

Generic OnOff Model Application Note

V 0.0.1

2019/1/29

修订历史（Revision History）

日期	版本	修改	作者	Reviewer
2018/09/25	Draft v0.0.1	初稿	hector	

Realtek Confidential

目 录

修订历史 (Revision History)	2
目 录	3
表目录	5
图目录	6
词汇表	7
1 背景介绍	8
2 状态	9
2.1 开关	9
3 SIG 消息	10
3.1 Generic OnOff Get	10
3.2 Generic OnOff Set	10
3.3 Generic OnOff Set Unacknowledged	10
3.4 Generic OnOff Status	10
4 Generic OnOff Server	12
4.1 接口	12
4.2 消息	12
4.2.1 GENERIC_ON_OFF_SERVER_GET	12
4.2.2 GENERIC_ON_OFF_SERVER_GET_DEFAULT_TRANSITION_TIME	12
4.2.3 GENERIC_ON_OFF_SERVER_SET	13
5 Generic OnOff Client	14
5.1 接口	14
5.2 消息	14
5.2.1 GENERIC_ON_OFF_CLIENT_STATUS	14
6 示例	15
6.1 Server Model 示例	15
6.2 Client Model 示例	16
参考文献	18
附录	19

Realtek Confidential

表目录

表 2-1 Generic OnOff 状态 9

表 3-1 Generic OnOff Set 消息 10

表 3-2 Generic OnOff 状态 10

表 4-1 Generic OnOff ServerModel 接口 12

表 4-2 Generic OnOff Server Get 消息 12

表 4-3 Generic OnOff Server Get Default Transition Time 消息 12

表 4-4 Generic OnOff Server Set 消息 13

表 5-1 Generic OnOff Client Model 接口 14

表 5-2 Generic OnOff Client Status 消息 14

表 6-1 Generic OnOff Server Model 示例 15

表 6-2 Generic OnOff Client Model 示例 16

图目录

未找到图形项目表。

Realtek Confidential

词汇表

缩写	含义

Realtek Confidential

1 背景介绍

Generic OnOff Model 主要是用来控制 element 的开关状态，是 SIG Model 中比较简单的一个，仅包含开关状态，比较适合用来做 Model 入门的学习。

Realtek Confidential

2 状态

Generic OnOff Model 只包含一个开关状态。

2.1 开关

开关只有两个状态，开和关，状态的含义由具体的 Model 指定。开关并不支持 transition 的中间状态，所以当 transition 开始从 0x00 转变为 0x01 时，开关的状态应该立刻变成开，同样当 transition 开始从 0x01 转变为 0x00 结束时，开关的状态应该立刻变成关。

表 2-1 Generic OnOff 状态

值	描述
0x00	关
0x01	开
0x02-0xFF	无效值

3 SIG 消息

Generic OnOff 的消息比较简单，主要有设置开关状态和获取开关状态。

3.1 Generic OnOff Get

此消息用来获取 element 的开关状态，远端收到此消息时会回一个 Generic OnOff Status 消息。此消息没有参数。

3.2 Generic OnOff Set

此消息用来设置 element 的开关状态，远端收到此消息时做相应的开关动作，并会回一个 Generic OnOff Status 消息。参数如表 3-1 所示。

表 3-1 Generic OnOff Set 消息

域	大小(字节)	注释
OnOff	1	目标 Generic OnOff 状态
TID	1	事务 ID，用来标记是新的消息还是重传的消息
Transition Time	1	Transition 的时间(可选)。此字段不存在时，开关状态应该立刻改变。存在时，开关状态应该在 Transition 的时间之内完成
Delay	1	消息执行的延时，5ms 步进(可选)，当 Transition 存在时，也会存在

3.3 Generic OnOff Set Unacknowledged

此消息用来设置 element 的开关状态，远端收到此消息时做相应的开关动作，此消息不需要回复。参数同表 3-1。

3.4 Generic OnOff Status

此消息用来报告 element 当前的开关状态。参数如表 3-2。

表 3-2 Generic OnOff 状态

域	大小(字节)	注释
---	--------	----

Present OnOff	1	当前开关状态
Target OnOff	1	目标开关状态
Remaining Time	1	切换到目标状态剩余时间(可选), 当 Target OnOff 字段存在时才会 存在

Realtek Confidential

4 Generic OnOff Server

4.1 接口

表 4-1 Generic OnOff ServerModel 接口

函数	作用
generic_on_off_server_reg	注册 Generic OnOff Server Model
generic_on_off_publish	Publish Generic OnOff 的状态

4.2 消息

4.2.1 GENERIC_ON_OFF_SERVER_GET

当 Model 需要获取 element 的开关状态时，会发送此消息到 App 层。消息的参数如表 4-2。

表 4-2 Generic OnOff Server Get 消息

```
typedef struct
{
    generic_on_off_t on_off;
} generic_on_off_server_get_t;
```

4.2.2 GENERIC_ON_OFF_SERVER_GET_DEFAULT_TRANSITION_TIME

Model spec 规定，当设置 element 的开关状态时如果不带 transition 参数，需要使用默认的 transition 参数，当默认的 transition 也不存在时，状态的改变立刻执行。所以 Model 会通过此消息从 App 层获取默认的 transition 参数，在没有默认 transition 参数的情况下，App 可以不处理这个消息。消息参数如表 4-3。

表 4-3 Generic OnOff Server Get Default Transition Time 消息

```
typedef struct
{
    generic_transition_time_t trans_time;
} generic_on_off_server_get_default_transition_time_t;
```

4.2.3 GENERIC_ON_OFF_SERVER_SET

当 Model 收到远端的设置开关状态消息时会发送此消息给 App 层。消息的参数如表 4-4，on_off 为目标状态，total_time 为总的 transition 时间，当 total_time 中的 num_steps 为 0 时，转换应该立刻生效，remaining_time 为转换剩余时间，当 remaining_time 中的 num_steps 为 0 时，转换也应该立刻生效。此消息在 transition 的每一步结束时都会通知到 App 层，相应的参数中只有 remaining_time 中的 num_steps 会改变。

表 4-4 Generic OnOff Server Set 消息

```
typedef struct
{
    generic_on_off_t on_off;
    generic_transition_time_t total_time;
    generic_transition_time_t remaining_time;
} generic_on_off_server_set_t;
```

5 Generic OnOff Client

5.1 接口

表 5-1 Generic OnOff Client Model 接口

函数	作用
generic_on_off_client_reg	注册 Generic OnOff Client Model
generic_on_off_get	获取远端 element 的 Generic OnOff 状态
generic_on_off_set	设置远端的 Generic OnOff 状态

5.2 消息

5.2.1 GENERIC_ON_OFF_CLIENT_STATUS

当 get 或者 set 远端的开关状态时，远端会返回 status 消息，Model 会把这个消息格式化后通知到 App 层。参数如表 5-2，其中 optional 为 TRUE 时 target_on_off 和 remaining_time 参数有效，反之亦然。

表 5-2 Generic OnOff Client Status 消息

```
typedef struct
{
    generic_on_off_t present_on_off;
    bool optional;
    generic_on_off_t target_on_off;
    generic_transition_time_t remaining_time;
} generic_on_off_client_status_t;
```

6 示例

使用 Generic OnOff Model 时主要有以下步骤：

1. 注册 Model
2. 实现 model_data_cb 函数

6.1 Server Model 示例

表 6-1 Generic OnOff Server Model 示例

```

/* generic onoff server model */
static mesh_model_info_t generic_on_off_server;
static generic_on_off_t current_on_off = GENERIC_ON;

static int32_t generic_on_off_server_data(const mesh_model_info_p pmodel_info, uint32_t type,
                                          void *pargs)
{
    generic_on_off_t current_on_off = GENERIC_OFF;

    switch (type)
    {
        case GENERIC_ON_OFF_SERVER_GET:
        {
            generic_on_off_server_get_t *pdata = pargs;
            pdata->on_off = current_on_off;
        }
        break;
        case GENERIC_ON_OFF_SERVER_SET:
        {
            generic_on_off_server_set_t *pdata = pargs;
            if (0 == pdata->remaining_time.num_steps)
            {
                current_on_off = pdata->on_off;
            }
        }
        break;
        default:
            break;
    }

    return MODEL_SUCCESS;
}

```

```

void genric_onoff_server_models_init(void)
{
    /* register generic onoff server model */
    generic_on_off_server.model_data_cb = generic_on_off_server_data;
    generic_on_off_server_reg(0, &generic_on_off_server);
}

```

6.2 Client Model 示例

表 6-2 Generic OnOff Client Model 示例

```

mesh_model_info_t model_goo_client;

static int32_t generic_on_off_client_data(const mesh_model_info_p pmodel_info,
                                          uint32_t type, void *pargs)
{
    UNUSED(pmodel_info);
    switch (type)
    {
        case GENERIC_ON_OFF_CLIENT_STATUS:
        {
            generic_on_off_client_status_t *pdata = pargs;
            if (pdata->optional)
            {
                data_uart_debug("goo client receive: present = %d, target = %d, remain time = step(%d), \
resolution(%d)\r\n", pdata->present_on_off, pdata->target_on_off,
                             pdata->remaining_time.num_steps, pdata->remaining_time.step_resolution);
            }
            else
            {
                data_uart_debug("goo client receive: present = %d\r\n", pdata->present_on_off);
            }
        }
        break;
    default:
        break;
    }

    return 0;
}

```



```
void generic_client_models_init(void)
{
    model_goo_client.model_data_cb = generic_on_off_client_data;
    generic_on_off_client_reg(0, &model_goo_client);
}
```

Realtek Confidential

参考文献

- [1] [Mesh Profile Specification](#)
- [2] [Mesh Model Specification](#)
- [3] [RTL8762C Mesh SDK User Guide](#)

Realtek Confidential

附录

Realtek Confidential