

窗帘Matter模块AT指令指南V1

版本	日期	修改说明
Version 1.0.0	2022/12/2	将波特率由115200改为9600
Version 1.1.1	2022/12/21	增加可充电电源信息配置的AT指令集；删减部分属性的：减少报文的交互；在介绍中增加属性值同步的重要性；名称改为中文方便于阅读；增加属性默认值说明；明确
Version 1.2.0	2023/2/24	所有电源信息Matter指令集中的object_id参数从2改为1
Version 1.3.0	2023/3/20	增加解绑清网的AT指令； 增加预览版剩余次数的AT指令查询和次数归零的AT主动 GMR版本号更改
Version 1.4.0	2023/3/29	增加SED poll周期的运行时更改AT指令
Version 1.5.0	2023/4/6	增加射频功率的运行时更改AT指令
Version 1.5.1	2023/4/19	优化AT解析和MT_FACTORYRESET可能引起的死锁现象
Version 1.5.2	2023/5/4	修复文档AT格式分隔符描述错误;修复MT_ATTNOTIFY应 误 增加对苹果平台的解绑清网的操作描述
Version 1.6.0	2023/5/5	增加Tilt窗帘的相关AT指令 大幅度调整优化AT文档内容

Version 1.7.0	2023/5/17	增加MT_CURTAIN_CTPP_UPDATE指令，用于同步更新
Version 1.8.0	2023/5/22	增加MT_CURTAIN_CPP_AUTOMODE指令，CPP值可根步，减少client端上报，简化client端开发流程
Version 1.9.0	2023/6/2	增加对HRN71,HAT31模组AT及日志引脚说明
Version 1.9.1	2023/8/1	增加低功耗模式的说明，修改模组AT的引脚图
Version 1.9.2	2023/10/7	在MT_EVENT指令这部分补充UNBIND_CAUSED_FACTOR
Version 1.9.3	2023/10/27	增加仓库模式建议
Version 1.9.4	2023/10/28	修改部分错误； 增加APP拖动条跳动影响体验的说明，以及给出建议方法
Version 1.10.0	2024/5/15	增加蓝牙部分AT指令集
Version 1.10.1	2024/6/4	增加蓝牙AT交互流程图 增加UUID默认值的说明及替换规则
Version 1.10.2	2024/6/12	增加获取蓝牙MAC地址的AT指令
Version 1.10.3	2024/6/18	增加蓝牙错误码指令集； 增加TX Notify属性的查询及主动通知。
Version 1.11.0	2024/6/24	增加SERVER通知指令的说明； 增加AT_RX_QUEUE_BUSY和AT_TX_QUEUE_BUSY状态

一、介绍

1. 模块UART通信标准:

- a. 波特率 9600
- b. 数据位 8位
- c. 停止位 1位
- d. 校验位 0位
- e. 流控 无

2. 模块支持以下2种低功耗工作模式。（支持模块：HRN71）

a. GPIO控制模式:

该模式下，模块空闲状态将进入低功耗，并且UART-RX会关闭。只有通过RX_EN引脚拉高，UART-RX才开启，并接收数据。而且当模块UART-TX发送数据时，TX_NTF引脚将拉高一段时间。

b. 前导码唤醒模式:

该模式下，模块空闲状态将进入低功耗，并且UART-RX会关闭。只有通过向模块UART-RX发送前导码，才能将模块UART唤醒并开启RX接收数据。而且模块UART-TX发送数据时，也会在数据帧头增加前导码。数据格式示例：前导码+AT指令，如0xC0 0xC0 0xC0 0xC0 0xC0 0xC0 0x4F 0x4B 0x0D 0x0A。

3. 本模块的AT指令集分为3大类：模块基础指令集、Matter基础指令集、Matter产品指令集。每条指令根据操作方式分为3种：查询命令（命令的查询或属性的读出）、执行命令（命令的执行或属性的写入）、通知消息。

4. 角色的定位：模块作为AT server，MCU作为AT client。client(MCU)负责发送AT指令和接收AT响应，server(模块)负责接收AT指令，开始执行对应的功能并发送AT响应。当某些状态或数值发生变化时以及产生事件时，server(模块)发送通知消息，client(MCU)接收和处理。

5. Matter基础指令集是Matter模块的一些通用命令，与业务无关。Matter产品指令集是以Matter现有支持的产品的标准物模型为核心，根据该产品类型使用的场景所制定的属性和命令的AT指令。

6. 模块的核心是本地的matter物模型，网关或平台会读取模块物模型中的属性值。因此，在使用模块时，尽可能的保证MCU中的数值与模块中的属性值同步，每当状态或属性发生变化时，都要通过AT指令的形式，将最新的数值写入到对应属性中。

7. 仓储模式的建议：可通过AT+RST命令将模块重启，然后不发送AT+MT_START命令，这样设备就不会开启matter协议栈线程，因此射频也不会开启，此时功耗最低，可用于仓储模式。

二、AT指令格式

AT指令以“AT”开始，代表Attention，以CR LF(\r\n)为结尾。输入的每条命令都会返回执行结果，成功OK，出错ERROR。

AT指令格式及规则：

指令类型	格式	说明
查询指令	AT+ <命令名称>[=<...>]?	client端发送。查询该指令下的当前参数值。
执行指令	AT+ <命令名称>[=<...>]	client端发送。根据参数要求填入参数，并执行。如果是Matter物模型的属性操作则写入对应的属性值。
指令响应	+ <命令名称>:<结果>[,<...>]	server端发送。执行指令后，server端对查询指令或执行指令发送应答，返回执行结果。
主动通知	> <命令名称>:<...>	server端发送。当状态发送变化或某属性发生变化时，server端主动发送通知消息。

- 不是每条 AT 指令都具备上述四种类型的命令。
- 尖括号 < > 内的参数不可以省略。
- 方括号 [] 内的参数可以省略。
- 多个参数用 , 分隔。
- 所有参数凡是数值的，都是以十进制格式输入和输出

三、模块基础指令集

1. SERVER AT运行状态通知

AT服务运行状态通知。

SERVER AT运行状态通知	
client(MCU)	server(module)
	通知消息 >SERVER:<server status>
参数 无	参数 <server status> string类型，AT server运行状态
示例 >SERVER:START >SERVER:AT_RX_QUEUE_BUSY >SERVER:AT_TX_QUEUE_BUSY	

<server status>，AT server运行状态：

START, AT服务开启, AT服务开启时主动发送的命令

AT_RX_QUEUE_BUSY, AT接收队列占满, 无法接受新AT

AT_TX_QUEUE_BUSY, AT发送队列占满, 无法发送新AT通知和应答

2. AT AT测试命令

测试AT指令是否有效, 能否正常响应。

AT AT测试命令	
client	server
执行命令 AT	响应 OK
参数 无	参数 无
示例 AT OK	

3. AT+RST 重启模块

模块复位重启

AT+RST 重启模块	
client	server
执行命令 AT+RST	响应 +RST:OK
参数 无	参数 无
示例 AT+RST +RST:OK	

4. AT+GMR 查看版本信息

查看模块版本信息，包含AT版本、软件版本、硬件版本、thread版本、Matter版本。

AT+GMR 查看版本信息	
client	server
查询命令 AT+GMR?	响应 +GMR:OK,<AT version>,<software version>,<hardware version>,<thread version>,<matter version>
参数 无	参数 <AT version> string类型，AT指令版本 <software version> string类型，模块软件版本号 <hardware version> string类型，模块硬件版本号 <thread version> string类型，thread版本号 <matter version> string类型，Matter版本号
示例 AT+GMR?	+GMR:OK,1.0,1.0.0-3f74bbc2e1baaf19,1.0.0,1.3,1.0

5. AT+RSSI 查看RSSI

查看模块与父节点的之间的RSSI值。

AT+RSSI 查看RSSI	
client	server
查询命令 AT+RSSI?	响应 +RSSI:OK,<rsssi> +RSSI:ERROR,<err_code>
参数 无	参数 <rsssi> int8类型，rsssi值 <err_code> int类型，错误代码
示例 AT+RSSI?	+RSSI:OK,-30

6. AT+FW_COUNT 查询预览版固件剩余次数

查询预览版固件里的窗帘移动可操作剩余次数

AT+FW_COUNT 查看预览版固件剩余次数	
client	server
查询命令 AT+FW_COUNT?	响应 + FW_COUNT:OK,<count> + FW_COUNT:ERROR,<err_code>
参数 无	参数 <count> int类型, 剩余次数值 <err_code> int类型, 错误代码
示例 AT+ FW_COUNT? + FW_COUNT:OK,400	

注：在预览版固件窗帘可操作移动次数归零后，模组会主动通知MCU

格式为：>SERVER:UNAUTHORIZED

四、Matter基础指令集

1. MT_EVENT 事件通知

事件通知

MT_EVENT 事件通知	
client	server
	通知消息 >MT_EVENT:<event type>
参数 无	参数 <event type> string类型, 事件类型
示例 >MT_EVENT:CONNECTED_BR	

<event type>, 事件类型:

REQUIRE_CREDENTIAL	未配网状态, thread未获得dataset
CREDENTIAL_READY	已配网状态, thread已获得dataset
COMMISSION_OPENING	配网窗口打开
COMMISSION_CLOSED	配网窗口关闭
COMMISSION_FAILED	配网失败
CONNECTING_BR	正在连接网关
CONNECTED_BR	已连接到网关
UNBIND_CAUSED_FACTORY_RESET	设备解绑

2. MT_IDENTIFY 识别命令通知

平台上的识别命令通知 (如苹果配网后点击“识别”)

Server端主动通知Client识别命令的到来, Client端进行相应动作

MT_IDENTIFY 识别通知	
client	server
	通知消息 >MT_IDENTIFY:<identify type>
参数 无	参数 <identify type> string类型, 识别类型
示例	>MT_IDENTIFY:START

<identifytype>, 识别类型:

START, 开始识别

STOP, 停止识别

3. AT+MT_START 启用Matter

启用matter。Matter开始启用后，若设备还未配网过，则进入配网状态，此时将产生COMMISSION_OPENING事件；若设备已经配网过，则设备将连接网关，连接成功后将产生CONNECTED_BR事件。

AT+MT_START 启用matter	
client	server
执行命令 AT+MT_START	响应 +MT_START:OK +MT_START:ERROR,<err_code>
参数 无	参数 <err_code> int类型，错误代码
示例 AT+MT_START +MT_START:OK	

4. AT+MT_RECMMSS 重新配网

重新开启matter配网

AT+RECMMSS 重新配网	
client	server
执行命令 AT+MT_RECMMSS	响应 +MT_RECMMSS:OK +MT_RECMMSS:ERROR,<err_code>
参数 无	参数 <err_code> int类型，错误代码
示例 AT+MT_RECMMSS +MT_RECMMSS:OK	

5. AT+MT_FACTORYRESET matter恢复出厂设置

matter恢复出厂设置，清除所有matter的数据并重启。**注意：由于清除了所有matter相关的数据，因此重启完成后需要重新配置和同步相关的属性。**

AT+MT_FACTORYRESET matter恢复出厂设置	
client	server
执行命令 AT+MT_FACTORYRESET	响应 +MT_FACTORYRESET:OK +MT_FACTORYRESET:ERROR,<err_code>
参数 无	参数 <err_code> int类型，错误代码
示例 AT+MT_FACTORYRESET +MT_FACTORYRESET:OK	

6. AT+MT_ATTNOTIFY 属性通知设置

开启/关闭属性通知。默认属性通知设置为开启状态。关闭后，属性值发生变化时Server端不会主动通知，需要Client端主动查询属性值。**不建议关闭。**

AT+MT_ATTNOTIFY 属性通知设置	
client	server
查询命令 AT+MT_ATTNOTIFY?	响应 +MT_ATTNOTIFY:OK,<enable> +MT_ATTNOTIFY:ERROR,<err_code>
执行命令 AT+MT_ATTNOTIFY=<enable>	响应 +MT_ATTNOTIFY:OK,<enable> +MT_ATTNOTIFY:ERROR,<err_code>
参数 <enable> enum类型 0, 关闭属性通知 1, 开启属性通知	参数 <enable> enum类型 0, 关闭属性通知 1, 开启属性通知

<err_code> int类型, 错误代码

示例

AT+MT_ATTNOTIFY?

+MT_ATTNOTIFY:OK,1

AT+MT_ATTNOTIFY=0

+MT_ATTNOTIFY:OK,0

7. AT+MT_UNPAIR 解绑-清网功能

查询或者设置解绑-清网功能是否开启以及解绑延迟多少（秒）后进行清网

模组在解绑时会主动通知MCU，通知格式为：>MT_EVENT:UNBIND_CAUSED_FACTORY_RESET

默认启用，默认延迟时长5s

注：在苹果界面需点击“从所有服务中移除”才会触发，点击“仅从我的家移除”不会触发

AT+MT_UNPAIR 查询/设置解绑-清网功能是否启用以及延迟时长

client

server

查询命令

响应

AT+MT_UNPAIR?

+MT_UNPAIR:OK,<enable>,<seconds>

+MT_UNPAIR:ERROR,<err_code>

执行命令

响应

AT+MT_UNPAIR=<enable>,<seconds>

+MT_UNPAIR:OK,<enable>,<seconds>

+MT_UNPAIR:ERROR,<err_code>

参数

参数

<enable> bool类型(0/1)

<num1> bool类型 (0/1)

0为关闭, 1为开启

<num2> uint32_t类型

<seconds> uint32_t类型

<err_code> int类型, 错误代码

解绑后延迟<seconds>秒清网

示例

AT+MT_UNPAIR?

+MT_UNPAIR:OK,1,5

AT+MT_UNPAIR=0,20

+MT_UNPAIR:OK,0,20

AT+MT_UNPAIR=2,20

+MT_UNPAIR:ERROR,-104

8. AT+MT_POLL_PERIOD 更改SED空闲时轮询周期

运行时更改sed空闲时轮询周期 (ms)

AT+MT_POLL_PERIOD 查询/设置运行时sed空闲时轮询周期

client	server
查询命令 AT+MT_POLL_PERIOD?	响应 +MT_POLL_PERIOD:OK,<period> +MT_POLL_PERIOD:ERROR,<err_code>
执行命令 AT+MT_POLL_PERIOD=<period>	响应 +MT_POLL_PERIOD:OK,<period> +MT_POLL_PERIOD:ERROR,<err_code>
参数 <period> 设置周期, 范围为10ms~236000ms	参数 <err_code> int类型, 错误代码
示例 AT+MT_POLL_PERIOD=? +MT_POLL_PERIOD:OK,500 AT+MT_POLL_PERIOD=800 +MT_POLL_PERIOD:OK,800	

9. AT+MT_TXPOWER 更改射频功率

运行时更改射频功率 (dBm)

AT+MT_TXPOWER 查询/设置运行时射频功率

client	server
--------	--------

查询命令 AT+MT_TXPOWER?	响应 +MT_TXPOWER:OK,<txpower> +MT_TXPOWER:ERROR ,<err_code>
执行命令 AT+MT_TXPOWER=< txpower >	响应 +MT_TXPOWER:OK, < txpower > +MT_TXPOWER:ERROR,<err_code>
参数 < txpower > 射频功率 范围为0~8dBm	参数 <err_code> int类型，错误代码
示例 AT+TXPOWER=? +MT_TXPOWER:OK,8 AT+MT_TXPOWER=6 +MT_TXPOWER:OK,6	

五、窗帘Matter公共指令集



matter窗帘物模型里分为Lift和Tilt两类控制方式。

object_id，表示当前可被操作对象的ID，该模块中当前可被操作对象的窗帘只有1个，对应ID为"1"，因此以下指令中的object_id参数都只用填"1"即可。

Lift: 指上下升降和左右滑动的控制方式，如**卷帘**和**开合帘**等。

Tilt: 指适用于带有垂直或水平条带的窗帘，如**百叶窗**的叶片控制等。

1. AT+MT_CURTAIN_TYPE_ATT 窗帘类型 属性操作?

TYPE属性，用来表示窗帘的主要类型。该属性为matter必选属性，需要在程序初始阶段进行配置。

属性默认值：0

AT+MT_CURTAIN_TYPE_ATT TYPE属性操作	
client	server
查询命令(read)	响应

AT+MT_CURTAIN_TYPE_ATT= <object_id>?	+MT_CURTAIN_TYPE_ATT:OK,<object_id>,<type> +MT_CURTAIN_TYPE_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_TYPE_ATT=<object_id>,<type>	响应 +MT_CURTAIN_TYPE_ATT:OK,<object_id>,<type> +MT_CURTAIN_TYPE_ATT:ERROR,<err_code>
参数 <object_id> uint16类型, 对象ID <type> enum类型, 窗帘的主要类型	参数 <object_id> uint16类型, 对象ID <type> enum类型, 窗帘的主要类型 <err_code> int类型, 错误代码
示例 查询 AT+MT_CURTAIN_TYPE_ATT=1? +MT_CURTAIN_TYPE_ATT:OK,1,0 写入 AT+MT_CURTAIN_TYPE_ATT=1,1 +MT_CURTAIN_TYPE_ATT:OK,1,1	

下表为TYPE属性值的定义:

Value	Type	Supported
0x00	Rollershade	Y
0x01	Rollershade - 2 Motor	Y
0x02	Rollershade - Exterior	Y
0x03	Rollershade - Exterior - 2 Motor	Y
0x04	Drapery (curtain)	Y
0x05	Awning	Y
0x06	Shutter	Y
0x07	Tilt Blind - Tilt Only	N
0x08	Tilt Blind - Lift & Tilt	Y
0x09	Projector Screen	Y

0xFF	Unknown	Y
------	---------	---

2. AT+MT_CURTAIN_EPTYPE_ATT 终端产品类型 属性操作ANYA

END PRODUCT TYPE属性，用来表示窗帘产品的类型，是对上面TYPE属性窗帘主要类型的详细补充。该属性为matter必选属性，需要在程序初始阶段进行配置。

属性默认值：0

AT+MT_CURTAIN_EPTYPE_ATT END PRODUCT TYPE属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_EPTYPE_ATT= <object_id>?	响应 +MT_CURTAIN_EPTYPE_ATT:OK,<object_id>,<type> +MT_CURTAIN_EPTYPE_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_EPTYPE_ATT= <object_id>,<type>	响应 +MT_CURTAIN_EPTYPE_ATT:OK,<object_id>,<type> +MT_CURTAIN_EPTYPE_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <type> enum类型，窗帘产品类型	参数 <object_id> uint16类型，对象ID <type> enum类型，窗帘产品类型 <err_code> int类型，错误代码
示例 查询 AT+MT_CURTAIN_EPTYPE_ATT=1? +MT_CURTAIN_EPTYPE_ATT:OK,1,0 写入 AT+MT_CURTAIN_EPTYPE_ATT=1,2 +MT_CURTAIN_EPTYPE_ATT:OK,1,2	

下表为END PRODUCT TYPE属性的定义：

Value	EndProductType	Indoor	Indicativ	Recommended Type & Features
-------	----------------	--------	-----------	-----------------------------

		Outdoor	e	Dimension	
0x00	Roller Shade	I	1D		0x00 Rollershade (Lift)
0x01	Roman Shade	I	1D		0x00 Rollershade (Lift)
0x02	Balloon Shade	I	1D		0x00 Rollershade (Lift)
0x03	Woven Wood	I	1D		0x00 Rollershade (Lift)
0x04	Pleated Shade	I	1D		0x00 Rollershade (Lift)
0x05	Cellular Shade	I	1D		0x00 Rollershade (Lift)
0x06	Layered Shade	I	1D		0x00 Rollershade (Lift)
0x07	Layered Shade 2D	I	2D		0x01 Rollershade - 2 Motor (Lift)
0x08	Sheer Shade	I	2D		0x08 Tilt Blind - Lift & Tilt (Lift, Tilt)
0x09	Tilt Only Interior Blind	I	1D		0x07 Tilt Blind - Tilt Only (Tilt)
0x0a	Interior Blind	I	2D		0x08 Tilt Blind - Lift & Tilt (Lift, Tilt)
0x0b	Vertical Blind, Strip Curtain	I	2D		0x08 Tilt Blind - Lift & Tilt (Lift, Tilt)
0x0c	Interior Venetian Blind	I	2D		0x08 Tilt Blind - Lift & Tilt (Lift, Tilt)
0x0d	Exterior Venetian Blind	O	2D		0x08 Tilt Blind - Lift & Tilt (Lift, Tilt)
0x0e	Lateral Left Curtain	I	1D		0x04 Drapery (Lift)
0x0f	Lateral Right Curtain	I	1D		0x04 Drapery (Lift)
0x10	Central Curtain	I	1D		0x04 Drapery (Lift)
0x11	Roller Shutter	O	1D		0x02 Rollershade - Exterior (Lift)
0x12	Exterior Vertical Screen	O	1D		0x02 Rollershade - Exterior (Lift)
0x13	Awning Terrace (Patio)	O	1D		0x05 Awning (Lift)
0x14	Awning Vertical Screen	O	1D		0x05 Awning (Lift)

0x15	Tilt Only Pergola	0	1D	0x06 Shutter (Tilt)
0x16	Swinging Shutter	0	1D	0x06 Shutter (Lift)
0x17	Sliding Shutter	0	1D	0x06 Shutter (Lift)
0xff	Unknown			

3. AT+MT_CURTAIN_CFGSTA_ATT 配置项 属性操作

4.

CONFIG STATUS属性，是设备配置项的状态标志位。该属性可在程序初始阶段配置。

属性默认值：9

AT+MT_CURTAIN_CFGSTA_ATT CONFIG STATUS属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_CFGSTA_ATT= <object_id>?	响应 +MT_CURTAIN_CFGSTA_ATT:OK,<object_id>,<status> +MT_CURTAIN_CFGSTA_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_CFGSTA_ATT= <object_id>,<status>	响应 +MT_CURTAIN_CFGSTA_ATT:OK,<object_id>,<status> +MT_CURTAIN_CFGSTA_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <status> uint8类型，状态标志	参数 <object_id> uint16类型，对象ID <status> uint8类型，状态标志 <err_code> int类型，错误代码
示例 查询 AT+MT_CURTAIN_CFGSTA_ATT=1? +MT_CURTAIN_CFGSTA_ATT:OK,1,1 写入 AT+MT_CURTAIN_CFGSTA_ATT=1,9 +MT_CURTAIN_CFGSTA_ATT:OK,1,9	

下表为CONFIG STATUS属性的标志位的定义：

Bit	含义	备注
0	0 = 不可操作 1 = 可操作	
1	预留	
2	0 = 运动方向正常 1 = 运动方向反向	
3	(Lift类型) 0 = 不支持位置感知 1 = 支持位置感知	
4	(Tilt类型) 0 = 不支持位置感知 1 = 支持位置感知	不支持Tilt类型，该位忽略
5	(Lift类型) 0 = 时间控制 1 = 编码器控制	
6	(Tilt类型) 0 = 时间控制 1 = 编码器控制	不支持Tilt类型，该位忽略

5. AT+MT_CURTAIN_OPSTA_ATT 操作状态 属性操作

OPERATIONAL STATUS 操作状态属性，表示当前的设备的操作状态。该属性跟踪当前正在进行的操作并适用于所有类型的设备。每当窗帘的状态发生变化时，需要及时同步到模块。

属性默认值：0

AT+MT_CURTAIN_OPSTA_ATT OPERATIONAL STATUS属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_OPSTA_ATT= <object_id>?	响应 +MT_CURTAIN_OPSTA_ATT:OK,<object_id>,<status>

	+MT_CURTAIN_OPSTA_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_OPSTA_ATT= <object_id>,<status>	响应 +MT_CURTAIN_OPSTA_ATT:OK,<object_id>,<status> +MT_CURTAIN_OPSTA_ATT:ERROR,<err_code>
参数 <object_id> uint16类型, 对象ID <status> uint8类型, 状态标志	参数 <object_id> uint16类型, 对象ID <status> uint8类型, 状态标志 <err_code> int类型, 错误代码
示例 查询 AT+MT_CURTAIN_OPSTA_ATT=1? +MT_CURTAIN_OPSTA_ATT:OK,1,5 写入 AT+MT_CURTAIN_OPSTA_ATT=1,10 +MT_CURTAIN_OPSTA_ATT:OK,1,10	

下表为OPERATIONAL STATUS属性的状态位的定义:

Bit	含义	备注
0..1	00b = 窗帘未移动 01b = 窗帘正在打开(opening) 10b = 窗帘正在关闭(closing)	该标志位不区分Lift类型和Tilt类型
2..3	(Lift类型) 00b = 窗帘未移动 01b = 窗帘正在打开(opening) 10b = 窗帘正在关闭(closing)	该标志位仅在Lift类型控制设备才有效
4..5	(Tilt类型) 00b = 窗帘未移动 01b = 窗帘正在打开(opening) 10b = 窗帘正在关闭(closing)	该标志位仅在Tilt类型控制设备才有效

6. AT+MT_CURTAIN_MODE_ATT MODE 属性操作

MODE属性，配置设备的模式。

属性默认值：0

AT+MT_CURTAIN_MODE_ATT MODE属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_MODE_ATT= <object_id>?	响应 +MT_CURTAIN_MODE_ATT:OK,<object_id>,<mode> +MT_CURTAIN_MODE_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_MODE_ATT=<object_id>, <mode>	响应 +MT_CURTAIN_MODE_ATT:OK,<object_id>,<mode> +MT_CURTAIN_MODE_ATT:ERROR,<err_code>
	通知消息 >MT_CURTAIN_MODE_ATT:<object_id>,<mode>
参数 <object_id> uint16类型，对象ID <mode> uint8类型，模式	参数 <object_id> uint16类型，对象ID <mode> uint8类型，模式 <err_code> int类型，错误代码
示例 查询 AT+MT_CURTAIN_MODE_ATT=1? +MT_CURTAIN_MODE_ATT:OK,1,0 写入 AT+MT_CURTAIN_MODE_ATT=1,1 +MT_CURTAIN_MODE_ATT:OK,1,1 模块主动通知 >MT_CURTAIN_MODE_ATT:1,1	

下表为MODE属性的标志位的定义：

BIT	含义
0	0 = 电机运动正常正转; 1 = 电机运动反转

1	0 = 正常模式; 1 = 校准模式
2	0 = 正常模式 1 = 维护模式
3	0 = LED关 1 = LED将显示反馈

7. AT+MT_CURTAIN_SAFSTA_ATT 设备安全状态 属性操作

SAFETY STATUS属性，安全状态标志。该属性反映了设备安全情况和常见问题。

属性默认值：0

AT+MT_CURTAIN_SAFSTA_ATT SAFETY STATUS属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_SAFSTA_ATT= <object_id>?	响应 +MT_CURTAIN_SAFSTA_ATT:OK,<object_id>,<status> +MT_CURTAIN_SAFSTA_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_SAFSTA_ATT= <object_id>,<status>	响应 +MT_CURTAIN_SAFSTA_ATT:OK,<object_id>,<status> +MT_CURTAIN_SAFSTA_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <status> uint16类型，状态标志	参数 <object_id> uint16类型，对象ID <status> uint16类型，状态标志 <err_code> int类型，错误代码
示例 查询 AT+MT_CURTAIN_SAFSTA_ATT=1? +MT_CURTAIN_SAFSTA_ATT:OK,1,0 写入 AT+MT_CURTAIN_SAFSTA_ATT=1,5 +MT_CURTAIN_SAFSTA_ATT:OK,1,5	

以下为SAFETY STATUS属性的标志位定义：

Bit	状态表示	描述
0	远程锁定	移动命令被忽略（锁定）。例如未获得授权。
1	篡改检测	在传感器或任何其他安全设备上检测到篡改。例如：设备在没有执行器的情况下被强制移动。
2	通讯失败	与传感器或其他安全设备的通信故障。
3	定位失败	设备未能到达所需位置。例如使用位置感知设备，在达到 TargetPosition 之前的时间已过。
4	过热保护	电机或电路过热保护已激活。
5	检测到障碍物	障碍物阻止执行器移动。
6	电源问题	设备存在与电源相关的问题或限制，例如设备在备用电池的帮助下运行，或者目前可能无法正常使用电源。
7	停止输入	本地安全传感器（不是直接障碍物）阻止移动（例如安全欧盟标准 EN60335）。
8	电机卡住	检测到与电机相关的机械问题。
9	硬件故障	电路、保险丝和其他电气问题。
10	手动操作	执行器是手动操作的，阻止执行器的正常移动（例如，执行器脱离）。
11	保护	保护被激活

8. AT+MT_CURTAIN_STOP_CMD 停止移动命令

STOP MOTION命令，接收到该命令后，需停止窗帘电机运动，同步TPP属性值和CPP属性值，将当前百分比位置值通过AT指令写入到TPP属性和CPP属性中。

AT+MT_CURTAIN_STOP_CMD STOP命令操作	
client	server
	通知消息 >MT_CURTAIN_STOP_CMD:<object_id>

参数

<object_id> uint16类型, 对象ID

示例

模块主动通知

>MT_CURTAIN_STOP_CMD:1

TPP: TARGET POSITION PERCENT, 见六.3.

CPP: CURRENT POSITION PERCENT, 见六.4.

9. AT+MT_CURTAIN_CPP_AUTOMODE CPP自动模式

- CPP自动模式开启后, 会根据TPP的值, 自动同步CPP的值。
- 当server端发送">MT_CURTAIN_TPP_ATT_TILT"或">MT_CURTAIN_TPP_ATT"通知时, 会在1秒后自动将当前的TPP属性值大小写入CPP属性 (CURRENT POSITION PERCENT) 中。
- 当client端使用方式为: 在窗帘电机移动过程中只上报一次当前位置。client端可以不用再上报, 开启CPP自动模式, 而由模块自动完成。

AT+MT_CURTAIN_CPP_AUTOMODE CPP自动模式

client	server
查询命令(read) AT+MT_CURTAIN_CPP_AUTOMODE?	响应 +MT_CURTAIN_CPP_AUTOMODE:OK,<status> +MT_CURTAIN_CPP_AUTOMODE:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_CPP_AUTOMODE=<status>	响应 +MT_CURTAIN_CPP_AUTOMODE:OK,<status> +MT_CURTAIN_CPP_AUTOMODE:ERROR,<err_code>
参数 <status> enum类型, 状态标志 0, 自动模式关闭; 1, 自动模式开启	参数 <status> enum类型, 状态标志 0, 自动模式关闭; 1, 自动模式开启 <err_code> int类型, 错误代码

示例

查询

AT+MT_CURTAIN_CPP_AUTOMODE?

+MT_CURTAIN_CPP_AUTOMODE:OK,0

写入

AT+MT_CURTAIN_CPP_AUTOMODE=1

+MT_CURTAIN_CPP_AUTOMODE:OK,1

六、(LIFT)窗帘Matter指令集

以下指令仅用于Lift控制方式的设备。

1. AT+MT_CURTAIN_IOL_ATT 限位开 属性操作

INSTALLED OPEN LIMIT属性，窗帘开的限制位置。

属性默认值：0

AT+MT_CURTAIN_IOL_ATT INSTALLED OPEN LIMIT属性操作

client	server
查询命令(read) AT+MT_CURTAIN_IOL_ATT=<object_id>?	响应 +MT_CURTAIN_IOL_ATT:OK,<object_id>,<limit val> +MT_CURTAIN_IOL_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_IOL_ATT=<object_id>,<limit val>	响应 +MT_CURTAIN_IOL_ATT:OK,<object_id>,<limit val> +MT_CURTAIN_IOL_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <limit val> int16类型，限位值	参数 <object_id> uint16类型，对象ID <limit val> int16类型，限位值 <err_code> int类型，错误代码

示例

查询

AT+MT_CURTAIN_IOL_ATT=1?

+MT_CURTAIN_IOL_ATT:OK,1,5000

写入

AT+MT_CURTAIN_IOL_ATT=1,10000

+MT_CURTAIN_IOL_ATT:OK,1,10000

2. AT+MT_CURTAIN_ICL_ATT 限位关 属性操作

INSTALLED CLOSE LIMIT属性，窗帘关的限制位置。

属性默认值：0

AT+MT_CURTAIN_ICL_ATT INSTALLED CLOSE LIMIT属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_ICL_ATT=<object_id>?	响应 +MT_CURTAIN_ICL_ATT:OK,<object_id>,<limit val> +MT_CURTAIN_ICL_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_ICL_ATT=<object_id>,<limit val>	响应 +MT_CURTAIN_ICL_ATT:OK,<object_id>,<limit val> +MT_CURTAIN_ICL_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <limit val> int16类型，限位值	参数 <object_id> uint16类型，对象ID <limit val> int16类型，限位值 <err_code> int类型，错误代码
示例 查询 AT+MT_CURTAIN_ICL_ATT=1? +MT_CURTAIN_ICL_ATT:OK,1,50 写入 AT+MT_CURTAIN_ICL_ATT=1,10 +MT_CURTAIN_ICL_ATT:OK,1,10	

3. AT+MT_CURTAIN_TPP_ATT 目标百分比位置 属性操作

TARGET POSITION PERCENT属性，目标位置百分比，单位0.01%，步进值0.01%，该属性值为1则表示为0.01%，10000表示为100%。当下发STOP MOTION命令时，请该属性值将设置为与CPP(CURRENT POSITION PERCENT)属性相同的值。

属性默认值：0

AT+MT_CURTAIN_TPP_ATT TARGET POSITION PERCENT属性操作

client	server
查询命令(read) AT+MT_CURTAIN_TPP_ATT=<object_id>?	响应 +MT_CURTAIN_TPP_ATT:OK,<object_id>,<percent100ths> +MT_CURTAIN_TPP_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_TPP_ATT=<object_id>,<percent100ths>	响应 +MT_CURTAIN_TPP_ATT:OK,<object_id>,<percent100ths> +MT_CURTAIN_TPP_ATT:ERROR,<err_code>
	通知消息 >MT_CURTAIN_TPP_ATT:<object_id>,<percent100ths>
参数 <object_id> uint16类型, 对象ID <percent100ths> uint16类型, 目标位置百分比 数值0~10000, 0~100%	参数 <object_id> uint16类型, 对象ID <percent100ths> uint16类型, 目标位置百分比 数值0~10000, 0~100% <err_code> int类型, 错误代码
示例 查询 AT+MT_CURTAIN_TPP_ATT=1? +MT_CURTAIN_TPP_ATT:OK,1,0 写入 AT+MT_CURTAIN_TPP_ATT=1,10000 +MT_CURTAIN_TPP_ATT:OK,1,10000 模块主动通知 >MT_CURTAIN_TPP_ATT:1,10000	

4. AT+MT_CURTAIN_CPP_ATT 当前百分比位置 属性操作

CURRENT POSITION PERCENT属性, 当前位置百分比, 表示当前窗帘的位置, 单位0.01%, 步进值0.01%, 该属性值为1则表示为0.01%, 10000表示为100%。

属性默认值: 0

AT+MT_CURTAIN_CPP_ATT CURRENT POSITION PERCENT属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_CPP_ATT=<object_id>?	响应 +MT_CURTAIN_CPP_ATT:OK,<object_id>,<percent100ths> +MT_CURTAIN_CPP_ATT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_CPP_ATT=<object_id>,<percent100ths>	响应 +MT_CURTAIN_CPP_ATT:OK,<object_id>,<percent100ths> +MT_CURTAIN_CPP_ATT:ERROR,<err_code>
参数 <object_id> uint16类型, 对象ID <percent100ths> uint16类型, 目标位置百分比 数值0~10000, 0~100%	参数 <object_id> uint16类型, 对象ID <percent100ths> uint16类型, 目标位置百分比 数值0~10000, 0~100% <err_code> int类型, 错误代码
示例 查询 AT+MT_CURTAIN_CPP_ATT=1? +MT_CURTAIN_CPP_ATT:OK,1,0 写入 AT+MT_CURTAIN_CPP_ATT=1,10000 +MT_CURTAIN_CPP_ATT:OK,1,10000	

5. AT+MT_CURTAIN_CTPP_UPDATE 同步更新TPP和CPP

同步写入更新CPP（CURRENT POSITION PERCENT）属性和TPP（TARGET POSITION PERCENT）属性。该命令用于以下情况：

- a. 1.client端收到MT_CURTAIN_STOP_CMD命令后，设备停止运动后，通过该命令更新CPP和TPP属性的值。
- b. 设备通过matter以外方式控制，如遥控，设备到达指令位置并停止后，通过该命令更新CPP和TPP属性的值。

AT+MT_CURTAIN_CTPP_UPDATE 同步更新TPP和CPP

client	server
执行命令(write) AT+MT_CURTAIN_CTPP_UPDATE=<object_id>,<percent100ths>	响应 +MT_CURTAIN_CTPP_UPDATE:OK,<object_id>,<percent100ths> +MT_CURTAIN_CTPP_UPDATE:ERROR,<err_code>
参数 <object_id> uint16类型, 对象ID <percent100ths> uint16类型, 位置百分比 数值0~10000, 0~100%	参数 <object_id> uint16类型, 对象ID <percent100ths> uint16类型, 位置百分比 数值0~10000, 0~100% <err_code> int类型, 错误代码
示例 AT+MT_CURTAIN_CTPP_UPDATE=1,5000 +MT_CURTAIN_CTPP_UPDATE:OK,1,5000	

七、(Tilt)窗帘Matter指令集

以下指令仅用于Tilt控制方式的设备。

1. AT+MT_CURTAIN_IOL_ATT_TILT 限位开 属性操作

INSTALLED OPEN LIMIT属性, 窗帘开的限制位置。

属性默认值: 0

AT+MT_CURTAIN_IOL_ATT_TILT INSTALLED OPEN LIMIT属性操作	
client	server
查询命令(read) AT+MT_CURTAIN_IOL_ATT_TILT=<object_id>?	响应 +MT_CURTAIN_IOL_ATT_TILT:OK,<object_id>,<limit val> +MT_CURTAIN_IOL_ATT_TILT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_IOL_ATT_TILT=<object_id>,<limit val>	响应 +MT_CURTAIN_IOL_ATT_TILT:OK,<object_id>,<limit val> +MT_CURTAIN_IOL_ATT_TILT:ERROR,<err_code>

参数		参数	
<object_id>	uint16类型, 对象ID	<object_id>	uint16类型, 对象ID
<limit val>	int16类型, 限位值	<limit val>	int16类型, 限位值
		<err_code>	int类型, 错误代码
示例			
查询			
	AT+MT_CURTAIN_IOL_ATT_TILT=1?		
	+MT_CURTAIN_IOL_ATT_TILT:OK,1,5000		
写入			
	AT+MT_CURTAIN_IOL_ATT_TILT=1,10000		
	+MT_CURTAIN_IOL_ATT_TILT:OK,1,10000		

2. AT+MT_CURTAIN_ICL_ATT_TILT 限位关属性操作

INSTALLED CLOSE LIMIT属性, 窗帘关的限制位置。

属性默认值: 0

AT+MT_CURTAIN_ICL_ATT_TILT INSTALLED CLOSE LIMIT属性操作			
client	server		
查询命令(read)	响应		
AT+MT_CURTAIN_ICL_ATT_TILT= <object_id>?	+MT_CURTAIN_ICL_ATT_TILT:OK,<object_id>,<limit val> +MT_CURTAIN_ICL_ATT_TILT:ERROR,<err_code>		
执行命令(write)	响应		
AT+MT_CURTAIN_ICL_ATT_TILT= <object_id>,<limit val>	+MT_CURTAIN_ICL_ATT_TILT:OK,<object_id>,<limit val> +MT_CURTAIN_ICL_ATT_TILT:ERROR,<err_code>		
参数		参数	
<object_id>	uint16类型, 对象ID	<object_id>	uint16类型, 对象ID
<limit val>	int16类型, 限位值	<limit val>	int16类型, 限位值
		<err_code>	int类型, 错误代码
示例			
查询			
	AT+MT_CURTAIN_ICL_ATT_TILT=1?		

+MT_CURTAIN_ICL_ATT_TILT:OK,1,50

写入

AT+MT_CURTAIN_ICL_ATT_TILT=1,10

+MT_CURTAIN_ICL_ATT_TILT:OK,1,10

3. AT+MT_CURTAIN_TPP_ATT_TILT 目标百分比位置 属性操作

TARGET POSITION PERCENT属性，目标位置百分比，单位0.01%，步进值0.01%，该属性值为1则表示为0.01%，10000表示为100%。当下发STOP MOTION命令时，请该属性值将设置为与CPP(CURRENT POSITION PERCENT)属性相同的值。

属性默认值：0

AT+MT_CURTAIN_TPP_ATT_TILT TARGET POSITION PERCENT属性操作

client	server
查询命令(read) AT+MT_CURTAIN_TPP_ATT_TILT=<object_id>?	响应 +MT_CURTAIN_TPP_ATT_TILT:OK,<object_id>,<percent100ths> +MT_CURTAIN_TPP_ATT_TILT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_TPP_ATT_TILT=<object_id>,<percent100ths>	响应 +MT_CURTAIN_TPP_ATT_TILT:OK,<object_id>,<percent100ths> +MT_CURTAIN_TPP_ATT_TILT:ERROR,<err_code>
	通知消息 >MT_CURTAIN_TPP_ATT_TILT:<object_id>,<percent100ths>
参数 <object_id> uint16类型，对象ID <percent100ths> uint16类型，目标位置百分比 数值0~10000，0~100%	参数 <object_id> uint16类型，对象ID <percent100ths> uint16类型，目标位置百分比 数值0~10000，0~100% <err_code> int类型，错误代码

示例

查询

AT+MT_CURTAIN_TPP_ATT_TILT=1?

+MT_CURTAIN_TPP_ATT_TILT:OK,1,0

写入

AT+MT_CURTAIN_TPP_ATT_TILT=1,10000

+MT_CURTAIN_TPP_ATT_TILT:OK,1,10000

模块主动通知

>MT_CURTAIN_TPP_ATT_TILT:1,10000

4. AT+MT_CURTAIN_CPP_ATT_TILT 当前百分比位置 属性操作

CURRENT POSITION PERCENT属性，当前位置百分比，表示当前窗帘的位置，单位0.01%，步进值0.01%，该属性值为1则表示为0.01%，10000表示为100%。

属性默认值：0

AT+MT_CURTAIN_CPP_ATT_TILT CURRENT POSITION PERCENT属性操作

client	server
查询命令(read) AT+MT_CURTAIN_CPP_ATT_TILT= <object_id>?	响应 +MT_CURTAIN_CPP_ATT_TILT:OK,<object_id>, <percent100ths> +MT_CURTAIN_CPP_ATT_TILT:ERROR,<err_code>
执行命令(write) AT+MT_CURTAIN_CPP_ATT_TILT= <object_id>,<percent100ths>	响应 +MT_CURTAIN_CPP_ATT_TILT:OK,<object_id>, <percent100ths> +MT_CURTAIN_CPP_ATT_TILT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <percent100ths> uint16类型，目标位置 百分比 数值0~10000，0~100%	参数 <object_id> uint16类型，对象ID <percent100ths> uint16类型，目标位置百分比 数值0~10000，0~100% <err_code> int类型，错误代码

示例

查询

AT+MT_CURTAIN_CPP_ATT_TILT=1?

+MT_CURTAIN_CPP_ATT_TILT:OK,1,0

写入

AT+MT_CURTAIN_CPP_ATT_TILT=1,10000

+MT_CURTAIN_CPP_ATT_TILT:OK,1,10000

5. AT+MT_CURTAIN_CTPP_TILT_UPDATE 同步更新TPP和CPP

同步写入更新CPP（CURRENT POSITION PERCENT）属性和TPP（TARGET POSITION PERCENT）属性。该命令用于以下情况：

- 1.client端收到MT_CURTAIN_STOP_CMD命令后，设备停止运动后，通过该命令更新CPP和TPP属性的值。
- 设备通过matter以外方式控制，如遥控，设备到达指令位置并停止后，通过该命令更新CPP和TPP属性的值。

AT+MT_CURTAIN_CTPP_TILT_UPDATE 同步更新TPP和CPP

client	server
执行命令(write) AT+MT_CURTAIN_CTPP_TILT_UPDATE= <object_id>,<percent100ths>	响应 +MT_CURTAIN_CTPP_TILT_UPDATE:OK,<object_id>, <percent100ths> +MT_CURTAIN_CTPP_TILT_UPDATE:ERROR, <err_code>
参数 <object_id> uint16类型，对象ID <percent100ths> uint16类型，位置百分比 数值0~10000，0~100%	参数 <object_id> uint16类型，对象ID <percent100ths> uint16类型，位置百分比 数值0~10000，0~100% <err_code> int类型，错误代码
示例 AT+MT_CURTAIN_CTPP_TILT_UPDATE=1,5000 +MT_CURTAIN_CTPP_TILT_UPDATE:OK,1,5000	

八、电源信息 Matter指令集

matter支持电源信息的配置，方便用户了解到设备的电源情况。电源信息作为matter的功能集，在AT指令中也需要通过对象ID的方式告知并操作。在该模块中，电源信息的对象id为1，以下指令的object_id参数填写“1”即可。

! ps:若要使用该功能，需要模块固件支持，使用前需与我司工作人员沟通确认。

• 可充电电源

可充电电源指设备电源由电池供电，并且可以通过有线或无线对设备进行充电。以下都是可充电电源信息相关的配置属性的AT指令。

1. AT+MT_PSBAT_VOL_ATT 电池电压属性操作

电池电压属性，单位mv。

属性默认值：0

AT+MT_PSBAT_VOL_ATT 电池电压属性	
client	server
查询命令(read) AT+MT_PSBAT_VOL_ATT= <object_id>?	响应 +MT_PSBAT_VOL_ATT:OK,<object_id>,<voltage> +MT_PSBAT_VOL_ATT:ERROR,<err_code>
执行命令(write) AT+MT_PSBAT_VOL_ATT= <object_id>,<voltage>	响应 +MT_PSBAT_VOL_ATT:OK,<object_id>,<voltage> +MT_PSBAT_VOL_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <voltage> uint32类型，电池电压 mv	参数 <object_id> uint16类型，对象ID <voltage> uint32类型，电池电压mv <err_code> int类型，错误代码
示例 查询 AT+MT_PSBAT_VOL_ATT=1? +MT_PSBAT_VOL_ATT:OK,1,3000 写入 AT+MT_PSBAT_VOL_ATT=1,3300	

+MT_PSBAT_VOL_ATT:OK,1,3300

2. AT+MT_PSBAT_PR_ATT 电池剩余电量 属性操作

电池剩余电量百分比，数值范围0~100。

属性默认值：0

AT+MT_PSBAT_PR_ATT 电池剩余电量属性	
client	server
查询命令(read) AT+MT_PSBAT_PR_ATT=<object_id>?	响应 +MT_PSBAT_PR_ATT:OK,<object_id>,<percent> +MT_PSBAT_PR_ATT:ERROR,<err_code>
执行命令(write) AT+MT_PSBAT_PR_ATT=<object_id>,<percent>	响应 +MT_PSBAT_PR_ATT:OK,<object_id>,<percent> +MT_PSBAT_PR_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <percent> uint8类型，剩余电量百分比	参数 <object_id> uint16类型，对象ID <percent> uint8类型，剩余电量百分比 <err_code> int类型，错误代码
示例 查询 AT+MT_PSBAT_PR_ATT=1? +MT_PSBAT_PR_ATT:OK,1,80 写入 AT+MT_PSBAT_PR_ATT=1,70 +MT_PSBAT_PR_ATT:OK,1,70	

3. AT+MT_PSBAT_CHGLV_ATT 电池电量等级 属性操作

电池电量等级，用于指示电池当前的电量状态，以及是否需要充电等。

属性默认值：0

电池电量等级属性	
client	server
查询命令(read) AT+MT_PSBAT_CHGLV_ATT=<object_id>?	响应 +MT_PSBAT_CHGLV_ATT:OK,<object_id>,<level> +MT_PSBAT_CHGLV_ATT:ERROR,<err_code>
执行命令(write) AT+MT_PSBAT_CHGLV_ATT=<object_id>,<level>	响应 +MT_PSBAT_CHGLV_ATT:OK,<object_id>,<level> +MT_PSBAT_CHGLV_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <level> enum类型，电量等级	参数 <object_id> uint16类型，对象ID <level> enum类型，电量等级 <err_code> int类型，错误代码
示例 查询 AT+MT_PSBAT_CHGLV_ATT=1? +MT_PSBAT_CHGLV_ATT:OK,1,0 写入 AT+MT_PSBAT_CHGLV_ATT=1,1 +MT_PSBAT_CHGLV_ATT:OK,1,1	

下表为level参数枚举值的定义：

数值	含义
0	电量正常
1	电量低
2	严重缺电，需立即更换电池或充电

4. AT+MT_PSBAT_REPND_ATT 电池更换需求 属性操作

电池更换需求属性，用于指示电池是否需要更换。如电池老化严重需要更换电池时，将对应的值写入该属性即可。

属性默认值：0

AT+ MT_PSBAT_REPND_ATT 电池更换需求属性	
client	server
查询命令(read) AT+MT_PSBAT_REPND_ATT=<object_id>?	响应 +MT_PSBAT_REPND_ATT:OK,<object_id>,<needed> +MT_PSBAT_REPND_ATT:ERROR,<err_code>
执行命令(write) AT+MT_PSBAT_REPND_ATT=<object_id>,<needed>	响应 +MT_PSBAT_REPND_ATT:OK,<object_id>,<needed> +MT_PSBAT_REPND_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <needed> enum类型， 0，无需更换电池 1，需要更换电池	参数 <object_id> uint16类型，对象ID <needed> enum类型， 0，无需更换电池 1，需要更换电池 <err_code> int类型，错误代码
示例 查询 AT+MT_PSBAT_REPND_ATT=1? +MT_PSBAT_REPND_ATT:OK,1,0 写入 AT+MT_PSBAT_REPND_ATT=1,1 +MT_PSBAT_REPND_ATT:OK,1,1	

5. AT+MT_PSBAT_REPBLT_ATT 电池可更换性 属性操作

电池可更换性属性，用于指示电池更换相关的规范性：电池是否可以更换；电池是否可以由用户自行更换；电池是否只能由厂商更换。

属性默认值：0

AT+ MT_PSBAT_REPBLT_ATT 电池可更换性属性

client	server
<p>查询命令(read)</p> <p>AT+MT_PSBAT_REPBLT_ATT=<object_id>?</p>	<p>响应</p> <p>+MT_PSBAT_REPBLT_ATT:OK,<object_id>,<specification></p> <p>+MT_PSBAT_REPBLT_ATT:ERROR,<err_code></p>
<p>执行命令(write)</p> <p>AT+MT_PSBAT_REPBLT_ATT=<object_id>,<specification></p>	<p>响应</p> <p>+MT_PSBAT_REPBLT_ATT:OK,<object_id>,<specification></p> <p>+MT_PSBAT_REPBLT_ATT:ERROR,<err_code></p>
<p>参数</p> <p><object_id> uint16类型, 对象ID</p> <p><specification> enum类型, 电池更换规范</p>	<p>参数</p> <p><object_id> uint16类型, 对象ID</p> <p><specification> enum类型, 电池更换规范</p> <p><err_code> int类型, 错误代码</p>
<p>示例</p> <p>查询</p> <p>AT+MT_PSBAT_REPBLT_ATT=1?</p> <p>+MT_PSBAT_REPBLT_ATT:OK,1,1</p> <p>写入</p> <p>AT+MT_PSBAT_REPBLT_ATT=1,2</p> <p>+MT_PSBAT_REPBLT_ATT:OK,1,2</p>	

下表为specification参数的枚举值定义：

数值	含义
0	未知
1	不可更换
2	用户可自行更换
3	只能由厂商更换

6. AT+MT_PSBAT_CHGSTA_ATT 电池充电状态 属性操作

电池充电状态属性，用于指示电池当前的充电状态：正在充电；已充满；未在充电。

属性默认值：3

AT+ MT_PSBAT_CHGSTA_ATT 电池充电状态属性	
client	server
查询命令(read) AT+MT_PSBAT_CHGSTA_ATT=<object_id>?	响应 +MT_PSBAT_CHGSTA_ATT:OK,<object_id>,<status> +MT_PSBAT_CHGSTA_ATT:ERROR,<err_code>
执行命令(write) AT+MT_PSBAT_CHGSTA_ATT=<object_id>,<status>	响应 +MT_PSBAT_CHGSTA_ATT:OK,<object_id>,<status> +MT_PSBAT_CHGSTA_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <status> enum类型，充电状态	参数 <object_id> uint16类型，对象ID <status> enum类型，充电状态 <err_code> int类型，错误代码
示例 查询 AT+MT_PSBAT_CHGSTA_ATT=1? +MT_PSBAT_CHGSTA_ATT:OK,1,1 写入 AT+MT_PSBAT_CHGSTA_ATT=1,2 +MT_PSBAT_CHGSTA_ATT:OK,1,2	

下表为status参数的枚举值定义：

数值	含义
0	未知
1	正在充电
2	电池已满电
3	不在充电状态

7. AT+MT_PSBAT_WKWCHG_ATT 电池充电可运行属性操作

电池充电可运行属性，用于指示电池在充电时，设备是否可以保持正常运行。

属性默认值：1

AT+ MT_PSBAT_WKWCHG_ATT 电池充电可运行属性	
client	server
查询命令(read) AT+MT_PSBAT_WKWCHG_ATT=<object_id>?	响应 +MT_PSBAT_WKWCHG_ATT:OK,<object_id>,<running> +MT_PSBAT_WKWCHG_ATT:ERROR,<err_code>
执行命令(write) AT+MT_PSBAT_WKWCHG_ATT=<object_id>,<running>	响应 +MT_PSBAT_WKWCHG_ATT:OK,<object_id>,<running> +MT_PSBAT_WKWCHG_ATT:ERROR,<err_code>
参数 <object_id> uint16类型，对象ID <running> enum类型， 0，电池充电时禁止运行设备 1，电池充电时可以正常运行设备	参数 <object_id> uint16类型，对象ID <running> enum类型， 0，电池充电时禁止运行设备 1，电池充电时可以正常运行设备 <err_code> int类型，错误代码
示例 查询 AT+MT_PSBAT_WKWCHG_ATT=1? +MT_PSBAT_WKWCHG_ATT:OK,1,1 写入 AT+MT_PSBAT_WKWCHG_ATT=1,0 +MT_PSBAT_WKWCHG_ATT:OK,1,0	

九. (可选) 窗帘蓝牙指令集

蓝牙错误码集

错误码	错误码含义
-65537	蓝牙未建立连接
-65538	未使能Notify
-65539	蓝牙发送失败
-65552	查询失败
-65568	设置失败
-65792	BLE模式未启用
-66408	BLE模式已启用

1. >BLE_RECV 蓝牙信息通知

该命令用于模组收到BLE数据后，主动通知给MCU。

>BLE_RECV 蓝牙信息通知
Server
主动通知 >BLE_RECV:<bytes>,<data>
参数 <bytes> int类型，字节数 <data> 字符串类型，要透传的数据
示例 >BLE_RECV:4,aabbccdd

2. >BLE_STATE 蓝牙状态通知

在每次蓝牙状态发生变化时，会通知MCU当前蓝牙状态。

>BLE_STATE: 蓝牙状态通知

Server

主动通知

>BLE_STATE:<state>

参数

<state> int类型，代表不同的蓝牙状态

0: HooRii蓝牙广播未开启

1: HooRii蓝牙广播已开启

2: 蓝牙已连接

3: 蓝牙连接已断开

示例

>BLE_STATE:0

3. >BLE_NOTIFY_STATE 蓝牙TX Notify属性状态通知

在每次TX Notify使能发生变化时，模组主动通知当前状态。

>BLE_NOTIFY_STATE: 蓝牙notify属性状态通知

Server

主动通知

>BLE_NOTIFY_STATE:<state>

参数

<state> int类型，代表不同的蓝牙TX Notify属性状态

0: Notify未使能

1: Notify已使能

示例

>BLE_NOTIFY_STATE:1

4. AT+BLE_STATE 查询蓝牙状态

查询蓝牙当前状态

AT+BLE_STATE 查询蓝牙状态	
client	server
执行命令 AT+BLE_STATE?	响应 +BLE_STATE:OK,<status>
	参数 <status> int类型, 状态代码 0: HooRii蓝牙广播未开启 1: HooRii蓝牙广播已开启 2: 蓝牙已连接 3: 蓝牙连接已断开
示例 AT+BLE_STATE? +BLE_STATE:OK,0	

5. AT+BLE_NOTIFY_STATE 查询Notify使能状态

查询Notify使能状态

AT+BLE_NOTIFY_STATE 查询蓝牙状态	
client	server
执行命令 AT+BLE_NOTIFY_STATE?	响应 +BLE_NOTIFY_STATE:OK,<status>
	参数 <state> int类型, 状态代码 0: Notify未使能 1: Notify已使能
示例 AT+BLE_NOTIFY_STATE?	

+BLE_NOTIFY_STATE:OK,1

6. AT+BLE_TRANSPORT 蓝牙发送数据

设置需要蓝牙发送出去的数据

AT+BLE_TRANSPORT 蓝牙发送	
client	server
执行命令 AT+BLE_TRANSPORT=<bytes>,<data>	响应 +BLE_TRANSPORT:OK,<data> +BLE_TRANSPORT:ERROR,<err_code>
参数 <bytes> 字节数 <data> 字符串类型，要发送的数据	参数 <err_code> int类型，错误代码
示例 AT+BLE_TRANSPORT=4,aabbccdd +BLE_TRANSPORT:OK,aabbccdd	

7. AT+BLE_ADV_DATA 设置/读取蓝牙广播数据

该命令用以设置/读取蓝牙广播时的数据内容

注意：TLV中的length相加请勿超过31字节

AT+BLE_ADV_DATA 设置/读取蓝牙广播数据	
client	server
查询命令(read) AT+BLE_ADV_DATA?	响应 +BLE_ADV_DATA:OK,<data>
执行命令(write) AT+BLE_ADV_DATA=<bytes>,<data>	响应 +BLE_ADV_DATA:OK,<data> +BLE_ADV_DATA:ERROR,<err_code>
参数	参数

<bytes> 字节数	<err_code> int类型, 错误代码
<data> 十六进制字符串类型, 要发送的TLV数据	

示例

查询

```
AT+BLE_ADV_DATA?
+BLE_ADV_DATA:OK,020106
```

操作

```
AT+BLE_ADV_DATA=3,020106
+BLE_ADV_DATA:OK,020106
```

8. AT+BLE_SCANRESP_DATA 设置/读取蓝牙扫描回复数据

该命令用以设置/读取蓝牙被扫描后的回复包

注意: TLV中的length相加请勿超过31字节

AT+ BLE_SCANRESP_DATA 设置/读取蓝牙广播数据	
client	server
查询命令(read) AT+BLE_SCANRESP_DATA?	响应 +BLE_SCANRESP_DATA:OK,<data>
执行命令(write) AT+BLE_SCANRESP_DATA=<bytes>,<data>	响应 +BLE_SCANRESP_DATA:OK,<data> +BLE_SCANRESP_DATA:ERROR,<err_code>
参数 <bytes> 字节数 <data> 十六进制字符串类型, 要发送的TLV数据	参数 <err_code> int类型, 错误代码
<p>示例</p> <p>查询</p> <pre>AT+BLE_SCANRESP_DATA? +BLE_SCANRESP_DATA:OK,020106</pre> <p>操作</p> <pre>AT+BLE_SCANRESP_DATA=3,020106</pre>	

+BLE_SCANRESP_DATA:OK,020106

9. AT+BLE_MODE 切换蓝牙模式开/关

该命令用于切换蓝牙/matter模式，执行后模组将先复位，之后切换模式并通知。

AT+BLE_MODE 切换蓝牙模式	
client	server
执行命令 AT+BLE_MODE=<mode>	响应 +BLE_MODE:OK,<mode> +BLE_MODE:ERROR,<err_code>
参数 <mode> int类型，切换的模式 0: 切换Matter模式 1: 切换HooRii蓝牙模式	参数 <err_code> int类型，错误代码
示例 AT+BLE_MODE=0 +BLE_MODE:OK,0	

10. AT+BLE_SERUUID 设置/读取服务UUID

注：UUID默认使用128位，默认UUID: `6e400001b5a3f393e0a9e50e24dcca9e`

厂家也可选择使用16位UUID，由和众模组嵌入默认UUID中。

替换规则为: `6e40XXXXb5a3f393e0a9e50e24dcca9e`

也可以使用自定义128位UUID，替代默认UUID。

AT+BLE_SERUUID 设置/读取服务UUID	
client	server
查询命令(read) AT+BLE_SERUUID?	响应 +BLE_SERUUID:OK,<UUID>
执行命令(write) AT+BLE_SERUUID=<UUID>	响应 +BLE_SERUUID:OK,<UUID>

			+BLE_SERUUID:ERROR,<err_code>
参数		参数	
<UUID>	十六进制字符串类型，	<err_code>	int类型，错误代码
示例			
操作			
	AT+BLE_SERUUID=1234		
	+BLE_SERUUID:OK,1234		
查询			
	AT+BLE_SERUUID?		
	+BLE_SERUUID:OK,6e401234b5a3f393e0a9e50e24dcca9e		

11. AT+BLE_TXUUID 设置/读取TX特征UUID

注：UUID默认使用128位，默认UUID: `6e400003b5a3f393e0a9e50e24dcca9e`

厂家也可选择使用16位UUID，由和众模组嵌入默认UUID中。

替换规则为: `6e40XXXXb5a3f393e0a9e50e24dcca9e`

也可以使用自定义128位UUID，替代默认UUID。

AT+BLE_TXUUID 设置/读取TX特征UUID	
client	server
查询命令(read)	响应
AT+BLE_TXUUID?	+BLE_TXUUID:OK,<UUID>
执行命令(write)	响应
AT+BLE_TXUUID=<UUID>	+BLE_TXUUID:OK,<UUID> +BLE_TXUUID:ERROR,<err_code>
参数	参数
<UUID>	<err_code>
十六进制字符串类型，	int类型，错误代码
示例	
查询	

AT+BLE_TXUUID=5678

+BLE_TXUUID:OK,1234

查询

AT+BLE_TXUUID?

+BLE_TXUUID:OK,6e405678b5a3f393e0a9e50e24dcca9e

12. AT+BLE_RXUUID 设置/读取RX特征UUID

注：UUID默认使用128位，默认UUID: 6e400002b5a3f393e0a9e50e24dcca9e

厂家也可选择使用16位UUID，由和众模组嵌入默认UUID中。

替换规则为： 6e40XXXXb5a3f393e0a9e50e24dcca9e

也可以使用自定义128位UUID，替代默认UUID。

AT+BLE_RXUUID 设置/读取RX特征UUID	
client	server
查询命令(read) AT+BLE_RXUUID?	响应 +BLE_RXUUID:OK,<UUID>
执行命令(write) AT+BLE_RXUUID=<UUID>	响应 +BLE_RXUUID:OK,<UUID> +BLE_RXUUID:ERROR,<err_code>
参数 <UUID> 十六进制字符串类型，	参数 <err_code> int类型，错误代码
示例 AT+BLE_RXUUID=7890 +BLE_RXUUID:OK,7890 查询 AT+BLE_RXUUID? +BLE_RXUUID:OK,6e407890b5a3f393e0a9e50e24dcca9e	

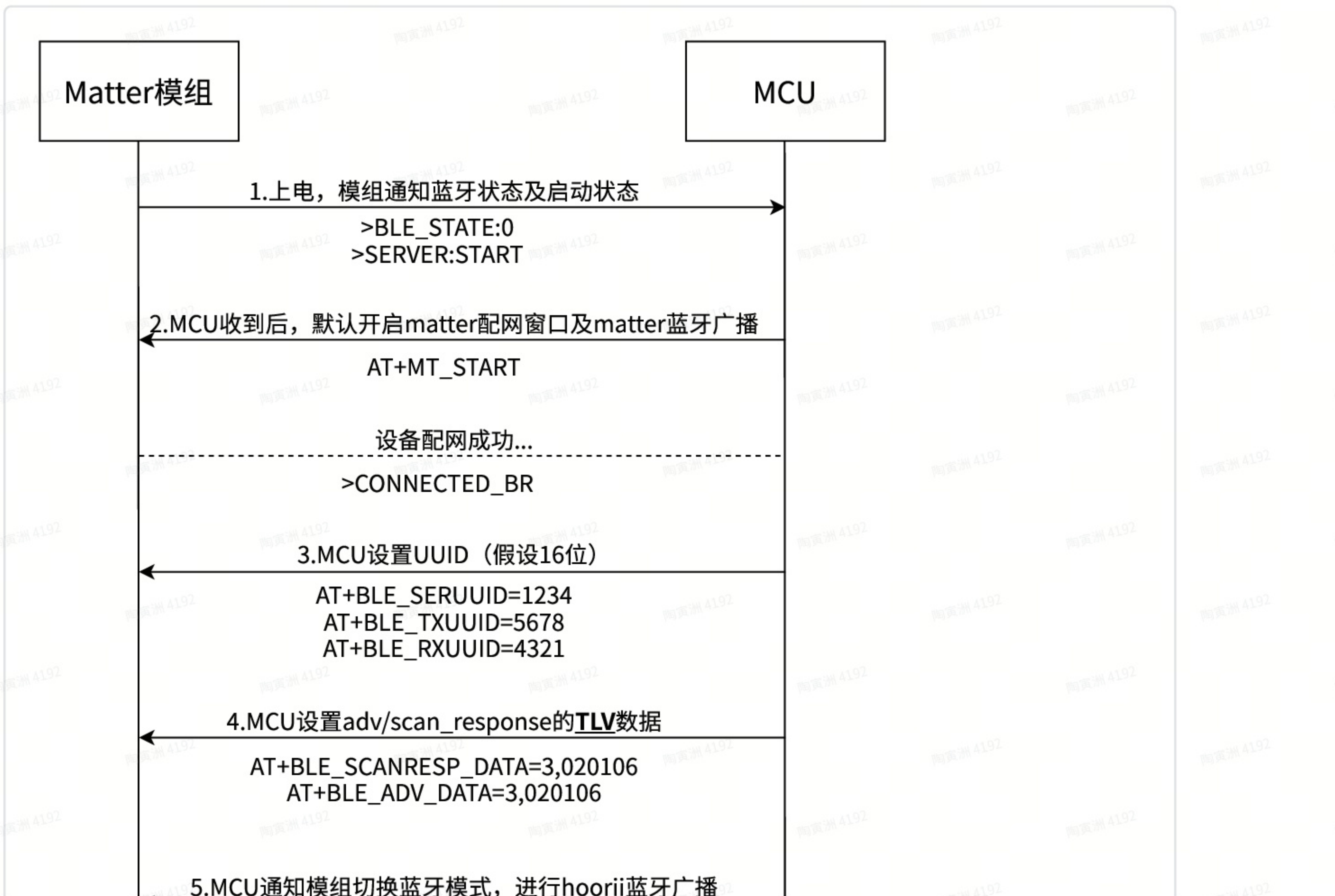
13. AT+BLE_MAC_ADDR 查询蓝牙MAC地址

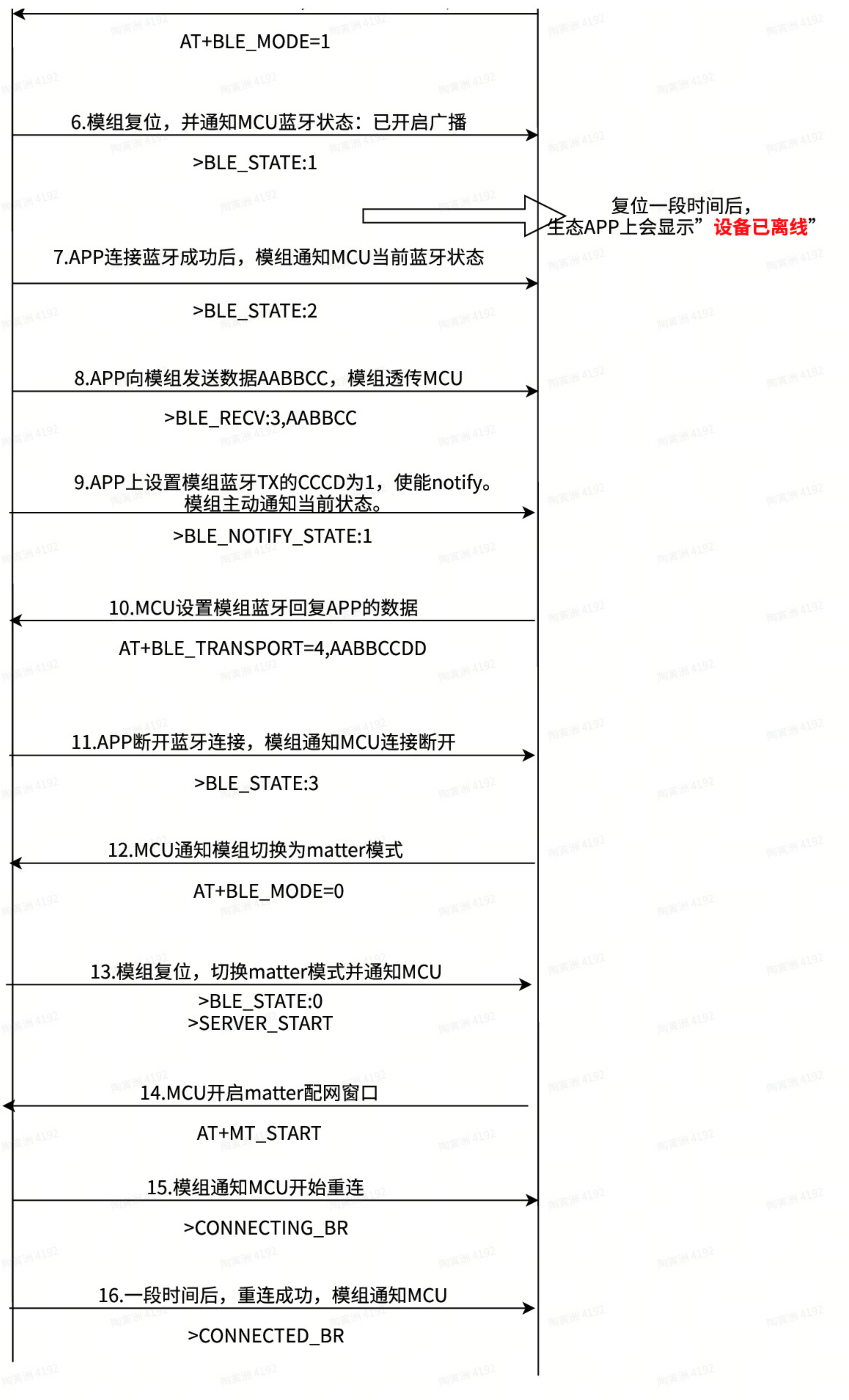
该命令用于查询蓝牙MAC地址

AT+BLE_MAC_ADDR 查询蓝牙MAC地址	
client	server
执行命令 AT+BLE_MAC_ADDR?	响应 +BLE_MAC_ADDR:OK,<addr>
	参数 <addr> 字符串类型，蓝牙mac地址
示例 AT+BLE_MAC_ADDR? +BLE_MAC_ADDR:OK,FEA74718F6AE	

14. 蓝牙AT流程示例

- 注：以下流程图的流程是先进行matter配网再进行蓝牙操作。





十、AT指令使用示例(下述指令以lift窗帘举例，仅供参考)

1. 开机，模块AT启动

将收到SERVER:START的通知

Server >SERVER:START

2. 开启matter，连接网关

2.1 开启matter，自动进入配网状态，使用配网工具进行配网

Client AT+MT_START

Server +MT_START:OK

>MT_EVENT:REQUIRE_CREDENTIAL

> MT_EVENT:COMMISSION_OPENING

> MT_EVENT:CONNECTED_BR

2.2 配网超时，需重新开启配网

Server >MT_EVENT:REQUIRE_CREDENTIAL

> MT_EVENT:COMMISSION_OPENING

> MT_EVENT:COMMISSION_CLOSED

Client AT+MT_RECMMSS

Server +MT_RECMMSS:OK

> MT_EVENT:COMMISSION_OPENING

>MT_EVENT: CONNECTED_BR

2.3 已经配网成功的设备，在下次上电时，开启matter将自动连接网关

Client AT+MT_START

Server +MT_START:OK

>MT_EVENT:CREDENTIAL_READY

> MT_EVENT:CONNECTING_BR

> MT_EVENT:CONNECTED_BR

2.4 设备已经配网，但是网关更换，新旧网关两者配网信息不同，则需要先清网再重新进行配网。清网完成后，设备会重启，需要重新开启matter。

Client AT+MT_FACTORYRESET

Server +MT_FACTORYRESET:OK

Server >SERVER:START

Client AT+MT_START

Server +MT_START:OK
>MT_EVENT:CREDENTIAL_READY
> MT_EVENT:CONNECTING_BR
> MT_EVENT:CONNECTED_BR

3. 配置属性

Client AT+MT_CURTAIN_TYPE_ATT=1,1

Server +MT_CURTAIN_TYPE_ATT:OK,1,1

Client AT+MT_CURTAIN_EPTYPE_ATT=1,1

Server +MT_CURTAIN_EPTYPE_ATT:OK,1,1

Client AT+MT_CURTAIN_MODE_ATT=1,1

Server +MT_CURTAIN_MODE_ATT:OK,1,1

Client AT+MT_CURTAIN_CPP_ATT=1,10000

Server +MT_CURTAIN_CPP_ATT:OK,1,10000

4. 读取属性

Client AT+MT_CURTAIN_CFGSTA_ATT=1?

Server +MT_CURTAIN_CFGSTA_ATT:OK,1,1

Client AT+MT_CURTAIN_OPSTA_ATT=1?

Server +MT_CURTAIN_OPSTA_ATT:OK,1,1

5. 接收命令

5.1 open命令

接收到open的命令，TPP属性值将被设置成0，此时模块发起主动通知。

Server >MT_CURTAIN_TPP_ATT:1,0

MCU接收到模块的通知后，开始转动电机移动窗帘，此时上报窗帘状态(正在打开)给模块，模块接收到AT指令后应答。

Client AT+MT_CURTAIN_OPSTA_ATT=1,5

Server +MT_CURTAIN_OPSTA_ATT:OK,1,5

上报一次CPP属性，避免影响用户体验，直接上报最终目标的位置

Client AT+MT_CURTAIN_CPP_ATT=1,0 //若开启CPP_AUTOMODE，则可以不用上报CPP

Server +MT_CURTAIN_CPP_ATT:OK,1,0

MCU控制窗帘移动到0%位置后停止移动，上报当前窗帘状态给模块，模块收到AT指令后应答。

Client AT+MT_CURTAIN_OPSTA_ATT=1,0

Server +MT_CURTAIN_OPSTA_ATT:OK,1,0

5.2 close命令

接收到close的命令，TPP属性值将被设置成10000，此时模块发起主动通知。

Server >MT_CURTAIN_TPP_ATT:1,10000

MCU接收到模块的通知后，开始转动电机移动窗帘，此时上报窗帘状态(正在关闭)给模块，模块接收到AT指令后应答。

Client AT+MT_CURTAIN_OPSTA_ATT=1,10

Server +MT_CURTAIN_OPSTA_ATT:OK,1,10

上报一次CPP属性，避免影响用户体验，直接上报最终目标的位置

Client AT+MT_CURTAIN_CPP_ATT=1,10000 //若开启CPP_AUTOMODE，则可以不用上报CPP

Server +MT_CURTAIN_CPP_ATT:OK,1,10000

MCU控制窗帘移动到100%位置后停止移动，上报当前窗帘状态给模块，模块收到AT指令后应答。

Client AT+MT_CURTAIN_OPSTA_ATT=1,0

Server +MT_CURTAIN_OPSTA_ATT:OK,1,0

5.3 got to lift percentage命令

接收到go to lift percentage的命令，TPP属性值将被设置成0~10000之间的任意数值，此时模块发起主动通知。

//例如go to lift percentage 600

Server >MT_CURTAIN_TPP_ATT:1,600

MCU接收到模块的通知后，开始转动电机移动窗帘，此时上报窗帘状态给模块，模块接收到AT指令后应答。窗帘状态可根据当前位置和目标位置进行判断：当前位置>目标位置为正在打开状态，当前位置<目标为正在关闭状态。

//例如CPP: 200, TPP: 600

Client AT+MT_CURTAIN_OPSTA_ATT=1,10

Server +MT_CURTAIN_OPSTA_ATT:OK,1,10

上报一次CPP属性，避免影响用户体验，直接上报最终目标的位置

Client AT+MT_CURTAIN_CPP_ATT=1,600 //若开启CPP_AUTOMODE，则可以不用上报CPP

Server +MT_CURTAIN_CPP_ATT:OK,1,600

MCU控制窗帘移动到指定位置后停止移动，上报当前窗帘状态给模块，模块收到AT指令后应答。

Client AT+MT_CURTAIN_OPSTA_ATT=1,0

Server +MT_CURTAIN_OPSTA_ATT:OK,1,0

5.4 stop motion命令

接收到stop motion的命令，此时模块发起主动通知。client端接收到命令后，电机停止运动并需将TPP和CPP属性值上报给server端，同步这两个属性值。

Server >MT_CURTAIN_STOP_CMD:1

MCU接收到模块的通知后，停止电机转动，此时同步目标位置属性、上报当前位置和窗帘状态给模块，模块接收到AT指令后应答。

//例如此时停止时的位置为600

Client AT+MT_CURTAIN_CTPP_UPDATE=1,600

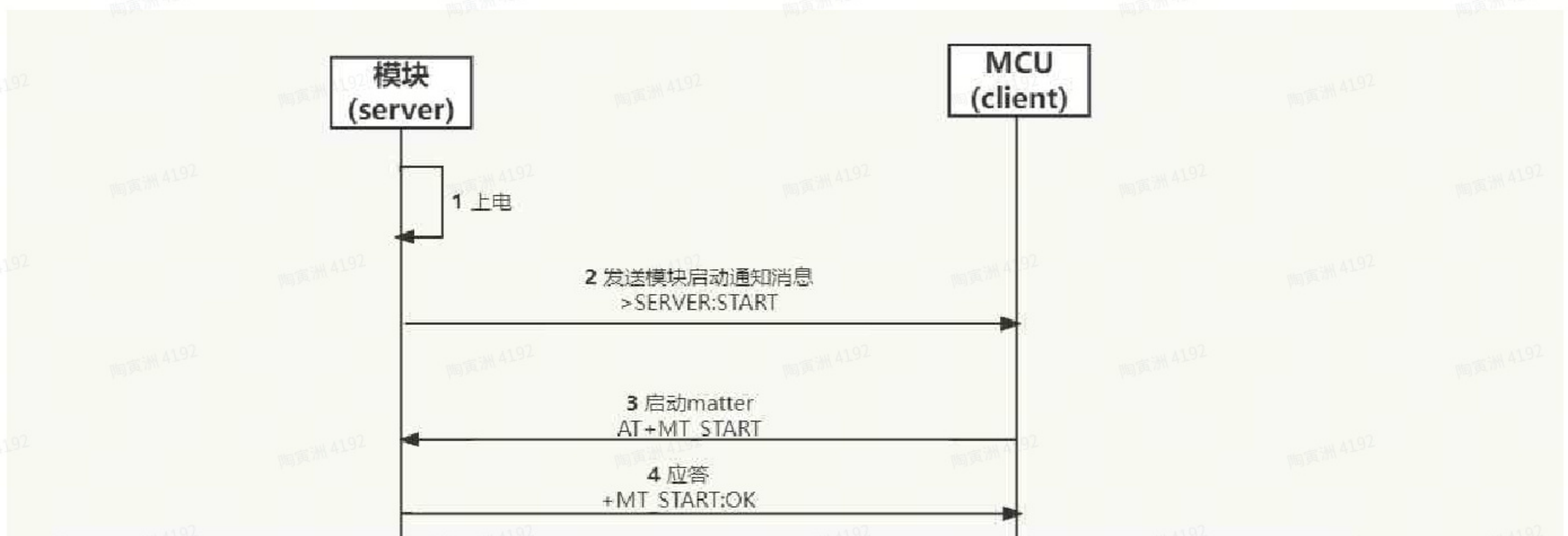
Server +MT_CURTAIN_CTPP_UPDATE:OK,1,600

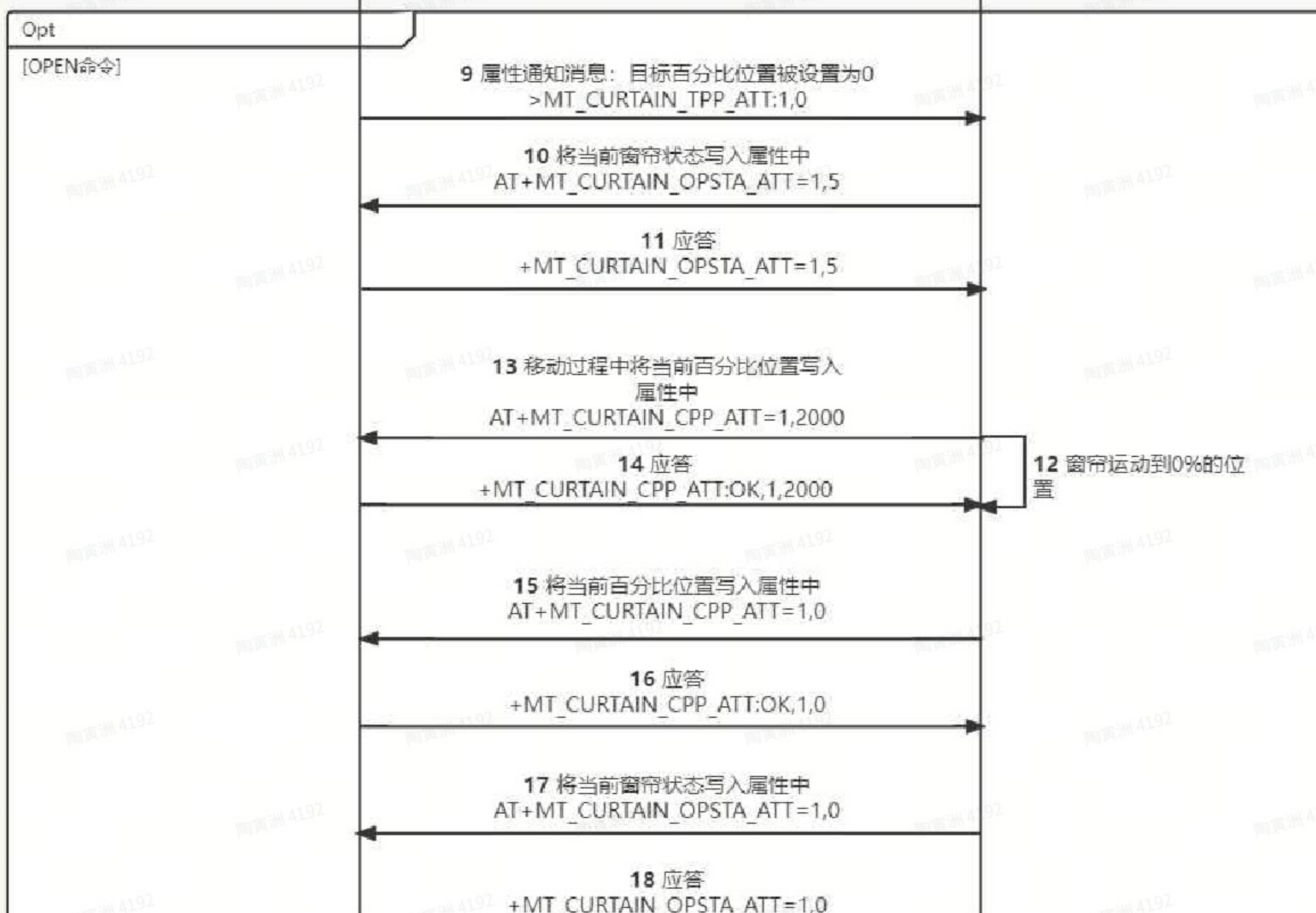
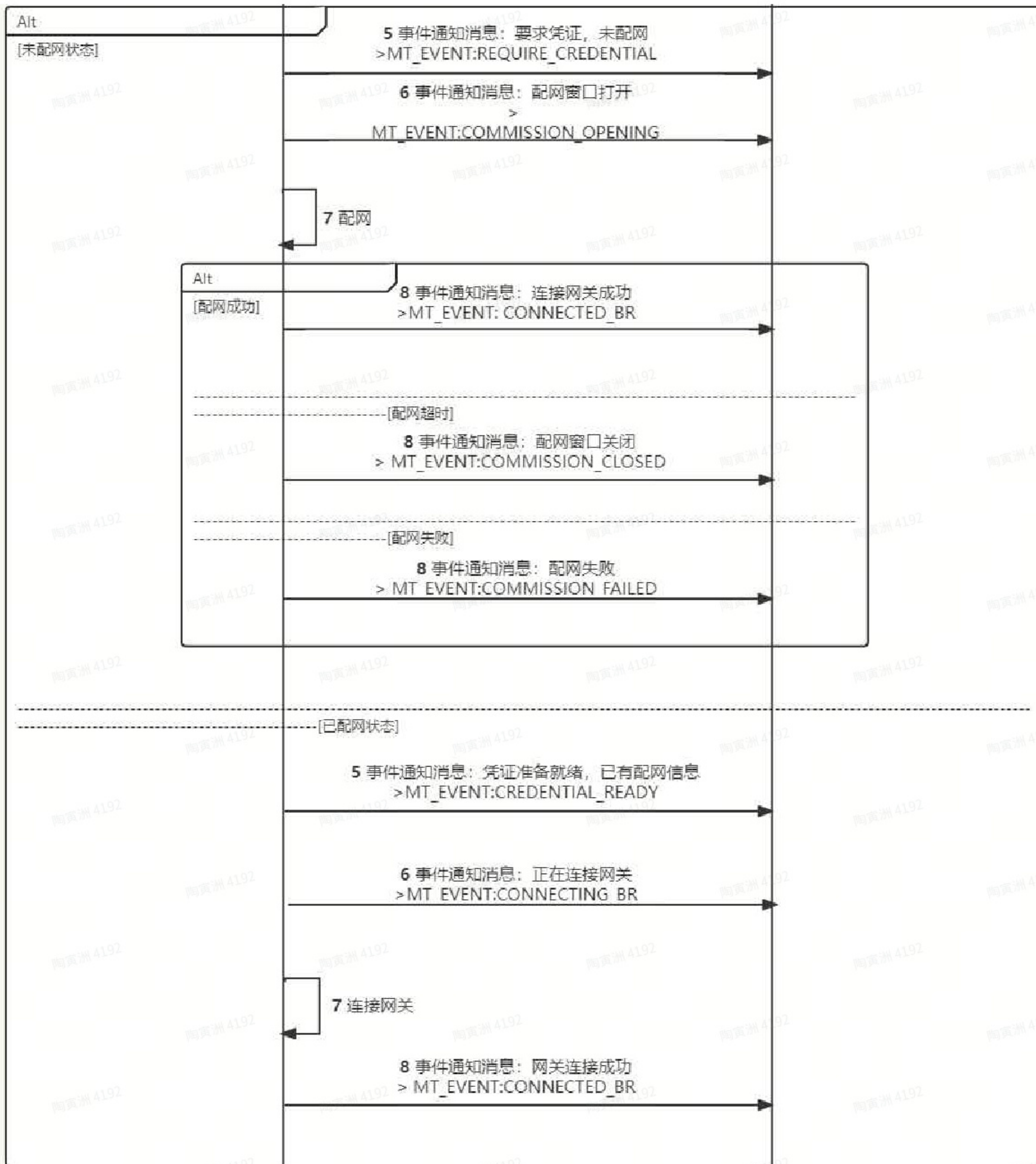
Client AT+MT_CURTAIN_OPSTA_ATT=1,0

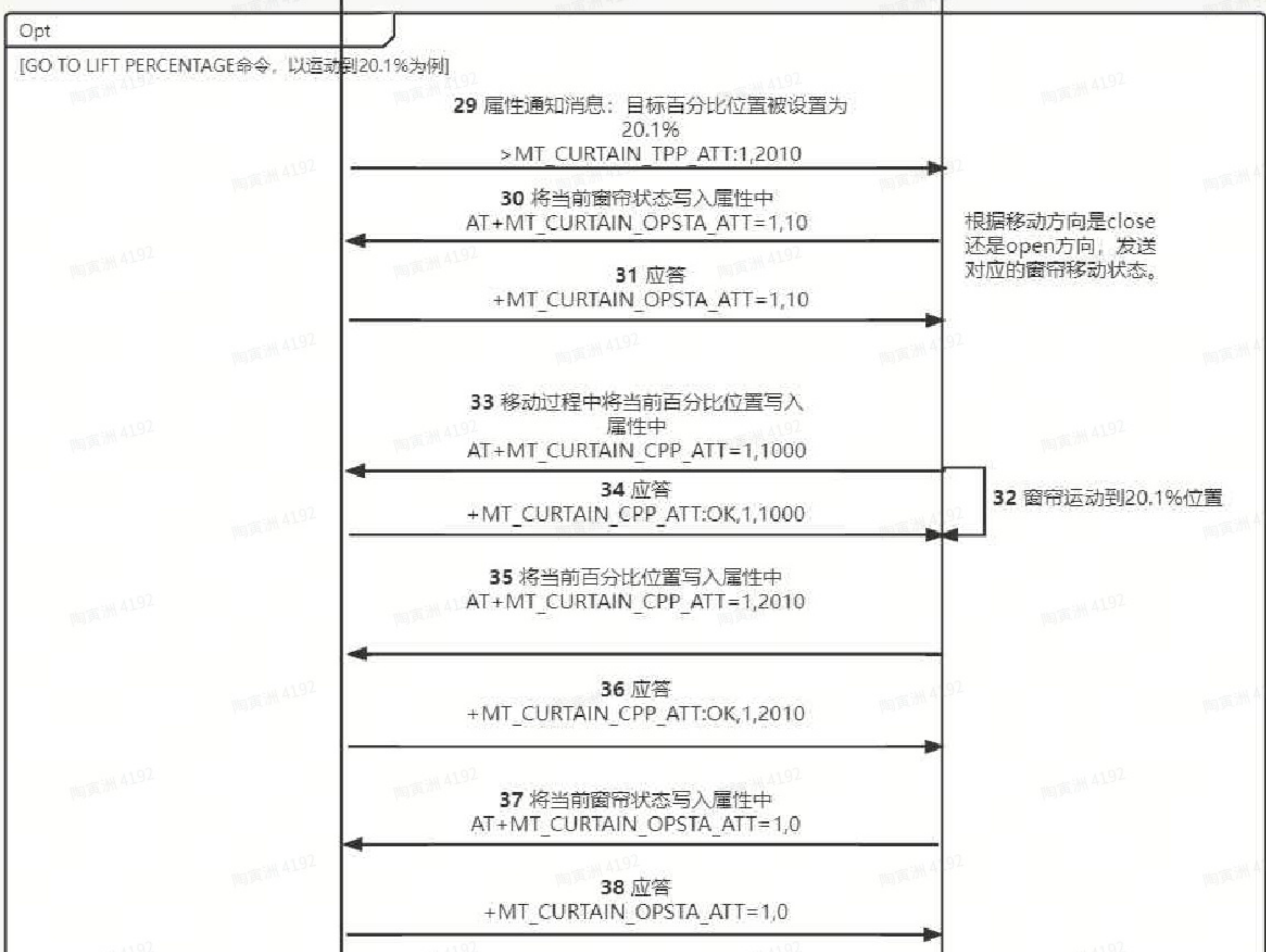
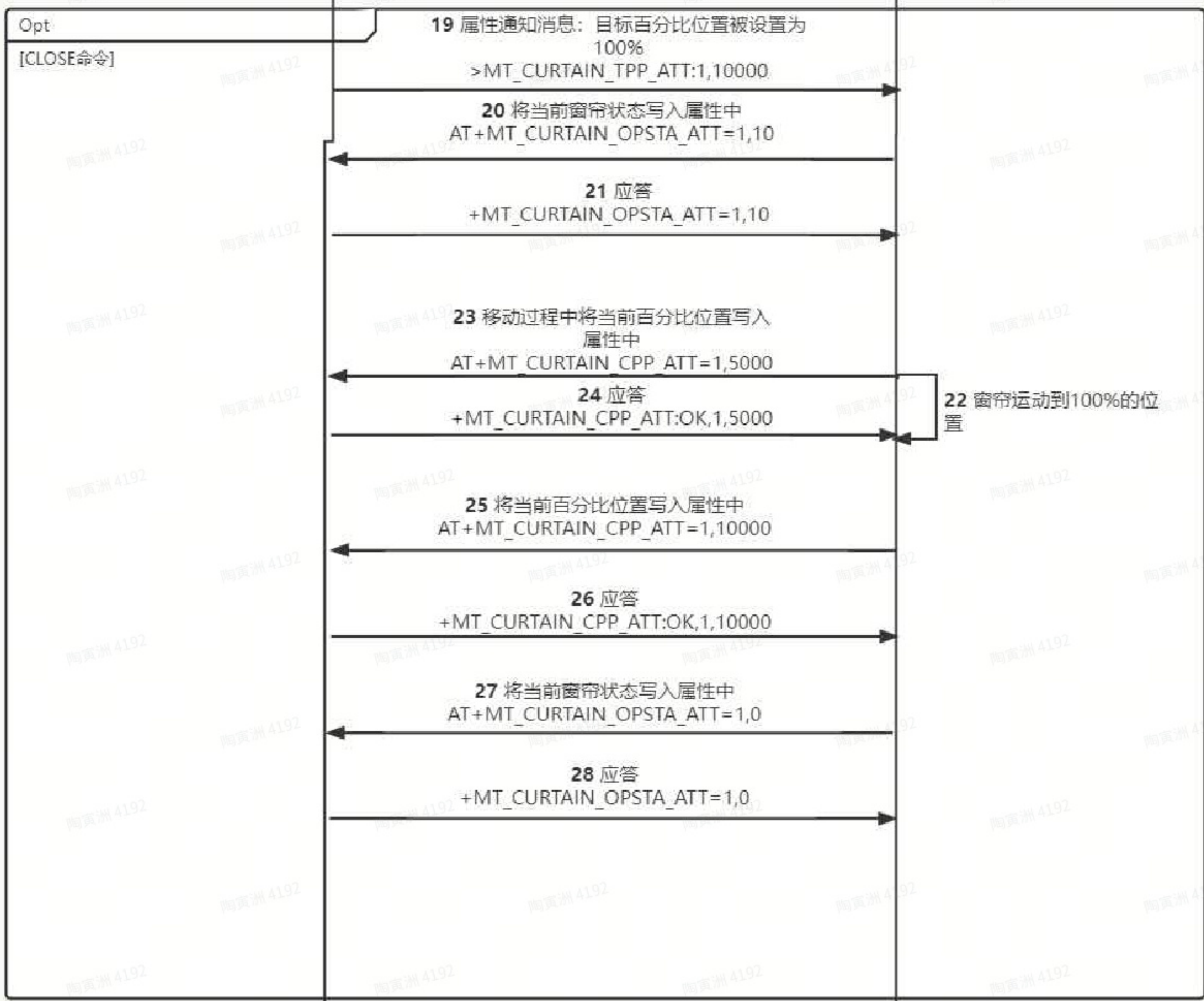
Server +MT_CURTAIN_OPSTA_ATT:OK,1,0

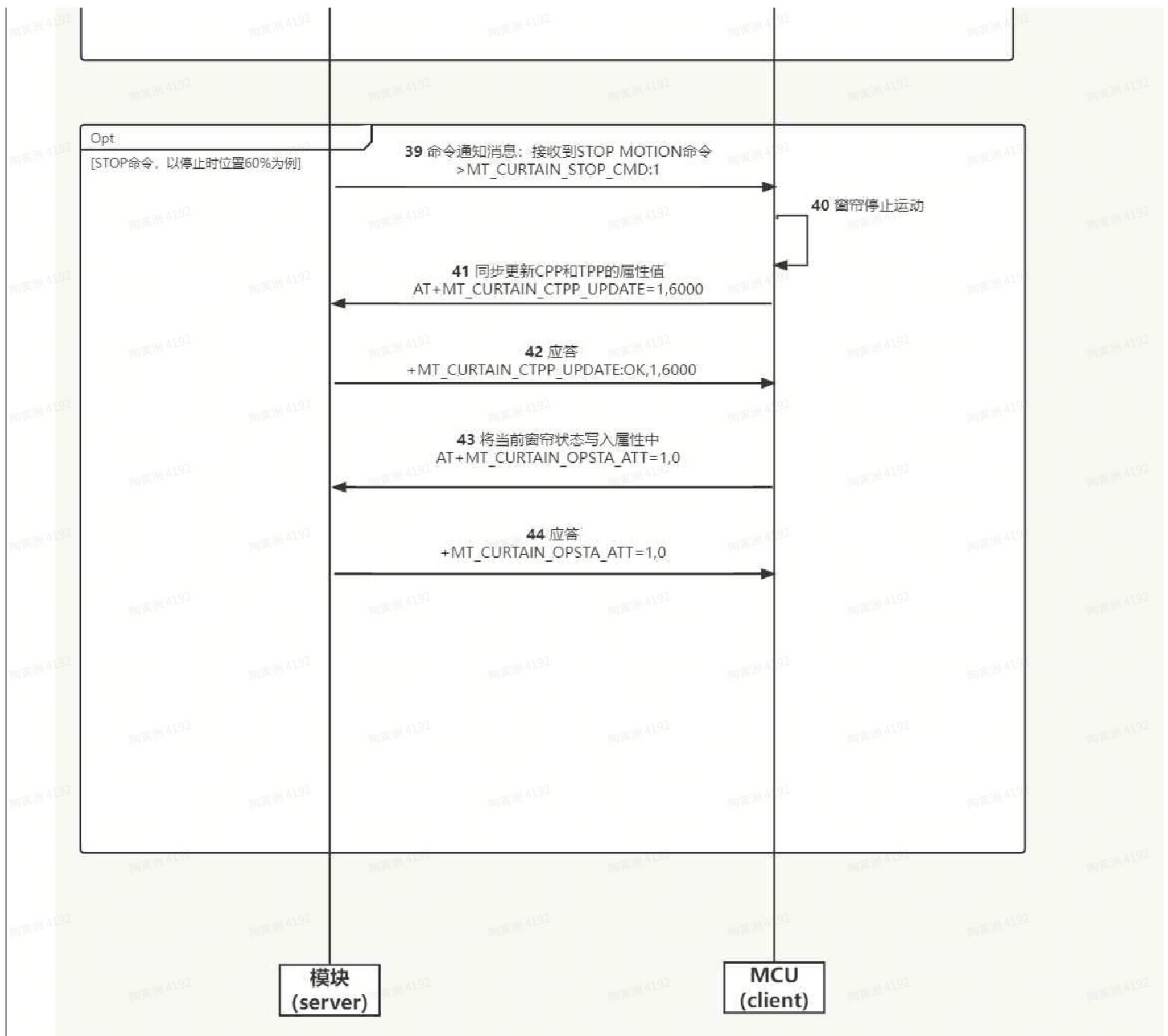
十一、AT通信交互流程

下图为配网以及窗帘的开关等主要功能的流程供参考。在实际使用时，需要在模块启动完成后，先配置和同步相关属性值，如窗帘类型、终端产品类型、配置项、限位、当前位置、电源信息相关属性等。







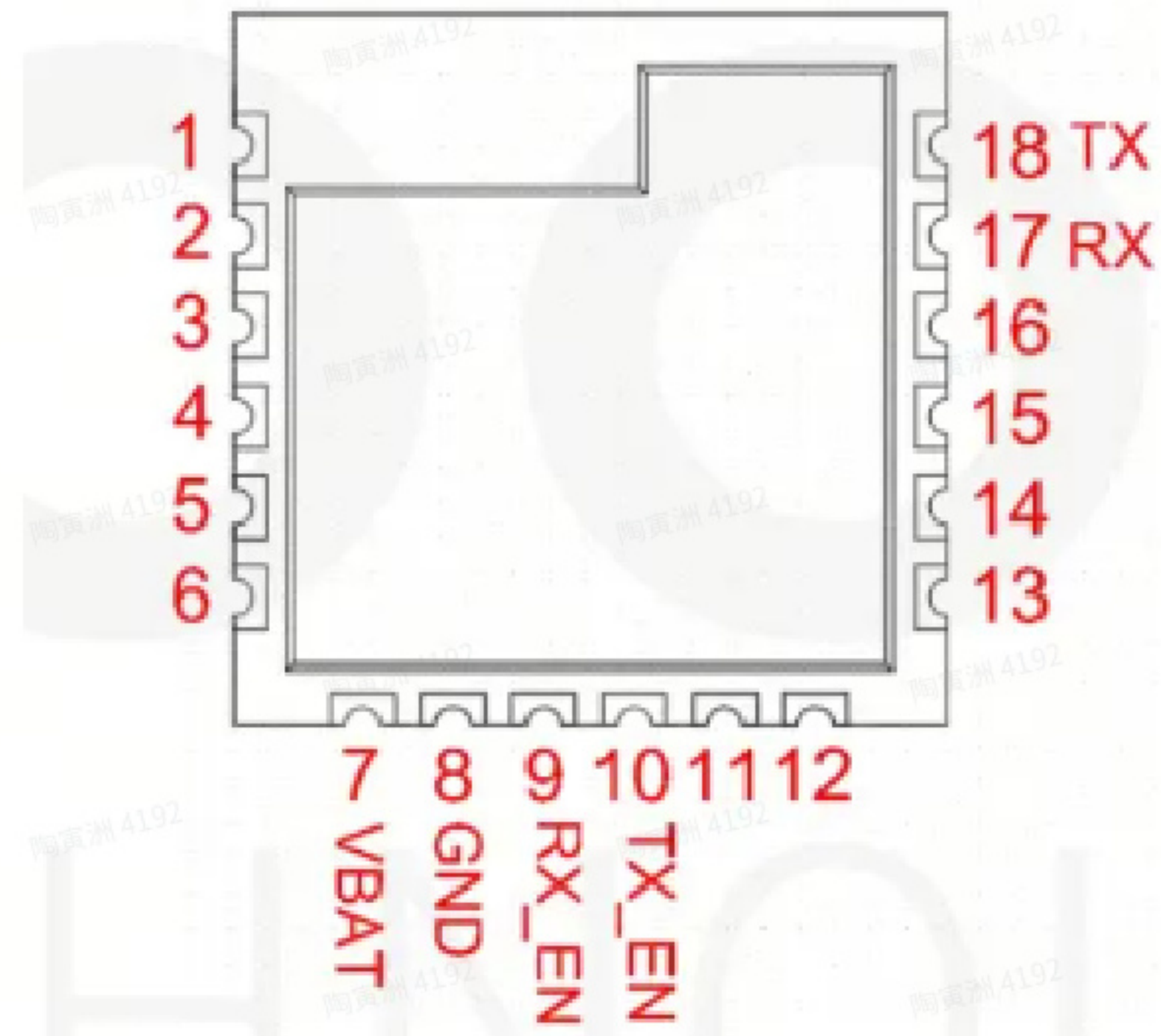


十二、注意事项

1. 在模块执行AT+MT_FACTORYRESET, 等待模块启动完成后, 需要重新配置和同步相关属性 (如 AT+MT_CURTAIN_CFGSTA_ATT配置项属性、AT+MT_CURTAIN_CPP_ATT当前位置属性)。
2. 在收到TPP命令时, 即下发的控制命令, 窗帘电机移动, 此时需要上报一次CPP的值, 我们建议此时上报的CPP值与TPP值相同, 因为在部分APP上, 实时上报位置会导致APP界面上拖动条跳动, 影响用户体验。但也不能不报, 不上报会影响设备的认证过测。
3. 当设备移动达不到0.01%精度时, 仅需移动到最接近目标位置的地方, 允许有物理误差, 但上报的当前位置值需与目标位置值相同。

十三、窗帘模组AT引脚图

HRN71:



HAT31:

