
无锡谷雨电子有限公司

ZG-Mxx 模块使用说明

www.wx-iot.com

2015/6/17

Start

| 版本记录 | 更改记录 | 撰写人 |
|------|--|-------------------|
| A | 初始化版本 | 戚二进 2013-12-05 |
| B | 1. 增加安全网络功能同时可修改密钥 2. 增加远程可控 IO 3. 增加远程读取 ADC 4. 可修改工作信道（11-26） | 戚二进 2015-04-28 |
| C | 1. 协调器、路由器与终端三者合而为一 2. 增加以新的网络环境去启动的功能与说明 3. 增加了测电量的指令（终端节点测电量才意义） | 戚二进 2015-06-17 |
| | | |

目录

| | |
|------------------------------|----|
| 1 前言..... | 3 |
| 2 ZG-Mxx 系列功能介绍 | 5 |
| 3 ZG-Mxx 基本概念 | 7 |
| 3.0 Zigbee 网络的节点 | 7 |
| 3.1 ZG-Mxx 模块组成网络 | 9 |
| 3.2 ZG-Mxx 网络组成及设置 | 9 |
| 3.3 更改网络 PANID 影响..... | 12 |
| 3.4 更改网络 ExtPANID 影响..... | 12 |
| 3.5 更改网络信道影响..... | 12 |
| 3.6 可视化 ZG-Mxx 网络结构 | 12 |
| 3.7 ZG-Mxx 模块数据传输功能 | 15 |
| 3.8 ZG-Mxx 远程 GPIO 控制 | 33 |
| 3.9 ZG-Mxx 模块网络打开与关闭功能 | 34 |
| 3.10 ZG-Mxx 模块网络环境清除 | 34 |
| 3.11 ZG-Mxx 远程 ADC 读取 | 35 |
| 3.12 ZG-Mxx 终端数据发送与接收 | 35 |
| 4 ZG-Mxx 系列设置指令描述 | 36 |
| 5 ZG-Mxx 可视化设置 | 47 |
| 6 ZG-Mxx 模块常见问题解答 | 47 |
| 联系我们: | 48 |

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。所以它们具有 Zigbee 协议的全部特点。



图 1 ZG-M0



图 2 ZG-M1

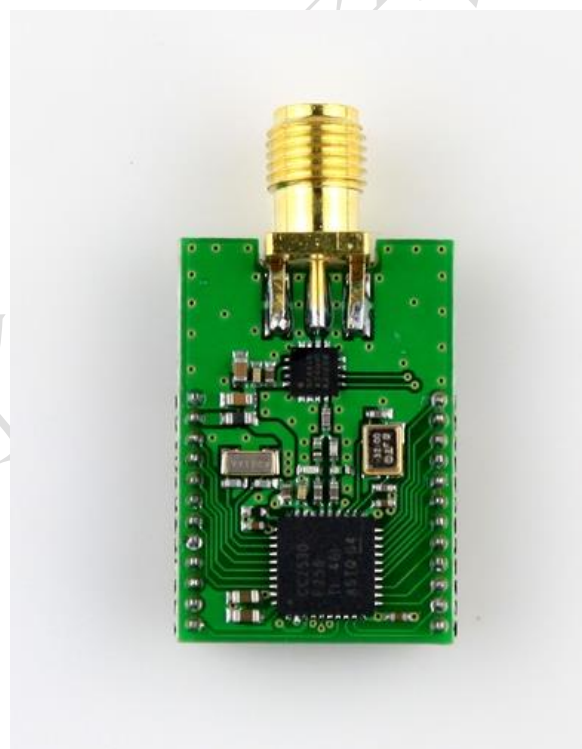


图 3 ZG-M1E

在 PC 端我们也推出了 ZG-8000 的 zigbee 设备，如图 4 所示。ZG-8000 在使用时，可将其插在 PC 端的 USB 插口中，只要 PC 端按装的了我们提供的驱动，就会在 PC 端形成了个串口。用户只要通过串口就可以向 zigbee 网络发送数据和收集数据。如果用户有在上位机使用 zigbee 方面的需求，ZG-8000 将是您不二的选择。



图 4 ZG-8000

我们推出这些模块，它们都具有上电自动寻找相应 PANID 网络，并自动加入的网络的功能，即自动组网。前提条件是在第一次上电时，网络中存在相应的 PANID 网络（如果此模块被配置成协调器，则不需要这个要求）。在上电并加入相应网络后，用户只要向模块的串口发送数据即可。我们的模块会自动将接收到的数据向目的地传输。在此期间用户不需要了解复杂的 Zigbee 协议，所有的工作都是我们的模块帮您完成。您要做的就是从串口接收数据和发送数据。这样您就可以将自己的主要精力放在您的数据处理上，而不用关心数据的无线传输。如果您身边的项目有这样的需求，我们的 ZG-Mxx 系列模块将是您不二的选择。

2 ZG-Mxx 系列功能介绍

如果用户要了解，ZG-Mxx 模块的

在继之前 ZG-Mxx 模块使用经验，我们一贯坚持提供简单易用的 Zigbee 数据透传模块。在此思想指导下，我们将具有协调器，路由器与终端功能合而为一，任由用户自己决定使用何种功能。由此带来的方便不言而喻，为此用户只要输入简单的七个串口数据，就可在 V3 软件版本的 ZG-Mxx 模块进行切换。V2，V1 只能在协调器与路由器上进行切换。所以用户在使用之前，一定要认识清楚。

ZG-Mxx 系列模块使用非常简单，简单到你可以像串口一样使用它。你可以称它为“无线串口”。所以你只要会使用串口，就可以使用我们的 ZG-Mxx 系列模块。

简单易用：您可以像串口一样使用我们的模块，来传输您的数据，完成您的无线传输。我们的模块会按照指定的模式向数据送到目的地。在此期间，您可以完全不了解 Zigbee 协议。

自动组网：所有的 ZG-Mxx 模块上电即自动组网，特别是 ZG-Mxx 模块在每一次上电期间，在模块区域内一定要有相应的协调器网络。（可以是作为协调器的 ZG-Mxx 模块，或是已经加入过网络的 ZG-Mxx 模块）。上电完成后，其自己的父设备会自动给自己分配网络地址，不需要手动分配网络地址。其网络发现，网络加入和相应应答等 Zigbee 组网流程都是 ZG-Mxx 模块自动完成。

简单数据传输：简单数据传输可以有两种方式。第一种为串口数据透传方式，第二种为网络内任意节点间数据传输。串口数据透传是 Coordinator(协调器)从串口接收到的数据会自动发给所有的节点，某个节点从串口接到的数据会自动发给协调器。这种方式协调器就

像是网络的集线器。网络内任意节点间数据传输,是为了实现网络内点对点的数据传输。这样就可以和串口数据透方式形成互补共同为用户服务。

唯一 MAC 地址: ZG-Mxx 系列模块采用 TI CC2530F256 芯片,芯片面出厂时已经自带 MAC 地址,用户无需另外购买 MAC 地址。只要芯片的 MAC 地址没有经过人为的修改,它将是全球唯一的,所以 MAC 地址可以作为 ZG-Mxx 模块的标识。

节点类型可更改: 用户可以通过串口向 ZG-Mxx 模块发送相应的指令,更改节点在网络中的类型(协调器或路由器(Router))。为了方便用户根据自身情况进行更改,我提供了 App (ZG-Mxx Final) 进行方便的设置。

网络地址可更改: 用户可以通过串口向 ZG-Mxx 模块发送相应的指令,更改节点在网络的网络地址,这样方便用户标识自己有意义的节点。但网络的协调器的网络地址是固定的不可更改。如果用户尝试更改,则会 error 伺候。为了方便用户根据自身情况进行更改,我提供了 App (ZG-Mxx Final) 进行方便的设置。

自定义节点地址: 用户可以通过串口向 ZG-Mxx 模块发送相应的指令,更改 ZG-Mxx 模块自定义地址,这样方便用户标识自己有意义的节点,ZG-Mxx 模块自定义地址默认为(0xFFFF)。在自定义节点地址功能,没有任何的限制。用户可以随心所欲进行更改。为了方便用户根据自身情况进行更改,我提供了 App (ZG-Mxx Setting) 进行方便的设置。

GPIO 方向可控: 用户可以通过串口向 ZG-Mxx 模块发送相应的指令,更改 ZG-Mxx 模块的 GPIO 端口的方向。如果 GPIO 为输入,则可以读取 GPIO 的引脚状态;如果 GPIO 为输出,则可以设置 GPIO 引脚的输出状态。为了方便用户根据自身情况进行更改,我提供了 App (ZG-Mxx Setting) 进行方便的设置。

远程 IO 可控: 用户可以通过其中的一个 ZG-Mxx 模块发送命令,可将网络中的任意一个节点的输出 IO 设置成高低电平,同时也可以读取输入 IO 当前的高低电平状态。且有等待重传机制,可安全的使用。

远程 ADC 采集: ZG-Mxx 模块自身集成了带有 ADC 采集功能。也可以采集本地模拟输入,最大优势是可以采集远程任意节点模拟输入。目前 ZG-Mxx 模块 ADC 采集模拟输入电压在 3.3V 电压范围内,且在 P0.0, P0.1, P0.4, P0.5, P0.6, P0.7 引脚上。且有等待重传机制,可安全的使用。如果在使用时传输超时,模块也会返回信息,提醒超时。

远程电量采集: ZG-Mxx 模块是支持电量的采集。当 ZG-Mxx 作为可以睡眠的终端时,对电量的实时采集是有必要的。当终端用电池供电时,对电量的监控可以帮助我们对电池进行更换。

PANID 可更改: 用户可以根据自己的需要更改节点的 PANID。如果节点类型为路由器更改 PANID 后,会在 1 秒后重启。它将会加入你指定的 PANID 网络。如果节点的类型为协调器,更改 PANID 会将整个网络的 PANID 都会更改,这包括加入网络中的各个节点。为了方便用户根据自身情况进行更改,我提供了 App (ZG-Mxx Final) 进行方便的设置。

信道可更改: 为了满足不同用户的操作,ZG-Mxx 模块的 zigbee 工作信道是可以更改的。且有效数值为 11 到 26。

扩展 PANID 更改: 用户可利用 ZG-Mxx Setting 工具,对 ZG-Mxx 模块进行配置,使其只能加入或者形成你设定的扩展 PANID 的网络。防止加入到别人的网络里。

密钥可更改: 如果用户使用的是安全网络的 zigbee 模块,我们也提供的修改密钥的工具 ZG-Mxx Setting。它可以方便用户进行自己网络密码的更改。

波特率可更改: 用户可以根据自己的需要更改 ZG-Mxx 模块的串口通信波特率。ZG-Mxx 模块出厂默认为 38400, 8, N, 1 格式。目前模块支持 9600, 19200, 38400, 57600, 115200 五种波特率。如果用户的设备可以达到 38400 波特率,强烈建议用户使用 38400 及以上的波特率。

3 ZG-Mxx 基本概念

本节主要将 zigbee 协议中涉及到的一些概念结合 ZG-Mxx 模块给用户简单的介绍一下，方便不了解 zigbee 协议的用户学习一下。

3.0 Zigbee 网络的节点

Zigbee 网络具有三种网络形态节点，Coordinator(协调器)，Router(路由器)，EndDevice(终端节点)。

Coordinator(协调器)：用来分创建一个 Zigbee 网络的一种器件。在网络中它是第一个器件。Coordinator 会扫描空间 RF 环境，它将根据 RF 环境选择一个信道和相应的 PANID，即网络标识，之后将会启动这个网络。如果在同一信道上存在一个相同 PANID 的 Coordinator，先启动的 Coordinator 会保持原来的 PANID，后启动的 Coordinator 会在原来的 PANID 的基础上加 1，以免引起 PANID 冲突。Coordinator 在网络的角色主要是启动与配置网络，一旦它完成启动与配置，它的行为就要是一个路由器。

Router(路由器)：Router 在网络中主要充当以下三个作用。第一是允许其它的节点加入网络，第二是转发数据包，第三是辅助它的孩子节点进行网络通信。通常路由器是由稳定电源供电。当一个网络由一个协调器及多个路由器构成时，这个网络才是一个网状网络。如图 5 所示，其中终端节点没有画出。

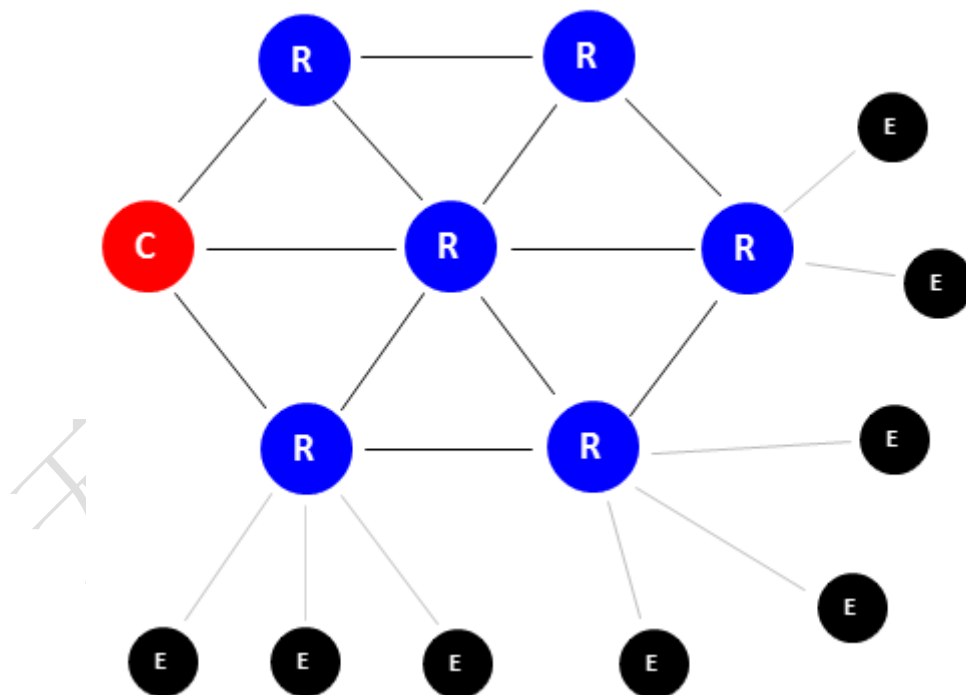


图 5 网状网络

如果一个网络由协调器，路由器和终端节点构成，那么这个网络理论上可以有 65535 个节点，然而实际的应用没有那么多的节点。ZG-Mxx 模块也遵循这个规律。如图 6 所示。

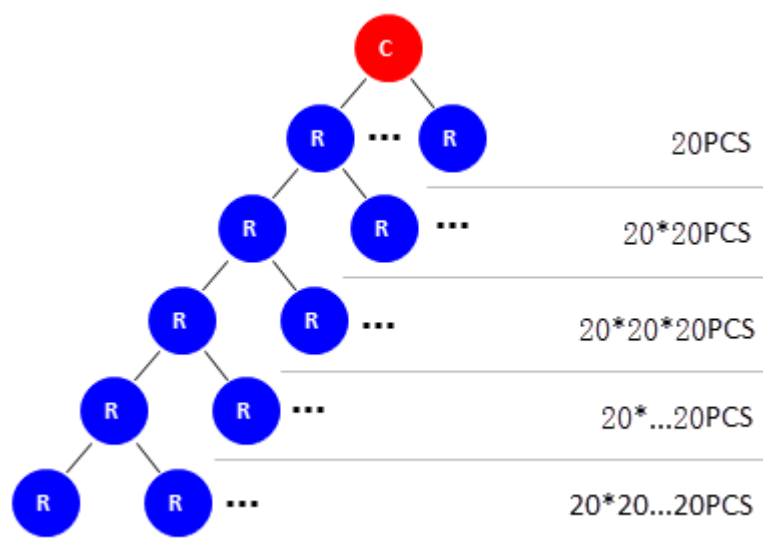


图 6 网络节点

End Device(终端节点): 终端节点对网络的维护没有什么特别的义务，它可以睡眠可以唤醒，因此它可用电池供电。在网络中，它可以发送和接收数据，不能进行数据的转发。在接收到数据会存在一定的延时。对于终端节点可以以协调器或路由器作为自己的父节点。加入网络后，它会定期的轮询父节点，是否有自己的数据。所以，End Device 通常适合接收少量的数据，周期性的发送数据。当 ZG-Mxx 模块配置成终端节点时，模块就会进入低功耗模式。其最低功耗在 1.1uA 到 1.7uA 之间。最大的功耗是在发送数据时，最大为 28mA 上下。



由以上三种 Zigbee 网络节点类型介绍可知，终端节点不适合作为实时数据传输节点。因为终端节点会睡眠，在数据传输方面会有一定的延时。如果要做数据的实时传输，且延时小时，可以采用我们 V1, V2 的软件版本的 ZG-Mxx 模块。如果在功耗方面有要求的，可以使用 V3 版本的 ZG-Mxx 模块。在 V3 版本的终端功耗已经做到十分出色了，在睡眠时的平均电流在 1.1uA 到 1.7uA，如上图所示。

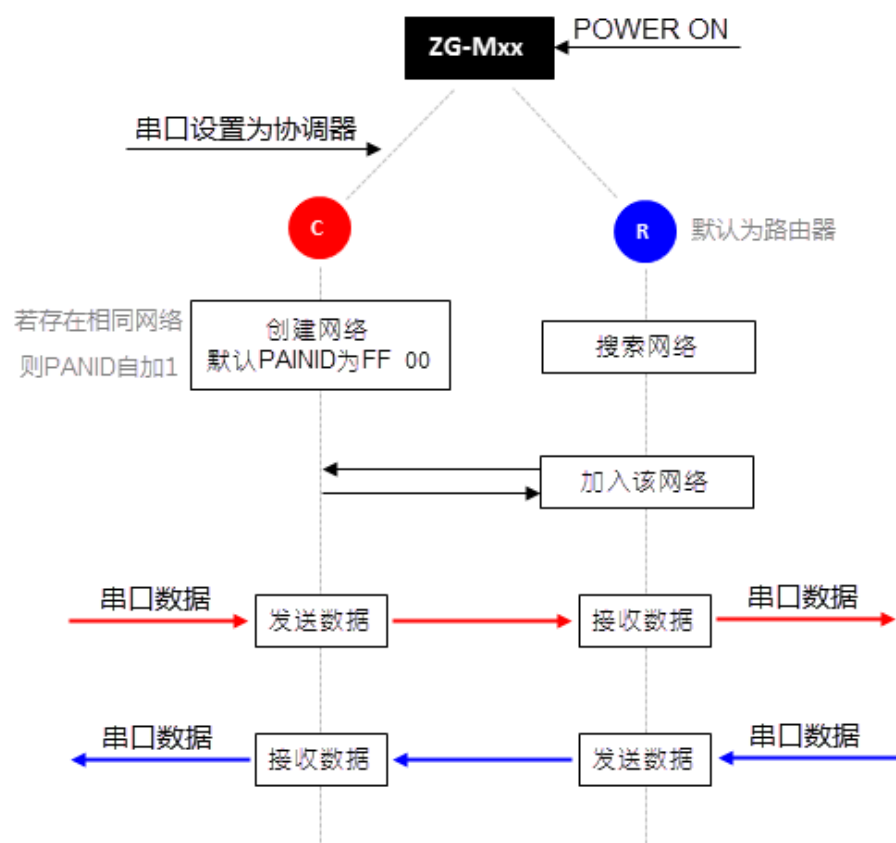
3.1 ZG-Mxx 模块组成网络

ZG-Mxx 模块在网络中只会以协调器或路由器或者终端形式存在。所以它们组成网络后，便形成了一个 MESH(网状)网络。其网络模型见图 5 所示。

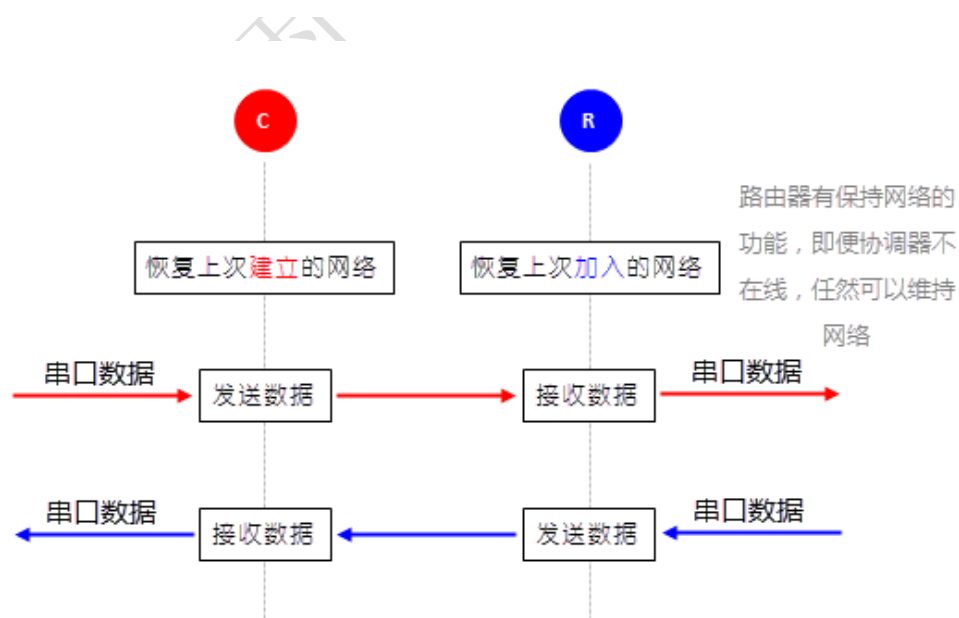
在 MESH 网络中，每个路由节点和协调节点都具有路由器功能，所以它们既能实时收发数据，也能转发数据。同时在网络中，每个路由节点都具有网络保持能力，只要加入网络后，其它节点都能通过该节点加入网络。ZG-Mxx 模块还具有保存网络参数的作用，只要该节点加入过网络，且再次上电可以不需要协调器的存在，但这时串口透传功能将不能使用，只能使用点对点的数据传输。同时在数据传输过程中，路由的计算都是路由器自动完成，在它的内部维护相应的路由表。通过 ZG-Mxx 进行的数据传输，用户不需要关注这些 Zigbee 网络的东西，用户只要将要发送的数据，通过串口向 ZG-Mxx 发送即可。

3.2 ZG-Mxx 网络组成及设置

ZG-Mxx 要组成一个 Zigbee 网络，最少需要二种网络类型节点，即协调器与路由器或终端节点。协调器创建的 PANID 与路由器要加入的 PANID 必须相同，否则它们属于不同的网络器件，路由器不会加入协调器创建的网络。下面两图分别说明模块初次上电的时序和加入网络后再次上电时序，以帮助用户了解 ZG-Mxx 模块工作机理。



【模块第一次上电示意图】



【模块再次上电示意图】

3.2.0 ZG-Mxx 模块出厂设置

ZG-Mxx 模块出厂时，模块上电默认为路由器功能。但模块可以配置成协调器或者终端节点，当然也可以再返回成为路由器。模块 PANID 默认为 0xFF00，信道为 11，自定义地址为 0xFFFF，串口参数为 38400，8，N，1；所有没有被占用的 GPIO 口 P0 是输入状态，且具有 ADC 采集功能，不能设置成输出。P1 和 P2 默认也为输入状态，可以配置成输出。P0 口可以完成远程 ADC 采集功能。而 P1 和 P2 可以完成远程 GPIO 状态读取，和远程 GPIO 输出控制功能。

3.2.1 ZG-Mxx 模块配置

1. 将其中的一个 ZG-Mxx 模块配置成协调器，来创建网络，其 PANID=0xFF00。为了方便用户对 ZG-Mxx 模块进行设置，我们提供了相应的 PC 端 APP 来帮助大家轻松完成你想要的配置。其操作的界面如图 7 所示。第一步就是连接 ZG-Mxx 模块，在图 7 的界面中选择相应的串口，选择相应的波特率，然后点击后面的串口按钮。如果在应串口上有 ZG-Mxx 模块，按钮就会变成绿色背景。如图 7 中 1 所示。接着就是用户根据自己的需要配置相应的参数。其中更改器件类型，PANID，都会引起 ZG-Mxx 模块的重启。如果对这些参数设置成功，ZG-Mxx 模块将会在 1 秒钟后重新启动。关与 ZG-Mxx Setting 使用说明，请参考该软件工具的使用说明。



图 7 ZG-Mxx V1.3 设置界面图

2. 将其它模块设成路由器，PANID = 0XFF00。
用户可以根据图 7 所示的界面对连接在 PC 的串口上的 ZG-Mxx 模块进行相应的设置，即可完成相应的操作。
3. 设置成路由器之后，如果其还没有加入网络，则其网络地址为 FF FE，父地址为 FF FF，父 MAC 地址为 0000000000000000，如图 7 所示。其底板上的绿灯会在闪烁，表示其正在寻找网络。而图 7 中的 PANID 一项显示其想要加入网络的网络标识。如果其成功加入网络，底板上的绿灯就会 2s 闪烁一次。而对于协调器而言，其本身就是创建网络的设备，所以只要 ZG-Mxx 模块是协调器，其底板上的绿灯就会闪烁。
4. 对于终端的 ZG-Mxx 模块，它的定时唤醒时间间隔默认为 5 秒。用户可以在图 7 的界面中观看到。当然用户也可以通过其后的更改按钮，更改默认的时间间隔。这个时间间隔最大为 65 秒。

3.3 更改网络 PANID 影响

- 如果要将某个网络中的 Router 加入到另一个网络，则只需要将这个节点的 PAN ID 改为另一个网络的 PAN ID（或 0x FF FF，一般应用不要设置成 FFFF），重启这个节点，则这个节点会自动加入网络。
- 当一个网络正常运行时，如果更改协调器的 PANID，可能会引起协调器离开网络。我们强烈的不建议在运行过程中更改 PANID，或者扩展 PANID。

3.4 更改网络 ExtPANID 影响

ExtPANID 是 panid 的 8 字节表示。Panid 是 16 位数据，可以表示的网络个数要相对 8 个字节长度的 ExtPANID 来说，是少了很多。在 ZG-Mxx 模块里，可以设定模块欲加入或产生 ExttPand 的网络标识。对于设定为协调器的，可以形成指定的 ExtPANID 网络标识，而对于路由器只能加入你指定的 ExtPanid 的网络标识。ZG-Mxx 模块默认的产生与形成的网络标识是 0000000000000000，即可以加入 PANID 标识相同的网络里，extPANID 没有启动过滤的作用，协调器就会以自己的 MAC 地址形成 ExtPANID 网络标识。

3.5 更改网络信道影响

Zigbee 协议规定了 Zigbee 工作的信道范围为 11-26，即 2.4G 频率附近。ZG-Mxx 默认工作的信道为 11 信道。更改信道，会使 ZG-Mxx 工作到其它信道上，从而会离开当前的网络，强烈建议不要在正常工作中修改信道。

3.6 可视化 ZG-Mxx 网络结构

为了给用户提供一个可视化的网络结构，ZG-Mxx 模块支持 ZTopology 软件，ZTopology 软件是在 TI 的 Sensor monitor 的基础上进行修改的。其外观与 Sensor monitor 相像。件来观看 ZG-Mxx 网络结构。其 TI Sensor Monitor 软件，用户可以在 TI 的官网上下载，也可以我

们的网盘上下载，其下载链接 <http://pan.baidu.com/s/1dDtjUbj>。

- V3 版本请使用 ZTop 工具，V1，V2 版本请使用 ZTopology。以下以 V3 版本为例。下载并安装 ZTop 软件。成功安装软件后，将配置成协调器的 ZG-Mxx 模块，连接到 PC 端的串口上。此时打开 ZTopology 软件版本为 X-3，其整个界面如图 8 所示。

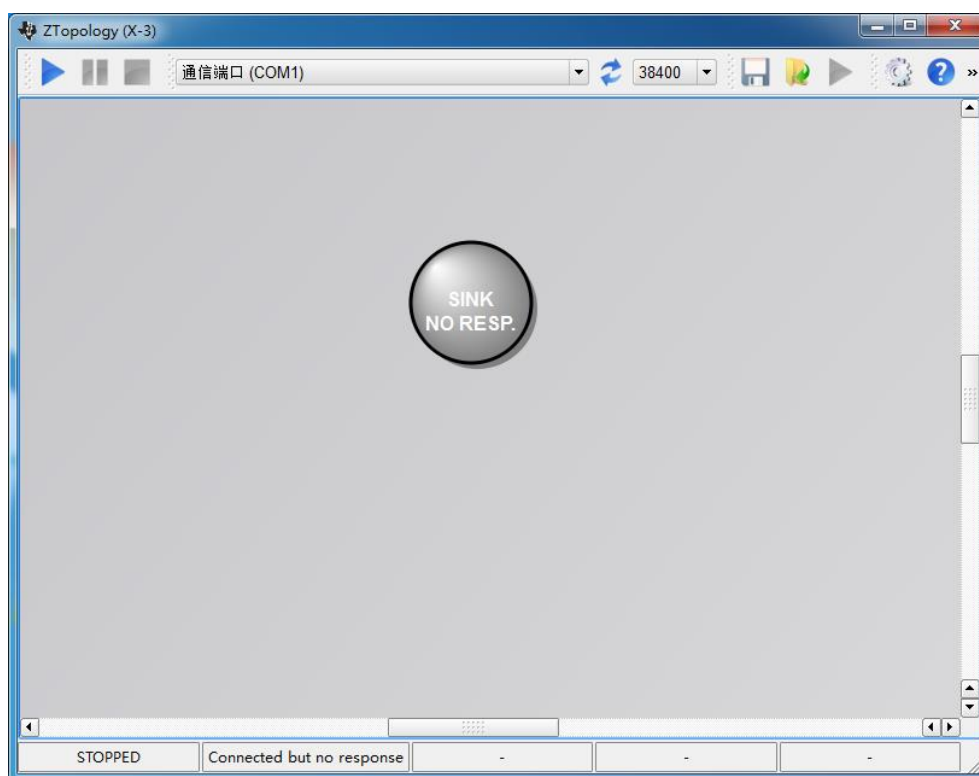


图 8 ZTopology 界面

- 在 ZTopology 软件界面中，串口选择框上选择协调器的串口号。然后点击开始按钮，此时 ZTopology 中的圆圈变成暗红色，表示协调器已经在线。如图 9 所示。

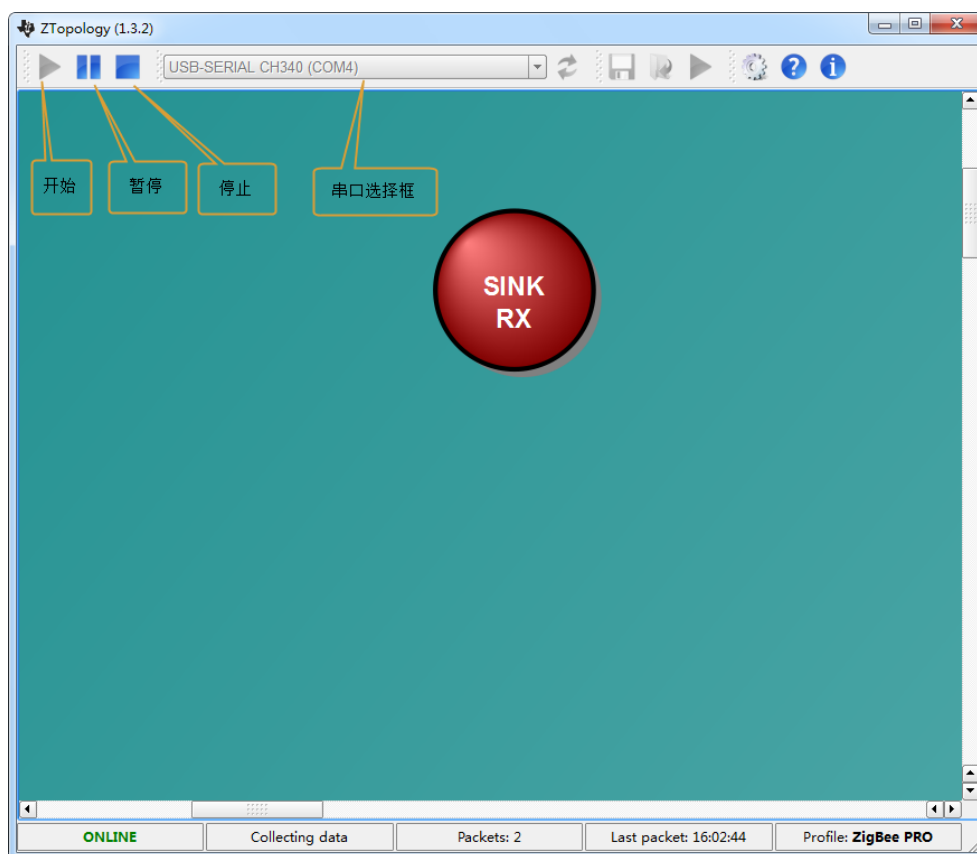


图 9 ZTopology(X-3)

- 在图 9 的界面中，按下方形的停止按钮，使 ZTopology 处于停止状态，其效果如图 8 所示。接下来将每个路由器依次上电，直到每个路由器都正常的加入网络，即 x 绿色灯 2s 闪烁一次。接着按下 ZTopology 界面中的开始按钮。此时 ZTopology 的界面如图 10 所示。

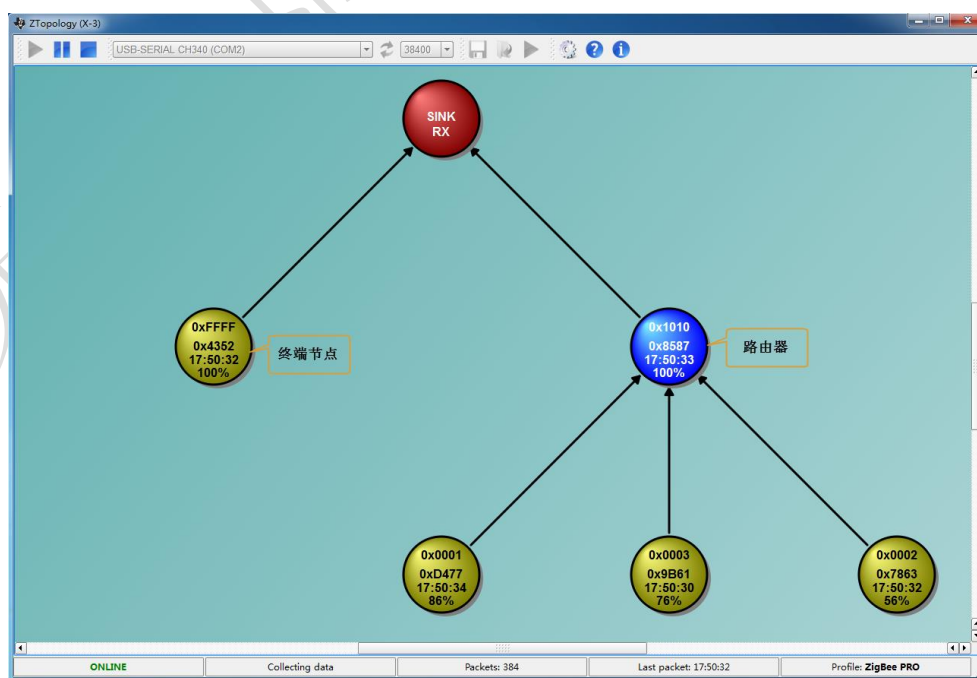


图 10 两个路由器 ZTopology

- 同理，其他节点也是同样的操作。如果用户在 ZTop 运行期间，上电其他的 ZG-Mxx 路由器或终端模块，等待其加入网络后，停止 ZTopology，然后单击开始按钮即可。
- 在 ZTop 中，我们可以看到一个节点的相关信息。其中包括自定义地址，zigbee 的网络地址，还有一个节点的电量，如图 11 所示。

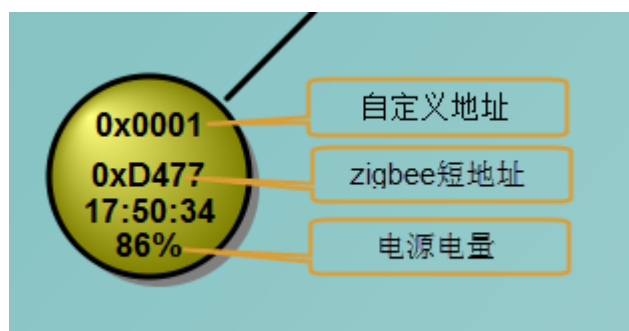


图 11 ZTop 的节点信息

- Zigbee 终端节点在失父节点时，会自动寻找其它的父节点。比如将图 10 中的协调器关闭，其孩子节点就会以 rejoin 的方式加到旁边的路由器中，在此过程中 zigbee 短地址与自定义地址保持不变。如图 12 所示。

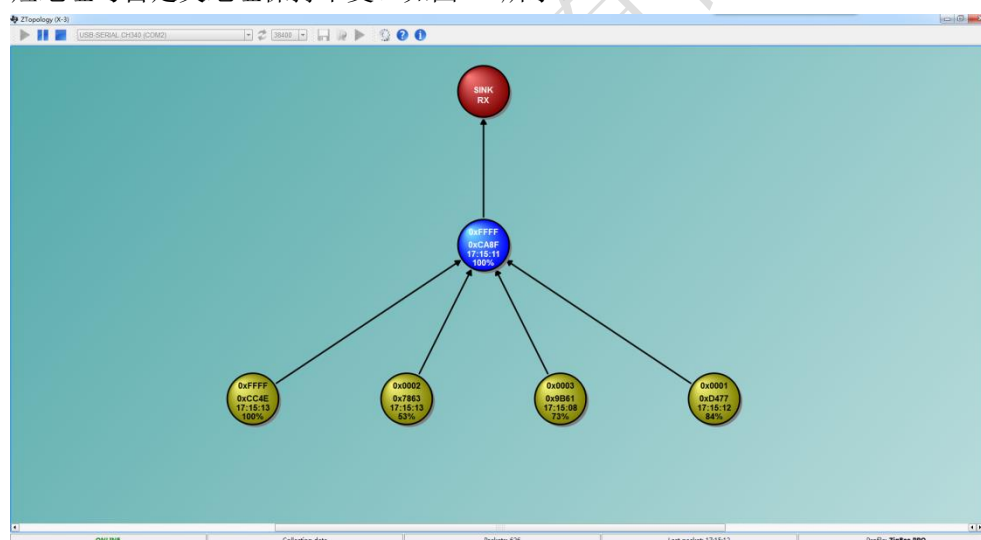


图 12 寻找父节点

- 使用 ZTopology 查看网络结构结束后，一定要退出，以便恢复串口，让其输出干净的数据的用户数据。

3.7 ZG-Mxx 模块数据传输功能

ZG-Mxx 模块拥有简单易用的数据输功能。按其传输的方式来分可为两种：① 串口透传 ② 点对点传输。按具体的指令格式来分可为七种数据传输。

- **数据透传**

只要发送数据的第一个字节不是 0xFE, 0xFD, 0xFC，则自动进入串口透传模式。协调器从串口接收到的数据，会自动发送给所有在网络中的节点；网络中某个节点从串口接收到数据，会自动发给协调器。

- **数据透传+网络短地址**

在数据透传的基础上，通过对发送模块的设置。发送模块在发送数据时将自己的网络短地址附加在数据的末尾，接收模块会收到的数据会多出 2 个字节。这个 2 个字节就是发送模块的网络地址，低字节在前，高字节在后。**使用此模式，发送数据长度必须限制在 80 个之内（包括 80 个）。**

- **数据透传+MAC 地址**

在数据透传的基础上，通过对发送模块的设置。发送模块在发送数据时将自己的 MAC 地址附加在发送数据的末尾，接收模块会收到的数据会多出 8 个字节。这个 8 个字节就是发送模块的 MAC 地址，低字节在前，高字节在后。**使用此模式，发送数据长度必须限制在 80 个之内（包括 80 个）。**

- **数据透传+自定义地址**

在数据透传的基础上，通过对发送模块的设置。发送模块在发送数据时将自己的自定义地址附加在发送数据的末尾，接收模块会收到的数据会多出 2 个字节。这个 2 字节的的就是发送模块的自定义地址，低字节在前，高字节在后。**使用此模式，发送数据长度必须限制在 80 个之内（包括 80 个）。**

- **点对点数据传输方式，用 Zigbee 网络地址寻址**

Zigbee 网络内任何节点间，都可以实现点对点数据传输。在发送方在数据的末尾附加发送方的网络短地址。接收方会多接收到 2 个字节的数据。**使用此模式，发送数据长度必须限制在 80 个之内（包括 80 个）。**

- **点对点的传输方式，用 Zigbee 短地址寻址，去掉包头包尾**

Zigbee 网络内的任意节点之间，可通过点对点传输数据。发送方根据串口发来的数据，从指定位置获取目的节点的地址，然后将数据发给目的节点。而接收方在收到数据时，将此帧的头和尾都去掉，只向串口发送有效的数据。**使用此模式，发送数据长度必须限制在 80（包括 80）。**

- **点对点的传输方式，用自定义地址寻址，去掉包头包尾**

Zigbee 网络内的任意节点之间，可通过点对点传输数据。发送方根据串口发来的数据，从指定位置获取目的节点的地址，然后将数据发给目的节点。而接收方在收到数据时，将此帧的头和尾都去掉，只向串口发送有效的数据。**使用此模式，发送数据长度必须限制在 80 内（包括 80）。**

3.7.1 数据透传

数据透传是 ZG-Mxx 模块最基本的数据传输功能，也是最重要数据传输方式。

在发送数据时，只要发送数据第一个字节不是 0xFE, 0xFD, 0xFC，则自动进入数据透传方式。在这里推荐用户在发送的数据之前加上一个不是 0xFE, 0xFD, 0xFC 的一个字节，如 0xFF。这样便会防范要发送数据第一个字节可能是 0xFE, 0xFD, 0xFC 而导致数据传输失败。

ZG-Mxx 在这个模式下，协调器从串口收到的数据，会向网络中所有节点发送数据。某个节点从串口收到的数据会自动发给协调器。这样网络内任意路由器与协调器之间，好像有一根串口线连接。

在数据透传模式下，最大数据包长度不能超过 80 个字节。但建议每个数据包应在 32 个字节内。在图 13 中，展示数据透传协调器发送模型；图 14 中，展示了数据透传协调器接收模型。

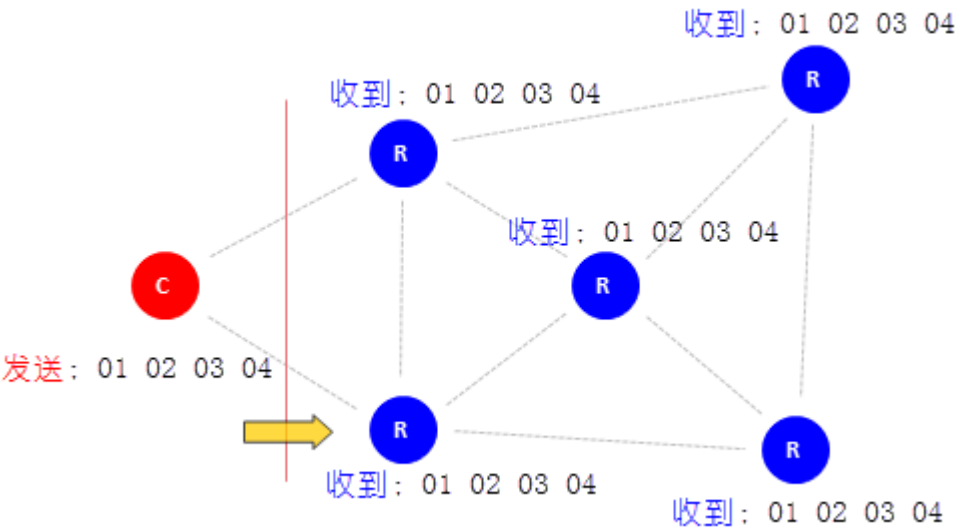


图 13 数据透传发送模型

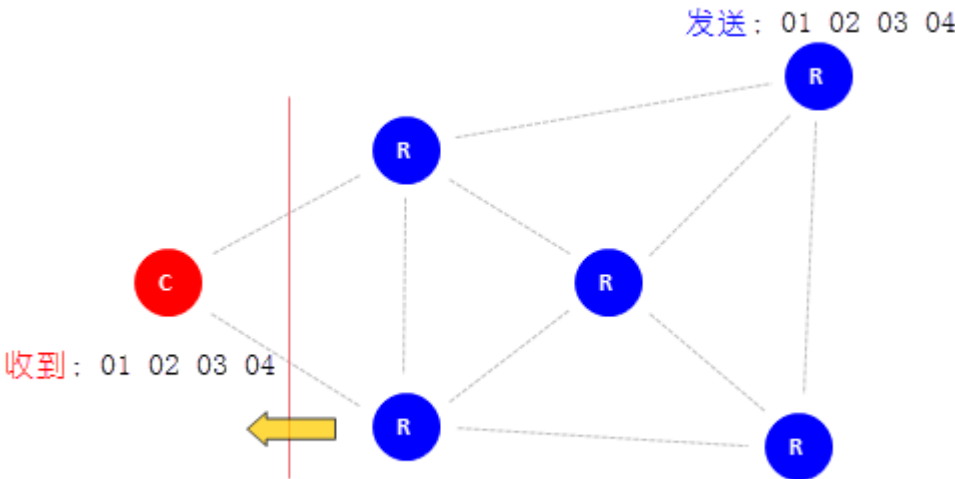


图 14 数据透传接收模型

数据透传模式下数据传输测试记录:

| 数据透传方向 | 数据包长度（字节） | 最快速率（ms） |
|--------------|-----------|----------|
| 路由器->协调器（单播） | 16 | 20 |
| | 32 | 50 |
| | 64 | 70 |
| | 80 | 100 |
| 协调器->路由器（单播） | 16 | 100 |
| | 32 | 100 |
| | 64 | 100 |

| | | |
|---|-----|-------|
| | 80 | 100 |
| | >80 | 不支持传送 |
| <p>测试条件：</p> <ol style="list-style-type: none">1. 室温，实验室条件2. ZG-Mxx 模块间距离 2 米，信号良好3. 传输深度 24. 串口波特率 38400，8，N，15. 连续发送，接 10K 字节，无误码，连续测试 10 次6. 测试软件：串口调试助手 SSCOM3;ZG-Mxx Debugger(我们自己的软件，免费提供) | | |

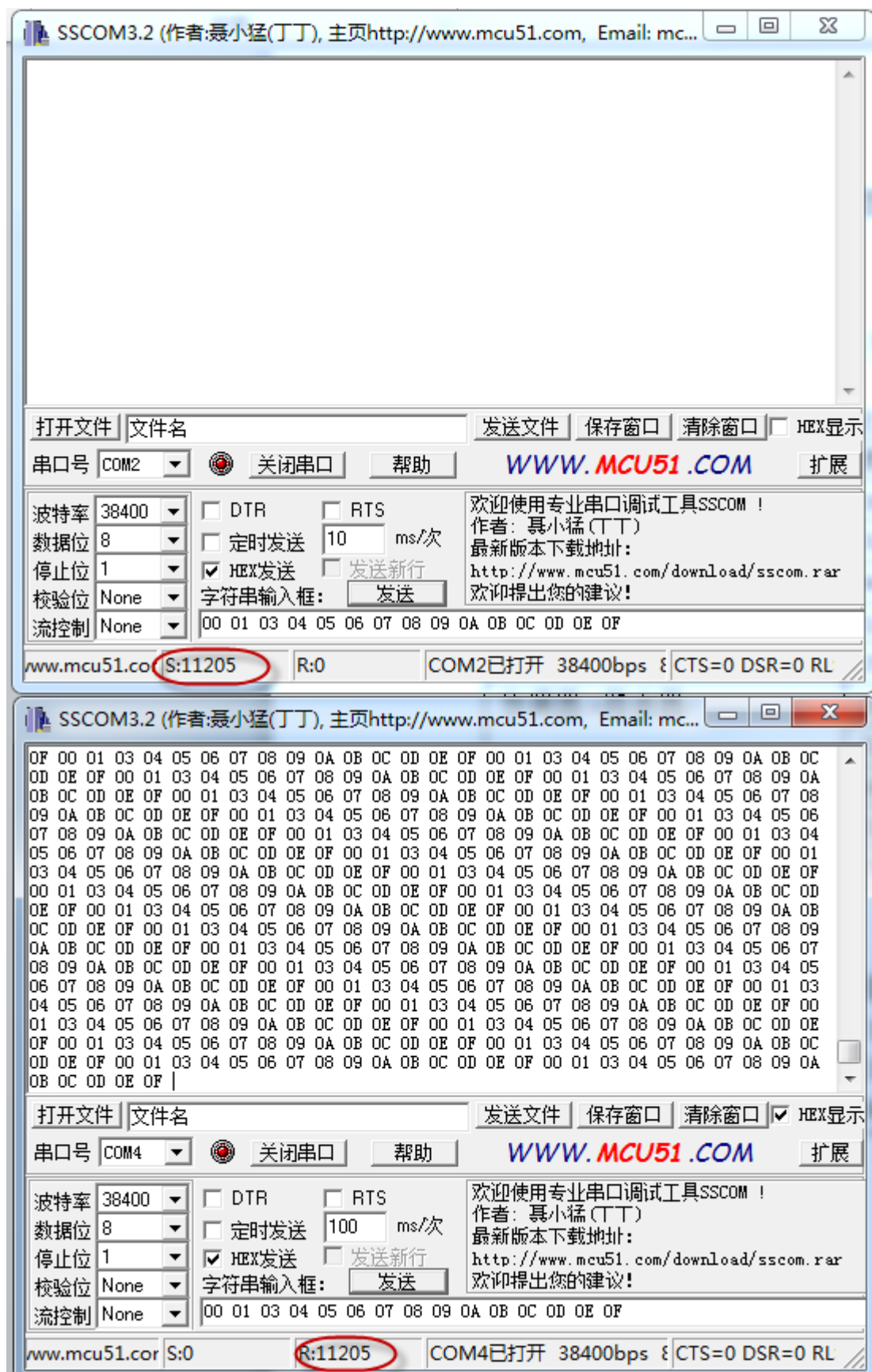


图 15 路由器到协调器数据传输测试

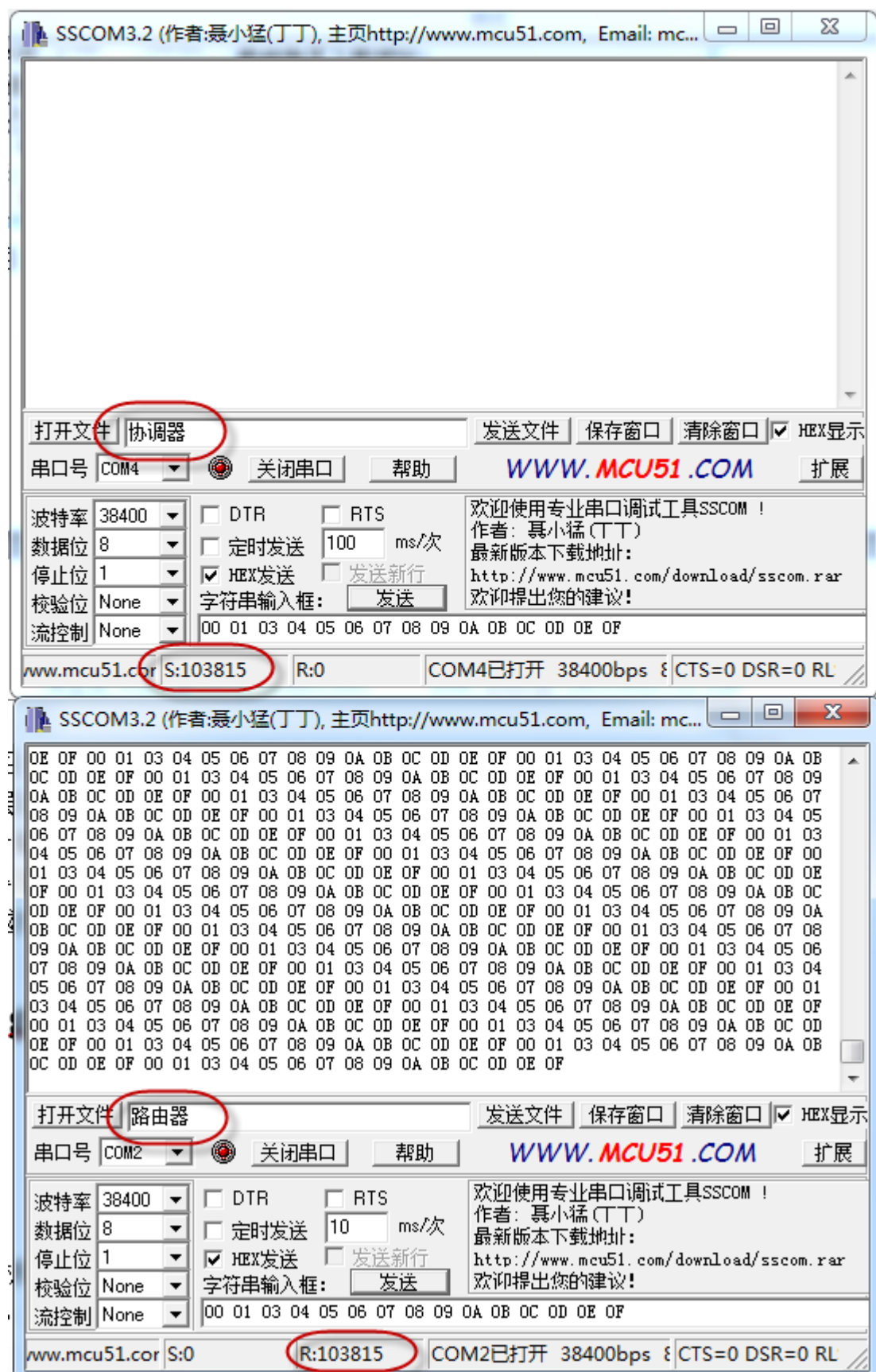


图 16 协调器到路由器数据传输测试

协调器到终端或路由器到终端的数据传输的速度主要取决于终端节点的唤醒定时时间。如果

终端的定时唤醒时间间隔为 1000 秒，那么协调器到终端的数据传输速度就是一帧每秒。路由器到终端的速度也是一帧每秒。但是终端到路由器，终端到协调器的传输速度与路由器到协调器的速度相同。具体信息见上述表格与图 15，图 16。

随着 ZG-Mxx 模块逻辑层次增加和传输距离增加，在传输的过程中可能会出现丢帧的情况，此时用户可以增加发送数据的间隔。路由器到协调器的数据传输方式是单播，而协调器到路由器的数据传输方式为广播方式，所以其速率会低一点。为了得到更好的传输质量，数据长度最好不要太长，最好在 32 个字节之内，越短越好。在传输过程中，并不要求所有的模块的波特率都一样。ZG-Mxx 模块的波特率可以不一样，只要在我们要求的范围内即可。为了使数据传输性能达到极致，建议每一帧的长充控制在 32 个字节之内。

3.7.2 数据透传+Zigbee 短地址

在数据透传的基础上，通过对发送模块的设置。发送模块在发送数据时将自己的网络短地址附加在数据的末尾，接收模块会收到的数据会多出 2 个字节。这个 2 个字节就是发送模块的网络地址，低字节在前，高字节在后。在图 17 中展示了路由器到协调器的数据透传+Zigbee 短地址模型

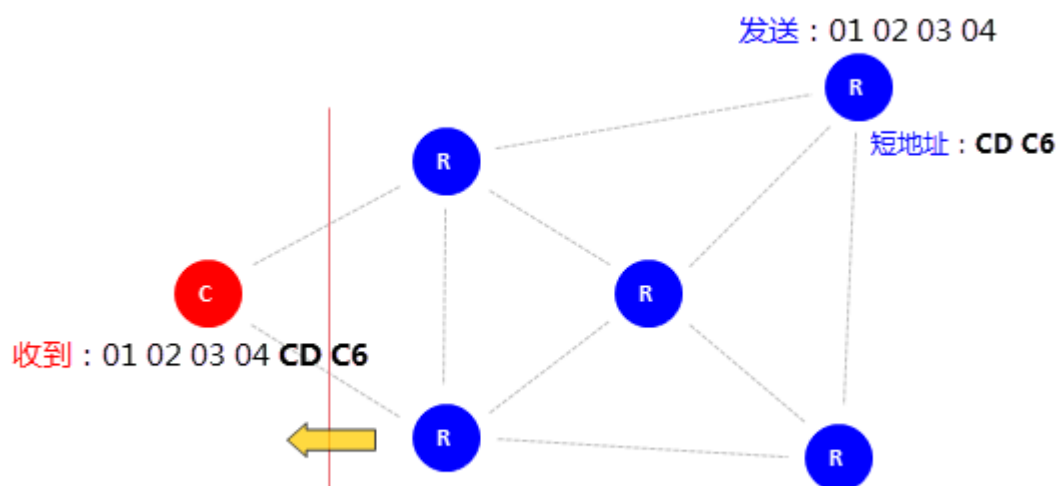


图 17 路由器到协调器传输数据附加短地址

由图 16 可以知道，某节点路由器地址为 CD C6。其向协调器发送数据。

发送数据为：FF 12 34(第一个字节不是 FC，FD，FE)

协调器接收数据为：FF 12 34 C6 CD

接收数据格式为：接收到全部数据及发送方的短地址

图 18，图 19 共同展示了数据透传+zigbee 短地址数据传输模式，其中所使用软件免费提供的测试软件 ZG-Mxx Debugger。

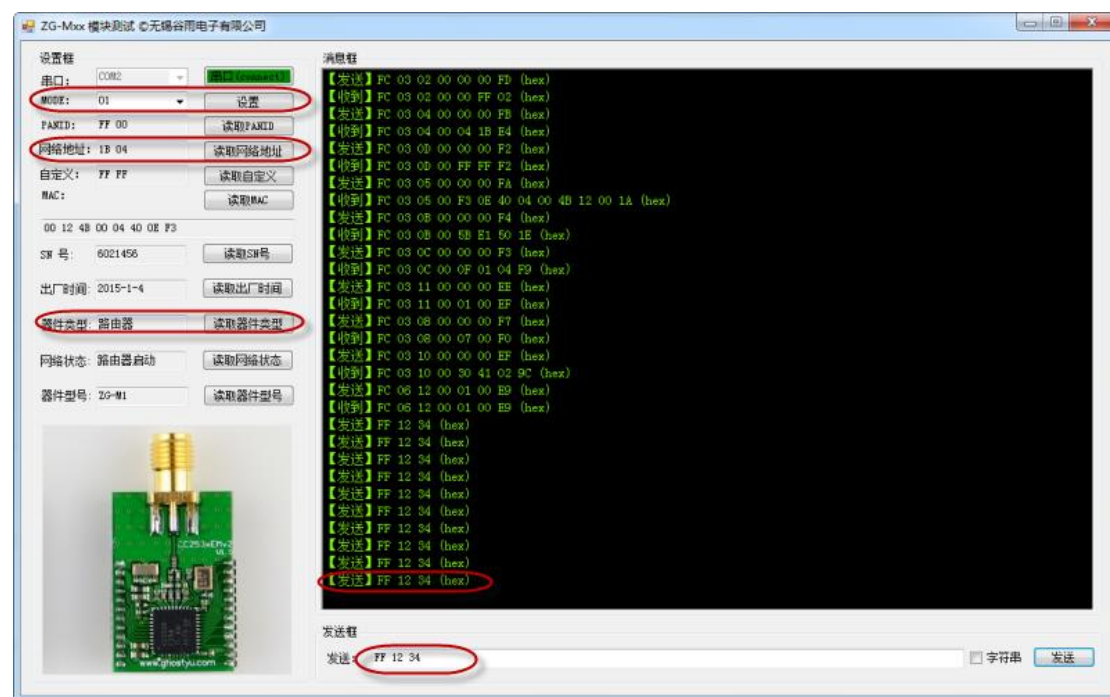


图 18 数据透传+Zigbee 短地址路由器节点

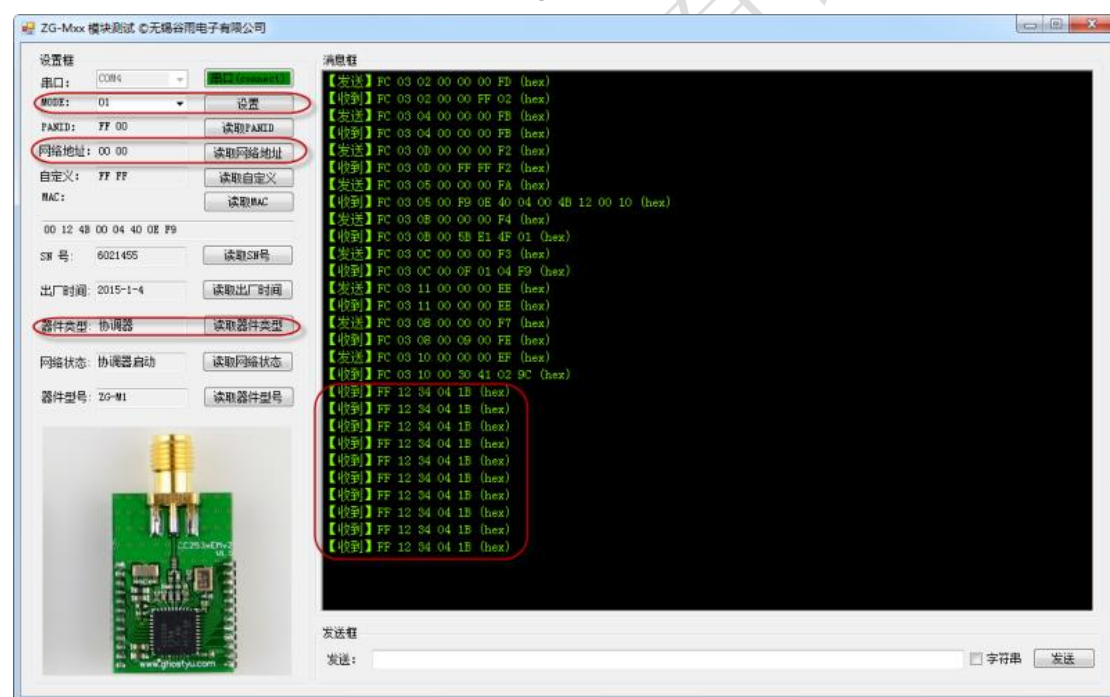


图 19 数据透传+Zigbee 短地址协调器节点

注意：在这个数据透传+Zigbee 短地址模式下，要发送数据长度加短地址长度不要超过 80 字节，超过的部分将不会发送。

3.7.3 数据透传+自定义地址

在数据透传的基础上，通过对发送模块的设置。发送模块在发送数据时将自己自定义短地址附加在数据的末尾，接收模块会收到的数据会多出 2 个字节。这个 2 个字节就是发送模

块的网络地址，低字节在前，高字节在后。这个模式的数据传输与数据透传+Zigbee 短地址相似，只是在接收方收到的短地址是发送端的自定义地址。所以此模式不进行说细的说明。用户可以查看 3.5.1 节的说明。

3.7.4 数据透传+MAC 地址

在数据透传的基础上，通过对发送模块的设置。发送模块在发送数据时将自己的 MAC 地址附加在发送数据的末尾，接收模块会收到的数据会多出 8 个字节。这个 8 个字节就是发送模块的 MAC 地址，低字节在前，高字节在后。图 20 展示了在数据透传+MAC 地址模式下路由器向协调器发送数据模型。

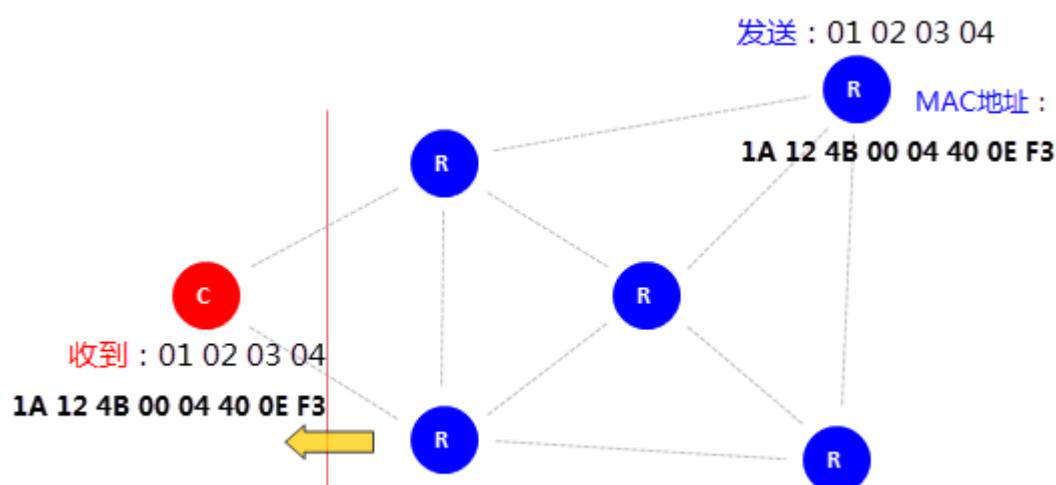


图 20 数据透传+MAC 地址

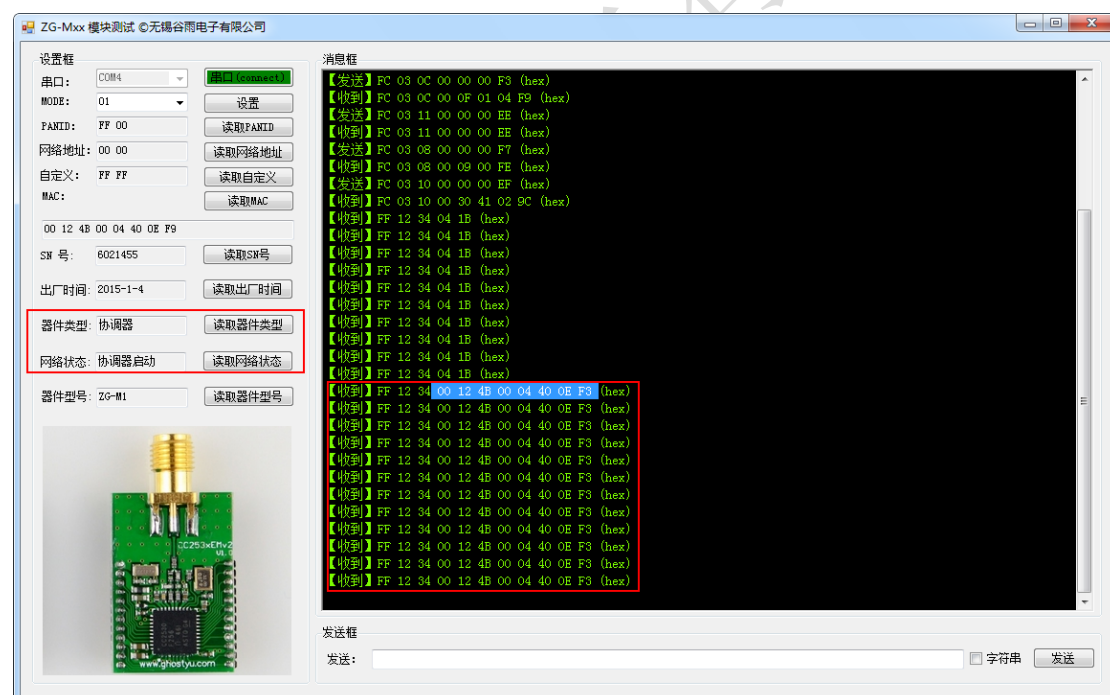
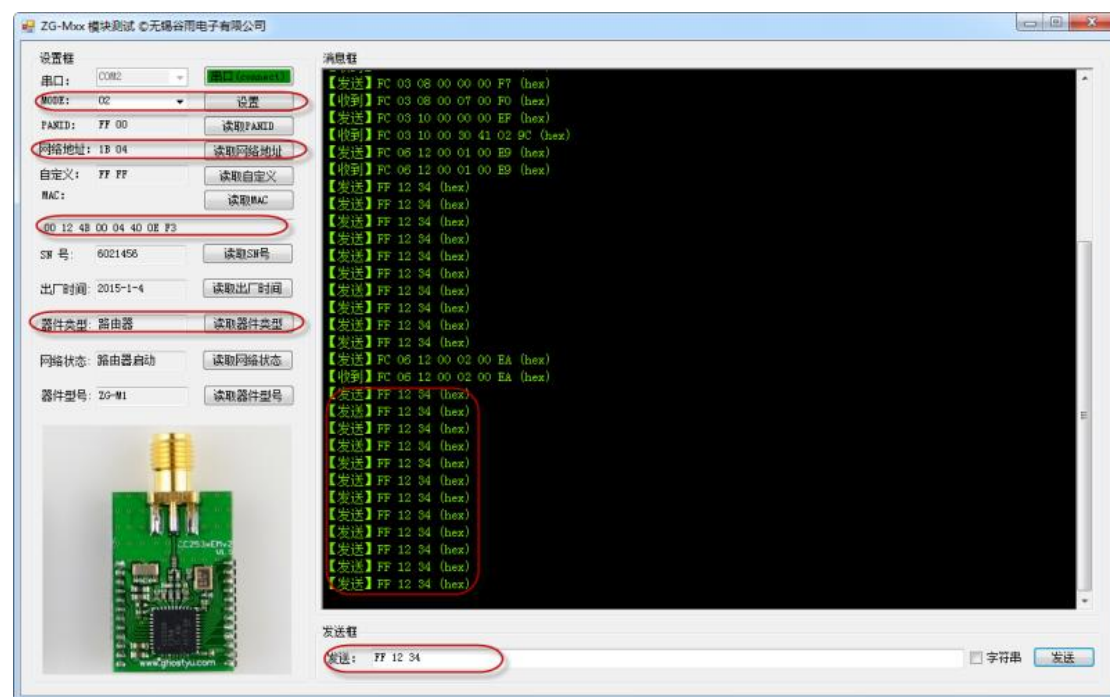
由图 18 可以知道，某节点路由器其 MAC 地址 1A 12 4B 00 04 40 0E F3。其向协调器发送数据。

发送数据为：FF 12 34(第一个字节不是 FC，FD，FE)

协调器接收数据为：FF 12 34 1A 12 4B 00 04 40 0E F3

接收数据格式为：接收到全部数据及发送方的 MAC 地址

图 21，图 22 共同展示了数据透传+MAC 地址数据传输模式，其中所使用软件免费提供的测试软件 ZG-Mxx Debugger。



注意: 在这个数据透传+MAC 地址模式下, 要发送数据长度加短地址长度不要超过 80 字节, 超过的部分将不会发送。

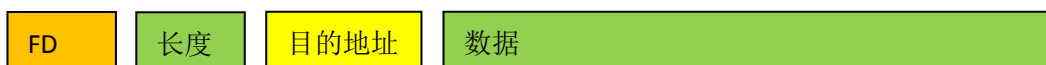
3.7.5 点对点数据传输, 用 ZIGBEE 短地址寻址 (一)

Zigbee 网络内任何节点间, 都可以实现点对点数据传输。在发送方在数据的末尾附加发

送方的网络短地址。接收方便会多接收到 2 个字节的数据。

发送数据格式：

点对点帧头 (FD) + 发送数据长度 (1 个字节) + 目标地址 (zigbee 短地址, 低字节在前, 高字节在后) + 数据 (最多 32 个字节, 超出部分丢弃)



例如：

发送：FD 0A 4C CB 01 02 03 04 05 06 07 08 09 0A

FD：点对点数据传输指令

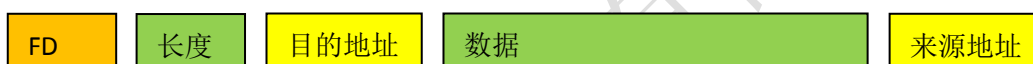
0A：数据域长度

4C CB：目标地址

01 02 03 04 05 06 07 08 09 0A：要发送的数据

接收数据格式：

点对点帧头 (FD) + 发送数据长度 (1 个字节) + 目标地址 (zigbee 短地址, 低字节在前, 高字节在后) + 数据 (最多 32 个字节) + 发送端短地址 (低字节在前, 高字节在后)



例如：FD 0A 4C CB 01 02 03 04 05 06 07 08 09 0A 04 1B

FD：点对点数据传输指令

0A：数据域长度

4C CB：目的地址 (对接收方来说, 即是自己短地址)

01 02 03 04 05 06 07 08 09 0A：接收到的数据

04 1B：数据发送端地址

图 23 展示了 Zigbee 短地址寻址的点对点数据传输模型。

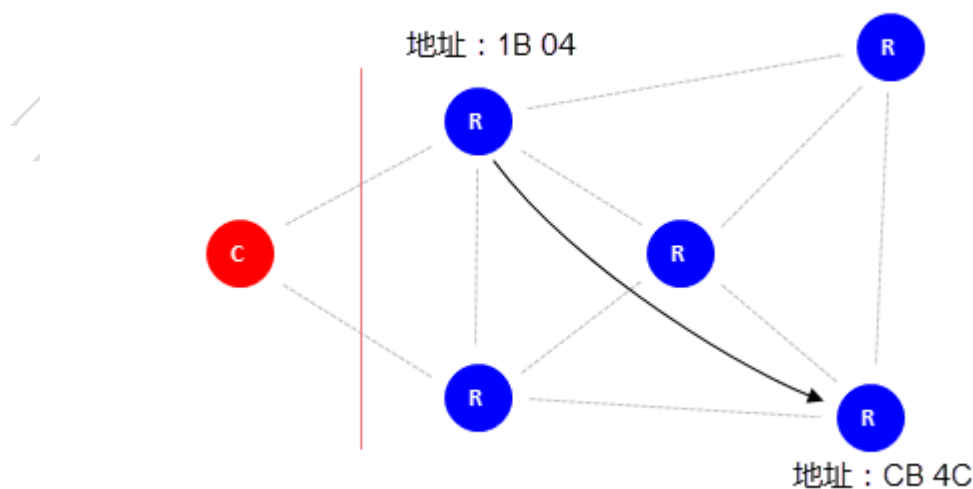


图 23 点对点数据传输模型图

图 24, 图 25 展示了点对点数据传输, 用 ZIGBEE 短地址寻址模式下数据传输过程, 其中所使用软件是本公司免费提供的测试软件 ZG-Mxx Debugger。

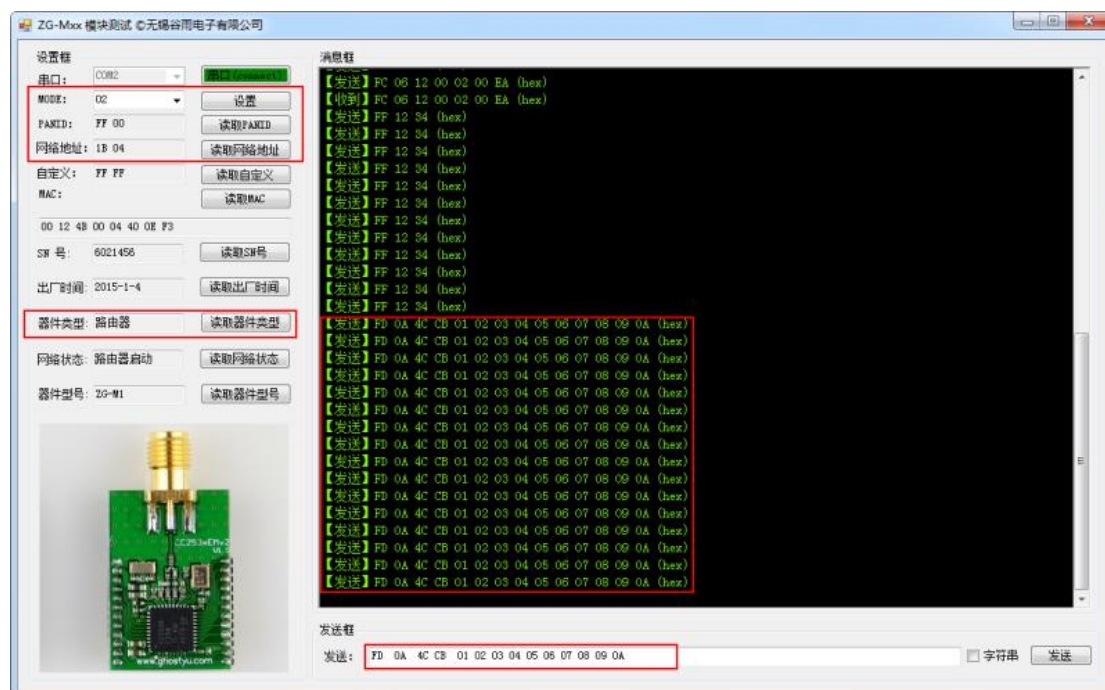


图 24 点对点数据传输发送方

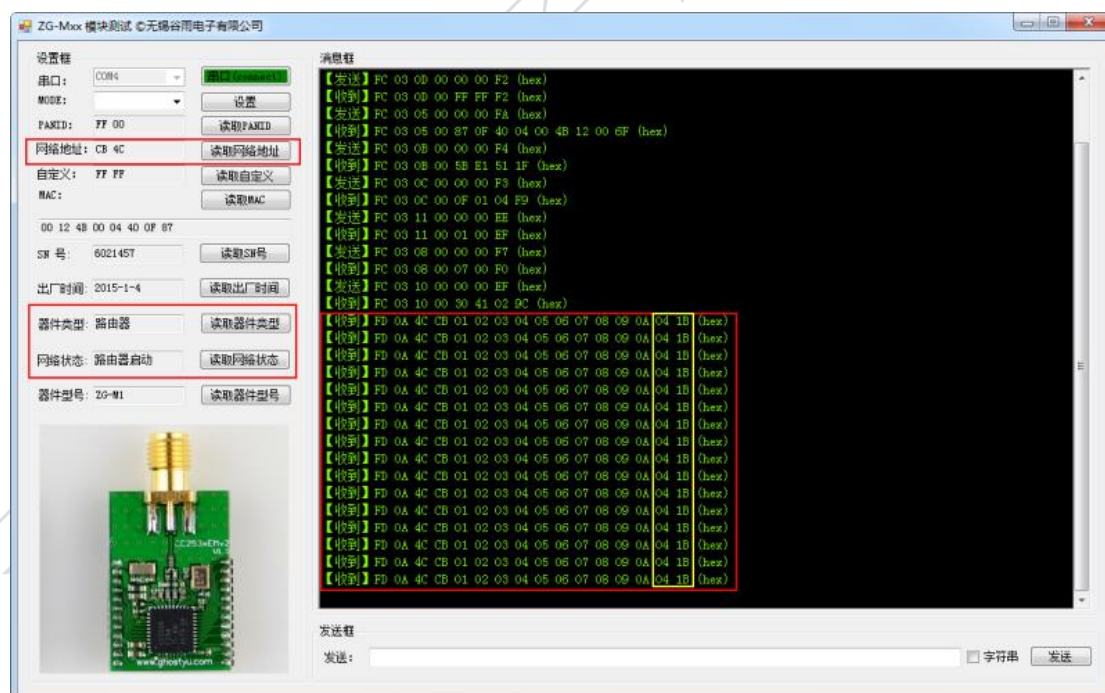


图 25 点对点数据传输接收方

在点对点数据传输方式中, 其传输可以在网络内任意节点之间进行。它具有以下几个特点。

- 即使协调器离开网络, 网络中只存在路由器, 点对点数据传输也可在节点之间进行
- 点对点数据传输最多只能发送 32 个字节数据, 即指令中数据域最多 32 个字节。
- 如果目的地址为 FF FF, 则 ZG-Mxx 模块将以广播的方式发给网络中所有节点。相应的目的地址为 00 00, 将会发给协调器。

点对点数据传输测试

| 传输方向 | 数据包长度（字节） | 最快间隔（ms） |
|--|-----------|----------|
| 路由器->路由器（单播） | 32 | 20 |
| 协调器->路由器（单播） | 32 | 20 |
| 路由器->协调器（单播） | 32 | 20 |
| 试条件： 7. 室温，实验室条件 8. ZG-Mxx 模块间距离 2 米，信号良好 9. 传输深度 2 10. 串口波特率 38400，8，N，1 11. 连续发送，接 10K 字节，无误码，连续测试 10 次 测试软件：串口调试助手 SSCOM3;ZG-Mxx Debugger(我们自己的软件，免费提供) | | |

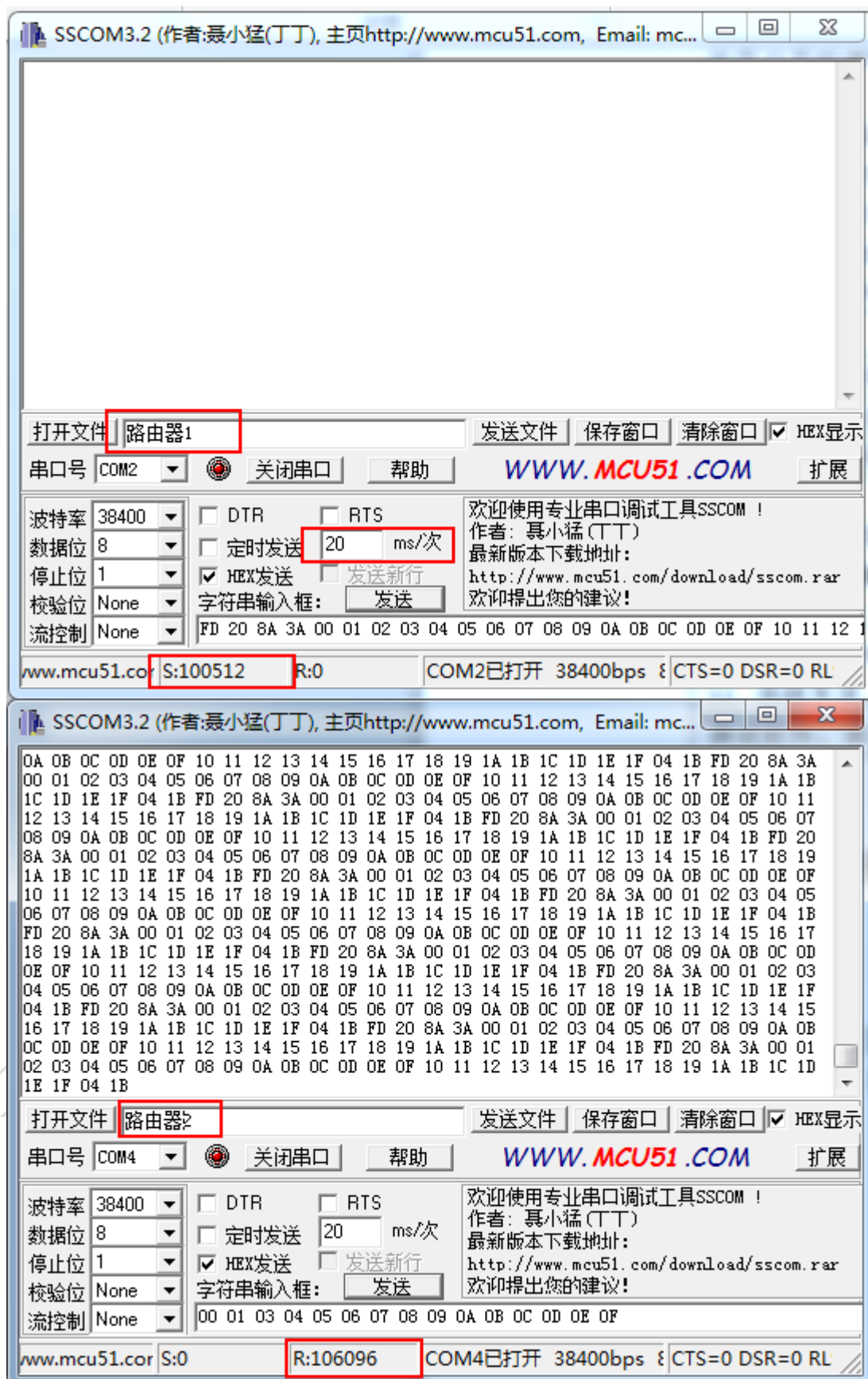


图 26 点对点数据传输测试

3.7.6 点对点数据传输，用 ZIGBEE 短地址寻址（二）

此模式与 3.7.4 的点对点数据传输相似。只是在接收到端会去掉包头包尾，只返回数据域中的数据。所以此处不做详细的说明，用户参考 3.7.5 节的说明内容即可。

3.7.7 点对点数据传输，自定义地址寻址

在 ZG-Mxx 模块中我们增加了用户自定义地址功能，用户可以为每一个 ZG-Mxx 模块设定一个地址。此地址与 Zigbee 短地址互不干涉共同存在，且掉电非易失。如果 ZG-Mxx 模块恢复出厂设置，自定义地址将会恢复成 FF FF。

修改 ZG-Mxx 模块的自定义，可用两种方式进行设置。一是通过串口指令进行设置，具体的指令请看《4.0 ZG-Mxx 模块设置》；二是通过我们提供的 PC 端 APP ZG-Mxx Setting 进行设置。下面的内容主要是基于 ZG-Mxx Setting，ZG-Mxx Debugger 软件进行说明。

将要使用的 ZG-Mxx 通过底板与 PC 的串口进行连接。打开我们提供的 ZG-Mxx Setting 工具软件，在 ZG-Mxx Setting 界面中选择与 ZG-Mxx 模块相连的串口，点击**串口（close）**按钮。如果串口没有被占用，且串口号选择正确，其按钮会变成橙色**串口（open）**，绿色**串口（connected）**，如图 27 所示。详细的操作见 ZG-Mxx Setting 软件说明书《ZG-Mxx Setting 使用说明书.pdf》。

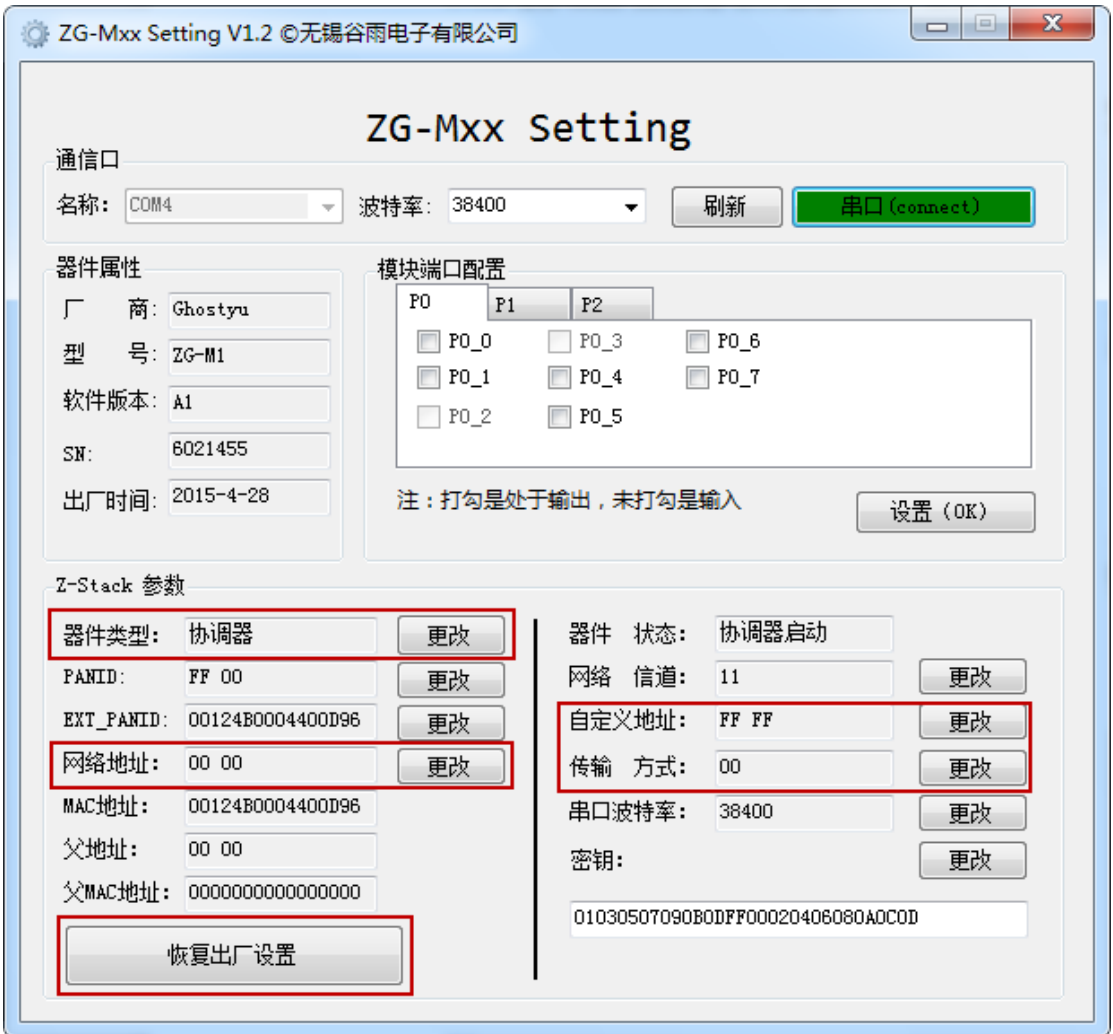


图 27 ZG-Mxx Setting

在软件界面中，用户主要修改的地方就是自定义地址，和传输方式。在 Z-Stack 参数一栏中点击自定义地址项后的更改按钮，便会弹出更改对话框，用户只要在修改地址项中填写要设定的自定义地址，按下确定按钮即可，注意这些数据都是 16 进制数据。如图 28 所示。更改成功之后，图 26 所示的界面中自定义地址项中就会显示用户更改成功的地址。

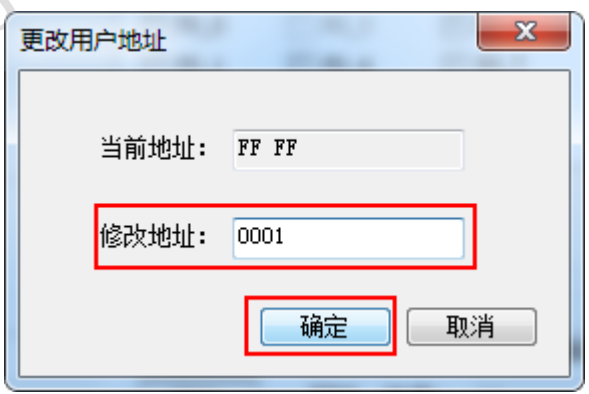


图 28 更改用户地址

相同的操作，用户更改传输方式。将传输方式设成 05 模式。其传输模式的设定可参考表三。

表三 传输模式设定指南

| 传输模式值 | 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 自定义地址寻址, 不 含包头包尾 |

经过上述设置后, 便可将这些 ZG-Mxx 模块组网, 并进行数据传输。

例如:

发送: FD 0A 01 00 01 02 03 04 05 06 07 08 09 0A

FD: 点对点数据传输指令

0A: 数据域长度

01 00: 自定义目标地址 (低字节在前, 高字节在后)

01 02 03 04 05 06 07 08 09 0A : 要发送的数据

接收: 01 02 03 04 05 06 07 08 09 0A

只有 ZG-Mxx 模块的自定义地址为 0001 的设备才能接收到数据。

图 29 展示了点对点数据传输自定义地址寻址模型

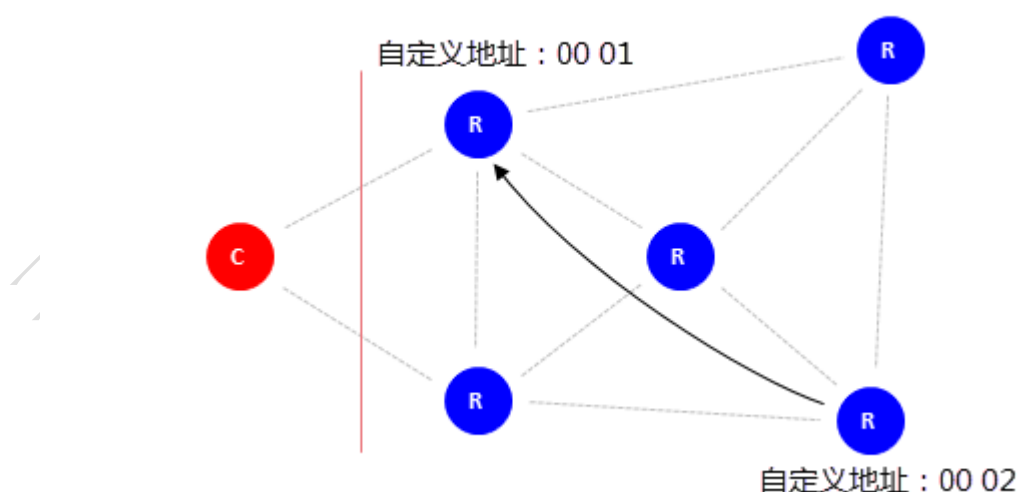


图 29 点对点数据传输自定义地址寻址数据传输模式

图 30 , 图 31 与图 32 是对自定义地址寻址的点对点数据传输的测试

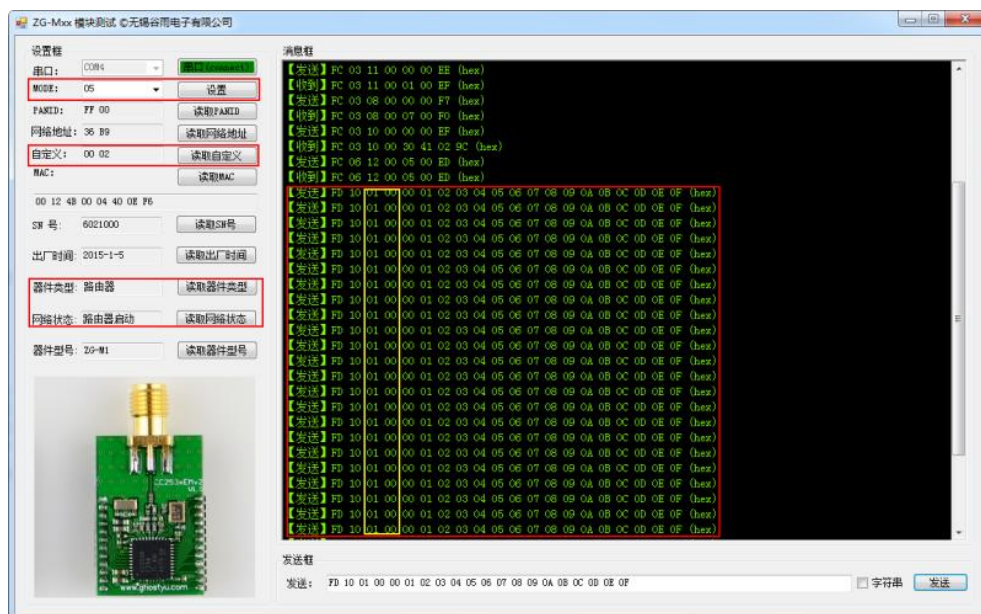


图 30 发送端

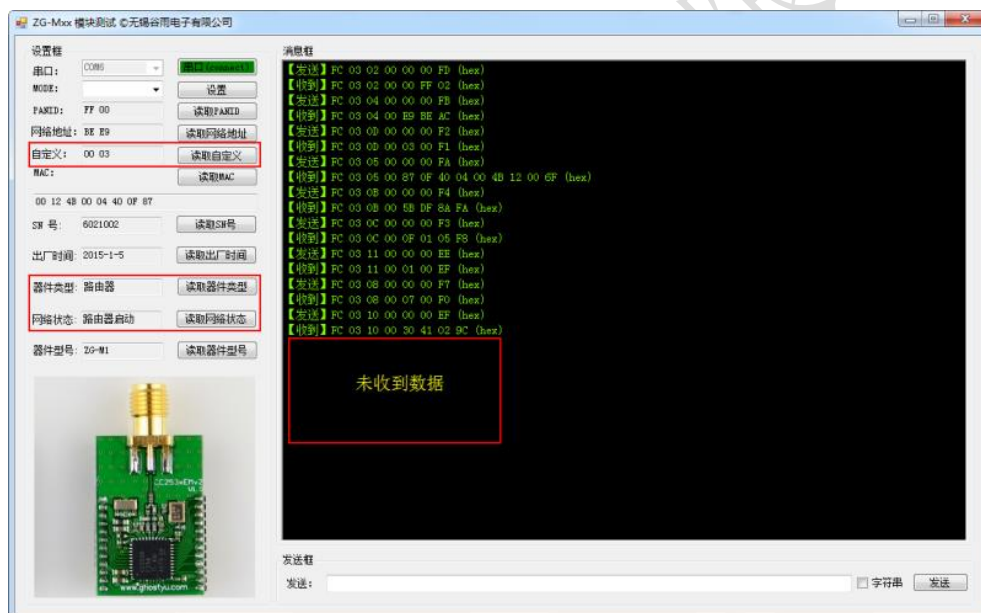


图 31 网络中其它节点

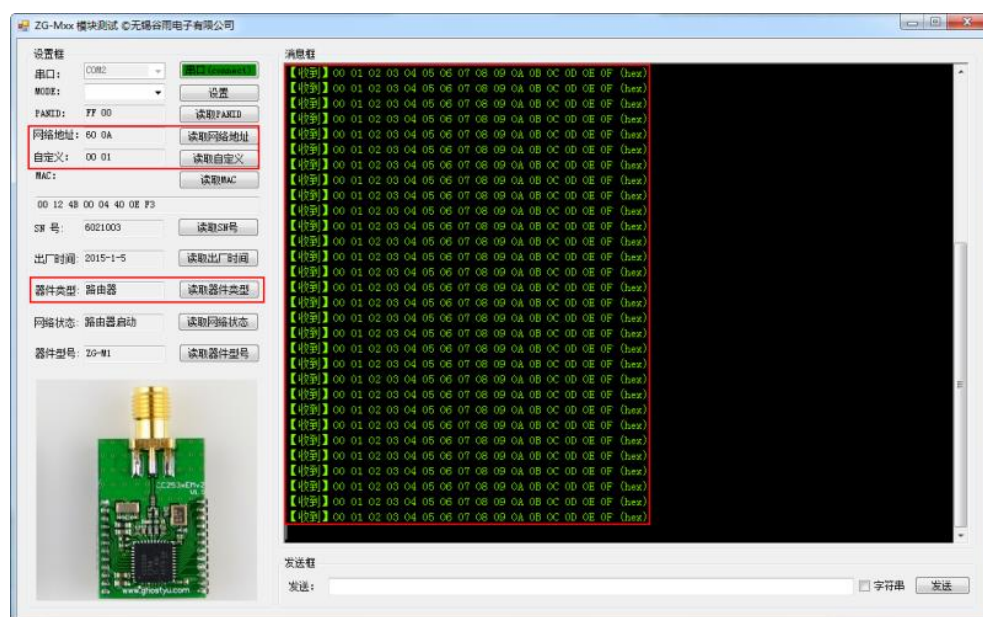


图 32 接收端

点对点数据传输自定义地址寻址模式，是非常实用的一个功能。因为自定义地址不会因为 ZG-Mxx 的网络结构改变而改变，且掉电非易失。如果想要更改自定义地址只以通过手动更改，这都在用户的可控之下。而 Zigbee 短地址会随网络结构的改变而改变，而这些改变是不可预测，不要控的。为了解决这个问题，我们在 V3 版本中加入了网络地址设定功能。模块在加入网络中，就会分配自己设定的网络地址。然而这个功能在模块出厂时是关掉的。如果用户想要打开这个功能，可以将网络地址设定成自己想要的数值，这样网络地址设定的功能就会打开。希望用户多用这个功能，这可以提高数据传输性能。而在 V1, V2 版本中，则没有这个网络地址的设定。只能使用自定义地址来克服了网络地址弊端，给用户的数据传输带来方便合理的传输。

3.8 ZG-Mxx 远程 GPIO 控制

在 V1, V2 版本中，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 版本，V2 版本默认是 P0 端口是输入，P1 和 P2 默认是输出，而在 V3, A4 版本中 P0, P1, P2 都是输入状态，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，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。

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。

如果接收者接收到了远程节点的 GPIO 控制请求，会处理相应的 GPIO 口。如果命令是输出，接收者会将相应的 GPIO 口设置成输出高低电平，并返回相应的响应数据，如果这个 GPIO 口无效，将返回无效操作响应。如果这个命令是读操作，接收者会将相应的引脚的状态返回，如果这个 GPIO 口无效，将返回无效操作响应。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。

接收者正常响应：FC 08 14 00 XX YY ZZ AA BB CC XOR

如果命令是 06，那么 CC 保持不变。

如果命令是 03，那么 CC 是引脚的状态，一个 bit 位，对应一个引脚状态

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

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

3.9 ZG-Mxx 模块网络打开与关闭功能

（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。

3.10 ZG-Mxx 模块网络环境清除

（此功能只能在 V3 版本的模块中使用）

在 ZG-Mxxs 模块中，当模块加入网络工作时，都会保存当前网络环境。这样带来的好处是再次上电时不需要在进行加入网络，而是直接恢复保存的网络环境。虽然这样会有好处，但有时我也希望清除一下网络环境，以新的模块启动。这时就会进行搜所网络，查找网络，加入网络等操作。

在 V1，V2 版本的后期模块中，可以通过串口向模块发送 FC 06 18 00 00 00 XOR 命令进行清除网络环境，也可以通过将 P0.1 拉成低电平，然后重新上电。设置成功之后，模块就会在一秒钟之后，重新启动模块。

在 V3 版本的模块中，不仅可以通过串口完成这个工作，也可以通过将 P0.1 拉成低电平，

然后重新上电。这时的模块就会以新的网络环境去启动模块，而不会去恢复网络环境。

3.11 ZG-Mxx 远程 ADC 读取

在 V1, V2 版本中，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 XOR

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

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：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 2047，即 0x7FF。这里计算原始信号的值的公式为：

$$x = \frac{3.3 \times y}{2047} (V), \text{ 假设这里的参考电压为 } 3.3V, \text{ ADC 的值为 } y.$$

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

3.12 ZG-Mxx 终端数据发送与接收

（此说明，只对终端节点有效）

在终端节点中，系统占用了 P0 的 P0.0, P0.1, P0.2, P0.3。其中 P0.2 与 P0.3 是作为串口使用，用于数据的发送与接收。P0.1 是输入引脚，默认电平是高电平。当在此引脚上出现低电平时，终端节点将会从睡眠状态唤醒。所以如果要从终端的串口向网络中其它节点发送数据时，必须在 P0.1 上产生低电平保持。在保持低电平期间，将要发送的数据从串口发送给终端节点，发送完成后，将引脚恢复成高电平，以至使模块能进入省电模式。

P0.0 是输出状态。默认输出高电平。此引脚是用来指示，终端节点收到网络信息并将信息从串口输出。当数据从串口输出之前，会将 P0.0 的输出从高电平状态变成低电平，在低电平保持期间，接收到的数据会从串口输出。输出完成之后，P0.0 恢复成高电平。如图 33 所示。此引脚可以用来唤醒与终端节点相连的 MCU，以便使 MCU 能正确的收到串口数据。

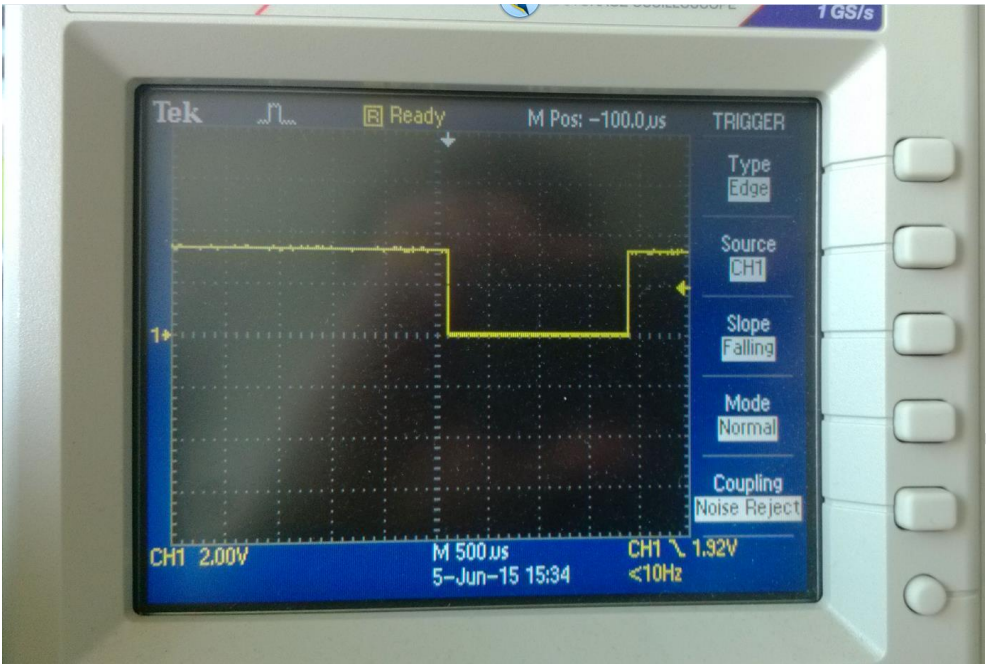


图 33 P0.0 电平变化

4 ZG-Mxx 系列设置指令描述

以下表格中的数据都是 16 进制数据
其设置命令格式如下：



FC：设置命令的帧头

03/06：功能码。03 是读，06 是写

命令 ID：表示此命令的功能，低字节在前

数据：指令携带的数据

XOR：校验码。指令中所有数据相异或，不包括 XOR 本身

ZG-Mxx 模块的 GPIO 端口引脚占用屏蔽位（GPIO0=0F，GPIO1=13，GPIO2=F8）

| 序号 | 指令 | 指令描述 | 响应 | 是否会重启 |
|----|--------------------------|-------------------------------|--|----------------------|
| 1 | FC 06 01 00 00 00 XOR | 恢复出厂设置 | 正确返回： FC 06 01 00 00 00 XOR 错误返回： FC 86 01 00 00 00 XOR | 1 秒钟后 ZG-Mxx 模块重启 |
| 2 | FC 03 02 00 00 00 XOR | 读取模块已加入 或 想 要 加 入 PANID | 正确返回： FC 03 02 00 XX XX XOR (XX XX 即 | 否 |

| | | | | |
|---|--|---|--|--|
| | | | 是 PANID, 低字节在前) 错误返回: FC 83 02 00 00 00 XOR | |
| 3 | FC 06 02 00 XX XX XOR (XX XX 是欲设定 PANID, 低字节 在前) | 设置模块想要加入 网络的 PANID 或是创建 PANID 网络 | 正确返回: FC 06 02 00 XX XX XOR 错误返回: FC 86 02 00 XX XX XOR | 如果模块是路 由器 1 秒钟后 ZG-Mxx 模块重 启; 如果模块是 协调器 3 秒后 通知网络修改 PANID |
| 4 | FC 03 03 00 00 00 XOR | 读取模块加入网 络的扩展 PANID | 正确返回: FC 03 02 00 XX XX XX XX XX XX XX XX XOR (XX XX.... 即是扩展 PANID, 低字节在 前) 错误返回: FC 83 02 00 00 00 XOR | 否 |
| 5 | FC 06 03 00 XX XX XX XX XX XX XX XX XOR (XX XX.... 即是 扩展 PANID, 低 字节在前) | 设定模块欲加入 或形成网络的扩 展 PANID | 正确返回: FC 03 02 00 XX XX XX XX XX XX XX XX XOR (XX XX.... 即是扩展 PANID, 低字节在 前) 错误返回: FC 83 02 00 XX XX XX XX XX XX XX XX XOR | 如果当前扩展 PANID 与 设定 PANID 相同, 则 不会重启。不相 等 1 秒钟后 ZG-Mxx 模块重 启 |
| 6 | FC 03 04 00 00 00 XOR | 读取模块的网络 地址 | 正确返回: FC 03 04 00 XX XX XOR (XX XX 模 块在网络中的地 址, 如果没有加 入网络则为 FF FE) 错误返回: FC 83 04 00 XX XX XOR | 否 |
| 7 | FC 06 04 00 XX | 设定模块在网络 | 正确返回: | 否 |

| | | | | |
|----|---------------------------------|--|--|---|
| | XX XOR (XX XX 欲设定模块在网络中的短地址) | 中的短地址（该指令只对路由器有效且已加入网络，如果对协调器设定将会返回错误） | FC 06 04 00 XX XX XOR 错误返回： FC 86 04 00 XX XX XOR | |
| 8 | FC 03 05 00 00 00 XOR | 读取模块的 MAC 地址 | 正确返回： FC 03 05 00 XX XX XX XX XX XX XX XX XOR (XX XX....是 MAC 地址，低字节在前) 错误返回： FC 83 05 00 00 00 XOR | 否 |
| 9 | FC 03 06 00 00 00 XOR | 读取模块的父设备网络地址 | 正确返回： FC 03 06 00 XX XX XOR (XX XX.... 是父设备地址，低字节在前。没有加入网络则返回 00 00) 错误返回： FC 83 06 00 00 00 XOR | 否 |
| 10 | FC 03 07 00 00 00 XOR | 读取模块的父设备 MAC 地址 | 正确返回： FC 03 07 00 XX XX XX XX XX XX XX XX XOR (XX XX.... 是父设备 MAC 地址，低字节在前。没有加入网络则返回 00 00...) 错误返回： FC 83 07 00 00 00 XOR | 否 |
| 11 | FC 03 08 00 00 00 XOR | 读取模块的状态 | 正确返回： FC 03 08 00 XX 00 XOR (XX 设备状态地址) 错误返回： FC 83 08 00 00 00 XOR | 否 |

| | | | | |
|----|--------------------------|---|---|---------|
| 12 | FC 03 09 00 00 00 XOR | 读取 Zigbee 网络的 工作的信道 | 正确返回: FC 03 09 00 XX 00 XOR (XX 设备 工作的信道, 未 加入网络返回 00) 错误返回: FC 83 09 00 00 00 XOR | 否 |
| 13 | FC 06 09 00 XX 00 XOR | 设定 zigbee 的工 作的信首, 其中 XX 的数值只能 为 11-26。超出范 围将返回无效响 应 | 正确返回: FC 06 09 00 XX 00 XOR 错误返回: FC 86 09 XX 00 00 XOR | 1 秒后会重启 |
| 14 | FC 03 0A 00 00 00 XOR | 读取模块的厂商 ID | 正确返回: FC 03 0A 00 01 00 XOR (01 模块 厂商 ID) 错误返回: FC 83 0A 00 00 00 XOR | 否 |
| 15 | FC 03 0B 00 00 00 XOR | 读取模块的 SN 号 | 正确返回: FC 03 0B 00 XX XX XX XOR (XX XX XX 十六进制数 据, 换成十进制 数据即可) 错误返回: FC 83 0B 00 00 00 XOR | 否 |
| 16 | FC 03 0C 00 00 00 XOR | 读取模块的出厂 时间 | 正确返回: FC 03 0C 00 XX XX XX XOR (XX XX XX 依次为年最 后 2 位, 月, 日 的 16 进制。例如 FC 03 0C 00 0F 01 05 XOR, 出厂 时间为 15-1-5) 错误返回: FC 83 0C 00 00 00 XOR | 否 |
| 17 | FC 03 0D 00 00 | 读模块的自定义 | 正确返回: | 否 |

| | | | | |
|----|---|----------------------------------|---|---|
| | 00 X0R | 地址 | FC 03 0D 00 XX XX X0R (XX XX 模块的自定义地址, 低字节在前) 错误返回: FC 83 0D 00 00 00 X0R | |
| 18 | FC 06 0D 00 XX XX X0R (XX XX 欲设定的模块自定义地址, 低字节在前) | 设定模块的自定义地址 | 正确返回: FC 06 0D 00 XX XX X0R 错误返回: FC 86 0D 00 XX XX X0R | 否 |
| 19 | FC 03 0E 00 00 00 X0R | 读取模块 GPIO 口输入输出状态 | 正确返回: FC 03 0E 00 XX XX XX X0R (XX XX XX 依次为 P0 P1 P2 端口引脚输入输出状态, 一 bit 位代表一个引脚的输入输出状态。1 代表输出, 0 代表输入, 除了占用引脚。) 错误返回: FC 83 0E 00 00 00 X0R | 否 |
| 20 | FC 06 0E 00 XX XX XX X0R (XX XX XX 依次为 P0 P1 P2 端口引脚输入输出状态, 一 bit 位代表一个引脚的输入输出状态。1 代表输出, 0 代表输入。占用引脚必设为 0。) | 设定 GPIO0, GPIO1, GPIO2 端口引脚的输入输出 | 正确返回: FC 06 0E 00 XX XX XX X0R 错误返回: FC 86 0E 00 XX XX XX X0R | 否 |
| 21 | FC 03 0F 00 XX XX X0R (第一个红色 XX 代表 GPIO 端口号, 只能为 0, 1, | 读取相应端口引脚状态 (本地) | 正确返回: FC 03 0F 00 XX XX X0R (第一个红色 XX 代表端口号, | 否 |

| | | | | |
|----|---|--------------------|---|----------------|
| | 2; 第二个绿色 XX 代表读取端口的引脚, 要读取引脚状态将相应 bit 位置 1。系统占用引脚必须设为 0) | | 第二个绿色 XX 代表读取引脚状态, 一位代表一个引脚状态) 错误返回: FC 86 0E 00 XX XX XOR | |
| 22 | FC 06 0F 00 XX XX xx XOR (第一个红色 XX 代表 GPIO 端口号, 只能为 0, 1, 2; 第二个绿色 XX 代表设置端口的引脚, 设置引脚状态相应 bit 位置 1; 第三个小 xx 表示输出的状态, 相应引脚 bit 为 0 或 1。系统占用引脚必须设为 0) | 设置相应端口引脚状态 (本地) | 正确返回: FC 06 0F 00 XX XX xx XOR 错误返回: FC 86 0E 00 XX XX xx XOR | 否 (恢复出厂设置不会恢复) |
| 23 | FC 03 10 00 00 00 XOR | 读取模块软件版本及型号信息 | 正确返回: FC 03 10 00 XX XX XX XOR (XX XX XX 依次为软件版本副号 ASCII, 软件版本主号 ASCII, 模块型号) 错误返回: FC 83 10 00 00 00 XOR | 否 |
| 24 | FC 03 11 00 00 00 XOR | 读取模块在 Zigbee 网络中类型 | 正确返回: FC 03 11 00 XX 00 XOR (XX 是器件在网络中类型, 00 代表协调器, 01 代表路由器) 错误返回: FC 86 11 00 00 00 XOR | 否 |
| 25 | FC 06 11 00 XX | 设置模块在 | 正确返回: | 1 秒钟后 ZG-Mxx |

| | | | | |
|----|--|---------------------------------|--|--------|
| | 00 XOR (XX 00 代表协调器, 01 代表路由器) | Zigbee 网络中的类型 | FC 06 11 00 XX 00 XOR 错误返回: FC 86 11 00 XX 00 XOR | 模块重启 |
| 26 | FC 03 12 00 00 00 XOR | 读取模块的数据传输模式 | 正确返回: FC 03 12 00 XX 00 XOR (XX 见表四说明) 错误返回: FC 83 12 00 00 00 XOR | 否 |
| 27 | FC 06 12 00 XX 00 XOR (XX 见表四说明) | 设定模块的数据传输模式 | 正确返回: FC 06 12 00 XX 00 XOR 错误返回: FC 86 12 00 XX 00 XOR | 否 |
| 27 | FC 03 13 00 XX 00 XOR (XX 见表五说明) | 读取模块重启后的波特率 | 正确返回: FC 03 13 00 XX 00 XOR (XX 见表五说明) 错误返回: FC 83 13 00 XX 00 XOR | 否 |
| 28 | FC 06 13 00 XX 00 XOR (XX 见表五说明) | 设定模块重启后的波特率 (新设定的波特率, 重启后生效) | 正确返回: FC 06 13 00 XX 00 XOR (XX 见表五说明) 错误返回: FC 86 13 00 XX 00 XOR | 需要手动重启 |
| 29 | FC 03 14 00 XX V1 V2 YY ZZ 00 XOR (XX 表示 V1 V2 的地址类型, 如果 XX 大于 0 表示 V1 V2 是 zigbee 短地址, 如果 XX 等于 0 表示 V1 V2 是自 | 读取远程节点的相应端口引脚的状态 | 正确返回: FC 08 14 00 XX V1 V2 YY ZZ HH XOR (HH 就是相应引脚的状态值, 一个 bit 位对应一个引脚) 错误返回 (远程): FC 88 14 00 | 否 |

| | | | | |
|----|--|-----------------------|--|---|
| | 定义地址。地址都是低字节在前,高字节在后。YY 表示端口号,ZZ 表示端口的引脚,一个 bit 位对应一个引脚) | | XX V1 V2 YY ZZ 00 XOR (这个是目标节点返回的错误信息) 本地错误返回: FC 83 14 00 XX V1 V2 YY ZZ 00 XOR(本地节点在检查帧的有效性返回无效响应) 响应超时返回: FC 04 14 00 XX V1 V2 YY ZZ 00 XOR (三秒之内,目标节点没有返回任何数据) | |
| 30 | FC 06 14 00 XX V1 V2 YY ZZ HH XOR (XX 表示 V1 V2 的地址类型,如果 XX 大于 0 表示 V1 V2 是 zigbee 短地址,如果 XX 等于 0 表示 V1 V2 是自定义地址。地址都是低字节在前,高字节在后。YY 表示端口号,ZZ 表示端口的引脚,一个 bit 位对应一个引脚。HH 就是将目标节点的输出引脚设置输出的高低电平。如果目标节点的引脚输入将返回错误) | 设定远程节点的相应端口引脚输出高低电平状态 | 正确返回: FC 08 14 00 XX V1 V2 YY ZZ HH XOR (HH 就是相应引脚的状态值,一个 bit 位对应一个引脚) 错误返回 (远程): FC 88 14 00 XX V1 V2 YY ZZ 00 XOR (这个是目标节点返回的错误信息) 本地错误返回: FC 86 14 00 XX V1 V2 YY ZZ 00 XOR(本地节点在检查帧的有效性返回无效响应) 响应超时返回: FC 04 14 00 XX V1 V2 YY ZZ 00 XOR (三秒之内,目标节点没有返回任何数据) | 否 |
| 31 | FC 03 15 00 00 00 XOR | 读取密钥 (只针对 V2 版本) | 正确返回: FC 03 15 00 X1 X2 X3 X4 X5 X6 X7 X8 | 否 |

| | | | | |
|----|--|--|---|--------|
| | | | X9 X10 X11 X12 X13 X14 X15 X16 X0R (X 就是安全网络的密钥) 错误返回: FC 83 15 00 00 00 X0R | |
| 32 | FC 06 15 00 X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X0R | 设定网络连接密钥。如果连接密钥与协调器的密钥不正确是加入不了网络, 安全网络的加入工作协调器必须在网络中, 如果协调器不在网络, 节点是加入不网络, 并且每隔 10s 就会重启 | 正确返回: FC 06 15 00 X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X0R 错误返回: FC 86 15 00 X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X0R | 重启之后生效 |
| 33 | FC 03 16 00 00 YY X0R (YY 对应引脚, 一个 bit 位对应一个引脚) | 读取自己的 P0 口对应引脚的 ADC 值 | 正确返回: FC 03 16 00 00 YY AD1 AD2 X0R (AD1 AD2 就是 ADC 值, 低字节在前, 高字节在后) 错误返回: FC 83 16 00 00 YY X0R | 否 |
| 34 | FC 03 17 00 XX V1 V2 00 ZZ 00 X0R | 读取目标节点的相应引脚 ADC 值 | 正确返回: FC 08 17 00 XX V1 V2 00 ZZ AD1 AD2 X0R (这是目标节点返回的相应引脚的 ADC 值) 错误返回: FC 88 17 00 XX V1 V2 00 ZZ 00 X0R (目标节点无效返回) 错误返回: FC 83 17 00 XX V1 V2 00 ZZ 00 X0R (命令参数无效返回) 超时返回: FC 04 17 00 XX V1 V2 | 否 |

| | | | | |
|----|---|---|--|-------|
| | | | 00 YY 00 XOR (三秒内目标节点没有回应) | |
| 35 | FC 06 18 00 00 00 XOR | 设定模块以新的网络环境启动 (之前保存的网络环境删除, 但信道, panid 自定义地址不变) | 正确返回: FC 06 18 00 00 00 XOR 错误返回: FC 86 18 00 00 00 XOR | 一秒后重启 |
| 36 | FC 03 19 00 00 00 XOR | 返回模块的定时唤醒时间间隔 (只对终端节点有效) | 正确返回: FC 03 19 00 XX 00 XOR (XX 是定时唤醒时间间隔, 单位是秒) | 否 |
| 37 | FC 06 19 00 XX 00 XOR | 设定终端节点定时唤醒时间间隔 (XX 是具体的秒数, 最大不超过 65。只对终端有效) | 正确返回: FC 06 19 00 XX 00 XOR (XX 是设定的时间间隔) 错误返回: FC 86 19 00 XX 00 XOR (设定未成功) | 否 |
| 38 | FC 03 1A 00 00 00 XOR | 读取终端节点的电池电量 | 正确返回: FC 03 1A 00 XX 00 XOR (XX 表示电池电量, 数值大小为 0-100) 错误返回: FC 83 1A 00 00 00 XOR | 否 |
| 39 | FC 03 1B 00 XX V1 V2 00 00 00 XOR (XX 表示 V1 V2 的地址类型, 如果 XX 大于 0 表示 V1 V2 是 zigbee 短地址, 如果 XX 等于 0 表示 V1 V2 是自定义地址。地址都是低字节在前, 高字节在后) | 读取远程节点的电池电量 | 正确返回: FC 08 1B 00 XX V1 V2 YY 00 00 XOR (YY 就是指定节点的电池电量, 数值大小为 0-100) 本地错误返回: FC 83 1B 00 XX V1 V2 00 00 00 XOR (本地节点在检查帧的有效性返回无效响应) 响应超时返回: FC 04 1B 00 XX V1 V2 00 00 00 XOR (三十秒之 | 否 |

| | | | | |
|----|---|-----------------------------|---|---|
| | | | 内，目标节点没有返回任何数据 | |
| 40 | FC 06 1C 00 XX 00 XOR (XX 为最大为 65 的数值，单位 为秒) | 设定终端节点最 长 ACK 时间 | 正确返回: FC 06 1C 00 XX 00 XOR 错误返回: FC 86 1C 00 XX 00 XOR | 否 |
| 41 | FC 03 1D 00 00 00 XOR | 读取协调器与路 由器的网络打开 与关闭状态 | 正确返回: FC 03 1D 00 XX 00 XOR (XX = 0 表示网 络关闭，不接收 其它节点的加 入。XX = 1 表示 网络打开，接收 其它节点加入) 错误返回: FC 83 1D 00 XX 00 XOR | 否 |
| 42 | FC 06 1D 00 XX 00 XOR (XX = 0 表示 关闭网络。 XX = 1 表示打开 网络) | 设定网络的打开 或者关闭 | 正确返回: FC 06 1D 00 XX 00 XOR 错误返回: FC 86 1D 00 XX 00 XOR | 否 |

表四 数据传输模式设定指南

| 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 自定义地址寻址，不含包头包尾 |
| >05 | 数据透传 | Zigbee 自定义地址寻址，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。

表五 波特率设定指南

| 设定值 | 波特率 |
|-----|-------|
| 00 | 9600 |
| 01 | 19200 |
| 02 | 38400 |

| | |
|-----|--------|
| 03 | 57600 |
| 04 | 115200 |
| >04 | 38400 |

5 ZG-Mxx 可视化设置

为了减轻用户因使用指令带来的麻烦，我们提供了免费的设置软件 ZG-Mxx Setting。此软件免安装，双击即可运行。在 winxp 环境下用户需要安装 Microsoft .NET Framwork 4.0（我们已经替用户下载好了），在 win7 环境下不需要安装任何辅助软件。图 33 展示了 ZG-Mxx Setting 的工具软件的界面。在 ZG-Mxx Setting 软件中，用户可以设置模块任何可以设置的参数。其使用说明可以查看《ZG-Mxx Setting 使用说明.pdf》。



图 33 ZG-Mxx 模块设置软件界面

6 ZG-Mxx 模块常见问题解答

- PANID 是什么？

PANID 是一个 ZigBee 网络的标识。不同的 Zigbee 网络具有不同的网络 ID 号。在同一空间但是具有不同网络 ID 号的 Zigbee 网络，使用时互不干扰。

- **模块的网络地址是什么意思？**

模块的网络地址（Short Address），是模块加入网络后由父设备分配的。网络地址表示其在网络中的标识符，该地址主要用来作为数据收发的目标地址。协调器的网络地址永远是 00 00，如果读到一个模块的网络地址不是 FF FE，则表示该模块已经加入网络，如 CD 4C。

- **PAN ID = 0xFF 0xFF 有什么优点、缺点？**

优点：如果将某个 Router 点的 PAN ID 修改为 FF FF，则这个节点重启后会自动寻找 Zigbee 网络并加入，至于要加入哪个网络，主要由链接质量（LQ）来决定，这个由模块内部计算决定，不需要用户干预。

缺点：如果有多个网络，不能控制加入哪个网络。

- **模块的 MAC 地址是什么？**

模块的 MAC 地址也叫 IEEE 地址，是芯片厂家从 IEEE 协会购买的地址，共 64Bit，全球唯一，可作为 Zigbee 模块的标识。

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>