



串口协议

文档版本: 20250930

[查看在线版本](#)

目录

1. 硬件连接	2
2. 低功耗唤醒	4
2.1. 电平唤醒	4
2.2. 脉冲唤醒	4
3. 波特率自适应	6
4. 串口帧格式说明	7
4.1. 命令字索引 (Command ID)	8
5. 配置类命令	11
5.1. 查询产品信息 (0x01)	11
5.2. 模组网络状态通知 (0x02)	12
5.3. 模组配网/复位 (0x03)	14
5.4. 解绑并清除数据通知 (0x00)	15
5.5. 配置唤醒等待时间 (0x2B)	16
5.6. 查询模组信息 (0x07)	17
5.7. 查询模组网络状态 (0x20)	19
5.8. 查询网关联网状态 (0x25)	20
5.9. 配置 Zigbee 网络参数 (0x26)	21
6. 应用数据收发类命令	28
6.1. DP 消息接收 (0x04)	28
6.2. DP 消息接收-群控 (0x2A)	30
6.3. DP 消息应答 (0x05)	32
6.4. DP 消息上报-可触发联动 (0x06)	33
6.5. DP 消息上报-不触发联动 (0x2C)	35
6.6. DP 查询 (0x28)	36
6.7. 发送广播 DP 消息 (0x27)	38
6.8. 发送群组 DP 消息 (0x43)	39
6.9. 发送群组标准命令 (0x42)	41
7. 场景类命令	44
7.1. 标准场景	44
7.2. 联动场景	45
7.3. 场景配置 (0x41)	45
7.4. 场景触发 (0x0A)	47
8. 产测类命令	49
8.1. RF 产测 (0x08)	49
8.2. 信标产测通知 (0x29)	50
8.3. Dongle 产测通知 (0x21)	52
8.4. Dongle 产测数据上报 (0x22)	53

9. MCU OTA 类命令	57
9.1. MCU 固件版本查询 (0x0B)	57
9.2. OTA 升级通知 (0x0C)	58
9.3. OTA 文件请求 (0x0D)	60
9.4. OTA 结果上报 (0x0E)	62
10. GPIO 控制类命令	65
10.1. GPIO 配置 (0x36)	66
10.2. GPIO 电平读取 (0x37)	68
10.3. GPIO 输出控制 (0x38)	70
10.4. GPIO 输入中断通知 (0x39)	71
11. 分布式联动命令	73
11.1. 分布式联动控制 (0x4A)	73
11.2. 分布式联动广播 DP 配置 (0x4B)	75
12. 时间和天气	77
12.1. 时间同步 (0x24)	77
12.2. 天气信息查询 (0x3A)	78
12.3. 天气信息通知 (0x3B)	82
13. 附表	85
13.1. 附表 1: 天气数据	85
13.2. 附表 2: 中文天气 UTF-8 编码	86
13.3. 附表 3: 风向简码编码表	87
14. 更新记录	88

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 处理产品本身的逻辑，搭配 Zigbee 模组实现无线通信。MCU 与 模组之间通过串口（UART）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**：子设备之间通信，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或非长期维护的通用对接模组端固件的协议一致性。

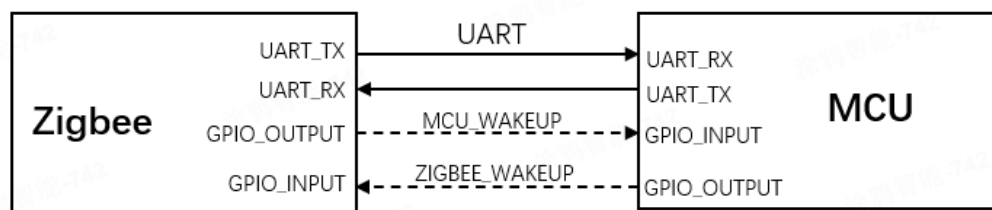


本文是涂鸦 Zigbee 串口通信类模组协议的集合，同时支持标准功耗设备和低功耗设备。不同模组因芯片平台不同，或相同模组因固件版本不同，其支持的命令和时序可能会有所不同。需以实际模组为准或向涂鸦咨询，建议采用公版固件最新版本。

1. 硬件连接

Zigbee 模组与 MCU 硬件连接如下图所示。

- 对于标准功耗设备，只需要连接 UART TX 和 UART RX 两个管脚。
- 对于低功耗设备，还需要额外连接两个唤醒管脚，分别用于 Zigbee 模组和 MCU 唤醒对方，让对方做好接收串口数据的准备。



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
ZS13S	PA5	PA6	PC1	PA3
ZSU	PA5	PA6	PC1	PA3
ZSC	PA5	PA6	PA0	PA3
TYZS3	PA0	PA1	PF5	PA2
TYZS5	PA0	PA1	PA4	PA3
ZTU	PB1	PB7	PB4	PB5/PD2

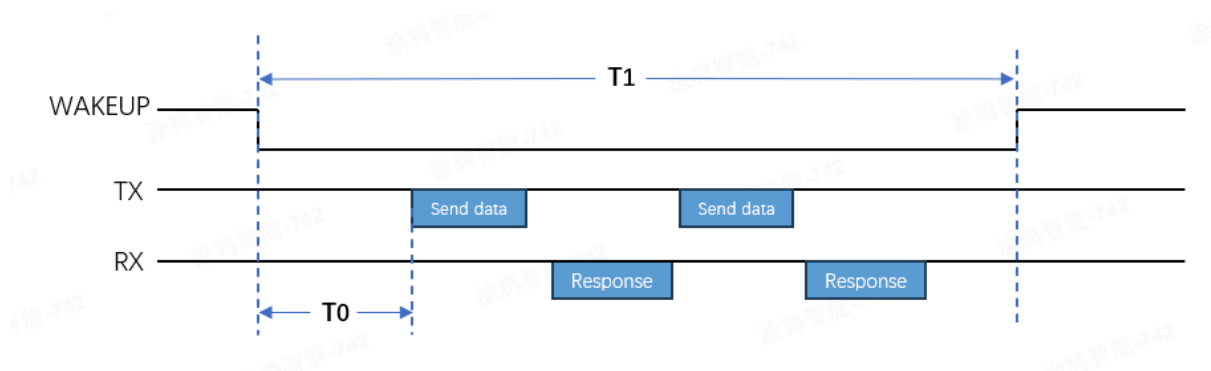
模组型号	模组 TX	模组 RX	唤醒 MCU	唤醒模组
ZT3L	PB1	PB7	PB4	PB5/PD2
ZT5	PB1	PB7	PB4	PB5/PD2
ZT2S	PB1	PB7	PB4	PB5/PD2

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，待本次串口数据全部发送完毕后，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（退出睡眠模式）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。



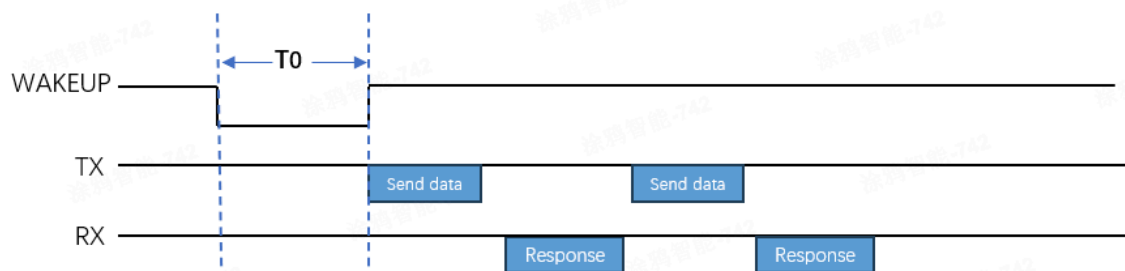
T0 是拉低唤醒管脚后的等待时间（待对方退出睡眠模式），T1 是单次唤醒对方的总时间。



- 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。



- 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。
- 是否支持脉冲唤醒，与 Zigbee 模组固件 Key 和版本有关。
- 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 。该字段会被用于帧格式检查（进行数值的比较，而不是字符/字符串的比较）。
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 。该字段会被用于帧格式检查（进行数值的比较，而不是字符/字符串的比较）。
SEQ	序列号	传输数据序列号（sequence number）， 1 ~ 0xffff0 循环。下文用 xx xx 表示
Command ID	命令字	命令 ID。持续更新。该字段会被用于帧命令解析（进行数值的比较，而不是字符/字符串的比较）。
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。如果有效数据长度为 0，则帧不包含该字段
Checksum	校验和	用于数据校验。从帧头开始按字节求和，然后将计算结果对 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** 两种传输模式。

- **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	√	√	√
0x03	M	模组配网/复位	√	√	√
0x00	Z	解绑并清除数据通知	√	√	√
0x2B	M	配置唤醒等待时间	×	√	×
0x07	M	查询模组信息	√	√	√
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	√	√	√

命令字	发送方	说明	强电版本	低功耗版本	场景版本
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	√	√	√
0x43	M	发送群组 DP 消息	×	×	√
0x42	M	发送群组标准命令	×	×	√
0x41	Z	场景配置	×	×	√
0x0A	M	场景触发	×	×	√
0x08	M	RF 产测	√	√	√
0x29	Z	信标产测通知	√	√	√
0x21	Z	Dongle 产测通知	√	√	√
0x22	M	Dongle 产测数据上报	√	√	√
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	√	√	√

命令字	发送方	说明	强电版本	低功耗版本	场景版本
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	√	√	√
0x4A	M	分布式联动控制	√	√	√
0x4B	M	分布式联动广播 DP 配置	√	√	√

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，都会主动发送此命令给 MCU。

MCU 需要先应答此命令，之后才能主动发送其它命令给模组。同时，此命令也用于 Zigbee 模组在出厂状态下，首次与 MCU 通信时确认 MCU 端的波特率。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 模组上电
6     模组->>MCU: (首次上电会进行波特率自适应过程) <br/>发送 0x01, 无 payload
7     MCU->>模组: 发送 0x01+产品信息
8     模组->>模组: 确认波特率<br/>处理产品信息
  
```

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

字段	字节数	说明
数据长度	2	N
数据	1	产品信息，示例 {"p":"fj5fqeg9","v":"1.0.0","g":1,"s":1,"lp":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" 或 "key":value 格式，下面是已支持的 key 和 value。

- **p**：必选。表示该产品在涂鸦开发者平台的 PID，例如上例中的 fj5fqeg9。
- **v**：必选。表示 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。模组收到组播消息后，只有 g 为 1 时，才会以 0x2A 命令通知 MCU。否则，发送 0x04 命令。建议 MCU 端支持此字段。
- **s**：可选。此字段是一个标志位，用于表示模组是否需要支持场景功能，只有 s 为 1 时支持场景功能。注意，仅 TuyaOS 合并版本固件支持此配置。
- **lp**：可选。此字段是一个标志位，用于表示模组是否为低功耗模式，只有 lp 为 1 时模组为低功耗模式，支持低功耗相关配置，同时固件支持低功耗模式和 router 模式切换。当设备已在网，切换模式时设备会先离开当前 Zigbee 网络。注意，仅 TuyaOS 合并版本固件支持此配置。

示例：55 AA 02 00 01 01 00 2F 7B 22 70 22 3A 22 66 6A 35 66 71 65 67 39 22 2C 22 76 22 3A 22 31 2E 30 2E 30 22 2C 22 67 22 3A 31 2C 22 73 22 3A 31 2C 22 6C 70 22 3A 31 7D xx

本例中，产品信息部分为：{"p":"fj5fqeg9","v":"1.0.0","g":1,"s":1,"lp":1}。

5.2. 模组网络状态通知 (0x02)

当网络状态发生改变，模组会主动发送此命令给 MCU。

网络状态是指 Zigbee 本地网络状态（非互联网连接状态），该状态不会因网关断电、父节点丢失等情况而变更。

```

1 sequenceDiagram
2 participant 模组

```

```

3 participant MCU
4
5 Note left of 模组: 模组网络状态改变
6 模组->>MCU: 发送 0x02+模组网络状态
7 MCU->>模组: 发送 0x02, 无 payload
8 MCU->>MCU: 处理模组网络状态

```

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, 表示模组还没有准确收到来自 MCU 的 [产品信息](#)。
- 0x03 : 设备正在配网。

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

MCU 应答

字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x02
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 02 00 00 xx

5.3. 模组配网/复位 (0x03)

MCU 可以主动发送此命令给模组，让模组软件复位或者开始配网。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想让模组软件复位/开始配网
6     MCU->>模组: 发送 0x03+0x00/0x01
7     模组->>MCU: 发送 0x03, 无 payload
8     模组->>模组: 软件复位/开始配网

```

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: 模组软件复位 0x01: 模组开始配网（先断开与网关的通信，再配网）
校验和	1	xx

示例： 55 AA 02 xx xx 03 00 01 01 xx 模组开始配网

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	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

5.4. 解绑并清除数据通知 (0x00)

在 App 端将设备 **解绑并清除数据** 后，模组主动发送此命令给 MCU。

MCU 收到该通知后可清除本地数据，即恢复出厂设置。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: App 解绑并清除数据
6     模组->>MCU: 发送 0x00+0x01
7     MCU->>模组: 发送 0x00+0x01
8     MCU->>MCU: 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

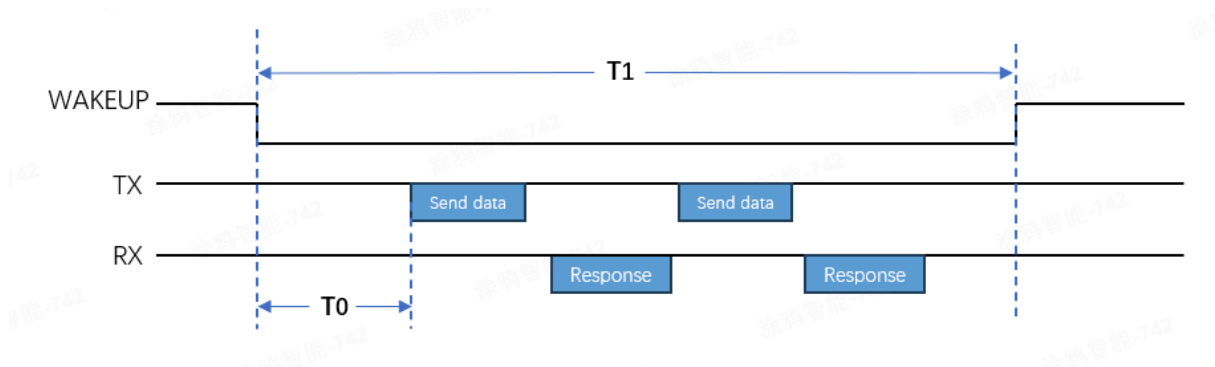
字段	字节数	说明
帧头	2	0x55AA
版本	1	0x02
序列号	2	xx xx
命令字	1	0x00
数据长度	2	0x0001
数据	1	0x01 默认值
校验和	1	xx

示例: 55 AA 02 xx xx 00 00 01 01 xx

5.5. 配置唤醒等待时间 (0x2B)

该命令只适用于低功耗类设备。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。不同 MCU 所需要的等待时间 T0 不同，MCU 在此等待时间内，做好接收串口数据的准备。

MCU 可以主动发送此命令给模组，配置等待时间 T0。



```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想设置模组等待 MCU 时间
6     MCU->>模组: 发送 0x2b+等待时间
7     模组->>模组: 处理等待时间
```

8 模组->>MCU：发送 0x2b+处理结果

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	2	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. 查询模组信息 (0x07)

MCU 可以主动发送此命令给模组，查询模组的固件版本号、授权信息、MAC 地址。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想查询模组信息
6     MCU->>模组: 发送 0x07+查询请求
7     模组->>MCU: 发送 0x07+模组信息
8     MCU->>MCU: 处理模组信息

```

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	0x07
数据长度	2	N
数据	N	请求模组信息 ID，可请求一个或者多个
校验和	1	xx

模组信息 ID

模组信息 ID	描述
0x01	模组固件版本号
0x02	模组授权信息
0x03	模组 MAC 地址

示例 1: 55 AA 02 xx xx 07 00 01 01 xx 表示请求模组固件版本号。

示例 2: 55 AA 02 xx xx 07 00 02 01 03 xx 表示请求模组固件版本号、模组 MAC 地址。

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	0x07
数据长度	2	N
字段	字节数	说明
数据	N	按照请求顺序，返回模组信息数据
校验和	1	xx

模组信息数据

字段	长度	说明
Type	1	类型，当前支持 1、2 和 3
Data	1/8	<ul style="list-style-type: none"> 当 type 为 1 或者 2 时，数据字段 1 个字节 当 type 为 3 时，数据字段 8 字节

5.7. 查询模组网络状态 (0x20)

MCU 可以主动发送此命令给模组，查询模组当前的网络状态。

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。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想查询模组网络状态
6     MCU->>模组: 发送 0x20, 无 payload
7     模组->>MCU: 发送 0x20+模组网络状态
8     MCU->>MCU: 处理模组网络状态

```

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
字段	字节数	说明
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
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

关于模组网络状态，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

5.8. 查询网关联网状态 (0x25)

MCU 可以主动发送此命令给模组，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。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想查询网关联网状态
6     MCU->>模组: 发送 0x25, 无 payload
7     Note right of 模组: 模组请求网关, 网关回复后
8     模组->>MCU: 发送 0x25+网关联网状态
9     MCU->>MCU: 处理网关联网状态
```

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	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

网关联网状态

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	网关联网状态
校验和	1	xx

网关联网状态

- 0x00：网关互联网离线。
- 0x01：网关互联网在线。
- 0x02：网关回复超时。

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

5.9. 配置 Zigbee 网络参数 (0x26)

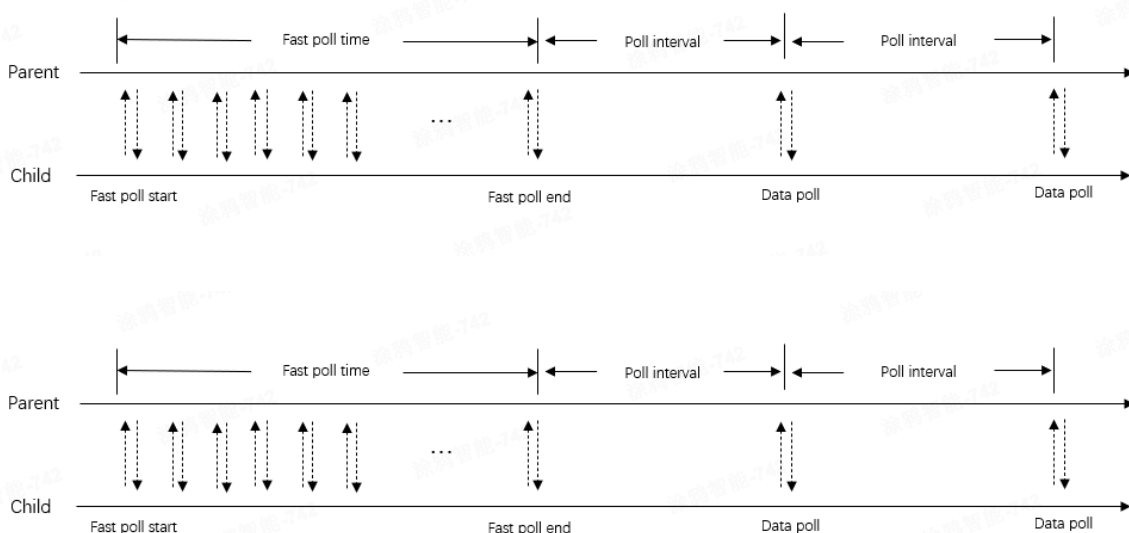
MCU 可以主动发送此命令给模组，设置模组的网络参数。

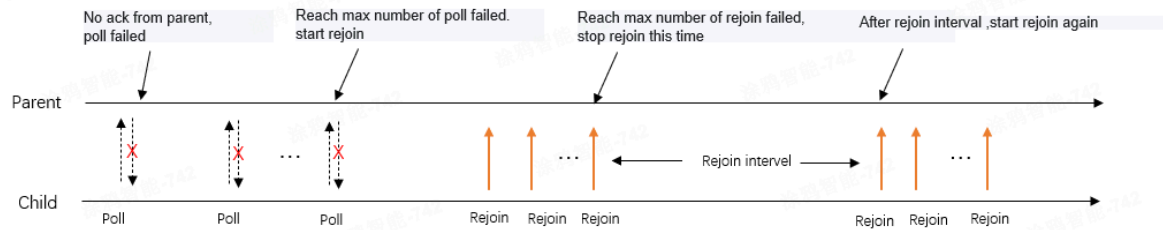
- 对于标准功耗设备，只需要关心配网时间和发射功率两个参数。
- 对于低功耗设备，需要关心所有参数。要理解这些参数，就需要先了解 Zigbee 低功耗设备的唤醒和数据收发机制。

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会暂存在其父设备处。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，则父设备会将数据发给子设备，反之，子设备可以再次进入休眠状态。

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 failed 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 失败最大尝试次数**，达到最大次数后会触发 Rejoin。
- **Fast poll time: 快速 Poll 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 的目的是加快与网关的数据交互，此时的 Poll 间隔为默认的 250 ms。
- **Rejoin interval: 每轮 Rejoin 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: 每轮 Rejoin 报文发送次数**，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，即设备每隔一个固定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。
- 对于标准功耗设备，网关会主动访问设备（作为心跳查询），因此标准功耗设备不需要主动发送心跳包给网关。

- 目前，只可以修改低功耗设备的心跳包发送周期，而不可以修改标准功耗设备的心跳包发送周期。

配网超时时间

本命令可以设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。

当 MCU 发送配网指令之后，模组开启配网。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

本命令可以设置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。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想设置模组的网络参数
6     MCU->>模组: 发送 0x26+网络参数
7     模组->>模组: 处理网络参数
8     模组->>MCU: 发送 0x26+处理结果
9     模组->>模组: 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	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	主动心跳间隔	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。
字节偏移	参数名称	适用设备	说明
2	每轮 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。

1	每轮 Rejoin 报文发送次数	低功耗	可配置 1 ~ 10 次，默认值为 1。
字节偏移	参数名称	适用设备	说明
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：

- 主动心跳间隔：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。
- 每轮 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。
- 每轮 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

版本	1	0x02
序列号	2	xx xx
字段	字节数	说明
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，模组会主动发送此命令给 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。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 模组收到 DP 消息 (0x00) <br/>如果是群控 DP 消息, <br/>且 MCU 不需要
    区分群控消息
6     模组->>MCU: 发送 0x04+DP 消息
7     MCU->>MCU: 处理 DP 消息
8     Note left of MCU: 需要上报 DP 消息<br/>(详情请参考 0x05 命令字)
  
```

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	...

1 个 DP 消息可能会包含多个 DP 数据，每 1 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 DP ID 的数据类型，部分数据类型的 Length 是固定的。
- Length：Value 的数据长度。
- Value：DP 有效数据。

Type：

Type	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

- 1 个 DP 消息可能会包含多个 DP 数据，例如 App 同时下发了多个 DP 给设备。
- 1 个 DP 消息最多只含有 1 个 Raw 类型的 DP 数据。即不会同时下发多个 Raw 类型的 DP，也不会混合下发其他类型的 DP。

MCU 回复

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	0x04
数据长度	2	0x0000
数据	0	无
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，且 MCU 需要区分群控消息，模组会主动发送此命令给 MCU。

如果使用涂鸦网关，那么 MCU 在收到此消息后，不需要进行 DP 消息应答。因为涂鸦网关发送群控消息后，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，而只支持 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 "g": "1" 时才会触发，否则模组收到群控消息后仍会通过 [DP 消息接收 \(0x04\)](#) 命令通知 MCU。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 模组收到群控 DP 消息 (0x00), <br/>且 MCU 需要区分群控消息
6     模组->>MCU: 发送 0x2a+DP 消息
7     MCU->>MCU: 处理 DP 消息
8     Note left of MCU: 不需要上报 DP 消息
```

模组发送

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	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 05 01 01 00 01 01 xx 表示模组收到了一条群控 DP 消息

- DP ID: 01
- Type: 01
- Length: 00 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 \(0x04\)](#)，MCU 需要发送此命令给模组，应答 DP 消息，以同步云端数据和 App 面板状态。

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

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 收到 0x04+DP 消息
6     模组->>MCU: 发送 0x04+DP 消息
7     MCU->>MCU: 处理 DP 消息
8     MCU->>模组: 发送 0x05+DP 消息
9     Note left of MCU: 模组上报 DP 消息给网关
10    模组->>MCU: 发送 0x05+上报结果

```

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

字段	字节数	说明
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 检测到本地状态发送变化，MCU 可以主动发送此命令给模组，上报发生变化的 DP（将本地状态同步到云端），并能够触发联动（包含本地联动和云端联动）。

模组收到此命令后会发送给网关，并且关心网关的响应，然后根据网关的响应情况应答 MCU。

MCU 可根据模组回复，判断模组与网关的链路是否通畅。

- 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**。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 检测到本地 DP 变化, <br/>想上报 DP 消息并触发联动
6     MCU->>模组: 发送 0x06+DP 消息
7     Note left of MCU: 模组上报 DP 消息给网关
8     模组->>MCU: 发送 0x06+上报结果

```

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 消息发给了网关，并在超时时间内收到网关的确认。
- 上报失败：网关确认超时、模组未加入网关。
- 1 个 DP 消息最多只含有 1 个 Raw 类型的 DP 数据。即不会同时下发多个 Raw 类型的 DP，也不会混合下发其他类型的 DP。

6.5. DP 消息上报-不触发联动 (0x2C)

当 MCU 检测到本地状态发送变化，MCU 可以主动发送此命令给模组，上报发生变化的 DP（将本地状态同步到云端），但不能触发联动（包含本地联动和云端联动）。

本命令是 [DP 消息上报 \(0x06\)](#) 的拓展，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

- 您需要确认所采用的模组和固件是否支持本命令。
- 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。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 检测到本地 DP 变化, <br/>想上报 DP 消息, 但不触发联动
6     MCU->>模组: 发送 0x2c+DP 消息
7     Note left of MCU: 模组上报 DP 消息给网关
8     模组->>MCU: 发送 0x2c+上报结果
```

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 消息发给了网关, 并在超时时间内收到网关的确认。
- 上报失败: 网关确认超时、模组未加入网关。
- 1 个 DP 消息最多只含有 1 个 Raw 类型的 DP 数据。即不会同时下发多个 Raw 类型的 DP, 也不会混合下发其他类型的 DP。

6.6. DP 查询 (0x28)

当网关主动查询设备的 DP, 模组会主动发送此命令给 MCU。MCU 收到本命令后, 需要先应答本命令, 再通过命令字 0x06 或 0x2C 上报对应的 DP。建议通过 0x06 上报。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。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 网关想查询设备 DP
```

```

6  模组->>MCU: 发送 0x28+DP ID List
7  MCU->>模组: 发送 0x28, 无 payload
8  MCU->>模组: 根据 DP ID List, <br/>发送 0x06/0x2c+DP 消息
9  Note left of MCU: 后续流程请参考对应命令字

```

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 ID List
校验和	1	xx

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
字段	字节数	说明
数据长度	2	0x0000

数据	0	无
校验和	1	xx

示例： 55 aa 02 xx xx 28 00 00 xx

6.7. 发送广播 DP 消息 (0x27)

MCU 可以主动发送此命令给模组，让模组发送 DP 广播消息，全网络的设备都可以接收到。如果存在低功耗设备，则该设备需要处于唤醒状态，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。

i

广播消息之间需要一定的时间间隔，以避免造成网络拥堵。间隔由网络的规模决定。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想广播发送 DP 消息
6     MCU->>模组: 发送 0x27+DP 消息
7     Note right of 模组: 模组广播发送 DP 消息
8     模组->>MCU: 发送 0x27+发送结果
```

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 00 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)

MCU 可以主动发送此命令给模组, 让模组以组播方式向群组发送 DP 消息, 以控制群组中支持涂鸦 DP 的设备。



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。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想发送群组 DP 消息
6     MCU->>模组: 发送 0x43+群组 DP 消息
7     Note right of 模组: 模组发送群组 DP 消息
8     模组->>MCU: 发送 0x43+发送结果
  
```

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	Group ID (2 字节) + DP 消息格式
校验和	1	xx

该数据中的 Group ID 是通过 [场景配置](#) 命令获取的。

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
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
字段	字节数	说明
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)

MCU 可以主动发送此命令给模组，让模组以组播方式向群组发送标准命令，以控制群组中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。



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。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想发送群组标准命令
6     MCU->>模组: 发送 0x42+群组标准命令
7     Note right of 模组: 模组发送群组标准命令
8     模组->>MCU: 发送 0x42+发送结果
```

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

该数据中的 Group ID 是通过 [场景配置](#) 命令获取的。

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，如果想获取每个标准命令的 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

- Off
- On
- Toggle

Level control cluster

- Move to level

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

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
- Stop move step
- Move color temperature
- Step color temperature

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），其作用是充当场景中的条件设备。这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 **标准场景** 和 **联动场景**。创建方式不同，创建出来的场景类型也不同。

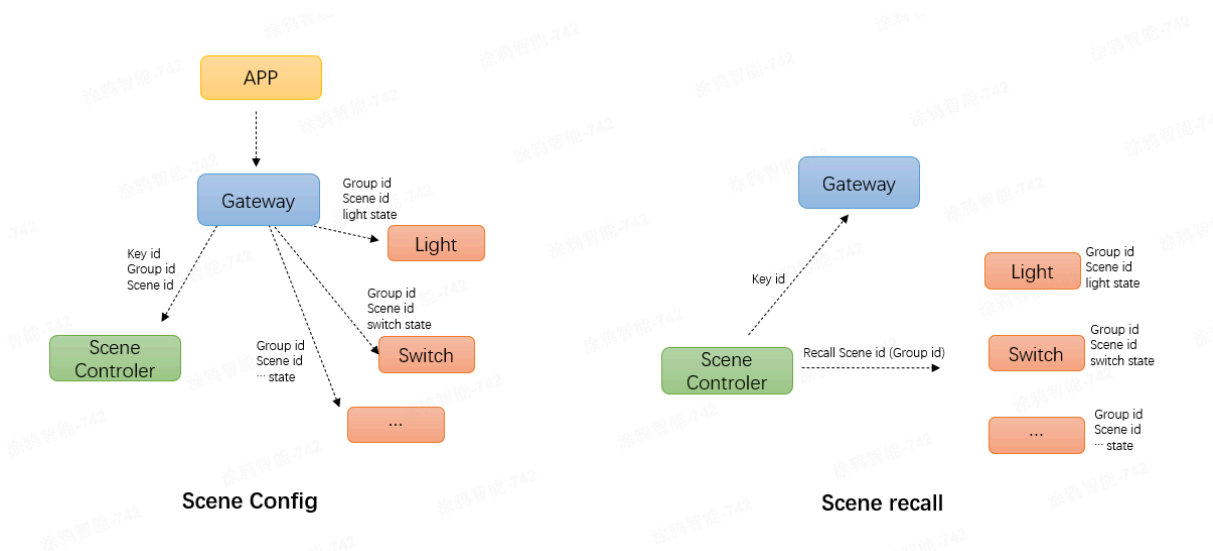
7.1. 标准场景

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，场景控制器直接发送控制命令给执行设备。

标准场景的创建条件：

- 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必须是同一个网关下的 Zigbee 设备。可以添加多个执行设备，但是不能有群组。如果添加群组作为执行设备，则会自动转为联动场景。

标准场景的创建与触发，如下图所示：



- **标准场景配置 (Scene config)：**在 App 上创建标准场景后（场景控制器设备的 App 面板），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）。对于群组地址和场景号与此相同的被控设备，App 会向这些设备发送群组

地址 (Group ID)、场景号 (Scene ID)和被控设备要执行的动作。此时，场景控制器和被控设备都被添加到同一个群组中。之后，场景控制器可以向这个群组地址发送命令。

- **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：指通过网关或云端（作为中转），触发设备将状态报告给网关。

网关会检查该联动关系是否已托管在本地。如果已托管在本地，则该场景称为 **网关联动**，反之，称为 **云端联动**。

- **网关联动**：由 Zigbee 网关保存联动关系，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而控制执行设备。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，云端根据联动关系而控制执行设备。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，因为这是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 (0x41)

当模组收到来自网关下发的按键号、群组地址和场景号，模组会主动发送此命令给 MCU。



本命令需要特定模组固件才支持，用于场景控制器类设备。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 网关下发场景消息
6     模组->>模组: 处理场景消息
7     Note right of 模组: 模组上报场景消息
8     模组->>MCU: 发送 0x41+场景信息
```



```

9   MCU->>MCU: 处理场景信息<br/>（例如获取 group id）
10  MCU->>模组：发送 0x41+处理结果

```

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	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 (0x0A)

MCU 可以主动发送此命令给模组，让模组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，检测到按键动作时，模组会发送标准场景触发命令（Recall scene，用于触发标准场景），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。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 检测到场景触发
6     MCU->>模组: 发送 0x0a+按键 ID
7     模组->>模组: 发送 recall scene
8     Note right of 模组: 模组上报按键 ID
9     模组->>MCU: 发送 0x0a+操作结果
10    模组->>模组: 处理场景信息
11    模组->>MCU: 发送 0x04/0x2a+DP 消息
```

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	0x0A
数据长度	2	0x0001
数据	1	0x00：失败 0x01：成功 (对于标准场景和云端联动，模组成功发送其中一个，都认为触发成功)
校验和	1	xx

示例： 55 AA 02 xx xx 0A 00 01 01 xx 触发成功。

8. 产测类命令

8.1. RF 产测 (0x08)

MCU 可以主动发送此命令给模组，让模组开始 RF 测试，从而测试模组的 RF 数据发送和接收是否正常。

开始 RF 测试前，需要准备一个产测 Dongle，并将产测 Dongle 配置为 RF 模式，详细流程请参考 [设置 Zigbee Dongle](#)。

模组收到该命令后，会进入 RF 测试，然后向空中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 每收到一包，都会回环式返回。如果模组收到的数据包，与自身发送的裸数据包内容相同，则会认为其成功接收了一个回环数据包。

RF 测试结束后，模组会发送 RF 测试结果（含接收的有效数据包个数）给 MCU，MCU 可以评估模组的发送和接收能力。但该命令暂未对信号强度进行检测。



RF 产测需要在未配网状态下进行。产测完成后，模组需要重启才能再次正常运行程序。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想测试模组的 Zigbee 数据收发能力
6     MCU->>模组: 发送 0x08+信道号
7     模组->>模组: 进行 RF 产测
8     模组->>MCU: 发送 0x08+产测结果
9     MCU->>MCU: 处理产测结果
```

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)，默认第 11 号信道

字段	字节数	说明
校验和	1	xx

信道号要与产测 Dongle 配置的信道号一致。

请勿在同一信道下同时对多台设备进行 RF 测试，否则会导致互相干扰。

示例： 55 AA 02 xx xx 08 00 01 0B xx 在第 11 号信道进行 RF 产测。

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	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（状态 1 字节，回环数据包个数 1 字节）：

- Data[0] = 0x00，表示测试失败。Data[1] = 0x00（固定值），表示未接收到回环的数据包。
- Data[0] = 0x01，表示测试成功。Data[1] = 回环数据包个数（1 ~ 100）。

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。

8.2. 信标产测通知 (0x29)

模组上电时，会有短时间的信标接收窗口（一般是 300 ms），如果模组在该窗口内接收到来自 Dongle 的信标，模组会主动发送此命令给 MCU。

MCU 收到后，进行效果或自检流程（成品产测）。例如，灯具没有开启产测的有效手段，所以可以通过这个方式进入产测流程，以检查灯具上各灯珠是否可以正常发光。

开始成品产测前，需要准备一个产测 Dongle，并将产测 Dongle 配置为 Beacon 模式，即 BC 模式。详细流程，参考 [设置 Zigbee Dongle](#)。Dongle 进入该模式后，会持续向空中发送特定格式的数据包（信标）。



- 信标产测需要在未配网状态下进行。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。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 模组进入 Beacon 产测
6     模组->>MCU: 发送 0x29+0x00
7     MCU->>MCU: 进行 Beacon 产测
8     MCU->>模组: 发送 0x29+产测结果
  
```

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

字段	字节数	说明
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。

8.3. Dongle 产测通知 (0x21)

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 模组进入 Dongle 产测模式
6     模组->>MCU: 发送 0x21, 无 payload
7     MCU->>模组: 发送 0x21, 无 payload
8     Note left of 模组: 模组收到上位机 Dongle 产测命令
9     模组->>MCU: 发送 0x21+Dongle 产测命令
10    MCU->>模组: 发送 0x21, 无 payload
11    MCU->>MCU: 进行 Dongle 产测
12    Note left of MCU: 需要上报产测结果<br> (详情请参考 0x22 命令字)
```

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	0x21
数据长度	2	N
数据	N	Dongle 产测数据如果是进入 Dongle 产测模式, 则数据为空, 数据长度为 0 如果是收到上位机产测数据, 则透传该数据

字段	字节数	说明
校验和	1	xx

i

进入 Dongle 产测时，模组可能还没有获取到波特率。所以模组进入 Dongle 后，会分别使用 9600 和 115200 波特率，各发送 3 次进入 Dongle 产测通知。

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	0x21
数据长度	2	0x0000
数据	0	无
校验和	1	xx

8.4. Dongle 产测数据上报 (0x22)

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 进行 Dongle 产测后，<br/>需要上报产测数据
6     MCU->>MCU: 进行 Dongle 产测
7     MCU->>模组: 发送 0x22+产测数据
8     Note right of 模组: 模组上报产测数据
9     模组->>MCU: 发送 0x22+上报数据
```

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	0x22
数据长度	2	N
数据	N	Dongle 产测数据
校验和	1	xx

Dongle 产测数据内容 ID

Type	ID
DONGLE_MF_RESULT_CMD	0x01
DONGLE_MF_BUTTON_NOTIFY	0x02
DONBLE_MF_BOOL_NOTIFY	0x03
DONBLE_MF_GENERAL_SEND	0x04

• DONGLE_MF_RESULT_CMD 格式

CMD ID	Result
0x01	<ul style="list-style-type: none"> 0 : MF_TEST_SUCCESS 1 : MF_TEST_FAILED 2 : MF_TEST_DOING

• DONGLE_MF_BUTTON_NOTIFY 格式

CMD ID	Key ID
0x02	4 Bytes 小端在前

• DONBLE_MF_BOOL_NOTIFY 格式

CMD ID	Sensor Type	Index	Result
0x03	传感类型	序号	结果
			<ul style="list-style-type: none"> 1 : TRUE 0 : FALSE

• DONBLE_MF_GENERAL_SEND 格式

CMD ID	CMD	Len	Data
0x04	命令字类型	数据长度	数据

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	0x22
数据长度	2	0x0001
数据	N	0: 成功 1: 失败
校验和	1	xx

Dongle 产测示例:

• 进入产测通知

1. 模组->MCU: 55 AA 02 xx xx 21 00 00 xx

• 指示灯测试

1. 模组->MCU: 55 AA 02 xx xx 21 00 03 09 08 02 xx

08 02 表示 LED 交替闪烁 (动作指令由产测 JSON 决定)

2. 产测上位机手动单击 LED 测试通过。

• 按键测试

1. 模组->MCU: 55 AA 02 xx xx 21 00 03 09 0A 00 xx

2. MCU->模组: 55 AA 02 xx xx 22 00 05 02 00 00 00 00 xx

0x00000000 表示 KeyID, 模组会处理成 {"keyID":0} 转给 Dongle

3. 产测上位机显示按键测试通过。

• 温湿度传感器测试 0x0C

1. 模组->MCU: 55 AA 02 xx xx 21 00 03 09 0C 00 xx

2. MCU->模组: 55 AA 02 xx xx 22 00 19 04 0C 16 7B 22 53 31 22 3A 32 35 30 30 2C 22

53 32 22 3A 38 30 30 30 7D 20 xx

数据内容表示为 {"S1":2500,"S2":8000}

3. 产测上位机显示数据温度 25°C、湿度 80%，温湿度传感测试通过。

• 亮度传感器测试

1. 模组->MCU: 55 AA 02 xx xx 21 00 1D 09 13 7B 22 74 79 70 65 22 3A 22 4C 75 6D 69

6E 61 6E 63 65 22 2C 22 63 68 22 3A 30 7D xx

数据内容表示为 {"type":"Luminance","ch":0}

2. MCU->模组: 55 AA 02 xx xx 22 00 29 04 13 26 7B 22 74 79 70 65 22 3A 22 4C 75 6D
69 6E 61 6E 63 65 22 2C 22 63 68 22 3A 30 2C 22 76 61 6C 22 3A 31 35 30 30 7D xx

数据内容表示为 {"type":"Luminance","ch":0,"val":1500}

3. 产测上位机显示亮度 1500 lux, 亮度传感器测试通过。

• PIR 传感测试

1. 模组->MCU: 55 AA 02 xx xx 21 00 02 0B 01 xx

2. MCU->模组: 55 AA 02 xx xx 22 00 04 03 01 00 01 xx 表示 PIR 状态 TRUE。 55 AA 02
xx xx 22 00 04 03 01 00 00 xx 表示 PIR 状态 FALSE。

3. 产测上位机 PIR 传感器测试通过。

• PIR 感应距离测试

1. 模组->MCU: 55 AA 02 xx xx 21 00 21 09 13 7B 22 74 79 70 65 22 3A 22 53 65 6E 73
65 44 69 73 74 61 6E 63 65 22 2C 22 63 68 22 3A 30 7D xx

数据内容表示为 {"type":"SenseDistance","ch":0}

2. MCU->模组: 55 AA 02 xx xx 22 00 2C 04 13 7B 22 74 79 70 65 22 3A 22 53 65 6E 73
65 44 69 73 74 61 6E 63 65 22 2C 22 63 68 22 3A 30 2C 22 76 61 6C 22 3A 35 30 30
7D xx

数据内容表示为 {"type":"SenseDistance","ch":0,"val":500}

3. 产测上位机显示 PIR 感应距离为 500, 感应距离测试通过。

• 门磁传感测试

1. 模组->MCU: 55 AA 02 xx xx 21 00 02 0B 00 xx

2. MCU->模组: 55 AA 02 xx xx 22 00 04 03 00 00 01 xx 表示门磁状态 TRUE。 55 AA 02
xx xx 22 00 04 03 00 00 00 xx 表示门磁状态 FALSE。

• 防拆测试

1. 模组->MCU: 55 AA 02 xxxx 21 00 02 0B 07 xx

2. MCU->模组: 55 AA 02 xx xx 22 00 04 03 07 00 01 xx 表示防拆状态 TRUE。 55 AA 02
xx xx 22 00 04 03 07 00 00 xx 表示防拆状态 FALSE。

3. 产测上位机防拆测试通过。

• 电池电压测试

1. 模组->MCU: 55 AA 02 xx xx 21 00 03 09 90 00 xx

2. MCU->模组: 55 AA 02 xx xx 22 00 13 04 90 10 7B 22 50 22 3A 31 2C 22 42 22 3A 33
30 30 30 7D xx

数据内容表示为 {"P":1,"B":3000}

3. 产测上位机显示电压为 3000 mV，电压测试通过。

9. MCU OTA 类命令

通过模组进行 MCU 固件 OTA 升级的流程如下：

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. 模组收到该通知后，模组会以 [OTA 升级通知](#) 命令通知 MCU。
4. MCU 收到该命令后，检查 OTA 升级信息。如果符合 OTA 升级条件，则 MCU 可以发送 [OTA 文件请求](#) 命令给模组。
5. 模组收到该命令后，会向网关请求 OTA 文件数据。网关返回后，模组会通过 [OTA 文件请求](#) 命令把 OTA 文件发送给 MCU。受 OTA 单包尺寸限制，需要多次进行多次 OTA 文件请求，才可以获取完整个 OTA 文件。
6. 待整个文件请求完毕，MCU 可以开始内部的升级操作。
7. MCU 升级完成后，[上报 OTA 结果](#)，并通过 [MCU 固件版本查询](#) 命令上报新的 MCU 固件版本号。

9.1. MCU 固件版本查询 (0x0B)

当模组收到来自网关的查询命令，模组会主动发送此命令给 MCU，MCU 需要应答 MCU 当前的固件版本号。例如，判断是否推送 OTA。

MCU 也可以通过此命令，主动发送 MCU 当前的固件版本号给模组，例如模组配网成功、MCU 固件 OTA 升级成功。

模组获取到 MCU 当前的固件版本号后，会同步 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。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 网关查询 MCU 固件版本号
6     模组->>MCU: 发送 0x0b, 无 payload
7     MCU->>模组: 发送 0x0b+MCU 固件版本号
8     Note right of 模组: 模组上报 MCU 固件版本号
```

模组发送

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	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，表示版本号 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 01 01 0011，即 1.1.3。

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。在面板上开启 MCU 固件升级后，网关会通知模组。

模组收到通知后，模组会发送此命令给 MCU，通知 MCU 准备 OTA 升级。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note left of 模组: 云端通知 MCU 固件 OTA 升级<br/>（在面板上单击升级）
6     模组->>MCU: 发送 0x0c+OTA 升级信息
7     MCU->>MCU: 校验本地信息与 OTA 升级信息
8     Note left of MCU: 如果校验成功，符合升级条件
9     MCU->>模组: 发送 0x0c+校验结果
10    Note left of MCU: 之后可通过 0x0d 命令字进行 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
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（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 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，则说明传输过程可能有问题，可以报升级失败。

示例：55 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：41 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：检查失败 0x01：检查成功
校验和	1	xx

示例： 55 AA 02 xx xx 0C 00 01 00 xx

目前，模组端不处理 MCU 对 0x0C 命令的应答帧。

9.3. OTA 文件请求 (0x0D)

在 MCU 收到 [OTA 升级通知](#) 命令，且 OTA 升级信息检查结果为正确的前提下，MCU 可以主动发送此命令给模组，让模组向网关请求获取 OTA 文件（即下载 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 and 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。

模组向网关请求后，如果不是最后 1 包 OTA 文件，且 3 秒内没有收到 MCU 发送的该命令，则会重新向网关请求获取当前偏移量的 OTA 文件，最多重新请求 2 次。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: 校验成功，符合升级条件<br/>（详情请参考 0x0c 命令字）
6     MCU->>模组: 发送 0x0d+OTA 升级数据请求
7     Note right of 模组: 模组请求 OTA 升级数据，网关返回后
8     模组->>MCU: 发送 0x0d+OTA 升级数据
9     MCU->>MCU: 处理 OTA 升级数据<br/>（例如写入 Flash）
10    Note left of MCU: 之后可重复上述步骤，继续升级
11    Note left of MCU: 直到 MCU 获取了最后一包 OTA 升级数据<br/>（详情请参考 0x0e 命令字）
```

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, 不能超过 48 字节。

示例: 55 AA 02 xx xx 0D 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

字段	字节数	说明
数据长度	2	N
数据	N	请求结果 (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

- 请求结果：
 - 0x00 请求成功。
 - 0x01 请求失败。如果请求失败，则后面的数据不存在。
- 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)

在 MCU OTA 升级完成后，MCU 可以主动发送此命令给模组，让模组将 OTA 升级结果上报给网关。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 获取了最后一包 OTA 升级数据
6     MCU->>MCU: 处理 OTA 升级数据<br/>(例如写入 Flash)
7     MCU->>模组: 发送 0x0e+OTA 升级结果
8     模组->>MCU: 发送 0x0e+0x00
9     Note right of 模组: 模组上报 OTA 升级结果

```

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。网关收到该 OTA 结果后，会通知面板提醒 OTA 升级成功。
- 0x01：升级失败。网关收到该 OTA 结果后，不会通知面板提醒 OTA 升级失败。当 OTA 升级超时后，面板才会提醒 OTA 升级超时。

示例： 55 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

- 升级结果： 0x00，表示成功
- 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：上报成功 0x01：上 报失败

字段	字节数	说明
校验和	1	xx

示例： 55 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（默认配置了输入中断）和输出。

要使用模组的 GPIO，就需要先通过 [GPIO 配置](#) 命令配置模组 GPIO。

- 如果模组 GPIO 被配置为输出，
- 则可以通过 [GPIO 输出控制](#) 命令，指定输出电平。
- 也可以通过 [GPIO 电平读取](#) 命令，获取当前电平。
- 如果模组 GPIO 被配置为输入，
- 则可以通过 [GPIO 电平读取](#) 命令，获取当前电平。
- 因为输入模式默认配置了输入中断，所以当模组检测到 GPIO 电平变化，模组就会通过 [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
1	GPIO_LEVEL_HIGH
2	GPIO_LEVEL_ALL

10.1. GPIO 配置 (0x36)

MCU 可以主动发送此命令给模组，配置模组 GPIO。MCU 可以一次性配置多个模组 GPIO。

```

1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想配置模组 GPIO
6     MCU->>模组: 发送 0x36+GPIO 配置
7     模组->>模组: 配置 GPIO
8     模组->>MCU: 发送 0x36+配置结果
  
```

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
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 :
- 0x01 : 表示配置失败。
- 0x00 : 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)

MCU 可以主动发送此命令给模组，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](#) 命令配置模组 GPIO。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想读取模组 GPIO 电平
6     MCU->>模组: 发送 0x37+GPIO 信息
7     模组->>模组: 读取 GPIO 电平
8     模组->>MCU: 发送 0x37+读取电平结果
```

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
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	...

- GPIO LEVEL:
- 0x01：表示高电平。
- 0x00：表示低电平。
- 0xff：表示失败，例如模组 GPIO 不存在或未配置。

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 GPIO 被配置为输出时，MCU 可以主动发送此命令给模组，控制该 GPIO 输出高电平或低电平。

使用此命令之前，需要先通过 [GPIO 配置](#) 命令配置模组 GPIO。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想写入模组 GPIO 电平
6     MCU->>模组: 发送 0x38+GPIO 信息
7     模组->>模组: 写 GPIO 电平
8     模组->>MCU: 发送 0x38+写电平结果
```

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	...

- GPIO LEVEL:
- 0x01 ：表示高电平。
- 0x00 ：表示低电平。

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，例如模组 GPIO 不存在或未配置为输出模式。

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 GPIO 被配置为输入中断，且检测到 GPIO 中断触发，则模组会通过本命令通知 MCU。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
```

```

5 Note left of 模组: 模组检测到模组 GPIO 触发输入中断
6 模组->>MCU: 发送 0x39+GPIO 信息
7 MCU->>模组: 发送 0x39, 无 payload
8 MCU->>MCU: 处理 GPIO 信息

```

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

目前，模组端不处理 MCU 对 0x39 命令的应答帧。

11. 分布式联动命令



分布式联动功能需要配合特定网关、App 使用，详情请咨询产品经理。

Zigbee 分布式联动功能将原本存储在云端的联动规则下放到各个 Zigbee 子设备中进行存储。下发成功后，不再依赖 Zigbee 网关可以脱离外网进行联动触发，可以降低对网关和云端的依赖，提高联动触发的稳定性。

分布式联动设备分为条件设备和执行设备两种，条件设备作为联动的触发端，当条件设备的 DP 发生变化的时候它会广播自身的 DP 值发生了变化。当相关的执行设备接收到这条广播，会检查自身存储的联动规则中条件是否包括了该设备的该 DP，并且该 DP 变化恰好导致了条件被满足。如果联动规则条件满足，则会触发动作执行。

如：我们配置了一条分布式联动规则，当门磁触发有人时（DP1 = 1 时）灯执行开灯动作。那么当门磁本地触发有人时，门磁会向外广播一条命令通知网络中的设备当前它的 DP1 已经变成了 1；当网络中的灯收到了这条命令、发现其配置的联动规则中条件恰好满足，则会执行开启的动作。

通用对接模组既支持作为条件设备，也支持作为执行设备：

- 作为执行设备：模组收到符合自身存储的联动规则中条件的消息后，模组会以 DP 的形式通知给 MCU。
- 作为条件设备：MCU 可配置（命令字 0x4A）模组收到哪些 DP 消息后广播自身 DP 值，也可手动发送（0x4B）启用、禁用，执行分布式联动广播。

11.1. 分布式联动控制（0x4A）

MCU 可以主动发起分布式控制命令，控制本地设备分布式联动启用、禁用，触发功能。

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	0x4A

数据长度	2	0x0006
数据	5	参考分布式数据内容格式
校验和	1	xx

分布式数据内容格式

名称	数据类型	说明
action	UINT8_T	0x00: 禁用 0x01: 启用 0x02: 执行
auto id	UINT16_T	分布式联动 ID, 由网关下发 (可配置通过 DP 形式下发)
loopback	UINT8_T	是否使能回环触发: 0x00: 禁用 0x01: 使能
delay time	UINT16_T	延时发送时间, 单位毫秒

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	0x4A
数据长度	2	0x0001
数据	1	0x00: 成功 0x01: 失败
校验和	1	xx

11.2. 分布式联动广播 DP 配置 (0x4B)

MCU 可以配置自身哪些 DP 在消息上报 (0x06) 的同时触发分布式 DP 广播。当已配置联动规则的执行设备接收到这条广播, 会检查自身存储的联动规则中条件是否包括了此 DP。如果

存储的联动规则中包含此 DP 并且此 DP 变化恰好满足联动规则，则会触发联动规则中的动作执行。



- 最多支持八个 DP 数据广播。
- 每次 DP 列表会覆盖之前的 DP 列表，不会向后追加。此列表存储在 Flash 中，重启不需要重新配置。
- 注意当前仅支持 DP 类型为 BOOL、ENUM、VALUE 类型的数据。

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	0x4B
数据长度	2	N
数据	N	参考广播 DP 配置信息
校验和	1	xx

广播 DP 配置信息

名称	数据类型	说明
Enable	BOOL_T	是否使能分布式广播功能： 0x00：禁用 0x01：使能
DP 数量	UINT8_T	广播 DP 数量
DP ID 1	UINT8_T	DP ID
...
DP ID N	UINT8_T	DP ID, N <= 8

模组返回

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	0x4B
数据长度	2	0x0001
数据	1	0x00: 成功 0x01: 失败
校验和	1	xx

12. 时间和天气

12.1. 时间同步 (0x24)

MCU 可以主动发送此命令给模组，让模组向网关请求时间信息。

网关回复模组后，模组通过本命令通知 MCU。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
4
5     Note right of MCU: MCU 想同步网关时间
6     MCU->>模组: 发送 0x24, 无 payload
7     Note right of 模组: 模组请求时间信息，网关返回后
8     模组->>MCU: 发送 0x24+时间信息
9     MCU->>MCU: 处理时间信息
```

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 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

12.2. 天气信息查询 (0x3A)

MCU 可以主动发送此命令给模组，让模组向网关请求天气/城市信息。

网关从互联网获取到天气/城市信息后回复给模组，然后模组通过 [天气信息通知](#) 命令通知 MCU。

地址类型

- 设备地址（网关激活地址）：即天气/城市信息以网关的激活地址为准。如果网关激活地址和当前网关所在地一致，那么这些天气/城市信息也可以视作设备所在地的天气/城市信息。
- App 设置的家庭地址：即天气/城市信息以用户在涂鸦 App 上设置的家庭地址为准。

预报（天气）

- 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。

数据

- 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](#)。
- 城市数据：可查询所在城市名称字符串，主要用于显示屏显示。



- 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。

```

1 sequenceDiagram
2   participant 模组
3   participant MCU
4
5   Note right of MCU: MCU 想查询天气/城市信息
6   MCU->>模组: 发送 0x3a+天气/城市信息请求
7   模组->>MCU: 如果模组不在线, 发送 0x3a+0x00<br/> (发送后不执行后续流程)
8   Note right of 模组: 模组请求天气/城市信息, 网关返回后
9   模组->>MCU: 发送 0x3b+天气/城市信息
10  MCU->>MCU: 处理天气/城市信息

```

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	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	0x01 ~ 0x07: 天气预报的天数
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

- 天气预报天数不可为 0，如只需实时天气，不需要天气预报可不添加天气预报标志位及天气预报天数两个字段。
- 实时天气标志位必须存在，且不可在没有天气预报标志位的前提下，是否需要实时天气也为 0，否则网关不会返回数据。

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 07 11 00 01 02 03 13 01 xx

- 11 : 版本号
- 00 : 设备地址（网关）
- 01 02 03 : 天气 ID, 01 温度, 02 湿度, 03 天气概况
- 13 : 实时天气标志位
- 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 : 设备地址（网关）

- 01 02 03 : 天气 ID, 01 温度, 02 湿度, 03 天气概况.
- 12 : 天气预报标志位
- 07 : 请求近 7 天的天气预报 (含今天)
- 13 : 实时天气标志位
- 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 79 xx

- 11 : 版本号
- 00 : 设备地址 (网关)
- 63 2E 61 72 65 61 : 区县请求名称
- 63 2E 63 69 74 79 : 城市请求名称

表示请求设备地址的城市名称和区县名称。

模组返回

- 若模组未入网或网关返回超时, 则通过本命令返回失败给 MCU。
- 若模组已入网, 且网关回复不超时, 则通过 [天气信息通知](#) 命令返回给 MCU。

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	0x3A
数据长度	2	0x0001
数据	1	0x00 (请求失败)
校验和	1	xx

12.3. 天气信息通知 (0x3B)

模组收到来自网关的天气/城市信息后, 模组通过本命令将天气信息发送给 MCU。

网关的天气/城市信息取自互联网。

```
1 sequenceDiagram
2     participant 模组
3     participant MCU
```

4
5 Note left of 模组: 模组收到天气/城市信息
6 模组->>MCU: 发送 0x3b+天气/城市信息
7 MCU->>MCU: 处理天气/城市信息

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
天气预报天数	1	0x00: 不包含天气预报 0x01 ~ 0x07: 天气预报的天数
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

天气信息

N

天气参数 1 + 天气参数 2 +

...

天气参数 的格式：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 **是否需要实时天气** = 0x01，则包含该字段。该字段的长度与天气 ID 有关。
- 预报值：若 **天气预报天数** ≠ 0x00，则包含该字段。根据 **天气预报天数** 决定该字段包含几组数据。

该帧的 payload 示例：

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°C	今天温度 8°C		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	天气概况ID		
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。

不能出现同时不包含实时值与预报值的情况。

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
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

该帧的 payload 示例： 11 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。

13. 附表

13.1. 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

13.2. 附表 2：中文天气 UTF-8 编码

天气 ID	编码	十六进制	描述
0x78	120	31 32 30	晴
0x65	101	31 30 31	大雨
0x66	102	31 30 32	雷暴
0x67	103	31 30 33	沙尘暴
0x68	104	31 30 34	小雪
0x69	105	31 30 35	雪
0x6a	106	31 30 36	冻雾
0x6b	107	31 30 37	暴雨
0x6c	108	31 30 38	局部阵雨
0x6d	109	31 30 39	浮尘
0x6e	110	31 31 30	雷电
0x6f	111	31 31 31	小阵雨
0x70	112	31 31 32	雨
0x71	113	31 31 33	雨夹雪
0x72	114	31 31 34	尘卷风
0x73	115	31 31 35	冰粒
0x74	116	31 31 36	强沙尘暴
0x75	117	31 31 37	扬沙
0x76	118	31 31 38	小到中雨
0x77	119	31 31 39	大部晴朗
0x79	121	31 32 31	雾
0x7a	122	31 32 32	阵雨
0x7b	123	31 32 33	强阵雨
0x7c	124	31 32 34	大雪
0x7d	125	31 32 35	特大暴雨
0x7e	126	31 32 36	暴雪
0x7f	127	31 32 37	冰雹
0x80	128	31 32 38	小到中雪
0x81	129	31 32 39	少云
0x82	130	31 33 30	小阵雪
0x83	131	31 33 31	中雪
0x84	132	31 33 32	阴
0x85	133	31 33 33	冰针
0x86	134	31 33 34	大暴雨
0x88	136	31 33 36	雷阵雨伴有冰雹

0x89	137	31 33 37	冻雨
0x8a	138	31 33 38	阵雪
0x8b	139	31 33 39	小雨
0x8c	140	31 34 30	霾
0x8d	141	31 34 31	中雨
0x8e	142	31 34 32	多云
0x8f	143	31 34 33	雷阵雨
0x90	144	31 34 34	中到大雨
0x91	145	31 34 35	大到暴雨
0x92	146	31 34 36	晴朗

13.3. 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

14. 更新记录

主要变更	修订日期	修订说明
优化文档	20250926	1. 修正 0x0A 命令模组应答帧描述 2. 修正 0x4A 命令的分布式数据内容格式描述 3. 修正 0x3A 命令的天气请求格式描述 4. 修正 ZSC 模组唤醒 MCU 的引脚描述
增加命令	20250522	1. 增加分布式联动控制 0x4A 命令字支持 2. 增加分布式联动广播 DP 配置 0x4B 命令字支持 3. 产品信息新增可选 s 、 lp 键值及其功能介绍 4. 优化指令文本的准确性
优化文档	20241203	1. 更新附表 2 中的中文天气 UTF-8 编码
优化文档	20241122	1. 修正 0x01 命令的产品信息示例
增加命令	20241112	1. 增加获取模组信息 0x07 命令字支持 2. 增加 Dongle 产测 0x21 ， 0x22 命令字支持 3. 增加时序图 4. 优化指令文本的准确性
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. 优化指令文本的准确性
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
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