ZigBee™ environment Demonstration (ZeD)

Embedded Software Design User’s Guide

Document Number: ZEDESDUG
Rev. 1.3
10/2008

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only*:**
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Contents

About This Book

Chapter 1
Introduction

Chapter 2
Getting Started With ZeD Embedded Applications

Chapter 3
ZeD Embedded Projects Configuration

Chapter 4
ZigBee Test Client (ZTC) Interface

About This Book

This guide describes how to develop an application over the Freescale BeeStack, using the Home Automation Profile, for the Freescale MC1321x Evaluation Kit (1321x-EVK) with features required by the ZigBee environment Demonstration (ZeD). This guide is part of the Freescale ZeD Software Package.

This guide is best used in conjunction with the ZigBee 2006 specification available at www.zigbee.org and the documentation regarding the general design of the software stack.

Audience

This document is intended for software, hardware, and system engineers who are responsible for working with BeeStack.

Organization

This document is organized into 4 chapters.

Chapter 1	Introduction — Provides an introduction to the software.
Chapter 2	Getting Started With ZeD Embedded Applications — Helps users get the embedded applications up and running quickly.
Chapter 3	ZeD Embedded Software Configuration — Provides a description of the available ZeD configurations.
Chapter 4	ZigBee Test Client (ZTC) Interface — Provides a description of the ZTC interface between the PC Application and the embedded applications.

Revision History

The following table summarizes revisions to this document since the previous release (Rev. 1.2).

Revision History

Location	Revisions
Entire document	Updated for release of MC1322x EVK.

Definitions, Acronyms, and Abbreviations

The following list defines the acronyms and abbreviations used in this document.

BeeApps	Applications implemented over BeeStack
BeeKit	BeeKit Wireless Connectivity Toolkit
BeeStack	The ZigBee Stack implementation from Freescale
Bind	Binding is a connection state between one to many devices routed through the ZigBee coordinator. For example, a switch sends an indirect toggle message to a light.
CAP	Contention Access Period
IEEE Address	The unique IEEE-address or MAC-address is given to a manufacturer of a device by IEEE.
PAN	Personal Area Network
PAN ID	PAN Identifier
PIB	Personal area network information base
RxOnWhenIdle	RxOnWhenIdle indicates that the device is expected to receive packets over the air during idle portions of the CAP
SAP	Service Access Point
ZCL	ZigBee Cluster Library
ZeD	ZigBee environment Demonstration

References

The following documents were referenced to build this document.

1. ZigBee Document 053474r13, ZigBee Specification (www.zigbee.org)
2. ZigBee Document 053520r16, ZigBee Home Automation Profile Specification (www.zigbee.org)
3. ZigBee Document 075123r01, ZigBee Cluster Library Specification (www.zigbee.org)
4. ZigBee environment Demonstration (ZeD) Quick Start Guide
5. Freescale ZigBee Application User’s Guide
6. Freescale BeeStack Software Reference Manual
7. BeeKit Wireless Connectivity Toolkit User’s Guide

Chapter 1 Introduction

The Freescale ZigBee environment Demonstration (ZeD) software was built for demonstration purposes to provide users with an easy to use and easy to understand presentation of ZigBee features (binding, association etc.) combined with the Home Automation Device profile (On/Off Light, On/Off Switch). It is necessary to define a clear interface between the ZeD Embedded software and the PC software component.

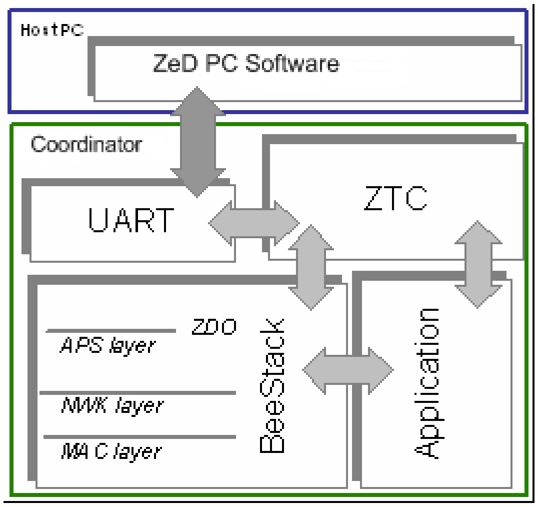


Figure 1-2. Interface Between ZeD Coordinator and ZeD PC Software

This chapter introduces the ZeD embedded component as well as the interacting PC software component and outlines the interface between them.

1.2 ZeD Primary Components

The components in the ZeD system are as follows:

- ZeD PC Software Component — Interacts with the ZeD coordinator through a UART interface with a specified ZigBee Test Client (ZTC) interface.
- ZeD Coordinator — Interacts with the ZeD PC software component. Acts as a normal coordinator on the network. It contains the Combined Interface Device implementation, part of the Home Automation Profile.
- ZeD Router — On/Off Light Device, part of the Home Automation Profile.
- ZeD End Device — Such as On/Off Switch Device, Temperature Device, Dimmer Switch Device, Thermostat Device or Dimmable Light Device.

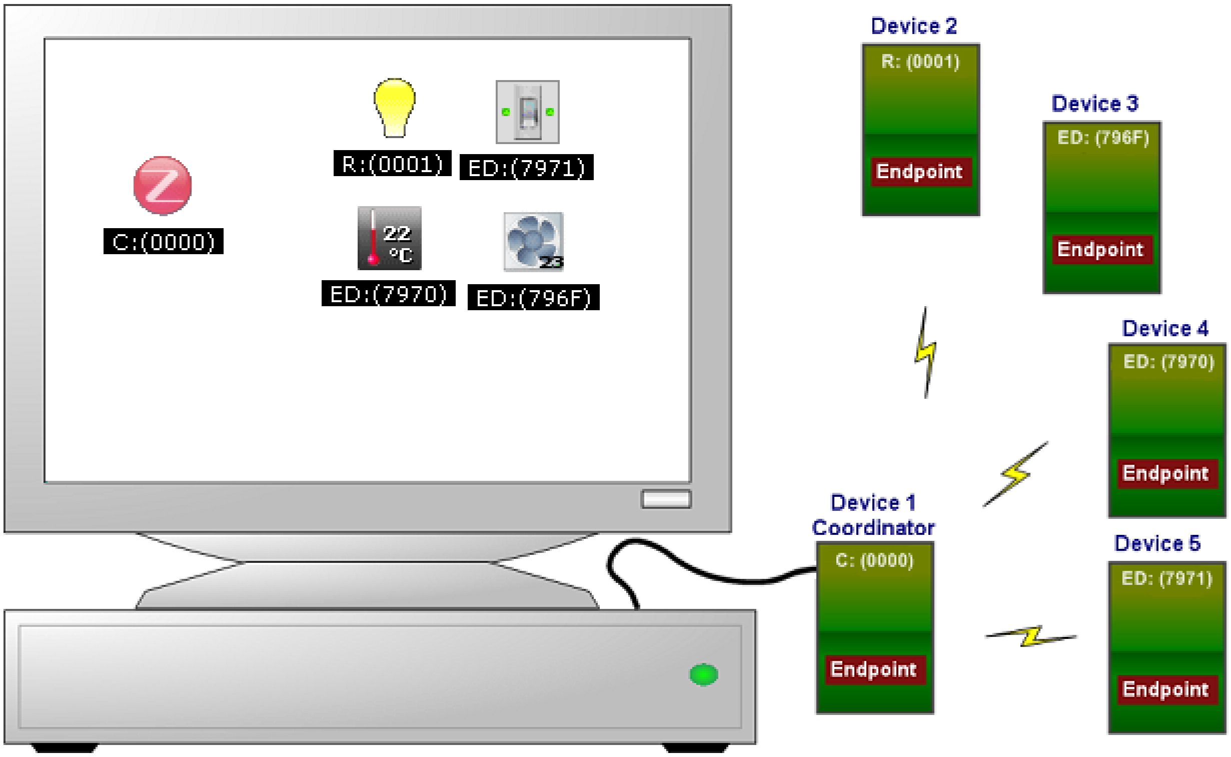
NOTE

All applications can be configured to run on the Network Controller Board (NCB) and Sensor Reference Board (SRB) (both part of the 1321x-EVK) as well as on the 1320x-QE128EVB development board.

1.3 Software Components

The software is divided into several embedded software components and a PC software component. The embedded components include all ZigBee functionality needed for the required ZeD features. The PC software component interacts with the coordinator through a set of commands. Likewise, the PC software component receives and handles events from the coordinator (see Figure 1-3).

Figure 1-3. PC and Embedded Components



1.4 Interface Overview

The embedded coordinator and PC software components each must support a ZigBee Test Client (ZTC) interface to operate with BeeStack and the Home Automation Applications. The physical transport layer for this protocol is a standard UART. The UART Interface supports both USB and RS232 connectors as the drivers for the physical layer are completely separated from the ZTC code itself in the embedded software component.

The ZTC interacts directly with the different layers in BeeStack. It mainly executes services from the ZDP layer, APS layer and NWK layer.

Refer to Chapter 4, “ZigBee Test Client (ZTC) Interface“ for details about the ZigBee Test Client (ZTC) interface.

Chapter 2
Getting Started With ZeD Embedded Applications

This chapter will help users get the embedded applications up and running as quickly as possible. This chapter contains information about the following:

- The source code package
- File structure
- How to build a project

2.1 ZeD Embedded Package Contents

The Freescale ZeD Embedded Source Code Package consists of this document supplemented by the following files:

1. Four (4) BeeKit solution files — Use these solution files to quickly and easily generate CodeWarrior projects for the embedded applications.

   Three (3) target oriented solution files. — The projects from each target oriented solution have the same hardware configuration (e.g. for an SRB target).

   One (1) EVK oriented solution file. — The projects in the EVK oriented solution have the same correspondence between the applications and their target hardware as do the preconfigured boards that come with the EVK.

   The application projects for the MC1321x-EVK are configured in BeeKit as follows:

   — Combined Interface Device is a coordinator application for NCB board

   — Dimmable Light Device and Thermostat Device are end device with RxOnWhenIdle applications for NCB boards

   — Dimmer Switch Device, Temperature Sensor Device and On/Off Switch Device are end device with RxOnWhenIdle applications for SRB boards

   — On/Off Light Device is a router application which runs on SRB board

   The application projects for the MC1322x-EVK are configured in BeeKit as follows:

   — Combined Interface Device is a coordinator application for SN board

   — Dimmable Light Device and Thermostat Device, 1st On Off/Light are router applications for NN boards

   — Dimmer Switch Device, Temperature Sensor Device are end device applications for SN boards with RxOnWhenIdle true

   — 2nd On/Off Light is a router application for SN board

   — 1st and 2nd On/Off Switch Devices are end device applications for LPN boards with Low Power Mode Enabled

   The solution files are located in the ZeD installation folder in the following directory structure:

```
..\Program Files\Freescale\ZeD 1.0.0\BeeKitSolutionFiles\ZigBeeHaApps_Mc1321xEVK.bksln
..\Program Files\Freescale\ZeD 1.0.0\BeeKitSolutionFiles\ZigBeeHaApps_Mc1322xEVK.bksln
..\Program Files\Freescale\ZeD 1.0.0\BeeKitSolutionFiles\ZigBeeHaApps_SRB.bksln
..\Program Files\Freescale\ZeD 1.0.0\BeeKitSolutionFiles\ZigBeeHaApps_NCB.bksln
..\Program Files\Freescale\ZeD 1.0.0\BeeKitSolutionFiles\ZigBeeHaApps_QE128EVB.bksln
```

2. All output files are for different targets (`.s19 & .bin` files). The output files are separated into distinct directories by function of the hardware target (SRB, NCB and 1320x-QE128EVB) or the MC1321x-EVK and MC1322x-EVK. They are located in the `Embedded` directory in the root of the ZeD installation directory. The EVK output files allow users to easily reprogram the boards to their initial applications.

NOTE

> The BeeKit Wireless Connectivity Toolkit software with a valid BeeStack license and CodeWarrior needs to be installed on the PC in order to generate the output files for the Home Automation Applications from the BeeKit solution files.

2.2 Starting the Applications

The ZeD Package is included with the MC1321x-EVK and MC1322x-EVK.

The boards contain preloaded software applications. In order to start the applications, perform the following tasks.

1. Attach power (or insert batteries) and turn on the boards.Only the coordinator board must be connected to the PC in order for the ZeD PC application to communicate with the Combined Interface Device Application. The Combined Interface Device Application is configured by default to use the USB connector for serial communication for the SN, NN, SRB, NCB, and 1320x-QE128EVB boards. Follow the instructions in the “Found New Hardware“ wizard on the PC. Choose “`Program Files/Freescale/Drivers`“ folder as the location for the drivers.

NOTE

> If using the onboard AA battery pack to power the SRB and the temperature sensor is being used, the readings will not be accurate. To obtain more accurate readings, power the SRB through its on board DC power connector.

2. To reset the boards, press the Reset switch on the SN, NN, SRB and NCB boards. Use the power switch on the 1320x-QE128EVB and LPN.
3. The ZigBee network must be formed by starting the coordinator board which runs the Combined Interface Application. The network is formed by pressing SW1 on the coordinator for a short period (less than a second). The network formation is complete when LED1 and LED2 on the board are on.
4. The other devices can join the network started by the coordinator by pressing SW1 on the boards for a short period. The devices will join only if the PAN ID and the channel are the same as those of the coordinator. Wait until each board joins the network before joining another one.
5. To see if an end device has joined the network verify that the LED1 is on.

6. To see if a router has joined the network verify that the LED 1 and LED 2 are on. The switches on the boards perform the following functions:

- Short press SW1 to start / join a network
- Short press SW 2 for toggling the permit joining of a device (not available on LPN)

  Permit join is ON: other devices can join to this device (LED2 is on)

  Permit join is OFF: no other devices can join the network on this device (LED2 is off)
- Short press SW 3 (SW 2 for LPN) to toggle binding between that device and another device on which SW3 (SW2 for LPN) was pressed.

  When the two devices have been bound, LED 3 (LED 3 for LPN) on both devices stops flashing.
- Short press SW4 short to change the channel
- Long press SW1 (more than a second) to change to the Application(Run) mode. In this mode, application specific functions are executed
- Long press SW2 for the device to leave the network
- Long press SW3 (Long press SW2 on LPN) to remove all bindings from the device.

For more information regarding the switches and the application specific code for application mode see Reference [5] or the code.

2.3 Building an Application

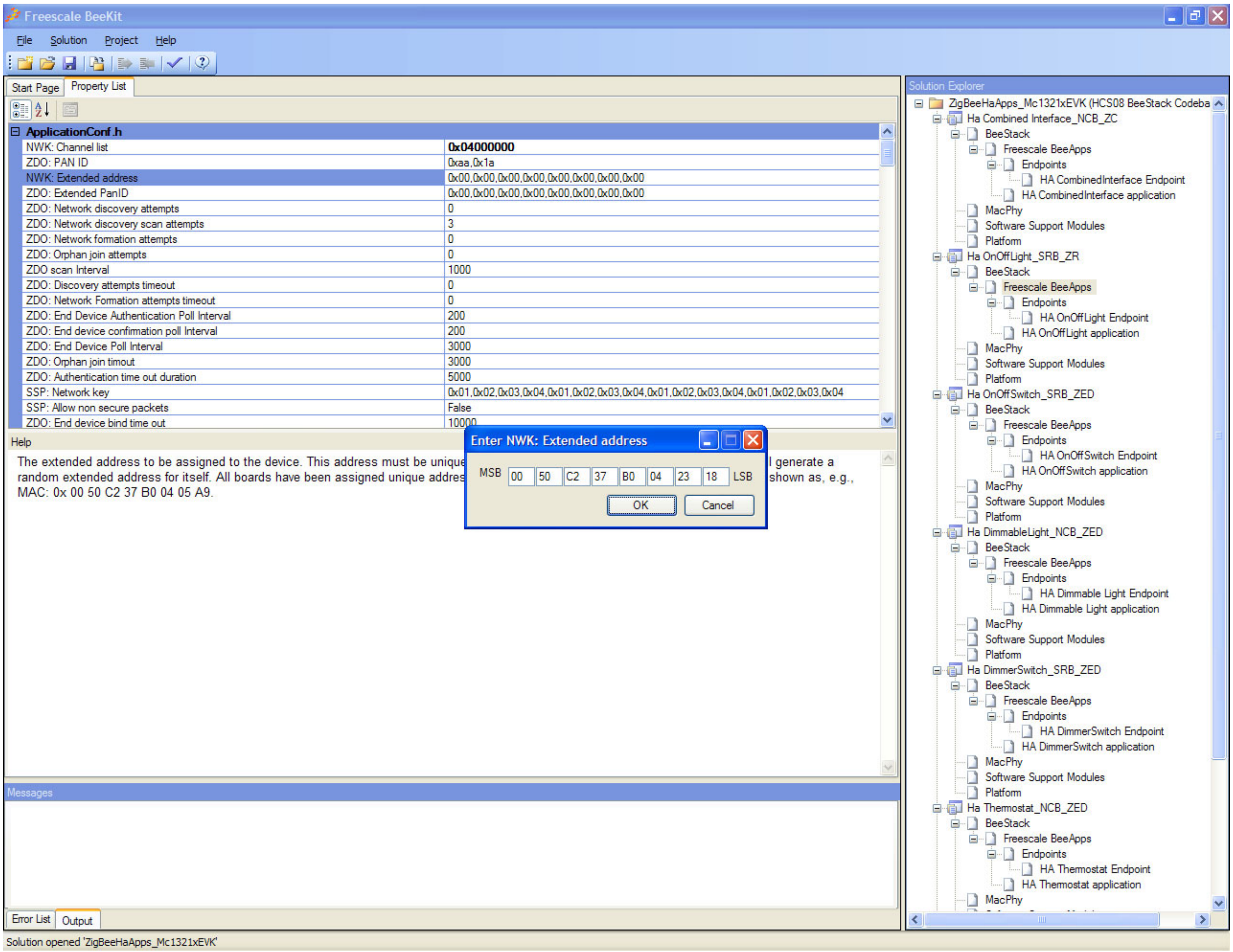
The steps for generating an output file (`.s19` file) from a BeeKit solution file are detailed in this section. This example is for the ZigBee 2006 MC1321x-EVK solutions. Steps for the MC1322x-EVK solutions are similar. The following example describes how to build the On/Off Light Device application, but it can serve as a template for other applications as well.

1. Start the BeeKit GUI.
2. Choose File->Select Codebase...
3. Select Codebase location: `..\Program Files\Freescale\Codebases\BeeKit BeeStack Codebase 1.0.4`
4. Choose File ->Open->Open Project/Solution.
5. Look in: `..\Program Files\Freescale\ZeD 1.1.0\BeeKitSolutionFiles\`

   File name: `ZigBeeHaApps_Mc1321xEVK.bksln`

   Files of type: BeeKit solution file (`*.bksln`)
6. Select Open.

7. Select the Freescale BeeApps of the Ha OnOffLight_SRB_ZR project and modify the NWK: Extended address property to contain the MAC device address. The MAC unique address is written on the label on the back of the SRB printed circuit board. For example for a MAC address equal to 0x 00 50 C2 37 B0 04 23 18 the value for this property must be set to as shown Figure 2-1.



ApplicationConf.h

NWK: Channel list	**0x04000000**
ZDO: PAN ID	0xaa,0x1a
NWK: Extended address	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
ZDO: Extended PanID	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
ZDO: Network discovery attempts	0
ZDO: Network discovery scan attempts	3
ZDO: Network formation attempts	0
ZDO: Orphan join attempts	0
ZDO scan Interval	1000
ZDO: Discovery attempts timeout	0
ZDO: Network Formation attempts timeout	0
ZDO: End Device Authentication Poll Interval	200
ZDO: End device confimation poll Interval	200
ZDO: End Device Poll Interval	3000
ZDO: Orphan join timout	3000
ZDO: Authentication time out duration	5000
SSP: Network key	0x01,0x02,0x03,0x04,0x01,0x02,0x03,0x04,0x01,0x02,0x03,0x04,0x01,0x02,0x03,0x04
SSP: Allow non secure packets	False
ZDO: End device bind time out	10000

Help

The extended address to be assigned to the device. This address must be unique l generate a random extended address for itself. All boards have been assigned unique addres shown as, e.g., MAC: 0x 00 50 C2 37 B0 04 05 A9.

Enter NWK: Extended address

MSB 00 50 C2 37 B0 04 23 18 LSB

Solution opened 'ZigBeeHaApps_Mc1321xEVK'

Figure 2-1. NWK: Extended Address Property

The value of this property as displayed in the `ApplicationConf.h` section of the Freescale BeeApps component is in reversed order.

8. Choose Solution->Export Solution

   Select Project(s): Ha OnOffLight_SRB_ZR

9. Select OK in order to export the project
10. Start the CodeWarrior Integrated Development Environment
11. Choose File->Import Project

12. Look in: `..\BeeKitSolutionFiles\Ha OnOffLight_SRB_ZR\`

    Object name: `Ha OnOffLight_SRB_ZR.xml`

    Object of: xml Files (`*.xml`)
13. Select Open
14. Save in: `..\BeeKitSolutionFiles\Ha OnOffLight_SRB_ZR\`

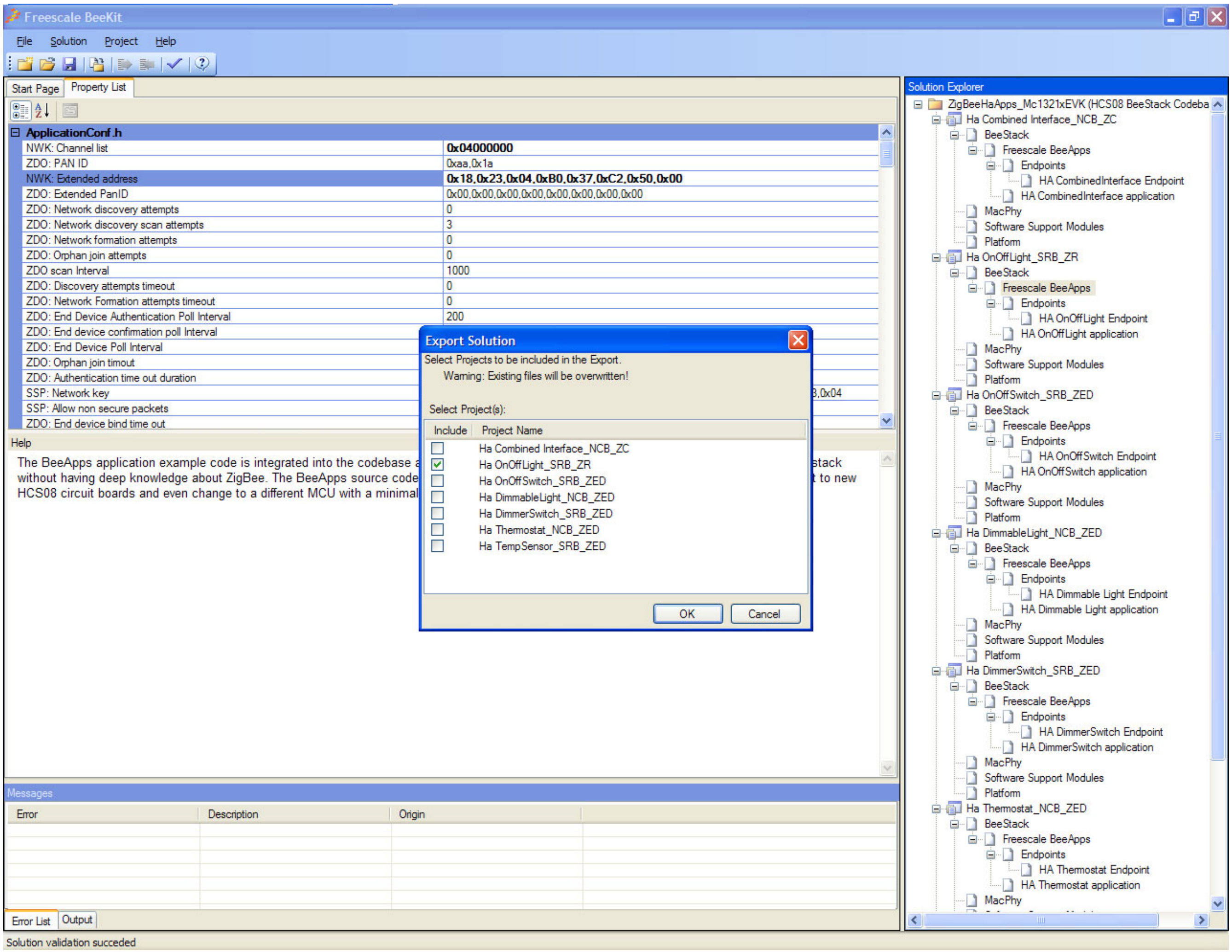
    File name: `Ha OnOffLight_SRB_ZR.mcp`

    Save as type: Project Files (`*.mcp`)
15. Select Save
16. The project is now loaded into the CodeWarrior IDE for compiling process.

NOTE

CodeWarrior Version 6.1 is required.

17. The steps for loading a code image into the Flash memory are described in the *ZigBee environment Demonstration Quick Start Guide*, Reference[4].



ApplicationConf.h	
NWK: Channel list	0x04000000
ZDO: PAN ID	0xaa,0x1a
NWK: Extended address	0x18,0x23,0x04,0xB0,0x37,0xC2,0x50,0x00
ZDO: Extended PanID	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
ZDO: Network discovery attempts	0
ZDO: Network discovery scan attempts	3
ZDO: Network formation attempts	0
ZDO: Orphan join attempts	0
ZDO scan Interval	1000
ZDO: Discovery attempts timeout	0
ZDO: Network Formation attempts timeout	0
ZDO: End Device Authentication Poll Interval	200
ZDO: End device confirmation poll Interval	
ZDO: End Device Poll Interval	
ZDO: Orphan join timout	
ZDO: Authentication time out duration	
SSP: Network key	3,0x04
SSP: Allow non secure packets	
ZDO: End device bind time out	

The BeeApps application example code is integrated into the codebase a … stack without having deep knowledge about ZigBee. The BeeApps source code … t to new HCS08 circuit boards and even change to a different MCU with a minimal

Select Projects to be included in the Export.
Warning: Existing files will be overwritten!

Select Project(s):

Include	Project Name
☐	Ha Combined Interface_NCB_ZC
☑	Ha OnOffLight_SRB_ZR
☐	Ha OnOffSwitch_SRB_ZED
☐	Ha DimmableLight_NCB_ZED
☐	Ha DimmerSwitch_SRB_ZED
☐	Ha Thermostat_NCB_ZED
☐	Ha TempSensor_SRB_ZED

Error	Description	Origin

Solution validation succeded

```
ZigBeeHaApps_Mc1321xEVK (HCS08 BeeStack Codeba
  Ha Combined Interface_NCB_ZC
    BeeStack
      Freescale BeeApps
        Endpoints
          HA CombinedInterface Endpoint
        HA CombinedInterface application
    MacPhy
    Software Support Modules
    Platform
  Ha OnOffLight_SRB_ZR
    BeeStack
      Freescale BeeApps
        Endpoints
          HA OnOffLight Endpoint
        HA OnOffLight application
    MacPhy
    Software Support Modules
    Platform
  Ha OnOffSwitch_SRB_ZED
    BeeStack
      Freescale BeeApps
        Endpoints
          HA OnOffSwitch Endpoint
        HA OnOffSwitch application
    MacPhy
    Software Support Modules
    Platform
  Ha DimmableLight_NCB_ZED
    BeeStack
      Freescale BeeApps
        Endpoints
          HA Dimmable Light Endpoint
        HA Dimmable Light application
    MacPhy
    Software Support Modules
    Platform
  Ha DimmerSwitch_SRB_ZED
    BeeStack
      Freescale BeeApps
        Endpoints
          HA DimmerSwitch Endpoint
        HA DimmerSwitch application
    MacPhy
    Software Support Modules
    Platform
  Ha Thermostat_NCB_ZED
    BeeStack
      Freescale BeeApps
        Endpoints
          HA Thermostat Endpoint
        HA Thermostat application
    MacPhy
```

Figure 2-2. ZeD ZigBeeHaApps_MC1321xEVK BeeKit Solution

2.4 Software Component File Structure

The application project file structure is obtained after exporting a project file from BeeKit. All projects have the same directory structure. The files that differ between projects are the ones containing the application specific data (e.g. `BeeApps.c`). These files are located in the root of the `BeeApps` directory. The `BeeApps` directory also contains the files that are Home Automation specific, the ZCL files, the configuration and the interface files between BeeStack and the application.

The other directories are:

- `MacPhy` — MAC specific files.
- `PLM` — Platform specific files.
- `SSM` — ZTC and Task Scheduler files.
- `ZigBee` — BeeStack Libraries and interface files.

Chapter 3
ZeD Embedded Projects Configuration

This chapter describes the basic compile-time configuration of the embedded software component. Every embedded application is configured by setting properties from BeeKit to a specific value.

3.1 Configuration

All ZeD embedded applications are based on the standard BeeKit Home Automation applications. In addition to the default settings, the properties as shown in the following sections must also be set the same way in BeeKit to obtain the required functionality for ZeD.

3.1.1 ZeD Coordinator Specific Properties

The ZTC Interface must be enabled on the ZeD coordinator. This feature is essential for ZeD in order for the embedded application to communicate with the PC application. The ZTC is enabled by setting the `ZTC:ZTC enabled` property from the Software Support Modules component to True. For the coordinator, only the following SAPs must be enabled in order to receive / transmit commands from / to the ZeD PC interface (all other SAPs must be disabled):

- ZDP SAP (enabled by setting the `ZTC: ZDP SAP Capability` property of the Software Support Modules component to True)
- APSDE SAP (enabled by setting the `ZTC: APSDE SAP Capability` property of the Software Support Modules component to True)
- APSME SAP (enabled by setting the `ZTC: APSME SAP Capability` property of the Software Support Modules component to True)
- NLME SAP (enabled by setting the `ZTC: NLME SAP Capability` property of the Software Support Modules component to True)

Large data packets are exchanged between the ZeD Coordinator and ZeD PC interface. To ensure a reliable communication, the hardware flow control is enabled both on PC application side and on embedded side. For the Combined Interface device this property is enabled by setting both `UART 1 Hardware flow control enabled` and `UART 2 Hardware flow control enabled` properties from the Platform component to True.

As a consequence of using hardware flow control the `UART RX Flow Control Skew` and `UART RX Flow Control Resume` properties from the Platform component must be set to 10.

In order to display the correct characteristics (active endpoints, simple descriptor, node descriptor, power descriptor, bind table and routing table) of each device on the ZeD GUI, the coordinator issues the following requests which are enabled in BeeKit:

- Active Endpoint request (enabled by setting the `ZDP: Enable Active_EP_req` property of the Freescale BeeApps component to True)

- Simple Descriptor request (enabled by setting the `ZDP: Enable Simple_Desc_req` property of the Freescale BeeApps component to True)
- Node Descriptor request (enabled by setting the `ZDP: Enable Node_Desc_req` property of the Freescale BeeApps component to True)
- Power Descriptor request (enabled by setting the `ZDP: Enable Power_Desc_req` property of the Freescale BeeApps component to True)
- Management Bind request (enabled by setting the `ZDP MGMT: Enable Mgmt_Bind_req` property of the Freescale BeeApps component to True)
- Management Routing request (enabled by setting the `ZDP MGMT: Enable Mgmt_Rtg_req` property of the Freescale BeeApps component to True).

3.1.2 General ZeD Applications Properties

The properties presented below are applicable for all ZeD applications including the ZeD coordinator.

To control which target to use (for example, 1321x-SRB) set the `Target Hardware` property from the Platform component to the desired value (for example, 1321x-SRB).

To have a unique MAC address for each device, set the `NWK: Extended address` property from the Freescale BeeApps component to the value written on the label on the back of the printed circuit board. For more details on this property see Section 2.3, “Building an Application.

The following additional features must be set in order for the ZeD system to run correctly:

- The default communication channel is 26. In order to change the channel value, change the `NWK: Channel list` property of the Freescale BeeApps component to the needed value.
- All the devices in the network must have MGMT_Bind response enabled in order to respond to the MGMT_Bind request. By doing this, the PC application is informed about the bindings in the network. For enabling MGMT_Bind response, the `ZDP MGMT: Enable Mgmt_Bind_rsp` property of the Freescale BeeApps component must be set to True.
- All the devices in the network must have MGMT_Rtg response enabled in order to respond to the MGMT_Rtg request. For enabling MGMT_Rtg response, the `ZDP MGMT: Enable Mgmt_Rtg_rsp` property of the Freescale BeeApps component must be set to True.
- The `ZDP: enable End_Device_Annce` property from the Freescale BeeApps component is enabled by default for all devices.
- The `NWK: Maximum Broadcast Transaction Table Entries` property from the Freescale BeeApps component must be set to 10 for all devices.

3.1.3 MC1321x-EVK ZeD Applications Specific Properties

The following features must be set for the ZeD system running on the boards in the MC1321x-EVK:

- Dimmable Light Device, Thermostat Device, On / Off Switch Device, Dimmer Switch Device and Temperature Sensor Device are end devices. To enable this feature, set the `Device Type` property of the MacPhy component to End Device
- On/Off Light Device is a router application. To enable this feature, set the `Device Type` property of the MacPhy component to Router
- On / Off Switch Device, Dimmer Switch Device, Temperature Sensor Device, Dimmable Light Device and Thermostat Device have the MAC Rx On When Idle capability enabled. To enable this feature, set the `MAC: Mac Capability: Rx On When Idle` property of the Freescale BeeApps component to True
- The `ZDO: Maximum number of Poll failure before rejoin` property from the Freescale BeeApps component must be set to 5 for all 5 end devices.
- Thermostat Device has the attributes reporting capability enabled. To enable this feature, set the `HA: Enable Reporting of attributes` property of the Freescale BeeApps component to True

3.1.4 1320x-QE128EVB, MC1322x ZeD Applications Specific Properties

ZeD 1.1.0 and later supports using the ZeD embedded applications on 1320x-QE128EVB development boards. ZeD 1.2.0 and later supports using the ZeD embedded applications on MC1322x development boards. The applications for this type of boards has a specific configuration as follows:

- Dimmable Light Device, Thermostat Device and On/Off Light Device are ZigBee routers. To enable this feature, the `Device Type` property in the MacPhy component is set to Router.
- On / Off Switch Device, Dimmer Switch Device and Temperature Sensor Device are end devices. To enable this feature, the `Device Type` property in the MacPhy component is set to End Device
- On / Off Switch Device, Dimmer Switch Device, Temperature Sensor Device have the MAC Rx On When Idle capability enabled. To enable this feature, set the `MAC: Mac Capability: Rx On When Idle` property of the Freescale BeeApps component to True
- The ZDO: Maximum number of Poll failure before rejoin property from the Freescale BeeApps component must be set to five (5) for all three end devices.
- Thermostat Device has the attributes reporting capability enabled. To enable this feature, set the HA: Enable Reporting of attributes property of the Freescale BeeApps component to True

Chapter 4
ZigBee Test Client (ZTC) Interface

Figure 4-1 shows the data format used for communication between the upper (host) and lower (embedded) software. It consists of a data field, augmented with a header that contains an opcode and length field. The same format is used in both directions.

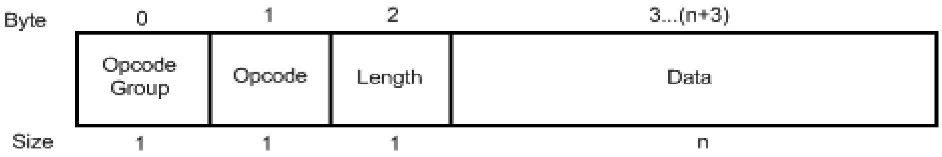


Figure 4-1. Data Format (Upper Host and Lower Embedded Software)

- Byte 0 — Opcode group
- Byte 1 — Opcode
- Byte 2 — Length of data field (excluding header)
- Byte 3 to (n+3) — Data, including timestamps where applicable

Figure 4-2 shows that the UART driver modules append sync word and error checking to the generic frame format.

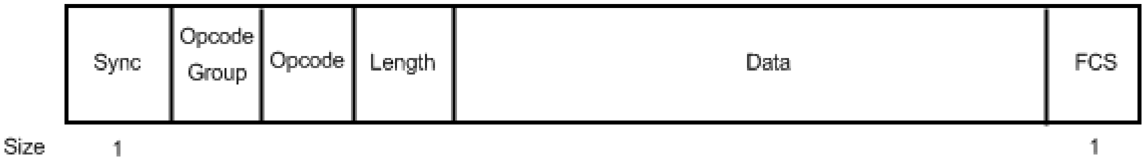


Figure 4-2. Generic Frame Format with Sync Word and Error Checking Appended

- Sync byte — Frame delimiter indicating packet start. Constant value = STX = 0x02.
- FCS byte — Frame checksum calculated as the XOR of all bytes in the opcode group, opcode, length and data fields.

4.1 ZTC Error Codes

The received command packets are verified in order to detect different errors. If an error is detected and the application is configured to report errors (`ZTC: Error reporting Capability` property from the Software Support Modules component set to TRUE), a message containing the error code is returned. Table 4-1 shows the possible error codes that can be returned.

Table 4-1. Error Codes(Status) Returned

Value	Code
0xF2	Unknown PIB
0xF7	Unknown opcode group
0xF8	Opcode group disabled (SAP disabled)
0xFA	Read only network information base identifier
0xFC	ZTC register end point messages not supported
0xFE	Message too big

Most messages passed to the embedded system will trigger a response message.

Also the response packets that are to be sent from the embedded side to the host side are verified for errors. If an error is detected and the application is configured to report errors, a message containing the error code is returned. Table 4-2 shows the possible error codes that can be returned.

Table 4-2. Error Codes(Status) Returned

Value	Code
0xF3	Application message too big
0xF4	Out of memory for message allocation
0xF5	Endpoint table is full
0xF6	Endpoint not found
0xFB	Unknown network information base identifier
0xFD	Unknown opcode

4.2 Commands

The ZTC commands are implemented over the BeeStack APS, ZDP, and NWK layers and over services from the Home Automation Profile. Normally, all the commands trigger a primitive to be sent on the over the air interface. The response from the remote device is sent back to the initiating device, which, in the ZeD system, sends this response to the PC application. For details on the Status return parameter or any other parameter from the tables shown in this section, see the ZigBee 2006 documentation, Home Automation Profile documentation, or source code. The following sections describe these commands and responses.

4.2.1 NWK Commands

The NLME_Get request is the only command used by ZeD to read the value of any network information base identifier on the coordinator. Table 4-3 shows how the NLME_Get request command is specified.

Table 4-3. NLME_Get Request Command

Byte	Parameter	Description
0	OPG	Opcode Group is 0xA3
1	OPC	Opcode is 0x22
2	Length	Total length
3	iID	Attribute Identifier
4	iIndex	Not used
5	iEntries	Not used
6	iEntrySize	Not used

Table 4-4 shows the NLME_Get confirm command, the response event for NLME_Get request.

Table 4-4. NLME_Get Confirm Event

Byte	Parameter	Description
0	OPG	Opcode Group is 0xA4
1	OPC	Opcode is 0x22
2	Length	Total length
3	Status	status
4	iID	Attribute Identifier
5	iIndex	Not used
6	IEntries	Not used
7	iEntrySize	Not used
8-9	DataLength	The size in bytes of the value of the iID
10-11	Data	The value of the specified attribute

4.2.2 APS Commands

Table 4-5 shows how the APS commands used by ZeD are specified.

Table 4-5. APS Commands

APS Command	Details
APSME_Bind Request	Table 4-6
APSME_UnBind Request	Table 4-7

The APSME_Bind request is used locally on the coordinator to bind it to another endpoint or a group. The endpoint (DstEndPoint) is located on the device which has DstAddr as its address.

Table 4-6. APSME_Bind Request

Byte	Parameter	Description
0	OPG	Opcode Group is 0x99
1	OPC	Opcode is 0x00
2	Length	Total length
3-10	SrcAddr	Source IEEE address
11	SrcEndPoint	The Source End Point for binding entry
12-13	ClusterID	The identifier of the cluster on the source device
14	DstAddrMode	Address mode of the destination device
15-22	DstAddr	Destination address / Group Id
23	DstEndPoint	The Destination end Point for binding entry

The APSME_UnBind request is used locally on the coordinator to unbind it from an endpoint or a group. The endpoint (DstEndPoint) is located on the device which has DstAddr as its address as shown in Table 4-7.

Table 4-7. APSME_UnBind Request

Byte	Parameter	Description
0	OPG	Opcode Group is 0x99
1	OPC	Opcode is 0x09
2	Length	Total length
3-10	SrcAddr	Source IEEE address
11	SrcEndPoint	End point on the source device
12-13	ClusterID	Cluster ID
14	DstAddr Mode	Address mode of the destination device
15-22	DstAddr	Destination address / Group Id
23	DstEndPoint	End point on the destination device

The APS commands will trigger the following response events. The events are specified as shown in Table 4-8 and Table 4-9.

Table 4-8. APSME_Bind Confirm Event

Byte	Parameter	Description
0	OPG	Opcode Group is 0x98
1	OPC	Opcode is 0x07
2	Length	Total length

Table 4-8. APSME_Bind Confirm Event (continued)

Byte	Parameter	Description
3	Status	status
4-11	SrcAddr	Source IEEE address
12	SrcEndPoint	End point on the source device
13-14	ClusterID	Cluster ID
15	DstAddr Mode	Address mode of the destination device
16-23	DstAddr	Destination address / Group Id
24	DstEndPoint	End point on the destination device

Table 4-9. APSME_UnBind Confirm Event

Byte	Parameter	Description
0	OPG	Opcode Group is 0x98
1	OPC	Opcode is 0x08
2	Length	Total length
3	Status	status
4-11	SrcAddr	Source IEEE address
12	SrcEndPoint	End point on the source device
13-14	ClusterID	Cluster ID
15	DstAddr Mode	Address mode of the destination device
16-23	DstAddr	Destination address / Group Id
24	DstEndPoint	End point on the destination device

The APSDE_Data Indication event (Table 4-10) retrieves responses for Home Automation commands. The asdu parameter contains the Home Automation response.

Table 4-10. APSDE_Data Indication Event

Size in bytes[1]	Parameter	Description
1	OPG	Opcode Group is 0x9D
1	OPC	Opcode is 0x01
1	Length	Total length
1	DstAddrMode	Address mode of the destination device
2 / 8[2]	DstAddr	Destination address
1	DstEndPoint	End point on the destination device
1	SrcAddrMode	Address mode of the source device
2 / 8	SrcAddr	Source address

Table 4-10. APSDE_Data Indication Event (continued)

Size in bytes[1]	Parameter	Description
1	SrcEndPoint	End point on the source device
2	ProfileID	Profile ID
2	ClusterID	Cluster ID
1	asduLength	The size in bytes of ASDU
variable	asdu	ASDU
1	WasBroadcast	TRUE if the transmission was a broadcast, FALSE otherwise
1	SecurityStatus	Security Status
1	LinkQuality	Link quality of the received packet

[1] Position of the bytes is variable.

[2] The number of bytes for destination / source address depends of the addressing mode.

4.2.3 ZDP Commands

Table 4-11 shows how the ZDP commands used in ZeD are specified.

Table 4-11. ZDP Commands

Command	OPG	OPC	Length	Param 1	Param 2	Param 3	Param 4
IEEE_Addr_Request	0xA2	0x01	0x06	Destination ShortAddr[2]	Network address[2]	Request Type	Start Index
Node_Desc_Request	0xA2	0x02	0x04	Destination ShortAddr[2]	Network address of interest[2]		
Simple_Desc_request	0xA2	0x04	0x05	Destination ShortAddr[2]	Network address[2]	Endpoint	
Power_Desc_Request	0xA2	0x03	0x04	Destination ShortAddr[2]	Network address of interest[2]		
Active_EP_Request	0xA2	0x05	0x04	Destination ShortAddr[2]	Network address[2]		
Mgmt_RTG_Request	0xA2	0x32	0x03	Destination ShortAddr[2]	Start Index		
Mgmt_Bind_Request	0xA2	0x33	0x03	Destination ShortAddr[2]	Start Index		
ZDP_Bind_Request	See Table 4-12						
ZDP_UnBind_Request	See Table 4-13						

Table 4-12. ZDP_Bind Request Command

Byte	Parameter	Description
0	OPG	Opcode group is 0xA2
1	OPC	Opcode is 0x21
2	Length	Total length
3-4	DestAddress	Short address of the device which receives this request
5-12	SrcAddr	Source IEEE address for binding
13	SrcEndPoint	End point on the source device for binding
14-15	ClusterID	Cluster ID
16	DstAddr Mode	Address mode of the destination device for binding
17-24	DstAddr	Destination address / Group Id for binding
25	DstEndPoint	End point on the destination device for binding

Table 4-13. ZDP_UnBind Request Command

Byte	Parameter	Description
0	OPG	Opcode group is 0xA2
1	OPC	Opcode is 0x22
2	Length	Total length
3-4	DestAddress	Short address of the device which receives this request
5-12	SrcAddr	Source IEEE address for binding
13	SrcEndPoint	End point on the source device for binding
14-15	ClusterID	Cluster ID
16	DstAddr Mode	Address mode of the destination device for binding
17-24	DstAddr	Destination address / Group Id for binding
25	DstEndPoint	End point on the destination device for binding

- The IEEE_Addr_Request, Node_Desc_Request, Simple_Desc_request, Power_Desc_Request and Active_EP_Request are requests made by ZeD to a remote device in order to retrieve the remote device’s IEEE address, node descriptor, simple descriptor, power descriptor or the number of active endpoints.
- The Mgmt_RTG_Request is used by ZeD for retrieving the all the entries from the routing table from a remote device which has Destination ShortAddr as address.
- The Mgmt_Bind_Request is used by ZeD to get the binding table of a remote device having Destination ShortAddr as its address.
- The ZDP_Bind_Request request is sent by ZeD for binding a remote device to another endpoint or group. The endpoint (DstEndPoint) is located on the device which has DstAddr as its address. The short address of the remote device is DestAddress.

- The ZDP_UnBind_Request request is sent by ZeD for unbinding a remote device from an endpoint or group. The endpoint (DstEndPoint) is located on the device which has DstAddr as its address. The short address of the remote device is DestAddress.

4.2.4 ZDP Responses

When a ZDP command is sent to a remote device, a response is sent back over the air interface to the initiating device. The responses are copied onto the UART interface and sent to the PC application. The following sections describe the responses that can be received by the ZeD system.

4.2.4.1 IEEE Address Response

Table 4-14 shows the response a device sends on an IEEE Address request.

Table 4-14. IEEE Address Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0x81
2	Length	Total length
3	Status	status
4-11	IEEEAddrRemoteDev	IEEE address for the remote device
12-13	NWKAddrRemoteDev	Network address for the remote device
14	NumOfAssociatedDevice	Number of associated devices
15	StartIndex	Start index
16-lastbyte[1]	ListOfShortAddresses	A list of short addresses, one corresponding to each associated device.

[1] The number of bytes varies in function of the number of associated devices.

4.2.4.2 Node Descriptor Response

Table 4-15 shows the response a device sends on a node descriptor request.

Table 4-15. Node Descriptor Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0x82
2	Length	Total length
3	Status	status
4-5	Network address of interest	Network address of interest
6	NODE-DESCRIPTOR_FIRST_BYTEr	This byte is composed by: Logical byte(3 bits), Complex descriptor avalable(1 bit), User descriptor avalable(1 bit), Reserved(3 bits)
7	NODE-DESCRIPTOR_SECOND_BYTE	This byte is composed by: APS flags(3 bits), Frequency band(5 bits)
8	NODE-DESCRIPTOR_MACcapabilityFlags_BYTE	Alternate PAN Coordinator, Device Type, Power Source, Rx on when idle, Reserved, Security capability, Allocate address
9-10	NODE-DESCRIPTOR_ManufacturerCode_BYTES	Manufacturer code allocated by ZigBee Alliance relating the manufacturer to the device
11	NODE-DESCRIPTOR_MaximumBufferSize_BYTE	Maximum value 0x7F, represents the size of data passed to of from the application and APS before fragmentation
12-13	NODE-DESCRIPTOR_MaximumTransferSize_BYTES	Maximum value 0x7FFF, maximum size in octets that can be transferred to of from this node in one single message transfer
14-15	NODE-DESCRIPTOR_ServerMask_BYTES	PrimaryTrustCenter(1bit), BckupTrustCenter(1bit), PrimaryBindTableCenter(1bit), BackupBindTableCenter(1bit), PrimaryDiscoveryCache(1bit), BackupDiscoveryCache(1bit), Reserved(10bits)

4.2.4.3 Power Descriptor Response

Table 4-16 shows the response that a device sends on a power descriptor request.

Table 4-16. Power Descriptor Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0x83
2	Length	Total length is 5
3	Status	status
4-5	Network address of interest	Network address of interest
6-7	Power descriptor	Power descriptor

4.2.4.4 Number of Active Endpoints Response

Table 4-17 shows the response to a number of active endpoints request to a network device.

Table 4-17. Active Endpoints Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0x85
2	Length	Total length
3	Status	status
4-5	NWKAddress	Network address
6	ActiveEPCount	The count of active endpoints on the remote device
7-lastbyte[1]	ActiveEPList	List of bytes each of which represents an 8 bit endpoint.

[1] The number of bytes varies in function of the number of active endpoints.

4.2.4.5 Simple Descriptor Response

Table 4-18 shows the response of a simple descriptor request to a ZigBee device within the network.

Table 4-18. Simple Descriptor Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0x84
2	Length	Total length
3	Status	status
4-5	Network address of interest	Network address of interest
6	Length	Size in bytes for the simple descriptor + size of Length
7	Endpoint	Endpoint
8-9	Application Profile ID	Application Profile ID
10-11	Application Device ID	Application Device ID
12	ApplicationDeviceAndVersionFlags	ApplicationDeviceAndVersionFlag
13	ApplicationNumberInClusters	Application number of input clusters
14-variablebyte[1]	ApplicationNumberInClustersList	List of input clusters
variablebyte+1	ApplicationNumberOutClusters	Application number of output clusters
variable	ApplicationNumberOutClustersList	List of output clusters

[1] The number of bytes varies in function of the number of clusters.

4.2.4.6 MGMT_RTG Response

Table 4-19 shows the response of a MGMT_RTG request to a ZigBee device within the network.

Table 4-19. Simple Descriptor Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0xB2
2	Length	Total length
3	Status	status
4	RoutingTableEntries	Number of routing table entries within the remote device
5	StartIndex	Start index
6	RoutingTableListCount	Number of Routing Table entries included within the RoutingTableList
7-lastbyte[1]	RoutingTableList	List of elements from Routing Table

[1] The number of bytes varies in function of the number of entries in the Routing Table.

4.2.4.7 MGMT_Bind Response

Table 4-20 shows the response of a MGMT_Bind request to a ZigBee device within the network.

Table 4-20. Simple Descriptor Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0xB3
2	Length	Total length
3	Status	status
4	BindingTableEntries	Number of binding table entries
5	StartIndex	Start index
6	BindingTableListCount	Number of Binding Table entries included within the BindingTableList
7-lastbyte[1]	BindingTableList	List of elements from Binding Table

[1] The number of bytes varies in function of the number of entries in the Routing Table.

4.2.4.8 ZDP_Bind Response

Table 4-21 shows the response of a ZDP_Bind request to a ZigBee device within the network.

Table 4-21. ZDP_Bind Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0xA1
2	Length	Total length
3	Status	status

4.2.4.9 ZDP_UnBind Response

Table 4-22 shows the response of a ZDP_UnBind request to a ZigBee device within the network.

Table 4-22. ZDP_UnBind Response

Byte	Parameter	Description
0	OPG	Opcode group is 0xA0
1	OPC	Opcode is 0xA2
2	Length	Total length
3	Status	status

4.2.5 Home Automation Commands

One of the benefits of using the ZigBee Cluster Library (ZCL) is the reporting of the cluster's attributes. However, only those attributes designated as reportable can be reported. So, a device can be configured to report its attributes periodically or on an attribute change event. If an attribute is “discrete”, it can be configured to send a reporting command when its value changes from one state to another. For an analog attribute, the reporting command can be sent when the attribute value changes with a value specified in a configure reporting command. A critical condition for a device to receive the reported attributes is to be bound with the device which reports them. Using this command, the ZeD PC application can be informed of the value of some attributes. For determining the initial values of the attributes, the PC application uses the Read Attribute command. This command is used when a device joins a network or when a user manually refreshes ZeD. At the same time, the Configure Reporting command is sent.

In the following subsections, a ZclZtc table refers to the commands and/or responses/reports exchanged between the ZeD PC software and the Combined Interface device. The Combined Interface device interprets the command, forms the ZCL frame and sends it over the air to the destination device using the APSDE_Data request primitive. Also, the responses and reports coming from the devices in the network are sent to ZeD PC software using the ZigBee Test Client (ZTC). The PC software interprets the responses and the reports by reading the payload from APSDE_Data indication and takes actions changing the graphical elements on the GUI. The ZclZtc response and report frames are in fact the ZCL response and report frames (refer to the following responses and reports tables).

All ZclZtc commands frames coming from the ZeD PC software contain parameters needed to be recognized by the embedded side and to form the APSDE_Data request primitive. Such parameters are: command opcode group (OPG), command opcode (OPC), address mode for the destination device (DestAddressMode), address for the destination device (DestAddress), end point on the destination device (DestEndPoint), transmit options (TxOptions), radius of transmission (Radius). The last fields in a ZclZtc command frame are specific to the ZCL command frame and are used by the Combined Interface device to form the over the air (OTA) ZCL frame that is included in the payload of APSDE_Data request primitive. The ZCL command frame is formed by adding the ZCL header to parameters or information provided by specific fields from ZclZtc command frames.

NOTE

The description of the over the air (OTA) ZCL frames is out of the scope of this chapter. The ZeD PC software transmits the parameters and information to the Combined Interface device to form the over the air ZCL frames that are sent to a destination device. Every response returns an error code included in the status field which is interpreted by the ZeD PC software.

For more details about the over the air ZigBee Cluster Library (ZCL) frames or errors code refer to References [3] and [4].

4.2.5.1 ZclZtc-Read Attributes Command

The ZclZtc-Read attributes command is generated when the ZeD PC software tries to determine the values of one or more attributes located on a device from network. Each attribute identifier field contains the identifier of the attribute to be read. Not all attributes can be read, only those designated as readable. The ZclZtc-Read attributes command format is shown in Table 4-23.

Table 4-23. ZclZtc-Read Attributes Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x00
	2	DestAddressMode	Address mode for the destination device (must be 0x02, only shortAddr is currently used)
	3-10	DestAddress	Address for the destination device (shortAddr)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17	Count	Number of attributes to be read
	18-XX	AttributesList	List of attributes IDs to be read (variable length); an attributeID has 2 bytes

4.2.5.2 ZclZtc-Read Attributes Response Command

The ZclZtc-Read attributes response command is generated in response to a ZclZtc-Read attributes command. The command frame contains a read attribute status record for each attribute identifier specified in the original ZclZtc-Read attributes command. The ZclZtc-Read attributes response command format is shown in Table 4-24.

Table 4-24. ZclZtc-Read Attribute Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x08 (direction response)[1].
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x01(read attribute response command)
Read attribute status record 1	attrId	2 byte	Attribute ID
	status	1 byte	Status of the read operation on this attribute (can be SUCCESS(0x00) if the operation was successful, else an error code)
	attrType	1 byte	Data type of the attribute in the same read attributes status record (is included if the associated status field contains a value of SUCCES(0x00))
	aData	variable length	Contain the current value of this attribute (is included if the associated status field contains a value of SUCCES)
...	...	...	...
Read attribute status record n[2]	attrId	2 byte	Attribute ID
	status	1 byte	Status of the read operation on this attribute (can be SUCCESS if the operation was successful, else an error code)
	attrType	1 byte	Data type of the attribute in the same read attributes status record (is included if the associated status field contains a value of SUCCES)
	aData	variable length	Contain the current value of this attribute (is included if the associated status field contains a value of SUCCES)

[1] For more details refer to Reference [3]

[2] n - is the last read attribute status record

4.2.5.3 ZclZtc-Write Attributes Command

The ZclZtc-Write attributes command is generated when the ZeD PC software wishes to change the values of one or more attributes located on a device from the network. Each write attribute record contains the identifier and the actual value of the attribute to be written. Not all attributes can be written, only the writable ones. The ZclZtc-Write attributes command format is illustrated in Table 4-25.

Table 4-25. ZclZtc-Write Attributes Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x02
	2	DestAddressMode	Address mode for the destination device (must be 0x02, only shortAddr is used)
	3-10	DestAddress	Address for the destination device (shortAddr)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17	ReqType	Request type: 0x02 -write, 0x03 - write undivided, 0x05- write with no response
	18	Count	Number of attribute records to be written
	19-20	AttributeID	Attribute ID 1
	21	AttributeDataType 1	Attribute Data Type 1
	22	attrsize 1	Size of the following data field
	23-XX	Data 1	Data to be written (variable length)
	...	...	...
	...	AttributeID n[1]	Attribute ID n
		AttributeDataType n	Attribute Data Type n
	...	attrsize n	Size of the following data field
	...	Data n	Data to be written (variable length)

[1] n - is the last attribute record to be written

4.2.5.4 ZclZtc-Write Attributes Response Command

The ZclZtc-Write attributes response command is generated in response to a ZclZtc-Write attributes command and using the ZTC, the ZeD PC software interprets it from APSDE-DATA.indication payload.The ZclZtc-Write attributes response command format is illustrated in Table 4-26.

Table 4-26. ZclZtc-Write Attributes Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x08 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x04 (write attribute response command)
Attribute status record 1	status	1 byte	Status of the write attribute operation on this attribute (can be SUCCESS if the operation was successful and is not included in order to save bandwidth, else an error code)
	attrId	2 byte	Attribute ID (is not included for successfully written attributes)
...	...	...	...
Attribute status record n[2]	status	1 byte	Status of the write attribute operation on this attribute (can be SUCCESS if the operation was successful and is not included in order to save bandwidth, else an error code)
	attrId	2 byte	Attribute ID (is not included for successfully written attributes)

[1] For more details refer to Reference [3]

[2] n - is the last attribute status record

4.2.5.5 ZclZtc-Configure Reporting Command

The ZclZtc-Configure Reporting command is generated when the ZeD PC software wishes to configure a device to automatically report the values of one or more of its attributes. The cluster definitions for that device specify which attributes are available for reporting. The ZclZtc-Configure Reporting command format is illustrated in Table 4-27.

Table 4-27. ZclZtc-Configure Reporting Command (To Server)

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x06
	2	DestAddressMode	Address mode for the destination device (must be 0x02, only shortAddr is used)
	3-10	DestAddress	Address for the destination device (shortAddr)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17	Count	Number of attribute reporting configuration records
	18	Direction 1	0x00 - sent to server
	19-20	AttributeID 1	Attribute ID 1
	21	AttributeDataType 1	Attribute Data Type 1
	22-23	MinReportingInterval 1	Minimum Reporting Interval
	24-25	MaxReportingInterval 1	Maximum Reporting Interval
	26	AttrLen 1	length of the ReportableChange field
	27-XX	ReportableChange 1	Minimum change to the attribute that will result in a report being issued (variable length)
	...	...	...
	...	Direction n[1]	0x00 - sent to server
	...	AttributeID n	Attribute ID n
		AttributeDataType n	Attribute Data Type n
	...	MinReportingInterval n	Minimum Reporting Interval
	...	MaxReportingInterval n	Maximum Reporting Interval
	...	AttrLen n	Length of the ReportableChange field
	...	ReportableChange n	Minimum change to the attribute that will result in a report being issued (variable length)

[1] n - is the last attribute reporting configuration record

4.2.5.6 ZclZtc-Configure Reporting Response Command

The ZclZtc-Configure Reporting Response command is generated in response to a ZclZtc-Configure Reporting command. On receipt of this command, the ZeD PC software is notified of the status of its original ZclZtc-Configure reporting command, for each attribute. The ZclZtc-Configure Reporting Response command format is illustrated in Table 4-28.

Table 4-28. ZclZtc-Configure Reporting Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x08 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x07 (configure reporting response command)
Attribute status record 1	status	1 byte	Status of the configure reporting operation on this attribute (can be SUCCESS if the operation was successful and is not included in order to save bandwidth, else an error code)
	direction	1 byte	Attributes are reported (0x00) or reports are received (0x01)
	attrId	2 byte	Attribute ID (is not included for successfully configured reporting attributes)
...	...	...	...
Attribute status record n[2]	status	1 byte	Status of the configure reporting operation on this attribute (can be SUCCESS if the operation was successful and is not included in order to save bandwidth, else an error code)
	direction	1 byte	Attributes are reported (0x00) or reports are received (0x01)
	attrId	2 byte	Attribute ID (is not included for successfully configured reporting attributes)

[1] For more details refer to Reference [3]

[2] n - is the last attribute status record

4.2.5.7 ZclZtc-Report Attributes Command

The ZclZtc-Report attributes command is used by a device to report the values of one or more of its attributes to Combined Interface device which using ZTC, sends the report to the ZeD PC software. The report attributes command is generated when a device has been configured by the ZeD PC software to report and when the conditions that have been configured are satisfied. Individual clusters define which attributes are to be reported and at what interval. The ZclZtc-Report attributes command format is illustrated in Table 4-29.

Table 4-29. Report Attributes Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x08 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x0x0A (report attribute command)

Table 4-29. Report Attributes Command (continued)

Sections	Parameter	Octets	Description
Attribute report 1	attrId	2 byte	Attribute ID
	attrDataType	1 byte	Attribute Data Type
	attrData	variable length	Actual value of the attribute being reported
...	...	...	...
Attribute report n[2]	attrId	2 byte	Attribute ID
	attrDataType	1 byte	Attribute Data Type
	attrData	variable length	Actual value of the attribute being reported

[1] For more details refer to Reference [3]

[2] n - is the last attribute report

4.2.5.8 ZclZtc-OnOff Command

The ZclZtc-OnOff command is generated when the ZeD PC software wishes to switch devices between 'On' and 'Off' states. The ZclZtc-Report attributes command format is illustrated in Table 4-30.

Table 4-30. ZclZtc-OnOff Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x50
	2	DestAddressMode	Address mode for the destination device (must be 0x02, only shortAddr is currently used)
	3-10	DestAddress	Address for the destination device (shortAddr)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Field specific to over the air (OTA) ZCL frame	17	Command	0x00 - Off, 0x01- On, 0x02 - Toggle

4.2.5.9 ZclZtc-Level Control Commands

The ZclZtc-Level Control commands are generated when the ZeD PC software wishes to control the characteristic of a device that can be set to a level between ‘On’ and ‘Off’. A device can move its current level to a specify value, in an up or down direction in a continuous mode or by step using ZclZtc-Level Control commands. Each command has two versions: one that performs only the level change without modifying the On/Off state and one that modifies the state, for instance turns off the device when the level is changed to the minimum value and turns it on when the level is increased. The ZclZtc-Level Control-Move to level commands format is shown in Table 4-31.The ZclZtc-Level Control-Move commands format is shown in Table 4-32. The ZclZtc-Level Control-Step command format is shown in Table 4-33. The ZclZtc-Level Control-Stop commands format is shown in Table 4-34.

Table 4-31. ZtcZc -Level Control-Move to Level Commands

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x60 without On/Off and 0x64 with On/Off
	2	DestAddressMode	Address mode for the destination device (0x00 - Indirect, 0x01- Group, 0x02 - Direct 16 bits, 0x03 - Direct 64 bits)
	3-10	DestAddress	Address for the destination device
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	18	Level	Level (0x00 - 0xff)
	19-20	TransitionTime	Transition time in 10th of a second (the time taken to move to the new level)

Table 4-32. ZclZtc - Level Control - Move commands

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x61 without On/Off and 0x65 with On/Off
	2	DestAddressMode	Address mode for the destination device (0x00 - Indirect, 0x01- Group, 0x02 - Direct 16 bits, 0x03 - Direct 64 bits)
	3-10	DestAddress	Address for the destination device
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17	MoveMode	Move mode: 0x00 - Up, 0x01 - Down
	18	Rate	Rate in units per second

Table 4-33. ZclZtc - Level Control - Step Commands

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x62 without On/Off and 0x66 with On/Off
	2	DestAddressMode	Address mode for the destination device (0x00 - Indirect, 0x01- Group, 0x02 - Direct 16 bits, 0x03 - Direct 64 bits)
	3-10	DestAddress	Address for the destination device (shortAddr)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17	StepMode	0x00-Up; 0x01-Down
	18	StepSize	Amount of units to step
	19-20	TransitionTime	Transition time in 10th of a second (time that shall be taken to perform a single step)

Table 4-34. ZclZtc - Level Control - Stop Commands

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x63 without On/Off and 0x67 with On/Off
	2	DestAddressMode	Address mode for the destination device (0x00 - Indirect, 0x01- Group, 0x02 - Direct 16 bits, 0x03 - Direct 64 bits)
	3-10	DestAddress	Address for the destination device (shortAddr)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission

4.2.5.10 ZclZtc-Group commands

Any endpoint on any device may be assigned to one or more groups (16-bit identifier, 0x0001 - 0xfff7). Using the APSDE_Data request primitive and having a DstAddrMode of 0x01 (group addressing) frames will be delivered to every endpoint assigned to the group address from DstAddr field. For remote management and commissioning of groups are defined different commands like ZclZtc-Add group command (refer to Table 4-35), ZclZtc-Remove group command (refer to Table 4-36), ZclZtc-Remove all group command (refer to Table 4-37), ZclZtc-Get Group Membership Command (refer to Table 4-38), ZclZtc-Add Group If Identifying Command (refer to Table 4-39),ZclZtc-View Group Command (refer to Table 4-40).

Table 4-35. ZclZtc-Add Group Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x30
	2	DestAddressMode	Address mode for the destination device (0x02 - Direct 16 bits shall be used)
	3-10	DestAddress	Address for the destination device (short address)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17-18	GroupID	Group ID to be added
	19	GroupNamelength	Number of characters of the group name, up to 16 (the actual value is 0x01)
	20-XX	GroupName	Name of Group (the actual value is 0x00)

Table 4-36. ZclZtc-Remove All Groups Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x34
	2	DestAddressMode	Address mode for the destination device (0x02 - Direct 16 bits shall be used)
	3-10	DestAddress	Address for the destination device (short address)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission

Table 4-37. ZclZtc-Remove Group Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x33
	2	DestAddressMode	Address mode for the destination device (0x02 - Direct 16 bits shall be used))
	3-10	DestAddress	Address for the destination device (short address)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17-18	GroupID	Group ID to be removed

Table 4-38. ZclZtc-Get Group Membership Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x32
	2	DestAddressMode	Address mode for the destination device (0x02 - Direct 16 bits shall be used)
	3-10	DestAddress	Address for the destination device (short address)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17	Count	Number of group IDs from the following list
	18-XX	GroupList	List of group IDs (a Group ID has 2 byte)

Table 4-39. ZclZtc-Add Group If Identifying Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x35
	2	DestAddressMode	Address mode for the destination device (0x02 - Direct 16 bits shall be used)
	3-10	DestAddress	Address for the destination device (short address)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17-18	GroupID	Group ID to be added
	19	GroupNamelength	Number of characters of the group name, up to 16 (the actual value is 0x01)
	20-XX	GroupName	Name of Group (the actual value is 0x00)

Table 4-40. ZclZtc-View Group Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x31
	2	DestAddressMode	Address mode for the destination device (0x02 - Direct 16 bits shall be used)
	3-10	DestAddress	Address for the destination device (short address)
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17-18	GroupID	Group ID

4.2.5.11 ZclZtc-Group Response Commands

The ZclZtc-Group Response Commands are generated by the server side of the group cluster as responses to ZclZtc-Group Commands received. The following tables describe the ZclZtc-Group Response Commands.

Table 4-41. ZclZtc-Add Group Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x09 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x00 (command ID)
The response status fields ...	Status	1byte	Status field (can be SUCCESS, DUPLICATE_EXISTS, or INSUFFICIENT_SPACE)
	Group ID	2 bytes	Group ID

[1] For more details refer to Reference [3]

Table 4-42. ZclZtc-View Group Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x09 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x01 (command ID)

Table 4-42. ZclZtc-View Group Response Command

Sections	Parameter	Octets	Description
The response status fields ...	Status	1 byte	Status of the response (can be SUCCESS or NOT_FOUND)
	GroupID	2 byte	Group ID
	GroupName	up to 16 byte	Name of the group (the actual value is 0x00)

[1] For more details refer to Reference [3]

Table 4-43. ZclZtc-Get Group Membership Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x09 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x02 (command ID)
The response status fields ...	Capacity	1 byte	Contains the remaining capacity of the group table
	Group count	2 byte	Number of Group IDs from the group list field
	GroupList	variable	Contains the Group IDs of all the group from group table

[1] For more details refer to Reference [3]

Table 4-44. ZclZtc-Remove Group Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x09 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x00 (command ID)
The response status fields ...	status	1 byte	Status of the response (can be SUCCESS or NOT_FOUND)
	GroupID	2 byte	Group ID

[1] For more details refer to Reference [3]

4.2.5.12 ZclZtc-Scenes Commands

The ZclZtc-Scenes Commands are generated when the ZeD PC software wishes to set up or recall a set of value of specific attributes. The user can store a scene (save the set of values of specific attributes at a moment; refer to Table 4-45) or to recall a scene (to restore the values of specific attributes that had been saved; refer to Table 4-46). Each scene is defined for a particular group and is stored in a store table, but scenes may also exist without a group (the commands are unicast in this case).

Table 4-45. ZclZtc-Store Scenes Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x44
	2	DestAddressMode	Address mode for the destination device (0x00 - Indirect, 0x01- Group, 0x02 - Direct 16 bits, 0x03 - Direct 64 bits)
	3-10	DestAddress	Address for the destination device
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17-18	GroupID	Store the scenes on Group ID
	19	SceneID	Scene ID to be stored

Table 4-46. ZclZtc-Recall Scenes Command

Specific fields	Byte	Parameter	Description
Parameters used to recognize the command by the embedded side and to form the APSDE_Data request primitive	0	OPG	Opcode Group is 0x70
	1	OPC	Opcode is 0x45
	2	DestAddressMode	Address mode for the destination device (0x00 - Indirect, 0x01- Group, 0x02 - Direct 16 bits, 0x03 - Direct 64 bits)
	3-10	DestAddress	Address for the destination device
	11	DestEndPoint	End point on the destination device
	12-13	ClusterID	Cluster ID
	14	SrcEndPoint	End point on the source device
	15	TxOptions	Transmit Options
	16	Radius	Radius of transmission
Fields specific to over the air (OTA) ZCL command frame	17-18	GroupID	Recall the scenes on Group ID
	19	SceneID	Scene ID to be recalled

4.2.5.13 ZclZtc-Store Scenes Response Command

The ZclZtc-Store Scenes Response Command is generated in response to a received ZclZtc-Store Scene Command (refer to Table 4-47). The ZclZtc-Recall Scenes Command has no response.

Table 4-47. ZclZtc-Store Scenes Response Command

Sections	Parameter	Octets	Description
ZCL header	frameControl	1byte	Set to 0x09 (direction response)[1]
	transactionId	1 byte	Identification number for the transaction
	command	1 byte	Set to 0x00 (command ID)
The response status fields ...	status	1 byte	Status of the store scenes response command (can be SUCCESS, INSUFFICIENT_SPACE or INVALID_FIELD)
	GroupID	2 byte	Group ID
	SceneID	1 byte	Scene ID

[1] For more details refer to Reference [3]