
# MQTT API DQ_HUB

# CONTENTS

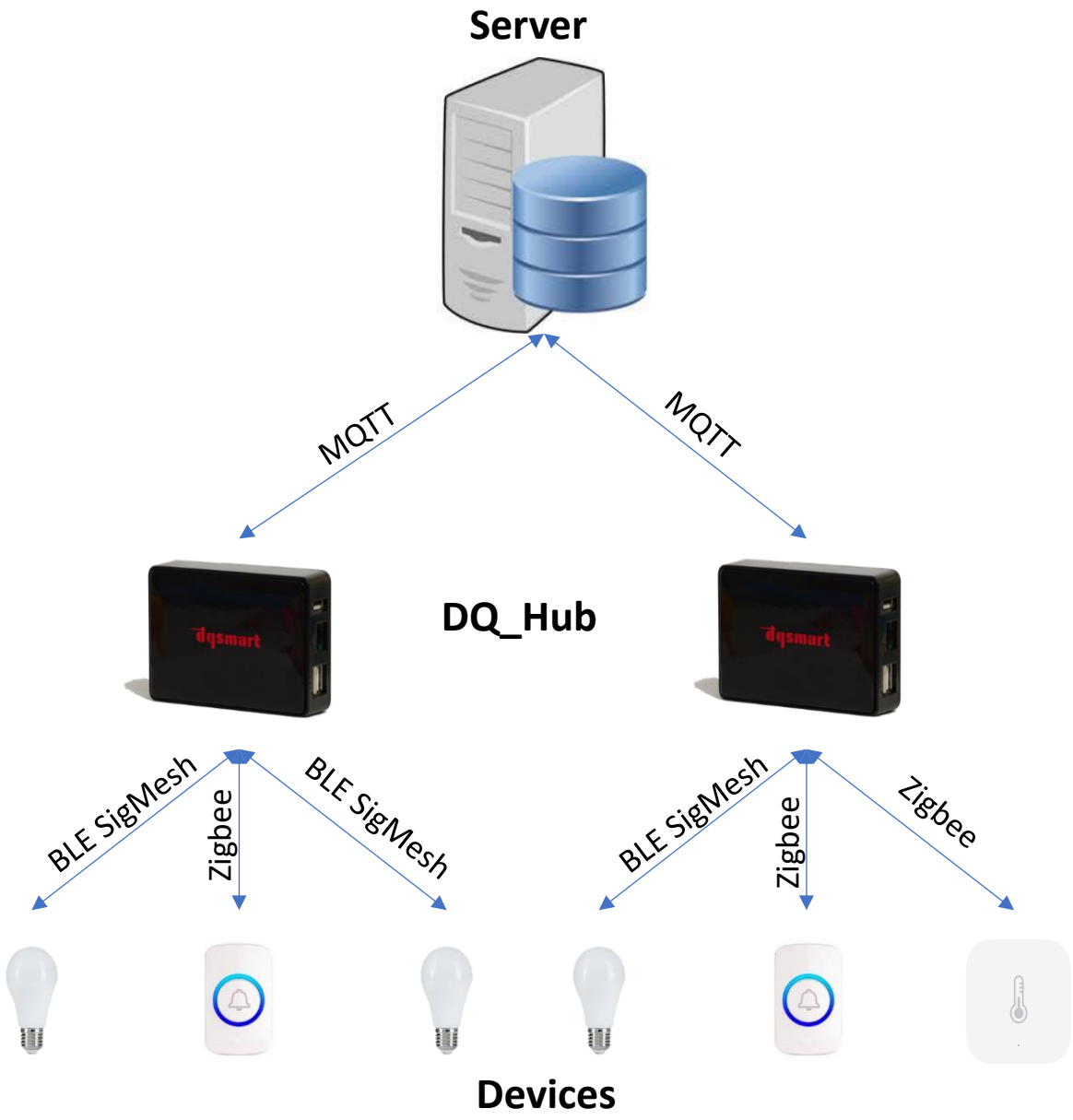
A. INTRODUCE ..... 3
B. API MQTT HUB ..... 5
I. API MQTT HUB WITH BLE SigMesh ..... 5
1. Get list devices from hub ..... 5
2. Get list Group from hub ..... 6
3. Get list Scene ..... 7
4. Set On/Off device or group ..... 8
5. Get On/Off device or group ..... 9
6. Set Lightness device or group ..... 10
7. Get Lightness device or group ..... 10
8. Set HSL ..... 11
9. Call Scene ..... 12
10. Set vendor model ..... 13
11. Get sensor ..... 13
12. Get status devices ..... 14
13. Device status returned ..... 15
II. Zigbee ..... 18
1. Set On off ..... 18
2. Set Lightness ..... 19
3. Get list devices ..... 20
4. Status device telemetry ..... 21
C. SCENE AND SCHEDULE HUB ..... 22
I. Scene Hub ..... 22
1. Creat scene Hub ..... 22
2. Update scene Hub ..... 24
3. Get scene Hub ..... 26
4. Delete scene Hub ..... 27
II. Schedule ..... 27
1. Creat schedule ..... 27
2. Update schedule ..... 29
3. Get schedule ..... 30
4. Delete schedule ..... 31

# A. INTRODUCE

DQ_hub connect BLE SIGMESH device and ZIGBEE device to Server via Internet, using MQTT Protocol.

Hub feature:

- Control and monitoring BLE SigMesh device
  - On/Off a light or group of lights
  - Lightness a light or group of lights
  - Change color a light or group of lights
  - Call scene
  - Save the light of state
  - The light broken detected
- Control and monitoring Zigbee device
  - On/Off a light
  - Lightness a light
  - Read status button
  - Sensor (temperature, humidity, …)
- Schedule a timer
- Auto run: control BLE SigMesh light using zigbee button or zigbee sensor
- Number of control devices is 120 (element + group)



# B. API MQTT HUB

After successfully connecting to the server, the hub sends information to the server every 1 minute

***Example Hub response message:***

Topic: v1/devices/me/telemetry

Payload:

```
{
  "infor": {
    "ip": "172.16.26.112",
    "mac": "02:81:71:87:6d:f4",
    "id": "abc123",
    "version": "1.0.3",
    "name": "hub test 1"
  }
}
```

## I. API MQTT HUB WITH BLE SigMesh

**To send control commands to Hub, server public to topic**:

`v1/devices/me/rpc/request/$request_id`

where $request_id is an integer request identifier

**To get feedback from Hub, server Subscribe to topic:**

`v1/devices/me/rpc/response/$request_id`

### 1. Get list devices from hub

This command to get a list of devices managed by the hub

Max devices is 120 (element + group)

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "get_list_device",
  "params": {
    "type": "ble_sigmesh"
  }
}
```

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "list_device_status",
  "params": {
```

```
    "type": "ble_sigmesh",
    "nodes": [
      {
        "unicastAddress": "0004",
        "name": "TUBE V1.4",
        "type": "light"
      },
      {
        "unicastAddress": "0005",
        "name": "RFID reader",
        "type": "rfid_reader"
      },
      {
        "unicastAddress": "0006",
        "name": "ENGERY TL",
        "type": "energy_sensor"
      },
      {
        "unicastAddress": "0007",
        "name": "DQ SUSEN01SM 2021",
        "type": "vacancy_sensor"
      },
      {
        "unicastAddress": "000C",
        "name": "Motion TL",
        "type": "motion_sensor"
      }
    ]
  }
}
```


- Name: name device
- Type: loại thiết bị (light, sensor,…)
- unicastAddress: device unicast address

## 2. Get list Group from hub

***Example message send to Hub:***

Topic: v1/devices/me/rpc/request/$request_id
Payload:

```
{
  "method": "get_list_groups",
  "params": {
    "type": "ble_sigmesh"
  }
}
```

***Example Hub response message:***

Topic: v1/devices/me/rpc/response/$request_id
Payload:

```
{
  "method": "list_groups_status",
  "params":{
    "type": "ble_sigmesh",
    "groups": [
      {
        "address": "C000",
        "name": "New Group",
        "parentAddress": "0000"
      }
    ]
  }
}
```

## 3. Get list Scene

***Example message send to Hub:***

Topic: v1/devices/me/rpc/request/$request_id
Payload:

```
{
  "method": "get_list_scenes",
  "params": {
    "type": "ble_sigmesh"
  }
}
```

***Example Hub response message:***

Topic: v1/devices/me/rpc/response/$request_id
Payload:

```
{
```

```
  "method": "list_scenes_status",
  "params": {
    "type": "ble_sigmesh",
    "scenes": [
      {
        "addresses": ["0005"],
        "name": "Scene 1",
        "number": "0001"
      },
      {
        "addresses": [],
        "name": "Scene 2",
        "number": "0002"
      },
      {
        "addresses": [],
        "name": "Scene 3",
        "number": "0003"
      }
    ]
  }
}
```


## 4. Set On/Off device or group

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:


```
{
  "method": "set_onOff",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0004",
    "value": 0
  }
}
```


+ unicast: unicast address of the device or group to be controlled
+ value: On = 1 or Off = 0

***Example Hub response message:***
Response received command from server
Topic: `v1/devices/me/rpc/response/$request_id`
Payload:

```
{
  "method": "set_onOff_status",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0004",
    "value": 0,
    "status": "00"
  }
}
```

## 5. Get On/Off device or group

***Example message send to Hub:***
Topic: `v1/devices/me/rpc/request/$request_id`
Payload:

```
{
  "method": "get_onOff",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0004"
  }
}
```

***Example Hub response message***:
Response received command from server
Topic: `v1/devices/me/rpc/response/$request_id`
Payload:

```
{
  "method": "get_onOff_status",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0004",
    "value": 0,
    "status": "00"
  }
}
```

## 6. Set Lightness device or group

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "set_lightness",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "C000",
    "value": 50
  }
}
```

+ unicast: unicast address of the device or group to be controlled
+ value: lightness 0->100%

***Example Hub response message***:

Response received command from server

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "set_lightness_status",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0004",
    "value": 50,
    "status": "00"
  }
}
```

## 7. Get Lightness device or group

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "get_lightness",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0004"
  }
```

```
}
```

***Example Hub response message:***

Response received command from server

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "get_lightness_status",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0004",
    "status": "00"
  }
}
```

## 8. Set HSL

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "set_HSL",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0005",
    "Lightness": 100,
    "Hue": 360,
    "Saturation": 100
  }
}
```

- Unicast: unicast address of the device or group to be controlled
- Lightness: value 0->100
- Hue: value 0->360
- Saturation: value 0->100

***Example Hub response message:***

Response received command from server

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "set_HSL_status",
  "params": {
```

```
    "type": "ble_sigmesh",
    "unicast": "0005",
    "Lightness": 100,
    "Hue": 360,
    "Saturation": 100,
    "status": "00"
  }
}
```

## 9. Call Scene

***Example message send to Hub:***

Response received command from server

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "scene_recall",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0005",
    "SceneNumber": 1
  }
}
```

- Unicast: unicast address of the device or group to be controlled
- SceneNumber: scene number to control

***Example Hub response message:***

Response received command from server

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "scene_recall_status",
  "params": {
    "type": "ble_sigmesh",
    "unicast": "0005",
    "SceneNumber": 1,
    "status": "00"
  }
}
```

## 10. Set vendor model

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
    "method":"set_vendor",
    "params":{
      "type": "ble_sigmesh",
      "unicast":"unicast node",
      "opcode":"lable-opcode | opcode",
      "message":"message( int | bool | str | dict)"
    }
  }
```

***Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "vendor_status",
  "params": {
    "address": "unicast node",
    "name": "sensor name",
    "opcode": "lable-opcode",
    "message": "message( int | bool | str)"
  }
}
```

## 11. Get sensor

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method":"get_sensor",
  "params":{
    "type": "ble_sigmesh",
    "unicast":"unicast node",
    "properties":"properties"
  }
```

```
}
```

***Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method":"sensor_status",
  "params":{
    "address":"0006",
    "<properties>": value
  }
}
```

## 12. Get status devices

Read about the status of all lights (onOff, Lightness, HSL)

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "get_status_devices",
  "params": {
    "type": "ble_sigmesh",
    "address": ["0006","0080"]
  }
}
```

- Address: list of device addresses to get state. If you want to get all devices, then “address”:[]
- The above example gets the status of 2 address lights "0006" and "0080"

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "status_devices",
  "params": {
    "type": "ble_sigmesh",
    "data_bleSigmesh": {
      "0006": {
        "onOffStatus": 1,
        "time": 1681526584.1812994,
        "status": true,
```

```
        "lightnessStatus": 80,
        "HSL_status": {
          "Lightness": 160,
          "Hue": 351,
          "Saturation": 100
        }
      },
      "0070": {
        "onOffStatus": 1,
        "time": 1681526561.7739537,
        "status": true,
        "lightnessStatus": 50,
        "HSL_status": {
          "Lightness": 100,
          "Hue": 351,
          "Saturation": 100
        }
      }
    }
  }
}
```


- Time: device's last state receiving time
- Status: status lights are connected to the network. If >5 minutes without receiving a signal from the device, the status is False (the light is disconnected) otherwise it is True.
- onOffStatus: light status On/Off
- lightnessStatus: lightness
- HSL_status: value HSL

## 13. Device status returned

To listen to the device status returned to the server, you need to subscribe to Topic:
v1/devices/me/telemetry
Payload:

```
[
  "data_bleSigmesh_<unicast>":[
     {
         "method":"<method>",
         "params":{<param>}
     }
```

```
    ]
]
```

<unicast>: device unicast address
<method>:

- **onOffStatus**:
  <param>: {
  “address”: ”<unicast address>”,
  ”value”: <1(on) or 0(off)>
  }

**Example On/Off status**

```
{
    "method": "onOffStatus",
    "params": {
        "address": "0004",
        "value": 1
     }
}
```

- **lightnessStatus**:
  <param>: {
  “address”: ”<unicast address>”,
  ”value”: <int (0->100%)>
  }

**Example Lightness status**

```
{
      "method": "lightnessStatus",
      "params": {
        "address": "0004",
        "value": 100
      }
}
```

- **HSL_status**:
  <param>: {
  “address”: “<unicast address>”,
  ”value”:
  {
  “Lightness”: <int(0->100%)>,

”Hue”: <int(0-360)>,
”Saturation”: <int(0->100)>
}
}

**Example HSL status**

```
{
      "method": "HSL_status",
      "params": {
        "address": "0005",
        "Lightness": 100,
        "Hue": 120,
        "Saturation": 100
      }
}
```

- **scene_status**
  <param>: {
  “address”:”<unicast address>”,
  ” currentScene”: <int>
  }

**Example Scene status**

```
{
      "method": "scene_status",
      "params": {
        "currentScene": 1,
        "address": "0005"
      }
}
```

- **vendor_status**
  <param>: {
  "address": "<unicast node>",
  "name": "<sensor name>",
  "opcode": "<lable-opcode>",
  "message": "message( int | bool | str | dict)"
  }

**Example vendor status**

```
{
```

```
    "method":"vendor_status",
    "params":{
      "address":"0007",
      "name":"vacancy_sensor",
      "opcode":"slots_status",
      "message":true
    }
}
```

- **sensor_status**
  <param>: {
  "address":"<unicast node>",
  "<properties>": <value>
  }

**Example sensor status**

```
{
    "method":"sensor_status",
    "params":{
      "address":"0006",
      "voltage":220,
      "current":0.5,
      "energy":0.5
    }
}
```

- **disconnectedEvent**
  <param>: {}

**Example devices Disconnect**

```
{
      "method": "disconnectedEvent",
      "params": {}
}
```

# II. Zigbee

## 1. Set On off

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "set_onOff",
  "params": {
    "type": "zigbee",
    "address": 12308,
    "DstEndPoint": 2,
    "SrcEndPoint": 1,
    "value": 0
  }
}
```

- Address: device address
- SrcEndPoint: endpoint hub, default = 1
- DstEndPoint: device's joystick endpoint
- Value: joystick status 1/0 (on/off)

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "set_onOff_status",
  "params": {
    "type": "zigbee",
    "address": 12308,
    "DstEndPoint": 2,
    "SrcEndPoint": 1,
    "value": 0
  }
}
```

## 2. Set Lightness

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "set_lightness",
  "params": {
    "type": "zigbee",
    "address": 12308,
```

```
    "DstEndPoint": 2,
    "SrcEndPoint": 1,
    "value": 50
  }
}
```

- Address: device address
- SrcEndPoint: endpoint hub, default = 1
- DstEndPoint: device's joystick endpoint
- Value: Lightness value 0->100%

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "set_lightness_status",
  "params": {
    "type": "zigbee",
    "address": 12308,
    "DstEndPoint": 2,
    "SrcEndPoint": 1,
    "value": 50
  }
}
```

## 3. Get list devices

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "get_list_device",
  "params": {
    "type": "zigbee"
  }
}
```

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "list_device_status",
```

```
  "params": {
    "62986": {
      "name": "CB1 LAU 4 TM",
      "addr": 62986,
      "model": "dqhome.re1"
    },
    "62989": {
      "name": "CB2 LAU 4 TM",
      "addr": 62989,
      "model": "dqhome.re1"
    }
  }
}
```

## 4. Status device telemetry

Update device status automatically from hub to server

```
[
  "data_zigbee_<unicast>":[
     {
        "method":"",
        "params":{}
     }
  ]
]
```

***Example:***

```
[
  "data_zigbee_15439":[
    {
      "method": "lumi.weather",
      "params": {
        "addr": 15439,
        "ep": 1,
        "state": {
            "name": "Relative Humidity",
```

```
            "val": "79", "unit": "%"
          }
       }
    }]
]
```

- Method: device model
- Params:
  - + Addr: device address
  - + Ep: device end point
  - + States:
    - Name:
      - Relative Humidity
      - Temperature
      - Status (button)
      - lightness: 0->100%
      - Battery percent: 0->100%
      - Battery voltage
    - Val: value
    - Unit

# C. SCENE AND SCHEDULE HUB

Setup shared scenarios for BLE and Zigbee or set a timer

## I. Scene Hub

### 1. Creat scene Hub

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "creat_scene_hub",
  "params": [
    {
      "comparison": "less"|"less_equal"|"greater"|"greater
_equal"|"equal",
      "enable":true,
      "id": "",
      "input": {
        "method": "status",
        "params": {
```

```
          "addr": 9959,
          "state": {"name": "Status", "val": "ON"}
        }
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "0004",
          "value": 1
        }
      }]
    }
  ]
}
```

- enable: true(ON) or false(OFF)
- id: id scene
- input: this is the return value of Zigbee
- output: this is the return value of BLE or Zigbee
- Comparison:
  - Less: less than (<)
  - less_equal: less than or equal (<=)
  - greater: greater than (>)
  - greater_equal: greater than or equal (>=)
  - equal (==)

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "creat_scene_hub_status",
  "params": [
    {
      "comparison": "less"|"less_equal"|"greater"|"greater
_equal"|"equal",
      "enable":true,
      "id": "",
      "input": {
```

```
        "method": "status",
        "params": {
          "addr": 9959,
          "state": {"name": "Status", "val": "ON"}
        }
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "0004",
          "value": 1
        }
      }]
    }
  ]
}
```

## 2. Update scene Hub

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "update_scene_hub",
  "params": [
    {
      "comparison": "less"|"less_equal"|"greater"|"greater
_equal"|"equal",
      "enable": true,
      "id": "",
      "input": {
        "method": "status",
        "params": {
          "addr": 9959,
          "state": {"name": "Status", "val": "ON"}
        }
      },
```

```
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "0004",
          "value": 1
        }
      }]
    }
  ]
}
```

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "update_scene_hub_status",
  "params": [
    {
      "comparison": "less"|"less_equal"|"greater"|"greater
_equal"|"equal",
      "enable": true,
      "id": "",
      "input": {
        "method": "status",
        "params": {
          "addr": 9959,
          "state": {"name": "Status", "val": "ON"}
        }
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "0004",
          "value": 1
        }
```

```
      }]
    }
  ]
}
```

### 3. Get scene Hub

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "get_scene_hub",
  "params": ""
}
```

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "scene_hub",
  "params": [
    {
      "comparison": "less"|"less_equal"|"greater"|"greater
_equal"|"equal",
      "id": "1",
      "enable": true,
      "input": {
        "method": "lumi.remote.b1acn01",
        "params": {
          "addr": 24224,
          "ep": 1,
          "state": {"name": "Status", "val": 1}
        }
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "0004",
          "value": 1
```

```
        }
      }]
    }
  ]
}
```

### 4. Delete scene Hub

Delete 1 or more scenes

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "delete_scene_hub",
  "params": ["id1", "id2"]
}
```

- ["id1", "id2"]: “id1” , “id2” are the ids of the scene to be deleted. If you want to delete all here, leave the array empty (“param”:[]).

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "delete_scene_hub_status",
  "params": ["id1", "id2"]
}
```

## II. Schedule

### 1. Creat schedule

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "creat_schedule_hub",
  "params": [
    {
      "enable": true,
      "id": "",
      "input": {
        "times": "00:00",
        "weekdays": [0, 1, 2, 3, 4, 5, 6]
```

```
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "C000",
          "value": 0
        }
      }]
    }
  ]
}
```

- enable: true(ON) or false(OFF)
- id: schedule id
- input: times
- times: time to set (in 24h format)
- weekdays: Day of the week is 0->6 Monday -> Sunday respectively
- output: joystick value if input is valid

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "creat_schedule_hub_status",
  "params": [
    {
      "enable":true,
      "id": "",
      "input": {
        "times": "00:00",
        "weekdays": [0, 1, 2, 3, 4, 5, 6]
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "C000",
          "value": 0
```

```
        }
      }]
    }
  ]
}
```

## 2. Update schedule

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "update_schedule_hub",
  "params": [
    {
      "enable":true,
      "id": "",
      "input": {
        "times": "00:00",
        "weekdays": [0, 1, 2, 3, 4, 5, 6]
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "C000",
          "value": 0
        }
      }]
    }
  ]
}
```

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "update_schedule_hub_status",
  "params": [
    {
```

```
      "enable":true,
      "id": "",
      "input": {
        "times": "00:00",
        "weekdays": [0, 1, 2, 3, 4, 5, 6]
      },
      "output": [{
        "method": "set_onOff",
        "params": {
          "type": "ble_sigmesh",
          "unicast": "C000",
          "value": 0
        }
      }]
    }
  ]
}
```


### 3. Get schedule

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:


```
{
  "method": "get_schedule_hub",
  "params": ""
}
```


***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:


```
{
  "method": "schedule_hub",
  "params": [
    {
      "enable": true,
      "id": "off",
      "input": { "times": "08:45", "weekdays": [0, 1, 2, 3, 4,
5, 6] },
      "output": [{
```

```
        "method": "set_onOff",
        "params": { "type": "ble_sigmesh", "unicast": "C000", "
value": 0 }
      }]
    },
    {
      "enable": true,
      "id": "on",
      "input": { "times": "08:43", "weekdays": [0, 1, 2, 3, 4,
5, 6] },
      "output": [{
        "method": "set_onOff",
        "params": { "type": "ble_sigmesh", "unicast": "C000", "
value": 1 }
      }]
    }
  ]
}
```

## 4. Delete schedule

***Example message send to Hub:***

Topic: `v1/devices/me/rpc/request/$request_id`

Payload:

```
{
  "method": "delete_schedule_hub",
  "params": ["id1", "id2"]
}
```

- ["id1", "id2"]: “id1” , “id2” are the ids of the scene to be deleted. If you want to delete all here, leave the array empty (“param”:[]).

***Example Hub response message:***

Topic: `v1/devices/me/rpc/response/$request_id`

Payload:

```
{
  "method": "delete_schedule_hub_status",
  "params": ["id1", "id2"]
}
```