

深圳金亚太科技有限公司

Shenzhen Geniatech Co.,Ltd.**产品规格书****型号:APC3588AI**

APPROVED BY GENIATECH		
PREPARED BY 编写	CHECKED BY 审核	APPROVED BY 批准

Please return the original copy after approved by your company with seal and signature.
请在贵公司盖章并签字后寄回正本一份。

APPROVED BY CUSTOMER		
COMMENTS 确认意见	APPROVED BY 批准签字	COMPANY SEAL 盖章

Website: www.geniatech.com

Address: Room 1002-1004, Block A, Building 8, Shenzhen International Innovation Valley, Dashi Road, Nanshan District, Shenzhen, Guangdong, China

CONTENT

1.入门指南.....1~7

2.硬件配置..... 7~11

3.安装.....11~12

4.软件使用指南.....12~43

更新记录

版本	日期	板卡号	页数	描述	撰写者
V1.0	2025/09/20				
V1.1	2026/01/17				

1. 入门指南

1.1 使用注意事项

1. 储存温度: 工规 (-45 °C ~ 90°C).
2. 请勿挤压、扭曲或拆卸设备.
3. 让电路板远离静电.
4. 让设备远离水和其他液体.
5. 不要使用过长的连接线, 以免影响性能和图像质量.

1.2 产品配件

序号	名称	数量
1	FSP 60W 适配器	1 个
2	AC 电源线	1 根
3	WiFi 棒状增益天线	1 根
4	GPS 棒状增益天线	1 根
5	3.81mm 2x3PIN 凤凰端子	5 个
6	金属支架	1 套

1.3 系统规格

系统信息	
CPU	Rockchip RK3588J, Quad-core Cortex-A76 and Quad-core Cortex-A55
GPU	Mali-G610 MP4 (4x256KB L2 Cache)
RAM	16GB LPDDR4
ROM	128GB EMMC
WatchDog Timer	32-bit WatchDog Timer
扩展功能	
M.2	1x M.2 KEY M, 1x M.2 KEY B
MINI PCIE	1x MINI PCIE
前面板 I/O	
SIM	1 x SIM 卡槽
Micro SD	1 x TF 卡槽
LED	4 x LED (红绿双色)
HDMI	1 x HDMI IN, 2 x HDMI OUT
USB	2 x USB2.0, 2 x USB3.0

Room 02-04, 10/F, Block A, Building 8, Shenzhen International Innovation Valley, Dashi Road,

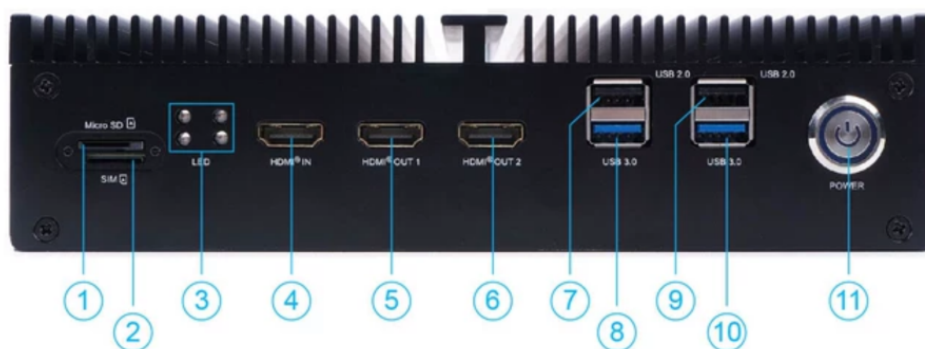
电源开关	1 x 金属开关			
后面板 I/O				
WiFi 天线	1 x SMA 端子			
4G 天线	2 x SMA 端子			
GPS 天线	1 x SMA 端子			
COM	2 x DB9 接口（RS232）			
LAN	3 x RJ45 接口			
电源接口	1 x 4PIN 3.81 端子			
DEBUG/RESET/Recovery	1 x 2 x 3PIN 3.81 端子			
GPIO	1 x 2 x 3PIN，3.81 端子			
RS232	1 x 2x3PIN，3.81 端子（RS232 x2）			
RS485	1x2x3PIN，3.81 端子（RS485 x2）			
CAN	1x2x3PIN，3.81 端子（CAN x2）			
显示				
图形芯片	ARM Mali-G610 MP4 四核 GPU			
音频				
数字音频编解码器	支持 2 路（DAC）,支持 I2S/PCM 接口，支持主从模式，支持 16 位采样分辨率，每个数字 DAC 通道支持三种混音模式			
以太网				
数据速率	3x1000M Ethernet			
LED 指示灯	状态		速率	
	LED	状态	LED	状态
	灯灭	未连接	灯灭	100M
	橙 色 常亮	已连接	绿 灯 常亮	1000M
	橙 色 灯闪烁	数 据 传 输中		
电源要求				
电压输入规格	DC12V			
电压输入连接器	4Pin 接线端子			
机械&环境				
工作温度	工规（-30 °C ~ 75°C）			
存储温度	工规（-45 °C ~ 90°C）			
工作湿度	10%≤相对湿度≤90%			
尺寸	249 mm x 226 mm x65mm			
重量	2.05kg			
振动测试	随机振动 1.PSD:0.00005 G² /Hz,0.21 Grms 2.系统状态：工作模式 3.测试频率：5~500Hz500Hz 4.测试轴：X,Y,Z			

Room 02-04, 10/F, Block A, Building 8, Shenzhen International Innovation Valley, Dashi Road,

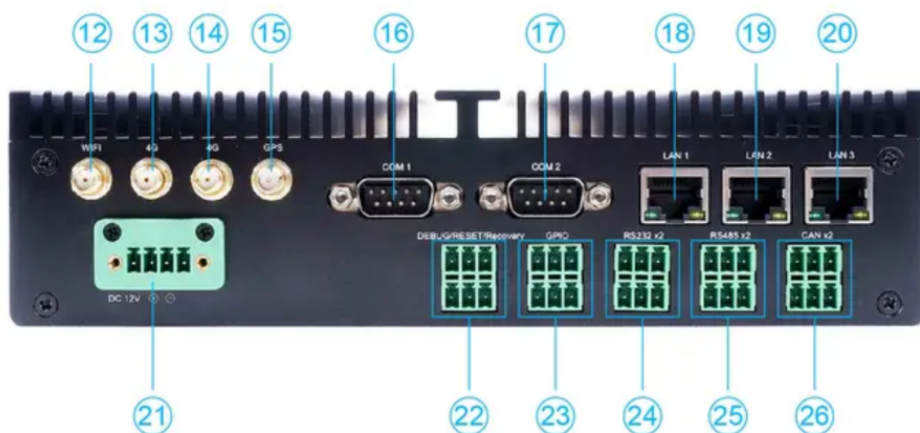
	5.测试时长：15 分钟/轴向 正弦扫频测试 1.测试加速度：1G ² /Hz 2.测试频率：5~200Hz 3.扫频速率：0.5 Oct/min 4.测试轴：X,Y,Z 5.测试时长：10 分钟/轴向 6.系统状态：关机
冲击测试	1.波形：半正弦波 2.加速度：2G 3.脉冲时间：11ms 4.冲击方向：±X,±Y,±Z 5.系统状态：工作模式
跌落测试	1.测试部位：4 条棱、6 个面 2.跌落高度：100mm 3.样品重量：2.05kg
IP 防护等级	IP40
支架	挂耳金属支架
软件支持	
操作系统信息	Debian11 & Ubuntu 20.04
认证	
法定强制性产品认证 / 市场准入类认证	CCC

1.4 系统概述

1.4.1 正面视图



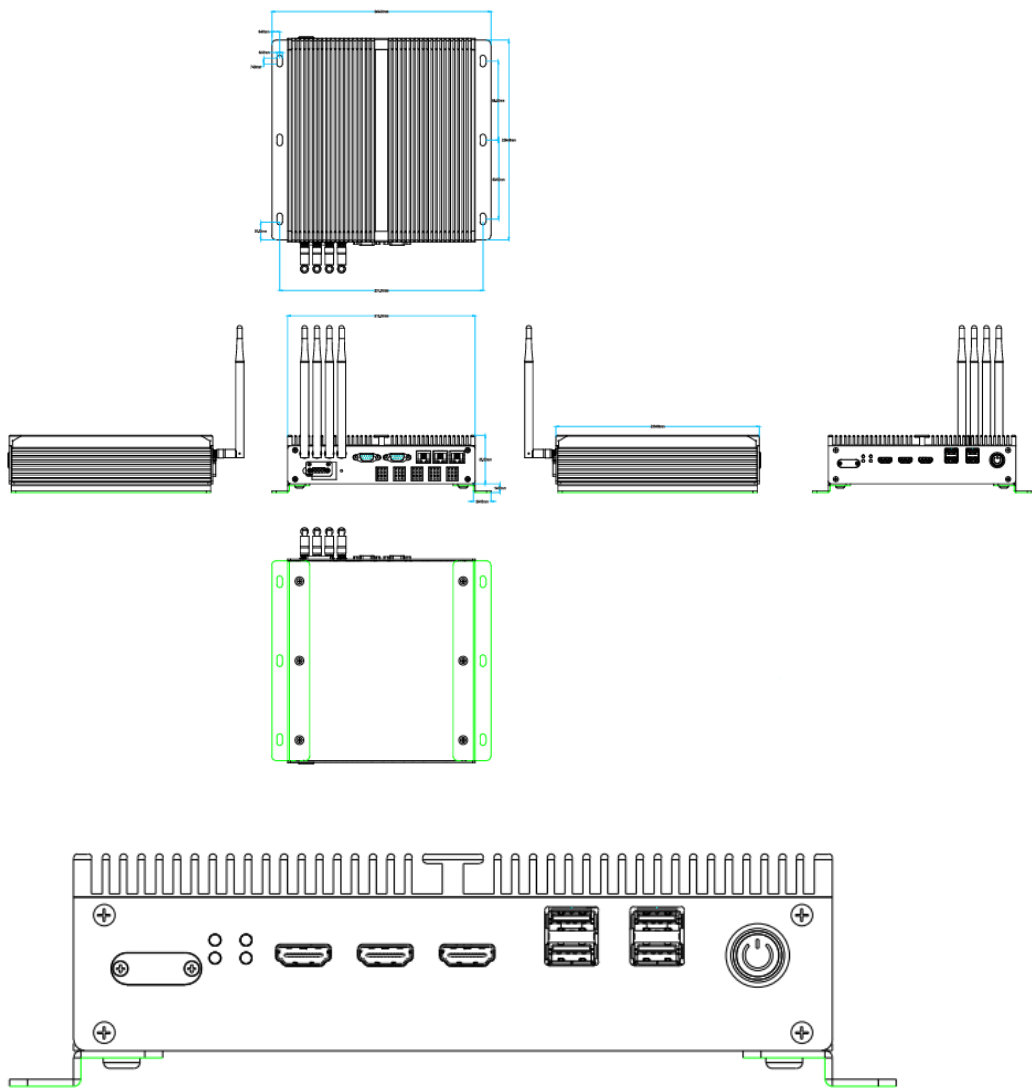
1.4.2 背面视图

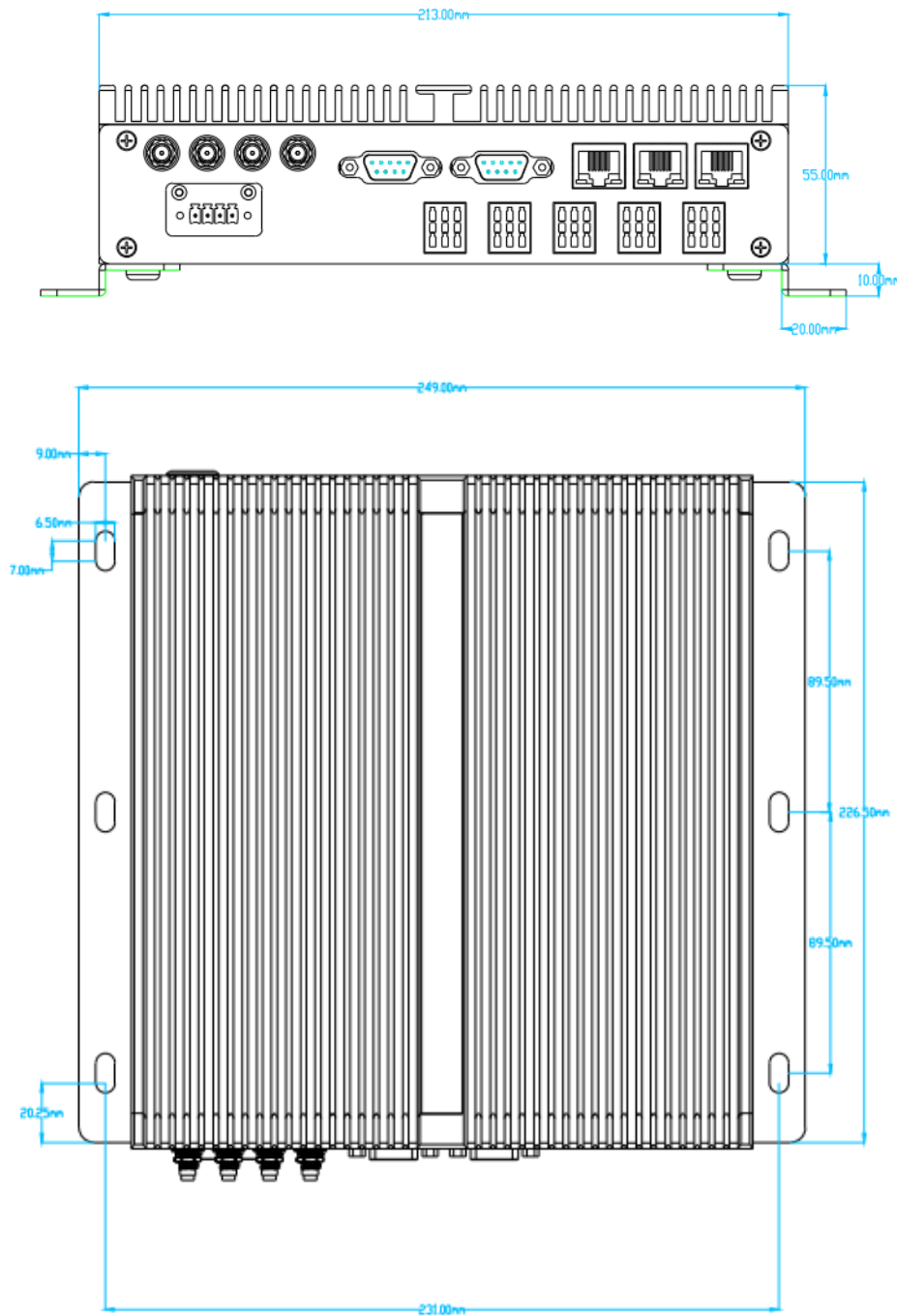


接口定义							
名称	序号	描述	PIN 定义				
电源开关	11	1x 金属开关					
USB 3.0（下）	10	1xUSB 2.0 Type-A 接口					
USB 2.0（上）	9	1xUSB 3.0 Type-A 接口					
USB 3.0（下）	8	1xUSB 2.0 Type-A 接口					
USB 2.0（上）	7	1xUSB 3.0 Type-A 接口					
HDMI OUT 2	6	1xHDMI 接口					
HDMI OUT 1	5	1xHDMI 接口					
HDMI IN	4	1xHDMI 接口					
LED	3	4xLED（红绿双色）					
SIM	2	1xSIM 卡槽					
Micro SD	1	1xTF 卡槽					
WiFi 天线	12	1xSMA 端子					
4G 天线	13	1xSMA 端子					
4G 天线	14	1xSMA 端子					
GPS 天线	15	1xSMA 端子					
COM2	16	1xDB9 接口（RS232）					
COM1	17	1xDB9 接口（RS232）					
LAN3	18	1xRJ45 接口					
LAN2	19	1xRJ45 接口					
LAN1	20	1xRJ45 接口					
电源接口	21	1x4PIN 3.81 端子		空	12V	GND	空
RESET（右上）	22	1x2x3PIN 3.81 端子		GND	RECOVER		RESET
				GND	DEBUG_TX		DEBUG_RX

GPIO	23	1x2x3PIN, 3.81 端子		GND	GPIO	GPIO
				GND	GPIO	GPIO
RS232 (上)	24	1x2x3PIN, 3.81 端子		GND	RS232_TX4	RS232_RX4
				GND	RS232_TX3	RS232_RX3
RS485 (上)	25	1x2x3PIN, 3.81 端子		GND	RS485B_2	RS485A_2
				GND	RS485B_1	RS485A_1
CAN(上)	26	1x2x3PIN, 3.81 端子		GND	CANH_2	CANL_2
CAN (下)				GND	CANH_1	CANL_1

1.5 产品结构图





1.6 操作指引

(a) 设备安装:

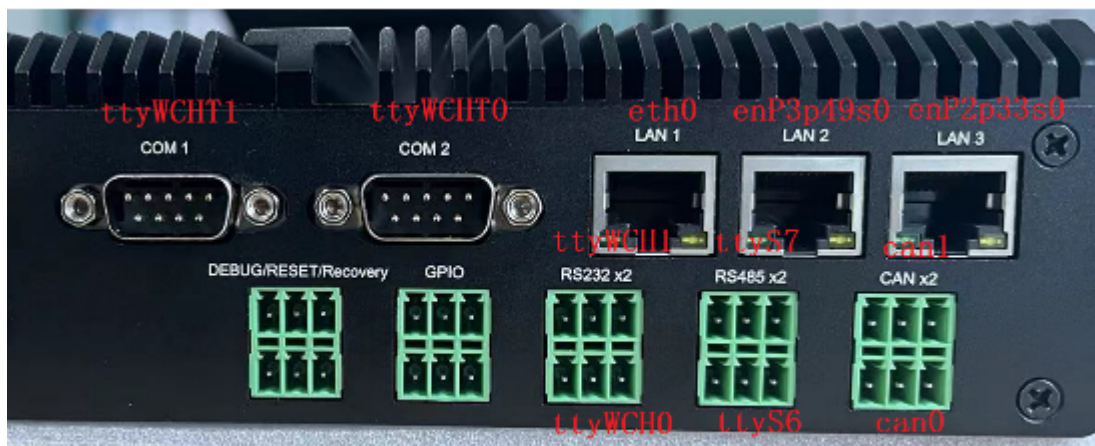
- 从包装中取出设备及配件，放置在合适位置。
- 核对装箱清单（配件）。
- 将电源线连接到设备。
- 将电源线插头插入电源插座。

- 按下设备上的“电源图标” 开机按键，启动设备。
- (b) 显示器安装：
 - 插入显示器线缆。
- (c) 键盘与鼠标安装：
 - 插入鼠标与键盘。
- (d) 系统开机操作：
 - 开启系统。

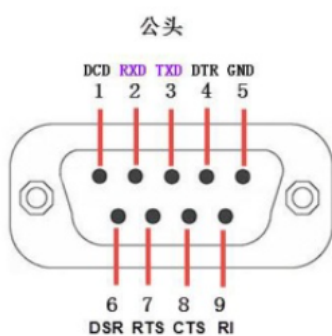
2. 硬件配置

2.1 连接器布局

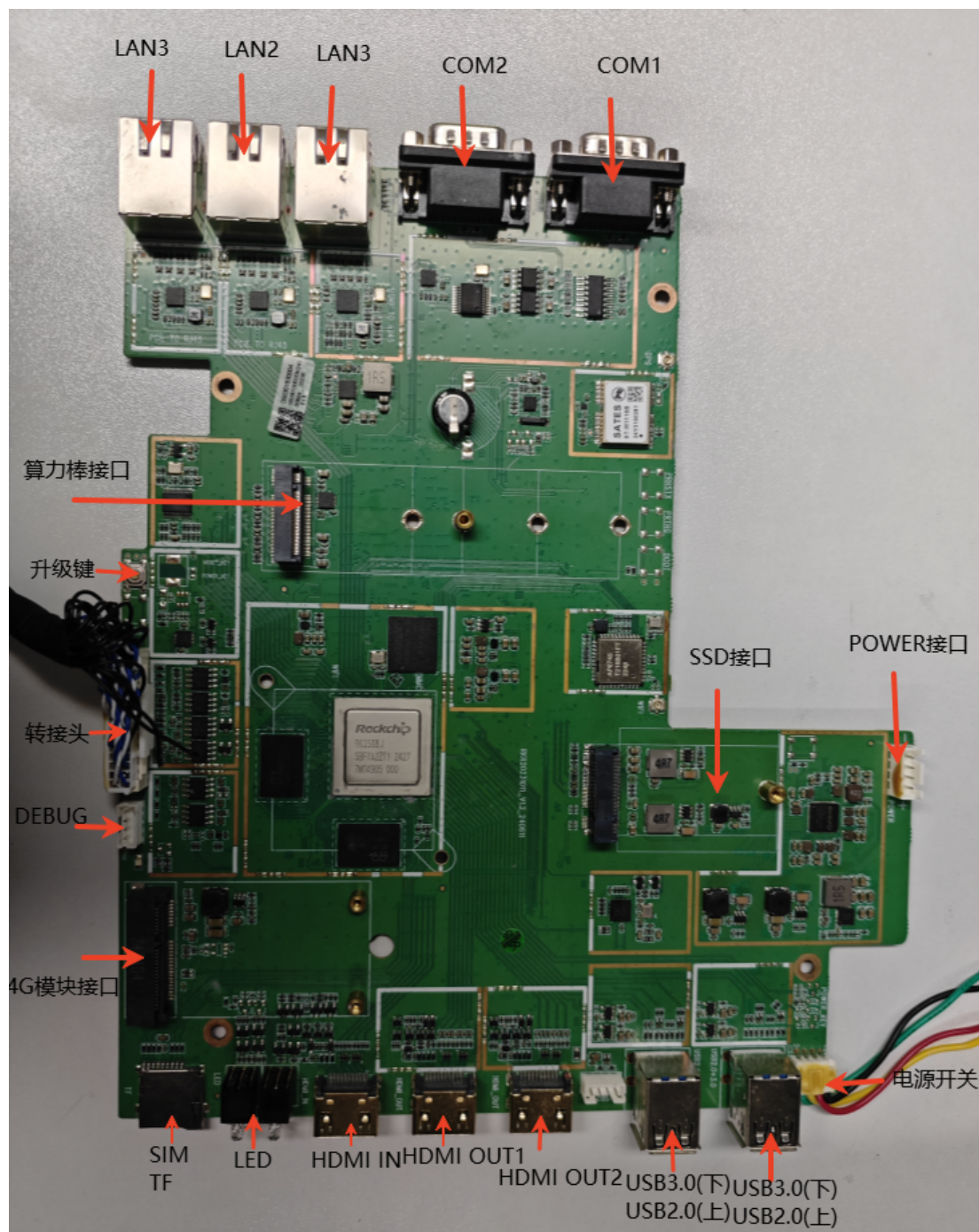
2.1.1 通用输入 / 输出连接器（GPIO）

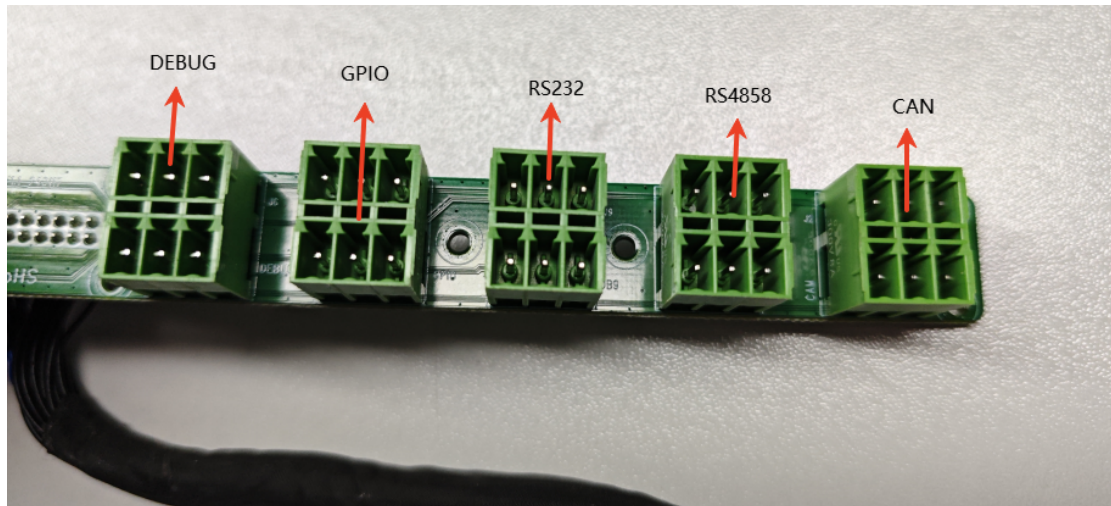


2.1.2 串行端口 1/2 连接器 (COM1/2)



2.2 接口概述





2.3 连机器清单

标识	功能	备注
电源开关	1x 金属开关	
USB 3.0 (下)	1xUSB 2.0 Type-A 接口	
USB 2.0 (上)	1xUSB 3.0 Type-A 接口	
USB 3.0 (下)	1xUSB 2.0 Type-A 接口	
USB 2.0 (上)	1xUSB 3.0 Type-A 接口	
HDMI OUT 2	1xHDMI 接口	
HDMI OUT 1	1xHDMI 接口	
HDMI IN	1xHDMI 接口	
LED	4xLED (红绿双色)	
SIM	1xSIM 卡槽	
Micro SD	1xTF 卡槽	
WiFi 天线	1xSMA 端子	
4G 天线	1xSMA 端子	
4G 天线	1xSMA 端子	
GPS 天线	1xSMA 端子	
COM2	1xDB9 接口 (RS232)	
COM1	1xDB9 接口 (RS232)	
LAN3	1xRJ45 接口	
LAN2	1xRJ45 接口	
LAN1	1xRJ45 接口	
电源接口	1x4PIN 3.81 端子	
DEBUG	1*2*3PIN 3.81 端子	
GPIO	1*2*3PIN, 3.81 端子	
RS232	1*2*3PIN, 3.81 端子	

Room 02-04, 10/F, Block A, Building 8, Shenzhen International Innovation Valley, Dashi Road,

10 / 44

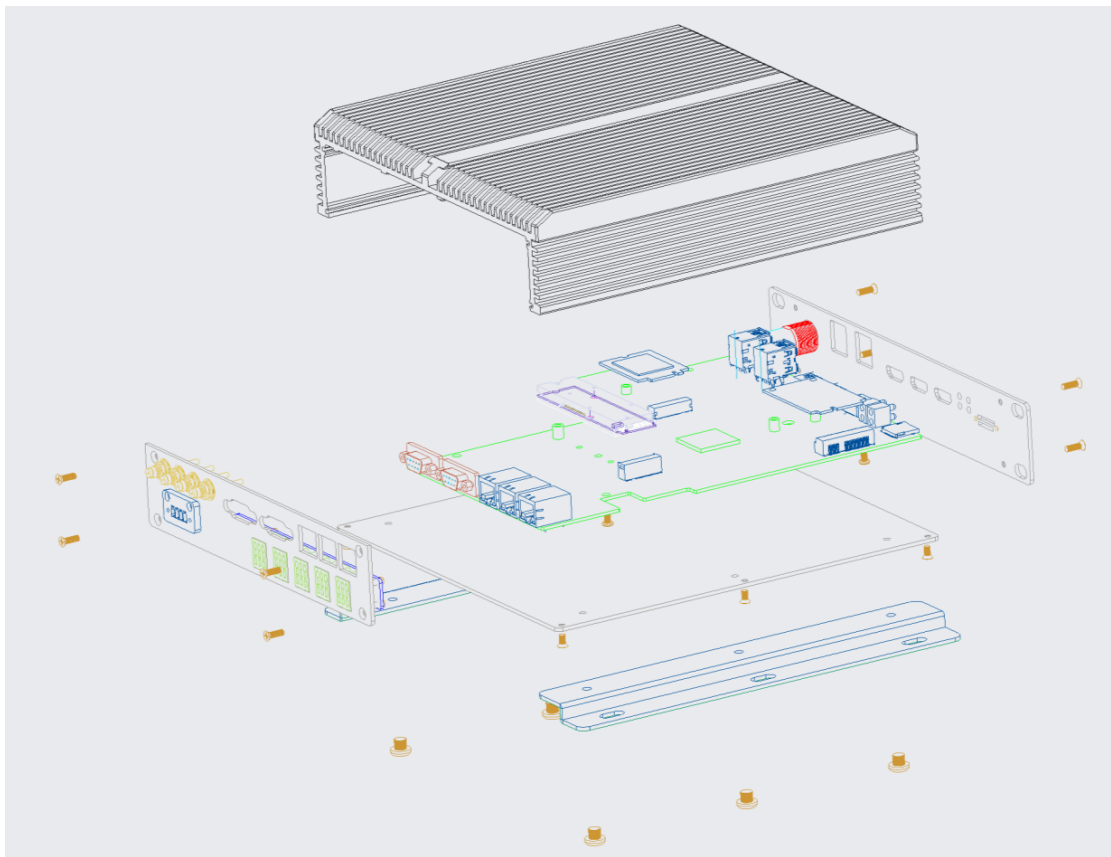
Nanshan District, Shenzhen, Guangdong, China

Email: support@geniatech.com Tel: (+ 86) 755 86028588

RS485	1*2*3PIN, 3.81 端子	
CAN	1*2*3PIN, 3.81 端子	
升级键	1x 自复位按键	
算力棒接口	M.2 接口	
转接头	1*15*2 连接器	
DEBUG	1*4 插针	
4G 模块接口	M.2 接口	
SSD 接口	M.2 接口	
POWER 接口	1*4 插针	

3. 安装

3.1 安装内存与 M.2 Key-B/ M.2 Key-M 卡



步骤 1: 取下产品底壳的 6 颗螺丝。

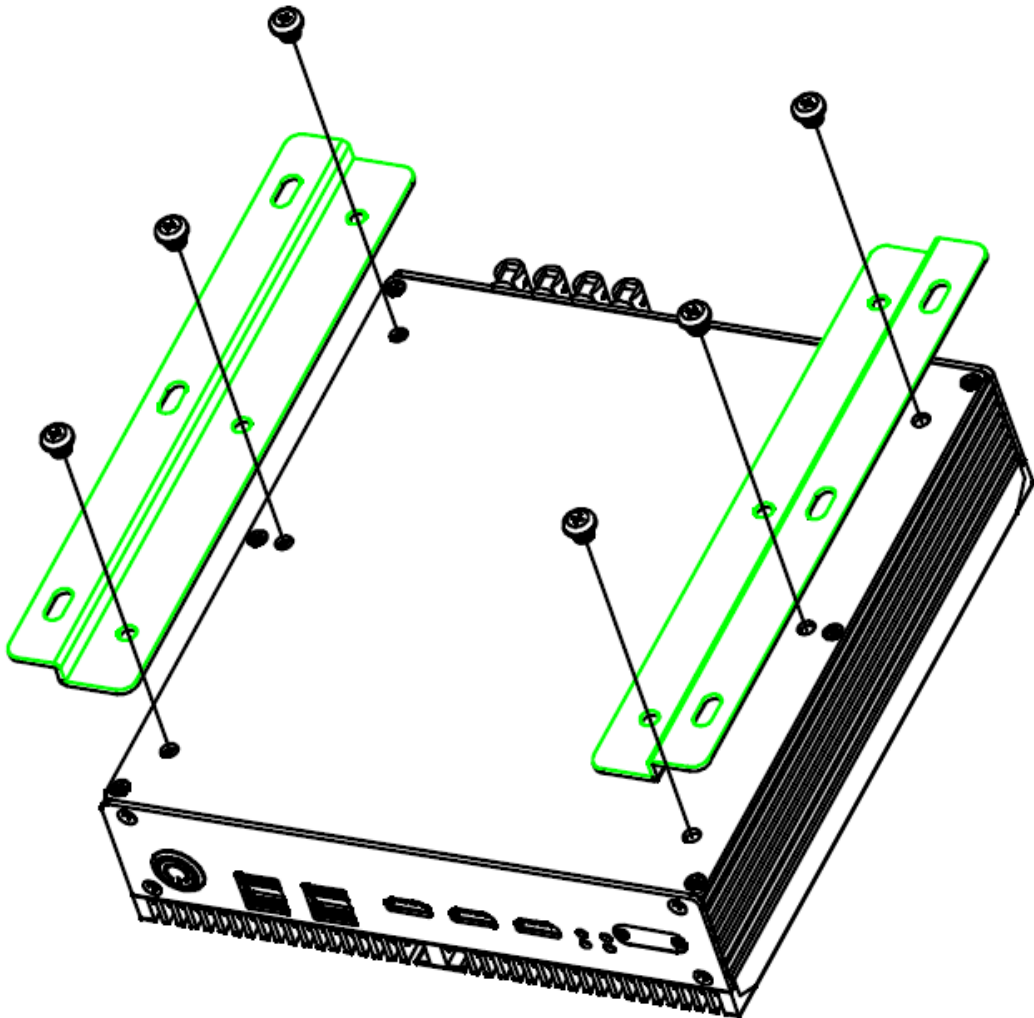
步骤 2: 取下产品前后挡板的 8 颗螺丝。

步骤 3: 取下产品主板的 4 颗螺丝，将主板取下。

步骤 4: 将内存插入内存插槽，向下按压直至内存完全卡紧、就位。

步骤 5: 将 M.2 Key-B 卡和 M.2 Key-M 卡插入指定位置; 将 M.2 设备用螺丝固定, 完成安装。

3.2 安装导轨支架



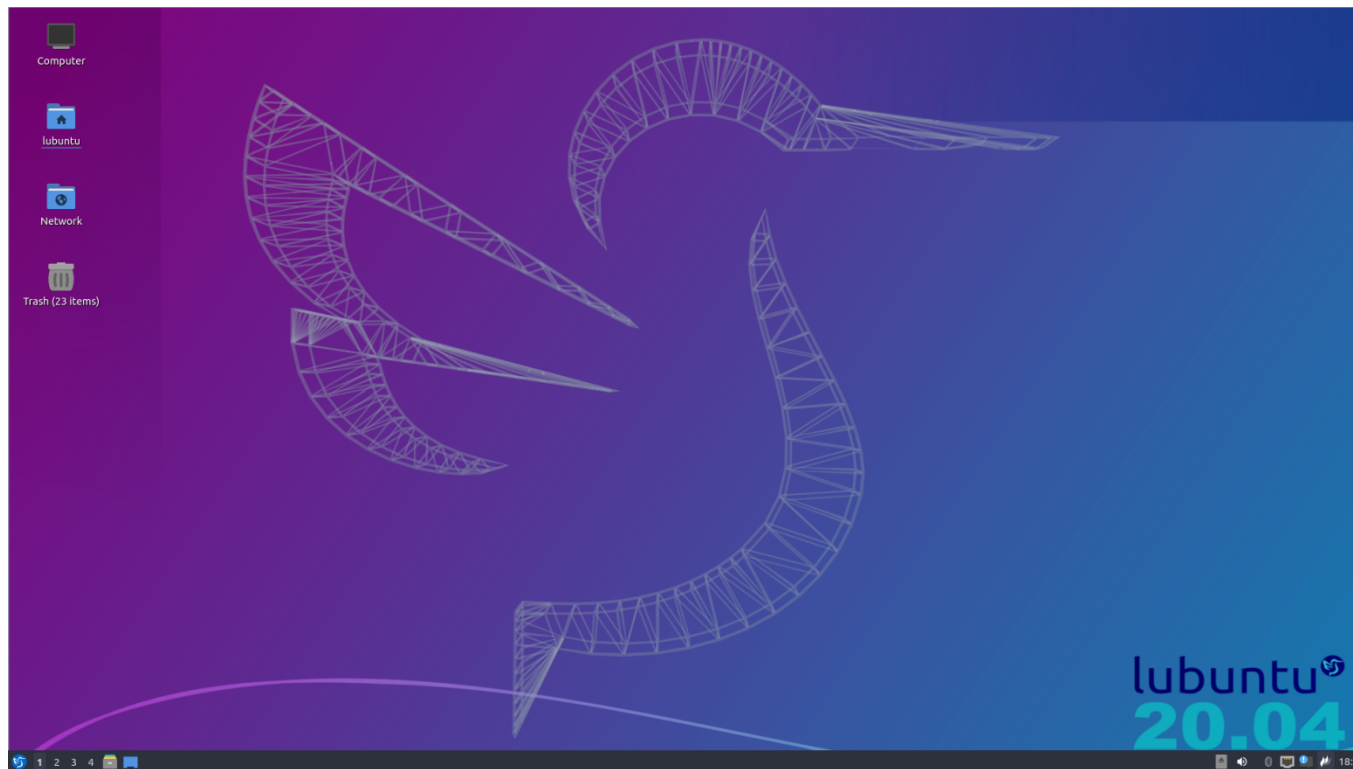
步骤 1: 将支架定位在设备两侧, 对齐设备上的安装孔。

步骤 2: 在设备两侧插入并拧紧螺丝, 固定安装支架。

4. 软件使用指南

4.1 AIB-3588AI-LC-01C Ubuntu 系统

AIB-3588AI-LC-01C 目前支持的系统: Ubuntu 20.04.1 LTS \n \1

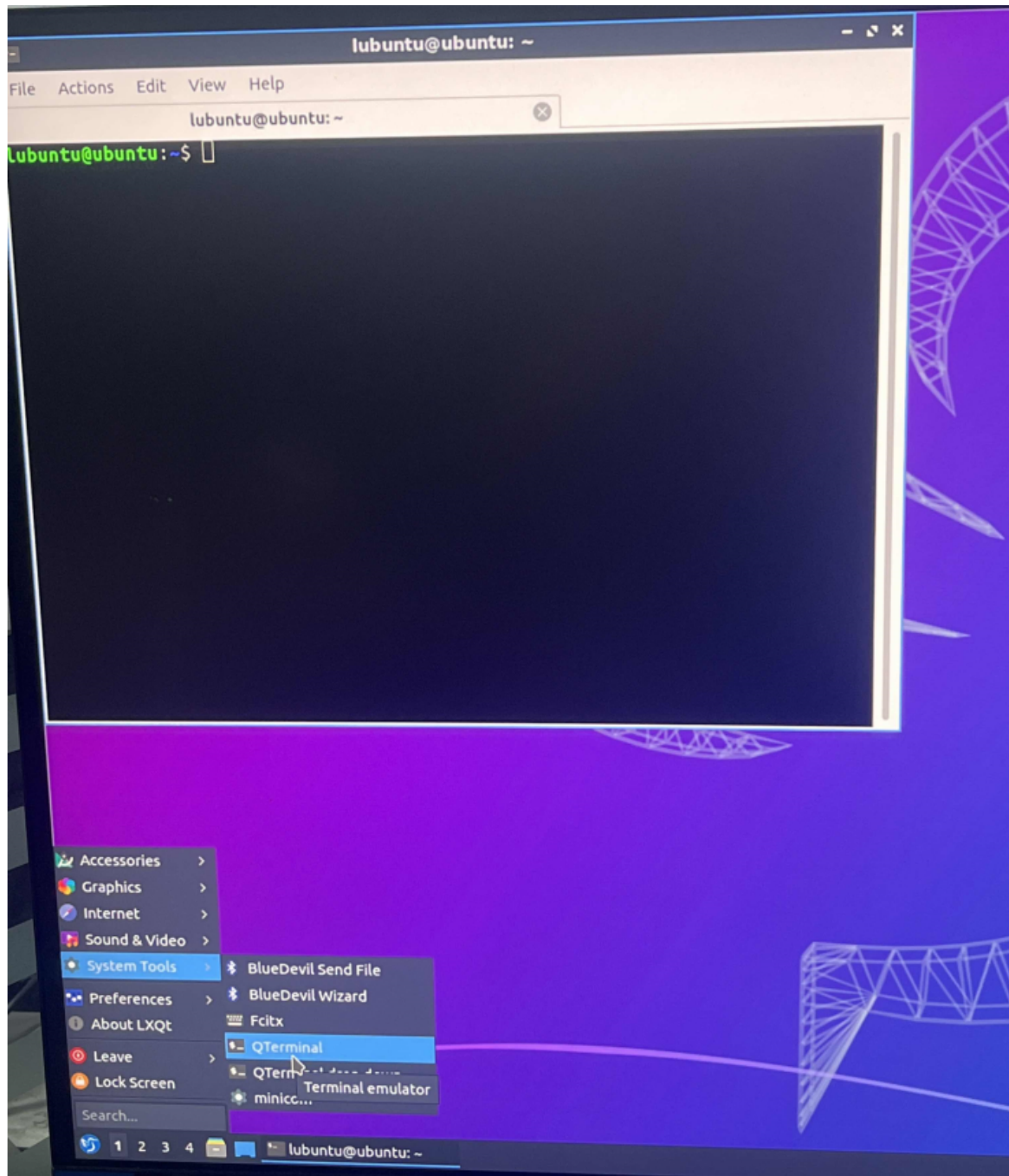


4.2 功能

4.2.1 如何访问系统

4.2.1.1 界面访问

AIB-3588AI-LC-01C Ubuntu 支持 HDMI OUT 输出显示，默认分辨率 1920x1080；连接 HDMI OUT 后上电，等待系统启动完成后，连接键盘，打开系统自带终端应用程序：System Tools ->QTerminal



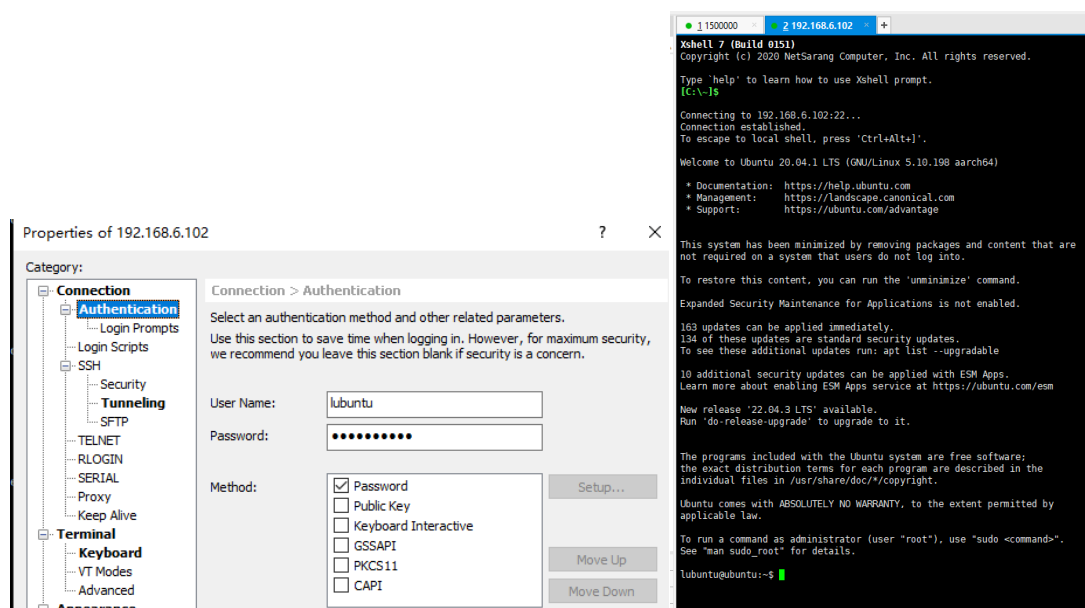
4.2.1.2 远程连接

将 PC 与 AIB-3588AI-LC-01C 连接在同一局域网中，获取 IP 地址并使用 SSH 通过串口工具（SecureCRT/Tera Term）进行连接。

协议：SSH2

用户名: lubuntu

密码: oelinux123

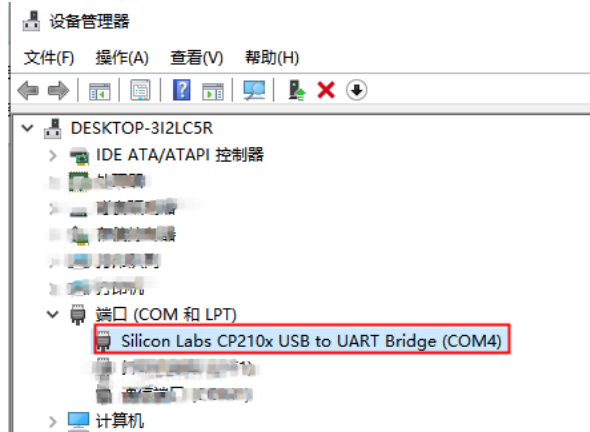


4.2.1.3 串口访问

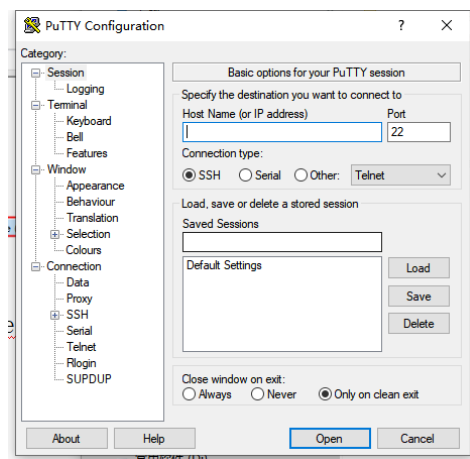
(1) 通过 USB 转 UART 进行串口访问，将 USB 口连接到计算机，另一头连接到板卡上的 DEBUG 口。



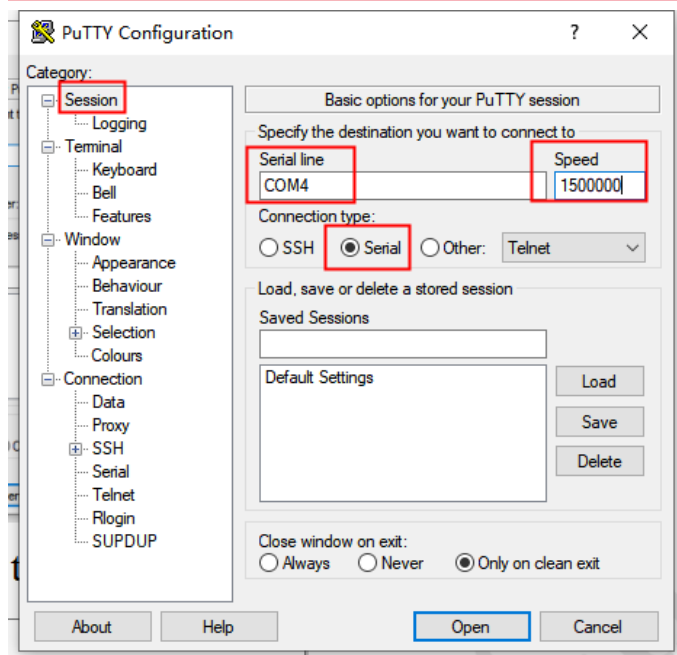
(2) 在 PC 端设备管理器中，可识别到 USB 转 UART 的端口号



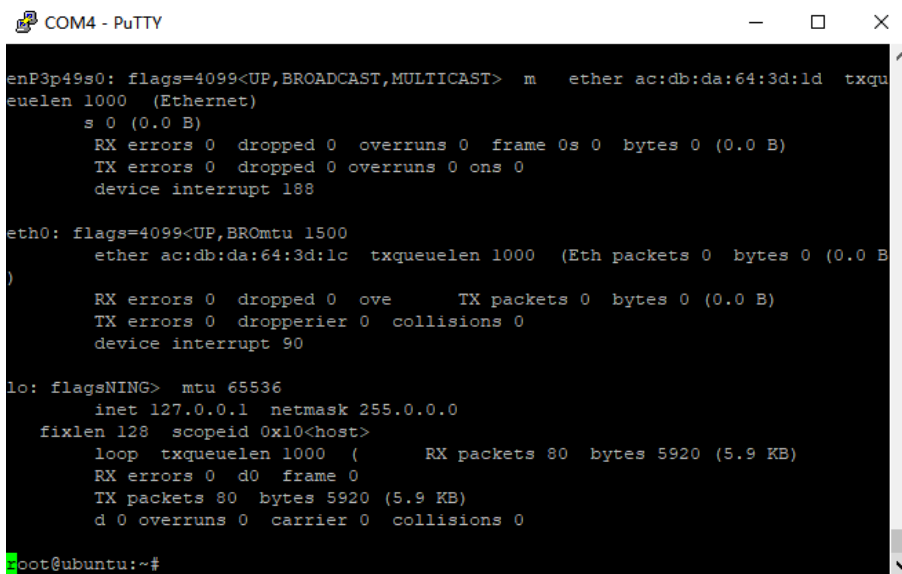
(3) 双击打开: putty.exe 串口工具



(4) 点击“Session”，选择串口“serial”，计算机连接到 UART 后，选择新的串口名称 comx，并设置相应的串口波特率。



(5) 点击“Open”启动工具。



4.2.2 网络功能

4.2.2.1 有线

接入网线，执行命令，查看获取到的 IP 地址；ping 百度，验证网络功能正常

ifconfig //查看 IP 地址

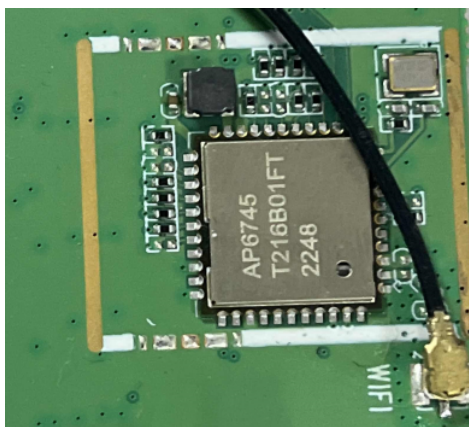
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.6.102 netmask 255.255.255.0 broadcast 192.168.6.255
    inet6 fe80::5096:1406:d1e9:b210 prefixlen 64 scopeid 0x20<link>
    ether ac:db:da:64:3d:1c txqueuelen 1000 (Ethernet)
    RX packets 14799 bytes 1715534 (1.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 13857 bytes 1202011 (1.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 90

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 22294 bytes 2049218 (2.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22294 bytes 2049218 (2.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@ubuntu:~# ping baidu.com
PING baidu.com (110.242.68.66): 56 data bytes
64 bytes from 110.242.68.66: icmp_seq=0 ttl=48 time=28.941 ms
64 bytes from 110.242.68.66: icmp_seq=1 ttl=48 time=28.700 ms
64 bytes from 110.242.68.66: icmp_seq=2 ttl=48 time=28.583 ms
64 bytes from 110.242.68.66: icmp_seq=3 ttl=48 time=28.178 ms
64 bytes from 110.242.68.66: icmp_seq=4 ttl=48 time=29.017 ms
64 bytes from 110.242.68.66: icmp_seq=5 ttl=48 time=28.851 ms
64 bytes from 110.242.68.66: icmp_seq=6 ttl=48 time=28.339 ms
64 bytes from 110.242.68.66: icmp_seq=7 ttl=48 time=29.038 ms
64 bytes from 110.242.68.66: icmp_seq=8 ttl=48 time=28.694 ms
64 bytes from 110.242.68.66: icmp_seq=9 ttl=48 time=28.478 ms
^C--- baidu.com ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.178/28.682/29.038/0.275 ms
root@ubuntu:~#
```

4.2.2.2 WiFi（可选）

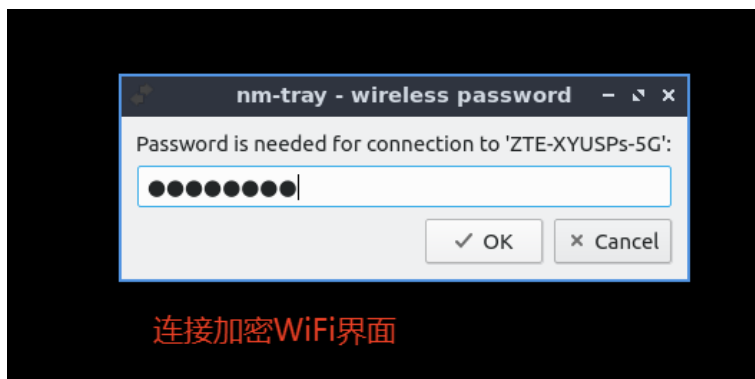
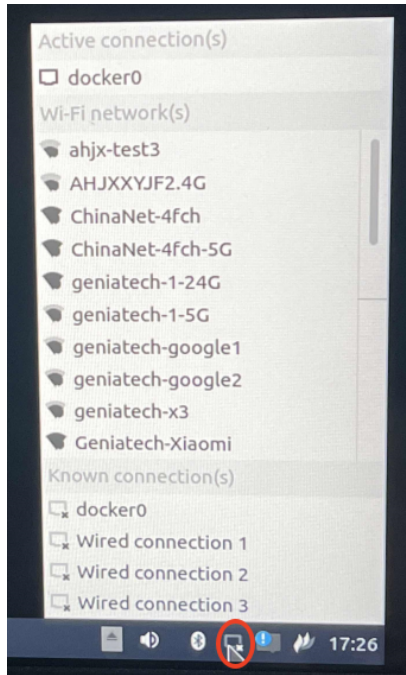
前置条件：硬件上是否贴 WiFi 芯片 ---可选配



WiFi 界面连接：左击右下角“网络状态”图标->选中搜索到的

WiFi->输密码/直接连接，连接成功后“网络状态”图标切换成

；执行指令，查看获取到的 IP 地址，通过 ping 百度，验证网络正常。



```
device interrupt 90

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 3531 bytes 322911 (322.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3531 bytes 322911 (322.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.5.186 netmask 255.255.255.0 broadcast 192.168.5.255
    inet6 fe80::1ba5:6125:9264:d093 prefixlen 64 scopeid 0x20<link>
    ether 50:41:1c:14:3a:1c txqueuelen 1000 (Ethernet)
    RX packets 54 bytes 4398 (4.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 59 bytes 5388 (5.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@ubuntu:/home/lubuntu# ping www.baidu.com
PING www.baidu.com (36.155.132.76): 56 data bytes
64 bytes from 36.155.132.76: icmp_seq=0 ttl=50 time=19.498 ms
64 bytes from 36.155.132.76: icmp_seq=1 ttl=50 time=23.439 ms
64 bytes from 36.155.132.76: icmp_seq=2 ttl=50 time=20.048 ms
64 bytes from 36.155.132.76: icmp_seq=3 ttl=50 time=19.297 ms
64 bytes from 36.155.132.76: icmp_seq=4 ttl=50 time=19.485 ms
64 bytes from 36.155.132.76: icmp_seq=5 ttl=50 time=15.437 ms
64 bytes from 36.155.132.76: icmp_seq=6 ttl=50 time=19.648 ms
64 bytes from 36.155.132.76: icmp_seq=7 ttl=50 time=18.820 ms
64 bytes from 36.155.132.76: icmp_seq=8 ttl=50 time=22.842 ms
64 bytes from 36.155.132.76: icmp_seq=9 ttl=50 time=18.639 ms
64 bytes from 36.155.132.76: icmp_seq=10 ttl=50 time=18.014 ms
^C--- www.baidu.com ping statistics ---
11 packets transmitted, 11 packets received, 0% packet loss
round-trip min/avg/max/stddev = 15.437/19.561/23.439/2.065 ms
root@ubuntu:/home/lubuntu#
```

4.2.2.3 蓝牙（可选） ---同 WiFi 模块一致

蓝牙界面连接：a. 点击右下角“蓝牙”图标->b. 选中并点击

“Devices”，进入“Bluetooth Devices”界面->c. 点击

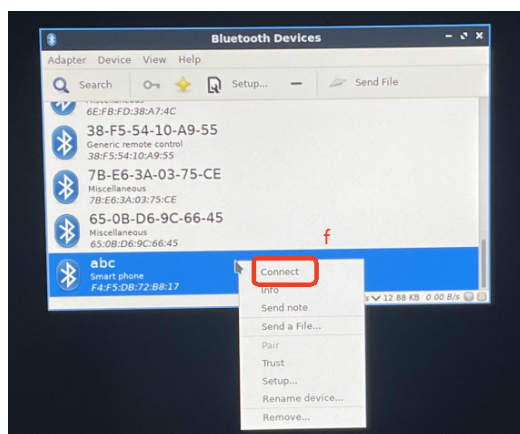
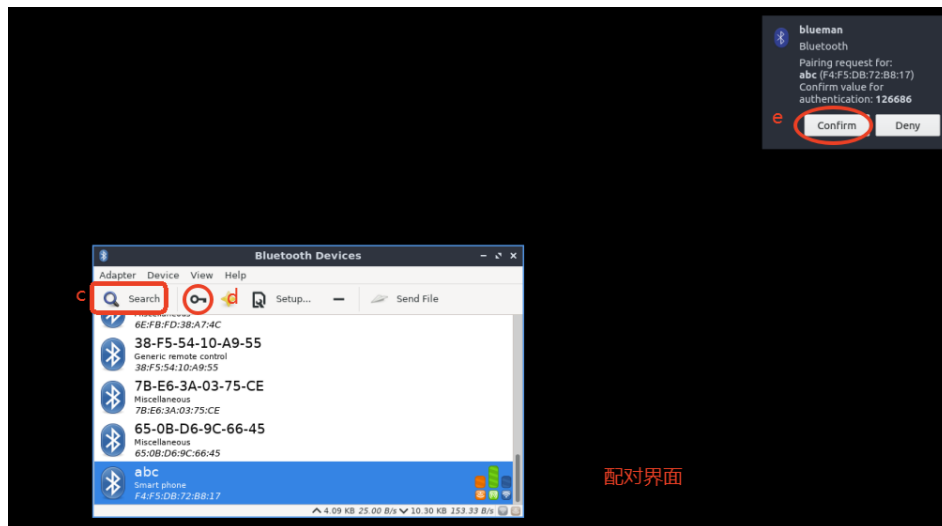
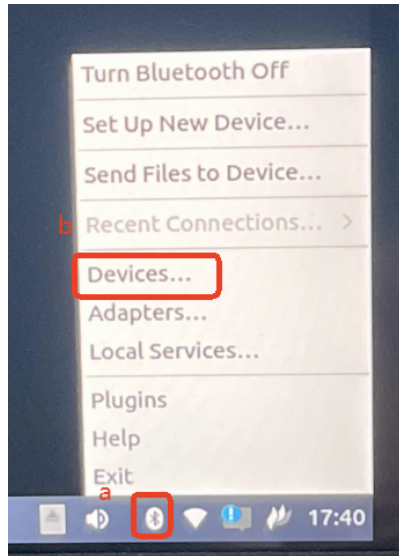
“Search”，搜索环境中蓝牙设备->d. 选中并点击“钥匙”图标，

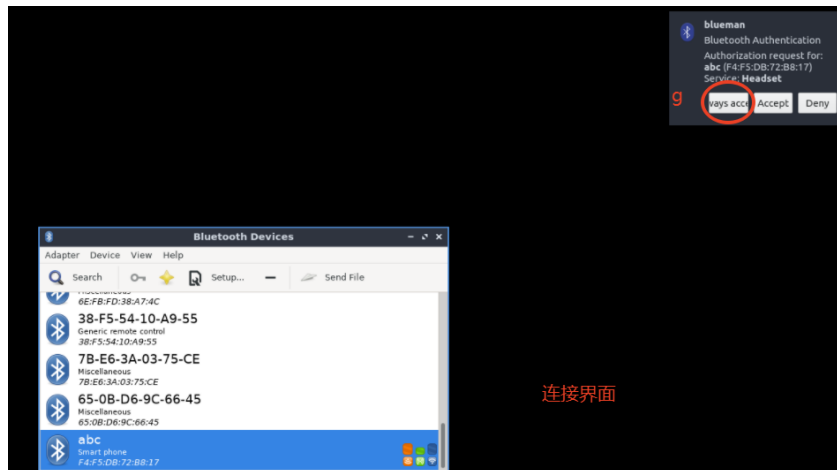
进行配对->e. 完成配对步骤后，右上角弹出提示框，点击

“Confirm”->f. 选中配对蓝牙设备并右击，选中“Connect”，进

行连接->g. 连接成功后，右上角弹出提示框，点击“Always

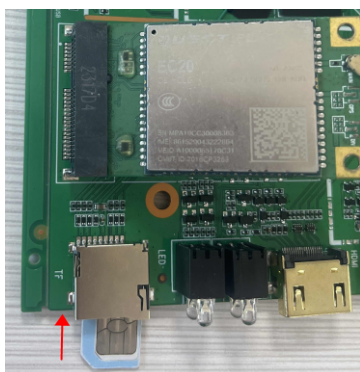
accept”。





4.2.2.4LTE 模块

(1) 准备工作：板卡断电状态下，接入 LTE 4G 模块（如：EC20）& 连接 4G 天线；插入 SIM 卡（缺口朝外）。



(2) 板卡上电，待系统启动完成后；等待 1min30 秒左右，可查询到 ppp0 节点与 IP 地址。通过 ping 百度，验证网络正常。

ifconfig //查看网络节点


```
root@linaro-alip:~# ifconfig
enP2p33s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:55:7b:b5:7d:f7 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 150 base 0x8000

enP3p49s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:55:7b:b5:7d:f7 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 160 base 0x8000

eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 6a:bc:9a:78:85:54 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 82

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 32 bytes 2592 (2.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 2592 (2.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 100.71.142.8 netmask 255.255.255.255 destination 10.64.64.64
    ppp txqueuelen 3 (Point-to-Point Protocol)
    RX packets 906 bytes 95899 (93.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 942 bytes 71518 (69.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

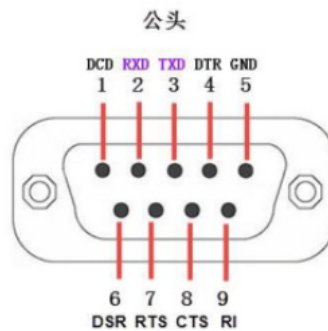
wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 50:41:1c:1d:22:d4 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
root@linaro-alip:~# ping baidu.com -I ppp0
PING baidu.com (39.156.66.10) from 100.71.142.8 ppp0: 56(84) bytes of data
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=52 time=95.2 ms
[ 2016.894224] 2023-12-22 04:26:01
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=52 time=894 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=52 time=97.5 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=4 ttl=52 time=896 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=5 ttl=52 time=101 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=6 ttl=52 time=937 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=7 ttl=52 time=101 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=8 ttl=52 time=936 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=9 ttl=52 time=143 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=10 ttl=52 time=959 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=11 ttl=52 time=100 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=12 ttl=52 time=898 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=13 ttl=52 time=100 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=14 ttl=52 time=856 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=15 ttl=52 time=120 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=16 ttl=52 time=895 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=17 ttl=52 time=139 ms
^C
--- baidu.com ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16438ms
rtt min/avg/max/mdev = 95.233/486.439/958.835/399.112 ms
root@linaro-alip:~#
```

4.2.3RS232*4（DB9/凤凰端子）

4.2.3.1 前置条件：

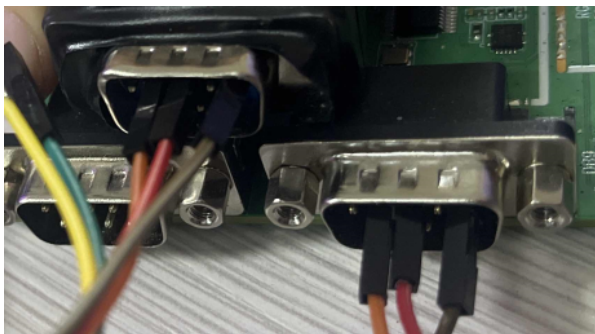
- USB 转 232 线
- 主机 PC（支持 64 位）
- 串口工具（ttermpro）
- 12V 电源线



计算机上的串口工具检测到 USB 转换为 RS232，将 USB 转 232 的 2/3/5 脚与板卡上的 2/3/5 脚对应连接；请选择相应的端口号，Bps/Par/Bits: 115200 8N1。（波特率可以切换，但请保持发送方和接收方设置相同）

4.2.3.2 终端操作：

（1）如下图连接，在 PC 端分别打开 DEBUG 口与 RS232 的调试窗口。



（2）在 Debug 口中输入指令，进行端口号（ttyWCHT0）及波特率等相关设置并保存，测试两个调试窗口能互相收发数据即可。

minicom -s //设置端口号和波特率

Minicom 使用方法：通过上下左右键进入不同选项，进入 serial port setup 中设置端口号和波特率可通过选项中前排字母进行更改（A：更改端口号，E：更改波特率；等）

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup          |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+

```

```
+-----+
| A - Serial Device      : /dev/ttyWCHT0 |
| B - Lockfile Location  : /var/lock      |
| C - Callin Program     :                |
| D - Callout Program    :                |
| E - Bps/Par/Bits       : 115200 8N1     |
| F - Hardware Flow Control : No          |
| G - Software Flow Control : No          |
| Change which setting? █                |
+-----+
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+

```

设置完成之后，

Save setup as df1

Exit

进入接收命令窗口，将显示终端发送数据。

如：打开 232 对应 COM 口，在串口 minicom 界面下输入任意值

（如：11111），232 对应的 COM 口会收到 11111；在 232 COM 口输入任意值（如：22222），串口应会收到 22222

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyWCHT0, 19:33:14

Press CTRL-A Z for help on special keys

22222█
```

```
COM13:115200baud - Tera Term VT
File Edit Setup Control Window Help
11111█
```

备注：以下所有端口号的测试方法同上，验证不同 RS232 接口是要对应不同端口号；端口号示意图见：2.5 和下图。



4.2.3GPS

(1) 在 debug 口中输入指令，设置端口号（/dev/ttyS3） 且需更改波特率为 115200

minicom -s //设置端口号或波特率等信息

Minicom 使用方法：通过上下左右键进入不同选项，进入 serial port setup 中设置端口号和波特率可通过选项中前排字母进行更改（A：更改端口号，E：更改波特率；等）

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+
```

```
+-----[Comm Parameters]-----+
| A - Serial De|
| B - Lockfile Loc| Current: 9600 8N1
| C - Callin Pro| Speed Parity Data
| D - Callout Pro| A: <next> L: None S: 5
| E - Bps/Par/B| B: <prev> M: Even T: 6
| F - Hardware Flo| C: 9600 N: Odd U: 7
| G - Software Flo| D: 38400 O: Mark V: 8
| E: 115200 P: Space
| Change which|
+-----+ Stopbits
| Screen a| W: 1 Q: 8-N-1
| Save set| X: 2 R: 7-E-1
| Save set|
| Exit|
| Exit fro| Choice, or <Enter> to exit? █
+-----+
```

```
+-----+
| A - Serial Device : /dev/ttyS3
| B - Lockfile Location : /var/lock
| C - Callin Program :
| D - Callout Program :
| E - Bps/Par/Bits : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
| Change which setting? █
+-----+
| Screen and keyboard |
| Save setup as df1 |
| Save setup as.. |
| Exit |
| Exit from Minicom |
+-----+
```

(2) 设置完成之后，选中

Save setup as df1

Exit

```
+-----[configuration]-----+
| Filenames and paths
| File transfer protocols
| Serial port setup
| Modem and dialing
| Screen and keyboard
| Save setup as df1 █
| Save setup as..
| Exit
| Exit from Minicom
+-----+
```

(3) 通过 minicom 进入端口后，GPS 不断刷新信息。


```
Welcome to minicom 2.7.1

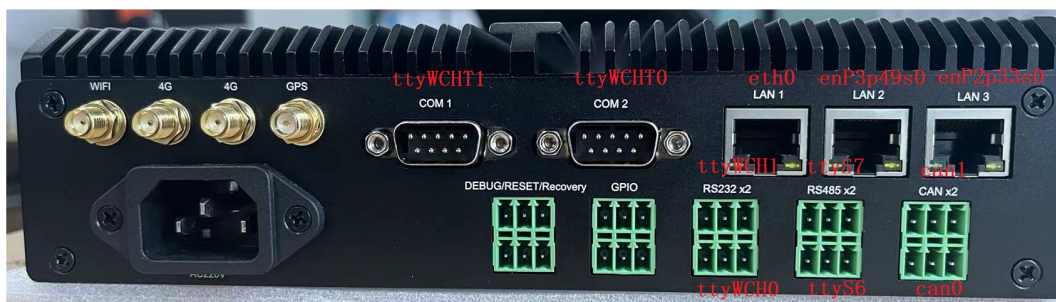
OPTIONS: I18n
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyS3, 17:42:44

Press CTRL-A Z for help on special keys

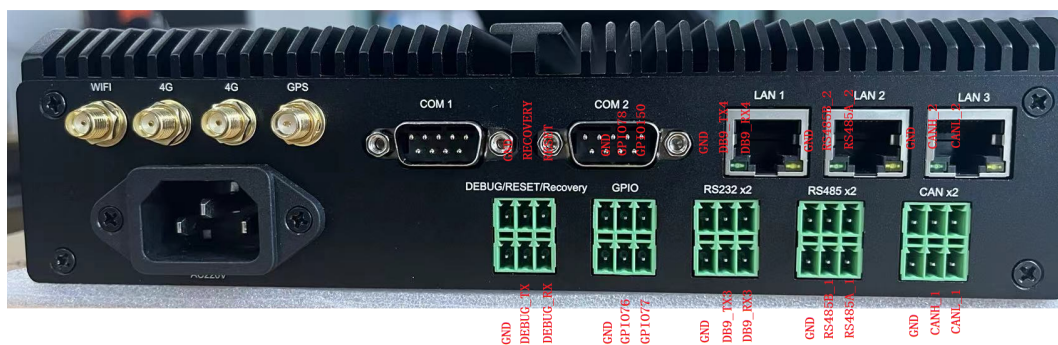
$GNGGA,,,,,0,00,,M,M,*78
$GNGSA,A,1,,,,,,,,,,,1*1D
$GNGSA,A,1,,,,,,,,,,,4*18
$GPGSV,1,1,00*79
$BDGSV,1,1,00*68
$GNRMC,V,,,,,,,,,0.0,E,N,V*5C
$GNZDA,,,00,00*56
$GNTXT,01,01,02,ANT_OPEN,B1,*30
$GNGGA,,,,,0,00,,M,M,*78
$GNGSA,A,1,,,,,,,,,,,1*1D
$GNGSA,A,1,,,,,,,,,,,4*18
$GPGSV,1,1,00*79
$BDGSV,1,1,00*68
$GNRMC,V,,,,,,,,,0.0,E,N,V*5C
$GNTXT,01,01,02,ANT_OPEN,B1,*30
```

4.2.5 转接板功能

接口-文件名对照图



接口-引脚定义图



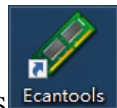
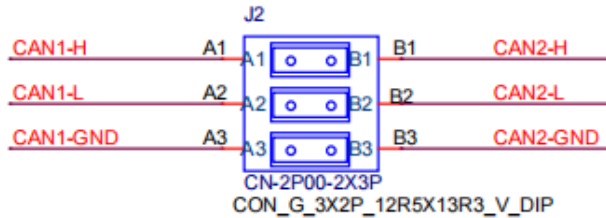
4.2.5.1 CAN

(1) 用 CAN 测试盒连接 PC 与 CAN0 口

参考 CAN 引脚示意图与 CAN 测试盒对应连接，即引脚 2 3 7 对应 L

G H

CAN



(2) 打开 PC 上的 Ecantools，点击打开设备并确认



(3) 串口查看确认已识别到 can 设备

ifconfig -a

```
root@linaro-alip:~# ifconfig -a
can0: flags=128<NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 91

can1: flags=128<NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 92

enP2p33s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:55:7b:b5:7d:f7 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 149 base 0x8000

enP3p49s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:55:7b:b5:7d:f7 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
```

(4) 串口输入如下指令:

ip link set can0 type can bitrate 500000 triple-sampling on

ifconfig can0 up

candump can0 //接收指令

cansend can0 123#0102030405060670780980 //发送指令

(5) Ecantools 按如下步骤自动识别波特率, 识别成功后; 点击发送, 串口会收到数据。(备注: CAN1 节点操作步骤同上, 只需更换指令中节点名即可)

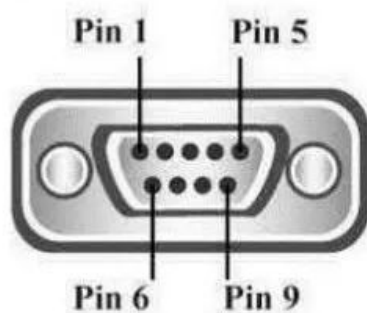


- 主机 PC（支持 64 位）
- 串口工具（ttermpro）
- 12V 电源线



Pin 1	TXD-
Pin 2	TXD+
Pin 3	RTS-
Pin 4	RTS+
Pin 5	GND
Pin 6	RXD-
Pin 7	RXD+
Pin 8	CTS
Pin 9	CTS+

RS422/485 Pinout (9 Pin)

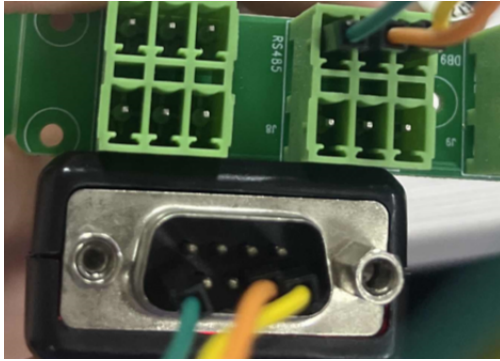


硬件

计算机上的串口工具检测到 USB 转换为 RS485，将 USB 转 485 的 1/2/5 脚与板卡上的 TX/RX/GND 脚对应连接；请选择相应的端口号，Bps/Par/Bits: 115200 8N1。（波特率可以切换，但请保持发送方和接收方设置相同）

4.2.5.2.2 终端操作：

（1）如下图连接，在 PC 端分别打开 DEBUG 口与 RS485 的调试窗口。



(2) 在 Debug 口中输入指令，进行端口号（ttyS6）及波特率等相关设置并保存，测试两个调试窗口能互相收发数据即可。

minicom -s //设置端口号和波特率

Minicom 使用方法：通过上下左右键进入不同选项，进入 serial port setup 中设置端口号和波特率可通过选项中前排字母进行更改（A：更改端口号，E：更改波特率；等）

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                          |
| Exit from Minicom            |
+-----+

```

```
+-----+
| A - Serial Device      : /dev/ttyS6 |
| B - Lockfile Location  : /var/lock   |
| C - Callin Program     :             |
| D - Callout Program    :             |
| E - Bps/Par/Bits       : 115200 8N1  |
| F - Hardware Flow Control : No       |
| G - Software Flow Control : No       |
| Change which setting? █             |
+-----+
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+-----+

```

设置完成之后，

Save setup as dfl

Exit

进入接收命令窗口，将显示终端发送数据。

如：打开 485 对应 COM 口，在串口 minicom 界面下输入任意值

（如：11111），485 对应的 COM 口会收到 11111；在 485 COM 口输入任意值（如：22222），串口应会收到 22222

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyS6, 19:54:02

Press CTRL-A Z for help on special keys

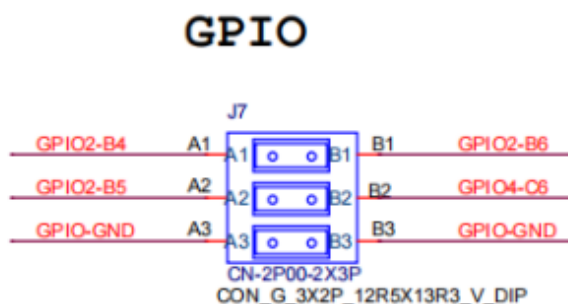
22222
```

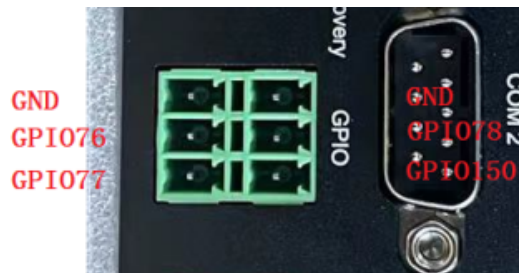
```
COM22:115200baud - Tera Term VT
File Edit Setup Control Window Help
11111
```

备注：以下所有端口号的测试方法同上，验证不同 RS485 接口是要对应不同端口号；端口号示意图见：2.5。

4.2.5.3GPIO

（1）GPIO 引脚示意图与实际引脚换算





(2) 查看 `cat /etc/autoload.sh` 配置文件中，默认所有 GPIO 口拉高

```
#for gtiot con
#2_B4
echo 76 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio76/direction
echo 1 >/sys/class/gpio/gpio76/value
#2_B5
echo 77 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio77/direction
echo 1 >/sys/class/gpio/gpio77/value
#2_B6
echo 78 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio78/direction
echo 1 >/sys/class/gpio/gpio78/value
#4_C6
echo 150 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio150/direction
echo 1 >/sys/class/gpio/gpio150/value
```

(3) 输入如下指令，控制 GPIO 口电压高低：

`echo 1 >/sys/class/gpio/gpio76/value` //电压拉高 5.24V

`echo 0 >/sys/class/gpio/gpio76/value` //电压拉低 0V

4.2.6RTC

指令设置：在 debug 口执行以下指令

`date -s "2024-01-01 22:43:00"` //设置系统时间

`hwclock -w` //时间写入硬件时钟

`date` //读取系统时间

`hwclock -r` //读取硬件时钟时间

4.2.7 Watchdog

(1) 通过 Debug 口执行以下指令，查看系统默认模式&喂狗时间。

`cat /sys/class/rtc_am1805/am1805_timer` //查看当前默认时长

`cat /sys/class/rtc_am1805/feed_dog` //查看模式

```
root@linaro-alip:~#  
root@linaro-alip:~# cat /sys/class/rtc_am1805/am1805_timer  
[ 7318.423493] am1805_timer_show ,run  
4000  
root@linaro-alip:~# cat /sys/class/rtc_am1805/feed_dog  
1 -- kernel feed  
root@linaro-alip:~#
```

(2) Debug 口执行以下指令，切换不同模式/喂狗时间

`echo 0 > /sys/class/rtc_am1805/feed_dog` //watchdog 关闭，

watchdog 不工作，系统不会重启

`echo 1 > /sys/class/rtc_am1805/feed_dog` //内核模式

`echo 2000 > /sys/class/rtc_am1805/am1805_timer` //如果间隔

时间手动改为低于 2000ms，内核来不及喂狗就会触发 WDT reset

`echo 2 > /sys/class/rtc_am1805/feed_dog` //用户模式，系统

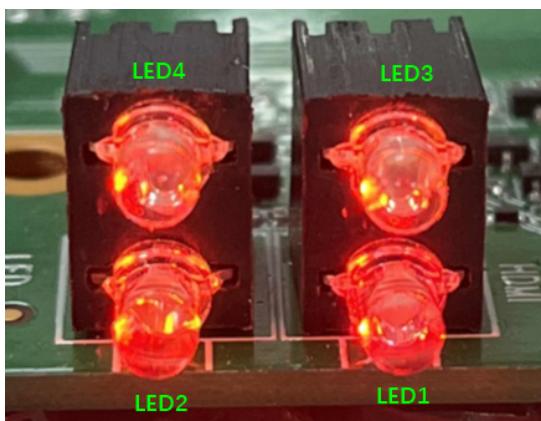
中此模式喂狗的时间间隔默认是 4s，设置后 4s 系统自动重启

`echo 10000 > /sys/class/rtc_am1805/am1805_timer` //更改喂

狗时间，设置 10s 后系统自动重启

4.2.8 LED 状态灯

(1) 查看 `cat /etc/autoload.sh` 配置文件中，默认所有 LED 灯的 GPIO 口拉高(红绿灯均点亮)---自定义：LED1/2/3/4，如下



```
#for led
#led 1
echo 50 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio50/direction
echo 1 >/sys/class/gpio/gpio50/value
#led 2
echo 51 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio51/direction
echo 1 >/sys/class/gpio/gpio51/value
#led 3
echo 52 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio52/direction
echo 1 >/sys/class/gpio/gpio52/value
#led 4
echo 53 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio53/direction
echo 1 >/sys/class/gpio/gpio53/value
#led 5
echo 54 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio54/direction
echo 1 >/sys/class/gpio/gpio54/value
#led 6
echo 55 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio55/direction
echo 1 >/sys/class/gpio/gpio55/value
#led 7
echo 60 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio60/direction
echo 1 >/sys/class/gpio/gpio60/value
#led 8
echo 61 >/sys/class/gpio/export
echo out >/sys/class/gpio/gpio61/direction
echo 1 >/sys/class/gpio/gpio61/value
root@linaro-alip:~#
```

(2) 输入如下指令，控制 LED 灯状态（点亮/熄灭）：

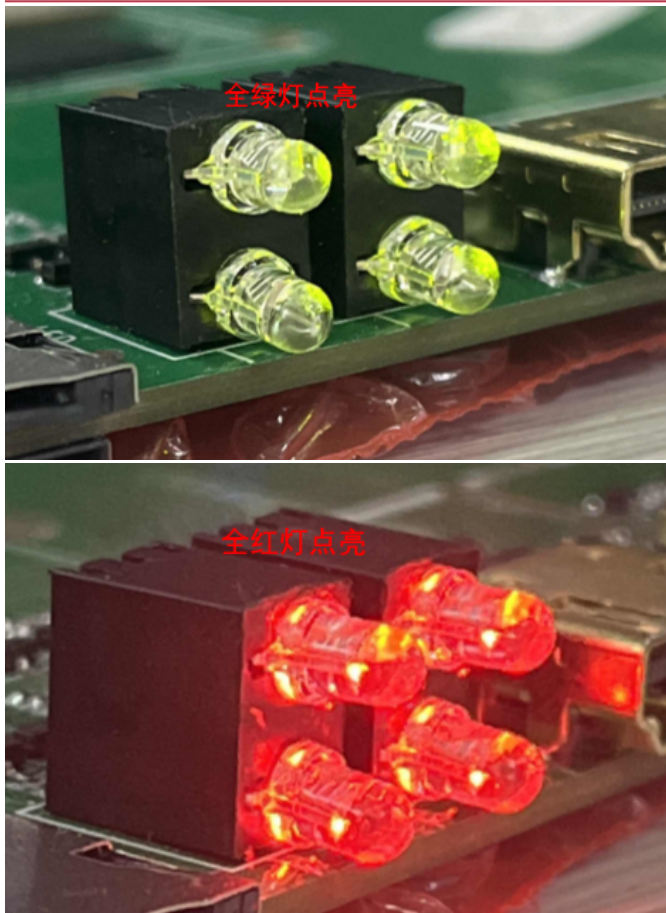
`echo 0 >/sys/class/gpio/gpio52/value` //LED2 红灯熄灭

`echo 1 >/sys/class/gpio/gpio52/value` //LED2 红灯点亮

`echo 0 >/sys/class/gpio/gpio53/value` //LED2 绿灯熄灭

`echo 1 >/sys/class/gpio/gpio53/value` //LED2 绿灯点亮

备注：GPIO50 -> LED1 红灯，GPIO51 -> LED1 绿灯（当前 LED1 绿灯常亮，无法熄灭），GPIO54 -> LED3 红灯，GPIO55 -> LED3 绿灯，GPIO60 -> LED4 红灯，GPIO61 -> LED4 绿灯。



4.2.9 HDMI IN

4.2.9.1 前置条件:

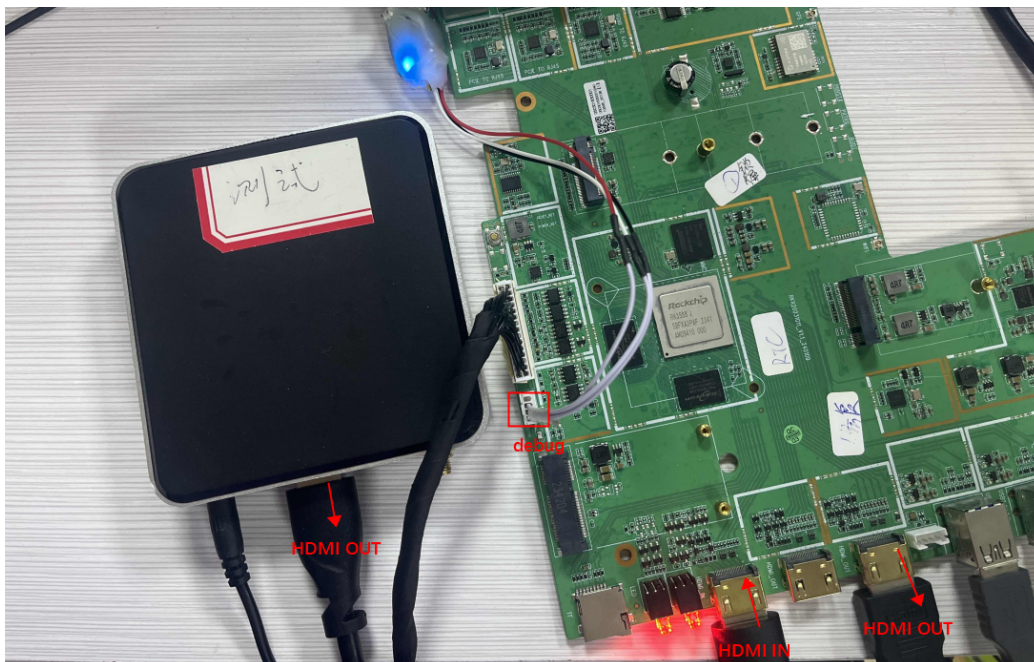
- AIB-3588AI-LC-01C Ubuntu 系统设备
- HDMI 线
- 任意带有 HDMI OUT 接口的设备

如图所示连接方式:



4.2.9.2 操作步骤:

(1) 连接 APC3588 与 HDMI IN 输入源(1920x1080P60)&调试串口, USB 鼠标; 如图所示下图:



(2) 需手动安装: `apt-get install gstreamer1.0-plugins-good`


```

root@ubuntu:~# apt-get install gstreamer1.0-plugins-good
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gstreamer1.0-plugins-base gstreamer1.0-x libaa1 libavc1394-0 libcac0
  libcdparanoia0 libdv4 libgstreamer-plugins-good1.0-0 libiec61883-0
  libraw1394-11 libshout3 libtag1v5 libtag1v5-vanilla libvisual-0.4-0
Suggested packages:
  libdv-bin oss-compat libraw1394-doc libvisual-0.4-plugins
The following NEW packages will be installed:
  gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-x libaa1
  libavc1394-0 libcac0 libcdparanoia0 libdv4 libgstreamer-plugins-good1.0-0
  libiec61883-0 libraw1394-11 libshout3 libtag1v5 libtag1v5-vanilla
  libvisual-0.4-0
0 upgraded, 15 newly installed, 0 to remove and 163 not upgraded.
Need to get 3016 kB of archives.
After this operation, 11.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://mirrors.ustc.edu.cn/ubuntu-ports focal/main arm64 libcdparanoia0 arm64 3.10.2+debian-13 [39
kB]
Get:2 https://mirrors.ustc.edu.cn/ubuntu-ports focal/main arm64 libvisual-0.4-0 arm64 0.4.0-17 [91.3 kB]
Get:3 https://mirrors.ustc.edu.cn/ubuntu-ports focal-updates/main arm64 gstreamer1.0-plugins-base arm64 1
.3-0ubuntu1.3 [543 kB]
Get:4 https://mirrors.ustc.edu.cn/ubuntu-ports focal/main arm64 libaa1 arm64 1.4p5-46 [45.5 kB]

```

(3) 更改脚本/etc/gst.sh: 在最后一行添加 lubuntu

`vi /etc/gst.sh` //更改脚本

`cat /etc/gst.sh` //查看脚本内容

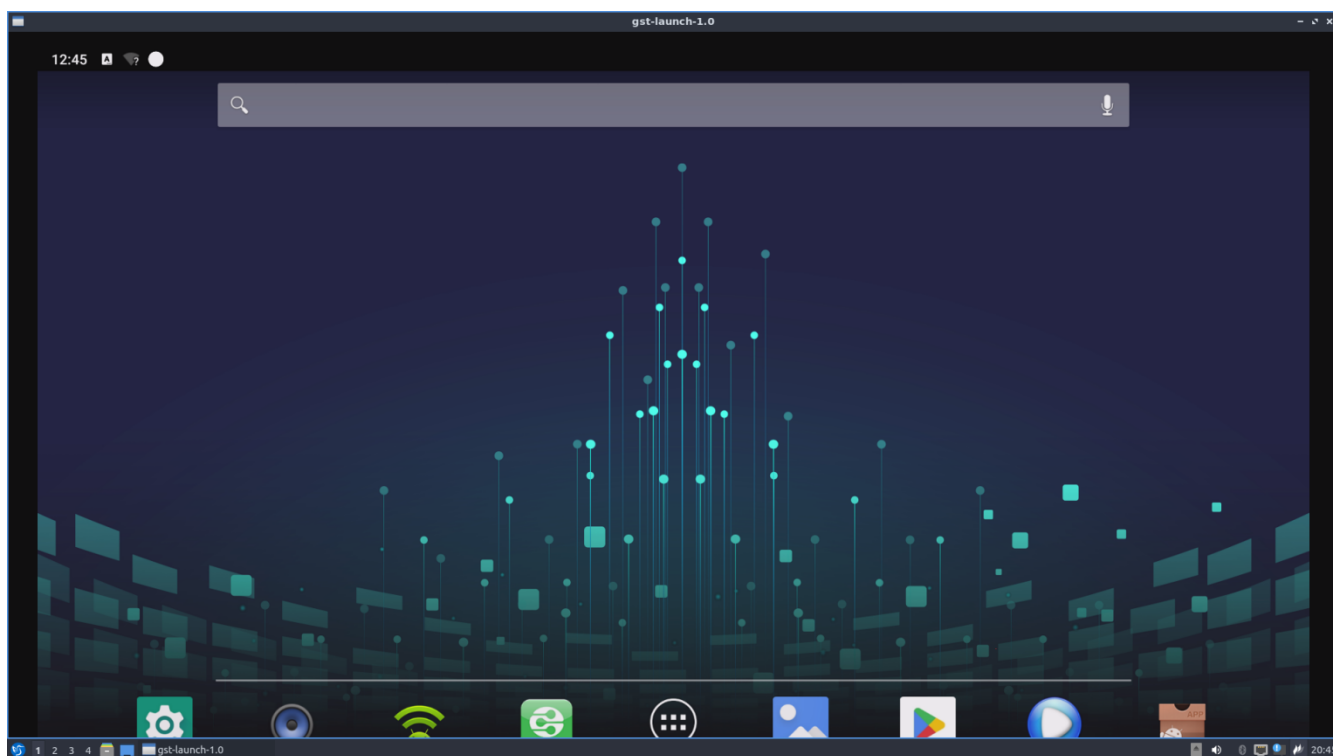
```

root@ubuntu:~# cat /etc/gst.sh
#!/bin/sh
export DISPLAY=:0.0
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/aarch64-linux-gnu/gstreamer-1.0
echo "Start RKISP Camera Preview!"
#media-ctl -d /dev/media0 --set-v4l2 '"rkisp-isp-subdev":0[fmt:UYVY8_2X8/1920x1080]'
#media-ctl -d /dev/media0 --set-v4l2 '"rkisp-isp-subdev":0[crop:(0,0)/1920x1080]'
#media-ctl -d /dev/media0 --set-v4l2 '"rkisp-isp-subdev":2[fmt:UYVY8_2X8/1920x1080]'
#media-ctl -d /dev/media0 --set-v4l2 '"rkisp-isp-subdev":2[crop:(0,0)/1920x1080]'
#v4l2-ctl -d /dev/video20 --set-selection=target=crop,top=0,left=0,width=1920,height=1080
arecord -D hw:2,0 --period-size=128 --buffer-size=256 -c 1 -f dat -t raw | aplay -D hw:0,0 -c 1 -f dat &
su lubuntu -c "gst-launch-1.0 v4l2src device=/dev/video20 ! video/x-raw,width=1920,height=1080 ! videoconver
t ! ximagesink sync=false"
root@ubuntu:~#
Unpacking libtag1v5:arm64 (1.11.1+dfsg-1-0.3ubuntu2) ...

```

(4) 进入 etc 路径下, 运行脚本: `./gst.sh`

```
root@ubuntu:~# cd /etc/
root@ubuntu:/etc# ./gst.sh
Start RKISP Camera Preview!
Recording raw data 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
Playing raw data 'stdin' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
bash: warning: setlocale: LC_ALL: cannot change locale (zh_CN.UTF-8)
[ 3194.864673] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[0] get remote terminal sensor failed!
[ 3194.864882] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[1] get remote terminal sensor failed!
[ 3194.864890] stream_cif_mipi_id1: update sensor info failed -19
[ 3194.865075] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[2] get remote terminal sensor failed!
[ 3194.865083] rkCIF_tools_id2: update sensor info failed -19
[ 3194.865269] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[2] get remote terminal sensor failed!
[ 3194.865277] stream_cif_mipi_id2: update sensor info failed -19
[ 3194.865463] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[3] get remote terminal sensor failed!
[ 3194.865471] stream_cif_mipi_id3: update sensor info failed -19
[ 3194.865656] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[0] get remote terminal sensor failed!
[ 3194.865665] rkCIF_scale_ch0: update sensor info failed -19
[ 3194.865849] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[1] get remote terminal sensor failed!
[ 3194.865857] rkCIF_scale_ch1: update sensor info failed -19
[ 3194.866041] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[2] get remote terminal sensor failed!
[ 3194.866049] rkCIF_scale_ch2: update sensor info failed -19
[ 3194.866232] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[3] get remote terminal sensor failed!
[ 3194.866240] rkCIF_scale_ch3: update sensor info failed -19
[ 3194.866425] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[0] get remote terminal sensor failed!
[ 3194.866433] rkCIF_tools_id0: update sensor info failed -19
[ 3194.866618] rkCIF-mipi-lvds2: rkCIF_update_sensor_info: stream[1] get remote terminal sensor failed!
[ 3194.866625] rkCIF_tools_id1: update sensor info failed -19
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
0:00:01.4 / 99:99:99.
```



4.2.10 SSD

接入 SSD 设备后，执行以下指令，手动挂载 SSD；进入该文件路径

Room 02-04, 10/F, Block A, Building 8, Shenzhen International Innovation Valley, Dashi Road,

42 / 44

Nanshan District, Shenzhen, Guangdong, China

Email: support@geniatech.com Tel: (+ 86) 755 86028588

下

fdisk -l //查看 SSD 在系统分区

mount /dev/sda /media/ //手动挂载 SSD

```
root@ubuntu:~# fdisk -l
Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes

Disk /dev/mmcblk0: 115.24 GiB, 123731968000 bytes, 241664000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: D2310000-0000-4C52-8000-015A00001C8B

Device            Start      End  Sectors  Size Type
/dev/mmcblk0p1     16384     24575     8192    4M unknown
/dev/mmcblk0p2     24576     32767     8192    4M unknown
/dev/mmcblk0p3     32768    163839    131072   64M unknown
/dev/mmcblk0p4     163840    425983    262144  128M unknown
/dev/mmcblk0p5     425984    491519     65536   32M unknown
/dev/mmcblk0p6     491520   42434559  41943040  20G unknown
/dev/mmcblk0p7     42434560  42696703    262144  128M unknown
/dev/mmcblk0p8     42696704  241663966 198967263 94.9G unknown

Disk /dev/nvme0n1: 119.25 GiB, 128035676160 bytes, 250069680 sectors
Disk model: aigo NVMe SSD P2000 128GB
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@ubuntu:~#
```

```
root@ubuntu:~# mount /dev/nvme0n1 /media/
root@ubuntu:~# ls /media/
sdb
root@ubuntu:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        20G   3.1G   16G   17% /
devtmpfs        3.7G     0   3.7G    0% /dev
tmpfs           3.9G     0   3.9G    0% /dev/shm
tmpfs           793M   2.3M  790M    1% /run
tmpfs           5.0M   4.0K   5.0M    1% /run/lock
tmpfs           3.9G     0   3.9G    0% /sys/fs/cgroup
tmpfs           3.9G   4.0K   3.9G    1% /tmp
/dev/mmcblk0p7   15M   12M   1.6M   89% /oem
/dev/mmcblk0p8   92G  279K   91G    1% /userdata
tmpfs           793M   8.0K  793M    1% /run/user/1000
tmpfs           793M     0  793M    0% /run/user/0
/dev/nvme0n1    117G   12K  111G    1% /media
root@ubuntu:~#
```

4.2.11 算力棒

接入算力棒设置，执行以下指令：

hailortcli run /etc/joyao_STM_plate.hef //自动结算

lspci //硬件识别

```
root@ubuntu:~# lspci
0000:00:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3588 (rev 01)
0000:01:00.0 Co-processor: Device 1e60:2864 (rev 01)
0002:20:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3588 (rev 01)
0002:21:00.0 Ethernet controller: Device 1f0a:6801 (rev 01)
0003:30:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3588 (rev 01)
0003:31:00.0 Ethernet controller: Device 1f0a:6801 (rev 01)
root@ubuntu:~# hailortcli run /etc/joyao_STM_plate.hef
Running streaming inference (/etc/joyao_STM_plate.hef):
  Transform data: true
    Type: auto
    Quantized: true
Network resnet_v1_18/resnet_v1_18: 100% | 12597 | FPS: 2516.42 | ETA: 00:00:00
> Inference result:
  Network group: resnet_v1_18
    Frames count: 12597
    FPS: 2516.49
    Send Rate: 3030.42 Mbit/s
    Recv Rate: 20.13 Mbit/s
```

AIB-3588AI-LC-01C 是一款基于 RK3588J 处理器设计的人工智能边缘计算盒子，外设资源丰富，接口齐全，搭载八核 64 位 CPU，主频高达 2.4GHz，集成 ARM Mali-G610 MP4 四核 GPU，内置人工智能加速器 NPU，可提供 6TOPs 的计算能力，支持三网口、蓝牙、Wi-Fi、音视频、摄像头、4G、SSD 硬盘、GPS、外扩算力等。它支持 RS232、RS485 UART、CAN 等本地通信，并具有各种视频输入和输出接口。适用于智慧交通、智能 NVR、云终端、工业控制、信息发布终端、多媒体广告机等场景。功能齐全，使用方便。

- (I) 支持 32 通道 1080P 30FPS，也支持 8K 视频同编同解。
- (II) 支持 LPDDR 16GB 和 EMMC 128GB。
- (III) 支持 HDMI IN 信号输入 和 USB camera 组合。
- (IV) 支持双 HDMI Output。
- (V) 支持 USB3.0/USB2.0/GPIO/CAN/RS485/RS232/TTL 外围接口。
- (VI) 支持 M.2 KEY M/M.2 KEY B /MINI PCIE 外围扩展接口
- (VII) 支持三个千兆有线网口，
- (VIII) 支持多操作系统: Debian11 & Ubuntu 20.04
- (IX) 内置看门狗和 RTC 电池，可以支持 7x24 小时稳定工作。
- (X) 支持内置芯动力 K230IT 算力模块