
BLE4.0透传模块使用手册

Bluetooth Low Energy UART module
Specification



目录

引言.....	2
● 产品概述.....	2
● 产品特点.....	2
● 功耗测试.....	3
● 模块工作示意图.....	4
● 模块尺寸及脚位定义.....	5
● 串口透传协议说明.....	7
➤ 串口相关指令.....	8
➤ 测试指令.....	8
➤ 修改模块名称.....	9
➤ 修改蓝牙模块波特率.....	9
➤ 修改广播间隔.....	9
➤ 修改连接间隔.....	9
➤ 查询模块MAC地址.....	10
➤ 重启模块.....	10
➤ 恢复出厂设置.....	10
➤ 查询固件版本号.....	10
● BLE协议说明（APP接口）.....	11
➤ 蓝牙数据通道【服务UUID:0xFFFD】.....	11
➤ 串口数据通道【服务UUID:0xFFFD】.....	11
➤ 模块参数设置通道【服务UUID:0xFFFD】.....	12
➤ 电池电量报告【服务UUID:0x180F】.....	12
➤ 设备信息【服务UUID:0x180A】.....	13
● APP测试BLE通信.....	14
参考代码.....	16
参考原理图.....	17
版本信息.....	17

引言

感谢您使用**蓝牙低功耗透传模块**，为了更好更快更有效的使用本模块，请在使用前认真仔细地阅读本说明书。我们将提供完整硬件、软件参考方案，缩短产品研发周期，为您节省成本投入。

● 产品概述

模块启动后会自动进行广播，已打开特定 APP 的手机会对其进行扫描和对接，成功连接后便可以通过 BLE 协议对其进行操作。

用户 CPU 可以通过模块的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 MCU。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。用户必须负责主 MCU 的代码设计，以及智能移动设备端 APP 代码设计。

● 产品特点

主要特点：

1. 使用简单，无需任何蓝牙协议栈应用经验；
2. 用户接口使用通用串口设计，全双工双向通讯；
3. 默认 20ms 连接间隔，连接快速；
4. 待机休眠功耗 0.4uA；
5. 支持 IO 口指示蓝牙连接状态，可接 LED 指示灯；
6. 支持 AT 指令修改串口波特率，软件复位模块，获取 MAC 地址，修改模块名；
7. 支持 AT 指令调整蓝牙广播间隔，控制不同连接速度。（动态功耗调整）
8. 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）；
9. 串口数据包长度，可以是 300byte 以下(含 300)的任意长度。（大包自动分发，波特率 115200）
10. 高速透传转发，最快可达 4K/S，可稳定工作在 2.5K-2.8K；
11. 支持移动设备 APP 修改模块名称，掉电保存；
12. 支持移动设备 APP 修改串口波特率，掉电保存；
13. 支持移动设备 APP 对模块进行远程复位；
14. 支持模块电量提示，电量读取，可自动通知。（设备电量提醒）；

● 功耗测试

测试选用 CC2540A1 模块作为测试对象，供电电压为 3.2V，VCC 脚串接一个万用表，测量模块不同状态时的电流参数。由于 BLE 模块工作状态下的电流为波动电流，本例测试只能作为参考值，官方测试需要在模块电源端串接 10R 电阻，用示波器测得压降波形，在进行积分计算。

工作状态	测试电流值	测试条件	备注
待机模式	0.3~0.5uA	EN 脚悬空	深度休眠，需要 IO 口唤醒
广播模式	160uA~790uA	EN 脚拉低，广播周期为 250ms	未连接
连接模式	560uA	EN 脚拉低，连接周期为 100ms	已连接
BLE 传输状态	980uA	EN 拉低，APP 端间隔 1s 给数据通道 FFF6 写数据	已连接
串口唤醒模式	8.25mA	EN 拉低，UART_WAKE 拉低	已连接

● 模块工作示意图

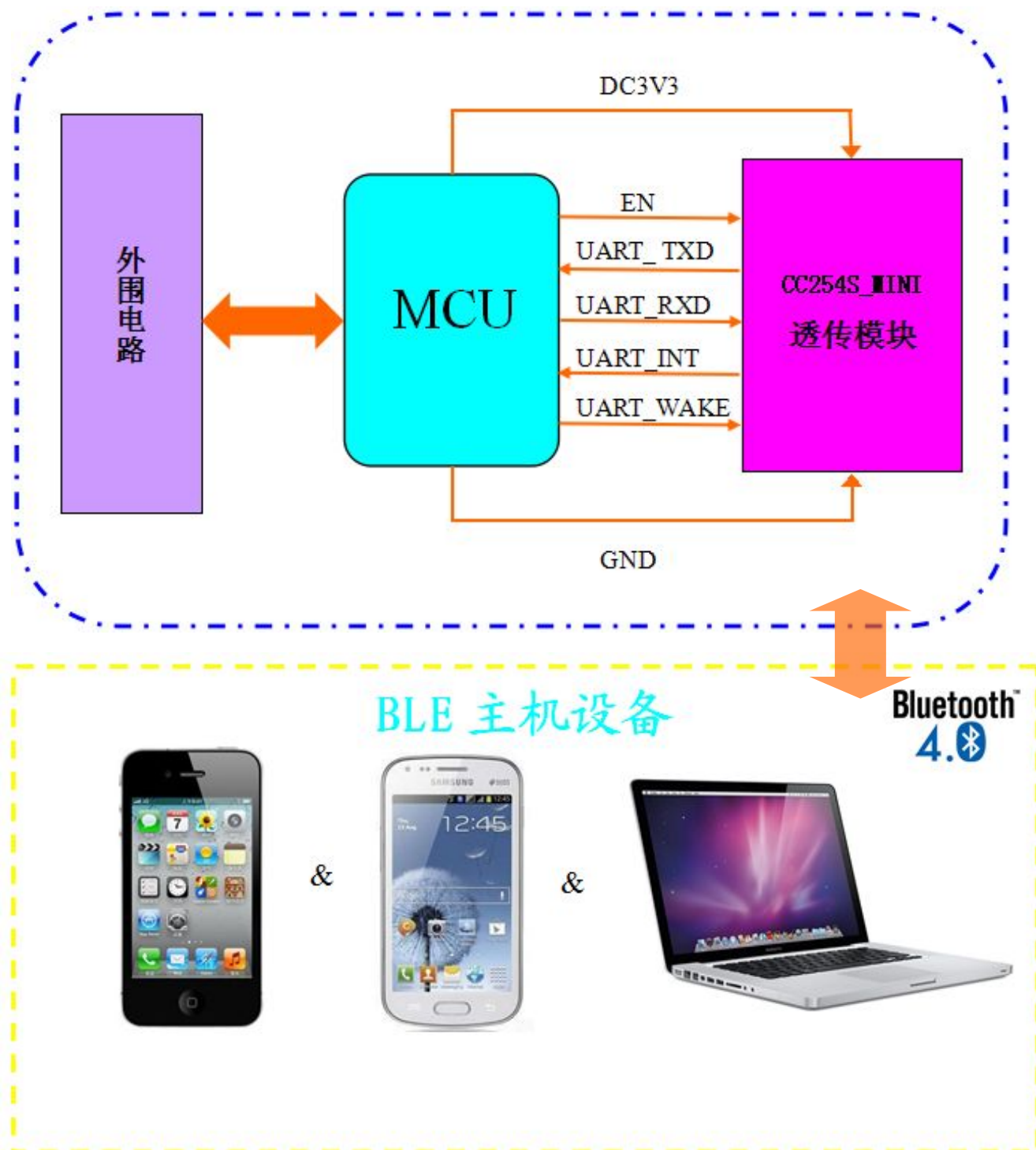


图 1：模块工作示意图

● 模块尺寸及脚位定义

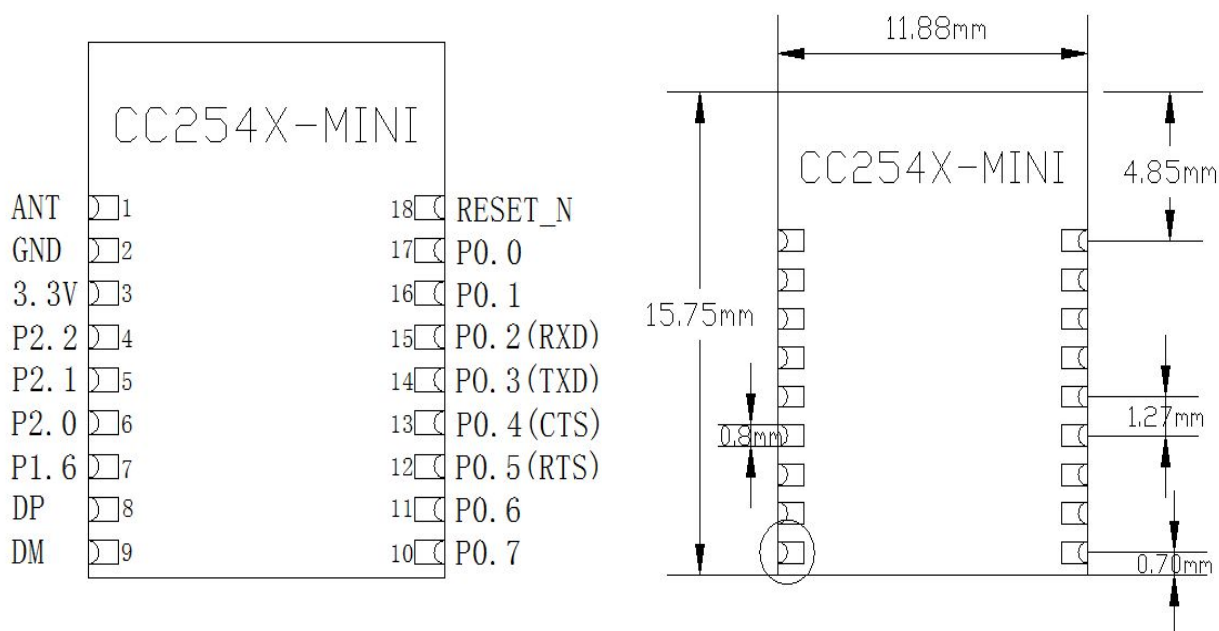


图2: CC254X-MINI 尺寸接口图

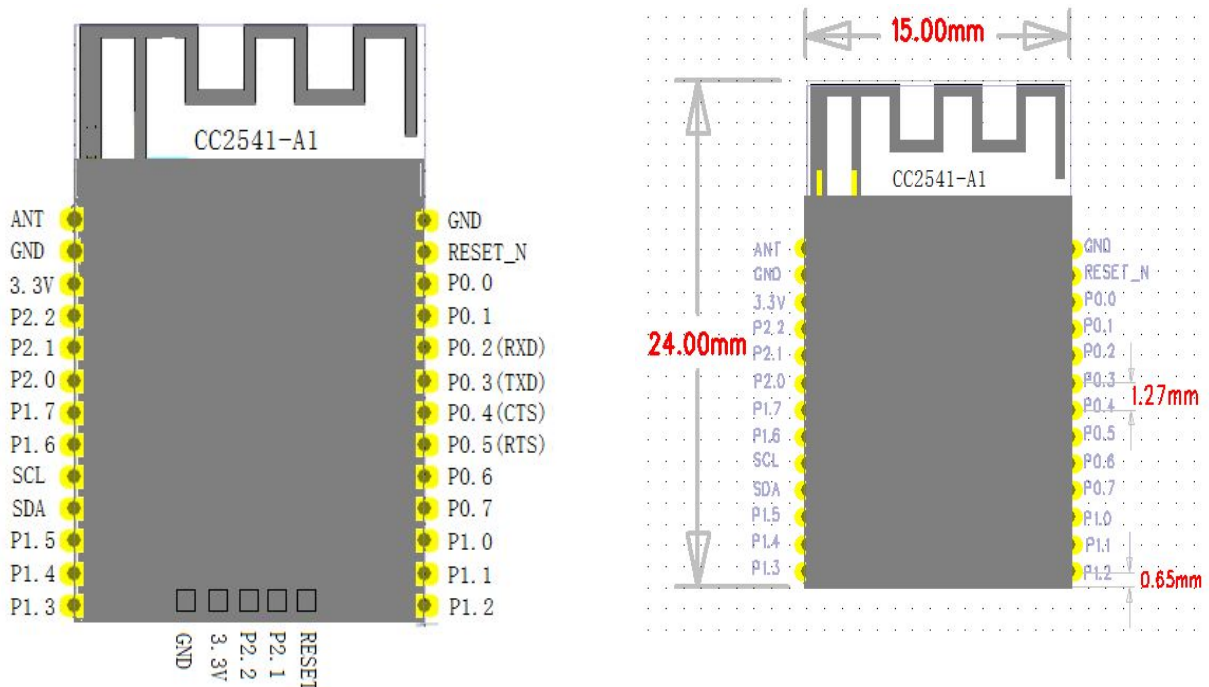


图3: CC254X 尺寸接口图

功能脚	模块脚位名称	芯片脚位名称	输入/输出	功能说明
VCC	3.3V	VCC	—	模块电源（3V-3.6V）
GND	GND	GND	—	模块地 GND
EN	P2.0	P2.0	输入	模块使能线，低有效带内部上拉 0：模块开始工作，进入广播状直到连接到移动设备 1：无论模块当前状态，立即进入睡眠状态（0.4uA）
UART_TXD	P0.3 (TXD)	P0.3	输出	模块串口发送脚
UART_RXD	P0.2 (RXD)	P0.2	输入	模块串口接收脚
UART_WAKE	P0.4 (CTS)	P0.4	输入	串口唤醒，数据发送请求 0：唤醒模块串口，开始向模块 RX 脚发送数据 1：主机无数据发送，将此信号线至 1，让模块进入休眠
UART_INT	P0.5 (RTS)	P0.5	输出	串口中断信号，用来唤醒主机 0：模块 TX 脚有数据，主机启动接收 1：模块 TX 脚无数据，主机关闭接收
STATUS	P1.6	P1.6	输出	模块状态指示脚，带上拉，可接 LED 1：模块未连接 0：模块已连接
DEBUG_DC	P2.2	P2.2	—	调试线，可不接
DEBUG_DD	P2.1	P2.1	—	调试线，可不接
RESET	RST_N	RST	—	模块复位脚，低有效
DP/SCL	DP/SCL	USB_P/SCL	—	USB/I2C 接口，未使用
DM/SDA	DM/SDA	USB_M/SDA	—	USB/I2C 接口，未使用

● 串口透传协议说明

所谓的透明传输，不管传的是什么，所采用的设备只是起一个通道作用，把要传输的内容完好的传到对方！透明传输不用关心下层协议的传输，比如你要寄信，只需要写地址交给邮局就行了，然后对方就能收到你的信，但是中途经过多少车站，火车，邮递员，你根本不知道，所以对于你来说邮递的过程是透明的。

由于是透明传输，用户需要自己定义数据格式，做好上层数据校验处理。

不同串口波特率和 BLE 连接间隔，以及不同的发送间隔，模块会有不同的数据吞吐能力，大数据量应用建议选用高速波特率 115200bps 以上。

模块可以从串口接收最大 300bytes 数据包，每个无线包最大 20bytes（BLE 协议决定最大包 20 字节），超过 300bytes 用户需自行分包处理。移动设备端 APP 必须自行分包，每包 20 字节，模块收到 APP 数据，会依次转发到串口。

模块在 BLE 通讯过程中一般为从机(peripheral)，需要主机（移动设备）发起连接，需要用户自主编写 APP，完成扫描、连接、数据收发。

① 串口参数配置，默认波特率为 115200bps，数据位 8bytes，无校验位，1 停止位。

② 模块蓝牙名称默认为 SerialCom，客户可通过 AT 指令修改，并可掉电保存。

③ EN 为高电平时，模块处于完全睡眠状态，此时功耗最低 0.4uA。置低 EN 脚后，模块切换到工作状态，并以默认 100ms 间隔开始广播，直到连接到移动端设备。EN 脚从低到高跳变，无论模块当前状态，立即进入睡眠状态。

④ 发射功率默认 0dBm。

⑤ 打开 APP 的手机与模块连接成功后，主机（MCU）如有数据要发送到串口，需先将模块 UART_WAKE 拉低，使模块进入活动状态，MCU 可在延时 100us 后发送数据。数据发送完毕后，主动将 UART_WAKE 脚拉高，使模块退出串口接收模式。

⑥ 当模块有数据上传请求时，模块 UART_INT 脚会从高电平变为低电平，可以用来唤醒 MCU，MCU 可以通过检测 UART_INT 脚电平变化判断是否有数据接收。模块 UART_TXD 数据发送完成后，UART_INT 脚自动变为高电平。

⑦ MCU 检测到有数据时，应尽快进入接收状态，如果没有及时接收，模块会一直等待接收，无法进入低功耗模式。

⑧ 若 UART_WAKE 脚一直保持低电平，会有较大功耗。

⑨ 模块只支持透传模式，不带任何校验格式，用户可自行定义封包格式。

注：每次读写串口数据，都需要将 CTS 脚拉低。如客户对功耗要求不高，可将 CTS 脚一直拉低，测试串口透传只需四根线（TX，RX，VCC，GND）。

➤ 串口相关指令

指令列表	格式	有效回应	相关参数	掉电保存	远控支持
测试指令	AT+TEST	OK\r\n	-	-	是
修改/查询模块名称	AT+NAME=para	OK+NAME:para\r\n	Para:模块蓝牙名称, 不超过 15 字节	是	是
	AT+NAME?			-	是
修改/查询波特率	AT+BPS=para	OK+BPS:para\r\n	Para:2400~230400 默认 115200	是	是
	AT+BPS?			-	是
修改/查询广播间隔	AT+ADVI=para	OK+ADVI:para\r\n	Para:00020~10240 默认间隔 100ms	是	否
	AT+ADVI?				
修改/查询连接间隔	AT+CONI=para	OK+CONI:para\r\n	Para:0010~4000 默认间隔 20ms	是	否
	AT+CONI?				
查询 MAC 地址	AT+MAC?	OK+MAC:para\r\n	Para 为 6 字节 16 进制数	-	是
重启模块	AT+RST	OK+RST\r\n	-	-	是
恢复出厂设置	AT+RENEW	OK+RENEW\r\n	-	-	是
查询固件版本号	AT+VER	OK+VER:para\r\n	Para 为固件版本	-	是

➤ 测试指令

向模块串口 RX 输入以下字符串

“AT+TEST”

模块串口 TX 回应

“OK” 表示所用波特率正常，串口可以正常通信。

注：该指令支持 APP 通道，可快速验证串口通信及 BLE 通信。

➤ 修改模块名称

向模块串口 RX 输入以下字符串

“AT+NAME=radioland”表示新蓝牙名称为 radioland，蓝牙名称不得超过 15 个字节模块串口 TX 回应。

“OK+NAME:radioland”蓝牙名称修改成功，如果蓝牙处于连接状态，蓝牙名称不会变，断开连接后重新扫描即可看到新蓝牙名称；广播状态时重新扫描可看到新蓝牙名称。

测试表明，由于 IOS 版本关系，设备名称修改在 IOS6 中可立即变更，在 IOS5 中无法立即变更。注：此名称掉电保存。

➤ 修改蓝牙模块波特率

向模块串口 RX 输入以下字符串

“AT+BPS=19200”19200 为新波特率，默认波特率为 115200。

模块串口 TX 回应

“OK+BPS:19200”模块波特率已经改为 19200，使用新波特率 19200 才能串口通信。

注：此名称掉电保存。由于波特率影响串口指令的使用，用户可以通过 PC 进行设置波特率，也可以通过移动设备的 BLE APP 接口进行设置。见《模块参数设置【服务 UUID：0xFF5】》。

➤ 修改广播间隔

向模块串口 RX 输入以下字符串

“AT+ADVI=00020”表示将模块广播间隔调整为 20ms，广播间隔调整范围 00020~10240，即 20ms 到 10.24s 之间，默认广播间隔为 100ms。

模块串口 TX 回应

“OK+ADVI:00020”重启模块后，模块 20ms 的广播间隔发出广播。

注：广播间隔越小，主机查找从机时间越短。广播间隔越大，模块广播状态下越省电。IOS 系统建议最大广播间隔为 1285ms，所以如果模块是用来和 IOS 设备连接，广播时间间隔尽量不要超过 1285ms。

➤ 修改连接间隔

向模块串口 RX 输入以下字符串

“AT+CONI=0100”表示将模块连接间隔调整为 100ms，连接间隔调整范围 0010~4000，即 10ms 到 4 之间，默认连接间隔为 20ms。

模块串口 TX 回应

“OK+CONI=0100”设置完成需重新启动模块，模块和主机以 100ms 的连接间隔进行连接。

注：连接间隔越小，数据交换速度越快。连接间隔越大，连接状态下功耗越大。苹果 IOS 通信时，连接间隔必须大于 20ms。

➤ 查询模块 MAC 地址

向模块串口 RX 输入以下字符串

“AT+MAC?”

模块串口回应

“4F 4B 2B 4D 41 43 3A B4 99 4C 71 5C 77 ” 由于模块 MAC 地址为六位十六进制码，串口调试助手要切换到十六进制显示，后面六位 “ B4 99 4C 71 5C 77” 即为 MAC 地址，ASCII 码显示状态下 “OK+MAC: ” 后为 MAC 地址，此时为乱码状态为正常现象。

➤ 重启模块

向模块串口 RX 输入以下字符串

“AT+RST”

模块串口 TX 回应

“OK+RST” 等待 50ms 模块断开连接自动重启，此时需要重新扫描连接。

➤ 恢复出厂设置

向模块串口 RX 输入以下字符串

“AT+RENEW”

模块串口 TX 回应

“OK+RENEW” 模块恢复出厂设置后，50ms 后自动重启，此时需要重新调整参数后再重新扫描连接。

注：若串口通信异常，无法通过串口恢复出厂设置，可以通过移动设备的 BLE APP 接口进行设置。见《模块参数设置【服务 UUID: 0xFFFD】》

➤ 查询固件版本号

向模块串口 RX 输入以下字符串

“AT+VER”

模块串口 TX 回应

“OK+VER:RL2541-V1.7” 表示模块当前固件版本为 RL2541-V1.7。

● BLE协议说明（APP接口）

➤ 蓝牙数据通道【服务 UUID:0xFFFD】

特征值 UUID	可执行操作	字节数	默认值	备注
FFF6 (handle:0x0023)	Read/Write	20	12345	写入数据会从串口 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》章节。

➤ 串口数据通道【服务 UUID:0xFFFD】

特征值 UUID	可执行操作	字节数	默认值	备注
FFF7 (handle:0x0026)	Read/Notification	20	12345	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 FFF7 通道的通知使能开关（如果使用 BTool 操作，需向 0x0026+1=0x0027 写入 0100），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》章节。

➤ **模块参数设置通道【服务 UUID:0xFFFD】**

特征值 UUID	可执行操作	字节数	默认值	备注
FFF5 (handle:0x001F)	Write/Notification	20	12345	写入指令，产生通知返回到移动设备

说明：向通道写相关 AT 指令可以修改模块参数，如果打开了 FFF5 通道的通知使能开关（如果使用 BTool 操作，需向 0x001F+1= 0x0020 写入 0100），每次写入命令，将会在此通道产生一个 notify 通知事件（回应信息），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》章节。

➤ **电池电量报告【服务 UUID:0x180F】**

特征值 UUID	可执行操作	字节数	默认值	备注
2A19 (handle:0x002B)	Read/Notification	1	供电电量的百分比	读取当前电量百分比或自动产生通知

说明：电池电量读取或通知通道。APP 通过 BLE API 接口向 2A19 通道读操作，来获取当前模块的供电电量的百分比。如果打开了此通道的通知使能（如果使用 BTool 操作，需向 0x002B+1=0x002C 写入 0100），每读取到一次电量后，将会在此通道产生一个 notify 通知事件，附带了电量百分比，最大值：100%(3V)，最小值：0%(2V)，APP 可以直接在回调函数中进行处理和使用。

➤ 设备信息【服务 UUID:0x180A】

特征值 UUID	可执行操作	字节数	默认值	备注
2A23 (handle:0x0030)	Read	8	xxxxxx0000xxxxxx (Hex)	系 统 ID , xxxxxxxxxxxxxx (低字节在前) 为模块芯片物理地址
2A26 (handle:0x0036)	Read	11	RL2541-V1.7	模块固件版本号

说明：模块信息读取通道。

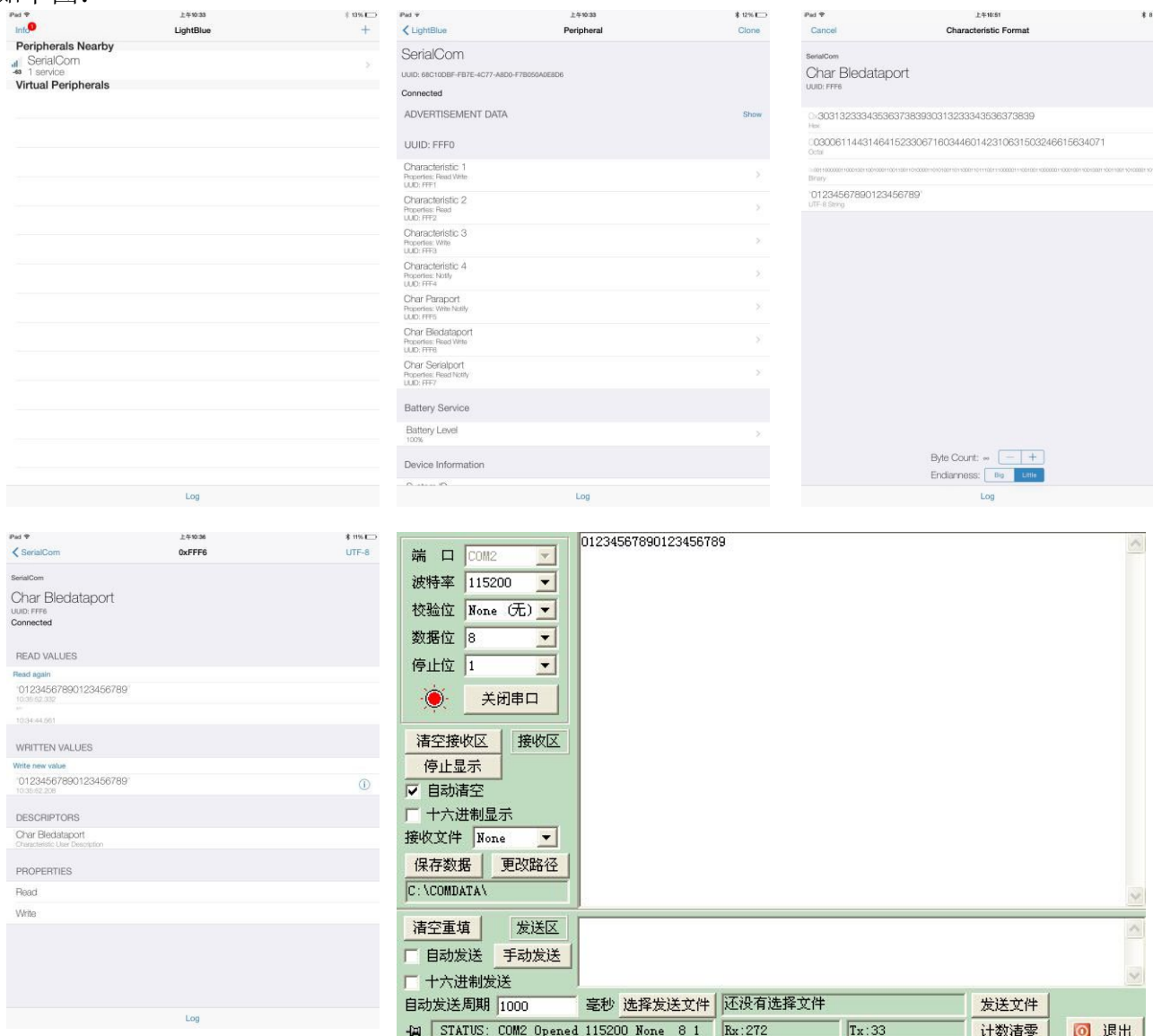
2A23 为模块信息获取通道，可以通过对此通道进行读操作，来获取此模块 ID。格式 xxxxxx0000xxxxxx 其中 xx 部分为模块芯片的物理地址 MAC，六个字节，低字节在前。

2A26 为模块软件版本号读取通道，可以通过对此通道进行读操作，来获取模块软件版本，格式为 RL2541-VX.x 为版本号。

● APP测试BLE通信

TI 官方测试 APP LightBlue 测试 BLE 通信。LightBlue 可以在 AppStore 下载到。打开 iPhone4S/ipad 中的 AppStore，搜索“LightBlue”，下载安装测试。

打开 LightBlue 后，扫描设备，模块默认名称 SerialCom，点击连接设备，APP 会自动查找相应 BLE 服务，打开服务 **蓝牙数据通道 (UUID:FFF6)**，将 APP 数据格式改为 UTF-8，向通道内写入字符串“01234567890123456789”，最大 20 字节。如果此时模块串口已经连接了 MCU 或者串口终端，即可进行串口双向通信，本测试蓝牙模块和 PC 上串行终端连接，用于发送接收串行数据。如下图所示：



串口终端有数据发送到手机 APP，必须保持 UART_WAKE 为低电平，防止模块进入休眠。APP 端需要开启 notification 功能，LightBlue 开启开关为 Listen for notifications。下图所示：



APP 编程时，根据低功耗蓝牙协议，移动设备发送数据可以通过蓝牙数据通道(发送)的对应服务（UUID）进行写操作。模块数据到移动设备的数据传送，是通过通知的形式进行，因此在 APP 启动后需要打开串口数据通道(接收)对应服务(UUID)的通知(Notification)使能，之后模块串口收到的数据包会自动发送到移动设备。

● 参考代码

说明：模块需要控制的脚有 EN，UART_WAKE 脚，这两个控制较常态高位（自带上拉），置低触发。如果模块有数据要发，模块会自动置低 UART_INT 脚，如果单片机有数据要发，置低 UART_WAKE 通知模块接收。STATUS 可以作为连接状态指示，也可用串口回应信息作为状态判断，未连接 STATUS 为高电平，已连接 STATUS 为低电平。示意性代码如下：

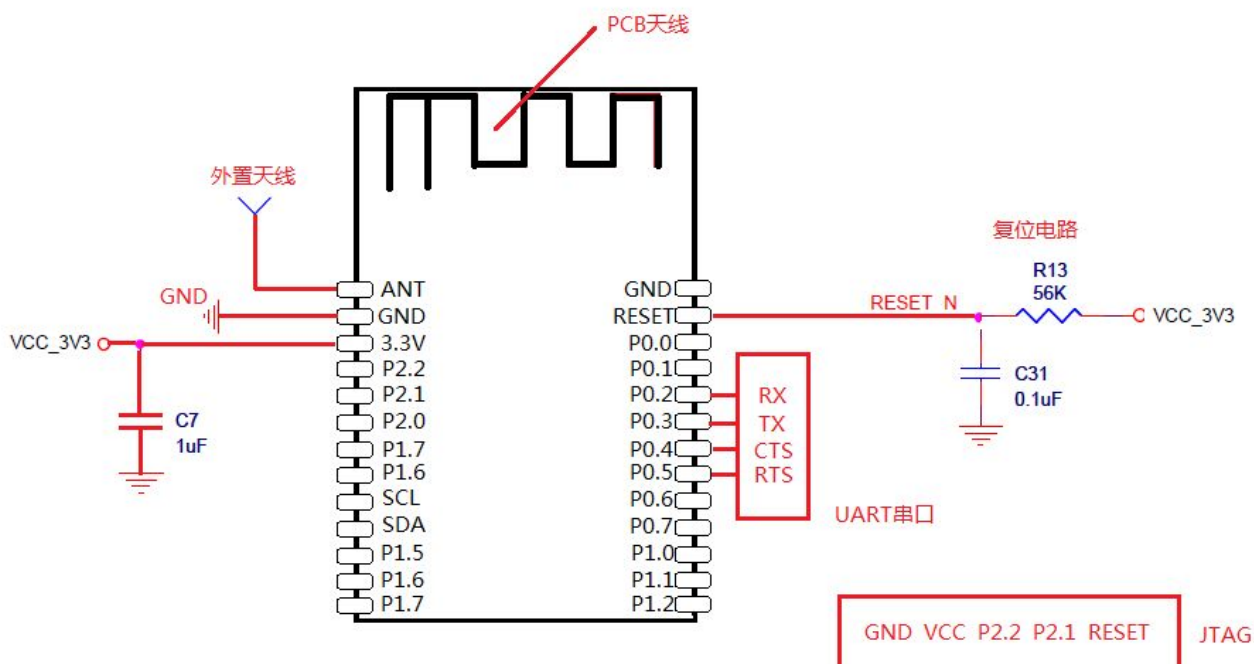
模块数据发送：

```
void main(void)
{
    EN=0; //模块进入工作模式
    while(!BLEModuleAck("Connected=OK")); //等待手机端扫描
    UART_WAKE=0; //唤醒模块，串口进入接收状态
    UARTWrite("radioland",10); //发送字符串
    Delay(50); //发送时间间隔 50ms
    UARTWrite("radioland",10); //发送的数据
    UART_WAKE=1; //数据发送完成，进入休眠
    EN=1; //退出工作模式
}
```

模块数据上传：

```
void main(void)
{
    EN=0; //模块进入工作模式
    while(!BLEModuleAck("Connected=OK\r\n")); //等待手机端扫描
    while(1) //也可采用中断方式，RTS 管脚电平变化触发中断
    {
        if(UART_INT==0) //判断是否有数据需要上传当有数据上传时 RTS 自动置低。
        {
            UARTRead( HAL_UART_PORT_0, buff, 10); //读取数据
        }
    }
}
```

● 参考原理图



● PCB layout注意事项

BLE 工作在2.4GHz频率下，应尽量避免各种因素对无线收发的影响，注意以下几点：

1. 包围模块的产品外壳部分 避免使用金属，如果外壳是金属的，应考虑使用外置天线。
2. 产品内部金属螺钉等应远离模块的射频部分。
3. 模块应放置于主板的四周，天线部分靠边或角，模块天线下方的主板区域不允许铺铜或走线。

● 版本信息

版本号	日期	修正内容
V1.2	2014-5-17	原始版本
V1.3	2014-5-30	增加 EN 脚
V1.7	2014-6-20	增加 AT 指令