



E180-ZG120A/B V1.0 用户手册 (ZigBee3.0 自组网模块)

目录

1. 模块介绍	3
1.1 ZigBee 简介	3
1.2 产品特点	3
1.3 支持产品系列	4
2. 功能及命令结构简介	5
2.1 功能引脚表	5
2.2 引脚连接说明	5
2.2.1 串口连接说明	5
2.2.2 引脚位置说明	6
3. 模组固件功能	6
3.1 三种串口模式	6
3.1.1 HEX 指令模式	6
3.1.2 AT 命令模式	9
3.1.3 数据透传模式	10
3.1.4 三种模式的相互切换	11
3.2 休眠节点串口唤醒	12
3.2.1 延迟法	12
3.2.2 连续点击法	13
3.2.3 插入法	13
3.3 模组的 ZCL 标准规范	13
3.3.1 E180-ZG120A/B ZCL 规范功能分配表	13
3.3.2 亿佰特透传协议的 ZCL 规范	14
3.4 功能按键、状态指示灯与 PWM 输出	15
3.4.1 管脚分配	15
3.4.2 网络功能键功能介绍	15
3.4.3 状态指示灯: (建议该引脚外接低电平驱动 LED)	16
4. 应用配置示例	16
4.1 示例 1: 协调器 HEX 命令控制入网节点	16
4.1.1 E180-ZG120A/B 协调器创建网络	16
4.1.2 新节点加入网络以及新节点的识别和配置	18
4.1.3 控制入网节点	21
4.2 示例 2: 终端节点 AT 命令配置和数据透传	26
4.2.1 模组配置成 AT 命令模式入网	26
4.2.2 数据透传	27
4.2.3 AT 命令控灯	28

4.3 示例 3:上位机软件配置组网和透传	31
4.3.1 上位机软件配置协调器	31
4.3.2 上位机软件配置终端节点	33
4.3.3 协调器和终端节点的数据通信	34
4.3.4 上位机设置数据透传	36
5. 用户须知	38
5.1 ZigBee 网络角色以及注意事项	38
5.2 网络结构	40
6. 定制合作	41
7. 关于我们	41

1. 模块介绍

1.1 ZigBee 简介

ZigBee 技术是一种近距离、低复杂度、低功耗、低速率、低成本的双向无线通讯技术。

在 ZigBee 网络中存在三种逻辑设备类型: Coordinator(协调器), Router(路由器)和 End-Device(终端设备)。ZigBee 网络由一个 Coordinator 以及多个 Router 和多个 End_Device 组成。

各类型设备功能如下:

(1) Coordinator(协调器)

协调器负责启动整个网络。它也是网络的第一个设备。协调器选择一个信道和一个网络 ID(也称之为 PAN ID, 即 Personal Area Network ID), 随后启动整个网络。

协调器也可以用来协助建立网络中安全层和应用层的绑定(bindings)。

注意, 协调器的角色主要涉及网络的启动和配置。一旦这些都完成后, 协调器的工作就像一个路由器(或者消失 go away)。由于 ZigBee 网络本身的分布特性, 因此接下来整个网络的操作就不在依赖协调器是否存在。

(2) Router(路由器)

路由器的功能主要是: 允许其他设备加入网络, 多跳路由和协助它自己的由电池供电的儿子终端设备的通讯。

通常, 路由器希望是一直处于活动状态, 因此它必须使用主电源供电。但是当使用树群这种网络模式时, 允许路由间隔一定的周期操作一次, 这样就可以使用电池给其供电。

(3) End-Device(终端设备)

终端设备没有特定的维持网络结构的责任, 它可以睡眠或者唤醒, 因此它可以是一个电池供电设备。

1.2 产品特点

序号	产品特点	特点描述
1	集中式网络管理	ZIGBEE 3.0 安全标准集中式入网机制, 数据安全、可靠;
2	互通性	符合 zigbee 3.0 标准网络机制, 可兼容 ZCL 网络协议。
3	大容量	256K 容量的 flash, 32K 容量的 RAM, 最大网络节点数量可以扩展到 80 个;
4	角色切换	用户可通过串口指令让设备在协调器, 路由器, 终端和休眠终端的四种类型中任意切换;
5	支持多种网络拓扑	点对点, 星型网, MESH 网;
6	网络自愈	网络中间节点丢失, 自动形成新的 Mesh 拓扑, 丢失节点重启后自动找回;
7	自动路由	模块支持网络路由功能;
8	开放关闭网络组网	协调器控制开放或关闭网络, 开放网络的时间段内符合 ZigBee3.0 标准的设备可以加入网络, 关闭网络后任何设备无法加入。协调器开放网络后若未操作关闭, 180 秒后自动关闭。

9	一键加网	节点不需要设置 PANID 和信道, 只需要在协调器开放网络的窗口时间内触发加网即可。
10	自动信道和 PANID	协调器自动在最优信道创建网络, 并自动分配 PANID, 避开和其它协调器重复。
11	自动获取 MAC 地址	协调器可在节点加网瞬间获取到节点 MAC 地址和短地址, 不需要在设备端再做处理。
12	地址搜索	用户可根据已加入网络节点的 MAC 地址 (唯一的, 固定的) 查找出相应的短地址, 同时也可以根据节点的短地址查找网络中每个节点相应的长地址;
13	数据安全	集成 ZIGBEE 3.0 安全通讯标准, 网络含有多级安全密钥;
14	串口配置	模块内置串口指令, 用户可通过串口指令配置 (查看) 模块的参数及功能;
15	PWM 控制	本地/远程的 PWM 控制, 3 路 PWM 通道供用户选择;
16	一键恢复波特率	如果用户忘记或不知波特率的情况下, 可使用该功能, 恢复默认波特率为 115200bps;
17	串口接收唤醒	支持串口接收唤醒功能, 当模块处于休眠状态下当接收到一个任意字节的数据时将被唤醒, 此数据为唤醒帧用于唤醒模块将不会被当做数据处理;
18	模块复位	用户可通过串口命令对模块进行复位操作;
19	空中配置	用户可使用空中配置指令远程配置网络中的其他设备。
20	多种命令格式	用户可使用 16 进制格式命令和 AT 命令配置和控制模组, 实现组网, 设置透传, 控灯等操作。

1.3 支持产品系列

序号	产品型号	射频芯片	频率(Hz)	空速 (bps)	功率 (dBm)	天线形式
1	E180-ZG120A/B	EFR32MG1B232F256GM32	2.4G	250K	20	PCB

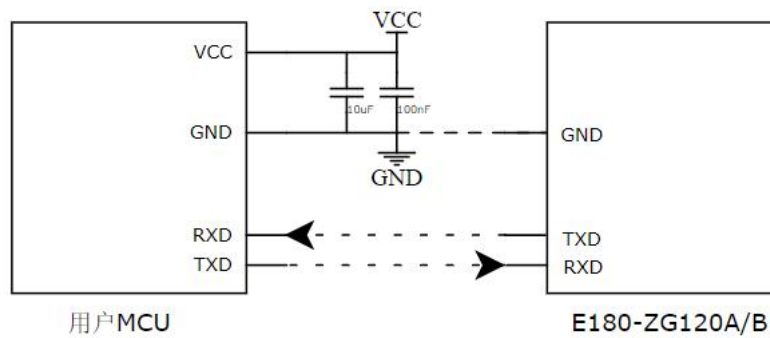
2. 功能及命令结构简介

2.1 功能引脚表

引脚封装参考硬件说明文档《E180-ZG120series_User Manual_CN_v1.0》。

2.2 引脚连接说明

2.2.1 串口连接说明



2.2.2 引脚位置说明

E180-ZG120A/B 组网模块采用 UART 串口通信方式, 用户可通过任意带 UART 功能的 MCU 与其连接, 进行数据交互, 具体引脚位置查看用户手册《E180-ZG120series_UserManual_CN_v1.0》, 具体连接方式如上图所示。

3. 模组固件功能

E180-ZG120A/B 模组支持 ZigBee 3.0 协议, 符合 ZCL (ZigBee cluster library) 标准规范, 可与其他厂商基于 ZigBee 的智能产品互操作。

该模组使用 UART 接口进行控制, 支持 HEX 指令配置与通信, AT 指令配置, 数据透传三种模式; 支持“一键组网”功能即通过 IO 口外接按键或电平信号实现组网控制; 模组可以配置成协调器、路由节点、普通终端节点、休眠终端节点四种 ZigBee 节点类型模式; 支持符合 ZCL 标准规范控制的 3 组 PWM 信号输出。

3.1 三种串口模式

3.1.1 HEX 指令模式

HEX 指令格式为“帧头+帧长+帧载荷”的固定模式, 指令帧长度可变, 指令输入不受指令粘包影响, 且输入指令有超时保护机制, 有效解决指令断包问题。每条输入指令都有对应的指令反馈用于确认模组是否正常工作以及是否正确执行指令。HEX 指令模式为全双工模式, 模组状态变化或收到数据均通过 UART_TX 口实时输出对应的 HEX 指令。

HEX 的格式与解析详见文档《亿佰特 ZigBee3.0 模组 HEX 命令标准规范》, 本文档重点强调 E180-ZG120 在 HEX 指令模式下的特性。

HEX 指令格式:

帧头 (1 字节)	帧长 (1 字节)	帧载荷 (变长 3~255 字节)			
		命令类型 (1 字节)	命令码 (1 字节)	命令数据 (变长 0~252 字节)	XOR 校验 (1 字节)

帧头: 十六进制的固定字节 0x55

帧长: 1 字节长度, 取值范围 3~255 (十六进制为 0x03~0xFF)

帧载荷: 帧载荷包含命令类型, 命令码, 命令数据和 XOR 校验, 长度由帧长决定。

命令类型: 根据命令的模式和工作机制, 进行分类。

命令码: 命令对应的编码, 长度 1 字节, 每条命令都有唯一的命令编码。

命令数据: 该命令执行的附带参数, 最小 0 字节, 最大 252 字节。

XOR 校验: 整个命令载荷 (命令类型, 命令编码, 命令数据) 的 XOR8 校验和。

三种 HEX 指令类型

输入命令:

上位机输入模组的命令, 可用于配置模组或无线发送。输入命令的命令类型小于 0x0F。

反馈命令:

模组收到并执行上位机命令后, 反馈执行结果给上位机。反馈命令的命令类型和命令码与输入命令相

同。

异步命令:

模组运行过程中主动发给上位机的命令, 该命令对应 ZigBee 应用中的异步事件。异步命令的命令类型大于 0x80。

输入命令 (含反馈命令)

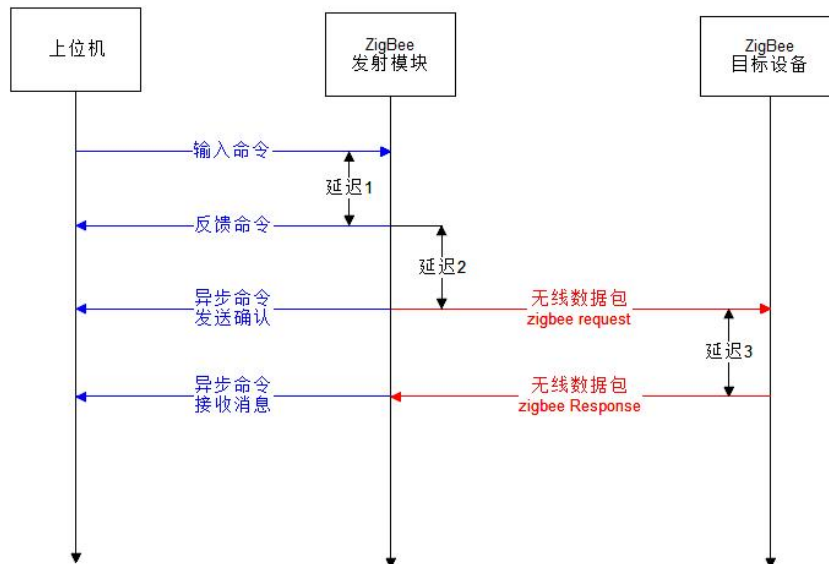
- 本地配置命令, 命令类型 0x00, 用于模组的本地设置。
- 网络管理命令, 命令类型 0x01, 用于组网时对其它模组进行网络层的管理。
- ZCL 发送命令, 命令类型 0x02, 用于模组对其它模组或第三方设备的控制, 符合 ZCL 规范。

异步命令:

- 系统通知命令, 命令类型 0x80, 模组状态变化通知。
- 网络管理返回命令, 命令类型 0x81, 其它模组或设备收到网关管理命令返回消息。
- ZCL 接收命令, 命令类型 0x82, 模组收到其它模组或设备的 ZCL 层消息或返回消息。
- 发送确认, 命令类型 0x8F, 用于诊断网络管理命令和 ZCL 发送命令的发送是否异常。

关于“发送确认”的正确使用方式:

(上位机) 向模组输入网络管理命令和 ZCL 发送命令并收到对应的反馈命令, 命令并未立即转化为无线信号发送出去, 而是在避让同网络中其它设备后再以 250kbps 的速率向空气中发射信号, “发送确认”命令即为无线信号的发送结果。在 E180ZG120 模组上发送确认只有两种状态: 0x00 = 发送成功, 0x66 = 发送失败。网络管理命令的最终正确性根据收到的对应网络管理返回命令来判断; ZCL 发送命令的最终正确性也根据收到的对应的 ZCL 接收命令来判断。发送确认可以用于提前结束等待返回消息, 以及可用于诊断异常的其它模组或设备, 放在将有限的网络资源浪费在无意义的设备节点上。



E180ZG120 模组在如果以广播的方式发送网络管理命令和 ZCL 发送命令时, 由于受广播洪泛(flooding)的影响, 发送确认会在反馈命令的 1 秒后才触发。因此使用该模组进行广播或组播时建议发送间隔周期大于 1 秒。

E180ZG120 支持的 HEX 命令目录

命令名称	命令类型	命令码
本地配置类		

查询模组当前状态	0x00	0x00
开始配网	0x00	0x02
停止配网	0x00	0x03
复位/恢复出厂	0x00	0x04
设置本机节点类型	0x00	0x05
查看本机加组	0x00	0x09
本机加组	0x00	0x0A
本机退组	0x00	0x0B
设置和查询当前发射功率	0x00	0x0D
读取本地属性	0x00	0x10
设置本地属性	0x00	0x11
自动绑定目标	0x00	0x14
进入 AT 命令模式	0x00	0x16
读取入网节点地址表	0x00	0x22
系统通知类		
模组启动	0x80	0x00
网络状态变更	0x80	0x01
打开关闭网络通知	0x80	0x02
检测节点入网	0x80	0x03
节点短地址通知	0x80	0x04
节点离网通知	0x80	0x06
自动绑定目标结果通知	0x80	0x10
ZDO 网络管理命令/网络管理返回		
查询节点短地址/返回	0x01/0x81	0x00
查询节点 MAC 地址/返回	0x01/0x81	0x01
查询节点端口信息/返回	0x01/0x81	0x04
查询节点端口数/返回	0x01/0x81	0x05
设置节点常连接绑定/返回	0x01/0x81	0x21
解除节点常连接绑定/返回	0x01/0x81	0x22
查看节点常连接绑定/返回	0x01/0x81	0x33
删除节点/返回	0x01/0x81	0x34
ZCL 命令与 ZCL 返回		
读取设备属性/返回	0x02/0x82	0x00
修改设备属性/返回	0x02/0x82	0x01
查询属性上报规律/返回	0x02/0x82	0x02
修改属性上报规律/返回	0x02/0x82	0x03
查看全部属性/返回	0x02/0x82	0x04
查看全部状态带扩展字段/返回	0x02/0x82	0x05
收到属性主动上报	0x82	0x0A
默认返回帧	0x82	0x0B
发送控制命令	0x02	0x0F
收到控制命令	0x82	0x0F

3.1.2 AT 命令模式

AT 命令为 ASCII 字符串格式,方便人工直接输入和助记。AT 命令采用“AT+命令码”的格式,命令码为固定字符串,AT 命令的详细解析见《亿佰特 ZigBee 3.0 模组 AT 命令标准规范》。AT 命令一共有三种输入形式:执行式、查询式、设置式。

执行式:

执行式命令的格式为“AT+命令码”的直接格式,执行命令的以命令码最后一个字节结束,后面不延续任何字节,包括回车符号也不允许出现,否则输入无效。如“AT+JOIN”、“AT+LEAVE”。执行命令输入有效返回“OK\r\n”,即收到一个带回车符结尾的“OK”。若输入命令结尾不正确,模组返回“INVALID\r\n”。

查询式:

查询式命令的格式为“AT+命令码?”的格式,即以 ASCII 的“?”(16 进制 0x3F)结束。查询命令用来查询模组当前某项参数的值,并以 ASCII 的格式通过串口打印查询值。

设置式:

设置式命令的格式为“AT+命令码=数值”的格式,即命令码结束后需要跟上 ASCII 的“=”,并在“=”后面跟上数值。数值根据不同命令,输入方式分别有 10 进制或 16 进制,以%d 或%x 的格式表示,如果输入多个参数需要用“,”隔开。详见《亿佰特 ZigBee 3.0 模组 AT 命令标准规范》。

AT 命令目录:

命令功能	命令码	执行	查询	设置
退出 AT 模式到 HEX 模式	AT+EXIT	Y	N	N
组网或创建新网络	AT+JOIN	Y	N	N
停止组网	AT+STOP	Y	N	N
模组复位	AT+RESET	Y	N	N
离开网络	AT+LEAVE	Y	N	N
进入透传模式	AT+SEND	Y	N	N
自动绑定目标	AT+FIND	Y	N	N
读取设备信息	AT+INFO	Y	N	N
设置或读取设备类型	AT+DEVTYPE	N	Y	Y
设置或读取波特率	AT+BAUD	N	Y	Y
设置或读取目标地址	AT+DSTADDR	N	Y	Y
设置或读取目标端口	AT+DSTEP	N	Y	Y
接通开关	AT+TURNON	Y	Y	Y
断开开关	AT+TURNOFF	Y	Y	Y
反置开关	AT+TOGGLE	Y	Y	Y
亮度升高	AT+LEVELUP	Y	Y	Y
亮度降到	AT+LEVELDOWN	Y	Y	Y
设置亮度	AT+LEVELSET	Y	Y	Y

标记目标	AT+IDENTIFY	Y	Y	Y
解除绑定	AT+UNBIND	Y	Y	Y

3.1.3 数据透传模式

数据透传模式下, 输入到串口的任何数据都会通过无线信号发送出去, 收到透传数据的模组在透传模式或 AT 命令模式下直接打印该数据帧, 如接收端在 HEX 指令模式下则以 ZCL 命令格式输出该帧数据。

数据透传的 ZCL 格式:

数据透传符合 ZCL 规范, 其规范如下

- 端口=1
- Profile = 0x0104
- cluster=0xFC08
- manufacture code=0x2000
- 命令类型: Special Command
- 命令方向: Server to Client
- 命令 ID: 0x00

备注: 协调器或其它节点处于 HEX 模式下给透传模式的 E180ZG120 模组发送数据, 也需要遵循该 ZCL 规范, 即使用 cluster=0xFC08 的簇, Manufacture Code=0x2000, 命令方向更改为 Client to Server, 命令 ID 为 0x00。

数据透传的目标设置

数据透传目标设置模组本地属性 DstAddr 和本地属性 DstEP, 该两个属性位于模组的端口 1, cluster=0xFC08。属性 ID 分别为 0x0001 和 0x0002, 数据类型分别为 UINT16 和 UINT8。

DstAddr 即为透传接收的短地址, DstEP 为透传接收的目标端口。透传具有点播发送, 广播发送, 组播发送, 绑定发送 4 种模式。DstEP 设置为其它值时用于预留带多串口的 ZigBee 模组透传。

透传模式	DstAddr	DstEP
点播 (到主串口)	对方短地址	1
点播 (到串口 2) 备注: 预留双串口功能	对方短地址	2
广播	0xFFFF	0xFF
组播	16bit 组地址	0
绑定发送	0xFFFE	0xFE

绑定透传目标设置

绑定透传模式下, 模组可通过 MAC 地址寻找透传目标, 以应对透传目标短地址发生变化。设置绑定有 3 种方式:

- a) 协调器通过 HEX 指令“设置节点常连接绑定 (命令码 0x21)”, 为模组指派透传目标, 若模组已知对方 MAC 地址, 也可在 HEX 命令模式下向自己发送该命令。

- b) 两个模组都在 HEX 命令模式下, 发送本地配置命令“自动绑定目标(命令码 0x14)”。透传目标模组先发, 等待约 1~3 秒 LINK 指示灯闪烁, 透传源模组再发送该命令。
- c) 在 AT 命令模式下, 两个模式使用“AT+FINDD”命令互绑, 操作方式和本地配置命令“自动绑定目标(命令码 0x14)”相同。
- d) 任何模式下, 两个需要互绑的模组的 PD13 引脚上输入一个按键信号(下降沿 20ms~200ms), 操作方式和本地配置命令“自动绑定目标(命令码 0x14)”相同。

数据透传的反馈信息

E180ZG120 模组在透传模式下发送数据会有结果反馈, 反馈信息小于等于 4 字节, 为区分反馈信息和收到的透传数据, 建议透传数据大于等于 5 字节。反馈信息如下

“OK”: 发送成功

“FAIL”: 发送失败

“ERRO”: 发送错误, 如发送缓存满, 模组未组网, 模组离线

“OFF”: 模组掉线, 终端节点和休眠终端会出现该现象

“NET”: 模组重新上线, 终端节点和休眠终端会出现该现象

3.1.4 三种模式的相互切换

三种模式的相互转换见表格

		目标模式		
		HEX 命令	AT 命令	透传模式
当前模式	HEX 命令		配置命令“进入 AT 命令模式”	配置命令“设置本地属性” 属性 0x0003 设置为“1”
	AT 命令	“AT+EXIT”		“AT+SEND”
	透传模式	发送 3 个字符 “+++”	发送 3 个字符“+AT”	

示例:

HEX 命令模式切换 AT 命令模式:

指令	应答
55 03 00 16 16	55 04 00 16 00 16
说明: 无	

AT 命令模式切换 HEX 命令模式:

指令	应答
AT+EXIT	OK
说明: 此处应答以\r\n收尾, \r\n(回车换行)	

HEX 命令模式切换透传模式:

指令	应答
55 07 00 11 00 03 00 01 13	55 04 00 11 00 11
说明: 无	

透传模式切换 HEX 命令模式:

指令	应答
+++	55 0D 80 00 00 10 E9 CE D6 FE FF 14 43 0C 3B
说明: E9 CE D6 FE FF 14 43 0C 3B 为 MAC 地址。	

AT 命令模式切换透传模式:

指令	应答
AT+SEND	SEND_MODE
说明: 此处应答以\r\n收尾, \r\n(回车换行)	

透传模式切换 AT 命令模式:

指令	应答
+AT	AT_MODE
说明: 此处应答以\r\n收尾, \r\n(回车换行)	

出厂默认模式

恢复出厂后, 模组为 HEX 命令模式。

复位后的模式

在任何非 HEX 命令模式下复位, 若模组已入网则默认为数据透传模式, 模组未入网默认为 AT 命令模式。

3.2 休眠节点串口唤醒

模组配置成休眠模式后, 如需要对其发送串口指令, 需要先发送唤醒帧。休眠终端的串口 RX 引脚收到任何信号 (主要是下降沿信号) 后从休眠状态唤醒, 唤醒信号后的 0.5ms~200ms 内, 串口 RX 能正常运行, 上位机需在该时间内写入串口指令或透传数据。目前有以下 3 种方式实现低功耗模式下的串口唤醒。

3.2.1 延迟法

向休眠节点发送串口命令, 先发送一个 0x00 的单字节, 然后在 1ms~200ms 内发送 HEX 命令或 AT 命令或透传报文。

3.2.2 连续点击法

该方法适合手动操作上位机，向休眠节点发送 HEX 命令或 AT 命令。先操作上位机向模组发送一条命令，该命令用于唤醒模组，但该命令接收不完整导致模组收到错误命令。然后需要再 200 毫秒内向休眠模组发出第二条命令。该方法不适合透传数据，因为透传数据没有错误检测，会导致错误的数据包被发送出去。

3.2.3 插入法

该方法适合 HEX 命令模式，AT 命令模式，数据透传模式。在命令或数据前插入若干字节 0x00，用于唤醒串口。插入 0x00 的个数由串口波特率决定，累计凑满 0.5ms 的波特率。因为模组在唤醒后 0.5ms 后 RX 开始生效，因此如果是 AT 模式或透传模式，插入的 0x00 既不能多，也不能少。

波特率	插入 0x00 字节数
230400	12
115200	6
57600	3
38400	2
19200	1
9600	1

3.3 模组的 ZCL 标准规范

E180-ZG120A/B 模组具备 4 个 APS 层端口，端口 1 为透传控制，端口 2、3、4 为 PWM 输出控制，其规格如下。

3.3.1 E180-ZG120A/B ZCL 规范功能分配表

端口号	轮廓(Profile)	设备 ID	输入簇(in cluster)	输出簇(out cluster)
1	0x0104 HA 协议	0x0050 网关类设备	0x0000: BASIC 0x0003: IDENTIFY 0x0004: GROUP 0x0007: ONOFF SWITCH 0xFC08: EBYTE_TRANS	0x0003: IDENTIFY 0x0006: ONOFF 0x0008: LEVEL
2	0x0104 HA 协议	0x0101 可调光灯	0x0003: IDENTIFY 0x0004: GROUP 0x0005: SCENE 0x0006: ONOFF	无

			0x0008: LEVEL	
3	0x0104 HA 协议	0x0101 可调光灯	0x0003: IDENTIFY 0x0004: GROUP 0x0005: SCENE 0x0006: ONOFF 0x0008: LEVEL	无
4	0x0104 HA 协议	0x0101 可调光灯	0x0003: IDENTIFY 0x0004: GROUP 0x0005: SCENE 0x0006: ONOFF 0x0008: LEVEL	无

3.3.2 亿佰特透传协议的 ZCL 规范

亿佰特 ZigBee 模组透传协议完全遵循 ZCL 规范, 使用了自定义的簇 ID (cluster ID) 和自定义厂商码 (Manufacture Code), 丰富了 ZCL 标准规范库。该模组可以接入到其它厂商的 ZigBee 网络中, 除完成数据透传功能外, 还可实现控灯和开关控制功能。

簇规范:

cluster ID = 0xFC08, Manufacture Code=0x2000

属性表:

AttrID	方向	描述符	名称	数据类型	操作
0x0000	Server	Baud	波特率	uint32	只读
0x0001	Server	targetAddr	默认目标短地址	uint16	读写
0x0002	Server	targetEP	默认目标端口	uint8	读写
0x0003	Server	sendMode	透传模式	bool	读写
0x0004	Server	LP Level	低功耗模式	enum8	只读+上报
0x0005	Server	target IEEE	目标 MAC 地址显示	EUI64	只读

ZCL 控制命令 (Special Command) 表:

cmdID	方向	描述符	名称	参数
0x00	C->S	UartSend	透传数据发送	uint8 data[]: 透传数据
0x00	S->C	UartNotify	透传数据通知	uint8 data[]: 透传数据
0x01	C->S	SetDstAddr	设置默认目标	uint16 dstAddr: 目标短地址 uint8 endpoint: 目标端口
0x01	S->C	SetDstAddrRsp	设置默认目标返回	uint8 status: ZCL 状态
0x02	C->S	SetBaud	设置波特率	uint32 baud: 设置的新波特率, 重启生效
0x02	S->C	SetBaudRsp	设置波特率返回	uint8 status: ZCL 状态
0x03	C->S	SetLP_Level	设置低功耗模式	uint8 LP_level: 低功耗等级

0x03	S->C	SetLP_LevelRsp	设置低功耗返回	uint8 status: ZCL 状态
0x04	C->S	Reset	模组重启	uint8 extAddr[8]: 模组的 MAC 地址

- 透传模组发送透传使用“透传数据通知”，网关或协调器给透传模组发数据使用“透传数据发送”。
- 波特率只支持 9600、19200、38400、57600、115200、230400 共 6 种模式
- 低功耗模式的值为 0~3，分别对应 0 = 1 秒唤醒、1 = 3 秒唤醒、2 = 5 秒唤醒、3 = 1 分钟唤醒。

3.4 功能按键、状态指示灯与 PWM 输出

E180ZG120 模组具有 2 组按键输入 IO 口，1 组 LED 闪烁指示 IO 口，3 组 PWM 输出 IO 口。

3.4.1 管脚分配

网络功能键 -> PD_13
波特率复位键 -> PB_11
状态指示灯 -> PF_7 (E180-ZG120A)
端口 0 PWM -> PF2 (E180-ZG120A)
端口 1 PWM -> PF3 (E180-ZG120A)
端口 2 PWM -> PF4 (E180-ZG120A)
状态指示灯 -> PF_3 (E180-ZG120B)
端口 0 PWM -> PB_14 E180-ZG120B)
端口 1 PWM -> PB_15 (E180-ZG120B)
端口 2 PWM -> PB_13 (E180-ZG120B)
UART_RX -> PA_1
UART_TX -> PA_0

3.4.2 网络功能键功能介绍

一键入网功能:

已配置成路由节点，终端节点和休眠终端节点的模组，在未入网或离开网络的情况下短按（小于 1 秒）网络功能键再松开，可以加入一个网络打开的协调器中。

如果模组配置成协调器，协调器未建立网络时短按网络功能键则直接新建网络，已有网络的协调器短按该键则开始配网，已开始配网的协调器短按此键则关闭网络。

自动绑定功能:

两个已入网的任意节点，各自先后短按此键（先后间隔 3 秒，待先按的模组状态指示灯闪烁），自动绑定数据透传；如需绑定控灯或开关，可先让灯或开关进入 Identify 状态，再短按模组上此键，绑定成功后模组可通过 AT 命令控制灯或开关，并接收灯和开关的状态属性上报。绑定灯或开关时可一次设置多个灯或开关进入 Identify 状态，模组一次按键可同时绑定多个灯或开关。协调器不支持自动绑定功能。

离网与恢复出厂:

已入网的模组, 长按此键 5 秒钟, 待状态指示灯常亮, 松开按键, 模组退出当前网络。模组在已退网的状态下长按此键 5 秒, 待状态指示灯常亮, 则恢复出厂设置。(离网的模组不能通过任何命令恢复出厂, 可通过该按键恢复出厂)

3.4.3 状态指示灯: (建议该引脚外接低电平驱动 LED)

该引脚平时均为高电平, 在以下事件发生时, 会输出连续变化的高低电平。

节点入网状态:

模组配置成路由, 终端节点, 休眠终端节点, 在第一次入网成功后, 该引脚输出 166ms 的低电平 3 次, 两个低电平之间间隔 166ms。

协调器创建网络成功:

协调器第一次创建网络成功, 该引脚输出 166ms 的低电平 3 次, 两个低电平之间间隔 166ms。

网络打开状态:

协调器开始配网, 或路由节点的网络被协调器打开, 该引脚输出 500ms 的低电平, 且间隔 500ms 再输出一低电平, 直到网络被关闭(含手动关闭和自动关闭)才保持高电平。

Identify 标记状态:

路由器或终端节点在被 Identify 标记状态下, 该引脚输出 500ms 的低电平, 直到 Identify 标记结束。该功能可用于目测某个 MAC 地址或短地址的模组在哪个空间位置。

注意事项:

路由节点的网络打开状态和 Identify 标记状态, 该引脚输出信号雷同, 因此在实际操作时应避免两种操作同时进行。

4. 应用配置示例

4.1 示例 1: 协调器 HEX 命令控制入网节点

示例简介:

E180ZG120 模组设置成协调器并创建一个网络。另一个 E180ZG120 模组(普通终端节点+透传模式)加入这个协调器中, 并完成设备识别, 数据透传控制和 PWM 控制。

4.1.1 E180-ZG120A/B 协调器创建网络

本阶段 E180-ZG120A/B 模组恢复出厂模式后, 设置成协调器, 启动并新建一个网络。

第一步: 查询模组当前是否已组网

发送配置命令“查询模组当前状态(类型 0x00、码 0x00)”:

```
55 03 00 00 00
```

收到“查询模组当前状态”反馈:

```
55 2A 00 00 00 02 6E 93 50 FE FF 14 2E 84 0F 7E CC D8 2B FB FD FE 7F BF DF 6F 37 57 AE 5C 01 D4  
35 FA D4 D3 9A 23 47 A2 3F 2D 30 27
```

解析串口反馈:

网络状态: 已组网

设备类型: 普通终端节点

MAC 地址: 6E 93 50 FE FF 14 2E 84

信道: 0x0F = 15 信道

PANID: 0xCC7E

第二步: 模组恢复出厂

发送配置命令“复位/恢复出厂(类型 0x00、码 0x04)”:

```
55 07 00 04 02 7E CC 0F BB
```

解析发送命令

复位模式: 0x02 - 恢复出厂, 即需要擦除 FLASH 保存的全部参数

PANID: 0xCC7E

信道: 0x0F - 15 信道

收到“复位/恢复出厂”反馈命令:

```
55 04 00 04 00 04
```

等待 2 秒, 模组重启。

收到异步命令系统通知“设备启动(类型 0x80、码 0x00)”:

```
55 0D 80 00 00 00 6E 93 50 FE FF 14 2E 84 92
```

解析异步反馈:

复位模式: 0x00 - 在 E180ZG120 上是看门狗软复位

版本: 0x00

MAC 地址: 6E 93 50 FE FF 14 2E 84

第三步: 设置设备类型为协调器并重启

发送配置命令“设置节点类型(类型 0x00、码 0x05)”:

```
55 04 00 05 00 05
```

解析发送命令

节点类型: 0x00 - 协调器

收到“设置节点类型”的反馈命令:

```
55 04 00 05 00 05
```

发送配置命令“复位/恢复出厂(类型 0x00、码 0x04)”:

```
55 07 00 04 00 FF FF 00 04
```

解析发送命令

复位模式: 0x00 - 直接复位

PANID: 0xFFFF - 不退网直接复位

信道: 0xFF - 不退网直接复位

收到“复位/恢复出厂”的反馈命令:

55 04 00 04 00 04

等待 1 秒后收到异步命令系统通知“设备启动 (类型 0x80, 码 0x00)”:

55 0D 80 00 00 00 6E 93 50 FE FF 14 2E 84 92

☆. 收到“设备启动通知”表示协调器设置成功

第四步: 协调器配网

发送配置命令“开始配网 (类型 0x00, 码 0x02)”:

55 03 00 02 02

收到“开始配网”的反馈命令:

55 04 00 02 00 02

解析反馈命令:

状态: 0x00-成功

等待协调器模组创建新网络 (本次等待 2 秒)

收到异步命令系统通知“网络状态变更 (类型 0x80, 码 0x01)”:

55 29 80 01 01 6E 93 50 FE FF 14 2E 84 14 BB B3 00 00 76 8E 17 47 E7 A7 27 27 E9 AB DE D6 A7 AD
D3 8C A0 FC A5 3E 7A C5 CC 2B E6

解析异步命令:

网络状态 - 0x01 已组网

MAC 地址: 6E 93 50 FE FF 14 2E 84

信道: 0x14 - 20 信道

PANID: 0xB3BB

短地址: 0x0000 - 协调器地址

扩展 PANID: 76 8E 17 47 E7 A7 27 27

网络密钥: E9 AB DE D6 A7 AD D3 8C A0 FC A5 3E 7A C5 CC 2B

继续收到异步命令系统通知“打开关闭网络通知 (类型 0x80, 码 0x02)”

55 04 80 02 B4 36

解析异步命令:

允许入网时间: 0xB4 -180 秒, 若再次收到该命令且打开时间变成 0, 则表示网络已关闭。

4.1.2 新节点加入网络以及新节点的识别和配置

本阶段将一个设置成普通终端节点+透传模式的 E180ZG120 模组加入到协调器网络中, 入网模组使用一键入网。入网后完成对入网模组的识别。

第一步: 协调器再次配网

发送配置命令“开始配网 (类型 0x00, 码 0x02)”:

55 03 00 02 02

收到“开始配网”的反馈命令

55 04 00 02 00 02

然后收到异步命令系统通知“打开关闭网络通知 (类型 0x80, 码 0x02)”

55 04 80 02 B3 31

第二步: 接受设备加入网络

☆. 另一个 E180ZG120 模组 (普通终端节点+透传模式) 需要在协调器允许入网的 180 秒内, 短按网络功能键, 加入该协调器。以下是协调器的串口指令操作。

收到异步命令系统通知 “检测节点入网 (类型: 0x80, 码 0x03) ”:

```
55 10 80 03 C1 5E CA FE FF 5F 32 50 C5 B9 00 00 00 96
```

解析异步命令:

MAC 地址: C1 5E CA FE FF 5F 32 50

短地址: 0xB9C5

父节点短地址: 0x0000

紧接着收到异步命令系统通知 “节点短地址通知 (类型 0x80, 码 0x04) ”:

```
55 0E 80 04 C1 5E CA FE FF 5F 32 50 C5 B9 02 93
```

解析异步命令

MAC 地址: C1 5E CA FE FF 5F 32 50

短地址: 0xB9C5

节点类型: 0x02 - 普通终端节点

第三步: 查看入网节点的端口数量

☆. 该步查询入网节点的应用层端口有多少个。

发送网络管理命令 “查询节点端口数 (类型 0x01, 码 0x05) ”

```
55 05 01 05 C5 B9 78
```

解析发送命令:

目标地址: 0xB9C5

收到 “查询节点端口数” 反馈

```
55 05 01 05 00 02 06
```

解析反馈命令:

执行状态: 0x00 - 成功

命令编号: 0x02, 该命令号为模组的系统分配

收到异步命令 “网络管理命令的发送确认”

```
55 07 8F 01 C5 B9 02 00 F0
```

解析发送确认:

目标地址: 0xB9C5

命令编号: 0x02

发送结果: 0x00 - 发送成功

收到异步命令 “查询节点端口数响应 (类型 0x81, 码 0x05) ”

```
55 0C 81 05 C5 B9 02 00 04 01 02 03 04 FA
```

解析异步响应命令

对方短地址: 0xB9C5

命令编号: 0x02

执行结果: 0x00

端口数: 0x04

端口列表: 0x01、0x02、0x03、0x04

第四步: 查看端口 1 的详细信息

发送网络管理命令 “查询节点端口信息 (类型 0x01, 码 0x04) ”:

```
55 06 01 04 C5 B9 01 78
```

解析发送命令:

目标短地址: 0xB9C5

查询端口: 0x01

收到“查询节点端口信息”的反馈

55 05 01 04 00 03 06

解析反馈命令:

执行状态: 0x00 - 成功

命令编号: 0x03

收到异步命令“查询节点端口信息响应(类型 0x81, 码 0x04)”

55 21 81 04 C5 B9 03 00 01 04 01 50 00 00 05 00 00 03 00 04 00 07 00 08 FC 04 03 00 06 00 08 00 08 FC A2

解析异步响应命令:

对方短地址: 0xB9C5

命令编号: 0x03

执行结果: 0x00

端口号: 0x01

端口轮廓: 0x0104

设备 ID: 0x0050 (网关控制类设备)

设备版本: 0x00

输入簇数量: 0x05

输入簇列表: 0x0000、0x0003、0x0004、0x0007、0xFC08

输出簇数量: 0x04

输出簇列表: 0x0003、0x0006、0x0008、0xFC08

☆. 此处响应命令和发送确认发生颠倒是因为 Silicon Labs 系统在发送命令时使能了 APS ACK, 让命令传输更可靠, 属正常现象。

收到异步命令“网络管理命令的发送确认”

55 07 8F 01 C5 B9 03 00 F1

解析发送确认:

目标地址: 0xB9C5

命令编号: 0x03

发送结果: 0x00 - 发送成功

第五步: 查看端口 2 的详细信息, 端口 3 和端口 4 操作方法雷同

发送网络管理命令“查询节点端口信息(类型 0x01, 码 0x04)”查询端口 2:

55 06 01 04 C5 B9 02 7B

收到该命令的反馈, 命令编号为 0x04:

55 05 01 04 00 04 01

收到异步命令“查询节点端口信息响应(类型 0x81, 码 0x04)”

55 19 81 04 C5 B9 04 00 02 04 01 01 01 00 05 03 00 04 00 05 00 06 00 08 00 00 F3

解析端口 2 的异步响应命令:

对方短地址: 0xB9C5

命令编号: 0x04

执行结果: 0x00

端口号: 0x02

端口轮廓: 0x0104
设备 ID: 0x0101 (可调光照明类设备)
设备版本: 0x00
输入簇数量: 0x05
输入簇列表: 0x0003、0x0004、0x0005、0x0006、0x0008
输出簇数量: 0x00

☆. 通过查询 ZCL 规范, 端口 2 为照明设备, 支持分组, 场景控制, 开关控制, 亮度调节
收到异步命令“网络管理命令的发送确认”

```
55 07 8F 01 C5 B9 03 00 F1
```

☆. 使用与查询端口 2 相同的方式, 查询端口 3 和端口 4, 获得其功能信息。

发送确认的建议使用方式:

在基于 Silicon 的 ZigBee 模组或设备上, 存在响应命令和发送确认命令颠倒的情况。建议主机在处理信息时仅处理响应命令, 发送确认命令可作为等待响应命令的超时结束等待判断依据; 也可将发送确认作为查询单个节点的阻塞条件, 在查询某个节点时未收到响应命令或发送确认任中一种命令时, 不得向该目标发送后续指令。

第六步: 设置入网节点的属性上报的绑定

☆. 端口 1 有输入簇 0xFC08, 端口 2、端口 3、端口 4 有输入簇 0x0006 和 0x0008, 使用网络管理命令“设置节点常连接绑定”将其绑定到协调器的串口控制端口(端口 1)。

发送网络管理命令“设置节点常连接绑定(类型 0x01, 码 0x21)”:

```
55 19 01 21 C5 B9 01 C1 5E CA FE FF 5F 32 50 08 FC 01 6E 93 50 FE FF 14 2E 84 D3
```

解析发送命令

目标短地址: 0xB9C5

源虚拟 SN: 01 C1 5E CA FE FF 5F 32 50, 入网节点的端口 1 的虚拟 SN

簇 ID: 0xFC08, 亿佰特串口透传功能簇

目标虚拟 SN: 01 6E 93 50 FE FF 14 2E 84, 协调器的串口端口虚拟 SN

收到该命令的反馈

```
55 05 01 21 00 08 28
```

收到异步命令“设置节点常连接绑定响应(类型 0x01, 码 0x21)”:

```
55 07 81 21 C5 B9 08 00 D4
```

解析异步响应命令

对方短地址: 0xB9C5

命令编号: 0x08

执行结果: 0x00 - 绑定成功

收到异步命令“发送确认”:

```
55 7 8F 01 C5 B9 08 00 FA
```

☆. 设置绑定时, 也发生了响应命令和发送确认颠倒的现象。按照该格式, 继续绑定入网节点的端口 2、端口 3、端口 4 的簇 0x0006 和簇 0x0008。

4.1.3 控制入网节点

第一步: 接收入网节点的属性上报

☆. 节点的属性上报可作为心跳包使用

收到 ZCL 异步命令“属性主动上报 (类型 0x82, 码 0x0A)”:

```
55 13 82 0A 00 C5 B9 01 07 01 08 FC 00 20 DF 01 04 00 30 00 CD
```

解析收到的异步命令:

接收端口: 0 - 模组的默认端口 (端口 1) 收到该消息

接收模式: 0 - 点播模式, 无法判断信号强度是否有效

源短地址: 0xB9C5

源端口: 0x01

帧序号: 0x07

命令方向: 0x01 - Server To Client

簇 ID: 0xFC08

厂商码: 0x2000

信号强度: 0xDF = -33dbm

上报属性总数: 0x01

上报属性 ID: 0x0004 入网模组的低功耗等级

属性数据类型: 0x30 = enum8 型

数据值: 0x00

收到其它端口的 ZCL 异步命令“属性主动上报 (类型 0x82, 码 0x0A)”:

```
55 13 82 0A 00 C5 B9 02 16 01 06 00 00 00 D9 01 00 00 10 01 2E
```

```
55 13 82 0A 00 C5 B9 02 17 01 08 00 00 00 DA 01 00 00 20 FF EC
```

```
55 13 82 0A 00 C5 B9 03 18 01 06 00 00 00 DA 01 00 00 10 01 22
```

```
55 13 82 0A 00 C5 B9 03 19 01 08 00 00 00 DA 01 00 00 20 FF E3
```

```
55 13 82 0A 00 C5 B9 04 1A 01 06 00 00 00 DA 01 00 00 10 01 27
```

```
55 13 82 0A 00 C5 B9 04 1B 01 08 00 00 00 DA 01 00 00 20 FF E6
```

解析收到的第一条异步命令:

接收端口: 0 - 端口 1

接收模式: 0 - 点播模式

源短地址: 0xB9C5

源端口: 0x02

帧序号: 0x16

命令方向: 0x01 - Server To Client

簇 ID: 0x0006

厂商码: 0x0000

信号强度: 0xD9 = -39dbm

上报属性总数: 0x01

上报属性 ID: 0x0000 PWM 输出开关状态

属性数据类型: 0x10 = bool 型

数据值: 0x01 PWM 输出打开

解析收到的第二条异步命令:

接收端口: 0 - 端口 1

接收模式: 0 - 点播模式

源短地址: 0xB9C5

源端口: 0x02

帧序号: 0x17

命令方向: 0x01 - Server To Client
簇 ID: 0x0008
厂商码: 0x0000
信号强度: 0xDA = -38dbm
上报属性总数: 0x01
上报属性 ID: 0x0000 PWM 输出亮度
属性数据类型: 0x20 = uint8 型
数据值: 0xFF PWM 输出占空比为 255/255, 即 100%。

第二步: 读取入网节点的属性

☆. 入网节点的端口 1 支持簇 ID=0x0000, 该簇下有节点的身份信息相关属性

发送 ZCL 命令 “读取设备属性 (类型 0x02, 码 0x00) ”

```
55 1F 02 00 00 C5 B9 01 A1 00 00 00 00 00 00 08 00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 D6
```

解析发送命令:

发送端口: 0 - 默认端口 (端口 1)
发送模式: 0 - 普通发送, 不加密
目标短地址: 0xB9C5
目标端口: 0x01
帧序号: 0xA1
命令方向: 0x00 - Client To Server
簇 ID: 0x0000
厂商码: 0x0000
应答模式: 0x00 - 使用 Default Response, 不开启 APS Ack
读取属性总数: 0x08
属性 ID 列表: 0x0000、0x0001、0x0002、0x0003、0x0004、0x0005、0x0006、0x0007

收到命令反馈:

```
55 05 02 00 00 A1 A3
```

收到 ZCL 命令的发送确认 (类型 0x8F, 码 0x02):

```
55 0A 8F 02 00 C5 B9 01 A1 00 00 51
```

解析发送确认:

发送端口: 0 - 默认端口 (端口 1)
发送模式: 0 - 普通发送, 不加密
目标短地址: 0xB9C5
目标端口: 0x01
帧序号: 0xA1
命令方向: 0x00 - Client To Server
发送结果: 0x00 - 发送成功

收到 ZCL 异步命令 “读取设备属性响应 (类型 0x82, 码 0x00) ”

```
55 4F 82 00 00 C5 B9 01 A1 01 00 00 00 00 D6 08 00 0000200801 0000201002 0000200003 0000200004  
0000420545 42 59 54 4505 0000420B 46 57 37 34 32 31 2D 30 2D 31 3006 00004208 32 30 32 32 30 39  
31 3607 00003000 8E
```

解析 ZCL 异步命令: 属性 ID 加下划线, 红色 ZCL 状态, 绿色属性数据类型, 蓝色属性数据值

接收端口: 0 - 端口 1
接收模式: 0 - 点播模式

源短地址: 0xB9C5

源端口: 0x01

帧序号: 0xA1

命令方向: 0x01 - Server To Client

簇 ID: 0x0000

厂商码: 0x0000

信号强度: 0xD6 = -42dbm

读到属性数量: 0x08

读到属性列表:

属性 0x0000 (ZigBee 版本) 有效, 数据类型 uint8, 值为 0x08

属性 0x0001 (软件版本) 有效, 数据类型 uint8, 值为 0x10

属性 0x0002 (协议版本) 有效, 数据类型 uint8, 值为 0x00

属性 0x0003 (硬件版本) 有效, 数据类型 uint8, 值为 0x00

属性 0x0004 (厂商名称) 有效, 数据类型 string, 值为 5 字节"EBYTE"

属性 0x0005 (产品型号) 有效, 数据类型 string, 值为 11 字节"FW7421-0-10"

属性 0x0006 (编译日期) 有效, 数据类型 string, 值为 8 字节"20220916"

属性 0x0007 (电源方式) 有效, 数据类型 enum8, 值为 0x00

☆. 使用读属性命令可以读取入网设备的身份信息, 还可以读取当前波特率, PWM 开关状态, PWM 亮度 (占空比) 等设备状态参数。

第三步: 接收节点的数据透传

收到 ZCL 异步命令 "控制命令 (类型 0x82, 码 0x0F) ":

```
55 19 82 0F 00 C5 B9 01 0E 01 08 FC 00 20 DB 00 31 32 33 34 35 36 37 38 39 30 F1
```

解析 ZCL 异步命令:

接收端口: 0 - 端口 1

接收模式: 0 - 点播模式

源短地址: 0xB9C5

源端口: 0x01

帧序号: 0x0E

命令方向: 0x01 - Server To Client

簇 ID: 0xFC08

厂商码: 0x2000

信号强度: 0xDB = -37dbm

命令 ID: 0x00 - 透传接收数据

命令参数: 31 32 33 34 35 36 37 38 39 30 - 入网节点透传发送的 ASCII 码字符串"1234567890"

第四步: 向入网节点发送串口透传数据

发送 ZCL 命令 "控制命令 (类型 0x02, 码 0x0F) ":

```
55 19 02 0F 00 C5 B9 01 8C 00 08 FC 00 20 00 00 31 32 33 34 35 36 37 38 39 30 29
```

解析发送命令:

发送端口: 0 - 默认端口 (端口 1)

发送模式: 0 - 普通发送, 不加密

目标短地址: 0xB9C5

目标端口: 0x01

帧序号: 0x8C

命令方向: 0x00 - Client To Server

簇 ID: 0xFC08

厂商码: 0x2000

应答模式: 0x00 -使用 Default Response, 不开启 APS Ack

命令 ID: 0x00 - 透传发送

命令参数: 31 32 33 34 35 36 37 38 39 30 - 入网节点打印输出 ASCII 码字符串"1234567890"

收到反馈命令:

55 05 02 0F 00 8C 81

收到 ZCL 命令的发送确认 (类型 0x8F, 码 0x02) :

55 0A 8F 02 00 C5 B9 01 8C 00 00 7C

收到 ZCL 异步命令 "默认返回帧 (类型 0x82, 码 0x0B) "

55 10 82 0B 00 C5 B9 01 8C 01 08 FC 00 20 D5 00 00 78

解析 ZCL 异步命令

接收端口: 0 - 端口 1

接收模式: 0 -点播模式

源短地址: 0xB9C5

源端口: 0x01

帧序号: 0x8C

命令方向: 0x01 - Server To Client

簇 ID: 0xFC08

厂商码: 0x2000

信号强度: 0xD5 = -43dbm

命令 ID: 0x00

ZCL 状态: 0x00 - 操作成功

第五步: 控制模组上的 PWM 通断 (控制对方的端口 2)

发送 ZCL 命令 "控制命令 (类型 0x02, 码 0x0F) " :

55 0F 02 0F 00 C5 B9 02 45 00 06 00 00 00 00 02 32

解析控制命令:

发送端口: 0 - 默认端口 (端口 1)

发送模式: 0 - 普通发送, 不加密

目标短地址: 0xB9C5

目标端口: 0x02

帧序号: 0x45

命令方向: 0x00 - Client To Server

簇 ID: 0x0006

厂商码: 0x0000

应答模式: 0x00 -使用 Default Response, 不开启 APS Ack

命令 ID: 0x02-开关切换

命令参数: 无

收到反馈命令:

55 05 02 0F 00 45 48

收到 ZCL 命令的发送确认 (类型 0x8F, 码 0x02) :

```
55 0A 8F 02 00 C5 B9 02 45 00 00 B6
```

收到 ZCL 异步命令“默认返回帧（类型 0x82，码 0x0B）”

```
55 10 82 0B 00 C5 B9 02 45 01 06 00 00 00 DA 02 00 6D
```

☆. 观察目标节点的 PWM 输出熄灭

收到 ZCL 异步命令“属性主动上报（类型 0x82，码 0x0A）”：

```
55 13 82 0A 00 C5 B9 02 0D 01 06 00 00 00 DA 01 00 00 10 00 37
```

```
55 13 82 0A 00 C5 B9 02 0E 01 08 00 00 00 DA 01 00 00 20 FF F5
```

☆. 对目标节点的 PWM 输出口进行开关灯操作，该 PWM 对应的属性也会上报到协调。

4.2 示例 2: 终端节点 AT 命令配置和数据透传

示例简介：

E180ZG120 模组配置成休眠终端节点，使用 AT 命令入网，并完成数据透传的 Demo。

4.2.1 模组配置成 AT 命令模式入网

第一步：配置 AT 命令模式

发送本地配置命令“进入 AT 命令模式（类型 0x00，码 0x16）”

```
55 03 00 16 16
```

收到反馈命令：

```
55 04 00 16 00 16
```

第二步：查询模组当前状态

输入 AT 命令“AT+INFO?”

模组输出：

```
"NO NET\r\n"
```

```
"TYPE=Coordinate\r\n "
```

```
"MAC=0x50325FFFCECA5EC1\r\n "
```

模组未入网

将模组设置为休眠终端

输入 AT 命令“AT+DEVTYPE?”

模组输出：

```
"DEVTYPE=COORDINATOR\r\n"
```

输入 AT 命令“AT+DEVTYPE=3”

模组输出：

```
"OK\r\n"
```

输入 AT 命令“AT+RESET”

模组输出：

```
"OK\r\n"
```

等待 1 秒

模组输出：

```
"BOOT=0\r\n"
```

```
"VERSION=0\r\n"
```

```
"AT_MODE\r\n"
```

第三步：模组入网

☆. 协调器打开允许入网时操作

输入 AT 命令 "AT+JOIN", 休眠终端输入 "\0\0\0\0\0\0 AT+JOIN"

模组输出:

```
"OK\r\n"
```

等待数秒钟

模组输出:

```
" NET:JOIN\r\n"
```

☆. 模组入网成功

输入 AT 命令 "AT+INFO?", 休眠终端输入 "\0\0\0\0\0\0 AT+INFO?"

模组输出:

```
" TYPE=SleepyEndDevice\r\n"
```

```
MAC=0x50325FFFFECA5EC1\r\n"
```

```
PANID=0xB3BB\r\n"
```

```
CHANNEL=20\r\n"
```

```
ADDR=0x3F48\r\n"
```

4.2.2 数据透传

第一步：设置透传目标

☆. 将协调器的串口端口设置为透传目标

输入 AT 命令 "AT+DSTADDR=0000", 休眠终端输入 "\0\0\0\0\0\0 AT+DSTADDR=0000"

模组输出:

```
"OK\r\n"
```

输入 AT 命令 "AT+DSTEP=1", 休眠终端输入 "\0\0\0\0\0\0 AT+DSTEP=1"

模组输出:

```
"OK\r\n"
```

第二步：切换至透传模式

方法 1: 输入 AT 命令 "AT+SEND", 休眠终端输入 "\0\0\0\0\0\0 AT+SEND "

模组输出:

```
"SEND_MODE\r\n"
```

方法 2: 输入 AT 命令 "AT+RESET", 休眠终端输入 "\0\0\0\0\0\0 AT+RESET ", 或复位模组

模组输出:

```
"OK\r\n"
```

等待 1~2 秒

模组输出:

```
"BOOT=0\r\n"
```

```
VERSION=0\r\n"
```

```
SEND_MODE\r\n"
```

等待 1~2 秒, 休眠终端连上父节点

模组输出: (注意结尾没有回车)

```
"NET"
```

第三步: 数据透传发送

输入透传数据 "HelloWorld", 休眠终端输入 "\0\0\0\0\0HelloWorld "

模组输出: (注意结尾没有回车)

```
"OK"
```

协调器端收到 ZCL 异步命令:

```
55 19 82 0F 00 48 3F 01 0B 01 08 FC 00 20 DB 00 48 65 6C 6C 6F 57 6F 72 6C 64 FE
```

第四步: 数据透传接收

☆. 协调器参照 "[向入网节点发送串口透传数据](#)" 向透传模组发送数据的操作

协调器发送 ZCL 命令:

```
55 19 02 0F 00 48 3F 01 8C 00 08 FC 00 20 00 00 31 32 33 34 35 36 37 38 39 30 22
```

透传模组收到:

模组输出:

```
" 1234567890"
```

第五步: 自动绑定透传目标

☆. 加入另一个节点 (路由节点)

输入 AT 命令 "At+*FIND*", 休眠节点输入 "\0\0\0\0\0AT+*FIND*", 路由比休眠节点提前 3 秒输入命令

模组输出:

```
" OK\r\n"
```

等待几秒钟

模组输出:

```
"FIND:ADDR=0x7436 EP=1 cluster=0xFC08\r\n"
```

```
FIND:ADDR=0x7436 EP=1 cluster=0xFC08\r\n"
```

```
RSP:0x7436-1 IDENT:SUCCESS\r\n"
```

```
RSP:0x7436-1 IDENT:SUCCESS\r\n"
```

☆. 出现以上打印信息, 绑定成功。

☆. 在路由节点一端的操作

输入 AT 命令 "At+*FIND*"

路由节点输出:

```
" OK\r\n"
```

等待约 1~3 秒

路由节点输出:

```
"FIND:MISS\r\n"
```

☆. 路由节点是被找的一方, 打印 "FIND:MISS" 后进入 Identify 状态等待其它节点绑定

4.2.3 AT 命令控灯

第一步: 设置绑定控灯

☆. 先在协调器上给休眠终端发送绑定指令, 绑定受控节点 (路由节点) 的 PWM 端口
协调器发送网络管理命令 “设置节点常连接绑定 (类型 0x01, 码 0x21)”, 休眠节点插入 6 个 0x00:

```
55 19 01 21 48 3F 01 C1 5E CA FE FF 5F 32 50 06 00 02 CB 5E CA FE FF 5F 32 50
```

解析发送命令:

目标短地址: 0x3F48

源虚拟 SN: 01 C1 5E CA FE FF 5F 32 50, 透传模组的端口 1 的虚拟 SN

簇 ID: 0x0006, 开关控制簇

目标虚拟 SN: 02 CB 5E CA FE FF 5F 32 50, 路由节点的端口 2 控制 PWM

收到反馈:

```
55 05 01 21 00 04 24
```

收到异步命令绑定响应 (类型 0x81, 码 0x21)

```
55 07 81 21 48 3F 04 00 D3
```

收到发送确认

```
55 07 8F 01 48 3F 04 00 FD
```

☆. 按照以上方式, 控制协调器将休眠终端的透传端口 (端口 1), 与路由节点的端口 2、端口 3、端口 4 分别绑定簇 ID=0x0006 和簇 ID=0x0008。

☆. 然后再测试休眠终端是否绑定路由节点的 PWM 输出。

输入 AT 命令 “AT+IDENTIFY?”

模组输出:

```
"(0) TRANS:SN=[01. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
(1) ONOFF:SN=[02. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
(2) LEVEL:SN=[02. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
(3) ONOFF:SN=[03. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
(4) LEVEL:SN=[03. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
(5) ONOFF:SN=[04. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
(6) LEVEL:SN=[04. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
OK\r\n"
```

☆. 模组打印了一个可控设备的菜单

输入 AT 命令 “AT+IDENTIFY=1”

模组输出:

```
"OK\r\n"
```

等待 10~100 毫秒

模组输出:

```
"RSP:0x7436-2 IDENT:SUCCESS\r\n"
```

☆. 观察现象: 路由节点的端口 2 对应的 PWM 输出 LED 以 1 秒频率闪烁

第二步: AT 命令控制灯的亮灭

输入 AT 命令 “AT+TURNOFF?”, 休眠节点输入 “\0\0\0\0\0\0AT+TURNOFF?”

模组输出:

```
"(0) ONOFF:SN=[02. CB. 5E. CA. FE. FF. 5F. 32. 50]\r\n
(1) ONOFF:SN=[03. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
(2) ONOFF:SN=[04. CB. 5E. CA. FE. FF. 5F. 32. 50] \r\n
OK\r\n"
```

输入 AT 命令 “AT+TURNOFF=0”

模组输出:

```
"OK\r\n"
```

等待 10~100 毫秒

模组输出:

```
"RSP:0x7436-2 ONOFF:SUCCESS\r\n"
```

☆. 观察现象: 路由节点端口 2 对应 PWM 输出 LED 熄灭了

输入 AT 命令 "AT+TURNOFF", 休眠节点输入 "\0\0\0\0\0\0AT+TURNOFF"

模组输出:

```
OK
```

等待 10~100 毫秒

模组每隔 10~100 毫秒陆续输出:

```
"RSP:0x7436-2 ONOFF:SUCCESS\r\n"
```

```
"RSP:0x7436-3 ONOFF:SUCCESS\r\n "
```

```
"RSP:0x7436-4 ONOFF:SUCCESS\r\n "
```

☆. 观察现象: 路由节点的全部 PWM 输出 LED 熄灭了

第三步: AT 命令控制灯的亮度

输入 AT 命令 "AT+LEVELSET?", 休眠终端输入 "\0\0\0\0\0\0AT+LEVELSET?"

模组输出:

```
"(0)LEVEL:SN=[02.CB.5E.CA.FE.FF.5F.32.50] \r\n"
```

```
"(1)LEVEL:SN=[03.CB.5E.CA.FE.FF.5F.32.50] \r\n"
```

```
"(2)LEVEL:SN=[04.CB.5E.CA.FE.FF.5F.32.50] \r\n"
```

```
OK\r\n"
```

输入 AT 命令 "AT+LEVELSET=1,2", 休眠终端输入 "\0\0\0\0\0\0AT+LEVELSET=1,2"

模组输出:

```
"OK\r\n"
```

等待 10~100 毫秒

模组输出:

```
"RSP:0x7436-4 LEVEL:SUCCESS\r\n"
```

☆. 观察现象: 路由节点的端口 2 对应的 PWM 输出 LED 亮度变得很暗

第四步: 设置带双向绑定的控灯

☆. 该功能需要先让受控的灯进入 Identify 模式, 然后发起控制的模组使用 AT+FINDD 功能绑定目标灯。设置成功后使用 AT 命令控制灯 (PWM) 时可以收到受控端 PWM 当前状态的上报消息。

路由节点发送 AT 命令 "AT+PWMIDENT=0,30"

路由节点模组输出:

```
"OK\r\n"
```

☆. 观察路由节点的 PWM0 引脚输出以 1 秒为周期变化的 PWM 信号。

休眠终端发送 AT 命令 "\0\0\0\0\0\0AT+FINDD"

休眠终端模组输出:

```
"OK\r\n"
```

等待 100 毫秒~1 秒

休眠终端模组输出:

```
FIND:ADDR=0x7436 EP=2 cluster=0x0006
```

```
FIND:ADDR=0x7436 EP=2 cluster=0x0008
```

☆. 该状态表示休眠终端和受控灯实现了双向绑定, 受控灯可以向休眠终端上报簇 ID=0x0006 下的属性"0n0ff (开关状态)"和簇 ID=0x0008 下的属性"currentLevel (当前亮度)".

休眠终端每 5 分钟收到以下消息

模组输出:

```
" RPT:0x7436-2 ONOFF=1\r\n"
```

```
" RPT:0x7436-2 LEVEL=255\r\n"
```

☆. 休眠终端收到路由节点上报的消息, 第一可作为受控端的状态心跳包, 第二可以监控受控端受控是否有效。

休眠终端发送 AT 命令" |0|0|0|0|0|0AT+LEVELSET=0,0", 该命令将端口 2 PWM 设置成 0%输出

模组输出:

```
"OK\r\n"
```

等待 10~100 毫秒

模组每隔 10~100 毫秒陆续输出:

```
"RSP:0x7436-2 LEVEL:SUCCESS\r\n "
```

```
"RPT:0x7436-2 LEVEL=125\r\n "
```

```
"RPT:0x7436-2 ONOFF=0\r\n "
```

```
"RPT:0x7436-2 LEVEL=0\r\n "
```

☆. 休眠终端 AT 命令设置路由节点的端口 2 对应 PWM 使其关灯, 端口 2 对应的状态上报到了休眠终端。

4.3 示例 3:上位机软件配置组网和透传

示例简介:

E180ZG120A/B 模组两个, 分别配置成协调器和终端节点, 然后相互数据透传

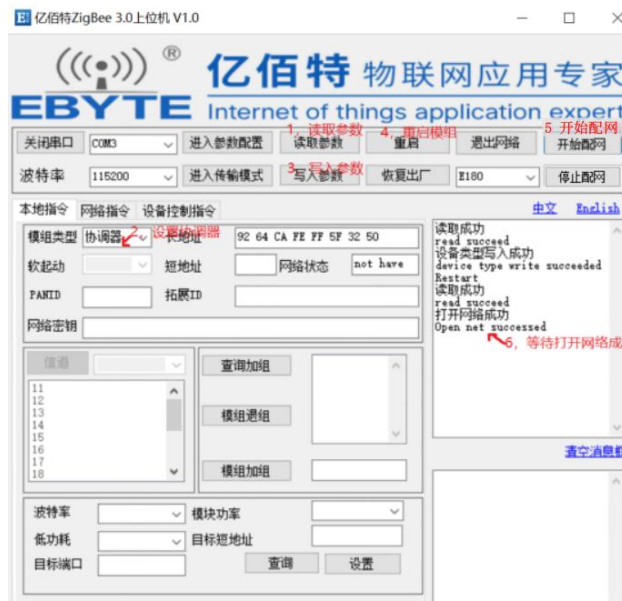
4.3.1 上位机软件配置协调器

第一步: 设置串口和模组类型



- 1, 选择 E180 模组对应的串口
- 2, 选择硬件型号为 E180
- 3, 设置初始波特率为 115200
- 4, 打开串口

第二步, 设置协调器并开始配网



- 1, 读取参数
- 2, 选择协调器模式
- 3, 写入配置参数
- 4, 复位模组
- 5, 点击开始配网
- 6, 等待网络打开成功

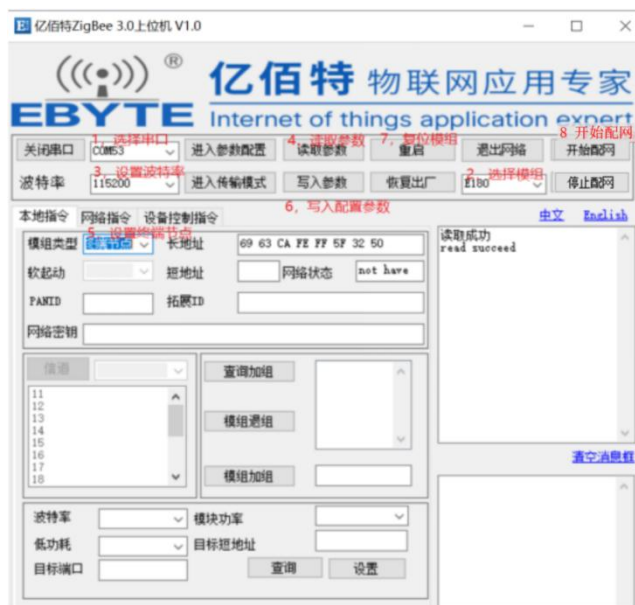
第三步, 协调器启动成功



- 1, 等待打开网络成功后再读取参数
- 2, 获取到模组的短地址为 0x0000, 是协调器
- 3, 获取到 PANID 为 0x577D, 扩展 PANID 为 98 59 D0 A0 40 E2 CF F7, 两个值都是协调器随机创建的
- 4, 获取到当前网络密钥, 该密钥可以在协议分析工具 (如 ubiqua) 中解析数据包
- 5, 协调器创建的信道是 19 信道, 频率为 2445MHz

4.3.2 上位机软件配置终端节点

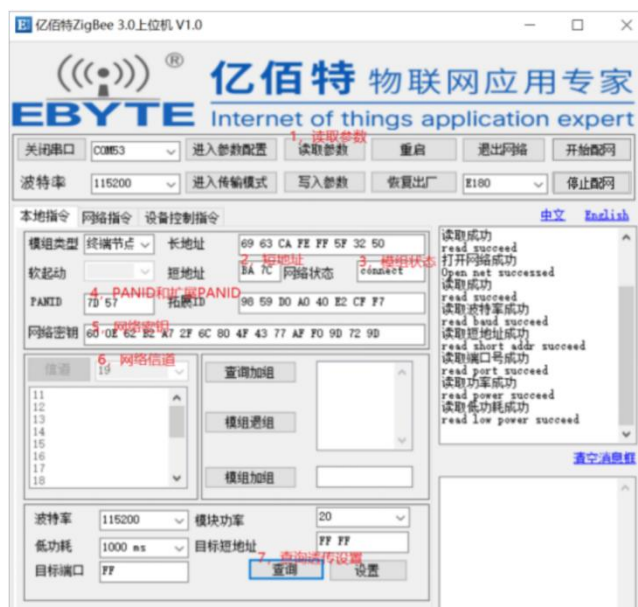
第一步: 设置串口和模组, 并设置入网节点



- 1, 选择模组对应串口
- 2, 设置模组为 E180
- 3, 设置默认波特率 115200

- 4, 读取参数, 未入网的模组只能获取到模组类型 (默认协调器) 和 MAC 地址
- 5, 将模组类型设置成终端节点
- 6, 写入配置参数
- 7, 复位模组
- 8, 执行“开始配网”使终端节点加入协调器, 协调器端必须先进入配网状态。
- 9, 等待模组入网成功

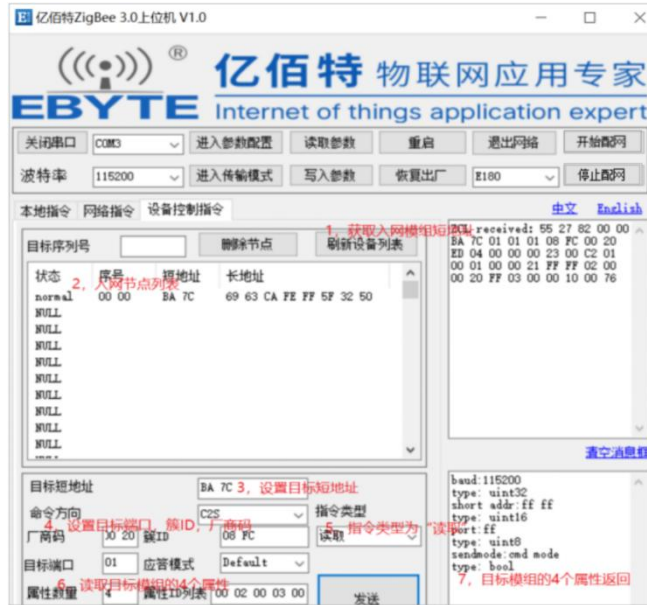
第二步: 终端节点入网成功, 查询模组参数



- 1, 入网成功后读取模组参数
- 2, 模组短地址为 0x7CBA (小端模式)
- 3, 模组状态为 connect, 表示已入网
- 4, PANID 为 0x575D, 和协调器相同。扩展 PANID 也和协调器相同
- 5, 网络密钥和协调器相同
- 6, 网络信道和协调器相同
- 7, 查询透传设置, 模组的波特率为 115200, 功率 20dbm, 低功耗为 1000ms 唤醒一次, 目标短地址和目标端口为广播地址。

4.3.3 协调器和终端节点的数据通信

第一步: 协调器查看入网的终端节点信息



- 1, 获取入网模块的MAC地址和短地址
- 2, 入网节点的MAC地址和短地址展示在协调器中, 注意E180模组不支持断电保存, 每次读取后上位机需要记录下来。
- 3, 设置目标短地址, 当前把入网节点的短地址填进去。
- 4, 目标端口01, 厂商码0x2000, 簇ID=0xFC08, 该格式符合ZCL标准结构, 小端模式格式填入, 注意命令方向选择“C2S”。
- 5, 设置指令类型为“读取”
- 6, 读取目标节点上的4个属性, 0x0000、0x0001、0x0002、0x0003, 以小端模式的数组格式依次填入。然后点击“发送”
- 7, 读取到4个属性, 对应入网节点的“波特率”, “目标地址”, “目标端口”, “命令模式”4个参数, 和入网节点模块的透传设置参数一致。

第二步: 协调器向入网节点传输数据



- 1, 填写目标地址

- 2, 选择命令方向为 C2S
- 3, 厂商码填 0x20000, 簇 ID 0xFC08, 都用小端模式填入
- 4, 指令类型选择“控制”
- 5, 目标端口填 0x01, 应答模式选择“Default”
- 6, 命令 ID 填 0x00
- 7, 输入要发送的数据, 以 16 进制数组格式输入, 然后点击“发送”
- 8, 消息框区域显示 55 10 82 0B 00 BA 7C 01 开头的数据包, 即为目标终端节点返回的消息

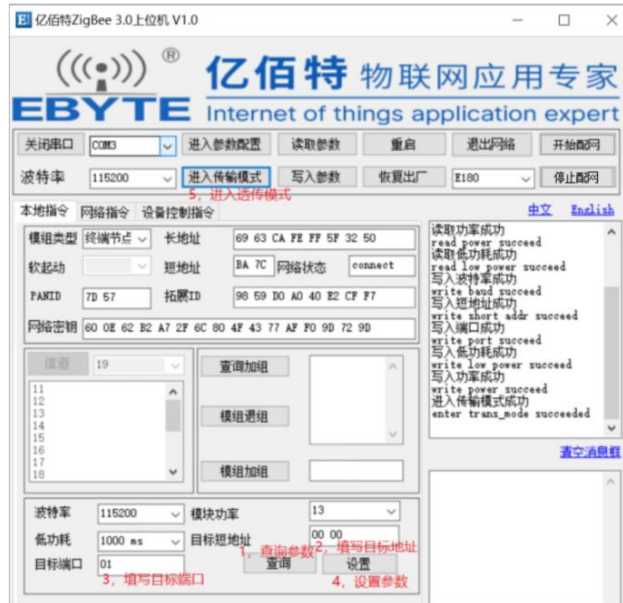
在终端节点的软件窗口中, 收到协调器发送的数据, 消息区输出

ZCL received: 55 17 82 0F 00 00 00 01 0E 00 08 FC 00 20 E8 00 31 32 33 34 35 36 37 38 B6



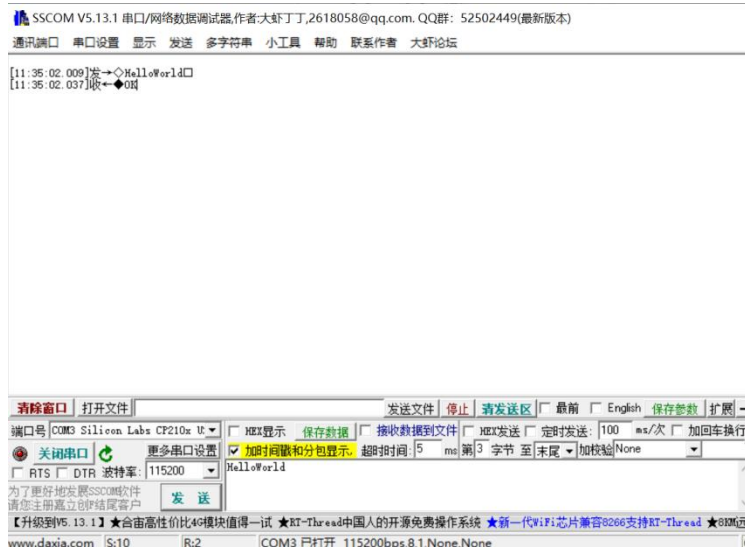
4.3.4 上位机设置数据透传

第一步, 终端节点设置透传模式



- 1, 参数数据当前数据透传相关参数
- 2, 修改透传目标地址为 0x0000
- 3, 修改目标端口为 0x01
- 4, 设置参数
- 5, 进入透传模式

第二步, 上位机软件配置完毕后切换到串口助手, 并硬件复位终端节点



- 1, 在串口助手的输入窗口输入要发送的数据, 并发送
- 2, 收到"OK"

第三步: 协调器接收透传数据



协调器上位机的消息接收窗口，收到消息

ZCL received: 55 19 82 0F 00 BA 7C 01 08 01 08 FC 00 20 EA 00 48 65 6C 6C 6F 57 6F 72 6C 64 7D
 其中 16 进制数据"48 65 6C 6C 6F 57 6F 72 6C 64"为终端节点发送的"HelloWorld"

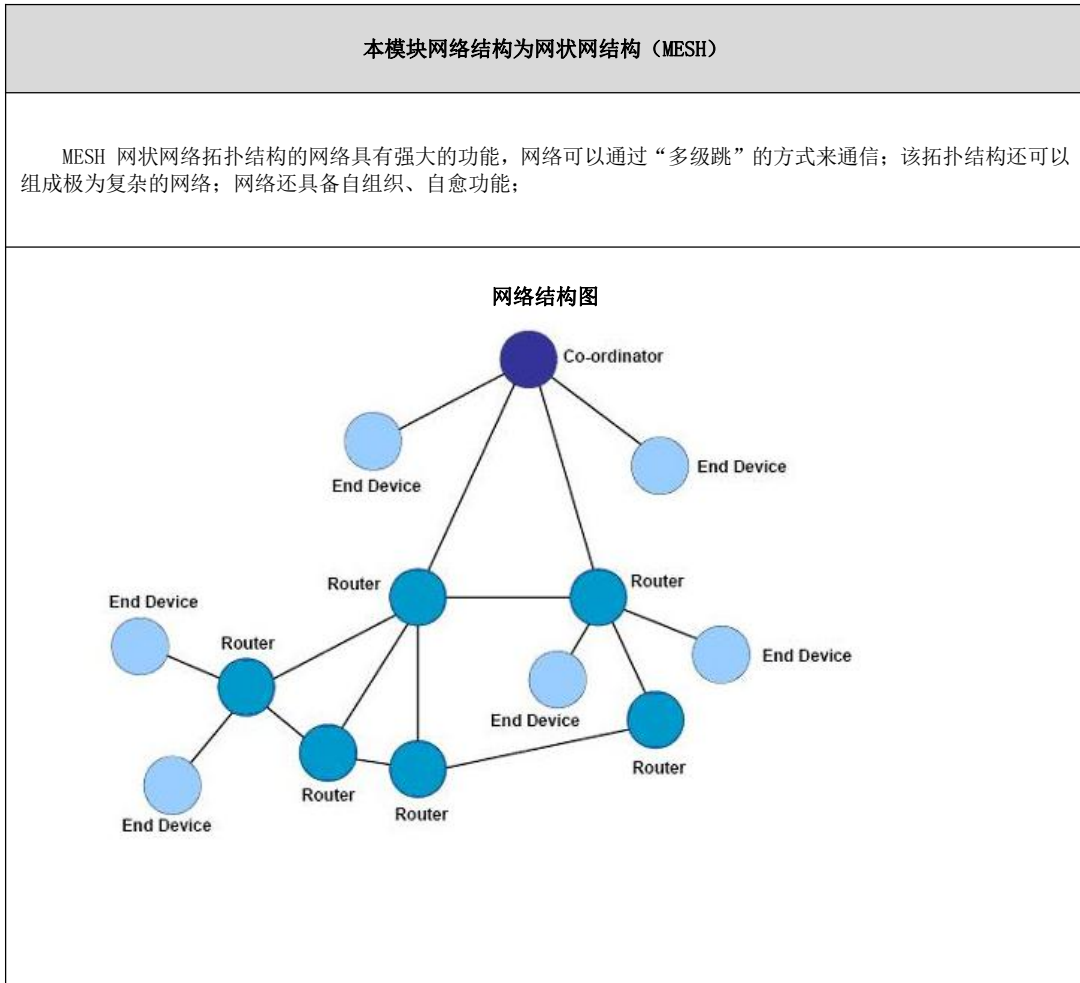
5. 用户须知

5.1 ZigBee 网络角色以及注意事项

序号	描述
1	本模块采用 ZigBee 网络组网，仅实现协调器、路由器功能。
2	Zigbee 网络为 Mesh 结构，不受网络深度影响，路由节点接入越多支持的子节点越多。（终端子节点总数 32，最大组网数 80）
3	协调器可为休眠终端保存数据 7 秒。
4	广播性能：5 秒内最大广播不超过 100 包，实际取决于接入设备的广播接收能力。
5	协调器在网络中是唯一的，短地址固定为 0000。
6	若点播地址为 FFFF, FFFD, FFFC, 则分别对应三种广播模式。若需要组播发送，目标端口设置为 0，目标短地址设置为组 ID。
7	网络参数 PANID 为 FFFF 时为自动生成 PANID，若需要手动设置 PANID 需要保证空间内无该 PANID 的协调器和路由器存在，包括上次加入过本协调器的路由器。
8	网络中所有设备都开启了广播功能，多个设备同时广播或单个设备较高频率的广播都可能导致网络严重堵塞，请尽量避免这种情况。

9	所有的无线命令都会产生发送确认, 发送目标不同发送确认的返回时间也不同, 甚至出现乱序。向某个具体目标发送无线命令后建议等待发送确认再向该目标发送下一条命令。但是向多个不同目标发送命令则不需要等待发送确认即可给下一个目标发送命令。比如在发送目标中有路由节点和休眠节点时, 路由节点的返回比休眠节点快。
10	ZigBee 网络中通信, 单包数据发送周期不能过快 (一般建议在 1 秒以上, 或等待该设备的发送确认或异步返回), 过快可能造成数据的丢失。(特别注意, 网络中节点太多, 广播周期过快可能会造成网络不稳定。)
11	根据节点入网通知和设备信息通知, 判断入网节点是第一次入网还是网络恢复。有第一次入网记录的设备可以算作合法设备。在删除节点时如果节点刚好关机或不在线, 可以认为这个设备已经非法了, 下次再次收到该设备的任何信息 (包括入网通知不是第一次入网) 立即发送删除指令

5.2 网络结构



6. 定制合作

- ★公司客户如需进行产品定制, 请联系我司。
- ★亿佰特已与多家知名企业达成深度合作。



7. 关于我们



亿佰特 (EBYTE) 是一家专业提供无线数传方案及产品的公司

- ◆自主研发数百个型号的产品及软件;
- ◆无线透传、WiFi、蓝牙、Zigbee、PKE、数传电台……等多系列无线产品;
- ◆拥有近百名员工, 数万家客户, 累计销售产品数百万件;
- ◆业务覆盖全球 30 多个国家与地区;
- ◆通过了 ISO 9001 质量管理体系、ISO 14001 环境体系认证;
- ◆拥有多项专利与软件著作权, 通过国际 FCC/CE/ROHS 等权威认证。



最专业的无线应用
微信公众平台
免费样品 技术资讯

【公司电话】028-61399028

【公司传真】028-64146160

【官方网站】www.ebyte.com

【在线商城】cdebyte.taobao.com

【技术支持】support@cdebyte.com

【李经理】raylee@cdebyte.com

【公司地址】四川省 成都市 高新西区 西芯大道 4 号创新中心 B333-D347