# ZIGBEE/BLUETOOTH LED CONTROLLER LD6

REFERENCE MANUAL

ubisys®

# Table of Contents

# 1. Overview

Thank you for purchasing the ubisys Zigbee and Bluetooth LED Controller LD6.

You have decided for a high-quality product with first-rate support!

This reference manual provides operating and maintenance instructions, interface specifications, command references and more. It is primarily intended for system integrators, not end-users.

An installation guide specific to your region is available as a separate document and is included in printed form in the product package.

If you have any questions or need additional support, please visit the support pages that best fit your background:

If you are a consumer (private household) or installer, please visit the Smart Home support pages at http://www.ubisys.de/en/smarthome/support.html for contact details.

As a commercial customer, please visit the Engineering support pages at https://www.ubisys.de/en/business-customers/ for contact details.

## 2. Features

- Zigbee 3.0 constant-voltage LED controller with six configurable outputs and three configurable inputs
- World's first device to feature Zigbee Direct, the unique combination of Bluetooth and Zigbee (since the firmware version 1.1.1 and later). The ubisys Zigbee Direct solution has been rewarded Golden Unit Designation by the Connectivity Standards Alliance
- Featuring ubisys Versalight™ universal light engine with unparalleled flexibility
  - Supports versatile configurations from one up to six primary LEDs per light endpoint
  - Supports one to six light endpoints in different setups: monochromatic/dimmable, CCT, RGB, RGBW, RGB+CCT, RGBA, RGBWA, etc.
  - Countless configuration options with the 23 most important profiles provided in the ubisys Smart Home App for iOS version 2.1.1 or higher at the touch of your fingertips.
    Notice: With the Zigbee Direct feature built into LD6 you can configure it with just the app and no gateway hardware and then add it to the hub of your choosing – or just use it without a hub for simple use-cases
  - Support for fine-tuning primary information if desired: Specify chromaticity and relative luminous flux for the LED strips or LED spots that are wired to the LD6 when this information is available from data sheets or chromaticity test reports
  - Advanced color mixing and improved color fading for visually pleasing transitions
  - Ability to apply output configuration changes on-the-fly (no reboot/power-cycle required any more). Endpoints and clusters are simply reconfigured dynamically as needed
- Suitable for decorative lighting (full color light strips) and general lighting (tunable white or monochromatic)
- Three configurable 3.3V inputs, one pre-configured for operating the first local output. Individually reconfigurable as on/off, level control, color temperature control, scene selector or automation switches. Profiles supporting momentary or stationary switches
- Zigbee Green Power Combined Device including Proxy and Sink functionality, supports on/off, level control and generic switches
- Local control even works when the device has not joined to a network (not commissioned yet)
- Supports groups, scenes, bindings, and reporting
- Reliable, unattended, maintenance-free operation; LD6 is designed for installation in false ceilings, behind mirrors, inside furniture, etc.
- Employs solid state switching for highest efficiency and durability
- Over-temperature, over-load, over-current, and short-circuit protected outputs
- Current limit: 6A per output, 12A in total (all outputs combined)
- 12/24V= supply voltage
- 3.3V regulated output provided to power sensors (touch, PIR, temperature, light-level etc.)
- Ready for I2C sensors (feature will be unlocked in a future firmware release)
- Made in Germany using high-quality, enduring parts for many years of life expectancy
- Leverages ubisys Zigbee stack for best-in-class reliability and performance

- Supports all channels in the 2.4 GHz band, i.e. channels 11-26 as per IEEE 802.15.4:
  - Primary = { 11, 15, 20, 25 };
  - Secondary = { 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 26 }
- Supports joining centralized and distributed security networks as router
- Supports forming simple centralized security networks as Coordinator and Trust Center
- Supports forming distributed security networks as router
- Three pre-configured Trust Center Link-Keys for joining:
  - Global Default Trust Center Link-Key (“ZigbeeAlliance09”)
  - Zigbee 3.0 Global Distributed Security Link-Key
  - Device-individual link-key derived from installation code - also printed as text and QR barcode
- Extended neighbor table with up to 26 entries for routers and end-devices
- Extended routing table with up to 96 entries for ad hoc and many-to-one routes - nearly ten times the capacity required by the standard (10)
- Extended buffering for sleepy end-devices with up to 24 buffers - 24 times the capacity required by the standard (1)
- Extended APS duplicate rejection table with up to 64 slots - 64 times the capacity required by the standard (1)
- Extensive transmit and receive queues for optimum through-put and minimum packet drop rate
- Reliable and scalable network-wide broadcasts featuring passive acknowledgments
- Reliable packet forwarding with automatic network-level retries
- Very sophisticated routing algorithm for reliable ad hoc routing - avoids routing loops even in case of concurrent route requests with overlapping source/destination; detects and breaks routing loops in environments with third-party stacks
- Firmware upgradable over-the-air during normal operation using Zigbee OTA Upgrade Cluster; with a suitable Zigbee Virtual Device that contains an OTA server, this is also possible via Bluetooth. The ubisys Smart Home App for iOS version 2.1 or later includes this feature, for example
- Man-Machine-Interface: A push-button and a LED for network steering, factory reset etc.
- Flame retardant housing (V-0); black, RAL 9005

## 3. Installation

The ubisys LD6 is designed for installation in ceilings, inside walls, behind mirrors, in furniture, caravans, and other places where a 12/24V DC supply is available. When the 24V supply is a safety low voltage (SELV), installation can be carried out without advanced professional skills according to the wiring diagrams provided in the leafleft of LD6.

## 4. Initial Device Start-Up

The device will search for an open Zigbee network to join when you first apply power to it.

Open the Zigbee network for joining and then power-up the device.

LD6 will blink quickly to indicate a search in progress. Once a search cycle is complete, it will either blink five times slowly to indicate it has joined a network, or keeps blinking quickly to indicate a joining failure - for example no network found, not permitted onto the network, etc. It will continue to search for a network in case of failure.

When the LD6 joins a new network, it blinks once, i.e. turns the output on to 100% for half a second,then off for half a second.

After joining a network, the device will prolong the joining window by three minutes via a fresh ZDO permit joining request.

Subsequently, when the device is power-cycled, it will blink five times slowly to indicate it is operating as a router on the network, ten times slowly to indicate that it is operating as a coordinator and trust center or blink quickly to indicate it is searching for a network to join. When commissioned, the router's LED will remain off after the five/ten blink cycles during normal operation. Afterwards, it will turn on only for as long as it is permitting other devices to join the network.

After reboot, if the device is properly commissioned, it resumes operation doing a "silent rejoin", i.e. it does not broadcast a device announcement in this case.

The device also starts advertising via Bluetooth such that Bluetooth central devices may discover it and connect via Bluetooth GATT. Refer to the Zigbee Direct specification for a detailed description of the commissioning and tunneling services provided by the LD6.

The device may turn on the load to a specific dimming level automatically after power is applied, based on the settings of the StartupOnOff and StartupLevel attributes. The default behavior is to return to the state prior to cutting power to the device.

# 5. Man-Machine interface (MMI)

LD6 offers a push-button behind a tiny hole in the front-face of the device, as well as a LED right next to it [1]. This man-machine interface provides access to a menu. In addition, it provides a handy shortcut to factory reset the device.

To factory reset the device, keep the button pressed for approximately 10 seconds until the LED starts to flash. This is equivalent to selecting menu item #5.
Notice that the device also supports a power-cycle sequencing factory reset as detailed in Section 5.1

To enter the menu, press and hold the push-button for more than a second until you see three short flashes followed by a sequence of one blink, pause, one blink, pause, etc. This indicates that you have successfully entered the menu. With each short button press (less than a second), you advance through the menu. For example, pressing the button once brings you to the second menu item, which LD6 indicates by two blinks, pause, two blinks pause, etc. Once you have reached the menu item that you want to run, press and hold the button for more than a second to execute the selected item. The exact meaning of each menu item depends on the particular device and may vary from firmware version to firmware version.

| Menu Item # | Operation |
|---|---|
| 1 | **Zigbee Commissioning: Network Steering**<br>While in this mode a single press on the button instigates Zigbee Network Steering (“EZ-mode”). If the device is on a network it will open the network for new devices for three minutes, otherwise it will attempt to join an open network. If the network is already open, the device will close the network instead such that repetitive presses toggle the network permit joining state. The LED is on, when the network is open for new devices, and off otherwise. |
| 2 | **Zigbee Commissioning: Finding & Binding**<br>Instigates the Zigbee Finding & Binding procedure (“EZ-mode”) on an initiator or target endpoint. Target endpoints will enter identify mode for three minutes. Initiator endpoints will query targets and create suitable bindings to these targets. After leaving the menu in this mode, the first button press starts the selection of the endpoint and each subsequent press (within half a second) increments the endpoint number. Once the desired endpoint is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled. |
| 3 | **Zigbee Commissioning: Clear Bindings**<br>Clears bindings on an initiator endpoint. Select this menu item and leave menu mode. Afterwards the device expects the selection of an initiator endpoint. The first button press starts the selection of the endpoint and each subsequent press (within half a second) increments the endpoint number. Once the desired endpoint is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled. This mode is active for one round only. Afterwards the device reverts to mode #1. |
| 4 | **Zigbee Commissioning: Set Device Role and Factory Reset**<br>Selects the Zigbee device role for this device on the network, resets the remaining settings to factory defaults and restarts the device. Select this menu item and leave menu mode. Afterwards the device expects the selection of an option. The first button press starts the selection of the option and each subsequent press (within half a second) increments the option number. Once the desired option is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled. The following options are available:<br>Option #1: Join an existing Zigbee network as router<br>Option #2: Form a new distributed security network as the first router<br>Option #3: Form a new centralized security network as the coordinator and trust center |
| 5 | **Factory Reset**<br>Put the device into the same state as it was when it left the factory, then reboot. The only exception is the outgoing network security frame counter, which is preserved across factory resets. The device will broadcast a network leave indication . |

| Menu Item # | Operation |
|---|---|
| 6 | **Zigbee Commissioning: Advanced Commands**<br>Provides a number of advanced Zigbee commissioning command options. Select this menu item and leave menu mode. Afterwards the device expects the selection of an option. The first button press starts the selection of the option and each subsequent press (within half a second) increments the option number.<br>Once the desired option is reached, wait for the LED to blink the selected number of times. Then, press a single time to accept the selection or wait for three seconds to cancel the command. The LED will blink one time to confirm an affirmative choice, or two times to indicate the request has been cancelled. The following options are available:<br>Option #1: Perform a simple reset (reboot), then continue operating seamlessly (silent re-join)<br>Option #2: Perform a simple reset (reboot), then re-join the network<br>Option #3: Perform a full factory reset, including security frame counters and preserved settings |
| 7 | Reserved for internal use. Do not leave the menu in this mode. |

## 5.1. Power-Cycle Sequencing Factory Reset

It is possible to instigate a factory reset using a special power-cycle sequence. This is equivalent to selecting menu item #5, with the advantage that you need no access to the device itself (only to its power supply).

1. Power the device for at least four seconds.
2. Interrupt the power supply for at least a second.
3. Reapply power for less than two seconds but more than half a second.
   Notice that at the end of this cycle, the device is off and should remain off for at least a second.
4. Repeat the previous step two more times, for a total of three short power cycles.
5. Apply power to the device and leave it powered on.

The device will now factory reset and reboot.

After above sequence, the dimmer will flash the connected light three times to indicate the factory reset sequence is in progress. This is the default reset sequence, equivalent to selecting menu item #5 as described above, or keeping the menu button pressed for more than 10 seconds, or removing the device from the network using a leave request.

[1] The high-voltage inputs of LD6 do not count as an MMI feature.

# 6. Zigbee Interface

Please refer to the following IEEE and Zigbee Alliance documents, which apply to this product:

[R1] IEEE Standard 802 - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)
[R2] ZigBee Specification, Revision 23, Document No. 05-3474-23
[R3] Zigbee 2015 Layer PICS and Stack Profiles, Revision 6, Document No. 08-0006-06
[R4] ZigBee Cluster Library Specification, Revision 8, Document No. 07-5123-08
[R5] Zigbee Base Device Behavior Specification, Revision 13, Document No. 13-0402-13
[R6] Zigbee PRO Green Power Feature Specification, Revision 26, Document No. 09-5499-26
[R7] Zigbee Home Automation Public Application Profile 1.2, Revision 29, Document No. 05-3520-29
[R8] Zigbee Smart Energy Standard 1.1b, Revision 18, Document No. 07-5356-18
[R9] Zigbee Device Library Specification, Revision 6, Document No. 19-02016-006
[R10] Zigbee WWAH ZCL Cluster Definition, Version 1.0, Document No. 17-01067-024
[R11] Zigbee Direct Specification, Version 1.0, Document No. 20-27688-037
[R12] Zigbee QR Code Requirements, Revision 9, Connectivity Standards Alliance Document 18-01000-009

## 6.1. Device Anatomy

The LD6 provides 6 controllable output PWM channels for a rated voltage of 12V or 24V and a current of up 12A combined or 6A per channel. It combines state-of-the-art solid-state technology with advanced firmware for a unique feature set. In addition, three inputs are available for local control or remote control of other on/off, level control or color control target devices or groups. Being a constantly powered device, LD6 also acts as a Zigbee router improving network connectivity and meshing capabilities as well as serving as a connection point (“parent”) for sleepy end-devices, Green Power Devices, and Zigbee Virtual Devices (Bluetooth).

LD6 provides up to twelve Zigbee application endpoints:

| Endpoint # | Profile | Application |
|---|---|---|
| 0 (0x00) | 0x0000:<br>Zigbee Device Profile | Zigbee Device Object (ZDO) - standard management features |
| 1 (0x01) | 0x0104:<br>Common Profile (HA) | This is the first out of up to six lighting endpoints that provide control of a light connected to one or more output channels via on/off, level control or color control clusters. It supports groups and scenes, as well as reporting for on/off state, level and color feedback. The identify cluster allows for push-button commissioning as a target (finding & binding). In addition, the ballast configuration cluster provides additional control over the dimming behavior.<br><br>Depending on the specific output configuration settings currently in use, the LD6 chooses an appropriate device type for this endpoint: Dimmable Light (0x0101), Color Temperature Light (0x010C), or Extended Color Light (0x010D). Consequently, the application endpoint has the ability to enable a suitable set of inbound and outbound clusters. Notice that the endpoint is dynamic, i.e. if no output channel is assigned to this endpoint, it will not be present. Please check Section 6.13.3.3 for details. |
| 2 (0x02) | 0x0104:<br>Common Profile (HA) | Dimmer Switch (0x0104)<br>Based on the input configurations set via the manufacturer-specific Device Setup Cluster (0xFC00) on endpoint #232, this application endpoint has the capability to transmit commands such as on/off, level control, color control, or manufacturer-specific automation switch/button commands. These commands are activated in response to local high-voltage inputs. |

| Endpoint # | Profile | Application |
|---|---|---|
| 3 (0x03) | 0x0104:<br>Common Profile (HA) | Dimmer Switch (0x0104)<br>Based on the input configurations set via the manufacturer-specific Device Setup Cluster (0xFC00) on endpoint #232, this application endpoint has the capability to transmit commands such as on/off, level control, color control, or manufacturer-specific automation switch/button commands. These commands are activated in response to local high-voltage inputs. |
| 4 (0x04) | 0x0104:<br>Common Profile (HA) | Dimmer Switch (0x0104)<br>Based on the input configurations set via the manufacturer-specific Device Setup Cluster (0xFC00) on endpoint #232, this application endpoint has the capability to transmit commands such as on/off, level control, color control, or manufacturer-specific automation switch/button commands. These commands are activated in response to local high-voltage inputs. |
| 5 (0x05) | 0x0104:<br>Common Profile (HA) | This endpoint provides the same functionality as endpoint #1, and allows independent control of a second light. This is a dynamic endpoint which may or may not be present, depending on the output configuration. Please refer to Section 6.13.3.3 for details. |
| 6 (0x06) | 0x0104:<br>Common Profile (HA) | This endpoint provides the same functionality as endpoint #1, and allows independent control of a third light. This is a dynamic endpoint which may or may not be present, depending on the output configuration. Please refer to Section 6.13.3.3 for details. |
| 7 (0x07) | 0x0104:<br>Common Profile (HA) | This endpoint provides the same functionality as endpoint #1, and allows independent control of a fourth light. This is a dynamic endpoint which may or may not be present, depending on the output configuration. Please refer to Section 6.13.3.3 for details. |
| 8 (0x08) | 0x0104:<br>Common Profile (HA) | This endpoint provides the same functionality as endpoint #1, and allows independent control of a fifth light. This is a dynamic endpoint which may or may not be present, depending on the output configuration. Please refer to Section 6.13.3.3 for details. |
| 9 (0x09) | 0x0104:<br>Common Profile (HA) | This endpoint provides the same functionality as endpoint #1, and allows independent control of a sixth light. This is a dynamic endpoint which may or may not be present, depending on the output configuration. Please refer to Section 6.13.3.3 for details. |
| 232 (0xE8) | 0x0104:<br>Common Profile (HA) | Device Management<br>Applied Device Type is Physical Devices (0x0507).<br>This endpoint provides a series of device management settings, including input configurations and output configurations.<br>This endpoint supports Over-The-Air firmware upgrade. |
| 242 (0xF2) | 0xA1E0:<br>Green Power Profile | GP Combo Basic (0x0066)<br>This endpoint supports Zigbee Green Power Combined Proxy and Sink. |

The ubisys Zigbee manufacturer code is 0x10F2. This manufacturer code can be used to identify OTA upgrade images, for accessing manufacturer-specific ZCL attributes and commands etc.

Depending on the currently applied specific output configuration settings, the endpoints #1, #5, #6, #7, #8, and #9 may be either present (visible) or absent. Please check Section 6.13.3.3 for details.

## 6.2. Installation Code

This device has a pre-configured link key, which is derived from the installation code printed on the back of the device's housing in text format and as a two-dimensional barcode (QR code). The format specified in [R5], section 5.4.8.1.1 is used with a full 128-bit installation code + 16-bit CRC. The QR code uses the recommended encoding as per [R12], as illustrated in the following example:

```
Z:001FEE0000009508$I:A160CF46F73495D921740180A9A72F8CF6C6%G$M:LD6%M:10F2
```

The content string initiates with an EUI-64 address (prefixed by "Z:"). It is followed by a complete 128-bit installation code along with its 16-bit CRC checksum (prefixed by "I:"). Following that are the model identifier string "LD6" (prefixed by "M:") and the manufacturer ID 0x10f2 (prefixed by "M:").

## 6.3. Application Endpoint #0 - Zigbee Device Object

Please refer to the Zigbee Specification [R2] for details on the Zigbee Device Object (ZDO) and the protocol used for over-the-air communication, called the Zigbee Device Profile (ZDP). Notice that the ZDP is fundamentally different from typical application endpoints, which build on the Zigbee foundation framework and the Zigbee Cluster Library (ZCL).

The ubisys Zigbee LD6 supports the following ZDO services:

| Primitive | Description |
|---|---|
| nwk_addr_req/<br>nwk_addr_rsp | **Network address request/response**<br>Translates a 64-bit IEEE address into a 16-bit network short address. Use only when really required, because this message employs a network-wide broadcast (flooding) |
| ieee_addr_req/<br>ieee_addr_rsp | **IEEE address request/response**<br>Translates a 16-bit network short address into a 64-bit IEEE address. |
| node_desc_req/<br>node_desc_rsp | **Node descriptor request/response**<br>Returns information such as the manufacturer ID, power supply, etc. |
| power_desc_req/<br>power_desc_rsp [2] | **Power descriptor request/response**<br>Returns information such as the power source and mode. |
| active_ep_req/<br>active_ep_rsp | **Active endpoints request/response**<br>Returns a set of available application endpoints on the device. |
| simple_desc_req/<br>simple_desc_rsp | **Simple descriptor request/response**<br>Returns a descriptor for a certain application endpoint with a list of available services (clusters). |
| match_desc_req/<br>match_desc_rsp | **Match descriptor request/response**<br>Searches for a certain cluster or set of clusters and returns the matching endpoints, if any. |
| device_annce | **Device announcement**<br>Advertises the presence of a new device in the network. |
| parent_annce/<br>parent_annce_rsp [3] | **Parent announcement/response**<br>This is part of the Zigbee 2015 end-device child management feature. |
| bind_req/<br>bind_rsp | **Bind request/response**<br>Creates an application binding |
| unbind_req/<br>unbind_rsp | **Unbind request/response**<br>Removes an application binding |
| mgmt_lqi_req/<br>mgmt_lqi_rsp | **Management: Neighbor table request/response**<br>Returns information about neighboring devices, including the link quality, device type etc. |
| mgmt_bind_req/<br>mgmt_bind_rsp | **Management: Binding table request/response**<br>Returns information about application bindings on the device. |
| mgmt_leave_req/<br>mgmt_leave_rsp | **Management: Leave request/response**<br>Makes the device leave the network or removes one of its end-device children. |
| mgmt_permit_joining_req/<br>mgmt_permit_joining_rsp | **Management: Permit joining request/response**<br>Opens the network for new devices to join. |

## 6.4. Application Endpoint #1

This is a Finding & Binding target endpoint.

### 6.4.1. Supported Device Types

The LD6 supports the following pre-defined Device types on endpoint #1:

| Device Type | Description |
|---|---|
| 0x0101 | **Dimmable Light**<br>Supports on/off control<br>Supports light level control |

| Device Type | Description |
|---|---|
| 0x010C | **Color Temperature Light**<br>Supports on/off control<br>Supports light level control<br>Supports color temperature control (tunable white) |
| 0x010D | **Extended Color Light**<br>Supports on/off control<br>Supports light level control<br>Supports full color (CIE 1931 xy or hue/saturation) and color temperature control. |

The LD6 will adapt the device type, the clusters, and capabilities (where applicable) to reflect the output configuration as specified by the user. Please refer to Section 6.13.3.3 for details.

#### 6.4.1.1. Dimmable Light Device Type

When the endpoint represents a dimmable light, it exposes the following seven inbound clusters:

| Cluster | Direction | Description |
|---|---|---|
| 0x0000 | Inbound (Server) | **Basic**<br>Provides basic information about the device, such as the manufacturer, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network. |
| 0x0003 | Inbound (Server) | **Identify**<br>Allows to put the endpoint into identify mode. Useful for identifying/locating devices and required for Finding & Binding. |
| 0x0004 | Inbound (Server) | **Groups**<br>Allows adding this endpoint to one or more groups. Afterwards the endpoint can be addressed using the group address. This is also a prerequisite for scenes. You may also query group membership and delete group associations. |
| 0x0005 | Inbound (Server) | **Scenes**<br>Allows storing one or more scenes per group, where each scene consists of a pre-set on/off state value. You may either store the current values as a scene, or provide scene settings when adding a scene, or delete scenes. |
| 0x0006 | Inbound (Server) | **On/off**<br>Provides on/off control of the attached light source. |
| 0x0008 | Inbound (Server) | **Level Control**<br>Provides level control of the attached light source. |
| 0x0301 | Inbound (Server) | **Ballast Configuration**<br>Provides a means to configure the minimum and maximum levels for the attached light source. |

#### 6.4.1.2. Color Temperature Light Device Type

When the endpoint represents a color temperature (tunable white) light, it exposes the following eight inbound clusters:

| Cluster | Direction | Description |
|---|---|---|
| 0x0000 | Inbound (Server) | **Basic**<br>Provides basic information about the device, such as the manufacturer, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network. |
| 0x0003 | Inbound (Server) | **Identify**<br>Allows to put the endpoint into identify mode. Useful for identifying/locating devices and required for Finding & Binding. |
| 0x0004 | Inbound (Server) | **Groups**<br>Allows adding this endpoint to one or more groups. Afterwards the endpoint can be addressed using the group address. This is also a prerequisite for scenes. You may also query group membership and delete group associations. |

| Cluster | Direction | Description |
|---|---|---|
| 0x0005 | Inbound (Server) | **Scenes** Allows storing one or more scenes per group, where each scene consists of a pre-set on/off state value. You may either store the current values as a scene, or provide scene settings when adding a scene, or delete scenes. |
| 0x0006 | Inbound (Server) | **On/off** Provides on/off control of the attached light source. |
| 0x0008 | Inbound (Server) | **Level Control** Provides level control of the attached light source. |
| 0x0301 | Inbound (Server) | **Color Control** Provides color temperature control of the attached light source. |
| 0x0301 | Inbound (Server) | **Ballast Configuration** Provides a means to configure the minimum and maximum levels for the attached light source. |

### 6.4.1.3. Extended Color Light Device Type

For application of extended color light device type, the application endpoint #1 exposes the following eight inbound clusters:

| Cluster | Direction | Description |
|---|---|---|
| 0x0000 | Inbound (Server) | **Basic** Provides basic information about the device, such as the manufacturer, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network. |
| 0x0003 | Inbound (Server) | **Identify** Allows to put the endpoint into identify mode. Useful for identifying/locating devices and required for Finding & Binding. |
| 0x0004 | Inbound (Server) | **Groups** Allows adding this endpoint to one or more groups. Afterwards the endpoint can be addressed using the group address. This is also a prerequisite for scenes. You may also query group membership and delete group associations. |
| 0x0005 | Inbound (Server) | **Scenes** Allows storing one or more scenes per group, where each scene consists of a pre-set on/off state value. You may either store the current values as a scene, or provide scene settings when adding a scene, or delete scenes. |
| 0x0006 | Inbound (Server) | **On/off** Provides on/off control of the attached light source. |
| 0x0008 | Inbound (Server) | **Level Control** Provides level control of the attached light source. |
| 0x0301 | Inbound (Server) | **Color Control** Provides full color and color temperature control of the attached light source. |
| 0x0301 | Inbound (Server) | **Ballast Configuration** Provides a means to configure the minimum and maximum levels for the attached light source. |

## 6.4.2. Basic Cluster (Server)

The basic cluster might be accessible via more than one endpoint. Most of its attributes are singleton attributes, i.e. all instances of the cluster share the same attribute value storage.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned8, read-only | **ZCLVersion** The version of the cluster library |
| 0x0001 | unsigned8, read-only | **ApplicationVersion** The application version |
| 0x0002 | unsigned8, read-only | **StackVersion** The stack version |

| Attribute | Type | Description |
|---|---|---|
| 0x0003 | unsigned8, read-only | **HWVersion**<br>Specifies the hardware revision number |
| 0x0004 | string, read-only | **ManufacturerName**<br>“ubisys” |
| 0x0005 | string, read-only | **ModelIdentifier**<br>“LD6” |
| 0x0006 | string, read-only | **DateCode**<br>“YYYYMMDD-XX-FBV”, where YYYY = year, MM = month, DD = day of production (hardware), XX = production facility, V = factory block version. For example, “20220412-DE-FB1” refers to a device built April 12, 2022 in Germany and programmed with factory block format version 1. |
| 0x0007 | enum8, read-only | **PowerSource**<br>Always set to mains-powered, single phase (regardless of the connected supply voltage) |
| 0x0008 | enum8, read-only | **GenericDeviceClass**<br>Specifies the field of application of the GenericDeviceType attribute.<br>Mostly for compatibility with Zigbee Light Link.<br>Always set to “0” indicating Lighting class |
| 0x0009 | enum8, read-only | **GenericDeviceType**<br>“Unspecified” by default. Mostly for compatibility with Zigbee Light Link. |
| 0x000A | raw binary, read-only | **ProductCode**<br>Specifies the GTIN (EAN) code for the product. |
| 0x000B | string, read-only | **ProductURL**<br>Specifies a link to a web page containing specific product information.<br>“www.ubisys.de/products/ld6” |
| 0x0010 | string, persistent | **LocationDescription**<br>Empty string by default. Might be set by commissioning software during installation |
| 0x0011 | unsigned8, persistent | **PhysicalEnvironment**<br>“Unspecified” by default. Might be set by commissioning software during installation |
| 0x4000 | string, read-only | **SWBuildID**<br>Firmware Build ID. Human readable string that represents the application’s firmware image version. This is the preferred version identifier to be used when identifying a version, e.g. when contacting customer support |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Cluster commands supported:

None

### 6.4.3. Identify Cluster (Server)

In identify mode, the output will be toggled, dimmed, undergo a color change or other visual effect.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned16 | **IdentifyTime**<br>The remaining time, in seconds, the device will be identifying itself.<br>Notice that LD6 will toggle the connected light in identify mode fully on and fully off once per second. |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Identify**<br>Puts the device into identify mode, or terminates identify mode. |

| Command | Description |
|---|---|
| 0x01 | **Query Identify**<br>Determines whether the device is currently identifying, and returns the remaining time if so. |
| 0x40 | **Trigger Effect**<br>Instigates a visual effect. |

### 6.4.4. Groups Cluster (Server)

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | bitmap8,<br>read-only | **NameSupport**<br>Always set to 0: This implementation does not support storing names for groups. |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Add Group**<br>Adds the endpoint to a group. |
| 0x01 | **View Group**<br>Determines whether the device belongs to a group and returns the group name, if supported. |
| 0x02 | **Get Group Membership**<br>Returns the set of groups this endpoint belongs to. |
| 0x03 | **Remove Group**<br>Removes this endpoint from the specified group. Also removes all scenes that refer to this group. |
| 0x04 | **Remove All Groups**<br>Removes this endpoint from all groups. Also removes all scenes that refer to any of the existing groups. |
| 0x05 | **Add Group if Identifying**<br>Adds this endpoint to the group, if the endpoint is identifying. |

### 6.4.5. Scenes Cluster (Server)

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned8,<br>read-only | **SceneCount**<br>Holds the total number of scenes (across all groups) currently stored on the device. |
| 0x0001 | unsigned8,<br>read-only | **CurrentScene**<br>If the SceneValid attribute is true, this attribute, together with the CurrentGroup attribute, indicates the currently active scene. |
| 0x0002 | unsigned8,<br>read-only | **CurrentGroup**<br>If the SceneValid attribute is true, this attribute, together with the CurrentScene attribute, indicates the currently active scene. |
| 0x0003 | bool,<br>read-only | **SceneValid**<br>If true, the scene identified by CurrentGroup and CurrentScene is currently active, i.e. all device attribute values match the values in the scene field set. |
| 0x0004 | bitmap8,<br>read-only | **NameSupport**<br>Always set to 1: This implementation DOES support storing names for scenes. |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Add Scene**<br>Adds a scene with or without a scene field set |
| 0x01 | **View Scene**<br>Returns the scene field set, name and transition times for a scene. |
| 0x02 | **Remove Scene**<br>Removes a scene from the scene table. |
| 0x03 | **Remove All Scenes**<br>Removes all scenes that belong to a particular group. |
| 0x04 | **Store Scene**<br>Stores the device's current state as a scene or updates a previously stored scene accordingly. |
| 0x05 | **Recall Scene**<br>Reverts the device's current state using the values from the previously stored field set. |
| 0x06 | **Get Scene Membership**<br>Returns the set of scenes (within the scope of the specified group) currently stored on the device. |
| 0x40 | **Enhanced Add Scene**<br>Similar to Add Scene with higher transition time resolution (10ths of a second instead of once second) |
| 0x41 | **Enhanced View Scene**<br>Similar to View Scene with higher transition time resolution (10ths of a second instead of once second) |
| 0x42 | **Copy Scene**<br>Copies a scene in a single operation as an alternative to (Enhanced) View Scene, (Enhanced) Add Scene |

### 6.4.6. On/off Cluster (Server)

This cluster uses the binding table for managing reporting targets.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | bool,<br>read-only,<br>reportable | **OnOff**<br>Indicates the current state of the output relay, either on = "true" or off = "false". |
| 0x4000 | bool,<br>read-only | **GlobalSceneControl**<br>Determines whether the next "Off with Effect" command will store the global scene. |
| 0x4001 | unsigned16,<br>read-only | **OnTime**<br>Time, in tenths of a second, the device remains on, before it automatically turns off. This value is set by the "On with timed off" command. This is a "live" down counter. |
| 0x4002 | unsigned16,<br>read-only | **OffWaitTime**<br>Time, in tenths of a second, the device ignores "On with timed off" commands. This is a "live" down counter. |
| 0x4003 | enum8,<br>persistent | **StartupOnOff**<br>Determines the start-up behavior of the device after reboot. Either always turn off (0x00) or turn on (0x01) the device; or restore the previous setting (0xFF) or invert the previous setting (0x02) |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Turn off**<br>Turns the output off. The attached load will be disconnected from the mains. |
| 0x01 | **Turn on**<br>Turns the output on. The attached load will be connected to the mains. |
| 0x02 | **Toggle**<br>Turns the output off, it was turned on or turns the output on, if it was turned off. |

| Command | Description |
|---|---|
| 0x40 | **Off with effect**<br>Turns the output off after having stored the current setting as the global scene and rejects further global scene storage requests by the “On with recall global scene” command, until the device is turned on again or the global scene has been restored. |
| 0x41 | **On with recall global scene**<br>Restores the global scene. Despite the name of this command, it does not always turn the device “on”; rather, it just recalls the previously stored setting. |
| 0x42 | **On with timed off**<br>Turns the output on and then automatically turns it off after the specified time has elapsed. |

### 6.4.7. Level Control Cluster (Server)

This cluster uses the binding table for managing reporting targets.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned8, read-only, reportable | **CurrentLevel**<br>Indicates the current level of the dimmable light output, between 1 and 254. |
| 0x0001 | unsigned16, read-only | **RemainingTime**<br>Specifies the amount of time, in units of 0.1 seconds, required to reach the specified target level |
| 0x000F | bitmap8, persistent | **Options**<br>Specifies the default command options for commands that support options; these are applied either when the option value and mask are not included in the command payload, or the mask indicates that certain default option bits apply.<br>Factory default setting for<br>bit0: "1" = Enable execute if off and<br>bit1: "0" = Disable LevelControlForLighting |
| 0x0010 | unsigned16, persistent | **OnOffTransitionTime**<br>Specifies the amount of time, in units of 0.1 seconds, which will be used during a transition to either the on or off state, when an on/off/toggle command of the on/off cluster is used to turn the light on or off |
| 0x0011 | unsigned8, persistent | **OnLevel**<br>Specifies the level that shall be applied, when an on/toggle command causes the light to turn on. By default this attribute is set to the invalid value (0xFF), the light will return to the previously active level (before it was turned off), when it is turned on again. |
| 0x4000 | unsigned8, persistent | **StartupLevel**<br>Specifies the initial level to be applied after reboot.<br>By default this attribute is set to the invalid value (0xFF), the light will return to the previously active level (before power was cut), when it is turned on again. Otherwise the current level will be set to the value specified here subject to range limitations imposed by the ballast configuration cluster on this endpoint. |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Manufacturer-specific attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned8, | **MinimumOnLevel** [4]<br>Specifies the minimum level that shall be applied, when an on/toggle command causes the light to turn on.<br>When the attribute is configured with the invalid value (0xFF), the corresponding feature becomes inactive, and the standard rule takes effect: the light will revert to its previous active level (prior to being turned off) if the "OnLevel" attribute is set to the invalid value (0xFF). Alternatively, it will adopt the specified value from the "OnLevel" attribute which falls within the range of 0x00 to 0xFE. If the "MinimumOnLevel" attribute value is within the range of 0x00 to 0xFE, the light will be set to the higher value between its previous active level and the "MinimumOnLevel" attribute value.<br>For example, if the previous level was 0x30 and the MinimumOnLevel was 0x40, then the light should turn on and move to the level 0x40. Conversely, if the previous level was 0x50, and the MinimumOnLevel was 0x40, then the light should turn on and move to to level 0x50.<br>By default the attribute is set to 0x1a = 10%. |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Move To Level**<br>Moves the current level to a certain target level within a specified transition time.<br>Supports command options. |
| 0x01 | **Move**<br>Moves the level either up or down at a specified rate.<br>Supports command options. |
| 0x02 | **Step**<br>Increments or decrements the level by a certain amount within a specified transition time.<br>Supports command options. |
| 0x03 | **Stop**<br>Stops any level change in progress due to a move, move to level, step or recall scene command. |
| 0x04 | **Move To Level with on/off**<br>Moves the current level to a certain target level within a specified transition time. Maintains the companion on/off cluster's OnOff attribute. |
| 0x05 | **Move with on/off**<br>Moves the level either up or down at a specified rate. Maintains the companion on/off cluster's OnOff attribute. |
| 0x06 | **Step with on/off**<br>Increments or decrements the level by a certain amount within a specified transition time. Maintains the companion on/off cluster's OnOff attribute. |
| 0x07 | **Stop**<br>Stops any level change in progress due to a move, move to level, step or recall scene command. |

### 6.4.8. Color Control Cluster (Server)

This cluster uses the binding table for managing reporting targets.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned8,<br>read-only,<br>reportable | **CurrentHue**<br>Contains the current hue value of the light. It is updated as fast as practical during commands that change the hue. |
| 0x0001 | unsigned8,<br>read-only,<br>reportable | **CurrentSaturation**<br>Contains the current saturation value of the light. It is updated as fast as practical during commands that change the saturation. |
| 0x0002 | unsigned16<br>read-only | **RemainingTime**<br>Contains the time remaining, in 1/10ths of a second, until the currently active command will be complete. |
| 0x0003 | unsigned16,<br>read-only,<br>reportable | **CurrentX**<br>Contains the current value of the normalized chromaticity value x, as defined in the CIE xyY Color Space. It is updated as fast as practical during commands that change the color. |

| Attribute | Type | Description |
|---|---|---|
| 0x0004 | unsigned16<br>read-only,<br>reportable | **CurrentY**<br>Contains the current value of the normalized chromaticity value y, as defined in the CIE xyY Color Space. It is updated as fast as practical during commands that change the color. |
| 0x0007 | unsigned16,<br>read-only,<br>reportable | **ColorTemperatureMireds**<br>Contains a scaled inverse of the current value of the color temperature in Kelvins. |
| 0x0008 | Enum8,<br>read-only | **ColorMode**<br>Indicates which attributes are currently determining the color of the device. The value of the ColorMode attribute cannot be written directly - it is set upon reception of any supported command listed in Table 2 to the appropriate mode for that command. |
| 0x000f | Bitmap8,<br>persistent | **Options**<br>A bitmap that determines the default behavior of some cluster commands. It specifies the default command options for commands that support options; these are applied either when the option value and mask are not included in the command payload, or the mask indicates that certain default option bits apply. Factory default setting for "execute if off" = "1". |
| 0x4000 | unsigned16,<br>read-only,<br>reportable | **EnhancedCurrentHue**<br>Represents non-equidistant steps along the CIE 1931 color triangle, and provides 16-bits precision.<br>The CurrentHue attribute contains a hue value in the range 0 to 254, calculated from the EnhancedCurrentHue attribute. |
| 0x4001 | Enum8,<br>read-only | **EnhancedColorMode**<br>Specifies which attributes are currently determining the color of the device.<br>If the ColorMode attribute is changed, e.g., due to one of commands listed in Table 2, its new value is copied to the EnhancedColorMode attribute. |
| 0x4002 | unsigned8,<br>read-only | **ColorLoopActive**<br>Specifies the current active status of the color loop. |
| 0x4003 | unsigned8,<br>read-only | **ColorLoopDirection**<br>Specifies the current direction of the color loop. |
| 0x4004 | unsigned16,<br>read-only | **ColorLoopTime**<br>Specifies the number of seconds it takes to perform a full color loop, i.e., to cycle all values of the EnhancedCurrentHue attribute. |
| 0x4005 | unsigned16,<br>read-only | **ColorLoopStartEnhancedHue**<br>Specifies the value of the EnhancedCurrentHue attribute from which the color loop is started. |
| 0x4006 | unsigned16,<br>read-only | **ColorLoopStoredEnhancedHue**<br>Specifies the value of the EnhancedCurrentHue attribute before the color loop was started. Once the color loop is complete, the EnhancedCurrentHue attribute is restored to this value. |
| 0x400A | Bitmap16,<br>read-only | **ColorCapabilities**<br>Specifies the color capabilities of LD6 supporting the color control cluster.<br>“0x001F” |
| 0x400B | unsigned16,<br>read-only | **ColorTempPhysicalMinMireds**<br>Indicates the minimum mired value supported by the device. |
| 0x400C | unsigned16,<br>read-only | **ColorTempPhysicalMaxMireds**<br>Indicates the maximum mired value supported by the device. |
| 0x400D | unsigned16,<br>read-only | **CoupleColorTempToLevelMinMireds**<br>Specifies a lower bound on the value of the ColorTemperatureMireds attribute for the purposes of coupling the ColorTemperatureMireds attribute to the CurrentLevel attribute when the CoupleColorTempToLevel bit of the Options attribute of the Level Control cluster is equal to 1.<br>By default the CoupleColorTempToLevel bit of the Options attribute of the Level Control cluster is set to 0 on LD6, see Section 6.4.7. |
| 0x4010 | unsigned16,<br>persistent | **StartUpColorTemperatureMireds**<br>Specifies the desired startup color temperature value LD6 uses when it is supplied with power and this value is reflected in the ColorTemperatureMireds attribute. |
| 0x0010 | unsigned8,<br>read-only | **NumberOfPrimaries**<br>Holds the number of color primaries implemented on this device.<br>“5” |
| 0x0011 | unsigned16,<br>read-only | **Primary1X**<br>Contains the normalized chromaticity value x for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary1X attribute by the relationship. |
| 0x0012 | unsigned16,<br>read-only | **Primary1Y**<br>Contains the normalized chromaticity value y for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary1Y attribute by the relationship. |

| Attribute | Type | Description |
|---|---|---|
| 0x0013 | unsigned8, read-only | **Primary1Intensity**<br>Contains a representation of the maximum intensity of this primary as defined in the Dimming Light Curve in the Ballast Configuration cluster (see 5.3 in [R4]), normalized such that the primary with the highest maximum intensity contains the value 0xfe. |
| 0x0015 | unsigned16, read-only | **Primary2X**<br>Contains the normalized chromaticity value x for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary2X attribute by the relationship. |
| 0x0016 | unsigned16, read-only | **Primary2Y**<br>Contains the normalized chromaticity value y for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary2Y attribute by the relationship. |
| 0x0017 | unsigned8, read-only | **Primary2Intensity**<br>Contains a representation of the maximum intensity of this primary as defined in the Dimming Light Curve in the Ballast Configuration cluster (see 5.3 in [R4]), normalized such that the primary with the highest maximum intensity contains the value 0xfe. |
| 0x0019 | unsigned16, read-only | **Primary3X**<br>Contains the normalized chromaticity value x for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary3X attribute by the relationship. |
| 0x001a | unsigned16, read-only | **Primary3Y**<br>Contains the normalized chromaticity value y for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary3Y attribute by the relationship. |
| 0x001b | unsigned8, read-only | **Primary3Intensity**<br>Contains a representation of the maximum intensity of this primary as defined in the Dimming Light Curve in the Ballast Configuration cluster (see 5.3 in [R4]), normalized such that the primary with the highest maximum intensity contains the value 0xfe. |
| 0x0020 | unsigned16, read-only | **Primary4X**<br>Contains the normalized chromaticity value x for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary4X attribute by the relationship. |
| 0x0021 | unsigned16, read-only | **Primary4Y**<br>Contains the normalized chromaticity value y for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary4Y attribute by the relationship. |
| 0x0022 | unsigned8, read-only | **Primary4Intensity**<br>Contains a representation of the maximum intensity of this primary as defined in the Dimming Light Curve in the Ballast Configuration cluster (see 5.3 in [R4]), normalized such that the primary with the highest maximum intensity contains the value 0xfe. |
| 0x0024 | unsigned16, read-only | **Primary5X**<br>Contains the normalized chromaticity value x for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary5X attribute by the relationship. |
| 0x0025 | unsigned16, read-only | **Primary5Y**<br>Contains the normalized chromaticity value y for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary5Y attribute by the relationship. |
| 0x0026 | unsigned8, read-only | **Primary5Intensity**<br>Contains a representation of the maximum intensity of this primary as defined in the Dimming Light Curve in the Ballast Configuration cluster (see 5.3 in [R4]), normalized such that the primary with the highest maximum intensity contains the value 0xfe. |
| 0x0028 | unsigned16, read-only | **Primary6X**<br>Contains the normalized chromaticity value x for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary6X attribute by the relationship. |
| 0x0029 | unsigned16, read-only | **Primary6Y**<br>Contains the normalized chromaticity value y for this primary, as defined in the CIE xyY Color Space. The value of x is related to the Primary6Y attribute by the relationship. |
| 0x002a | unsigned8, read-only | **Primary6Intensity**<br>Contains a representation of the maximum intensity of this primary as defined in the Dimming Light Curve in the Ballast Configuration cluster (see 5.3 in [R4]), normalized such that the primary with the highest maximum intensity contains the value 0xfe. |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Manufacturer-specific attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | bitmap8, read-only, persistent, preserved | **AdvancedOptions** [5] Advanced options for the ubisys Versalight engine that provide additional flags and options for fine-tuning color mixing if needed |

### 6.4.8.1. *AdvancedOptions* Attribute

The *AdvancedOptions* attribute is 8-bit unsigned integer (ZCL Data type: 0x08). It is readable and writable and retains its value across reboots (persistent storage) and normal factory resets (preserved). Its value defaults to zero (e.g. after upgrading from a firmware version prior to 1.4.1).

A combination of the following bit flags may be written to this attribute:

*Table 1. Each Entry in InputConfigurations Array*

| Flag | Bit(s) | Description |
|---|---|---|
| Don't use color for white | #0 (0x01) | When this bit is set, white tones in color temperature mode will not be composed of color primaries (red, green, blue, amber, turqoise, violet, other), provided that there are two white primaries (cool-white and warm-white) available, e.g. when driving an RGB+CCT LED strip. This will limit the color temperature range between the first and second white. This setting is ignored if there is just one white primary available, e.g. does not apply to an RGBW LED strip. |
| Don't use first white for color | #1 (0x02) | When this bit is set, the first white primary (typically, cool-white for a dual-white strip or neutral or warm-white for an RGBW strip) shall not contribute to colored light, i.e. when operating in CIE 1931 xy or hue/saturation mode. White primaries are beneficial for composing soft tones (pastel colors) and for most use-cases it is recommended to leave this option disabled, i.e. utilize the first white primary in color mode. |
| Don't use second white for color | #2 (0x04) | When this bit is set, the second white primary (typically, warm-white for a dual-white strip) shall not contribute to colored light, i.e. when operating in CIE 1931 xy or hue/saturation mode. White primaries are beneficial for composing soft tones (pastel colors) and for most use-cases it is recommended to leave this option disabled, i.e. utilize the second white primary in color mode. |
| RFU | #3…#7 (0xF8) | All other bits are reserved for future use and must be written as 0 and ignored when read. |

Cluster commands supported:

*Table 2. Table of supported commands of color control cluster*

| Command | Description |
|---|---|
| 0x00 | **Move to Hue** On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and then moves from its current hue to the value given in the Hue field. |
| 0x01 | **Move Hue** On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and then moves from its current hue in an up or down direction in a continuous fashion. |
| 0x02 | **Step Hue** On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and then moves from its current hue in an up or down direction by one step. |
| 0x03 | **Move to Saturation** On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and then moves from its current saturation to the value given in the Saturation field. |
| 0x04 | **Move Saturation** On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and then moves from its current saturation in an up or down direction in a continuous fashion. |
| 0x05 | **Step Saturation** On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and then moves from its current saturation in an up or down direction by one step. |
| 0x06 | **Move to Hue and Saturation** On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and then moves from its current hue and saturation to the values given in the Hue and Saturation fields. |

| Command | Description |
|---|---|
| 0x07 | **Move to Color**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x01 and then moves from its current color to the color given in the ColorX and ColorY fields. |
| 0x08 | **Move Color**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x01 and then moves from its current color in a continuous fashion according to the rates specified. |
| 0x09 | **Step Color**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x01 and then moves from its current color by the color step indicated. |
| 0x0A | **Move to Color Temperature**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x02 and then moves from its current color to the color given by the Color Temperature Mireds field. |
| 0x40 | **Enhanced Move to Hue**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and sets the EnhancedColorMode attribute to the value 0x03. LD6 then moves from its current enhanced hue to the value given in the Enhanced Hue field. |
| 0x41 | **Enhanced Move Hue**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and sets the EnhancedColorMode attribute to the value 0x03. LD6 then moves from its current enhanced hue in an up or down direction in a continuous fashion. |
| 0x42 | **Enhanced Step Hue**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and sets the EnhancedColorMode attribute to the value 0x03. LD6 then moves from its current enhanced hue in an up or down direction by one step. |
| 0x43 | **Enhanced Move to Hue and Saturation**<br>On receipt of this command, LD6 sets the ColorMode attribute to the value 0x00 and sets the EnhancedColorMode attribute to the value 0x03. LD6 then moves from its current enhanced hue and saturation to the values given in the enhanced hue and saturation fields. |
| 0x44 | **Color Loop Set**<br>Allows a color loop to be activated such that the color lamp cycles through its range of hues. |
| 0x47 | **Stop Move Step**<br>Allows Move to and Step commands to be stopped. |
| 0x4B | **Move Color Temperature**<br>Allows the color temperature of a lamp to be moved at a specified rate.<br>On receipt of this command, LD6 sets both the ColorMode and EnhancedColorMode attributes to 0x02. LD6 then moves from its current color temperature in an up or down direction in a continuous fashion. |
| 0x4C | **Step Color Temperature**<br>Allows the color temperature of LD6 to be stepped with a specified step size.<br>On receipt of this command, LD6 sets both the ColorMode and EnhancedColorMode attributes to 0x02. LD6 then moves from its current color temperature in an up or down direction by one step. |

### 6.4.9. Ballast Configuration Cluster (Server)

The ballast configuration cluster is used for configuring a lighting ballast. It allows configuration of e.g. a minimum and maximum level.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned8,<br>read-only | **PhysicalMinLevel**<br>The minimum dimming level supported by the hardware.<br>“0x01” |
| 0x0001 | unsigned8,<br>read-only | **PhysicalMaxLevel**<br>The maximum dimming level supported by the hardware.<br>“0xFE” |
| 0x0002 | bitmap8,<br>read-only | **BallastStatus**<br>This attribute is 8 bits in length and specifies the activity status of the ballast functions. LD6 supports fully operational ballast and all connected lamps in socket.<br>“0x00” |

| Attribute | Type | Description |
|---|---|---|
| 0x0010 | unsigned8, persistent | **MinLevel**<br>The minimum dimming level supported by the attached load. This can be set to a higher value to account for loads (such as CFLs), which cannot operate below a certain dimming level. It can also be used to make sure that an incandescent bulb emits visible light, such that a minimum on level is not confused with the “off” state.<br>“0x01” |
| 0x0011 | unsigned8, persistent | **MaxLevel**<br>The maximum dimming level supported by the attached load. This can be set to a lower value to enforce power saving in an office or public environment, e.g. in the hospitality sector.<br>“0xFE” |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

## 6.5. Application Endpoint #2 Primary Dimmer Switch

This is a Finding & Binding initiator endpoint.

### 6.5.1. Supported Device Types

Endpoint #2 is a Dimmer Switch (device type 0x0104):

#### 6.5.1.1. Dimmer Switch Device Type

Application endpoint #2 exposes the following inbound and outbound clusters:

| Cluster | Direction | Description |
|---|---|---|
| 0x0000 | Inbound (Server) | **Basic**<br>Provides basic information about the device, such as the manufacturer ID, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network. |
| 0x0003 | Inbound (Server) | **Identify**<br>Allows to put the endpoint into identify mode. Useful for identifying/locating devices and required for Finding & Binding. |
| 0x0003 | Inbound (Server) | **Identify**<br>Allows sending cluster-specific commands to target devices during Finding & Binding as initiator. |
| 0x0005 | Outbound (Client) | **Scenes**<br>Allows sending cluster-specific commands to target devices as required by the application. |
| 0x0006 | Outbound (Client) | **On/off**<br>Allows to send cluster-specific commands to target devices as required by the application. |
| 0x0008 | Outbound (Client) | **Level Control**<br>Allows to send cluster-specific commands to target devices as required by the application. |
| 0x0300 | Outbound (Client) | **Color Control**<br>Allows to send cluster-specific commands to target devices as required by the application. |
| 0xfc02 | Outbound (Client) | **Managed Input Cluster**<br>This is a manufacturer-specific cluster (ubisys proprietary). Allows to send cluster-specific commands to target devices as required by the application. Used for Matter-like automation buttons and automation switches. |

### 6.5.2. Basic Cluster (Server)

Please refer to Section 6.4.2 for details.

### 6.5.3. Identify Cluster (Server)

Please refer to Section 6.4.3 for details.

The identify cluster instance on this endpoint does **NOT** provide any visual or audible feedback. It is mainly present to comply with standards requirements.

### 6.5.4. Identify Cluster (Client)

The Identify cluster client takes on the role of the Finding & Binding initiator. Once activated, LD6 initiates the identification process by broadcasting identify query request command to detect target(s) operating in identify mode.

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Identify**<br>Puts the remote device into identify mode, or terminates identify mode. |
| 0x01 | **Query Identify**<br>Determines whether the remote device is currently identifying, and returns the remaining time if so. |

### 6.5.5. Scenes Cluster (Client)

This cluster does not use the binding table for managing command targets.
Instead, recall scene command templates include the group address and scene number to recall. The Device Setup cluster on the Device Management endpoint #232 can be used to configure the behavior of the cluster, i.e. which identifiers will be sent in the message body, when the input state changes. Using the device setup cluster, you can configure this endpoint either to be used with a push-button or a switch.

This cluster instance is intended for recalling scenes on the local and/or remote devices using the local inputs and is not bound when factory fresh.

Cluster commands transmitted:

| Command | Description |
|---|---|
| 0x05 | **Recall Scene**<br>Recalls a scene. The group address in the recall scene command payload is also used as the target for the command. |

### 6.5.6. On/off Cluster (Client)

This cluster uses the binding table for managing command targets.
When factory fresh, this cluster is bound to endpoint #1 to enable local control. This works even when the device is not commissioned yet.
In addition, the Device Setup cluster on the Device Management endpoint #232 can be used to configure the behavior of this cluster, i.e. which of the commands mentioned below will actually be sent, when the input state changes. Using the Device Setup cluster, you can configure this endpoint either to be used with a push-button (momentary switch, one stable position) or a rocker switch (two stable positions).

Cluster commands transmitted:

| Command | Description |
|---|---|
| 0x00 | **Turn off**<br>Turns the target device or group of devices off. |
| 0x01 | **Turn on**<br>Turns the target device or group of devices on. |
| 0x02 | **Toggle**<br>Toggles the target device state. Should not be used to control groups. |

### 6.5.7. Level Control Cluster (Client)

This cluster uses the binding table for managing command targets.
When factory fresh, this cluster is bound to endpoint #1 to enable local control. This works even when the device is not commissioned yet.
In addition, the Device Setup cluster on the Device Management endpoint #232 can be used to configure the behavior of the cluster, i.e. which of the commands mentioned below will actually be sent, when the input state changes. Using the Device Setup cluster, you can configure this endpoint to be used either with one or with two push-buttons.

Cluster commands transmitted:

| Command | Description |
|---|---|
| 0x05 | **Move with on/off**<br>Moves the level either up or down at a specified rate. Maintains the companion on/off cluster's OnOff attribute. |
| 0x07 | **Stop with on/off**<br>Stops any level change in progress due to a move, move to level, step or recall scene command. Maintains the companion on/off cluster's OnOff attribute. |

### 6.5.8. Color Control Cluster (Client)

This cluster uses the binding table for managing command targets.
When factory fresh, this cluster is not bound to any target device. The Device Setup cluster on the Device Management endpoint can be used to configure the behavior of this cluster, i.e. which of the commands mentioned below will actually be sent, when the input state changes. Using the Device Setup cluster, you can configure this endpoint to be used with one push-button.

Cluster commands transmitted:

| Command | Description |
|---|---|
| 0x0a | **Move To Color Temperature**<br>Requires the remote device to move from its current color temperature in Mireds to the color specified by the ColorTemperatureMireds field within the given TransitionTime. |
| 0x47 | **Stop Move Step**<br>Allow Move to and Step commands to be stopped. |
| 0x4b | **Move Color Temperature**<br>Requires the remote device to move the color temperature in Mireds either up or down at a specified rate. |

### 6.5.9. Managed Input Cluster (Client)

This is a ubisys manufacturer-specific (manufacturer code = 0x10f2) outbound cluster. It is used to send commands for automation buttons and automation switchs that mimic the Matter command set for generic switches.

These are the manufacturer-specific commands generated:

*Table 3. Actions supported by automation button*

| Action | Manufacturer Command Code |
|---|---|
| Button Initial press | 0x01 (payload 01) |
| Button Long press | 0x02 (payload 01) |
| Button Short release | 0x03 (payload 01) |
| Button Long release | 0x04 (payload 01) |

*Table 4. Actions supported by automation switch*

| Action | Manufacturer Command Code |
|---|---|
| Switch position '0' | 0x00 (payload 00) |
| Switch position '1' | 0x00 (payload 01) |

## 6.6. Application Endpoint #3 - Secondary Dimmer Switch

Similar to Endpoint #2, see Section 6.5 for details.

## 6.7. Application Endpoint #4 - Tertiary Dimmer Switch

Similar to Endpoint #2, see Section 6.5 for details.

## 6.8. Application Endpoint #5

Similar to Endpoint #1, see Section 6.4 for details.

## 6.9. Application Endpoint #6

Similar to Endpoint #1, see Section 6.4 for details.

## 6.10. Application Endpoint #7

Similar to Endpoint #1, see Section 6.4 for details.

## 6.11. Application Endpoint #8

Similar to Endpoint #1, see Section 6.4 for details.

## 6.12. Application Endpoint #9

Similar to Endpoint #1, see Section 6.4 for details.

## 6.13. Application Endpoint #232 - Device Management

This endpoint provides device management functions using standard Zigbee foundation paradigms and the Zigbee Cluster Library [R4]. You may use the standard ZCL frames to enumerate, read and write attributes, configure attribute reporting, invoke commands, etc.

The application endpoint exposes the following clusters:

| Cluster | Direction | Description |
|---|---|---|
| 0x0000 | Inbound (Server) | **Basic**<br>Provides basic information about the device, such as the manufacturer, vendor and model name, stack profile, ZCL version, production date, hardware revision etc. Allows a factory reset of attributes, without the device leaving the network. |
| 0x003D | Inbound (Server) | **Zigbee Direct Configuration**<br>Provides an interface for configuring Zigbee Direct to enable Zigbee Direct behavior on platforms that support Bluetooth and IEEE 802.15.4 concurrently (or quasi-concurrently). |
| 0xFC00 | Inbound (Server) | **Device Setup**<br>This is a manufacturer-specific cluster that provides basic configuration capabilities: It contains both the configuration of inputs and associated actions and the configuration of outputs.<br><br>The cluster allows the LD6 to assign up to 3 physical inputs associated with 3 endpoints and to control the behavior of inputs (i.e. permanent switches, push-button switches, normally open vs. normally closed, on/off, level control, scene selection, setting or adjusting color temperature , etc.). The cluster also allows LD6 to configure light devices with up to 6 output channels that can be controlled individually or combined as a tunable white or multi-colored light. |
| 0xFC57 | Inbound (Server) | **Works With All Hubs**<br>This is a manufacturer-specific cluster with the manufacturer code 0x1217. It provides the proprietary functionality of the Amazon Works With All Hubs (WWAH) server cluster. The goal of this cluster is to enable hubs to provide improved security, interoperability, and customer experience for devices supporting this cluster [R10]. |
| 0x0003 | Outbound (Client) | **Identify**<br>Allows to identify targets during Finding & Binding push-button commissioning. |
| 0x0019 | Outbound (Client) | **OTA Upgrade**<br>Pull-oriented firmware upgrade. Searches the network for a suitable server and allows the server to control all stages of the upgrade process, including which image to download, when to download, at what rate and when to install the downloaded image. |

### 6.13.1. Basic Cluster (Server)

Please refer to Section 6.4.2 for details.

### 6.13.2. Zigbee Direct Configuration Cluster (Server)

On centralized security networks, the Zigbee Direct Configuration cluster requires APS encryption with the trust center link-key.
Use unicast transmissions when sending commands to this cluster instance.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | Bitmap8, read-only, persistent | **InterfaceState**<br>The least significant bit of the attribute allows to disable the Zigbee Direct feature under application control, i.e. enabled when LSB = '1', disabled otherwise.<br>All other bits are reserved. |
| 0x0001 | Unsigned24, read-only, persistent | **AnonymousJoinTimeout**<br>Represents the timeout, in seconds, after a power-cycle or local manufacturer-specific stimulus during which a ZVD may establish a secure provisioning session with the ZDD using the Anonymous Well-Known Secret "ZigBeeAlliance18" when the Zigbee Network is open to accept new devices. See [R11] for details. |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Configure Zigbee Direct Interface**<br>Enables or disables the Zigbee Direct Interface on the device. |

| Command | Description |
|---|---|
| 0x01 | **Configure Zigbee Direct Anonymous Join Timeout**<br>Sets the anonymous join timeout. |

### 6.13.3. Device Setup Cluster (Server)

This cluster is a manufacturer-specific cluster with the cluster identifier 0xFC02 and the manufacturer code 0x10F2. It allows to change advanced device setup options, which are not covered by standard clusters defined in the Zigbee Cluster Library.

This ubisys-proprietary cluster provides two basic sets of configuration capabilities:

- Input Configuration
- Output Configuration

LD6 supports both, input and output configurations.

#### 6.13.3.1. Input Configuration

The Input Configuration set contains the manufacturer attributes summarized below.

*Table 5. Manufacturer Specific Attributes of the Input Configuration Set:*

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | Array of data8, persistent, preserved | **InputConfigurations**<br>This array of 8-bit data holds exactly one entry per physical device input and allows disabling the input or inverting the “normal” signal level (normally closed vs. normally open). |
| 0x0001 | array of raw data, persistent, preserved | **InputActions**<br>This array of raw data strings maps physical inputs to application endpoints on the device (for example, an on/off switch application) and translates action changes to appropriate cluster commands (e.g. on/off/toggle or move/stop). |

##### 6.13.3.1.1. *InputConfigurations* Attribute

The *InputConfigurations* attribute is an array (ZCL Data type: 0x48) of 8-bit data (ZCL Data type: 0x08). It is readable and writable and retains its value across reboots (persistent storage); when being written, the size of the attribute is not allowed to change, and the entries must be of the same type “8-bit data” (0x08). Each physical device input line has a one-to-one relation to a slot in this array.

Each entry of the array can take a combination of the following bit flags as input configurations:

*Table 6. Each Entry in InputConfigurations Array*

| Flag | Bit(s) | Description |
|---|---|---|
| Disable | #7<br>(0x80) | When this bit is set, the input is disabled. |
| Invert | #6<br>(0x40) | Determines the “active” level. The default is active-high, meaning that a high voltage level (within 10% of the supply voltage) translates to an active input, and a low voltage level translates to in inactive signal. If this bit is set, the input is “active-low”, instead. For normally open circuits, leave this bit clear; for normally closed circuits make sure this bit is set. |
| RFU | #5…#0<br>(0x3F) | All other bits are reserved for future use and must be written as 0 and ignored when read. |

For LD6 up to three physical device inputs are supported, and this relation is as follows: an array with three entries

*Table 7. InputConfigurations Entries on LD6*

| Entry | Description |
|---|---|
| 0x00 | **Configuration options for high/low-voltage physical input #1**<br>Defines the configuration for the physical input marked with a switch symbol and the digit “1”.<br>Factory default value of input configurations: 0x00 |
| 0x01 | **Configuration options for high/low-voltage physical input #2**<br>Defines the configuration for the physical input marked with a switch symbol and the digit “2”.<br>Factory default value of input configurations: 0x00 |
| 0x02 | **Configuration options for high/low-voltage physical input #3**<br>Defines the configuration for the physical input marked with a switch symbol and the digit “3”.<br>Factory default value of input configurations: 0x00 |

For LD6 in connection with three physical inputs, this example shows the *InputConfigurations* attribute value:

```
48          Data type: 0x48 (array)
08          Element type: 0x08 (8-Bit Data)
03 00       Elements number: 0x0003 (3 entries)

00          Entry #0: 0x00 (default value: enable input #1)
00          Entry #1: 0x00 (default value: enable input #2)
00          Entry #2: 0x00 (default value: enable input #3)
```

#### 6.13.3.1.2. *InputActions* Attribute

This *InputActions* attribute is an array (ZCL Data type: 0x48) of raw binary data (ZCL Data type: 0x41). It is readable and writable and retains its value across reboots (persistent storage); when being written, the elements must be of the same type “raw data” (0x41).

This attribute contains instructions, a kind of micro-code, which allow flexible reconfiguration of the commands sent in response to activity on the physical inputs.

Each entry of the array takes the following form:

*Table 8. Each Entry in InputActions Array*

| Field | Data Type | Description |
|---|---|---|
| InputAndOptions | unsigned8 | A four-bit input index in the LSBs and four option flags in the MSBs |
| Transition | unsigned8 | Specifies the action transition to which this particular instruction applies |
| Endpoint | unsigned8 | Each instruction must be assigned to a suitable source endpoint in the local device. Acceptable endpoints for LD6 are Endpoints #2, #3 and #4. |
| ClusterID | unsigned16 | The 16-bit cluster ID that will be used to transmit the ZCL command payload. e.g. 0x0006 for the On/Off cluster. |
| CommandTemplate | raw data | A variable length ZCL command template, which will be sent from the source endpoint specified above, using the cluster ID specified above.<br>e.g. for the On/Off cluster, the command temple can be a simple Off (0x00), On (0x01) or Toggle (0x02) command. There might be code in the device that modifies the command template according to intrinsic rules or option flag settings.<br>A manufacturer-specific command template in terms of automation switch and automation push button is also supported.<br>Notice that this is the raw payload and does not include a leading length byte. |

- *InputAndOptions* Field

  The 4 LSBs of this field contains the physical input number, while the 4 MSBs of this field includes the input and option flags.

| Field | Bit(s) | Description |
|---|---|---|
| RFU | #7…#5 (0xE0) | These bits are reserved for future use and must be written as 0 and ignored when read. |
| flagManufacturerSpecific | #4 (0x10) | When this bit is set, the instructions for one of the two manufacturer-specific input configurations:<br>- Automation Button<br>- Automation Switch<br>will be selected accordingly.<br>Please refer to Section 6.5.9 for the related manufacturer-specific Cluster ID and supported commands. |
| inputMask | #3…#0 (0x0f) | In the lower four bits, the physical input number is contained, which indexes into the InputConfigurations array. For example, a value of 0 identifies the first physical input.<br>For LD6, this field may take the values 0, 1 and 2:<br>- 0x00: the first physical input<br>- 0x01: the second physical input<br>- 0x02: the third physical input |

- *Transition* Field

  The Transition field specifies when (i.e. under which conditions/circumstances) this particular instruction shall be executed.

| Field | Bit(s) | Description |
|---|---|---|
| HasAlternate | #7 (0x80) | When this bit is set, there is another instruction that shall be executed in alternating order with the current one, each time the other conditions are met. In this case two entries must exist, one with the Alternate bit clear, and one with the Alternate bit set. |
| Alternate | #6 (0x40) | If this bit is set, this is the alternate instruction of a pair of two instructions which are executed in alternating order. If this flag is set, “HasAlternate” must be set as well. |
| RFU | #5…#4 (0x30) | These bits are reserved for future use and must be written as 0 and ignored when read. |
| Initial State | #3…#2 (0x0C) | These bits indicate the initial state of the input:<br>00b: Ignore: Don’t care<br>01b: Pressed: The input was initially active (for less than a second)<br>10b: Kept pressed: The input has initially been active (for more than a second)<br>11b: Released: The input was initially inactive |
| Final State | #1…#0 (0x03) | These bits indicate the final state of the input:<br>00b: Ignore: Don’t care<br>01b: Pressed: The input is active (for less than a second)<br>10b: Kept pressed: The input is still active (for more than a second)<br>11b: Released: The input is inactive |

  The “Ignore: Don’t care” states are no real states, but rather placeholders for **any** other state. At any given time the input is either released, pressed or kept pressed.

- *Endpoint* Field

  Identifies the source endpoint that will be used in the outgoing ZCL command frame. Notice, that you should only use an endpoint, which hosts a cluster with the appropriate cluster ID and role (client vs. server).
  The acceptable endpoints for LD6 are Endpoints #2, #3 and #4.

- *ClusterID* Field

  Specifies the cluster ID that will be used in the outgoing ZCL command frame. The cluster ID must be an “operational” cluster, according to ZCL definitions.

  LD6 supports following clusters for *InputActions* attribute configuration:

  - 0x0005: the Scenes cluster

- 0x0006: the On/Off cluster
- 0x0008: the Level Control cluster
- 0x0300: the Color Control cluster
- 0xFC02: the manufacturer-specific Managed Input cluster
- etc.

- *CommandTemplate* Field

  This is the template of a ZCL command frame payload, which will be sent to the target device once the instruction is selected for execution by the universal input logic unique to ubisys Zigbee devices. This can be a simple toggle command, a more complex move or recall scene command, or a manufacturer-specific command (See Section 6.5.9).

  Notice that LD6 will use its binding table to determine the target for the command. The command may be sent to one or more targets, each being a group (specified as a group address) or a single application instance (specified as IEEE EUI-64 address and endpoint number)

#### 6.13.3.2. Input Configuration Patterns and Examples

Currently LD6 supports a diverse set of switch usage patterns. Please refer to the Zigbee Device Physical Input Configurations - Integrator's Guide for more details and examples.

Below is an example input configuration for LD6, as stored in the *InputActions* attribute. This configuration illustrates three use-cases: single-button dimming and on/off, single-button color temperature control, and single-button scene recall for two scenes.

```
41          element type: 0x41 (raw data)
0a 00       element count: 0x000a (10 entries)


06          Element #1: six bytes
00          InputAndOptions: 0x00 (the first physical input)
07          Transition: pressed -> released
02          Source: Endpoint #2 (hosts primary on/off client cluster on LD6)
06 00       Cluster ID: 0x0006 (On/Off Cluster)
02          ZCL Command Template: Toggle

08          Element #2: eight bytes
00          InputAndOptions: 0x00 (the first physical input)
86          Transition: pressed -> kept pressed, has alternate
02          Source: Endpoint #2 (hosts primary level control client cluster
            on LD6)
08 00       Cluster ID: 0x0008 (Level Control Cluster)
05 00 32    ZCL Command Template: Move with on/off, upwards, rate = 50

08          Element #3: eight bytes
00          InputAndOptions: 0x00 (the first physical input)
C6          Transition: pressed -> kept pressed, is alternate
02          Source: Endpoint #2 (hosts primary level control client cluster
            on LD6)
08 00       Cluster ID: 0x0008 (Level Control Cluster)
05 01 32    ZCL Command Template: Move with on/off, downwards, rate = 50

06          Element #4: six bytes
```

```
00          InputAndOptions: 0x00 (the first physical input)
0B          Transition: kept pressed -> released
02          Source: Endpoint #2 (hosts primary level control client cluster
            on LD6)
08 00       Cluster ID: 0x0008 (Level Control Cluster)
07          ZCL Command Template: Stop with on/off


0a          Element #5: ten bytes
01          InputAndOptions: 0x01 (the second physical input)
07          Transition: pressed -> released
03          Source: Endpoint #3 (hosts secondary color control client cluster
            on LD6)
00 03       Cluster ID: 0x0300 (Color Control)
0a          ZCL Command Template: Move to Color Temperature,
fa 00       ColorTemperatureMireds = 250,
0a 00       TransitionTime = 10s

0d          Element #6: thirteen bytes
01          InputAndOptions: 0x01 (the second physical input)
86          Transition: pressed -> kept pressed, has alternate
03          Source: Endpoint #3 (hosts secondary color control client cluster
            on LD6)
00 03       Cluster ID: 0x0300 (Color Control)
4b          ZCL Command Template: Move Color Temperature,
01          MoveMode = Up
19 00       MoveRate = 25
00 00       ColorTemperatureMinimumMireds = 0x0000
00 00       ColorTemperatureMaximumMireds = 0x0000

0d          Element #7: thirteen bytes
01          InputAndOptions: 0x01 (the second physical input)
C6          Transition: pressed -> kept pressed, is alternate
03          Source: Endpoint #3 (hosts secondary color control client cluster
            on LD6)
00 03       Cluster ID: 0x0300 (Color Control)
4b          ZCL Command Template: Move Color Temperature,
03          MoveMode = Down
19 00       MoveRate = 25
00 00       ColorTemperatureMinimumMireds = 0x0000
00 00       ColorTemperatureMaximumMireds = 0x0000

06          Element #8: six bytes
01          InputAndOptions: 0x01 (the second physical input)
0B          Transition: kept pressed -> released
03          Source: Endpoint #3 (hosts secondary color control client cluster
            on LD6)
00 03       Cluster ID: 0x0300 (Color Control)
47          ZCL Command Template: Stop Move Step


09          Element #9: nine bytes
02          InputAndOptions: 0x02 (the third physical input)
0d          Transition: released, -> pressed
04          Source: Endpoint #4 (hosts tertiary scenes client cluster on LD6)
05 00       Cluster ID: 0x0005 (Scenes)
05          ZCL Command Template: Recall Scene
02 00       GroupID = 0x0002
00          SceneID = 0x00
```

```
09          Element #10: nine bytes
02          InputAndOptions: 0x02 (the third physical input)
03          Transition: any -> released
04          Source: Endpoint #4 (hosts tertiary scenes client cluster on LD6)
05 00       Cluster ID: 0x0005 (Scenes)
05          ZCL Command Template: Recall Scene
02 00       GroupID = 0x0002
01          SceneID = 01
```

For the first physical input (InputAndOptions: 0x00), the configuration *Single Momentary Switch (Push Button) as Dimmer Switch* is applied that allows to control a dimmer with one push button. A short press will toggle the light on/off, while a longer press starts dimming up or down (alternating) in order to allow adjusting the brightness with the button. Dimming stops, when the button is released.

For the second physical input (InputAndOptions: 0x01), the configuration *Single Momentary Switch (Push Button) as White Tone Button* is applied that allows to control a tunable white light with one push button. A short press will transition the color temperature to the specified value in mireds, whereas a longer press starts moving color temperature up or down (alternating) in order to allow adjusting the white tone between cool and warm. When the button is released the color temperatue will stop changing.

For the third physical input (InputAndOptions: 0x02), the configuration *Single Stationary Switch as Scene Selector Switch* is applied that allows to recall a previously stored scene. The input is as a stationary switch with two stable positions, each stable position is assigned to a scene. Two separate scenes may be recalled from a single button. For the LD6, a scene setting would comprise on/off, level and color states. Other products, including things that are no lights, could be added to the same scene, e.g. window blings, power sockets, or other lights etc.

The ubisys Smart Home app for iOS and Android recognizes and generates a certain set of instructions, and lets users pick those easily. You may request an up-to-date list of optimized instructions by mailing support@ubisys.de and asking for these recommended instructions for typical use cases. But avoid using other instructions.

#### 6.13.3.3. Output Configuration

The Output Configuration is used to inform the LD6 about the light sources that are attached and the way they are intended to be used and partitioned.

In general the Output Configuration set contains the manufacturer-specific attributes summarized below.

*Table 9. Manufacturer Specific Attributes of the Output Configurations Set:*

| Attribute | Type | Description |
|---|---|---|
| 0x0010 | Array of raw data, persistent, preserved | **OutputConfigurations**<br>This array of raw data strings maps physical outputs to application endpoints on the device and specifies output configurations e.g. color or temperature color settings depending on the supported device type on the associated endpoints. |
| 0x0011 | Bitmap16, read-only | **OutputEndpoints**<br>Each bit of the attribute denotes a dynamic output application endpoint that can be enabled/assigned via the output configuration. |

The LD6 is a multi-channel dimmer with six output channels that can be controlled individually or combined as tunable white (brightness or color temperature) or full color lights. Depending on the

light output configuration, the LD6 creates up to six individual lighting application endpoints to represent the lights attached to its six output channels.

#### 6.13.3.3.1. *OutputConfigurations* Attribute

The *OutputConfigurations* attribute is an array (ZCL data type 0x48) of raw binary data (ZCL data type 0x41). It is readable and writable and retains its value across reboots (persistent storage); when being written, the entries must be of the same type “raw data” (0x41).

Each entry of the *OutputConfigurations* array contains the configuration of one channel. It defines the light application endpoint and functionality of the channel including details about the primary, i.e. x, y position in the CIE 1931 color space and the (relative/normalized) luminous flux Y.

It is possible to use more than one channel, e.g. to achieve higher output power. Note that these channels act as s single channel from the Zigbee side. To control multiple channels individually, create an additional ApplicationConfiguration element.

Each entry of the the array shall take the following form:

*Table 10. Each Entry in OutputConfigurations*

| Field | Data Type | Description |
|---|---|---|
| EndpointAndFunction | unsigned8 | Bit fields for an associated end-point and the assigned function/mode.<br>4 MSBs of this field: Endpoint mask<br>4 LSBs of this field: Function/Mode mask. |
| Flux (normalized) | unsigned8 | The normalized (relative) flux, i.e. intensity of the primary. |
| CIE 1931 x | unsigned16 | The x coordinate of the primary in the CIE 1931 xy space. |
| CIE 1931 y | unsigned16 | The y coordinate of the primary in the CIE 1931 xy space. |

The LD6 is a multi-channel LED controller with six output channels. The number of entries of the OutputConfigurations array is therefore always equal to six.

- *EndpointAndFunction* Field

  The *EndpointAndFunction* field is used to assign the physical output channel to logical light endpoint; in addition, it also tells the ubisys Versalight engine what kind of primary is attached.

| Field | Bit(s) | Description |
|---|---|---|
| Endpoint | #7…#4 (0xF0) | An endpoint mask that indicates the application endpoint of this channel<br>0001b(1): endpoint #1<br>0101b(5): endpoint #5<br>0110b(6): endpoint #6<br>0111b(7): endpoint #7<br>1000b(8): endpoint #8<br>1001b(9): endpoint #9<br>Other values on the 4 MSBs of *EndpointAndFunction* must not be used. |
| Function | #3…#0 (0x0F) | A function mask that specifies the function of this channel<br>0000b(0): M - mono<br>0001b(1): CW - first (or only) white, typically cool white or neutral white<br>0010b(2): WW - second white (typically, warm White)<br>0011b(3): R - Red<br>0100b(4): G - Green<br>0101b(5): B - Blue<br>0110b(6): A - Amber<br>0111b(7): T - Turquoise<br>1000b(8): V - Violet<br>1001b(9): F - other arbitrary "free" color<br>Other values on the 4 LSBs of *EndpointAndFunction* are void, i.e. unused. |

- *Primary-Intensity* Field

  The *Primary-Intensity* Field contains a representation of the maximum intensity of this primary for full color lights or tunable white lights (color temperature). The range of this field is 0..0xFE.

  The normalized intensity value *i* SHALL be related to the *Primary-Intensity* value by the relationship

  *i* = *Primary-Intensity* / 254 (*Primary-Intensity* in the range 0 to 254 inclusive)

  If equal to UINT8_MAX, it denotes either an invalid value or not applicable value. A value of 0 indicates that the primary information is void, i.e. not used.

- *CIE 1931 x* Field

  The *CIE 1931 x* field contains the value of the primary x coordinate in the CIE1931 xyY space. The range is 0 to 65279 inclusive.

  The normalized x chromaticity value of the primary SHALL be related to the *CIE 1931 x* value by the relationship

  x = *CIE 1931 x* / 65536 (*CIE 1931 x* in the range 0 to 65279 inclusive)

  If equal to UINT16_MAX, it denotes an invalid/unknown value.

- *CIE 1931 y* Field

  The *CIE 1931 y* field contains the value of the primary y coordinate in the CIE1931 xyY space. The range is 0 to 65279 inclusive.

  The normalized y chromaticity value of the primary SHALL be related to the *CIE 1931 y* value by the relationship

  y = *CIE 1931 y* / 65536 (*CIE 1931 y* in the range 0 to 65279 inclusive)

  If equal to UINT16_MAX, it denotes an invalid/unknown value.

#### 6.13.3.3.2. *OutputEndpoints* Attribute

The manufacturer-specific attribute *OutputEndpoints* is a bitmap attribute that indicates dynamic output application endpoints that can be enabled/assigned via the output configuration. This is useful for sharing a single client side application logic that can be used configure the light output configuration for various products that employ the ubisys Versalight engine, including the LD6, but also the M7B-Q95 module, white-label products, etc.

LD6 has the default *OutputEndpoints* value 0x03e2 with details:

| Bitmap | Endpoint |
|---|---|
| Bit 1 | Endpoint #1 |
| Bit 5 | Endpoint #5 |
| Bit 6 | Endpoint #6 |
| Bit 7 | Endpoint #7 |
| Bit 8 | Endpoint #8 |

| Bitmap | Endpoint |
|---|---|
| Bit 9 | Endpoint #9 |

These are the six endpoints that are configurable to represent one up to six individually addressable logical lights.

### 6.13.3.4. Output Configurations Patterns and Examples

The LD6 supports a diverse set of output configurations. Bear in mind that the Versalight engine assumes a certain order of primaries and deviations might result in false color reproduction. It is recommended that users choose one of the configurations provided below to implement their use-case:

- Output Configuration: One Light Source

| Item | Device Type | Description |
|---|---|---|
| 1 | **Endpoint #1**:<br>Dimmable Light | **1 x Dimmable (mono)**<br>Endpoint #1 - Output Channel 1:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 10 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 2 | **Endpoint #1**:<br>Color Temperature Light | **1 x White (Cool/Warm)**<br>Endpoint #1 - Output Channel 1:C, Output Channel 2:W<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 11 fe 42 50 d9 52<br>06 12 fe b9 75 1d 69<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 3 | **Endpoint #1**:<br>Extended Color Light | **1 x Color**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 4 | **Endpoint #1**:<br>Extended Color Light | **1 x Color and White**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B, Output Channel 4:W<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 11 fe 64 61 72 60<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |

| Item | Device Type | Description |
|---|---|---|
| 5 | **Endpoint #1**:<br>Extended Color Light | **1 x Color and White (Cool/Warm)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B, Output Channel 4:C, Output Channel 5:W<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 11 fe 42 50 d9 52<br>06 12 fe b9 75 1d 69<br>06 00 ff ff ff ff ff |
| 6 | **Endpoint #1**:<br>Extended Color Light | **1 x Extended Color Gamut**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:A, Output Channel 3:G, Output Channel 4:T, Output Channel 5:B, Output Channel 6:V<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 45 86 b1 ef 4e<br>06 16 c6 59 9a 80 65<br>06 14 fe 39 1d 82 d3<br>06 17 b4 9e 0b 83 4b<br>06 15 4d c6 1f cc 0e<br>06 18 6c 2d 2c 3a 01 |

In the case listed above, the endpoints #5, #6, #7, #8, #9 shall be disabled or hidden.

- Output Configuration: Two Light Sources

| Item | Device Type | Pattern Description |
|---|---|---|
| 1 | **Endpoint #1, #5**:<br>Dimmable Light | **2 x Dimmable (mono)**<br>Endpoint #1 - Output Channel 1:M<br>Endpoint #5 - Output Channel 2:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 10 ff ff ff ff ff<br>06 50 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 2 | **Endpoint #1, #5**:<br>Color Temperature Light | **2 x White (Cool/Warm)**<br>Endpoint #1 - Output Channel 1:C, Output Channel 2:W<br>Endpoint #5 - Output Channel 3:C, Output Channel 4:W<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 11 fe 42 50 d9 52<br>06 12 fe b9 75 1d 69<br>06 51 fe 42 50 d9 52<br>06 52 fe b9 75 1d 69<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |

| Item | Device Type | Pattern Description |
|---|---|---|
| 3 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5**:<br>Color Temperature Light | **1 x RGBW and 1 x White(Cool/Warm)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B, Output Channel 4:W<br>Endpoint #5 - Output Channel 5:C, Output Channel 6:W<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 11 fe 64 61 72 60<br>06 51 fe 42 50 d9 52<br>06 52 fe b9 75 1d 69 |
| 4 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5**:<br>Color Temperature Light | **1 x Color and 1 x White(Cool/Warm)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B<br>Endpoint #5 - Output Channel 5:C, Output Channel 6:W<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 00 ff ff ff ff ff<br>06 51 fe 42 50 d9 52<br>06 52 fe b9 75 1d 69 |
| 5 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5**:<br>Dimmable Light | **1 x Color and 1 x Dimmable (Mono)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B<br>Endpoint #5 - Output Channel 4:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 50 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 6 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5**:<br>Dimmable Light | **1 x RGBW and 1 x Dimmable (Mono)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B, Output Channel 4:W<br>Endpoint #5 - Output Channel 5:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 11 fe 64 61 72 60<br>06 50 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 7 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5**:<br>Dimmable Light | **1 x RGBCW and 1 x Dimmable (Mono)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B, Output Channel 4:C, Output Channel 5:W<br>Endpoint #5 - Output Channel 6:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 11 fe 42 50 d9 52<br>06 12 fe b9 75 1d 69<br>06 50 ff ff ff ff ff |

| Item | Device Type | Pattern Description |
|---|---|---|
| 8 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5**:<br>Extended Color Light | **2 x Color**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B<br>Endpoint #5 - Output Channel 4:R, Output Channel 5:G, Output Channel 6:B<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 53 47 06 b1 ef 4e<br>06 54 a0 39 1d 82 d3<br>06 55 42 c6 1f cc 0e |

In the case listed above, the endpoints #6, #7, #8, #9 shall be disabled or hidden.

- Output Configuration: Three Light Sources

| Item | Device Type | Pattern Description |
|---|---|---|
| 1 | **Endpoint #1, #5, #6**:<br>Dimmable Light | **3 x Dimmable (mono)**<br>Endpoint #1 - Output Channel 1:M<br>Endpoint #5 - Output Channel 2:M<br>Endpoint #6 - Output Channel 3:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 10 ff ff ff ff ff<br>06 50 ff ff ff ff ff<br>06 60 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 2 | **Endpoint #1, #5, #6**:<br>Color Temperature Light | **3 x White (Cool/Warm)**<br>Endpoint #1 - Output Channel 1:C, Output Channel 2:W<br>Endpoint #5 - Output Channel 3:C, Output Channel 4:W<br>Endpoint #6 - Output Channel 5:C, Output Channel 6:W<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 11 fe 42 50 d9 52<br>06 12 fe b9 75 1d 69<br>06 51 fe 42 50 d9 52<br>06 52 fe b9 75 1d 69<br>06 61 fe 42 50 d9 52<br>06 62 fe b9 75 1d 69 |
| 3 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5, #6**:<br>Dimmable Light | **1 x Color and 2 x dimmable (mono)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B<br>Endpoint #5 - Output Channel 4:M<br>Endpoint #6 - Output Channel 5:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 50 ff ff ff ff ff<br>06 60 ff ff ff ff ff<br>06 00 ff ff ff ff ff |

| Item | Device Type | Pattern Description |
|---|---|---|
| 4 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5**:<br>Color Temperature Light<br><br>**Endpoint #6**:<br>Dimmable Light | **1 x Color, 1 x White(Cool/Warm) and 1 x Dimmable (mono)**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B<br>Endpoint #5 - Output Channel 5:C, Output Channel 6:W<br>Endpoint #6 - Output Channel 4:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 60 ff ff ff ff ff<br>06 51 fe 42 50 d9 52<br>06 52 fe b9 75 1d 69 |
| 5 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5, #6**:<br>Dimmable Light | **1 x RGBW and 2 x Dimmable**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B, Output Channel 4:W<br>Endpoint #5 - Output Channel 5:M<br>Endpoint #6 - Output Channel 6:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 11 fe 64 61 72 60<br>06 50 ff ff ff ff ff<br>06 60 ff ff ff ff ff |

In the case listed above, the endpoints #7, #8, #9 shall be disabled or hidden.

- Output Configuration: Four Light Sources

| Item | Device Type | Pattern Description |
|---|---|---|
| 1 | **Endpoint #1, #5-#7**:<br>Dimmable Light | **4 x Dimmable (mono)**<br>Endpoint #1 - Output Channel 1:M<br>Endpoint #5 - Output Channel 2:M<br>Endpoint #6 - Output Channel 3:M<br>Endpoint #7 - Output Channel 4:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 10 ff ff ff ff ff<br>06 50 ff ff ff ff ff<br>06 60 ff ff ff ff ff<br>06 70 ff ff ff ff ff<br>06 00 ff ff ff ff ff<br>06 00 ff ff ff ff ff |
| 2 | **Endpoint #1**:<br>Extended Color Light<br><br>**Endpoint #5-#7**:<br>Dimmable Light | **Color and 3 x Dimmable**<br>Endpoint #1 - Output Channel 1:R, Output Channel 2:G, Output Channel 3:B<br>Endpoint #5 - Output Channel 2:M<br>Endpoint #6 - Output Channel 3:M<br>Endpoint #7 - Output Channel 4:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 13 47 06 b1 ef 4e<br>06 14 a0 39 1d 82 d3<br>06 15 42 c6 1f cc 0e<br>06 50 ff ff ff ff ff<br>06 60 ff ff ff ff ff<br>06 70 ff ff ff ff ff |

In the case listed above, the endpoints #8, #9 shall be disabled or hidden.

- Output Configuration: Five Light Sources

| Item | Device Type | Pattern Description |
|---|---|---|
| 1 | **Endpoint #1, #5-#8**: Dimmable Light | **5 x Dimmable (mono)**<br>Endpoint #1 - Output Channel 1:M<br>Endpoint #5 - Output Channel 2:M<br>Endpoint #6 - Output Channel 3:M<br>Endpoint #7 - Output Channel 4:M<br>Endpoint #8 - Output Channel 5:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 10 ff ff ff ff ff<br>06 50 ff ff ff ff ff<br>06 60 ff ff ff ff ff<br>06 70 ff ff ff ff ff<br>06 80 ff ff ff ff ff<br>06 00 ff ff ff ff ff |

In the case listed above, the endpoint #9 shall be disabled or hidden.

- Output Configuration: Six Light Sources

| Item | Device Type | Pattern Description |
|---|---|---|
| 1 | **Endpoint #1, #5-#9**: Dimmable Light | **6 x Dimmable (mono)**<br>Endpoint #1 - Output Channel 1:M<br>Endpoint #5 - Output Channel 2:M<br>Endpoint #6 - Output Channel 3:M<br>Endpoint #7 - Output Channel 4:M<br>Endpoint #8 - Output Channel 5:M<br>Endpoint #9 - Output Channel 6:M<br><br>**Reference Configuration**:<br>48 41 06 00<br>06 10 ff ff ff ff ff<br>06 50 ff ff ff ff ff<br>06 60 ff ff ff ff ff<br>06 70 ff ff ff ff ff<br>06 80 ff ff ff ff ff<br>06 90 ff ff ff ff ff |

Below are some application examples demonstrating relevant output configurations.

- Example 1: One light source - 1 x Dimmable (mono)

  Following output configurations define the relation of 1 application endpoint to 1 output channel. The supported application endpoint of LD6 is attached to a monochromatic (M) and dimmable light source, the other 5 output channels are not configured, i.e. unused.

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
10          EndpointAndFunction: 0x10 - endpoint #1, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff

06          Element #2: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #3: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #4: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #5: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #6: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)
```

Note that for an output channel *n* (*n*=2, .., 6), which is not specified with the light output configuration, the associated entry is set as follows

```
06          Element #n: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)
```

- Example 2: One light source - 1 x White (Cool/Warm)

  Following output configurations define the relation of 1 application endpoint to the 2 output channels. The endpoint is attached to a tunable white(CW/WW) light source. The other 4 output channels are not configured, i.e. unused.

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
11          EndpointAndFunction: 0x11 - endpoint #1, cool white (CW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
42 50       CIE 1931 x: 0x5042 (x_CW = 20546/65536 = 0.31352)
d9 52       CIE 1931 y: 0x52d9 (y_CW = 21209/65536 = 0.32363)

06          Element #2: six bytes
12          EndpointAndFunction: 0x12 - endpoint #1, warm white (WW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
b9 75       CIE 1931 x: 0x75b9 (x_WW = 30137/65536 = 0.45986)
1d 69       CIE 1931 y: 0x691d (y_WW = 26909/65536 = 0.41060)

06          Element #3: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #4: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #5: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #6: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)
```

- Example 3: One light source - 1 x Color

  Following output configurations define the relation of 1 application endpoint to the 3 output channels and the endpoint is attached to a multi-colored(RGB) light source. The other 3 output channels are not configured, i.e. unused.

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
13          EndpointAndFunction: 0x13 - endpoint #1, red (R)
47          Flux (normalized): 0x47 = 71 (71/254 = 0.28)
06 b1       CIE 1931 x: 0xb106 = 45318 (x_R = 45318/65536 = 0.69147)
ef 4e       CIE 1931 y: 0x4eef = 20207 (y_R = 20207/65536 = 0.3083)

06          Element #2: six bytes
14          EndpointAndFunction: 0x14 - endpoint #1, green (G)
a0          Flux (normalized): 0xa0 = 163 (163/254 = 0.63)
39 1d       CIE 1931 x: 0x1d39 = 7481 (x_G = 7481/65536 = 0.1146)
82 d3       CIE 1931 y: 0xd382 = 54146 (y_G = 54146/65536 = 0.8262)

06          Element #3: six bytes
15          EndpointAndFunction: 0x15 - endpoint #1, blue (B)
42          Flux (normalized): 0x42 = 66 (66/254 = 0.26)
c6 1f       CIE 1931 x: 0x1fc6 = 8134 (x_B = 8134/65536 = 0.1241)
cc 0e       CIE 1931 y: 0x0ecc = 3788 (y_B = 3788/65536 = 0.0578)

06          Element #4: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #5: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #6: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)
```

- Example 4: One light source - 1 x Color and neutral white

  Following output configurations define the relation of 1 application endpoint to the 4 output channels and the endpoint is attached to a light source supporting both multi-color(RGB) and tunable white(Brightness) as well. The other 2 output channels are not configured, i.e. unused.

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
13          EndpointAndFunction: 0x13 - endpoint #1, red (R)
47          Flux (normalized): 0x47 = 71 (71/254 = 0.28)
06 b1       CIE 1931 x: 0xb106 = 45318 (x_R = 45318/65536 = 0.69147)
ef 4e       CIE 1931 y: 0x4eef = 20207 (y_R = 20207/65536 = 0.3083)

06          Element #2: six bytes
14          EndpointAndFunction: 0x14 - endpoint #1, green (G)
a0          Flux (normalized): 0xa0 = 163 (163/254 = 0.63)
39 1d       CIE 1931 x: 0x1d39 = 7481 (x_G = 7481/65536 = 0.1146)
82 d3       CIE 1931 y: 0xd382 = 54146 (y_G = 54146/65536 = 0.8262)

06          Element #3: six bytes
15          EndpointAndFunction: 0x15 - endpoint #1, blue (B)
42          Flux (normalized): 0x42 = 66 (66/254 = 0.26)
c6 1f       CIE 1931 x: 0x1fc6 = 8134 (x_B = 8134/65536 = 0.1241)
cc 0e       CIE 1931 y: 0x0ecc = 3788 (y_B = 3788/65536 = 0.0578)

06          Element #4: six bytes
11          EndpointAndFunction: 0x11 - endpoint #1, first/only white (W)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
64 61       CIE 1931 x: 0x6164 (x_NW = 24932/65536 = 0.38043)
72 60       CIE 1931 y: 0x6072 (y_NW = 24690/65536 = 0.37673)

06          Element #5: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)

06          Element #6: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)
```

- Example 5: One light source - 1 x Color and White (Cool/Warm)

  Following output configurations define the relation of 1 application endpoint to to the 5 output channels and the endpoint is attached to a light source supporting both multi-color(RGB) and color-temperature tunable white. The other 1 output channel is not configured, i.e. unused.

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
13          EndpointAndFunction: 0x13 - endpoint #1, red (R)
47          Flux (normalized): 0x47 = 71 (71/254 = 0.28)
06 b1       CIE 1931 x: 0xb106 = 45318 (x_R = 45318/65536 = 0.69147)
ef 4e       CIE 1931 y: 0x4eef = 20207 (y_R = 20207/65536 = 0.3083)

06          Element #2: six bytes
14          EndpointAndFunction: 0x14 - endpoint #1, green (G)
a0          Flux (normalized): 0xa0 = 163 (163/254 = 0.63)
39 1d       CIE 1931 x: 0x1d39 = 7481 (x_G = 7481/65536 = 0.1146)
82 d3       CIE 1931 y: 0xd382 = 54146 (y_G = 54146/65536 = 0.8262)

06          Element #3: six bytes
15          EndpointAndFunction: 0x15 - endpoint #1, blue (B)
42          Flux (normalized): 0x42 = 66 (66/254 = 0.26)
c6 1f       CIE 1931 x: 0x1fc6 = 8134 (x_B = 8134/65536 = 0.1241)
cc 0e       CIE 1931 y: 0x0ecc = 3788 (y_B = 3788/65536 = 0.0578)

06          Element #4: six bytes
11          EndpointAndFunction: 0x11 - endpoint #1, first white (CW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
42 50       CIE 1931 x: 0x5042 (x_CW = 20546/65536 = 0.31352)
d9 52       CIE 1931 y: 0x52d9 (y_CW = 21209/65536 = 0.32363)

06          Element #5: six bytes
12          EndpointAndFunction: 0x12 - endpoint #1, second white (WW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
b9 75       CIE 1931 x: 0x75b9 (x_WW = 30137/65536 = 0.45986)
1d 69       CIE 1931 y: 0x691d (y_WW = 26909/65536 = 0.41060)

06          Element #6: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)
```

- Example 6: One light source - 1 x Extended Color Gamut

  Following output configurations define the relation of 1 application endpoint to all six output channels and the endpoint is attached to a multi-colored light source (R/G/B/A/T/V).

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
13          EndpointAndFunction: 0x13 - endpoint #1, red (R)
45          Flux (normalized): 0x45 = 69 (69/254 = 0.27)
86 b1       CIE 1931 x: 0xb186 = 45446 (x_R = 45446/65536 = 0.69345)
ef 4e       CIE 1931 y: 0x4eef = 20207 (y_R = 20207/65536 = 0.3083)

06          Element #2: six bytes
16          EndpointAndFunction: 0x16 - endpoint #1, amber (A)
c6          Flux (normalized): 0xc6 = 198 (198/254 = 0.78)
59 9a       CIE 1931 x: 0x9a59 = 39513 (x_A = 39513/65536 = 0.6029)
80 65       CIE 1931 y: 0x6580 = 25984 (y_A = 25984/65536 = 0.39648)

06          Element #3: six bytes
14          EndpointAndFunction: 0x14 - endpoint #1, green (G)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
39 1d       CIE 1931 x: 0x1d39 = 7481 (x_G = 7481/65536 = 0.1146)
82 d3       CIE 1931 y: 0xd382 = 54146 (y_G = 54146/65536 = 0.8262)

06          Element #4: six bytes
17          EndpointAndFunction: 0x17 - endpoint #1, turquoise (T)
b4          Flux (normalized): 0x64 = 180 (180/254 = 0.71)
9e 0b       CIE 1931 x: 0x0b9e = 2974 (x_T = 2974/65536 = 0.04538)
83 4b       CIE 1931 y: 0x4b83 = 19331 (y_T = 19331/65536 = 0.294968)

06          Element #5: six bytes
15          EndpointAndFunction: 0x15 - endpoint #1, blue (B)
4d          Flux (normalized): 0x4d = 77 (77/254 = 0.30)
c6 1f       CIE 1931 x: 0x1fc6 = 8134 (x_B = 8134/65536 = 0.1241)
cc 0e       CIE 1931 y: 0x0ecc = 3788 (y_B = 3788/65536 = 0.0578)

06          Element #6: six bytes
18          EndpointAndFunction: 0x18 - endpoint #1, violet (V)
6c          Flux (normalized): 0x6c = 108 (108/254 = 0.43)
2d 2c       CIE 1931 x: 0x2c2d = 11309 (x_V = 11309/65536 = 0.17256)
3a 01       CIE 1931 y: 0x013a= 314 (y_V = 3788/65536 =0.00479)
```

- Example 7: Three light sources - 1 x Color, 1 x White(Cool/Warm) and 1 x Dimmable (mono)

  Following output configurations define 3 application endpoints to the 6 output channels respectively:

  - the endpoint #1 attached to a color (RGB) light source,
  - the endpoint #5 attached to a color temperature light source or white(CW/WW) light source,
  - the endpoint #6 attached to a monochromatic (M) and dimmable light source.

    i.e., up to 3 different light sources are controllable depending on this output configurations.

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
13          EndpointAndFunction: 0x13 - endpoint #1, red (R)
47          Flux (normalized): 0x47 = 71 (71/254 = 0.25)
06 b1       CIE 1931 x: 0xb106 = 45318 (x_R = 45318/65536 = 0.69147)
ef 4e       CIE 1931 y: 0x4eef = 20207 (y_R = 20207/65536 = 0.3083)

06          Element #2: six bytes
14          EndpointAndFunction: 0x14 - endpoint #1, green (G)
a0          Flux (normalized): 0xa0 = 163 (163/254 = 0.63)
39 1d       CIE 1931 x: 0x1d39 = 7481 (x_G = 7481/65536 = 0.1146)
82 d3       CIE 1931 y: 0xd382 = 54146 (y_G = 54146/65536 = 0.8262)

06          Element #3: six bytes
15          EndpointAndFunction: 0x15 - endpoint #1, blue (B)
42          Flux (normalized): 0x42 = 66 (66/254 = 0.26)
c6 1f       CIE 1931 x: 0x1fc6 = 8134 (x_B = 8134/65536 = 0.1241)
cc 0e       CIE 1931 y: 0x0ecc = 3788 (y_B = 3788/65536 = 0.0578)

06          Element #4: six bytes
60          EndpointAndFunction: 0x60 - endpoint #6, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff

06          Element #5: six bytes
51          EndpointAndFunction: 0x51 - endpoint #5, first white (CW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
42 50       CIE 1931 x: 0x5042 (x_CW = 20546/65536 = 0.31352)
d9 52       CIE 1931 y: 0x52d9 (y_CW = 21209/65536 = 0.32363)

06          Element #6: six bytes
52          EndpointAndFunction: 0x52 - endpoint #5, second white (WW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
b9 75       CIE 1931 x: 0x75b9 (x_WW = 30137/65536 = 0.45986)
1d 69       CIE 1931 y: 0x691d (y_WW = 26909/65536 = 0.41060)
```

- Example 8: Six light sources - 6 x Dimmable (Mono)

  Following output configurations define 6 application endpoints to the 6 output channels. Each supported application endpoint of LD6 is attached to a monochromatic (M) and dimmable light source respectively. i.e. up to 6 light sources are controllable depending on the output configurations.

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
10          EndpointAndFunction: 0x10 - endpoint #1, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff

06          Element #2: six bytes
50          EndpointAndFunction: 0x50 - endpoint #5, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff

06          Element #3: six bytes
60          EndpointAndFunction: 0x60 - endpoint #6, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff

06          Element #4: six bytes
70          EndpointAndFunction: 0x70 - endpoint #7, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff

06          Element #5: six bytes
80          EndpointAndFunction: 0x10 - endpoint #8, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff

06          Element #6: six bytes
90          EndpointAndFunction: 0x10 - endpoint #9, mono (dimmable)
ff          Flux (normalized): 0xff
ff ff       CIE 1931 x: 0xffff
ff ff       CIE 1931 y: 0xffff
```

The LD6 allows free configuration of primary colors. The primaries must be properly arranged. Specifically, primaries on the convex hull of the gamut must be provided first, counter-clock-wise along the perimeter, starting with the red primary. White primaries, i.e. warm white, cool white or neutral white must be provided last. Errors in the ordering of primaries in the xy space will result in poor color reproduction and, in the worst case, flicker or unpleasant intensity fluctuations (beyond what is technically necessary due to intensity variations in primaries) while changing color. Therefore, we encourage integrators to use the set of configurations supported in the ubisys Smart Home app and provided here for reference.

The ubisys Smart Home app for iOS recognizes and generates a certain set of configurations, and lets users pick those easily. You may request an up-to-date list of optimized configurations by mailing support@ubisys.de and asking for these recommended configurations for typical use cases. It is recommended to avoid using other configurations.

#### 6.13.3.4.1. Default color temperature to CIE xy mappings

The ubisys Versalight Engine incorporated into the LD6 uses the following reference points for neutral, cool, warm and super warm white:

**cool**: x = 0.31352, y = 0.32363 (corresponds to 6500K)
**neutral**: x = 0.38044, y = 0.37675 (corresponds to 4000K)
**warm**: x = 0.45986, y = 0.41060 (corresponds to 2700K)

Other white tones can be specified using the xy parameters (note these need to be converted to Zigbee fixed-point numbers):

**super-warm**: x = 0.48614, y = 0.41467 (corresponds to 2400K)
**ultra-warm**: x = 0.54924, y = 0.40823 (corresponds to 1800K)

For other CCT to xy mappings you may use a look-up table, an online calculator or similar converter application.

Notice that the defaults offered in the ubisys iOS app have warm-white and cool-white set to the same intensities, whereas typical light strips have a higher intensity cool-white primary compared to the warm-white primary. For best results use the CIE 1931 information provided in the datasheet of the LED light-strip or other light source, including relative intensities.

#### 6.13.3.4.2. Extracting and using technical data from test reports

If you have a spectrum test report, sometimes also referred to as chromaticity test report or similar, available then you can easily extract the information required to fine-tune your LD6 to the LEDs used on the light strip (or LED spot etc.) that is attached to it.

A typical report would include CIE 1931 x and y coordinates and the luminous flux for each primary LED alongside a whole host of other information like dominant wavelengths, purity, corresponding CCT, CRI, render indices, as well as some electrical parameters like wattage, and maybe test conditions etc. All you need for fine-tuning the LD6 are the x and y coordinates, as well as the relative luminous flux (intensity) for primary LED.

Let us assume you have a LED strip with five primaries: red, green, blue, cool-white and warm-white. The data-sheet of the LED strip states following information:

**Red:** x = 0.6941, y = 0.3052, flux = 88.95lm
**Green:** x = 0.1639, y = 0.7391, flux = 256.8lm
**Blue:** x = 0.1336, y = 0.0636, flux = 54.57lm
**Warm-white:** x = 0.4662, y = 0.4159, flux = 619.9lm
**Cool-white:** x = 0.3247, y = 0.3507, flux = 672.7lm

The first step is to convert these numbers to fixed point representations (integers), as they are used within the Zigbee and Matter data model for lighting devices. For the x and y coordinates, this means multiplying by $2^{16}$ = 65536 = 0x10000. For the luminous flux, it means normalizing (i.e. dividing by the largest absolute flux, which is typically cool-white for a CCT or RGB+CCT strip, white for an RGBW strip, or green for an RGB strip) and multiplying by $2^8$ - 1 = 254 = 0xfe:

**Red:** x = 45489, y = 20002, Y/Y0 = 34
**Green:** x = 10741, y = 48438, Y/Y0 = 97
**Blue:** x = 8756, y = 4168, flux = Y/Y0 = 21

**Warm-white:** x = 30553, y = 27256, Y/Y0 = 234
**Cool-white:** x = 21280, y = 22983, Y/Y0 = 254

The next step is to make sure that you specify these values with primaries on the convex hull of the color gamut listed first, starting with red, and progressing counter-clock-wise; then continue with LEDs contained in the hull (whites), starting with cool-white.

**Red:** x = 45489, y = 20002, Y/Y0 = 34
**Green:** x = 10741, y = 48438, Y/Y0 = 97
**Blue:** x = 8756, y = 4168, flux = Y/Y0 = 21
**Cool-white:** x = 21280, y = 22983, Y/Y0 = 254
**Warm-white:** x = 30553, y = 27256, Y/Y0 = 234

As a final step, we take example 5 from above and emplace the values above at the right spots:

```
48          Data type: 0x48 (array)
41          Element type: 0x41 (raw data)
06 00       Element count: 0x0006 (6 entries)

06          Element #1: six bytes
13          EndpointAndFunction: 0x13 - endpoint #1, red (R)
47          Flux (normalized): 0x22 = 34
b1 b1       CIE 1931 x: 0xb1b1 = 45489
22 4e       CIE 1931 y: 0x4e22 = 20002

06          Element #2: six bytes
14          EndpointAndFunction: 0x14 - endpoint #1, green (G)
61          Flux (normalized): 0x61 = 97
f5 29       CIE 1931 x: 0x29f5 = 10741
36 bd       CIE 1931 y: 0xbd36 = 48438

06          Element #3: six bytes
15          EndpointAndFunction: 0x15 - endpoint #1, blue (B)
42          Flux (normalized): 0x42 = 21
34 22       CIE 1931 x: 0x2234 = 8756
48 10       CIE 1931 y: 0x1048 = 4168

06          Element #4: six bytes
11          EndpointAndFunction: 0x11 - endpoint #1, first white (CW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
20 53       CIE 1931 x: 21280 = 0x5320
c7 59       CIE 1931 y: 22983 = 0x59c7

06          Element #5: six bytes
12          EndpointAndFunction: 0x12 - endpoint #1, second white (WW)
fe          Flux (normalized): 0xfe = 254 (254/254 = 1.0)
45 77       CIE 1931 x: 30553 = 0x7745
78 6a       CIE 1931 y: 27256 = 0x6a78

06          Element #6: 0x06 (6 bytes)
00          EndpointAndFunction: 0x00 - not mapped to an endpoint/function
ff          Flux (normalized): 0xff (invalid value)
ff ff       CIE 1931 x: 0xffff (invalid value)
ff ff       CIE 1931 y: 0xffff (invalid value)
```

### 6.13.4. Works With All Hubs (Server)

Works With All Hubs cluster is a manufacturer-specific cluster from Amazon, identifiable through the cluster identifier 0xfc57 and manufacturer code 0x1217 and can exist at most only once on each device node [R9]. This cluster provides an interface for enabling various features of the Works With All Hubs cluster on LD6.

Manufacturer-specific attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0002 | Boolean, read-only, persistent | **DisableOTADowngrades**<br>Set the attribute to TRUE if the OTA client shall prohibit downgrades to an earlier version |
| 0x0003 | Boolean, read-only, persistent | **MGMTLeaveWithoutRejoinEnabled**<br>Set the attribute to FALSE, the node SHALL ignore Management Leave Without Rejoin commands that do not include APS encryption using the TC link key. |
| 0x0004 | Unsigned8, read-only | **NWKRetryCount**<br>Number of network-level retries that will be made, in a row, in case that a MAC data request fails.<br>“3” |
| 0x0005 | Unsigned8, read-only | **MACRetryCount**<br>Number of MAC Level Retries the device performs when sending a unicast command, not including APS retries.<br>“3” |
| 0x0006 | Boolean, read-only, persistent | **RouterCheckInEnabled**<br>set to TRUE if the router’s check-in algorithm has been enabled through the 'Enable Periodic Router Check-Ins' command.<br>“FALSE” |
| 0x0007 | Boolean, read-only, persistent | **TouchlinkInterpanEnabled**<br>set the attribute to FALSE if support for Touchlink Interpan messages has been disabled On the device.<br>“TRUE” |
| 0x0008 | Boolean, read-only, persistent | **WWAHParentClassificationEnabled**<br>Set the attribute to TRUE, then the device SHALL enable the WWAH Parent Classification Advertisements feature.<br>“FALSE” |
| 0x0009 | Boolean, read-only, persistent | **WWAHAppEventRetryEnabled**<br>Set the attribute to TRUE if the WWAH Application Event Retry Algorithm is enabled.<br>“FALSE” |
| 0x000A | Unsigned8, read-only, persistent | **WWAHAppEventRetryQueueSize**<br>This attribute is the queue size for re-delivery attempts in the WWAH Application Event Retry Algorithm.<br>“32” |
| 0x000B | Boolean, read-only, persistent | **WWAHRejoinEnabled**<br>Set this attribute TRUE if the WWAH Rejoin Algorithm (described below) is enabled.<br>“FALSE” |
| 0x000C | Unsigned8, read-only, persistent | **MACPollFailureWaitTime**<br>This attribute describes the time in seconds the device waits before retrying a data poll when a MAC level data poll fails for any reason.<br>“2” |
| 0x000D | Boolean, read-only, persistent | **ConfigurationModeEnabled**<br>Set the attribute TRUE, the device will accept all ZDO commands without APS encryption.<br>Set the attribute FALSE, all ZDO commands except those specified in [R9] should be disabled (unless sent encrypted using the Trust Center Link Key) |
| 0x000E | Unsigned8, read-only | **CurrentDebugReportID**<br>This attribute depicts the ID of the current debug report stored on the node. A value of 0x00 indicates that no debug report is available.<br>“0” |
| 0x000F | Boolean, read-only, persistent | **TCSecurityOnNwkKeyRotationEnabled**<br>Set the attribute TRUE, the node processes network key rotation commands, specifically APS Transport Key, which are sent via unicast and are encrypted by the Trust Center Link Key. |

| Attribute | Type | Description |
|---|---|---|
| 0x0010 | Boolean, read-only, persistent | **WWAHBadParentRecoveryEnabled**<br>Set the attribute TRUE, the end device enables the WWAH Bad Parent Recovery feature described as in [R9].<br>“FALSE” |
| 0x0011 | Unsigned8, read-only, persistent | **PendingNetworkUpdateChannel**<br>This attribute describes the channel number of the only channel the device accepts in a ZDO Mgmt Network Update command (and/or a Mgmt Network Enhanced Update command). |
| 0x0012 | Unsigned16, read-only, persistent | **PendingNetworkUpdatePANID**<br>This attribute contains the only short PAN ID the device SHALL accept in a NLME Network Update command. |
| 0x0013 | Unsigned16, read-only, persistent | **OTAMaxOfflineDuration**<br>This attribute describes the maximum time in seconds the device may be unavailable after completing its OTA download (while restarting, etc).<br>“60” |
| 0xFFFD | unsigned16 | **ClusterRevision**<br>The version of the cluster specification that this implementation adheres to |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **Enable APS Link Key Authorization**<br>This command enforces that all cluster commands for the cluster have APS level security. |
| 0x01 | **Disable APS Link Key Authorization**<br>This command removes the enforcement of APS level security on cluster commands. |
| 0x02 | **APS Link Key Authorization Query**<br>On receipt of this command, a device generates an appropriate APS Link Key Authorization Query Response command and unicast it to the requestor. |
| 0x03 | **Request New APS Link Key** On receipt of this command, a device requests a new Trust Center Link Key from the Trust Center. |
| 0x04 | **Enable WWAH App Event Retry Algorithm**<br>This command enables using the WWAH App Event Retry Algorithm to increase reliability of these events when the device temporarily loses contact with the hub. |
| 0x05 | **Disable WWAH App Event Retry Algorithm**<br>This command disenables using the WWAH App Event Retry Algorithm. |
| 0x06 | **Request Time**<br>This command enables a device to obtain the current values of Time Cluster attributes from the Time Cluster Server. |
| 0x07 | **Enable WWAH Rejoin Algorithm**<br>This command enables applying WWAH Rejoin Algorithm on end devices to recover and reconnect to the Zigbee network faster when connection is lost. |
| 0x08 | **Disable WWAH Rejoin Algorithm**<br>This command disables applying WWAH Rejoin Algorithm and devices are supposed to go back to the default behavior. |
| 0x09 | **Set IAS Zone Enrollment Method**<br>This command is used to dynamically change among the 3 enrollment methods of an IAS Zone Server specified in ZCL specification. |
| 0x0A | **Clear Binding Table** On receipt of this command, a device clears its binding table. |
| 0x0B | **Enable Periodic Router Check-Ins**<br>This command enables the Periodic Router Check-in WWAH functionality with the Check-In Interval specified. |
| 0x0C | **Disable Periodic Router Check-Ins**<br>This command disables the router periodic check-in behavior (if enabled). |
| 0x0D | **Set MAC Poll Failure Wait Time**<br>This command enables to update the MACPollFailureWaitTime attribute with the Wait Time value passed. |
| 0x0E | **Set Pending Network Update**<br>This command enables updating the PendingNetworkUpdateChannel and PendingNetworkUpdatePANID attributes with the Channel and PAN ID values passed, to protect the device from unauthorized Network Update. |
| 0x0F | **Require APS ACKs on Unicasts**<br>This command enforces that all unicast commands have APS ACKs enabled. |

| Command | Description |
|---|---|
| 0x10 | **Remove APS ACKs on Unicasts Requirement**<br>This command enables to roll back any changes caused by the Require APS ACKs on Unicasts command. The device can default back to its normal settings. |
| 0x11 | **APS ACK Requirement Query**<br>This command is meant to query the current Require APS ACKs on Unicasts settings. |
| 0x12 | **Debug Report Query**<br>This command enables requesting the debug report based on the specified Debug Report ID. |
| 0x13 | **Survey Beacons**<br>On receipt of this command, the device performs a scan for beacons advertising the device's network (using the default scan duration), generate a Survey Beacons Response command and unicast it to the requestor. |
| 0x14 | **Disable OTA Downgrades**<br>On receipt of this command, a device disallows downgrades of all components of its firmware. |
| 0x15 | **Disable MGMT Leave Without Rejoin**<br>On receipt of this command, a device responds with NOT_AUTHORIZED to Management Leave Without Rejoin commands that have not been APS encrypted with the trust center link key. |
| 0x16 | **Disable Touchlink Interpan Message Support**<br>On receipt of this command, a device ignores Touchlink interpan messages. |
| 0x17 | **Enable WWAH Parent Classification**<br>On receipt of this command, a device enables the WWAH Parent Classification Advertisements feature. |
| 0x18 | **Disable WWAH Parent Classification**<br>On receipt of this command, a device disables the WWAH Parent Classification Advertisements feature. |
| 0x19 | **Enable TC Security On Nwk Key Rotation**<br>On receipt of this command, a device processes only network key rotation commands, specifically APS Transport Key, which are sent via unicast and are encrypted with the Trust Center Link Key. |
| 0x1A | **Enable WWAH Bad Parent Recovery**<br>On receipt of this command, a device enables the WWAH Bad Parent Recovery feature. |
| 0x1B | **Disable WWAH Bad Parent Recovery**<br>On receipt of this command, a device disables the WWAH Bad Parent Recovery feature. |
| 0x1C | **Enable Configuration Mode**<br>This command enables to protect the device from unauthorized configuration changes during normal optional, i.e., to to put a device back into its default state where configuration changes are allowed. |
| 0x1D | **Disable Configuration Mode**<br>On receipt of this command, a device disables processing of all ZDO commands that have not been encrypted using the Trust Center Link key. |
| 0x1E | **Use Trust Center for Cluster**<br>This command forces the device to use ONLY the Trust Center as the cluster server for the set of clusters specified and persists this configuration across reboots. |
| 0x1F | **Trust Center for Cluster Server Query** On receipt of this command, a device generates an appropriate Trust Center for Cluster Server Query Response command and unicasts it to the requestor. |

### 6.13.5. Identify Cluster (Client)

The Identify cluster client takes on the role of the Finding & Binding initiator. Once activated, LD6 initiates the identification process by broadcasting identify query request command to detect target(s) operating in identify mode.

Cluster commands transmitted:

| Command | Description |
|---|---|
| 0x00 | **Identify**<br>Puts the remote device into identify mode, or terminates identify mode. |
| 0x01 | **Query Identify**<br>Determines whether the remote device is currently identifying, and returns the remaining time if so. |

### 6.13.6. Over-the-Air Upgrade Cluster (Client)

The image type for LD6 is 0x7B2C. You may request the latest firmware in Zigbee OTA image format

to upgrade devices in the field using your own OTA server and back-end. If you operate the LD6 with the Zigbee/Ethernet Gateway ubisys G1, the latest firmware for LD6 will always be available automatically and you do not need to request it explicitly. Notice that you may also upgrade the device over Bluetooth using a suitable Zigbee Virtual Device (ZVD) with associated OTA server. The ubisys Smart Home app for iOS version 2.2.0 or later supports this feature. You might have to download the corresponding OTA file from the ubisys OTA repository web-site and put it into the "ota" folder of the Smart Home app on your iPhone or iPad.

## 6.14. Application Endpoint #242 - Zigbee Green Power

This endpoint provides the Zigbee Green Power feature according to the 2015 edition of the specification, i.e. including support for Green Power Devices with IEEE EUI-64 and bidirectional commissioning. The Zigbee traffic between Proxies and Sinks utilizes standard Zigbee foundation paradigms and the Zigbee Cluster Library [R4]. You may use the standard ZCL frames to enumerate, read and write attributes, invoke commands, etc. The application endpoint exposes the following clusters:

| Cluster | Direction | Description |
|---|---|---|
| 0x0021 | Inbound (Server) | **Zigbee Green Power Sink**<br>Allows pairing the dimmer with Zigbee Green Power switches. The sink can directly receive Green Power frames but can also utilize proxies in the network. |
| 0x0021 | Outbound (Client) | **Zigbee Green Power Proxy**<br>Allows sinks on the network to configure this device as a “Proxy”, i.e. access point for Zigbee Green Power Devices into the Zigbee mesh network. |

### 6.14.1. Green Power Cluster (Server)

The server-side of the Green Power cluster provides the Zigbee Green Power Sink functionality, i.e. provides the ability to process Green Power frames. The device can receive those frames directly if they are in radio range; or it can process frames tunnelled over the Zigbee network via other Green Power Proxies. This implementation supports unidirectional and bidirectional33 GPDs. Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0000 | unsigned8, read-only | **gpsMaxSinkTableEntries**<br>The number of sink table entries supported by this device |
| 0x0001 | extended raw binary, read-only, persistent | **SinkTable**<br>Entries in the sink table create a pairing between the sink and a Green Power Device. For an operational pairing, one or more entries in the translation table are also required |
| 0x0002 | bitmap8 | **gpsCommunicationMode**<br>Determines the default communication mode between sink and proxies for new entries added by the sink to the sink table when paired with a GPD |
| 0x0003 | bitmap8 | **gpsCommissioningExitMode**<br>Specifies under which circumstances the sink should leave commissioning mode |
| 0x0004 | unsigned16 | **gpsCommissioningWindow**<br>Time, in seconds, before a commissioning session expires |
| 0x0005 | bitmap8, persistent | **gpsSecurityLevel**<br>Determines the minimum security level a GPD is required to support as well as whether or not the Trust Center needs to be involved into the commissioning process |
| 0x0006 | bitmap24, read-only | **gpsFunctionality**<br>Indicates Green Power features and building blocks supported by this device |
| 0x0007 | bitmap24, read-only | **gpsActiveFunctionality**<br>Allows to disable certain Green Power features on this device |

| Attribute | Type | Description |
|---|---|---|
| 0x0020 | bitmap8, persistent | **gpSharedSecurityKeyType**<br>Determines the security key type to use for devices with bidirectional commissioning capabilities, i.e. out-of-the-box individual key, shared GP key, etc. |
| 0x0021 | key128, persistent | **gpSharedSecurityKey**<br>The 128-bit AES-CCM* key that is being used to secure Green Power data frames |
| 0x0022 | key128, persistent | **gpLinkKey**<br>The 128-bit AES-CCM* key that is being used to deliver keying material to Green Power devices |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x00 | **GP Notification**<br>Tunnels GP frames from a Green Power Device to one or more sinks or groups of sinks |
| 0x04 | **GP Commissioning Notification**<br>Tunnels GP frames from a Green Power Device to a sink in commissioning mode |
| 0x05 | **GP Sink Commissioning Mode**<br>Makes the sink enter or leave commissioning mode for a particular endpoint or all endpoints |
| 0x09 | **GP Pairing Configuration**<br>Creates, updates or removes pairings by modifying the sink and translation table entries accordingly |
| 0x0A | **GP Sink Table Request**<br>Allows to query the sink table for a certain Green Power Device or read out the table in chunks |

Cluster commands transmitted:

| Command | Description |
|---|---|
| 0x01 | **GP Pairing**<br>Creates, updates or removes proxy table entries |
| 0x02 | **GP Proxy Commissioning Mode**<br>Makes the proxy enter commissioning mode for a particular sink, or leave commissioning mode |
| 0x06 | **GP Response**<br>Tunnels GP data frames from a sink to a bidirectional Green Power Device |
| 0x0A | **GP Sink Table Response**<br>Conveys a set of sink table entries to a management application |

#### 6.14.1.1. Green Power Device Interaction

##### 6.14.1.1.1. Commissioning a Green Power Switch

The LD6 can be put in commissioning mode either by using a suitable commissioning tool (for example the ubisys Network Manager) or by instigating Finding & Binding as a target. Once in commissioning mode, the device is ready to pair with Green Power switches.

##### 6.14.1.1.2. Supported Device Types

The LD6 can be paired with the following pre-defined Green Power Device types:

| Device Type | Description |
|---|---|
| 0x02 | **On/off Switch**<br>Default command set: on, off and toggle commands |
| 0x03 | **Level Control Switch**<br>Default command set: move, step and stop commands (with and without on/off) |
| 0x07 | **Generic Switch**<br>Default command set: press and release (up to eight buttons) |

In addition, any Green Power device that sends at least one of the supported commands listed in the

next section can also be paired with LD6.

#### 6.14.1.1.3. Supported Green Power Commands

The following Green Power commands are supported by default translations:

| Command | Description |
|---|---|
| 0x10 | **Recall scene #1** |
| 0x11 | **Recall scene #2** |
| 0x12 | **Recall scene #3** |
| 0x13 | **Recall scene #4** |
| 0x14 | **Recall scene #5** |
| 0x15 | **Recall scene #6** |
| 0x16 | **Recall scene #7** |
| 0x17 | **Recall scene #8** |
| 0x18 | **Store scene #1** |
| 0x19 | **Store scene #2** |
| 0x1A | **Store scene #3** |
| 0x1B | **Store scene #4** |
| 0x1C | **Store scene #5** |
| 0x1D | **Store scene #6** |
| 0x1E | **Store scene #7** |
| 0x1F | **Store scene #8** |
| 0x20 | **Turn Off** |
| 0x21 | **Turn On** |
| 0x22 | **Toggle** |
| 0x30 | **Move up** |
| 0x31 | **Move down** |
| 0x32 | **Step up** |
| 0x33 | **Step down** |
| 0x34 | **Stop** |
| 0x35 | **Move up with on/off** |
| 0x36 | **Move down with on/off** |
| 0x37 | **Step up with on/off** |
| 0x38 | **Step down with on/off** |
| 0x69 | **Generic Press** |
| 0x6A | **Generic Release** |

### 6.14.1.2. Generic Switch Commissioning

This product provides enhanced support for latest generation Green Power generic switches with the ability to distinguish short (< 1 second) and long (> 1 second) presses. In fact, a generic switch behaves like a physical switch directly connected to the high-voltage input lines, resulting in a consistent user experience. Each contact of a switch is commissioned separately, such that the desired button layout can be achieved on any generic switch. First commission button A, then button B, then C and so forth. Depending on the number of commissioned contacts, the behaviour is different. The following Green Power commands are supported by default translations:

| Number of Commissioned Contacts | Behavior |
|---|---|
| 1 | **Short Press**<br>Toggles the light on and off<br>**Long Press**<br>Dims the light up or down while pressed. The direction alternates on each subsequent press |
| 2 | **Short Press Button A**<br>Turns the light on<br>**Long Press Button A**<br>Dims the light up while pressed<br>**Short Press Button B**<br>Turns the light off<br>**Long Press Button B**<br>Dims the light down while pressed |
| 3 | **Press Button A**<br>Turns the light on<br>**Press Button B**<br>Turns the light off<br>**Press Button C**<br>Dims the light up or down while pressed. The direction alternates on each subsequent press |
| 4 | **Press Button A**<br>Turns the light on<br>**Press Button B**<br>Turns the light off<br>**Press Button C**<br>Dims the light up while pressed<br>**Press Button D**<br>Dims the light down while pressed |

### 6.14.2. Green Power Cluster (Client)

The client-side of the Green Power cluster provides the Zigbee Green Power Proxy functionality, i.e. makes the device act as an “access point” for Green Power Devices (GPDs). This implementation supports unidirectional and bidirectional34 GPDs.

Attributes supported:

| Attribute | Type | Description |
|---|---|---|
| 0x0010 | unsigned8, read-only | **gppMaxProxyTableEntries**<br>The number of proxy table entries supported by this device |
| 0x0011 | extended raw binary, read-only, persistent | **ProxyTable**<br>Entries in the proxy table create a link between Green Power Devices and Green Power Sinks |
| 0x0016 | bitmap24, read-only | **gppFunctionality**<br>Indicates Green Power features and building blocks supported by this device |
| 0x0017 | bitmap24, read-only | **gppActiveFunctionality**<br>Allows to disable certain Green Power features on this device |
| 0x0020 | bitmap8, persistent | **gpSharedSecurityKeyType**<br>Determines the security key type to use for devices with bidirectional commissioning capabilities, i.e. out-of-the-box individual key, shared GP key, etc. |
| 0x0021 | key128, persistent | **gpSharedSecurityKey**<br>The 128-bit AES-CCM* key that is being used to secure Green Power data frames |
| 0x0022 | key128, persistent | **gpLinkKey**<br>The 128-bit AES-CCM* key that is being used to deliver keying material to Green Power devices |

Cluster commands supported:

| Command | Description |
|---|---|
| 0x01 | **GP Pairing**<br>Creates, updates or removes proxy table entries |
| 0x02 | **GP Proxy Commissioning Mode**<br>Makes the proxy enter commissioning mode for a particular sink, or leave commissioning mode |
| 0x06 | **GP Response**<br>Tunnels GP data frames from a sink to a bidirectional Green Power Device |
| 0x0B | **GP Proxy Table Request**<br>Allows to query the proxy table for a certain Green Power Device or read out the table in chunks |

Cluster commands transmitted:

| Command | Description |
|---|---|
| 0x00 | **GP Notification**<br>Tunnels GP frames from a Green Power Device to one or more sinks or groups of sinks |
| 0x04 | **GP Commissioning Notification**<br>Tunnels GP frames from a Green Power Device to a sink in commissioning mode |
| 0x0B | **GP Proxy Table Response**<br>Conveys a set of proxy table entries to a sink or management application |

[2] Available in ZigBee stack version 1.60 and above. Legacy ZCP requirement – do not use in applications

[3] Available in Zigbee stack version 1.56 and above.

[4] Since the application version 1.4.0. Prior, the minimum on level was unsupported, i.e. the behavior equivalent to its value being set as the invalid value (0xFF).

[5] Available in LD6 firmware version 1.4.1 and above.

# 7. Physical Dimension

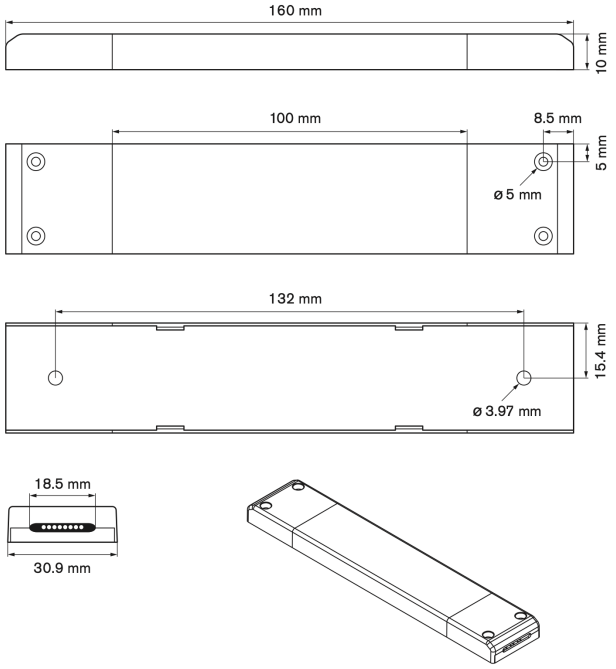


*Figure 1. Zigbee/Bluetooth LED controller LD6 – Dimensions*

# 8. Ordering Information

The following table lists the product variants available. Use the specified order code for your orders.

Orders for products sold under different brand names need to be placed with the respective vendor.

For any customized versions, please contact ubisys sales. OEM customers can always contact ubisys for more information.

| Vendor/Brand | Order Code | Description |
|---|---|---|
| ubisys | 1472 | Zigbee Driver LD6 For Luminaires |

## 9. General Terms & Conditions Of Business

When placing your order you agree to be bound by our General Terms & Conditions of Business, “Allgemeine Geschäftsbedingungen”, which are available for download here:
www.ubisys.de/en/smarthome/terms.html

# 10. Declaration Of Conformity

CE

We – ubisys technologies GmbH, Neumannstraße 10, 40235 Düsseldorf, Germany – declare under our sole responsibility that the ubisys Zigbee/Bluetooth LED Controller LD6 with order codes as detailed in Chapter 8 under the trade name “ubisys” to which this declaration relates are in conformity with the following directives and standards:

| Directive/Standard | Description/Scope |
|---|---|
| 2014/53/EU | Radio Equipment Directive (RED) |
| 2014/30/EU | Electromagnetic Compatibility Directive (EMC) |
| 2014/35/EU | Low Voltage Directive (LVD) |
| 2012/19/EU | Waste Electrical and Electronic Equipment Directive (WEEE) |
| 2011/65/EU | Restriction of Hazardous Substances Directive (RoHS) |
| EN 300 328 V2.2.2 | ERM; Wideband transmission systems; 2.4 GHz ISM band |
| EN 300 440 V2.2.1 | ERM; Radio equipment to be used in the 1 GHz to 40 GHz frequency range |
| EN 301 489-1 V2.1.1 | EMC |
| IEEE 802.15.4:2020 | IEEE Standard 802 – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) |
| Zigbee 3.0 | Zigbee 2023 with Zigbee Direct and Green Power |
| Bluetooth | Bluetooth 4.2 |

| | |
|---|---|
| Düsseldorf, Germany | September 16, 2021 |
| Place of issue | Date of issue |
| Dr.-Ing. Arasch Honarbacht | Managing Director, Head of Research & Development |
| Full name of Authorized Signatory | Title of Authorized Signatory |
|  |   |
| Signature | Seal |

# 11. Revision History

| Revision | Date | Remarks |
|---|---|---|
| 0.1 | 10/01/2021 | Initial draft |
| 0.2 | 01/09/2024 | Update according to the application firmware 1.4.0 |
| 1.0 | 01/18/2024 | Initial public version |
| 1.1 | 03/08/2024 | Editorials and additional information regarding Versalight engine defaults and behavior |
| 1.2 | 03/11/2024 | Minor corrections and editorial changes |
| 1.3 | 03/28/2024 | Minor corrections |
| 1.4 | 04/09/2024 | Add example for converting data sheet specifications for LED primaries to the Zigbee and Matter data model for lighting |
| 1.5 | 04/18/2024 | Add ubisys Versalight AdvancedOptions attribute documentation |

## 12. Contact

ubisys technologies GmbH
Neumannstr. 10
40235 Düsseldorf
Germany

T: +49. 211. 54 21 55 - 19
F: +49. 211. 54 21 55 - 99

www.ubisys.de
info@ubisys.de