Application Note

ZigBee 3.0 Controller and Switch

This Application Note provides example applications for a controller and a switch in a ZigBee 3.0 network that employs the NXP KW41Z wireless microcontrollers. An example application can be employed as:

- **A demonstration using the supplied pre-built binaries that can be run on FRDM-KW41Z boards**
- **A starting point for custom application development using the supplied C source files and associated project files**

The controller and switch described in this Application Note are based on ZigBee device types from the ZigBee Lighting & Occupancy (ZLO) Device Specification.

The Application Note also includes an example of a typical ZigBee Green Power (GP) Energy Harvesting switch.

The ZigBee 3.0 nodes presented in this Application Note can be used in conjunction with other ZigBee 3.0 nodes, their Application Notes being available on the NXP web site.

1 Introduction

A ZigBee 3.0 wireless network comprises of a number of ZigBee software devices that are implemented on hardware platforms to form nodes. This Application Note is concerned with implementing the device types for a controller and a switch on the Kinetis KW41Z platforms.

This Application Note provides example implementations of a controller and a switch that use one of the following device types from the ZigBee Lighting & Occupancy (ZLO) Device Specification:

- Colour Scene Controller
- Dimmer Switch

The above device types are detailed in the *ZigBee 3.0 Devices User Guide* and the clusters used by the devices are detailed in the *ZigBee Cluster Library User Guide*.

Note: If you are not familiar with ZigBee 3.0, you are advised to refer to the *ZigBee 3.0 Stack User Guide* for a general introduction.

2 Development Environment

2.1 Software

In order to make use of this Application Note, you need to install the following software:

- IAR Embedded Workbench, version 8.22.2 or above.
- MCUXpresso, version 10.2.0
- KW41Z_ZigBee_3.0_Software_v.6.0.7

2.2 Hardware

Hardware boards are available from NXP to support the development of ZigBee 3.0 applications. The following board is recommended for running these applications:

- NXP FRDM-KW41Z Evaluation Board

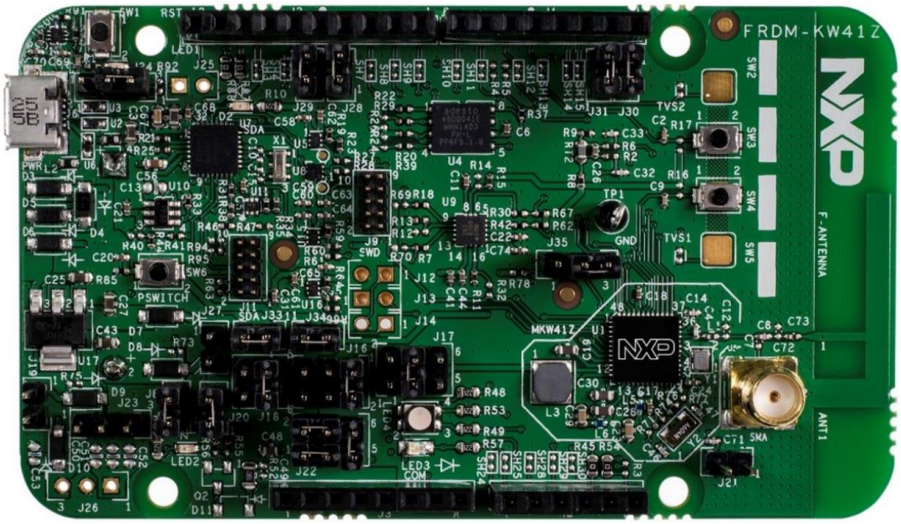


Figure 1. NXP FRDM-KW41Z Evaluation Board

3 Application Note Overview

The example applications provided in this Application Note are listed in the table below with the switch/controller device types that they support.

Application Name	Device Type
App_ColorSceneController	Colour Scene Controller
App_DimmerSwitch	Dimmer Switch

Table 1: Example Applications

For each application, source files and pre-built binary files are provided in the software package. The pre-built binaries can be downloaded and run on the FRDM-KW41Z board.

- To load the pre-built binaries into the evaluation board and run the demonstration application, refer to Section 5.
- To start developing you own applications based on the supplied source files, refer to Section 8.

3.1 Compatibility

The software provided along with this Application Note has been tested with the following evaluation kits and SDK versions.

Product Type	Revision/Version	Supported Chips
FRDM-KW41Z Evaluation Board	Rev. A/A2/A3	KW41Z
IAR Embedded Workbench	v.8.22.2	KW41Z
MCUXpresso	v.10.2.0	KW41Z
KW41Z_Zigbee_3.0_Software_v6.0.7	v.6.0.7	KW41Z

Table 2: Compatibility Information

4 Supported Device Types

As indicated in Section 3, the supported ZLO device types in this Application Note are:

- Colour Scene Controller
- Dimmer Switch

These switch/controller device types must be paired for usability with lighting device types. Example applications for the paired device types are provided in the *ZigBee 3.0 Light Bulbs* Application Note. Two device types can be paired if they support the same cluster (Colour Control, Level Control or On/Off cluster) in order that the cluster client on the switch/controller device type can access/control attributes of the cluster server on the lighting device type.

The table below lists the switch/controller device types (as well as the ZigBee Base Device) and, for each device type, indicates which types of cluster attributes can potentially be controlled (on the lighting device).

Device Type	Attribute Types					
	OnOff	**Level**	**X & Y Colour**	**Hue & Saturation**	**Colour Temperature**	**Colour Loop**
Dimmer Switch	Yes	Yes	No	No	No	No
Colour Scene Controller	Yes	Yes	Yes	Yes	Yes	Yes
Control Bridge	Yes	Yes	Yes	Yes	Yes	Yes
Base Device Coordinator	Yes	No	No	No	No	No
Base Device End Device	Yes	No	No	No	No	No

Table 3: Switch/Controller Device Types and Controllable Attributes

The table below lists the lighting device types (as well as the ZigBee Base Device) and, for each device type, indicates which types of cluster attributes could potentially be written/read.

Device Type	Attribute Types					
	OnOff	**Level**	**X & Y Colour**	**Hue & Saturation**	**Colour Temperature**	**Colour Loop**
Dimmable Light	Yes	Yes	No	No	No	No
Extended Colour Light	Yes	Yes	Yes	Yes	Yes	Yes
Colour Temperature Light	Yes	Yes	No	No	Yes	Yes
Base Device Router	Yes	No	No	No	No	No

Table 4: Lighting Device Types and Accessible Attributes

5 Running the Demonstration Application

This section describes how to use the supplied pre-built binaries to run the example applications on the components of the FRDM-KW41Z board.

5.1 Loading the Applications

The table below lists the application binary files supplied with this Application Note and indicates the hardware with which the binaries can be used. These files are located in the **../tools/wireless/binaries** directories for the relevant applications.

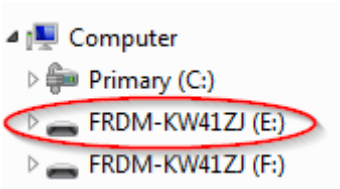
Application	Binary File	Expansion Board
App_ColorSceneController	color_scene_controller_frdmkw41z.bin	FRDM-KW41Z
App_DimmerSwitch	dimmer_switch_frdmkw41z.bin	FRDM-KW41Z

Table 5: Application Binaries and Hardware Components

A binary file can be loaded into the Flash memory of a KW41Z device using the FRDM-KW41Z on-board bootloader (hosted by the OpenSDA chip) or using Test Tool 12 software, available via the NXP web site.

To load an application binary file into the KW41Z Flash memory using the existing on-board bootloader, follow the instructions below:

1. Connect an USB port of your PC to the micro-USB port on the FRDM-KW41Z board. At this point, you may be prompted to install the necessary drivers.
2. After driver installation completes, a new drive letter will be assigned to the FRDM-KW41Z bootloader. This can be checked using Windows Explorer.



3. Using drag-n-drop, copy the binary file to this new drive.
4. Once the download has successfully completed, disconnect and reconnect the USB cable.

Operating instructions for the different applications are provided in the sections below.

5.2 Using App_DimmerSwitch

This section describes how to commission and operate the App_DimmerSwitch application in a ZigBee 3.0 network. To use this application, you must have programmed the application binary into the FRDM-KW41Z board.

Note: To use this application which is based on the Dimmer Switch device type, you will also need to implement the paired Dimmable Light device type as described in the *ZigBee 3.0 Light Bulbs* Application Note.

5.2.1 Dimmer Switch Functionality

The ZLO Dimmer Switch device resides on a node which acts as a low power End Device in the network. As an End Device, it can only join an existing network and cannot form a new network. This switch device can be used to perform the commissioning of the light devices and control them, using the controls indicated in the diagrams below.

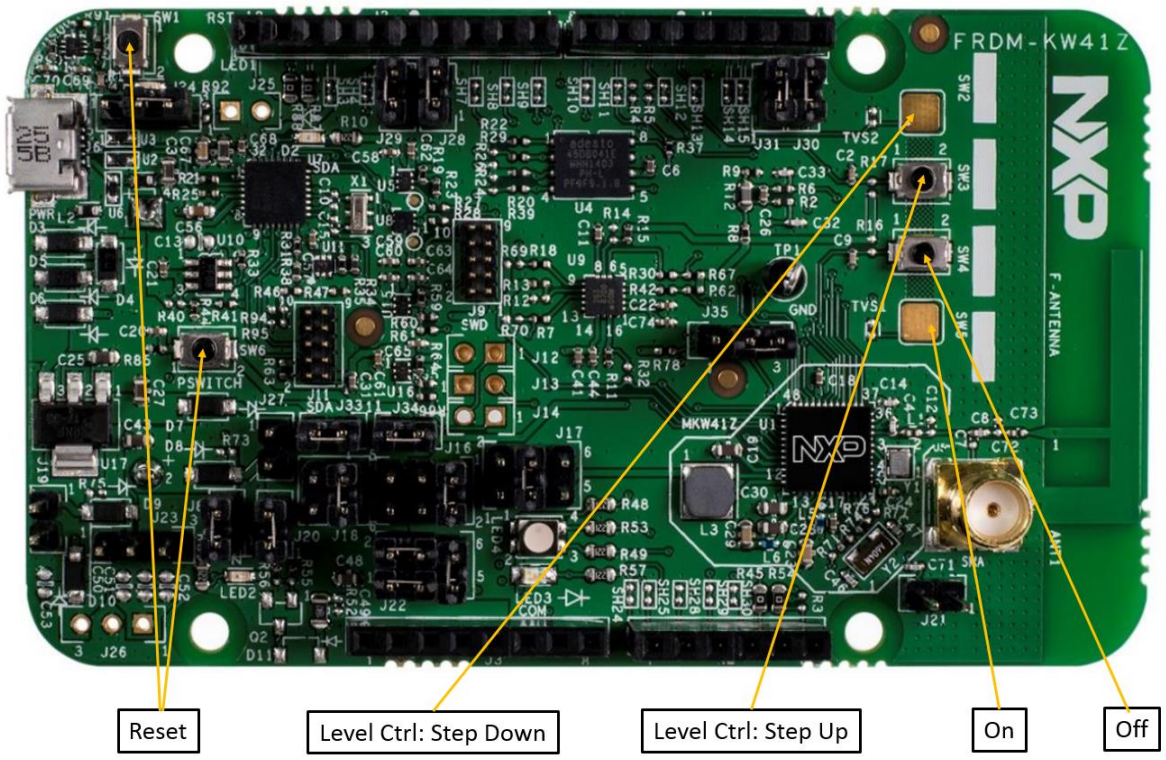


Figure 2: Using the FRDM-KW41Z buttons for commissioning the network

Note 1: The Reset switch from above will reset the board but will not erase the persistent data. To reset the board to factory-new state, refer to *Performing a Factory Reset* chapter.

Note 2: On the FRDM-KW41Z board only the buttons SW3 and SW4 will wake the device from low power for 3 seconds. If the device is in low power, the events associated with the respective buttons will not be processed on the first button press..

5.2.2 Joining a Network

1. Once there is a Coordinator running, trigger ‘Network Steering for a device on a network’ to open the network for joiners.

   This will cause a Management Permit Join Request to be broadcasted to the network to open the Permit Join status for a period of 180 seconds.

2. The switch device is an End Device and, as such, must join an existing ZigBee network.

Start Network Steering on the switch by pressing any button from **SW2 to SW5** on the FRDM-KW41Z board. As soon as a button-press is detected, the device will search for a suitable network to join.

The outcome of Network Steering is now indicated as follows:

- If a suitable open network is found ('Permit Join' is true), the switch node will join.
- If a suitable network is not found, the switch node goes into deep sleep mode, which is indicated by all LEDs being switched off.

5.2.3 Binding Devices

The Dimmer Switch supports the On/Off and Level Control clusters as a client and implements the Finding and Binding process as a Initiator. To trigger Finding and Binding as a Initiator, do the following:

1. Press the **SW2** button on the FRDM-KW41Z board of all the target devices.

2.Start Finding and Binding on the Initiator device - Long press the **SW4** button on Dimmer Switch.

This will cause the target device to self-identify for 180 seconds, while the Initiator will try to find the identifying devices, query their capabilities and create bindings on those with matching operational clusters. As part of this process, the target device may receive an Add Group Command and/or a Binding Request Command from Dimmer Switch.

5.2.4 Operating the Device

The operational functionality of this device in this demonstration is provided by the On/Off and Level Control clusters. Since the device supports the On/Off and Level Control clusters client, the target device (e.g. Dimmable Light device) will respond to commands sent by bound devices. The Dimmer Switch can send OnOff Toggle (toggling a light) and Level Control: Step Down or Up (dim or brighten a light) commands to the FRDM-KW41Z board.

5.2.5 Performing a Factory Reset

The Dimmer Switch in this Application Note can be reset to the factory-new state as follows:

- Hold down the **SW5** button for more than 8 seconds

The factory default state is indicated by the blinking LED3.

The reset maintains the value of the outgoing network frame counter across the reset. This parameter is never cleared out by a reset.

5.3 Using the Dimmer Switch to control the Light devices

The following setup uses the Dimmer Switch to exercise basic commands over the Light devices. A ZigBee End Device (ZED) will be commissioned into the network to exercise the On/Off and Level Control clusters. In this case, the ZED must be bound with the Dimmer Switch device using the 'Finding and Binding' procedure.

This setup requires at least three FRDM-KW41Z boards, one of the boards will be programmed with the Dimmer Switch application, one with the Dimmable Light application and one with a Control Bridge application to form the network.

First, the Dimmer Switch application needs to be downloaded to one of the FRDM-KW41Z boards. The Control Bridge binary file is located at the following relative path:

..\tools\wireless\binaries

On how to program the board, read the “Loading the Applications” chapter.

Then, the second FRDM-KW41Z board needs to be programmed. The recommended application is the Dimmable Light application. The third board will be programmed with the Control Bridge application.

After all boards are programmed, connect the Control Bridge to a PC/laptop and open the ZGWUI (ZigBee Gateway User Interface) application.

The ZGWUI application is **available as a TestTool toolbar option:**

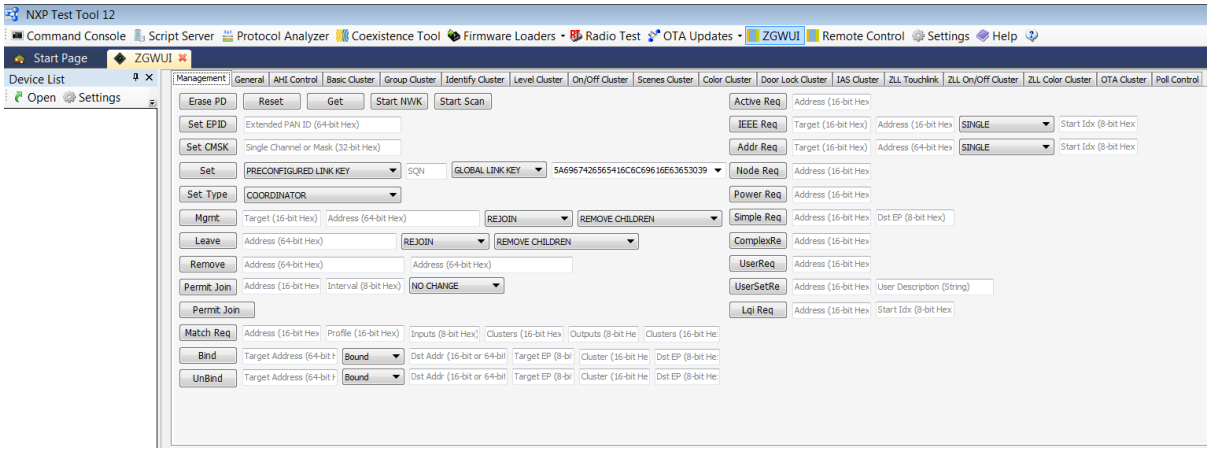
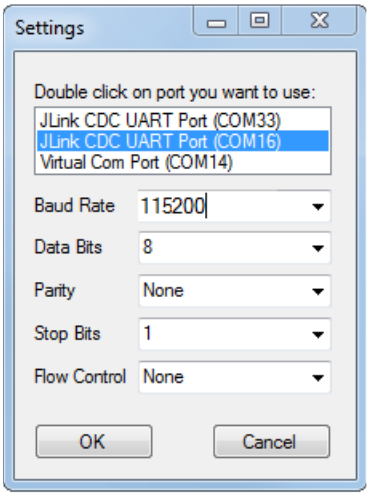


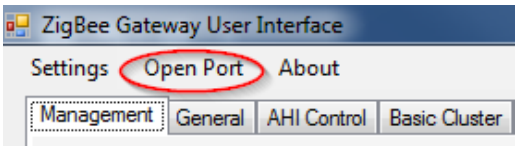
Figure 3 - ZigBee Gateway User Interface

To setup a new network, follow the below steps:

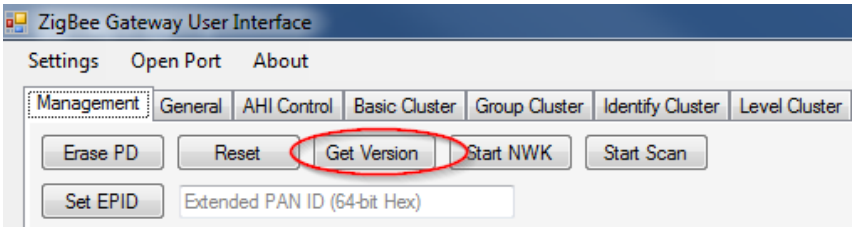
- Click on “Settings” menu item
- The “Settings” window will open
- Select the Virtual COM port assigned to FRDM-KW41Z Control Bridge



- Leave the other parameters unchanged and click OK to close the window
- Click on “Open Port” menu item

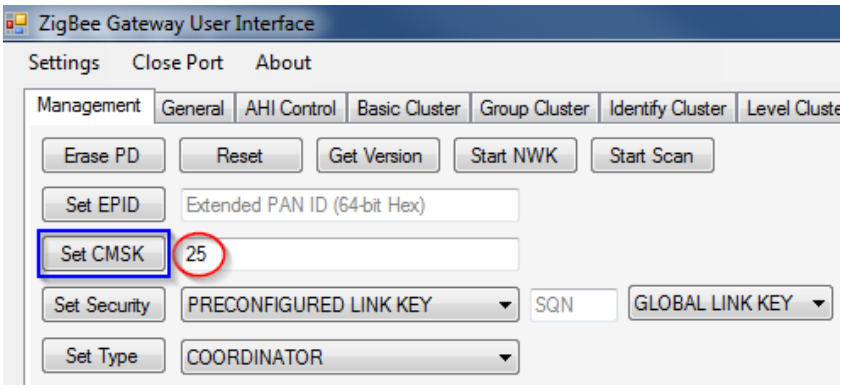


- Check the connectivity with the Control Bridge by pressing the “Get Version” button

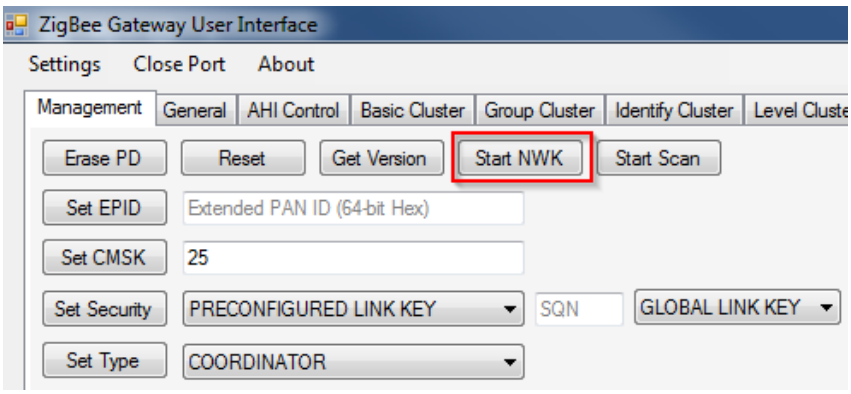


- In the Received Message View, if the communication is functional, the following message shall appear:

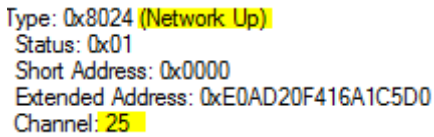
- Set the desired channel (or channel mask) for the new ZigBee 3.0 network to be created by editing the below box followed by the press of “Set CMSK”



- Start the network on selected channel by pressing the “Start NWK” button



- In the Received Message View, we got the confirmation that the network is up and running on the selected channel

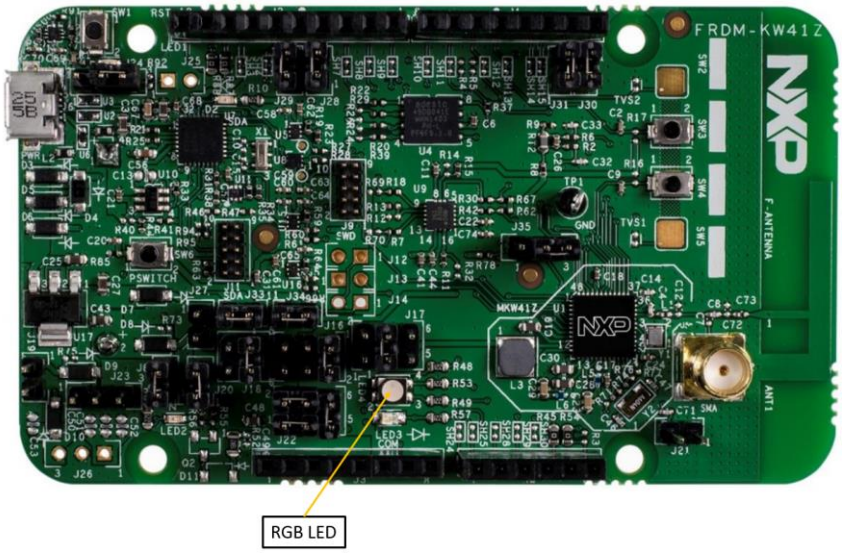


- The next step is to join the FRDM-KW41Z Dimmer Switch and Dimmable Light devices to the network just created. To do so, the network must be first “opened” to joiners. This can be achieved by editing the fields as below and then by pressing the “Permit Join” button.

The value of 0xFFFC mask means “all joiners” and the value 0xB4 is the timeout (0xB4 = 180 decimal = 3 minutes). After this period the network will “close” to joiners until another “permit join” is issued.

- Within the specified timeout period (3 min.), press **SW3** button on the FRDM-KW41Z Dimmable Light device and any button from **SW2 to SW5** on Dimmer Switch device to initiate the join procedure to the existing network. When the joining procedure completes, the RGB LED (LED3) will become solid.
- At this point we have created a small ZigBee network with three devices, the IoT Control Bridge (as Coordinator) and the Dimmer Switch and Dimmable Light devices. Further devices can be added to network.

- Bind the Dimmer Switch together with the Dimmable Light device by pressing the **SW2** button on the target device (Dimmable Light), followed by a long press of the **SW4** button on the initiator device (Dimmer Switch), after the device has been awaked from the low power state.
- Once bound, the Dimmer Switch application can send commands to the On/Off cluster. By pressing the **SW5** button on the Dimmer Switch board, the RGB LED from the Dimmable Light board is switched on and at a **SW4** button press, the LED switches off.



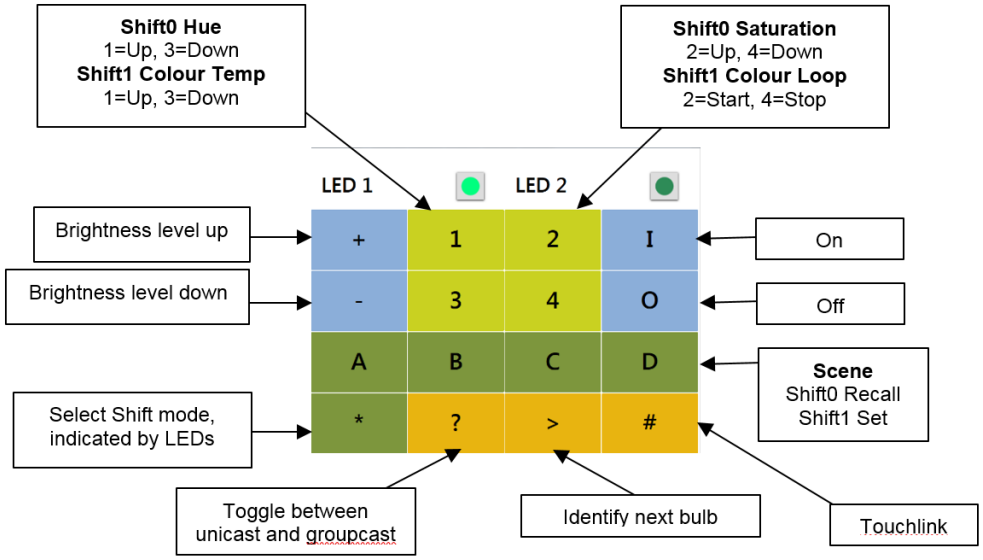
- The Dimmer Switch sends commands to the Level Control cluster on press of the **SW2** and **SW3** buttons to decrease and respectively, to increase the brightness levels of the lights, after the device has been awaked from the low power state.

5.4 Functionality of the Colour Scene Controller

The following setup uses the Colour Scene Controller to exercise commands over the Light devices. A ZigBee End Device (ZED) will be commissioned into the network to exercise the On/Off and Level Control clusters. In this case, the ZED must be bound with the Colour Scene Controller device using the 'Finding and Binding' procedure by pressing the **SW2** button on both devices.

This setup requires at least three FRDM-KW41Z boards, one of the boards will be programmed with the Colour Scene Controller application, one with the Dimmable Light or Colour Temperature Light application and one with a Control Bridge application to form the network, as described in Section 5.3.

The Windows-based Remote Control GUI provides the same buttons and works in the same way as pressing the Colour Scene Controller's buttons would.



Note 1: When the controller device is in the factory-new state, only the buttons # and **+** are available (for commissioning ZLO devices into the network). The # button is used for Touch link, while the + button is used for joining an open network through scanning.

Note 2: The controller device can operate the light device(s) using one of two addressing modes – unicast or groupcast. Unicast mode allows the user to operate a single light device. Groupcast mode allows the user to operate a group of light devices simultaneously. Use the button '**?**' to toggle between these two modes.

The controller device can operate in four Shift modes (0, 1, 2 and 3) to accommodate maximum functionality. The Shift mode is indicated by a combination of two LEDs on the Remote Control Unit, as shown in the table below. You can press button ***** to move to the next Shift mode.

Shift Mode	Left LED	Right LED
Shift0	Off	Off
Shift1	On	Off
Shift2	Off	On
Shift3	On	On

Shift Modes

The four tables below summarise the button functions in the four Shift modes.

Shift0 Mode Operation	Button
On: Send a command to switch on the light(s) The transmission mode will depend on the current mode selected.	**I**

Off with Effect: Send a command to switch off the light(s) using the 'Off with Effect' option - this has the effect of saving the current settings as the Global Scene.	**O**
Increase Brightness: Increase the brightness level of the light(s). If the light is off, this will switch on the light and then increase its level. The brightness will stop increasing when the button is released.	**+**
Decrease Brightness: Decrease the brightness level of the light(s). The brightness will stop decreasing when the button is released.	**–**
Enhanced Move Hue Up: Send a command to move the hue of the light(s) up. The movement will stop when the button is released.	**1**
Enhanced Move Hue Down: Send a command to move the hue of the light(s) down. The movement will stop when the button is released.	**3**
Increase Saturation: Send a command to move the saturation of the light(s) up. The movement will stop when the button is released.	**2**
Decrease Saturation: Send a command to move the saturation of the light(s) down. The movement will stop when the button is released.	**4**
Recall Scene 1: Groupcast a Recall Scene command to restore scene 1.	**A**
Recall Scene 2: Groupcast a Recall Scene command to restore scene 2.	**B**
Recall Scene 3: Groupcast a Recall Scene command to restore scene 3.	**C**
Recall Scene 4: Groupcast a Recall Scene command to restore scene 4.	**D**
Shift Menu: Cycle through the four Shift modes (0 → 1 → 2 → 3 → 0 etc).	*****
Groupcast/Unicast: Toggle between groupcast and unicast transmission modes. On waking from sleep, this mode will always be groupcast. After Touchlinking to a light, the mode will always be unicast with that light selected.	**?**
Select next light: Select the next light in the light database to be controlled by unicast. An Identify command will be sent to the relevant light, and unicast transmission mode will be selected.	**⏵**
Touchlink: Start Touchlink commissioning to add new devices to the network or to gather endpoint information about existing devices in the network.	#

Button Functions in Shift0 Mode

Shift1 Mode Operation	Button
On: Send a command to switch on the light(s). The transmission mode will depend on the current mode selected.	**I**
Off with Effect: Send a command to switch off the light(s) using the 'Off with Effect' option - this has the effect of saving the current settings as the Global Scene.	**O**
Increase Brightness: Increase the brightness level of the light(s). If the light is off, this will switch on the light and then increase its level. The brightness will stop increasing when the button is released.	**+**

Decrease Brightness: Decrease the brightness level of the light(s). The brightness will stop decreasing when the button is released.	**–**
Increase Colour Temperature: Send a command to increase the colour temperature of the light(s). The increase will stop when the button is released.	**1**
Decrease Colour Temperature: Send a command to decrease the colour temperature of the light(s). The decrease will stop when the button is released.	**3**
Set the Colour Loop: Send a command to the light(s) to start a colour loop. The light will start cycling through colours.	**2**
Stop Colour Loop: Send a command to the light(s) to stop the colour loop. A colour loop must be stopped before other colour control commands will be accepted by colour lights.	**4**
Store Scene 1: Groupcast a Store Scene command to save the current settings as Scene 1. Note that following the saving of a scene, the menu level will automatically revert to Shift0.	**A**
Store Scene 2: Groupcast a Store Scene command to save the current settings as Scene 2. Note that following the saving of a scene, the menu level will automatically revert to Shift0.	**B**
Store Scene 3: Groupcast a Store Scene command to save the current settings as Scene 3. Note that following the saving of a scene, the menu level will automatically revert to Shift0.	**C**
Store Scene 4: Groupcast a Store Scene command to save the current settings as Scene 4. Note that following the saving of a scene, the menu level will automatically revert to Shift0.	**D**
Shift Menu: Cycle through the four Shift modes (0 → 1 → 2 → 3 → 0 etc).	*****
Groupcast/Unicast: Toggle between groupcast and unicast transmission modes. On waking from sleep, this mode will always be groupcast. After Touchlinking to a light, the mode will always be unicast with that light selected.	**?**
Select next light: Select the next light in the light database to be controlled by unicast. An Identify command will be sent to the relevant light, and unicast transmission mode will be selected.	**⏵**
Find & Bind: Start find and bind as as a target to add new devices bindings to the binding table.	**#**

Button Functions in Shift1 Mode

Shift2 Mode Operation	Button
On: Send a command to switch on the light(s). The transmission mode will depend on the current mode selected.	**I**
Off with Effect: Send a command to switch off the light(s) using the 'Off with Effect' option - this has the effect of saving the current settings as the Global Scene.	**O**
Increase Brightness: Increase the brightness level of the light(s). If the light is off, this will switch on the light and then increase its level. The brightness will stop increasing when the button is released.	**+**
Decrease Brightness: Decrease the brightness level of the light(s). The brightness will stop decreasing when the button is released.	**–**
Goto Hue and Saturation: Goes to a series of pre-defined enhanced hue values at maximum saturation. Steps up through the series.	**1**
Goto Hue and Saturation: Goes to a series of pre-defined enhanced hue values at maximum saturation. Stepsdown through the series.	**3**
No function assigned	**2**
No function assigned	**4**
No function assigned	**A**
No function assigned	**B**
Network steering: Trigger network steering for a device on a network, broadcast a ZigBee Management command to the network to instruct Routers to set their 'permit joining' state to TRUE for 180 seconds. This opens the network to classical joining.	**C**
Channel Change: Broadcast a ZigBee Management command to change the operational channel to one of the other ZLL primary channels, selected at random.	**D**
Shift Menu: Cycle through the four Shift modes (0 → 1 → 2 → 3 → 0 etc).	*****
Groupcast/Unicast: Toggle between groupcast and unicast transmission modes. On waking from sleep, this mode will always be groupcast. After Touchlinking to a light, the mode will always be unicast with that light selected.	**?**
Select next light: Select the next light in the light database to be controlled by unicast. An Identify command will be sent to the relevant light, and unicast transmission mode will be selected.	**⏵**
Touchlink: Start Touchlink commissioning to add new devices to the network or to gather endpoint information about existing devices in the network.	#

Button Functions in Shift2 Mode

Shift3 Mode Operation	Button Sequence
On: Send a command to switch on the light(s). The transmission mode will depend on the current mode selected.	**I**
Off with Effect: Send a command to switch off the light(s) using the 'Off with Effect' option - this has the effect of saving the current settings as the Global Scene.	**O**
Factory Reset: Factory reset the Remote Control Unit, restoring the application and stack persistent data to its factory-new state.	**- + -**
No function assigned	**1**
No function assigned	**3**
Basic Reset: Send a Basic Cluster Reset Command to the light(s) to reset the attributes of the ZCL	**2**
No function assigned	**4**
Identify Effect Blink: Send a command to the light(s) to trigger the 'Blink' effect.	**A**
Identify Effect Breathe: Send a command to the light(s) to trigger the 'Breathe' effect.	**B**

Identify Effect Okay: Send a command to the light(s) to trigger the 'Okay' effect.	**C**
Identify Effect Channel Change: Send a command to the light(s) to trigger the 'Channel Change' effect.	**D**
Shift Menu: Cycle through the four Shift modes (0 → 1 → 2 → 3 → 0 etc).	*****
Groupcast/Unicast: Toggle between groupcast and unicast transmission modes. On waking from sleep, this mode will always be groupcast. After Touchlinking to a light, the mode will always be unicast with that light selected.	**?**
Select next light: Select the next light in the light database to be controlled by unicast. An Identify command will be sent to the relevant light, and unicast transmission mode will be selected.	**⏵**
Touchlink: Start Touchlink to send a Factory New (reset) command to the target device. This button cannot be used to add new devices to the network.	**#**

Button Functions in Shift3 Mode

6 Over-The-Air (OTA) Upgrade

Over-The-Air (OTA) Upgrade is the method by which a new firmware image is transferred to a device that is already installed and running as part of a network. This functionality is provided by the OTA Upgrade cluster. In order to upgrade the devices in a network, two functional elements are required.

- **OTA Server:** First the network must host an OTA server, which will receive new OTA images from manufacturers, advertise the OTA image details to the network, and then deliver the new image to those devices that request it.
- **OTA Clients:** The second requirement is for OTA clients, which are located on the network devices that may need to be updated. These devices periodically interrogate the OTA server for details of the firmware images that it has available. If a client finds a suitable upgrade image on the server, it will start to request this image, storing each part as it is received. Once the full image has been received, it will be validated and the device will boot to run the new image.

New images are always pulled down by the clients, requesting each block in turn and filling in gaps. The server never pushes the images onto the network.

Note: OTA Upgrades are not supported for Dimmer Switch and Dimmable Light device types.

7 ZigBee Green Power (GP) Switch

This Application Note also provides an example software implementation of a ZigBee Green Power (GP) device based on the following switches:

1. Level Control switch
2. On/Off switch

For information on the NXP implementation of ZigBee Green Power, refer to the ZigBee Green Power User Guide.

Note: This GP device could be implemented on Energy Harvesting (EH) hardware, although in this demonstration EH hardware is not used.

7.1 Loading the application

The table below indicates the GP application binary file supplied with this Application Note as well as the KW41Z hardware components on which the binary can be used. This file is located in the Build directory for the application.

Application	Binary File	Hardware Components
EH_SWITCH	**eh_switch_frdmkw41z.bin**	FRDM-KW41Z

7.2 Using the EH Switch

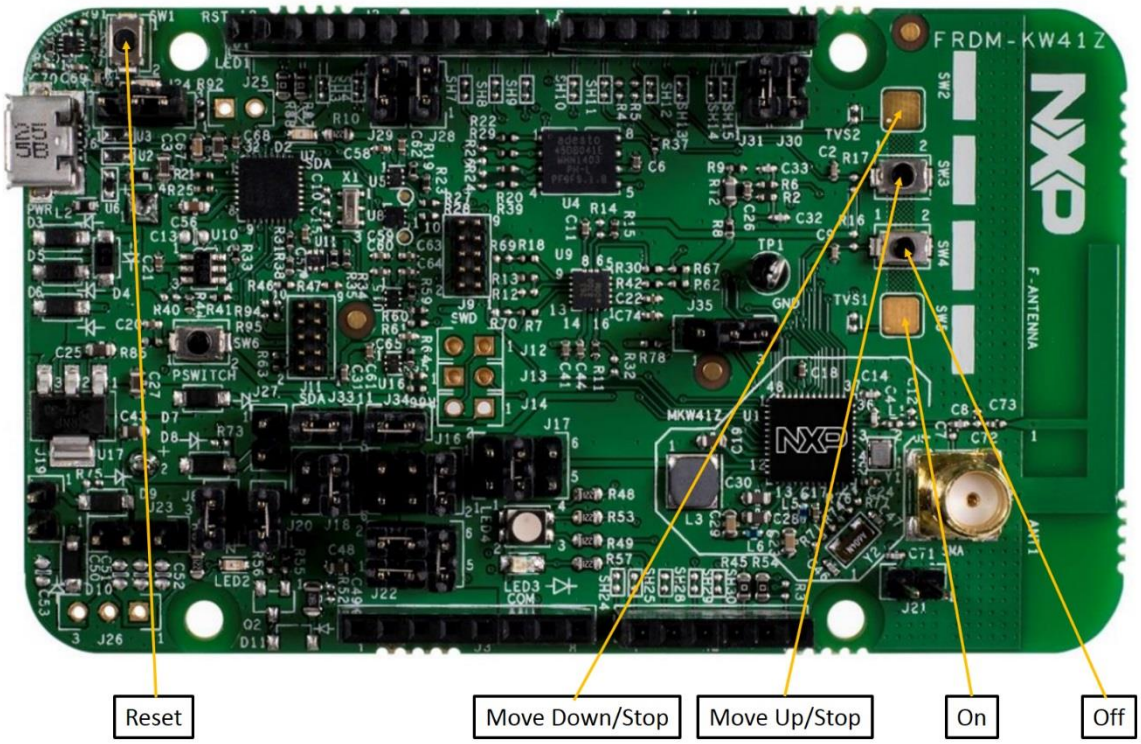
This section describes how to commission and operate the GP Switch in a ZigBee3.0 network with GP 'infrastructure devices'. The network can be set up using:

3. Coordinator application.
4. A device with GP support, e.g. Dimmable Light.

To use this application, you must have programmed the application **eh_switch_frdmkw41z.bin** into the KW41Z module, as described in Section 5.1

7.2.1 GP Switch Functionality

The GP Switch application acts as a GP Energy Harvesting switch, this device can be used to perform the commissioning of a Dimmable Light and control it.



7.2.2 Commissioning

To commission the GP Switch to operate with one or more lights in a ZigBee 3.0 network that support the GP Combo Basic device, follow the procedure below:

1. Put the light into Commissioning mode (SW2 on the FRDM-KW41Z board for the Dimmable Light application). The light will start flashing to identify itself.
2. On the EH Switch node, press any button (e.g. SW4 on the FRDM-KW41Z) repeatedly at one-second intervals until the light stops flashing and returns to its original state (the EH Switch sends commissioning packets on all channels until it finds the light).
3. The EH Switch is now paired with the light node and can be used to control the light. The SW2-SW5 buttons on the FRDM-KW41Z board of the EH Switch can be respectively used to send On, Off, Brighter and Dimmer commands to the light note as presented in section 7.2.1.

7.3 EH Switch Application Code

The code required to build the GP Switch application can be found within the Application Note package in the EH_Switch/Source directory. The GP Switch also uses the button handling file from the Common/Source directory.

8 Developing with the Application Note

The example applications provided in this Application Note were developed using the KW41Z_Zigbee_3.0_Software_v.6.0.6 and IAR Embedded Workbench 7.80 or MCUXpresso v10.1.1.

Throughout your ZigBee 3.0 application development, you should refer to the documentation listed in Section 9.

8.1 Dimmer Switch Application Code

This section describes the application code for App_DimmerSwitch, which is provided in the **Source** directory for the application. You may wish to use this code as a basis for your own application development. You can rebuild your customised application as described in Section 8.4.

App_DimmerSwitch.c is specific to the Dimmer Switch. It includes endpoint registration and constructor, reporting configuration, Basic cluster attribute initialisation, and Identify handler.

The operational state machine (**sDeviceDesc.eNodeState**) is located within **app_zlo_switch_node.c** for the Dimmer Switch. Additional states must be added to this switch statement if further operational modes are required. All the network and ZigBee Base Device related events are also handled inside **app_zlo_switch_node.c**.

Buttons are debounced in **APP_cbTimerButtonScan()** contained in **app_buttons.h**. Any button events are then passed into the **APP_msgAppEvents** queue for processing in **APP_ZLO_vSwitch_Task()** in **app_zlo_switch_node.c** where the button event is eventually handled in **vApp_ProcessKeyCombination()**. Any alteration to the key map to allow different functionality should go in this function.

Similarly, the **vApp_ProcessKeyCombination()** function can be modified to add a function that is called upon release of a key.

The function that handles a button press and release for different operational modes is located in **app_switch_state_machine.c**

bdb_options.h defines the parameters used by the ZigBee Base Device, such as primary and secondary channel masks.

zcl_options.h defines the ZCL options, such as which clusters are supported, whether a client and/or a server, and which optional commands and attributes are supported. Mandatory commands and attributes of the selected cluster will be automatically included.

8.2 Colour Scene Controller Application Code

This section describes the application code for App_ColorSceneController, which is provided in the **Source** directory for the application. You may wish to use this code as a basis for your own application development. You can rebuild your customised application as described in Section 8.4.

Code in **Source** holds the files specific to the Colour Scene Controller, but also the files to configure the ZigBee Base Device and ZigBee Cluster Library.

App_ColorSceneController.c contains code specific to the Colour Scene Controller, dealing with endpoint registration and constructor, and initialisation of the Basic cluster attributes.

bdb_options.h defines the parameters used by the ZigBee Base Device, such as primary and secondary channel masks.

zcl_options.h defines the ZCL options, such as which clusters are supported, whether a client and or a server, and which optional commands and attributes are supported. Mandatory commands and attributes of the selected cluster will be automatically included.

8.3 Common Code

The following are the main application files and are common to all controller type devices.

app_start_controller.c manages the chip start-up, calls the initialisation functions and launches the main program loop.

app_main.c hosts the main program loop, and defines and initialises system resources, queues, timers etc.

zlo_controller_node.c hosts the event handlers for the application and the ZigBee Base Device callback. This callback receives ZigBee Base Device events and AF Stack events after the Base Device has completed any processing that it requires. These events can then be further processed by the application. These events include data indications that are passed to the ZCL for processing, and network management events, such as Joined or Failed to Join events, in order to keep the application informed of the network state. The application event queue is processed to receive button-press events which are passed to the menu event handler. Sleep scheduling and polling for data are also handled here.

app_colour_commands.c contains a collection of functions that interact with the Colour Control cluster of the ZCL to send colour control commands.

app_general_commands.c contains a set of utility functions for the menu handler.

app_group_commands.c contains a collection of functions that interact with the Groups cluster of the ZCL to send group control commands.

app_identify_commands.c contains a collection of functions that interact with the Identify cluster of the ZCL to send identify control commands.

app_level_commands.c a collection of functions that interact with the Level Control Cluster of the ZCL to send level control commands.

app_on_off_commands.c contains a collection of functions that interact with the On/Off cluster of the ZCL to send on/off control commands.

app_scenes_commands.c contains a collection of functions that interact with the Scenes cluster of the ZCL to send scene recall and save commands.

app_serial_interface.c contains functions to process a character received from the UART, manage the serial interface protocol, validate messages and pass key events to the application event queue. It also manages the protocol to send LED messages to the GUI.

uart.c contains an interrupt handler and deals with UART management for the serial interface.

8.4 Rebuilding the Applications

This section describes how to rebuild the supplied applications, which you will need to do if you customise the applications for your own use.

8.4.1 Pre-requisites

It is assumed that you have installed the relevant NXP development software on your PC, as detailed in Section 2.

The project files should be located in the directory:

..\boards\frdmkw41z\wireless_examples\zigbee_3_0\[application name]\freertos

The ZigBee devices' projects presented in the use case are highlighted below:

The application's code is located at the following relative path, where "x.y.z" is the software current version. Each of the applications has 2 subfolders, **Config** and **Source**.

..\middleware\wireless\zigbee_3_0_x.y.z\examples

8.4.2 Build Instructions

The software provided with this Application Note can be built for the KW41Z device. The applications can be built from IAR Embedded Workbench and MCUXpresso.

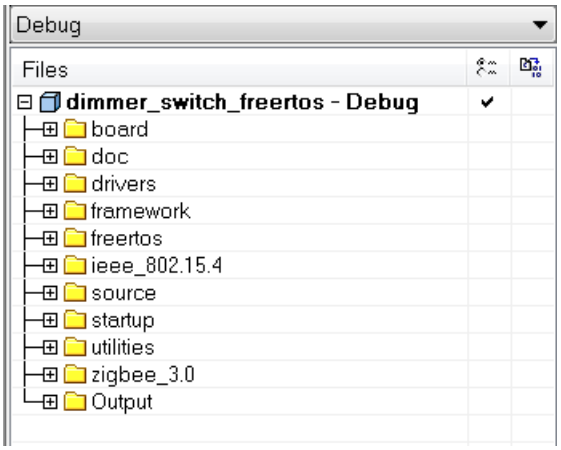
8.4.2.1 Using IAR Embedded Workbench

This section describes how to use IAR Embedded Workbench to build the example application. To build the application and load it into KW41Z boards, follow the instructions below.

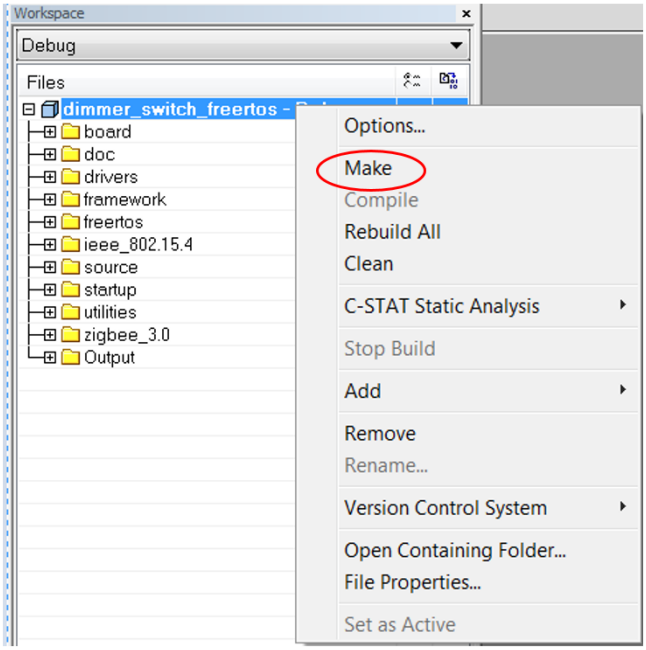
In this application note, the Dimmer Switch project is considered. The project workspace file is located at the following relative path:

..\boards\frdmkw41z\wireless_examples\zigbee_3.0\dimmer_switch\freertos\iar\dimmer_switch_freertos.eww

1.Open the project workspace file by double click or by drag and drop into the IAR EW IDE.

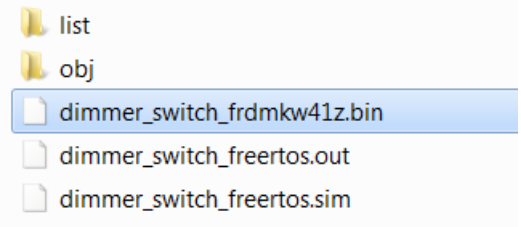


2.Build an application. To do this use the drop down menu at the top of the Workspace panel to select the relevant build configuration, Debug or Release, right click on the application name in the Workspace panel and select Make. Repeat this to build the other applications.



After the build process completes, the output folder will contain the executable application (*.out) and the map file. The binary file is also generated, but is not visible in this view. To access it, right click on the *.out file and then click on “Open Containing Folder”.

The containing folder will be opened in Windows Explorer and exactly in the same folder, the binary file is located.



3.Load the resulting binary files into the board. You can do this using the integrated flash programmer in IAR or using Firmware Loader from NXP Test Tool 12.

8.4.2.2 Using MCUXpresso

In order to load a project using MCUXpresso, please refer to the documentation provided, “Getting started with MCUXpresso SDK.pdf “and follow the steps described in chapter 7 “Run a demo using MCUXpresso IDE”.

9 Application Code Sizes

The applications of this Application Note have the following memory footprints on the KW41Z device, when using the KW41Z_ZigBee_3.0_Software_v6.0.6 SDK.

Application	Read-only code (Bytes)	Read-only data (Bytes)	RW data (Bytes)
App_DimmerSwitch	206384	14888	30813
App_ColorSceneController	217232	15220	29818

10 Release Details

10.1 Compatibility

Product Type	Version	Supported Chips
Version 1001		
FRDM-KW41Z	Rev A/A2/A3	KW41Z
IAR Embedded Workbench	V8.22.2	KW41Z
MCUXpresso	v10.2.0	KW41Z
KW41Z_Zigbee_3.0_Software_v6.0.7	v6.0.7	KW41Z

11 Related Documents

The following manuals will be useful in developing custom applications based on this Application Note:

- ZigBee 3.0 Stack User Guide
- ZigBee Device User Guide
- ZigBee Cluster Library User Guide

All the above manuals are available as PDF documents from the ZigBee 3.0 page of the NXP web site.

Revision History

Version	Notes
0	First KW41Z release
1	Updates for KW41Z ZigBee 3.0 Alpha/EAR Release
2	Updates for KW41Z ZigBee 3.0 Beta/PRC Release
3	Updates for KW41Z ZigBee 3.0 GA/RFP Release
4	Updates for KW41Z Zigbee 3.0 Maintenance Release1

Important Notice

NXP Semiconductors

For the contact details of your local NXP office or distributor, refer to:

www.nxp.com