

BLE_MESH_AT_MODULE

1. 概述

1.1 目的

1.2 名词解释

1.3 介绍

2. AT 指令语法

2.1. AT请求消息格式

2.2 AT响应消息格式

2.3 AT事件消息格式

3 AT指令总览

3.1 AT控制指令

3.2 AT事件

4 AT指令说明

4.1 基础AT命令

4.1.1 AT

4.1.2 AT+HELP

4.1.3 AT+REBOOT

4.1.4 AT+RST

4.1.5 AT+MESHAT

4.1.6 AT+MESHQUERYSTA

4.2 Provisioner AT命令

4.2.1 AT+MESHPROVCONFIG

4.2.2 AT+MESHPROVAUTOPROV

4.2.3 AT+MESHPROVEN

4.2.4 AT+MESHPROVFILTERDEV

4.2.5 AT+MESHPROVMACFILTER

4.2.6 AT+MESHPROVMACFILTERADD

4.2.7 AT+MESHPROVMACFILTERRM

4.2.8 AT+MESHPROVMACFILTERCLR

4.2.9 AT+MESHPROVSHOWDEV

4.2.10 AT+MESHPROVSETNODELPM

4.2.11 AT+MESHGETNODEINFO

4.2.12 AT+MESHADDDEV

4.2.13 AT+MESHAUTOCONFIG

4.2.14 AT+MESHDELDEV

4.2.15 AT+MESHOOB

4.2.16 AT+MESHGETNETKEY

4.2.17 AT+MESHSETNETKEY

4.2.18 AT+MESHGETAPPKEY

4.2.19 AT+MESHSETAPPKEY

4.2.20 AT+MESHCLEARRPL

4.3 Config Client model AT命令

4.3.1 AT+MESHADDAPPKEY

4.3.2 AT+MESHBINDAPPKEY

4.3.3 AT+MESHUNBINDAPPKEY

4.3.4 AT+MESHGETCOMP

4.3.5 AT+MESHRELAY

4.3.6 AT+MESHPROXY

4.3.7 AT+MESHFRIEND

4.3.8 AT+MESHGETSUB

4.3.9 AT+MESHSETSUB

4.3.10 AT+MESHDELSUB

4.3.11 AT+MESHSUBLISTOVERWRITE

4.3.12 AT+MESHGETPUB

4.3.13 AT+MESHSETPUB

4.3.14 AT+MESHRST

4.4 Generic OnOff Client model AT命令

4.4.1 AT+MESHONOFF

4.5 Generic Level Client model AT命令

4.5.1 AT+MESHLEVEL

4.5.2 AT+MESHLEVELMOVE

4.5.3 AT+MESHLEVELDELTA

4.6 Light Lightness Client model AT命令

4.6.1 AT+MESHLIGHTNESSRANGE

4.6.2 AT+MESHLIGHTNESSDEF

4.6.3 AT+MESHLIGHTNESS

4.6.4 AT+MESHLIGHTNESSLIN

4.7 Light CTL Client model AT 命令

4.7.1 AT+MESHCTLTEMPRANGE

4.7.2 AT+MESHCTLDEF

4.7.3 AT+MESHCTL

4.7.4 AT+MESHCTLTEMP

4.8 Vendor Client model AT命令

4.8.1 AT+MESHTRS

5 AT事件

5.1 provisioner扫描设备信息上报事件

5.2 provisioner扫描设备超时事件

5.3 Provisioner设备入网失败事件

5.4 Provisioner设备入网成功事件

5.5 OOB输入提示事件

5.6 node节点信息事件

5.7 Mesh Model 上报事件

5.7.1 config model status

5.7.1.1 appkey add status

5.7.1.2 appkey bind status

5.7.1.3 friend status

5.7.1.4 proxy status

5.7.1.5 relay status

5.7.1.6 RST status

5.7.1.7 composition data status

5.7.1.8 mesh sub list status

5.7.1.9 mesh sub set status

5.7.1.10 mesh sub overwrite status

5.7.1.11 mesh pub status

5.7.1.12 mesh pub set status

5.7.2 onoff model status

5.7.3 level model status

5.7.4 light lightness model status

5.7.4.1 lightness status

5.7.4.2 lightness linear status

5.7.4.3 lightness range status

5.7.4.4 lightness last status

5.7.5 light ctl model status

5.7.5.1 light ctl status

5.7.5.2 light temperature status

5.7.5.3 ctl temperature range status

5.7.5.4 ctl defalut status

5.7.6 vendor model status

5.7.6.1 透传数据

5.7.6.2 autoconfig status

6 AT 使用指南

6.1 准备工作

6.2 获取未入网设备

6.3 设备过滤

6.4 自动配网

6.4.1 单播地址自动配网

6.4.2 组播地址自动配网

6.5 手动配网

6.6 AT App key切换

6.7 SUB/PUB地址配置

6.7.1 provisioner pub /node sub

6.7.2 provisioner sub /node pub

7. 附录

7.1 Mesh Sig Model ID查询表

7.2 Mesh venor Model ID查询表

文档版本	v1.2
发布日期	2021-07-08



1. 概述

1.1 目的

本文主要介绍Mesh配网模块支持的AT指令集，为有Mesh配网需求的开发人员提供帮助。开发者可以使用配网模块发现，添加未入网设备，对入网设备进行配置，管理。

1.2 名词解释

下面介绍本文中涉及的一些专有名词

名词	说明
BLE	bluetooth Low energy, 低功耗蓝牙
BLE Mesh	基于BLE技术实现的多对多通信网络
MESH配网模块	基于AT命令实现的Provisioner
Provision	将一个未入网的设备加入ble mesh网络的过程，期间会通过认证和加密的方式将密钥和地址分配给设备
Address	Mesh节点的地址
RPL	replay protection list, 重放攻击保护列表。
OOB	Out of band, 带外数据，用于Provision过程中鉴权。
node	已入网设备
Composition Data	node内部的可用信息
CTL	Color-tunable light, 色温可调灯，支持色温和亮度控制

1.3 介绍

本文将会涉及到的如下几部分按章节介绍：

1、AT命令语法

2、AT命令

AT命令将分为如下几个类别

- 基础AT命令
包括通信检查，帮助，复位等功能的AT命令
- Provisioner AT命令
配网相关功能AT命令，包括发现，添加，配置设备等功能
- Mesh Client models AT命令
包括Config Client model, Generic OnOff Client model, Generic Level Client model, Light
Lightness Client model, Light CTL Client model, Vendor Client model的AT命令

3、AT事件

4、AT使用指南

在6.0章节，以灯控为例，介绍了Mesh网络的构建和配置。

2. AT 指令语法

AT命令基于ASCII码编码，以回车符<\r>作为结束符。

2.1. AT请求消息格式

AT请求消息格式：AT+[OP][para-1,para-2,.....para-n]<\r>

AT 请求消息格式

域	说明
AT+	命令消息前缀
CMD	命令消息前缀
OP	指令操作符。可以是以下内容：“=”：表示写指令，用于参数设置。“?”：表示读指令，用于查询参数的当前值。“”：表示执行指令。“=?”：表示测试指令，查询指令使用方法。
para-1,para-2,.....	表示设置的参数值，或者是指定要查询的参数
\r	回车结束符，ASCII码为0x0D

2.2 AT响应消息格式

AT响应消息为针对AT命令的响应。

AT响应消息格式：<\r\n>[+CMD:][para-1,para-2,.....para-n]<\r\n>

或者：<\r\n><\r\n>

或者上述两者都有。

AT响应消息格式

域	说明
+	响应消息前缀
CMD	响应消息前缀
para-1,para-2,.....	表示设置的参数值，或者是指定要查询的参数
\r\n	响应消息结束符，ASCII码为0x0D0A
STATUS	OK或者ERROR，表示成功或者失败

2.3 AT事件消息格式

AT事件是设备主动上报的消息

AT事件格式：<\r\n>[+CMD:][para-1,para-2,.....para-n]<\r\n>

域	说明
+	事件前缀
CMD	事件前缀
para-1,para-2,.....	表示设置的参数值，或者是指定要查询的参数
\r\n	响应消息结束符，ASCII码为0x0D0A

3 AT指令总览

3.1 AT控制指令

命令	说明
AT	测试命令
AT+HELP	帮助命令
AT+REBOOT	设备重启
AT+RST	设备恢复出厂设置
AT+MESHEN	使能MESH AT命令
AT+MESHQUERYSTA	查询Mesh model status
AT+MESHPROVCONFIG	Provisioner配置
AT+MESHPROVAUTOPROV	Provisioner自组网
AT+MESHPROVEN	Provisioner使能

AT+MESHCLRARRPL	清空RPL
AT+MESHPROVMACFILTER	Provisioner使能MAC过滤
AT+MESHPROVMACFILTERADD	Provisioner MAC过滤添加设备列表
AT+MESHPROVMACFILTERRMCLR	Provisioner MAC过滤删除设备列表
AT+MESHPROVMACFILTER	Provisioner MAC过滤清空设备列表
AT+MESHPROVFILTERDEV	Provisioner上报设备过滤
AT+MESHPROVSHOWDEV	Provisioner上报查找到的未入网设备
AT+MESHPROVSETNODELPM	Provisioner设置低功耗节点
AT+MESHADDDEV	Provisioner添加未入网设备
AT+MESHDELDEV	Provisioner删除未入网设备
AT+MESHOOB	Provisioner添加OOB信息
AT+MESHAUTOCONFIG	Provisioner对node节点进行自动配置
AT+MESHGETNODEINFO	Provisioner查询node信息
AT+MESHGETCOMP	获取node节点组成信息
AT+MESHADDAPPKEY	添加node app key
AT+MESHBINDAPPKEY	绑定node model和app key
AT+MESHUNBINDAPPKEY	解绑node model和app key
AT+MESHRELAY	配置node relay 属性
AT+MESHPROXY	配置node proxy属性
AT+MESHFRINED	配置node friend属性
AT+MESHGETSUB	获取node model订阅地址
AT+MESHSETSUB	设置node model订阅地址
AT+MESHDELSUB	删除node model订阅地址
AT+MESHSUBLISTOVERWRITE	覆盖node model订阅地址
AT+MESHGETPUB	获取node model发布属性
AT+MESHSETPUB	设置node model发布属性

命令	说明
AT+MESHIRST	清除node配网信息
AT+MESHONOFF	配置node onoff状态
AT+MESHLEVEL	配置node generic level
AT+MESHLEVELMOVE	配置node generic level move
AT+MESHLEVELDELTA	配置node generic level delta
AT+MESHLIGHTNESSRANGE	配置 node lightness range
AT+MESHLIGHTNESSDEF	配置node lightness default
AT+MESHLIGHTNESS	配置node lightness
AT+ MESHLIGHTNESSLIN	配置node lightness linear
AT+MESHCTLTEMPRANGE	配置node CTL temperature range
AT+MESHCTLDEF	配置 node CTL default
AT+MESHCTL	配置 node CTL
AT+MESHCTLTEMP	配置node CTL temperature
AT+MESHTRS	发送透传数据

3.2 AT事件

事件	说明
+MESHDEV	未入网设备信息上报
+MESHFOUNDDEVTIMEOUT	查找设备超时
+MESHPROVFAILD	Provisioner添加设备失败
+MESHNODEADD	Provisioner添加设备成功
+MESHNODEINFO	node节点信息
+MESHOOBINPUT	Provisioner输入oob信息
+MESHAPPKEYADD	node添加appkey状态
+MESHAPPKEYBIND	node model绑定appkey状态
+MESHFRIEND	node friend属性
+MESHPROXY	node proxy属性
+MESHRELAY	node relay属性

事件	说明
+MESHRST	节点RST信息
+MESHCOMP	节点Composition Data信息
+MESHSUBLIST	node model sub信息
+MESHSUBSET	node model sub 设置状态
+MESHSUBLISTOVERWRITE	node model sub 覆盖状态
+MESH PUB	node model pub信息
+MESH PUBSET	node model pub设置状态
+MESH ONOFF	onff model 状态
+MESH LEVEL	Level model 状态
+MESH LIGHTNESS	Lightness model状态
+MESH LIGHTNESS LIN	Lightness model linear状态
+MESH LIGHTNESS RANGE	Lightness model range状态
+MESH LIGHTNESS LAST	Lightness model last状态
+MESH LIGHT CTL	LIGHT CTL model状态
+MESH CTL TEMP	LIGHT CTL model temperature状态
+MESH CTL TEMP RANGE	LIGHT CTL model range状态
+MESH CTL DEF	LIGHT CTL model default状态
+MESH AUTO CONFIG	node节点自动配置状态

4 AT指令说明

4.1 基础AT命令

主要完成AT状态测试、设备重启、AT查询、MESH model 状态查询等功能

4.1.1 AT

基本功能：测试AT模块状态

指令类型	格式	响应
测试指令	无	无
查询指令	无	无
写指令	无	无
执行指令	AT	OK
参数说明	无	
返回值说明	无	
示例	AT OK	
注意事项	无	

4.1.2 AT+HELP

基本功能：测试AT支持地命令

指令类型	格式	响应
测试指令	无	无
查询指令	无	无
写指令	无	无
执行指令	AT	OK
参数说明	无	
返回值说明	无	
示例	AT+HELP +HELP:HELP; IREBOOT; MESHQUERYSTA; MESHPROVEN; MESHPROVFILTERDEV; MESHPROVSHOWDEV; MESHPROVSETNODELPM;	

示例	MESHGETNODEINFO; MESHADDDEV; MESHAUTOCONFIG; MESHDELDEV; MESHOOB; MESHSETAPPKEY; MESHADDAPPKEY; MESHBINDAPPKEY;	
----	--	--

	MESHUNBINDAPPKEY; MESHGETCOMP; MESHRELAY; MESHPROXY; MESHFRIEND; MESHGETSUB; MESHSETSUB; MESHDELSUB; MESHSUBLISTOVERWRITE; MESHGETPUB; MESHSETPUB; MESHRST; MESHONOFF; MESHLEVEL; MESHLEVELMOVE; MESHLEVELDELTA; MESHLIGHTNESSRANGE; MESHLIGHTNESSDEF; MESHLIGHTNESS; MESHLIGHTNESSLIN; MESHCTLTEMPRANGE; MESHLIGHTCTLDEF; MESHLIGHTCTL; MESHCTLTEMP; MESHTRS; OK
注意事项	无

4.1.3 AT+REBOOT

基本功能：重启设备

指令类型	格式	响应
测试指令	无	无
查询指令	无	无
写指令	无	无
执行指令	AT	OK
参数说明	无	
返回值说明	无	
示例	AT OK	
注意事项	重启后，会上报事件，见AT事件说明	

4.1.4 AT+RST

基本功能：设备恢复出厂设置

指令类型	格式	响应
测试指令	无	无
查询指令	无	无
写指令	无	无
执行指令	AT+RST	OK
参数说明	无	
返回值说明	无	
示例	AT+RST OK	
注意事项	设备恢复出厂设置后，设备自动重启	

4.1.5 AT+MESHAT

基本功能：mesh AT命令使能

指令类型	格式	响应
测试指令	AT+MESHAT=?	+MESHAT= < onoff> OK
查询指令	AT+MESHAT?	+MESHAT: < onoff>
写指令	AT+MESHAT=< onoff>	OK 或者 +MESHAT:< ERR> ERROR
执行指令	无	无
参数说明	onoff – 0(关闭)/1(使能)	
返回值说明	无	
示例	AT+MESHAT=0x01 OK	
注意事项	想要使用MESH相关AT命令，必须使能，掉电不保存	

4.1.6 AT+MESHQUERYSTA

基本功能：查询model的status状态

指令类型	格式	响应
测试指令	AT+MESHQUERYSTA=?	+MESHQUERYSTA= < unicast_addr>,< query_mod_id>,< query_opcode>,[netkey_idx], [appkey_idx],[cid] OK
查询指令	无	无
写指令	AT+MESHQUERYSTA=< unicast_addr>,< query_mod_id>,< query_opcode>,[netkey_idx], [appkey_idx]	OK 或者 +MESHQUERYSTA:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 查询目标地址 query_mod_id – 查询所使用的model id query_opcode – 查询所使用的opcode netkey_idx – 发送使用的netkey id(默认为0) appkey_idx – 发送使用的appkey id(默认为0) cid –vendor model id (默认无)	
返回值说明	无	
示例	AT+MESHQUERYSTA=0x0002,0x1001,0x8201 OK +MESHONOFF=0002,01	
注意事项	使用前需使能PROVISIONER 常见的状态查询对应的model id 及opcode见附录model状态查询表	

4.2 Provisioner AT命令

4.2.1 AT+MESHPROVCONFIG

基本功能：Provisioner配置

指令类型	格式	响应
测试指令	AT+MESHPROVCONFIG=?	+MESHPROVCONFIG= < start_addr>,< end_addr> OK
查询指令	无	无
写指令	AT+MESHPROVCONFIG=< start_addr>,< end_addr>	OK 或者 +MESHPROVCONFIG:< ERR> ERROR
执行指令	无	无
参数说明	start_addr – 分配给网关的起始地址 end_addr – 分配给网关的结束地址	
返回值说明	无	
示例	AT+MESHPROVCONFIG=0x0001,0x3fff OK	
注意事项	start_addr < end_addr end_addr – start_addr > element_num(element_num为网关自身的元素个数) 该命令掉电保存	

4.2.2 AT+MESHPROVAUTOPROV

基本功能：Provisioner自组网

指令类型	格式	响应
测试指令	无	无
查询指令	无	无
写指令	无	无
执行指令	AT+MESHPROVAUTOPROV	OK 或者 +MESHPROVAUTOPROV: < ERR> ERROR
参数说明	无	
返回值说明	无	
示例	AT+MESHPROVAUTOPROV OK	
注意事项	网关若还未组网可使用该命令进行入网 该命令掉电保存	

4.2.3 AT+MESHPROVEN

基本功能：Provisioner使能

指令类型	格式	响应
测试指令	AT+MESHPROVEN=?	+MESHPROVEN= < onoff> OK
查询指令	AT+MESHPROVEN?	+MESHPROVEN: < onoff> 或者 +MESHPROVEN:< ERR> < ERROR>
写指令	AT+MESHPROVEN=< onoff>	OK 或者 +MESHPROVEN:< ERR> ERROR
执行指令	无	无
参数说明	onoff – 0(关闭)/1(使能)	
返回值说明	无	
示例	AT+MESHPROVEN=0x01 OK	
注意事项	该命令掉电保存	

4.2.4 AT+MESHPROVFILTERDEV

基本功能：Provisioner设置上报过滤

指令类型	格式	响应
测试指令	AT+MESHPROVFILTERDEV=?	+MESHPROVFILTERDEV= < onoff>,[uuid_str],[filter_start] OK
查询指令	无	无
写指令	AT+MESHPROVFILTERDEV= < onoff>,[uuid_str], [filter_start]	OK 或者 +MESHPROVFILTERDEV:< ERR> ERROR
执行指令	无	无
参数说明	onoff – 0(关闭)/1(使能) uuid_str – 待匹配的uuid段 filter_start – 开始匹配的位置	
返回值说明	无	
示例	AT+MESHPROVFILTERDEV=0x01,06050403,00 OK	
注意事项	设置UUID过滤后，会从查找到的设备UUID filter_start 位置处开始匹配uuid_str，若匹配成功，则上报该设备	

4.2.5 AT+MESHPROVMACFILTER

基本功能：Provisioner MAC 过滤开关

指令类型	格式	响应
测试指令	AT+MESHPROVMACFILTER= ?	+MESHPROVMACFILTER= < onoff> OK
查询指令	无	无
写指令	AT+MESHPROVMACFILTER= < onoff>	OK 或者 +MESHPROVMACFILTER:< ERR> ERROR
执行指令	无	无
参数说明	onoff – 0(关闭)/1(使能)	
返回值说明	无	
示例	AT+MESHPROVMACFILTER=0x01 OK	
注意事项	使能MAC过滤后，需添加MAC过滤列表才可生效	

4.2.6 AT+MESHPROVMACFILTERADD

基本功能：Provisioner MAC 过滤添加过滤设备

指令类型	格式	响应
测试指令	AT+MESHPROVMACFILTERADD?	+MESHPROVMACFILTERADD=< size> < addr>,< addr_type>,... OK
查询指令	无	无
写指令	AT+MESHPROVMACFILTERADD=< size> < addr>,< addr_type>,...	OK 或者 +MESHPROVMACFILTERADD:< ERR> ERROR
执行指令	无	无
参数说明	size – 过滤列表大小 addr – 设备MAC地址 addr_type – 设备MAC地址类型,0(public)/1(random)	
返回值说明	无	
示例	AT+MESHPROVMACFILTERADD=0x01,C1:01:02:03:04:05,0 OK	
注意事项	使能MAC过滤后并添加添加MAC过滤列表后，配网器只上报过滤列表中的设备 地址类型只支持一位数据（0/1）	

4.2.7 AT+MESHPROVMACFILTERRM

基本功能：Provisioner MAC 过滤删除过滤设备

指令类型	格式	响应
测试指令	AT+MESHPROVMACFILTERRM? M?	+MESHPROVMACFILTERRM= < size> < addr>,< addr_type>,... OK
查询指令	无	无
写指令	AT+MESHPROVMACFILTERRM= M=< size> < addr>,< addr_type>,...	OK 或者 +MESHPROVMACFILTERRM: < ERR> ERROR
执行指令	无	无
参数说明	size – 过滤列表大小 addr – 设备MAC地址 addr_type – 设备MAC地址类型,0(public)/1(random)	
返回值说明	无	
示例	AT+MESHPROVMACFILTERRM=0x01,C1:01:02:03:04:05,0 OK	
注意事项	无	

4.2.8 AT+MESHPROVMACFILTERCLR

基本功能：Provisioner MAC 过滤列表清除

指令类型	格式	响应
测试指令	无	无
查询指令	无	无
写指令	无	无
执行指令	AT+MESHPROVMACFILTERCLR	OK 或者 +MESHPROVMACFILTERCLR :< ERR> ERROR
参数说明	无	
返回值说明	无	
示例	AT+MESHPROVMACFILTERCLR OK	
注意事项	无	

4.2.9 AT+MESHPROVSHOWDEV

基本功能：Provisioner设备上报功能开启

指令类型	格式	响应
测试指令	AT+MESHPROVSHOWDEV=?	+MESHPROVSHOWDEV= <onoff>,[timeout] OK
查询指令	无	无
写指令	AT+MESHPROVSHOWDEV=<onoff>,[timeout]	OK 或者 +MESHPROVSHOWDEV:<ERR> ERROR
执行指令	无	无
参数说明	onoff – 0(关闭)/1(使能) timeout –查找设备超时(单位s, 默认无超时)	
返回值说明	无	
示例	AT+MESHPROVSHOWDEV=0x01 OK	
注意事项	查找超时后上报事件+MESHFOUNDDEVTIMEOUT	

4.2.10 AT+MESHPROVSETNODELPM

基本功能：Provisioner设备标识节点是否为低功耗节点

指令类型	格式	响应
测试指令	AT+MESHPROVSETNODELPM M=?	+MESHPROVSETNODELPM= <unicast_addr>,<flag> OK
查询指令	无	无
写指令	AT+MESHPROVSETNODELPM M=<unicast_addr>,<flag>	OK 或者 +MESHPROVSETNODELPM:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 节点首要element单播地址 flag– 0(非低功耗设备)/1(低功耗设备)	
返回值说明	无	
示例	AT+MESHPROVSETNODELPM=0x01 OK	
注意事项	非低功耗节点无需使用该命令；掉电保存	

4.2.11 AT+MESHGETNODEINFO

基本功能：Provisioner获取本地节点及入网节点信息

指令类型	格式	响应
测试指令	AT+MESHGETNODEINFO=?	+MESHGETNODEINFO=< unicast_addr> OK
查询指令	无	无
写指令	AT+MESHGETNODEINFO=< unicast_addr>	OK 或者 +MESHGETNODEINFO:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – node unicast 地址	
返回值说明	无	
示例	AT+MESHGETNODEINFO=0x0002 +MESHNODEINFO:0x0002,NULL,01:02:03:04:05:06,00,06050 40302011e886d15f1ce28ade02,00 ,01,00,00,00 OK	
注意事项		

4.2.12 AT+MESHADDDEV

基本功能：Provisioner添加待入网设备

指令类型	格式	响应
测试指令	AT+MESHADDDEV=?	+MESHADDDEV=< addr>,< addr_type>,< uuid>,< oob_info>,< bearer>,< [auto_add_appkey]> OK
查询指令	无	无
写指令	AT+MESHADDDEV=< addr>,< addr_type>,< uuid>,< oob_info>,< bearer>,< [auto_add_appkey]>	OK 或者 +MESHADDDEV:< ERR> ERROR
执行指令	无	无
参数说明	<p>Addr – 设备MAC地址，格式为xx:xx:xx:xx:xx:xx</p> <p>Addr_type – 设备地址类型</p> <p>0:Public</p> <p>1:random</p> <p>Uuid – 设备UUID,16个字节</p> <p>Oob_info –</p> <p>0x00: NO OOB</p> <p>0x01:static oob</p> <p>Bearer –</p> <p>0x01:PB-ADV</p> <p>0x02:PB-GATT</p> <p>auto_add_appkey – 0(不自动添加appkey)/1(自动添加appkey),默认为1</p>	
返回值说明	无	
示例	AT+MESHADDDEV=01:02:03:04:05:06,00,06050403020111e8 86d15f1ce28ade02,00,01 OK	
注意事项		

4.2.13 AT+MESHAUTOCONFIG

基本功能：Provisioner自动配置已入网节点

指令类型	格式	响应
测试指令	AT+MESHAUTOCONFIG=?	+MESHAUTOCONFIG= < addr>,[sub_addr],[timeout],[retry]
查询指令	无	无
写指令	AT+MESHAUTOCONFIG= < addr>,[sub_addr],[timeout],[retry]	OK 或者 +MESHAUTOCONFIG:< ERR> ERROR
执行指令	无	无
参数说明	addr – 设备地址（支持单播/组播/广播） sub_addr – 自动绑定订阅地址 timeout – 自动绑定超时时间(单位s,最小值为3s,默认3s) retry – 自动绑定重试次数(单播默认5次，组播默认20次)	
返回值说明	无	
注意事项		

4.2.14 AT+MESHDELDEV

基本功能：Provisioner删除待入网设备

指令类型	格式	响应
测试指令	AT+MESHDELDEV=?	+MESHDELDEV=< addr>,< addr_type>,< uuid> OK
查询指令	无	无
写指令	AT+MESHDELDEV=< addr>,< addr_type>,< uuid>	OK 或者 +MESHDELDEV:< ERR> ERROR
执行指令	无	无
参数说明	addr:设备MAC地址, 格式为xx:xx:xx:xx:xx:xx addr_type:设备地址类型 0:Public 1:random uuid:设备UUID,16个字节	
返回值说明	无	
示例	AT+MESHDELDEV=01:02:03:04:05:06,00,06050403020111e8 86d15f1ce28ade02 OK	
注意事项	该命令会删除队列中未入网设备；若设备在入网过程中，会停止入网，节点上报+MESHPROVFAILED事件,若设备已入网则无法删除	

4.2.15 AT+MESHOOB

基本功能：Provisioner OOB输入

指令类型	格式	响应
测试指令	AT+MESHOOB=?	+MESHOOB=< oob> OK
查询指令	无	无
写指令	AT+MESHOOB=< oob>	OK 或者 +MESHOOB:< ERR> ERROR
执行指令	无	无
参数说明	oob – oob信息	
返回值说明	无	
示例	+MESHOOBINPUT:03,16 AT+MESHOOB=0123456789ABCDEF OK	
注意事项	根据+MESHOOBINPUT事件输入oob数据及相应长度 若node节点支持OOB信息，用户须在入网过程中根据OOB输入提示事件，利用该指令进行输入	

4.2.16 AT+MESHGETNETKEY

基本功能：获取指定netkey

指令类型	格式	响应
测试指令	AT+MESHGETNETKEY=< netkey_idx>	+MESHGETNETKEY=< netkey_idx> OK
查询指令	无	无
写指令	AT+MESHGETNETKEY=< netkey_idx>	OK 或者 +MESHGETNETKEY:< ERR> ERROR
执行指令	无	无
参数说明	netkey_idx – netkey索引	
返回值说明	无	
示例	AT+MESHGETNETKEY=0 +MESHGETNETKEY:00,82611e940faca7da42180fcca1027a82 OK	
注意事项	无	

4.2.17 AT+MESHSETNETKEY

基本功能：添加netkey

指令类型	格式	响应
测试指令	AT+MESHSETNETKEY=<netkey_idx>,[key_val]	+MESHSETNETKEY=<netkey_idx>,[key_val] OK
查询指令	无	无
写指令	AT+MESHSETNETKEY=<netkey_idx>,[key_val]	OK 或者 +MESHSETNETKEY:< ERR> ERROR
执行指令	无	无
参数说明	netkey_idx – netkey索引 key_val – appkey val	
返回值说明	无	
示例	AT+MESHSETNETKEY=1 +MESHSETNETKEY:1 OK	
注意事项	若未设置netkey_val,将自动生成netkey,且返回的netkey_idx为已有key idx的递增, 如已有netkey 0,则在未输入key_val的情况下,输出的key_idx 为1;若设置了key_val,则将会强行将当前key_idx所对应的netkey删除并设置为输入的net_val,且输出的netkey_idx为输入值	

4.2.18 AT+MESHGETAPPKEY

基本功能：获取指定appkey

指令类型	格式	响应
测试指令	AT+MESHGETAPPKEY=?	+MESHGETAPPKEY=< netkey_idx>,< appkey_idx> OK
查询指令	无	无
写指令	AT+MESHGETAPPKEY=< netkey_idx>,< appkey_idx>	OK 或者 +MESHGETAPPKEY:< ERR> ERROR
执行指令	无	无
参数说明	netkey_idx – netkey索引 appkey_idx – appkey索引	
返回值说明	无	
示例	AT+MESHGETAPPKEY=0x00,0x00 +MESHGETAPPKEY:00,00,adb8b735560e520aa039d82780d 2a35f OK	
注意事项	无	

4.2.19 AT+MESHSETAPPKEY

基本功能：添加app key

指令类型	格式	响应
测试指令	AT+MESHSETAPPKEY=< netkey_idx>,< appkey_idx>, [key_val]	+MESHSETAPPKEY=< netkey_idx>,< appkey_idx>, [key_val] OK
查询指令	无	无
写指令	AT+MESHSETAPPKEY=< netkey_idx>,< appkey_idx>, [key_val]	OK 或者 +MESHSETAPPKEY:< ERR> ERROR
执行指令	无	无
参数说明	netkey_idx – netkey索引 appkey_idx – appkey索引 key_val – appkey val	
返回值说明	无	
示例	AT+MESHSETAPPKEY=0,0 +MESHSETAPPKEY:0,1 OK	
注意事项	若未设置key_val,将自动生成appkey,且返回的appkey_idx为已有key_idx的递增,如当provisioner中已有appkey 0,则在未输入key_val的情况下,输出的key_idx 为1;若设置了key_val,则将会强行将当前key_idx所对应的appkey删除并设置为输入的key_val,且输出的appkey_idx为输入值	

4.2.20 AT+MESHCLEARRPL

基本功能：清空RPL记录

指令类型	格式	响应
测试指令	AT+MESHCLEARRPL=?	+MESHCLEARRPL=< unicast_addr>
查询指令	无	无
写指令	AT+MESHCLEARRPL=< unicast_addr>	OK
执行指令	AT+MESHCLEARRPL	OK
参数说明	无	
返回值说明	无	
示例	AT+MESHCLEARRPL OK	
注意事项	不带参数情况下，清除所有节点RPL记录，带参数情况下，清除指定node RPL记录	

4.3 Config Client model AT命令

4.3.1 AT+MESHADDAPPKEY

基本功能：通过CFG CLI 给node节点添加app key

指令类型	格式	响应
测试指令	AT+MESHADDAPPKEY=< unicast_addr>,< appkey_idx>	+MESHADDAPPKEY=< unicast_addr>,< appkey_idx> OK
查询指令	无	无
写指令	AT+MESHADDAPPKEY=< unicast_addr>,< appkey_idx>,<	OK 或者 +MESHADDAPPKEY:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 节点unicast addr key_app_idx – app key 索引	
返回值说明	无	
示例	AT+MESHADDAPPKEY=0x0002,0 OK +MESHAPPKEYADD:0002,0	
注意事项	执行成功后对端节点返回+MESHAPPKEYADD事件	

4.3.2 AT+MESHBINDAPPKEY

基本功能：通过CFG CLI 给node节点绑定app key

指令类型	格式	响应
测试指令	AT+MESHBINDAPPKEY=?	+MESHBINDAPPKEY=< unicast_addr>,< appkey_idx>,< mod_id>,[cid] OK
查询指令	无	无
写指令	AT+MESHBINDAPPKEY=< unicast_addr>,< appkey_idx>,< mod_id>,[cid]	OK 或者 +MESHBINDAPPKEY:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr appkey_idx –app key 索引 mod_id – model ID cid – vendor model company id	
返回值说明	无	
示例	AT+MESHBINDAPPKEY=0x0002,0x0000,0x1000 OK +MESHAPPKEYBIND:0002,0	
注意事项	执行成功后对端节点返回+MESHAPPKEYBIND事件	

4.3.3 AT+MESHUNBINDAPPKEY

基本功能： 通过CFG CLI 给node节点解绑app key

指令类型	格式	响应
测试指令	AT+MESHUNBINDAPPKEY=?	+MESHUNBINDAPPKEY=< unicast_addr>,< appkey_idx>,< mod_id>,[cid] OK
查询指令	无	无
写指令	AT+MESHUNBINDAPPKEY=< unicast_addr>,< appkey_idx>,< mod_id>,[cid]	OK 或者 +MESHUNBINDAPPKEY:< ERR> ERROR
执行指令	无	
参数说明	unicast_addr – 目标unicast addr app_key_idx – app key 索引 mod_id – model ID cid – vendor model company id	
返回值说明	无	
示例	AT+MESHUNBINDAPPKEY=0x0002,0,0x1000 OK +MESHAPPKEYBIND:0002,0	
注意事项	执行成功后对端节点返回+MESHAPPKEYBIND事件	

4.3.4 AT+MESHGETCOMP

基本功能： 通过CFG CLI 获取node节点信息

指令类型	格式	响应
测试指令	AT+MESHGETCOMP=?	+MESHGETCOMP=< unicast_addr>,[page] OK
查询指令	无	无
写指令	AT+MESHGETCOMP=< unicast_addr>,[page]	OK 或者 +MESHGETCOMP:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr page – comp page	
返回值说明	无	
示例	<p style="text-align: right;">AT+MESHGETCOMP=0x0001</p> <p>OK</p> <p>+MESHCOMPDATA:0001,01a8,0000,0000,000a,0007,0000,08 ,01,0000,0001,0002,0003,1001, 1003,1302,1305,01a8,0001</p>	
注意事项	无	

4.3.5 AT+MESHRELAY

基本功能：通过CFG CLI 配置node relay属性

指令类型	格式	响应
测试指令	AT+MESHRELAY=?	+MESHRELAY=< unicast_addr>,< onoff>, [count],[interval] OK
查询指令	无	无
写指令	AT+MESHRELAY=< unicast_addr>,< onoff>, [count],[interval]	OK 或者 +MESHRELAY:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr onoff – 关闭或使能relay属性 count – Relay 重试次数 interval – 重试周期	
返回值说明	无	
示例	OK	AT+MESHRELAY=0x0002,0x01 +MESHRELAY:0002,01
注意事项	执行成功后返回+MESHRELAY事件	

4.3.6 AT+MESHPROXY

基本功能：通过CFG CLI 配置node proxy属性

指令类型	格式	响应
测试指令	AT+MESHPROXY=?	+MESHPROXY= < unicast_addr>,< onoff> OK
查询指令	无	无
写指令	AT+MESHPROXY=< unicast_addr>,< onoff>	OK 或者 +MESHPROXY:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr: 目标unicast addr onoff :关闭或使能proxy属性	
返回值说明	无	
示例	AT+MESHPROXY=0x0002,0x01 OK +MESHPROXY:0002,01	
注意事项	执行成功后返回+MESHPROXY事件	

4.3.7 AT+MESHFRIEND

基本功能： 通过CFG CLI 配置node friend属性

指令类型	格式	响应
测试指令	AT+MESHFRIEND=?	+MESHFRIEND=< unicast_addr>,< onoff> OK
查询指令	无	无
写指令	AT+MESHFRIEND=< unicast_addr>,< onoff>	OK 或者 +MESHFRIEND:< ERR> ERROR
执行指令	无	无
参数说明	无	
返回值说明	无	
示例	AT+MESHFRIEND=0x0002,0x01 OK	+MESHFRIEND:0002,01
注意事项	执行成功后返回+MESHFRIEND事件	

4.3.8 AT+MESHGETSUB

基本功能：通过CFG CLI 获取node model sub属性

指令类型	格式	响应
测试指令	AT+MESHGETSUB=?	+MESHGETSUB=< unicat_addr>,< mod_id>,[cid] OK
查询指令	无	无
写指令	AT+MESHGETSUB=< unicat_addr>,< mod_id>,[cid]	OK 或者 +MESHGETSUB:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr mod_id – 设备model ID cid – vendor model company id	
返回值说明	无	
示例	AT+MESHGETSUB=0x0002,0x1000 OK +MESHSUBLIST:0002,ffff,1000,f000,c000	
注意事项	执行成功返回+MESHSUBLIST事件	

4.3.9 AT+MESHSETSUB

基本功能：通过CFG CLI 配置node model sub属性

指令类型	格式	响应
测试指令	AT+MESHSETSUB=?	+MESHSETSUB=< unicat_addr>,< mod_id>,< sub_addr>,[cid] OK
查询指令	无	无
写指令	AT+MESHSETSUB=< unicat_addr>,< mod_id>,< sub_addr>,[cid]	OK 或者 +MESHSETSUB:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr mod_id – 设备model ID sub_addr – SUB地址 cid – vendor model company id	
返回值说明	无	
示例	AT+MESHSETSUB=0x0002,0x1000,0xC000 OK +MESHSUBSET:0002,0	
注意事项	执行成功返回+MESHSUBSET事件	

4.3.10 AT+MESHDELSUB

基本功能：通过CFG CLI 删除node model sub地址

指令类型	格式	响应
测试指令	AT+MESHDELSUB=?	+MESHDELSUB=< unicat_addr>,< mod_id>,< sub_addr>,[cid] OK
查询指令	无	无
写指令	AT+MESHDELSUB=< unicat_addr>,< mod_id>,< sub_addr>[cid]	OK 或者 +MESHDELSUB:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr mod_id – 设备model ID sub_addr – SUB地址 cid – vendor model company id	
返回值说明	无	
示例	AT+MESHDELSUB=0x0002,0x1000,0xf000 OK +MESHSUBSET:0002,0	
注意事项	执行成功返回+MESHSUBSET事件；注意当SUB设置为0x0000，则清除该model下所有订阅地址	

4.3.11 AT+MESHSUBLISTOVERWRITE

基本功能：通过vendor model 覆盖node model sub地址

指令类型	格式	响应
测试指令	AT+MESHSUBLISTOVERWRITE=?	+MESHSUBLISTOVERWRITE=<unicat_addr>,<mod_id>,<cid>,<list_size>,<subaddr_list> OK
查询指令	无	无
写指令	AT+MESHSUBLISTOVERWRITE=<unicat_addr>,<mod_id>,<cid>,<list_size>,<subaddr_list>	OK 或者 +MESHSUBLISTOVERWRITE< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr mod_id – 设备model ID cid – vendor model company id, 无则输入0xffff list_size – 订阅地址列表大小 sub_addr – 订阅地址列表	
返回值说明	无	
示例	AT+MESHSUBLISTOVERWRITE=0X02,0X1000,0Xffff,0x02,0xC001,0XC002 OK +MESHSUBLISTOVERWRITE:0002,00	
注意事项	执行成功返回+MESHSUBLISTOVERWRITE事件；注意该命令需要节点和配网测都支持vendor_model	

4.3.12 AT+MESHGETPUB

基本功能：通过CFG CLI 获取node model pub属性

指令类型	格式	响应
测试指令	AT+MESHGETPUB=?	+MESHGETPUB=< unicat_addr>,< mod_id>,[cid] OK
查询指令	无	无
写指令	AT+MESHGETPUB=< unicat_addr>,< mod_id>,[cid]	OK 或者 +MESHGETPUB:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr mod_id – 设备model ID cid – vendor model company id	
返回值说明	无	
示例	AT+MESHGETPUB=0x0002,0x1000 OK +MESHGETPUB:0002,ffff,1000,c000,03,43,03,64	
注意事项	执行成功返回+MESHGETPUB事件 节点默认无pub参数	

4.3.13 AT+MESHSETPUB

基本功能： 通过CFG CLI 配置node model pub属性

指令类型	格式	响应
测试指令	AT+MESHSETPUB=?	+MESHSETPUB=< unicat_addr>,< mod_id>,< pub_addr>,[ttl],[period], [count],[interval], [appkey_idx],[cid] OK

查询指令	无	无
写指令	AT+MESHSETPUB=< unicat_addr>,< mod_id>,< pub_addr>,[ttl],[period], [count],[interval], [appkey_idx],[cid]	OK 或者 +MESHSETPUB:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr mod_id – model ID addr – PUB地址 wei ttl – PUB TTL,默认为3 period – PUB周期, 默认为0 count – PUB重传次数, 默认为0 interval – 重发间隔, 默认为0 appkey_idx – pub使用的app key索引, 默认为0 cid – vendor model company id, 默认无	
返回值说明	无	
示例	AT+MESHSETPUB=0x0002,0x1000,0xC000,0x03,0x43,3,0x64 OK +MESHSETPUB:0002,0	
注意事项	成功执行返回+MESHSETPUB事件 PUB参数设备不支持删除操作, 若想更新pub参数, 可直接利用该命令设置新的参数 若想订阅对端节点pub的消息, 应使用AT+MESHSETSUB订阅该地址 period pub周期设置方法如下: 00x40 0x80 0xC0 interval % 50 需为0	

4.3.14 AT+MESHRST

基本功能：通过CFG CLI 将node移除Mesh网络

指令类型	格式	响应
测试指令	AT+MESHRST=?	+MESHRST=< unicast_addr> OK
查询指令	无	无
写指令	AT+MESHRST=< unicast_addr>	OK 或者 +MESHRST:< ERR> ERROR
执行指令	无	无
参数说明	unicast_addr – 目标unicast addr	
返回值说明	无	
示例	<p style="text-align: right;">AT+MESHRST=0x0002</p> <p>OK</p> <p>+MESHRST:0002</p>	
注意事项	成功执行返回+MESHRST事件	

4.4 Generic OnOff Client model AT命令

4.4.1 AT+MESHONOFF

基本功能：通过ONOFF CLI 发送消息

指令类型	格式	响应
测试指令	AT+MESHONOFF=?	+MESHONOFF=< addr>,< onoff>,[ack],[appkey_idx] OK
查询指令	无	无
写指令	AT+MESHONOFF=< addr>,< onoff>,[ack],[appkey_idx]	OK 或者 +MESHONOFF:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) onoff – onoff status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHONOFF=0x0002,0x01,0x01 OK +MESHONOFF:0002,01	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询	

4.5 Generic Level Client model AT命令

4.5.1 AT+MESHLEVEL

基本功能：通过LEVEL CLI 设置level

指令类型	格式	响应
测试指令	AT+MESHLEVEL=?	+MESHLEVEL=< addr>,< level>,[ack],[appkey_idx] OK
查询指令	无	无
写指令	AT+MESHLEVEL=< addr>,< level>,[ack],[appkey_idx]	OK 或者 +MESHLEVEL:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) level – level status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHLEVEL=0x0002,0x0001,0x01 OK +MESHLEVEL:0002,0001	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询	

4.5.2 AT+MESHLEVELMOVE

基本功能：通过LEVEL CLI 设置level move

指令类型	格式	响应
测试指令	AT+MESHLEVELMOVE?	+MESHLEVELMOVE=< addr>,< move>,[ack], [appkey_idx] OK
查询指令	无	无
写指令	AT+MESHLEVELMOVE=< addr>,< move>,[ack], [appkey_idx]	OK 或者 +MESHLEVELMOVE:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) move – level move status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHLEVELMOVE=0x0002,0x0001,0x01 OK +MESHLEVEL:0002,0001	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询	

4.5.3 AT+MESHLEVELDELTA

基本功能：通过LEVEL CLI 发送消息设置level delta

指令类型	格式	响应
测试指令	AT+MESHLEVELDELTA=?	+MESHLEVELDELTA=< addr>,< delta>,[ack], [appkey_idx] OK
查询指令	无	无
写指令	AT+MESHLEVELDELTA=< addr>,< delta>,[ack], [appkey_idx]	OK 或者 +MESHLEVELDELTA:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) delta – level delta status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHLEVELDELTA=0x0002,0x0001,0x01 OK +MESHLEVEL:0002,0002	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询	

4.6 Light Lightness Client model AT命令

4.6.1 AT+MESHLIGHTNESSRANGE

基本功能：通过LEVEL CLI 设置lightness range

指令类型	格式	响应
测试指令	AT+MESHLIGHTNESSRANGE =?	+MESHLIGHTNESSRANGE=< addr>,< range_min>,< range_max>,[ack], [appkey_idx] OK
查询指令	无	无
写指令	AT+MESHLIGHTNESSRANGE =< addr>,< range_min>,< range_max>,[ack], [appkey_idx]	OK 或者 +MESHLIGHTNESSRANGE:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) range_min – lightness最小值(>0) range_max – lightness 最大值(>0) ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – appkey 索引, 默认为0	
返回值说明	无	
示例	AT+MESHLIGHTNESSRANGE=0x0004,0x0001,0x0005,0x01 OK +MESHLIGHTNESSRANGE:0002,00,0001,0005	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询	

4.6.2 AT+MESHLIGHTNESSDEF

基本功能：通过LEVEL CLI设置lightness default

指令类型	格式	响应
测试指令	AT+MESHLIGHTNESSDEF=?	+MESHLIGHTNESSDEF=<addr>,<lightness>,[ack],[appkey_idx] OK
查询指令	无	无
写指令	AT+MESHLIGHTNESSDEF=<addr>,<lightness>,[ack],[appkey_idx]	OK 或者 +MESHLIGHTNESSDEF:<ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) lightness – light lightness default status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHLIGHTNESSDEF=0x0002,0x0002,0x01 OK +MESHLIGHTNESSDEF:0002,0002	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询 使用前需使用AT+MESHLIGHTNESSRANGE进行设置	

4.6.3 AT+MESHLIGHTNESS

基本功能: 通过LEVEL CLI 设置lightness

指令类型	格式	响应
测试指令	AT+MESHLIGHTNESS=?	+MESHLIGHTNESS=< addr>, < lightness>,[ack], [appkey_idx] OK
查询指令	无	无
写指令	AT+MESHONOFF=< addr>,< lightness>,[ack],[appkey_idx]	OK 或者 +MESHLIGHTNESS:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址 lightness – light lightness status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHLIGHTNESS=0x0002,0x0003,0x01 OK +MESHLIGHTNESS:0002,0003	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询 使用前需使用AT+MESHLIGHTNESSRANGE进行设置	

4.6.4 AT+MESHLIGHTNESSLIN

基本功能：通过LEVEL CLI 设置lightness linear

指令类型	格式	响应
测试指令	AT+MESHLIGHTNESSLIN=?	+MESHLIGHTNESSLIN=< addr>,< lightnesslin>,[ack], [appkey_idx] OK
查询指令	无	无
写指令	AT+MESHLIGHTNESSLIN=< addr>,< lightnesslin>,[ack], [appkey_idx]	OK 或者 +MESHLIGHTNESSLIN:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) lightnesslin – light lightness linear status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHLIGHTNESSLIN=0x0002,0x0001,0x01 OK +MESHLIGHTNESSLIN:0002,0001	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询	

4.7 Light CTL Client model AT 命令

4.7.1 AT+MESHCTLTEMPRANGE

基本功能：通过LIGHT CTL 设置light ctl temperature range状态

指令类型	格式	响应
测试指令	AT+MESHCTLTEMPRANGE= ?	+MESHCTLTEMPRANGE=< addr>,< range_min>,< range_max>,[ack], [appkey_idx] OK
查询指令	无	无
写指令	AT+MESHCTLTEMPRANGE= < addr>,< range_min>,< range_max>,[ack], [appkey_idx]	OK 或者 +MESHCTLTEMPRANGE:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) range_min – temperature最小值(>= 0x320) range_max – temperature 最大值(ack – 0(不应 答)/1(应答), 默认不带应答 appkey_idx – appkey 索引, 默认为0	
返回值说明	无	
示例	AT+MESHCTLTEMPRANGE=0x0002,0x0340,0x0450,0x01 OK +MESHCTLTEMPRANGE:0002,00,0340,0450	
注意事项	默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可 通过AT+MESHQUERYSTA进行查询	

4.7.2 AT+MESHCTLDEF

基本功能：通过LIGHT CTL 设置light ctl default状态

指令类型	格式	响应
测试指令	AT+MESHCTLDEF=?	+MESHCTLDEF=< addr>,< lightness>,< temperature>,< delta_UV>,[ack],[appkey_idx] OK OK
查询指令	无	无
写指令	AT+MESHCTLDEF=< addr>,< lightness>,< temperature>,< delta_UV>,[ack],[appkey_idx]	OK 或者 +MESHCTLDEF:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) lightness – light lightness default status temperature – light temperature default status delta_UV – light UV default status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHCTLDEF=0x0002,0x0030,0x430,0x0000,0x01 OK +MESHCTLDEF:0002,0030,0430,0000	
注意事项	使用前需使能PROVISIONER默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询 使用前需使用AT+MESHCTLTEMPRANGE进行设置	

4.7.3 AT+MESHCTL

基本功能： 通过LIGHT CTL 设置light ctl状态

指令类型	格式	响应
测试指令	AT+MESHCTL=?	+MESHCTL=< addr>,< lightness>,< temperature>,< delta_UV>,[ack],[appkey_idx] OK
查询指令	无	无
写指令	AT+MESHCTL=< addr>,< lightness>,< temperature>,< delta_UV>,[ack],[appkey_idx]	OK 或者 +MESHCTL:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址 lightness – light lightness status temperature – light temperature status delta_UV – light UV status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHCTL=0x0003,0x0100,0x0342,0x0001,0x01 OK +MESHCTL:0002,0100,0342	
注意事项	使用前需使能PROVISIONER默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询 使用前需使用AT+MESHCTLTEMPRANGE进行设置	

4.7.4 AT+MESHCTLTEMP

基本功能：通过LIGHT CTL 设置light ctl temperature状态

指令类型	格式	响应
测试指令	AT+MESHCTLTEMP=?	+MESHCTLTEMP=< addr>,< temperature>,< delta_UV>,< [ack]>,< [appkey_idx]> OK
查询指令	无	无
写指令	AT+MESHONOFF=< addr>,< onoff>,< [ack]>,< [appkey_idx]>	OK 或者 +MESHONOFF:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) temperature – light temperature status delta_UV – light UV status ack – 0(不应答)/1(应答), 默认不带应答 appkey_idx – app key 索引	
返回值说明	无	
示例	AT+MESHCTLTEMP=0x0002,0x340,0x0003,0x01 OK +MESHCTLTEMP:0002,0340,0003	
注意事项	使用前需使能PROVISIONER默认不带ack, 且使用组播/广播地址时不推荐使用带ACK消息, 可通过AT+MESHQUERYSTA进行查询 使用前需使用AT+MESHCTLTEMPRANGE进行设置	

4.8 Vendor Client model AT命令

4.8.1 AT+MESHTRS

基本功能： 通过venodr model cli透传数据

指令类型	格式	响应
测试指令	AT+MESHTRS=?	+MESHTRS=< addr>,< len>,< data>,[appkey_idx] OK
查询指令	无	无
写指令	AT+MESHTRS=< addr>,< len>,< data>,[appkey_idx]	OK 或者 +MESHTRS:< ERR> ERROR
执行指令	无	无
参数说明	addr – 发送地址(支持单播/广播/组播地址) len – 发送数据长度 (最大44字节) data – 待发送数据 (十六进制表示)	
返回值说明	无	
示例	AT+MESHTRS=0x0002,0x02,0102 OK	
注意事项	不带ACK功能	

5 AT事件

5.1 provisioner扫描设备信息上报事件

消息格式	+MESHDEV:< addr>,< addr_type>,< uuid>,< oob_info>,< bearer>
返回值说明	Addr:设备MAC地址, 格式为xx:xx:xx:xx:xx:xx Addr_type:设备地址类型 0:Public 1:random Uuid:设备UUID,16个字节 Oob_info: 0x00: NO OOB 0x01:static oob Bearer: 0x01: PB-ADV 0x02:PB-GATT
示例	AT+MESHPROVSHOWDEV=0x01 OK +MESHDEV:01:02:03:04:05:05,00,050504030 20111e886d15f1ce28ade02,00,01
注意事项	无

5.2 provisioner扫描设备超时事件

消息格式	+MESHFOUNDDEVTIMEOUT
返回值说明	无
示例	AT+MESHPROVSHOWDEV=0x01,5 OK +MESHFOUNDDEVTIMEOUT
注意事项	该类事件只有在开启设备上报功能时添加设备查找超时时间, 且在超时时间未找到设备才会上报

5.3 Provisioner设备入网失败事件

消息格式	+MESHPROVFAILED:< reason>
返回值说明	reason – 设备配网失败原因
示例	+MESHPROVFAILED:1
注意事项	无

5.4 Provisioner设备入网成功事件

消息格式	+MESHNODEADD:< unicast_addr>,< element_num>,< uuid>
返回值说明	unicast_addr – 入网设备地址 element_num – elem 数量 uuid – 设备uuid
示例	AT+MESHADDDEV=01:02:03:04:05:05,00,0505040302011e886d15f1ce28ade02,00,01,00 OK +MESHNODEADD:0002,01,0505040302011e886d15f1ce28ade02
注意事项	无

5.5 OOB输入提示事件

消息格式	+MESHOOBINPUT:< oob_info>,< oob_len>
返回值说明	oob_info – oob类型 0x01:number oob 0x02:string oob 0x03:static oob oob_len – oob长度
示例	+MESHOOBINPUT:01,03
注意事项	无

5.6 node节点信息事件

消息格式	+MESHNODEINFO:< unicast_addr>,< node_name>,< addr>,< addr_type>,< uuid>,< net_idx>,< elem_num>,< oob_info>,< flags>,< iv_index>
返回值说明	unicast_addr – 入网设备地址 node_name – 设备名称 addr – 设备mac地址 addr_type – mac地址类型 uuid – 设备uuid net_idx – net idx elem_num – element 数量 oob_info – oob information flags – key更新/iv更新标志 iv_index – iv index
示例	AT+MESHGETNODEINFO=0x0002 OK +MESHNODEINFO:0002,NULL,01:02:03:04:05:06,00,0605040302011e886d15f1ce28ade02,00 ,01,00,00,00
注意事项	无

5.7 Mesh Model 上报事件

5.7.1 config model status

5.7.1.1 appkey add status

消息格式	+MESHAPPKEYADD:< unicast_addr>,< status>
返回值说明	unicast_addr – 数据源地址 status – appkey添加状态 0x00:成功 其它：失败
示例	AT+MESHADDAPPKEY=0x0002,0x0001 OK +MESHAPPKEYADD:0002,0
注意事项	无

5.7.1.2 appkey bind status

消息格式	+MESHAPPKEYBIND:< unicast_addr>,< status>
返回值说明	unicast_addr – 数据源地址 status – appkey绑定状态 0x00:成功 其它：失败
示例	AT+MESHBINDAPPKEY=0x0002,0x0000,0x1000 OK +MESHAPPKEYBIND:0002,0
注意事项	无

5.7.1.3 friend status

消息格式	+MESHFRIEND:< unicast_addr>,< status>
返回值说明	unicast_addr – 数据源地址 status – 0(关闭)/1(开启)
示例	AT+MESHQUERYSTA=0x0002,0x0001,0x800f OK +MESHFRIEND:0002,01
注意事项	无

5.7.1.4 proxy status

消息格式	+MESHPROXY:< unicast_addr>,< status>
返回值说明	unicast_addr – 数据源地址 status – 0(关闭)/1(开启)
示例	AT+MESHQUERYSTA=0x0002,0x0001,0x8012 OK +MESHPROXY:0002,01
注意事项	无

5.7.1.5 relay status

消息格式	+MESHRELAY:< unicast_addr>,< status>
返回值说明	unicast_addr – 数据源地址 status – 0(关闭)/1(开启)
示例	AT+MESHQUERYSTA=0x0002,0x0001,0x8026 OK +MESHRELAY:0002,01
注意事项	无

5.7.1.6 RST status

消息格式	+MESHRST:< unicast_addr>
返回值说明	unicast_addr – 数据源地址
示例	AT+MESHRST=0x0002 OK +MESHRST:0002
注意事项	无

5.7.1.7 composition data status

消息格式	+MESHCOMPDATA: < unicast_addr>,< cid>,< pid>,< vid>,< crpl>,< feature>,< elem_idx>,< sig_model_num>,< vnd_model_num>,< sig_model_id0>,..< sig_model_idn>,< vnd_model_cid0>,< vnd_model_id0>..< vnd_model_cidn>,< vnd_model_idn>
返回值说明	unicast_addr – 数据源地址 cid – company id pid – product id vid – verision id crpl – rpl 存储个数 feature – 节点属性, 包含如下属性: relay:0x01 proxy:0x02 friend:0x04 lpn:08 elem_idx – elem 索引 sig_model_num – sig model个数 vnd_model_num – vendor model个数 sig_model_id0 – sig model 0 model id sig_model_idn – sig model n model id vnd_model_cid0 – vendor model 0 company id vnd_model_id0 – vendor model 0 model id vnd_model_idn – vendor model n model id
示例	AT+MESHGETCOMPDATA=0x0001 OK +MESHCOMPDATA:0001,01a8,0000,0000,000a,0007,0000,08,01,0000,0001,0002,0003,1001, 1003,1302,1305,01a8,0001
注意事项	feature各属性可组合

5.7.1.8 mesh sub list status

消息格式	+MESHSUBLIST:< unicast_addr>,< cid>,< mod_id>,< sub_addr0>...< sub_addrn>
返回值说明	unicast_addr – 数据源地址 cid – vendor model company id mod_id –model ID sub_addr0 – model订阅地址0 sub_addrn – model订阅地址n
示例	AT+MESHGETSUB=0x0002,0x1000 OK +MESHSUBLIST:0002,ffff,1000,f000,c000
注意事项	一个model默认最大支持两个subaddr,若该model无sub addr 则返回NULL

5.7.1.9 mesh sub set status

消息格式	+MESHSUBSET:< unicast_addr>,< status>
返回值说明	unicast_addr – 数据源地址 status – sub 设置状态 0x00:成功 其它: 失败
示例	AT+MESHSETSUB=0x0002,0x1000 OK +MESHSUBSET:0003,00
注意事项	

5.7.1.10 mesh sub overwrite status

消息格式	+MESHSUBLISTOVERWRITE:< unicast_addr>,< status>
返回值说明	unicast_addr – 数据源地址 status – sub 设置状态 0x00:成功 其它：失败
示例	AT+MESHSUBLISTOVERWRITE=0X02,0X1000 ,0Xffff,0x02,0xC001,0XC002 OK +MESHSUBLISTOVERWRITE:0002,00
注意事项	

5.7.1.11 mesh pub status

消息格式	+MESH PUB:< unicast_addr>,< cid>,< mod_id>,< pub_addr>,< ttl>,< period>,< count>< interval>
返回值说明	unicast_addr – 数据源地址 cid – vendor model company id mod_id –model ID pub_addr – model发布地址 ttl – 消息ttl period – 消息发送周期 count – 消息重传次数 interval – 消息重传间隔
示例	AT+MESHGETPUB=0x0002,0x1000 OK +MESH PUB:0002,ffff,1000,c000,03,43,03,64
注意事项	

5.7.1.12 mesh pub set status

消息格式	+MESH PUBSET:< unicast_addr>,< status>
返回值说明	status – pub设置状态码 0x00:成功 其它: 失败
示例	AT+MESHSETPUB=0x0002,0x1000,0xC000,0x03,0x44,0x03,0x64 OK +MESH PUBSET:0002,00
注意事项	

5.7.2 onoff model status

消息格式	+MESH ONOFF:< unicast_addr>,< onoff>
返回值说明	unicast_addr – 数据源地址 onoff – onoff状态
示例	AT+MESHQUERYSTA=0x0002,0x1001,0x8201 OK +MESH ONOFF:0002,01
注意事项	无

5.7.3 level model status

消息格式	+MESH LEVEL:< unicast_addr>,< level>
返回值说明	unicast_addr – 数据源地址 level – level状态
示例	AT+MESHQUERYSTA=0x0002,0x1003,0x8205 OK +MESH LEVEL:0002,0000
注意事项	无

5.7.4 light lightness model status

5.7.4.1 lightness status

消息格式	+MESHLIGHTNESS:< unicast_addr>,< lightness>
返回值说明	unicast_addr – 数据源地址 lightness – lightness状态
示例	AT+MESHQUERYSTA=0x0002,0x1302,0x824b OK +MESHLIGHTNESS:0002,0000
注意事项	无

5.7.4.2 lightness linear status

消息格式	+MESHLIGHTNESS:< unicast_addr>,< lightnesslin>
返回值说明	unicast_addr – 数据源地址 lightnesslin – lightness linear状态
示例	AT+MESHQUERYSTA=0x0002,0x1302,0x824f OK +MESHLIGHTNESSLIN:0002,0000
注意事项	无

5.7.4.3 lightness range status

消息格式	+MESHLIGHTNESSRANGE:< unicast_addr>,< code>,< range_min>,< range_max>
返回值说明	unicast_addr-数据源地址 code-上一次操作RANGE结果: 0x00-成功 0x01-设置最小值失败 0x02-设置最大值成功 其它-RFU range_min- lightness min range_max -lightness max
示例	AT+MESHQUERYSTA=0x0002,0x1302,0x8257 OK +MESHLIGHTNESSRANGE:0002,00,0000,0000
注意事项	无

5.7.4.4 lightness last status

消息格式	+MESHLIGHTNESSLAST:< unicast_addr>,< lightness>
返回值说明	unicast_addr - 数据源地址 lightness - lightness last状态
示例	AT+MESHQUERYSTA=0x0002,0x1302,0x8253 OK +MESHLIGHTNESSLAST:0002,0000
注意事项	无

5.7.5 light ctl model status

5.7.5.1 light ctl status

消息格式	+MESHCTL:< unicast_addr>,< lightness>,< temperature>
返回值说明	unicast_addr-数据源地址 lightness-当前lightness temperature-当前temperature
示例	AT+MESHQUERYSTA=0x0002,0x1305,0x825 D OK +MESHCTL:0002,0000,0000
注意事项	无

5.7.5.2 light temperature status

消息格式	+MESHCTLTEMP:< unicast_addr>,< temperature>,< delta_UV>
返回值说明	unicast_addr-数据源地址 temperature-当前temperature delta_UV-当前UV
示例	AT+MESHQUERYSTA=0x0002,0x1305,0x8261 OK +MESHCTLTEMP:0002,0000,0000
注意事项	无

5.7.5.3 ctl temperature range status

消息格式	+MESHCTLTEMPRANGE:< unicast_addr>,< code>< range_min>,< range_max>
返回值说明	unicast_addr-数据源地址 code-上一次操作RANGE结果: 0x00-成功 0x01-设置最小值失败 0x02-设置最大值成功 其它-RFU range_min- temperature min range_max -temperature max
示例	AT+MESHQUERYSTA=0x0002,0x1305,0x826 2 OK +MESHCTLTEMPRANGE:0002,00,0000,0000
注意事项	无

5.7.5.4 ctl defalut status

消息格式	+MESHCTLDEF:< unicast_addr>,< lightness>,< temperature>,< delta_UV>
返回值说明	unicast_addr-数据源地址 temperature-当前temperature delta_UV-当前UV
示例	AT+MESHQUERYSTA=0x0002,0x1305,0x826 7 OK +MESHCTLDEF:0002,0000,0000,0000
注意事项	无

5.7.6 vendor model status

5.7.6.1 透传数据

消息格式	+MESHTRS:< unicast_addr>,< len>,[data]
返回值说明	unicast_addr-数据源地址 len-数据长度 data-当前UV
示例	+MESHTRS:0002,03,010203
注意事项	无

5.7.6.2 autoconfig status

消息格式	+MESHAUTOCONFIG:< unicast_addr>,< status>
返回值说明	unicast_addr-数据源地址 status-自动配置状态 0x00:成功 0x01:未进行 0x02:自动配网进行中 0x03:自动配网失败
示例	AT+MESHQUERYSTA=0x0002,0x0001,0xd701a8,0,0,0xd701a8 OK +MESHAUTOCONFIG:0002,0
注意事项	无

6 AT 使用指南

6.1 准备工作

这里准备3块开发板，其中1块作为provisioner节点，2块作为Mesh节点，provisioner使用AT命令发现设备，并入网设备

6.2 获取未入网设备

provisioner 输入如下AT命令，其中 -> 代表命令响应或者上报事件

```

1  /*开启MESHAT功能*/
2  AT+MESHAT=0x01
3  ->OK
4  /*配置provisioner可用地址*/
5  AT+MESHPROVCONFIG=0x01,0xff
6  ->OK
7  /*provisioner自组网*/
8  AT+MESHPROVAUTOPROV
9  ->OK
10 ->+MESHNODEADD:0001,01,c6c5c4c3c2c100000000000000000000
11 ->+MESHAPPKEYADD:0001,0
12 /*provisioner使能*/
13 AT+MESHPROVEN=0x01
14 ->OK
15 /*provisioner设备上报开启*/
16 AT+MESHPROVSHOWDEV=0x01
17 /*扫描到的未入网设备*/
18 ->+MESHDEV:C1:01:02:00:00:00,00,0000000201c111e886d15f1ce28ade02,00,01
19 ->+MESHDEV:C1:01:02:00:00:00,00,0000000201c111e886d15f1ce28ade02,00,02
20 /*provisioner设备上报关闭*/
21 AT+MESHPROVSHOWDEV=0x00

```

6.3 设备过滤

若待入网设备较多，可使用uuid过滤或者mac过滤的功能，只显示部分设备

```

1  /*开启MESHAT功能*/
2  AT+MESHAT=0x01
3  ->OK
4  /*provisioner设备上报开启*/
5  AT+MESHPROVSHOWDEV=0x01
6  /*使能MAC过滤*/
7  AT+MESHPROVMACFILTER=0x01
8  /*添加MAC过滤地址*/
9  AT+MESHPROVMACFILTERADD=0x01,C1:01:02:00:00:00,0
10 /*provisioner设备上报打开*/
11 AT+MESHPROVSHOWDEV=0x01
12 ->OK
13 /*扫描到的未入网设备*/
14 ->+MESHDEV:C1:01:02:00:00:00,00,0000000201c111e886d15f1ce28ade02,00,01
15 ->+MESHDEV:C1:01:02:00:00:00,00,0000000201c111e886d15f1ce28ade02,00,02
16 /*删除MAC过滤地址*/
17 AT+MESHPROVMACFILTERRM=0x01,C1:01:02:00:00:00,0
18 /*关闭MAC过滤*/
19 AT+MESHPROVMACFILTER=0x00
20 /*provisioner设备上报关闭*/
21 AT+MESHPROVSHOWDEV=0x00

```

6.4 自动配网

使用自动配网功能，Provisioner将自动设置绑定所有model的app key，配置订阅地址和发布地址

6.4.1 单播地址自动配网

通过AT+MESHAUTOCONFIG命令对指定node进行配网

```
Shell | 复制代码
1  /*添加未入网设备进行自动配网*/
2  AT+MESHADDDEV=01:02:03:04:05:05,00,06050403020111e886d15f1ce28ade02,00,01
3  ->OK
4
5  /*添加第二个未入网设备进行自动配网*/
6  AT+MESHADDDEV=01:02:03:04:05:06,00,06050403020111e886d15f1ce28ade02,00,01
7  ->OK
8  /*设备1入网成功*/
9  ->+MESHnodeADD:0002,01,05050403020111e886d15f1ce28ade02
10 /*设备1 appkey添加成功*/
11 ->+MESHAPPKEYADD:0002,0
12 /*设备2入网成功*/
13 ->+MESHnodeADD:0003,01,06050403020111e886d15f1ce28ade02
14 /*设备2 appkey添加成功*/
15 ->+MESHAPPKEYADD:0003,0
16
17 /*设备1自动配网*/
18 AT+MESHAUTOCONFIG=0x0002
19 ->OK
20 /*设备1自动配网完成*/
21 ->+MESHAUTOCONFIG:0002,0
22
23 /*设备2自动配网*/
24 AT+MESHAUTOCONFIG=0x0003
25 ->OK
26 /*设备2自动配网完成*/
27 ->+MESHAUTOCONFIG:0003,0
```

6.4.2 组播地址自动配网

通过组播地址，对指定组的所有node进行配网，适用于大规模网络配置


```
1 /*添加未入网设备进行自动配网*/
2 AT+MESHADDDEV=01:02:03:04:05:05,00,06050403020111e886d15f1ce28ade02,00,01
3 ->OK
4 /*添加第二个未入网设备进行自动配网*/
5 AT+MESHADDDEV=01:02:03:04:05:06,00,06050403020111e886d15f1ce28ade02,00,01
6 ->OK
7
8 ->+MESHnodeADD:0002,01,05050403020111e886d15f1ce28ade02/
9 ->+MESHAPPKEYADD:0002,0
10 ->+MESHnodeADD:0003,01,06050403020111e886d15f1ce28ade02
11 ->+MESHAPPKEYADD:0003,0
12
13 /*node vendor model默认订阅该地址,使用组播地址不返回执行状态*/
14 AT+MESHAUTOCONFIG=0xF000
15 ->OK
16
17 /*查找设备1自动配网状态*/
18 AT+MESHQUERYSTA=0x0002,0x0001,0xd701a8,0,0,0x01a8/*手动查询0x0002号node自动配网状
    态*/
19 ->OK
20 /*设备1自动配网完成*/
21 ->+MESHAUTOCONFIG:0002,0
22
23 /*查找设备2自动配网状态*/
24 AT+MESHQUERYSTA=0x0003,0x0001,0xd701a8,0,0,0x01a8/*手动查询0x0003号node自动配网状
    态*/
25 ->OK
26 /*设备2自动配网完成*/
27 ->+MESHAUTOCONFIG:0003,0/*自动配网完成*/
```

6.5 手动配网

通过单独AT命令，配置指定node的app key，绑定对应model的app key，设置订阅地址和发布地址。

```
1 /*添加入网设备进行手动配网*/
2 AT+MESHADDDEV=01:02:03:04:05:05,00,05050403020111e886d15f1ce28ade02,00,01,00
3 ->OK
4 ->+MESHnodeADD:0002,01,05050403020111e886d15f1ce28ade02
5 /*添加appkey 0*/
6 AT+MESHADDAPPKEY=0x0002,0x0
7 ->OK
8 /*appkey添加成功*/
9 ->+MESHAPPKEYADD:0002,0
10
11 /*获取节点组成信息，方便对对端节点所有model进行绑定*/
12 AT+MESHGETCOMP=0x0002
13 ->OK
14 -
15 >+MESHCOMPDATA:0002,01a8,0000,0000,000a,0007,0000,08,01,0000,0002,1000,1002,1
16 300,
17 1303,1304,1306,01a8,0000
18
19 /*手动绑定appkey 0和0x1000 model*/
20 AT+MESHBINDAPPKEY=0x0002,0x0000,0x1000
21 ->+MESHAPPKEYBIND:0002,0
22 /*手动绑定appkey 0和0x1002 model*/
23 AT+MESHBINDAPPKEY=0x0002,0x0000,0x1002
24 ->+MESHAPPKEYBIND:0002,0
25 /*手动绑定appkey 0和0x1300 model*/
26 AT+MESHBINDAPPKEY=0x0002,0x0000,0x1300
27 ->+MESHAPPKEYBIND:0002,0
28 /*手动绑定appkey 0和0x1303 model*/
29 AT+MESHBINDAPPKEY=0x0002,0x0000,0x1303
30 ->+MESHAPPKEYBIND:0002,0
31 /*手动绑定appkey 0和0x1304 model*/
32 AT+MESHBINDAPPKEY=0x0002,0x0000,0x1304
33 ->+MESHAPPKEYBIND:0002,0
34 /*手动绑定appkey 0和0x1306 model*/
35 AT+MESHBINDAPPKEY=0x0002,0x0000,0x1306
36 ->+MESHAPPKEYBIND:0002,0
37 /*手动绑定appkey 0和0x01a800 vendor model*/
38 AT+MESHBINDAPPKEY=0x0002,0x0000,0x0000,0x01a8
39 ->+MESHAPPKEYBIND:0002,0
40 /*手动设置0x1000 model的订阅地址0xf000*/
41 AT+MESHSETSUB=0x0002,0x1000,0xf000
42 /*手动设置0x1002 model的订阅地址0xf000*/
43 AT+MESHSETSUB=0x0002,0x1002,0xf000
44 /*手动设置0x1300 model的订阅地址0xf000*/
45 AT+MESHSETSUB=0x0002,0x1300,0xf000
46 /*手动设置0x1303 model的订阅地址0xf000*/
47 AT+MESHSETSUB=0x0002,0x1303,0xf000
48 /*手动设置0x1304 model的订阅地址0xf000*/
49 AT+MESHSETSUB=0x0002,0x1304,0xf000
```

```
47 AT+MESHSETSUB=0x0002,0x1304,0xf000
48 /*手动设置0x1306 model的订阅地址0xf000*/
49 AT+MESHSETSUB=0x0002,0x1306,0xf000
50 /*手动设置0x01a800 model的订阅地址0xf000*/
51 AT+MESHSETSUB=0x0002,0x0000,0xf000,0x01a8
52 /*手动设置0x1000 model的发布地址0xf001*/
53 AT+MESHSETPUB=0x0002,0x1000,0xf001,3,0,0,0
```

6.6 AT App key切换

使用App key1替换app key 0

```
1 /*provisioner本地添加appkey 0*/
2 AT+MESHSETAPPKEY=0x0000
3 /*appkey 0已存在, 产生appkey1*/
4 ->+MESHSETAPPKEY:1
5 ->OK
6
7 /*发送新产生的appkey到node节点*/
8 AT+MESHADDAPPKEY=0x0002,0x0001
9 ->OK
10 /*appkey 1添加成功*/
11 ->+MESHAPPKEYADD:0002,0
12 /*解除node节点onoff srv model与appkey 0的绑定*/
13
14 AT+MESHUNBINDAPPKEY=0x0002,0x0000,0x1000
15 /*appkey 0解绑成功*/
16 ->+MESHAPPKEYBIND:0002,0
17 /*绑定node节点onoff srv model与appkey 1*/
18
19 AT+MESHBINDAPPKEY=0x0002,0x0001,0x1000
20 ->OK
21 /*appkey 1绑定成功*/
22 ->+MESHAPPKEYBIND:0003,0
23
24 /*发送新产生的appkey到node节点*/
25 AT+MESHADDAPPKEY=0x0001,0x0001
26 ->OK
27 /*appkey 1添加成功*/
28 ->+MESHAPPKEYADD:0001,0
29 /*解除provisioner节点onoff cli model与appkey 0的绑定*/
30 AT+MESHUNBINDAPPKEY=0x0001,0x0000,0x1001
31 ->OK
32 /*appkey 0解绑成功*/
33 ->+MESHAPPKEYBIND:0001,0
34 /*绑定provisioner节点onoff cli model与appkey 1*/
35 AT+MESHBINDAPPKEY=0x0001,0x0001,0x1001
36 ->OK
37 /*使用APPKEY 1发送不带ACK开灯*/
38 AT+MESHONOFF=0x0002,0x01,0x00,0x0001
39 ->OK
40
41 /*使用appkey1查询onoff status*/
42 AT+MESHQUERYSTA=0x0002,0x1001,0x8201,0x0000,0x0001
43 ->OK
44 +MESHONOFF:0002,01
45
46 /*使用APPKEY 1发送带ACK关灯*/
47 AT+MESHONOFF=0x0002,0x00,0x01,0x0001
48 ->OK
```

6.7 SUB/PUB地址配置

6.7.1 provisioner pub /node sub

```

1 /*node节点ONOFF srv model增加订阅地址0xC000*/
2 AT+MESHSETSUB=0x0002,0x1000,0xC000
3 ->OK
4 /*订阅成功*/
5 ->+MESHSUBSET:0002,0
6 /*向0xC000地址发送开灯命令*/
7 AT+MESHONOFF=0xC000,0x01,0x01
8 ->OK
9 ->+MESHONOFF:0002,01

```

Shell 复制代码

6.7.2 provisioner sub /node pub

```

1 /*查询provisioner节点ONOFF cli model已订阅地址*/
2 AT+MESHGETSUB=0x0001,0x1001
3 ->OK
4 ->+MESHUBLIST:0001,ffff,1001,f001,c000
5 /*provisioner节点 model订阅地址最多为两个，若需增加新地址，需删除已有地址*/
6 AT+MESHDELSUB=0x0001,0x1001,0xC000
7 ->OK
8 ->+MESHSUBSET:0001,0
9 /*provisioner节点ONOFF cli model增加订阅地址0xC001*/
10 AT+MESHSETSUB=0x0001,0x1001,0xC001
11 ->OK
12 /*订阅成功*/
13 ->+MESHSUBSET:0001,0
14 /*设置node节点pub地址0xC001及pub周期*/
15 AT+MESHSETPUB=0x0002,0x1000,0xC001,0x03,0x44,0x00,0x00
16 ->OK
17 /*provisioner节点周期性地接收到node节点周期性pub的onoff status*/
18 ->+MESHSTATUS:0002,0
19 ->+MESHONOFF:0002,01
20 ->+MESHONOFF:0002,01ig

```

Shell 复制代码

7. 附录

7.1 Mesh Sig Model ID查询表

model id	说明
config server model	0x0000
config client model	0x0001
health server model	0x0002
health client model	0x0003
generic onoff server model	0x1000
generic onoff client model	0x1001
generic level server model	0x1002
generic level client model	0x1003
light lightness server model	0x1300
lightlightness client model	0x1302
light CTL server model	0x1303
light CTL client model	0x1305

7.2 Mesh vendor Model ID查询表

model id	CID	说明
vendor server model	0X01a8	0x0000
vendor client model	0x01a8	0x0001

7.3 Mesh Model 状态查询表

status	query model id	query opcode	cid
friend status	0x0001	0x800f	NULL
proxy status	0x0001	0x8012	NULL
relay status	0x0001	0x8026	NULL
generic onoff status	0x1001	0x8201	NULL
generic level status	0x1003	0x8205	NULL
light lightness status	0x1302	0x824b	NULL
light lightness linear status	0x1302	0x824f	NULL
light lightness last status	0x1302	0x8253	NULL
light lightness default status	0x1302	0x8255	NULL
lightness range status	0x1302	0x8257	NULL
light CTL status	0x1305	0x825D	NULL
light CTL temperature status	0x1305	0x8261	NULL
light CTL default status	0x1305	0x8267	NULL
light CTL range status	0x1305	0x8262	NULL
vendor autoconfig status	0xd701a8	0x0001	0x01a8