



# 串口协议

文档版本: 20240726

[查看在线版本](#)

## 目录

<b>1. 硬件连接 .....</b>	<b>2</b>
<b>2. 低功耗唤醒 .....</b>	<b>3</b>
2.1. 电平唤醒 .....	3
2.2. 脉冲唤醒 .....	3
<b>3. 波特率自适应 .....</b>	<b>5</b>
<b>4. 串口帧格式说明 .....</b>	<b>6</b>
4.1. 命令字索引 (Command ID) .....	6
<b>5. 配置类命令 .....</b>	<b>9</b>
5.1. 查询产品信息 (0x01) .....	9
5.2. 模组网络状态通知 (0x02) .....	10
5.3. 解绑并清除数据通知 (0x00) .....	11
5.4. 模组配网/复位 (0x03) .....	12
5.5. 配置唤醒等待时间 (0x2B) .....	13
5.6. 查询模组网络状态 (0x20) .....	14
5.7. 查询网关联网状态 (0x25) .....	15
5.8. 配置 Zigbee 网络参数 (0x26) .....	16
<b>6. 应用数据收发类命令 .....</b>	<b>22</b>
6.1. DP 消息接收 (0x04) .....	22
6.2. DP 消息接收-群控 (0x2A) .....	24
6.3. DP 消息应答 (0x05) .....	25
6.4. DP 消息上报-可触发联动 (0x06) .....	26
6.5. DP 消息上报-不触发联动 (0x2C) .....	28
6.6. DP 查询 (0x28) .....	29
6.7. 发送广播 DP 消息 (0x27) .....	30
6.8. 发送群组 DP 消息 (0x43) .....	32
6.9. 发送群组标准命令 (0x42) .....	33
<b>7. 场景类命令 .....</b>	<b>36</b>
7.1. 标准场景 .....	36
7.2. 联动场景 .....	37
7.3. 场景配置-场景控制器设备 (0x41) .....	37
7.4. 场景触发-场景控制器设备 (0x0A) .....	38
<b>8. 产测类命令 .....</b>	<b>40</b>
8.1. 模组 RF 测试 (0x08) .....	40
8.2. 信标产测通知 (0x29) .....	41
<b>9. MCU OTA 类命令 .....</b>	<b>43</b>
9.1. MCU 固件版本查询 (0x0B) .....	43

---

9.2. OTA 升级通知 (0x0C) .....	44
9.3. OTA 文件请求 (0x0D) .....	46
9.4. OTA 结果上报 (0x0E) .....	47
<b>10. GPIO 控制类命令 .....</b>	<b>49</b>
10.1. GPIO 配置 (0x36) .....	50
10.2. GPIO 电平读取 (0x37) .....	52
10.3. GPIO 输出控制 (0x38) .....	53
10.4. GPIO 输入中断通知 (0x39) .....	55
<b>11. 时间和天气 .....</b>	<b>56</b>
11.1. 时间同步 (0x24) .....	56
11.2. 天气信息查询 (0x3A) .....	57
11.3. 天气信息通知 (0x3B) .....	60
<b>12. 附表 .....</b>	<b>64</b>
12.1. 附表 1: 天气数据 .....	64
12.2. 附表 2: 中文天气 UTF-8 编码 .....	65
12.3. 附表 3: 风向简码编码表 .....	65
<b>13. 更新记录 .....</b>	<b>66</b>

涂鸦 Zigbee 串口通信模组（以下简称模组），适用于采用 MCU + Zigbee 模组方案的产品。这类产品通常由 MCU 负责处理产品本身的逻辑，搭配 Zigbee 模组实现无线通信，MCU 与 模组之间通过 UART 串口通信。模组的主要功能有：

- **Zigbee 网络管理**：例如设备配网、离网、本地群组和本地场景。
- **Zigbee 网关通信**：设备与网关通信，从而实现设备与涂鸦云的数据交互（仅支持涂鸦 DP 协议）。
- **Zigbee 设备通信**：子设备与子设备之间进行通信，支持涂鸦 DP 协议和部分 Zigbee 3.0 标准协议。
- **MCU OTA**：可通过 Zigbee 模组接收 MCU 的 OTA 文件，完成 MCU 端的 OTA 升级。

串口协议不断地完善迭代，例如新增命令字，优化命令交互时序等。本协议尽量做到向前兼容，但是无法保障部分低版本固件或非长期维护固件的协议一致性。

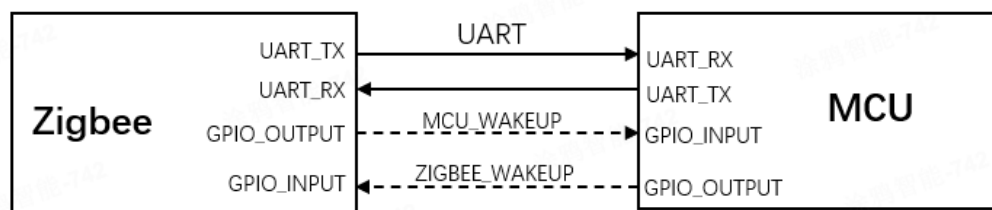


本文是涂鸦 Zigbee 串口通信类模组协议的集合，同时包含了低功耗设备和非低功耗设备。不同模组因芯片平台不同，或相同模组不同的固件版本，其所支持的具体命令和时序可能会有所不同。需以实际模组为准或向涂鸦咨询，建议采用公版固件最新版本。

## 1. 硬件连接

Zigbee 模组与 MCU 硬件连接如下图所示。

- 对于非低功耗设备，只需要连接 UART TX 和 UART RX 两个管脚。
- 对于低功耗设备，则需另外连接两个唤醒管脚，分别用于 Zigbee 模组和 MCU 唤醒对方，让对方准备接收串口数据。



### 串口通信参数

- 波特率：9600/115200 bps
- 数据位：8
- 奇偶校验：无
- 停止位：1
- 数据流控：无
- 供电电压：Zigbee 模组和 MCU 主控均采用 DC 3.3V

下表列出了部分 Zigbee 模组对应的串口管脚和唤醒管脚。固件不同，可能会有所差异，请以实际为准。

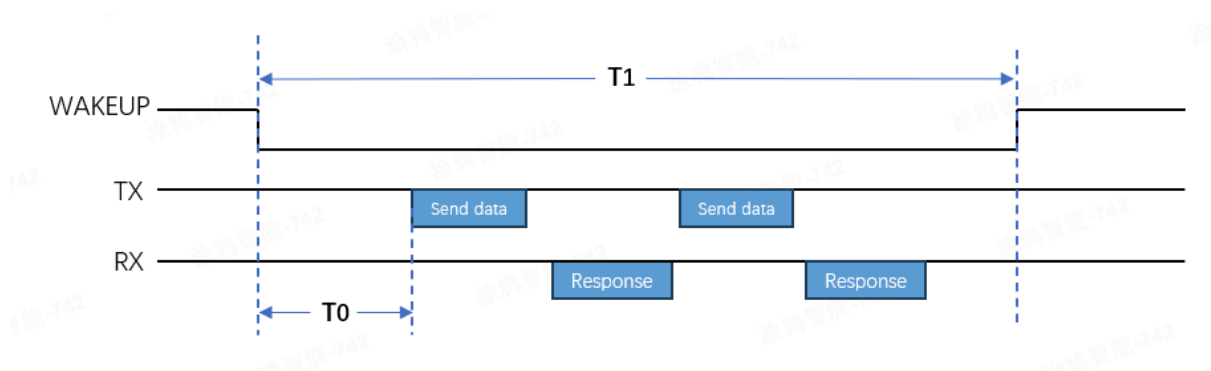
模组型号	模组 TX	模组 RX	唤醒 MCU	唤醒模组
ZS3L	PA5	PA6	PC1	PA3
TYZS3	PA0	PA1	PF5	PA2
TYZS5	PA0	PA1	PA4	PA3
ZTU	PB1	PB7	PB4	PB5/PD2
ZT5	PB1	PB7	PB4	PB5/PD2
TYZS5	PB1	PB7	PB4	PB5/PD2

## 2. 低功耗唤醒

低功耗设备在进入睡眠模式后，可能无法接收串口数据。因此，Zigbee 模组和 MCU 在主动向对方发送串口数据之前，应先将对方唤醒，待本次所有串口数据发送完毕后，通知对方进入睡眠模式。

### 2.1. 电平唤醒

通过拉低唤醒管脚电平唤醒对方，待对方被唤醒后再开始发送串口数据，本次数据发送完毕后再拉高唤醒管脚。当无数据发送时，唤醒管脚应保持高电平，如下图所示。



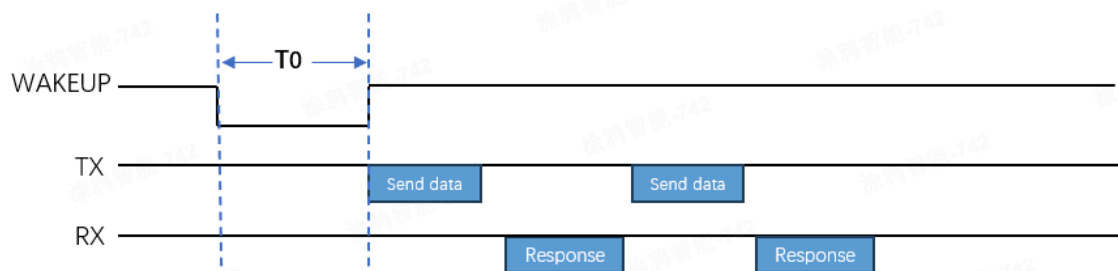
T0 是拉低唤醒管脚后的等待时间（待对方退出睡眠模式），T1 是单次唤醒对方的总时间。



- MCU 唤醒 Zigbee 模组，T0 建议 1 ~ 10 毫秒。对于 ZT 系列模组，T0 至少 10 毫秒。
- MCU 唤醒 Zigbee 模组，唤醒总时间 T1 不能超过 2 分钟，否则 Zigbee 模组将复位重启。
- Zigbee 模组唤醒 MCU，T0 是 5 毫秒，MCU 也可通过 0x2B 命令对此 T0 进行配置。

### 2.2. 脉冲唤醒

MCU 在唤醒管脚上发送一个低电平脉冲，以此来唤醒 Zigbee 模组，并开始串口数据发送。脉冲唤醒比电平唤醒能够节省更多的电量。



- 是否支持脉冲唤醒，与 Zigbee 模组固件 Key 和版本有关。
- 脉冲唤醒只适用于 MCU 唤醒模组，模组不会采用此方式来唤醒 MCU。
- 脉冲宽度  $T_0$  建议 ZS 系列模组 1 ~ 5 毫秒，ZT 系列模组至少 10 毫秒。

### 3. 波特率自适应

Zigbee 模组支持两种串口波特率，分别是 9600 bps 和 115200 bps，MCU 可以采用这两种波特率中的任意一种。模组有波特率自适应机制，即在首次与 MCU 通讯时，模组会交替使用 9600 bps 和 115200 bps 波特率发送 [查询产品信息](#) 命令，MCU 在其中一个波特率下回复命令后，模组将该波特率保存，并持续生效。



- Zigbee 模组固件 Key 和版本号不同，波特率自适应流程可能会有差异。
- 波特率自适应流程，只在模组出厂模式下与 MCU 首次通信时进行，目前机制是波特率确认后不可修改。



## 4. 串口帧格式说明

Frame Head	Version	SEQ	Command ID	Data Length	Data	Checksum
2 bytes	1 byte	2 bytes	1 byte	2 bytes	N bytes	1 byte

字段	中文	说明
Frame Head	帧头	固定为 0x55AA
Version	版本号	串口通信协议版本，升级扩展用，本协议版本号为 0x02
SEQ	序列号	传输数据序列号 (sequence number)，0 ~ 0xffff0 循环，下文用 xx xx 表示
Command ID	命令字	命令 ID，持续更新
Data Length	数据长度	传输的有效数据长度
Data	数据	传输的有效数据
Checksum	校验和	用于数据校验，从帧头开始按字节求和得出的结果对 256 求余，下文用 xx 表示



不同 Zigbee 模组固件，在进行无线数据收发时有 **不支持分包传输** 和 **支持分包传输** 两种。

- **不支持分包传输：**受 Zigbee 无线数据单帧长度限制，发送和接收 Data 最大支持 62 字节。
- **支持分包传输：**接收 Data 最大支持 120 字节，发送 Data 最大支持 246 字节。
- 超过 1 字节数据，均采用大端格式。

### 4.1. 命令字索引 (Command ID)

命令字	发送方	说明	强电版本	低功耗版本	场景版本
0x01	Z	查询产品信息	√	√	√
0x02	Z	模组网络状态通知	√	√	√

命令字	发送方	说明	强电版本	低功耗版本	场景版本
0x03	M	模组配网/复位	√	√	√
0x00	Z	App 端解绑并清除数据通知	√	√	√
0x2B	M	配置唤醒等待时间	×	√	×
0x20	M	查询模组网络状态	√	√	√
0x25	M	查询网关联网状态	√	√	×
0x26	M	配置 Zigbee 网络参数	√	√	×
0x04	Z	DP 消息接收	√	√	√
0x2A	Z	DP 消息接收 (群控)	√	√	×
0x05	M	DP 消息应答	√	√	√
0x06	M	DP 主动上报 (可触发联动)	√	√	√
0x2C	M	DP 主动上报 (不触发联动)	√	√	√
0x28	Z	DP 查询	√	√	√
0x27	M	广播消息发送	√	√	√
0x42	M	标准命令组播发送	×	×	√
0x43	M	私有命令组播发送	×	×	√
0x41	Z	场景配置 (场景控制类设备)	×	×	√

命令字	发送方	说明	强电版本	低功耗版本	场景版本
0x0A	M	场景触发 (场景控制类设备)	×	×	√
0x08	M	模组 RF 测试	√	√	√
0x29	Z	信标产测通知	√	√	√
0x0B	Z	MCU 固件版本查询	√	√	√
0x0C	Z	OTA 升级通知	√	√	√
0x0D	M	OTA 固件请求	√	√	√
0x0E	M	OTA 升级结果上报	√	√	√
0x36	M	GPIO 配置	√	√	√
0x37	M	GPIO 电平读取	√	√	√
0x38	M	GPIO 输出控制	√	√	√
0x39	M	GPIO 中断通知	√	√	√
0x24	M	时间同步	√	√	√
0x3A	M	天气信息查询	√	√	√
0x3B	Z	天气信息通知	√	√	√

## i

- **M**: 发送方是 MCU，表示该命令是由 MCU 发送给 Zigbee 模组。
- **Z**: 发送方是 Zigbee 模组，表示该命令是由 Zigbee 模组发送给 MCU。

## 5. 配置类命令

### 5.1. 查询产品信息 (0x01)

Zigbee 模组每次上电后都会主动向 MCU 发送此命令，MCU 需在应答此命令后才能主动发送其他命令给模组。**产品信息** 包括产品的 PID 和 MCU 固件版本号等。同时，此命令也有 Zigbee 模组在出厂状态下，首次与 MCU 通信时用于确认 MCU 端的波特率的作用。

#### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x01
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例： 55 AA 02 xx xx 01 00 00 xx

#### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x01
数据长度	2	N
数据	1	产品信息，示例 {“p”:“Alp08kLI”,“v”:“2.0.0”,“g”:“1”}
校验和	1	xx

**产品信息** 是 Json 格式的字符串，包含大括号 {}，大括号内部是 "key":"value" 格式，下面是已支持的 key 和 value。

- **p**：必选，对应该产品在涂鸦开发者平台的 PID，例如上例中的 AIp08kLI。
- **v**：必选，MCU 固件版本号，x.y.z 格式，其中 x：0~3，y：0~3，z：1~15，因此可表示的固件版本范围是：0.0.1~3.3.15。
- **g**：可选，此字段是一个标志位，用于表示 MCU 是否需要区分单播消息和组播（含广播）消息。模组在收到组播消息后，只有当 **g** 为 1 时，才会以 **0x2A 命令** 命令通知 MCU。否则，发送 **0x04 命令**。建议 MCU 端支持此字段。

示例：55 AA 02 xx xx 01 00 48

7b2270223a2241497030386b4c49222c2276223a22322e302e30222c2267223a2231227d xx

本例中，**产品信息** 部分为：{"p":"AIp08kLI","v":"2.0.0","g":"1"}。

## 5.2. 模组网络状态通知 (0x02)

网络状态是指 Zigbee 本地网络状态（非互联网连接状态），当模组加入网关后即为 **已入网**。

网络状态不会因为网关断电、父节点丢失等原因变更。当模组的网络状态发生变化时，会主动通知 MCU。

### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x02
数据长度	2	0x0001
数据	1	模组网络状态
校验和	1	xx

模组网络状态：

- 0x00：未入网，例如设备首次上电、设备配网失败、App 移除等。
- 0x01：已入网，表示设备已加入一个 Zigbee 网络。
- 0x02：网络异常。
- 0x03：设备正在配网中。

示例： 55 AA 02 xx xx 02 00 01 01 xx 模组已入网

#### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	N
命令字	1	0x02
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例： 55 AA 02 xx xx 02 00 00 xx

### 5.3. 解绑并清除数据通知 (0x00)

在 App 端将设备 **解绑并清除数据** 时，模组以此命令通知 MCU。MCU 收到该通知后可清除本地数据，即恢复出厂设置。

#### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x00
数据长度	2	0x0001
数据	1	0x01 默认值
校验和	1	xx

数据字段目前默认是 0x01，保留该字段将来用作拓展。

示例： 55 AA 02 xx xx 00 00 01 01 xx

## MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	N
命令字	1	0x00
数据长度	2	0x0001
数据	1	0x01
校验和	1	xx

示例: 55 AA 02 xx xx 00 00 01 01 xx

## 5.4. 模组配网/复位 (0x03)

MCU 可以发送此命令让模组进入配网或者软件复位。

## MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x03
数据长度	2	0x0001
数据	1	0x00: 模组复位重启 0x01: 模组配网（先离网，再配网）
校验和	1	xx

示例: 55 AA 02 xx xx 03 00 01 01 xx 模组配网

## 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x03
数据长度	2	0x0000
数据	1	无
校验和	1	xx

示例： 55 AA 02 xx xx 03 00 00 xx

### 5.5. 配置唤醒等待时间 (0x2B)

该命令针对低功耗类设备，在模组需要主动发送数据给 MCU 时，会先拉低 MCU 唤醒管脚，并等待一个时间 T0 后再开始发送数据。MCU 在此等待时间内，准备好串口数据接收。由于不同 MCU 所需要的等待时间不同，MCU 可以通过此命令对等待时间 T0 进行配置。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x2B
数据长度	2	0x0002
数据	1	单位 ms，默认值为 5ms， 取值范围 3 ~ 300ms
校验和	1	xx

示例： 55 AA 02 xx xx 2B 00 02 00 0A xx 配置唤醒等待时间为 10 ms

#### 模组响应



字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x2B
数据长度	2	0x0001
数据	1	0x00：失败 0x01：成功
校验和	1	xx

示例： 55 AA 02 xx xx 2B 00 01 01 xx 配置成功

## 5.6. 查询模组网络状态（0x20）

模组网络状态发生变化时，会通过 [模组网络状态通知](#) 主动通知 MCU，MCU 也可以主动查询模组当前的网络状态。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x20
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例： 55 AA 02 xx xx 20 00 00 xx

### 模组应答

字段	字节数	说明
----	-----	----

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x20
数据长度	2	0x0001
数据	1	模组网络状态
校验和	1	xx

模组网络状态 参考 模组网络状态通知 命令

示例： 55 AA 02 xx xx 20 00 01 01 xx 模组当前状态为：已入网

## 5.7. 查询网关联网状态 (0x25)

通过此命令可查询 Zigbee 网关是否连接互联网。若网关回复互联网在线，可以认为本产品也是互联网在线的，MCU 可以据此做一些状态显示。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x25
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例： 55 AA 02 xx xx 25 00 00 xx

### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x25
数据长度	2	0x0001
数据	1	0x00: 网关互联网离线
		0x01: 网关互联网在线
		0x02: 网关回复超时
校验和	1	xx

示例： 55 AA 02 xx xx 25 00 01 01 xx 网关当前互联网在线

## 5.8. 配置 Zigbee 网络参数 (0x26)

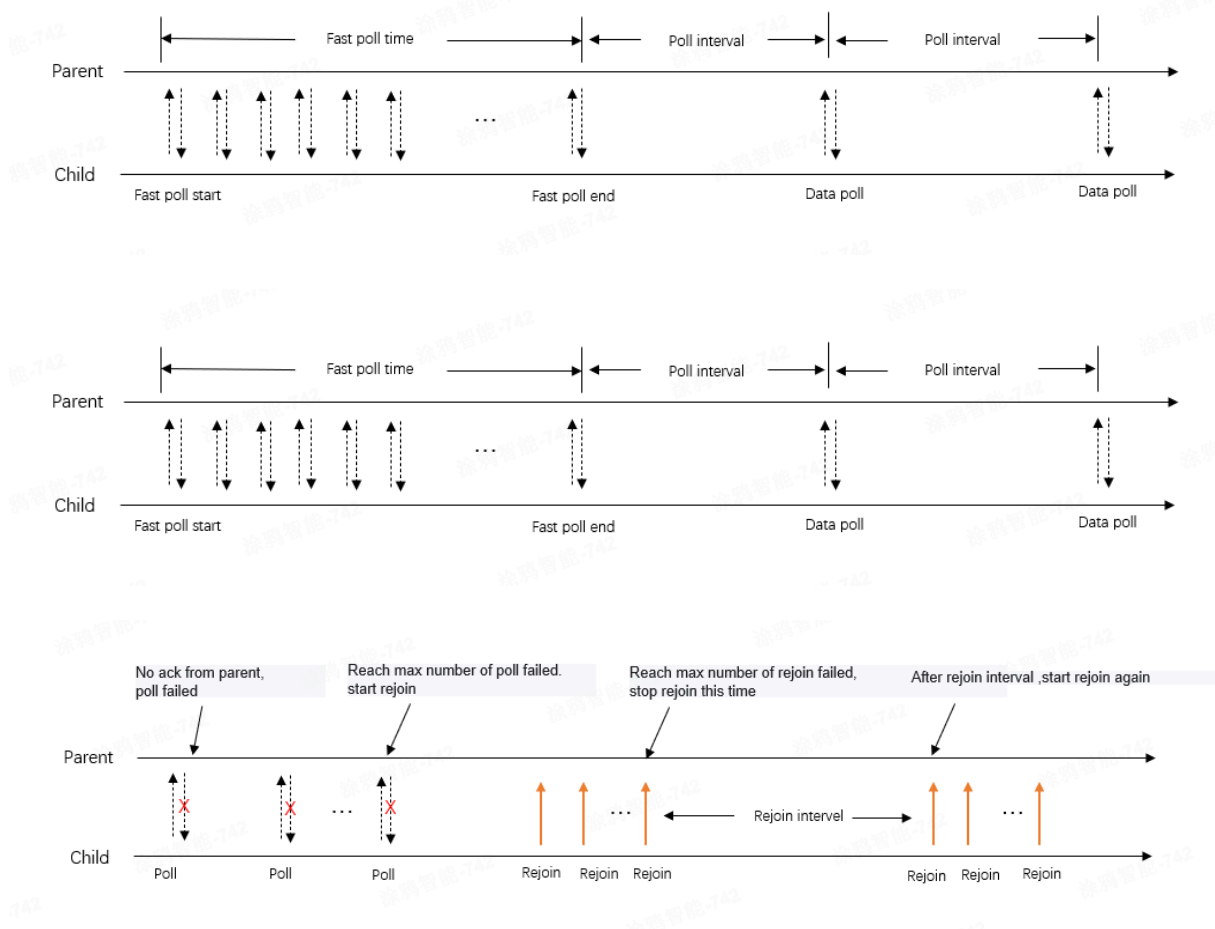
本命令可以设置模组的网络参数。

- 对于非低功耗类设备，只需要关注配网时间和发射功率两个参数即可。
- 对于低功耗类设备，要理解参数的含义，需要先了解 Zigbee 低功耗设备的唤醒和数据收发机制。

### 低功耗参数

按照 Zigbee 协议规范，低功耗设备在加入到 Zigbee 网络后，会以某个非低功耗设备（路由设备或者网关）作为父设备，形成父子关系。低功耗设备大多数时候处于休眠状态（为了省电），此状态下任何发送给该设备的消息都是暂存在其父设备处。子设备可以定期 (Poll interval) 醒来，询问其父设备是否有数据 (Poll data)。如果父设备处有缓存的数据，则会将数据发给子设备，反之，子设备可以再次进入休眠。

如果子设备在向父设备请求数据时，父设备没有任何应答 (有可能是该父设备断电了)，则认为本次数据请求失败 (Poll failed)。子设备会在多次请求失败后 (Poll max times) 认为该父设备丢失了 (Parent Lost)，于是尝试寻找新的父设备以期能够回连 (Rejoin) 到原来的 Zigbee 网络中。如果连续多次回连也失败了 (Rejoin max times)，则停止这一轮的回连，在一定时间后 (Rejoin interval) 重新尝试下一轮的回连。



#### 低功耗参数概念：

- **Poll interval:** Poll 间隔，也就是设备正常休眠状态下的定期唤醒周期。通常需要实时接收数据的设备，可以将定期唤醒周期设置得比较短（例如单火开关 250 ms，门锁 1000ms）。传感器类设备可以关闭周期唤醒（Poll interval = 0），这样模组不会主动唤醒，在有数据需要上报时通过 MCU 唤醒模组即可。周期唤醒时间设置得越短，对于网关数据下发的实时性越好，同时也更加耗电。
- **Poll failed max times:** Poll 失败的最大尝试次数，达到最大次数后会触发 Rejoin。
- **Fast poll time:** 快速 Poll 总时间，模组在设备配网成功、已入网设备上电，Rejoin 成功时会开启快速 Poll。快速 Poll 的目的是为了加快与网关的数据交互，此时的 Poll 间隔为默认的 250 ms。
- **Rejoin interval:** Rejoin 轮次时间间隔，即本轮 Rejoin 失败，到下一轮 Rejoin 开始的时间间隔，默认是 3 分钟。注意 Rejoin 是因为发现丢失了父节点（Parent Lost）后才会触发的动作，Rejoin interval 是指每一轮 Rejoin 的间隔，而不是每一个 Rejoin 请求报文的发送间隔。

每一轮中会有多个 Rejoin 请求报文，可以通过下发的 Rejoin max times 参数，设置每一轮发起多少次 Rejoin 请求包。

- **Rejoin max times**：每一轮 Rejoin 请求尝试次数，当 Rejoin 成功后会自动停止。当到达此次数如果依然失败，也会停止，并等待下一轮。或者等待 MCU 发送无线消息时触发下一轮 Rejoin。

### 主动心跳间隔

本命令可以设置子设备与网关之间的主动心跳间隔，所谓 **心跳** 是 Zigbee 设备与涂鸦 Zigbee 网关维护数据链路的手段，即设备每隔一个固件周期给网关发送一个心跳包。

- 对于低功耗设备，当网关检测到该设备长时间没有心跳时，会认为其已离线，此时在涂鸦 App 上也会显示该设备离线。
- 对于非低功耗设备，网关会主动访问设备（作为心跳查询），因此非低功耗设备是否主动发送心跳包给网关，已经作用不大了。
- 目前仅可修改低功耗设备的心跳包发送周期，非低功耗设备不可修改。

### 配网超时时间

当 MCU 发送配网指令之后，模组开启配网。MCU 可以预先配置模组配网的超时时间，超时时间默认为 180 秒，可配置范围为 30 ~ 600 秒。配网开始后，模组会通过 [模组网络状态通知](#) 命令通知 MCU 网络状态的变化，例如 **配网中**、**配网超时**、**配网成功**。

### 发射功率

模组发送无线数据的功率。默认值为 11 dBm，取值范围 3 ~ 19 dBm，与模组的芯片平台有关。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x26
数据长度	2	0x000E
数据	0x0E	Zigbee 网络参数
校验和	1	xx

### Zigbee 网络参数

字节偏移	参数名称	适用设备	说明
2	心跳间隔	低功耗	单位：秒，可配置 10 ~ 5*3600 秒。 默认 4*3600 秒（4 小时）。
2	配网超时时间	通用	单位：秒，可配置 30 ~ 600 秒。默认 180 秒。
2	Rejoin 间隔	低功耗	单位：秒，可配置 3 ~ 3600 秒。默认 180 秒，即如果丢 失了父节点，默认 每隔 180 秒尝试 Rejoin。
2	Poll 间隔	低功耗	单位：毫秒，可配 置 0 或 200 ~ 10000 毫秒，0 表 示关闭主动唤醒。 默认值 5000 毫秒。
2	快速 Poll 时间	低功耗	单位：秒，可配置 10 ~ 3000。默认 30 秒。
1	Poll 失败最大尝试 次数	低功耗	可配置 3 ~ 40，当 达到最大值时触发 Rejoin。默认值 4。
1	MCU 消息发送触发 Rejoin	低功耗	0：不触发 1：触 发，默认值。丢失 了父节点时，MCU 发送无线消息是否 自动触发 Rejoin。

字节偏移	参数名称	适用设备	说明
1	Rejoin 报文发送次数	低功耗	可配置 1 ~ 10 次，默认值为 1。
1	发射功率	通用	单位：dBm，可配置 3 ~ 19。默认值为 11。模组支持的最大发射功率与芯片平台有关，请参考模组规格书，例如 ZT、ZP 系列模组最大支持 11 dBm。

以上 Zigbee 网络参数的特殊值：

- 2 字节参数，0xFFFF 表示采用上一次配置的值，0xFFFE 表示采用默认值。
- 1 字节参数，0xFF 表示采用上一次配置的值，0xFE 表示采用默认值。

示例：55 AA 02 xx xx 26 00 0E FF FE 00 64 FF FE 07 D0 00 32 FE 01 FE FE xx

本示例中 Zigbee 网络参数如下：

- 心跳间隔：0xFFFE，即采用默认值。非低功耗设备 150 ~ 180 秒随机，低功耗设备 4\*3600 秒。
- 配网超时时间：0x0064，即配网超时时间为 100 秒。
- Rejoin 间隔：0xFFFE，即采用默认值 180 秒。
- Poll 间隔：0x07D0，即设备定期唤醒间隔为 2000 毫秒。
- 快速 Poll 时间：0x0032，即设备进入快速唤醒的总时长为 50 秒。
- Poll 失败最大尝试次数：0xFE，即采用默认值 4 次。
- MCU 消息发送触发 Rejoin：0x01 即使能。
- Rejoin 报文发送次数：0xFE，即采用默认值 1 次。
- 发射功率：0xFE，即采用默认值 11 dBm。

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA

字段	字节数	说明
版本	1	0x02
序列号	2	xx xx
命令字	1	0x26
数据长度	2	0x0001
数据	1	0x00: 配置失败 0x01: 配置成功
校验和	1	xx

示例： 55 AA 02 xx xx 26 00 01 01 xx 表示配置成功



## 6. 应用数据收发类命令

### 6.1. DP 消息接收 (0x04)

当模组收到 DP 消息时通过此命令通知 MCU，DP 消息通常来自 Zigbee 网关或其他 Zigbee 子设备。例如，通过涂鸦智能 App 下发一个 DP 消息给设备时，该消息会先到达网关，然后由网关通过 Zigbee 网络发送给设备。MCU 在收到此命令后，需要进行 **DP 消息应答** (只应答对应的 DP 消息) 这样涂鸦智能 App 上该设备的状态才会相应地更新。

#### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x04
数据长度	2	N
数据	N	DP 消息格式
校验和	1	xx

#### DP 消息格式:

DP Message								
DP1				DP2				...
DP ID	Type	Length	Value	DP ID	Type	Length	Value	...
1 byte	1 byte	2 bytes	N bytes	2 bytes	1 byte	2 bytes	N bytes	...

接收的消息中，可能会同时包含多个 DP 数据，每一个 DP 数据的格式如下：

- DP ID：对应该产品 (PID) 在涂鸦开发者平台的 DP ID。
- Type：对应该 DPID 的数据类型，部分数据类型其 Length 是固定的。
- Length：Value 部分的数据长度。
- Value：DP 有效数据。

#### Type:

Type	对应的 DP 类型	数据长度	说明
------	-----------	------	----

Type	对应的 DP 类型	数据长度	说明
0x00	Raw	自定义	Raw 数据，Hex 格式，可自定义每个字节的含义
0x01	Bool	1	布尔型，0x00、0x01
0x02	Value	4	数值型，例如 1，23，104
0x03	String	自定义	字符串，可自由定义含义
0x04	Enum	1	枚举型，范围 0-255
0x05	Bitmap	1/2/4	Bitmap 型，主要用于错误码上报等

示例： 55 AA 02 xx xx 04 00 05 03 01 00 01 01 xx 模组收到 DP 消息，包含 1 个 DP 数据，如下所示：

- DP ID: 03
- Type: 01
- Length: 00 01
- Value: 01

**i**

- DP 消息中可能会包含多个 DP 数据结构，比如 App 同时下发了多个 DP 给设备。
- 当 DP 消息是 Raw 类型时，只会有 1 个 DP 数据结构，即 Raw 类型的 DP 不会多个同时下发，也不会和其他类型的 DP 混合下发。

## MCU 回复

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x04
数据长度	2	0x0000
数据	0	无

字段	字节数	说明
校验和	1	xx

示例： 55 AA 02 xx xx 04 00 00 xx

## 6.2. DP 消息接收-群控 (0x2A)

当模组收到群控（包括组播、广播）DP 消息后，采用此命令通知 MCU。如果使用的是涂鸦网关，那么 MCU 在收到此消息后，不需要进行 DP 消息应答，涂鸦网关在发送群控消息后，会主动读取 DP。

该命令的 **数据** 部分与 [DP 消息接收 \(0x04\)](#) 相同。



- 该命令对 Zigbee 模组固件版本有要求，低版本的模组固件可能不支持该命令，只支持 0x04 命令。
- 该命令只在 MCU 应答 [查询产品信息 \(0x01\)](#) 命令时，**数据** 部分包含了 "g": "1" 时才会触发，否则模组收到群控消息后依然通过 [DP 消息接收 \(0x04\)](#) 命令通知 MCU。

### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x2A
数据长度	2	N
数据	N	DP 消息格式
校验和	1	xx

示例： 55 AA 02 xx xx 2A 00 04 01 01 00 01 01 xx 表示模组收到了一条群控 DP 消息

- DP ID: 01
- Type: 00
- Length: 01

- Value: 01

#### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x2A
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例: 55 AA 02 xx xx 2A 00 00 xx

### 6.3. DP 消息应答 (0x05)

当 MCU 收到 [DP 消息接收 \(0x04\)](#) 后, 可通过此命令进行 DP 消息应答, 以同步云端数据和 App 面板状态, 本消息的 **数据** 字段与 0x04 命令相同。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x05
数据长度	2	N
数据	N	DP 消息格式
校验和	1	xx

示例: 55 AA 02 xx xx 05 00 05 03 01 00 01 01 xx MCU 对 DP 消息接收进行应答

- DP ID: 03

- DP Type: 01
- DP Length: 00 01
- DP Value: 01

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x05
数据长度	2	0x0001
数据	1	0x00: 失败 0x01: 成功
校验和	1	xx

示例: 55 AA 02 xx xx 05 00 01 01 xx

### 6.4. DP 消息上报-可触发联动 (0x06)

当 MCU 检测到本地状态有变化, 例如用户手动按压了一个开关, 可通过此命令主动上报发生了变化的 DP, 以便将本地状态同步到云端。模组收到此命令后会发送给网关, 并且关心网关的响应, 根据网关响应情况应答 MCU。通过本命令上报 DP 数据, 能够触发联动, 包含本地联动和云端联动。

- 如果网关在等待时间内确认, 模组应答 MCU 上报成功。
- 如果网关未在等待时间内确认, 模组应答 MCU 上报失败。
- 如果模组未入网, 模组立即应答 MCU 上报失败。
- MCU 可根据模组回复, 判断模组与网关的链路是否通畅。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx

字段	字节数	说明
命令字	1	0x06
数据长度	2	N
数据	N	DP 消息格式
校验和	1	xx

示例： 55 AA 02 xx xx 06 00 05 03 01 00 01 01 xx

本例展示 MCU 主动上报一个 DP 数据，且该 DP 上报可用于触发联动。

- DP ID: 03
- DP Type: 01
- DP Length: 00 01
- DP Value: 01

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x06
数据长度	2	0x0001
数据	1	0x00: DP 上报失败 0x01: DP 上报成功
校验和	1	xx

示例： 55 AA 02 xx xx 06 00 01 01 xx DP 消息上报成功



- 上报成功：模组将 DP 消息发送给了网关，并收到了网关的确认。
- 上报失败：模组未加入网关、网关确认超时等。
- **数据** 部分最多只能包含一个 Raw 类型的 DP，且不能与其他类型的 DP 合并上报。

## 6.5. DP 消息上报-不触发联动 (0x2C)

本命令是 **DP 消息上报 (0x06)** 的拓展命令，目的是同步本地状态到云端，但是不希望直接触发联动。例如，开关和灯具配置了联动关系：当开关打开时，灯具打开，当开关关闭时，灯具关闭。那么当开关断电后再上电时，它的状态应该同步到云端，但是不希望去触发灯具的动作。为了将这种单纯的状态同步（不希望触发联动）与真实的状态变化后的主动上报（希望触发联动）区分开，特增加本命令。

### i

- 设备配网成功或设备上电后，建议通过本命令上报 DP 消息，以便同步状态。注意，为了避免大批量的设备同时上报造成网络拥堵，设备应该设定随机 5 ~15 秒延时上报。
- 您需要确认所采用的模组和固件是否支持本命令。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x2C
数据长度	2	N
数据	N	DP 消息格式
校验和	1	xx

示例： 55 AA 02 xx xx 2C 00 05 03 01 00 01 01 xx

本例展示 MCU 主动上报一个 DP 数据, 目的只是为了同步状态到云端，不用于触发联动。

- DP ID: 03
- DP Type: 01
- DP Length: 00 01
- DP Value: 01

### 模组应答

字段	字节数	说明

帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x2C
数据长度	2	0x0001
数据	1	0x00: DP 上报失败 0x01: DP 上报成功
校验和	1	xx

示例： 55 AA 02 xx xx 2C 00 01 01 xx DP 消息上报成功



- 上报成功：模组将 DP 消息发送给了网关，并收到了网关的确认。
- 上报失败：模组未加入网关、网关确认超时等。
- **数据** 部分最多只能包含一个 Raw 类型的 DP，且不能与其他类型的 DP 合并上报。

## 6.6. DP 查询 (0x28)

本指令用于网关主动查询设备的 DP，MCU 收到本命令后，先应答本命令，后上报对应的 DP。这种情况通常发生在网关断电后再上电、网关与某个子设备长时间断连后恢复等。

### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x28
数据长度	2	N
数据	N	DP List



字段	字节数	说明
校验和	1	xx

MCU 首先应答本命令，再通过命令字 0x2C 或 0x06 上报（建议通过 0x06 上报）。

**DP ID List 有两种情况：**

- DP ID List 为空，即数据长度字段为 0，表示网关要查询本设备的所有 DP，MCU 应该在应答本命令后，主动上报全部 DP。
- DP ID List 不为空，例如 0x01 0x02 表示网关要查询本设备的 DP ID 为 0x01 和 0x02 的两个 DP，MCU 应该在应答本命令后，主动上报 DP1 和 DP2。

示例 1： 55 AA 02 xx xx 28 00 00 xx 网关要查询本设备所有 DP。示例 2： 55 AA 02 xx xx 28 00 02 01 02 xx 网关要查询本设备 DP1 和 DP2。

**MCU 应答**

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x28
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例： 55 aa 02 xx xx 28 00 01 01 xx

## 6.7. 发送广播 DP 消息（0x27）

本命令用于发送全网通知的 DP 广播消息，可以让全网络中的设备接收到。如果存在低功耗设备，该设备需要处于周期唤醒的状态，且唤醒周期需要小于广播周期。否则，在唤醒前，下一条广播数据就会将之前的广播数据覆盖。



广播消息之间需要有一定的时间间隔，间隔由网络的规模决定，否则容易造成网络拥堵。

## MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x27
数据长度	2	N
数据	N	DP 消息格式
校验和	1	xx

示例： 55 AA 02 xx xx 27 08 05 02 00 04 00 00 00 1E xx 发送一条广播消息，消息内容如下：

- DP ID: 05
- DP Type: 02
- DP Length: 00 04
- DP Value: 00 00 00 1E

## 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x27
数据长度	2	0x0001
数据	1	0x00: 发送失败 0x01: 发送成功
校验和	1	xx

示例： 55 AA 02 xx xx 27 00 01 01 xx 表示广播消息发送成功。

## 6.8. 发送群组 DP 消息 (0x43)

以组播方式向群组发送 DP 消息，控制那些支持涂鸦 DP 的设备。这里发送的 DP 消息格式是被控设备的 DP 格式，包含 DPID + DP Type + DP Length + DP Value，该命令中 Group ID 是通过 [场景配置-场景控制器设备](#) 命令获取到的。



当前，只有场景控制器类设备支持此命令，且 MCU 需要清楚被控设备的 DP 格式和内容。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x43
数据长度	2	N
数据	N	DP 消息格式
校验和	1	xx

示例： 55 AA 02 xx xx 43 00 07 2A 08 01 01 00 01 01 xx 以组播方式发送一条 DP 消息，内容如下：

- Group ID: 0x2A08
- DP ID: 01
- DP Type: 01
- DP Length: 00 01
- DP Value: 01

### 模组响应

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02

字段	字节数	说明
序列号	2	xx xx
命令字	1	0x43
数据长度	2	0x0001
数据	1	0x00: 发送失败 0x01: 发送成功
校验和	1	xx

示例： 55 AA 02 xx xx 43 00 01 01 xx 表示 DP 组播发送成功。

## 6.9. 发送群组标准命令 (0x42)

设备以组播形式向群组中发送标准命令，控制那些支持标准命令的设备。这里的标准命令由 Zigbee 3.0 规范定义，通常为 Cluster ID + Command ID + Payload，该命令中的 Group ID 是通过 [场景配置-场景控制器设备](#) 命令获取到的。



当前，只有场景控制器类设备支持此命令，且 MCU 开发者需要了解 Zigbee 3.0 规范中的 Cluster + Command 格式。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x42
数据长度	2	N
数据	N	Group ID (2 字节) + 标准命令格式
校验和	1	xx

**标准命令格式：**

## Zigbee 3.0 Command

Cluster ID	Command ID	Payload
2 bytes	1 byte	N bytes

示例： 55 AA 02 xx xx 42 00 05 2A 08 00 06 01 xx 组播发送一条标准命令，命令内容如下：

- Group ID: 0x2A08
- Cluster ID: 0x0006 (onoff cluster)
- Command ID: 0x01 (on)
- Payload: null

**模组响应**

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x42
数据长度	2	0x0001
数据	N	0x00: 发送失败 0x01: 发送成功
校验和	1	xx

示例： 55 AA 02 xx xx 42 00 01 01 xx 组播标准命令发送成功。

目前支持的标准命令如下，每一个标准命令的 Payload 定义，请参考 Zigbee 官方文档 [Zigbee Cluster Library Specification R8](#)。本链接由 CSA 联盟提供，后面可能会更新。

**On/off cluster**

- On
- Off
- Toggle

**Level cluster**

- Move to level
- Move
- Step
- Stop
- Move to level (with on/off)
- Move (with on/off)
- Step (with on/off)
- Stop

**Color control cluster**

- Move to hue
- Move hue
- Step hue
- Move to saturation
- Move saturation
- Step saturation
- Move to hue and saturation
- Move to color
- Move color
- Step color
- Move to color temperature
- Move color temperature
- Step color temperature
- Stop move step

**Window covering cluster**

- Up/open
- Down/close
- Stop
- Go to lift value
- Go to lift percentage
- Go to tilt value
- Go to tilt percentage

**Scene cluster**

- Recall scene

## 7. 场景类命令

场景有时候也称作联动，即当满足某个条件时，执行某个动作。条件可以是时间、天气、某个设备的状态等，动作可以是对另一些设备执行控制命令。例如，温度传感器检测到温度超过 30°C 时，打开空调并设置目标温度为 25°C。

有一类设备称为场景控制器（场景开关、手持遥控等），它们的作用是充当场景中的条件设备。该类设备通常配备多个按键，每个按键都可以用来触发不同的场景。例如，按键 1 触发灯具全关，按键 2 触发灯具全开。

目前涂鸦 Zigbee 支持的场景有 **标准场景** 和 **联动场景**，不同的创建方式决定了创建出来的场景的类型。

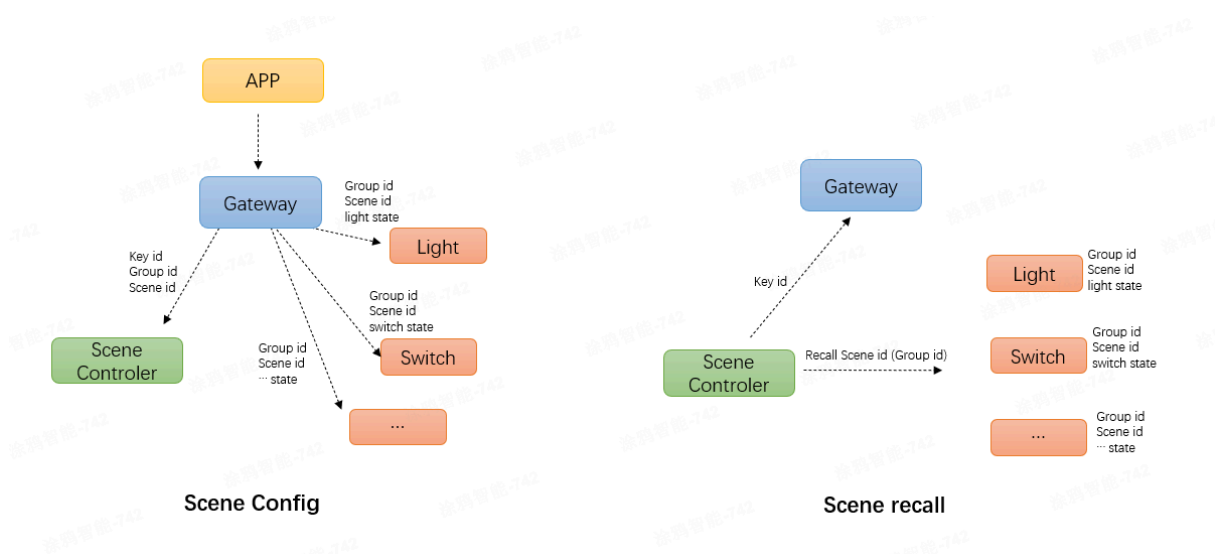
### 7.1. 标准场景

**标准场景**：即 CSA 联盟 ZCL SPEC 文档中定义的 Scene。标准场景只在创建时依赖网关（需要通过网关下发相关配置信息），而以后的场景触发则不依赖网关，即使网关断电也不影响标准场景触发，场景控制器是直接发送控制命令给执行设备的。

标准场景的创建需要满足以下条件：

- 创建入口：需要从场景控制器设备的 App 面板进入，选定某个按键后进行场景创建。
- 执行设备：执行设备是和场景控制器同一个网关下的 Zigbee 设备，可以添加多个执行设备，但是不能有群组。当添加群组作为执行设备时，会自动转成联动场景。

标准场景的创建与触发，如下所示：



- **标准场景配置 (Scene config)：** 在 App 上创建标准场景后，App 会通过网关向场景控制器发送按键号 (Key ID)、群组地址 (Group ID) 和场景号 (Scene ID)。相同的群组地址和

场景号也会发送给被控设备，同时会包含被控设备要执行的动作。此时，场景控制器和被控设备都被加入到了同一个群组中，以后场景控制器可以向该群组地址发送命令。

- **标准场景触发 (Scene recall)**：操作场景控制器按键，场景控制器向该按键绑定的群组地址发送场景唤醒命令，被控设备收到该消息后，进入该场景 ID 对应的状态。

## 7.2. 联动场景

**联动场景**：是指通过网关或云端作为中转，触发设备将状态报告给网关，网关检查该联动关系是否已托管在本地。如果已托管在本地，则可以称该场景为 **网关联动**，若未托管在本地，则称为 **云端联动**。

- **网关联动**：由 Zigbee 网关保存着联动关系，触发设备将状态报告给网关，网关根据本地保存的联动关系去控制执行设备。例如，上面举例的温度传感器和空调，当这两个设备都是同一个网关下的 Zigbee 设备时，通常采用网关联动。
- **云端联动**：由云端保存着联动关系，触发设备状态报告给网关后，网关报告给云端，云端根据联动关系去控制执行设备。例如，一个 Zigbee 设备作为触发条件，Wi-Fi 设备作为执行设备时，通常采用云端联动。

当创建的场景是联动场景时，开发者无需关心它是网关联动还是云端联动，这是由涂鸦后台根据用户创建时选择的不同设备类型决定的。



目前，场景类命令只开放给场景控制器类设备使用，且需要特定的模组固件，用户创建场景的方式不同，最终会生成不同的场景类型。

## 7.3. 场景配置-场景控制器设备 (0x41)



本命令需要特定模组固件才支持，用于场景控制器类设备。模组收到来自网关下发的按键号、群组地址和场景号，通过本命令通知 MCU。

### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02



字段	字节数	说明
序列号	2	xx xx
命令字	1	0x41
数据长度	2	0x0004
数据	4	Key ID (1 字节) + Group ID (2 字节) + Scene ID (1 字节)
校验和	1	xx

示例： 55 AA 02 xx xx 41 00 04 01 00 02 02 xx 表示收到来自网关的场景配置命令：

- Key ID： 01
- Group ID： 0x0002
- Scene ID： 0x02

#### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x41
数据长度	2	0x0001
数据	1	0x00：失败 0x01：成功
校验和	1	xx

示例： 55 AA 02 xx xx 41 00 01 01 xx

## 7.4. 场景触发-场景控制器设备 (0x0A)

触发标准场景和云端场景。云端场景是指在涂鸦 App 上配置的设备间联动（配置入口不在场景控制器设备的面板中），本设备作为联动触发设备，当检测到按键动作时，发送标准场景触

发命令（Recall scene），同时向网关上报按键号用于触发云端联动。若未配置标准场景，则本命令仅用于触发云端联动。



本命令需要特定模组固件才支持，用于场景控制器类设备。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0A
数据长度	2	0x0001
数据	1	按键 ID
校验和	1	xx

示例： 55 AA 02 xx xx 0A 00 01 01 xx 触发场景，Key ID 为 1 。

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x41
数据长度	2	0x0001
数据	1	0x00：失败 0x01：成功
校验和	1	xx

示例： 55 AA 02 xx xx 0A 00 01 01 xx 触发成功，云端联动和标准场景，任意一个只要模组发送成功即认为是触发成功。

## 8. 产测类命令

### 8.1. 模组 RF 测试 (0x08)

该命令用于测试模组的 RF 数据发送和接收是否正常，在进行 RF 测试时需要搭配一个产测 Dongle，并将产测 Dongle 配置为 RF 模式，可参考 [设置 Zigbee Dongle](#)。模组在收到此命令后，会向空中发送裸数据，通常是 100 包，Dongle 每次收到一包这样的裸数据会回环式返回。模组同时开启接收模式，对接收到的数据进行检查。如果相同，则认为有效数据，并返回给 MCU 收到的数据包个数。



模组 RF 测试需要在未配网状态下进行，且测试完成之后，需要重启模组才能再次正常运行模组程序。默认信道为 11 信道。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x08
数据长度	2	0x0001
数据	1	信道号 (11 ~ 26)
校验和	1	xx

信道号要与产测 Dongle 配置的信道一致，请勿在同一个信道中同时对多台设备进行 RF 测试，避免互相干扰。

示例：55 AA 02 xx xx 08 00 01 0B xx 在 11 信道进行 RF 测试。

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02

字段	字节数	说明
序列号	2	xx xx
命令字	1	0x08
数据长度	2	0x0001
数据	2	RF 测试结果
校验和	1	xx

RF 测试结果：

- Data[0] = 0x00，表示测试失败。
- Data[1] = 0x00，失败原因是未接收到回环的数据包。
- Data[0] = 0x01，表示测试成功，此时
- Data[1] = 回环数据包个数（1 ~ 100），100 表示所有发出去的数据都收到了来自产测 Dongle 的回环。

示例：55 AA 02 xx xx 08 00 01 01 62 xx 表示 RF 测试成功，发送了 100 个数据包，收到了 98 个数据包。

进入 RF 测试后，模组发送特定格式的裸数据包，每包间隔 20 ms，共发 100 个包。产测 Dongle 收到之后会回环发出，此时模组会收到与自身发出去的相同的数据包。以此来测试模组的发送和接收能力，暂未对信号强度进行检测。

## 8.2. 信标产测通知 (0x29)

用于成品产测，需要搭配产测 Dongle，且要将 Dongle 配置为 Beacon 模式，即 BC 模式，此时 Dongle 会持续向空中发送特定数据包，被称为信标。模组上电时会有短时间的信标接收窗口（一般是 300 ms）。当模组接收到来自 Dongle 的信标时，会通过本命令通知 MCU，MCU 收到后可以做一些效果或自检流程。例如灯具，无有效手段开启产测，可通过此方式进入产测流程，以检查灯具上的各灯珠是否正常发光。



- 信标产测，只在设备未配网时是有效的。如果设备已配网，则模组上电后不会有信标接收窗口，且如果配网超过 15 分钟，则永久关闭信标产测。
- 模组会依次以 9600 bps 和 115200 bps 波特率发送信标产测通知。

模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x29
数据长度	2	0x0001
数据	1	0x00
校验和	1	xx

示例： 55 AA 02 xx xx 29 00 01 00 xx 表示模组在上电信标接收窗口期，收到了来自产测 Dongle 的信标。

#### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x29
数据长度	2	0x0001
数据	1	0x00：失败 0x01：成功
校验和	1	xx

示例： 55 AA 02 xx xx 29 00 01 01 xx

注意，模组不会对 MCU 返回的成功或失败进行检查。

## 9. MCU OTA 类命令

支持通过模组进行 MCU OTA 升级，通常的流程如下：

1. 网关通过 [MCU 固件版本查询](#) 命令获取到 MCU 当前的固件版本号，并上报给涂鸦云端。
2. 涂鸦云端检测到该 PID (或该设备 ID) 配置了 MCU OTA 升级，且设备当前的 MCU 固件版本号符合升级条件，则通知设备进行 MCU OTA。
3. 此通知到达模组后，模组会以 [OTA 升级通知](#) 命令通知 MCU。
4. MCU 可以开始进行 [OTA 文件请求](#)，待整个文件请求完毕后，开始内部的升级操作。
5. 升级完成后，[上报 OTA 结果](#)，并上报新的 MCU 固件版本号，即 [MCU 固件版本查询](#) 中的 MCU 应答帧格式。

### 9.1. MCU 固件版本查询 (0x0B)

同步 MCU 当前的固件版本号给涂鸦云端，用于展示或者 OTA 升级的判断。若产品需要 MCU OTA 升级，则必须支持本命令。该命令实际有两个作用：

- 网关查询设备 MCU 版本号：模组收到来自网关的查询命令后通知给 MCU，MCU 需要进行应答。
- 设备主动上报 MCU 版本号：例如配网成功、OTA 成功，MCU 主动发送应答帧格式给模组，模组收到后会上报给网关。

#### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0B
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例： 55 AA 02 xx xx 0B 00 00 xx 查询 MCU 的当前版本号。

#### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0B
数据长度	2	0x0001
数据	1	MCU 固件版本号（1 字节）
校验和	1	xx

MCU 固件版本号，1 字节，8 个 Bits 按二进制 `xx.yy.zzzz` 格式，例如 `0x40` 二进制为 `01 00 0000`，则表示版本号为 1.0.0，由于字节限制，最大版本号为 3.3.15。

示例：55 AA 02 xx xx 0B 00 01 53 xx 表示当前版本号为 `0x53`，二进制为 `0101 0011`，即 1.1.3。



- 该命令通常是用于涂鸦云端获取设备的 MCU 版本号信息，用作是否推送 OTA 的判断。
- MCU 可以在未收到模组查询命令的情况下，主动发送 MCU 应答帧，模组收到后也会上报给网关，从而实现设备主动上报 MCU 版本号。

## 9.2. OTA 升级通知 (0x0C)

在涂鸦开发者平台配置了设备 MCU 升级，并开启了推送后，如果设备满足升级条件，则会在客户端 App 上进行提示，或者无提示自动升级。该命令由网关发送给模组，模组收到后通知 MCU 准备 OTA 升级。

### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx

字段	字节数	说明
命令字	1	0x0C
数据长度	2	0x0011
数据	17	PID（8 字节）+ 新固件版本号（1 字节）+ 固件大小 bytes（4 字节）+ 固件校验和（4 字节）
校验和	1	xx

- PID（8 字节）：与当前设备相同的 PID，例如 AIp08kLI。
- 新固件版本号（1 字节）：例如 0x41，表示 1.0.1。
- 固件大小（4 字节）：单位为字节数。
- 固件校验和（4 字节）：从固件第一个字节开始对整个固件的校验和。MCU 在接收完整个升级固件后，应该计算校验和并与本字段进行对比。如果不一致，说明传输过程中可能有问题，可以报升级失败。

示例：0x55 AA 02 xx xx 0C 00 11 41 49 70 31 38 6b 4c 49 41 00 00 78 00 30 31 32 33

xx OTA 升级通知，内容为：

- PID：0x41 49 70 31 38 6b 4c 49，即 AIp08kLI
- 新固件版本号：0x41，表示 1.0.1
- 固件大小：0x00007800，即 30 KB
- 固件校验和：0x30313233

#### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0C
数据长度	2	0x0001
数据	1	0x00 默认值



字段	字节数	说明
校验和	1	xx

示例： 55 AA 02 xx xx 0C 00 01 00 xx

目前，模组端没有处理 MCU 对 0x0C 命令的应答帧。如果 MCU 检查到数据有误，可以重新发起该数据包的请求。

### 9.3. OTA 文件请求 (0x0D)

设备在收到来自网关的 [OTA 升级通知](#) 后，可以开始下载 OTA 文件。下载过程是由设备端控制的，即设备发送文件数据请求命令，包含数据偏移量和数据包大小。网关收到后根据设备端的请求信息，将 OTA 文件中对应位置和长度的数据包返回给设备。该命令由 MCU 端发起，模组收到后发送给网关。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0D
数据长度	2	0x000E
数据	14	PID (8 字节) + 新固件版本号 (1 字节) + 数据包偏移量 (4 字节) + 数据包大小 (1 字节)
校验和	1	xx

- PID (8 字节)：与当前设备相同的 PID，例如 AIP08kLI。
- 新固件版本号 (1 字节)：例如 0x41，表示 1.0.1。
- 数据包偏移量 (4 字节)：单位字节数，即本次请求的数据，在 OTA 文件中的偏移量。
- 数据包大小 (1 字节)：单位字节数，即本次请求的数据包的大小，最大不能超过 48 字节。

示例： 0x55 AA 02 xx xx 0C 00 0E 41 49 70 31 38 6b 4c 49 41 00 00 10 00 30 XX 表示请求 4096 位置开始的 48 字节数据包：

- PID： 0x41 49 70 31 38 6b 4c 49 ，即 AIp08kLI 。
- 新固件版本号： 0x41 ，表示 1.0.1。
- 数据包偏移量： 0x00001000 ，即偏移量为 4096 位置。
- 数据包大小： 0x30 ，即请求 48 字节的数据。

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0D
数据长度	2	N
数据	N	结果（1 字节）+ PID（8 字节）+ 新固件版本号（1 字节）+ 数据包偏移量（4 字节）+ 数据包内容
校验和	1	xx

- 结果： 0x00 请求成功， 0x01 请求失败，失败状态时后面的数据不存在。
- PID（8 字节）：与当前设备相同的 PID，例如 AIp08kLI 。
- 新固件版本号（1 字节）：例如 0x41 ，表示 1.0.1。
- 数据包偏移量（4 字节）：单位字节，即本次请求的数据，在 OTA 文件中的偏移量。
- 数据包内容（N 字节）：本次返回的数据包的内容。

示例： 55 AA 02 xx xx 0D xxxx 00 41 49 70 31 38 6b 4c 49 41 00 00 00 32 ... xx 表示请求成功 (数据部分用省略号表示)。

## 9.4. OTA 结果上报 (0x0E)

设备在进行 MCU OTA 升级后，将 OTA 升级结果上报给网关，该命令由 MCU 发起。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
字段	字节数	说明
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0E
数据长度	2	0x000A
数据	10	升级结果（1 字节）+ PID （8 字节）+ 新固件版本号 （1 字节）
校验和	1	xx

- 升级结果（1 字节）： 0x00 升级成功， 0x01 升级失败。

示例： 0x55 AA 02 xx xx 0E 00 0A 00 41 49 70 31 38 6b 4c 49 41 26 表示设备 MCU OTA 成功

- MCU OTA 结果： 0x00 成功
- PID： 0x41 49 70 31 38 6b 4c 49 ，即 AIp08kLI
- 新固件版本号： 0x41 ，表示 1.0.1

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x0E
数据长度	2	0x0001
数据	1	0x00: OK0x01: Error
校验和	1	xx

示例： 0x55 AA 02 xx xx 0E 00 01 00 xx

## 10. GPIO 控制类命令

模组通过 UART TX 和 UART RX 与 MCU 通信（低功耗类还有两个唤醒管脚）。除了这几个管脚外，模组还有其他的 GPIO。当 MCU 的 GPIO 不足时，可以使用模组的 GPIO 来作为输入和输出。要使用模组的 GPIO 先要对 GPIO 进行配置，使用 [GPIO 配置](#) 对模组的管脚进行初始化。

- 如果需要模组 GPIO 作为输出，则通过 [GPIO 输出控制](#) 命令指定要输出的电平。
- 如果模组 GPIO 被配置为输入，则可以通过 [GPIO 电平读取](#) 命令获取模组当前 GPIO 的状态。
- 如果模组 GPIO 被配置为输入中断，则当模组检测到 GPIO 的电平变化时，会通过 [GPIO 输入中断通知](#) 命令通知 MCU。



GPIO 是否能作为中断输入以及 GPIO 作为输出时的驱动能力，不同芯片平台会不一样，需要结合芯片平台的硬件手册。

通过 Port Number 和 Pin Number 指定模组的具体 GPIO，通过 GPIO Mode 指定模组 GPIO 的输入输出模式，具体约定如下：

- **Port Number**: 0 ~ 9 表示 PORT\_A ~ PORT\_K，即 Port Number = 0 表示指定的 GPIO Port 为 PORT\_A，以此类推。
- **Pin Number**: 0 ~ 15 表示 PIN\_0 ~ PIN\_15，即 Pin Number = 0 表示指定的 GPIO PIN 为 PIN\_0，以此类推。
- **GPIO Mode**: 不同芯片平台，支持的 GPIO 模式不一样，具体需要参考芯片手册，否则可能会导致配置失败。

GPIO Mode	描述
0	GPIO_MODE_INPUT_HIGH_IMPEDANCE
1	GPIO_MODE_INPUT_PULL
2	GPIO_MODE_OUTPUT_PP
3	GPIO_MODE_OUTPUT_OD
4	GPIO_MODE_OUTPUT_OD_PULL_UP
5	GPIO_MODE_OUTPUT_OD_PULL_DOWN

- **GPIO Level**: 当 GPIO 作为输入中断时，用于指定触发中断的电平。当 GPIO 作为输出时，用于指定输出电平。

GPIO Level	描述
0	GPIO_LEVEL_LOW

GPIO Level	描述
1	GPIO_LEVEL_HIGH
2	GPIO_LEVEL_ALL

### 10.1. GPIO 配置 (0x36)

用于配置模组的 GPIO 模式，MCU 可以一次性配置多个模组 GPIO。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x36
数据长度	2	N
数据	N	GPIO 个数 (1 字节) + GPIO 配置列表
校验和	1	xx

- GPIO 个数 (1 字节)：要配置的 GPIO 个数。
- GPIO 配置列表：每个 GPIO 占用 4 字节，具体格式如下图所示：

GPIO CONFIG LIST								
GPIO1				GPIO2				...
PORT NUMBER	PIN NUMBER	GPIO MODE	GPIO LEVEL	PORT NUMBER	PIN NUMBER	GPIO MODE	GPIO LEVEL	...
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	...

示例： 55 AA 02 xx xx 36 00 05 01 00 00 02 01 xx 表示要配置 1 个 GPIO，具体为 PA0，GPIO\_MODE\_OUTPUT\_PP，输出高电平。

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA

字段	字节数	说明
版本	1	0x02
序列号	2	xx xx
命令字	1	0x36
数据长度	2	N
数据	N	GPIO 个数（1 字节） + GPIO 配置结果
校验和	1	xx

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x36
数据长度	2	N
数据	N	GPIO 个数（1 字节） + GPIO 配置结果
校验和	1	xx

GPIO CONFIG RESPONSE

GPIO1			GPIO2			...
PORT NUMBER	PIN NUMBER	STATUS	PORT NUMBER	PIN NUMBER	STATUS	...
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	...

- STATUS : 0x00 表示配置失败, 0x01 表示配置成功。

示例: 55 AA 02 xx xx 36 00 05 01 00 00 02 01 xx

## 10.2. GPIO 电平读取 (0x37)

通过本命令获取模组指定 GPIO 当前的电平值。使用此命令之前，需要先通过 [GPIO 配置](#) 完成模组 GPIO 的初始化。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x37
数据长度	2	N
数据	N	GPIO 个数 (1 字节) + GPIO 读取列表
校验和	1	xx

- GPIO 个数 (1 字节)：要读取的 GPIO 个数。
- GPIO 读取列表：每个 GPIO 占用 2 字节，具体格式如下图所示：

GPIO READ LIST				
GPIO1		GPIO2		...
PORT NUMBER	PIN NUMBER	PORT NUMBER	PIN NUMBER	...
1 byte	1 byte	1 byte	1 byte	...

示例： 55 AA 02 xx xx 37 00 05 02 00 00 01 01 xx 表示要读取 PA0 和 PB1 的电平。

### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x37



字段	字节数	说明
数据长度	2	N
数据	N	GPIO 个数（1 字节） + GPIO 电平列表
校验和	1	xx

- GPIO 个数（1 字节）：要读取的 GPIO 个数。
- GPIO 电平列表：每个 GPIO 占用 3 字节，具体格式如下图所示：

GPIO LEVEL LIST						
GPIO1			GPIO2			...
PORT NUMBER	PIN NUMBER	GPIO LEVEL	PORT NUMBER	PIN NUMBER	GPIO LEVEL	...
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	...

示例：55 AA 02 xx xx 37 00 07 02 00 00 01 01 01 00 xx 表示当前 PA0 是高电平，PB1 是低电平。

### 10.3. GPIO 输出控制（0x38）

当模组的 GPIO 配置为输出时，通过此命令控制输出高电平或低电平。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x38
数据长度	2	N
数据	N	GPIO 个数（1 字节） + GPIO 输出控制列表
校验和	1	xx

- GPIO 个数（1 字节）：要控制输出的 GPIO 个数。

- GPIO 输出控制列表：每个 GPIO 占用 3 字节，具体格式如下图所示：

GPIO OUTPUT CONCTRL LIST						
GPIO1			GPIO2			...
PORT NUMBER	PIN NUMBER	GPIO LEVEL	PORT NUMBER	PIN NUMBER	GPIO LEVEL	...
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	...

示例：55 AA 02 xx xx 38 00 07 02 00 00 01 01 01 00 xx 表示要控制模组的 PA0 输出高电平，PB1 输出低电平。

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x38
数据长度	2	N
数据	N	GPIO 个数（1 字节） + GPIO 输出控制响应
校验和	1	xx

- GPIO 个数（1 字节）：要读取的 GPIO 个数。
- GPIO 输出控制响应：每个 GPIO 占用 3 字节，具体格式如下图所示：

GPIO OUTPUT CONCTRL RESPONSE						
GPIO1			GPIO2			...
PORT NUMBER	PIN NUMBER	STATUS	PORT NUMBER	PIN NUMBER	STATUS	...
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	...

- STATUS：0x00 表示成功，0x01 表示失败。

示例：55 AA 02 xx xx 38 00 07 02 00 00 00 01 01 00 xx 表示模组 PA0 输出高电平成功，PB1 输出低电平成功。

## 10.4. GPIO 输入中断通知 (0x39)

当模组的 GPIO 被配置为输入中断，且检测到 GPIO 中断已经触发了，则通过本命令通知 MCU。

### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x39
数据长度	2	0x0003
数据	3	Port Number (1 字节) + Pin Number (1 字节) + GPIO Level (1 字节)
校验和	1	xx

示例： 55 AA 02 xx xx 39 00 03 00 00 01 xx 表示检测到 PA0 中断，产生中断的电平是高电平。

### MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x39
数据长度	2	0x0000
数据	0	无
校验和	1	xx

模组发送完 GPIO 中断通知命令后，不会判断 MCU 是否应答。

## 11. 时间和天气

### 11.1. 时间同步 (0x24)

部分设备关心当前时间信息，例如用于本地定时或者显示屏显示。设备可以向网关请求时间信息，在网关应答时间信息时已经完成了时区转换。该命令由 MCU 发起，模组收到该命令后向网关发起时间请求，在收到网关回复后通过本命令的应答帧，将时间信息返回给 MCU。

#### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x24
数据长度	2	0x0000
数据	0	无
校验和	1	xx

示例： 55 AA 02 xx xx 24 00 00 xx MCU 发起，请求时间同步。

#### 模组应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x24
数据长度	2	0x0008
数据	8	标准时间戳（4 字节）+ 本地时间戳（4 字节）
校验和	1	xx

- 标准时间戳：格林威治时间，以 1970.01.01 00:00:00 为起点到现在的总秒数。
- 本地时间戳：考虑时区和夏令时的本地时间，即标准时间 ± 时差。一般设备以本地时间为准。

示例： 55 AA 02 xx xx 24 00 08 66 45 DB F0 66 46 4C 70 xx

- 标准时间： 0x6645DBF0
- 本地时间： 0x66464C70

## 11.2. 天气信息查询 (0x3A)

部分设备关心天气数据，例如用于展示在显示屏上，设备可以向网关请求天气信息，网关从互联网获取到天气信息后回复给设备。天气获取由 MCU 端发起，模组收到本命令后，会向网关发送天气信息请求。待网关回复后，通过命令 [天气信息通知](#) 通知 MCU。

按照时间可以分为两类：

- 实时天气：即从云端获取到的当前时刻的天气，例如温度、湿度、风向和风力等。
- 预报天气：包括今天在内的未来几天的天气，当前最多可以获取到近 7 天的天气数据。

按地址可以分为两类：

- 网关激活地址天气：即天气信息是以网关的激活地址为准的。如果网关激活地址和当前网关所在地址没有变化，那么也可以理解为本身设备的所在地。
- App 家庭地址天气：即天气信息是以用户在涂鸦 App 上设置的家庭地址为准。

天气数据内容：

- 天气数据：支持多种天气数据，以天气 ID 表示，例如 01：温度，02：湿度，03：天气概况。详见 [附表 1 天气数据](#)。
- 城市名称：可查询所在城市名称字符串，主要用于显示屏显示。



- 目前，天气命令只在部分模组固件上实现，更多的模组会陆续支持。在使用之前，请确认所使用模组是否支持本命令。
- 天气数据和城市名称，需分别请求，不支持同时请求。

### MCU 发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02

字段	字节数	说明
序列号	2	XX XX
命令字	1	0x3A
数据长度	2	N
数据	N	天气请求格式 / 城市名称请求格式
校验和	1	XX

## 天气请求格式

字段	字节数	说明
版本号	1	0x11 固定值
地址类型	1	0x00: 设备地址（以网关激活地址为准）0x01: App 设置的家庭地址
天气 ID	N	表示要请求哪些天气数据，每个天气数据有唯一 ID（1 字节）最少请求 1 个 ID
天气预报标志位	1	0x12 固定值，同时也作为天气 ID 的结束标志。
天气预报天数	1	1 ~ 7，最多可查询近 7 天的预报，含当天
实时天气标志位	1	0x13 固定值
是否需要实时天气	1	0x00: 不需要实时天气 0x01: 需要实时天气

- 天气预报标志位和天气预报天数：非必填字段。如果不需要天气预报，则不应该包含这两个字段。
- 实时天气标志位和是否需要实时天气：为非必填字段。如果不需要实时天气，则不应该包含这两个字段。

- 天气预报和实时天气，二者至少需要包含一个，否则网关不会返回任何数据。

#### 城市名称请求格式

字段	字节数	说明
版本号	1	0x11 固定值
地址类型	1	0x00: 设备地址（以网关激活地址为准）0x01: App 设置的家庭地址
区县请求标志	6	0x63 0x2E 0x61 0x72 0x65 0x61 对应字符串: c.area
城市请求标志	6	0x63 0x2E 0x63 0x69 0x74 0x79 对应字符串: c.city

示例 1: 55 AA 02 xx xx 3A 00 09 11 00 01 02 03 12 00 13 01 xx

- 11 : 版本号
- 00 : 取设备地址（网关）
- 01 02 03 : 请求的天气数据 ID, 01 温度, 02 湿度, 03 天气概况
- 12 : 天气预报标志
- 00 : 不需要天气预报
- 13 : 是否需要实时天气标志
- 01 : 需要实时天气

表示请求设备地址的实时天气数据，包含温度、湿度、天气概况，不需要天气预报。

示例 2: 55 AA 02 xx xx 3A 00 09 11 00 01 02 03 12 07 13 00 xx

- 11 : 版本号
- 00 : 取设备地址（网关）
- 01 02 03 : 请求的天气数据 ID, 01 温度, 02 湿度, 03 天气概况.
- 12 : 天气预报标志
- 07 : 请求近 7 天的天气预报（含今天）
- 13 : 是否需要实时天气标志
- 00 : 不需要实时天气

表示请求设备地址的近 7 天天气预报，包含温度、湿度和天气概况，不需要实时天气。

示例 3: 55 AA 02 xx xx 3A 00 0D 11 00 63 2E 61 72 65 61 63 2E 63 69 74 xx 请求设备地址的城市名称和区县名称。

- 11：版本号
- 00：取设备地址（网关）
- 63 2E 61 72 65 61：请求区县名称
- 63 2E 63 69 74 79：请求城市名称

#### 模组返回

- 若模组未入网或网关返回超时，则返回失败。
- 若模组已入网，且网关进行了天气回复，则以 [天气信息通知](#) 返回给 MCU。

### 11.3. 天气信息通知（0x3B）

模组收到了来自网关的天气信息，通过本命令将天气信息给到 MCU。网关的天气信息取自互联网。

#### 模组发送

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x3B
数据长度	2	N
数据	N	天气数据格式 / 城市区域名称格式
校验和	1	xx

#### 天气数据格式

字段	字节数	说明
版本号	1	0x11 固定值
地址类型	1	0x00：设备地址（以网关激活地址为准）0x01：App 设置的家庭地址
天气预报标志位	1	0x12 固定值



字段	字节数	说明
天气预报天数	1	0x00: 不包含天气预报 n (0x01 ~ 0x07) 包含天气预报的天数
实时天气标志位	1	0x13 固定值
包含实时	1	0x00: 不包含实时数据 0x01: 包含实时数据
天气信息	N	天气参数 1 + 天气参数 2 + ...

**天气参数** 的格式为：天气 ID + 实时值 + 今天预报 + 明天预报 + 后天预报 + ……（取决于预报天数）

- 天气 ID：参考 [附表 1 天气数据](#)。
- 实时值：若 **是否包含实时数据** = 1，则有此字段，字段长度与天气 ID 有关。
- 预报值：预报天数 ≠ 0 时，则有此字段，根据预报天数决定包含几组数据。

举例：

0x11	0x00	0x12	0x03	0x13	0x01	0x01	0x00
版本号	地址类型	预报标志	预报天数(1~7)	实时标志	包含实时	温度ID	实时温
0x08	0x00	0x09	0xFF	0xFF	0xFF	0xFF	0x02
度 8℃	今天温度 8℃		明天温度(未获取到)		后天温度(未获取到)		湿度ID
0x00	0x45	0x00	0x3E	0xFF	0xFF	0xFF	0xFF
实时湿度 69		今天湿度 62		明天湿度(未获取到)		后天湿度(未获取到)	
0x03	0x8C	0x8C	0x8E	0x8E	5-14℃		
天气概况ID	实时 "140" 霾	今天 "140" 霾	明天 "142" 多云	后天 "140" 多云			

本例中只展示了 **天气数据格式** 部分，其中：

- 地址类型： 0x00 ，表示以设备所在地址为准。
- 预报天数： 0x03 ，表示数据中包含了 3 天的预报数据。
- 包含实时： 0x01 ，表示数据中包含了实时数据。
- 因此，解析时按每个天气包含 4 组数据，实时数据在前，预报数据按今天，明天，后天……的顺序排列。

若预报天数为 0 ，则字段中不包含预报数据。预报数据和实时数据，必须至少有一项，不能出现不含实时数据并且预报天数又为 0 的情况。

**城市区域名称格式**

字段	字节数	说明
版本号	1	0x11 固定值
地址类型	1	0x00: 设备地址（以网关 激活地址为准）0x01: App 设置的家庭地址
区县标志	6	0x63 0x2E 0x61 0x72 0x65 0x61 对应字符串: c.area
区县名称	N	不定长度，最长 24 字节

字段	字节数	说明
版本号	1	0x11 固定值
地址类型	1	0x00: 设备地址（以网关 激活地址为准）0x01: App 设置的家庭地址
区县标志	6	0x63 0x2E 0x61 0x72 0x65 0x61 对应字符串: c.area
区县名称	N	不定长度，最长 24 字节
城市标志	6	0x63 0x2E 0x63 0x69 0x74 0x79 对应字符串: c.city

字段	字节数	说明
版本号	1	0x11 固定值
地址类型	1	0x00: 设备地址（以网关 激活地址为准）0x01: App 设置的家庭地址
区县标志	6	0x63 0x2E 0x61 0x72 0x65 0x61 对应字符串: c.area

字段	字节数	说明
区县名称	N	不定长度，最长 24 字节
城市标志	6	0x63 0x2E 0x63 0x69 0x74 0x79 对应字符串: c.city
城市名称	N	不定长度，最长 24 字节

字段	字节数	说明
版本号	1	0x11 固定值
地址类型	1	0x00: 设备地址（以网关激活地址为准）0x01: App 设置的家庭地址
区县标志	6	0x63 0x2E 0x61 0x72 0x65 0x61 对应字符串: c.area
区县名称	N	不定长度，最长 24 字节
城市标志	6	0x63 0x2E 0x63 0x69 0x74 0x79 对应字符串: c.city
城市名称	N	不定长度，最长 24 字节

举例： 0x11 01 63 2E 61 72 65 61 E8 A5 BF E6 B9 96 E5 8C BA 63 2E 63 69 74 79 E6 9D AD E5 B7 9E

- 11：版本号，固定值
- 01：地址类型 01，表示后续数据对应的地址为 App 设置的家庭地址
- 63 2E 61 72 65 61：字符串 c.area 表示后面接区县名称
- E8 A5 BF E6 B9 96 E5 8C BA：西湖区
- 63 2E 63 69 74 79：字符串 c.city 表示后面接城市名称
- E6 9D AD E5 B7 9E：杭州

### MCU 应答

模组不关心 MCU 的应答帧。

## 12. 附表

### 12.1. 附表 1：天气数据

天气 ID	数据段	含义	类型	描述
1	Temperature	大气温度	INT16	支持 7 天内数据，中国大陆地区不支持
2	Humidity	空气湿度	UINT16	支持 7 天内数据
3	ConditionNum	天气概况编码	UINT8	支持 7 天内数据
4	RealFeel	体感温度	UINT16	不支持 7 天内数据
5	uvi	紫外线指数	UINT8	支持 7 天内数据
6	windSpeed	风速	UINT8	支持 7 天内数据，此值是实际风速 x10 的结果 使用时需要除以 10
7	windDir	风向	UINT8	支持 7 天内数据
8	windLevel	风级	UINT8	支持 7 天内数据
9	aqi	空气质量指数	UINT16	仅有当天数据
10	Pm10	可吸入颗粒物 PM10	UINT16	仅有当天数据
11	Pm25	细颗粒物 PM2.5	UINT16	仅有当天数据
12	O3	臭氧浓度	UINT16	仅有当天数据
13	No2	二氧化氮浓度	UINT16	仅有当天数据
14	Co	一氧化碳浓度	UINT16	仅有当天数据
15	So2	二氧化硫浓度	UINT16	仅有当天数据
16	T-high	最高温度	INT16	仅有预报数据
17	T-low	最低温度	INT16	仅有预报数据
18	Data	预报天数	UINT8	$1 \leq n \leq 7$ ，当且仅当支持预报的数据才能添加此字段
19	Currdate	实时天气获取	Bool	1：带上实时数据。0：不带上实时数据

## 12.2. 附表 2：中文天气 UTF-8 编码

天气 ID	编码	描述
0x78	120	晴
0x65	101	大雨
0x66	102	雷暴
0x67	103	沙尘暴
0x68	104	小雪
0x69	105	雪
0x6a	106	冻雾
0x6b	107	暴雨
0x6c	108	局部阵雨
0x6d	109	浮尘

## 12.3. 附表 3：风向简码编码表

风向 ID	风向简码（字符串）	描述
0x00	N	北
0x01	NNE	东北偏北
0x02	NE	东北
0x03	ENE	东北偏东
0x04	E	东
0x05	ESE	东南偏东
0x06	SE	东南
0x07	SSE	东南偏南
0x08	S	南
0x09	SSW	西南偏南
0x0a	SW	西南
0x0b	WSW	西南偏西
0x0c	W	西
0x0d	WNW	西北偏西
0x0e	NW	西北
0x0f	NNW	西北偏北

## 13. 更新记录

主要变更	修订日期	修订说明
优化文档	20240725	1. 增加 0x0C 命令 MCU 应答帧描述
优化文档	20240717	1. 修正 0x0D 命令的最大请求数据包长度 2. 修正几处笔误
优化文档	20240708	1. 优化文档排版，优化各命令的描述
增加命令	20240521	1. 增加联动过滤 0x2C 命令字支持 2. 增加 0x36 ， 0x37 ， 0x38 ， 0x39 GPIO 配置命令字支持 3. 增加 0x3A ， 0x3B 天气获取命令字支持
优化文档	20221114	1. 优化协议排版，优化内容描述
增加命令	20220902	1. 增加 MCU 查询网关联网状态
增加命令	20220817	1. 增加 0x2A 、 0x2B 、 0x41 、 0x42 、 0x43 命令字及其功能 2. 强电版本增加发射功率可配 3. 产品信息新增可选 g 键值及其功能介绍
优化文档	20220322	1. 修改波特率自适应部分描述
增加命令	20211214	1. 增加 App，恢复出厂设置通知，MCU 配置 Zigbee 网络参数，MCU 广播数据上报，读取 DP 数据的协议

主要变更	修订日期	修订说明
优化文档	20201118	1. 明确 MCU 唤醒模组和模组唤醒 MCU 的方式为电平或脉冲，兼容 2. 明确网络参数配置中 Rejoin 的时间是长期 Rejoin 失败的时间
增加命令	20200801	1. 增加模组查询 MCU 类型帧 2. 增加 MCU 设置模组网络参数帧
优化协议	20200715	1. seq 的范围 2. 数据长度修改为 62 3. RF 测试时，当传入的信道非法时，默认 11 信道 4. 增加协议描述
时序调整	20200204	1. 模组增加配网状态的状态发送。说明：现阶段，协议没有对配网状态做出指示，客户的配网指示灯的逻辑没有把控。
修订文档	20181029	1. 添加可靠性传输 2. 添加休眠设备唤醒机制
修订文档	20180731	1. 添加场景开关协议
修订文档	20180702	1. 添加被动上报以及超时重传机制 2. 增加硬件握手 IO
修订文档	20180628	1. 修改超时时间 2. 修改被动上报方式 3. 修改模组配置方式
创建文档	20180623	创建文档