**SIG opcode的INI 格式如下：**

typedef struct{

u16 nk\_idx; //netkey index

u16 ak\_idx; //app\_key index

u8 retry\_cnt; //应用层需要retry的次数，适用于需要回复的命令，为0表示使用“retry”控件的值。

u8 rsp\_max; //表示需要回复节点的个数，0和1是效果相同，用来判断是否需要retry

u16 adr\_dst; // 目标地址

u8 op; //操作码的第一个byte，取值范围0x00—0xBF

u8 par[MESH\_CMD\_ACCESS\_LEN\_MAX]; //操作码的剩余byte + parameter after opcode

}mesh\_bulk\_cmd\_par\_t;

其中 op + par 就是access layer的参数。

如果par的参数区中存在tid字段，该字段写0,即可，协议栈内部会统一维护tid的值，并进行赋值。

**Vendor opcode的INI 格式如下：**

typedef struct{

u16 nk\_idx; //netkey index

u16 ak\_idx; //app\_key index

u8 retry\_cnt; // 应用层需要retry的次数，适用于需要回复的命令，为0表示使用“retry”控件的值。

u8 rsp\_max; // 表示需要回复节点的个数，0和1是效果相同，用来判断是否需要retry

u16 adr\_dst; // 目标地址

u8 op; //操作码，取值范围0xC0—0xFF

u16 cid; //company id，即vendor id

u8 op\_rsp; //伪参数，内部使用，不会发送出去，详见下文说明

u8 tid\_pos; //伪参数，内部使用，不会发送出去，详见下文说明

u8 par[MESH\_CMD\_ACCESS\_LEN\_MAX]; // parameter after opcode

};

Vendor opcode 的op\_rsp：

Opcode对应的status opcode：取值范围0x00，或者0xC0—0xFF，如果为0，则表示没有status 回复。

tid\_pos：

参数区中tid的位置：如果为0，则表示没有tid，如果非零，则表示tid在par中的偏移位置等于tid\_pos-1；

如果par的参数区中存在tid字段，该字段写0,即可，协议栈内部会统一维护tid的值，并进行赋值。

**注意：**op\_rsp和tid\_pos都是内部使用的参数，并且不会发送出去，也不计入参数长度，但是必须填写。

**SIG opcode举例：g\_all\_on：**

CMD-g\_all\_on = e8 ff 00 00 00 00 00 00 ff ff 82 02 01 00

e8 ff：标识符, 不包含在mesh\_bulk\_cmd\_par\_t里面。

00 00：network\_key index

00 00：App\_key index

00：应用层retry次数使用“retry”控件的值，默认是2

00：不设置回复的节点的个数，对于有回复的命令，效果等同于1.

ff ff：目标地址为广播地址

82 02： generic onoff命令

01：表示开

00：tid

**vendor opcode举例：vendor\_on：**

CMD- vendor\_on = e8 ff 00 00 00 00 00 00 ff ff c2 11 02 c4 02 01 00

e8 ff：标识符, 不包含在mesh\_bulk\_cmd\_par\_t里面。

00 00：network\_key index

00 00：App\_key index

00：应用层retry次数使用“retry”控件的值，默认是2

00：不设置回复的节点的个数

ff ff：目标地址为广播地址

c2： vendor\_on命令

11 02：company id是0x0211

C4：表示vendor\_on（C2） 的的status 回复是C4。

02：表示vendor\_on命令有tid字段，tid在par中的偏移位置等于1 (=2-1)

01：表示开

00：tid