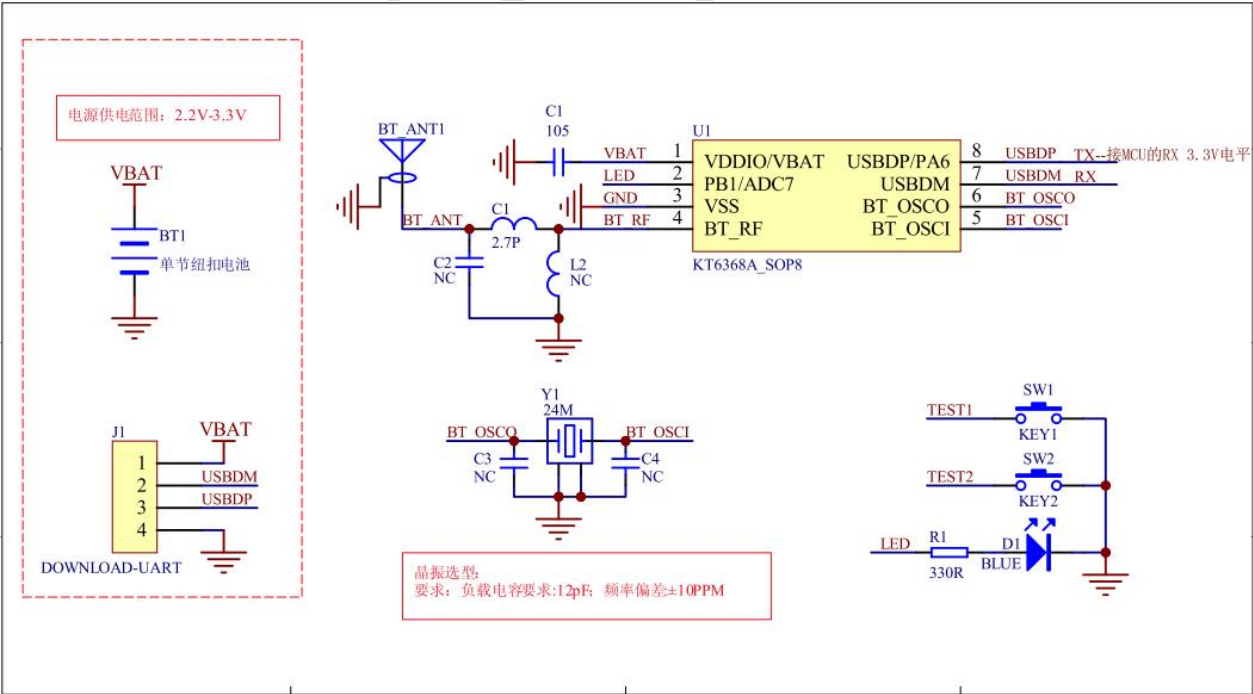


KT6368A 芯片使用手册

文件状态：	文件标识：	外部公开
[] 草稿	当前版本：	V2.1
[√] 正式发布	作 者：	清月电子
[] 正在修改	完成日期：	2022-11-03



版本历史

版本	日期	原因
V2.1	2022/11/03	1、新增 spp 的密码的设定，详见5.4章节 2、新增芯片2脚的功能设定，可以设置为未连接闪烁，连接常亮，详见3.2章节 ==》【AT+CF01 代表未连接闪烁，连接常亮】【AT+CF00未连接低，连接高】 3、新增 ble 的 uuid 的 AT 指令设定，详见4.7章节 4、完善文档的细节，同时此版本向下兼容以前的老版本 5、新增章节8的特别说明、版本说明 6、新增蓝牙发射功率的调整接口，详见3.12章节 7、新增批量生产测试的相关说明，详见章节9 8、 新增章节10，硬件注意事项，新增软件说明--设置蓝牙名重要说明

目 录

1. 概述	5
1.1 简介	5
KT6368A 芯片是一款支持蓝牙双模的纯数据芯片，蓝牙 5.1 版本。芯片的亮点在超小尺寸，超级价格。以及简单明了的透传和串口 AT 控制功能。大大降低了嵌入蓝牙在其它产品的开发难度和成本	
1.2 硬件说明	5
1.3 芯片功耗初步说明	6
1.4 芯片的简单测试说明	6
1.5 硬件设计--脚位说明	6
1.6 硬件设计--蓝牙天线的说明	7
1.7 硬件设计--蓝牙搭配的晶振说明	7
1.8 硬件设计--统计到目前为止客户的错误案例-很重要	8
3. 串口通讯协议	8
3.1 通讯格式	8
3.2 通讯指令举例	9
3.3 指定芯片的波特率和复位和恢复出厂设置【CT】[CZ][CW]	10
3.4 设置 BLE 蓝牙的名称和地址[BM][BN][BD]	10
3.5 查询 BLE 蓝牙的名称和地址[TM][TN][TD]	11
3.6 芯片上电信息和串口调试助手	12
3.7 芯片低功耗指令说明【CL】	13
3.8 芯片 BLE 使能和 SPP 使能[B4][B5][T4][T5]	15
3.9 芯片返回的错误信息说明【ER】	15
3.10 芯片上电回传信息关闭指令【CR】	15
3.11 指定芯片的 BLE 的广播间隔【UT】	16
3.12 芯片 RF 蓝牙发射功率[BR]	17
4. 蓝牙透传的详细说明--BLE	18
4.1 BLE 的透传说明	18
4.2 BLE 的 UUID 说明	18
4.3 BLE 的测试说明	18
4.4 BLE 的手机端 APP 测试说明--LIGHTBLUE 测试	19
4.5 BLE 的大数据量测试	20
4.6 BLE 的广播包数据说明--ADVERTISDATA	21
4.7 BLE 的 UUID 通过 AT 指令去修改的说明[U0][U1][U2][U3]	21
5. 蓝牙透传的详细说明--SPP	22
5.1 SPP 的透传说明	22
5.2 SPP 的透传效果演示说明	22

1、 SPP 透传效果演示： HTTPS://V.QQ.COM/X/PAGE/B0766JQW0P5.HTML	22
5.3 SPP 的透传测试说明	22
5.4 SPP 的连接密码说明[BE][B1]	22
6. 关于 AT 指令和透传数据的详细说明	23
7. 常见问题集锦	24
8. 新增特别的说明	28
9. 新增批量生产说明	29
10. 新增一些使用注意事项--很重要!!!	31
10.1 关于 KT6368A 的串口和 MCU 之间的电平转换以及串电阻	31
10.2 关于 KT6368A 的晶振重要说明	32
10.3 关于 KT6368A 芯片修改蓝牙名_波特率_等等需要记忆的参数重要说明	32
10.4 关于 KT6368A 芯片的供电问题	33

1. 概述

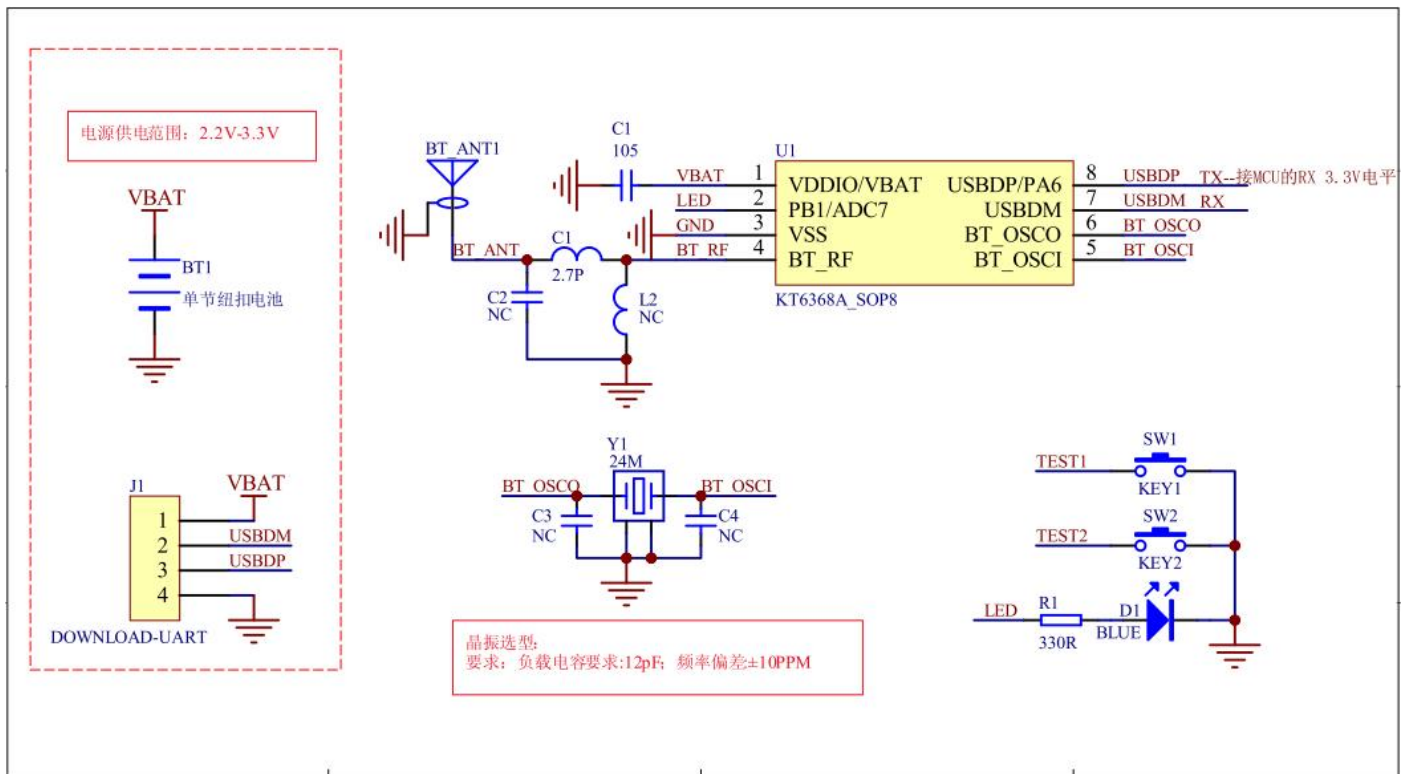
1.1 简介

KT6368A 芯片是一款支持蓝牙双模的纯数据芯片，蓝牙 5.1 版本。芯片的亮点在超小尺寸，超级价格。以及简单明了的**透传和串口 AT 控制功能**。大大降低了嵌入蓝牙在其它产品的开发难度和成本

同时支持 SPP 和 BLE 。但是只能任选其中一个协议使用。

备注：这款芯片最大的特点，就是成本低，使用简单，生产简单。无其他。同时支持低功耗详见 3.7 章节

1.2 硬件说明



细节	参数说明
UART 接口	标准串口，TTL 电平，波特率可设，连接 PC 需要电平转换[如：CH340G--USB 转 TTL]
输入电压	建议给 3.3V 的电压【2.2V--3.4V】
额定电流	芯片上电启动是20mA，马上进入低功耗广播20uA 和唤醒4mA 交替。 连接成功就一直都是4mA
低功耗电流	芯片算的是平均电流，因为内部是不断的低功耗、唤醒交替进行
工作温度	[-40度] -- [80度]
湿度	5% ~ 95%
主芯片型号	KT6368A[SOP8][管装出货]-----KT6328A[SOP8][管装出货]

1.3 芯片功耗初步说明

- 1、我们目前分 KT6368A 和 KT6328A 两个版本
- (1)、KT6368A 版本，是不带低功耗，双模的版本，开机 15mA ，后续一直稳定在 6mA 左右
- (2)、KT6328A 版本，低功耗版本，只有 BLE，详细参数，如：3.7 章节
- (3)、这两个芯片版本的硬件一模一样，KT6328A 存在的目的就是满足需要低功耗的客户
- 2、KT6368A 版本的特点如下：**
- (1)、双模 SPP + BLE，手册里面的全部功能具备 。就是不具备低功耗
- (2)、但是这个版本，成本更低一些。
- 3、KT6268A 版本的特点如下：**
- (1)、功耗更低，详见 3.7 章节描述
- (2)、成本略高一点点 。
- ==》不同的版本，通过蓝牙名是可以识别出来的

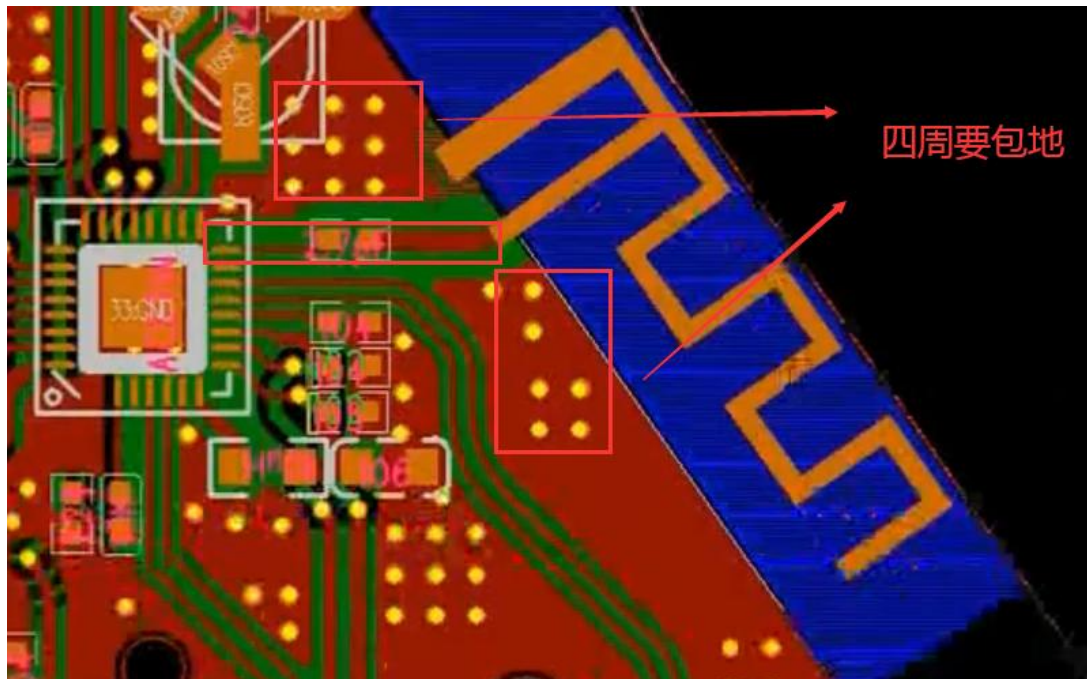
1.4 芯片的简单测试说明

序号	操作说明
第一步	搭建好芯片的外围电路，供电 3.3V 即可。蓝牙天线可以直接焊一根线即可
第二步	查询芯片的 2 脚是否开机有 1 秒钟的高电平输出，接个指示灯出来
第三步	连接好电脑的串口助手，看看芯片的 TX 脚是否有数据返回，115200 的波特率
第四步	做自己实际的板子，搭配 MCU 进行调试

1.5 硬件设计--脚位说明

序号	Layout 的注意事项
UART 注意点	1、我们芯片内部的 IO 电压是 3.3V 。主要看 1 脚 的输入电压 2、和外部的 MCU 相连接时，RX 和 TX 请串电阻，大概 100 欧姆就可以了。MCU 超过 3.3V 的 IO 电平，那么这个电阻可以加大到 1K。 3、芯片的 8 脚 是 KT6368A 的 TX，连接 MCU 的 RX。 7 脚 是我们的 RX，连接 MCU 的 TX
电源注意点	芯片的供电电压，最高位为 3.4V 。一定不能超过这个电压，最好给 3.3v
2 脚注意点	第 2 脚 ，为连接状态脚。连接成功输出高电平。未连接则是高组态。调试时建议接一个指示灯出来。或者连接到外部 MCU 。注意下拉一个 10K 电阻到地
细节描述	1、请严格按照我们给出的供电电压，去供电 。电源这一块没什么太大的讲究 2、蓝牙天线，按照我们给出的封装画就行。技术很成熟了，基本上距离都超过 15M 3、芯片的 7/8 脚两个必须预留测试点，这个是升级接口，以防万一要升级 升级的测试点排列，建议是 1/7/8/3 这 4 个脚顺序排列。引出测试点，很重要

1.6 硬件设计--蓝牙天线的说明







- (1)、注意芯片的蓝牙天线引脚，出去，要预留安全间距
- (2)、天线四周，一定要注意，包地处理
- (3)、天线周边一定要隔空，不要铺绿油，背面和正面不要有金属

1.7 硬件设计--蓝牙搭配的晶振说明

- (1)、由于蓝牙对频偏要求比较高，所以晶振的品质对蓝牙的性能至关重要，选型过程中必须保证晶振的一致性和稳定性。晶振的频率偏差必须 $\leq \pm 10\text{ppm}$ ，负载 CL 推荐 12pF。
- (2)、体积无要求的，推荐我 DEMO 上面的晶振，成本低，性能好
- (3)、体积要求小的，推荐 24M-3225 的，成本稍高，性能好

建议直接用我们配套的晶体，相信比外面随意采购的要优惠和质量保障

 <p>关注店铺收藏商品优先发货</p>	 <p>都不能用这种封装的 --质量差</p> <p>11.0592/12/24.576/8/3.57954/4/4.096/13.56 6/3.728/8/1.02/1.5/1.8/4.32/1.6/2/2.4/2.6/3.0/4.0</p>	<p>不可以用。千万不要用</p>
		<p>可以用。</p> <p>1、成本不敏感追求稳定性，推荐 3225 封装，质量超级好</p> <p>2、成本敏感，且体积无要求的推荐 M49-SMD-2PIN</p>
<p>左图为 3225 封装，右图为 M49-smd-2pin 封装。成本有区别</p> <p>晶振旁边不需要贴电容，不需要贴电容。可以预留，但是批量可以不用贴</p>		

1.8 硬件设计--统计到目前为止客户的错误案例-很重要

1、关于 KT6368Ahe MCU 之间的连接，以及电平转换的问题？	
2、关于晶振周围的匹配电容问题	
3、关于用户	

3. 串口通讯协议

AT 串口指令作为一种在控制领域常用的通信，我们进行了优化和定制，这样大大简化了用户使用的难度，请严格按照我们给出的指令格式进行操作

3.1 通讯格式

<p>支持异步串口通讯模式, 通过串口接受上位机发送的命令</p> <p>通讯标准: 115200 bps --- 用户可以通过串口指令设置详见3.2</p> <p>数据位 :8 停止位 :1 校验位 :none 流控制 :none</p> <p>注意: 所有的指令的设计, 都是有规律的, 不是随意划分的, 可以对照下面找一下规律</p>	
<p>控制指令格式: AT+<CMD>[<param>]\r\n ---- 所有的都是字符, 不是十六进制数</p>	
<p>数据反馈格式: <IND>[<param>]\r\n</p>	
数据特性	详细说明
AT+	控制指令是控制主机给 BT201的控制命令, 以 “AT+ ” 开始
<CMD>	后面紧跟<CMD>控制 , 通常是2个字符
[<param>]	如果 CMD 后面有参数, 则紧跟着 [<param>]
\r\n	最后以 ” \r\n” 结束, 字符型为换行, windows 就是回车键。 十六进制为0x0D, 0x0A
<IND>	1、数据反馈 是蓝牙把各种状态和数据信息反馈给主机, 以<IND>作为开头
	2、后面紧跟着的是芯片回传的参数

这里<CMD>重点说明:		
功能划分	命令	备注
公共指令特性	AT+C?	公共指令是以 AT+C 开头, 后面的 “?” 就是具体细化的功能命令

音乐指令特性	AT+A?	音乐指令是以 AT+A 开头，后面的“？”就是具体细化的功能命令
蓝牙指令特性	AT+B?	蓝牙指令是以 AT+B 开头，后面的“？”就是具体细化的功能命令

这里<CMD>重点说明：		
举例	命令	备注
控制指令 1	AT+CZ\r\n	代表系统复位
查询返回的结果 1	QA+01	详见 4.4.1 返回的查询信息永远是 Qn+xx 其中 n 和前面是相对应
查询返回的结果 2	QG+01	详见 4.2.12

3.2 通讯指令举例

公共部分--控制指令 -- 说明		
CMD	对应的功能	详细说明
AT+CT	设置波特率	后面有参数，详见3.3 举例：AT+CT01/r/n
AT+UT	设置蓝牙 BLE 的广播间隔	后面有参数，详见3.11 举例：AT+UT01/r/n
AT+CZ	芯片复位	芯片软复位，详见3.3 举例：AT+CZ/r/n
AT+CW	芯片恢复出厂设置	恢复出厂设置，清除所有之前记忆的参数，详见3.3 举例：AT+CW/r/n
AT+CL	芯片低功耗设置	详见3.7章节
AT+CF	2脚连接状态设置	【AT+CF01 代表未连接闪烁，连接常亮】【AT+CF00未连接低，连接高】
AT+CR	芯片上电回传信息关闭	详见3.10章节，注意默认是开启的
AT+BM	设置 BLE 蓝牙名称	详见3.4章节
AT+BN	设置 BLE 的 MAC 地址	详见3.4章节
AT+BD	设置 SPP 蓝牙名称	详见3.4章节
AT+BS	设置 BLE 连接密码	详见3.4章节，此功能没有实现，主要在于手机的兼容性不行
AT+BR	设置蓝牙的发射功率	详见3.12章节，不在意蓝牙发射功率就不用管这条指令，芯片默认最大
AT+QT	查询系统的波特率	详见3.3章节.返回的数据为
AT+QL	查询系统的低功耗状态	详见3.7章节.返回的数据为 QL+00
AT+TM	查询 BLE 蓝牙名称	详见3.5章节
AT+TN	查询 BLE 蓝牙地址	详见3.5章节
AT+TD	查询 SPP 蓝牙名称	详见3.5章节
AT+TS	查询 BLE 蓝牙连接密码	保留

测试推荐的指令

AT+BM1234\r\n -- 设置 BLE 的名称

AT+BN112233445566\r\n --ble 的地址

AT+BD223344\r\n -- 设置 SPP 的名称

AT+CT01\r\n

AT+CZ\r\n

AT+CW\r\n

AT+QT\r\n

AT+TM\r\n

AT+TN\r\n

AT+TD\r\n



3.3 指定芯片的波特率和复位和恢复出厂设置【CT】【CZ】【CW】

AT+CT01\r\n == 9600	AT+CT06\r\n == 256000	AT+CT11\r\n == 31250
AT+CT02\r\n == 19200	AT+CT07\r\n == 512000	AT+CT12\r\n == 2400
AT+CT03\r\n == 38400	AT+CT08\r\n == 230400	AT+CT13\r\n == 4800
AT+CT04\r\n == 57600	AT+CT09\r\n == 460800	
AT+CT05\r\n == 115200	AT+CT10\r\n == 1000000	
1、一旦设置了波特率之后，芯片会记忆。下一次开机，波特率就变成了您所设置的.当然可以查询[AT+QT]		
2、设置完波特率之后，请等待 1 秒钟，再发送复位[AT+CZ]，或者断电一下即可		
3、如果要恢复默认的波特率，请发送恢复出厂设置的命令，此时芯片会自动擦除所有的配置		
4、由于我们芯片的主频很高，所以尽量把串口的波特率调高，越高越好		

3.4 设置 BLE 蓝牙的名称和地址[BM][BN][BD]

AT+BMBLE-1234\r\n	设置蓝牙名称为“BLE-1234”
AT+BN112233445566\r\n	设置 BLE 的地址。手机端显示的地址是：66 55 44 33 22 11
AT+BDSPP-1234\r\n	设置蓝牙名称为“SPP-1234”

- 1、设置蓝牙名称之后，需要让芯片复位，发指令或者断电上电都可以，这样会显示新的蓝牙名称。我们默认的蓝牙名为“KT6368A-BLE”。设置的蓝牙名最长为“30”个字节，请不要超过这个范围
- 2、如果 AT 指令修改蓝牙名称之后，注意，你的手机端可能没有同步更新，还是显示之前的名称
 - (1)、因为你只修改了蓝牙的名称，蓝牙的 MAC 地址是没有变化的，所以手机端那边是不会更新名字
 - (2)、你要做的就是，换一台手机搜索试试，或者之前的手机删掉配对信息，重新在搜索

(3)、只要设置了蓝牙名，蓝牙名一定是更新过来了的，不用怀疑。芯片上电也会返回蓝牙名给您查看

3.5 查询 BLE 蓝牙的名称和地址[TM][TN][TD]

AT+TM\r\n	返回 TM+1234\r\n 代表蓝牙名为 1234
AT+TN\r\n	返回 TN+12345678AABB\r\n BLE 的蓝牙地址：0xBB、0xAA、0x78、0x56、0x34、0x12
AT+TD\r\n	返回 TD+SPP1234\r\n 代表蓝牙名为 SPP1234

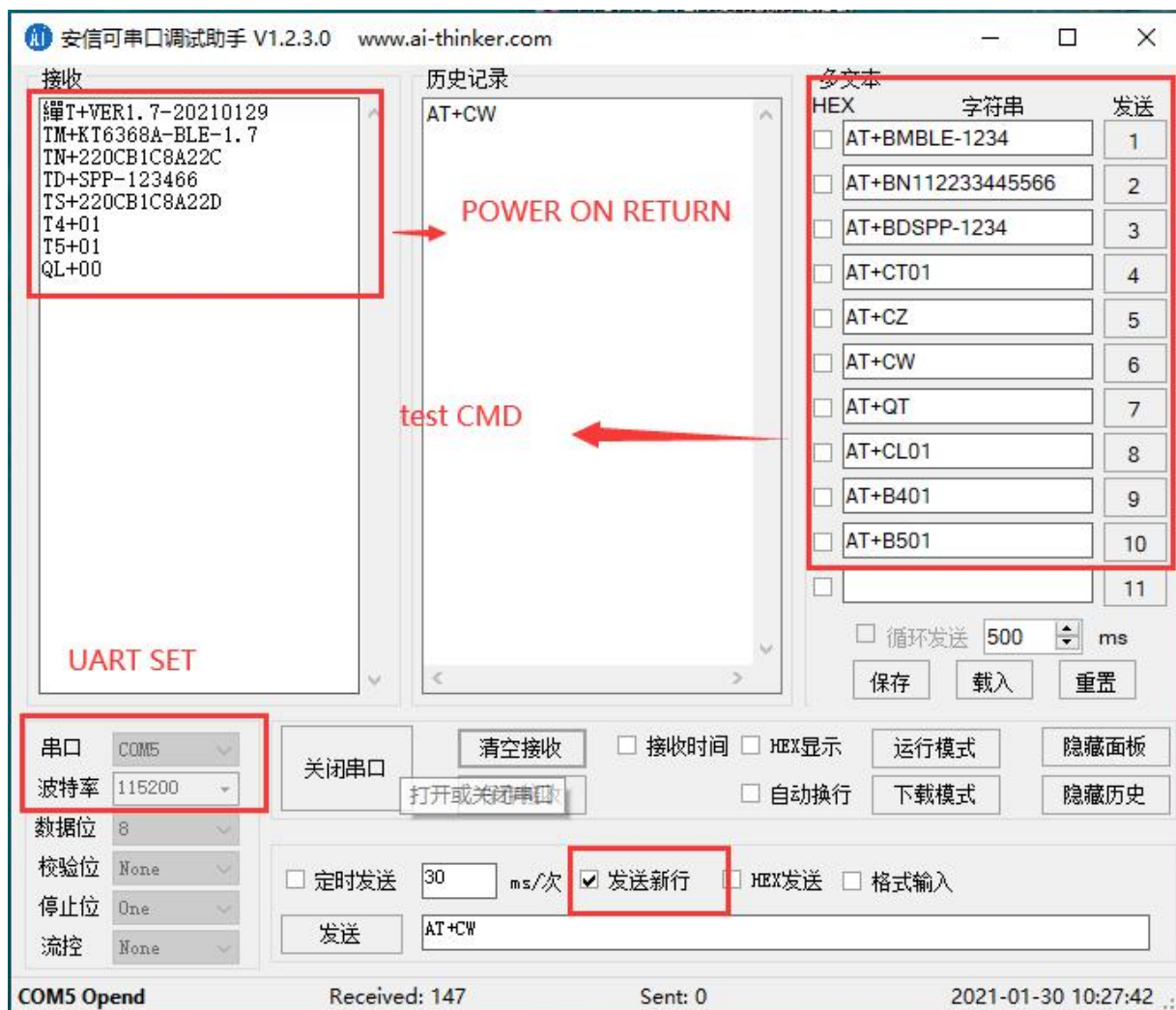
1、这里重点描述一下蓝牙的 MAC 地址：BLE 和 SPP 的 MAC 地址是共生的，所以设置一个就行了

(1)、芯片在第一次通电的时候，会自动生成蓝牙的 MAC 地址，并且是随机生存。这样做的好处是免除了 单独设置地址的问题

(2)、同样经过优秀的算法，出现重复的概率是百万分之一。蓝牙的 mac 地址是标准的，6 个字节

2、SPP 的地址，是在 BLE 地址的最高字节加 1 处理的 。所以只用设置 BLE 的地址即可。SPP 的地址也就没做查询指令，可以自己计算一下

3.6 芯片上电信息和串口调试助手



测试环境：KT6368A 测试板

串口软件：串口调试助手_aithinker_serial_tool_v1.2.3

1、接收窗口，芯片返回给电脑的数据。这个是固件的版本以及最后修改的日期

==》这个数据的返回，无任何意义。主要是方便客户，上电测试串口是否连接正常，以及查看芯片运行状态

==》芯片上电是一定会返回的，如果没有返回，说明硬件连接有误

TM+KT6368A-BLE-1.7	代表的是当前芯片的 BLE 的名称，以及对应手册的版本为 1.7
TN+220CB1C8A22C	代表的是当前芯片的 BLE 的地址
TD+KT6368A-SPP-1.7	代表的是当前芯片的 SPP 的名称，以及对应手册的版本为 1.7
TS+220CB1C8A22D	代表的是当前芯片的 SPP 的地址 此地址是根据 BLE 的地址计算得来的
T4+01	代表的是当前 BLE 功能是打开的，详见 3.8 章节
T5+01	代表的是当前 SPP 功能是打开的，详见 3.8 章节
QL+00	代表的是当前是正常工作模式，详见 3.7 章节

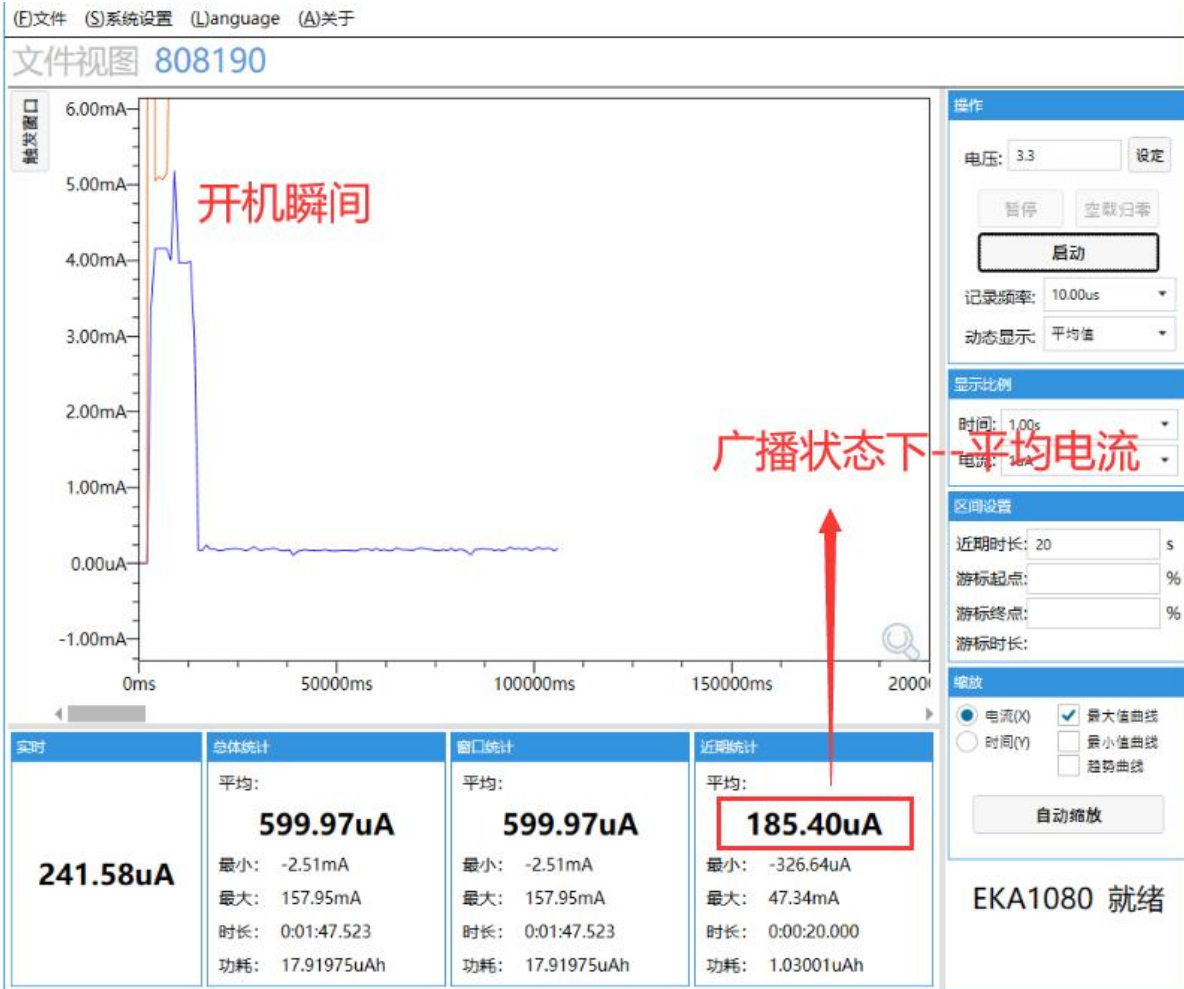
这里的很多返回的信息，用户可以不关注，因为这个存在的目的是方便客户初次调试的时候看

3.7 芯片低功耗指令说明【CL】

AT+CL00\r\n	不进入低功耗模式。下次上电有效。设置之后注意要重新上电
AT+CL01\r\n	进入低功耗模式。下次上电有效。设置之后注意要重新上电 --- 芯片默认进入此状态，不用设置

- 1、这个指令，是记忆型的，发送指令成功之后，芯片就存起来。下次上电就切换了。**同时发了这个指令芯片会自动复位**
- 2、这个指令，由于很多地方受限，所以默认是关闭的
- 3、设置低功耗之后。上电芯片的 UART 还是会主动返回相关的数据。
 - (1)、但是所有的 AT 指令全部失效了，因为芯片会进入低功耗，所有的外设全部关闭
 - (2)、当连接成功之后，芯片就处于正常工作状态。但是此时只具备透传的功能
 - (3)、所以需要设置 AT 指令的地方，必须切换回非低功耗模式，也就是 AT+CL00\r\n
- 4、当然芯片，出厂上电默认是，正常工作模式。
- 4、如果进入低功耗模式，芯片的所有 IO 口，都是高阻态。这点很重要
 - (1)、如果可以的话，芯片的 2/7/8 脚，接上拉电阻。来确定我们的 IO 状态
 - (2)、因为有的客户反映，芯片进入低功耗模式之后。**他的 MCU 不断的收到 FF 的数据**
 - (3)、所以这种应用，尽量用 KT6328A 的 2 脚来确定，芯片是否连接。未连接则不接收任何数据
- 5、设置为低功耗模式之后，芯片在未连接状态下。开机的前 5 秒可以识别 AT 指令，5 秒之后就不能识别 AT 指令
 - (1)、因为要低功耗，所以芯片的所有外设全部关闭
 - (2)、但是很多客户的应用可能是在低功耗状态下需要修改一些参数。所以设定 5 秒超时之后才进入低功耗而这 5 秒之类，是可以正常识别 AT 指令的
 - (3)、如果需要 AT 指令设置参数，尽量在连接状态下发送。因为连接之后，我们自动退出低功耗模式

	序号	电流	说明
AT+CL01 状态，进入低功耗 工作模式	开机瞬间	11mA	1、芯片开机需要初始化外设。瞬间电流比较大 2、这个时间维持 300ms，就进入低功耗状态了
	工作状态-未连接	20uA 4mA 交替	3、芯片正常工作状态，正常对外广播，处于一个睡眠、唤醒广播、睡眠这样的周期性状态。目的是为了节省功耗 4、周期 500ms。100ms 广播一次，400ms 睡眠 5、 广播一次电流就是 4mA 。进入睡眠，就变成 20uA
	工作状态-以连接	4.3mA	6、当连接成功之后，芯片就不再进入睡眠。而是一次处于工作状态了
AT+CL00 进入正 常工作模式	开机瞬间	25mA	1、芯片开机需要初始化外设。瞬间电流比较大 2、这个时间维持 300ms，就进入 5mA 工作状态
	不管连接还是未 连接。	5mA	3、芯片一直处于工作状态。电流很小的波动，忽略不计



- 1、可以看到开机瞬间的电流在 5mA ，随后降到 4mA 等待几秒之后，就进入低功耗广播状态了
- 2、低功耗的广播状态，平均电流是 185.4uA
- 3、最低的时候，是 20uA 。由于此 uA 表软件采样率不够，所以曲线上体现不出来



AT+VER2.0-20211111
 TM+KT6328A-BLE-2.0 --- 手机端会搜索到这个名字
 TN+3031E54D77D9
 T4+01
 T5+00
 QL+01 -- 进入低功耗模式

3.8 芯片 BLE 使能和 SPP 使能[B4][B5][T4][T5]

AT+B401\r\n	开启 BLE 的功能。当然 AT+B400\r\n 则是关闭了
AT+B500\r\n	关闭 SPP 的功能。当然 AT+B501\r\n 则是开启了
AT+T4\r\n	查询 BLE 功能是否开启。芯片会返回 T4+01或者 T4+00
AT+T5\r\n	查询 SPP 功能是否开启。芯片会返回 T5+01或者 T5+00

- 1、关闭 BLE 功能之后，必须重新上电，此功能才生效。当然开启也是一样的
- 2、只用设置一次，芯片自动保存参数，下一次不用设置了
- 3、关闭 BLE 功能之后，手机就搜不到 BLE 的名称了
- 4、关闭 SPP 功能之后，必须重新上电，此功能才生效。当然开启也是一样的
只用设置一次，芯片自动保存参数，下一次不用设置了。关闭 SPP 功能之后，手机就搜不到 SPP 的名称了

3.9 芯片返回的错误信息说明【ER】

ER+1\r\n	接收的数据帧不对。意思是 6368 芯片串口收到的数据，开头不是 AT
ER+2\r\n	接收的命令不存在，也就是你发的 AT+KK 这样的字符串查找不到
ER+3\r\n	接收的 AT 指令，没有收到回车换行，也就是\r\n
ER+4\r\n	发送的指令给的参数超范围了，或者指令的格式不对。请检查您的 AT 指令
ER+7\r\n	MCU 发送数据给手机，但是手机端没有打开 notify。在 ble 连接成功状态下
ER+8\r\n	保留--无意义

芯片内部对一些错误的状态，会进行实时的反馈。具体的请对照上面的表格

重点描述一下 notify [监听]，手机端的测试 APP 连接上蓝牙芯片之后，必须打开 notify。蓝牙芯片才能发送数据给手机。手机发数据给蓝牙芯片，用 write 这个特征就足够了。

3.10 芯片上电回传信息关闭指令【CR】

AT+CR00\r\n	关闭上电的回传信息。设置之后注意要重新上电
AT+CR01\r\n	开启芯片上电的回传信息。下次上电有效。设置之后注意要重新上电

- 1、有的客户反馈，芯片上电主动返回的信息，很烦人。所以我们就新增一个指令来关闭这个



- 2、注意，这个默认芯片出厂是打开的。关闭之后会存起来，就永久的关闭了
- 3、同时，也会关闭芯片主动返回的 OK 或者 ER+X 的回传信息

3.11 指定芯片的 BLE 的广播间隔【UT】

目前此功能，仅限于 KT6328A 的版本，也就是单模 BLE 的低功耗版本

对应的指令	代表的含义	参考的功耗
AT+UT00\r\n	0--对应--250ms 广播间隔	平均功耗是 300uA
AT+UT01\r\n	1--对应--500ms 广播间隔	平均功耗是 180uA
AT+UT02\r\n	2--对应--750ms 广播间隔	平均功耗是 140uA
AT+UT03\r\n	3--对应--1000ms 广播间隔	平均功耗是 100uA
AT+UT04\r\n	4--对应--1500ms 广播间隔	平均功耗是 70uA
AT+UT05\r\n	5--对应--2000ms 广播间隔	平均功耗是 62uA
AT+UT06\r\n	6--对应--3000ms 广播间隔	平均功耗是 40uA
AT+UT07\r\n	7--对应--4000ms 广播间隔	平均功耗是 30uA



- 1、一旦设置了广播间隔参数之后，芯片会记忆。下一次开机，广播间隔就变成了您所设置的. 不支持查询
- 2、但是芯片每次上电会主动的返回当前的连接间隔参数，详见上面的截图
- 3、如果要恢复默认的广播间隔，请发送恢复出厂设置的命令，此时芯片会自动擦除所有的配置
- 4、具体的间隔时间，还是要根据自己的产品，来定义。因为广播间隔越长，手机搜索到的时间就越长

3.12 芯片 RF 蓝牙发射功率[BR]

AT+BR01\r\n	设置蓝牙的发射功率为 1 级，发送之后马上生效，取值范围是[0---9]
AT+BR09\r\n	设置蓝牙的发射功率为9级，这个就是最大了

1、发送之后马上生效，取值范围是[0---9]，
2、设置 RF 功率值之后。断电自动保存，芯片会自动记忆，下次上电就不用重新设置了
3、芯片默认就是最大功率，也就是等级 9 。当然发射功率越高，距离越远，功耗相对要高一点，反之一样
4、如果没有必要，不用去修改这个参数

4. 蓝牙透传的详细说明--BLE

目前支持 BLE 纯数传，芯片可以实现透传。目前 BLE 和 SPP 均只能作为从。也就是“SERVER”端。

请注意，一旦蓝牙被连接之后，芯片自动进入透传模式。不再识别 AT 指令。一定要在 app 里面去搜索

4.1 BLE 的透传说明

- | |
|--|
| 1、单次吞吐的数据最大为 1024 个字节 ，支持 16 位或者 128 位的 UUID --- 128 位的需要特别定制 |
| 2、如果使用 BLE 作为数传，请连接模块的“KT6368A-BLE”这个蓝牙名 |
| 3、当然可以自己修改 BLE 的蓝牙名以及 MAC 地址了，通过 AT 指令 |

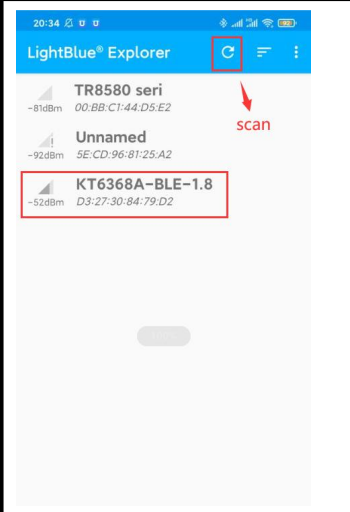
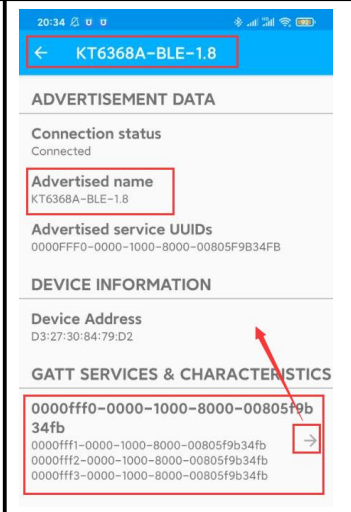
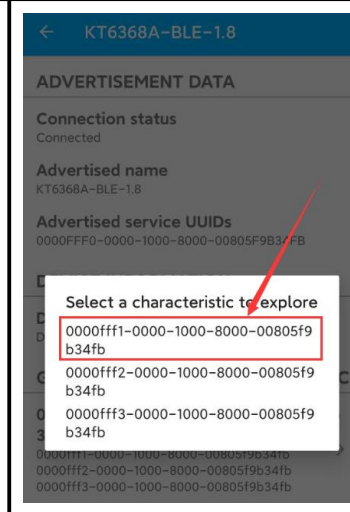
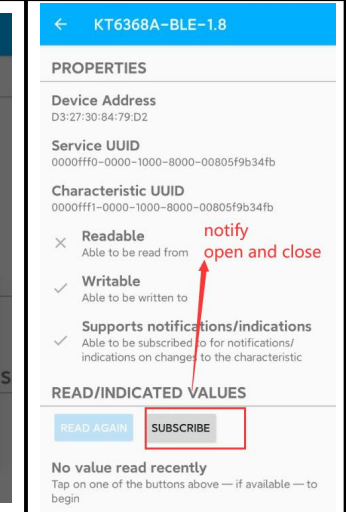
4.2 BLE 的 UUID 说明

- | |
|---|
| 1、主 UUID 是“FFF0” |
| 2、特征 1 的 UUID 是“FFF1”，特征是“WRITE_WITHOUT_RESPONSE ” “NOTIFY” |
| 3、特征 2 的 UUID 是“FFF2”，特征是“READ ” “NOTIFY” |
| 4、特征 3 的 UUID 是“FFF3”，特征是“WRITE_WITHOUT_RESPONSE” |
| 1、如果需要特别的 UUID，可以联系我们定制。 -- 请注意列清楚需求，特征，uuid 等等信息，越详细越好
BLE 透传效果演示： https://v.qq.com/x/page/q07660mlbta.html |

4.3 BLE 的测试说明



- | |
|---|
| 1、安卓手机的 ios 手机[苹果]，推荐使用“BLE 调试宝”软件 |
| 2、苹果的可以直接在“APP Store”里面搜索下载 |
| 3、安卓的，我们会在资料包里面提供安装的程序 |
| 4、请注意，安卓的手机也是可以测试 BLE 的，测试 BLE 不是一定只能用苹果的手机 |
| 5、安卓的 BLE 不是不能用，而是不好用，安卓的版本必须是在 4.3 版本以上的才支持 BLE |
| 6、正因为安卓的 BLE 不好用的原因，所以才会有双模，安卓用 SPP。苹果用 BLE |
| 7、因为苹果如果要用 SPP，这需要买 MFI 认证芯片，超级贵，目前也没人用了 |
| 8、如果默认没有修改过蓝牙名称的，连接“KT6368A-BLE”这个蓝牙名 |
| 9、BLE 测试说明演示视频： https://v.qq.com/x/page/o0766ubm78n.html |

4.4 BLE 的手机端 app 测试说明--lightblue 测试

			
<p>第一步：</p> <ol style="list-style-type: none"> 1、启动 lightblue 的 app 2、这里是安卓测试环境 3、Ios 的界面略有不同 4、要打开定位权限 5、当然也有很多其他的测试 app，操作略有不同 	<p>第二步</p> <ol style="list-style-type: none"> 1、连上芯片之后的界面 2、可以看到名字和服务 3、点击“箭头” 	<p>第三步：</p> <ol style="list-style-type: none"> 1、找到需要的服务 2、也就是 FFF1 3、点击 	<p>第四步：</p> <ol style="list-style-type: none"> 1、打开 notify 2、这样 KT6368A 收到数据之后，就可以发送给手机了 3、手机发数据，直接发就行了，KT6368A 会直接串口转发出去

1、注意测试的时候，最好打开手机的定位权限，因为很多 app 需要这个权限
2、无论是用户自己开发 app 还是微信小程序，操作的步骤和上面截图也是一样的
3、推荐用户只用第一个特征，也就是 FFF1 .他的特征是写和监听，足够使用了
4、基础的问题，请自行百度解决。其实这些描述，网上也是很容易找的，不复杂

4.5 BLE 的大数据量测试

 <p>上位机通过 uart 单次发送 1832 个字节的数据</p>	 <p>这个是接收完成的截图 总共的耗时：220ms 的样子</p>
--	---

以上测试的数据，是建立在我们芯片设置的连接间隔基础上的测试
其实缩短连接间隔，也是可以加快数据的通信。但是同时也增加的功耗

注意，这里手机收到的数据，还是基本遵循 20 个字节分包。因为我们芯片内部默认设置的最大包的长度是 20 个字节

正常的流程，是 APP 那边连接完蓝牙芯片之后，可以主动发起请求 MTU 【最大通信包长度】--网上可以自己搜搜

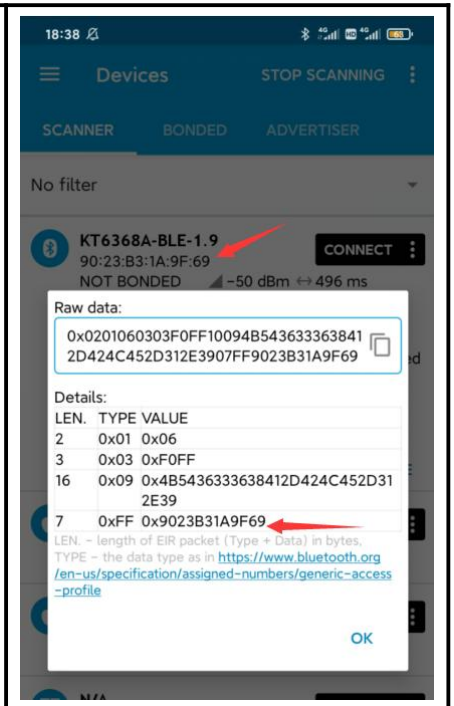
设置了 MTU 之后，单次的数据包，就不再是 20 个字节了。从而加快的数据交互的速度

4.6 BLE 的广播包数据说明--advertisData

这里我们在广播包里面，添加了芯片蓝牙的 MAC 地址
对比右边的截图，即可知道规律

这里我们称之为：advertisData，做这个的目的，有如下原因：

- 1、微信小程序开发：无法直接获取蓝牙芯片的 mac 地址，没有相应的 API，所以可以通过这个获取到，具体网上可以搜一下
- 2、APP 开发--IOS 端，也没办法直接获取 MAC 地址，也是通过这种方式得到蓝牙芯片的 MAC 地址
- 3、APP 开发--安卓端，没有这个问题，直接通过 API 时可以获取到蓝牙芯片的 mac 地址的。所以用不用这个功能，都无所谓



4.7 BLE 的 UUID 通过 AT 指令去修改的说明[U0][U1][U2][U3]

AT+U0F000\r\n	指定服务 UUID 为 F000
AT+U1F001\r\n	指定特征码1为 F001，他的特性是“写不应答”+“监听”
AT+U2F002\r\n	指定特征码2为 F002，他的特性是“读”+“监听”
AT+U3F003\r\n	指定特征码3为 F003，他的特性是“写不应答”

- 1、这里我们给出了三个特征，目的是为了兼容很多客户不同的需求
==》有的客户希望写和监听放在不同的特征上，这样手机和蓝牙的交互，数据可以更可靠
==》有的客户，仅仅只用传少量的数据，所以需要简单，就把写和监听，放在一个特征码上

2、用户设置的时候，最好这 4 个都设置一下，不管你用不用

3、如果没有 APP 的客户，建议使用特征码 1 和特征码 2。分开两个不同的特征

4、如果已经量产的产品，想换我们的芯片，则只需要修改这个 UUID 即可，**即使不满足我们也可以协助适配**

写不应答	= ATT_WRITE_WITHOUT_RESPONSE
监听	= ATT_NOTIFY
读	= ATT_READ

5. 蓝牙透传的详细说明-- SPP

Spp 走的还是经典蓝牙的 2.1 的协议，不推荐使用，新产品建议直接使用 BLE。要在系统里面先连接 KT6368A-SPP

5.1 SPP 的透传说明

1、单次吞吐的数据最大为 1024 个字节。需要连接“KT6368A-SPP-04”
2、如果使用 SPP 作为数传， 请不要主动连接模块的“KT6368A-BLE”这个蓝牙名 ，或者自己设置的 BLE 蓝牙名
3、注意 SPP 是属于经典蓝牙里面的一个子链路而已。
4、SPP 数传和 BLE 是互斥的，如果你只用 SPP 的数传，那么请关闭掉 BLE。

5.2 SPP 的透传效果演示说明

1、SPP 透传效果演示：<https://v.qq.com/x/page/b0766jqw0p5.html>

5.3 SPP 的透传测试说明

1、安卓手机的测试使用“蓝牙串口”这个 app，可以在“应用宝”里面下载
2、如果默认没有修改过蓝牙名称的，连接“KT6368A-SPP”这个蓝牙
3、SPP 测试说明演示视频： https://v.qq.com/x/page/e0766bz15fw.html

SPP 的大数据量的透传演示视频：

<https://v.qq.com/x/page/c0843j975hl.html> 测试的方法，可以看一下我们资料包里面的视频演示。

5.4 SPP 的连接密码说明[BE][B1]

AT+BE1234\r\n	设置蓝牙 SPP 的连接密码为“1234”，这个只能是数字，不能是字母
AT+B101\r\n	这个01代表0x01，打开配对密码，也就是下次连接需要输入密码才能连接 spp

1、注意 spp 的密码只能是“4”个字节，请不要超过这个范围。
2、同时如果你手机连过这个蓝牙芯片的 spp，即使设置了密码，也不需要输入密码了，因为互相已经认识，可以更换其他的手机去连接。或者修改一下当然蓝牙芯片的 MAC 地址
2、如果需要开启密码，发送 AT+BE1234\r\n，然后芯片重新上电就可以了
3、如果需要关闭密码，发送 AT+B100\r\n，关闭密码配对使能，然后芯片重新上电就可以了

6. 关于 AT 指令和透传数据的详细说明

1、目前我们的串口指令，支持 AT 指令，同时支持蓝牙数据透传

2、AT 指令，是存在于整个芯片的生命周期，只要芯片初始化蓝牙之后，那么蓝牙数据透传，就会一直在后台运行，无论是连接还是未连接状态，都支持 AT 指令

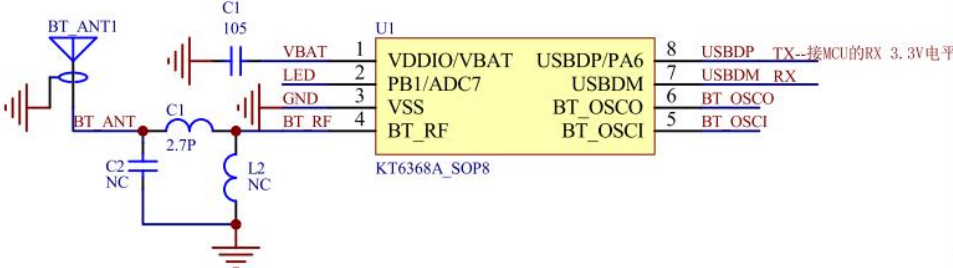
3、但是请留意，我们还有一个低功耗的模式，详见 3.7 章节的详细说明

问题 1	什么是蓝牙透传，有什么特点呢？
答疑	<p>1、蓝牙数据透传，指上位机 MCU 通过串口，发任何的数据，蓝牙芯片收到之后会直接转发给手机端</p> <p>2、同时，手机端发送任何的数据，蓝牙芯片都会通过串口下发给 MCU，通过串口 uart 的形式</p> <p>3、我们的方案中，蓝牙透传，是不需要任何的指令或者设置的</p>

问题 2	芯片是如何区分 AT 指令和透传的数据呢？												
答疑	<p>1、对于 MCU 发送的指令，只要不是正常的 AT 指令，我们都会透传出去，举例说明如下：</p> <table border="1"> <thead> <tr> <th>MCU 端发送的数据</th><th>说明</th></tr> </thead> <tbody> <tr> <td>AT+CT00\r\n</td><td>这个就是正常的 AT 指令，是不会被透传出去的。KT6368A 会直接处理</td></tr> <tr> <td>AT+CT00</td><td>这个就是异常的指令，是会被透传出去的，因为没有加换行，KT6368A 也会返回 ER+7</td></tr> <tr> <td>KT+CT00\r\n</td><td>这个也会被透传出去，因为他不是 AT 指令开头</td></tr> <tr> <td>1234AT+CM00\r\n</td><td>这个也会被透传出去，因为他的起始数据不是 AT 开头。AT 的指令仅仅只是在中间，所以会被透传</td></tr> <tr> <td>12121212121212kkk</td><td>这个就是纯粹的透传数据了，所以会被透传至手机</td></tr> </tbody> </table> <p>至于这些透传的数据，如何去处理，就留给聪明的您去自由发挥啦</p> <p>2、对于手机端发送的数据，则更容易理解 --- SPP 和 BLE 透传说明</p> <p>(1)、任何数据都是透传下去的。哪怕手机端发送的 AT+CT00\r\n 这种正常的指令，也是被透传的</p> <p>蓝牙芯片收到之后，也是不会处理的，只会串口输出给 MCU</p>	MCU 端发送的数据	说明	AT+CT00\r\n	这个就是正常的 AT 指令，是不会被透传出去的。KT6368A 会直接处理	AT+CT00	这个就是异常的指令，是会被透传出去的，因为没有加换行，KT6368A 也会返回 ER+7	KT+CT00\r\n	这个也会被透传出去，因为他不是 AT 指令开头	1234AT+CM00\r\n	这个也会被透传出去，因为他的起始数据不是 AT 开头。AT 的指令仅仅只是在中间，所以会被透传	12121212121212kkk	这个就是纯粹的透传数据了，所以会被透传至手机
MCU 端发送的数据	说明												
AT+CT00\r\n	这个就是正常的 AT 指令，是不会被透传出去的。KT6368A 会直接处理												
AT+CT00	这个就是异常的指令，是会被透传出去的，因为没有加换行，KT6368A 也会返回 ER+7												
KT+CT00\r\n	这个也会被透传出去，因为他不是 AT 指令开头												
1234AT+CM00\r\n	这个也会被透传出去，因为他的起始数据不是 AT 开头。AT 的指令仅仅只是在中间，所以会被透传												
12121212121212kkk	这个就是纯粹的透传数据了，所以会被透传至手机												

7. 常见问题集锦

问题 0	KT6368A 是什么？有什么功能？特点是什么？适用于什么场景？配什么晶振呢？KT6368A 批量有优惠吗？ 蓝牙天线预留的元器件怎么办，焊还是不焊？
回答	<p>1、KT6368A 芯片属于蓝牙芯片，支持蓝牙 5.1 版本 BLE。同时支持 2.1 版本的 SPP 功能</p> <p>2、KT6368A 芯片支持连接手机，进行数据的双向交互，俗称“蓝牙透传”。通过 UART 接口 ==》支持常用的 AT 指令，如：设置名称、设置地址、设置波特率等等。详见手册</p> <p>3、KT6368A 芯片最大的特点，就是成本低，使用简单，SOP8 的封装，也便于生产</p> <p>4、KT6368A 芯片，适用于纯数据通讯的场合，如：客户自己开发 APP、微信小程序等等</p> <p>5、目前 KT6368A 的程序，只做了从机版本，只能和手机连接</p> <p>6、搭配 24M 的晶振，参数是 12pF 的负载，精度是 10ppM 。当然可以是 3225 封装或者其他晶振的选择，直接影响的是蓝牙的频偏，也就是蓝牙距离，所以别随使用，到时候搜不到蓝牙名，就又跑来问为什么了，我们有提供晶振的样品。可以顺便拿几个回去测试</p> <p>晶振的电容不用焊，建议预留，我们开机芯片会自动校准晶振的负载电容，软件处理的</p> <p>8、芯片批量基本没什么优惠了，价格超级敏感的，请选择其它</p> <p>9、蓝牙天线脚，预留的元器件，做样品直接不焊，接一个 C1 的电容即可。批量建议预留，预防做认证，或者天线要求极高的场合 。只接 C1 电容蓝牙距离也是妥妥的超过 10 米以上</p>

问题 1	KT6368A 有测试板吗？ 拿到芯片如何开始测试呢？ 有什么硬件上的注意事项？
回答	<p>芯片是 SOP8 封装的，总共的引脚就很少很少，使用也很简单。暂时没有测试模块</p>  <p>1、1 脚供电。然后对地焊一个 105 的电容就够了。或者不接也行。量产加上</p> <p>2、蓝牙天线，直接焊一根线就可以了，连接到芯片的 4 脚。实际做产品就加个 2p7 电容</p> <p>3、主要是晶振比较难焊，不要紧，可以配套我们给的晶振，M49 2 脚的焊一下就可以了</p> <p>4、剩下的就是串口了，因为是 3.3V 的电平，所以 3.3V 的 mcu 直接直连即可</p> <p>5、初次调试，建议使用串口调试助手调试 。USB 转 TTL 的选用 CH340G，某宝很多</p> <p>6、为什么我们不做测试版，主要是成本的原因，所以麻烦客户自己动手</p>

问题 2	KT60368A 支持微信小程序吗？ 默认的 uart 波特率是多少？
回答	1、微信小程序，只是用到了 BLE 而已。也就是说支持 BLE 就可以支持微信小程序 2、芯片是 BLE5.0 的协议，微信小程序需要客户自己开发。我们只是透传，无其他作用 3、芯片给的 uart 缓存是 1K 字节。默认的波特率是 115200

问题 3	KT6368A 这颗芯片供电电压多少 V？ 电流多少？ 透传的速率是多少 BLE 和 SPP
回答	1、建议给 3.3V 的电压【2.2V--3.4V】。 2、开机瞬间电流是 26mA。稳定大概 1 秒左右，就降到 4mA 左右 3、芯片给的 uart 的缓存是 1K 字节，默认波特率是 115200 4、对于 BLE 的速率，我们没有做完整的测试，需要高速传输的请自己测试一下 5、SPP 的传输，建议是单次最高不超过 512 字节一包数据，传输速率建议自己测 6、BLE 的传输速率，由于不同手机版本，都会有差别。所以速率没办法统一说明，用户自己测

问题 4	如何区分 AT 指令和串口透传数据？ 如何知道蓝牙是否连接？
回答	1、AT 指令，在任何状态下都有效 2、 只要蓝牙连接成功之后，就进入透传了，AT 指令还是可以正常被识别，详见章节 6 3、这个要看芯片的第 2 脚。未连接输出低电平。连接成功输出高电平 4、当然，你可以接一个指示灯来看。或者也可以连接到 mcu 的 gpio 上面

问题 5	如何确定芯片是否工作正常呢？以及串口接线正常呢？
回答	1、芯片上电瞬间，2 脚会输出 1 秒钟高电平，然后马上拉低。所以接 1 个指示灯来看一下状态。 2、芯片上电串口是一定会返回信息的。 注意是一定，如果没收到，说明串口有问题

问题 6	支持单芯片出货吗？ 芯片是什么参数？什么包装？芯片出货稳定吗
回答	1、芯片是 sop8 封装，管装，100 片一管。当然量大可以自己去编带 2、芯片出厂会烧录好固件，用户可以直接使用 3、芯片出货很稳定，因为这个是大品类的应用，如自拍杆、防丢器、等等量大的产品用的多所以成本就很低， 4、另外不支持讲价。价格也没什么空间了，请留意

问题 7	支持修改 uuid，以及蓝牙名和蓝牙 MAC 地址吗
回答	1、支持修改蓝牙名，以及蓝牙 MAC 地址 2、当然也支持 UUID 和设置和读取了，详见 4.7 章节。但是特征不支持修改

问题 8	支持单芯片出货吗？ 芯片是什么参数？什么包装
回答	4、芯片是 sop8 封装，管装，100 片一管。当然量大可以自己去编带 5、芯片出厂会烧录好固件，用户可以直接使用

问题 9	硬件设计，有什么需要注意的地方吗？
回答	<p>1、请严格按照我们给出的供电电压，去供电。电源这一块没什么太大的讲究</p> <p>2、蓝牙天线，按照我们给出的封装画就可以了。因为技术很成熟了，所以基本上距离都超过 15M</p> <p>3、芯片的 7/8 脚两个必须预留测试点，这个是升级接口，以防万一要升级</p> <p>4、升级的测试点排列，建议是 1/7/8/3 这 4 个脚顺序排列。引出测试点，很重要</p> <p>5、KT6368A 芯片的串口与 MCU 之间的连接，一定要串电阻，TX 和 RX 都要加，100 欧姆</p>

问题 10	支持买几个样品，帮我修改波特率到 9600 吗？
回答	<p>1、原则上不支持修改，因为几个样品，客户自己动手发 AT 指令改一下。我们默认是 115200</p> <p>2、实在要修改，收人工费 500。</p>
问题 11	支持按照我们特定的 uuid，以及服务，然后修改出样品吗？
回答	<p>1、原则上，不支持修改。因为样品阶段是给客户测试功能的。用户可以先做硬件，后期确实是做产品的，我们会配合修改的。</p> <p>测试芯片的性能。不可能几块钱的东西，我们都要工程师参与配合修改，这样效率太低了</p> <p>2、实在需要修改的，可以，收人工费 500 修改</p>

问题 12	KT6368A 大批量订货有货吗？后续能保证供货吗？
回答	<p>1、大批量订货是有货的，大可放心。除非那种单次超过 200K 以上的订单，需要提前打招呼以外。其他的数量，无需提前沟通。注意是 不用打招呼，直接下单就可以了</p> <p>核心的重点，是货您不用担心，就怕您没订单，一直问问问，这个就很麻烦了</p> <p>2、后续也会有长期有货的，无论订单多少，我们仓库一般都会做库存，大概保持在 300K 的。</p>

问题 13	KT6368A 用 USB 转 TTL 连接好之后，在 PC 端通过串口调试助手，就能和手机通讯了吗？
回答	<p>1、手册写的很清楚了。答案是肯定的，在我们给出的测试方法里面，就已经有描述了，请详细看一看，不懂网上查一下</p> <p>2、首先，要保证串口能连接好。其次要在手机上面打开测试 app。具体看我们的测试说明</p>

问题 14	KT6368A 连接好手机的测试 APP 之后，一定要打开 notify 呢？那我实际提交给 APP 开发人员，该如何说明呢？
回答	<p>1、是的，一定要打开 notify。串口发给 KT6368A 芯片的数据，才能通过蓝牙发送给手机这个蓝牙的标准，一定要这么做才行，</p> <p>2、做 app 开发或者小程序开发的工程师，都知道连接蓝牙芯片之后，主动打开 nortify 的，这个是固定的 API。做单片机开发的工程师可以不用管这个</p>

问题 15	KT6368A 为什么一定要在 app 里面连接呢？系统设置的蓝牙界面可以看到但是连不上？
回答	<p>1、注意，BLE 一定要在 APP 里面搜索和连接。是一定，无论是安卓还是 IOS 的设备，都一样</p> <p>2、同时，SPP 也要在安卓的 app 里面搜索和连接。注意 IOS 设备是没有 SPP 的</p>

问题 16	为什么我用串口调试助手发送 AT 命令，蓝牙芯片会给我返回 ER+1 或者 ER+3 呢？
回答	<p>1、对于这个问题，您第一时间，就应该查阅芯片手册 3.9 章节，里面都有错误信息的意思解答</p> <p>2、ER+1 ---- 代表的是芯片接收串口的数据，开头的两个数据，不是 AT 。而是其他</p> <p>3、ER+3 ---- 代表的是芯片接收串口的数据，末尾不是回车换行，意思是不是\r\n</p> <p>4、再次说明，这个东西测试很简单，能自己排查一下就自己排查一下，很多客户一上来就说</p>

问题 17	为什么我在蓝牙连接成功之后，发送 AT 指令给蓝牙芯片，没有透传出去呢？
回答	详细查阅一下我们的手册，章节 6，有详细说明的

问题 18	为什么我用的 KT6328A 版本，芯片开始还能接收 AT 指令，过几秒钟就不行了呢？
回答	 <p>详见里面的说明文档，看是不是购买的低功耗的版本，因为低功耗版本的机制比较特别一点</p>

问题 19	为什么我用的 KT6368A 双模的芯片，spp 能够正常连接，但是 ble 不能连接？
回答	<p>1、ble 的连接一定要用 app 去连_不要在系统设置里面去连接，app 可以用我们提供的 lightblue ，或者 nrf conenct 或者其他，很多很多</p> <p>2、还有一种情况，客户的晶振频偏过大，导致 ble 连不上，此时注意找我们客服买配套的晶振</p>

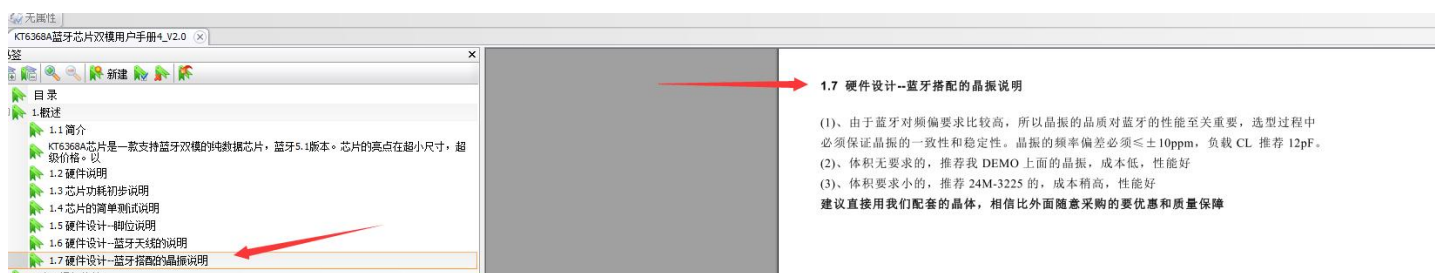
剩下的请移步至章节 10 的特别说明

8. 新增特别的说明

最近一段时间，问的最多的问题：

为什么我焊上 KT6368A 的芯片到板子上面，做了几十个板子，有的板子手机端搜不到蓝牙信号，有的手机能搜索但是连接不上，有的手机能连上，但是很容易掉线？？？？？

==》面对这个问题，我们给出的基本判断，就是用户随便使用晶振，导致的问题，我们手册里面硬件设计部分，也单独把这个注意事项拿出来描述了的



==》晶振一定不能随便用，这里再次说明一下，希望引起重视

==》KT6368A 芯片的不良率是非常非常低的，因为出厂之前，都要烧录程序。而烧录程序是所有的脚都要被用到的，所以即使有不良，也会是在烧录阶段就挑出来了，给到客户的基本上不良率是低于千分之一的

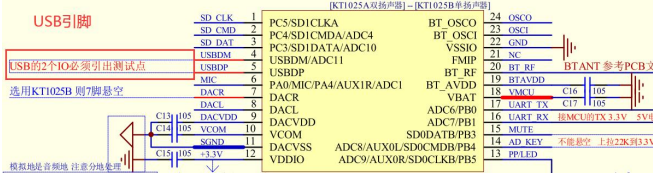

==》实在排查不出来原因，我们建议到淘宝上面去买一个测试模块，搜索关键词“KT6368A 蓝牙芯片”

目前 KT6368A 蓝牙芯片，总共开发了三个版本，他们硬件 IO 定义是一模一样

三个版本也都是为了不同的应用场景和客户需求而开发。

版本	对应的版本功能说明
KT6368A 双模版本	1、此版本不具备低功耗功能，开机就是 15mA，等待 1 秒后是 6mA 的平均电流 2、支持 BLE 和 SPP 同时存在 3、不太在乎功耗的场景，可以选择此版本
KT6328A 单模低功耗版本	1、只支持 BLE，不支持 SPP 2、超低功耗版本，广播平均电流是 100uA。当然根据不同的应用场景，可以设置不同的广播周期，来优化芯片整体的平均功耗 3、连接上之后维持 5mA 的电流
KT6368A 主机版本	1、支持连接我们配套的从机芯片，KT6368A 或者 KT6328A 这两个任意一个 2、连接成功之后，可以进行双向数据透传 3、当然支持 AT 指令修改目标从机的名称

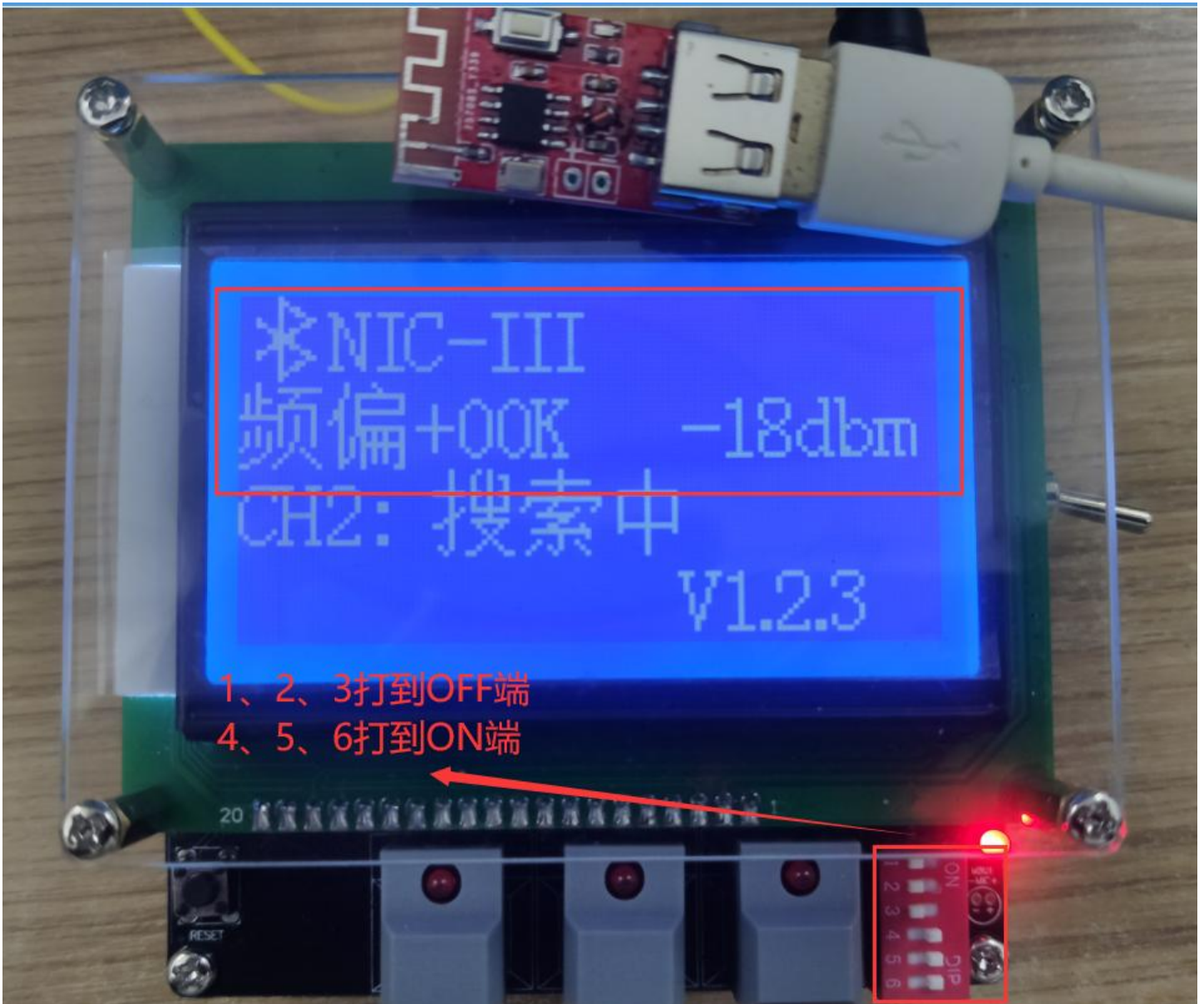
9. 新增批量生产说明

注意点	详细描述
1、蓝牙搭配的晶振	<p>(1)、芯片搭配的晶振是 24M ， 负载 12pF ， 精度 10ppm</p> <p>(2)、这个很关键，直接关系到蓝牙的稳定性、距离等等参数</p>
2、USB 留测试点	<p>(1)、一定要把芯片的 USB 引脚接出来</p> <p>(2)、如果有 USB 座子，那就更好，预防后续的升级</p> 
3、生产工具 蓝牙自动测试盒	 <p>(1)、贴片完成之后，需要对 PCBA 的蓝牙性能进行测试，最重要的是频偏</p> <p>(2)、使用我们标配的蓝牙测试盒，可以测试 蓝牙音频、通话、BLE。全自动的</p>

1、第一步，做样品阶段，一定要重视晶振的选择。选用测试盒测试一下晶振的频偏，购买晶振一定要找原厂，或者我们。**不要找华强北，不要找华强北，不要找华强北**
必须要求，换了晶振供应商之后，挑选 5-10 个晶振，在常温下上板测试一下频偏，一致性比较好才能确定这个供应商。同时要注意 蓝牙芯片的供电，不要有大的电压波动，不要有强的干扰

2、大批量生产阶段，贴好 PCBA 之后，都必须要用测试盒测试一下频偏。只要能连上就可以了，这样就证明蓝牙的通路是好的，蓝牙这一关就算过了

3、详细的测试盒设置、以及测试效果，以及拨码开关设置，如下图：



具体的含义，可以看测试盒背面的标签纸。

这个的设置，只测试 BLE 。Ble 正常了，。其实 spp 是不用测试的，因为是共享一个 RF【射频电路】

10. 新增一些使用注意事项--很重要!!!

这里，标记出来一些生产有问题的客户的设计不良，方便其他客户更顺利的生产

10.1 关于 KT6368A 的串口和 mcu 之间的电平转换以及串电阻

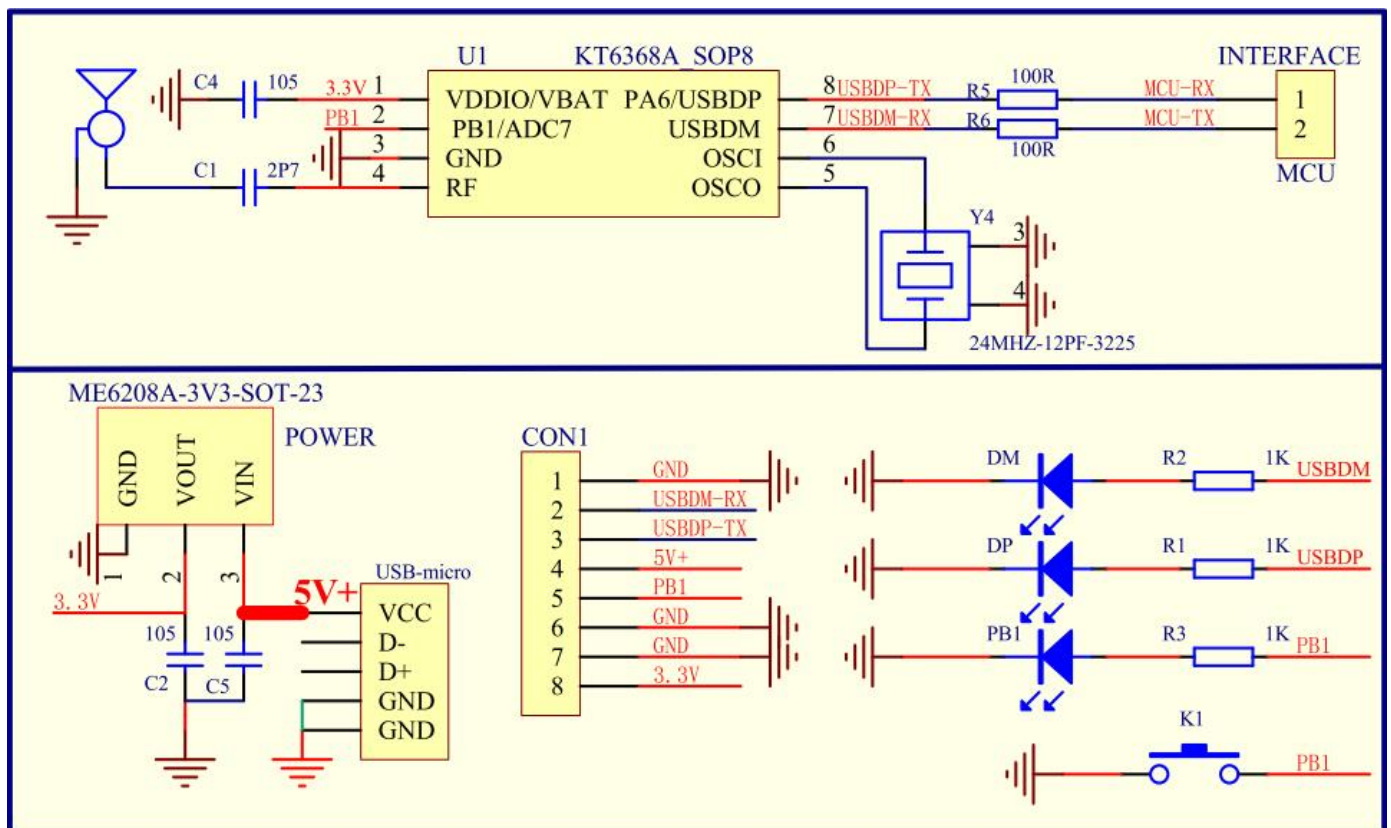
KT6368A 和 MCU 之间的串口连接，必须串电阻，TX 和 RX 之间都要串电阻

因为 KT6368A 的芯片内核工作电压比较低，所以 MCU 的串口 TX【注意是 MCU 的 TX】一般都是输出高电平，所以会给 KT6368A 倒灌，导致 KT6368A 工作异常或者不能顺利的复位，这是一个隐患

1、目前测试使用 CH340G 的串口板，对接 KT6368A 的芯片，TX 和 RX 分别都串 1K 的电阻。测试效果非常的好，不存在漏电，也不存在异常

2、测试 TX 和 RX 分别都串 100 欧姆的电阻，效果非常的好，没有漏电

==》推荐串 100 欧姆的电阻，如果是 5V 的 MCU，那么建议串 1K 的电阻



如上图的 R5 和 R6 的设定

10.2 关于 KT6368A 的晶振重要说明

其实这个，在前面的章节都很清楚的描述了，但是很多的客户生产依然没有注意可以详细的回看 1.7 章节的“硬件设计--蓝牙搭配的晶振”说明

结论：晶振旁边的匹配电容一定不要贴，可以预留，一定不要贴。不放心的话可以拿“蓝牙测试盒”去测一下晶振和 KT6368A 芯片起振频偏，详见“章节 9--批量生产说明”

10.3 关于 KT6368A 芯片修改蓝牙名_波特率_等等需要记忆的参数重要说明

KT6328A 或者 KT6368A 芯片在第一次上电的时候，系统内部有很多很多的校准操作这个时间的消耗大概是 2.5 秒。所以串口发指令必须要上电 3 秒左右才能发指令但是第二次或者第三次上电，以及以后上电，时间消耗大概是 500ms。

1、所以用户在使用过程中，尤其那种固定上电时间，发 AT 指令修改蓝牙名的操作一定要注意好这个发送的时间，不然会导致一些奇怪的问题

3、建议芯片上电 2.5 秒---3 秒之间去修改蓝牙的参数，比如：蓝牙名、地址、波特率等等需要记忆的参数

@@什么是芯片的第一次启动？

- 1、芯片我们经过烧录器烧完之后，直接出货给客户，客户拿到芯片之后，只要没有通电，那么芯片就没有启动过，称为第一次启动
- 2、只要芯片通过电，那么就代表芯片以后的上电就是第二次或者第三次，或者以后
- 3、如果在芯片上电校准期间，断电了，再上电会继续没校准完的参数，继续校准

@@总之：--- 希望能认真的留意

- 1、为了保证芯片的稳定性，发指令设置参数，必须是等待 2.5 秒之后才能发
- 2、或者等待蓝牙芯片**返回初始化数据之后**，才能发 AT 指令设置参数，比如：蓝牙名、地址、波特率等等
- 3、芯片第一次启动，不仅时间长，而且电流大很多，高峰可以达到 30mA 的启动电流

10.4 关于 KT6368A 芯片的供电问题

有部分客户在使用 KT6368A 的时候，选择使用 MCU 的 IO 口去给 KT6368A 或者其他版本的芯片供电，这样做是不可取的，原因如下：

1、有的 MCU 的 GPIO 驱动电流并不够

2、KT6368A 的芯片，是周期性的平均电流，也就是广播的时候瞬间电流能达到 25mA，但是持续时间很短，大部分时候都是不广播，所以平均电流才小

==》这个的意义就是说，MCU 的 GPIO 一般都是扛不住广播的瞬间电流，导致电压被拉低，从而导致 KT6368A 芯片的工作异常，这个很重要

结论如下：

不可以用 MCU 的 GPIO 去直接供电，一定要加 LDO 去供电，最舒服的供电电压是 2.8V--3.3V 之间，超过 3.4V，芯片是会被烧掉的