

Trimlight V2 Oauth API Documentation

Version	Description	Date
1.0.0	initial version.	2021-04-20
1.1.0	Add <code>currentEffect</code> data to the device details response data.	2021-11-22
1.2.0	Add group function APIs. (19 - 23)	2021-12-04
1.2.1	Add the "sync a group" API. (24)	2021-12-08
1.3.0	Add the "Notify update shadow data" API. (25)	2022-08-18

1. Auth

Each request needs to carry the following request headers

```
"authorization": "<accessToken>"
"S-ClientId": "<clientId>"
"S-Timestamp": 111 // Timestamp (milliseconds relative to 1970.1.1)
```

You need to follow the steps below to calculate the **accessToken**:

1. Concatenate strings: "Trimlight|<S-ClientId>|<S-Timestamp>"
2. Compute the **HMAC-SHA256** of the string concatenated in step 1. The secret key for HMAC-SHA256 is **clientSecret**.
3. The value of the **accessToken** is the base64 encoding of the computed HMAC-SHA256 value.

Please contact our business to obtain [clientId] and [clientSecret]

For example:

clientId: tester, clientSecret: test_secret, timestamp: 1713166849256

The concatenated string is: "Trimlight|tester|1713166849256", and the base64 of HMAC-SHA256 value is: "z02N77XySuOww5OSUe0vrPwprRITb656xKPuIS9ooXI=", and that is the access token.

2. Base URL

```
POST https://trimlight.ledhue.com/trimlight
```

3. Get device list

```
GET /v1/oauth/resources/devices
```

Request params:

```
{
  "page": 1 // 10 devices on one page
            // If the value is 0 or null, it will return to the list of all devices
}
```

Response result:

```
{
  "code": 0,
  "desc": "success",
  "payload": {
    "total": 2,
    "current": 1,
    "data": [
      {
        "deviceId": "xxxxxxxxxx1",
        "name": "xxxx1",
        "switchState": 0,
        "connectivity": 1,
        "state": 0,
        "fwVersionName": "1.1.1"
      },
      {
        "deviceId": "xxxxxxxxxx2",
        "name": "xxxx2",
        "switchState": 0,
        "connectivity": 1,
        "state": 0,
        "fwVersionName": "1.1.1"
      }
    ]
  }
}
```

Response result code details

Field	Description	Type
<code>code</code>	Result code. [1]	Integer
<code>desc</code>	Result decription.	String
<code>payload</code>	Result payload.	Object

Page fields details

Field	Description	Type
<code>total</code>	Number of all devices.	Integer
<code>page</code>	The page number of the current data.	Integer
<code>data</code>	device list.	List

Device fields details

Field	Description	Type
<code>deviceId</code>	Unique ID of the device.	String
<code>name</code>	Device name.	String
<code>switchState</code>	Device switch state. 0 : light off. 1 : manual mode. 2 : timer mode.	Integer
<code>connectivity</code>	Device connectivity state. 0 : offline. 1 : online.	Integer
<code>state</code>	Device state. 0 : normal. 1 : upgrading.	Integer
<code>fwVersionName</code>	Device firmware version name.	String

4. Get device detail data

POST `/v1/oauth/resources/device/get`

Request params:

```

{
  "deviceId": "<device-id>",
  "currentDate": {
    "year": 21, // 2021
    "month": 1,
    "day": 1,
    "weekday": 1,
    "hours": 1,
    "minutes": 1,
    "seconds": 1
  }
}

```

Current date details

Field	Description	Type
year	Current date year. Years relative to 2000.	Integer
month	Current date month. Range: [1, 12].	Integer
day	Current date day. Range: [1, 31].	Integer
weekday	Current day of week. SUNDAY = 1, MONDAY = 2, TUESDAY = 3, WEDNESDAY = 4, THURSDAY = 5, FRIDAY = 6, SATURDAY = 7	Integer
hours	Current date year. Range: [0, 23].	Integer
minutes	Current date year. Range: [0, 59].	Integer
seconds	Current date year. Range: [0, 59].	Integer

Response result:

```

{
  "code": 0,
  "desc": "success",

```

```
"payload": {
  "name": "xxxx2",
  "switchState": 0,
  "connectivity": 1,
  "state": 0,
  "colorOrder": 0,
  "ic": 0,
  "ports": [
    {
      "id": 0,
      "start": 1,
      "end": 1024
    },
    {
      "id": 1,
      "start": 1,
      "end": 1024
    },
    {
      "id": 2,
      "start": 1,
      "end": 1024
    },
    {
      "id": 3,
      "start": 1,
      "end": 1024
    }
  ],
  "fwVersionName": "1.1.1",
  "effects": [
    {
      "id": 0,
      "name": "New Year",
      "category": 0,
      "mode": 0,
      "speed": 100,
      "brightness": 100,
      "pixelLen": 30,
      "reverse": false
    },
    {
      "id": 1,
      "name": "xxxxxxxx2",
      "category": 1,
      "mode": 0,
      "speed": 100,
      "brightness": 100,
      "pixels": [
        {
          "index": 0,
          "count": 5,
```

```
        "color": 16711680,
        "disable": false
    },
    {
        "index": 1,
        "count": 10,
        "color": 65280,
        "disable": false
    },
    {
        "index": 2,
        "count": 10,
        "color": 255,
        "disable": false
    }
]
}
],
"combinedEffect": {
    "effectIds": [0, 2, 3],
    "interval": 5
},
"daily": [
    {
        "id": 0,
        "enable": true,
        "effectId": 0,
        "repetition": 1,
        "startTime": {
            "hours": 10,
            "minutes": 1
        },
        "endTime": {
            "hours": 11,
            "minutes": 1
        }
    },
    {
        "id": 1,
        "enable": true,
        "effectId": 1,
        "repetition": 1,
        "startTime": {
            "hours": 10,
            "minutes": 1
        },
        "endTime": {
            "hours": 11,
            "minutes": 1
        }
    }
]
],
```

```
"calendar": [  
  {  
    "id": 0,  
    "effectId": 1,  
    "startDate": {  
      "month": 12,  
      "day": 31  
    },  
    "endDate": {  
      "month": 1,  
      "day": 1  
    },  
    "startTime": {  
      "hours": 10,  
      "minutes": 1  
    },  
    "endTime": {  
      "hours": 11,  
      "minutes": 1  
    }  
  },  
  {  
    "id": 1,  
    "effectId": 2,  
    "startDate": {  
      "month": 12,  
      "day": 31  
    },  
    "endDate": {  
      "month": 1,  
      "day": 1  
    },  
    "startTime": {  
      "hours": 10,  
      "minutes": 1  
    },  
    "endTime": {  
      "hours": 11,  
      "minutes": 1  
    }  
  }  
],  
  "currentEffect": {  
    "category": 1,  
    "mode": 1,  
    "speed": 174,  
    "brightness": 204,  
    "pixelLen": 37,  
    "reverse": false  
  }  
}
```

4.1 Device fields details

Field	Description	Type
name	Device name.	String
switchState	Device switch state. 0 : light off. 1 : manual mode. 2 : timer mode.	Integer
connectivity	Device connectivity state. 0 : offline. 1 : online.	Integer
state	Device state. 0 : normal. 1 : upgrading.	Integer
colorOrder	Color order. [2].	Integer
ic	IC. [3]	Integer
ports	The pixel setting for each port. id: Port ID, range from 0 to 3, correspond to port1, port2, port3, port4. start: The start pixel of the port. range: [1, 2048] end: The end pixel of the port. range: [1, 2048]. "start" should not greater than "end".	List
fwVersionName	Device firmware version name.	String
effects	All the effects stored in the device.	List
combinedEffect	Combine effect.	Object
daily	Daily schedules. Each device has two daily schedules.	List
calendar	Calendar schedules.	List
currentEffect	Device current running effect. (If the device's switch state is timer mode, although the light is off at this time, it will return the last running effect data.) Note: The effect ID will be -1, when the controller is running a preview effect (not yet saved).	Object

4.2 Effect fields details

Field	Description	Type
<code>id</code>	Effect ID. ID of the saved effect. Note that it may be -1 in <code>currentEffect</code>, when the controller is running a preview effect (not yet saved).	Integer
<code>category</code>	Effect category. 0 : build-in effect. 1 : custom effect.	Integer
<code>mode</code>	Effect mode. Build-in effect (category value is 0) mode range: [0, 179]. [4] Custom effect (category value is 1) mode range: [0, 16]. [5]	Integer
<code>speed</code>	Effect speed. Speed range: [0, 255].	Integer
<code>brightness</code>	Effect brightness. Brightness range: [0, 255].	Integer
<code>pixelLen</code>	Effect pixel length. (Only required for build-in effects) Pixel length range: [1, 90].	Integer
<code>reverse</code>	Reverse effect. (Only required for build-in effects)	Boolean
<code>pixels</code>	Custom effect pixels. (Only required for build-in effects)	List

4.3 Pixel fields details

Field	Description	Type
<code>index</code>	Custom effect pixel index.	Integer
<code>count</code>	Custom effect pixel count.	Integer
<code>color</code>	Custom effect pixel color.	Integer
<code>disable</code>	Disable pixel.	Boolean

4.4 Combined effect fields details

Field	Description	Type
<code>effectIds</code>	A list of each effects' ID in the combined effect.	List
<code>interval</code>	The interval between switching to the next effect. (Unit: minute.)	Integer

4.5 Daily schedule fields details

Field	Description	Type
<code>id</code>	Daily schedule ID. 0: daily schedule 1. 1: daily schedule 2.	List
<code>enable</code>	Enable daily schedule.	Boolean
<code>effectId</code>	Effect ID in the daily schedule.	Integer
<code>repetition</code>	The repetition of daily schedule.	Integer
<code>startTime</code>	The start time of daily schedule. See schedule time fields details.	Object
<code>endTime</code>	The end time of daily schedule. See schedule time fields details.	Object

4.6 Schedule Time fields details

Field	Description	Type
<code>hours</code>	Schedule hours, range from [1, 24].	Integer
<code>minutes</code>	Schedule minutes, range from [1, 60].	Integer

4.7 Calendar schedule fields details

Field	Description	Type
<code>id</code>	Calendar schedule ID. Schedule ID range: [0, 59]. Up to 60 calendar schedules can be saved.	List
<code>effectId</code>	Effect ID in the calendar schedule.	Integer
<code>startDate</code>	The start date of calendar schedule. See schedule date fields details.	Object
<code>endDate</code>	The end date of calendar schedule. See schedule date fields details.	Object
<code>startTime</code>	The start time in each day of calendar schedule. See schedule time fields details.	Object
<code>endTime</code>	The end time in each day of calendar schedule. See schedule time fields details.	Object

4.8 Schedule date fields details

Field	Description	Type
<code>month</code>	Schedule month, range from [1, 12].	Integer
<code>day</code>	Schedule day, range from [1, 31].	Integer

4.9 Effect fields details

Field	Description	Type
<code>category</code>	Effect category. 0 : build-in effect. 1 : custom effect.	Integer
<code>mode</code>	Effect mode. Build-in effect (category value is 0) mode range: [0, 179]. [4] Custom effect (category value is 1) mode range: [0, 16]. [5]	Integer
<code>speed</code>	Effect speed. Speed range: [0, 255].	Integer
<code>brightness</code>	Effect brightness. Brightness range: [0, 255].	Integer
<code>pixelLen</code>	Effect pixel length. (Only available for build-in effects) Pixel lenght range: [1, 90].	Integer
<code>reverse</code>	Reverse effect. (Only available for build-in effects)	Boolean
<code>pixels</code>	Custom effect pixels. (Only available for build-in effects)	List

5. Set device switch state

POST `/v1/oauth/resources/device/update`

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "switchState": 0
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

6. Set device name

```
POST /v1/oauth/resources/device/update
```

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "name": "xxx"
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

7. Set device color order

```
POST /v1/oauth/resources/device/update
```

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "colorOrder": 0
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

8. Set device IC

POST /v1/oauth/resources/device/update

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "ic": 0
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

9. Set device port

POST /v1/oauth/resources/device/update

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "ports": [
      {
        "id": 0, // [0, 3]
        "start": 1,
        "end": 1024
      }
    ]
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

10. Preview build-in effect

```
POST /v1/oauth/resources/device/effect/preview
```

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "category": 0,
    "mode": 0,
    "speed": 100,
    "brightness": 100,
    "pixelLen": 10,
    "reverse": false
  }
}
```

Effect fields details

Field	Description	Type
category	Effect category. 0 : build-in effect. 1 : custom effect. Here the value is 0.	Integer
mode	Build-in effect mode. Build-in effect mode range: [0, 179]. [4]	Integer
speed	Effect speed. Speed range: [0, 255].	Integer
brightness	Effect brightness. Brightness range: [0, 255].	Integer
pixelLen	Effect pixel length. Pixel lenght range: [1, 90].	Integer
reverse	Reverse effect.	Boolean

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

Example

Build-in effect

```
{
  "deviceId": "<device-id>",
  "payload": {
    "category": 1,
    "mode": 1,
    "speed": 100,
    "brightness": 100,
    "pixelLen": 10,
    "reverse": false
  }
}
```

11. Preview custom effect

```
POST /v1/oauth/resources/device/effect/preview
```

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "category": 1,
    "mode": 0,
    "speed": 100,
    "brightness": 100,
    "pixels": [
      {
        "index": 0,
        "count": 5,
        "color": 16711680, // (0xFF0000)
        "disable": false
      },
      {
        "index": 1,
        "count": 10,

```

```

        "color": 16711680,
        "disable": false
    }
}
}
}

```

Effect fields details

Field	Description	Type
<code>category</code>	Effect category. 0 : build-in effect. 1 : custom effect. Here the value is 1.	Integer
<code>mode</code>	Effect mode. Custom effect mode range: [0, 16]. [5]	Integer
<code>speed</code>	Effect speed. Speed range: [0, 255].	Integer
<code>brightness</code>	Effect brightness. Brightness range: [0, 255].	Integer
<code>pixels</code>	Custom effect pixels.	List

Pixel fields details

Field	Description	Type
<code>index</code>	Custom pixel index. Index range: [0, 29].	Integer
<code>count</code>	Custom pixel count. Pixel count range: [0, 60].	Integer
<code>color</code>	Custom pixel color decimal value. eg: 0xFF0000 => 16711680.	Integer
<code>disable</code>	Custom pixel disable.	Boolean

Example

Custom effect

```

{
  "deviceId": "<device-id>",
  "payload": {

```



```

    "category": 2,
    "mode": 1,
    "speed": 100,
    "brightness": 100,
    "pixels": [
      {
        "index": 0,
        "count": 10,
        "color": 255,
        "disable": false
      },
      {
        "index": 1,
        "count": 2,
        "color": 0,
        "disable": true
      },
      {
        "index": 2,
        "count": 3,
        "color": 65280,
        "disable": false
      },
      {
        "index": 2, // The same index will be overwritten
        "count": 3,
        "color": 16711680,
        "disable": false
      }
    ]
  }
}

```

12. Add/Update effect

POST /v1/oauth/resources/device/effect/save

Request params:

```

{
  "deviceId": "<device-id>",
  "payload": {
    "id": -1
    "name": "xxxx",
    "category": 1/2,
    "mode": 0,
    "speed": 100,
    "brightness": 100,

```

```

    "pixelLen": 10,
    "reverse": false,
    "pixels": [
      {
        "index": 0,
        "count": 5,
        "color": 16711680, // (0xFF0000)
        "disable": false
      },
      {
        "index": 1,
        "count": 10,
        "color": 16711680,
        "disable": false
      }
    ]
  }
}

```

Effect fields details

Field	Description	Type
<code>id</code>	Effect ID. If ID is -1 or null, it will be saved as a new effect, if not, the effect corresponding to the ID will be updated. Up to 60 effects can be saved.	Integer
<code>category</code>	Effect category. 0 : build-in effect. 1 : custom effect.	Integer
<code>mode</code>	Effect mode. Custom effect mode range: [0, 16]. [5]	Integer
<code>speed</code>	Effect speed. Speed range: [0, 255].	Integer
<code>brightness</code>	Effect brightness. Brightness range: [0, 255].	Integer
<code>pixelLen</code>	Effect pixel length. Pixel lenght range: [1, 90]. (Only required for build-in effects)	Integer
<code>reverse</code>	Reverse effect. (Only required for build-in effects)	Boolean
<code>pixels</code>	Custom effect pixels. See pixel fields details. (Only required for custom effects)	List

Response result:

```
{
  "code": 0,
  "desc": "success",
  "payload": {
    "id": 10    // effect id
  }
}
```

13. Check out effect

```
POST /v1/oauth/resources/device/effect/view
```

Request params:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 0
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

14. Delete effect

```
POST /v1/oauth/resources/device/effect/delete
```

Request prams:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 0
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

15. Update daily schedule

POST /v1/oauth/resources/device/daily/save

Request prams:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 1,
    "enable": true,
    "effectId": 1,
    "repetition": 1,
    "startTime": {
      "hours": 10,
      "minutes": 1
    },
    "endTime": {
      "hours": 11,
      "minutes": 1
    },
    "currentDate": {
      "month": 1,
      "day": 1
    }
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

16. Add/Update calendar schedule

```
POST /v1/oauth/resources/device/calendar/save
```

Request prams:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 0,
    "effectId": 1, // The combined effect id is fixed at 200
    "startDate": {
      "month": 12,
      "day": 31
    },
    "endDate": {
      "month": 1,
      "day": 1
    },
    "startTime": {
      "hours": 10,
      "minutes": 1
    },
    "endTime": {
      "hours": 11,
      "minutes": 1
    }
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success",
  "payload": {
    "id": 10 // calendar id
  }
}
```

17. Delete calendar schedule

```
POST /v1/oauth/resources/device/calendar/delete
```

Request prams:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "id": 0
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

18. Set combined effect

POST /v1/oauth/resources/device/combined-effect/save

Request prams:

```
{
  "deviceId": "<device-id>",
  "payload": {
    "effectIds": [0, 2, 3], // up to 5 effects
    "interval": 5
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

19. Get group list

GET /v1/oauth/resources/groups

Request params:

```
{
  "page": 1  // 10 groups on one page
            // If the value is 0 or null, it will return to the list of all groups
}
```

Response result:

```
{
  "code": 0,
  "desc": "success",
  "payload": {
    "total": 2,
    "current": 1,
    "data": [
      {
        "groupId": "xxx",
        "name": "group1",
        "masterDevice": {
          "deviceId": "<device_id>",
          "name": "<device_name>"
        },
        "devices": [
          {
            "deviceId": "<device_id>",
            "name": "<device_name>"
          }
        ]
      },
      {
        "groupId": "xxx",
        "name": "group2",
        "masterDevice": {
          "deviceId": "<device_id>",
          "name": "<device_name>"
        },
        "devices": [
          {
            "deviceId": "<device_id>",
            "name": "<device_name>"
          }
        ]
      }
    ]
  }
}
```

20. Add a new group

```
GET /v1/oauth/resources/group/add
```

Request params:

```
{
  "name": "<new group name>",
  "masterDevice": "<master device ID>",
  "devices": [
    "<master device ID>",
    "<device ID>",
    "<device ID>"
  ]
}
```

Response result:

```
{
  "code": 0,
  "desc": "success",
  "payload": {
    "groupId": "<new group ID>"
  }
}
```

21. Update a group

```
GET /v1/oauth/resources/group/update
```

Request params:

```
{
  "groupId": "<group ID>",
  "masterDevice": "<master device ID>",
  "devices": [
    "<master device ID>",
    "<device ID>",
    "<device ID>"
  ]
}
```

Response result:


```
{
  "code": 0,
  "desc": "success",
  "payload": {
    "groupId": "<new group ID>"
  }
}
```

22. Rename a group

GET /v1/oauth/resources/group/rename

Request params:

```
{
  "groupId": "<group ID>",
  "name": "<new group name>"
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

23. Delete a group

GET /v1/oauth/resources/group/delete

Request params:

```
{
  "groupId": "<group ID>"
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

24. Sync a group

```
GET /v1/oauth/resources/group/sync
```

Request params:

```
{
  "groupId": "<group ID>"
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

PS: The master device must be online.

25. Notify update shadow data

Before requesting detailed data for the device, you can send this request to notify the device to report the latest shadow data.

```
GET /v1/oauth/resources/device/notify-update-shadow
```

Request params:

```
{
  "deviceId": "<device-id>",
  "currentDate": {
    "year": 21, // 2021
    "month": 1,
    "day": 1,
    "weekday": 1,
    "hours": 1,
    "minutes": 1,
    "seconds": 1
  }
}
```

Response result:

```
{
  "code": 0,
  "desc": "success"
}
```

Appendix

[1] Result code

Code	Description
0	Success.
10001	Error.
10002	Wrong password.

[2] Color order

Color order	Value
RGB	0
RBG	1
GRB	2
GBR	3
BRG	4
BGR	5

[3] IC

IC	Value
UCS1903	0
DMX512	1

[4] Build-in effect mode

Color order	Value
Rainbow Gradual Chase	0
Rainbow Comet	1
Rainbow Segment	2
Rainbow Wave	3
Rainbow Meteor	4
Rainbow Gradual	5
Rainbow Jump	6
Rainbow Stars	7
Rainbow Fade In Out	8
Rainbow Spin	9
Red Stacking	10
Green Stacking	11

Blue Stacking	12
Yellow Stacking	13
Cyan Stacking	14
Purple Stacking	15
White Stacking	16
Full Color Stack	17
Red to Green Stack	18
Green to Blue Stack	19
Blue to Yellow Stack	20
Yellow to Cyan Stack	21
Cyan to Purple Stack	22
Purple to White Stack	23
Red Comet	24
Green Comet	25
Blue Comet	26
Yellow Comet	27
Cyan Comet	28
Purple Comet	29
White Comet	30
Red Meteor	31
Green Meteor	32
Blue Meteor	33
Yellow Meteor	34
Cyan Meteor	35
Purple Meteor	36
White Meteor	37
Red Wave	38
Green Wave	39

Blue Wave	40
Yellow Wave	41
Cyan Wave	42
Purple Wave	43
White Wave	44
Red Green Wave	45
Red Blue Wave	46
Red Yellow Wave	47
Red Cyan Wave	48
Red Purple Wave	49
Red White Wave	50
Green Blue Wave	51
Green Yellow Wave	52
Green Cyan Wave	53
Green Purple Wave	54
Green White Wave	55
Blue Yellow Wave	56
Blue Cyan Wave	57
Blue Purple Wave	58
Blue White Wave	59
Yellow Cyan Wave	60
Yellow Purple Wave	61
Yellow White Wave	62
Cyan Purple Wave	63
Cyan White Wave	64
Purple White Wave	65
Red Dot Pulse	66
Green Dot Pulse	67
Blue Dot Pulse	68

Yellow Dot Pulse	69
Cyan Dot Pulse	70
Purple Dot Pulse	71
White Dot Pulse	72
Red Green Blank Pulse	73
Green Blue Blank Pulse	74
Blue Yellow Blank Pulse	75
Yellow Cyan Blank Pulse	76
Cyan Purple Blank Pulse	77
Purple White Blank Pulse	78
Red with Purple Pulse	79
Green with Cyan Pulse	80
Blue with Yellow Pulse	81
Yellow with Blue Pulse	82
Cyan with Green Pulse	83
Purple with Purple Pulse	84
Red Comet Spin	85
Green Comet Spin	86
Blue Comet Spin	87
Yellow Comet Spin	88
Cyan Comet Spin	89
Purple Comet Spin	90
White Comet Spin	91
Red Dot Spin	92
Green Dot Spin	93
Blue Dot Spin	94
Yellow Dot Spin	95
Cyan Dot Spin	96

Purple Dot Spin	97
White Dot Spin	98
Red Segment Spin	99
Green Segment Spin	100
Blue Segment Spin	101
Yellow Segment Spin	102
Cyan Segment Spin	103
Purple Segment Spin	104
White Segment Spin	105
Red Green Gradual Snake	106
Red Blue Gradual Snake	107
Red Yellow Gradual Snake	108
Red Cyan Gradual Snake	109
Red Purple Gradual Snake	110
Red White Gradual Snake	111
Green Blue Gradual Snake	112
Green Yellow Gradual Snake	113
Green Cyan Gradual Snake	114
Green Purple Gradual Snake	115
Green White Gradual Snake	116
Blue Yellow Gradual Snake	117
Blue Cyan Gradual Snake	118
Blue Purple Gradual Snake	119
Blue White Gradual Snake	120
Yellow Cyan Gradual Snake	121
Yellow Purple Gradual Snake	122
Yellow White Gradual Snake	123
Cyan Purple Gradual Snake	124
Cyan White Gradual Snake	125

Purple White Gradual Snake	126
Red White Blank Snake	127
Green White Blank Snake	128
Blue White Blank Snake	129
Yellow White Blank Snake	130
Cyan White Blank Snake	131
Purple White Blank Snake	132
Green Yellow White Snake	133
Red Green White Snake	134
Red Yellow Snake	135
Red White Snake	136
Green White Snake	137
Red Stars	138
Green Stars	139
Blue Stars	140
Yellow Stars	141
Cyan Stars	142
Purple Stars	143
White Stars	144
Red Background Stars	145
Green Background Stars	146
Blue Background Stars	147
Yellow Background Stars	148
Cyan Background Stars	149
Purple Background Stars	150
Red White Background Stars	151
Green White Background Stars	152
Blue White Background Stars	153

Yellow White Background Stars	154
Cyan White Background Stars	155
Purple White Background Stars	156
White White Background Stars	157
Red Breath	158
Green Breath	159
Blue Breath	160
Yellow Breath	161
Cyan Breath	162
Purple Breath	163
White Breath	164
Red Yellow Fire	165
Red Purple Fire	166
Green Yellow Fire	167
Green Cyan Fire	168
Blue Purple Fire	169
Blue Cyan Fire	170
Red Strobe	171
Green Strobe	172
Blue Strobe	173
Yellow Strobe	174
Cyan Strobe	175
Purple Strobe	176
White Strobe	177
Red Blue White Strobe	178
Full Color Strobe	179

[5] Custom effect mode

Color order	Value
STATIC	0
CHASE FORWARD	1
CHASE BACKWARD	2
CHASE MIDDLE TO OUT	3
CHASE OUT TO MIDDLE	4
STARS	5
BREATH	6
COMET FORWARD	7
COMET BACKWARD	8
COMET MIDDLE TO OUT	9
COMET OUT TO MIDDLE	10
WAVE FORWARD	11
WAVE BACKWARD	12
WAVE MIDDLE TO OUT	13
WAVE OUT TO MIDDLE	14
STROBE	15
SOLID FADE	16

[6] Repitition

Repitition	数值
Today Only	0
Everyday	1
Week Days	2
Weekend	3