



QSG102: Thread Border Router Add-On Kit Quick-Start Guide

The Silicon Labs Thread Border Router Add-On Kit (RD-0004-0201) adds a border router demonstration system to the EM35x Development Kit (EM35x-DEV) or EFR32 Mighty Gecko Wireless SoC Start Kit (SLWSTK6000A).

This document describes the Thread Border Router Kit, how to setup required hardware and install required software, and how to demonstrate the Thread Border Router.

This document refers to the Thread Border Router version 0.7.1, which requires Silicon Labs Thread stack version 2.2.1 GA, or later.

KEY FEATURES

- Set up the Thread Border Router Hardware
- Set up the Thread End Node Hardware
- Install Simplicity Studio
- Install the Border Router Software
- Install the Thread End Device Software
- Install the Android or iOS Device Software
- Demonstrate the Thread Border Router
- Resources

1 Introduction

A border router is an essential component of a Thread network that manages the traffic between the Thread network and adjacent IP networks. A border router facilitates a number of key capabilities including the following:

- Commissioning of Thread nodes
- GUA (Global Unicast Address) or ULA (Unique Local Address) assignment
- IP routing between the Thread network and adjacent IP networks

The Silicon Labs Thread Border Router Add-On Kit (RD-0004-0201) adds a border router demonstration system to the EM35x Development Kit (EM35x-DEV) and EFR32 Mighty Gecko Wireless SoC Starter Kit (SLWSTK6000A). This document describes:

- Setting up the Thread Border Router Hardware
- Setting up the Thread End Node Hardware
- Installing Simplicity Studio
- Installing the Border Router Software
- Installing the Thread End Device Software
- Installing the Android or iOS Device Software
- Demonstrating the Thread Border Router
- Accessing other resources

This version of the Thread Border Router offers:

- Important bug fixes
- Support for EFR32 Mighty Gecko Wireless SoC Start Kit
- Support for the Thread Group commissioning app for Android or iOS
- Expanded support for IPv6 end-to-end routing
- Support for package manager installation
- Support for command line updating of NCP software

Typical Thread Border Router setups are shown in the following figures, and include:

- Thread Border Router with Wi-Fi Access Point and Web Server
- Thread End Devices such as EM3588 Development Kit or EFR32 Mighty Gecko Wireless SoC Starter Kit
- Web Browser Device such as a handset, tablet or PC
- Commissioning device such as an Android or iOS handset
- IPv6 Ethernet Connectivity

You may choose to commission end devices using the Thread Group commissioning app for Android or iOS, or you may choose to use pre-commissioned devices that do not require the commissioning app. The default is the pre-commissioned configuration. Instructions for configuration are provided in section **Install the Border Router Software** and **Install the Thread End Device Software**.

2 Set Up the Thread Border Router Hardware

2.1 Unpack the Thread Border Router

The Thread Border Router Add-On Kit contains:

- Raspberry Pi 2 Model B with case, also referred to as the Thread Border Router
- MicroSD card with Raspberry Pi 2 B operating system pre-installed
- Edimax USB Wi-Fi Adapter, providing a Wi-Fi network for demonstration purposes
- CEL EM3588 USB Thread Adapter NCP (network co-processor)
- Power supply



ATTENTION: Observe precautions for handling electrostatic sensitive devices.

2.2 Connect and Power On the Thread Border Router

The Thread Border Router hardware is preassembled out of the box. The remaining steps are:

1. Connect the USB Wi-Fi Adapter to one of the Raspberry Pi's USB ports (if not already connected).
2. Connect the CEL EM3588 USB Thread Adapter NCP to one of the Raspberry Pi's USB ports.
3. Connect a monitor to the HDMI port and a keyboard to a free USB port. It is possible to login remotely to the Thread Border Router with a utility such as PuTTY or ssh. Refer to *UG116: Developing Custom Border Router Applications*, for more information.
4. Plug in the Thread Border Router power supply. In the following steps you will power the Thread Border Router on and off by connecting and disconnecting its power supply. The red power LED will illuminate on the Border Router and the green activity LED will blink until the boot process has completed.

3 Set Up the Thread End Node Hardware

The EM35x-DEV or SLWSTK6000A development kit is required to set up Thread End Node hardware and obtain a software registration and support key. Three EM35x breakout boards from the EM35x-DEV Development Kit are shown as Thread End Nodes in the first figure below, and three wireless starter kit mainboards from the EFR32 Mighty Gecko Wireless SoC Starter Kit are shown in the second figure below. Kits may be ordered from <http://www.silabs.com/products/wireless/mesh-networking/zigbee/Pages/zigbee.aspx>.

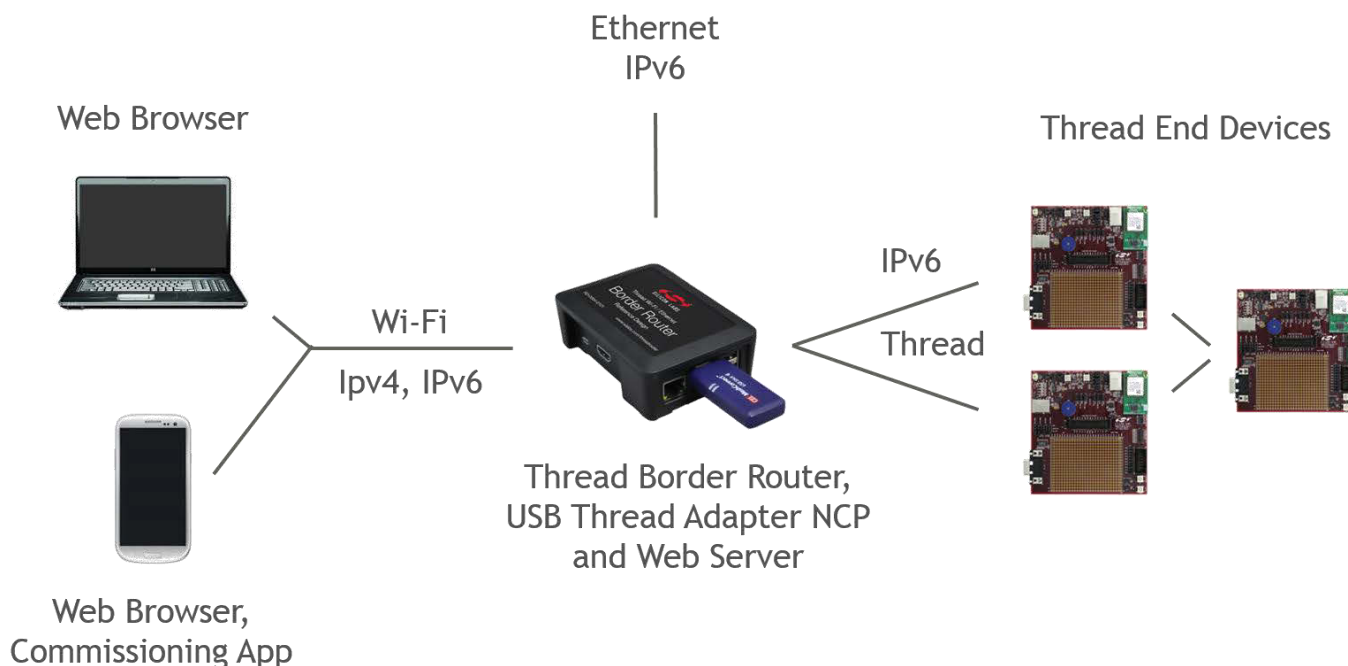


Figure 1. Thread Border Router System Components with EM35x-DEV Development Kit

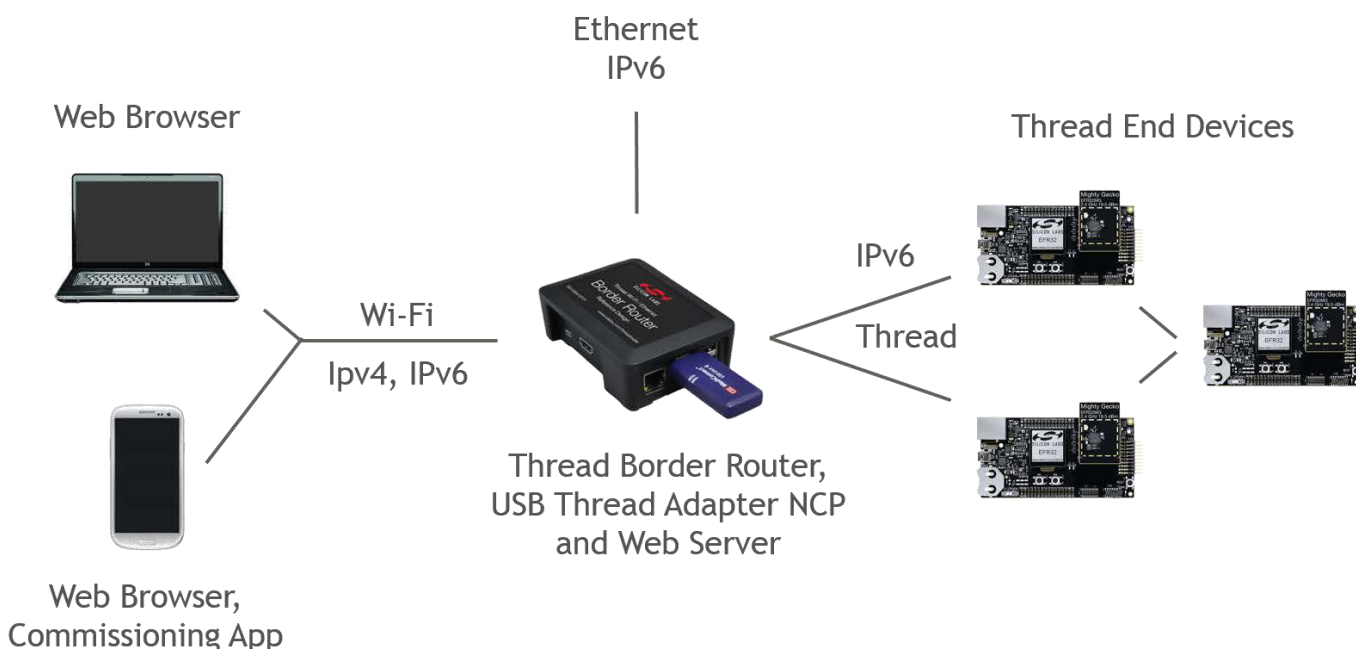


Figure 2. Thread Border Router System Components with EFR32 Mighty Gecko Wireless SoC Starter Kit

3.1 Option 1: Order the EFR32 Mighty Gecko Wireless SoC Starter Kit (Preferred)

This is the preferred kit for Thread End Nodes because of the performance improvements available with the EFR32 Mighty Gecko. The EFR32 Mighty Gecko Wireless SoC Starter Kit includes the following:

- 3 x Wireless starter kit mainboard
- 3 x EFR32MG 2.4 GHz 19.5 dBm radio board
- 3 x EFR32MG 2.4 GHz 13 dBm radio board
- AA Battery Board (supports running +19.5 from battery)
- Cables and batteries
- EFR32MG Getting Started Card
- *QSG113: Getting Started with Silicon Labs Thread*

3.2 Option 2: Order the EM35x-DEV Development Kit with EM3588 modules

The EM35x-DEV Development Kit with EM3588 modules includes the following:

- (3) EM3588 Radio Control Module Board
- (3) EM35x Breakout Board
- (3) EM35x Debug Adapter (ISA3)
- EM35x Module Variety Pack (1 Box with 6 Modules)
- Radio Frequency Cable
- (3) Packet Trace Port Cable
- (3) Data Emulation Interface Cable
- (3) Power Supply and Battery Pack
- 8 Port Switch with 4 x Power over Ethernet (PoE) Ports
- *QSG101: Getting Started with the EM35x Development Kit (included in development kit)*
- *QSG113: Getting Started with Silicon Labs Thread*

IMPORTANT: Kits purchased before February 2015 shipped with EM357 modules. These kits need to be upgraded with EM3588-MLR-AN-C or EM3588-M-AN-C modules to include to support Thread. Modules may be ordered from: <http://www.silabs.com/products/wireless/mesh-networking/Pages/em35x-mesh-networking-kits-zigbee-thread.aspx>.

3.3 Register on the Support Portal

Registration allows access to the support portal, and is a necessary step to access required software. The instructions that come with the EM35x Development Kit and the EFR32 Mighty Gecko Wireless SoC Starter Kit include a unique registration key and explain the registration procedure.

3.4 Set Up Hardware

Refer to *QSG113: Getting Started with Silicon Labs Thread* for instructions on setting up EFR32 Mighty Gecko Wireless SoC Starter Kit and EM35x-DEV Development Kit. The instructions for the EM35x-DEV Development Kit also reference *QSG101: Getting Started with the EM35x Development Kit (included in development kit)*.

4 Install Simplicity Studio

Simplicity Studio is required to flash Thread end node firmware. Refer to *QSG113: Getting Started with Silicon Labs Thread*, for a detailed tutorial. The tutorial includes installation and configuration instructions, and describes the process for compiling and running an example for the EM35x or EFR32MG devices. Simplicity Studio is required for the steps in section **Demonstrate the Thread Border Router**.

5 Install the Border Router Software

The following instructions describe how to install software on the USB Thread Adapter NCP and the Border Router hardware.

5.1 Install the Thread Border Router Operating System and Packages

The Thread Border Router package include the border router application, IP driver application, commissioning application, web application, and all required drivers and support packages. **Note:** For these steps it is recommended that you do not remote login to the Thread Border Router.

1. Power down the Thread Border Router and remove the SD card.
2. Install the Raspbian Jessie Lite operating system on the SD card as described here: <https://www.raspberrypi.org/downloads/raspbian/>. **Note:** The link provided above points to the latest version of Raspbian Jessie. However, due to driver requirements the 5-31-16 version must be used, and can be downloaded here: http://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2016-05-31/.
3. Install the SD card in the Thread Border Router and power it on.
4. Connect the Raspberry Pi's Ethernet port to the Internet with an Ethernet cable.
5. Login with the default username "pi" and password "raspberrypi."
6. Append the following text to the end of /etc/apt/sources.list:

```
deb http://devtools.silabs.com/solutions/apt/ jessie main
```

7. Configure the keys.

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 90CE4F77
```

8. Install the silabs-border-router package. Note that you will be asked to accept the Thread Border Router license agreement.

```
$ sudo apt-get update
$ sudo apt-get install silabs-border-router
$ sudo reboot
```

9. Configure the border router for commissioning or pre-commissioned operation (default). Edit /etc/siliconlabs/border-router.conf and set `USE_COMMISSIONER 1` for commissioning or `USE_COMMISSIONER 0` for pre-commissioned operation. Be certain this choice matches the choice in **Install the Thread End Device Software**.

5.2 Install the USB Thread Adapter NCP Software

The NCP software may be installed from the console with the following steps.

1. Stop the border router applications.

```
$ sudo service border-router-apps stop
```

2. Start the bootloader. The red LED will illuminate and green LED will flash.

```
$ cd /opt/siliconlabs/threadborderrouter/tools/ncp-updater
$ sudo ./bootload-ncp-util-app --uart /dev/ttyUSB0 --launchbootloader
```

3. Update the NCP software and reboot. The red LED will flash.

```
$ sudo ./bootload-ncp-uart-app /opt/siliconlabs/threadborderrouter/firmware/ncp-uart/em3588/ncp-uart-hw-em3588.ebl 0 -1 -p /dev/ttyUSB0
$ sudo reboot
```

6 Install the Thread End Device Software

Pre-compiled Thread end node firmware files are distributed with the Thread Border Router file system and provide options for using commissioning or pre-commissioned configuration. Be certain the selection matches the one made in **Install the Border Router Software**. These files may be transferred from the border router to a host with a utility such as WinSCP or scp as described in *UG116: Developing Custom Border Router Applications*.

Device	Programming File (Commissioning)
EM3588	/opt/siliconlabs/threadborderrouter/firmware/sensor-actuator-node/em3588/ sensor-actuator-node-commissioning-em3588.s37
EFR32MG	/opt/siliconlabs/threadborderrouter/firmware/sensor-actuator-node/efr32mg1p232f256gm48/ sensor-actuator-node-commissioning-efr32mg1p232f256gm48.s37

Device	Programming File (Pre-commissioned)
EM3588	/opt/siliconlabs/threadborderrouter/firmware/sensor-actuator-node/em3588/ sensor-actuator-node-pre-commissioned-em3588.s37
EFR32MG	/opt/siliconlabs/threadborderrouter/firmware/sensor-actuator-node/efr32mg1p232f256gm48/ sensor-actuator-node-pre-commissioned-efr32mg1p232f256gm48.s37

A bootloader is required, and is available in the border router file system.

Device	Bootloader
EM3588	/opt/siliconlabs/threadborderrouter/firmware/sensor-actuator-node/em3588/ app-bootloader-local-storage.s37
EFR32MG	/opt/siliconlabs/threadborderrouter/firmware/sensor-actuator-node/efr32mg1p232f256gm48/ app-bootloader-spiflash.s37

IMPORTANT: The device should be erased before programming.

7 Install the Android or iOS Device Software

An Android or iOS handset or tablet with the Thread Group commissioning app is required to commission Thread end devices when commissioning is selected as described in **Install the Border Router Software** and **Install the Thread End Device Software**. The Thread Group commissioning app is distributed in the Google Play store and is also distributed in the Thread Border Router file system. The file may be transferred from the border router to a host with a utility such as WinSCP or scp as described in *UG116: Developing Custom Border Router Applications*. To access the source for the commissioning app, refer to the Thread Group Bitbucket repository <https://bitbucket.org/threadgroup/>, and request access to the repository by through the Thread Group.

Device	Programming File
Android 5.0+	/opt/siliconlabs/threadborderrouter/commissioning/thread-commissioning-app-1.01.01.apk

8 Demonstrate the Thread Border Router

8.1 Start the Border Router

Power on the Thread Border Router.

8.2 Start the End Nodes

Power on each Thread End Node.

Each device has a 64-bit EUI and a fixed, eight-digit, base32 joining passphrase (also called the Joining Device Credential) derived from the EUI, both of which are required by the commissioning application. The EFR32MG wireless starter kit mainboard display will indicate the EUI and joining passphrase. The joining passphrase for the EM35x and EFR32MG kits prints to stdout on boot, and the EUI is available with the `info` command.

The EFR32MG wireless starter kit mainboard display will indicate the type of node. LED1 on the EM35x and EFR32MG kits also indicate the type of node:

- Very fast blink with pause (either on or off) = joining
- Fast blink = router
- Slow blink = end node
- Alternating fast and slow blink = leader

8.3 Commission the End Nodes with an Android or iOS Device (optional)

Note: This step is only required if commissioning was configured in [Install the Border Router Software](#) and [Install the Thread End Device Software](#).

1. Connect to the Thread Border Router Wi-Fi Access Point. Select the SSID "Silicon Labs Thread ABCD," where "ABCD" is an arbitrary hex number, and enter the passphrase "solutions" when prompted.
2. Launch the Android or iOS commissioning app. The app will search for available Thread networks, request the Thread admin password, and request an EUI and connect code for each Thread End Node. The Thread admin password is set at compile time by the Border Router application and printed on stdout immediately after boot. The admin password is "COMMPW1234" for this version of the Thread Border Router.
3. Enter the EUI and connect code for a Thread End Node as noted in section [Start the End Nodes](#). Press on the Thread End Node Button1 (EM35x kit) or PB1 (EFR32MG kit) to issue a net reset command and scan all available channels for the border router. The Thread End Node will scan all channels twice and then go idle unless the border router is found. **Note:** It is important to enter the connect code on the app first and press Button1/PB1 second.

Hint: Create a QR code at <http://goqr.me> with using the string format `v=1&cc=EL7TUCB0&eui=000B57FFFE07F6B3` where the version `v=1`, joining passphrase `cc= EL7TUCB0`, and EUI `EUI=000B57FFFE07F6B3`.

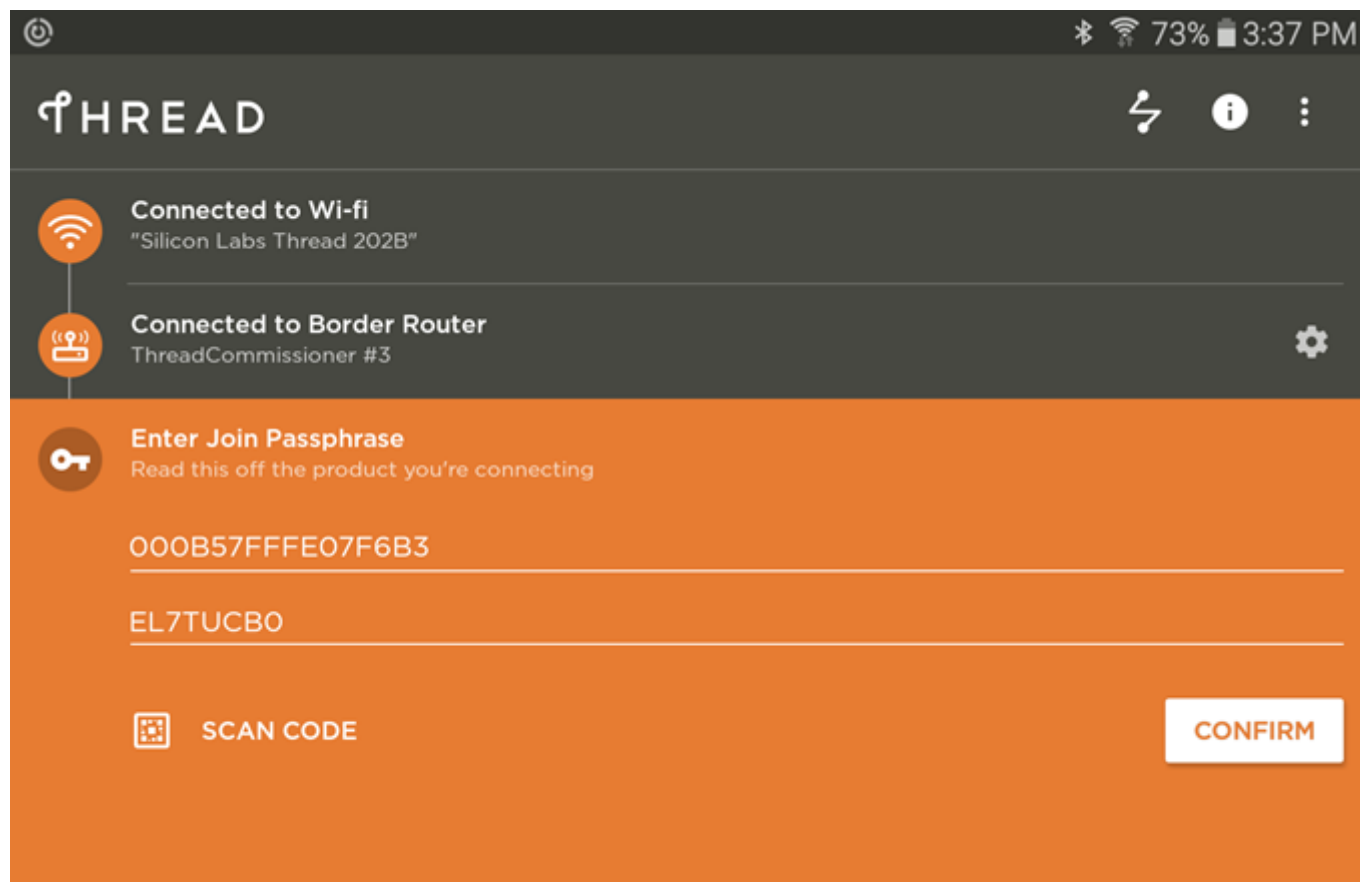


Figure 3. Thread Commissioning App

8.4 Monitor and Control the Thread Devices (Border Router Web Server Method) – Basic Demonstration

1. Connect to the Thread Border Router Wi-Fi Access Point. Select the SSID "Silicon Labs Thread ABCD," where "ABCD" is an arbitrary hex number, and enter the passphrase "solutions" when prompted. Open a web browser and open the Thread Control App at <http://192.168.42.1> or [http://\[2001:db8:8569:b2b1::1\]](http://[2001:db8:8569:b2b1::1]). Note that the Web Browser device can be the Android or iOS device referenced in section **Commission the End Nodes with an Android or iOS Device**.
2. In the Silicon Labs Thread Control App, select **[Discover Devices]**. The Border Router will issue a discover request and each Thread Device will appear in the Control App device list labeled with its IPv6 address and EUI (Extended Unique ID). An example of a Thread End Node (labeled "Device") is shown in the following figure. Communication to and from each Thread Device is possible from the Control App. Temperature is reported every 10 seconds. Button0 (EM35x) or PB0 (EFR32MG) toggle events, LED0 toggle requests, and buzzer requests (EM35x only) are sent immediately.
3. In the Silicon Labs Thread Control App, select **[Clear Devices]** to remove devices from the web server list. This offers a convenient method to confirm which Thread End Nodes actively respond to the discovery request. Note the clearing operation does not affect the Thread network in any way.
4. In the Silicon Labs Thread Control App, select **[Detach Device From Network]** to issue a net reset command to a Thread End Device to remove it from the Thread network. Note that Pressing Button0 (EM35x) or PB0 (EFR32MG) has the same effect. The Thread End Device will immediately scan for the border router, and once found, the Thread End Device will automatically rejoin the thread network. To disable automatic rejoin, clear the commissioning app cache while it is active, or close the commissioning app.
5. In the Silicon Labs Thread Control App, select **[Detach Border Router From Network]** to issue a net reset command to the border router to remove it from the Thread network. Any Thread End Devices on the Thread network will be isolated and must be re-commissioned.

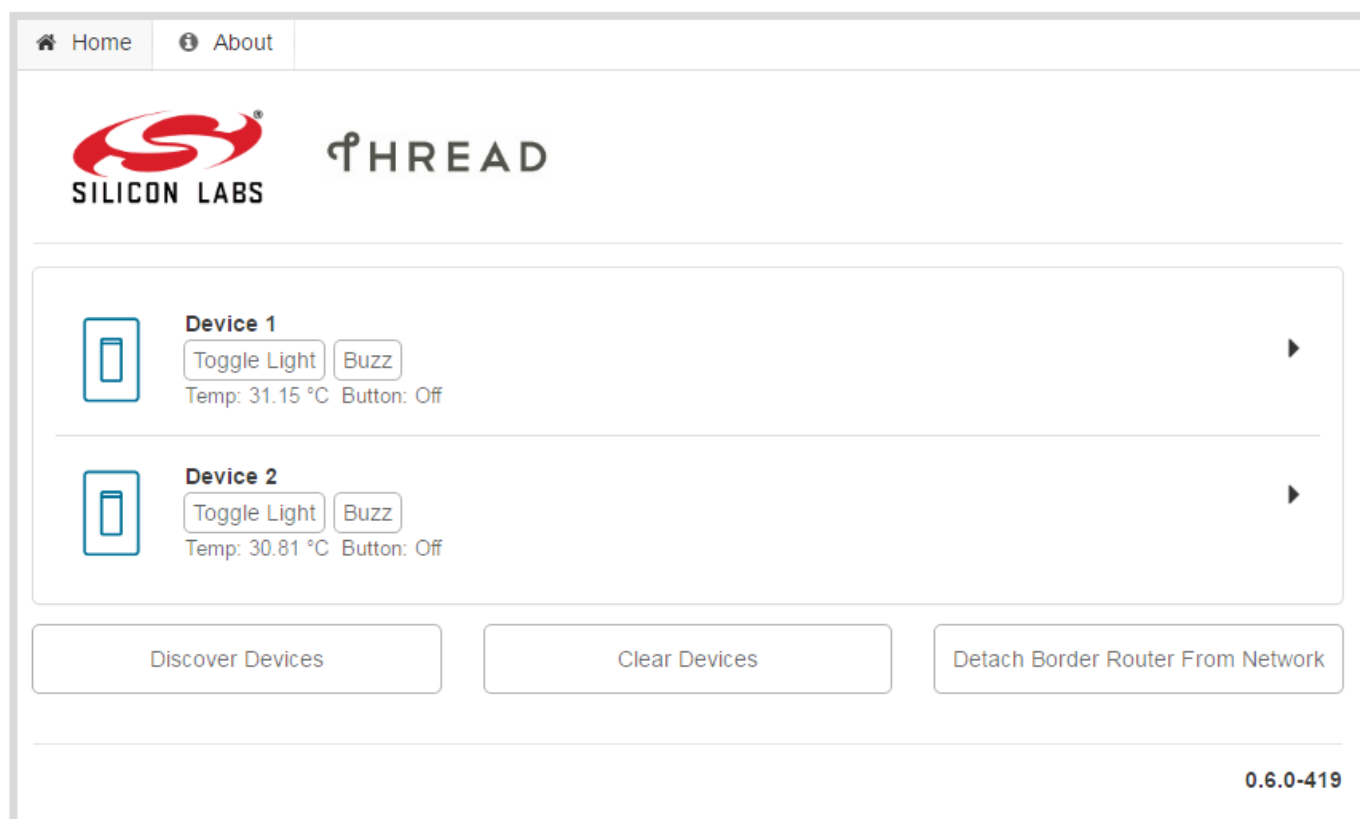


Figure 4. Web Browser Device UI

8.5 Monitor and Control the Thread Devices (CoAP over IPv6 Method) – Advanced Demonstration

It is possible to use a CoAP application such as the Copper plugin for Mozilla Firefox to send CoAP messages to Thread End Nodes. In the examples below the Thread End Node global address is 2001:db8:385:9318:497a:3ebb:2be8:dd8e.

Table 1. CoAP Message Examples

Type	Format	Payload	Note
POST	coap://[2001:db8:385:9318:497a:3ebb:2be8:dd8e]/device/buzzer		EM35x-DEV kit only
POST	coap://[2001:db8:385:9318:497a:3ebb:2be8:dd8e]/device/led		Toggle LED
POST	coap://[2001:db8:385:9318:497a:3ebb:2be8:dd8e]/device/onoffout	0, 1	0 = Off, 1 = On

9 Next Steps

The next steps include modifying and compiling the Thread End Node firmware and Thread Border Router software. Refer to *UG116: Developing Custom Border Router Applications* for more information.

10 Resources

10.1 Getting started with Thread

<http://www.silabs.com/products/wireless/Pages/thread-getting-started.aspx>

10.2 Thread Training Center

<http://www.silabs.com/products/wireless/Pages/thread-networking-learning-center.aspx>

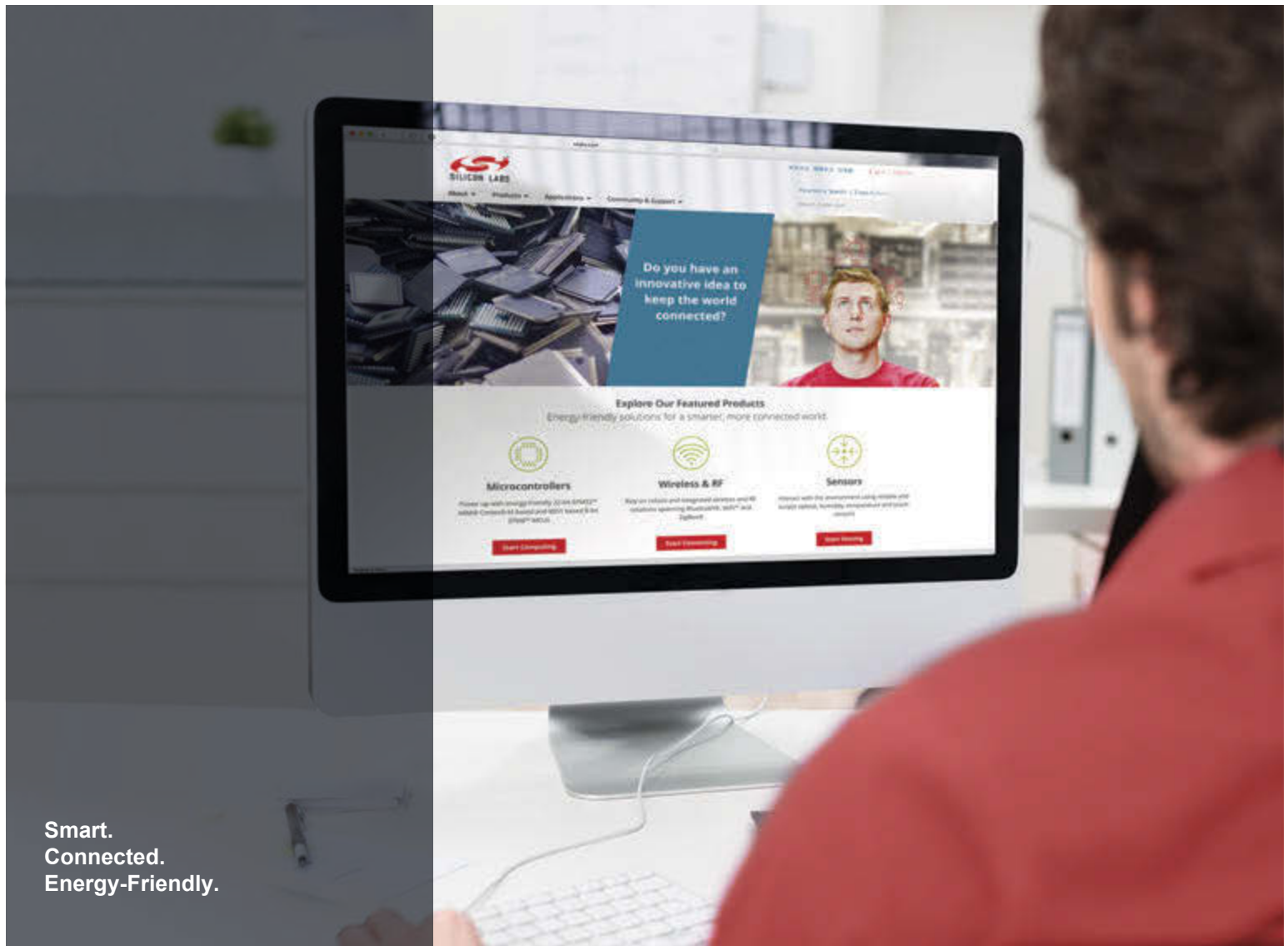
10.3 Documentation

- QSG101: *EM35x Development Kit Quick-Start Guide (included in development kit)*
- QSG113: *Getting Started with Silicon Labs Thread*
- UG103-11: *Application Development Fundamentals: Thread*
- UG116: *Developing Custom Border Router Applications*

10.4 Community & Support

<http://community.silabs.com/>

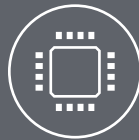
<http://www.silabs.com/support>



Smart.
Connected.
Energy-Friendly.



Products
www.silabs.com/products



Quality
www.silabs.com/quality



Support and Community
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®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® 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>