

VT-SBC-3588 单板计算机



硬件手册

版本：1.5

© 成都万创科技股份有限公司 版权所有

版本记录

编号	版本	描述	日期
1	V1.0	首次发布	2022年6月4日
2	V1.1	更新部分接口说明	2022年8月30日
3	V1.2	增加串口调试说明	2023年5月6日
4	V1.3	根据操作系统的不同，区分串口调试和GPIO调试的命令	2023年10月17日
5	V1.4	将 Debian 系统串口调试命令更新为与其他系统一致	2023年12月8日
6	V1.5	将 Ubuntu 和 Debian 系统的串口调试命令与 Android 系统的调试命令加以区分	2023年12月15日

目录

前言	1
第 1 章 引言	5
1.1 产品概述	6
1.2 术语/缩写	6
1.3 框图	7
1.4 规格	8
1.5 操作系统	9
1.6 机械尺寸	9
1.7 电源及功耗	9
1.8 环境参数	9
第 2 章 接口说明	10
2.1 主板布局	11
2.2 内存及存储	11
2.2.1 LPDDR4 内存	11
2.2.2 eMMC 闪存	11
2.2.3 扩展存储	11
2.3 识别第 1 引脚	12
2.4 连接头和跳线	12
2.4.1 J1 电源端子 (1)	12
2.4.2 J22 以太网口 (2)	13
2.4.3 J21 以太网口 + USB 2.0 (3)	13
2.4.4 U27 USB 3.0 Type-A (4)	13
2.4.5 J6 USB Type-C (5)	13
2.4.6 J7/J8/J9 USB 2.0 Host 连接头 (6)	14
2.4.7 J13 LVDS 接口 (7)	15
2.4.8 J15 背光连接头 (8)	16
2.4.9 J9 HDMI 接口 (9)	16
2.4.10 J17 eDP 接口 (10)	16
2.4.11 J16 MIPI DSI 接口 (11)	18
2.4.12 J10/J11 MIPI CSI 接口 (12)	19
2.4.13 J28 M.2 B-Key 插槽 (13)	21
2.4.14 J25 M.2 M-Key 插槽 (14)	21
2.4.15 J36 RS485/RS232 连接头 (15)	21
2.4.16 J34 RS232/RS485 (凤凰端子) (16)	23
2.4.17 J32 CAN 连接头 (17)	24
2.4.18 J39 SPI (18)	25
2.4.19 J37 GPIO (凤凰端子) (19)	25
2.4.20 J38 GPIO 排座 (20)	26
2.4.21 J24 麦克风连接头 (21)	26
2.4.22 J23 音频输出连接头 (22)	27
2.4.23 J18/J19 Wi-Fi & 蓝牙天线接口 (23)	27
2.4.24 J5 RTC 电池连接头 (24)	27
2.4.25 J31 调试接口 (25)	28
2.4.26 J4 风扇连接头 (26)	28
2.4.27 J14 电源按键连接头 (27)	29
2.4.28 J26 SATA 3.0 接口 (28)	29

2.4.29	J27 SATA 电源接头 (29)	30
2.4.30	J29 Micro SIM 卡槽 (30)	30
2.4.31	J3 Micro SD 卡槽 (31)	30
第 3 章	首次使用调试	31
3.1	串口接头	32
3.1.1	Android 系统	33
3.1.2	Ubuntu 和 Debian 系统	36
3.2	CAN	38
3.3	GPIO	39
3.3.1	Android 系统	39
3.3.2	Ubuntu 和 Debian 系统	40
3.4	看门狗计时器	41
第 4 章	Android 系统手册	42
4.1	启用开发者选项	43
4.2	在 Windows 主机上设置 ADB	44
4.3	通过 ADB 命令安装应用程序	46
4.4	Windows 环境下升级固件	47
第 5 章	Debian 系统手册	50
5.1	系统介绍	51
5.1.1	用户名与密码	51
5.1.2	系统信息	51
5.2	系统设置	52
5.2.1	语言设置	53
5.2.2	时间和日期设置	55
5.2.3	网络配置	56
5.3	Ubuntu 环境升级镜像	57
5.3.1	环境准备	57
5.3.2	烧写镜像	57
第 6 章	Ubuntu 系统手册	59
6.1	系统介绍	60
6.1.1	用户名与密码	60
6.1.2	系统信息	60
6.2	系统设置	61
6.2.1	语言	61
6.2.2	日期和时间	61
6.2.3	网络/Wi-Fi	61
6.2.4	用户	61
6.3	Ubuntu 环境升级镜像	62
6.3.1	环境准备	62
6.3.2	烧写镜像	62
6.4	ROS 教程	63
第 7 章	废弃处理与质保	64
7.1	废弃处理	65
7.2	质保	66

前言

感谢购买 VT-SBC-3588 单板计算机（“主板”或“产品”）。本手册旨在就产品的设置、操作及维护提供必要的指导和帮助。请仔细阅读本手册，并确保您在使用产品前已理解产品的结构和功能。

目标用户

本手册旨在提供给：

- 嵌入式软件开发人员
- 二次开发软件工程师
- 其他合格的技术人员

版权说明

成都万创科技股份有限公司（“万创”）保留本手册的所有权利，包括随时更改内容、形式、产品功能和规格的权利，恕不事先另行书面通知。您可访问 www.vantrontech.com.cn 获取本手册最新版本。

本手册中的商标和注册商标均为其各自所有者的财产。本手册的任何部分均不得复制、翻印、翻译或出售。未经万创事先书面同意，不得对本手册进行任何更改或将其用于其他用途。万创保留对本手册所有公开发布副本的权利。

免责声明

尽管已对本手册包含的所有信息进行了仔细检查，以确保其技术细节和印刷排版的准确性，但万创对因本手册的任何错误或特性造成的，或由于本手册或软件的不当使用造成的后果不承担任何责任。

产品额定功率或者特性发生变化时，或者发生重大结构变更时，我们会更换配件编号。产品规格如有变更，我们或不会另行通知。

技术支持与帮助

如您遇到本手册未曾提及的情况，请联系您的销售代表了解相关解决方案。请在来函中附上以下信息：

- 产品名称和订单编号；
- 关于相关问题的描述；
- 收到的报错信息，如有。

美国：Vantron Technology, Inc.

地址：48434 Milmont Drive, Fremont, CA 94538

电话：(650) 422-3128

邮箱：sales@vantrontech.com

中国：成都万创科技股份有限公司

地址：四川省成都市武侯区武科东三路9号1号楼6楼610045

电话：86-28-8512-3930/3931, 86-28-8515-7572/6320

邮箱：sales@vantrontech.com.cn

符号约定

本手册使用以下符号，提醒用户注意相关信息。

	提醒可能会造成潜在的系统损坏或人员伤害。
	提示重要信息或法规。

一般安全说明

产品应当由合格熟练的技术人员按照当地及/或国际电气规范和法规进行安装。为保证人身安全并防止产品损坏，请于产品安装和运行前，仔细阅读并遵守以下安全说明。请保留本手册，以供将来查阅。

- 请勿拆卸或以其他方式改装产品。此类行为可能造成发热、起火或人身伤害等其他损害，且导致产品保修失效。
- 保持产品远离加热器、散热器、发动机机壳等热源。
- 请勿将任何物品塞入产品，否则可能导致产品故障或烧坏。
- 为确保产品正常运行，防止产品过热，请勿阻挡产品通风口。
- 请使用提供或推荐的安装工具并遵守安装说明。
- 作业工具的使用或放置应当遵守此类工具的实施规程，避免产品短路。
- 检查产品前，请切断电源，避免出现人身伤害或产品损坏。

电缆和配件安全说明

- ⚠ 仅使用满足条件的电源。确保使用符合手册规定范围的供电电压。
- ⚠ 请确保合理放置电缆，避免受到挤压。
- ⚠ 产品包含纽扣电池，为实时时钟提供备用电源。因此，请在搬运或高温操作过程中避免电池短路。
- ⚠ 清洁说明：
 - 清洁前请关闭产品电源
 - 请勿使用喷雾清洁剂
 - 使用湿布进行清洁
 - 除非使用除尘器，否则请勿清洁裸露的电子组件
- ⚠ 出现以下故障时，请关闭电源并联系万创技术支持工程师：
 - 产品损坏
 - 温度过高
 - 根据手册检修后，故障仍然无法解决
- ⚠ 请勿在易燃易爆环境中使用：
 - 远离易燃易爆环境
 - 远离通电电路
 - 未经授权，不得拆开产品外壳拔掉电源之前，请勿更换零件某些情况下，拔掉电源后，产品仍有余电。因此，更换零件前，必须停止充电并等待产品完成放电。

第 1 章 引言

1.1 产品概述

万创 VT-SBC-3588 嵌入式单板计算机由瑞芯微（Rockchip）RK3588 全新一代旗舰 AIoT 芯片驱动，搭载八核六十四位 CPU、ARM Mali-G610 MP4 四核 GPU，内置 AI 加速器 NPU，能够提供 6 TOPS 算力，支持主流深度学习框架。随着行业技术的发展，工业机器人、自动化控制、无人机等工控市场无疑将对具有 AI 算力的产品产生旺盛需求，万创 VT-SBC-3588 单板计算机应运而生。

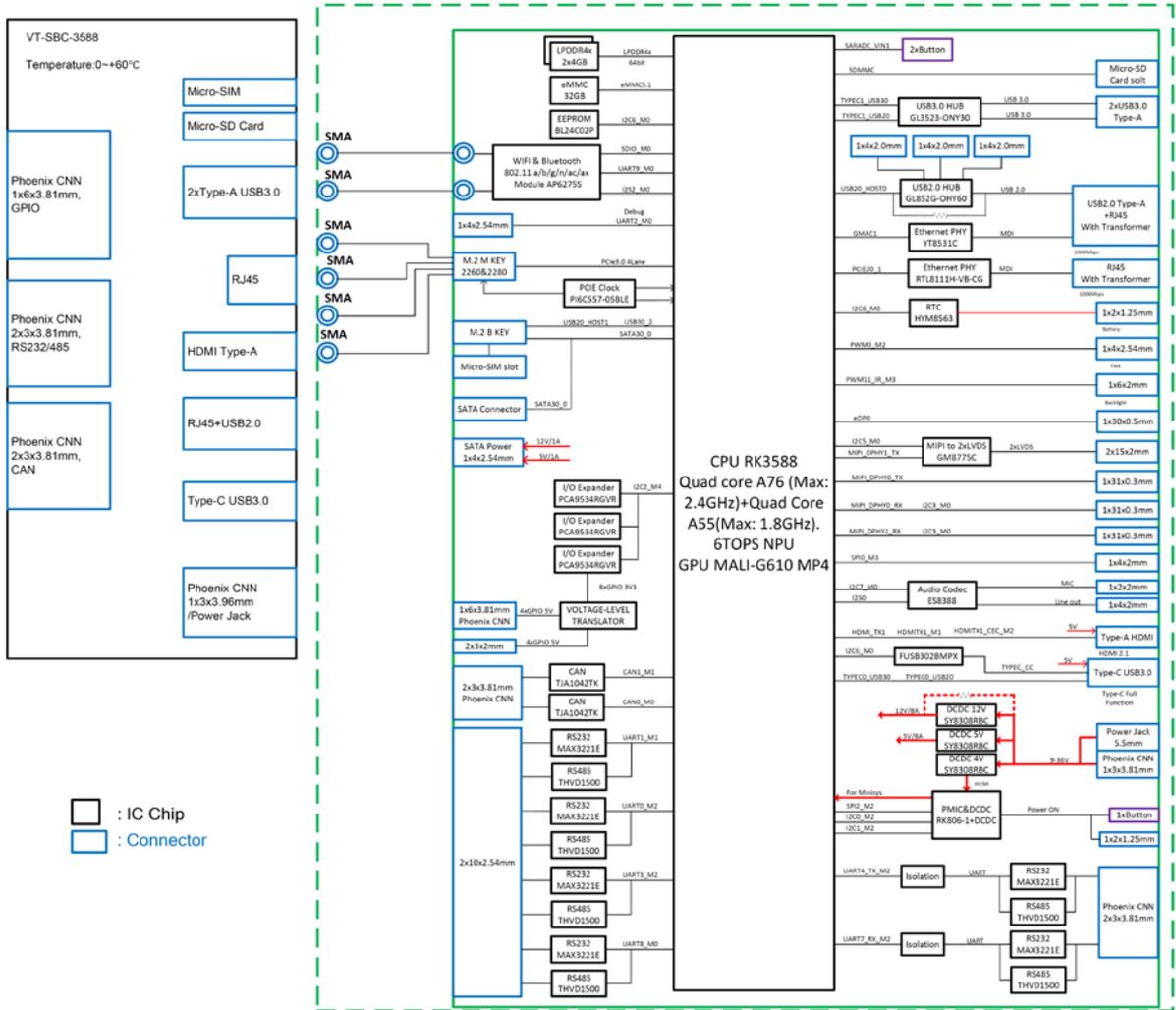
VT-SBC-3588 提供两个千兆网口，支持 2.4GHz/5GHz Wi-Fi 6 和蓝牙 5.0，且支持 M.2 扩展 4G/5G 通信，保证通信不受阻碍。支持 8K 视频编码解码，优化显示性能。

此外，VT-SBC-3588 提供丰富的外设接口，可接入大量外围设备，灵活用于 ARM PC、边缘计算、云服务器、智能 NVR 等领域。另外，产品提供不同的操作系统，可以保证用户系统环境的稳定性和安全性。

1.2 术语/缩写

术语	说明
NC	无连接
VCC	共集电极电压
GND	接地
/	低电平有效信号
+	差分信号正数
-	差分信号负数
I	电源输入
O	出站数据
I/O	输入/输出
P	电源/接地
A	模拟信号
OD	开源信号
CMOS	3.3V CMOS
LVC MOS	低电压 CMOS
LVTTL	低电压 TTL
CK/CLK	时钟（线）
PWM	脉宽调变
MISO	主输入从输出
MOSI	主输出从输入

1.3 框图



1.4 规格

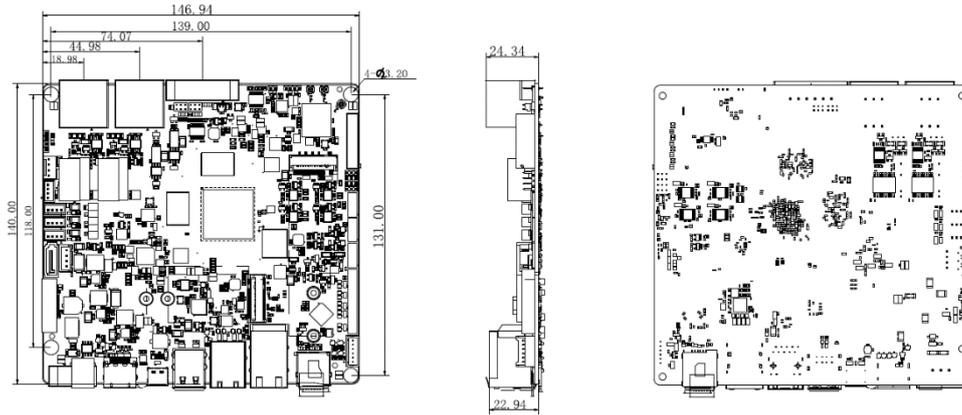
VT-SBC-3588			
系统	CPU	RK3588 四核 Cortex-A76 + 四核 Cortex-A55 处理器, 最高频率 2.4GHz	
	GPU	ARM Mali-G610 MP4, 最高频率 1GHz	
	NPU	6 TOPS	
	内存	LPDDR4 8GB (可选: 16GB)	
	存储	eMMC 32GB, 最大 128GB	1 x Micro SD 卡槽 支持 M.2 M-Key/M.2 B-Key 扩展 SSD (256GB ~ 1TB)
通信	以太网	2 x RJ45, 1000Mbps	
	4G/5G	支持 (由 M.2 B-Key 扩展)	
	Wi-Fi 及蓝牙	Wi-Fi 802.11 a/b/g/n/ac/ax + 蓝牙5.0	
多媒体	显示	1 x HDMI 2.1 (4096 x 2160 @60Hz) 1 x 双路LVDS (1920 x 1200 @60Hz)	1 x eDP (1920 x 1080 @60Hz) 1 x MIPI DSI (1920 x 1080 @60Hz)
	摄像头	2 x MIPI CSI	
	音频	1 x 麦克风连接头	1 x 音频连接头
输入/输出	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 连接头
	串口	2 x RS232/RS485 (凤凰端子) 4 x RS232/RS485 连接头	
	TTL UART	1 x TTL UART, 用于调试	
	风扇	1 x CPU 风扇连接头	
	SIM 卡槽	1 x SIM 卡槽	
	RTC	支持	
	看门狗	支持	
扩展	M.2 插槽	1 x M.2 M-Key (2260/2280), PCIe 3.0 x 4, 用于 SSD 扩展 1 x M.2 B-Key (2242/3052), USB3.0/SATA3.0, 用于 4G/5G/SSD 扩展	
	GPIO	4 x GPIO (凤凰端子)	3 x GPIO 连接头
	SPI	1 x SPI	
	CAN	2 x CAN (凤凰端子)	
软件	操作系统	Debian, Android, Ubuntu	
	语言	英语 (默认), 中文	
	设备管理平台 OTA 平台	BlueSphere MDM (仅适用 Android 设备) BlueSphere OTA	
电源 机械	输入	12V/5A, 24V/3A	1 x 电源端子
	尺寸	146.94mm x 140mm x 24.34mm	
环境	温度	工作温度: 0°C ~ +60°C	存储温度: -40°C ~ +85°C
	湿度	相对湿度 0~95% (无凝露)	
	认证	FCC, CCC	

1.5 操作系统

VT-SBC-3588 支持 Debian、Android 及 Ubuntu 操作系统。

1.6 机械尺寸

- 146.94mm x 140mm x 24.34mm



1.7 电源及功耗

VT-SBC-3588 输入电源为 12V/24V DC，通过主板上的电源端子供电。

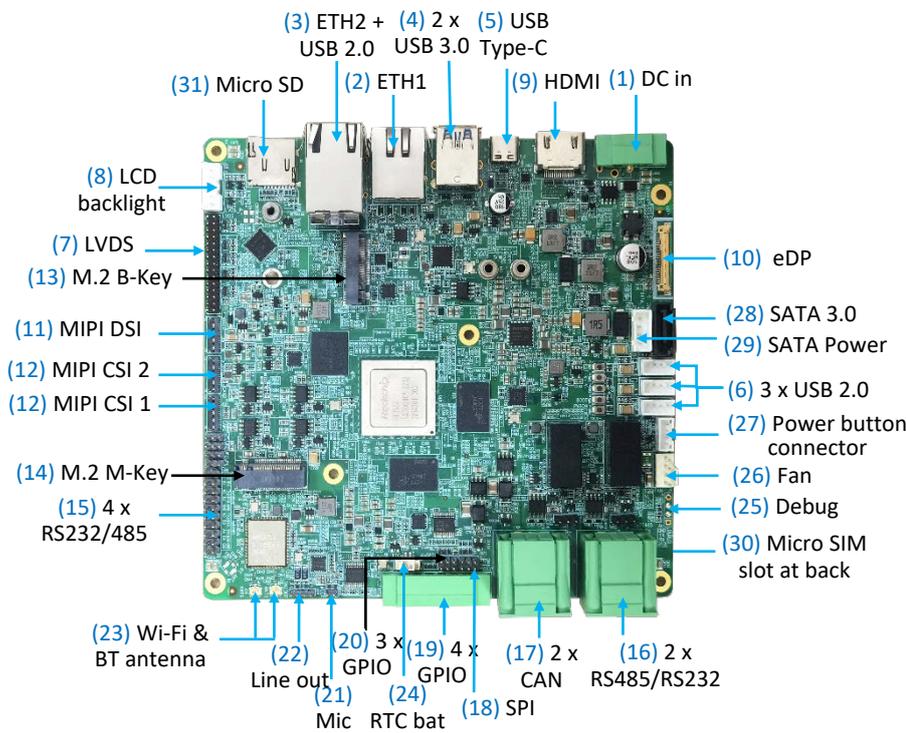
主板功耗在 30W 以内。需要指出的是，功耗在很大程度上是由主板的内存、存储容量和其他配置决定的。

1.8 环境参数

VT-SBC-3588 的工作温度范围为 0°C至+60°C，其存储温度范围为-40°C至+85°C，相对湿度范围不超过 95%（无凝露）。

第 2 章 接口说明

2.1 主板布局



 在 2.4 连接头和跳线一节，主板上的各个接口将按照上图标注的数字番号进行详细描述。

2.2 内存及存储

2.2.1 LPDDR4 内存

VT-SBC-3588 默认配置 8GB LPDDR4 内存，用户也可以选择 16GB 内存。

2.2.2 eMMC 闪存

VT-SBC-3588 提供最大 128GB 的 eMMC V5.1 闪存，默认为 32GB。其主要用作默认启动和存储设备。

2.2.3 扩展存储

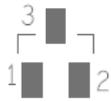
VT-SBC-3588 配置一个 Micro SD 卡槽，用于扩展存储。另外，还提供 M.2 M-Key/M.2 B-Key，支持 SSD 扩展存储。

2.3 识别第 1 引脚

除非另有说明，一般而言，连接器的第 1 引脚位于方形焊盘上，其他引脚则在圆形焊盘上。有时，第 1 引脚也靠近主板的三角形标记处。当一个连接器上有两行引脚时，第 1 引脚所在行的引脚编号均为奇数，另一行引脚编号则为偶数。



通常情况下，主板上连接器的引脚旁边会有数字或标记，表明引脚的位置。

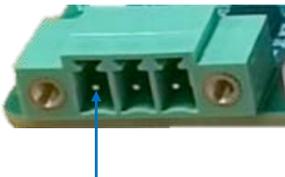


2.4 连接头和跳线

本节将简要介绍主板上的连接器/排针及相应的引脚位置。

2.4.1 J1 电源端子 (1)

VT-SBC-3588 支持 12V/24V 直流电源输入（推荐 12V 5A），通过一个 3 针电源端子（1 x 3 x 3.81mm）供电。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	GND	P	接地
2	-VDC	P	电源输入-
3	+VDC	P	电源输入+

2.4.2 J22 以太网口 (2)

VT-SBC-3588 配置了两个 RJ45 网口，支持 10M/100M/1000M Base-T4。每个网口上有两个 LED 指示灯，其中绿灯为连接指示灯，黄灯为信号指示灯。J22 为单独的网口，第二个网口与一个 USB 2.0 Type-A 接口集成在一起。



2.4.3 J21 以太网口 + USB 2.0 (3)

第二个网口与一个 USB 2.0 Type-A 接口堆叠配置。其中，USB 2.0 Type-A 接口最大输出为 5V/0.5A。



2.4.4 U27 USB 3.0 Type-A (4)

VT-SBC-3588 配置了两个 USB 3.0 Type-A 接口，可以连接外围设备。每个接口对外最大输出为 5V/1A。

2.4.5 J6 USB Type-C (5)

VT-SBC-3588 配置了一个 USB Type-C 接口，该接口支持 USB 3.0 OTG 和 DP 1.4 视频显示输出，其最大输出为 5V/1.5A。

2.4.6 J7/J8/J9 USB 2.0 Host 连接头 (6)

VT-SBC-3588 提供三个板载 USB 2.0 连接器（1 x 4 x 2.0mm），默认为 USB host 模式，每个连接头对外最大输出为 5V/0.5A。

引脚 1



J7 连接头引脚说明：

引脚编号	名称	类型	说明
1	HUB_HOST_1	P	5V 电源输出，限流 0.5A
2	HUB_HOST1_DM	I/O	USB Host1 DATA-
3	HUB_HOST1_DP	I/O	USB Host1 DATA+
4	GND	P	接地

J8 连接头引脚说明：

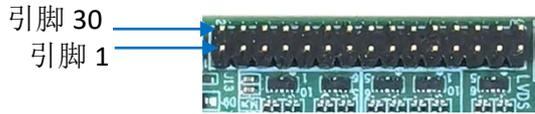
引脚编号	名称	类型	说明
1	HUB_HOST_2	P	5V 电源输出，限流 0.5A
2	HUB_HOST2_DM	I/O	USB Host2 DATA-
3	HUB_HOST2_DP	I/O	USB Host2 DATA+
4	GND	P	接地

J9 连接头引脚说明：

引脚编号	名称	类型	说明
1	HUB_HOST_3	P	5V 电源输出，限流 0.5A
2	HUB_HOST3_DM	I/O	USB Host3 DATA-
3	HUB_HOST3_DP	I/O	USB Host3 DATA+
4	GND	P	接地

2.4.7 J13 LVDS 接口 (7)

VT-SBC-3588 提供一个双通道 LVDS 接口 (2 x 15 x 2.0mm)，用于连接高分辨率显示屏，接口分辨率最高支持 1920 x 1200 @60Hz。为确保信号传输的可靠性，建议采用 LVDS 信号屏蔽双绞电子线进行连接。



连接器引脚说明：

引脚编号	名称	类型	说明
1	VCC5V0_LCD_LVDS	P	5V 电源输出
2	VCC5V0_LCD_LVDS	P	5V 电源输出
3	VCC5V0_LCD_LVDS	P	5V 电源输出
4	GND	P	接地
5	GND	P	接地
6	GND	P	接地
7	TXA0N	O	A 通道 LVDS 数据 0 路差分输出负端
8	TXA0P	O	A 通道 LVDS 数据 0 路差分输出正端
9	TXA1N	O	A 通道 LVDS 数据 1 路差分输出负端
10	TXA1P	O	A 通道 LVDS 数据 1 路差分输出正端
11	TXA2N	O	A 通道 LVDS 数据 2 路差分输出负端
12	TXA2P	O	A 通道 LVDS 数据 2 路差分输出正端
13	GND	P	接地
14	GND	P	接地
15	TXACN	O	A 通道 LVDS 时钟差分输出负端
16	TXACP	O	A 通道 LVDS 时钟差分输出正端
17	TXA3N	O	A 通道 LVDS 数据 3 路差分输出负端
18	TXA3P	O	A 通道 LVDS 数据 3 路差分输出正端
19	TXB0N	O	B 通道 LVDS 数据 0 路差分输出负端
20	TXB0P	O	B 通道 LVDS 数据 0 路差分输出正端
21	TXB1N	O	B 通道 LVDS 数据 1 路差分输出负端
22	TXB1P	O	B 通道 LVDS 数据 1 路差分输出正端
23	TXB2N	O	B 通道 LVDS 数据 2 路差分输出负端
24	TXB2P	O	B 通道 LVDS 数据 2 路差分输出正端
25	GND	P	接地
26	GND	P	接地
27	TXBCN	O	B 通道 LVDS 时钟差分输出负端
28	TXBCP	O	B 通道 LVDS 时钟差分输出正端
29	TXB3N	O	B 通道 LVDS 数据 3 路差分输出负端
30	TXB3P	O	B 通道 LVDS 数据 3 路差分输出正端

2.4.8 J15 背光接头 (8)

LVDS 接口旁配置了一个背光连接器 (1 x 6 x 2.0mm)，用于连接背光组件，为 LCD 屏幕背光供电。主要在光线不足的情况下提高 LCD 屏幕的可识别度。



引脚 1

引脚说明:

引脚编号	名称	类型	说明
1	VCC12V_LCD	P	12V 背光电源
2	VCC12V_LCD	P	12V 背光电源
3	PANEL_BKLTEN	O	LCD 背光电源控制输出, 5V 电平
4	PANEL_BL_PWM	O	LCD 背光亮度控制输出, 5V 电平
5	GND	P	接地
6	GND	P	接地

2.4.9 J9 HDMI 接口 (9)

VT-SBC-3588 的 HDMI 接口支持 HDMI 2.1，用于提供图像输出。接口分辨率最高支持 4096 x 2160 @60Hz。

2.4.10 J17 eDP 接口 (10)

VT-SBC-3588 提供一个 eDP 接口 (1 x 30 x 0.5mm)，用于提供图像输出，接口分辨率最高支持 1920 x 1080 @60Hz。为确保信号传输可靠性，建议采用 eDP 信号屏蔽双绞电子线进行连接。



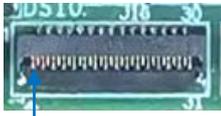
引脚 1

接口引脚说明见下页。

引脚编号	名称	类型	说明
1	VCC3V3_EDP	P	3.3V 电源输出
2	VCC3V3_EDP	P	3.3V 电源输出
3	VCC3V3_EDP	P	3.3V 电源输出
4	NC		
5	GND	P	接地
6	eDP0_TX_D0N	O	eDP 发送差分数据通路 0 -
7	eDP0_TX_D0P	O	eDP 发送差分数据通路 0 +
8	GND	P	接地
9	eDP0_TX_D1N	O	eDP 发送差分数据通路 1 -
10	eDP0_TX_D1P	O	eDP 发送差分数据通路 1 +
11	GND	P	接地
12	eDP0_TX_D2N	O	eDP 发送差分数据通路 2 -
13	eDP0_TX_D2P	O	eDP 发送差分数据通路 2 +
14	GND	P	接地
15	eDP0_TX_D3N	O	eDP 发送差分数据通路 3 -
16	eDP0_TX_D3P	O	eDP 发送差分数据通路 3 +
17	GND	P	接地
18	eDP_TX_AUXN		eDP 旁带差分数据通路 -
19	eDP_TX_AUXP		eDP 旁带差分数据通路 +
20	GND	P	接地
21	GND	P	接地
22	LCD_BL_EN_H	O	LCD 背光控制输出，3.3V 电平
23	LCD_BL_PWM1	O	LCD 背光亮度控制输出，3.3V 电平
24	GND	P	接地
25	GND	P	接地
26	GND	P	接地
27	VCC12V_LCD_EDP	P	12V 背光电源输出
28	VCC12V_LCD_EDP	P	12V 背光电源输出
29	VCC12V_LCD_EDP	P	12V 背光电源输出
30	VCC12V_LCD_EDP	P	12V 背光电源输出

2.4.11 J16 MIPI DSI 接口 (11)

VT-SBC-3588 提供一个 MIPI DSI 接口 (1 x 31 x 0.3mm)，用于连接显示设备，接口最高分辨率支持 1920 x 1080 @60Hz。



引脚 1

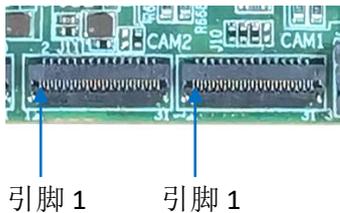
引脚说明:

引脚编号	名称	类型	说明
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	接地
10	GND	P	接地
11	MIPI_DPHY0_TX_D2P	O	MIPI DPHY0 发送差分数据通路 2 +
12	MIPI_DPHY0_TX_D2N	O	MIPI DPHY0 发送差分数据通路 2 -
13	GND	P	接地
14	MIPI_DPHY0_TX_D1P	O	MIPI DPHY0 发送差分数据通路 1 +
15	MIPI_DPHY0_TX_D1N	O	MIPI DPHY0 发送差分数据通路 1 -
16	GND	P	接地
17	MIPI_DPHY0_TX_CLKP	O	MIPI DPHY0 发送差分时钟 +
18	MIPI_DPHY0_TX_CLKN	I/O	MIPI DPHY0 发送差分时钟 -
19	GND	P	接地
20	MIPI_DPHY0_TX_D0P	O	MIPI DPHY0 发送差分数据通路 0 +
21	MIPI_DPHY0_TX_D0N	O	MIPI DPHY0 发送差分数据通路 0 -
22	GND	P	接地
23	MIPI_DPHY0_TX_D3P	O	MIPI DPHY0 发送差分数据通路 3 +
24	MIPI_DPHY0_TX_D3N	O	MIPI DPHY0 发送差分数据通路 3 -
25	GND	P	接地
26	NC		
27	MIPI_RST_L	O	MIPI LCD 复位信号输出, 1.8V
28	NC		
29	VCC1V8_LCD	P	1.8V 电源输出

30	VCC3V3_LCD	P	3.3V 电源输出
31	VCC3V3_LCD	P	3.3V 电源输出

2.4.12 J10/J11 MIPI CSI 接口 (12)

VT-SBC-3588 提供两个 MIPI CSI 接口 (1 x 31 x 0.3mm)，用于连接摄像头。



J10 (CAM 1) 引脚说明:

引脚编号	名称	类型	说明
1	GND	P	接地
2	MIPI_DPHY0_RX_D3N	I	MIPI DPHY0 接收差分数据通路 3 -
3	MIPI_DPHY0_RX_D3P	I	MIPI DPHY0 接收差分数据通路 3 +
4	GND	P	接地
5	MIPI_DPHY0_RX_D2N	I	MIPI DPHY0 接收差分数据通路 2 -
6	MIPI_DPHY0_RX_D2P	I	MIPI DPHY0 接收差分数据通路 2 +
7	GND	P	接地
8	MIPI_DPHY0_RX_D1N	I	MIPI DPHY0 接收差分数据通路 1 -
9	MIPI_DPHY0_RX_D1P	I	MIPI DPHY0 接收差分数据通路 1 +
10	GND	P	接地
11	MIPI_DPHY0_RX_D0N	I	MIPI DPHY0 接收差分数据通路 0 -
12	MIPI_DPHY0_RX_D0P	I	MIPI DPHY0 接收差分数据通路 0 +
13	GND	P	接地
14	MIPI_DPHY0_RX_CLKN	I	MIPI DPHY0 接收差分时钟 -
15	MIPI_DPHY0_RX_CLKP	I	MIPI DPHY0 接收差分时钟 +
16	GND	P	接地
17	I2C3_SCL_M0_CAM1	O	I2C 串行时钟, 电压域 1.8V
18	I2C3_SDA_M0_CAM1	I/O	I2C 串行数据, 电压域 1.8V
19	MIPI_CAM1_RESET	O	CAM1 复位输出, 电压域 1.8V
20	MIPI_CAM1_PDN	O	CAM1 断电控制输出, 电压域 1.8V
21	GND	P	接地
22	MIPI_CAM1_MCLK	O	CAM1 主时钟输出, 电压域 1.8V
23	GND	P	接地
24	NC		
25	VCC_1V8_CAM1	P	1.8V 电源输出
26	VCC_1V8_CAM1	P	1.8V 电源输出

27	VCC_1V2_CAM1	P	1.2V 电源输出，硬件支持电压调整
28	VCC_2V8_AF_CAM1	P	2.8V 马达电源输出
29	VCC_2V8_CAM1	P	2.8V 电源输出
30	NC		
31	GND	P	接地

J11 (CAM 2) 引脚说明:

引脚编号	名称	类型	说明
1	GND	P	接地
2	MIPI_DPHY1_RX_D3N	I	MIPI DPHY1 接收差分数据通路 3 -
3	MIPI_DPHY1_RX_D3P	I	MIPI DPHY1 接收差分数据通路 3 +
4	GND	P	接地
5	MIPI_DPHY1_RX_D2N	I	MIPI DPHY1 接收差分数据通路 2 -
6	MIPI_DPHY1_RX_D2P	I	MIPI DPHY1 接收差分数据通路 2 +
7	GND	P	接地
8	MIPI_DPHY1_RX_D1N	I	MIPI DPHY1 接收差分数据通路 1 -
9	MIPI_DPHY1_RX_D1P	I	MIPI DPHY1 接收差分数据通路 1 +
10	GND	P	接地
11	MIPI_DPHY1_RX_D0N	I	MIPI DPHY1 接收差分数据通路 0 -
12	MIPI_DPHY1_RX_D0P	I	MIPI DPHY1 接收差分数据通路 0 +
13	GND	P	接地
14	MIPI_DPHY1_RX_CLKN	I	MIPI DPHY1 接收差分时钟 -
15	MIPI_DPHY1_RX_CLKP	I	MIPI DPHY1 接收差分时钟 +
16	GND	P	接地
17	I2C3_SCL_M0_CAM2	O	I2C 串行时钟, 电压域 1.8V
18	I2C3_SDA_M0_CAM2	I/O	I2C 串行数据, 电压域 1.8V
19	MIPI_CAM2_RESET	O	CAM2 复位输出, 电压域 1.8V
20	MIPI_CAM2_PDN	O	CAM2 断电控制输出, 电压域 1.8V
21	GND	P	接地
22	MIPI_CAM2_MCLK	O	CAM2 主时钟输出, 电压域 1.8V
23	GND	P	接地
24	NC		
25	VCC_1V8_CAM2	P	1.8V 电源输出
26	VCC_1V8_CAM2	P	1.8V 电源输出
27	VCC_1V2_CAM2	P	1.2V 电源输出，硬件支持电压调整
28	VCC_2V8_AF_CAM2	P	2.8V 马达电源输出
29	VCC_2V8_CAM2	P	2.8V 电源输出
30	NC		
31	GND	P	接地

2.4.13 J28 M.2 B-Key 插槽 (13)

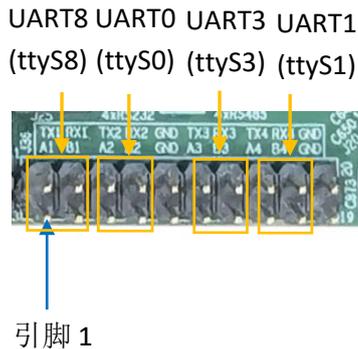
VT-SBC-3588 提供一个 M.2 B-Key 插槽，支持 2242 和 3052 两种尺寸的外围设备，可支持 SATA 3.0（与 J26 接口的 SATA 信号为二选一的关系）和 USB 3.0 总线，分别用于连接 SSD 扩展存储，以及连接 4G/5G 模块，进行无线通信。

2.4.14 J25 M.2 M-Key 插槽 (14)

VT-SBC-3588 提供一个 M.2 M-Key 插槽，支持 2260 和 2280 两种尺寸的外围设备，可支持 PCIe 3.0 x 4 总线，用于连接 SSD 扩展存储。

2.4.15 J36 RS485/RS232 连接头 (15)

VT-SBC-3588 配置 4 个 RS232/RS485 复用串口接头（2 x 10 x 2mm），从左至右分别为 UART8、UART0、UART3 和 UART1，对应的软件名称分别为/dev/ttyS8、/dev/ttyS0、/dev/ttyS3 和/dev/ttyS1，复用串口通过软件实现串口模式切换。

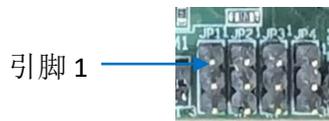


J36 RS485/RS232 复用接头引脚说明：

引脚编号	名称	类型	说明
1	485_A_1		RS485 Bus1 I/O 接口, A
2	232_TX_1	O	RS232_1 数据输出
3	485_B_1		RS485 Bus1 I/O 接口, B
4	232_RX_1	I	RS232_1 数据输入
5	485_A_2		RS485 Bus2 I/O 接口, A
6	232_TX_2	O	RS232_2 数据输出
7	485_B_2		RS485 Bus2 I/O 接口, B
8	232_RX_2	I	RS232_2 数据输入
9	GND	P	接地
10	GND	P	接地
11	485_A_3		RS485 Bus3 I/O 接口, A
12	232_TX_3	O	RS232_3 数据输出

13	485_B_3		RS485 Bus3 I/O 接口, B
14	232_RX_3	I	RS232_3 数据输入
15	485_A_4		RS485 Bus4 I/O 接口, A
16	232_TX_4	O	RS232_4 数据输出
17	485_B_4		RS485 Bus4 I/O 接口, B
18	232_RX_4	I	RS232_4 数据输入
19	GND	P	接地
20	GND	P	接地

当串口为 RS485 模式时，串口左侧的 JP1、JP2、JP3、JP4 端接电阻连接头分别用于连接 RS485_1、RS485_2、RS485_3、RS485_4 的端接电阻（120Ω），防止网络两端信号反射，提高信号质量。



JP1 端接电阻连接头引脚说明：

引脚编号	名称	类型	说明
1	485_B_1		RS485 Bus1 I/O 接口, B.串联 120Ω 电阻
2	485_A_1		RS485 Bus1 I/O 接口, A
3	NC		

JP2 端接电阻连接头引脚说明：

引脚编号	名称	类型	说明
1	485_B_2		RS485 Bus2 I/O 接口, B.串联 120Ω 电阻
2	485_A_2		RS485 Bus2 I/O 接口, A
3	NC		

JP3 端接电阻连接头引脚说明：

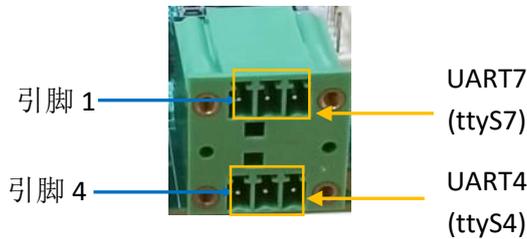
引脚编号	名称	类型	说明
1	485_B_3		RS485 Bus3 I/O 接口, B.串联 120Ω 电阻
2	485_A_3		RS485 Bus3 I/O 接口, A
3	NC		

JP4 端接电阻连接头引脚说明：

引脚编号	名称	类型	说明
1	485_B_4		RS485 Bus4 I/O 接口, B.串联 120Ω 电阻
2	485_A_4		RS485 Bus4 I/O 接口, A
3	NC		

2.4.16 J34 RS232/RS485（凤凰端子）(16)

VT-SBC-3588 提供一个 6 针凤凰端子（2 x 3 x 3.81mm），配置为两个隔离的 RS232/RS485 复用串口接头，分别为 UART4 和 UART7，对应的软件名称分别为 /dev/ttyS7 和 /dev/ttyS4。



引脚说明：

引脚编号	名称	类型	说明
1	RS485_A_2 / RS232_TX2		RS485 Bus2 I/O 接口, A / RS232_2 数据输出
2	RS485_B_2/ RS232_RX2		RS485 Bus2 I/O 接口, B / RS232_2 数据输入
3	ISO_GND_2	P	RS485_2/ RS232_2 隔离地 2
4	RS485_A_1/ RS232_TX1		RS485 Bus1 I/O 接口, A / RS232_1 数据输出
5	RS485_B_1/ RS232_RX1		RS485 Bus1 I/O 接口, B / RS232_1 数据输入
6	ISO_GND_1	P	RS485_1/ RS232_1 隔离地 1

当串口为 RS485 模式时，凤凰端子后方的 J33 和 J35 端接电阻连接头分别用于连接 UART 4 和 UART 7 的端接电阻（120Ω），防止网络两端信号反射，提高信号质量。



引脚 1

J33 端接电阻连接头引脚说明：

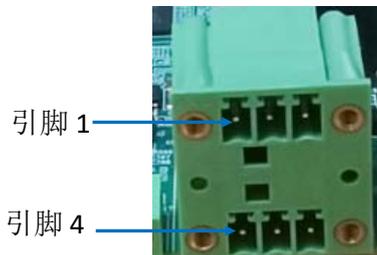
引脚编号	名称	类型	说明
1	RS485_B_1		RS485 Bus1 I/O 接口, B.串联 120Ω 电阻
2	RS485_A_1		RS485 Bus1 I/O 接口, A
3	NC		

J35 端接电阻连接头引脚说明：

引脚编号	名称	类型	说明
1	RS485_B_2		RS485 Bus2 I/O 接口, B.串联 120Ω 电阻
2	RS485_A_2		RS485 Bus2 I/O 接口, A
3	NC		

2.4.17 J32 CAN 连接头 (17)

VT-SBC-3588 另外提供了一个 6 针凤凰端子 (2 x 3 x 3.81mm)，配置为两个 CAN 接口。



引脚说明：

引脚编号	名称	类型	说明
1	CAN0_H		高电平 CAN0 总线
2	CAN0_L		低电平 CAN0 总线
3	GND	P	接地
4	CAN1_H		高电平 CAN1 总线
5	CAN1_L	P	低电平 CAN1 总线
6	GND	P	接地

2.4.18 J39 SPI (18)

VT-SBC-3588 配置一个 SPI 接头（2x3x2mm），用于连接外围设备，进行数据传输和通信。

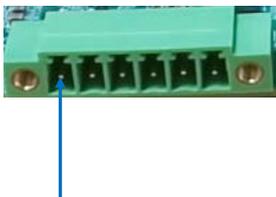


引脚说明：

引脚编号	名称	类型	说明
1	SPIO_MISO_M3_3V3	I/O	SPI MISO, 3.3V 电平
2	SPIO_MOSI_M3_3V3	I/O	SPI MOSI, 3.3V 电平
3	SPIO_CLK_M3_3V3	I/O	SPI CLK, 3.3V 电平
4	SPIO_CS0_M3_3V3	I/O	SPI CS, 3.3V 电平
5	VCC_3V3_S0	P	3.3V 电源输出
6	GND	P	接地

2.4.19 J37 GPIO（凤凰端子）(19)

VT-SBC-3588 凤凰端子上的 4 个 GPIO 接口（1 x 6 x 3.81mm），便于用户连接和控制外部设备。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	GPIO8_5V	I/O	GPIO 5V 电平
2	GPIO7_5V	I/O	GPIO 5V 电平
3	GPIO6_5V	I/O	GPIO 5V 电平
4	GPIO5_5V	I/O	GPIO 5V 电平
5	VCC_5V0	P	5V 电源输出
6	GND	P	接地

2.4.20 J38 GPIO 排座 (20)

VT-SBC-3588 另外提供一个 GPIO 排座，支持 3 路 GPIO，其功能可以根据需求自定义。



引脚说明：

引脚编号	信号	类型	说明
1	GPIO4_5V	I/O	GPIO 5V 电平
2	GPIO3_5V	I/O	GPIO 5V 电平
3	GPIO2_5V	I/O	GPIO 5V 电平
4	NC		
5	VCC_5V0	P	5V 电源输出
6	GND	P	接地

2.4.21 J24 麦克风连接头 (21)

VT-SBC-3588 提供一个麦克风连接头 (1 x 2 x 2mm)，实现音频输入。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	MIC2P	I	MIC 信号输入
2	MIC2N	I	MIC 信号输入

2.4.22 J23 音频输出连接头 (22)

麦克风连接头旁边为音频输出连接头（1 x 4 x 2mm），用于连接扬声器，实现音频输出。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	GND	P	接地
2	ROUT1	O	右声道输出
3	LOUT1	O	左声道输出
4	GND	P	接地

2.4.23 J18/J19 Wi-Fi & 蓝牙天线接口 (23)

主板上，J18 为 Wi-Fi（2.4G/5GHz）天线接口，J19 是 Wi-Fi（2.4G/5GHz）和蓝牙天线接口，上述接口为公插针接口。

2.4.24 J5 RTC 电池连接头 (24)

主板通过 RTC 电池连接头连接电池，为 RTC 供电。推荐使用 3V RTC 电池。



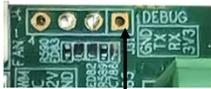
引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	RTC_P	PI	RTC 电源输入，3V
2	GND	P	接地

2.4.25 J31 调试接口 (25)

VT-SBC-3588 配置一个调试串口，用于主板调试或配置。使用时，建议将串口波特率参数配置为默认的 1.5Mbps。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	VCC_3V3_S3	P	3.3V 电源输出
2	UART2_RX_M0_DEBUG	I	串口接收，默认波特率 1.5Mbps，TTL 3.3V
3	UART2_TX_M0_DEBUG	O	串口发送，默认波特率 1.5Mbps，TTL 3.3V
4	GND	P	接地

2.4.26 J4 风扇接头 (26)

VT-SBC-3588 配置一个风扇接头（1 x 4 x 2.54mm），用于连接风扇，进行主动散热。推荐风扇功率要小于 6W（12V/0.5A）。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	GND	P	接地
2	FAN_12V	PO	12V 电源输出，限流 0.5A
3	NC		
4	PWM0_M2_FAN	O	PWM 信号输出，电压域 5V

2.4.27 J14 电源按键连接头 (27)

主板上的电源按键连接头（1 x 4 x 2mm）用于提供电源按键控制信号，实现电源控制。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	LED+	P	LED +
2	GND	P	接地
3	PWRON	I	电源按键控制信号输入
4	GND	P	接地

2.4.28 J26 SATA 3.0 接口 (28)

SATA 3.0 接口可以连接 SSD 卡，用于扩展存储。此接口的 SATA 信号与 M.2 B-Key 插槽的 SATA 信号是复用关系。若选择使用此接口的 SATA 信号，则 M.2 B-Key 插槽的 SATA 信号将无法使用；若选择使用 M.2 B-Key 插槽的 SATA 信号，则此接口的 SATA 信号将无法使用。

2.4.29 J27 SATA 电源接头 (29)

VT-SBC-3588 配置一个 SATA 电源接头（1 x 4 x 2.54mm），用于为 SATA 存储设备提供电源，保证存储性能高效、可靠。



引脚 1

引脚说明：

引脚编号	信号	类型	说明
1	VCC_5V0	P	5V 电源输出
2	GND	P	接地
3	GND	P	接地
4	VCC12V_DCIN	P	12V 电源输出

2.4.30 J29 Micro SIM 卡槽 (30)

VT-SBC-3588 背面提供一个 Micro SIM 卡槽。规格：Micro SIM，自弹式，支持热插拔。

2.4.31 J3 Micro SD 卡槽 (31)

VT-SBC-3588 提供一个 Micro SD 卡槽，最大支持 128GB 的存储设备。

第 3 章 首次使用调试

本章主要针对接口或软件应用的首次使用调试进行说明。如果主板运行 Linux 系统，您可以使用鼠标、键盘和显示器连接 VT-SBC-3588，直接在设备控制台中进行调试，或者，也可以使用 USB Type-A 转 Type-C 数据线连接 VT-SBC-3588 和主控电脑，进行远程调试。

3.1 串口接头

VT-SBC-3588 提供一个凤凰端子，配置为两个 RS232/RS485 复用串口接头（UART4 和 UART7），在软件系统中分别映射为/dev/ttyS4 和 ttyS7。另外还提供四个 RS232/RS485 复用接头（UART8、UART0、UART3 和 UART1），分别对应/dev/ttyS8、/dev/ttyS0、/dev/ttyS3、/dev/ttyS1。

请参考 2.4.15 和 2.4.16，了解上述接头的引脚说明。



用户可以使用合适的 USB 转串口适配器连接主板的串口和主控电脑（RS232 接线：TX-RX, RX-TX, GND-GND; RS485 接线：A-A, B-B, GND-GND），然后在主控电脑上使用串口工具进行串口通信，测试串口功能是否正常。

或者，也可以交叉连接任意两个串口，如果设备运行 Debian 或 Ubuntu 系统，则直接在设备控制台执行命令，并查看命令运行结果；如果设备运行 Android 系统，则请连接设备和主控电脑并在主控电脑上执行命令。

3.1.1 Android 系统

主板运行 Android 系统时，两个串口在不同模式下(RS232、RS485) 的收发命令汇总如下：

RS232 模式		
UART 0	发送数据	# echo 0 > /sys/class/gpio/gpio504/value # echo 1 > /sys/class/gpio/gpio503/value # echo "Test string" > /dev/ttyS0
	接收数据	# echo 0 > /sys/class/gpio/gpio504/value # echo 1 > /sys/class/gpio/gpio503/value # cat /dev/ttyS0
UART 1	发送数据	# echo 0 > /sys/class/gpio/gpio508/value # echo 1 > /sys/class/gpio/gpio507/value # echo "Test string" > /dev/ttyS1
	接收数据	# echo 0 > /sys/class/gpio/gpio508/value # echo 1 > /sys/class/gpio/gpio507/value # cat /dev/ttyS1
UART 3	发送数据	# echo 0 > /sys/class/gpio/gpio506/value # echo 1 > /sys/class/gpio/gpio505/value # echo "Test string" > /dev/ttyS3
	接收数据	# echo 0 > /sys/class/gpio/gpio506/value # echo 1 > /sys/class/gpio/gpio505/value # cat /dev/ttyS3
UART 4	发送数据	# echo 0 > /sys/class/gpio/gpio497/value # echo 1 > /sys/class/gpio/gpio498/value # echo "Test string" > /dev/ttyS4
	接收数据	# echo 0 > /sys/class/gpio/gpio497/value # echo 1 > /sys/class/gpio/gpio498/value # cat /dev/ttyS4
UART 7	发送数据	# echo 0 > /sys/class/gpio/gpio499/value # echo 1 > /sys/class/gpio/gpio500/value # echo "Test string" > /dev/ttyS7
	接收数据	# echo 0 > /sys/class/gpio/gpio499/value # echo 1 > /sys/class/gpio/gpio500/value # cat /dev/ttyS7
UART 8	发送数据	# echo 0 > /sys/class/gpio/gpio502/value # echo 1 > /sys/class/gpio/gpio501/value # echo "Test string" > /dev/ttyS8
	接收数据	# echo 0 > /sys/class/gpio/gpio502/value # echo 1 > /sys/class/gpio/gpio501/value # cat /dev/ttyS8

RS485 模式		
UART 0	发送数据	# echo 0 > /sys/class/gpio/gpio504/value # echo 0 > /sys/class/gpio/gpio503/value # echo "Test string" > /dev/ttyS0
	接收数据	# echo 0 > /sys/class/gpio/gpio504/value # echo 0 > /sys/class/gpio/gpio503/value # cat /dev/ttyS0
UART 1	发送数据	# echo 0 > /sys/class/gpio/gpio508/value # echo 0 > /sys/class/gpio/gpio507/value # echo "Test string" > /dev/ttyS1
	接收数据	# echo 0 > /sys/class/gpio/gpio508/value # echo 0 > /sys/class/gpio/gpio507/value # cat /dev/ttyS1
UART 3	发送数据	# echo 0 > /sys/class/gpio/gpio506/value # echo 0 > /sys/class/gpio/gpio505/value # echo "Test string" > /dev/ttyS3
	接收数据	# echo 0 > /sys/class/gpio/gpio506/value # echo 0 > /sys/class/gpio/gpio505/value # cat /dev/ttyS3
UART 4	发送数据	# echo 0 > /sys/class/gpio/gpio497/value # echo 0 > /sys/class/gpio/gpio498/value # echo "Test string" > /dev/ttyS4
	接收数据	# echo 0 > /sys/class/gpio/gpio497/value # echo 0 > /sys/class/gpio/gpio498/value # cat /dev/ttyS4
UART 7	发送数据	# echo 0 > /sys/class/gpio/gpio499/value # echo 0 > /sys/class/gpio/gpio500/value # echo "Test string" > /dev/ttyS7
	接收数据	# echo 0 > /sys/class/gpio/gpio499/value # echo 0 > /sys/class/gpio/gpio500/value # cat /dev/ttyS7
UART 8	发送数据	# echo 0 > /sys/class/gpio/gpio502/value # echo 0 > /sys/class/gpio/gpio501/value # echo "Test string" > /dev/ttyS8
	接收数据	# echo 0 > /sys/class/gpio/gpio502/value # echo 0 > /sys/class/gpio/gpio501/value # cat /dev/ttyS8

下文的示例说明了如何在 RS232 模式下测试串口的功能是否正常：

1. 交叉连接任意两个串口（TX-RX, RX-TX, GND-GND）；



2. 打开一个终端，使用其中一个串口（如 UART 7）接收数据；

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

3. 打开另一个终端，并使用另一个串口（UART 4）发送数据；

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

4. 第一个终端将打印接收到的数据；
5. 参考上页的表格，了解不同模式下，不同数据传输动作所使用的命令。

本例中，用户也可以使用单个串口进行数据收发。

3.1.2 Ubuntu 和 Debian 系统

主板运行 Ubuntu 或 Debian 系统时，两个串口在不同模式下(RS232、RS485) 的收发命令汇总如下：

RS232 模式		
UART 0	发送数据	# gpiochip6 3=0 # gpiochip6 2=1 # echo "Test string" > /dev/ttyS0
	接收数据	# gpiochip6 3=0 # gpiochip6 2=1 # cat /dev/ttyS0
UART 1	发送数据	# gpiochip6 7=0 # gpiochip6 6=1 # echo "Test string" > /dev/ttyS1
	接收数据	# gpiochip6 7=0 # gpiochip6 6=1 # cat /dev/ttyS1
UART 3	发送数据	# gpiochip6 5=0 # gpiochip6 4=1 # echo "Test string" > /dev/ttyS3
	接收数据	# gpiochip6 5=0 # gpiochip6 4=1 # cat /dev/ttyS3
UART 4	发送数据	# gpiochip7 4=0 # gpiochip7 5=1 # echo "Test string" > /dev/ttyS4
	接收数据	# gpiochip7 4=0 # gpiochip7 5=1 # cat /dev/ttyS4
UART 7	发送数据	# gpiochip7 6=0 # gpiochip7 7=1 # echo "Test string" > /dev/ttyS7
	接收数据	# gpiochip7 6=0 # gpiochip7 7=1 # cat /dev/ttyS7
UART 8	发送数据	# gpiochip6 1=0 # gpiochip6 0=1 # echo "Test string" > /dev/ttyS8
	接收数据	# gpiochip6 1=0 # gpiochip6 0=1 # cat /dev/ttyS8

下文的示例说明了如何在 RS232 模式下测试串口的功能是否正常：

1. 按照下图所示，交叉连接两个串口（TX-RX, RX-TX, GND-GND）；



2. 打开一个终端，使用其中一个串口（如 UART 7）接收数据；

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

3. 打开另一个终端，并使用另一个串口（UART 4）发送数据；

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

4. 第一个终端将打印接收到的数据；
5. 参考上页的表格，了解不同模式下，不同数据传输动作所使用的命令。

3.2 CAN

设备配置了两个 CAN 总线接头（CAN0 和 CAN1）。用户可以交叉连接两个接头（H-H, L-L, GND-GND），然后使用其中一个接头发送数据，另外的接头接收数据，测试接头功能是正常。请参考 2.4.17，了解接头引脚定义。

- 使用 CAN0 接收数据，CAN1 发送数据：

1. 打开终端设置 CAN0 的波特率，并输入命令接收数据：

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

2. 打开另外一个终端，设置 CAN1 的波特率，并输入命令发送数据。

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

- 使用 CAN1 接收数据，CAN0 发送数据：

1. 打开终端设置 CAN1 的波特率，并输入命令接收数据：

```
# ip link set can1 up type can bitrate 125000 （若之前已设置，则略过此步）  
# candump can1
```

2. 打开另外一个终端，设置 CAN0 的波特率，并输入命令发送数据。

```
# ip link set can0 up type can bitrate 125000 （若之前已设置，则略过此步）  
# cansend can0 500#11.22.33.44
```

- 查看 CAN0 的状态：

```
# 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:/#
```

- 查看 CAN1 的状态：

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

3.3 GPIO

设备配置了 7 路 GPIO，包含凤凰端子上的 4 路和 6 针排座上的 3 路。GPIO2 ~ GPIO8 在软件系统中的编号分别为 gpio486、gpio487、gpio488、gpio489、gpio490、gpio491、gpio492。用户可以参考 2.4.19 和 2.4.20，了解各 GPIO 接口的引脚定义。

在终端输入以下命令，查看 GPIO 状态：

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

Android 系统下设置 GPIO 管脚高/低电平的命令与 Ubuntu 和 Debian 系统中使用的命令不同。用户可以参考后文总结的命令，设置管脚的高/低电平，然后使用电压表测试设置是否有效。

3.3.1 Android 系统

下表列出了 Android 系统下各 GPIO 接口管脚的高/低电平设置命令。

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

切换 GPIO 接口的方向：

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

3.3.2 Ubuntu 和 Debian 系统

下表列出了 Ubuntu 和 Debian 系统下各 GPIO 接口管脚的高/低电平设置命令。

GPIOs		
GPIO 2	高电平	# gpioset gpiochip8 1=1
	低电平	# gpioset gpiochip8 1=0
GPIO 3	高电平	# gpioset gpiochip8 2=1
	低电平	# gpioset gpiochip8 2=0
GPIO 4	高电平	# gpioset gpiochip8 3=1
	低电平	# gpioset gpiochip8 3=0
GPIO 5	高电平	# gpioset gpiochip8 4=1
	低电平	# gpioset gpiochip8 4=0
GPIO6	高电平	# gpioset gpiochip8 5=1
	低电平	# gpioset gpiochip8 5=0
GPIO7	高电平	# gpioset gpiochip8 6=1
	低电平	# gpioset gpiochip8 6=0
GPIO8	高电平	# gpioset gpiochip8 7=1
	低电平	# gpioset gpiochip8 7=0

3.4 看门狗计时器

看门狗计时器默认被禁用。开启后，看门狗计时器会在每 44 秒内与系统通信一次（“喂狗”）。如果通信中断，系统将在看门狗计时结束时重启，这也意味着系统或者应用出现故障。

1. 写入除大写字母‘V’以外的任意字符，开启看门狗;

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

2. 设置喂狗频率（如，22 秒）以及设备异常时自动重启的时间间隔（如，2 分钟）;

```
[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. 禁用看门狗;

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

4. 用户也可以通过“systemd”命令开启硬件看门狗，并按照上述说明进行配置。

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

第 4 章 Android 系统手册

前提条件：

- VT-SBC-3588
- Windows 主机
- VT-SBC-3588 软件发布包
- USB 鼠标、键盘和显示器，用于连接 VT-SBC-3588，实现便捷操作
- 12V/24V 适配器，用于启动主板
- USB Type-A 转 Type-C 数据线，用于连接设备和 Windows 主机

4.1 启用开发者选项

请根据以下步骤启用 VT-SBC-3588 的开发者选项：

1. 将鼠标、键盘、显示器与 VT-SBC-3588 相连，便于后续操作；
2. 系统启动后，上划进入应用抽屉；
3. 依次点击**设置 > 关于 <设备>**；
4. 向下滚动鼠标至**版本号**，连续点击版本号七次，打开**开发者选项**；
5. 回到**设置 > 系统 > 高级 > 开发者选项**，打开设备的 **USB 调试**，之后即可自定义 VT-SBC-3588 的设置。

 根据 Android 版本的不同，入口或许略有差异。

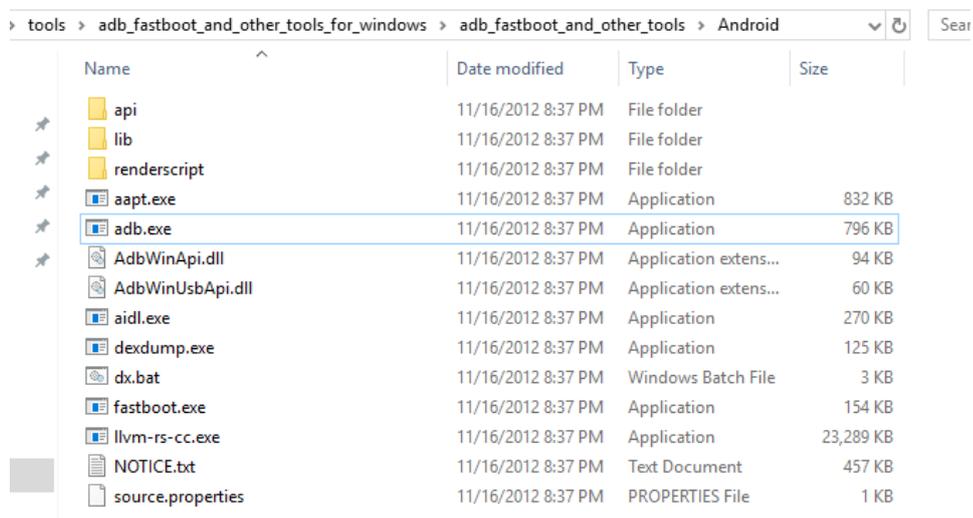
4.2 在 Windows 主机上设置 ADB

安卓调试桥（ADB）是一种用于直接连接用户开发工作站与其安卓设备的工具，用户可以通过该工具完成调试、设备升级、应用安装等。

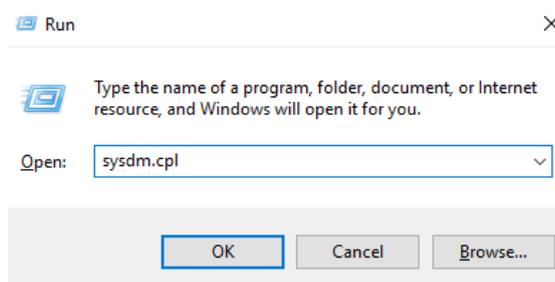
将 ADB 可执行文件添加至系统的环境变量后，无论用户当前工作路径如何，均可以直接在任意位置使用 ADB 工具。

请根据以下步骤在 Windows 主机上完成 ADB 设置。

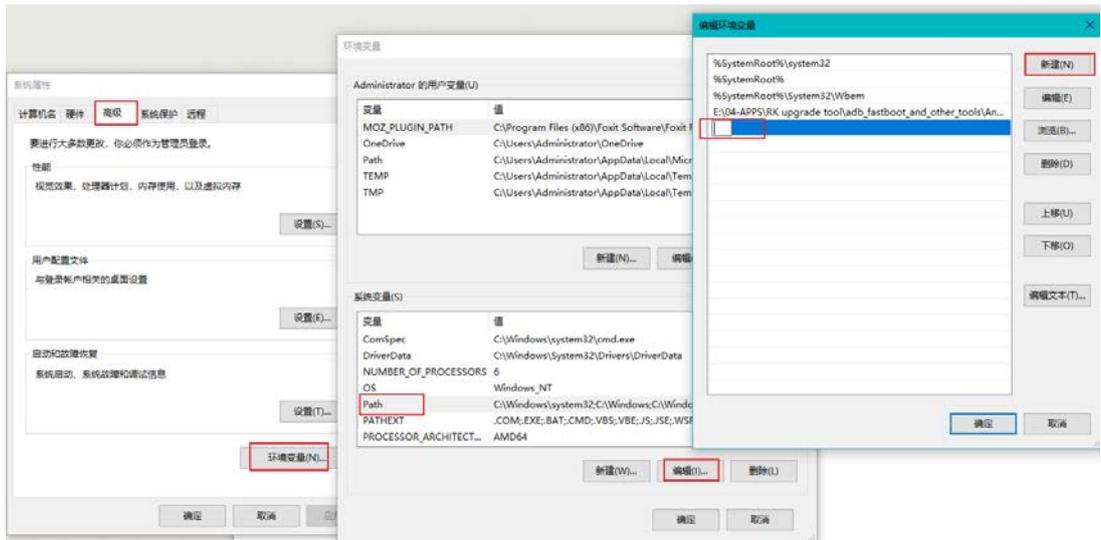
1. 将软件发布包解压缩，然后进入以下目录：\SW\tools；
2. 解压 **adb_fastboot_and_other_tools_for_windows** 文件夹；
3. **Android** 文件夹包含了 ADB 工具套件，导航至该文件夹并复制其路径；



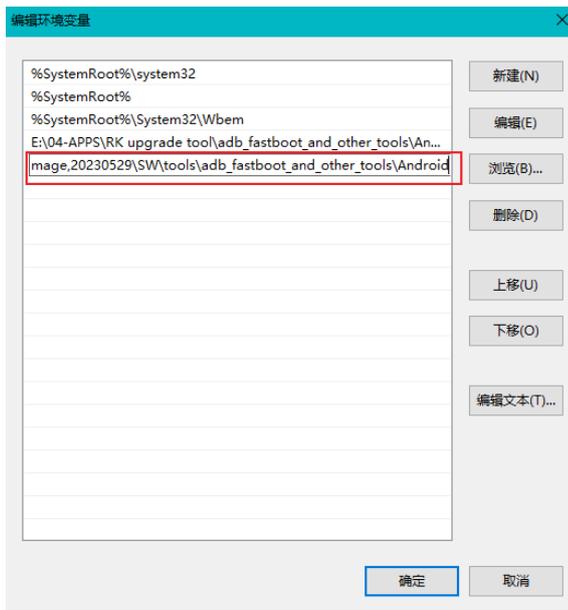
4. 按下“Win + R”键，并在对话框中输入 **sysdm.cpl**，打开设置界面；



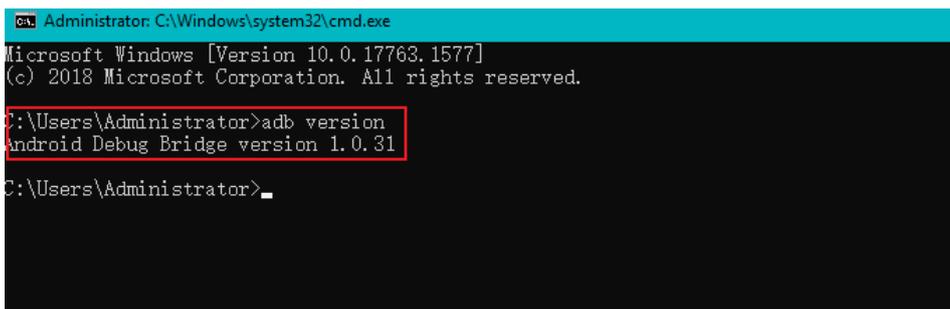
- 依次点击高级 > 环境变量 > 系统变量 > path > 编辑，然后在弹窗内选择新建：



- 将复制的 Android 文件夹路径粘贴至环境变量对话框中，然后逐一点击确定关闭对话框；



- 按下“Win + R”键，并在对话框中输入 cmd；
- 在命令提示框内输入 adb version，查看 ADB 工具是否已安装。



4.3 通过 ADB 命令安装应用程序

针对运行 Android 操作系统的设备，除了标准预装的 Android 应用程序外，用户还可以在设备上安装自己的应用。只要 Windows 主机上安装了 ADB 工具套件，并且用户具有访问权限，即可以在 Windows 环境下安装上述应用程序。

1. 使用 USB Type-A 转 Type-C 数据线连接主板和控制电脑；
2. 按下“Win + R”键，并在对话框中输入 `cmd`；
3. 在命令提示框内输入 `adb devices -l`，查看主板是否已连接控制电脑；

```
C:\Users\Administrator>adb devices -l
List of devices attached
874cc0aaede7d049    device product:occam model:Nexus_4 device:mako transport_id:1
```

4. 如果命令下方显示设备信息，则说明主板已经通过 ADB 与控制电脑连接，用户可以复制设备序列号（上图标识的区域），便于下一步使用；
5. 在命令提示框内输入以下命令安装 apk 应用程序：

```
adb -s <序列号> install <APK 文件路径>
```

6. 输入命令行后即执行安装，安装的结果将显示于命令下方；

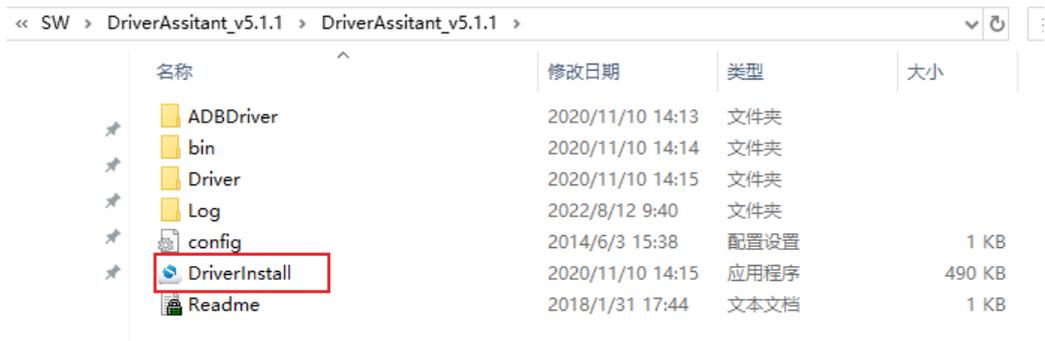
```
C:\Users\Administrator>adb -s 674cc0aaede7d049 install "C:\Users\Administrator\Desktop\Libraries for developers_v3.83_apkpure.com.apk"
Performing Streamed Install
Success
```

7. 新安装的应用程序将按照字母顺序在应用抽屉上显示。

- ▶ 以上截图仅作参考，并不代表您当前所持设备的编号及其他设备信息。
- ▶ 上述第 5 步中，用户也可以直接将本地的 .apk 文件拖拽至命令行，替换手动输入的应用程序路径。
- ▶ 如果安装失败，请尝试添加双引号将 .apk 文件的路径转换为绝对路径。

4.4 Windows 环境下升级固件

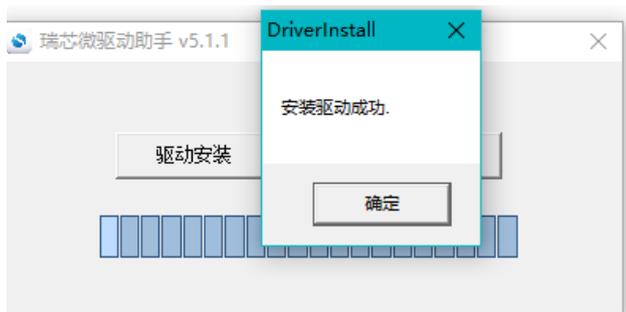
1. 请根据 4.2 中的步骤完成 Windows 主机的 ADB 设置；
2. 将软件发布包解压缩，打开升级驱动目录（\SW\ DriverAssitant_vxxx），找到 **DriverInstall.exe**；



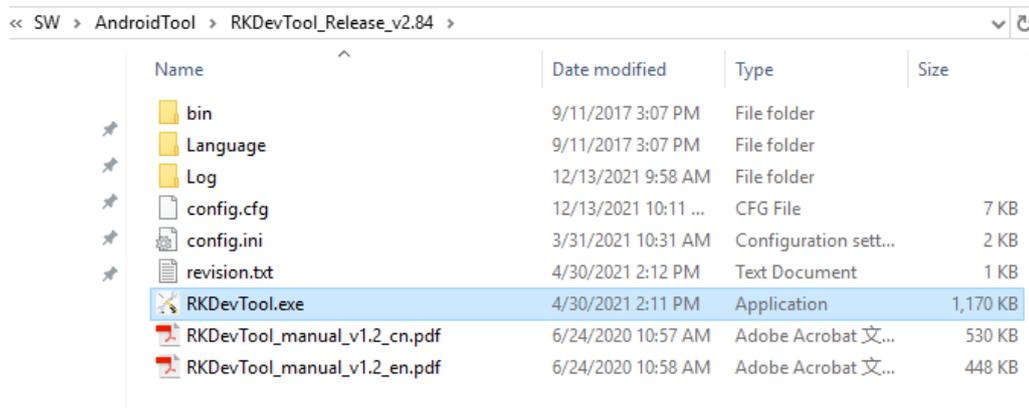
3. 点击鼠标右键，以管理员的身份运行驱动程序；
4. 首先点击**驱动卸载**，卸载之前安装的驱动版本（如有），然后点击**驱动安装**并等待；



5. 数秒后，将出现安装成功的弹窗；



6. 回到 SW 文件夹，打开升级工具所在目录（\SW\AndroidTool\RKDevTool_Release_vxxx\RKDevTool.exe）；



7. 双击升级驱动程序 **RKDevTool.exe**;
8. 打开升级窗口；



9. 使用 USB Type-A 转 Type-C 数据线连接 VT-SBC-3588 和 Windows 主机；
10. 同时按下 Windows 和 R 键并在对话框中输入 `cmd`，打开命令提示框；
11. 在命令提示框中输入 `adb devices` 查看主板是否与 Windows 主机相连；
12. Windows 主机识别到主板后，在命令提示框内输入 `adb reboot loader` 命令，重启设备并进入 Loader 模式；

13. 之后，升级窗口将提示发现一个 LOADER 设备，表示升级准备就绪。



14. 在 RK 开发工具窗口，依次点击**升级固件** > **固件**；

15. 打开升级文件 **update.img** 的路径（\SW\Image），固件的详细信息将自动填充到固件信息框内；

16. 点击**升级**按钮，系统将自动开始下载镜像并升级；



17. 升级完成后，系统将自动重启。

第 5 章 Debian 系统手册

5.1 系统介绍

Debian 是 Linux 系统的一个发行版本。VT-SBC-3588 基于 Debian GNU/Linux 11 操作系统创建。

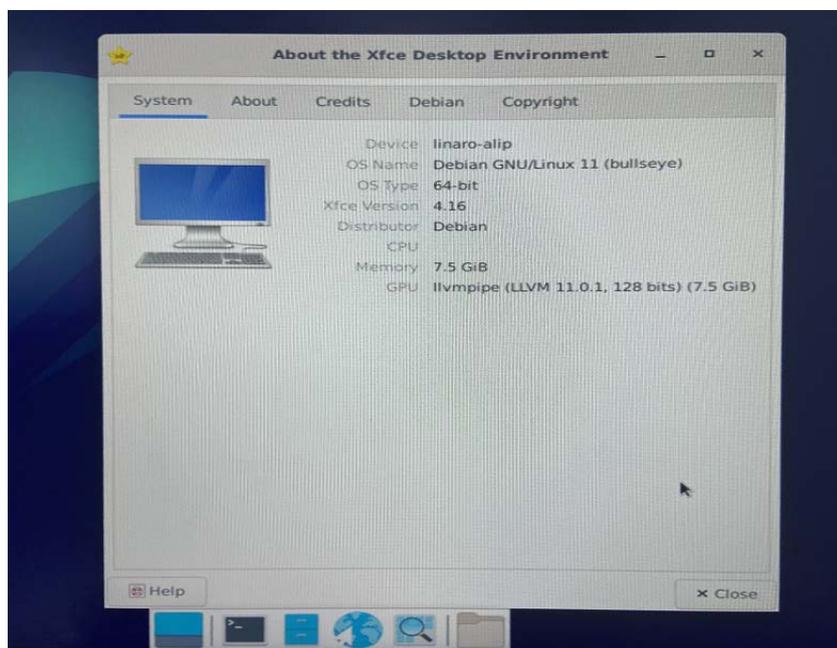
5.1.1 用户名与密码

系统开机后自动默认登陆 **linaro** 账号。

- 用户：linaro
- 密码：linaro

5.1.2 系统信息

1. 为方便操作，请使用键盘、鼠标和显示器连接设备；
2. 设备通电，系统将默认登录 linaro 用户；
3. 鼠标右键点击屏幕空白处，打开设备属性对话框；
4. 依次点击应用程序 > 关于 Xface，进入桌面环境，然后可以查看系统信息，包括设备名称、操作系统、版权声明等。



5.2 系统设置

点击屏幕右上角的 **linaro**，用户可以选择：

- 锁屏；
- 切换用户；
- 关机/挂起设备；
- （点击“注销”后）注销/重启/关机/挂起设备或者切换用户，同时选择保存当前会话，用于将来登录。

用户可以通过屏幕上方的菜单栏：

- 修改设备亮度和电源设置；
- 调节系统音量和声音设置；
- 将设备添加至 **Wi-Fi** 网络并与蓝牙设备配对；
- 切换不同的工作区域。

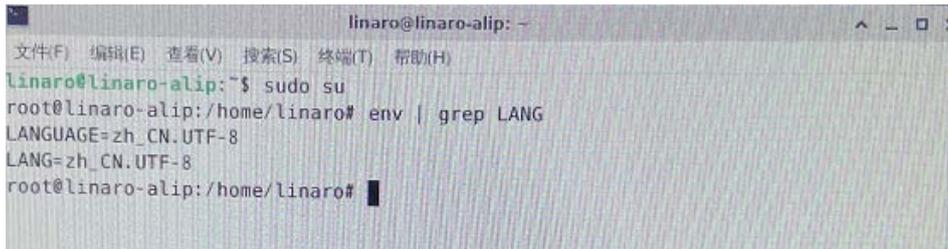
用户可以点击屏幕左上角的**所有应用程序**或者右键点击屏幕空白处，在弹出的设备属性对话框中点击**应用程序**，然后进行系统其他设置。

5.2.1 语言设置

假设当前系统语言为中文，下文的例子将说明如何将系统语言更改为英文（English US）。

1. 从桌面底部打开终端并输入 `sudo su`，切换至 `root` 权限；
2. 运行下列命令查看系统当前使用的语言（系统默认使用 `zh_CN.UTF-8` 区域设置）；

```
# env | grep LANG
```



```
linaro@linaro-alip: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
linaro@linaro-alip:~$ sudo su  
root@linaro-alip:/home/linaro# env | grep LANG  
LANGUAGE=zh_CN.UTF-8  
LANG=zh_CN.UTF-8  
root@linaro-alip:/home/linaro#
```

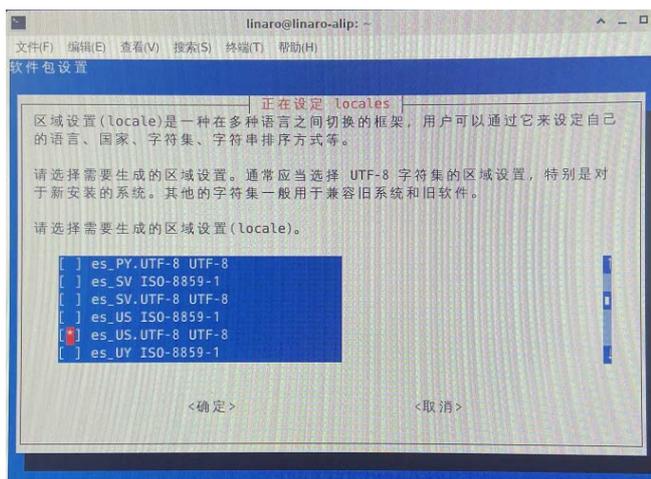
3. 输入下列命令调用区域设置的配置页面（`en_US.UTF-8` 为用户希望切换的目标语言所在区域）；

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

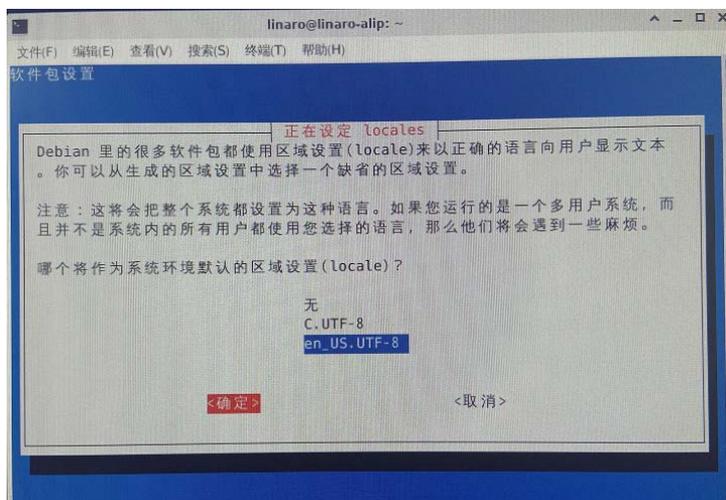
4. 输入下列命令重新设定区域设置；

```
# dpkg-reconfigure locales
```

5. 使用键盘的上、下箭头移动至目标区域设置（`en_US.UTF-8`）；
6. 按下空格键，选择目标区域设置；



7. 移动至当前区域设置(zh_CN.UTF-8)，按空格键取消选择；
8. 然后按 Tab 键将鼠标移动至“确定”选项，再按回车键确认选择；
9. 使用上、下箭头在列表中选择需要设定为默认值的区域设置(en_US.UTF-8)，然后按 Tab 键将鼠标移动至“确定”选项，再按回车键确认选择；



10. 等待几秒钟，待所选的区域设置生效；

```
linaro@linaro-alip: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
linaro@linaro-alip:~$ sudo su  
root@linaro-alip:/home/linaro# env | grep LANG  
LANGUAGE=zh_CN.UTF-8  
LANG=zh_CN.UTF-8  
root@linaro-alip:/home/linaro# export LANG=en_US.UTF-8  
root@linaro-alip:/home/linaro# dpkg-reconfigure locales  
Generating locales (this might take a while)...  
  en_US.UTF-8... done  
Generation complete.  
root@linaro-alip:/home/linaro#
```

11. 输入 `reboot` 重启设备；
12. 设备重启后选择“更新名称”，将文件更新至当前语言；
13. 输入 `locale-a` 查看所选区域设置是否为系统默认设定。

5.2.2 时间和日期设置

用户可以点击屏幕左上角的**所有应用程序**或者右键点击屏幕空白处，在弹出的设备属性对话框中点击**应用程序**，调出系统菜单：

1. 点击**系统 > 时间和日期**，打开时间和日期设置；
2. 点击**解锁**并输入密码（linaro）进行设置；
3. 点击时区，并从列表中选择用户所在地时区；
4. 选择与互联网服务器保持同步或者手动输入时间；



5. 点击**锁定**，等待设置生效。

5.2.3 网络配置

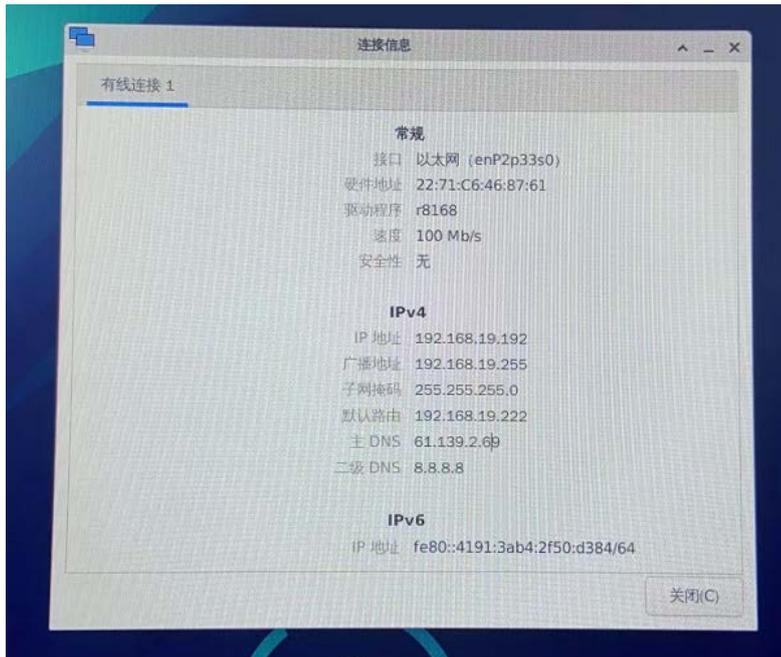
VT-SBC-3588 配置为连接外部以太网或 Wi-Fi 无线网络后即可上网。

连接无线网络前，请确保设备已安装 Wi-Fi 天线。

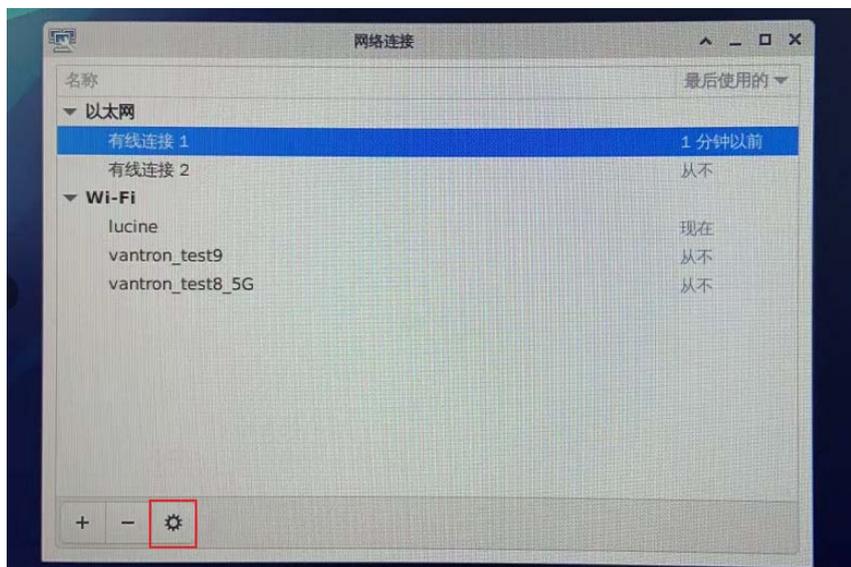
1. 点击菜单栏上的网络连接图标（双电脑/Wi-Fi 图标），从列表中搜索目标网络；



2. 如需查看当前的网络连接，用户可以将鼠标移动至网络连接图标处，然后点击鼠标右键，选择**连接信息**：



3. 如需编辑网络配置，用户可以在网络连接图标处点击鼠标右键，选择**编辑连接**，然后选择一个连接点，再点击齿轮图标编辑网络设置。



5.3 Ubuntu 环境升级镜像

5.3.1 环境准备

- VT-SBC-3588
- 运行 Ubuntu 的主机（推荐 Ubuntu 18.04 或以上系统）
- USB Type-A 转 Type-C 数据线
- VT-SBC-3588 软件发布包

5.3.2 烧写镜像

1. 使用 USB Type-A 转 Type-C 数据线连接设备和主机；
2. 打开终端，输入以下命令，安装 ADB 工具；

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

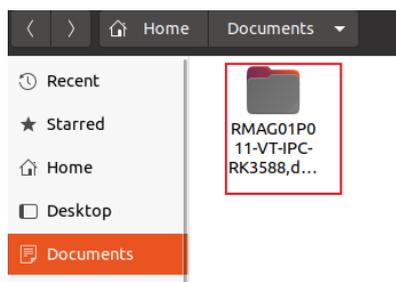
3. 检查设备是否通过 ADB 连接至 Ubuntu 主机；

```
$ adb devices -l
```

4. 在终端中输入 `adb shell`，便于在设备上执行 shell 命令；
5. 输入 `reboot loader`，设备将自动重启并进入 bootloader 模式；

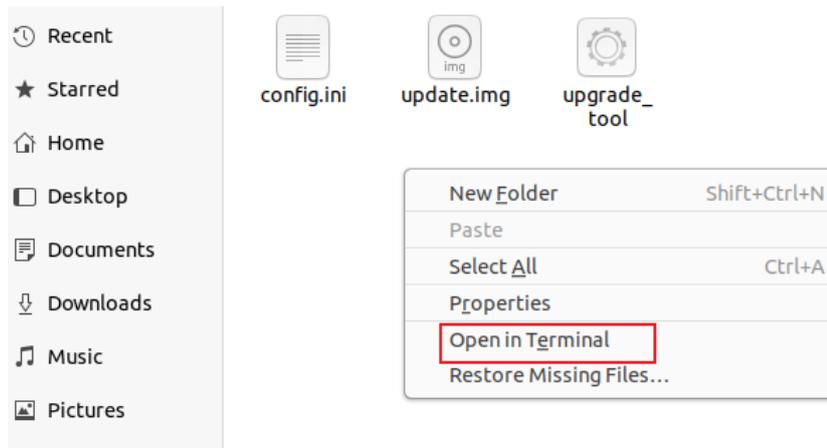
```
e@ubuntu:~$ adb shell
root@linaro-alip:/# reboot loader
root@linaro-alip:/# @ubuntu:~$
```

6. 将万创提供的升级镜像包拷贝至 Ubuntu 主机（例如，拷贝至文档文件夹）；



7. 将镜像包解压缩并打开 `img` 文件夹；

8. 在空白处点击鼠标右键并选择在**终端中打开**，然后在新的终端内执行后续命令；



9. 输入以下命令下载升级镜像并开始升级；

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

10. 输入 sudo 密码后开始下载系统镜像；

```
root@ubuntu:~/Documents/RMAG01P011-VT-IPC-RK3588,debian,V1.0.0,2022-08-01/img
$
sudo ./upgrade_tool uf update.img
[sudo] password for root:
Using /home/root/Documents/RMAG01P011-VT-IPC-RK3588,debian,V1.0.0,2022-08-01/
img/config.ini
Program Log will save in the /root/upgrade_tool/log/
Loading firmware...
Support Type:RK3588      FW Ver:1.0.00   FW Time:2022-07-30 16:22:22
Loader ver:1.0b Loader Time:2022-07-30 16:02:36
Download Image Total(5829691K),Current(2798535K)
```

11. 下载完成后，系统将立即开始升级，升级完成后，设备将自动重启。

```
Program Log will save in the /root/upgrade_tool/log/
Loading firmware...
Support Type:RK3588      FW Ver:1.0.00   FW Time:2022-07-30 16:22:22
Loader ver:1.0b Loader Time:2022-07-30 16:02:36
Upgrade firmware ok.
```

第 6 章 Ubuntu 系统手册

6.1 系统介绍

Ubuntu 是 Linux 系统的一个发行版本。VT-SBC-3588 当前运行的 Ubuntu 版本为 Ubuntu 20.04. LTS。

6.1.1 用户名与密码

开机后，用户需登陆账号进入系统。

- 用户名：linaro
- 密码：linaro

6.1.2 系统信息

1. 为方便操作，请使用键盘、鼠标和显示器连接设备；
2. 设备通电，输入密码后登录默认用户（linaro）；
密码：linaro
3. 点击屏幕左下角**显示应用程序**图标 ，进入应用程序；
4. 依次点击**设置 > 关于**，可以查看设备系统的相关信息。

6.2 系统设置

点击**进入应用程序**，用户可以点击或搜索单个应用，使用相关功能。也可以进入**设置**页面对系统基本设置进行修改。

6.2.1 语言

用户可以通过**设置 > 区域和语言**，设置系统的显示语言和格式，也可以配合**语言支持**应用添加或删除输入语言。所有更改将在系统重启后生效。

如果使用命令更改系统语言，操作步骤与 5.2.1 中的步骤一致。

6.2.2 日期和时间

用户可以点击**设置 > 日期和时间**，打开设置日期和时间页面：

1. 点击**解锁**并输入密码进行设置；
2. 打开**自动设置日期和时间**，保持设备时间与网络时间一致，或者，关闭该选项，手动输入日期和时间；
3. 选择/搜索目标时区并设置时间格式；
4. 点击**锁定**，保存设置；
5. 退出并等待设置生效。

6.2.3 网络/Wi-Fi

VT-SBC-3588 配置为连接外部以太网或 Wi-Fi 无线网络后即可上网。用户也可以在**设置 > 网络**页面，添加 VPN，使用虚拟网络。

连接 Wi-Fi 前，请确保设备已安装 Wi-Fi 天线。

用户可以输入命令，查看网络接口信息：`ip link show`。

6.2.4 用户

输入密码解锁当前用户后，用户可以对该账号进行编辑，也可以添加或删除新的用户。

6.3 Ubuntu 环境升级镜像

6.3.1 环境准备

- VT-SBC-3588
- 运行 Ubuntu 的主机（推荐 Ubuntu 18.04 或以上系统）
- USB Type-A 转 Type-C 数据线
- VT-SBC-3588 软件发布包

6.3.2 烧写镜像

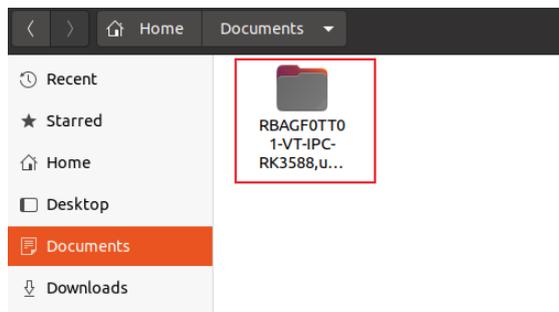
1. 使用 USBType-A 转 Type-C 数据线连接主板和 Ubuntu 主机；
2. 打开终端，输入以下命令，安装 ADB 工具；

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

3. 检查设备是否通过 ADB 连接至 Ubuntu 主机；

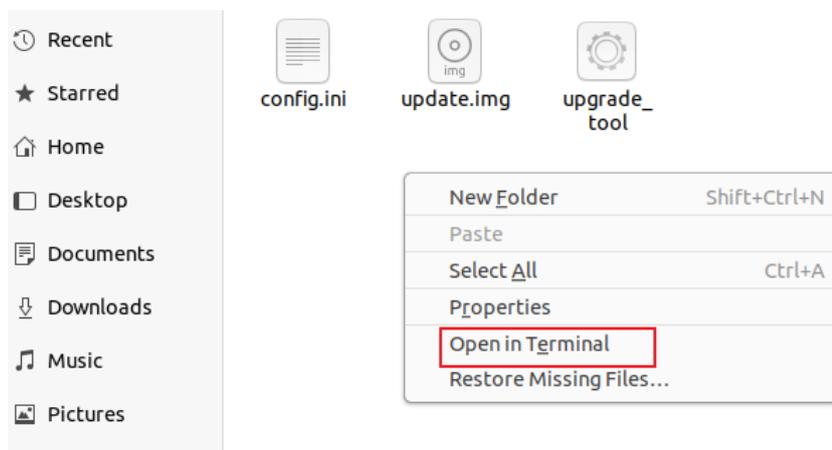
```
$ adb devices -l
```

4. 在终端中输入 `adb shell`，便于在设备上执行 shell 命令；
5. 输入 `reboot loader`，主板将自动重启并进入 bootloader 模式；
6. 将万创提供的升级镜像包拷贝至 Ubuntu 主机（例如，拷贝至文档文件夹）；



7. 将镜像包解压缩并打开 `img` 文件夹；

8. 在空白处点击鼠标右键并选择在**终端中打开**，然后在新的终端内执行后续命令；



9. 输入以下命令下载升级镜像并开始升级；

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

10. 输入 `sudo` 密码后开始下载系统镜像；
11. 下载完成后，系统将立即开始升级，升级完成后，设备将自动重启。

6.4 ROS 教程

请参考 <http://wiki.ros.org/ROS/Tutorials>，了解 ROS 系统详细信息。

第 7 章 废弃处理与质保

7.1 废弃处理

当设备到了使用期限，为了环境 and 安全，建议您适当地处理设备。

处理设备前，请备份您的数据并将其从设备中删除。

建议在处理前拆解设备，以符合当地法规。请确保废弃的电池已按照当地关于废物处理的规定进行处理。电池具有爆炸性，请勿将其扔进火中或放入普通垃圾桶中。标有“爆炸性”标志的产品或产品包装不应该按照家庭垃圾处理，应当送到专门的电气和电子垃圾回收/处理中心。

妥善处理这类废物有助于避免对周围环境和人们的健康造成伤害和不利影响。请联系当地机构或回收/处理中心，了解更多相关产品的回收/处理方法。

7.2 质保

产品质保

万创向客户保证，万创或万创分包商制造的产品从万创发运时将严格符合双方商定的规格，不存在工艺和材料上的缺陷（由客户提供的除外）。万创的质保义务限于产品的更换或维修（由其自行决定）。如果出现质量问题，产品发货后，客户应当自开具发票之日起 **24 个月**内，自付运费将产品返回万创工厂。经检查后，万创合理确认产品具有缺陷的，由万创承担质保责任。之后，由万创承担将产品发运给客户的运输费用。

保修期外的维修

万创将按照当时的服务费率为已过保修期的产品提供维修服务。只要市场有售，万创将根据客户要求向客户提供非保修期内的维修部件，但客户需提前下达采购订单。维修部件有 3 个月的延长保修期。

产品退回

任何根据上述条款被认定为有缺陷并在保修期内的产品，只有在客户收到并参照万创提供的退货授权（RMA）号码后，才能退回给万创。万创应在客户提出要求后的 3（三）个工作日内提供 RMA。万创应在向客户发出退货产品后，向客户提供新的发票。在客户因拒收或保修期内的缺陷而退回任何产品之前，应向万创提供在客户所在地检查该产品的机会。除非拒收或缺陷的原因被确定为万创的责任，否则经检查的产品不得退回万创。万创应在收到产品后的 14（十四）个工作日内，向客户发出缺陷产品的替换。如果万创由于其无法控制的原因而不能提供上述服务，万创应记录这种情况并立即通知客户。