ZBEXDK006UG
ZigBee Demo Applications User's Guide

ZigBee Examples DK006: JN5189/JN5188/K32W061/K32W041
Rev. 0 — 19 February 2020 User's Guide

Contents

1 Overview

This document applies to the JN5189, JN5188, K32W061, and K32W041 ZigBee 3.x wireless microcontrollers used with the DK006 (OM15076 Development Board) platform. These microcontrollers are referred to as the ZigBee devices throughout this document.

The Getting Started with MCUXpresso SDK for JN5189/K32W061 User's Guide contains instructions for installing MCUXpresso.

This document should be used in conjunction with the ZigBee example applications to demonstrate the features and operation of the Base Device in a ZigBee 3.x network that employs the NXP JN518x/K32W061/K32W041 (ZigBee Device) microcontrollers.

The examples provide sufficient functionality to demonstrate ZigBee Base Device and can be a starting point for developing real world devices.

2 Introduction

A ZigBee 3.x wireless network comprises a number of ZigBee software devices that are implemented on hardware platforms to form nodes. These ZigBee examples are concerned with implementing the ZigBee Base Device on the NXP ZigBee Device (JN5188, JN5189, K32W06 or K32W041).

This application note document provides example implementations of the following ZigBee logical device types:

- Coordinator
- Router
- End Device with Receiver always ON
- End device with Receiver OFF when Idle (“Sleepy End Device”).

The examples of the above device types are not real-world devices but provide the basic behavior required by the ZigBee Base Device Behavior Specification. These examples should serve as templates on which to base further development into real physical devices. The ZigBee Base Device is introduced and detailed in the *ZigBee 3.0 Devices User Guide [JN-UG-3131]*.

The ZigBee Base Device Behavior Specification provides definitions, procedures and methods for forming, joining, and maintaining ZigBee 3.x networks. It also defines the method for service discovery, in which a client and server of an operational cluster are bound together in order to achieve the functionality of the physical devices.

NOTE

If you are not familiar with ZigBee 3.x, you are advised to refer the ZigBee 3.0 Stack User Guide [JN-UG-3130] for a general introduction.

3 Development environment

- Software
- Hardware

3.1 Software

To use the ZigBee examples, you must install MCUXpresso and Software Developer Kit (SDK) that are appropriate for your DK006 wireless microcontroller:

- MCUXpresso IDE
- JN518x ZigBee 3.0 SDK
- K32W061/K32W041 ZigBee 3.0/Bluetooth SDK

The MCUXpresso software and installation instructions are described in the Getting Started with MCUXpresso SDK for JN5189/ K32W061.

The DK006 wireless microcontroller-specific resources and documentation are available via the MCUXpresso website to authorized users.

3.2 Hardware

Hardware kits are available from NXP to support the development of ZigBee 3.x applications. The following kits provide a platform for running these applications:

- JN518x-DK006 Development Kit, which features JN5189 devices
- IOTZTB-DK006 Development Kit, which features K32W061 & JN5189 devices

4 Examples overview

The example applications provided are listed in the following table.

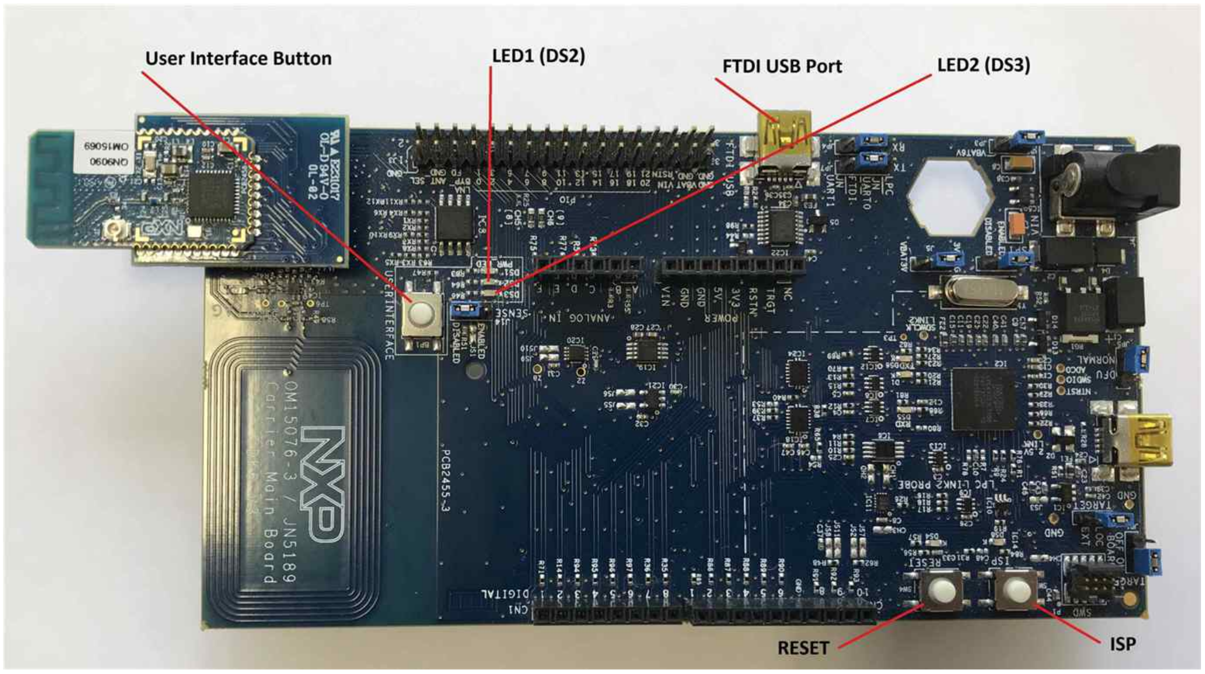
Table 1. Example Applications and Device Types

Application	Device Type
Coordinator	Coordinator/Trust Centre
Router	Router
End Device RX On	End Device (Non-Sleeping)
End Device RX Off when Idle	End Device (Sleeping)

The examples are provided as part of the wireless examples for ZigBee. The binaries generated by these examples are targeted to run on the DK006 board supplied in the JN5189-DK006 or IOTZTB-DK006 development kits.

4.1 DK006 base board

The DK006 OM15076-3 Carrier Board is used by all of the ZigBee examples. The important LEDs and user control buttons are highlighted below.



5 Running the demonstration examples

This section describes how to use the supplied pre-built binaries to run the example applications on a ZigBee device. All the applications run on a JN518x, K32W061 or K32W041 module (ZigBee module) on a OM15076 DK006 Carrier Board. The examples do not use any other expansion boards. All the ZigBee wireless demo application are configured to use a default channel 12. SINGLE_CHANNEL=12 in the pre-processor configuration can be overwritten to change to a different channel.

The ZigBee examples work in the Centralized (Trust Centre) network only, and all devices are expected to join with ZigBee09 key. Once joined, the devices then automatically negotiate a new TCLK key.

For details of the differences between a Centralized (Trust Centre) network and a Distributed network, see the *ZigBee Devices User Guide [JN-UG-3114]*.

5.1 Loading the applications

The table lists the binaries generate by the examples.

Table 2. Application binaries and hardware components

Hardware Platform	Binary Files

OM15076-3	coordinator.bin
	routerClient.bin
	enddevice_rxonClient.bin
	enddevice_rxoffClient.bin

JN5189-USB Dongle / K32W0161-USB Dongle	coordinator.bin

A binary file can be loaded into the Flash memory of the ZigBee device using the *DK6 Production Flash Programmer [JN-SW-4407]*. This software tool is described in the *DK6 Production Flash Programmer User Guide [JN-UG-3127]*.

To load an application binary file into a ZigBee module on a DK006 Carrier Board, follow the instructions below:

1. Connect a USB port of your PC to the USB Mini B port marked “FTDI USB” on the DK006 Carrier Board (next to the 34-pin header) using a ‘USB A to Mini B’ cable. You may be prompted to install the driver for the USB virtual COM port.
2. Determine which serial communications port on your PC has been allocated to the USB connection.
3. Put the ZigBee device into programming mode by holding down the ISP button while pressing and releasing the RESET button.
4. To program the image, you can run the command below:

   ```
   dk6Programmer.exe -e FLASH -s <COM PORT> -p <FULL PATH TO THE BIN>
   ```

   This programs the image in flash at the offset 0 and erases all of flash. This starts the device in a factory new state.
5. Once the download has successfully completed, reset the board or module to run the application.

Similarly, to load an application binary into a JN5189-USB/K32W061-USB dongle, follow the instructions below:

1. Insert the USB dongle into the USB port of your PC.
2. Determine which serial communications port on your PC has been allocated to the USB dongle.
3. To program the image, run the command below:

   ```
   dk6Programmer.exe -e FLASH -s <COM PORT> -p <FULL PATH TO THE BIN>
   ```

   This programs the image in flash at the offset 0 and erases all of flash. This starts the device in a factory new state. The US dongle is automatically placed into programming mode by the DK6 Programmer and reset at the end of programming where the application started.

5.2 Coordinator functionality

The functionality of the Coordinator application is described below.

The Coordinator is responsible for initially forming the network and then, via the Trust Centre functionality, managing which other devices can join the network and distributing security materials to those that are allowed to join. The Coordinator supports the mandatory clusters and features of the Base Device as defined in the ZigBee Base Device Behavior Specification.

For demonstrating the ‘Finding and Binding’ functionality, the Coordinator also supports the On/Off Cluster as a client.

The coordinator can also act as an OTA server. This enables the coordinator to advertise any ZigBee OTA upgrade image. The maximum image size which can be hosted is 288 Kbytes. It is assumed that the example ZigBee coordinator memory footprint remains less than 288 Kbytes.

The Coordinator is controlled using serial commands issued from a terminal program running on a PC connected to the Zigbee device through the USB connection. The Coordinator application is configured to communicate at 115200 baud, 8bit data, 1 stop bit, no parity or flow control. The serial interface in not case-sensitive. For a summary of the serial interface, refer to Summary of Serial Interface Commands.

5.2.1 Forming a network

A network can be formed from a factory-new Coordinator (Network Steering while not on a network), Enter “form” on the serial interface (Dongle or Carrier Board).

The Coordinator then starts a network. Using a ZigBee packet sniffer (running on a separate, on a USB Dongle), the periodic link status messages should be present on the operational channel.

5.2.2 Allowing other nodes to join (network steering)

Once a network has been formed, the network must be opened to allow other devices to join (Network Steering while on a network), to initiate this enter “steer” on the serial interface (Dongle or Carrier Board).

The Coordinator then broadcasts a Management Permit Join Request to the network to open the ‘permit join’ window for 180 seconds. The Network Steering process (for devices not on a network) can now be triggered on the devices that are to join the network.

5.2.3 Binding nodes

‘Finding and Binding’ is the process whereby ZigBee controlling devices find ZigBee controlled devices by matching operational clusters and create entries in the Binding table. The Coordinator supports Finding and Binding as an ‘Initiator’ that tries to find targets to bind to. For demonstration, the Coordinator supports the On/Off Cluster as a client, so the Finding and Binding process looks for devices that support the On/Off cluster as a server in order to create bindings.

To start Finding and Binding as an Initiator, first trigger Finding and Binding on a second ‘Target’ device (such as the Router in this ZigBee example) by pressing the User Interface button on the 'Target' device.

NOTE

The target device must have first joined the ZigBee network.

Next, enter “find” on the serial interface (of the Coordinator).

When Finding and Binding for a target has completed and a binding has been created, the Coordinator sends an Identify Off command to the target device, in order to signal completion of the process for the Target. Depending on the type of bindings being created (either unicast or groupcast), an Add Group command may be sent to the target device.

Reporting is a mandatory feature in ZigBee 3.x. A device wishing to receive periodic and on-change reports from an operational server should create a remote binding for itself on the target device. This Coordinator sends a Binding Request to a target with an On/Off cluster server. It then receives periodic and on-change reports from that device, reporting the state of the OnOff attribute (0x0000) of the On/Off cluster. The frequency of the reports depends on the default report configuration of the individual target device. The device receiving the reports can request the change by sending a Report Configuration command.

5.2.4 Operating the device

The operational functionality of this device in this demonstration is provided by the On/Off cluster. You can now send an OnOff Toggle command to the bound devices (in the Binding table) when you Enter “toggle” in the serial interface (Carrier Board).

5.2.5 Rejoining the network

As a Coordinator, when this device is restarted in a state which is not factory-new, it resumes operation in its previous state. All application, binding, group, and network parameters are preserved in non-volatile memory.

5.2.6 Performing a factory reset

The Coordinator can be returned to its factory-new state, erasing all persistent data except the outgoing network frame counter, if you enter “factory reset” on the serial interface.

5.2.7 Summary of serial interface commands

The serial port connection to the Coordinator application is configured to use 115200 baud, 8 data bits, 1 stop bit, no parity. The serial commands are not case-sensitive.

Table 3. Serial interface commands

Serial Command	Action
toggle	Sends an OnOff Toggle command to bound devices
steer	Triggers Network Steering for a device on the network

Table continues on the next page...

Table 3. Serial interface commands (continued)

Serial Command	Action
form	Triggers network formation for a device not on a network
find	Triggers Finding and Binding as an Initiator
factory reset	Factory resets the device, erasing persistent data
soft reset	Triggers a software reset (no loss of data)

5.3 Router functionality

For demonstrating the ‘Finding and Binding’ functionality, the Router also supports the On/Off Cluster as a server.

5.3.1 Forming or joining a network

The Router is capable of only joining an existing network. If it does not find the network it continues discovering the network till it can find a network to join.

5.3.1.1 Joining an existing network using network steering

A factory-new Router can only join an existing ZigBee once the network has been opened to accept new joiners (Network Steering for a device on a network). This is achieved as follows:

1. Trigger Network Steering on one of the devices already on the network (Coordinator or another Router in the same ZigBee network).
2. Then reset (using the RESET button) or power-on the joining Router device.

This causes the Router to start a network discovery and the associate process. Association is followed by an exchange of security materials and an update of the Trust Centre Link Key (if joining a Centralized Trust Centre Network).

If the join is unsuccessful, it can be retried by power-cycling.

Allowing Other Devices to Join the Network

Once the Router is part of a network, the network must be opened to allow other devices to join (Network Steering while on a network).

To do this, press the USER INTERFACE button on the Carrier Board (the same button is also used to start Finding and Binding, described in Binding Devices).

The Router then broadcasts a Management Permit Join Request to the network to open the ‘permit join’ window for 180 seconds. The Network Steering process (for devices not on a network) can now be triggered on the devices that are to join the network.

5.3.2 Operating the Device

The operational functionality of this device in this demonstration is provided by the On/Off cluster. Since the device supports the On/Off cluster server, its operation is passive and it responds to commands sent by bound devices. It responds to an OnOff Toggle command from a bound controller device by toggling the LED1 on the carrier board.

5.3.3 Rejoining a network

As a Router, when this device is restarted in a state which is not factory-new, it resumes operation in its previous state. All application, binding, group, and network parameters are preserved within non-volatile memory of the device.

5.3.4 Performing a factory reset

The Router can be returned to its factory-new state (erasing all persistent data except the outgoing network frame counter) as follows:

- Hold down the USER INTERFACE button and press the RESET button on the Carrier Board.

The Router then broadcasts a Leave Indication on the old network, then delete all persistent data (except the outgoing network frame counter) and perform a software reset.

There are two supported over-the-air commands for removing a device from the network - these are:

- Network Leave Request without rejoin
- ZDO Management Network Leave Request without rejoin

The Reset command of the Basic cluster causes the ZCL to be reset to its factory-new defaults, resetting all attributes and configured reports. This does not remove the device from the network and all network parameters, groups and bindings will remain in place.

5.4 End device functionality

The End Device is not capable of either forming a network or being a parent to other devices joining the network. There are two types of End Device, 'RX On' devices, which are always ready to communicate in the network, and sleepy 'RX Off when Idle' End Devices which can sleep for periods of time during which it can not communicate. The End Device supports the mandatory clusters and features of the Base Device as defined in the ZigBee Base Device Behavior Specification.

For purpose of demonstrating the ‘Finding and Binding’ functionality, the End Device also supports the On/Off cluster as a client.

All communications to/from the End Device are passed through its parent Coordinator or Router. In the case of an RX Off when Idle End Device communication is initiated from the End Device through Poll Requests. The parent devices will buffer data for the child End Device for a period of time so the End Device must send periodic Poll Requests to its parent in order to receive any messages which may be waiting for it.

For the RX On device no poll is required and messages are sent directly to it from the parent. However, regular messages must be sent otherwise the device may get aged out.

5.4.1 Joining an existing network using network steering

A factory-new End Device can join an existing network once the network has been opened to accept new joiners (Network Steering for a device on a network). This is achieved as follows:

1. Trigger Network Steering on one of the devices already on the network.
2. Then reset (using the RESET button) or power-on the End device.

5.4.2 Operating the device

The operational functionality of this device in this demonstration is provided by the On/Off cluster and requires that the OM15082 Generic Expansion Board is fitted to the expansion headers of the DK006 Carrier Board. You can now send an OnOff Toggle command to the bound devices bound (in the Binding table) as follows

- Press the button SW1 on the OM15082 Generic Expansion Board.

This effect that the command has on a bound device depends on the functionality related to the On/Off cluster on the device – for the Router in this demonstration, it toggles a light (see Router Functionality).

5.4.3 Rejoining a network

As an End Device, when this device is restarted in a state which is not factory-new, it automatically sends a Network Rejoin Request to re-establish contact with its previous parent. If this fails, it then tries to join any Router on the network that will host it. The rejoin is attempted at power-on and when woken from deep sleep. All application, binding, group, and network parameters are preserved in non-volatile memory.

5.4.4 Performing a factory reset

The End Device can be returned to its factory-new state (erasing all persistent data except the outgoing network frame counter) as follows:

- Hold down the USER INTERFACE button and press the RESET button on the Carrier Board.

The End Device will then unicast a Leave Indication to its parent, which rebroadcasts this message to the old network. The End Device deletes all persistent data (other than the outgoing network frame counter) and perform a software reset.

There are two supported over-the-air commands for removing a device from the network - these are:

- Network Leave Request without rejoin
- ZDO Management Network Leave Request without rejoin

The Reset command of the Basic cluster causes the ZCL to be reset to its factory-new defaults, resetting all attributes, and configured reports. This does not removes the device from the network and all network parameters, groups, and bindings remain in place.

5.5 Binding devices

The Router and End Device supports the On/Off cluster as a server and implements the Finding and Binding process as a Target. To trigger Finding and Binding as a target, do the following:

1. Press the USER INTERFACE button on the Carrier Boards of all the target devices (the same button is used to start Network Steering, described in Allowing Other Devices to Join the Network).
2. Start Finding and Binding on the Initiator device.

This causes the End Device or router to self-identify for 180 seconds, while the Initiator tries to find the identifying devices, query their capabilities and create bindings on those with matching operational clusters. As part of this process, the Route or End Device may receive an Add Group Command and/or a Binding Request Command.

Reporting is a mandatory feature in ZigBee 3.x. The Router and End Device supports the On/Off cluster as a server and the OnOff attribute of this cluster is a reportable attribute as defined in the ZigBee Base Device Behavior Specification. The Router and End Device holds a default configuration for reporting the state of the OnOff attribute. Once a device wishing to receive these periodic and on-change reports has created a remote binding, the Router starts to send reports to this bound device. The frequency of the reports depends on the default report configuration of the individual target device, 60 seconds in this case. The device receiving the reports can request the change by sending a Report Configuration command.

6 ZigBee 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 ZigBee network. This functionality is provided by the OTA Upgrade cluster. In order to upgrade the devices on a network, two functional elements are required.

- **OTA Server:** First the network must host an OTA server, which receives 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 on the network devices that may 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 starts to request this image, storing each part as it is received. Once the full image has been received, it validates and the device boots to run the new image.

The clients pull down the new images, requesting each block in turn and filling in the gaps. The server never pushes the images onto the network.

6.1 Overview

Support for the OTA Upgrade cluster as a client has been included for the Router and End Device devices. By default all the devices only support encrypted OTA.

The internal Flash memory is used to store the upgrade image by default.

The initial client binaries to be programmed into the DK006 devices are Version 1 files:

routerClient.bin

enddevice_rxonClient.bin

The OTA images are V2/V3 .ota files:

- routerClient_UpgradeImagewithOTAHeaderV2_Enc.ota
- routerClient_UpgradeImagewithOTAHeaderV3_Enc.ota
- enddevice_rxoffClient_UpgradeImagewithOTAHeaderV2_Enc.ota
- enddevice_rxoffClient_UpgradeImagewithOTAHeaderV3_Enc.ota
- enddevice_rxonClient_UpgradeImagewithOTAHeaderV2_Enc.ota
- enddevice_rxonClient_UpgradeImagewithOTAHeaderV3_Enc.ota

6.2 OTA upgrade operation

To add an image to the coordinator the OTA images must be programmed as follows:

Dk6Programmer.exe -s <COM port> -p FLASH@294912="< .OTA *image to program>"*

When adding an image to a non-factory new coordinator, care must be taken not to use the -e option to erase flash.

Any devices with OTA clients in the network periodically send Match Descriptor Requests in order to find an OTA server. Once a server responds, it then sends an IEEE Address Request in order to confirm its address details. The clients the periodically send OTA Image Requests to determine whether the server is hosting an image for that client device. In response to the Image Request, the server returns details of the image that it currently hosts - Manufacturer Code, Image Tag, and Version Number. The client checks these credentials and decides whether it requires this image. If it does not, it will query the server again at the next query interval. If the client does require the image, it starts to issue Block Requests to the server to get the new image. Once all blocks of the new image have been requested and received, the new image is verified, the old one invalidated, and the device reboots and runs the new image. The client resumes periodically querying the server for new images.

The End Device which is RX Off is allowed to enter sleep, It stays awake for 5 seconds and then sleeps for 1 second when not doing find and binding.

6.3 Image credentials

There are four main elements of the OTA header that are used to identify the image, so that the OTA client is able to decide whether it should download the image. These are Manufacturer Code, Image Type, File Version, and OTA Header String:

- Manufacturer Code: This is a 16-bit number that is a ZigBee-assigned identifier for each member company. In this application, this number has been set to 0x1037, which is the identifier for NXP. In the final product, this should be changed to the identifier of the manufacturer. The OTA client compares the Manufacturer Code in the advertised image with its own and the image downloads only if they match.
- Image Type: This is a manufacturer-specific 16-bit number in the range 0x0000 to 0xFFBF. Its use is for the manufacturer to distinguish between devices. In this application, the Image Type is normally set to the ZigBee Device Type however this application uses 0x0001 for the Router device and 0x0002 for the End Device (0x8001 and 0x8002 are used for encrypted images). The OTA client compares the advertised Image Type with its own and only downloads the image if they match. The product designer is entirely free to implement an identification scheme of their own.
- File Version: This is a 32-bit number representing the version of the image. The OTA client compares the advertised version with its current version before deciding whether to download the image.
- OTA Header String: This is a 32 byte character string and its use is manufacturer- specific. In this application, the OTA client compares the string in the advertised image with its own string before accepting an image for download. If the strings matches, then the image is accepted. In this way, the string can be used to provide extra detail for identifying images, such as hardware subtypes.

6.4 Upgrade and downgrade

The decision to accept an image following a query response is under the control of the application. The code, as supplied, accepts an upgrade or a downgrade. As long as the notified image has the right credentials and a version number which is different from the current version number, the image is downloaded. For example, if a client is running a v3 image and a server is loaded with a v2 image then the v2 image is downloaded. If it is required that the client should only accept upgrade images (v2 -> v3 -> v5), or only accept sequential upgrade images (v2 -> v3 -> v4 -> v5) then the application callback function that handles the Image.

6.5 LED indication table

- Coordinator
- Router
- End device RX on
- End device RX off (sleepy device)

6.5.1 Coordinator

Table 4. Coordinator

LED1	LED2	NOTES
OFF	OFF	Device is not on the network
OFF	BLINKING ON/OFF every 500 ms	Network Steering/Permit join is active
OFF	BLINKING ON/OFF every 1 second	Find and Bind initiated and in still active
OFF	ON	Device is active
OFF	BLINKING ON/OFF every 250 ms	Both Network steering/Permit join and find and bind are active.

6.5.2 Router

Table 5. Router

LED1	LED2	NOTES
OFF	OFF	Device is not on the network
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	BLINKING ON/OFF every 250 ms	Network Steering/Permit join is active
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	BLINKING ON/OFF every 2 seconds	OTA aborted or failed
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	BLINKING ON/OFF every 500 ms	OTA in progress

6.5.3 End device RX on

Table 6. End device RX on

LED1	LED2	NOTES
OFF	OFF	Device is not on the network
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	BLINKING ON/OFF every 250 ms	Find and Bind active
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	BLINKING ON/OFF every 2 seconds	OTA aborted or failed
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	BLINKING ON/OFF every 500 ms	OTA in progress

6.5.4 End device RX off (sleepy device)

Table 7. End device RX off (sleepy device)

LED1	LED2	NOTES
OFF	OFF	Device is not on the network
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	ON	Device is not sleeping and is active
OFF/ON (Current ON/OFF cluster status) or BLINKING ON/OFF (Identifying)	Blinking ON for 5 seconds and off for 1 second	Device is going through sleep and wake cycle

7 Related documents

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

- ZigBee 3.0 Getting Started Application Note [JN-AN-1260]
- DK6 Production Flash Programmer User Guide [JN-UG-3127]
- ZigBee 3.0 Stack User Guide [JN-UG-3130]
- ZigBee 3.0 Devices User Guide [JN-UG-3131]
- ZigBee 3.0 Cluster Library User Guide [JN-UG-3132]
- Core Utilities User Guide [JN-UG-3133]
- ZigBee 3.0 Green Power User Guide [JN-UG-3134]
- Encryption Tool User Guide [JN-UG-3135]

8 Revision history

This table summarizes revisions to this document.

Table 8. Revision history

Revision number	Date	Substantive changes
0	02/2020	Initial release

Date of release: 19 February 2020

Document identifier: ZBEXDK006UG