Kinetis ZigBee Stack Demo Applications
User's Guide

1. About this document

This document provides an introduction to the deployment and operation of the wireless network applications created using the ZigBee network protocol stack running on Kinetis KW2xD or Kinetis with MCR20A development platforms.

This document is an operation manual for the sample applications included with the Kinetis ZigBee® Stack software builds.

1.1. Audience

This document intend to help firmware and system developers creating ZigBee enabled products. The document provides a high-level description of the ZigBee application scenarios which can be exercised using Kinetis development boards.

Contents

1. About this document ......................................................... 1
   1.1. Audience ..................................................................... 1
2. Introduction ........................................................................ 2
3. Kinetis BeeStack hardware platforms .................................. 2
   3.1. Kinetis wireless MCU (KW2x) .................................... 2
   3.2. MCR20A discrete IEEE 802.15.4 transceiver ............... 5
4. Deploying BeeStack and application software .................... 10
   4.1. Deploying Kinetis SDK .............................................. 10
   4.2. Deploying Kinetis BeeStack software ......................... 12
5. Using Project Cloner to create BeeStack applications ......... 16
6. Running ZigBee example applications ................................. 18
   6.1. Example applications user interfaces ......................... 18
   6.2. Application support library (ASL) keys ....................... 18
   6.3. ASL configuration mode .......................................... 18
   6.4. HA on/off light ...................................................... 20
   6.5. HA on/off switch ..................................................... 20
   6.6. HA dimmable light ................................................. 21
   6.7. HA dimmer switch .................................................. 21
   6.8. HA temperature sensor ......................................... 22
   6.9. Generic application ................................................ 22
   6.10. HA combined interface ........................................... 22
   6.11. ZLL color controller ............................................. 23
   6.12. ZLL color light ..................................................... 23
   6.13. ZigBee light link network ....................................... 24
7. EZ-Mode commissioning .................................................... 25
   7.1. Network steering ..................................................... 25
   7.2. Finding and binding ................................................ 26
   7.3. Software implementation ........................................ 26
8. Revision history ............................................................... 28

© 2016 Freescale Semiconductor, Inc. All rights reserved.
2. Introduction

The Kinetis BeeStack includes a set of example applications for the ZigBee Pro Feature Set using the Home Automation or ZigBee Light Link supplemented by a generic application. These applications require you to install IAR Embedded WorkBench version 7.4 or later.

NOTE
It is assumed that you are familiar with the purpose and major features of the ZigBee wireless networks. The document provides a cursory overview of the project Cloner and the IDE to load the applications on the development boards. For more information, see the appropriate reference manuals and ZigBee/802.15.4 specifications.

3. Kinetis BeeStack hardware platforms

The Kinetis BeeStack version 5.0.x supports the following platforms which have radio frequency transceiver capabilities compatible with the IEEE 802.15.4 standard used by ZigBee technology.

- Kinetis KW2xD wireless microcontroller family integrates an ARM® Cortex®-M4 core with an IEEE 802.15.4 transceiver.
- MCR20A discrete IEEE 802.15.4 transceiver which can be paired over a SPI bus with other MCU families running the BeeStack software.

3.1. Kinetis wireless MCU (KW2x)

The following section lists the KW2xD characteristics and development boards.

3.1.1. KW2xD characteristics

The Kinetis KW2x wireless microcontrollers expand the Kinetis microcontrollers family based on the ARM Cortex-M4 CPU core. The Kinetis KW2x integrates a class-leading 2.4 GHz RF transceiver, ARM Cortex-M4 core and a robust feature set for a reliable, secure, and low-power IEEE® 802.15.4 wireless solution. Dual PAN support allows the system to simultaneously participate in two IEEE 802.15.4 or ZigBee networks, eliminating the need for multiple radios.

The KW2x family is available as multiple device numbers. These device types differ in flash, RAM, and peripheral characteristics as in Table 1.

<table>
<thead>
<tr>
<th>Device</th>
<th>Flash</th>
<th>RAM</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKW21D256VHA5</td>
<td>256 KB</td>
<td>32 KB</td>
<td>FlexMemory: 64KB FlexNVM/4KB FlexRAM, No USB</td>
</tr>
<tr>
<td>MKW21D512VHA5</td>
<td>512 KB</td>
<td>64 KB</td>
<td>No USB</td>
</tr>
<tr>
<td>MKW22D512VHA5</td>
<td>512 KB</td>
<td>64 KB</td>
<td>USB</td>
</tr>
<tr>
<td>MKW24D512VHA5</td>
<td>512 KB</td>
<td>64 KB</td>
<td>USB</td>
</tr>
</tbody>
</table>
The KW2x family has the following characteristics:

- ARM Cortex-M4 core with DSP up to 50 MHz core provides a broad range of application support.
- Up to 512 KB of flash and up to 64 KB of RAM. Large memory footprint provides enough memory to run complicated protocol stacks and user applications on a single IC.
- Up to 64 KB of FlexMemory flash configuration (optional). FlexMemory provides user segment byte write/erase EEPROM and wear leveling handled in hardware.
- Secure flash. Protects code and data from unauthorized access or modification.
- Tamper detect. Protects critical IP by detecting tamper events. If a tamper occurs, secure RAM is asynchronously erased and an interrupt can be generated so that the application firmware can take additional actions, including a system reset.
- Cryptography acceleration unit. Coprocessor supports a set of specialized operations to improve throughput of encryption/decryption operations as well as message digest functions, including DES, 3DES, AES, MDA and SHA algorithms.
- High-performance IEEE 802.15.4-2011 transceiver. Supports a number of 802.15.4 protocol stacks, including ZigBee technology, 6LowPAN, WirelessHART and ISA 100.11a.
- Packet processor. Radio handles many 802.15.4 functions in hardware to reduce the software stack size and reduce power consumption by off loading functions from the CPU.
- Single ended and diversity. Single 50 ohm antenna uses single balun to reduce component count and cost. Freescale fast antenna diversity allows the hardware to automatically select between two antennas for improved reliability in high-interference environments.
- Dual PAN support. System can simultaneously participate in two ZigBee networks, eliminating the need for multiple radios.
- Output of +10 to +8 dBm and -102 dBm sensitivity. 110 dB link budget improves range and lowers cost by reducing the need for external power amplifiers.
- TX 17 mA @ 0 dBm (CPU sleep) and RX 15 mA. Significantly reduces power consumption and extends battery life. Low Power Preamble Search (LPPS) receiver mode.
- 128-bit random number generator. Meets the FIPS 140 security requirements for cryptographic modules.
- UART, SPI, I2C, optional USB 2.0 FS/LS H/D/OTG, 16-bit ADC.
- 1.8–3.6 V operating range Provides wide voltage range to maximize usable voltage for battery operation.
- Small footprint. Smaller size and low component count reduces cost.
- Operational temperature range of −40 °C to +105 °C. Ideal for applications that need extended temperature ranges.
3.1.2. KW2x development boards

The following development boards are based on the Freescale Kinetis KW2x based IEEE 802.15.4 and ZigBee wireless microcontrollers.

- FRDM-KW24D512
- USB-KW24D512

3.1.2.1. FRDM-KW24D512

The FRDM-KW24D512 Freescale Freedom platform is a Freedom format board based on the KW2xD family device MKW24D512.

For more information and board documentation on FRDM-KW24D512 see the page at: http://www.freescale.com/frdm-kw24d512.
3.1.2.2. USB-KW24D512

The USB-KW24D512 board is a USB dongle format board based on the KW2xD family device MKW24D512. It uses the on-chip USB port on the MKW24D512 to communicate directly to USB host.

For more information and board documentation on USB-KW24D512 see the page at: http://www.freescale.com/usb-kw24d512.

![USB-KW24D512 board](image)

Figure 2. USB-KW24D512 board

3.2. MCR20A discrete IEEE 802.15.4 transceiver

This section lists the MCR20A characteristics and development boards.

3.2.1. MCR20A characteristics

The MCR20A is a low power, high-performance 2.4 GHz, IEEE 802.15.4-compliant transceiver with connectivity over the SPI bus to a broad range of other devices, including the rest of the Kinetis microcontroller families.

The MCR20A has the following characteristics.

- Transceiver performance
  - 2.4 GHz (2360 to 2480 MHz) covers ISM band
  - Fractional-N PLL supports 1 MHz and 5 MHz channels
  - 250 kbps data rate
  - OQPSK modulation
  - Programmable output power
  - Sensitivity of -102 dBm RX
- Transceiver features
Kinetis BeeStack hardware platforms

- High-performance IEEE 802.15.4 transceiver. Supports a number of 802.15.4 protocol stacks, including Thread, ZigBee technology, WirelessHART, and ISA 100.11a.
- Packet processor. Radio handles many 802.15.4 functions in hardware to reduce the software stack size and reduce power consumption by offloading functions from the CPU.
- Support for Dual PAN mode.
- Onboard trim of reference crystal.
- 128-byte RAM data buffer.
- Low-power operating modes with single SPI command device wake-up.
- On-chip voltage regulators.
- Clear Channel Assessment, Energy Detect, Link Quality Indicator.

- Radio peripherals:
  - 24-bit event timer with interrupts
  - Eight (8) software programmable GPIOs
  - Control port for antenna diversity mode or external PA and LNA

- Microcontroller Interface:
  - Programmable frequency clock output (CLK_OUT)
  - SPI command channel and interface
  - Interrupt request output

- Operating Characteristics
  - 1.8 V to 3.6 V operating voltage
  - 17mA TX, 19mA RX, < 1mA idle/doze, < 1uA hibernate typical current
  - operational temperature range: –40°C to +105°C

- Physical Characteristics: RoHS-compliant, 5 mm x 5 mm, 32-pin MLGA

For more information on MCR20A, visit http://freescale.com/MCR20A.

3.2.2. MCR20A development boards

This section lists the MCR20A development boards.

3.2.2.1. FRDM-CR20A

The FRDM-CR20A is a reference design for the MCR20A transceiver, in a shield form factor, compatible with existing Freescale Freedom development platforms.

For more information and board documentation on FRDM-CR20A see the page at: http://www.freescale.com/FRDM-CR20A
3.2.2.2. FRDM-K64F with FRDM-CR20A

The FRDM-K64F development Freedom platform is supported as a motherboard software development platform for BeeStack applications in conjunction with the FRDM-CR20A.

For more information and board documentation on FRDM-K64F Freedom platforms see the page at: http://www.freescale.com/FRDM-K64F.
Figure 4. FRDM-K64F with FRDM-CR20A
3.2.2.3. FRDM-KL46Z with FRDM-MCR20A

For more information and board documentation on FRDM-KL46Z platform see the page at: http://www.freescale.com/FRDM-KL46Z

Figure 5. FRDM-KL46Z with FRDM-MCR20A
4. Deploying BeeStack and application software

The Kinetis BeeStack software package includes components necessary to begin ZigBee wireless mesh network application development on Kinetis platforms.

These components include:

- ZigBee-based mesh network software libraries
- IEEE 802.15.4 Media Access Control (MAC) software libraries and Physical Layer (PHY) drivers
- Home Automation 1.2 profile
- ZigBee Light Link profile
- ZTC over FSCI
- OTA upgrade
- Example applications

The Kinetis BeeStack Stack software package does not contain the Kinetis peripheral drivers, platform startup code, RTOS kernel software or other generic Kinetis platform software but reuses these components as provided by the Kinetis Software Development Kit (Kinetis SDK or KSDK) pre-requisite package which must be deployed before installing the BeeStack software or building demo projects.

4.1. Deploying Kinetis SDK

To deploy the Kinetis SDK 1.2.0:

1. In a web browser, access the KSDK web page at: http://freescale.com/KSDK.
2. Click the Downloads link in the tab bar.
3. Click the Download button.
4. Login with or create a freescale.com user ID if required to continue through the login page.
5. On the KSDK downloads page, Select Product Search, enter 1.2.0 and click Search.
6. In the search results, click the KSDK v1.2.0 Mainline link matching the Installer: Kinetis SDK 1.2.0 Mainline – Windows® operating system entry.

![Figure 7. KSDK download search](image)

7. Review the License Agreement and click I Agree if the terms are accepted.

8. Click the File Name entry: Kinetis SDK 1.2.0 Mainline – Windows.exe.

![Figure 8. KSDK 1.2.0 Mainline download](image)

**NOTE**

You can apply the Kinetis BeeStack source code 5.0.x example projects only on Windows OS using the IAR EWARM IDE. Other operating system versions of the KSDK are not supported.

9. Wait for the download to start and complete.

10. Launch the Kinetis SDK 1.2.0 Mainline.exe installer.

11. Select Yes in the dialog box confirming KSDK installation.
Deploying BeeStack and application software

12. Leave the default installation path or alternatively enter a customized one and click **OK**.

13. The installation progress appears. Wait for its completion.

14. Confirm the installation is successful. You have now deployed the Kinetis SDK and are ready to install the Kinetis BeeStack software.

### 4.2. Deploying Kinetis BeeStack software

To deploy the BeeStack software:

1. In a web browser, access the ZigBee network technology home page [here](#).
2. In the **Software & Tools** area select **Kinetis BeeStack** then proceed to the Download tab to select the Kinetis BeeStack installer package.
3. Login with or create a freescale.com user ID if required to continue through the login page.
4. Review any License Agreements shown and click **I Agree** if the terms are accepted.
5. Wait for the installer download to start and complete.
6. Launch the **Kinetis BeeStack 5.0.x.exe** installer package and click **Next** to continue.
Figure 13. Kinetis BeeStack setup

7. Review the License Agreement and click **I Agree** if the terms are accepted.

8. Confirm the location of the Kinetis SDK installation set in the previous section.

**NOTE**

The Kinetis BeeStack installer makes a set of limited changes and add supplemental files to the Kinetis SDK library and RTOS folders installation. This facilitates the integration with BeeStack for the supported Kinetis BeeStack platforms. Back-ups are created in the same location to the files which are modified.

9. After acknowledging the KSDK path, click **Next**.
10. Choose the Path of the Kinetis BeeStack software installation by leaving in the default or alternatively entering a customized path then click **Next**.
11. Choose the name of the start menu folder for the Kinetis BeeStack installation by leaving in the default or alternatively entering a customized entry then click Install.

12. The installation progress appears. Wait for its completion.

13. Click **Finish**. By default, Release notes will be displayed on installer completion for users to review.
5. Using Project Cloner to create BeeStack applications

Cloning an example application project allows creating a duplicate of the application source code of a sample configuration based on one of the EWARM workspace and project sets for a board and RTOS configuration found in the examples root folder. This allows users to load and develop application firmware without modifying the core installation source files or default project configurations. Project cloning can also be useful when developing multiple applications based on the same initial example template.

To create a clone of an example project:

1. Launch Project Cloner executable which is located at: `<Kinetis_BeeStack_Installation>`. For example: `C:\Freescale\Kinetis_BeeStack_5.0.0\tools\project_cloner\project_cloner.exe`.

2. The Project Cloner dialog is displayed.

Figure 17. Setup complete

14. Confirm the installation is successful. You have now deployed the Kinetis BeeStack package to chosen installation location.
Using Project Cloner to create BeeStack applications

Figure 18. Project cloner

3. Select a <example-apps>_<mcu> in the Clone example app drop down.

4. Select a <development-board>_<rtos>_<toolchain>_<device-type>_<security-configuration> in the Clone configuration drop down.

5. Change the Cloned app name or leave as is – the app name entered in the text box will become the name of the parent root folder and the *.eww/*.ewp/*.ewd files in the output file.

6. Select a Destination root path to the output path – a folder named based on the Cloned app name text field above will be create at the output path if it does not exist.

7. Select using the radio box to:
   a) Clone application files only (minimum number of application files) – the other files will be linked from the stack installation folder using an environment variable.
   b) Clone the full codebase, including all the other folders and libraries.

8. Press Clone Project to create a clone. A message log will be displayed in the log box as files are being copied. Desintation files are overwritten if they already exist.

9. Navigate to the <Destination root path>\<Cloned app name> folder to access the application files. The <*.ewp>/<*.eww>/<*.ewd> files can be found and launched in IAR EWARM from the root of this folder.
6. Running ZigBee example applications

This section lists the example applications user interfaces and the application support library keys.

6.1. Example applications user interfaces

Almost all of the example applications are using four LEDs and four push-buttons (SW1-SW4). On boards with an RGB LED the functionality of three LEDs may be grouped into the RGB LED. The example applications run on any of the boards.

6.2. Application support library (ASL) keys

Most of the applications use a common routines library referred to as the Application Support Library (ASL). These routines include all of the common user interface elements of the applications, including the concept of two keyboard modes (configuration mode and run-mode), and a common set of keys that map to specific ZigBee functions.

All the example applications use the same keys to cause any activity that they all can do, such as join or leave a network. This section describes the key presses that are common to all the example applications.

The included key driver can read a long (approximately one second) keypress and a short (less than one second) keypress from each of the four keys. The example applications take advantage of this to support up to eight user-initiated events in each mode. The keypress interpretation is entirely in the application code; the applications can be rewritten to do anything.

The applications have the following two modes:

- Configuration
- Run

The applications always start in configuration mode, where key presses cause the network formation and setup. When that is done, another key press takes the application to run mode, where it can do whatever the application is designed to do, such as turn a light on or off remotely, report data, etc.

6.3. ASL configuration mode

This section describes the ASL configuration mode with EZ-Mode commissioning and for other applications.

6.3.1. ASL configuration mode with EZ-Mode commissioning (home automation 1.2)

Short button press functionality:

- SW1 – Start EZ-Mode Commissioning: Network Steering (form the network if ZC, join if ZR or ZED) followed by Finding and Binding process
• SW2 – Start EZ-Mode Commissioning: only Network Steering (form the network if ZC, join if ZR or ZED)
• SW3 – Start EZ-Mode Commissioning: only Finding and Binding process
• SW4 – Change radio channel: each press steps to the next higher channel and briefly displays a channel indicator on all four LEDs: 0000 is channel 11, 1111 is channel 26

Long button press functionality:
• Long SW1 – Go to run mode
• Long SW2 – EZ-Mode Commissioning: Reset to Factory Defaults
• Long SW3 – EZ-Mode Commissioning: Reset to Factory Defaults
• Long SW4 – Not used

6.3.2. ASL configuration mode for other applications

The behavior below is for other applications than Home Automation 1.2.

Short button press functionality:
• SW1 – Form/join network (form if ZC, join if ZR or ZED) with previous configuration
• SW2 – Toggle permit join (ZC/ZR) or toggle low power mode (ZED)
• SW3 – End device bind/match
• SW4 – Change radio channel: each press steps to the next higher channel and briefly displays a channel indicator on all four LEDs: 0000 is channel 11, 1111 is channel 26

Long button press functionality:
• Long SW1 – Go to run mode
• Long SW2 – Leave the network
• Long SW3 – Remove all bindings
• Long SW4 – Form/join network (form if ZC, join if ZR or ZED) with new configuration

6.3.3. Run mode for other applications

The behavior below is for other applications than Home Automation 1.2.

Short button press functionality
• SW1 – This does the main action of the application: toggle a light, etc.
• SW2 – Not used
• SW3 – Toggle identify mode
• SW4 – Recall scene

Long button press functionality
Running ZigBee example applications

- Long SW1 – Go to configuration mode
- Long SW2 – Not used
- Long SW3 – Send “Add group if in Identify mode” command
- Long SW4 – Send “Store scene if in Identify mode” command

6.4. HA on off light

This application acts as a lamp that can be turned on or off, but it cannot be dimmed.

6.4.1. Keyboard

This application allows the user to press a switch that toggles an LED on the same board. This feature is intended to allow the user to verify that the board is in the run mode and operating properly regardless of the state of any other node in the network.

- SW1 – Toggle local light (if on, turn off; if off, turn on)

6.4.2. Display

- The local light in this application is LED2.
- LED2 represents the state of the light.

6.5. HA on off switch

A common use of a home automation lighting control is turning on and off a remote entity. It can run in any ZigBee node type, including a ZED with low power operation enabled (a “sleepy” end device). This application includes the ability to request that the receiving node acknowledge the command.

6.5.1. Keyboard

In run mode, key presses do the following:

- SW1 - Toggle remote light
- SW2 - Based on the SwitchActions attribute, moves the switch from state 1 to state 2
- Long SW2 - Based on the SwitchActions attribute, moves the switch from state 2 to state 1

6.5.2. Display

The LEDs in run mode do the following:
• LED2 – Toggles with each toggle command (this can get out of synchronization with the remote light because the remote light’s state can also be changed by other methods).
• LED2 on – Sent On Light Command with acknowledge.
• LED2 off – Sent Off Light Command with acknowledge.

6.6. **HA dimmable light**

The dimmable light application uses LED2-LED4 or the RGB LED to represent a light with four states.

6.6.1. **Keyboard**

• SW1 – Decrease local light’s level
• SW2 – Increase local light’s level
• Long SW2 – Toggle local light on/off

6.6.2. **Display**

Once in run mode, the dimmable light control configuration makes the following LED assignments to indicate low through high lighting intensity.

• All LEDs off – Light off
• LED2 on – Low light intensity
• LED2 and LED3 on – Medium light intensity
• LED2, LED3 and LED4 on – Maximum light intensity

The LCD display on the MC1322x Network Node shows the light intensity as a percentage and a histogram.

6.7. **HA dimmer switch**

This controls the dimmable light.

6.7.1. **Keyboard**

• SW1 – Decrease level of light
• SW2 – Increase level of light
• Long SW2 – Toggle light on/off
Running ZigBee example applications

6.7.2. Display

- LED2 toggle – When sending an increase or decrease message to a dimmable light

6.8. HA temperature sensor

The temperature sensor application uses a simulated temperature value.

The Temperature Sensor can send its temperature value using indirect mode, that is, using the local binding table. It typically sends this data when another node has requested that the Temperature Sensor enter reporting mode using the ZigBee Cluster Library (ZCL) report attribute command or when the sensor is indicated to report using a Long SW2 press. Use SW3 End Device Bind or the Configuration Tool application to bind the Temperature Sensor to a Thermostat, so that the Thermostat can receive temperature reports from the Sensor.

6.8.1. Keyboard

- SW1 – Decrease simulated temperature when hardware sensor is not used.
- SW2 – Increase simulated temperature when hardware sensor is not used.
- Long SW2 – Turn the Report Temperature On, and begins sending the temperature using indirect transmission (reports are sent to the devices to which the sensor is bound).

6.8.2. Display

- LED1 flashing – sensor is sending OTA temperature reports to bound devices.
- LED2 flashing – sensed temperature is below -5°C.
- LED2, LED3, LED4 Off – sensed temperature is between -5°C and 10°C.
- LED2 On, LED3, LED4 Off – sensed temperature is between 10°C and 20°C.
- LED2 and LED3 On, LED4 Off – sensed temperature is between 20°C and 30°C.
- LED2, LED3 and LED4 On – sensed temperature is between 30°C and 40°C.
- LED2, LED3 On and LED4 Flashing – sensed temperature is above 40°C.

6.9. Generic application

The generic application is a device with no unique features. It only uses the common user interface.

6.10. HA combined interface

The Combined Interface application acts as a gateway between a personal computer and the ZigBee network by using the ZTC library for UART communication. The run mode of the application does not include any functionality except the “Go to configuration mode” which is
enabled by a long press of SW1 of the evaluation board. The behavior of the switches and LEDs in Configuration mode is the same as all other Home Automation applications.

6.11. ZLL color controller

The application emulates a device that performs color lights control. It is able to start a network using Touchlink commissioning procedure, EZ mode commissioning procedure or a standard commissioning procedure and to control the light intensity, light color, to switch on/off.

6.11.1. Keyboard

In Configuration mode, the keys functionality is specified below:
- SW1 - Touchlink commissioning (Identify + Start Commissioning)
- SW2 - Clasical Commissioning (EZ mode commissioning or ZDO start)
- SW3 - EZ mode network steering followed by finding and binding
- Long SW1 - Move to application mode
- Long SW2 - Touchlink Commissioning - Factory New Request
- Long SW3 - Touchlink Commissioning - Factory Fresh

In RUN mode, the keys functionality is specified below:
- SW1 - Touchlink commissioning - Identify + Start Commissioning)
- SW2 - Toggle the light
- SW3 - Decrease the level (Step Down)
- SW4 - Increase the level (Step Up)
- Long SW1 - Switch to Configuration Mode
- Long SW3 - Change the color (x, y random values)

6.12. ZLL color light

This application acts as a lamp that can be turned on or off, can be dimmed and also supports the color control cluster.

6.12.1. Keyboard

In configuration mode, the keys functionality is specified below:
- SW1 - Touchlink commissioning - Identify + Start Commissioning)
- SW2 - Clasical Commissioning (EZ mode commissioning or ZDO start)
- SW3 - EZ mode network steering followed by finding and binding
6.13. ZigBee light link network

To form a ZLL network, user must use at least one initiator device such as ZLL Color Controller and one or more target devices such as ZLL Color Light devices.

The following sections list the steps to form a ZLL network.

6.13.1. Enable the receiver on the target device

Based on the Freescale implementation, if the Color Light application is a Router the receiver is always ON and no user interaction is required. An End Device enables its receiver either through the short press of SW1, or via the ZTC command ZTC_ZllTouchlinkCommissioning_STARTrequest with the following parameters:

- Device Type: End Device (0x20)
- Device Role: Target (0x00)
- Functionality: 0x01 (Start commissioning)

6.13.2. Initiate TouchLink commissioning procedure on the initiator device

Keep the initiator device (ZLL Color Controller) closed to one of the target devices such as ZLL Color Light – ZR device and press short SW1 or use the ZTC command ZTC_ZllTouchlinkCommissioning_STARTrequest with the following parameters:

- Device Type: according with the device type: Router (0x80) or End Device (0x20)
- Device Role: Initiator (0x01)
- Functionality: 0x05 (Identify and Start commissioning)

If the commissioning procedure ends with success the LED will become solid, otherwise repeat previous steps.
6.13.3. Controlling the ZLL color light from the color controller

Press Long SW1 on Color Controller device to enter in the application mode. Control the Color Light devices by using keyboard (see section 1.1.1) or use the ZTC commands for color control, level control, on/off, identify clusters.

Toggling the light will not be visible at the lowest light level (default). To increase the light level press SW4 on the ZLL Color Controller.

To change the color of the LED to a random value long press SW3.

For setting the color of the LED to a specific value in the CIE xyY Color Space the ZTC command ColorControl_MoveToColor may be used with the desired values for ColorX and ColorY, as shown below:

- DestAddressMode: 0x02
- DestAddress: 0x0000000000000002
- DestEndPoint: 0x08
- ClusterId: 0x0300
- SrcEndPoint: 0x08
- TxOptions: 0x00
- Radius: 0x05
- TransactionSeqNo: 0x01
- ColorX: 0x5555
- ColorY: 0x8888
- TransitionTime: 0x0001

7. EZ-Mode commissioning

EZ-Mode Commissioning is a network steering and commissioning method focused on the network and device installer, on the ability to easily configure and commission ZigBee systems during development and testing. It became a mandatory part of the Home Automation 1.2 Profile and requires two steps which can also be combined to be overloaded on a single installer action such as a physical device button press:

- Network Steering: forming/joining to the proper network;
- Finding and Binding: discovering and creating source bindings between a specific lists of clusters;

A method to restore the device to the factory settings is also available: EZ-Mode Commissioning Factory Reset.

7.1. Network steering

Based on the device role, when the network steering is invoked the device may attempt to join to an existing network (on the Router, EndDevice, Combo Device). If no networks are found, it may form a network (on the Coordinator, Combo Device).
EZ-Mode commissioning

When this procedure is called on a ZigBee Router or Coordinator and the node is already joined to a network, the node shall broadcast a minimum PermitJoin time (EZModeTime) and also shall set its own PermitJoin time to this value.

7.2. Finding and binding

To discover and match services (clients finding the server, server finding clients, controllers/gateways finds sensors, etc.) at this step, the EZ-mode procedure introduces the roles of Initiator and Target.

The Initiator is a device that has at least one cluster that is an initiator of operational transactions.

For a Target, the finding and binding step requires to put the device into the identify state to a minimum of EZModeTime (default value is 180 seconds).

When EZ-mode Finding and Binding is invoked on an Initiator, the device broadcasts the Identify Query command, and when the Identify Query Response is received, it should attempt to discover clusters that match a particular application that the source device needs to bind to. Service discovery involves the use of one of the available discovery mechanism, such as the ZDP command for Simple Descriptor Request. Once the Initiator completes binding, it may set the Identify Time of its Targets to zero.

7.3. Software implementation

Starting with HCS08 BeeStack Codebase 3.1.1 the EZ-Mode Commissioning procedure is included in the Home Automation 1.2 Projects for MC1323x and offers a default configuration for the following applications.

<table>
<thead>
<tr>
<th>Device</th>
<th>Initiator</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>On/Off Switch</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Dimmer Switch</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>On/Off Light</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Dimmable Light</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Generic App</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Temperature Sensor</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Combined Interface</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

The files EzCommissioning.c and EzCommissioning.h comprise the EZ-Mode commissioning functionality. All configurations are made at compile-time through proprieties located in ZclOptions.h:

- gASL_EnableEZCommissioning_d: enables or disables the EZ-Mode Commissioning procedures
- gASL_EnableEZCommissioning_Initiator_d: enables the EZ-Mode Commissioning Initiator Role, otherwise the device is a Target
- gASL_EzCommissioning_EnableGroupBindCapability_d: enables the Group Support for EZ-Mode Commissioning; if it is enabled the device should also have the Group Client Cluster
- gASL_ZclCmdEzModeInvokeReq_d: enables support to invoke EZ-Mode over the air
- gASL_ZclCmdUpdateCommissioningStateReq_d: enables support for the update commission state command (Identify cluster) used to update the Commission state attribute over the air.

The list of particular clusters used in the binding process is gEZModeClusterList, and it is located in each Home Automation 1.2 Project in BeeApp.c file. The list can be modified according with the Home Automation specification and manufacturer requirements.

Below it is presented the flowchart for a simple network with two devices (one Coordinator - Target and An End Device – Initiator).

![Flowchart for a simple network with two devices](image-url)

**Figure 20. Flowchart for a simple network with two devices**
8. Revision history

Table 3. Revision history

<table>
<thead>
<tr>
<th>Revision number</th>
<th>Date</th>
<th>Substantive changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>01/2016</td>
<td>Updated for software version 5.0.0</td>
</tr>
</tbody>
</table>