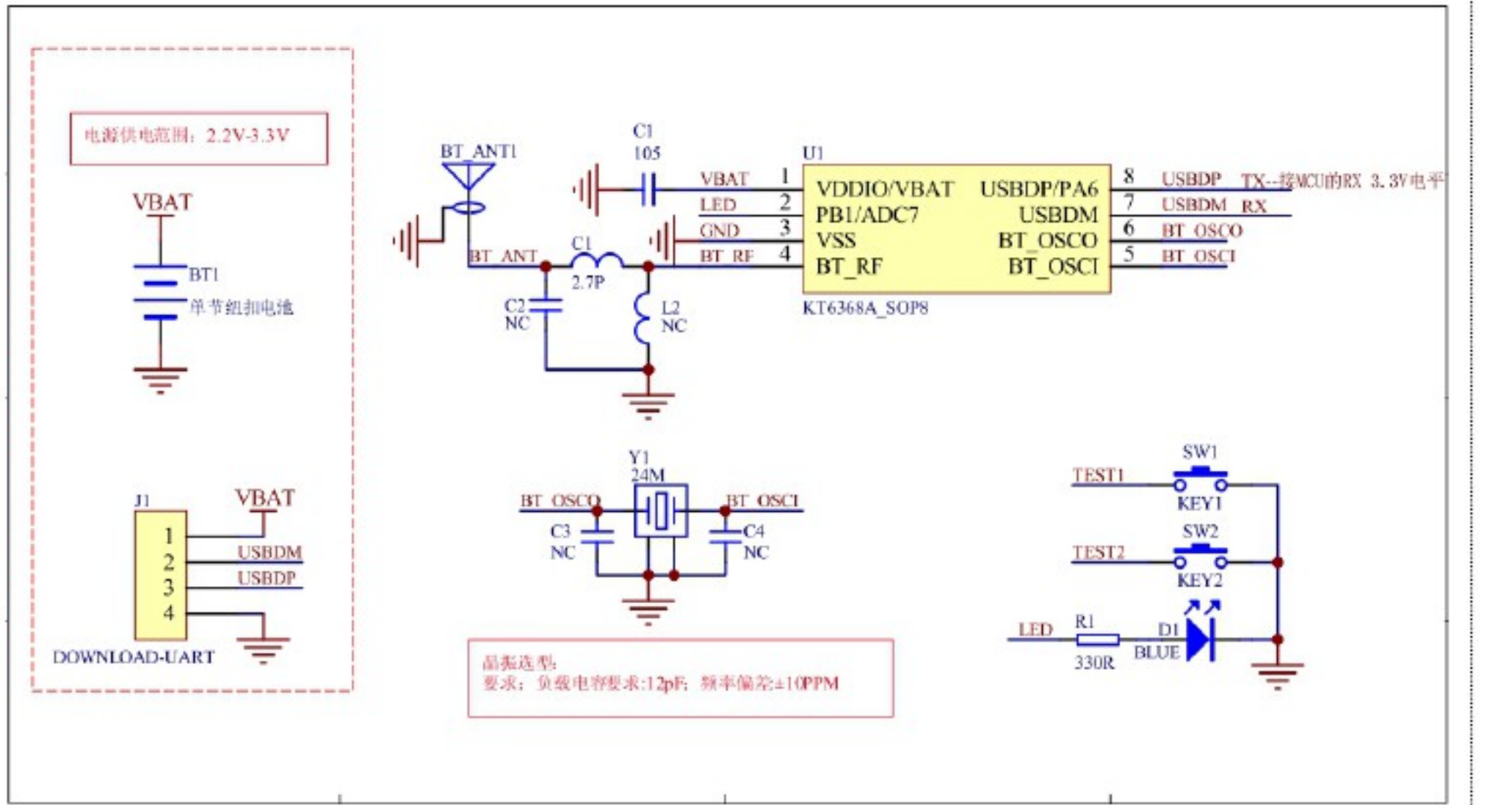


KT6368A 芯片使用手册

文件状态:	文件标识:	外部公开
[] 草稿	当前版本:	V1.8
[√] 正式发布	作 者:	
[] 正在修改	完成日期:	2021-02-20



目 录

1. 概述.....	3
1.1 简介.....	3
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.....	3
1.3 芯片功耗初步说明.....	4
1.4 芯片的简单测试说明.....	4
1.5 硬件设计--脚位说明.....	4
1.6 硬件设计--蓝牙天线的说明.....	5
1.7 硬件设计--蓝牙搭配的晶振说明.....	5
3. 串口通讯协议.....	6
3.1 通讯格式.....	6
3.2 通讯指令举例.....	7
3.3 指定芯片的波特率和复位和恢复出厂设置【CT】【CZ】【CW】.....	8
3.4 设置 BLE 蓝牙的名称和地址【BM】【BN】【BD】.....	8
3.5 查询 BLE 蓝牙的名称和地址【TM】【TN】【TD】.....	8
3.6 芯片上电信息和串口调试助手.....	9
3.7 芯片低功耗指令说明【CL】.....	10
3.8 芯片 BLE 使能和 SPP 使能【B4】【B5】【T4】【T5】.....	11
3.9 芯片返回的错误信息说明【ER】.....	11
4. 蓝牙透传的详细说明--BLE.....	12
4.1 BLE 的透传说明.....	12
4.2 BLE 的 UUID 说明.....	12
4.3 BLE 的测试说明.....	12
4.4 BLE 的手机端 APP 测试说明.....	13
5. 蓝牙透传的详细说明--SPP.....	14
5.1 SPP 的透传说明.....	14
5.2 SPP 的透传效果演示说明.....	14
1、SPP 透传效果演示： HTTPS://V.QQ.COM/X/PAGE/B0766JQW0P5.HTML	14
5.3 SPP 的透传测试说明.....	14
6. 关于 AT 指令和透传数据的详细说明.....	15
7. 常见问题集锦.....	16

1. 概述

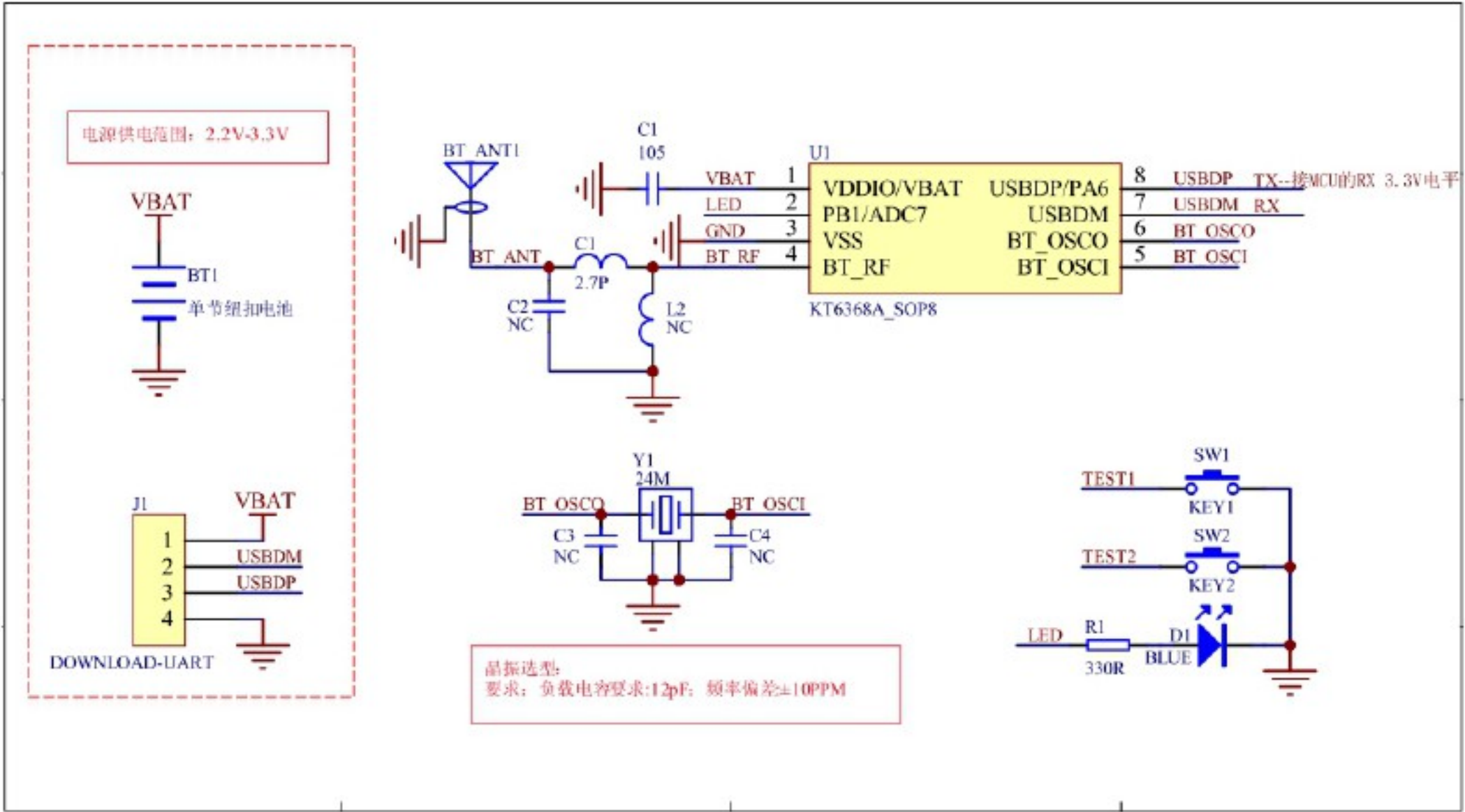
1.1 简介

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]

1.3 芯片功耗初步说明

1、注意此功耗，只在 AT 指令设置 AT+CL01 开启低功耗使能之后，才是如下列表的方式		
2、芯片默认出厂，是没有进入正常工作模式。方便客户测试 详细的，请参见 3.7 章节的说明。		
序号	电流	说明
开机瞬间	25mA	1、芯片开机需要初始化很多外设。所以瞬间电流比较大 2、但是这个时间仅仅维持 300ms，就进入低功耗状态了
工作状态-未连接	20uA 4mA 交替	1、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。其实目的是为了节省功耗 2、周期是 500ms。100ms 广播一次，400ms 就睡眠 3、广播一次电流就是 4mA 然后进入睡眠，就变成 20uA
工作状态-以连接	4. 3mA	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

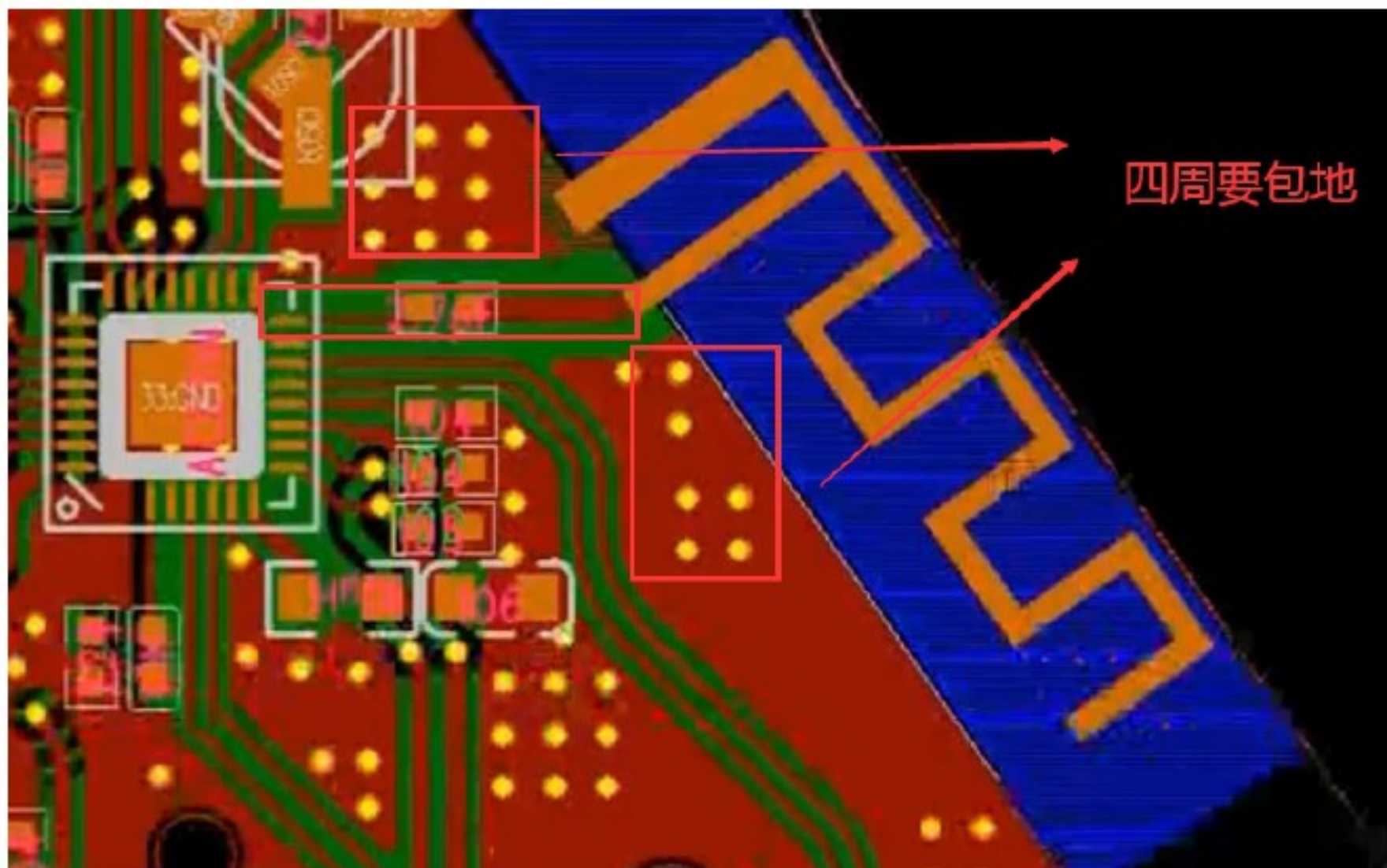
1.4 芯片的简单测试说明

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 I/O 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 I/O 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。 9 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
2 脚注意点	第 2 脚，为连接状态脚。连接成功输出高电平。未连接则是高阻态。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

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

支持异步串口通讯模式, 通过串口接受上位机发送的命令	
通讯标准: 115200 bps —— 用户可以通过串口指令设置详见3.2	
数据位 :8	
停止位 :1	
校验位 :none	
流控制 :none	
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	
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	
数据反馈格式：<IND>[<param>]\r\n	
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+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+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+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+BD SPP-1234\r\n	设置蓝牙名称为“SPP-1234”

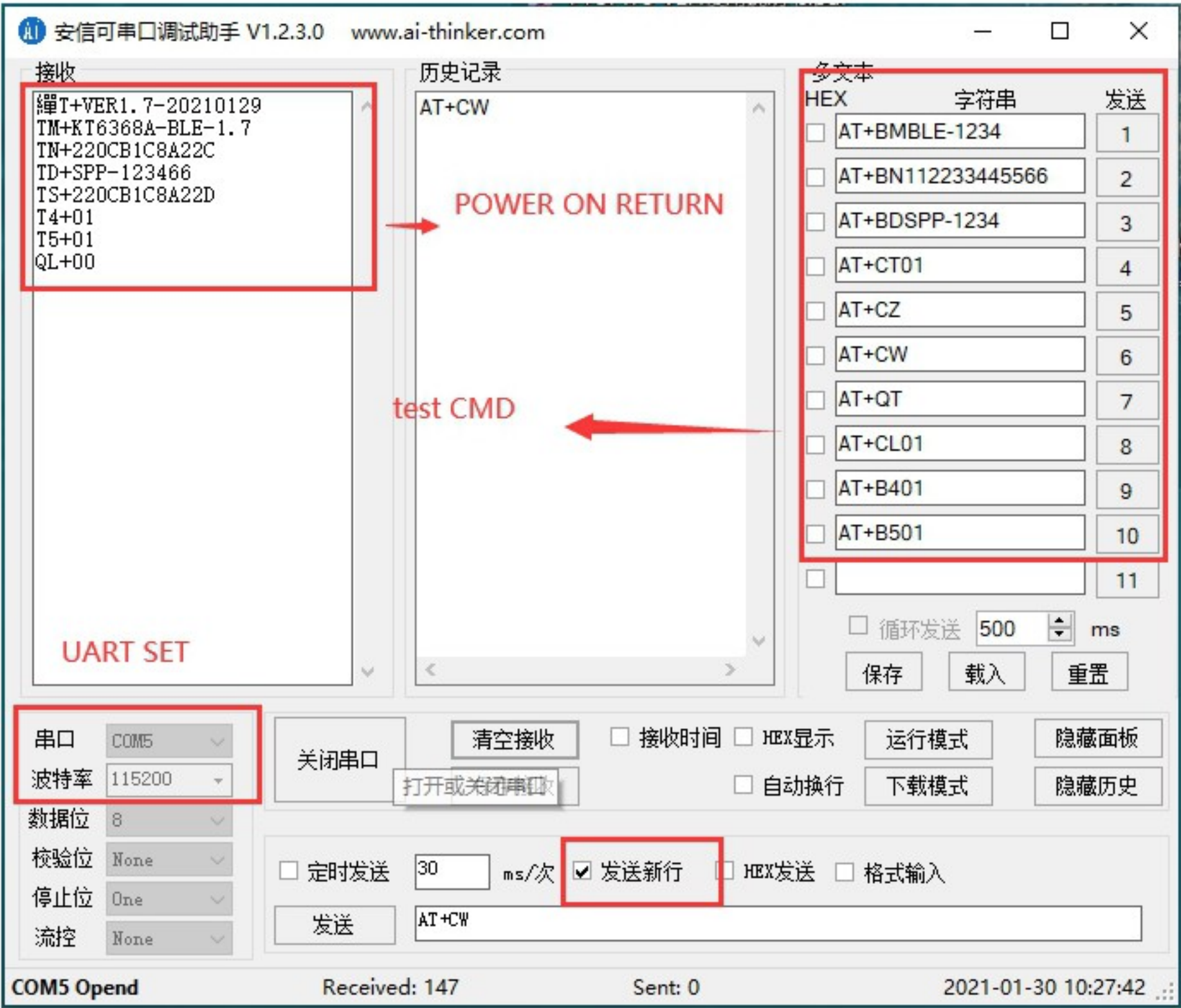
1、设置蓝牙名称之后，需要让芯片复位，发指令或者断电上电都可以，这样会显示新的蓝牙名称。我们默认的蓝牙名为“KT6368A-BLE”。设置的蓝牙名最长为“30”个字节，请不要超过这个范围
2、如果 AT 指令修改蓝牙名称之后，注意，你的手机端可能没有同步更新，还是显示之前的名称 (1)、因为你只修改了蓝牙的名称，蓝牙的 MAC 地址是没有变化的，所以手机端那边是不会更新名字 (2)、你要做的就是，换一台手机搜索试试，或者之前的手机删掉配对信息，重新在搜索 (3)、只要设置了蓝牙名，蓝牙名一定是更新过来了的，不用怀疑。芯片上电也会返回蓝牙名给您查看

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，用户可以不必关注，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

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 I/O 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 I/O 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 KT6368A 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	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

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
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+5\r\n	保留
ER+6\r\n	保留
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。

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

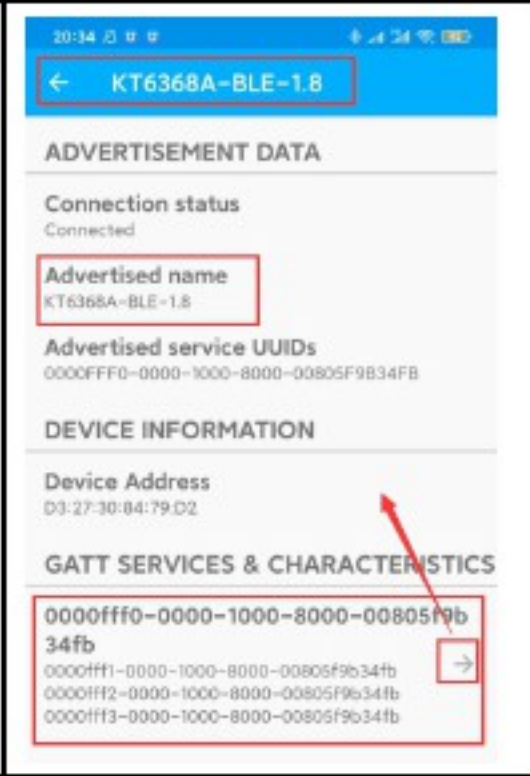
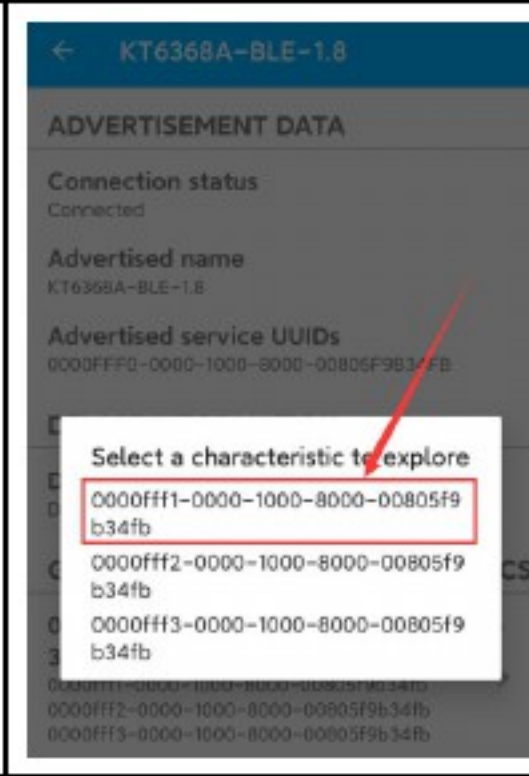
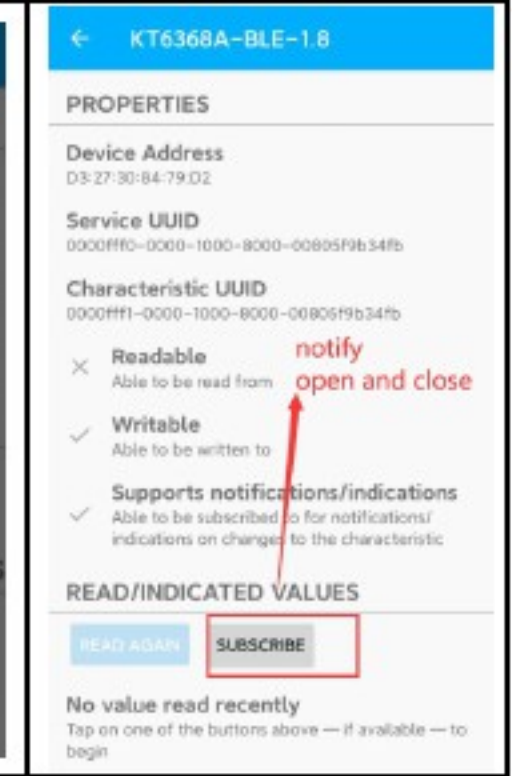
4.2 BLE 的 UUID 说明

1、主 UUID 是“FFF0”
2、特征 1 的 UUID 是“FFF1”，特征是“WRITE ” “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”
4、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“Lightblue”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

			
第一步： 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	第二步 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”	第三步： 1、找到需要的服务 2、也就是 FFF1 3、点击	第四步： 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

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，不推荐使用了，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。

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、但是请注意，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	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 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 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

问题 2	芯片是如何区分 AT 指令和透传的数据呢？												
答疑	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： <table><tr><th>MCU 端发送的数据</th><th>说明</th></tr><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>121212121212kkk</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></table> 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 2、对于手机端发送的数据，则更容易理解 —— SPP 和 BLE 透传说明 (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 蓝牙芯片收到之后，也是不会处理的，只会串口输出给 MCU	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	121212121212kkk	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												
121212121212kkk	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	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 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 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 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 5、目前 KT6368A 的程序，只做了从机版本，只能和手机连接 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 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 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 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

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	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 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 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 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 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 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 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

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，只在蓝牙未连接的状态有效。 2、只要蓝牙连接成功之后，就进入透传了，AT 指令无效了。 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， AT 指令读取 。Uuid 暂时不支持客户自己修改， 后期会加上

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	4、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 5、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 6、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 7、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

问题 10	支持买几个样品， 帮我修改波特率到 9600 吗？
回答	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 2、实在要修改， 收人工费 500。
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	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。 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 2、实在需要修改的， 可以， 收人工费 500 修改