



产品规格书

产品名称：蓝牙数据模块

产品型号：F-9088

文件编号：XZX-SPEC-BT-RD-018

版本：V1.8

生效日期：2017-12-14

编制	审核	批准

(此说明书对应的软件版本为 9088 串口透传 V5.0)

文件含芯中芯 (C-CHIP)机密文件，未经许可，不可外传



目录

一、产品概述.....	4
1.1 概述.....	4
1.2 模块特点.....	4
1.3 模块参数.....	5
1.4 模块引脚定义.....	6
1.5 引脚功能说明.....	7
二、应用.....	8
2.1 应用领域.....	8
2.2 应用举例.....	8
2.3 低功耗的应用.....	8
三、通讯模式.....	9
3.1透传模块与MCU（STM8L152C6T6)的连接.....	9
3.2MCU主机透传控制说明及程序参考.....	9
四、AT指令测试.....	12
4.1 指令集.....	12
4.2 指令说明.....	13
4.3 AT指令测试.....	16
4.3.1 测试环境搭建.....	17
4.3.2 查询指令的测试.....	21
五、串口透传测试.....	25
5.1 串口到蓝牙的透传测试.....	25
5.2 蓝牙到串口的透传测试.....	33
六、PP和MCU编程参考.....	38
6.1 IOS编程参考.....	38
6.2 安卓编程参考.....	40
6.3 IOS、安卓、MCU编写者需要的参数知识.....	40
七、客户定制方案规则约定与参考电路.....	41



一、产品概述

1.1 概述

F - 9088 蓝牙模块为 BLE 单模数据传输模块（适合小数据低功耗传输，不支持语音，主要用于控制），现我司给客户样品模块为串口透传模块，其他方案、应用则需要定制（设计约定与参考电路见说明书末尾《十五、客户定制方案规则 约定与参考电路》）。

1.2 模块特点

- 1.2.1 芯中芯已率先完美解决安卓系统、IOS 系统同时与F-9088 蓝牙模块双向透传的问题。（安卓4.4 可完美兼容，4.3 系统只支持单向）
- 1.2.2 用户接口使用通用串口设计，全双工双向通讯，波特率范围9600 ~ 115200bps。
- 1.2.3 默认200ms 连接间隔，连接快速。
- 1.2.4 支持AT 指令软件复位模块，获取软件版本信息。
- 1.2.5 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）。
- 1.2.6 支持 AT 指令调整发射功率，修改广播间隔，自定义广播数据，设定数据延时（用户CPU 串口接备时间），修改串口波特率，修改模块名，均掉电保存。
- 1.2.7 串口数据包长度每次20byte 透传。
- 1.2.8 低功耗的待机模式，CC2541 芯片官方数据睡眠电流0.4uA，模块实测功耗如下：

事件	平均电流	测试条件
休眠	1uA	
广播	500uA	广播间隔为100ms ， RTS = 1
连接	500uA	连接间隔为500ms ， RTS = 1
串口发送	8mA	RTS = 1
串口接收	8mA	RTS = 0

备注：表格中的RTS 为模块串口接收使能脚的状态。以上数据为 F-9088 蓝牙模块抽样实测数据，以上表格中的广播、连接和串口发送事件的平均电流是在串口接收未使能的状态下测量得到，当串口接收被使能，模块功耗会到达8mA，若希望得到更低功耗，可适当增大连接间隔或者广播周期。

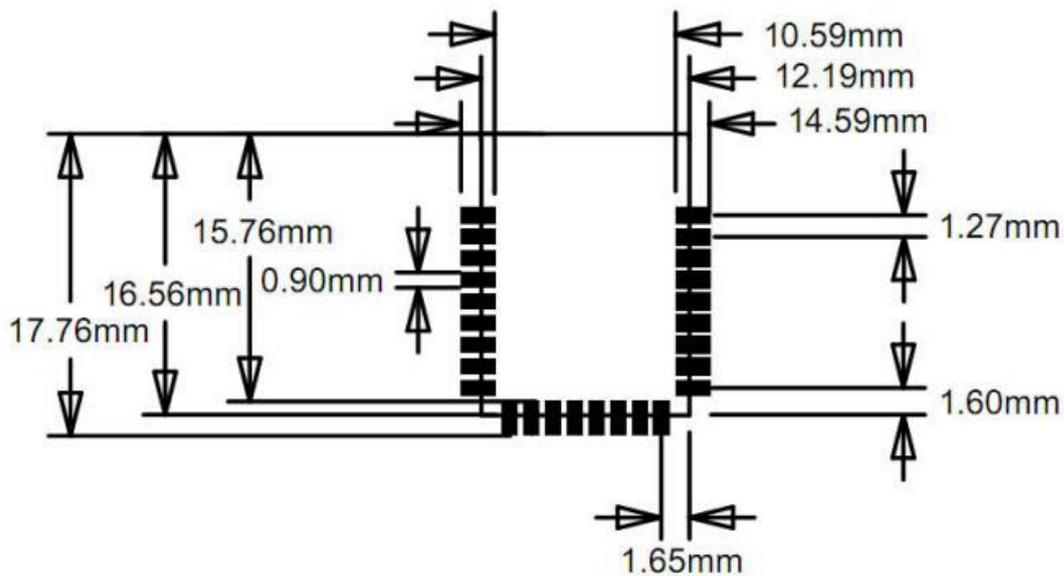
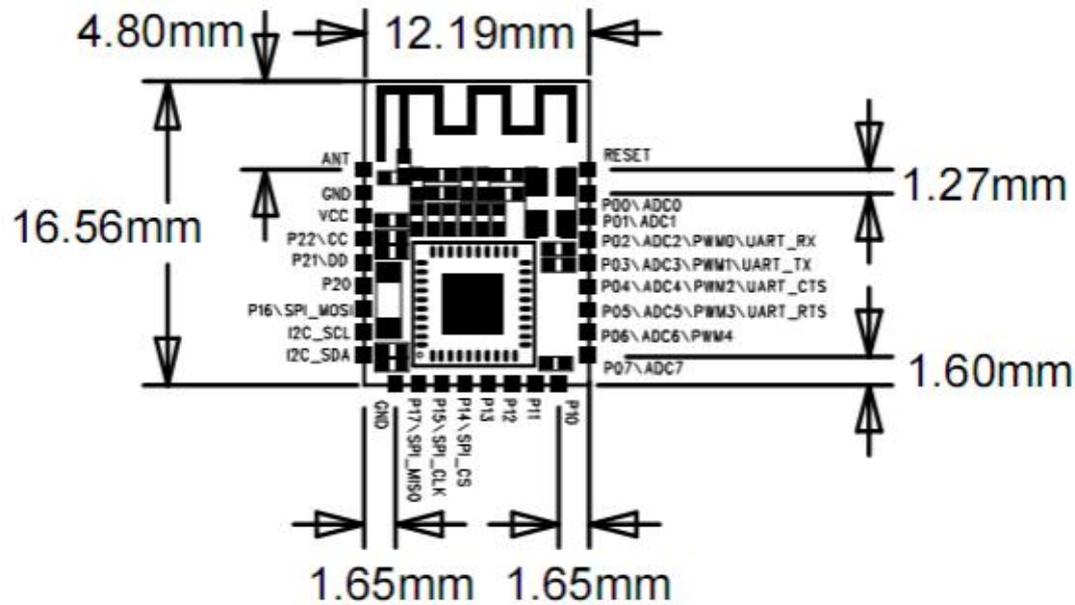


1.3 模块参数

型号	F-9088
蓝牙规格	Bluetooth V4.0
供电电压	2.0-3.6V
支持 V4.0 蓝牙协议栈	ATT,GATT,SMP,L2CAP,GAP
工作电流	≤10mA(简单应用 200uA~1mA)
休眠电流	低于 5uA
温度范围	-40°C to +80°C
无线传输范围	0~100 米
传输功率	最大可调4dBm
灵敏度	-93dBm<0.1%BER
频率范围	2.4GHz-2.480GHz
对外接口	IO,UART,SPI,USB2.0,PWM,ADC
模块尺寸	16.6mm*12.2mm*1.8



1.4 模块引脚定义





1.5 引脚功能说明

引脚号	引脚名	输入输出	说明备注
1	ANT	----	天线发射
2、10	GND	----	模块地
3	VDD	----	模块电源正极 2.0-3.6v
4	P2.2/CC	I/O	可编程双向 IO \CC
5	P2.1/DD	I/O	可编程双向 IO\CD
6	P2.0	I/O	可编程双向 IO
7	P1.6/SPI_MOSI	I/O	可编程双向 IO
8	USB_DP\I2C_SCL	I/O	USB 正极/I2C_SCK
9	USB_DN\I2C_SDA	I/O	USB 负极/I2C_SDA
11	P1.7\SPI_MISO	I/O	可编程双向 IO\SPI_MISO
12	P1.5\SPI_CLK	I/O	可编程双向 IO\SPI_CLK
13	P1.4\SPI_TMS	I/O	可编程双向 IO\SPI_TMS
14	P1.3	I/O	可编程双向 IO
15	P1.2	I/O	可编程双向 IO
16	P1.1	I/O	可编程双向 IO
17	P1.0	I/O	可编程双向 IO
18	P0.7\ADC7	I/O	可编程双向 IO\ADC7
19	P0.6\PWM4\ADC6	0	模块地蓝牙连接状态 0: 蓝牙状态为已连接 1: 蓝牙状态为未连接
20	P0.5\UART_RTS\ADC5\PWM3	I	作为数据发送请求 (RTS, 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
21	P0.4\UART_CTS\ADC4\PWM2	0	数据输入信号 (CTS 用来唤醒主控 MCU) 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
22	P0.3\UART_TX\ADC3\PWM1	0	模块串口 UART_TX
23	P0.2\UART_RX\ADC2\PWM0	I	模块串口 UART_RX
24	P0.1\ADC1	I/O	可编程双向 IO\ADC1
25	P0.0\ADC0	I	模块蓝牙功能控制引脚 0: 模块蓝牙功能打开 1: 模块蓝牙功能关闭
26	RESET	I	模块复位, 低有效



二、应用

2.1 应用领域

- 》 运动领域
- 》 安全领域、
- 》 智能家居领域、
- 》 工业自动控制领域、
- 》 移动电话配件领域、
- 》 室内定位等控制领域。
- 》 医疗和健康护理领域、

2.2 应用举例

- 》 手表
- 》 防丢器
- 》 心率计
- 》 体重计
- 》 电子计步器
- 》 血压计和血糖计

2.3 低功耗的应用

F-9088串口透传具有两种工作模式：（1）低功耗模式，（2）非低功耗模式。

（1）**低功耗模式**：在低功耗模式下，模块具有极低的功耗，所以低功耗模式适合于具有低功耗要求的电路设计中，同时在低功耗模式下具有两个使能端P0⁰和P0⁵，P0⁰为蓝牙广播使能端，P0⁵为串口透传使能端。当休眠状态下，只需要给P0⁰一个下降沿后，BLE模块进入广播，P0⁵为低电平时使能串口透传。（注意：默认是不开启低功耗模式的）

（2）**非低功耗模式**：非低功耗模式具有上电自动广播，并可以直接使用串口透传的特点，无需控制P0⁰和P0⁵两个使能端，所以非低功耗模式的使用较为方便。

（3）**低功耗与非低功耗模式切换**：F-9088 串口透传的这两种模式都可以相互切换，通过AT指令“AT+SLPM+ Para”（Para：0或1）可以切换两种模式，当Para取1时进入低功耗模式，当Para取0时进入非低功耗模式（具体设置见“四、AT指令测试”）。

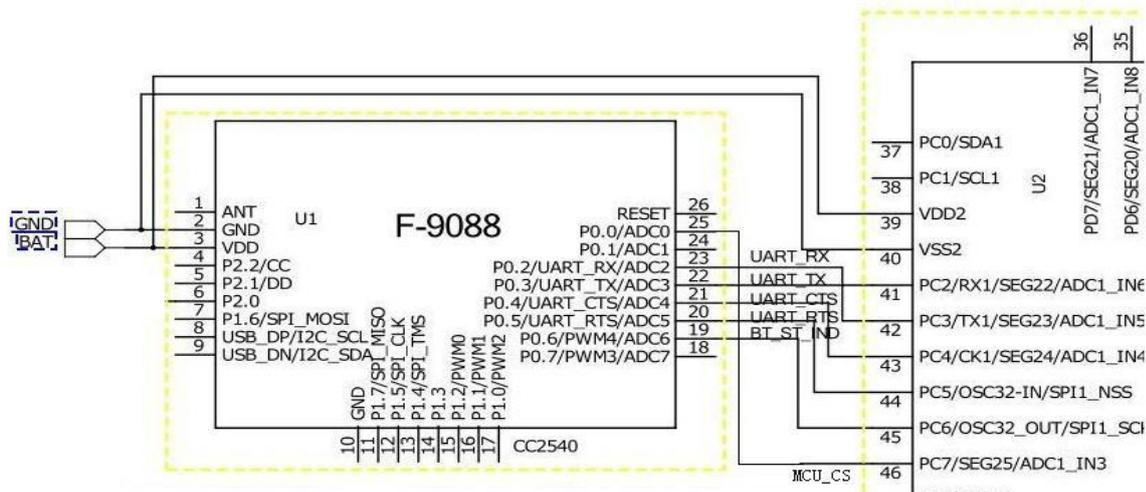


三、通讯模式

模块的工作模式为透传模式。

透传模式下，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。

3.1 透传模块与 MCU (STM8L152C6T6) 的连接



说明：由于新接触蓝牙 BLE 透传的客户使用全线串口接法有一定的难度，所以默认软件没有使用全线串口接法，默认软件使用 4 线即可测试通信(即 VCC,GND,UART_TX,UART_RX)，如果实际应用需要全线功能，使用对应 AT 指令使能改功能即可！

3.2 MCU主机透传控制说明及程序参考

3.2.1 引脚逻辑关系

P0.2/RX <--- MCU TX

P0.3/TX ---> MCU RX

P0.4/CTS ---> MCU RTS (Low -Module have data to send , High - Module no data to send)

P0.5/RTS <--- MCU CTS (Low -MCU(Master)have data send , High - MCU no data to send)

P0.6/ BT_ST_IND ---> MCU BT_ST_IND(High - Not Connect, Low - Connect)

P0.0/ CS ---> MCU CS

CTS、RTS 是硬件流控，若您的 MCU 没有硬件流控，则用普通 IO 口操作即可。为什么会有这 2 根线？也许有客户会有此疑问。如此做是因为，模块 F-9088 在接收 MCU 的数据时电流无可避免的达到 8mA（BLE 模块发给 MCU 数据时最高电流为 2.6mA）。因此每



次发完后一定需要将RTS 拉高（BLE 模块接收到手机的数据先拉低CTS，发送完毕后，自动拉高CTS，电流会自动回落到1mA 以下）。

BT_ST_IND 是指示 BLE 模块蓝牙的连接状态，方便 MCU 端的数据传输、时序调整等动作。Module_CS 是 MCU 控制 BLE 模块状态引脚。当休眠状态下，Module_CS 脚为低电平后，BLE 模块进入广播；当 MCU 想要断开与手机连接时，只需要在连接状态下给 Module_CS 一个高电平信号。

通信过程

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。为协调低速 CPU 的使用，默认波特率为 115200bps，在有大数据量传输，或者高实时性需求的应用中，建议设定为高速串口波特率 115200bps，支持掉电保存。当模块 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（4K/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。

- （1）、串口硬件协议：波特率 115200bps，数据长度 8，无校验位，停止位长度 1。
- （2）、Module_CS 为高电平，蓝牙模块处于完全睡眠状态。Module_CS 为低电平时，模块开始广播，默认广播间隔为 100ms，直到和手机对接成功，此时模块将 BT_ST_IND 拉低。当 Module_CS 在连接状态时，收到一个高电平信号将会断开连接。
- （3）、连接成功之后，MCU 如有数据发送至 BLE 模块，需将 RTS 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 RTS，让模块退出串口接收模式。
- （4）、当模块有数据上传请求时，模块会置低 CTS，最快会在 500us 之后开始发送，直到数据发送完毕。据发送完毕，模块会将 CTS 置高。
- （5）、如若主机的 RTS 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）、模块的蓝牙默认连接间隔 500ms，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（最长连接间隔 2000ms），每个连接间隔最多传输 80 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：

$$V = 80 * 1000 / T (V \text{ 只和 } T \text{ 有关})$$

如果模块的蓝牙连接间隔为 20ms，而每个间隔最多传输 80byte，因此理论最高传输能力（转发速率）为 $80 * 50 = 4K \text{ byte/s}$ 。测试表明，转发速率在 2K/s 以下，漏包机率很低。安全起见，无论是低速还是高速数据转发应用，都建议在上层做校验重传处理。



MCU 参考代码

```
void main(void)
{
    Module_CS = 0 ; halMcuWaitMs(1);
        //延迟1ms
    Module_CS = 1 ; //实现一个下降沿信号，使能模块，开始广播
    while(BT_ST_IND); //等待手机端与BLE 模块连接，判断连接提示信号线的电平
    while(1) //循环收发测试
    {
        if(CTS == 0) //检测，若BCTS 置低则准备接收
        {
            while(CTS==0); //等待发送完毕，也可限时等待
            if(UARTRead(uartBuffer) == SUCCESS) //串口读取数据
            {
                ... //数据处理
            }
        }
        BRTS = 0; //RTS 置低通知2540 模块准备接收
        halMcuWaitMs(2); //延迟2ms send_TX("12345678901234567890"); //
        发送任意数据（20byte） halMcuWaitMs(50); //延迟50ms(不同的波特率、连接间
        隙，时间不同) BRTS = 1; //RTS 置高，发送完毕
    }
}
```



四、AT指令测试

(1)、透传模式传数据时，请不要将“AT+”(十六进制 41 54 2B)作为透传的数据头。透传时的格式，需自行定义。不论是否包含通信头，校验。每次透传以 20 字节进行发送。本模块为从模块，相关UUID (16 位)如下：

SeviceUUID : 0xFFFF0

CharacteristicUUID :0xFFFF4

(2)、命令模式时，“AT+”字符串的16 进制编码是41 54 2B，\CR\LF 的16 进制为0D 0A，请客户注意。每次设置了某个项目后，由于基本都是掉电保存项目，因此需要上电重启模块或使用AT 指令复位。

4.1 指令集

	指令	功能
基础指令	AT+RSET	模块复位
	AT+CONB	断开连接
查询指令	AT+ROLE	查询 - 模块主从模式
	AT+VERS	查询 - 软件版本
	AT+GADD	查询 - 模块地址
	AT+GNAM	查询 - 模块名称
	AT+GCON	查询 - 连接间隔
	AT+GPWR	查询 - 发射功率
	AT+GLPM	查询-低功耗模式
	AT+GPAM	配对模式
	AT+GUAR	串口波特率
设置指令	AT+SUAR	设置 - 串口波特率
	AT+SNAM	设置 - 模块名称
	AT+SCON	设置 - 连接间隔
	AT+SPWR	设置 - 发射功率
	AT+SPAM	设置-配对模式
	AT+SLPM	设置-低功耗模式



4.2 指令说明

AT+ROLE

AT+ROLE\CR\LF: 查询 模块主从模式		
查询命令:	响应	ROLE:SLAVE
AT+ROLE\CR\LF	说明	None
举例: 发送查询命令: AT+ROLE, 返回: ROLE:SLAVE		

AT+VERS

AT+VERS: 查询 - 软件版本		
查询命令:	响应	Version:5.0
AT+VERS\CR\LF	说明	None
举例: 发送查询命令: AT+VERS, 返回软件版本信息: Version:5.0		

AT+GADD

AT+GADD: 查询 - 模块地址		
查询命令:	响应	BLE ADDRESS: Para
AT+GADD\CR\LF	说明	Para:12 位蓝牙地址
举例: 发送查询命令: AT+GADD, 返回12位地址: BLEADDRESS:0xB85FF98FC320		

AT+GNAM

AT+GNAM: 查询 - 模块名称		
查询命令:	响应	NAME :Para
AT+GNAM\CR\LF	说明	Para: 模块名称
举例: 发送查询命令: AT+GNAM, 返回当前的名称: NAME:CC2541_SPP		

AT+SNAM

AT+SNAM: 设置 - 模块名称		
设置命令:	响应	Ok
AT+SNAM+Para	说明	Para:字符串名字, 最长 20 字节
举例: 设置名称为 xinzhongxin, 则发送指令: AT+SNAM+xinzhongxin, 设置成功后返回 ok		



AT+GCON

AT+GCON: 查询 - 连接间隔		
查询命令:	响应	CONNECTION INTERVAL: Para
AT+GCON\CR\LF	说明	Para: 连接间隔 (范围: 8~1600, 单位 1.25毫秒)
举例: 发送指令 AT+GCON, 返回当前连接间隔: CONNECTIONINTERVAL:160 (默认为160)		

AT+SCON

AT+SCON: 设置 - 连接间隔		
设置命令:	响应	Ok
AT+SCON+Para\CR\LF	说明	Para:连接间隔 (范围: 8~1600, 单位 1.25毫秒)
举例: 设置连接间隔为 100ms。则发送设置命令: AT+SCON+80, 设置成功后返回: ok		

AT+GPWR

AT+GPWR: 查询 - 发射功率		
查询命令:	响应	Tx_power: Para dBm
AT+GPWR\CR\LF	说明	Para: - 23, - 6,0,4dBm 其中一个数
举例: 发送查询命令: AT+GPWR, 返回当前发送功率: Tx_power:-6dBm		

AT+SPWR

AT+SPWR: 设置 - 发射功率		
设置命令:	响应	Tx_power: Para dBm
AT+SPWR+Para\CR\LF	说明	Para:输入 -23,0,-6, 0,4 其中 1 个数 (注意: 4 dBm 只对cc2540 起作用, 对 cc2541 不起作用)
举例: 设置发射功率-6 dBm。则发送设置命令: AT+SPWR+-6, 设置成功后返回: Tx_power:-6dBm		



AT+GUAR

AT+GUAR: 设置 - 串口波特率		
设置命令:	响应	UART BAUDRATE:Para
AT+GUAR\CR\LF	说明	Para: 串口波特率为以下 5 个数中的一个。 (1): 9600 (2): 19200 (3): 38400 (4): 57600 (5): 115200
举例: 发送查询命令: AT+GUAR, 返回:UARTBAUDRATE:9600		

AT+SUAR

AT+SUAR: 设置 - 串口波特率		
设置命令:	响应	Ok
AT+SUAR+Para \CR\LF	说明	Para: 串口波特率为以下 5 个数中的一个。 (1): 9600 (2): 19200 (3): 38400 (4): 57600 (5): 115200
举例: 设置波特率为38400。则发送设置命令AT+SUAR+38400, 设置成功后返回ok。(注意: 设置了一个新波特率后, 串口调试助手的波特率也要做相应的调整, 本例中要调为38400)、		

AT+GLPM

AT+GLPM: 查询-低功耗模式		
设置命令:	响应	LOW POWER MODE: Para
AT+GLPM\CR\LF	说明	Para: 低功耗模式 ENABLE或DISABLE
举例: 发送查询命令: AT+GLPM, 返回: LOW POWER MODE:ENABLE		

AT+SLPM

AT+SLPM: 设置-低功耗模式		
设置命令:	响应	Ok
AT+SLPM+ Para	说明	Para: 0: 关闭低功耗式 1: 开启低功耗模式
举例: 设置开启低功耗模式, 发送: AT+SLPM+1		



AT+GPAM

AT+GPAM: 查询-配对模式		
设置命令:	响应	PAIR MODE: Para
AT+GLPM\CR\LF	说明	Para: 配对模式 ENABLE 或 DISABLE
举例: 发送查询命令: AT+GLPM, 返回: PAIRMODE:DISABLE		

AT+SPAM

AT+SPAM: 设置-配对模式		
设置命令:	响应	Ok
AT+SLPM+Para\CR\LF	说明	Para: 0: 关闭配对模式 1: 开启配对模式
举例: 设置开启配对模式, 发送: AT+SPAM+1		

AT+CONB

AT+CONB: 断开连接		
设置命令:	响应	CONNTION IS BROKEN
AT+CONB\CR\LF	说明	None
举例: 发送断开连接命令: AT+CONB, 返回: Connectionisbroken		

AT+RSET

AT+RSET: 模块复位		
复位命令:	响应	None
AT+RSET\CR\LF	说明	None
举例: 直接发送命令: AT+RSET, 就可以复位了		

4.3 AT指令测试

说明: 因为 F-9088 串口透传可以分为“低功耗”与“非低功耗”两种, 非低功耗模式下的 F-9088 串口透传上电就自动广播, 并可以直接使用串口透传。而低功耗模式下的 F-9088 串口透传则上电是不会自动广播, 而是要手动开启广播, 和手动开启串口透传功能。

以下讲解的设置例子都是基于非低功耗模式下的 F-9088 串口透传而讲解的, 低功耗模式下的使用跟非低功耗的 F-9088 串口透传的使用基本一样, 只需要加多以下两步。

- (1) 每次开机, 复位或休眠状态下都需要给 Module_CS (即 P0_0 管脚) 脚置低电平, 并



一直保持，而使BLE 模块开启广播。

(2) 每次开机，复位若想要开启串口接收模式则需将RTS（即P0_5 管脚）拉低， 并一直保持低电平直到要退出串口接收模式时才拉高。

（详细介绍见3.2.1、引脚逻辑关系和3.2.2 通信过程）

4.3.1 测试环境搭建

(1) 使用到的工具：

串口调试助手 sscom32（版本 1.0.0.1），利用百度直接搜索并下载“串口调 试助手 sscom32”；下载完后直接打开应用，不需要安装。

安卓系统手机：装有BLE 读写器等蓝牙测试软件，（BLE 读写器可以利用百度搜索“BLE 读写器”网上有很多的下载连接） 苹果系统工具：装有“LightBlue”等蓝牙测试软件, LightBlue 可以在苹果手机上的“APP Store”软件里下载。

步骤1： 打开苹果手机的“APP Store”软件。

步骤2： 点击搜索





步骤 3: 在搜索里输入Light Blue, 并点击搜索



步骤 4: 下载并安装 Light Blue



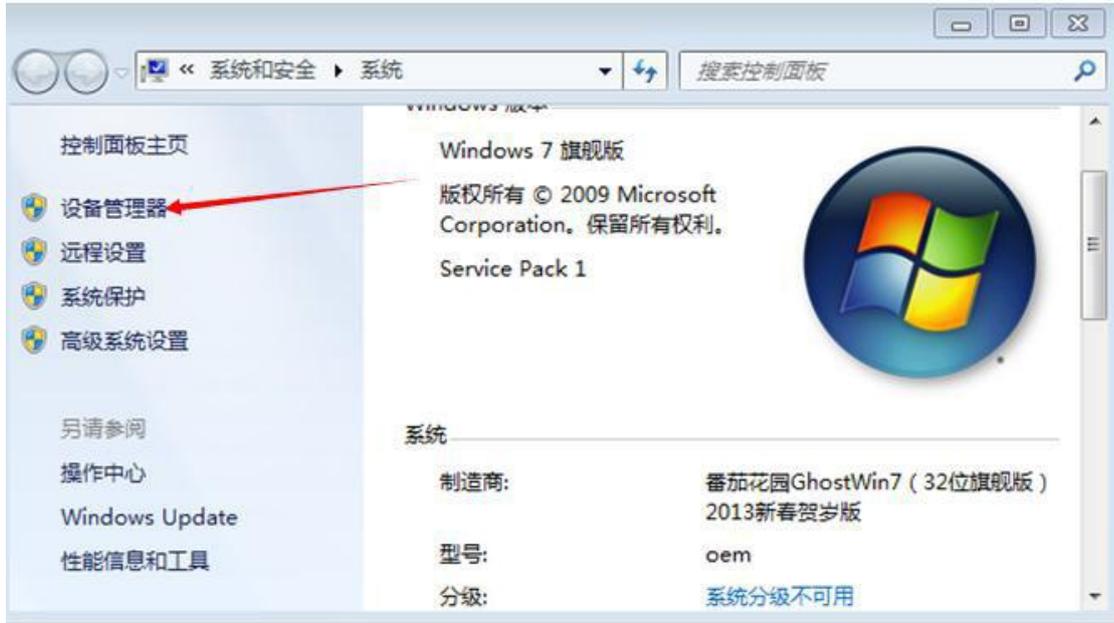


(2) 环境搭建

》连接好串口模块，F-9088 模块管脚图见上图

》查询串口号：

步骤1： win7 系统下，“计算机” 右击，选择“属性” 点击属性可以打开下面的界面。



步骤2： 点击“设备管理器” 进入下图所示的界面。

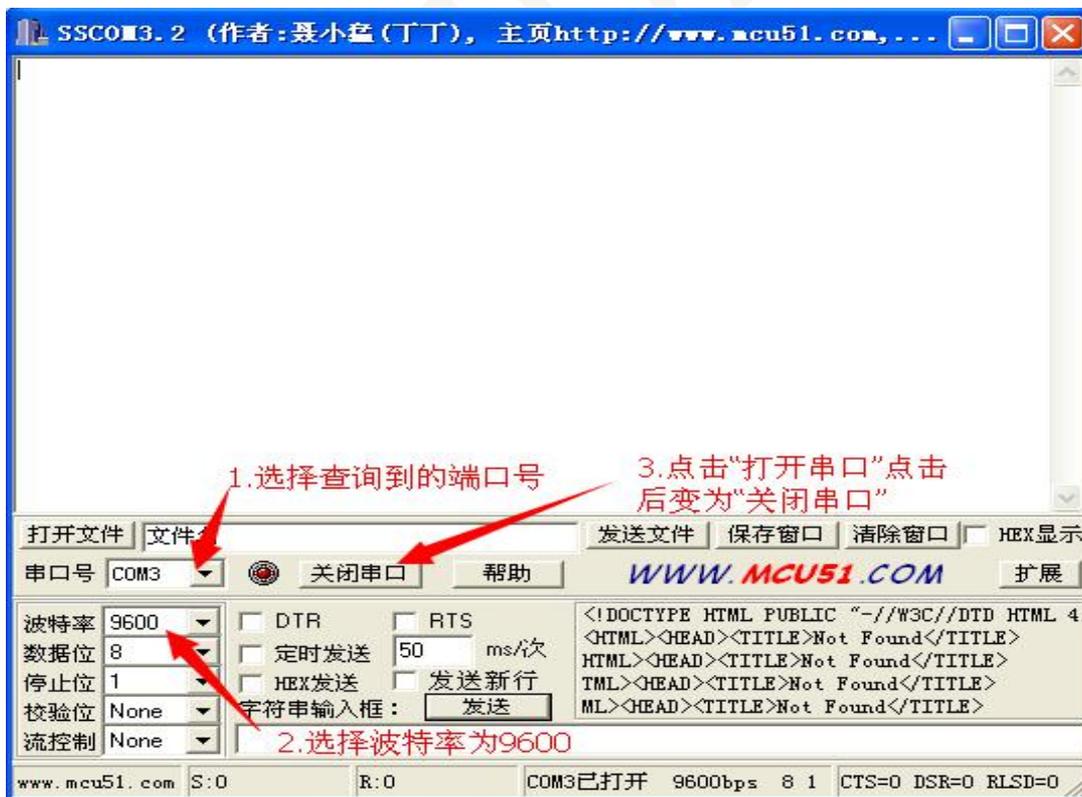




步骤 3: 点击“端口”选项，可以看到 CP210x USB to UART Bridge Controller (COM3)，COM3 就是端口号，注意：它是一个可变的数字，所以不同的设备 端口号不一样，所以要根据自己所以查到的为准，记下你所以查到的端口号。



步骤 4: 打开附件中的串口调试助手 sscom32，设置波特率为 9600（注意：F_9088 透传默认波特率为 9600，），在串口号选项下选择刚才你查到的端口号，并点击 打开串口 按键，至此环境搭建完成，串口调试助手环境设置如下图所示。



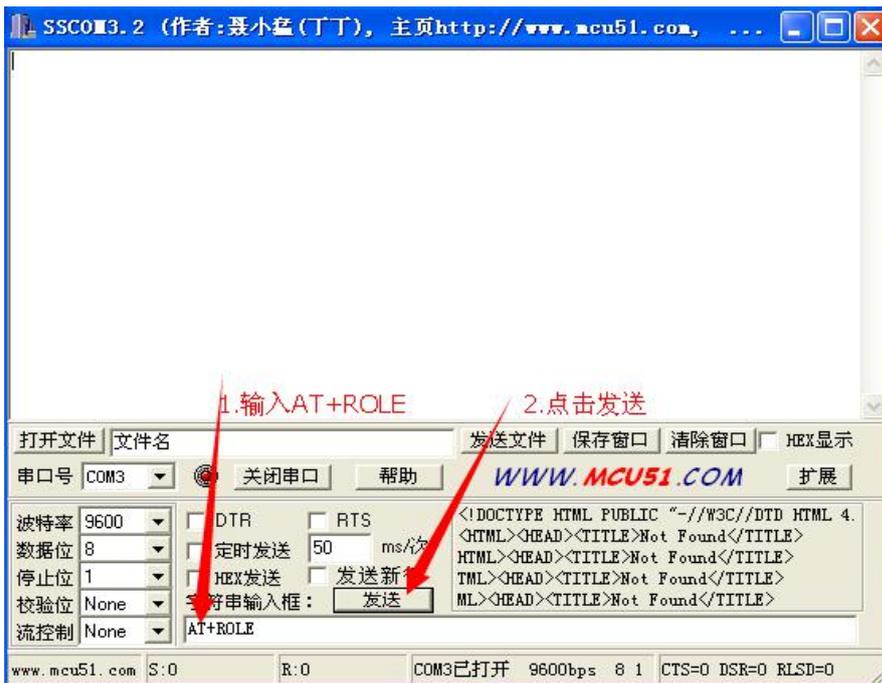


4.3.2 查询指令的测试

因为各种查询指令和基础指令的使用方法与步骤基本一样，所以在此仅以讲解查询模块主从模式指令的使用方法为例，其余的查询指令和基础指令的使用方法参考查询模块主从模式指令的使用。

(1) 询 - 模块主从模式：指令：AT+ROLE

步骤1：在串口调试助手的“字符串输入框”中输入查询模块主从模式的AT指令：AT+ROLE，并点击“发送”。



步骤2：点击完“发送”按键后，可以看到串口调试助手显示当前模块的工作模式：ROLE:SLAVE，则该指令功能正确。





4.3.3 设置指令的测试

因为各种设置指令的使用方法与步骤基本一样，所以在此仅讲解设置 - 连接间隔指令的使用方法，其余的设置指令的使用方法参考设置 - 连接间隔指令的使用。（注意：设置波特率时，当完成设置波特率后，需要在串口调试助手的波特率选项中重新选择新的波特率。）

- (1) 设置连接间隔，指令：AT+SCON+ Para 说明：Para:为要设置的连接间隔 在此，举例设置模块的连接间隔为 100ms

步骤 1: 在串口调试助手的“字符串输入框”中输入 AT+SCON+80，并点击“发送”。



步骤 2: 点击完“发送”按键后，如果串口调试助手显示 OK，再复位 9088 模块。



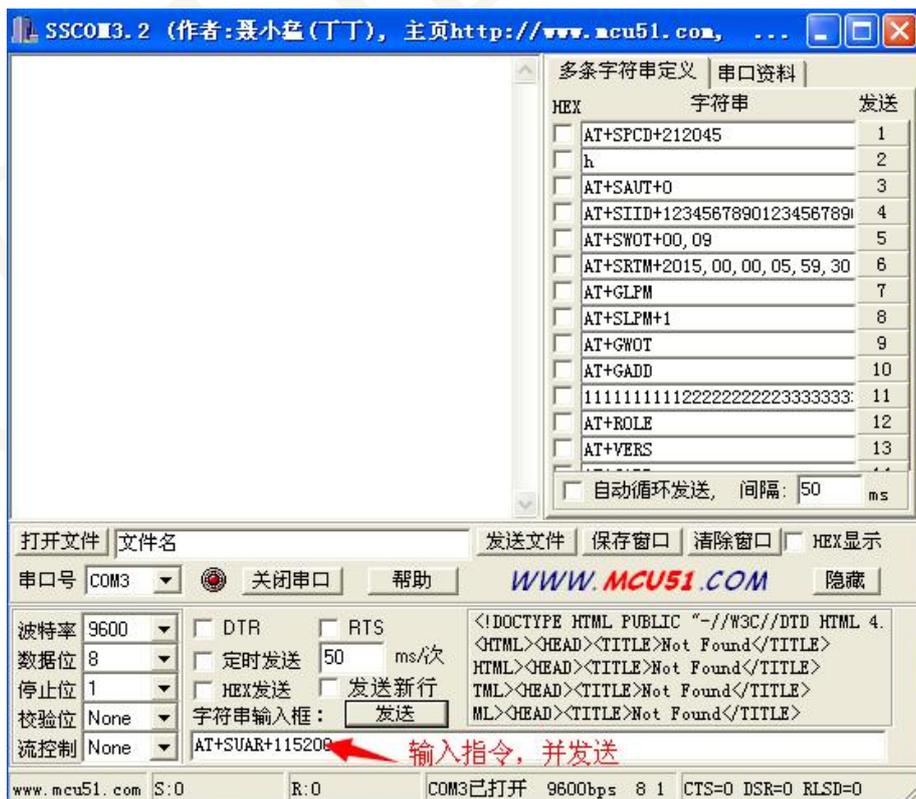


步骤 3: 复位 9088 模块后，再查询 F-9088 模块的连接间隔（查询方法见 4.3.2），本次的查询结果见下图所示，由下图可知，连接间隔成功设置为 80(单位： 1.25ms)，即 100ms。



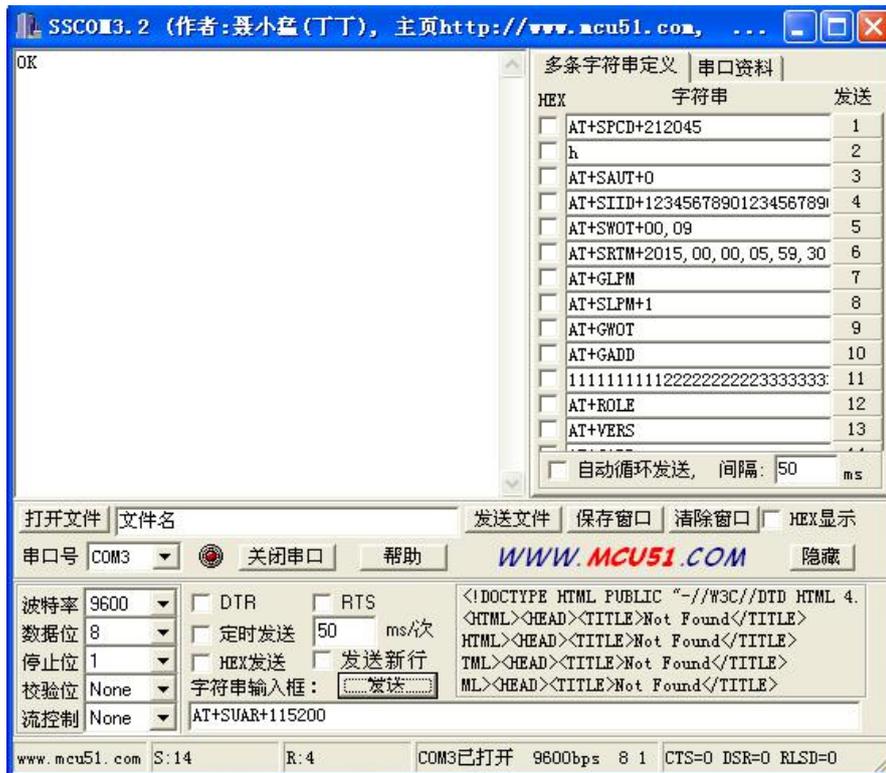
(2) 设置波特率为，指令：AT+SUAR+ Para 说明：Para:为要设置的波特率 在此，举例设置模块的波特率为 115200

步骤 1: 在串口调试助手的“字符串输入框”中输入 AT+SUAR+115200，并点击“发送”。

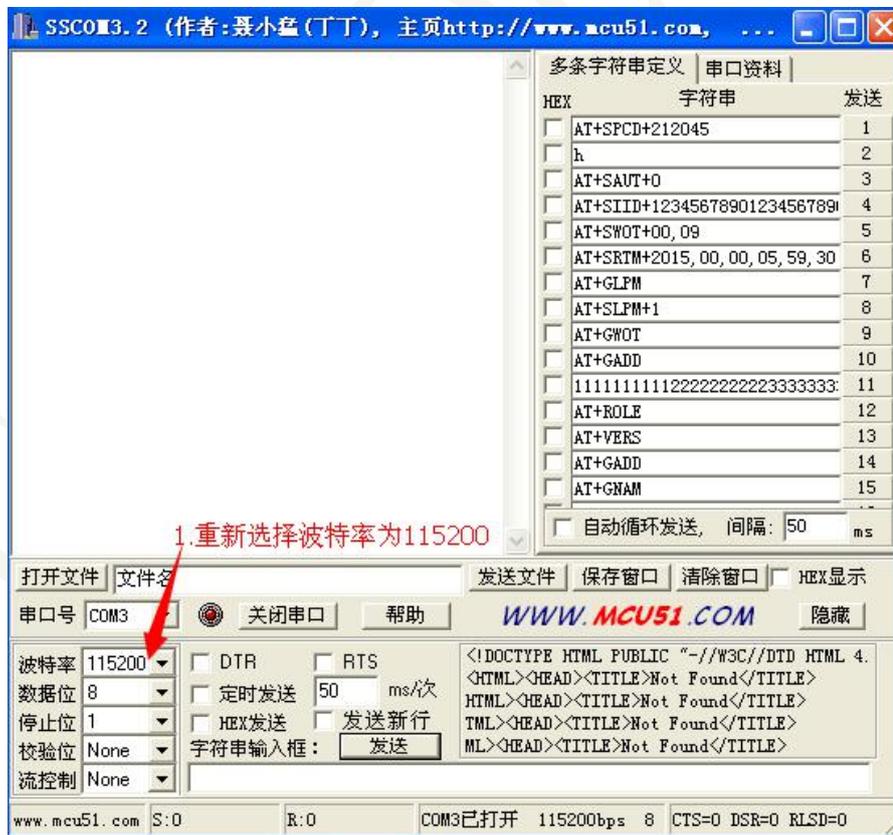




步骤 2: 点击完“发送”按键后, 如果串口调试助手显示 OK, 再复位 9088 模块。



步骤 3: 复位 9088 模块后, 重新选择串口助手波特率为 115200。



步骤 4: 说明: 此后的举例, 都是基于波特率为 115200.进行的。



步骤 2: 后点击确定后, 如果手机屏幕先显示 Connect Success, 再显示 Service DiscoverySuccess, 则连接上了蓝牙。



步骤 3: 连接上蓝牙后, 点击 Beacon 按键, 打开服务。





步骤 4: 点击 Beacon 按键，打开服务后会显示下图所示界面。再点击 Passconde 按键，



步骤 5: 点击 Passconde 按键后会出现下图所示界面。再点击“开始通知按键”。

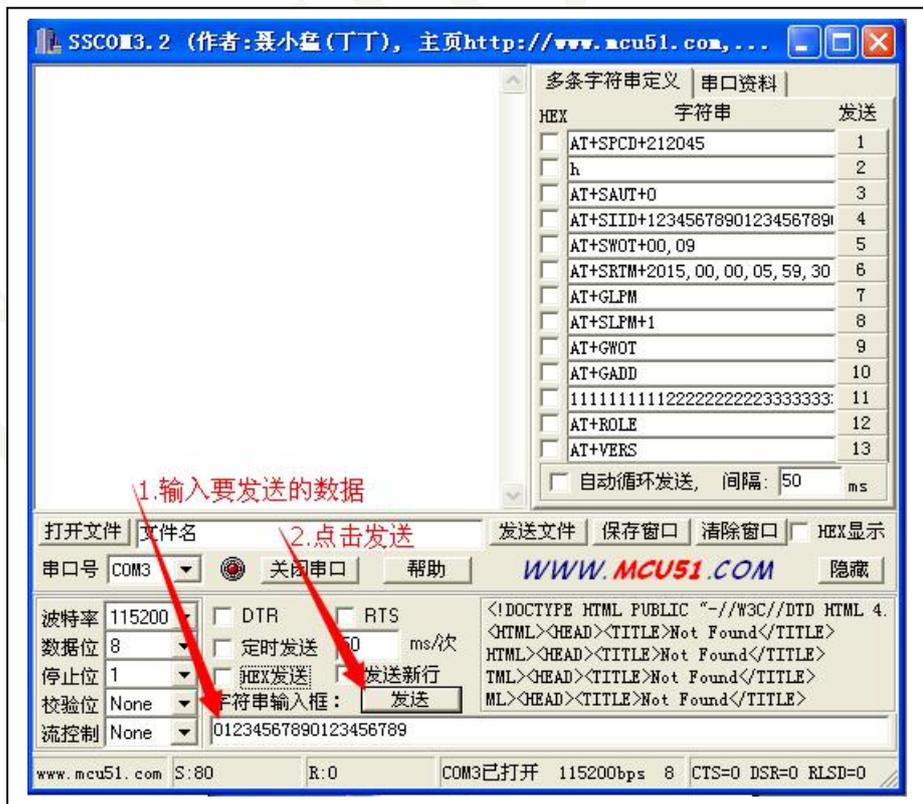




步骤 6: 点击“开始通知按钮”后, 会出现下图所示界面, 即开启了通知。



步骤 7: 在串口调试助手的“字符串输入框”中输入 20 字节的透传数据, 在本例中输入了数据为: “01234567890123456789” 的 20 字节的数据, 并点击“发送”。





步骤 8: 点击发送后, 在安卓手机上可以看到透传过来的数据。



(2) 基于苹果系统的透传测试

步骤 1: 首先, 打开苹果手机的“Light Blue”软件, 在 Light Blue 中找到你的蓝牙设备 (默认名称为: CC2541_SPP), 并点击连接, 连接时若需要输入连接密码, 默认密码为 123456, 输入密码后点击确定。





步骤 2: 点击确定后, 如果连接成功会进入下图所示界面。:



步骤 3: 在当前界面下, 往下拉, 拉到最低端, 可以看到一下界面, 并点击 UUID 为 FFF4 的选项。

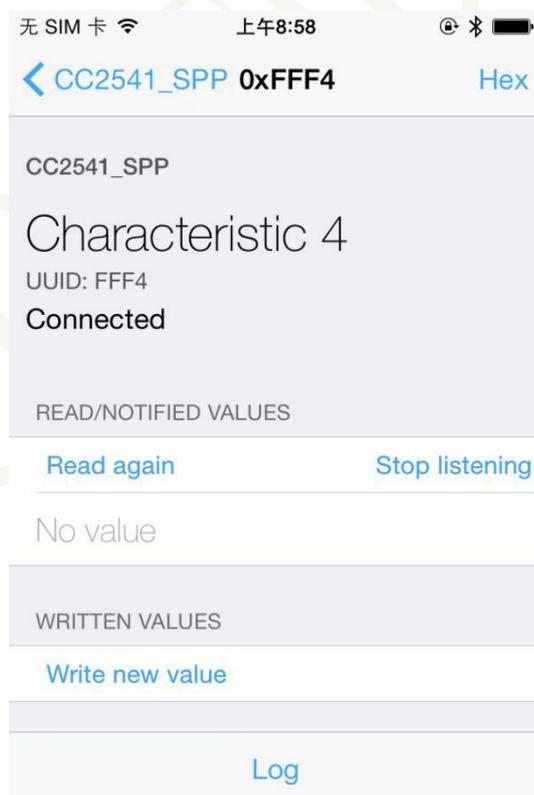




步骤 4: 点击Characteristic4按键后进入下图所示界面，再点击“Listenfor notifications”按键，开启通知。

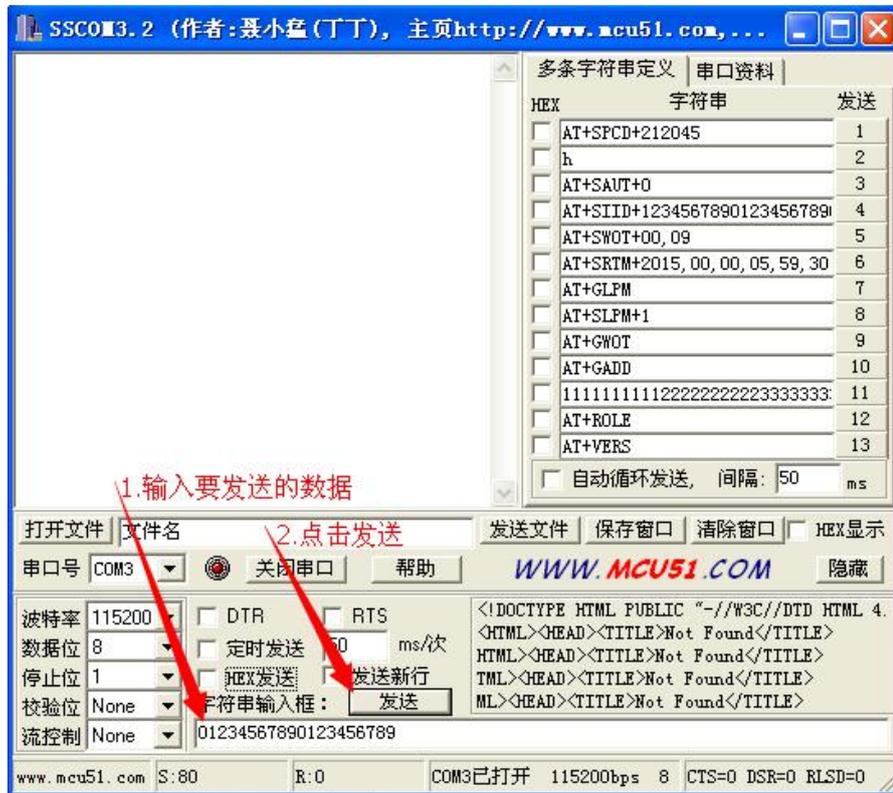


步骤 5: 点击“开始通知按键”后，会出现下图所示界面，即开启了通知。





步骤 6: 在串口调试助手的“字符串输入框”中输入 20 字节的透传数据，在本例中输入了数据为：“01234567890123456789”的 20 字节的数据，并点击“发送”



步骤 7: 点击发送后，在苹果手机上可以看到透传过来的数据。



步骤 8: 点击 Log 按键，查看所有传输过来的数据。



5.2 蓝牙到串口的透传测试

(1) 基于安卓系统的透传测试

步骤1: 打开串口调试助手，根据4.3.1的方法设置串口调试助手，再选择“HEX 显示”



步骤2: 重复 4.2.13 中的安卓系统下的透传测试步骤 1 到步骤 4 的操作，操作后 进入下图所示的界面，再点击写入按键。





步骤 3: 点击写入按键后进入下图所示界面，



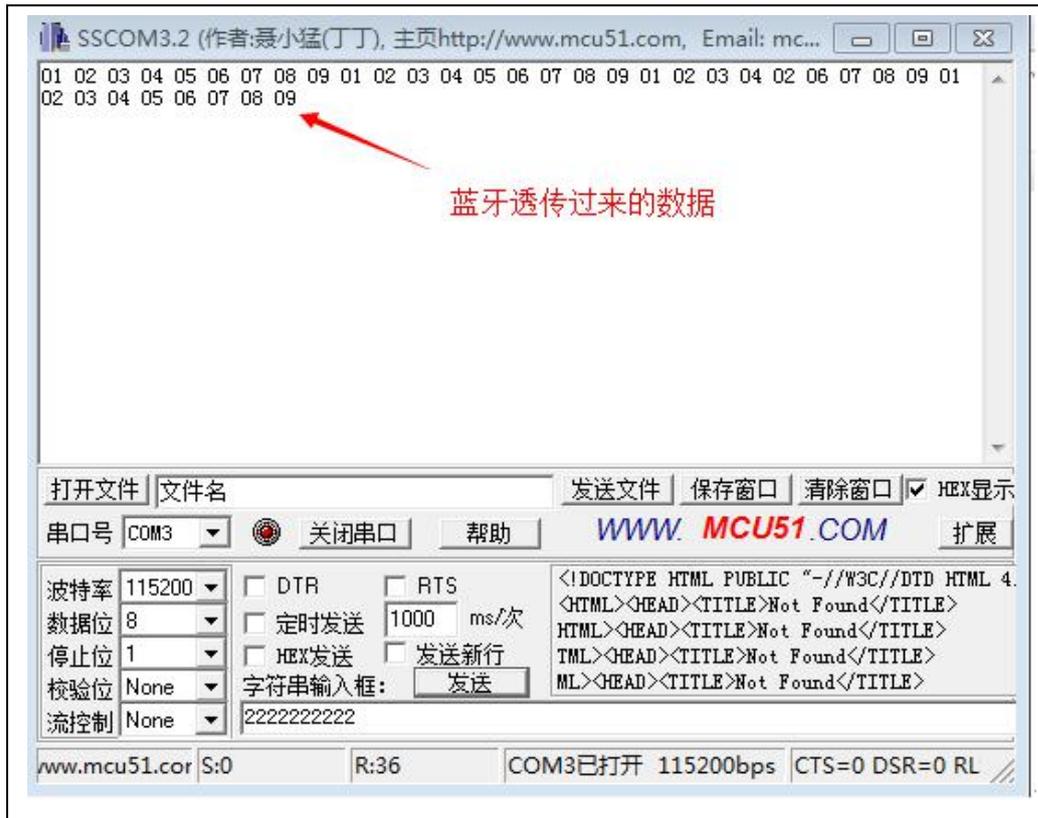
步骤 4 在“十六进制: 0x”下写入要透传的数据（注意：写入的数据必须为十六进制），在本例中写入了以下数据：

“010203040506070809010203040506070809010203040506070809010203040506070809”，写完数据后点击确定，就被发送出去了。





步骤 5: 点击确定后, 在串口调试助手中可以看到刚才发送过来的数据。



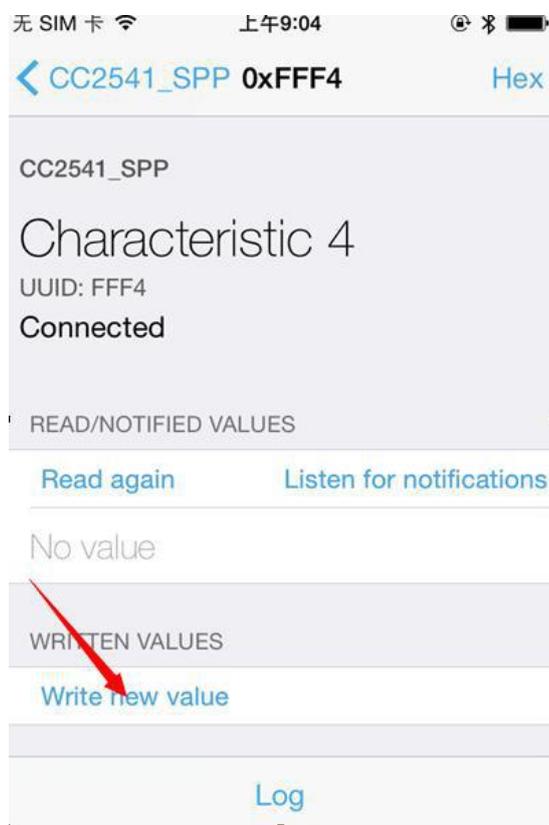
(2) 基于苹果系统的透传测试

步骤 1: 打开串口调试助手, 根据 4.3.1 的方法设置串口调试助手, 再选择“HEX 显示”。

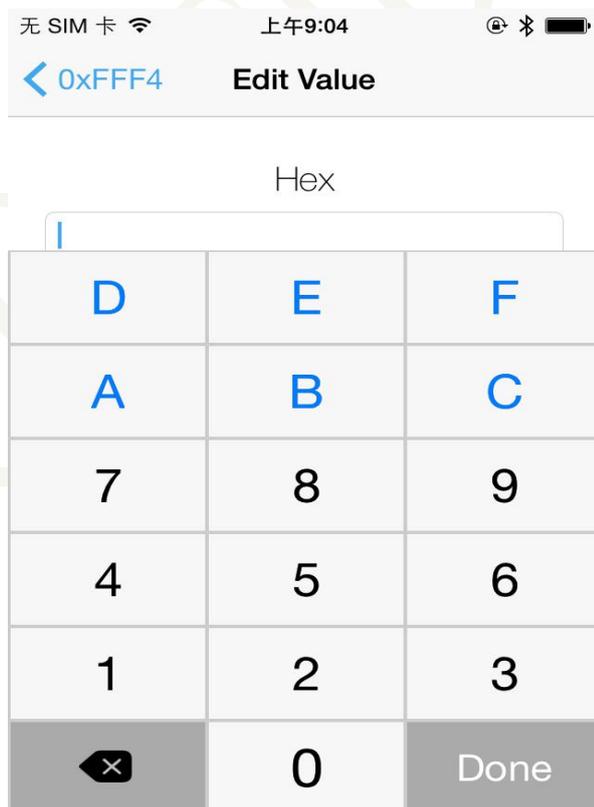




步骤 2: 重复 4.2.13 中的苹果系统下的透传测试步骤 1 到步骤 3 的操作, 操作后 进入下图所示的界面, 再点击 Write new value 按钮。

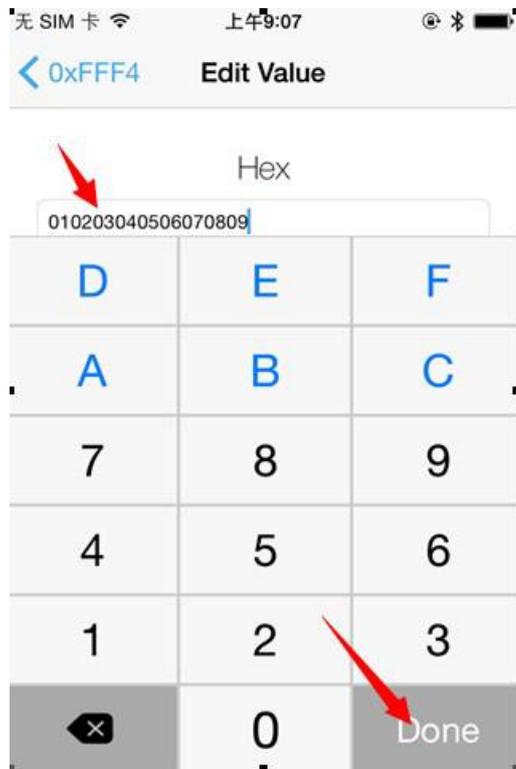


步骤 3: 点击 Write new value 按钮后进入下图所示界面,





步骤 4: 输入要传输的十六进制透传数据, 在这里输入了“010203040506070809”, 点击 Done



步骤 5: 点击 Done 后, 再串口调试助手里可以看到透传过来的数据。





六、PP和MCU编程参考

6.1 IOS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。本模块connectionhandle默认为0，通过UUID进行通讯。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 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。

备注：IOS 的最佳测试BLE 软件是LightBlue,可在网上下载到源码。



6.2 安卓编程参考

安卓4.4 系统才能完全和蓝牙4.0 模块进行透传。connectionhandle 默认为0，通过UUID 进行通讯。下载安卓官网上的BLEdemo.apk，即可与F-9088 蓝牙模块进行串口透传。

6.3 IOS、安卓、MCU编写者需要的参数知识

连接间隔：connInterval，1.25ms 的倍数，最小值为6(即 7.5ms)，最大值 3200(即 4.0s)。监督超时：supervisionTimeout，10ms 的倍数，最小值 10 (即100ms)，最大值3200 (即 32.0s)。必须大于： $(1+slaveLatency)*connInterval$

从机潜伏：slaveLatency，最小值 0，最大值 499。必须小于：

$(supervisionTimeout/connInterval)-1$ 不同连接参数的特点：两设备都会以高功耗运行

高数据吞吐量 发送等待时间短 长连接间隔 两设备都会以低能耗运行 低数据吞吐量 发送等待时间长低或者0 潜伏值：从设备以高能耗运行从设备可以快速收到来自master 设备的数

高潜伏值 从设备在没有数据发送的情况下可以低能耗运行 从设备无法及时收到来自主设备的数据 主设备能及时收到来自从设备的数据短监督超时 当信号弱或者信号不稳定时，可以

及时“觉察”到连接断开 长监督超时当信号不稳定丢包时，若在监督时间内又重新收到数据包时，认为连接没有断开 说明和建议：连接间隔可以简单理解为，是两个连接着的蓝牙设备

发送“心跳包”的时间间隔。蓝牙设备认定它们之间的连接是否断开，就是看心跳包是否

及时到达。举个例子，比如设置 connInterval=100ms, slaveLatency=1, supervisionTimeout=1s。

connInterval=100ms，是指蓝牙主机每100ms 发一次心跳包给从机，从机收到后回复一次。

slaveLatency=1，是指如果从机没有数据发送时，可以跳过一次心跳包的回复，让自己省

电。supervisionTimeout=1s，对从机来说，当它发现连续1 秒钟都没有收到心跳包，就认为连接断开。对主机说，当它连续发了11 个心跳包，都没有得到回复，认为连接断开。

根据 BLE4.0 协议规定：master 设备可以随时发送一个连接更新请求到slave，来改变

连接参数。在链路层，连接参数的更新总是被master 发起，但是L2CAP 层允许slave 向

master 发送一个连接参数更新请求。BLE 协议允许应用层根据实际需要动态的调整连接

参数，这个会产生相应的功耗及数据吞吐量。当两蓝牙设备每次创建连接时，这三个连

接参数都由 是主机给定的。比如iPhone4S 和iPhone5，设定的连接参数都是：24, 0, 72。

转换一下：

$connInterval=24*1.25ms=30ms;$



slaveLatency=0;

supervisionTimeout=72*10ms=720ms;

我们看到iPhone 的连接间隔设置的比较短，所以数据吞吐量大，但是能耗比较大，大概平均电流达到900uA~1000uA 左右，监督超时720ms，快速监测到连接断开。另外，三星galaxy S3 设定的连接参数值为54, 0, 42。根据经验，从机潜伏设定值一般要低一点或者为0，监督超时一般也不宜太长，连接间隔可以根据不同应用需要而设定。数据交换少，对功耗敏感的应用，连接间隔可以设置长一点。总而言之，对于BLE 连接参数的设定，可以多做试验，得到一组在数据吞吐量和功耗方面都比较满意的值。另外，当该模块与 iOS 设备连接时， Apple 公司规定，iOS 设备的蓝牙附件的连接间隔参数，除了要符合 Sig 组的规定外，要必须符合Apple 的规定：

$\text{connInterval} * (\text{slaveLatency} + 1) \leq 2 \text{seconds}$ $\text{connInterval} \geq 20 \text{ms}$

$\text{slaveLatency} \leq 4$ $\text{supervisionTimeout} \leq 6 \text{seconds}$

$\text{connInterval} * (\text{slaveLatency} + 1) * 3 < \text{supervisionTimeout}$.

七、客户定制方案规则约定与参考电路