NFC2COM 用户手册
V1.4

文档修改历史

版本	描述	日期
V1.0	基础版本	
V1.1	增加手机检测功能	2016.8.9
V1.2	管脚变换位置	2017.4.17
V1.3	增加 Q&A	2021.12.22
V1.4	增加 IQR 时延控制 P2.0 控制接口 固件版本号： 5	2022.8.19

样片购买：
https://item.taobao.com/item.htm?&id=530432859170

企业批量采购请联系我们，会有大量优惠

第 1 章 功能介绍

NFC2COM 是风火轮科技推出的一款 NFC 模组，主要是实现 NFC 手机与外部设备之间进行 NFC 通信功能，外部设备通过串口连接 NFC2COM 模组，可以非常快速添加 NFC 功能，本模组就是设备与 NFC 手机之间的一条通信渠道，对设备这一端来说只是串口通信，要发送或接受数据直接操作串口就行，不需要关注 NFC 通信协议栈，而对手机端来说，外部设备就是一个 TAG，Type A 的标准标签，操作起来非常容易和简便，兼容性更好。

NFC2COM 通过动态模拟卡方式实现与手机快速交互数据，可以交换任意长度的数据，只要手机不离开天线区域。

NFC2COM 模组通过静态卡模拟方式，可以模拟成 NFC Forum 定义的标准功能卡：蓝牙配对卡，TEXT 文本卡，网址标签卡。

- **数据传输：**
  外部主控用串口可以通过 NFC2COM 与手机实现任意长度的数据通信，
  目前速率:1000byte/s

- **卡模拟：**外部主控（PC/单片机）可以通过串口设置 NFC2COM 模拟成标准论坛卡：
  蓝牙配对卡
  TEXT 文本卡
  网址标签卡
  更多类型需要委托风火轮团队定制开发（NFCteam@smartfire.cn）
- **板载：** ◎1 个蜂鸣器驱动电路（可以外接蜂鸣器并通过串口命令控制它的开关）
  ◎1 个 LED 驱动电路（可以外接 LED 灯并通过串口命令控制它的开关）
  ◎GPIO:4 个（P0.6/P0.5/P0.4/P3.1），可以通过串口命令控制和读取
  ◎IRQ 中断口:当模组要输出串口数据前 1 毫秒，它会输出高电平，平时为低可用于唤醒外部主控。

- **波特率：**
  NFC2COM 可以支持 UART 串口 TTL 电平如下波特率：
  115200 8 N 1
  9600 8 N 1
  4800 8 N 1
  可以用串口命令设置

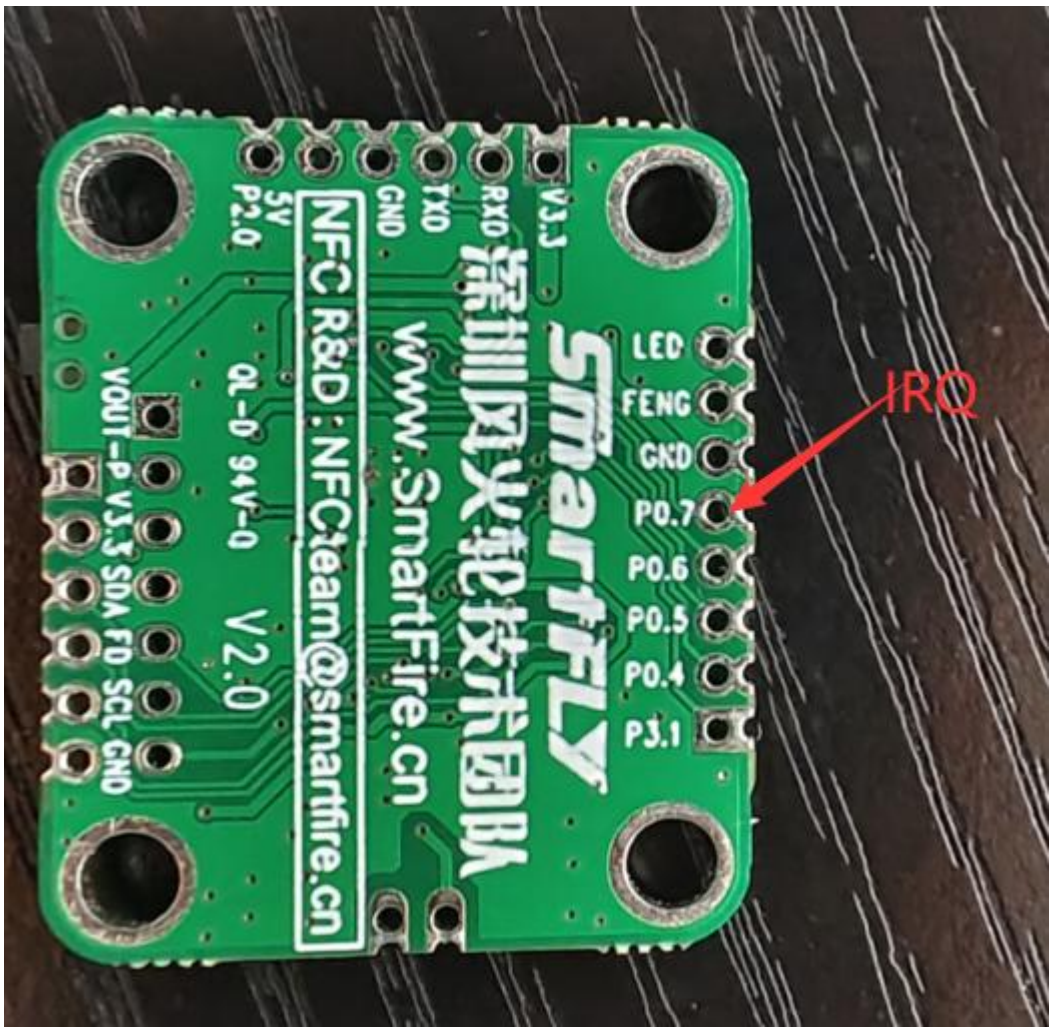
- **超低功耗：**
  工作功耗：2mA(3.3V)
  待机功耗：4uA
- **工作温度：** -40 ～ 85 ℃

- **中断 IRQ：**

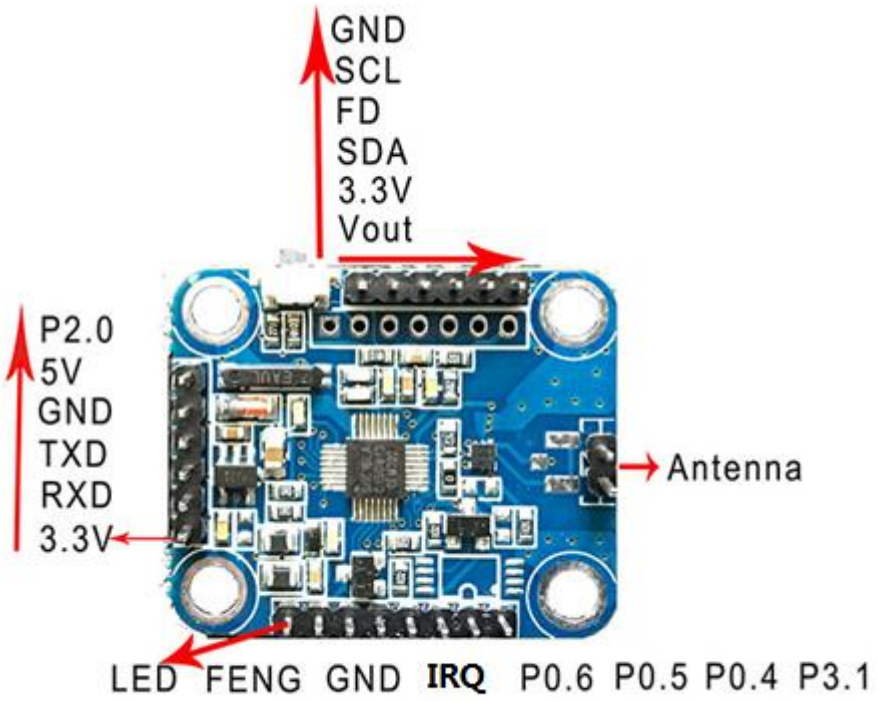
当串口有数据要发出来，会先通过 IRQ 管脚输出一个高电平，平时为低电平

外部 MCU 可以接 IRQ 管脚作为中断，监听上升沿作为信号，当得到中断就唤醒并接收模组传过来的串口数据

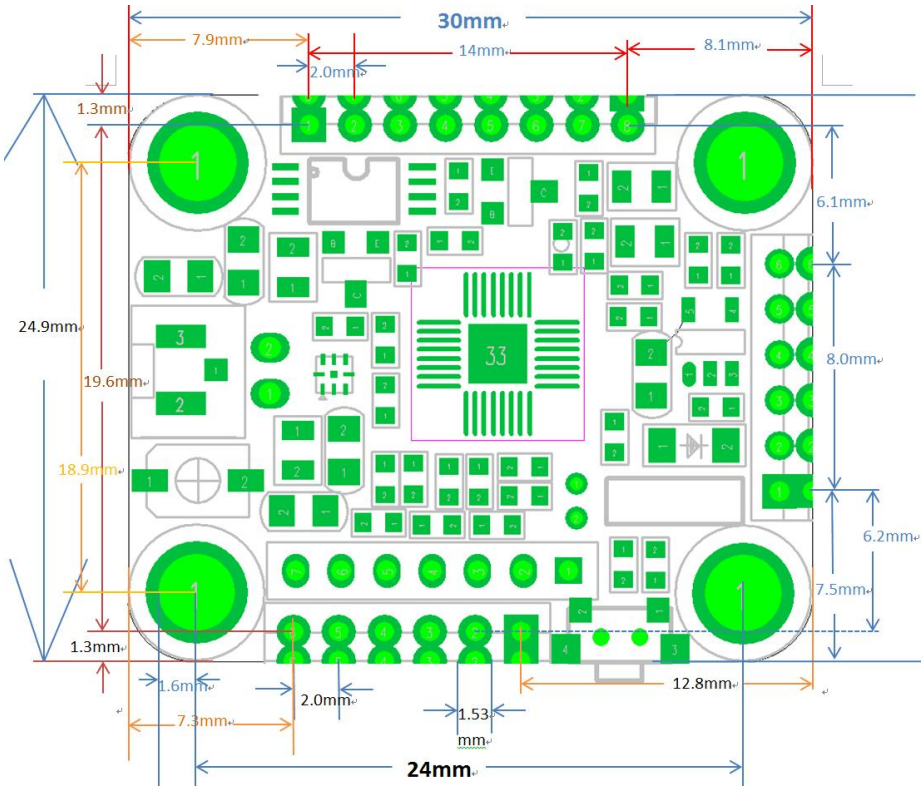
这个时间可以通过命令进行修改，默认是 1ms，可以加大，详细看 4.9 章



第 2 章 板载硬件介绍



尺寸图:



板型尺寸：30mm x 24mm

第 3 章 快速开始教程

本章是指导您在 PC 环境下，用标准串口与 NFC2COM 模组进行互动，实现 NFC 功能，通过这样的环节，让你了解到这个模组的工作原理，知道怎么去控制模组来实现你想要的功能，进而在不同平台下，操作串口去与 NFC2COM 通信，指挥它去实现你想要的功能。

因为 NFC2COM 模组是能过串口发送指令与之互动，所以真正实现跨平台，不所谓你的主机是什么系统什么芯片，只要是用串口通信，就能 NFC 通信。

3.1 测试平台搭建

PC windows 系统

1、 安装串口调试助手，串口调试助手请选用你熟悉的软件,在我们提供的资料包里，也有一个串口调试工作，您可以选用，

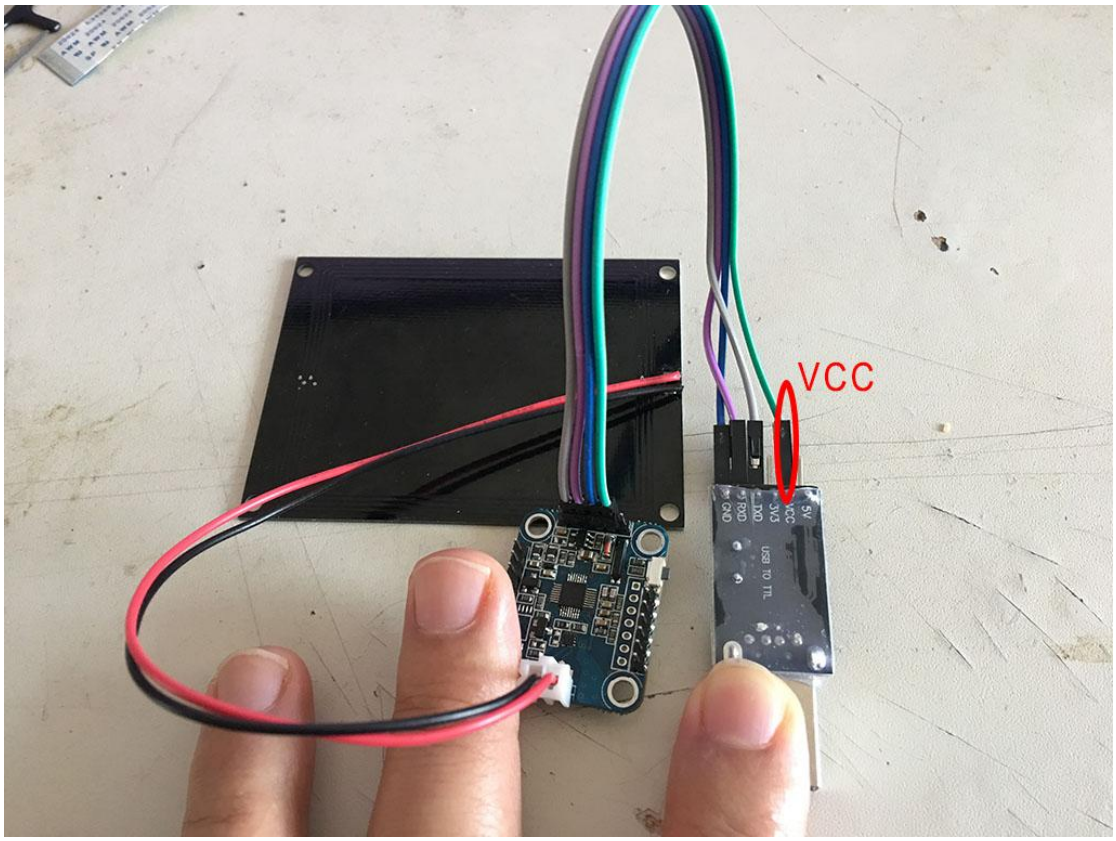
2、 安装风火轮提供的 NFC 操作工作软件，该软件专门为 NFC2COM 模组开发，可以实现和测试所有 NFC2COM 的功能，这也是一个示例软件，风火轮开放源代码，可以进行二次开发，也可以委托风火轮定制开发（需要交研发费）。

3、

串口调试助手截图

硬件连接

PC+USB 转串口 --→ NFC2COM



接线图如上

接线关系如下图

NFC 模组　　USB 转串口模块

GND → GND

RXD → RXD

TXD → TXD

5V → VCC

风火轮 PC 端测试软件

（待补充）

3.2　第一次使用：

用户合到我们的模组，可以先按这个步骤验证一下硬件是否工作正常。

测试需要：

1,NFC 功能手机（Android 系统）　一部

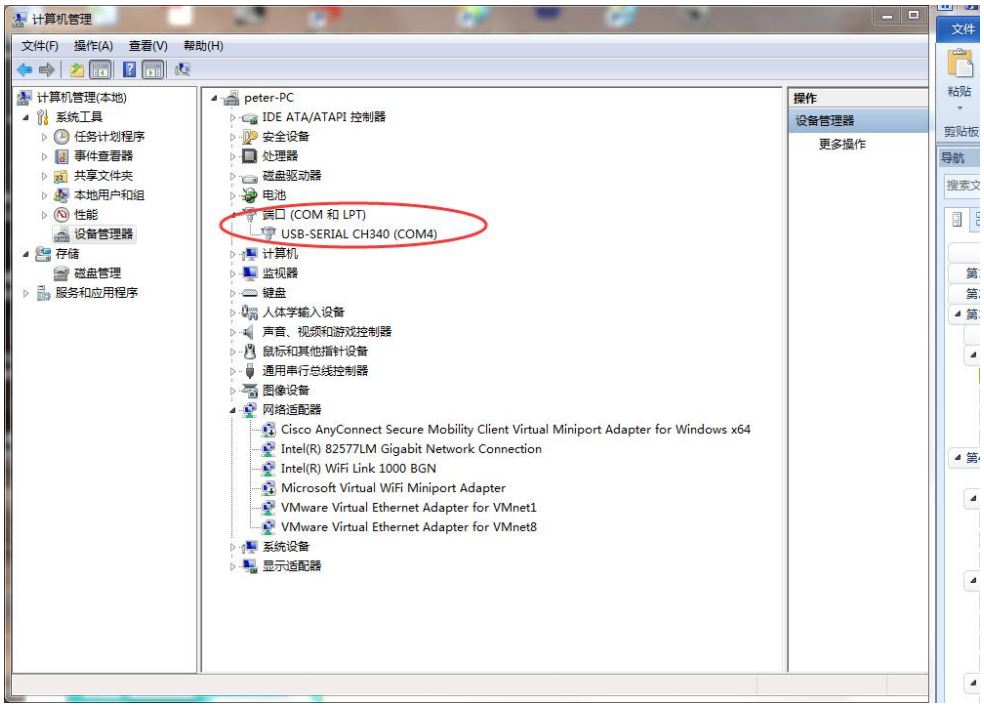
2，NFC2COM 模组 + 天线　　一套

3，USB 转串口（TTL 电平）　　一块

4，PC 电脑装 windows 系统　　一台

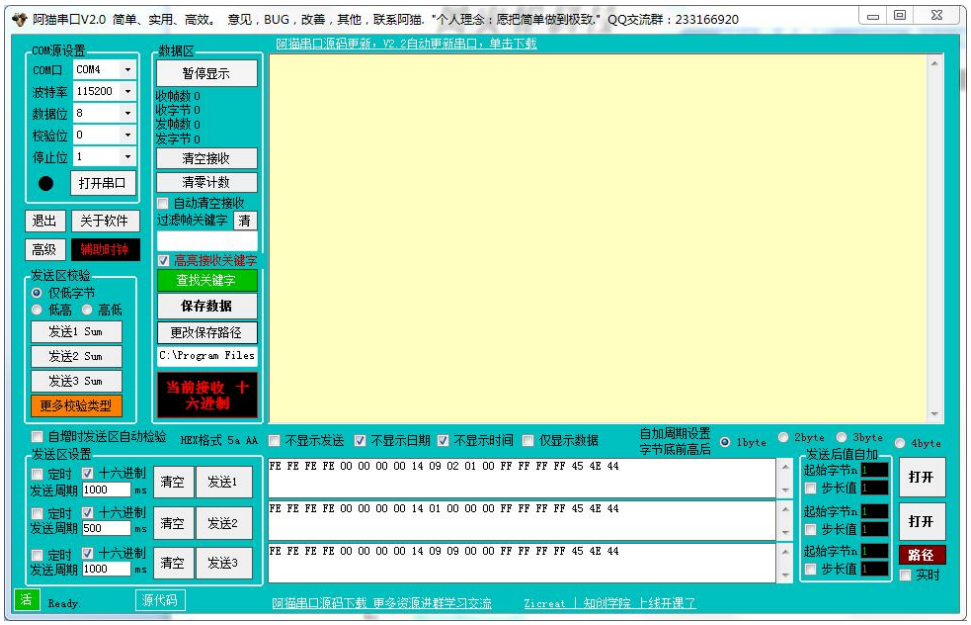
3.2.1 连接硬件

请确认你的串口是能正常工作的，电脑插上 USB 转串口模块，会看到该模块灯亮起，然后在设备管理器中看到生成相应的串口



如果发现没有，则是需要手动安装 USB 驱动，请百度“CH340 驱动”找到对应你的系统版本的驱动安装，直到出现上图中串口才算正常。

然后，打开串口调试助手，设置串口参数 115200 8 N 1，　　16 进制显示



效果如上图，记得选中对应你 PC 的 COM 口，点“打开串口”

NFC2COM 模组与 USB 转串口板的接线如 3.1 所示

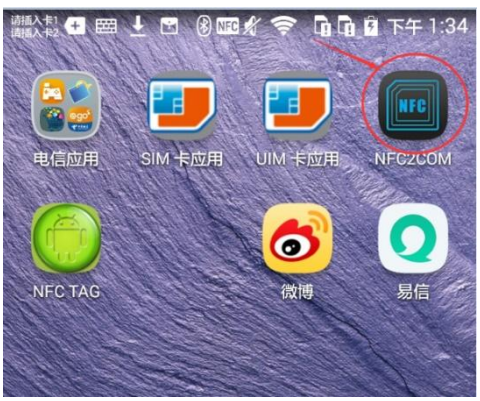
此时按模组上的重启按钮，就能看到 PC 串口调试助手上打印一些数据

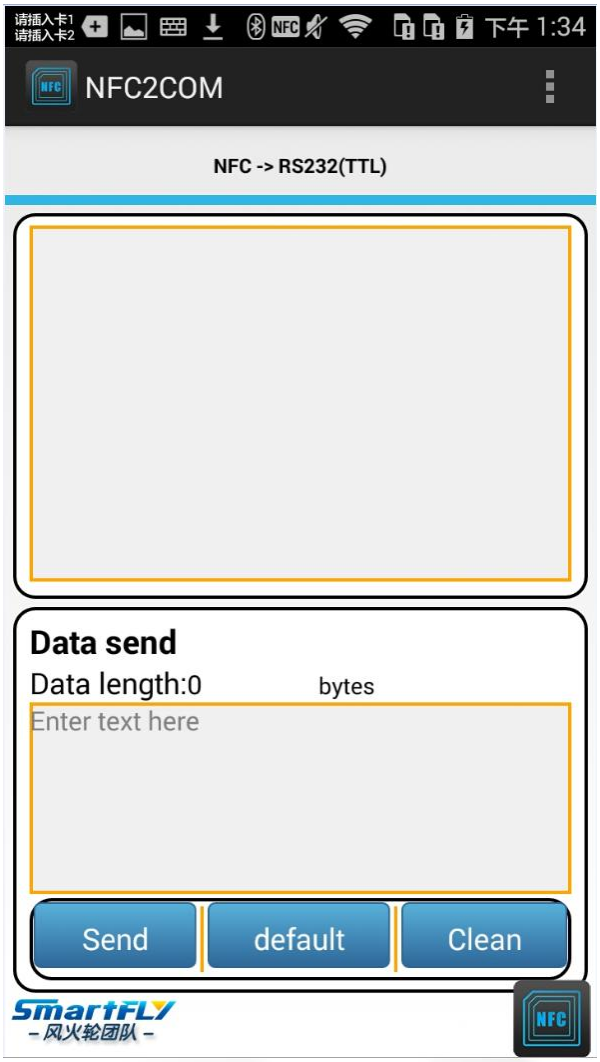
3.2.2 发送查询固件版本的命令，测试硬件是否工作正常

发送查询固件版本的命令，测试硬件是否工作正常。
发送：fe fe fe fe 00 00 00 00 14 01 00 00 00 FF FF FF FF 45 4E 44
如果返回：FE FE FE FE 00 00 00 00 14 06 04 00 FF FF FF FF 45 4E 44 这样的数据，证明是工作正常了。(04 这个是版本号 可能会因为不同的版本而变化)

3.2.3 NFC 手机

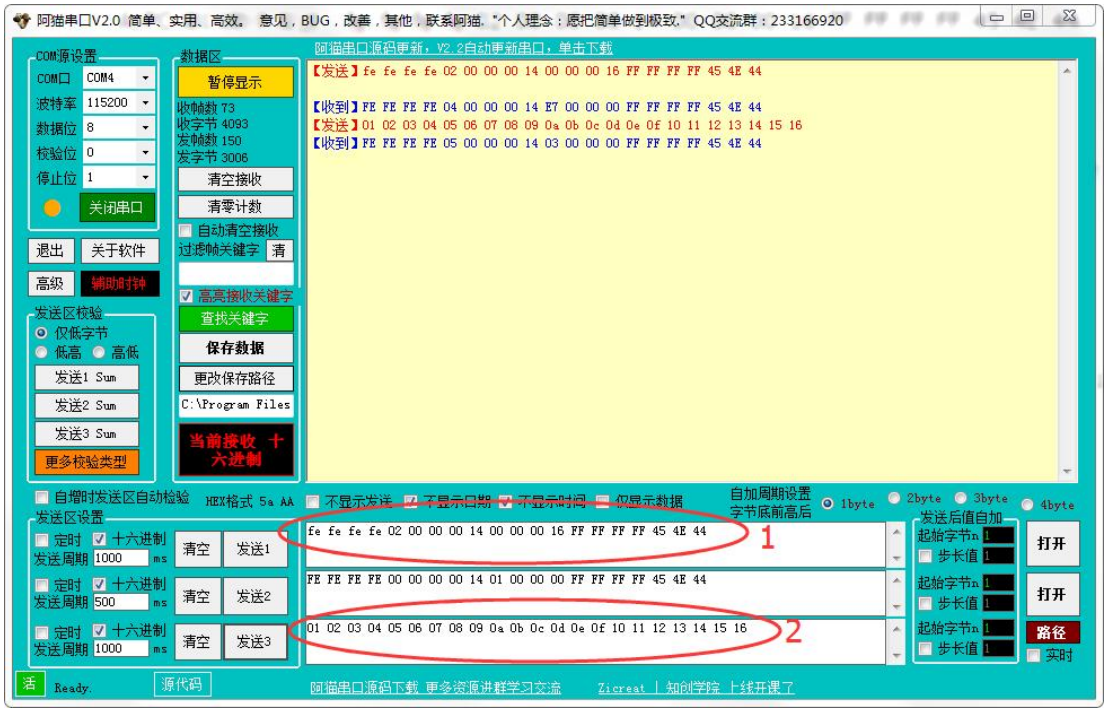
有 NFC 功能的手机安装测试 APK,然后打开 APK,把手机放到天线上。就能看到手机正在发送数据的提示，然后看到串口助手收到一堆数据，这就是手机 APK 默认发送的测试数据。




请插入卡1
请插入卡2
下午 1:34
NFC2COM
NFC -> RS232(TTL)
Data send
Data length:0
bytes
Enter text here
Send
default
Clean
SmartFLY
-风火轮团队-

3.2.4 此时手机不要挪开天线区，继续可以测试模组往手机发数据，

A, PC 串口助手发送：fe fe fe fe 02 00 00 00 14 00 00 00 16 FF FF FF FF 45 4E 44



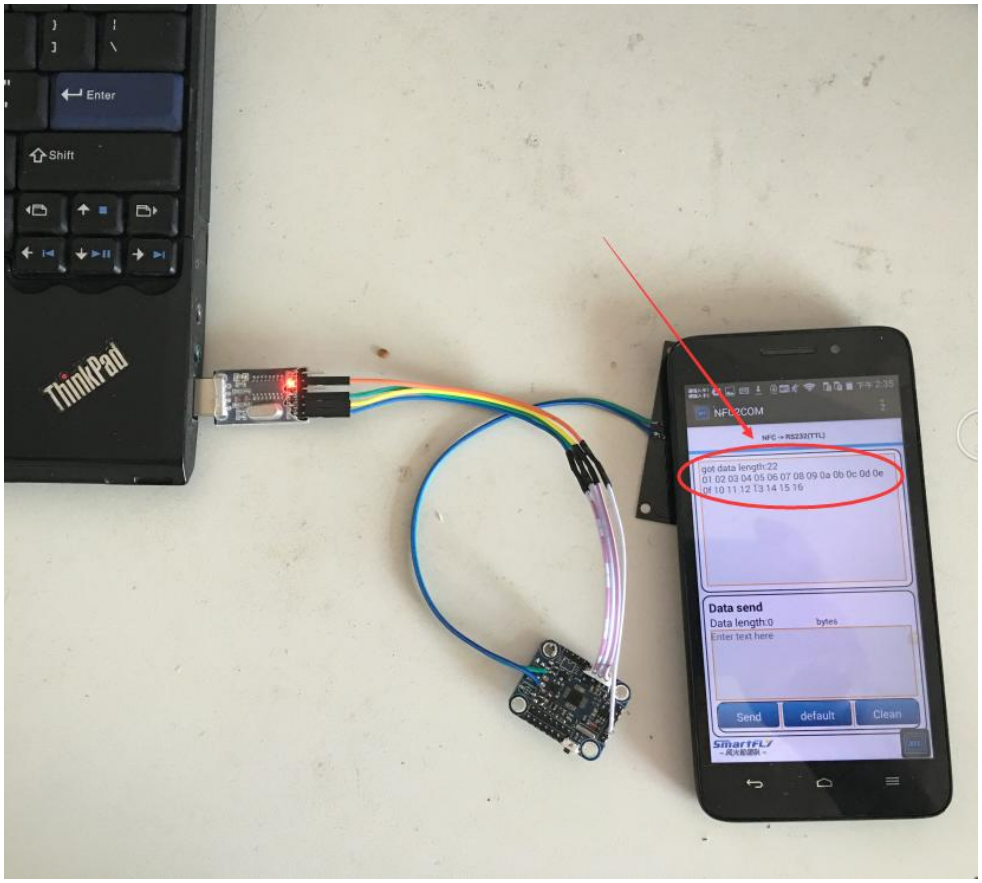
如果收到的是：FE FE FE FE 04 00 00 00 14 e7 00 00 FF FF FF FF 45 4E 44 表示当前可以发送，可以继续下一步

如果收到的是：FE FE FE FE 04 00 00 00 14 e9 00 00 FF FF FF FF 45 4E 44 表示当前设备忙，重新发送上一步的请求。

B 上面成功后:

发送数据：01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 （这串数据长度是前面的请求帧里的长度决定的）

这时手机上就能看到发过来的数据了。



这个测试，是串口往手机发送数据的演示。
实际使用中，用户可以修改成自己长度与数据。

3.2.5 大数据交互

大数据发送测试，此时手机也不要去动它，而是点击手机 APK 上的 default 按钮，再点击 send，就能看到手机在往模组发送 4K 的数据，模组这边是当传到 2K 的时候，就有一堆数据过来，到 4K 的时候，再有一堆数据过来，这些数据要拼接起来。

第 4 章 API 接口说明

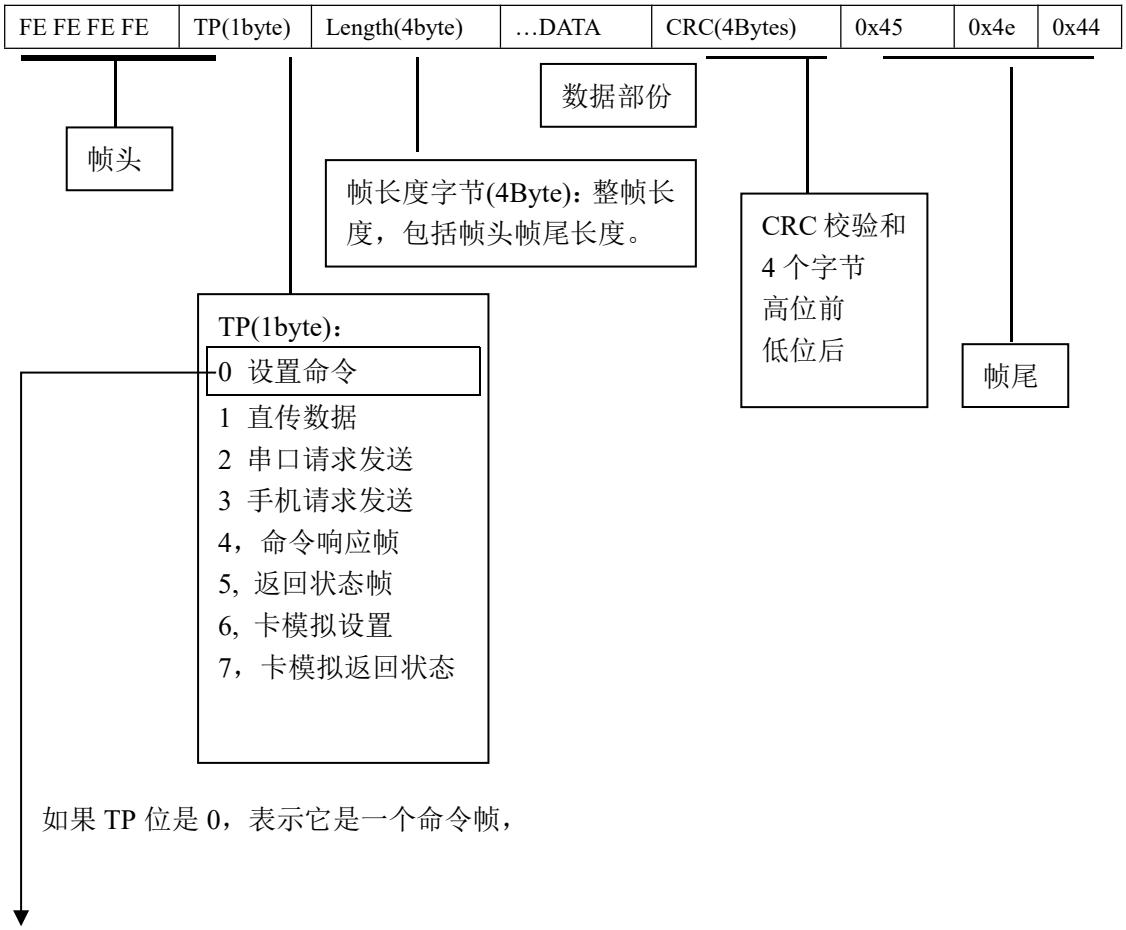
NFC2COM 提供了一些串口命令 API 与外部主控交互。

4.1 帧结构

这是外部主控的串口与手机间的通信

{0xFE,0xFE,0Xfe,0xFE}= 帧头

{0x45,0x4e,0x44}=" END";



如果 TP 位是 0，表示它是一个命令帧，

4.1.1 命令帧：

DATA 部份，固定是 4bytes 数据，定义如下

Byte[9]	Byte[10]	Byte[11]	Byte[12]				

Byte9: 命令代码

1 --- 查询固件版本

2---查询当前通信波特率

3---设置串口波特率

4---设置 P2P 数据

5---设置工作模式

6---查询固件应答

7---查询波特率应答

8—设置蜂鸣器

9—设置 IO

10—设置 LED

11—设置 IRQ 延迟数

0xe7 ---可以发送

0xe9 ----不能发送

解释

Byte9 值	Byte10
1	0~0
2	0~0
3	串口通信波特率 1~6 1---115200 2---9600 3---4800
4	
5	1---快速 SRAM 大数据传输 2—NDEF 数据
6	1~255 返回当前固件版本
7	预留
8	设置蜂鸣器 1 设置为开 0 设置为关
9	设置 IO，详细看 4.7 章
10	板载 LED 设置， 1 设置为开 0 设置为关 详细看 4.6.3 章节
11	设置 IRQ 延迟数 0—14 之间的数都可以，代表延迟数，可以根据自己需要去调节，出厂默认是 0

4.2 数据交互（动态卡模拟）

NFC2COM 模拟成 tag 方式与 NFC 手机进行数据交互，通过高速动态的修改模拟数据，实现快速交互，达到的效果是像模组的串口数据与 NFC 手机进行点对点通信那样。但是整个交互过程不需要操作，只需要把要传的数据在模组串口端发送或是在手机端发送，另一端就能收到。

4.2.1 串口发数据到手机

步骤 1：发送请求命令(包含要发送数据长度)，

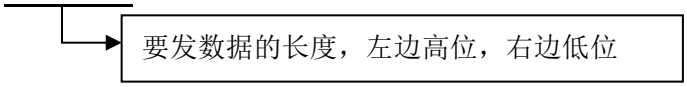
步骤 2：等待串口返回命令，如果是允许信号进行下一步

步骤 3：发送数据（没有帧头帧尾等信息）

举例：

Uart 请求发数据

fe fe fe fe 02 00 00 00 14 xx xx xx xx FF FF FF FF 45 4E 44



例如要发送 16byte 数据

fe fe fe fe 02 00 00 00 14 00 00 00 10 FF FF FF FF 45 4E 44

收到串口返回的数据有如下两种情况：

①FE FE FE FE 04 00 00 00 14 E7 00 00 FF FF FF FF 45 4E 44 ----可以发送

②FE FE FE FE 04 00 00 00 14 E9 00 00 FF FF FF FF 45 4E 44 ----当前忙,不能发送

当收到①，则开始发送真正要交互的 16byte 数据出去。

当收到②，则稍后再请求。

fe fe fe fe 02 00 00 00 14 00 00 00 10 FF FF FF FF 45 4E 44

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0F

注意：发数据的时候，如果数据长度大于 2048byte,需要分成 （n 帧+余帧）方式，

例如要发送的长度为 4562byte 数据，则是分成 2048+2048+466 三帧来发送，而且只能是最后一帧是不足 2048 字节的。

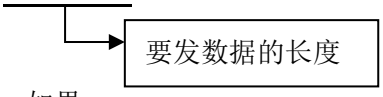
如果需要发送多帧，则每一帧发送后，模组会返回发送状态 ，如果出错了，会发出如前面章节描述的错误帧

4.2.2 手机发送数据到串口

步骤 1：发送请求命令

手机端请求发数据(在 JAVA 里高用 NFCA 接口实现，详细看我们提供的例子代码)

fe fe fe fe 03 00 00 00 14 xx xx xx xx FF FF FF FF 45 4E 44



步骤 2：监听返回的状态，如果

可以发送：FE FE FE FE 04 00 00 00 14 E7 00 00 FF FF FF FF 45 4E 44

当前忙： FE FE FE FE 04 00 00 00 14 E9 00 00 FF FF FF FF 45 4E 44

步骤 3：如果收到是 E7 的，就发送真实数据

4.3 模拟标准论坛卡（静态卡模拟）

本功能可以模拟 NFC type A，NFC forum type 2 论坛(NFC Forum)定义的各种功能卡片

（目前只提供蓝牙配对卡，TEXT 卡，网址 URI 卡这三种卡的模拟接口，需要其他卡的模拟，需要找风火轮定制）

注意：在操作这个命令的时候，手机必须离开天线区。

4.3.1 命令格式

FE FE FE FE	0x06	Length(4byte)	…DATA	CRC(4Bytes)	0x45	0x4e	0x44

DATA		
Byte[9]---type		
1 ： TEXT ， 文本卡	从 byte[10]起是 TEXT 数据	
2 ： URI ， 网卡卡	从 byte[10]是 Protocol fied 类型码，byte[11]开始 URI 数据	
3 ： BTPair，蓝牙配对卡	Byte[10]---name 长度+name+MAC(6byte)+Service(3byte)	
4 ： 卡擦除	擦成空白卡，Byte[10]---> 要擦的 block 数（1~64）	

NFC2COM 命令例子

例 1：模拟 TEXT:”123”

FE FE FE FE 06 00 00 00 14 **01** 31 32 33 FF FF FF FF 45 4E 44

例 2：模拟 TEXT:”hello,my name is NFC2COM”

FE FE FE FE 06 00 00 00 29 01 68 65 6c 6c 6f 20 6d 79 20 6e 61 6d 65 20 69 73 20 4e 46 43 32 43 4f 4d FF FF FF FF 45 4E 44

表格 4.4.1

Protocol fied 类型码	
0x01	http://www.
0x 02	https://www.
0x 03	http://
0x 04	https://
0x 05	tel:
0x 06	mailto:

0x 07	ftp://anonymous:anonymous@
0x 08	ftp://ftp.
0x 09	ftps://
0x 0a	sftp://
0x 0b	smb://
0x 0c	nfs://
0x 0d	ftp://
0x 0e	dav://
0x 0f	news:
0x 10	telnet://
0x 11	imap:
0x 12	rtsp://
0x 13	urn:
0x 14	pop:
0x 15	sip:
0x 16	sips:
0x 17	tftp:
0x 18	btspp://
0x 19	btl2cap://
0x 1a	btgoep://
0x 1b	tcpobex://
0x 1c	irdaobex://
0x 1d	file://
0x 1e	urn:epc:id:
0x 1f	urn:epc:tag:
0x 20	urn:epc:pat:
0x 21	urn:epc:raw:
0x 22	urn:epc:
0x 23	urn:epc:raw:

4.3.2 蓝牙配对卡：

NFC 手机读到这样的卡片，会自动调用蓝牙配对程序与配取卡中指定信息的蓝牙设备不用安装第三方 APK 就能实现的功能

此标签包含三种信息：

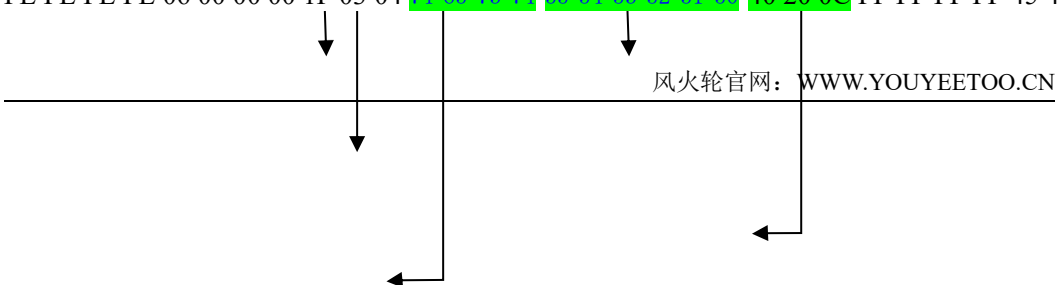
1、MAC 地址：蓝牙地址

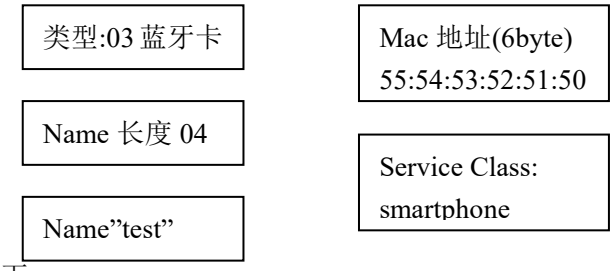
2、Deivce name:蓝牙设备名称

3、Device class :这个是蓝牙服务的定义详细看后面的表

例子：

FE FE FE FE 06 00 00 00 1F 03 04 74 65 73 74 55 54 53 52 51 50 40 20 0C FF FF FF FF 45 4E 44





蓝牙信息如下
name 描述：test
蓝牙地址：55:54:53:52:51:50
Serveice:　Smartphone

service class 有如下范围	
smartphone	40:20:0C
printer	04:06:80
camera	04:06:20
Wearable handset	20:04:04

4.3.3 TEXT 文本信息卡

NFC 手机在不装第三方 APK 情况下，读到这样的卡，能显示一条文本信息

例 1：模拟卡携带 TEXT 信息 :"hello,my name is NFC2COM"
命令如下：
FE FE FE FE 06 00 00 00 29 01 68 65 6c 6c 6f 20 6d 79 20 6e 61 6d 65 20 69 73 20 4e 46 43 32 43 4f 4d FF FF FF FF 45 4E 44
需要把字符转成 ASCII 码，装载到命令里。

例 2：模拟 TEXT 数据："风火轮科技"

FE FE FE FE 06 00 00 00 20 01 E9 A3 8E E7 81 AB E8 BD AE E7 A7 91 E6 8A 80 FF FF FF FF 45 4E 44

中文采用 UTF8 编码，所以要先编码后，再装载到命令里，详细装载格式看 4.4.1

4.3.4 URI 标签

NFC 手机读到这样的卡片，会直接用网页浏览器打开相应的地址。

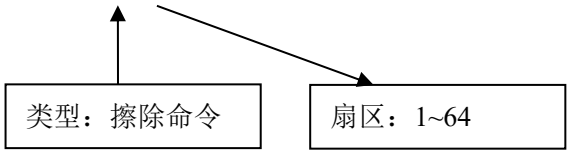
例 1：模拟网址卡：http://www.baidu.com

NFC2COM 命令如下：

FE FE FE FE 06 00 00 00 1B **02** 01 62 61 69 64 75 2E 63 6F 6D FF FF FF FF 45 4E 44

4.3.5 擦除卡

FE FE FE FE 06 00 00 00 14 **04** 01 00 00 FF FF FF FF 45 4E 44



4.4 设置波特率

NFC2COM 可以设置通信串口波特率，设置成功后，需要重启模组生效

4.4.1 命令格式

FE FE FE FE	0	00 00 00 14	DATA(4bytes)	CRC(4Bytes)	0x45	0x4e	0x44

DATA 部份，固定是 4bytes 数据，定义如下

Byte[9]	Byte[10]	Byte[11]	Byte[12]				

Byte9: 命令代码

1 --- 查询固件版本

2---查询当前通信波特率

3---设置串口波特率

4---设置 P2P 数据

5---设置工作模式

6---查询固件应答

7---查询波特率应答

0xe7 ---可以发送

0xe9 ----不能发送

Byte8

Byte8 值	
1	0~0
2	0~0
3	串口通信波特率 1~3 1---115200 2---9600 3---4800
4	
5	1---快速 SRAM 大数据传输 2—NDEF 数据
6	1~255 返回当前固件版本
7	返回当前设置的波特率 1~3 1---115200 2---9600 3---4800

例子：设置 115200

FE FE FE FE **00** 00 00 00 14 **03 01** 00 00 FF FF FF FF 45 4E 44

例子：设置 9600

FE FE FE FE **00** 00 00 00 14 **03 02** 00 00 FF FF FF FF 45 4E 44

例子：设置 4800

FE FE FE FE **00** 00 00 00 14 **03 03** 00 00 FF FF FF FF 45 4E 44

4.5 返回状态码

状态返回帧

FE FE FE FE	0x05	00 00 00 14(帧长度)	…DATA（4byte）	CRC(4Bytes)	0x45	0x4e	0x44

状态代码

返回 error 状态：

返回 FE FE FE FE 05 00 00 00 14 01 00 00 FF FF FF FF 45 4E 44

状态代码

状态代码	
0x01	Error:Uart 请求后发送后超时，如果再要发送，需要重新请求
0x02	Error:Uart 发送数据时 RF 写错误，可能是手机没在天线范围内 Uart 端收到这个错误信息，考虑重新发送当前帧。
0x03	**返回 ok 状态，表示 success**
0x04	**等待 NFC 基带响应超时**
0x05	
0x06	**手机 NFC 信号掉线**
0x07	**NFC 基带无响应，进入重启基带**
0x08	
0x09	**串口设置 ERROR**
0x0A	**设置卡模拟 TEXT 出错**

0x0B	**删除卡模拟里的数据出错**
0x0c	**模拟蓝牙卡出错**

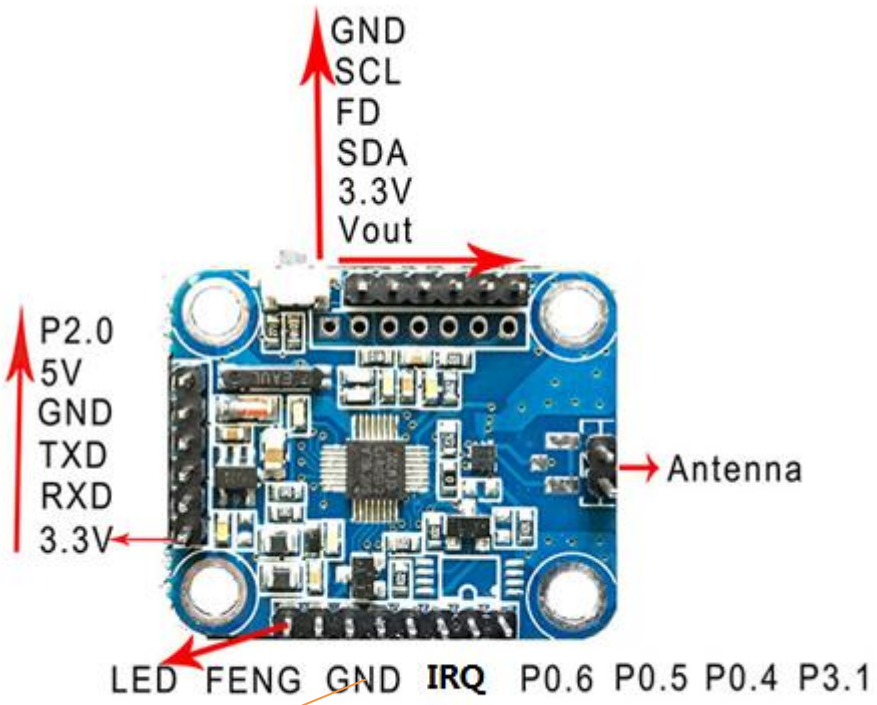
返回“OK”状态

FE FE FE FE 05 00 00 00 14 **03** 00 00 FF FF FF FF 45 4E 44

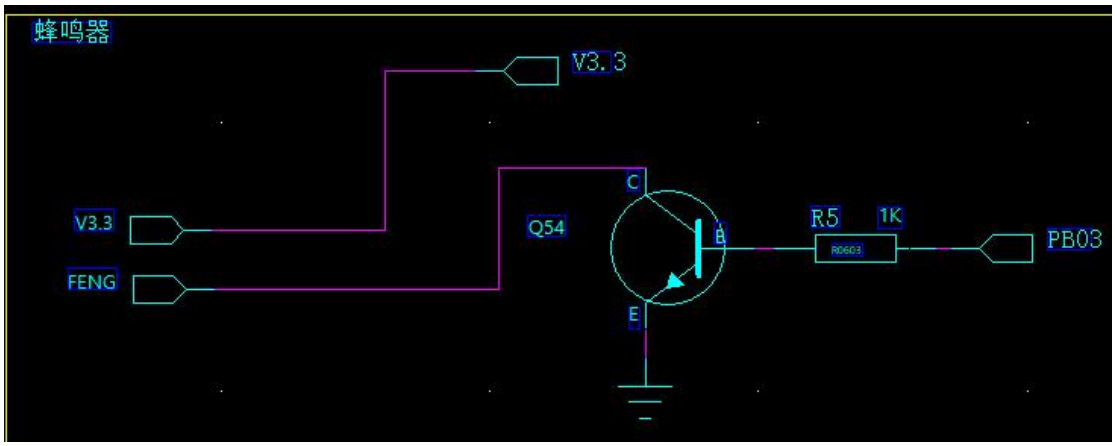
4.6 板载资源控制

◎1 个蜂鸣器驱动电路（可以外接蜂鸣器并通过串口命令控制它的开关）
◎1 个 LED 驱动电路（板载 LED 灯 并通过串口命令控制它的开关）
◎GPIO:3 个（P0.6/P0.5/P0.4/P3.1），可以通过串口命令控制和读取

4.6.1 蜂鸣器



FENG 这个管脚 ，低电平使能，也就是使能了，这是接地，需要搭配 V3.3 使用，



蜂鸣器开命令：

FE FE FE FE	0	Length(4byte)	…DATA	CRC(4Bytes)	0x45	0x4e	0x44

BYTE[9]	8 表明设置蜂鸣器命令
BYTE[10]	1 设置为开 0 设置为关
BYTE[11]	0
BYTE[12]	0

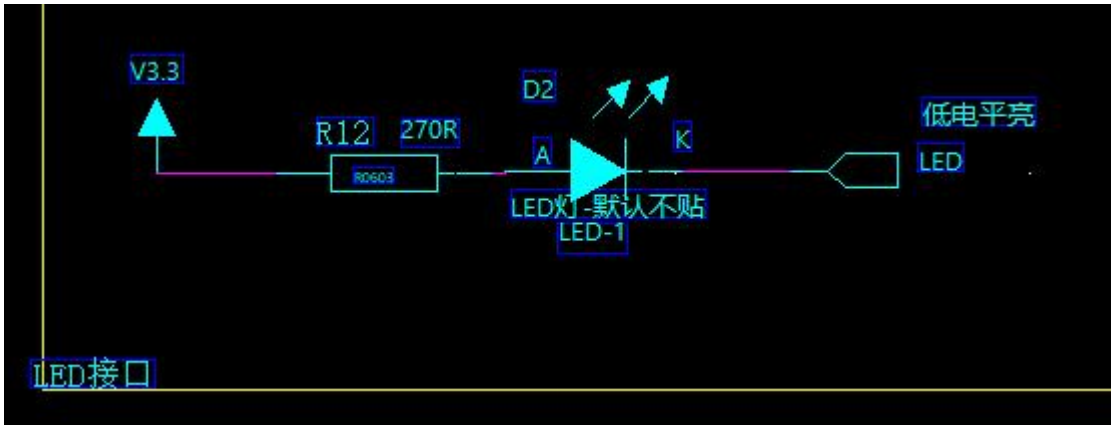
开：FE FE FE FE 00 00 00 00 14 08 01 00 00 FF FF FF FF 45 4E 44

关：FE FE FE FE 00 00 00 00 14 08 00 00 00 FF FF FF FF 45 4E 44

4.6.2 LED 灯（外接）

高电平 ----LED 关
低电平 ----LED 开

这个 LED 管脚，与板子上的 LED 灯也是同时连接的，但是默认状态下，板上的 LED 我们没有贴，所以控制 LED 也就是控制这个管脚的高低电平输出而已，外面可以接一个 LED 灯，也可以在我们板子上贴上 LED 灯，
如果 GPIO 不够用，其实这个就是 GPIO 功能，可以控制输出高低电平



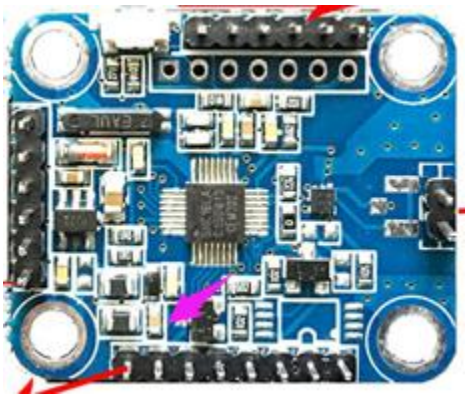
FE FE FE FE	0	Length(4byte)	…DATA	CRC(4Bytes)	0x45	0x4e	0x44

BYTE[9]	10 表明设置 LED 脚的命令
BYTE[10]	1 设置为开 低电平 0 设置为关 高电平输出
BYTE[11]	0
BYTE[12]	0

开：FE FE FE FE 00 00 00 00 14 0a 01 00 00 FF FF FF FF 45 4E 44

关：FE FE FE FE 00 00 00 00 14 0a 00 00 00 FF FF FF FF 45 4E 44

4.6.3 板载 LED 灯控制



如上图所示，板载了一个 LED 灯
低电平有效（PB04）

默认版本是没有贴 LED 灯的，如果需要板载 LED，贴上 D2，R12
这个 LED 灯与板子上 LED 管脚 连接的，

控制命令
就是与 LED 管脚同一个

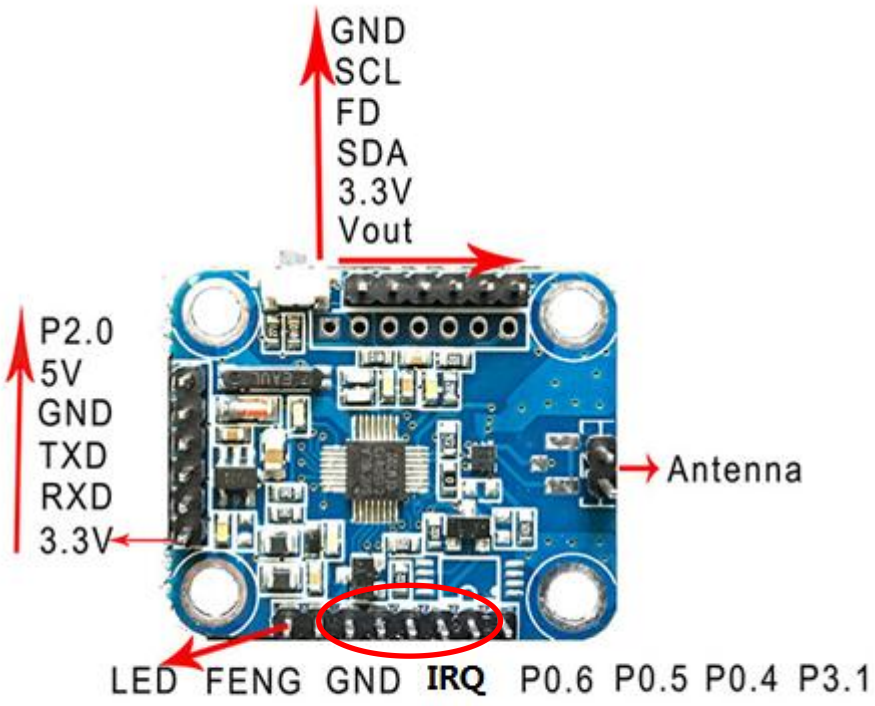
FE FE FE FE	0	Length(4byte)	…DATA	CRC(4Bytes)	0x45	0x4e	0x44

BYTE[9]	10 表明设置 LED 脚的命令
BYTE[10]	1 设置为开 0 设置为关
BYTE[11]	0
BYTE[12]	0

开：FE FE FE FE 00 00 00 00 14 0a 01 00 00 FF FF FF FF 45 4E 44

关：FE FE FE FE 00 00 00 00 14 0a 00 00 00 FF FF FF FF 45 4E 44

4.7 IO 口控制



注意，要读取 IO 值的时候，不能让对应 IO 悬空，悬空状态下读到的值是为高的，

FE FE FE FE	0	Length(4byte)	…DATA	CRC(4Bytes)	0x45	0x4e	0x44

BYTE[9]	9:表明设置 IO 命令
BYTE[10]	1: P0.6 输出 2 :P0.5 输出 3 : 读取 P0.6 电平 4 : 读取 P0.5 电平 5：读到 IO 的返回值 6：P0.4 输出 7：P3.1 输出 8 : 读取 P0.4 电平 9 : 读取 P3.1 电平 A: p2.0 输出 详细看 4.7 章
BYTE[11]	0:输出低电平 1:输出高电平
BYTE[12]	0

例子：
P0.6 输出高电平

：FE FE FE FE 00 00 00 00 14 09 01 **01** 00 FF FF FF FF 45 4E 44

P0.6 输出低电平

：FE FE FE FE 00 00 00 00 14 09 01 **00** 00 FF FF FF FF 45 4E 44

P0.5 输出高电平

：FE FE FE FE 00 00 00 00 14 09 02 **01** 00 FF FF FF FF 45 4E 44

P0.5 输出低电平

：FE FE FE FE 00 00 00 00 14 09 02 **00** 00 FF FF FF FF 45 4E 44

读取 P0.6 电平

：FE FE FE FE 00 00 00 00 14 09 03 00 00 FF FF FF FF 45 4E 44

读取 P0.5 电平

：FE FE FE FE 00 00 00 00 14 09 04 00 00 FF FF FF FF 45 4E 44

读取 P0.4 电平

：FE FE FE FE 00 00 00 00 14 09 08 00 00 FF FF FF FF 45 4E 44

读取 P3.1 电平

：FE FE FE FE 00 00 00 00 14 09 09 00 00 FF FF FF FF 45 4E 44

IO 状态回复

：FE FE FE FE 00 00 00 00 14 09 05 **00** 00 FF FF FF FF 45 4E 44

P0.4 输出高电平

：FE FE FE FE 00 00 00 00 14 09 06 **01** 00 FF FF FF FF 45 4E 44

P3.1 输出低电平

：FE FE FE FE 00 00 00 00 14 09 07 **00** 00 FF FF FF FF 45 4E 44

P2.0 输出高电平

：FE FE FE FE 00 00 00 00 14 09 0a **01** 00 FF FF FF FF 45 4E 44

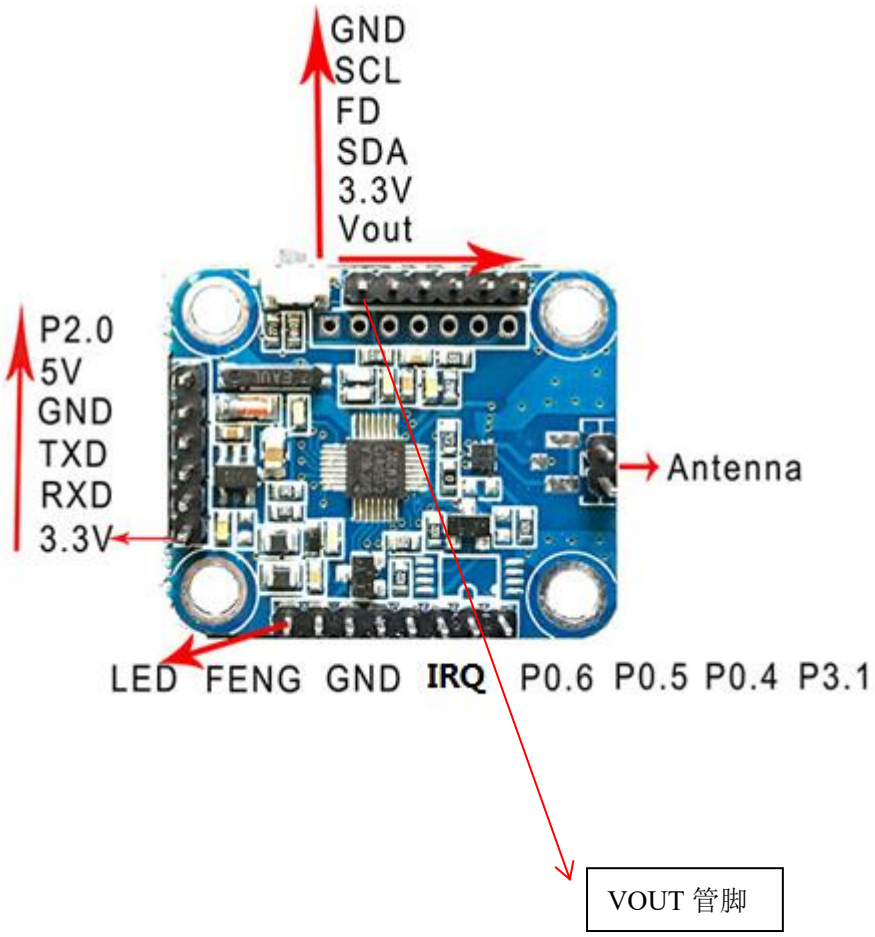
P2.0 输出低电平

：FE FE FE FE 00 00 00 00 14 09 0a **00** 00 FF FF FF FF 45 4E 44

4.8 手机靠近检测功能

当手机靠近 NFC2COM 天线区域，在模组的 VOUT 管脚，就会有一个高电平输出，电压：1.8~2.9V，

手机离开天线区域，则该管脚会变为低电平。

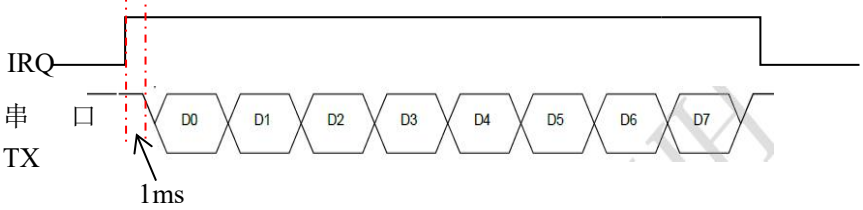


用户可以使用它作为一个手机靠近的提示信号，触发外部单片机或主控。

4.9 IRQ 引脚信号延迟设置

- **中断 IRQ：**

当串口有数据要发出来，会先通过 IRQ 管脚输出一个高电平，平时为低电平



外部 MCU 可以接 IRQ 管脚作为中断，监听上升沿作为信号，当得到中断就唤醒并接收模组传过来的串口数据

这个时间可以通过命令进行修改，默认是 1ms，可以加大

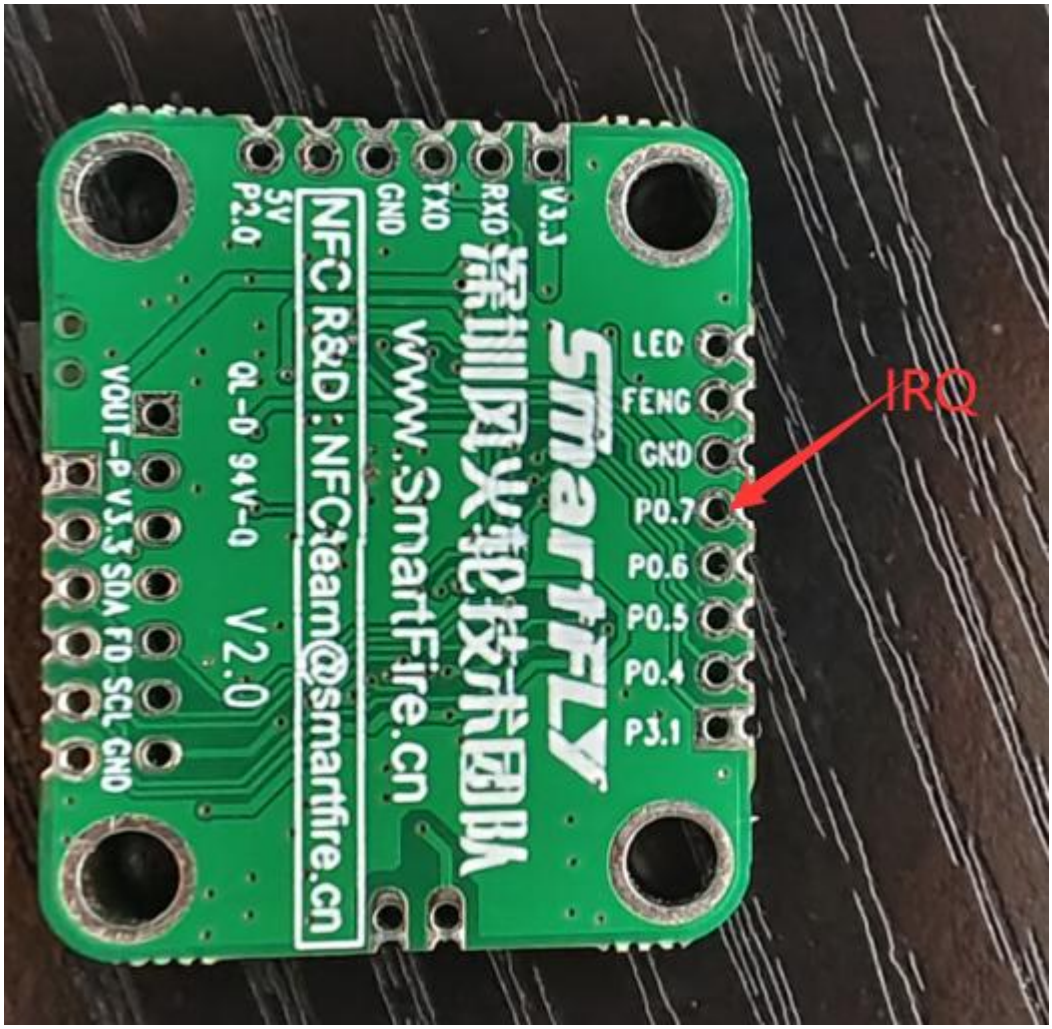
命令

FE FE FE FE	0	Length(4byte)	…DATA	CRC(4Bytes)	0x45	0x4e	0x44

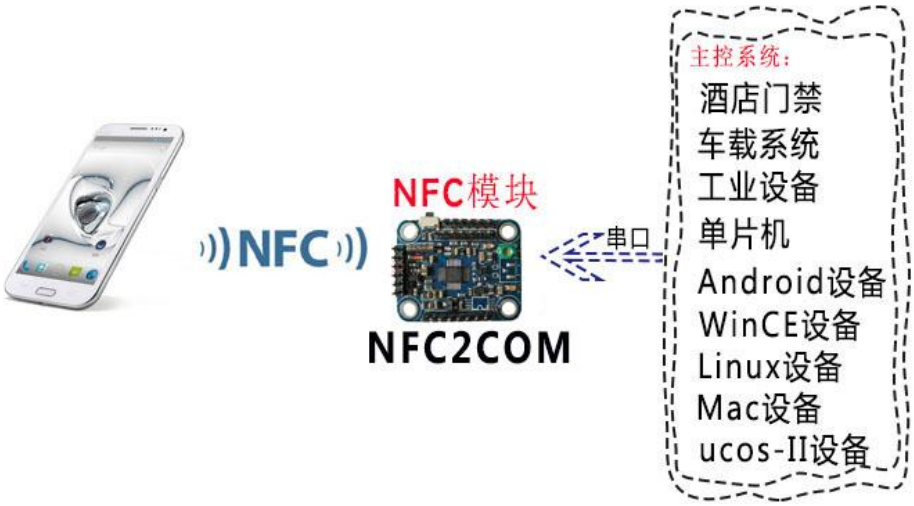
BYTE[9]	11(0x0b)设置 IRQ 延迟的命令
BYTE[10]	这个字节的值标识 IRQ 信号发生后延迟多久，才正式发串口数据 0 -- 0x0E 之间的数字都可以，单位是 ms。 建议尽量越小越好， 出厂默认是 0. 不建议调整。
BYTE[11]	0
BYTE[12]	0

设置 1ms：FE FE FE FE 00 00 00 00 14 0b 01 00 00 FF FF FF FF 45 4E 44
设置 5ms：FE FE FE FE 00 00 00 00 14 0b 05 00 00 FF FF FF FF 45 4E 44

回复出厂设置：FE FE FE FE 00 00 00 00 14 0b 00 00 00 FF FF FF FF 45 4E 44


IRQ
LED
FENG
GND
P0.7
P0.6
P0.5
P0.4
P3.1
VJ.J
RXD
TXD
GND
5V
P2.0
NFC R&D:NFCteam@smartfire.cn
www.SmartFire.cn
深圳风火轮技术团队
SmartFLY
V2.0
VOUT-P VJ.J SDA FD SCL GND

第 5 章 应用场景



5.1 NFC 卡模拟器应用

典型应用场景是用于与手机进行通信，手机是读写器角色，实现支付功能。

NFC2COM 模组 + 外壳 = NFC 智能卡模拟器（跨平台）

应用领域有如下

〇网上银行及网上购物
〇电子商务
〇电子钱包余额查询
〇网络访问
〇客户积分优惠
〇身份验证
〇票务
〇网上博彩
〇停车场收费系统
〇自动收费系统
〇公共交通
〇门禁系统
〇考勤
〇自动贩卖机
〇非接触式公用电话
〇物流及供应链管理

该应用场景，**NFC2COM 模组使用 USB 接口与主控制器连接**，例如与 PC 连接，风火轮提供 PC 上的开发示例 DEMO 源码，

5.2 嵌入式产品应用

NFC2COM 模组---应用于其它嵌入式平台（串口）

本应用场景，针对一些其它的嵌入式平台，例如

WINCE 平台的工业设备

UCOS 平台的工业设备

Linux 平台的工业设备

android 平台的工业设备

...等等

只要通过 UART（RS232）口与 NFC 模组连接，我们已把 NFC 功能提取成指令形式，只要会串口编程，就能方便的使用 NFC 通信功能，不用去研究复杂的 NFC 应用协议栈。

这种方式，用户可以最快的方式，尽量少改动已有系统 把 NFC 功能加入您的设备。

5.3 NFC2COM 模块用在广告机上

在广告机的基础上嵌入 NFC 模块(NFC2COM)，升级为 NFC 广告机，使得广告机更加智能和便捷。实现优惠券派发，用户签到和打卡功能等等

1、会员登录

非 NFC 广告机的会员登录需要手动输入手机号或者相应的会员 ID，才能进行深入的操作。当广告机增加 NFC 功能后，只需 NFC 手机在 NFC 广告机感应一下，即可完成登录验证

2、优惠券下载

当消费者在 NFC 广告机上看到感兴趣的优惠信息，把 NFC 手机放到 NFC 广告机的感应处，即可完成优惠信息的下载到手机上。

3、提供详情

当消费者需要了解广告机上某种商品的具体信息时，而广告机因为自身设备的限制，无法提供更加详细的产品信息，可以选择把产品的信息网址，用 NFC 手机在 NFC 广告机上感应一下，即可在手机打开该产品的信息网址。

5.4 门禁应用

NFC2COM 模组，可以直接引出蜂鸣器，内置控制开锁的 GPIO，可直接应用于门禁系统，可以用手机来作为开门的工具(需安装指定 APK 授权)，

以上功能，有需要可向风火轮团队定制　peter@smartfire.cn

第 6 章 ASCII 码表

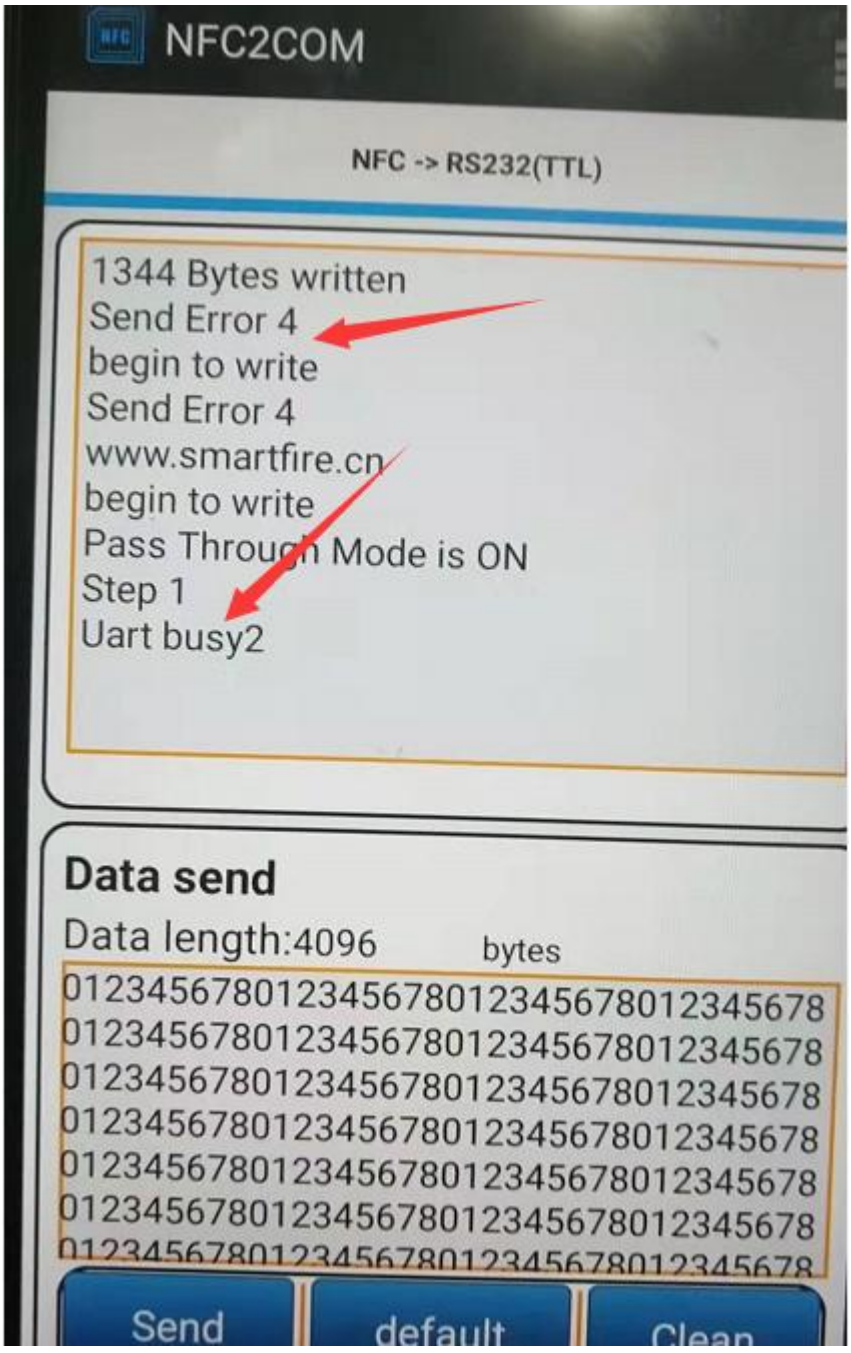
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

第 7 章 常见问题 Q&A

7.1 经常失败的问题

有些用户反馈，APP 端经常会遇到这样的失败显示



这种出错，一般有如下原因造成

1， 如果模组接的是单片机，请检查单片机是不是频繁地向模组串口发送数据，一直占用资源，建议单片机向模组每次发串口数据**间隔 750ms** 以上，以便有时间给手机端有回应的机会，甚至 1 秒以上，更佳。例如，单片机要向手机发数据，发一了帧报告帧，获取模

组回应为 E7 信道有空，则应等 750 毫秒再发送数据帧。而不是当场直接发送收到 E9 的话，建议要等 2 秒再重新发。

2， 如果模组是接 USB 转串口接到 PC 上调试，那有可能是天线信号不好，或者没有对准手机这边的天线，造成信号差，通信慢，释放资源慢。