot for expansion of the storage capacity, supporting up to 128GB storage.

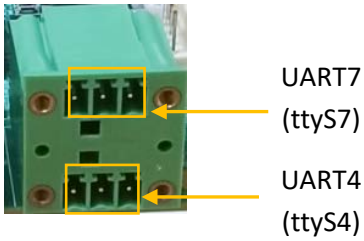
CHAPTER 3 FIRST-USE DEBUGGING

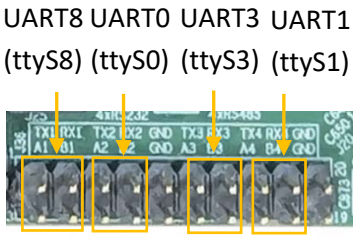
This chapter is mainly about the first-use debugging of interfaces and software applications. You can connect a keyboard, a mouse and a monitor to VT-SBC-3588 and debug the Board directly in the console when it is running a Linux distribution, or, you can use a USB Type-A to Type-C cable to connect the Board to a host computer for the same purpose.

3.1 Serial Connectors

VT-SBC-3588 offers two RS232/RS485 multiplexers on the Phoenix terminal (UART 4 and UART 7), mapped as /dev/ttyS4 and ttyS7, respectively. There are another four RS232/RS485 multiplexers (UART8, UART0, UART3 and UART1) on the Board, mapped as /dev/ttyS8, /dev/ttyS0, /dev/ttyS3, and /dev/ttyS1, respectively.

Refer to 2.4.15 & 2.4.16 for the pinout of the connectors.





You can use a proper USB-to-serial adapter to connect one serial port and the host computer first (TX-RX, RX-TX, GND-GND for RS232 wiring, A-A, B-B, GND-GND for RS485 wiring), then use a serial communication program to check if the serial port functions properly.

Alternatively, you can cross connect any two serial ports and test the functionality of the ports directly in the device console if the device is running Debian or Ubuntu system. If the device is running Android system, please remember to connect it to a host computer and execute the commands on the host computer.

3.1.1 Android system

The following is a summary of the commands for the serial connectors in different modes when the Board is running the Android system.

RS232 mode		
UART 0	Transmit data	# echo 0 > /sys/class/gpio/gpio504/value # echo 1 > /sys/class/gpio/gpio503/value # echo "Test string" > /dev/ttyS0
	Receive data	# echo 0 > /sys/class/gpio/gpio504/value # echo 1 > /sys/class/gpio/gpio503/value # cat /dev/ttyS0
UART 1	Transmit data	# echo 0 > /sys/class/gpio/gpio508/value # echo 1 > /sys/class/gpio/gpio507/value # echo "Test string" > /dev/ttyS1
	Receive data	# echo 0 > /sys/class/gpio/gpio508/value # echo 1 > /sys/class/gpio/gpio507/value # cat /dev/ttyS1
UART 3	Transmit data	# echo 0 > /sys/class/gpio/gpio506/value # echo 1 > /sys/class/gpio/gpio505/value # echo "Test string" > /dev/ttyS3
	Receive data	# echo 0 > /sys/class/gpio/gpio506/value # echo 1 > /sys/class/gpio/gpio505/value # cat /dev/ttyS3
UART 4	Transmit data	# echo 0 > /sys/class/gpio/gpio497/value # echo 1 > /sys/class/gpio/gpio498/value # echo "Test string" > /dev/ttyS4
	Receive data	# echo 0 > /sys/class/gpio/gpio497/value # echo 1 > /sys/class/gpio/gpio498/value # cat /dev/ttyS4
UART 7	Transmit data	# echo 0 > /sys/class/gpio/gpio499/value # echo 1 > /sys/class/gpio/gpio500/value # echo "Test string" > /dev/ttyS7
	Receive data	# echo 0 > /sys/class/gpio/gpio499/value # echo 1 > /sys/class/gpio/gpio500/value # cat /dev/ttyS7
UART 8	Transmit data	# echo 0 > /sys/class/gpio/gpio502/value # echo 1 > /sys/class/gpio/gpio501/value # echo "Test string" > /dev/ttyS8
	Receive data	# echo 0 > /sys/class/gpio/gpio502/value # echo 1 > /sys/class/gpio/gpio501/value # cat /dev/ttyS8

RS485 mode		
UART 0	Transmit data	# echo 0 > /sys/class/gpio/gpio504/value # echo 0 > /sys/class/gpio/gpio503/value # echo "Test string" > /dev/ttyS0
	Receive data	# echo 0 > /sys/class/gpio/gpio504/value # echo 0 > /sys/class/gpio/gpio503/value # cat /dev/ttyS0
UART 1	Transmit data	# echo 0 > /sys/class/gpio/gpio508/value # echo 0 > /sys/class/gpio/gpio507/value # echo "Test string" > /dev/ttyS1
	Receive data	# echo 0 > /sys/class/gpio/gpio508/value # echo 0 > /sys/class/gpio/gpio507/value # cat /dev/ttyS1
UART 3	Transmit data	# echo 0 > /sys/class/gpio/gpio506/value # echo 0 > /sys/class/gpio/gpio505/value # echo "Test string" > /dev/ttyS3
	Receive data	# echo 0 > /sys/class/gpio/gpio506/value # echo 0 > /sys/class/gpio/gpio505/value # cat /dev/ttyS3
UART 4	Transmit data	# echo 0 > /sys/class/gpio/gpio497/value # echo 0 > /sys/class/gpio/gpio498/value # echo "Test string" > /dev/ttyS4
	Receive data	# echo 0 > /sys/class/gpio/gpio497/value # echo 0 > /sys/class/gpio/gpio498/value # cat /dev/ttyS4
UART 7	Transmit data	# echo 0 > /sys/class/gpio/gpio499/value # echo 0 > /sys/class/gpio/gpio500/value # echo "Test string" > /dev/ttyS7
	Receive data	# echo 0 > /sys/class/gpio/gpio499/value # echo 0 > /sys/class/gpio/gpio500/value # cat /dev/ttyS7
UART 8	Transmit data	# echo 0 > /sys/class/gpio/gpio502/value # echo 0 > /sys/class/gpio/gpio501/value # echo "Test string" > /dev/ttyS8
	Receive data	# echo 0 > /sys/class/gpio/gpio502/value # echo 0 > /sys/class/gpio/gpio501/value # cat /dev/ttyS8

The example below illustrates how to test the functionality of the ports in RS232 mode:

1. Cross connect any two serial connectors (TX-RX, RX-TX, GND-GND);

2. Open a terminal and use one port (UART 7 for instance) to receive data;

```
# echo 0 > /sys/class/gpio/gpio499/value
# echo 1 > /sys/class/gpio/gpio500/value
# cat /dev/ttyS7
```

3. Open another terminal and use the other serial port (UART 4) to transmit data;

```
# echo 0 > /sys/class/gpio/gpio497/value
# echo 1 > /sys/class/gpio/gpio498/value
# echo "Test string" > /dev/ttyS4
```

4. The first terminal will print the data received;

5. Refer to the tables in the previous page for the commands specific to different modes and data transport purposes.

In this example, you can surely use single one serial port for transmitting and receiving the data.

3.1.2 Ubuntu and Debian systems

The following is a summary of the commands for the serial connectors in different modes when the Board is running the Ubuntu or Debian system.

RS232 mode		
UART 0	Transmit data	# gpioset gpiochip6 3=0 # gpioset gpiochip6 2=1 # echo "Test string" > /dev/ttyS0
	Receive data	# gpioset gpiochip6 3=0 # gpioset gpiochip6 2=1 # cat /dev/ttyS0
UART 1	Transmit data	# gpioset gpiochip6 7=0 # gpioset gpiochip6 6=1 # echo "Test string" > /dev/ttyS1
	Receive data	# gpioset gpiochip6 7=0 # gpioset gpiochip6 6=1 # cat /dev/ttyS1
UART 3	Transmit data	# gpioset gpiochip6 5=0 # gpioset gpiochip6 4=1 # echo "Test string" > /dev/ttyS3
	Receive data	# gpioset gpiochip6 5=0 # gpioset gpiochip6 4=1 # cat /dev/ttyS3
UART 4	Transmit data	# gpioset gpiochip7 4=0 # gpioset gpiochip7 5=1 # echo "Test string" > /dev/ttyS4
	Receive data	# gpioset gpiochip7 4=0 # gpioset gpiochip7 5=1 # cat /dev/ttyS4
UART 7	Transmit data	# gpioset gpiochip7 6=0 # gpioset gpiochip7 7=1 # echo "Test string" > /dev/ttyS7
	Receive data	# gpioset gpiochip7 6=0 # gpioset gpiochip7 7=1 # cat /dev/ttyS7
UART 8	Transmit data	# gpioset gpiochip6 1=0 # gpioset gpiochip6 0=1 # echo "Test string" > /dev/ttyS8
	Receive data	# gpioset gpiochip6 1=0 # gpioset gpiochip6 0=1 # cat /dev/ttyS8

RS485 mode		
UART 0	Transmit data	# gpioset gpiochip6 3=1 # gpioset gpiochip6 2=0 # echo "Test string" > /dev/ttyS0
	Receive data	# gpioset gpiochip6 3=0 # gpioset gpiochip6 2=0 # cat /dev/ttyS0
UART 1	Transmit data	# gpioset gpiochip6 7=1 # gpioset gpiochip6 6=0 # echo "Test string" > /dev/ttyS1
	Receive data	# gpioset gpiochip6 7=0 # gpioset gpiochip6 6=0 # cat /dev/ttyS1
UART 3	Transmit data	# gpioset gpiochip6 5=1 # gpioset gpiochip6 4=0 # echo "Test string" > /dev/ttyS3
	Receive data	# gpioset gpiochip6 5=0 # gpioset gpiochip6 4=0 # cat /dev/ttyS3
UART 4	Transmit data	# gpioset gpiochip7 4=1 # gpioset gpiochip7 5=0 # echo "Test string" > /dev/ttyS4
	Receive data	# gpioset gpiochip7 4=0 # gpioset gpiochip7 5=0 # cat /dev/ttyS4
UART 7	Transmit data	# gpioset gpiochip7 6=1 # gpioset gpiochip7 7=0 # echo "Test string" > /dev/ttyS7
	Receive data	# gpioset gpiochip7 6=0 # gpioset gpiochip7 7=0 # cat /dev/ttyS7
UART 8	Transmit data	# gpioset gpiochip6 1=1 # gpioset gpiochip6 0=0 # echo "Test string" > /dev/ttyS8
	Receive data	# gpioset gpiochip6 1=0 # gpioset gpiochip6 0=0 # cat /dev/ttyS8

The example below illustrates how to test the functionality of the ports in RS232 mode:

1. Cross connect the serial connectors as shown below (TX-RX, RX-TX, GND-GND);

2. Open a terminal and use one port (UART 7 for instance) to receive data;

```
# gpioset gpiochip7 6=0
# gpioset gpiochip7 7=1
# cat /dev/ttyS7
```

3. Open another terminal and use the other serial port (UART 4) to transmit data;

```
# gpioset gpiochip7 4=0
# gpioset gpiochip7 5=1
# echo "Test string" > /dev/ttyS4
```

4. The first terminal will print the data received;

5. Refer to the tables in the previous page for commands in different modes and different data transport purposes.

3.2 CAN

The device implements two CAN bus connectors (CAN0 and CAN1). You can cross connect the two connectors (H-H, L-L, GND-GND), and use one to receive data and the other to transmit data to test the functionality of the connectors. Refer to 2.4.17 for the pinout of the connectors.

- Use CAN0 to receive data and CAN1 to transmit data:

1. Open a terminal to set the baud rate of CAN0 and input the command to receive data;

```
# ip link set can0 up type can bitrate 125000
# candump can0
```

2. Open another terminal to set the baud rate of CAN1 and input the command to send data.

```
# ip link set can1 up type can bitrate 125000
# cansend can1 500#11.22.33.44
```

- Use CAN1 to receive data and CAN0 to transmit data:

1. Open a terminal to set the baud rate of CAN1 and input the command to receive data;

```
# ip link set can1 up type can bitrate 125000 (skip this if you have set the baud rate)
# candump can1
```

2. Open another terminal to set the baud rate of CAN0 and input the command to send data.

```
# ip link set can0 up type can bitrate 125000 (skip this if you have set the baud rate)
# cansend can0 500#11.22.33.44
```

- Check the status of CAN0:

```
# ip -details link show can0
```

```
root@linaro-alip:/# ip -details link show can0
2: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT
 group default qlen 10
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can state ERROR-PASSIVE (berr-counter tx 0 rx 0) restart-ms 1
          bitrate 125000 sample-point 0.871
          tq 60 prop-seg 57 phase-seg1 57 phase-seg2 17 sjw 1
          rockchip_canfd: tseg1 1..128 tseg2 1..128 sjw 1..128 brp 1..256 brp-i
nc 2
          clock 99000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max
_segs 65535
root@linaro-alip:/#
```

- Check the status of CAN1:

```
# ip -details link show can1
```

3.3 GPIO

VT-SBC-3588 offers 7 GPIOs, including 4 on the Phoenix terminal and 3 on a 6-pin header. GPIO2 ~ GPIO8 are mapped as gpio486, gpio487, gpio488, gpio489, gpio490, gpio491, gpio492 in the software system, respectively. You can refer to 2.4.19 and 2.4.20 for the pinout of the GPIOs.

Input the following command to check the status of the GPIOs:

```
# cat /sys/kernel/debug/gpio
```

The commands for setting the high/low level of the GPIO pins in Android system are different from those used in Ubuntu and Debian systems. You are recommended to set the high/low level of the GPIO pins with a reference to the commands for the respective system and use a voltmeter to check if the settings are valid.

3.3.1 Android system

The following table summarizes the commands for setting the high/low level of the GPIO pins in Android system.

GPIOs		
GPIO2	High level	# echo 1 > sys/class/gpio/gpio486/value
	Low level	# echo 0 > sys/class/gpio/gpio486/value
GPIO3	High level	# echo 1 > sys/class/gpio/gpio487/value
	Low level	# echo 0 > sys/class/gpio/gpio487/value
GPIO4	High level	# echo 1 > sys/class/gpio/gpio488/value
	Low level	# echo 0 > sys/class/gpio/gpio488/value
GPIO5	High level	# echo 1 > sys/class/gpio/gpio489/value
	Low level	# echo 0 > sys/class/gpio/gpio489/value
GPIO6	High level	# echo 1 > sys/class/gpio/gpio490/value
	Low level	# echo 0 > sys/class/gpio/gpio490/value
GPIO7	High level	# echo 1 > sys/class/gpio/gpio491/value
	Low level	# echo 0 > sys/class/gpio/gpio491/value
GPIO8	High level	# echo 1 > sys/class/gpio/gpio492/value
	Low level	# echo 0 > sys/class/gpio/gpio492/value

To switch the direction of the GPIOs:

```
# echo in > sys/class/gpio/gpioxxx/direction

# echo out > sys/class/gpio/gpioxxx/direction
```

3.3.2 Ubuntu and Debian systems

The following table summarizes the commands for setting the high/low level of the GPIO pins in Ubuntu or Debian system.

GPIOs		
GPIO2	High level	# gpioset gpiochip8 1=1
	Low level	# gpioset gpiochip8 1=0
GPIO3	High level	# gpioset gpiochip8 2=1
	Low level	# gpioset gpiochip8 2=0
GPIO4	High level	# gpioset gpiochip8 3=1
	Low level	# gpioset gpiochip8 3=0
GPIO5	High level	# gpioset gpiochip8 4=1
	Low level	# gpioset gpiochip8 4=0
GPIO6	High level	# gpioset gpiochip8 5=1
	Low level	# gpioset gpiochip8 5=0
GPIO7	High level	# gpioset gpiochip8 6=1
	Low level	# gpioset gpiochip8 6=0
GPIO8	High level	# gpioset gpiochip8 7=1
	Low level	# gpioset gpiochip8 7=0

3.4 Watchdog Timer

The watchdog timer is disabled by default. Once enabled, the watchdog timer is supposed to communicate with the system within every 44 seconds (dog feeding). If not, the system will reboot when the watchdog times out, indicating a system or application failure.

1. Write any character except capital V to open the watchdog;

```
# echo A > /dev/watchdog
```

2. Set the Runtime for dog feeding (e.g., 22 seconds) and the interval for system reboot upon an abnormality (e.g., 2 minutes);

```
[Manager]
LogLevel=warning
LogTarget=journal
#LogColor=yes
#LogLocation=no
#DumpCore=yes
#ShowStatus=yes
#CrashChangeVT=no
#CrashShell=no
#CrashReboot=no
#CtrlAltDelBurstAction=reboot-force
#CPUAffinity=1 2
#NUMAPolicy=default
#NUMAMask=
RuntimeWatchdogSec=22s
RebootWatchdogSec=2min
#ShutdownWatchdogSec=10min
#KExecWatchdogSec=0
#WatchdogDevice=
```

3. Disable the watchdog;

```
# echo V > /dev/watchdog
```

4. Users can also enable hardware watchdog in “systemd” and configure it likewise.

```
# /etc/systemd/system.conf
```

CHAPTER 4 ANDROID SYSTEM MANUAL

Prerequisites:

- VT-SBC-3588
- A Windows host computer
- Software release package of VT-SBC-3588
- A USB mouse, keyboard, and monitor for connecting the board for easier operation
- A 12V/24V power adapter for booting the board
- A USB Type-A to Type-C cable for connecting the board and the host computer

4.1 Enable Developer Options

To enable Developer options of VT-SBC-3588, follow the steps below:

1. Connect VT-SBC-3588 to a mouse, a keyboard, and a display for easier operations;
2. After the system has booted, swipe up to access the application drawer;
3. Click **Settings** > **About <device>** in sequence;
4. Scroll down to **Build number**, and click it consecutively for at least 7 times to enable **Developer options**;
5. Go back to **Settings > System > Advanced** > **Developer options** and toggle on **USB debugging**, then you can customize the device settings.

Depending on the Android version, the entry might vary slightly.

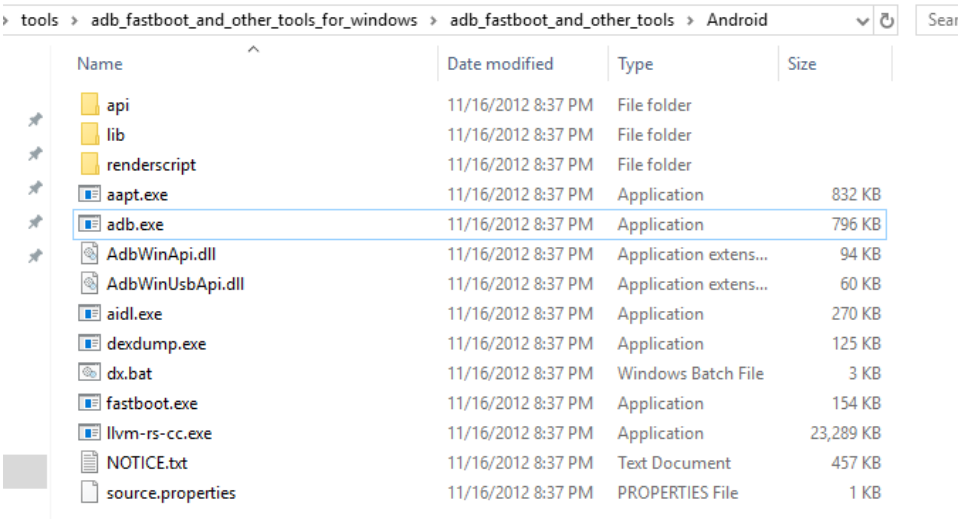
4.2 ADB Setup on the Windows host

Android Debug Bridge (ADB) is a tool that is designed to connect your development workstation directly to your Android device for debugging, device upgrading, app installation, etc.

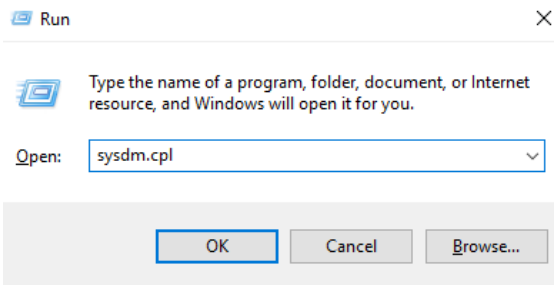
Adding the ADB executable file to the system's environment variable allows you to run the ADB tool regardless of your current working directory.

Follow the steps below to set up the ADB on the Windows host.

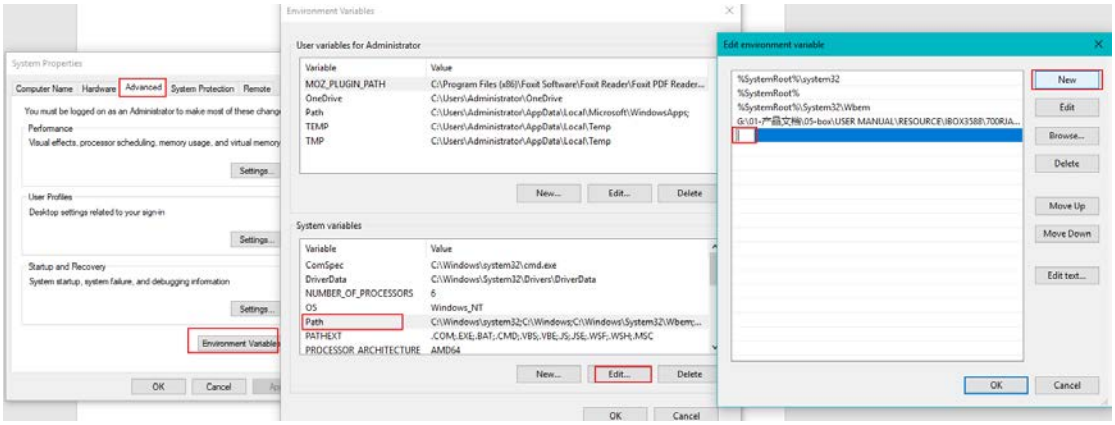
1. Unzip the software release package and navigate to the following directory: \SW\tools;
2. Extract the **adb_fastboot_and_other_tools_for_windows** zip file;
3. Navigate to the Android folder that contains the ADB tool kit, and copy the folder path;



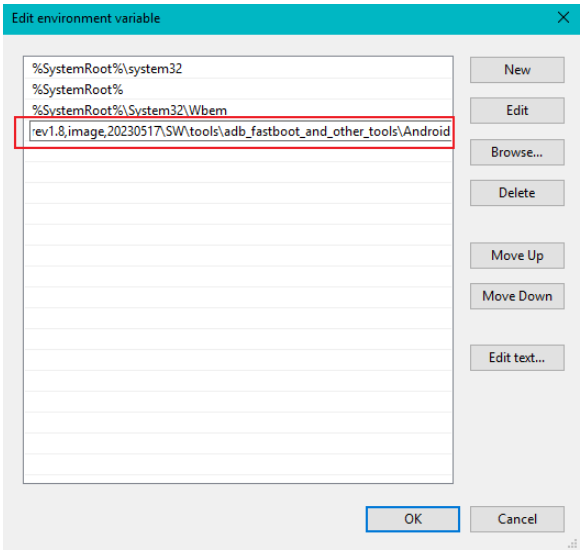
4. Press "Win + R" and input sysdm.cpl in the dialogue box to open the settings interface;



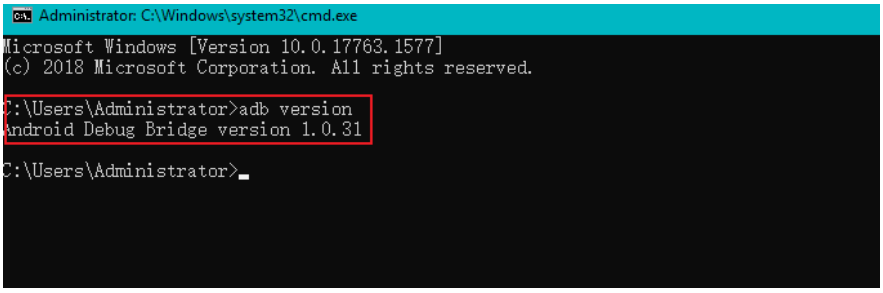
5. Click in sequence **Advanced > Environment Variables > System Variables > Path > Edit**, and click **New** in the pop-up;



6. Paste the path of the **Android** folder, and click **OK** one by one to confirm and exit;



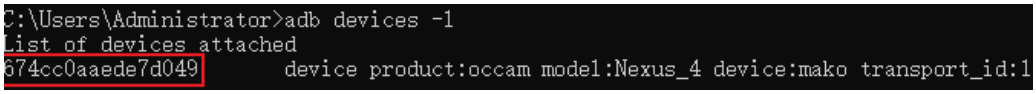
7. Press “Win + R” and input cmd in the dialogue box;
8. Input adb version in the command prompt to check if the ADB tool is installed.



4.3 App Installation via ADB Commands

In addition to the standard pre-installed Android applications, users can install their own applications on the board provided that it runs an Android system and the ADB tool kit is installed and accessible on the Windows host computer.

1. Connect the board and the host computer via a USB Type-A to Type-C cable;
2. Press "Win + R" and input cmd in the dialogue box;
3. Input adb devices -l in the command prompt to check if the board is connected to the host computer;



4. When the device information is displayed under the command, you can copy the serial number (squared as shown above) for the next-step use;
5. Input the following command line to install the app;

```
adb -s <serial number> install <APP path>
```

6. The installation will be executed after the command is input and the result of installation will be displayed below;

7. The newly installed app will be displayed on the App drawer in the Alphabetic order.

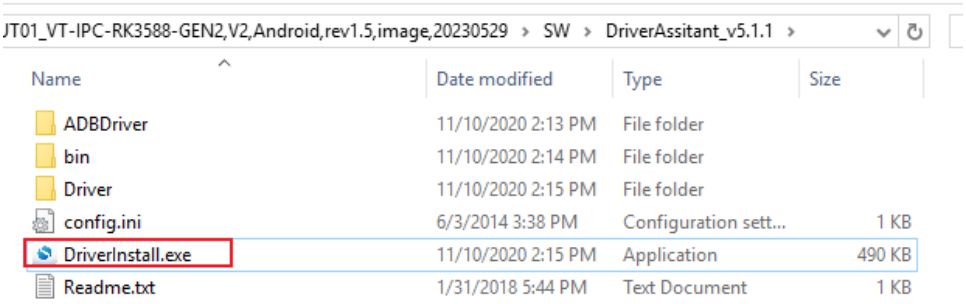
The screenshots are for illustration only and are not intended to represent the actual device number and other information of the device currently in your possession.

In step 5 shown above, you can drag the .apk file from the local directory to the command line to replace the <App path> you typed in manually.

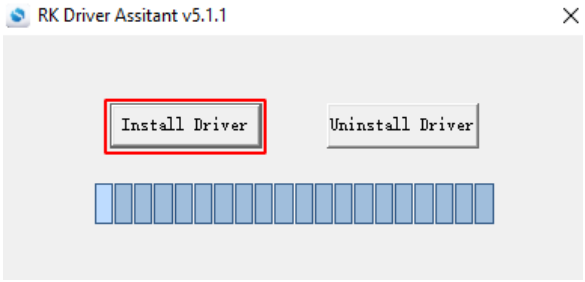
If you failed to install the apk, try using the absolute path of the .apk file enclosed in double quotation marks.

4.4 Firmware Upgrade on a Windows Host

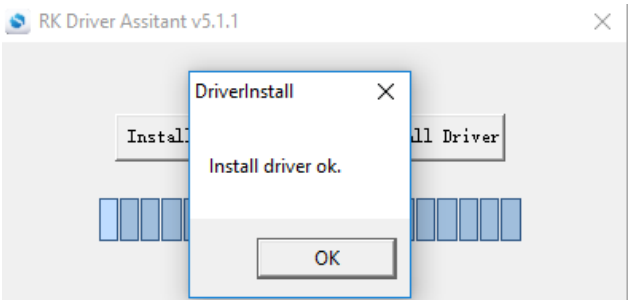
1. Follow the steps in 4.2 to finish ADB setup on the host computer;
2. Unzip the release package, and open the directory of the upgrade driver (path: \SW\ DriverAssitant_vxxx), and locate the **DriverInstall.exe**;



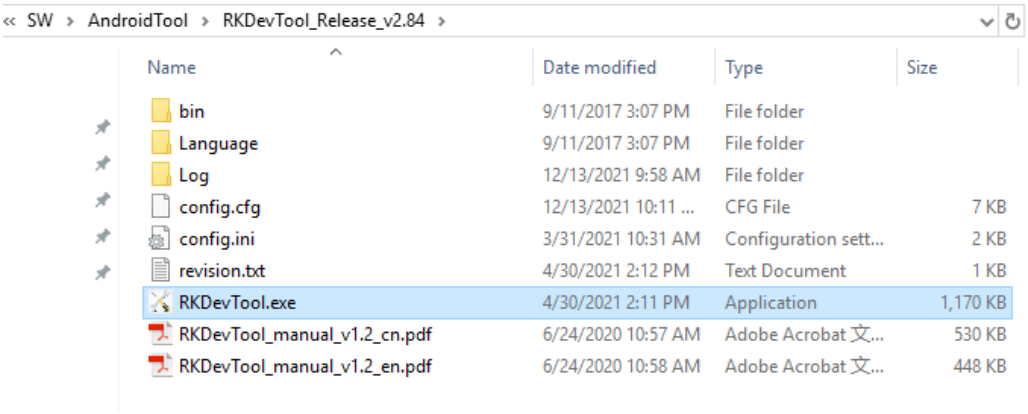
3. Right click the mouse and run the program as administrator;
4. Click **Uninstall Driver** first to remove the previously installed driver, if any, then click **Install Driver** and wait;



5. A pop-up will appear in a second suggesting the driver is installed;

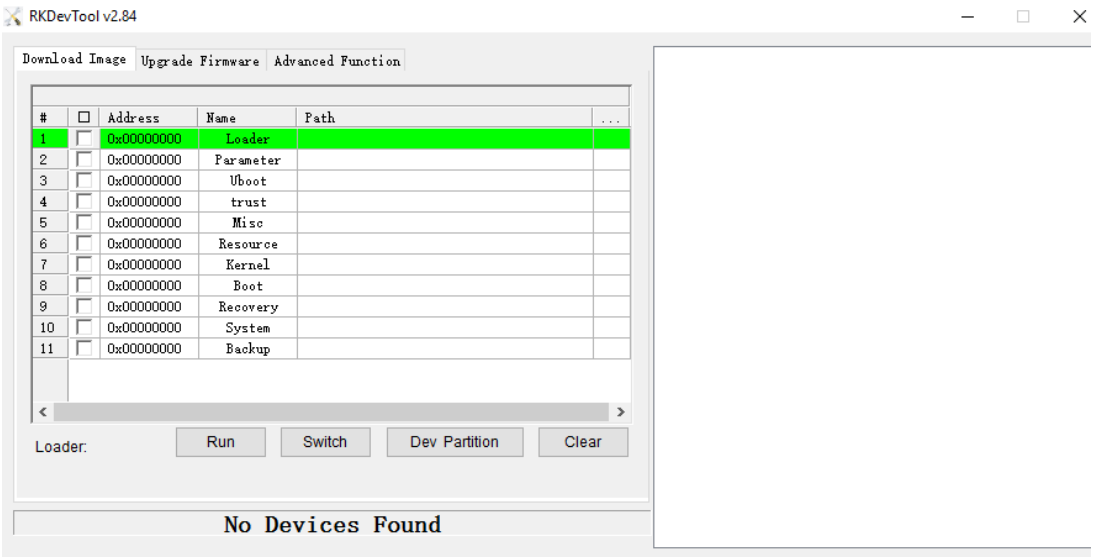


6. Return to the SW folder and open the directory of the upgrade tool (\SW\AndroidTool \RKDevTool_Release_ vxxx\RKDevTool.exe);



7. Double click the program to install the **RKDevTool.exe**;

8. Open the upgrade window;



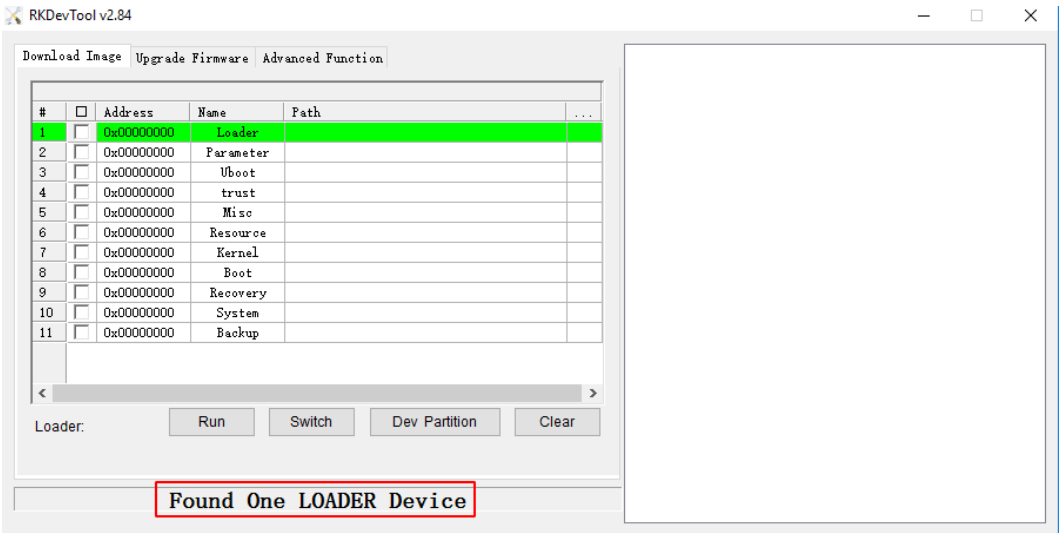
9. Connect VT-SBC-3588 to the Windows host with the USB Type-A to Type-C cable;

10. Press Windows key + R and input cmd in the dialog box to open the command prompt;

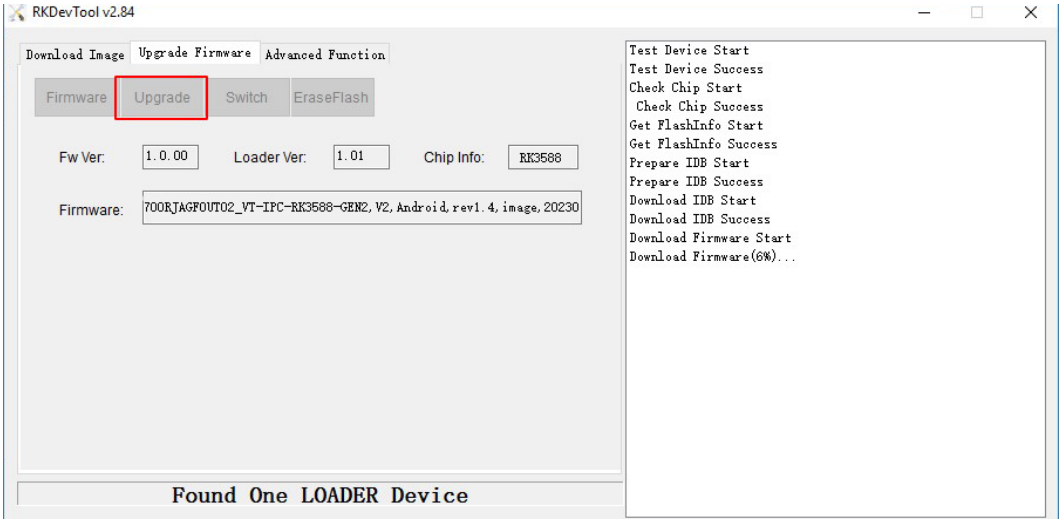
11. Input adb devices in the command prompt to check if the board is connected to the Windows host;

12. Once the board is identified by the Windows host, input adb reboot loader to reboot the board into the bootloader mode;

13. Then the upgrade window will prompt for the existence of a Loader device, indicating that the upgrade process is ready.



14. Click **Upgrade Firmware > Firmware** in upgrade window;
15. Open the image file **update.img** from the directory (\SW\Image), and the firmware details will be automatically populated;
16. Click the **Upgrade** button and the system will start to download the image and upgrade;



17. When the upgrade finishes, the system will reboot automatically.

CHAPTER 5 DEBIAN SYSTEM MANUAL

5.1 About the System

Debian is a Linux distribution and the board was built by Debian GNU/Linux 11 operating system.

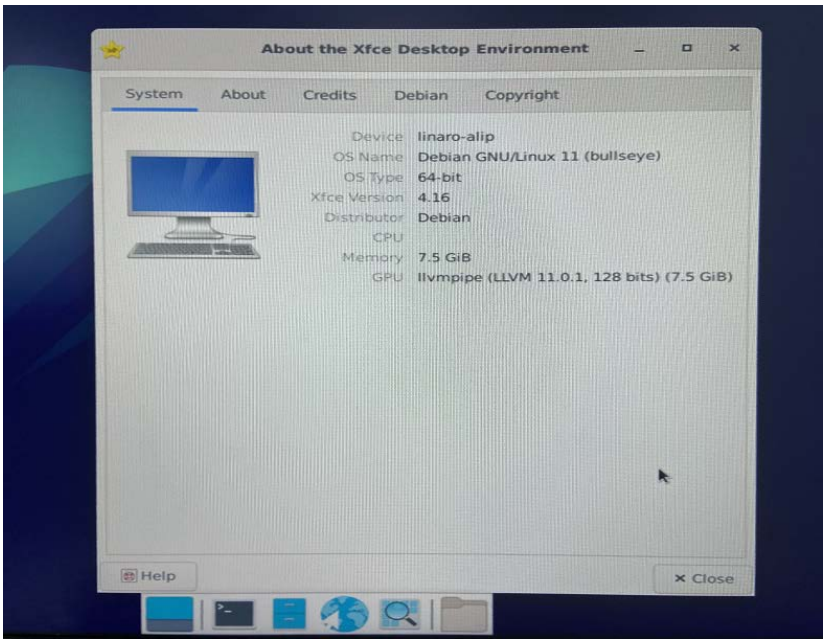
5.1.1 User and password

The system will automatically log in to **linaro** as the default user after bootup.

- User: linaro
- Password: linaro

5.1.2 System information

1. Connect a keyboard, a mouse and a monitor to the device for easy operation;
2. Power on the board and the system will log in to linaro as the default user;
3. Right click the mouse in an empty area to open the property dialogue of the board;
4. Click **Applications > About Xface** to enter the desktop environment, then you can check the system information like device name, operating system, copyright statement, etc.



5.2 System Settings

By clicking on **linaro** on the top right corner of the screen, you can

- Lock the screen;
- Switch the user;
- Shut down/suspend the board;
- (Upon a click of the “Log out” option) Log out/restart/shut down/suspend the board or switch the user while saving the session for future logins.

The menu bar on the top of the screen allows the user to:

- Change the brightness and power settings of the board;
- Adjust the system volume and sound settings;
- Add the board to a Wi-Fi network and pair it with a Bluetooth device;
- Switch between different workspaces.

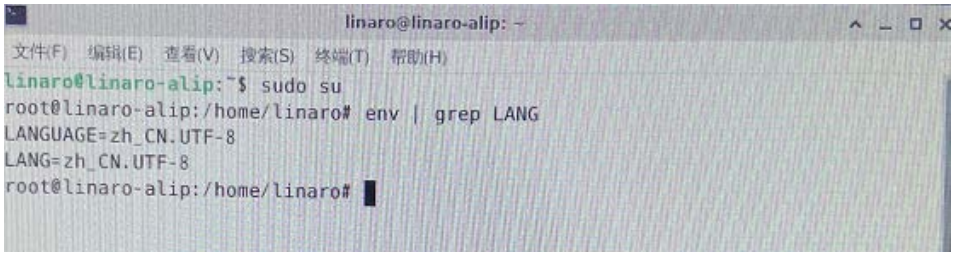
Other system settings are accessible from **Applications** on the top left corner of the screen or upon a right click of the mouse in an empty area of the screen.

5.2.1 Language setup

Assume the system langue is Simplified Chinese. The following example demonstrates how to change the system langue to English (US).

1. Open the Terminal from the bottom of the desktop and input sudo su to switch to the root mode;
2. Run the following command to check the language currently used by the system (the system uses zh_CN.UTF-8 as the locale);

```
# env | grep LANG
```



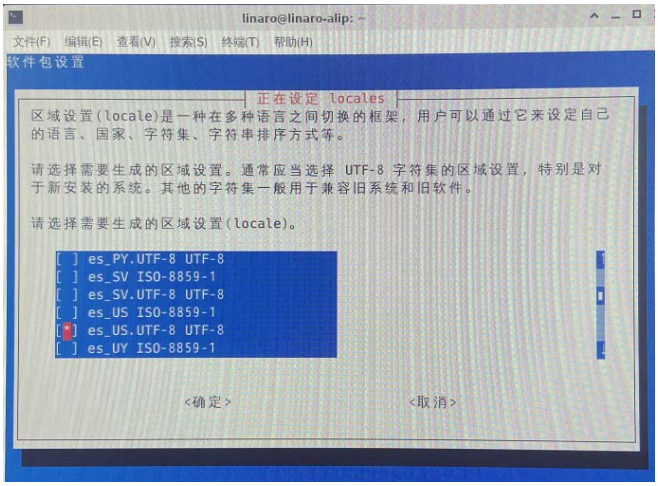
3. Run the following command to call the locale configuration page (en_US.UTF-8 refers to the locale for the target language that you wish to switch to);

```
# export LANG=en_US.UTF-8
```

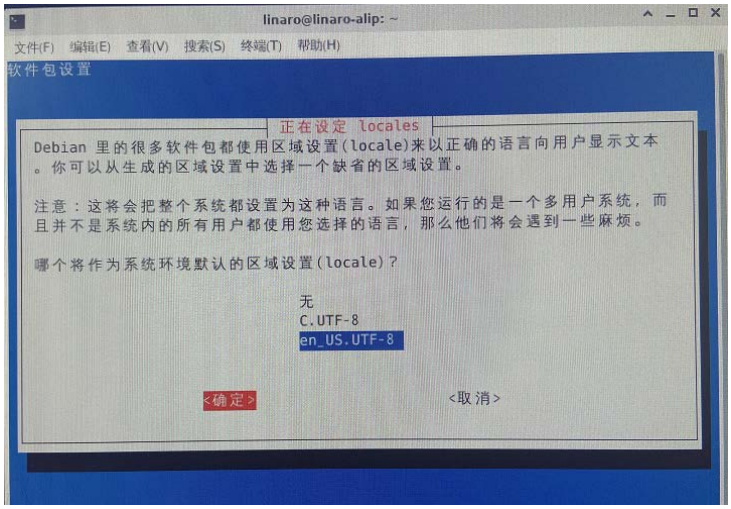
4. Use the following command to re-configure the locales;

```
# dpkg-reconfigure locales
```

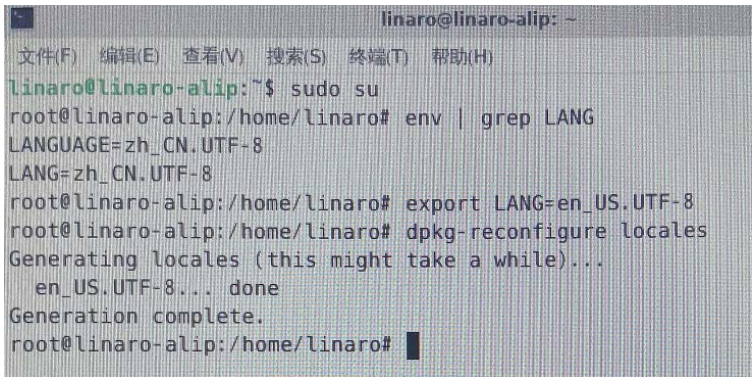
5. Use the up and down arrows on the keyboard to move to the target locale (en_US.UTF-8);
6. Press “Space” in front of the target locale to select the language;



7. Move down to the current locale (zh_CN.UTF-8) and press “Space” to uncheck the current language;
8. Press “Tab” to move the cursor to the left option <OK> (确定 in Chinese) and press “Enter” to confirm your choice;
9. In the following pop-up, use the up and down arrows to move to the target locale (en_US.UTF-8) in the list, then press “Tab” to move the cursor to the left option <OK> (确定 in Chinese) on and press “Enter” to confirm your choice;



10. Wait a few seconds for the locale to take effect;

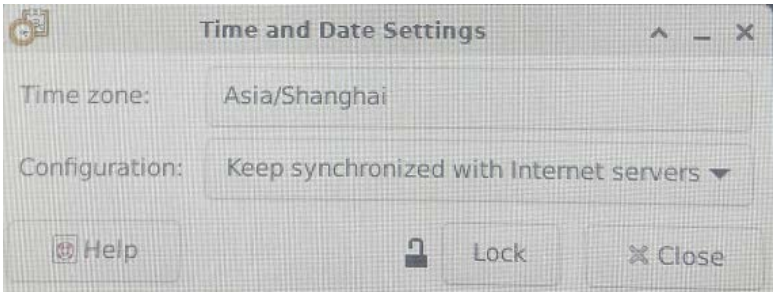


11. Then input reboot to restart the system;
12. Select “Update names” after device reboot to update the files to the current language;
13. Run locale -a command to check if the locale is set as the default system language.

5.2.2 Time and date setup

To change the system time and date, click **Applications** on the top left corner of the screen or right click the mouse in an empty area of the screen to call the menu:

1. Click **System** > **Time and Date** to open the Time and Date setup page;
2. Click **Unlock** and enter the password (linaro) before editing;
3. Click Time zone and select your own time zone from the list;
4. Choose to synchronize the time with the Internet server or fill in the time manually;



5. Click **Lock** to let the settings take effect.

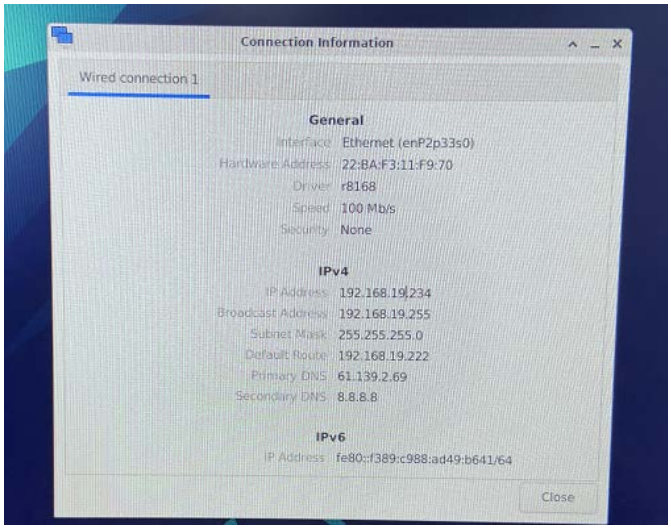
5.2.3 Network configuration

The board is configured to have Internet access once it connects to a live Ethernet or joins to a Wi-Fi network.

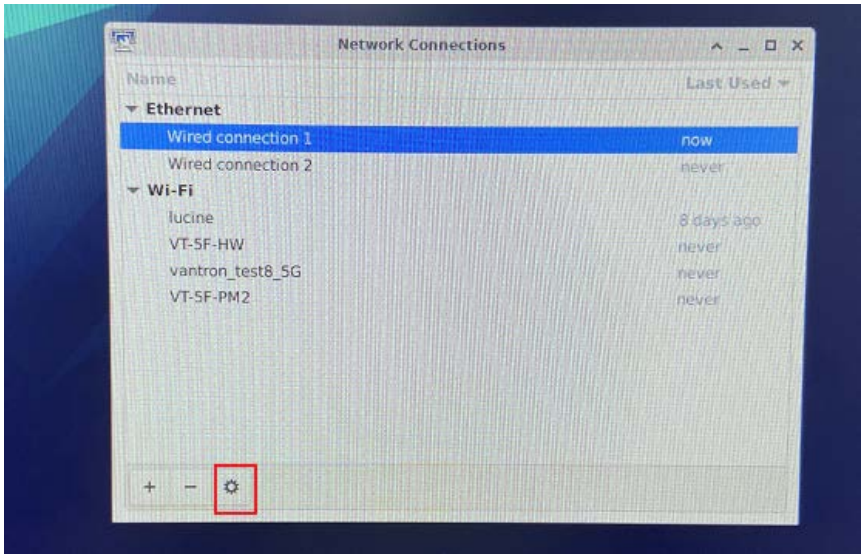
Make sure you have installed the Wi-Fi antennas before connecting to an existing wireless network.

1. To join to a Wi-Fi network, click the network icon (dual-host/Wi-Fi icon) on the menu bar and search for the target network in the list;

2. To check the current network connection, you can move the cursor to the network icon, and right click the mouse to select **Connection Information**:



3. To edit the network configurations, you can right click the network icon and select **Edit Connections**, then select a connection point and click the cog icon to edit the network settings.



5.3 Image Flashing on an Ubuntu Host

5.3.1 Prerequisites

- VT-SBC-3588
- An Ubuntu host (Ubuntu 18.04 or later recommended)
- A USB cable (Type-A to Type-C)

5.3.2 Image flashing

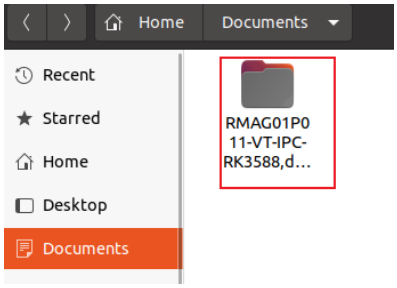
1. Connect the board with the host computer via the USB Type-A to Type-C cable;
2. Open the Terminal and input the following command line to install ADB tool;

```
$ sudo apt-get install adb -y
```

3. Check if the board is connected to the Ubuntu host via ADB;

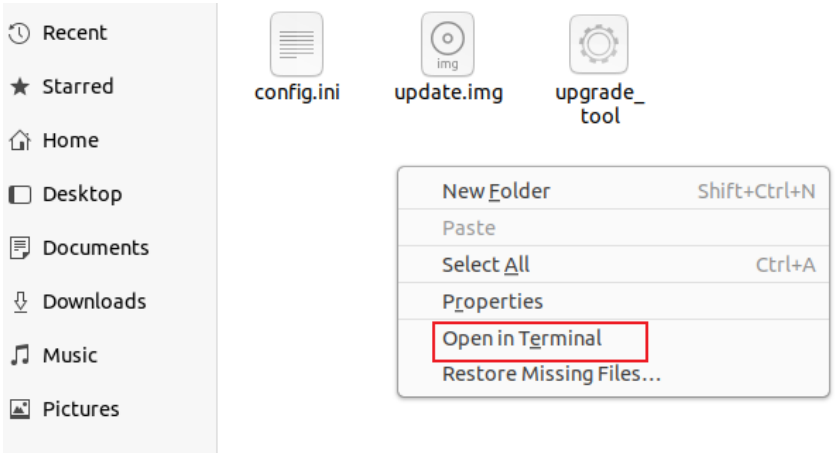
```
$ adb devices -l
```

4. Run `adb shell` to execute shell commands on the board;
5. Enter `reboot loader`, and the board will reboot automatically and enter the bootloader mode;
6. Copy the image package provided by Vantron to the Ubuntu host (for instance, in the **Documents** folder);



7. Unzip the package and open the **img** folder;

8. Right click the mouse in an empty area and click **Open in Terminal** to execute the subsequent commands in the new Terminal;

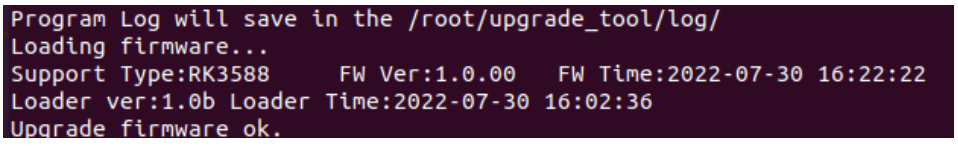


9. Input the following command to start the upgrade process;

```
sudo ./upgrade_tool uf update.img
```

10. Input the password for the user of the Ubuntu host to download the system image;

11. The system will start upgrading once the download finishes, and the board will reboot automatically when the upgrade finishes.



CHAPTER 6 UBUNTU SYSTEM MANUAL

6.1 About the System

Ubuntu is a Linux distribution and the board is running Ubuntu 20.04.LTS.

6.1.1 User and password

Users need to login to the system after system boot.

- User name: linaro
- Password: linaro

6.1.2 System information

1. Connect the board with a monitor using an HDMI cable, or using an HDMI to VGA adapter if necessary;
2. Power on the board and input the password to login the default user (“linaro”);

   Password: linaro
3. Click the **Show Applications** icon  on the bottom left corner to access the system applications;
4. Click **Settings > About** to check more information about the device system.

6.2 System Settings

Upon a click of **Show Applications**, users can click or search a single application for specific use, or modify system settings in **Settings** application.

6.2.1 Language setup

Users may set the display language and language format after accessing **Settings > Region & Language**. With the use of **Language Support**, users can also install/remove input languages. All the modifications will take effect after system reboot.

Commands for language settings are same as those in 5.2.1.

6.2.2 Date & time setup

To change the system date and time, click **Settings > Date & Time** to open the setup page：

1. Click **Unlock** and enter the password before editing;
2. Toggle on **Automatic Date & Time** to keep the device time with the Internet, or toggle off the option and input the date and time manually;
3. Select/search the target time zone and set the time format;
4. Click **Lock** at the top of the interface to save the settings;
5. Exit and await the settings to take effect.

6.2.3 Network/Wi-Fi

The board is configured to have Internet access once it connects to a live Ethernet or joins to a Wi-Fi network. Users can add a VPN network in **Settings > Network** interface.

Make sure you have installed the Wi-Fi antennas before connecting to an existing wireless network.

You can input the following command to check the information of the network interface: ip link show.

6.2.4 User

Users will be able to edit the system account and add/delete a user after inputting the password and unlocking the current user.

6.3 Image Flashing on an Ubuntu Host

6.3.1 Prerequisites

- VT-SBC-3588
- An Ubuntu host (Ubuntu 18.04 or later recommended)
- A USB Type-A to Type-C cable
- Software release package of VT-SBC-3588

6.3.2 Image flashing

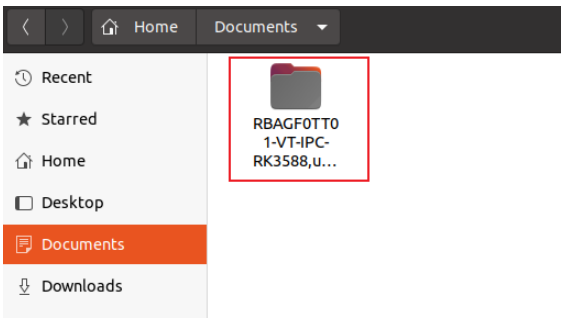
1. Connect the board with the host computer via the USB Type-A to Type-C cable;
2. Open the Terminal and input the following command line to install ADB tool;

```
$ sudo apt-get install adb -y
```

3. Check if the board is connected to the Ubuntu host via ADB;

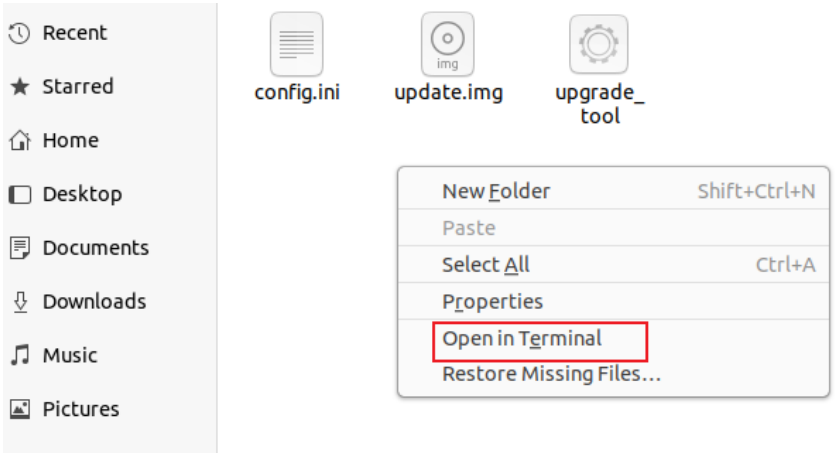
```
$ adb devices -l
```

4. Run adb shell to execute shell commands on the board;
5. Enter reboot loader, and the board will reboot automatically and enter the bootloader mode;
6. Copy the image package provided by Vantron to the Ubuntu host (for instance, in the Documents folder);



7. Unzip the package and open the **img** folder;

8. Right click the mouse in an empty area and click **Open in Terminal** to execute the subsequent commands in the new terminal;



9. Input the following command line in the Terminal to download the upgrade image;

```
sudo ./upgrade_tool uf update.img
```

10. Input the sudo password to download the system image;
11. The system will start upgrading once the download finishes, and the board will reboot automatically when the upgrade finishes.

6.4 ROS Tutorials

Please refer to http://wiki.ros.org/ROS/Tutorials for the details of the ROS system.

CHAPTER 7 DISPOSAL AND WARRANTY

7.1 Disposal

When the device comes to end of life, you are suggested to properly dispose of the device for the sake of the environment and safety.

Before you dispose of the device, please back up your data and erase it from the device.

It is recommended that the device is disassembled prior to disposal in conformity with local regulations. Please ensure that the abandoned batteries are disposed of according to local regulations on waste disposal. Do not throw batteries into fire or put in common waste canister as they are explosive. Products or product packages labeled with the sign of “explosive” should not be disposed of like household waste but delivered to specialized electrical & electronic waste recycling/disposal center.

Proper disposal of this sort of waste helps avoid harm and adverse effect upon surroundings and people’s health. Please contact local organizations or recycling/disposal center for more recycling/disposal methods of related products.

7.2 Warranty

Product warranty

VANTRON warrants to its CUSTOMER that the Product manufactured by VANTRON, or its subcontractors will conform strictly to the mutually agreed specifications and be free from defects in workmanship and materials (except that which is furnished by the CUSTOMER) upon shipment from VANTRON. VANTRON's obligation under this warranty is limited to replacing or repairing at its option of the Product which shall, within **24 months** after shipment, effective from invoice date, be returned to VANTRON's factory with transportation fee paid by the CUSTOMER and which shall, after examination, be disclosed to VANTRON's reasonable satisfaction to be thus defective. VANTRON shall bear the transportation fee for the shipment of the Product to the CUSTOMER.

Out-of-Warranty Repair

VANTRON will furnish the repair services for the Product which are out-of-warranty at VANTRON's then-prevailing rates for such services. At customer's request, VANTRON will provide components to the CUSTOMER for non-warranty repair. VANTRON will provide this service as long as the components are available in the market; and the CUSTOMER is requested to place a purchase order up front. Parts repaired will have an extended warranty of 3 months.

Returned Products

Any Product found to be defective and covered under warranty pursuant to Clause above, shall be returned to VANTRON only upon the CUSTOMER's receipt of and with reference to a VANTRON supplied Returned Materials Authorization (RMA) number. VANTRON shall supply a RMA, when required within three (3) working days of request by the CUSTOMER. VANTRON shall submit a new invoice to the CUSTOMER upon shipping of the returned products to the CUSTOMER. Prior to the return of any products by the CUSTOMER due to rejection or warranty defect, the CUSTOMER shall afford VANTRON the opportunity to inspect such products at the CUSTOMER's location and no Product so inspected shall be returned to VANTRON unless the cause for the rejection or defect is determined to be the responsibility of VANTRON. VANTRON shall in turn provide the CUSTOMER turnaround shipment on defective Product within **fourteen (14) working days** upon its receipt at VANTRON. If such turnaround cannot be provided by VANTRON due to causes beyond the control of VANTRON, VANTRON shall document such instances and notify the CUSTOMER immediately.