TI ZigBee 协调器串口命令

版本	日期	功能描述
V1.0	20210122	协议文档第一版
v1.1	20210123	完善人体红外，增加麦乐克光感，涂鸦电机
v1.2	20210125	补充 OTA 升级部分的协议
v1.3	20210126	1，增加目录 2，增加 ZDO 指令绑定，解绑定，查询绑定 3，增加 OTA 状态属性描述
v1.4	20210126	更正错误，增加目录描述信息
v1.5	20210130	✧ 增加绑定，解绑定，查询绑定命令。 ✧ 增加自动绑定通知，获取设备失败通知。
v1.6	20210401	1，收到 OTA 完成通知，附带 OTA 信息。 2，增加激活码输入命令
V1.7	20210418	1，属性表中增加访问方式项
V1.8	20210522	1，新增查询属性命令 2，新增版本日期的属性
V1.9	20210813	1，增加白名单和 Install Code 导入
V1.10	20210820	1，增加激活返回“重复激活”状态
V1.11	20210909	1，可发送 Default Response。 2，ZCL 命令的地址模式字段升级成接口控制字段，增加了多 Profile 的支持。
V1.12	20210911	ZDO 命令增加超时返回消息
V1.12.1	20210912	增加信标信道扫描
V1.13.0	20210930	增加流程框图
V2.0	20211007	增加自动识别设备框图
V2.1	20211012	增加查询协调器缓存地址表命令
V2.2	20211020	增加读写 NV 的指令
V2.3	20211101	增加查询协调器直连 End Device 设备指令
V2.4	20211119	收到 ZCL 命令，增加回复 Default Response 标志，详见“4.1.1《接口控制表》”

✧

✧

1. 串口命令格式 6

1.1. 硬件参数 6

1.2. 协调器模块的功能特点 6

1.2.1. 优化的串口命令流程 6

1.2.2. 自动化的设备入网识别 7

1.3. 串口命令格式 8
1.3.1. 命令帧格式以及解析 8
1.3.2. ACK 命令帧（ACK 状态表） 9
2. 协调器配置命令与协调器异步状态命令 10
2.1. 协调器配置命令，命令大类 0x00，下行 10
2.1.1. 查询协调器状态 10
2.1.2. 优选信道 11
2.1.3. 启动网络/允许入网 11
2.1.4. 扫描信标信道 12
2.1.5. 协调器导入白名单 12
2.1.6. 协调器重启/恢复出厂 13
2.1.7. 关闭允许入网 14
2.1.8. 设置协调器 UTC 时间 14
2.1.9. 查询协调器 UTC 时间 14
2.1.10. 查询协调器缓存地址表 15
2.1.11. 协调器加组 15
2.1.12. 协调器删组 16
2.1.13. 协调器查组 16
2.1.14. 增加动态 Profile 和 Endpoint 17
2.1.15. 协调器软启动 17
2.1.16. 查询直连协调器的 End Device 的短地址 18
2.1.17. 激活协调器 18
2.1.18. 读取 NV 数据 19
2.1.19. 强行写 NV 19
2.1.20. 初始化 NV 20
2.2. 协调器异步状态命令，命令大类 0x80，上行 20
2.2.1. 协调器启动 21
2.2.2. 允许入网时间 21
2.2.3. 网络启动状态 21
2.2.4. 扫描结果 22

2.2.5. 设备入网消息 23
2.2.6. 设备功能消息（Device ID 表和 Cluster ID 表） 23
2.2.7. 设备注销消息 26
2.2.8. 设备离网消息 27
2.2.9. 设备入网过程 27
2.2.10. 设备自动绑定通知 28
2.2.11. 设备功能获取失败通知 28
3. ZDO 命令 28
3.1. ZDO 命令的帧格式 29
3.1.1. 下发 ZDO 命令，命令大类 0x01，下行 29
3.1.2. 下发 ZDO 命令反馈，命令大类 0x01，上行 30
3.1.3. 收到 ZDO 命令，命令大类 0x81，上行 30
3.1.4. ZDO 命令表 30
3.1.5. ZDO 状态表 31
3.1.6. 无线通信状态表 31
3.2. ZDO 命令解析 32
3.2.1. NWK_addr_req 和 NWK_addr_rsp，查询短地址 32
3.2.2. IEEE_addr_req 和 IEEE_addr_rsp，查询 IEEE 地址 32
3.2.3. Simple_Desc_req 和 Simple_Desc_rsp，查询端口信息 33
3.2.4. Acitve_EP_req 和 Acitve_EP_rsp，查询设备端口数 34
3.2.5. Bind_req 和 Bind_rsp，设置设备与设备绑定 34
3.2.6. Unind_req 和 Unind_rsp，解除设备之间的绑定 35
3.2.7. Mgmt_Bind_req 和 Mgmt_Bind_rsp，查看设备的绑定表 35
3.2.8. Mgmt_Leave_req 和 Mgmt_Leave_rsp，主动删除设备 36
4. ZCL 通用命令 36
4.1. ZCL 通用命令帧格式 37
4.1.1. 下发 ZCL 通用命令，命令大类 0x02，下行 37
4.1.2. 下发 ZCL 通用命令反馈，命令大类 0x02，上行 38
4.1.3. 收到 ZCL 通用命令，命令大类 0x82，上行 39
4.1.4. ZCL 通用命令表 39

4.1.5. ZCL 属性数据类型表 39
4.1.6. ZCL 状态表 41
4.1.7. 常见状态属性表 42
4.2. ZCL 通用命令解析 43
4.2.1. ZCL_CMD_READ 与 ZCL_CMD_READ_RSP 43
4.2.2. ZCL_CMD_WRITE 与 ZCL_CMD_WRITE_RSP 44
4.2.3. ZCL_CMD_CONFIG_REPORT 与 ZCL_CMD_CONFIG_REPORT_RSP 45
4.2.4. ZCL_CMD_READ_REPORT_CFG 与 ZCL_CMD_READ_REPORT_CFG_RSP 46
4.2.5. ZCL_CMD_REPORT 46
4.2.6. ZCL_CMD_DEFAULT_RSP（收发均为此结构） 47
4.2.7. ZCL_CMD_DISCOVER_ATTRS 与 ZCL_CMD_DISCOVER_ATTRS_RSP 47
5. ZCL 专有命令 48
5.1. ZCL 专有命令帧结构 49
5.1.1. 下发 ZCL 专有命令，命令大类 0x03，下行 49
5.1.2. 下发 ZCL 专有命令反馈，命令大类 0x03，上行 50
5.1.3. 收到 ZCL 专有命令，命令大类 0x83，上行 50
5.1.4. ZCL 专有命令相关表格 51
5.2. 标记类命令，cluster=0x0003 52
5.2.1. 标记类的状态属性 52
5.2.2. IDENTIFY，设置设备标记 53
5.2.3. IDENTIFY_QUERY 和 IDENTIFY_QUERY_RSP，查询标记 53
5.3. 分组类命令，cluster=0x0004 53
5.3.1. 分组类的状态属性 53
5.3.2. GROUP_ADD 与 GROUP_ADD_RSP，设备加组 53
5.3.3. GROUP_VIEW 与 GROUP_VIEW_RSP，查询组信息 54
5.3.4. GROUP_GET_MEMBERSHIP 与 GROUP_GET_MEMBERSHIP，查询设备加入的组 54
5.3.5. GROUP_REMOVE 与 GROUP_REMOVE_RSP，删除设备指定分组。 55
5.3.6. GROUP_REMOVE_ALL，删除全部分组 56
5.4. 场景类命令，cluster=0x0005 56

5.4.1. 场景类的状态属性 56
5.4.2. SCENE_ADD 与 SCENE_ADD_RSP，添加场景 56
5.4.3. SCENE_VIEW 与 SCENE_VIEW_RSP，查询场景内容 57
5.4.4. SCENE_REMOVE 与 SCENE_REMOVE_RSP，删除场景 57
5.4.5. SCENE_REMOVE_ALL 与 SCENE_REMOVE_ALL_RSP，删除所有场景 58
5.4.6. SCENE_STORE 与 SCENE_STORE_RSP，保存场景 58
5.4.7. SCENE_RECALL，执行场景 58
5.4.8. SCENE_GET_MEMBERSHIP 与 SCENE_GET_MEMBERSHIP_RSP，查询所有保存场景 59
5.5. 开关类命令，cluster = 0x0006 59
5.5.1. 开关类状态属性 59
5.5.2. OFF，ON 和 TOGGLE，开关控制 59
5.6. 亮度类命令，cluster = 0x0008 60
5.6.1. 亮度类状态属性 60
5.6.2. MOVE_TO_LEVEL 与 MOVE_TO_LEVEL_ONOFF，调至亮度 60
5.6.3. MOVE 与 MOVE_ONOFF，自动变化亮度 61
5.6.4. STEP 与 STEP_ONOFF，单步亮度 61
5.6.5. STOP 与 STOP_ONOFF，停止渐变 62
5.7. 安防类命令，cluster = 0x0500 62
5.7.1. 安防类状态属性 62
5.7.2. STATUS_CHANGE_NOTIFICATION，报警状态通知 63
5.7.3. STATUS_ENROLL_REQUEST，安防注册请求 64
5.7.4. STATUS_ENROLL_RESPONSE，安防注册响应 64
5.8. 光感类命令，cluster = 0x0400 64
5.8.1. 光感类状态属性 64
5.9. 涂鸦自定义命令，cluster = 0xEF00 65
5.9.1. 涂鸦自定义状态属性 65
5.9.2. TUYA_RX，协调器发给涂鸦模块的命令 65
5.9.3. TUYA_TX，涂鸦模块发给协调器的命令 65
5.9.4. 涂鸦数据包解析，窗帘电机部分 65

6. OTA 类命令 67
6.1. OTA 命令帧格式 67
6.1.1. 下发 OTA 命令，命令大类 0x04 68
6.1.2. 收到 OTA 命令，命令大类 0x84 68
6.1.3. OTA 命令表 68
6.1.4. OTA 相关状态属性，cluster = 0x0019 69
6.2. OTA 命令解析 69
6.2.1. 版本通知命令，命令 ID=0x00 69
6.2.2. 开始升级命令（请求开始升级与回复开始升级），命令 ID=0x01 70
6.2.3. 文件传输命令（请求文件传输和回复文件传输），命令 ID=0x02 70
6.2.4. 升级完成通知，命令 ID=0x03 71

✧
✧
✧

1. 串口命令格式

1.1. 硬件参数

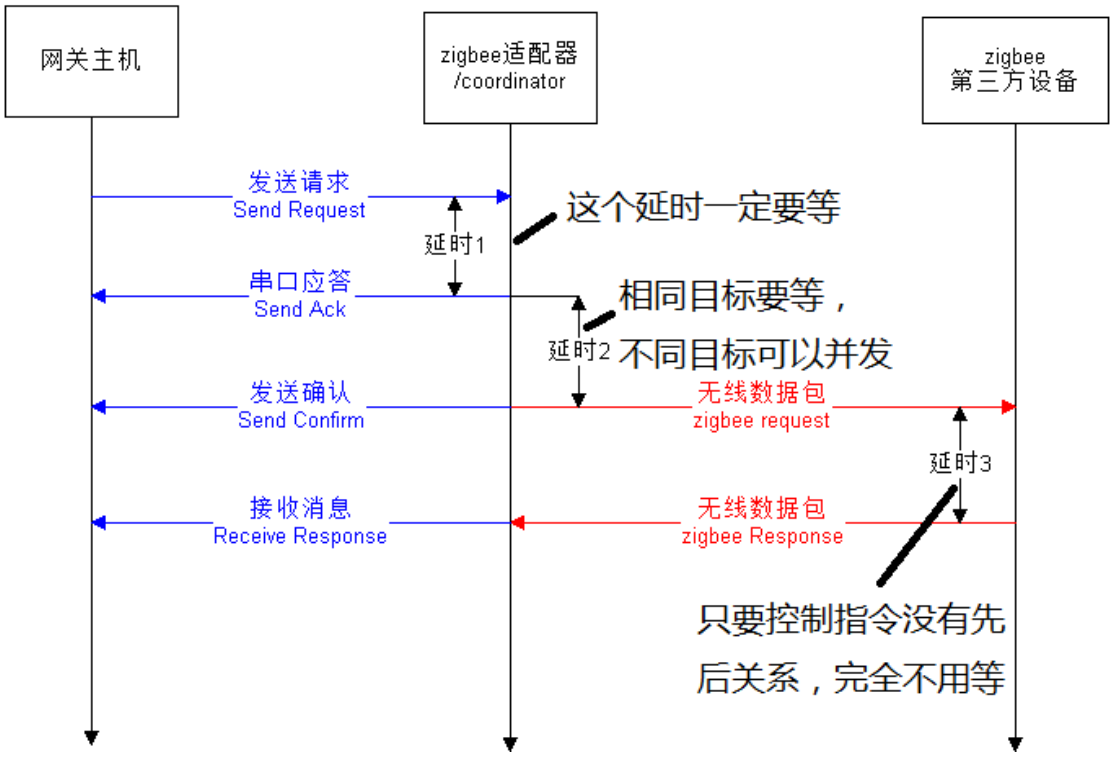
供电：3.3V
波特率：230400
IO 口定义： DIO12 UART_RX
DIO13 UART_TX
DIO15 BOOTLOADER

注意事项：

- ✧ 串口协议中所有参数均采用小端模式。
- ✧ 协调器有防连包机制，上位机发给协调器的 2 条命令连在一起不受影响。
- ✧ 协调器有命令超时机制，上位机发给协调器的命令发生断包，不影响后续的命令。
- ✧ 协调器主动发给上位机的命令，有 5 字节的间隔时间，按照 460800 波特率计算每条命令之间约 0.1ms 的最小间隔。

1.2. 协调器模块的功能特点

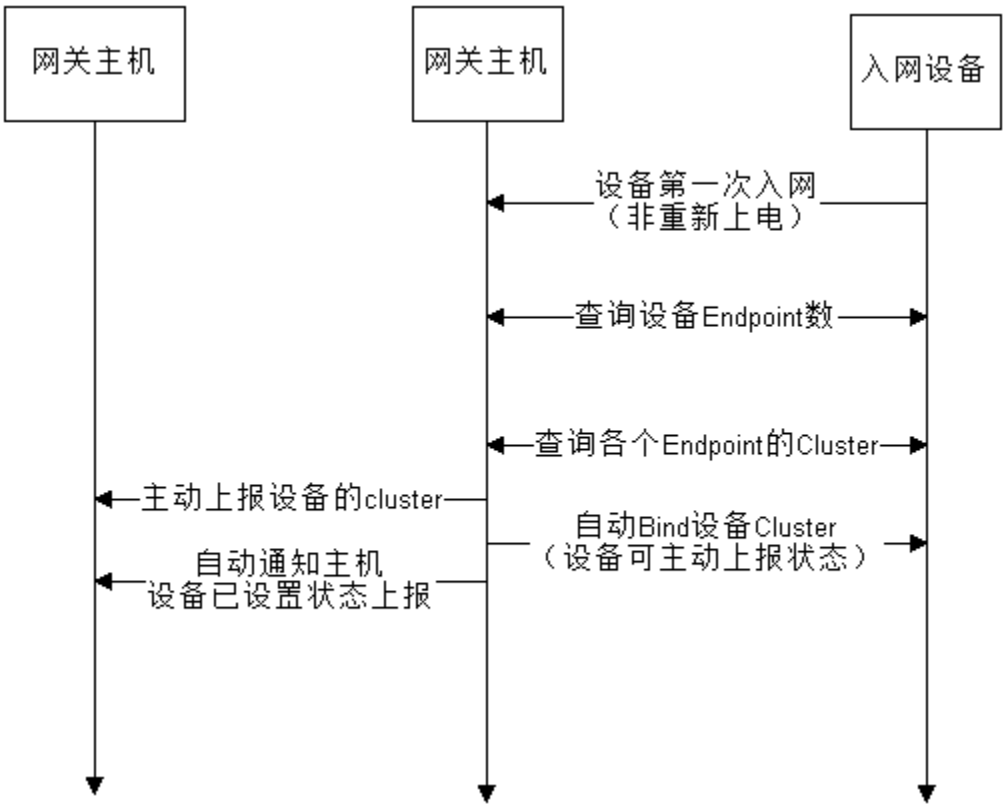
1.2.1. 优化的串口命令流程



流程框图说明：

- 整体架构基于速度阶梯思想，即上位机速度 > 协调器 CPU 速度 > UART 通信速度 > zigbee 传输速度。
- 上位机每次发送 UART 命令给协调器，都需要等待 Send Ack，以确保协调器处于正常工作状态（主要避免上位机为连接 zigbee 协调器）。
- 上位机在向多个 zigbee 目标设备发送消息时，可以根据每个设备返回的发送确认来确保目标设备正常工作。在等待某个设备返回发送确认的间隙可以向其它设备发送消息。该机制用于目标设备数量多，且有部分设备关机，关机设备不拖慢整体控制。
- 协调器收到任何消息都会发给上位机，在不确定消息是否有效（需应用层决定）的情况下协调器会把所有消息都发给上位机，包括在等待“发送确认”的窗口时间内。
- 由于 UART 采用全双工工作机制，因此图中标记的任何延时时间，在其期间内都是有可能收到其它 UART 命令的。
- 该机制适用于 ZDP 命令和 ZCL 命令。且针对于 ZDP 命令必须有 Response 的特性，增加了 Response 自动超时命令机制，适用于瞬间对多个设备同时发起 ZDP 请求（如超过 10 个以上设备同时上电瞬间）。

1.2.2. 自动化的设备入网识别



- 协调器底层区分设备是第一次入网还是重新上电，有效杜绝“僵尸设备”对系统的影响。
- 对于 zigbee 设备入网时需要查询 Endpoint 和 Cluster，传统的 zigbee 协调器需要依赖上位机的控制，UART 传输延迟和操作系统延迟（如 Linux 响应延迟）会导致识别延迟，特别是一些低功耗设备会无法识别。而本方案直接在协调器 MCU 中快速查询 Endpoint 和 Cluster，设备识别效率高，延迟低。识别结果全部通知到主机。
- 具备一定的容错机制，包括传输丢包和重传机制。在若干设备同时入网的应用场景下设备识别率高，效果优于 X 鸦和 X 舟的网关方案。
- 自动设备 Binding Cluster，对于有状态上报功能的设备自动打开状态上报的功能（上报周期为设备出厂默认周期），主机在不需要变更设备状态上报周期时可以不用做任何设置就能获取设备状态上报信息。
- 建议在上位机软件设计中，设备中一个 Endpoint 对应一个虚拟设备，一个 cluster 对应一个功能模型（物模型）。
- 该功能详见 2.2.6 章节“设备功能消息”。

1.3. 串口命令格式

1.3.1. 命令帧格式以及解析

名称	同步头	长度	ACK 标志/帧序号	命令大类	Payload	CRC
字节数	2	1	1(bit7 / bit[0:6])	1	变长	2

✧ 同步头：2 字节，连续的 0xA55A 两个字节表示有效的串口命令（0x5A 前面出现多个 0xA5 也不会出错）

- 长度：1 字节，包含该字节后面所有到 CRC，不包含自己。
- ACK 标志/帧序号：该字节最高位为 1 表示是 ACK，其余 7 位为命令序号，从 0~127 有效。ACK 帧的序号与发给协调器的相等。
- 命令大类：见《命令大类列表》
- Payload：命令的内容，变长。
- CRC：从 ACK 标志/帧序号开始到 Payload 最后一个字节，使用 CRC16 校验，CRC 初始值 0xFFFF，该值为小端模式。

命令大类列表		
值	描述	方向
0x00	协调器配置命令	下行
0x80	协调器异步状态命令	上行
0x01	下发 ZDO 命令	下行
0x01	下发 ZDO 命令反馈	上行
0x81	收到 ZDO 命令	上行
0x02	下发 ZCL 通用命令	下行
0x02	下发 ZCL 通用命令反馈	上行
0x82	收到 ZCL 通用命令	上行
0x03	下发 ZCL 专有命令	下行
0x03	下发 ZCL 专有命令反馈	上行
0x83	收到 ZCL 专有命令	上行
0x04	下发 OTA 命令	下行
0x84	收到 OTA 命令	上行
0xFF	协调器错误状态	上行

- 上位机发给协调器的是下行命令，协调器发给上位机的是上行命令。
- 所有下行命令都会产生 ACK。
- ZDP 命令和 ZCL 命令因为需要无线发送，都会产生命令反馈。命令反馈主要用于检测无线有没有发出去，什么时候发出去，对方有没有收到。合理利用命令反馈可以检测网络良好性。
- 下行命令的帧序号由上位机产生，从 0~127 循环累加，虽然重复不会导致协调器出错，但是上位机需要记住自己产生的帧序号。
- 上行命令的帧序号由协调器产生，从 0~127 循环累加。

1.3.2. ACK 命令帧（ACK 状态表）

名称	同步头	长度	ACK 标志/帧序号		命令大类	Payload		CRC
			ACK=1	帧序号		状态	扩展参数	
字节数	2	1	1/8	7/8	1	1	变长	2

- 任何输入有效的下行命令都会产生 ACK 帧。
- ACK 帧的标志位置 1，帧序号与下行帧保持一致。
- ACK 帧返回状态见《ACK 状态表》
- 部分 ACK 帧会携带扩展参数。

ACK 状态表	
状态值	状态描述

0x00	操作成功
0x01	操作无效
0x02	参数错误
0x10	内存错误
0x11	内存满
0x12	模式不支持
0x13~0xFF	其它未知错误

例：输入协调器软启动命令 A5 5A 05 0C 00 0F F1 C7

ACK 返回 A5 5A 05 8C 00 00 B0 2B

2. 协调器配置命令与协调器异步状态命令

✧

2.1. 协调器配置命令，命令大类 0x00，下行

协调器配置命令统一格式

名称	串口帧 Payload	
	命令 ID	扩展参数
字节数	1	变长

✧ 命令 ID：见《协调器配置命令表》

✧ 扩展参数：各条命令对应的配置参数，最小 0 字节。

协调器配置命令	
命令 ID	命令名称
0x00	查询协调器状态
0x01	优选信道
0x02	启动网络/允许入网
0x05	协调器恢复出厂
0x06	关闭允许入网
0x07	设置协调器 UTC 时间
0x08	查询协调器 UTC 时间
0x0B	协调器加组
0x0C	协调器删组
0x0D	协调器查组
0x0F	协调器软启动

2.1.1. 查询协调器状态

功能描述：查询协调器是否启动网络，以及协调器的 IEEE 地址。

下行：

名称	命令 ID	扩展参数
	0x00	（无）
字节数	1	0

AKC：

名称	状态	扩展参数
		IEEE 地址
字节数	1	8

- ✧ 状态：操作成功 – 协调器有网络，操作无效-协调器无网络。
- ✧ IEEE 地址：协调器本身的 IEEE 地址。

例

发送：A5 5A 05 0C 00 00 B1 C3

返回：A5 5A 0D 8C 00 00 A4 A2 B4 21 00 4B 12 00 C3 32

A4 A2 B4 21 00 4B 12 00 是协调器的 IEEE 地址（小端模式，保存数据库需要颠倒过来）

2.1.2. 优选信道

功能描述：设置协调器的优选信道，只在启动网络前设置才有效。协调器恢复出厂也可以设置。

下行：

名称	命令 ID	扩展参数	
	0x01	操作类型	信道号
字节数	1	1	1

- ✧ 操作类型：0 – 禁用信道，1 – 使能信道
- ✧ 信道值：需要禁用或使能的信道值，如果填 0 则返回当前使能信道。（协调器默认使能 11， 14， 15， 19， 20， 24， 25 共 7 个信道）

AKC：

名称	状态	扩展参数
		信道 Mask 表
字节数	1	4

- ✧ 状态：见《ACK 状态表》
- ✧ 信道 Mask 表：uint32 的信道值，第 11 位到第 26 位表示信道的使能状态，默认值 0x0318c800。

例

发送：A5 5A 07 0A 00 01 00 0B AC 06

返回：A5 5A 09 8A 00 00 00 C0 18 03 7A F5

设置屏蔽掉 0x0B（11）信道，返回 0x0318C000，11 信道被屏蔽掉。

2.1.3. 启动网络/允许入网

功能描述：出厂模式的协调器使用该命令创建新网络，已创建网络的协调器打开允许入网 180 秒，该命令只有在软启动后生效。

下行：

名称	命令 ID	扩展参数
	0x02	白名单
字节数	1	1

✧ 白名单：1 - 开启白名单，0 - 关闭白名单。普通入网该位填 0

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：见《ACK 状态表》

例

发送：A5 5A 06 0B 00 02 00 03 60

返回：A5 5A 05 8B 00 00 01 EA

打开允许入网，不开白名单，返回表示允许入网打开成功

2.1.4. 扫描信标信道

功能描述：网关主机可控制协调器扫描信道，在软启动前或后均有效。有两种扫描模式，分别是扫描空间内其它 zigbee 网络（含网关和 zigbee 路由设备），以及扫描信道信号强度（含 2.4G 频段的 WIFI 和蓝牙信号）

下行：

名称	命令 ID	扩展参数		
	0x03	信道位使能	单信道时间	扫描模式
字节数	1	4	1	1

✧ 信道位使能：32bit 的位使能，第 n 位设置为 1 表示扫描该信道，有效范围 11 位~26 位。建议设置值为 0x07FFF800（需小端模式填入该项目）以便扫描从 11~26 信道

✧ 单信道时间：每个信道的扫描时间，1~5 有效。1 = 30ms，2=60ms，3=120ms，4=240ms，5=480ms。时间越长扫描越慢，但是准确度越高。

✧ 扫描模式：0-扫描信标，返回空间内其它 zigbee 设备；1-扫描信道，返回各个信道的信号强度，包括 WIFI 和蓝牙的干扰。

扫描结果通过异步状态命令“扫描结果”返回

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：见《ACK 状态表》

注意事项：如果一次扫描还未结束就发起下次扫描，会返回错误结果。扫描结束可以在异步状态命令“扫描结果”中获取。

2.1.5. 协调器导入白名单

功能描述：向协调器导入待入网设备的白名单数据（IEEE 地址）或者 Install code。在允许入网时开启白名

单模式时任何设备入网都必须通过 IEEE 地址白名单验证或 Install code 验证才能入网。不开启白名单模式时仅过滤需要导入 Install code 的设备。

下行：

名称	命令 ID	扩展参数	
	0x04	设备 IEEE 地址	Install Code
字节数	1	8	18\0

✧ 设备 IEEE 地址：待入网设备的 IEEE 地址

✧ Install Code：待入网设备的 Install Code

注意事项：允许入网时开启白名单模式，任何设备入网都必须输入 IEEE 地址，有 Install Code 的设备需要输入 Install Code。允许入网是关闭白名单模式，没有 Install Code 的设备不需要输入 IEEE 地址，但是有 Install Code 的设备必须输入 IEEE 地址和 Install Code。

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：见《ACK 状态表》

2.1.6. 协调器重启/恢复出厂

功能描述：已经创建网络的协调器恢复出厂，该命令需要在协调器已经创建网络后才能生效，参数 PANID 和信道值需要填写正确。

下行：

名称	命令 ID	扩展参数	
	0x05	PAN ID	信道
字节数	1	2	1

✧ PAN ID：协调器创建网络的 PANID，可以在软启动协调器时获取，重启填入 0xFFFF。

✧ 信道：协调器创建网络的信道，可以在软启动协调器时获取，重启填入 0x00。

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：见《ACK 状态表》

例

发送：A5 5A 08 0B 00 05 D9 65 13 BA CA

返回：A5 5A 05 8B 00 01 C0 2A

输入协调器 PANID= 0x65D9，信道 0x13=19，实际协调器 PANID = 0x654C，实际信道 0x0E=14，因此返回“恢复出厂失败”。

2.1.7. 关闭允许入网

功能描述：关闭协调器的入网允许，和启动网络/允许入网配合使用。

下行：

名称	命令 ID	扩展参数
	0x06	无
字节数	1	0

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：见《ACK 状态表》

例

发送：A5 5A 05 0D 00 06 60 01

返回：A5 5A 05 8D 00 00 E1 EB

2.1.8. 设置协调器 UTC 时间

功能描述：设置协调器的 UTC 时间，部分入网设备（如电表）会在协调器上读取当前时间，协调器起到 UTC 时间服务的功能。协调器本身器件计时精度不高，建议 1 分钟与上位机同步一次。

下行：

名称	命令 ID	扩展参数
	0x07	UTC32 时间
字节数	1	4

✧ UTC32 时间：UTC 格式的 uint32 时间值，单位为秒，小端模式，当值=0x00000000 时表示时间 2000-01-01 00:00:00。

AKC：

名称	状态	扩展参数
字节数	1	

✧ 状态：任何时候都返回“操作成功”

例：暂无

2.1.9. 查询协调器 UTC 时间

功能描述：查询协调器当前 UTC 时间

下行：

名称	命令 ID	扩展参数
	0x08	空

字节数	1	0

AKC：

名称	状态	扩展参数
		UTC32 时间
字节数	1	4

✧ 状态：任何时候都返回“操作成功”

✧ UTC32 时间：协调器当前的 UTC32 时间，这个时间一直在跑。

例：暂无

2.1.10. 查询协调器缓存地址表

功能描述：设备入网后，协调器中会缓存入网设备的 IEEE 地址和短地址，如果网关中的数据库丢失，且在使用“查询短地址（3.2.1）”命令查询无返回时，可以通过该表查询 IEEE 地址和短地址。

下行：

名称	命令 ID	扩展参数
	0x09	表项 index
字节数	1	2

✧ 表项 index：查询地址表的 index 序号，从 0~255 有效，小端模式

AKC：

名称	状态	扩展参数		
		表项 index	短地址	IEEE 地址
字节数	1	2	2	8

✧ 状态：任何时候都返回“操作成功”

✧ 表项 index：查询的地址表 index 序号。

✧ 短地址：设备的短地址

✧ IEEE 地址：设备的 IEEE 地址

2.1.11. 协调器加组

功能描述：协调器加入一个组

下行：

名称	命令 ID	扩展参数
	0x0B	组 ID
字节数	1	2

✧ 组 ID：协调器加入的组 ID，从 0x0001 开始到 0xFFF0 有效。

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：0x00 - 加组成功，0xB8 - 已经加入过该组，其它-加组失败

例

发送：A5 5A 07 1B 00 0B 01 00 30 50

返回：A5 5A 05 9B 00 B8 00 5D

协调器加入 0x0001 组，返回 0xB8 表示协调器已经存在该组中。

2.1.12. 协调器删组

功能描述：协调器删除一个组

下行：

名称	命令 ID	扩展参数
	0x0C	组 ID
字节数	1	2

✧ 组 ID：协调器需要删除的组 ID

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：见《ACK 状态表》

例

发送：A5 5A 07 1D 00 0C 01 00 09 91

返回：A5 5A 05 9D 00 00 E0 2E

01 00 表示删除协调器的 0x0001 组

返回的 00 表示删组成功

2.1.13. 协调器查组

功能描述：查询协调器加入了哪些分组

下行：

名称	命令 ID	扩展参数
	0x0D	无
字节数	1	0

AKC：

名称	状态	扩展参数	
		组总数	组 ID 表
字节数	1	1	2*变长

✧ 状态：见《ACK 状态表》

✧ 组总数：加入的分组总数

例

发送：A5 5A 05 1C 00 0D 71 C3

返回：A5 5A 12 9C 00 00 06 01 00 02 00 03 00 04 00 05 00 06 00 C0 AB

06 表示协调器加入 6 个组

01 00 02 00 03 00 04 00 05 00 06 00 按照小端模式，分别代表 0x0001，0x0002，0x0003，0x0004，0x0005，0x0006 共 6 个大组。

2.1.14. 增加动态 Profile 和 Endpoint

功能描述：协调器默认 Endpoint 1 支持 Profile 0x0104，如果有第三方设备需要协调器其它端口作为上传目标，或者需要支持其它 Profile，可通过主机创建新的 Profile 和 Endpoint

下行：

名称	命令 ID	扩展参数	
	0x0E	Endpoint	Profile
字节数	1	1	2

- Endpoint：新建的 Endpoint
- Profile：新建的 Profile

AKC：

名称	状态	扩展参数
		Profile 编号
字节数	1	1

- 状态：见《ACK 状态表》
- 当状态为“成功”新建 Profile 分配到的编号

2.1.15. 协调器软启动

功能描述：任何时候协调器上电后需要软启动，防止协调器在上电后比上位机先启动，在上位机启动过程中串口向上位机输出信息，导致上位机出错。协调器软启动可以反复进行，不会影响协调器运行。

下行：

名称	命令 ID	扩展参数
	0x0F	无
字节数	1	0

AKC：

名称	状态	扩展参数
		无
字节数	1	0

- 状态：见《ACK 状态表》

例

发送：A5 5A 05 0C 00 0F F1 C7

返回：A5 5A 05 8C 00 00 B0 2B

2.1.16. 查询直连协调器的 End Device 的短地址

功能描述：在使用“查询短地址”（3.2.1）时无果，再使用该指令作为补充。两条指令都查不到时可视作设备离线。

下行：

名称	命令 ID	扩展参数
	0x13	IEEE 地址
字节数	1	8

✧ IEEE 地址：要查询设备的 IEEE 地址

ACK：

名称	状态	扩展参数		
		IEEE 地址	短地址	设备关系
字节数	1	8	2	1

✧ 状态：0-读到有效设备，1-读到无效设备
✧ IEEE 地址：被查询设备的 IEEE 地址
✧ 短地址：被查询设备的短地址
✧ 设备关系：1 - 休眠终端节点， 2 - 不可休眠终端节点，其它，路由设备。

2.1.17. 激活协调器

功能描述：未激活的协调器有以下限制

✧ 365 天使用期限，到期后清除组网。
✧ 不能使用 OTA 指令对第三方设备升级。
✧ 不能使用白名单功能

下行：

名称	命令 ID	扩展参数
	0x24	激活码
字节数	1	16

✧ 激活码：由协调器的 IEEE 地址通过 AES 加密得到 16 字节激活码

AKC：

名称	状态	扩展参数
		无
字节数	1	0

状态：0-激活成功，1-激活失败，2-重复激活（设备已激活过）。

注意：输入一串不正确的激活码（如全 0x00 或全 0xFF），通过返回“重复激活”状态，可以判断协调器是

否激活过。

例：

发送：A5 5A 15 0F 00 24 57 D0 99 D2 43 9B E3 5A C8 0F E7 25 76 E6 AF 07

返回：A5 5A 05 8F 00 01 81 EB，激活失败

2.1.18. 读取 NV 数据

功能描述：读取协调器中的 NV 数据

下行：

名称	命令 ID	扩展参数		
	0x20	层级	表项	条目
字节数	1	1	1	2

✧ 层级：0 - 协议层数据，1 - 应用层数据

✧ 表项：见表

层级	表项	名称	最大条目数
协议层	0	协议栈参数	0 ~ 0x1FF
	1	地址表	240 条
	3	子节点表	32 条
	4	设备密钥表	200 条
	5	Install Code 表	40 条
	6	全局密钥表	3 条
	7	NWK 密钥种子表	5 条
	8	分组表	32 条
应用层	0	应用数据	3
	2	Green Power 表	5

✧ 条目：数据条目，从 0 ~ 最大条目数有效

ACK：

名称	状态	扩展参数
		读到的数据
字节数	1	变长

返回帧包含读到的数据

2.1.19. 强行写 NV

功能描述：强行写入 NV 数据，但是需要对 NV 初始化

下行：

名称	命令 ID	扩展参数			
	0x29	层级	表项	条目	NV 数据
字节数	1	1	1	2	变长

✧ 层级：0 - 协议层数据，1 - 应用层数据

✧ 表项：见“读取 NV”的表
✧ 条目：需要写入的数据条目
✧ NV 数据：需要写入的数据，长度应与读到的长度一致

ACK：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：返回状态表示写入结果

2.1.20. 初始化 NV

功能描述：初始化 NV ROM，为 NV 数据提前分配空间，协调器不用建立网络就有 ROM 空间。该命令仅在网络未建立，且协调器未软启动时才有效。

下行：

名称	命令 ID	扩展参数
	0x2A	无
字节数	1	0

ACK：

名称	状态	扩展参数
		无
字节数	1	0

✧ 返回状态不为 Success，表示初始化不成功

2.2. 协调器异步状态命令，命令大类 0x80，上行

协调器异步状态命令统一格式

名称	串口帧 Payload	
	命令 ID	扩展参数
字节数	1	变长

✧ 命令 ID：见《协调器异步状态命令表》
✧ 扩展参数：各条命令对应的配置参数，最小 0 字节。

协调器异步状态命令	
命令 ID	命令名称
0x80	协调器启动
0x81	允许入网时间
0x82	网络启动状态
0x84	设备入网消息
0x85	设备功能消息

0x86	设备注销消息
0x87	设备离网消息
0x88	设备入网过程
0x89	设备自动绑定
0x8F	设备功能获取失败

2.2.1. 协调器启动

功能描述：协调器上电或复位，都会发出该命令。上位机收到该命令后需要对协调器进行软启动。

上行：

名称	命令 ID	扩展参数	
	0x80	复位模式	软件版本
字节数	1	1	1

- ✧ 复位模式：0 - 上电复位，1 - 引脚复位，6 - 软复位
- ✧ 软件版本：协调器的应用程序版本，1 字节，范围 1~255。该版本和 OTA 无关，仅作为协调器运行程序的记号作用。

例

收到：A5 5A 07 46 80 80 06 07 C7 85

表示协调器是软复位，软件版本为 7。

2.2.2. 允许入网时间

功能描述：打开允许入网和关闭允许入网，或者有新设备加入网络都会产生该消息。在新设备入网过程中，会加长设备允许入网的时间。

上行：

名称	命令 ID	扩展参数	
	0x81	允许入网时间	入网锁
字节数	1	1	1

- ✧ 允许入网时间：最新的允许入网时间，大于 0 表示打开允许入网，等于 0 表示关闭允许入网。
- ✧ 入网锁：0 - 解除锁定，1 - 锁定。协调器在锁定状态下，入网允许不能被其它设备打开。

例

收到：A5 5A 07 79 80 81 B4 00 B7 22

表示允许入网打开，时间 0xB4=180 秒，入网锁解锁，其它设备可以续长允许入网时间。

2.2.3. 网络启动状态

功能描述：协调器网络启动状态，在协调器软启动成功，网络启动成功开启允许入网时，都会产生该消息。

上行：

名称	命令 ID	扩展参数						
	0x82	网络状态	信道	PAN ID	Ext PANID	NWK 密钥	密钥号	软件版本

字节数	1	1	1	2	8	16	1	1

- ✧ 网络状态：0 - 未建立网络，1 - 已建立网络， 2 - 建立网络且允许入网打开
- ✧ 信道：协调器建立网络的信道，11~26 是有效值。
- ✧ PANID：协调器的 PANID，任何非 0xFFFF 都是有效值。
- ✧ Ext PANID：允许入网状态下为协调器随机产生的 Ext PANID，其它状态为协调器 IEEE 地址。
- ✧ NWK 密钥：协调器随机产生的 NWK 密钥，配合抓包工具使用。
- ✧ 密钥号：用于查看协调器 NWK 密钥的更新次数，目前协调器无密钥更新机制，该值为 0。
- ✧ 软件版本：和“协调器启动”中的软件版本一致。

例

收到：A5 5A 23 66 80 82 02 0E 4C 65 34 1A 0D 86 C3 61 B0 58 8E B8 50 38 87 01 57 D5 24 EE 24 32 8C B3 C4 08 00 07 D9 6F

02 表示打开允许入网

0E 表示当前信道是 14 信道

4C 65 表示当前 PANID 是 0x654C

34 1A 0D 86 C3 61 B0 58 是协调器的 Ext PANID（小端模式），每次协调器恢复出厂，Ext PANID 随机变化

8E B8 50 38 87 01 57 D5 24 EE 24 32 8C B3 C4 08 是协调器的 NWK 密钥，可以复制粘贴到抓包工具

00 07 不用管

2.2.4. 扫描结果

功能描述：返回“扫描信标信道”的扫描结果，有 3 中类型：扫描到单个信标，信标扫描结束，信道信号强度扫描。

名称	命令 ID	扩展参数
	0x83	扫描结果
字节数	1	14/2/变长

扫描结果：一共有 3 种数据格式

扫描到单个信标,固定 14 字节。

信道	短地址	PANID	Ext PANID	LQI 强度
1	2	2	8	1

- ✧ 信道：发出信标的设备所在信道
- ✧ 短地址：发出信标的设备的短地址
- ✧ PANID：发出信标的设备的 PANID
- ✧ Ext PANID：发出信标的设备的 Ext PANID
- ✧ LQI 强度：信标的信号强度，最大 255.

扫描信标结束，固定 2 字节

标识符	结果状态
1	1

- ✧ 标识符：该标识符为 0
- ✧ 结果状态：0x00 - 扫描结束，扫到了信标；0xEA - 扫描结束了一个信标都没扫到。

扫描信道，变长。

标识符	信道位使能	扫描信道个数	各个信道信号强度
1	4	1	变长（1~16）

- 标识符：该值为 0xFF 表示信道扫描结束
- 信道位使能：扫描过的信道的位使能
- 扫描信道个数：扫描到的信道的格式
- 各个信道信号强度：各个信道的 LQI 信号强度，最大 255，该值越大干扰越大。

2.2.5. 设备入网消息

功能描述：设备入网时上报 IEEE 地址和短地址，同时附带入网时的父节点短地址，是否第一次入网，入网设备是否支持路由功能和低功耗功能。

上行：

名称	命令 ID	扩展参数				
	0x84	IEEE 地址	短地址	父节点	入网模式	设备类型
字节数	1	8	2	2	1	1

- IEEE 地址：入网设备的 IEEE 地址
- 短地址：入网设备的短地址
- 父节点：入网设备的父节点短地址（删除低功耗设备需要这个）
- 入网模式：0x00 - 入网失败，0x01 - 第一次入网，0x02 - 设备重启或切换父节点，0x03 - 设备重新认证
- 设备类型：0x01 - 设备支持路由转发，0x02 - 设备支持低功耗

例

收到：A5 5A 13 6B 80 84 70 0F 00 00 AD 38 C1 A4 31 17 00 00 01 01 93 BD

70 0F 00 00 AD 38 C1 A4 是入网设备的 IEEE 地址（小端模式）

31 17 是入网设备的短地址，等于 0x1731

00 00 表示设备入网的父节点是协调器，没有经过路由设备入网

01 表示设备是第一次入网

01 表示设备是支持路由功能

重启设备后收到

收到：A5 5A 13 75 80 84 70 0F 00 00 AD 38 C1 A4 31 17 FE FF 02 01 FB 33

IEEE 地址，短地址雷同

FE FF 表示父节点点变成了 0xFFFE 是一个无效地址，因为路由设备重启后就不需要父节点了。

02 表示设备重启

01 表示设备是支持路由功能，路由功能和低功耗功能不受重启影响

2.2.6. 设备功能消息（Device ID 表和 Cluster ID 表）

功能描述：设备入网后，每个端口都上传一次该消息。该消息关联了设备的 IEEE 地址和短地址，包含了每个 Endpoint 值，以及每个 Endpoint 的 Device ID，每个 Endpoint 下的所有 cluster ID，可以完整的区分不同的设备，彻底区分窗帘电机和红外传感器。

上行：

名称	命令 ID	扩展参数							
	0x85	结束	长地址	短地址	endpoint	Profile ID	Device ID	In cluster 列表	Out cluster 列表
字节数	1	1	9	2	1	2	2	变长	变长

- ✧ 结束：一个设备入网后会上报多条该命令，最后一条的结束标志是“0x01”
- ✧ 长地址：此处引入长地址的概念，由 IEEE 地址+设备端口组成（小端模式）
- ✧ 短地址：设备的短地址
- ✧ endpoint：设备的 endpoint 号
- ✧ profile ID：设备当前 endpoint 中的 profile ID，等于 0x0104 的 endpoint 有实际的设备意义，比如控灯，传感器等实际意义。
- ✧ device ID：设备的 Device ID，见 Device ID 表。
- ✧ In cluster 列表：In cluster 表中的 cluster ID，表示该设备支持哪些功能（受别人控制的功能，zigbee 里面叫 Sever）。
- ✧ Out cluster 列表：Out cluster 表中的 cluster ID，表示该设备可以控制哪些功能（控制别人，zigbee 里面叫 client）。

Cluster 列表		
名称	cluster ID 数量	cluster ID 数组
字节数	1	2 * N

- ✧ cluster ID 数量：表示该 cluster 表的大小
- ✧ cluster ID 数组：所有的 cluster 构成的数组，可以得到该设备的全部 cluster ID，分析出该设备的应用功能，cluster ID 对应功能见《cluster ID 表》

Device ID 表			
分类	Device	设备名	Device ID
Generic	On/Off Switch	通断开关	0x0000
	Level Control Switch	级数控制器（旋钮）	0x0001
	On/Off Output	开关输出端	0x0002
	Level Controllable Output	旋钮输出端	0x0003
	Scene Selector	场景控制器	0x0004
	Configuration Tool		0x0005
	Remote Control	遥控器	0x0006
	Combined Interface		0x0007
	Range Extender		0x0008
	Mains Power Outlet		0x0009
	Door Lock	门锁	0x000A
	Door Lock Control	门锁控制器	0x000B
	Simple Sensor	简单传感器	0x000C
	Consumption Awareness Device		0x000D
Light 照明类	On/Off Light	开关灯	0x0100
	Dimmable Light	调光灯	0x0101

	Color Dimmable Light	彩色灯	0x0102
	On/Off Light Switch	开关灯控制器	0x0103
	Dimmer Switch	调光灯控制器	0x0104
	Color Dimmer	彩色控制器	0x0105
	Light Sensor	光传感器	0x0106
	Occupancy Sensor		0x0107
Closures 门窗类	Shade		0x0200
	Shade Controller		0x0201
	Window Cover	窗帘	0x0202
	Window Cover control	窗帘控制器	0x0203
HVAC 暖通类	Heating/Cooling Unit	冷暖控制器	0x0300
	Thermostat	温控器	0x0301
	Temperature Sensor	温控传感器	0x0302
	Pump		0x0303
	Pump Controller		0x0304
	Pressure Sensor		0x0305
	Flow Sensor		0x0306
IAS 安防类	IAS Control and Indicating Equipment		0x0400
	IAS Ancillary Control Equipment		0x0401
	IAS Zone		0x0402
	IAS Warning Device		0x0403

cluster ID 表		
cluster ID	描述	功能
0x0000	ZCL_CLUSTER_ID_GEN_BASIC	设备基础信息
0x0003	ZCL_CLUSTER_ID_GEN_IDENTIFY	设备标记（ident）
0x0004	ZCL_CLUSTER_ID_GEN_GROUPS	组功能协议
0x0005	ZCL_CLUSTER_ID_GEN_SCENES	场景功能协议
0x0006	ZCL_CLUSTER_ID_GEN_ON_OFF	开关灯协议
0x0008	ZCL_CLUSTER_ID_GEN_LEVEL_CONTROL	调亮度协议
0x0019	ZCL_CLUSTER_ID_OTA	OTA 升级
0x0400	ZCL_CLUSTER_ID_MS_ILLUMINANCE_MEASUREMENT	光感协议
0x0500	ZCL_CLUSTER_ID_SS_IAS_ZONE	安防类协议
0x1000	ZCL_CLUSTER_ID_TOUCHLINK	Touch Link 功能

例

收到 3 条命令，对应设备 3 个端点

第一条：A5 5A 28 6E 80 85 00 01 70 0F 00 00 AD 38 C1 A4 31 17 01 04 01 01 01 07 00 00 03 00 04 00 05 00 06 00

08 00 00 10 01 19 00 E2 B6。

长地址 01 70 0F 00 00 AD 38 C1 A4。

短地址 0x1731。

端口 0x01。

Profile=0x0104，我们现在的应用只处理 Profile=0x0104 的端口。

Device ID=0x0101，查表为调光灯。

In Cluster 一共 7 条，分别为：

0x0000, 0x0003, 0x0004, 0x0005, 0x0006, 0x0008, 0x1000。

支持设备基础信息，设备标记，组功能，场景功能，开关灯，调亮度，Touch Link。

Out Cluster 只有 1 条 0x0019，只有 OTA 升级功能。

第二条：A5 5A 20 70 80 85 00 02 70 0F 00 00 AD 38 C1 A4 31 17 02 04 01 06 01 03 00 00 03 00 06 FC 01 19 00 F7 91

长地址 02 70 0F 00 00 AD 38 C1 A4。

短地址 0x1731。

端口 0x02。

Profile=0x0104。

Device ID=0x0106，查表为光传感器。

In Cluster 一共 3 条，分别为：

0x0000, 0x0003, 0xFC06

支持设备基础信息，设备标记和光感，不支持场景和分组，也不能调光。

Out Cluster 只有 1 条 0x0019，只有 OTA 升级功能。

第三条：A5 5A 1A 71 80 85 01 F2 70 0F 00 00 AD 38 C1 A4 31 17 F2 E0 A1 61 00 00 01 21 00 EC CF

该命令为最后一个端口的信息

长地址 F2 70 0F 00 00 AD 38 C1 A4

端口 0xF2

Profile ID=0xA1E0，Device ID=0x0061

该端口是用于 ZigBee Green Power 协议的，Device=0x0061 是 Green Power 桥接器，由于我们的应用中暂时没有 Green Power 设备，所以不需要对接。

2.2.7. 设备注销消息

功能描述：设备离网时，如果入网时通过了父节点桥接，协调器会收到父节点发来的注销设备消息。

上行：

名称	命令 ID	扩展参数		
	0x86	短地址	IEEE 地址	父节点
字节数	1	2	8	2

- ✧ 短地址：退网设备的短地址
- ✧ IEEE 地址：退网设备的 IEEE 地址
- ✧ 父节点：退网设备的父节点

例，设备清网时，协调器收到

A5 5A 11 79 80 86 31 17 70 0F 00 00 AD 38 C1 A4 00 00 FA 51

设备短地址 0x1731

IEEE 地址 0xA4C138AD00000F70

父节点短地址 0x0000，是协调器

注意：如果一个设备有路由功能，同时通过另一个设备桥接进来，这个设备清网时先前桥接的路由设备已经关机或清网，该消息会收不到。

2.2.8. 设备离网消息

功能描述：设备离网时，发给协调器的离网指令。

上行：

名称	命令 ID	扩展参数				
	0x87	短地址	IEEE 地址	离网模式	删子节点	切换父节点
字节数	1	2	8	1	1	1

- ✧ 短地址：退网设备的短地址
- ✧ IEEE 地址：退网设备的 IEEE 地址
- ✧ 离网模式：0 – IEEE 地址和短地址对应的设备本身离网，1 – 设备自己不清网而是踢掉桥接的子设备，一般情况下都是 0。
- ✧ 删子节点：0 – 设备仅自己清网，1 – 设备清网时捎带了桥接的全部子设备。一般情况是 0。
- ✧ 切换父节点：0 – 设备真正清网了，1 – 设备没有真正清网，只是切换了父节点。

A5 5A 12 78 80 87 31 17 70 0F 00 00 AD 38 C1 A4 00 00 00 10 12

设备短地址 0x1731

IEEE 地址 0xA4C138AD00000F70

后面 3 个标志位全是 0x00，表示这个设备是真正的清网了，而且只有它一个清网。

注意：如果清网的设备距离协调器太远，或者多个设备一起清网，协调器可能会收不到这个消息。因此该消息要与“设备注销消息”一起使用判断设备离开。

2.2.9. 设备入网过程

功能描述：该命令用于诊断设备入网过程，可以区分入网设备是否 ZigBee 3.0 模式。

上行

名称	命令 ID	扩展参数			
	0x88	交互状态	IEEE 地址	短地址	密钥信息
字节数	1	1	8	2	18

- ✧ 交互状态：0 – 准备交互密钥。1 – 密钥交互成功。 2 – 密钥交互失败。
- ✧ IEEE 地址：设备的 IEEE 地址。
- ✧ 短地址：设备短地址。
- ✧ 密钥信息：2 字节密钥存储 index，16 字节 APS Link Key。

备注：任何设备入网，都有“准备交互密钥”的动作，但是最终只有符合 ZigBee 3.0 的设备才会有“密钥交互成功”，非 ZigBee 3.0 的设备会“密钥交互失败”。极端情况下多个 ZigBee 3.0 的设备同时入网也会出现

“密钥交互失败”，设备会自动清网然后重新再自动配网。

2.2.10. 设备自动绑定通知

功能描述：设备第一次入网后，自动设置绑定，并通知网关。

上行：

名称	命令 ID	扩展参数		
	0x89	长地址	绑定 cluster	无线状态
字节数	1	9	2	1

- ✧ 长地址：由设备端口+IEEE 地址构成设备长地址，同 2.2.5“设备功能消息”。
- ✧ 绑定 cluster：设备绑定网关的 cluster，为后续自动上报数据做准备。
- ✧ 无线状态：见 3.1.6《无线通信状态表》。

2.2.11. 设备功能获取失败通知

功能描述：出现该情况是在 2.2.5“设备功能消息”时出现设备掉线或者丢包，有可能是设备入网没成功，又自己退了。该消息仅在设备新入网的时候出现。

上行：

名称	命令 ID	扩展参数			
	0x8F	IEEE 地址	短地址	端口	失败原因
字节数	1	8	2	1	1

- ✧ IEEE 地址：获取失败的设备 IEEE 地址。
- ✧ 短地址：获取失败的设备短地址。
- ✧ 端口：获取失败的设备端口。
- ✧ 失败原因：综合《ACK 状态表》和《无线通信状态表》中的错误。

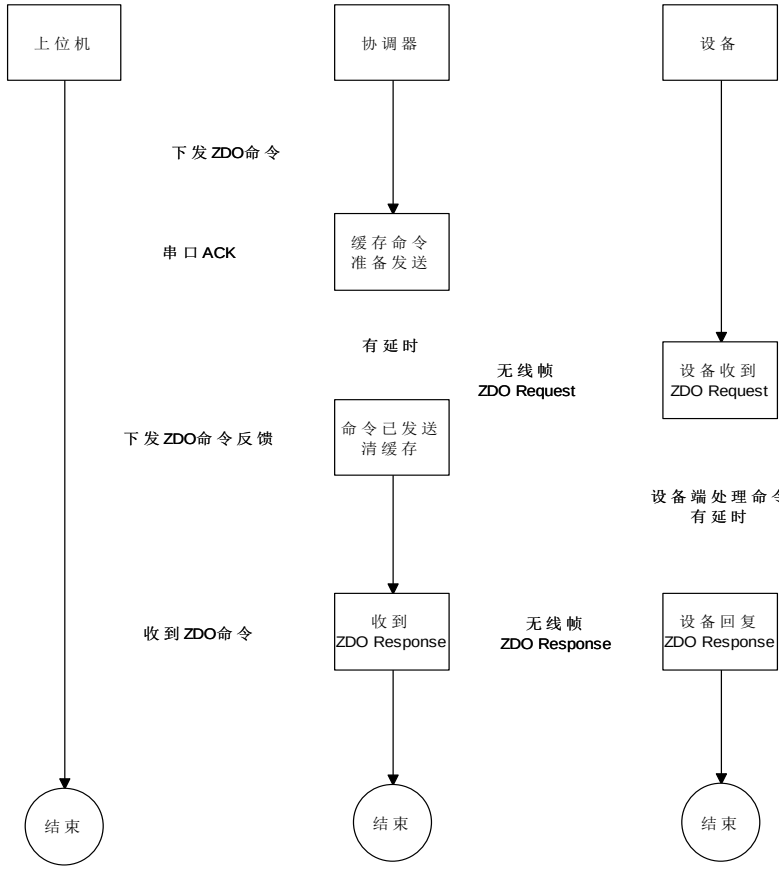
注意：网关收到该消息后，“设备功能消息”有可能会重传成功。如果网关收到“设备功能消息”后可以忽略该消息。

3. ZDO 命令

✧

ZDO 命令是 ZigBee Device Object 的缩写，用于管理所有设备的入网，清网，识别等基础功能操作。无论是什么设备，只要加入了协调器的网络中，就能被协调器通过 ZDO 命令进行管理。

ZDO 命令有下发 ZDO 命令，下发 ZDO 命令反馈和收到 ZDO 命令三种。ZDO 命令属于无线通信命令，因此命中率和成功率受无线通信的物理形式制约。



ZDO 命令通信流程图

3.1. ZDO 命令的帧格式

3.1.1. 下发 ZDO 命令，命令大类 0x01，下行

功能描述：上位机向协调器下发 ZDO 命令，协调器会立即返回串口 ACK（时间约 1 毫秒左右）。

下行：

名称	串口帧 Payload		
	命令 ID	短地址	扩展参数
字节数	2	2	变长

- ✧ 命令 ID：2 字节的命令 ID，见《ZDO 命令表》。
- ✧ 短地址：目标设备短地址
- ✧ 扩展参数：命令对应的参数，最小长度为 0。

ACK：

名称	串口帧 Payload	
	状态	无线序号
字节数	1	1

- ✧ 状态：见《ACK 状态表》，只要返回“操作成功”，必然产生“下发 ZDO 命令反馈”。
- ✧ 无线序号：由协调器产生的 0~255 循环增长的序号，可用来查找对应的“下发 ZDO 命令反馈”和“收到 ZDO 命令”。

3.1.2. 下发 ZDO 命令反馈，命令大类 0x01，上行

功能描述：协调器在把上位机输入的命令发送完毕后，会向上位机发送“下发 ZDO 命令反馈”，该命令中包含了无效发送状态，表示无线通信是否成功。

上行：

名称	串口帧 Payload			
	命令 ID	短地址	无线状态	无线序号
字节数	2	2	1	1

- ✧ 命令 ID：反馈上位机曾经发送的命令 ID。
- ✧ 短地址：反馈上位机曾经发送过的目标设备短地址。
- ✧ 无线状态：见《无线通信状态表》。
- ✧ 无线序号：上位机从“下发 ZDO 命令”的串口 ACK 中，收到的无线序号。

3.1.3. 收到 ZDO 命令，命令大类 0x81，上行

功能描述：协调器收到设备发来的 ZDO 命令，输出给上位机。

上行：

名称	串口帧 Payload				
	命令 ID	短地址	ZDO 状态	无线序号	扩展参数
字节数	2	2	1	1	变长

- ✧ 命令 ID：反馈上位机曾经发送的命令 ID。
- ✧ 短地址：反馈上位机曾经发送过的目标设备短地址。
- ✧ ZDO：设备端回复 ZDO 命令的结果，见《ZDO 状态表》。当该值为 0xFF 时为等待 Response 超时。（在有很多返回时，超时可以当做终结符号使用）
- ✧ 无线序号：上位机从“下发 ZDO 命令”的串口 ACK 中，收到的无线序号。

3.1.4. ZDO 命令表

ZDO 命令表

- ✧ 该表格中仅包含“下发 ZDO 命令”和“收到 ZDO 命令”
- ✧ 在“方向”一栏中，下发 ZDO 命令简称“发”，收到 ZDO 命令简称“收”

命令 ID	方向	描述符	功能解释
0x0000	发	NWK_addr_req	根据 IEEE 地址查短地址
0x8000	收	NWK_addr_rsp	返回查询的短地址
0x0001	发	IEEE_addr_req	根据短地址查 IEEE 地址
0x8001	收	IEEE_addr_rsp	返回查询的 IEEE 地址
0x0004	发	Simple_Desc_req	查询设备端口信息
0x8004	收	Simple_Desc_rsp	返回查询设备端口信息
0x0005	发	Active_EP_req	查询设备端口数量
0x8005	收	Active_EP_rsp	返回查询设备端口数量
0x0021	发	Bind_req	设置设备与设备绑定
0x8021	收	Bind_rsp	返回设备绑定结果
0x0022	发	Unbind_req	解除设备与设备绑定

0x8022	收	Unbind_rsp	返回设备解绑结果
0x0033	发	Mgmt_Bind_req	查询设备绑定关系
0x8033	收	Mgmt_Bind_rsp	返回查询设备绑定结果
0x0034	发	Mgmt_Leave_req	删除设备
0x8034	收	Mgmt_Leave_rsp	返回删除设备结果

3.1.5. ZDO 状态表

ZDO 状态表		
状态 ID	描述符	功能解释
0x00	ZDP_SUCCESS	操作成
0x80	ZDP_INVALID_REQTYPE	无效操作
0x81	ZDP_DEVICE_NOT_FOUND	设备未找到
0x82	ZDP_INVALID_EP	不正确的端点
0x83	ZDP_NOT_ACTIVE	端点不存在
0x84	ZDP_NOT_SUPPORTED	设备不支持该命令
0x85	ZDP_TIMEOUT	设备处理超时
0x86	ZDP_NO_MATCH	设备处理匹配失败
0x88	ZDP_NO_ENTRY	设备不存在该项信息
0x89	ZDP_NO_DESCRIPTOR	短地址不是当前设备的
0x8a	ZDP_INSUFFICIENT_SPACE	无存储空间
0x8b	ZDP_NOT_PERMITTED	当前状态不支持该操作
0x8c	ZDP_TABLE_FULL	表格存储已满
0x8d	ZDP_NOT_AUTHORIZED	操作未被认证通过
0x8e	ZDP_BINDING_TABLE_FULL	绑定表已满

3.1.6. 无线通信状态表

无线通信状态表

- ✧ APS ACK 是消息经过多跳后回复的，可以保证消息到达最终目的地。
- ✧ MAC ACK 是消息直接传给设备回复的，不能保证消息到达最终目的地。

状态值	状态描述
0x00	通信成功
0xcd	发送失败，目标设备不存在
0xb7	没收到 APS ACK
0xe1	信道干扰导致丢包
0xe9	没收到 MAC ACK
0xf0	发送给休眠设备的消息没发出去超时了
0xf1	发送队列满了

3.2. ZDO 命令解析

此处仅解析所有 ZDO 命令的扩展参数部分，每一条下发 ZDO 命令对应一条收到 ZDO 命令。

3.2.1. NWK_addr_req 和 NWK_addr_rsp，查询短地址

功能描述：根据 IEEE 地址查询设备短地址，该命令只能使用广播地址 0xFFFD 发送到所有路由设备进行查询。

发：

名称	命令 ID	扩展参数		
	0x0000	查子节点	子节点 index	IEEE 地址
字节数	2	1	1	8

- 查子节点：默认填 0
- 子节点 index：默认填 0
- IEEE 地址：要查询的设备的 IEEE 地址

收

名称	命令 ID	扩展参数			
	0x8000	IEEE 地址	子节点 index	子节点个数	子节点短地址
字节数	2	8	1	1	2*N

- IEEE 地址：设备的 IEEE 地址。
- 子节点 index：和查询时填的一样
- 子节点个数：设备的子节点个数
- 子节点短地址：设备所有的子节点的短地址

该命令从“收到 ZDO 命令”的头部，获取设备短地址。

例

发送：A5 5A 12 0B 01 00 00 FD FF 00 00 DB 16 18 FE FF CC CC CC 11 5E

短地址 0xFFFD，IEEE 地址 DB 16 18 FE FF CC CC CC

收到 ACK：A5 5A 06 8B 01 00 07 3B C2，命令有效，分配无线序号 0x07

收到 ZDO 反馈：A5 5A 0A 40 01 00 00 FD FF 00 07 14 A5，无线序号 0x07 的命令发送成功

（仅此条命令解析 ACK 和 ZDO 反馈，后续的解析命令省略）

收到 ZDO 命令：A5 5A 14 41 81 00 80 AC 88 00 07 DB 16 18 FE FF CC CC CC 00 00 A4 BC，从该命令中解析出设备短地址 0x88AC，IEEE 地址与发出命令的 IEEE 地址相同。

3.2.2. IEEE_addr_req 和 IEEE_addr_rsp，查询 IEEE 地址

功能描述：查询设备的 IEEE 地址，该命令不支持广播查询

发

名称	命令 ID	扩展参数	
	0x0001	查子节点	子节点 index
字节数	2	1	1

- 查子节点：默认填 0
- 子节点 index：默认填 0

收

名称	命令 ID	扩展参数			
	0x8001	IEEE 地址	子节点 index	子节点个数	子节点短地址
字节数	2	8	1	1	2*N

- IEEE 地址：设备的 IEEE 地址。
- 子节点 index：和查询时填的一样
- 子节点个数：设备的子节点个数
- 子节点短地址：设备所有的子节点的短地址

该命令返回结构和 NWK_addr_rsp 的一样

例

发送：A5 5A 0A 0B 01 01 00 AC 88 00 00 B1 D3，短地址 0x88AC

ACK 和反馈省略

收到：A5 5A 14 63 81 01 80 AC 88 00 08 DB 16 18 FE FF CC CC CC 00 00 FF 4A

短地址 0x88AC，IEEE 地址 DB 16 18 FE FF CC CC CC

3.2.3. Simple_Desc_req 和 Simple_Desc_rsp，查询端口信息

功能描述：查询设备端点的 Device ID，Profile ID 和 cluster ID。该命令查询到的内容和“设备功能消息”上报的一致。

发

名称	命令 ID	扩展参数
	0x0004	设备端口
字节数	2	1

- 设备端口：被查询的设备端口号

收

名称	命令 ID	扩展参数					
	0x8004	设备端口	Profile ID	Device ID	保留	In Cluster 列表	Out Cluster 列表
字节数	2	1	2	2	1	变长	变长

- 设备端口：被查询的设备端口。
- Profile ID：该端口的 Profile ID。
- Device ID：该端口的 Device ID。
- In Cluster 列表：见“设备功能消息”中的《cluster 列表》。
- Out Cluster 列表：见“设备功能消息”中的《cluster 列表》。

例

发送：A5 5A 09 0C 01 04 00 A7 1A 01 5C 52，目标短地址是 0x1AA7，查询端口 0x01

收到：A5 5A 22 56 81 04 80 A7 1A 00 1E 01 04 01 01 01 00 07 00 00 03 00 04 00 05 00 06 00 08 00 00 10 01 19 00 E9 C6，该命令是目标 0x1AA7 的端口 0x01 返回的命令

Profile ID = 0x0104

Device ID = 0x0101，查《Device ID 表》发现是可调光灯

In Cluster 一共 7 个，可以看出支持设备标记，分组，场景，开关灯，调光的功能

Out Cluster 只有 1 个，用于 OTA 升级

3.2.4. Acitve_EP_req 和 Acitve_EP_rsp，查询设备端口数

功能描述：查询目标设备的端口数和所有端口 ID，如果“设备功能消息”获取到完整的信息，该命令没有什么用。

发

名称	命令 ID	扩展参数
	0x0005	无
字节数	2	0

该命令无扩展参数

收

名称	命令 ID	扩展参数	
	0x8005	端口总数	设备端口列表
字节数	2	1	变长

- ✧ 端口总数：被查询设备的端口总数
- ✧ 设备端口列表：列出该设备支持的所有端口

例

发送：A5 5A 08 09 01 05 00 A7 1A C6 75，查询设备短地址 0x1AA7 的全部 Endpoint

收到：A5 5A 0E 7E 81 05 80 A7 1A 00 1F 03 01 02 F2 A4 BE，查到该设备共 3 个 Endpoint，分别是 0x01，0x02，0xF2。和“设备功能消息”中得到的信息一致

3.2.5. Bind_req 和 Bind_rsp，设置设备与设备绑定

功能描述：设置一个设备绑定另一个设备。当一个设备设置绑定另一个设备后，两个设备之间直接通过 IEEE 地址寻找对方，并使用正确的通信协议互相控制，不需要网关介入。

发

名称	命令 ID	扩展参数		
	0x0021	绑定 cluster	源长地址	目标长地址
字节数	2	2	9	9

- ✧ 绑定 cluster：参与绑定的 cluster ID，源设备和目标设备的 cluster ID 一定要互补绑定才有意义。
- ✧ 源长地址：源设备的长地址，长地址见 2.2.5“设备功能消息”一节，设备的端口号+IEEE 地址小端模式组成 9 字节的长地址。
- ✧ 目标长地址：目标设备的长地址，绑定单个设备时输入设备端口号+设备 IEEE 地址，绑定分组时输入 0xFF + 2 字节组地址 + 6 个 0xFF

收

名称	命令 ID	扩展参数
	0x8021	空
字节数	2	0

绑定返回命令，仅返回一个带有 ZDO 状态的空帧头。

3.2.6. Unind_req 和 Unind_rsp，解除设备之间的绑定

功能描述：两个已经建立绑定的设备，解除绑定关系。

发

名称	命令 ID	扩展参数		
	0x0022	绑定 cluster	源长地址	目标长地址
字节数	2	2	9	9

- ✧ 绑定 cluster：需要解除绑定的 cluster ID。
- ✧ 源长地址：源设备的长地址。
- ✧ 目标长地址：目标设备的长地址。

收

名称	命令 ID	扩展参数
	0x8022	空
字节数	2	0

绑定返回命令，仅返回一个带有 ZDO 状态的空帧。

3.2.7. Mgmt_Bind_req 和 Mgmt_Bind_rsp，查看设备的绑定表

功能描述：通过读取设备的绑定表，查看设备绑定了多少个其它目标设备。

发

名称	命令 ID	扩展参数
	0x0033	起始索引
字节数	2	1

- ✧ 起始索引：查询绑定表的第 X 个条目，可以返回以 X 为起始的 3~4 个绑定目标。

收

名称	命令 ID	扩展参数			
	0x8033	绑定总数	起始索引	返回条数 N	绑定表信息
字节数	2	1	1	1	20*N

- ✧ 绑定总数：设备总共绑定目标数。
- ✧ 起始索引：本次查询返回的第一条信息的索引。
- ✧ 返回条数：本次查询返回的条数。
- ✧ 绑定表信息：见《绑定表信息》

	绑定表信息		
名称	绑定 cluster	源长地址	目标长地址

字节数	2	9	9

- ✧ 绑定 cluster：该条记录绑定的 cluster。
- ✧ 源长地址：该条绑定记录的源长地址，由于源长地址中的 IEEE 地址已经等于设备本身 IEEE 地址，所以只需要关注端口号。
- ✧ 目标长地址：绑定目标的

3.2.8. Mgmt_Leave_req 和 Mgmt_Leave_rsp，主动删除设备

功能描述：根据设备 IEEE 地址删除指定设备。如果是删除休眠设备，需要把指令发给它的父节点。也可以广播删除，但是协调器会收到多个路由设备返回“没有找到设备”的冗余信息。

发

名称	命令 ID	扩展参数		
	0x0034	IEEE 地址	切换父节点	删除子节点
字节数	2	8	1	1

- ✧ IEEE 地址：需要被删掉的设备的 IEEE 地址。
- ✧ 切换父节点：此处填 0，如果填 1 不能删除设备。
- ✧ 删除子节点：此处填 0，不需要删除的子节点可以切换到其它父节点。

收

名称	命令 ID	扩展参数
	0x8034	无
字节数	2	0

例

发送：A5 5A 12 45 01 34 00 A7 1A 70 0F 00 00 AD 38 C1 A4 00 00 3C 09，删除 IEEE 地址为

收到：A5 5A 0A 24 81 34 80 A7 1A 00 21 15 E9

设备删除成功后，会伴随收到多条“设备离网消息”和“设备注销消息”

A5 5A 12 25 80 87 A7 1A 70 0F 00 00 AD 38 C1 A4 00 00 00 CD FF

A5 5A 11 26 80 86 A7 1A 70 0F 00 00 AD 38 C1 A4 00 00 15 4C

A5 5A 11 27 80 86 A7 1A 70 0F 00 00 AD 38 C1 A4 C3 41 44 8C

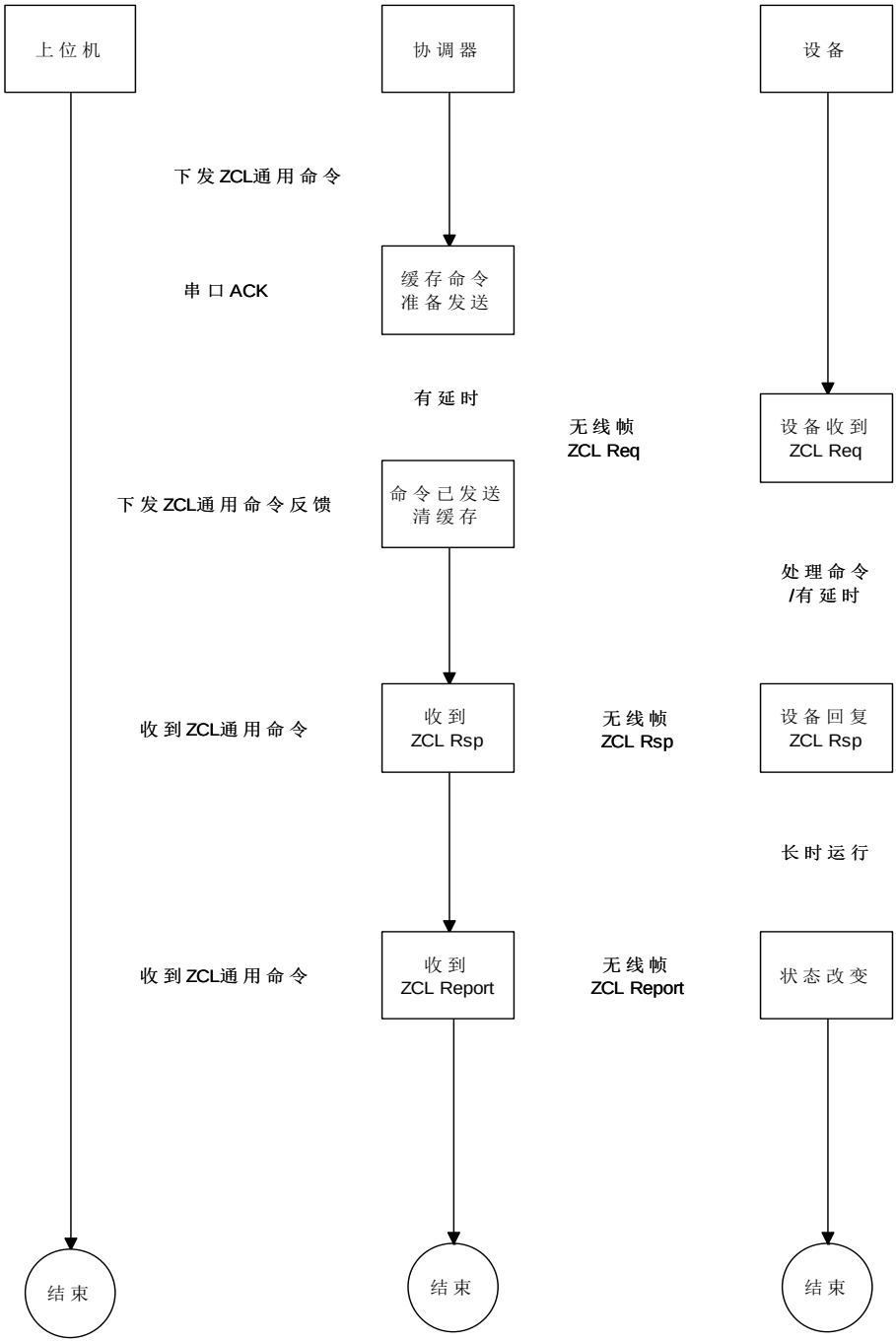
A5 5A 11 28 80 86 A7 1A 70 0F 00 00 AD 38 C1 A4 C3 41 0B 88

4. ZCL 通用命令

ZCL 是 ZigBee Cluster Library 的缩写，ZCL 即是 ZigBee 设备应用控制功能。

ZCL 通用命令用于对设备的各个属性（即状态参数）的所有操作，包括读，写，主动上报等操作。

和 ZDO 命令一样，ZCL 通用命令也是无线控制命令，会产生命令反馈。下发 ZCL 通用命令到达协调器会产生 ZCL Req 命令。收到 ZCL 通用命令有两种，一种是与下发 ZCL 通用命令相关的，属于 ZCL Rsp 命令，另一种与下发 ZCL 通用命令无关，属于 ZCL Report 命令。



ZCL 通用命令通信流程

✧

4.1. ZCL 通用命令帧格式

1.1.1.

4.1.1. 下发 ZCL 通用命令，命令大类 0x02，下行

下行：

名称	串口帧 Payload										
	接口控制	短地址	端口	Profile ID	Cluster ID	命令 ID	请求序号	方向	应答	厂商码	ZCL Payload
字节数	1	2	1	2	2	1	1	1	1	2	变长

- ✧ 接口控制：见“接口控制”表
- ✧ 短地址：目标设备的短地址，组播模式则填组 ID。
- ✧ 端口：目标设备的端口，组播模式填 0xFF。
- ✧ Profile ID：填 0x0104 的小端模式，即 04 01
- ✧ Cluster ID：填 Cluster ID 的小端模式
- ✧ 命令 ID：见《ZCL 通用命令表》
- ✧ 请求序号：上位机生成一个 0~255 循环的请求序号，收到 ZCL 通用命令后匹配请求序号
- ✧ 方向：填 0 访问 In Cluster 下的属性，填 1 访问 Out Cluster 下的属性。
- ✧ 应答：应答标志，0 – 使能默认返回，1 – 关闭默认返回，2 – 关闭默认返回同时开启 APS ACK 功能。默认返回功能见 4.2.6“ZCL_CMD_DEFAULT_RSP”。
- ✧ 厂商码：针对私有协议定义的 Cluster 和属性需要填入正确的厂商码，绝大多数时候直接填 0x0000。
- ✧ ZCL Payload：ZCL 通用命令的 Payload。

AKC：

名称	状态	扩展参数
		无
字节数	1	0

- ✧ 状态：见《ACK 状态表》

接口控制表

名称	无效标志	端口编号	IEEE 地址标志	保留	地址模式
bit	7	[4:6]	3	2	[1:0]

- ✧ 无效标志：设备发给协调器的 Client2Server 命令，协调器不支持。
- ✧ 端口编号：协调器累计支持 8 个端口，0 号 Endpoint 为 1，Profile 为 0x0104；1 号端口用于 OTA 升级；2 号端口用于 Green Power；其余参见“2.1.14 增加动态 Profile 和 Endpoint”。
- ✧ IEEE 地址：该标志为 1 时，2 字节短地址变成 8 字节长地址。
- ✧ 地址模式：0-单播，1-组播，2-广播，3-单播需回复 Default Response。

4.1.2. 下发 ZCL 通用命令反馈，命令大类 0x02，上行

上行：

名称	串口帧 Payload								
	接口控制	短地址	端口	Profile ID	Cluster ID	命令 ID	请求序号	方向	无线状态
字节数	1	2	1	2	2	1	1	1	1

- ✧ 接口控制：与上位机发送“下发 ZCL 通用命令”的相同。
- ✧ 短地址：与上位机发送“下发 ZCL 通用命令”的相同。
- ✧ 端口：与上位机发送“下发 ZCL 通用命令”的相同。
- ✧ Profile ID：与上位机发送“下发 ZCL 通用命令”的相同。
- ✧ Cluster ID：与上位机发送“下发 ZCL 通用命令”的相同。
- ✧ 命令 ID：与上位机发送“下发 ZCL 通用命令”的相同。
- ✧ 请求序号：与上位机发送“下发 ZCL 通用命令”的相同。
- ✧ 方向：与上位机发送“下发 ZCL 通用命令”的相同。

✧ 见《无线通信状态表》。

4.1.3. 收到 ZCL 通用命令，命令大类 0x82，上行

上行：

名称	串口帧 Payload										
	接口控制	短地址	端口	Profile ID	Cluster ID	命令 ID	请求序号	方向	信号	厂商码	ZCL Payload
字节数	1	2	1	2	2	1	1	1	1	2	变长

✧ 接口控制：见“接口控制”表

✧ 短地址：发送消息给协调器的设备短地址

✧ Profile ID：收到的 Profile ID 只能是 0x0104。

✧ Cluster ID：收到命令的 cluster ID。

✧ 命令 ID：见《ZCL 通用命令表》。

✧ 请求序号：由设备端产生，如果请求序号与下发 ZCL 通用命令的一致，则说明该命令是下发 ZCL 通用命令的 ZCL Rsp。

✧ 方向：设备端如果是 In Cluster 发出来的为 1，是 Out Cluster 发出来的则为 0。如果是 ZCL Rsp 帧，则和“下发 ZCL 通用命令”的方向相反。

✧ 信号：信号强度，0~255 的百分比，越大信号越好。

✧ 厂商码：私有协议定义的厂商码。

✧ Payload：收到命令附带的数据 Payload。

4.1.4. ZCL 通用命令表

ZCL 通用命令表			
命令 ID	描述	功能	收发
0x00	ZCL_CMD_READ	读属性请求	下行
0x01	ZCL_CMD_READ_RSP	读属性返回	上行
0x02	ZCL_CMD_WRITE	写属性请求	下行
0x04	ZCL_CMD_WRITE_RSP	写属性返回	上行
0x06	ZCL_CMD_CONFIG_REPORT	设置属性上报请求	下行
0x07	ZCL_CMD_CONFIG_REPORT_RSP	设置属性上报返回	上行
0x08	ZCL_CMD_READ_REPORT_CFG	查询属性上报请求	下行
0x09	ZCL_CMD_READ_REPORT_CFG_RSP	查询属性上报返回	上行
0x0A	ZCL_CMD_REPORT	属性主动上报	上行
0x0B	ZCL_CMD_DEFAULT_RSP	默认返回（Default Response）	上行
0x0C	ZCL_CMD_DISCOVER_ATTRS	查询所有属性	下行
0x0D	ZCL_CMD_DISCOVER_ATTRS_RSP	查询所有属性返回	上行

4.1.5. ZCL 属性数据类型表

ZCL 属性数据类型表					
类别	数据类型	ID	字节数	无效值	Report 对齐
NULL	nodata	0x00	0		0

普通数据	data8	0x08	1		0
	data16	0x09	2		0
	data24	0x0a	3		0
	data32	0x0b	4		0
	data40	0x0c	5		0
	data48	0x0d	6		0
	data56	0x0e	7		0
	data64	0x0f	8		0
逻辑数据	bool	0x10	1	0xff	0
二进制位数据	map8	0x18	1		0
	map16	0x19	2		0
	map24	0x1a	3		0
	map32	0x1b	4		0
	map40	0x1c	5		0
	map48	0x1d	6		0
	map56	0x1e	7		0
	map64	0x1f	8		0
无符号整数	uint8	0x20	1		4
	uint16	0x21	2		4
	uint24	0x22	3		4
	uint32	0x23	4		4
	uint40	0x24	5		8
	uint48	0x25	6		8
	uint56	0x26	7		8
	uint64	0x27	8		8
有符号整数	int8	0x28	1		4
	int16	0x29	2		4
	int24	0x2a	3		4
	int32	0x2b	4		4
	int40	0x2c	5		8
	int48	0x2d	6		8
	int56	0x2e	7		8
	int64	0x2f	8		8
枚举	enum8	0x30	1	0xff	0
	enum16	0x31	2	0xffff	0
浮点	semi	0x38	2		4
	single	0x39	4		4

	double	0x3a	8		8
字符串	octstr	0x41	第一字节	头为 0xff	0
	string	0x42	第一字节	头为 0xff	0
	octstr16	0x43	第一双字节	头为 0xffff	0
	string16	0x44	第一双字节	头为 0xffff	0
序列型	array	0x48	2+内容长度总和	头为 0xffff	0
	struct	0x4C	2+内容长度总和	头为 0xffff	0
时间	ToD	0xe0	4	0xffffffff	4
	date	0xe1	4	0xffffffff	4
	UTC	0xe2	4	0xffffffff	4
标识符	clusterID	0xe8	2	0xffff	0
	attriID	0xe9	2	0xffff	0
	bacOID	0xea	4	0xffffffff	0
其它数据	EUI64	0xf0	8	0xffffffff	0
	key128	0xf1	16		0

4.1.6. ZCL 状态表

ZCL 状态表		
Value	描述	出现的情况
0x00	操作成功	全部命令
0x01	操作失败	全部命令
0x7E	该操作未授权	读写 Attribute 时
0x80	命令格式不正确	发送专有命令
0x81	不支持此 ZCL 专有命令	发送专有命令
0x82	不支持此 ZCL 通用命令	发送通用命令
0x83	不支持厂商定义 ZCL 专有命令	发送带厂商 ID 专有命令
0x84	不支持厂商定义 ZCL 通用命令	发送带厂商 ID 通用命令
0x85	无效字段	专有命令的参数错误
0x86	不支持的 Attribute	通用命令
0x87	错误的输入值	全部命令
0x88	Attribute 只读	写 Attribute 时
0x89	空间不足	专有命令（带存储功能）

0x8A	存在重复项	专有命令（带存储功能）
0x8B	没找到	专有命令（带存储功能）
0x8C	Attribute 不支持主动上报	配置主动上报或读配置
0x8D	数据类型无效	通用命令带数据类型
0x8E	选项无效	专有命令
0x8F	Attribute 只写	读 Attitude 时
0x90	启动状态不一致	
0x91	Out Of Band	
0x92	不一致错误	
0x93	拒绝此操作	
0x94	超时	
0x95	Abort	OTA 时
0x96	无效的 image 数据	OTA 时
0x97	等待数据	OTA 或其它大数据传输
0x98	没有 image 文件	OTA 时
0x99	需要更多的 image 数据	OTA 时
0xc0	硬件错误	
0xc1	软件错误	
0xc2	校准错误	

4.1.7. 常见状态属性表

	常见的状态属性列表					
类别	状态名称	Cluster ID	Attr ID	数据类型	方向	访问方式
设备信息	ZigBee 版本	0x0000	0x0000	uint8	0	只读
	软件版本		0x0001	uint8	0	只读
	协议版本		0x0002	uint8	0	只读
	硬件版本		0x0003	uint8	0	只读
	厂商名称		0x0004	string	0	只读
	产品型号		0x0005	string	0	只读
	生产日期		0x0006	string	0	只读
	电源方式		0x0007	enum8	0	只读
	软件日期		0x4000	string	0	只读
标记	标记时间	0x0003	0x0000	uint16	0	读写
分组	支持分组名	0x0004	0x0000	map8	0	只读
场景	已加场景数	0x0005	0x0000	uint8	0	只读
	当前场景		0x0001	uint8	0	只读
	当前场景组		0x0002	uint16	0	只读
	场景已执行		0x0003	bool	0	只读
开关	开关状态	0x0006	0x0000	bool	0	只读/上报

亮度	亮度状态	0x0008	0x0000	uint8	0	只读/上报
光感	当前光感值	0x0400	0x0000	uint16	0	只读/上报
	最小光感值		0x0001	uint16	0	只读
	最大光感值		0x0002	uint16	0	只读
安防	安防状态	0x0500	0x0000	enum8	0	只读
	安防类型		0x0001	enum16	0	只读
	警报状态		0x0002	map16	0	只读
	接收器地址		0x0010	uint8	0	只读
	安防 ID		0x0011	uint8	0	只读
OTA 状态	服务端 IEEE	0x0019	0x0000	IEEE	1	只读
	当前版本		0x0002	uint32	1	只读
		注： 1，同 Cluster ID 的状态，可以同时读，不同 cluster ID 的状态分多次读 2，除了将来可能用到的 OTA 状态和多媒体设备状态方向为 TO CLIENT，绝大多数状态都是 TO SERVER 3，情景面板没有实际物理意义的状态，可以用 ZigBee 版本，软件版本，协议版本，硬件版本任意一个状态作为心跳包。凡是有物理意义的设备都以物理意义状态做心跳包，没有的才用任意版本做心跳包。				

4.2. ZCL 通用命令解析

说明：ZCL 通用命令仅解析 ZCL Payload 部分

4.2.1. ZCL_CMD_READ 与 ZCL_CMD_READ_RSP

功能描述：读属性值，可以同时读取一个 cluster ID 下的多个属性。

发送：

名称	命令 ID	ZCL Payload	
	0x00	属性数量 N	属性 ID 数组
字节数	1	1	变长（2*N）

- ✧ 属性数量 N：属性数量 N
- ✧ 属性 ID 数组：需要读取的属性 ID 写成数组（小端模式）

收到：

名称	命令 ID	ZCL Payload	
	0x01	读属性数量 N	读属性返回列表
字节数	1	1	N*变长

- ✧ 读属性数量 N：属性单元数量 N。
- ✧ 读属性列表：由 N 个读属性返回单元构成，见《读属性返回单元》

名称	读属性返回单元			
	属性 ID	ZCL 状态	数据类型	属性数据值
字节数	2	1	1 或 0	变长

- ✧ 属性 ID：读属性返回的属性 ID

- ZCL 状态：读取该属性 ID 返回的 ZCL 状态，见《ZCL 状态表》
- 数据类型：只有返回 ZCL 状态为“操作成功”才有该项，该项用于描述“属性数据值”的字节数，见《ZCL 属性数据类型表》
- 属性数据值：返回的属性数据值

例，命令 ID 颜色

发送：A5 5A 1F 0D 02 00 C0 90 02 04 01 06 FC 00 35 00 00 00 00 06 00 00 01 00 02 00 03 00 04 00 05 00 97 9B

目标是电源 0x90C0 的端口 2，读取 cluster=0xFC06 即恒照度对应的属性。总共读取 6 个属性，分别是 0x0000, 0x0001, 0x0002, 0x0003, 0x0004, 0x0005, 0x0006

收到：A5 5A 30 6D 82 00 C0 90 02 04 01 06 FC 01 35 01 66 00 00 06 00 00 00 21 92 00 01 00 00 21 FF FF 02 00 00 20 FF 03 00 86 04 00 86 05 00 00 21 FF FF E0 6A

返回从电源 0x90C0 的端口 2 读取的关于 cluster=0xFC06 即恒照度的属性值，返回 6 个属性值的结果，属性 0x0000 是 uint16 的传感器值=0x0092，属性 0x0001 是 uint16 的关灯光照度=0xFFFF，属性 0x0002 是 uint8 的最小亮度=0xFF，属性 0x0003 和 0x0004 返回状态 0x86 即该属性在设备上不存在，属性 0x0005 是恒照度基准值=0xFFFF。

4.2.2. ZCL_CMD_WRITE 与 ZCL_CMD_WRITE_RSP

功能描述：写属性值，可以一次写多个属性，但是下发的属性 ID 必须存在于目标设备上，且下发命令中的属性对应的数据类型要一致，并且该属性支持写操作。

发送：

名称	命令 ID	ZCL Payload	
	0x02	属性数量 N	写属性请求列表
字节数	1	1	N*变长

- 属性数量 N：属性单元数量 N。
- 写属性请求列表：由 N 个写属性请求单元构成，见《写属性请求单元》

名称	写属性请求单元		
	属性 ID	数据类型	数据值
字节数	2	1	变长

- 属性 ID：需要被写入的属性 ID
- 数据类型：属性 ID 对应的数据类型
- 数据值：新的属性数据值

收到：

名称	命令 ID	ZCL Payload	
	0x04	属性数量 N	写属性返回列表
字节数	1	1	N*3

- 属性数量 N：属性单元数量 N，通常只罗列写失败的属性，泰凌微芯片除外。
- 写属性返回列表：由 N 个写属性返回单元构成，见《写属性返回单元》。

名称	写属性返回单元	
	属性 ID	ZCL 状态
字节数	2	1

- 属性 ID：写属性返回的属性 ID，通常只返回写失败的属性，泰凌微除外。

✧ ZCL 状态：见《ZCL 状态表》。

例

发送：A5 5A 1D 0D 02 00 C0 90 02 04 01 06 FC 02 35 00 00 00 00 02 00 00 21 FF FF 05 00 21 90 00 6D 7E

向 LED 电源的光感端口，写入恒照度相关的 2 个属性 0x0000 和 0x0005，属性 0x0000 写成 0xFFFF ，属性 0x0005 写成 0x0090。

收到：A5 5A 19 73 82 00 C0 90 02 04 01 06 FC 04 35 01 5D 00 00 02 00 00 88 05 00 00 3E 7C

属性 ID 0x0000 返回状态 0x88，该属性为只读。属性 ID 0x0005 写入成功，打开了恒照度模式。

4.2.3. ZCL_CMD_CONFIG_REPORT 与 ZCL_CMD_CONFIG_REPORT_RSP

功能描述：用于设置某些属性值自动上报（以 ZCL_CMD_REPORT 形式），前提条件是需要对设备的 cluster 设置绑定目标。

发送：

名称	命令 ID	ZCL Payload	
	0x06	属性数量 N	配置属性上报请求列表
字节数	1	1	N*变长

✧ 属性数量 N：属性单元数量 N

✧ 配置属性上报请求列表：由 N 个配置属性上报单元构成，见《配置属性上报请求单元》

名称	配置属性上报请求单元					
	属性 ID	模式	最小时间	最大时间	数据类型	变量 Delta
字节数	2	1	2	2	1	0/4/8

✧ 属性 ID：配置上报的属性 ID

✧ 模式：填 0x00

✧ 最小时间：两次连续上报的最小间隔时间，设置成 0xFFFF 则一直不上报。

✧ 最大时间：两次连续上报的最大间隔时间，可以当成心跳包用。

✧ 数据类型：该属性的数据类型

✧ 变量 Delta：属性值变化量超过变量 Delta 触发上报，根据数据类型分别有 0 字节，4 字节，8 字节 3 种模式，见《ZCL 属性数据类型表》中 Report 对齐。

收到：

名称	命令 ID	ZCL Payload	
	0x07	属性数量 N	配置属性上报返回列表
字节数	1	1	N*4

✧ 属性数量：属性单元数量 N，通常只返回配置失败的属性

✧ 配置属性上报结果列表：有 N 个配置属性上报返回单元构成，见《配置属性上报返回单元》。

名称	配置属性上报返回单元		
	属性 ID	ZCL 状态	模式
字节数	2	1	1

✧ 属性 ID：配置上报的属性 ID。

✧ ZCL 状态：配置属性上报的结果，通常只返回配置失败的属性。

✧ 模式：默认为 0x00，与下发配置属性上报命令时一致。

4.2.4. ZCL_CMD_READ_REPORT_CFG 与 ZCL_CMD_READ_REPORT_CFG_RSP

功能描述：查询属性自动上报的配置信息。

发送：

名称	命令 ID	ZCL Payload	
	0x08	属性数量 N	查询属性上报请求列表
字节数	1	1	N*变长

- ✧ 属性数量 N：属性单元数量 N
- ✧ 查询属性上报请求列表：由 N 个查询属性上报请求单元构成，见《查询属性上报请求单元》。

名称	查询属性上报请求单元	
	属性 ID	模式
字节数	2	1

- ✧ 属性 ID：查询属性上报的属性 ID
- ✧ 模式：默认填 0x00

收到：

名称	命令 ID	ZCL Payload	
	0x09	属性数量 N	查询属性上报返回列表
字节数	1	1	N*变长

- ✧ 属性数量 N：属性单元数量 N
- ✧ 查询属性上报返回列表：由 N 个查询属性上报返回单元构成，见《查询属性上报返回单元》

名称	查询属性上报返回求单元						
	属性 ID	ZCL 状态	模式	最小时间	最大时间	数据类型	变量 Delta
字节数	2	1	1	2	2	1	0/4/8

属性 ID：查询属性上报的属性 ID

ZCL 状态：见《ZCL 状态表》

模式：返回为 0x00，与请求命令一致

最小时间：属性上报的最小时间，0xFFFF 是关闭属性上报。

最大时间：属性上报的最大间隔时间。

数据类型：查询的属性的数据类型，见《ZCL 属性数据类型表》。

变量 Delta：属性值变化量超过变量 Delta 触发上报，根据数据类型分别有 0 字节，4 字节，8 字节 3 种模式，见《ZCL 属性数据类型表》中 Report 对齐。

4.2.5. ZCL_CMD_REPORT

功能描述：设备的属性值主动上报给协调器

收到：

名称	命令 ID	ZCL Payload	
	0x0A	属性数量 N	属性上报列表
字节数	1	1	N*变长

- ✧ 属性数量 N：属性单元数量 N。

✧ 属性上报列表：由 N 个属性上报单元构成，见《属性上报单元》

名称	属性上报单元		
	属性 ID	数据类型	属性数据值
字节数	2	1	变长

✧ 属性 ID：上报的属性 ID
✧ 数据类型：见《ZCL 属性数据类型表》
✧ 属性数据值：上报的属性数据值

4.2.6. ZCL_CMD_DEFAULT_RSP（收发均为此结构）

功能描述：默认返回 Default Response 是一条“万能指令”，ZCL 通用命令和 ZCL 专有命令都能产生该命令。该命令可以回复一切请求命令，并反馈操作结果的 ZCL 状态，返回的请求序号与下发请求命令的请求序号相等。

最新固件，协调器可按此 Payload 结构发送该帧数据。

名称	命令 ID	ZCL Payload	
	0x0B	返回命令 ID	ZCL 状态
字节数	1	1	1

✧ 返回命令 ID：与请求命令的命令 ID 相同，通常使用默认返回命令，返回 ZCL 状态来表示设备端不支持协调器发来的命令。
✧ ZCL 状态：见《ZCL 状态表》

注意事项：

✧ 默认返回命令通常只返回 ZCL 专有命令，只有在目标设备不支持某个 ZCL 通用命令时才使用默认返回命令回复协调器，并且 ZCL 状态都是报错。
✧ 下行命令头中“应答标志”中选择“禁用默认返回”，只能禁止目标设备回复 ZCL 状态为“操作成功”的默认返回。
✧ 上位机收到默认返回，不能根据返回命令 ID 来判断回复对应的是 ZCL 通用命令还是 ZCL 专有命令，而是要结合设备的短地址，请求序号两个条件来判断该返回对应哪条指令。默认返回通常是“报忧不报喜”的原则，因此上位机不能丢弃任何默认返回。
✧ 上位机收到“Client to Sever”的专有命令后，如果该专有命令无固定回复格式，可以用该命令回复对方。

4.2.7. ZCL_CMD_DISCOVER_ATTRS 与 ZCL_CMD_DISCOVER_ATTRS_RSP

功能描述：查询 Cluster 下所有的 Attribute。

发送：

名称	命令 ID	ZCL Payload	
	0x0C	查询属性个数	起始属性 ID
字节数	1	1	2

✧ 查询属性个数：期望查询返回的属性个数
✧ 起始属性 ID：期望查询的起始属性 ID

收到：

名称	命令 ID	ZCL Payload		
	0x0D	查询完成标志	返回属性个数	查询属性返回列表
字节数	1	1	1	3*N

- ✧ 查询完成标志：返回的属性是否完整（包含最后一个属性）。
- ✧ 返回属性个数：查询属性返回个数，小于或等于发送命令时填入的“查询属性个数”。
- ✧ 查询属性返回列表：查询属性时返回的属性信息，见《查询属性返回单元》

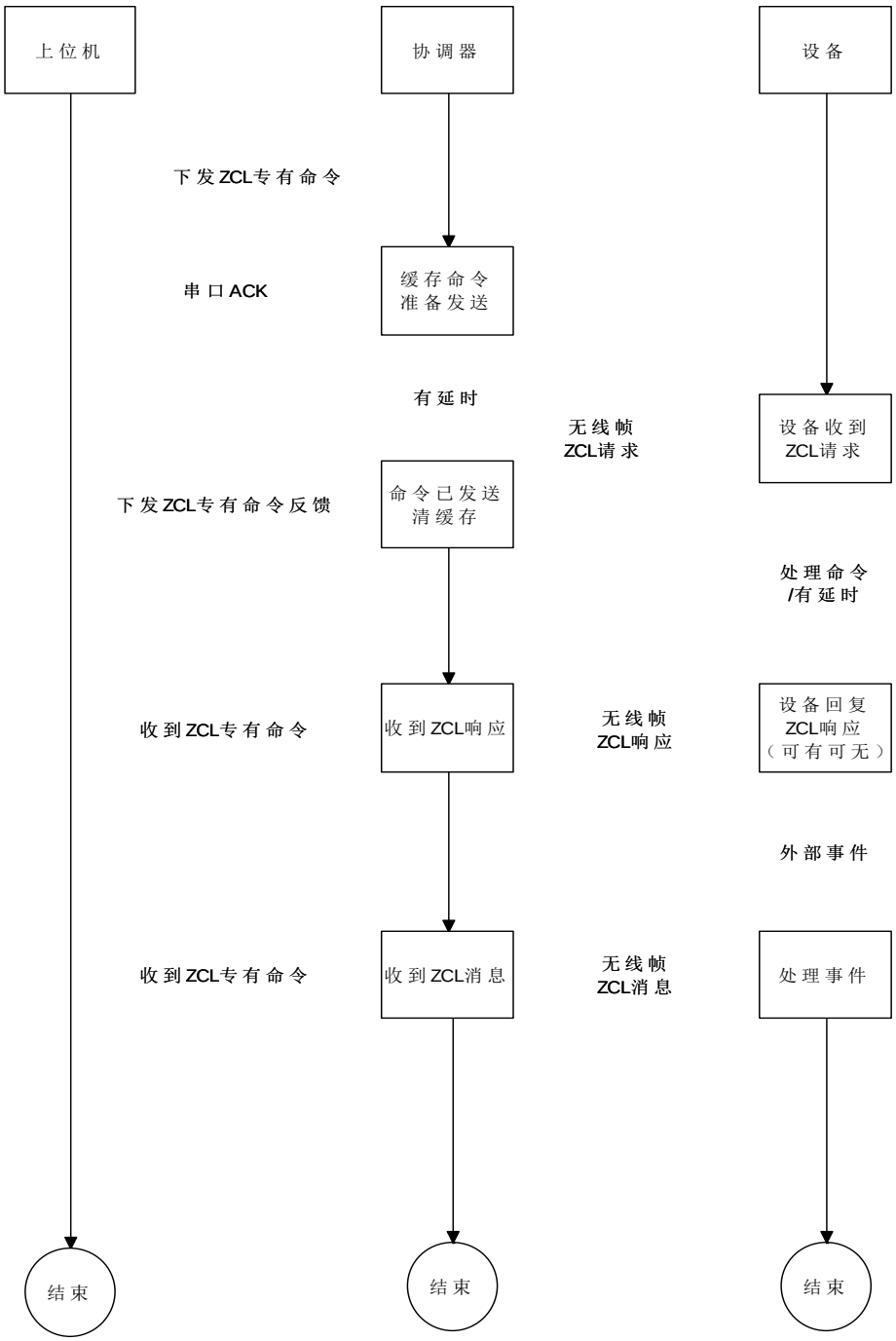
名称	查询属性返回单元	
	属性 ID	数据类型
字节数	2	1

- ✧ 属性 ID：查询到的属性 ID
- ✧ 数据类型：见《ZCL 属性数据类型表》

5. ZCL 专有命令

ZCL 专有命令有与 ZCL 通用命令相同结构的命令头部，但是 ZCL Payload 部分没有严格的结构。ZCL 专有命令用于不同功能设备的控制和访问，可以是很随意的数组或结构体。

通用命令与专有命令对比		
对比项目	ZCL 通用命令	ZCL 专有命令
改变设备状态方式	直接改变属性值	控制设备间接改变
载荷类型	零散的单个变量	复杂的数据
结构性	不同应用场景有统一的结构	不同应用场景自定义各自的结构
回复响应命令	必须有	可有可无



✧

5.1. ZCL 专有命令帧结构

ZCL 专有命令有和 ZCL 通用命令相似的结构

5.1.1. 下发 ZCL 专有命令，命令大类 0x03，下行

下行：

名称	串口帧 Payload										
	接口控制	短地址	端口	Profile ID	Cluster ID	命令 ID	请求序号	方向	应答	厂商码	ZCL Payload

字节数	1	2	1	2	2	1	1	1	1	2	变长

- ✧ 接口控制：见“接口控制”表
- ✧ 短地址：目标设备的短地址，组播模式则填组 ID。
- ✧ 端口：目标设备的端口，组播模式填 0xFF。
- ✧ Profile ID：填 0x0104 的小端模式，即 04 01。
- ✧ Cluster ID：填 Cluster ID 的小端模式
- ✧ 命令 ID：ZCL 专有命令的命令 ID，需要与命令方向配合使用。见《ZCL 专用命令表》，专有命令会不断更新。
- ✧ 请求序号：上位机生成一个 0~255 循环的请求序号，收到 ZCL 专有命令后匹配请求序号
- ✧ 方向：填 0 访问 In Cluster 下的属性，填 1 访问 Out Cluster 下的属性。
- ✧ 应答：应答标志，0 – 使能默认返回，1 – 关闭默认返回，2 – 关闭默认返回同时开启 APS ACK 功能。默认返回功能见 4.2.6“ZCL_CMD_DEFAULT_RSP”。
- ✧ 厂商码：针对私有协议定义的 Cluster 和属性需要填入正确的厂商码，绝大多数时候直接填 0x0000。
- ✧ ZCL Payload：ZCL 专有命令的 Payload。

AKC：

名称	状态	扩展参数
		无
字节数	1	0

- ✧ 状态：见《ACK 状态表》

5.1.2. 下发 ZCL 专有命令反馈，命令大类 0x03，上行

上行：

名称	串口帧 Payload								
	接口控制	短地址	端口	Profile ID	Cluster ID	命令 ID	请求序号	方向	无线状态
字节数	1	2	1	2	2	1	1	1	1

- ✧ 接口控制：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ 短地址：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ 端口：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ Profile ID：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ Cluster ID：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ 命令 ID：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ 请求序号：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ 方向：与上位机发送“下发 ZCL 专有命令”的相同。
- ✧ 见《无线通信状态表》。

5.1.3. 收到 ZCL 专有命令，命令大类 0x83，上行

上行：

名称	串口帧 Payload										
	接口控制	短地址	端口	Profile ID	Cluster ID	命令 ID	请求序号	方向	信号	厂商码	ZCL Payload
字节数	1	2	1	2	2	1	1	1	1	2	变长

- ✧ 接口控制：见“接口控制”表
- ✧ 短地址：发送消息给协调器的设备短地址
- ✧ Profile ID：收到的 Profile ID 只能是 0x0104
- ✧ Cluster ID：收到命令的 cluster ID
- ✧ 命令 ID：ZCL 专有命令的命令 ID，需要与命令方向配合使用。见《ZCL 专用命令表》，专有命令会不断更新。
- ✧ 请求序号：由设备端产生，如果请求序号与下发 ZCL 专有命令的一致，则说明该命令是下发 ZCL 专有命令的对应 ZCL 响应，如果不一致则为独立的 ZCL 消息。
- ✧ 方向：设备端如果是 In Cluster 发出来的为 1，是 Out Cluster 发出来的则为 0。一般情况下串口收到的都是 1。（目前只有场景控制命令是 0）
- ✧ 信号：信号强度，0~255 的百分比，越大信号越好。
- ✧ 厂商码：私有协议定义的厂商码。
- ✧ Payload：收到命令附带的数据 Payload。

新增无效命令：

协调器收到无效命令后上报网关，同时也会回复对方设备是无效命令。无效命令可用于检测空中的数据包的存在，从而判断一个设备的存在。

5.1.4. ZCL 专有命令相关表格

ZCL 专有命令表					
cluster	命令 ID	方向	描述符	功能	收发
0x0003 Identify 标记	0x00	0	IDENTIFY	设置设备标记	发
	0x01	0	IDENTIFY_QUERY	查询设备标记	发
	0x00	1	IDENTIFY_QUERY_RSP	查询标记返回	收
0x0004 Group 分组	0x00	0	GROUP_ADD	设备加组	发
	0x00	1	GROUP_ADD_RSP	设备加组返回	收
	0x01	0	GROUP_VIEW	查询组信息	发
	0x01	1	GROUP_VIEW_RSP	查询组信息返回	收
	0x02	0	GROUP_GET_MEMBERSHIP	查询设备加入的组	发
	0x02	1	GROUP_GET_MEMBERSHIP_RSP	查询设备加入的组返回	收
	0x03	0	GROUP_REMOVE	删除设备指定分组	发
	0x03	1	GROUP_REMOVE_RSP	删除设备指定分组返回	收
	0x04	0	GROUP_REMOVE_ALL	删除设备所有分组	发
	0x05	0	GROUP_ADD_IF_IDENTIFYING	标记状态设备自动加组	发
0x0005 Scene 场景	0x00	0	SCENE_ADD	添加场景	发
	0x00	1	SCENE_ADD_RSP	添加场景返回	收
	0x01	0	SCENE_VIEW	检验场景	发
	0x01	1	SCENE_VIEW_RSP	检验场景返回	收
	0x02	0	SCENE_REMOVE	删除场景	发
	0x02	1	SCENE_REMOVE_RSP	删除场景返回	收
	0x03	0	SCENE_REMOVE_ALL	删除所有场景	发
	0x03	1	SCENE_REMOVE_ALL_RSP	删除所有场景返回	收

	0x04	0	SCENE_STORE	保存场景	发
	0x04	1	SCENE_STORE_RSP	保存场景返回	是
	0x05	0	SCENE_RECALL	执行场景	发/收
	0x06	0	SCENE_GET_MEMBERSHIP	查询所有场景	发
	0x06	1	SCENE_GET_MEMBERSHIP_RSP	查询所有场景返回	收
0x0006 On Off 开关灯	0x00	0	OFF	关灯	发
	0x01	0	ON	开灯	发
	0x02	0	TOGGLE	反置	发
0x0008 Level 调亮度	0x00	0	MOVE_TO_LEVEL	调至亮度	发
	0x01	0	MOVE	自动变化亮度	发
	0x02	0	STEP	单步亮度	发
	0x03	0	STOP	停止自动亮度	发
	0x04	0	MOVE_TO_LEVEL_ONOFF	调至亮度带开关	发
	0x05	0	MOVE_ONOFF	自动变化亮度带开关	发
	0x06	0	STEP_ONOFF	单步亮度带开关	发
	0x07	0	STOP_ONOFF	停止自动亮度带开关	发
0x0500 IAS ZONE 红外安防	0x00	1	STATUS_CHANGE_NOTIFICATION	报警状态通知	收
	0x01	1	STATUS_ENROLL_REQUEST	安防注册请求	收
	0x00	0	STATUS_ENROLL_RESPONSE	安防注册响应	发
0xEF00 涂鸦协议	0x00	0	TUYA_RX	涂鸦模块接收	发
	0x01	1	TUYA_TX	涂鸦模块发送	收

定义各类命令下的状态属性格式，各项状态属性如没有说明，均为只读，可写和可配置上报会在状态的说明中强调

Cluster =			
状态名称	Attr ID	数据类型	方向

各条命令解析仅解析 ZCL Payload 部分，各个参数以变量类型表示

名称	命令 ID	方向	ZCL Payload
	0x00		
字节数	1	1	

5.2. 标记类命令，cluster=0x0003

功能描述：设置设备进入标记状态，设备可以进入闪烁状态，或者发出声音，供肉眼识别。同时进入标记状态的设备可以被其它设备自动搜索到，完成加组，保存场景等一系列动作。

5.2.1. 标记类的状态属性

标记类状态属性			
状态名称	Attr ID	数据类型	方向
标记时间	0x0000	uint16	0

✧ 标记时间：标记状态的时间，设备处于标记状态时值不等于 0，可以用写状态属性的方

式把设备设置成标记状态。（可写）

5.2.2. IDENTIFY，设置设备标记

功能描述：设备进入标记状态

下行：IDENTIFY

名称	命令 ID	方向	ZCL Payload
	0x00	0	标记时间
字节数	1	1	uint16

✧ 标记时间：设备处于标记状态的时间，单位（秒），设备在该时间内持续闪烁或发出响声，设置为 0 关闭标记状态。

5.2.3. IDENTIFY_QUERY 和 IDENTIFY_QUERY_RSP，查询标记

功能描述：查找进入标记状态的设备，可以广播查找。如果非要强行单播对一个非标记状态的设备发查询命令，只能收到默认返回命令。

下行：IDENTIFY_QUERY

名称	命令 ID	方向	ZCL Payload
	0x01	0	空
字节数	1	1	NULL

✧ 该命令以广播方式发送，目标端口也用 0xFF。

上行：IDENTIFY_QUERY_RSP

名称	命令 ID	方向	ZCL Payload
	0x00	1	超时时间
字节数	1	1	uint16

✧ 超时时间：设备处于标记状态的剩余时间。

5.3. 分组类命令，cluster=0x0004

功能描述：设备分组管理，包括设备加组，删组等操作。

5.3.1. 分组类的状态属性

分组类状态属性			
状态名称	Attr ID	数据类型	方向
支持分组名	0x0000	map8	0

✧ 支持分组名：设备是否支持分组取名，0 – 不支持取名，1 – 支持取名。

5.3.2. GROUP_ADD 与 GROUP_ADD_RSP，设备加组

功能描述：设备加组

下行：GROUP_ADD

名称	命令 ID	方向	ZCL Payload	
	0x00	0	组 ID	组名称
字节数	1	1	uint16	string

✧ 组 ID：设备加入的组 ID，需要精确到设备的端口。（电源光感端口不支持加组）

✧ 组名称：第一个字节是字符串长度，后面就是字符串，不需要的话第一个字节填 0x00。

上行：GROUP_ADD_RSP

名称	命令 ID	方向	ZCL Payload	
	0x00	1	ZCL 状态	组 ID
字节数	1	1	uint8	uint16

✧ ZCL 状态：设备加组的结果状态，见《ZCL 状态表》。

✧ 组 ID：设备加组的组 ID

例，向短地址 0xD799 端口 1 发送加组命令，加入 0x0001 组，组名字为 0

发送：A5 5A 15 0E 03 00 99 D7 01 04 01 04 00 00 62 00 00 00 00 01 00 00 B3 6A

接收：A5 5A 15 64 83 00 99 D7 01 04 01 04 00 00 62 01 D3 00 00 00 01 00 CC 60

发送一次，返回 0x00 加组成功

5.3.3. GROUP_VIEW 与 GROUP_VIEW_RSP，查询组信息

功能描述：查询设备分组内容，主要目的是查询分组名字

下行：GROUP_VIEW

名称	命令 ID	方向	ZCL Payload
	0x01	0	组 ID
字节数	1	1	uint16

✧ 组 ID：查询设备的组 ID

上行：GROUP_VIEW_RSP

名称	命令 ID	方向	ZCL Payload		
	0x01	1	ZCL 状态	组 ID	组名称
字节数	1	1	uint8	uint16	string

✧ ZCL 状态：设备加组的结果状态，见《ZCL 状态表》。

✧ 组 ID：设备加组的组 ID.

✧ 组名称：加组时设置的组名称字符串，第一字节表示长度

例，查询设备 0xD799 端口 1 的组 ID 0x0001，返回该设备在组里面，组名字长度 0。

发送：A5 5A 14 0E 03 00 99 D7 01 04 01 04 00 01 63 00 00 00 00 01 00 48 7F

收到：A5 5A 16 66 83 00 99 D7 01 04 01 04 00 01 63 01 D3 00 00 00 01 00 00 55 2F

5.3.4. GROUP_GET_MEMBERSHIP 与 GROUP_GET_MEMBERSHIP，查询设备加入的组

功能描述：查询设备加入的组

下行：GROUP_GET_MEMBERSHIP

名称	命令 ID	方向	ZCL Payload	
	0x02	0	加组数量 N	组 ID 列表
字节数	1	1	uint8	uint16 [N]

✧ 组 ID 数量：需要查询的组 ID 数量

✧ 组 ID 列表：由组 ID 数量决定，小端模式的 uint16 格式的组 ID 数组。

注意：如果组 ID 数量为 0，则能查到设备加入的全部分组

上行：GROUP_GET_MEMBERSHIP_RSP

名称	命令 ID	方向	ZCL Payload		
	0x02	1	加组容量	加组数量 N	组 ID 列表
字节数	1	1	uint8	uint8	uint16 [N]

✧ 加组容量：设备还能加组的数量。

✧ 加组数量：设备当前已加入组的数量。

✧ 组 ID 列表：设备已加入的组 ID，uint16 数组。

例：查询设备 0xD799 端口 1 加入的组

发送：A5 5A 13 0E 03 00 99 D7 01 04 01 04 00 02 63 00 00 00 00 00 58 08

收到：A5 5A 1A 76 83 00 99 D7 01 04 01 04 00 02 63 01 D4 00 00 11 03 01 00 02 00 03 00 82 41

查到设备已加入 0x03 个组，还能再加 0x11=17 个组，已加入 0x0001, 0x0002, 0x0003 组。

5.3.5. GROUP_REMOVE 与 GROUP_REMOVE_RSP，删除设备指定分组。

功能描述：删除设备上一个分组

下行：GROUP_REMOVE

名称	命令 ID	方向	ZCL Payload
	0x03	0	组 ID
字节数	1	1	uint16

✧ 组 ID：需要删除的组 ID

上行：GROUP_REMOVE_RSP

名称	命令 ID	方向	ZCL Payload	
	0x03	1	ZCL 状态	组 ID
字节数	1	1	uint8	uint16

✧ ZCL 状态：设备删组结果，见《ZCL 状态表》。

✧ 组 ID：被删除的组 ID。

注意：删除分组后，设备端在这个分组下创建的场景会被全部清除

例：设备 0xD799 端口 1，删除组 ID 0x0003，删除成功。

发送：A5 5A 14 0E 03 00 99 D7 01 04 01 04 00 03 42 00 00 00 00 03 00 F9 C4

收到：A5 5A 15 7A 83 00 99 D7 01 04 01 04 00 03 42 01 D0 00 00 00 03 00 75 F1

5.3.6. GROUP_REMOVE_ALL，删除全部分组

功能描述：删除设备中所有分组，如果分组中有场景，也会一并删除

下行：GROUP_REMOVE_ALL

名称	命令 ID	方向	ZCL Payload
	0x04	0	空
字节数	1	1	0

5.4. 场景类命令，cluster=0x0005

功能描述：设备的场景功能，不同的设备可以通过设置场景的方式。在统一的执行场景指令下触发各自应该有的动作。

5.4.1. 场景类的状态属性

场景类状态属性			
状态名称	Attr ID	数据类型	方向
已有场景数	0x0000	uint8	0
当前场景	0x0001	uint8	0
当前场景组	0x0002	uint16	0
场景执行状态	0x0003	bool	0
支持场景名	0x0004	map8	0

- ✧ 已有场景数：设备已设置的场景数
- ✧ 当前场景：当前处于的场景模式的场景 ID
- ✧ 当前场景组：当前处于场景模式的组 ID
- ✧ 场景执行状态：设备是否处于当前场景和场景组模式，如果场景执行后又做了其它操作为 0。
- ✧ 支持场景名：0 – 不支持，1 – 支持

5.4.2. SCENE_ADD 与 SCENE_ADD_RSP，添加场景

功能描述：强行为设备添加场景，需要匹配场景对应的执行数据

下行：SCENE_ADD

名称	命令 ID	方向	ZCL Payload				
	0x00	0	组 ID	场景 ID	渐变时间	场景名	场景数据
字节数	1	1	uint16	uint8	uint16	string	变长

- ✧ 组 ID：为场景分配的组 ID，设备必须先加到该组中才行。
- ✧ 场景 ID：场景执行的 ID，组 ID 和场景 ID 复用可以设置 65534*254 种场景
- ✧ 渐变时间：场景执行时的过度时间，避免设备状态突然变化
- ✧ 场景名：字符串的场景名称，第一个字节为名字长度。
- ✧ 场景数据：场景保存时的数据，由 ZCL 协议定义格式，比较复杂。

上行：SCENE_ADD_RSP

名称	命令 ID	方向	ZCL Payload		
	0x00	1	ZCL 状态	组 ID	场景 ID
字节数	1	1	uint8	uint16	uint8

✧ ZCL 状态：添加场景的返回状态，见《ZCL 状态表》。

✧ 组 ID：添加场景的组 ID

✧ 场景 ID：添加场景的场景 ID

5.4.3. SCENE_VIEW 与 SCENE_VIEW_RSP，查询场景内容

功能描述：查询设备上保存场景的内容，包括场景名，场景数据。

下行：SCENE_VIEW

名称	命令 ID	方向	ZCL Payload	
	0x01	0	组 ID	场景 ID
字节数	1	1	uint16	uint8

✧ 组 ID：查询场景的组 ID。

✧ 场景 ID：查询场景的场景 ID。

上行：SCENE_VIEW_RSP

名称	命令 ID	方向	ZCL Payload					
	0x01	1	ZCL 状态	组 ID	场景 ID	渐变时间	场景名	场景数据
字节数	1	1	uint8	uint16	uint8	uint16	string	变长

✧ ZCL 状态：查询场景的返回结果，见《ZCL 状态表》。

✧ 组 ID：查询场景的组 ID。

✧ 场景 ID：查询场景的场景 ID。

✧ 渐变时间：场景执行时的过度时间。

✧ 场景名：字符串的场景名称，第一个字节为名字长度。

✧ 场景数据：场景保存时的数据。

5.4.4. SCENE_REMOVE 与 SCENE_REMOVE_RSP，删除场景

功能描述：删除一个场景

下行：SCENE_REMOVE

名称	命令 ID	方向	ZCL Payload	
	0x02	0	组 ID	场景 ID
字节数	1	1	uint16	uint8

✧ 组 ID：删除场景的组 ID

✧ 场景 ID：删除场景的场景 ID

上行：SCENE_REMOVE_RSP

名称	命令 ID	方向	ZCL Payload		
	0x02	1	ZCL 状态	组 ID	场景 ID
字节数	1	1	uint8	uint16	uint8

✧ ZCL 状态：删除场景的状态返回，见《ZCL 状态表》

✧ 组 ID：删除场景的组 ID

✧ 场景 ID：删除场景的场景 ID

5.4.5. SCENE_REMOVE_ALL 与 SCENE_REMOVE_ALL_RSP，删除所有场景

功能描述：删除一个组下所有场景。

下行：SCENE_REMOVE_ALL

名称	命令 ID	方向	ZCL Payload
	0x03	0	组 ID
字节数	1	1	uint16

✧ 组 ID：被删除的场景所属组 ID，该组下场景全部清除。

上行：SCENE_REMOVE_ALL_RSP

名称	命令 ID	方向	ZCL Payload	
	0x03	1	ZCL 状态	组 ID
字节数	1	1	uint8	uint16

✧ ZCL 状态：删除场景的状态返回，见《ZCL 状态表》

✧ 组 ID：删除场景的组 ID

例

5.4.6. SCENE_STORE 与 SCENE_STORE_RSP，保存场景

功能描述：把设备的当前状态保存为场景，设备端自动打包场景数据到本地 ROM 中。

下行：SCENE_STORE

名称	命令 ID	方向	ZCL Payload	
	0x04	0	组 ID	场景 ID
字节数	1	1	uint16	uint8

✧ 组 ID：保存场景的组 ID

✧ 场景 ID：保存场景的场景 ID

上行：SCENE_STORE_RSP

名称	命令 ID	方向	ZCL Payload		
	0x04	1	ZCL 状态	组 ID	场景 ID
字节数	1	1	uint8	uint16	uint8

✧ ZCL 状态：保存场景返回状态，见《ZCL 状态表》

✧ 组 ID：保存场景的组 ID

✧ 场景 ID：保存场景的场景 ID

5.4.7. SCENE_RECALL，执行场景

功能描述：控制设备执行预设的场景，该命令用广播或组播方式发送

下行：SCENE_RECALL

名称	命令 ID	方向	ZCL Payload	
	0x05	0	组 ID	场景 ID

字节数	1	1	uint16	uint8

✧ 组 ID：执行场景的组 ID

✧ 场景 ID：执行场景的场景 ID

例

5.4.8. SCENE_GET_MEMBERSHIP 与 SCENE_GET_MEMBERSHIP_RSP，查询所有保存场景

功能描述：查询某个组下保存的全部场景 ID

下行：SCENE_GET_MEMBERSHIP

名称	命令 ID	方向	ZCL Payload
	0x06	0	组 ID
字节数	1	1	uint16

✧ 组 ID：需要查询全部场景的组 ID

上行： SCENE_GET_MEMBERSHIP_RSP

名称	命令 ID	方向	ZCL Payload				
	0x06	1	ZCL 状态	场景容量	组 ID	场景数量 N	场景 ID 列表
字节数	1	1	uint8	uint8	uint16	uint8	uint8 [N]

✧ ZCL 状态：查询返回的状态，见《ZCL 状态表》。

✧ 场景容量：设备还能设置场景的数量。

✧ 组 ID：被查询的组 ID。

✧ 场景数量：被查询的组中场景总数。

✧ 场景 ID 列表：组中所有场景的列表。

例

5.5. 开关类命令，cluster = 0x0006

功能描述：该类命令用于设备的开关状态控制，任何可以用 0 和 1 状态表示的控制对象，都能用该命令控制。

5.5.1. 开关类状态属性

开关类状态属性			
状态名称	Attr ID	数据类型	方向
开关状态	0x0000	bool	0

✧ 开关状态：0 - 关灯状态，1 - 开灯状态。（可上报）

5.5.2. OFF，ON 和 TOGGLE，开关控制

功能描述：控制设备的关，开，和置反。发送该命令后设备的属性“开关状态”值会发生变化。

下行：OFF

名称	命令 ID	方向	ZCL Payload

	0x00	0	空
字节数	1	1	NULL

例，广播关灯，短地址和端口都是 0xFF

发送：A5 5A 12 0F 03 00 FF FF FF 04 01 06 00 00 30 00 00 00 00

下行：ON

名称	命令 ID	方向	ZCL Payload
	0x01	0	空
字节数	1	1	NULL

例，广播开灯，短地址和端口都是 0xFF

发送：A5 5A 12 10 03 00 FF FF FF 04 01 06 00 01 31 00 00 00 00 70 BB

下行：TOGGLE

名称	命令 ID	方向	ZCL Payload
	0x02	0	空
字节数	1	1	NULL

例，广播置反，短地址和端口都是 0xFF

发送：A5 5A 12 11 03 00 FF FF FF 04 01 06 00 02 32 00 00 00 00 F5 18

5.6. 亮度类命令，cluster = 0x0008

功能描述：调节设备的功率输出，包括对照明设备的亮度调节。

5.6.1. 亮度类状态属性

亮度类状态属性			
状态名称	Attr ID	数据类型	方向
当前亮度	0x0000	uint8	0
剩余渐变时间	0x0001	uint16	0

- ✧ 当前亮度：设备的当前输出亮度或功率比例，范围 1~254。（可上报）
- ✧ 剩余渐变时间：设备在渐变调光时的剩余渐变时间，会越来越少变成 0。

5.6.2. MOVE_TO_LEVEL 与 MOVE_TO_LEVEL_ONOFF，调至亮度

功能描述：设备调至一个目标亮度，有不带开关效果和带开关效果两种。

下行：MOVEL_TO_LEVEL

名称	命令 ID	方向	ZCL Payload	
	0x00	0	亮度	渐变时间
字节数	1	1	uint8	uint16

- ✧ 亮度：调至目标亮度，范围 1~254，输入 0 也只能到 1。
- ✧ 渐变时间：设备亮度变化过程中的渐变时间，单位 100 毫秒。

下行：MOVEL_TO_LEVEL_ONOFF

名称	命令 ID	方向	ZCL Payload	
	0x04	0	亮度	渐变时间
字节数	1	1	uint8	uint16

- ✧ 亮度：调至目标亮度，范围 0~255，输入 0 关灯。
- ✧ 渐变时间：设备亮度变化过程中的渐变时间，单位 100 毫秒。

例，广播调亮度，亮度 50%（对应值 0x7F），渐变时间 1 秒

发送：A5 5A 15 14 03 00 FF FF FF 04 01 08 00 04 4A 00 00 00 00 7F 0A 00 F8 4C

5.6.3. MOVE 与 MOVE_ONOFF，自动变化亮度

功能描述：控制设备按照固定的亮度比例，调至最大或最小亮度。

下行：MOVE

名称	命令 ID	方向	ZCL Payload	
	0x01		升降模式	亮度比率
字节数	1	1	uint8	uint8

- ✧ 升降模式：0 - 升亮度，1 - 降亮度，降亮度模式调至最低亮度
- ✧ 亮度比率：单次调节亮度值（以 100 毫秒为单位）

下行：MOVE_ONOFF

名称	命令 ID	方向	ZCL Payload	
	0x05		升降模式	亮度比率
字节数	1	1	uint8	uint8

- ✧ 升降模式：0 - 升亮度，1 - 降亮度，降亮度模式调至关灯
- ✧ 亮度比率：单次调节亮度值（以 100 毫秒为单位）

5.6.4. STEP 与 STEP_ONOFF，单步亮度

功能描述：单步调节设备亮度

下行：STEP

名称	命令 ID	方向	ZCL Payload		
	0x02	0	模式	步进	渐变时间
字节数	1	1	uint8	uint8	uint16

- ✧ 模式：0 - 升亮度，1 - 降亮度，最多降至亮度 1。
- ✧ 步进：亮度变化量。
- ✧ 渐变时间：亮度变化的过度时间。

下行：STEP_ONOFF

名称	命令 ID	方向	ZCL Payload		
	0x06	0	模式	步进	渐变时间
字节数	1	1	uint8	uint8	uint16

- ✧ 模式：0 - 升亮度，1 - 降亮度，直到关灯。
- ✧ 步进：亮度变化量。

✧ 渐变时间：亮度变化的过度时间。

5.6.5. STOP 与 STOP_ONOFF，停止渐变

功能描述：正在亮度渐变的设备停止渐变

下行：STOP

名称	命令 ID	方向	ZCL Payload
	0x03	0	无
字节数	1	1	NULL

下行：STOP_ONOFF

名称	命令 ID	方向	ZCL Payload
	0x07	0	无
字节数	1	1	NULL

5.7. 安防类命令，cluster = 0x0500

功能描述：安防设备的消息，包括人体红外，门磁警报，烟雾类警报

5.7.1. 安防类状态属性

安防类状态属性			
状态名称	Attr ID	数据类型	方向
安防状态	0x0000	enum8	0
安防类型	0x0001	enum16	0
警报状态	0x0002	map16	0
接收器地址	0x0010	EUI64	0
安防 ID	0x0011	uint8	0

✧ 安防状态：0 - 未注册接收设备，1 - 已注册接收设备。
✧ 安防类型：安防传感器类型，不同的传感器警报的意义不同。
✧ 警报状态：安防警报状态，以 16 个标志位表示不同的警报。
✧ 接收器地区：注册的警报接收器的 IEEE 地址。（可写）
✧ 安防 ID：设备出厂的设置值

安防类型			
值	安防类型名称	警报标志 1	警报标志 2
0x0000	基础设备	系统警报	无
0x000d	运动传感器	入侵指示	存在指示
0x0015	接触开关	1 路门户开关	2 路门户开关
0x0016	门窗检测	门窗状态 1	门窗状态 2
0x0028	火传感器	有火	无
0x002a	水传感器	水浸	无
0x002b	一氧化碳传感器	有一氧化碳	厨房报警

0x002c	个人急救传感器	人跌倒	紧急按钮
0x002d	振动运动传感器	检测到运动	振动
0x010f			
0x0115			
0x012d			
0x0225			
0x0226			
0x0229			

- ✧ 门窗状态 1 和门窗状态 2 都为 0 为关闭状态，都为 1 为打开状态，一个 1 一个 0 表示半开半闭状态。

警报状态标志位	
bit	解释
0	警报标志 1
1	警报标志 2
2	故意破坏
3	电池警报
4	监管通知
5	恢复通知
6	故障
7	AC 供电报警
8	进入测试模式
9	电池故障

安防类命令

0x00	1	STATUS_CHANGE_NOTIFICATION	状态改变通知	收
0x01	1	STATUS_ENROLL_REQUEST	安防注册请求	收
0x00	0	STATUS_ENROLL_RESPONSE	安防注册响应	发

5.7.2. STATUS_CHANGE_NOTIFICATION，报警状态通知

上行：STATUS_CHANGE_NOTIFICATION

名称	命令 ID	方向	ZCL Payload			
	0x00	1	警报状态	扩展状态	安防 ID	延迟时间
字节数	1	1	uint16	uint8	uint8	uint16

- ✧ 警报状态：与状态属性中的“警报状态”相同，是状态属性触发变化时的消息推送。
- ✧ 扩展状态：暂时没用
- ✧ 安防 ID：用于表示安防设备的安装位置
- ✧ 延迟时间：传感器检测警报到传送消息的延迟时间，主要是针对网络堵塞时也要发出消息，单位 1/4 秒。

例，协调器收到麦乐克人体红外/光感一体式传感器的人体检测信息

收到：A5 5A 18 75 83 00 AC 88 01 04 01 00 05 00 67 01 D3 00 00 31 00 00 00 00 00

传感器短地址和端口是 0x88AC 和 0x01

传感器报警信息 cluster = 0x0500，命令 ID=0x00，方向为 1。

报警状态是 0x0031，bit 位使能 bit0，bit4，bit5

延迟时间是 0x0000

5.7.3. STATUS_ENROLL_REQUEST，安防注册请求

上行：STATUS_ENROLL_REQUEST

名称	命令 ID	方向	ZCL Payload	
	0x01	1	安防类型	厂商码
字节数	1	1	uint16	uint16

- 安防类型：同“安防类状态属性”中“安防类型”（Attribute ID = 0x0001）。
- 厂商码：设备的厂商码，和命令帧头的厂商码相同。

5.7.4. STATUS_ENROLL_RESPONSE，安防注册响应

下行：STATUS_ENROLL_RESPONSE

名称	命令 ID	方向	ZCL Payload	
	0x00	0	回复状态	安防 ID
字节数	1	1	uint8	uint8

- 回复状态：见附表
- 安防 ID：主机给安防设备分配一个位置 ID。

状态码	意义	详细描述
0	Success	成功
1	Not supported	主机不支持该类型的安防设备
2	No enroll permit	主机在该时间窗口内不再接受新设备注册
3	Too many zones	主机支持的安防设备数量超过上限

注意事项：网关收到安防设备发上来的 STATUS_ENROLL_REQUEST 后，把该命令报告给云平台或 APP，云平台或 APP 来生成安防 ID 回复给安防设备。回复的时延不限，但是建议越快越好，不回复也没影响。

5.8. 光感类命令，cluster = 0x0400

功能描述：光照度传感器的数据，该类命令无 ZCL 专有命令，完全通过对状态属性访问来控制。

5.8.1. 光感类状态属性

光感类状态属性			
状态名称	Attr ID	数据类型	方向
当前光照度	0x0000	uint8	0
最小光照度	0x0001	uint8	0

最大光照度	0x0002	uint8	0

✧ 当前光照度：光感感应到的光感值，不是直接 lux 值，而是转换成对数的值。（可上报）

✧ 最小光照度：光感能检测到的最小照度值，也是转换成对数的。

✧ 最大光照度：光感能检测到的最大照度值，还是转换成对数的。

光感对数值与 lux 转换公式：

照度值 = 10000 * 1og Lux + 1。

Lux = 10 ^ ((照度值-1)/10000)

例：收到麦乐克人体红外光感一体式传感器的光感上报数据（通用命令 ZCL_REPORT）

收到：A5 5A 18 2D 82 00 AC 88 01 04 01 00 04 0A 18 01 72 00 00 01 00 00 21 A6 33 7E 06

短地址是 0x88AC，端口是 0x01。

光感类的 cluster = 0x0400，ZCL_REPORT 命令 ID=0x0A，光感主动发给协调器方向是 1。

上报了 1 个状态属性 0x0000，数据类型 0x21 是 uint16，值为 0x33A6 = 13222。

按照公式 lux = 10 ^ 1.3221 = 20.994 lux

5.9. 涂鸦自定义命令，cluster = 0xEF00

功能描述：涂鸦串口模块自定义的通信协议，包括涂鸦窗帘电机，涂鸦传感器

5.9.1. 涂鸦自定义状态属性

涂鸦自定义协议状态属性为空。

5.9.2. TUYA_RX，协调器发给涂鸦模块的命令

下行：TUYA_RX

名称	命令 ID	方向	ZCL Payload
	0x00	0	涂鸦数据包
字节数	1	1	uint8 [∞]

✧ 涂鸦数据包：涂鸦自定义的数据包格式

5.9.3. TUYA_TX，涂鸦模块发给协调器的命令

上行：TUYA_TX

名称	命令 ID	方向	ZCL Payload
	0x01	1	涂鸦数据包
字节数	1	1	uint8 [∞]

✧ 涂鸦数据包：涂鸦自定义的数据包格式

5.9.4. 涂鸦数据包解析，窗帘电机部分

涂鸦窗帘控制协议，解析如下，所有内容可能是大端模式。

名称	涂鸦数据包				
	保留	涂鸦命令	保留	参数长度	参数

字节数	2	2	1	1	变长

- ✧ 涂鸦命令：2 字节大端模式
- ✧ 参数长度：命令参数长度
- ✧ 参数：大端模式的参数

下发命令表

功能	涂鸦命令	参数长度	参数
窗帘开关控制	0x01, 0x04	0x01	0x00~0x02
窗帘位置控制	0x02, 0x02	0x04	0x00, 0x00, 0x00, (0x00~0x64)

- ✧ 窗帘开关控制：0x00 开窗帘，0x01 停止，0x02 关窗帘
- ✧ 窗帘位置控制：0x00~0x64，表示窗帘覆盖率 0%~100%

上传命令表，窗帘电机的状态反馈

功能	涂鸦命令	参数长度	参数
窗帘运动方向	0x01, 0x04	0x01	0x00~0x02
窗帘计划位置	0x02, 0x02	0x04	0x00, 0x00, 0x00, (0x00~0x64)
窗帘最终位置	0x03, 0x02	0x04	0x00, 0x00, 0x00, (0x00~0x64)

- ✧ 窗帘运动方向：包括发命令和手拉窗帘，都能产生该命令，0x00 开窗帘，0x01 窗帘停止（受外力或停止命令），0x02 关窗帘。
- ✧ 窗帘计划位置：发命令或手拉窗帘后，窗帘将要运动到的位置。
- ✧ 窗帘最终位置：窗帘停止运动后，最终停留的位置。

例，控制窗帘运动到 50%

发送：A5 5A 1C 08 03 00 12 61 01 04 01 00 EF 00 41 00 00 00 00 00 00 02 02 00 04 00 00 00 32 46 D5，红色为涂鸦数据包。

收到 3 包数据：

第一包：A5 5A 19 78 83 00 12 61 01 04 01 00 EF 01 43 01 D0 00 00 00 00 01 04 00 01 01 6A 4E

电机停下来了

第二包：A5 5A 1C 79 83 00 12 61 01 04 01 00 EF 01 44 01 D1 00 00 00 00 02 02 00 04 00 00 00 32 C2 FC

电机的目标位置是 0x32 的位置

第三包：A5 5A 1C 7A 83 00 12 61 01 04 01 00 EF 01 45 01 D1 00 00 00 00 03 02 00 04 00 00 00 32 C9 70

电机停在了 0x32 的位置

例 2，手拉窗帘运动

启动收到 2 包数据

A5 5A 19 01 83 00 12 61 01 04 01 00 EF 01 46 01 D2 00 00 00 00 01 04 00 01 00 CF 54

A5 5A 1C 02 83 00 12 61 01 04 01 00 EF 01 47 01 D1 00 00 00 00 02 02 00 04 00 00 00 00 2B 69

窗帘往开窗方向运动，目标是 0%的覆盖率。

停止时收到 2 包数据

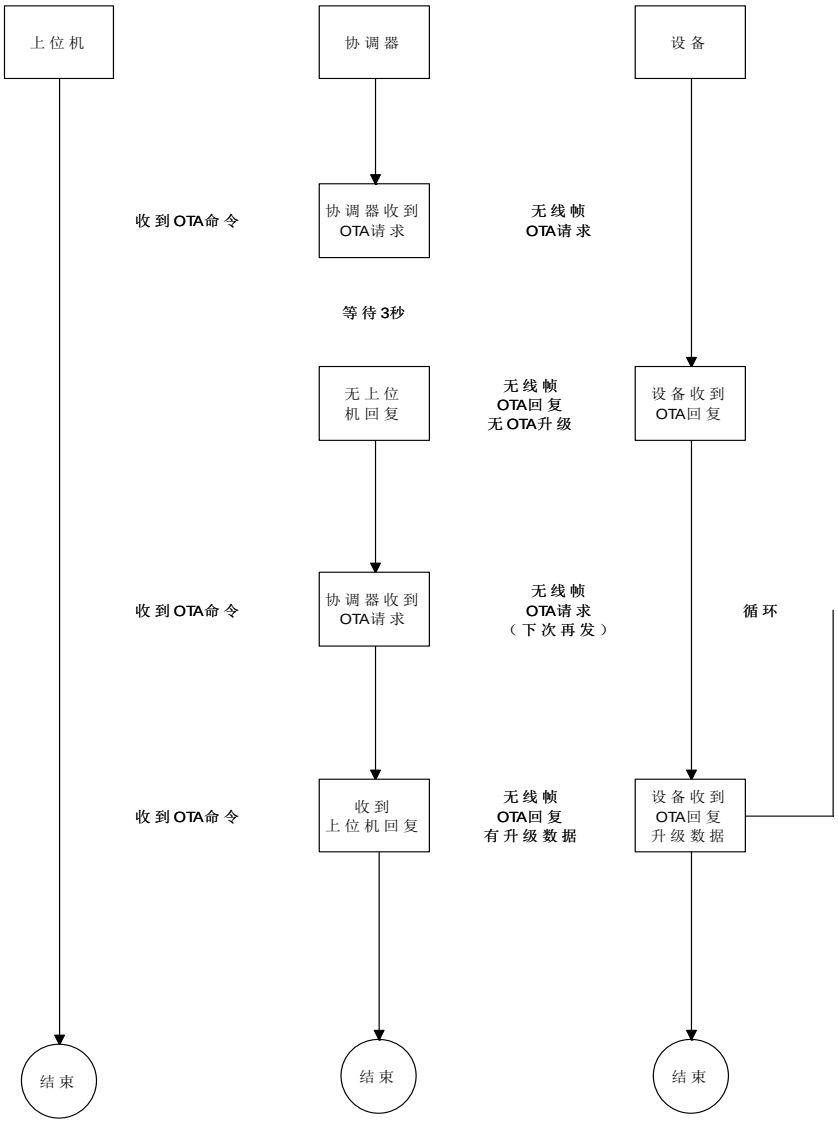
A5 5A 1C 02 83 00 12 61 01 04 01 00 EF 01 47 01 D1 00 00 00 00 02 02 00 04 00 00 00 00 2B 69

A5 5A 1C 04 83 00 12 61 01 04 01 00 EF 01 49 01 D1 00 00 00 00 03 02 00 04 00 00 00 00 72 21
窗帘停止在 0%的覆盖率位置

6. OTA 类命令

OTA 命令是一种特殊的 ZCL 专有命令，因此独立定义了串口格式。上位机通过向协调器发送 ZCL 专有命令时上位机是控制者，设备端是受控者，上位机给设备发请求，设备向上位机返回响应。但是在 OTA 升级时这个逻辑又被颠倒过来了，设备端向上位机发请求，上位机向设备端返回消息。而且由于在 OTA 模式下，协调器向设备端回复 OTA 消息不会输出状态反馈（应用层已过滤）。

协调器收到 OTA 请求后，会向上位机传送“收到 OTA 命令”，如果在 3 秒钟内上位机不向协调器发送“下发 OTA 命令”，协调器自动回复无 OTA 数据的消息，这样会导致设备端暂停向协调器发送 OTA 请求。



✧

6.1. OTA 命令帧格式

6.1.1. 下发 OTA 命令，命令大类 0x04

下行

名称	串口帧 Payload				
	命令 ID	短地址	端口	保留	OTA 参数
字节数	1	2	1	1	变长

命令 ID：1 字节，见《OTA 命令表》。

短地址：目标设备短地址。

端口：目标设备的 OTA 端口。

保留：该位填 0

OTA 参数：下发命令时填充的 OTA 参数

AKC：

名称	状态	扩展参数
		无
字节数	1	0

✧ 状态：见《ACK 状态表》

6.1.2. 收到 OTA 命令，命令大类 0x84

上行

名称	串口帧 Payload				
	命令 ID	短地址	端口	保留	OTA 参数
字节数	1	2	1	1	变长

命令 ID：1 字节，见《OTA 命令表》。

短地址：源设备短地址。

端口：源设备的 OTA 端口。

保留：该位填 0。

OTA 参数：设备 OTA 请求的 OTA 参数。

6.1.3. OTA 命令表

OTA 命令表			
命令名称	命令 ID	命令大类	命令方向
版本通知	0x00	0x00	下行
请求开始升级	0x01	0x84	上行
回复开始升级	0x01	0x04	下行
请求文件传输	0x02	0x84	上行
回复文件传输	0x02	0x04	下行
升级完成通知	0x03	0x84	上行

名称	OTA 参数			
	厂商代码	设备型号	固件版本	
字节数	uint16	uint16	uint32	

6.1.4. OTA 相关状态属性，cluster = 0x0019

注意：OTA 的状态属性，由于协调器是受控方，设备端是主控方，因此属性的访问方向为 1，即（Server->Client）。

可以使用 ZCL 通用命令访问设备的 OTA 状态属性，查看 OTA 进程。

OTA 类状态属性，cluster = 0x0019			
状态名称	Attr ID	数据类型	方向
OTA 服务器	0x0000	EUI64	1
固件 offset	0x0001	uint32	1
当前固件版本	0x0002	uint32	1
当前协议栈版本	0x0003	uint16	1
下载的固件版本	0x0004	uint32	1
下载的协议栈版本	0x0005	uint16	1
升级状态	0x0006	enum8	1
厂商代码	0x0007	uint16	1
设备型号	0x0008	uint16	1

- ✧ OTA 服务器：OTA 固件下载的服务器 IEEE 地址，通常是协调器 IEEE 地址。
- ✧ 固件 offset：设备当前以更新固件的 offset。
- ✧ 当前固件版本：设备当前运行的固件版本。
- ✧ 当前协议栈版本：设备当前运行固件的协议栈版本。
- ✧ 下载的固件版本：设备正在下载的固件版本。
- ✧ 下载的协议栈版本：设备正在下载的固件的协议栈版本。
- ✧ 升级状态：设备升级状态，见《OTA 状态表》。
- ✧ 厂商代码：16 位厂商编码。
- ✧ 设备型号：16 位设备型号。

OTA 状态表	
状态值	描述
0x00	一般模式
0x01	下载中
0x02	下载完成
0x03	等待升级响应
0x04	升级倒计时
0x05	等待更多升级文件

6.2. OTA 命令解析

6.2.1. 版本通知命令，命令 ID=0x00

功能描述：如果网关有更新固件版本，可以通知设备升级。否则设备需要到了升级检测的时间才开始升级。该命令可以广播，适当设置升级概率参数，可以随机使部分设备升级。

下行：通知升级

名称	OTA 参数				
	比较参数	升级概率	厂商代码	设备型号	固件版本

字节数	uint8	uint8	uint16	uint16	uint32

比较参数：0-不比较任何，1-比较厂商码，2-比较设备型号，3-比较固件版本

升级概率：0 - 100%升级，255 - 0%升级，值越大，收到指令后设备开始升级概率越低。

厂商代码：16 位厂商代码，uint16 小端模式。

设备型号：16 位设备型号，uint16 小端模式。

固件版本：设备当前的 32 位固件版本，uint32 小端模式。

6.2.2. 开始升级命令（请求开始升级与回复开始升级），命令 ID=0x01

功能描述：设备向协调器请求开始升级，如果上位机有该设备的升级文件就回复开始升级命令。

上行：请求开始升级

名称	OTA 参数					
	厂商代码	设备型号	固件版本	附标志	硬件版本	设备 IEEE 地址
字节数	uint16	uint16	uint32	uint8	uint16/0	uint8 [8]/0

厂商代码：16 位厂商代码，uint16 小端模式。

设备型号：16 位设备型号，uint16 小端模式。

固件版本：设备当前的 32 位固件版本，uint32 小端模式。

附标志： bit0 – 该帧带有硬件版本，bit1 – 该帧带有设备 IEEE 地址。

硬件版本：请求升级的设备的硬件版本。

设备 IEEE 地址：请求升级的设备的 IEEE 地址。

下行：回复开始升级

名称	OTA 参数					
	厂商代码	设备型号	固件版本	回复状态	附标志	升级文件大小
字节数	uint16	uint16	uint32	uint8	uint8	uint32

厂商代码：16 位厂商代码，uint16 小端模式。

设备型号：16 位设备型号，uint16 小端模式。

固件版本：最新的固件版本，uint32 小端模式。

回复状态：上位机有升级文件时，要回 0x00。

附标志： 0x00-请求开始升级不带 IEEE 地址，0x01-请求开始升级带 IEEE 地址。

升级文件大小：新版本固件的文件大小，uint32 小端模式。

注意：请求升级的设备，厂商代码和设备型号与上位机保存的一样，上位机上的固件比设备当前固件版本高，上位机就可以回复开始升级。

6.2.3. 文件传输命令（请求文件传输和回复文件传输），命令 ID=0x02

功能描述：设备从上位机下载升级文件。

上行

名称	OTA 参数				
	厂商代码	设备型号	固件版本	固件 offset	数据大小
字节数	uint16	uint16	uint32	uint32	uint8

厂商代码：16 位厂商代码，uint16 小端模式。

设备型号：16 位设备型号，uint16 小端模式。

固件版本：设备请求下载的固件版本，uint32 小端模式。

固件 offset：设备请求下载的固件的 offset，uint32 小端模式。

数据大小：设备请求该次下载的数据大小。

下行

名称	OTA 参数						
	厂商代码	设备型号	固件版本	回复状态	固件 offset	数据大小	固件数据
字节数	uint16	uint16	uint32	uint8	uint32	uint8	uint8 [变长]

厂商代码：16 位厂商代码，uint16 小端模式。

设备型号：16 位设备型号，uint16 小端模式。

固件版本：升级固件的版本，uint32 小端模式。

回复状态：回复 0x00，否则设备会终止升级。

固件 offset：回复升级固件的 offset。

数据大小：该帧升级包的数据大小。

固件数据：升级固件的数据。

6.2.4. 升级完成通知，命令 ID=0x03

功能描述：设备端完成固件传输后通知网关的消息，有可能升级成功，也有可能升级不成功。

名称	OTA 参数					
	完成标志	ZCL 状态	保留位	厂商代码	设备型号	固件版本
字节数	uint8	uint8	uint8	uint16	uint16	uint32

完成标志：收到完成标志是 0x00。

ZCL 状态：见《ZCL 状态表》，0x00 表示升级成功。

保留位：默认为 0x00

厂商代码：16 位厂商代码，uint16 小端模式。

设备型号：16 位设备型号，uint16 小端模式。

固件版本：升级固件的版本，uint32 小端模式。