

---

无锡谷雨电子有限公司

# ZG-Mxx 快速开始

---

[www.wx-iot.com](http://www.wx-iot.com)

2015/9/18

Start

版本记录	更改记录	撰写人
A	初始化版本	戚二进 2015-09-18

## 目录

1 前言.....	3
1.1 产品简介.....	错误!未定义书签。
1.2 功能特点.....	错误!未定义书签。
1.3 应用领域.....	错误!未定义书签。
2 ZG-Mxx 硬件说明 .....	9
2.1 引脚定义.....	9
2.2 DC 特性 .....	9
2.3 ADC 特性.....	错误!未定义书签。
2.4 复位引脚.....	错误!未定义书签。
2.5 ZG-M0.....	13
2.6 ZG-M1.....	20
2.7 ZG-M1E.....	21

## 1 前言

ZG-M 系列 zigbee 模块目前包括 ZG-M0, ZG-M1 和 ZG-M1E 模块。它们是不同的形式的 zigbee 模块以满足不同的需求。ZG-M0 是以 PCB 天线, 且带有半孔封装有形式存在。如图 1 所示。ZG-M1 是以外接 SMA 天线, 且引脚是 1.27 间距的排针的形式存在。如图 2 所示。而 ZG-M1E 是在 ZG-M1 硬件的基础上, 增加了功放模块, 目的使信号的传输距离更远, 且穿墙的能力更加的出色。如图 3 所示。ZG-M0, ZG-M1, ZG-M1E 都是基于 TI 公司的 CC2530F256 芯片, 运行 Zigbee PRO Z-Stack 协议。



图 1 ZG-M0



图 2 ZG-M1

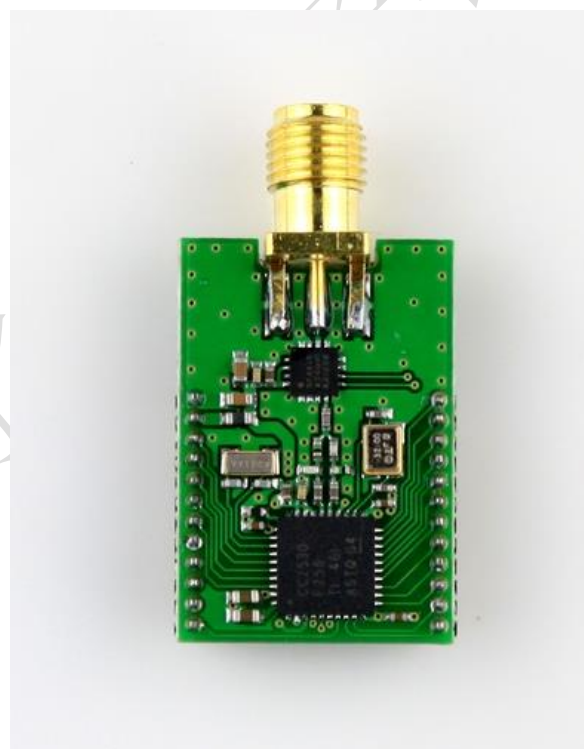


图 3 ZG-M1E

## 1.1 简单介绍

这篇文档是基于无锡谷雨电子的 ZG-Mxx 模块与 ZG-Mxx 测试底板（NewSmartRF 开发板

也可以) 进行描述与操作。如果用户没有测试底板或者 NewSmartRF 开发板, 也可以通过相关的串口线将 ZG-Mxx 模块的串口与 PC 机进行联接。(串口线可以是 USB 转 TTL 串口线)。

## 1.2 下载相关软件工具

当用户收到 ZG-Mxx 模块时, 可以到无锡谷雨的官网 <http://www.wx-iot.com> 产品中心点选 ZIGBEE 透传模块中任何一个模块都可以在其资料下载页面中找到, 也可以到 <http://pan.baidu.com/s/1qdp9H5> 这个百度云盘中下载测试与配置工具, 以便在 PC 端能进行方便快捷进行测试。

其软件工具包括以下几个:

- 数据调试工具 V1.4.rar 截止写作日期, 已经更新到 V1.4
- 参数配置工具 V1.5.rar 截止写作日期, 已经更新到 V1.5
- ZTop.rar V3 版本使用
- ZTopology.rar V1, V2 版本使用
- 波特率测试工具.rar

数据调试工具是类似于一个串口调试工具的一个上位机软件。在其中可以收发串口数据, 也可以读取模块的相关内部的相关参数。其界面如下所示。

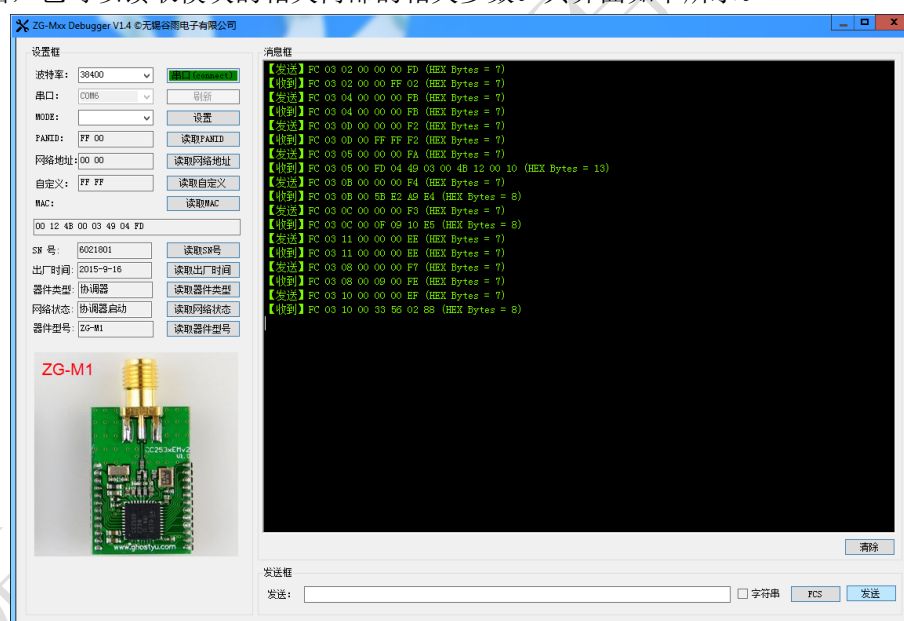


图 1 ZG-Mxx Debugger

模块参数配置工具可以读取模块内部的参数, 也可以对相关的参数进行设定。其工作界面如下所示。



图 2 ZG-Mxx Setting

ZTOP 工具软件可以查看网的连接结构。在 ZTOP 中可以方便快捷知道当前网络的连接信息。其界面如下

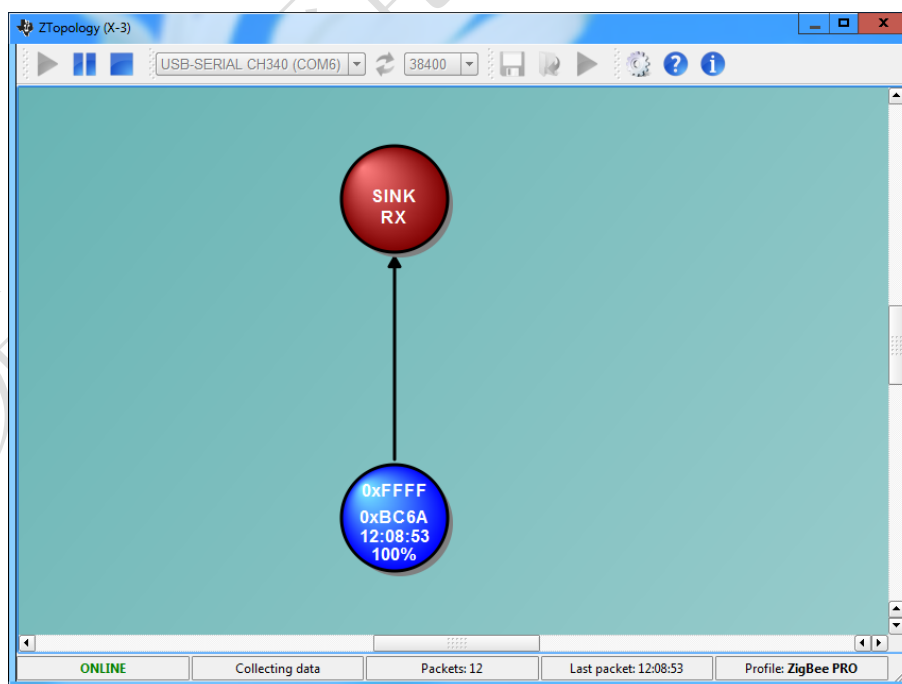


图 3 ZTOP

波特率测试工具是可以测试出当前模块工作串口波特率大小。当用户不知道当前模块串口波特率时，使用这个模块可以轻松完成模块波特率的测试。其界面如下图所示。

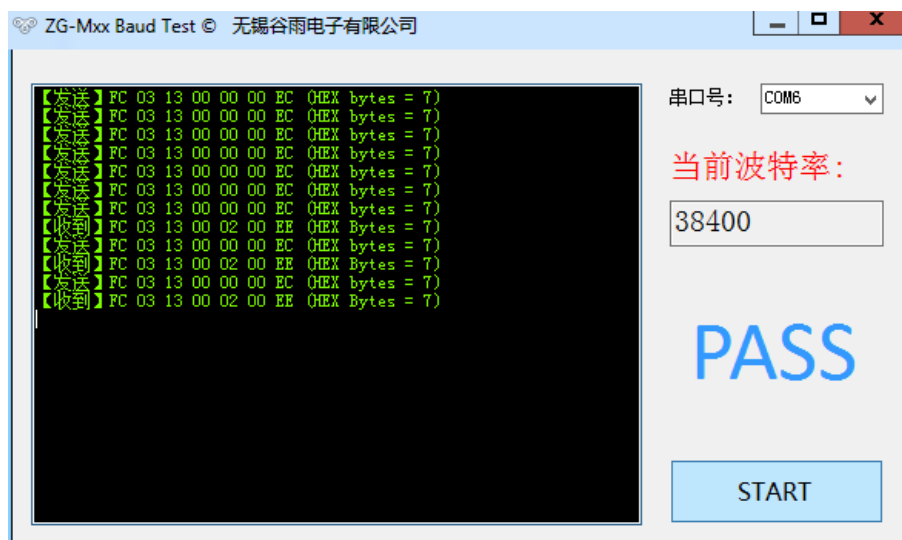
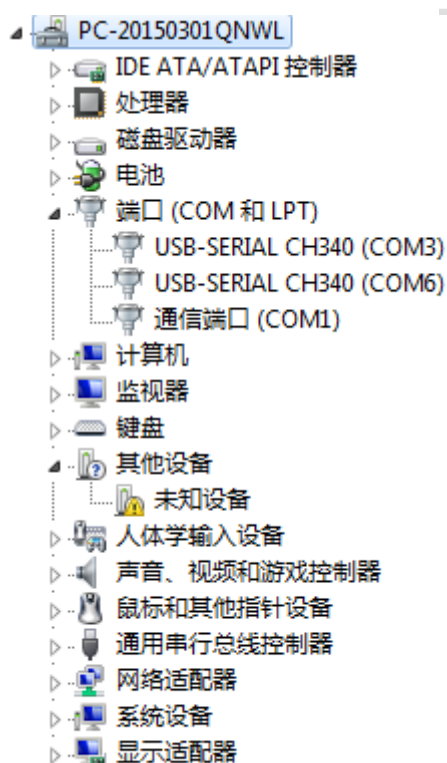


图 4 ZG-Mxx Baud Test

### 1.3 测试所用主板

无锡谷雨电子的 ZG-Mxx 测试底板配合 ZG-Mxx 模块，可以轻松完成测试系统组建。每个测试底板安装一个 ZG-Mxx 模块，就可以组成一个 zigbee 网络节点。测试底板通过 Mini USB 或者通过 RS232 连接到电脑，会在电 PC 上出现一个串口。如图所示。



ZG-Mxx 模块支持 ZG-Mxx 测试底板，NewSmartRF 开发板。两个测试底板，完成的功能都是一样的，都是 PC 端产生的一个串口。下图是两个测试底板图片。





图 5 ZG-Mxx 测试底板

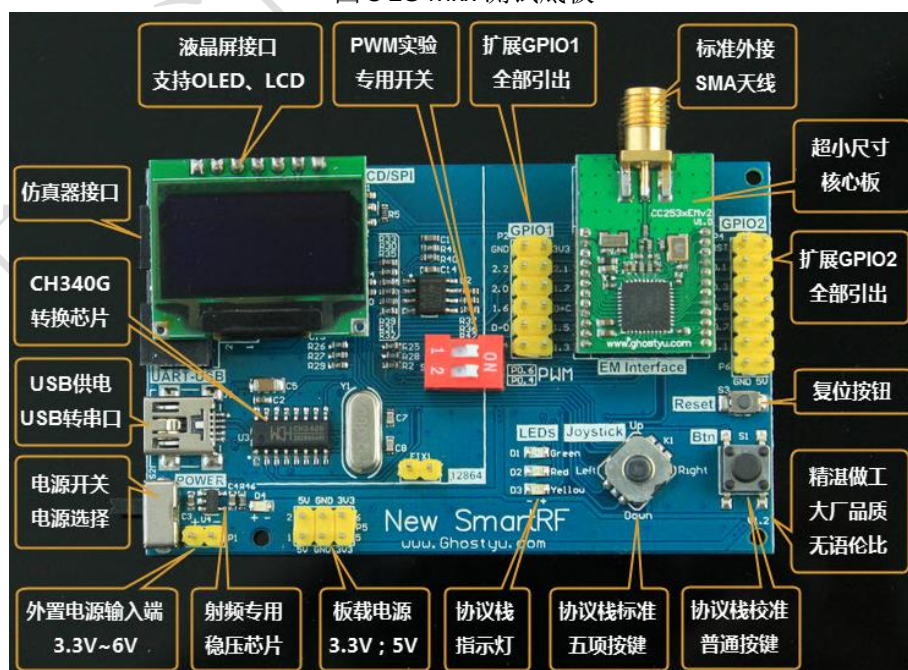


图 6 NewSmartRF 开发板

## 2 ZG-Mxx 模块测试

ZG-Mxx 有三种型号分别为 ZG-M0, ZG-M1, ZG-M1E。详细的硬件说明, 可以查看《ZG-Mxx 硬件说明.pdf》

### 2.1 测试软件运行要求

ZG-Mxx 所有相关的测试软件, 都是为了 winxp 或更新的 win7 操作系统所编写。目前还没有 MAC 和 linux 操作系统上运行的测试软件。

对个人 PC 要求如下:

- ✧ 运行 windows xp 或更新的操作系统
- ✧ .NET Framework 4.0 (win7 及以上的操作系统本身自带)
- ✧ 至少一个 USB 接口或 RS232 接口, 用于数据通信
- ✧ 如果使用 MiniUSB 线, 则要安装一个 CH340G 的驱动

### 2.2 ZG-Mxx 协调器配置

ZG-Mxx 模块在出厂时, 默认都是路由器的 ZIGBEE 模块。所以上电不会自动组网, 除非空间里存在一个在相同信道, 有一个相同的 PanID 协调器。

现在我们要做的事情就是将其中的一个 ZG-Mxx 模块, 设置成协调器。

将 ZG-Mxx 模块安装在 ZG-Mxx 测试底板, 用 MiniUSB 线连接 ZG-Mxx 测试底板与 PC 机。这时 PC 端设备管理器里就会多一个串口。

打开 ZG-Mxx Setting.exe 软件, 在通信口名称里选择上述的串口号, 然后点击串口(close)按钮进行连接。此时的按钮就会从橙色变成绿色。如下图所示。



图 7 ZG-Mxx Setting 与 ZG-M1 通信

要将其设成协调器，只要点击**器件类型**后的更改按钮，就会弹出对话框。在更改器件类型一栏中选择协调器，然后点击确定即可。设定后的模块参数信息如下图 8 所示。



图 8 协调器设定成功

设定成路由器或终端也是按照上述方法一样进行操作。其他参数的设定也是一样的操作。只要在 ZG-Mxx Setting 界面里有更改按钮的参数，都是可以进行修改的。用户可以自己根据需要进行操作与测试。

如果操作过程中，希望回到出厂时的状态，只要在界面里点击恢复出厂设置按钮即可完成。

## 2.3 组网进行数据测试

上述的操作，是将其中一个模块设成协调器。协调器启动好后，空间中就会存在 panid 标识的 zigbee 网络。这时只要上电其他 ZG-Mxx 模块，他们就会加入协调器建立的网络里。其主要标识就是测试底板的 LED 灯会 2 秒中闪烁一次。另外一个标识就是在 ZG-Mxx Setting 里的父地址，与父 MAC 地址就会有数字出现，不再是 0。其如下图 9 所示。



图 9 加入网络后的 ZG-Mxx Setting

加入网络后两个模块就可以进行数据无线收发测试了。在数据接收测试过程中我们使用 ZG-Mxx Debugger 进行数据的测试工作。

打开两个 ZG-Mxx Debugger 数据测试工具，在串口一栏中选择相应的串口号，然后点击串口（Close）按钮进行连接。在没有数据交互时按钮的背景色是橙色的，如果有数据交互橙色背景就会绿色。如下图 10 所示。

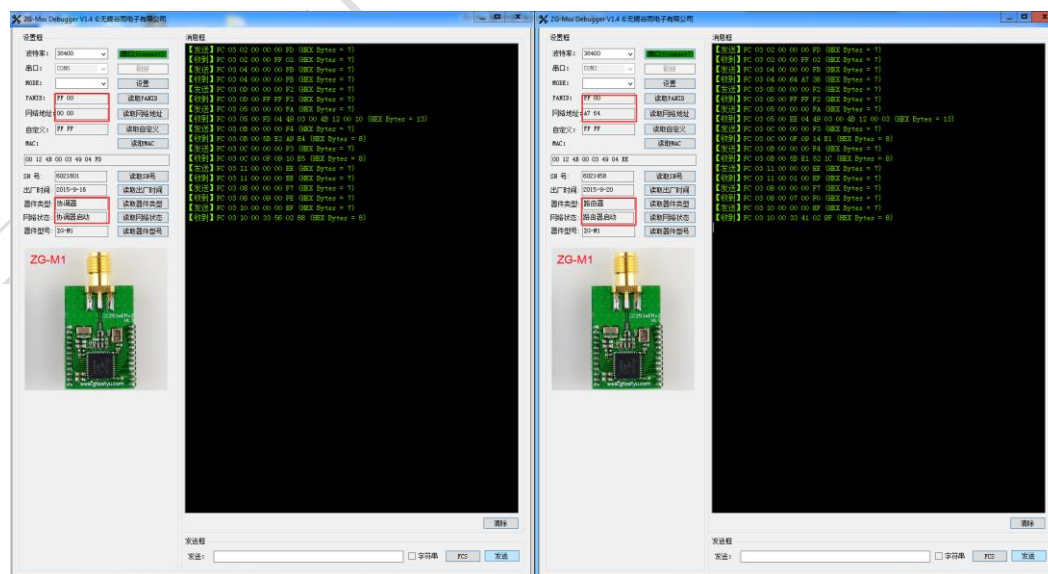
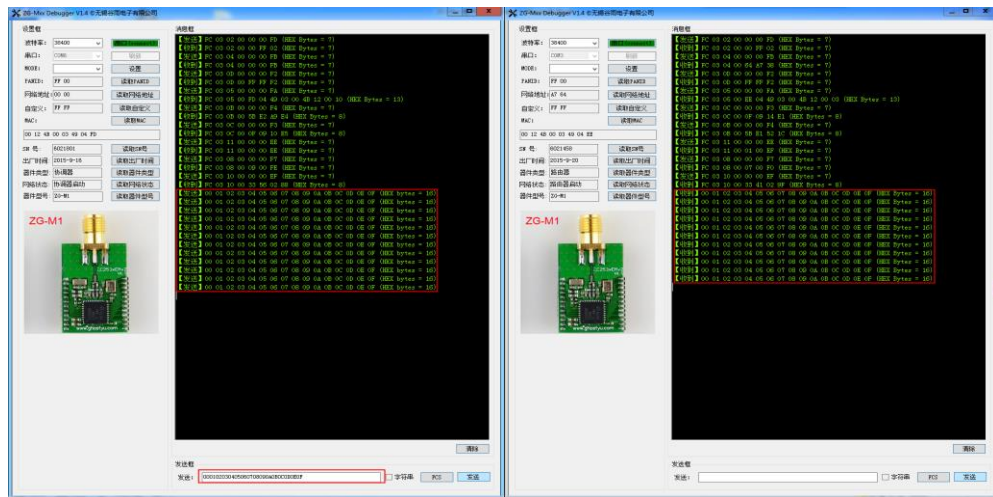


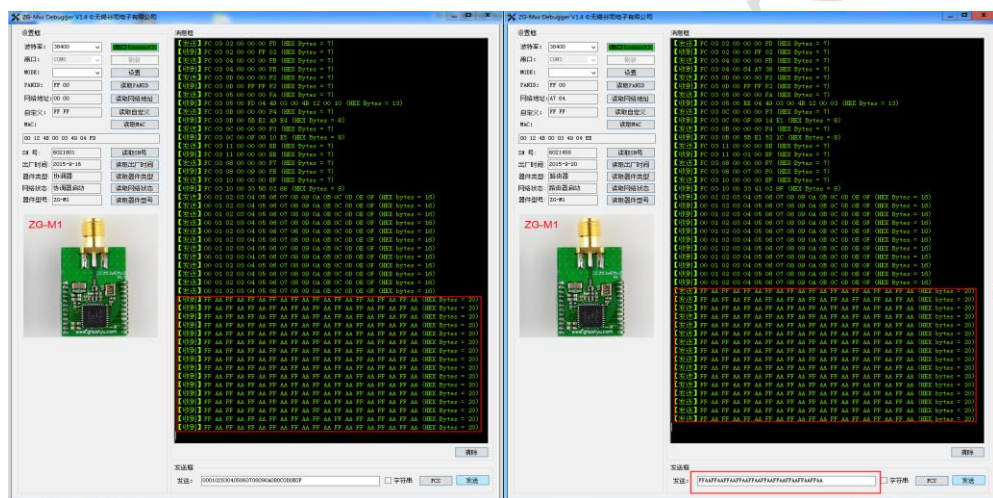
图 10 ZG-Mxx Debugger 与数据连接图

- ◇ 协调器向数网络中广播数据（协调器默认发送数据方式是广播）





◇ 路由器向协调器发关数据（默认路由器向协调器单播）



## 2.4 数据通信方式设定与通信测试

ZG-Mxx 模块的数据通信方式有五种，见下表所示。默认数据通信方式为 0。

表一 数据传输方式

传输模式值（十六进制）	数据透传方式下	点对点传输方式下
00	数据透传	Zigbee 短地址寻址，含包头包尾
01	数据透传+zigbee 短地址	Zigbee 短地址寻址，含包头包尾
02	数据透传+MAC 地址	Zigbee 短地址寻址，含包头包尾
03	数据透传+自定义地址	Zigbee 短地址寻址，含包头包尾
04	数据透传	Zigbee 短地址寻址，不含包头包尾
05	数据透传	Zigbee 自定义地址，不含包头包尾
>05	数据透传	Zigbee 自定义地址，不含包头包尾

注：当数据传输方式为数据透传 01，02，03 及点对点传输方式时，最大能传输的数据包大小必须限制在 80 个字节之内，否则超过的部分将会被丢弃。但我们推荐帧大小限制在 32 个字节之内。

## 2.4.1 传输模式为 0

### ◇ 数据透传

ZG-Mxx 默认的数据传输方式 0，具体的数据测试如 2.3 节。此处不作叙述。

### ◇ 点对点数据传输

用 ZG-Mxx Debugger 将其中一个 ZG-Mxx 模块的数据传输方式设成 00，然后在发送框中采用 FD 发送数据。

例如：FD 02 64 A7 02 03

FD：点对点数据传输方式的标志

02：用户数据的长度

64：目标地址的低字节

A7：目标地址的高字节

02：用户数据

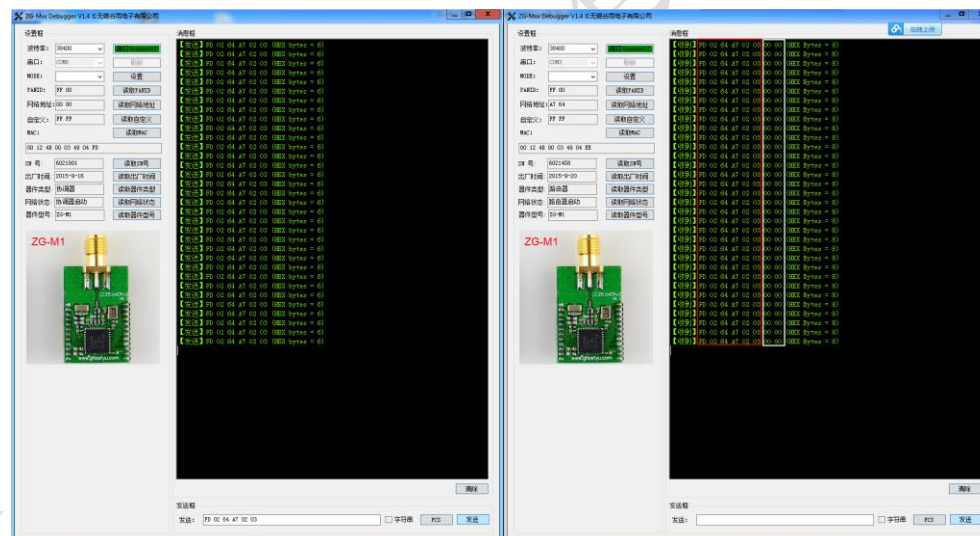
03：用户数据

接收方 A7 64 的 ZG-Mxx 模块就会收到

FD 02 64 A7 02 03 00 00

00：发送方地址的低字节

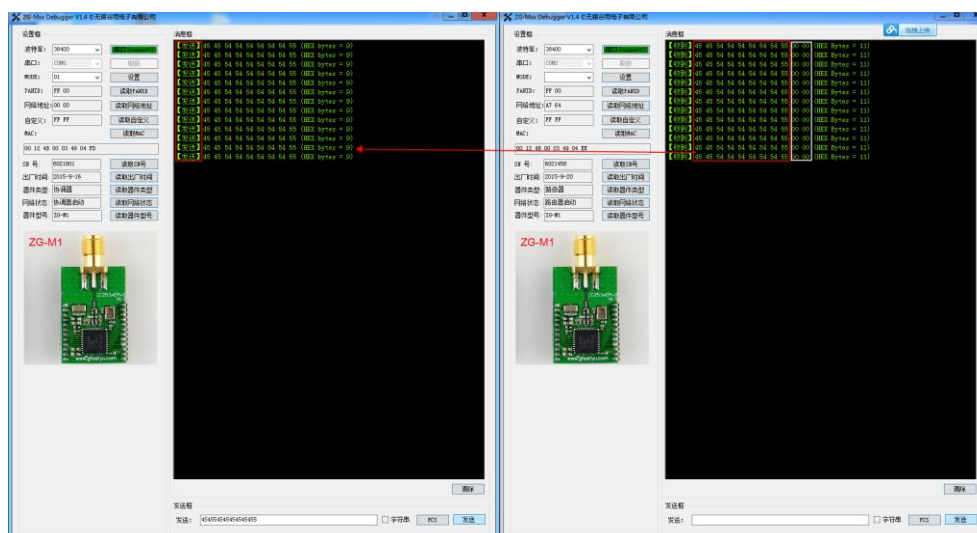
00：发送方地址的高字节



## 2.4.2 传输模式为 01

### ◇ 数据透传

用 ZG-Mxx Debugger 将其中的一个模块的数据传输方式设成 01，这时数据发送方会在发送数据的后面加上自己的 zigbee 网络地址。如下图所示。



### ◇ 点对点数据传输

用 ZG-Mxx Debugger 将其中一个 ZG-Mxx 模块的数据传输方式设成 00, 然后在发送框中采用 FD 发送数据。

例如: FD 02 64 A7 02 03

FD: 点对点数据传输方式的标志

02: 用户数据的长度

64: 目标地址的低字节

A7: 目标地址的高字节

02: 用户数据

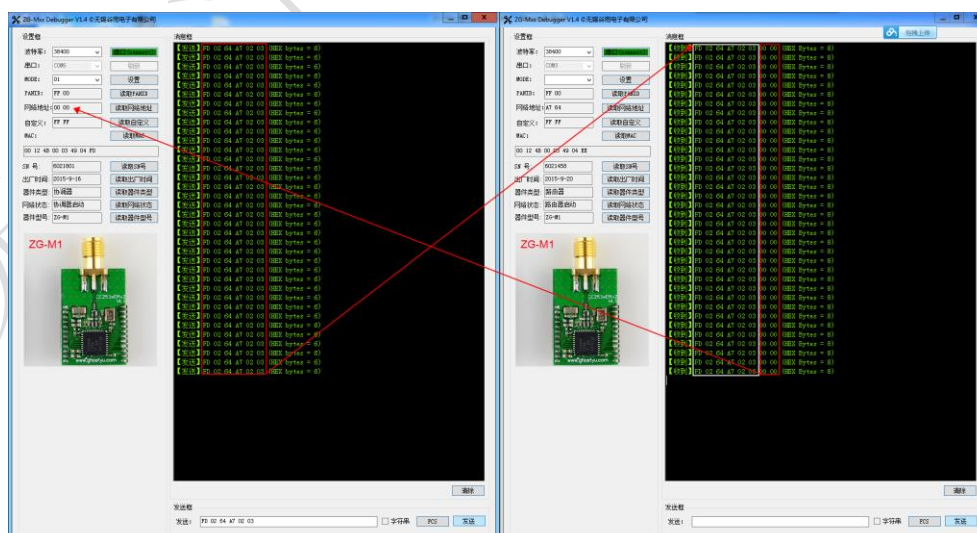
03: 用户数据

接收方 A7 64 的 ZG-Mxx 模块就会收到

FD 02 64 A7 02 03 00 00

00: 发送方地址的低字节

00: 发送方地址的高字节

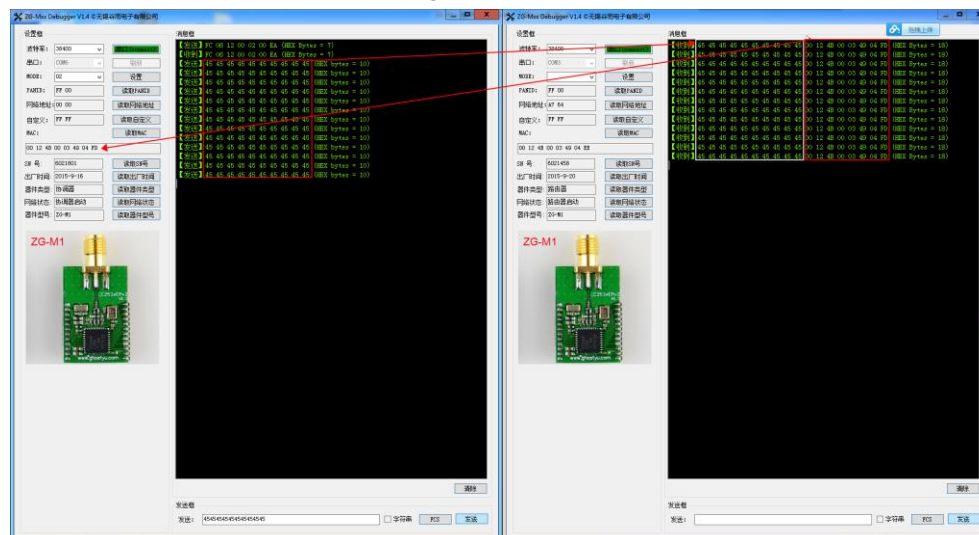




## 2.4.3 传输模式 02

### ◇ 数据透传

用 ZG-Mxx Debugger 将其中的一个模块的数据传输方式设为 02，这时数据发送方会在发送数据的后面加上自己的 zigbee MAC 地址。如下图所示。



### ◇ 点对点数据传输

用 ZG-Mxx Debugger 将其中一个 ZG-Mxx 模块的数据传输方式设为 02，然后在发送框中采用 FD 发送数据。

例如：FD 02 64 A7 02 03

FD：点对点数据传输方式的标志

02：用户数据的长度

64：目标地址的低字节

A7：目标地址的高字节

02：用户数据

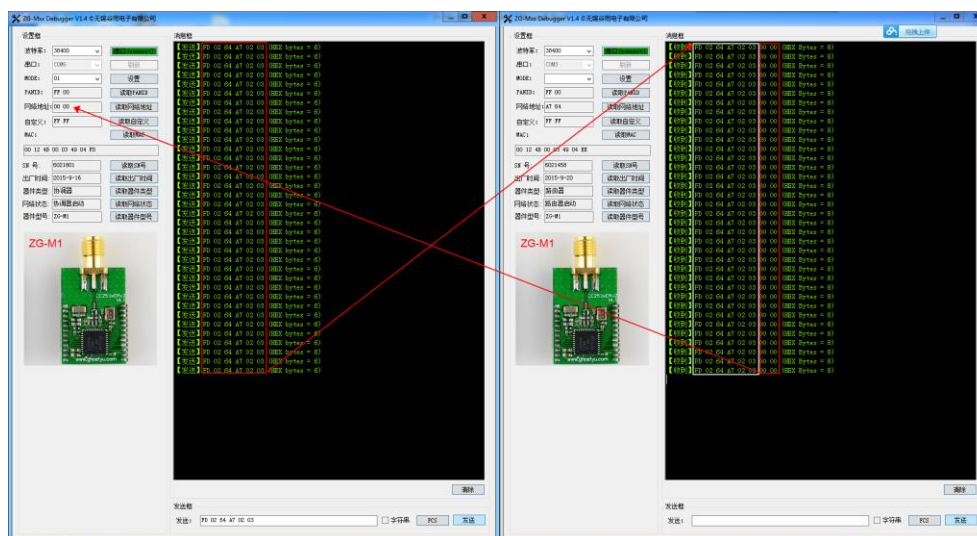
03：用户数据

接收方 A7 64 的 ZG-Mxx 模块就会收到

FD 02 64 A7 02 03 00 00

00：发送方地址的低字节

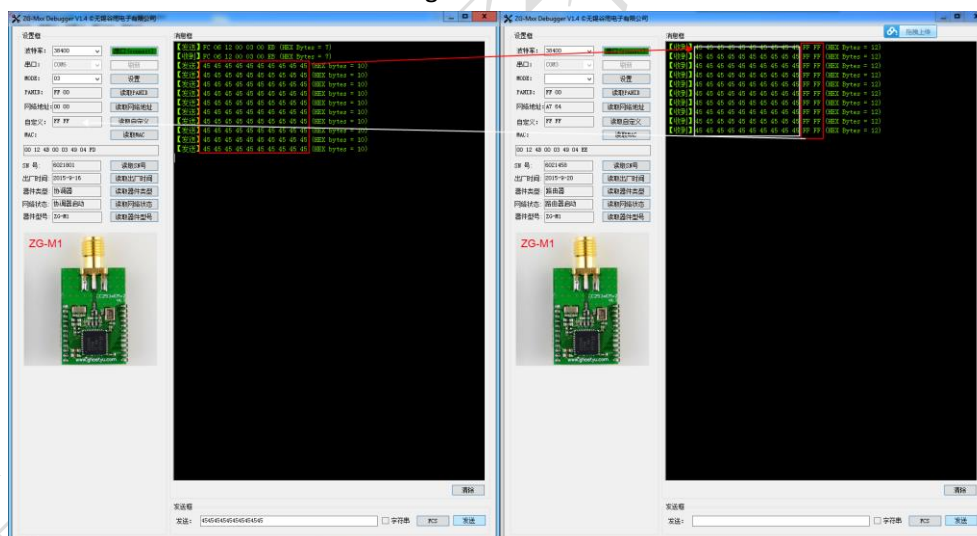
00：发送方地址的高字节



## 2.4.4 传输模式为 03

### ◇ 数据透传

用 ZG-Mxx Debugger 将其中的一个模块的数据传输方式设为 03，这时数据发送方会在发送数据的后面加上自己的 zigbee 自定义地址。如下图所示。



### ◇ 点对点数据传输

用 ZG-Mxx Debugger 将其中的一个 ZG-Mxx 模块的数据传输方式设为 03，然后在发送框中采用 FD 发送数据。

例如：FD 02 64 A7 02 03

FD：点对点数据传输方式的标志

02：用户数据的长度

64：目标地址的低字节

A7：目标地址的高字节

02：用户数据

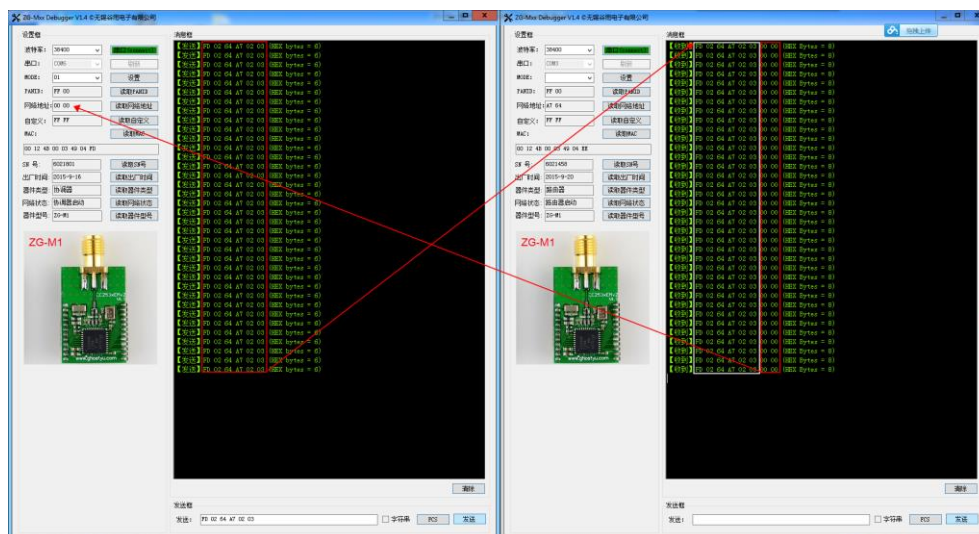
03：用户数据

接收方 A7 64 的 ZG-Mxx 模块就会收到

FD 02 64 A7 02 03 00 00

00: 发送方地址的低字节

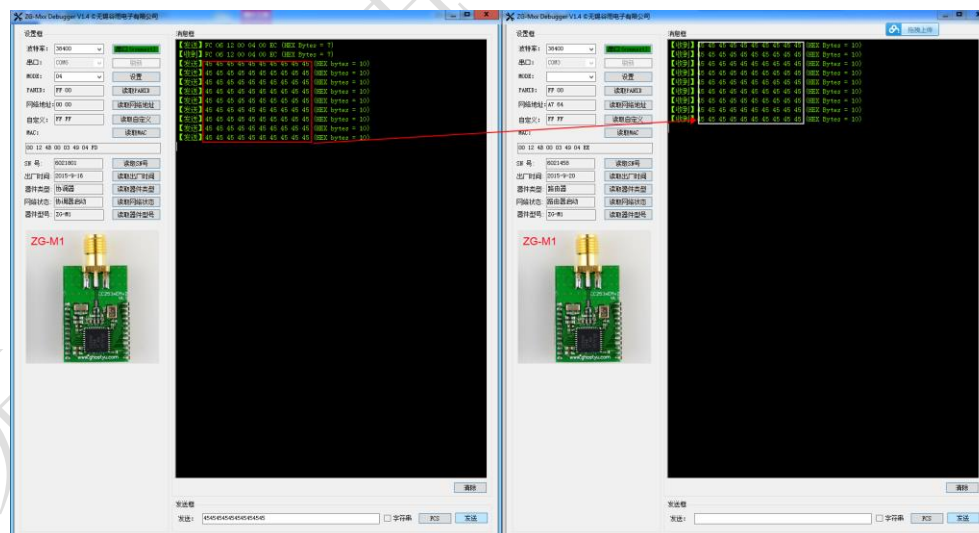
00: 发送方地址的高字节



## 2.4.5 传输模式为 04

### ◇ 数据透传

用 ZG-Mxx Debugger 将其中的一个模块的数据传输方式设为 04。这个与传输模式 0 一样是最基本的数据传输方式，也是模块默认的数据传输方式。如下图所示。



### ◇ 点对点数据传输

用 ZG-Mxx Debugger 将其中一个 ZG-Mxx 模块的数据传输方式设为 04，然后在发送框中采用 FD 发送数据。

例如：FD 02 64 A7 02 03

FD: 点对点数据传输方式的标志

02: 用户数据的长度

64: 目标地址的低字节

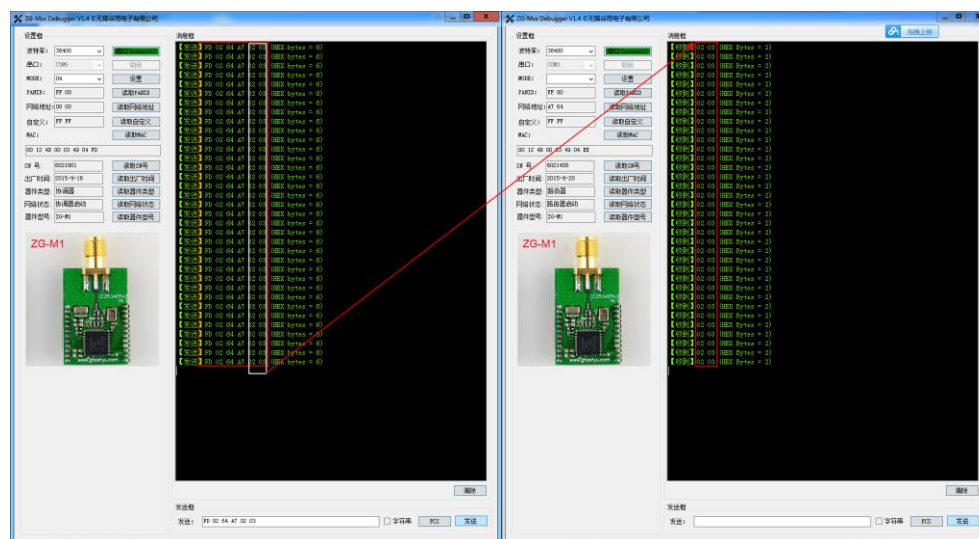
A7: 目标地址的高字节

02: 用户数据

03: 用户数据

接收方 A7 64 的 ZG-Mxx 模块就会收到

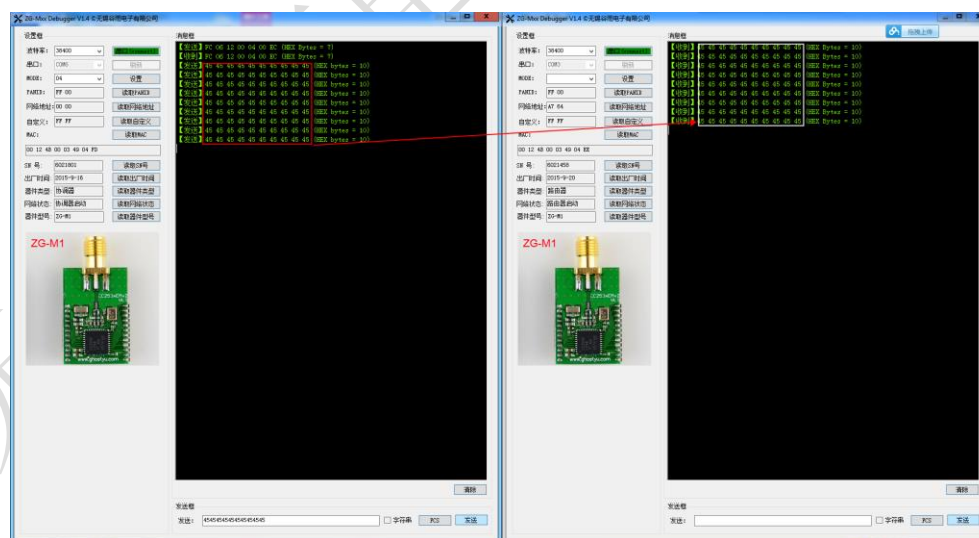
02 03



## 2.4.6 传输模式为 05

### ◇ 数据透传

用 ZG-Mxx Debugger 将其中的一个模块的数据传输方式设成 05。这个与传输模式 0 一样是最基本的数据传输方式，也是模块默认的数据传输方式。如下图所示。



### ◇ 点对点数据传输

用 ZG-Mxx Debugger 将其中一个 ZG-Mxx 模块的数据传输方式设成 05，然后在发送框中采用 FD 发送数据。

例如：FD 02 64 A7 02 03

FD: 点对点数据传输方式的标志

02: 用户数据的长度

01: 目标自定义地址的低字节

00: 目标自定义地址的高字节

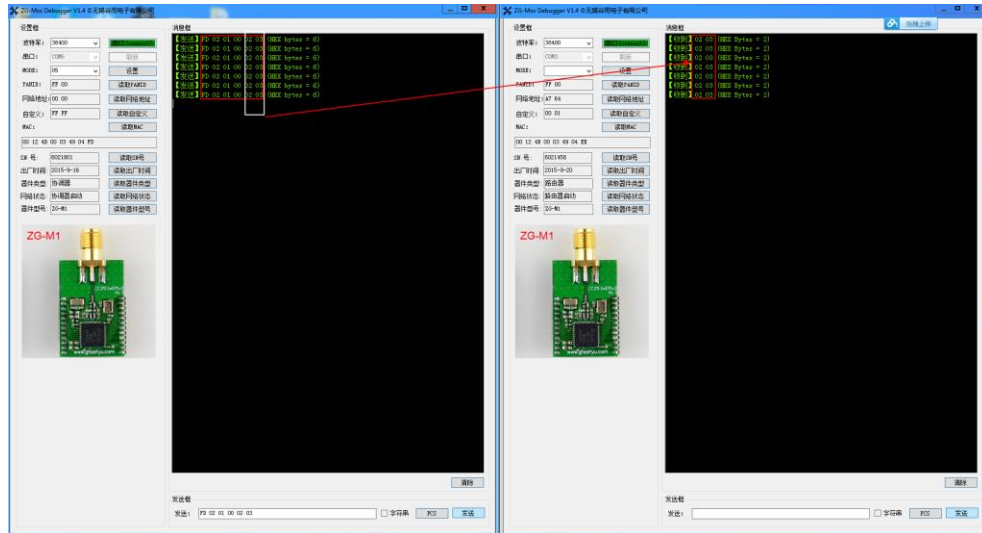
02: 用户数据

03: 用户数据

接收方 00 01 的 ZG-Mxx 模块就会收到

02 03

注: 自定义地址可以通过 ZG-Mxx Setting 来设定。



## 2.5 远程 GPIO 控制

在 ZG-Mxx 模块中,我们加入了可以远程控制各个网络节点的相关 GPIO 的引脚的状态。如果引脚配置成了输出,那么它就可以接收远程的控制,可以设置它输出高低电平,以便达到控制执行器的开关工作。ZG-Mxx V1, V2 版本默认是 P0 端口是输入, P1 和 P2 默认是输出,而在 V3 版本中 P0, P1, P2 都是输入状态, P0 端口不能被设成输出。如果用户想要控制远程节点的 GPIO 高低电平状态,需要将指定的 GPIO 口配置成为输出(可以通过 ZG-Mxx Setting 来配置,具体参考 ZG-Mxx Setting 软件操作说明书),且默认为高电平。在远程控制 GPIO 口功能,加入了等待重传的机制。其相关的指令如下所述。

例如: FC 03/06 14 00 xx yy zz AA BB CC XOR

FC : 设置命令的帧头

03/06: 功能码。03 是读, 06 是写

14 00: 命令ID, 表示此命令的功能, 低字节在前

XX : 表示指令地址的类型, 00为自定义地址, 大于00为ZIGBEE短地址

yy zz: 表示目标的地址。低字节在前, 高字节在后

AA: 表示端口号

BB: 表示引脚, 一个引脚占一位。例如0引脚就是0x01, 1脚就是0x02, 2脚就是0x04

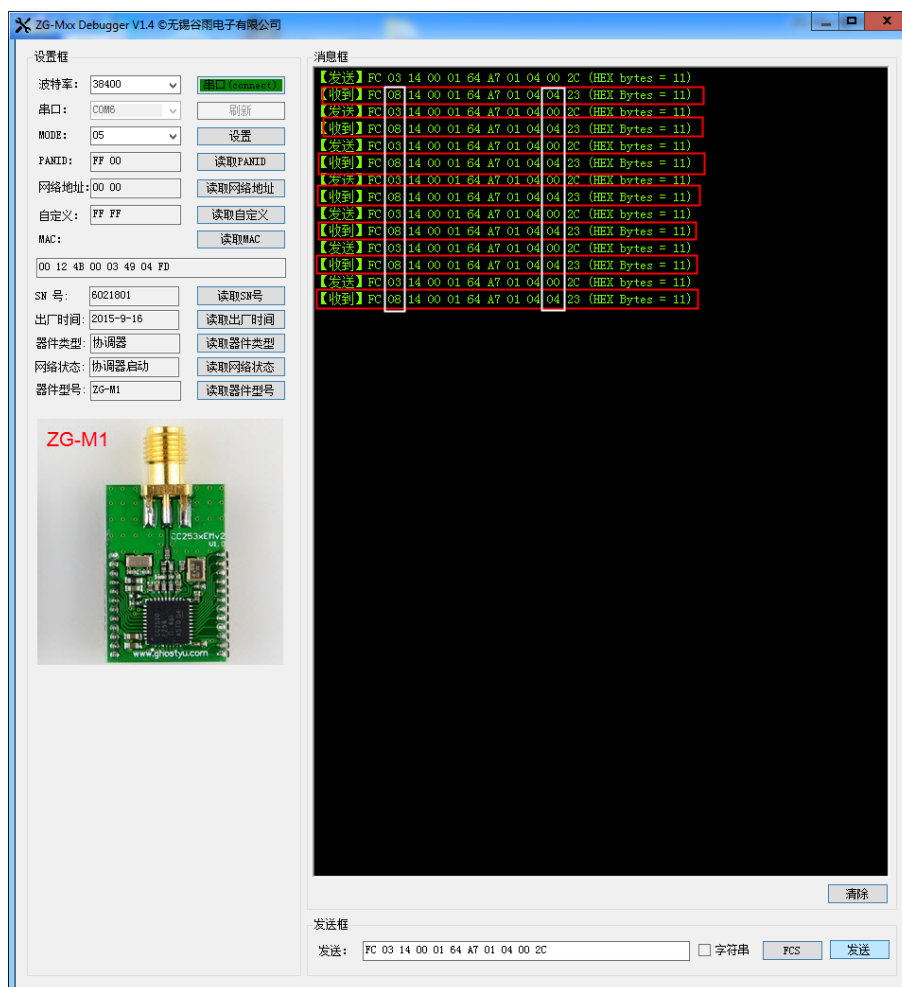
CC: 表示设置引脚输出的高低电平值。其值与引脚位对应。如果是读状态, 其值无效。

XOR: 校验码。

✧ 对网络地址为 A7 64 的 Zigbee 模块 P1.2 引脚进行读取

协调器发送: FC 03 14 00 01 64 A7 01 04 00 2C





返回 FC 08 14 00 01 64 A7 01 04 04 23

其中 FC 后的 08 表示正确返回，如果出现 04 表示读取超时。而引脚状态信息在倒数第二个字节中。本例中是 04 表示，表示当前那个读取引脚的状态为逻辑 1。

- ✧ 对网络地址为 A7 64 的 Zigbee 模块 P1.2 引脚进行高低电平控制

协调器发送：FC 06 14 00 01 64 A7 01 04 00 29

协调器会收到：FC 08 14 00 01 64 A7 01 04 00 27 表示设定正确，用户可以用万用表或显示器测量一下相应引脚的电压是否发生变化。

协调器也有可能收到：FC 04 14 00 01 64 A7 01 04 00 2B 表示设定不成功，出现超时。

## 2.6 ZG-Mxx 远程 ADC 读取

在ZG-Mxx模块中，我们加入了可以远程采集各个网络节点的ADC和可以采集本地的ADC。其相关的指令如下。

FC 03 17 00 XX YY ZZ AA BB CC X0R

FC：设置命令的帧头

03：功能码。03 是读

17 00：命令ID，表示此命令的功能，低字节在前。十六位数据是0x0017

XX :表示指令地址的类型, 00为自定义地址, 大于00为ZIGBEE短地址

YY ZZ : 表示目标的地址。低字节在前, 高字节在后

AA: 表示端口号。只能是00

BB: 表示引脚, 一个引脚占一位。P0端口的引脚0-3被系统占用。所以只能是0x80, 0x40, 0x20, 0x10其中的数值。

CC: 00

XOR: 校验码

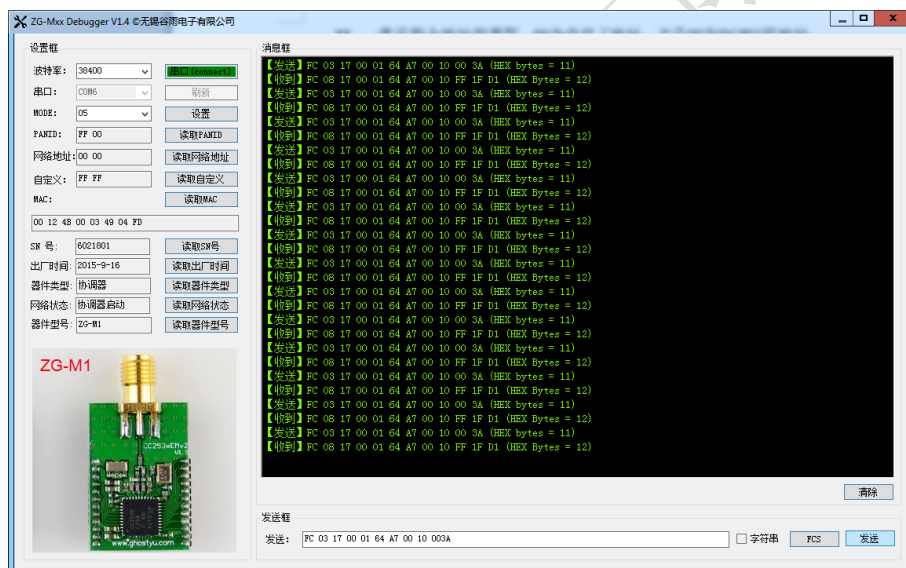
如果接收者接收到了远程节点的ADC采集请求, 会处理相应的ADC引脚, 其ADC 引脚必须在P0口。如果这个GPIO口无效, 将返回无效操作响应。如果命令有效, 接收者将会采集相应引脚的模拟输入, 并将结果返回。再如果远程端在1秒内没有响应, 发送者会尝试重传, 如果连续三次都没有响应, 将会返回超时响应信息。ADC的参考电压是其工作电压, 满量程的ADC 数值是4095, 即0xFFF。这里计算原始信号的值的公式为:  $x=3.3 \times y/4095/(V)$ , 假设这里的参考电压为3.3V, ADC的值为y。

接收者正常响应: FC 08 17 00 XX YY ZZ AA BB c1 c2 XOR

C1 C2:就是ADC的值, 低字节在前, 高字节在后。即十六位数据表示为0XC2C1。

接收者无效响应: FC 88 14 00 XX YY ZZ AA BB CC XOR

响应超时: FC 04 14 00 XX YY ZZ AA BB CC XOR



## 2.7 网络打开与关闭

(此功能只能在V3版本的模块中使用)

在V3版本中, 系统中占用了P0.1引脚, 作为打开或关闭网络的输入引脚。其输入为下降沿有效。模块网络默认是打开的, 如果有户想要将网络关闭, 可以通过模块的P0.1引脚输入一个下降沿信号。如果网络关闭, 则模块的P0.0引脚就会输出高电平, 以指示网络的状态。用户可以在P0.0引脚上接一个指示的led, 可以方便观察。这个关闭网络的功能引脚只能在协调器与路由器中存在, 在终端节点中不存在这个功能说明。设置之后, 模块会记住当前的设定, 下次上电会继续保持。

在打开或关闭网络功能, 除了通过手动在P0.1上产生一个下降沿以外, 还可以通过串口指令完成此功能。用户可以通过串口向模块发送一个FC 06 1D 00 00 00 XOR来关闭网络, 用

户也可以向串口发送一个FC 06 1D 00 01 00 XOR 来打开网络。当网络打开后，其它的节点就可以加入这个网络，否则其它节点是不能加入这个网络的。

协调器与路由器对打开或者关闭网络，产生不同的操作行为。当协调器接收这个操作时，不仅会在自己模块内产生结果，还会将此操作广播到网络中其它路由节点，使网络内其它节点产生相同的操作，实现网络内同步。而对于路由器则只能在本地操作，不会向网络中的其它节点发送操作信息。

用户可以测试一下，在网络打开，与网络关闭状态，新的网络节点会不会加其中。

## 2.8 联系我们

无锡谷雨电子有限公司

戚二进 tel:151-6166-5245

公司网址: <http://www.wx-iot.com>

官网店铺: <http://ghostyu.taobao.com>