AN1117: Migrating the Zigbee HA Profile to Zigbee 3.0

This document is provided to assist developers who are migrating their Zigbee HA projects to Zigbee 3.0. It discusses the specifics of changing configurations to Zigbee 3.0. It then discusses the configuration required for interoperability with legacy devices and networks.

KEY POINTS

- Migrating the EmberZNet Stack
- Updating ZCL Clusters
- New and removed plugins
- Generation and rebuilding
- Interoperability

1. Introduction

Zigbee 3.0 unites most of the different application profiles, such as HA (Home Automation) and ZLL (Zigbee Light Link), into one common application layer. Furthermore, it introduces greater test coverage for product certification, ensuring better interoperability for Zigbee devices in the field. The Zigbee 3.0 document suite contains both revised and completely new material for the Zigbee application specification. For more information, see *UG103.02: Zigbee Fundamentals*. As of this writing, the Zigbee Smart Energy profile is not included in Zigbee 3.0.

This document first describes the changes to the AppBuilder configurations to make an HA profile project into a Zigbee 3.0 project. It then specifically discusses the configuration required for interoperability with legacy devices.

2. Migrating the Zigbee HA Profile to Zigbee 3.0

This section reviews the differences between legacy and Zigbee 3.0 profiles. The differences are presented by AppBuilder tab. In order to migrate a project you will need to edit configurations as described below.

2.1 EmberZNet Stack

The approach to security changed significantly from HA to Zigbee 3.0. Set **Security Type** to "Zigbee 3.0 Security" as shown in the following figure.

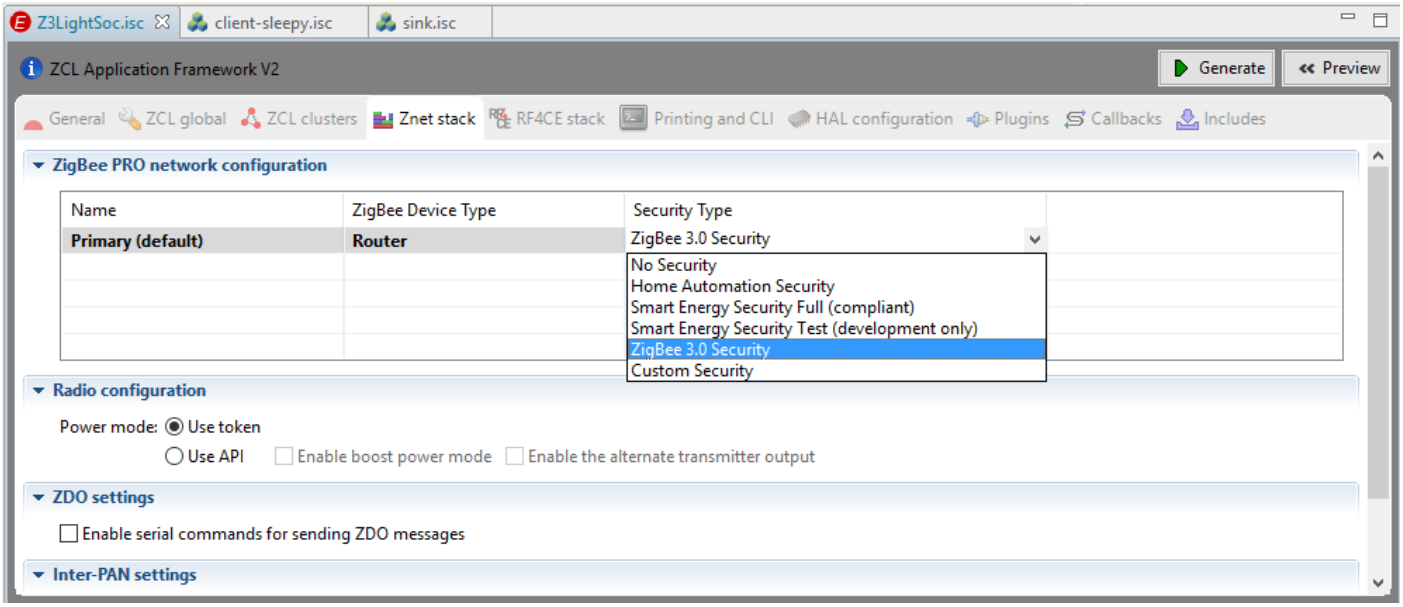


Figure 2.1. Security Type Setting

2.2 ZCL Clusters

Although the cluster names and application profile ID remain the same when migrating from HA devices and Zigbee 3.0 devices, a new set of ZCL device types from the Lighting & Occupancy (“LO”) working group of Zigbee is now provided to achieve selections of client and server clusters that conform to the Zigbee 3.0 Base Device Behavior Specification. These are selectable in the ZCL Device Type picklist of AppBuilder from the “LO devices” section and supersede the legacy selections available in the “HA devices” section.

An HA router application normally has at least one endpoint called Primary, as shown in the following figure.

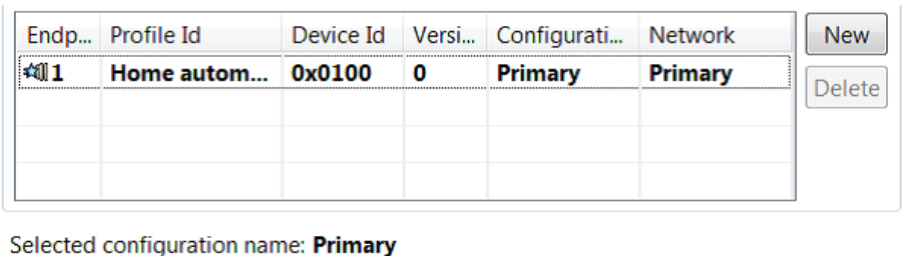


Figure 2.2. The ZCL Configuration of an HA Profile

In contrast, a Zigbee 3.0 router application always has more than one endpoint. The Green Power Combo Basic device is required by the Zigbee 3.0 specification for any application configured as a router or coordinator node type. The easiest way to implement this is to create a separate endpoint as a ZCL device type 'GP Combo Basic," as shown in the following figure from the "Z3 Light" example application.

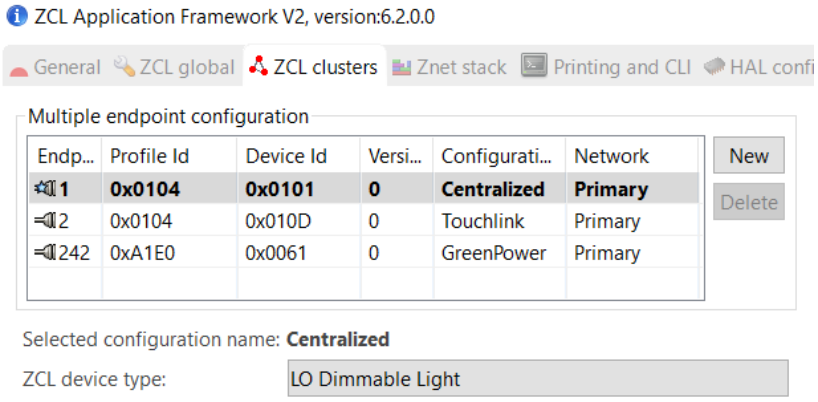


Figure 2.3. The ZCL Configuration of a Zigbee 3.0 Profile

Having a dedicated endpoint with Zigbee Light Link (ZLL) Commissioning cluster support (such as the "Touchlink" endpoint in the figure above) allows for the "touchlink" style of commissioning between a touchlink initiator and a touchlink target, either of which could be present in a legacy ZLL or Zigbee 3.0 network. Touchlink commissioning support is an optional feature for Zigbee 3.0 and is therefore not required or necessarily supported by all networks or devices, but including it in your ZCL configuration gives the most flexibility for commissioning within the Zigbee 3.0 network.

Additionally, some lighting devices may wish to take advantage of ZLL-specific extensions to clusters like On/Off, Scenes, Identify, and Color Control. Providing separate endpoints for ZCL controller or switch device types with and without these extensions offers better compatibility among a mix of devices with different capabilities in the field. For example, the Z3 Light SoC example application provides the "LO Dimmable Light" on endpoint #1 (shown in the figure above) and the "LO Extended Color Light" device type on endpoint #2.

2.3 Plugins

This section first lists the plugins that you would typically de-select in the HA device profile, and then lists and explains the new Zigbee 3.0 plugins.

Plugins to be de-selected in the transition from the HA profile:

- Identify Feedback
- EZ-Mode Commissioning
- Network Find

The following plugins are superseded by new plugins that are Zigbee 3.0-compatible.

- Green Power Library - Provides functionality for Green Power infrastructure devices (mandatory only for router devices).
- Install Code Library - Provides an initial link key based on an installation code manufacturing token in the device. The key is hashed according to the Zigbee specification and can be used by applications running Smart Energy 1.x profile or Home Automation 1.2 profile. However, note that for Zigbee 3.0 the install code length must be 16 bytes (plus LSB CRC16), as opposed to the variable lengths used by Smart Energy 1.x. See *AN1089: Using Installation Codes with Zigbee Devices* for instructions on programming these codes.
- Green Power Client - Implements the client-side functionality of the Green Power cluster.
- Green Power Common - Provides common functionalities between client and server sides of the Green Power plugins.
- Green Power Server - Implements the server-side functionality of the Green Power cluster.
- Scan Dispatch - Supports multiple consumers of the stack 802.15.4 scan results.
- Find and Bind Target - Provides the functionality for a target device to start identifying on user-defined endpoints. The target device can then act as a target for a finding and binding initiator. See Zigbee 3.0 Base Device Behavior (BDB) specification for more details about Find and Bind.
- Find and Bind Initiator - Provides the functionality for an initiator device to find devices that are performing the find and bind process for a target, and then optionally attempting to bind with those devices.
- Zigbee Light Link Library (optional) - The ZLL library provides support for an application running the ZLL profile by generating and processing ZLL touchlink messages.
- Network Creator (required for a coordinator, optional for a router) - Performs the necessary steps to create a network according to the Base Device Behavior specification. The plugin performs an active scan followed by an energy scan across a primary channel set in order to decide which channel(s) are valid candidates for network formation. If the plugin fails to form a network on any primary channels, it moves to a secondary channel mask. Note that this plugin requires the Network Creator Security plugin, as this plugin will set up the stack to use Zigbee 3.0 security before every attempt at network formation. Routers that want to have the capability to form a network in distributed trust center mode may also include this plugin.
- Network Creator Security (required when using the Network Creator plugin) - Performs the necessary security initialization to form a Zigbee 3.0-compliant network.
- Network Steering - Performs the necessary steps to join the network of any Zigbee Profile. It tries first to perform a join using an install code using the Primary Channel Mask, and then the secondary channel mask. If that does not work, it then tries to use the default well-known link key (ZigBeeAlliance09) to join on the primary channel mask, and then the secondary channel mask.
- Update TC Link Key (required for any joining device) - Provides the functionality to update the trust center link key of a device on a Zigbee R21 network.
- ZLL Commissioning (required only if supporting touchlinking) - Silicon Labs implementation of the commissioning mechanism used by the Zigbee Light Link profile. If initiating the touchlink process, select the Initiator option in this plugin.
- ZLL Utility Server Cluster (required only if supporting touchlinking) - Silicon Labs implementation of ZLL Utility functionality, necessary for handling ZLL scan requests on a ZLL Commissioning server endpoint.

2.4 Generation and Rebuilding

Once you have completed configuring your project with Zigbee 3.0-compliant settings, you should be able to generate and compile your project. Other steps may be required for full project migration, for example manually rebuilding your project callbacks file or modifying any custom code.

3. Interoperability

Zigbee 3.0 has been designed to allow for interoperability between Zigbee 3.0 devices and legacy HA and ZLL devices. With proper configuration, ZLL and HA devices can join Zigbee 3.0 networks and, similarly, Zigbee 3.0 devices have the functionality to join legacy networks operating with either ZLL or HA networking.

3.1 Zigbee 3.0 Devices on ZLL or HA Networks

Zigbee 3.0 devices contain the profiles necessary to join with HA and ZLL networks.

Zigbee 3.0 network principles are the same as with HA networks. To join an HA network, program the Zigbee 3.0 device with the Zigbee default link key, and it will join in the same manner as any HA device.

To join a ZLL network, configure the Zigbee 3.0 device to support touchlinking, which requires enabling the ZLL Commissioning client and server clusters for at least one endpoint, as described in section 2.2 ZCL Clusters, and then enabling both the ZLL Commissioning plugin with the Link Initiator option enabled, and the Zigbee Light Link Library plugin. For devices that will be the targets of touchlink commissioning rather than the initiators, the Link Initiator option in the ZLL Commissioning plugin should remain unchecked, and the ZLL Utility Server plugin should be enabled.

Note that we only include the certification test keys (not to be used in production) in our software release. To obtain ZLL production keys, please contact the Zigbee Alliance.

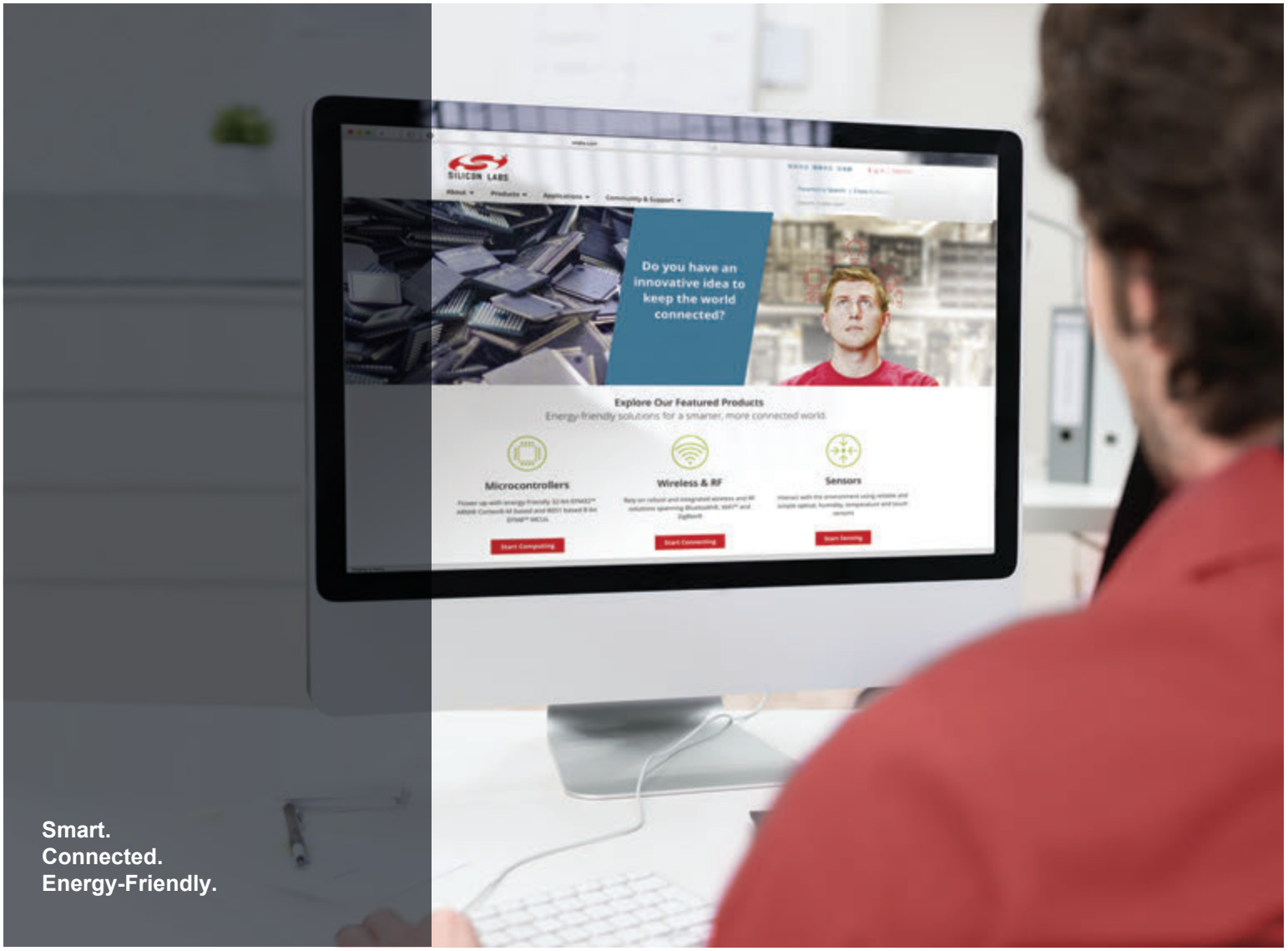
3.2 Legacy Devices on Zigbee 3.0 Networks

ZLL touchlinking is provided as an optional mode of network security within Zigbee 3.0. As long ZLL touchlinking is allowed as a policy within the Zigbee 3.0 network, ZLL devices can join it.

To allow HA devices to join a Zigbee 3.0 network, implement both the HA profile for networking and the Zigbee default link key on the network. The HA device can then join with a standard HA-style join.

Using a default link key makes the Zigbee 3.0 network vulnerable to a number of security threats. These security issues must be explored and carefully considered before deciding on the path to take within a Zigbee 3.0 application. Some options include the following:

- Using the default link key as the primary link key within the network. This solves any HA interoperability issues, but introduces the vulnerabilities that Zigbee 3.0 security was designed to prevent.
- Using the default link key as a transient link key (a link key with a timeout, after which it will no longer work). This allows HA devices to join a network in a small window, minimizing security vulnerabilities but can create problems. If a legacy device associates using the default link key and then detaches from the network, it will not be able to rejoin using an unsecured rejoin. Also, legacy devices may not be programmed to update their link key.



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information
Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS®

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com