串口协议

设备连接 > MCU 开发接入 > Zigbee 通用方案 > 软件开发

文档版本: 20201118

查看在线版本

目录

1 串口通信协议约定 **2**

2 帧格式 **4**

3 帧格式说明 **5**

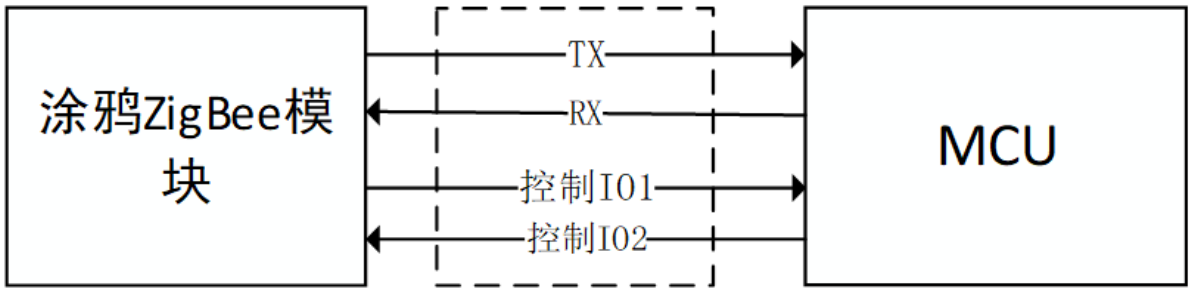
4 协议详述 **9**

4.1 模组查询 MCU 设备类型 9
4.2 查询产品信息 10
4.3 报告模组网络状态 12
4.4 查询模组网络状态 13
4.5 配置 Zigbee 模组 14
4.6 命令下发 16
4.7 状态上报（被动） 18
4.8 状态上报（主动） 20
4.9 Zigbee 模组功能性测试 21
4.10 时间同步 23
4.11 场景开关协议 24
4.12 查询按键信息 24
4.13 场景唤醒命令 26

5 MCU OTA 协议 **28**

5.1 OTA 版本请求的数据格式 28
5.2 OTA 升级通知 29
5.3 OTA 固件内容请求 31
5.4 OTA 固件升级结果上报 32
5.5 MCU 广播数据 33
5.6 MCU 配置 Zigbee 网络策略参数 34

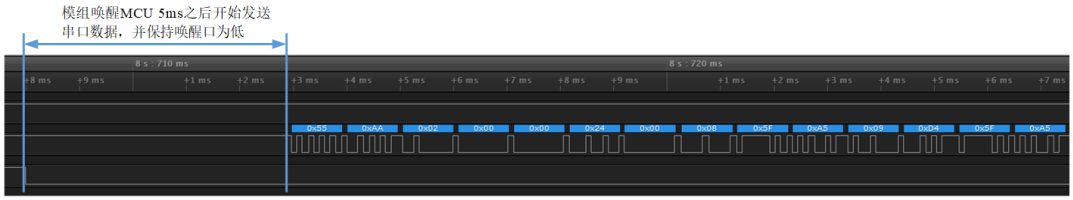
涂鸦 Zigbee 串口通用协议为涂鸦定制的 Zigbee 模组串口通用协议，主要用于涂鸦 Zigbee 模组与其它 MCU 串口直连做串口通信，涂鸦 Zigbee 串口协议结构如下图所示。



{width=400px}

1 串口通信协议约定

- 波特率：9600/115200
- 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
- 休眠模式：
  - 对于带休眠的低功耗设备，Zigbee 模组与 MCU 之间预留 2 个 GPIO 口（PWM1 和 PWM2），作为 MCU 和模组唤醒时使用，唤醒方式为电平触发。Zigbee 模组和 MCU 之间，每次主动发起命令之前，发起方都需要做一次握手连接，具体唤醒参考下图。



  - 对于不带休眠的强电设备，串口处于长监听状态，硬件上不需要连接 I/O1 和 I/O2。
- MCU 唤醒模组：拉低之后可延时 1~5 ms 发送数据，只要保持唤醒口持续低电平，模组会一直处于唤醒状态，当拉高之后，模组会在约 300~550ms 之后进入休眠，减少不必要的唤醒时间，以降低功耗。

> **说明**：固件中有脉冲唤醒的方式，即 mcu 端每次发送串口数据之前都需要先在唤醒口上给一个低脉冲，时间为 1~5ms，然后再发送串口数据。考虑到长时间拉低唤醒，模组功耗偏高的情况，优化为脉冲方式唤醒模组。

```
set_gpio_low();
delay(1);
set_gpio_high();
uart_send_buffer();
```

2 帧格式

涂鸦 Zigbee 模组与 MCU 之间的 UART 通信数据帧由帧头（Front），版本（Ver），命令字（Cmd），数据长度（Length），数据（Data）和校验和（Check）组成，定义和描述如下所示：

Octets：2	1	2	1	2	Variable	1
Front	Ver	Seq	Cmd	Length	Data	Check

3 帧格式说明

帧格式说明如下表所示：

字段	说明
帧头（Front）	2 个字节的前导符，固定为 0x55aa
版本（Ver）	串口通信协议版本，升级扩展用
序列号（seq）	传输数据序列号，范围 0-0xfff0，到达 0xfff0 之后重新回到 0
命令字（Cmd）	具体帧类型，参考文章下方表 2
数据长度（Length）	传输的有效数据长度，length 的长度值为 62 字节
数据（Data）	传输的有效数据
校验和（Check）	数据校验，从帧头开始按字节求和得出的结果对 256 求余

> **注意**：帧中的数据长度（Length）由 Zigbee 模组单个空中数据包的长度决定，涂鸦会对 Zigbee 空中数据格式重新封装，目前可以使用的数据大小为 62 字节。

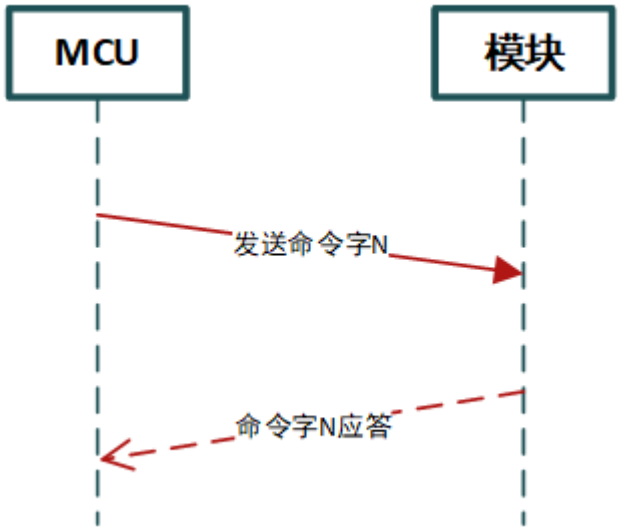
Cmd 描述如下表所示：

Cmd ID	说明
0x01	产品信息查询/上报
0x02	设备状态查询/上报
0x03	Zigbee 设备重置
0x04	命令下发
0x05	状态上报
0x06	状态查询
0x07	reserved

Cmd ID	说明
0x08	Zigbee 设备功能测试
0x09	查询按键信息（仅场景开关类设备有效）
0x0A	场景唤醒命令（仅场景开关类设备有效）
0x24	时间同步
0x25	模组查询 MCU 设备类型

所有大于 1 个字节的数据均采用大端模式传输。

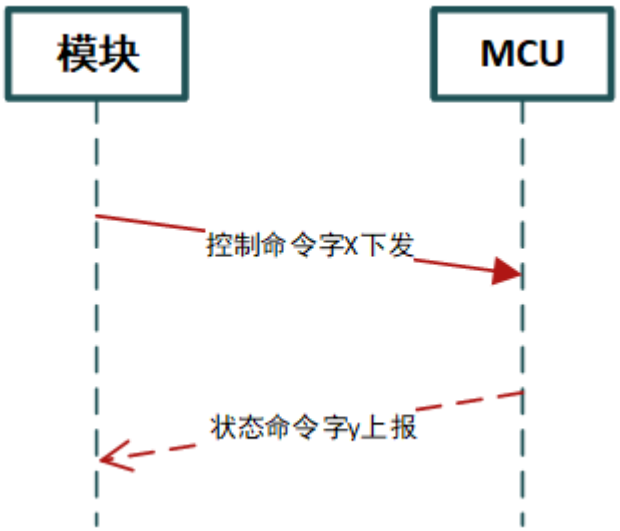
一般情况下，采用命令字一发一收同步机制，即一方发出命令，另一方应答，若发送方超时未收到正确的响应包，则传输超时，如下图所示：



> **说明**：具体通信方式以“协议详述”章节中为准。

模组控制命令下发及 MCU 状态上报则采用异步模式，假设模组控制命令下发“命令字”为 x，MCU 状态上报“命令字”为 y，如下所示：

- 模组控制命令下发：



模组命令下发处理流程

模组通过 04 指令下发命令，内容为可下发的 DP 数据；

Mcu 收到 04 指令之后，需要回复 04 指令，表示串口接收到该命令；

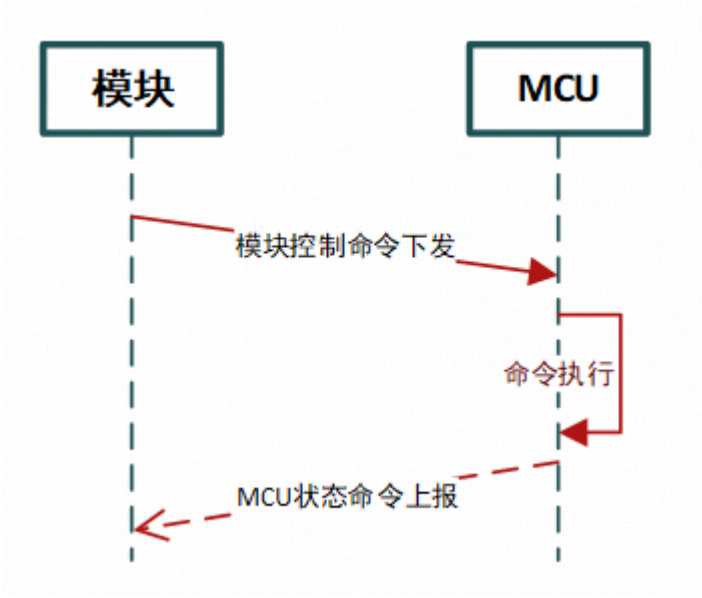
Mcu 再通过 05 指令将执行的结果上报给云端；

05 指令的 seq 和 04 指令保持一致。

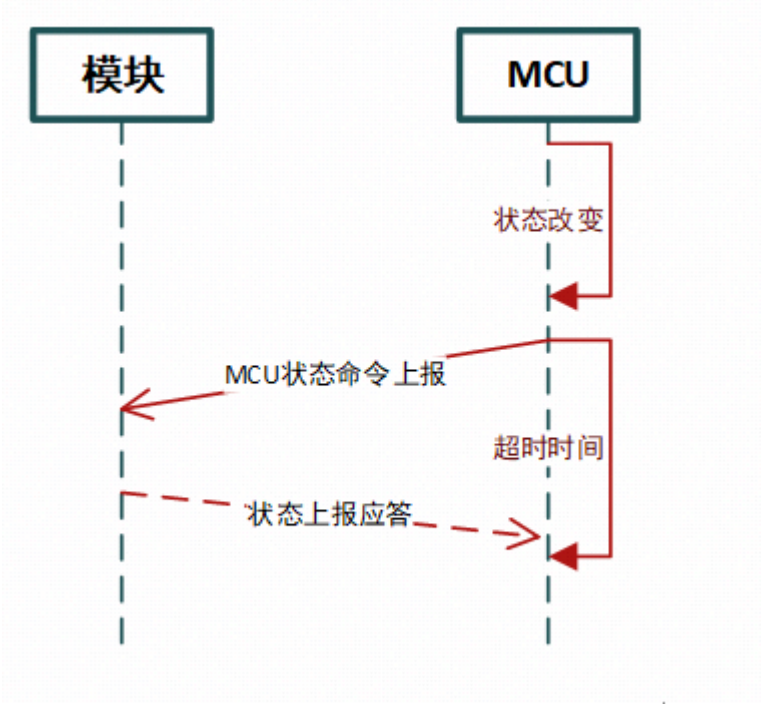
- MCU 状态上报：

  MCU 状态上报分为被动上报和主动上报两种情况；

  - 被动上报：由模组端发送数据命令给 MCU，MCU 执行之后将状态返回；



- 主动上报：MCU 端状态发生改变（物理操作或者断电重启等），主动将当前状态上报到模组；MCU 主动上报为异步操作，在超时时间内没有收到状态上报应答帧，或者收到的应答帧里状态不成功，MCU 端必须进行重传。



4 协议详述

4.1 模组查询 MCU 设备类型

上电之后模组用来查询 MCU 的设备类型，查询成功之后，模组会保存当前设备类型，以后不会再开启查询。

> **说明**：新增功能，需要客户测试模组中的固件是否支持该功能。

当接收到 MCU 的应答之后，Zigbee 模组将重新启动，载入完成参数之后，继续和模组进行数据交互。

模组上电之后先以 9600 波特率发送查询指令，如果没有收到 MCU 应答，则使用 115200 波特率进行检测。

> **注意**：
>
> 检测过程中，全部按照低功耗设备处理，先在模组唤醒 MCU 的 I/O 口发送一个 50ms 的低脉冲，再发送串口数据，可以保证 MCU 收到数据。
>
> 之前已经发布的 mcu 固件不需要进行修改，可以继续使用最新固件，新产品接入，需要实现这个协议。

模组发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x25
数据长度	2	0x0000
数据	0	0
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

MCU 返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x25
数据长度	2	0x0001
数据	1	01 强电类对接设备 02 低功耗设备 03 强电场景面板
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

4.2 查询产品信息

- 产品信息由 product ID、MCU 软件版本构成。
- product ID：对应涂鸦开发者平台 PID (产品标识)，在创建产品时由涂鸦开发者平台自动生成，用于云端记录产品相关信息。
- MCU 软件版本号格式定义：采用点分十进制形式，”x.x.x”，x 为十进制数。
- 当模组复位后，会主动查询，如果 MCU 没有回复，或者回复内容有误，将会间隔 5 秒重复查询。

> **注意**：OTA 相关命令用单字节表示 MCU 版本时，最大版本由于字节长度限制，最大版本号可到 3.3.15。

模组发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x01

字段	长度（byte）	说明
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 01 0000 xx

MCU 返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x01
数据长度	2	N
数据	N	{“p”:”AIp08kLI”, “v”:”1.0.0” }
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

{“p”:” AIp08kLI”,“v”:”2.0.0” }

p 表示产品 ID 为 AIp08kLI

v 表示 MCU 版本为 2.0.0

0x55aa 02 N 01 00 1c 7b2270223a2241497031386b4c49222c2276223a22312e302e30227d xx

4.3 报告模组网络状态

设备状态 ID	描述
0x00	设备为未入网状态
0x01	设备为已入网状态
0x02	设备网络状态异常
0x03	设备为配网中状态

- 设备未入网状态：设备第一次上电、或者入网失败、或者离网的情况下，设备状态为未入网状态；并将该状态下发至 MCU 。
- 设备为已入网状态：设备入网成功之后，设备状态为已入网状态；并将该状态下发至 MCU 。
- 当模组的网络状态发生变化，则主动下发模组网络状态至 MCU 。
- 网络状态是指 Zigbee 的网络的状态，当模组配网成功之后，即设备已加入网络，不因网关断电，父节点丢失等原因变更网络状态。

模组发送

字段	长度（byte）	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	0x0001
数据	1	指示模组工作状态：0x00：状态 1 0x01：状态 2
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 02 0001 00 xx

MCU 返回

字段	长度（byte）	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	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 02 0000 xx

4.4 查询模组网络状态

新增功能，MCU 可以查询 Zigbee 当前网络状态，MCU 端如需增加此功能，请测试使用的模组版本是否支持该功能。

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x20
数据长度	2	0x0000

字段	长度（byte）	说明
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 20 0000 xx

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x20
数据长度	2	0x0001
数据	1	网络状态: 参见网络状态表
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 20 0001 xx xx

4.5 配置 Zigbee 模组

配置 Zigbee 模组命令分为两种，如下所示；

命令	说明
0x00	将模组软件复位

命令	说明
0x01	将模组配置为开始配网状态（先离网再配网）

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x03
数据长度	2	0x0001
数据	0	0x00/0x01
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 03 0001 01 xx

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x03
数据长度	2	0x0000
数据	0	无

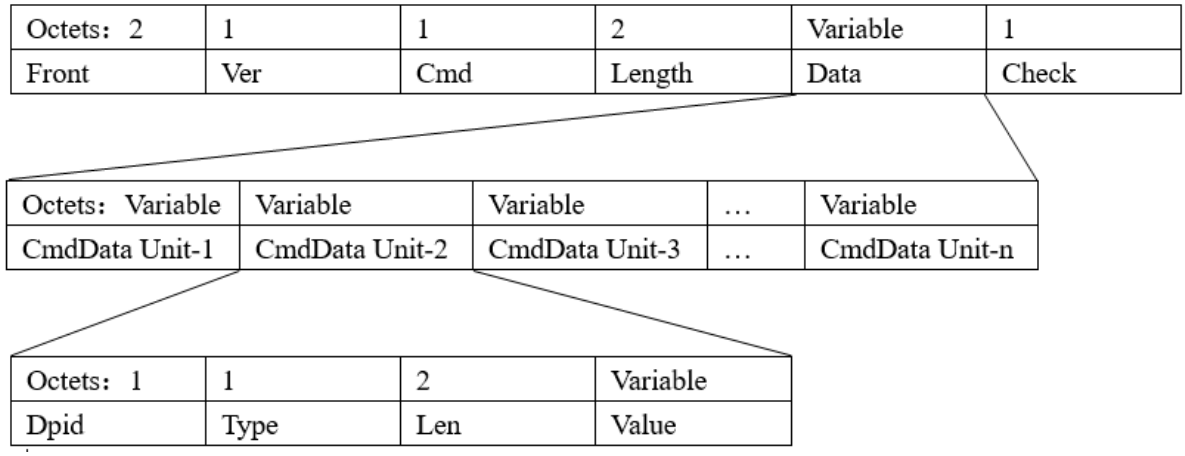
字段	长度（byte）	说明
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 03 0000 xx

4.6 命令下发

- 命令下发帧格式：



- datapoint 命令/状态数据单元如下所示：

数据段	长度（byte）	说明
Dpid	1	datapoint 序号
Type	1	对应开放平台上某 datapoint 具体的数据类型，通过如下“表示值”标识
类型	表示值	长度（字节） 说明

数据段	长度（byte）	说明	
raw	0x00	N	对应于 raw 型 datapoint（模组透传）
bool	0x01	1	value 范围：0x00/0x01
value	0x02	4	对应 int 类型，大端表示
string	0x03	N	对应于具体字符串
enum	0x04	1	枚举类型，范围 0-255
bitmap	0x05	1/2/4	长度大于 1 字节时，大端表示
Len	2	长度对应 value 的字节数	
Value	1/2/4/N	hex 表示，大于 1 字节采用大端传输	

- datapoint 命令/状态数据单元除”raw” 类型外，其他类型均属于 “obj” 型 datapoint。
- “命令下发” 可包含多个 datapoint “命令数据单元”，raw 类型只能单条命令下发。
- “命令下发” 为异步处理协议，对应于 MCU 的 datapoint “状态上报”。

模组发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x04

字段	长度（byte）	说明
数据长度	2	取决于“命令数据单元”类型以及个数
数据	N	参考 DP 格式
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

系统开关对应 3 号 DP, 使用 bool 型变量，开机数值为 1

0x55aa 02 N 04 0005 **03 01 0001 01** xx

4.7 状态上报（被动）

- 当 MCU 收到模组端下发的命令，并执行相应动作之后，需要将新的状态被动上报给模组端；

  状态正确执行之后，只上报执行了操作的 datapoint 状态；
- “状态上报（被动）”为同步处理协议，模组端收到 datapoint 信息之后会立即返回 ACK 给 MCU
- “状态上报（被动）”可包含多个“obj”型 datapoint“命令数据单元”，datapoint 状态数据单元说明详见“DP 格式表”。
- “raw”类型数据不能和其他数据一起上报。

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x05

字段	长度（byte）	说明
数据长度	2	取决于“状态数据单元”类型以及个数
数据	N	参考命令下发的 DP 格式
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

湿度对应 5 号 DP, 使用 value 型变量，湿度为 30℃

0x55aa 02 N 05 00 08 **05 02 0004 0000001e** xx

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x05
数据长度	2	0x0001
数据	0	0x00/0x01 0x00：状态上报失败 0x01：状态上报成功
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

模组返回 ACK 状态成功给 MCU

0x55aa 02 N 05 0001 01 xx

4.8 状态上报（主动）

- MCU 主动检测到 datapoint 有变化，或者 MCU 重启等情况下，需要将变化后的 datapoint 状态发送至模组。
  - 正常变化时，只上报有变化的 datapoint；
  - 重启等异常情况下，需要上报所有的 datapoint；
- “状态上报（主动）” 为异步处理协议，模组端在超时时间内收到网关的 response 之后，会将状态返回给 MCU 端；如果状态返回超时，或者返回状态为 fail，MCU 需要做随机退避的重传机制。
- “状态上报（主动）” 可包含多个 “obj” 型 datapoint “命令数据单元”，datapoint 状态数据单元说明详见 “ DP 格式表”。
- “raw” 类型数据不能和其他数据一起上报。
- 如果需要在配网成功之后，上报 DP 数据用于同步 app 面板，最好增加一定的延时上报，5 秒为宜。

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x06
数据长度	2	取决于 “状态数据单元” 类型以及个数
数据	N	DP 格式表
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

湿度对应 5 号 DP, 使用 value 型变量，湿度为 30℃

0x55aa 02 N 06 08 **05 02 0004 0000001e** xx

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x06
数据长度	2	0x0001
数据	0	0x00/0x01 0x00: 状态上报失败 0x01: 状态上报成功
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

模组返回 ACK 状态成功给 MCU

0x55aa 02 N 06 0001 01 xx

4.9 Zigbee 模组功能性测试

扫描指定信道的 RSSI 值，返回扫描结果和信号强度百分比；该命令必须在设备未配网情况下才可正常运行，单次测试完成之后必须重启模组。

> **注意**：默认使用 11 信道，MCU 发送时，直接选择 11 信道即可。

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N

字段	长度（byte）	说明
命令字	1	0x08
数据长度	2	0x0001
数据	Data	信道值（11-26）
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

MCU 要求模组扫描 11 信道的 RSSI 值

0x55aa 02 N 08 0001 0b xx

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x08
数据长度	2	0x0002
数据	2	数据长度为 2 字节： Data[0]: 0x00 失败, 0x01 成功
		当 Data[0] 为 0x01，即成功时，Data[1] 表示信号强度 (0-100, 0 信号最差，100 信号最强)

字段	长度（byte）	说明
		当 Data[0] 为 0x00，即失败时，Data[1] 为 0x00 表示未在指定信道扫描到 RSSI，Data[1] 为 0x01 表示模组未烧录授权 key
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

信道有效值为（11-26），无效信道会默认使用 11 信道

0x55aa 02 N 08 0002 01 64 xx

4.10 时间同步

时间同步用来 MCU 同步网关的网络时间使用。

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x24
数据长度	2	0x0000
数据	Data	NA
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x024
数据长度	2	0x0002
数据	2	数据长度为 8 字节的时间值，格式参考如下时间同步数据格式
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

时间同步的数据格式，包含标准时间戳和本地时间戳。

Variable	Variable
标准时间戳 (4 byte)	本地时间戳 (4 byte)

- 标准时间戳为格林威治时间 1970 年 01 月 01 日 00 时 00 分 00 秒起至现在的总秒数
- 本地时间戳为标准时间戳 + 标准时间和本地时间相差的秒数（包含时区和夏令时）

4.11 场景开关协议

在场景开关设备中，MCU 只需要通过串口透传协议告知 Zigbee 模组设备有几个按键，以及当前操作的是哪个按键即可。

4.12 查询按键信息

在模组重启之后会发送查询按键信息。

目前最多支持 10 个按键，即 10 个场景，仅作为场景面板使用，不支持其他 DP 和自定义功能 DP。

创建产品时，需要选择的 DP 为场景 ID 组和场景编号 (1~10)。场景编号必须从小到大，从场景 1 开始，直到最多按键个数。

模组发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x09
数据长度	2	0x0000
数据	Data	NA
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

模组要求获取 MCU 的按键总数值

0x55aa 02 N 09 0000 xx

MCU 返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x09
数据长度	2	0x0001
数据	2	面板开关的按键总个数

字段	长度（byte）	说明
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 09 0001 02 xx

4.13 场景唤醒命令

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x0A
数据长度	2	0x0001
数据	Data	按键 ID
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

按键 ID 和按键总个数一一对应，如按键总个数为 4，则按键 ID 依次为 1,2,3,4；

示例

MCU 要求模组执行按键 1 对应的场景：

0x55aa 02 N 0A 0001 01 xx

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x0A
数据长度	2	0x0001
数据	1	0: 场景唤醒失败；1: 场景唤醒成功；
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55aa 02 N 0A 0001 01 xx

> **注意**：模组应答成功，表示该按键在 app 成功绑定了场景，且该场景已经被成功执行，如果回复失败，则认为 app 端没绑定场景，则该场景不会被执行。这里说的场景为场景面板本地发送出的场景。

当按键按下时，场景面板还会向网关发送一个按键值，用于联动云端场景，当 MCU 有按键上报时，即会上报按键给网关，即如果该场景面板仅使用云端场景功能，模组无论回复成功和失败都可以认为 MCU 上报按键成功。

云端场景和本地场景的区别，本地场景即标准的 Zigbee 场景，满足 Zigbee 协议，注意，不是所有的命令都支持本地场景，目前设备端保存的场景的数据为某些属性值，而有些命令不支持的，这些功能需要走云端场景。

云端场景，其本质是云端联动控制，和本地场景只是文字上叫法相似，但区别很大。

5 MCU OTA 协议

OTA 的流程为云端发出 OTA 通知，MCU 接收到通知之后，回复通知，然后开始发起数据请求，数据请求的包大小最大为 50 字节，MCU 发送数据请求之后，模组会将该请求转发给网关，网关根据当前偏移量和数据包大小回复数据。该过程中，有时会出现，网关等待一段时间才回复的情况，因此需要 MCU 端完善数据请求逻辑，需要增加超时机制，当发出数据请求在一段时间内没有回复时，需要重新发送该请求，建议超时时间 3~5 秒，超时超过一定次数再认为 OTA 升级异常，取消 OTA 升级，建议次数 5 次。

5.1 OTA 版本请求的数据格式

若支持 MCU 升级必须实现此命令，网关会主动查询 MCU 版本号，MCU 侧也可主动上报；查询场景：1. 配网成功时 2.MCU 升级过程异常时；主动上报场景：1. 配网成功后（必须添加）2. 升级结束。

模组发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	模组产生
命令字	1	0x0B
数据长度	2	0x0000
数据	0	NA
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55 AA 02 00 f0 0B 00 00 XX

MCU 响应

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	MCU 下发的 SEQ
命令字	1	0x0B
数据长度	2	0x0001
数据	1	版本号（当前版本）（Bits） 01.00.0001 表示 1.0.1
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

55 AA 02 00 39 0B 00 01 40 XX

> **注意**：OTA 相关命令用单字节表示 MCU 版本时，最大版本由于字节长度限制，最大版本号可到 3.3.15。

5.2 OTA 升级通知

模组发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	模组产生
命令字	1	0x0C
数据长度	2	0x0011
数据	8	Data[0]~ Data[7] PID

字段	长度（byte）	说明
数据	1	版本号（升级版本）（Bits） 01.00.0001 表示 1.0.1
数据	4	固件大小最大 256K
数据	4	固件校验和：从固件第一个字节按字节求和得出的结果对 2^32 求余
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55 AA 02 00 1C 0C 00 0F 30 31 32 33 34 35 36 37 40 00 01 00 00 30 31 32 33 XX

MCU 响应

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	模组下发的 seq
命令字	1	0x0C
数据长度	2	0x0001
数据	1	0x00：OK，0x01:error
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55 AA 02 00 1C 0C 00 01 00 XX

5.3 OTA 固件内容请求

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	0x0000
命令字	1	0x0D
数据长度	2	0x000E
数据	8	PID
数据	1	版本号（升级版本）（Bits）01.00.0001 表示 1.0.1
数据	4	数据包的偏移量 (固件的位置)
数据	1	数据包的大小（最大 50 字节）
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55 AA 02 00 00 0D 00 0E 30 31 32 33 34 35 36 37 40 00 00 00 01 32 XX

> **注意：**每次拉去的数据包大小最大为 50 字节。

模组响应

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	0x0000

字段	长度（byte）	说明
命令字	1	0x0D
数据长度	2	0x0006+N
数据	1	Status 0：成功 1：失败
数据	8	PID
数据	1	版本号（升级版本）（Bits） 01.00.0001 表示 1.0.1
数据	4	数据包的偏移量 (固件的位置)
数据	N	数据
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

55 AA 02 00 39 0D XXXX 00 30 31 32 33 34 35 36 37 40 00 00 00 01 XX

5.4 OTA 固件升级结果上报

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	MCU 下发的 SEQ
命令字	1	0x0E
数据长度	2	0x000A
数据	1	Status 0：成功 1：失败
数据	8	PID

字段	长度（byte）	说明
数据	1	版本号（升级后的当前版本）（Bits）01.00.0001 表示 1.0.1
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55 AA 03 00 f0 0E 00 0A 00 30 31 32 33 34 35 36 37 40 26

模组响应

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
业务序列号（Seq）	2	MCU 下发的 seq
命令字	1	0x0E
数据长度	2	0x0001
数据	1	0x00：OK，0x01：error
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

示例

0x55 AA 02 00 1C 0E 00 01 00 XX

5.5 MCU 广播数据

MCU 端需要将数据进行全网通知时，使用该帧数据，注意广播之间需要有一定的时间间隔，间隔由网络的规模决定。该数据可以让全网络中的设备接收到，有个有低功耗设备，其需要处于周期唤醒的状态，且唤醒周期需要小于广播周期，不然还没有唤醒，下一条广播数据就会将之

前的广播数据覆盖。

MCU 发送

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x27
数据长度	2	N
	N	DP 格式表
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x027
数据长度	2	0x0001
数据	1	00 上报失败 01 上报成功
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

5.6 MCU 配置 Zigbee 网络策略参数

该指令可以在接收到模组发送的查询 pid 帧之后，进行 ms 级延时之后发送。

> **注意**：该命令接收完成并回复成功应答之后，模组将会执行重启。

- 心跳时间

  心跳时间是用来维护设备和网关之间的数据链路是否正常的手段，强电设备的心跳时间默认为 150+random（30）秒，低功耗设备的心跳时间默认为 4 小时，且网关判定 12 小时内没有收到心跳则认为设备离线，心跳时间修改仅支持低功耗设备的心跳时间。

- 超时时间

  当 MCU 发送配网指令之后，模组会开启一段时间的配网，并发送当前网络状态为配网状态，当在这一段时间内由于某些原因，例如附近没有开启配网的网络，或者距离较远的原因导致模组没有加入到合适网络，则配网超时，配网超时之后，模组将处于未配网状态，同时也会将此状态发送给 MCU。

- 轮询（Poll）

  Poll 周期是指已经加入到网络的低功耗模组会周期内唤醒，其唤醒之后会发送 data request 给其父节点，用于告知父节点，其当前处于唤醒状态，父节点是否为其缓存数据，如果有缓存数据则父节点可以将数据发送给它。

  Poll 是用于接收父节点的数据，即设备对控制的实时性要求很高，例如单火开关，可以将这个值设置为 250ms，其他产品，例如传感器，只有当状态发生变化时，或者周期上报，即只需要上报数据时，就可以把 poll 关闭。模组将收不到网关下发的控制指令，因此一般是在上电之后，设置一段时间的快速 poll，可以在这个时间窗内将网关的配置命令下发下来。上电之后的快速 poll 的时间默认为 30 秒，支持 MCU 设置。如果设备需要关闭 poll，且有网关的配置需要下发，建议将快速 poll 的时间窗增加。

  当关闭 poll 时，网关会帮忙缓存数据，当模组上报数据时，会携带 data request，此时网关会将数据发下给模组。

  > **注意**：该值主要是影响功耗，唤醒周期越短，功耗越大。poll 最小值为 200ms，小于最小值按照最小值处理。最大值建议 8s，如果设置为 0 则关闭 poll。

- 重连（Rejoin）

  模组发送 data request 之后，父节点首先是需要回复 ack，该 ack 是对 data request 的应答，然后如果有缓存数据则将数据发送给模组。如果没有数据发送，则仅需回复 ack。如果模组发送了 data request，但由于环境、距离、父节点断电等因素导致模组没有收到 ack，则模组的 poll 失败次数会加 1，如果在累加的过程中重新收到父节点的 ack，则累加清零，当累加到的一定的值时（Poll 失败次数），认为模组丢失父节点，需要触发 rejion。

  触发 rejion，目前由 2 种方式，应用层有数据发送则触发 rejion 或者定时触发 rejion，即当模组处于离线状态，周期性开始触发 rejion，直到模组重新加入到之前网络。两者目

前独立。Rejion 成功，只能表示其和父节点能够正常通信，数据能否到达网关需要看网关和父节点的路由情况。目前这两个参数都可以由 MCU 灵活配置。

rejion 即重新加入到网络，这里不是指配网，无须网关开启配网，是一种专门用于低功耗设备在父节点丢失时，重新加入的网络的一种机制。

rejoin 间隔时间：即上面提到的周期触发 rejion 的时间间隔，对应一些对数据要求严格的场合，或者功耗满足要求的场景，可以将 rejion 间隔设置的短一些例如 3~5 秒。对于一些传感设备或者通过数据上报触发的场景，可以将时间间隔设置久一些，例如 1 小时。

Rejoin 尝试次数，是指设备触发 rejion 之后，模组可以发送多少次 rejion，对于 poll 时间较短的场合，以及 rejion 间隔短的应用，可以设置少些，例如 1~2 次，对于 rejion 间隔较长的场合，可以将 rejion 尝试次数稍微增加，时例如 3~4 次。

> **注意**：设置参数不在范围内的，参数值不变化。

MCU 发送

字段

```
<th><b>长度</b></th>
<th><b>说明</b></th>
```

帧头

2

```
<td><b>0x55aa</b></td>
```

版本

1

```
1       <td>0x02</td>
```

序列号

2

```
1       <td>N</td>
```

命令字

1

```
1       <td>0x26</td>
```

```
1  <tr>
```

数据长度

2

```
1       <td>0x0e</td>
```

```
1  <tr>
```

数据

```
<tr>
    <td>2</td>
    <td>心跳时间（秒）设置为0xffff表示当前值，0xfffe表示默认值
```

只有低功耗设备支持心跳修改，低功耗设备的心跳默认为 4 小时，设置范围为（10~5*3600 秒）

```
</tr>
<tr>
    <td>2</td>
    <td>配网超时时间，设置为0xffff表示当前值，0xfffe表示默认值，配网超
        时时间默认为180秒，
```

设置范围为（30~600）

```
</tr>
    <tr>
    <td>2</td>
    <td>Rejoin间隔时间（秒）
```

设置 0xffff 表示当前值设置 0xfffe 表示默认值范围为（3~3600）默认值为 180 秒，即当设备丢失父节点时，会间隔 180 秒尝试 rejion

```
</tr>
        <tr>
    <td>2</td>
    <td>Poll 时间（唤醒周期）
```

单位 ms 0 表示关闭 poll，0xffff 表示当前值，0xfffe 表示默认为 5000ms 范围（200~10000）模组会间隔 poll 时间唤醒一次，用于确认父节点是否有数据发送给它，如果产品为传感类型，仅有数据上传，则可以将其设置为 0

```
</tr>
            <tr>
    <td>2</td>
    <td>Poll 上电之后持续快速poll的时间段设置（秒），快速poll的时间为
        250ms，当时间到达时，按照poll设置的时间运行
```

默认为 30 秒，设置为 0xffff 表示当前值，0xfffe 表示默认值。范围：（10~3000）

```
</tr>
                    <tr>
    <td>1</td>
    <td>Poll失败次数
```

设置为 0xff，为当前值 0xfe 为默认值 4 次范围为（3 ~ 40）当达到最大值时，如果配置了时间触发 rejion，则到规定的时间则会触发设备 rejion

```
</tr>
 <tr>
    <td>1</td>
    <td>应用数据发送是否触发rejion
```

设置为 0xff 为当前值设置为 0xfe 为默认值 0 表示不触发 1 表示触发默认值为 1

```
</tr>
     <tr>
    <td>1</td>
    <td>Rejoin尝试次数
```

设置为 0xff 为当前值设置为 0xfe 为默认值默认值为 1 范围（1~10）

```
</tr>
    </tr>
     <tr>
    <td>1</td>
    <td>发射功率设置
```

设置为 0xff 当前值设置为 0xfe 默认值默认值为 11 范围（3~19）dB

校验和

1

```
    <td>从帧头开始按字节求和得出的结果对 256 求余</td>
```

模组返回

字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x02
序列号	2	N
命令字	1	0x026
数据长度	2	0x0001
数据	1	00 失败 01 成功
校验和	1	从帧头开始按字节求和得出的结果对 256 求余