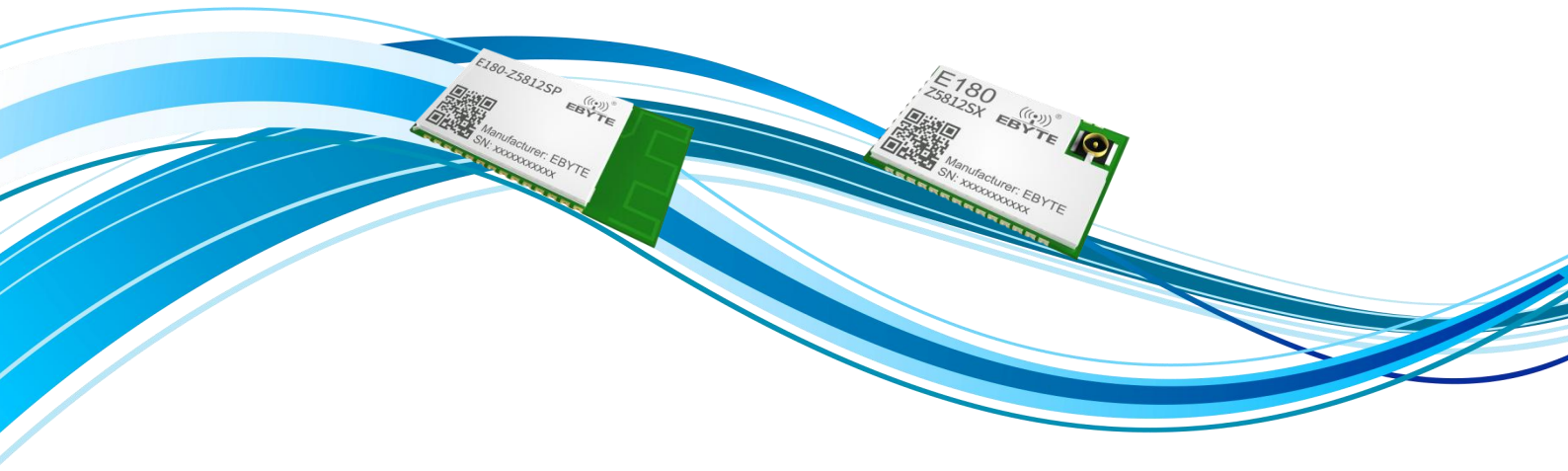
产品规格书

TLSR8258 2.4GHz ZigBee3.0 多功能 SoC 无线模块



成都亿佰特电子科技有限公司

Chengdu Ebyte Electronic Technology Co.,Ltd.

免责申明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。 文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反 言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中所得测试数据均为亿佰特实验室测试所得，实际结果可能略有差异。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

最终解释权归成都亿佰特电子科技有限公司所有。

注 意 ：

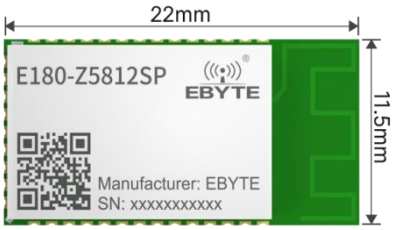
由于产品版本升级或其他原因，本手册内容有可能变更。亿佰特电子科技有限公司保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，成都亿佰特电子科技有限公司尽全力在本手册中提供准确的信息，但是成都亿佰特电子科技有限公司并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

第一章 概述

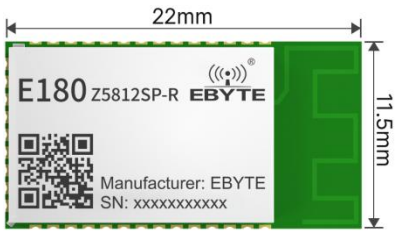
1.1 产品简介

E180-Z5812 系列是成都亿佰特基于 TELINK TLSR8258 无线 SOC 设计生产的一款小体积、低功耗、高可靠性、工作在 2.4GHz 频段的 ZIGBEE 模块，芯片自带高达 48Mhz 的 32 位高性能 MCU，发射功率最高可达到 12dBm,其最低周期休眠电流 2uA。

TLSR8258 是非常有潜力成为未来智能家具、物联网改造、工业自动化首选的无线微控制器，其网络特性符合 ZIGBEE 3.0 标准，并提供一个完整的基于 IEEE802.15.4 标准 ISM 频段的应用集成方案。产品经过系列权威射频仪器的检验和认证，并结合多年的市场经验和该行业用户的实际需求，将无线产品极复杂的通讯协议集成到内置的 SoC 中，支持串口透明传输模式，并集成快捷易用的自组网功能，提供多路可配置的 ADC、IO、PWM 接口，化繁为简，大幅简化无线产品复杂的开发过程，使您的产品以更低的成本快速投入市场。

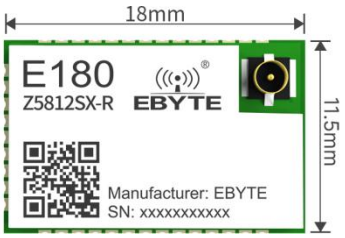


E180-Z5812SP(终端)



E180-Z5812SP-R（路由器）

E180-Z5812SX(终端)



E180-Z5812SX-R(路由器)

1.2 功能特点

- 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；
- 支持 ZCL 标准协议及可配置 ZCL 协议规范，可模拟 zigbee 协议下上百个门类上万种智能设备的接入和控制。可接入多家智能家居网关和平台，无需专门订制固件。
- 可支持多家的智能家居终端产品的接入和控制（限协调器版），支持对 zigbee 协议下上百个门类上万种智能设备的接入和控制，无需专门订制固件。
- 大容量：512K 容量的 flash，64K 容量的 RAM，网络节点可以扩展到 100 以上；
- 角色切换：用户可通过串口指令让设备在终端和休眠终端的两种种类型中任意切换（限终端节点版）；
- 支持多种网络拓扑：点对点，星型网，MESH 网；
- 网络自愈：网络中间节点丢失，其他网络自动加入或保持原网络；
- 地址搜索：用户可根据已加入网络节点的 MAC 地址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。

- 组网管理：协调器版可管理所有组网节点（路由节点和终端节点）MAC 地址，自由添加和删除节点。
- 数据安全： 集成 ZIGBEE 3.0 安全通讯标准，网络含有多级安全密匙；
- 串口配置： 模块内置串口指令，用户可通过出串口指令配置（查看）模块的参数及功能。
- 网络 PAN_ID 更改： 网络 PAN_ID 的任意切换，用户可自定义 PAN_ID 加入相应网络或者将自动选择 PAN_ID 加入网络；
- PWM 控制： 本地/远程的 PWM 控制， 4 路 PWM 通道供用户选择；
- ADC 控制： 本地/远程的 ADC 读取，2 路 ADC 通道供用户选择；
- 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 115200。
- 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帧小于等于 10 个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。
- 模块复位：用户可通过串口命令对模块进行复位操作。
- 恢复出厂设置：用户可通过串口命令对模块进行出厂设置的恢复
- 远程管理和远程配置：用户可使用 ZCL 指令远程管理和远程配置网络中的其他设备

1.3 设备类型介绍

在 ZigBee 网络中存在四种逻辑设备类型：Coordinator(协调器)，Router(路由器)，End-Device(非休眠终端)和 Sleep- End-Device(休眠终端)。ZigBee 网络由一个 Coordinator 以及多个 Router 和多个 End_Device 组成（其终端节点可分为休眠终端和非休眠终端）。本产品有三种形态，分别是专用协调器，专用路由器，专用终端节点。其中专用终端节点支持 End-Device(非休眠终端)和 Sleep-End-Device(休眠终端)两种设备类型。

注：E180-Z5812 系列仅支持路由器和终端，E180-Z5812SP、E180-Z5812SX 为终端模块，E180-Z5812SP-R、E180-Z5812SX-R 为路由器模块。

1.3.1 协调器

具备建立和管理网络的作用，控制着是否允许其它节点加入网络中，存储网络信息， 并具备路由设备的所有功能，其主要任务为管理网络，记录子节点信息， 转发报文，同时，协调器需要对请求入网的终端权限鉴别。

1.3.2 路由器

允许其它节点与路由设备相连，以扩大网络的覆盖范围，其主要任务为转发报文，起到中继路由作用，并具备终端设备的所有功能。如果一个节点通往另一个节点存在多条路径时， 当其中一条路径出现故障，网络会自动调整到其它最优的路径进行传输，以确保数据到达。路由器可以建立自己的网络，也可以加入别人的网，路由器一直处于活动状态，因此它必须使用主电源供电。

1.3.3 非休眠终端

终端设备的主要任务是发送和接收消息，不允许其它节点与终端设备相连。非休眠终端，一直处于工作状态，任意时刻都可以接收和发送数据。

1.3.4 休眠终端

休眠终端，当没有数据收发时，则进入休眠状态，休眠电流低至 2uA 左右。

当需要发送无线数据或进行指令操作时，需先通过串口发送唤醒帧，长度至少 1 个字节（建议用“00”），唤醒时间持续 250ms 时间，期间内可以处理串口数据（HEX 命令、有效负载），当成功接收到一帧串口数据后，设备处理完串口数据然后重新进入休眠。

休眠终端唤醒也可以通过功能引脚 WAKE 唤醒，WAKE 默认为高电平，拉低 WAKE 引脚则模组持续唤醒，释放 WAKE 引脚则恢复默认的高电平模组恢复休眠。

当需要接收数据时，是通过周期性的唤醒来接收数据，唤醒周期设置的越长接收就越延迟。唤醒周期通过端口 01 下 cluster 0xFC08 下的 Attribute 0x0004 来设置 4 个档位（0~3），分别对应 1 秒，3.3 秒，5 秒和 1 分钟。若只需上传数据则可以把唤醒周期设置大于 1 分钟来降低功耗比如电池供电的传感器。

1.4 应用场景

- 智能家居以及工业传感器等；
- 安防系统、定位系统；
- 无线遥控，无人机；
- 无线游戏遥控器；
- 医疗保健产品；
- 无线语音，无线耳机；
- 高级抄表架构(AMI)；
- 汽车行业应用；
- 楼宇自动化解决方案；
- 农业大棚自动化应用；

第二章 规格参数

2.1 极限参数

主要参数	性能		备注
	最小值	最大值	
电源电压（V）	1.9	3.6	超过 3.6V 永久烧毁模块
阻塞功率（dBm）	-	10	近距离使用烧毁概率较小
工作温度（℃）	-40	+85	工业级

2.2 工作参数

主要参数		性能			备注
		最小值	典型值	最大值	
工作电压（V）		1.9	3.3	3.6	≥3.3V 可保证输出功率
通信电平（V）			3.3		使用 5V TTL 有风险烧毁
工作温度（℃）		-40	-	+85	工业级设计
工作频段（MHz）		2405	-	2480	支持 ISM 频段
功耗	发射电流（mA）		24		12dbm 时的最大 24mA
	接收电流（mA）		9		
	休眠电流（μA）		2.5		周期性休眠电流平均为 2.5uA
最大发射功率（dBm）			12		
空中速率（bps）			250k		

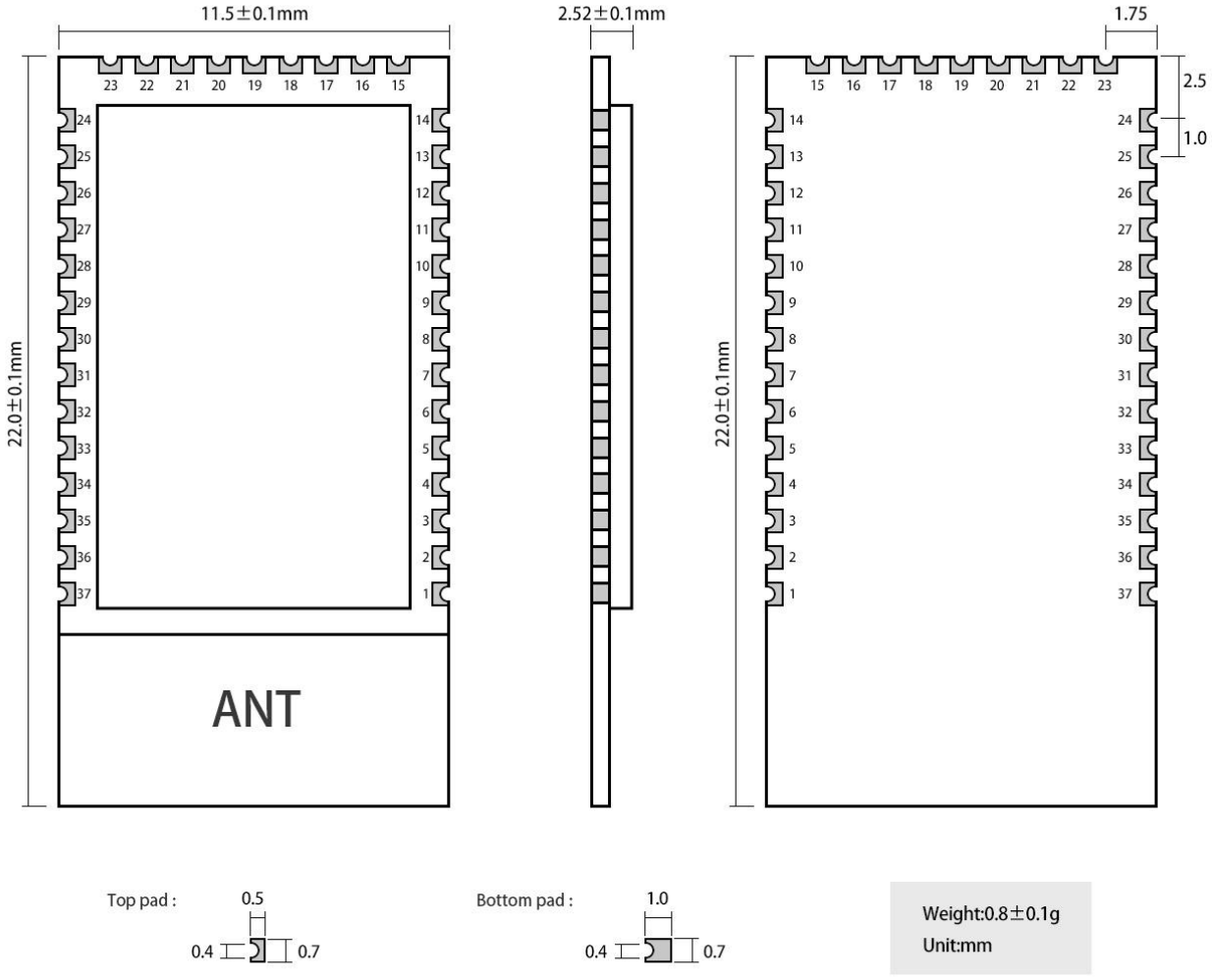
主要参数	描述	备注
参考距离	200m/500m	两点之间（zigbee 网络支持路由多跳功能，可通过增加路由器达到延长传输距离的目的）。
重量	0.9g	
支持协议	Zigbee 3.0	
封装方式	贴片式	
接口方式	1.27mm	邮票孔
IC 全称	TLSR8258F512ET32	
FLASH	512KB	
RAM	64KB	
内核	32 位 MCU	
外形尺寸	11.5*22mm	E180-Z5812SP、E180-Z5812SP-R 尺寸
	11.5*18mm	E180-Z5812SX、E180-Z5812SX-R 尺寸
天线接口	PCB	E180-Z5812SP、E180-Z5812SP-R,等效阻抗约 50Ω
	IPEX	E180-Z5812SX、E180-Z5812SX-R,等效阻抗约 50Ω

2.3 性能参数

主要参数	描述	备注
路由表大小	48	路由器模组
路由器最大邻居个数	26	路由器模组
路由器最大终端连接个数	16	路由器模组
透传最大帧长度	77 字节	广播或组播
	240 字节	单播
波特率（bps）	9600/19200/38400/57600/115200	
PWM 输出路数	4 路	
PWM 精度	1KHz	
PWM 最小渐变周期	10ms	
ADC 采样精度	12bit	
ADC 路数	3	2 路外部，1 路内部电压
最大可创建 ZCL 端口数量	8	路由器
	6	终端
最大可创建 ZCL 簇累计条数	32 条（路由器）	全部端口共享全部簇表
	24 条（终端）	
最大可创建 ZCL 属性累计数量	128 项（路由器）	全部端口共享全部属性空间
	96 项（终端）	
最大 ZCL 属性累计数据空间	1024 Byte（路由器）	全部端口下全部属性指向的数据的累计大小
	768 Byte（终端）	

第三章 机械尺寸与引脚定义

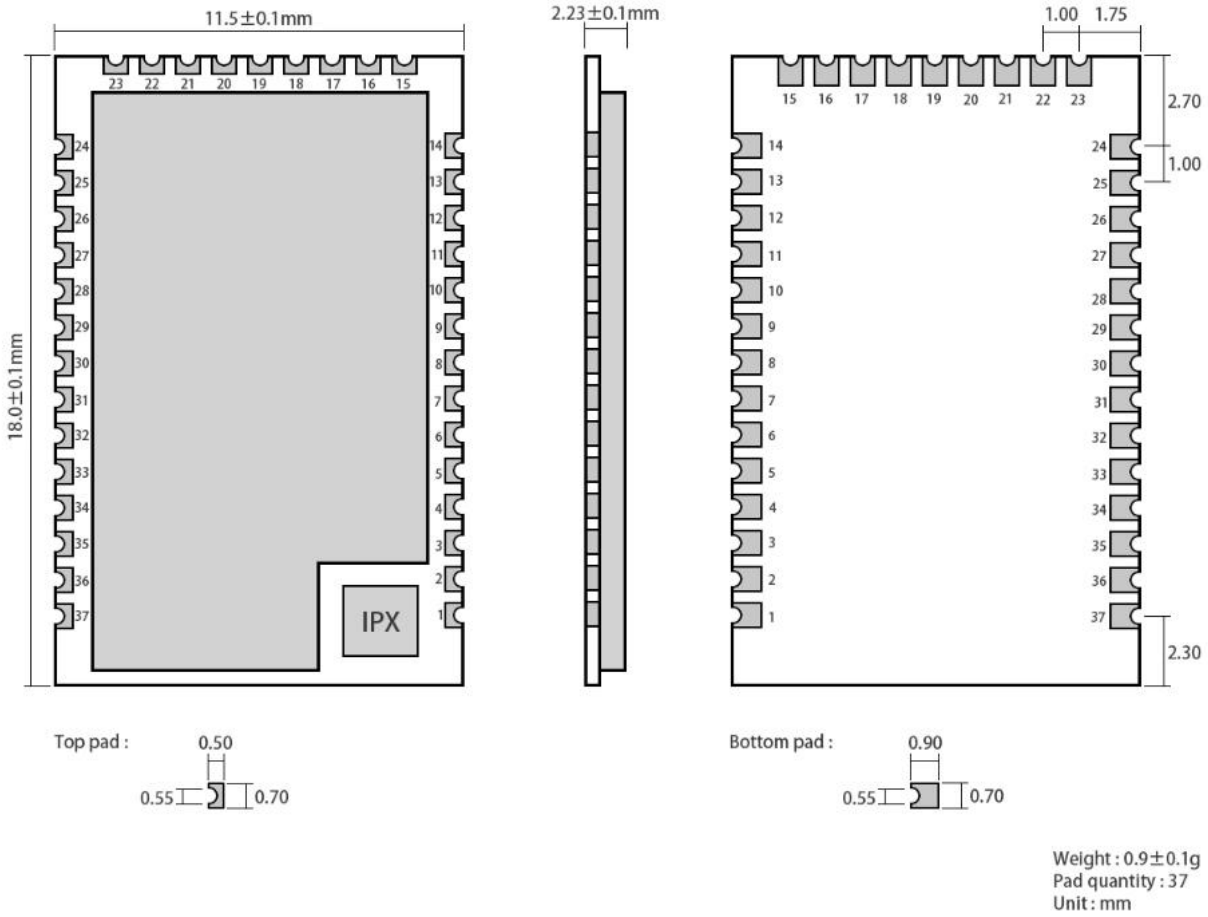
E180-Z5812SP、E180-Z5812SP-R:



引脚序号	引脚名称	引脚方向	引脚用途
1	NC	-	预留，直接悬空
2	GND	-	地线，连接到电源参考地
3	NC	-	预留，直接悬空
4	PD3(WAKE)	输入	WAKE 引脚主要用于对休眠终端唤醒使用，上电时为高电平，当外部将该引脚拉低时，休眠的终端设备将被唤醒
5	PD7(TX)	输出	串口发送端口 TX
6	PA0(RX)	输入	串口接收端口 RX
7	NC	-	预留，直接悬空
8	NC	-	预留，直接悬空
9	PD4(LINK)	输入	网络连接引脚。

10	PA1(BAUD_R)	输入	UART_BAUD_RESET 引脚用于设备波特率复位，上电默认高电平，任何模式下，该引脚拉低 1000ms 以上模块串口参数将恢复默认的 115200
11	PB1(LED1)	输出	预留 LED 指示
12	PC0(GPIO0)	输入/输出	GPIO 输入/输出端口 0
13	VCC	-	模块电源正参考电，电压范围
14	GND	-	地线，连接到电源参考地
15	PB4(GPIO1)	输入/输出	GPIO 输入/输出端口 1
16	NC	-	预留，直接悬空
17	NC	-	预留，直接悬空
18	PB5(AUX)	输出	AUX 引脚指示当前设备工作状态，当引脚为低电平时，指示设备繁忙，高电平指示设备空闲
19	NC	-	预留，直接悬空
20	NC	-	预留，直接悬空
21	PB6(ADC1)	输入	ADC 检测端口 1
22	PB7(ADC2)	输入	ADC 检测端口 2
23	NC	-	预留，直接悬空
24	NC	-	预留，直接悬空
25	SWCLK	输入/输出	串行调试接口，串行线时钟
26	SWDIO	输入/输出	串行调试接口，串行数据输入输出
27	PC2(PWM0)	输出	PWM 输出端口 0
28	PC3(PWM1)	输出	PWM 输出端口 1
29	PC4(PWM2)	输出	PWM 输出端口 2
30	PC1(NET)	输出	NET 引脚指示模块当前网络状态，闪烁表示正在配网或正在绑定
31	NC	-	预留，直接悬空
32	PD2(PWM3)	输出	PWM 输出端口 3
33	NC	-	预留，直接悬空
34	NC	-	预留，直接悬空
35	NC	-	预留，直接悬空
36	GND	输入/输出	地线，连接到电源参考地
37	nRESET	输入	复位引脚

E180-Z5812SX、E180-Z5812SX-R:



引脚序号	引脚名称	引脚方向	引脚用途
1	NC	-	预留，直接悬空
2	GND	-	地线，连接到电源参考地
3	NC	-	预留，直接悬空
4	PD3(WAKE)	输入	WAKE 引脚主要用于对休眠终端唤醒使用，上电时为高电平，当外部将该引脚拉低时，休眠的终端设备将被唤醒
5	PD7(TX)	输出	串口发送端口 TX
6	PA0(RX)	输入	串口接收端口 RX
7	NC	-	预留，直接悬空
8	NC	-	预留，直接悬空
9	PD4(MODE)	输入	工作模式切换引脚，当拉低时间大于 500ms 时工作模式切换。
10	PA1(BAUD_R)	输入	UART_BAUD_RESET 引脚用于设备波特率复位，上电默认高电平，任何模式下，该引脚拉低 1000ms 以上模块串口参数将恢复默认的 115200
11	PB1(ACK)	输出	ACK 引脚用于指示上一次用户数据发送状态，启动发送前该引脚拉低，发送成功后引脚拉高
12	PC0(GPIO0)	输入/输出	GPIO 输入/输出端口 0
13	VCC	-	模块电源正参考电，电压范围
14	GND	-	地线，连接到电源参考地
15	PB4(GPIO1)	输入/输出	GPIO 输入/输出端口 1
16	NC	-	预留，直接悬空

17	NC	-	预留，直接悬空
18	PB5(AUX)	输出	AUX 引脚指示当前设备工作状态，当引脚为低电平时，指示设备繁忙，高电平指示设备空闲
19	NC	-	预留，直接悬空
20	NC	-	预留，直接悬空
21	PB6(ADC1)	输入	ADC 检测端口 1
22	PB7(ADC2)	输入	ADC 检测端口 2
23	NC	-	预留，直接悬空
24	NC	-	预留，直接悬空
25	SWCLK	输入/输出	串行调试接口，串行线时钟
26	SWDIO	输入/输出	串行调试接口，串行数据输入输出
27	PC2(PWM0)	输出	PWM 输出端口 0
28	PC3(PWM2)	输出	PWM 输出端口 2
29	PC4(PWM3)	输出	PWM 输出端口 3
30	PC1(LINK)	输出	LINK 引脚指示模块当前网络状态，输出高电平表已加入网络
31	NC	-	预留，直接悬空
32	PD2(PWM1)	输出	PWM 输出端口 1
33	NC	-	预留，直接悬空
34	NC	-	预留，直接悬空
35	NC	-	预留，直接悬空
36	GND	输入/输出	地线，连接到电源参考地
37	nRESET	输入	复位引脚

第四章 串口输入输出

串口输入输出帧有 4 种操作模式：HEX 指令模式（配置模式），万能模式，透传模式，AT 指令模式。

注意：E180-Z5812SP、E180-Z5812SP-R 与老版本 E180-Z5812SP 通信不兼容，E180-Z5812SX、E180-Z5812SX-R 与老版本 E180-Z5812SX 通信不兼容。新、老版本模组识别方式，模块连接串口调试助手，将模块复位或重新上电，串口调试助手打印类似于 55 0D 80 00 00 10 E5 7E 40 E0 C3 38 C1 A4 35 数据的为新版本模组，不打印数据的为老版本模组。

4.1 HEX 指令模式

模组出厂默认为 HEX 指令模式，该模式下只能输入输出 HEX 指令格式的数据帧。

4.1.1 HEX 指令帧格式

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。

4.1.2 HEX 指令分类

HEX 指令根据输入输出方式，可分 3 大类：

- 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。

HEX 指令又可以进一步细分以下 7 种：

输入命令（含反馈命令）

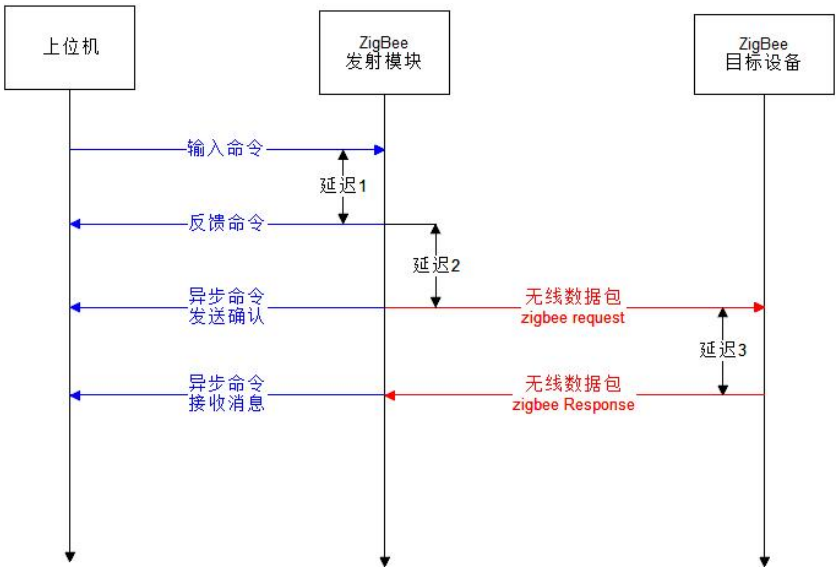
- 本地配置命令：命令类型 0x00，用于模组的本地设置。
- 网络管理命令：命令类型 0x01，用于组网时对其它模组进行网络层的管理。

- 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。

注意事项：“发送确认”命令的（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。在模组上发送确认只有两种状态：0x00 = 发送成功，其它值 = 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。



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。

4.1.3 本产品支持的 HEX 指令

EBYTE全系列Zigbee产品使用统一的HEX指令，但由于芯片方案和软件配置差异，导致实际支持的HEX指令有所差异，本节展示该产品支持HEX指令目录，HEX指令格式与内容参考《亿佰特ZigBee3.0模组HEX命令标准规范》

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（限终端节点）	0x00	0x05
查询与设置设置信道	0x00	0x06
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
读取入网节点地址表	0x00	0x22
重传设备信息通知	0x00	0x28
创建 ZCL 端口	0x00	0x40
添加属性	0x00	0x41
保存端口和属性	0x00	0x42
读写属性	0x00	0x43
清空端口和属性	0x00	0x44
设置当前场景数据	0x00	0x45
查询和设置属性自动上报规律	0x00	0x46
查看已创建端口信息	0x00	0x47
查看已添加属性列表	0x00	0x48
编辑修改自动绑定	0x00	0x4C
查看模组自己的绑定对象	0x00	0x4D
删除绑定对象	0x00	0x4E
发送 ZCL 控制命令到全部绑定目标	0x00	0x4F
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
设备信息通知	0x80	0x05
节点离网通知	0x80	0x06
属性远程修改通知	0x80	0x40
Identify 通知	0x80	0x41
场景执行通知	0x80	0x43
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

- E180-Z5812支持ADC，使用“读取本地属性”命令，对属性0x0100和0x0101进行读取，这两个属性分别为ADC1和ADC2的ADC值，也可以通过ZCL命令远程访问ADC，读取簇0xFC08下的属性0x0100和0x0101（需使用厂商码0x2000）。

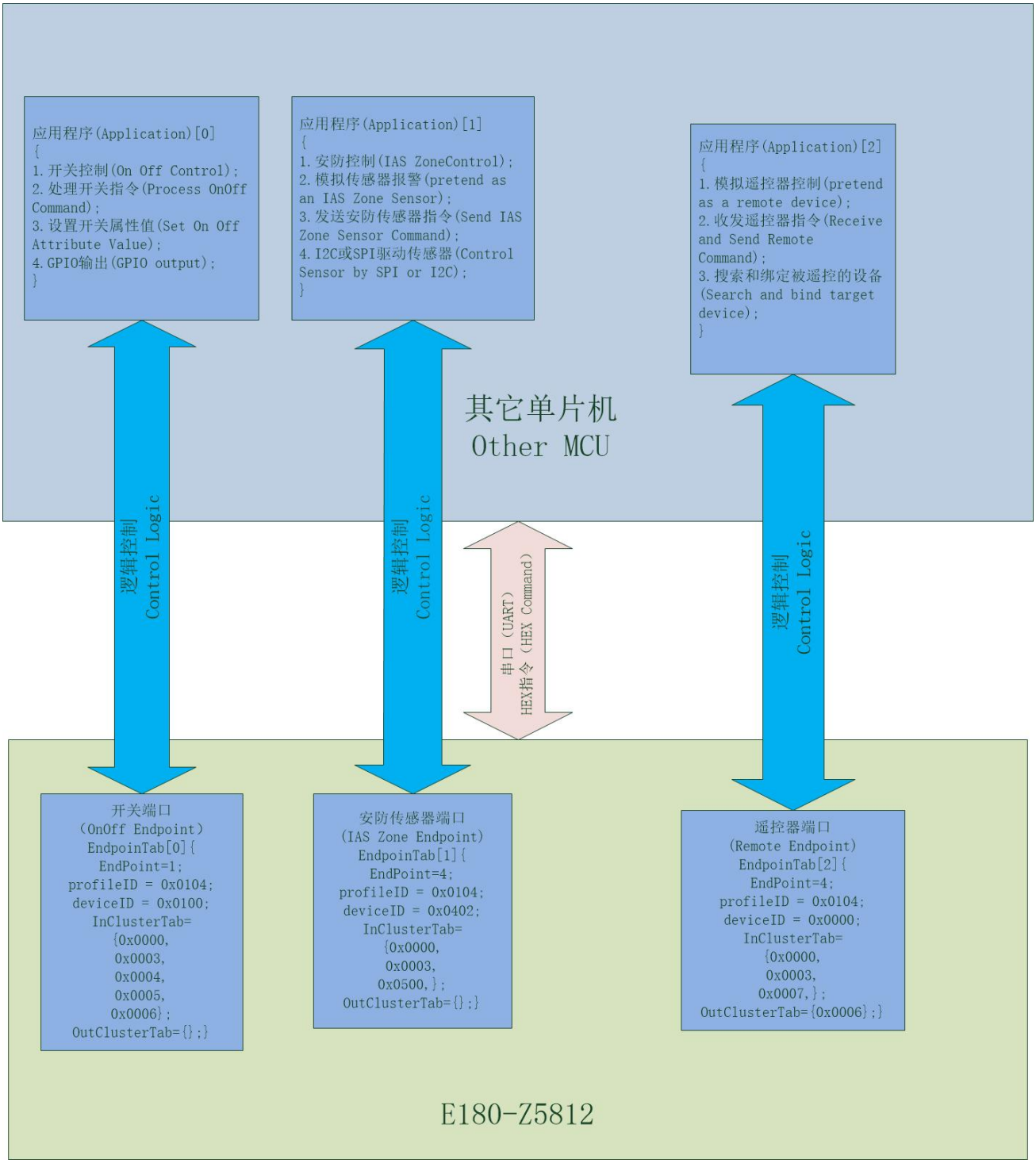
4.2 万能模式

万能模式是一个 HEX 指令模式下的新功能。该模式下可把模组按照 ZCL 标准规范进行配置，模拟各种符合 ZCL 规范的智能设备。万能模式下配置后的模组接入第三方智能网关（如某鸦 zigbee 网关）可被正确识别成对应的接入设备，并收发网关与该设备交互的控制指令。

4.2.1 万能模式的注意事项

- 万能模式必须在未组网时对节点进行设备配置，才能模拟各种zigbee智能设备的接入和控制指令的收发。
- 使用万能模式时，必须先使用“创建ZCL端口”命令创建设备端口，然后使用“添加属性”添加这个端口下的所有属性，然后使用“保存端口和属性”保存这个端口种创建的属性。
- 添加属性时，必须满足先属性ID从小到大，再簇ID从小到大的添加顺序。一个簇下的属性全部加满后再增加下一个簇的属性。
- 创建端口时须填入该端口下所有簇的累计属性，实际添加属性大于该数值，会导致添加失败。
- 添加属性时，实际加入的属性的簇可以比创建端口时计划添加的簇多，这些簇在设备接入网关后不会被网关发现，但是网关如果强制读这些簇下面的属性则可以读到有效数值。（也可以叫做隐藏簇）
- 协调器和路由节点模式下，累计添加端口不超过8个，8个端口累计添加属性不超过32个簇，累计属性不超过128项。
- 终端节点和休眠节点模式下，累计添加端口不超过6个，6个端口累计添加属性不超过24个簇，累计属性不超过96项。
- 需要修改ZCL属性配置信息，或回退到透传模组模式，直接使用“清空端口和属性”，该命令必须在退出网络或者没有组网的状态下才能生效。

4.2.2 万能模式示意框图



- 外接MCU与E180-Z5812模组交互完全通过HEX指令，外接MCU可以通过不同的HEX模拟不同的设备接入与控制。
- 该框图只是一个软件架构示意图，创建了3个端口用于表示三种不同设备，实际可根据外接MCU的处理能力，创建数量更少的端口，或创建多个功能相同的端口，以减轻外接MCU的处理压力。
- 端口创建和属性值的初始化信息保存在E180-Z5812的FLASH中，因此可以使用PC机串口指令在zigbee模组上创建端口和初始化属性值，然后模组硬件接驳MCU，MCU应用程序只处理应用层的指令交互和设备控制，无需再设计创建端口和初始化属性。

4.2.3 万能模式使用教程

注：万能模式使用教程见第六章节

4.3 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。

4.3.1 数据透传的 ZCL 规范

数据透传符合 ZCL 规范，其规范如下。任何 Zigbee 设备只要按照以下规范设置，都可以收发 E180-Z5812 串口输入输出的数据。

- 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。

4.3.2 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

4.3.3 绑定透传目标

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（《HEX 指令》）”，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+FIND”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 PD4 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

"BUSY"：透传时前一包数据没传完又继续传后一包数据会造成数据拥堵

4.4 AT 指令模式

AT模式是透传模式下的一种特殊状态，用于透传模式下的模组配置。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。

4.4.1 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”。

4.4.2 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。

4.4.3 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》。

4.4.4 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
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
设置或读取休眠等级	AT+LPLEVEL	Y	Y	Y
设置或读取 ModbusID	AT+MBID	Y	Y	Y
PWM 输出	AT+PWM	N	N	Y
采集 ADC	AT+ADC	N	N	Y
进入测试模式	AT+TEST	N	N	Y

注：上表红色字体为 E180-Z5812 系列模组特殊支持指令。

4.4.5 新增 AT 命令

① 设置或读取 Modbus ID

操作 1，设置 Modbus ID

输入命令："AT+MBID=%d"，%d 为 0~255 的整数

返回命令："OK\r\n"

操作 2，查看 Modbus ID

输入命令："AT+MBID?"，%d 为 0~255 的整数

返回命令："MBID=%d\r\n"

② 设置 PWM 输出

设置 PWM 输出

输入命令："AT+PWM=%d,%d"，第一个%d 是 PWM 通道，取值范围 0~3，第二个%d 为 PWM 脉宽，取值范围 0~1000。

返回命令："PWMCH=%d, PWM=%d\r\n"，第一个%d 是 PWM 通道，第二个%d 为 PWM 脉宽。

③ 设置 ADC 采集

采集 ADC 读数

输入命令："AT+ADC=%d"，%d 为 ADC 通道，取值范围 0~2

返回命令："CH:%d ADC=%d\r\n"，两个%d 分别为 ADC 通道和 ADC 读数

④ 进入测试模式

输入命令："AT+TEST=%d"，%d 为测试信道，11~26 为有效值。

返回命令："OK\r\n"

使用该命令后，模组会连续不断的输出信标请求帧，可以在频谱仪上检查到信号。

4.5 模式切换

4.5.1 模式切换表

		目标模式			
		HEX 命令	万能模式	AT 命令	透传模式
当前模式	HEX命令		配置命令“创建 ZCL 端口”，“添加属性”，“保存端口和属性”（见《HEX 命令》2.1.28 至 2.1.30）	配置命令“设置本地属性（《HEX 命令 2.1.16》）” --- 把属性 0x0003 设置为非 0 值	配置命令“设置本地属性（《HEX 命令 2.1.16》）” --- 把属性 0x0003 设置为非 0 值
	AT 命令	″AT+EXIT″			″AT+SEND″
	透传模式	发送 3 个字符″+++″		发送 3 个字符″+AT″	
	万能模式	退出网络后使用配置命令“清空端口和属性” --- 直接恢复出厂		无	无

4.5.2 模式切换注意事项

- 模组实际只有 HEX 指令模式和透传模式，AT 指令模式可以看做透传模式下的一种特殊状态。
- AT 命令模式下重启模组，模组缺省状态为透传模式。
- HEX 指令模式切换透传模式或 AT 模式，由上一次切换会 HEX 模式前的模式决定。
- 万能模式下只能使用 HEX 命令进行控制，万能模式下也可通过 HEX 指令来模拟数据透传。

第五章 模组配网与数据传输

5.1 模组配网

5.1.1 HEX 指令配网

未配网的模组无法传输数据，配过网的模组永远在配网的网络内，除非进行退网操作。

配网时先让协调器配网。新出厂的协调器没有配网，使用 HEX 指令“开始配网（《HEX 命令》2.1.3）”，协调器会新建一个网络。协调器会自动生成一个有效的 PANID 表示建立网络成功，同时会开辟一个信道运行该网络。协调器重启后继续运行在该网络。

如果需要路由器，终端节点，休眠终端加入该网络，协调器需要再执行一次“开始配网”指令，表示协调器接纳新设备入网。需要入网的路由器，终端节点，休眠终端同时也执行“开始配网”指令。路由器，终端节点，休眠终端配网结束后会，无论配网成功与否都会收到“网络状态变更通知”指令。“网络状态变更通知”指令会告知上位机用户配网是否成功，也可以使用“查询模组当前状态”指令查看模组是否已配网。

5.1.2 AT 指令配网

模组在透传模式下，可以开启 AT 指令进行配网，透传模式下输入"+AT" 3 个字符开启 AT 模式。

协调器在 AT 模式下，输入"AT+JOIN"打开网络。路由器和终端节点输入"AT+JOIN"打开网络。

5.1.3 按键配网

引脚 PD4（网络连接引脚）输入一个持续 10ms~200ms 的低电平信号触发按键功能，路由器或终端节点在未配网的情况下主动连接协调器，协调器在该引脚输入按键信号则进入配网模式。

需要注意已经配网的路由器和终端节点，如果同时多个这类设备同时按下配网按键会触发绑定模式，即互相将对方设置成透传数据的目标设备。

5.2 数据传输

数传模组无论在 HEX 指令模式还是数据透传模式，数据发送方式都只有广播和点播，其中广播模式下包括组播发送，点播模式下包括绑定发送。

5.2.1 广播模式

该模式下，目标短地址为 0xFFFC（只到协调器或路由器），0xFFFD（包括非休眠终端节点）和 0xFFFF（包括休眠终端的全部设备）。目标端口为 0xFF。

5.2.2 点播模式

该模式下可以精准控制数据传输给谁，或者精准控制某个模组的 PWM 输出。该模式下短地址为需要精准控制的模组的短地址，目标端口为需要精准控制的模组的端口号（4 路 PWM 输出端口对应 2,3,4,5）

5.2.3 组播模式

该模式下可以对部分需要被控制的模组进行广播控制。目标地址为需要控制的分组号，目标端口为 0。使用该功能需要提前对目标模组进行分组操作，分组操作需要精确到模组的端口。如果仅需要控制数据透传的输入输出，需将模组的端口 1 加入期望分组。如果需要分组控制 PWM 输出，则需要将模组对应的 PWM 端口加入分组。分组控制 PWM 时，可将多个不同模组的多个不同 PWM 序号加入指定分组，在组播控制它们时可以实现希望的控制效果。

5.2.3 绑定传输模式

绑定传输模式是一种轮询式点播，通过锁定目标设备 MAC 地址来实现寻址，不需要记住对方短地址。E180-Z5812 系列模组只有在两种情况下使用绑定传输。一个节点可以同时绑定多个不同目标地址。

- 绑定透传：透传模式下，目标地址为 0xFFFE，目标端口为 0xFF，透传数据点播传输到绑定表中最新绑定的目标设备。
- 状态上报：E180-Z5812 在设置绑定后，会将串口模式、Modbus ID、PWM 通断状态，PWM 脉宽上报到目标设备，比如协调器会在检测到路由器或终端接入时自动设置路由器或终端绑定协调器自己，路由器或终端会按照固定周期把串口模式、Modbus ID、PWM 通断状态，PWM 脉宽上报至协调器，且如果以上状态发生改变后也会上报至协调器。同时这些状态也可以上报至其它路由器或终端，方便使用路由器或终端控制路由器或终端的应用时获取对方状态。

5.3 设置绑定

5.3.1 协调器远程配置

按照《HEX 指令》手册中“设置节点常连接绑定（3.3.6）”设置绑定源设备和目标设备。注意绑定的对象均为各个节点上的端口（虚拟设备），即每个端口都要端口号+MAC 地址组成的唯一 SN 号。

注：每个节点都有若干个虚拟设备都可以作为绑定对象

使用亿佰特上位机工具，可以方便的设置绑定源和绑定目标。远程配置绑定时，除了可以在协调器上操作，也可以在路

由器或终端节点上操作。另外如果绑定目标 SN 全部为 0，表示绑定目标是自己。该方法可以让路由器或终端节点作为主控时，让其它节点绑定自己，方便获取对方状态。

5.3.2 一键设置绑定

该方式仅限绑定透传目标，无法绑定状态上报。

路由器或终端配网后，在 LINK 按键（PD4）输入一个 50ms~500ms 的低电平信号触发按键。两个路由器或终端模组，先后分别触发 LINK 按键。先触发的模组的 NET 引脚（PC1）输出 1Hz 的高低电平，第二个模组再触发按键。

一键绑定也可以用 HEX 指令“自动绑定目标（2.1.17）”或 AT 命令“AT+FIND”来代替按键触发。

5.3.3 查看绑定

使用协调器可以查看节点的绑定，使用 HEX 命令“查看节点常连接绑定（3.3.8）”查看绑定表。输入节点短地址，就可以看到该节点下所有的虚拟 SN 绑定了哪些其它设备的虚拟 SN。

使用亿佰特上位机软件，也可以查看各个设备绑定的其它设备。

注：上位机软件“查看常连接”可以查看所有绑定

第六章 万能模式使用教程

万能模式下需要自备一个 zigbee 智能网关（可选择某鸦，某士丹利，某微等），使用该网关配套的 APP 添加 E180-Z5812 系列模组。

6.1 例程 1：模拟一个三路开关接入网关并使用 APP 控制开关

6.1.1 创建第一个开关端口并添加属性

第一个端口的索引号为 0（用数组表示端口列表），端口号可从 1 到 240 任意分配，但为了更好的网关和云平台的兼容性，推荐第一个端口号使用 0x01。

输入“创建 ZCL 端口”命令：

```
55 15 00 40 [15] [01] [04 01] [00 01] [05] { [00 00] [03 00] [04 00] [05 00] [06 00] } [00] 51
```

“[ ]”中内容依次为

- 开关累计 0x15=21 个属性
- 创建端口号 = 0x01
- 轮廓 ID = 0x0104
- 设备 ID = 0x0100
- 输入簇数量 = 0x05
- 输入簇{0x0000, 0x0003, 0x0004, 0x0005, 0x0006}
- 输出簇数量 = 0x00

返回命令:

```
55 16 00 40 [00] [00] [01] [04 01] [00 01] [05] {[00 00] [03 00] [04 00] [05 00] [06 00]} [00] 44
```

“[ ]”中内容依次为

- 执行成功
- 模组分配的端口索引 = 0
- 创建端口号 = 0x01
- 轮廓 ID = 0x0104
- 设备 ID = 0x0100
- 输入簇数量 = 0x05
- 输入簇{0x0000, 0x0003, 0x0004, 0x0005, 0x0006}
- 输出簇数量 = 0x00

然后向这个端口添加属性，由于当前创建的端口 0x01 是活跃状态，因此添加的属性都是添加到该端口。如果需要创建其它端口，需要将这个端口的属性添加完毕后再保存。如果此时模组重启，活跃的端口 0x01 会被清除掉，这个时候需要重新创建端口 0x01，或者创建其它的端口。

ZCL 开关类设备，需要添加以下属性：

名称	簇 ID	厂商码	属性 ID	数据类型	支持操作	属性方向
ZCL 版本	0x0000	0x0000	0x0000	0x20 (uint8)	0x01 (只读)	Sever

协议中版本	0x0000	0x0000	0x0001	0x20 (uint8)	0x01 (只读)	Sever
软件版本	0x0000	0x0000	0x0002	0x20 (uint8)	0x01 (只读)	Sever
硬件版本	0x0000	0x0000	0x0003	0x20 (uint8)	0x01 (只读)	Sever
厂商名称	0x0000	0x0000	0x0004	0x42 (string)	0x01 (只读)	Sever
产品型号	0x0000	0x0000	0x0005	0x42 (string)	0x01 (只读)	Sever
生产日期	0x0000	0x0000	0x0006	0x42 (string)	0x01 (只读)	Sever
供电方式	0x0000	0x0000	0x0007	0x30 (enmu8)	0x01 (只读)	Sever
属性终止符	0x0000	0x0000	0xFFFD	0x21 (uint16)	0x01 (只读)	Sever
Identity 标记状态	0x0003	0x0000	0x0000	0x21 (uint16)	0x03 (可读写)	Sever
属性终止符	0x0003	0x0000	0xFFFD	0x21 (uint16)	0x01 (只读)	Sever
支持分组字符串命名	0x0004	0x0000	0x0000	0x18 (bitmap8)	0x01 (只读)	Sever
属性终止符	0x0004	0x0000	0xFFFD	0x21 (uint16)	0x01 (只读)	Sever
累计场景数量	0x0005	0x0000	0x0000	0x20 (uint8)	0x01 (只读)	Sever
当前场景 ID	0x0005	0x0000	0x0001	0x20 (uint8)	0x01 (只读)	Sever
当前场景组 ID	0x0005	0x0000	0x0002	0x21 (uint16)	0x01 (只读)	Sever
当前场景有效状态	0x0005	0x0000	0x0003	0x10 (bool)	0x01 (只读)	Sever
支持场景字符串命名	0x0005	0x0000	0x0004	0x18 (bitmap8)	0x01 (只读)	Sever
属性终止符	0x0005	0x0000	0xFFFD	0x21 (uint16)	0x01 (只读)	Sever
开关状态	0x0006	0x0000	0x0000	0x10 (bool)	0x05 (只读/上报)	Sever
属性终止符	0x0006	0x0000	0xFFFD	0x21 (uint16)	0x01 (只读)	Sever

按照表格顺序，使用“添加属性”命令，依次加入表格中的属性。

添加第一个属性：

```
55 0C 00 41 [00 00] [00 00] [00 00] [20] [01] [00] 60
```

“[ ]”中内容依次为

- 簇 ID = 0x0000
- 厂商码 = 0x0000
- 属性 ID = 0x0000
- 数据类型 = 0x20（数据类型 0x20 = uint8）
- 操作方式 = 0x01，支持只读
- 属性方向 = 0x00，该属性为 sever 方向（受控端方向）

返回命令：

```
55 0E 00 41 [00] [01] [00 00] [00 00] [01] {[00 00] [20] [01]}  60
```

“[ ]”中内容依次为

- 添加成功
- 当前端口=0x01
- 簇 ID = 0x0000
- 厂商码 = 0x0000
- 添加属性计数=1
- “{ }”中为已添加属性的结构体，4 字节对齐
- 属性 ID = 0x0000
- 数据类型 = 0x20

- 操作方式 = 0x01，支持只读

添加属性命令中，簇 ID，厂商码，属性 ID 均按照小端模式输入。例如添加“硬件版本”这个属性，应当输入命令：

```
55 0C 00 41 [00 00] [00 00] [04 00] [42] [01] [00] 06
```

“[ ]”中内容依次为

- 簇 ID = 0x0000
- 厂商码 = 0x0000
- 属性 ID = 0x0004
- 数据类型 = 0x42（数据类型 42 = string）
- 操作方式 = 0x01，支持只读
- 属性方向 = 0x00，该属性为 sever 方向（受控端方向）

由于此时簇 0x0000 已添加了 5 个属性，因此返回命令如下：

```
55 1E 00 41 [00] [01] [00 00] [00 00] [05] {[00 00] [20] [01], [01 00] [20] [01], [02 00] [20] [01], [03 00] [20] [01], [04 00] [42] [01]} 02
```

“[ ]”中内容依次为

- 添加成功
- 当前端口=0x01
- 簇 ID = 0x0000
- 厂商码 = 0x0000
- 添加属性计数=5
- “{ }”中为已添加属性的结构体，4 字节对齐
- 第一个属性”[00 00] [20] [01]”，属性 ID=0x0000，数据类型=0x20，操作方式=0x01
- 第二个属性”[01 00] [20] [01]”，属性 ID=0x0001，数据类型=0x20，操作方式=0x01
- 第三个属性”[02 00] [20] [01]”，属性 ID=0x0002，数据类型=0x20，操作方式=0x01
- 第四个属性”[03 00] [20] [01]”，属性 ID=0x0003，数据类型=0x20，操作方式=0x01
- 第五个属性”[04 00] [42] [01]”，属性 ID=0x0004，数据类型=0x42，操作方式=0x01

按照表中的属性，全部添加完毕后，保存属性。

输入命令：

```
55 03 00 42 42
```

返回命令：

55 04 00 42 [00] 42

“[ ]”中内容依次为

- 保存成功

保存一个端口以及属性成功后，可以继续创建新的端口，并添加相同的属性，由于本例程需要模拟一个 3 路开关，因此需要再继续添加两个相同的端口，为了保证网关的兼容性，推荐使用 0x02 和 0x03 作为后续的两个端口。

6.1.2 创建其它开关端口并添加属性

输入“创建 ZCL 端口”命令创建端口 2：

```
55 15 00 40 [15] [02] [04 01] [00 01] [05] { [00 00] [03 00] [04 00] [05 00] [06 00] } [00] 52
```

“[ ]”中内容依次为

- 开关累计 0x15=21 个属性
- 创建端口号 = 0x02
- 轮廓 ID = 0x0104
- 设备 ID = 0x0100
- 输入簇数量 = 0x05
- 输入簇{0x0000, 0x0003, 0x0004, 0x0005, 0x0006}
- 输出簇数量 = 0x00

返回命令:

```
55 16 00 40 [00] [01] [02] [04 01] [00 01] [05] {[00 00] [03 00] [04 00] [05 00] [06 00]} [00] 44
```

“[ ]”中内容依次为

- 执行成功
- 模组分配的端口索引 = 1
- 创建端口号 = 0x02
- 轮廓 ID = 0x0104
- 设备 ID = 0x0100
- 输入簇数量 = 0x05
- 输入簇{0x0000, 0x0003, 0x0004, 0x0005, 0x0006}
- 输出簇数量 = 0x00

然后继续添加表格中的全部属性，添加完毕后保存属性。然后继续添加端口号 0x03，并为该端口添加属性并保存。

待全部属性添加完毕后，使用复位命令复位模组，模组将引导新建的属性进行启动。

发送复位命令：

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

返回：

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

收到异步命令表示复位成功：

```
55 0D 80 00 00 [10] [E5 C7 F9 3E 7D 38 C1 A4] 55
```

“[]”中内容依次为

模组软件版本=0x10

模组 MAC 地址= E5 C7 F9 3E 7D 38 C1 A4

6.1.3 初始化属性值

初始化属性值可以使模组在接入网关后被正确识别到，E180-Z5812 模组中创建的每个属性都可以支持读写。其中 E180-Z5812 对簇 ID=0x0000 下的 8 个基础属性，以及其余 8 个任意簇下的任意属性设置一个掉电保存值。注意如果设置掉电保存的属性值，多个端口如果都支持该属性，那么全部端口下的这些属性都会在重启后变成掉电保存的值，也就是该属性的默认值。

设置属性“厂商名称”的默认值：

```
55 1C 00 43 [02] [00] [00] [00 00] [00 00] [04 00] [45 42 59 54 45 20 5A 69 67 42 65 65 20 33 2E 30] 31
```

“[ ]”中内容依次为

- 操作方式 = 0x02，写属性且保存 FLASH
- 端口索引 = 0x00，被写入的属性位于创建的第一个端口

- 属性方向 = 0x00，被写入的属性是 Server 端
- 簇 ID=0x0000，被写入的属性位于簇 0x0000
- 厂商码=0x0000，被写入的属性厂商码为 0x0000
- 属性 ID=0x0004，被写入的属性的 ID 是 0x0004
- 写入的属性数据 16 字节，是字符串“EBYTE ZigBee 3.0”的 16 进制格式

返回命令：

```
55 04 00 43 00 43
```

- 写入成功

继续设置属性”产品型号”和“生产日期”，绝大多数网关会验证接入设备是否有该字段，通常输入任意字符串即可。

设置属性“产品型号”，产品型号为“EBYTEOnOffSwitch”

```
55 1C 00 43 02 00 00 00 00 00 00 05 00 45 42 59 54 45 4F 6E 4F 66 66 53 77 69 74 63 68 57
```

设置属性“生产日期”，生产日期为“20231204”

```
55 14 00 43 02 00 00 00 00 00 00 06 00 32 30 32 33 31 32 30 34 43
```

然后复位模组，使 3 个端口均套用相同的属性值。

6.1.4 接入网关

使用网关提供的 APP，添加设备。通常 APP 具有扫描添加任意设备和添加指定设备的功能，两种方法均可添加

APP 开启设备添加模式后，向模组发送“开始配网”命令，配网之前要先确认模组是路由器，终端节点或者休眠终端，其中终端节点和休眠终端可以通过“设置节点类型”进行切换。

发送“开始配网命令”

```
55 03 00 02 02
```

返回命令：

```
55 04 00 02 00 02
```

模组开始进入配网模式，等待 4 秒后，返回异步命令

```
55 29 80 01 [02] [E5 C7 F9 3E 7D 38 C1 A4] [0B] [26 B1] [4C 91] [55 A1 42 ED B3 6D B3 66] [E5 0F F3 F9 9A CC 9E 13 28 3C DF 2E DE 65 08 2E] 14
```

“[ ]”中内容依次为

- 配网成功
- MAC 地址= E5 C7 F9 3E 7D 38 C1 A4
- 网关信道=0x0B，11 信道，频率 2405MHz
- 网关 PANID=0xB126
- 短地址=0x914C
- 扩展 PANID=55 A1 42 ED B3 6D B3 66
- 网关的 NWK 层秘钥= E5 0F F3 F9 9A CC 9E 13 28 3C DF 2E DE 65 08 2E

同时在 APP 界面上，出现搜索到新设备的界面，是一个三路开关。

6.1.5 使用 APP 控制模拟的 3 路开关

进入 APP 的 3 路开关界面，触发开关 2。

模组收到 ZCL 控制命令：

55 0F 82 0F [A1] [00 00] [01] [66] [00] [06 00] [00 00] [DE] [01] 92

“[ ]”中内容依次为

- 接收模式=0xA1，收到信号强度有效，且不需要回复默认返回命令的控制命令，本地端口索引 1 收到该命令。根据创建端口时的记录，端口索引 1 对应端口 0x02。

- 源短地址=0x0000，这条控制命令是协调器发的。
- 源端口=0x01，这条命令是协调器用端口 0x01 发的。
- 帧序号=0x66
- 命令方向=0x00，是一条 Client->Sever 的命令，也就是控制端发给受控端的命令。
- 簇 ID=0x0006，是开关控制类的命令。
- 厂商码=0x0000
- 信号强度 RSSI=0xDE，换算得-33dbm
- 命令 ID=0x01，位于簇 ID=0x0006 下的命令 ID=0x01，对应的指令是“打开开关”，该命令无命令参数。

收到该命令后，把端口索引 1 下的属性“开关状态”改成“1”，该属性位于簇 ID=0x0006 的属性 ID=0x0000，因此使用命令“读写属性”修改该属性的值，注意开关状态切勿保存到 FLASH 中作为默认值。

发送读写属性命令：

```
55 0D 00 43 [01] [01] [00] [06 00] [00 00] [00 00] [01] 44
```

- 操作方式 = 0x01，写属性操作不保存 FLASH
- 端口索引 = 0x01，被写入的属性位于创建的第二个端口（索引 1）
- 属性方向 = 0x00，被写入的属性是 Server 端
- 簇 ID=0x0006，被写入的属性位于簇 0x0006
- 厂商码=0x0000，被写入的属性厂商码为 0x0000
- 属性 ID=0x0000，被写入的属性的 ID 是 0x0000
- 属性值=0x01

由于“开关状态”这个属性在添加时设置了自动上报，因此修改属性值后，会同步到 APP。

6.2 例程 2：模拟人体红外传感器接入网关

6.2.1 创建传感器端口并添加相关属性

创建一个 1 路的人体红外传感器，端口号使用 0x01。

首先把模组退出网络，可以使用“复位/恢复出厂”命令退出网络，也可以从 APP 发起删除设备操作。

发送“清空端口和属性”命令

```
55 03 00 44 44
```

返回命令

```
55 04 00 44 00 44
```

该命令表示执行成功。

创建一个传感器的端口：

```
55 11 00 40 [11] [01] [04 01] [02 04] [03] {[00 00] [03 00] [00 05]} [00] 56
```

“[ ]”中内容依次为

- 开关累计 0x11=17 个属性
- 创建端口号 = 0x01
- 轮廓 ID = 0x0104
- 设备 ID = 0x0402，对应安防传感器类设备
- 输入簇数量 = 0x03
- 输入簇{0x0000, 0x0003, 0x0500}
- 输出簇数量 = 0x00

返回命令：

```
55 12 00 40 [00] [00] [01] [04 01] [02 04] 03 00 00 03 00 00 05 00 47
```

“[ ]”中内容依次为

- 执行成功
- 模组分配的端口索引 = 0
- 创建端口号 = 0x01
- 轮廓 ID = 0x0104
- 设备 ID = 0x0402
- 输入簇数量 = 0x03
- 输入簇{0x0000, 0x0003, 0x0500 }
- 输出簇数量 = 0x00

ZCL 安防传感器类设备，需要添加以下属性：

名称	簇 ID	厂商码	属性 ID	数据类型	支持操作	属性方向
ZCL 版本	0x0000	0x0000	0x0000	0x20 (uint8)	0x01 (只读)	Sever
协议中版本	0x0000	0x0000	0x0001	0x20 (uint8)	0x01 (只读)	Sever
软件版本	0x0000	0x0000	0x0002	0x20 (uint8)	0x01 (只读)	Sever
硬件版本	0x0000	0x0000	0x0003	0x20 (uint8)	0x01 (只读)	Sever
厂商名称	0x0000	0x0000	0x0004	0x42 (string)	0x01 (只读)	Sever
产品型号	0x0000	0x0000	0x0005	0x42 (string)	0x01 (只读)	Sever

生产日期	0x0000	0x0000	0x0006	0x42 (string)	0x01 (只读)	Sever
供电方式	0x0000	0x0000	0x0007	0x30 (enmu8)	0x01 (只读)	Sever
属性终止符	0x0000	0x0000	0xFFFD	0x21 (uint16)	0x01 (只读)	Sever
Identity 标记状态	0x0003	0x0000	0x0000	0x21 (uint16)	0x03 (可读写)	Sever
属性终止符	0x0003	0x0000	0xFFFD	0x21 (uint16)	0x01 (只读)	Sever
安防注册状态	0x0500	0x0000	0x0000	0x30 (enmu8)	0x01 (只读)	Sever
安防传感器类型	0x0500	0x0000	0x0001	0x31 (enmu16)	0x01 (只读)	Sever
安防报警状态	0x0500	0x0000	0x0002	0x19 (bitmap16)	0x01 (只读)	Sever
报警上报 MAC 地址	0x0500	0x0000	0x0010	0xF0 (EUI 64)	0x03 (可读写)	Sever
防区 ID	0x0500	0x0000	0x0011	0x20 (uint8)	0x01 (只读)	Sever

6.2.2 设置属性初始化值并接入网关

添加完以上属性后，保存并复位重启，然后修改属性“厂商名称”，“产品型号”，“生产日期”。只需要输入有效的字符串即可。

安防传感器，需要编辑属性“安防传感器类型”，按照 ZCL 规范的表格中“Motion sensor”对应的值为 0x000D，修改属性值并保存 FLASH。

输入读写属性命令修改“安防传感器类型”

```
55 0E 00 43 [02] [00] [00] [00 05] [00 00] [01 00] [0D 00] 48
```

“[ ]”中内容依次为

- 操作方式 = 0x02，写属性且保存 FLASH
- 端口索引 = 0x00，被写入的属性位于创建的第一个端口
- 属性方向 = 0x00，被写入的属性是 Server 端
- 簇 ID=0x0500，被写入的属性位于簇 0x0500
- 厂商码=0x0000，被写入的属性厂商码为 0x0000
- 属性 ID=0x0001，被写入的属性的 ID 是 0x0001
- 写入的属性值=0x000D

然后复位模组，再把模组加入网关。

6.2.3 传感器触发报警

发送 ZCL 控制命令：

```
55 15 02 0F [00] [00 00] [01] [75] [01] [00 05] [00 00] [00] [00] [01 00 00 FE 01 00] 83
```

“[ ]”中内容依次为

- 发送模式=0x00，端口索引 0 发送命令，无任何模式。
- 目标短地址=0x0000
- 目标端口=0x01
- 帧序号=0x75
- 命令方向=Server->Client，这是一条受控端发给控制端的命令，传感器都是受控端。
- 簇 ID=0x0500
- 厂商码=0x0000
- 应答模式=0
- 命令 ID=0x00
- 命令参数=[01 00 00 FE 01 00]

按照《Zigbee Cluster Library》解析命令参数，安防报警命令的参数如表中所示

Bits	16	8	8	16
Data Type	map16	map8	uint8	uint16
Field Name	Zone Status	Extended Status	Zone ID	Delay

命令参数：

```
[01 00] [00] [FE] [01 00]
```

“[ ]”中内容依次为

- 报警状态=0x0001，按照传感器协议规范还需要操作写属性“安防报警状态”为该值。
- 扩展状态=0x00
- 防区 ID=0xFE，该值也需要同步写属性“防区 ID”
- 报警延迟=0x0001，单位 250ms。

附：传感器触发控制命令的注意事项：

- 传感器多为休眠终端，发送命令失败率较高，每次重传报警命令时需要根据时间累加“报警延迟”
- 报警命令中的“报警状态”和“防区 ID”，同设备中的属性“安防报警状态”和“防区 ID”需要同步，建议 MCU 在运行中检测到报警，先写这两个属性值，再以 250ms 为周期定时读这两个属性并发送报警命令，直到报警命令发送成功才停止定时。

发出 ZCL 控制命令后收到反馈命令：

```
55 05 02 0F [00] 75 78
```

- 发送命令有效

收到发送确认命令：

```
55 0A 8F 02 [00] [00 00] [01] [75] [01] [00] F8
```

“[ ]”中内容依次为

- 发送模式=0x00，端口索引 0 发送命令，无任何模式。
- 目标短地址=0x0000
- 目标端口=0x01
- 帧序号=0x75
- 命令方向=Server->Client，这是一条受控端发给控制端的命令，传感器都是受控端。

- 发送结果=0x00，发送成功

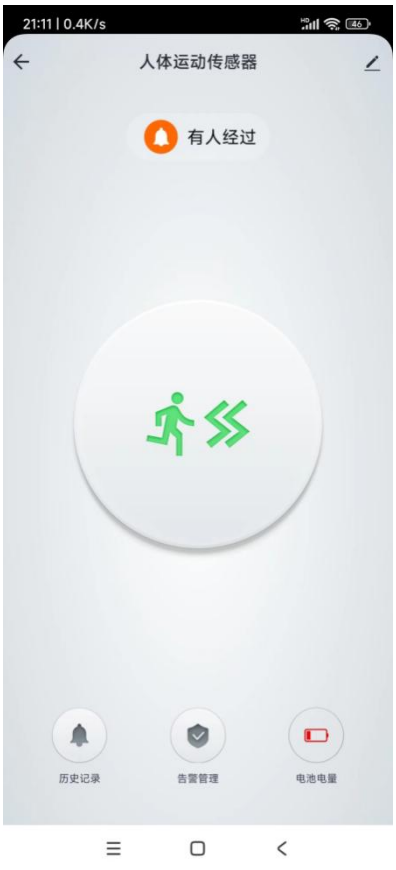
由于发送时未关闭默认返回帧，因此还收到了协调器发来的默认返回帧。

55 10 82 0B [20] [00 00] [01] [75] [00] [00 05] [00 00] [C8] [00] [00] 10

“[ ]”中内容依次为

- 接收模式=0x20，收到信号强度有效，本地端口索引 0 也就是端口 0x02 收到该命令。
- 源短地址=0x0000，这条控制命令是协调器发的。
- 源端口=0x01，这条命令是协调器用端口 0x01 发的。
- 帧序号=0x75，和发送的 ZCL 控制命令的帧序号相同。
- 命令方向=0x00，是一条 Client->Sever 的命令，也就是控制端发给受控端的命令。
- 簇 ID=0x0500，是安防传感器类的命令。
- 厂商码=0x0000
- 信号强度 RSSI=0xC8，换算得-56dbm
- 命令 ID=0x00，对应刚发的控制命令
- ZCL 状态=0x00，刚才发的命令被网关正确接收处理。

该命令发出后，APP 上显示传感器报警



第七章 常见问题

7.1 传输距离不理想

- 当存在直线通信障碍时，通信距离会相应的衰减；
- 温度、湿度，同频干扰，会导致通信丢包率提高；
- 地面吸收、反射无线电波，靠近地面测试效果较差；
- 海水具有极强的吸收无线电波能力，故海边测试效果差；
- 天线附近有金属物体，或放置于金属壳内，信号衰减会非常严重；
- 功率寄存器设置错误、空中速率设置过高（空中速率越高，距离越近）；
- 室温下电源低压低于推荐值，电压越低发功率越小；
- 使用天线与模块匹配程度较差或天线本身品质问题。

7.2 模块易损坏

- 请检查供电电源，确保在推荐供电电压之间，如超过最大值会造成模块永久性损坏；
- 请检查电源稳定性，电压不能大幅频繁波动；
- 请确保安装使用过程防静电操作，高频器件静电敏感性；
- 请确保安装使用过程湿度不宜过高，部分元件为湿度敏感器件；
- 如果没有特殊需求不建议在过高、过低温度下使用。

7.3 误码率太高

- 附近有同频信号干扰，远离干扰源或者修改频率、信道避开干扰；
- 电源不理想也可能造成乱码，务必保证电源的可靠性；
- 延长线、馈线品质差或太长，也会造成误码率偏高。

修订历史

版本	修订日期	修订说明	维护人
1.0	2024-01-04	初始版本	Bin

关于我们

销售热线：4000-330-990 技术支持：support@cdebyte.com
官方网站：www.ebyte.com
公司地址：四川省成都市高新西区西区大道 199 号 B5 栋

成都亿佰特电子科技有限公司
Chengdu Ebyte Electronic Technology Co.,Ltd.