

RC2657A

蓝牙5.0 (BLE) 串口透传模块

数据手册 V1.0



文档信息

型号	RC2657A	
名称	低功耗蓝牙透传模块	
文档类型	数据手册	
文档编号	RCBM-M01	
版本日期	V1.0.0	2021-5-24

版本更新

版本号	文档日期	更新内容
V1.0	2021/05/24	✓ 第一次发布;

注：本文档中的信息仅供智汉RF Crazy®的授权用户或许可人使用。没有智汉RF Crazy®的书面授权，请勿将本文档或其部分内容印制或作为电子文档副本传播。

目录

1. 概述.....	4
1.1. 特点.....	4
2. 工作模式示意图.....	6
3. 尺寸与引脚定义图.....	7
3.1. 引脚定义表.....	8
4. 串口透传协议说明.....	10
5. 串口 AT 指令.....	12
6. BLE 协议说明(APP 接口).....	23
6.1. 蓝牙数据通道【服务 UUID: 0xFFE5】.....	23
6.2. 串口数据通道【服务 UUID: 0xFFE0】.....	23
6.3. 防劫持密钥【服务 UUID: 0xFFC0】.....	23
6.4. 设备信息【服务 UUID: 0x180A】.....	25
6.5. 模块参数设置【服务 UUID: 0xFF90】.....	26
7. 广播数据设置.....	29
8. IOS APP 编程参考.....	31
9. 主机参考代码 (透传).....	30
附录 A: BLE 模块应用方案提示.....	35

1. 概述

智汉科技的RC2657A低功耗蓝牙模块，采用Maxscend MX2657作为核心处理器。模块运行在2.4 GHz ISM band, GFSK 调制方式(高斯频移键控), 40 频道2 MHz 的通道间隙, 3 个固定的广播通道, 37 个自适应自动跳频数据通道, 物理层可以和经典蓝牙RF组合成双模设备, 2 MHz 间隙能更好地防止相邻频道的干扰。

RC2657A的设计目的是迅速桥接电子产品和智能移动设备, 可广泛应用于有此需求的各种电子设备, 如仪器仪表、个人防护、智慧医疗、智能家居、智能穿戴、汽车电子、休闲玩具、定位跟踪等。用户可借此模块, 以最短的开发周期整合现有方案或产品, 以最快的速度占领市场, 同时为企业的发展注入崭新的技术力量。

模块可以工作在桥接模式(透明传输模式)。

模块启动广播后, 已打开特定 APP 的手机会对其进行扫描和对接, 成功之后便可以通过 BLE 协议对其进行监控。

桥接模式下, 用户 CPU 可以通过模块的通用串口和移动设备进行双向通讯, 用户也可以通过特定的串口 AT 指令, 对某些通讯参数进行管理控制。用户数据的具体含义由上层应用程序自行定义。移动设备可以通过 APP 对模块进行写操作, 写入的数据将通过串口发送给用户的 CPU。模块收到来自用户 CPU 串口的数据包后, 将自动转发给移动设备。此模式下的开发, 用户必须负责主 CPU 的代码设计, 以及智能移动设备端 APP 代码设计。

1.1. 特点

- ◆ 使用简单, 无需任何蓝牙协议栈应用经验;
- ◆ 用户接口使用通用串口设计, 全双工双向通讯, 最低波特率支持 4800bps;
- ◆ 默认 20ms 连接间隔, 连接快速, 并且 Android 与 IOS 的兼容性好;
- ◆ 支持 AT 指令软件复位模块;
- ◆ 获取 MAC 地址, 支持 AT 指令修改 MAC 地址(需要重新复位后生效);
- ◆ 支持 AT 指令调整蓝牙连接间隔, 控制不同的转发速率; (动态功耗调整)
- ◆ 支持 AT 指令调整发射功率, 修改广播间隔, 自定义设备识别码, 修改串口波特率, 修改模块名, 详情请查看 AT 指令表;
- ◆ 串口数据包长度, 可以是 500Byte 以下(含 500)的任意长度(大包自动分发);
- ◆ 高速透传转发, 最快可达 8.2K/S, 可稳定工作在 5K/S;
- ◆ 支持移动设备 APP 修改模块名称, 修改串口波特率, 产品识别码, 自定义广播内容, 广播周期;
- ◆ 支持移动设备 APP 对模块进行远程复位, 设置发射功率;

- ◆ 支持移动设备 APP 调节蓝牙连接间隔(动态功耗调整)；
- ◆ 支持防劫持密码设置、修改和恢复，防止第三方恶意连接。也可不使用。独立的密码操作结果通知，方便 APP 编程；
- ◆ 支持单脚位下地(长按)5S 恢复默认密码、下地(长按)20S 出厂恢复设置，APP 远程恢复出厂设置；
- ◆ 广播内容提示模块实时系统状态，包括 MAC 地址、连接间隙、广播周期、数据延迟时间、串口波特率、自定义设备识别码与防劫持密码使能等设置信息；
- ◆ 支持浅恢复和深度恢复模式，灵活恢复用户数据，而保留产品必须配置；
- ◆ 极低功耗的待机模式，RS02A1 芯片睡眠电流 1.7uA，模块实测功耗如下：

事件	平均电流 (打开 EN 内部上拉)	平均电流 (关闭 EN 内部上拉)	持续 时间	测试条件/备注
模块睡眠功耗	3.62uA	3.62uA	-	EN 拉高
广播	244.30uA	161.93uA	3.85ms	广播周期 200ms
广播	124.40uA	39.87uA	3.85ms	广播周期 1000ms
连接事件	360.33uA	266.84uA	2.25ms	连接周期 30ms
连接事件	162.52	86.22	2.25ms	连接周期 100ms

*注：使用 6 位半台式万用表测试方式：用直流 20mA 量程、0.006S 采样频率进行统计测量。如果希望得到更低功耗，可适当增大连接间隔或者广播周期，详见《[模块参数设置](#)》和《[串口 AT 指令](#)》相关章节。

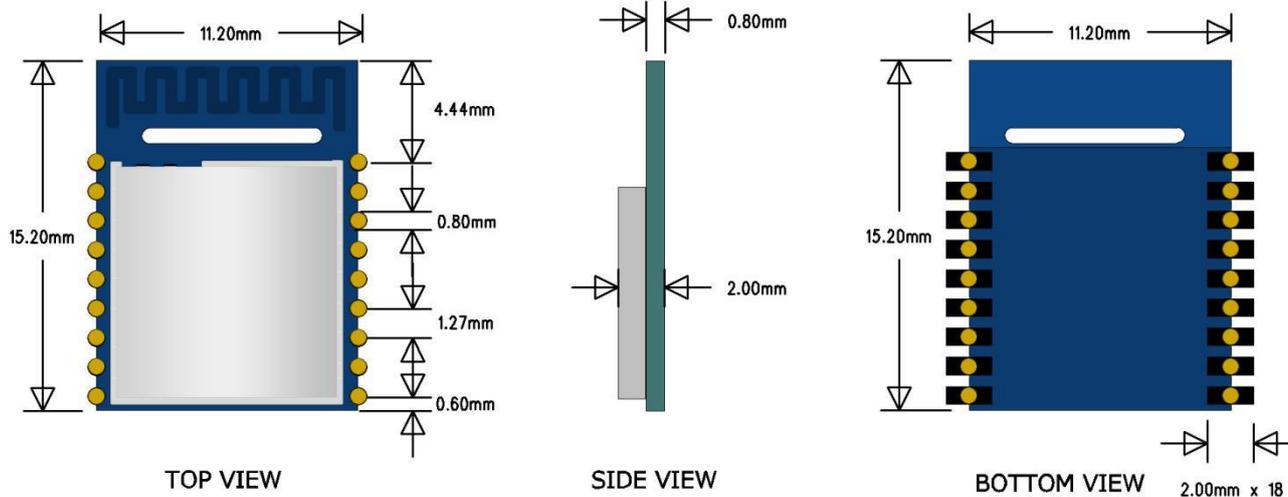
2. 工作模式示意图



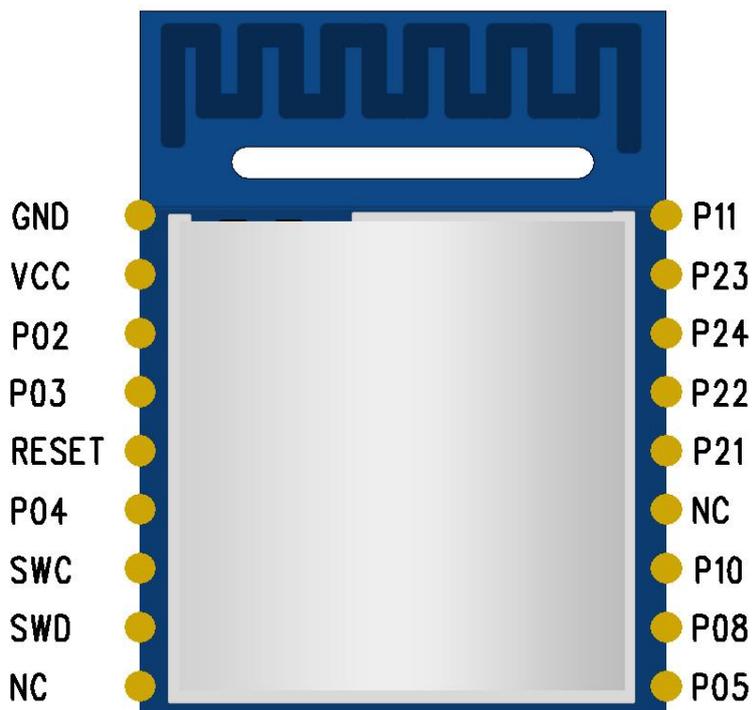
模块桥接模式和直驱模式示意图

注意：为避免用户 CPU 的IO 和模块 IO 的输出电平差异导致大电流，建议在模块的输出信号线 TX， BCTS 上串入一小额隔离电阻。

3. 尺寸与引脚定义图



模块尺寸



模块引脚定义图

3.1. 引脚定义表

脚位 序号	脚位 名称	芯片脚 位名称	输入/ 输出	说明
Pin1	GND	GND	-	模块地 GND
Pin2	VCC	VCC	-	模块电源 1.6 ~ 3.6V
Pin3	IO7	P02	O	输出口（可定时翻转）/睡眠状态指示
Pin4	IO6	P03	O	连接状态指示 0: 蓝牙已连接 1: 蓝牙未连接
Pin5	RST	RST	I	复位输入脚，低电平有效，无内部上拉
Pin6	EN	P04	-	模块使能控制线（低电平有效） 0: 模块开始广播，直到连接到移动设备 1: 无论模块当前状态，立即进入完全睡眠状态
Pin7	SWD	SWD	-	JTAG 数据脚
Pin8	SWC	SWC	-	JTAG 时钟脚
Pin9	NC	-	O	-
Pin10	RESTORE /IO3	P05	I	保持此引脚低电平 5s ，系统会恢复部分参数（浅恢复），若保持 20s 以上则将会恢复全部参数（深度恢复）（见《系统复位与恢复》章节）
Pin11	IO2	P08	O	-
Pin12	IO1	P10	O	-
Pin13	NC	-	O	-
Pin14	BRTS	P21	I	作为数据发送请求（用来唤醒模块） 0: 主机有数据发送，模块将等待接收来自主机的数据，此时模块不睡眠 1: 主机无数据发送，或主机数据发送完毕之后，应该将此信号线置 1

Pin15	BCTS	P22	O	数据输入信号（用来唤醒主机，可选） 0：模块有数据发送到主机，主机接收模块数据 1：模块无数据发送到主机，或模块数据发送完毕之后，会将此信号置 1
Pin16	TX	P24	O	模块串口发送端
Pin17	RX	P23	I	模块串口接收端
Pin18	ADC	-	O	-

4. 串口透传协议说明

模块的桥接模式是指，通过通用串口和用户 CPU 相连，建立用户 CPU 和移动设备之间的双向通讯。用户可以通过串口，使用指定的 AT 指令对串口波特率，BLE 连接间隔进行重置(详见后面《[串口 AT 指令](#)》章节)。针对不同的串口波特率以及 BLE 连接间隔，以及不同的发包间隔，模块将会有不同的数据吞吐能力。模块默认波特率为 115200bps。

模块 **BLE 连接间隔为 20 ms，串口波特率为 115200 bps 时**，模块具有最高理论转发能力 (8.2K/S)。这里就在电平使能模式下，这种配置为例，对透传协议做详细介绍。

模块可以根据获取到的已连接设备的 MTU 自定义串口包，模块会根据数据包大小自动分包发送，每个无线包最大载荷为 MTU 减去 3 个字节（例如：模块和安卓手机连接，获取 MTU 为 251，则模块发送数据给手机每个最大的包为 248 个字节）。移动设备方发往模块的数据包，必须自行分包（每包 1 个字节到 (MTU-3) 个字节之间）发送。模块收到无线包后，会依次转发到主机串口接收端。

- 1、串口硬件协议：115200bps，8，无校验位，1 停止位。
- 2、EN 为高电平，蓝牙模块处于完全睡眠状态。EN 置低时，模块会以默认 200ms 的间隔开始广播，直到和手机对接成功。当 EN 从低到高跳变，不论模块状态，会立即进入睡眠。
- 3、连接成功之后，主机（MCU）如有数据发送至 BLE 模块，需将 BRTS 拉低，主机可在约 100us 后开始发送数据。发送完毕之后主机应主动抬高 BRTS，让模块退出串口接收模式。要注意的是，抬高 BRTS 之前请确认串口数据完全发送完毕，否则会出现数据截尾现象。
- 4、当模块有数据上传请求时，模块会置低 BCTS，最快会在 500us 之后开始发送，直到数据发送完毕。数据发送完毕，模块会将 BCTS 置高。
- 5、如若主机的 BRTS 一直保持低电平，则蓝牙模块会一直处于串口接收模式，会有较高的功耗。
- 6、在模块连接成功后，会从 TX 给出 "**TTM:CONNECTED\r\n**" 字符串，可以根据此字符串来确定是否可以正常转发操作。也可以通过手机发送一个特定的确认字符串到模块，主机收到后即可确认已经连接。当连接被 APP 端主动断开后，会从 TX 给出 "**TTM:DISCONNECT\r\n**" 字符串提示。
- 7、模块的蓝牙**默认连接间隔为 20 ms**，如果需要节省功耗采用低速转发模式，需通过 AT

指令调整连接间隔（最长连接间隔 2000ms），每个连接间隔最多传输 248 个字节，连接间隔为 T(单位: ms)，那么每秒最高转发速率 **V (单位byte/s)** 为：

$$V = 248 * 1000 / T \quad (V \text{ 和 } T \text{ 有关})$$

如果模块的蓝牙连接间隔为 **20ms**，而每个间隔最多传输 **248 byte** 有效数据，因此理论最高传输能力(转发速率)为**248*1000/20= 12.4K Byte/s**。测试表明，转发速率在**5K/s** 以下，在信号较好时，基本上无漏包情况。**安全起见，无论是低速或者高速转发应用，都建议在上层做校验重传处理。**

注：Android 的MTU 为251 字节，IOS 的为185 字节，发送时每包大小为 MTU-3 个字节。

8、串口数据包的大小可以不定长，长度可以是 1k 字节以下的任意值，同样满足以上条件即可。但为最大效率地使用通讯的有效载荷，同时又避免通讯满负荷运行，推荐使用**20, 80, 248** 字节长度的串口数据包，包间间隔取大于 20ms。

注：经测试，在 IOS 中，调用对 Characteristic 的写函数使用 CBCharacteristicWriteWithResponse 参数，使用带回应写模式，这种模式会降低部分转发效率，但可保证单个数据包的正确性，而使用 CBCharacteristicWriteWithoutResponse 参数，使用不带回应写模式，这种模式会有利于提高转发效率，但数据包的正确性需要APP 上层去校验。

5. 串口 AT 指令

以"TTM"开头的字符串会当成 AT 指令进行解析并执行，**并从串口原样返回**，之后会追加输出执行结果，"TTM:OK\r\n\0"或"TTM:ERP\r\n\0"等。向串口 RX 输入的所有字符串均为 ASCII 码格式。**不以“TTM”开头的串口数据包，将被视为透传数据。**

➤ 连接间隔设定

向串口 RX 输入以下字符串，设定 BLE 连接间隔：

"TTM:CIT-Xms"

其中 X="20", "30", "50", "100", "200", "300", "400", "500", "1000", "1500", "2000", 单位 ms (以上数据格式都为 ASCII 码)。如"TTM:CIT-20ms"表示设定连接间隔为 20ms。在执行完此指令之后，会从串口 TX 得到以下确认：

"TTM:OK\r\n\0"表示更改成功，正以新的连接间隔在运行；

这个连接间隔设定的成功与否取决于移动设备对连接间隔的限制，不同的 IOS 版本最大连接间隔也有不同。使用 iPhone (IOS 8 及以上系统)中测试，最快支持 20ms，最慢支持 2s，另外，由于 BLE 协议内部机制，不同的连接间隔下此指令会有不同的执行效率。在 IOS 8 及以上系统中，从当前连接间隔为 2000ms 的情况下(最长 2000ms)，改变到其他连接间隔，可能最长需要等待约 100s 左右，而在其他高频度连接间隔（如：100ms）下执行此 AT 指令，会有很快的执行效率。

➤ 获取模块名称

向串口 RX 输入以下字符串：

"TTM:NAM-?"

会从 TX 收到：

" TTM:NAM-xxxxxxxxxxxx\r\n\0"

字符串后面"xxxxxxxxxxxx"为蓝牙模块名称。

➤ 模块重命名

向串口 RX 输入以下字符串，- 以后为模块名称，长度为 16 个字节以内，ASCII 码格式，

" TTM:REN-" + Name

如"TTM:REN-ABC123"表示将模块重命名为"ABC123"。

同样会从 TX 收到 "TTM:OK\r\n\0" 确认, 如果指令格式不对, 则会返回:

"TTM:ERP\r\n\0"

测试表明, 由于 IOS 版本关系, 设备名称修改在 IOS6 以上版本中可立即变更, 在 IOS5 中无法立即变更。

➤ **获取波特率**

向串口 RX 输入以下字符串, 设定波特率:

"TTM:BPS-?"

会从 TX 收到:

"TTM:BPS-X"

其中 X="4800", "9600", "19200", "38400", "57600", "**115200**", (以上数据格式都为 ASCII 码)。

➤ **波特率设定**

向串口 RX 输入以下字符串, 设定波特率:

"TTM:BPS-X"

其中 X="4800", "9600", "19200", "38400", "57600", "**115200**", "256000" (以上数据格式都为 ASCII 码)。如 "TTM:BPS-115200" 表示设定波特率为 115200bps。在执行完此指令之后, 会从串口 TX 得到以下确认:

之后会从 TX 收到 "TTM:BPS SET AFTER 2S..." 确认, 如果设置值不在选项中, 或者指令格式不对, 则返回:

"TTM:ERP\r\n\0"

➤ **获取物理地址MAC**

向串口 RX 输入以下字符串:

"TTM:MAC-?"

会从 TX 收到:

" TTM:MAC-xxxxxxxxxxx\r\n\0"

字符串后面 "xxxxxxxxxxx" 为 6 字节模块蓝牙地址。

➤ **设置模块MAC 地址**

向串口 RX 输入以下字符串：

"TTM:MAC-xxxxxxxxxxxx"

会从 TX 脚收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

设定掉电保存，重启模块后，模块将按照新的 MAC 地址进行工作。

➤ **获取模块版本号**

向串口 RX 输入以下字符串：

"TTM:VER-? "

会从 TX 脚收到 "TTM:VER-XXXXXX" ， X 为模块的版本号，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

➤ **获取模块连接密码**

向串口 RX 输入以下字符串：

"TTM:PWD-? "

会从 TX 脚收到 "TTM:PWD-XXXXXX" ， X 为 6 个字节的连接密码，如果指令格式不对，则会返回：

回：

"TTM:ERP\r\n\0"

➤ **设置模块连接密码**

向串口 RX 输入以下字符串，可以设置或者更改模块的连接密码，长度为 6 个字节的数字格式。

"TTM:PWD-xxxxxx"

会从 TX 脚收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

➤ **模块复位**

向串口 RX 输入以下字符串：

"TTM:RST-SYSTEMRESET"

会迫使模块软复位一次。

➤ **深度恢复**

向串口 RX 输入以下字符串：

"TTM:RST-RESET"

会迫使模块深度复位一次，恢复所有参数为出厂设定值。

➤ **恢复出厂密码**

向串口 RX 输入以下字符串：

"TTM:RST-RSTPWD"

会迫使模块浅复位一次，恢复密码参数为出厂默认值（清除密码）。

➤ **广播周期设定**

向串口 RX 输入以下字符串，设置模块的广播周期， $T = X * 100ms$

"TTM:ADP-(X)"

其中 X = "2", "5", "10", "15", "20", "25", "30", "40", "50" 之一（以上数据格式都为 ASCII 码）。如 "TTM:ADP-(2)" 表示设定广播周期为 200ms。会从 TX 脚收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

广播周期设定掉电保存，重启模块后，模块将按照新的广播周期进行广播。

➤ **附加自定义广播内容**

向串口 RX 输入以下字符串，自定义广播内容，

"TTM:ADD-" + Data

其中 Data 为准备附加的广播的数据，长度 $0 < L \leq 16$ ，以 ASCII 码格式输入。例如向串口 RX 输入 "TTM:ADD-Advertisement!"，会从 TX 脚收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

此指令设置后立即生效，可以通过此功能广播一些自定义内容。如果设置为 16 个全 0 数据，则认为不使用自定义广播数据，而是使用默认广播内容。

➤ **定义产品识别码**

向串口 RX 输入以下字符串，自定义产品识别码，

"TTM:PID-"+ Data

其中 Data 为两个字节的识别码，范围 0x0000~0xFFFF (L = 2)，每个字符以ASCII 码格式向串口 RX 输入。例如向串口 RX 输入 "TTM:PID-RS" ("RS 对应的十六进制为 0x5253")，会从 TX 脚收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

此识别码会出现在广播中，可以以此来过滤设备或判断是否是特定的产品。

➤ **发射功率设定**

向串口 RX 输入以下字符串，设置相应的发射功率，单位 dBm。

"TTM:TPL-(X)"

其中 X="5", "0", "-5", "-21" (以上数据格式都为 ASCII 码) *。如 "TTM:TPL-(0)" 表示设定发射功率为 0dBm。之后会从 TX 脚收到 "TTM:OK\r\n\0" 确认，并且模块立即使用新的发射功率进行通讯，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

➤ **RSSI 信号强度输出**

向串口 RX 输入以下字符串，设置开启 RSSI 信号强度定时打印，间隔时间 1 秒。

"TTM:RSI-ON"

向串口 RX 输入以下字符串，设置关闭 RSSI 信号强度定时打印。

"TTM:RSI-OFF"

设置成功后 TX 脚收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

如果已开启RSSI 打印功能，则每间隔 1 秒钟时间打印一次 RSSI 信号强度字符串，

"TTM:RSI-xx\r\n\0"(例如：RSSI 为-63dBm 则打印字符为"TTM:RSI-63\r\n\0") **注：**

此参数掉电不保存，并且连接断开后自动关闭 RSSI 输出。

➤ **EN 脚内部使能**

向串口 RX 输入以下字符串，设置 EN 脚开启内部上拉，默认状态。

"TTM:EUP-ON"

向串口RX 输入以下字符串，设置关闭 EN 脚开启内部上拉，EN 脚拉低使能广播时，能节省 40 多 uA 电流。

"TTM:EUP-OFF"

设置成功后 TX 脚收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

➤ **RTC 设定与获取**

向串口 RX 输入以下字符串，设置 RTC 时间，格式为年 4 位，月、日、时、分、秒各

2 位。

"TTM:RTC-xxxxxxxxxxxxxx"

比如设定2017年1月2日3时4分5秒则需输入的字符为

"TTM:RTC-20170102030405"

设置成功后 TX 脚收到 "TTM:OK\r\n\0" 确认, 如果指令格式不对, 则会返回:

"TTM:ERP\r\n\0"

向串口 RX 输入以下字符串, 设置获取当前系统时间。

"TTM:RTC-?"

设置成功后 TX 脚收到 "TTM:RTC-xxxxxxxxxxxxx\r\n\0", 格式与设定 RTC 格式相同; 如果指令格式不对, 则会返回: "TTM:ERP\r\n\0"

注: 此参数掉电不保存, 并且模块重新上电后 RTC 需要重新设定。

➤ 数据延时设定

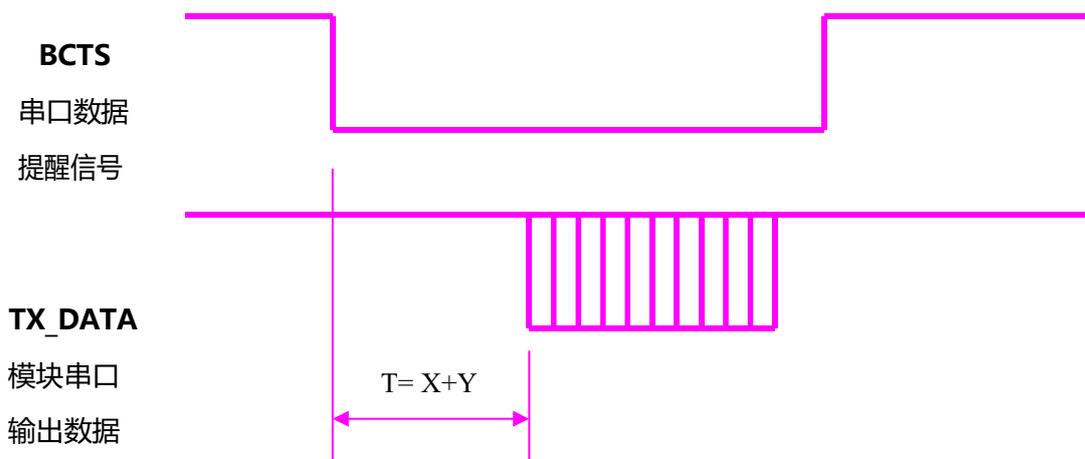
向串口 RX 输入以下字串，设置 BCTS 输出低到串口 TX 输出数据之间的延时，单位 ms。

"TTM:CDL-Xms"

其中 X="0","2","5","10","15","20","25" 之一（以上数据格式都为 ASCII 码）。如 "TTM:CDL-2ms" 表示设定延时为 2ms，如果指令无误，会从 TX 收到 "TTM:OK\r\n\0" 确认，如果指令格式不对，则会返回：

"TTM:ERP\r\n\0"

为了让用户 CPU 有足够的时间从睡眠中唤醒，到准备接收，模块提供了这个延时(X) 设定，在模块串口有数据发出之前会置低 BCTS，而 BCTS 输出低到模块 TX 输出数据之间的延时由此参数设定。可以保证最小延时不小于 X，实际延时会是 $T = (X + Y)$ ms，其中 $500\mu s < Y < 1ms$ 。此参数掉电保存。



模块串口输出数据延时设定示意图

➤ AT 指令表

AT 指令格式	掉电保存	参数说明	可能的回应	含义
"TTM:CIT-Xms" (连接成功后才有效)	否	X="20", "30", "50", "100", "200", "300", "400", "500", "1000", "1500", "2000" 设置相应的 BLE 连接间隔, 单位 ms	"TTM:TIMEOUT\r\n\0" "TTM:OK\r\n\0" "TTM:ERP\r\n\0"	设置超时 设置成功 错误参数
"TTM:NAM-?"	是	获取模块名称	TTM:NAM-xxxxxxxxxx xx, "xxxxxxxxxxxx"为 模块名称	返回模块名称
"TTM:REN- "+ Name	是	Name, 新模块名, 长度 为16 字节以内的任意字 串。	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	设置成功 错误参数
"TTM:BPS-?"	-	获取波特率	TTM:BPS-X, "X" 为波 特率	返回波特率
"TTM:BPS-X"	是	X="4800", "9600", "19200", "38400", "57600", "115200", "256000" 设置相应的波特率	"TTM:BPS SET AFTER 2S ...\r\n\0" "TTM:ERP\r\n\0"	设置成功, 会在 两秒后使用新的 波特率 错误参数
"TTM:MAC-?"	-	获取 MAC 地址	"TTM:MAC-xxxxxxxx xxx" xxxxxxxxxxxx 为 模块 MAC 地址	返回 MAC 地址
"TTM:MAC-X"	是	X 为 12 位 MAC 字符, 比 如 123456789ABC	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	设置成功 错误参数

TTM:VER-?	-	获取版本号	"TTM:VER-XXXXXXX"为模块版本号	返回版本号
"TTM:PWD-?"	-	获取连接密码	"TTM:PWD-xxxxxx"为模块的连接密码 "TTM:ERP\r\n\0"	返回模块的连接密码 (防劫持秘钥) 错误参数
"TTM:PWD-xxxxxx"	是	X 为 6 位纯数字, 比如 123456	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	设置成功 错误参数
"TTM:RST-SYSTEMRESET"	-	复位模块系统	无	复位模块
TTM:RST-RSTPWD	—	恢复出厂密码	Module is working!	恢复出厂密码
TTM:RST-RESET	—	深度恢复	Module is working!	恢复所有参数
TTM:ADP-(X)	是	X = "2", "5", "10", "15", "20", "25", "30", "40", "50" 设置相应的广播周期, T = X * 100ms	TTM:OK\r\n\0 TTM:ERP\r\n\0	设置广播周期, 如设置为 "5", 则为500ms
"TTM:ADD-" + Data	是	Data 为自定义广播数据, 数据长度 L <= 16;	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	设置成功 错误参数
"TTM:PID-" + Data	是	Data 为自定义产品识别码, 数据长度 L = 2, 默认为"RS";	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	设置成功 错误参数
"TTM:TPL-(X)"	否	X="5", "0", "-5", "-21" 设置相应的发射功率, 单位 dBm	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	设置成功 错误参数

TTM:RSI-ON	—	开启定时 1 秒获取RSSI 信号功能	TTM:OK\r\n\0 TTM:ERP\r\n\0	定时获取 RSSI 信号值
TTM:RSI-OFF	—	关闭获取 RSSI 信号功能	TTM:OK\r\n\0 TTM:ERP\r\n\0	关闭获取 RSSI 信号值
"TTM:EUP-ON"	是	开启 EN 脚上拉使能	TTM:OK\r\n\0 TTM:ERP\r\n\0	EN 脚内部上拉开启
"TTM:EUP-OFF"	是	关闭 EN 脚上拉使能	TTM:OK\r\n\0 TTM:ERP\r\n\0	EN 脚内部上拉关闭
TTM:RTC-?	—	获取 RTC 时间	TTM:RTC-xxxxxxxxxxx xxx, "xxxxxxxxxxxxxxxx" 为年、月、日、时、分秒	获取 RTC 时间
TTM:RTC-X	—	X 为年、月、日、时、分秒	TTM:OK\r\n\0 TTM:ERP\r\n\0	设置 RTC 时间
"TTM:CDL-Xms"	是	X="0","2","5","10","15","20","25" 设置 BCTS 输出低到串口输出数据之间的延时, 单位 ms	"TTM:OK\r\n\0" "TTM:ERP\r\n\0"	最小延时不于 X, 实际延时会是 X+Y ms , 500us<Y<1ms .

* 注：芯片发射功率硬件支持-21dBm~+5dBm，此版透传设置范围在-21dBm~5dBm。

6. BLE 协议说明(APP 接口)

6.1. 蓝牙数据通道【服务 UUID: 0xFFE5】

特征值 UUID	可执行的操作	字节数	默认值	备注
FFE9	Write	MTU-3	无	写入的数据将会从串口 TX 输出

说明：蓝牙输入转发到串口输出。APP 通过 BLE API 接口向此通道写操作后，数据将会从串口 TX 输出。详细操作规则见《串口透传协议说明(桥接模式)》章节。

6.2. 串口数据通道【服务 UUID: 0xFFE0】

特征值 UUID	可执行的操作	字节数	默认值	备注
FFE4	Notify	MTU-3	无	从串口 RX 输入的数据将会在此通道产生通知发给移动设备

说明：串口输入转发到蓝牙输出。如果打开了 FFE4 通道的通知使能开关，主 CPU 通过串口向模块 RX 发送的合法数据后，将会在此通道产生一个 notify 通知事件，APP 可以直接在回调函数中进行处理和使用。详细操作规则见《串口透传协议说明(桥接模式)》章节。

注：MTU 指最大传输单元 (Maximum Transmission Unit, MTU)，通信协议所能通过的最大数据包大小 (以字节为单位)。

6.3. 防劫持密钥【服务 UUID: 0xFFC0】

模块支持防劫持加密，此服务可以有效防止被非授权移动设备(手机)连接到此模块。模块的初始密码为 000000 (ASCII)，此情况下 APP 无需提交密码，视为不使用密码，任何安装指定 APP 的移动设备可以对其发起连接。

新密码(非全0)的设置和备份保存由 APP 完成，如果设置了新密码(非全0)，开始启用防劫持密码。在 APP 对此模块进行连接后，必须在蓝牙连接后的 20 秒内向模块提交一次曾经设置的连接密码，否则模块会断开连接。在 APP 提交正确密码到模块之前，无法对服务通道进行任何除提交密码之外的写

操作。

如果想恢复密码，需**拉低 RESTORE 脚位（见脚位定义表），并保持 5 秒**，模块密码会被恢复出厂设置。为了安全起见，模块不提供密码读操作，密码的记忆由 APP 来负责。

协议提供了密码通道来实现密码的提交、修改和取消密码服务。同样也提供了密码事件通知服务来通知 APP 对密码操作的结果，其中包括密码正确、密码错误、密码修改成功、取消使用密码四个事件。

特征值 UUID	可执行的 操作	字 节 数	举例	备注
FFC1 (handle: 0x0022)	write (掉电 保存)	12	"123456123456" (ASCII)	提交当前密码 123456 ，新密码和旧密码必须一致
			"123456888888" (ASCII)	把旧密码 123456 修改为新密码 888888 ，旧密码必须正确
			"888888000000" (ASCII)	取消密码，新密码修改为 000000 ，旧密码必须正确
FFC2 (handle: 0x0024)	notify	1	0 (PWD_RIGHT_EVENT)	提交密码正确
			1 (PWD_ERROR_EVENT)	提交密码错误
			2 (PWD_UPDATED_EVENT)	密码修改成功
			3 (PWD_CANCEL_EVENT)	取消密码

说明：

- 1、密码结构为 12 字节 ASCII 码，红色部分为当前密码，蓝色部分为新密码；
- 2、当前密码在被 APP 修改之前，默认为 "000000"；
- 3、通过打开通道 **FFC2** 的通知使能，将会在此通道产生有关密码操作的执行结果通知。
- 4、当 APP 提交密码 "**123456123456**"，新密码和当前密码相同，APP 会在 FFC2 通道得到通知 notify:0(PWD_RIGHT_EVENT)，表示提交密码正确；
- 5、当 APP 提交密码(红色部分)和当前密码不一致，如："123455xxxxxx"，x 部分不论是何值，APP 会在 FFC2 通道得到通知 notify:1(PWD_ERROR_EVENT)，表示密码提交错误；
- 6、当 APP 提交密码 "**123456888888**"，新密码为 "888888"，当前密码为 "123456"，APP 会在 FFC2 通道得到通知 notify:2(PWD_UPDATED_EVENT)，表示密码修改成功；
- 7、当 APP 提交密码 "**888888000000**"，新密码被修改为全 0，则表示取消使用密码，APP 会在 FFC2 通道得到通知 notify:3(PWD_CANCEL_EVENT)。

6.4. 设备信息【服务 UUID: 0x180A】

特征值 UUID	可执行的操作	字节数	默认值	备注
2A23	Read	8	0x0000000000000000 (Hex)	系统 ID
2A26	Read	17	Tv5.10u_xxxxxx_EP (ASCII)	模块软件版本号 xxxxxx 为软件生成日期
2A27	Read	9	RC2657A (ASCII)	模块硬件版本号
2A29	Read	8	RF Crazy (ASCII)	生产商名称

说明：模块信息读取通道。

2A26 为模块软件版本获取通道，可以通过对此通道进行读操作，来获取此模块软件版本信息。

格式如 Tv5.10u_xxxxxx_EP，其中 xxxxxx 部分为软件的生成日期，如 190628 表示 2019 年 6 月 28 日。

6.5. 模块参数设置【服务UUID: 0xFF90】

特征值 UUID	可执行的操 作	字 节 数	默认值	备注
FF91 (handle: 0x0028)	Read/Write	16	Tv510u-xxxxxxx x (带 结束符 的 ASCII 字符串)	设备名称, xxxxxxxx 为物理地址的后 四个字节
FF92 (handle: 0x002A)	Read/Write	1	3	蓝牙通讯连接间隔: 0: 20ms 1: 30ms 2: 50ms 3: 100ms 4: 200ms 5: 300ms 6: 400ms 7: 500ms 8: 1000ms 9: 1500ms A: 2000ms
FF93 (handle: 0x002C)	Read/Write	1	5	设定串口波特率: 0: 4800 bps 1: 9600 bps 2: 19200 bps 3: 38400 bps 4: 57600 bps 5: 115200 bps 6: 256000bps
FF94 (handle: 0x002E)	Write	1	无	远程复位恢复控制通道: 远程复位控制, 写入 0x55 对模块进 行复位 远程浅恢复控制, 写入 0x35 对模块 进行浅恢复 (仅仅恢复用户设定密 码), 并复位 远程深度恢复控制, 写入 0x36 对模块进行深度恢复 (让模块所有参数回到出厂设 置), 并复位

FF95 (handle: 0x0030)	Read/Write	1	3	设定广播周期: 0: 200 ms, 1: 500 ms, 2: 1000 ms, 3: 1500 ms, 4: 2000 ms, 5: 2500 ms, 6: 3000 ms, 7: 4000 ms, 8: 5000 ms
FF96 (handle: 0x0032)	Read/Write	2	0x5253	设定产品识别码
FF97 (handle: 0x0034)	Read/Write	1	1	设定发射功率: 0: 5dBm 1: 0 dBm 2: -5dBm 3: -21dBm
FF98 (handle: 0x0036)	Read/Write	16	默认广播内容 (详见《广播数据设置》章节)	设定自定义广播数据 自定义广播数据, $0 < n \leq 16$

说明：模块信息配置通道。

FF91 为设备名称设置通道。可以通过对此通道进行读写操作，来获取和设定模块名称。设置的名称长度 L，必须满足 $0 < L < 17$ ，**建议以结束符结尾（‘\0’）**。默认为“Tv5vvv-xxxxxxx\0”（16 byte），vvv 为固件版本号，xxxxxxx 为MAC 地址后四个字节。

FF92 为模块连接间隔设置通道。可以通过对此通道进行写操作，来设定移动设备和模块之间的连接间隔，借此可以灵活控制设备功耗，以及数据吞吐量。测试表明，使用iPhone(IOS 8 及以上系统)从连接间隔为 500ms 修改为其他连接间隔，需要大约 30s 的等待时间。相反从高频度的连接间隔(如 30ms)进行变更，会有很高的执行效率(BLE 协议导致)。

FF93 为模块串口波特率设置通道。可以通过对此通道进行读写操作，来设定模块通用串口波特率，两秒后开始启用新的波特率。出厂设置默认为 5(115200 bps)。

FF94 为远程复位恢复控制通道，通过写入不同值，可以实现不同的控制功能。

- 1、对此通道写入 **0x55**，**对模块进行软件复位。**
- 2、对此通道写入 **0x35**，**对模块进行浅恢复，所有用户密码将恢复到出厂设置控制，之后会复位模块。**
- 3、对此通道写入 **0x36**，**对模块进行深度恢复，所有系统参数将恢复到出厂设置控制，之后会复位模块。**

FF95 为模块广播周期设置通道。可以通过对此通道进行读写操作，来设定模块广播周期。出厂设置默认为 3(200ms)。

FF96 为模块产品识别码设置通道。可以通过对此通道进行读写操作，来设定模块识别码，APP 端可以通过此识别码来进行过滤和连接指定的产品类型。出厂设置默认为 0x5253。

FF97 为模块发射功率设置通道。可以通过对此通道进行写操作，来设定模块发射功率。出厂设置默认为 1(0 dBm)。

FF98 为模块广播内容设置通道。可以通过对此通道进行写操作，来自定义模块的广播数据。当数据为全 0(16 byte)时，认为不使用自定义广播数据，而使用默认的广播数据，详见《广播数据设置》章节。

7. 广播数据设置

默认广播数据：当模块的 EN 脚被置低后，模块将会进行间隔为默认 200ms 的广播，在广播数据中的 GAP_ADTYPE_MANUFACTURER_SPECIFIC (IOS 编程中官方定义宏)

域中包含了以下内容，默认广播内容为 16 个字节：

```
{
    0x52,0x53,          自定义设备类型编码，默认为“RS”，可由AT指令及APP进行设
定;
    0x19,0x16,          模块固件生成日期，默认为0x19,0x16，为19年16周生成;
    0x00,0x00,0x00,0x00,0x00,0x00,          模块MAC地址;
    0x05,              模块BPS参数，默认为5，为115200bps;
    0x05,              模块CTS参数，默认为10，为CTS拉低5ms后发送数据;
    0x00,              模块广播间隙参数，默认为200mS;
    0x01,              模块发射功能参数，默认为0dBm;
    0x00,              模块连接间隙参数，默认为20ms;
    0x00,              模块防劫持密码超时使能，默认0表示未开启。
}
```

广播中的数据为首次编译后的初始设定值，并不会通过 AT 指令或 APP 设定新参数后进行改变。

自定义广播数据：如果使用 AT 指令自定义了广播内容，最大长度为 16 字节(蓝色部分)，在广播数据中的 GAP_ADTYPE_MANUFACTURER_SPECIFIC 域中将包含了以下内容，长度为 2+n 个字节：

```
{
    0x00,0x00,          自定义设备类型编码，默认为00 00,可由AT指令进行设定;
    Data [n],          自定义广播数据，n <= 16;
}
```

注：自定义广播数据可通过 AT 指令修改，并且掉电保存。重新上电后，将会使用最后自定义的广播数

据。如果自定义广播数据为全 0 (16 byte), 则认为不使用自定义广播, 而使用系统默认的广播内容。为避免广播数据过长带来多余的功耗, 也可以通过设置自定义广播数据为 1 字节的任意值。

8. IOS APP 编程参考

模块总是以从模式进行广播，等待智能移动设备做为主设备进行扫描，以及连接。这个扫描以及连接通常是由APP 来完成，由于 BLE 协议的特殊性，在系统设置中的扫描蓝牙连接没有现实意义。智能设备必须负责对BLE 从设备的连接，通讯，断开等管理事宜，而这一切通常是在 APP 中实现。

有关BLE 在IOS 下的编程，最关键的就是对**特征值(Characteristic, 本文叫通道)的读, 写, 以及开启通知开关**。通过对通道的读写即可实现对模块直驱功能的直接控制，无需额外的 CPU。典型函数说明摘抄如下：

```

/!*
 * @method writeValue:forCharacteristic:withResponse:
 * @param data The value to write.
 * @param characteristic The characteristic on which to perform the write operation.
 * @param type The type of write to be executed.
 * @discussion Write the value of a characteristic.
 * The passed data is copied and can be disposed of after the call finishes.
 * The relevant delegate callback will then be invoked with the status of the request.
 * @see peripheral:didWriteValueForCharacteristic:error:
 */
- (void)writeValue:(NSData *)data forCharacteristic:(CBCharacteristic *)characteristic type:(C
BCharacteristicWriteType)type;

```

说明：对某个特征值进行写操作。

```

NSData *d = [[NSData alloc] initWithBytes:&data length:mdata.length]; [p
writeValue:d
forCharacteristic:c
type:CBCharacteristicWriteWithoutResponse];

```

```

/!*
 * @method readValueForCharacteristic:
 * @param characteristic The characteristic for which the value needs to be read.
 * @discussion Fetch the value of a characteristic.
 * The relevant delegate callback will then be invoked with the status of the request.
 * @see peripheral:didUpdateValueForCharacteristic:error:
 */
- (void)readValueForCharacteristic:(CBCharacteristic *)characteristic;

```

说明：读取某个特征值。

```
[p readValueForCharacteristic:c];
```

```
/*!
 * @method setNotifyValue:forCharacteristic:
 * @param notifyValue The value to set the client configuration descriptor.
 * @param characteristic The characteristic containing the client configuration.
 * @discussion Ask to start/stop receiving notifications for a characteristic.
 * The relevant delegate callback will then be invoked with the status of the request.
 * @see peripheral:didUpdateNotificationStateForCharacteristic:error:
 */
```

```
- (void)setNotifyValue:(BOOL)notifyValue forCharacteristic:(CBCharacteristic *)characteristic;
```

说明：打开特征值通知使能开关。

```
[self setNotifyValue:YES forCharacteristic:c]; //打开通知使能开关
[self setNotifyValue:NO forCharacteristic:c]; //关闭通知使能开关
```

```
/*!
 * @method didUpdateValueForCharacteristic
 * @param peripheral Peripheral that got updated
 * @param characteristic Characteristic that got updated
 * @error error Error message if something went wrong
 * @discussion didUpdateValueForCharacteristic is called when CoreBluetooth has updated a
 * characteristic for a peripheral. All reads and notifications come here to be processed.
 *
 */
```

```
- (void)peripheral:(CBPeripheral *)peripheral didUpdateValueForCharacteristic:(CBCharacteristic *)characteristic error:(NSError *)error
```

说明：每次执行完读取操作后，会执行到这个回调函数。应用层在此函数内保存读取到的数据。

有关设备的扫描，连接以及其他通讯细节，可以参考智汉科技提供的基于 IOS 的透传模块测试 APP 源码。里面实现了对 FFE9 和 FFE4 转发蓝牙数据到串口，转发串口数据到蓝牙两个通道（特征值）的操作（通知和写操作），都是通过对某个通道（特征值）的读写来实现。只是通道 UUID 以及读写字节数不同。（相关源码请向业务索取）

9. 主机参考代码 (透传)

逻辑关系：模块间是用 BCTS, BRTS 两个 IO 口进行发送接收的通知和控制。

这两个 IO 常态高位，置低触发，如果模块有数据要发，置低 BCTS 通知单片机接收，如果单片机有数据要发，置低 BRTS 通知模块接收。示意性代码如下：

```
void main(void)
{
    EN = 0; //使能EN, 开始广播

    while(!BLEMoudleAck("TTM:OK\r\n0")); //等待手机端扫描, 连接
    //等待连接成功, 也可加入限时等待
    //也可判断连接提示信号线的电平

    BRTS = 0; //BRTS 置低通知 RC2657AX 模块准备接收
    halMcuWaitMs(2); //延迟 2ms
    UARTWrite( HAL_UART_PORT_0, "TTM:CIT-100ms", 14);
    //修改连接间隔, 从串口得到确认:

    halMcuWaitMs(5); //延迟 5ms,确保数据已经发出
    BRTS = 1; //RTS 置高, 发送完毕
    while(!BLEMoudleAck("TTM:OK\r\n0")); //等待设置成功, 也可加入限时等待

    while(1){ //循环收发测试
        while(1){
            if(BCTS == 0){ //检测, 若BCTS 置低则准备接
                收while(BCTS==0); //等待发送完毕, 也可限时等待
                if(UARTRead(uartBuffer) == SUCCESS) //串口读取数据
                {... ...} //使用数据
            }
            BRTS = 0; //RTS 置低通知 RC2657AX 模块准备接收
            halMcuWaitMs(2); //延迟 2ms
            send_TX("1234567890"); //发送任意数据 (200byte 以内)
            halMcuWaitMs(5); //延迟 5ms,确保数据已经发出
            BRTS = 1; //RTS 置高, 发送完毕
            halMcuWaitMs(20); //延迟再发下一个包, 延时视包大小而定
        }
    }
}
```

```
}  
}  
}
```

推荐操作条件

功能操作在以下表格中各条件参数值的极限之外不能保证其性能，长期在这个极限之外操作或多或少会影响模块的可靠性。

注意：

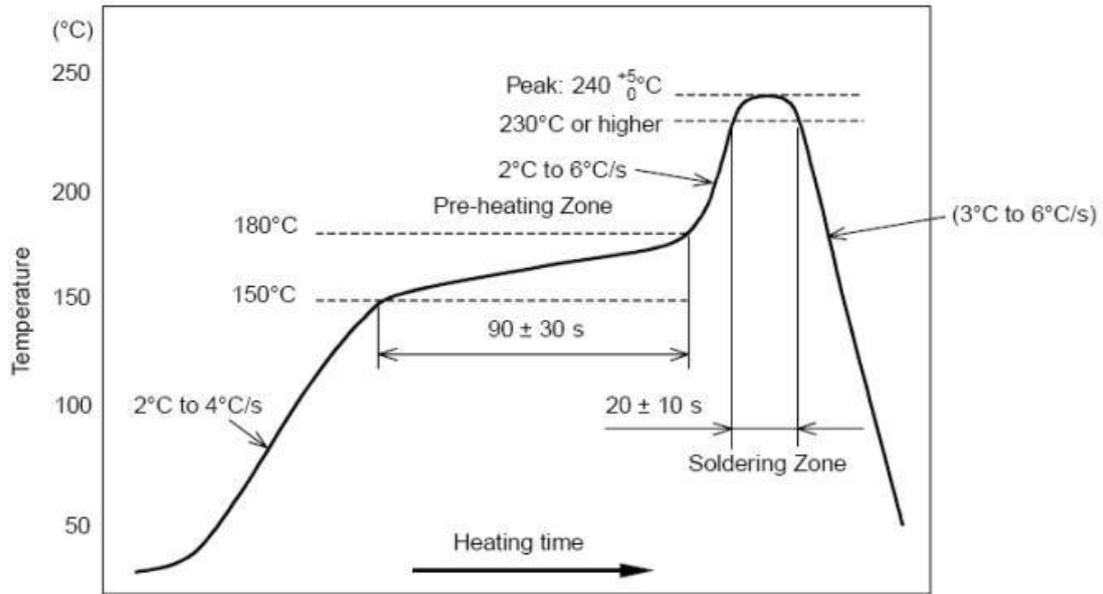
操作温度受晶体频率的变化限制；

为了确保无线射频性能，电源上纹波必须小于 300mV。

标识	条件	最小值	典型值	最大值	单位
电源与 IO	电池模式	1.6	3.3	3.6	V
操作温度	/	-40	25	85	°C
环境热摆		-20		20	°C/分钟

回流焊条件

- 1、加热方法：常规对流或 IR 对流；
- 2、允许回流焊次数：2 次，基于以下回流焊（条件）（见下图）；
- 3、温度曲线：回流焊应按照下列温度曲线（见下图）；
- 4、最高温度：245°C。



部件的焊接耐热性温度曲线（焊接点）图

静电放电警示



模块会因静电释放而被损坏，RF CRAZY 建议所有模块应在以下 3 个预防措施下处理：

- 1、必须遵循防静电措施，不可以裸手拿模块。
- 2、模块必须放置在能够预防静电的放置区。
- 3、在产品设计时应该考虑高电压输入或者高频输入处的防静电电路。

静电可能导致的结果为细微的性能下降到整个设备的故障。由于非常小的参数变化都可能导致设备不符合其认证要求的值限，从而模块会更容易受到损害。

附录 A: BLE 模块应用方案提示

计数采集(计步器, 弹跳球, 心率计), 86 盒插座改造, 遥控开关, 调光照明, 环境渲染背景光, 医疗检测〔血压, 血氧, 体温〕, 互动遥控玩具〔开关量, 模拟量, 输入, 输出〕, 机器人, 直升飞机, 玩具车, 防丢寻物, 电量采集, 充电管理, 外置gps, 温湿度计, 蓝牙手表, 飞镖机, 保龄球等娱乐设备新接口, (智能设备)遥控接口, 报警器, 门禁考勤(蓝牙锁), 巡逻寻根器, 反控(智能设备)应用(紧急拨号, 遥控拍摄), 蓝牙打印, 空调控制器, 机顶盒控制器, 物流统计管理, 胎压检测, 汽车自动锁, 遥控按摩器, 车位记录, 户外点阵广告, 运动计量〔跑步, 自行车, 高尔夫〕, 定时开关, 宠物监管, 婴儿儿童护理(实时体温检测, 防丢失), 运动健身玩具(手机配合), 距离感应触发应用, 调速应用, 智能家居(遥控类), 仪器仪表无线接口, 设备无线配置接口, 景点定位, 区域软禁控制, 定量计时, 可穿戴设备, 蓝牙读卡器, 便携仪表, 设备固件远程升级接口

* 部分可以利用模块透传功能进行开发, 部分直接使用直驱功能即可完成设计。