Air168 模块硬件设计手册 V1.02

上海合宙通信科技有限公司

版本号	修改记录	日期	作者
V1.01	新建	2018-11-14	Jinyi
V1.02	修改模块 flash 描述改为 32Mb	2019-9-17	Jinyi

目录

图表目录

表格目录

1. 绪论

本文档定义了Air168模块及其硬件接口规范，电气特性和机械细节，通过此文档的帮助，结合我们的应用手册和用户指导书，客户可以快速应用Air168模块于各种应用的二次开发。

1.1. 相关文档

表格 1：相关文档

编号	文件名	注释
1	Air168 参考设计	已开放
2	Air168 封装库	已开放
3	Luat 专用看门狗芯片设计手册	已开放
4	PCB 天线设计封装	已开放
5	Luat 模块阻抗线及天线设计建议	已开放
6	上层软件 luaTask (Demo + Lib)	已开放
7	底层软件 CORE V00xx	已开放

注意：所有文档均可在 www.openluat.com 下载

2. 综述

Air168模块是一款带SPI摄像头接口的大容量flash MCU模块。

Air168 最大支持SPI 3M像素摄像头，内置32MbNorFlash + 32MbSRAM，并支持合宙特有的Luat开源平台，方便客户做二次开发，极大的减少了客户的开发周期和成本。

Air168 有丰富的外围接口，支持UART，SPI，I2C等各种接口，可支持最多35个GPIO，并支持ADC，5*5阵列键盘，专用LED以及背光驱动管脚，音频输入和输出功能，满足各种应用场景的使用要求。

Air168是贴片式模块，采用LCC封装，可以通过其管脚焊盘内嵌于客户应用中，提供了模块与客户主板间丰富的硬件接口。

Air168模块完全符合RoHS标准。

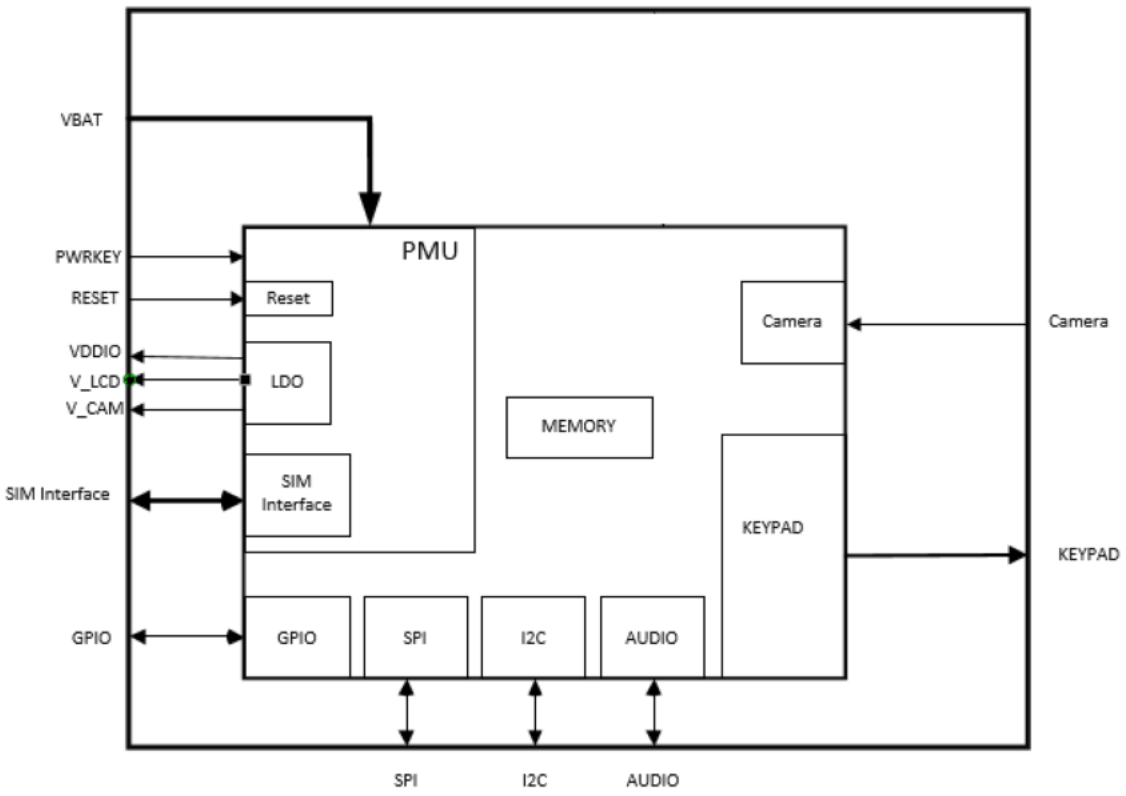
表格 2：模块主要特征

特征	说明
供电	♦ VBAT 3.4V ~ 4.2V，典型值3.8V
省电模式耗流	♦ 0.9mA
CPU 特性	♦ 32-bit XCPU ♦ RISC指令集 ♦ 312Mhz ♦ 4k 指令缓存和4k 数据缓存
存储器特性	♦ 32Mb 1.8V SPI NOR Flash ♦ 32Mb 1.8V DDR PSRAM
温度范围	♦ 正常工作温度：-40°C ~ +85°C ♦ 存储温度：-45°C ~ +90°C
音频接口	支持录音以及播放功能，可直接驱动8欧姆喇叭
外围接口	● 两个通用串口，支持软硬件流控；1个调试串口。 ● 3个I2C接口 ● 两个标准SPI接口，支持3线/4线模式；1个LCD专用SPI接口；1个camera专用SPI接口 ● 2个ADC接口 ● 5*5键盘整列 ● 3个LED驱动管脚
尺寸	22.0±0.15 × 22.0±0.15 ×2.3±0.2mm 重量：4g

2.1 功能图

下图为Air168功能框图，阐述了其主要功能：

- 存储器
- 电源管理
- 接口部分



图表 1：功能框图

2.2 评估板

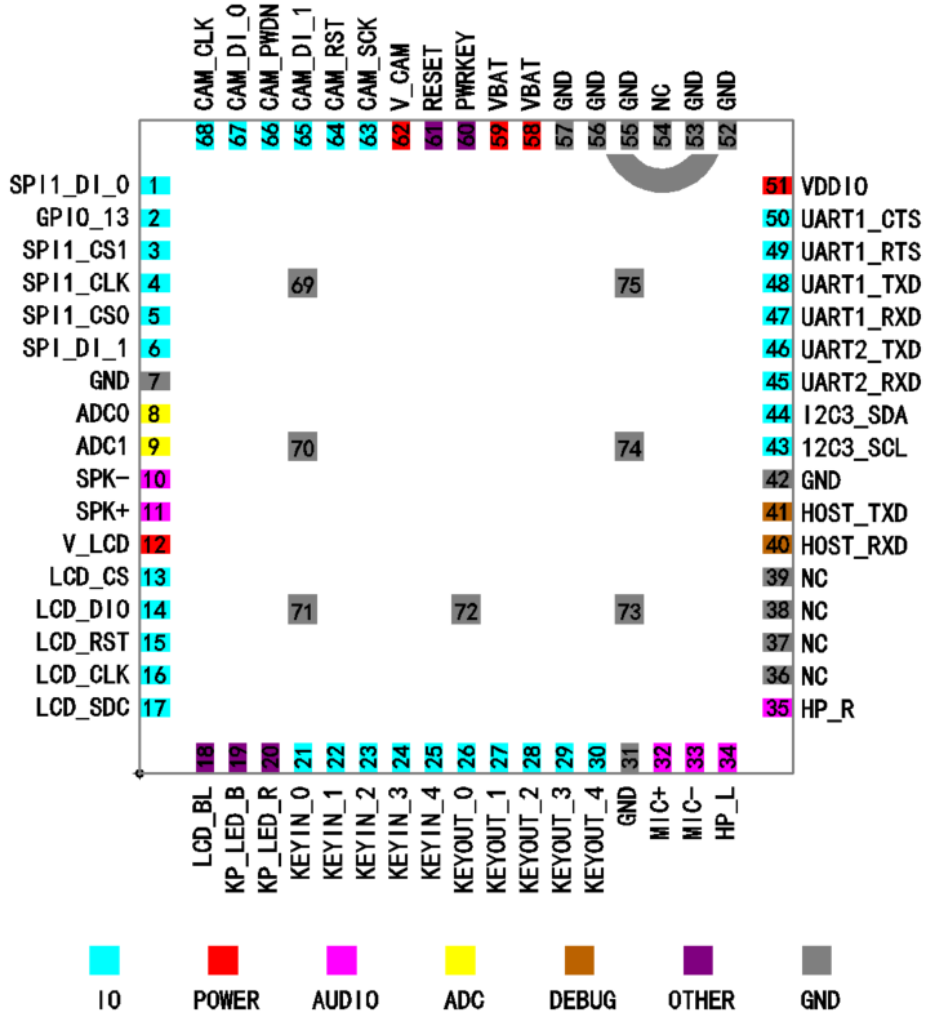
为了有助于测试及使用Air168，合宙提供一套评估板。评估板包括Air168模块、EVB_Air168、UART转USB线等。

3 应用接口

模块采用LCC封装，38个SMT焊盘管脚，后续章节详细阐述了以下接口的功能：

- 电源供电
- 开关机

3.1 管脚排列图（top view）



TOP VIEW

表格 3：Air168 管脚分配

管脚号	管脚名	输入/输出	管脚号	管脚名	输入/输出
1	SPI1_DI_0/GPIO_11	I/O	31	GND	
2	GPIO_13	I/O	32	MIC+	I
3	SPI1_CS1/GPIO_10	I/O	33	MIC-	I
4	SPI1_CLK/GPIO_8	I/O	34	HP_L	O
5	SPI1_CS0/GPIO_9	I/O	35	HP_R	O
6	SPI_DI_1/GPIO_12	I/O	36	NC	
7	GND		37	NC	
8	ADC0	I	38	NC	
9	ADC1	I	39	NC	
10	SPK-	O	40	HOST_RXD	I
11	SPK+	O	41	HOST_TXD	O
12	V_LCD	O	42	GND	
13	LCD_CS/GPIO_15	I/O	43	UART2_RTS /GPIO_6	I/O
14	LCD_DIO/GPIO_17	I/O	44	UART2_CTS/GPIO_7	I/O
15	LCD_RST/GPIO_14	I/O	45	UART2_RXD/GPIO_4	I/O
16	LCD_CLK/GPIO_16	I/O	46	UART2_TXD/GPIO_5	I/O
17	LCD_SDC/GPIO_18	I/O	47	UART1_RXD/GPIO_0	I/O
18	LCD_BL	I	48	UART1_TXD/GPIO_1	I/O
19	KP_LED_B	I	49	UART1_RTS/GPIO_2	I/O
20	KP_LED_R	I	50	UART1_CTS/GPIO_3	I/O
21	KEYIN_0/GPIO_25	I/O	51	VDDIO	O
22	KEYIN_1/GPIO_26	I/O	52	GND	
23	KEYIN_2/GPIO_27	I/O	53	GND	
24	KEYIN_3/GPIO_28	I/O	54	NC	
25	KEYIN_4/GPIO_29	I/O	55	GND	
26	KEYOUT_0/GPIO_30	I/O	56	GND	
27	KEYOUT_1/GPIO_31	I/O	57	GND	

28	KEYOUT_2/GPIO_32	I/O	58	VBAT	I
29	KEYOUT_3/GPIO_33	I/O	59	VBAT	I
30	KEYOUT_4/GPIO_34	I/O	60	PWRKEY	I
管脚号	管脚名	输入/输出	管脚号	管脚名	输入/输出
61	RESET	I	69	GND	
62	V_CAM	O	70	GND	
63	CAM_SCK/GPIO_22	I/O	71	GND	
64	CAM_RST/GPIO_20	I/O	72	GND	
65	CAM_DI_1/GPIO_24	I/O	73	GND	
66	CAM_PWDN/GPIO_19	I/O	74	GND	
67	CAM_DI_0/GPIO_23	I/O	75	GND	
68	CAM_CLK/GPIO_21	I/O			

表格4：管脚描述与配置

Air168 PINNO.	GPIO number	Name.	Power Domain	Pull-Up	Pull-Up/Down Resistor	At Reset			After Reset			Function0				Function1				Function2				Function3			
						H/L/Hiz	pull	PinState	H/L/Hiz	pull	PinState	Function0	Type	PULL	Description	Function1	Type	PULL	Description	Function2	Type	PULL	Description	Function3	Type	PULL	Description
47	GPIO0	UART1_RXD/GPIO_0	VDDIO 2.8V	2.8V,	166K	L	DN	INPUT	L	DN	INPUT	GPIO_0	I/O	DN	通用 GPIO	UART1_RXD	I	UP	串口数据接收	SPI2_CLK	I/O	DN	SPI 时钟信号				
48	GPIO1	UART1_TXD/GPIO_1		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	GPIO_1	I/O	DN	通用 GPIO	UART1_TXD	O	OFF	串口数据发送	SPI2_CS_0	I/O	UP	SPI 片选信号 0				
49	GPIO2	UART1_RTS/GPIO_2		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	GPIO_2	I/O	DN	通用 GPIO	UART1_RTS	I	UP	串口清除发送	SPI2_CS_1	I/O	UP	SPI 片选信号 1				
50	GPIO_3	UART1_CTS/GPIO_3		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	GPIO_3	I/O	DN	通用 GPIO	UART1_CTS	O	OFF	串口请求发送	SPI2_DI_0	I/O	DN	SPI 数据输入/输出				
45	GPIO_4	UART2_RXD/GPIO_4		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	GPIO_4	I/O	DN	通用 GPIO	UART2_RXD	I	UP	串口数据接收	SPI2_DI_1	I	DN	SPI 数据输入	LPG	O	OFF	低速 PWM 输出
46	GPIO_5	UART2_TXD/GPIO_5		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	GPIO_5	I/O	DN	通用 GPIO	UART2_TXD	O	OFF	串口数据发送	PWT	O	OFF	高速 PWM 输出				
43	GPIO_6	UART2_RTS/GPIO_6		2.8V,	33K	L	DN	INPUT	L	DN	INPUT	GPIO_6	I/O	DN	通用 GPIO	UART2_RTS	I	UP	串口清除发送	I2C3_SCL	I/O	UP	I2C 时钟信号				
44	GPIO_7	UART2_CTS/GPIO_7		2.8V,	33K	L	DN	INPUT	L	DN	INPUT	GPIO_7	I/O	DN	通用 GPIO	UART2_CTS	O	OFF	串口请求发送	I2C3_SDA	I/O	UP	I2C 数据信号				
4	GPIO_8	SPI1_CLK/GPIO_8	V_MMC 1.8/1.9/2.0/2.6/2.8/3.0/3.3V	1.9V,	166K	L	OFF	OUTPUT	26MHz	OFF	OUTPUT	~~SDMMC_CLK~~	~~O~~	~~OFF~~	~~SD 时钟信号~~	GPIO_8	I/O	DN	通用 GPIO	SPI1_CLK	I/O	DN	SPI 时钟信号	I2C1_SCL	I/O	UP	I2C 时钟信号
5	GPIO_9	SPI1_CS0/GPIO_9		1.9V,	50K	H	UP	INPUT	H	UP	INPUT	~~SDMMC_CMD~~	~~I/O~~	~~UP~~	~~SD 命令信号~~	GPIO_9	I/O	DN	通用 GPIO	SPI1_CS_0	I/O	UP	SPI 片选信号 0	I2C1_SDA	I/O	UP	I2C 数据信号
3	GPIO_10	SPI1_CS1/GPIO_10		1.9V,	50K	H	UP	INPUT	H	UP	INPUT	~~SDMMC_DATA_0~~	~~I/O~~	~~UP~~	~~SD 数据信号 0~~	GPIO_10	I/O	DN	通用 GPIO	SPI1_CS_1	I/O	UP	SPI 片选信号 1				
1	GPIO_11	SPI1_DI_0/GPIO_11		1.9V,	50K	H	UP	INPUT	H	UP	INPUT	~~SDMMC_DATA_1~~	~~I/O~~	~~UP~~	~~SD 数据信号 1~~	GPIO_11	I/O	DN	通用 GPIO	SPI1_DI_0	I/O	DN	SPI 数据输入/输出				
6	GPIO_12	SPI1_DI_1/GPIO_12		1.9V,	50K	H	UP	INPUT	H	UP	INPUT	~~SDMMC_DATA_2~~	~~I/O~~	~~UP~~	~~SD 数据信号 2~~	GPIO_12	I/O	DN	通用 GPIO	SPI1_DI_1	I	DN	SPI 数据输入				
2	GPIO_13	GPIO_13		1.9V,	50K	H	UP	INPUT	H	UP	INPUT	~~SDMMC_DATA_3~~	~~I/O~~	~~UP~~	~~SD 数据信号 3~~	GPIO_13	I/O	DN	通用 GPIO								
15	GPIO_14	LCD_RST/GPIO_14	V_LCD 1.8/2.8V	1.9V,	166K	H	OFF	OUTPUT	H	OFF	OUTPUT	LCD_RSTB	O	OFF	LCD 复位信号	GPIO_14	I/O	DN	通用 GPIO	DAI_RST	I	DN	DAI 复位信号	I2S_BCK	I/O	DN	串行数据时钟
13	GPIO_15	LCD_CS/GPIO_15		1.9V,	166K	H	OFF	OUTPUT	H	OFF	OUTPUT	SPI_LCD_CS	O	OFF	LCD 片选信号	GPIO_15	I/O	DN	通用 GPIO	DAI_CLK	O	OFF	DAI 时钟信号	I2S_LRCK	I/O	DN	左右声道选择
16	GPIO_16	LCD_CLK/GPIO_16		1.9V,	166K	H	OFF	OUTPUT	H	OFF	OUTPUT	SPI_LCD_SCK	O	OFF	LCD 时钟信号	GPIO_16	I/O	DN	通用 GPIO	DAI_DI	I	DN	DAI 数据输入	I2S_DI	I	DN	数字语音输入
14	GPIO_17	LCD_DIO/GPIO_17		1.9V,	166K	L	DN	OUTPUT	L	DN	OUTPUT	SPI_LCD_DIO	I/O	DN	LCD 数据信号	GPIO_17	I/O	DN	通用 GPIO	DAI_DO	O	OFF	DAI 数据输出	I2S_DO	O	OFF	数字语音输出
17	GPIO_18	LCD_SDC/GPIO_18		1.9V,	166K	H	OFF	OUTPUT	H	OFF	OUTPUT	SPI_LCD_SDC	O	OFF	LCD 数据命令选择信号	GPIO_18	I/O	DN	通用 GPIO								
66	GPIO_19	CAM_PWDN/GPIO_19	V_CAM 1.8/2.8V	1.9V,	33K	H	OFF	OUTPUT	H	OFF	OUTPUT	CAM_PWDN	O	OFF	Camera 使能信号	GPIO_19	I/O	DN	通用 GPIO	I2C2_SCL	I/O	UP	I2C 时钟信号				
64	GPIO_20	CAM_RST/GPIO_20		1.9V,	33K	L	OFF	OUTPUT	L	OFF	OUTPUT	CAM_RSTB	O	OFF	Camera 复位信号	GPIO_20	I/O	DN	通用 GPIO	I2C2_SDA	I/O	UP	I2C 数据信号				
68	GPIO_21	CAM_CLK/GPIO_21		1.9V,	166K	L	OFF	OUTPUT	L	OFF	OUTPUT	CAM_CLK	O	OFF	Camera 时钟输出	GPIO_21	I/O	DN	通用 GPIO								
63	GPIO_22	CAM_SCK/GPIO_22		1.9V,	166K	H	DN	OUTPUT	H	DN	OUTPUT	SPI_CAM_SCK	I/O	DN	SPI 时钟信号	GPIO_22	I/O	DN	通用 GPIO								
67	GPIO_23	CAM_DI_0/GPIO_23		1.9V,	166K	L	DN	INPUT	L	DN	INPUT	SPI_CAM_DI_0	I	DN	SPI 数据输入 0	GPIO_23	I/O	DN	通用 GPIO	SPI_CAM_DI_1	I	DN	SPI 数据输入 1				
65	GPIO_24	CAM_DI_1/GPIO_24		1.9V,	166K	L	DN	INPUT	L	DN	INPUT	SPI_CAM_DI_1	I	DN	SPI 数据输入 1	GPIO_24	I/O	DN	通用 GPIO	SPI_CAM_DI_0	I	DN	SPI 数据输入 0	SPI_CAM_SSN	I/O	DN	Camera SSN 输出
21	GPIO_25	KEYIN_0/GPIO_25	VDDIO 2.8V	2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYIN_0	I	DN	键盘扫描输入	GPIO_25	I/O	DN	通用 GPIO								
22	*GPIO_26	KEYIN_1/GPIO_26		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYIN_1	I	DN	键盘扫描输入	GPIO_26	I/O	DN	通用 GPIO								
23	*GPIO_27	KEYIN_2/GPIO_27		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYIN_2	I	DN	键盘扫描输入	GPIO_27	I/O	DN	通用 GPIO								
24	*GPIO_28	KEYIN_3/GPIO_28		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYIN_3	I	DN	键盘扫描输入	GPIO_28	I/O	DN	通用 GPIO								
25	GPIO_29	KEYIN_4/GPIO_29		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYIN_4	I	DN	键盘扫描输入	GPIO_29	I/O	DN	通用 GPIO								
26	GPIO_30	KEYOUT_0/GPIO_30		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYOUT_0	I/O	DN	键盘扫描输出	GPIO_30	I/O	DN	通用 GPIO								
27	GPIO_31	KEYOUT_1/GPIO_31		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYOUT_1	I/O	DN	键盘扫描输出	GPIO_31	I/O	DN	通用 GPIO								
28	GPIO_32	KEYOUT_2/GPIO_32		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYOUT_2	I/O	DN	键盘扫描输出	GPIO_32	I/O	DN	通用 GPIO								
29	GPIO_33	KEYOUT_3/GPIO_33		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYOUT_3	I/O	DN	键盘扫描输出	GPIO_33	I/O	DN	通用 GPIO					DBG_CLK(32Kclock out)	O	OFF	32K 时钟输出
30	GPIO_34	KEYOUT_4/GPIO_34		2.8V,	166K	L	DN	INPUT	L	DN	INPUT	KEYOUT_4	I/O	DN	键盘扫描输出	GPIO_34	I/O	DN	通用 GPIO					HST_CLK	I	DN	调试串时钟输入
40	GPIO_35	HSOT_RXD/GPIO_35	VDDIO 2.8V	2.8V,	166K	H	UP	INPUT	H	UP	INPUT	HST_RXD	I	UP	调试串口输入	GPIO_35	I/O	DN	通用 GPIO								
41	GPIO_36	HSOT_TXD/GPIO_36		2.8V	166K	H	UP	OUTPUT	H	UP	OUTPUT	HST_TXD	I/O	UP	调试串口输出	GPIO_36	I/O	DN	通用 GPIO								
40	ADC0		I	数模转换输入 0,输入量程 0~1.85V																							

41	ADC1	I	数模转换输入 1,输入量程 0~1.85V
10	SPK-	O	语音差分信号输出-，内置音频功放 classAB
11	SPK+	O	语音差分信号输出+，内置音频功放 classAB
32	MIC+	I	语音差分信号输入+，内置 mic 偏置
33	MIC-	I	语音差分信号输入-，内置 mic 偏置
34	HP_L	O	立体声耳机左声道
35	HP_R	O	立体声耳机右声道
39	VDDIO	O	2.8V 参考电平输出（不可配置）,输出能力 100mA
38	V_LCD	O	LCD 供电输出,（电压可配置）,输出能力 200mA
37	V_CAM	O	camera 供电输出,（电压可配置）,输出能力 200mA
18	LCD_BL	I	LDC 背光控制，电流源，最大驱动能力 60mA
19	KP_LED_B	I	LED 控制，电流源，最大驱动能力 30mA
20	KP_LED_R	I	LED 控制，电流源，最大驱动能力 30mA
60	PWRKEY	I	开机关机控制，低电平有效
61	RESET	I	硬件关机控制，低电平有效
59	VBAT	I	模块主电源输入，工作电压 3.4~4.2V
58	VBAT		

*注：GPIO26,GPIO27,GPIO28在开机过程中禁止上下拉，不然会进入特殊的BOOT模式

3.2 工作模式

下表简要的叙述了接下来几章提到的各种工作模式。

表格 5：工作模式

模式	功能	
正常工作	SLEEP	CPU进入休眠，串口关闭，进入低功耗状态。默认状态下，如果没有任务进行，会自动进入休眠，可以通过GPIO中断，定时器唤醒，网络数据唤醒，来电唤醒。 如果lua脚本中调用pm.wake()则不会进入休眠。
	NORMAL	软件正常运行。
关机模式	通过调用rtos.poweroff()进入关机流程或者使用“RESET”管脚来实现正常关机。电源管理芯片关断基带部分的供电，并且只保留RTC供电。软件不运行，串口无法访问。保持VBAT电源供电。	

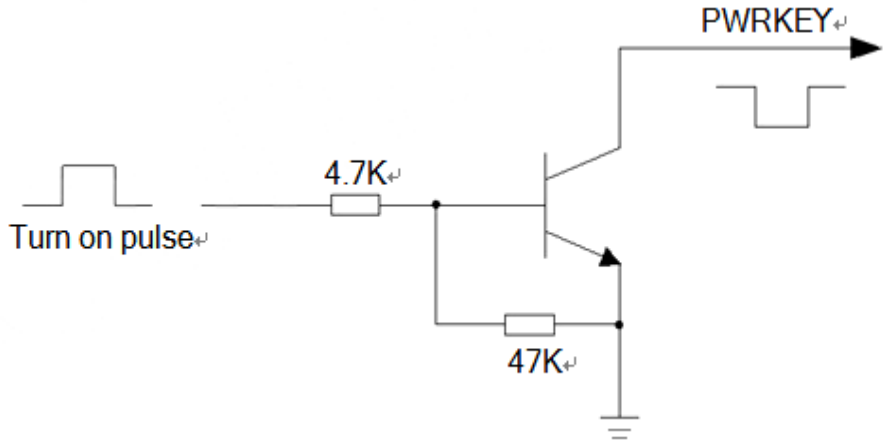
3.4 开关机

3.4.1 开机

Air168模块可以通过PWRKEY管脚开机。关机状态下长按开机键2s以上，模块会进入开机流程，软件会检测VBAT管脚电压若VBAT管脚电压大于软件设置的开机电压（默认3.55V），会继续开机动作直至系统开机完成；否则，会停止执行开机动作，系统会关机。

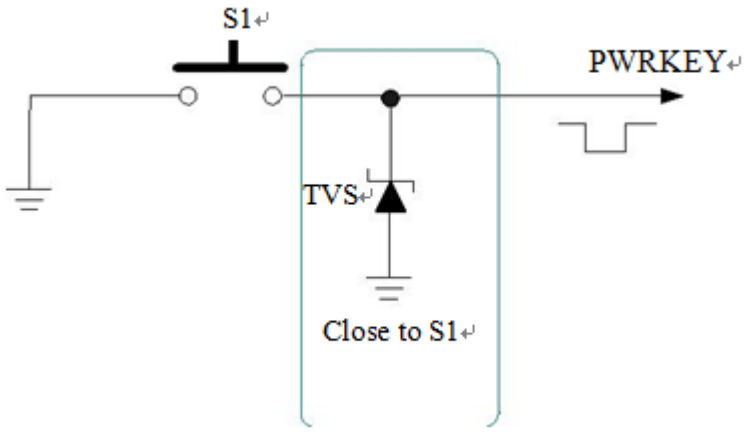
3.4.1.1 PWRKEY 管脚开机

VBAT上电后，PWRKEY管脚可以启动模块，把PWRKEY管脚拉低持续1s之后开机，开机成功后PWRKEY管脚可以释放。可以通过检测VDDIO管脚的电平来判别模块是否开机。推荐使用开集驱动电路来控制PWRKEY管脚。下图为参考电路：



图表 2：开集驱动参考开机电路

另一种控制PWRKEY管脚的方法是直接使用一个按钮开关。按钮附近需放置一个 TVS管用以ESD保护。下图为参考电路：



图表 3：按键开机参考电路

3.4.1.2 上电开机

将 PWRKEY 管脚直接接地可以实现上电自动开机功能。需要注意，在上电开机模式下，将无法关机，只要 VBAT 管脚的电压大于开机电压即使软件调用关机接口，模块仍然会再开机起来。另外，在此模式下，要想成功开机起来 VBAT 管脚电压任然要大于软件设定的开机电压值，如果不满足，模块会关闭，就会出现反复开关机的情况。

3.4.2 关机

以下的方式可以关闭模块：

- 正常关机：调用rtos.pweroff()软关机。
- 低压自动关机：模块检测到低压时关机。
- 紧急关机：通过RESET管脚关机。

3.4.2.1 正常关机

在 lua 正在运行时可以通过调用 rtos.pweroff()进行软关机。

表格 6：lua 关机接口

rtos.poweron()	软件关机
语法	rtos.poweron()
参数	无

返回值	无

可以通过 VDDIO 管脚电压来判断是否关机完成，如果 VDDIO 的管脚电压为接近 0V 则说明关机完成

3.4.2.2 低电压自动关机

模块在运行状态时当 VBAT 管脚电压低于软件设定的关机电压时（默认设置 3.4V），软件会执行关机动作关闭模块，以防低电压状态下运行出现各种异常。

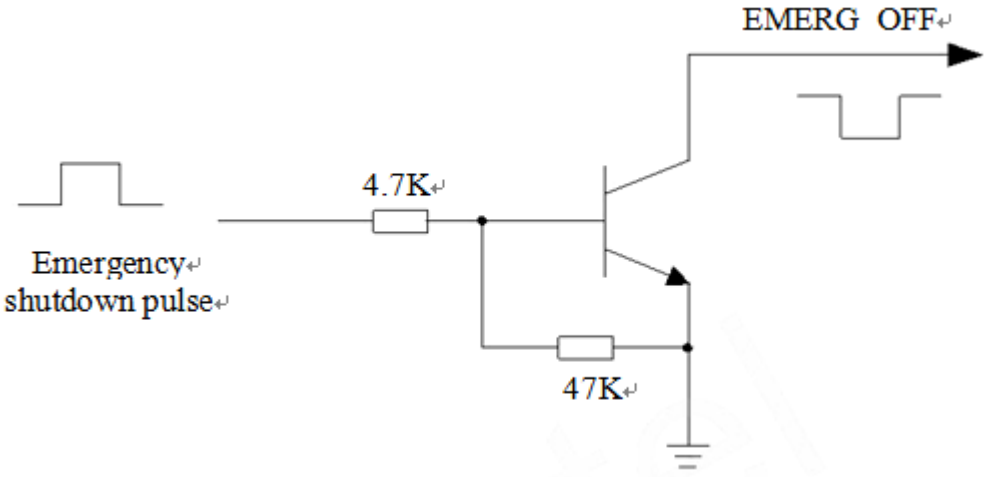
注意：对于 PWRKEY 接地的上电开机模式，在 VBAT 电压低于关机电压时，模块仍然会执行关机动作，但是关机后会因为硬件上电开机而会再次被开机起来，由于此时 VBAT 电压在开机电压以下，软件又会执行关闭动作，因而会出现反复开关机的现象，直到 VBAT 电压低于硬件开机电压 3.0V，硬件不再开机起来为止。

3.4.2.3 RESET 紧急关机

Air168模块第60脚为reset管脚，其功能是硬件关机。

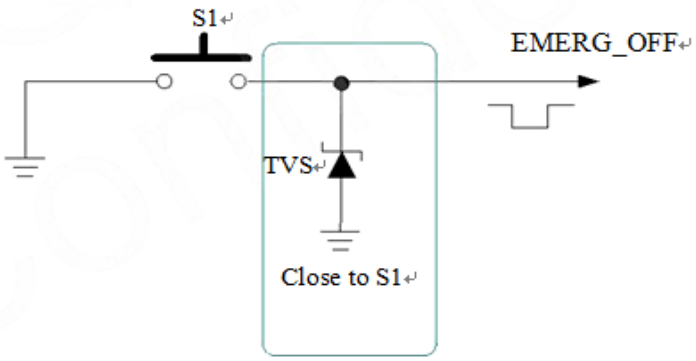
可以通过拉低RESET管脚200ms左右来关机，之后释放。推荐使用OC驱动电路来控制RESET管脚。下图为参考电路：

注意：三级管集电极和发射级不能串联大于1欧姆的电阻。



图表 4：开集驱动参考紧急关机电路

另一种控制 RESET 管脚紧急关机的方法是直接使用一个按钮开关。按钮附近需放置一个 TVS 管用以 ESD 保护。下图为参考电路：



图表 5：按键紧急关机参考电路

注意：上电开机配置情况下，RESET 管脚拉低后，模块会关机后又会上电开机起来，RESET 管脚间接起到重启的作用。

3.5 省电技术

根据系统需求，模块可以进入两种低功耗模式。睡眠模式和飞行模式

3.5.1 飞行模式

飞行模式可以将模块功能减少到最小程度，此模式下内部射频部分和 SIM 卡部分的功能将会关闭（尽管模块并未引出相关引脚，关闭这部分功能任然能降低耗流）。而串口依然有效。此模式可以通过调用 net.switchFly()进入或者退出飞行模式。

表格 7：lua 飞行模式接口

net.switchFly()	设置飞行模式
语法	net.switchFly(mode)
参数	mode：bool 值，true：飞行模式开，飞行模式关
返回值	无

3.5.2 睡眠模式

Air168 支持睡眠模式，默认状态下系统是会根据系统的繁忙情况自动的进入或退出睡眠模式，一般情况下不需要写代码去控制进入或者退出休眠状态。但是对于禁止进入休眠的应用需求，可以调用 pm.wake()让系统一直不进休眠，调用此接口后如果需要进入休眠，可以调用 pm.sleep().

表格 8：lua 唤醒模块接口

pm.wake()	某个 Lua 应用唤醒系统
语法	pm.wake(tag)
参数	tag：sting 类型，某个 Lua 应用的唤醒标记，用户自定义
返回值	无
备注	如果不是故意控制的不休眠，一定要保证 pm.wake("A")了，有地方去调用 pm.sleep("A")

3.5.3 睡眠唤醒

当模块处于睡眠模式，以下方法可以唤醒模块。

- 调用pm.sleep()接口

表格 9：lua 睡眠接口

pm.sleep()	某个 Lua 应用休眠系统
语法	pm.sleep(tag)
参数	tag：sting 类型，某个 Lua 应用的唤醒标记，用户自定义，跟 wake 中的标记保持一致
返回值	无

- GPIO中断。
- 定时器

3.5.4 模式切换汇总

表格 10：模式切换汇总

当前模式	下一模式		
	关机	正常模式	睡眠模式
关机		使用 PWRKEY 开机	
正常模式	调用 rtos.poweroff()，或使用 RESET 管脚，或 VBAT 电压低于关机电压		自动休眠或者软件调用睡眠接口

睡眠模式	调用 rtos.pweroff()或 RESET 管脚或 VBAT 电压低于关机电压	调用 pm.wake(),GPIO 管脚中断、定时器	

3.6. 串口

Air168 模块支持 3 路 TTL2.8V 串口： UART1， UART2 和下载调试串口 HOST_UART。模块支持范围 1200bps 到 921600bps。其中 UART1 和 UART2 不分主次功能相同，都支持硬件流控（软件版本暂不支持硬件流控功能）。下载调试串口 HOST_UART 内置特殊的通信协议，只能用于下载和调试无法用于数据通信，固定波特率 921600bps。

串口逻辑电平如下表所示：

表格 11：串口逻辑电平

参数	最小值	最大值	单位
V_{IL}	0	0.25×VDDIO	V
V_{IH}	0.75×VDDIO	VDDIO +0.3	V
V_{OL}	0	0.15×VDDIO	V
V_{OH}	0.85×VDDIO	VDDIO	V

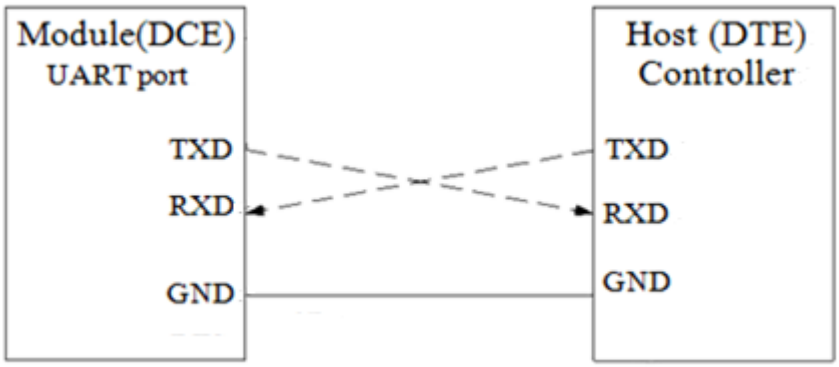
表格 12：串口管脚定义

接口	名称	管脚	作用
主串口 UART1	UART1_TXD	48	串口发送数据
	UART1_RXD	47	串口接收数据
	UART1_CTS	50	清除发送（lua 暂时不支持）
	UART1_RTS	49	DTE 请求发送数据（lua 暂时不支持）
辅串口 UART2	UART2_RXD	45	串口接收数据
	UART2_TXD	46	串口发送数据
	UART2_CTS	44	清除发送（lua 暂时不支持）
	UART2_RTS	43	DTE 请求发送数据（lua 暂时不支持）

3.6.1. 串口连接方式

3.6.1.1 三线串口连接方式

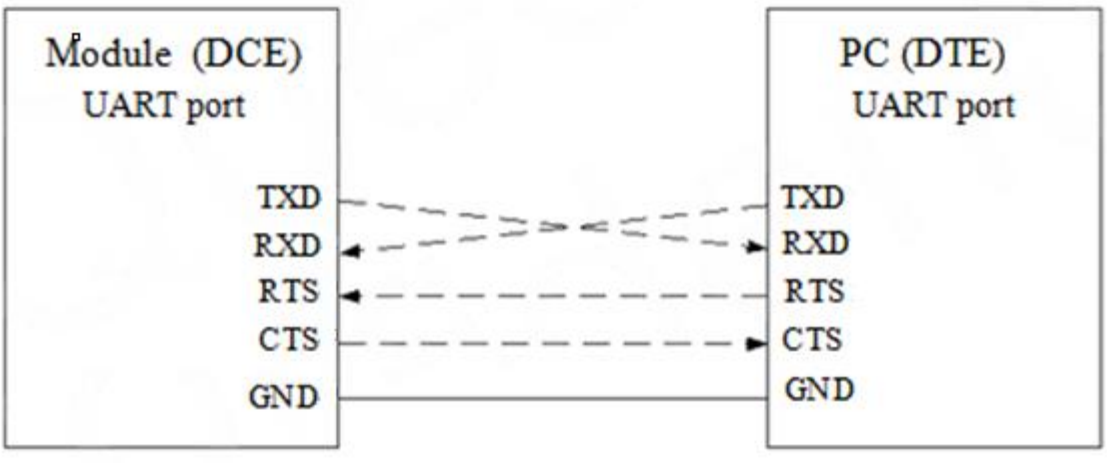
三线串口是最常用的连接方式，与外设的连接只需要到TX,RX,GND三根线，UART1/UART2/HOST都可以用此方式连接，连接方式如下：



图表 6：串口三线制连接方式示意图

3.6.1.2 带硬件流控功能的串口连接方式（lua 暂不支持）

对于串口传输比较大的数据时，建议使用硬件流控功能以确保可靠传输。使能流控功能时连接方式如下：

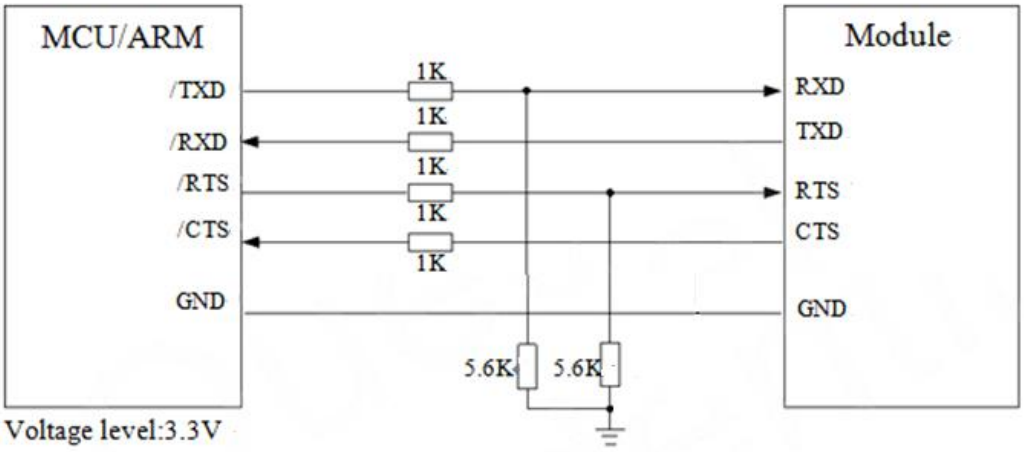


图表 7：全功能串口连接方式示意图

3.6.1.3 3.3V 串口连接方式

对于 3.3V 电压系统情况下的串口电平匹配电路参考如下，强烈建议在 RXD 等模块输入的端口上使用分压电阻的方式，将电压分压到 2.8V。

如果是 3V 系统，根据分压原理建议将 5.6K 电阻改为 10K 电阻。客户不允许分压方式下，也建议必须串接 1K 电阻。

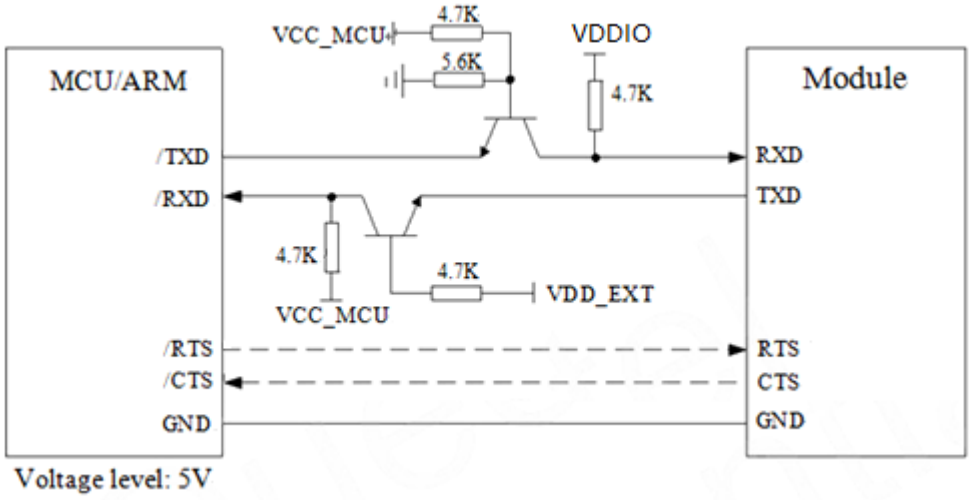


图表 8：3.3V 电平转换电路

3.6.1.4 5V 串口连接方式

5V 系统的电平匹配，模块和外设之间的电平匹配可以参考如下的连接方式，如下的虚线部分可以参考上面的实线电路（虚线部分模块发送参考模块 TXD 的电路设计，虚线部分模块接收参考模块 RXD 的电路设计）。

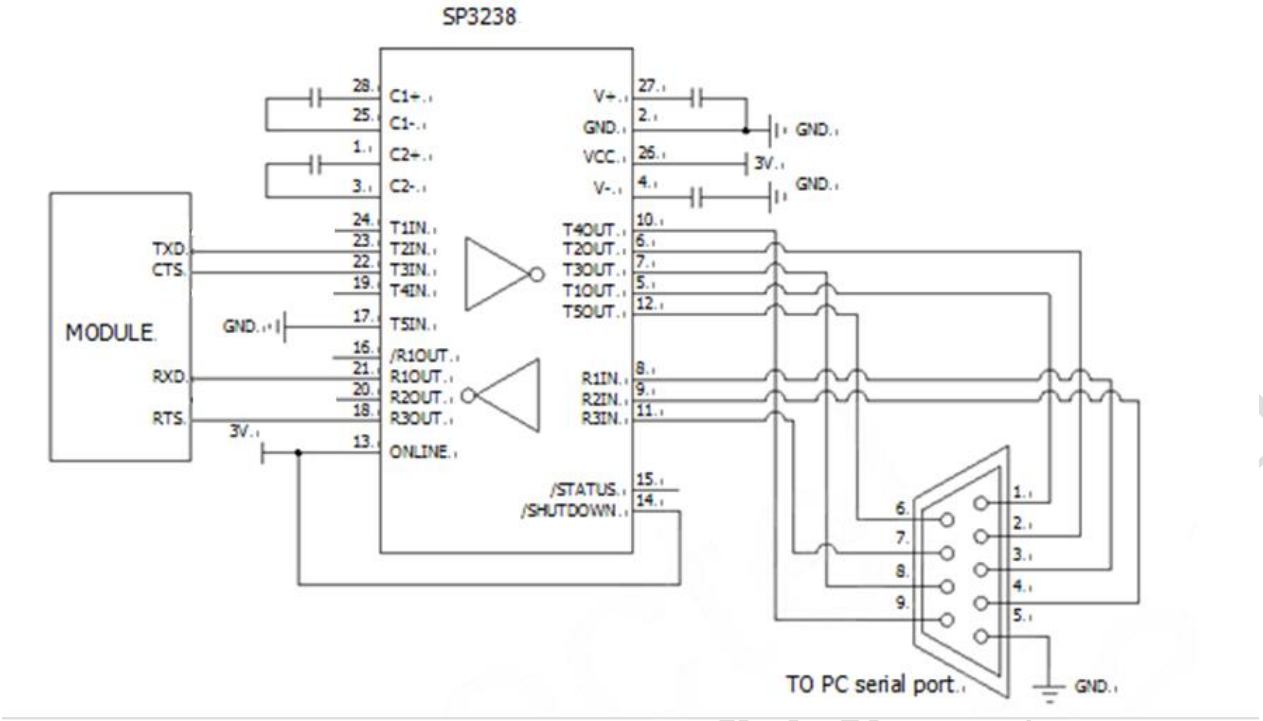
VCC_MCU 是客户端的 I/O 电平电压。VDDIO 是模块输出的 I/O 电平电压。



图表 9：5V 电平转换电路

3.6.1.5 模块串口与 RS232 串口连接方式

当模块和 PC 机进行通信时，需要在他们之间加 RS232 电平转换电路。因为模块的串口配置都不是 RS232 电平，仅支持 CMOS 电平。下图为模块跟 PC 通信时，串口电平转换电路。



图表 10：RS232 电平转换电路

3.6.2. 串口相关 API 接口

Lua 中关于使用先要通过 uart.setup()来进行串口参数的配置和初始化，之后可以通过 uart.write()向串口输出数据或者通过 uart.read()来从串口读取数据。具体串口使用方法请参考 script_LuaTask/demo/uart

表格 13：lua 串口设置接口

uart.setup()	Uart 串口设置	
语法	baud = uart.setup(id, baud, databits, parity, stopbits,[msgmode,txDoneReport])	
参数	Id	串口号，可选 0,1,2
	baud	波特率，可选 1200，2400，4800，9600，10400，14400，19200，28800，38400，57600，115200，230400，460800，921600
	databits	数据位，7 或 8
	parity	校验位，可选 uart.PAR_EVEN，uart.PAR_ODD 或 uart.PAR_NONE
	stopbits	停止位，可选 uart.STOP_1，uart.STOP_2
	msgmode	0 或者默认 - 消息通知，1 - 无消息上报需要用户主动轮询
	txDoneReport	txdone 消息上报开关。0：关闭，1：打开

返回值	串口的真实波特率

表格 14：lua 串口写接口

uart.write()	向串口写字符串或者整型数据	
语法	uart.write(id, data1, [data2], ..., [datan])	
参数	id	串口号，可选 0,1,2
	data1	第一个字符串或 8 位整型数据
	data2	第二个字符串或 8 位整型数据
	datan(可选)	第 n 个字符串或 8 位整型数据
返回值	无	

表格 15：lua 串口读接口

uart.read()	从串口读取字符串	
语法	str = uart.read(id, format)	
参数	id	串口号
	格式化	*l：读取到结束字符\n 或者阻塞发送
		*n：读取整型数据
		*s：读取到空格字符
		数字，number 类型：只读取 number 长度的数据
返回值	串口读出到的数据	

3.7. I2c 接口

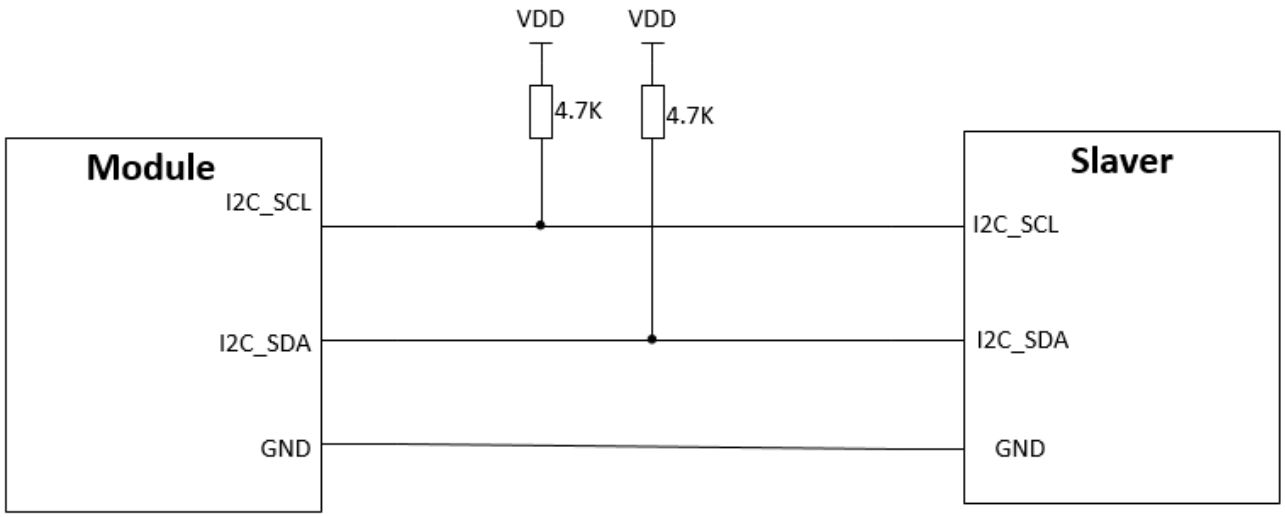
Air168 模块提供 3 个 I2C 接口，速率支持 FAST(400KHz)，SLOW（100KHz）。外设地址支持 0x00-0x7f。3 路 I2C 分别是有由如下管脚复用：

表格 16：I2C 接口管脚定义

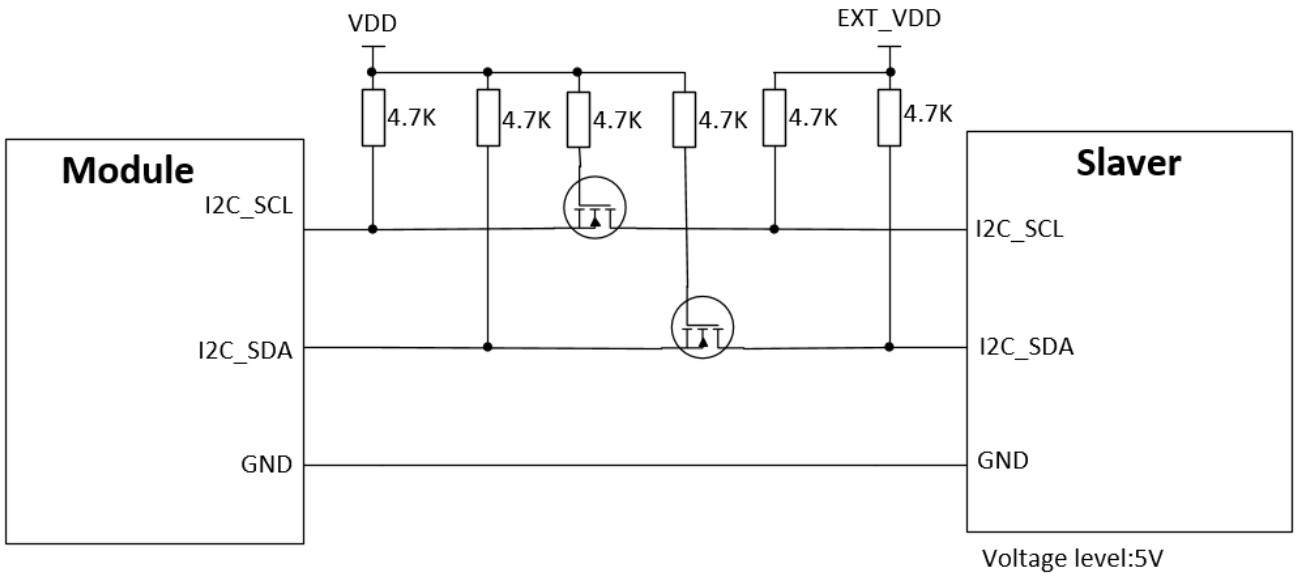
接口	名称	管脚	作用	电压域
I2C1	SPI1_CLK(I2C1_SCL)	4	I2C 时钟信号	V_MMC
	SPI1_CS_0(I2C1_SDA)	5	I2C 数据信号	
I2C2	CAM_PWDN(I2C2_SCL)	19	I2C 时钟信号	V_CAM
	CAM_RST(I2C2_SDA)	20	I2C 数据信号	
I2C3	UART2_RTS(I2C3_SCL)	43	I2C 时钟信号	VDDIO
	UART2_CTS(I2C3_SDA)	44	I2C 数据信号	

3.7.1. IC 接口连接方式

如下是 I2C 接口的参考线路，注意模块 I2C 只能做主设备，外部需要上拉，在配置 FAST 速率时，上拉电阻阻值不要大于 4.7K。三路 ADC 分别属于不同的电压域，使用时要注意打开对应的电压域。I2C1 和 I2C3 要用 VDDIO 上拉，而 I2C2 需要用 V_CAM 上拉。



图表 11：I2C 连接参考线路



图表 12：I2C 5V 外设连接参考线路

3.7.2. I2C 相关 API 接口

使用 I2C 接口的一般步骤是先用 i2c.setup()初始化 I2C 接口,之后就可以用 i2c. write ()向从设备写数据以及用 i2c.read()从从设备读取数据。具体使用方法请参考 script_LuaTask/demo/i2c

表格 17：lua I2C 初始化接口

i2c.setup()	打开并且配置 I2C 接口	
语法	speed = i2c.setup(id, speed [,slaveaddr])	
参数	id	i2c 接口 id，例如 id=0 对应 I2C1，id=1 对应 I2C2 以此类推
	speed	i2c.FAST (400KHz)，i2c.SLOW (100KHz)
	slaveaddr	可选，i2c 外设地址 0x00-0x7f
返回值	可以根据返回的频率值判断是否成功打开 i2c	

表格 18：lua I2C 写接口

i2c.wirte()	往指定的寄存器地址 reg 传输数据	
语法	wrote = i2c.write(id, reg, data)	
参数	id	i2c 接口 id，例如 id=0 对应 I2C1，id=1 对应 I2C2 以此类推
	reg	写入 i2c 从设备的寄存器起始地址

	dat	number / string / table，自动根据参数类型写数据，num 只写 1 个字节，string/table
返回值	传输成功的字节数	

表格 19：lua I2C 初始化接口

i2c.read()	读取指定寄存器地址 reg 的数据内容	
语法	data = i2c.read(id, reg, num)	
参数	id	i2c 接口 id，例如 id=0 对应 I2C1，id=1 对应 I2C2 以此类推
	speed	读取 i2c 从设备的寄存器起始地址
	num	读取数据字节数
返回值	返回读取的数据，二进制数据会包含非可见字符，请使用 string.byte 打印数据流	

3.8. SPI 接口

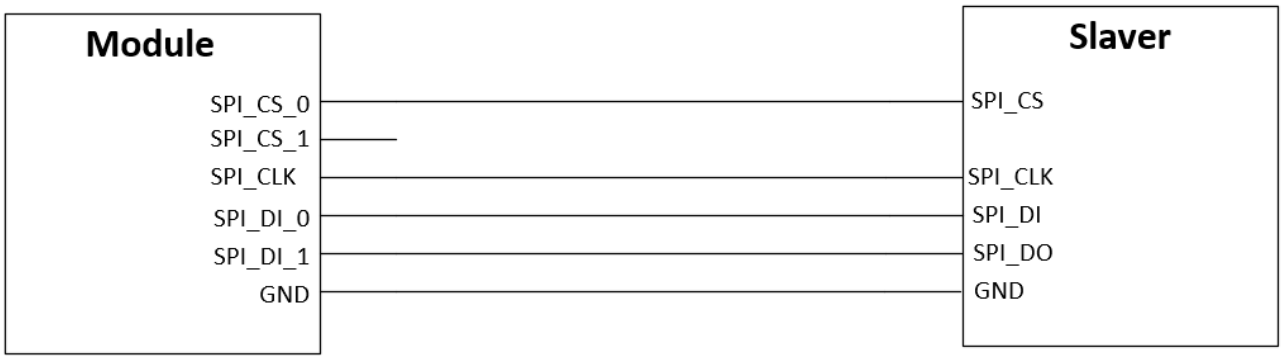
模块提供了 2 路标准 SPI 接口，1 路 LCD 专用 SPI 接口，一路 CAMERA 专用 SPI 接口,便于连接 SPI 接口的外设或者外置 SPI 存储器。本章节只讨论标准 SPI 接口，LCD 和 CAMERA 接口将会在其他章节详细说明。其中标准 SPI 接口支持 110K 到 13M 的时钟频率，支持半双工和全工双模式。SPI 接口有如下管脚复用：

表格 20：SPI 接口管脚定义

接口	名称	管脚	作用	电压域
SPI1	SPI1_CLK	4	SPI 时钟信号	V_MMC
	SPI1_CS_0	5	SPI 片选信号 0	
	SPI1_CS_1	3	SPI 片选信号 1	
	SPI1_DI_0	1	SPI 数据输入/输出	
	SPI1_DI_1	6	SPI 数据输入	
SPI2	UART1_RXD(SPI2_CLK)	47	SPI 时钟信号	VDDIO
	UART1_TXD(SPI2_CS_0	48	SPI 片选信号 0	
	UART1_RTS(SPI2_CS_1	49	SPI 片选信号 1	
	UART1_CTS(SPI2_DI_0)	50	SPI 数据输入/输出	
	UART2_RXD(SPI2_DI_1	45	SPI 数据输入	

3.8.1. SPI 接口连接方式

SPI 连接方式如下：



图表 13：I2C 5V 外设连接参考线路

3.8.2. SPI 相关 API 接口

在使用 SPI 接口时需要先调用 spi.setup()初始化接口，然后通过 spi.send()和 spi.recv()接口发送和接收数据，注意在发送和接收前需要用 GPIO 的操作方式去拉低 spi_cs 管脚，发送或接收完成后拉高 spi_cs 管脚。注意在使用 SPI 前要注意对应的电压域的 LDO 是否有打开。具体使用方法请参考 script_LuaTask/demo/spi

表格 21：lua SPI 初始化接口

spi. setup()	打开以及配置 spi 接口	
语法	spi.setup(id,chpa,cpol,dataBits,clock,duplexMode)	
参数	id	SPI 的 ID，spi.SPI_1 表示 SPI1，spi.SPI_2 表示 SPI2
	chpa	第几个 clk 的跳变沿传输数据，仅支持 0 和 1，0 表示第 1 个，1 表示第 2 个
	cpol	spi_clk idle 的状态，仅支持 0 和 1，0 表示低电平，1 表示高电平
	dataBits	数据位，仅支持 8
	clock	spi 时钟频率，支持 110K 到 13M（即 110000 到 13000000）之间的整数（包含 110000 和 13000000）
	duplex	是否全双工，仅支持 0 和 1，0 表示半双工（仅支持输出），1 表示全双工。此参数可选，默认半双工
返回值	number 类型，1 表示成功，0 表示失败	

3.9. 屏幕接口

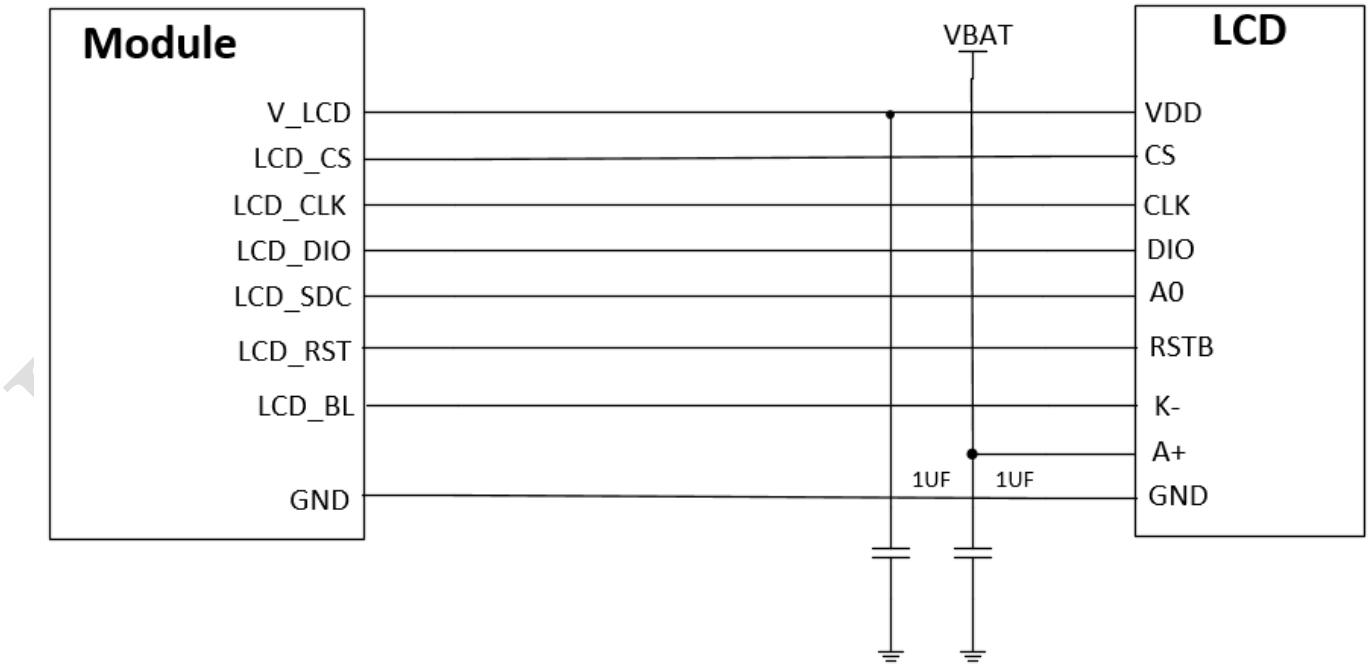
模块提供了一组 SPI 屏专用 SPI 接口，用于驱动 SPI 接口 LCD 屏幕。通过调节 VLCD 电压可以适配 1.8V/2.8V IO 电压的屏幕。注意，模块仅支持 4 线 1 通道配置的 SPI 屏幕。同时，配合背光 LED 驱动管脚可以调节屏背光的亮灭以及亮度。同时也可以使用标准 SPI 相关管脚驱动屏。屏幕专用 SPI 管脚如下：

表格 22：SPI 接口管脚定义

接口	名称	管脚	作用	电压域
LCD_SPI	V_LCD	38	LCD 供电输出	VLCD
	LCD_RST	15	LCD 复位信号	
	LCD_CS	13	LCD 片选信号	
	LCD_CLK	16	LCD 时钟信号	
	LCD_DIO	14	LCD 数据信号	
	LCD_SDC	17	LCD 数据命令选通信	
	LCD_BL	18	LCD 背光驱动信号	

3.9.1. LCD 接口连接方式

LCD 接口的连接方式如下：

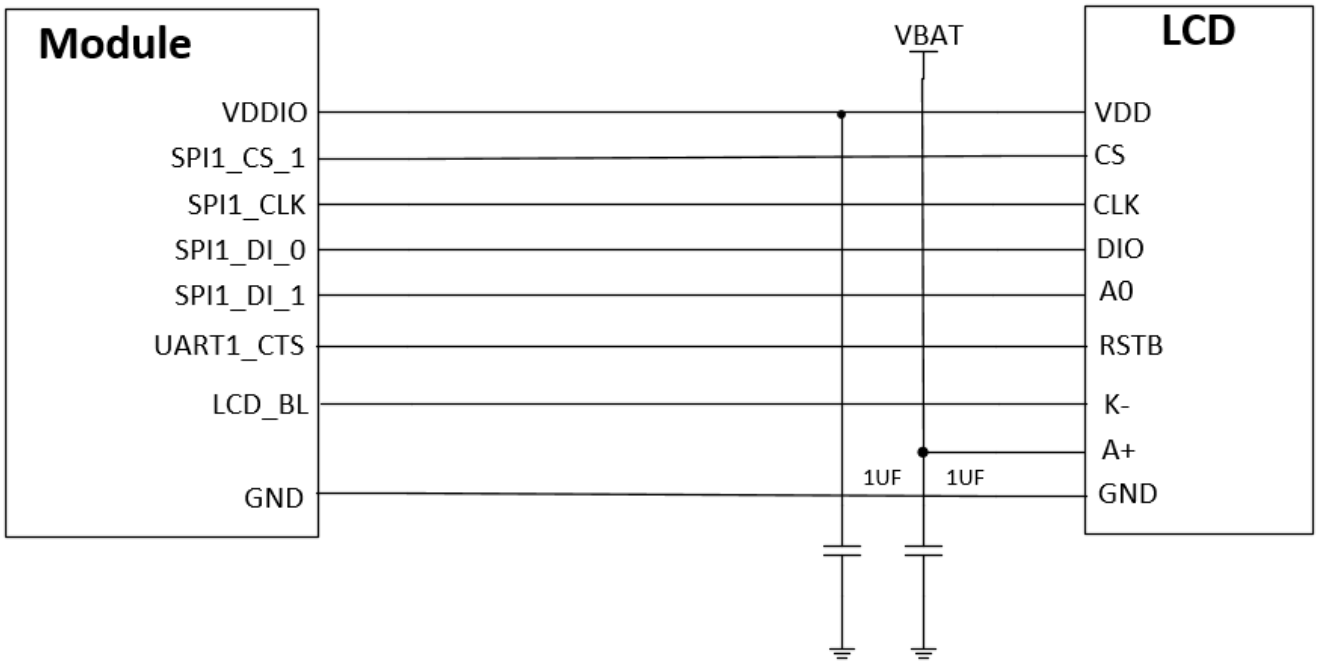


图表 14：LCD 接口参考线路

注意：模块最大只能支持 240*320 分辨率的 LCD，建议选择选择在 240*240 或以下分辨率的 LCD。

3.9.2. 用标准 SPI 接口连接屏幕的连接方式

模块也提供了用标准 SPI 接口连接屏幕的方案以供选择。



图表 15：标准 SPI 接 LCD 参考线路

注意：如果要用 SPI 接屏建议使用 SPI1 通道，在脚本库中已经集成了相关的驱动直接调用就可以。另外 SPI 接口只支持 2.8V LCD，不支持 1.8V。

3.9.3. 屏幕相关 API 接口

在 script_LuaTask/demo/ui 的 DEMO 中已经集成了常用 LCD 型号的驱动，在器件选型时注意选择列表中的 LCD 驱动芯片方案，这样可以节省调试时间。已经集成的 LCD 驱动型号如下：

```
-- require "mono_std_spi_sh1106"
-- require "mono_std_spi_ssd1306"
-- require "mono_std_spi_st7567"
require "color_std_spi_st7735"
-- require "color_std_spi_st77351"
-- require "color_std_spi_ILI9341"
-- require "color_lcd_spi_ILI9341"
-- require "mono_lcd_spi_sh1106"
-- require "mono_lcd_spi_ssd1306"
-- require "mono_lcd_spi_st7567"
-- require "color_lcd_spi_st7735"
-- require "color_lcd_spi_gc9106"
-- require "mono_i2c_ssd1306"
```

其中背光的引脚需要调用 pmd.ldoset(1,pmd.LDO_LCD)来打开背光。

3.10. 摄像头接口

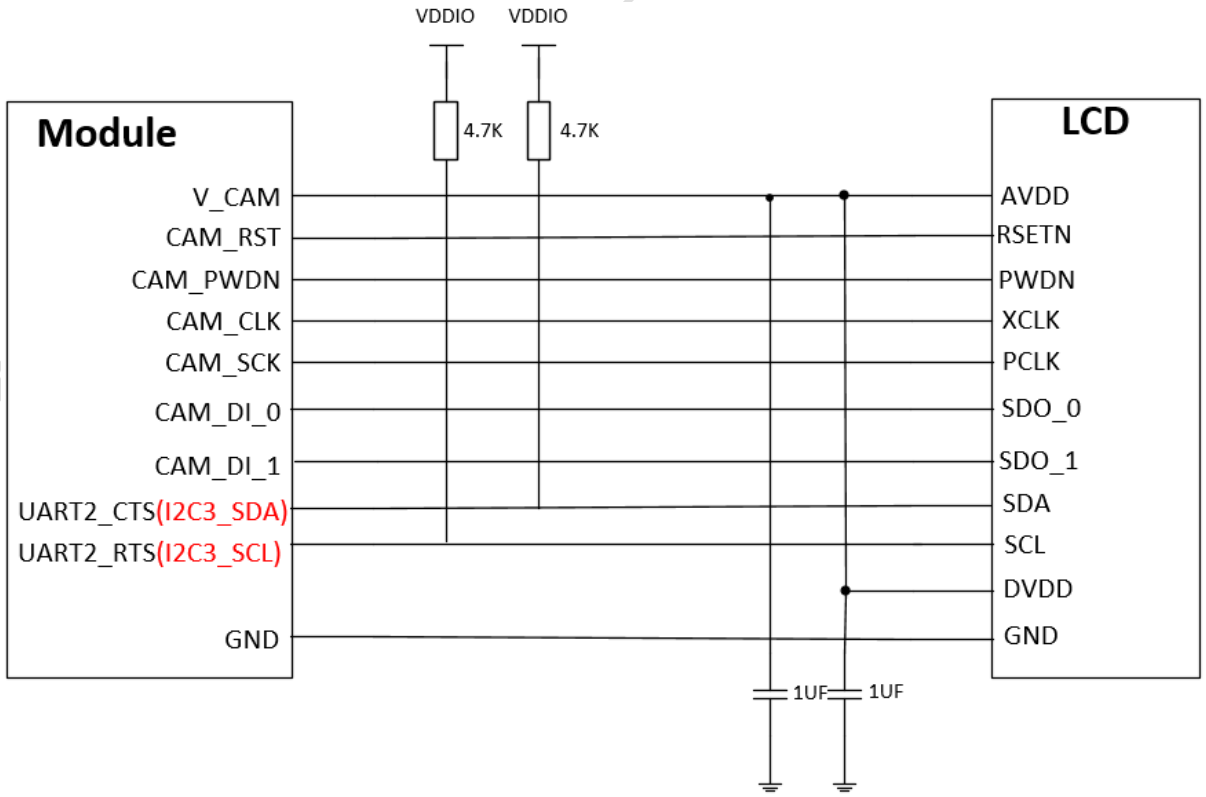
模块内置一路摄像头接口用于驱动 SPI 串行摄像头。最大支持 320*240 像素，支持 1.8V，3.3V 摄像头。摄像头接口管脚定义如下：

表格 23：SPI 接口管脚定义

接口	名称	管脚	作用	电压域
CAM	V_CAM	37	camera 供电输出	V_CAM
	CAM_PWDN	66	camera 复位信号	
	CAM_RST	64	camera 片选信号	
	CAM_CLK	68	camera 时钟信号	
	CAM_SCK	63	SPI 时钟信号	
	CAM_DI_0	67	SPI 数据输入 0	
	CAM_DI_1	65	SPI 数据输入 0	

3.10.1. 摄像头接口的连接方式

如下是摄像头接口的参考设计，其中要连接一路 I2C 接口，建议优先用 I2C3。



图表 16：摄像头接口参考线路

3.10.2. 摄像头相关 API 接口说明

摄像头使用的一般步骤为：先用 disp.cameraopen(1)打开摄像头，然后打开预览可以调用 disp.camerapreview()，可以调用 disp.cameracapture()进行拍照，同时调用 disp.camerasavephoto()保存图片，但是要注意模块存储空间是否足够。注意：摄像头接口需 V0028 以及以上 core 版本才能支持

表格 24：lua 打开摄像头接口

disp.cameraopen()	打开摄像头	
语法	disp.cameraopen()	
参数	无	
返回值	result	ture-成功，false-失败

表格 25：lua 打开预览接口

disp.camerapreview()	打开预览	
语法	disp.camerapreview(layerx,layery,startx,starty,endy)	
参数	layerx	视频图层在虚拟屏幕上的起始坐标 x
	layery	视频图层在虚拟屏幕上的起始坐标 y
	startx	视频图层在视频图层的起始坐标 x
	starty	视频图层在视频图层的起始坐标 y
	endx	视频图层在视频图层的结束坐标 x
	endy	视频图层在视频图层的结束坐标 y
返回值	无	

表格 26：lua 拍照接口

disp.cameracapture()	拍照	
语法	disp.cameracapture(width, height)	
参数	width	宽
	height	高
返回值	无	

表格 27：lua 保存照片接口

disp.camerasavephoto()	保存照片	
语法	disp.camerasavephoto(filename)	
参数	filename	全路文件名
返回值	结果码：0 成功；-1 文件路径错误或者其他错误	

3.11. PWM 接口

Air168 模块支持 2 路 PWM 输出：PWT,LPG，分别由 UART2_RX 和 UART2_TXD 复用。其中 LPG 为低速 PWM，用于低频率的应用如驱动 LED 闪烁，只能设定固定的 7 种周期（单位 ms）：125,250,500,1000,1500,2000,2500,3000。以及 15 种高电平时间。PWT,频率范围（80-65535HZ），以及可以设置各种占空比。

表格 28：串口管脚定义

接口	名称	管脚	作用
PWM	UART2_TXD(PWT)	46	高速 PWM
	UART2_RXD(LPG)	45	低速 PWM

PWM 相关的 API 接口：详细请参考 script_LuaTask/demo/PWM

表格 29：lua PWM 接口

misc.openPwm()	打开并配置 PWM	
语法	misc.openPwm(id, period, level)	
参数	id	number， PWM 输出通道，仅支持 0 和 1，0 对应 LPG，1 对应 PWT
	period	number， 当 id 为 0 时，period 表示频率，单位为 Hz，取值范围为 80-1625，仅支持整数。 当 id 为 1 时，取值范围为 0-7，仅支持整数，表示时钟周期，单位为毫秒，0-7 分别对应 125、250、500、1000、1500、2000、2500、3000 毫
	height	number， 当 id 为 0 时，level 表示占空比，单位为 level%，取值范围为 1-100，仅支持整数。 当 id 为 1 时，取值范围为 1-15，仅支持整数，表示一个时钟周期内的高电平时间，单位为毫秒，1-15 分别对应 15.6、31.2、46.9、62.5、78.1、93.7、110、125、141、156、172、187、203、219、234 毫秒

返回值	无
说明	当 id 为 0 时：period 取值在 80-1625 Hz 范围内时，level 占空比取值范围为：1-100； period 取值在 1626-65535 Hz 范围时，设 x=162500/period，y=x * level / 100，x 和 y 越是接近正的整数，则输出波形越准确

表格 30：lua PWM 关闭接口

misc.closePwm()	关闭 PWM	
语法	misc.closePwm(id)	
参数	id	number， PWM 输出通道，仅支持 0 和 1，0 对应 LPG，1 对应 PWT
返回值	无	

3.12. ADC 接口

模块内置两路 ADC，可以用来做电池电压检测，温湿度检测，TDS 检测等应用。ADC 精度为 10bit，测量输入范围为 0 到 1.85V,测量误差范围为+-20mV。若测量的电压输入范围大于 1.85V 则需要进行分压后再输入 ADC。

表格 31：串口管脚定义

接口	名称	管脚	作用
ADC	ADC0	40	数模转换输入
	ADC1	41	数模转换输入

ADC 相关接口：详细请参考 script_LuaTask/demo/ADC

表格 32：lua 打开 ADC 接口

adc.open()	打开对应 ID 的通道	
语法	result = adc.open(id)	
参数	id	number， adc 通道，仅支持 0 和 1，0 对应 adc0，1 对应 adc1
返回值	1:成功 ； 0：其他	

表格 33：lua ADC 读接口

adc.read()	读取原始测量数据和电压值，电压值单位为 mv
语法	adcValue,voltValue = adc.read(id)

参数	id	number， adc 通道，仅支持 0 和 1，0 对应 adc0，1 对应 adc1
返回值	adcValue:原始数据 ad 值，无效值为 0xFFFF	
	voltvalue:电压值，单位为 mv，无效值为 0xFFFF	

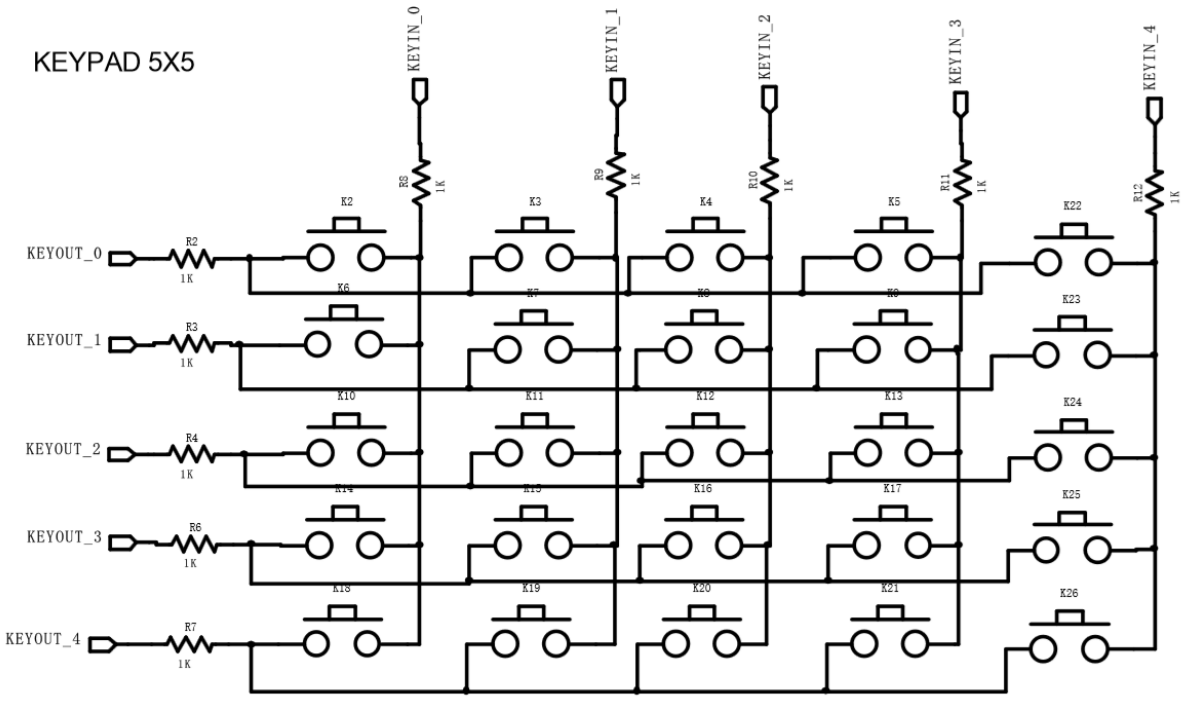
3.13. 键盘接口

模块支持 5*5 阵列键盘，支持各种键盘的应用。

表格 34：键盘管脚定义

接口	名称	管脚	作用
KEYPAD	KEYIN_0	21	键盘扫描阵列输入 0
	KEYIN_1	22	键盘扫描阵列输入 1
	KEYIN_2	23	键盘扫描阵列输入 2
	KEYIN_3	24	键盘扫描阵列输入 3
	KEYIN_4	25	键盘扫描阵列输入 4
	KEYOUT_0	26	键盘扫描阵列输出 0
	KEYOUT_1	27	键盘扫描阵列输出 1
	KEYOUT_2	28	键盘扫描阵列输出 2
	KEYOUT_3	29	键盘扫描阵列输出 3
	KEYOUT_4	30	键盘扫描阵列输出 4

键盘参考设计：



图表 17：键盘接口参考线路

注意：1.键盘走线请尽量远离天线，以免对天线造成干扰。

2. 键盘走线串联 1K 电阻以做 ESD 防护。

3.14. 通用 GPIO

模块多数管脚都可以复用为通用 GPIO 管脚，多达 35 个 GPIO 口，所以 GPIO 口均可配置为输入和输出，且都可以响应中断，作为输入时都可以配置为上拉或下拉。使用 GPIO 口时一定要打开 GPIO 对应电压域的 LDO 才能正常工作，具体复用的 GPIO 请参考表格 3。

GPIO 电气特性：

表格 35：GPIO 电气特性

Symbol	*Description*	*Min.(V)*	*Typical(V)*	*Max.(V)*
VDD	All of power for digital usage	VDD-0.2	1.8/2.8	VDD+0.2
VIL	CMOS Low Level Input Voltage	0	-	0.3*VDD
VIH	CMOS High Level Input Voltage	0.7*VDD	-	VDD
VTH	CMOS Threshold Voltage	-	0.5*VDD	-

GPIO 相关 API 接口：详细请参考 script_LuaTask/demo/gpio

表格 36：lua GPIO 输入输出设置接口

pio.pin.setdir()	GPIO 输入输出设置	
语法	pio.pin.setdir(direction,pin1,pin2,...,pinn)	
参数	direction	管脚描述，可选 pio.INPUT ， pio.OUTPUT， pio.INT
	pin1	第一个管脚
	pin2(可选)	第二个管脚
	pinn(可选)	第 n 个管脚
返回值	无	

表格 37：lua GPIO 读取接口

pio.pin.getval()	读取 GPIO 管脚的值	
语法	val1, val2, ..., valn = pio.pin.getval(pin1, pin2, ..., pinn)	
参数	pin1	第一个管脚
	pin2(可选)	第二个管脚
	pinn(可选)	第 n 个管脚
返回值	number 类型，0 表示低电平，1 表示高电平	

表格 38：lua GPIO 输出低接口

pio.pin.setlow()	设置 GPIO 管脚为低电平	
语法	pio.pin.setlow(pin1,pin2,...,pinn)	
参数	pin1	第一个管脚
	pin2(可选)	第二个管脚
	pinn(可选)	第 n 个管脚
返回值	无	

表格 39：lua GPIO 输出高接口

pio.pin.sethigh()	设置 GPIO 管脚为高电平	
语法	pio.pin.sethigh(pin1,pin2,...,pinn)	
参数	pin1	第一个管脚

	pin2(可选)	第二个管脚
	pinn(可选)	第 n 个管脚
返回值	无	

表格 40：lua GPIO 输出高接口

pio.pin.sethigh()	设置 GPIO 管脚为高电平	
语法	pio.pin.sethigh(pin1,pin2,...,pinn)	
参数	pin1	第一个管脚
	pin2(可选)	第二个管脚
	pinn(可选)	第 n 个管脚
返回值	无	

表格 41：lua GPIO 上下拉接口

pio.pin.setpull()	设置 GPIO 的上下拉状态	
语法	pio.pin.setpull(method,pin)	
参数	method	pio.PULLUP：上拉模式 。pio.PULLDOWN：下拉模式。pio.NOPULL：高阻态
	pin	配置管脚
返回值	无	

3.15. LDO 输出

模块内置 3 路 LDO 输入可以对外供电，分别是 VDDIO，V_LCD，V_CAM，其中 VDDIO 为固定输出不可调节，其余两个 LDO 均可以软件控制打开与关闭，以及电压值。

表格 42：LDO 管脚定义

接	名称	管脚	电压	电流
ADC	VDDIO	39	2.8+-0.1V	100mA
	V_CAM	37	2.8+-0.1V/1.8+-0.1V	200mA
	V_LCD	38	2.8+-0.1V/1.8+-0.1V	200mA

LDO 相关 API 接口：

pmd.ldoset()	LDO 控制	
语法	pmd.ldoset(level,id1,[id2],...,[idn])	
参数	level	do 亮度 0 - 7 级 0 级关闭 （注意目前仅对 LDO_VMMC 有效） 0--关闭 1--1.8V 2--1.9V 3--2.0V 4--2.6V 5--2.8V 6--3.0V 7--3.3V
	di	要设置的第一个 ldo
	di2	要设置的第二个 ldo
	din(可选)	要设置的第 n 个 ldo
返回值	无	
备注	一旦设置了某一个电压域的电压等级，受该电压域控制的所有 GPIO 的高电平都与设置的电压等级一致，相关 GPIO 对应的电压域请参考表格 3	

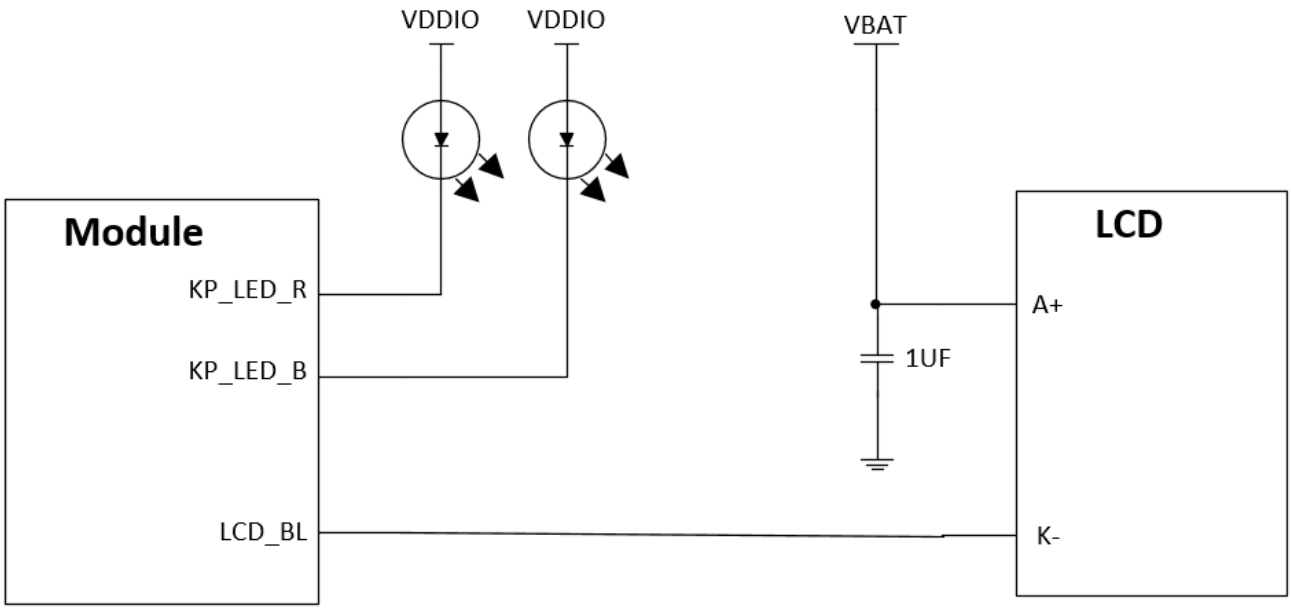
3.16. 背光输出控制管脚

模块内置 3 路背光驱动控制管脚，分别为 LCD_BL，KP_LED_B，KP_LED_R。这三路管脚与其他 GPIO 不同，为电流源，只能向内部输入电流，无法输出。通常用来接 LED 的阴极。其中 LCD_BL 最大驱动电流为 60mA，通常用于驱动屏幕背光。KP_LED_B，KP_LED_R 最大驱动电流为 30mA，通常用来驱动发光二极管。

表格 43：LDO 管脚定义

接口	名称	管脚	最大驱动电流
LED	LCD_BL	18	60mA
	KP_LED_B	19	30mA
	KP_LED_R	20	30mA

相关参考设计如下：



图表 18：LED 驱动电路参考线路

3.17. 音频接口

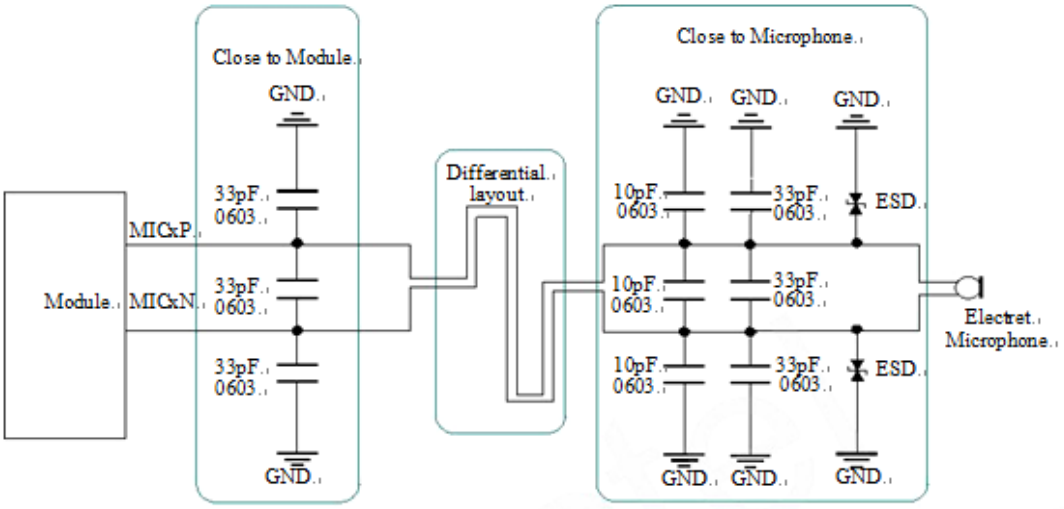
模块一共有三路模拟音频输入输出通道，支持通话、录音和播放等功能。分别是差分麦克风输入接口，差分喇叭输出接口和立体声耳机接口。

表格 44：模拟音频管脚定义

接口	名称	管脚	作用	说明
SPEAKER	SPK+	10	音频输出+	默认 AB 类输出，内部集成音频输出功放，最大输出功率 1W
	SPK-	11	音频输出-	
MIC	MIC+	32	音频输入+	内置 mic 偏置电路，内置隔直电容。
	MIC-	33	音频输入-	
HEADPHONE	HP_L	34	立体声耳机左声道	立体声耳机输出
	HP_R	35	立体声耳机右声道	

3.17.1. 麦克风接口参考电路

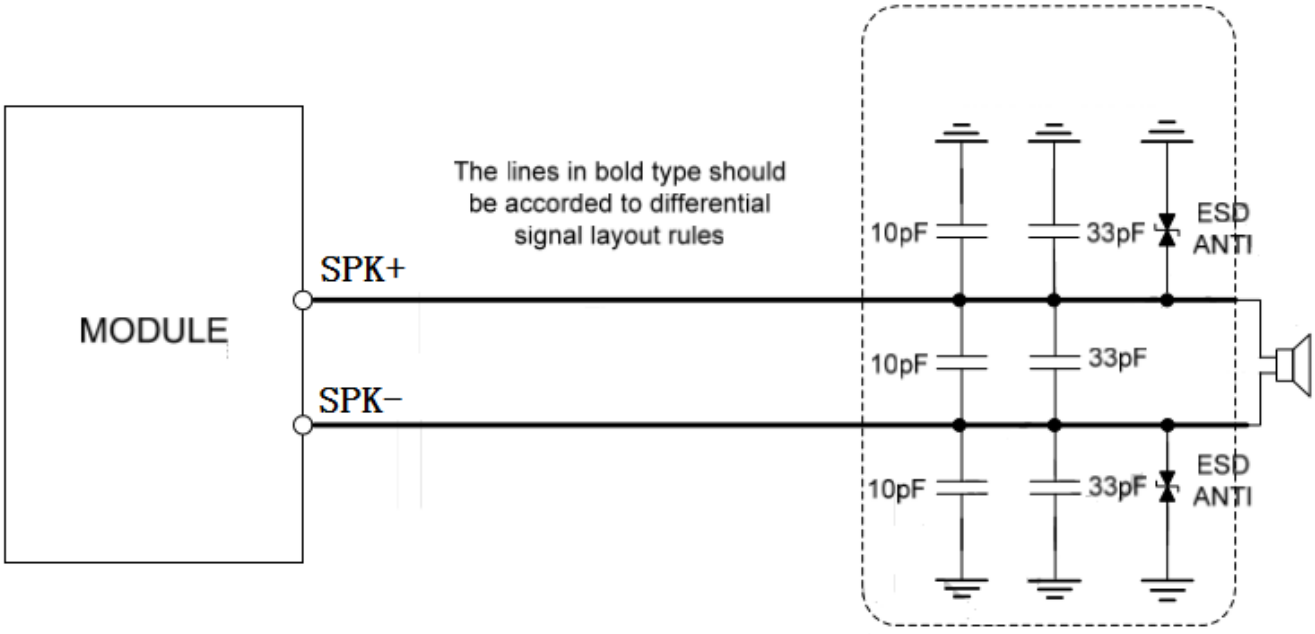
麦克风通道参考电路下图所示：



图表 19：麦克风通道接口电路

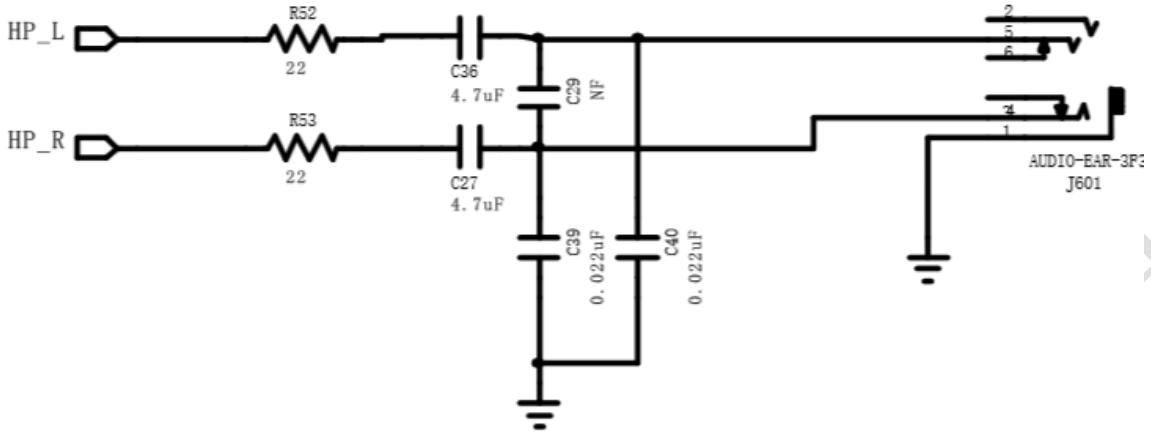
3.17.2. 音频输出接口参考电路

SPK音频输出接口可以直接与驱动8欧姆喇叭。



图表 20：参考线路

3.17.3. 耳机接口参考电路



图表 21：耳机参考线路

3.17.4. 音频接口相关 API 接口说明

使用音频接口时注意设置对应的音频通道。详细请参考 script_LuaTask/demo/audio

表格 45：lua GPIO 读取接口

audio.play()	播放音频文件	
语法	audio.play(priority, type, path[, vol=4][, cbFnc=nil][, dup=nil][, dupInterval=0])	
参数	priority	number， 音频优先级，数值越大，优先级越高
	type	string，音频类型，目前仅支持"FILE"、"TTS"、"TTSCC","RECORD"
	path	string，音频文件路径，跟 typ 有关 typ 为"FILE"时：表示音频文件路径 typ 为"TTS"时：表示要播放的 UTF8 编码格式的数据 typ 为"TTSCC"时：表示要播放给通话对端的 UTF8 编码格式的数据
	vol	number， 此参数可选，默认值为： 4 ，播放音量，取值范围 0 到 7，0 为静音
	cbFnc	function， 此参数可选，默认值为： nil，音频播放结束时的回调函数，回调函数的调用形式如下： cbFnc(result) result 表示播放结果： 0-播放成功结束； 1-播放出错 2-播放优先级不够，没有播放

		3-传入的参数出错，没有播放 4-被新的播放请求中止
	dup	bool ， 此参数可选，默认值为： nil，是否循环播放，true 循环，false 或者 nil 不循环
	dupInterval	number，此参数可选，默认值为： 0，循环播放间隔(单位毫秒)，dup 为 true 时，此值才有意义
返回值	result，bool 或者 nil 类型，同步调用成功返回 true，否则返回 nil	
备注	如要使用 TTS 功能，需要使用支持 TTS 功能的 core 底层软件	

表格 46：lua 设置喇叭音量接口

audio.setVolume()	设置喇叭音量等级	
语法	audio.setVolume(vol)	
参数	vol	number，音量值为 0-7，0 为静音
返回值	bool result，设置成功返回 true，失败返回 false	

表格 47：lua 设置麦克风音量接口

audio.setMicVolume()	设置麦克风音量等级	
语法	audio.setMicVolume(vol)	
参数	vol	number，音量值为 0-7，0 为静音
返回值	bool result，设置成功返回 true，失败返回 false	

表格 48：lua 设置音频通道接口

audiocore.setchannel()	设置音频通道	
语法	result = audiocore.setchannel(channel)	
参数	channel	固定值，audiocore.EARPIECE 对应耳机通道 audiocore.LOUDSPEAKER 对应 speaker 通道
返回值	result，设置成功返回 1，失败返回 0	

图表 22：射频焊接方式建议

4. 电器特性，可靠性，射频特性

4.1. 绝对最大值

下表所示是模块数字、模拟管脚的电源供电电压电流最大耐受值。

表格 49：绝对最大值

参数	最小	最大	单位
V_{BAT}	-0.3	4.5	V
数字管脚处电压	-0.3	3.3	V
模拟管脚处电压	-0.3	3.0	V
关机模式下数字/模拟管脚处电压	-0.25	0.25	V

4.2. 工作温度

表格 50：工作温度

温度	最低	典型	最高	单位
正常工作温度	-40	25	85	℃
存储温度	-45		90	℃

4.3. 电压额度值

表格 51：模块电源额度值

参数	描述	条件		最小	典型	最大	单位
V_{BAT}	供电电压	电压必须在该范围之内，包括电压跌落，纹波和尖峰时		3.4	4.0	4.2	V
I_{VBAT}	平均供电电流	关机模式	第一次上电		34		uA
			开机后关机（RTC 正常工作）		195		uA
		底电流			0.8		mA

		飞行模式 AT+CFUN=4		0.896		mA

4.4. 静电防护

在模块应用中，由于人体静电，微电子间带电摩擦等产生的静电，通过各种途径放电给模块，可能会对模块造成一定的损坏，所以 ESD保护必须要重视，不管是在生产组装、测试，研发等过程，尤其在产品设计中，都应采取防 ESD保护措施。如电路设计在接口处或易受 ESD点增加 ESD保护，生产中带防ESD手套等。

下表为模块重点PIN脚的ESD耐受电压情况。

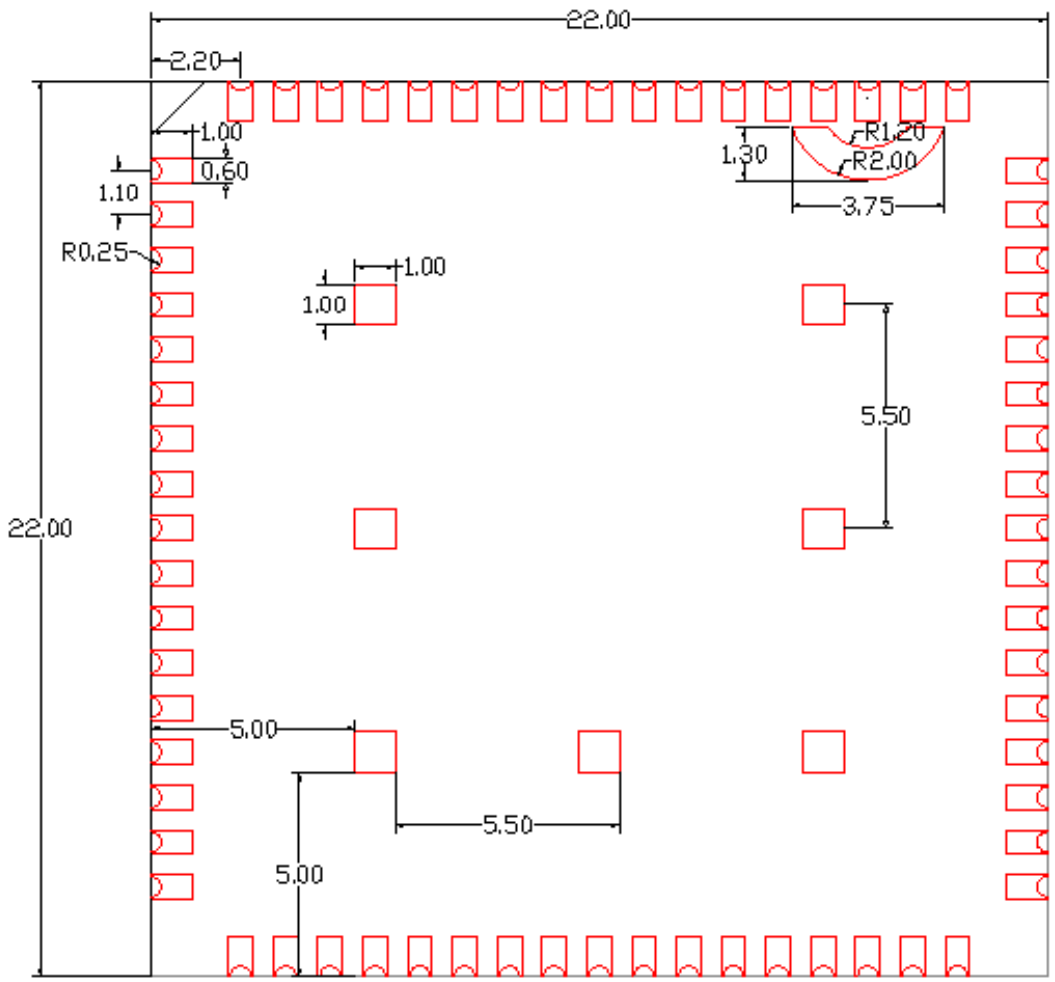
表格 52：ESD 性能参数（温度：25℃，湿度：45%）

管脚名	接触放电	空气放电
VBAT,GND	±5KV	±10KV
RF_ANT	±5KV	±10KV
TXD, RXD	±2KV	±4KV
Others	±0.5KV	±1KV

5. 机械尺寸

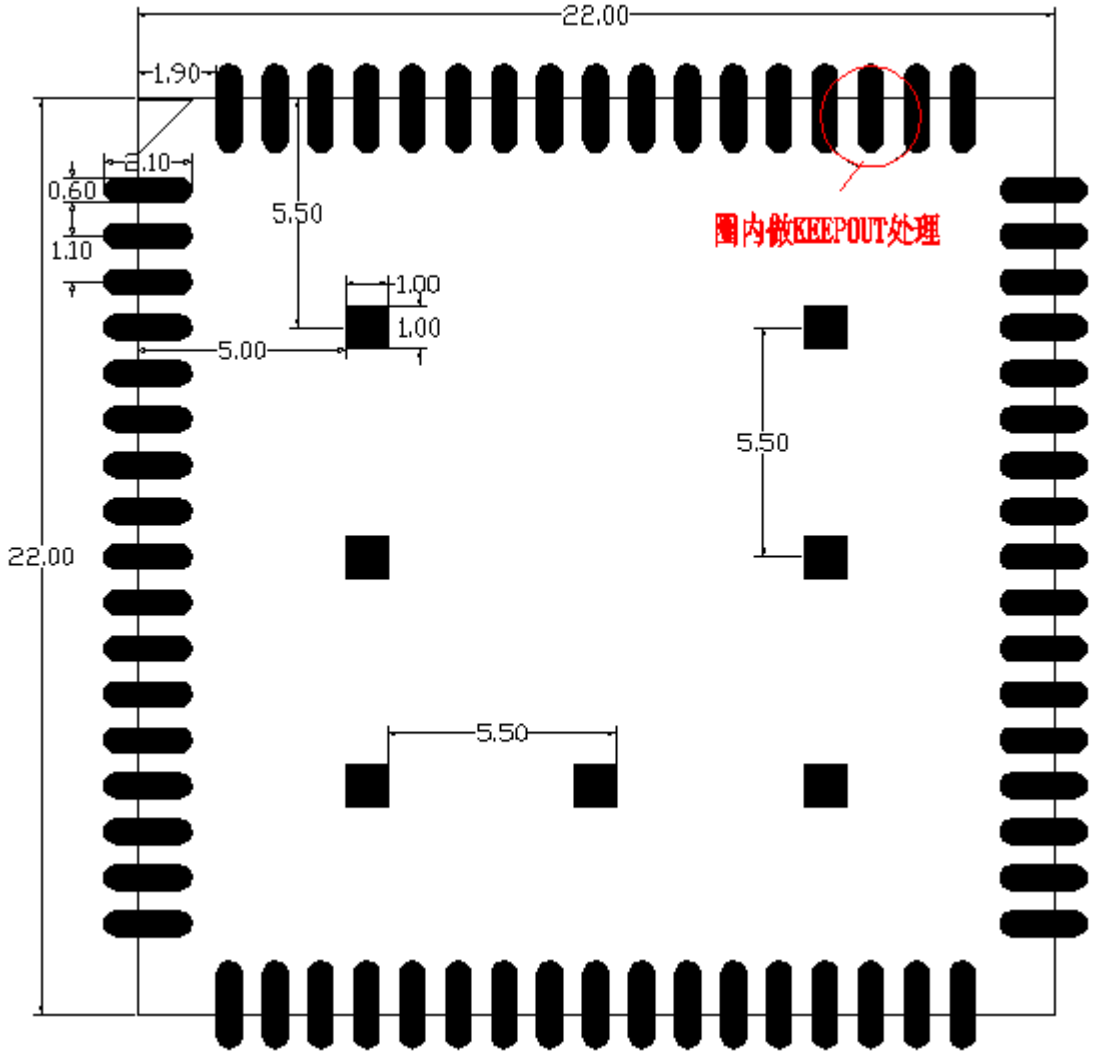
该章节描述模块的机械尺寸以及客户使用该模块设计的推荐封装尺寸。

5.1. 模块机械尺寸



图表 23：Air168 正视图（单位：毫米）

5.2. 推荐 PCB 封装



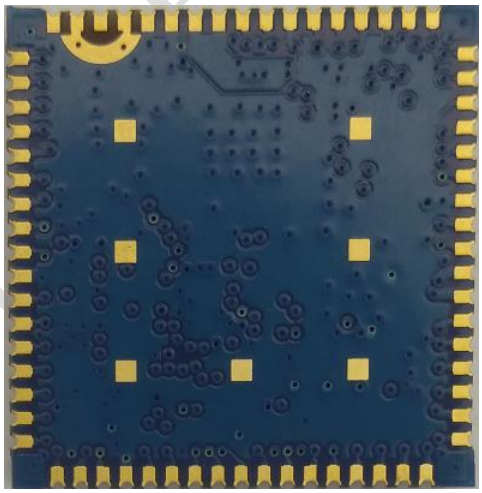
图表 24：推荐封装（单位：毫米）

注意：保证 **PCB** 板上模块和其他元器件之间间距至少 **3mm**。

5.3. 模块正视图

图表 25：模块正视图

5.4. 模块底视图



图表 26：模块底视图

6. 存储和生产

6.1. 存储

Air168以真空密封袋的形式出货。模块的存储需遵循如下条件：

环境温度低于40摄氏度，空气湿度小于90%情况下，模块可在真空密封袋中存放12个月。

当真空密封袋打开后，若满足以下条件，模块可直接进行回流焊或其它高温流程：

- 模块环境温度低于30摄氏度，空气湿度小于60%，工厂在72小时以内完成贴片。
- 空气湿度小于10%

若模块处于如下条件，需要在贴片前进行烘烤：

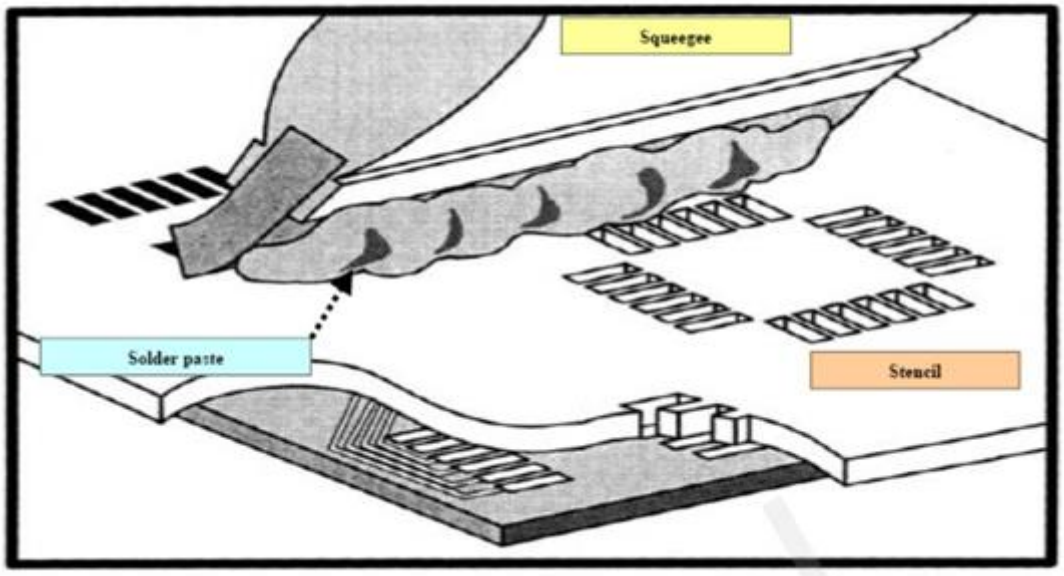
- 当环境温度为23摄氏度（允许上下5摄氏度的波动）时，湿度指示卡显示湿度大于10%
- 当真空密封袋打开后，模块环境温度低于30摄氏度，空气湿度小于60%，但工厂未能在72小时以内完成贴片
- 当真空密封袋打开后，模块存储空气湿度大于10%

如果模块需要烘烤，请在 125 摄氏度下（允许上下 5 摄氏度的波动）烘烤 48 小时。

注意：模块的包装无法承受如此高温，在模块烘烤之前，请移除模块包装。如果只需要短时间的烘烤，请参考 **IPC/JEDECJ-STD-033** 规范。

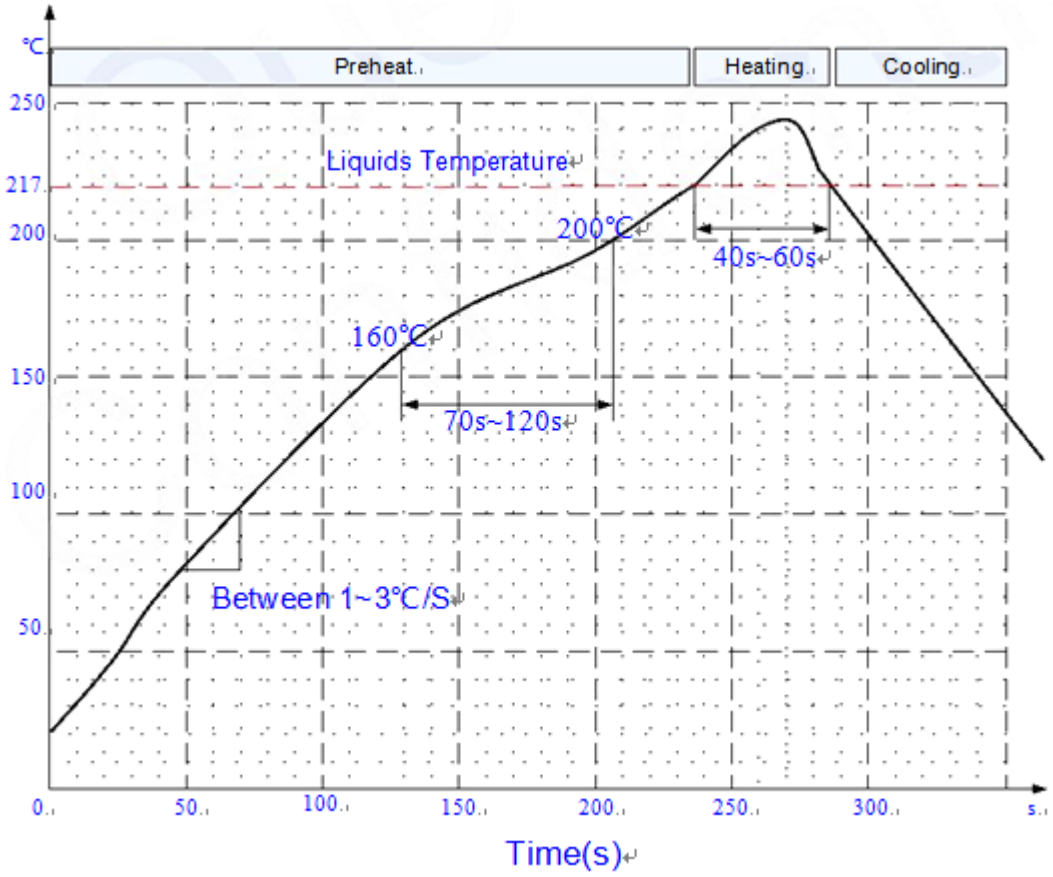
6.2. 生产焊接

用印刷刮板在网板上印刷锡膏，使锡膏通过网板开口漏印到 PCB上，印刷刮板力度需调整合适，为保证模块印膏质量，Air168模块焊盘部分对应的钢网厚度应为 0.2mm。



图表 27：印膏图

为避免模块反复受热损伤，建议客户 PCB板第一面完成回流焊后再贴模块。推荐的炉温曲线图如下图所示：



图表 28：炉温曲线

7. 联系我们

（1）淘宝店铺名称：合宙物联网

https://luat.taobao.com

https://openluat.taobao.com

（2）Luat 之家网站：

http://www.openluat.com/

（3）合宙 OpenLuat 开源模块技术支持

QQ 讨论群：201848376

（4）GitHub：

https://github.com/openLuat/Luat_Air168-Air800-Air201

（5）微信公众号: Luat

Lua+AT=Luat

Open+Luat=OpenLuat

www.OpenLuat.com

合宙--》Luat--》发烧友--》客户--》产品

共建开源好生态！