
Yeelink™

Yeelight Sunflower GW API

Version <1.0>

\

Revision History

Date	Revision	Summary	Author
2013-12-24	1.0	Initial version	Gordon Tang

Catalogue

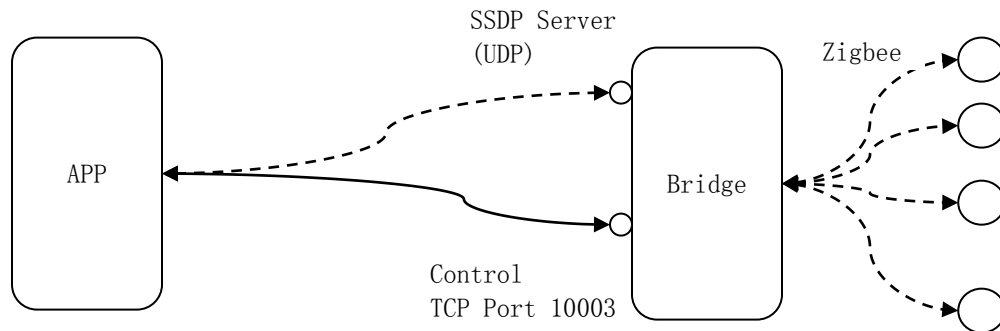
1. Overview	4
2. Architecture	4
3. Design	4
3.1 Communicate Mechanism	4
3.1.1 Device Discover Mechanism	4
3.1.2 Control Mechanism	5
3.2 Message Interface	5
3.2.1 Overview	5
3.2.2 HB/HACK	6
3.2.3 GL/GLB	6
3.2.4 C/CB	6
3.2.5 T/TB	7
3.2.6 RT/RTB	7
3.2.7 S	7
3.2.8 NEW	8

1. Overview

Yeelight Sunflower provides the smart control for LED with mobile App, including LED, Bridge (Also called Control box) and APP. Bridge and mobile connect to the same LAN in the home so that Bridge could forward the message from APP to the LED.

2. Architecture

The first version only covers the LAN mode. APP communicates with Bridge via TCP/IP as below.



After Bridge is initiated, APP could discover it with SSDP M-SEARCH. Bridge will respond with 200 OK if receives any SSDP packet. After that, APP sends "GL" to get the LED info from Bridge.

After that, APP is able to send control message(C/T/RT), and Bridge responds with the message(CB/TB/RTB). Furthermore, APP could receive status message(S/NEW) or send Heartbeat message((HB/HACK).

3. Design

3.1 Communicate Mechanism

3.1.1 Device Discover Mechanism

Bridge will listen on UDP port 1900. APP will unicast SSDP packet via address 239.255.255.250 to search the Bridge. If any Bridge receives this, it can return back the message 200 OK with its local info such as IP address.

```
APP                Bridge
|----- M-SEARCH ----->|
|<----- 200 OK -----|
```

APP SSDP Message Example:

```
M-SEARCH * HTTP/1.1\r\n
HOST:239.255.255.250:1900\r\n
MAN:{"ssdp:discover"}\r\n
ST:yeelink:yeebox\r\n
MAC:<MAC addr>\r\n
MX:3\r\n
\r\n
```

Note: local MAC address should be covered into message.

Bridge 200 OK message Example:

```
HTTP/1.1 200 OK\r\n\
ST: yeelink:yeebox\r\n\
USN: uuid:f5f52821-8290-294a-aed9-7dd1a5ca2165\r\n\
Ext:\r\n\
Cache-Control: max-age = 1800\r\n\
LOCATION: <IP addr>\r\n\
MAC: <MAC addr>\r\n\
BridgeVer: <version>\r\n\
Content-Length: 0\r\n\
\r\n
```

3.1.2 Control Mechanism

Once APP gets the IP address of Bridge by 200 OK response, it can create a socket to connect to Bridge via port 10003, then send the control message(use ASCII) which can be divided by “\n” or “\r\n”. Please refer to section 3.2.

3.2 Message Interface

3.2.1 Overview

APP transfers message to Bridge with ASCII which can be divided by “\n” or “\r\n”. The message covers two field: method and parameter. For example:

```
GLB 0001,1,1,86,255,255,255,100,0;143E,1,1,86,255,0,0,100;
```

“GLB” is the method and the rest are the parameters. When APP sends the method “GL” to get the current LED list, this message will be received. This message indicates that there are two LEDs whose ID is 0001 and 143E separated by “;”.

Method List:

Method	Discription
HB	Heartbeat
HACK	Heartbeat ACK
GL	Get list
GLB	Get list response
C	Control
CB	Control response
T	Time Delay
TB	Timer Delay ACK
RT	Retrive time delay setting
RTB	Retrive time delay setting response
S	Status Search
NEW	New LED joined

Please refer to the following section for specific parameter info.

3.2.2 HB/HACK

This mechanism is used for check the TCP/IP connection. APP sends HB for several seconds. If no HACK received, it suppose there is something wrong with the connection. Some action such as re-connecting should be taken at that time.

```
APP                Bridge
|----- HB ----->|
|<----- HACK -----|
```

There are no parameters for both messages.

3.2.3 GL/GLB

“GL” message is used for getting LED list from Bridge and “GLB” is for the response of “GL”.

```
APP                Bridge
|----- GL ----->|
|<----- GLB -----|
```

There is no parameter for GL.

GLB parameters:

GLB <ID>,<type>,<online>,<LQI>,<R>,<G>,,<L>,<effect>;...

,
ID: string(4 bytes), LED Zigbee ID
Type: number, 1 – RGB LED
Online: number, 0 – offline, 1 – online
LQI: number, 0~255, LED zigbee signal
R: number, 0~255, red color
G: number, 0~255, red color
B: number, 0~255, red color
L: number, 0~255, red color
Effect: reserved

3.2.4 C/CB

APP sends “C” to Bridge to control the color/on/off/brightness of LED. “CB” is the response of “C”.

```
APP                Bridge
|----- C ----->|
|<----- CB -----|
```

Parameter format:

C <ID>,<R>,<G>,,<L>,<effect>

The parameters are the same with “GLB”, please refer to section 3.2.3.

Turn off LED: C <ID>,,,,0,
Turn on LED: C <ID>,,,,100,
Change Color: C <ID>,<R>,<G>,,,
Change Brightness: C <ID>,,,,<L>,
Note: If ID is “G000”, it indicates for all LEDs.

3.2.5 T/TB

“T” is used for setting the delay on/off for LED. “TB” is the response of it.

```
APP                Bridge
|----- T ----->|
|<----- TB -----|
```

Parameter Format:

T <ID>,<T>,<S>
TB <ID>

ID: string(4 bytes), LED Zigbee ID
T: number, delay time(minutes); 0 is used for cancel;
S: number, 0 – delay off, 1 – delay on;

For example:

Turn off LED 0001 in 10 mins: T 0001,10,0
Cancel the time delay setting: T 0001,0,

3.2.6 RT/RTB

“RT” is used for checking the delay on/off setting for LED. “RTB” is the response of it.

```
APP                Bridge
|----- RT ----->|
|<----- RTB -----|
```

Parameter Format:

RT <ID>
RTB <ID>,<T>,<S>

ID: string(4 bytes), LED Zigbee ID
T: number, delay time(minutes); 0 is used for cancel;
S: number, 0 – delay off, 1 – delay on;

For example:

Search time delay setting: RT 0001
Response LED 0001 setting: RTB 0001,10,1

3.2.7 S

When the status of LED changes, Bridge will notify APP with the message “S”.

```
APP                Bridge
|----- S -----|
```

Parameter Format:

S <ID>,<online>,<LQI>,<R>,<G>,,<L>,<effect>

The parameters are the same with “GLB”, please refer to section 3.2.3

For example:

LED online: S <ID>,,1,,,,,
LED offline: S <ID>,,0,,,,,

3.2.8 NEW

When power on LED or a new LED joined to the network, bridge will send “NEW” message to APP.

```
APP                Bridge
|<----- NEW -----|
|<----- NEW -----|
|<----- NEW -----|
```

Parameter Format:

`NEW <ID>,<online>,<LQI>,<R>,<G>,,<L>,<effect>`

The parameters are the same with “S”, please refer to 3.2.7.