

Application note

Philips Hue lamps controlled from homeLYnk

How to configure homeLYnk for communication with Hue lamps.



Safety Information

Important Information



Read these instructions carefully before trying to install, configure, or operate this software. The following special messages may appear throughout this bulletin or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of either symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, can result in death or serious injury.

CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, can result in minor or moderate injury.

NOTICE


NOTICE is used to address practices not related to physical injury. The safety alert symbol shall not be used with this signal word.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction, installation, and operation of electrical equipment and has received safety training to recognize and avoid the hazards involved.

Safety Precautions

 WARNING
<p>HAZARD OF INCORRECT INFORMATION</p> <ul style="list-style-type: none">• Do not incorrectly configure the software, as this can lead to incorrect reports and/or data results.• Do not base your maintenance or service actions solely on messages and information displayed by the software.• Do not rely solely on software messages and reports to determine if the system is functioning correctly or meeting all applicable standards and requirements.• Consider the implications of unanticipated transmission delays or failures of communications links. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information that is contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2014 Schneider Electric. All rights reserved

Table of Contents

- 1 Introduction 6
- 2 Design 7
- 3 Configuration 8
 - 3.1 Hue Zigbee Bridge Connection Settings 8
 - 3.1.1 Communication between homeLYnk and Hue lamps 8
 - 3.1.2 Description of User library user.Hue_library 9
 - 3.1.3 Commands for Hue lamps control - description 10
 - 3.1.4 Examples of Hue lamps control command in Lua 13
- 4 Usage example 14
- 5 Conclusion 15
- 6 Appendix 15
 - 6.1 Glossary 15
 - 6.2 Reference 15

1 Introduction

This application note describes possibility to control Philips Hue lamps from KNX installation using homeLYnk. Philips Hue is RGB LED bulb, which is mainly intended to be controlled from smartphones and tablets.

Added value of this application note

Hue lamps are very state-of-art LED light bulbs, which are becoming very popular. They are mainly controlled by smartphones or special remote controllers. Using homeLYnk you can make your installation more complex and unified. You do not need any special application or remote controller, but you do all the setting of your lights directly from homeLYnk visualization or using KNX pushbuttons. Next and very important added value is, that you can use your KNX installation and homeLYnk logic to create advanced and modern set-up. You can control your Hue lamps to specific light level or you can control your light by movement sensor.

Competencies

This document is intended for readers who have been trained on homeLYnk, spaceLYnk products. The integration should not be attempted by someone who is new to the installation of either products. In addition we recommend that you be familiar with:

- The concepts of KNX
- Basic technical knowledge on software technologies such:
 - HTTP, JSON
 - LUA scripting

System prerequisites

Software	Version	Download
homeLYnk	1.2.1 and newer	http://www.schneider-electric.com

Table 1: software versions of used software

A glossary is available in the appendix chapter of this document. Please refer to it whenever necessary.

2 Design

Philips Hue lamps are controlled over ZigBee communication. There is Hue ZigBee bridge, which translates http requests from LAN network into ZigBee communication (proprietary protocol). HomeLYnk sends http request on events triggered by KNX devices or homeLYnk visualization.



Picture 1: homeLYnk - Philips Hue communication scheme

3 Configuration

3.1 Hue Zigbee Bridge Connection Settings

- Step 1** Connect Hue ZigBee bridge to LAN port of WiFi router.
- Step 2** Turn on DHCP in your router and choose one following options:
- a. **MAC->IP rule**
 - i. Set MAC-IP rule, which ensure, that your Hue bridge will get the same IP address after every request (MAC address of Hue bridge is written on the bottom of the device).
 - ii. Keep DHCP functions of your Wifi router turned on.
 - b. **Use mobile IP to set static IP address of Hue bridge**
 - i. Download official Philips Hue app for iOS or Android to your smartphone.
 - ii. Connect your smartphone into your home network.
 - iii. Search for Hue bridge using the iOS or Android app.
 - iv. Turn off DHCP setting of your Hue bridge (it will make its IP address static).
 - v. Now you can turn off DHCP functions of your WiFi router, if you do not need it for other purposes.

3.1.1 Communication between homeLYnk and Hue lamps

- Step 1** Open homeLYnk Configurator in your browser.
- Step 2** Restore user script library *user.Hue_library*.
- Step 3** Open script user.Hue_library in the script editor.
- Step 4** Set the IP of address of your bridge (parameter *ip_add*).
- Step 5** Click **Save and close**.
- Step 6** Append following code in to intial-script:

```
require('user.Hue_library')
if checkUser() == false then
    bridgeSetup()
    searchHueLights()
end
```

- Step 7** Sync project data by pressing button in right-bottom corner.
- Step 8** Press button on the top of Hue bridge.
- Step 9** Restart your homeLYnk (*Configurator >> Utilities >> System >> System >> Reboot*).
- Step 10** Check that connection has been established correctly (see message in log).

3.1.2 Description of User library `user.Hue_library`

User library `user.Hue_library` has been added in previous subchapter. It contains functions, which are used to control Hue lamps using http requests. All functions from this library are described below.

- **`sendToLight(Light_num,body_request)`**

Sends http request to specific Hue lamp and change its configuration. More details about http requests are provided in next subchapter.

- `Light_num` [integer value] – number of the Hue lamp, where the command will be sent
- `Body_request` [string] – http request, which specify the light settings.

- **`sendToGroup(Group_num,body_request)`**

Sends http request to specific group of Hue lamps and change its configuration. More details about http requests are provided in next subchapter.

- `Group_num` [integer value] – number of the Hue lamps group, where the command will be sent.
- `Body_request` [string] – http request, which specify the settings of light group.

- **`setRGB(Light_num,RGB_variable)`**

Change color of the lamp according to the RGB variable. It can be used together with `homeLYnk` colorpicker object.

- `Light_num` [integer value] – number of the Hue lamp, where the command will be sent.
- `RGB_variable` [3byte RGB color] – color of the lamp specified in RGB format.

- **`checkUser()`**

Check if the `homeLYnk` has been paired with Hue zigbee bridge. It returns true, if the pairing has been done already.

- **`bridgeSetup()`**

Perform a pairing procedure for `homeLYnk` and Hue zigbee bridge. Note that button on the Hue zigbee bridge has to be pressed before execution of this function.

- **`searchHueLights()`**

Execute auto discovery procedure, which will add new Hue lamps to the Hue zigbee bridge configuration.

3.1.3 Commands for Hue lamps control - description

There are various properties of Hue lamps configuration, which can be changed using http request.

The request can be send using user functions *sendToLight(Light_num,body_request)* and *sendToGroup(Group_num,body_request)*, which are described in subchapter above. Input variable *body_request* contains the string of http request. Examples of sending http requests using functions from user library are placed in the next subchapter. Parameters to be set in the http request are described in the table below.

Parameter	Type	Description	Possible values
"on"	bool	Turn the light on/off.	true, false
"bri"	uint8	Set brightness level of the lamps. Maximum brightness level at bri = 255. Note that bri=0 is minimal brightness, but the light is not turned off.	0..255
"xy"	[float,float]	X and Y coordinates from CIE color space (see picture below). Only colors inside green triangle can be chosen. Coordinates outside the color space will be approximate by closest color.	[0..1,0..1]
"hue"	uint16	Hue value set to light. The hue value is a wrapping value between 0 and 65535. Both 0 and 65535 are red, 25500 is green and 46920 is blue. e.g. "hue": 50000 will set the light to a specific hue. (see the picture below)	0..65535
"sat"	uint8	Saturation of the light. 255 is the most saturated (colored) and 0 is the least saturated (white).	0..255
"ct"	uint16	The Mired Color temperature of the light. 2012 connected lights are capable of 153 (6500K) to 500 (2000K).	153..500
"alert"	string	"none" – The light is not performing an alert effect. "select" – The light is performing one breathe cycle. "lselect" – The light is performing breathe cycles for 30 seconds or until an "alert": "none" command is received.	"none", "lselect", "select"
"effect"	string	"none" – The light is not performing any effect. "colorloop" – The light will cycle through all hues using the current brightness and saturation settings.	"none", "colorloop"

In order to better understand the setting of Hue lamps it is needed to explain the color space interpretation at first. The picture below shows the CIE color space and the green triangle outlines the colors, which can be produced by Hue lamps. You can see that some highly saturated colors and almost all green tones cannot be reached using Hue lamps.

There are 3 different ways how to determine the requested color of the light:

1. **XY coordinates in CIE color space.**

Select the color from CIE color space using XY coordinates. CIE color space is displayed in Picture 2.

If you select color outside the green triangle, closest color inside the triangle will be chosen.

Color is fully specified by two parameters: X and Y.

2. **Hue color + saturation**

The hue value is a wrapping value between 0 and 65535. Both 0 and 65535 are red, 25500 is green and 46920 is blue. See Picture 3. Saturation of the color can be set from 0..255. Full saturation basically goes around the green triangle in Picture 2. As the saturation value is decreasing, colors are going to be less intense and more white. In other words, decreasing the saturation value moves the color from the triangle edge towards the white point inside the triangle.

Color is fully specified by two parameters: Hue and saturation.

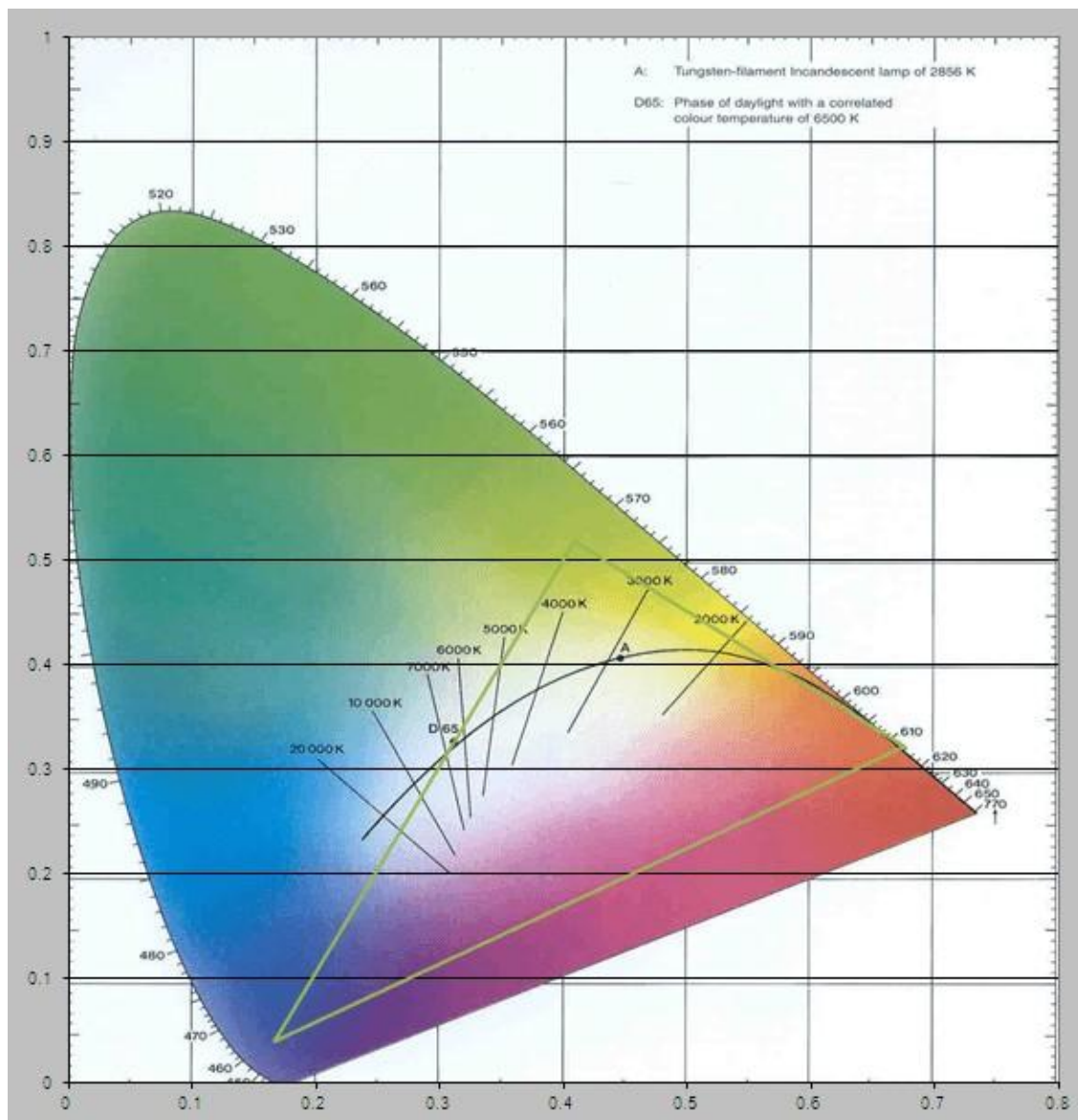
3. **White color temperature**

Black curve in the middle of Picture 2 follows white colors from the cold white to warm white. Hue lamps can produce white light between 2000K (warm) and 6500K (cold). The temperature of the white color can be specified by "ct" parameter (color temperature), which is scaled using scale named "reciprocal megakelvin" or "mirek". Using this scale, the warmest white 2000K corresponds to 500mirek ("ct":500) and the coldest white 6500K to 153mirek ("ct":153).

Color is fully specified by one parameter: color temperature.

RGB color setting

homeLYnk as KNX oriented devices supports KNX data type 232.600 RGB color. It can be useful and easier to specify color using RGB specification rather than some of three options listed above. In this case there has been created user function **setRGB(Light_num,RGB_variable)**, which is part of user library user.Hue_library. This function transform RGB value into XY coordinates and send it to a specified hue lamp.



Picture 2: CIE color space of Hue lamps

Hue	Final-x	Final-y
0	6.7500E-01	3.2200E-01
12750	5.4200E-01	4.2000E-01
25500	4.0900E-01	5.1800E-01
36210	2.8800E-01	2.7900E-01
46920	1.6700E-01	4.0000E-02
56100	4.2100E-01	1.8100E-01
65280	6.7500E-01	3.2200E-01

Picture 3: Common Hue values and corresponding XY values

3.1.4 Examples of Hue lamps control command in Lua

Example 1: Turn on Lamp 1 and set the brightness to 50 %.

```
sendToLight(1, '{ "on":true, "bri":127 }')
```

Example 2: Turn off Lamp 3.

```
sendToLight(3, '{ "on":false }')
```

Example 3: Set orange color of Lamp 1.

```
sendToLight(1, '{ "xy":[0.6,0.35] }')
```

Example 4: Turn on Lamp 2, set full brightness and blue color with full saturation.

```
sendToLight(2, '{ "on":true, "bri":255, "hue":46920, "sat":255 }')
```

Example 5: Change color of Lamp 2 from current state to warm white (2000K)

```
sendToLight(2, '{ "ct":500 }')
```

Example 6: Set color of Lamp3 to light blue using RGB object.

```
rgb=0x11bbff -Red=0x11, Green=0xbb, Blue=0xff  
setRGB(3, rgb)
```

Note: You can use group object of KNX data type 232.600 RGB color. Value of this object can be easily changed in homeLYnk visualization using color picker object.

Example 7: Make all lights blinking for 30s.

```
sendToGroup(0, '{ "alarm":"lselect" }')
```

Note: Group 0 is implicitly created in Hue bridge. It contains all available lamps. If you want to send command to all you lamps use command `sendToGroup(0, '{yourcommand}')`.

Example 8: Make all lights changing colors in the loop.

```
sendToGroup(0, '{ "effect":"colorloop" }')
```

4 Usage example

This application note contains ready to use example of Philips Hue integration with homeLYnk.

In order to use the example follow the steps below:

- Step 1** Navigate to *Configurator >> Utilities >> Restore*.
- Step 2** Chose file to restore: *AN020_homeLYnk_backup.tar.gz*
- Step 3** Find IP address of your Hue zigbee bridge (see Chapter 3.1).
- Step 4** Navigate to *Configurator >> Scripting >> User libraries*.
- Step 5** Open user.Hue_library in the script editor.
- Step 6** Set IP address of your Hue zigbee bridge.
- Step 7** Click **Save and close**.
- Step 8** Click **Sync project data** (right-bottom corner).
- Step 9** Pres button on the top of your Hue zigbee bridge.
- Step 10** Restart your homeLYnk.
- Step 11** Example can be used to control Hue lamps now.



Picture 4: Visualization of homeLYnk usage example

5 Conclusion

This application note provides step-by-step description how to pair your homeLYnk with Hue ZigBee bridge and your Hue lamps. Very important part is user library `user.Hue_library`, which helps to easy control Hue lamps from homeLYnk scripts. The main options of Hue lamps are described above and all of them are used in examples in Chapter 3.1.4. If you are looking for more detailed description of Hue API, please refer to the Hue API documentation website, which is listed below.

It is very suggested to use Usage example in the Chapter 4, which contains full homeLYnk backup and helps to better understand whole content of this application note.

6 Appendix

6.1 Glossary

The following table describes the acronyms and defines the specific terms used in this document.

Abbreviation	Description
DHCP	Dynamic Host Configuration Protocol
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
RGB	Red Green Blue (color representation)

Table 2: specific terms

6.2 Reference

Document title	Reference
homeLYnk User Guide	www.schneder-electric.com
Hue API documentation	www.developers.meethue.com/philips-hue-api

Table 3: reference

Schneider Electric Industries SAS

Head Office

35, rue Joseph Monier

92506 Rueil-Malmaison Cedex

FRANCE

www.schneider-electric.com