# AN1256: Using the Silicon Labs RCP with the OpenThread Border Router

A Thread Border Router connects a Thread Network to other IP-based networks, such as Wi-Fi® or Ethernet®. A Thread Network requires a Border Router to connect to other networks. The Border Router provides services for devices within the Thread Network, including routing services for off-network operations. Silicon Labs provides a Border Router Add-On Kit containing a Raspberry Pi device and an example Radio Co-Processor (RCP) application required to build Border Router software.

KEY POINTS

- Build and installation instructions for the Border Router Host
- Build and installation instructions for the RCP images
- OpenThread Border Router configuration information
- OpenThread resources

# 1 Introduction

This application note is intended for software engineers who wish to develop an OpenThread Border Router (OTBR). It assumes some familiarity with OpenThread and basic Thread concepts. For an introduction to OpenThread and information on Thread concepts, visit https://openthread.io/. For information on OTBR setup and installation, refer to https://openthread.io/guides/border-router.

This application notes assumes that you have downloaded Simplicity Studio 5 (SSv5) and the Silicon Labs OpenThread SDK and are generally familiar with the SSv5 Launcher perspective. SSv5 installation and getting started instructions along with a set of detailed references can be found in the online *Simplicity Studio 5 User's Guide*, available on https://docs.silabs.com/ and through the SSv5 help menu. More information about configuring, building, and flashing OpenThread sample applications can be found in *QSG170: Silicon Labs OpenThread Quick Start Guide*.

This application note addresses the following topics.

- **Build and Installation Instructions for the OpenThread Border Router**

  Defines the build and installation procedure for the OpenThread Border Router on POSIX-based platforms.
- **Build and Installation Instructions for the RCP Images**

  Explains the build and installation procedure for the Radio Co-Processor (RCP) image.
- **OTBR Configuration Information**

  Provides OTBR information such as how to configure various Border Router features and the Network Address Translation (NAT64) interface.
- **Additional OpenThread Resources**

  Includes links to OpenThread Resources.

## 1.1 Hardware Requirements

A Thread Border Router has two components:

1. A Raspberry Pi host with Thread Border Router support (Recommended: Raspberry Pi 3 Model B+)
2. A Thread-capable Silicon Labs Radio Co-processor (RCP)

To create the RCP, you need the following:

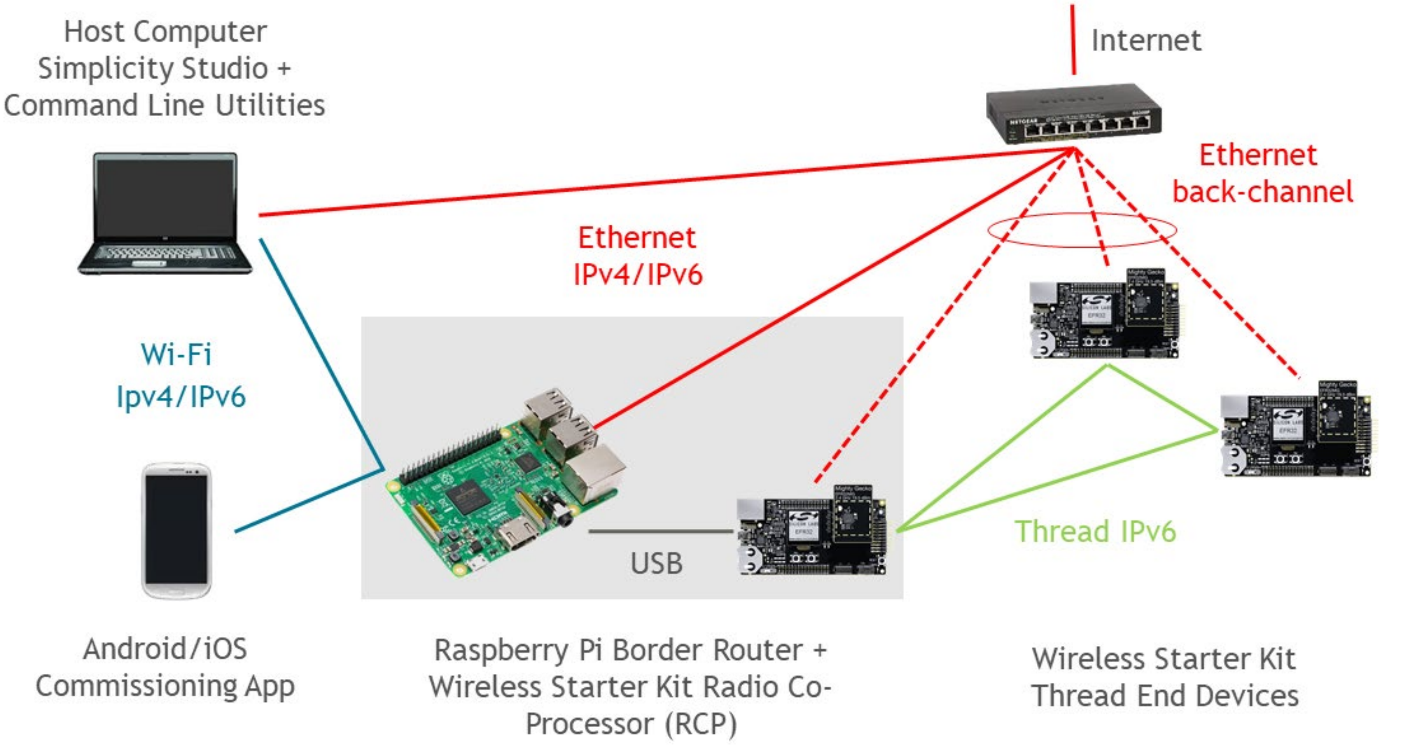
- EFR32™ Mighty Gecko Wireless Starter Kit
- Silicon Labs board capable of Thread communication

  **Note:** See Table 1. Silicon Labs Precompiled RCP Images for a list of supported Silicon Labs boards.

# 2 Build and Installation Instructions for the Border Router Host

## 2.1 Install Hardware

1. Connect each wireless starter kit mainboard and the host computer to an Ethernet switch with an Ethernet cable as shown in the following figure. These connections will permit programming and network analysis of the RCP and end devices. Optionally, end devices may be connected to the host computer by USB rather than Ethernet.



## 2.2 Set Up a Raspberry Pi for Use with RCP

Consult the OpenThread Border Router website for information on how to set up a Raspberry Pi for use with RCP: https://openthread.io/guides/border-router/raspberry-pi-3b.

Complete build instructions are detailed here: https://openthread.io/guides/border-router/build

During the above process, check and note your configuration changes to files such as `/etc/dhcpcd.conf`, `/etc/default/otbr-agent` or `/etc/tayga.conf` in case you need to reapply them later.

**Important Notes**

- Remember to run all OpenThread setup scripts with root-level access.
- When pulling the ot-br-posix repository during setup, check out the branch that Silicon Labs officially supports as part of the GSDK. For more information, refer to the Release Notes for the supported version.
- Check out  the ot-br-posix repository and update it as follows:

```
> git clone https://github.com/openthread/ot-br-posix
```

Refer to Step 1 in https://openthread.io/guides/border-router/build.
Run the following command before proceeding with the installation:

```
> git checkout <commit-hash or branch referred to in the release notes>
> git submodule update --init --recursive
```

### 2.3 Update the OTBR Host

If you decide to update the OTBR, Silicon Labs recommends that you first back up all of your configuration files.

Run the following command before proceeding with the installation:

```
> git checkout <commit-hash or branch referred to in the release notes>
> git submodule update --init --recursive
```

You can use the script `/ot-br-posix/script/update` to update your Raspberry Pi. If you made any configuration changes to files such as `/etc/dhcpcd.conf`, `/etc/default/otbr-agent,` or `/etc/tayga.conf`, you will need to reapply them. The update process restores the configuration to the default settings.

# 3 Build and Installation Instructions for the RCP Images

## 3.1 Use Precompiled RCP Images

Silicon Labs has precompiled images available for these boards with their associated image locations.

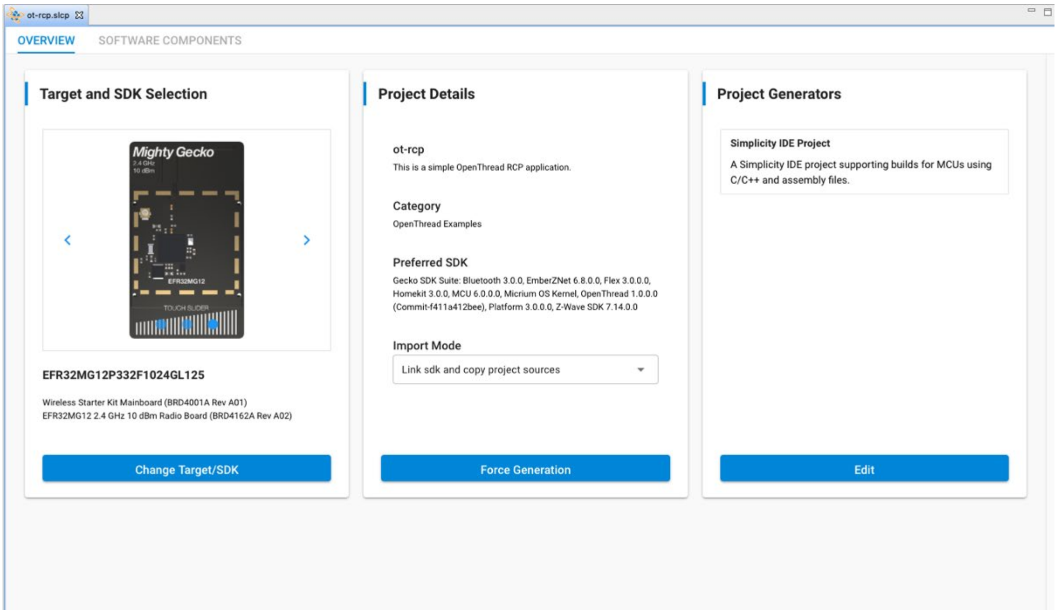
**Table 1. Silicon Labs Precompiled RCP Images**

| Board Name | Image Location |
|---|---|
| brd4161a | protocol/openthread/demos/ot-rcp/ot-rcp-brd4161a.s37 |
| brd4166a | protocol/openthread/demos/ot-rcp/ot-rcp-brd4166a.s37 |
| brd4168a | protocol/openthread/demos/ot-rcp/ot-rcp-brd4168a.s37 |
| brd4180a | protocol/openthread/demos/ot-rcp/ot-rcp-brd4180a.s37 |
| brd4304a | protocol/openthread/demos/ot-rcp/ot-rcp-brd4304a.s37 |

## 3.2 Build RCP Images Using Simplicity Studio 5

Silicon Labs has sample applications for a number of standard OpenThread images.

1. Select **ot-rcp** as an example for the default RCP image for the OpenThread Border Router.
2. With your target part connected to your computer, open SSv5's File menu and select New > Silicon Labs Project Wizard. The Target, SDK, and Toolchain Selection dialog opens. Click **NEXT**.
3. The Example Project Selection dialog opens. Use the Technology Type and Keyword filters to search for **ot-rcp** as an example for the default RCP image for the OpenThread Border Router. Select it and click **NEXT**.
4. The Project Configuration dialog opens. Here you can rename your project, change the default project file location, and determine if you will link to or copy project files. Note that if you change any linked resource, it is changed for any other project that references it. Click **FINISH.**
5. The Simplicity IDE Perspective opens with the Project Configurator open to the OVERVIEW tab. See the online Simplicity Studio 5 User's Guide for details about the functionality available through the Simplicity IDE perspective and the Project Configurator.



EFR32MG12P332F1024GL125

Wireless Starter Kit Mainboard (BRD4001A Rev A01)
EFR32MG12 2.4 GHz 10 dBm Radio Board (BRD4162A Rev A02)

ot-rcp
This is a simple OpenThread RCP application.

Category
OpenThread Examples

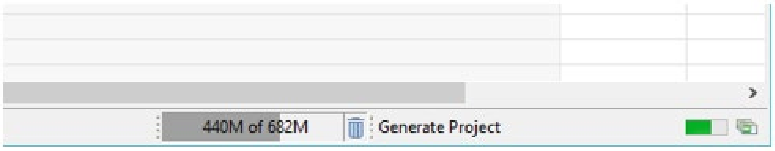
Preferred SDK
Gecko SDK Suite: Bluetooth 3.0.0, EmberZNet 6.8.0.0, Flex 3.0.0.0, Homekit 3.0.0, MCU 6.0.0.0, Micrium OS Kernel, OpenThread 1.0.0.0 (Commit-f411a412bee), Platform 3.0.0.0, Z-Wave SDK 7.14.0.0

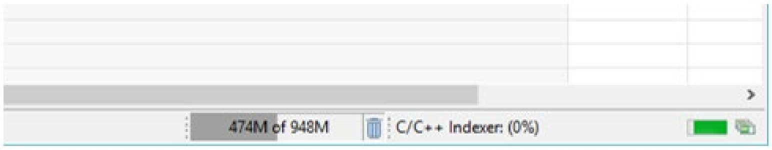
Import Mode
Link sdk and copy project sources

Simplicity IDE Project
A Simplicity IDE project supporting builds for MCUs using C/C++ and assembly files.

For users accustomed to Simplicity Studio 4, in Simplicity Studio 5 project changes are autosaved and project files are autogenerated. The Force Generation function on the OVERVIEW tab is available for the rare case when autogeneration does not occur, usually because a file has been changed outside of SSv5.

6. Make any configuration changes to the software components, as described in the next section. You can see autogeneration progress in the bottom right of the Simplicity IDE perspective. Make sure that progress is complete before you build.



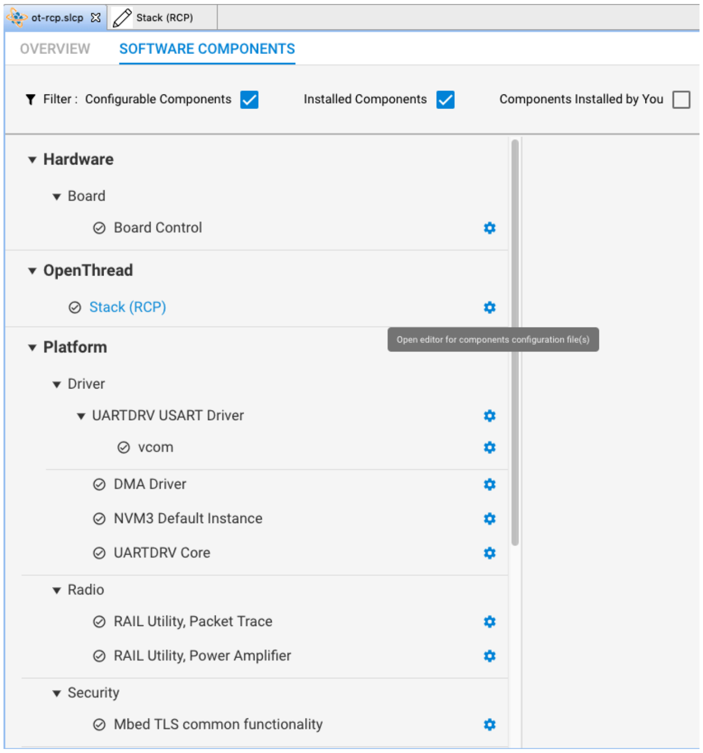


7. Compile and flash the application image as described in *QSG170: Silicon Labs OpenThread Quick Start Guide*.

### 3.3 Configure OpenThread Options in the RCP Images Using Simplicity Studio 5

1. Under the **SOFTWARE COMPONENTS** tab in your RCP project (.slcp), expand the **OpenThread** menu. Select **Stack (RCP)** entry for an RCP build. Click **Configure** to change the settings associated with the OpenThread build.

**Note:** You can select the **Configurable Components** and **Installed Components** checkboxes to filter down to only those components you can configure successfully.



2. You can configure various compile-time settings for your RCP project. The various build options are explained in the OpenThread documentation at https://openthread.io/guides/build.

# 4 OTBR Configuration Information

## 4.1 OTBR Feature Configuration

For information on how to properly configure various OpenThread Border Router features and services, visit:

https://openthread.io/guides/border-router#features_and_services

## 4.2 otbr-agent Configuration Notes

You need to configure the tty port you wish to use for the OTBR to connect your RCP at startup.

1. Look for the tty port of your RCP device. The easiest way to do this is to look for a /tty/dev… entry once the RCP is connected. It should generally either be /dev/ttyUSB0 or /dev/ttyACM0.
2. Edit the `/etc/default/otbr-agent` file and look for the `OTBR_AGENT_OPTS` configuration. Include the tty port name in that parameter as follows:

```
OTBR_AGENT_OPTS="-I wpan0 spinel+hdlc+uart:///dev/ttyACM0"
```

Use the `> sudo ot-ctl state` command on your Raspberry Pi to see the status of the connection between the host and RCP.

## 4.3 NAT64/Tayga Configuration Notes

Use the `/etc/tayga.conf` file to configure the NAT64 interface.

**Note:** You cannot use the default prefix to connect to non-routable IPv4 addresses. You must configure your `/etc/tayga.conf` file to use a non-default prefix.

## 4.4 Using ot-ctl to Configure and Control your OpenThread Border Router

For a full command list, run

```
> sudo ot-ctl help
```

Refer to https://openthread.io/guides/border-router/external-commissioning for examples on how to manually set up a Thread Network and examples on how to enable an external commissioner.

You can run this command to check for a running Thread Network:

```
> sudo ot-ctl state and > sudo ot-ctl ifconfig
```

**Note:** The error message `OpenThread Daemon is not running` indicates a problem with the RCP connection. Check for a valid /dev/tty entry, configured in step 2 in section 4.2 otbr-agent Configuration Notes and check that a valid RCP application was flashed onto the device.

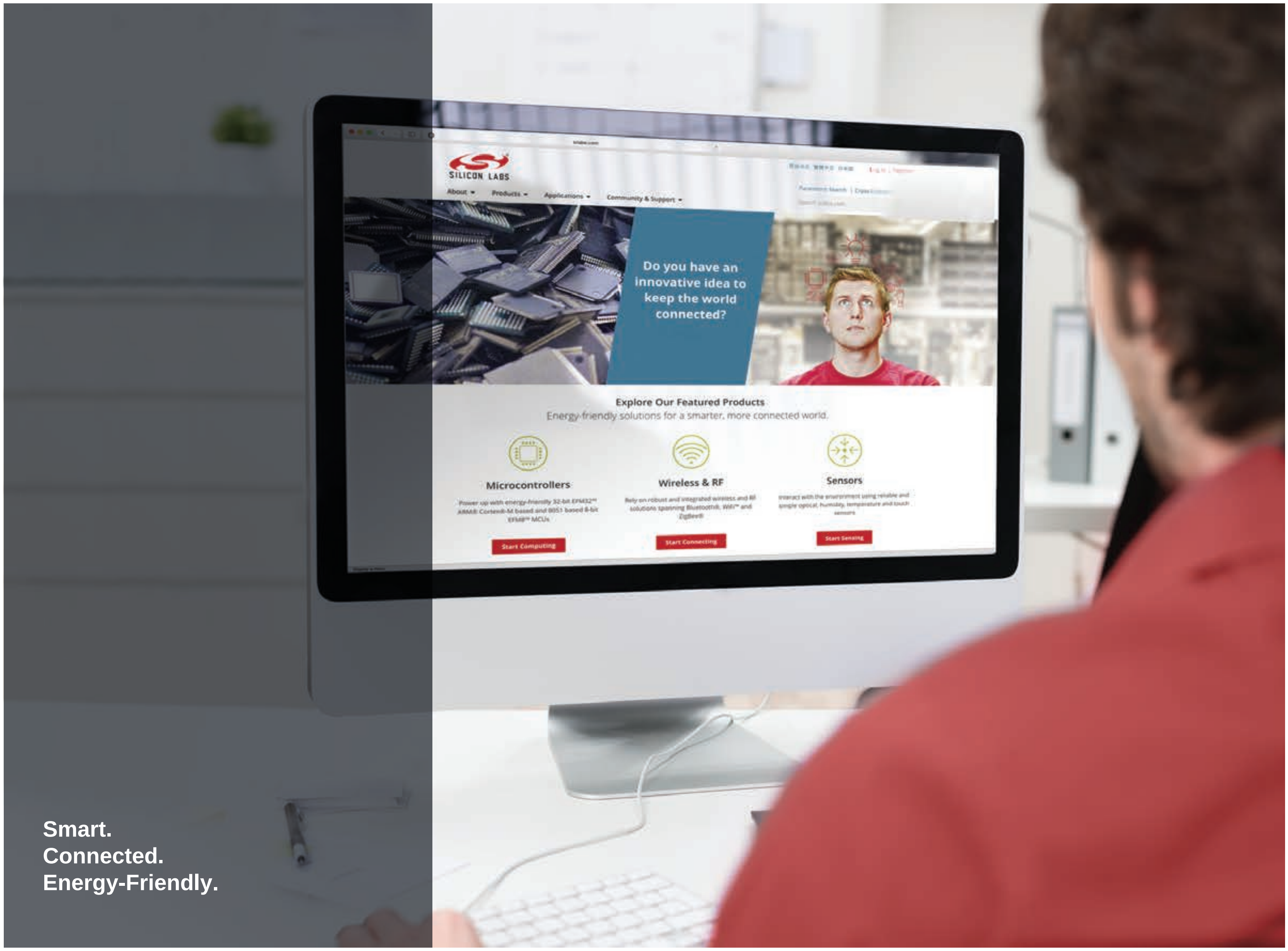
# 5 OpenThread Resources

To find more resources or take advantage of the OpenThread community pages, visit: https://openthread.io/resources

For information about the OpenThread Border Router, visit: https://openthread.io/guides/border-router

Consult these troubleshooting webpages for more information:

- https://openthread.io/guides/border-router/build#verify-services
- https://openthread.io/guides/border-router/access-point#troubleshooting


Smart.
Connected.
Energy-Friendly.
SILICON LABS
Do you have an innovative idea to keep the world connected?
Explore Our Featured Products
Energy-friendly solutions for a smarter, more connected world.
Microcontrollers
Wireless & RF
Sensors
Start Computing
Start Connecting
Start Sensing

**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 to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems 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. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

**Trademark Information**

SILICON LABS

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

**http://www.silabs.com**