



# UG252: Lighting Reference Design Demo Board Kit User's Guide (RD-0035-0601, RD-0085-0401, RD-0098-0401)

Silicon Labs' compact and cost-effective lighting reference design is based on the Zigbee 3.0 standard and can be used in conjunction with a Zigbee 3.0 home automation system. This reference design kit features Silicon Labs' EM3585 (RD-0035-0601), EFR32MG1P732F256GM32 (RD-0085-0401) or EFR32MG12P432F1024GM32 (RD-0098-0401) Wireless Microcontroller (MCU).

This document contains instructions and guidelines for the following: a quick-start demonstration and next steps, system overview and operations, hardware, firmware and ecosystem considerations, and engineering and manufacturing testing. This guide applies to EmberZNet PRO 6.0 and later.

The reference design comes preprogrammed with the Zigbee Z3ColorControlLight sample application and is fully functional out of the box. All hardware design files are available within Simplicity Studio, Zigbee firmware application source code is available within the EmberZNet PRO stack, and Thread firmware application source is available within the Thread stack. Access to Zigbee and Thread stacks is available after purchasing and registering a wireless starter kit such as the SLWSTK6000B. Additional information about acquiring a wireless starter kit is available: <https://www.silabs.com/products/development-tools/wireless/mesh-networking/mighty-gecko-starter-kit>. A wireless starter kit mainboard (sold separately) and SLSDA001A Simplicity debug adapter board (included after December, 2017, and also sold separately) are required to program the RD-0085-0401 and RD-0098-0401. An ISA3 debug adapter (sold separately) is required to program the RD-0035-0601. Additional information about acquiring the SLSDA001A Simplicity debug adapter board is available: <https://www.silabs.com/products/development-tools/mcu/32-bit/simplicity-debug-adapter-board>.

The Lighting reference design RD-0098-0401 may be reprogrammed with the Dimmable Light sample application from the Thread 2.4 stack and later. While this document is specific to the Zigbee function, efforts have been made to achieve functional parity. Future versions of this document will discuss Thread functionality in detail.

## KEY POINTS

- Quick-start demonstration
- Recommended next steps
- Kit overview
- Device operation
- Hardware and firmware considerations
- Ecosystem considerations
- Engineering and manufacturing testing

## 1. Introduction

The Silicon Labs Zigbee Lighting Reference Design modules and demonstration boards are designed to demonstrate Zigbee lighting applications with EFR32MG or EM3585 Zigbee chipsets. The modules are preprogrammed for an "out of the box" demonstration using the demonstration board shown in the figure below. The modules can also be directly integrated into a connected lighting application as described in [6.3 Module Hardware Integration Guidelines](#). An example of such an application is shown in the figure below. Hardware design files and firmware application source code are available for all items in the kit.

This document will address quickstart for demonstration and next steps, system overview and operation, hardware and firmware considerations, ecosystem considerations, and engineering and manufacturing test.

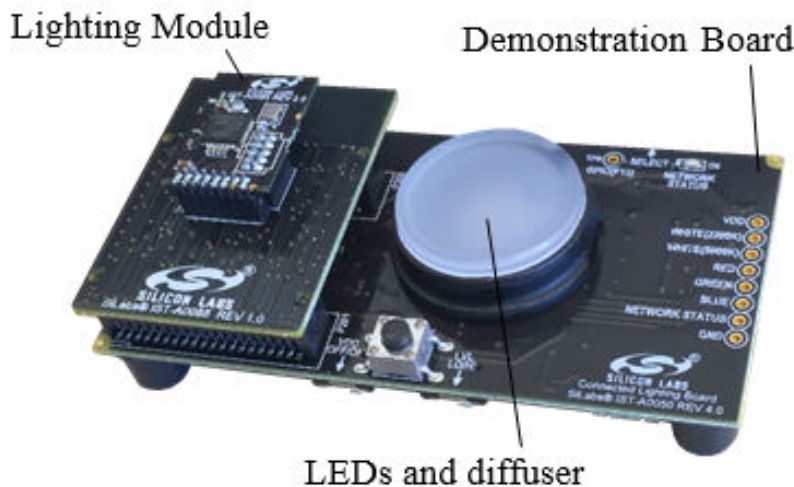


Figure 1.1. Lighting Reference Design module and demonstration board

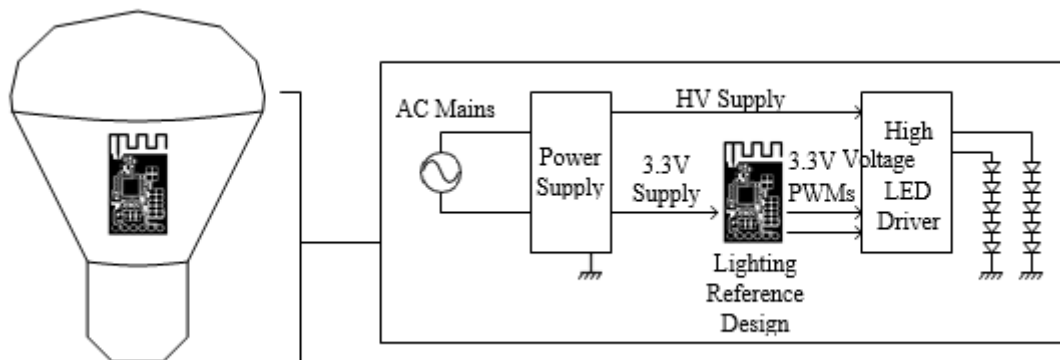
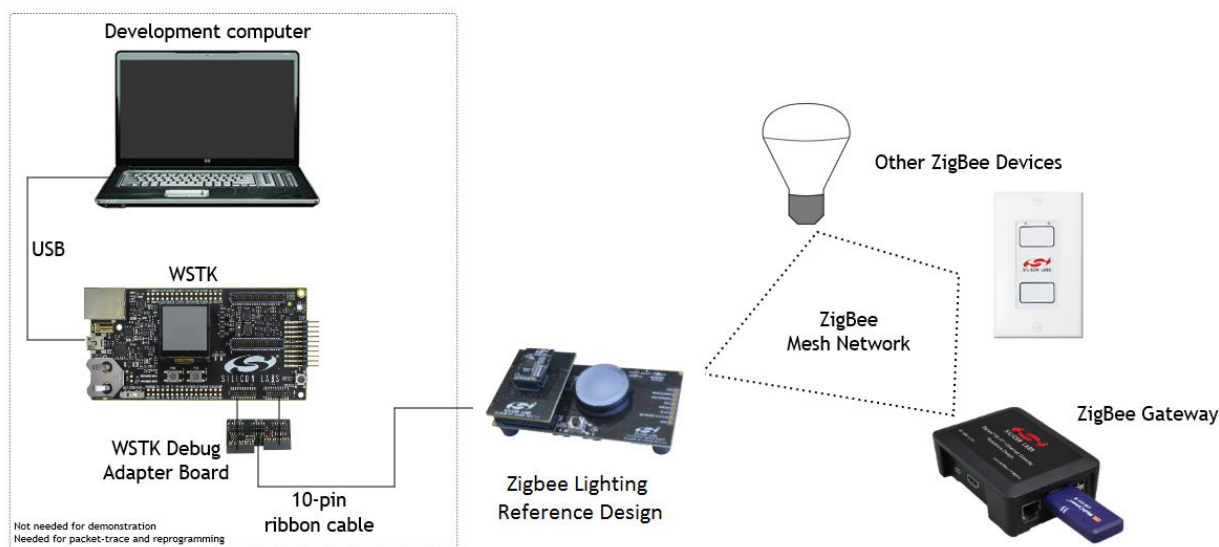


Figure 1.2. Connected Lighting Application Example

## 2. Quick-Start Demonstration

The Lighting Reference Design demo board can be demonstrated as part of a Zigbee network. All Zigbee networks must have one device, such as a gateway, that plays the role of coordinator and allows commissioning of new devices on the network. Other Zigbee devices may be present in order to evaluate interoperability with the lighting reference design. An example Zigbee network is shown in the figure below.



**Figure 2.1. Example Zigbee network used for demonstration and evaluation**

### 2.1 Verify the Kit Contents

Make sure the following items are part of your Zigbee Lighting Reference Design kit.

1. Zigbee Lighting Reference Design module and demo board assembled hardware
2. micro-USB cable for power
3. Quickstart card to obtain the latest reference design collateral

### 2.2 Set Up a Zigbee 3.0 Gateway

Silicon Labs offers RD-0002-0201 or RD-0001-0201 Zigbee gateways (sold separately) for the purpose of evaluating the Zigbee Lighting Demo Board. However, a commercially available Zigbee 3.0 compliant gateway that supports level lighting with device profile ID=0x0100, 0x0101 and 0x0102 may work for this purpose provided that the gateway does not require device whitelisting or optional clusters not implemented in the reference design firmware of the Zigbee lighting module. See the section titled [2. Quick-Start Demonstration 2. Quick-Start Demonstration](#) for more information.

### 2.3 Supply Power to the Zigbee Lighting Demo Board

1. Use a cable such as the one provided in the kit to connect the micro-USB header on the demo board to a standard USB port or +5VDC power supply.
2. The demo board is powered when the supply indicator LED is lit.

### 2.4 Join the Zigbee Lighting Demo Board to the Zigbee Network

1. Follow the directions provided with the Zigbee gateway to enable new devices to join the network.
2. Press the demo board power supply interrupt button S1 rapidly 10 times to begin the network join procedure. This mimics the effect of power cycling a light bulb 10 times.
3. The demo board has joined the Zigbee network when LEDs D1-D5 flash 10 times.
4. On RD-0085-0401 and RD-0098-0401 network status is also indicated by the green network status LED. Solid green color means it is on a network. LED off means the bulb is not on a network. Note that the slider switch S2 needs to be set towards "NETWORK STATUS" to enable this feature.

## 2.5 Demonstrate Lighting Control

1. Use the gateway interface to turn LEDs D1, D2, and D3-D5 (RD-0085-0401 and RD-0098-0401 only) on and off.
2. Set the intensity (also called the brightness or dimming level).
3. Set the color of D1.
4. Set the color temperature.
  - a. On RD-0035-0601, color temperature is produced using RGB LED D1.
  - b. On RD-0085-0401 and RD-0098-0401, color temperature is produced using RGB LED D1 when the module is loaded with the Z3ColorControlLight example application or using color-temperature tuned LEDs D2, D3, D4 and D5 when the module is loaded with Z3ColorTempLight example application. The reference designs are pre-programmed with the Z3ColorControlLight example application.

## 3. Recommended Next Steps

### 3.1 Evaluate the Demo Board

The Zigbee Lighting Reference Design has been designed to support common Zigbee lighting application requirements. Typical areas for evaluation include the following:

- PWM frequency and duty cycles.
  - RF performance such as channel power and range.
  - Zigbee network behavior such as network join and leave
- The remainder of this document covers these topics in detail.

### 3.2 Evaluate the Firmware

If firmware modification is necessary:

Visit the Zigbee getting started page and order a Wireless Starter Kit SLWSTK6000B: <https://www.silabs.com/products/development-tools/wireless/mesh-networking/mighty-gecko-starter-kit>.

## 4. Overview

The Silicon Labs Lighting Reference Design demo board facilitates the evaluation of the Zigbee Lighting Reference Design module and the customization of lighting application firmware. In addition to programming and debug access, the demo board has a number of features specifically useful for lighting applications such as RGB and Color-Temperature LEDs, a power-interrupt button to mimic switching of the mains supply, and PWM test points that can be wired directly into a light engine.

### 4.1 Part Numbers

The part numbers convention is RD-XXXX-YYYY, where:

RD	Reference Design
XXXX	Reference Design Number
YYYY	Reference Design Component

This document will call out the reference design number (i.e., RD-XXXX) when describing the complete design and the reference design component (i.e., RD-XXXX-YYYY) when describing a specific component.

The table below provides a description and PCB marking for each part number. Note that in some cases there is not sufficient space on the PCB and an internal "IST" marking appears on the PCB instead of the "RD" part number.

**Table 4.1. Part Numbers and Descriptions**

Part Number	PCB Marking	Description
RD-0098-0401	N/A	Lighting Reference Design with EFR32MG12P432F1024GM32 Demo Board Kit
N/A	IST-A0050	Lighting Demo Board
RD-0098-0101	IST-A0098	Lighting Reference Design with EFR32MG12P432F1024GM32 Demo Board - module only
RD-0085-0401	N/A	Lighting Reference Design with EFR32MG1P732F256GM32 Demo Board Kit
N/A	IST-A0050	Lighting Demo Board
RD-0085-0101	IST-A0085	Lighting Reference Design with EFR32MG1P732F256GM32 - module only
RD-0035-0601	N/A	Lighting Reference Design with EM3585 Demo Board Kit
RD-0035-0501	RD-0035-0501 IST-A21	Lighting Reference Design with EM3585 Demo Board
RD-0035-0101	IST-A35	Lighting Reference Design with EM3585

### 4.2 Features and Benefits of the RD-0085-0401 and RD-0098-0401 Demo Board Kits

#### 4.2.1 Module

The reference design kits include an IST-A0085 lighting module. This module is pre-programmed with the Z3ColorControlLight Zigbee lighting application.

### 4.2.2 Adapter Board

Each reference design module is mounted on an adapter board to provide connectivity to the Zigbee lighting demo board as well as the Silicon Labs Wireless STK. The adapter board and lighting demo board also have mounted EEPROMs with board-specific information stored. This allows Silicon Labs software to easily detect the connected hardware and configure themselves to show relevant information.

### 4.2.3 Lighting Demo Board

The Zigbee lighting demo board provides for the evaluation of the Zigbee lighting reference design module. Features are listed in the table below.

Demo Board Feature	Benefit
Low and high temperature white LEDs, tri-color RGB LED	Simulate LED lighting scenarios
Network status LED	LED ON means on a Zigbee network. LED OFF means not on a Zigbee network.
Micro-USB connector	Provides power
10-pin mini-Simplicity compatible connector	Facilitates programming, debug, and packet trace
Power down button	Mimics momentary mains switch
Green power indicator LED	Demo board power indication
SW1 brightness switch	Selects between High and Low LED brightness
SW2 network status LED switch	Selects between routing PWM output to Network Status LED or TP9. Can be used to turn off the LED or select GPIO for different purpose.
SW3 power switch	Mimics mains power switch
LiPo battery connector	Provides additional power options

**Note:** The single-cell LiPo battery begins charging with power supplied through the micro-USB connector.

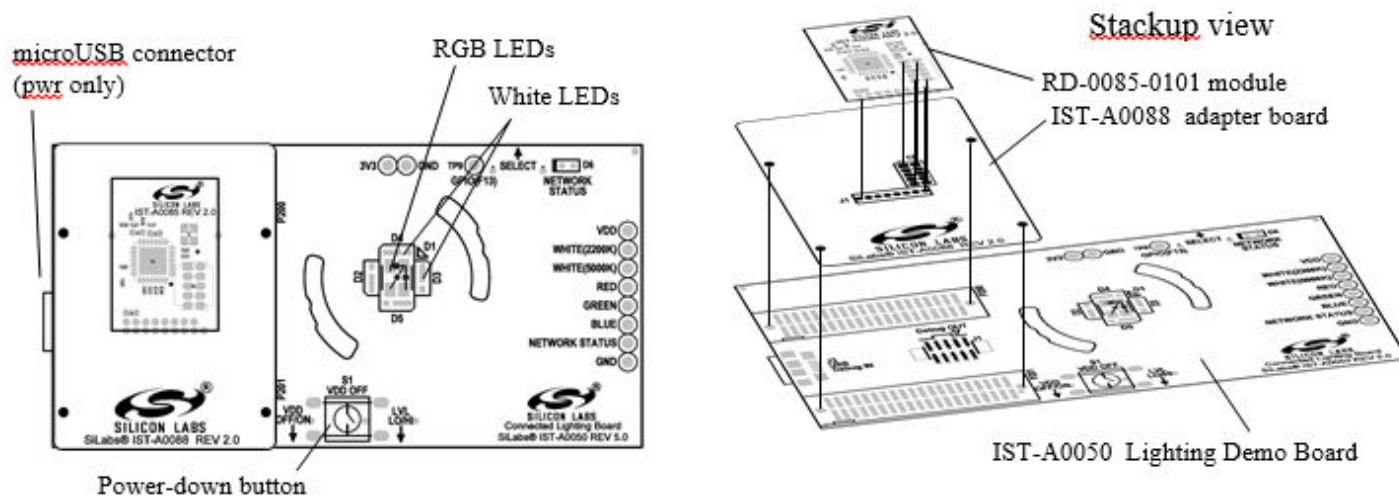


Figure 4.1. Complete RD-0085-0401 Zigbee Lighting Reference Design Board Kit

## 4.3 Features and Benefits of the RD-0035-0601 Demo Board Kit

### 4.3.1 Board Lighting Reference Design Module

The reference design kits include an RD-0035-0101 lighting module. These modules are pre-programmed with the Z3ColorControlLight Zigbee lighting application.

### 4.3.2 Adapter Board

Each reference design module is mounted on an adapter board to provide connectivity to the Zigbee lighting demo board.

### 4.3.3 Lighting Demo Board

The Zigbee lighting demo board provides for the evaluation of the Zigbee lighting reference design module. Features are listed in the table below.

Demo Board Feature	Benefit
White LED, tri-color RGB LED	Simulates LED lighting scenarios
10-pin ISA3 compatible connector	Facilitates Zigbee wireless packet debugging
Power down button	Eases network commissioning (join/leave)
Red power indicator LED	Demo board power indication
Micro-USB and LiPO battery connectors	Provides dual-power options

**Note:** The single-cell LiPo battery begins charging with power supplied through the micro-USB connector.

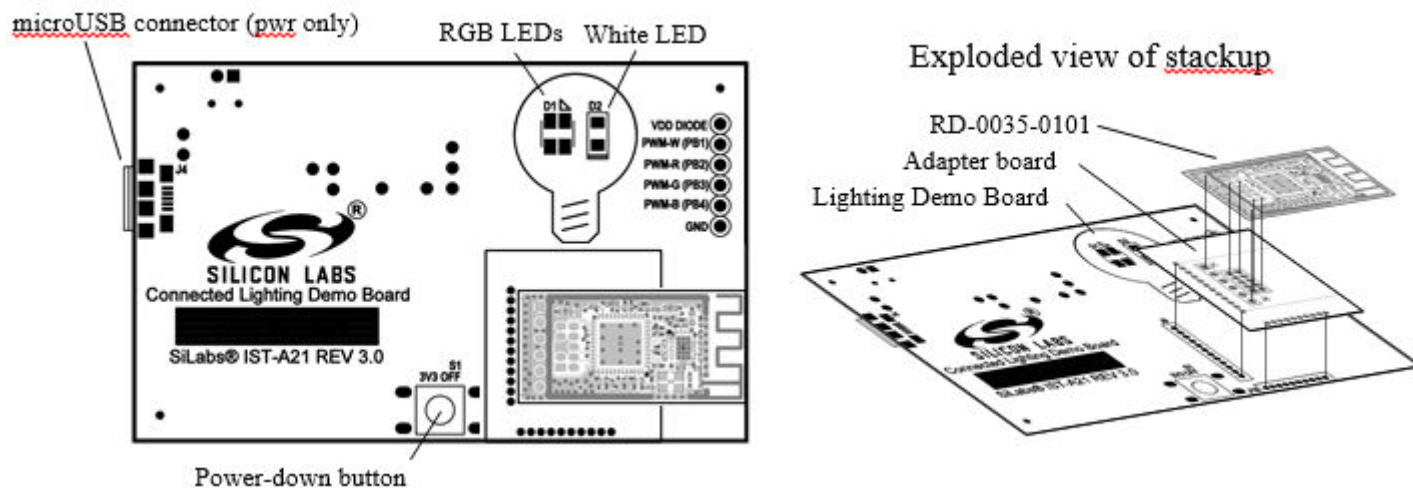


Figure 4.2. Complete RD-0035-0601 Zigbee Lighting Reference Design Board Kit



## 4.4 Features of the Lighting Reference Designs Modules

### 4.4.1 Lighting Reference Design Hardware Features

	RD-0098-0101	RD-0085-0101	RD-0035-0101
Processor	32-bit ARM® Cortex-M4		32-bit ARM® Cortex-M3
Crystal frequency	38.4 MHz		24 MHz
Encryption Accelerators	AES128		
Advanced Crypto Accelerators	SHA-1, SHA-2, ECC		—
Peripherals	ADC, UART, SPI, TWI		ADC, UART
Flash size	1024 kB	768 kB	512 kB
Number of PWMs	6		4
MAC/PHY	2.4 GHz IEEE 802.15.4		
Packet trace	Included		
Antenna	PCB, Inverted F		
Voltage range	2.3V - 3.6V		
Temperature range	-40C to 125C		
Transmit power	+20 dBm		
FCC certification	Pre-certified FCC Part 15		

### 4.4.2 Lighting Reference Design Firmware Features

Application firmware is available both as precompiled binaries and as source via the EmberZNet PRO stack. Firmware can be updated either via a hardwired debugger connection or other-the-air (OTA) using an OTA-enabled gateway.

The following clusters are used in the reference design application.

Cluster Name	Cluster Hex Value
Basic	0x0000
Identify	0x0003
Groups	0x0004
Scenes	0x0005
On/Off	0x0006
Level	0x0008
Over-the-Air Boot Loading	0x0019
Color Control	0x0030
Diagnostics	0x0B05

## 5. Operation Details for the Lighting Reference Designs

### 5.1 Zigbee Network Behavior

This section will describe the various operational states a powered lighting reference design is able to occupy as it joins and leaves a Zigbee network. The input mechanisms for operation utilize a physical power-cycling methodology which can either be employed by externally disabling and enabling the demo board's power supply or by utilizing button S1 where a button assertion equates to powering off and releasing equates to power up. The white and tri-color LEDs operate as output signaling information as the demo board traverses or occupies a functional state. Refer to the figure below for a pictorial representation of this process.

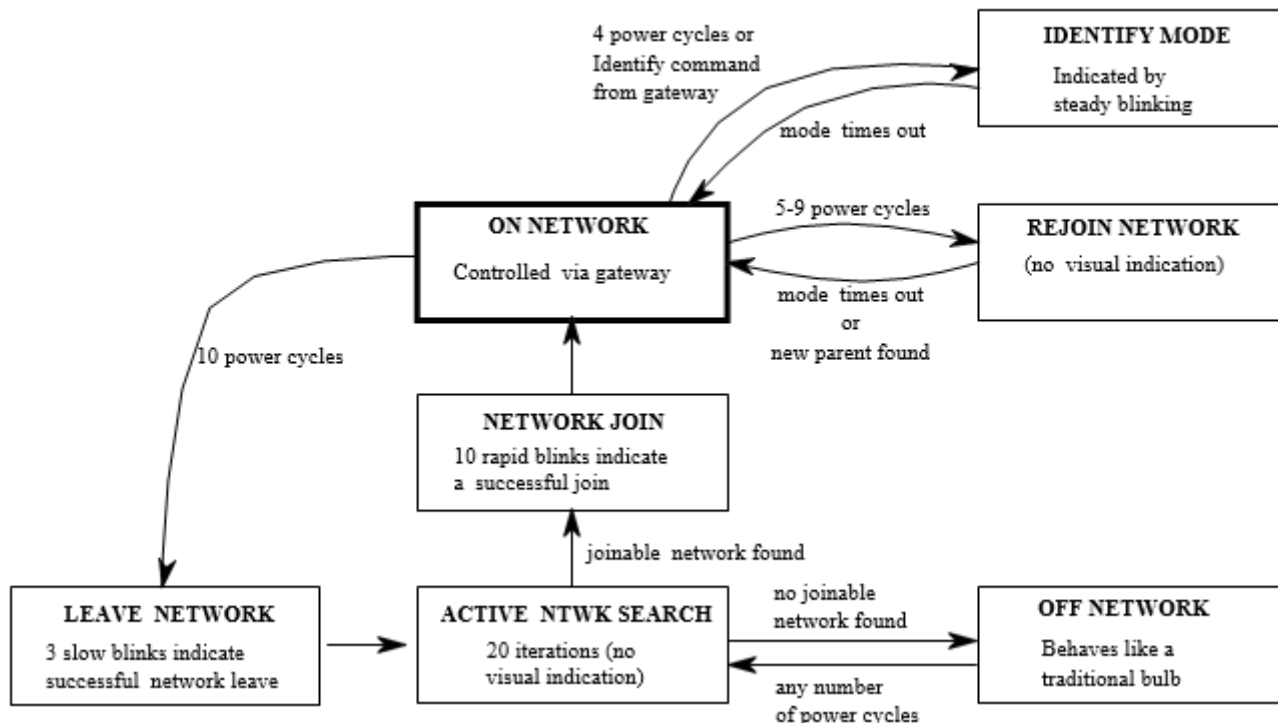


Figure 5.1. Device Operation State Diagram

#### 5.1.1 First Time Demo Board Power-Up

After the hardware has been powered up for the first time, the device will be in a factory default mode and be placed in the ACTIVE NTWK SEARCH state.

#### 5.1.2 OFF NETWORK State

This state is a condition where the demo board is not connected to a network and is not aggressively searching for a network. To leave this state and to enter the ACTIVE NTWK SEARCH state, the power to the board needs to be cycled at least one time (most easily accomplished by pressing button S1). The device does continue to perform infrequent searches for networks while in the OFF NETWORK state.

The white/tri-color LEDs on the board will operate like a conventional non-connected light when in this state.

#### 5.1.3 ACTIVE NTWK SEARCH State

This state is a condition where the demo board is actively searching for a network but has not joined a network. This is an operation for actively searching for open network and if a suitable one is found it will attempt to join. The process of searching will repeat for three different security keys, install code derived preconfigured link key, default global trust center link key and distributed security global link key. If no open network is found, the whole process will start again. It will repeat for up to 20 times. The demo board will then enter into the OFF NETWORK state. If a network is found it will enter the NETWORK JOIN state and attempt to join the network with the security key it is currently used for searching and joining. If the joining is successful it will enter the ON NETWORK state. If the joining is not successful it will fall back to the previous state of ACTIVE NTWK SEARCH. Power-cycling while in ACTIVE NETWORK SEARCH state will reset the timer.

### 5.1.4 NETWORK JOIN State

After a demo board leaves the ACTIVE NTWK SEARCH state and enters this state, the white/tri-color LEDs will flash 10 times to indicate successfully finding and joining a network. This state is transitory as the light moves immediately to the ON NETWORK state once the LEDs complete their flash sequence. Power-cycling at this time will only impair the 10 LED flashes, but not prevent entry into the ON NETWORK STATE once the power is reapplied.

### 5.1.5 ON-NETWORK State

This state is a condition where the demo board is connected to a Zigbee 3.0 network and is not searching for a network. While in this state, the white/tri-color LEDs illumination on the demo board will correspond to commands the demo boards receive from the Zigbee network. This state contains multiple entry and exit paths depending on various factors such as number power-cycles received within a short time period of a few consecutive seconds.

*If the power to the demo board is removed for long periods of time, the demo board will immediately rejoin its original joined network when power becomes available and will not seek other networks, unless the user forces it to leave its joined network as will be described within this section.*

*If the demo board is power-cycled four times, the demo board will temporarily enter the IDENTIFY state while remaining on its joined network.*

*If the demo board is power-cycled from five to nine times, the demo board will temporarily enter the NETWORK STATUS state while remaining on its joined network.*

*If the demo board is power-cycled ten times OR if the demo board receives a leave network command, the demo board will enter the LEAVE NETWORK state.*

### 5.1.6 IDENTIFY State

The purpose of this state is to help an installer (or user) identify which bulb is being communicated to by visually flashing in a specified pattern so this person can locate the targeted light. After a demo board leaves the ON-NETWORK state and enters this state, the white/tri-color LEDs will flash at a 0.5 Hz rate. This state is temporary and will return the demo board into the ON NETWORK state once the state's timer reaches its time limit or if power is cycled at least once to the demo board. There are two ways to enter this mode depending on the intended application.

The first method is done locally to the bulb by power cycling it 4 times. The main purpose of this method is to support EZ Mode commissioning, whereby the light becomes discoverable by other devices as a candidate for pairing. The bulb will blink at 0.5 Hz until the Identify Mode times out after 3 minutes.

The second method is done remotely by sending an Identify command from a gateway. The main purpose of this method is to allow an installer to visually locate the bulb in question. The bulb will blink at 0.5 Hz until the Identify Mode for the duration is requested by the gateway.

### 5.1.7 NETWORK STATUS State

The purpose of this state is to help an installer (or user) identify if the light is successfully connected to a network. After a demo board leaves the ON-NETWORK state and enters this state, the white/tri-color LEDs will flash at a 2 Hz rate. This state is temporary and will return the demo board into the ON NETWORK state once the state's timer reaches its time limit or if power is cycled at least once to the demo board.

### 5.1.8 LEAVE NETWORK State

After a demo board leaves the ON NETWORK state and enters this state, the demo board will leave the joined network and will indicate this process is in action by slow flashing the white/tri-color LEDs three times at a 0.5 Hz rate. This state is temporary and will place the demo board into the ACTIVE NTWK SEARCH state once the LEDs complete their flash sequence. Power-cycling will only impair the slow LED flashes but not prevent entry into the ACTIVE NTWK SEARCH state once the power is reapplied.

## 6. Hardware Details for the Lighting Reference Design

This section describes the key aspects of the reference design.

### 6.1 Lighting Reference Design RD-0085-0101 and RD-0098-0101

This section describes the key hardware specifics of the RD-0085-0101 and RD-0098-0101 modules.

### 6.1.1 Key Highlights

- EFR32MG Zigbee PRO radio with stacked flash (RD-0085-0101) or internal flash (RD-0098-0101)
- Two-layer 0.062" printed circuit board with all components on primary side
- 8-pin 0.050" application header with VDD, GND, and six PWM output pins
- 10-pin debug header with packet trace functionality

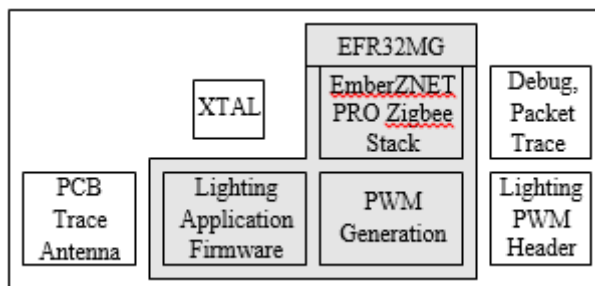


Figure 6.1. Module Block Diagram

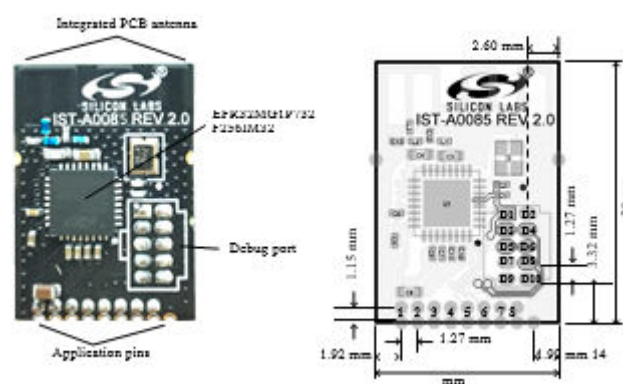


Figure 6.2. Module Photo, Pin Numbering, and Dimensions

Table 6.1. Lighting PWM Header

Pin #	Pin Name	Preconfigured Pin Connection	Pin Function
1	VDD		3.3V Supply Voltage
2	GND		Ground
3	PB14	PWM - Timer 0, Channel 0	White Light (Higher Temperature)
4	PB15	PWM - Timer 0, Channel 1	White Light (Lower Temperature)
5	PB11	PWM - Timer 0, Channel 2	Network Status
6	PC10	PWM - Timer 1, Channel 0	Red Light
7	PD15	PWM - Timer 1, Channel 1	Green Light
8	PC11	PWM - Timer 1, Channel 2	Blue Light

**Note:** Additional pin functions require custom firmware. Refer to the data sheet for available functions.

**Table 6.2. Debug Port**

Pin #	Pin Name	Pin Functions	Description
D1	VDD	VDD	3.3V Supply Voltage
D2	GND	GND	Ground
D3	RESETn	RESETn	MCU Reset
D4	PA1	USART_RX	USART RX
D5	PA0	USART_TX	USART TX
D6	PF2	SW_O/TDO	Serial Wire Viewer Output
D7	PF1	SW_DIO	Serial Wire Data Input/Output
D8	PF0	SW_CLK	Serial Wire Clock
D9	PB13	PTI_SYNC	Packet Trace Enable
D10	PB12	PTI_DATA	Packet Trace Data

**Note:** The debug connector is compatible with the mini-simplicity packet trace port connector, connected from the top side.

### 6.1.2 Conducted RF Connector

The RF connector on the secondary side of the RD-0098-0101 board provides an easy way to access the RF output signal from the chip. This can be useful in setting up conducted RF setups for testing and demos. Note that the signal coming to the connector is not optimized for performance and FCC emission requirements. Only the radiated signal is guaranteed to perform within FCC specifications.

In order to use the connector, depopulate resistor R1, close solder bridge J4, and populate connector J3.

Connector Manufacturer: Murata Electronics

Connector Part Number: MM5829-2700RJ4

Please check manufacturers site for compatible mating connectors and cable assemblies.

## 6.2 Zigbee Lighting Reference Design RD-0035-0101

This section describes the key hardware specifics of the RD-0035-0101 module.

### 6.2.1 Key Highlights

- EM3585 Zigbee Pro radio with embedded flash
- SKY66019 power amplifier (PA)
- Four-layer 0.062" printed circuit board with all components on primary side
- 6-pin 0.050" application header with VDD, GND, and four PWM output pins
- 10-pin debug header with packet trace functionality

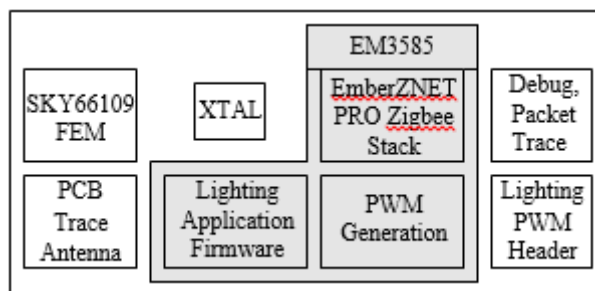


Figure 6.3. Module Block Diagram

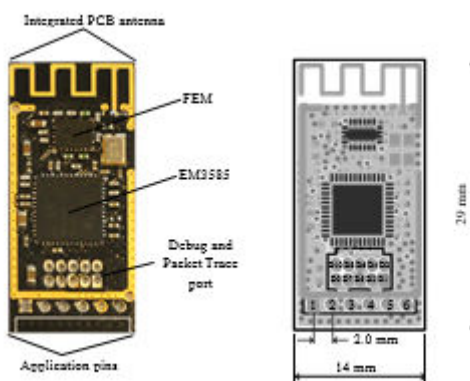


Figure 6.4. Module Photo, Pin Numbering, and Dimensions

## 6.2.2 Connector Pin Assignments

**Table 6.3. Lighting PWM Header**

PIN #	Pin Name	Function					Description
		Preconfigured		Additional			
1	GND						Ground
2	VDD						3.3 Supply Voltage
3	PB1	PWM1	SC1MISO	SC1MOSI	SC1TXD	SC1SDA	GPIO PB1, SPI1 MISO, SPI1 MOSI, UART1 TX Out, I2C1 Data, PWM1 Output
4	PB2	PWM2	SC1MISO	SC1MOSI	SC1RXD	SC1SCL	GPIO PB2, SPI1 MISO, SPI1 MOSI, UART1 RX In, I2C1 Clock PWM2 Output
5	PB3	PWM3	SC1SCLK		SC1nCTS		GPIO PB3, SPI1 Clock, UART1 Clear to Send, PWM3 Output
6	PB4	PWM4	SC1nSSEL		SC1nRTS		GPIO PB4, SPI1 Chip Select, UART1 Request to Send, PWM4 Output

**Note:** Secondary pin functions require custom firmware.

**Table 6.4. Packet Trace Port**

Pin #	Pin Name	Pin Functions	Description
D1	VBRD		3.3V Supply Voltage
D2	PC2	JTDO/SWO	JTAG Data Out, Serial Wire Out
D3	PC0	nJRST	JTAG Reset
D4	PC3	JTDI	JTAG Data In
D5	GND		Ground
D6	SWCLK	JTCK/SWCLK	JTAG Clock, Serial Wire Clock
D7	PC4	JTMS/SWDIO	JTAG Mode Select, Serial Wire Data In/Out



Pin #	Pin Name	Pin Functions	Description
D8	nRESET		EM3xxx Reset
D9	PA4	PTF	Packet Trace Enable
D10	PAS	PTD	Packet Trace Data

**Note:** The debug connector is compatible with the ISA3 packet trace port connector when placed on the bottom side of the PCB.

### 6.3 Module Hardware Integration Guidelines

This section contains guidelines on how to integrate the lighting module in a custom application. Schematically, the integration is straightforward: the user only needs to provide power and ground hookups and ensure that the PWM signals are wired to the appropriate locations in the light engine. Layout and mechanical design can be more involved, so tips are provided below.

The Power, Ground and PWM signals are not intended to carry high-frequency information and can therefore be handled using standard best practices of PCB design.

Placing any type of material near the RF section and antenna, particularly metals and strong dielectrics, will have a big effect on antenna performance and on spurious emissions. The region nearest the antenna within 1 wavelength, which at 2.4 GHz is approximately

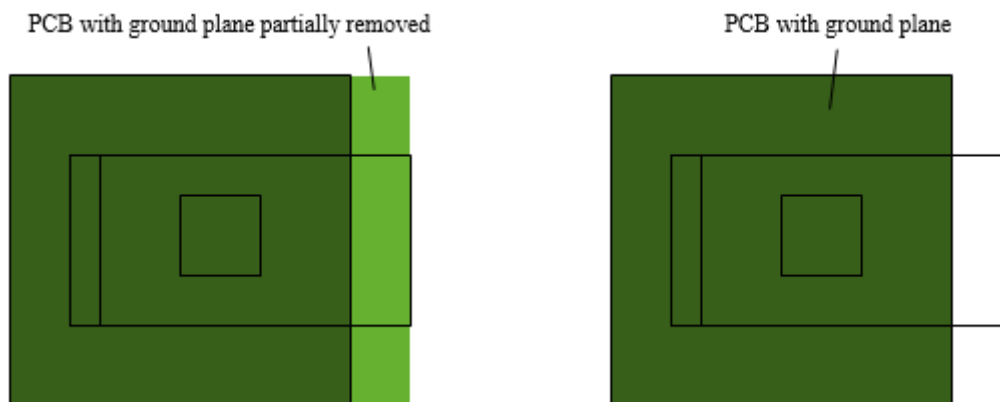
12.5 cm or 5 inches, is known as the near-field region. Ideally no foreign materials should be introduced in this region.

In practice, lightbulbs are not large enough to allow for that keep-out. However, we have observed that keeping metals at least 2.5 cm away from the antenna tends to produce good results. Metal that is part of a radiating ground plane is not harmful and may often be extended without issue, although it does have an effect on antenna matching. This generally includes ground planes under the module (but not under the antenna).

The following images show the most common layouts as an example of which layout practices are suggested

#### 6.3.1 Horizontal Mounting Guidelines

Either hang the antenna off the PCB edge (right figure) or remove the PCB ground plane beneath the antenna and extending beyond the module along the edge (left figure). Both RD-0098, RD-0085 and RD-0035 can be mounted using a through-hole header. In addition, RD-0085 and RD-0098 have castellation pins around the perimeter that allow it to be used in a standard SMT flow without the need for an intervening header.

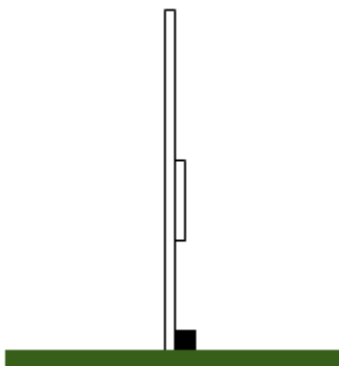


The following connectors are recommended for horizontal mounting applications.

	Through-hole mount	Surface mount
RD-0098-0101	FTS-108-01-F-S (Samtec)	Solder directly to castellation pins
RD-0085-0101	FTS-108-01-F-S (Samtec)	Solder directly to castellation pins
RD-0035-0101	TMM-106-01-G-S (Samtec)	N/A

### 6.3.2 Vertical Mounting Guidelines

The lighting reference design modules can also be mounted vertically using a right angle connector. There is no inherent advantage in this arrangement other than to provide additional flexibility in the mechanical design of the host application.



The following right-angle connectors are recommended for vertical mounting applications.

	Through-hole mount	Surface mount
RD-0098-0101	TM-108-01-F-S-RA (Samtec)	N/A
RD-0085-0101	TM-108-01-F-S-RA (Samtec)	N/A
RD-0035-0101	TMM-106-01-G-S (Samtec)	N/A

### 6.3.3 Connector Information

The following links provide further information about recommended connectors:

- <http://suddendocs.samtec.com/prints/fts-1xx-xx-x-sx-mkt.pdf>
- <http://suddendocs.samtec.com/prints/tms-1xx-xx-xx-x-xx-xxx-mkt.pdf>
- <http://cloud.samtec.com/Prints/TMM-1XX-XX-XX-X-XX-XXX-MKT1.pdf>
- <http://cloud.samtec.com/Prints/TMM-STH.PDF>
- <http://cloud.samtec.com/Prints/TMM-1XX-XX-XX-X-RA-XXX-MKT.pdf>

## 6.4 Module Hardware Modification Guidelines

Modification of the reference design module is discouraged, however we recognize that some users may require slight changes in order to satisfy mechanical requirements in specific installations.

### 6.4.1 Modify With Care

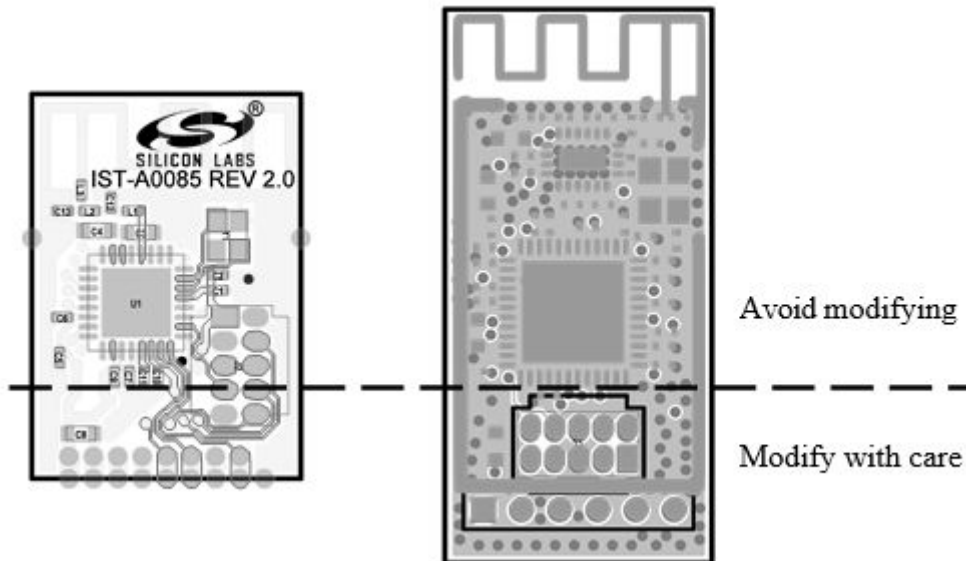
While any modification is discouraged, those listed here should have less effect on RF performance.

- Changing the type of connector
- Pitch, spacing, rows, size
- Changing from castellation to through hole, and vice versa
- Selecting different GPIO pins
- Pins close to the RF Frontend should not be modified

## 6.4.2 Avoid Modification

All changes listed here may have serious impacts on module performance.

- Modifying the spacing near any IC, especially near the RF ports
- Modifying the layout of RF bypass capacitors
- Changing the PCB Antenna Length
- Altering any part of the Ground Pour Area
- Changing the number of metal layers



## 7. Firmware Details for the Lighting Reference Design

The purpose of this section is to describe the lighting reference design application firmware.

### 7.1 Obtaining the Firmware Application

The firmware application is loaded into the flash of the EM3585, EFR32MG1P232F256 or EFR32MG12P432F1024 during the manufacturing process. The firmware application source code is delivered as part of EmberZNet PRO, which is available to registered users of a wireless starter kit such as the SLWSTK6000B.

For more information, visit the Zigbee [Getting Started](#) page.

### 7.2 Building the Firmware Application

The instructions below describe how to build the device firmware.

1. Install EmberZNet PRO.
2. Create a new Application Framework Configuration in AppBuilder and select the SoC stack release.
3. Create a project using Z3ColorControlLight sample application.
4. During Project Setup, verify that the proper board and chip is selected.
5. In AppBuilder, under the “hal configuration tab”, verify the architecture and header files:
  - a. EM3585
  - b. EFR32MG1P732F256GM32 and Board header ist\_a0085
  - c. EFR32MG12P432F1024GM32 and Board header ist\_a0098
6. Generate and note the directory in which the project files were created.
7. Save the project file into the directory you just created.
8. Compile with IAR (for version compatibility information see the EmberZNet PRO release notes and QSG106: Getting Started with EmberZNet PRO). At this point you can load the image onto the Smart Outlet reference design using the Board Header Reprogramming method described above.

### 7.3 Programming the RD-0085-0101 and RD-0098-0101 Lighting Reference Design

The reference design provides two methods to re-program the demo board:

- Using a WSTK debugger and mini-simplicity adapter board to upload an .s37, .bin or .hex image
- Using the over-the-air (OTA) upgrade feature with an .ota image file.

#### 7.3.1 Board Header Reprogramming

The demo board can be reprogrammed with an available .s37 or .hex file and a WSTK programmer. Please refer to Application Note AN958 for more information on how to successfully program the demo board with the WSTK programmer and mini-simplicity adapter board. The debug connector on the demo board is the J2 header. Note the orientation of the connector where the keyed side of the connector corresponds to the key marking found with the silkscreen drawing surrounding the J2 header.

#### 7.3.2 Over-the-Air Programming

The demo board can be reprogrammed with an available .ota file and a device that can perform OTA upgrades such as the RD-0002-0101 Zigbee USB Gateway Kit supported by Silicon Labs. Refer to the gateway documentation for more information on how to reprogram via OTA upgrade.

### 7.4 Programming the RD-0035-0101 Lighting Reference Design

The reference design provides two methods to reprogram the demo board:

- Using the ISA3 debugger with an .s37 or .hex image file
- Using the over-the-air (OTA) upgrade feature with an .ota image file

### 7.4.1 Board Header Reprogramming

The demo board can be reprogrammed with an available .s37 or .hex file and an ISA3 programmer. Please refer to the Silicon Labs UG110 User's Guide on ISA3 for more information on how to successfully program the demo board with the ISA3 programmer. The ISA3 connector on the demo board is the J3 header.

**Note:** The orientation of the connector where the keyed side of the connector corresponds to the ISA3 key marking is found with the silkscreen drawing surrounding the J3 header.

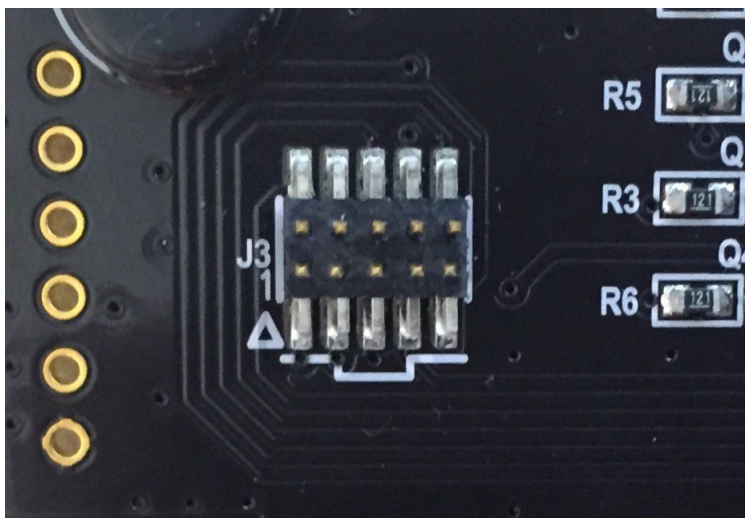


Figure 7.1. Demo Board's ISA3 Header

### 7.4.2 Over-the-Air Reprogramming

The demo board can be reprogrammed with an available .ota file and a device that can perform OTA upgrades such as the RD-0002-0101 Zigbee USB Gateway Kit supported by Silicon Labs. Refer to the gateway documentation for more information on how to reprogram via OTA upgrade.

## 7.5 Lighting Reference Design Cluster Documentation

Several plugins were developed or altered as part of the lighting reference design work. This section documents these new clusters. Here are some highlights of the new functionality:

- Ability to configure radio transmit power and specific bulb-related attributes (such as PWM frequency) using OTA commands
- Ability to execute RF test commands using a radio command
- Ability to directly configure the PWM pulse width for individual PWMs using a CLI command
- An example on setting up and using an LED calibration table in the chip's memory

### 7.5.1 Bulb-UI

The purpose of this plugin is to drive the user interface for the bulb. The user interface is primarily concerned with deciding when to join or leave a network. It does this by counting the number of rapid power cycles. By default, power cycling the bulb 10 times in quick succession (< 8 seconds of on time) will cause a joined bulb to leave the network and begin to search for a new network.

### 7.5.1.1 Theory of Operation

The primary task of the bulb-ui plugin is to manage the network connectivity state. When the bulb is not on the network, it will kick off the joining procedure. It begins by scanning on the Zigbee preferred channels. If it does not find a joinable network, it will continue on to scanning on all legal 802.15.4 channels. It proceeds to scan for up to 20 times or until it finds a joinable network, at which point it will stop scanning. If the bulb-ui plugin has not found a network, it will assume there are no legal networks at this time, operate as a normal non-connected LED bulb, and kick off a new join procedure after a power cycle.

Once on the network, the bulb-ui plugin will proceed to user interface mode. A user can input commands to the bulb by means of a series of rapid on/off power cycles. The bulb will count these power cycles and take actions as indicated by the number of power cycles. If the bulb is power cycled 4 times, it will enter identify mode. The bulb will begin to blink at the identify rate of 0.5 Hz (as defined by the bulb project's `callbacks.c` file) for 3 minutes, as specified by the EZ Mode commissioning behavior in the Zigbee Home Automation profile.

Power cycling the bulb between 5 and 9 times will initiate rejoin behavior. The bulb is designed to be a non-sleepy end device. As such, it relies on a connection through a parent device for network connectivity. If this connection becomes unstable, the user can initiate the rejoin behavior to coerce the bulb into finding a new, perhaps better parent through which to communicate with the network.

Power cycling the bulb for 10 or more times will force the bulb to leave the network and attempt to find a new network. A side effect of this behavior is that all of the bulb parameters (on/off, level, color control if available) will be reset to the factory defaults.

Lastly, to facilitate manufacturing tests, the lighting reference designs include our manufacturing library CLI plugin for driving the radio in test modes so that technicians may make various RF parameter measurements. This can be tricky because the manufacturing library cannot be called if the bulb is in the middle of scanning for a network. To facilitate this behavior, the manufacturing library CLI plugin has a command to enable the manufacturing library token. If this token has been enabled, the bulb ui will pause 10 seconds before kicking off the network scan in order to allow the test equipment time to invoke the manufacturing library behavior. There are more details on this behavior in the manufacturing library CLI plugin section.

### 7.5.1.2 Configurable Parameters

The bulb user interface has no parameters that are configurable through application builder. It does, however, have some parameters that are configurable through the configuration cluster server. These are the minimum and maximum on time for counting power up pulses, as well as the default bulb power up behavior.

The minimum on time parameter is used to tell the bulb-ui to ignore really short pulses. By default, the plugin will ignore pulses that are shorter than 100 mS. However, setting this parameter, the user can configure the bulb to ignore longer on time pulses so they will not be counted as part of the bulb user interface.

The maximum on time parameter is the amount of time that the bulb will wait before it declares a final value for the number of pulses. This is 2 seconds by default.

Finally, the user can configure the default power up behavior for the bulb. There are three options: turn the bulb on to full brightness on power up, turn the bulb off on power up, and return the bulb to the previous state on power up. By default, the bulb will return to the last known state on power up.

### 7.5.1.3 CLI Commands

The bulb-ui plugin has no CLI commands.

## 7.5.2 Mfg-lib-cli

To assist with the manufacturing process, the lighting reference designs include the manufacturing library as well as the manufacturing library CLI commands. On the factory floor, a lighting module that has been programmed with these images may be used to also make radio measurements.

### 7.5.2.1 Theory of Operation

The purpose of the manufacturing library CLI plugin (or `mfg-lib` CLI plugin for short) is to allow access to commands that will put the radio into test mode so that the radio modules can be tested at manufacturing time with the final bulb firmware images.

### 7.5.2.2 Configurable Parameters

There are no configurable parameters for the manufacturing library CLI plugin.

### 7.5.2.3 CLI Commands

**Table 7.1. Commands for the Manufacturing Library Command Line Interface (CLI)**

CLI Command	Notes
plugin mfglib mfgenable <0 1>	Set the token to pause bulb-ui plugin from scanning for channels to allow for starting the manufacturing library before the stack operation begins. 0 clears the token, 1 sets the token.
plugin mfglib start <0 1>	Starts the manufacturing library, which will prevent network activity as well as enable the rest of the manufacturing library commands. The argument dictates whether or not to track incoming messages and provide receiver statistics on the incoming packets.
plugin mfglib stop	Stops the manufacturing library.
plugin mfglib set-channel <channel>	Set the radio to the specified channel. Note: channel must be between 11 and 26 inclusive.
plugin mfglib set-power <power> <mode>	Sets the radio power to be used for manufacturing library transmit commands. The mode, enables or disables boost mode for the EM35x series of chips.
plugin mfglib stream <start stop>	
plugin mfglib tone <start stop>	

**Note:** There are additional CLI commands for use with the manufacturing library plugin. They can be explored by typing the command "plugin mfglib <enter>" and seeing a list of valid commands. Above we are detailing the commands that are most commonly useful to a lighting reference design during the manufacturing process.

Here is the procedure for starting the manufacturing library:

1. Connect to the lighting reference design and issue the mfgenable command: "plugin mfglib mfgenable 1"
2. Power cycle the reference design.
3. Within 10 seconds of power cycle, issue the start command "plugin mfglib start 0"

At this point, any of the manufacturing library commands can be safely executed without interference from the networking stack.

For example to take measurements on a modulated tone on channel 15 at a power of 20, issue the following commands:

1. plugin mfglib set-channel 15
2. plugin mfglib set-power 20 0
3. plugin mfglib stream start
4. At this time, take your measurements.
5. plugin mfglib stream stop

**Note:** Transmit power of 20 is only valid for certain flavors of the EFR32 radio chip at this time.

When the manufacturing tests are completed, it is important that you disable the manufacturing library delay at bootup with the following command:

```
plugin mfglib mfgenable 0
```

It is not necessary to issue the manufacturing library stop command unless you wish to initiate normal networking activity without a power cycle as the manufacturing library operation will be reset at the next power cycle.

### 7.5.3 Configuration Cluster Server

The configuration cluster server plugin allows a user to change certain parameters of the light bulb at the time of manufacture. This prevents the need for recompiling the images for small changes to the design. For example, different models of bulb may require that the PWMs be driven at different frequencies or that the transmit power be adjusted to meet FCC requirements. The configuration cluster server allows manufacture time configuration of these parameters so that the same binary image may be used across multiple model numbers.

#### 7.5.3.1 Theory of Operation

This feature uses a Silicon Labs custom cluster to define special Zigbee cluster library commands that contain configuration information for pre-selected parameters of the lighting reference designs. Any Zigbee coordinator (including the Z3Gateway Reference Design) that is set up with the Silicon Labs custom clusters for OTA configuration can send these commands to the lighting reference design to configure these parameters. It is important to note that these parameters are stored in non-volatile memory and will be a new, permanent configuration for the device after it has been configured. Also, one of the parameters is a lock down feature that prevents further configuration of these parameters once the device has been shipped into the field.

The first step in the process is to connect the lighting device to a gateway that has this cluster enabled. Once the lighting design joins the network, the gateway can send it a command to configure any of the configurable parameters. Upon receiving the configuration message from the gateway, the lighting reference design will update its non-volatile memory storage of the configuration parameters. It is important to note that configuration parameters are utilized at bootup, and the changes do not take effect until the next power cycle.



### 7.5.3.2 Configurable Parameters

**Table 7.2. Configurable Parameters**

Name	Creator	Description
OTA_CONFIG_LOCK	0x7000	Token to prevent OTA configuration and OTA RF test commands from working
TX_POWER	0x7001	Sets the TX power for channels 11-24
TX_POWER25	0x7002	Sets the TX power for channel 25
TX_POWER26	0x7003	Sets the TX power for channel 26
MODEL_NAME	0x7004	Configures the model identifier attribute in the basic cluster
MANUFACTURER_NAME	0x7005	Configures the manufacturer name attribute in the basic cluster
HW_VERSION	0x7006	Configures the hardware version attribute in the basic cluster
BULB_PWM_FREQUENCY_HZ	0x7007	Sets the frequency of the PWM signals
BULB_PWM_MIN_ON_US	0x7008	When the bulb is on and the level is set to 1, this is the amount of time per cycle that the PWM will be in the active state
BULB_PWM_MAX_ON_US	0x7009	When the bulb is on and the level is set to the maximum, this is the amount of time per cycle that the PWM will be in the active state
BULB_UI_MIN_ON_TIME	0x700A	When power cycling the bulb, this sets the minimum time the bulb must be on to register as a single power cycle event
BULB_UI_TIMEOUT	0x700B	When power cycling the bulb, this sets the amount of time the bulb must be on to reset the boot counter and take the appropriate action (if any)
BULB_UI_POWER_UP_BEHAVIOR	0x700C	This sets the on/off state of the bulb after a power cycle. It can either be always on (0), always off (1) or last known state (2)

### 7.5.3.3 CLI Commands—Device

The parameters may be configured at manufacture time through direction connection to the lighting design by means of the standard command line interface. Here are a list of these commands:

**Table 7.3. Commands for the Configuration Cluster Server Plugin**

CLI Command	Notes
plugin configuration-server lock <true false>	Will enable/disable the custom OTA commands from the server.  Note: It is important that the bulb firmware be locked before final shipment to end customers.
plugin configuration-server set <creator> {<data>}	Sets the configuration token specified by the creator to the value in data. Note: the data is little endian (i.e. least significant byte first) in the format of {XX XX ... }, where XX is 2 hexadecimal digits specifying a single byte of data.
plugin configuration-server read	Print out the complete list of the configuration tokens as well as their current values.

For example, to configure the device for a PWM frequency of 700 Hz (which is 0x02BC), issue the following command:

- plugin configuration-server set 0x7007 {BC 02}

### 7.5.3.4 CLI Commands—Gateway

The parameters may be configured at manufacture time using over the air commands from a gateway that supports the ota configuration client cluster. This is particularly convenient if the lighting module resides inside a finished light bulb product, and provides flexibility for the user to set up the manufacturing flow.

**Table 7.4. Commands for the Configuration Cluster Client (Gateway) Device**

CLI Command	Notes
zcl mfg-code 0x1002	Sets the manufacturing code to 0x1002, which is the Silicon Labs code used for these custom clusters.  Note: The manufacturing code will remain set until it is explicitly unset, and normal CL commands will not work when in this mode.
zcl mfg-code 0	Unsets the manufacturing code.
zcl ota-config lock	Explicit command to lock the OTA configuration functionality of the reference designs
zcl ota-config read <creator>	Command to read the data from the token specified by the creator value
zcl ota-config setToken <creator> {<data>}	Sets the configuration token specified by the creator to the value in data. Note: the data is little endian (i.e. least significant byte first) in the format of {XX XX ... }, where XX is 2 hexadecimal digits specifying a single byte of data.
zcl ota-config unlock {<unlock key>}	Unlocks the OTA configuration functionality of the reference design. See an FAE for support.
custom send <device entry>	Send the command to the reference design from the Zigbee Gateway Reference Design.

Here is an example of the commands used to set the PWM frequency of a light bulb that exists at location 0 in the device table to 700 Hz.

**Note:** It is assumed the gateway already created a network and that the lighting reference design has joined this network:

1. zcl mfg-code 0x1002
2. zcl ota-config setToken 0x7007 {BC 02}
3. custom send 0
4. zcl mfg-code 0

**Note:** If you are setting multiple configuration parameters using OTA commands, the final "zcl mfg-code 0" is only required after the final message has been sent to the device being configured.

### 7.5.4 Manufacturing Library OTA

This plugin implements the manufacturing library custom cluster from Silicon Labs. It provides a means, through radio messages, to put the device into RF testing mode. In the past, enabling such features on a device required the user to physically alter the device (by cutting the case, adding a connector, and connecting to a ribbon cable). For some tests, this is practically impossible. For others, it will alter the results of the test (such as for FCC tests). To overcome these issues, Silicon Labs developed a way to invoke the manufacturing library commands using radio commands.

### 7.5.4.1 Theory of Operation

Silicon Laboratories has created a custom Zigbee cluster that details RF commands for the purpose of putting the device receiving the commands into manufacturing test mode. Specifically, it can force the device to transmit an unmodulated tone, transmit a modulated tone, or receive packets for a short time to check the RX path of the device in the field.

First, the user must obtain a gateway that supports this functionality. The Z3Gateway application supports the MFGLIB Cluster, although it is possible to use our development environment to create such a gateway.

After the gateway creates the network and the lighting reference design joins the network, the gateway can send commands to the lighting reference design to tell it which RF test to perform and which parameters to use, such as channel and TX power. Each of the RF commands also includes a timeout parameter, which sets the amount of time (in milliseconds) to enable the command. Using a timeout parameter of 0 means that the device will remain in the specified RF test until the next device power cycle.

### 7.5.4.2 Configurable Parameters

There are no configurable parameters for these plugins.

### 7.5.4.3 CLI Commands—Device

There are no device side CLI commands for this plugin.

### 7.5.4.4 CLI Commands—Gateway

The following table details the commands for sending the manufacturing test commands over the air from a gateway that supports this functionality.

**Table 7.5. Commands for the Configuration Cluster Client (Gateway) Device**

CLI Command	Notes
zcl mfg-code 0x1002	Sets the manufacturing code to 0x1002, which is the Silicon Labs code used for these custom clusters.  Note: The manufacturing code will remain set until it is explicitly unset, and normal CL commands will not work when in this mode.
zcl mfg-code 0	Unsets the manufacturing code.
zcl mfglib rx-mode <channel> <power> <time>	Put the device into RX mode. Note: when in RX mode, the device under test will keep track of the number of packets received, the RSSI and the LQI of the first packet received. It is recommended to use a non-zero timeout as these parameters are stored in volatile memory and will be reset at power cycle.
zcl mfglib stream <channel> <power> <time>	Set the radio to transmit a modulated carrier. Note: the carrier will be modulated with a random stream of characters.
zcl mfglib tone <channel> <power> <time>	Set the radio to transmit a modulated carrier.  Note: The carrier will be modulated with a random stream of characters.
custom send <device entry>	Send the command to the reference design from the Zigbee Gateway Reference Design.

Here is an example of the commands used to command the lighting design to transmit a modulated carrier at channel 15 with a power of 20 dBm until the next device power cycle.

**Note:** It is assumed the gateway already created a network and that the lighting reference design has joined this network:

1. zcl mfg-code 0x1002
2. zcl mfglib-stream 15 20 0
3. custom send 0

### 7.5.5 LED-RGB-pwm

This plugin takes commands from the on/off, level control, and color control plugins. It will then compute the correct PWM drive values to implement the current color mode output based on the on/off and level parameters for three separate strings of red, green, and blue LEDs.

### 7.5.6 LED-Temp-PWM

This plugin takes commands from the on/off, level control, and color temperature server plugins. It then computes the correct PWM drive values to implement the current temperature and level. It assumes that it is driving two independent strings of LEDs, where one LED string is a warm LED string and one LED string is a cool LED string.

**Note:** The RD-0085 and RD-0098 hardware have color temperature LEDs at 2200LK and at 5000K. The PWM signals are used to blend these two colors to create intermediate color temperatures. The RD-0035 hardware does not have color temperature LEDs and cannot demonstrate this feature.

### 7.5.7 LED-Dim-PWM

This plugin takes commands from the on/off cluster and the level control cluster and translates them into the PWM drive values for a single PWM. It is intended for use by the dimmable LED light bulb reference design. Because it controls a single LED string, it does not support any of the color modes.

### 7.5.8 Diagnostic Server

This plugin has been extended to record the RSSI, LQI, and number of retransmissions per APS packet sent. The plugin will populate the appropriate attributes of the diagnostic server so that they may be read by other devices or reported by the bulb.

### 7.5.9 Bulb-pwm-cli

The lighting reference designs include pwm drivers designed to drive most standard LED bulb architectures. The bulb pwm CLI plugin facilitates the testing of this hardware. It includes commands for setting the PWMs to different output values. Note: these values are hardware specific, so this plugin will require knowledge of your specific device's chip architecture and output frequency.

### 7.5.9.1 Theory of Operation

The purpose of this plugin is to allow the user, through the CLI, to dial in specific pulse widths in the PWM hardware. This is done by means of a direct serial connection either through the WSTK or over the serial lines in the 10-pin connector.

You can look to the hal bulb pwm driver section below for more details on the PWM operation, but here are a few details you need to know to use this CLI.

By default, the PWMs are set to operate at 1 kHz. The EM35x series of chips will drive the peripheral clock at 6 MHz, which means that it will take 6,000 ticks for the PWM to count through an entire cycle. This means that setting a PWM value of 0 will drive the signal to low the entire time, and setting the PWM value of 6000 will drive it to full high through the entire PWM cycle.

For the EFR32 series of chips, the peripheral clock is running much faster. It will take 38,400 ticks for the PWM to count through an entire cycle at 1 kHz. Therefore, you need to set the PWM to 38,400 to drive the PWM to full on.

Finally, the PWM drive has the concept of a channel handler, or channel for short. Because the lighting design uses multiple PWM channels across multiple PWM peripherals, it was convenient to have a single, monotonic number set for all of the channels. Here is a table for the EFR:

**Table 7.6. Default PWM Channel Configurations—EFR32**

Channel	Function	GPIO Port	VPIO Pin	Timer	Timer Channel
1	White	B	15	Timer 0	0
2	Low temp White	B	14	Timer 0	1
3	Status LED	B	11	Timer 0	2
4	Red	C	10	Timer 1	0
5	Green	D	15	Timer 1	1
6	Blue	C	11	Timer 1	2

**Table 7.7. Default PWM Channel Configurations—EM35x**

Channel	Function	GPIO Port	VPIO Pin	Timer	Timer Channel
1	White	B	1	Timer 2	1
2	Red	B	2	Timer 2	2
3	Green	B	3	Timer 2	3
4	Blue	B	4	Timer 2	4

### 7.5.9.2 Configurable Parameters

There are no configurable parameters for this plugin.

### 7.5.9.3 CLI Commands

**Table 7.8. CLI Commands for Driving the PWM Outputs**

CLI Command	Notes
plugin bulb-pwm set <channel> <value>	This will set the PWM of the specified channel to the specified value.

If you wish to set the blue PWM to maximum value for the EFR32, use the following command:

- plugin bulb-pwm set 6 38400

And if you wish to set the blue PWM to maximum value for the EM35x, use the following command:

- plugin bulb-pwm set 4 6000

### 7.5.10 Hal-Bulb-PWM-Driver

The hal bulb PWM driver configures the PWM hardware to drive external LEDs. These PWM signals can be connected directly to bulb power supplies to drive the LED strings to the levels required by the Zigbee communication.

#### 7.5.10.1 Theory of Operation

The pinouts and hardware is specified by a board header file. After creating a new lighting reference design project in Simplicity Studio, there will be a file of the form <project>\_board.h, also known as the board header file. It will contain a section in which you may reconfigure the PWM definitions depending on your final hardware. The default settings are designed to work with the lighting reference design hardware, and any changes will make this project incompatible with the lighting reference design.

Upon boot up, the hal bulb PWM driver will use the settings in the board header file and use those settings to set up the hardware: which GPIO drives which PWM channel, and how are these tied to the functionality required of the lighting reference design (such as the white LED).

Also as part of the startup configuration, the hal bulb PWM driver will make calls into the protocol to obtain the PWM drive frequency, which for the lighting reference designs is one of the configurable parameters.

Finally, the hal bulb PWM driver implements a generic blinking function. It provides the ability to turn the LEDs on for a fixed amount of time, turn the LEDs off for a fixed amount of time, blink at a specific rate for a specific number of times, and finally to blink a generic pattern for a specific number of times. Note that at the end of these blinking patterns, the hal bulb PWM driver will make a call into the protocol stack to indicate the blinking finished, and will expect calls from the protocol stack to set the new, non-blinking PWM values appropriate for the on/off, level, and color parameters.

#### 7.5.10.2 Configurable Parameters

There is one configurable parameter for this plugin. The blink pattern is stored in a statically allocated array. The size of this array is configurable through the plugin UI in Simplicity Studio.

#### 7.5.10.3 CLI Commands

See the above plugin for instructions on driving this plugin through CLI interaction.

## 8. Ecosystem Considerations

The lighting reference design firmware examples have been made to be Zigbee 3.0 certifiable. As such, they should work with any of the Zigbee certified Home Automation gateways in the market. However, each gateway may have some individual requirements beyond the Zigbee 3.0 requirements. It is therefore recommended that you contact the individual gateway manufacturers for support in getting your device onto their gateway. Several gateway manufacturers have developer websites for such tasks.

In our testing, we have discovered that virtually all gateway manufacturers require a device to support two optional attributes in the Basic cluster: manufacturer name and model identifier. While our reference designs have values for these attributes that are reflective of the Silicon Laboratories reference design work, it is expected that each individual customer would customize these attributes for their own product.

In our testing, we have also discovered that a few of the gateway manufacturers use the manufacturer name and model identifier (among other things) to implement a white list. A white list is a list of approved devices that the gateway will allow onto its network. If your device is not on the white list, it will not be allowed to join the network. For these gateways, you will need to contact the gateway manufacturer directly to add your manufacturer name and model identifier to their gateway's white list.



## 9. Engineering Tests

This reference design has undergone the following product testing for pre-certification purposes:

- Zigbee 3.0
- FCC Emissions
- Antenna Radiation Patterns

### 9.1 Zigbee 3.0

All modules have gone through a preliminary Zigbee 3.0 testing to verify their compliance with the Zigbee standard.

### 9.2 FCC Emissions Testing

All modules have gone through a preliminary FCC pre-scan testing to verify their compliance with FCC Part 15 restrictions. Such testing is done to ensure that customers are given a design that can pass FCC, but it does not transfer FCC compliance to modules built according to this reference design, even if the design is copied exactly.

## 10. Manufacturing Tests for RD-0085-0101 and RD-0098-0101

This section describes how a user can develop a low cost manufacturing test methodology in order to test products that utilize this reference design. It will outline recommended test equipment and test procedure.

### 10.1 Test Coverage

By following these guidelines, the following items will be tested and verified for functionality.

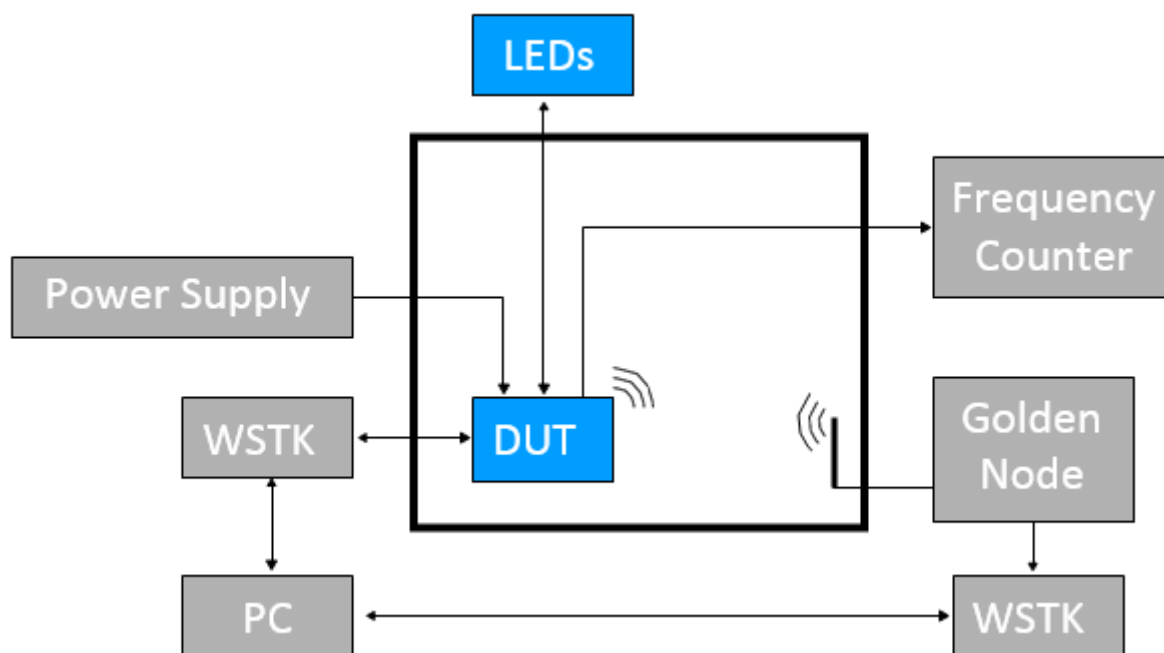
- RF TX/RX Performance
- Current consumption
- PWM functionality
- Common Zigbee operations
- Crystal frequency offset measurement

### 10.2 Test Equipment List

The test system is composed of the following components:

- Desktop PC
- RF Shielded Box
- Power Supply (Agilent E3646A)
- 2x WSTK programming/packet trace adapters (Silicon Labs)
- Golden Node to act as a Zigbee coordinator
- Frequency counter accurate to 1 ppm
- Device under test (DUT)

### 10.3 Test System Diagram



#### 10.4 Test System Connection Procedure

Wire the system as indicated above. All WSTK adapters will be connected to the PC through USB. One adapter will connect to the DUT to both reprogram and interact with the DUT, and the second one will connect to the Golden Node for testing network functionality. The WSTK adapters may need to be powered off during some tests. This can be done by adding a power switch to the test jig, or by using the power selection switch already on the WSTK.

Indicators as simple as LEDs may be used to verify PWM functionality. The only thing in the shielded box should be connectors linked to the DUT, and connectors for 50 ohm antennas.

#### 10.5 Configuring the WSTK Adapters

For this setup, the PC will communicate to the WSTK Adapter over USB. Each adapter has a unique J-Link Serial Number that is displayed on the LCD screen. This number will be used to reference to the specific WSTK and communicate with it.

#### 10.6 Communicating with Targets

Communication to the WSTK needs to be established through the Simplicity Commander tool. This includes commands to program the DUT or shut off programming capabilities of the WSTK itself, if that is required.

Communication to the DUT can be established through serial connection with the Virtual COM port of the WSTK. All device behavior tests are performed by sending specific commands which the device interprets and acts upon.

#### 10.7 Additional Details

For a more thorough checklist, please refer to Application Note AN700 or contact Silicon Labs (<http://www.silabs.com>) for more information.

## 11. Manufacturing Tests for RD-0035-0101

This section describes how a user can develop a low cost manufacturing test methodology in order to test products that utilize this reference design. It will outline recommended test equipment and test procedure.

### 11.1 Test Coverage

By following these guidelines, the following items will be tested and verified for functionality:

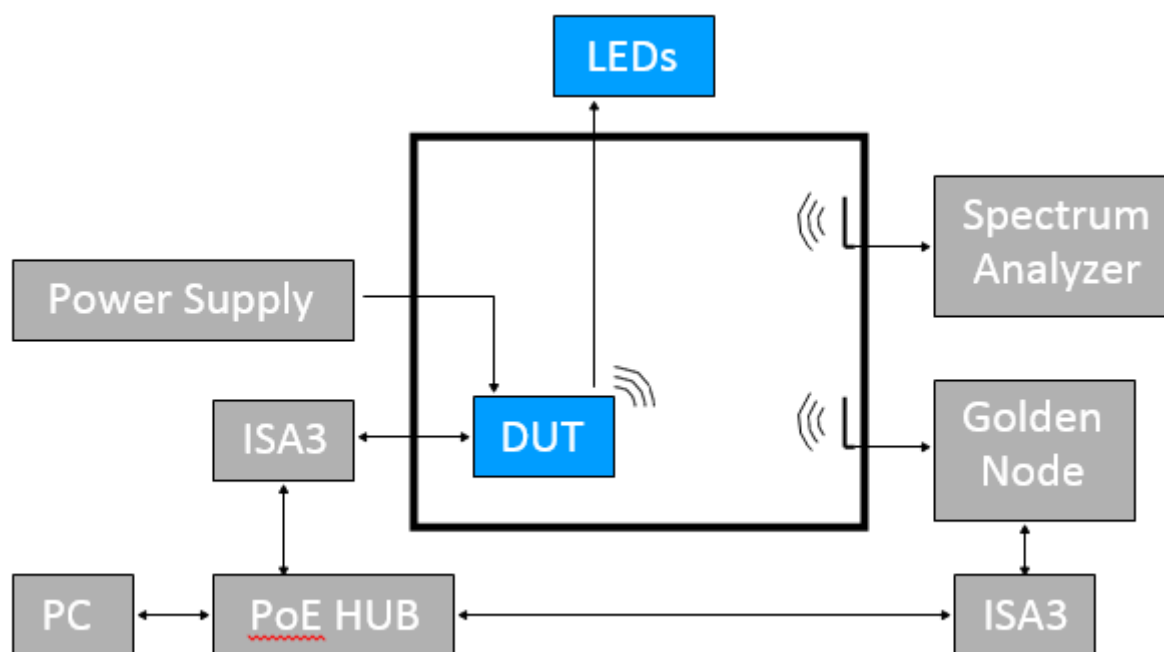
- RF TX/RX performance
- Current consumption
- PWM functionality
- Common Zigbee operations
- Frequency offset

### 11.2 Test Equipment List

The test system is composed of the following components:

- Desktop PC
- RF Shielded Box
- Power Supply (Agilent E3646A)
- PoE Hub (Netgear Prosafe 108P)
- 2 x ISA3 programming/packet trace adapters (Silicon Labs ISA3)
- Golden Node
- Spectrum Analyzer or frequency counter accurate to 1 ppm
- Device under test (DUT)

### 11.3 Test System Diagram



#### 11.4 Test System Connection Procedure

Wire the system as indicated above. Both ISA3 adapters and the PC will be connected to the PoE hub. One ISA3 adapter will connect to the DUT to both reprogram and interact with the DUT, whereas the second ISA3 adapter will connect to the Golden Node for testing network functionality. The ISA3 Adapters connected to the DUT should never be turned off. Instead, power for the DUT will be provided by the Power Supply for live current measurements.

Indicators as simple as LEDs may be used to verify PWM functionality. The only thing in the shielded box should be connectors linked to the DUT, and connectors for 50 ohm antennas.

#### 11.5 Configuring the ISA3 Adapters

For this setup, the PC will communicate to the ISA3 Adapter with an Ethernet connection over Telnet. To do this, first the ISA3 adapters need to be configured to use a static IP address. For instructions on how to set up static IPs, reference sections 6.3 and 6.4 in Silicon Labs' UG110 User's Guide. Use the Admin interface over USB to set up the ISA3 adapters.

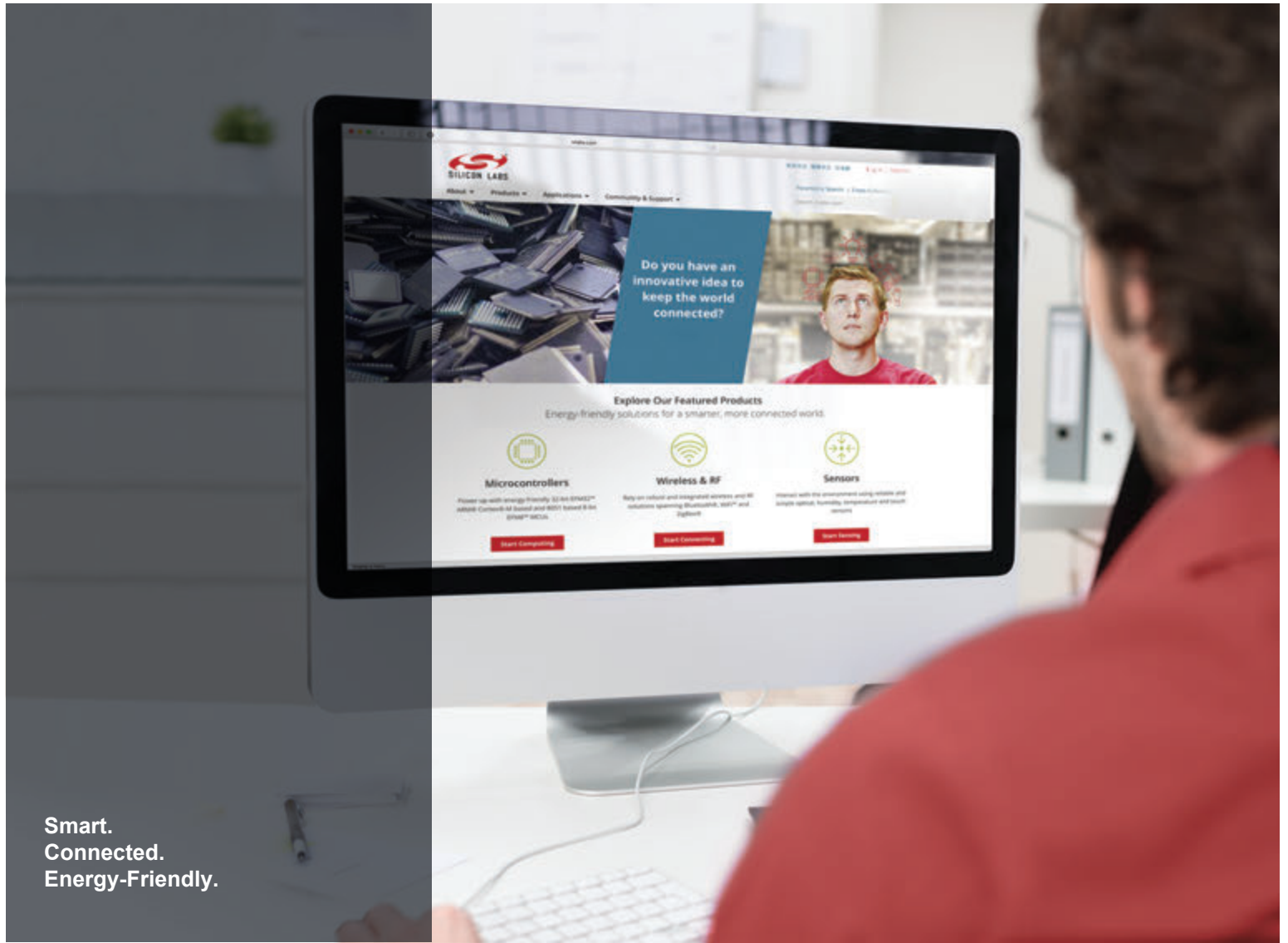
#### 11.6 Communicating with Targets

Through the Telnet interface, both the ISA3 adapters and their respective connected targets can be communicated with simultaneously. To communicate with the ISA3 adapters, a Telnet session should be opened to its static IP at port 4902. To communicate with a target connected to the ISA3 adapter through virtual UART, connect to the ISA3's static IP at port 4900.

Once connected, press Enter, or send a '\r\n' sequence which will result in a prompt ending with a '>' character.

#### 11.7 Additional Details

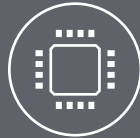
For a more thorough checklist, please refer to Application Note AN700 or contact Silicon Labs (<http://www.silabs.com>) for more information.



Smart.  
Connected.  
Energy-Friendly.



**Products**  
[www.silabs.com/products](http://www.silabs.com/products)



**Quality**  
[www.silabs.com/quality](http://www.silabs.com/quality)



**Support and Community**  
[community.silabs.com](http://community.silabs.com)

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

**Trademark Information**

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



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

<http://www.silabs.com>