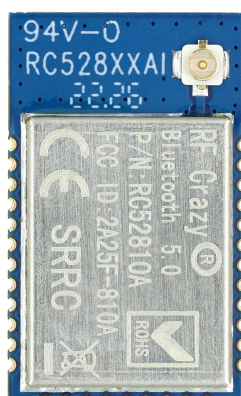


# RC52810AI

## 蓝牙 5.3 (BLE) 透传模块

### 说明手册



#### 摘要

此技术规格文档主要介绍 RC52810AI Bluetooth®低功耗蓝牙串口通信说明及使用规范。如需自主开发固件，您可以使用 SDK 集成开发环境（IDE）将自己的应用嵌入到集成的低功耗蓝牙协议栈之上。

#### 文档信息

型号	RC52810AI	备注
名称	低功耗蓝牙透传模块	
文档类型	数据手册	
文档编号	RCBM-N04	
版本日期	V1.0	2022-07-05

## 版本更新

版本号	文档日期	更新内容
V1.0	2022/07/05	✓ 第一次发布

注：本文档将会不定期优化更新，在使用此文档前，请确保为最新版本。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. 产品概述 .....	5
➢ 主要特点 .....	5
➢ 设备模式说明 .....	6
➢ 设备状态 .....	7
2. 封装尺寸及脚位定义 .....	7
3. 串口透传协议说明 .....	10
4. BLE 协议说明(APP 接口) .....	10
➢ Service UUID .....	10
➢ BLE 数据接收 UUID .....	10
➢ BLE 数据发送 UUID .....	10
➢ AT 指令操作 UUID .....	11
5. 串口 AT 指令 .....	12
AT 命令表 .....	12
➢ 进入 AT 指令模式 .....	13
➢ 退出 AT 指令模式 .....	13
➢ 设备名称 .....	13
➢ MAC 地址 .....	14
➢ 串口回显 .....	14
➢ 显示设备状态 .....	15
➢ 设备角色 .....	15
➢ 从角色广播参数 .....	16
➢ 连接间隔 .....	17
➢ 从角色 Service .....	18
➢ 断开连接 .....	18
➢ Beacon 参数 .....	19
➢ 发射功率 .....	19
➢ 休眠模式 .....	20
➢ 串口波特率 .....	21
➢ 用户鉴权 .....	21
➢ 设备重启 .....	22
➢ 恢复出厂设置 .....	22
➢ 固件版本 .....	23
6. 用 APP 测试透传功能 .....	24
7. PC 端测试工具 .....	27

➤ 使用前准备 .....	27
➤ 连接步骤说明 .....	27
8. IOS APP 编程参考 .....	30
9. 主机（MCU）参考代码（透传） .....	32
联系我们 .....	33
附录 A：BLE 模块应用方案提示 .....	34
附录 B：BLE 模块硬件规格说明 .....	35

## 1. 产品概述

智汉科技 RF Crazy® RC52810AI 是基于 Nordic 的 nRF52810 SoC 设计开发的高性能、高灵敏、超低功耗的蓝牙 5.3 (BLE) 模块。它集成了 ARM Cortex-M4 内核, 2.4GHz BLE 收发器和 IPEX 天线座, 并提供了一个完整的射频解决方案, 无需额外的蓝牙射频设计, 帮助用户加快产品上市时间。

RC52810AI 的应用目标是 2.4GHz 蓝牙低能耗系统、人机交互设备(键盘、鼠标和远程控制)、手机配件、运动和医疗设备、Beacon 标签、工业数据监测控制设备。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。

RC52810AI 工作在**桥接模式**(透传传输模式)。桥接模式下, 用户 MCU 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 MCU。模块收到来自用户 MCU 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 MCU 的代码设计, 以及智能移动设备端 APP 代码设计。

**为了保证程序的稳定性, 模块上电 500ms 之后才开始正常工作。**

### ➤ 主要特点

- 1、使用简单, 无需任何蓝牙协议栈应用经验;
- 2、同时支持 BLE 从角色模式和 Beacon 模式;
- 3、默认 20 ms 连接间隔, 连接快速, 并且 Android 与 IOS 的兼容性好;
- 4、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 4800 bps, 最高支持 460800bps;
- 5、支持 AT 指令软件复位模块;
- 6、获取 MAC 地址, 支持 AT 指令修改 MAC 地址(要重新复位后生效);
- 7、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);
- 8、支持 AT 指令调整发射功率、修改广播间隔、修改串口波特率、修改模块名, 详情请查看 AT 指令表;
- 9、支持 AT 指令修改 Service UUID;
- 10、可通过 APP 发送 AT 指令;
- 11、高速透传转发, 30 KB/s 稳定传输;
- 12、极低工作功耗, 模块实测功耗如下:

事件	平均电流	测试条件/备注
模块睡眠功耗	1.23 $\mu$ A	-
广播	70.33 $\mu$ A	广播周期 200 ms
广播	28.38 $\mu$ A	广播周期 500 ms
广播	14.02 $\mu$ A	广播周期 1000 ms
广播	7.64 $\mu$ A	广播周期 2000 ms
广播	3.49 $\mu$ A	广播周期 5000 ms
连接事件	64.72 $\mu$ A	连接周期 50 ms
连接事件	33.8 $\mu$ A	连接周期 100 ms

## ➤ 设备模式说明

设备一共有以下 2 种工作模式：

- 1、BLE 从角色模式（Slave）；
- 2、Beacon 模式。

默认启动角色为从角色模式，可通过 AT 指令“AT+ROLE”切换角色调整工作模式，详细查看 AT 指令说明，**Beacon** 模式下串口处于关闭状态，只能通过手机 APP 发送指令切换模式。

## ➤ Slave 模式默认配置

- 1、设备名称：RF-CRAZY；
- 2、广播间隔：200 ms；
- 3、连接间隔：20-75 ms；
- 4、UUID 默认为 128 位；
- 5、广播为可连接模式；
- 6、设备为透传状态。

## ➤ Beacon 模式默认配置

- 1、Company ID: 0x0059 (Nordic)；
- 2、Major UUID: 0x0102；
- 3、Minor UUID: 0x0304；
- 4、RSSI: -50 dBm；
- 5、UUID: 0x01, 0x02, 0x03, 0x04,0x05, 0x06, 0x07, 0x08,0x09, 0x0A, 0x0B, 0x0C,0x0D, 0x0E, 0x0F, 0x10。

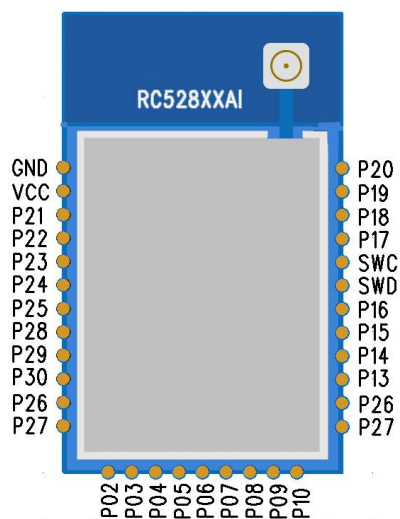
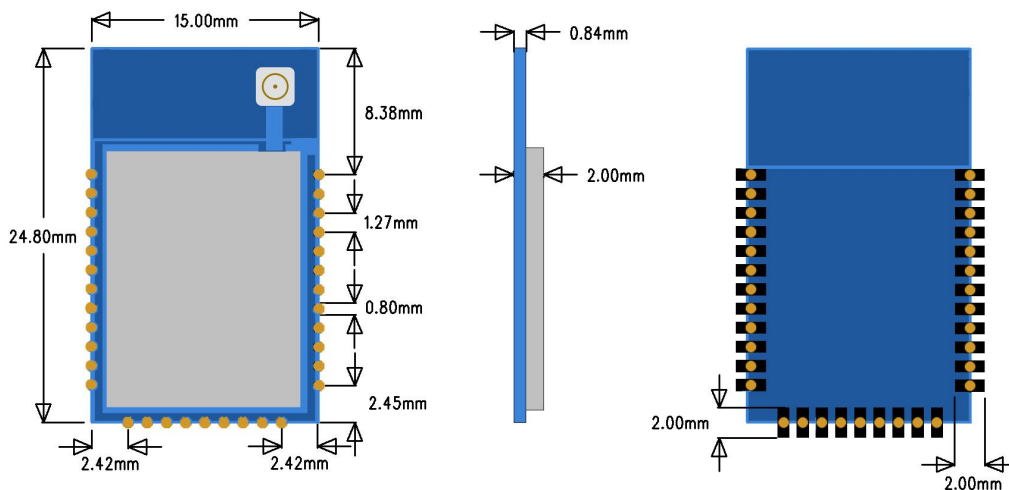
此模式下串口处于关闭状态不可用。

## ➤ 设备状态

- 1、DEVICE START: 设备启动;
- 2、S:CONNECTED: BLE 从角色连接成功;
- 3、S:DISCONNECTED: BLE 从角色断开连接;
- 4、B:CONNECTED: Beacon 连接成功;
- 5、B:DISCONNECTED: Beacon 断开连接;
- 6、DEVICE ERROR!: 设备发生异常错误。

以上状态可通过 AT 指令开启或关闭显示，详情请查看 AT 指令章节。

## 2. 封装尺寸及脚位定义



RC52810AI 模块引脚图

引脚定义表

引脚序号	名称	功能	备注
1	GND	模块地	模块地
2	VCC	电源正极输入	模块电源, 1.7~3.6V, 推荐 3.3V
3	P21/RST	复位	复位脚, 低电平有效
4	P22	I/O	
5	P23	I/O	
6	P24	I/O	
7	P25	I/O	
8	P28	I/O	
9	P29	连接指示状态	高电平: 未连接 低电平: 已连接
10	P30	广播状态指示	高电平: 未广播 低电平: 广播中
11	P26	I/O	
12	P27	I/O	
13	P02	I/O	
14	P03	I/O	
15	P04	I/O	
16	P05	I/O	
17	P06	I/O	
18	P07	I/O	
19	P08	I/O	
20	P09	I/O	
21	P10	I/O	



22	P11	I/O	
23	P12	I/O	
24	P13	TX	模块串口发送端
25	P14	RX	模块串口接收端
26	P15	I/O	
27	P16	I/O	
28	SWDIO	固件下载	固件下载数据 IO
29	SWCLK	固件下载	固件下载时钟 IO
30	P17	I/O	
31	P18	I/O	
32	P19	RESTORE	上电检测到置低 3 秒以上，所有参数恢复出厂设置
33	P20	I/O	

### 3. 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 连接间隔进行重设置 (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 115200bps。

串口在除了 **beacon** 的其他模式下默认都是开启的，若需要关闭，请使用“AT+SLEEP=1”指令关闭串口。

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 244 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 244 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。

串口硬件协议：115200 bps，8，无校验位，1 停止位。

### 4. BLE 协议说明(APP 接口)

#### ➤ Service UUID

6E400001B5A3F393E0A9E50E24DCCA9E

#### ➤ BLE 数据接收 UUID

特征值 UUID	6E400002B5A3F393E0A9E50E24DCCA9E
可执行的操作	Write
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。

#### ➤ BLE 数据发送 UUID

特征值 UUID	6E400003B5A3F393E0A9E50E24DCCA9E
可执行的操作	Notify
说明	串口输入转发到蓝牙输出，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。

## ➤ AT 指令操作 UUID

特征值 UUID	6E400004B5A3F393E0A9E50E24DCCA9E
可执行的操作	Write/Notify
说明	支持全部指令列表的指令，任何数据都会当做指令处理(不用+++进入指令模式)，且用户可不用发送回车换行符进行操作，串口是必须要回车换行符作为指令结束。 主角色需要打开 notify 才能收到模块发送的数据。

## 5. 串口 AT 指令

AT 指令可细分为四种类型：

类型	指令格式	描述
测试指令	AT+[x]=?	该命令用于查询设置指令的参数以及取值范围
查询指令	AT+[x]?	该命令用于返回参数的当前值
设置指令	AT+[x]=<...>	该命令用于设置用户自定义的参数值
执行指令	AT+[x]	该命令用于执行不可变参数的功能

### 注意：

- 1、指令既可以通过 APP 发送，也可以通过串口发送。
- 2、串口默认波特率为 115200，8bit 数据位，1 位停止位，无校验。
- 3、不是每条指令都具备上述 4 种类型的命令。
- 4、AT 命令必须大写，并且以回车换行符结尾(CRLF)。
- 5、AT 命令查询中返回的 <> 表示可选填参数，[] 表示必填参数；若命令所有参数都是选填参数，则至少填一个参数，否则也是为指令错误。

例：AT+ADS=<0,1>,<0,1>,<20,10240>，可填写为 AT+ADS=,,500。

- 6、选填参数命令为填写的参数位置**必须保留**，参考上一条举例。
- 7、任何指令中参数不能包含逗号，及空格、制表符等不可见字符。

AT 命令表

指令	功能	备注
<a href="#">+++</a>	进入 AT 命令模式	
<a href="#">AT+EXIT</a>	退出 AT 命令模式	立即生效
<a href="#">AT+NAME</a>	查询/设置设备名称	重启后生效；掉电保存
<a href="#">AT+VERSION</a>	查询设备固件版本	
<a href="#">AT+MAC</a>	查询/设置设备 MAC 地址	重启后生效；掉电保存
<a href="#">AT+ECHO</a>	查询/设置串口是否回显	立即生效；掉电不保存
<a href="#">AT+STATUS</a>	查询/设置是否显示设备状态	立即生效；掉电不保存
<a href="#">AT+ADS</a>	查询/设置 slave 角色广播参数	重启后生效；掉电保存
<a href="#">AT+CNT_INTERVAL</a>	查询/设置设备连接间隔	重启后生效；掉电保存
<a href="#">AT+SERVICE</a>	查询/设置 BLE service 相关参数	重启后生效；掉电保存
<a href="#">AT+DISCONNECT</a>	断开已连接设备	立即生效；掉电不保存
<a href="#">AT+BEACON</a>	查询/设置 beacon 相关参数	重启后生效；掉电保存

<a href="#">AT+ROLE</a>	查询/设置设备角色	重启后生效；掉电保存
<a href="#">AT+POWER</a>	查询/设置设备功率	重启后生效；掉电保存
<a href="#">AT+SLEEP</a>	查询/设置设备休眠(单次有效)	立即生效；掉电不保存
<a href="#">AT+UART</a>	查询/设置串口波特率	重启后生效；掉电保存
<a href="#">AT+AUTH</a>	查询/设置用户鉴权	重启后生效；掉电保存
<a href="#">AT+RESTART</a>	重启设备	立即生效
<a href="#">AT+RESET</a>	设备参数恢复出厂设置并重启	立即生效
<b>命令返回值</b>		
OK	指令操作成功	
FAIL	指令操作失败	
ERROR	指令操作错误	
BUSY	指令操作忙，请等待上一条操作	

### ➤ 进入 AT 指令模式

<b>+++</b>	
功能	进入 AT 指令
示例	+++
返回值	OK
说明	<b>需要退出 AT 指令模式才能切换进入到透传模式</b>

### ➤ 退出 AT 指令模式

<b>EXIT</b>	
功能	退出 AT 指令模式，切换进入到透传模式
示例	AT+EXIT
返回值	OK

### ➤ 设备名称

<b>AT+NAME?</b>	
功能	查询设备名称
示例	AT+NAME?
返回值	AT+NAME=RF-CRAZY OK
说明	指令正确返回设备名称

AT+NAME=	
功能	设置设备名称
示例	AT+NAME=TEST-NAME
返回值	OK
说明	设置成功后新的设备名称在重启后生效；掉电保存

## ➤ MAC 地址

AT+MAC?	
功能	查询设备 MAC 地址
示例	AT+MAC?
返回值	AT+MAC=8A:E5:84:7A:E7:C9 OK
说明	返回的 MAC 地址为 16 进制字符

AT+MAC=	
功能	设置设备 MAC 地址
示例	AT+MAC=F1:F2:F3:F4:F5:F6
返回值	OK
说明	设置成功后新的 MAC 地址在重启后生效；掉电保存

## ➤ 串口回显

AT+ECHO=?	
功能	查询此命令参数范围
示例	AT+ECHO=?
返回值	AT+ECHO=[0,1] OK
说明	0, 关闭回显；1, 打开回显。

AT+ECHO?	
功能	查询串口回显状态
示例	AT+ECHO?
返回值	AT+ECHO=0 OK
说明	0, 回显为关闭状态；1, 回显为打开状态。

AT+ECHO=	
功能	设置串口回显状态
示例	AT+ECHO=1
返回值	OK
说明	0, 关闭回显; 1, 打开回显。设定立即生效、掉电不保存。

## ➤ 显示设备状态

AT+STATUS=?	
功能	查询此命令参数范围
示例	AT+STATUS=?
返回值	AT+STATUS=[0,1] OK
说明	0, 状态显示功能关闭 1, 状态显示功能打开

AT+STATUS?	
功能	查询显示设备状态功能的当前状态
示例	AT+STATUS?
返回值	AT+STATUS=0 OK

AT+STATUS=	
功能	设置设备状态显示功能
示例	AT+STATUS=0
返回值	OK
说明	关闭设备状态显示功能。设定立即生效、掉电不保存。

## ➤ 设备角色

AT+ROLE=?	
功能	查询此命令参数范围
示例	AT+ROLE=?
返回值	AT+ROLE=[0,1] OK

说明	0: slave, 从角色 (默认角色) 1: beacon 重启设备生效
----	---

AT+ROLE?	
功能	查询设备当前角色
示例	AT+ROLE?
返回值	AT+ROLE=0 OK
说明	设备当前角色为 slave 从角色

AT+ROLE=	
功能	设置设备当前角色
示例	AT+ROLE=1
返回值	OK
说明	设置设备当前角色为 Beacon 角色 重启后生效、掉电保存

## ➤ 从角色广播参数

AT+ADS=?	
功能	查询此命令参数范围
示例	AT+ADS=?
返回值	AT+ADS=<0,1>,<0,1>,<20-10240> OK
说明	参数 1: 设备广播状态设置(0, 关; 1, 开, 立即生效) 参数 2: 设备广播模式设置(0, 不可连接广播; 1, 可连接广播, 重启后生效) 参数 3: 设备广播间隔设置(单位毫秒, 重启后生效)

AT+ADS?	
功能	查询广播参数
示例	AT+ADS?
返回值	AT+ADS=1,1,200 OK



说明	参数 1: 广播状态中 参数 2: 可连接广播 参数 3: 广播间隔为 200 ms
----	--

AT+ADS=	
功能	设置广播参数
示例	AT+ADS=1,0,500
返回值	OK
说明	设置开启不可连接、间隔为 500 ms 的广播。重启后生效、掉电保存。

### ➤ 连接间隔

AT+CNT_INTERVAL=?	
功能	查询此命令参数范围
示例	AT+CNT_INTERVAL=?
返回值	AT+CNT_INTERVAL=[8-4000] OK
说明	连接间隔参数范围为 8~4000 毫秒(参数不支持浮点数据, 设置 8ms 实际使用间隔为 7.5ms)

AT+CNT_INTERVAL?	
功能	查询设备当前连接间隔
示例	AT+CNT_INTERVAL?
返回值	AT+CNT_INTERVAL=10 OK
说明	设备当前连接间隔为 10 毫秒

AT+CNT_INTERVAL=	
功能	设置设备连接间隔
示例	AT+CNT_INTERVAL=20
返回值	OK
说明	设置设备当前连接间隔 20 毫秒。重启后生效、掉电保存。 连接间隔设置过后, 只有在设备初始化时生效, 相当于重启生效; 手机 App 连接最小连接间隔 20ms (苹果最小连接间隔 30ms), 模块连接可以达到 20ms 以下 (最小 7.5ms); 连接间隔生效后会根据不同的手机协调连接间隔, 最大协调到“设置的连接间隔+55ms”

## ➤ 从角色 Service

AT+SERVICE=?	
功能	查询此命令参数范围
示例	AT+SERVICE=?
返回值	AT+SERVICE=<0,1>,<0-FFFF>,<0-FFFF>,<0-FFFF>,<0-FFFF>,<0-FF...> OK
说明	参数 1: 128bit UUID 功能(0:16bit, 1: 128bit, 重启设备生效) 参数 2: 设备 service UUID(第 3、4 字节) 参数 3: 设备接收 channel UUID(128bit 模式的第 3、4 字节) 参数 4: 设备发送 channel UUID(128bit 模式的第 3、4 字节) 参数 5: 设备 AT 命令 channel UUID(128bit 模式的第 3、4 字节) 参数 6: 128bit 基础 UUID 值, (基础 UUID 的第 3、4 字节替换为以上参数的 UUID 构成设备 128bit 的实际 UUID) 此指令只对从角色生效

AT+SERVICE?	
功能	查询设备当前 service 配置参数
示例	AT+SERVICE?
返回值	AT+SERVICE=1,0001,0002,0003,0004,9ECADC240EE5A9E093F3A3B50000406E OK

AT+SERVICE=	
功能	设置设备 service 相关属性
示例	AT+SERVICE=0,FFF0,FFF1,FFF2,FFF3
返回值	OK
说明	设置设备为 16bit UUID 模式, UUID 分别为: FFF0,FFF1,FFF2,FFF3。 重启后生效、掉电保存。

## ➤ 断开连接

AT+DISCONNECT	
功能	断开当前所有连接状态
示例	AT+DISCONNECT
返回值	OK
说明	立即生效; 掉电不保存

## ➤ Beacon 参数

AT+BEACON=?	
功能	查询此命令参数范围
示例	AT+BEACON=?
返回值	AT+BEACON=<0-FFFF>,<0-FFFF>,<0-FFFF>,<-90-4>,<0-FF...> OK
说明	<p>查询 beacon 参数支持范围。</p> <p>参数 1: company id</p> <p>参数 2: major uuid</p> <p>参数 3: minor uuid</p> <p>参数 4: 1 米距离参考 rssi</p> <p>参数 5: 自定义 UUID 数据。</p> <p>注: 此模式下串口处于关闭状态不可用。</p>

AT+BEACON?	
功能	查询 Beacon 配置参数
示例	AT+BEACON?
返回值	AT+BEACON=0059,0102,0304,-50,0102030405060708090A0B0C0D0E0F10 OK

AT+BEACON=	
功能	设置 Beacon 参数
示例	AT+BEACON=F1F2,,,-60,
返回值	OK
说明	<p>设置 beacon company id: F1F2</p> <p>设置参考 rssi: -60</p> <p>重启后生效、掉电保存</p>

## ➤ 发射功率

AT+POWER=?	
功能	查询此命令参数范围
示例	AT+POWER=?
返回值	AT+POWER=[-40,-20,-16,-12,-8,-4,0,3,4] OK
说明	设备支持 9 档发射功率

AT+POWER?	
功能	查询设备当前发射功率
示例	AT+POWER?
返回值	AT+POWER=4 OK
说明	设备当前发射功率为 4 dBm
AT+POWER=	
功能	设置设备发射功率
示例	AT+POWER=-12
返回值	OK
说明	设置设备发射功率为-12 dBm 重启后生效、掉电保存

## ➤ 休眠模式

AT+SLEEP=?	
功能	查询此命令参数范围
示例	AT+SLEEP=?
返回值	AT+SLEEP=<0,1>,<0,1>,<0,1> OK
说明	参数 1: 设备串口功能开关 参数 2: 设备 BLE 功能开关 参数 3: 设备串口唤醒功能开关 0, 关闭; 1, 打开

AT+SLEEP?	
功能	查询设备当前休眠模式
示例	AT+SLEEP?
返回值	AT+SLEEP=1,1,1 OK
说明	参数 1: 设备串口功能开 参数 2: 设备 BLE 功能开 参数 3: 设备串口唤醒功能开, 串口收到任意数据均会重新打开串口

AT+SLEEP=	
功能	设置设备休眠模式

示例	AT+SLEEP=0,0,0
返回值	OK
说明	关闭 BLE 功能，若设备处于连接状态则会立即断开当前连接 设置立即生效、掉电不保存

## ➤ 串口波特率

AT+UART=?	
功能	查询此命令参数范围
示例	AT+UART=?
返回值	AT+UART=[4800,9600,38400,57600,115200,230400,250000,460800] OK
说明	设备支持 8 种串口波特率

AT+UART?	
功能	查询当前串口波特率
示例	AT+UART?
返回值	AT+UART=115200 OK
说明	当前串口波特率为 115200 bps

AT+UART=	
功能	设置串口波特率
示例	AT+UART=9600
返回值	OK
说明	设置串口波特率为 9600 bps，设备重启后生效，掉电保存

## ➤ 用户鉴权

AT+AUTH=?	
功能	查询此命令参数范围
示例	AT+AUTH=?
返回值	AT+AUTH=<0,1>,<*****>,<1-65535> OK

说明	查询参数列表和取值范围 参数 1: 关闭/启用用户鉴权 参数 2: 密钥, 最大 16 字节任意可见字符 参数 3: 鉴权有效时间(秒) <b>备注: 启用鉴权后重启生效, 且有效时间内未收到主角色鉴权密钥从角色会自动断开连接。(数据传输特征值发送鉴权密钥)</b>
----	---

AT+AUTH?	
功能	查询用户鉴权功能当前状态
示例	AT+AUTH?
返回值	AT+AUTH=1,12GH**__),15 OK
说明	参数 1: 1, 用户鉴权功能已启用 参数 2: 密钥为 12GH**__)) 参数 3: 鉴权有效时间为 15 秒

AT+AUTH=	
功能	设置用户鉴权功能
示例	AT+AUTH=1,12GH**__),10
返回值	OK
说明	启用用户鉴权, 设置密钥为“12GH**__))”, 有效时间为 10 秒 重启后生效、掉电保存。

## ➤ 设备重启

AT+RESTART	
功能	设备重启
示例	AT+RESTART
返回值	OK
说明	设置成功后设备立即重启

## ➤ 恢复出厂设置

AT+RESET	
功能	恢复设备出厂设置
示例	AT+RESET
返回值	OK

说明	设置成功后设备立即重启
----	-------------

## ➤ 固件版本

AT+VERSION	
功能	查询设备固件版本
示例	AT+VERSION
返回值	AT+VERSION=v0.1.0, Apr 13 2021, 17:40:42 OK
说明	获取设备固件版本信息及时间

## 6. 用 APP 测试透传功能

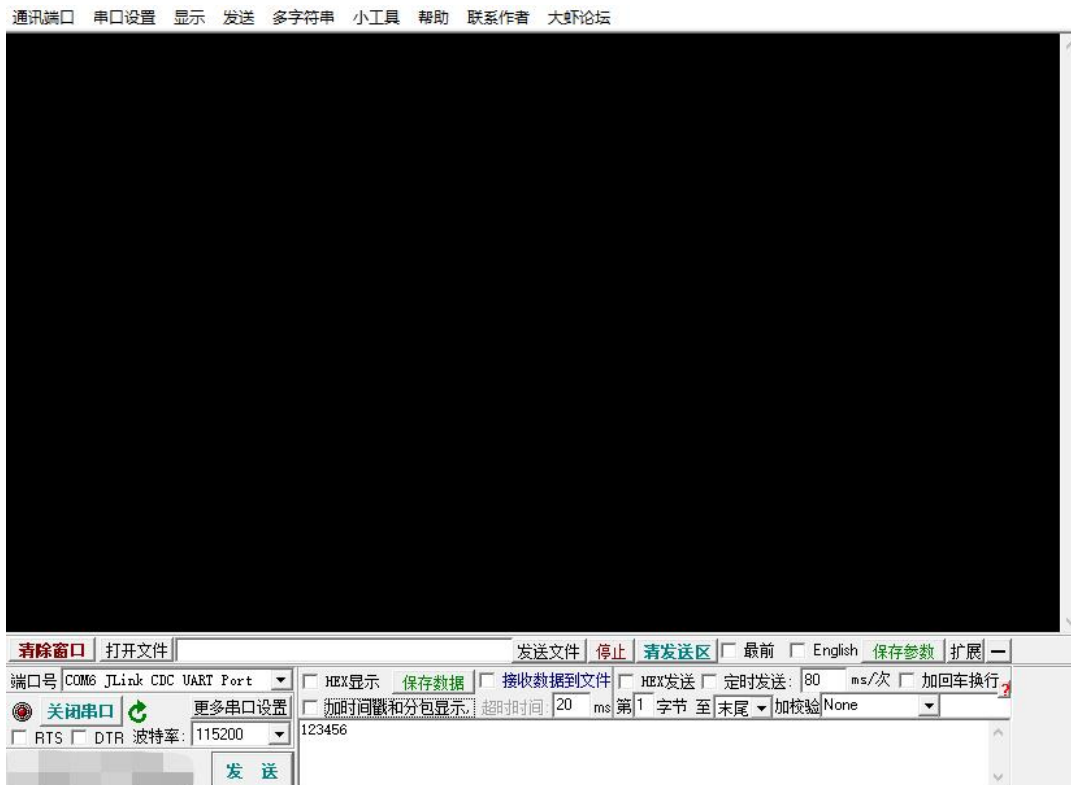
模块的手机端测试工具(APP)可以在 App Store 和应用市场下载到。打开 App Store 和应用市场，搜索 nRF Connect 并下载安装，进行测试。（此文档以 Android 版本的 nRF Connect 为例，iOS 操作方法大同小异。）



nRF Connect

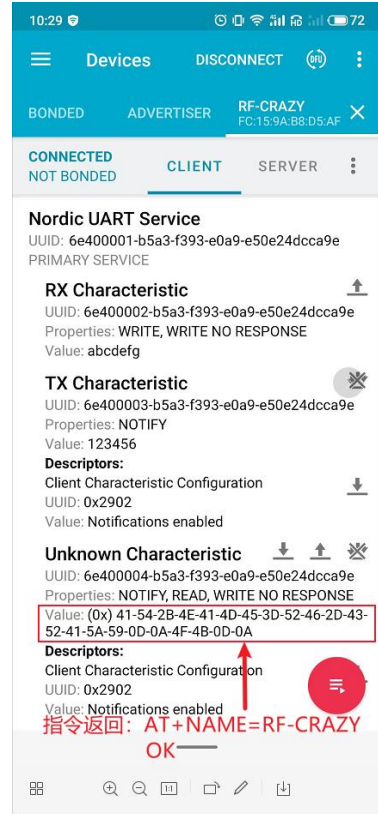
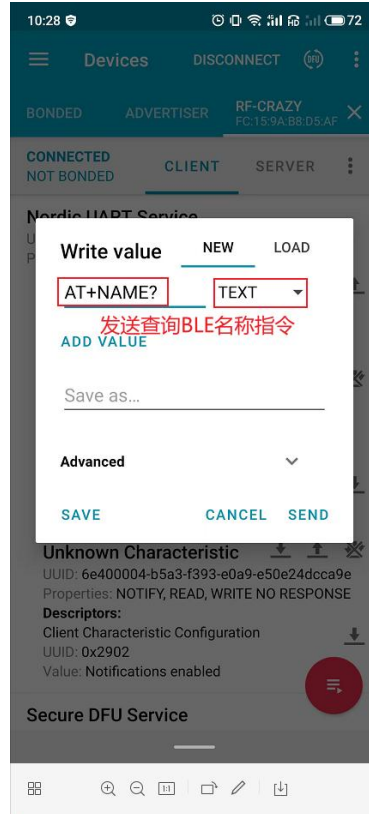
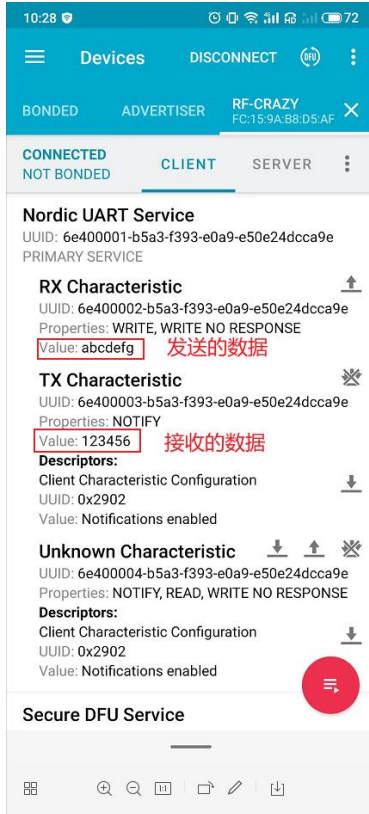
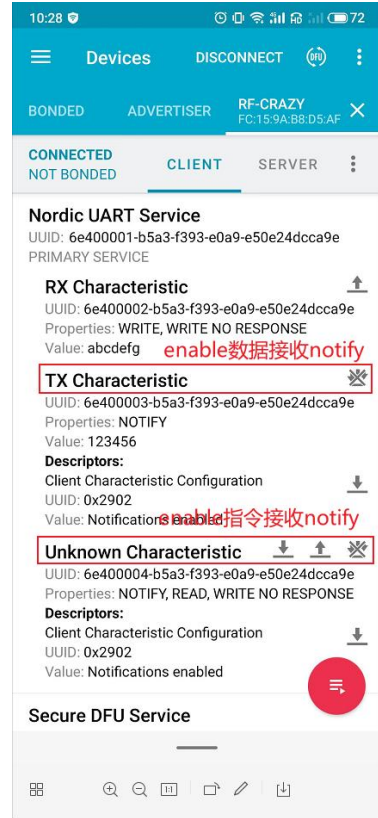
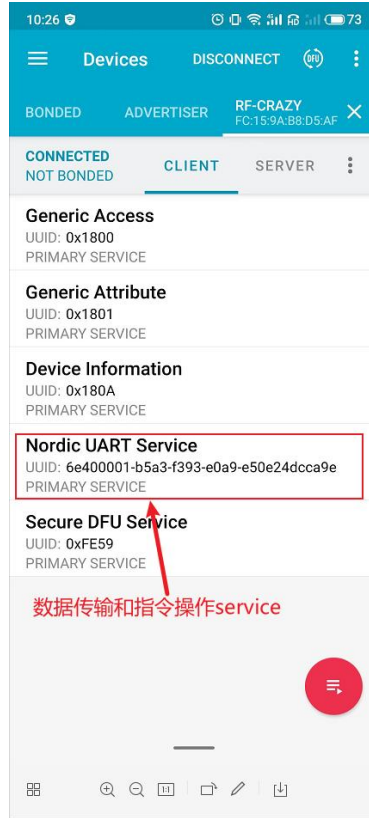
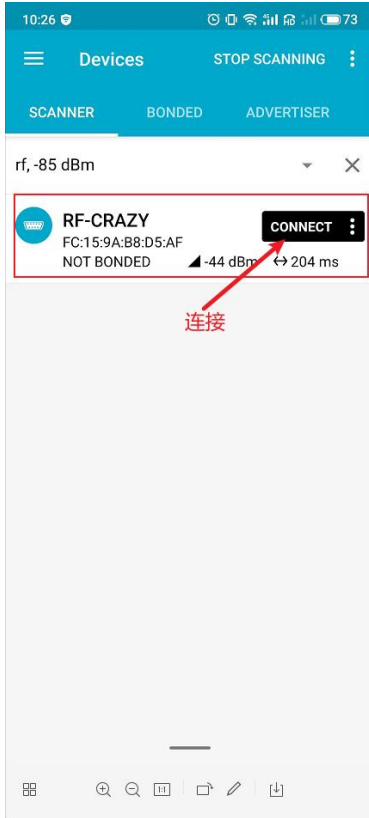
将模块通过 USB 转串口工具连接到电脑 COM 口上，并查看使用的电脑端口号（步骤：右击计算机 → 管理 → 设备管理器 → 端口）。

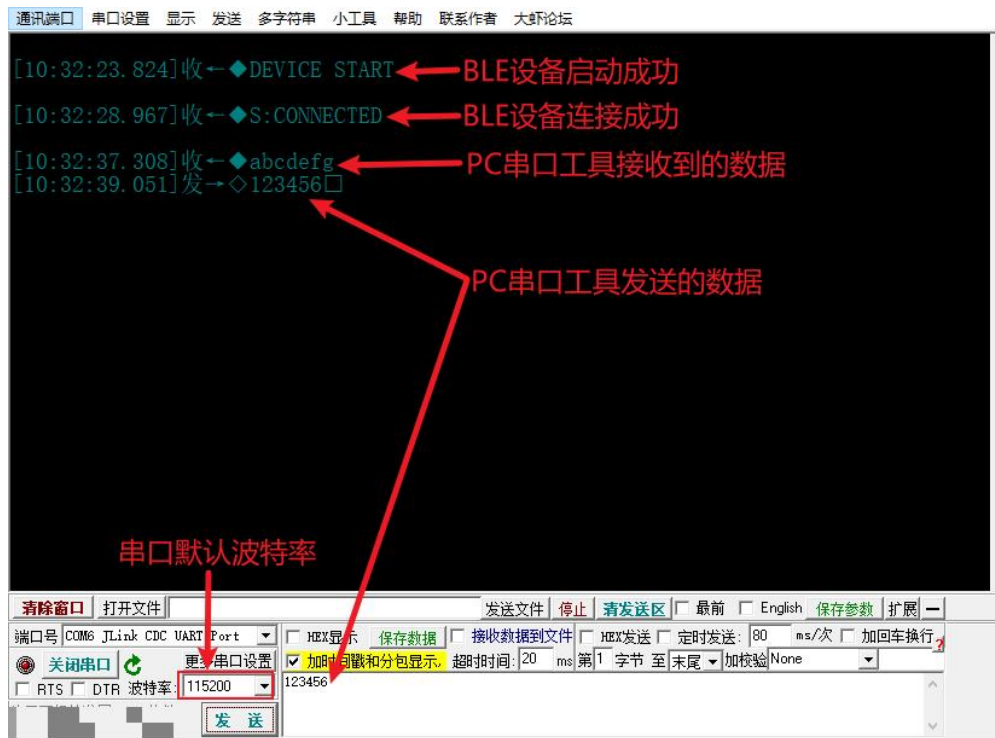
打开串口调试工具，设置正确的端口号和波特率，模块的初始波特率115200（模块默认波特率为115200，数据位为8，校验位为无，停止位为1）。



打开 APP 搜索（手机蓝牙需打开），会出现附近正在广播的 BLE 设备列表。点击其中一个 BLE 设备，会开始进入连接过程（**RC52810AI 的默认出厂名称为：RF-CRAZY**）。连接成功后，手机 APP 端出现模块的 Service UUID，电脑串口调试工具端出现 S:CONNECTED。在 APP 找到数据传输和指令操作 Service，打开接收和 AT 指令操作的 notify 使能，之后就可以开始进行手机与 PC 端（模拟 MCU）之间的双向数据透传和 AT 指令了。如下几张图所示：







## 7. PC 端测试工具

### ➤ 使用前准备

#### 1、硬件需求：

- 带有标准 Type-A USB 接口的 PC。
- nRF52840 Dongle
- RC52810AI、RC52832A/B 模块

#### 2、软件需求：

- nRF Connect for Desktop 客户端

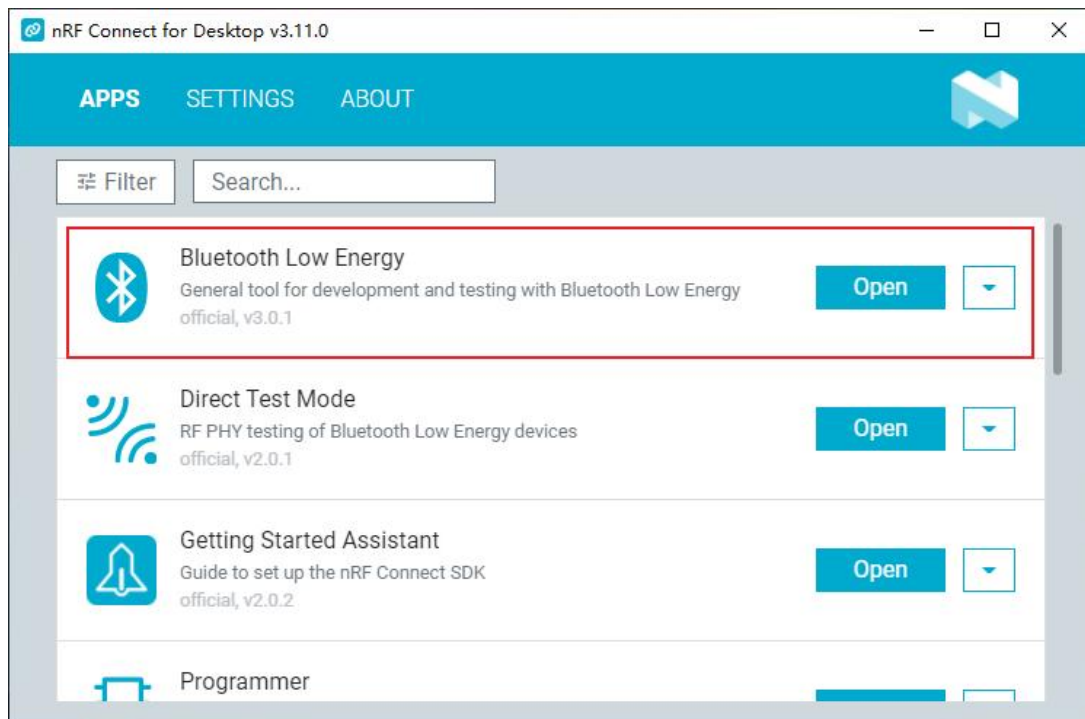
下载地址：

<https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-desktop/Download#infotabs>

- 操作系统： macOS, Linux, or Windows 7 or later

### ➤ 连接步骤说明

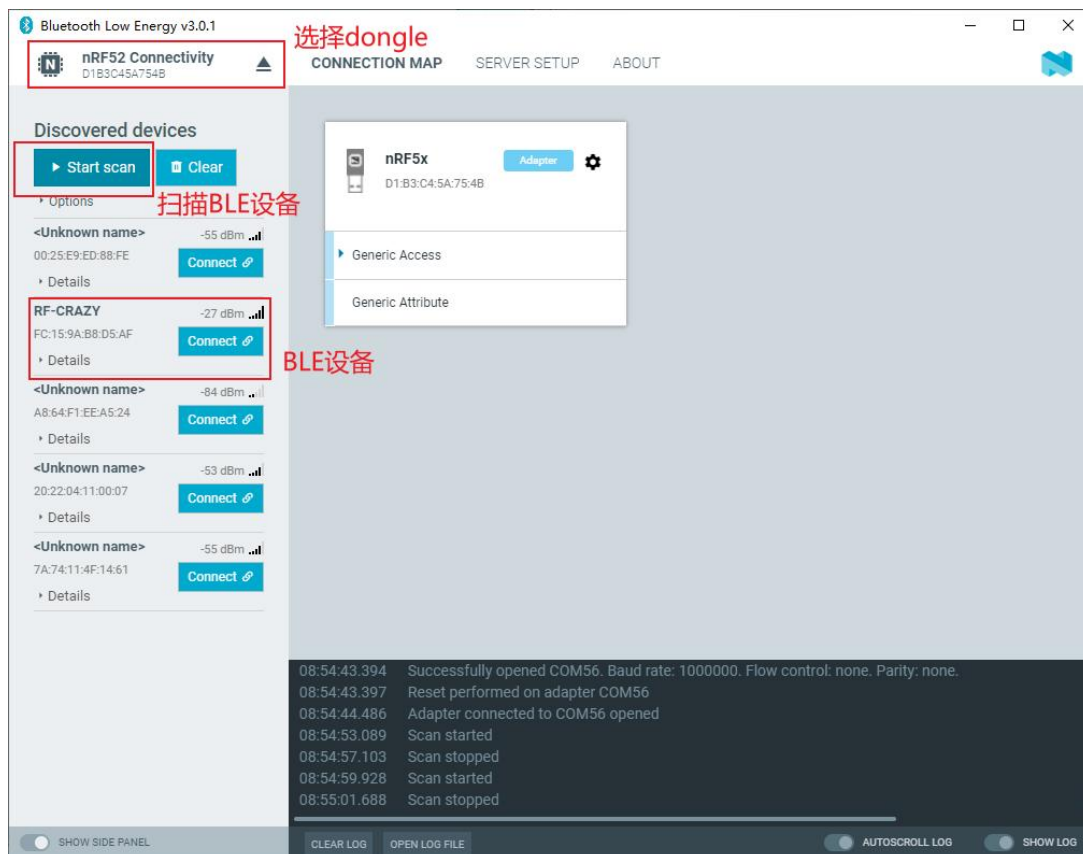
1、软件安装成功后，将 dongle 通过 USB 连接至 PC 端，打开 nRF Connect for Desktop，选择 Bluetooth Low Energy，如果没有 Open 选项，则需要先点击 Install 安装，如下图。

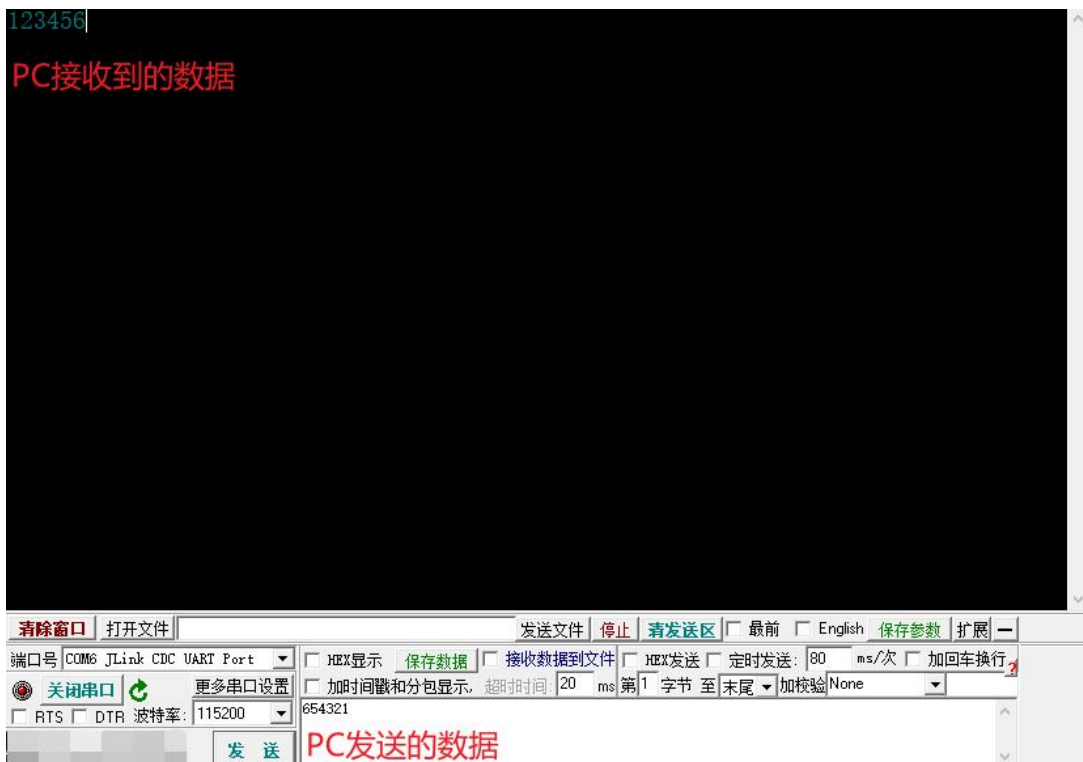
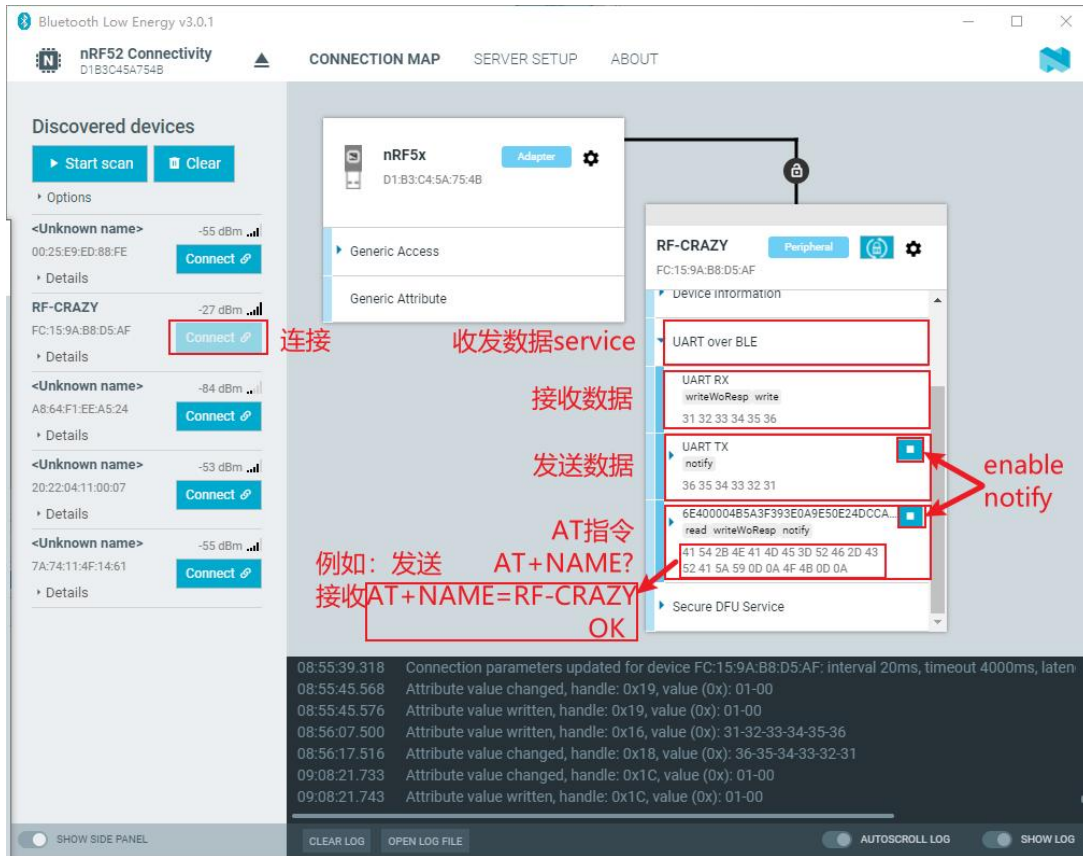


2、安装完后点击 Open 按钮后会出现操作界面，这时候还需选择对应的 Dongle 设备才能正常使用，点击 Select device 会下拉弹出设备名称，然后点击选择相应设备（这个设备名称实际对应设备的 MAC 地址）。

3、Dongle 设备选择成功之后，可以点击 **Generic Access** 查看设备作为主机的相关信息。接着点击 **Start scan** 按钮就可以开始扫描广播设备了，之后会将扫描到的从机设备列在右侧的窗口内，找到对应从机设备点击 **Connect** 连接后（RC52810AI）的默认出厂名称为：**RF-CRAZY**。

4、设备连接成功之后，选中打开“**UART over BLE**”服务，使能接收通道的 **Notify**，之后就可以开始进行模块与 PC 端（借助 Dongle 作为主设备）之间的双向数据透传和 AT 指令了。如下几张图所示：





## 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:(CBCharacteristicWriteType)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 to.
* @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。

## 9. 主机（MCU）参考代码（透传）

示意性代码如下：

```
void main(void)
{
    //等待 BLE 模块启动成功
    while (1) {
        if(!memcmp(mcu_uart_read_data(),"DEVICE START\r\n",strlen("DEVICE START\r\n")))
        {
            break;
        }
    }

    //延迟 50 毫秒
    mcu_delay_ms(50);

    while (1) {
        //打印串口收到 BLE 模块的数据(数据来自于远端连接设备)
        if(mcu_uart_get_data_len() > 0) {
            mcu_uart_log(mcu_uart_get_data(),mcu_uart_get_data_len());
        }

        //延迟 50 毫秒
        mcu_delay_ms(50);

        //发送数据给 BLE 模块(BLE 模块会传输给远端连接的设备)
        mcu_send_data_to_ble_mode("test data.");
    }
}
```



## 联系我们

想要了解更多产品信息，请登录 <http://www.rfcrazy.com> 访问我们。

深圳市智汉科技有限公司

ShenZhen RFCrazy Technology CO.,LTD.

技术服务电话: 86-134 1739 4552 邮箱: [sales@rfcrazy.com](mailto:sales@rfcrazy.com)

地址: 深圳市宝安区西乡街道宝源路华源科技创新园 A 座 3 楼

## 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, 血糖, 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, 电池保护板, 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, 胎心仪, 手冲咖啡秤, 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, 育婴用品(实时体温检测, 呼吸机, 吸奶器, 电动摇椅, 纸尿裤接收器), 家居类电子产品(香薰机, 甲醛检测仪), 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。

## 附录 B：BLE 模块硬件规格说明

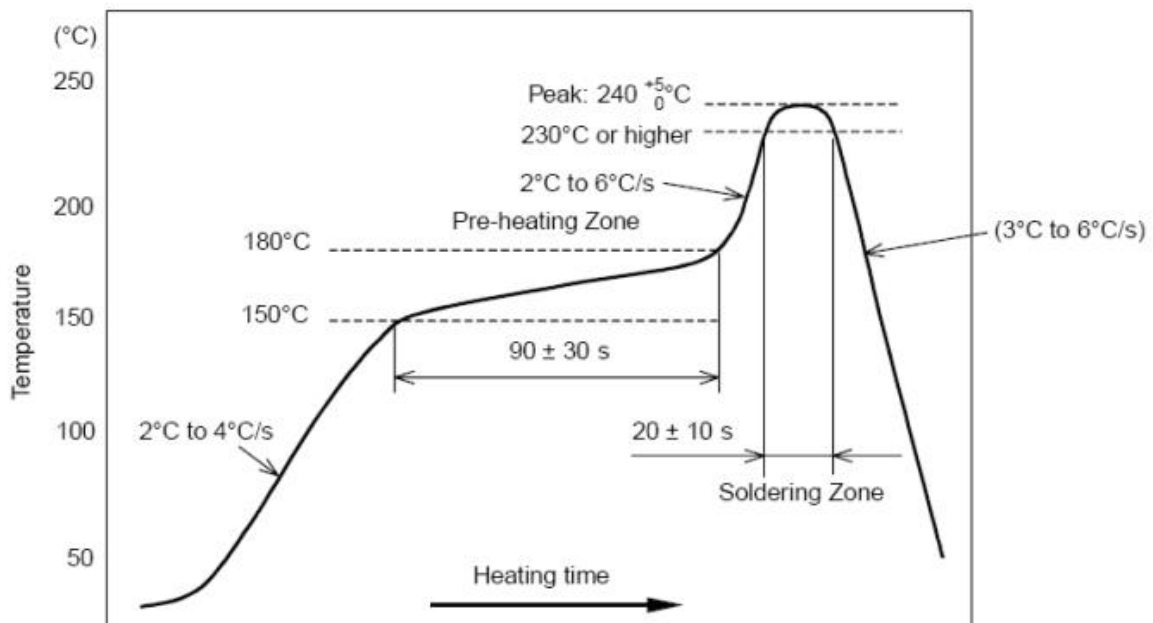
### 一、模块参数

- 工作电压：1.7 V~3.6 V，推荐工作电压：3.3 V
- 工作频段：2402 MHz~2480 MHz
- 最大发射功率：4 dBm (-20 dBm~+4 dBm，可编程)
- 接收灵敏度：-96 dBm
- 频率误差：±20 kHz
- 工作温度：-40°C ~ +85°C
- 储存温度：-40°C ~ +125°C

### 二、注意事项

使用本模块注意事项：

- 1、在运输、使用过程中要注意防静电。
- 2、器件接地要良好，减少寄生电感。
- 3、尽量手工焊接，如需机贴，请控制回流焊温度不要超过 245 摄氏度，如下图所示。
- 4、模块天线下面不要铺铜，最好挖空，以防止阻抗改变。
- 5、天线应远离其他电路，防止辐射效率变低和影响其他电路正常使用。
- 6、模块的接入电源建议使用 2.2uF+0.1uF 滤波电容对地。



部件的焊接耐热性温度曲线（焊接点）