

SW7990E VER2.1 PF1331

技术规格书

Bozz Technology

发布版本:V1.1

日期:2019.11.2

免责声明

您购买的产品、服务或特性等应受深圳博时特科技有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，深圳博时特科技有限公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为技术规格说明和使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

版权所有 © 深圳博时特科技有限公司 2019

非经本公司许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

深圳博时特科技有限公司

地址：深圳市龙华新区油松路 101-1 (华油工业区内 101)

网址：www.bozztek.com

客户服务电话：0755-29307923

客户服务传真：0755-29524432

客户服务邮箱：sales@bozztek.com

 深圳博时特科技有限公司
Bozz Technology(shenzhen),Co.,Ltd

前言

概述

本文档主要介绍 SW7990E VER2.1 基本功能特点和硬件特性、多功能硬件配置、软件调试操作使用方法，旨在帮助开发人员更快、更准确地使用 SW7990E VER2.1 进行应用开发，熟悉 SW7990E VER2.1 解决方案。

产品版本

本文档对应的产品版本如下：

产品名称	平台名称	产品版本
PF1331	SW7990E VER2.1	

适用对象

本文档主要适用于以下工程师：

- 技术支持工程师
- 硬件开发工程师
- 嵌入式软件开发工程师
- 应用软件开发工程师
- 测试工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前说有文档版本的更新内容。

修订日期	版本号	作者	修订说明
2019-11-2	V1.0	卢君豪、农阳州	初始发布
2019-11-2	V1.1	卢君豪	修正硬件

缩略语

缩略语包括文档中常用词组的简称。

DDR	Double Data Rate	双倍速率同步动态随机存储器
eMMC	Embedded Multi Media Card	内嵌式多媒体存储卡
I ² C	Inter-Integrated Circuit	内部整合电路(两线式串行通讯总线)
JTAG	Joint Test Action Group	联合测试行为组织定义的一种国际标准测试协议 (IEEE 1149.1 兼容)
LDO	Low Drop Out Linear Regulator	低压差线性稳压器
LVDS	Low-Voltage Differential Signaling	低电压差分信号
MIPI	Mobile Industry Processor Interface	移动产业处理器接口
PMIC	Power Management IC	电源管理芯片
PMU	Power Management Unit	电源管理单元
RK	Rockchip Electronics Co.,Ltd.	瑞芯微电子股份有限公司
SD Card	Secure Digital Memory Card	安全数码卡
SDIO	Secure Digital Input and Output	安全数字输入输出接口
SDMMC	Secure Digital Multi Media Card	安全数字多媒体存储卡
TF Card	Micro SD Card (Trans-flash Card)	外置记忆卡
USB	Universal Serial Bus	通用串行总线

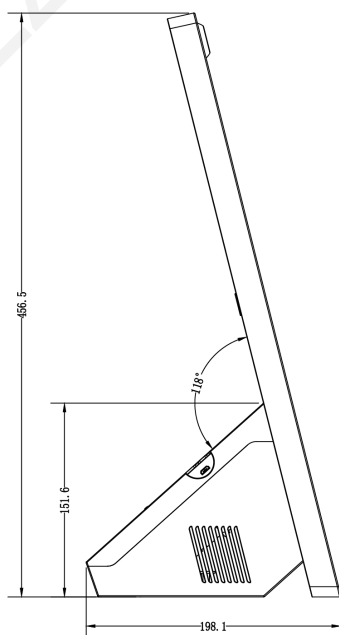
一. 博时特 PF1331 产品功能概述

1.1 PF1331 产品外观

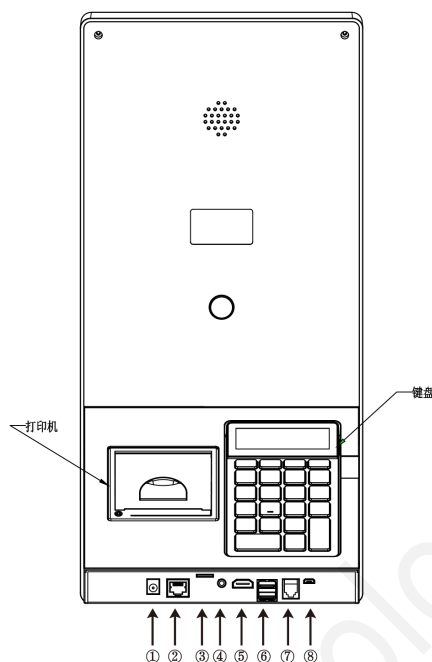
1.1.1 主屏



1.1.3 产品尺寸



1.1.4 外部接口

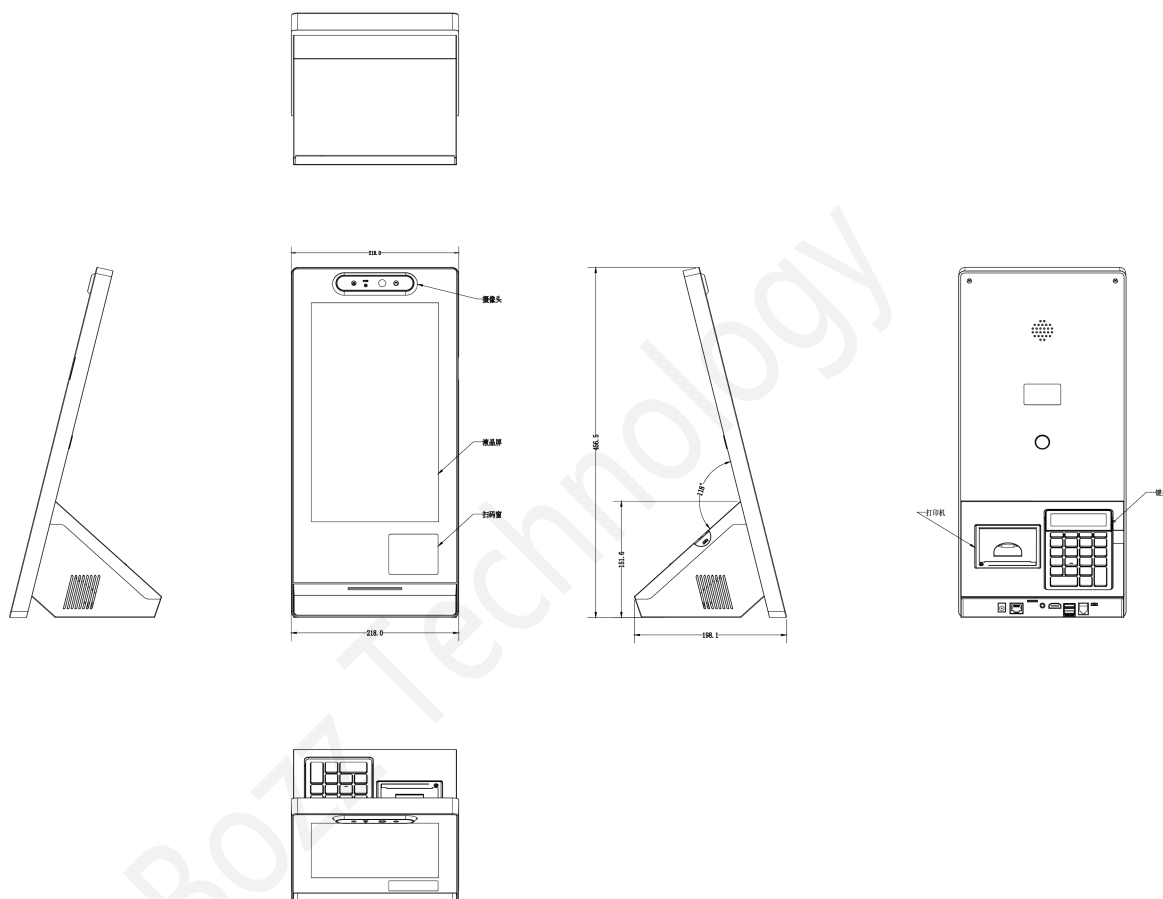


接口说明:

编号	接口	数量	用途
①	电源	1 个	连接电源适配器 12V/3A
②	有线网口	1 个	连接有线网络
③	TF 卡槽	1 个	插入 TF 存储卡
④	音频接口	1 个	连接音箱或耳机
⑤	HDMI 口	1 个	输出 HDMI 信号
⑥	USB 口	2 个	连接 USB 设备, U 盘、键盘、鼠标、移动硬盘等
⑦	RJ11 口	1 个	连接收银机, 钱箱等

⑧	Micro USB 口	1 个	调试
---	-------------	-----	----

1.1.4 PF1331 六面线框图



1.2 PF1331 产品功能概述

PF1331 产品是我司基于 RK3399 芯片开发的单屏支付设备，支持人脸识别支付、扫码支付、小票打印、小键盘收银等。

主芯片搭载 Dual-core Cortex-A72 up to 1.8GHz、Quad-core Cortex-A53 up to 1.4GHz 和 Mali-T864 GPU，运行内存 4GB，存储 8GB，搭载 Android 7.1.2 系统。

配置 13.3 英寸 IPS 高清显示屏、13.3 英寸电容触摸屏、华捷 A200 3D 结构光摄像头、WIFI 模组、扬声器；

支持多点电容触摸，人脸识别，以太网、WIFI 多种联网模式和多种格式的音视频编解码。外围接口配置 USB 和 MICRO USB 接口，可通过 USB 升级和扩展连接各种 USB 设备。

1.3 PF1331 产品硬件规格

CPU	Dual-core Cortex-A72 up to 1.8GHz Quad-core Cortex-A53 up to 1.4GHz Mali-T864 GPU RK3399 瑞芯微电子
ROM	eMMC 8G
RAM	4G 双通道 64 位 LPDDR4/1600MHz
系统	Android 系统：7.1.2
触摸屏	主屏：I2C 或者 USB 通讯透射式电容触摸屏
屏幕	双 Mipi 或者 EDP 和 DP
通讯接口	USBx2：A 类母头/USB2.0 HDMI：A 类母头/ 1.4 and 2.0 operation RJ11 ： RS232 x1 UART ： TTL3.3V x4/RS232 x1/485 x2 内部 USB ： USB x4 USB-OTGx1 ： Micro USB 母座 以太网口：10/100M/1000M

	无线:2.4G WIFI 钱箱: x1
摄像头	华捷 A200 结构光摄像头: 接口类型: HOST/USB 传感器像素: 200 万 最高分辨率: 1920*1080
耳机	3.5MM 耳机输入
扬声器	内阻: 8 欧姆 额定功率: 2W 磁路形式: 内磁式 音强: 不低于 90 分贝

1.4 PF1331 产品行业应用市场概述

PF1331 是一款桌面式单屏聚合支付设备，具备华捷 A200 结构光摄像头，具有核验速度快、准确度高、人脸防伪、用户交互性好等特性。产品采用高清屏设计，整机外观精致大方，用户界面简洁直观。

可适用于商场、超市、小卖部等场所，进店购买商品的客户可以选择微信刷脸支付，支付宝，微信，银联云闪付等扫码支付。

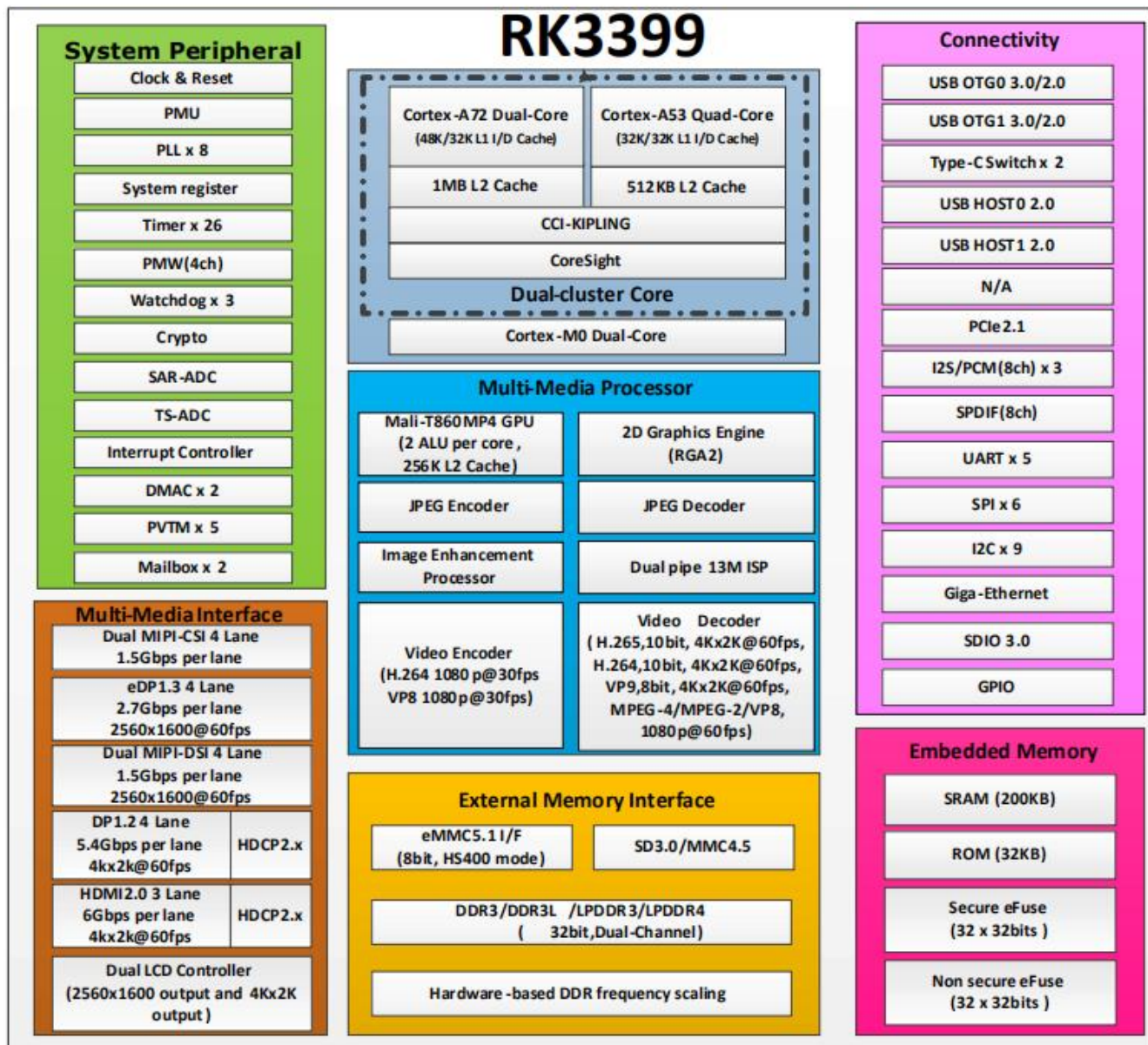
二、SW7990E VER2.1 平台简介

SW7990E VER2.1 博时特科技有限公司基于 RK3399 处理芯片开发的集参考设计、软硬件调试和测试、功能验证一体的硬件方案，用于给客户展示 SW7990E VER2.1 强项目大的多媒体接口和丰富的外围接口，同时为客户提供基于 SW7990E VER2.1 的硬件参考设计和二次开发定制，使客户不需修改或者只需要简单修改参考设计的模块电路，就可以完成新产品的硬件定制和软件开发开发。SW7990E VER2.1 基于 Android7.1 的标准 SDK，支持应用软件的开发、调试和运行等。

2.1 RK3399 主控芯片介绍

SW7990E VER2.1 的主控芯片是 RK3399，其 CPU 采用 big.LITTLE 大小核架构，双 Cortex-A72 大核+四 Cortex-A53 小核结构，对整数、浮点、内存等作了大幅优化，在整体性能、功耗及核心面积三个方面都具革命性提升。RK3399 的 GPU 采用四核 ARM 新一代高端图像处理器 Mali-T864，集成更多带宽压缩技术：如智能送加、ASTC、本地像素存储等，还支持更多的图形和计算接口，总体性能比上一代提升 45%。

芯片架构框图参考图 2-1



2.2 RK3399 芯片功能

2.2.1 CPU

- Big.LITTLE 大小核架构：双 Cortex-A72 大核+四 Cortex-A53
- 64 位高性能 CPU

- 内置低功耗 MCU Cortex-M0

2.2.2 GPU

- 四核 ARM Mali-T860MP4 高性能 GPU
- 支持 OpenGL ES1.1/2.0/3.0/3.1、OpenVG1.1、OpenCL、DX
- 支持 AFBC(帧缓冲压缩)

2.2.3 存储

- 双通道 DDR3/DDR3L/LPDDR3/LPDDR4
- 支持 eMMC 5.1, SDIO 3.0

2.2.4 多媒体

- 支持 4K VP9 and 4K 10bits H265/H264 视频解码, 高达 60
- 1080P 多格式视频解码 (WMV、MPEG-1/2/4、VP8)
- 1080P 视频编码, 支持 H.264, VP8 格式
- 视频后期处理器: 反交错、去噪、边缘/细节/色彩优化

2.2.5 显示

- 双 VOP: 分辨率分别支持 4096x2160 AFBC 及 2560x1600
- 支持双通道 MIPI-DSI (每通道 4 线)
- 显示支持: eDP 1.3 (4 线, 5.4Gbps)
- HDMI 2.0 支持 4K 60fps 显示, 支持 HDCP 1.4/2.2
- 支持 DisplayPort 1.2 (4 线, 最高支持 4K 60Hz)
- 支持 Rec.2020 及 Rec.709

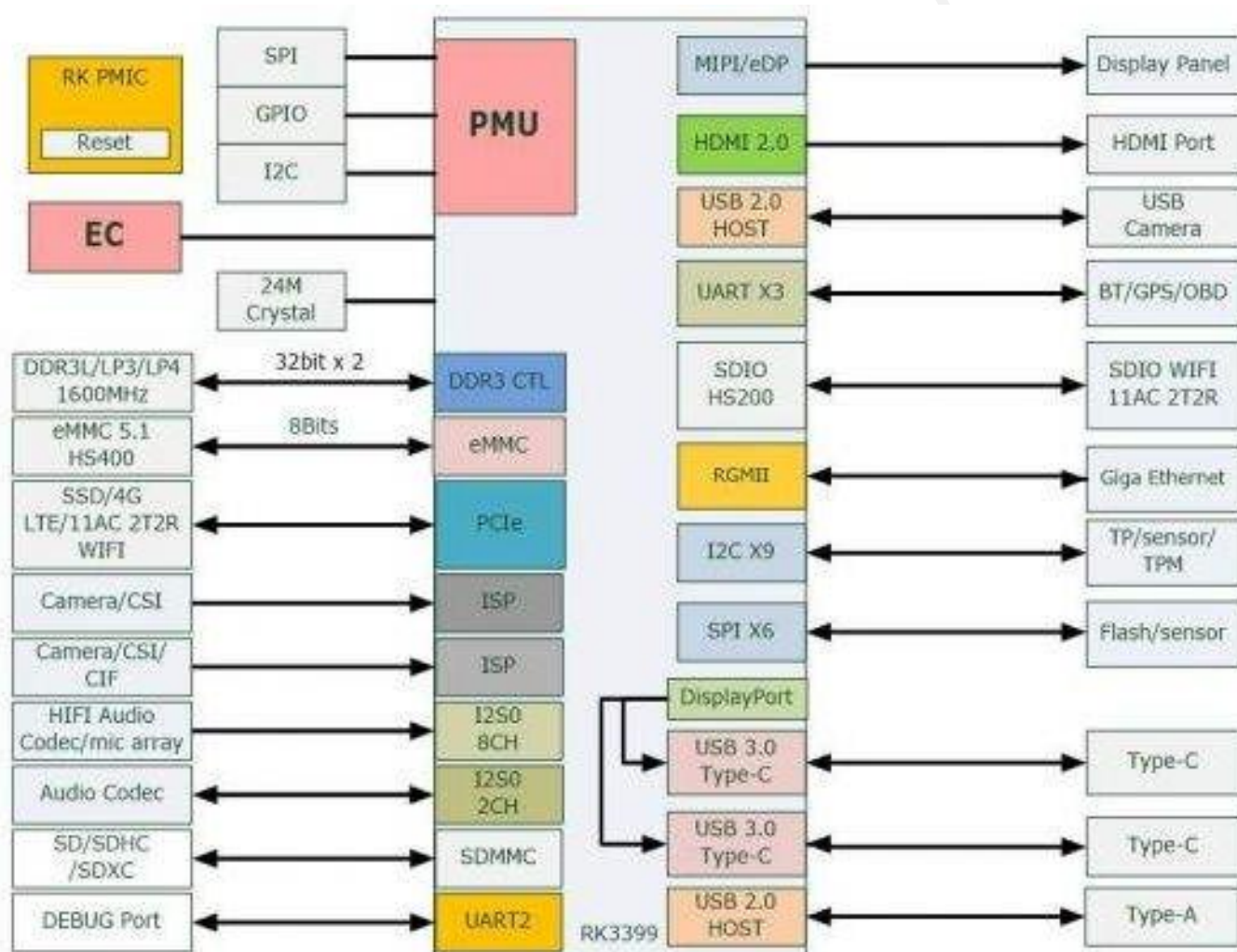
2.2.6 摄像头

双 ISP 像素处理能力高达 13MPix/s，支持双路摄像头数据同

2.3.7 外部接口

支持双 USB 2.0 OTG 以及双 USB 2.0 HOST 接口

2.3 SW7990E VER2.1 平台系统框图



2.4 SW7990E VER2.1 台功能概述

SW7990E VER2.1 包含的功能如下：

- RK808-D电源路径管理系统
- 64bit DDR4，总容量4GByte
- 8bit eMMC，总容量8GByte
- USB OTG：系统升级使用，可以支持Host/Device切换
- USB HOST：USB 2.0接口*2 支持各种USB设备
- 系统按键：Power、Maskrom
- SDIO Wifi：2.4G
- Audio out：支持耳机、扬声器
- Audio in：支持录音
- RMII:支持千兆网络
- Uart Debug：开发板Debug使用
- 显示触摸接口：EDP、DP接口，双TP接口，双Mipi接口

2.5 SW7990E VER2.1 平台规格

主要硬件指标	
板卡配置	Dual-core Cortex-A72 up to 1.8GHz
	Quad-core Cortex-A53 up to 1.4GHz
	Mali-T864 GPU RK3399 瑞芯微电子
解码分辨率	支持几乎全格式的 H.264 1080p@60fps 解码，支持 H.265 1080p@60fps 解码，也支持 h.264

	1080p@30fps 编码，以及高品质的 JPEG 的编/解码。
系统功能	
操作系统	Android 7.1.2
APP 功能	依据客户需求而定。
多媒体支持	
媒体格式	支持 MPEG1、MPEG2、MPEG4、H.264、H.265, WMV、MKV、TS、flv 等主流视频格式；支持 MP3 等音频格式；支持 JPG、JPEG、BMP、PNG、GIF 等图片格式。
基本接口	
电源接口	× 1
以太网接口	× 1, 10/100M/1000M 以太网
USB OTG 接口	X 1, MICRO_USB_5PIN
USB 2.0 接口	USB HOST×2
打印串口	打印串口×1 (UART2)
I2C 触摸	× 2
耳机接口	× 1
扬声器接口	× 1
EDP 接口	× 1
DP 接口	× 1
Mipi 接口	× 2
HDMI 接口	× 1

SIM 卡座	NANO 卡 × 1
ADC 接口	× 1
附加功能	
RTC 时钟	实时时钟电路
看门狗复位	采用内部看门狗，异常自动恢复
选配功能	
存储	默认 4GB+8GB
显示	单屏单触
喇叭	支持 2 组
音频输出	3.5MM 耳机口
G-SENSOR	重力感应功能
内存卡扩展	支持 SDMMC 2.0/3.0，支持 64G
其他功能扩展	支持 USB/RS232/UART 协议通讯的功能模块（需软件调试）
电气指标	
输入电源	12V/3A 直流电源输入(可选配，依功放喇叭功率、显示屏与背光而定)

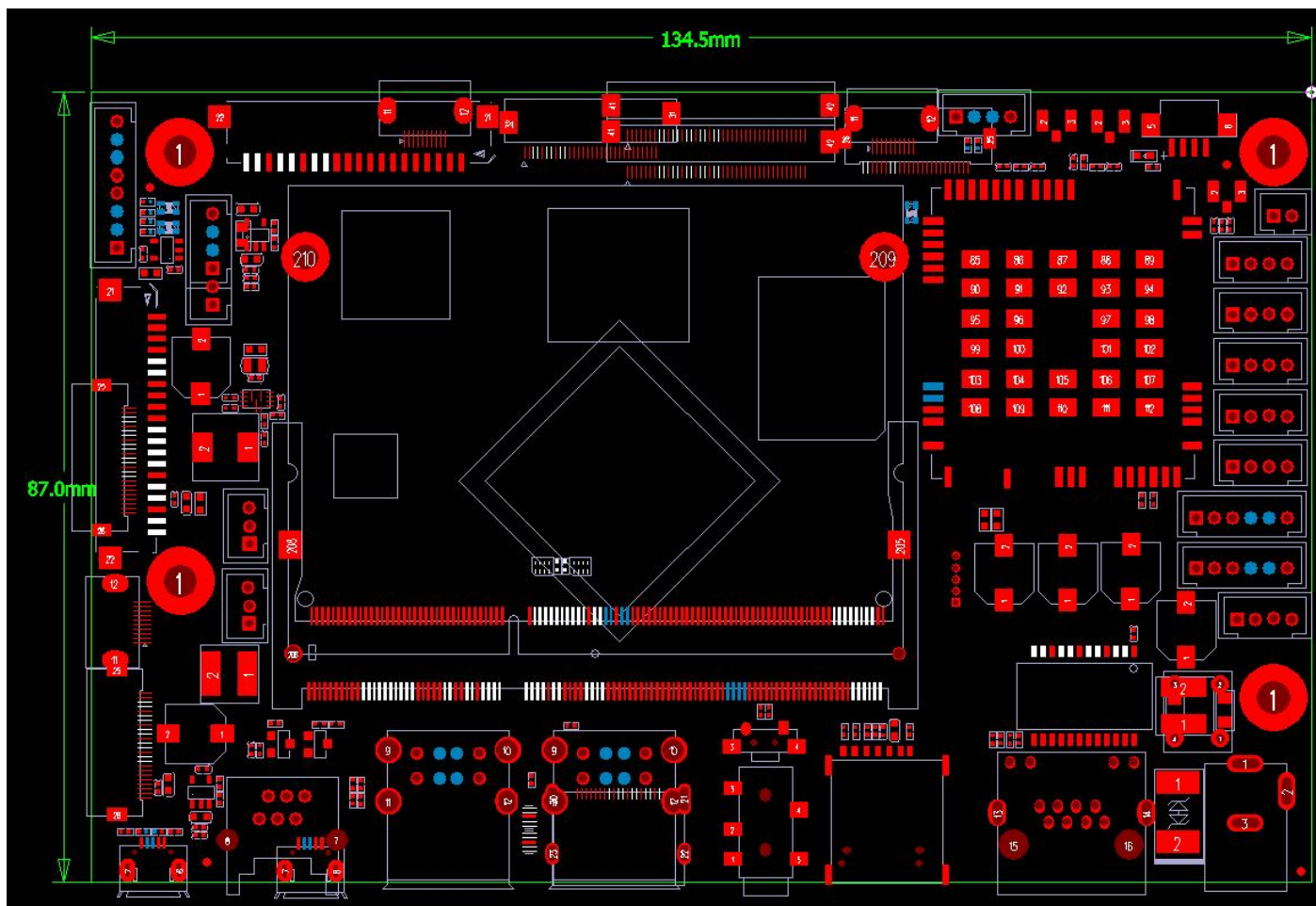
2.6 PF1331 包装组件

PF1331 主要包括以下物品：

- PF1331整机
- 电源适配器，规格：输入 100V AC~240V AC，50Hz；输出 12V DC，3.0A

三、SW7990E VER2.1 硬件尺寸与接口说明

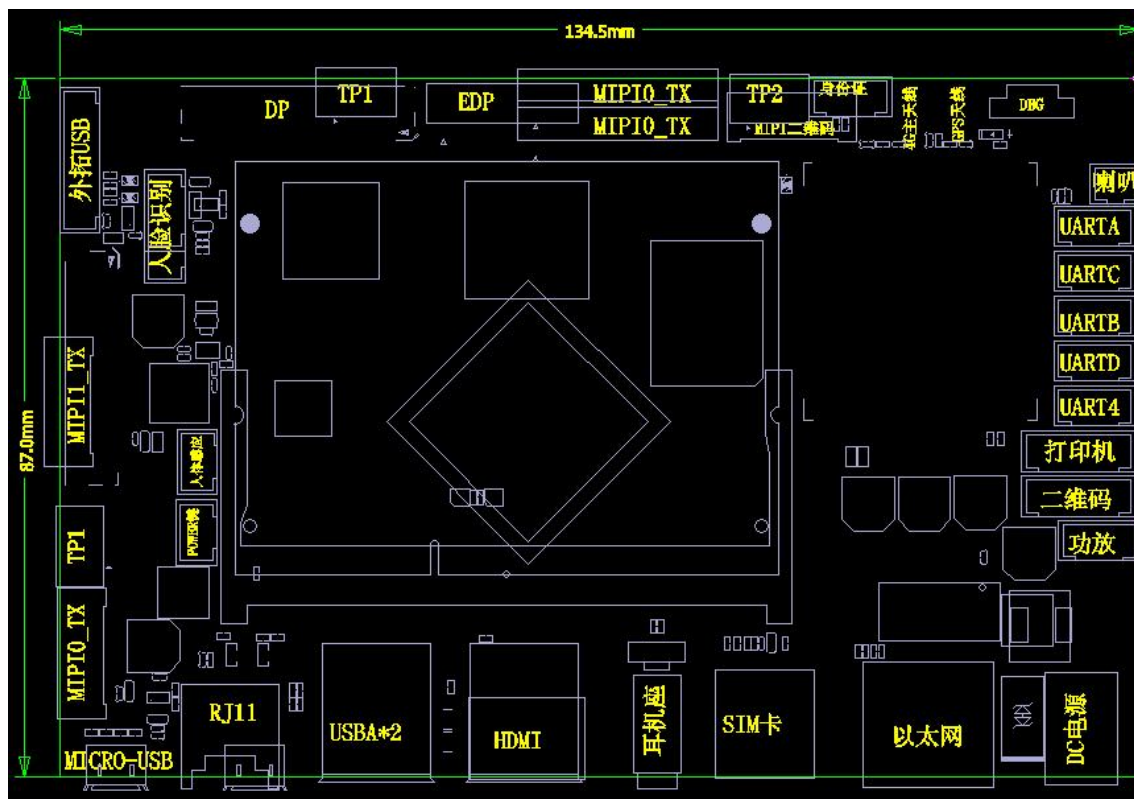
3.1 PCBA 尺寸



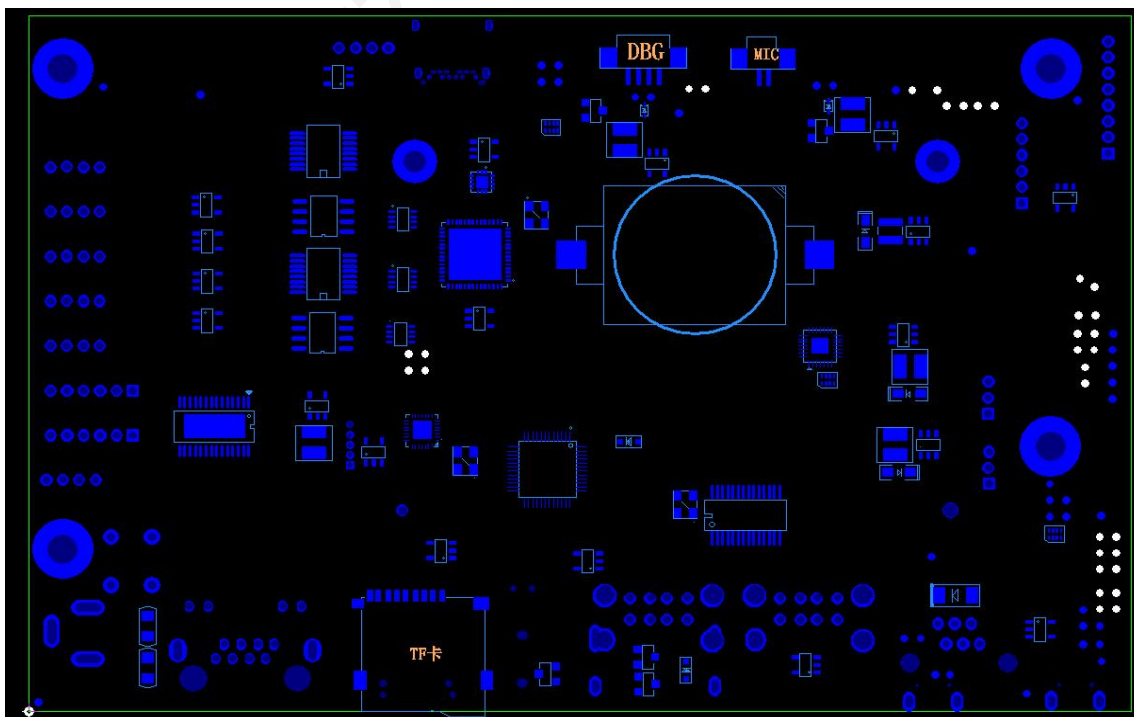
PCB 相关尺寸及规格:长 134.5mm、宽 87.0mm、厚 1.6MM,

螺丝孔规格: 直径 3.0MM。

3.2.TOP 主要硬件接口说明



3.2.BOTTOM 主要硬件接口说明



3.3.主要硬件接口定义说明

J9(DC-IN): POWER 电源接口

序号	定义	描述
1	POWER	电源输入
2	GND	地
3	GND	地

J39(RJ45-屏蔽带灯): 以太网接口

序号	定义	描述
1	MDI0+	发送数据+
2	MDI0-	发送数据-
3	MDI1+	接收数据+
4	MDI2+	数据+
5	MDI2-	数据-
6	MDI1-	数据-
7	MDI3+	数据+
8	MDI3-	数据-
9	LED0	指示灯
10	VCC3V3	3.3V电源
11	LED1	指示灯
12	VCC3	3.3V电源
13	NC	NC

14	NC	NC
----	----	----

J7(SIM_NANO): 4G 卡座

序号	定义	描述
1	SIM-VCC	SIM卡电源
2	SIM-RST	SIM卡复位
3	SIM-CLK1	SIM卡时钟
4	SIM_DET	SIM卡检测
5	GND	地
6	SIM-VCC	VPP
7	SIM-DATA1	SIM卡数据

K1/J17 (默认烧录模式): 烧录模式/3.5MM 耳机口

序号	定义	描述
按键	按键按下	进入烧录模式
1	DET	耳机检测
2	HP_R	音频输出右声道
3	HP_L	音频输出左声道
4	GND	地
5	MIC_IN	声音输入

J18(双层 USB-A 母座): USB 接口

序号	定义	描述
1	FE1.1_VCC2	USB电源

2	FE1.1_DM1	USB1 D-
3	FE1.1_DP1	USB1 D+
4	GND	地
5	FE1.1_VCC2	USB电源
6	FE1.1_DM2	USB2 D-
7	FE1.1_DP2	USB2 D+
8	GND	地

J8(双层 USB-A 母座): USB 接口

序号	定义	描述
1	HUB2.0_VC C2	USB电源
2	FE2.1_DM5	USB1 D-
3	FE2.1_DP5	USB1 D+
4	GND	地
5	HUB2.0_VC C2	USB电源
6	FE2.1_DM6	USB2 D-
7	FE2.1_DP6	USB2 D+
8	GND	地

J28(HDMI 母座): HDMI 接口

序号	定义	描述
1	HDMI_TX2P	HDMI信号
2	GND	地
3	HDMI_TX2N	HDMI信号
4	HDMI_TX1P	HDMI信号
5	GND	地
6	HDMI_TX1N	HDMI信号
7	HDMI_TX0P	HDMI信号
8	GND	地
9	HDMI_TX0N	HDMI信号
10	HDMI_TXCP	HDMI信号
11	GND	地
12	HDMI_TXCN	HDMI信号
13	NC	
14	NC	
15	HDMI_SCL	SCL信号
16	HDMI_SDA	SDA信号
17	GND	地
18	5V	5V输出
19	HDMI0-HPD	HDMI0-HPD

J35 (RJ11): RJ11 接口

序号	定义	描述
1		
2	GND	地
3		
4	RS232_TX	RS232_TX
5	RS232_TX	RS232_RX
6	GND	地

J23 (MIRCO-USB-5P): USB OTG 接口

序号	定义	描述
1	VCC	USB电源
2	TYPEC0_DM	USB OTG D-
3	TYPEC0_DP	USB OTG D+
4	OTG_ID	OTG检测
5	GND	地

JX1(24PIN/0.5MM): MIPI 接口

序号	定义	描述
1	NC	空脚
2	NC	空脚
3	MIPI_D0P	MIPI 数据
4	MIPI_D0N	MIPI 数据

5	GND	接地
6	MIPI_D1P	MIPI 数据
7	MIPI_D1N	MIPI 数据
8	GND	接地
9	MIPI_CLKP	MIPI 时钟
10	MIPI_CLKN	MIPI 时钟
11	GND	接地
12	MIPI_D2P	MIPI 数据
13	MIPI_D2N	MIPI 数据
14	GND	接地
15	MIPI_D3P	MIPI 数据
16	MIPI_D3N	MIPI 数据
17	GND	接地
18	RST	复位
19	GND	接地
20	VCC3.3	屏电源，3.3V
21	VDD1.8	IO电源，1.8V
22	GND	接地
23	LED-	背光负极
24	LED+	背光正极

J3 (10PIN/0.5MM) : TP1 接口 (I2C)

序号	定义	描述
1	GND	接地
2	GND	接地
3	RST	触摸屏复位信号
4	INT	触摸屏中断信号
5	GND	接地
6	I2C7_SCL_T P	触摸屏I2C时钟信号
7	I2C7_SDA_T P	触摸屏I2C数据信号
8	VCC_TP	触摸屏供电电源
9	GND	接地
10	GND	接地

J2(DF14_1.25MM 端子): MIPI 接口

序号	定义	描述
1	RST	复位
2	LED-	背光负极
3	LED+	背光正极
4	VDD1.8	IO电源, 1.8V
5	MIPI_TX1_RX1_D0P	MIPI 数据
6	MIPI_TX1_RX1_D0N	MIPI 数据

7	TP2-RST	触摸屏复位信号
8	TP2-INT	触摸屏中断信号
9	I2C6_SCL_TP	触摸屏I2C时钟信号
10	I2C6_SDA_TP	触摸屏I2C数据信号
11	MIPI_TX1_RX1_D1P	MIPI 数据
12	MIPI_TX1_RX1_D1N	MIPI 数据
13	MIPI_TX1_RX1_CLK P	MIPI 时钟
14	MIPI_TX1_RX1_CLK N	MIPI 时钟
15	VCC3.3	屏电源，3.3V
16	MIPI_TX1_RX1_D2P	MIPI 数据
17	MIPI_TX1_RX1_D2N	MIPI 数据
18	GND	接地
19	MIPI_TX1_RX1_D3P	MIPI 数据
20	MIPI_TX1_RX1_D3N	MIPI 数据

J20 (8PIN/2.0MM)：外扩 USB_HOST

序号	定义	描述
1	VBUS	V-BUS
2	FE2.1_DM3	USB数据信号负极
3	FE2.1_DP3	USB数据信号正极

4	GND	接地
5	VBUS	V-BUS
6	FE2.1_DM4	USB数据信号负极
7	FE2.1_DP4	USB数据信号正极
8	GND	接地

JP1(3PIN/2.0MM): POWER 按键

序号	定义	描述
1	POWER_KEY	POWER按键
2	VCC_SYS_5V	灯电源
3	GND	接地

JP1(3PIN/2.0MM): 人体感应

序号	定义	描述
1	GPIO	人体感应数据接收口
2	GND	接地
3	VCC	VCC_3.3V

J31 (6PIN/2.0MM) : 高拍仪及补光灯

序号	定义	描述
1	VCC	补光灯电源
2	GND	接地
3	VBUS	USB 5V
4	TYPEC1_DM	USB数据信号负极

5	TYPEC1_DP	USB数据信号正极
6	GND	接地

J13(DF14_1.25MM 端子): MIPI 接口

序号	定义	描述
1	BL_PWR	背光电源输出
2	BL_PWR	背光电源输出
3	BL_PWM_DIM	背光亮度控制信号
4	ON_OFF	背光使能控制信号
5	GND	接地
6	GND	接地
7	TP2-RST	触摸屏复位信号
8	TP2-INT	触摸屏中断信号
9	I2C6_SCL_TP	触摸屏I2C时钟信号
10	I2C6_SDA_TP	触摸屏I2C数据信号
11	LCD_VCC	LCD屏电源输出
12	LCD_VCC	LCD屏电源输出
13	AUX_CH_N	DP辅助通道信号
14	AUX_CH_P	DP辅助通道信号
15	GND	接地
16	DP_TX0P	DP数据信号
17	DP_TX0N	DP数据信号

18	GND	接地
19	DP_TX1P	DP数据信号
20	DP_TX1N	DP数据信号

J5(30P_0.5MM): EDP 屏接口

序号	定义	描述
1	NC	NC
2	GND	地
3	EDP_TX1N	EDP 信号输入
4	EDP_TX1P	EDP 信号输入
5	GND	地
6	EDP_TX0N	EDP 信号输入
7	EDP_TX0P	EDP 信号输入
8	GND	地
9	EDP_AUXP	EDP 信号输入
10	EDP_AUXN	EDP 信号输入
11	GND	地
12	EDP_3.3V	3.3V供电
13	EDP_3.3V	3.3V供电
14	NC	NC
15	GND	地
16	GND	地

17	NC	HPD
18	GND	背光负极
19	GND	背光负极
20	GND	背光负极
21	GND	背光负极
22	BL-EN0	背光使能
23	PWM-0	背光调节
24	NC	NC
25	NC	NC
26	12V	背光正极
27	12V	背光正极
28	12V	背光正极
29	12V	背光正极
30	NC	NC

J4 (10PIN/0.5MM) : TP2 接口 (I2C)

序号	定义	描述
1	GND	接地
2	GND	接地
3	RST	触摸屏复位信号
4	INT	触摸屏中断信号
5	GND	接地

6	I2C6_SCL_T P	触摸屏I2C时钟信号
7	I2C6_SDA_T P	触摸屏I2C数据信号
8	VCC_TP	触摸屏供电电源
9	GND	接地
10	GND	接地

J19 (4PIN/2.0MM) : USB_HOST(身份证)

序号	定义	描述
1	VBUS	V-BUS
2	FE1.1_DM4	USB数据信号负极
3	FE1.1_DP4	USB数据信号正极
4	GND	接地

J22 (4PIN/2.0MM) : UARTA RS232 485

序号	定义	描述
1	VCC	3.3V/5V电源输出
2	UART_TXA	TTL3.3V/TTL5V接口数据发送
3	UART_RXA	TTL3.3V/TTL5V接口数据接收
4	GND	接地

J25 (4PIN/2.0MM) : UARTC RS232

序号	定义	描述
1	VCC	3.3V/5V电源输出
2	UART_TXC	TTL3.3V接口数据发送
3	UART_RXC	TTL3.3V接口数据接收
4	GND	接地

J24 (4PIN/2.0MM) : UARTB RS232

序号	定义	描述
1	VCC	3.3V/5V电源输出
2	UART_TXB	TTL3.3V接口数据发送
3	UART_RXB	TTL3.3V接口数据接收
4	GND	接地

J26 (4PIN/2.0MM) : UARTD

序号	定义	描述
1	VCC	3.3V/5V电源输出
2	UART_TXD	TTL3.3V/TTL5V接口数据发送
3	UART_RXD	TTL3.3V/TTL5V接口数据接收
4	GND	接地

J20 (4PIN/2.0MM) : UART4 RS232 485

序号	定义	描述
1	VCC	3.3V/5V电源输出

2	UART_TXA	TTL3.3V/TTL5V接口数据发送
3	UART_RXA	TTL3.3V/TTL5V接口数据接收
4	GND	接地

J10 (6PIN/2.0MM) : 打印机

序号	定义	描述
1	VCC_12V	打印机电源12V
2	GND	接地
3	VBUS	V-BUS
4	FE2.1_DM2	USB数据信号负极
5	FE2.1_DP2	USB数据信号正极
6	GND	接地

J21 (6PIN/2.0MM) : 二维码及补光灯/指示灯

序号	定义	描述
1	VCC	补光灯电源
2	GND	接地
3	VBUS	V-BUS/指示灯电源
4	FE2.1_DM1	USB数据信号负极
5	FE2.1_DP1	USB数据信号正极
6	GND	接地

J32(2P_2.0MM): 扬声器接口双通道

序号	定义	描述
----	----	----

1	LP	左声道+/喇叭+ 1
2	LN	左声道-/喇叭- 2
3	RN	右声道-/喇叭-
4	RP	右声道+/喇叭+

J1(4P_1.25MM): 调试串口接口 UART2

序号	定义	描述
1	VCC3V3_S5	3.3V电源
2	UART2DBG_ TX	串口数据发送
3	UART2DBG_ RX	串口数据接收
4	GND	地

J1(2_1.25MM): MIC 接口

序号	定义	描述
1	MIC_IN1P	MIC数据信号负极
2	MIC_IN1N	MIC数据信号正极

第四章 7990E VER2.1 PCBA 开发板的使用

4.1 开关机

开机和关机方法介绍如下：

1、开机方法：

(1) 使用 DC 12V 供电，打开电源总开关，即可开机。后续开关机可以通过 PWRON 按钮实现；

2、关机方法：

长按 Power 键 2 秒，在显示屏窗口界面点击关机。

3、异常关机方法：

使用 DC 12V 供电，异常情况下，可以通过关闭船型开关电源来关闭开发板电源。

4、异常重启方法：

使用 DC 12V 供电，异常情况下，长按 Power 键 10 秒，系统自动重启。

4.2 USB 驱动安装

在固件烧写、驱动升级以及 ADB 连接前需要先安装 USB 驱动程序，请与业务获取驱动包，驱动版本为 DriverAssitant_v4.5

打开“DriverInstall.exe”，点击“驱动安装”，提示“安装驱动成功”即可。如果已安装旧驱动，请点击“驱动卸载”，并重新安装驱动。

驱动文件目前仅支持 Windows。

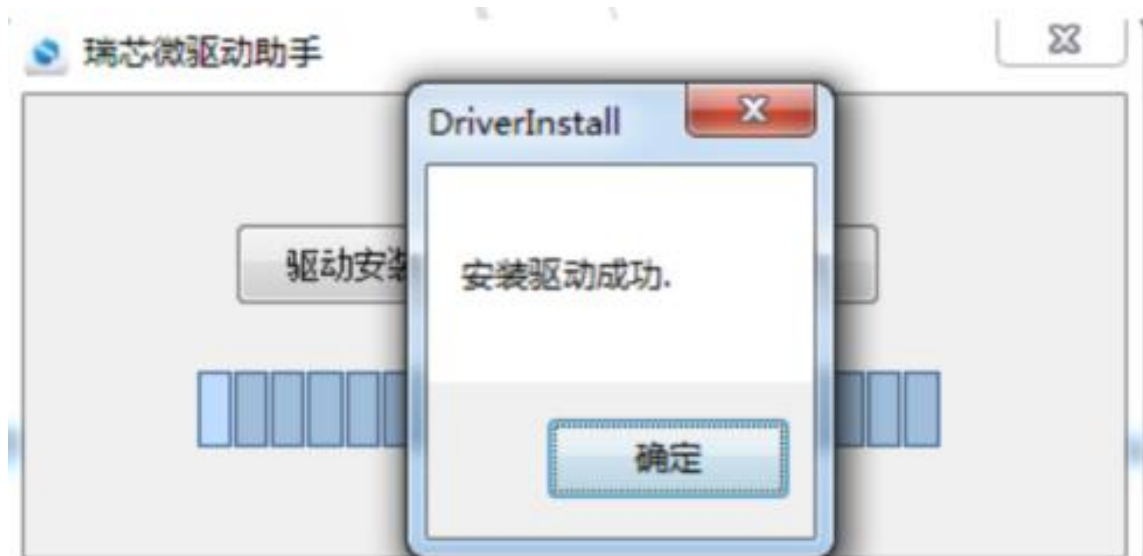


图 4-2-1 驱动安装成功示意图

4.3 固件烧写方法

SW7990 有两种固件烧写方式：

4.3.1 Maskrom 烧写模式

基本原理是在系统上电前将 FLASH_D0 对地短路，使 Flash 引导失败，从而进入 Maskrom 状态。适用于烧写了错误的 bootloader 文件，无法正常引导系统开机的情况下。

具体步骤如下：

1. 连接 USB 到电脑 PC 端，并按住开发板的 Maskrom 按键不放；
2. 给 7990E 供电 12V；
3. 等待会儿开发工具将显示“发现一个 Maskrom 设备”
4. 开发工具选择对应的 image 文件。

5. 点击执行，即进入升级状态，在工具的右侧有进度显示栏，显示下载与校验情况。



图4-3-1 进入Maskrom烧写模式工具上示意图

4.3.2 Loader 烧写模式

基本原理在系统上电或重启前保证 ADC2_KEY_IN 是低电平，上电或重启后系统将进入 Loader 状态。适用于正常情况下，更换固件中的一小部分或者全部。

具体步骤如下：

1. 设备正常启动后，连接 USB 到电脑 PC 端。
2. 此时打开开发工具，会提示发现 ADB 设备，点击切换按钮。
3. 等待会儿开发工具将显示“发现一个 Loader 设备”。
4. 开发工具选择对应的 image 文件。
5. 点击执行，即进入升级状态，在工具的右侧有进度显示栏，显示下载与校验情况。

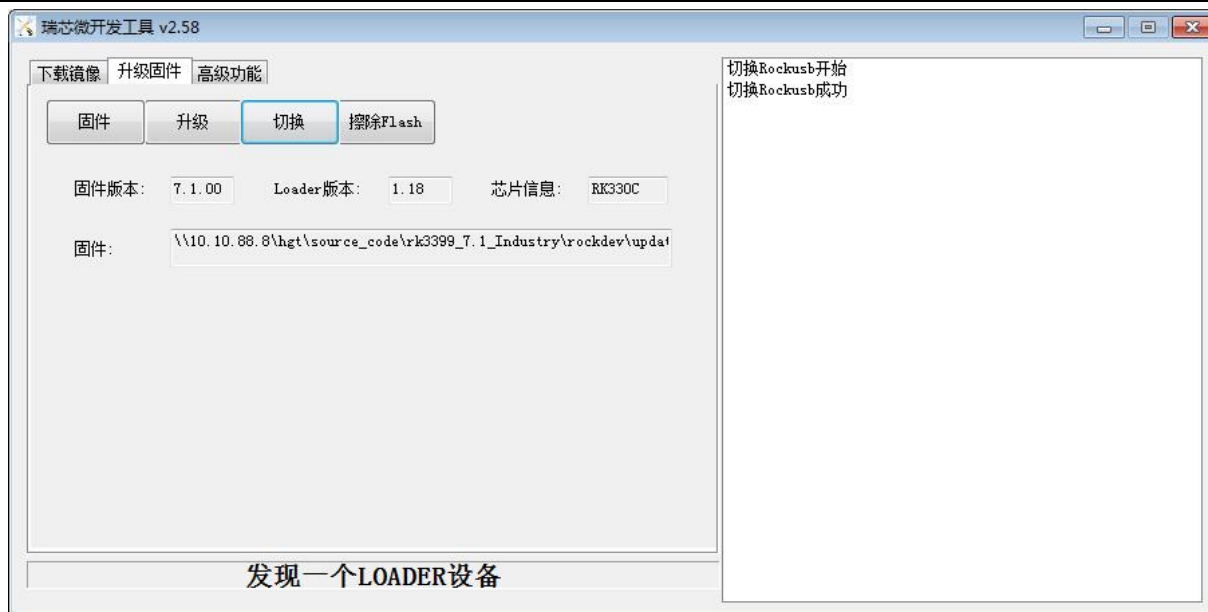


图4-3-2 进入Loader烧写模式工具上示意图

4.4 串口调试

4.4.1 连接串口

需要使用串口工具连接到电脑 PC 端，在 PC 端设备管理器中得到当前端口的 COM 号，然后接到主板的 serial Debug 口上。

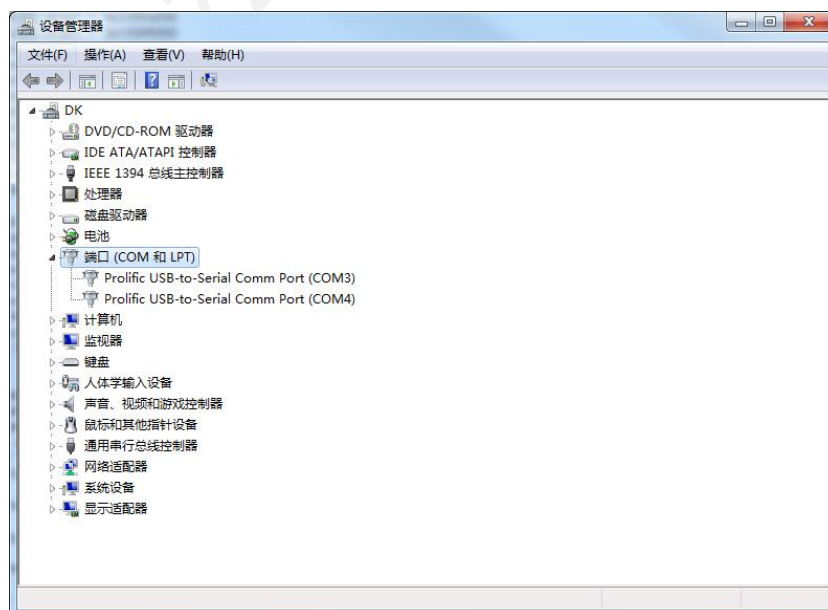


图 4-4-1 获取当前端口COM号

打开串口工具“SecureCRT”，点击“快速连接”按钮。

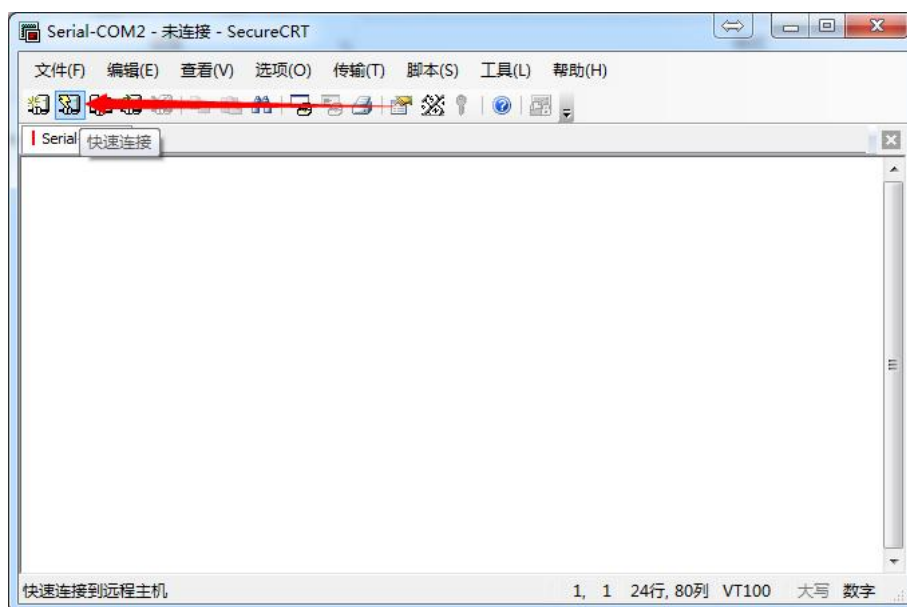


图 4-4-2 串口工具SecureCRT界面

配置串口，如下图所示，端口选择连接开发板的端口号，波特率选择 1.5M（RK3399 默认），流控 RTS/CTS 不需勾选。

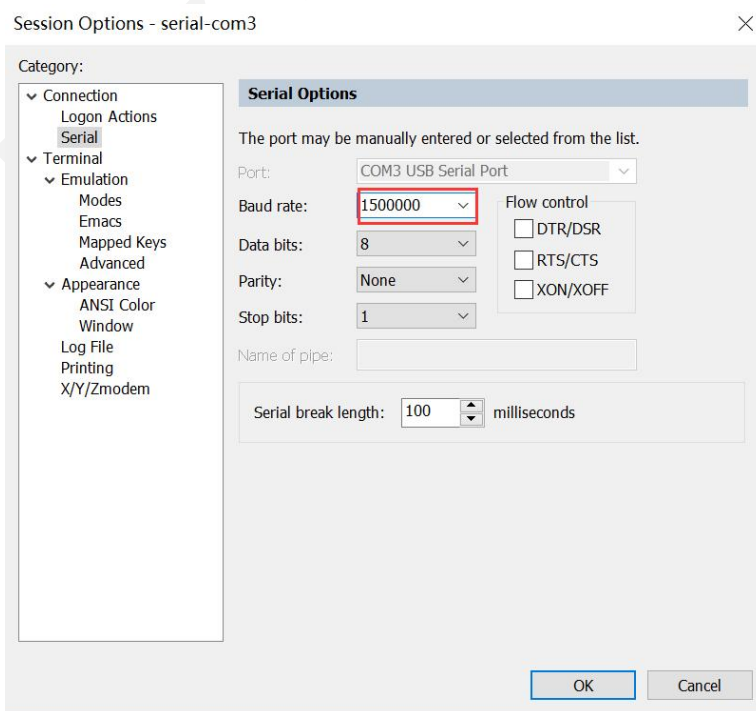


图 4-4-3 配置串口信息

点击连接，就能正常连接设备了。为方便调试，配置会话选项，点击工具栏“会话选项”，回滚缓冲区设置较大数，可以保存更多的 log 信息。

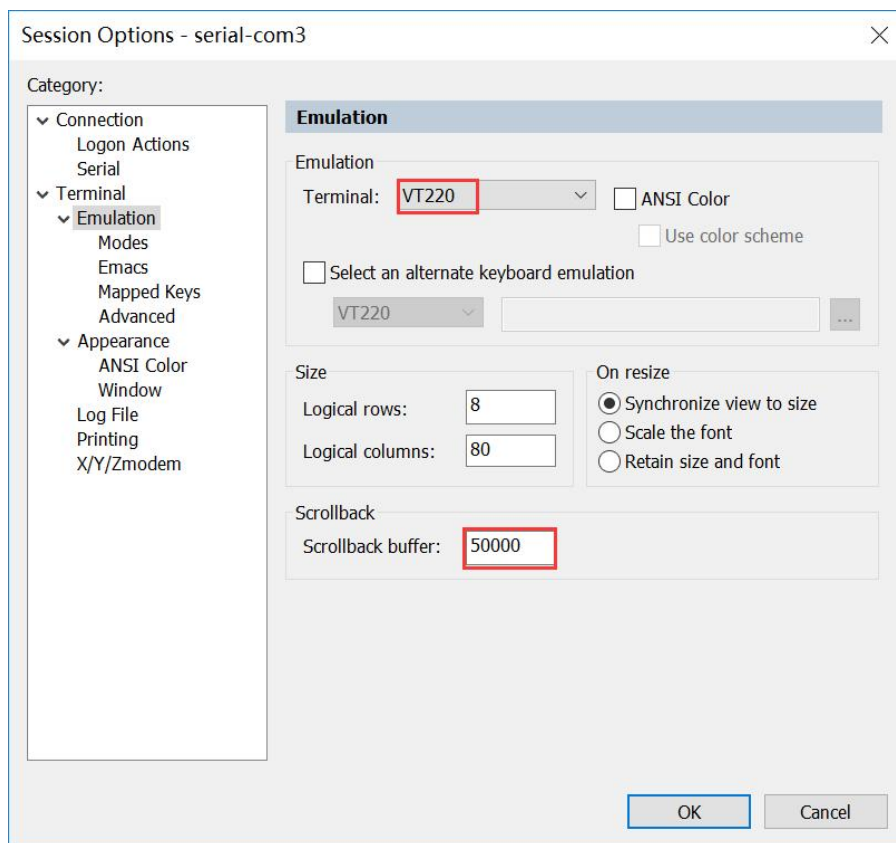


图 4-4-4 配置串口工具选项

4.5 ADB 调试

本节介绍 开发过程中的一些 ADB 调试工具和调试方法，帮助开发者快速进行基础系统调试，并做出正确的分析。

4.5.1 ADB 工具概述

ADB (Android Debug Bridge) 是 Android 调试的一个工具，用这个工具可以操作

管理

Android 模拟器或真实的 Android 设备。主要功能有：

- ① 1.运行设备的 shell（命令行）
- ① 2.管理模拟器或设备的端口映射
- ① 3.计算机和设备之间上传/下载文件
- ① 4.将本地 apk 软件安装至模拟器或 Android 设备

ADB 是一个“客户端—服务器端”程序，其中客户端主要是指 PC，服务器端是 Android 设备的实体机器或者虚拟机。根据 PC 连接设备的方式不同，ADB 可以分为两类：

- ① 网络 ADB：主机通过有线/无线网络（同一局域网）连接到 STB 设备
- ① USB ADB：主机通过 USB 线连接到 STB 设备

4.5.2 USB ADB 使用说明

USB ADB 使用有以下限制：

- ① 1.只支持 USB OTG 口
- ① 2.不支持多个客户端同时使用（如 cmd 窗口，eclipse 等）
- ① 3.只支持主机连接一个设备，不支持连接多个设备

连接步骤如下：

- 1、设备已经运行 Android 系统，设置->开发者选项->已连接到计算机打开，usb 调试开关打开。
- 2、PC 主机只通过 USB 线连接到机器 USB OTG 口，然后电脑通过如下命令与设备相连。

```
adb shell
```

3、测试是否连接成功，运” adb devices” 命令，如果显示机器的序列号，表示连接成功。

4.5.2 网络 ADB 使用要求

ADB 早期版本只能通过 USB 来对设备调试，从 adb v1.0.25 开始，增加了对通过 tcp/ip 调试 Android 设备的功能。

如果你需要使用网络 ADB 来调试设备，必须要满足如下条件：

- 1、设备上面首先要有网口，或者通过 WiFi 连接网络。
- 2、设备和研发机（PC 机）已经接入局域网，并且设备设有局域网的 IP 地址。
- 3、要确保研发机和设备能够相互 ping 得通。
- 4、研发机已经安装了 ADB。
- 5、确保 Android 设备中 adbd 进程（ADB 的后台进程）已经运行。adbd 进程将会监听端口 5555 来进行 ADB 连接调试。

4.5.3 SDK 网络 ADB 端口配置

SDK 修改了对网络 ADB 端口配置 是 5555。

修改了 device/rockchip/rkxxxx/system.prop 文件，添加了如下配置：

```
service.adb.tcp.port=5555
```

4.5.4 网络 ADB 使用

本节假设设备的 IP 为 192.168.1.5, 下文会用这个 IP 建立 ADB 连接, 并调试设备。

1、首先 Android 设备需要先启动, 如果可以的话, 可以确保一下 adbd 启动(ps 命令查看)。

2、在 PC 机的 cmd 中, 输入:

```
adb connect 192.168.1.5:5555
```

如果连接成功会进行相关的提示, 如果失败的话, 可以先 kill-server 命令, 然后重试连接。

```
adb kill-server
```

3、如果连接已经建立, 在研发机中, 可以输入 ADB 相关的命令进行调试了。比如

```
adb shell
```

l, 将会通过 TCP/IP 连接设备上面。和 USB 调试是一样的。

4、调试完成之后, 在研发机上面输入如下的命令断开连接:

```
adb disconnect 192.168.1.5:5555
```

4.5.5 手动修改网络 ADB 端口号

若 SDK 未加入 ADB 端口号配置, 或是想修改 ADB 端口号, 可通过如下方式修改:

1、首先还是正常地通过 USB 连接目标机, 在 windows cmd 下执行 adb shell 进入。

2、设置 ADB 监听端口:

```
#setprop service.adb.tcp.port 5555
```

3、通过 ps 命令查找 adbd 的 pid

4、重启 adbd

```
#kill -9<pid>, 这个 pid 就是上一步找到那个 pid
```

杀死 adbd 之后，Android 的 init 进程后自动重启 adbd。adbd 重启后，发现设置了 service.

adb.tcp.port, 就会自动改为监听网络请求。

4.5.6 ADB 常用命令详解

(1) 查看设备情况

查看连接到计算机的 Android 设备或者模拟器：

```
adb devices
```

返回的结果为连接至开发机的 Android 设备的序列号或是 IP 和端口号 (Port)、状态。

(2) 安装 APK

将指定的 APK 文件安装到设备上：

```
adb install <apk 文件路径>
```

示例如下：

```
adb install "F:\WishTV\WishTV.apk"
```

重新安装应用：

```
adb install -r <apk 文件路径>
```

示例如下：

```
adb install -r "F:\WishTV\WishTV.apk"
```

（3）卸载 APK

完全卸载：

```
adb uninstall <package>
```

示例如下：

```
adb uninstall com.wishtv
```

（4）使用 rm 移除 APK 文件：

```
adb shell rm <filepath>
```

示例如下：

```
adb shell rm "system/app/WishTV.apk"
```

示例说明：移除“system/app”目录下的“WishTV.apk”文件。

（5）进入设备和模拟器的 shell

进入设备或模拟器的 shell 环境：

```
adb shell
```

（6）从电脑上传文件到设备

用 push 命令可以把本机电脑上的任意文件或者文件夹上传到设备。本地路径一般指本机电脑；远程路径一般指 ADB 连接的单板设备。

注意此操作需要 ROOT 权限和 disable-verity、remount 命令搭配使用

adb disable-verity 命令需要最新版本的 ADB，请到官网下载

```
adb push <本地路径><远程路径>
```

示例如下：

```
adb root
```

```
adb disable-verity
```

```
adb reboot
```

此时设备重启后,再继续执行以下命令

```
adb root
```

```
adb remount
```

```
adb push "F:\WishTV\WishTV.apk" "system/app"
```

示例说明：将本地“WishTV.apk”文件上传到 Android 系统的“system/app”目录下。

（7）从设备下载文件到电脑

pull 命令可以把设备上的文件或者文件夹下载到本机电脑中。

```
adb pull <远程路径><本地路径>
```

示例如下：

```
adb pull system/app/Contacts.apk F:\
```

示例说明：将 Android 系统“system/app”目录下的文件或文件夹下载到本地“F:\”目录下。

（8）查看 bug 报告

需要查看系统生成的所有错误消息报告，可以运行 adb bugreport 指令来实现，该指令会将

Android 系统的 dumphsys、dumpstate 与 logcat 信息都显示出来。

（9）查看设备的系统信息

在 adb shell 下查看设备系统信息的命令。

```
adb shell getprop
```

4.6 Log 日志自动保存系统

为了简化客户的操作，让没有 android 开发基础的人抓取出问题时的 log 日志，在 RK3399 上新增一个 log 日志自动保存功能。

4.6.1 使用方法：

(1)在开发者选项中点击 “系统日志收集器” 来开启此功能：（开发者选项 需要连续点击版本号 5 次打开）

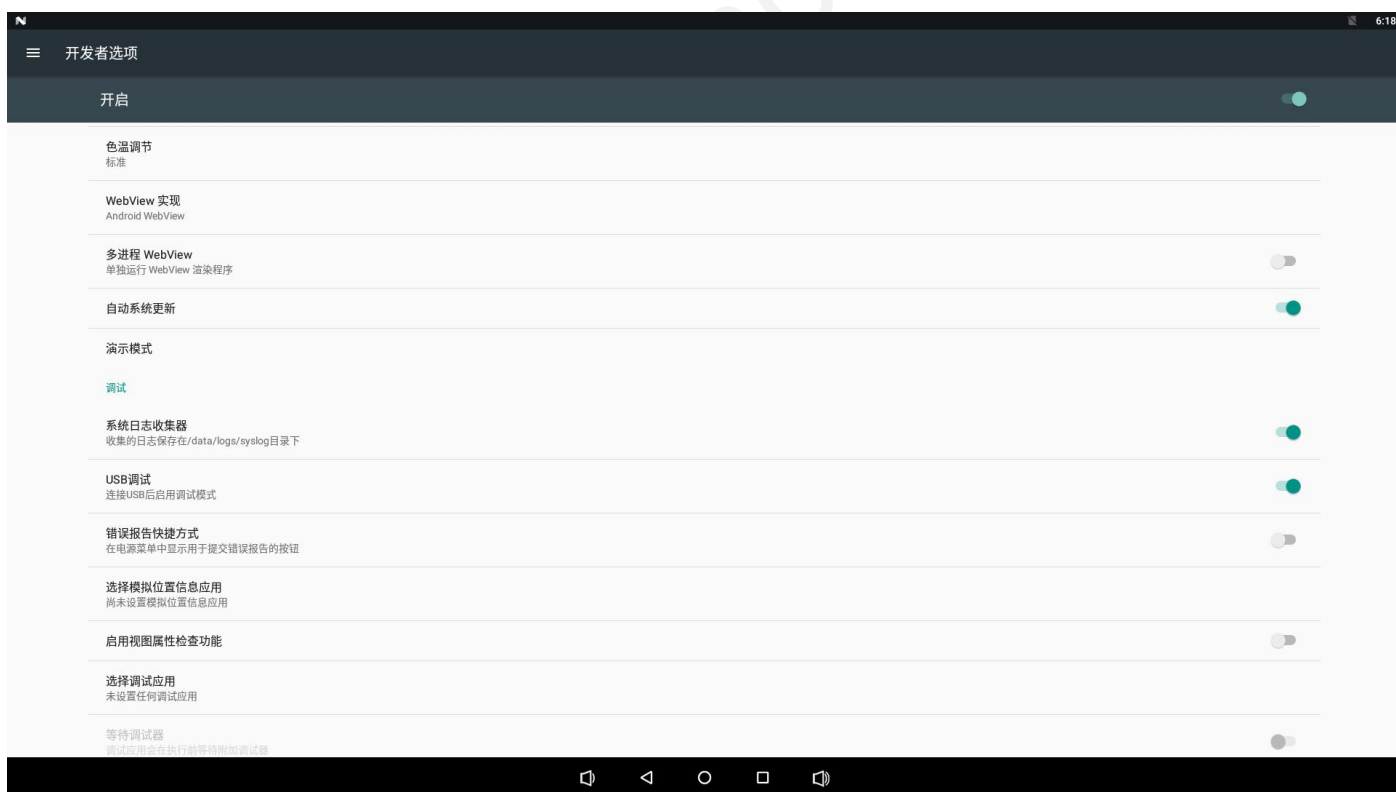


图-打开日志收集器

(2)导出日志到 PC 的使用方法：

重新开机后连接电脑，选择可以看到/data/media/0 目录下有 rk_logs 目录有以

下内容，并且有 COPY-COMPLETE 表示拷贝完成。可以直接通过

```
adb pull /data/media/0/rk_logs
```

导出日志，

或者打开 usb 传输模式，直接拷贝到电脑上，见下图。

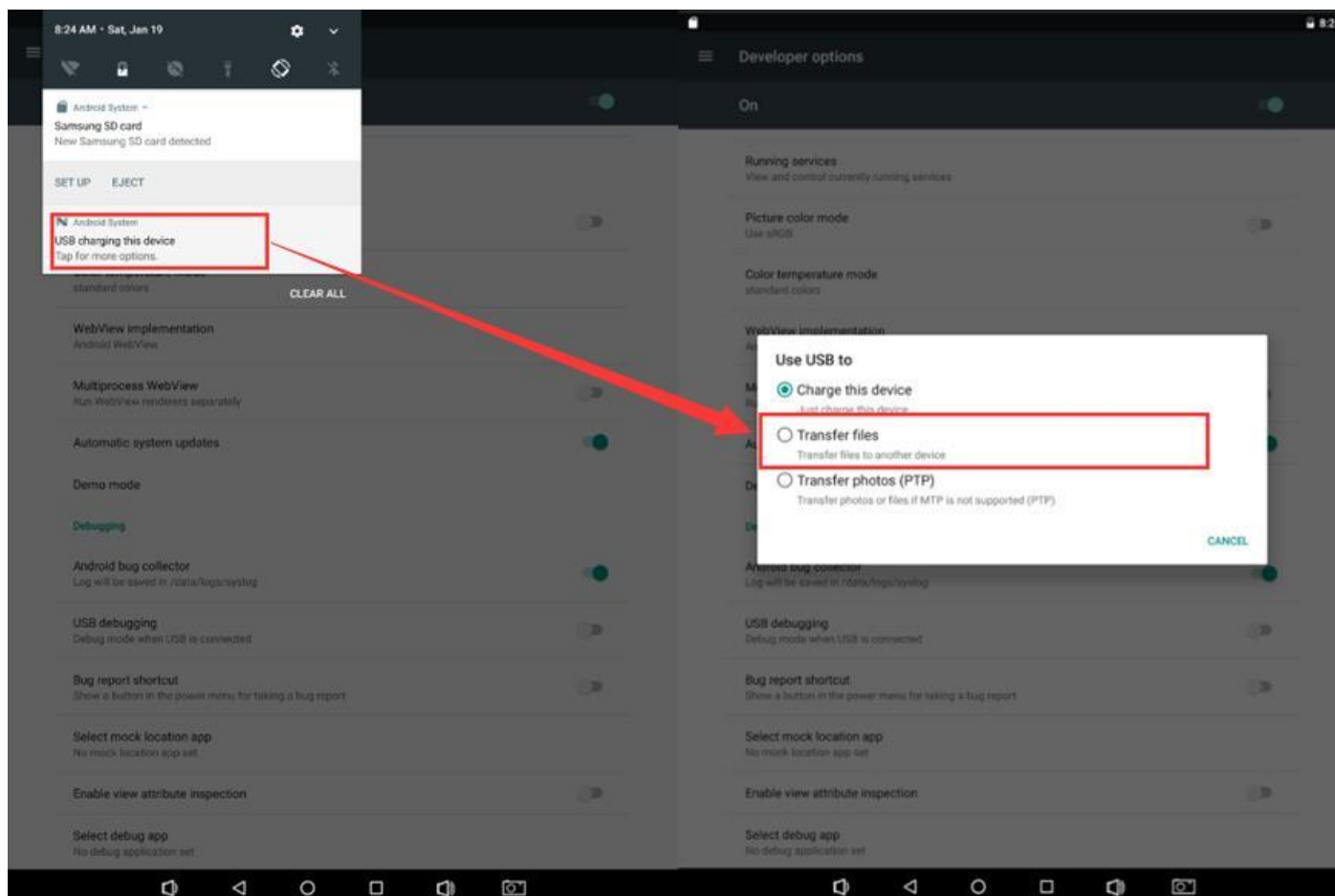
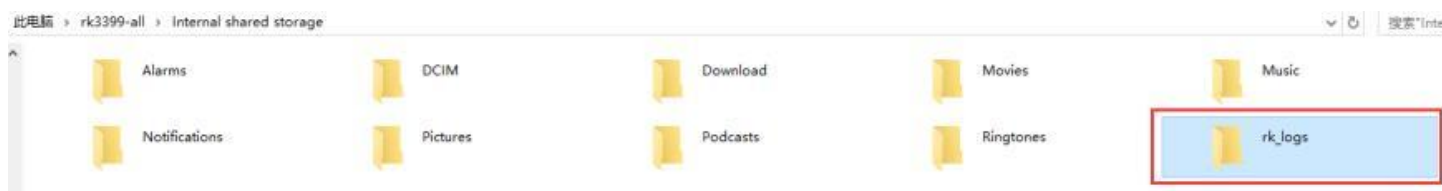


图- 打开 USB 传输文件

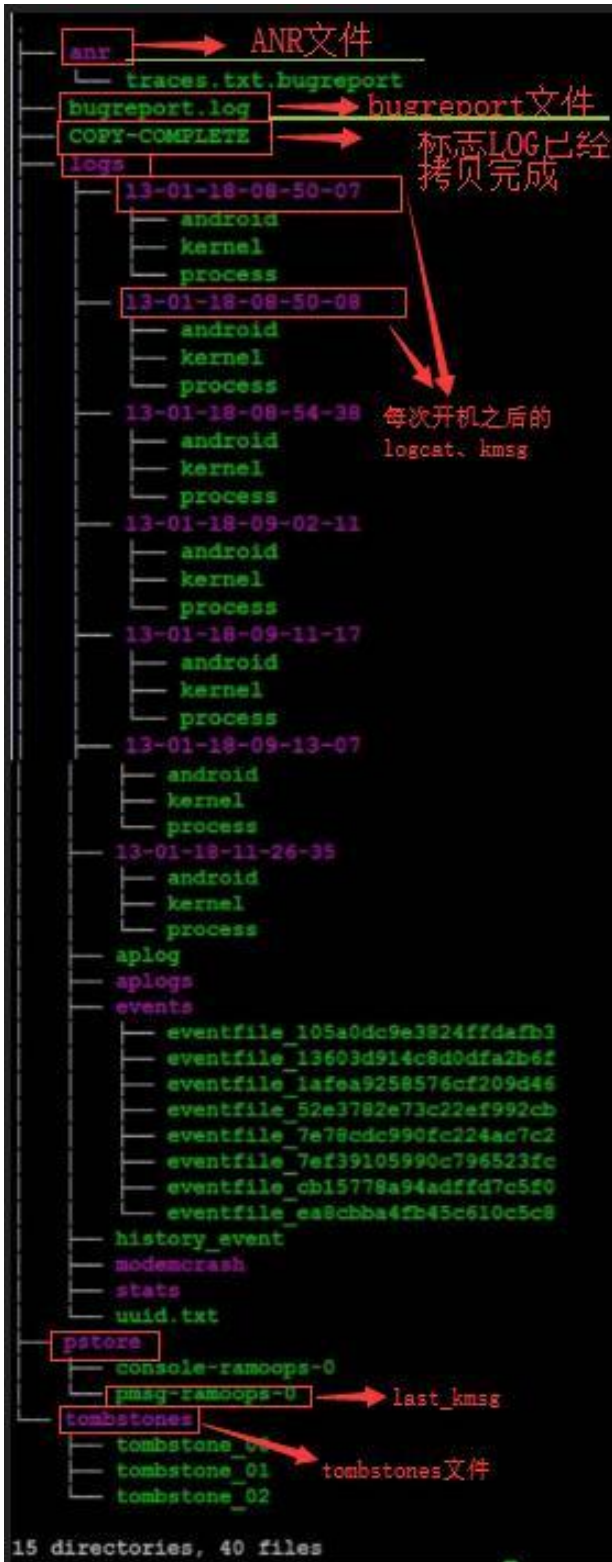
文件夹所在图：



(3)导出 Log 到 SD 卡的使用方法

sd 卡插入后，几秒后(生成 bugreport 需要一些时间)，后可以看到

/storage/8527-18E3/ 目录(该目录每个产品可能各有不同)下有 rk_logs 目录有以下内容, 并且有 COPY-COMPLETE 表示拷贝完成。



4.7 序列号烧录 -WNpctool 工具

本平台使用 WNpctool 工具进行序列号信息的烧写。请与业务申请获取工具包
以下说明该工具的基本用法。

4.7.1 使用 WNpctool 写入序列号

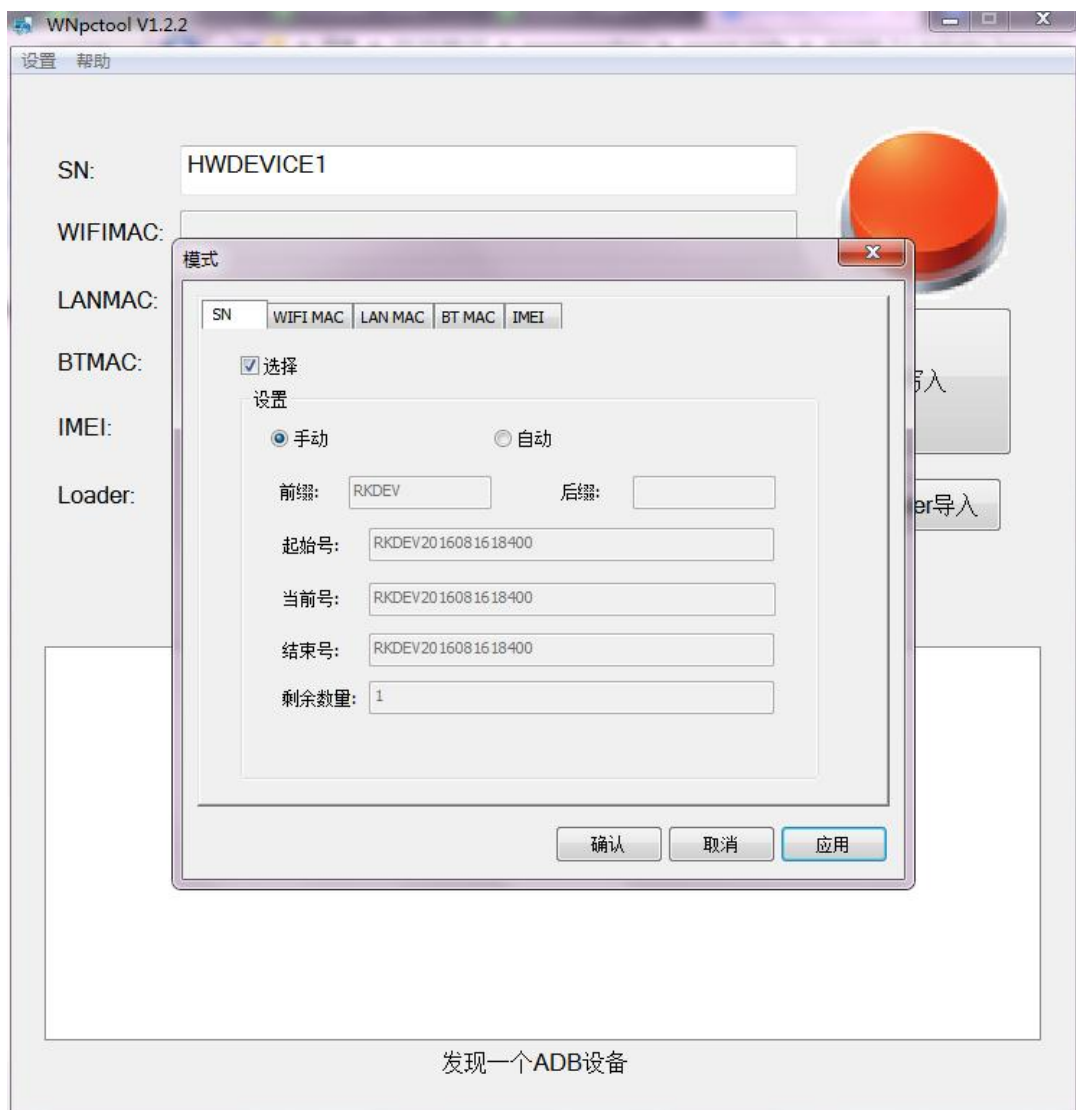


图 WNpctool 工具

①

1. 点击“设置”菜单，下拉框中取消勾选“写后重启”。
2. 点击“设置”菜单，点击“模式”，弹出“模式”窗口
3. 点击“SN”，勾选“选择”，勾选“手动”

4.同样的把 ” WIFI MAC” ,” LAN MAC” ,” BT MAC” , “IMEI” ,的 “选择” 都取消勾选

5.设置完成后, 点击 “应用” 按钮, 关闭模式设置窗口, 返回主窗口。

6.点击 “写入” 按钮即可。

4.7.2 使用 WNPctool 读取序列号

1.进入 loader 模式。

2.点击 “设置” 菜单, 下拉框中勾选 “读取” 。

(勾选 “读取” 进行读取, 未勾选 “读取” 则切换到写入功能)

3.点击 “读取” 按钮即可。

第五章 主板 API 接口功能说明

5.1 基于 Andorid 标准 API 进行应用开发适配

基于 Andorid 7.1 标准 SDK 进行开发

compileSdkVersion 25

targetSdkVersion 25

5.2 系统设备序列号获取方法

```
public static final String SERIAL = getString("ro.boot.serialno");
```

在系统设置-关于平板电脑--状态信息---序列号 栏显示出来

上层只要去读 “ro.boot.serialno” 即可

/* 需要量产序列号的设备，请说明并提供序列号范围表 */

5.3 用系统签名 签 APK 的方法

```
java -Djava.library.path=. -jar signapk.jar platform.x509.pem platform.pk8  
$APK_NAME.apk $APK_NAME.signed.apk
```

5.4 静默安装 APK 方法

设备已经默认开启静默安装 APK，无需特殊操作，直接安装即可

5.5 UART 串口节点

默认开启串口对应节点路径如下：

UART4: `dev/ttyS4`

5.6 GPIO 口节点

开放 GPIO 对应节点路径如下：

扫码补光灯: GPIO1_C7: `/sys/class/gpio/gpio55/value`

灯条: GPIO1_C2: `/sys/class/gpio/gpio50/value`

5.6.1 java 读写 GPIO 节点方法(GPIO6 为例子):

```
public static final String NFC_RST_PIN = "/sys/class/gpio/gpio6/value";
```

```
public String gpio_pullup()
```

```
{
```

```
try {
```

```
    BufferedWriter bufWriter = null;
```

```
    bufWriter = new BufferedWriter(new FileWriter(NFC_RST_PIN));
```

```
bufWriter.write("1" + "");

bufWriter.flush();

bufWriter.close();

return "GPIO 设为高电平";

} catch (IOException e) {

    return "GPIO 设置错误: " + Log.getStackTraceString(e);

}

}
```

5.6.2 C 语言读写 GPIO 节点方法(GPIO6 为例子):

```
#define RST_GPIO "/sys/class/gpio/gpio6/value"

void GPIO_WriteBit(char * GPIO_NAME,int onValue)

{

    FILE *p=NULL;

    if(onValue == 1){

        p = fopen(GPIO_NAME,"w");

        fprintf(p,"%d",1);

    }
```

```
fclose(p);

} else{

    p = fopen(GPIO_NAME,"w");

    fprintf(p,"%d",0);

    fclose(p);

}

}
```

5.6.3 ADB 测试 GPIO 节点方法

//在保证 ADB 可以正常连接调试情况下

例 1： 改变 GPIO 状态(GPIO6 为例子):

```
$ adb shell
$ su
# echo "1">/sys/class/gpio/gpio6/value //GPIO 拉高
```

5.7 显示/隐藏下方导航栏 API

系统默认隐藏下方导航栏，监听广播来实现动态改变

显示下方导航栏广播：com.bozz.show_nav_bar

隐藏下方导航栏广播：com.bozz.hide_nav_bar

代码示例：

```
Intent intent = new Intent("com.bozz.show_nav_bar");  
sendBroadcast(intent);
```

ADB 测试方法：

```
$ adb shell  
$ adb shell am broadcast -a com.bozz.show_nav_bar //显示导航栏  
$ adb shell am broadcast -a com.bozz.hide_nav_bar //隐藏导航栏
```

5.8 打开/关闭调试 USB 调试模式 API

系统默认关闭调试 USB 调试模式，监听广播来实现动态改变

打开 USB 调试模式广播：android.intent.action.ADBENABLED

关闭 USB 调试模式广播：android.intent.action.ADBDISABLED

代码示例：

```
Intent intent = new Intent("android.intent.action.ADBENABLED");  
sendBroadcast(intent);
```

ADB 测试方法：

```
$ adb shell  
$ adb shell am broadcast -a android.intent.action.ADBDISABLED // 关闭
```

USB 调试模式

*ADB 无法测试打开 USB 调试模式

5.9 打开开发者选项方法

- 1.进入 “系统设置”
- 2.进入 “关于平板状态”
- 3.点击 “基带版本” 3 次, “版本号” 3 次, ” 内核版本” 3 次
- 4.返回, 即可看见 “开发者选项” 已经打开

第六章 启动测试模式的方法

6.1 启动压力老化测试:

- 1)如果需要压力老化测试: 需要将提供的 Aging_Test.bin 放入 U 盘根目录, 插入设备 USB 口, 将会自动打开 CPU, DDR, GPU,VPU 的压力老化测试

6.2 打开厂测的方法:

新建 flag.txt 文本, 文本内容填写为 devices

将 flag.txt 拷贝到 U 盘根目录, 插入设备 USB 口, 将会自动打开厂测模式

*激活厂测模式会自动激活 USB 调试

第七章 使用注意事项

- 1.相对湿度: $\leq 80\%$;
2. 存储温度: $-0\sim 60^{\circ}\text{C}$;

3. 使用温度: 0~50℃ ;
4. 请使板卡远离静电 ;
5. 勿受重压及弯折变形, 跌落 ;
6. 正确接好驱屏线前请勿接通电源 ;
7. 当板卡正在工作时切勿在板卡上掉入可导电物体 ;
8. 请勿拆解此板卡 ;
9. 如果板卡有灰尘, 请用干布擦拭。