

F4D

技术规格书

Bozz Technology

发布版本:V1.5

日期:2020.03.02

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，本档内容会不定期进行更新。除非另有约定，本档仅作为技术规格说明和使用指导，本档中的所有陈述、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、复制本档内容的部分或全部，并不得以任何形式传播。

Bozz Technology

深圳博时特科技有限公司


地址：深圳市龙华新区油松路 103-1 (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@bozzteck.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 SW7330-F4D 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 SW7330 进行应用开发，熟悉 SW7330 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
F4D	SW7330	

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
2019-05-28	V1.0	陈政、王清明	初始发布

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

目 录

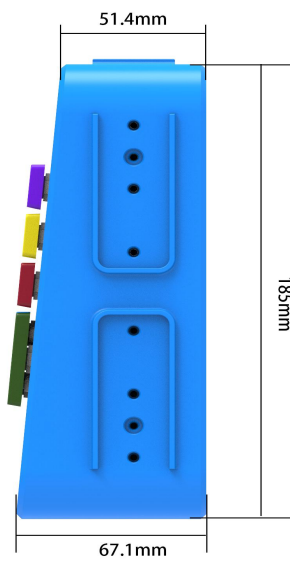
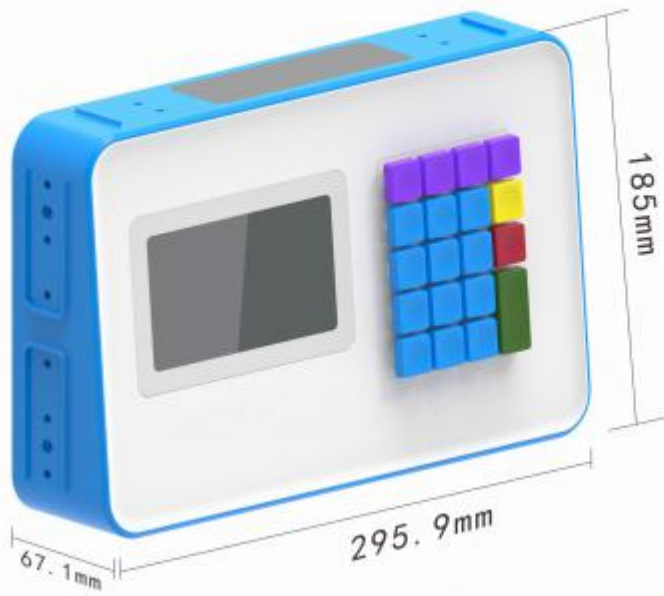
前言.....	3
概述.....	3
产品版本.....	3
适用对象.....	3
修订记录.....	4
缩略语.....	5
1. 博时特 F4D 产品功能概述.....	8
1.1 F4D 产品外观尺寸.....	8
1.1.1 产品尺寸.....	8
1.1.2 主屏及按键.....	9
1.1.3 客屏及扫码窗.....	10
1.1.3 外围接口.....	10
1.2 F4D 产品功能概述.....	11
1.3 F4D 产品硬件规格.....	11
1.4 F4D 产品行业应用市场概述.....	12
2. SW7330 平台概述.....	13
2.1 SW7330 平台简介.....	13
2.2 PX30 主控芯片介绍.....	13
2.3 PX30 芯片功能特点.....	15
2.4 SW7330 平台系统框图.....	16
2.5 SW7330 平台功能概述.....	17
2.6 SW7330 平台规格.....	17
2.7 SW7330 包装组件.....	19
3. SW7330 硬件尺寸与接口说明.....	20
3.1 PCBA 尺寸.....	20
3.2.TOP 主要硬件接口说明.....	21
3.2.BOTTOM 主要硬件接口说明.....	22
3.3.主要硬件接口定义说明.....	22
4. SW7330 PCBA 开发板的使用.....	30
4.1 开关机和待机.....	30
4.2 USB 驱动安装.....	31
4.3 固件烧写方法.....	32
4.3.1 Maskrom 烧写模式.....	32
4.3.2 Loader 烧录模式.....	33
4.4 串口调试.....	34
4.4.1 连接串口.....	34
4.4.2 ADB 调试.....	37
5. SW7330 API 接口功能说明.....	38
5.1 GPIO 口.....	38
5.1.1 GPIO 口开放的 sys 节点路径.....	38
5.1.2 查看 GPIO 全局配置的方法.....	38
5.1.3 用 java 读写 GPIO 节点方法:	38

- 5.1.4 用 C 语言读写 GPIO 节点方法: 39
- 5.2 LED 跑马灯控制 GPIO 口。40
 - 5.2.1 F4D 主板开放了 3 个 RGB led GPIO 口用于控制 LED 灯 40
 - 5.2.2 RGB led GPIO 对应的 sys 节点路径..... 40
 - 5.2.3 用 ADB 测试 LED 节点方法.....41
 - 5.2.4 用 JAVA 控制 LED 闪烁的方法: 41
- 5.3 读取键盘值用于应用开发.....43
- 5.4 显示/隐藏导航栏.....43
 - 5.4.1 应用层..... 43
 - 5.4.2 adb 测试显示/隐藏导航栏..... 43
- 5.5 系统设备序列号获取方法..... 44
- 5.6 系统签名获取系统权限的方法..... 44
- 5.7 基于 Andorid 标准 API 进行应用开发适配..... 44
- 6.启动测试模式的方法: 45
 - 6.1 启动老化测试: 45
 - 6.2 打开厂测的方法: 45
 - 6.3 打开压力测试的方法: 45
- 7.使用注意事项..... 46

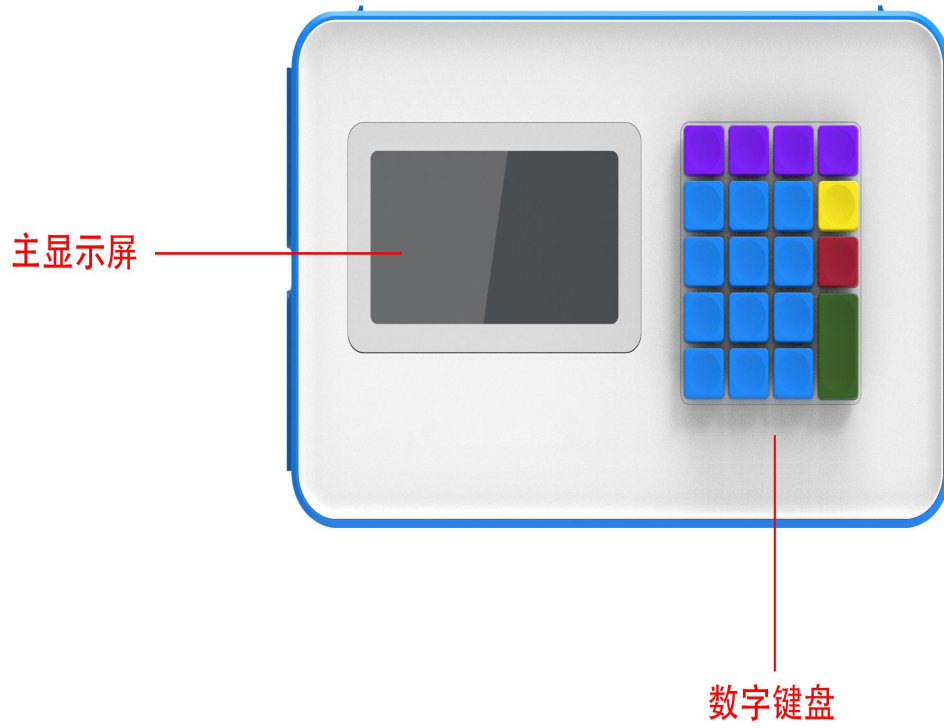
1. 博时特 F4D 产品功能概述

1.1 F4D 产品外观尺寸

1.1.1 产品尺寸

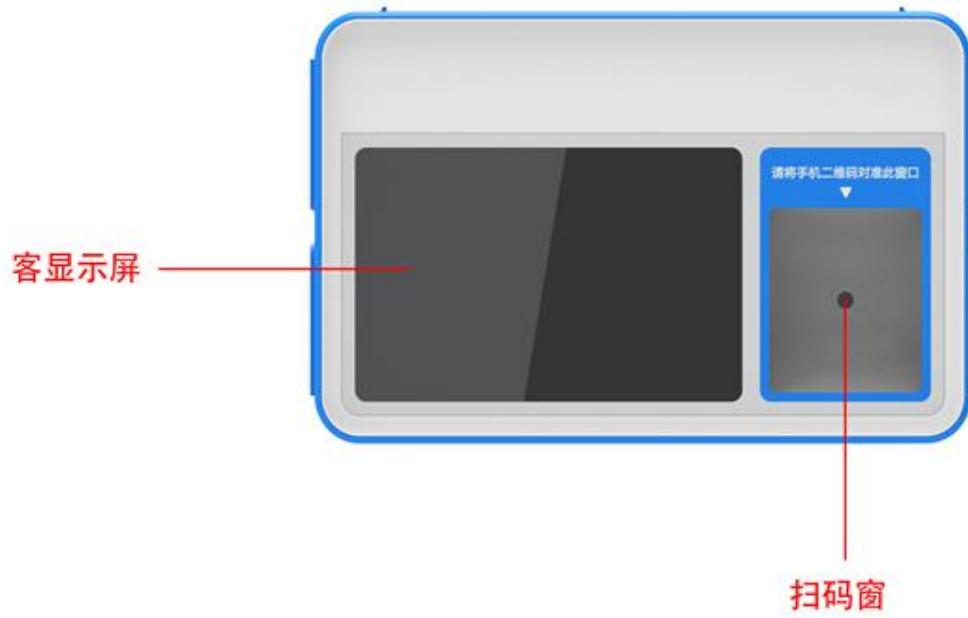


1.1.2 主屏及按键

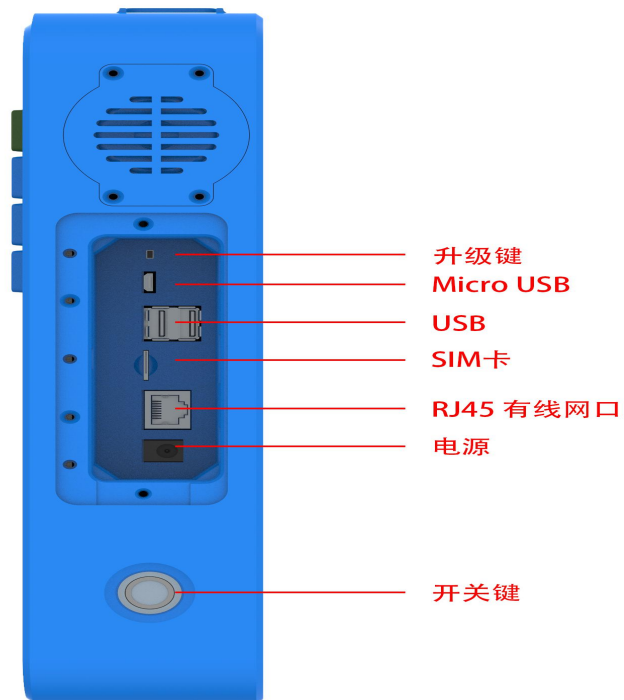


Bozz

1.1.3 客屏及扫码窗



1.1.3 外围接口



1.2 F4D 产品功能概述

F4D 产品是深圳博时特科技有限公司基于 SW7330 平台开发的双屏智能收银终端，通过扫码方式完成便捷支付，支持双屏单触，通过聚合支付为客户提供便捷的收银体验，提升高峰阶段的收银效率。

主频率高达 1.5GHZ，运行内存 1GB，存储 8GB，搭载 Android8.1 系统，配有 5 寸和 8 寸高清显示屏、5 寸电容触摸屏、1600*1200 分辨率 200 万像素高清摄像头、19 键数字键盘、4G 模块、WIFI 模组，三色 LED 扫码窗、防水扬声器。支持双屏同显，双屏异显，5 点电容触摸，19 键数字键盘输入，高灵敏度的条码、二维码扫码识别能力；支持 4G、以太网、WIFI 多种联网模式；支持多种格式的音视频编解码。外围配有 1 个 MICRO USB，2 个 USB 2.0 接口，可通过 USB 升级、连接各种 USB 设备。

1.3 F4D 产品硬件规格

CPU	RK PX30 瑞星微电子 Quad-core Cortex-A35 up to 1.5GHz Mali-G31 GPU
ROM	EMMC 8GB
RAM	SAMSUNG DDR3L 1GB
系统	Android 系统：8.1.0
触摸屏	主屏：5 英寸 5 点触摸
屏幕	主屏：TN 液晶屏 5 英寸、分辨率 854*480 副屏：IPS 液晶屏 8 英寸、分辨率 1280*800
基本接口	USB：x2 A 类母头/USB2.0 MICRO USB (OTG)：x1 MicroUSB 母座 以太网口：x1 10/100M SIM 卡座：x1 无线：2.4G WIFI DC 电源：x1 12V/1.5A

摄像头	扫码识别摄像头： 接口类型：USB 传感器像素：200 万 最高分辨率：1600*1200
4G 模块	4G/全网通，外置 NANO SIM 卡槽，支持热插拔
扬声器	内阻：8 欧姆 额定功率：3W 音强：不低于 90 分贝

1.4 F4D 产品行业应用市场概述

F4D 是一款可定制触控 POS 终端，搭配 19 键数字键盘和 5 寸电容触摸，操作方便快捷，可应用于各种不同的支付场景，产品应用涉及餐饮、零售、娱乐休闲等各种支付领域，高灵敏度的条码、二维码扫码能力，高清晰度的双屏显示，在应用中能提供精确、简单快捷、实时可见的用户体验。

2. SW7330 平台概述

2.1 SW7330 平台简介

SW7330 是博时特科技有限公司基于 PX30 多媒体处理芯片（以下简称 PX30 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 SW7330 项目强大的多媒体接口和丰富的外围接口，同时为客户提供基于 SW7330 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。SW7330 基于 Android8.1 标准 SDK，支持应用软件的开发、调试和运行等。

SW7330 可通过 USB 线与电脑连接，作为 USB DEVICE 设备下载更新程序，或实现更完全的开发系统或演示环境，此时连接如下设备或部件：

- 电源
- LCM MIPI屏
- TF Card存储设备
- 耳机或音箱
- 摄像头模组

2.2 PX30 主控芯片介绍

SW7330 的主控芯片是 Rockchip PX30，它是一颗高性能的四核处理器芯片，频率最高可达 1.5GHz,主要应用于智能 POS 机、平板、智能音频设备，内置 Mali-G31 GPU，支持 OpenGL ES3.2, Vulkan 1.0, OpenCL 2.0。该芯片集成度高，拥有丰富的外围接口，芯片架构框图参考图 2-1

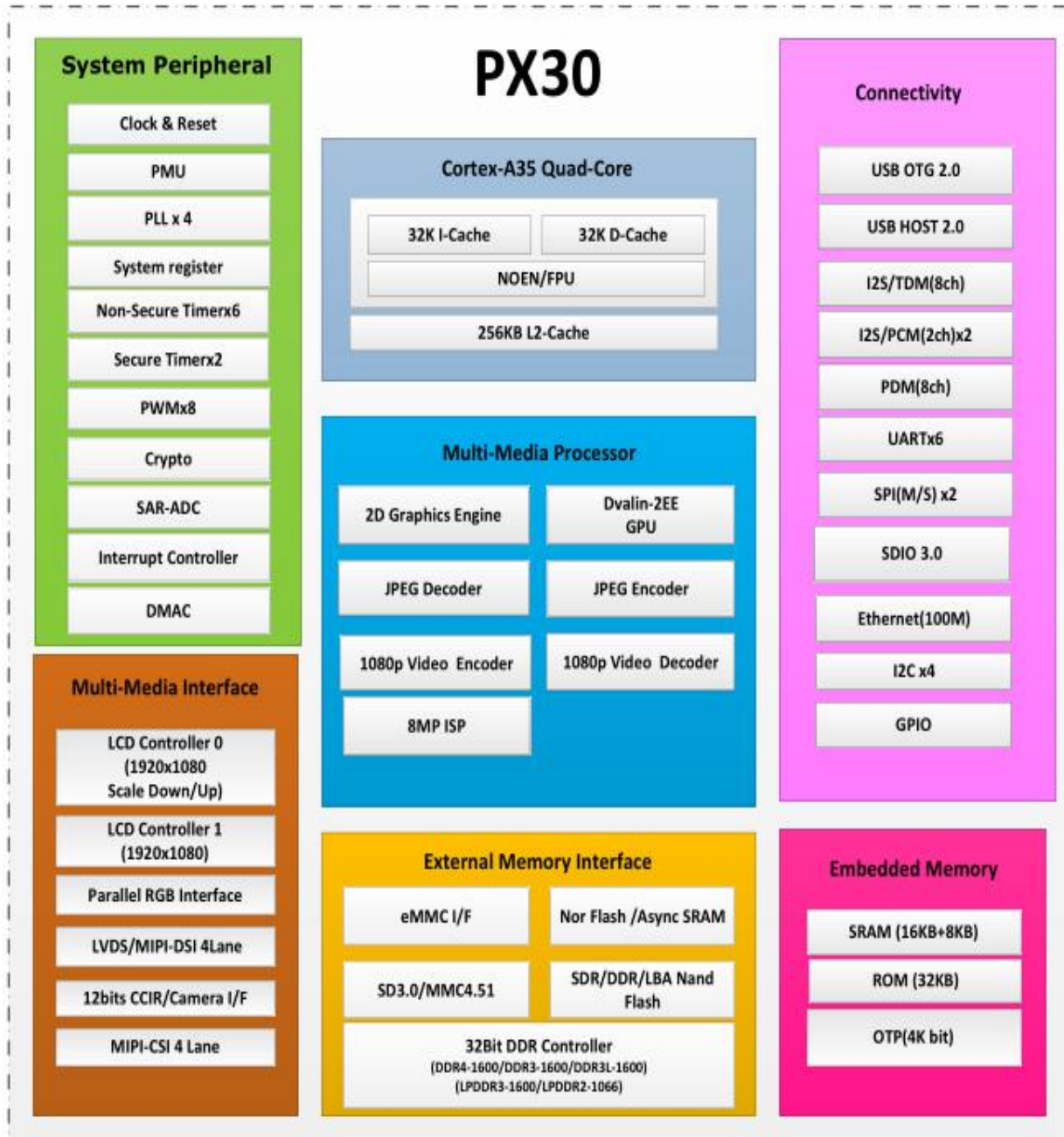


图 2-1 PX30芯片架构

2.3 PX30 芯片功能特点

1、Rockchip PX30 内置 3D GPU，能够完全兼容 OpenGL ES1.1/2.0/3.2、DirectX 11.1、OpenCL 2.0 和 Vulkan 1.0。特殊的 MMU 2D 硬解码器能最大限度地提高显示性能，提供流畅的体验操作；

2、支持不同类型的 DDR 存储器接口（DDR3/DDR3L/DDR4/LPDDR2/LPDDR3）；

3、支持双屏异显，支持 RGB/LVDS/MIPI-DSI，分辨率最高 1920x1080；

4、支持有线网络 10/100M、无线网络 2.4G、4G 全网通；

5、SW7330 是为智能 POS 机定制的主板，支持 200 万像素 CIF 摄像头、支持 RS232/USB/UART 通讯接口功能扩展、RTC 时钟、智能语音播报、OTG 功能等。

2.4 SW7330 平台系统框图

平台系统框图可以让开发人员对整个系统的架构和原理有一个直观的认识，整个系统由电源适配器或者电池供电，通过 UART 串口进行调试，验证各功能模块。SW7330 平台带有大部分接口，配有 Camera 输入，WIFI 模组，USB OTG，USB HOST，TF 卡，音频接口，视频接口，百兆网口，满足大多数情况下不同应用需求，有利于各种行业应用方案的深入研发与快速产品化。

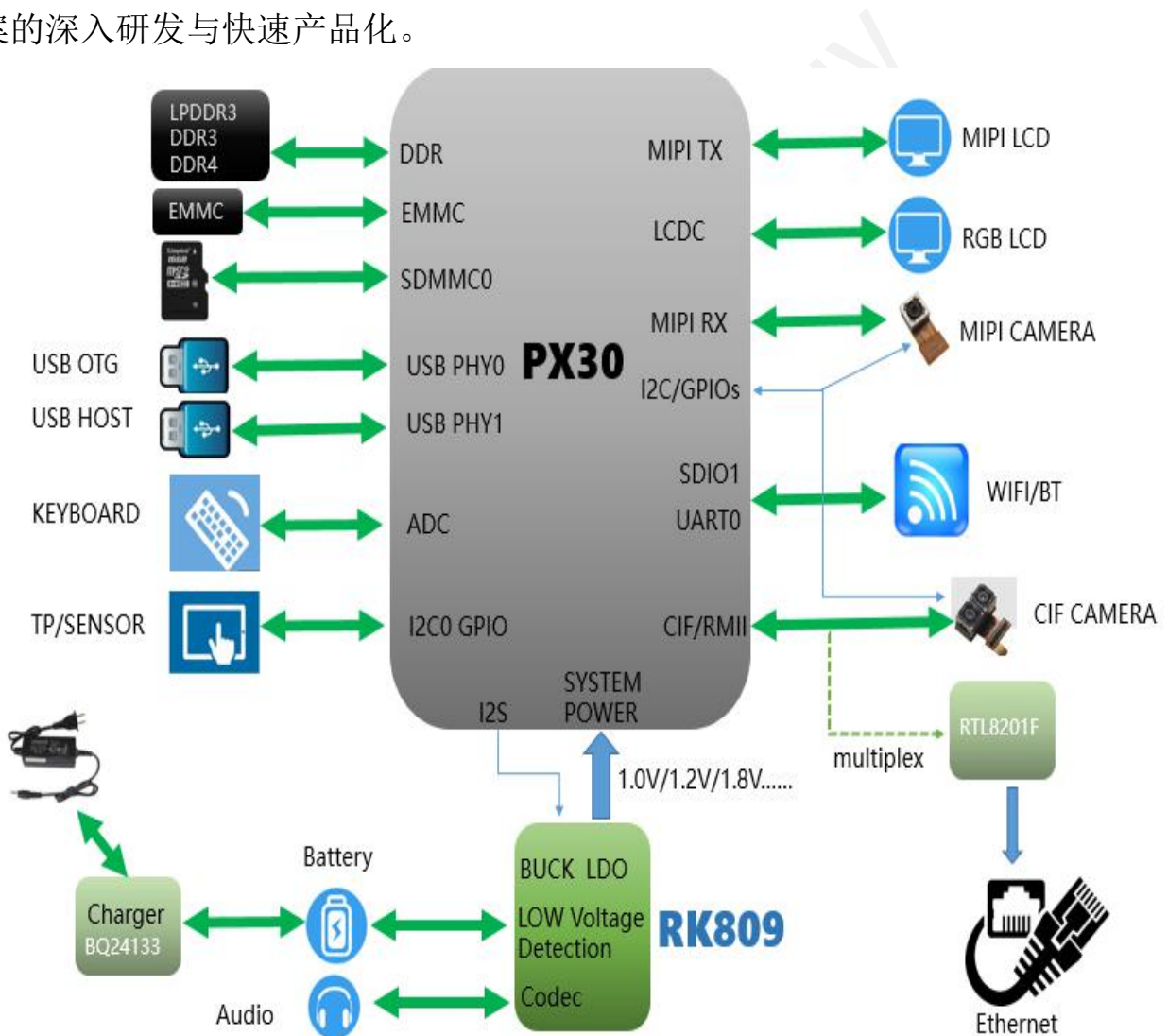


图 2-2 SW7330-F4D系统框图

2.5 SW7330 平台功能概述

SW7330 包含的功能如下：

- RK809-1电源路径管理系统
- 4 * 256M * 16bit DDR3, 总容量2GByte
- 8bit eMMC, 总容量16GByte
- TF Card: 支持外部扩展存储容量
- USB OTG: 系统升级使用, 可以支持Host/Device切换
- USB HOST: 支持各种USB设备
- 系统按键: Power、VOL+、VOL-、RESET、Maskrom
- SDIO Wifi: 支持无线上网功能
- Audio out: 支持耳机、扬声器
- Audio in: 支持录音
- RMII:支持百兆网络
- Uart Debug: 开发板Debug使用
- CIF Camera: OV5288, 200W像素
- 扩展接口包含: CIF、RGB24bit、Raspberry Pi

2.6 SW7330 平台规格

主要硬件指标	
板卡配置	Rockchip PX30, CPU 四核 ARM Cortex A35, Mali-G31 GPU, SAMSUNG DDR3L 1GB, SanDisk EMMC 8GB
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 8.0
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 以太网
USB OTG 接口	X 1, MICRO_USB_5PIN
USB 2.0 接口	USB 2.0×7 (2 个内置 4P_2.0 座子, 2 路 USB-A 母座, 1 路用于 4G, 1 路用于 WIFI, 1 路用于 RGMII)
扩展功能接口	RS232 ×2 (UART0, UART1), 打印串口×1 (UART2)
LVDS 屏接口	× 1 (可选)
RGB 屏接口	× 1
MIPI 屏接口	× 1
TF 卡座	× 1
I2C 触摸	× 2
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 1GB+8GB, 支持 2GB+32GB
显示	双屏双触、双屏异显
MIC	语音输入
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/1.5A 直流电源输入(可选配, 依功放喇叭功率、显示屏与背光而定)
------	--

2.7 SW7330 包装组件

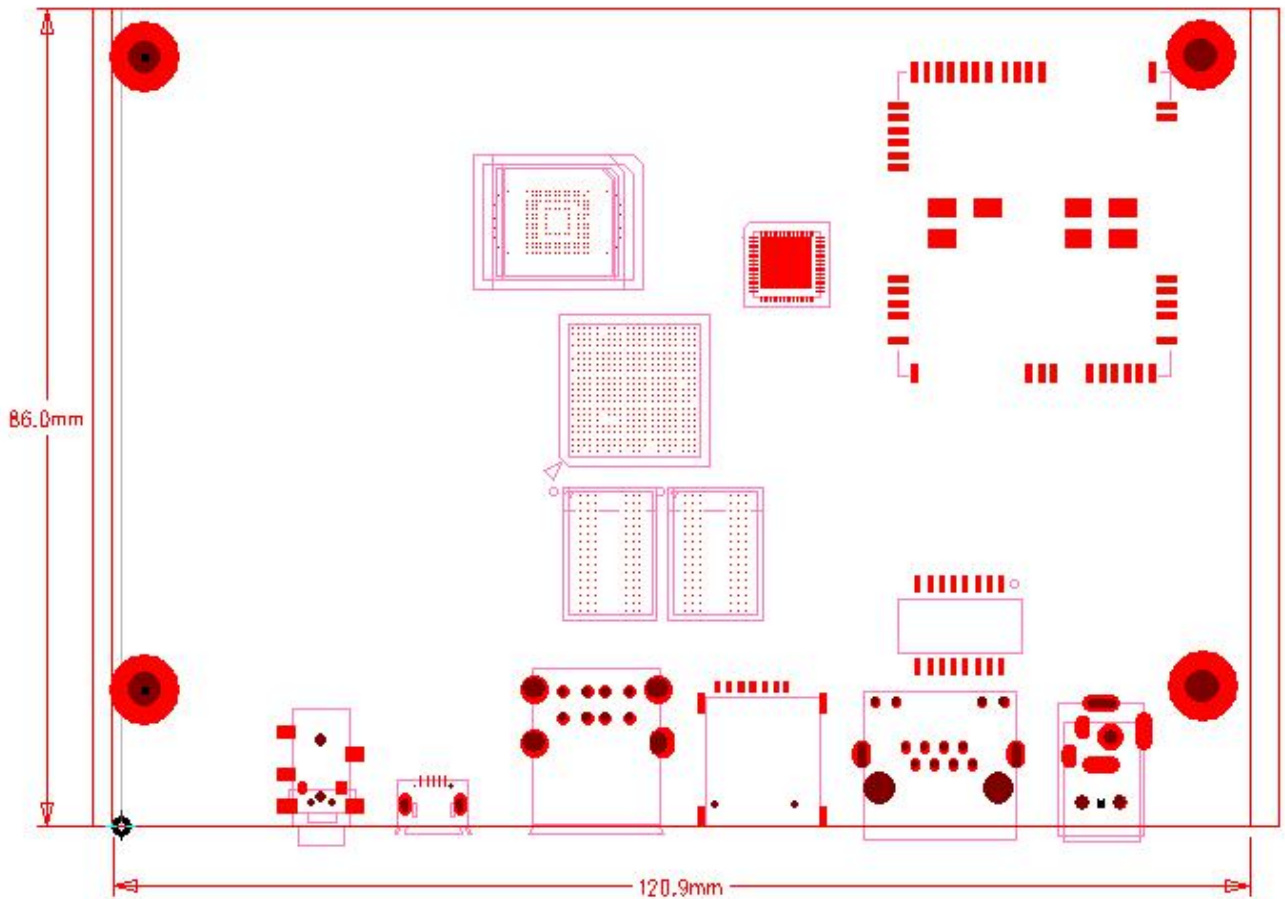
SW7330 F4D 主要包括以下物品:

- SW7330整机
- 电源适配器, 规格: 输入 100V AC~240V AC, 50Hz; 输出 12V DC, 2.0A
- 包装材料

Bozz Technology

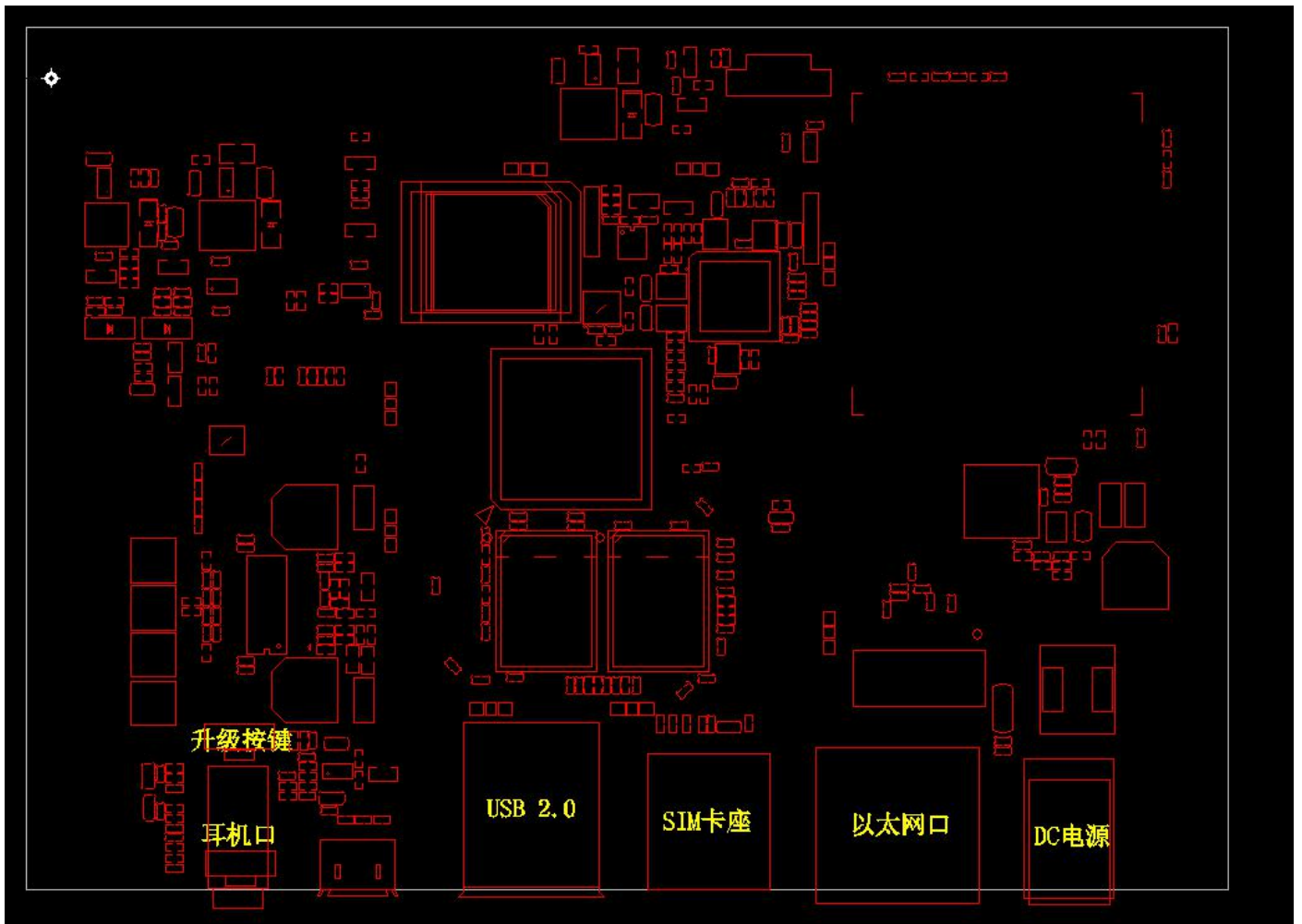
3. SW7330 硬件尺寸与接口说明

3.1 PCBA 尺寸

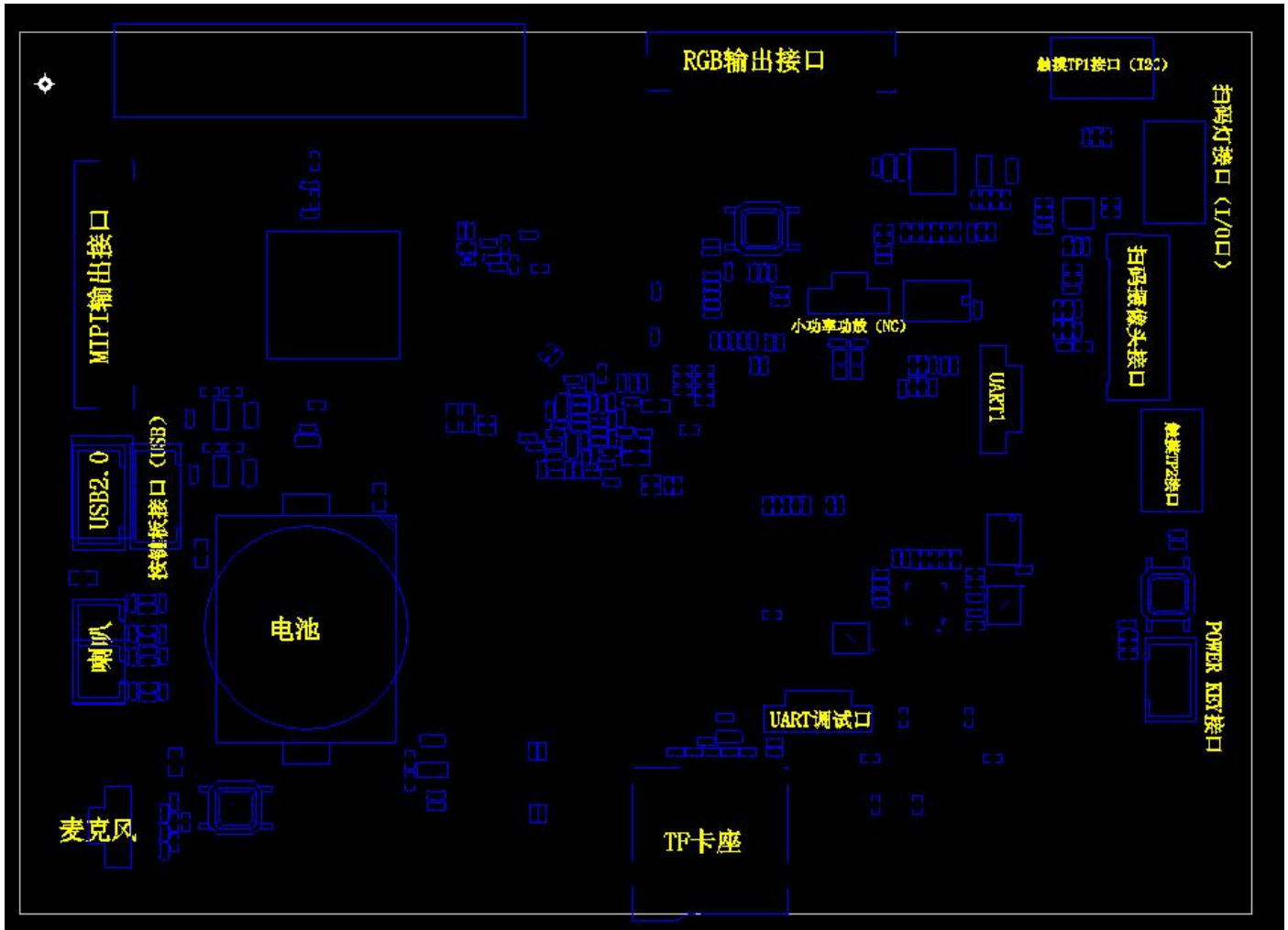


PCB 相关尺寸及规格:长 120.9MM、宽 86MM、厚 1.2MM, 螺丝孔规格: 直径 3.0MM。

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

J34(RJ45-屏蔽带灯): 以太网接口

序号	定义	描述
1	TX+	发送数据+
2	TX-	发送数据-
3	RX+	接收数据+
4	NC	NC
5	NC	NC
6	RX-	接收数据-
7	NC	NC
8	NC	NC
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

J1(SIM_NANO): 4G 卡座

序号	定义	描述
1	VCC	SIM卡电源
2	RST	SIM卡复位
3	CLK	SIM卡时钟
4	DET	SIM卡检测
5	GND	地
6	VPP	接地
7	IO	SIM卡数据

J29(双层 USB-A 母座): USB 接口

序号	定义	描述
1	VCC	USB电源
2	GND	地
3	DM1	USB1 D-
4	DP1	USB1 D+
5	VCC	USB电源
6	DM2	USB2 D-
7	DP2	USB2 D+
8	GND	地

J13 (MIRCO-USB-5P): USB OTG 接口

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

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

J5 (4P_1.25MM): 串口通讯接口 UART0/RS232

序号	定义	描述
1	VCC	串口电源
2	TX0	串口0数据发送
3	RX0	串口0数据接收
4	GND	地

J10(2P_2.0MM): 喇叭

序号	定义	描述
1	LN	喇叭-
2	LP	喇叭+

J27(2P_1.25MM): MIC

序号	定义	描述
1	MIC_N	咪头-
2	MIC_P	咪头+

J2(4P_2.0MM): 键盘接口

序号	定义	描述
1	VCC	键盘电源
2	KEY_DM	键盘数据D-
3	KEY_DP	键盘数据D+
4	GND	地

J20(MIRCO-TF): TF 卡座

序号	定义	描述
1	D2	数据2
2	D3	数据3
3	CMD	命令/响应
4	VCC	电源
5	CLK	时钟
6	VSS	接地
7	D0	数据0
8	D1	数据1
9	DET	检测

J7(4P_1.25MM): 调试串口接口 UART2/RS232

序号	定义	描述
1	NC	NC
2	TX	串口数据发送
3	RX	串口数据接收
4	GND	地

J19 (4P_1.25MM): 串口通讯接口 UART1/RS232

序号	定义	描述
1	VCC	串口电源
2	TX1	串口1数据发送
3	RX1	串口1数据接收
4	GND	地

JX1(24P_0.5MM): CIF 摄像头接口

序号	定义	描述
1	NC	

2	GND	地
3	SDA	I2C数据
4	AVDD	2.8V电源
5	SCL	I2C时钟
6	RST	复位
7	VSYNC	VSYNC信号
8	PDN0	PDN0信号
9	HREF	HREF信号
10	DVDD	1.8V电源
11	DOVDD	2.8V电源
12	DATA7	数据7
13	MCLK	MCLK信号
14	DATA6	数据6
15	GND	地
16	DATA5	数据5
17	PCLK	PCLK信号
18	DATA4	数据4
19	DATA0	数据0
20	DATA3	数据3
21	DATA1	数据1
22	DATA2	数据2
23	NC	
24	NC	

JP4(3P_2.0MM): 开机键接口

序号	定义	描述
1	PWR_ON	锁屏/开关机按键信号
2	VCC5V	5V背光灯电源
3	GND	地

J47(10P_0.5MM): TP2 接口

序号	定义	描述
1	GND	地
2	GND	地
3	TP2_RST	TP2复位

4	TP2_INT	TP2中断
5	GND	地
6	SCL	I2C时钟
7	SDA	I2C数据
8	VCC	3.0V供电
9	GND	地
10	GND	地

J31(10P_0.5MM): 扫码灯接口

序号	定义	描述
1	GND	地
2	VCC3V	3.0V电源
3	UART1_RX	UART1_接收数据
4	UART1_TX	UART1_发送数据
5	GND	地
6	BLUE_LED	蓝色灯控制口
7	GREEN_LED	绿色灯控制口
8	RED_LED	红色灯控制口
9	VCC5V	地
10	GND	地

J4 (10P_0.5MM): TP1 接口

序号	定义	描述
1	GND	地
2	GND	地
3	TP1_RST	TP1复位
4	TP1_INT	TP1中断
5	GND	地
6	SCL	I2C时钟
7	SDA	I2C数据
8	VCC	3.0V供电
9	GND	地
10	GND	地

J3(2P_1.25MM): PMIC RK809 内置功放 1.3W 输出接口 (未调试)

序号	定义	描述
----	----	----

1	SPK_N	SPK-
2	SPK_P	SPK+

CN1(40P_0.5MM): RGB 屏接口

序号	定义	描述
1	LED-	LED-
2	LED+	LED+
3	GND	地
4	VCC	3.3V电源
5	LCD_D16	NC
6	LCD_D17	NC
7	LCD_D18	NC
8	LCD_D19	NC
9	LCD_D20	NC
10	LCD_D21	NC
11	LCD_D22	NC
12	LCD_D23	NC
13	LCD_D8	G3 数据输入
14	LCD_D9	G4 数据输入
15	LCD_D10	G5 数据输入
16	LCD_D11	R0 数据输入
17	LCD_D12	R1 数据输入
18	LCD_D13	R2 数据输入
19	LCD_D14	R3 数据输入
20	LCD_D15	R4 数据输入
21	LCD_D0	B0 数据输入
22	LCD_D1	B1 数据输入
23	LCD_D2	B2 数据输入
24	LCD_D3	B3 数据输入
25	LCD_D4	B4 数据输入
26	LCD_D5	G0 数据输入
27	LCD_D6	G1 数据输入
28	LCD_D7	G2 数据输入
29	GND	地
30	CLK	屏像素时钟信号
31	RST	屏复位
32	HSYNC	HSYNC 行同步信号

33	VSYNC	VSYNC 场同步信号
34	DE	DE 数据使能信号
35	VCC1V8	1.8V电源
36	GND	地
37	SPI_CLK	SPI时钟
38	SPI_MOSI	SPI输入
39	SPI_MISO	
40	SPI_CS	SPI片选

CN2(40P_0.5MM): MIPI 屏接口 (兼容 LVDS 屏)

序号	定义	描述
1	VCOM	VCOM电压
2	VCC3V3	3.3V电源
3	VCC3V3	3.3V电源
4	NC	
5	RST	屏复位
6	VCC3V3	3.3V电源
7	GND	地
8	D0N	D0N 数据信号
9	D0P	D0P 数据信号
10	GND	地
11	D1N	D1N 数据信号
12	D1P	D1P 数据信号
13	GND	地
14	D2N	D2N 数据信号
15	D2P	D2P 数据信号
16	GND	地
17	CLKN	CLKN 时钟信号
18	CLKP	CLKP 时钟信号
19	GND	地
20	D3N	D3N 数据信号
21	D3P	D3P 数据信号
22	GND	地
23	VCC1V8	1.8V电源
24	VCC1V8	1.8V电源
25	GND	地
26	NC	NC

27	NC	NC
28	NC	NC
29	AVDD	AVDD电压
30	GND	地
31	LED2-	LED2-
32	LED2-	LED2-
33	VCC3V3	3.3V电源
34	GND	地
35	VGL	VGL电压
36	GND	地
37	GND	地
38	VGH	VGH电压
39	LED2+	LED2+
40	LED2+	LED2+

J6(4P_2.0MM): 扩展 USB 协议模块接口

序号	定义	描述
1	VCC	模块电源
2	USB_DM	USB数据D-
3	USB_DP	USB数据D+
4	GND	地

4. SW7330 PCBA 开发板的使用

4.1 开关机和待机

开机和关机方法介绍如下：

1、开机方法：

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 2s，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、待机的方法：

在桌面或者应用场景下，按下 Power 键，系统会进入一级待机状态。在没有连接 USB 的情况下，不做任何操作，系统会在一段时间后，由一级待机转入二级待机状态

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，驱动工具路径：

SDK\RKTools\windows\Release_DriverAssitant 目录下，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。

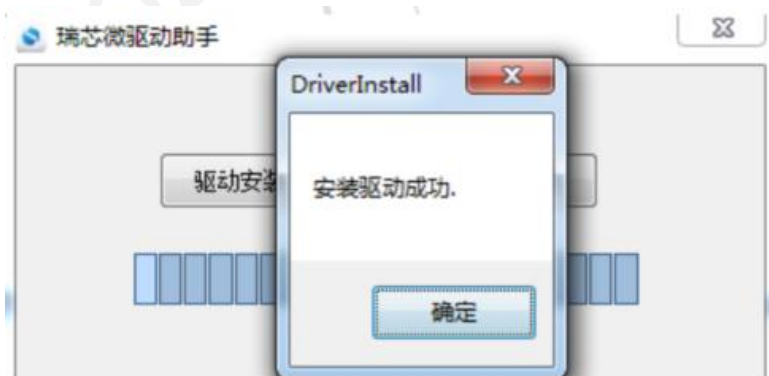


图 4-0-1 驱动安装成功示意图

4.3 固件烧写方法

SW7330 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 SW7330 供电 12V，并打开船型开关；要是已经处于上电情况下，请按下复位按键。
- 3) 等待会儿开发工具将显示“发现一个 Maskrom 设备”，需要注意的是在 Maskrom 状态下需要同时选择对应的 Loader 才能升级。
- 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。

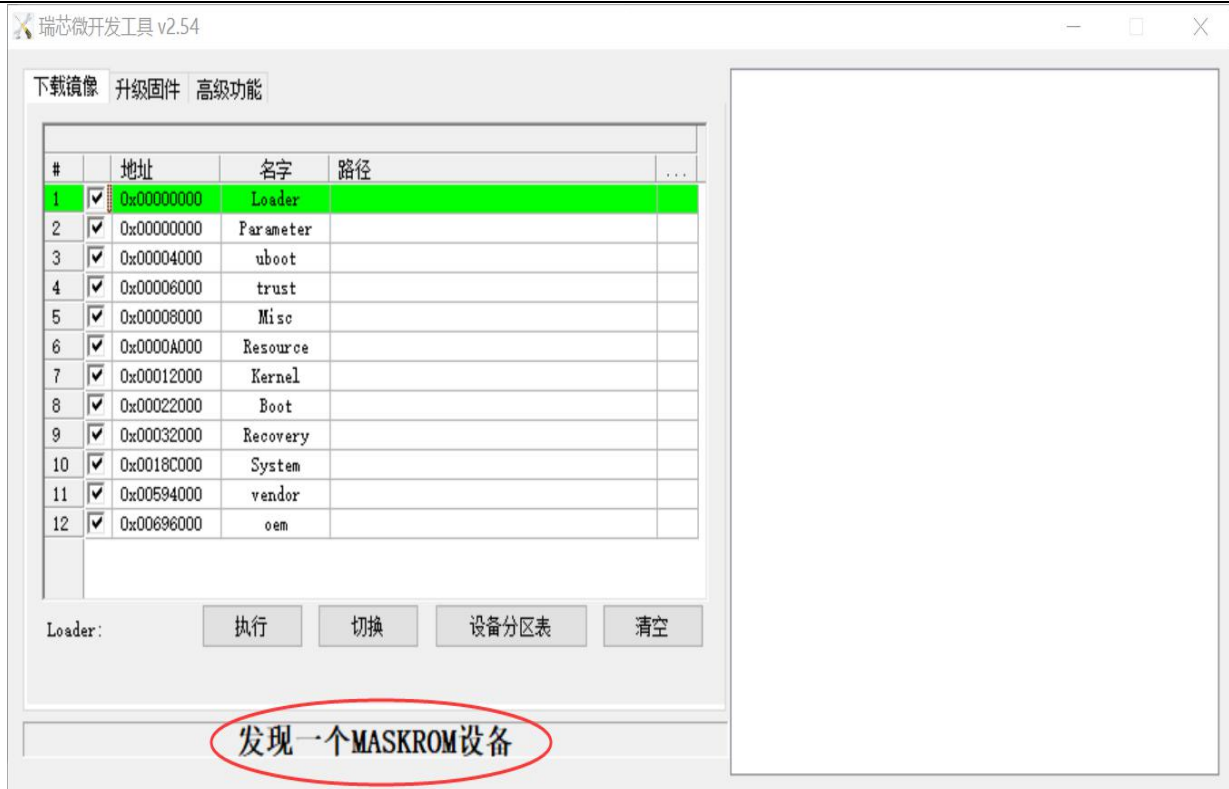


图4-0-2 进入Maskrom烧写模式工具上示意图

4.3.2 Loader 烧录模式

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，更换固件中的部分或者全部。

具体步骤如下：

- 1) 并按住开发板的 Vol+/RECOVER 按键不放，连接 USB 到电脑 PC 端。
- 2) 给 SW7330 供电 12v，并打开船型开关；要是已经处于上电情况下，请按下复位按键。
- 3) 等待会儿开发工具将显示“发现一个 Loader 设备”，需要注意的是在 Loader 模式下，不需要烧录完整的固件，可以只选择需要更新的 image 文件。
- 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。

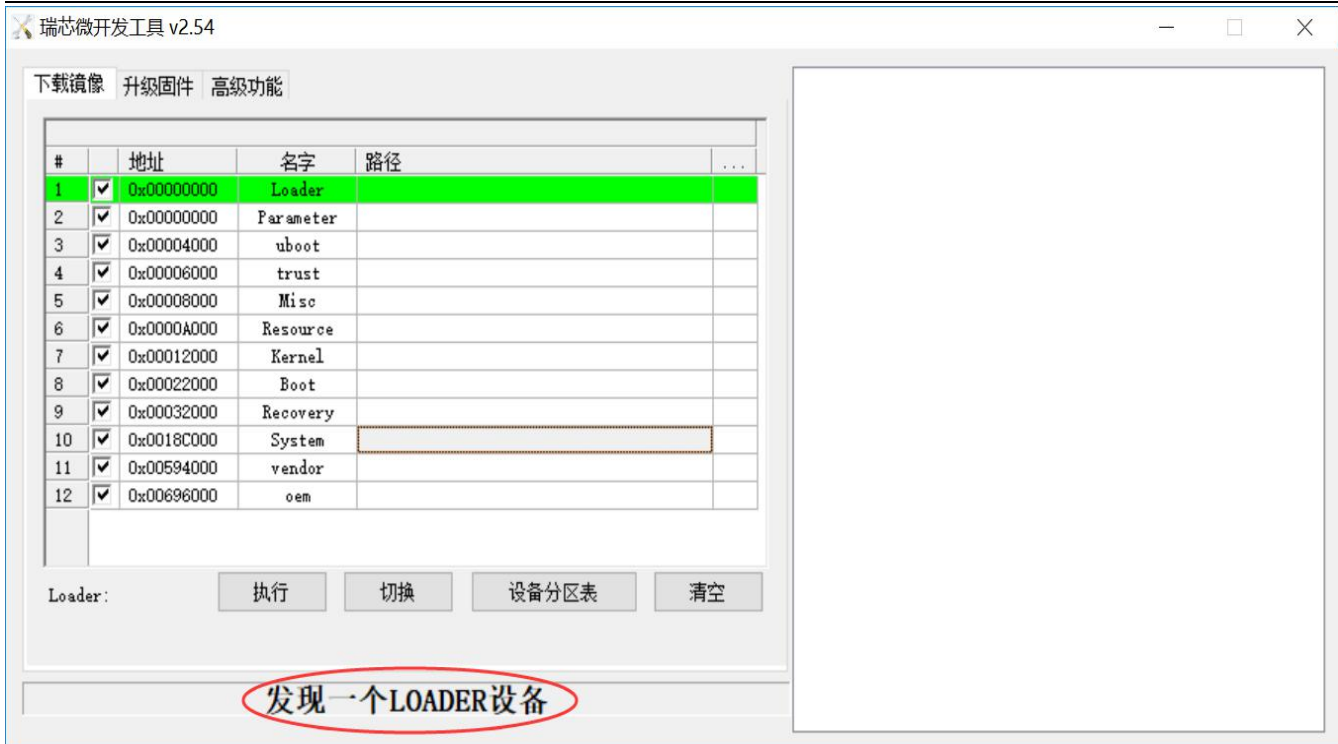


图4-0-3 进入Loader烧写模式工具上示意图

4.4 串口调试

4.4.1 连接串口

连接 SW7330 的 USB Debug 到电脑 PC 端，在 PC 端设备管理器中得到当前端口的 COM 号。

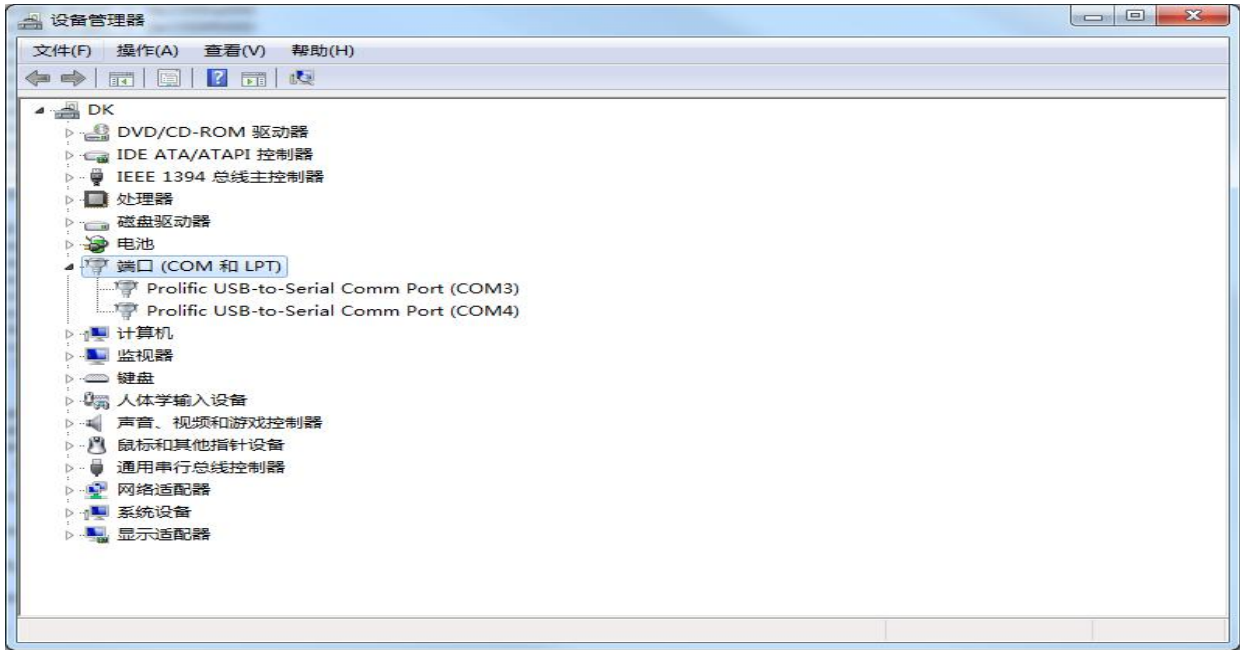


图 4-0-4 获取当前端口COM号

打开串口工具“SecureCRT”，点击“快速连接”按钮。

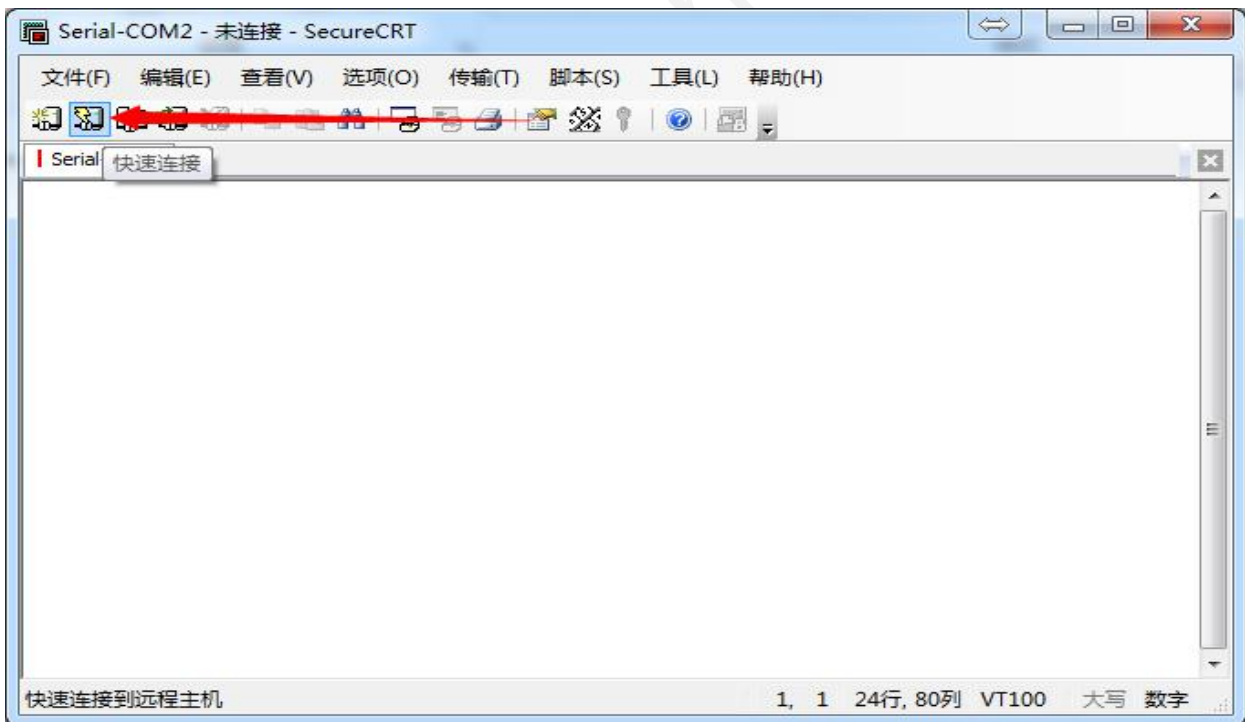


图 4-0-5 串口工具SecureCRT界面

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，流控 RTS/CTS 不需勾选。

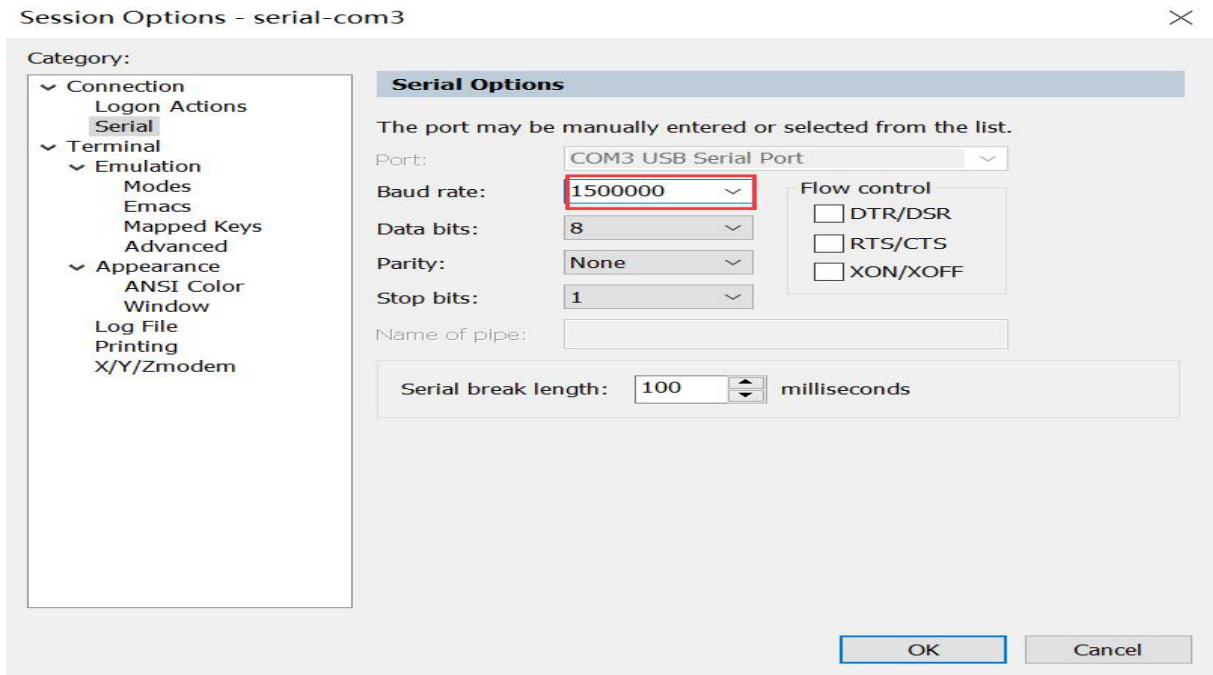


图 4-0-6 配置串口信息

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。

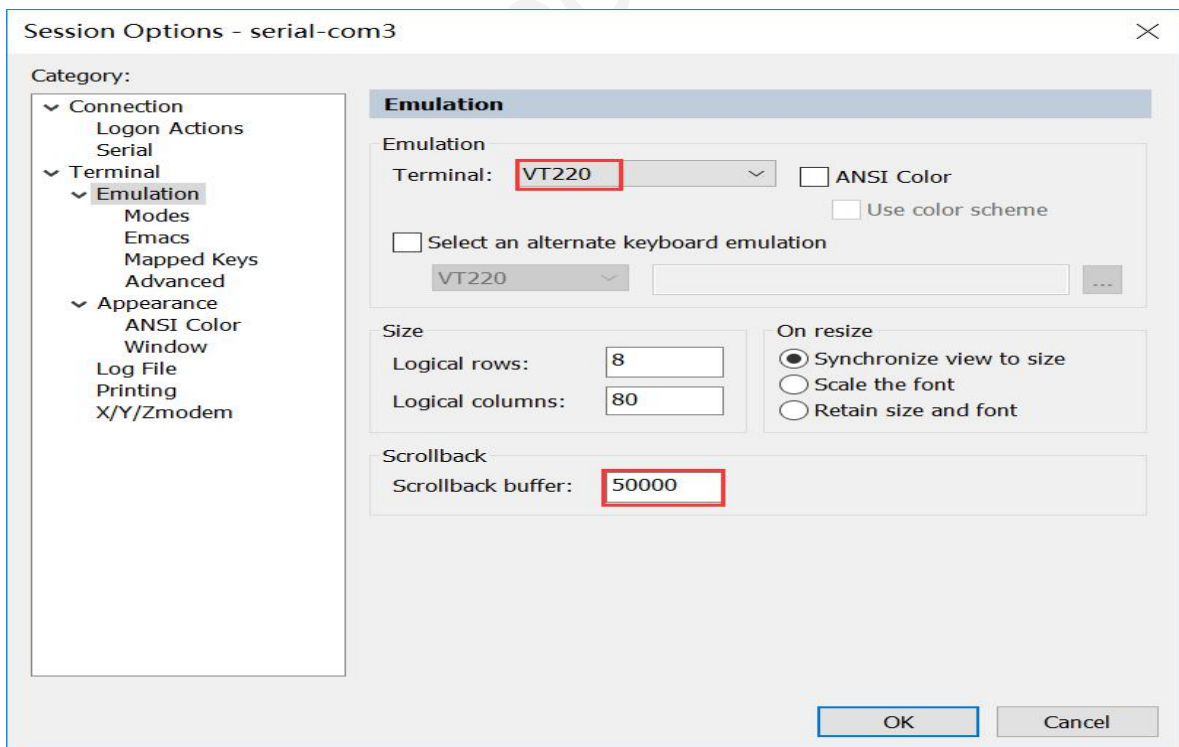


图 4-0-7 配置串口工具选项

4.4.2 ADB 调试

- 1) 确保驱动安装成功，PC 连接开发板的 USB OTG 口；
- 2) 开发板上电，开机进入系统，再进入 setting 项，选择“developer options”，勾选“USB debugging”；
- 3) 电脑 PC 端，点击“开始---运行”，输入 cmd，进入 adb.exe 工具所在的目录，输入“adb devices”，可以查询到连接的设备，表示连接正常；
- 4) 输入“adb shell”，进入 ADB 调试。

5. SW7330 API 接口功能说明

5.1 GPIO 口

5.1.1 GPIO 口开放的 sys 节点路径

```
/sys/class/gpio/
```

5.1.2 查看 GPIO 全局配置的方法

```
$adb root
```

```
$adb remount
```

```
$adb shell cat /sys/kernel/debug/gpio
```

5.1.3 用 java 读写 GPIO 节点方法:

//在保证 ADB 可以正常连接调试情况下

```
public static final String NFC_RST_PIN = "/sys/class/gpio/gpio106/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);
    }
}
```

5.1.4 用 C 语言读写 GPIO 节点方法:

```
#define RST_GPIO "/sys/class/gpio/gpio106/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);

    }
}
```

```
}
}
}
```

5.2 LED 跑马灯控制 GPIO 口。

5.2.1 F4D 主板开放了 3 个 RGB led GPIO 口用于控制 LED 灯

```
#blue
/sys/class/gpio/gpio53/value

#red
/sys/class/gpio/gpio54/value

#green
/sys/class/gpio/gpio55/value
```

5.2.2 RGB led GPIO 对应的 sys 节点路径

```
chmod 0777 /sys/class/gpio/
chmod 0777 /sys/class/gpio/export
#blue
write /sys/class/gpio/export 53
chmod 0777 /sys/class/gpio/gpio53/
chmod 0777 /sys/class/gpio/gpio53/direction
chmod 0777 /sys/class/gpio/gpio53/value
write /sys/class/gpio/gpio53/direction out
write /sys/class/gpio/gpio53/value 0
#red
write /sys/class/gpio/export 54
chmod 0777 /sys/class/gpio/gpio54/
chmod 0777 /sys/class/gpio/gpio54/direction
chmod 0777 /sys/class/gpio/gpio54/value

write /sys/class/gpio/gpio54/direction out
```



```
write /sys/class/gpio/gpio54/value 0
#green
write /sys/class/gpio/export 55
chmod 0777 /sys/class/gpio/gpio55/
chmod 0777 /sys/class/gpio/gpio55/direction
chmod 0777 /sys/class/gpio/gpio55/value

write /sys/class/gpio/gpio55/direction out
write /sys/class/gpio/gpio55/value 0
```

5.2.3 用 ADB 测试 LED 节点方法

//在保证 ADB 可以正常连接调试情况下

例 1: 读取 LED 灯状态

```
$adb shell
$su
$cat /sys/class/gpio/gpio55/value
adb 会看到 电平返回值 0/1
```

例 2: 改变 LED 状态:

```
$ adb shell
$su
# echo /sys/class/gpio/gpio55/value 1 //拉高
```

/ echo 在 ADB 中为 输出 */*

/ cat 在 ADB 中为 读取 */*

5.2.4 用 JAVA 控制 LED 闪烁的方法:

```
public static final String RST_PIN = "/sys/class/gpio/gpio106/value";

public String led_flash() {

    try {

        BufferedWriter bufWriter = null;
```

```

        BufferedWriter bufWriter2 = null;

        bufWriter = new BufferedWriter(new FileWriter(RST_PIN));
        bufWriter2 = new BufferedWriter(new FileWriter(RST_PIN));

        bufWriter.write("1" + "");
        bufWriter.flush();
        bufWriter.close();
        delay(5000);
        bufWriter2.write("0" + "");
        bufWriter2.flush();
        bufWriter2.close();

        return "LED 闪烁控制正确";
    } catch (IOException e) {
        return " LED 闪烁控制错误: " + Log.getStackTraceString(e);
    }
}

private void delay(int ms) {
    try {
        Thread.currentThread();
        Thread.sleep(ms);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

```

```
}  
  
}
```

5.3 读取键盘值用于应用开发

如果需要读取键盘值：

```
$adb root
```

```
$adb remount
```

```
$adb shell am start -a rk.intent.action.startDevicetest
```

5.4 显示/隐藏导航栏

5.4.1 应用层

隐藏导航栏：

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

显示导航栏：

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

5.4.2 adb 测试显示/隐藏导航栏

//在保证 ADB 可以正常连接调试情况下

//显示导航栏

```
$ adb shell am broadcast -a com.bozz.show_nav_bar
```

```
//隐藏导航栏
```

```
$ adb shell am broadcast -a com.bozz.hide_nav_bar
```

5.5 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 */
```

5.6 系统签名获取系统权限的方法

```
java -Djava.library.path=. -jar signapk.jar platform.x509.pem platform.pk8
```

```
$APK_NAME.apk $APK_NAME.signed.apk
```

5.7 基于 Andorid 标准 API 进行应用开发适配

基于 Andorid 8.1 标准 SDK 进行开发

```
compileSdkVersion 28
```

```
targetSdkVersion 28
```

6.启动测试模式的方法:

6.1 启动老化测试:

1)如果需要进行老化测试: 需要将 Aging_Test.bin 和 Aging_Test_Video.mp4 都 push 到 /sdcard/目录下才能进行老化测试;

```
$adb root
```

```
$adb remount
```

```
$adb push Aging_Test_Video.mp4 /sdcard/
```

```
$adb push Aging_Test.bin /sdcard/
```

6.2 打开厂测的方法:

将 flag.txt 拷贝到 U 盘根目录, flag.txt 中的内容为: devices

6.3 打开压力测试的方法:

将 flag.txt.stress 拷贝到 U 盘根目录, 并改名为 flag.txt,其中的内容为: stress

或者使用命令:

```
$adb shell am start -a android.rk.intent.action.startStressTest
```

7.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: $0\sim 50^{\circ}\text{C}$;
- 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。